

# 容大打印

---安卓端开发文档

## 修订记录

版本号	作者	修订日期	备注
V1.0	Null	2017.01.21	初版
V1.1	薛锦福	2017.10.11	增加串口相关[AP02] ps:AP02 使用的是 Esc 指令
V1.1.1	薛锦福	2017.10.17	增加定制接口，读取打印是否成功状态 状态通过 handler 回传； 参考：本文 <a href="#">2.1.9 状态查询( RP58A-MBU 适用)</a> Ps: 适用于 RP58A-MBU，蓝牙通信
v1.1.2	薛锦福	2018.02.06	增加打印机 Codepage 设置 仅支持 Esc 指令 参考： <a href="#">本文 3.3.12 public void setHsCodepage、</a> <a href="#">本文 3.3.13 public void setHsCodepageForCom</a>

# 目录

容大打印 .....	1
--安卓端开发文档 .....	1
修订记录 .....	1
目录 .....	2
1、PrintDriver 与 APP 通讯方法 .....	6
1.1 PrintDriver 类 .....	6
1.2 创建一个 Handler 子类 ConnStateHandler, 处理 driver 发.....	6
送过来的消息如下: .....	6
1.3 通过以下方法设置 Handler: .....	7
1.4 public void setHandler(Handler handler) .....	7
1.5 Contants 连接状态类的状态标志: .....	7
1.6 蓝牙连接方法 (该方法在 SDK 中无封装, 在此仅做建议使用): .....	7
1.7 USB 连接方法 (该方法在 SDK 中无封装, 在此仅做建议使用): .....	7
1.8 WIFI 连接方法 (该方法在 SDK 中无封装, 在此仅做建议使用): .....	8
1.9 COM-AP02 串口连接方法 (该方法在 SDK 中无封装, 在此仅做建议使用): .....	8
2、PrintDriver 类提供的方法 .....	8
2.1 HsBuletoothPrintDriver 类: .....	8
2.1.1 public static HsBluetoothPrintDrivergetInstance() .....	8
2.1.2 public synchronized void connect(BluetoothDevice device) .....	8
2.1.3 public synchronized void start() .....	8
2.1.4 public void BT_Write(String dataString) .....	8
2.1.5 public void BT_Write(byte[] out, int dataLen) .....	8
2.1.6 public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEPTYPE,byte WifiType) .....	8
2.1.7 public void setDhcp(boolean enable) .....	9
2.1.8 public void setStaticIp(String ip, String subMask, String gateWay) .....	9
2.1.9 状态查询( RP58A-MBU 适用) .....	9
public void setHandler(Handler handler) .....	9
2.2 HsUsbPrintDriver 类: .....	9
2.2.1 public static HsUsbPrintDrivergetInstance() .....	9
2.2.2 public void setUsbManager(UsbManagerusbManager) .....	9
2.2.3 public void USB_Write(String dataString) .....	9
2.2.4 public void USB_Write(byte[] out) .....	9
2.2.5 public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEPTYPE,byte WifiType) .....	9
2.2.6 public void setDhcp(boolean enable) .....	10
2.2.7 public void setStaticIp(String ip, String subMask, String gateWay) .....	10
2.3 HsWifiPrintDriver 类: .....	10
2.3.1 public static HsWifiPrintDrivergetInstance() .....	10

2.3.2	public BooleanWIFISocket(String ip, int port).....	10
2.3.3	public void WIFI_Write(String dataString).....	10
2.3.4	public void WIFI_Write(byte[] out).....	10
2.3.5	public void IsNoConnection(String ip) .....	10
2.3.6	public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEType,byte WifiType) .....	10
2.3.7	public void setDhcp(boolean enable) .....	11
2.3.8	public void setStaticIp(String ip, String subMask, String gateWay) .....	11
<b>2.4</b>	<b>HsComPrintDriver 类: .....</b>	<b>11</b>
2.4.1	public static HsComPrintDriver getInstance() .....	11
2.4.2	public void setHandler(Handler handler) .....	11
2.4.3	public void setCharsetName(String charsetName) .....	11
2.4.4	public void connect() .....	11
2.4.5	public void disconnect().....	11
2.4.6	public void WriteCmd(String dataString).....	11
2.4.7	public void WriteCmd(String dataString, boolean bGBK) .....	11
2.4.8	public void WriteCmd(byte[] out).....	12
2.4.9	public synchronized int getState().....	12
<b>3、</b>	<b>打印指令公共部分.....</b>	<b>12</b>
<b>3.1</b>	<b>打印基本指令.....</b>	<b>12</b>
3.1.1	public synchronized intgetState() .....	12
3.1.2	public synchronized void stop() .....	12
3.1.3	public Boolean IsNoConnection() .....	12
3.1.4	public void SetDefaultSetting().....	12
3.1.5	public void Begin().....	12
3.1.6	public void LF() .....	12
3.1.7	public void CR() .....	12
3.1.8	public void SelftestPrint().....	12
3.1.9	public void Beep(byte times, byte time) .....	12
3.1.10	public void StatusInquiry() .....	13
3.1.11	public void CLS() .....	13
3.1.12	public void endPro ().....	13
<b>3.2</b>	<b>打印位置相关方法 .....</b>	<b>13</b>
3.2.1	public void SetRightSpacing(byte Distance) .....	13
3.2.2	public void SetAbsolutePrintPosition(byte nL, byte nH).....	13
3.2.3	public void SetRelativePrintPosition(byte nL, byte nH) .....	13
3.2.4	public void SetDefaultLineSpacing().....	13
3.2.5	public void SetLineSpacing(byte LineSpacing) .....	13
3.2.6	public void SetLeftStartSpacing(byte nL, byte nH) .....	13
3.2.7	public void SetAreaWidth(byte nL, byte nH) .....	13
<b>3.3</b>	<b>打印字符设置相关方法.....</b>	<b>14</b>
3.3.1	public void SetCharacterPrintMode(byte CharacterPrintMode) .....	14
3.3.2	public void SetUnderline(byte UnderlineEn) .....	14
3.3.3	public void SetBold(byte BoldEn).....	14

3.3.4	public void SetCharacterFont(byte Font).....	14
3.3.5	public void SetRotate(byte RotateEn) .....	14
3.3.6	public void SetPrintRotate(byte Rotate).....	15
3.3.7	public void SetAlignMode(byte AlignMode).....	15
3.3.8	public void SetInvertPrint(byte InvertModeEn).....	15
3.3.9	public void SetFontEnlarge(byte FontEnlarge).....	15
3.3.10	public void SetBlackReversePrint(byte BlackReverseEn) .....	16
3.3.11	public void setCharsetName(String charsetName) .....	16
3.3.12	public void setHsCodepage(byte selCodepage) .....	16
3.3.13	public void setHsCodepageForCom(byte selCodepage) .....	16
<b>3.4</b>	<b>汉字控制命令相关方法.....</b>	<b>16</b>
3.4.1	public void SetChineseCharacterMode(byte ChineseCharacterMode).....	16
3.4.2	public void SelChineseCodepage() .....	17
3.4.3	public void CancelChineseCodepage() .....	17
3.4.4	public void SetChineseUnderline(byte ChineseUnderlineEn) .....	17
<b>3.5</b>	<b>钱箱控制命令方法 .....</b>	<b>17</b>
3.5.1	public void OpenDrawer(byte DrawerNumber, byte PulseStartTime, byte PulseEndTime).....	17
<b>3.6</b>	<b>切纸命令控制相关方法.....</b>	<b>18</b>
3.6.1	public void CutPaper() .....	18
3.6.2	public void PartialCutPaper() .....	18
3.6.3	public void FeedAndCutPaper(byte CutMode) .....	18
3.6.4	public void FeedAndCutPaper(byte CutMode, byte FeedDistance) .....	18
<b>3.7</b>	<b>特殊打印相关方法 .....</b>	<b>18</b>
3.7.1	public void SetHRIPosition(byte position) .....	18
3.7.2	public void SetCodeMarginLeft(byte nL,byte nH) .....	18
3.7.3	public void AddCodePrint(BarcodeTypeCodeType, String data) .....	19
3.7.4	private void CODE_QR_CODE(String data) .....	19
3.7.5	public void printImage(Bitmap bitmap, int type).....	19
	public boolean printImage(Bitmap bitmap, int type, PrintImageCallback callback) .....	19
	public boolean printImageSync(Bitmap bitmap, int type) .....	20
3.7.6	public boolean defineNVLogo(String[] logolist, int type).....	20
3.7.7	public void printNVLogo(byte nvLogoIndex, int type) .....	20
<b>3.8</b>	<b>状态查询（---非定制版无效） .....</b>	<b>20</b>
3.8.1	public byte StatusInquiryFinish() ---弃用 .....	20
3.8.2	public void StatusInquiryFinish(int handlerSign, Handler mHandler) ---弃用 .....	21
3.8.3	public byte StatusInquiry(byte type) ---弃用 .....	21
<b>4、</b>	<b>标签指令部分 .....</b>	<b>22</b>
<b>4.1</b>	<b>主要的类 .....</b>	<b>22</b>
<b>4.2</b>	<b>与热敏指令相同的标签指令 .....</b>	<b>22</b>
<b>4.3</b>	<b>标签系统设定指令 .....</b>	<b>23</b>
4.3.1	public void SetSize(String width, String length) .....	23
4.3.2	public void SetGAP (String width, String length).....	23
4.3.3	public void SetOFFSET(String length).....	23

4.3.4 public void SetSPEED(String speed) .....	23
4.3.5 public void SetDENSITY (String ensity) .....	23
4.3.6 public void SetDIRECTION (String dir) .....	23
4.3.7 public void SetREFERENCE(String posi_x, String posi_y) .....	23
4.3.8 public void SetSHIFT(String dots).....	23
4.3.9 public void SetCODEPAGE(String num) .....	23
4.3.10 public void SetFEED (String dots).....	24
4.3.11 public void SetBACKFEED (String dots).....	24
4.3.12 public void SetFORMFEED() .....	24
4.3.13 public void SetHOME().....	24
4.3.14 public void SetPRINT(String setM, String copyN) .....	24
4.3.15 public void SetSOUND(String level, String interval) .....	24
4.3.16 public void SetLIMITFEED (String n).....	25
<b>4.4 标签内容设定指令 .....</b>	<b>25</b>
4.4.1 public void SetBAR(String x, String y, String width, String height) .....	25
4.4.2 public void drawBox(String x_start, String y_start, String x_end, String y_end, String lineThickness) .....	25
4.4.3 public void drawBitMap(String x, String y, String width, String height, String mode, byte[] data).....	25
4.4.4 public void PutBmp(String x, String y, String fileName) .....	26
4.4.5 public void PutPcx(String x, String y, String fileName) .....	26
4.4.6 public void ClearData(String X_start, String Y_start, String X_width, String Y_height).....	26
4.4.7 public void SetReverse(String X_start, String Y_start, String X_width, String Y_height).....	26
4.4.8 public void PrintText(String X, String Y, String font, String rotation, String x_multi, String y_multi, String content).....	26
4.4.9 public void loadProFile(String FileName) { .....	27
4.4.10 public void loadDateFile(String FileName, String dataSize, String data) { .....	27
4.4.11 public void setCounter(String num, String step) { .....	28
4.4.12 public void CodePrint(String X, String Y, String codeType, String height, String readable, String rotation, String narrow, String wide, String code) .....	28
<b>5、pdf 打印 .....</b>	<b>28</b>
5.1 操作步聚 .....	28
<b>附页 1 Codepage 对应列表.....</b>	<b>29</b>

# 1、PrintDriver 与 APP 通讯方法

## 1.1 PrintDriver 类

- |                           |          |
|---------------------------|----------|
| 1) HsBluetoothPrintDriver | 蓝牙       |
| 2) HsUsbPrintDriver       | USB      |
| 3) HsWifiPrintDriver      | WIFI     |
| 4) HsComPrintDriver       | COM AP02 |

## 1.2 创建一个 Handler 子类 ConnStateHandler，处理 driver 发

送过来的消息如下：

```
private class ConnStateHandler extends Handler {
    @Override
    public void handleMessage(Message msg) {
super.handleMessage(msg);
        Bundle data = msg.getData();
        switch (data.getInt("flag")) {
            //连接状态发生变化时接收到消息
            case Contants.FLAG_STATE_CHANGE:
                //表示当前的连接状态
                //有 5 种连接状态（在 Contants 类中可以找到）：
                UNCONNECTED、
                CONNECTED_BY_BLUETOOTH、
                CONNECTED_BY_USB、
                CONNECTED_BY_WIFI、
                CONNECTED_BY_COM //AP02-串口连接
        int state = data.getInt("state");
            //在这里可以写入你的处理代码

            break;

            //连接失败时接收到消息
            case Contants.FLAG_FAIL_CONNECT:
                //在这里可以写入你的处理代码

            break;

            //连接成功时接收到消息
            case Contants.FLAG_SUCCESS_CONNECT:
                //在这里可以写入你的处理代码

            break;
```

```
    }
}
}
```

### 1.3 通过以下方法设置 Handler:

```
ConnStateHandler connStateHandler = new ConnStateHandler();
HsBluetoothPrintDriver.getInstance().setHandler(connStateHandler);
HsUsbPrintDriver.getInstance().setHandler(connStateHandler);
HsWifiPrintDriver.getInstance().setHandler(connStateHandler);
HsComPrintDriver.getInstance().setHandler(connStateHandler); //AP02
```

### 1.4 public void setHandler(Handler handler)

说明：程序刚启动时，用于设置 Handler

### 1.5 Contants 连接状态类的状态标志:

包括：UNCONNECTER  
 CONNECTED\_BY\_BLUETOOTH  
 CONNECTED\_BY\_USB  
 CONNECTED\_BY\_WIFI  
 CONNECTED\_BY\_COM  
 FLAG\_STATE\_CHANGE  
 FALG\_FAIL\_CONNECT  
 FLAG\_SUCCESS\_CONNECT  
 FLAG\_MSG\_READ

### 1.6 蓝牙连接方法(该方法在 SDK 中无封装, 在此仅做建议使用):

```
private void connectBluetooth(BluetoothDevice bluetoothDevice) {
    HsBluetoothPrintDriver hsBluetoothPrintDriver = HsBluetoothPrintDriver.getInstance();
    hsBluetoothPrintDriver.start();
    hsBluetoothPrintDriver.connect(bluetoothDevice);
}
```

### 1.7 USB 连接方法(该方法在 SDK 中无封装, 在此仅做建议使用):

说明：通过调用 `UsbManager.getDeviceList()` 获取一个包含所有已挂载的 USB 设备的 `HashMap`。通常，这个 `HashMap` 会包含两个 USB 设备，第一个是安卓手机自身，第二个才是打印机，需要选取第二个迭代到的 `UsbDevice` 进行连接。

```
private void connectUsb(UsbDevice usbDevice){
    HsUsbPrintDriver hsUsbPrintDriver = HsUsbPrintDriver.getInstance();
    hsUsbPrintDriver.connect(usbDevice);
}
```

## 1.8 WIFI 连接方法(该方法在 SDK 中无封装,在此仅做建议使用):

```
private void connectWifi(String ip,int port) {  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
HsWifiPrintDriverhsWifiPrintDriver = HsWifiPrintDriver.getInstance();  
hsWifiPrintDriver.WIFISocket(ip,port);  
        }  
    }).start();  
}
```

## 1.9 COM-AP02 串口连接方法 (该方法在 SDK 中无封装,在此仅做建议使用):

```
private void connectCOM(){//串口连接 for AP02  
    HsComPrintDriver.getInstance().connect();  
}
```

# 2、PrintDriver 类提供的方法

## 2.1 HsBuletoothPrintDriver 类:

### 2.1.1 public static HsBluetoothPrintDrivergetInstance()

说明: HsBuletoothPrintDriver 类实例化一个蓝牙对象

### 2.1.2 public synchronized void connect(BluetoothDevice device)

说明: 开启连接线程, 蓝牙初始化连接设备

参数: device:便携式蓝牙打印机的 MAC 地址

### 2.1.3 public synchronized void start()

说明: 开始监听打印设备蓝牙

### 2.1.4 public void BT\_Write(String dataString)

说明: 打印字符串数据方法

### 2.1.5 public void BT\_Write(byte[] out, int dataLen)

说明: 发送命令方法

### 2.1.6public void setWifiParam(String ssid,String password,byte type,byte

WPAType,byte WPAEncryType,byte WEPTType,byte WifiType)

说明: 设置 WIFI 信息

ssid: wifi 名



password: wifi 密码  
type: wifi 类型 (开放/WPA/WEP)  
WPAType: WPA 类型 (WPA-PSK/WPA2-PSK)  
WPAEncryType: WPA 加密方式 (AES/TPIK)  
WEPTYPE: WEP 类型 (OPEN/SHARE)  
WifiType: wifi 模式 (STA/AP)

### 2.1.7 public void setDhcp(boolean enable)

说明: 设置 DHCP 开关

Enable: true 是开, false 是关

### 2.1.8 public void setStaticIp(String ip, String subMask, String gateWay)

说明: 设置 ip

### 2.1.9 状态查询( RP58A-MBU 适用)

#### public void setHandler(Handler handler)

说明: 程序刚启动时调用, 用于设置 Handler, 可接收回传信息, 回传信息类型查看 Contants 中, 如下:

```
int FLAG_BLUETOOTH_RECEIVE_DATA = 0x98;  
// 回传的值需要 resultVal&0xff, 才能得到 success:0x80    fail:0x00  
具体实现可参考 RTExample 代码
```

## 2.2 HsUsbPrintDriver 类:

### 2.2.1 public static HsUsbPrintDrivergetInstance()

说明: HsUsbPrintDriver 类实例化一个 USB 对象

### 2.2.2 public void setUsbManager(UsbManagerusbManager)

说明: 程序刚启动时, 用于设置 UsbManager

参数: usbManager USB 管理器

### 2.2.3 public void USB\_Write(String dataString)

说明: 打印字符串数据方法

### 2.2.4 public void USB\_Write(byte[] out)

说明: 发送命令方法

### 2.2.5 public void setWifiParam(String ssid,String password,byte type,byte

WPAType,byte WPAEncryType,byte WEPTYPE,byte WifiType)

说明: 设置 WIFI 信息

ssid: wifi 名  
password: wifi 密码  
type: wifi 类型 (开放/WPA/WEP)  
WPAType: WPA 类型 (WPA-PSK/WPA2-PSK)  
WPAEncryType: WPA 加密方式 (AES/TPIK)  
WEPTYPE: WEP 类型 (OPEN/SHARE)  
WifiType: wifi 模式 (STA/AP)

### 2.2.6 public void setDhcp(boolean enable)

说明: 设置 DHCP 开关  
Enable: true 是开, false 是关

### 2.2.7 public void setStaticIp(String ip, String subMask, String gateWay)

说明: 设置 ip

## 2.3 HsWifiPrintDriver 类:

### 2.3.1 public static HsWifiPrintDrivergetInstance()

说明: HsWifiPrintDriver 类实例化一个 WIFI 对象

### 2.3.2 public BooleanWIFISocket(String ip, int port)

说明: WIFI 通讯管道设置  
参数: ip IP 地址  
port 设备端口号

### 2.3.3 public void WIFI\_Write(String dataString)

说明: 打印字符串数据方法

### 2.3.4 public void WIFI\_Write(byte[] out)

说明: 发送命令方法

### 2.3.5 public voidIsNoConnection(String ip)

说明: 判断当前 WIFI 是否连接  
参数: ip IP 地址

### 2.3.6 public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEPTYPE,byte WifiType)

说明: 设置 WIFI 信息  
ssid: wifi 名  
password: wifi 密码  
type: wifi 类型 (开放/WPA/WEP)

WPAType: WPA 类型 (WPA-PSK/WPA2-PSK)

WPAEncryType: WPA 加密方式 (AES/TPIK)

WEType: WEP 类型 (OPEN/SHARE)

WifiType: wifi 模式 (STA/AP)

### 2.3.7 public void setDhcp(boolean enable)

说明: 设置 DHCP 开关

Enable: true 是开, false 是关

### 2.3.8 public void setStaticIp(String ip, String subMask, String gateWay)

说明: 设置 ip

## 2.4 HsComPrintDriver 类:

### 2.4.1 public static HsComPrintDriver getInstance()

说明: HsComPrintDriver 类实例化一个对象

### 2.4.2 public void setHandler(Handler handler)

说明: 程序刚启动时调用, 用于设置 Handler, 可接收回传信息, 回传信息类型查看 Contents 中, 如下:

```
int CONNECTED_BY_COM = 0x14;  
int FLAG_STATE_CHANGE = 0x20;  
int FLAG_FAIL_CONNECT = 0x21;  
int FLAG_SUCCESS_CONNECT = 0x22;  
int FLAG_COM_RECEIVE_DATA = 0x99;
```

### 2.4.3 public void setCharsetName(String charsetName)

说明: 设置字符编码, 可设置为 "UTF-8"、"GBK"等

### 2.4.4 public void connect()

说明: 执行连接动作

### 2.4.5 public void disconnect()

说明: 断开连接

### 2.4.6 public void WriteCmd(String dataString)

说明: 可直接打印文字, 写指令到打印机, 默认字符编码

参数: 字符串

### 2.4.7 public void WriteCmd(String dataString, boolean bGBK)

说明: 可直接打印文字, 写指令到打印机, 指定字符编码

参数:

dataString 字符串  
bGBK 编码格式

#### 2.4.8 public void WriteCmd(byte[] out)

说明：写指令到打印机

参数：

out byte[]形式

#### 2.4.9 public synchronized int getState()

说明：获取连接状态

参考：Contants 类， 可参考 2.4.2 中描述

## 3、打印指令公共部分

### 3.1 打印基本指令

#### 3.1.1 public synchronized int getState()

说明：返回当前的连接状态

#### 3.1.2 public synchronized void stop()

说明：断开与打印机的连接

#### 3.1.3 public Boolean IsNoConnection()

说明：判断终端设备与打印机设备连接状态, True:没有连接, False:连接成功

#### 3.1.4 public void SetDefaultSetting()

说明：选择默认设置模式

#### 3.1.5 public void Begin()

说明：初始化打印机，打印机复位，清空缓存

#### 3.1.6 public void LF()

说明：在输入的数据中加入一个打印并换行的指令

#### 3.1.7 public void CR()

说明：打印并回车指令。允许自动进纸时，这条命令与 LF 命令的功能相同；  
不允许自动进纸时，这条命令将被忽略。

#### 3.1.8 public void SelftestPrint()

说明：打印自测页指令

#### 3.1.9 public void Beep(byte times, byte time)

说明：蜂鸣器提示音指令

参数：times 蜂鸣次数

time 单次蜂鸣时间

### 3.1.10 public void StatusInquiry()

说明：状态查询指令

### 3.1.11 public void CLS()

说明：清除缓冲区

### 3.1.12 public void endPro ()

说明：打印结束指令

## 3.2 打印位置相关方法

### 3.2.1 public void SetRightSpacing(byte Distance)

说明：设置字符的右间距指令。每点的距离与打印头分辨率相关

参数：Distance 右间距距离

范围：0 ≤ Distance ≤ 255

### 3.2.2 public void SetAbsolutePrintPosition(byte nL, byte nH)

说明：设置绝对打印位置指令。设定从一行的开始到将要打印字符的位置之间的距离。从一行的开始到打印位置的距离为 $[(nL + nH * 256) * 0.125 \text{ mm}]$ 。

参数：0 ≤ nL ≤ 255 , 0 ≤ nH ≤ 255

### 3.2.3 public void SetRelativePrintPosition(byte nL, byte nH)

说明：设置相对打印位置指令。

参数：0 ≤ nL ≤ 255 , 0 ≤ nH ≤ 255

### 3.2.4 public void SetDefaultLineSpacing()

说明：设置缺省行间距指令。缺省值为 3.75mm (30\*0.125mm)。

### 3.2.5 public void SetLineSpacing(byte LineSpacing)

说明：设置行间距指令。

参数：LineSpacing 行间距=LineSpacing\*0.125mm

范围：0 ≤ LineSpacing ≤ 255

### 3.2.6 public void SetLeftStartSpacing(byte nL, byte nH)

说明：设置左侧空白量指令。左边空白量设置为  $[(nL + nH * 256) * 0.125 \text{ 毫米}]$ 。  
在标准模式下，该命令仅在一行的起始位置处理时有效。

参数：0 ≤ nL ≤ 255 , 0 ≤ nH ≤ 255

### 3.2.7 public void SetAreaWidth(byte nL, byte nH)

说明：设置打印区域宽度

参数：0 ≤ nL ≤ 255 , 0 ≤ nH ≤ 255

### 3.3 打印字符设置相关方法

#### 3.3.1 public void SetCharacterPrintMode(byte CharacterPrintMode)

说明：设置字符打印模式指令。通过指定参数 CharacterPrintMode 的值选择打印模式。参数 CharacterPrintMode 的定义如下：

位	关/开	十六进制码	十进制码	功能
0	关	00	0	字符字型A (12×24)。
	开	01	1	字符字型B (9×17)。
1	-	-	-	未定义。
2	-	-	-	未定义。
3	关	00	0	解除粗体模式。
	开	08	8	设置粗体模式。
4	关	00	0	解除倍高模式。
	开	10	16	设置倍高模式。
5	关	00	0	解除倍宽模式。
	开	20	32	设置倍宽模式。
6	-	-	-	未定义。
7	关	00	0	解除下划线模式。
	开	80	128	设置下划线模式。

参数：0 ≤ CharacterPrintMode ≤ 255

#### 3.3.2 public void SetUnderline(byte UnderlineEn)

说明：设定/解除下划线指令。0x00 或 0x30 - 解除下划线模式  
0x01 或 0x31 - 设定下划线模式（1 点粗）  
0x02 或 0x32 - 设定下划线模式（2 点粗）

参数：UnderlineEn

#### 3.3.3 public void SetBold(byte BoldEn)

说明：设定/解除粗体打印指令。当 BoldEn 的最低有效位为 0 时，解除粗体打印模式当 BoldEn 的最低有效位为 1 时，设定粗体打印模式

参数：0 ≤ BoldEn ≤ 255

#### 3.3.4 public void SetCharacterFont(byte Font)

说明：选择字型指令。0x00 或 0x30 - 选择字型 A（12\*24）  
0x01 或 0x31 - 选择字型 B（9\*17）

参数：Font

#### 3.3.5 public void SetRotate(byte RotateEn)

说明：设置/解除顺时针旋转 90° 指令。0x00 或 0x30 -解除顺时针 90°旋转模式  
0x01 或 0x31-设置顺时针 90°旋转模式

参数：RotateEn

### 3.3.6 public void SetPrintRotate(byte Rotate)

说明：设置左旋、右旋、正常模式打印

参数：RotateEn: 2 设置 左旋，1 设置右旋，0 设置正常

### 3.3.7 public void SetAlignMode(byte AlignMode)

说明：设置对齐方式指令。0x00 或 0x30 - 左对齐

0x01 或 0x31 - 居中

0x02 或 0x32 - 右对齐

参数：AlignMode

### 3.3.8 public void SetInvertPrint(byte InvertModeEn)

说明：设置/解除颠倒打印模式指令。

当 InvertMode 的最低有效位为 0 时，关闭颠倒打印模式

当 InvertMode 的最低有效位为 1 时，打开颠倒打印模式

参数：0 ≤ InvertModeEn ≤ 255

### 3.3.9 public void SetFontEnlarge(byte FontEnlarge)

说明：设置字符大小指令。用 0 到 3 位设定字符高度 4 到 7 位设定字符宽度如下所示。

位	关/开	十六进制	十进制	功能
0		字符高度设定。见表2。		
1				
2				
3				
4		字符宽度设定。见表1。		
5				
6				
7				

表 1  
字符宽度设定

十六进制	十进制	宽度
00	0	1(普通)
10	16	2(倍宽)
20	32	3
30	48	4
40	64	5
50	80	6
60	96	7
70	112	8

表 2  
字符高度设定

十六进制	十进制	宽度
00	0	1(普通)
01	1	2(倍高)
02	2	3
03	3	4
04	4	5
05	5	6
06	6	7
07	7	8

参数：0 ≤ FontEnlarge ≤ 255

### 3.3.10 public void SetBlackReversePrint(byte BlackReverseEn)

说明：设置/解除反白打印模式。

当 BlackReverseEn 的最低有效位为 0 时，反白模式关闭

当 BlackReverseEn 的最低有效位为 1 时，反白模式打开

参数：0 <= BlackReverseEn <= 255

### 3.3.11 public void setCharsetName(String charsetName)

说明：设置 charsetName，不设置的话，默认为 GBK 编码。

参数：charsetName 打印中英文设置为“GBK”，其它语言设置为“UTF-8”

### 3.3.12 public void setHsCodepage(byte selCodepage)

说明：设置打印机的 codepage，该方法适用于便携 pos，如 RPP200,02N 等，仅支持 Esc

参数：selCodepage， byte 类型，

接受的值：请参考：[附页 1 codepage 对应列表](#)

### 3.3.13 public void setHsCodepageForCom(byte selCodepage)

说明：设置打印机的 codepage，该方法适用于 AP02 打印机，仅支持 Esc

参数：selCodepage， byte 类型

接受的值：请参考：[附页 1 codepage 对应列表](#)

## 3.4 汉字控制命令相关方法

### 3.4.1 public void SetChineseCharacterMode(byte ChineseCharacterMode)

说明：设置汉字字符打印模式组合指令。参数设置如下：



位	关/开	十六进制	十进制	ASB 状态
0	—	—	—	未定义。
1	—	—	—	未定义。
2	关	00	0	禁止倍宽模式。
	开	04	4	允许倍宽模式。
3	关	00	0	禁止倍高模式。
	开	08	8	允许倍高模式。
4	—	—	—	未定义。
5	—	—	—	未定义。
6	—	—	—	未定义。
7	关	00	0	禁止下划线模式。
	开	80	128	允许下划线模式。

参数：0 <= ChineseCharacterMode <= 255

### 3.4.2 public void SelChineseCodepage()

说明：选择汉字模式指令。在打印数据中加入一个选择汉字模式的命令（codepage 设置为 255）。

### 3.4.3 public void CancelChineseCodepage()

说明：取消汉字模式指令。在打印数据中加入一个取消汉字模式的命令（codepage 设置为 0）。

### 3.4.4 public void SetChineseUnderline(byte ChineseUnderlineEn)

说明：设置/取消汉字字符下划线模式指令。0x00 或 0x30 - 解除下划线模式

0x01 或 0x31 - 设定下划线模式（1 点粗）

0x02 或 0x32 - 设定下划线模式（2 点粗）

参数：ChineseUnderlineEn

## 3.5 钱箱控制命令方法

### 3.5.1 public void OpenDrawer(byte DrawerNumber, byte PulseStartTime, byte PulseEndTime)

说明：开钱箱指令。0x00 或 0x30 - 脉冲发送到钱箱输出引脚 2（钱箱 1）

0x01 或 0x31 - 脉冲发送到钱箱输出引脚 5（钱箱 2）

参数：DrawerNumber 钱箱号 1 或者 2

PulseStartTime 开机时间= PulseStartTime x 2 millisecond

PulseEndTime 关机时间= PulseEndTime x 2 millisecond

范围：0 <= PulseStartTime <= 2550 <= PulseEndTime <= 255

## 3.6 切纸命令控制相关方法

### 3.6.1 public void CutPaper()

说明：全切指令。

### 3.6.2 public void PartialCutPaper()

说明：部分切纸指令

### 3.6.3 public void FeedAndCutPaper(byte CutMode)

说明：选择切纸模式并切纸指令。

参数：CutMode 可配置值为 1 或者 49，打印模式只有部分切纸，无全切纸

### 3.6.4 public void FeedAndCutPaper(byte CutMode, byte FeedDistance)

说明：选择切纸模式并切纸指令。

参数：CutMode 可配置值为 66，打印模式只有部分切纸，无全切纸

FeedDistance = 0 时，打印机进纸到切纸位置并切纸；

FeedDistance ≠ 0 时，打印机进纸到(切纸位置 + [Feed\*0.125 毫米 {0.0049 英寸}])

并切纸。

## 3.7 特殊打印相关方法

### 3.7.1 public void SetHRIPosition(byte position)

说明：选择 HRI 字符的打印位置

参数：position 选择打印位置

打印条形码时选择HRI字符的打印位置

n 选择打印位置如下图所示：

n	打印位置
0, 48	不打印
1, 49	在条形码上方
2, 50	在条形码下方
3, 51	在条形码的上方及下方

• HRI 表示可阅读的条形码对应字符。

### 3.7.2 public void SetCodeMarginLeft(byte nL,byte nH)

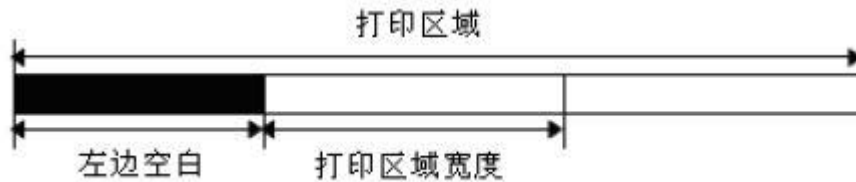
说明：设定左侧空白量

参数：nL

nH

用nL 和 nH设定左边空白量。

- 左边空白量设置为  $[(nL + nH \times 256) \times 0.125 \text{ 毫米}]$ 。



### 3.7.3 public void AddCodePrint(BarcodeTypeCodeType, String data)

说明：一维条码打印指令。

参数：CodeType 十种一维条码类型

data 条码数据

条码类型枚举：

```
public enum BarcodeType{
    UPC_A,
    UPC_E,
    EAN13,
    EAN8,
    CODE39,
    ITF,
    CODABAR,
    CODE93,
    CODE128,
    QR_CODE
}
```

### 3.7.4 private void CODE\_QR\_CODE(String data)

说明：二维码打印指令。

参数： data 二维码数据

### 3.7.5 public void printImage(Bitmap bitmap, int type)

```
public boolean printImage(Bitmap bitmap, int type, PrintImageCallback
```

callback)

说明：打印图片指令

参数： bitmap 图片数据

type TYPE\_58: 对应 58 机型所使用热敏纸模式 (384dots)

TYPE\_80: 对应 80 机型所使用热敏纸模式(576dots)

//TYPE\_300: 对应 RPP300 机型所使用热敏纸模式(640dots)

callback 打印图片结束,回调,以防打印乱序,可 new PrintImageCallback(),  
在实现方法中,进行打印图片后的操作

建议使用以下方法打印图片：

### **public boolean printImageSync(Bitmap bitmap, int type)**

说明：打印图片指令（同步）

参数： bitmap            图片数据  
        type             TYPE\_58: 对应 58 机型所使用热敏纸模式（384dots）  
                          TYPE\_80: 对应 80 机型所使用热敏纸模式(576dots)

备注：打印图片为耗时操作，建议在子线程中调用该方法；

### **3.7.6 public boolean defineNVLogo(String[] logolist, int type)**

说明：下载定义 NV Logo 到打印机

参数： **logolist** 图片的完整路径数组，支持定义多张位图。如  
**/storage/emulated/0/DCIM/Camera/1.bmp"**

type    TYPE\_58: 对应 58 机型所使用热敏纸模式（384dots）  
          TYPE\_80: 对应 80 机型所使用热敏纸模式(576dots)  
          TYPE\_300: 对应 RPP300 机型所使用热敏纸模式(640dots)

### **3.7.7 public void printNVLogo(byte nvLogoIndex, int type)**

说明：打印之前下载的 NV Logo。

参数： nvLogoIndex 图片 Logo 的序号，如 1，2，3 等

type    TYPE\_58: 对应 58 机型所使用热敏纸模式（384dots）  
          TYPE\_80: 对应 80 机型所使用热敏纸模式(576dots)  
          TYPE\_300: 对应 RPP300 机型所使用热敏纸模式(640dots)

## **3.8 状态查询（---非定制版无效）**

### **3.8.1 public byte StatusInquiryFinish() ---弃用**

---非定制版无效

说明：打印后调用该接口，查询打印是否完成。蓝牙 WIFI 接口才支持。

参数：无

返回值：如果返回 0x80 表示打印完成

### 3.8.2 public void StatusInquiryFinish(int handlerSign, Handler mHandler) ---弃用

---非定制版无效

说明：打印后调用该接口，查询打印是否完成。蓝牙 WIFI 接口才支持。

参数：handlerSign：自定义的 handler 标记

mhandler：传入的 handler 对象

返回：

Eg: Bundle data = msg.getData();

data.getInt("flag")为 handlerSign

int inquiry\_status = data.getInt("state", 0) & 0xFF;

inquiry\_status 为状态值，0x80 表示打印完成

### 3.8.3 public byte StatusInquiry(byte type) ---弃用

---非定制版无效

说明：打印机状态查询。

参数：type 值可以为 1 2 3 4

返回值：打印机相关状态值,详见下表：

type = 1: 打印机状态

位	关/开	十六进制码	十进制码	功能
0	关	00	0	未使用, 选定为关。
1	开	02	2	未使用, 选定为开。
2	关	00	0	钱箱打开/关闭信号为低（连接引脚 3）。
	开	04	4	钱箱打开/关闭信号为高（连接引脚 3）。
3	--	--	--	未定义。
4	开	10	16	未使用, 选定为开。
5.6	--	--	--	未定义。
7	关	00	0	未使用, 选定为关。

type = 2: 脱机状态

位	关/开	十六进制码	十进制码	功能
0	关	00	0	未使用, 选定为关。
1	开	02	2	未使用, 选定为开。
2	关	00	0	盖板已关闭。
	开	04	4	盖板已打开。
3	关	00	0	不通过进纸键进纸。
	开	08	8	通过进纸键进纸。
4	开	10	16	未使用选定为开。
5	--	--	--	未定义。
6	关	00	0	没有错误。
	开	40	64	发生错误。

7	关	00	0	未使用, 选定为关。
---	---	----	---	------------

type = 3: 错误状态

位	关/开	十六进制码	十进制码	功能
0	关	00	0	未使用, 选定为关。
1	开	02	2	未使用, 选定为开。
2	--	--	--	未定义。
3	关	00	0	没有自动切纸错误。
	开	08	8	发生自动切纸错误。
4	开	10	16	未使用, 选定为开。
5	关	00	0	没有不可恢复的错误。
	开	20	32	出现不可恢复的错误。
6	关	00	0	没有可自动恢复的错误。
	开	40	64	出现可自动恢复的错误。
7	关	00	0	未使用, 选定为关。

位 6: 如果打印期间打开卷纸盖或打印头的温度过高, 位6将被置为开, 直到打印头温度有效地降下来。

type = 4: 连续用纸传感器状态

位	关/开	十六进制码	十进制码	功能
0	关	00	0	未使用, 选定为关。
1	开	02	2	未使用, 选定为开。
2. 3	--	--	--	未定义。
4	开	10	16	未使用, 选定为开。
5. 6	关	00	0	纸尽传感器, 有纸。
	开	60	96	纸尽传感器, 检测到卷纸末端。
7	关	00	0	未使用, 选定为关。

用法如下:

当 type=2 时, 判断是否通过进纸键进纸

```
Byte b=StatusInquiry(2);
```

```
if ((b & 8) == 8){ 通过进纸键进纸 }
```

## 4、标签指令部分

### 4.1 主要的类

- 1) LabelBluetoothPrintDriver 蓝牙
- 2) LabelUsbPrintDriver USB
- 3) LabelWifiPrintDriver WIFI

### 4.2 与热敏指令相同的标签指令

Begin(), SetCLS(), stop(), WIFISocket(), WIFI\_Write(), BT\_Write(), SetPrintRotate(),  
SetCLS(), LF(), CR(), Beep(), StatusInquiry(), SelftestPrint(), IsNoConnection(), endPro()

## 4.3 标签系统设定指令

### 4.3.1 public void SetSize(String width, String length)

说明：设定卷标纸的宽度和长度

参数：width:宽 length 长度

### 4.3.2 public void SetGAP (String width, String length)

说明：设置垂直间距

参数：high：两标签纸中间的垂直距离 length 标签纸张长度

### 4.3.3 public void SetOFFSET(String length)

说明：纸张停止的距离

参数：纸张停止的距离（公厘）

### 4.3.4 public void SetSPEED(String speed)

说明：控制打印速度

参数：speed = 2/3/4/5

### 4.3.5 public void SetDENSITY (String ensity)

说明：控制打印的浓度

参数：ensity = 0~15

### 4.3.6 public void SetDIRECTION (String dir)

说明：控制打印出纸盒打印字体的方向

参数：dir = 0/1

### 4.3.7 public void SetREFERENCE(String posi\_x, String posi\_y)

说明：定义卷标的参考坐标原点

参数：posi\_x：水平方向的坐标位置，以点(dot)为单位

posi\_y：垂直方向的坐标位置，以点(dot)为单位

### 4.3.8 public void SetSHIFT(String dots)

说明：打印偏移量

参数：dots：点数

### 4.3.9 public void SetCODEPAGE(String num)

说明：选择对应的国际字符集

参 数

num

说 明

字符集的名字或代表号，可进一步区分为 7-bit 或 8-bit

7-bit code page 字符集名称

USA: USA  
 BRI: British  
 GER: German  
 FRE: French  
 DAN: Danish  
 ITA: Italian  
 SPA: Spanish  
 SWE: Swedish  
 SWI: Swiss

#### 8-bit code page 字符集代表号

437: United States  
 850: Multilingual  
 852: Slavic  
 860: Portuguese  
 863: Canadian/French  
 865: Nordic

#### Windows code page

1250: Central Europe  
 1252: Latin I  
 1253: Greek  
 1254: Turkish

#### 4.3.10 public void SetFEED (String dots)

说明: 控制进纸的距离

参数: dots :进纸点数 dot = 9999 1mm = 8dots

#### 4.3.11 public void SetBACKFEED (String dots)

说明: 控制退纸的距离

参数: dots :进纸点数  $1 \leq \text{dot} \leq 9999$

#### 4.3.12 public void SetFORMFEED()

说明: 打印进 1 张纸

#### 4.3.13 public void SetHOME()

说明: 在使用含有间隙或黑标的标签纸时,若不能确定第一张标签纸是否在正确打印位置时,此指令可将标签纸向前推送至下一张标签纸的起点开始打印

#### 4.3.14 public void SetPRINT(String setM, String copyN)

说明: 打印缓冲区的数据

参数: setM: 指定打印的份数  $1 \leq \text{setM} \leq 999999999$

copyN: 指定每份打印的张数  $1 \leq \text{copyN} \leq 999999999$

#### 4.3.15 public void SetSOUND(String level, String interval)

说明: 控制蜂鸣器频率



参数: level: 音阶      范围 0~9  
Interval: 时间长度    范围 1~4095

#### 4.3.16 public void SetLIMITFEED (String n)

说明: 该指令用于设定打印机进纸时, 若经过所设定的长度仍无法侦测到垂直间距, 则打印机发生错误, 停止进纸

参数: n: 可使用公厘

### 4.4 标签内容设定指令

#### 4.4.1 public void SetBAR(String x, String y, String width, String height)

说明: 在标签上画线

参 数	说 明
x	左上角水平方向起始点, 以点 (dot) 表示
y	左上角垂直方向起始点, 以点 (dot) 表示
width	线宽, 以点 (dot) 表示
height	线高, 以点 (dot) 表示

#### 4.4.2 public void drawBox(String x\_start, String y\_start, String x\_end, String y\_end, String lineThickness)

说明: 用于在卷标上绘制矩形

参 数	说 明
X_start	矩形水平方向左上角起始位置以点 (dot) 表示
Y_start	矩形垂直方向左上角起始位置以点 (dot) 表示
X_end	矩形水平方向右下角起始位置以点 (dot) 表示
Y_end	矩形垂直方向右下角起始位置以点 (dot) 表示
line thickness	矩形框线厚度, 以点 (dot) 表示

#### 4.4.3 public void drawBitMap(String x, String y, String width, String height, String mode, byte[] data)

说明: 绘制位图

参 数	说 明
x	点阵影像的水平起始位置
y	点阵影像的垂直起始位置
width	影像的宽度, 以 byte 表示
height	影像的高度, 以点 (dot) 表示
mode	影像绘制模式
0	OVERWRITE
1	OR
2	XOR
data	影像数据

#### 4.4.4 public void PutBmp(String x, String y, String fileName)

说明：打印单色 BMP 格式图档

参 数 说 明

X: BMP 图形左上角 X 坐标

Y: BMP 图形左上角 Y 坐标

Filename: 已载至打印机的 BMP 图档

注意事项：建议使用仅有黑白二种颜色的 BMP 图档

#### 4.4.5 public void PutPcx(String x, String y, String fileName)

说明：打印单色 PCX 格式图档

参 数 说 明

X: BMP 图形左上角 X 坐标

Y: BMP 图形左上角 Y 坐标

Filename: 已载至打印机的 BMP 图档

#### 4.4.6 public void ClearData(String X\_start, String Y\_start, String X\_width, String Y\_height)

说明：清除影像缓冲区中指定大小的区域

<u>参 数</u>	<u>说 明</u>
X_start	欲清除区域的左上角 X 坐标，单位：dot
Y_start	欲清除区域的左上角 Y 坐标，单位：dot
X_width	欲清除区域的宽度，单位：dot
Y_height	欲清除区域的高度，单位：dot

#### 4.4.7 public void SetReverse(String X\_start, String Y\_start, String X\_width, String Y\_height)

说明：将图像缓存中指定的区域做反相的动作

<u>参 数</u>	<u>说 明</u>
X_start	欲清除区域的左上角 X 坐标，单位：dot
Y_start	欲清除区域的左上角 Y 坐标，单位：dot
X_width	欲清除区域的宽度，单位：dot
Y_height	欲清除区域的高度，单位：dot

注意事项： 200 DPI: 1 mm = 8 dots

300 DPI: 1 mm = 12 dots

#### 4.4.8 public void PrintText(String X, String Y, String font, String rotation, String x\_multi, String y\_multi, String content)

说明：打印文字

<u>参 数</u>	<u>说 明</u>
------------	------------

X	文本框左上角 X 坐标
Y	文本框左上角 Y 坐标
<b>font</b>	<b>字型名称</b>
1	8 x 12 英数字体
2	12 x 20 英数字体
3	16 x 24 英数字体
4	24 x 32 英数字体
5	32 x 48 英数字体
6	14 x 19 英数字体 OCR-B
7	14 x 25 英数字体 OCR-A
8	21 x 27 英数字体 OCR-B
TST24.BF2	繁体中文 24x24 字体 (大五码)
TSS24.BF2	简体中文 24x24 字体 (GB 码)
K	韩文 24x24 字体 (KS 码)
<b>Rotation</b>	<b>顺时针旋转角度</b>
0	不旋转
90	顺时针旋转 90 度
180	顺时针旋转 180 度
270	顺时针旋转 270 度
X-multi:	水平放大值, 最大可放大至 10 倍 有效系数: 1~10
Y-multi:	垂直放大值, 最大可放大至 10 倍 有效系数: 1~10

注意事项: 若文字内容包括双引号 ( “ ) 时, 则请用 \[ “ ] 取代  
若要打印 0D(hex) 字符时, 请在程序中使用 \[R] 来打印 CR  
若要打印 0A(hex) 字符时, 请在程序中使用 \[A] 来打印 LF  
五号字英文字母仅可打印大写字母

#### 4.4.9 public void loadProFile(String FileName) {

说明: 下载程序文件

参数: FileName: 储存于打印机中的文件名称

注意事项:

- (1) 大小写将表示不同的文件名称
- (2) 程序文件的扩展名必须为 “.BAS ”

#### 4.4.10 public void loadDateFile(String FileName, String dataSize, String data) {

说明: 下载资料文件

参数: FileName: 储存于打印机中的文件名称

DATA SIZE 不含文件头的实际文件大小, 以字节数计算

注意事项: 行与行之间的资料以 CR(0x0D) 及 LF(0x0A) 做分隔

#### 4.4.11 public void setCounter(String num, String step) {

说明：设定计数器及增量

参 数：

num: 计数器的初始值。总计有 50 (@0~@49) 个计数器可同时使用

Step: 计数器的增量值可为正数亦可为负数。-999999999<= step <=999999999

如果计数器只是被当作固定的变量使用时，请将增量值设为 0

#### 4.4.12 public void CodePrint(String X, String Y, String codeType, String height, String readable, String rotation, String narrow, String wide, String code)

说明：打印一维条码，下方为所支持的条码列表。

参 数	说 明
X	指定条码左上角 X 坐标
Y	指定条码左上角 Y 坐标
codeType:	条码类型: UPCA\EAN8\EAN13\39\39C\ITF\CODEBAR\128M\25\25C.....
height	条码高度(dot)
Readable 0	无人眼可识别码 1 :有人眼可识别码
rotation	顺时针旋转条码角度(0, 90, 180, 270)
narrow	窄条码比例因子(dot)
wide	宽条码比例因子(dot)
code	条码值

范例：

```
CodePrint ("100","100","39","96",1,0,2,4,"1000")
```

## 5、pdf 打印

### 5.1 操作步骤

- 1.添加 pdf-import\libs\目录下的所有程序包到自己的工程
- 2.添加 pdf-import 文件夹下的对应资源到工程(资源文件风格界面，可根据需要自己修改)
- 3.参照 pdf-import 文件夹下 AndroidManifest.xml 和 build.gradle 的文件,更改自己工程对应的文件
- 4.调用以下语句就可以了

```
Intent intent = new Intent();
intent.setClass(this, ChoosePDFActivity.class);
startActivity(intent);
```

注意：请参考 RTEExample 来编写，如发现一些资源文件没有的，请到 RTEExample 工程下找

## 附页 1 Codepage 对应列表

255 = "CHINESE"  
254 = "BIG5"  
253 = "UTF-8"  
252 = "SHIFT-JIS"  
251 = "EUC-KR"  
250 = "UNICODE"  
249 = "Kazakhstan"  
0 = "CP437 [U.S.A. = Standard Europe]"  
1 = "Katakana"  
2 = "CP850 [Multilingual]"  
3 = "CP860 [Portuguese]"  
4 = "CP863 [Canadian-French]"  
5 = "CP865 [Nordic]"  
6 = "WCP1251 [Cyrillic]"  
7 = "CP866 Cyrillic #2"  
8 = "MIK[Cyrillic /Bulgarian]"  
9 = "CP755 [East Europe, Latvian 2]"  
10 = "Iran"  
15 = "CP862 [Hebrew]"  
16 = "WCP1252 Latin I"  
17 = "WCP1253 [Greek]"  
18 = "CP852 [Latina 2]"  
19 = "CP858 Multilingual Latin I +Euro"  
20 = "Iran II"  
21 = "Latvian"  
22 = "CP864 [Arabic]"  
23 = "ISO-8859-1 [West Europe]"  
24 = "CP737 [Greek]"  
25 = "WCP1257 [Baltic]"  
26 = "Thai"  
27 = "CP720[Arabic]"  
28 = "CP855"  
29 = "CP857[Turkish]"  
30 = "WCP1250[Central Eurpoe]"  
31 = "CP775"  
32 = "WCP1254[Turkish]"  
33 = "WCP1255[Hebrew]"  
34 = "WCP1256[Arabic]"  
35 = "WCP1258[Vietnam]"  
36 = "ISO-8859-2[Latin 2]"

---

37 = "ISO-8859-3[Latin 3]"  
38 = "ISO-8859-4[Baltic]"  
39 = "ISO-8859-5[Cyrillic]"  
40 = "ISO-8859-6[Arabic]"  
41 = "ISO-8859-7[Greek]"  
42 = "ISO-8859-8[Hebrew]"  
43 = "ISO-8859-9[Turkish]"  
44 = "ISO-8859-15 [Latin 3]"  
45 = "Thai2"  
46 = "CP856"  
47 = "Cp874"  
48 = "TCVN3"  
49 = "Armenian"