Class:	IoT Systems	**Semester:**	Fall 2024
Project topic:	*Plant Monitoring and Alert System*		
Student:	*Yanai Avila*		

1. Project description

a. General idea of your IoT system

My project is an environmental monitoring and alert system designed to ensure optimal plant health in indoor setups. It monitors 2 environmental conditions that are important to the health of a plant: light levels and soil moisture. My system provides real-time feedback though the Blynk interface and it also triggers physical indicators. LEDs and buzzers are used to signal critical conditions locally.

b. System outcomes (profits to the end user)

(1) Real-time Monitoring: Users can monitor environmental conditions from anywhere using the Blynk mobile or web application.
(2) Alerts: Critical conditions like low soil moisture or inadequate light are immediately flagged with visual and audible alarms.
(3) Improved Plant Health: Continuous monitoring helps maintain optimal growth conditions, reducing plant stress and increasing growth.
(4) User-Friendly: Intuitive visualization and control through the LEDs, buzzer and the Blynk platform.

c. Technical description

My system consists of a sensor node that gathers environmental data and send it t another node via MQTT (Message Queuing Telemetry Transport). This node then processes the data, updates it in Blynk, and triggers the physical alerts if thresholds are crossed.

Core Components:

- 2 MKR100s
- Sensors for light and soil moisture
- 2 LEDs
- A buzzer

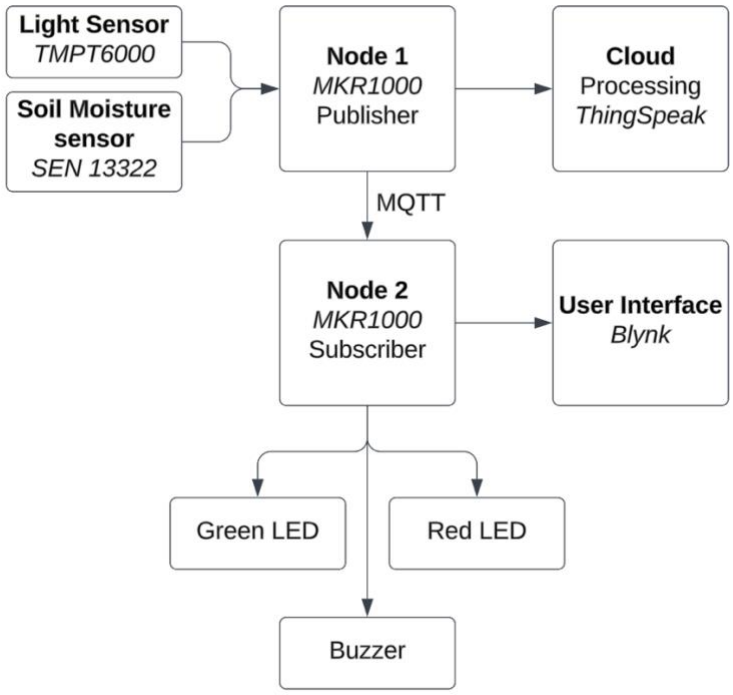
Communication:

- MQTT protocol for data transmission between nodes
- ThingSpeak cloud for average processing
- Blynk interface for remote monitoring and control

Outputs:

- RGB LED indicates conditions:
  - green = optimal
  - red = critical
- A buzzer provides an audible alert for critical scenarios (beeping)

d. Diagrams



Above is the system diagram for my system.

e. List of sensors

My system has 2 sensors: A light sensor and a soil moisture sensor

Vendor	Model	Comment
SparkFun	TMPT6000	Light sensor
SparkFun	SEN 13322	Soil Moisture sensor

f. List of nodes

My system has 2 nodes. Both are Arduino MKR1000s.

Vendor	Model	Comment
Arduino	MKR1000	Node that is the publisher for MQTT
Arduino	MKR1000	Node that is the subscriber for MQTT

g. List of other hardware components

Other components used are a red LED, and green LED, and a buzzer.

Vendor	Model	Comment
Many Vendors	LEDs	1 green LED and 1 red LED
SparkFun	COM-07950	Buzzer

h. Selected cloud service

The cloud service I chose was ThingSpeak. I used ThingSpeak so that the user could see the average light and soil moisture over time. The averages for the last 10 entries are outputted to the output terminal. I used MATLAB analysis for this. I provided screenshots of this later in this report.

2. Project operation

a. Describe how your project works

This system monitors lighting and soil moisture for the plants in a house. It sends reminders/alerts if the plants need water or if it needs more light. Node 1 (Arduino MKR1000) collect the soil moisture sensor and light sensor data. This data is sent to the cloud for processing where the readings are averaged so the user can see. This node acts is the publisher and it sends the information to the client using MQTT. Node 2 (Arduino MKR1000) has 2 LEDs and a buzzer. This node subscribes to MQTT topics related to the plant status. The messages it receives triggers the LEDs and buzzer if the lighting or soil moisture are too low. A green LED indicates that the conditions are optimal. A red LED indicates that one or both readings are critical conditions.

b. Describe what's the data flow in your project

text

i. specify publisher(s): Node 1 (Aduino MKR1000 #1)

ii. specify subscriber(s): Node 2 (Aduino MKR1000 #2)

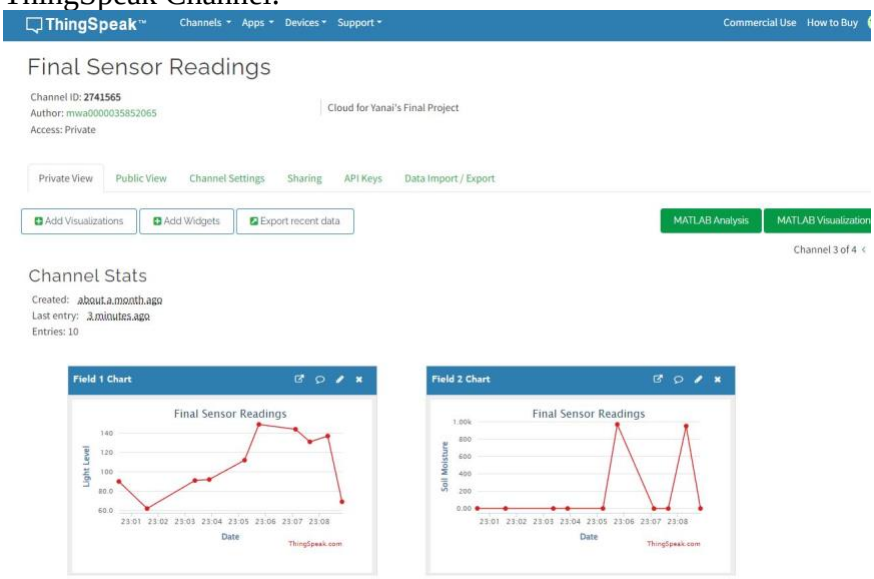
iii. specify the location and type of MQTT broker: My system uses the Mosquitto broker. This broker is located on the machine running the subscriber code.

iv. specify the hierarchy / list of MQTT channels used: The 2 channels are the light level channel and the soil moisture channel.

c. Describe cloud operation, including what's the data processing mechanism

The cloud code finds the averages of the last 10 entries to the ThingSpeak channel. I used MATLAB to be able to do this analysis in Thingspeak.

ThingSpeak Channel:



IoT Systems – Final project

MATLAB Analysis Code and Output:

Name

Final Cloud Code

MATLAB Code

```
% Channel settings
channelID = 2741565; % Channel ID
readAPIKey = '9PDIEA2A1OBL2MBN'; % Read API key

% Fetch data from the last 10 points
data = thingSpeakRead(channelID, 'Fields', [1, 2], 'NumPoints', 10, 'ReadKey', readAPIKey);

% Check if data is retrieved
if isempty(data)
    disp('No data retrieved.');
else
    % Calculate averages
    avgLightLevel = mean(data(:, 1), 'omitnan');
    avgSoilMoisture = mean(data(:, 2), 'omitnan');

    % Output averages to terminal
    fprintf('Average Light Level: %.2f\n', avgLightLevel);
    fprintf('Average Soil Moisture: %.2f\n', avgSoilMoisture);
end
```

Save and Run | Save

Output

```
Average Light Level: 107.70
Average Soil Moisture: 192.60
```

3. Screenshots / pictures

a. Publisher(s) operation (data being sent to MQTT broker)

MQTT_publisher.ino

```
1   #include <ArduinoMqttClient.h>
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino MKR1000' on 'COM6')

```
22:10:27.251 -> Sending message to topic: light_level_topic
22:10:27.251 -> 113.00
22:10:27.251 -> Sending message to topic: soil_moist_topic
22:10:27.251 -> 944.00
22:10:27.251 ->
22:10:35.259 -> Sending message to topic: light_level_topic
22:10:35.259 -> 52.00
22:10:35.259 -> Sending message to topic: soil_moist_topic
22:10:35.259 -> 1.00
22:10:35.259 ->
22:10:43.263 -> Sending message to topic: light_level_topic
22:10:43.263 -> 123.00
22:10:43.263 -> Sending message to topic: soil_moist_topic
22:10:43.263 -> 0.00
22:10:43.263 ->
22:10:51.253 -> Sending message to topic: light_level_topic
22:10:51.253 -> 62.00
22:10:51.253 -> Sending message to topic: soil_moist_topic
22:10:51.253 -> 1.00
22:10:51.253 ->
22:10:59.219 -> Sending message to topic: light_level_topic
22:10:59.219 -> 123.00
22:10:59.219 -> Sending message to topic: soil_moist_topic
22:10:59.265 -> 0.00
22:10:59.265 ->
22:11:07.217 -> Sending message to topic: light_level_topic
22:11:07.217 -> 65.00
22:11:07.264 -> Sending message to topic: soil_moist_topic
22:11:07.264 -> 0.00
22:11:07.264 ->
22:11:15.239 -> Sending message to topic: light_level_topic
22:11:15.239 -> 123.00
22:11:15.239 -> Sending message to topic: soil_moist_topic
22:11:15.239 -> 2.00
22:11:15.239 ->
```

b. Subscriber(s) operation (data being received from MQTT broker)

mqtt_client.ino

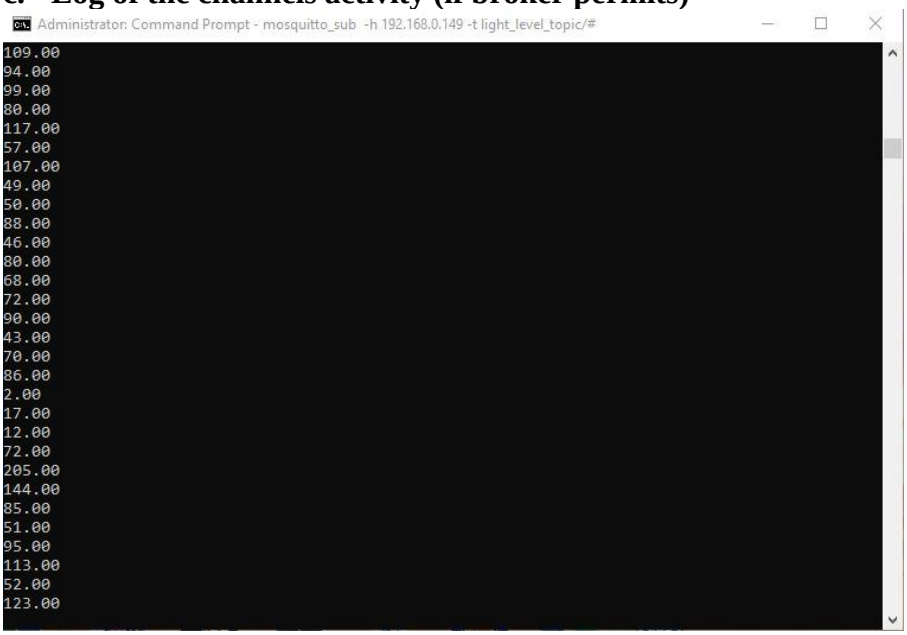
```
4
5  #include <ArduinoMqttClient.h>
```

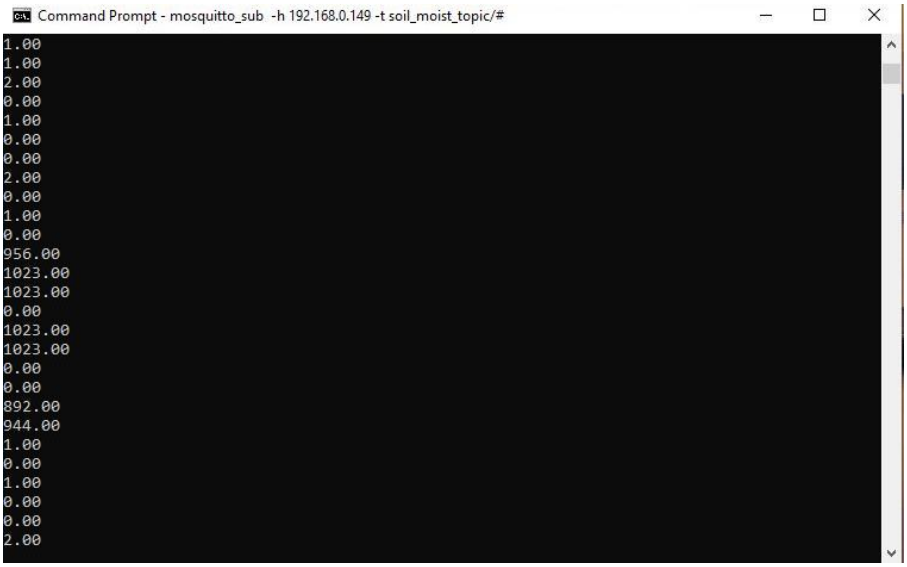
Output Serial Monitor ×

Message (Enter to send message to 'Arduino MKR 1000 WiFi' on 'COM8')

```
Received message on topic 'soil_moist_topic': 2.00
Received message on topic 'light_level_topic': 46.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 80.00
Received message on topic 'soil_moist_topic': 1.00
Received message on topic 'light_level_topic': 68.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 72.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 90.00
Received message on topic 'soil_moist_topic': 2.00
Received message on topic 'light_level_topic': 43.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 70.00
Received message on topic 'soil_moist_topic': 1.00
Received message on topic 'light_level_topic': 86.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 2.00
Received message on topic 'soil_moist_topic': 956.00
Received message on topic 'light_level_topic': 17.00
Received message on topic 'soil_moist_topic': 1023.00
Received message on topic 'light_level_topic': 12.00
Received message on topic 'soil_moist_topic': 1023.00
Received message on topic 'light_level_topic': 72.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 205.00
Received message on topic 'soil_moist_topic': 1023.00
Received message on topic 'light_level_topic': 144.00
Received message on topic 'soil_moist_topic': 1023.00
Received message on topic 'light_level_topic': 85.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 51.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 95.00
Received message on topic 'soil_moist_topic': 892.00
Received message on topic 'light_level_topic': 113.00
Received message on topic 'soil_moist_topic': 944.00
Received message on topic 'light_level_topic': 52.00
Received message on topic 'soil_moist_topic': 1.00
Received message on topic 'light_level_topic': 123.00
Received message on topic 'soil_moist_topic': 0.00
Received message on topic 'light_level_topic': 62.00
Received message on topic 'soil_moist_topic': 1.00
```

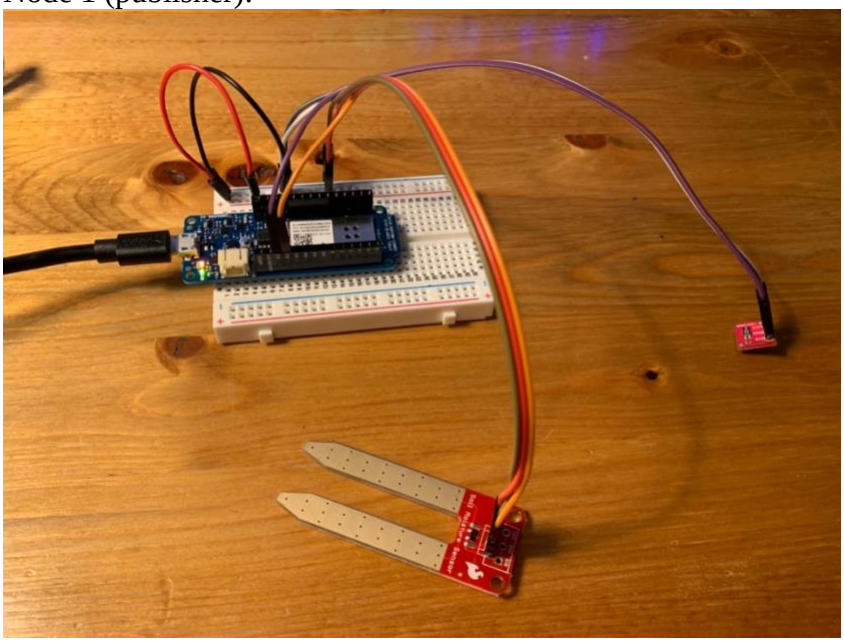
c. Log of the channels activity (if broker permits)



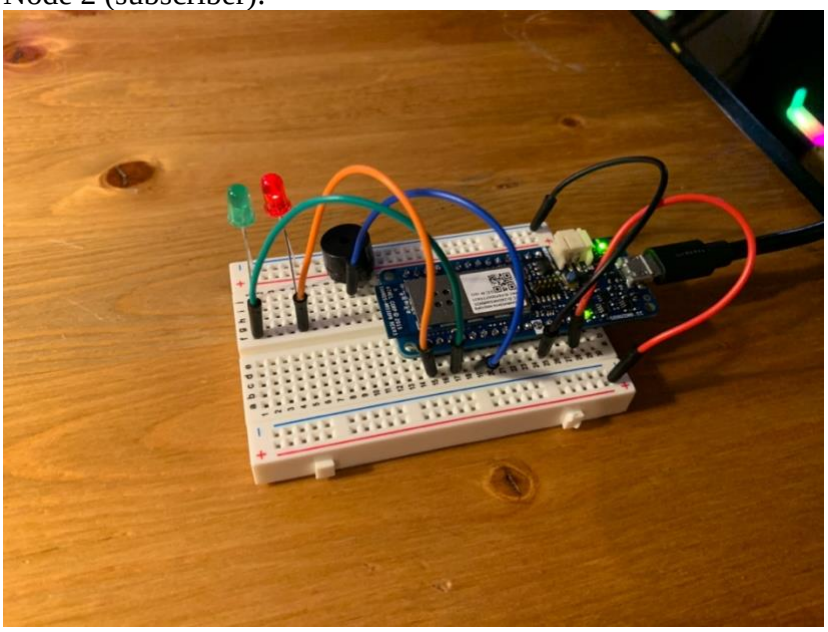


d. Picture(s) of wired circuit

Node 1 (publisher):



Node 2 (subscriber):



4. Describe technical challenges you had and how you solved them

The main challenge I faced was that my code stopped working at one point. Everything worked fine up until this point, so it took me a long time to figure out what was wrong. At first I thought it was my broker, but I used a broker on another device and it still did not fix it. At this point I knew the problem was not with my subscriber code because everything worked fine before, so I looked at my publisher code. I realized that the latest changes I made to my publisher code was add code for ThingSpeak. My code worked as before when I took this code out, so that allowed me to fix the ThingSpeak code to be able to work with the MQTT code.

5. Code listings for each node and cloud codes

- MQTT publisher code (Node 1)
- MQTT subscriber code (Node 2)
- ThingSpeak code for data processing

YouTube Link for video demo: https://youtu.be/GbuoHMk6XGo