

# Android SDK


## 开发指南

文档版本： V1.0

**版权所有©2023 深圳市度点科技有限公司保留一切版权。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明

 驰腾是深圳市度点科技有限公司的注册商标。

本文档提及的其它所有商标或注册商标，由各自的所有人拥有。

## 注意

由于产品版本升级或其它原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 概述

本文档用于指导用户在 Android 设备上，通过调用应用程序接口，轻松实现与驰腾打印机的交互。

Android 设备与打印机的连接方式：蓝牙、WiFi、USB。蓝牙、WiFi 是打印机的扩展功能，若打印机没有相应的模块则无法实现连接交互。

## 读者对象

本文档主要适用于以下工程师：

- 软件开发工程师
- 技术支持工程师

## 版本记录

第一次正式发布。

深圳市度点科技有限公司

地址：深圳市宝安区新安街道兴东社区 67 区中粮商务公园 3 栋 1101 邮编： 518100

网址： <https://ctaiot.com/>

客户服务电话： (0755) 2758 8185

客户服务传真： (0755) 2758 8185

客服邮箱： [xujie@ctaiot.com](mailto:xujie@ctaiot.com)

1 SDK 使用说明 .....	4
1.1 简介 .....	4
1.2 配置 SDK .....	4
1.3 兼容性 .....	5
2 打印机基础知识 .....	5
2.1 像素 .....	5
2.2 打印一张合格标签的基本流程 .....	5
3 API 函数说明 .....	6
3.1 连接打印机 .....	6
3.2 断开与打印机连接 .....	6
3.3 清除缓存 .....	6
3.4 执行打印 .....	7
3.5 查询硬件配置 .....	7
3.6 查询打印机型号 .....	8
3.7 查询高速打印支持状态 .....	8
3.8 查询固件信息 .....	8
3.9 查询纸张设置 .....	9
3.10 查询记忆打印参数设置 .....	9
3.11 查询浓度参数设置 .....	10
3.12 查询速度参数设置 .....	10
3.13 查询自动关机参数设置 .....	10
3.14 查询展示信息 .....	11
3.15 查询打印模式参数设置 .....	11
3.16 查询硬件版本 .....	12
3.17 查询当前电量 .....	12
3.18 设置纸张类型 .....	12
3.19 设置记忆打印参数状态 .....	13
3.20 设置自动关机时间 .....	13
3.21 恢复默认设置 .....	14
3.20 设置打印尺寸 .....	14
3.20 设置打印速度 .....	14
3.21 设置打印浓度 .....	15
3.21 设置打印方向 .....	15
3.22 绘制矩形 .....	16
3.22 绘制文本 .....	16
3.23 绘制条码 .....	17
3.24 绘制二维码 .....	17
3.25 绘制图片 .....	18
附录 返回值解释 .....	19

# 1 SDK 使用说明

## 1.1 简介



CT Android SDK 为驰腾打印机客户端提供了 API 调用服务，使 Android 应用开发者可以方便快捷地集成打印应用程序。

## 1.2 配置 SDK

导入 SDK

1. 将 ctaiotCtpl1.0.0.jar 文件复制到工程的 libs 文件夹中。
2. 导入 jar 包。

- Studio 开发

单击，弹出 Project Structure 对话框，选择需要依赖的工程，单击“Dependencies”；单击界面右侧的，选择“Jar dependency”，弹出 jar 包的路径选择对话框，选择“libs ”-> “ctaioCtpl1.0.0.jar ”，单击 OK。

- 添加权限

需在清单文件 AndroidManifest.xml 中配置以下权限：

```
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION"
    android:maxSdkVersion="30" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
    android:name="android.permission.ACCESS_BACKGROUND_LOCATION"
    android:maxSdkVersion="30" />
<uses-permission
    android:name="android.permission.BLUETOOTH"
    android:maxSdkVersion="30" />
<uses-permission
    android:name="android.permission.BLUETOOTH_ADMIN"
    android:maxSdkVersion="30" />

<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
```

ctaioCtpl1.0.0.jar 使用说明

单例对象：CTPL

实现类构造方法：

```
CTPL.getInstance().init(App.getInstance(), new RespCallback());
```

参数说明：

Application：上下文环境。

RespCallback：用于接收连接状态和参数返回值。

后续所有功能操作和结果反馈都通过 CTPL.getInstance() 完成

## 1.3 兼容性

CT Android SDK V1.0.0 版本支持 Android 8.0 及以上系统。

# 2 打印机基础知识

## 2.1 像素

像素 (pixel) 是打印机的最小打印成像单位，单位为 dots。1dot=1in÷打印机分辨率。打印机的分辨率与毫米的换算关系如下：

分辨率	像素点
203dpi	1 mm=203÷25.4≈8 dots
300dpi	1 mm=300÷25.4≈12 dots
600dpi	1 mm=600÷25.4≈24 dots

## 2.2 打印一张合格标签的基本流程

### 连接设备

```
CTPL.getInstance().init(Application, RespCallback);
```

2.2.1 连接设备，选择连接方式，如 SPP, BLE 等。然后等待通过 init() 时传入的 callback 通知连接成功

```
CTPL.getInstance().connect(new Device());
```

2.2.2 接收到回调后，设置参数，使用链式调用，并发送至打印机

```
CTPL.getInstance()  
    .setPrintSize(宽, 高)//设置纸张尺寸, 单位:毫米  
    .drawBitmap(new Rect(左, 上, 右, 下), bitmap, false, 150)//绘制图像,  
单位:像素  
    .print(打印份数)  
    .execute();//执行打印
```

## 3 API 函数说明

### 3.1 连接打印机

```
CTPL.getInstance().connect(new Device());
```

#### 【描述】

该函数的作用是使 Android 移动设备与打印机进行连接。

#### 【参数】

Device: 打印机的连接参数选项。

#### 【返回值】

无

#### 【范例】

详见 Demo。

### 3.2 断开与打印机连接

```
CTPL.getInstance().disconnect()
```

#### 【描述】

该函数的作用是使 Android 移动设备与打印机断开连接。

#### 【参数】

无

#### 【返回值】

无

#### 【范例】

详见 Demo。

### 3.3 清除缓存

```
CTPL.getInstance().clean()
```

**【描述】**

该函数的作用是清除缓存中未发送的指令数据。在发送一张新的标签内容到打印机前，建议先使用此函数清空打印机缓存中的数据内容。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.4 执行打印

`CTPL.getInstance().execute()`

**【描述】**

该函数的作用是将描述参数完成的打印指令发送到打印机执行

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.5 查询硬件配置

`CTPL.getInstance().queryHardwareConfig()`

**【描述】**

该函数的作用是批量查询硬件配置参数信息, 通过 `RespCallback` 返回, 返回的数据格式是若干条 `key-value`。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.6 查询打印机型号

`CTPL.getInstance().queryHardwareModel()`

### 【描述】

该函数的作用是查询打印机型号, 通过 `RespCallback` 返回, 返回的数据格式是一条 key-value。

### 【参数】

无

### 【返回值】

无

### 【范例】

详见 Demo。

## 3.7 查询高速打印支持状态

`CTPL.getInstance().queryCompressPrint()`

### 【描述】

该函数的作用是查询打印机是否支持高速打印点阵图, 通过 `RespCallback` 返回, 返回的数据格式是一条 key-value。返回值是“true”时, `drawBitmap()` 方法可以传递 true 参数提高打印速度

### 【参数】

无

### 【返回值】

无

### 【范例】

详见 Demo。

## 3.8 查询固件信息

`CTPL.getInstance().queryFirmwareInfo()`

### 【描述】

该函数的作用是查询打印机固件参数设置信息, 通过 `RespCallback` 返回, 返回的数据格式是若干条 key-value。



**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.9 查询纸张设置

`CTPL.getInstance().queryPaperType()`

**【描述】**

该函数的作用是查询打印机当前适用的纸张类型, 通过 `RespCallback` 返回, 返回的数据格式是一条 `key-value`。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.10 查询记忆打印参数设置

`CTPL.getInstance().queryMemoryPrint()`

**【描述】**

该函数的作用是查询打印机记忆打印功能的开关状态, 通过 `RespCallback` 返回, 返回的数据格式是一条 `key-value`。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.11 查询浓度参数设置

CTPL.*getInstance().queryDensity()*

**【描述】**

该函数的作用是查询打印机当前打印浓度的档位数值, 通过 RespCallback 返回, 返回的数据格式是一条 key-value。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.12 查询速度参数设置

CTPL.*getInstance().querySpeed()*

**【描述】**

该函数的作用是查询打印机当前打印速度的档位数值, 通过 RespCallback 返回, 返回的数据格式是一条 key-value。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.13 查询自动关机参数设置

CTPL.*getInstance().queryAutoShutdown()*

**【描述】**

该函数的作用是查询打印机当前自动关机设定的时间, 通过 RespCallback 返回, 返回的数据格式是一条 key-value, 单位:分钟, 数值为“0”时禁用自动关机

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.14 查询展示信息

`CTPL.getInstance().queryDisplayInfo()`

**【描述】**

该函数的作用是批量查询打印机用于展示的参数信息，例如 SN 码、当前电量等，通过 `RespCallback` 返回，返回的数据格式是若干条 key-value。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.15 查询打印模式参数设置

`CTPL.getInstance().queryPrintMode()`

**【描述】**

该函数的作用是查询打印机当前打印模式设置，通过 `RespCallback` 返回，返回的数据格式是一条 key-value。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.16 查询硬件版本

CTPL.*getInstance()*.queryHardwareVersion()

### 【描述】

该函数的作用是查询打印机硬件版本，通过 RespCallback 返回, 返回的数据格式是一条 key-value。

### 【参数】

无

### 【返回值】

无

### 【范例】

详见 Demo。

## 3.17 查询当前电量

CTPL.*getInstance()*.queryBattery()

### 【描述】

该函数的作用是查询便携打印机当前电量，桌面打印机返回的数值为 0，通过 RespCallback 返回, 返回的数据格式是一条 key-value。

### 【参数】

无

### 【返回值】

无

### 【范例】

详见 Demo。

## 3.18 设置纸张类型

CTPL.*getInstance()*.setPaperType(PaperType)

### 【描述】

该函数的作用是设置打印机打印纸张类型。

**【参数】**

PaperType: 枚举值 “Label” 设置为标签间隙纸  
枚举值 “Receipt” 设置为票据连续纸  
枚举值 “BlackMark” 设置为黑标纸

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.19 设置记忆打印参数状态

CTPL.getInstance().setMemoryPrint(boolean)

**【描述】**

该函数的作用是开启打印机的撕纸功能。

**【参数】**

true: 开启  
false: 关闭

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.20 设置自动关机时间

CTPL.getInstance().setAutoShutDown(int)

**【描述】**

该函数的作用是设置打印机的自动关机功能。

**【参数】**

0: 关闭自动关机功能

1~60: 设置自动关机时间, 单位: 分钟

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.21 恢复默认设置

`CTPL.getInstance().resetFirmware()`

**【描述】**

该函数的作用是将打印机的参数设置恢复到出厂状态。

**【参数】**

无

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.20 设置打印尺寸

`CTPL.getInstance().setSize(int width, int height)`

**【描述】**

该函数的作用是设置打印机的打印尺寸,需要在所有绘制指令之前设置

**【参数】**

width: 宽度,单位:毫米

height: 高度,单位:毫米

**【返回值】**

无

**【范例】**

详见 Demo。

### 3.20 设置打印速度

`CTPL.getInstance().setPrintSpeed(int speed)`

**【描述】**

该函数的作用是设置打印机的打印速度。

**【参数】**

speed: 1~8

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.21 设置打印浓度

`CTPL.getInstance().setPrintDensity(int density)`

**【描述】**

该函数的作用是设置打印机的打印浓度。

**【参数】**

density: 1~15

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.21 设置打印方向

`CTPL.getInstance().setOrientation(Direction)`

**【描述】**

该函数的作用是设置打印机的内容打印方向。

**【参数】**

Direction:     Degree0 设置 0° 正常打印方向  
                 Degree180 设置 180° 颠倒打印方向

**【返回值】**

无

**【范例】**

详见 Demo。

## 3.22 绘制矩形

CTPL.*getInstance()*.drawRect(Rect r,boolean isFill,int boldDot)

### 【描述】

该函数的作用是绘制一个矩形,所有绘制指令设置之前,需要设置打印纸张尺寸。

### 【参数】

Rect : 设置矩形的左,上,右,下.用以确定矩形起始坐标和宽高,单位:像素

isFill : 设置矩形框内是否填充颜色,颜色填充时,可视做绘制线条功能使用

boldDot : 设置矩形框粗细,单位:像素

### 【返回值】

无

### 【范例】

详见 Demo。

## 3.22 绘制文本

CTPL.*getInstance()*.drawText(Point p, int xScale, int yScale, String text)

### 【描述】

该函数的作用是绘制一段文本,所有绘制指令设置之前,需要设置打印纸张尺寸。注意缩放后的文字如果大于设置的纸张尺寸,将会不予显示。

### 【参数】

Point p: 设置文本的起始左上角,单位:像素

xScale: 1~10 设置文本的水平缩放倍率

yScale: 1~10 设置文本的垂直缩放倍率

text: 需要打印的内容

### 【返回值】

无

### 【范例】

详见 Demo。



## 3.23 绘制条码

`CTPL.getInstance().drawBarCode(Point, heightDot, BarCode, Paint.Align, Rotate, int narrow, int wide, text);`

### 【描述】

该函数的作用是绘制一个条码点阵图,所有绘制指令设置之前,需要设置打印纸张尺寸。

### 【参数】

Point p: 设置文本的起始左上角,单位:像素  
heightDot: 条码高度,单位:像素  
BarCode: 条码格式类型枚举(CODE39, CODE93, CODE128……)  
Paint.Align: 条码文本排版(LEFT, CENTER, RIGHT),值为 null 时不展示文本  
Rotate: 旋转角度枚举  
narrow: 条码窄竖条比例  
wide: 条码宽竖条比例  
text: 条码内容

### 【返回值】

无

### 【范例】

`CTPL.getInstance().drawBarCode(new Point(0,0), 80, BarCode.CODE_128, null, null, 2, 2, "12345678");`

## 3.24 绘制二维码

`CTPL.getInstance().drawQrCode(Point, QRLevel, cellWidth, QREncodeMode, text);`

### 【描述】

该函数的作用是绘制一个二维码点阵图,所有绘制指令设置之前,需要设置打印纸张尺寸。

### 【参数】

Point p: 设置文本的起始左上角,单位:像素  
QRLevel: 二维码纠错等级枚举  
cellWidth: 1~10 二维码单元格大小  
QREncodeMode: 模式类型枚举(AUTO, MANUAL)  
text: 二维码内容

### 【返回值】

无

**【范例】**

```
CTPL.getInstance().drawQrCode(new Point(0,0),  
    QRLevel.ECC_H, 5, QREncodeMode.AUTO, "12345678");
```

## 3.25 绘制图片

```
CTPL.getInstance().drawBitmap(Rect, Bitmap, boolean isFastly, Integer quality);
```

**【描述】**

该函数的作用是绘制一个图片,所有绘制指令设置之前,需要设置打印纸张尺寸。

**【参数】**

Rect:	设置图片的起始左,上,宽,高。单位:像素
Bitmap:	内容图片,需要使用白色作为背景底色
isFastly:	根据 queryCompressPrint() 查询结果,传入 true/false
quality:	二值化图片黑白转化判定阈值,参数为 null 时,使用默认值

**【返回值】**

无

**【范例】**

```
CTPL.getInstance().drawBitmap(new Rect(left,top,left + bitmap.getWidth(),top +  
bitmap.getWidth(),bitmap,false,null);
```

## 附录 返回值解释

回调返回值	解释	值
1	初始化参数错误	
4	断开设备连接	
256	蓝牙 BLE 连接成功	
257	蓝牙 SPP 连接成功	
258	USB 连接成功	
512	未支持的连接类型	
513	连接中进行重复连接	
514	已连接未断开进行重复连接	
515	参数输入错误	
516	权限不足以完成后续流程	
517	手机系统错误	
518	蓝牙 SPP 配对失败	
519	蓝牙 BLE 服务匹配失败	
DeviceCover	设备是否开盖	"Open"/"Close"
DevicePause	设备是否暂停	"true"/"false"
DeviceIdle	设备是否空闲	"true"/"false"
DeviceOverHeat	设备是否过热	"true"/"false"
CommandMode	指令模式	"Label"/"Receipt"
Density	浓度设置参数	"Int"
Speed	速度设置参数	"Int"
Battery	当前电量	"Int"
MemoryPrint	记忆打印设置开关	"On"/"Off"
SmartVelocity	变速打印设置开关	"On"/"Off"
PaperType	纸张类型设置参数	"Label"/"Receipt"/"BlackMark"
AutoShutdownMinute	自动关机时间(分钟)	"Int"
PrintDirection	打印方向	"Int"
Dpi	打印机分辨率	"Int"

## Android SDK 开发指南

PrintAlignment	卡纸器方向	"Left"/"Center"/"Right"
PrintMethod	打印方式	"Thermal"/"Carbon"
HardwareVersion	硬件版本	String
PrintMode	打印模式	"Label-Gap": 标签指令-间隙纸 "Recepit-Continuous": 票据指令-连续纸 "Label-Continuous": 标签指令-连续纸 "Label-BlackMark": 标签指令-黑标纸
SupportCompress	设备是否支持高速打印	"true"/"false"
DeviceCutter	设备是否支持切刀	"true"/"false"
SupportCarbonDetection	设备是否支持切刀检测	"true"/"false"
DevicePrintMaxWidthMM	设备最大打印宽度(毫米)	"Int"
DevicePrintMaxHeightMM	设备最大打印高度(毫米)	"Int"
SupportBeep	设备是否支持蜂鸣器	"true"/"false"
CutterReady	切刀是否准备就绪	"true"/"false"
PaperReady	纸张是否检测正常	"true"/"false"
CarbonReady	碳带是否检测正常	"true"/"false"
GapHeightMM	标签间隙高度(毫米)	"Int"
FeedOffsetMM	送纸偏移长度(毫米)	"Int"
PrintOffsetYMM	打印垂直偏移距离(毫米)	"Int"
PrintOffsetXMM	打印水平偏移距离(毫米)	"Int"
CalibrationMM	校准走纸长度(毫米)	"Int"
FirmwareVersion	固件版本	String