uniapp 安卓创享科技一体机UTS打印插件

目　录

使用方法

介绍

安卓创享科技一体机UTS打印插件是创想科技一体式打印机打印服务SDK开发的插件，插件已包含SDK的全部功能，欢迎使用

联系作者

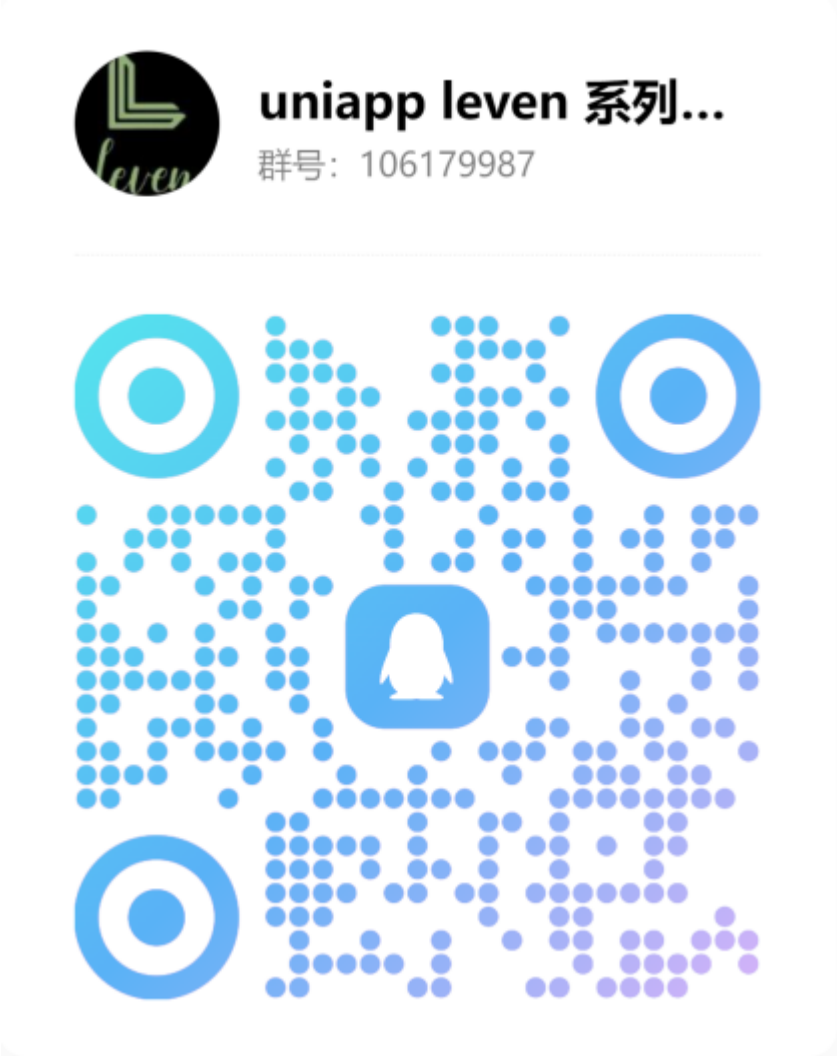
关注微信公众号可联系作者

插件地址

https://ext.dcloud.net.cn/plugin?id=22792

演示程序

加群下载演示程序



用法

在需要使用插件的页面加载以下代码

```
import * as module from "@/uni_modules/leven-uts-cxPrinter"
```

示例程序

示例程序页面较多，请通过插件市场插件页面导入示例项目

更新日志

2025-03-27

首次发布

功能类型说明

func功能类型说明

标识	描述	示例
TX_FONT_SIZE	控制字体的放大倍数功能，后面的 2 个参数 0 为正常大小，递增 1 为增大 1 倍，如此类推，最大为 7；参数 1(param1)为字体宽的倍数，参数 2(param2)为字体高的倍数	doFunction("TX_FONT_SIZE","TX_SIZE_3X","TX_SIZE_2X");
TX_FONT_ULINE	控制是否有下划线功能，只需 1 个参数param1=TX_ON(有下划线)，=TX_OFF(无小划线)，param2="" 就可以了。	doFunction("TX_FONT_ULINE","TX_ON","")
TX_FONT_BOLD	控制是否粗体功能，只需 1 个参数param1=TX_ON(使用粗体)，=TX_OFF(不使用粗体),param2="" 就可以了。	doFunction("TX_FONT_BOLD","TX_ON","")
TX_SEL_FONT	选择英文字体功能，也是只需要 1 个参数，TX_FONT_A(12X24) 或 TX_FONT_B(9X17),param2="" 就可以了。	doFunction("TX_SEL_FONT","TX_FONT_B","")
TX_FONT_ROTATE	字体是否旋转 90 度功能，只需要 1 个参数	doFunction("TX_FONT_ROTATE","TX_ON","")
TX_ALIGN	控制对齐功能，参数为 TX_ALIGN_XXX，只需要 1 个参数，可参考下方的对齐方式说明	doFunction("TX_ALIGN","TX_ALIGN_CENTER","")
TX_CHINESE_MODE	中文/英文模式切换功能，只需要 1 个参数	doFunction("TX_CHINESE_MODE","TX_ON","")
TX_FEED	执行走纸功能，只需要 1 个参数，参数以 0.125 毫米为单位，这个参数最大不能超过 255	doFunction("TX_FEED","240","")
TX_UNIT_TYPE	设置动作单位(无效)	-

标识	描述	示例
TX_CUT	执行切纸功能，第一参数指明类型，第二参数指明切纸前的走纸距离	doFunction("TX_CUT", "TX_PURECUT_FULL","")
TX_HOR_POS	绝对水平定位功能，只需要 1 个参数，参数以 0.125 毫米为单位。	doFunction("TX_HOR_POS","160","")
TX_LINE_SP	设置行间距功能，只需要 1 个参数，参数是以点数为单位的。	doFunction("TX_LINE_SP","30","")
TX_BW_REVERSE	设置字体是否黑白翻转功能，只需要 1 个参数	doFunction("TX_BW_REVERSE", "TX_ON","")
TX_UPSIDE_DOWN	设置是否倒置打印功能，只需要 1 个参数	doFunction("TX_UPSIDE_DOWN","TX_ON","")
TX_INET_CHARS	选择国际字符集功能，只需要 1 个参数,通常这个设置也是在英文方式下使用的，只是针对12 个特定的 ASCII 码不同国家的使用的字形不同，默认是 0，表示是使用美国的字符集。	doFunction("TX_INET_CHARS", "1","")
TX_CODE_PAGE	选择字符代码页功能，只需要 1 个参数，通常是 0~n,表示选择的代码页参数，祥见打印机说明书，一般默认是 0，是表示 PC437 的代码页，这个功能要只在英文方式下有效	doFunction("TX_CODE_PAGE","3","");
TX_CH_ROTATE	设定汉字旋转功能，只需要 1 个参数，可以表示 3 种选择，详见下方说明	doFunction("TX_CH_ROTATE","TX_CH_ROTATE_LEFT","");
TX_CHK_BMARK	寻找黑标黑标，TX_CHK_BMARK 是执行黑标检测，这个动作本身是不需要参数的	doFunction("TX_CHK_BMARK","","")
TX_SET_BMARK	设置黑标相关偏移量功能，这里有 2 个参数要设置，起始打印位置相对于黑标检测位置的偏移量和切/撕纸位置相对于黑标检测位置的偏移量	-

标识	描述	示例
TX_PRINT_LOGO	打印已下载好的 LOGO 的功能，只需 1 个参数，可以表示 4 种选择，请参考下方说明	doFunction("TX_PRINT_LOGO","TX_LOGO_1x1","")
TX_BARCODE_HEIGHT	设定条码高度功能,只需要 1 个参数,单位为点数	doFunction("TX_BARCODE_HEIGHT","15","");
TX_BARCODE_WIDTH	设定条码宽度功能,只需要 1 个参数,单位为点数, 不能小于 2	doFunction("TX_BARCODE_WIDTH","3","");
TX_BARCODE_FONT	选择条码 HRI 字符的打印位置,只需要 1 个参数，可以表示 4 种选择，请参考下方说明	doFunction("TX_BARCODE_FONT","TX_BAR_FONT_DOWN","");
TX_FEED_REV	执行反向走纸功能，只需要 1 个参数，参数以 0.125 毫米为单位	doFunction("TX_FEED_REV","240","");
TX_QR_DOTSIZE	设置 QR 码的点大小，2<param1<10	doFunction("TX_QR_DOTSIZE","6","");
TX_QR_ERRLEVEL	设置 QR 码的纠错等级，有 4 个等级可以选择，等级越高，可以支持的字符数越少（在同一个版本的情况下）,请参考下方说明	doFunction("TX_QR_ERRLEVEL","TX_QR_ERRLEVEL_L","")
TX_HOR_POS_REL	相对于当前位置的水平定位功能，只需要 1 个参数，参数以 0.125 毫米为单位	doFunction("TX_HOR_POS_REL","160","");
TX_DIRECTION	设置页模式下的打印方向，只需要 1 个参数，参数值为 0,1,2,3 这 4 种选择，请参考下方说明	doFunction("TX_DIRECTION","0","")
TX_VERT_POS	页模式下设置垂直打印位置，设置相对于起始位置的垂直打印位置，单位为 0.125mm,只需要 1 个参数	doFunction("TX_VERT_POS","160","")

其他类型定义

标识	描述
TX_BM_START	起始打印位置相对于黑标检测位置的偏移量
TX_BM_TEAR	切/撕纸位置相对于黑标检测位置的偏移量

TX_UNIT_TYPE类型定义

标识	描述
TX_UNIT_PIXEL	0
TX_UNIT_MM	1

doFunction 的参数的定义（需和功能类型对应）

标识	描述
TX_ON	功能设置有效的参数
TX_OFF	功能设置无效的参数
TX_FONT_A	12X24 点阵的设置参数
TX_FONT_B	9X17 点阵的设置参数

控制字体倍数的参数设置

标识	描述
TX_SIZE_1X	正常大小
TX_SIZE_2X	2 倍大小
TX_SIZE_3X	3 倍大小
TX_SIZE_4X	4 倍大小
TX_SIZE_5X	5 倍大小
TX_SIZE_6X	6 倍大小
TX_SIZE_7X	7 倍大小
TX_SIZE_8X	8 倍大小

页模式下的打印方向功能说明

标识	描述
0	从左到右

标识	描述
1	从底到上
2	从右到左
3	从上到下

QR 码的纠错等级功能说明

标识	描述
TX_QR_ERRLEVEL_L	0x31
TX_QR_ERRLEVEL_M	0x32
TX_QR_ERRLEVEL_Q	0x33
TX_QR_ERRLEVEL_H	0x34

条码 HRI 字符的打印位置功能说明

标识	描述
TX_BAR_FONT_NONE	不打印
TX_BAR_FONT_UP	打印在条码的上面
TX_BAR_FONT_DOWN	打印在条码的下面
TX_BAR_FONT_BOTH	条码的上面下面都打印

LOGO功能说明

标识	描述
TX_LOGO_1X1	对应 203X203 的点密度，就是正常大小
TX_LOGO_1X2	对应 203x101 的点密度，就是 LOG 在水平方向上放大到了 2 倍
TX_LOGO_2X1	对应 101x203 的点密度，就是 LOG 在垂直方向上放大到了 2 倍。
TX_LOGO_2X2	对应 101x101 的点密度，就是 LOG 在水平和垂直方向上放大到了 2 倍。

对齐方式说明

标识	描述
TX_ALIGN_LEFT	左对齐
TX_ALIGN_CENTER	居中对齐
TX_ALIGN_RIGHT	右对齐

切纸类型说明

标识	描述
TX_CUT_FULL	全切的设置参数(和黑标检测关联）
TX_CUT_PARTIAL	半切的设置参数（和黑标检测关联）
TX_PURECUT_FULL	全切（与黑标无关的切纸）
TX_PURECUT_PARTIAL	半切（与黑标无关的切纸）

汉字旋转功能说明

标识	描述
TX_CH_ROTATE_NONE	不旋转
TX_CH_ROTATE_LEFT	向左旋转
TX_CH_ROTATE_RIGHT	向右旋转

打印机状态说明

标识	描述
TX_STAT_UNKNOWN	未知错误
TX_STAT_NOERROR	无故障
TX_STAT_SELECT	处于联机状态
TX_STAT_PAPEREND	缺纸
TX_STAT_BUSY	繁忙

标识	描述
TX_STAT_DRAW_HIGH	钱箱接口的电平（整机使用的，模块无用）
TX_STAT_COVER	打印机机芯的盖子打开
TX_STAT_ERROR	打印机错误
TX_STAT_RCV_ERR	可恢复错误（需要人工干预）
TX_STAT_CUT_ERR	切刀错误
TX_STAT_URCV_ERR	不可恢复错误
TX_STAT_ARCV_ERR	可自动恢复的错误
TX_STAT_PAPER_NE	快要没有纸了

申请插件所需权限

方法名

`requestPermissions`

用法

- 用法如下：

```
module.requestPermissions(res => {
  console.log(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {
    "grantedList": [
      "android.permission.READ_EXTERNAL_STORAGE",
      "android.permission.WRITE_EXTERNAL_STORAGE"
    ]
  },
  "message": "权限被拒绝",
  "code": -1
}
```

- 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

申请插件所需权限

参数名	参数类型	参数描述
data.grantedList	Array	申请的权限列表，**被拒绝的时候会返回**
code	Integer	返回类型，0.成功，其他：失败

连接打印机

方法名

`open`

> 使用插件前需先连接打印机

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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

关闭连接

方法名

`close`

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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

初始化

方法名

`init`

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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

执行特殊功能

方法名

`doFunction`

用法

- 用法如下：

```
module.doFunction({
  //功能类型
  func: "TX_FONT_ULINE",
  //参数一
  param1:"TX_ON",
  //参数二
  param2:""
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
func	String	是	无	功能类型
param1	String	是	无	参数一，可以是数值字符串
param2	String	否	无	参数一，可以是数值字符串

> 有的功能需要传入两个参数，有的功能只需要一个参数（param2=""）。这个函数可以执行的功能比较多，详见下面的具体说明。
> 就是向打印机发送设置指令，或着走纸指令，定位指令等，具体要见下面做好的定义
> 有的功能需要传入两个参数，有的功能只需要一个参数（此时 param2="" 就可以了）。
> 功能类型和参数可参考【功能类型说明】

回调

- 示例

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

- 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

输出字符串并换行

方法名

`outputStringLn`

用法

- 用法如下：

```
module.outputStringLn({
  //字符串内容
  text: "字符串内容"
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
text	String	是	无	字符串内容

回调

- 示例

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

- 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

获取打印机状态

方法名

`getStatus`

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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.status	String	状态类型，具体可参考【功能类型说明】
code	Integer	返回类型，0.成功，其他：失败

获取打印机状态2

方法名

`getStatus2`

> 与 `getStatus` 类似，区别是 USB 接口通过指令查询状态。这个函数下，USB 可以获得打印机更多的信息。最好是在调用前一个函数确认打印有纸，无故障，不忙的情况下调用此函数。

用法

- 用法如下：

```
module.getStatus2(res => {
  console.log(res)
});
```

- 参数说明

  无

回调

- 示例

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

- 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

获取打印机状态2

参数名	参数类型	参数描述
data.status	String	状态类型，具体可参考【功能类型说明】
code	Integer	返回类型，0.成功，其他：失败

输出字符串不换行

方法名

`outputString`

用法

- 用法如下：

```
module.outputString({
  //字符串内容
  text: "字符串内容"
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
text	String	是	无	字符串内容

回调

- 示例

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

- 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

输出回车换行

方法名

`newline`

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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

恢复字体效果

方法名

`resetFont`

> 恢复字体效果（大小、粗体等）为原始状态。

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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

打印条码

方法名

`printBarcode`

用法

- 用法如下：

```
module.printBarcode({
  //类型
  type: "TX_BAR_UPCA",
  //条码内容
  text: "123456"
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
type	String	是	无	类型，可参考【功能类型说明】
text	String	是	无	字符串内容

回调

- 示例

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

- 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

打印图形文件

方法名

`printImage`

用法

- 用法如下：

```
module.printImage({
  //文件路径
  path: "/storage/sdcard/0/a1.png"
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
path	String	是	无	文件路径

回调

- 示例

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

- 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

`printQRcode`

用法

- 用法如下：

```
module.printQRcode({
  //条码内容
  text: "123456"
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
text	String	是	无	字符串内容

回调

- 示例

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

- 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

选择页模式

方法名

`selectPageMode`

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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

选择页模式并设置打印区域

方法名

`selectPageMode2`

用法

- 用法如下：

```
module.selectPageMode2({
  //水平起始位置（单位根据所设置的）
  x: 1,
  //垂直起始位置（单位同上）
  y: 2,
  //打印区域宽度（单位同上）
  dx: 1,
  //打印区域高度（单位同上）
  dy: 2
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
x	int	是	无	水平起始位置（单位根据所设置的）
y	int	是	无	垂直起始位置（单位同上）
dx	int	是	无	打印区域宽度（单位同上）
dy	int	是	无	打印区域高度（单位同上）

回调

- 示例

选择页模式并设置打印区域

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

- 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

页模式下打印数据

方法名

`printPage`

用法

- 用法如下：

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

- 参数说明

  无

回调

- 示例

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

- 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