**NOLO VR Windows SDK**

**Interfaces Documentation**

LYRobotix Co., Ltd

May 2017

**Directory**

[1 Document overview 3](#_Toc483354201)

[2 SDK Interfaces Description 3](#_Toc483354202)

[2.1 Using interfaces 3](#_Toc483354203)

[2.1.1 Communication process 3](#_Toc483354204)

[2.1.2 Calling process 4](#_Toc483354205)

[2.2 Interfaces Detail 4](#_Toc483354206)

[2.2.1 Interface of opening NOLO ZeroMQ client 4](#_Toc483354207)

[2.2.2 Interface of closing NOLO ZeroMQ client 5](#_Toc483354208)

[2.2.3 Interface of connection successful notification between client and NOLO server 5](#_Toc483354209)

[2.2.4 Interface of disconnection notification between client and NOLO server 5](#_Toc483354210)

[2.2.5 Interface of getting all NOLO devices data 6](#_Toc483354211)

[2.2.6 Interface of getting data from leftcontroller of NOLO device 6](#_Toc483354212)

[2.2.7 Interface of getting data from rightcontroller of NOLO device 6](#_Toc483354213)

[2.2.8 Interface of getting data from headset marker of NOLO device 6](#_Toc483354214)

[2.2.9 Interface of getting expanded data from NOLO device 7](#_Toc483354215)

[2.2.10 Interface of getting NOLO device headset initial position 7](#_Toc483354216)

[2.2.11 Interface of getting NOLO device status 7](#_Toc483354217)

[2.2.12 Interface of getting NOLO device electricity quantity 7](#_Toc483354218)

[2.2.13 Interface of getting NOLO device headset calibration value 8](#_Toc483354219)

[2.2.14 Interface of getting NOLO device version 8](#_Toc483354220)

[2.2.15 Interface of getting data from controllers of NOLO device 8](#_Toc483354221)

[2.2.16 Interface of getting NOLO device position and attitude 8](#_Toc483354222)

[2.2.17 Interface of setting vibration data to controllers of NOLO device 9](#_Toc483354223)

# Document overview

This document describes the interfaces description provided by the NOLO VR Windows SDK module, which the module is provided by LYRobotix. It is convenient for the developers to integrate the SDK to get NOLO device data。

# SDK Interfaces Description

## Using interfaces

### Communication process

As shown in the figure below, the NOLO device consists of a base station, a headset marker and two controllers, the base station and the controllers interact with the headset marker in a wireless communication. The headset marker gather the data and communicate with computer in two-way through the USB protocol. Computer-side Nolo\_driver\_for\_windows software can get the data information of NOLO device, and transfer data to nolo\_api in two-way through the ZeroMQ protocol. Nolo\_driver\_for\_windows software is the socket server of PUB and Router ZMQ, noloRuntime.dll is the socket client of SUB and Dealer ZMQ.

NOLO

headset marker

Nolo\_driver\_for\_windows software

NOLO base station

NOLO

controllers\*2

Nolo\_api

wireless

wireless

USB

ZeroMQ

noloRuntime.dll

### Calling process

Nolo\_api provides all external interface functions and data structure of noloRuntime.dll, the specific operation process is as follows.

1. Install Nolo\_driver\_for\_windows\_setup.msi (necessary).
2. Install NOLO\_Windows\_Setup.msi (optional).
3. Call open\_Nolo\_ZeroMQ to open the NOLO ZeroMQ client, returns true if successful, false if it fails. When it fails, check if the NOLO ZeroMQ client has been opened, if successful, you can operate other functional interfaces that communicate with the NOLO device.
4. When you exit the application, you need to call the close\_Nolo\_ZeroMQ interface actively to release the resources connected to the Nolo\_driver\_for\_windows software.

## Interfaces Detail

### Interface of opening NOLO ZeroMQ client

|  |  |
| --- | --- |
| Function | Bool open\_Nolo\_ZeroMQ() |
| Functionality | Open NOLO ZeroMQ client |
| parameter |  |
| return value | Return opening status: false; true |

### 2.2.2 Interface of closing NOLO ZeroMQ client

|  |  |
| --- | --- |
| Function | void close\_Nolo\_ZeroMQ() |
| Functionality | Close the communication between SDK and NOLO |
| parameter |  |
| return value |  |

### 2.2.3 Interface of connection successful notification between client and NOLO server

|  |  |
| --- | --- |
| Function | Bool connectSuccess\_FunCallBack(funcCallBack func) |
| Functionality | Call the registered func function when the connection between client and Nolo\_driver\_for\_windows software server is successful |
| parameter | Parameter func, custom function pointer,  typedef void(\*funcCallBack)(); |
| return value | Returns the function registration status: false; true |

### 2.2.4 Interface of disconnection notification between client and NOLO server

|  |  |
| --- | --- |
| Function | Bool disConnect\_FunCallBack(funcCallBack func) |
| Functionality | Call the registered func function when the connection between client and Nolo\_driver\_for\_windows software server is disconnected |
| Parameter | Parameter func, custom function pointer,  typedef void(\*funcCallBack)(); |
| return value | Returns the function registration status: false; true |

### Interface of getting all NOLO devices data

|  |  |
| --- | --- |
| Function | NoloData get\_Nolo\_NoloData() |
| Functionality | Get all data of NOLO devices, such as the data of headset marker, controllers and base station. |
| parameter |  |
| return value | Returns the NoloData structure data, see nolo\_api for properties |

### 2.2.6 Interface of getting data from leftcontroller of NOLO device

|  |  |
| --- | --- |
| Function | Controller get\_Nolo\_LeftControllerData() |
| Functionality | Get data from leftcontroller of NOLO device |
| parameter |  |
| return value | Returns the Controller structure data, see nolo\_api for properties |

### 2.2.7 Interface of getting data from rightcontroller of NOLO device

|  |  |
| --- | --- |
| Function | Controller get\_Nolo\_RightControllerData() |
| Functionality | Get data from rightcontroller of NOLO device |
| parameter |  |
| return value | Returns the Controller structure data, see nolo\_api for properties |

### 2.2.8 Interface of getting data from headset marker of NOLO device

|  |  |
| --- | --- |
| Function | HMD get\_Nolo\_HMDData() |
| Functionality | Get data from headset marker of NOLO device |
| parameter |  |
| return value | Returns the HMD structure data, see nolo\_api for properties |

### 2.2.9 Interface of getting expanded data from NOLO device

|  |  |
| --- | --- |
| Function | BYTE\* get\_Nolo\_ExpandData() |
| Functionality | Get expanded data from NOLO device, such as double-click system button, double-click menu button |
| parameter |  |
| return value | Returns the packet address of BYTE data [64]  data[0]>>0 :Double click Menu  data[0]>>1 :Double click System |

### 2.2.10 Interface of getting NOLO device headset initial position

|  |  |
| --- | --- |
| Function | Vector3 get\_Nolo\_HMDInitPosition() |
| Functionality | Get the initial position of NOLO device |
| parameter |  |
| return value | Returns the Vector3 structure data, see nolo\_api for properties |

### 2.2.11 Interface of getting NOLO device status

|  |  |
| --- | --- |
| Function | int get\_Nolo\_StateByDeviceType(NoloDeviceType type) |
| Functionality | Get status data from NOLO device |
| parameter | Parameter type is an enumeration type, see nolo\_api for properties |
| return value | Returns int type data: 0: blocked; 1: normal |

### Interface of getting NOLO device electricity quantity

|  |  |
| --- | --- |
| Function | int get\_Nolo\_Battery(NoloDeviceType deviceType) |
| Functionality | Get the data of NOLO device electricity quantity |
| parameter | Parameter deviceType is an enumeration type, see nolo\_api for properties |
| return value | Returns int type data: 0-100: the percentage of electricity quantity; 255: turn the power off |

### Interface of getting NOLO device headset calibration value

|  |  |
| --- | --- |
| Function | int get\_Nolo\_HMDTwoPointDriftAngle() |
| Functionality | Get the calibration value between two points ( This interface is valid only for the DK2 protocol of NOLO device ) |
| parameter |  |
| return value | The calibration value between two points |

### Interface of getting NOLO device version

|  |  |
| --- | --- |
| Function | int get\_Nolo\_VersionID(NoloDeviceType devicetype) |
| Functionality | Get NOLO device version |
| parameter | Parameter deviceType is an enumeration type, see nolo\_api for properties |
| return value | Returns int type data: device version |

### Interface of getting data from controllers of NOLO device

|  |  |
| --- | --- |
| Function | ControllerStates get\_Nolo\_ControllerStates(NoloDeviceType type) |
| Functionality | Get data from controllers of NOLO device, such as the data of0 buttons, touch and Axis |
| parameter | Parameter type is an enumeration type, see nolo\_api for properties |
| return value | Returns the ControllerStates structure data, see nolo\_api for propertie |

### Interface of getting NOLO device position and attitude

|  |  |
| --- | --- |
| Function | Nolo\_Pose get\_Nolo\_Pose(NoloDeviceType devicetype) |
| Functionality | Get NOLO device position and attitude information |
| parameter | Parameter deviceType is an enumeration type, see nolo\_api for properties |
| return value | Returns the Nolo\_Pose structure data, see nolo\_api for propertie |

### Interface of setting vibration data to controllers of NOLO device

|  |  |
| --- | --- |
| Function | Void set\_Nolo\_TriggerHapticPulse(NoloDeviceType type,int intensity) |
| Functionality | Set vibration data to controllers of NOLO device |
| parameter | Parameter deviceType is an enumeration type, see nolo\_api for properties  Parameter intensity means vibration intensity, in the range (0~100), larger is more intense |
| return value |  |