NOLO VR Windows SDK

Interfaces Documentation

Version: Windows SDK 0.1RC

LYRobotix Co., Ltd

1st December 2017

Directory

1 Document Overview 3

1.1 New Version Content 3

2 SDK Interfaces Description 3

2.1 Interfaces Functions Description 3

2.2 Using process Description 7

1 Document Overview

This document offers the interfaces description of NOLO VR Windows SDK module, which is provided by LYRobotix. The NOLO VR Windows SDK is the external interface that applies to NOLO CV1 product. It is convenient for developers to access NOLO device data through this SDK.

1.1 New Version Content

1. Added Kalman filtering algorithm, the device data will be smoother and more fluid.

2. Added some statistical information function.

3. Added directional buttons on the touchpad (i.e. up, down, left, right).

2 SDK Interfaces Description

2.1 Interfaces Functions Description

Function: search_Nolo_Device

Functionality: Search Nolo devices
Parameter: NULL
Return value: NULL
Attention: In addition to registering and setting related functions, this function should be called with priority so that the SDK can be used normally.

Function: close_Nolo_Device

Functionality: Turn off Nolo device
Parameter: NULL
Return value: NULL
Attention: Call this function when you need to exit the program, so that the resources requested

within the SDK can be released.

Function: registerDisConnectCallBack

Functionality: Register callback function, this callback function is used to monitor the message when the device is disconnected.

Parameter:

@callBackFun: Pointer of callback function

@context: User-defined parameters (the content will not be changed)

Return value: Returns true if successful, false otherwise.

Attention: NULL

Function: registerConnectSuccessCallBack

Functionality: Register callback function, this callback function is used to monitor the message when the device is disconnected.

Parameter:

@callBackFun : Pointer of callback function

@context : User-defined parameters (the content will not be changed)

Return value: Returns true if successful, false otherwise.

Attention: NULL

Function: registerExpandDataNotifyCallBack

Functionality: Register callback function, this callback function is used to monitor the double-click event of system button and menu button.

Parameter:

@callBackFun : Pointer of callback function

@context : User-defined parameters (the content will not be changed)

Return value: Returns true if successful, false otherwise.

Attention: NULL

Function: registerNoloDataNotifyCallBack

Functionality: Register callback function, callback function will be called whenever new data arrives.

Parameter:

@callBackFun : Pointer of callback function

@context : User-defined parameters (the content will not be changed)

Return value: Returns true if successful, false otherwise.

Attention: NULL

Function: set_Nolo_TriggerHapticPulse

Functionality: Vibrate the controller

Parameter:

@deviceType Effective when the controller type is LeftControllerDevice or RightControllerDevice.

@intensity Vibration intensity, in the range (0 ~ 100).

Return value: Returns true if effective, false otherwise

Attention: NULL

Function: get_Nolo_NoloData

Functionality: Get Nolo data

Parameter: NULL

Return value: NoloData structure

Attention: NULL

Function: get_Nolo_LeftControllerData

Functionality: Get Leftcontroller data

Parameter: NULL

Return value: Controller structure

Attention: NULL

Function: get_Nolo_RightControllerData

Functionality: Get Rightcontroller data

Parameter: NULL

Return value: Controller structure

Attention: NULL

Function: get_Nolo_HMDData

Functionality: Get HMD data

Parameter: NULL

Return value: HMD structure

Attention: NULL

Function: get_Nolo_BaseStationData

Functionality: Get BaseStation data

Parameter: NULL

Return value: BaseStation structure

Attention: NULL

Function: get_Nolo_Battery

Functionality: Get device battery status

Parameter: @deviceType: Device Type

Return value: EBattery: Enum value

Attention: NULL

Function: set_Nolo_TurnAroundKey

Functionality: Set turn around button, the default is the menu button

Parameter: @Type: New button value

Return value: NULL

Attention: NULL

Function: set_Nolo_PlayMode

Functionality: Set the base station placement mode to " normal (front-facing)" or "ceiling" mode

Parameter: @Type: Mode type, refer to EPlayModel

Return value: NULL

Attention: NULL

Function: set_Nolo_HmdTrackingCenter

Functionality: Set HMD's rotation center and HMD type name

Parameter:

@vec: HMD's rotation center, the relative position between the midpoint of two lenses and the center point of NOLO headset marker (with the midpoint of two lenses as the origin, the left and right for the X axis, up and down for the Y axis, front and rear for the Z axis).

@strHmdTypeName: HMD type name, which is set by the docking user, specific HMD will be optimized by Nolo, such as "HTC_VIVE2".

Return value: NULL

Attention: Due to the difference in HMD size and center settings among HMD manufacturers, user can set HMD's rotation center and HMD type name to improve precision and user experience according to different HMD types.

Function: set_PermitHttp

Functionality: Set whether to allow sending Http messages

Parameter: @bPermit: Set to true if allowed, otherwise false

Return value: NULL

Attention: The default value is true, the SDK will gather abnormal information, as well as usage time.

2.2 Using process Description

①In the developer's process, load NoLo_USBHID.dll;

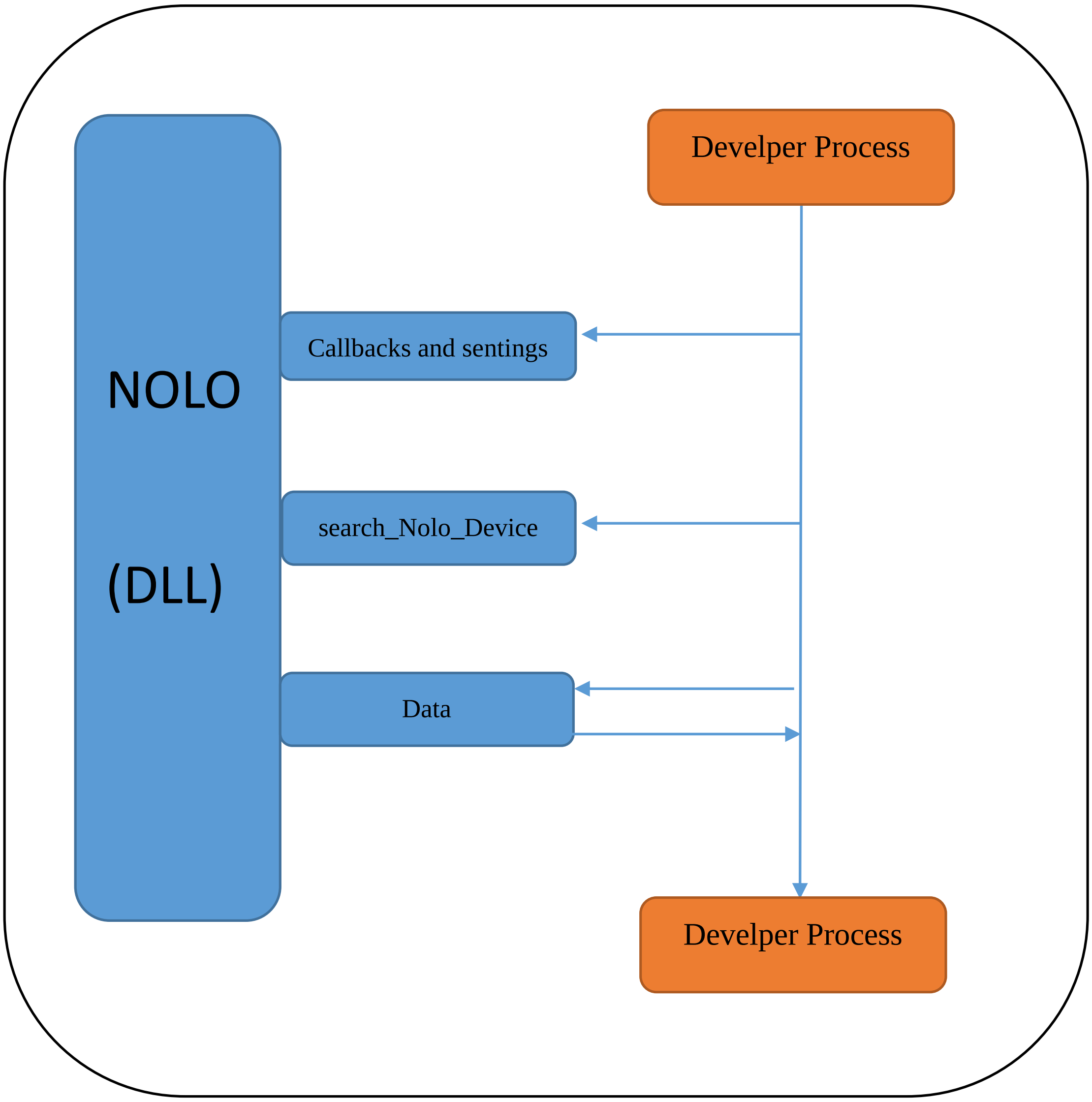
②Call setting (optional): HMD's rotating center, HMD's type name, base station placement mode, etc.；

③Search device: search_Nolo_Device；

④There are two ways to read Nolo data :

1. Read in the callback function (Use the callback function: registerNoloDataNotifyCallBack to get the NOLO data).

2. Read directly using get_Nolo_NoloData.


Develper Process
NOLO
(DLL)
Callbacks and sentings
search_Nolo_Device
Data
Develper Process