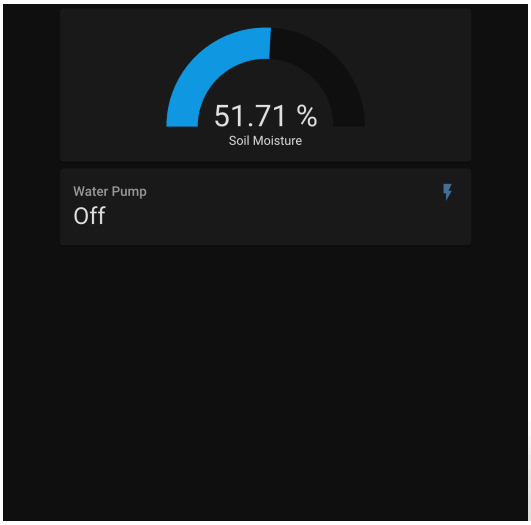
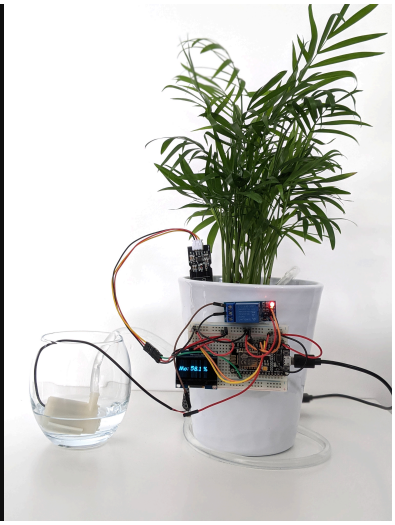
Home Assistant

Automation and Smart Plant

Project description:

In this project, you will further explore **Home Assistant** and learn the power of **Automation**. You will create a **moisture station** to automatically water your favourite plant at home when the moisture level of the plant soil drops below a certain arbitrary value. We all know how tedious it is to provide an ideal growing environment for your plants. To be honest, even remembering to water them regularly is a challenge. Let's tackle how Home Assistant Automation can assist you in creating a self watering plant.





Project objectives:

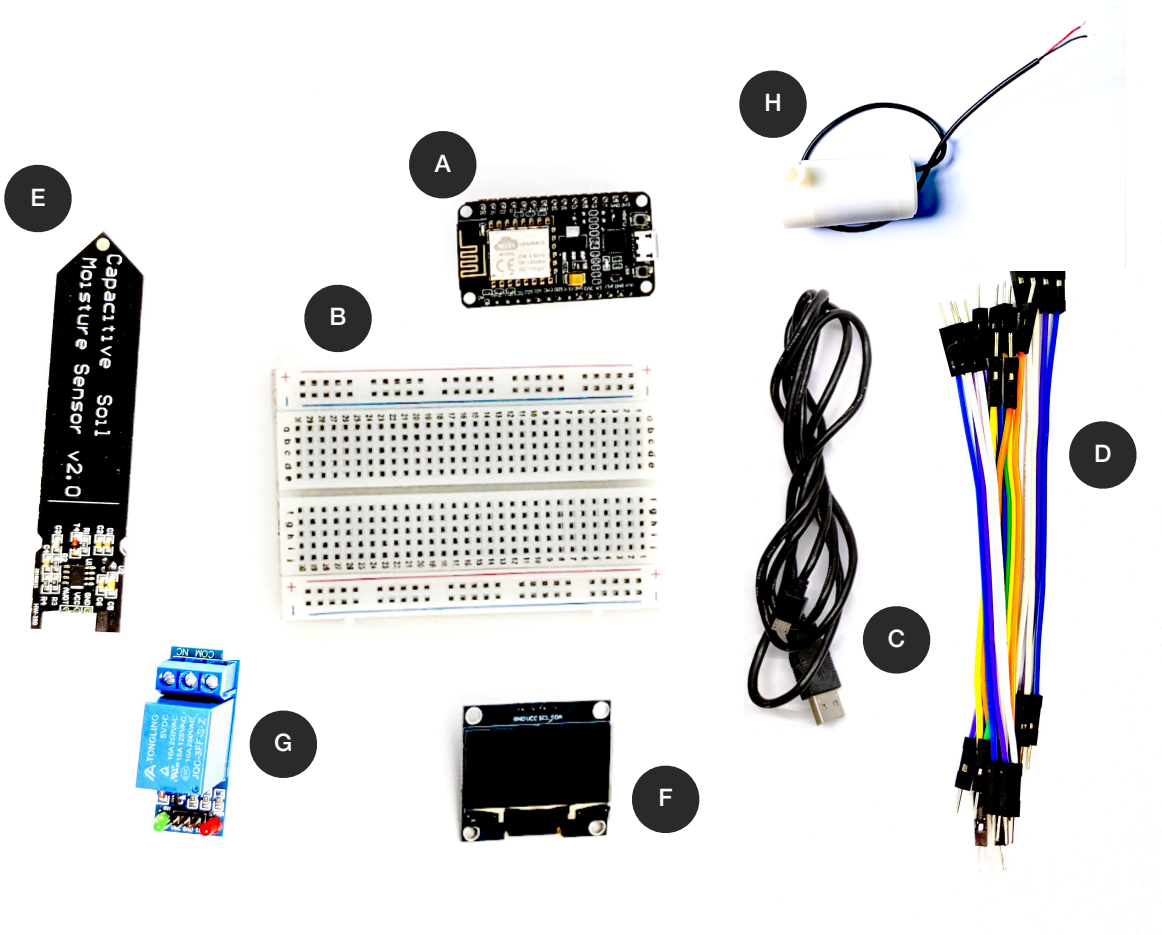
- Create a circuit to monitor the moisture level of your plant
- Push live temperature data to Home Assistant
- Monitor your data through the Home Assistant interface
- Use automation to trigger behaviours

Project components:

Please note: This exercise is an extension task and some of the components are not part of the kit that we suggested for this module. You can either buy the missing components or simulate the automation task with alternative components:

- Swap the Soil Moisture Sensor with the DHT11 Humidity Sensor
- Swap the 3V Water Pump with an LED

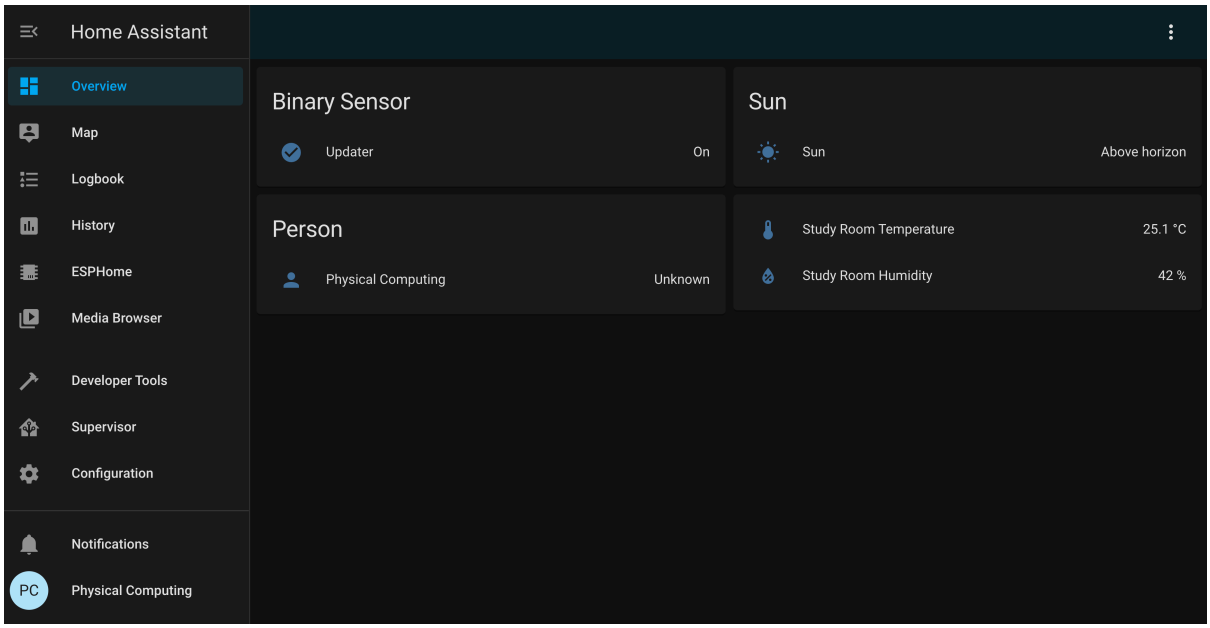
Component Reference	Component Quantity	Component Name	Component Description
A	1	ESP8266 WiFi Module	A low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability
B	1	Breadboard	A rectangular plastic board with conductive rails for fast circuit prototyping
C	1	Micro USB Cable	A USB cable to power and upload instructions to a microcontroller
D	15	Jumper Wires	Conductive cables frequently used with a breadboard to connect two points in a circuit
E	1	Soil Moisture Sensor	An electronic component which uses capacitance to measure the water content of soil and converts the input data into electronic data
F	1	1.3” OLED IIC LCD Display	A low power consumption LCD display for visual data output
G	1	1 Channel 5V Relay Module	An electric switch that use electromagnetism to convert small electrical stimuli into larger currents
H	1	3V Submersible Water Pump	A mini component which allows to move water from one area to another



You should now be familiar with how to run and access Home Assistant. If not, you can refer back to the previous “Temperature and Humidity Sensor” Home Assistant exercise for instructions.

Go ahead and start your Home Assistant virtual machine.

Accessing **http://homeassistant.local:8123/** on your browser will direct you to the Home Assistant dashboard where you should still have the temperature and humidity data of your study room from the previous exercise:



Step Two
Building the Moisture Station Circuit

The circuit assembly is very similar to the temperature station circuit that you built on the previous Home Assistant exercise. The main difference is that you will swap the temperature/humidity sensor with the moisture sensor and add both the relay module and water pump to the circuit.

The idea is to use the LCD display to show your plant soil moisture level. When the moisture level drops below 30%, the water pump will be activated and will hydrate your plant until its moisture level exceeds 70%.

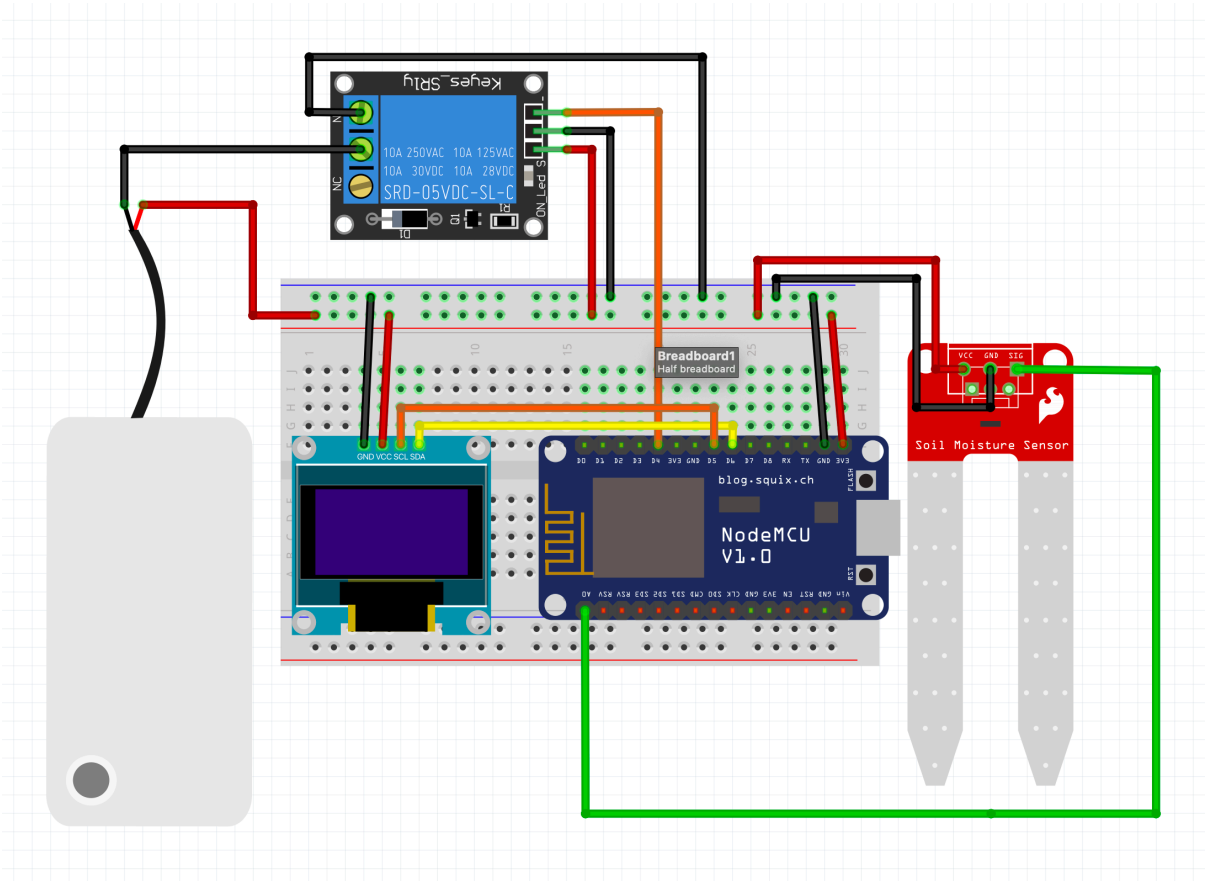
The moisture sensor: this sensor has three pins. It requires a power and a ground connection, as well as a third connection to read the moisture data. The latter refers to the AUTO pin of the moisture sensor which you will connect to your ESP8286 board **A0** pin.

The LCD display: this sensor has four pins. It requires a power and a ground connection, as well as two additional connections to display the temperature data. These additional connections are **SCL** (clock line) and **SDA** (data line) which are used to synchronise the data transfer. You will connect them respectively to your ESP8286 board **D5** and **D6** pins.

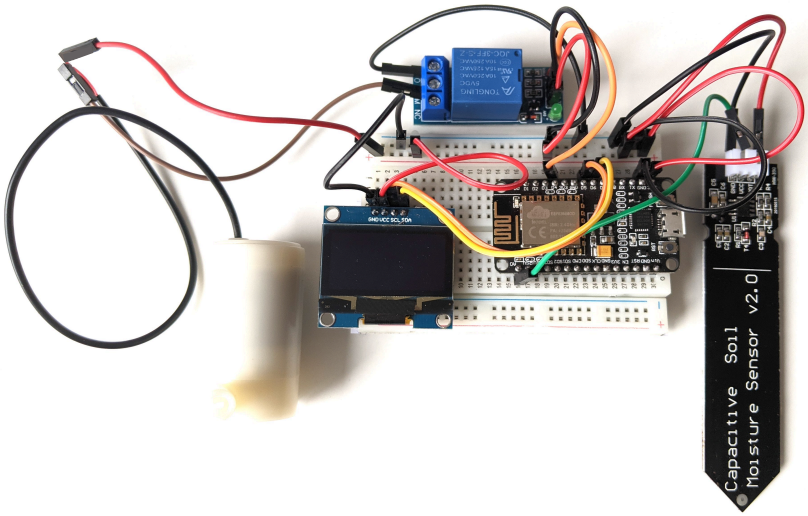
The relay module: this component is not really required for this circuit as the water pump can operate with the 3V supplied by your ESP8286 board. This component has three pins. It requires a power and a ground connection, as well as a third connection to activate the relay. Once activated, the relay will allow current to flow between the two output connections. It is very useful as you can trigger the relay with minimal current (like the one provided by the ESP8286), and control higher voltage-current components. The water pump can easily run with the electricity provided by the ESP8286 board and a relay is not really required but you will integrate it to learn its mechanism. You will connect the activation pin (IN) to your ESP8286 board **D4** pin.

The water pump: this component is very simple. It only requires a power and a ground connection. You will connect the water pump to the ground and power pins of your ESP8286 board and use the relay to act as an activator to switch the water pump on and off. You might need crocodile cables to properly connect the water pump to the circuit as the water pump wires are very tin.

It is now your turn to assembly the circuit.
The diagram below shows you how the circuit should be assembled:



Furthermore, see below a picture of the circuit assembled:

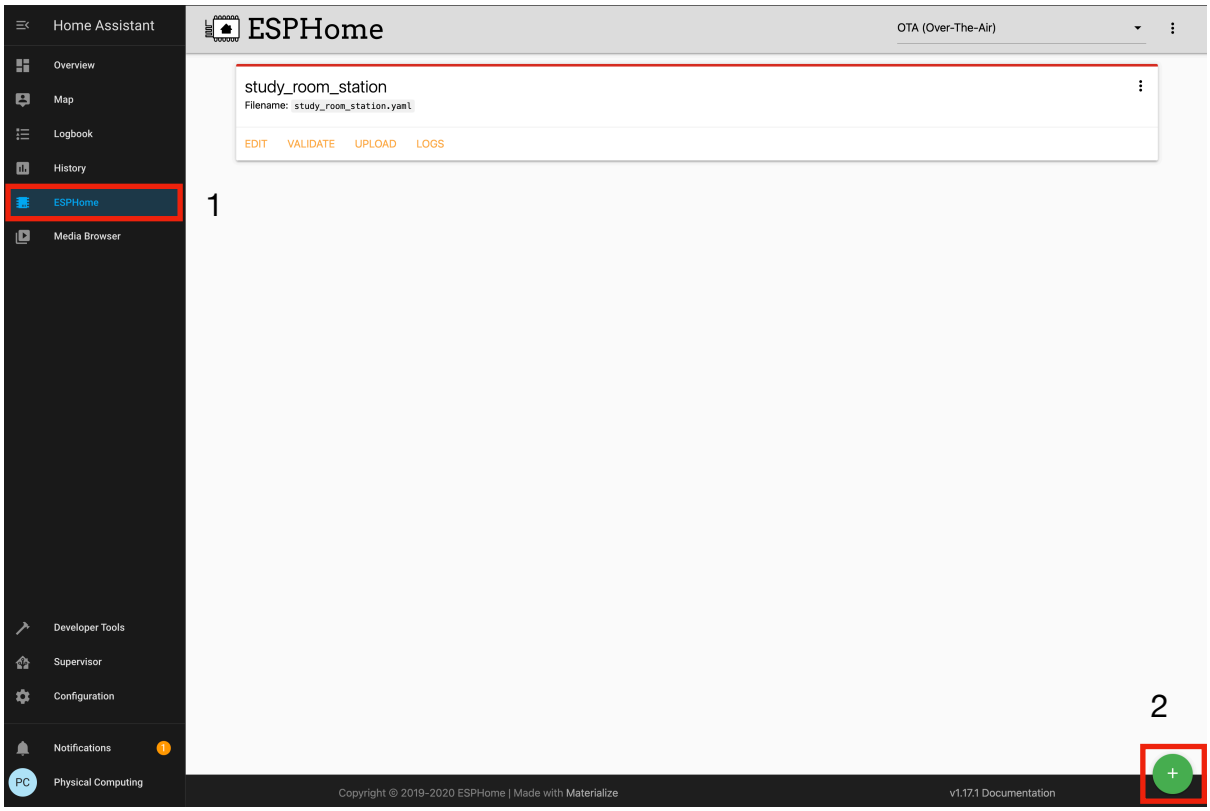


At this point, you should have your moisture station circuit assembled. This means that your ESP8266 board is connected to the LCD display component, the moisture sensor component, the relay component, and the water pump.

Next up is to configure Home Assistant with a new node so that you can program the circuit to read both the moisture of your plant soil, display the moisture value on the LCD screen, and push the data via WiFi to Home Assistant .

Step Three
The “Smart Plant” ESPHome Node

With Home Assistant running, head to the “ESPHome” section (1), click on the plus button (2), and create a new node called “smart_plant”.



You can also edit the existing “study_room_station” node if you do not want to create a new node. Please refer back to the previous “Temperature and Humidity Sensor” Home Assistant exercise if you need a refresher on how to setup a ESPHome node.

Once added the new “smart_plant” node, the configuration file should look like the following:

Editing: `Smart_Plant.yaml`

```
1  esphome:
2    name: smart_plant
3    platform: ESP8266
4    board: esp01_1m
5
6  wifi:
7    ssid: "[redacted]"
8    password: [redacted]
9
10   # Enable fallback hotspot (captive portal) in case wifi connection fails
11   ap:
12     ssid: "Smart Plant Fallback Hotspot"
13     password: [redacted]
14
15 captive_portal:
16
17 # Enable logging
18 logger:
19
20 # Enable Home Assistant API
21 api:
22   password: [redacted]
23
24 ota:
25   password: [redacted]
```

Your node and board information

Your wifi information

Fall back information if no wifi

Logger enabled for debugging

Your Home Assistant and Over the Air passwords

Click on the **Upload** button located at the bottom of the YAML file window to upload the firmware to your board. You can view the logs and verify that the upload was successful and that the board is connected to your wifi network:

Compiling & Uploading: `Smart_Plant.yaml`

```
[10:15:02][D][wifi:324]: Starting scan...
[10:15:08][D][wifi:339]: Found networks:
[10:15:08][I][wifi:385]: - '9BAC Hyperoptic Fibre Broadband' (DC:D9:AE:B1:85:89)
[10:15:08][D][wifi:386]:     Channel: 11
[10:15:08][D][wifi:387]:     RSSI: -42 dB
[10:15:08][D][wifi:389]: - ''  (6A:D9:AE:B1:85:8C)
[10:15:08][D][wifi:389]: - 'Alderaan 2.4Ghz'  (C8:5A:9F:D2:54:9C)
[10:15:08][D][wifi:389]: - '90DB Hyperoptic 1Gb Fibre 2.4Ghz'  (7C:39:53:F5:90:DB)
[10:15:08][D][wifi:389]: - ''  (FA:8F:CA:3A:BC:CE)
[10:15:08][D][wifi:389]: - ''  (FA:8F:CA:9E:63:7A)
[10:15:08][D][wifi:389]: - 'SKYD5AFF'  (A0:BD:CD:07:35:22)
[10:15:08][D][wifi:389]: - 'SKYD5AFF'  (38:A6:CE:9A:1D:38)
[10:15:08][I][wifi:194]: WiFi Connecting to '9BAC Hyperoptic Fibre Broadband'...
[10:15:10][I][wifi:457]: WiFi Connected!
[10:15:10][C][wifi:303]:   SSID: '9BAC Hyperoptic Fibre Broadband'
[10:15:10][C][wifi:304]:   IP Address: 192.168.1.109
[10:15:10][C][wifi:306]:   BSSID: DC:D9:AE:B1:85:89
[10:15:10][C][wifi:307]:   Hostname: 'smart_plant'
[10:15:10][C][wifi:311]:   Signal strength: -44 dB
[10:15:10][C][wifi:315]:   Channel: 11
[10:15:10][C][wifi:316]:   Subnet: 255.255.255.0
[10:15:10][C][wifi:317]:   Gateway: 192.168.1.1
[10:15:10][C][wifi:318]:   DNS1: 192.168.1.1
[10:15:10][C][wifi:319]:   DNS2: (IP unset)
```

Now that the connection is established, it is time to write the code to read the moisture data, display the moisture value on the LCD screen, and control the water pump activation relay to turn the pump on and off manually.

Step Three
The Moisture Station Firmware

It is now the time to modify the “Smart_Plant.yaml” file with additional code. Open the “Smart_Plant.yaml” and add the following snippets of code at the end of the document (make sure the code is indented properly):

The moisture sensor:

```
# Configuration entry for the moisture sensor
sensor:
platform: adc
pin: A0
id: moisture_sensor
name: "Soil Moisture"
unit_of_measurement: "%"
update_interval: 1s
filters:
# Sensor dry at around 0.825 Volt, sensor wet at around 0.425 Volt.
  - calibrate_linear:
    - 0.825 -> 0.0
    - 0.425 -> 100.0
```

The above snippet of code configures the reading of the moisture level from the moisture sensor. You can check more information about the moisture sensor configuration here: https://esphome.io/components/sensor/adc.html

Pin: A0 is the pin to which the moisture sensor data pin is connected to.

The calibration is a standard value for the moisture sensor: a reading of 0.825V means that the sensor is dry and a reading of 0.425 means that the sensor is wet. Any other value in between is mapped from 0% to 100%.

The LCD display:

```
# Configuration entry for the temperature/humidity sensor:
i2c:
  sda: 12
  scl: 14

font:
  - file: "font.ttf"
    id: my_font
    size: 20

# Display the moisture in %
display:
  - platform: ssd1306_i2c
    model: "SH1106_128x64"
    reset_pin: 16
    address: 0x3C
    lambda: |-
      it.printf(0, 16, id(my_font), "Mo: %.1f %%", id(moisture_sensor).state);
```

You should be familiar with the above code as it is identical to the code you wrote on you previous Home Assistant exercise. The major change is that the lambda function now renders the value of the moisture sensor.

In case you need a quick refresher:

The above snippet of code configures the LCD display component to render the moisture value. You can check more information about the component setup here: https://esphome.io/components/display/ssd1306.html

Pin: 12 is the equivalent of the D6 pin on the ESP8266 board, the pin to which the display SDA data pin is connected to.

Pin: 14 is the equivalent of the D5 pin on the ESP8266 board, the pin to which the display SCL data pin is connected to.

Make sure that both the **model** and the **address** values are correct for your display component. You can find the model on the component data sheet and the address on the logs during upload.

The **lambda** key is used to print information on the LCD display:
printf(display matrix row, display matrix column, font id, text, reference variable)

Note: remember that the LCD display requires a font file to work correctly. Please refer back to the previous “Temperature and Humidity Sensor” Home Assistant exercise if you need a refresher on how to add fonts to Home Assistant.

The relay switch:

```
# Water pump switcher
switch:
  - platform: gpio
    pin: 2
    name: “Water Pump”
     Inverted: yes
```

The above snippet of code is pretty simple to understand. It is essentially a switch that allows you to turn the water pump on and off.

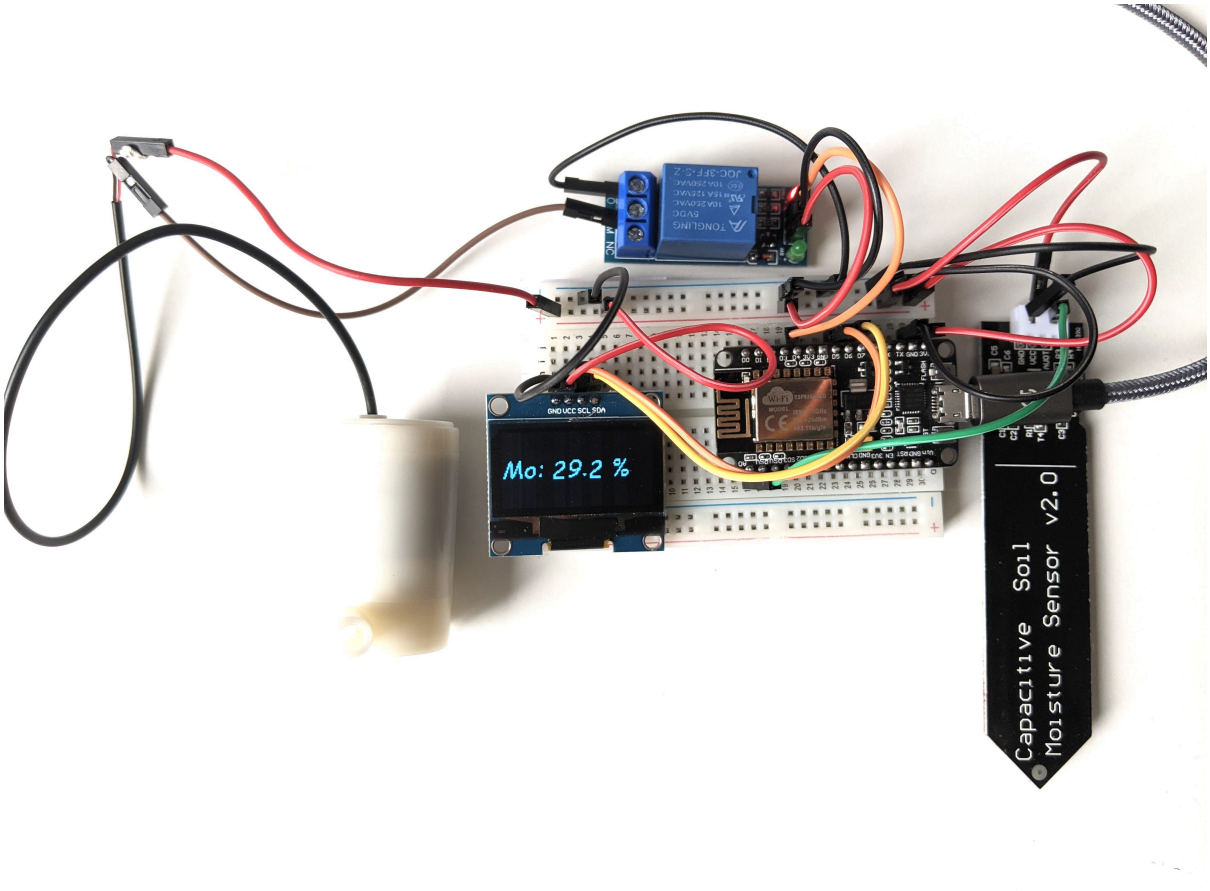
You can check more information about the component setup here:
https://esphome.io/components/switch/gpio.html

Pin: 2 is the equivalent of the D4 pin on the ESP8266 board, the pin to which the relay activation pin is connected to.

Inverted: by default the pin is set to HIGH or ON. You want the water pump to be off when you load the script to the ESP8266 board. Setting it to “yes” makes the pin LOW or OFF.

Excellent, you are now ready to upload the latest firmware to your ESP8266 board. Head back to the Home Assistant ESPHome tab and upload the firmware to your board.

You should now be able to see the moisture data on the LCD display component:



Step Four

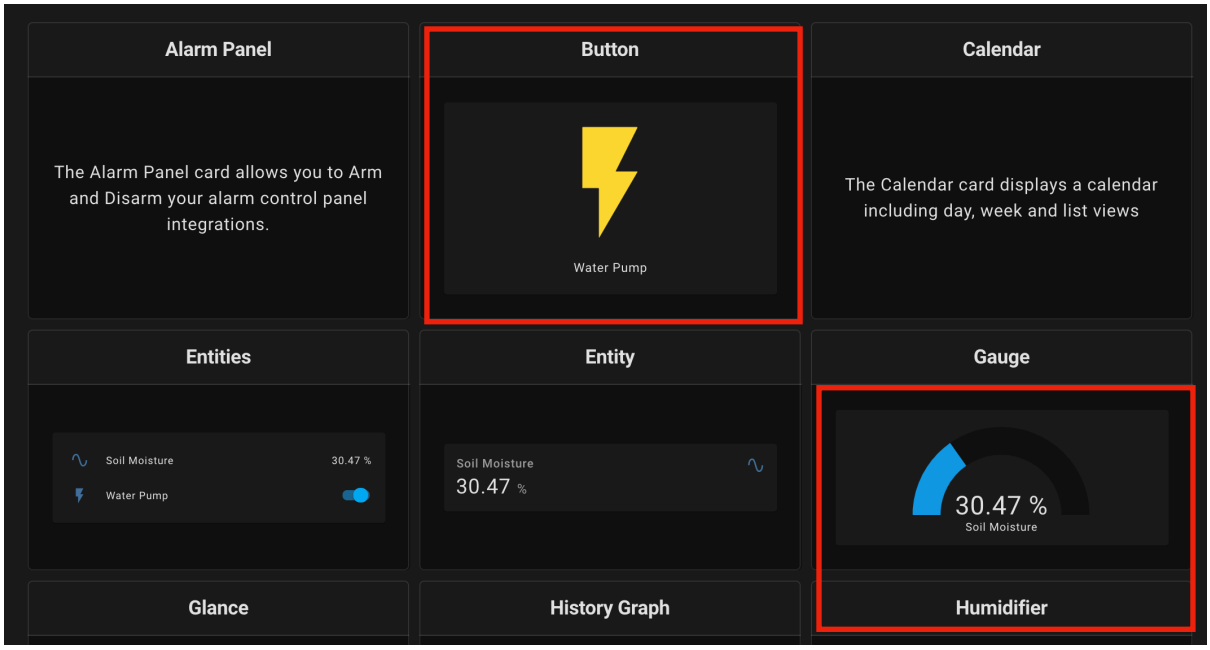
The Moisture and the Water Pump Switch Cards

Great, the circuit is now assembled and you can visualise the moisture data on the LCD screen component. The next step is to create two cards on the Home Assistant dashboard to monitor the moisture level and to manually activate and deactivate the water pump.

Go ahead and add the ‘smart_plant’ integration to your Home Assistant integrations section. If you need a refresher on how to add the integration please refer back to the previous “Temperature and Humidity Sensor” Home Assistant exercise.

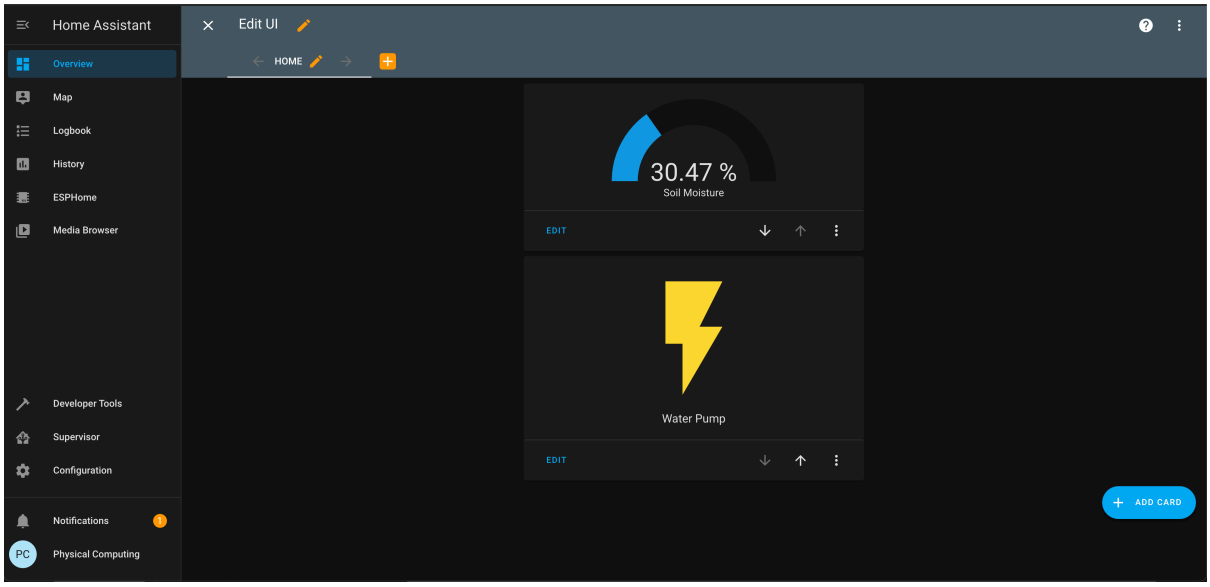
Now that the integration has been configured, you can attach it to a UI card on the Home Assistant dashboard to display your data.

Head to the Home Assistant dashboard and add two cards:



- **The Button card:** you can use this card to turn on and off the water pump
- **The Humidifier:** you can use this card to monitor the moisture level

Now, if you head back to the Home Assistant dashboard you will see your added cards:



The Humidifier card shows the level moisture sent by your moisture sensor. You can now track the moisture sensor both on the LCD display and from Home Assistant.

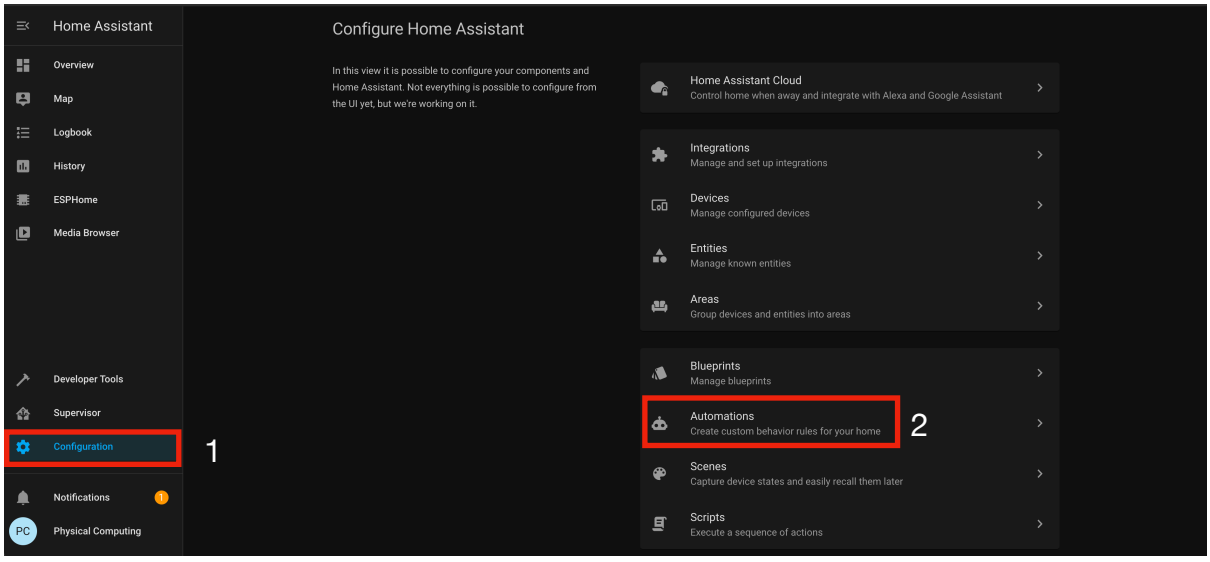
The Button card allows you to turn ON and OFF the water pump. Give it a go and click on the card to turn on the water pump **(do not place your pump inside water yet)**. When you activate the switch, a small electric signal is sent to the activation pin of the relay component. The relay then allows current to flow through the water pump component, turning it on.

Congratulations for getting this far, next you will see how you can automate the water pump activation when the moisture level goes below a threshold level.

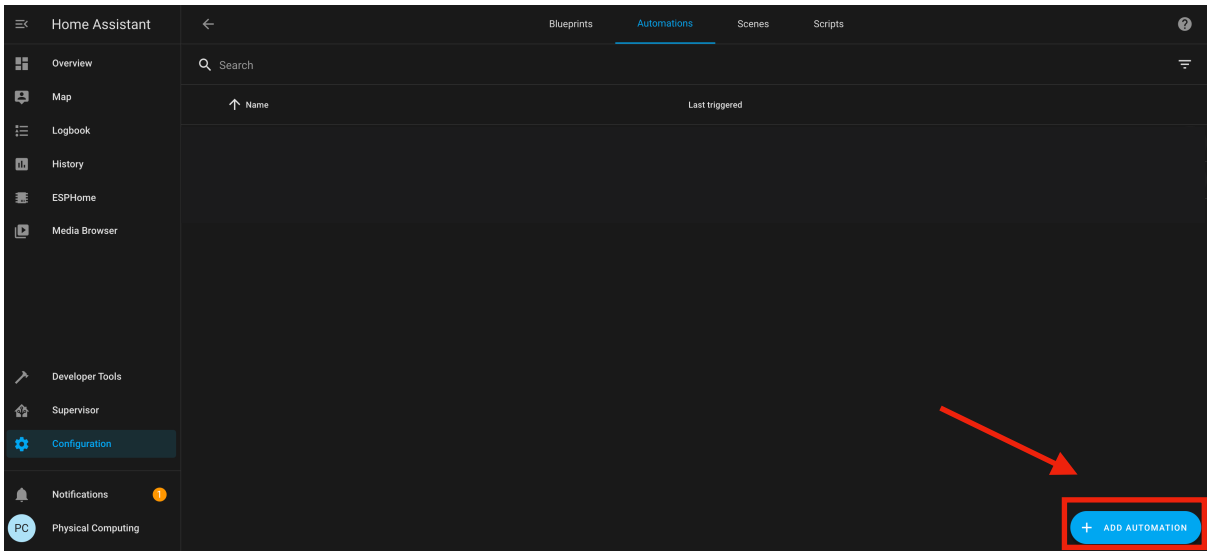
Step Five
Smart Plant Automation

The smart plant project is finally coming together. You can read moisture data, display the data on the LCD screen, and manually turn on and off the water pump. This is great but you do not want to stare at the screen or the LCD display all day long to eventually activate the water pump when the plant soil is dry. Would it be great if you could automate the process? What about turning on the water pump when the soil moisture level is below 30% and turning it off when the soil moisture level is above 70% automatically? Well, Home Assistant offers an easy way to automate your connected devices and components.
Let's see how you can automate the moisture station circuit.

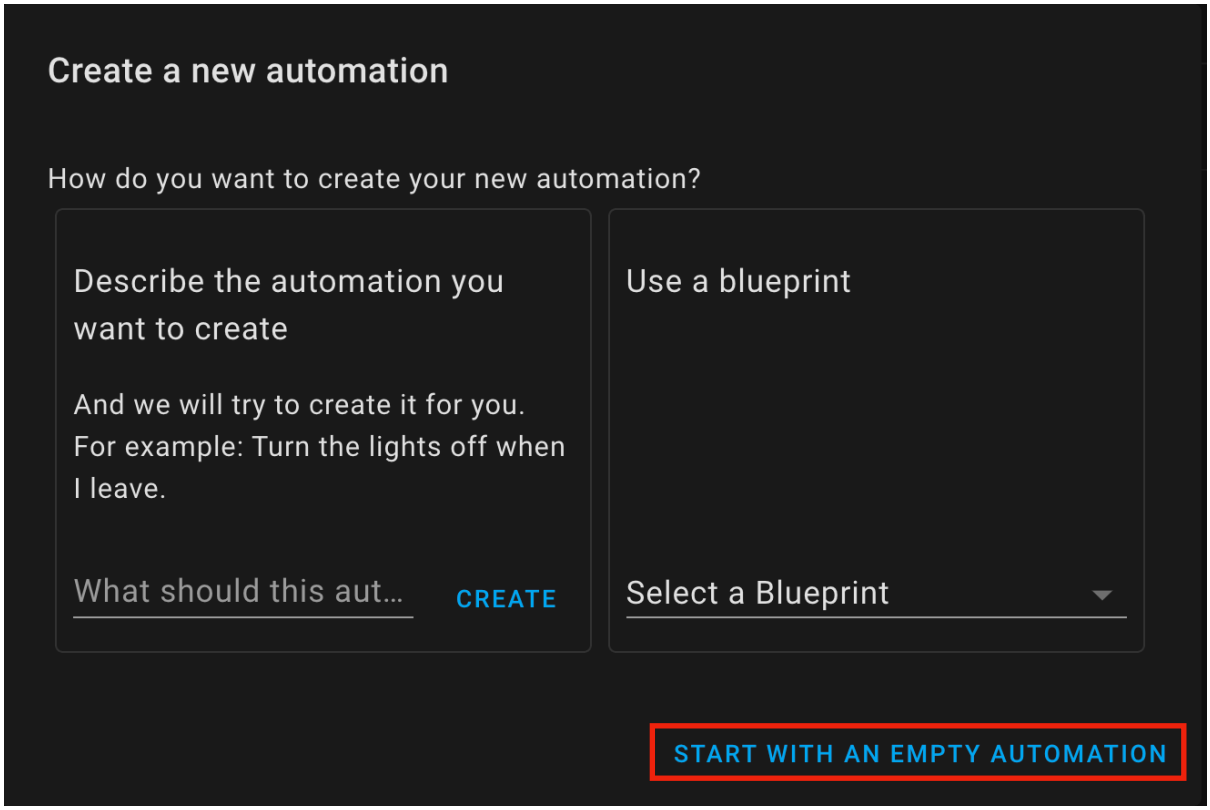
First thing first, head back to the Home Assistant **configuration** section (1), and click on the **Automations** tab (2):



This will open the automation panel of Home Assistant. Initially you will not have any automation registered. Click on the ADD AUTOMATION button:

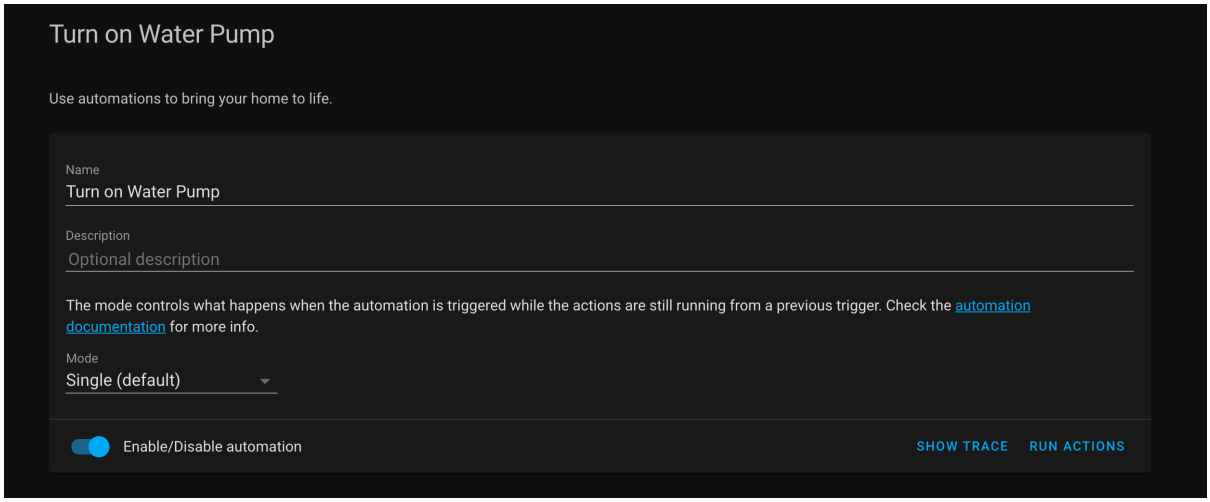


The first automation will turn on the water pump when the moisture level goes below 30%. Click on the START WITH AN EMPTY AUTOMATION button:

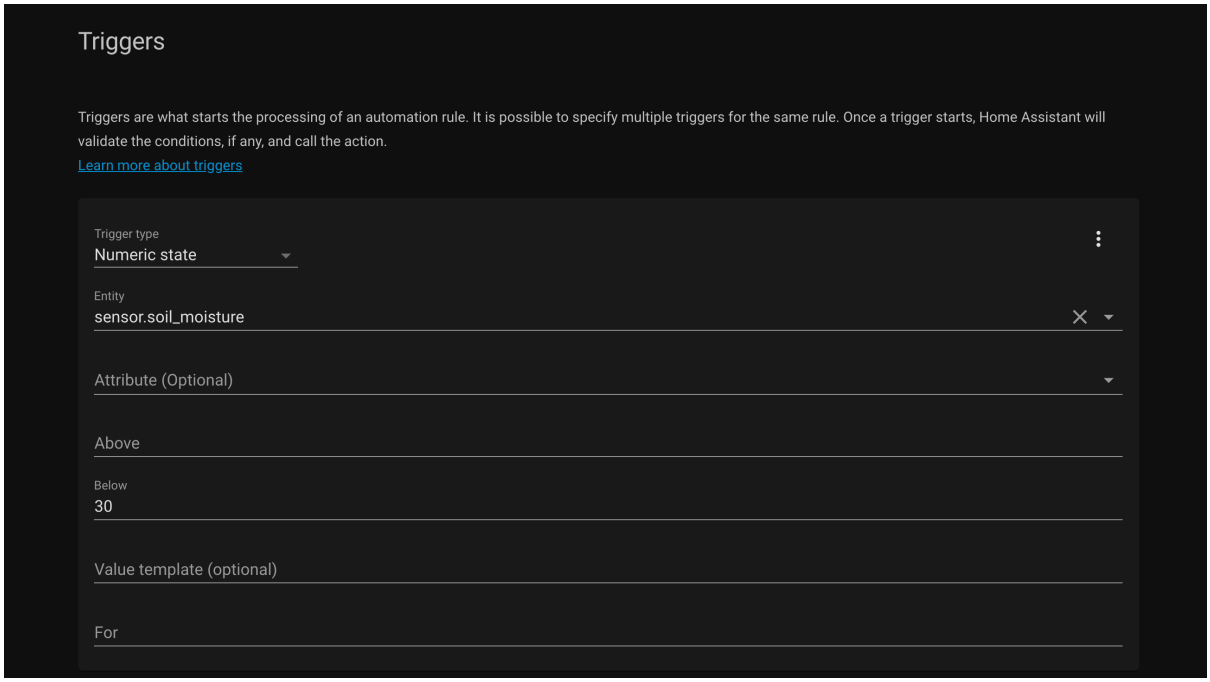


Now configure the automation as follow.

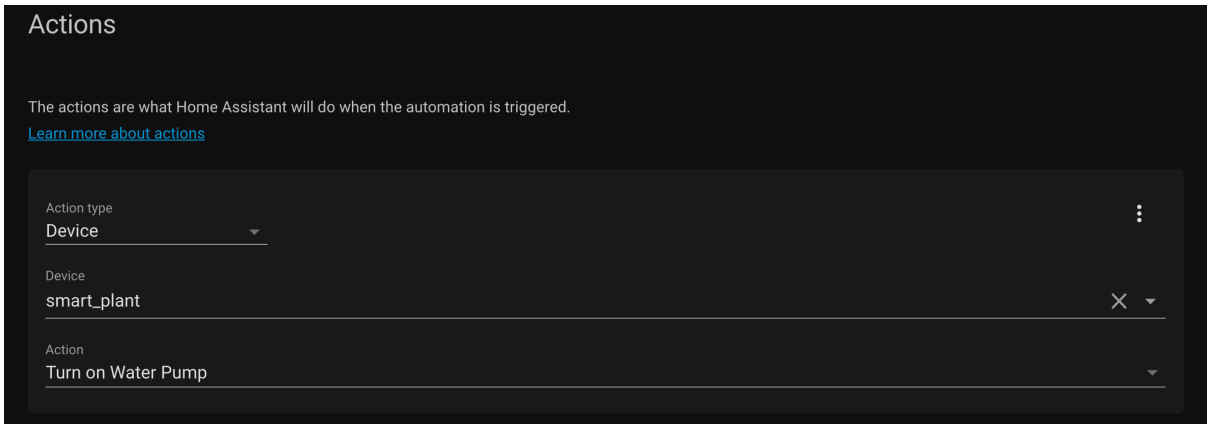
First you have to give your automation task a title. You want to turn on the water pump so “Turn on Water Pump” is reasonable as a title. You can optionally add a description:



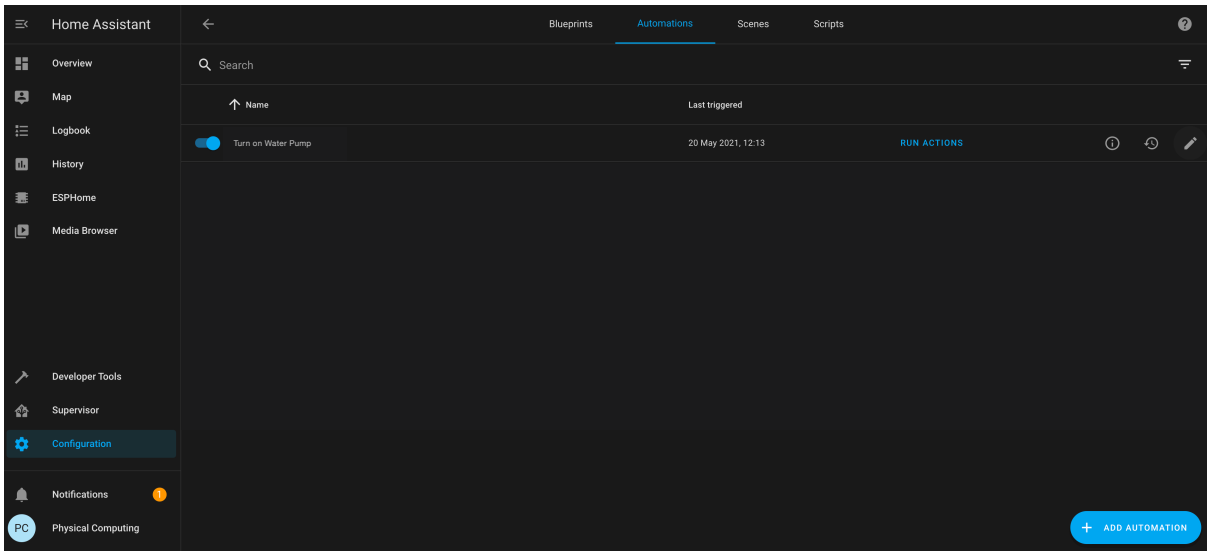
Next you want to set a trigger. Triggers are signals that Home Assistant will listen to. You want to set a trigger when the moisture level is below 30%:



Finally you want to set an action. Actions are a set of steps that Home Assistant will do when the trigger condition is met. You want to turn on the water pump when the moisture level is below 30%:



You first automation is now configured. If you head back to the automation section of Home Assistant, you will see that the “Turn on Water Pump” automation was added to the dashboard:



With you circuit powered up, you can now see that the water pump turns on automatically when the moisture level is below 30%. This is fantastic but can you see a problem here?

The water pump will never turn off. The trigger is set when the moisture level goes below 30%. At that point the water pump turns on and will be activated forever or until you manually turn it off from the button card on your dashboard.

Go ahead and add another automation task to turn off the water pump when the moisture level exceeds 70%. You will need:

- Title: “Turn off the Water Pump” is consistent with your work
- Trigger: add a trigger when the moisture level is above 70%
- Action: turn off the water pump when the trigger is met

Step Six
The smart plant in action

Now that you have successfully added automation, you can finally integrate the circuit to your plant and turn it into a smart entity.

- Place the moisture sensor inside the soil of your plant.
- Place the water pump inside a glass of water. Be careful of not spilling water on the circuit. The voltage and current are very low but they could damage your circuit.
- Arrange both the circuit and the water pipe around the vase of your plant
- Avoid using your computer as a power source for this circuit. You can use a 5V USB portable charger to power your circuit

Here the final result on my Chamaedorea indoor plant: