

WASP Aggregator 1.4.3 – User Guide

Introduction

The WASP Aggregator is an application that runs on a Raspberry PI 3B+. It receives ANT+ messages transmitted over Ethernet by a WASP using the WASP Packet Protocol using Multicast UDP messages. Each ANT+ message is converted and reformatted as structured JSON data. This JSON data is available to any connected client via a WebSocket.

Setup

- Plug the micro-USB plug from the power supply into the WASP Aggregator.
- Plug an Ethernet cable from your network into the WASP Aggregator.
- Plug the power supply into an outlet.
- The WASP Aggregator will:
 - Boot its operating system and start the application.
 - Get an IP address from your network via DHCP.
 - Begin processing ANT+ packets from the WASP(s).
 - Begin listening for a WebSocket connection from a client.
- Optionally you may connect an HDMI monitor to the WASP aggregator to observe the operation of the application and debug operational issues.

Features

The WASP Aggregator does not convert all ANT+ messages into JSON. It specifically processes messages from the following ANT+ sensors:

- Bicycle Power Device Profile
 - Power Only Sensor Data
 - Torque Sensor Data (wheel torque)
- Bike Speed and Cadence Device Profile
 - Bike Speed Sensor Data
 - Bike Cadence Sensor Data
 - Combined Bike Speed and Cadence Sensor Data
- Heart Rate Device Profile

To receive the JSON data over Ethernet, a client must connect to the WASP Aggregator with a WebSocket. Only one client can be connected at a time. Subsequent connect requests will be queued and when the active connection disconnects the first request in the queue will be connected.

The WASP Aggregator also generates ANT+ heart rate messages in the Fitcare RC900 format.

The WASP Aggregator will respond to the local address WASPAggregator.local. If you have more than one you will need to determine the IP address of each one to connect to a specific unit. If an HDMI monitor is connected to the WASP Aggregator the IP address will be displayed.

JSON Data Format

The JSON output consist of three parts: 1) the version of the JSON output, 2) an array of sensor data objects, and 3) the sensor specific data. The first two parts have the same format for all types of sensors. The following example shows the output format of version 1.1 JSON data.

Note: the data for manufacturer_id and serial_number is provided on background pages and they may take a while to be populated correctly, since the data is transmitted infrequently by the sensor.

```
{ "1.1": {  
  "sensors": [  
    {  
      "device_type" : "<sensor specific>",  
      "rssi" : 19,  
      "device_number" : 43139,  
      "manufacturer_id" : 0,  
      "serial_number" : 0,  
      "timestamp" : 1617952523171,  
      "packets_per_second" : 7.9,  
      "sensor_data" : { <sensor specific> }  
    }  
  ]  
}}
```

Heart Rate Data

```
"device_type":"heartrate",  
"sensor_data" : {  
  "heartrate" : { "value" : 72, "symbol" : "BPM", "valid" : true }  
}
```

Bike Power Only

```
"device_type":" bike_power ",
```

```
"sensor_data" : {  
  "cadence" : { "value" : 72, "symbol" : "RPM", "valid" : true },  
  "power" : { "value" : 72, "symbol" : "W", "valid" : true },  
  "accumulated_power" : { "value" : 72, "symbol" : "W" },  
  "accumulated_event_count" : { "value" : 72, "symbol" : "Events" }  
}
```

Bike Power Wheel Torque

```
"device_type":" bike_power ",  
"sensor_data": {  
  "cadence" : { "value" : 72, "symbol" : "RPM", "valid" : true },  
  "distance" : { "value" : 1.5, "symbol" : "m" },  
  "speed" : { "value" : 12.1, "symbol" : "km/h" },  
  "accumulated_event_count" : { "value" : 72, "symbol" : "Events" }  
}
```

Bike Cadence

```
"device_type":" bike_cadence ",  
"sensor_data": {  
  "cadence" : { "value" : 72.0, "symbol" : "RPM" },  
  "cadence_event_time" : { "value" : 72.0, "symbol" : "S" },  
  "accumulated_cadence_revolutions" : { "value" : 72, "symbol" : "Events" }  
}
```

Bike Speed

```
"device_type":" bike_speed ",  
"sensor_data": {  
  "speed" : { "value" : 72.0, "symbol" : "m/s" },  
  "speed_event_time" : { "value" : 72.0, "symbol" : "S" },  
  "accumulated_speed_revolutions" : { "value" : 72, "symbol" : "Events" }  
}
```

Bike Speed and Cadence

```
"device_type":" bike_speed_cadence ",
```

```
"sensor_data": {  
  "cadence" : { "value" : 72, "symbol" : "RPM" },  
  "speed" : { "value" : 72.0, "symbol" : "m/s" },  
  "cadence_event_time " : { "value" : 72.0, "symbol" : "S" },  
  "speed_event_time " : { "value" : 72.0, "symbol" : "S" },  
  " accumulated_cadence_revolutions " : { "value" : 72, "symbol" : "Events" },  
  " accumulated_speed_revolutions " : { "value" : 72, "symbol" : "Events" }  
}
```

Frequently Asked Questions

- Can I use this SD card image on a Raspberry PI 3B?

No, this image will only work on a Raspberry PI 3B+. The hardware of the 3B+ is significantly different than the 3B or any other model of the Raspberry PI.

- It's not working, now what should I do?

The easiest way to debug problems with the WASP Aggregator is to hook up an HDMI monitor. It will show you:

- If the operating system boots.
 - If the application starts up.
 - The IP address that the WASP Aggregator was assigned by your DHCP server.
 - If the application is receiving data from the WASP(s).
 - If the application is receiving and accepting connection requests.
- What if the operation system doesn't boot?

If the operating system does not boot, then you probably have a bad (or missing) SD card. First make sure that the SD card is present. Turn off the power, remove and re-insert the SD card. Reapply power. If the operating system still doesn't boot, then you need a new SD card.

- What if the application doesn't run?

This is basically the same problem as the previous question – a bad SD card.

- What if no IP address is displayed?

If the WASP Aggregator didn't get an IP address, then you've got a network problem. First check that you connected the Ethernet cable from the WASP Aggregator to your network. Second, the WASP Aggregator uses DHCP to get an IP address, so verify that your network DHCP server is working and that it is responding to the WASP Aggregator.

- What if I have an IP address, but it's not seeing my client's connect requests?

Verify that the client and the WASP Aggregator are connected to the same IP network. If you are trying to use the local address (WASPAggregator.local) then try the IP address instead. Use an IP ping command on the client and verify that the client can get to the WASP Aggregator.

- What if I have an IP address, but I'm not seeing any data from the WASP(s)?

This is similar to the previous question – it's probably a network issue where the WASP and the WASP Aggregator are not on the same IP network. But it might be because there just aren't any ANT+ sensors sending data that we care about. Verify that you have a device that contains one of the ANT+ sensors mentioned in the features section.