

## Plant moisture sensing system

This practical session should be a bit of fun for you.

The objective is to build a system to sense the moisture content of the soil in a plant pot and display the value on a dashboard. The hardware used is a Wemos D1 Mini (ESP8266-based microcontroller) mounted on a breadboard and a Raspberry Pi running Node-RED and Mosquitto (MQTT).



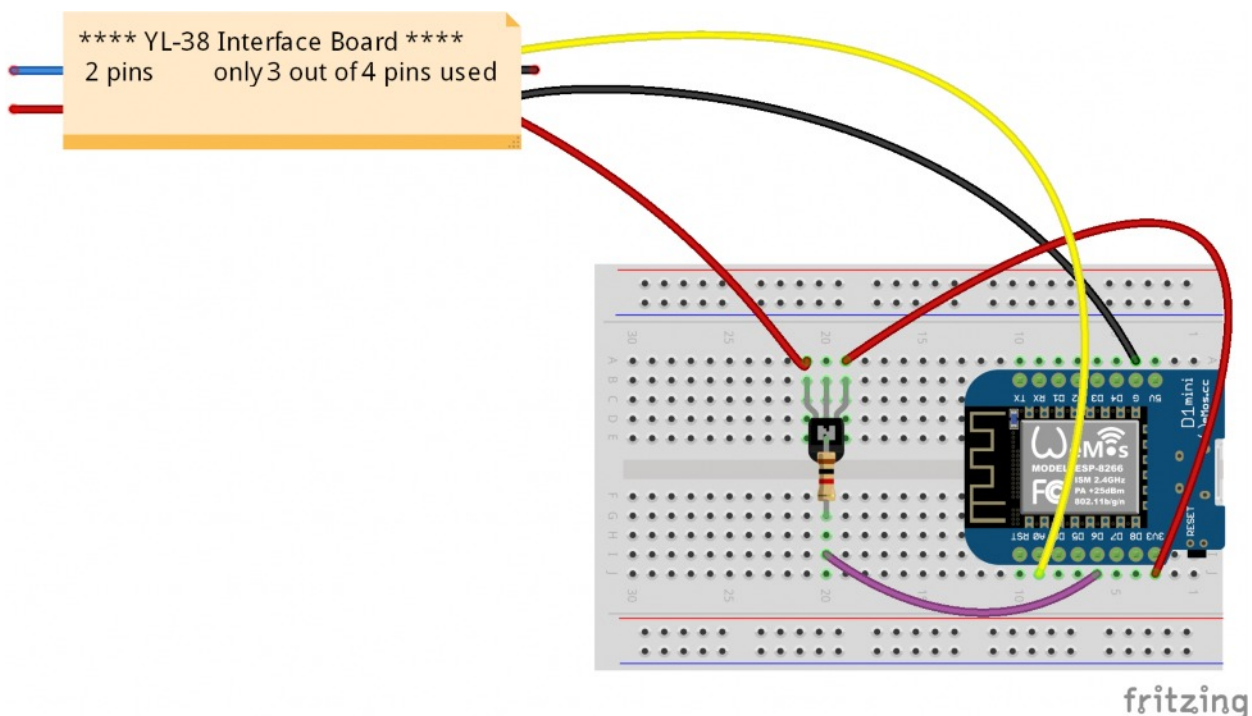
Here's a photo of the sensing part of the system.

The items on the breadboard are:

- \* Wemos D1 Mini
- \* YL-38 miniature interface board
- \* BC547 npn small-signal transistor
- \* 1K ohm resistor
- \* Pieces of single-stranded connecting wire
- \* 5V power block

And of course a plant pot with the soil sensor

Here's a diagram showing the breadboard connections.





## Plant moisture sensing system

Although I've used a BC547 npn transistor any small-signal npn transistor should work as the amount of current being switched is minimal. Connect the transistor's collector to +3V3 on the Wemos D1 Mini, and the emitter to the Vcc pin on the YL-38. The transistor's base is connected to D6 on the Wemos via a 1K ohm resistor.

Connect a ground wire from the Wemos to the YL-38. The analog output from the YL-38 (labelled A0) goes to the A0 pin on the Wemos.

Double-check your wiring BEFORE connecting the 5V power block.

### Setting-up the Wemos D1 Mini

The next step is to set-up the devices and write some rules.

	Task	Enabled	Device	Name	Port	Ctr (IDX)	GPIO	Values
<a href="#">Edit</a>	1	✓	Analog input - internal	paw			ADC (TOUT)	value: <b>10</b>
<a href="#">Edit</a>	2							
<a href="#">Edit</a>	3							
<a href="#">Edit</a>	4							
<a href="#">Edit</a>	5							
<a href="#">Edit</a>	6							
<a href="#">Edit</a>	7							
<a href="#">Edit</a>	8							
<a href="#">Edit</a>	9							
<a href="#">Edit</a>	10							
<a href="#">Edit</a>	11							
<a href="#">Edit</a>	12	✓	Generic - Dummy Device	var				activation: <b>0</b> sent: <b>1</b> : : <b>0</b> <b>0</b>

Two devices are needed.

One to handle the analog input from the moisture sensor and one to create some variables. The Generic - Dummy Device is used for this task.



## Plant moisture sensing system

Here's a screen shot of the settings for the analog input.

### Task Settings

Device: Analog input - internal ?

Name:

Enabled:

Oversampling:

### Two Point Calibration

Calibration Enabled:

Point 1:

Point 2:

### Data Acquisition

Send to Controller 1

---

Interval:  [sec]

### Values

#	Name	Formula <span>?</span>	Decimals
1	<input type="text" value="value"/>	<input type="text" value="(1024-%value%)/100"/>	<input type="text" value="0"/>

Notice the name is “paw” (potentially available water) and the value-name is called “value”. The formula simply converts and scales the analog reading to a value between 0 and 10 (where 10 means very wet and 0 very dry).

Notice the sampling interval is set to 5-seconds.

Also notice that the “Send to Controller” option is left un-ticked.

Right that's the analog bit sorted.

### Setting-up the named variables

This system makes use of “rules” and “timers” and “named variables”.

The named variables are declared by using a Generic - Dummy Device which is described on the next page.



## Plant moisture sensing system

I've named the Generic Dummy Device as "var" and used two of the four available names (e.g. "activation" and "sent") as shown below.

### Task Settings

Device: Generic - Dummy Device ?

Name:

Enabled:

Simulate Data Type:

### Data Acquisition

Send to Controller  **1**

---

Interval:  [sec]

### Values

#	Name	Decimals
1	<input type="text" value="activation"/>	<input type="text" value="0"/>
2	<input type="text" value="sent"/>	<input type="text" value="0"/>
3	<input type="text"/>	<input type="text" value="0"/>
4	<input type="text"/>	<input type="text" value="0"/>

You can set a value in these variables (in the rules) by using:

`TaskValueSet,12,1,number //sets 'activation' to a number`

`TaskValueSet,12,2,number //sets 'sent' to a number`

You can read the value held in a variable (in the rules) by using:

`[var#activation]` or `[var#sent]`

### Writing some simple rules

The listings on the next page shows the various rules used in the system.



## *Plant moisture sensing system*

The first rule runs at system boot-up and resets Timer1 and the variables.

```
on System#boot do
    timerSet,1,0
    TaskValueSet,12,1,0    //Set var#activation = 0
    TaskValueSet,12,2,0    //Set var#sent = 0
endon
```

The second rule is triggered when an event-command is issued from Node-RED. The npn transistor (connected to D6) is switched-on, an acknowledgement message is sent back to Node-RED (via MQTT), the two named variables are set and Timer1 is triggered for 6-seconds.

```
on activate do
    gpio,12,1    //Set GPIO(D6)=1 (Turn on the transistor)
    publish node26/status, "activated"
    TaskValueSet,12,1,1    //Set var#activation = 1
    TaskValueSet,12,2,0    //Set var#sent = 0
    timerSet,1,6
endon
```

The next rule is triggered whenever the analog input changes.

```
on paw#value do
    if [var#activation]>0 and [var#sent]=0
        publish node26/reading, {"paw": [paw#value]}
        TaskValueSet,12,2,1 //Set var#sent = 1
    endif
endon
```

Providing the variable [var#activation] is set and the variable [var#sent] is clear then the analog reading is published to Node-RED via MQTT. At the same time the flag [var#sent] is set to stop any further transfers.



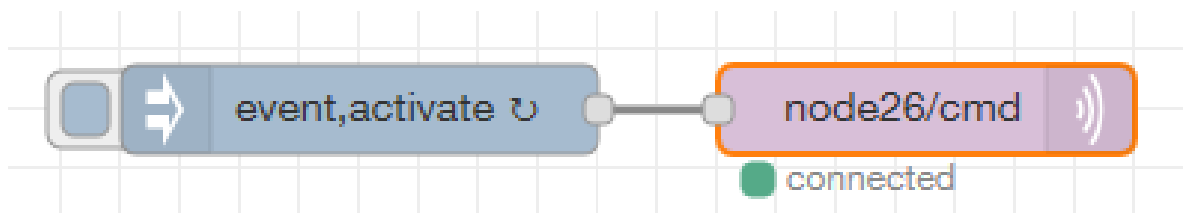
The last rule is triggered when Timer1 expires.

```
on Rules#timer=1 do
  gpio,12,0          //Set GPIO(D6)=0 (Turn off the transistor)
  publish node26/status, "deactivated"
  TaskValueSet,12,1,0 //Set var#activation = 0
endon
```

This rule switches off the transistor and publishes a message to Node-RED via MQTT. It also resets the variable [var#activation] to zero - ready for the next trigger sequence.

### Setting-up the Node-RED flows

This part of the flow sends an event to trigger the Wemos D1 Mini.



This screen shot shows the setting for the 'Inject' node.

✉ Payload	▼ a_z event, activate
☰ Topic	

This screen shot shows the settings for the MQTT send node

🌐 Server	192.168.1.142:1883
☰ Topic	node26/cmd



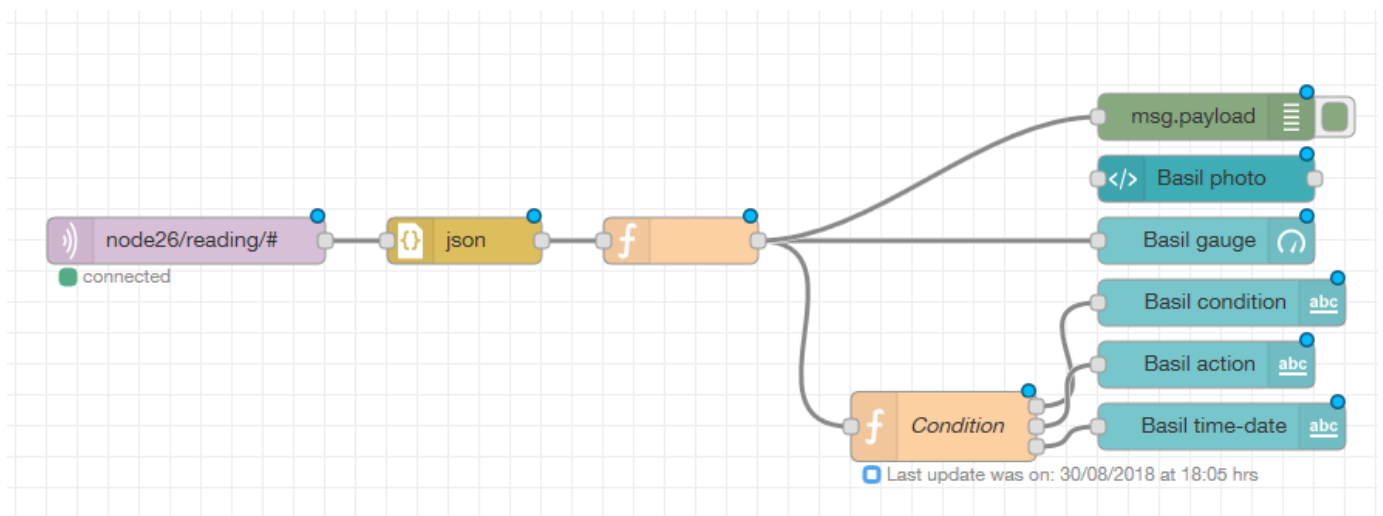
## Plant moisture sensing system

### Feedback

This optional part of the flow detects status-feedback from the Wemos D1 Mini.

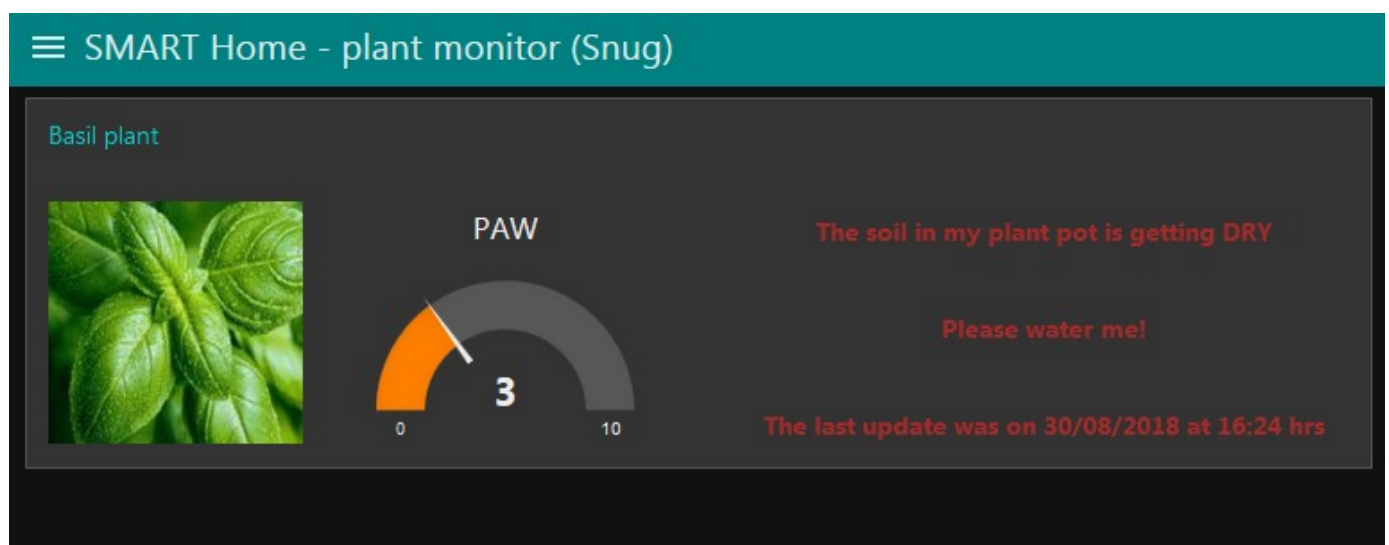


This is the main flow that processes the data before sending it to the dashboard gauge and text boxes.



### Dashboard

Here's a screen shot of the dashboard.



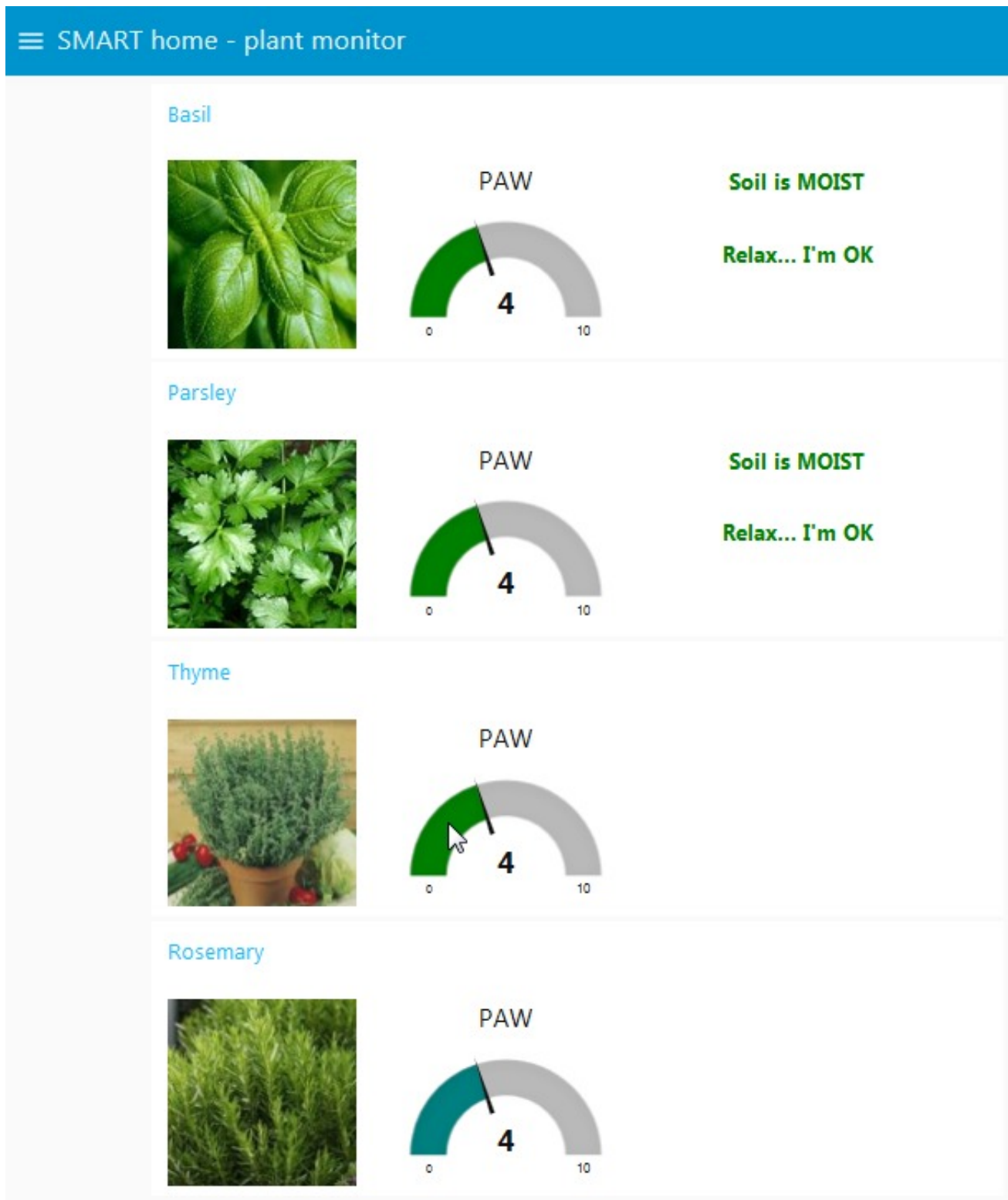
It is made up of three elements: an image, a gauge and a text box. I'm sure you could construct a much better-looking dashboard.



## Plant moisture sensing system

### Future enhancement

One of the things I would like to do is add extra sensors so I could monitor a group of plants. The best way to achieve this is to make use of an analog multiplexor like the ADS1115. This is a 4-channel device which means I could monitor four pot plants as shown on the demonstration dashboard below.



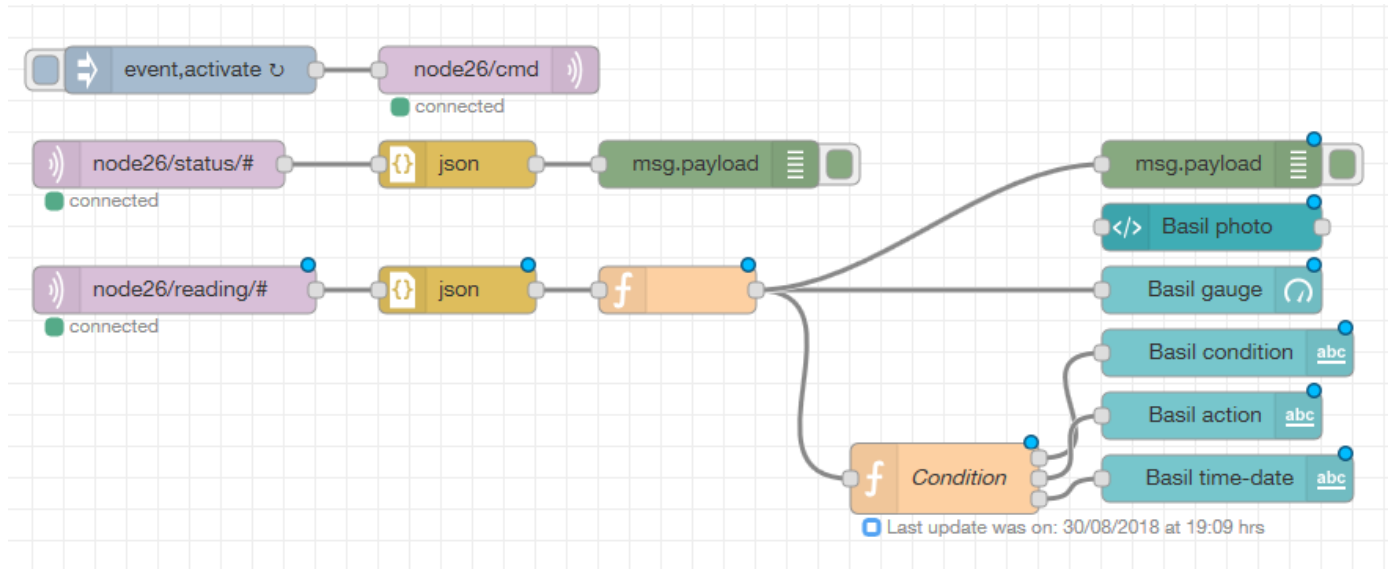




## Plant moisture sensing system

### Node-RED flow (graphical)

Here's a view of the complete Node-RED flow.



### Node-RED flow (textual)

Here's a link to the text file for the flow.

[http://resources-area.co.uk/node-red-flows/soil\\_moisture\\_monitor.txt](http://resources-area.co.uk/node-red-flows/soil_moisture_monitor.txt)

You need to copy and paste this text file into Node-RED using the “Import clipboard” facility on the top-right of Node-RED’s main screen.

### Summary

This tutorial has shown how Node-RED can communicate with a Wemos D1 Mini to trigger a series of actions - in this case to read the soil-moisture value.

It has also shown how a simple, but visually attractive, dashboard can be constructed using the basic elements available in Node-RED.

There are many optional extras that could be introduced into this project - for example an OLED display panel could be connected to the Wemos D1 Mini to show the moisture value and watering actions that might be needed. You could also introduce an electric pump to automatically water the plants.

I hope you have enjoyed reading through this tutorial and will be inspired to have a go at creating your very own soil moisture monitoring system.