

简介

uniapp SDK 是基于我公司打印机开发的一套 uniapp 函数接口。此版本接口包含 CPCL 指令集和 ESC/POS 指令集接口，使打印机兼容 PT 系列打印机，基础版帮助客户更便捷的使用我们公司打印机。此 SDK 只支持蓝牙通讯方式。

仅支持 APP 端，小程序端请使用小程序 SDK

官方网址：<https://www.szputy.com/index.html>

打印宽度

指打印机可支持的最大横向打印范围，由打印机本身决定。例如：50mm 纸宽的打印机，打印最大有效宽度为 48mm(384 点)。

打印区域

打印区域可通过指令设置。打印区域一定小于等于打印宽度。

蓝牙操作

初始化

```
putyBleUtils.openBluetoothAdapter();
```

扫描蓝牙

```
putyBleUtils.startBluetoothDevicesDiscovery()
```

接收到打印机设备

```
putyBleUtils.onBluetoothDeviceFound()
```

停止搜索

```
putyBleUtils.stopBluetoothDevicesDiscovery()
```

关闭蓝牙模块

```
putyBleUtils.closeBluetoothAdapter()
```

发送 CPCL 指令到打印机

```
printCpcl (data,callback)
```

参数名	必选	类型	说明
data	是	string	数据
callback	是	Function	接收进度回调

发送 ESC 指令到打印机

```
printEsc (data,callback)
```

参数名	必选	类型	说明
data	是	array	数据
callback	是	Function	接收进度回调

搜索蓝牙综合案例

```
import putyBleUtils from '../..//puty/putyBleUtils';
import {alert} from '../..//puty/putyBaseUtils';
Page({
  /**
   * 页面的初始数据
   */
  data: {
    isBleInit: false, //是否初始化过蓝牙
    btDevices: [], //存放蓝牙设备数组
  },
  //生命周期函数--监听页面初次渲染完成
  onReady() {
    wx.startPullDownRefresh()
  },
  //下拉刷新
  onPullDownRefresh() {
    this.setData({btDevices:[]})
    const {isBleInit} = this.data;
    if(isBleInit){
      this.startBluetoothDevicesDiscovery();
    }else{
      this.openBluetoothAdapter()
    }
  },
  //初始化蓝牙
  async openBluetoothAdapter() {
    try {
      let res = await putyBleUtils.openBluetoothAdapter();
      this.setData({isBleInit:true})
      this.startBluetoothDevicesDiscovery();
    } catch (error) {
      console.log(error)
      alert(error)
      wx.stopPullDownRefresh();
    }
  }
})
```

```

    }
  },
  //开始扫描蓝牙
  async startBluetoothDevicesDiscovery() {
    try {
      let res = await putyBleUtils.startBluetoothDevicesDiscovery();
      this.onBluetoothDeviceFound();
    } catch (error) {
      alert(error.errMsg)
    }
  },
  //接收到打印机设备
  async onBluetoothDeviceFound() {
    let res = await putyBleUtils.onBluetoothDeviceFound();
    let btDevices = this.data.btDevices;
    btDevices.push(...res)
    let obj = {}
    let data = btDevices.reduce((item, next) => {
      obj[next.deviceId] ? '' : obj[next.deviceId] = true &&
      item.push(next)
      return item
    }, [])
    this.setData({btDevices:data})
  },
  //点击保存蓝牙信息
  blueToothItemTap(e) {
    wx.setStorageSync('bleDevice', e.currentTarget.dataset.deviceId);
    wx.reLaunch({url:"../index/index"})
  },
  //监听页面卸载
  onUnload: function() {
    //停止搜索
    putyBleUtils.stopBluetoothDevicesDiscovery();
    //关闭蓝牙模块
    putyBleUtils.closeBluetoothAdapter();
  }
})

```

开启下拉刷新

```

{
  "enablePullDownRefresh": true
}

```

CPCL 指令

初始化

`init(offset,width,height,page)` //注意必须放在所有指令之前调用

参数名	必选	类型	说明
-----	----	----	----

offset	是	int	整个标签的水平偏移量, 单位毫米
width	是	int	标签宽 单位毫米 最大 48
height	是	int	标签最大高度, 单位毫米 最大 60
page	是	int	打印标签的数量, 最大 1024 张

示例

```
let printData = putyCpclUtils.init(0,48,30,1).next().put();
putyBleUtils.printCpcl(printData);
```

打印文本

text(rotate,font,size,x,y,content)

参数名	必选	类型	说明
rotate	是	int	旋转角度 0、90、180、270
font	是	int	24 字符(12 24) 汉字(24 24) 55 字符(8 16) 汉字(16 16)
size	是	int	字符高度选择 ----> 0 1 2 3 4 5 6 7 纵向放大 size 纵向放大 1(正常) 2(2 倍高) 3 4 5 6 7 8 字符宽度选择 ----> 0 10 20 30 40 50 60 70 横向放大 size 1(正常) 2(2 倍高) 3 4 5 6 7 8 示例: 当传入 3 时, 纵向放大 3 倍 当传入 30 时, 横向放大 3 倍 当传入 33 时, 横纵向同时放大 3 倍
x	是	int	x 坐标, 单位毫米
y	是	int	y 坐标, 单位毫米
content	是	string	文本内容

示例

```
let printData = putyCpclUtils.init(0,48,30,1).text(0,24,11,2,2,"普贴打印").next().put();
putyBleUtils.printCpcl(printData);
```

设置文本间距

spacing(n)

参数名	必选	类型	说明
n	是	int	字符与字符之间的间隔大小 n*0.125mm

示例

```
let printData = putyCpclUtils.init(0,48,30,1).spacing(10).text(0,24,1
```

```
1,2,2,"普贴打印").next().put();
putyBleUtils.printCpcl(printData);
```

文本加粗

textBoldStart() textBoldEnd()

示例

```
let printData = putyCpclUtils.init(0,48,30,1).textBoldStart().text(0,
24,11,2,2,"普贴打印").textBoldEnd().next().put();
putyBleUtils.printCpcl(printData);
```

文本对齐方式

textAlign(aligned)

参数名	必选	类型	说明
align	是	int	① 0: 左对齐 1: 居中 2: 右对齐 ② 只对字符，条码有效，对线段 BOX 指令无效。 ③ 当 X 坐标为 0 时，字符和条码沿坐标原点对称打印。

示例

```
putyCpclUtils.init(0,48,30,1).align(0).text(0,24,11,2,2,"普贴打印").ne
xt().put(); //左对齐
```

```
putyCpclUtils.init(0,48,30,1).align(1).text(0,24,11,2,2,"普贴打印").ne
xt().put(); //居中对齐
```

```
putyCpclUtils.init(0,48,30,1).align(2).text(0,24,11,2,2,"普贴打印").next().put(); //右对齐
```

二维码

```
QrCode(direction,x,y,Mn,Un,rank,content)
```

参数名	必选	类型	说明
direction	是	Boolean	true 打印横向条码 false 打印纵向条码
x	是	int	x 坐标，单位毫米
y	是	int	y 坐标，单位毫米
Mn	是	int	QR 条码模式.n 范围：1 或者 2，默认：2
Un	是	int	放大倍数.n 范围：1 到 6，默认：6
rank	是	int	纠错等级 H-可靠性等级(Level H) Q-高可靠性等级(Level Q) M-标准等级(Level M) L-高密度等级(Level L)

data	是	string	二维码数据
------	---	--------	-------

示例

```
let printData = putyCpclUtils.init(0,48,30,1).QrCode(true,2, 2, 2, 6,
"Q", "123456").next().put();
putyBleUtils.printCpcl(printData);
```

一维条码

Barcode(direction,type,width,ratio,height,x,y,content)

参数名	必选	类型	说明
direction	是	Boolean	true 打印横向条码 false 打印纵向条码
type	是	string	条码类型 看下方条码类型表
width	是	int	条码宽度. 点为单位 宽度是根据内容及黑白条比例定
ratio	是	int	宽条宽和窄条宽比率 看下方宽条宽和窄条宽比率表
x	是	int	x 坐标, 单位毫米
y	是	int	y 坐标, 单位毫米
data	是	string	一维码内容

条码类型表

Type 值	条码类型
UPCA	UPC-A
UPCE	UPC-E
EAN13	JAN13 (EAN13)
EAN8	JAN 8 (EAN8)
39	CODE39
CODABAR	CODABAR
93	CODE93
128	CODE128 (Auto)

宽条宽和窄条宽比率表

比率	值
0	1.5:1
1	2.0:1
2	2.5:1
3	3.0:1
4	3.5:1
20	2.0:1
21	2.1:1

22	2.2:1
23	2.3:1
24	2.4:1
25	2.5:1
26	2.6:1
27	2.7:1
28	2.8:1
29	2.9:1
30	3.0:1

示例:打印无文本一维码

```
let printData = putyCpclUtils.init(0,48,30,1).Barcode(true,"128",1,1,8,2,2,"12345678910").next().put();
putyBleUtils.printCpcl(printData);
```

一维条码文本

BarcodeText(font,size,offset)

参数名	必选	类型	说明
font	是	int	24 字符(12 24) 汉字(24 24) 55 字符(8 16) 汉字(16 16)
size	是	int	字符高度选择 ----> 0 1 2 3 4 5 6 7 纵向放大 size 纵向放大 1(正常) 2(2 倍高) 3 4 5 6 7 8 字符宽度选择 ----> 0 10 20 30 40 50 60 70 横向放大 size 1(正常) 2(2 倍高) 3 4 5 6 7 8 示例:当传入 3 时, 纵向放大 3 倍 当传入 30 时, 横向放大 3 倍 当传入 33 时, 横纵向同时放大 3 倍
offset	是	int	文字偏移 单位 mm

示例:打印有文本一维码

```
let printData = putyCpclUtils.init(0,48,30,1).BarcodeText(8,0,1).Barcode(true,"128",2,1,8,2,2,"12345678").BarcodeTextEnd().next().put();
putyBleUtils.printCpcl(printData);
```

打印矩形

rectangle(xTop,yTop,xBottom,yBottom,width)

参数名	必选	类型	说明
xTop	是	int	左上角 X 坐标
yTop	是	int	左上角 Y 坐标

xBottom	是	int	右下角 X 坐标
yBottom	是	int	右下角 Y 坐标
width	是	int	形成矩形的线宽

xTop, yTop 代表左上角的起始点
xBottom, yBottom 代表右下角的结束点

```
let printData = putyCpclUtils.init(0,48,30,1).rectangle(1,1,20,20,1).next().put();
putyBleUtils.printCpcl(printData);
```

打印线段

line(xTop,yTop,xBottom,yBottom,width)

参数名	必选	类型	说明
xTop	是	int	左上角 X 坐标
yTop	是	int	左上角 Y 坐标
xBottom	是	int	右下角 X 坐标
yBottom	是	int	右下角 Y 坐标
width	是	int	形成线段的线宽

xTop, yTop 代表左上角的起始点
xBottom, yBottom 代表右下角的结束点

```
let printData = putyCpclUtils.init(0,48,30,1).line(1,1,20,1,1).next().put();
putyBleUtils.printCpcl(printData);
```

打印图片

image(ctx,width,height)

参数名	必选	类型	说明
ctx	是	obj	canvas2d 接口的 ctx 对象
width	是	int	canvas 宽度 单位 px 需要是 8 的倍数
height	是	int	canvas 高度 单位 px 需要是 8 的倍数

示例

```
//异步处理 canvasdom 回调
createSelectorQuery(id){
  return new Promise((resolve,reject) => {
```

```

        wx.createSelectorQuery()
        .select('#'+id)
        .fields({
            node: true,
            size: true,
        })
        .exec((res) => {
            resolve(res[0])
        });
    });
}
//canvas 基础绘制
async canvasInit(){
    let e = await this.createSelectorQuery("imageCanvas");
    let canvas = e.node;
    canvas.width = e.width;
    canvas.height = e.height;
    let ctx = canvas.getContext('2d');
    ctx.fillStyle="#000000";
    ctx.lineWidth = 8;
    ctx.strokeRect(8,8,368,216); //绘制边框
    ctx.moveTo(8,80);
    ctx.lineTo(372,80);
    ctx.moveTo(8,152);
    ctx.lineTo(372,152);
    ctx.stroke();
    ctx.font="30px Arial";
    ctx.fillText("品名: 卫龙大辣条",20,54);
    ctx.fillText("价格: ¥5",20,126);
    ctx.fillText("数量: 1",20,198);
    return {ctx,width:e.width,height:e.height};
}
//打印图片
async imageTap(){
    let canvas = await this.canvasInit();
    try {
        let printData = putyCpclUtils.init(0,48,30,1).image(canvas.ctx,canvas.width,canvas.height).next().put();
        let res = await putyBleUtils.printCpcl(printData,(e)=>{
            console.log(`已打印: ${e}%`)
        });
    } catch (error) {
        alert(error)
    }
}
}

```

综合应用

```

let printData = putyCpclUtils.init(0,48,30,1)
                                .qrCode(true,30, 8, 2, 6, "Q", "123456")
//打印二维码
                                .text(0,24,11,13,1,"普贴打印") //打印标题
                                .text(0,24,0,2,8,"产品名称: PT-66DC") //打
印文本
                                .text(0,24,0,2,13,"产品价格: ¥999") //打
印文本

```

```
.next() //自动走纸到缝隙处
.put(); //导出数据
putyBleUtils.printCpcl(printData);
```

自动走纸到缝隙处

next()

导出数据

put()

ESC 指令

初始化

init(width) //注意必须放在所有指令之前调用

参数名	必选	类型	说明
width	是	int	标签宽 单位毫米 最大 48

示例

```
let cmd = putyEscUtils.init(48).next().put()
putyBleUtils.printEsc(cmd);
```

打印浓度

setPotency(n)

参数名	必选	类型	说明
n	是	int	表示从浓度 1 到浓度 15

示例

```
let cmd = putyEscUtils.init(48).setPotency(15).next().put()
putyBleUtils.printEsc(cmd);
```

打印速度

setSpeed(n)

参数名	必选	类型	说明
n	是	int	表示从速度 1 到速度 5

示例

```
let cmd = putyEscUtils.init(48).setSpeed(5).next().put()  
putyBleUtils.printEsc(cmd);
```

纸张类型

setPaperType(n)

参数名	必选	类型	说明
n	是	int	n=0: 连续纸;n=1: 黑标纸;n=2: 间隙纸

示例

```
let cmd = putyEscUtils.init(48).setPaperType(2).next().put()  
putyBleUtils.printEsc(cmd);
```

打印空白行

printWhiteLine(n)

参数名	必选	类型	说明
n	是	int	打印几毫米

示例

```
let cmd = putyEscUtils.init(48).printWhiteLine(10).next().put()  
putyBleUtils.printEsc(cmd);
```

打印文本

text(content)

参数名	必选	类型	说明
content	是	string	文本内容

示例

```
let cmd = putyEscUtils.init(48).text("普贴打印").next().put()  
putyBleUtils.printEsc(cmd);
```

打印位置对齐方式

printAlign(n)

参数名	必选	类型	说明
n	是	int	0 左 1 中 2 右

示例

```
let cmd = putyEscUtils.init(48).printAlign(1).text("普贴打印").next().put()
putyBleUtils.printEsc(cmd);
```

设置行高

lineHeight(n)

参数名	必选	类型	说明
n	是	int	行高

示例

```
let cmd = putyEscUtils.init(48).lineHeight(6).text("普贴打印").next().put()
putyBleUtils.printEsc(cmd);
```

设置字符大小

fontSize(n)

参数名	必选	类型	说明
n	是	int	字体大小 使用十六进制 低位代表字体高度 高位代表字体宽度

示例

```
let cmd = putyEscUtils.init(48).fontSize(0x11).text("普贴打印").next().put(); //0x11 表示文字放大一倍
putyBleUtils.printEsc(cmd);
```

选择/取消 顺时针旋转 90 度

rotate(n)

参数名	必选	类型	说明
n	是	boolean	旋转还是取消

示例

```
let cmd = putyEscUtils.init(48).rotate(true).text("普贴打印").next().p
```

```
ut();  
putyBleUtils.printEsc(cmd);
```

选择/取消 绘制下划线

rotate(n)

参数名	必选	类型	说明
n	是	boolean	绘制下划线还是取消

示例

```
let cmd = putyEscUtils.init(48).underline(true).text("普贴打印").next()  
().put();  
putyBleUtils.printEsc(cmd);
```

选择/取消 加粗模式

fontWeight(n)

参数名	必选	类型	说明
n	是	boolean	加粗还是取消

示例

```
let cmd = putyEscUtils.init(48).fontWeight(true).text("普贴打印").next()  
().put();  
putyBleUtils.printEsc(cmd);
```

文本间距

spacing(n)

参数名	必选	类型	说明
n	是	int	n*0.125mm

示例

```
let cmd = putyEscUtils.init(48).spacing(20).text("普贴打印").next().pu  
t();  
putyBleUtils.printEsc(cmd);
```

一维码

BarCode(type,width,height,textAlign,content)

参数名	必选	类型	说明
type	是	int	条码类型 见下表(条码类型表)
width	是	int	见下方条码宽度模块表
height	是	int	条码高度 单位: mm
textAlign	是	int	0 不打印 1 条码上方 2 条码下方
content	是	string	内容

条码类型表

值	类型	内容长度
65	UPC-A	内容长度 11 - 12
66	UPC-E	内容长度 11 - 12
67	EAN13	内容长度 12 - 13
68	EAN8	内容长度 7 - 8
69	CODE39	内容长度 1 - 255
70	ITF	内容长度 1 - 255
71	CODABAR	内容长度 1 - 255
72	CODE93	内容长度 1 - 255
73	CODE128	内容长度 1 - 255

宽度模块表

n	单基本模块宽度 (mm)	双基本模块宽度	
		窄基本模块 (mm)	宽基本模块 (mm)
2	0.25 (2 点)	0.25 (2 点)	0.625 (5 点)
3	0.375 (3 点)	0.375 (3 点)	1.0 (8 点)
4	0.5 (4 点)	0.5 (4 点)	1.25 (10 点)
5	0.625 (5 点)	0.625 (5 点)	1.625 (13 点)
6	0.75 (6 点)	0.75 (6 点)	1.875 (15 点)

示例

```
let cmd = putyEscUtils.init(48).Barcode(73,2,15,2,"123456789104").next().put();
putyBleUtils.printEsc(cmd);
```

二维码

qrcode(content,size,rank)

参数名	必选	类型	说明
content	是	string	内容
size	是	int	大小 单位 mm

示例

```
let cmd = putyEscUtils.init(48).qrcode("12345678",10,3).next().put()  
putyBleUtils.printEsc(cmd);
```

打印图片

image(ctx,width,height)

参数名	必选	类型	说明
ctx	是	obj	canvas2d 接口的 ctx 对象
width	是	int	canvas 宽度 单位 px 需要是 8 的倍数
height	是	int	canvas 高度 单位 px 需要是 8 的倍数

示例

```
//异步处理 canvasdom 回调  
createSelectorQuery(id){  
  return new Promise((resolve,reject) => {  
    wx.createSelectorQuery()  
      .select('#'+id)  
      .fields({  
        node: true,  
        size: true,  
      })  
      .exec((res) => {  
        resolve(res[0])  
      });  
  })  
}  
  
//canvas 基础绘制  
async canvasInit(){  
  let e = await this.createSelectorQuery("imageCanvas");  
  let canvas = e.node;  
  canvas.width = e.width;  
  canvas.height = e.height;  
  let ctx = canvas.getContext('2d');  
  ctx.fillStyle="#000000";  
  ctx.lineWidth = 8;  
  ctx.strokeRect(8,8,368,216);//绘制边框  
  ctx.moveTo(8,80);  
  ctx.lineTo(372,80);  
  ctx.moveTo(8,152);  
  ctx.lineTo(372,152);  
  ctx.stroke();  
  ctx.font="30px Arial";  
  ctx.fillText("品名: 卫龙大辣条",20,54);  
  ctx.fillText("价格: ¥5",20,126);  
  ctx.fillText("数量: 1",20,198);  
  return {ctx,width:e.width,height:e.height};  
}  
  
//打印位图
```



```
async imageTap(){
    let canvas = await this.canvasInit();
    try {
        let cmd = putyEscUtils.init(48).image(canvas.ctx, canvas.width, canvas.height).next().put()
        let res = await putyBleUtils.printEsc(cmd);
        alert("打印完成")
    } catch (error) {
        alert(error)
    }
}
```

综合应用

```
let cmd = putyEscUtils.init(48)
    .printAlign(1)
    .fontWeight(true)
    .fontSize(0x11)
    .lineHeight(6)
    .text("普贴打印")
    .fontWeight(false)
    .fontSize(0)
    .lineHeight(5)
    .enter()
    .printWhiteLine(2)
    .printAlign(0)
    .text("产品名称: PT-66DC")
    .enter()
    .text("产品价格: ¥999")
    .printAlign(1)
    .enter()
    .BarCode(73, 2, 5, 2, "123456789104")
    .next()
    .put()
putyBleUtils.printEsc(cmd);
```

打印并回车

```
enter()
```

自动走纸到缝隙处

```
next()
```

导出数据

```
put()
```