

## How to setup the MultiSensor

In the following chapters we will go thru the process of: how to flash the MultiSensor unit with Tasmota, how to configure Tasmota so the unit connects to your network, how to configure setting within Tasmota to make sure the sensors and switches send an MQTT signal to Home Assistant, and we will setup Home Assistant so that the sensors and switches show their information properly.

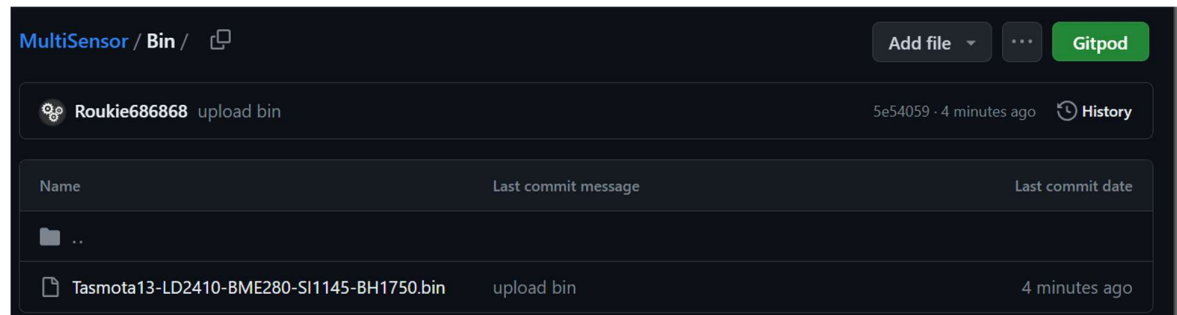
## Table of Contents

How to setup the MultiSensor .....	1
Download the Bin file.....	2
Flashing the ESP8266 .....	2
Configuring the switches in Tasmota .....	6
Home Assistant .....	8
Issues.....	12
Other sensors I2C sensors.....	12
Extra Information .....	12
Building up the PCB.....	13
The SR505 .....	13
The Neo Pixel .....	14
The WEMOS D1 Mini (ESP8266) .....	15
The RCWL-0516.....	16
The I2C boards .....	17

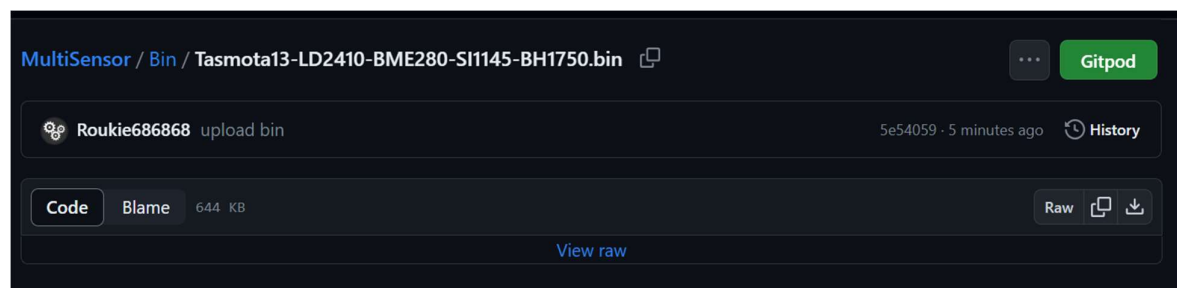
## Download the Bin file

First download the file “**Tasmota13-LD2410-BME280-SI1145-BH1750.bin**” so we can flash the ESP8266 (Wemos D1 Mini). Visit the GitHub page where the BIN file is stored.

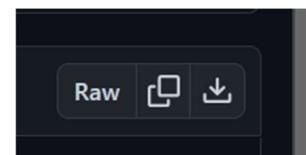
<https://github.com/Roukie686868/MultiSensor/tree/main/Bin>



Click the on filename so that it opens, giving you the following view



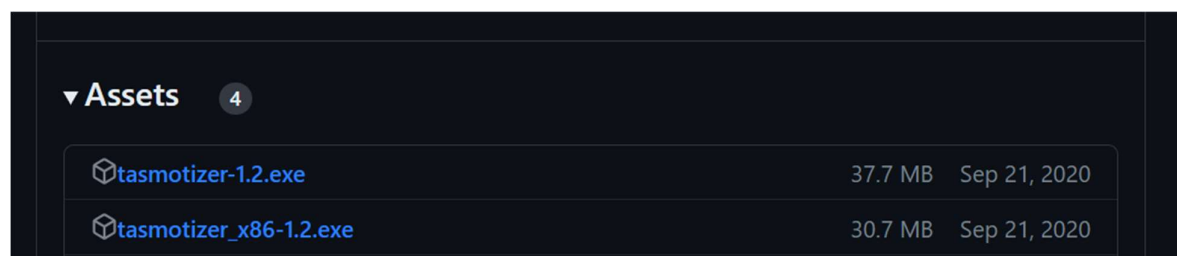
Now click on the download icon for the file (you find this all the way to the right of the screen). This is the one where the arrow points downwards into the tray.



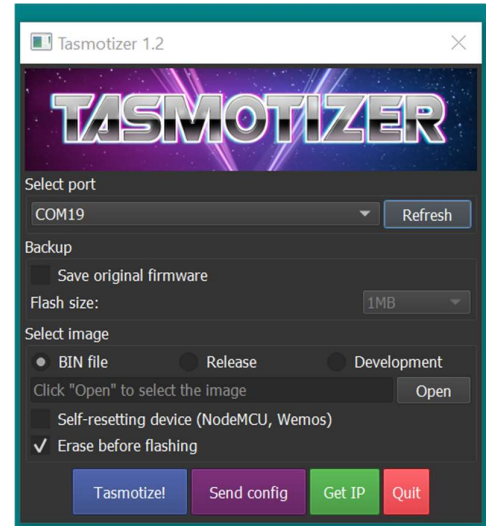
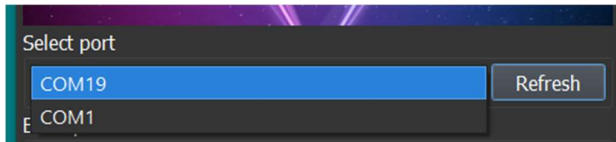
The file should download and you should be able to find it in your download folder under your documents (for Windows machines).

## Flashing the ESP8266

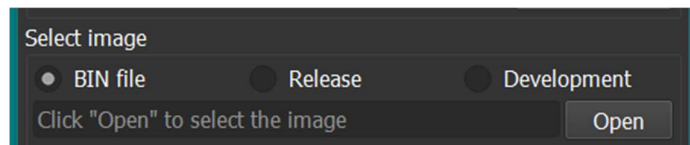
The tool we will use to flash the unit is “Tasmotizer”. It works very well for Tasmota devices as it allows to flash and configure the ESP8266 with custom Tasmota BIN file. Go to <https://github.com/tasmota/tasmotizer/releases> and download “**tasmotizer-1.2.exe**” by clicking on the file. (For older PC with a 32-bit system there is the tasmotizer\_x86-1.2.exe”)



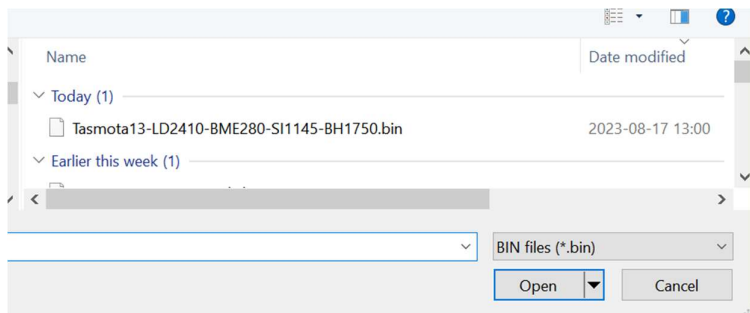
The download should start automatically and the file ends up in your download folder under your documents. From this folder you can double click the file and Tasmotizer will start. First thing to do is figure out which COM: port the ESP8266 is using. Click on the dropdown arrow to see which ports are there. Now connect the ESP8266, “Refresh” the ports, and see what appeared new. Now select that port. (When you have multiple EPS8266 units connected to your PC, you don’t want to make the mistake of wiping/flashng the wrong unit.)



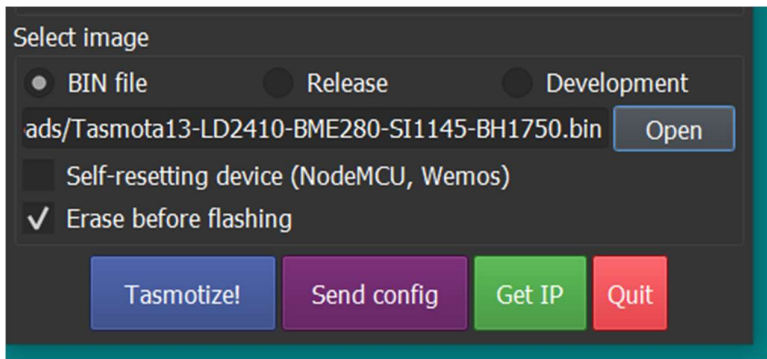
Now that we have the right COM: port we need to select the BIN file that we downloaded earlier from the GitHub page. Make sure the “Select image” is set for “BIN file”. Now click on the “Open” button to select the BIN file that you downloaded.



Find your file in the download folder and click “Open”



The Tasmota screen should look like the next screenshot

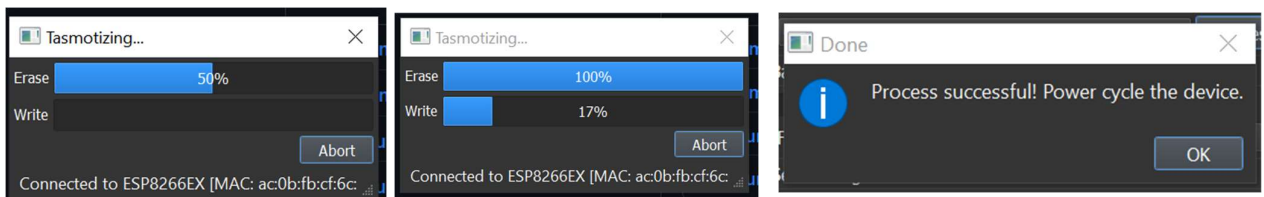


Verify that the “Erase before flashing” is marked. (This to make sure older settings are wiped from the ESP8266)

Now click the blue “Tasmotize!” button to flash the ESP8266.

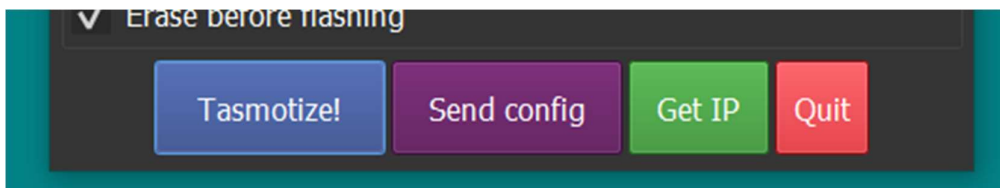
Tasmotizer will first Erase the content of the ESP8266.

Then it will flash the BIN file to the EPS8266.



When the process is successful the last popup will appear telling all worked well. Now Click “OK”. (With the WEMOS ESP8266 there is typically no need to power cycle.)

Now it is time to setup the Network and MQTT configurations. Click the “Send config” button.



The following configuration menu shows up.

Enable the “WiFi” and “MQTT” boxes so that we can fill in the needed information.

#### WiFi

- Type your Network name and password in the two fields under WiFi.
- As the password will not be visible it makes sense to first type it in Notepad and then copy and paste it over to Tasmotizer. This to make sure that are no Typos.

#### MQTT

- HOST - Point to your MQTT server by listing the IP address of the MQTT server
- Port – Standard is 1883 but if you have chosen a different number list it here
- Topic – Give a name you like or start with “tasmota” as we can change that later
- FullTopic – Make sure to not delete anything out of this line. (The prefix is use by Tasmota for cmdnd, tele and stat to communicate with Home Assistant)
- FriendlyName - Typically this is the same as the Topic to keep things easy (Again we can all change this later within the Tasmota Webpage)
- User – When you use credentials for the MQTT server than list the username here
- Password - When you use credentials for the MQTT server than list the password here (maybe here as well type it first in a Notepad and copy it over)

Enable the Module/template box

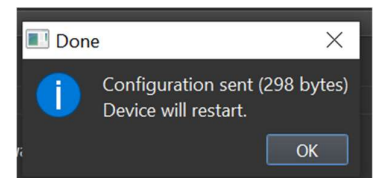
- Select “Template” and copy the line below into the textbox

```
{"NAME":"MultiSensor","GPIO":[1,9248,1,9280,640,608,0,0,161,1376,160,1,1,1],"FLAG":0,"BASE":18}
```

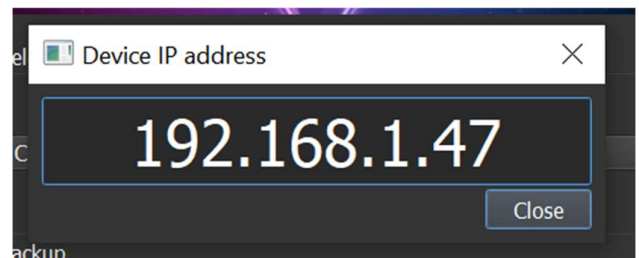
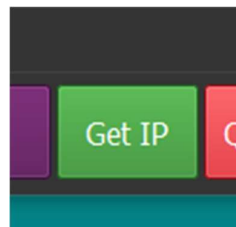
This will setup Tasmota with the correct inputs we need for all the sensors.

An example of that below.

Now click save the save button. Tasmotizer will now write your information to the Tasmota software on the ESP8266. Depending the amount of data you have listed the number of bytes will change.



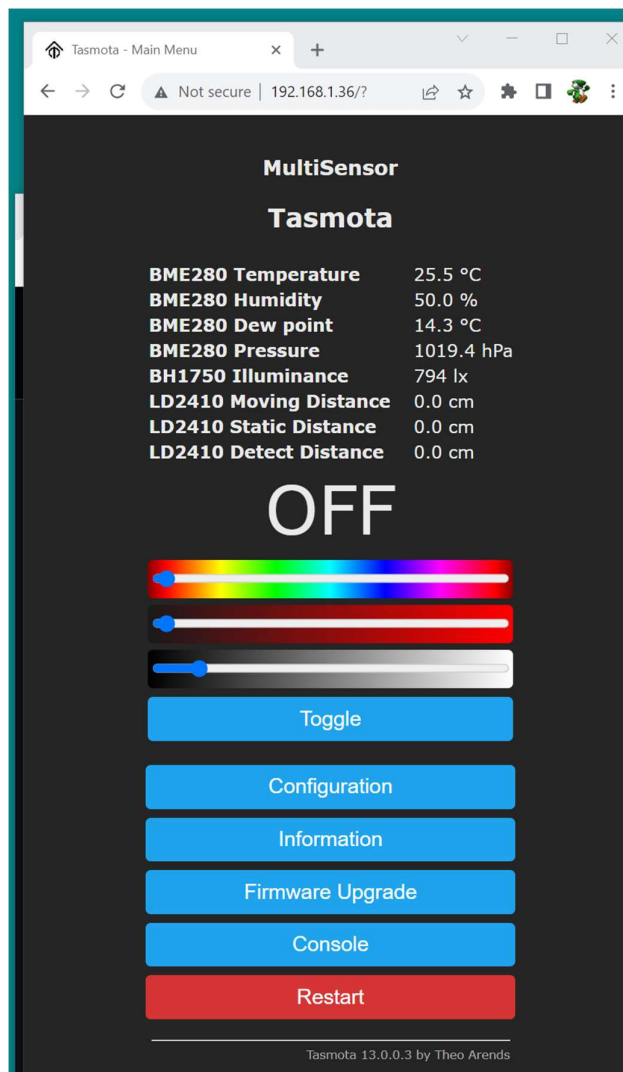
Wait about 30 second and click on the Get IP" button. When all went well there should be an IP address listed that we can visit in a browser.



If the IP address shows as xxx.xxx.xxx.xxx then head over to **"Home Assistant"** section to get to the unit. We will come back at this point to configuring the switches in Tasmota.

### Configuring the switches in Tasmota

Open up a web browser and go to the IP address that was listed in your popup.



When all was installed properly a similar view as above is what you should see. Depending what sensor units that are installed on the MultiSensor board different measurements will show up. As the LD2410 uses a serial protocol it will always show on the screen even when it is not there. As Tasmota does not transmit the data for the HLK-LD2410 when it is not there, we will not see this info in HA. The HLK-LD2410 and the RCWL-0516 share the same space on the PCB. When the RCWL-0516 is removed the more expensive HLK-LD2410 can be placed later.

Now let's click on the "Console" button. The console screen will show you what is happening within the unit but it also allows for making modifications to the Tasmota settings. On the command bar enter 5 different commands printed below in **BOLD**:

**SwitchMode1 15** // This command lets switch 1 (The PIR) send MQTT messages

**SwitchMode2 15** // This command lets switch 2 (The Microwave) send MQTT messages

**SetOption114 1** // This command decouples the switches from relays

**SwitchText1 mmWave** // This tells Home Assistant the name of the switch1

**SwitchText2 PIR** // This tells Home Assistant the name of the switch2

If you want to read up on these Tasmota settings they can be found on the following Tasmota webpages:

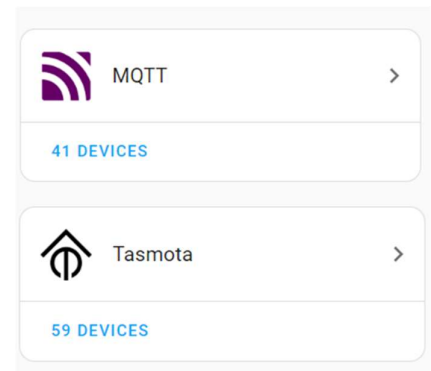
<https://tasmota.github.io/docs/Buttons-and-Switches/#setoption114>

<https://tasmota.github.io/docs/Buttons-and-Switches/#switchmode-15>

## Home Assistant

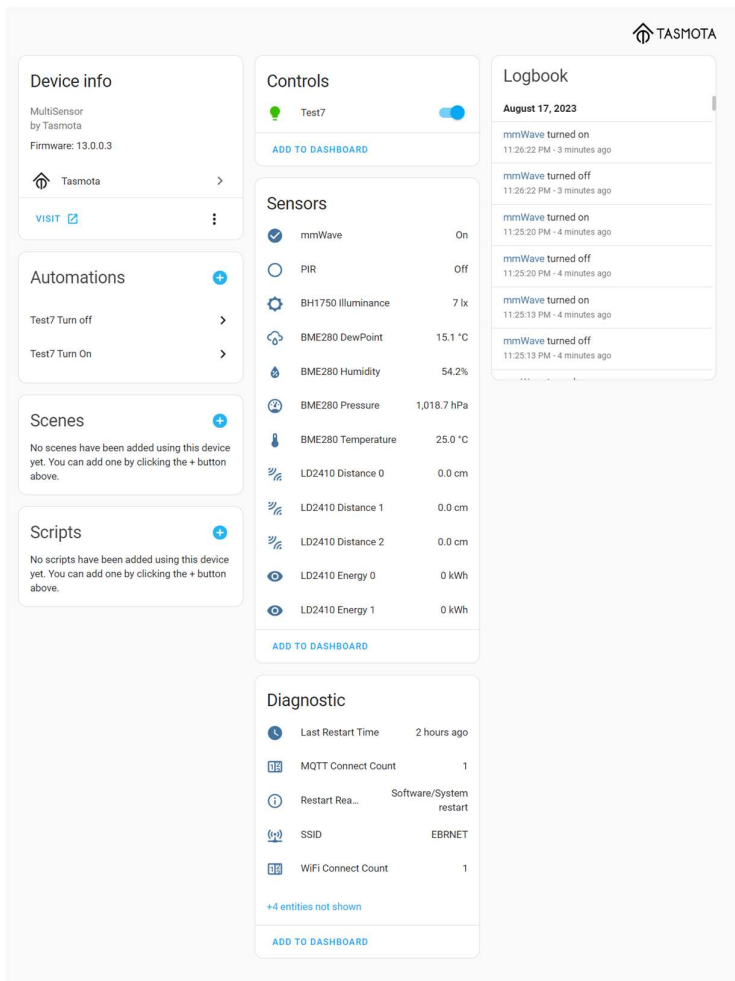
Both **MQTT** and **Tasmota** have to be install within Home Assistant. If not do so first. When installed the Device can be found by:

- Click on "Settings" (left bottom corner of the screen)
- Click on "Devices & Services"
- Select the blue "DEVICES" under the Tasmota Icon
- In the list, the device should show up as "**Tasmota**" (or any other name that you use in "Tasmotizer" when sending over the configuration file.
- Click on "Tasmota"

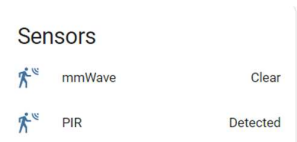


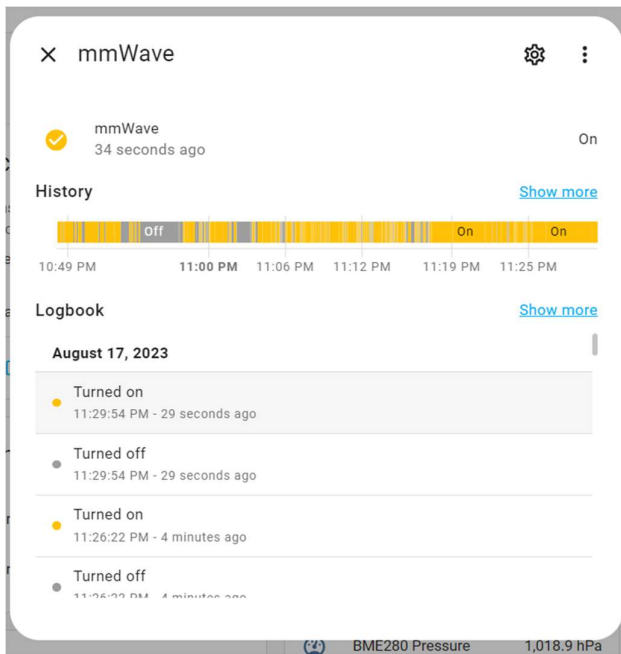
Now a device screen shows up as seen in the next picture. If you had issues with the IP address of the device, click on the blue "Visit" icon in the "Device Info" section to go to the device and make the settings you were not able to do in the previous chapter. When the Tasmota settings are done come back at this point and continue.





Now let's change the Icon and the text that the sensors display to something that is more meaningful as seen on the right image. Click on the "mmWave" text so that the popup shows as seen in the next image.

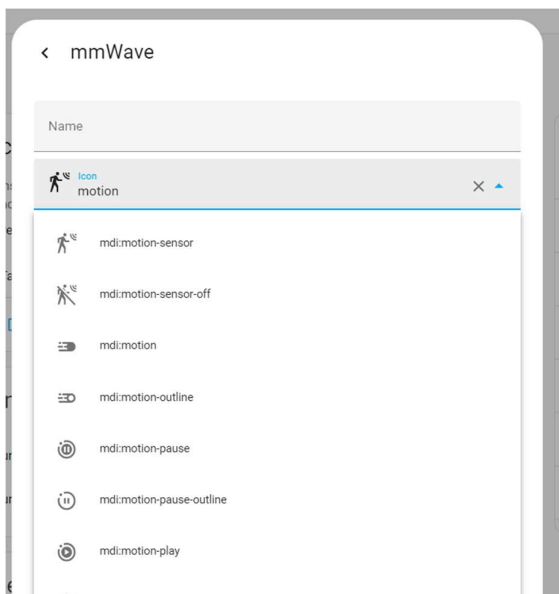




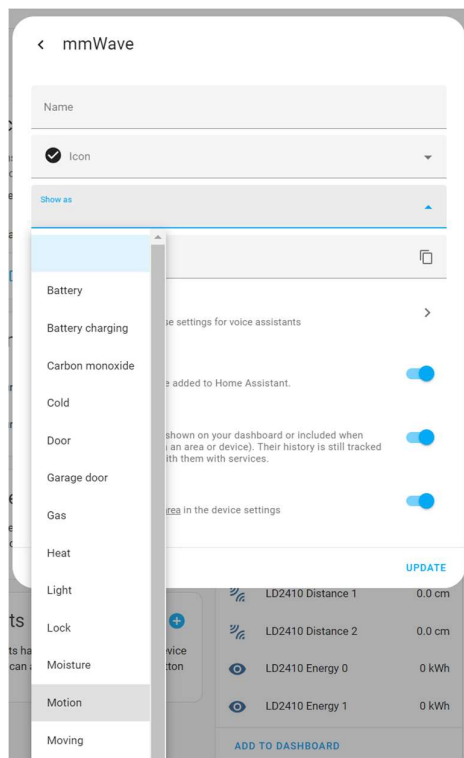
Now click on the little star wheel in the top. In the next popup we are going to change the “Icon” and the “Show as” field.

Click right on “Icon” and type the word “motion” in the Icon field (typing “Presence” could be another option) and pick the symbol that you like for this sensor.

Name	
<input checked="" type="radio"/> Icon	▼
Show as ▼	

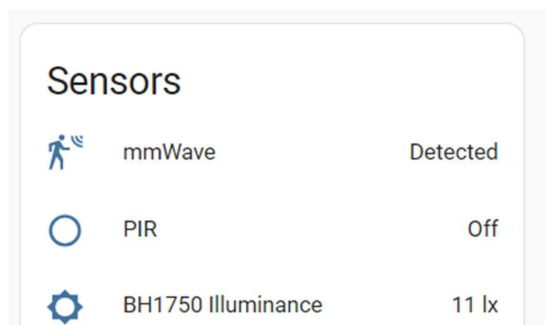


Now click on “Show as” and type motion again (typing “Presence” could be another option)



In the right bottom of the popup click on “UPDATE” to make sure your settings are saved.

Below you can see what your changes have accomplished. Repeat the same for the PIR sensor.



Congratulations. You are ready to further integrate your sensor into you Home Assistance system.

## Issues

- When the switch does not show a reading but says “**unknown**”
  - o Go back to the Tasmota console and change the **SwitchText1** or **SwitchText2** for a low case text. Follow this by a **SetOption19 1** to turn the device off in Home Assistant and turn in back on again with **SetOption19 0**. This should reset HA.

## Other sensors I2C sensors

When you add new [I2C](#) sensors to the MultiSensor PCB that fits on one of the four female 4-pin headers but they do not show up on the Tasmota screen then the BIN file will have to be upgraded for that sensor. Just send me a request with that sensor and I will make you the new BIN file. I will post that on the GitHub page. One restriction is that the Sensor must be known by Tasmota. Please verify that Tasmota can read your sensor by going to the following Tasmota page:

<https://tasmota.github.io/docs/I2CDEVICES/>

## Extra Information

More details on the project can be found on the following GitHub Page

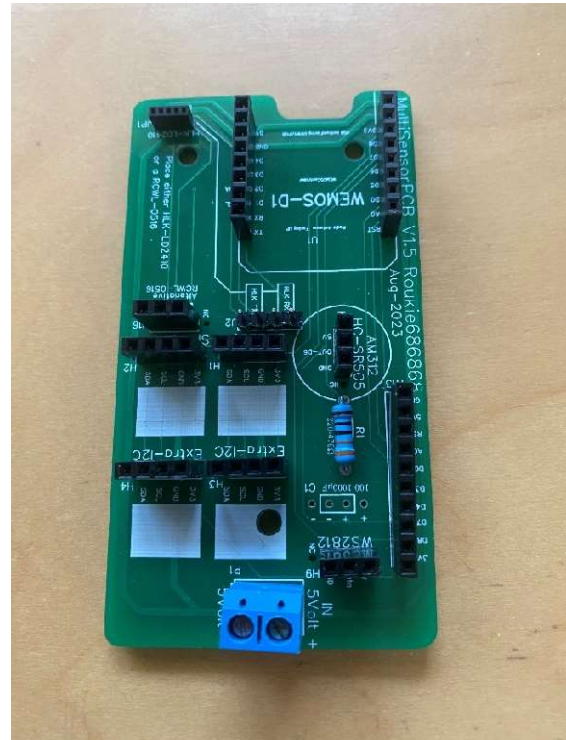
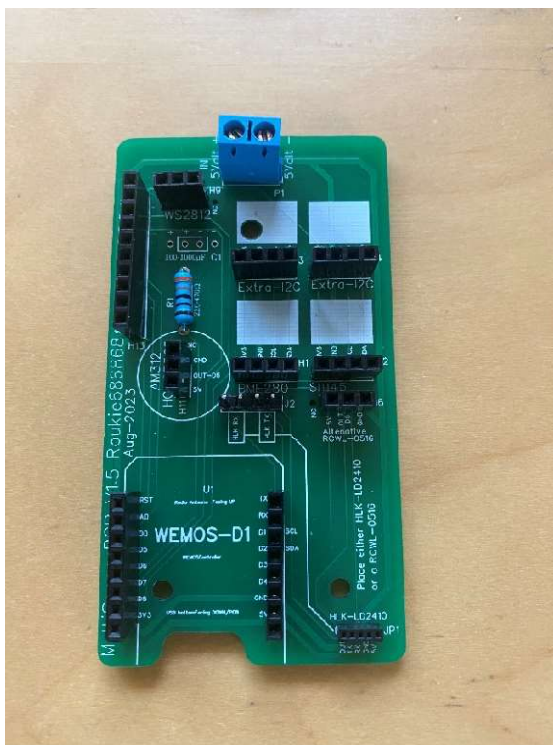
<https://github.com/Roukie686868/MultiSensor>

## Building up the PCB

To fully equip the board, you will need to solder the following parts on the PCB:

- 4 headers (4-pin female) for I2C devices (BME280, GY1145 and 2 spare slots)
- 1 header (5-pin female 1.27 mm spacing) for the 24GHz sensor (HLK-LD2410)
- 1 header (3-pin female) for a microwave motion sensor (RCWL-0516)
  - o Note: The HLK-LD2410 & the RCWL-0516 share the same space and only one can be placed
- 1 header (3-pin female) for the PIR sensor (HC-SR505 or AM312)
- 1 header (10-pin female) to expose the residual ports A0, D0, D3, D4, D7 & D8 on the Wemos plus 5V, 3V3 and GND
- 2-pin male for 2 jumpers to turn on or off the TX & RX signal from the 24GHz sensor
- 1 optional capacitor [100-1000µF]
- 1 Header (3-pin female) for the addition of a WS2812 Neo-Pixel strip to locally indicate sensor status
- 1 resistor [220-470Ω] to protect the Neo-Pixel input signal

When all is in place it should look like the next two images.



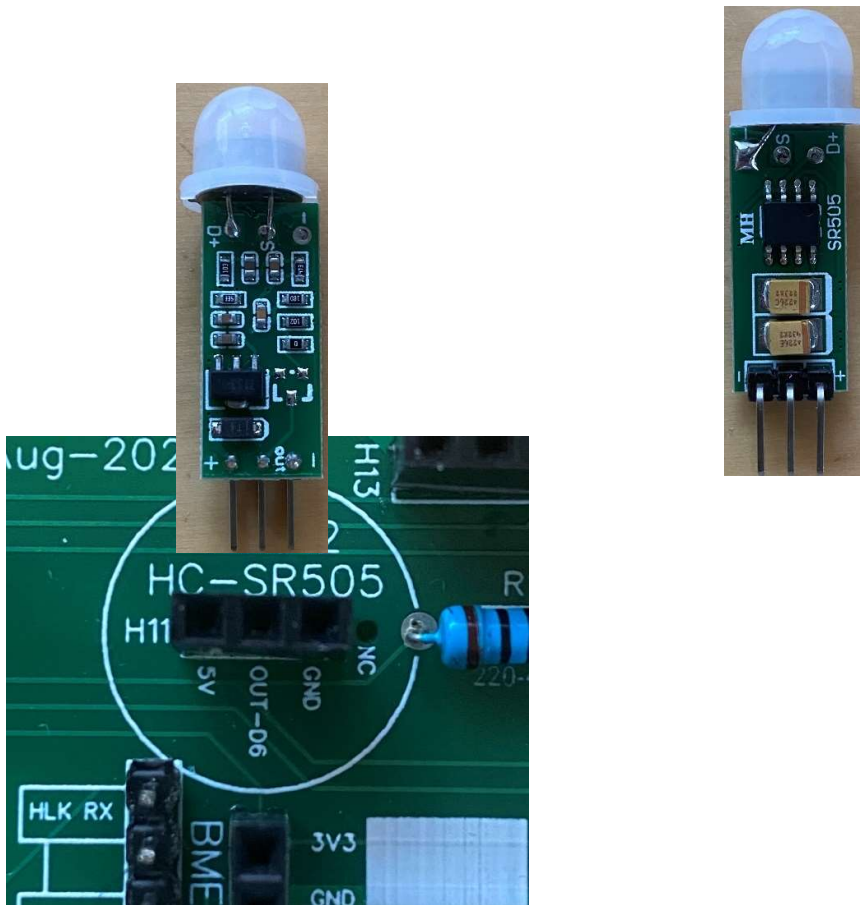
Next we need to place the individual breakout board and make sure they are place properly. As each port on each header is labeled this should not be too much of a problem. But let's review each item just to make sure:

### The SR505

Connect the 3 pins on the SR505 to the 3-pin female header on the PCB as shown in the next picture.

- |                            |    |                         |
|----------------------------|----|-------------------------|
| - <b>5V</b> on the PCB     | to | <b>+</b> on the SR505   |
| - <b>OUT-D6</b> on the PCB | to | <b>out</b> on the SR505 |
| - <b>GND</b> on the PCB    | to | <b>-</b> on the SR505   |

The labeling on the SR505 is small so take care to place it the right way around



### The Neo Pixel

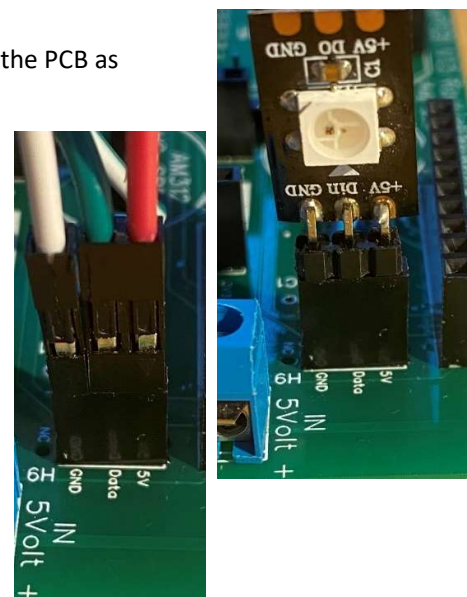
Connect the 3 pins on the Neo Pixel to the 3-pin female header on the PCB as shown in the next pictures.

For the test Neo Pixel

- |                          |    |                              |
|--------------------------|----|------------------------------|
| - <b>5V</b> on the PCB   | to | <b>+5V</b> on the Neo Pixel  |
| - <b>Data</b> on the PCB | to | <b>Din</b> on the Neo Pixel  |
| - <b>GND</b> on the PCB  | to | <b>-GND</b> on the Neo Pixel |

For typical Neo Pixel wiring use the following

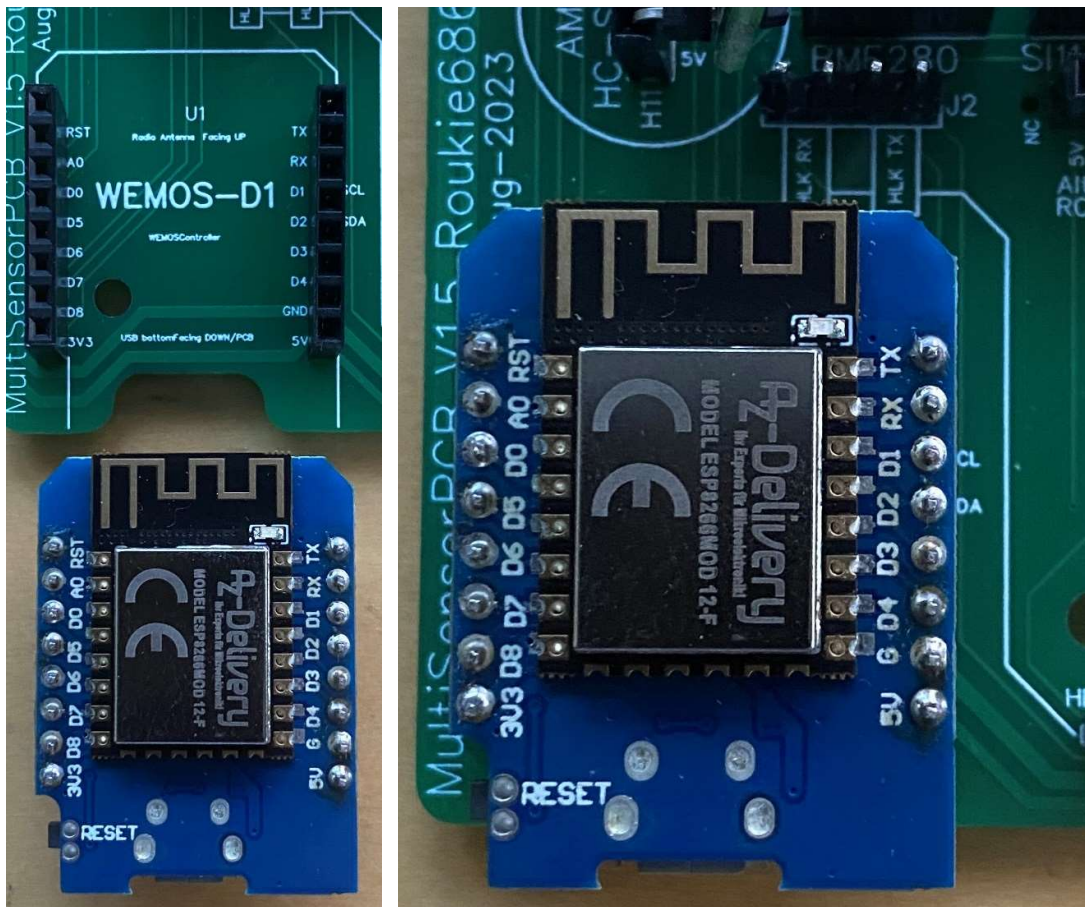
- |                          |    |                               |
|--------------------------|----|-------------------------------|
| - <b>5V</b> on the PCB   | to | <b>Red</b> on the Neo Pixel   |
| - <b>Data</b> on the PCB | to | <b>Green</b> on the Neo Pixel |
| - <b>GND</b> on the PCB  | to | <b>White</b> on the Neo Pixel |





## The WEMOS D1 Mini (ESP8266)

Make sure to solder the 8-pin male header such that the antenna faces upwards and place the WEMOS with the USB port (on the bottom) facing the outside of the PCB like seen in the picture.

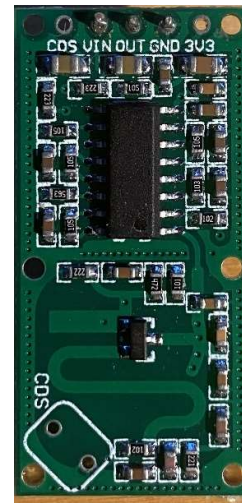
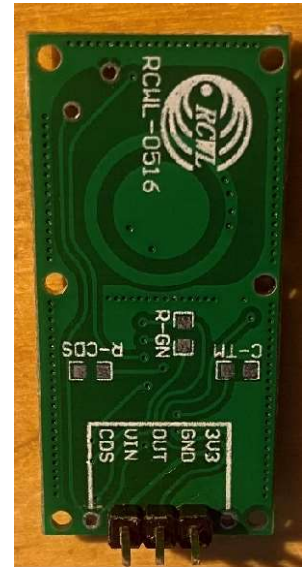


## The RCWL-0516

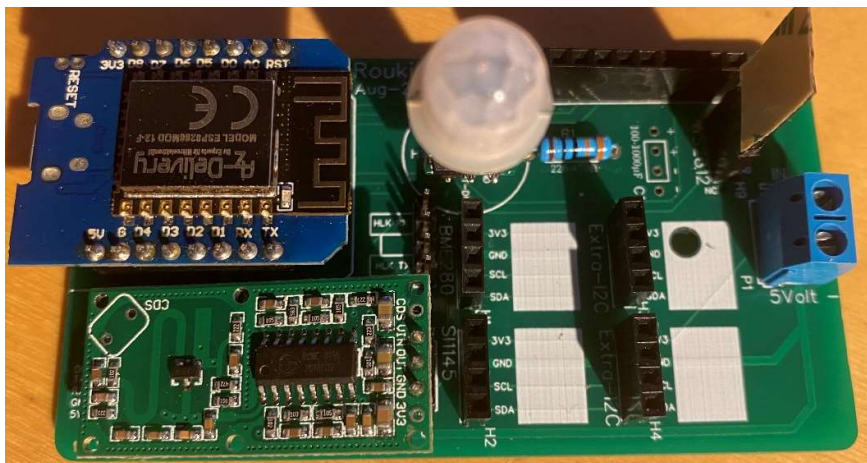
The RCWL-0516 has 5 pins of which we will only use the middle 3. Make sure to solder the 3 pins in the middle part as the **CDS** and **3V3** are not needed.

Connect the 3 pins on the RCWL-0516 to the 3-pin female header on the PCB as shown in the next picture.

- |                            |    |                             |
|----------------------------|----|-----------------------------|
| - <b>5V</b> on the PCB     | to | <b>VIN</b> on the RCWL-0516 |
| - <b>OUT-D5</b> on the PCB | to | <b>OUT</b> on the RCWL-0516 |
| - <b>GND</b> on the PCB    | to | <b>GND</b> on the RCWL-0516 |



Result so far should look like this:





## The I2C boards

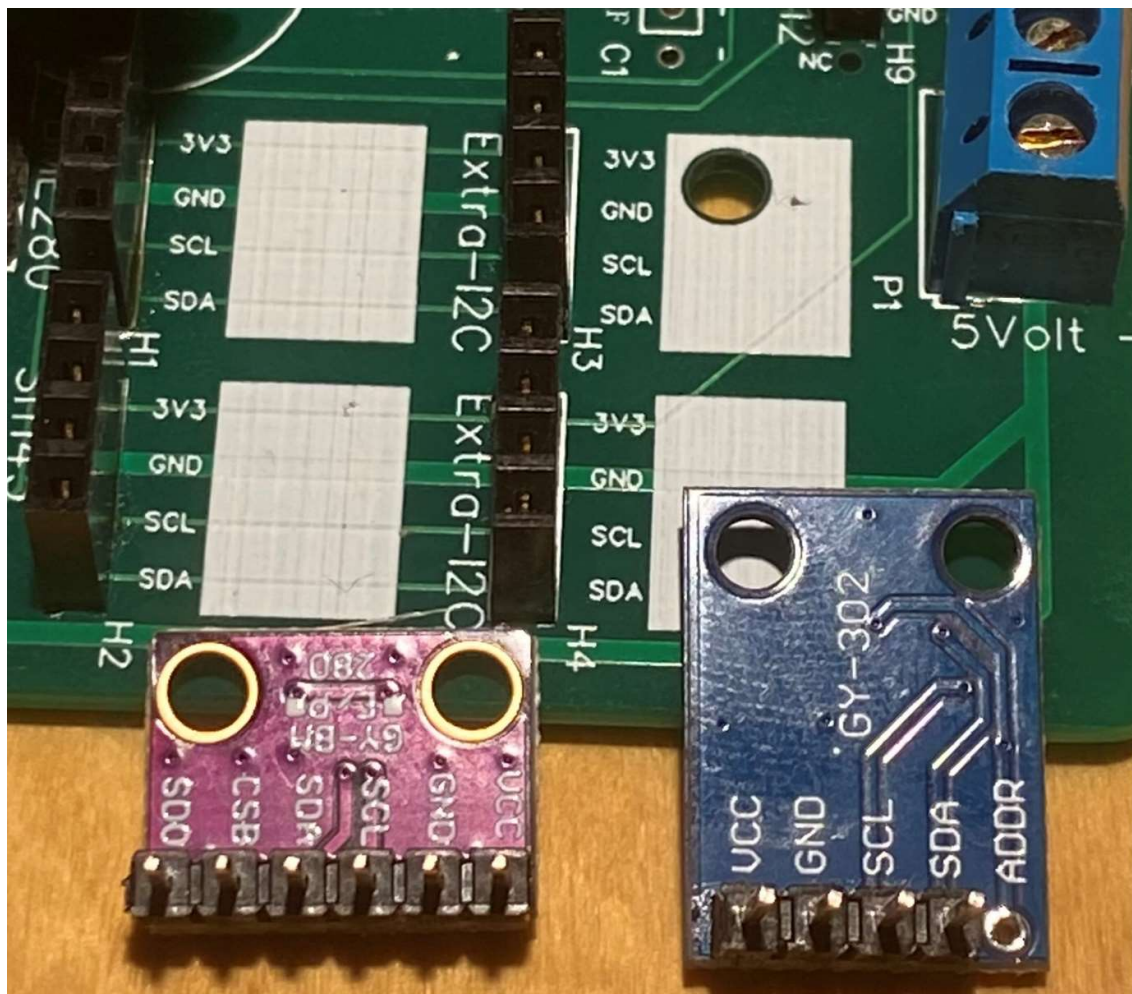
The I2C boards all look a bit different and they will fit the PCB when they have the following pin order

3V3 – GND – SCL – SDA. It was the intent to have them all on the right of the 4-pin header but some I2C board have their sensor such that you will have to rotate the board over the position of another header. As there are 4 spots available there should be no issue to place at least 2 I2C boards.

- On the 6-pin BME280 board the CSB and SDO are not needed.
- On the 4-pin BME280 board there are pins are used (did not have a picture as there were on order)
- On the 5 GY-302 or BH1750 board the ADDR pin is not needed

Connect the 4 pins on the RCWL-0516 to the 3-pin female header on the PCB as shown in the next picture.

- |                         |    |                                  |
|-------------------------|----|----------------------------------|
| - <b>3V3</b> on the PCB | to | <b>VCC, VIN</b> on the I2C board |
| - <b>GND</b> on the PCB | to | <b>GND</b> on the I2C board      |
| - <b>SCL</b> on the PCB | to | <b>SCL</b> on the I2C board      |
| - <b>SDA</b> on the PCB | to | <b>SDA</b> on the I2C board      |



Two examples how to fill the headers

