NOLO VR Android SDK FOR C Interfaces Documentation

LYRobotix Co., Ltd

May 2017

Directory

1. Introduction

NOLO VR Android SDK is the interfaces description which is provided by LYRobotix used for NOLO CV1, It is convenient for the APP accessor to integrate the SDK to get NOLO device data.

2. SDK Interfaces Description

2.1 Interfaces Detail

NOLO VR Android SDK FOR C has 9 interfaces, The name，function，functionality，parameter and return value of each interfaces are as follows.

Name	Description	
Interface of SDK connecting to NOLO device	Function	bool NoloDevice_conn()
	Functionality	Connect NOLO device interface
	Parameter	
	Return value	Return connection status code, false: connection failed; true: connection is successful
Interface of SDK sending data to NOLO device	Function	bool NoloDevice_sendData(int data[],int len)
	Functionality	Send data to NOLO device
	Parameter	mbyte: data to be sent Byte[4]: [0xAA(First word of frame head), 0x66(Second word of frame head), 0x00(leftcontroller vibration intensity, in the range (0x00 ~ 0x64)),

		0x00(rightcontroller vibration intensity, in the range (0x00 ~ 0x64))] len:Send data length, vibration length of 4
	Return value	true： send seccussful，false：send failed
Interface of SDK to disconnect NOLO device	Function	bool NoloDevice_finish()
	Functionality	Disconnect with NOLO device
	Parameter	
	Return value	true: finish successful false: finish failed
Interface of getting NOLO device version	Function	int GetVersionByDeviceType(int type)
	Functionality	Get NOLO device version
	Parameter	Parameter type means device type, 0: headset; 1: leftcontroller; 2: rightcontroller; 3:base station;
	Return value	Device version return value, 1: DK2; 2: CV1
Interface of getting NOLO device electricity quantity	Function	int GetElectricityByDeviceType(int type)
	Functionality	Get NOLO device electricity quantity
	Parameter	Parameter type means device type， 0: headset; 1: leftcontroller; 2: rightcontroller; 3:base station;
	Return value	NOLO device electricity quantity
Interface of getting NOLO device connection status	Function	int GetDeviceTrackingStatus(int type)
	Functionality	Get NOLO device connection status
	Parameter	Parameter type means device type， 0: headset; 1: leftcontroller; 2: rightcontroller; 3:base station;
	Return value	NOLO device connection status: 0: not connected or

		blocked; 1: normal
Interface of getting NOLO device position and attitude	Function	Nolo_Pose GetPoseByDeviceType(int type)
	Functionality	Get NOLO device position and attitude information
	Parameter	Parameter type means device type， 0: headset; 1: leftcontroller; 2: rightcontroller; 3:base station;
	Return value	Position and attitude information of NOLO device, see the attributes of Nolo_Pose in addendix
Interface of getting NOLO device feedback	Function	Nolo_ControllerStates GetControllerStatesByDeviceType(int type)
	Functionality	Get NOLO device feedback information
	Parameter	Parameter type means device type， 0: headset; 1: leftcontroller; 2: rightcontroller; 3:base station;
	Return value	Feedback information of NOLO device, see the attributes of Nolo_ControllerStates in addendix
Interface of getting NOLO device headset initial position	Function	Nolo_Vector3 GetHmdInitPosition()
	Functionality	Get the coordinate point on surface when the helmet is calibrated
	Parameter	
	Return value	Return the coordinate point on surface when the helmet is calibrated
	Return value	The calibration value between two points

Appendix: Head file

```
#pragma once

#ifndef NOLO_API
#define NOLO_API extern "C"
#endif
#include <windows.h>
#include <math.h>

namespace NOLO
{
    typedef struct Nolo_Vector2
    {
        float x;
        float y;
    }Nolo_Vector2;

    typedef struct Nolo_Vector3
    {
        float x;
        float y;
        float z;
    }Nolo_Vector3;

    typedef struct Nolo_Quaternion
    {
        float x;
        float y;
        float z;
        float w;
    }Nolo_Quaternion;

    typedef struct Nolo_Pose
    {
        Nolo_Vector3 pos;
        Nolo_Quaternion rot;
    }Nolo_Pose;

    /*
        buttons:
```

```
        TouchPad = 1 << 0;
        Trigger = 1 << 1;
       Menu = 1 << 2;
       System = 1 << 3;
       Grip = 1 << 4;
    */
    typedef struct Nolo_ControllerStates
    {
        UINT buttons;
        UINT touches;
        Nolo_Vector2 touchpadAxis;
    }Nolo_ControllerStates;

    /**
    ********************************************************************************
    * Function description: After the application gets the usb permission, call this
method to establish a connection
    * Parameter: null
    * Return Value: bool
    *【应用获取到 usb 读取权限之后，调用此方法与建立连接】
    ********************************************************************************
    */
    NOLO_API bool _cdecl NoloDevice_conn();

    /**
    ********************************************************************************
    * Function description: Disconnect the application from the usb device
    * Parameter: null
    * Return Value: bool
    *【断开应用与 usb 设备的连接】
    ********************************************************************************
    */
    NOLO_API bool _cdecl NoloDevice_finish();

    /**
    ********************************************************************************
    * Function description: Get device tracking status
    * Parameter: int
```

```
    0:hmd
    1:conroller one
    2:controller two
    3:basestation
    * Return Value: int
    *【获取设备跟踪状态】
    ********************************************************************************
***
    */
    NOLO_API int _cdecl GetDeviceTrackingStatus(int type);

    /**
    ********************************************************************************
***
    * Function description: Get the device hardware version
    * Parameter: int
    0:hmd
    1:conroller one
    2:controller two
    3:basestation
    * Return Value: int
    *【获取设备硬件版本号】
    ********************************************************************************
***
    */
    NOLO_API int _cdecl GetVersionByDeviceType(int type);

    /**
    ********************************************************************************
***
    * Function description: Get NOLO device electricity quantity
    * Parameter: int
    0:hmd
    1:conroller one
    2:controller two
    3:basestation
    * Return Value: int
    *【获取设备电量信息】
    ********************************************************************************
***
    */
    NOLO_API int _cdecl GetElectricityByDeviceType(int type);
```

```
    /**
    ********************************************************************************
    * Function description: Get device tracking information
    * Parameter: int
    0:hmd
    1:conroller one
    2:controller two
    3:basestation
    * Return Value: Nolo_Pose
    *【获取设备定位信息】
    ********************************************************************************
    */
    NOLO_API Nolo_Pose _cdecl GetPoseByDeviceType(int type);

    /**
    ********************************************************************************
    * Function description: Get the controller status information
    * Parameter: int
    0:hmd
    1:conroller one
    2:controller two
    3:basestation
    * Return Value: Nolo_ControllerStates
    *【获取手柄状态信息】
    ********************************************************************************
    */
    NOLO_API Nolo_ControllerStates _cdecl GetControllerStatesByDeviceType(int type);

    /**
    ********************************************************************************
    * Function description: Get the headset marker when calibrating the position
    * Parameter: NULL
    * Return Value: Nolo_Vector3
    *【获取头盔标定时位置】
    ********************************************************************************
    */
```

```
    NOLO_API Nolo_Vector3 _cdecl GetHmdInitPosition();

    /**
    ********************************************************************************
    * Function description: Send data to NOLO device
    * Parameter: int data[] int len
    * Return Value: bool
    *【震动数据接口】
    ********************************************************************************
    */
    NOLO_API bool _cdecl NoloDevice_sendData(int[] data, int len);
}
```