

目录

| | |
|--|---|
| 1、PrintDriver 与 APP 通讯方法..... | 3 |
| 1.1 PrintDriver 类 | 3 |
| 1.2 创建一个 Handler 子类 ConnStateHandler, 处理 driver 发..... | 3 |
| 送过来的消息如下: | 3 |
| 1.3 通过以下方法设置 Handler: | 4 |
| 1.4 public void setHandler(Handler handler) | 4 |
| 1.5 Contants 连接状态类的状态标志: | 4 |
| 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): | 4 |
| 1.7 USB 连接方法 (该方法在 SDK 中无封装, 在此仅做建议使用): | 4 |
| 1.8 WIFI 连接方法 (该方法在 SDK 中无封装, 在此仅做建议使用): | 4 |
| 2、PrintDriver 类提供的方法 | 5 |
| 2.1 HsBuletoothPrintDriver 类: | 5 |
| 2.1.1 public static HsBluetoothPrintDriver getInstance()..... | 5 |
| 2.1.2 public synchronized void connect(BluetoothDevice device) | 5 |
| 2.1.3 public synchronized void start()..... | 5 |
| 2.2 HsUsbPrintDriver 类: | 5 |
| 2.2.1 public static HsUsbPrintDriver getInstance() | 5 |
| 2.2.2 public void setUsbManager(UsbManager usbManager) | 5 |
| 2.3 HsWifiPrintDriver 类: | 5 |
| 2.3.1 public static HsWifiPrintDriver getInstance()..... | 5 |
| 2.3.2 public boolean WIFISocket(String ip, int port)..... | 5 |
| 3、打印指令公共部分..... | 6 |
| 3.1 打印基本指令..... | 6 |
| 3.1.1 public synchronized int getState() | 6 |
| 3.1.2 public synchronized void stop() | 6 |
| 3.1.3 public boolean IsNoConnection() | 6 |
| 3.1.4 public void SetDefaultSetting()..... | 6 |
| 3.1.5 public void Begin()..... | 6 |
| 3.1.6 public void LF() | 6 |
| 3.1.7 public void CR() | 6 |
| 3.1.8 public void SelftestPrint()..... | 6 |
| 3.1.9 public void Beep(byte times, byte time) | 6 |
| 3.1.10 public void StatusInquiry() | 6 |
| 3.2 打印位置相关方法 | 6 |
| 3.2.1 public void SetRightSpacing(byte Distance) | 6 |
| 3.2.2 public void SetAbsolutePrintPosition(byte nL, byte nH)..... | 7 |
| 3.2.3 public void SetRelativePrintPosition(byte nL, byte nH) | 7 |
| 3.2.4 public void SetDefaultLineSpacing() | 7 |
| 3.2.5 public void SetLineSpacing(byte LineSpacing) | 7 |
| 3.2.6 public void SetLeftStartSpacing(byte nL, byte nH)..... | 7 |
| 3.2.7 public void SetAreaWidth(byte nL, byte nH) | 7 |

| | | |
|------------|--|-----------|
| 3.3 | 打印字符设置相关方法..... | 7 |
| 3.3.1 | public void SetCharacterPrintMode(byte CharacterPrintMode)..... | 7 |
| 3.3.2 | public void SetUnderline(byte UnderlineEn)..... | 8 |
| 3.3.3 | public void SetBold(byte BoldEn) | 8 |
| 3.3.4 | public void SetCharacterFont(byte Font) | 8 |
| 3.3.5 | public void SetRotate(byte RotateEn)..... | 8 |
| 3.3.6 | public void SetAlignMode(byte AlignMode)..... | 8 |
| 3.3.7 | public void SetInvertPrint(byte InvertModeEn) | 8 |
| 3.3.8 | public void SetFontEnlarge(byte FontEnlarge)..... | 9 |
| 3.3.9 | public void SetBlackReversePrint(byte BlackReverseEn) | 9 |
| 3.4 | 汉字控制命令相关方法..... | 9 |
| 3.4.1 | public void SetChineseCharacterMode(byte ChineseCharacterMode) | 9 |
| 3.4.2 | public void SelChineseCodepage()..... | 10 |
| 3.4.3 | public void CancelChineseCodepage() | 10 |
| 3.4.4 | public void SetChineseUnderline(byte ChineseUnderlineEn) | 10 |
| 3.5 | 钱箱控制命令方法 | 10 |
| 3.5.1 | public void OpenDrawer(byte DrawerNumber, byte PulseStartTime, byte PulseEndTime) | 10 |
| 3.6 | 切纸命令控制相关方法..... | 11 |
| 3.6.1 | public void CutPaper() | 11 |
| 3.6.2 | public void PartialCutPaper()..... | 11 |
| 3.6.3 | public void FeedAndCutPaper(byte CutMode) | 11 |
| 3.6.4 | public void FeedAndCutPaper(byte CutMode, byte FeedDistance) .. | 11 |
| 3.7 | 特殊打印相关方法 | 11 |
| 3.7.1 | public void AddCodePrint(BarcodeType CodeType, String data) | 11 |
| 3.7.2 | private void CODE_QR_CODE(String data) | 11 |
| 3.7.3 | public void printImage(Bitmap bitmap) | 12 |

1、PrintDriver 与 APP 通讯方法

1.1 PrintDriver 类

- 1) HsBluetoothPrintDriver 蓝牙
- 2) HsUsbPrintDriver USB
- 3) HsWifiPrintDriver WIFI

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  
                //有四种连接状态（在 Contants 类中可以找到）：  
                UNCONNECTED、  
                CONNECTED_BY_BLUETOOTH、  
                CONNECTED_BY_USB、  
                CONNECTED_BY_WIFI  
                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);
```

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  
FLAG_STATE_CHANGE  
FALG_FAIL_CONNECT  
FLAG_SUCCESS_CONNECT  
FLAG_MSG_READ  
TYPE_58    //限制打印图片最大的宽度为 58mm  
TYPE_80    //限制打印图片最大的宽度为 80mm
```

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):

```
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() {  
            HsWifiPrintDriver hsWifiPrintDriver = HsWifiPrintDriver.getInstance();  
            hsWifiPrintDriver.WIFISocket(ip,port);  
        }  
    })
```

```
    }).start();  
}
```

2、PrintDriver 类提供的方法

2.1 HsBluetoothPrintDriver 类:

2.1.1 public static HsBluetoothPrintDriver getInstance()

说明: HsBluetoothPrintDriver 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

2.2 HsUsbPrintDriver 类:

2.2.1 public static HsUsbPrintDriver getInstance()

说明: HsUsbPrintDriver 类实例化一个 USB 对象

2.2.2 public void setUsbManager(UsbManager usbManager)

说明: 程序刚启动时, 用于设置 UsbManager

参数: usbManager USB 管理器

2.3 HsWifiPrintDriver 类:

2.3.1 public static HsWifiPrintDriver getInstance()

说明: HsWifiPrintDriver 类实例化一个 WIFI 对象

2.3.2 public boolean WIFISocket(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

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

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

3.2 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： $0 \leq nL \leq 255$, $0 \leq nH \leq 255$

3.2.3 public void SetRelativePrintPosition(byte nL, byte nH)

说明：设置相对打印位置指令。

参数： $0 \leq nL \leq 255$, $0 \leq nH \leq 255$

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： $0 \leq \text{LineSpacing} \leq 255$

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： $0 \leq nL \leq 255$, $0 \leq nH \leq 255$

3.2.7 public void SetAreaWidth(byte nL, byte nH)

说明：设置打印区域宽度

参数： $0 \leq nL \leq 255$, $0 \leq nH \leq 255$

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 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

3.3.7 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

3.3.8 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

3.3.9 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

3.4 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 <= 255 0 <= 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 AddCodePrint(BarcodeType CodeType, 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  
}
```

3.7.2 **private void CODE_QR_CODE(String data)**

说明： 二维码打印指令。

参数： data 二维码数据

3.7.3 public void printImage(Bitmap bitmap, int type)

说明： 打印图片指令

参数： bitmap 图片数据

type 限制图片最大的打印宽度，TYPE_58: 58MM 纸

TYPE_80: 80MM 纸