

Using the Heart Rate Monitor plugin for HoloLens and Windows 10

Plugin v1.3.4.0

contact: customerservice@roguish.com

Release Notes

1.3.4.0 (2018/04)

- Updated and tested for the latest Microsoft Mixed Reality Toolkit, Unity, Windows OS, HoloLens OS, etc. Details below.
- Added Windows 10 examples.
- Expanded HRM device compatibility. Now compatibility with Zoom HRV watch and probably other HRMs that previously may not have worked.
 - The Plugin now simply disregards the flags byte and returns the value of the second byte as the heart rate. This is valid for most devices, however the full Bluetooth spec for Heart Rate Service allows the heart rate to be returned as 2-bytes, allowing for much faster heart rates (mainly for animals). Unfortunately, the ZoomHRV returns a flag byte of 00, which is pretty much nonsense according to the BLE spec and was causing an attempt to read heart rate from 2-bytes, but there was only one to read from. Also, in HRV mode the Zoom HRV broadcasts a flag byte of 10, which matches a SCOSCHE armband I recently tried. The new plugin has been tested with the Zoom HRV and works in both modes (active workout and HRV mode) that use flags 00 and 10, so it should work with a SCOSCHE armband as well.
 - I also removed the application of a gatt protection level. In prior versions it was set to expect an encrypted communication channel. Removing that was necessary to connect with the Zoom HRM, so I suspect their broadcast is not using the BLE encryption setting. For users who want to use one of the protection levels I've enabled it with a property "gattProtectionLevel" specified below.
- Updated to conform to new Name reporting of Windows BLE. DeviceInformation.Name used to return the displayable name of the device, but now it returns "Heart Rate". The displayable name must be retrieved for each DeviceInformation object in the list.
- Added Advertisement Listening for RSSI updating.
- Restricted the byte array data returned with receivedMeasurementData to only the first 2 bytes of the data (the flags byte and the heart rate byte). Please use receivedMeasurementDataLimited now. Users who want RR data please contact me for a pro version of the plugin.

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

1.1.2.0

- Original release

Plugin Overview

The Heart Rate Monitor plugin for Unity enables apps built for HoloLens and Windows 10 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?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+, Polar OH1, Zoom HRV, etc.). HRM devices that work with mobile apps like Polar Beat should 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 only one of the paired devices at a time.

Compatibility Testing

- Polar H7 – Chest strap. The benchmark device I test with. Full support.
- Zephyr HxM Smart – Tested on versions of the plugin prior to v1.3.4.0.
- Zoom HRV – Wrist-worn PPG based HRM. Works.
- Polar OH1 – Tested on Windows by a customer and confirmed to work.
- Scosche Rhythm+ – Untested, however verified that the device sends data with a 00 flag byte followed by heart rate byte, so it should work as well as the Zoom HRV and Polar OH1 which both use the same 00 flag byte pattern.

There are several applications I recommend for testing your HRM device's BLE capabilities:

- Android/iOS – nRF Connect from Nordic Semiconductor:
 - <https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=en>
 - <https://itunes.apple.com/us/app/nrf-connect/id1054362403?mt=8>
- Microsoft Bluetooth LE Explorer for Windows 10
 - <https://www.microsoft.com/en-us/store/p/bluetooth-le-explorer/9n0ztkf1qd98>

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

- See Publication Sample Screenshot Images in the delivery package
- 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).
- Toolchain
 - Mixed Reality Toolkit v.2017.2.1.3 Hot Fix
<https://github.com/Microsoft/MixedRealityToolkit-Unity/releases/tag/2017.2.1.3>
The toolkit has been included in the sample project.
 - Unity Editor v.2017.2.1p2
<https://unity3d.com/unity/qa/patch-releases/2017.2.1p2>
 - Visual Studio Community 2017 (15.6.1)
 - .NET Framework v.4.7.02556 (in Visual Studio About panel)
 - HoloLens OS v.10.0.14393.2125 (version: Settings > Software > OS build)
 - Windows 10 v.1709 (Fall Creator's)
 - When upgrading any part of the toolchain, be sure to read the latest information about compatibility. The Unity version, Mixed Reality Toolkit (MRTK is the new name for the old HoloToolkit) and even Windows version must all be compatible. Sometimes a new/latest version of one will come out and break compatibility with the others so don't assume that the latest version of each part of the toolchain will guarantee compatibility.

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 enum PluginGattProtectionLevel

Values that mirror the Microsoft GattProtectionLevel enum values, and one additional value: Omit

<https://docs.microsoft.com/en-us/uwp/api/windows.devices.bluetooth.genericattributeprofile.gattprotectionlevel>

public PluginGattProtectionLevel gattProtectionLevel

No ProtectionLevel is specified on the Characteristic unless you change this field. It is set to PluginGattProtectionLevel.Omit by default. To use any of the 4 Microsoft protection levels, use a PluginGattProtectionLevel enum with a matching name. The plugin will apply the Microsoft equivalent named enum as the Characteristic's ProtectionLevel. Set this before calling initializeService.

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.

[REMOVED IN FREE VERSION – use receivedMeasurementDataLimited instead]

public List<byte[]> receivedMeasurementData

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

public List<byte[]> receivedMeasurementDataLimited

Archives received data byte arrays. New entries added to the end. Stores the first 2 bytes of each data

packet: Flags and Heart Rate. Subsequent values in the data are not stored in the Free version. S Pro version is available broadcasts the full data packet including RR values.

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 PluginGattProtectionLevel gattProtectionLevel

Optional setting for the Gatt Protection Level to set when connecting with a device. Mirrors the Microsoft enums of a similar name.

public bool useWatchAdvertisements = true

If True, after scan is complete an AdvertisementWatcher is started which will continually watch for new Advertisements from nearby BLE devices. This is useful for monitoring the RSSI Signal Strength of nearby devices. You may find the RSSI value for a device like this:

`_hrmPlugin.hrsDevices[0].properties["RSSI"]`

public const string VERSION

Plugin version number.