NOLO VR Android SDK

接口说明

北京凌宇智控科技有限公司

2017 年 5 月

目录

一、简介

NOLO VR Android SDK 是北京凌宇智控科技有限公司针对其 NOLO CV1 产品对外提供的接口说明，便于 APP 接入方通过该 SDK 获取 NOLO 设备数据。

二、SDK 接口说明

1. 接口详细说明

NOLO VR Android SDK 共有 13 个接口，每个接口的名称、原型、功能、参数和返回值如下表所示。

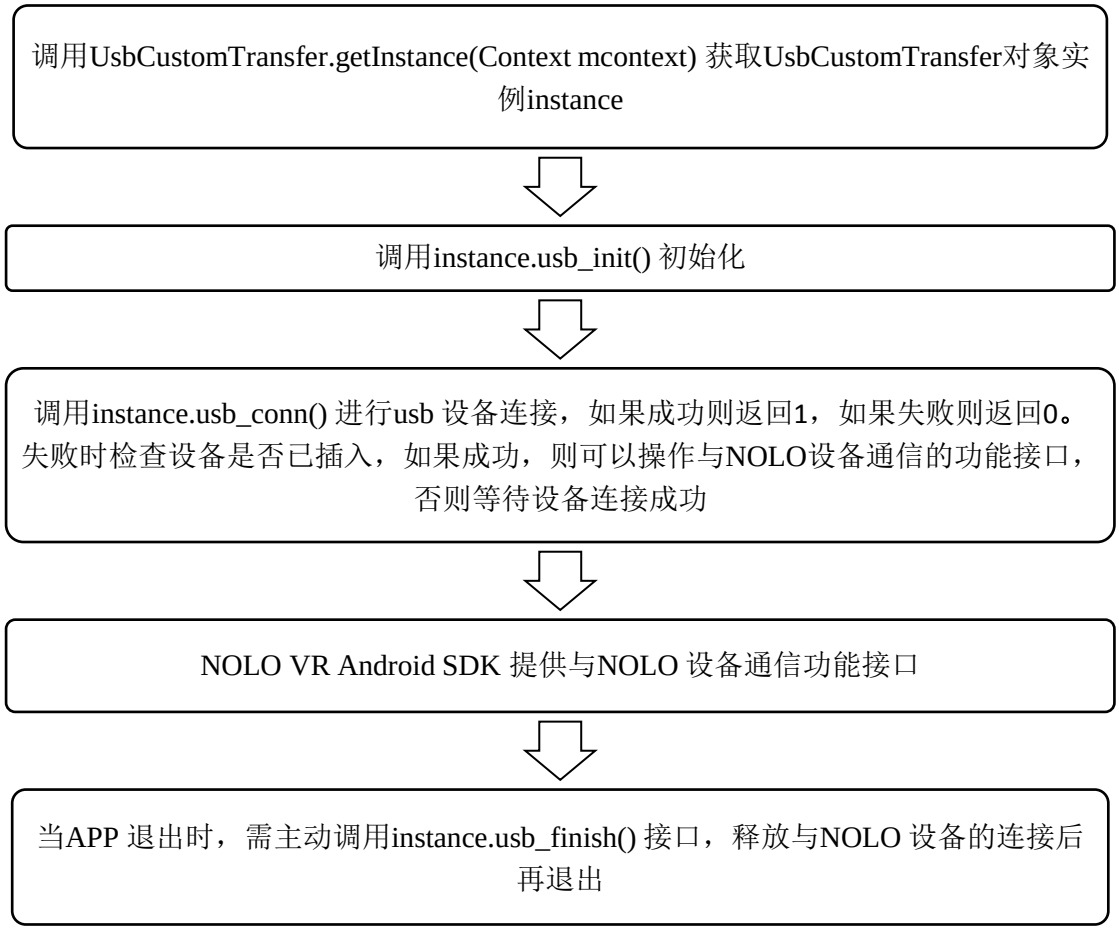
接口名称	详细说明	
获取 SDK 实例接口	原型	UsbCustomTransfer getInstance(Context context)
	功能	获得 UsbCustom Transfer 实例对象
	参数	Context：上下文环境
	返回值	UsbCustomTransfer 实例对象
SDK 初始化接口	原型	void usb_init()
	功能	初始化 SDK
	参数	无
	返回值	无
SDK 连接 NOLO 设备接口	原型	int usb_conn()
	功能	连接 NOLO 设备接口
	参数	无
	返回值	返回连接状态码， 0： 连接失败，1：连接成功
SDK 向 NOLO 设备发送数据接口	原型	void usb_sendData(byte[] mbyte)
	功能	向 NOLO 设备发送数据
	参数	mbyte: 待发送的数据，例如： Byte[4]: [

		0xAA(帧头第一位), 0x66(帧头第二位), 0x00(左手柄的震动强度，范围在(0x00~0x64)), 0x00(右手柄的震动强度，范围在(0x00~0x64))]
	返回值	无
设置接收断开通知接口	原型	Void setDisconnectedCallback(DisconnectedCallback mdis)
	功能	设置 NOLO 设备断开通知接口，DisconnectedCallback 的具体实现详见 Demo;
	参数	mdis: APP 接收 SDK 断开通知的对象
	返回值	无
SDK 断开 NOLO 设备接口	原型	void usb_finish()
	功能	断开与 NOLO 设备的连接
	参数	无
	返回值	无
获取 NOLO 设备版本号接口	原型	int getVersionByDeviceType(int type)
	功能	该接口获取 NOLO 设备的版本信息
	参数	参数 type 表示设备类型， 0：表示头盔；1：表示左手柄；2：表示右手柄； 3：表示基站
	返回值	设备的版本号 返回值 1：DK2 返回值 2：CV1
获取 NOLO 设备电量接口	原型	int getElectricityByDeviceType(int type)
	功能	获取 NOLO 设备的电量
	参数	参数 type 表示设备类型， 0：表示头盔；1：表示左手柄；2：表示右手柄； 3：表示基站
	返回值	NOLO 设备电量
获取 NOLO 设备连	原型	int getDeviceTrackingStatus(int type)

接状态接口	功能	获取 NOLO 设备的连接状态
	参数	参数 type 表示设备类型， 0：表示头盔；1：表示左手柄；2：表示右手柄； 3：表示基站
	返回值	NOLO 设备的连接状态 0：未连接或遮挡，1：正常
获取 NOLO 设备位置与姿态接口	原型	Nolo_Pose getPoseByDeviceType(int type)
	功能	获取 NOLO 设备的位置与姿态信息
	参数	参数 type 表示设备类型， 0：表示头盔；1：表示左手柄；2：表示右手柄； 3：表示基站
	返回值	NOLO 设备的位置与姿态信息，Nolo_Pose 的属性详见 Demo
获取 NOLO 设备的反馈接口	原型	Nolo_ControllerStates getControllerStatesByDeviceType(int type)
	功能	获取 NOLO 设备的反馈信息
	参数	参数 type 表示设备类型， 0：表示头盔；1：表示左手柄；2：表示右手柄； 3：表示基站
	返回值	NOLO 设备的反馈信息，Nolo_ControllerStates 的属性详见 Demo
获取 NOLO 设备头盔初始位置接口	原型	Nolo_Vector3 getHmdInitPosition()
	功能	获取头盔标定高度时地面点的位置坐标
	参数	无
	返回值	返回头盔标定高度时地面点的位置坐标
获取 NOLO 设备头盔标定值接口	原型	int getHmdCalibration()
	功能	获取两点之间的标定值(该接口只对 NOLO 设备的 DK2 协议有效)
	参数	无
	返回值	两点之间的标定值

2. 接口调用流程说明

APP 通过 UsbCustomTransfer 类实例完成所有和 SDK 的交互请求，操作流程如下。



附录：接口中部分特殊的返回值类型

```
 public class Nolo_Vector3
{
   private float x;
   private float y;
   private float z;

   public void setX(float mx)
   {
      this.x = mx;
   }

   public void setY(float my) {
      this.y = my;
   }

   public void setZ(float mz) {
      this.z = mz;
   }

   public float getX() {
      return this.x;
   }

   public float getY() {
      return this.y;
   }
```

```
    public float getZ() {
        return this.z;
    }
}
public class Nolo_Quaternion
{
    private float x;
    private float y;
    private float z;
    private float w;

    public void setX(float mx)
    {
        this.x = mx;
    }

    public void setY(float my) {
        this.y = my;
    }

    public void setZ(float mz) {
        this.z = mz;
    }

    public void setW(float mw) {
        this.w = mw;
    }

    public float getX() {
        return this.x;
```

```
    }

    public float getY() {
        return this.y;
    }

    public float getZ() {
        return this.z;
    }

    public float getW() {
        return this.w;
    }
}
public class Nolo_Pose
{
    private Nolo_Vector3 pos;
    private Nolo_Quaternion rot;

    public void setPos(Nolo_Vector3 mpos)
    {
        this.pos = mpos;
    }

    public void setNolo_Quaternion(Nolo_Quaternion mrot) {
        this.rot = mrot;
    }

    public Nolo_Vector3 getPos() {
        return this.pos;
```

```
    }

    public Nolo_Quaternion getNolo_Quaternion() {
        return this.rot;
    }
}
public class Nolo_Vector2
{
    private float x;
    private float y;

    public void setX(float mx)
    {
        this.x = mx;
    }

    public void setY(float my) {
        this.y = my;
    }

    public float getX() {
        return this.x;
    }

    public float getY() {
        return this.y;
    }
}
public class Nolo_ControllerStates
{
```

```
    private int buttons;
    private int touches;
    private Nolo_Vector2 touchpadAxis;

    public void setButtons(int mbuttons)
    {
        this.buttons = mbuttons;
    }

    public void setTouches(int mtouches) {
        this.touches = mtouches;
    }

    public void setTouchpadAxis(Nolo_Vector2 mtouchpadAxis) {
        this.touchpadAxis = mtouchpadAxis;
    }

    public int getButtons() {
        return this.buttons;
    }

    public int getTouches() {
        return this.touches;
    }

    public Nolo_Vector2 getTouchpadAxis() {
        return this.touchpadAxis;
    }
}
public enum NoloButtonID
```

```
{
    TouchPad = 0,
    Trigger,
    Menu,
    System,
    Grip
}
public enum NoloTouchID
{
    TouchPad = 0
}
public class ButtonMask
{
    public const uint TouchPad = 1 << (int)NoloButtonID.TouchPad;
    public const uint Trigger = 1 << (int)NoloButtonID.Trigger;
    public const uint Menu = 1 << (int)NoloButtonID.Menu;
    public const uint System = 1 << (int)NoloButtonID.System;
    public const uint Grip = 1 << (int)NoloButtonID.Grip;
}
public class TouchMask
{
    public const uint TouchPad = 1 << (int)NoloTouchID.TouchPad;
}
```