Android ESC SDK 说明文档

一 SDK 的组成

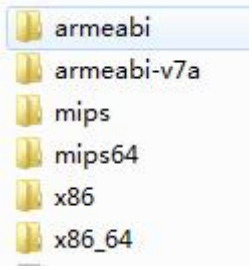
1.1 SDK jar 包

1).连接接口有蓝牙、USB、WIFI 及串口。

2).打印指令的接口，例如打印文本、条码、图片等。

1.2 SO 库

如图所示：



二 连接方式

2.1 蓝牙连接

int portOpenBT(Context context, String portSetting)

参数：

context ：上下文对象。

portSetting：蓝牙地址。

例子：

Print.portOpenBT(context,MAC)

MAC:打印机的蓝牙地址。

返回：

0：连接成功。

-1：连接失败。

2.2 WIFI 连接

int portOpenWifi(Context context, String portSetting)

参数：

context　：上下文对象。

portSetting：ip 地址。

例子：

Print.portOpenWifi(context,IP)

IP:打印机的 IP 地址。

返回：

0：连接成功。

-1：连接失败。

2.3 USB 连接

int PortOpen(Context context, UsbDevice usbdevice)

参数：

context　：上下文对象。

usbdevice：usb 设备对象。

例子：

Print.PortOpen(context,usbdevice)

usbdevice：UsbDevice 的对象。

返回：

0：连接成功。

-1：连接失败。

2.4 串口连接

int PortOpen(Context context, String portSetting)

参数：

context ：上下文对象。

portSetting：

〝Serial,〞+port+〞,〞+baudrate

port:串口的节点。（不同的机型不一样）例如 /dev/ttyS1

baudrate：波特率。 例如：9600

例子：

```
Print.PortOpen(context,
                “Serial,”+port+”,”+baudrate)
```

返回：

0：连接成功。

-1：连接失败。

2.5 断开连接

boolean PortClose()

参数：

无

例子：

Print.PorClose()

返回：

true：断开成功。

false：断开失败。

2.6 端口是否连接

boolean IsOpened()

注意：

该接口不能实时检测蓝牙连接情况，想要实时检测蓝牙连接情况需要监听系统蓝牙广播。

参数：

无

例子：

Print.IsOpened()

返回：

true：蓝牙已连接。

false：蓝牙未连接。

三 打印指令

3.1 走纸

int PrintAndLineFeed()

参数：

无

例子：

Print.PrintAndLineFeed()

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.2 打印后走纸

int PrintAndFeed(int distance)

参数：

distance:走纸长度（单位：distance*y 轴的移动单元 mm）。

例子：

Print.PrintAndFeed(distance)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.3 打印后回退

int PrintAndReverseFeed(int distance)

参数：

distance:回退长度（单位：distance*y 轴的移动单元 mm）。

例子：

Print.PrintAndReverseFeed(distance)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.4 打印并走纸 N 行

int PrintAndFeedNLine(byte distance)

参数：

distance:行数。

例子：

Print.PrintAndFeedNLine(lines)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.5 打印并回退 N 行

int PrintAndReverseFeedNLine(int distance)

参数：

distance:行数。

例子：

Print.PrintAndReverseFeedNLine(lines)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.6 设置行间距

1）int SetDefaultTextLineSpace()

注释：设置默认的行间距。（3.75mm）

例子：

```
Print.SetDefaultTextLineSpace()
```

2）int SetTextLineSpace(byte lineSpace)

参数：

lineSpace:行间距（lineSpace*y 方向的移动单元 mm）

例子：

```
Print.SetTextLineSpace(byte lineSpace)
```

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.7 选择字体

int SelectCharacterFont(byte characterFont)

参数：

characterFont：

0：FontA 大字体。

1：FontB 小字体。

例子：

Print.SelectCharacterFont(0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.8 设置语言

int SetCharacterSet(byte characterSet)

参数：

characterSet：编码代号（具体查找表 1-1）

例子：

Print.SetCharacterSet(byte characterSet)

设置中文：

Print.LanguageEncode=”gb2312”

Print.SetCharacterSet(0)

设置英文：

Print.LanguageEncode=”iso8859-1”

Print.SetCharacterSet(0)

设置繁体：

Print.LanguageEncode=”big5”

Print.SetCharacterSet(0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.9 对齐方式

int SetJustification(int justification)

参数：

justification：对齐方式

0：左对齐。

1：居中。

2：右对齐。

例子：

Print.SetJustification(int justification)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.10 获取打印机状态

1）函数：

GetRealTimeStatus(byte realTimeItem)

参数：

realTimeItem：

1：传输打印机状态。

2：传输打印机状态。

3：传输打印机状态。

4：传输纸张状态。

返回：

statusData：返回的状态（如下表），长度为 1。

realTimeItem=1：

位	0/1	HEX	Decimal	功能
0	0	00	0	固定为 0

1	1	02	2	固定为 1
2	0	00	0	钱箱 3 引脚为低电平
	1	04	4	钱箱 3 引脚为高电平
3	0	00	0	联机
	1	08	8	脱机
4	1	10	16	固定为 1
5	0	00	0	不等待在线恢复
	1	20	32	等待在线恢复
6	0	00	0	没有按下走纸键
	1	40	64	按下走纸键
7	0	00	0	固定为 0

realTimeItem=2：

位	0/1	HEX	Decimal	功能
0	0	00	0	固定为 0
1	0	00	0	固定为 0
2	0	00	0	固定为 0
3	0	00	0	固定为 0
4	0	00	0	固定为 0
5	0	00	0	固定为 0
6	0	00	0	打印机状态正常
7	1	40	64	打印机状态异常
	0	00	0	固定为 0

realTimeItem=3：

位	0/1	HEX	Decimal	功能
0	0	00	0	固定为 0
1	0	00	0	固定为 0
2	0	00	0	固定为 0
3	0	00	0	固定为 0
4	0	00	0	固定为 0
5	0	00	0	上盖关
	1	20	00	上盖开
6	0	00	0	打印头温度正常
	1	40	64	打印头温度异常
7	0	00	0	固定为 0

realTimeItem=4：

位	1/0	HEX	Decimal	功能
0	0	00	0	固定为 0
1	0	00	0	固定为 0
2,3	0	00	0	纸存在传感器检测到无纸
	1	0C	12	纸存在传感器检测到有纸
4	0	00	0	固定为 0
5,6	0	00	0	有纸
	1	60	96	纸尽
7	0	00	0	固定为 0

statusData.length==0 获取失败

例子：

```
byte statusData  =
Print.GetRealTimeStatus((byte)4);
```

2）函数：（适用于串口读取）

Print.GetTransmitStatus(int transmitItem)

参数：

transmitItem：

1：获取纸张状态。

2：获取钱箱状态。

返回：

statusData：

返回的状态（如下表），长度为 1。

查询纸张：

位	关/开	十六进制	十进制	状态
0, 1	关	00	0	纸将尽传感器：纸充足
	开	03	3	纸将尽传感器：纸将尽
2, 3	关	00	0	纸尽传感器：纸存在
	开	0c	12	纸尽传感器：纸不存在
4	关	00	0	固定
5, 6	--	--	--	保留
7	关	00	0	固定

查询钱箱：

位	关/开	十六进制	十进制	状态
0	关	00	0	钱箱插座引脚3信号为低
	开	01	1	钱箱插座引脚3信号为高
1~3	--	--	--	保留
4	关	00	0	固定
5,6	--	--	--	保留
7	关	00	0	固定

例子：

```
byte statusData = Print.GetTransmitStatus(1);
```

3.11 初始化打印机

int Initialize()

注意：

将打印机还原成开机时的状态。

参数：

无

例子：

Print.Initialize()

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.12 设置打印机浓度

int SetPrintDensity(byte density)

参数：

density：浓度（根据打印机机型而定）

函数：

Print.SetPrintDensity(0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.13 设置打印机速度

int SetPrintSpeed(byte speed)

参数：

speed：速度。（根据打印机机型而定）

函数：

Print.SetPrintSpeed(1)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.14 切纸

int CutPaper(int cutMode)

参数：

cutMode：切刀模式。

0：全切。

1：半切。

例子：

Print.CutPaper(1)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.15 钱箱

int OpenCashdrawer(int openMode)

参数：

openMode：

0：打开 1 号钱箱。

1：打开 2 号钱箱。

2：2 个钱箱都打开。

例子：

Print.OpenCashdrawer(2)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.16 蜂鸣器

int BeepBuzzer(byte times, byte t1, byte t2)

参数：

times： 响的次数。

t1： 响的时间（t1× 100ms）。

t2： 停止的时间（t2× 100ms）。

函数：

Print.BeepBuzzer(byte times,byte t1,byte t2)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.17 打印文本

1）函数：

int PrintText(String data)

参数：

data： 文本内容。

例子：

Print.PrintText(“TEXT\n”)

2）函数：

int PrintText(String data, int alignment, int attribute,
int textSize)

参数：

data：文本内容。

alignment：对齐方式。

0：左对齐。

1：居中。

2：右对齐。

attribute：样式。

0：大字体，不加粗，不加下划线，不加反白。

1：小字体，不加粗，不加下划线，不加反白。

2：大字体，加粗，不加下划线，不加反白。

3：小字体，加粗，不加下划线，不加反白。

4：大字体，不加粗，加下划线，不加反白。

5：小字体，不加粗，加下划线，不加反白。

6：大字体，加粗，加下划线，不加反白。

7：小字体，加粗，加下划线，不加反白。

8：大字体，不加粗，不加下划线，加反白。

9：小字体，不加粗，不加下划线，加反白。

10：大字体，加粗，不加下划线，加反白。

11：小字体，加粗，不加下划线，加反白。

12：大字体，不加粗，加下划线，加反白。

13：小字体，不加粗，加下划线，加反白。

14：大字体，加粗，加下划线，加反白。

15：小字体，加粗，加下划线，加反白。

textSize：字体放大倍数。

[范围]：textSize=（0 到 7, 16 到 23, 32，39, 48 到 55, 64 到 71, 80 到 87, 96 到 103, 112 到 119；）

字体高的放大倍数=textSize%8；

字体宽的放大倍数=textSize/8;

例子：

Print.PrintText(“TEXT\n”,0,14,0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.18 打印条码

1)int PrintBarCode(int bcType, String bcData)

参数：

bcType：条码类型。

m	Bar code system	Range of *n*	Range of *d*
65	UPC-A	$n = 11, 12$	$48 \le d \le 57$
66	UPC-E	$n = 11, 12$	$48 \le d \le 57$ [where $d1 = 48$]
67	JAN13 / EAN13	$n = 12, 13$	$48 \le d \le 57$
68	JAN8 / EAN8	$n = 7, 8$	$48 \le d \le 57$
69	CODE39	$1 \le n \le 255$	$48 \le d \le 57, 65 \le d \le 90,$ $d = 32, 36, 37, 42, 43, 45, 46, 47$
70	ITF	$2 \le n \le 254$ (even number)	$48 \le d \le 57$
71	CODABAR (NW-7)	$2 \le n \le 255$	$48 \le d \le 57, 65 \le d \le 68,$ $97 \le d \le 100,$ $d = 36, 43, 45, 46, 47, 58$ [where $65 \le d1 \le 68, 65 \le dn \le 68,$ $97 \le d1 \le 100, 97 \le dn \le 100$]
72	CODE93	$1 \le n \le 255$	$0 \le d \le 127$
73	CODE128	$2 \le n \le 255$	$0 \le d \le 127$ [where $d1 = 123, 65 \le d2 \le 67$]

n 表示条码数据字节数。

d 指定条形码数据 。

bcData：条码内容。

2)int PrintBarCode(int bcType,String bcData,int width,int height,int HRIPosition, int justification)

参数：

bcType：条码类型（同上）。

bcData：条码内容。

width：条码宽度。范围： (1-6)

	宽度（mm）	窄条码 (mm)	宽条码 (mm)
1	0.125	0.125	0.250
2	0.25	0.25	0.625
3	0.375	0.375	2.303
4	0.5	0.5	1.250
5	0.625	0.625	1.625
6	0.750	0.750	2

height：条码高度。范围：1-255。

HRIPosition：打印条形码时选择 HRI 字符的打印位置 。

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	在条形码上方及下方

justification：对齐方式。

0：左对齐。

1：居中。

2：右对齐。

例子：

```
Print.PrintBarCode(73,"{BS/N:{C\014\042\070\116
                        {A3",1,50,2,0);
//这个是打印 code128 内容是 S/N:123456783
```

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.19 打印二维码

1)int PrintQRCode(String bcData)

参数：

bcData:二维码内容。

2)int PrintQRCode(String bcData,int sizeOfModule,int errorLevel,int justification)

参数：

bcData：二维码内容。

sizeOfModule：二维码大小。范围 1-16；

errorLevel：纠错等级。

N	功能	参考：可恢复字码比例
48	选择纠错等级 L	7%
49	选择纠错等级 M	15%
50	选择纠错等级 Q	25%
51	选择纠错等级 R	30%

justification：对齐方式。

0：左对齐。

1：居中。

2：右对齐。

例子：

Print.PrintQRCode(“二维码内容”,6,48,0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.20 打印图片

函数：

PrintBitmap(Bitmap bmp,int halftoneType,

int luminance)

参数：

bmp：图片对象。

halftoneType：图片的算法类型。

0：黑白。

1：抖动。

2：聚集。

luminance：亮度。（范围: -100 到 100）

例子：

Print.PrintBitmap(bmp,1,0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.21 向打印机发送数据

int WriteData(byte[] bData)

参数：

bData：向打印机写入的数据。

例子：

Print.WriteData(“123abc\n”.getBytes("GB2312"))

//向打印机发送文本是 123abc 的数据给打印机。

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.22 从打印机读数据

byte[] ReadData(int time)

参数：

time：超时时间（单位 毫秒）。

例子：

Print.ReadData(2000)

//读取打印机的数据。

返回：

从打印机返回的数据，length 等于 0 打印机无数据返回。

3.23 打印 PDF417

函数：

```
int PrintPDF417(String bcData,
                byte dataColumns,
                byte dataRows,
                byte moduleWidth,
                byte rowHeight,
                byte errorMode,
                byte errorLevel,
                byte options)
```

参数：

bcData：数据内容。

dataColumns：设置打印数据区域的列数（范围：0-30）。0：自动处理。根据打印范围来确定打印列数。

dataRows：设置 PDF417 的行数（范围：0，3-90）。0：自动处理。根据打印范围来确定打印行数。

moduleWidth：设置模块宽度（范围：2-8）。

rowHeight：设置模块高度=n*宽度（范围：2-n-8）。

errorMode：纠错模式。

48：等级模式。

49：比率模式。

errorLevel：根据纠错模式分为两种（n）。

等级模式：

n	功能	纠错码字数量
48	选择纠错等级 0	2
49	选择纠错等级 1	4
50	选择纠错等级 2	8
51	选择纠错等级 3	16
52	选择纠错等级 4	32

53	选择纠错等级 5	64
54	选择纠错等级 6	128
55	选择纠错等级 7	256
56	选择纠错等级 8	512

比率模式：［数据码字×n ×0.1＝（A）］（小数部分四省五入）

A	功能	纠错码字数量
0~3	选择纠错等级 1	4
4~10	选择纠错等级 2	8
11~20	选择纠错等级 3	16
21~45	选择纠错等级 4	32
46~100	选择纠错等级 5	64

101~200	选择纠错等级 6	128
201~400	选择纠错等级 7	256
400 以上	选择纠错等级 8	512

options：选择可选项。

0：选择标准 PDF417

1：选择压缩 PDF417

返回：

≠-1：发送给打印机成功。

＝-1：发送失败。

例子：

Print.PrintPDF417(“123456”,(byte)0,(byte)0,(byte)3,(byte)3,(byte)49,(byte)1,(byte)0)

3.24 标签定位

int GotoNextLabel()

注意：

该指令只用于标签的定位，连续纸不可用。

例子：

Print.GotoNextLabel()

//定位到标签纸缝标。

返回：

≠-1：发送给打印机成功。

＝-1：发送失败。

3.25 页模式

int SelectPageMode()

注释：

必须是打印机支持页模式功能才能进入。在页模式下你可以设置你想要打印的区域，坐标，方向。

参数：

无

例子：

```
Print.SelectPageMode()
//进入页模式。
Print.SelectPageMode()
//设置打印区域。
Print.SetPageModePrintArea(0,0,200,200)
//设置打印方向
Print.SetPageModePrintDirection(0)
//设置 X,Y 的坐标。
```

```
Print.SetPageModeAbsolutePosition(0,0)
//打印二维码（你也可以打印文字和条码）。
Print.PrintQRCode("abcdef",4,48,1)
//打印。
Print.PrintDataInPageMode()
```

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.26 设置打印区域（页模式下）

int SetPageModePrintArea(int horizontal, int vertical, int width, int height)

注释：

必须是打印机支持页模式功能。并且已经进入页模式才会生效。

参数：

horizontal:起始点的横坐标。

vertical：起始点的纵坐标。

width：区域的宽度。

height：区域的高度。

例子：

Print.SetPageModePrintArea(0,0,200,200)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.27 设置打印方向（页模式下）

int SetPageModePrintDirection(int direction)

注释：

必须是打印机支持页模式功能并且已经进入页模式后才会生效。

参数：

direction:打印方向。

0：0 度。

1：90 度。

2：180 度。

3：270 度。

例子：

Print.SetPageModePrintDirection(0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.28 设置打印坐标（页模式下）

int SetPageModeAbsolutePosition(int xPosition,
int yPosition)

注释：

必须是打印机支持页模式功能并且已经进入页模式后才会生效。

参数：

xPosition：X 坐标。

yPosition：Y 坐标。

例子：

Print.SetPageModeAbsolutePosition(0,0)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.29 打印（页模式下）

int PrintDataInPageMode()

注释：

必须是打印机支持页模式功能并且已经进入页模式后才会生效。

参数：

无

例子：

Print.PrintDataInPageMode()

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.30 获取 NV 位图列表

int RefreshImageList(List<byte[]> lbImageIndex)

注释：

必须是打印机支持 NV 位图功能才会生效。

参数：

lbImageIndex：图片列表序列号。

例子：

Print.RefreshImageList(lbImageIndex)

返回：

-1：打印机不支持 NV 位图功能。

1：获取列表成功。

3.31 获取 NV 位图总内存大小

int QueryNVStoreCapacity(int[] iSpace)

注释：

必须是打印机支持 NV 位图功能才会生效。

参数：

iSpace：内存大小。

例子：

```
iSpace=new int[1];
Print.QueryNVStoreCapacity(iSpace);
```

返回：

-1：打印机不支持 NV 位图功能。

1：获取成功。

3.32 获取 NV 位图剩余内存大小

int QueryNVStoreRemainingCapacity(int[] rEpaces)

注释：

必须是打印机支持 NV 位图功能才会生效。

参数：

rEpaces：剩余内存大小。

例子：

```
storeRemainingCapacity=new int[1];
Print.QueryNVStoreRemainingCapacity(storeRemainingCapacity);
```

返回：

-1：打印机不支持 NV 位图功能。

1：获取成功。

3.33 打印 NV 位图

int PrintNVImage(String imageNo,int scaleMode)

注释：

必须是打印机支持 NV 位图功能才会生效。

参数：

imageNo：图片序列号。

scaleMode：模式（默认 0）。

例子：

```
Print.PrintNVImage(imageNo,0);
```

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.34 删除指定 NV 位图

int DeleteSpecifiedNVImage(String sImageIndex)

注释：

必须是打印机支持 NV 位图功能才会生效。

参数：

sImageIndex：图片序列号。

例子：

Print.DeleteSpecifiedNVImage(sImageIndex)

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.35 删除所有 NV 位图

int DeleteAllNVImage()

注释：

必须是打印机支持 NV 位图功能才会生效。

例子：

Print.DeleteAllNVImage();

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.36 下载图片到打印机

int DefineNVImage(String[] sArrFile, Handler handler)

注释：

必须是打印机支持 NV 位图功能才会生效。

参数：

sArrFile：图片路径。

handler：Handler 的对象。

message.what：最大包数。

mesage.arg1：下载进度。

例子：

Print.DefineNVImage(sArrFile,handler);

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.37 打印 Bin 文件

boolean PrintBinaryFile(String strPRNFile)

参数：

strPRNFile：bin 文件路径。

例子：

Print.PrintBinaryFile(strPRNFile);

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.38 获取打印机功能列表

int CapturePrinterFunction(

int ModelPropertyKeyBeep,

int[] propType,

byte[] value,

int[] dataLen)

参数：

ModelPropertyKeyBeep：功能代号。

MODEL_PROPERTY_KEY_BEEP：蜂鸣器。

MODEL_PROPERTY_KEY_CUT：切纸。

MODEL_PROPERTY_KEY_DRAWER：钱箱。

MODEL_PROPERTY_KEY_BARCODE：

条码。

MODEL_PROPERTY_KEY_PAGEMODE：

页模式。

MODEL_PROPERTY_KEY_GET_REMAININ

G_POWER：电源。

MODEL_PROPERTY_CONNECT_TYPE：连接方式。

MODEL_PROPERTY_KEY_PRINT_RECEIPT：小票。

propType：种类编号。

Value：是否支持。

（蜂鸣器、切纸、钱箱、页模式、电源、小票）

Value[0]==0 支持。否则不支持。

（条码）

String barcode =new String(Value);

barcode 包含 QRCODE 支持二维码

barcode 包含 PDF417 支持 PDF417

dataLen：返回数据的长度。

例子：

int[] propType=new int[1];

byte[] Value=new byte[500];

int[] DataLen=new int[1];

Print.CapturePrinterFunction(ModelPropertyKey Beep,propType,Value,DataLen);

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.39 清除页模式缓存区数据

int ClearPageModePrintAreaData()

参数：

无

例子：

Print.ClearPageModePrintAreaData();

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.40 打印并返回行模式

int PrintAndReturnStandardMode()

参数：

无

例子：

Print.PrintAndReturnStandardMode();

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.41 设置左边距

int SetLeftMargin(int iLeftMargin)

参数：

iLeftMargin：左边距（单位 px）

例子：

Print.PrintAndReturnStandardMode();

返回：

≠-1：发送给打印机成功。

-1：发送失败。

3.42 读取磁卡信息

void setTrackCardReaderMode(int track,CardReader cardReader,int outTime)

注意：

拥有刷磁卡功能的打印机才支持该功能。

参数：

track：磁道（范围：1-5）。

CardReader：数据返回接口。

Succeed(byte[] data);//返回的数据。

Failure(int error);

//error--> 1:连接断开,2：超时,3:其他错误。

outTime：超时时间（单位：毫秒）。

例子：

```
Print.setTrackCardReaderMode(track,new
Print.CardReader() {
  @Override
  public void Succeed(final byte[] data) {}
```

```
@Override
public void Failure(final int error) {}
  },30*1000);
```

3.43 退出磁卡模式

boolean CancelTrackCardReaderMode()

参数：

无

例子：

```
Print.CancelTrackCardReaderMode();
```

返回：

true：成功。

false：失败。

3.44 设置移动单元

int setPrintResolution(int x,int y)

参数：

该接口是设置 x 轴和 y 轴的移动单元。

单位：25.4/ x　mm，25.4/ y　mm。

范围：（0-255）。

返回：

>0：发送给打印机成功。

-1：发送失败。

-2：参数错误。

例子：

```
Print.setPrintResolution(203,203);
```

3.45 打印矩形框

int PrintPageRectangle(int x,int y,int width,

int height,int lineWidth)

注意：

该接口只有部分打印机支持。

参数：

x：左上角 x 坐标。(单位：PX)

y: 左上角 y 坐标。(单位：PX)

width：矩形框的宽度。

height：矩形框的高度。

lineWidth：线条宽度。

返回：

>0：发送给打印机成功。

-1：发送失败。

例子：

Print.PrintPageRectangle(0,0,100,100,2);

3.46 打印线条

int PrintPageLine(int x1,int y1,int x2,int y2,

int lineWidth)

注意：

该接口只有部分打印机支持。单位：PX。

参数：

x1：起始 x 坐标。

y1：起始 y 坐标。

x2：终点 x 坐标。

y2：终点 y 坐标。

lineWidth：线条宽度。

返回：

大于 0：发送给打印机成功。

-1：发送失败。

例子：

Print.PrintPageLine(0,0,100,100,2);

3.47 图片数据压缩打印

int PrintBitmapLZO(Bitmap bitmap,int halftoneType)

注意：

该接口只有部分打印机支持，且打印机打印的是图片原始大小。8 px=1 mm。

参数：

bitmap：图片对象。

halftoneType：图片的算法类型。

0：二值算法。

1：半色调算法。

返回：

大于 0：发送给打印机成功。

-1：发送失败。

例子：

```
Print.PrintBitmapLZO(bitmap,0);
```

3.48 获取打印机 SN

String getPrintSN()

注意：

该接口只有部分打印机支持。

参数：

无

返回：

大于 0：发送给打印机成功。

-1：发送失败。

例子：

```
Print.getPrintSN();
```

3.49 获取打印机电量

Int getPrinterQuantity()

注意：

该接口只有部分打印机支持。

参数：

无

返回：

-1：发送失败。

-2：打印机不支持

其他：电池电量

例子：

```
Print.getPrinterQuantity();
```

3.50 获取打印机 IP 地址

String getPrintIP()

注意：

该接口只有部分打印机支持。

参数：

无

返回：

IP 地址

例子：

```
Print.getPrintIP();
```

3.51 获取 DHCP 状态

boolean isDHCP ()

注意：

该接口只有部分打印机支持。

参数：

无

返回：

true ： DHCP 开启

false ： DHCP 关闭

例子：

```
Print.isDHCP();
```

3.52 获取打印机子网掩码

String getPrintSubnetMask()

注意：

该接口只有部分打印机支持。

参数：

无

返回：

子网掩码

例子：

```
Print.getPrintSubnetMask();
```

3.53 获取打印机网关

String getPrintGateway()

注意：

该接口只有部分打印机支持。

参数：

无

返回：

网关

例子：

```
Print.getPrintGateway();
```

3.54 设置打印机 IP 地址

int setPrintIP(String ip)

注意：

该接口只有部分打印机支持。

参数：

ip:需要设置的 IP 地址。

返回：

-1：发送失败，0：发送成功

例子：

```
Print.setPrintIP(“192.168.1.1”);
Print.setPrintSave();//保存
```

3.55 设置 DHCP 状态

int setDHCP(int dhcp)

注意：

该接口只有部分打印机支持。

参数：

dhcp:

0：关闭

1：开启

返回：

-1：发送失败，0：发送成功

例子：

```
Print.setDHCP(“1”);
Print.setPrintSave();//保存
```

3.56 设置打印机子网掩码

int setPrintSubnetMask(String subnetMask)

注意：

该接口只有部分打印机支持。

参数：

subnetMask：子网掩码

返回：

-1：发送失败，0：发送成功

例子：

```
Print.setPrintSubnetMask(“255.255.255.0”);
Print.setPrintSave();//保存
```

3.57 设置打印机网关

int setPrintGateway(String gateway)

注意：

该接口只有部分打印机支持。

参数：

gateway：网关

返回：

-1：发送失败，0：发送成功

例子：

```
Print.setPrintGateway(“192.168.1.1”);
Print.setPrintSave();//保存
```

3.58 设置保存

int setPrintSave()

注意：

该接口只有部分打印机支持。

参数：

无

返回：

-1：发送失败，0：发送成功

例子：

```
Print.setPrintSave();//保存
```

3.59 打印表格

void printTable(Table table)

参数：

table：表格实体类。

public Table(String column, String regular, int[] columnWidth)

column 为以参数 regular 分隔的表头。形如″序号,单价,数量,金额″

regular 为表内字符串的分隔符。如上面的是”,”

columnWidth 为表格每一列的字符宽度。默认字体大小的计算方法是中文 2 个字符，英文 1 个字符，然后相加，如”序号”的宽度为 4 个字符。

public void addRow(String row)添加一行数据

参数说明：row 一行数据

数据格式与表头格式一致。若某一单元格的数据超出限定的字符宽度，会自动换行打印，若需要手动换行，可在需要换行处加″\n″。

返回：

无

例子：

```
String column = "品名;数量;单价;金额";
Table table = new Table(column, ";", new int[]
{ 14, 6, 6, 6 });
table.addRow("保鲜袋"+ ";10.00;1;10.00");
table.addRow("铁丝挂钩" + ";5.00;2;10.00");
table.addRow("雨伞"+ ";5.00;3;15.00");
Print.printTable(table);
```

表 1-1

名称	characterSet	codepage
Default	0	gb2312
Chinese Simplified	0	gb2312
Chinese Traditional	0	big5
PC437(USA)	0	iso8859-1
KataKana	1	Shift_JIS
PC850(Multilingual)	2	iso8859-3
PC860(Portuguese)	3	iso8859-6
PC863(Canadian-French)	4	iso8859-1
PC865(Nordic)	5	iso8859-1

PC857(Turkish)	13	IBM857
PC737(Greek)	14	iso8859-7
ISO8859-7(Greek)	15	iso8859-7
WCP1252	16	iso8859-1
PC866(Cyrillic #2)	17	iso8859-5
PC852(Latin 2)	18	iso8859-2
PC858(Euro)	19	iso8859-15
KU42	20	ISO8859-11
TIS11(Thai)	21	ISO8859-11
TIS18(Thai)	26	ISO8859-11
PC720	32	iso8859-6
WPC775	33	iso8859-1

PC855(Cyrillic)	33	iso8859-5
PC862(Hebrew)	36	iso8859-8
PC864(Arabic)	37	iso8859-6
ISO8859-2(Latin2)	39	iso8859-2
ISO8859-15(Latin9)	40	iso8859-15
WPC1250	45	iso8859-2
WPC1251(Cyrillic)	46	iso8859-5
WPC1253	47	iso8859-7
WPC1254	48	iso8859-3
WPC1255	49	iso8859-8
WPC1256	50	Windows-1256
WPC1257	51	iso8859-1

WPC1258	52	bg2312
MIK(Cyrillic/Bulgarian)	54	iso8859-15
CP755	55	iso8859-5
Iran	56	iso8859-6
Iran II	57	iso8859-6
Latvian	58	iso8859-4
ISO-8859-1(West Europe)	59	iso8859-1
ISO-8859-3(Latin 3)	60	iso8859-3
ISO-8859-4(Baltic)	61	iso8859-4
ISO-8859-5(Cyrillic)	62	iso8859-5
ISO-8859-6(Arabic)	63	iso8859-6

ISO-8859-8(Hebrew)	**64**	**iso8859-8**
ISO-8859-9(Turkish)	**65**	**iso8859-9**
PC856	**66**	**iso8859-8**
ABICOIM	**67**	**iso8859-15**