

uniapp 安卓UVC/USB多摄像头UTS原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
API	2
清除缓存	2.1
请求usb权限 v1.1.0	2.2
获取摄像头列表 v1.1.0	2.3
申请插件所需权限 v1.1.0	2.4
监听usb状态 v1.1.2	2.5
摄像头组件	3
初始化配置	3.1
方法	3.2
打开摄像头	3.2.1
关闭摄像头	3.2.2
开启预览	3.2.3
关闭预览	3.2.4
开始录像	3.2.5
结束录像	3.2.6
拍照	3.2.7
获取摄像头列表	3.2.8
切换摄像头	3.2.9
全屏	3.2.10
退出全屏	3.2.11
获取当前摄像头亮度百分比	3.2.12
设置当前摄像头亮度百分比	3.2.13
获取当前摄像头的对比度百分比	3.2.14
设置当前摄像头对比度百分比	3.2.15
设置镜像	3.2.16
获取支持的分辨率	3.2.17
获取摄像头当前分辨率	3.2.18
监听按钮	3.2.19
开始推流	3.2.20
停止推流	3.2.21
事件	3.3

摄像头加载成功	3.3.1
摄像头关闭	3.3.2
摄像头打开成功	3.3.3
摄像头被取消	3.3.4
摄像头被卸载	3.3.5
设备被关闭	3.3.6
设备被打开	3.3.7
错误事件	3.3.8
退出全屏	3.3.9
录像结果	3.3.10
拍照结果	3.3.11

使用方法

介绍

安卓UVC/USB多摄像头UTS原生插件集成了同一个组件可在一个页面多次加载不同摄像头，插件支持自定义摄像头排版，同时集成了推流，录像、拍照、设置镜像、亮度调节，摄像头设备按钮监听等功能，拍照可自定义水印，支持推流的同时摄像头录像

使用建议

在app启动的时候可以先调用API的清除缓存功能，因为插件中使用了持久性缓存，有可能在app异常退出的时候加载的摄像头在缓存中导致摄像头再次加载时失败

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=24792>

权限

1. android.permission.CAMERA
2. android.hardware.camera
3. android.hardware.camera.autofocus

使用方法

4. android.permission.WRITE_EXTERNAL_STORAGE
5. android.permission.READ_EXTERNAL_STORAGE
6. android.hardware.usb.host

演示程序下载

无

用法

API

插件使用

在使用插件的地方引入以下代码：

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

页面内容

```
<template>
  <view>
    <button type="primary" @click="clearCache">清除缓存</button>
  </view>
</template>

<script>
  // const module = uni.requireNativePlugin("leven-utsUvcCamare-CamareModule");
  import * as module from "@uni_modules/leven-uts-uvcCamera"
  export default {
    data() {
      return {

      }
    },
    methods: {
      clearCache() {
        module.clearCache(res => {
          console.log(res);
        })
      }
    }
  }
}
```

```

    })
  }
}
}
</script>

<style>
</style>

```

摄像头组件

插件使用

在使用插件的地方引入以下代码：

```

<leven-uts-uvCamera ref="refLevenUvcCamera" style="flex:1; height: 500px; margin:
  @onAttach="onAttach" @onCameraOpen="onCameraOpen" @onCameraClose="onCameraC
  @onDetach="onDetach" @onDeviceClose="onDeviceClose" @onDeviceOpen="onDeviceC
  @onQuitFullScreen="onQuitFullScreen" @onRecordingResult="onRecordingResult"
  @onTackPicture="onTackPicture"></leven-uts-uvCamera>

```

页面内容

```

<template>
  <view>
    <leven-uts-uvCamera ref="refLevenUvcCamera" style="flex:1; height: 500px; ma
      @onAttach="onAttach" @onCameraOpen="onCameraOpen" @onCameraClose="onCameraC
      @onDetach="onDetach" @onDeviceClose="onDeviceClose" @onDeviceOpen="onDeviceC
      @onQuitFullScreen="onQuitFullScreen" @onRecordingResult="onRecordingResult"
      @onTackPicture="onTackPicture"></leven-uts-uvCamera>
    <view>
      <button type="primary" @click="open">开启摄像头</button>
      <button type="primary" @click="close">关闭摄像头</button>
      <button type="primary" @click="startPreview">开启预览</button>
      <button type="primary" @click="stopPreview">关闭预览</button>
      <button type="primary" @click="startRecording">开始录像</button>
      <button type="primary" @click="stopRecording">结束录像</button>
      <button type="primary" @click="takePicture">拍照</button>
      <button type="primary" @click="getDeviceList">获取摄像头列表</button>
      <button type="primary" @click="switchCamera">切换摄像头</button>
      <button type="primary" @click="fullScreen">全屏</button>
    </view>
  </view>

```

```

<button type="primary" @click="getBrightnessPercent">获取摄像头亮度百分比</button>
<button type="primary" @click="setBrightnessPercent">设置摄像头亮度百分比</button>
<button type="primary" @click="getContrastPercent">获取摄像头的对比度百分比</button>
<button type="primary" @click="setContrastPercent">设置摄像头对比度百分比</button>
<button type="primary" @click="setMirror">设置镜像</button>
<button type="primary" @click="getPreviewSize">获取预览分辨率</button>
<button type="primary" @click="getSupportedSizeList">获取支持的分辨率</button>
<button type="primary" @click="setButtonCallback">监听按钮</button>
</view>
</view>
</template>

<script>
export default {
  data() {
    return {
      config: {
        //视图的圆角值
        radius: 10,
        //摄像头旋转角度
        rotation: 90,
        //图片文件保存目录
        pictureDir: "/storage/emulated/0/Pictures/LevenUvcCamera/",
        //视频保存目录
        videoDir: "/storage/emulated/0/Movies/LevenUvcCamera/",
        //默认打开的摄像头deviceName,如果不是通过这个打开默认摄像头请留空
        // deviceName: "/dev/bus/usb/003/003",
        //默认打开的摄像头产品名称
        // productName: "RXGD-CAMERA",
        //默认分辨率
        // size: {
        //   width: 800,
        //   height: 600
        // },
      },
      //摄像头列表
      deviceList: [],
      //当前摄像头的索引
      deviceIndex: 0,
      //镜像模式
      mirror: 0
    }
  },
  methods: {
    //事件
    onTackPicture(e) {
      console.log(e)
    },
    onRecordingResult(e) {
      console.log(e)
    }
  }
}

```

```
},
onQuitFullScreen(e) {
    console.log(e)
},
onError(e) {
    console.log(e)
},
onDeviceOpen(e) {
    console.log(e)
},
onDeviceClose(e) {
    console.log(e)
},
onDetach(e) {
    console.log(e)
},
onCancel(e) {
    console.log(e)
},
onCameraClose(e) {
    console.log(e)
},
onCameraOpen(e) {
    console.log(e)
},
onAttach(e) {
    console.log(e)
},
//按钮点击事件
setButtonCallback() {
    this.$refs.refLevenUvcCamera.setButtonCallback(res => {
        console.log(res)
    })
},
//获取支持的分辨率
getSupportedSizeList() {
    this.$refs.refLevenUvcCamera.getSupportedSizeList(res => {
        console.log(res)
    })
},
//获取摄像头的对比度百分比
getPreviewSize() {
    this.$refs.refLevenUvcCamera.getPreviewSize(res => {
        console.log(res)
    })
},
//设置镜像
setMirror() {
    if (this.$refs.refLevenUvcCamera) {
        this.mirror++;
    }
}
```

```

        if (this.mirror > 4) {
            this.mirror = 0;
        }
        this.$refs.refLevenUvcCamera.setMirror({
            //镜像, 0.正常, 1.垂直, 2.水平, 3.垂直+水平
            mirror: 4
        }, res => {
            this.writeLog(JSON.stringify(res))
        })
    },
    //设置摄像头对比度百分比
    setContrastPercent() {
        this.$refs.refLevenUvcCamera.setContrastPercent({
            percent: 30
        }, res => {
            console.log(res)
        })
    },
    //获取摄像头的对比度百分比
    getContrastPercent() {
        this.$refs.refLevenUvcCamera.getContrastPercent(res => {
            console.log(res)
        })
    },
    //设置当前摄像头亮度百分比
    setBrightnessPercent() {
        this.$refs.refLevenUvcCamera.setBrightnessPercent({
            percent: 30
        }, res => {
            console.log(res)
        })
    },
    //获取当前摄像头亮度百分比
    getBrightnessPercent() {
        this.$refs.refLevenUvcCamera.getBrightnessPercent(res => {
            console.log(res)
        })
    },
    //全屏
    fullScreen() {
        this.$refs.refLevenUvcCamera.fullScreen(res => {
            console.log(res)
        })
    },
    //切换摄像头
    switchCamera() {
        if (this.$refs.refLevenUvcCamera) {
            if (this.deviceList.length == 0) {
                this.showToast("请先获取摄像头列表")
            }
        }
    }
}

```

```

        return;
    }
    if (this.deviceIndex >= this.deviceList.length) {
        this.deviceIndex = 0;
    } else {
        this.deviceIndex++;
    }
    this.$refs.refLevenUvcCamera.switchCamera({
        //摄像头名称
        deviceName: this.deviceList[this.deviceIndex].deviceName
    }, res => {
        console.log(res)
    })
}
},
//获取摄像头列表
getDeviceList() {
    this.$refs.refLevenUvcCamera.getDeviceList(res => {
        console.log(res);
        if (res.code == 0 && res.data && res.data.deviceList && Array.isArray(res.data.deviceList)) {
            this.deviceList = res.data.deviceList;
        }
    })
},
//拍照
takePicture() {
    if (this.$refs.refLevenUvcCamera) {
        this.$refs.refLevenUvcCamera.takePicture({
            //水印列表，不传或为空不添加水印
            water: [{
                //水印内容
                text: "这是一条测试水印",
                //文本颜色，默认：白色
                color: "#FFFFFF",
                //文本大小，默认：20
                textSize: 40,
                //文本样式，NORMAL：常规（默认），BOLD：加粗，ITALIC：斜体，BOLD_ITALIC：斜体
                typeFace: "NORMAL",
                //是否使用抗锯齿功能，默认：true
                antiAlia: true,
                //透明度（0~255），默认：128
                alpha: 128,
                //水印位置，0：左上角（默认），1.右上角，2.左下角，3.右下角
                position: 0,
                //水印左边距，默认：40
                leftMargin: 40,
                //水印右边距，默认：40
                rightMargin: 40,
                //水印上边距，默认：40
                topMargin: 40,
            }],
        })
    }
}

```

```

        //水印下边距，默认：40
        bottomMargin: 40
    }]
  }, res => {
    console.log(res)
  })
}
},
//结束录像
stopRecording() {
  this.$refs.refLevenUvcCamera.stopRecording(res => {
    console.log(res)
  })
},
//开始录像
startRecording() {
  this.$refs.refLevenUvcCamera.startRecording(res => {
    console.log(res)
  })
},
//关闭预览
stopPreview() {
  this.$refs.refLevenUvcCamera.stopPreview(res => {
    console.log(res)
  })
},
//开启预览
startPreview() {
  this.$refs.refLevenUvcCamera.startPreview(res => {
    console.log(res)
  })
},
//关闭摄像头
close() {
  this.$refs.refLevenUvcCamera.close(res => {
    console.log(res)
  })
},
//开启摄像头
open() {
  this.$refs.refLevenUvcCamera.open(res => {
    console.log(res)
  })
},
showToast(title) {
  uni.showToast({
    icon: "none",
    title: title
  })
}
}

```

使用方法

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

更新日志

2025-10-23 v1.1.5

- [优化] 修复推流异常闪退问题
- [优化] 拍照新增参数 `base64` 是否返回base64数据

2025-10-13 v1.1.2

- [新增] API新增接口【[监听usb状态](#)】

2025-08-27 v1.1.0

- [新增] API新增接口【[请求usb权限](#)】
- [新增] API新增接口【[获取摄像头列表](#)】
- [新增] API新增接口【[申请插件所需权限](#)】

2025-08-17

首次发布

清除缓存

方法名

clearCache

用法

- 用法如下：

```
module.clearCache(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

请求usb权限 v1.1.0

方法名

requestUsbPermission

用法

- 用法如下：

```
module.requestUsbPermission({
  //摄像头名称
  deviceName: "",
}, res => {
  console.log(res)
})
```

- 参数说明
无

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

获取摄像头列表 v1.1.0

方法名

getDeviceList

v1.1.0

用法

- 用法如下：

```
module.getDeviceList(res => {  
  console.log(res)  
  if (res && res.data) {  
    let deviceList = res.data.deviceList;  
    if (Array.isArray(deviceList) && deviceList.length > 0) {  
      this.deviceName = deviceList[0].deviceName;  
    }  
  }  
})
```

- 参数说明
无

回调

- 示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "deviceList": [{  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }]  
  }  
}
```

- 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.deviceList	Array	摄像头数据集合
data.deviceList.productId	Integer	产品id
data.deviceList.productName	String	产品名称
data.deviceList.serialNumber	String	串口号
data.deviceList.deviceName	String	设备名称
data.deviceList.vendorId	Integer	厂家id
data.deviceList.deviceId	Integer	设备id
code	Integer	返回类型，0.成功，其他：失败

申请插件所需权限 v1.1.0

方法名

requestPermissions v1.1.0

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "权限被拒绝",
  "code": -202
}
```

- 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，-200：权限未全部申请，-201：权限被永久拒绝，-202：权限被拒绝

监听usb状态 v1.1.2

方法名

usbStatusListener v1.1.2

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "device": {
      "productName": "Web Camera",
      "deviceId": 1002,
      "deviceName": "/dev/bus/usb/001/002",
      "vendorId": 7119,
      "productId": 8833
    },
    "type": "onAttached"
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

参数名	参数类型	参数描述
data	Object	数据对象
data.type	String	当前状态，onListener：监听成功，onAttached：USB插入，onDetached：USB拔出，onPermission：申请usb权限成功
data.device	Object	当前usb设备数据
data.device.productName	String	产品名称
data.device.deviceId	Integer	设备id
data.device.deviceName	String	设备名称
data.device.vendorId	Integer	型号id
data.device.productId	Integer	产品id
code	Integer	返回类型，0.成功，其他：失败

初始化配置

config

组件初始化配置

参数说明

参数名	参数类型	是否必填	默认值	参数描述
radius	Integer	否	0	摄像头预览圆角
rotation	Integer	否	0	摄像头旋转角度
pictureDir	String	否	项目目录下	图片保存目录
videoDir	String	否	项目目录下	录像保存目录
deviceName	String	否	无	默认打开的摄像头 路径， 如：/dev/bus/us b/003/003
productName	String	否	无	默认打开的摄像头 产品名称，如： RXGD-CAMERA
size	Object	否	无	默认分辨率
size .width	Integer	否	无	分辨率的宽
size .height	Integer	否	无	分辨率的高
size .fps	Integer	否	30	帧率

方法

打开摄像头

方法名

open

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.open(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

关闭摄像头

方法名

close

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.close(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

开启预览

方法名

startPreview

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.startPreview(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

关闭预览

方法名

stopPreview

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.stopPreview(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

开始录像

方法名

startRecording

在停止录像后，录像文件的路径会在【录像结果】事件中返回

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.startRecording(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

结束录像

方法名

stopRecording

在停止录像后，录像文件的路径会在【录像结果】事件中返回

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.stopRecording(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

拍照

方法名

takePicture

拍照图片路径在【[拍照结果](#)】事件中返回

用法

- 用法如下：

```
this.$refs.refLevenUvcCamera.takePicture({
  //是否返回base64数据，默认：false
  base64: true,
  //水印列表
  water: [{
    //水印内容
    text: "这是一条测试水印",
    //文本颜色，默认：白色
    color: "#FFFFFF",
    //文本大小，默认：20
    textSize: 40,
    //文本样式，NORMAL：常规（默认），BOLD：加粗，ITALIC：斜体，BOLD_ITALIC：斜体加粗
    typeFace: "NORMAL",
    //是否使用抗锯齿功能，默认：true
    antiAlia: true,
    //透明度（0~255），默认：128
    alpha: 128,
    //水印位置，0：左上角（默认），1.右上角，2.左下角，3.右下角
    position: 0,
    //水印左边距，默认：40
    leftMargin: 40,
    //水印右边距，默认：40
    rightMargin: 40,
    //水印上边距，默认：40
    topMargin: 40,
    //水印下边距，默认：40
    bottomMargin: 40
  }]
}, res => {
  console.log(res)
})
```

拍照

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
base64	Boolean	否	false	是否返回base64数据
water	Array[Object]	是	无	水印列表
water.text	String	是	无	水印文本内容
water.color	String	否	#FFFFFF	文本颜色
water.textSize	Integer	否	20	文本大小
water.typeFace	String	否	NORMAL	文本样式， NORMAL：常规 (默认)，BOLD：加粗，ITALIC：斜体， BOLD_ITALIC：斜体加粗
water.antiAlia	Boolean	否	true	是否使用抗锯齿功能
water.alpha	Integer	否	128	透明度 (0~255)
water.position	Integer	否	0	水印位置，0：左上角(默认)，1.右上角，2.左下角，3.右下角
water.leftMargin	Integer	否	40	水印左边距
water.rightMargin	Integer	否	40	水印右边距
water.topMargin	Integer	否	40	水印上边距
water.bottomMargin	Integer	否	40	水印下边距

回调

- 示例

拍照

```
{
  "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

获取摄像头列表

方法名

getDeviceList

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {  
  this.$refs.refLevenUvcCamera.getDeviceList(res => {  
    console.log(res);  
  })  
}
```

- 参数说明
无

回调

- 示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "deviceList": [{  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }]  
  }  
}
```

- 回调说明：

获取摄像头列表

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.deviceList	Array	摄像头数据集合
data.deviceList.productId	Integer	产品id
data.deviceList.productName	String	产品名称
data.deviceList.serialNumber	String	串口号
data.deviceList.deviceName	String	设备名称
data.deviceList.vendorId	Integer	厂家id
data.deviceList.deviceId	Integer	设备id
code	Integer	返回类型，0.成功，其他：失败

切换摄像头

方法名

switchCamera

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.switchCamera({
    //摄像头名称
    deviceName: this.deviceList[this.deviceIndex].deviceName
  }, res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

全屏

方法名

fullScreen

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.fullScreen(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

退出全屏

方法名

quitFullScreen

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.quitFullScreen(res => {
    console.log(res)
  })
}
```

- 参数说明
无

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

获取当前摄像头亮度百分比

方法名

getBrightnessPercent

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.getBrightnessPercent(res => {
    console.log(res)
  })
}
```

- 参数说明
无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "percent": 50
  }
}
```

- 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.percent	Integer	亮度，范围：[0,100]
code	Integer	返回类型，0.成功，其他：失败

设置当前摄像头亮度百分比

方法名

setBrightnessPercent

用法

- 用法如下：

```
this.$refs.refLevenUvcCamera.setBrightnessPercent({
  //亮度百分比
  "percent": 70
}, 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
percent	Integer	是	无	亮度百分比，取值范围：[0,100]

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

设置当前摄像头亮度百分比

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

获取当前摄像头的对比度百分比

方法名

getContrastPercent

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.getContrastPercent(res => {
    console.log(res)
  })
}
```

- 参数说明
无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "percent": 50
  }
}
```

- 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.percent	Integer	对比度，范围：[0,100]
code	Integer	返回类型，0.成功，其他：失败

设置当前摄像头对比度百分比

方法名

setContrastPercent

用法

- 用法如下：

```
this.$refs.refLevenUvcCamera.setContrastPercent({
  //对比度百分比
  "percent": 70
}, 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
percent	Integer	是	无	对比度百分比， 取值范围： [0,100]

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

设置当前摄像头对比度百分比

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

设置镜像

方法名

setMirror

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.setMirror({
    //镜像，0.正常，1.垂直，2.水平
    mirror: 1
  }, 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
mirror	Integer	无	0	镜像方式，0.正常，1.垂直镜像，2.水平镜像

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

获取支持的分辨率

方法名

getSupportedSizeList

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {  
  this.$refs.refLevenUvcCamera.getSupportedSizeList(res => {  
    this.writeLog(JSON.stringify(res))  
  })  
}
```

- 参数说明
无

回调

- 示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "list": [  
      {  
        "type": 7,  
        "fps": 30,  
        "height": 720,  
        "width": 1280,  
        "fpsList": [  
          30  
        ]  
      },  
      {  
        "type": 7,  
        "fps": 30,  
        "height": 120,  
        "width": 160,  
      }  
    ]  
  }  
}
```

```
    "fpsList": [
      30
    ]
  },
  {
    "type": 7,
    "fps": 30,
    "height": 144,
    "width": 176,
    "fpsList": [
      30
    ]
  },
  {
    "type": 7,
    "fps": 30,
    "height": 240,
    "width": 320,
    "fpsList": [
      30
    ]
  },
  {
    "type": 7,
    "fps": 30,
    "height": 288,
    "width": 352,
    "fpsList": [
      30
    ]
  },
  {
    "type": 7,
    "fps": 30,
    "height": 480,
    "width": 848,
    "fpsList": [
      30
    ]
  },
  {
    "type": 7,
    "fps": 30,
    "height": 600,
    "width": 800,
    "fpsList": [
      30
    ]
  },
  {
```

获取支持的分辨率

```
    "type": 7,
    "fps": 30,
    "height": 540,
    "width": 960,
    "fpsList": [
      30
    ]
  },
  {
    "type": 7,
    "fps": 30,
    "height": 720,
    "width": 1280,
    "fpsList": [
      30
    ]
  },
  {
    "type": 7,
    "fps": 30,
    "height": 480,
    "width": 640,
    "fpsList": [
      30
    ]
  }
]
```

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.list	Array[Object]	列表
data.list.type	Integer	类型
data.list.fps	Integer	帧率
data.list.height	Integer	高度
data.list.width	Integer	宽度
data.list.fpsList	Array[Integer]	帧率列表
code	Integer	返回类型，0.成功，其他：失败

获取摄像头当前分辨率

方法名

getPreviewSize

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.getPreviewSize(res => {
    console.log(res)
  })
}
```

- 参数说明
无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "type": 7,
    "fps": 30,
    "height": 600,
    "width": 800,
    "fpsList": [
      30
    ]
  }
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

参数名	参数类型	参数描述
data	Object	数据对象
data.type	Integer	类型
data.fps	Integer	帧率
data.height	Integer	高度
data.width	Integer	宽度
data.fpsList	Array[Integer]	帧率列表
code	Integer	返回类型，0.成功，其他：失败

监听按钮

方法名

setButtonCallback

用法

- 用法如下：

```
if (this.$refs.refLevenUvcCamera) {
  this.$refs.refLevenUvcCamera.setButtonCallback(res => {
    console.log(res)
  })
}
```

- 参数说明
无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "button": 1,
    "status": 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
data.button	Integer	按钮标识
data.status	Integer	状态

监听按钮

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

开始推流

方法名

startPusher

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.startPusher({
    // 推流地址
    url: "rtmp://195964.push.tliveloud.com/live/leven?txSecret=e387f3e5f5"
  })
}
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	推流地址

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

参数名	参数类型	参数描述
data.status	String	推流状态，onConnecting：服务器连接中，onConnectSuccess：服务器连接成功，onConnectFailed：服务器连接失败，onPushing：推流中，onStopPushing：停止推流
code	Integer	返回类型，0.成功，其他：失败

停止推流

方法名

stopPusher

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {  
  this.$refs.refLevenUsbCamera.stopPusher(res=>{})  
}
```

- 参数说明

无

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

事件

摄像头加载成功

方法名

onAttach

用法

- 用法如下：

```
onAttach(e) {  
    console.log("onDestroy:" + JSON.stringify(e))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：

参数名	参数类型	参数描述
device	Object	当前摄像头信息
device.productId	Integer	产品id
device.productName	String	产品名称

摄像头加载成功

参数名	参数类型	参数描述
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id
deviceId.deviceId	Integer	设备id

摄像头关闭

方法名

onCameraClose

用法

- 用法如下：

```
onCameraClose(e) {
  console.log("onCameraClose:" + JSON.stringify(e))
}
```

- 参数说明

无

回调

- 示例

```
{
  "device": {
    "productId": 12337,
    "productName": "RXGD-CAMERA",
    "serialNumber": "200901010001",
    "deviceName": "/dev/bus/usb/003/003",
    "vendorId": 3034,
    "deviceId": 3003
  }
}
```

- 回调说明：

参数名	参数类型	参数描述
device	Object	当前摄像头信息
device.productId	Integer	产品id
device.productName	String	产品名称

参数名	参数类型	参数描述
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id
deviceId.deviceId	Integer	设备id

摄像头打开成功

方法名

onCameraOpen

用法

- 用法如下：

```
onCameraOpen(e) {  
    console.log("onCameraOpen:" + JSON.stringify(e))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：

参数名	参数类型	参数描述
device	Object	当前摄像头信息
device.productId	Integer	产品id
device.productName	String	产品名称

摄像头打开成功

参数名	参数类型	参数描述
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id
deviceId.deviceId	Integer	设备id

摄像头被取消

方法名

onCancel

用法

- 用法如下：

```
onCancel(e) {  
    console.log("onCancel:" + JSON.stringify(e))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：

参数名	参数类型	参数描述
device	Object	当前摄像头信息
device.productId	Integer	产品id
device.productName	String	产品名称

摄像头被取消

参数名	参数类型	参数描述
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id
deviceId.deviceId	Integer	设备id

摄像头被卸载

方法名

onDetach

用法

- 用法如下：

```
onDetach(e) {  
    console.log("onDetach:" + JSON.stringify(e))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：

参数名	参数类型	参数描述
device	Object	当前摄像头信息
device.productId	Integer	产品id
device.productName	String	产品名称

摄像头被卸载

参数名	参数类型	参数描述
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id
deviceId.deviceId	Integer	设备id

设备被关闭

方法名

onDeviceClose

用法

- 用法如下：

```
onDeviceClose(e) {  
    console.log("onDeviceClose:" + JSON.stringify(e))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：

参数名	参数类型	参数描述
device	Object	当前摄像头信息
device.productId	Integer	产品id
device.productName	String	产品名称

设备被关闭

参数名	参数类型	参数描述
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id
deviceId.deviceId	Integer	设备id

设备被打开

方法名

onDeviceOpen

用法

- 用法如下：

```
onDeviceOpen(e) {  
    console.log("onDeviceOpen:" + JSON.stringify(e))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：

参数名	参数类型	参数描述
device	Object	当前摄像头信息
device.productId	Integer	产品id
device.productName	String	产品名称

设备被打开

参数名	参数类型	参数描述
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id
deviceId.deviceId	Integer	设备id

错误事件

- 用法如下：
- 参数说明

无
- 示例

退出全屏

方法名

onQuitFullScreen

用法

- 用法如下：

```
onQuitFullScreen(e) {  
    console.log("onQuitFullScreen:" + JSON.stringify(e))  
}
```

- 参数说明

无

回调

- 示例

```
{}
```

- 回调说明：
无

录像结果

方法名

onRecordingResult

用法

- 用法如下：

```
onRecordingResult(e) {  
    let detail = e.detail;  
    console.log("onError:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

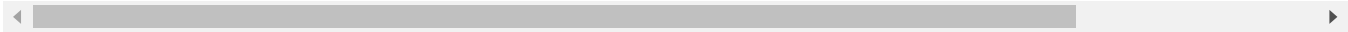
```
{  
  "type": "onRecording",  
  "timeStamp": 1709286770395,  
  "target": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "currentTarget": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "detail": {  
    "path": "/storage/emulated/0/Android/data/test.leven.uniplugin.com/video/1
```

录像结果

```
}  
}
```

- 回调说明：

参数名	参数类型	参数描述
path	String	录像文件路径



拍照结果

方法名

onTackPicture

用法

- 用法如下：

```
onTackPicture(e) {  
    let detail = e.detail;  
    console.log("onError:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "type": "onTackPicture",  
  "timeStamp": 1709286770395,  
  "target": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "currentTarget": {  
    "id": "",  
    "dataset": {},  
    "offsetLeft": 0,  
    "offsetTop": 0  
  },  
  "detail": {  
    "path": "/storage/emulated/0/Android/data/test.leven.uniplugin.com/image/1
```

拍照结果

```
}  
}
```

- 回调说明：

参数名	参数类型	参数描述
path	String	文件路径
base64 v1.1.5	String	图片的base64数据

