Smart Plant Monitoring

By Team Exemplary

Asm Nurussafa
Tasawar Siddiquy
Arfat Kamal
Nirojan Navaratnarajah

For the module Advanced Embedded System.

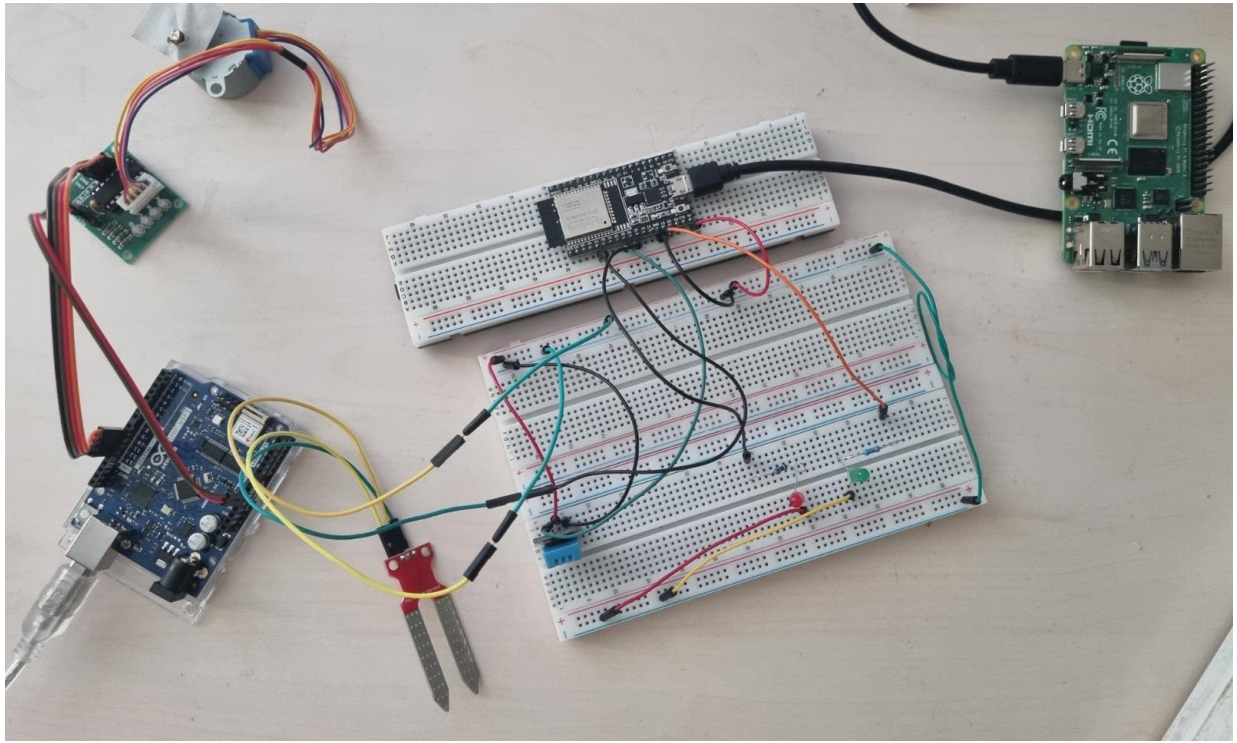
Picture source: https://www.vecteezy.com/vector-art/3223354-smart-plant-pot-icon-on-white

Agenda

1. Motivation. (Arfat)
2. Concept description . (Arfat)
3. Technologies used. (Tasawar)
4. Applications and use cases. (Nurussafa)
5. Implementation description. (Neero)
6. Video demonstration. (Neero)
7. Summary.

1. Motivation

2. Concept Description

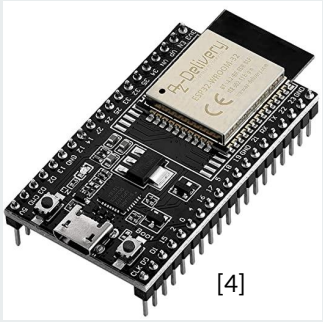


Hardware components used in the project.

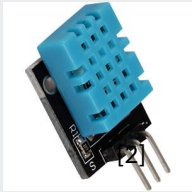
Broker

[1]

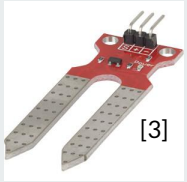
Client 01



[4]



[2]



[3]

Client 03

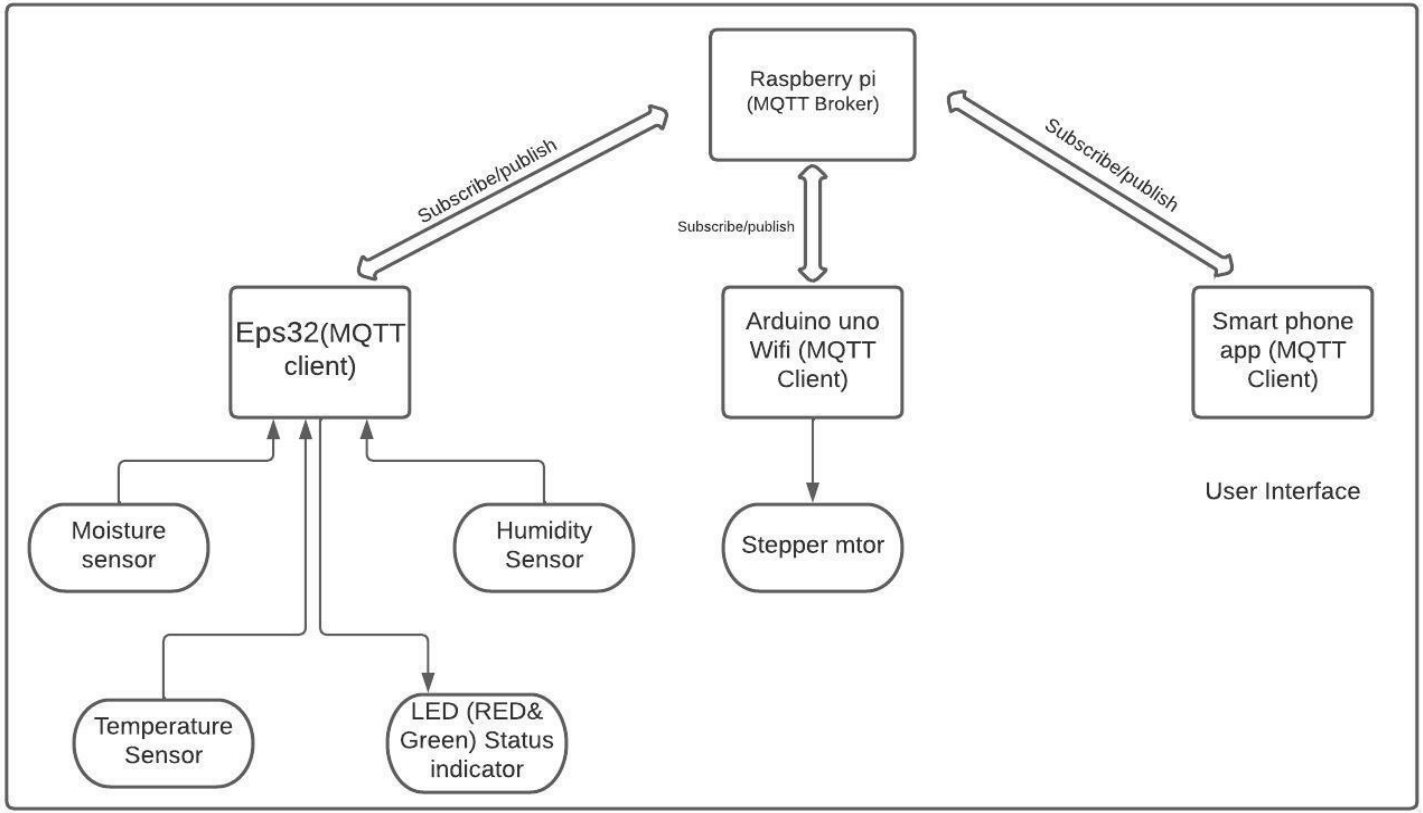
[7]

Client 02

[5]

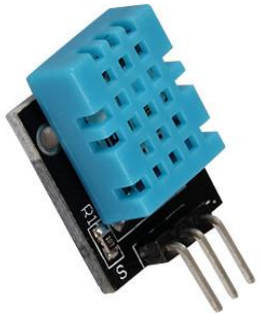
[6]

Block diagram

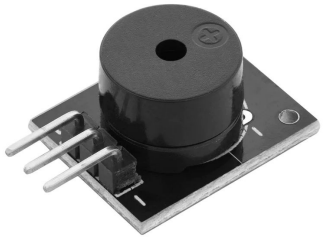


3. Technologies

Temperature and Humidity Sensor (DHT11) & Soil Moisture Sensor (ME110)



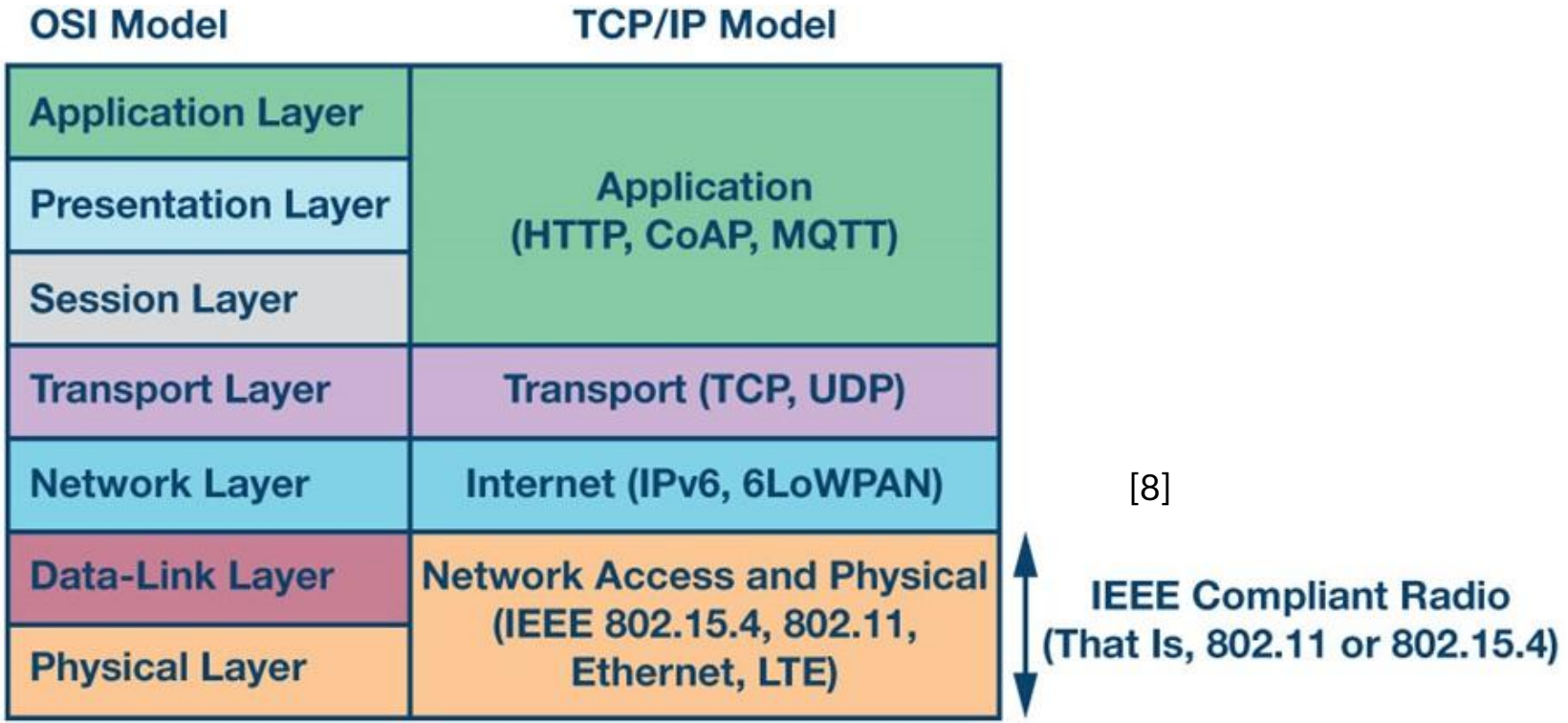
LED , Buzzer & Stepper motor



MQTT - Message Queuing Telemetry Transport

MQTT is a bi-directional communication protocol where each client can both produce and consume data by publishing messages and subscribing to topics

MQTT protocol is **an Application layer protocol**.



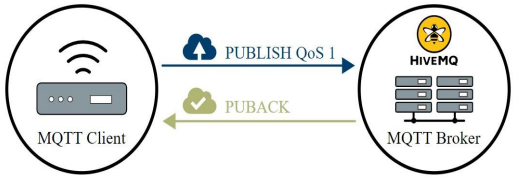
[8]

QOS Levels

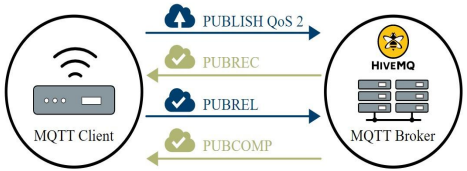
There are 3 QoS levels in MQTT: (Quality of Service)

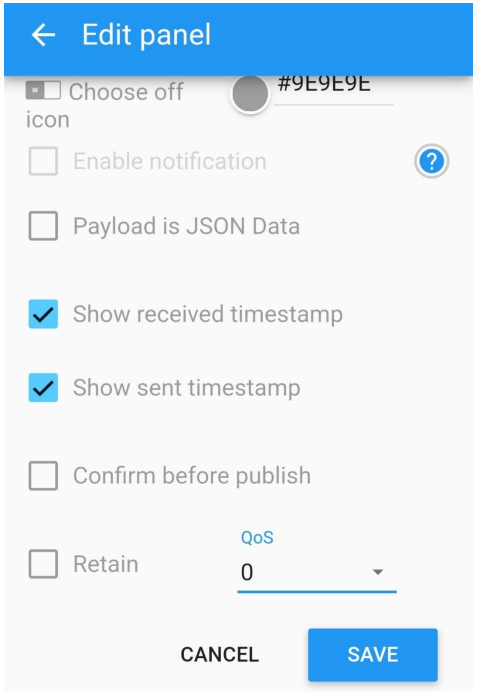
- *At most once* (0)

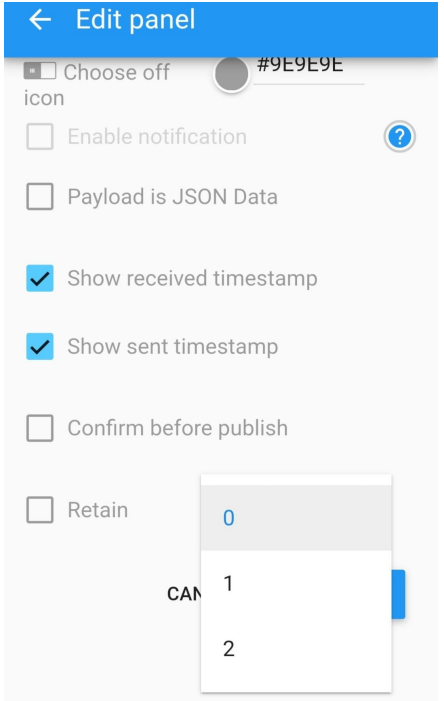
- *At least once* (1)



- *Exactly once* (2).

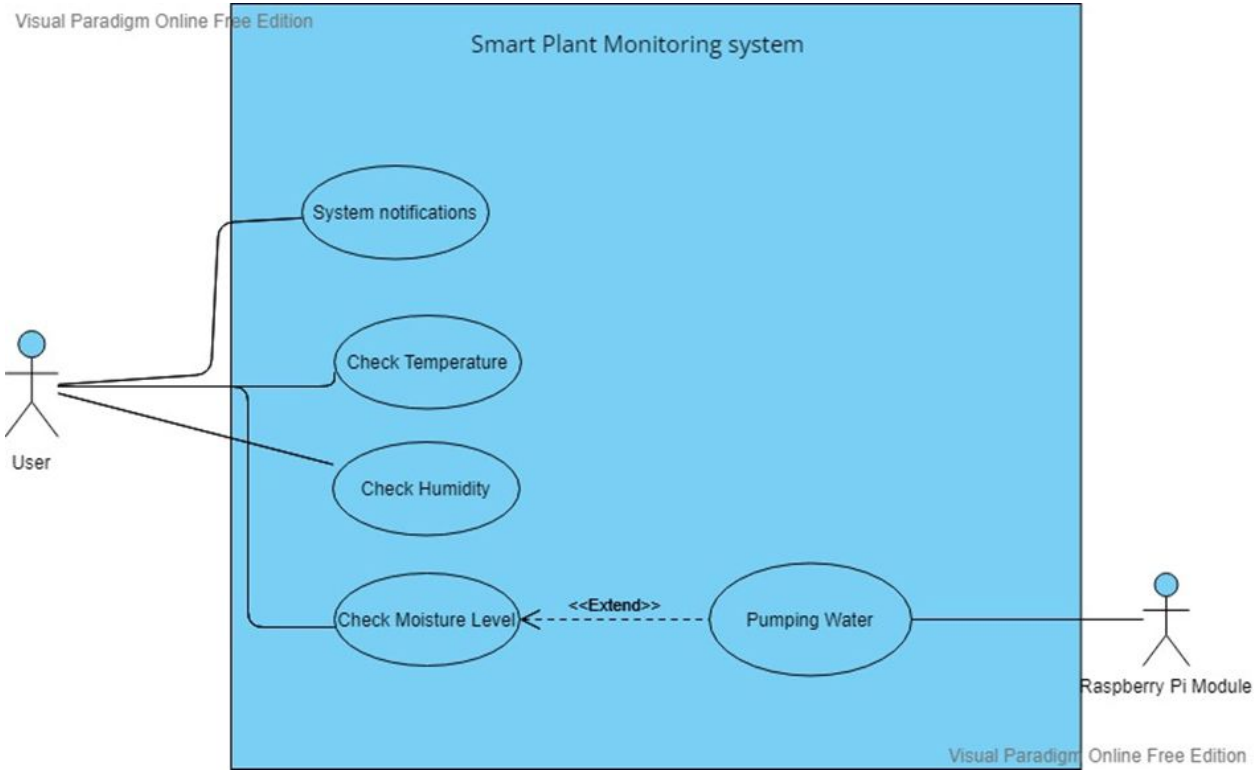




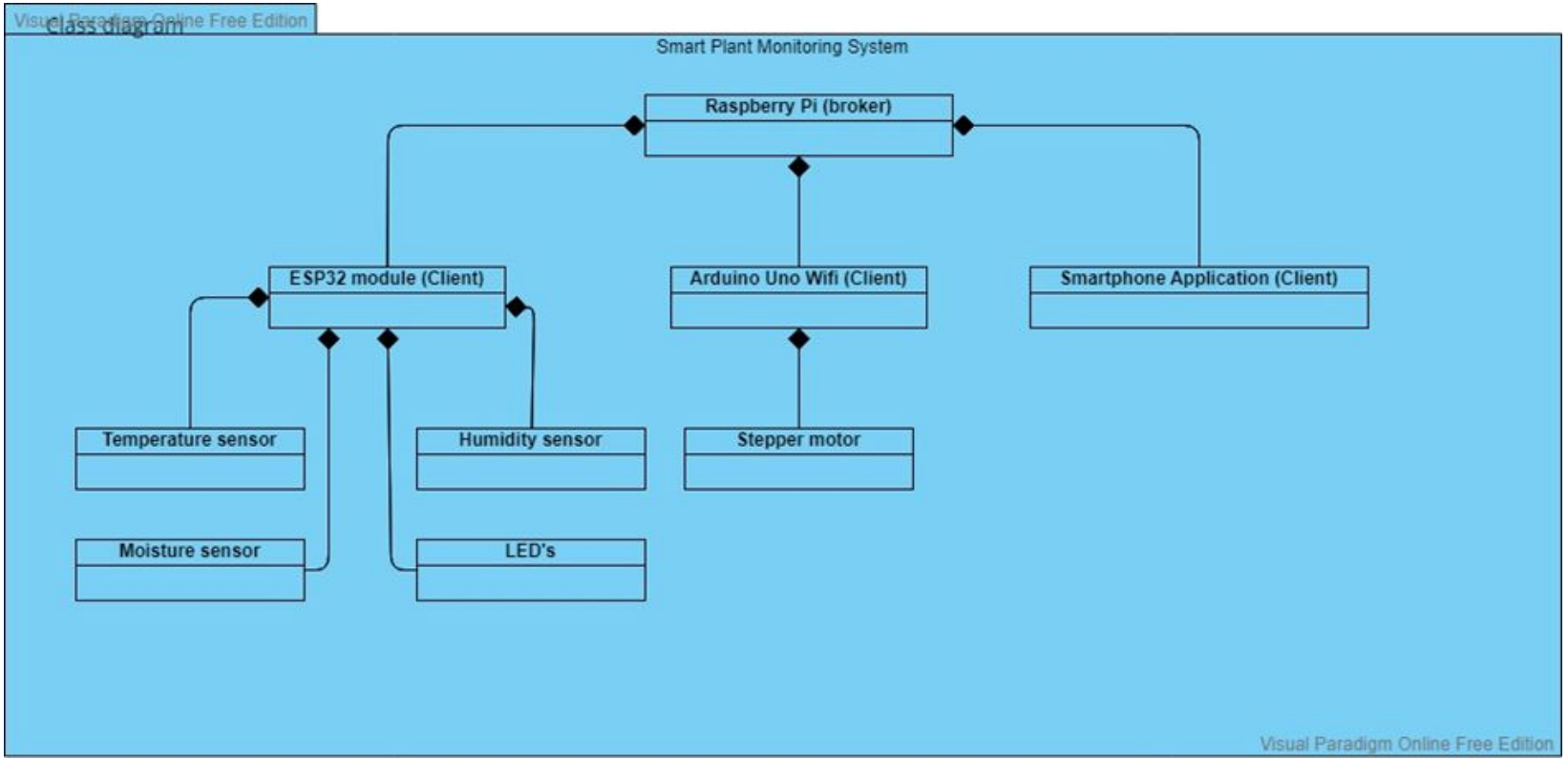


4. Application and use cases

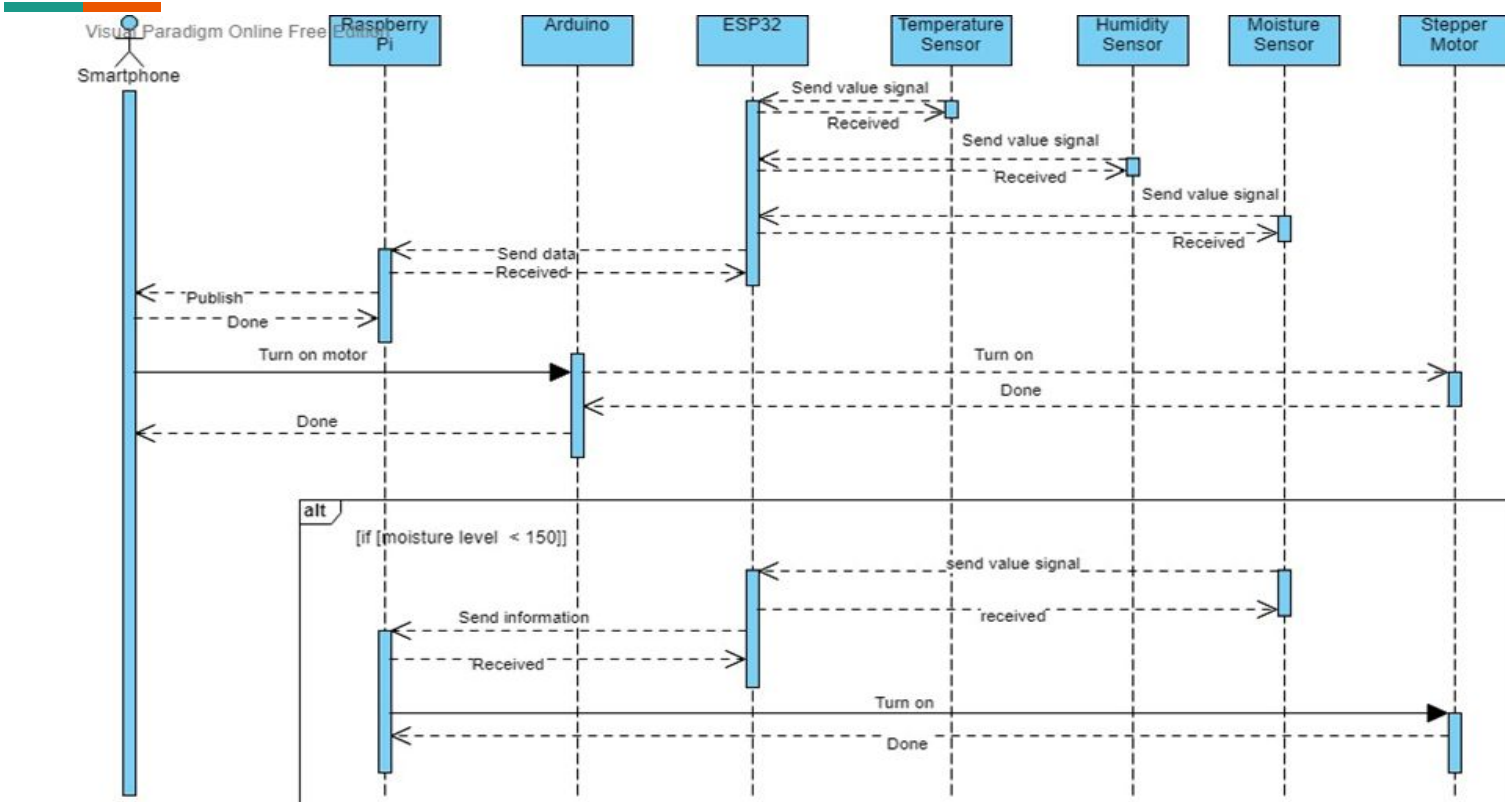
Use cases:


Visual Paradigm Online Free Edition
Smart Plant Monitoring system
System notifications
Check Temperature
Check Humidity
Check Moisture Level
<<Extend>>
Pumping Water
User
Raspberry Pi Module
Visual Paradigm Online Free Edition

Class diagram:

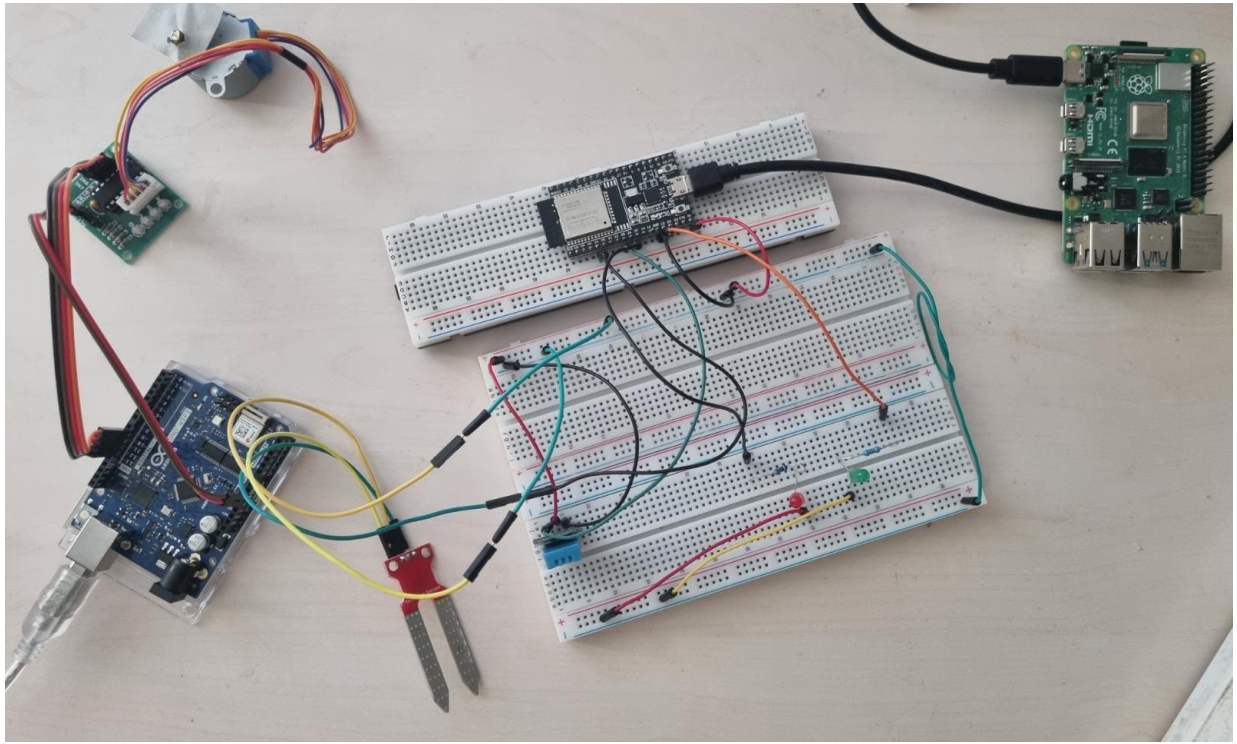


Sequence diagram:

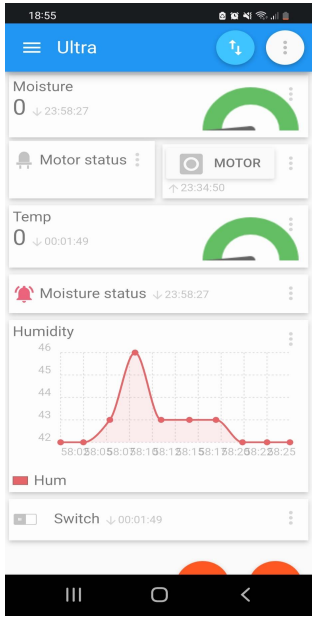


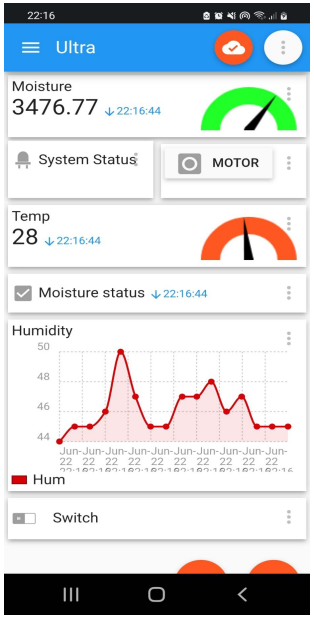
5. Implementation description

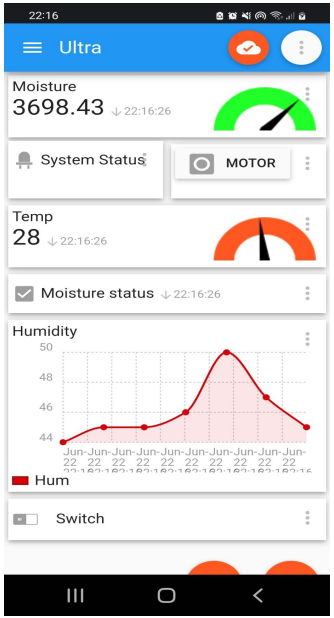
Static structure



User Interface







Arduino code

```
char ssid[] = "Nirojan98";        // your network SSID
char pass[] = "nirojan98";    // your network password

WiFiClient wifiClient;
MqttClient mqttClient(wifiClient);

const char broker[] = "192.168.229.1";
int        port     = 1883;
const char topic[]  = "esp32/motor";
```

```
void onMqttMessage(int messageSize) {
  // we received a message, print out the topic and contents
  Serial.println("Received a message with topic '");
  Serial.print(mqttClient.messageTopic());

// step one revolution in one direction:
if (mqttClient.messageTopic()){
  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(1000);

  // step one revolution in the other direction:
  Serial.println("counterclockwise");
  myStepper.step(-stepsPerRevolution);
  delay(1000);}
```

Esp32 code

```
const char* ssid = "Nirojan98"; // mobile hotspot
const char* password = "nirojan98";

// Add your MQTT Broker IP address, example:
const char* mqtt_server = "192.168.229.2";   //--- IP address of Raspbery pi
```

```
float temp = DHT.temperature;

pointer_to_hum_string = dtostrf(hum, 6, 2, msg_hum_Buffer);
client.publish("esp32/hum",pointer_to_hum_string);

pointer_to_created_string = dtostrf(temp, 6, 2, msgBuffer);

client.publish("esp32/temp",pointer_to_created_string);
```

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("ESP8266Client")) {
      Serial.println("connected");
      // Subscribe
      client.subscribe("esp32/output");
      client.subscribe("esp32/motor");
```

Automating the whole process of watering the plant

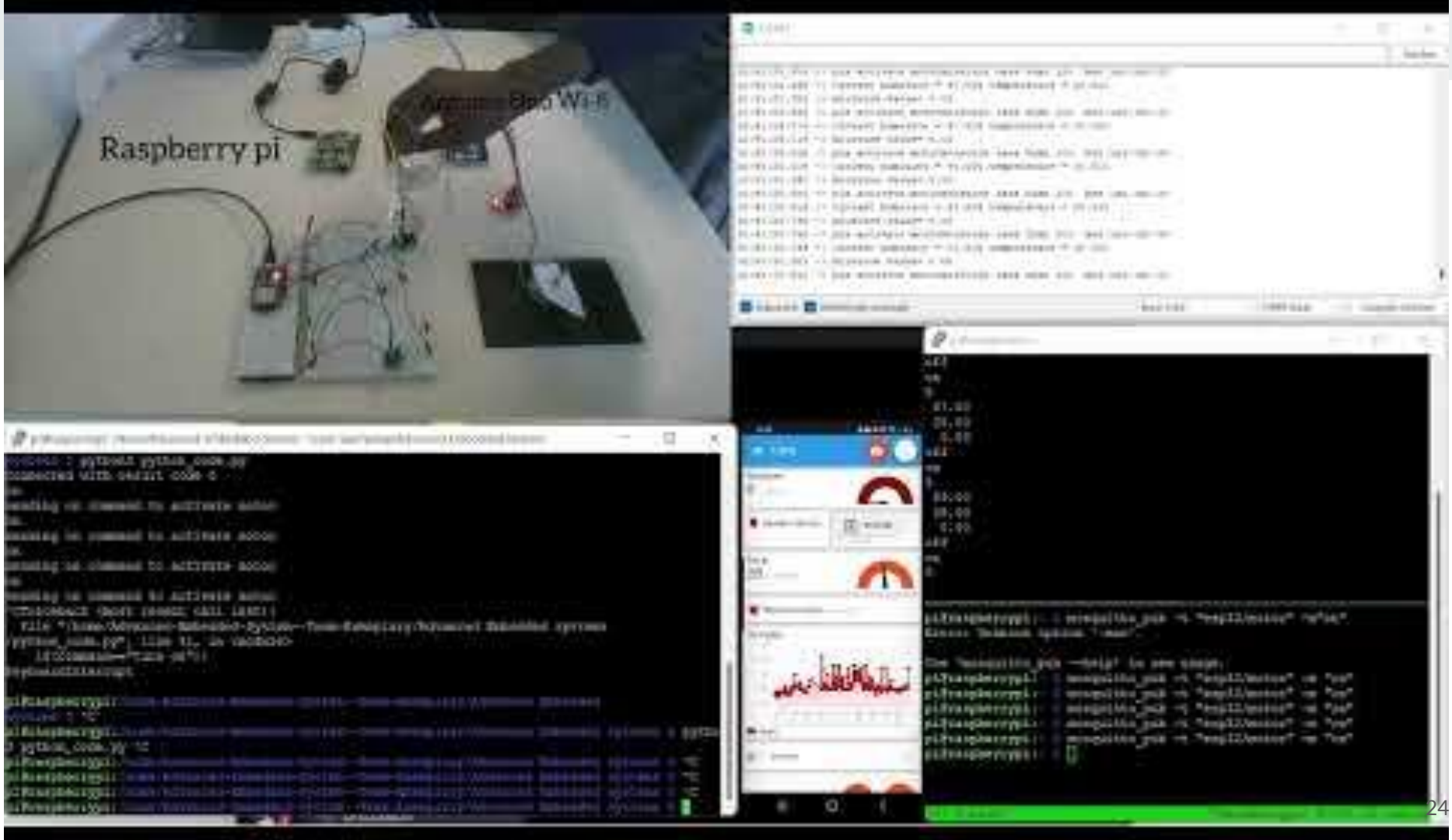
A simple python script is implemented to monitor the changes in the moisture level of the soil.

```
while True:
     if(command=="turn on"):
        client.publish("esp32/motor", "on")
        print("sending on command to activate motor")


def on_message(client, userdata, msg):
    global command
    command = msg.payload.decode()
    print(   command)
    if(command=="on"):
        client.publish("esp32/motor", "on") # automatically turning on the motor when moisture level goes down
        print("sending on command to activate motor")
  #client.disconnect()

client = mqtt.Client()
client.connect("localhost",1883) #localhost refers to the IP of the Rpi itself
```

6. Video Demonstration


Raspberry pi

7. Summary

- How IoT and WSN have been integrated in the domain project.
- How such sensors and actuators technologies have been used in our project.
- How we can describe such a project using diagrams.
- How MQTT communication protocol has been used in our project.
- How we can establish communication between ESP32, Arduino and Raspberry Pi.
- How we use different programming interfaces together for such a project.

References

Github repository- https://github.com/Asm-Nurussafa/Advanced-Embedded-System--Team-Exemplary

[1] https://www.distrelec.de/de/raspberry-pi-modell-gb-ram-raspberry-pi-raspberry-pi-model/p/30085264

[2]https://www.reichelt.de/entwicklerboards-temp-feuchte-dht-11-debo-bo-dht-11-p239086.html?PROVID=2788&gclid=CjwKCAjwtcCVBhA0EiwAT1fY7y57Nj8I8es0CEmqigvr6Wq0yGRKT4GLmyhwdNRq4uctxCmUWw0o7hoCi3sQAvD_BwE

[3]https://www.reichelt.de/arduino-feuchtesensor-fuer-bodenfeuchte-ard-sen-wet1-p282511.html?PROVID=2788&gclid=CjwKCAjwtcCVBhA0EiwAT1fY79TXnUVht3-BB0SvqBkxO_Metz4OtC305AhsqlU8XDhH7D93AqykkhoCj_0QAvD_BwE

[4] https://www.amazon.de/ESP-32-Dev-Kit-V4-Parent/dp/B07Z83MF5W?th=1

[5]https://www.reichelt.de/arduino-uno-wifi-rev2-atmega-4809-arduino-uno-rev2-p248661.html?PROVID=2788&gclid=CjwKCAjwtcCVBhA0EiwAT1fY72fjJJK2zmu35t0cO3TaVqIX53rqmufEVOivmeAEYSIqXZ9DDc4wwRoCFugQAvD_BwE

[6]https://www.reichelt.de/entwicklerboards-schrittmotor-inkl-steuerung-uln2003-debo-moto1-p192146.html?PROVID=2788&gclid=CjwKCAjwtcCVBhA0EiwAT1fY72AfZkL5vcBqoo4m-7V_FgP7vVCvEDnr6O2wXquNZuJG9z2Nv_P_wBoC5iMQAvD_BwE

[7] https://lovepik.com/image-610825885/life-series-hand-drawn-cartoon-mobile-phone-iphone.html

[8] https://www.analog.com/ru/technical-articles/intelligence-at-the-edge-part-3-edge-node-communication.html