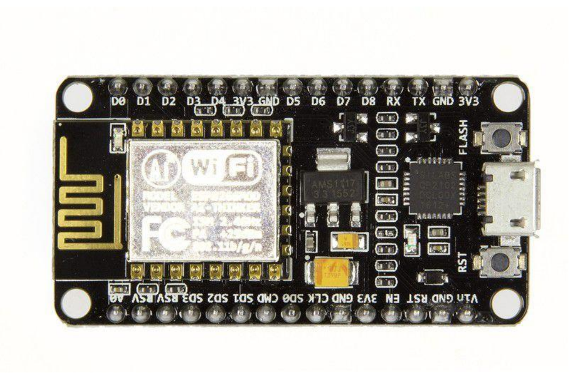
AIM:

To create a self-watering plant system using a soil sensor, water pump and nodemcu 8266.

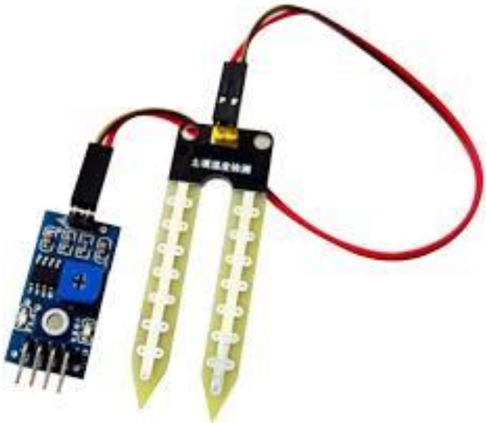
Components Used:

1. nodemcu8266



2. Water pump(3-6v)

3. Soil sensor



4. 2-channel Relay Module

5. Breadboard

6.Jumper wires

Soil sensor module has the following pins:

The sensor itself has 2 pins to transmit data to the LM393 module which has 4 pins:

- VCC: It supplies power for the module. It is around 205 to 3.3 volts.
- A0: analog output
- D0:Digital Output
- GND: It is the Ground Pin and needs to be connected to the GND pin on the NodeMCU.

Relay module has:

Control Pins:

VCC pin supplies power to the built-in optocoupler and optionally to the electromagnet of the relay (if you keep the jumper in place)

GND is the common Ground connection.

IN1 & IN2 pins are used to control the relay. These are active low pins, meaning the relay will be activated when you pull the pin LOW and it will become inactive when you pull the pin HIGH.

Power Supply Selection Pins:

JD-VCC supplies power to the electromagnet of the relay. When the jumper is in place, it takes power from the Arduino’s 5V line. Without the jumper cap, you have to connect it to an independent power source.

VCC With the jumper cap on, this pin is shorted to the JD-VCC pin. If you remove the jumper, keep this pin unconnected.

GND is the common Ground connection.

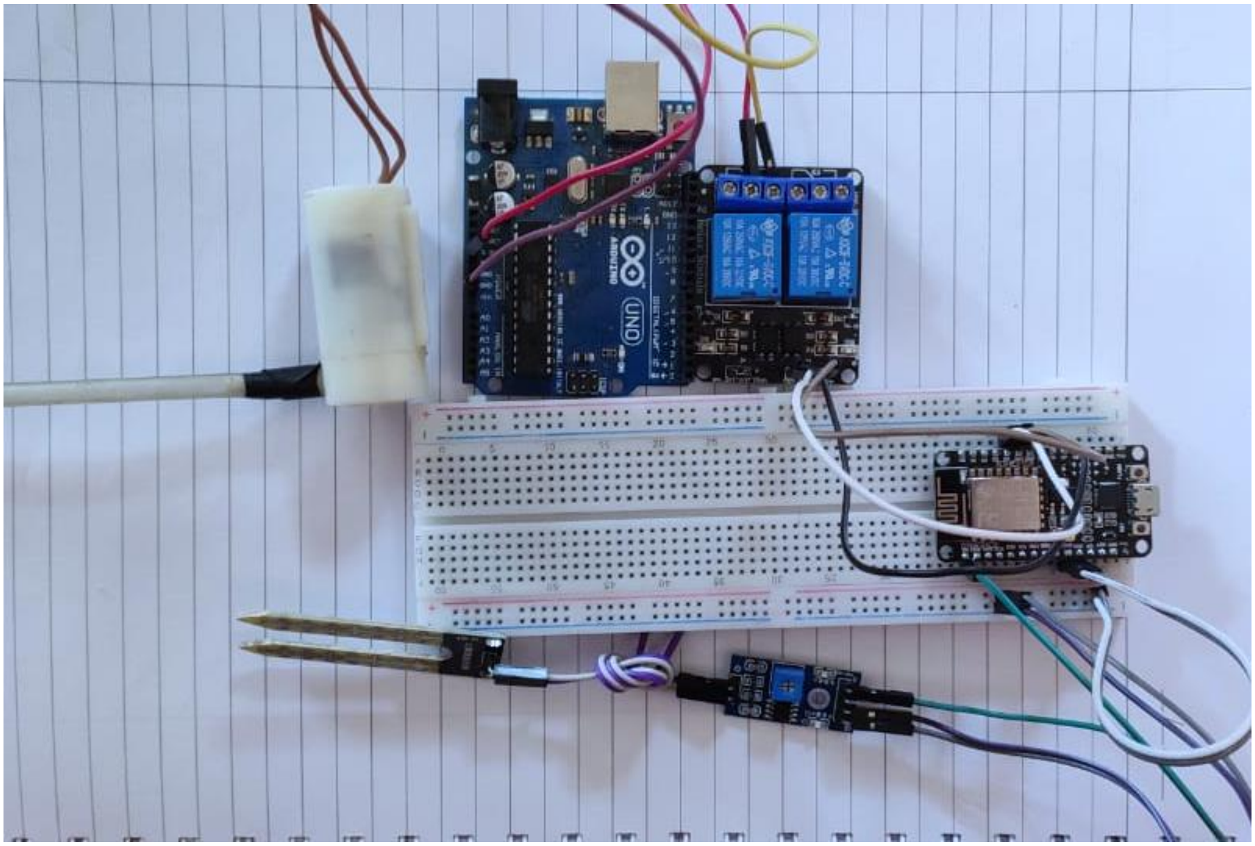
Output Terminals:

COM pin is connected to the signal you are planning to switch.
NC pin is connected to the COM pin by default, unless you send a signal from the Arduino to the relay module to break the connection.
NO pin is open by default, unless you send a signal from the Arduino to the relay module to make the connection.

Software Used:

- Arduino Ide (for coding)
- Adafruit.io (Cloud platform to store data)
- IFTTT

Connections:



The connections to nodemcu are as follows:

1. IN1 pin Is connected to D4 pin in nodeMCU
2. GND pin is connected to GND pin in nodeMCU
3. Vcc pin is connected to 3v3 pin in nodeMCU
4. COM pin is connected to a 3.3v output
5. NO pin is connected to anode of water pump
6. COM pin is connected to cathode of the water pump
7. A0 pin is connected to the A0 pin in nodeMCU
8. GND pin is connected to GND pin in nodeMCU

9. Vcc pin is connected to 3v3 pin in nodeMCU
10. Soil sensor pins are connected to the LM393 pins

Project:

STEP1: obtaining the soil moisture levels from the soil sensor

The soil moisture sensor is placed inside the plant and measures the moisture and sends the data across to the NodeMCU through the LM393

The only libraries we use are for the MQTT server to upload our data and publish it on the server as well as IFTTT app.

STEP1(Continuation):

- We define the sensor pin for A0 which is used for the soil moisture sensor

- Initially the sensor gives us the value for the soil moisture which will give us the readings on the IFTTT app as well as the adafruit.io server

STEP2:based on the soil moisture level the motor pump is controlled through the relay module

In the same manner as the last step, we initialize and define the necessary variables.

Based on the moisture levels, the logic will be applied and the plant will be watered by using the motor pump which is powered through a relay module if the plant is underwatered.

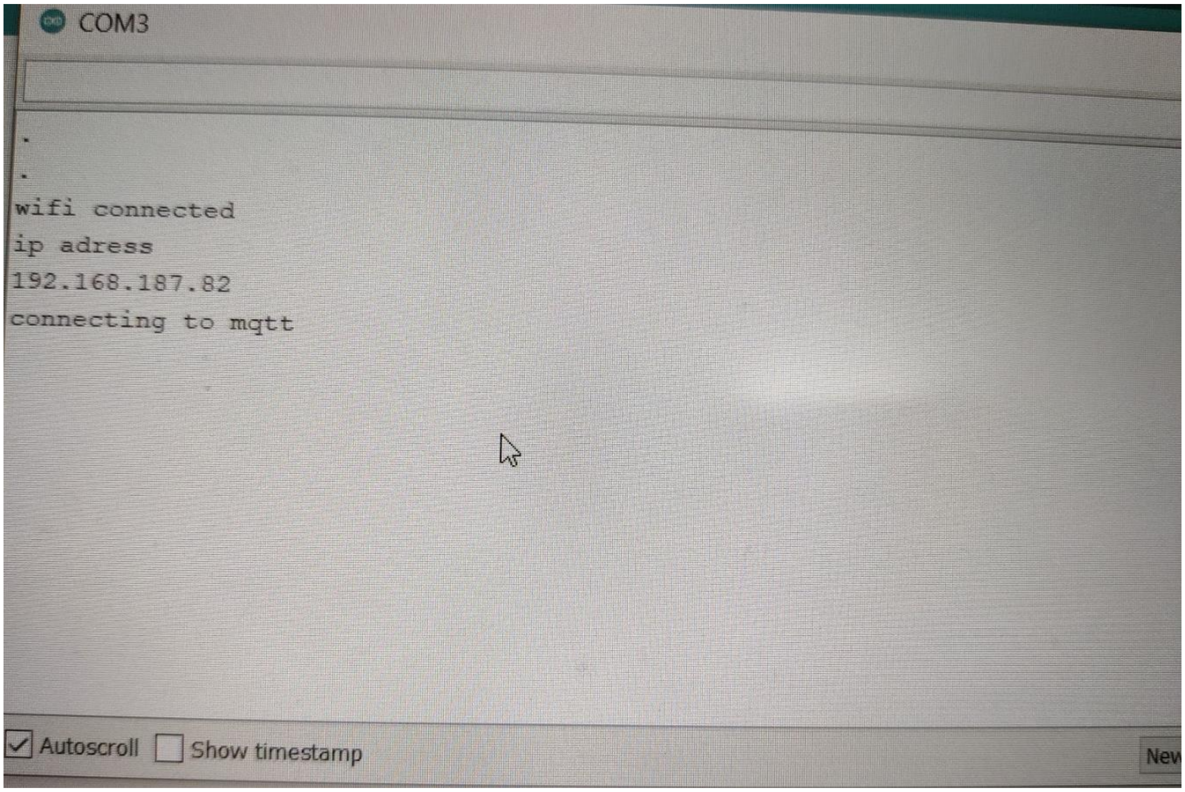
STEP3: Publishing data for the user to monitor

We publish the data so that the user can see for himself/herself what is happening real time and can fix issues if there are any.
For this purpose, we use 2 more libraries:

- ESP8266Wifi
- Adafruit_MQTT_Client
- IFTTT

ESP8266Wifi:

- Our NodeMCU has a wifi module inbuilt in it. Therefore, we can connect our NodeMCU to our local network, i.e. , Wifi and access the internet using our NodeMCU module.



```
.
.
wifi connected
ip adress
192.168.187.82
connecting to mqtt
```

Adafruit_MQTT_Client:

- Once , our NodeMCU is connected to the internet, we must also connect it to the cloud platform so that the output provided, so as to upload the data of our sensors to the server for monitoring purposes
- The cloud platform we are using is Adafruit.io which uses MQTT protocol for communication.

- MQTT is a simple publish-subscribe network protocol that transfers messages between devices.

IFTTT:

First we download the app on the phone and connect it to the adafruit.io server using basic steps that we are guided through by the app itself.

The notifications update us if our plants are OVERWATERED, WATERED correctly or UNDERWATERED

STEP3:(Continuation)

- We have to first go to Adafruit website, create an account if we don't have one. We have to note the API key of our Adafruit account which we'll further use in the code .
- We then have to create a dashboard and then inside the dashboard create a field where we would be uploading the recorded moisture data.
- Now, coming back to our code, we must define our server and port (1883 default) and also define our Adafruit username and API key.
- Now, we could publish our Adafruit field .
- Now, once our NodeMCU is connected to the internet and our MQTT Client is established, the data will get published every 5 seconds on the server

APPLICATIONS:

As such there can be 1 application for this project, that is the automated plant watering system, it can also be used in synergy with adding other systems to the project like varying sunlight that is provided to the plants, changing of humidity and controlling more factors for the plants.