

uniapp 安卓USB摄像头插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
常见问题	1.3
插件方法	2
切换摄像头	2.1
开始拍照	2.2
关闭预览	2.3
开启预览	2.4
开始录制	2.5
结束录制	2.6
设置分辨率	2.7
获取支持的分辨率	2.8
关闭摄像头 v1.3.0	2.9
开启摄像头 v1.3.0	2.10
释放资源 v1.3.0	2.11
监听摄像头状态 v1.4.0	2.12
插件属性	3
设置分辨率	3.1
配置 v1.4.2	3.2
插件事件	4
事件返回参数说明	4.1
卸载事件	4.2
插入事件	4.3
拔出事件	4.4
连接事件	4.5
断开连接事件	4.6
取消连接事件	4.7
错误事件	4.8

使用方法

介绍

安卓USB摄像头，支持拍照，录像，停止预览，开启预览，支持多摄像头切换

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=10835>

示例文件

演示程序

用法

在需要使用插件的页面加载以下代码

```
<leven-usbCamera ref="refLevenUsbCamera" style="flex:1; height: 300px;" @onDestroy:  
@onConnect="onConnect" @onDisconnect="onDisconnect" @onCancel="onCancel" @onError=
```

```
</leven-usbCamera>
```

插件仅支持nvue页面

页面内容

```
<template>
  <view>
    <uni-card title="usb拍照">
      <leven-usbCamera ref="refLevenUsbCamera" style="flex:1; height: 300px;" @onl
        @onDettach="onDettach" @onConnect="onConnect" @onDisconnect="onDisconnect"
      </leven-usbCamera>
      <button type="primary" @click="changeCamera">切换摄像头</button>
      <button type="primary" @click="capture">开始拍照</button>
      <button type="primary" @click="stopPreview">关闭预览</button>
      <button type="primary" @click="openPreview">开启预览</button>
      <button type="primary" @click="startRecord">开始录制</button>
      <button type="primary" @click="stopRecord">结束录制</button>
      <button type="primary" @click="setSize">设置分辨率</button>
      <button type="primary" @click="getSupportedSize">获取支持的分辨率</button>
    </uni-card>
  </view>
</template>

<script>
  export default {
    data() {
      return {
        // 摄像头列表
        deviceList: [],
        // 当前预览的摄像头索引
        previewIndex: 0,
        // 当前分辨率的索引
        sizeIndex: 0
      }
    },
    methods: {
      // 切换摄像头
      changeCamera() {
        if (this.deviceList.length == 0) {
          this.showToast("设备列表为空");
          return false;
        }
        this.previewIndex++;
        let index = this.previewIndex % this.deviceList.length;
```

```
    if (this.$refs.refLevenUsbCamera) {
      this.$refs.refLevenUsbCamera.changeCamera({
        deviceName: this.deviceList[index].deviceName
      }, res => {
        console.log(res)
      })
    }
  },
  // 拍照
  capture() {
    if (this.$refs.refLevenUsbCamera) {
      this.$refs.refLevenUsbCamera.capture(res => {
        console.log(res)
      })
    }
  },
  // 关闭预览
  stopPreview() {
    if (this.$refs.refLevenUsbCamera) {
      this.$refs.refLevenUsbCamera.stopPreview(res => {
        console.log(res)
      })
    }
  },
  // 开启预览
  openPreview() {
    if (this.$refs.refLevenUsbCamera) {
      this.$refs.refLevenUsbCamera.openPreview(res => {
        console.log(res)
      })
    }
  },
  // 开始录制
  startRecord() {
    if (this.$refs.refLevenUsbCamera) {
      this.$refs.refLevenUsbCamera.startRecord(res => {
        console.log(res)
      })
    }
  },
  // 结束录制
  stopRecord() {
    if (this.$refs.refLevenUsbCamera) {
      this.$refs.refLevenUsbCamera.stopRecord(res => {
        console.log(res)
      })
    }
  },
  // 设置分辨率
  setSize() {
```

```

    if (this.$refs.refLevenUsbCamera) {
      let sizeList = [{
        width: 1280,
        height: 720
      }, {
        width: 640,
        height: 480
      }, {
        width: 800,
        height: 600
      }, {
        width: 848,
        height: 480
      }];
      this.sizeIndex++;
      let index = this.sizeIndex % sizeList.length;
      this.$refs.refLevenUsbCamera.setSize({
        width: sizeList[index].width,
        height: sizeList[index].height
      }, res => {
        console.log(res)
      })
    }
  },
  // 获取支持的分辨率
  getSupportedSize() {
    if (this.$refs.refLevenUsbCamera) {
      this.$refs.refLevenUsbCamera.getSupportedSize(res => {
        console.log(res)
      })
    }
  },
  // 组件卸载
  onDestory(e) {
    console.log("onDestroy:" + JSON.stringify(e))
  },
  // 组件加载完成
  onAttach(e) {
    let detail = e.detail;
    this.deviceList = detail.deviceList || [];
    console.log("onAttach:" + JSON.stringify(e.detail))
  },
  // 组件卸载
  onDettach(e) {
    let detail = e.detail;
    this.deviceList = detail.deviceList || [];
    console.log("onDettach:" + JSON.stringify(e.detail))
  },
  // 摄像机连接成功
  onConnect(e) {

```

```
    let detail = e.detail;
    this.deviceList = detail.deviceList || [];
    console.log("onConnect:" + JSON.stringify(e.detail))
  },
  // 断开连接
  onDisconnect(e) {
    let detail = e.detail;
    this.deviceList = detail.deviceList || [];
    console.log("onDisconnect:" + JSON.stringify(e.detail))
  },
  // 取消连接
  onCancel(e) {
    let detail = e.detail;
    this.deviceList = detail.deviceList || [];
    console.log("onCancel:" + JSON.stringify(e.detail))
  },
  // 系统错误
  onError() {
    let detail = e.detail;
    this.deviceList = detail.deviceList || [];
    console.log("onError:" + JSON.stringify(e.detail))
  },
  // 提示信息
  showToast(content) {
    uni.showToast({
      icon: "none",
      title: content
    })
  }
}
</script>

<style>

</style>
```

更新日志

2025-07-02 v1.4.5

- [优化] `config` 配置新增参数 `captureBase64` 拍照是否返回base64数据

2025-01-09 v1.4.4

- [优化] 修复1.4.3版本引出的打开默认相机异常错误问题

2025-01-02 v1.4.3

- [优化] 组件配置新增属性 `defaultDeviceName` （默认打开的摄像头）
- [优化] 组件配置新增属性 `startPreview` （组件加载后默认是否预览）

2024-08-26 v1.4.2

- [优化] 支持自定义文件保存目录

2024-07-12 v1.4.1

- [优化] 适配安卓4.4

2024-05-31 v1.4.0

- [新增] [监听摄像头状态](#)

2024-01-03 v1.3.3

- [优化] 修复异常闪退问题

2023-12-25 v1.3.2

- [新增] 组件加载时动态权限验证
- [优化] 拍照返回base64格式数据

2023-12-04 v1.3.1

- [优化] 优化页面返回后无法打开摄像头的问题

2023-11-30 v1.3.0

- [新增] 关闭摄像头【[转到](#)】
- [新增] 开启摄像头【[转到](#)】
- [新增] 释放资源【[转到](#)】

2023-10-12 v1.2.0

- [新增] 组件设置分辨率参数
- [新增] 设置分辨率api接口【[转到](#)】
- [新增] 获取支持的分辨率api接口【[转到](#)】

2023-10-08 v1.1.0

- [新增]新增多摄像头切换模式【[转到](#)】
- [新增]错误事件【[转到](#)】
- [优化]插件事件新增返回参数获取到的usb摄像头列表

2023-02-12

首次发布

常见问题

1. 使用demo里面的示例为什么会闪退

可能是你设置了不支持的分辨率导致闪退，你可以将demo示例中的size属性删除

2. 使用demo中的示例为什么会先 `onConnect` 然后 `onDisconnect`

可能是你设置了不支持的分辨率导致闪退，你可以将demo示例中的size属性删除

切换摄像头

方法名

changeCamera

切换摄像头是否成功需要在连接事件中查看

用法

- 用法如下：

```
if (this.deviceList.length == 0) {
  this.showToast("设备列表为空");
  return false;
}
this.previewIndex++;
let index = this.previewIndex % this.deviceList.length;
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.changeCamera({
    deviceName: this.deviceList[index].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	摄像头名称（路径）

回调

- 示例

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

切换摄像头

```
"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

开始拍照

方法名

capture

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.capture(res => {
    console.log(res)
  })
}
```

- 参数说明

无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "filePath": "/storage/emulated/0/Android/data/test.camera.com/images/2"
  }
}
```

- 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.filePath	String	拍照后的保存路径

开始拍照

参数名	参数类型	参数描述
data.fileBase64	String	拍照后的图片base64，是否返回该数据可在config中配置
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.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.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

开启预览

方法名

openPreview

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.openPreview(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

开始录制

方法名

startRecord

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.startRecord(res => {
    console.log(res)
  })
}
```

- 参数说明

无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "filePath": "/storage/emulated/0/Android/data/test.camera.com/video/20
  }
}
```

- 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.filePath	String	录制保存路径

开始录制

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

结束录制

方法名

stopRecord

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.stopRecord(res => {
    console.log(res)
  })
}
```

- 参数说明

无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "filePath": "/storage/emulated/0/Android/data/test.camera.com/video/20
  }
}
```

- 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.filePath	String	录制保存路径

结束录制

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

设置分辨率

- 用法如下：

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
width	Integer	是	无	宽度
height	Integer	是	无	高度

- 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

获取支持的分辨率

方法名

getSupportedSize

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {  
  this.$refs.refLevenUsbCamera.getSupportedSize(res => {  
    console.log(res)  
  })  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "list": [  
      {  
        "width": 1280,  
        "height": 720  
      },  
      {  
        "width": 160,  
        "height": 120  
      },  
      {  
        "width": 176,  
        "height": 144  
      },  
      {  
        "width": 1280,  
        "height": 720  
      }  
    ]  
  }  
}
```

获取支持的分辨率

```
        "width": 320,
        "height": 240
    },
    {
        "width": 352,
        "height": 288
    },
    {
        "width": 848,
        "height": 480
    },
    {
        "width": 800,
        "height": 600
    },
    {
        "width": 960,
        "height": 540
    },
    {
        "width": 1280,
        "height": 720
    },
    {
        "width": 640,
        "height": 480
    }
]
}
```

- 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	支持的分辨率列表
data.list.width	Integer	分辨率宽度
data.list.height	Integer	分辨率高度
code	Integer	返回类型，0.成功，其他：失败

关闭摄像头 v1.3.0

方法名

close v1.3.0

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.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

开启摄像头 v1.3.0

方法名

open v1.3.0

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.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

释放资源 v1.3.0

方法名

releaseResources v1.3.0

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.releaseResources(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

监听摄像头状态 v1.4.0

方法名

onListenerCameraStatus v1.4.0

监听usb摄像头状态，可用在vue3项目中，vue3项目不支持组件事件，所以可以调用该方法来监听摄像头的状态

用法

- 用法如下：

```
if (this.$refs.refLevenUsbCamera) {
  this.$refs.refLevenUsbCamera.onListenerCameraStatus(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
data.type	Object	状态类型，请参考下方说明

参数名	参数类型	参数描述
data.deviceList	Array	设备列表，请参考 事件返回参数说明
data.device	Object	当前设备
code	Integer	返回类型，0.成功，其他：失败

设置分辨率

方法名

size

用法

- 用法如下：

```
<leven-usbCamera ref="refLevenUsbCamera" :size="[800, 600]">
</leven-usbCamera>
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
size	Array	否	[1280, 720]	分辨率尺寸

回调

无

配置 v1.4.2

方法名

config

用法

- 用法如下：

```
<leven-usbCamera ref="refLevenUsbCamera" :config="config">
</leven-usbCamera>
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
config.savePath	String	否	无	自定义文件保存目录
config.defaultDeviceName v1.4.3	String	否	无	默认打开的usb摄像头，一般存在多个摄像头时可用此参数
config.startPreview v1.4.3	Boolean	否	true	组件加载后默认是否开启预览
config.captureBase64 v1.4.5	Boolean	否	true	拍照是否返回base64数据

回调

无

事件返回参数说明

事件参数说明

参数名	参数类型	参数描述
deviceList	Array	当前获取到的usb摄像头列表
deviceList.productId	Integer	产品id
deviceList.productName	String	产品名称
deviceList.serialNumber	String	串口号
deviceList.deviceName	String	设备名称
deviceList.vendorId	Integer	厂家id
deviceList.deviceId	Integer	设备id
usbControlBlock	Object	usb数据信息
usbControlBlock.deviceName	String	设备名称
usbControlBlock.usbVersion	String	usb版本
usbControlBlock.deviceKey	Integer	设备key
usbControlBlock.deviceKeyName	String	设备key名称
usbControlBlock.productName	String	产品名称
usbControlBlock.vendorId	Integer	厂家id
usbControlBlock.serial	String	串口号
usbControlBlock.version	String	设备版本
usbControlBlock.productId	Integer	产品id
usbControlBlock.deviceId	Integer	设备id
device	Object	设备信息
device.productId	Integer	产品id
device.productName	String	产品名称
device.serialNumber	String	串口号
device.deviceName	String	设备名称
device.vendorId	Integer	厂家id

事件返回参数说明

参数名	参数类型	参数描述
deviceId.deviceId	Integer	设备id

卸载事件

方法名

onDestroy

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{  
  "deviceList": [  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/001/008",  
      "vendorId": 3034,  
      "deviceId": 1008  
    },  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }  
  ],  
  "usbControlBlock": {
```

卸载事件

```
"deviceName": "/dev/bus/usb/003/003",
"usbVersion": "2.00",
"deviceKey": -1803267709,
"deviceKeyName": "3034#12337#239#2#1",
"productName": "RXGD-CAMERA",
"vendorId": 3034,
"serial": "200901010001",
"version": "0.01",
"productId": 12337,
"deviceId": 3003
},
"device": {
  "productId": 12337,
  "productName": "RXGD-CAMERA",
  "serialNumber": "200901010001",
  "deviceName": "/dev/bus/usb/003/003",
  "vendorId": 3034,
  "deviceId": 3003
}
}
```

- 回调说明：
事件返回参数请参考[事件返回参数说明](#)

插入事件

方法名

onAttach

用法

- 用法如下：

```
onAttach(e) {  
  let detail = e.detail;  
  this.deviceList = detail.deviceList || [];  
  console.log("onAttach:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "deviceList": [  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/001/008",  
      "vendorId": 3034,  
      "deviceId": 1008  
    },  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }  
  ]  
}
```

插入事件

```
],  
  "usbControlBlock": {  
    "deviceName": "/dev/bus/usb/003/003",  
    "usbVersion": "2.00",  
    "deviceKey": -1803267709,  
    "deviceKeyName": "3034#12337#239#2#1",  
    "productName": "RXGD-CAMERA",  
    "vendorId": 3034,  
    "serial": "200901010001",  
    "version": "0.01",  
    "productId": 12337,  
    "deviceId": 3003  
  },  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：
事件返回参数请参考[事件返回参数说明](#)

拔出事件

方法名

onDettach

用法

- 用法如下：

```
onDettach(e) {  
  let detail = e.detail;  
  this.deviceList = detail.deviceList || [];  
  console.log("onDettach:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "deviceList": [  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/001/008",  
      "vendorId": 3034,  
      "deviceId": 1008  
    },  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }  
  ]  
}
```

```
],  
  "usbControlBlock": {  
    "deviceName": "/dev/bus/usb/003/003",  
    "usbVersion": "2.00",  
    "deviceKey": -1803267709,  
    "deviceKeyName": "3034#12337#239#2#1",  
    "productName": "RXGD-CAMERA",  
    "vendorId": 3034,  
    "serial": "200901010001",  
    "version": "0.01",  
    "productId": 12337,  
    "deviceId": 3003  
  },  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：
事件返回参数请参考[事件返回参数说明](#)

连接事件

方法名

onConnect

用法

- 用法如下：

```
onConnect(e) {  
  let detail = e.detail;  
  this.deviceList = detail.deviceList || [];  
  console.log("onConnect:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "deviceList": [  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/001/008",  
      "vendorId": 3034,  
      "deviceId": 1008  
    },  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }  
  ]  
}
```

```
],  
"usbControlBlock": {  
  "deviceName": "/dev/bus/usb/003/003",  
  "usbVersion": "2.00",  
  "deviceKey": -1803267709,  
  "deviceKeyName": "3034#12337#239#2#1",  
  "productName": "RXGD-CAMERA",  
  "vendorId": 3034,  
  "serial": "200901010001",  
  "version": "0.01",  
  "productId": 12337,  
  "deviceId": 3003  
},  
"device": {  
  "productId": 12337,  
  "productName": "RXGD-CAMERA",  
  "serialNumber": "200901010001",  
  "deviceName": "/dev/bus/usb/003/003",  
  "vendorId": 3034,  
  "deviceId": 3003  
}  
}
```

- 回调说明：
事件返回参数请参考[事件返回参数说明](#)

断开连接事件

方法名

onDisconnect

用法

- 用法如下：

```
onDisconnect(e) {  
  let detail = e.detail;  
  this.deviceList = detail.deviceList || [];  
  console.log("onDisconnect:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "deviceList": [  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/001/008",  
      "vendorId": 3034,  
      "deviceId": 1008  
    },  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }  
  ]  
}
```

断开连接事件

```
],  
  "usbControlBlock": {  
    "deviceName": "/dev/bus/usb/003/003",  
    "usbVersion": "2.00",  
    "deviceKey": -1803267709,  
    "deviceKeyName": "3034#12337#239#2#1",  
    "productName": "RXGD-CAMERA",  
    "vendorId": 3034,  
    "serial": "200901010001",  
    "version": "0.01",  
    "productId": 12337,  
    "deviceId": 3003  
  },  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：
事件返回参数请参考[事件返回参数说明](#)

取消连接事件

方法名

onCancel

用法

- 用法如下：

```
onCancel(e) {  
  let detail = e.detail;  
  this.deviceList = detail.deviceList || [];  
  console.log("onCancel:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "deviceList": [  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/001/008",  
      "vendorId": 3034,  
      "deviceId": 1008  
    },  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }  
  ]  
}
```

取消连接事件

```
],
"usbControlBlock": {
  "deviceName": "/dev/bus/usb/003/003",
  "usbVersion": "2.00",
  "deviceKey": -1803267709,
  "deviceKeyName": "3034#12337#239#2#1",
  "productName": "RXGD-CAMERA",
  "vendorId": 3034,
  "serial": "200901010001",
  "version": "0.01",
  "productId": 12337,
  "deviceId": 3003
},
"device": {
  "productId": 12337,
  "productName": "RXGD-CAMERA",
  "serialNumber": "200901010001",
  "deviceName": "/dev/bus/usb/003/003",
  "vendorId": 3034,
  "deviceId": 3003
}
}
```

- 回调说明：
事件返回参数请参考[事件返回参数说明](#)

错误事件

方法名

onError

用法

- 用法如下：

```
onError(e) {  
  let detail = e.detail;  
  this.deviceList = detail.deviceList || [];  
  console.log("onError:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "deviceList": [  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/001/008",  
      "vendorId": 3034,  
      "deviceId": 1008  
    },  
    {  
      "productId": 12337,  
      "productName": "RXGD-CAMERA",  
      "serialNumber": "200901010001",  
      "deviceName": "/dev/bus/usb/003/003",  
      "vendorId": 3034,  
      "deviceId": 3003  
    }  
  ]  
}
```

```
],  
  "usbControlBlock": {  
    "deviceName": "/dev/bus/usb/003/003",  
    "usbVersion": "2.00",  
    "deviceKey": -1803267709,  
    "deviceKeyName": "3034#12337#239#2#1",  
    "productName": "RXGD-CAMERA",  
    "vendorId": 3034,  
    "serial": "200901010001",  
    "version": "0.01",  
    "productId": 12337,  
    "deviceId": 3003  
  },  
  "device": {  
    "productId": 12337,  
    "productName": "RXGD-CAMERA",  
    "serialNumber": "200901010001",  
    "deviceName": "/dev/bus/usb/003/003",  
    "vendorId": 3034,  
    "deviceId": 3003  
  }  
}
```

- 回调说明：
事件返回参数请参考[事件返回参数说明](#)