STEAMVR™

Testing Object Connectivity

SteamVR™ Tracking

Introduction

While the ultimate test of a SteamVR™ trackable object is seeing it appear in virtual reality, there are frequently situations where early prototype hardware fails to track due to hardware issues or software configuration problems that SteamVR™ cannot diagnose. In these situations, an important first step in debugging is to establish that the object is successfully connected to the host machine and sending data to SteamVR™. This will help to rule out electrical problems before focusing on the JSON file and general tracking quality.

Physical Layer Connection

All SteamVR™ objects connect to the host machine either through a hard-wired USB connection, or wirelessly over a USB dongle. This dongle may be discrete, or it may be integrated into another piece of hardware such as a head mounted display (HMD).

USB Connection

To connect over USB, simply plug your device's USB cable into a USB port on the host or on a hub connected to the host. If your device depends on USB for power (i.e. it does not have a battery), make sure to use a USB hub that can provide an appropriate level of power.

If successful, Windows should recognize the device. You can verify the connection to SteamVR™ by following the steps in the "Software Layer Connection" section.

Wireless Connection

If your wireless device is already paired with a wireless dongle, connect the dongle like you would connect a USB device. Note that on many USB3 motherboards, radio interference near USB ports may cause connection issues. If you are having difficulty connecting, move the dongle to an external USB hub.

If your device is not paired, you will need to pair it using lighthouse console and the following steps.

1. Remove all SteamVR™ devices from the host machine except for the dongle you want to pair to the object.
2. Launch `lighthouse_console.exe` which can be found under `C:\Program Files (x86)\Steam\steamapps\common\SteamVR Tracking HDK\tools\bin\win32`. If it fails to launch, check the connection of your dongle to the host machine.
   a. If you were unable to remove all other SteamVR™ devices (for example, if your dongle is integrated into a headset) then follow the instructions in the Software Layer Connection section for connecting to a dongle using the `serial` command.
3. Power on the object by holding the system button for half a seconds. Note that the device will not pair if it is connected over USB.
4. Place your device in pairing mode by holding down the menu and system buttons for 2 seconds until the status LED is steady blue.
5. In lighthouse console, type `pair` and hit enter. If successful, you should see the device LED turn solid green.

```
lh> pair
```

6. If pairing is not successful, verify that your device is in pairing mode and its antenna is attached.

Software Layer Connection

With your SteamVR™ device connected to the host machine, launch `lighthouse_console.exe` which can be found in `C:\Program Files (x86)\Steam\steamapps\common\SteamVR Tracking HDK\tools\bin\win32` and follow these steps to connect to your device.

1. Type the command `serial` and press enter. Lighthouse console displays `Attached lighthouse receiver devices` followed by a list of the serial numbers of all of the devices currently connected. It then connects to the device with the lowest serial number. Note that objects connected over USB devices all have serial numbers with the prefix “`LHR-`”

```
lh> serial
Attached lighthouse receiver devices:
        LHR-684A4E27
        LHR-71D20826
        69F2CECEEE
```

2. If lighthouse_console has not connected to your device, type `serial` followed by the first few characters of your device’s serial number, and press enter. You do not need to enter the entire serial number. Lighthouse console only needs enough characters to uniquely identify it from the other connected devices.
3. If successful, you should see “`Attempting HID Open IMU: <serial number>`” followed by some more USB enumeration details and finally “`Lighthouse IMU HID opened`.” The object is now connected.

```
lh> serial 69F2
Attempting HID Open IMU: 69F2CECEEE
hid_open_nths
        vid=0x28de, pid=0x2101, sn=69F2CECEEE
HID opened: VID 28de PID 2101 serial 69F2CECEEE seq 1 | if -1
Lighthouse IMU HID opened
```

Sensor tests

Once the object is physically connected, you can verify that all of the sensors are configured correctly and behaving nominally. Replacing a faulty sensor or making small changes to your configuration is often all that’s needed to get an object to start tracking when it wasn’t before.

IMU

The IMU is responsible for measuring the rotation and acceleration of your device. Lighthouse console can provide raw data from the IMU as well as statistical information that should help you evaluate its function.

Raw Data

1. Once connected to the object, type `dump` and press enter. Multiple sensor systems on the object have the ability to output sensor data. The `dump` command is a catch-all to enable/disable data output. Once dump is enabled, different sensor systems must be enabled individually.

```
lh> dump
```

2. Type `imu` and press enter to enable IMU data dumping.

```
lh> imu
```

3. You should see a continuous rapid stream of IMU data including rotation and acceleration data. Gyroscope data is provided in degrees per second while IMU data is presented in meters per second. An example line is provided below. Note that you won’t be able to see what you are typing for further commands as your key inputs will be interlaced with the IMU data. Even if you can’t see what you are typing, lighthouse_console is still accepting inputs.

```
0.698666 316531        gyro -0.03 +0.01 +0.02 accel -1.05 +0.43 +9.76
```

4. Type the `imu` command again to disable the dump.
5. The IMU dump is a good way to verify the connection to your device. When shaking your device, you should note fluctuations in the IMU data streaming out.
6. The magnitude of the gyro output should be close to 0°/s, while the magnitude of the accel vector should be close to 9.81 m/s^2.

Statistical data

1. Once connected to your device, enter the `imustats` command to get statistics on the IMU in your device.

```
lh> imustats
```

2. These statistics include the frequency at which the host PC receives IMU data packets, an estimation of the magnitude of the gravity vector, and other statistical information. An example is provided below.

```
imu 183947 rate 995.7Hz interval 1.0ms sigma 0.138ms grav 9.78m/s/s sigma 0.037
```

Optical sensors

In addition to verifying that all of the optical sensors are physically connected and receiving data, the following lighthouse_console commands help identify which physical sensor is associated with which software input. This is critical for finding problems in the JSON file that may prevent the object from tracking.

Optical Sensor Statistics

1. Before testing sensors, make sure that the base stations are turned on and in view of your device.
2. Enter the `dis` command to enable the disambiguator. The disambiguator is responsible for associating incoming sensor data with the appropriate base station.

```
lh> dis
Enabled tdm disambiguator.
```

3. With the disambiguator enabled, enter the `period` command. An example output is provided below. If one sensor is not receiving pulses or it has received substantially fewer than the other sensors even though it has a clear view of the base station, it could be faulty. An example of such an output and a description of the relevant values is provided below.

```
lh> period
base:C4054792 axis:0 min_sensor_ppm: 3.33
 id 3: hits 1 angle 1.88791 sigma 0 var 0 ppm 0.00
 id 4: hits 2136 angle 1.96758 sigma 3.19073e-005 var 1.01807e-009 ppm 5.08
 id 5: hits 4894 angle 1.97183 sigma 2.21614e-005 var 4.91127e-010 ppm 3.53
 id 7: hits 4899 angle 1.97466 sigma 2.27956e-005 var 5.19638e-010 ppm 3.63
 id 9: hits 9800 angle 1.97387 sigma 0.00233606 var 5.45717e-006 ppm 371.80
 id 11: hits 4897 angle 1.98340 sigma 2.21272e-005 var 4.89613e-010 ppm 3.52
 id 13: hits 4898 angle 1.98947 sigma 2.56557e-005 var 6.58217e-010 ppm 4.08
 id 15: hits 4898 angle 1.98512 sigma 2.38761e-005 var 5.7007e-010 ppm 3.80
 id 17: hits 4893 angle 1.99622 sigma 2.09381e-005 var 4.38405e-010 ppm 3.33
 id 19: hits 9552 angle 1.99275 sigma 0.00208552 var 4.34938e-006 ppm 331.92
 id 21: hits 4898 angle 1.99760 sigma 2.3218e-005 var 5.39074e-010 ppm 3.70
base:C4054792 axis:1 min_sensor_ppm: 4.05
 id 4: hits 617 angle 2.04503 sigma 3.16112e-005 var 9.99266e-010 ppm 5.03
 id 5: hits 4896 angle 2.03978 sigma 2.693e-005 var 7.25222e-010 ppm 4.29
 id 7: hits 4899 angle 2.03563 sigma 2.75024e-005 var 7.56381e-010 ppm 4.38
 id 9: hits 9802 angle 2.02408 sigma 0.00707877 var 5.01089e-005 ppm 1126.62
 id 11: hits 4899 angle 2.03676 sigma 2.81081e-005 var 7.90063e-010 ppm 4.47
 id 13: hits 4900 angle 2.03435 sigma 2.57367e-005 var 6.62376e-010 ppm 4.10
 id 15: hits 4899 angle 2.03219 sigma 2.73512e-005 var 7.4809e-010 ppm 4.35
```

```
id 17: hits 4900 angle 2.03668 sigma 2.54739e-005 var 6.48917e-010 ppm 4.05
id 19: hits 9784 angle 2.02350 sigma 0.00528445 var 2.79254e-005 ppm 841.05
id 21: hits 4903 angle 2.03051 sigma 0.000183894 var 3.38168e-008 ppm 29.27
```

- **Base** - identifier of base station
- **Axis** - which laser sweep hit the sensor in question
- **Id** - Sensor number
- **Hits** - number of hits since last clear command.

4. Typing the `period` command again provides an updated version of the same information.
5. Typing `clear` will clear the statistics. Subsequent `period` commands will only show data since the last time `clear` was entered.

```
lh> clear
```

Raw Data

1. Once connected to your device, enter the `dump` command and press enter. Multiple sensor systems on your device have the ability to “dump” raw data. The `dump` command is a catch-all to enable/disable all systems. Once dump is enabled, sensor systems must be enabled individually.

```
lh> dump
```

2. Type `sample` and press enter to enable optical sensor dumping.

```
lh> sample
```

3. You should see a continuous rapid stream of optical data. Each line represents a sync pulse or laser sweep hitting a particular sensor. The sensor number along with the width of the sync pulse/laser sweep are listed. An example is provided below.

```
30.012599 l=28404 r=28544       sample 05 width  140
```

In this example, the sensor number is 5 and the width of the pulse is 140 clock ticks. Note that you won’t be able to see what you are typing for further commands as your key inputs will be interlaced with the optical sensor data. Even if you can’t see what you are typing, lighthouse console is still accepting inputs.

4. This output provides a good opportunity for checking your sensor mapping. By physically obstructing all but one of the sensors, you can limit the data stream to a single sensor and verify that the sensor number that lighthouse console reports matches the expected sensor number from your JSON file and 3D CAD.

5. You can also verify that the sensor data is valid. If the object remains motionless, looking at the duration information should reveal consistent bands of large and small widths that represent the sync pulse and laser sweep. Erratic data may point to a faulty sensor or sensor connection.

6. Type the `sample` command again to disable the dump.

```
lh> sample
```

Wireless Range Test

The information shown in either the IMU or optical sensor dumps represents the raw data delivered to SteamVR™. Therefore, it is a good way to verify that the object has a good connection to the host machine. This is especially important for wireless devices.

The simplest way to test the range of a wireless device is to enable an IMU or optical sensor data dump and then increase the distance between the device and the wireless dongle attached to the host. As you begin to exit the wireless range of this connection, you should see the stream of sensor data begin to slow down or stop entirely.