10422030

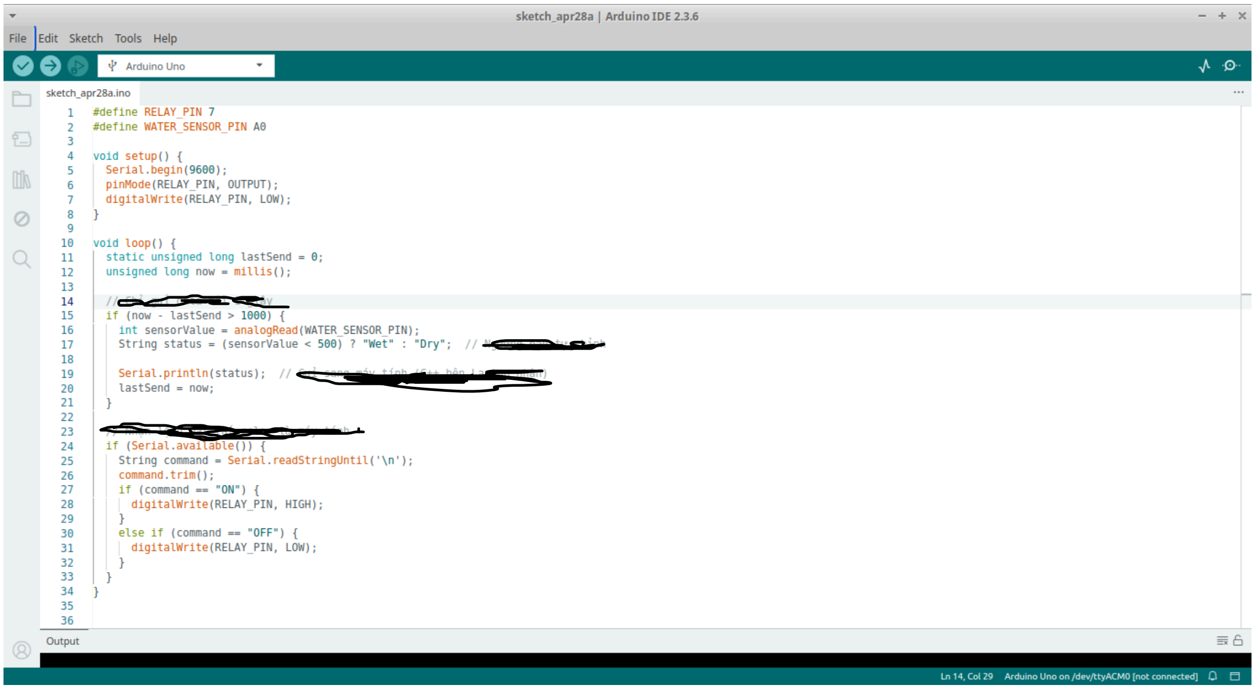
Over the last 6 lessons you have learned about digital agriculture and how to use IoT devices to gather data to predict plant growth, and automate watering based off soil moisture readings.

Use what you have learned to build a new IoT device using a sensor and actuator or your choice. Send telemetry to an IoT Hub, and use that to control an actuator via serverless code. You can use a sensor and an actuator you have already used in this or the previous project, or if you have other hardware try something new.

Answer

In this exercise, I will build an IoT system using "Water Sensor" and 5V Relay. Data from the water sensor will be sent to the internet via MQTT to Adafruit IO and from Adafruit IO send data back to control 5V Relay to turn on or off.

1. First we need to write code to control the Water sensor and 5V Relay via the Arduino IDE.

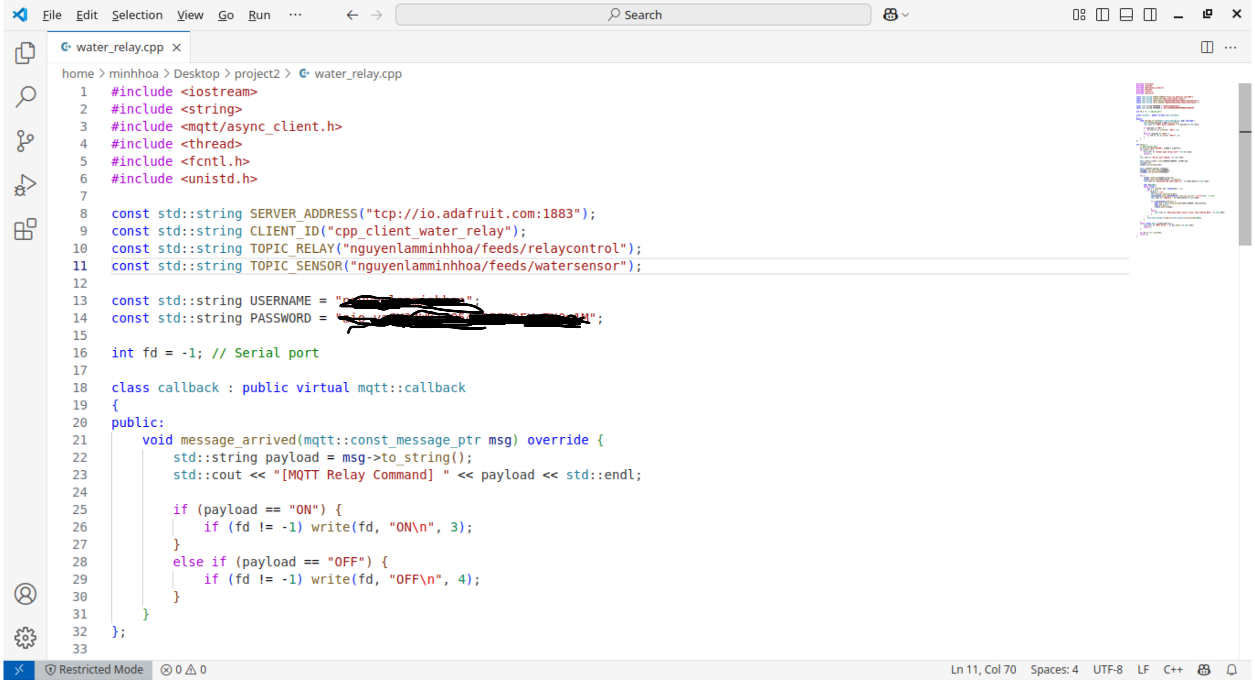

sketch_apr28a | Arduino IDE 2.3.6
File
Edit
Sketch
Tools
Help
Arduino Uno
sketch_apr28a.ino
Output
Ln 14, Col 29
Arduino Uno on /dev/ttyACM0 [not connected]

```
#define RELAY_PIN 7
#define WATER_SENSOR_PIN A0

void setup() {
  Serial.begin(9600);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);
}

void loop() {
  static unsigned long lastSend = 0;
  unsigned long now = millis();

  // [redacted]
  if (now - lastSend > 1000) {
    int sensorValue = analogRead(WATER_SENSOR_PIN);
    String status = (sensorValue < 500) ? "Wet" : "Dry";  // [redacted]

    Serial.println(status);  // [redacted]
    lastSend = now;
  }

  [redacted]
  if (Serial.available()) {
    String command = Serial.readStringUntil('\n');
    command.trim();
    if (command == "ON") {
      digitalWrite(RELAY_PIN, HIGH);
    }
    else if (command == "OFF") {
      digitalWrite(RELAY_PIN, LOW);
    }
  }
}
```

The code shows the value of the Water sensor will be wet or dry based on the water threshold that the Water sensor measures. The Relay 5V will receive the value to control from Adafruit via PC.

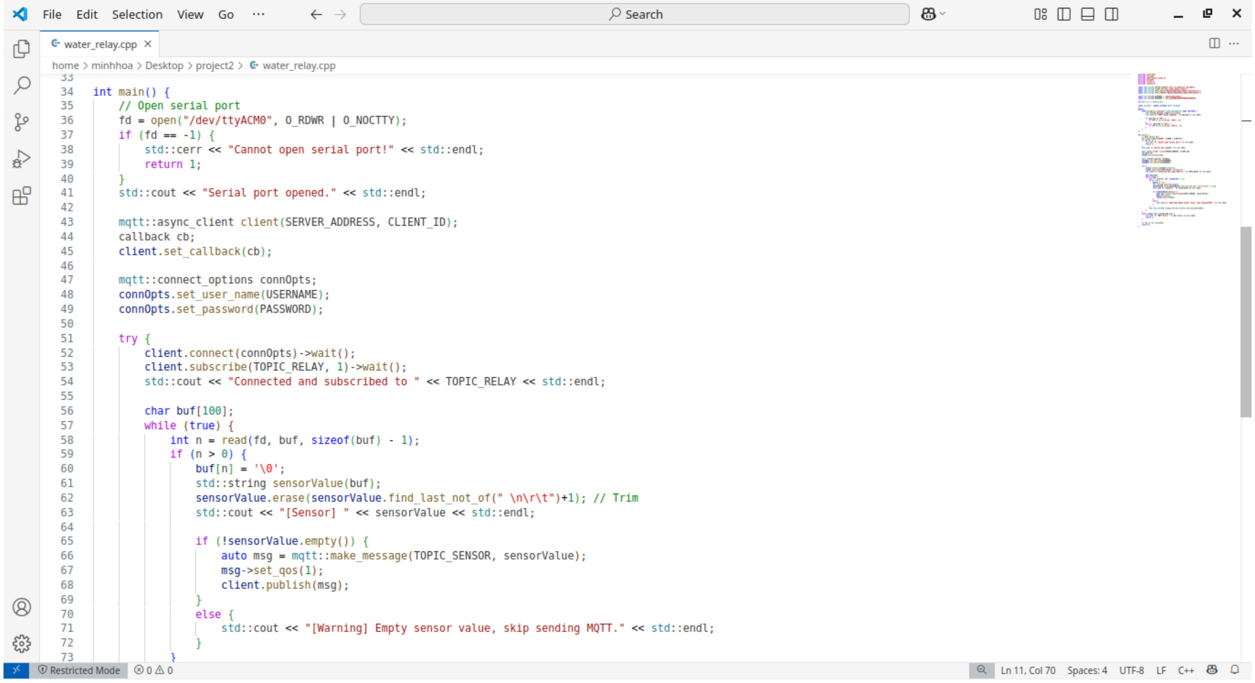
2. Write code to make connection between Adafruit IO and PC

To be able to connect to Adafruit IO, we need to install some necessary libraries, and also need to define so that the sensor data can work properly.



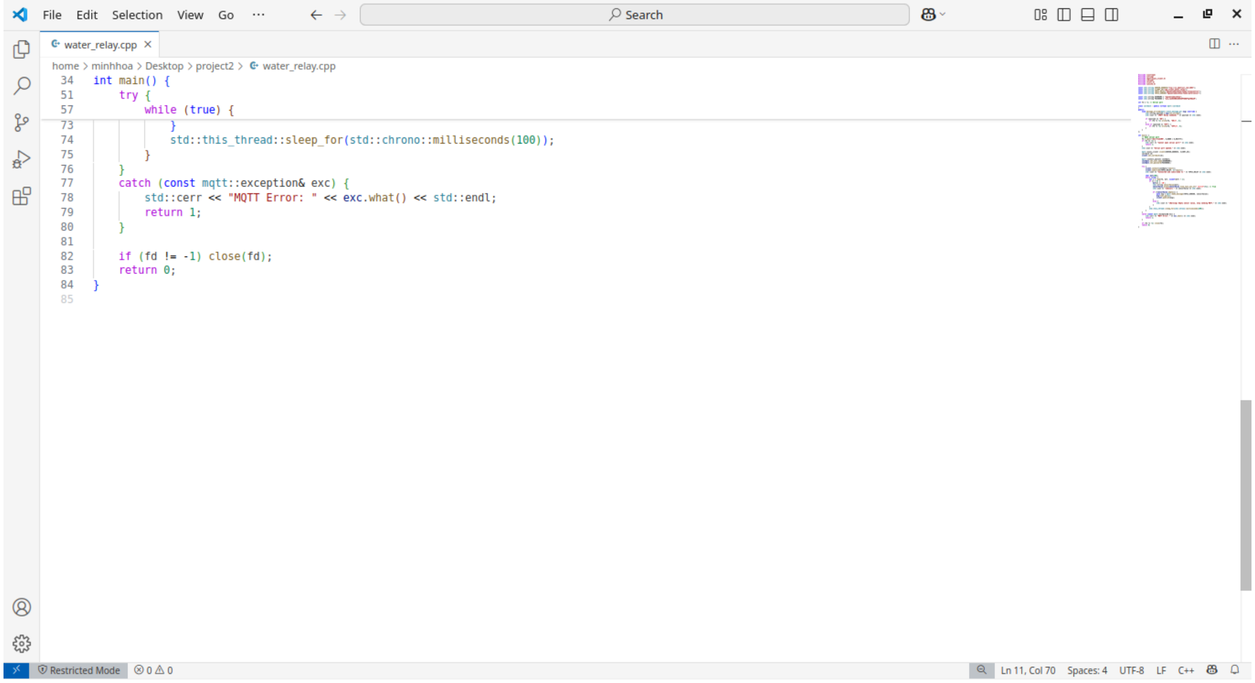
```cpp
#include <iostream>
#include <string>
#include <mqtt/async_client.h>
#include <thread>
#include <fcntl.h>
#include <unistd.h>

const std::string SERVER_ADDRESS("tcp://io.adafruit.com:1883");
const std::string CLIENT_ID("cpp_client_water_relay");
const std::string TOPIC_RELAY("nguyenlamminhhoa/feeds/relaycontrol");
const std::string TOPIC_SENSOR("nguyenlamminhhoa/feeds/watersensor");

const std::string USERNAME = [redacted]";
const std::string PASSWORD = [redacted]";

int fd = -1; // Serial port

class callback : public virtual mqtt::callback
{
public:
    void message_arrived(mqtt::const_message_ptr msg) override {
        std::string payload = msg->to_string();
        std::cout << "[MQTT Relay Command] " << payload << std::endl;

        if (payload == "ON") {
            if (fd != -1) write(fd, "ON\n", 3);
        }
        else if (payload == "OFF") {
            if (fd != -1) write(fd, "OFF\n", 4);
        }
    }
};
```

The above code includes the definitions needed for MQTT and Adafruit, information about Feeds on the Adafruit destination account, and the username with the AIO key.



```cpp
int main() {
    // Open serial port
    fd = open("/dev/ttyACM0", O_RDWR | O_NOCTTY);
    if (fd == -1) {
        std::cerr << "Cannot open serial port!" << std::endl;
        return 1;
    }
    std::cout << "Serial port opened." << std::endl;

    mqtt::async_client client(SERVER_ADDRESS, CLIENT_ID);
    callback cb;
    client.set_callback(cb);

    mqtt::connect_options connOpts;
    connOpts.set_user_name(USERNAME);
    connOpts.set_password(PASSWORD);

    try {
        client.connect(connOpts)->wait();
        client.subscribe(TOPIC_RELAY, 1)->wait();
        std::cout << "Connected and subscribed to " << TOPIC_RELAY << std::endl;

        char buf[100];
        while (true) {
            int n = read(fd, buf, sizeof(buf) - 1);
            if (n > 0) {
                buf[n] = '\0';
                std::string sensorValue(buf);
                sensorValue.erase(sensorValue.find_last_not_of(" \n\r\t")+1); // Trim
                std::cout << "[Sensor] " << sensorValue << std::endl;

                if (!sensorValue.empty()) {
                    auto msg = mqtt::make_message(TOPIC_SENSOR, sensorValue);
                    msg->set_qos(1);
                    client.publish(msg);
                }
                else {
                    std::cout << "[Warning] Empty sensor value, skip sending MQTT." << std::endl;
                }
            }
```

The main() function defines the PC's connection port to the Arduino Uno, and also manages the connection between the Adafruit IO and the PC via the internet.

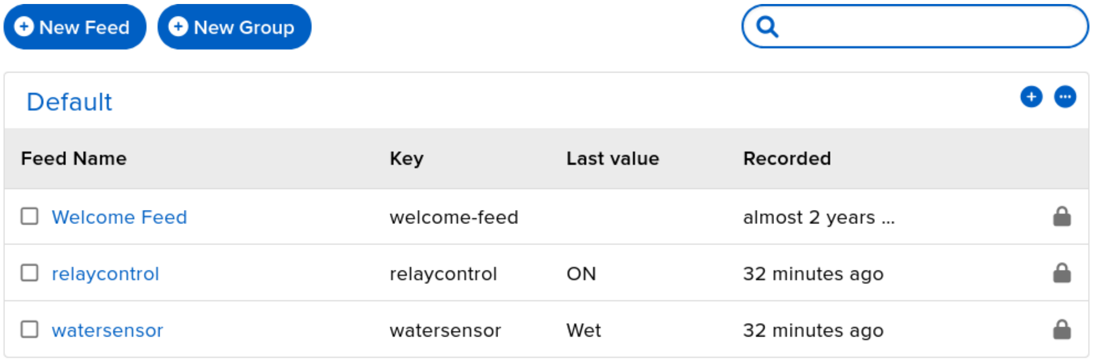

File Edit Selection View Go ···
Search
water_relay.cpp
home > minhhoa > Desktop > project2 > water_relay.cpp
Restricted Mode ⊗ 0 ⚠ 0
Ln 11, Col 70 Spaces: 4 UTF-8 LF C++

```cpp
int main() {
    try {
        while (true) {
            }
            std::this_thread::sleep_for(std::chrono::milliseconds(100));
        }
    }
    catch (const mqtt::exception& exc) {
        std::cerr << "MQTT Error: " << exc.what() << std::endl;
        return 1;
    }

    if (fd != -1) close(fd);
    return 0;
}
```

The last group of functions manages errors when the connection fails and notifies the user via Terminal.

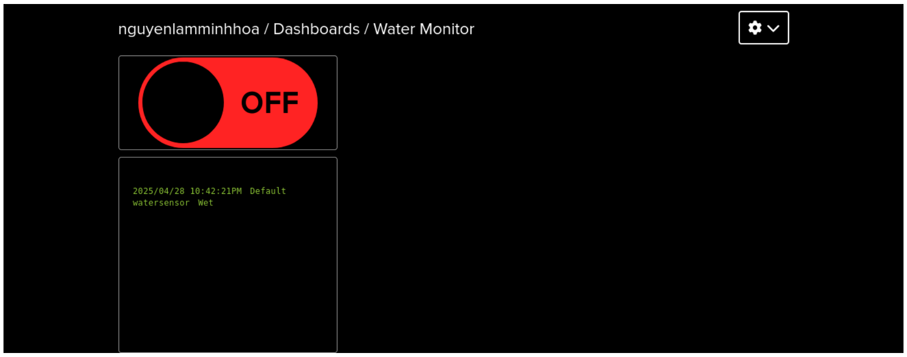
3. Adafruit IO Setup

First we will create 2 Feeds, “watersensor” and “relaycontrol”, to receive data from the sensor sent from Arduino.



Feed Name	Key	Last value	Recorded
□ Welcome Feed	welcome-feed		almost 2 years ...
□ relaycontrol	relaycontrol	ON	32 minutes ago
□ watersensor	watersensor	Wet	32 minutes ago

Then design a Dashboard to be able to display data on Adafruit.

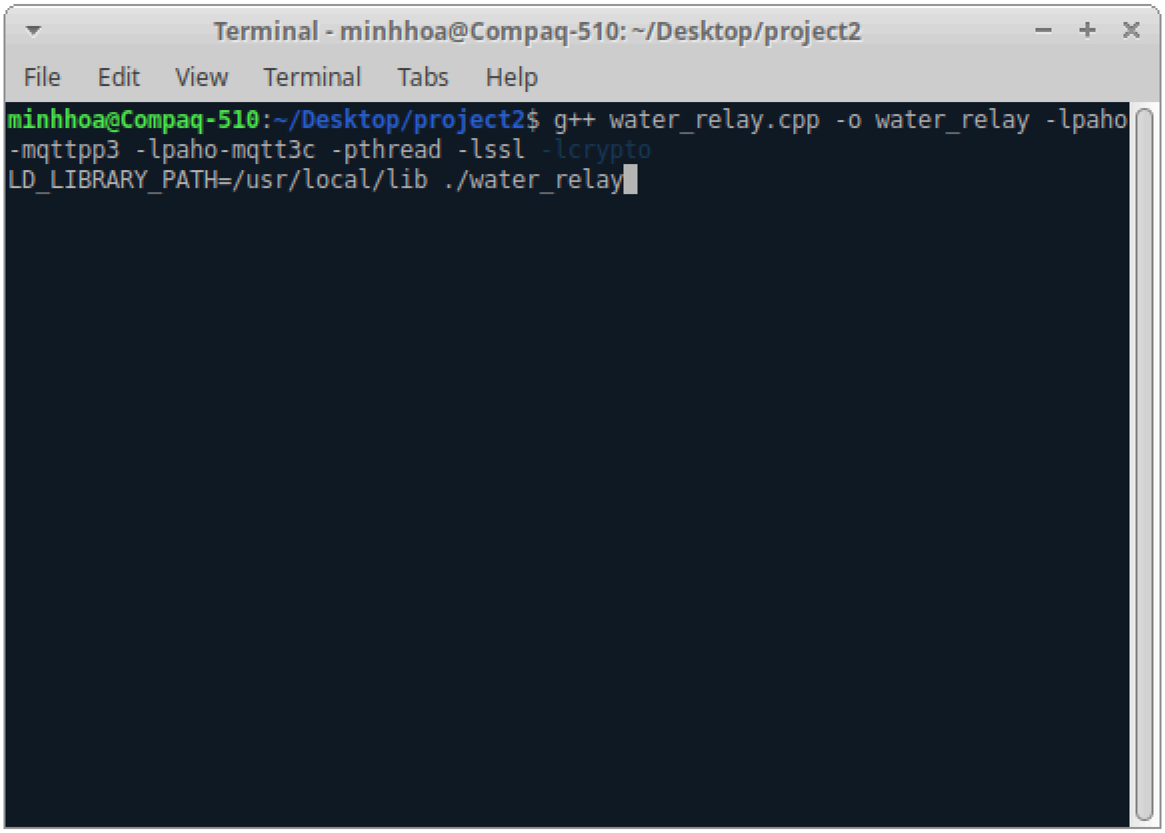


```
2025/04/28 10:42:21PM Default
watersensor Wet
```

4. Test

We connect the 5V Relay and Water sensor to Arduino Uno. The Arduino Uno's connection port is connected to the PC via USB.

Then we open Terminal and use the command to start running the program.



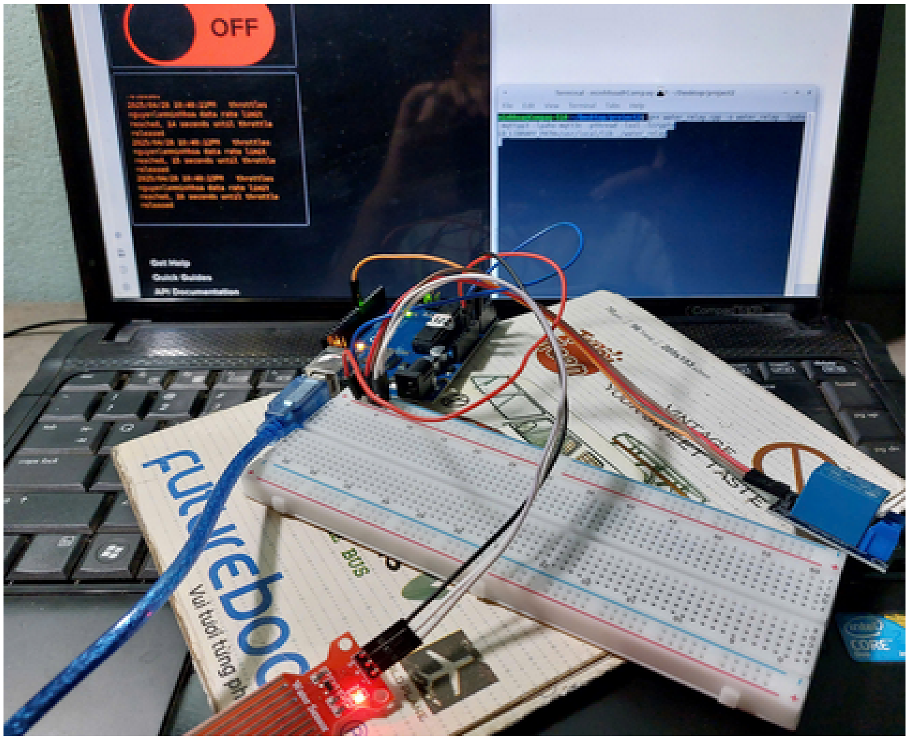
Terminal - minhhoa@Compaq-510: ~/Desktop/project2
File Edit View Terminal Tabs Help

```
minhhoa@Compaq-510:~/Desktop/project2$ g++ water_relay.cpp -o water_relay -lpaho
-mqttpp3 -lpaho-mqtt3c -pthread -lssl -lcrypto
LD_LIBRARY_PATH=/usr/local/lib ./water_relay
```

When the program connects successfully, the Adafruit IO Dashboard will display the status information of the measured humidity.

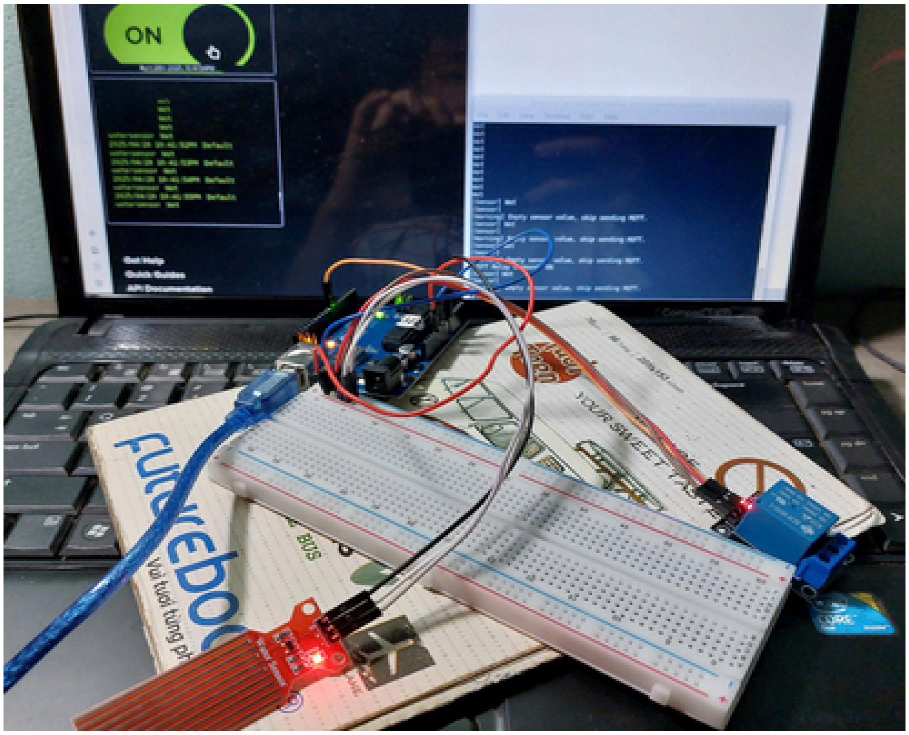
A relay that is not lit indicates that it is not working and is similar to the Relay switch on the Adafruit IO being OFF.

A red light on the water sensor indicates that it is working.



After turning on the "Relay 5V" control ON button, the Relay will light up red, indicating that it has received the command via MQTT and started working.

The Water sensor status display board also shows information indicating that data from the Water sensor has been sent to Adafruit via MQTT.



6. Conclusion

The 5V Relay and Water sensor are working fine. The data sent to Adafruit IO via MQTT has been successful and recorded the necessary information. The data sent down from Adafruit IO by turning on and off the "5V Relay" is also completed. The IoT device used to measure water humidity and turn on and off the Relay is working fine via the internet with MQTT.