Documentation HeartRatePlugin ver. 14

Content:

Important Notice:

This version differs in structure basically to the previous versions!

Description

This plugin scans, connects and reads data from a heart rate sensor(hrs), based on events.
You can connect to multiple hrs at the same time

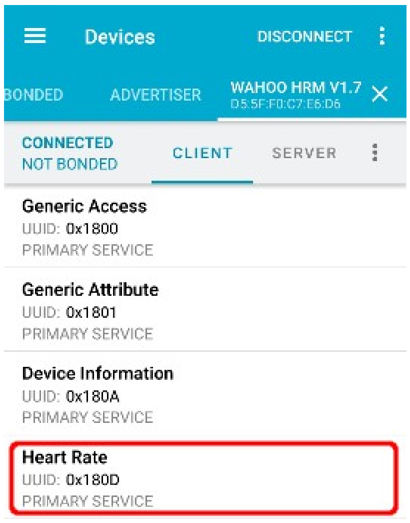
Requirements

- Bluetooth and GPS have to be turned on
- The heart rate sensor(hrs) has to use the standard heart rate protocol defined by SIG:
    Heart Rate Service: 0x180D
    Heart Rate Measurement Characteristics: 0x2A37

How to check, if your device supports the standard heart rate protocol:

- Install nRF Connect app
- connect to your hrs

Heart Rate should be displayed



Generic Access
UUID: 0x1800
PRIMARY SERVICE

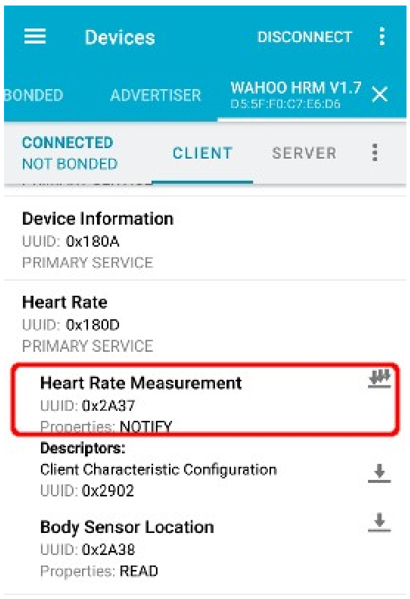
Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

Device Information
UUID: 0x180A
PRIMARY SERVICE

Heart Rate
UUID: 0x180D
PRIMARY SERVICE

- Tap on Heart Rate

Heart Rate Measurement should be displayed



Device Information
UUID: 0x180A
PRIMARY SERVICE

Heart Rate
UUID: 0x180D
PRIMARY SERVICE

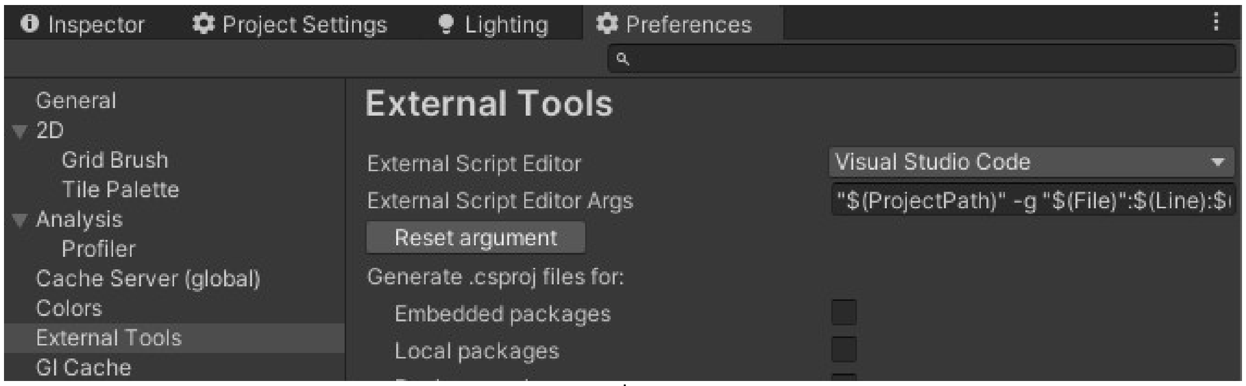
Heart Rate Measurement
UUID: 0x2A37
Properties: NOTIFY
Descriptors:
Client Characteristic Configuration
UUID: 0x2902

Body Sensor Location
UUID: 0x2A38
Properties: READ

Preferences

Android:

- Check out Unity/Preferences
- In External Tools tab under Android please be sure to have all checkboxes activated



External Tools

External Script Editor: Visual Studio Code

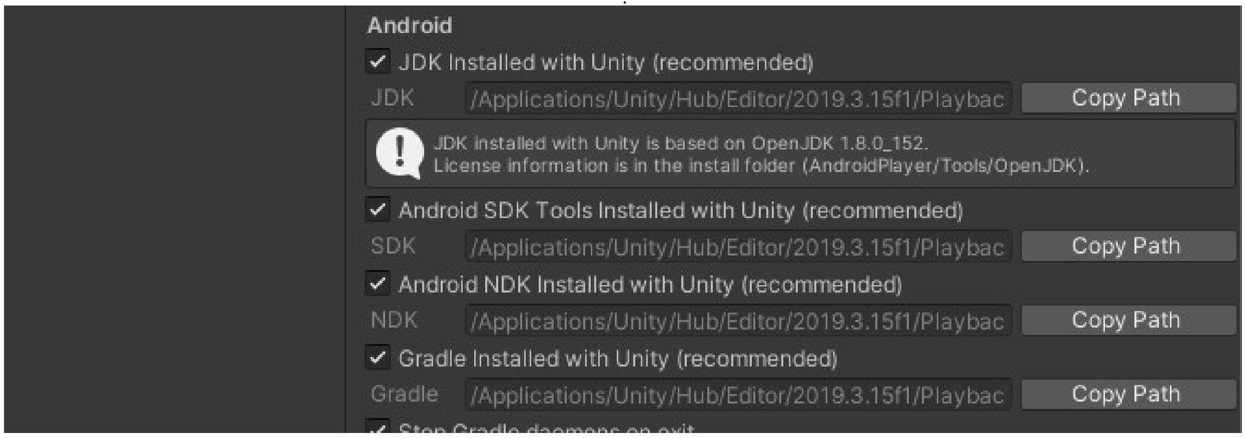
External Script Editor Args: "$(ProjectPath)" -g "$(File)":$(Line):$

Reset argument

Generate .csproj files for:

Embedded packages

Local packages



Android

✓ JDK Installed with Unity (recommended)

JDK /Applications/Unity/Hub/Editor/2019.3.15f1/Playbac Copy Path

JDK installed with Unity is based on OpenJDK 1.8.0_152.
License information is in the install folder (AndroidPlayer/Tools/OpenJDK).

✓ Android SDK Tools Installed with Unity (recommended)

SDK /Applications/Unity/Hub/Editor/2019.3.15f1/Playbac Copy Path

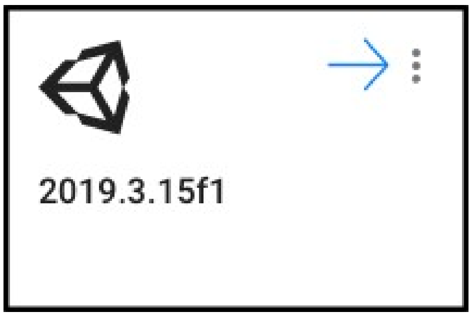
✓ Android NDK Installed with Unity (recommended)

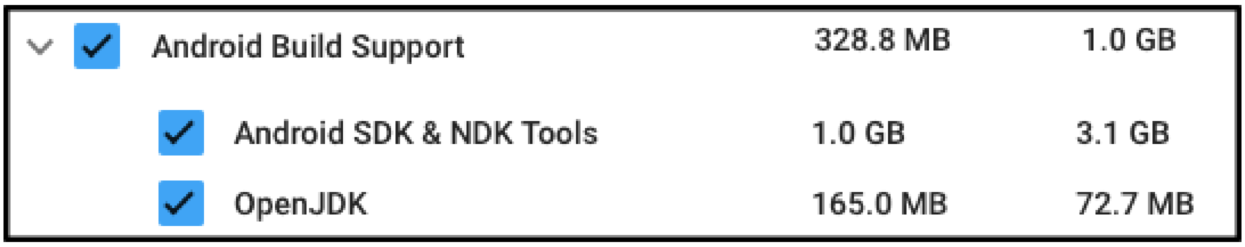
NDK /Applications/Unity/Hub/Editor/2019.3.15f1/Playbac Copy Path

✓ Gradle Installed with Unity (recommended)

Gradle /Applications/Unity/Hub/Editor/2019.3.15f1/Playbac Copy Path

- Please check in Unity Hub, if all moduls for Android Build Support are added.





✓	Android Build Support	328.8 MB	1.0 GB
✓	Android SDK & NDK Tools	1.0 GB	3.1 GB
✓	OpenJDK	165.0 MB	72.7 MB

- Check, if the correct platform for the plugin is set:
  - select Assets>HeartRatePlugin>Scripts>Bridge>HeartRateMulti
  - In the Inspector under Select platforms for plugin, Android has to be selected (nothing else)

MacOS:

- Check, if the correct platform for the plugin is set:
  - select Assets>HeartRatePlugin>Scripts>Bridge>MacOSHeartRatePlugin.bundle
  - In the Inspector under Select platforms for plugin, Editor and Standalone have to be selected (nothing else)

iOS:

- Check, if the correct platform for the plugin is set:
  - select Assets>HeartRatePlugin>Scripts>Bridge>libiOSHeartRatePlugin
  - In the Inspector under Select platforms for plugin, iOS has to be selected (nothing else)

HeartRatePlugin.Event:

Properties:

- **MacID**:
  - Identifier(iOS) or MAC-Adress(Android) of the sensor, who raised the Event
- **Info**:
  - Information depending on the EventType
- **Type**:
  - SYSTEM_NOT_SCANNING
    - raised, if scan is stopped. Contains reason in Info
  - SYSTEM_SCANNING
    - raised, if scan is started
  - NEW_SENSOR
    - raised, if a new hrs was found upon scanning
  - REMOVE_UNCONNECTED
    - raised, if a sensor is removed from HeartRateSensor-Model
  - CONNECTING
    - raised, if a sensor is connecting
  - CONNECTED
    - raised, if a sensor is connected
  - DISCONNECTED
    - raised, if a sensor got disconnected
  - NOTIFICATION_CONTROLPOINT
    - raised, if Heart Rate Control Point is detected
  - NOTIFICATION_BODYSENSORLOCATION
    - raised, if Body Sensor Location is detected
  - NOTIFICATION_MEASUREMENT
    - raised, if a sensor send measurement data

Model

The HeartRateSensor-model contains informations and data of the hrs

Fields:

- **MacId**:
  - Identifier(iOS) or MAC-Adress(Android)
- **Name**:
  - Name the sensor provides
- **Rssi**:
  - Distance from hrs to MobileDevice
- **IsConnecting**:
  - System started a connection to the hrs
- **IsConnected**:
  - Connection was successfully established
- **PulseRate**:
  - Actual pulse rate the hrs send
- **SCStatus**:
  - Sensor contact status of the hrs:
    - NOT_SUPPORTED
    - NOT_SUPPORTED_1
    - NO_CONTACT
    - CONTACT
- **EnergyExpended**:
  - actual amount of expended energy. Null if not present
- **RrInterval**:
  - Array of last heart rate variability. Null if not present
- **SensorLocation**:
  - Body sensor location of the hrs. Mandatory.
    - OTHER
    - CHEST
    - WRIST
    - FINGER
    - HAND
    - EARLOBE
    - FOOT
- **HR_ControlPoint**:
  - Control point of the Alert Notification server. Null if not supported

All discovered sensors are stored in the Dictionary HeartRateSensor.Sensors[key].
Key: MacId.

Methods

- **public void StartScan():**
  Checks if mobile device is capable to scan for bluetooth devices and starts the scan process.
  Sets IsScanning to true.
  At Start, all unconnected hrs are removed from HeartRateSensor.Sensors
  EventType.SYSTEM_SCANNING is raised.
- **public void StopScan()**
  Stops the scan process.
  Sets IsScanning to false.
  EventType.SYSTEM_NOT_SCANNING is raised.
- **public void Connect(string MacId)**
  Connects to the given MacId.
  Sets HeartRateSensor.Sensors[MacId].IsConnecting to true.
  EventType.CONNECTING is raised
- **public void Disconnect(string MacId, bool All)**
  Disconnects the given MacId
  All: all hrs connected to the mobile device will be disconnected

Properties

- **IsInitialized**
  Returns true if the plugin is completely initialized
- **IsScanning**
  Returns true if system is scanning

How to use the HeartRatePlugin

Please take a look at the example scenes.

1. Create an empty object in your scene
2. It is mandatory to name it exactly HeartRatePlugin and attach HeartRatePlugin.cs to it (Assets/HeartRatePlugin/Scripts/HeartRatePlugin.cs)

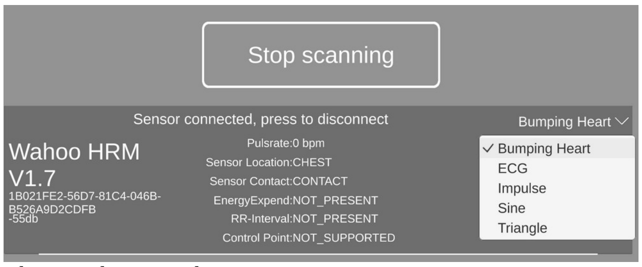
Setup for your scene script:

3. Create reference to HeartRatePlugin:
   [SerializeField]
   private HeartRatePlugin heartRatePlugin;
4. Drag HeartRatePlugin-object to the property in the Inspector of your scene script
5. Create method:
   void OnHeartRateEvent(object sender, HeartRatePlugin.EventArgs e) with switch(e.Type) as shown in SuperSimpleExample.cs
6. In your Start() or OnEnable() (or where you want to), attach HeartRatePlugin.Event to this method:
   HeartRatePlugin.Event += OnHeartRateEvent;
7. Start scanning with:
   heartRatePlugin.StartScan();
8. React to the arriving Events:
   E.g. ff you get EventType.NEW_SENSOR you can connect to it via:
   heartRatePlugin.Connect(e.MacId);
9. Settings of the sensor arrive at:
   - EventType.NOTIFICATION_CONTROLPOINT:
     check if Heart Rate Control Point is supported at HeartRateSensor.Sensors[e.MacId].HR_ControlPoint.
     O or Null means NOT_SUPPORTED
   - EventType.NOTIFICATION_BODYSENSORLOCATION:
     check what sensor location the hrs is attached to at HeartRateSensor.Sensors[e.MacId].SensorLocation.
10. Data arrives at EventType.NOTIFICATION_MEASUREMENT:
    Using e.MacID you can access all data at HeartRateSensor.Sensors[e.MacId].Foo (→ Fields)

Examples

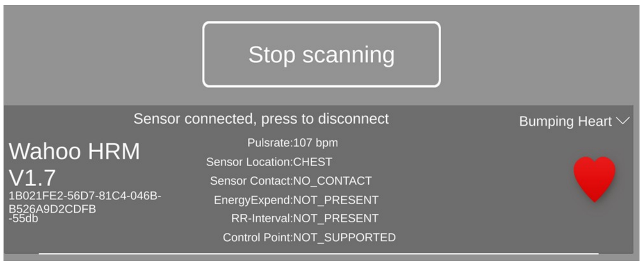
Two Example scenes are included.

- **MultiConnection:**
  This Example shows, what this plugin is capable of. Some visualization styles are included:



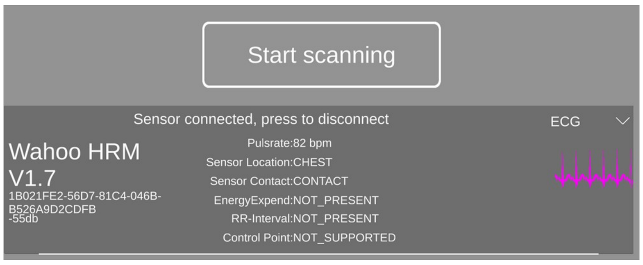
Sensor connected, press to disconnect
Wahoo HRM V1.7
1B021FE2-56D7-81C4-046B-B526A9D2CDFB
-55db
Pulsrate:0 bpm
Sensor Location:CHEST
Sensor Contact:CONTACT
EnergyExpend:NOT_PRESENT
RR-Interval:NOT_PRESENT
Control Point:NOT_SUPPORTED

select visualization style



Sensor connected, press to disconnect
Wahoo HRM V1.7
1B021FE2-56D7-81C4-046B-B526A9D2CDFB
-55db
Pulsrate:107 bpm
Sensor Location:CHEST
Sensor Contact:NO_CONTACT
EnergyExpend:NOT_PRESENT
RR-Interval:NOT_PRESENT
Control Point:NOT_SUPPORTED

e.g. Bumping Heart



Sensor connected, press to disconnect
Wahoo HRM V1.7
1B021FE2-56D7-81C4-046B-B526A9D2CDFB
-55db
Pulsrate:82 bpm
Sensor Location:CHEST
Sensor Contact:CONTACT
EnergyExpend:NOT_PRESENT
RR-Interval:NOT_PRESENT
Control Point:NOT_SUPPORTED

e.g. Ecg

- **SuperSimpleExample:**
  Pure scan and connect, data is logged

  Setup for SuperSimpleExample:

1. Create empty object
2. Name it as HeartRatePlugin
3. Attach HeartRatePlugin.cs to it (Assets/HeartRatePlugin/Scripts/HeartRatePlugin.cs)
4. Attach SuperSimpleExample.cs
5. Drag Object HeartRatePlugin to Script SuperSimpleExample
6. Take a look at the logs

Known issues

- If you run unity on a windows machine, you can build for iOS and Andoid, but you can’t play in editor.
- Please be sure, you choose the correct platform.