Bohunt School *(Wokingham)*

Solar-Powered Weather Station

Stage 4: Measuring and reporting air temperature, humidity and pressure

I'm one of the students who has been attending the (after-school) IoT Computer Club at Bohunt Wokingham academy here in the UK.

This is a write-up of my personal project to design and build a solar-powered weather station.

Objectives

My overall objective is to build a solar-powered weather station that will be located in one of the school's gardens to measure temperature, humidity and air pressure. The data will be sent via WiFi to the 'Cloud' and by writing some programs the information should be made available locally or remotely.

The task I'm going to describe in this particular document is:

- ***Measuring and reporting air temperature, humidity and pressure and sending this data to the 'Cloud'***

Compared to some of the other tasks I've documented, this one should be a lot easier as there are only three main parts to cover.

- **Setting up the Device Drivers in the Wemos**
- **Publishing the transducer values to BeeBotte**
- **Subscribing to BeeBotte in Node-RED**

Setting-up the Device Drivers in the Wemos

Here's a screen-shot showing the two auxiliary devices I used in the weather station, namely the BME environment sensor (BME280) and the 4-channel analog to digital converter (ADS1115).

Edit	8	✓	Environment - BMx280	bme280			GPIO-4 GPIO-5	temperature: humidity: pressure:	26.07 40 1024
Edit	9	✓	Analog input - ADS1115	analog_3			GPIO-4 GPIO-5	a3:	1.18
Edit	10	✓	Analog input - ADS1115	analog_2			GPIO-4 GPIO-5	a2:	1.18
Edit	11	✓	Analog input - ADS1115	analog_1			GPIO-4 GPIO-5	a1:	4.11
Edit	12	✓	Analog input - ADS1115	analog_0			GPIO-4 GPIO-5	a0:	3.28

Powered by **Let's Control It** community

The ADS1115 was mainly used during the experiments I conducted with the solar-panel and the Li-Ion battery as explained in the Stage-3 document.

Bohunt School *(Wokingham)*

Solar-Powered Weather Station

Stage 4: Measuring and reporting air temperature, humidity and pressure

Once I've got the weather station working satisfactorily, I might investigate attaching some soil moisture sensors to the system. These sensors produce a voltage proportional to the moisture content in the soil, so the ADS1115 could be used to handle these readings (i.e. convert from analog to digital values).

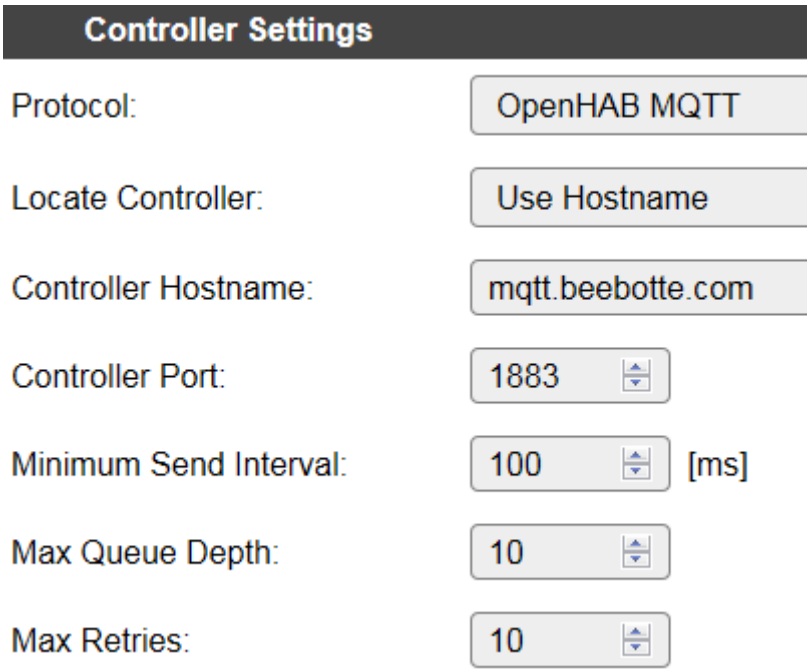
Task 8, on the previous page, creates a device named 'bme280' with variables called 'temperature', 'humidity' and 'pressure'. Each variables can be referenced using <device-name#variable-name>.
For example, 'bme280#temperature' or 'bme280#humidity'.

Tasks 9, 10, 11 and 12 create a device for each of the four channels that are available within the ADS1115. I named the devices as 'analog_0' through to 'analog_3' and the variables for the channels as 'a0', 'a1', 'a2' and 'a3'. So, in the same way as the above, the variables can be referenced using <device-name#variable-name>. Here's an example, 'analog_0#a0' or 'analog_3#a3'.

Publishing the transducer values to MQTT

As mentioned on page-7 in the Stage-2 document, I wanted to be able to see the various readings from the Wemos at my home as well as school, so I registered for an account with BeeBotte at... https://beebotte.com

Here's a screen-shot of the Controller Settings inside the Wemos.



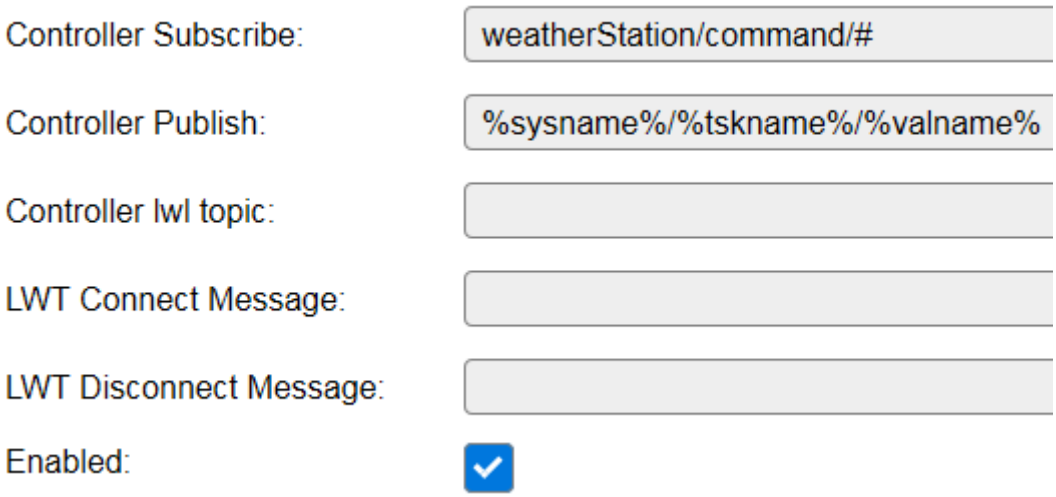
Basically the settings are similar to the ones you would enter for a local MQTT broker except instead of using an IP address you use a Hostname.

Bohunt School *(Wokingham)*

Solar-Powered Weather Station

Stage 4: Measuring and reporting air temperature, humidity and pressure

Here's the second part of the Controller Settings.



As can be seen from the first entry in the screen-shot, the Wemos is listening for any messages sent to the topic 'weatherStation/command'. Although there is an entry for Controller Publish, I didn't make use of this as all my publishing was done using one of the Rule-Sets within the Wemos (as described below).

Here's a listing of the rule-set to publish to MQTT and ThingSpeak

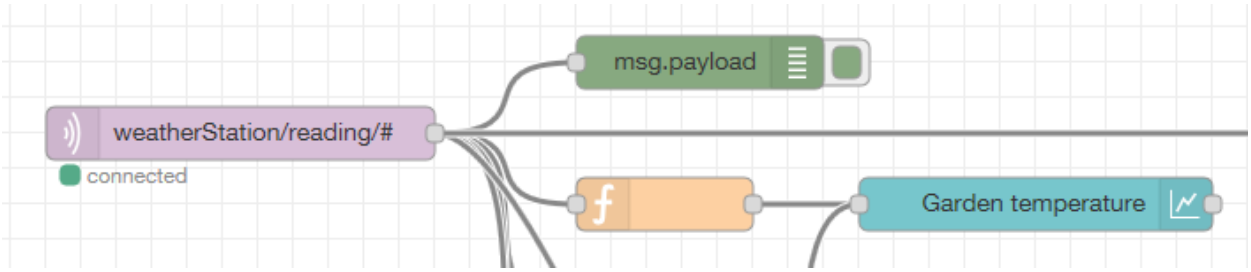
```
on publishReadings do
   publish weatherStation/reading,{"solar_panel":[analog_3#a3],
     "lion_battery":[analog_2#a2],
     "supply_to_wemos":[analog_1#a1],
     "wemos_3v3_supply":[analog_0#a0],
     "temperature":[bme280#temperature],
     "humidity":[bme280#humidity],
     "pressure":[bme280#pressure]}

   if [switch_d5#sw_d5]=0  //Publish to ThingSpeak if the D5 link is intact
      SendToHTTP api.thingspeak.com,80,/update?api_key=<insert key here>
        &field1=[bme280#temperature]
        &field2=[bme280#humidity]
        &field3=[bme280#pressure]

      SendToHTTP api.thingspeak.com,80,/update?api_key=<insert key here>
        &field1=[analog_3#a3]
        &field2=[analog_2#a2]
        &field3=[analog_1#a1]
        &field4=[analog_0#a0]
   endif
endon
```

Note: The string after the 'publish' and 'SendToHTTP' commands should all be on the same line. It has been broken to fit neatly on this page.

Note: The 'publish' string has been formatted as a 'json' string in the Wemos.

Subscribing to MQTT in Node-RED

Here's a screen-shot of the first part of my Node-RED flow that subscribes to 'BeeBotte' (the cloud-based MQTT broker I used for my project).



As you see from the next screen-shot, I've set the 'Topic' to match the channel setting I declared in BeeBotte (i.e. 'weatherStation). If you look back to page-3 of this document you can see the publish command in the Rule-Set.

```
publish weatherStation/reading,{"solar_panel": etc……}
```

I also set the output of the MQTT-In node to be 'a parsed JSON object' so I could extract the transducer values easily in a function node.

If you read the document that describes Stage-5 of my project you will see how I used the transducer values to populate a dashboard within Node-RED.

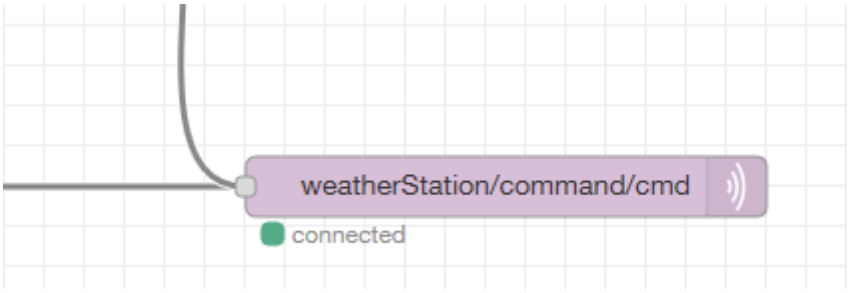
On the next page is a screen-shot of the MQTT-Out node I used to send commands that triggered 'events' within the Wemos D1 Mini.

Bohunt School *(Wokingham)*

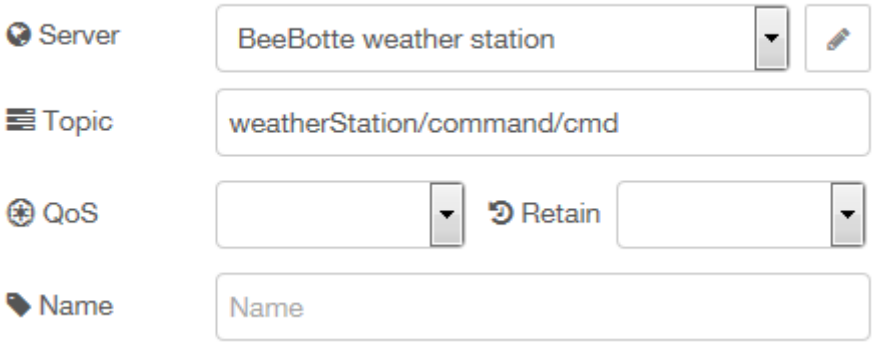
Solar-Powered Weather Station

Stage 4: Measuring and reporting air temperature, humidity and pressure

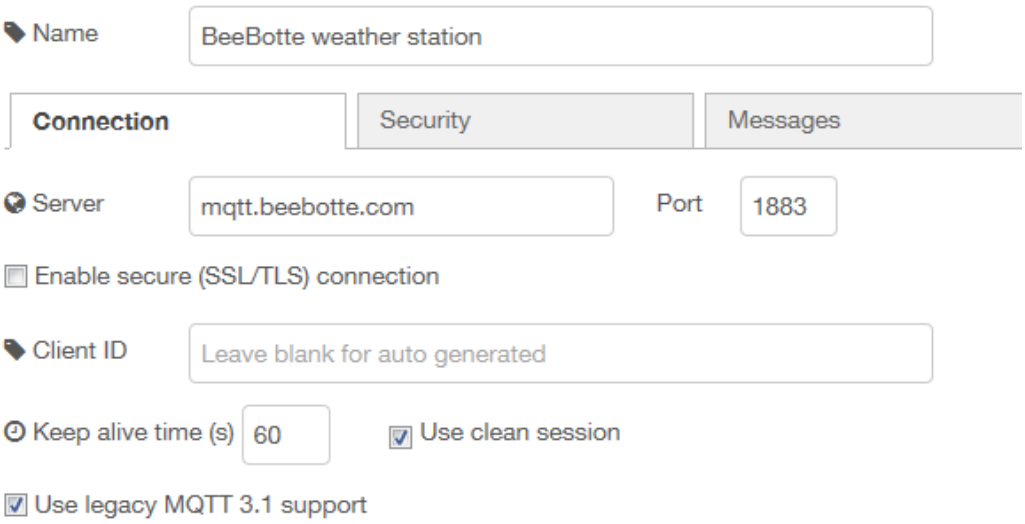
MQTT-Out node



Here's a screen-shot of the node's properties



And here's a screen-shot of the server's settings.



The next thing you need to do is read the write-up for Stage-5.
"Writing a Node-RED flow to process and display the data (which has been extracted from the 'Cloud') remotely and/or locally"

I need to thank Mr D for helping and encouraging me with this project