Internet of Things (IoT)



**Student Id:** 21110214

**Submitted to:** Dr. Murad Yaghi

**Submitted by:** Saif Kamal Salim Haddad

**Submitted on:** 27-1-2025

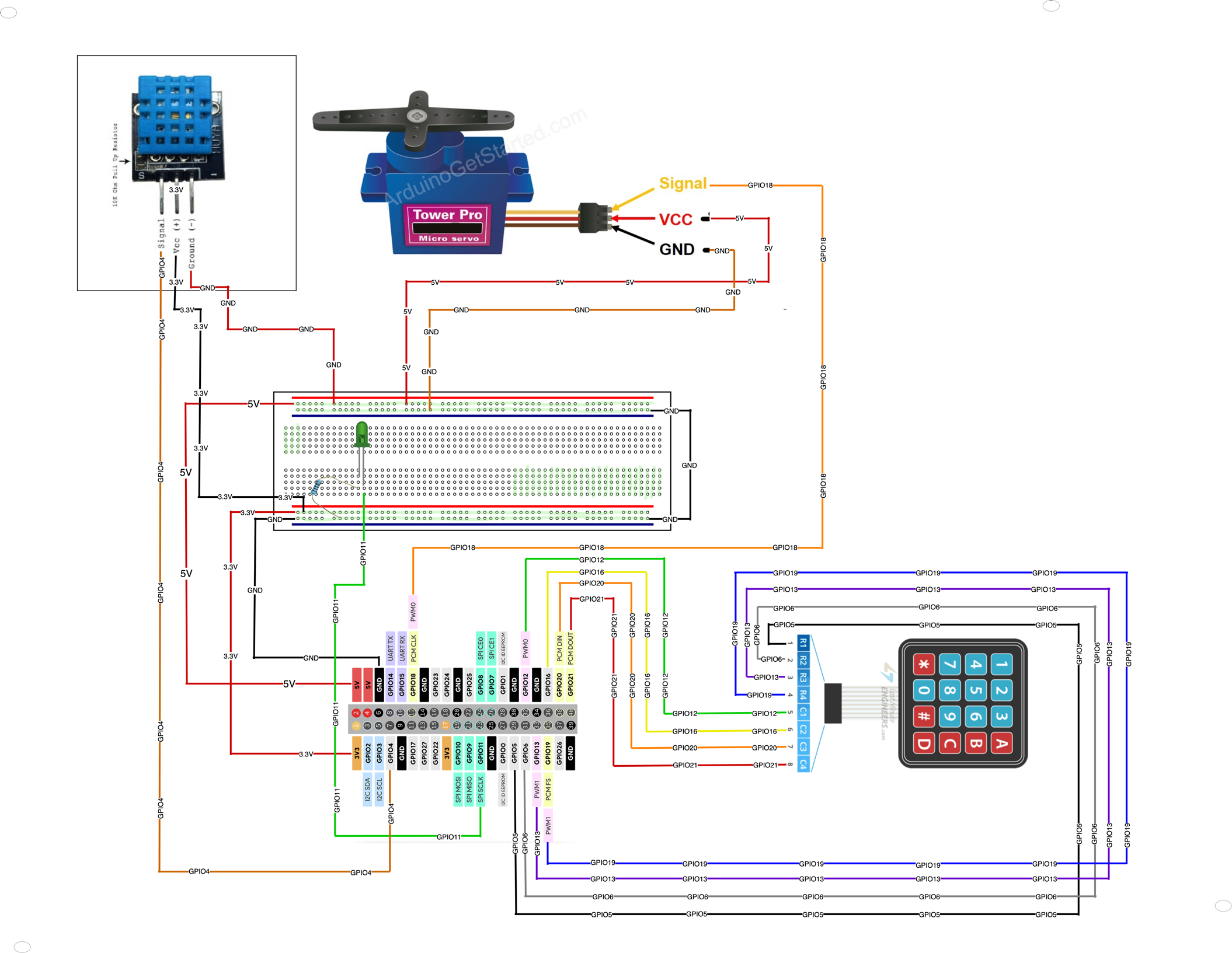
Table of Contents

[Task 1: My Schematic Diagram For All The Connections. 3](#_Toc188913741)

[Task 3.2: How To Use MQTT Without Internet: 4](#_Toc188913742)

[Task 3.3: steps to enable this functionality: 5](#_Toc188913743)

# Task 1: My Schematic Diagram For All The Connections.



# Task 3.2: How To Use MQTT Without Internet:

1. **Use a Local MQTT Broker:**

This system is considered one of the simplest and the most efficient methods that could be used in case there is no internet, and we want to enable the MQTT to communicate without the internet, all this can be achieved just by making a Local MQTT Broker. Using this method will allow all the devices that are in the greenhouses to communicate with each other just over a private network by using LANs without the need for the internet. This broker will act as the central hub for all MQTT communications. This system uses static IP values for all the devices to ensure strong communication within the private network

**2-Use Mesh Networks:**

This system could be used in situations where the GreenHouses are spread too far from each other even if they were in different regains, but using a mesh network enables a decentralized system where devices communicate directly with one another, and data can hop between devices until it reaches its destination. But from the disadvantages of this system if one of the devices drops the network will drop also. LoRa, ZigBee, and Thread are just a few examples of technologies over which mesh networks can be established. They allow for long-range communication, and the possibility of going through obstacles like walls or plants makes them quite suitable for greenhouses. Every device in a mesh network acts both as a data point and a repeater to ensure robustness in communication.

**3- Use Edge Computing:**

Also, this system could be considered one of the most efficient systems to use because this system uses devices such as Rbi that act like an edge server rather than using a cloud-based internet. This device can collect data and then do the processing and analyze the data from the greenhouse devices. One of the best advantages of edge computing is that this system can respond to the real-time condition of the devices of the Greenhouses without the need of the internet. All the data that are collected in the greenhouses can be processed locally, and also the messages that are from the MQTT can be sent to all the greenhouse devices without internet, and all the data that was collected can be stored locally, which enhances the privacy and the security of the GreenHouses data.

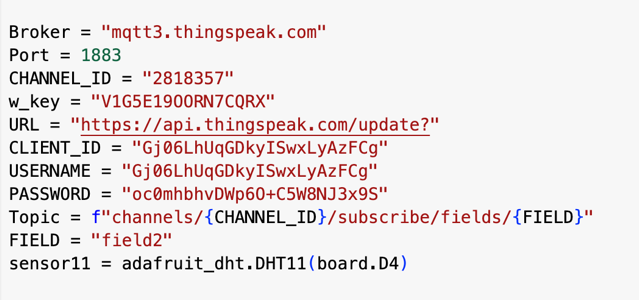
**4-Use Ad-Hoc Wi-Fi Networks:**

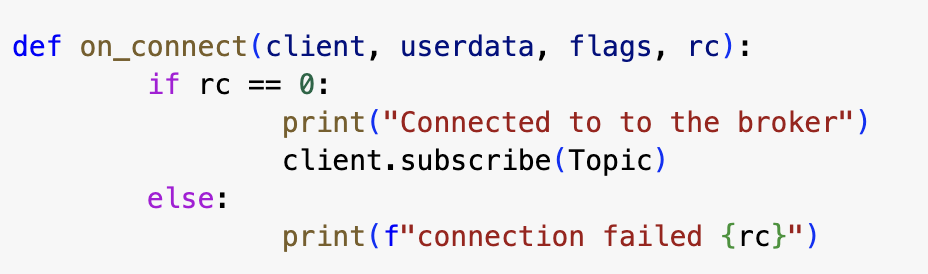
This system is another way to enable the MQTT to communicate without an internet signal. This system enables the devices to send messages and communicate with each other without the need for a centralized router internet or an edge device, the devices communicate directly to each other. Using this system, which could be described as peer to peer communication protocol enables the MQTT to send messages between the GreenHouses or between the all the devices in each GreenHouse without internet.

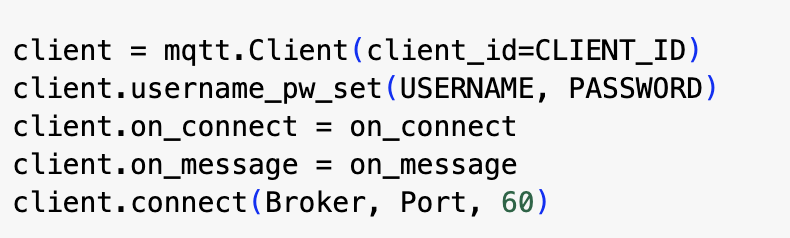
# Task 3.3: steps to enable this functionality:

**Step 1: Setting Up MQTT for Communication:**

At the beginning, is started by configuring the MQTT client for communication with the broker and initializing everything. I started initializing the Broker address and port, I used mqtt3.thingspeak.com as the broker and the port number was 1883. By using the paho.mqtt.client library, I initialized the MQTT and also the (client ID, username, and password)which were taken from creating an MQTT device, this step was done inside the ThinkSpeak, and subscription topics such as channel ID and field ID, all of them was taken from the ThinkSpeak website from my chanal. This ensured the Raspberry Pi could connect to the broker and subscribe to changes in the humidity field; this was shown in my code by creating the on\_connect() callback to subscribe to the topic and log connection status, enabling real-time updates from ThingSpeak.







**Step 2: Reading Data from the DHT11 Sensor**

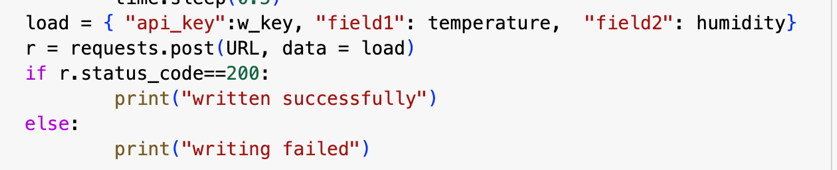
Then I decided to connect it with DHT11, which can keep track of data regarding the current humidity and temperature. Using an adafruit\_dht library, setup the sensor to read on the temperature, humidity data on GPIO pin 4 on the Raspberry Pi. Sensor\_write() provides a way for recording those readings, gracefully handling any type of runtime failures, and then preparing this data for the upload onto the ThingSpeak channel. I followed it with sending the sensor data to the server using the ThingSpeak API key. This step involved formatting the data into a POST request to ensure that it was MQTT accessible and refreshed.



**Step 3: Publishing Data to ThingSpeak**

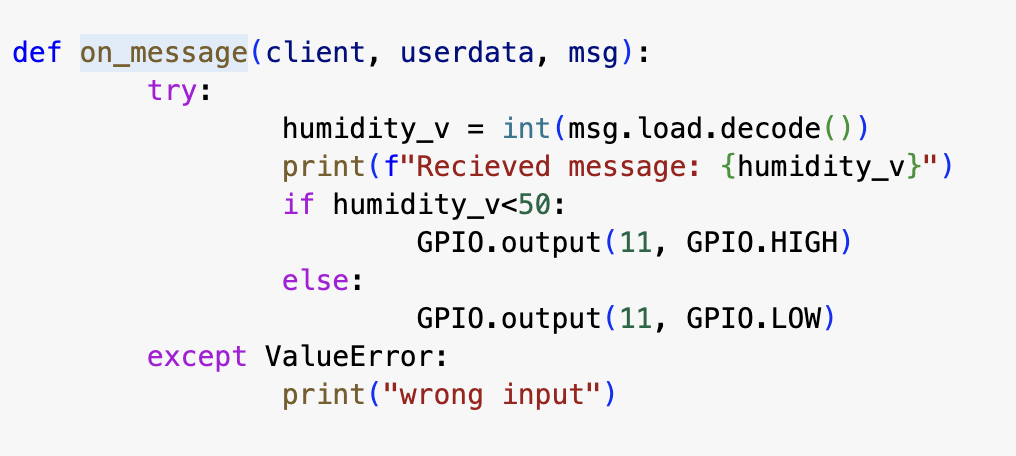
Then the data had to be uploaded to ThingSpeak, for which the URL of the ThingSpeak API was created, with the payload of the data to be ready to send with sensor values and an API key. Then, it delivers this payload to the server using requests.post(). Logs from the solution were meant to show whether the update was successful or some faults occurred, along with status checks that will confirm each upload of data has been executed. This step was very important to my project in ensuring that ThingSpeak can process and store all environmental data in real time.

\



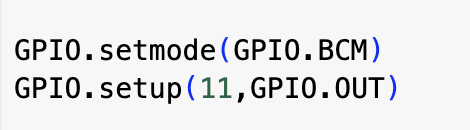
**Step 4: Subscribing to Data and Controlling GPIO**

This ste was made to enable dynamic Led contrul based on the humidity level by subscribe to the topic provided values ThingSpeak. The on\_message() callback was used to extract the value of humidity and decode the incoming messages. This value, after extraction, was compared with a threshold value, which in my project is 50%. Operating an LED: To accomplish this, GPIO pin 11 is toggled to HIGH or LOW depending on the value. In this sense, real-time actuation based on sensor readings may now provide immediate input regarding one's environment.



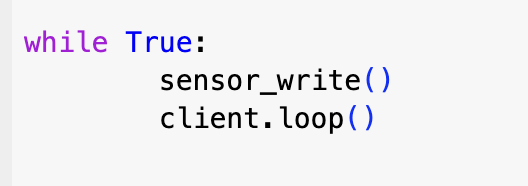
**Step 5: Configuring GPIO**

Then the setup of the GPIO pins by of the RPi.GPIO library. In this, to make sure the LED remained off until the conditions of the humidity turned it on, GPIO pin 11 was set as an output pin and to its initial state of LOW. To make sure that this step of the GPIO configuration was complete before entering the main loop, it was an early addition in the code



**Step 6: Continuous Operation in the Main Loop:**

The last step was to create an endless loop that could handle the incoming messages through MQTT, post data on ThingSpeak, and monitor the sensor. In order for there to be a smooth flow of data and actual time communication, the while True method integrated the sensor\_write() function along with the client.loop() function. This step was representative of the fact that the system had to work in a continuum without any human interference.



References:

<https://www.researchgate.net/publication/329380987_A_comprehensive_study_on_MQTT_as_a_low_power_protocol_for_internet_of_things_application>

<https://mqtizer.com/blog/can-iot-work-without-the-internet/>

https://www.hashstudioz.com/blog/connecting-internet-of-things-iot-with-mqtt/#:~:text=Yes%2C%20MQTT%20may%20work%20without,ethernet%20cable%20make%20a%20network.