VRTRIX Data Glove Unity SDK Reference

Generated by Doxygen 1.8.16

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 VRTRIX Namespace Reference

Classes

- class VRTRIXDataWrapper

  VRTRIX Data Glove data wrapper class.
- class VRTRIXGloveDataStreaming

  VRTRIX Data Glove data streaming class.
- class VRTRIXUtilities

  Bone utility functions class.

Enumerations

- enum GloveIndex {
  None = -1, **Device0** = 0, **Device1** = 1, **Device2** = 2,
  Device3 = 3, **Device4** = 4, **Device5** = 5, **Device6** = 6,
  Device7 = 7, **Device8** = 8, **Device9** = 9, **Device10** = 10,
  Device11 = 11, **Device12** = 12, **Device13** = 13, **Device14** = 14,
  Device15 = 15, **MaxDeviceCount** = 16 }

  GloveIndex enum.
- enum HANDTYPE { **RIGHT_HAND**, **LEFT_HAND**, **BOTH_HAND**, **NONE** }

  Hand type enum.
- enum GLOVEVERSION { **DK1**, **DK2**, **PRO** }

  Glove hardware version.
- enum VRTRIXGloveStatus {
  CLOSED, **NORMAL**, **PAUSED**, **DISCONNECTED**,
  MAGANOMALY }

  Glove connection status.
- enum VRTRIXGloveEvent {
  VRTRIXGloveEvent_None = 0, **VRTRIXGloveEvent_Idle** = 1, **VRTRIXGloveEvent_Connected** = 2, **VR↩
  TRIXGloveEvent_Disconnected** = 3,
  VRTRIXGloveEvent_PortClosed = 4, **VRTRIXGloveEvent_LowBattery** = 5, **VRTRIXGloveEvent_↩
  BatteryFull** = 6, **VRTRIXGloveEvent_Paired** = 7,
  VRTRIXGloveEvent_MagAbnormal = 8, **VRTRIXGloveEvent_TrackerConnected** = 9, **VRTRIXGlove↩
  Event_TrackerDisconnected** = 10, **VRTRIXGloveEvent_ChannelHopping** = 11 }

  Glove event enum.

- enum VRTRIXBones {
  R_Hand = 0, **R_Thumb_1** = 1, **R_Thumb_2** = 2, **R_Thumb_3** = 3,
  R_Index_1 = 4, **R_Index_2** = 5, **R_Index_3** = 6, **R_Middle_1** = 7,
  R_Middle_2 = 8, **R_Middle_3** = 9, **R_Ring_1** = 10, **R_Ring_2** = 11,
  R_Ring_3 = 12, **R_Pinky_1** = 13, **R_Pinky_2** = 14, **R_Pinky_3** = 15,
  L_Hand = 16, **L_Thumb_1** = 17, **L_Thumb_2** = 18, **L_Thumb_3** = 19,
  L_Index_1 = 20, **L_Index_2** = 21, **L_Index_3** = 22, **L_Middle_1** = 23,
  L_Middle_2 = 24, **L_Middle_3** = 25, **L_Ring_1** = 26, **L_Ring_2** = 27,
  L_Ring_3 = 28, **L_Pinky_1** = 29, **L_Pinky_2** = 30, **L_Pinky_3** = 31,
  R_Arm = 32, **L_Arm** = 33, **NumOfBones** = 34 }

  Hand joints enum.

4.1.1 Enumeration Type Documentation

4.1.1.1 GloveIndex

```
enum VRTRIX.GloveIndex  [strong]
```

GloveIndex enum.

Enum of supported gloves hardware index.

4.1.1.2 GLOVEVERSION

```
enum VRTRIX.GLOVEVERSION  [strong]
```

Glove hardware version.

Supported hardware version, currently DK1, DK2 & PRO are supported.

4.1.1.3 HANDTYPE

```
enum VRTRIX.HANDTYPE  [strong]
```

Hand type enum.

The chirality of the hand, used to identify data glove attribute.

4.1.1.4 VRTRIXBones

```
enum VRTRIX.VRTRIXBones  [strong]
```

Hand joints enum.

Enum of joints for both hands.

4.1.1.5 VRTRIXGloveEvent

```
enum VRTRIX.VRTRIXGloveEvent  [strong]
```

Glove event enum.

Define the glove events while running.

4.1.1.6 VRTRIXGloveStatus

```
enum VRTRIX.VRTRIXGloveStatus  [strong]
```

Glove connection status.

Define the glove connection status.

Chapter 5

Class Documentation

5.1 VRTRIX.VRTRIXDataWrapper.VRTRIX_Quat Struct Reference

Quaternion data struction used in unmanaged C++ API.

Public Attributes

- float qx

    x component in quaternion
- float qy

    y component in quaternion
- float qz

    z component in quaternion
- float qw

    w component in quaternion

5.1.1 Detailed Description

Quaternion data struction used in unmanaged C++ API.

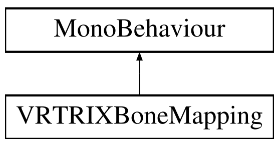
The documentation for this struct was generated from the following file:

- VRTRIXBasicDataStreaming/VRTRIXDataWrapper.cs

5.2 VRTRIXBoneMapping Class Reference

Bone mapping class.

Inheritance diagram for VRTRIXBoneMapping:



Public Member Functions

- GameObject MapToVRTRIX_BoneName (string bone_name)
  Get custom model joint as gameobject according to bone name specified.

Public Attributes

- Transform[ ] **MyCharacterFingers** = new Transform[(int)VRTRIXBones.NumOfBones]

Static Public Attributes

- static VRTRIXBoneMapping **UniqueStance**

5.2.1 Detailed Description

Bone mapping class.

A class to map bone names of custom model with VRTRIX bone setup.

5.2.2 Member Function Documentation

5.2.2.1 MapToVRTRIX_BoneName()

```
GameObject VRTRIXBoneMapping.MapToVRTRIX_BoneName (
             string bone_name )
```

Get custom model joint as gameobject according to bone name specified.

Parameters

bone_name	Given VRTRIX bone name.

Returns

joint on the custom model as gameobject.

The documentation for this class was generated from the following file:

- VRTRIXBasicDataStreaming/VRTRIXBoneMapping.cs

5.3 VRTRIX.VRTRIXDataWrapper Class Reference

VRTRIX Data Glove data wrapper class.

Classes

- struct VRTRIX_Quat

    Quaternion data struction used in unmanaged C++ API.

Public Member Functions

- delegate void ReceivedDataCallback (IntPtr pUserParam, IntPtr ptr, int data_rate, byte radio_strength, IntPtr cal_score_ptr, float battery, int hand_type, int radio_channel)

    The delegate data receive function called inside unmanaged C++ API.
- delegate void ReceivedEventCallback (IntPtr pUserParam, IntPtr pEvent)

    The delegate event receive function called inside unmanaged C++ API.
- static IntPtr InitDataGlove (bool AdvancedMode, GLOVEVERSION HardwareVersion)

    Intialize the data glove and returns the interface pointer.
- static bool OpenPort (IntPtr sp, int glove_id, HANDTYPE type)

    Open the serial port
- static void StartStreaming (IntPtr sp)

    Read the data from serial port asynchronously.
- static bool ClosePort (IntPtr sp)

    Close the serial port
- static void OnSaveCalibration (IntPtr sp)

    Save calibration result to hardware
- static void OnCloseFingerAlignment (IntPtr sp)

    Align the close finger pose
- static void OnOkPoseAlignment (IntPtr sp)

    Align the OK finger pose
- static void RegisterDataCallback (IntPtr pUserParam, IntPtr sp, ReceivedDataCallback receivedData↩ Callback)

    Register receiving and parsed frame calculation data callback
- static void RegisterEventCallback (IntPtr sp, ReceivedEventCallback receivedEventCallback)

    Register receiving hardware event callback
- static void VibratePeriod (IntPtr sp, int msDurationMillisec)

    Vibrate the data glove for given time period.
- static void ChannelHopping (IntPtr sp)

    Randomly channel hopping.
- static void SetAdvancedMode (IntPtr sp, bool bIsAdvancedMode)

    Set Advanced Mode.
- static void SetProximalThumbOffset (IntPtr sp, double offset_x, double offset_y, double offset_z)

    Set Proximal Thumb Offset.
- static void SetIntermediateThumbOffset (IntPtr sp, double offset_x, double offset_y, double offset_z)

    Set Intermediate Thumb Offset.
- static void SetDistalThumbOffset (IntPtr sp, double offset_x, double offset_y, double offset_z)

    Set Distal Thumb Offset.
- static void SetThumbSlerpRate (IntPtr sp, double slerp_proximal, double slerp_middle)

    Set Thumb Slerp Rate.
- VRTRIXDataWrapper (bool AdvancedMode, int GloveIndex, GLOVEVERSION HardwareVersion)

    Wrapper class construction method.
- bool Init (HANDTYPE type)

    Initialization method.
- bool ClosePort ()

Close port to stop data streaming.
- void RegisterCallBack ()

  Register call back function to the C++ umanaged dll.
- VRTRIXGloveStatus GetReceivedStatus ()

  Get data glove status.
- float GetReceivedGestureAngle (VRTRIXBones bone)

  Get the angle between specific joint and wrist joint to detect gesture.
- int GetReceivedDataRate ()

  Get data rate received per second.
- int GetReceiveRadioStrength ()

  Get radio strength of data glove.
- int GetReceiveRadioChannel ()

  Get current radio channel of data glove used.
- float GetReceiveBattery ()

  Get current battery level in percentage of data glove.
- int GetReceivedCalScore (VRTRIXBones bone)

  Get current calibration score for specific IMU sensor.
- int GetReceivedCalScoreMean ()

  Get current calibration score average value.
- Quaternion GetReceivedRotation (VRTRIXBones bone)

  Get current rotation for specfic joint.
- void OnSaveCalibration ()

  Save calibration parameters to hardware flash.
- void VibratePeriod (int msDurationMillisec)

  Trigger a haptic vibration for a certain period.
- void OnCloseFingerAlignment (HANDTYPE type)

  Align current gesture to finger close pose, used for calibration when advanced mode is activated.
- void StartStreaming ()

  Start data streaming of data glove.
- void ChannelHopping ()

  Trigger channel switching mannually, only used in testing/debuging.
- void SetAdvancedMode (bool bIsAdvancedMode)

  Activate advanced mode so that finger's yaw data will be unlocked.
- void SetThumbOffset (Vector3 offset, VRTRIXBones joint)

  Set thumb offset to counteract the difference between hands & gloves sensor installation.
- void SetThumbSlerpRate (double slerp_proximal, double slerp_middle)

  Set thumb slerp rate to counteract the difference between hands & gloves sensor installation.

Static Public Attributes

- static ReceivedDataCallback **receivedDataCallback**
- static ReceivedEventCallback **receivedEventCallback**

5.3.1 Detailed Description

VRTRIX Data Glove data wrapper class.

A wrapper class to communicate with low-level unmanaged C++ API.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 VRTRIXDataWrapper()

```
VRTRIX.VRTRIXDataWrapper.VRTRIXDataWrapper (
            bool AdvancedMode,
            int GloveIndex,
            GLOVEVERSION HardwareVersion )
```

Wrapper class construction method.

Parameters

AdvancedMode	Whether the advanced mode is activated
GloveIndex	Data glove index (Maximum is 15, if larger number is set, then only one pair of glove per PC is supported).
HardwareVersion	Data glove hardware version, currently DK1, DK2 and PRO are supported.

5.3.3 Member Function Documentation

5.3.3.1 ChannelHopping()

```
static void VRTRIX.VRTRIXDataWrapper.ChannelHopping (
            IntPtr sp )
```

Randomly channel hopping.

Parameters

sp	The serial port object

5.3.3.2 ClosePort() [1/2]

```
bool VRTRIX.VRTRIXDataWrapper.ClosePort ( )
```

Close port to stop data streaming.

Returns

whether data streaming is stopped successfully.

5.3.3.3 ClosePort() [2/2]

```
static bool VRTRIX.VRTRIXDataWrapper.ClosePort (
             IntPtr sp )
```

Close the serial port

Parameters

sp	The serial port object

Returns

Whether the serial port is closed successfully

5.3.3.4 GetReceiveBattery()

```
float VRTRIX.VRTRIXDataWrapper.GetReceiveBattery ( )
```

Get current battery level in percentage of data glove.

Returns

current battery level in percentage of data glove.

5.3.3.5 GetReceivedCalScore()

```
int VRTRIX.VRTRIXDataWrapper.GetReceivedCalScore (
             VRTRIXBones bone )
```

Get current calibration score for specific IMU sensor.

Parameters

bone	specific joint of hand.

Returns

current calibration score for specific IMU sensor. Lower value of score means better calibration performance.

5.3.3.6 GetReceivedCalScoreMean()

```
int VRTRIX.VRTRIXDataWrapper.GetReceivedCalScoreMean ( )
```

Get current calibration score average value.

Returns

current calibration score average value.

5.3.3.7 GetReceivedDataRate()

```
int VRTRIX.VRTRIXDataWrapper.GetReceivedDataRate ( )
```

Get data rate received per second.

Returns

data rate received per second.

5.3.3.8 GetReceivedGestureAngle()

```
float VRTRIX.VRTRIXDataWrapper.GetReceivedGestureAngle (
             VRTRIXBones bone )
```

Get the angle between specific joint and wrist joint to detect gesture.

Parameters

bone	specific joint of hand.

Returns

the gesture angle for specific joint.

5.3.3.9 GetReceivedRotation()

```
Quaternion VRTRIX.VRTRIXDataWrapper.GetReceivedRotation (
             VRTRIXBones bone )
```

Get current rotation for specfic joint.

Parameters

bone	specific joint of hand.

Returns

current calibration score average value.

5.3.3.10 GetReceivedStatus()

```
VRTRIXGloveStatus VRTRIX.VRTRIXDataWrapper.GetReceivedStatus ( )
```

Get data glove status.

Returns

data glove status.

5.3.3.11 GetReceiveRadioChannel()

```
int VRTRIX.VRTRIXDataWrapper.GetReceiveRadioChannel ( )
```

Get current radio channel of data glove used.

Returns

current radio channel of data glove used.

5.3.3.12 GetReceiveRadioStrength()

```
int VRTRIX.VRTRIXDataWrapper.GetReceiveRadioStrength ( )
```

Get radio strength of data glove.

Returns

radio strength of data glove.

5.3.3.13 Init()

```
bool VRTRIX.VRTRIXDataWrapper.Init (
             HANDTYPE type )
```

Initialization method.

Parameters

type	Data glove hand type.

Returns

whether data glove is initialized successfully.

5.3.3.14 InitDataGlove()

```
static IntPtr VRTRIX.VRTRIXDataWrapper.InitDataGlove (
            bool AdvancedMode,
            GLOVEVERSION HardwareVersion )
```

Intialize the data glove and returns the interface pointer.

Parameters

AdvancedMode	Unlock the yaw of fingers if set true
HardwareVersion	Specify the data glove hardware version

Returns

The serial port object as IntPtr

5.3.3.15 OnCloseFingerAlignment() [1/2]

```
void VRTRIX.VRTRIXDataWrapper.OnCloseFingerAlignment (
            HANDTYPE type )
```

Align current gesture to finger close pose, used for calibration when advanced mode is activated.

Parameters

type	Hand type of data glove

5.3.3.16 OnCloseFingerAlignment() [2/2]

```
static void VRTRIX.VRTRIXDataWrapper.OnCloseFingerAlignment (
            IntPtr sp )
```

Align the close finger pose

Parameters

sp	The serial port object

5.3.3.17 OnOkPoseAlignment()

```
static void VRTRIX.VRTRIXDataWrapper.OnOkPoseAlignment (
            IntPtr sp )
```

Align the OK finger pose

Parameters

sp	The serial port object

5.3.3.18 OnSaveCalibration()

```
static void VRTRIX.VRTRIXDataWrapper.OnSaveCalibration (
            IntPtr sp )
```

Save calibration result to hardware

Parameters

sp	The serial port object

5.3.3.19 OpenPort()

```
static bool VRTRIX.VRTRIXDataWrapper.OpenPort (
            IntPtr sp,
            int glove_id,
            HANDTYPE type )
```

Open the serial port

Parameters

sp	The serial port object
glove←_id	Data glove index id (from 0 - 15), if anything larger is set,then only one pair of glove is supported
type	Hand type.

Returns

Whether the port is opened successfully

5.3.3.20 ReceivedDataCallback()

```
delegate void VRTRIX.VRTRIXDataWrapper.ReceivedDataCallback (
             IntPtr pUserParam,
             IntPtr ptr,
             int data_rate,
             byte radio_strength,
             IntPtr cal_score_ptr,
             float battery,
             int hand_type,
             int radio_channel )
```

The delegate data receive function called inside unmanaged C++ API.

Parameters

pUserParam	Pointer of the user defined parameter which registered previously.
ptr	Array of the data received, where contains all joint rotation values.
data_rate	Data rate per second.
radio_strength	Radio transmission strength in dB
cal_score_ptr	Array of the calibration score received.
battery	Current battery level in percentage.
hand_type	The hand type of current hand pose.
radio_channel	Current radio channel used by wireless transmission.

5.3.3.21 ReceivedEventCallback()

```
delegate void VRTRIX.VRTRIXDataWrapper.ReceivedEventCallback (
             IntPtr pUserParam,
             IntPtr pEvent )
```

The delegate event receive function called inside unmanaged C++ API.

Parameters

pUserParam	Pointer of the user defined parameter which registered previously.
pEvent	Enum of current event received.

5.3.3.22 RegisterDataCallback()

```
static void VRTRIX.VRTRIXDataWrapper.RegisterDataCallback (
             IntPtr pUserParam,
             IntPtr sp,
             ReceivedDataCallback receivedDataCallback )
```

Register receiving and parsed frame calculation data callback

Parameters

pUserParam	User defined parameter/pointer passed into plugin interface, which will return in callback function.
sp	The serial port object
receivedDataCallback	received data callback.

5.3.3.23 RegisterEventCallback()

```
static void VRTRIX.VRTRIXDataWrapper.RegisterEventCallback (
             IntPtr sp,
             ReceivedEventCallback receivedEventCallback )
```

Register receiving hardware event callback

Parameters

sp	The serial port object
receivedEventCallback	received event callback.

5.3.3.24 SetAdvancedMode() [1/2]

```
void VRTRIX.VRTRIXDataWrapper.SetAdvancedMode (
             bool bIsAdvancedMode )
```

Activate advanced mode so that finger's yaw data will be unlocked.

Parameters

bIsAdvancedMode	Advanced mode will be activated if set to true.

5.3.3.25 SetAdvancedMode() [2/2]

```
static void VRTRIX.VRTRIXDataWrapper.SetAdvancedMode (
            IntPtr sp,
            bool bIsAdvancedMode )
```

Set Advanced Mode.

Parameters

sp	The serial port object
bIsAdvancedMode	The boolean value to set

5.3.3.26 SetDistalThumbOffset()

```
static void VRTRIX.VRTRIXDataWrapper.SetDistalThumbOffset (
            IntPtr sp,
            double offset_x,
            double offset_y,
            double offset_z )
```

Set Distal Thumb Offset.

Parameters

offset↩ _x	x-axis offset to set
offset↩ _y	y-axis offset to set
offset↩ _z	z-axis offset to set

5.3.3.27 SetIntermediateThumbOffset()

```
static void VRTRIX.VRTRIXDataWrapper.SetIntermediateThumbOffset (
            IntPtr sp,
            double offset_x,
            double offset_y,
            double offset_z )
```

Set Intermediate Thumb Offset.

Parameters

sp	The serial port object
offset↩ _x	x-axis offset to set
offset↩ _y	y-axis offset to set
offset↩ _z	z-axis offset to set

5.3.3.28 SetProximalThumbOffset()

```
static void VRTRIX.VRTRIXDataWrapper.SetProximalThumbOffset (
             IntPtr sp,
             double offset_x,
             double offset_y,
             double offset_z )
```

Set Proximal Thumb Offset.

Parameters

sp	The serial port object
offset←_x	x-axis offset to set
offset←_y	y-axis offset to set
offset←_z	z-axis offset to set

5.3.3.29 SetThumbOffset()

```
void VRTRIX.VRTRIXDataWrapper.SetThumbOffset (
             Vector3 offset,
             VRTRIXBones joint )
```

Set thumb offset to counteract the difference between hands & gloves sensor installation.

Parameters

offset	Offset vector to set.
joint	the specific thumb joint to set.

5.3.3.30 SetThumbSlerpRate() [1/2]

```
void VRTRIX.VRTRIXDataWrapper.SetThumbSlerpRate (
             double slerp_proximal,
             double slerp_middle )
```

Set thumb slerp rate to counteract the difference between hands & gloves sensor installation.

Parameters

slerp_proximal	Proximal joint slerp rate to set.
slerp_middle	Middle joint slerp rate to set.

5.3.3.31 SetThumbSlerpRate() [2/2]

```
static void VRTRIX.VRTRIXDataWrapper.SetThumbSlerpRate (
             IntPtr sp,
             double slerp_proximal,
             double slerp_middle )
```

Set Thumb Slerp Rate.

Parameters

sp	The serial port object
slerp_proximal	thumb proximal joint slerp rate to set
slerp_middle	thumb middle joint slerp rate to set

5.3.3.32 StartStreaming()

```
static void VRTRIX.VRTRIXDataWrapper.StartStreaming (
             IntPtr sp )
```

Read the data from serial port asynchronously.

Parameters

sp	The serial port object

Returns

Whether the read process successfully

5.3.3.33 VibratePeriod() [1/2]

```
void VRTRIX.VRTRIXDataWrapper.VibratePeriod (
             int msDurationMillisec )
```

Trigger a haptic vibration for a certain period.

Parameters

msDurationMillisec	vibration period

5.3.3.34 VibratePeriod() [2/2]

```
static void VRTRIX.VRTRIXDataWrapper.VibratePeriod (
            IntPtr sp,
            int msDurationMillisec )
```

Vibrate the data glove for given time period.

Parameters

sp	The serial port object
msDurationMillisec	Vibration duration in milliseconds

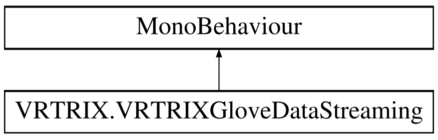
The documentation for this class was generated from the following file:

- VRTRIXBasicDataStreaming/VRTRIXDataWrapper.cs

5.4 VRTRIX.VRTRIXGloveDataStreaming Class Reference

VRTRIX Data Glove data streaming class.

Inheritance diagram for VRTRIX.VRTRIXGloveDataStreaming:



Public Member Functions

- void OnConnectGlove ()
  Connect data glove and initialization.
- void OnDisconnectGlove ()
  Disconnect data glove and uninitialization.
- void OnHardwareCalibrate ()
  Save hardware calibration parameters in IMU, only used in magnetic field changed dramatically.
- void OnVibrate ()
  Trigger a haptic vibration on data glove.
- void OnChannelHopping ()
  Switch radio channel of data glove. Only used for testing/debuging. Automatic channel switching is enabled by default in normal mode.
- void SetAdvancedMode (bool bIsAdvancedMode)
  Activate advanced mode so that finger's yaw data will be unlocked.
- void OnAlignFingers ()

Align five fingers to closed gesture (only if advanced mode is set to true). Also align wrist to the game object chosen.

- Quaternion GetRotation (VRTRIXBones bone)

  Get current rotation of specific joint.
- int GetCalScore (VRTRIXBones bone)

  Get current calibration score for specific IMU sensor.
- int GetReceiveRadioStrength (HANDTYPE type)

  Get radio strength of data glove.
- int GetReceiveRadioChannel (HANDTYPE type)

  Get current radio channel of data glove used.
- float GetBatteryLevel (HANDTYPE type)

  Get current battery level in percentage of data glove.
- int GetReceivedCalScoreMean (HANDTYPE type)

  Get current calibration score average value.
- int GetReceivedDataRate (HANDTYPE type)

  Get data rate received per second.
- bool GetGloveConnectionStat (HANDTYPE type)

  Get data glove connection status.
- VRTRIXGloveStatus GetReceivedStatus (HANDTYPE type)

  Get data glove status.
- VRTRIXGloveGesture GetGesture (HANDTYPE type)

  Get the gesture detected.

Static Public Member Functions

- static GameObject CheckDeviceModelName (HANDTYPE type=HANDTYPE.NONE, InteractiveDevice device=InteractiveDevice.NONE)

  Check the tracked device model name stored in hardware config to find specific hardware type. (SteamVR Tracking support)

Public Attributes

- bool IsVREnabled = false

  VR environment enable flag, set to true if run the demo in VR headset.
- GameObject LH_ObjectToAlign

  If VR is NOT enabled, wrist joint need an object to align, which can be the camera, or parent joint of wrist(if a full body model is used), or can just be any other game objects.
- GameObject RH_ObjectToAlign

  If VR is NOT enabled, wrist joint need an object to align, which can be the camera, or parent joint of wrist(if a full body model is used), or can just be any other game objects.
- Vector3 RHTrackerOffset = new Vector3(0.01f, 0, -0.035f)

  If VR is enabled, HTC tracker is the default wrist tracking hardware, which is fixed to side part of data glove, this offset represents the offset between tracker origin to right wrist joint origin.
- Vector3 LHTrackerOffset = new Vector3(-0.01f, 0, -0.035f)

  If VR is enabled, HTC tracker is the default wrist tracking hardware, which is fixed to side part of data glove, this offset represents the offset between tracker origin to left wrist joint origin.
- GLOVEVERSION version

  Hardware version of VRTRIX data gloves, currently DK1, DK2 and PRO are supported.
- bool IsEnableMultipleGloves

  Mutiple gloves enable flag, set to true if run multiple gloves on the same PC.
- GloveIndex Index

If mutiple gloves mode is enbaled, specify different index for different pair of gloves. Otherwise, just select None.

- Vector3 ql_modeloffset

  Model mapping parameters for left hand, only used when finger joint axis definition is different from wrist joint, otherwise, just set to 0,0,0.
- Vector3 qr_modeloffset

  Model mapping parameters for right hand, only used when finger joint axis definition is different from wrist joint, otherwise, just set to 0,0,0.
- Vector3[ ] ql_axisoffset = new Vector3[3]

  Model mapping parameters for left hand, only used when wrist joint axis definition is different from hardware wrist joint, otherwise, just set to identity matrix {(1,0,0),(0,1,0),(0,0,1)}. Please read the sdk tutorial documentation to learn how to set this parameter properly.
- Vector3[ ] qr_axisoffset = new Vector3[3]

  Model mapping parameters for right hand, only used when wrist joint axis definition is different from hardware wrist joint, otherwise, just set to identity matrix {(1,0,0),(0,1,0),(0,0,1)}. Please read the sdk tutorial documentation to learn how to set this parameter properly.
- Vector3[ ] thumb_offset = new Vector3[3]

  Model mapping parameters for thumb joint, used to tune thumb offset between the model and hardware sensor placement. Please read the sdk tutorial documentation to learn how to set this parameter properly.
- double thumb_proximal_slerp

  Model mapping parameters for thumb proximal joint, used to tune thumb slerp algorithm parameter. Please read the sdk tutorial documentation to learn how to set this parameter properly.
- double thumb_middle_slerp

  Model mapping parameters for thumb middle joint, used to tune thumb slerp algorithm parameter. Please read the sdk tutorial documentation to learn how to set this parameter properly.
- VRTRIXDataWrapper **LH**

5.4.1 Detailed Description

VRTRIX Data Glove data streaming class.

A basic data streaming class for demonstration.

5.4.2 Member Function Documentation

5.4.2.1 CheckDeviceModelName()

```
static GameObject VRTRIX.VRTRIXGloveDataStreaming.CheckDeviceModelName (
            HANDTYPE type = HANDTYPE.NONE,
            InteractiveDevice device = InteractiveDevice.NONE )  [static]
```

Check the tracked device model name stored in hardware config to find specific hardware type. (SteamVR Tracking support)

Parameters

type	Hand type to check(if wrist tracker for data glove is the hardware to check).
device	Device type to check(if other kind of interactive hardware to check).

Returns

the gameobject of the tracked device.

5.4.2.2 GetBatteryLevel()

```
float VRTRIX.VRTRIXGloveDataStreaming.GetBatteryLevel (
             HANDTYPE type )
```

Get current battery level in percentage of data glove.

Parameters

type	Data glove hand type.

Returns

current battery level in percentage of data glove.

5.4.2.3 GetCalScore()

```
int VRTRIX.VRTRIXGloveDataStreaming.GetCalScore (
             VRTRIXBones bone )
```

Get current calibration score for specific IMU sensor.

Parameters

bone	specific joint of hand.

Returns

current calibration score for specific IMU sensor. Lower value of score means better calibration performance.

5.4.2.4 GetGesture()

```
VRTRIXGloveGesture VRTRIX.VRTRIXGloveDataStreaming.GetGesture (
             HANDTYPE type )
```

Get the gesture detected.

Parameters

type	Data glove hand type.

Returns

the gesture detected.

5.4.2.5 GetGloveConnectionStat()

```
bool VRTRIX.VRTRIXGloveDataStreaming.GetGloveConnectionStat (
             HANDTYPE type )
```

Get data glove connection status.

Parameters

type	Data glove hand type.

Returns

data glove connection status.

5.4.2.6 GetReceivedCalScoreMean()

```
int VRTRIX.VRTRIXGloveDataStreaming.GetReceivedCalScoreMean (
             HANDTYPE type )
```

Get current calibration score average value.

Parameters

type	Data glove hand type.

Returns

current calibration score average value.

5.4.2.7 GetReceivedDataRate()

```
int VRTRIX.VRTRIXGloveDataStreaming.GetReceivedDataRate (
            HANDTYPE type )
```

Get data rate received per second.

Parameters

type	Data glove hand type.

Returns

data rate received per second.

5.4.2.8 GetReceivedStatus()

```
VRTRIXGloveStatus VRTRIX.VRTRIXGloveDataStreaming.GetReceivedStatus (
            HANDTYPE type )
```

Get data glove status.

Parameters

type	Data glove hand type.

Returns

data glove status.

5.4.2.9 GetReceiveRadioChannel()

```
int VRTRIX.VRTRIXGloveDataStreaming.GetReceiveRadioChannel (
            HANDTYPE type )
```

Get current radio channel of data glove used.

Parameters

type	Data glove hand type.

Returns

current radio channel of data glove used.

5.4.2.10 GetReceiveRadioStrength()

```
int VRTRIX.VRTRIXGloveDataStreaming.GetReceiveRadioStrength (
            HANDTYPE type )
```

Get radio strength of data glove.

Parameters

type	Data glove hand type.

Returns

radio strength of data glove. Higher value of score means better radio strength.

5.4.2.11 GetRotation()

```
Quaternion VRTRIX.VRTRIXGloveDataStreaming.GetRotation (
            VRTRIXBones bone )
```

Get current rotation of specific joint.

Parameters

bone	specific joint of hand.

Returns

current rotation of specific joint.

5.4.2.12 SetAdvancedMode()

```
void VRTRIX.VRTRIXGloveDataStreaming.SetAdvancedMode (
            bool bIsAdvancedMode )
```

Activate advanced mode so that finger's yaw data will be unlocked.

Parameters

bIsAdvancedMode	Advanced mode will be activated if set to true.

The documentation for this class was generated from the following file:

- VRTRIXBasicDataStreaming/VRTRIXGloveDataStreaming.cs

5.5 VRTRIX.VRTRIXUtilities Class Reference

Bone utility functions class.

Static Public Member Functions

- static string GetBoneName (int id)

  Get current bone name for specific bone ID.
- static int GetBoneIndex (string name)

  Get current bone index for specific bone name.

5.5.1 Detailed Description

Bone utility functions class.

5.5.2 Member Function Documentation

5.5.2.1 GetBoneIndex()

```
static int VRTRIX.VRTRIXUtilities.GetBoneIndex (
            string name )  [static]
```

Get current bone index for specific bone name.

Parameters

name	Bone name.

Returns

current bone index for specific bone name.

5.5.2.2 GetBoneName()

```
static string VRTRIX.VRTRIXUtilities.GetBoneName (
             int id )  [static]
```

Get current bone name for specific bone ID.

Parameters

id	id of bone.

Returns

current bone name for specific bone ID.

The documentation for this class was generated from the following file:

- VRTRIXBasicDataStreaming/VRTRIXUtilities.cs

Index