uniapp 热敏打印机插件

目　录

使用方法

介绍

安卓热敏打印机打印插件，自动授权，打印机连接监听，打印文本，条形码，二维码，切纸，打印机状态，打印结果查询等

联系作者

关注微信公众号可联系作者

插件地址

https://ext.dcloud.net.cn/plugin?id=13037

所需权限

```
<uses-permission android:name="android.hardware.usb.host" />
```

用法

在需要使用插件的页面加载以下代码

```
const module = uni.requireNativePlugin("leven-printer-PrinterModule");
```

状态码

打印机连接状态

状态码	描述
-101	未知错误
-102	打印机连接失败
-103	打印机连接关闭
-104	未找到打印机设备
-105	没有usb权限
-106	usb插入
-106	usb拔出

打印机状态

状态码	描述
0	打印机正常
-201	接收数据失败
-202	打印机缺纸
-203	打印机纸将尽
-204	打印机开盖
-205	发送数据失败

打印机打印结果状态

状态码	描述
0	打印完成
-301	未知错误
-302	正在打印中
-303	打印未完成因为缺纸

状态码

状态码	描述
-304	打印未完成，纸舱盖开盖
-305	打印未完成，与打印机通信失败
-306	查询指令发送失败，通信异常
-307	接收数据格式不正确

更新日志

2024-01-12 `v1.1.1`

- [优化] 修复能获取到打印机设备但是连接失败的bug

2023-11-07 `v1.1.0`

- [新增]初始化usb接口
- [优化]usb连接

2023-06-14

首次发布

初始化usb v1.1.0

方法名

`initUsb`

> 初始化usb是为了获取连接的usb设备和usb打印机设备，为usb连接做准备

用法

- 用法如下

```
module.initUsb(res => {
  if (res.code == 0) {
    let data = res.data;
    this.usbDeviceList = data.usbDevices;
    this.usbPriterList = data.printerDevices;
  } else {
    this.showToast(res.message)
  }
})
```

- 参数说明

  无

回调示例


```
{
    "message": "",
    "code": 0,
    "data": {
        "usbDevices": [
            {
                "deviceName": "/dev/bus/usb/003/002",
                "deviceClass": 239,
                "productName": "RXGD-CAMERA",
                "vendorId": 3034,
                "productId": 12337,
                "deviceId": 3002
            },
            {
                "deviceName": "/dev/bus/usb/001/005",
```

```
                "deviceClass": 0,
                "productName": "USB OPTICAL MOUSE ",
                "vendorId": 10077,
                "productId": 2982,
                "deviceId": 1005
            },
            {
                "deviceName": "/dev/bus/usb/001/004",
                "deviceClass": 0,
                "productName": "TOT2D PRODUCT DEFHID+KBW",
                "vendorId": 44953,
                "productId": 32772,
                "deviceId": 1004
            },
            {
                "deviceName": "/dev/bus/usb/001/003",
                "deviceClass": 0,
                "productName": "REGO PRINTER",
                "vendorId": 1155,
                "productId": 22304,
                "deviceId": 1003
            },
            {
                "deviceName": "/dev/bus/usb/001/006",
                "deviceClass": 239,
                "productName": "Android",
                "vendorId": 11388,
                "productId": 24581,
                "deviceId": 1006
            }
        ],
        "printerDevices": [
            {
                "deviceName": "/dev/bus/usb/001/003",
                "deviceClass": 0,
                "productName": "REGO PRINTER",
                "vendorId": 1155,
                "productId": 22304,
                "deviceId": 1003
            }
        ]
    }
}
```


回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.usbDevices	Array	所有的USB列表
data.usbDevices.productId	Integer	产品id
data.usbDevices.vendorId	Integer	供应商id
data.usbDevices.productName	String	产品名称
data.usbDevices.deviceClass	Integer	设备序列号
data.usbDevices.devicesName	String	设备名称
data.usbDevices.deviceId	Integer	设备id
data.printerDevices	Array	所有的打印机设备列表
data.printerDevices .productId	Integer	产品id
data.printerDevices .vendorId	Integer	供应商id
data.printerDevices .productName	String	产品名称
data.printerDevices .deviceClass	Integer	设备序列号
data.printerDevices .devicesName	String	设备名称
data.printerDevices .deviceId	Integer	设备id
code	Integer	返回类型，0.成功，其他：失败

usb连接

方法名

`usbConnect`

> 所有的连接监听也在此方法中，包括关闭连接、断开连接等

用法

- 用法如下

```
module.usbConnect({
  deviceName: this.usbPriterList[0].deviceName
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
deviceName	String	是	无	USB初始化后返回的设备名称

回调示例

```
{
    "message": "连接成功",
    "code": 0,
    "data": {
        "deviceName": "/dev/bus/usb/001/003",
        "deviceClass": 0,
        "productName": "REGO PRINTER",
        "vendorId": 1155,
        "productId": 22304,
        "deviceId": 1003
    }
}
```

回调说明

usb连接

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.devicesName	String	设备名称
data.deviceClass	String	设备序列号
data.productName	String	产品名称
data.vendorId	Integer	供应商id
data.productId	Integer	产品id
data.deviceId	Integer	设备id
code	Integer	返回类型，0.成功，其他：失败

关闭连接

方法名

`close`

用法

```
module.close(res => {
  console.log(res)
})
```

回调示例

```
{
    "message": "",
    "code": 0,
    "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

检查状态

方法名

`printStatus`

用法

```
module.printStatus(res => {
  console.log(res)
})
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.printStatus	Number	状态值，0.正常，其他.异常（请参考状态码）
code	Integer	返回类型，0.成功，其他：失败

回调示例

```
{
    "message": "打印机状态正常",
    "code": 0,
    "data": {
        "printStatus": 0
    }
}
```

打印文本

方法名

`printText`

用法

```
// 自定义打印
module.printText({
  content: "默认打印内容",
  align: 1, //对齐方式，0.左对齐，1.居中对齐，2.右对齐
  bold: 1, //是否加粗，0.不加粗，1.加粗
  underline: 1, //是否加下划线，0.不加下划线，1.加下划线
  width: 1, //文字宽度放大倍数，0.不放大，1.放大1倍，2.放大2倍...
  height: 1, //文字高度放大倍数，0.不放大，1.放大1倍，2.放大2倍...
  wrapLineCount: 1, //换行行数，1.换1行，2.换2行
}, res => {
  console.log(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
content	String	是	无	打印内容
align	Number	否	0	对齐方式，0.左对齐，1.居中对齐，2.右对齐
bold	Number	否	0	是否加粗，0.不加粗，1.加粗
underline	Number	否	0	是否加下划线，0.不加下划线，1.加下划线
width	Number	否	0	文字宽度放大倍数，0.不放大，1.放大1倍，2.放大2倍...

参数名	参数类型	是否必填	默认值	描述
height	Number	否	0	文字高度放大倍数，0.不放大，1.放大1倍，2.放大2倍...
wrapLineCount	Number	否	0	换行行数，1.换1行，2.换2行

回调示例

```
{
    "message": "处理完成",
    "code": 0,
    "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印条形码

方法名

`printBarcode`

用法

```
module.printBarcode({
  type: "code128", //类型，code128（默认），code39，codabar，itf，code93，upc_a，upc_e
  content: "1234567890", //打印内容
  align: 1, //对齐方式，0.左对齐（默认），1.居中对齐，2.右对齐
  width: 1, //宽度，默认0
  height: 1, //高度，默认0
  textPosition: 1, //内容显示,0.不显示（默认），1.上方，2.下方，3.上下都有
  wrapLineCount: 1, //换行行数，0.不换行（默认），1.换1行，2.换2行
}, res => {
  console.log(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	否	code128	条形码类型，类型包含：code128，code39，codabar，itf，code93，upc_a，upc_e，ean13，ean8
content	String	是	无	打印内容
align	Number	否	0	对齐方式，0.左对齐，1.居中对齐，2.右对齐
width	Number	否	0	宽度
height	Number	否	0	高度

参数名	参数类型	是否必填	默认值	描述
textPosition	Number	否	0	内容显示,0.不显示（默认），1.上方，2.下方，3.上下都有
wrapLineCount	Number	否	0	换行行数，1.换1行，2.换2行

回调示例

```
{
    "message": "处理完成",
    "code": 0,
    "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印二维码

方法名

用法

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	否	qrcode	二维码类型，qrcode（默认），pdf417，datamatrix
content	String	是	无	打印内容
align	Number	否	0	对齐方式，0.左对齐，1.居中对齐，2.右对齐
width	Number	否	0	qrcode图形版本号，PDF417每行字符数，1~30，DataMatrix.条码宽度
level	Number	否	7	纠错等级，0~8，默认：7
multiple	Number	否	5	放大倍数1~6
wrapLineCount	Number	否	0	换行行数，1.换1行，2.换2行

回调示例

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

切纸

方法名

`cutPaper`

> 打印机要有切纸功能

用法

```
module.cutPaper({
  type: 0, //类型，0.全切（默认），1.半切
}, res => {
  console.log(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	Number	否	0	类型，0.全切，1.半切

回调示例

```
{
    "message": "处理完成",
    "code": 0,
    "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示

切纸

参数名	参数类型	参数描述
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

一键打印

方法名

`printList`

用法

```
module.printList({
  list: [{
    printType: 1, //打印类别，1.文本，2.条形码，3.二维码
    content: "默认打印内容",
    align: 1, //对齐方式，0.左对齐，1.居中对齐，2.右对齐
    bold: 1, //是否加粗，0.不加粗，1.加粗
    underline: 1, //是否加下划线，0.不加下划线，1.加下划线
    width: 1, //文字宽度放大倍数，0.不放大，1.放大1倍，2.放大2倍...
    height: 1, //文字高度放大倍数，0.不放大，1.放大1倍，2.放大2倍...
    wrapLineCount: 1, //换行行数，1.换1行（默认），2.换2行
  }, {
    printType: 2, //打印类别，1.文本，2.条形码，3.二维码
    type: "code128", //类型，code128（默认），code39，codabar，itf，code93，upc_a，upc
    content: "1234567890", //打印内容
    align: 1, //对齐方式，0.左对齐（默认），1.居中对齐，2.右对齐
    width: 1, //宽度，默认0
    height: 1, //高度，默认0
    textPosition: 1, //内容显示,0.不显示（默认），1.上方，2.下方，3.上下都有
    wrapLineCount: 1, //换行行数，0.不换行（默认），1.换1行，2.换2行
  }, {
    printType: 3, //打印类别，1.文本，2.条形码，3.二维码
    type: "qrcode", //类型，qrcode（默认），pdf417，datamatrix
    content: "1234567890", //打印内容
    align: 1, //对齐方式，0.左对齐（默认），1.居中对齐，2.右对齐
    width: 1, //qrcode图形版本号，PDF417每行字符数，1~30， DataMatrix.条码宽度
    level: 1, //纠错等级，0~8，默认：7
    multiple: 1, //放大倍数1~6
    wrapLineCount: 1, //换行行数，0.不换行（默认），1.换1行，2.换2行
  }],
  isCutPaper: true, //是否切纸，true/false
  cutPaperType: 0, //切纸类型，0.全切，1.半切
}, res => {
  console.log(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
list	Array	是	无	打印列表，具体参数说明可参考对应的方法
list.printType	Number	是	无	打印类别，1.文本，2.条形码，3.二维码
isCutPaper	Boolean	否	false	是否切纸
cutPaperType	Number	否	无	切纸类型，0.全切，1.半切（isCutPaper 为true时有效）

回调示例

```
{
    "message": "处理完成",
    "code": 0,
    "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

检查打印状态

方法名

`printingStatus`

用法

```
module.printingStatus({
  timeout: 2000, //检查超时时间，默认：1000，单位：毫秒
}, res => {
  console.log(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
timeout	Number	否	1000	检查超时时间，默认：1000，单位：毫秒

回调示例

```
{
    "message": "打印完成",
    "code": 0,
    "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

检查打印状态

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败(具体可参考状态码)