Irrigation System Automation Using Home Assistant

Summary

- Design an affordable irrigation system
- Use automation for efficiency and avoid water waste
- Never forget to water plants
- Access remotely
- Analyze network hierarchy features

Use Case

The design implements IoT technology that provides an automatic irrigation system thereby saving time, money, and power for farms and gardens.

The traditional farmland irrigation techniques require manual intervention, furthermore the using automated irrigation technology, the human intervention can be minimized.

Home Assistant Background

Home Assistant is an open-source home automation system. Originally, created by Paulus Schoutsen written in Python 3.8 and was first published on GitHub in November 2013. In July 2017, Home Assistant was introduced to an operating system called Hass.io to be installed on a Raspberry Pi in order to manage backups, updates, and extending the functionality options of the software with add-ons.

Home Assistant Setup

1. Attach SD card to computer 

2. Download and start Balena Etcher
3. Select “Flash from URL” for the Raspberry Pi
4. Paste the URL into Balena Etcher and click “OK”
5. Balena Etcher will download the image, when that is done click “Select target”

6. Select the SD card for the Raspberry Pi
7. Click on “Flash!” to start writing the image
8. Confirmation

Welcome to Home Assistant Installation Process

- Insert the installation media (SD card) into the Pi
- Attach the ethernet cable
- Attach the power cable
- Within a few minutes able to access Home Assistant on homeassistant.local:8123

Home Assistant Add ons

- **ESPHome**
- **Node-RED**
- **File editor**
- **SSH & Web Terminal**

ESP32

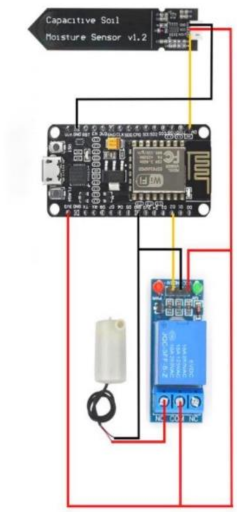
- ESP32 is a microcontroller chip with integrated Wi-Fi
- It is a power amplifier, safe, reliable, and scalable to a variety of applications.
- Easy to setup and integrate into home assistant
- For the purpose of this project, ESP32 was used to power and manage data

Additional Parts Irrigation System

- Water Pump
- Soil Moisture Sensor
- Capacitive Soil Moisture Sensor
- 1 Channel 5V Relay
- Water Pump + 1M Vinyl Tubing
- DHT22 Temperature Humidity Sensor

Wiring

- The ESP32 is powered vai 3V3 and GND
- Moisture sensor has 3 wires, the black is connected to the GND, Red VCC is power the Yellow AOUT is for the Data
- Notice the relay and water pump also powered via the VCC Red Wire and GND is The Black Wire



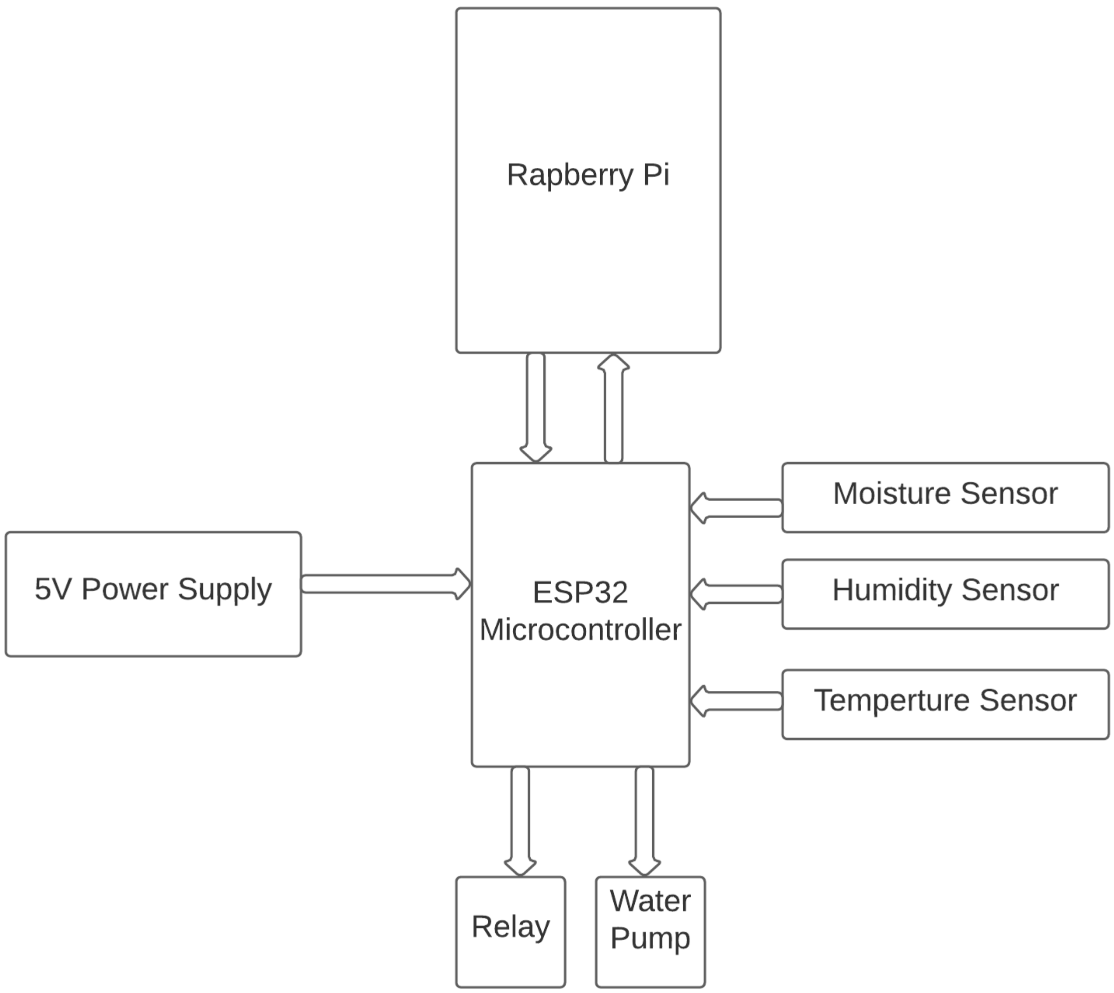
Yaml Code

```
esphome:
  name: node_one_plant
  platform: ESP32
  board: esp-wrover-kit
wifi:
  ssid: "Name"
  password: "passwor"


switch:
  - platform: gpio
    pin: 23
    name: "Activate"
    inverted: yes
```

```
sensor:
  - platform: dht
    pin: 25
    model: dht22
    temperature:
      name: "Temperature"
    humidity:
      name: "Humidity"
    update_interval: 30s

sensor:
  - platform: adc
    pin: 33
    name: "Soil Moisture Level"
    update_interval: 1s
    unit_of_measurement: "%"
    icon: "mdi:flower-outline"
    attenuation: 11db
    filters:
    - median:
        window_size: 7
        send_every: 2
        send_first_at: 1
    - calibrate_linear:
        - 1.25 -> 100.00  #dry
        - 2.8 -> 0.00     #wet
    - lambda: if (x < 0) return 0; else if (x > 100) return 100; else return (x);
```

Block Diagram of the entire system

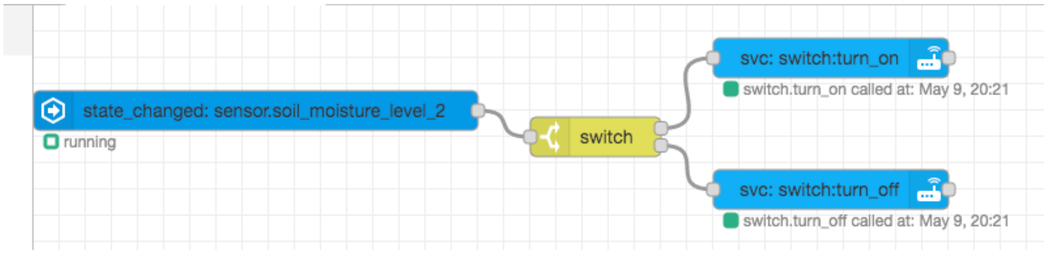
- The soil moisture, temperature & humidity sensors, and water pump, are in sync with the ESP32 updating the Pi with values periodically on Home Assistant.
- While the ESP32 is communicating with the Pi the Macbook is also engaged in the conversation with the Pi to display and send data.



Automation using Nod-RED

- Create an event this case a sensor event
- Connected to a switch

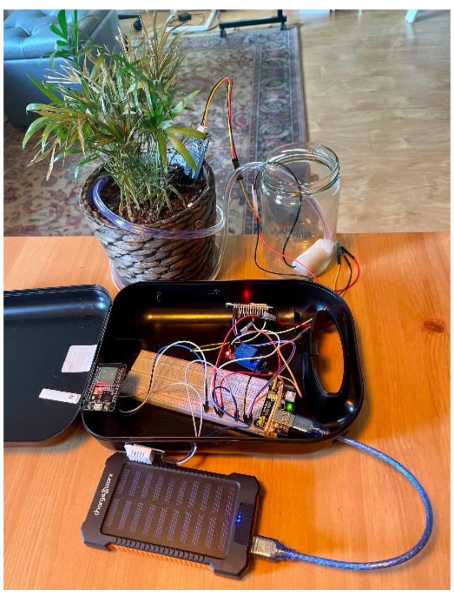
- Turn on/off



Prototype

The prototype is showing the microcontrollers one is connected to temperature & humidity sensor, and the other to soil moisture sensor, relay, and water pump.

These sensors sense the various parameter of the soil, temp & humidity, and the motor is used to provide water upon need, and finally the relay control the water pump.



Demo: http://drive.google.com/file/d/1SJH3JUE4xARLlg1hML80Bh-fEqLD68TB/view

Communication Analysis

TCPDUMP

- Install & run
- Captured data on the Pi
- Uploaded data onto Wireshark

IP Addresses:

- Macbook == 192.168.0.62
- Pi HA == 192.168.0.103
- Gateway == 192.168.0.1
- Node1P == 192.168.0.158
- Node2H&T == 192.168.0.82

Wireshark · Conversations · capture3.pcap

Ethernet | IPv4 · 24 | IPv6 · 3 | TCP · 43 | UDP · 28

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.0.158	224.0.0.251	2	222	2	222	0	0	82.193815	0.0994	17k	0
192.168.0.103	192.168.0.158	216	14k	114	6894	102	7300	0.073398	169.3957	325	344
192.168.0.103	192.168.0.104	180	21k	108	7530	72	14k	9.874274	150.0270	401	754
192.168.0.103	192.168.0.128	123	15k	73	5064	50	10k	9.874274	150.0248	270	548
192.168.0.103	239.255.255.250	3	414	3	414	0	0	44.925833	120.0043	27	0
192.168.0.103	224.0.0.22	12	720	12	720	0	0	44.935360	124.8119	46	0
192.168.0.103	192.168.0.140	15	9399	0	0	15	9399	46.120325	122.6685	0	612
192.168.0.103	224.0.0.251	7	702	7	702	0	0	81.960501	28.8065	194	0
192.168.0.82	192.168.0.103	61	3930	25	1734	36	2196	1.191444	165.6309	83	106
192.168.0.82	224.0.0.251	2	220	2	220	0	0	82.181716	0.1003	17k	0
192.168.0.62	192.168.0.103	361	49k	201	19k	160	29k	0.000000	170.0890	927	1406
192.168.0.62	224.0.0.251	11	1371	11	1371	0	0	3.058585	153.1885	71	0
192.168.0.62	192.168.0.255	5	460	5	460	0	0	26.722898	120.0220	30	0
192.168.0.55	224.0.0.251	16	5838	16	5838	0	0	109.706631	5.3331	8757	0
192.168.0.1	192.168.0.103	24	7854	24	7854	0	0	44.926431	120.0050	523	0
172.30.32.3	224.0.0.251	32	2976	32	2976	0	0	81.959938	10.0025	2380	0
172.30.32.3	224.0.0.22	48	2880	48	2880	0	0	81.971306	18.7842	1226	0
172.30.32.2	172.30.32.6	24	2170	14	1284	10	886	114.511835	15.1298	678	468
172.30.32.1	224.0.0.251	248	61k	248	61k	0	0	3.058975	153.1887	3196	0
172.30.32.1	172.30.32.2	628	244k	312	204k	316	39k	11.642268	157.8327	10k	1999
172.30.32.1	172.30.33.0	18	1320	12	876	6	444	42.171721	120.8751	57	29
172.30.32.1	172.30.32.3	8	800	4	360	4	440	130.001139	0.0008	—	—
169.254.239.55	239.255.255.250	1	203	1	203	0	0	165.878751	0.0000	—	—
127.0.0.1	127.0.0.1	6	435	6	435	0	0	42.171155	120.8749	28	0

Conclusion

This system ensures an adequate water supply that nourishes the soil and avoiding water waste while minimizing human intervention.

This project allowed to have hands on experience with IoT devices for highly effective and safe use of water resources for agricultural production. It demonstrates efficiency and inexpensive idea that is paving the path future level of agricultural development.