

Using the Heart Rate Monitor plugin for HoloLens

Plugin v1.2.0.0

contact: customerservice@roguish.com

Release Notes

1.1.2.0

- Original release

1.2.0.0

- Archive and Expose the un-processed Byte Array data received from the HRM device in a List: receivedMeasurementData
- Added customizable size limit to the hrms and receivedMeasurementData Lists
- Added VERSION string

Plugin Overview

The Heart Rate Monitor plugin for Unity3D enables apps built for HoloLens devices to receive data from an external Bluetooth LE Heart Rate Monitor.

HRM Device

Any Heart Rate Monitor (HRM) device that broadcasts using the standard Bluetooth LE Heart Rate Measurement Characteristic should work with the Heart Rate Monitor plugin. Find more information about the Bluetooth Heart Rate Measurement characteristic here:

[https://www.bluetooth.com/specifications/gatt/viewer?](https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.characteristic.heart_rate_measurement.xml)

[attributeXmlFile=org.bluetooth.characteristic.heart_rate_measurement.xml](https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.characteristic.heart_rate_measurement.xml)

Most modern chest-worn HRMs broadcast using the Bluetooth LE Heart Rate Measurement characteristic (Polar H7, Wahoo (several models), Zephyr HxM Smart, etc.). Wrist-worn devices often do not broadcast using the Bluetooth LE Heart Rate Measurement characteristic, though some do (Mio Alpha 2, Scosche RHYTHM+, etc.). HRM devices that work with mobile apps like Polar Beat will work with the plugin.

Pair your HoloLens device with any nearby HRM devices you would like to connect with. Bluetooth pairing settings may be found in the HoloLens Settings menu > Devices > Bluetooth. Your HoloLens device may be paired with more than one HRM device, and the Heart Rate Monitor plugin may connect with one of the paired devices at a time.

Plugin Installation

Find the HeartRateMonitor_WSA_Plugin folder. In that folder is a Plugins folder. Copy the whole Plugins folder and all of its contents into the Assets folder of your Unity Project.

High-Level Description of Plugin Use

- Pair HRM Bluetooth LE device with HoloLens
- Scan for paired HRM devices
- Connect with a paired HRM device
- Detect intermittent updates that are broadcast from HRM device and display latest HR data as updates are received. HRM devices broadcast updates approximately once per second.

Unity Project Configuration

- Set all the HoloLens settings as described in the HoloLens documentation.
- To enable Bluetooth, select the following checkbox: Edit>Project Settings>Player>Publishing Settings>Capabilities>Bluetooth
- Place the plugin files in the Assets folder. The WSAUnityHRM.dll file should be in the Assets>Plugins folder and the similarly-named WSAUnityHRM.dll file inside the WSA folder should remain in the WSA folder (the WSA folder should be copied into Assets > Plugins).
- The sample Unity project requires the HoloToolkit from Microsoft. It was built with the HoloToolkit that includes this note in the ReadMe: “Current Unity Editor Project Version: 5.5.0f3”. The toolkit has been included in the sample project. The most recent toolkit may be downloaded here: <https://github.com/Microsoft/HoloToolkit-Unity>

API

Public Methods

public void scan(Queue<Action> pActionQueue = null, Action pActionMethod = null)

Find nearby devices that use the Bluetooth Heart Rate Measurement Characteristic. Optionally pass a reference to a Queue of Actions and an Action. When the plugin has finished scanning for devices the Action that was specified will be enqueued in the specified Queue. You may detect changes in the count of the Queue, dequeue the actions and call them. Alternately, you can detect that the scan is complete by polling the field *isScanComplete*.

public void initializeService(HeartRateServiceDevice pDevice, Queue<Action> pActionQueue = null, Action pActionMethod = null)

Pass the preferred *HeartRateService* Device you want to connect with. (Retrieve the list of available *HeartRateService* devices from the field *hrsDevices* after the device scan is complete). Optionally pass a reference to a Queue of Actions and an Action. The Action will be enqueued into the Queue whenever a new HR value is ready (Approximately once per second). You must manually dequeue actions from the Queue and call them.

public void disconnectService()

Disconnects the Heart Rate Measurement Characteristic if one is connected.

public bool getHasNewHRValue()

Returns *true* there is a new HR value to retrieve. Used when manually polling the plugin.

public ushort getLastHRM(Boolean pResetAfterRetrieval = true)

Returns the most recent HR value if there is a new HR value to return. If no new value is ready the return value is 0. Normally this method is called after you detect that there is a new HR value, so a return value of 0 is uncommon. You must pass *true* as the *pResetAfterRetrieval* parameter to refresh the plugin state to prepare it to be able to indicate that a new value has been received the next time the plugin receives a message from the HRM device (or optionally you may call *setHasNewHRValue* with its parameter set to *false* to refresh the plugin).

public void setHasNewHRValue(Boolean pHasNewValue = true)

Sets the plugin to indicate that there is/is not a new HR value ready to retrieve.

Public Fields

public bool isScanComplete

True when the scan for paired HRM devices has finished

public bool isServiceInitialized

True when an HRM device is connected and broadcasting HR values.

public List<ushort> hrms

Archives HR values received. New entries are added to the end.

public List<HeartRateServiceDevice> hrsDevices

The list of *HeartRateServiceDevice* entries representing HRM devices paired with the HoloLens and discovered during the scan.

public List<byte[]> receivedMeasurementData

Archives received data byte arrays. New entries added to the end.

public int HrmStorageMax

Get/Set the maximum number of HR values stored in the hrms List. Defaults to 100.

public int ReceivedMeasurementDataStorageMax

Get/Set the maximum number of byte array values stored in the receivedMeasurementData List. Defaults to 100.

public const string VERSION

Version number.