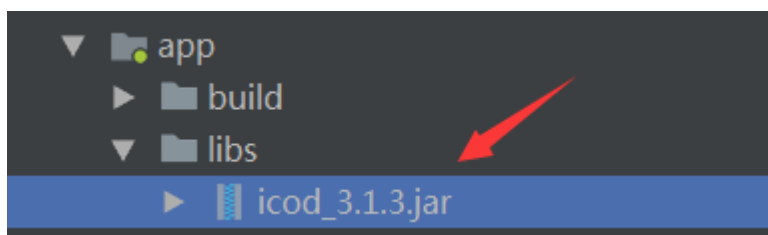


icod SDK 使用手册

版本: v2. 1. 1

一、集成 SDK

1. 将 sdk 的 jar 拷贝到 libs 文件下.点击 jar 右键找到 as a library



2. 将你需要的对应系统类型.so 文件拷贝到 jniLibs 文件



二、参数配置

在 AndroidManifest.xml 中添加权限（android 6.0 以上请添加动态写入文件权限）：

```
<uses-feature android:name="android.hardware.usb.host" />
<!-- WIFIManage permission -->
<!-- Allows applications to access information about networks -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<!-- Allows applications to access information about Wi-Fi networks -->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<!-- Allows applications to change network connectivity state -->
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
```

```

<!-- Allows applications to change network connectivity state -->
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<!-- Allows applications to enter Wi-Fi Multicast mode -->
<uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<!-- Internet permission -->
<uses-permission android:name="android.permission.INTERNET" >
</uses-permission>
<!-- SDCard permission -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<!-- If your application uses Wi-Fi, declare so with a <uses-feature> element in the manifest file
-->
<uses-feature android:name="android.hardware.wifi" />
<!-- Bluetooth permission -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

```

三、简单的初始化

```

Utils.init( context: this);
// 增加日志文件，true:开启日志输出
// Add log file, true: turns log output on
PrinterAPI.getInstance().setOutput(true);
// 添加终端日志
// Add terminal log
Logger.addLogStrategy(new AndroidLogCatStrategy());

```

四、函数调用

0、返回值声明

- 0: 为成功
- 1: 为失败
- 2: 参数错误
- 3: 为字符长度不支持
- 4: 字符不支持（后面两个是用于条码打印的）

1、连接打印机

函数: `connect(io)`

| 连接方式 | InterfaceAPI io | 参数说明 |
|-----------|--|--|
| USB | <code>new USBAPI(Context context)</code> | context: 上下文 |
| Bluetooth | <code>new BluetoothAPI(Context context)</code> | context: 上下文 |
| Serial | <code>new SerialAPI(File device, int baudRate, int flowControl)</code> | device: 串口 baudRate: 波特率 flowControl: 是否开流控 (0: 关闭 1:开启) |
| WIFI | <code>new SocketAPI(String site, int port)</code> | Site: ip 地址 Port: 端口号 |
| USBNative | <code>new UsbNativeAPI()</code> | |

2、关闭连接和释放资源

函数: `disconnect()`

参数: 无

3、打印机连接状态

函数: `isConnect ()`

参数: 无

返回: `true` 已连接 `false` 未连接

4、写入指令 发送自定义指令

函数: `sendOrder(byte[] cmd)`

参数: `cmd` - 指令数组

函数: `sendOrder(byte[] cmd, int timeOut)`

参数: `cmd` - 指令数组

`timeOut` - 超时时间 单位 ms

函数: hexMsg (String msg, boolean isHex)

参数: msg - 指令字符串, 格式是##0a 23 0d##

isHex - true 数据是十六进制 false 是十进制

5、读取指令

函数: readI0(byte[] readBuffer,

int offsetSize,

int readSize,

int waitTime)

参数: readBuffer - 指令字节数组

offsetSize - 偏移量

readSize - 读取大小

waitTime - 等待时间

6、初始化

函数: init()

参数: 无

说明: 这个指令 用于清空缓冲区和恢复默认值 一般在打印图片前或者后调用, 避免打不全对后面数据的影响 可以跟 initAllPrinter 方法一起用

函数: initAllPrinter(int n)

参数: n - 1. 错误恢复从错误行开始打印 2. 错误恢复和清除缓冲内容

7、写入指令

函数: writeI0(byte[] writeBuffer,

int offsetSize,

int writeSize,

int waitTime)

参数: writeBuffer - 指令字节数组

offsetSize - 偏移量, 一般为 0

writeSize - 数组大小

waitTime - 等待时间

8、打印自检页

函数: selfTestPage ()

参数: 无

9、执行测试打印

函数: doTestPrint (int n, int m)

参数: n - 0 为 basic sheet; 1 2 为卷纸

m - 1 让打印机进入十六进制; 2 打印机状态打印; 3 卷纸模式打印

10、设置日志输出文件标志

函数: setOutput(boolean output)

参数: output - true. 开启日志输出

11、实时状态传送

函数: getStatus()

参数: 无

返回: 打印机实时状态数组, 可用 parseStatus 函数解析

函数: getPrinterStatus (int n)

参数: n - n = 1, 49

返回: 12 为缺纸状态 3 为纸将尽状态

说明: 串口连接时传送状态

12、解析打印机状态

函数: parseStatus(byte[] mCmd)

参数: mCmd - 状态数据

13、打印字符串

函数: `printString(String text)`

参数: `text` - 打印的字符串

函数: `printString(String text,
String charsetName,
boolean isFeed)`

参数: `text` - 打印的字符串

`charsetName` - 字符集

`isFeed` - 是否 换行

14、打印光栅位图

函数: `printRasterBitmap(Bitmap bitmap)`

参数: `bitmap` - 位图

函数: `printRasterBitmap(Bitmap bitmap, boolean toGray)`

参数: `bitmap` - 位图

`toGray` - true 变灰

函数: `printRasterBitmap(Bitmap bitmap,
boolean toGray,
int timeOut)`

参数: `bitmap` - 位图

`toGray` - true 变灰

`timeOut` - 超时时间

函数: `printRasterBitmap(Bitmap bitmap,
boolean toGray,
int timeOut,
boolean check)`

参数: `bitmap` - 位图

`toGray` - true 变灰

`timeOut` - 超时时间

check - 是否需要检测 数据是否已经全部下发到打印机. 解决第一张打印正常第二张乱码的问题.

函数: printRasterBitmap(Bitmap bitmap,
boolean toGray,
int timeOut,
boolean isJini,
boolean check)

参数: bitmap- 位图

toGray - true 变灰

timeOut - 超时时间

isJini - 是否使用本地方法

check - 是否需要检测 数据是否已经全部下发到打印机. 解决第一张打印正常第二张乱码的问题.

函数: printImageForPin (Bitmap bitmap)

参数: bitmap- 位图

说明: 针式打印图片

15、设置条形码高度

函数: setBarCodeHeight (int n)

参数: n - [0-255], 默认为 162

16、设置条形码宽度

函数: setBarCodeWidth (int n)

参数: n - [2-16] 我方打印机范围为 2-6 通用的打印机 2-16 所以有些值不管用

17、打印条形码

函数: printBarCode (int m, int n, String barcode)

参数: m - 使用条码系统类型

n - 使用条码系统类型

barcode - 字符串

18、打印二维码

函数: `printQRCode(String text,`
`int modeSize,`
`boolean isCut)`

参数: `text` - 为二维码内容
 `modeSize` - 设置条码大小 (1FAIL6) 默认为 6
 `isCut` - 打印二维码后是否切纸

19、PDF 条码打印

函数: `printPDFCode(String text,`
`int errorSize,`
`int hSize,`
`int vSize)`

参数: `text` - 为 PDF 条码内容
 `errorSize` - 纠错等级, 可控制条码大小建议 4-8
 `hSize` - 宽度 0-30 (建议 0)
 `vSize` - 高度 3-90 (建议 0)

20、打印并换行

函数: `printFeed()`

参数: 无

21、打印并退纸

函数: `printBackFlow(int n)`

参数: `n` - 退纸 `n/144` 英寸

22、打印并进纸

函数: `printAndFeedPaper (int n)`

参数: $n - n * 0.125$ 毫米

23、将标记打印纸进纸到打印起始位置

函数: feedToStartPos ()

参数: 无

24、切纸

函数: fullCut ()

参数: 无

说明: 全切

函数: halfCut ()

参数: 无

说明: 半切

函数: cutPaper (int m, int n)

参数: $m - 66$ (保留一点不切)
 $n - 0$

说明: 选择裁纸模式

25、选择对齐方式

函数: setAlignMode (int type)

参数: type - 0 为左对齐, 1 为居中对齐, 2 为右对齐

26、选择标准模式

函数: standardMode ()

参数: 无

27、选择页模式

函数: pageMode ()

参数: 无

28、页模式下回标准模式

函数: `printAndBackToStd()`

参数: 无

29、设置相对当前打印位置 $(nl+nh*256)*0.125$ 毫米

函数: `setRelativePosition (int nl, int nh)`

参数: `nl` - 0-255

`nh` - 0-255

30、设置绝对(当前行打印的第一个位置)打印位置 $(nl+nh*256)*0.125$ 毫米

函数: `setAbsolutePosition(int nl, int nh)`

参数: `nl` - 0-255

`nh` - 0-255

31、设定左侧空白量 $(nl+nh*256)*0.125$ 毫米

函数: `setLeftMargin (int nl, int nh)`

参数: `nl` - 0-255

`nh` - 0-255

32、设置打印区域宽度

函数: `setPrnAreaWidth (int nL, int nH)`

参数: 宽度为 $(nL + nh*256)*0.125$ 毫米

82.5 毫米纸宽型号 缺省值 `nL=128,nh=2`

79.5 毫米纸宽型号 缺省值 `nL=64,nh=2`

60 毫米纸宽型号缺省值 `nL=192,nh=1`

58 毫米纸宽型号 缺省值 `nL=176,nh=1`

33、设置行间距

函数：setLineSpace (int n)
参数： n - n*0.125 毫米 缺省值为 30

34、设置缺省行间距

函数：setDefaultLineSpace ()
参数： 无

35、设置字符右间距

函数：setCharRightSpace (int n)
参数： n - n*0.125 毫米

36、设置 58mm

函数：set58mm()
参数：无

37、设置 80mm

函数：set80mm()
参数：无

38、选择打印模式

函数：setFontStyle(int type)
参数： type - 取值如下[0-255]

| 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 | 解除加重模式 |

| | | | | |
|---|---|----|-----|---------|
| | 开 | 08 | 8 | 设置加重模式 |
| 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 |

39、设定/解除粗体打印

函数：setEmphasizedMode (int n)

参数： n -最低位有效:1 为设定 0 位解除

40、设定/解除下划线

函数：setEnabledUnderLine (int enable)

参数： enable -最低位有效:1 为设定 0 位解除

41、设定/解除重叠打印

函数：setOverlapMode (int n)

参数： n -最低位有效:1 为设定 0 位解除

42、设置/解除平滑模式

函数：setEnabledSmoothPrn (int n)

参数： n - 最低位有效:1 为设定 0 位解除

43、设置/解除顺时针 90 ° 旋转

函数：setRotate(int n)

参数： n - 0,48 解除 1,49 设置

44、设置/解除颠倒

函数：setReverse(boolean reverse)

参数： reverse - true 设置颠倒 false 取消颠倒

45、激活/禁止面板按键

函数：setEnabledPanelButton (int n)

参数： n - 最低位有效:1 禁止 0 位激活

46、设置浓度

函数：setPrintColorSize (int n)

参数： n - 仅支持 1-4 浓度 一般字体加重的浓度

47、设置波特率

函数：changeBd (int bd)

参数： bd - 波特率

48、设置字体倍宽倍高的 设定字符大小

函数：setCharSize (int hsize, int vsize)

参数： hsize - 0-7(正常为 0)

vsize - 0-7(正常为 0)

49、选择国际字符集

函数：setInterCharSet (int n)

参数： n - [0-13]，默认为 0

| n | 字符集 |
|---|-----|
| 0 | 美国 |

| | |
|----|--------|
| 1 | 法国 |
| 2 | 德国 |
| 3 | 英国 |
| 4 | 丹麦 I |
| 5 | 瑞典 |
| 6 | 意大利 |
| 7 | 西班牙 I |
| 8 | 日本 |
| 9 | 挪威 |
| 10 | 丹麦 II |
| 11 | 西班牙 II |
| 12 | 拉丁美洲 |
| 13 | 韩国 |

50、选择字符代码表

函数：setCharCodeTable (int n)

参数： n – [0-5]、[16-19]、255

| | |
|-----|---------------|
| n | 页 |
| 0 | PC437[美国欧洲标准] |
| 1 | 片假名 |
| 2 | PC850[多语言] |
| 3 | PC860[葡萄牙语] |
| 4 | PC863[加拿大-法语] |
| 5 | PC865[北欧] |
| 16 | WPC1252 |
| 17 | PC866[斯拉夫语 2] |
| 18 | PC852[拉丁语 2] |
| 19 | PC858[欧洲] |
| 255 | Space Page |

51、选择页模式

函数: `pageMode ()`

参数: 无

52、页模式下回标准模式

函数: `printAndBackToStd()`

参数: 无

函数: `pagePrintAndBack2Standard()`

参数: 无

53、在页模式 设置相对当前打印位置 $(nl+nh*256)*0.125$ 毫米

函数: `setPageRelativePosition (int nl, int nh)`

参数: `nl` - 0-255

`nh` - 0-255

54、设置页模式打印区域 单位:mm

函数: `pageModeArea(int x, int y, int width, int height)`

参数: `x` - 打印水平起始位置

`y` - 打印垂直起始位置

`width` - 打印区域宽度

`height` - 打印区域高度

55、页模式下打印缓冲数据

函数: `pagePrint()`

参数: 无

56、页模式下取消打印数据

函数: `pageRemoveAllData ()`

参数：无

57、在页模式下选择打印方向

函数：pageSelectDirection (int n)

参数： n - 0 从左到右 1 从底到上 2 从右到左 3 从上到下

58、在页模式设置绝对打印位置 $(nl+nh*256)*0.125$ 毫米

函数：setAbsolutePosition(int nl, int nh)

参数： nl - 0-255

nh - 0-255

59、设定黑标偏移量

函数：markLengthSet(int printerType,
int ticketType,
int Q0,
int L0,
int mPrintLen,
int mTicketLen,
int mCutLen)

参数： printerType - 0 针式打印机 是乘 0.175 ;其他是 0.125

ticketType - 0 是打印首行在黑标之前;其他在黑标之后

L0 - 黑标传感器位置到撕纸线位置的距离

Q0 - 打印针头到黑标传感器位置的距离

mTicketLen - 票长

mPrintLen - 黑标到首行打印位置距离

mCutLen - 黑标到切纸位置的距离

60、送黑标纸至打印起始位置

函数：pointTest()

参数：无

61、是否为黑标状态

函数：pointTest()

参数：无

返回：true 当前是黑标状态

62、返回 switch 拨码

函数：getSwitch1Value ()

参数：无

返回：switch-1 拨码开关的 value;-1 时表示错误

函数：getSwitch2Value ()

参数：无

返回：switch-2 拨码开关的 value;-1 时表示错误

63、进入十六进制

函数：comeInHex ()

参数：无

64、选择打印纸传感器以输出缺纸信号

函数：setPaperSensor (int n)

参数： n - 8 传输缺纸状态, 2 传输纸将尽状态

65、选择打印纸传感器以停止打印

函数：setSensorToStopPrint (int n)

参数： n - 2 传送纸将尽

66、汉字模式

函数: chineseFontSet (int n)

参数: n - 0 为禁止 4 允许倍宽 8 允许倍高 128 允许下划线

67、小字体

函数: smallFontSizeSet ()

参数: 无

68、钱箱

函数: openCashDrawer(int n, int t1, int t2)

参数: n - 0 pin2 上输出 1 pin5 上输出

t1 - 开启时间 t1*2

t2 - 关闭时间 t2*2

69、NV 位图下载

函数: downLoadNvBitmap(Bitmap[] bitmaps)

参数: bitmaps - 位图数组

70、插除指令

函数: resetPrinter ()

参数: 无

说明: 这个指令用于插除打印机, 不建议经常调用。

71、设置黑标开关

函数: setBlackSwitch ()

参数: n - 0、关闭 1、开启

