

uniapp 安卓商米电子秤称重UTS原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
初始化	2.1
断开连接	2.2
称重监听	2.3
取消称重监听	2.4
获取电子秤服务版本号	2.5
获取固件版本号	2.6
清零	2.7
去皮	2.8
数字去皮	2.9
读取加速度数据	2.10
获取铅封状态	2.11
读取标定按钮开关状态	2.12
读取电子秤参数信息	2.13
重启电子秤	2.14
设置单价	2.15
获取当前已经设置的单价	2.16
设置价格计算时的重量单位	2.17
获取当前价格计算的重量单位	2.18
获取当前电子秤的计量类型	2.19
释放资源	2.20

使用方法

介绍

安卓商米电子秤称重UTS原生插件是商米电子秤称重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=22536>

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-sanmi-scale"
```

页面内容

```
<template>
  <view style="display: flex; flex-direction: column; padding: 16px;">
    <view style="margin-bottom: 10px;">
      <u-button type="primary" text="初始化" @click="init"></u-button>
```

```

</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="断开连接" @click="disconnect"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="称重监听" @click="scaleListener"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="取消称重监听" @click="cancelScaleListener"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="获取电子秤服务版本号" @click="getServiceVersion"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="获取固件版本号" @click="getFirmwareVersion"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="清零" @click="zero"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="去皮" @click="tare"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="数字去皮" @click="digitalTare"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="读取加速度数据" @click="readAcceleData"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="获取铅封状态" @click="readSealState"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="读取标定按钮开关状态" @click="getCalStatus"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="读取电子秤参数信息" @click="getCalInfo"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="重启电子秤" @click="restart"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="设置单价" @click="setUnitPrice"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="获取当前已经设置的单价" @click="getUnitPrice"></u-button>
</view>
<view style="margin-bottom: 10px;">
  <u-button type="primary" text="设置价格计算时的重量单位" @click="setUnit"></u-button>
</view>
<view style="margin-bottom: 10px;">

```

```

    <u-button type="primary" text="获取当前价格计算的重量单位" @click="getUnit"></u-
  </view>
  <view style="margin-bottom: 10px;">
    <u-button type="primary" text="获取当前电子秤的计量类型" @click="getScaleType">
  </view>
  <view style="margin-bottom: 10px;">
    <u-button type="primary" text="释放资源" @click="release"></u-button>
  </view>
</view>
</template>

<script>
import * as module from "@uni_modules/leven-uts-sanmi-scale"
export default {
  data() {
    return {}
  },
  onLoad() {

  },
  methods: {
    //释放资源
    release() {
      module.release(res => {
        console.log(JSON.stringify(res))
      })
    },
    //获取当前电子秤的计量类型
    getScaleType() {
      module.getScaleType(res => {
        console.log(JSON.stringify(res))
      })
    },
    //获取当前价格计算的重量单位
    getUnit() {
      module.getUnit(res => {
        console.log(JSON.stringify(res))
      })
    },
    //设置价格计算时的重量单位
    setUnit() {
      module.setUnit({
        //计量单位, 0 : 按g计重, 1 : 按100g计重, 2 : 按500g计重, 3 : 按kg计重
        unit: 1
      }, res => {
        console.log(JSON.stringify(res))
      })
    },
    //获取当前已经设置的单价
    getUnitPrice() {

```

```

    module.getUnitPrice(res => {
      console.log(JSON.stringify(res))
    })
  },
  //设置单价
  setUnitPrice() {
    module.setUnitPrice({
      //单价
      unitPrice: "1.5"
    }, res => {
      console.log(JSON.stringify(res))
    })
  },
  //重启电子秤
  restart() {
    module.restart(res => {
      console.log(JSON.stringify(res))
    })
  },
  //读取电子秤参数信息
  getCalInfo() {
    module.getCalInfo(res => {
      console.log(JSON.stringify(res))
    })
  },
  //读取标定按钮开关状态
  getCalStatus() {
    module.getCalStatus(res => {
      console.log(JSON.stringify(res))
    })
  },
  //获取铅封状态
  readSealState() {
    module.readSealState(res => {
      console.log(JSON.stringify(res))
    })
  },
  //读取加速度数据
  readAcceleData() {
    module.readAcceleData(res => {
      console.log(JSON.stringify(res))
    })
  },
  //数字去皮
  digitalTare() {
    module.digitalTare({
      //去皮的重量
      tare: 500
    }, res => {
      console.log(JSON.stringify(res))
    })
  }
}

```

```
    })
  },
  //去皮
  tare() {
    module.tare(res => {
      console.log(JSON.stringify(res))
    })
  },
  //清零
  zero() {
    module.zero(res => {
      console.log(JSON.stringify(res))
    })
  },
  //获取固件版本号
  getFirmwareVersion() {
    module.getFirmwareVersion(res => {
      console.log(JSON.stringify(res))
    })
  },
  //获取电子秤服务版本号
  getServiceVersion() {
    module.getServiceVersion(res => {
      console.log(JSON.stringify(res))
    })
  },
  //取消称重监听
  cancelScaleListener() {
    module.cancelScaleListener(res => {
      console.log(JSON.stringify(res))
    })
  },
  //称重监听
  scaleListener() {
    module.scaleListener(res => {
      console.log(JSON.stringify(res))
    })
  },
  //断开连接
  disconnect() {
    module.disconnect(res => {
      console.log(JSON.stringify(res))
    })
  },
  //初始化
  init() {
    module.init(res => {
      console.log(JSON.stringify(res))
    })
  }
}
```

使用方法

```
    }  
  }  
</script>  
  
<style>  
</style>
```

更新日志

2025-03-10

首次发布

初始化

方法名

init

使用插件前需先初始化

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "status": "Connect"
  },
  "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	初始化结果，Connect：连接成功，Disconnect：断开连接

初始化

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

断开连接

方法名

disconnect

用法

- 用法如下：

```
module.disconnect(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

称重监听

方法名

监听只需要调用一次即可，会持续返回数据

用法

- 用法如下：

- 参数说明

无

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
data.type	String	类别，successListener：监听成功，scaleResult：称重结果，scaleStatus：称重状态，scalePrice：称重价格
data.net	Integer	称量净重，单位：克
data.tare	Integer	称量皮重，单位：克
data.isStable	Boolean	秤稳定状态，true:稳定 false：浮动
data.isLightWeight	Boolean	秤是否过轻，true:过轻，false:正常

参数名	参数类型	参数描述
data.overload	Boolean	秤是否过载，true:过载，false:正常
data.clearZeroErr	Boolean	秤是否有清零错误，true:错误，false:正常
data.calibrationErr	Boolean	秤是否有标定错误，true:错误，false:正常
data.unit	Integer	重量单位，0：g（默认），1：100g，2：500g，3：kg
data.unitPrice	String	当前计价设置的单价,默认为0，无单位（由业务方自定义）
data.totalPrice	String	电子秤计算后的总价
code	Integer	返回类型，0.成功，其他：失败

取消称重监听

方法名

cancelScaleListener

用法

- 用法如下：

```
module.cancelScaleListener(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

获取电子秤服务版本号

- 用法如下：

- 参数说明

无

- 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
data.version	String	版本号
code	Integer	返回类型，0.成功，其他：失败

获取固件版本号

方法名

getFirmwareVersion

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "version": "1.0"
  },
  "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.version	String	版本号
code	Integer	返回类型，0.成功，其他：失败

清零

方法名

zero

用法

- 用法如下：

```
module.zero(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

去皮

方法名

tare

用法

- 用法如下：

```
module.tare(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

数字去皮

方法名

digitalTare

用法

- 用法如下：

```
module.digitalTare({
  //去皮的重量
  tare: 500
}, 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
tare	Integer	是	无	去皮重量

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

读取加速度数据

方法名

readAcceleData

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "value": [  
      1,  
      2,  
      3  
    ]  
  },  
  "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

参数名	参数类型	参数描述
data.value	Array[Integer]	[0][1][2]分别为 X , Y , Z方向数据
code	Integer	返回类型 , 0.成功 , 其他 : 失败

获取铅封状态

方法名

readSealState

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "state": 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.state	Integer	铅封状态，0：正常 1：铅封被破坏
code	Integer	返回类型，0.成功，其他：失败

读取标定按钮开关状态

方法名

getCalStatus

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "state": 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.state	Integer	按钮开关状态，0：未按下 1：按下
code	Integer	返回类型，0.成功，其他：失败

读取电子秤参数信息

方法名

getCalInfo

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "info": [[6, 2],[15,5]]
  },
  "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.info	Array[Array[Integer]]	返回值为一个多个量程的二维数组例如量程为6/15kg e=2/5g 多量程电子秤将返回[[6, 2], [15,5]]

读取电子秤参数信息

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

重启电子秤

方法名

restart

用法

- 用法如下：

```
module.restart(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

设置单价

方法名

setUnitPrice

用法

- 用法如下：

```
module.setUnitPrice({
  //单价
  unitPrice: "1.5"
}, 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
unitPrice	String	是	无	单价

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

获取当前已经设置的单价

方法名

getUnitPrice

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "unitPrice": "1.5"
  },
  "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.unitPrice	String	单价
code	Integer	返回类型，0.成功，其他：失败

设置价格计算时的重量单位

方法名

setUnit

用法

- 用法如下：

```
module.setUnit({
  //计量单位，0：按g计重，1：按100g计重，2：按500g计重，3：按kg计重
  unit: 1
}, 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
unit	Integer	是	无	计量单位，0：按g计重，1：按100g计重，2：按500g计重，3：按kg计重

回调

- 示例

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

- 回调说明：

设置价格计算时的重量单位

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

获取当前价格计算的重量单位

方法名

getUnit

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "unit": 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.unit	Integer	计量单位，0：按g计重，1：按100g计重，2：按500g计重，3：按kg计重
code	Integer	返回类型，0.成功，其他：失败

获取当前电子秤的计量类型

方法名

getScaleType

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "scaleType": 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.scaleType	Integer	计量类型，0：公斤秤，1：磅秤
code	Integer	返回类型，0.成功，其他：失败

释放资源

方法名

release

用法

- 用法如下：

```
module.release(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