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**

[1 文档概述 3](#_Toc12541)

[2 SDK接口说明 3](#_Toc12234)

[2.1 接口使用说明 3](#_Toc10196)

[2.1.1 接口调用流程说明 3](#_Toc2774)

[2.2 接口详细说明 3](#_Toc18203)

[2.2.1 获取SDK实例接口 3](#_Toc20063)

[2.2.2 SDK初始化接口 4](#_Toc142)

[2.2.3 SDK连接NOLO设备接口 4](#_Toc23932)

[2.2.4 SDK向NOLO设备发送数据接口 4](#_Toc18929)

[2.2.5 设置接收断开通知接口 5](#_Toc8651)

[2.2.6 SDK 断开NOLO设备接口 5](#_Toc3521)

[2.2.6 获取NOLO 设备版本号接口 5](#_Toc19902)

[2.2.8 获取NOLO设备电量接口 5](#_Toc9939)

[2.2.9 获取NOLO设备连接状态接口 6](#_Toc8802)

[2.2.10 获取NOLO设备位置与姿态接口 6](#_Toc25984)

[2.2.11 获取NOLO设备的反馈接口 6](#_Toc23607)

[2.2.12 获取NOLO设备头盔初始位置接口 7](#_Toc21385)

[2.2.13 获取NOLO设备头盔标定值接口 7](#_Toc18465)

[附录：接口中部分特殊的返回值类型 8](#_Toc21473)

# 文档概述

该文档描述的是北京凌宇智控科技有限公司NOLO VR Android SDK模块对外提供的接口说明，方便APP 接入方集成该SDK获取NOLO设备数据。

# SDK接口说明

## 接口使用说明

### 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。

1. 调用UsbCustomTransfer.getInstance(Context mcontext) 获取UsbCustomTransfer对象实例instance。
2. 调用instance.usb\_init() 初始化。
3. 调用instance.usb\_conn() 进行usb 设备连接，如果成功则返回1，如果失败则返回0**。**失败时检查设备是否已插入，如果成功，则可以操作与NOLO设备通信的功能接口，否则等待设备连接成功。
4. NOLO VR Android SDK 提供与NOLO 设备通信功能接口。
5. 当APP 退出时，需主动调用instance.usb\_finish() 接口释放与NOLO 设备的连接后再退出。

## 接口详细说明

### 2.2.1 获取SDK实例接口

|  |  |
| --- | --- |
| 原型 | UsbCustomTransfer getInstance(Context context ) |
| 功能 | 获得 UsbCustomTransfer 实例对象 |
| 参数 | Context：上下文环境 |
| 返回值 | UsbCustomTransfer 实例对象 |

### 2.2.2 SDK初始化接口

|  |  |
| --- | --- |
| 原型 | void usb\_init() |
| 功能 | 初始化SDK |
| 参数 | 无 |
| 返回值 | 无 |

### 2.2.3 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 |

### 2.2.4 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 |

### 2.2.6 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 |

### 2.2.8 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 |

### 2.2.9 获取NOLO设备连接状态接口

|  |  |
| --- | --- |
| 原型 | int getDeviceTrackingStatus(int type) |
| 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 |

### 2.2.10 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 |

### 2.2.11 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 |

# 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;

}