

# 蓝牙打印文档

V1.0

2023/11/15

## 目录

蓝牙打印文档.....	1
文件结构:.....	3
运行方法:.....	3
集成步骤:.....	4
代码解析:.....	5

## 文件结构:

文件名: bluetooth-printer.zip

```
.
├─ App.vue
├─ CPCL 指令手册.pdf // 指令集参考手册
├─ LICENSE
├─ README.md
├─ libs
│   └─ print.js // uni-app 打印插件
├─ main.js // Demo 入口文件
├─ manifest.json // uni-app
├─ node_modules // 依赖
├─ pages
│   └─ index
│       └─ index.vue // Demo 页面，业务上使用可参考此文件
│   └─ setting
│       └─ index.vue // 打印机连接，可以集成到应用的系统设置功能
├─ pages.json // 页面配置文件
├─ uni.scss // uni-app 样式文件
└─ yarn.lock
```

## 运行方法:

1. 将代码解压缩，导入到 HBuilderX 中；
2. 连接手机；
3. 运行到手机或模拟器--运行到 Android App 基座；
4. 待运行完毕，在 蓝牙设置 界面点击“搜索设备”并进行连接；
5. 点击“Demo 界面”按钮跳转到 Demo，填写表单信息后点击“打印测试”按钮进行测试；



## 集成步骤:

1. 将 Demo 中 libs/print.js 复制到项目目录下;

2. 参考 Demo 中 `pages/setting/index.vue` 在项目合适位置集成打印机连接配置功能，一般在系统设置；
3. 在需要的文件中引用 `libs/print.js` 的 `print` 方法；
4. 拼接指令集，见 demo，更多使用方法参见 《CPCL 指令集变成文档.pdf》；
5. 调用 打印插件的 `print` 方法进行打印，入参为 打印机 ID 和 拼接好的指令集字符串；

## 代码解析：

```
*****libs/print.js*****
/**
 * 打印
 * @param mac_address 打印机 ID
 * @param data 指令集字符串，为了灵活起见，指令集在业务代码中进行转换然后传递进来
 */
export const print = (mac_address, data) => {
  var that = this
  if (!mac_address) {
    uni.showModal({
      title: "提示",
      content: "请选择蓝牙打印机",
      showCancel: false,
    })
    return
  }
  if (!data) {
    uni.showModal({
      title: "提示",
      content: "请提供打印数据.",
      showCancel: false,
    })
    return
  }

  main = plus.android.runtimeMainActivity()
  BluetoothAdapter = plus.android.importClass("android.bluetooth.BluetoothAdapter")
  var UUID = plus.android.importClass("java.util.UUID")
  uuid = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB")
  BAdapter = BluetoothAdapter.getDefaultAdapter()
  if (!BAdapter.isEnabled()) {
    uni.showModal({
      title: "提示",
      content: "蓝牙处于关闭状态，是否打开？",
      success: (_) => {
```

```

        if (confirm) {
            BAdapter.enable()
        }
    },
})
console.log("蓝牙处于关闭状态, 正在打开...")
return
}

device = BAdapter.getRemoteDevice(mac_address)
plus.android.importClass(device)
bluetoothSocket = device.createInsecureRfcommSocketToServiceRecord(uuid)
plus.android.importClass(bluetoothSocket)
if (!bluetoothSocket.isConnected()) {
    console.log("检测到设备未连接, 尝试连接....")
    bluetoothSocket.connect()
}
console.log("设备已连接")
if (bluetoothSocket.isConnected()) {
    var outputStream = bluetoothSocket.getOutputStream()
    plus.android.importClass(outputStream)
    outputStream.write([0x1b, 0x40]) //打印复位
    outputStream.flush()

    var bytes = plus.android.invoke(data, "getBytes", "gbk") /*utf-8*/

    outputStream.write(bytes)
    outputStream.flush()

    device = null //这里关键
    bluetoothSocket.close()
}
}

```

\*\*\*\*\*pages/index/index.vue\*\*\*\*\*

```

<template>
  <view>
    <view>
      <view>
        <input v-model="formData.name" class="uni-input" placeholder="公司名称" />
        <input v-model="formData.model" class="uni-input" placeholder="车型" />
        <input v-model="formData.code" class="uni-input" placeholder="条码" />
        <input v-model="formData.line" class="uni-input" placeholder="产线" />
        <input v-model="formData.box" class="uni-input" placeholder="箱号" />
      </view>
    </view>
  </view>
</template>

```

```

        <input v-model="formData.date" class="uni-input" placeholder="日期" />
        <input v-model="formData.operator" class="uni-input" placeholder="装箱人" />
        <input v-model="formData.auditor" class="uni-input" placeholder="确认人" />
    </view>
    <view class="buttos-bar">
        <button class="plain-button plain-button--blue" @click="printTest"> 打 印 测 试
</button>

        <navigator url="/pages/setting/index" hover-class="navigator-hover">
            <button type="default">跳转到设置界面</button>
        </navigator>
    </view>
</view>
</view>
</view>
</template>
<script>
    // 引入打印插件的打印方法
    import {
        print
    } from '@/libs/print.js'

    export default {
        name: 'PrintDemo',
        data() {
            return {
                // 业务数据
                formData: {
                    name: "鹤山市捷仕克汽车配件有限公司",
                    model: "型号 123456789",
                    code: "编码 123456789",
                    line: "产线 1",
                    box: "序号 1",
                    date: "2023/11/15",
                    operator: "操作人",
                    auditor: "审核人",
                }
            }
        },
        methods: {
            printTest() {
                // 从缓存中获取已经连接的打印机信息
                var printerid = uni.getStorageSync('ble_printerId')
                if (printerid) {
                    if (printerid !== null && printerid.length > 0) {

```

```

const data = this.formData

// 标签开始, 固定开头, 详见 指令集文档
var str = "! 0 200 200 350 1 " + '\r\n';
// 设置打印纸张宽度
str += "PAGE-WIDTH 600" + '\r\n';

// 标签内容
// 文本 {command} {font} {size} {x} {y} {data}
str += "TEXT 24 0 30 50 " + data.name + "\r\n";
// 二维码 {command} {type} {x} {y} [M n] [U n] {data}
str += "B QR 380 20 M 2 U 5" + '\r\n';
str += "MA," + data.code + "\r\n";
str += "ENDQR" + '\r\n';

str += "TEXT 24 0 30 100 车型:" + data.model + "\r\n";
str += "TEXT 24 0 30 150 条码编号:" + data.code + "\r\n";
str += "TEXT 24 0 320 150 生产线号:" + data.line + "\r\n";
str += "TEXT 24 0 30 200 装箱序号:" + data.box + "\r\n";
str += "TEXT 24 0 320 200 日期:" + data.date + "\r\n";
str += "TEXT 24 0 30 250 装箱人:" + data.operator + "\r\n";
str += "TEXT 24 0 320 250 确认人:" + data.auditor + "\r\n";

// 标签结束
str += "GAP-SENSE" + '\r\n';
str += "FORM " + '\r\n';
str += "PRINT " + '\r\n';

// 指令集拼接完成, 调用打印插件打印方法进行打印
print(printerid, str);
    }
} else {
    uni.showModal({
        title: '提示',
        content: '请先选择已配对的蓝牙打印机, 再进行测试.',
        showCancel: false
    })
}
},
},
}
</script>

```



\*\*\*\*\*pages/setting/index.vue\*\*\*\*\*

方法解释:

searchDevices //开始搜寻附近的蓝牙外围设备

onConn // 连接打印机