

打印机 Android SDK 使用说明

----- 2018.4.17

打印机 Android SDK 使用说明	1
一：说明	4
二：常量和函数	4
1. 常量定义	5
2. 公用接口	8
1) .SDK_Version	8
2) .Prn_GetPortList	8
3) .Prn_Connect	9
4) .Prn_Disconnect	11
3. 热敏打印机接口	12
1) .Prn_PrinterInit	12
2) .Prn_Status	12
3) .Prn_PowerStatus	13
4) .int Prn_PrintText	13
5) .Prn_PrintEscText	15
6) .Prn_PrintString	16
7) .Prn_PrintBarcode	16
8) .Prn_PrintQRCode	18
9) .Prn_PrintBitmap	18
10) .Prn_PrintBmp	19
11) .Prn_CutPaper	20
12) .Prn_CutPaper	21
13) .Prn_OpenCashbox	21
14) .Prn_LineFeed	22
15) .Prn_MarkFeed	23
16) .Prn_SetCharacterSet	23
17) .Prn_SetInterCharacterSet	24
18) .Prn_SetLineSpacing	25
19) .Prn_SetFontStyle	25
20) .Prn_SetFontSize	26
21) .Prn_SetAlignment	27
22) .Prn_SendData	28
23) .Prn_ReadData	28
24) .Prn_BeginTransaction	29
25) .Prn_EndTransaction	29
26) .Prn_GetMsrTrack	30

4. 标签打印机接口	32
1) .prn_PageSetup.....	32
2) .prn_PagePrint.....	32
3) .prn_PageClear.....	33
4) .prn_PrinterStop.....	33
5) .prn_DrawBox.....	33
6) .prn_DrawLine	34
7) .prn_DrawInverseLine	34
8) .prn_DrawText	35
9) .prn_DrawTextAlign	37
10) .prn_DrawBarcode.....	38
11) .prn_BarcodeText.....	40
12) .prn_PrintBitmap.....	40
13) .prn_PrinterStatus.....	41
14) .prn_AddLabelCommand.....	41
15) .prn_DrawWatermark.....	42
5. 条码打印机接口	43
1) .bar_SetSize.....	43
2) .bar_SetGap.....	43
3) .bar_SetBline	45
4) .bar_SetOffset.....	45
5) .bar_SetShift	46
6) .bar_SetSpeed.....	46
7) .bar_SetDensity	47
8) .bar_SetDirection.....	47
9) .bar_SetPeel.....	48
10) .bar_SetCharacterSet	49
11) .bar_CLS.....	50
12) .bar_Print.....	50
12) .bar_DrawBar	50
13) .bar_Barcode	51
14) .bar_QRCode.....	52
15) .bar_Bitmap.....	53
16) .bar_DrawBox.....	54
17) .bar_PutPCX.....	54
18) .bar_SetReverse.....	55
19) .bar_Text	56
20) .bar_DownloadBAS.....	57
21) .bar_DownloadFile	57
22) .bar_KillFile.....	58
23) .bar_RunFile	58
24) .bar_SetCounter	59
25) .bar_PrintCounter	59
三: 例程.....	60

四：其他问题.....	61
1. 访问相关端口的权限问题	61
2. 批处理模式	62
3. 打印语言问题	63

一：说明

1. 名称：ZQPrinterSDK

本 SDK 支持 WIFI(或有线网口)、蓝牙(包括蓝牙 2.0 和 4.0)、USB、串口(包括 USB 转串口)共 4 种端口类型。指令方式主要以 AB-88H 和 AB-3XXM 系列手持热敏打印机、AB-322M 标签打印机以及 AB-F820 条码打印机进行测试。

2. 开发环境：

Eclipse

Android SDK 12-19

3. 库文件：

ZQPrinterSDK.jar

Armeabi\libzqcom.so (注:如没有使用到串口或 USB 转串口时, 可不加入本库文件)

4. 项目中使用库

1)新建 Android 项目。

2)请将 ZQPrinterSDK.jar 以及 Armeabi 目录放在项目的 libs 目录(注:如果不存在则新建文件夹命名为 *libs*)下, 然后在 Eclipse 中选中项目后 Refresh 一次。

3)热敏打印, 定义一个成员变量 `private ZQPrinterSDK prn = null`, 程序初始位置 `prn = new ZQPrinterSDK ()`. 即可调用相关接口

4)标签打印, 定义一个成员变量 `private ZQLabelSDK prn = null`, 程序初始位置 `prn = new ZQLabelSDK ()`. 即可调用相关接口

5)条码打印, 定义一个成员变量 `private ZQBarcodeSDK prn = null`, 程序初始位置 `prn = new ZQBarcodeSDK ()`. 即可调用相关接口

二：常量和函数

1. 常量定义

相关常量或参数通过 PrinterConst 类进行调用，比如返回值 PrinterConst.ErrorCode.SUCCESS

分类	名称	值	定义
返回值			
ErrorCode	SUCCESS	0	操作成功
	PORTERROR	-1	端口错误
	WRITEERROR	-2	写失败
	READERROR	-3	读失败
	NOPERMISSION	-4	没有权限
	INVALIDPARAM	-5	参数无效
	UKERROR	-99	未知错误
文本对齐方式			
Alignment	LEFT	0	左对齐
	CENTER	1	居中
	RIGHT	2	右对齐
字体效果			
Font	DEFAULT	0	默认效果
	FONTB	1	B 字体
	BOLD	2	加粗
	UNDERLINE	4	加下划线
	REVERSE	8	黑白反转
字体倍宽			
WidthSize	SIZE0	0	正常宽度
	SIZE1	16	放大 1 倍
	SIZE2	32	放大 2 倍
	SIZE3	48	放大 3 倍
	SIZE4	64	放大 4 倍
	SIZE5	80	放大 5 倍
	SIZE6	96	放大 6 倍
	SIZE7	112	放大 7 倍
字体倍高			
HeightSize	SIZE0	0	正常宽度
	SIZE1	1	放大 1 倍
	SIZE2	2	放大 2 倍
	SIZE3	3	放大 3 倍
	SIZE4	4	放大 4 倍
	SIZE5	5	放大 5 倍
	SIZE6	6	放大 6 倍
	SIZE7	7	放大 7 倍

打印机内部代码页

PageCode	PC437	0	请参阅相关代码页说明 和打印机指令集说明
	KATAKANA	1	
	PC850	2	
	PC860	3	
	PC863	4	
	PC865	5	
	WPC1253	6	
	TRAN	7	
	WPC1256	8	
	PC737	9	
	WPC1250	10	
	WPC1252	16	
	PC866	17	
	PC852	18	
	PC858	19	
USER	255		

打印机状态

PrinterStatus	ONLINE	0	在线
	COVEROPEN	-1	开盖
	EMPTY PAPER	-2	缺纸
	OFFLINE	-3	掉线
	OTHER ERROR	-4	其他错误

条码类型

Barcode	EAN13	103	请参阅相关条码说明
	JAN13	104	
	EAN8	105	
	JAN8	106	
	Code39	107	
	Code128	111	
	Code128_ZQ	112	

条码文本

BarcodeText	TEXT_NONE	0	不打印
	TEXT_ABOVE	1	在条码顶部打印
	TEXT_BELOW	2	在条码底部打印

电池状态

PowerStatus	HIGH	700	电量高
	MIDDLE	701	电量中
	LOW	702	电量低
	SMALL	703	电量少
	NO	704	没电

二维码内容类型

QRCode	NUM	49	数字
--------	-----	----	----

	ALPH	50	字母
	CHN	51	中文
位图大小			
BitmapSize	ZQSIZE0	1	不同的参数打印位图的大小不同，但是不同的机型定义可能不同，请以实际效果为准。
	ZQSIZE1	2	
	ZQSIZE2	3	
	ZQSIZE3	4	
	SIZE0	5	
	SIZE1	6	
	SIZE2	7	
	SIZE3	8	
磁卡轨道			
MsrTrack	TRACK1	1	手动模式返回轨道1
	TRACK2	2	手动模式返回轨道2
	TRACK3	3	手动模式返回轨道3
	TRACK12	4	手动模式返回轨道1和轨道2
	TRACK23	5	手动模式返回轨道2和轨道3
	TRACK123	6	自动模式返回轨道1、轨道2和轨道3
连接状态			
ConnectStatus	CONNECTED	0	连接
	DISCONNECTED	-1	断开
钱箱位置			
Cashbox	PIN3	0	连接在 PIN3 上，只有一个位置时固定连接在 PIN3
	PIN5	1	连接在 PIN5 上
切刀选项			
CutMode	FULLCUT	49	全切
	PARTIALCUT	50	半切
串口流控			
FlowControl	NONE	0	无流控
	XONXOFF	1	软流控
	HARDWARE	2	硬件流控
端口类型			
PortType	WIFI	0	WIFI 或有线网口
	BLUETOOTH	1	蓝牙
	USB	2	USB
	COM	3	串口或 USB 转串口
单位			
Unit	UNIT_INCH	0	英寸

	UNIT_MM	1	毫米
	UNIT_DOT	2	点
绘制图像模式			
ImgMode	MODE_OVERWRITE	0	OverWrite
	MODE_OR	1	Or
	MODE_XOR	2	Xor

2. 公用接口

1) .SDK_Version

```
String SDK_Version()
```

描述: 读取当前 SDK 的版本信息

参数: 无

返回值: 当前 SDK 的版本信息，比如 1.0.

备注: 无

2) .Prn_GetPortList

```
String[] Prn_GetPortList(int nPortType, Context context)
```

描述: 获取可用的打印机端口。

参数:

Int nPortType: 端口类型，如下表：

值		描述
PortType.WIFI	0	WIFI或有线网口
PortType.BLUETOOTH	1	蓝牙
PortType.USB	2	USB
PortType.COM	3	串口或USB转串口

`Context context`: `Activity` 的上下文,USB 端口类型不能为 `null`, 蓝牙端口类型如果传入 `null`, 则在没有开启蓝牙时不会自动弹出开启蓝牙, 其他端口类型可以传入 `null`。

返回值: `null`: 该类型端口不支持或没有找到打印机, 其他表示可用的打印机端口。

端口类型	返回值格式
WIFI	通过网络查询获取当前在线打印机的ip地址(当前机型是否支持网络查询请打印自检页获取当前打印机的版本号后咨询销售或售后技术支持), 如果未搜索到则返回当前网段IP最后一位为10的地址。
BLUETOOTH	返回“蓝牙地址”。仅返回当前设备已配对的打印机。
USB	返回“USB序号”, 序号从0开始
COM	返回当前系统可用的串口或USB转串口, 如ttyS0、ttyS1... 或ttyUSB0、ttyUSB1...

备注: 本接口可以在 `Prn_Connect` 之前调用。

3) `.Prn_Connect`

```
int Prn_Connect(String strAddr, Context context)
```

描述: 连接打印机。

参数:

`String strAddr`: 打印机端口参数, 格式如下:

WIFI 或 有线网口	IP:端口 (一般为 9100)	例子: 192.168.1.10:9100	注: 部分网口/WIFI 机型, 比如 88H, 88V 等, 如果主机在一定时间没有与打印机通讯时, 打印机将会自动断开网络连接, 这时候主机需要重新连接。
Bluetooth/BLE	蓝牙地址	例子: 00:12:6F:33:00:A5	注: 连续多次连接可能会偶尔失败, 或者偶尔连接耗时较长, 因此建议尽量减少连接断开的次数

USB	USB+序号 (从 0 开始)	例子: USB0、USB1...	注: 部分 USB 打印机没有数据返回功能, 需要读取数据的相关接口将无法使用, 比如 Prn_Status、Prn_ReadData 等。
COM 或 USB 转串口	串口名称:波特率	例子: ttyS1:9600 或 ttyUSB0:9600 或 s3c2410_serial0:9600	注: 不支持其他自定义的串口名称, 请联系设备供应商将串口改为标准 ttyS.

Context context: Activity 的上下文, USB 端口类型不能为 null, 蓝牙端口类型如果传入 null, 则在没有开启蓝牙时不会自动弹出开启蓝牙, 其他端口类型可以传入 null。

返回值:

ErrorCode.SUCCESS	打开成功
ErrorCode.INVALIDPARAM	打印机参数无效
ErrorCode.PORTERROR	端口错误
ErrorCode.NO PERMISSION	没有该端口的访问权限

备注: 1. 相关端口的访问权限问题请参阅本说明的第四部分《其他问题·访问相关端口的权限问题》。

2. 对于 USB 端口, 请不要将 SDK 实例化和端口连接放在主线程中一起调用。比如不要在 Activity 的 onCreate 或在一个按钮响应事件中同时进行实例化和连接操作, 应在程序初始化时进行 SDK 实例化, 在按钮响应事件或多线程中进行连接和打印。如需要在程序初始化中进行数据发送, 请在实例化后使用多线程进行数据发送。比如, 在进行 usb 打印机打印时, 以下代码将会导致首次使用打印机时会有 10 秒延迟:

```
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_main);  
//  
    prn = new ZQPrinterSDK();  
    prn.Prn_PrintString("This is a good sample!\r\n");  
}
```

应改成类似下面代码:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    //  
    prn = new ZQPrinterSDK();  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            int nRet = prn.Prn_Connect("USB0", context);  
            if (nRet != PrinterConst.ErrorCode.SUCCESS)  
            {  
                return;  
            }  
            prn.Prn_PrintString("This is a good sample!\r\n");  
        }  
    }).start();  
}
```

4) .Prn_Disconnect

```
int Prn_Disconnect()
```

描述: 断开打印机连接。

参数: 无

返回值: `ErrorCode.SUCCESS` 表示操作成功。

备注:

3. 热敏打印机接口

1) .Prn_PrinterInit

```
int Prn_PrinterInit()
```

描述:打印机初始化, 打印机非永久保存的参数恢复为初始值, 比如倍高倍宽、行间距等等, 相当于打印机 1B 40 指令。

参数:无

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:

2) .Prn_Status

```
int Prn_Status()
```

描述:获取打印机状态

参数:无

返回值:

PrinterStatus.ONLINE	状态正常
PrinterStatus.COVEROPEN	开盖
PrinterStatus.EMPTY PAPER	缺纸
PrinterStatus.OFFLINE	掉线
PrinterStatus.OTHERERROR	未连接 (WIFI 此时可能端口不通, 其他端口类

型可能是调用 Prn_Connect 失败)

备注:

1. 蓝牙在休眠时，查询状态将会返回缺纸，并唤醒蓝牙，唤醒后再查询状态才是正确的。
2. 此接口仅是简单判断状态，因为部分机型只有一个传感器，不区分开盖和缺纸，即开盖和缺纸时均返回缺纸。如果需要打印机更详细的状态，需要发送 0x10 0x04 n (n 可以为 0x01、0x02、0x03、0x04) 指令到打印机，然后接收打印机的返回值进行判断（各个机型的返回值定义请参阅对应机型的指令说明书）。
3. USB 端口时，部分机型可能不支持数据返回，状态查询将返回 OFFLINE。

3) .Prn_PowerStatus

```
int Prn_PowerStatus()
```

描述: 获取打印机电池状态**参数:** 无**返回值:**

PowerStatus.HIGH	电量高
PowerStatus.MIDDLE	电量中
PowerStatus.LOW	电量低
PowerStatus.SMALL	电量少
PowerStatus.NO	没电
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注: 此接口只有 AB-3X0M 系列才支持。

4) .int Prn_PrintText

```
int Prn_PrintText(String Data, int Alignment, int Attribute,
int TextSize)
```

描述:打印字符串

参数:

`String Data`:要打印的字符内容，一般最后应包含回车换行符，如果没有，打印机将会等到字符串满一行时再打印。

`int Alignment`:设定打印字符的对齐方式，可选以下值：

值		描述
<code>Alignment.LEFT</code>	0	左对齐
<code>Alignment.CENTER</code>	1	居中
<code>Alignment.RIGHT</code>	2	右对齐

`int Attribute`:设定打印字符的属性，可选以下值或它们的组合：

值		描述
<code>Font.DEFAULT</code>	0	FontA字体，12x24
<code>Font.FONTB</code>	1	FontB字体，9x17，仅对ASCII字符有效
<code>Font.BOLD</code>	2	加粗打印
<code>Font.UNDERLINE</code>	4	加下划线打印
<code>Font.REVERSE</code>	8	反转打印

注：不同的机型部分效果可能会有差别；在反转打印时，下划线无效。

`int TextSize`:设定打印字符的大小，可选以下值：

值		描述
<code>WidthSize.SIZE0</code>	0	原始宽度
<code>WidthSize.SIZE1</code>	16	宽度 x2
<code>WidthSize.SIZE2</code>	32	宽度 x3
<code>WidthSize.SIZE3</code>	48	宽度 x4
<code>WidthSize.SIZE4</code>	64	宽度 x5
<code>WidthSize.SIZE5</code>	80	宽度 x6
<code>WidthSize.SIZE6</code>	96	宽度 x7
<code>WidthSize.SIZE7</code>	112	宽度 x8
<code>HeightSize.SIZE0</code>	0	原始高度
<code>HeightSize.SIZE1</code>	1	高度 x2
<code>HeightSize.SIZE2</code>	2	高度 x3
<code>HeightSize.SIZE3</code>	3	高度 x4
<code>HeightSize.SIZE4</code>	4	高度 x5
<code>HeightSize.SIZE5</code>	5	高度 x6
<code>HeightSize.SIZE6</code>	6	高度 x7
<code>HeightSize.SIZE7</code>	7	高度 x8

注:WidthSize 和 HeightSize 可以进行组合, 比如 WidthSize1|HeightSize1 表示宽度和高度都 x2

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:无

5) .Prn_PrintEscText

```
int Prn_PrintEscText(String strPrint)
```

描述:打印字符串

参数:

String strPrint:按 Esc 格式的待打印字符串, 支持的 Esc 格式如下:

命令	描述
ESC [#]P	切刀。#替换为100则为全切, 其他值则为半切。
ESC [#]fP	走纸6行并切刀, #替换为100则为全切, 其他值则为半切。
ESC #B	打印内置的NV位图, #替换为相应的NV位图序号。
ESC [#]lF	走纸。#替换为相应走纸行数, 如果没有指定行数则走纸1行。
ESC bC	加粗打印
ESC uC	加下划线打印
ESC 1C	正常大小打印
ESC 2C	倍宽打印
ESC 3C	倍高打印
ESC 4C	倍宽倍高打印
ESC #hC	倍宽打印, #替换为倍宽的数值, 范围为0-8。
ESC #vC	倍高打印, #替换为倍高的数值, 范围为0-8。
ESC cA	居中打印
ESC rA	右对齐打印
ESC lA	左对齐打印
ESC N	正常打印

返回值:

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.PORTERROR</code>	Prn_Connect 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注:无

6) .Prn_PrintString

```
int Prn_PrintString(String strPrint)
```

描述:打印字符串

参数:

`String strPrint`:待打印字符串, 打印效果取决于当前的打印机状态。

返回值:

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.PORTERROR</code>	Prn_Connect 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注:无

7) .Prn_PrintBarcode

```
int Prn_PrintBarcode(String strBarcode, int Symbology, int
Height, int Width, int TextPosition)
```

描述:打印一维条码

参数:

`String strBarcode`: 条码内容, 注意所选的条码类型不同, 所支持的字符范围不同。

`int Symbology`: 条码类型, 可选以下值:

类型	支持的条码字符个数n	支持的条码字符范围Data
Barcode.EAN8 Barcode.JAN8	$7 \leq n \leq 8$	纯数字, 即 $48 \leq \text{data} \leq 57$
Barcode.EAN13 Barcode.JAN13	$12 \leq n \leq 13$	纯数字, 即 $48 \leq \text{data} \leq 57$
Barcode.CODE39	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57$ $65 \leq \text{data} \leq 90$ Data=32 36 37 43 45 46 47
Barcode.CODE128	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
Barcode.CODE128_ZQ	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$

注意: 部分机型只支持标准 CODE128 格式, 部分机型只支持我们 CODE128 格式, 或者两种格式均支持, 请根据实际情况使用。

`int Height`: 条码高度, 以点为单位, 范围为 1-255

`int Width`: 条码宽度, 范围为 2~7。注意: 如果条码内容较多, 条码的总宽度超出打印纸的可打印宽度, 条码将会不打印。

`int TextPosition`: 条码文本的打印位置, 可选以下值:

值	描述
BarcodeText.NONE	0 不打印
BarcodeText.ABOVE	1 打印在条码上方
BarcodeText.BELOW	2 打印在条码下方

注意: 部分机型可能不支持设置条码文本的打印位置, 比如 AB-321M V1.0B2 版本, JAN8 和 JAN13 固定打印在条码下方, CODE39 和 CODE128 则固定为不打印文本。

返回值:

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.INVALIDPARAM</code>	参数错误
<code>ErrorCode.PORTERROR</code>	Prn_Connect 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注: 如果该机型支持设置条码对齐方式, 则可通过 `Prn_SetAlignment` 接口进行设置, 注意, 部分机型不支持条码对齐方式时, 设置右对齐、居中时, 可能会因为超出打印范围而无法打印条码。

8) .Prn_PrintQRCode

```
int Prn_PrintQRCode(int nMode, String strCode)
```

描述:打印二维条码

参数:

int nMode:二维条码类型, 如下:

值		描述
QRCode.NUM	0	数字
QRCode.ALPH	1	字母
QRCode.CHN	2	中文

String strCode:二维条码内容

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.INVALIDPARAM	参数错误
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:部分机型不支持二维码打印, 则本接口无效。

9) .Prn_PrintBitmap

```
int Prn_PrintBitmap(Bitmap image, int Type)
```

描述:打印位图

参数:

Bitmap image:Android Bitmap 类型描述的位图数据

`int Type`:指定打印格式，可选以下值：（注意：打印机型号不同，对应值和对应的效果可能不同，请以实际效果为准。）

值		描述
<code>BitmapSize.ZQSIZE0</code>	0	正常大小
<code>BitmapSize.ZQSIZE1</code>	1	倍宽
<code>BitmapSize.ZQSIZE2</code>	2	倍高
<code>BitmapSize.ZQSIZE3</code>	3	倍宽倍高

返回值：

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.INVALIDPARAM</code>	参数错误
<code>ErrorCode.PORTERROR</code>	<code>Prn_Connect</code> 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注：

1. 打印机仅支持单色位图打印，如果 `Bitmap` 的原始图像为彩色位图，在自动转换后，输出图像效果可能较差。

2. 如果该机型支持位图对齐方式，请调用 `Prn_SetAlignment` 接口进行设置，如果该机型打印位图不支持对齐方式，如要实现对齐，请在图片的左边加上适当宽度的空白即可。

3. 因为各个机型对打印位图指令的兼容存在差别，如果通过本接口打印乱码，请尝试调用 `Prn_PrintBmp` 指令打印。

4. 图片的宽度不要超过打印机的可打印宽度，比如 58mm 最大宽度为 384 像素，80mm 最大宽度为 576 像素；另外注意如果使用了倍宽参数，则可打印的图像宽度需减半，否则在使用部分机型时可能会出现图像不准确。

10) `.Prn_PrintBmp`

```
int Prn_PrintBmp(String strFileName, int Type)
```

描述：打印位图

参数：

`Bitmap strFileName`:Windows BMP 单色位图文件。

`int Type`:指定打印格式，可选以下值：（注意：打印机型号不同，对应值和对应的效果可能不同，请以实际效果为准。）

值		描述
<code>BitmapSize.ZQSIZE0</code>	0	8点单密度，双倍宽3倍高
<code>BitmapSize.ZQSIZE1</code>	1	8点双密度，单倍宽3倍高
<code>BitmapSize.ZQSIZE2</code>	2	24点单密度，双倍宽单倍高
<code>BitmapSize.ZQSIZE3</code>	3	24点双密度，原始大小
<code>BitmapSize.SIZE0</code>	4	8点单密度，双倍宽3倍高
<code>BitmapSize.SIZE1</code>	5	8点双密度，单倍宽3倍高
<code>BitmapSize.SIZE2</code>	6	24点单密度，双倍宽单倍高
<code>BitmapSize.SIZE3</code>	7	24点双密度，原始大小

注意：因为我们的大部分打印机支持的 1B 2A 指令是不需要在最后加 0A 指令进行换行的，而 AB-88H 等机型(具体机型请咨询销售或售后技术支持)则需要最后增加 0A 指令进行换行。`SIZE0 ~ SIZE3` 相对于 `ZQSIZE0 ~ ZQSIZE3` 就是在最后增加了 0A 指令，因此 `SIZE0 ~ SIZE3` 等参数适用于 AB-88H 等打印机，而 `ZQSIZE0 ~ ZQSIZE3` 等参数适用于其他打印机，AB-88H 不仅需要 0A 指令进行换行，并且需要设置适当的行间距，否则打印位图将会出现重叠或者有间隔，并且在打印位图后需要再次设置行间距以适应文本打印。

`int Alignment`:设定打印位图的对齐方式，可选以下值：

值		描述
<code>Alignment.LEFT</code>	0	左对齐
<code>Alignment.CENTER</code>	1	居中
<code>Alignment.RIGHT</code>	2	右对齐

返回值：

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.INVALIDPARAM</code>	参数错误
<code>ErrorCode.PORTERROR</code>	Prn_Connect 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注：

11) `.Prn_CutPaper`

```
int Prn_CutPaper()
```

描述:切刀

参数:无

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:不带切刀的机型本接口将不起作用。

12) .Prn_CutPaper

```
int Prn_CutPaper(int nCutMode)
```

描述:切刀

参数: int nCutMode:切刀模式。可选以下值:

值		描述
CutMode.FULLCUT	0x30	全切
CutMode.PARTIALCUT	0x31	半切

注意: 部分机型虽然带了切刀, 但是切刀可能是只能半切或只能全切, 则不管传入的参数是什么, 都将半切或全切。

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:不带切刀的机型本接口将不起作用。对于较老的机型可能不支持本接口的指令, 则只能通过 Prn_CutPaper() 接口进行切刀, 如果该机型支持半切和全切, 请直接通过 Prn_SendData 接口发送相关的切刀指令进行切刀。

13) .Prn_OpenCashbox

```
int Prn_OpenCashbox(int iWitch)
```

描述:弹开钱箱

参数:

`int iWitch`:指定弹开哪个钱箱。不指定此参数则默认打开 PIN3 上的钱箱。

值		描述
Cashbox.PIN3	0	接在PIN3上的钱箱
Cashbox.PIN5	1	接在PIN5上的钱箱

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:无

14) .Prn_LineFeed

```
int Prn_LineFeed(int nLine)
```

描述:打印纸走空白行

参数:

`int nLine`:走纸行数

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:无

15) .Prn_MarkFeed

```
int Prn_MarkFeed()
```

描述:打印纸走纸到黑标

参数:无

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:

1. 部分机型不支持黑标打印，则本接口无效。
2. 部分特殊机型可能跟本接口的指令不兼容，则需要另外通过 Prn_SendData 接口进行控制。

16) .Prn_SetCharacterSet

```
int Prn_SetCharacterSet(String strCode)
```

描述:设置发送到打印机数据流的编码方式

参数:

String strCode:编码方式，比如：gb2312 为简体中文，big5 为繁体中文，其他编码格式请参阅 Android 的 System.setProperty("file.encoding", String value) 的相关说明。

返回值:

<code>ErrorCode.SUCCESS</code>	操作成功
--------------------------------	------

备注：默认的编码方式为英文，如要打印机其他语言的字符（比如中文），则一般在程序初始化时调用一次即可。

17) `.Prn_SetInterCharacterSet`

```
int Prn_SetInterCharacterSet(int Value)
```

描述：设置打印机的编码格式

参数：

`Int Value`:编码格式，如下表：

值		描述
PC437	0	请参阅打印机说明
KATAKANA	1	
PC850	2	
PC860	3	
PC863	4	
PC865	5	
WPC1253	6	
TRAN	7	
WPC1256	8	
PC737	9	
WPC1250	10	
WPC1252	16	
PC866	17	
PC852	18	
PC858	19	
USER	255	

返回值：

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.PORTERROR</code>	<code>Prn_Connect</code> 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注：

1. 部分机型可能需要打印机重启后生效，打印机内部编码方式为永久性生效，只需设置一次。
2. 部分机型可能仅支持拨码方式设置，此接口对该机型无效。

18) .Prn_SetLineSpacing

```
int Prn_SetLineSpacing(int nSpace)
```

描述:设置打印机行间距

参数:

`int nSpace`:行间距值大小,因为各款机型的打印头以及默认值可能不同,因此此处的移动单位以及可取的值范围各款机型可能不一致,请以实际为准,具体请参阅对应机型的 1B 33 指令。当 `nSpace=0` 时,恢复为打印机默认的初始值。

返回值:

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.PORTERROR</code>	<code>Prn_Connect</code> 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注:

19) .Prn_SetFontStyle

```
int Prn_SetFontStyle(int nStyle)
```

描述:设置字体效果

参数:

`int nFontStyle`: 设定打印字符的属性, 可选以下值或组合:

值		描述
<code>Font.DEFAULT</code>	0	FontA字体

Font.FONTB	1	FontB字体，一般比FontA要小
Font.BOLD	2	加粗打印
Font.UNDERLINE	4	加下划线打印
Font.REVERSE	8	反转打印

返回值：

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注： 如果与 `Prn_SetFontSize` 同时使用时，请放在 `Prn_SetFontSize` 的前面，否则将无法改变字体的大小。

20) .Prn_SetFontSize

```
int Prn_SetFontSize(int nSize)
```

描述：控制打印字符的倍宽倍高

参数：

`int nSize`:倍宽倍高设置，可选值为：

值		描述
WidthSize.SIZE0	0	原始宽度
WidthSize.SIZE1	16	宽度 x2
WidthSize.SIZE2	32	宽度 x3
WidthSize.SIZE3	48	宽度 x4
WidthSize.SIZE4	64	宽度 x5
WidthSize.SIZE5	80	宽度 x6
WidthSize.SIZE6	96	宽度 x7
WidthSize.SIZE7	112	宽度 x8
HeightSize.SIZE0	0	原始高度
HeightSize.SIZE1	1	高度 x2
HeightSize.SIZE2	2	高度 x3
HeightSize.SIZE3	3	高度 x4
HeightSize.SIZE4	4	高度 x5
HeightSize.SIZE5	5	高度 x6
HeightSize.SIZE6	6	高度 x7

HeightSize.SIZE7	7	高度 x8
------------------	---	-------

注:WidthSize 和 HeightSize 可以进行组合,比如 WidthSize1|HeightSize1 表示宽度和高度都 x2。

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:如果与 Prn_SetFontStyle 同时使用,请放在 Prn_SetFontStyle 的后面,否则在调用 Prn_SetFontStyle 后字体将会恢复为原始大小。

21) .Prn_SetAlignment

```
int Prn_SetAlignment(int nAlignment)
```

描述:设置打印内容的对齐方式。注:部分机型可能不支持,或者并不对所有内容都生效;一般对打印字符生效,部分机型对位图和条码也生效。

参数:

int Alignment:设定打印字符的对齐方式,可选以下值:

值		描述
Alignment.LEFT	0	左对齐
Alignment.CENTER	1	居中
Alignment.RIGHT	2	右对齐

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:

22) .Prn_SendData

```
int Prn_SendData(byte[] byData, int nLen)
```

描述:发送字节流数据到打印机

参数:

Byte[] byData:字节流数据, 可以是打印机控制指令也可以是打印字符串。

Int nLen:指定要发送的字节长度

返回值:

ErrorCode.SUCCESS	操作成功
ErrorCode.PORTERROR	Prn_Connect 失败
ErrorCode.WRITEERROR	发送数据到打印机失败

备注:

23) .Prn_ReadData

```
int Prn_ReadData(byte []by, int nLen, int nTimeout)
```

描述:接收打印机回传的数据

参数:

Byte []by:存放数据的缓冲区

int nLen:要接收的字节个数, 如果 nLen 大于 by 的长度, 将最多接收的字节个数为 by 的长度。

int nTimeOut:超时设置, 单位为 ms。

返回值:

>0	读取成功的字节数
----	----------

ErrorCode.PORTERROR

Prn_Connect 失败

备注:

1、 需要对打印机进行更多控制时才需要调用本接口进行接收数据。比如，因为不同的机型的查询指令返回值可能不同，如果 Prn_Status 接口无法准确的返回打印机的当前状态，可以发送 10 04 n 指令到打印机，然后接收打印机的返回数据，该指令说明请参阅打印机说明书的指令集说明。

2、 USB 端口时，部分机型可能不支持数据返回

24) .Prn_BeginTransaction

```
int Prn_BeginTransaction()
```

描述: 启用批处理模式

参数: 无

返回值:

ErrorCode.SUCCESS

操作成功

备注:

1. 启用批处理模式后，除 Prn_Status、Prn_PowerStatus、Prn_GetMsrTrack 这几个接口外，其他接口的数据将不会立即发送，而是缓冲在本地，直到退出批处理模式 EndTransaction 时才真正发送到打印机。

2. 批处理模式问题请参阅本说明的第四部分《其他问题·批处理模式》

25) .Prn_EndTransaction

```
int Prn_EndTransaction()
```

描述: 退出批处理模式

参数: 无

返回值：

<code>ErrorCode.SUCCESS</code>	操作成功
<code>ErrorCode.PORTERROR</code>	<code>Prn_Connect</code> 失败
<code>ErrorCode.WRITEERROR</code>	发送数据到打印机失败

备注：支持批处理打印模式的网口打印机，在退出批处理模式后，最好立即断开连接，以便下一个连接的主机发送的数据能够立即打印。批处理模式问题请参阅本说明的第四部分《其他问题·批处理模式》

26) `.Prn_GetMsrTrack`

```
byte[] Prn_GetMsrTrack(int nTrack, int nTimeout)
```

描述：手持打印机 AB-3X0M 系列打印机获取刷卡数据

参数：

`int nTrack`:要读取的轨道，如下表：

值		描述
<code>MsrTrack.TRACK1</code>	1	手动模式返回轨道1
<code>MsrTrack.TRACK2</code>	2	手动模式返回轨道2
<code>MsrTrack.TRACK3</code>	3	手动模式返回轨道3
<code>MsrTrack.TRACK12</code>	4	手动模式返回轨道1和轨道2
<code>MsrTrack.TRACK23</code>	5	手动模式返回轨道2和轨道3
<code>MsrTrack.TRACK123</code>	6	自动模式返回轨道1、轨道2和轨道3

手动模式：刷卡后调用本接口返回指定轨道的数据。

自动模式：刷卡后调用本接口固定返回 3 个轨道的数据。

切换刷卡器的操作模式，请参阅打印机说明书。

`int nTimeout`:读取数据的超时值，单位为毫秒。

返回值：

非空	操作成功，返回具体读取的轨道数据，轨道数据格式请参阅打印机说明书的指令集说明
空	操作失败，可能是刷卡器没有刷到数据。

备注:

4. 标签打印机接口

1) .prn_PageSetup

```
void prn_PageSetup (int pageHeight, int pageWidth)
```

描述:设置标签页面大小

参数:

int pageWidth: 标签宽度

int pageHeight: 标签高度

返回值:无

备注:打印机支持的最大页面尺寸为 576 点×2000 点，旋转 90 度则为 2000 点×576 点。

2) .prn_PagePrint

```
int prn_PagePrint(int rotate)
```

描述:打印标签

参数:

int rotate: 是否旋转，可选以下值：

值	描述
0	不旋转
1	顺时针旋转90度

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.WRITEERROR	-2	表示发送数据到打印机失败，可能打印机中途掉线

备注:无

3) .prn_PageClear

```
void prn_PageClear()
```

描述:清除页面, 之前设置的内容全部清空

参数:无

返回值: 无

备注:无

4) .prn_PrinterStop

```
int prn_PrinterStop()
```

描述:打印机停止打印, 并清空打印机缓存

参数:无

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.WRITEERROR	-2	表示发送数据到打印机失败, 可能打印机中途掉线

备注:无

5) .prn_DrawBox

```
void prn_DrawBox(int lineWidth, int x0, int y0, int x1, int y1)
```

描述:画方框

参数:

`int lineWidth`:线条的粗细, 以点为单位。

`int x0`:起始位置的 x 轴坐标

`int y0`:起始位置的 y 轴坐标

`int x1`:结束位置的 x 轴坐标

`int y1`:结束位置的 y 轴坐标

返回值: 无

备注:无

6) `.prn_DrawLine`

```
void prn_DrawLine(int lineWidth, int x0, int y0, int x1, int  
y1)
```

描述:画线

参数:

`int lineWidth`:线条的粗细,以点为单位。

`int x0`:起始位置的 x 轴坐标

`int y0`:起始位置的 y 轴坐标

`int x1`:结束位置的 x 轴坐标

`int y1`:结束位置的 y 轴坐标

返回值: 无

备注:无

7) `.prn_DrawInverseLine`

```
void prn_DrawInverseLine(int x0, int y0, int x1, int y1)
```

描述:画黑方块,一般用于整个区域的反转打印

参数:

int x0:起始位置的 x 轴坐标

int y0:起始位置的 y 轴坐标

int x1:结束位置的 x 轴坐标

int y1:结束位置的 y 轴坐标

返回值: 无

备注:整体区域反转打印效果如 **SAVE**, 要实现此效果, 需要调用 prn_DrawInverseLine 打印整个黑块, 然后再调用 prn_DrawText 正常打印内容即可, 注意此时 prn_DrawText 的参数中不需要启用反转。

8) .prn_DrawText

```
void prn_DrawText (int x, int y, String text, String fontName, int  
fontsize, int rotate, int bold, int underline, int reverse)
```

描述:打印文本

参数:

int x:起始位置的 x 轴坐标

int y:起始位置的 y 轴坐标

String text:文本内容

String fontName:字体名称,见下表。

`int fontsize`: 字体大小, 见下表。

英文字体:

fontName	fontsize	宽×高	描述
0	0	8×16	fixed pitch dot font --深色
0	1	16×16	fixed pitch dot font --深色
0	2	8×32	fixed pitch dot font --深色
0	3	16×32	fixed pitch dot font --深色
0	4	24×32	fixed pitch dot font --深色
0	5	16×32	fixed pitch dot font --深色
0	6	32×48	fixed pitch dot font --深色
1	0	16×48	fixed pitch dot font --斜体
2	0	16×16	fixed pitch dot font --浅色
4	0	24×48	fixed pitch dot font --浅色
4	1	24×72	fixed pitch dot font --浅色
4	2	48×48	fixed pitch dot font --深色
4	3	24×72	fixed pitch dot font --浅色
4	4	24×72	fixed pitch dot font --浅色
4	5	24×72	fixed pitch dot font --浅色
4	6	24×72	fixed pitch dot font --浅色
4	7	24×72	fixed pitch dot font --浅色
5	0	12×24	fixed pitch dot font --深色
5	1	12×24	fixed pitch dot font --深色
5	2	24×48	fixed pitch dot font --深色
5	3	24×48	fixed pitch dot font --深色
7	0	12×48	fixed pitch dot font --浅色
7	1	12×48	fixed pitch dot font --浅色
8	0	12×24	fixed pitch dot font --浅色
55	0	8×16	fixed pitch dot font --浅色

中文字体:

fontName	fontsize	宽×高	描述
55	高半字节表示宽度放大倍数 低半字节表示高度放大倍数	中文:16×16 英文:8×8	fixed pitch dot font
24	高半字节表示宽度放大倍数 低半字节表示高度放大倍数	中文:24×24 英文:12×12	fixed pitch dot font
32	高半字节表示宽度放大倍数 低半字节表示高度放大倍数	中文:32×32 英文:16×32	
128	0	中文:128×128 英文:64×128	注:该字体一般用于水印打印, 部分机型可能不支持

例子: `fontsize=0x11`, 表示宽度和高度都放大 1 倍, 即 2 倍宽和 2 倍高。

`fontsize=0x10`, 表示宽度放大 1 倍, 高度保持原始大小

`fontsize=0x23`,表示宽度放大 2 倍,高度放大 3 倍。

`int rotate`:是否旋转,可选以下值:

值	描述
0	不旋转
1	顺时针旋转90度
2	顺时针旋转180度
3	顺时针旋转270度

`int bold`:是否加粗,可选值为 0-5, 0 时为正常不加粗。

`int underline`:是否加下划线,可选以下值:

值	描述
0	无下划线
1	1点宽下划线

`int reverse`:是否黑白反转,可选以下值:

值	描述
0	不反转,即白底黑字
1	黑白反转,即黑底白字

返回值:无

备注:无

9) .prn_DrawTextAlign

```
void prn_DrawTextAlign (int x0, int y0, int x1, int y1, int
nAlign, String text, String fontName, int fontsize, int rotate,
int bold, int underline, int reverse)
```

描述:指定一个方块区域,按指定的对齐方式打印文本

参数:

`int x0`:起始位置的 x 轴坐标

`int y0`:起始位置的 y 轴坐标

`int x1`:结束位置的 x 轴坐标

`int y1`:结束位置的 y 轴坐标

`int nAlign`:对齐方式, 可选以下值:

值	描述
0	左对齐
1	居中
2	右对齐

`String text`:文本内容

`String fontName`:字体名称, 见 `prn_DrawText` 接口的同参数说明。

`int fontsize`:字体大小, 见 `prn_DrawText` 接口的同参数说明。

`int rotate`:是否旋转, 可选值见 `prn_DrawText` 接口的同参数说明。

`int bold`:是否加粗, 可选值为 0-5, 0 时为正常不加粗。

`int underline`:是否加下划线, 可选值见 `prn_DrawText` 接口的同参数说明。

`int reverse`:是否黑白反转, 可选值见 `prn_DrawText` 接口的同参数说明。

返回值: 无

备注:注意对齐方式仅是水平方向的对齐, 纵向对齐需要自己计算 y 轴的起始坐标。

10) `.prn_DrawBarcode`

```
int prn_DrawBarcode(int x, int y, String text, int barcodetype,
int rotate, int linewidth, int height)
```

描述:打印条码

参数:

`int x`:起始位置的 x 轴坐标

`int y`: 起始位置的 `y` 轴坐标

`String text`: 条码内容

`int barcodetype`: 标签类型, 可选值对应的条码类型如下:

值	描述
0	JAN13 (EAN13)
1	JAN8 (EAN8)
2	CODE39
3	CODE93
4	CODE128
5	CODABAR
6	ITF
7	UPC-A
8	UPC-E
9	EAN13+2
10	EAN13+5
11	EAN8+2
12	EAN8+5
13	UPCA+2
14	UPCA+5
15	UPCE+2
16	UPCE+5
17	Postnet
18	MSI
19	QR (二维码)
128	CODE128
39	CODE39
93	CODE93

`int rotate`: 是否旋转, 可选以下值:

值	描述
0	不旋转
1	顺时针旋转90度
2	顺时针旋转180度
3	顺时针旋转270度

`int linewidth`: 一维条码 (`barcodetype != 19`) 时为窄条码比例因子, 二维条码 (`barcodetype = 19`) 时表示 QR 码类型, 可选范围为 1 或 2。

`int height`: 一维条码 (`barcodetype!=19`) 时指定以点为单位的条码高度, 二维条码 (`barcodetype=19`) 时, 为条码样式宽度/高度级别, 可选值为 1~32。

返回值:无

备注:无

11) `.prn_BarcodeText`

```
void prn_BarcodeText (String fontName, int fontsize, int  
offset)
```

描述:启用打印条码文本

参数:

`String fontName`:字体名称, 见 `prn_DrawText`。

`int fontsize`:字体大小, 见 `prn_DrawText`。

`int offset`:条码文本与条码的间隙。

返回值:无

备注:1) 必须在 `prn_Barcode` 之前先调用才能生效。

2) 条码文本固定跟条码居中对齐。

3) 调用此接口后, 当前标签打印均生效, 如果下一个条码希望不打印条码文本, 则需要在下个条码打印前调用 `prn_AddLabelCommand("BARCODE-TEXT OFF\r\n");`

12) `.prn_PrintBitmap`

```
void prn_PrintBitmap(int startx, int starty, Bitmap bmp)
```


描述:打印 Bitmap 位图

参数:

`int startx`:起始位置的 x 轴坐标

`int starty`: 起始位置的 y 轴坐标

`Bitmap bmp`:Bitmap 类型表示的位图内容

返回值: 无

备注:无

13) .prn_PrinterStatus

```
int prn_PrinterStatus()
```

描述:返回打印机状态

参数:无

返回值: -1:表示打印机无反应

0:打印机状态正常

>0:打印机异常,其中 bit0 表示打印机纸仓盖开, bit1 表示打印机缺纸, bit2 表示温度异常, bit3 表示打印机忙, bit4 表示电量低, bit5 表示正在打印

备注:无

14) .prn_AddLabelCommand

```
int prn_AddLabelCommand(String strCmd)
```

描述:发送标签打印机的相关指令

参数:

`String strCmd`:相关指令, 具体指令请参阅打印机指令集

返回值: 无

备注:仅用于需要实现特殊打印效果。

15) `.prn_DrawWatermark`

```
void prn_DrawWatermark (int x, int y, String text, String
fontName,int fontsize, int rotate, int pattern, int gray)
```

描述:打印水印

参数:

`int x`:起始位置的 x 轴坐标

`int y`:起始位置的 y 轴坐标

`String text`:文本内容

`String fontName`:字体名称, 见 `prn_DrawText` 接口相关参数说明。

`int fontsize`:字体大小, 见 `prn_DrawText` 接口相关参数说明。

`int rotate`:是否旋转, 见 `prn_DrawText` 接口相关参数说明。

`int pattern`:水印模板, 可选以下值:

值	描述
100	不采用模板功能
101	横线条
102	竖线条
103	右斜上线条

104	左斜上线条
105	井形状(正方形)线条
106	菱形线条
107	点

`int gray`:灰度等级,可选 0-4,0-black,4-gray。

5. 条码打印机接口

1) .bar_SetSize

```
int bar_SetSize (double dWidth, double dHeight, int nUnit)
```

描述:设置标签页面大小

参数:

`double dWidth`:标签宽度

`double dHeight`:标签高度

`int nUnit`: 单位,可选以下值:

值		描述
Unit.UNIT_INCH	0	英寸
Unit.UNIT_MM	1	毫米
Unit.UNIT_DOT	2	点

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

2) .bar_SetGap

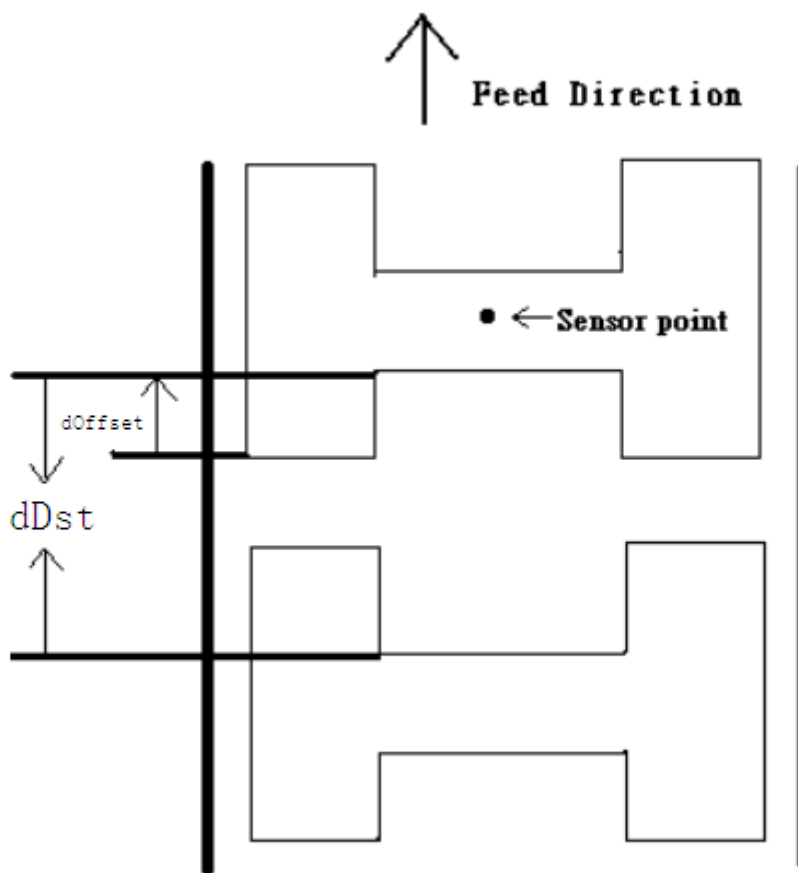
```
int bar_SetGap (double dDst, double dOffset, int nUnit)
```

描述: 设置两张标签之间的间隙距离

参数:

double dDst: 定义标签间隙高度

double dOffset: 定义标签间隙高度的补偿值, 一般标签此值为 0. 特殊标签如下图:



int nUnit: 单位, 可选以下值:

值		描述
Unit.UNIT_INCH	0	英寸
Unit.UNIT_MM	1	毫米
Unit.UNIT_DOT	2	点

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

3) .bar_SetBline

```
int bar_SetBline (double dHeight, double dExtra, int nUnit)
```

描述: 设置黑标高度及使用者定义标签打印完后标签额外送出的长度

参数:

`double dHeight`: 黑标高度

`double dExtra`: 额外送出纸张长度, 0 时为连续纸

`int nUnit`: 单位, 可选以下值:

值		描述
<code>Unit.UNIT_INCH</code>	0	英寸
<code>Unit.UNIT_MM</code>	1	毫米

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

4) .bar_SetOffset

```
int bar_SetOffset (double dOffset, int nUnit)
```

描述: 设置标签打印完后额外推出的长度, 特别是在撕纸时, 可用来调整标签停止的位置, 在打印下一张时打印机会将原先多推出或少推出的部分以回拉的方式补偿回来

参数:

`double dOffset`: 偏移距离

`int nUnit`: 单位, 可选以下值:

值		描述
<code>Unit.UNIT_INCH</code>	0	英寸
<code>Unit.UNIT_MM</code>	1	毫米

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:5) `.bar_SetShift`

```
int bar_SetShift (int nShift)
```

描述: 设置打印零点位置偏差

参数:

`int nShift`: 偏差值, 单位为每点行

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:6) `.bar_SetSpeed`

```
int bar_SetSpeed (int nSpeed)
```

描述:设置打印机的打印速度

参数:

int nSpeed: 速度, 范围为 $1 \leq nSpeed \leq 6$

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

7) .bar_SetDensity

```
int bar_SetDensity (int nDensity)
```

描述:设置打印机的打印浓度

参数:

int nDensity: 浓度, 范围为 $0 \leq nDensity \leq 15$

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

8) .bar_SetDirection

```
int bar_SetDirection (int nDirection, int nMirror)
```

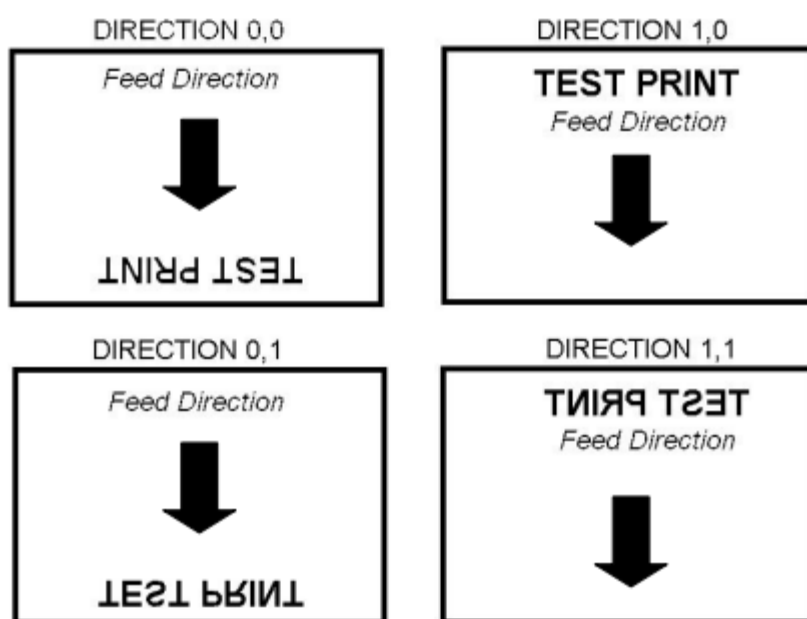
描述: 设置打印机的方向和镜像翻转

参数:

int nDirection: 方向, 0 或 1

int nMirror: 镜像翻转, 0 或 1

效果如下图所示:



返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

9) .bar_SetPeel


```
int bar_SetPeel (boolean bFlag)
```

描述: 启动/关闭自动剥纸功能, 开启时, 打印机会在每印完一张时暂停, 直到标签纸被取走后才会打印下一张。

参数:

boolean bFlag: true 时开启, false 时关闭

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

10) .bar_SetCharacterSet

```
int bar_SetCharacterSet (String strCode)
```

描述: 设置发送到打印机数据流的编码方式

参数:

String strCode: 编码方式, 比如: gb2312 为简体中文, big5 为繁体中文, 其他编码格式请参阅 Android 的 System.setProperty("file.encoding", String value) 的相关说明。

返回值:

ErrorCode.SUCCESS	操作成功
-------------------	------

备注: 默认的编码方式为英文, 如要打印其他语言的字符 (比如中文), 则一般在程序初始化时调用一次即可。

11) .bar_CLS

```
void bar_CLS ()
```

描述:清除缓存

参数:

无

返回值:无

备注: 此函数必须置于 bar_SetSize() 函数之后。

12) .bar_Print

```
int bar_Print (int nSets, int nCopies)
```

描述:打印存于缓存中的标签

参数:

int nSets: 打印张数

int nCopies: 每张标签需重复打印的张数

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

12) .bar_DrawBar

```
int bar_DrawBar (int nX, int nY, int nWidth, int nHeight)
```

描述:画线或长条物

参数:

int nX:线条左上角 x 坐标, 单位为点

int nY:线条左上角 y 坐标, 单位为点

int nWidth:线条宽度, 单位为点

int nHeight:线条高度, 单位为点

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

13) .bar_Barcode

```
int bar_Barcode (int nX, int nY, String strCodeType, int
nHeight, int nReadable, int nRotation, int nNarrow, int nWide,
String strCode)
```

描述:打印一维条码

参数:

int nX:条码左上角 x 坐标, 单位为点

int nY:条码左上角 y 坐标, 单位为点

String strCodeType:条码类型, 目前只支持“128”。

int nHeight:条码高度, 单位为点

`int nReadable`: 0 不打印文本, 1 为打印文本

`int nRotation`: 是否旋转, 可选以下值:

值	描述
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

`int nNarrow`: 窄条码比例因子

`int nWide`: 宽条码比例因子

`String strCode`: 条码内容

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注: 目前只完成 CODE128 以及部分参数, 请以当前打印机实际为准.

14) `.bar_QRCode`

```
int bar_QRCode (int nX, int nY, int nWidth, int nHeight, String
strCode)
```

```
int bar_QRCode (int nX, int nY, int nWidth, int nHeight, String
strCode, Bitmap markBMP)
```

描述: 打印二维条码

参数:

`int nX`:条码左上角 x 坐标, 单位为点

`int nY`:条码左上角 y 坐标, 单位为点

`int nWidth`:条码宽度, 单位为点

`int nHeight`:条码高度, 单位为点

`String strCode`:条码内容

`Bitmap markBMP`:此参数为可选, 为二维码中心打印水印 Logo。

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

15) `.bar_Bitmap`

```
int bar_Bitmap (int nX, int nY, int nMode, String strFileName)
```

```
int bar_Bitmap (int nX, int nY, int nMode, Bitmap image)
```

描述:打印图形

参数:

`int nX`:图形左上角 x 坐标, 单位为点

`int nY`:图形左上角 y 坐标, 单位为点

`int nMode`:绘制图形方式, 可选如下方式:

<code>PrinterConst.ImgMode.MODE_OVERWRITE</code>	0	OverWrite
<code>PrinterConst.ImgMode.MODE_OR</code>	1	Or
<code>PrinterConst.ImgMode.MODE_XOR</code>	2	Xor

String strFileName:Windows BMP 单色位图文件

或

Bitmap image:Android Bitmap 类型描述的位图数据

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

16) .bar_DrawBox

```
int bar_DrawBox (int nXStart, int nYStart, int nXEnd, int
nYEnd, int nThickness)
```

描述:绘制方框

参数:

int nXStart:方框左上角 x 坐标, 单位为点

int nYStart:方框左上角 y 坐标, 单位为点

int nXEnd:方框右下角 x 坐标, 单位为点

int nYEnd:方框右下角 y 坐标, 单位为点

int nThickness:方框线条粗细, 单位为点

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

17) .bar_PutPCX

```
int bar_PutPCX (int nX, int nY, String strFileName)
```

描述:打印 PCX 文件

参数:

int nX:图形左上角 x 坐标, 单位为点

int nY:图形左上角 y 坐标, 单位为点

String strFileName:PCX 文件全路径

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

18) .bar_SetReverse

```
int bar_SetReverse (int nX, int nY, int nWidth, int nHeight)
```

描述:将指定区域做黑白反转打印

参数:

int nX:区域左上角 x 坐标, 单位为点

int nY:区域左上角 y 坐标, 单位为点

int nWidth:区域宽度, 单位为点

int nHeight:区域高度, 单位为点

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

19) .bar_Text

```
int bar_Text(int nX, int nY, String strFontName, int
nRotation, int nXMulti, int nYMulti, String strContent)
```

描述:打印文字

参数:

`int nX`:文字方块左上角 x 坐标, 单位为点

`int nY`:文字方块左上角 y 坐标, 单位为点

`String strFontName`:字体名称, 可选以下字体

字体名称	宽 x 高
1	8 x 12
2	12 x 20
3	16 x 24
4	24 x 32
5	32 x 48
7	14 x 25
8	21 x 27
中文	24 x 24 (英文 12 x 24)

`int nRotation`:是否旋转, 可选以下值:

值	描述
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

`int nXMulti`:宽放大倍数, 范围为 1 ~ 10

`int nYMulti`:高放大倍数, 范围为 1 ~ 10

`String strContent`:文字内容, 若文字内容包含双引号 (") 时, 请用 \["] 取代

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

20) .bar_DownloadBAS

```
int bar_DownloadBAS(String strFileName, String strContent)
```

描述: 下载程序档

参数:

`String strFileName`:程序档名称, 名称不能 8 个字节, 后缀名必须为 BAS

`String strContent`:程序档内容, 不需要包括 DOWNLOAD 和 EOP, 接口会自动添加。

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

21) .bar_DownloadFile

```
int bar_DownloadFile(String strFileName, String
strLocalFileName)
```

描述: 下载文件档

参数:

`String strFileName`: 文件档名称, 名称不能 8 个字节, 后缀名不能超过 3 个字节。打印机支持的文件档格式请以当前打印机的实际情况为准。

`String strLocalFileName`: 文件档的本地全路径

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

22) .bar_KillFile

```
int bar_KillFile(String strFileName)
```

描述: 删除存储的程序档或文件档

参数:

`String strFileName`: 程序档或文件档名称, 名称不能 8 个字节, 后缀名不能超过 3 个字节。必须传入完整的名称。

返回值:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	操作成功
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	参数错误

备注:

23) .bar_RunFile

```
int bar_RunFile(String strFileName)
```

描述:运行程序档

参数:

String strFileName:程序档名称,名称不能 8 个字节,后缀名不能超过 3 个字节。可包含".BAS"或不包含。

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

24) .bar_SetCounter

```
int bar_SetCounter(int n, int nStep, String strExp)
```

描述:设置计数器

参数:

int n:计数器序号,范围为 0~50

int nStep:计数器增量,范围为-999999999 <= nStep <= 999999999

String strExp:初始字符串

返回值:

PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

25) .bar_PrintCounter

```
int bar_PrintCounter(int nX, int nY, String strFontName, int
nRotation, int nXMulti, int nYMulti, int nCounter)
```

描述:打印计数器

参数:

int nX:文字方块左上角 X 坐标, 单位为点

int nY:文字方块左上角 Y 坐标, 单位为点

String strFontName:字体名称, 请参考 bar_Text 接口说明

int nRotation:是否旋转, 请参考 bar_Text 接口说明

int nXMulti:宽放大倍数, 范围为 1 ~ 10

int nYMulti:高放大倍数, 范围为 1 ~ 10

int nCounter: 计数器序号, 范围为 0~50

返回值:

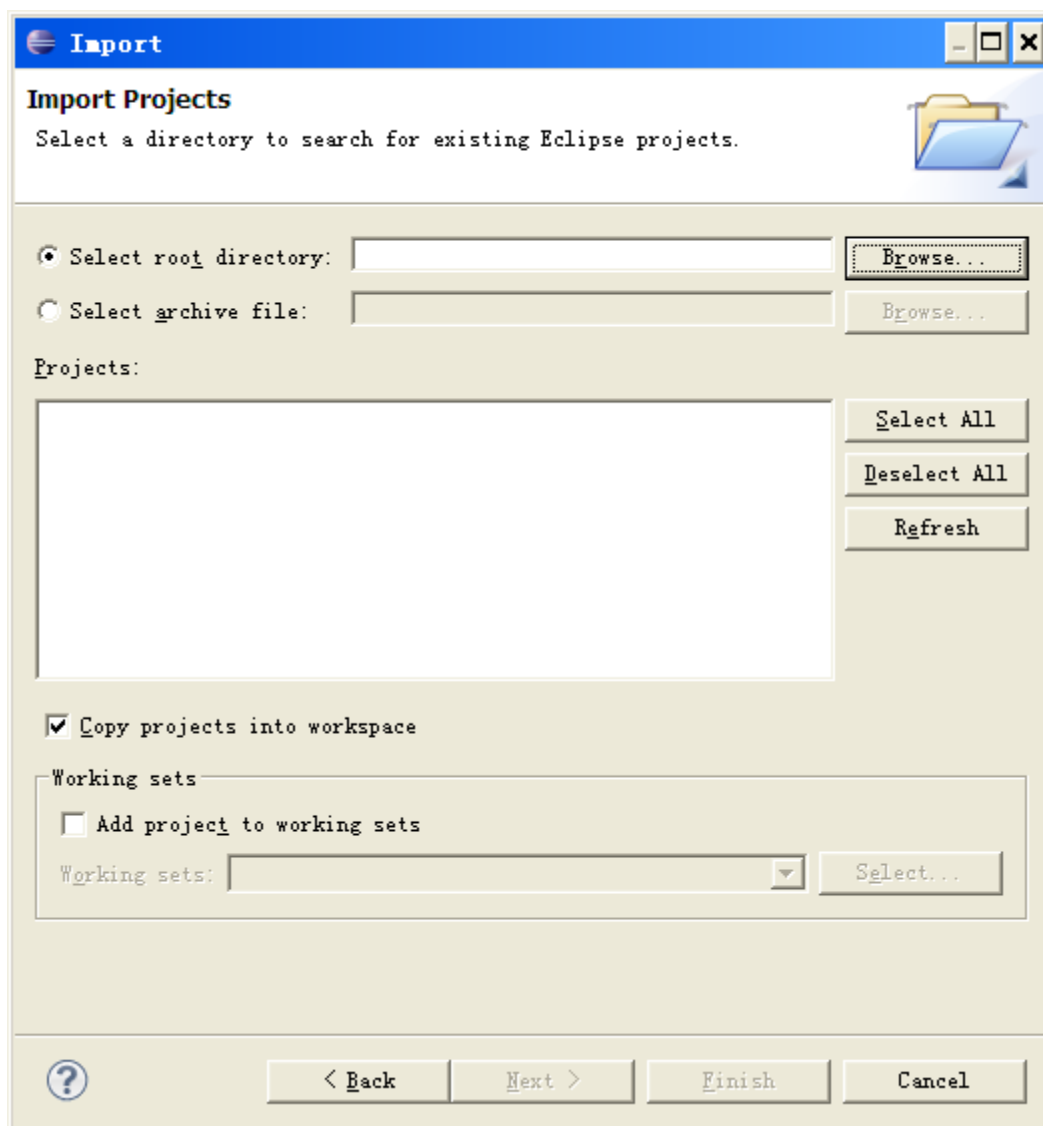
PrinterConst.ErrorCode.SUCCESS	0	操作成功
PrinterConst.ErrorCode.INVALIDPARAM	-2	参数错误

备注:

三: 例程

ZQPrinterSDKDemo 是 Eclipse 环境下的热敏打印机例程。ZQLabelSDKDemo 是 Eclipse 环境下的标签打印机例程。ZQBarcodeSDKDemo 是 Eclipse 环境下的条码打印机例程。

在 Eclipse 主界面下, 点击 File->Import->General->Existing Projects into Workspace



点击 Browse 选择到 ZQPrinterSDKDemo 或 ZQLabelSDKDemo 或 ZQBarcodeSDKDemo 目录, 然后点击 Finish 完成导入即可。

四：其他问题

1. 访问相关端口的权限问题

各种类型的端口均需要有相应的访问权限, 否则在运行时将会导致程序崩溃。比如:

WIFI: 需在 AndroidManifest.xml 中加入:

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

蓝牙:需在 AndroidManifest.xml 中加入:

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

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

串口:如果没有访问权限则需要 root 后用 chmod 指令增加相关端口的读写权限。(注:SDK 中已经尝试调用 su 获取读写权限,不过由于不同供应商的平板可能存在差异,如果 SDK 无法自动获取读写权限,则需要另外实现获取读写权限,具体请咨询平板的供应商,或者请确保 root 后再试。)

USB:

- 1) 需要有 android.hardware.usb.host 权限,如果没有将无法使用 USB 打印机,增加该权限需要有 root 权限。检查或修改如下:

/system/etc/permissions 目录下的 android.hardware.usb.host.xml 文件,以及 handheld_core_hardware.xml 或 tablet_core_hardware.xml 文件中,<permissions> 里面应包含<feature name="android.hardware.usb.host"/>。

- 2) 每次系统启动后首次使用会弹出系统授权窗口,在该窗口中需要勾选允许访问。如想系统默认授权不弹出授权窗口,则需要重新编译 Android 系统,或者将该 USB 端口与程序进行绑定(本方法在开机并且 USB 打印机在线后将会自动运行该程序,并且只能实现绑定一个访问该 USB 端口的程序,如果有多个程序需要访问 USB 打印则其他程序在访问时仍然需要手动授权)。

2. 批处理模式

一般打印机是独占设备,同一时间只能与一台主机进行通讯,如果同时有多台主机发送数据到打印机,打印机可能会出现打印混乱,因此如果有多台主机使用同一台打印机时(比如餐饮行业的厨打),一般是需要将其中一台主机作为网络打印服务器,其他主机将打印数据发送到网络打印服务器,统一由网络打印服务器将打印数据发送到打印机。我们 AB-88H 网口/WIFI 打印机因为支持批处理模式,因此可以支持多台主机与打印机同时通讯,使用时注意:

- 1) 一般仅支持 5 台主机同时连接,因此如果同时连接的主机太多,则仍然需要上位机实现网络打印服务器。

- 2) 打印机的判断规则:A. 哪台主机先连接则先打印该台主机发送的打印数据. B. 只有前一个连接的主机连接断开后, 才打印下一台连接主机的打印数据. 因此, 要实现网络打印服务功能, 必须调用批处理的相关接口. 同时, 连接主机在发送完所有打印数据后 (即调用 Prn_EndTransaction 接口), 应立即断开连接 (注: 如果没有主动断开连接, 打印将在一定超时时自动断开连接), 以便下一台主机的数据能够立即打印.

3. 打印语言问题

- 1) 首先确认打印机支持的语言, 一般打印机支持拨码设置语言, 或者通过 Windows 上位机软件进行设置. 部分机型 (如 AB-88H), 还支持通过接口 Prn_SetInterCharacterSet 进行单字节语言 (中文属双字节语言) 的切换.
- 2) 通过 Prn_SetCharacterSet 接口设置发送到打印机的编码格式, 比如简体中文为 gb2312, 繁体中文为 big5.