DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING & TECHNOLOGY

IOT PROJECT REPORT

SUBJECT TITLE : INTERNET OF THINGS

SUBJECT CODE: 15IT422E

SUBMITTED TO: Prof Kayalvizhi Jayavel

DRIP IRRIGATION SYSTEM

HARISRIGUHAN S

RA1611008010005

Department of Information Technology

SRM
UNIVERSITY
(Under section 3 of UGC Act 1956)

SRM University, SRM Nagar, Kattankulathur-603203
Kanchipuram District, Tamil Nadu

LINKS TO GITHUB AND YOUTUBE:

YouTube:

https://www.youtube.com/watch?v=pmvjlaZf0Ls&feature=youtu.be

Github:

https://github.com/Harisriguhan/Drip-Irrigation-System

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to our IoT Professor Mrs Kayalvizhi Jayavel for being out there at every step of my course and guiding us all along the way to be capable of moulding our ideas into smart projects.

I would also like to thank my parents, without whom I wouldn’t be able to anything in any regard.

ABSTRACT

Drip irrigation is more effective and less expensive if a large amount of soil can be wetted with each emitter without losing water or nutrients below the root zone. The distance that water spreads horizontally from a drip line and the volume of soil wetted are limiting factors that determine the spacing and number of drip lines and emitters, the frequency of irrigation, and thus the cost of irrigation. We used numerical simulations and field trials to investigate the effects of application rate, pulsed water application, and antecedent water content on the spreading of water from drip emitters. Simulation results showed that pulsing and lower application rates produced minor increases in horizontal spreading at the end of water application. The small increases were primarily due to longer irrigation times, however, and not to flow phenomena associated with pulsing or low application rates. Moreover, the small increases mostly disappeared after the infiltrated water had redistributed for a period of 24 h. Field trials confirmed the simulation findings, with no statistically significant difference in wetting being found among five water application treatments involving pulsed applications and varying application rates. The simulations showed that higher antecedent water content increases water spreading from drip irrigation systems, but the increases were greater in the vertical direction than in the horizontal, an undesirable outcome if crop roots are shallow or groundwater contamination is a concern. Overall, soil texture (hydraulic properties) and antecedent water content largely determine the spreading and distribution of a given water application, with pulsing and flow rate having very little impact.

HARDWARE REQUIRED:

- NodeMCU ESP8266
- 9V DC Motor
- Bread Board
- Female-Male Jumper wires

- Male-Male Jumper wires
- 2-Channel Relay
- Soil Moisture Sensor
- 9V Battery

SOFTWARE REQUIRED :

- Arduino IDE.
- ESP8266 library.
- Adafruit MQTT Library.

MISCELLANEOUS :

- Wifi Internet Connection
- Soldering tool
- Soldering wires

TOTAL COST OF COMPONENTS :- RS.1800 - RS.2000.

SYSTEM OVERVIEW

The main component of the setup is the Nodemcu ESP8266 module. All the other hardware components are connected to the Nodemcu. The board is programmed in Arduino IDE and uses the ESP8266, Adafruit MQTT libraries. These libraries have been added to the Arduino IDE. The MPU6050 module with gyroscope and accelerometer is directly connected to the Nodemcu using a micro USB cable.

Programming NodeMCU and setting up Adafruit Dashboard :

- Import Adafruit Library to the Arduino IDE
- Fill your Adafruit token, ssid, and wifi password to the code.
- Upload drip irrigation code to your NodeMCU
- Setup your Adafruit dashboard, add a custom widget, choose two state widget and give it name.

CODE

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#define WLAN_SSID       "HSG"
#define WLAN_PASS       "goyyale123"
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "HSGSIVAK"
#define AIO_KEY         "65affc8256a747cca4d64c7190519337"
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER,
AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
Adafruit_MQTT_Subscribe relaymotor =
Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/relay-
motor");
```

```
Adafruit_MQTT_Publish soilmoisture =
Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/soil-
moisture-sensor");
void MQTT_connect();
const int sensor_pin = A0;
int relayInput =D1;
void setup() {
  pinMode(relayInput, OUTPUT);
  Serial.begin(9600);
  digitalWrite(relayInput, HIGH);
  delay(10);
  Serial.println(F("Drip Irrigation"));
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);
  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(50);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
  mqtt.subscribe(&relaymotor);
}
uint32_t x=0;
void loop() {
  MQTT_connect();
  Adafruit_MQTT_Subscribe *subscription;
  float moisture_percentage;
  moisture_percentage = ( 100.00 -
( (analogRead(sensor_pin)/1023.00) * 100.00 ) );
  Serial.print("Soil Moisture(in Percentage) = ");
  Serial.print(moisture_percentage);
  Serial.println("%");
  if(! soilmoisture.publish((float)moisture_percentage))
    Serial.println(F("Failed"));
  else
    Serial.println(F("Uploaded"));
  if(moisture_percentage<20.0)
   digitalWrite(relayInput, LOW);
  else
```

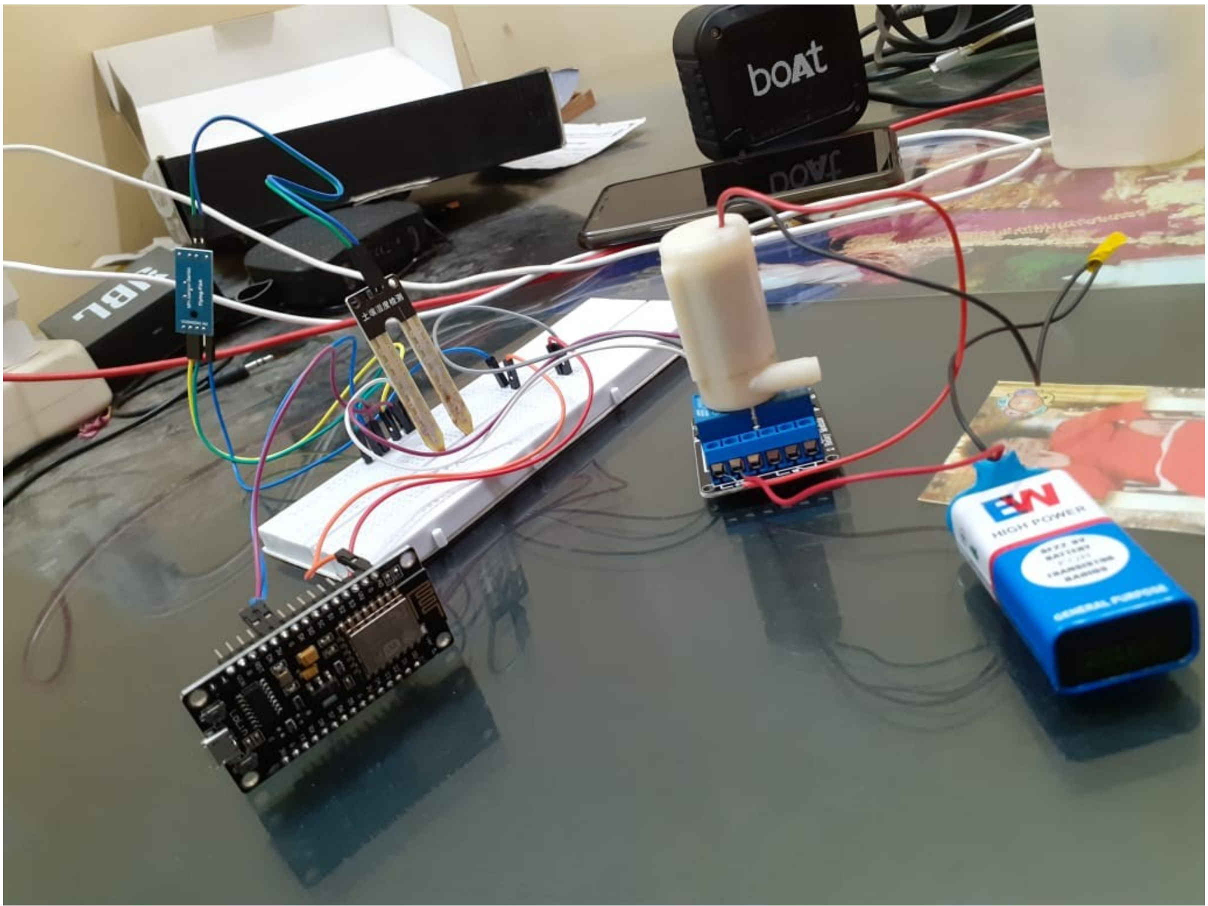
```
   digitalWrite(relayInput, HIGH);
  while ((subscription = mqtt.readSubscription(5000)))
  {
    if (subscription == &relaymotor)
    {
      Serial.println((char*)relaymotor.lastread);
      if(strcmp((char*)relaymotor.lastread,"ON")==0)
       digitalWrite(relayInput, LOW);
      else
       digitalWrite(relayInput, HIGH);
    }
  }
}
void MQTT_connect() {
  int8_t ret;

  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }

  Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0
for connected
       Serial.println(mqtt.connectErrorString(ret));
       Serial.println("Retrying MQTT connection in 5 seconds...");
       mqtt.disconnect();
       retries--;
       if (retries == 0) {
         // basically die and wait for WDT to reset me
         while (1);
       }
  }
  Serial.println("MQTT Connected!");
}
```

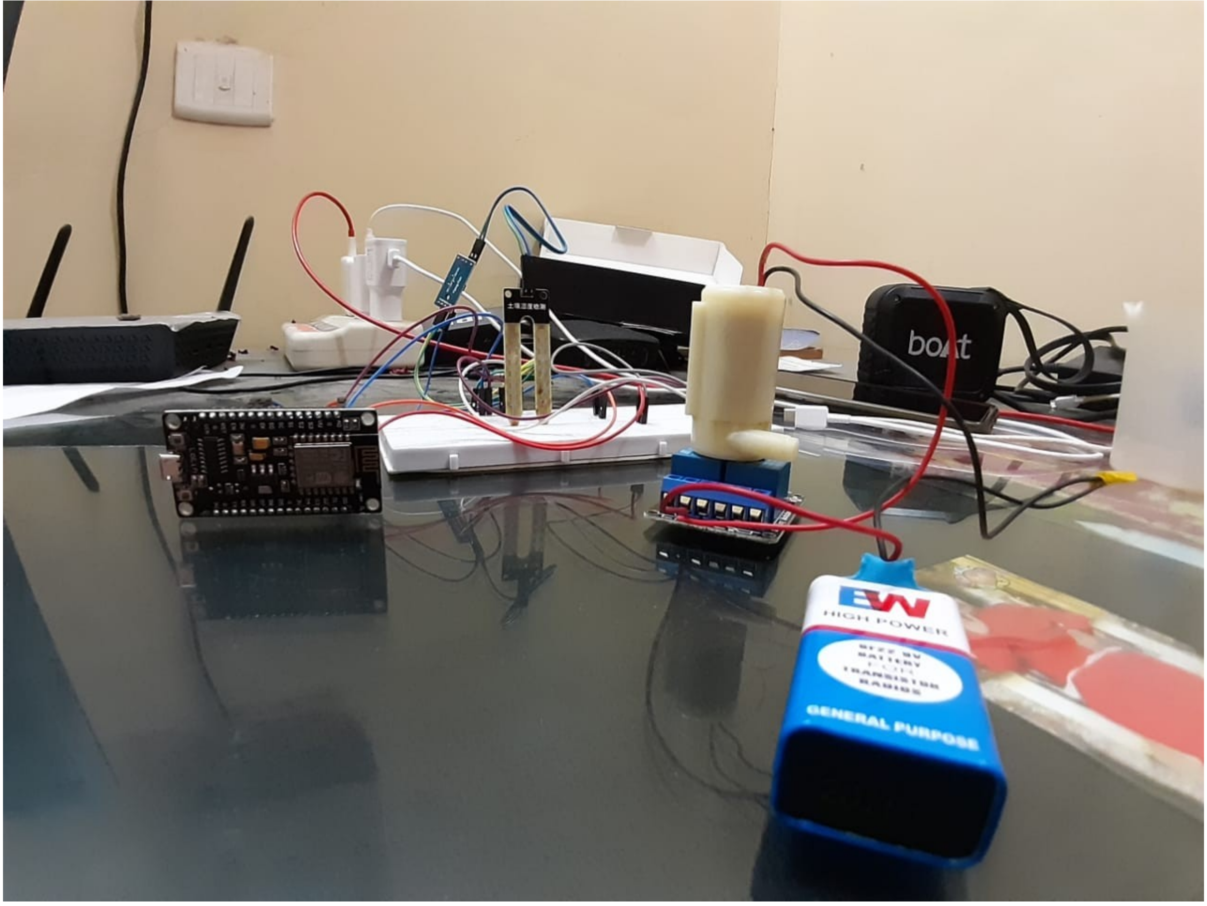
Adafruit Dashboard, Widgets and Triggers :

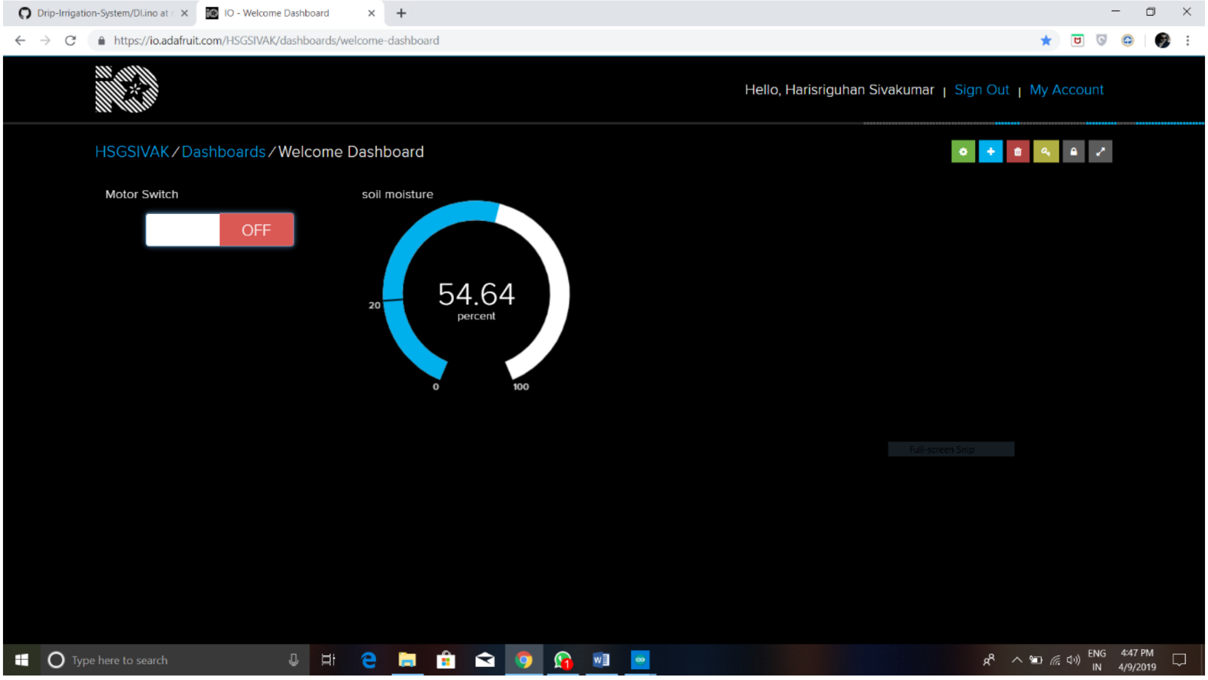
We give our WiFi SSID and password inside the code before compiling and uploading it to the Nodemcu. Also in order to connect to the Adafruit database, we give the Adafruit username, Adafruit password and Adafruit client ID so that it can connect via MQTT.

Result :

Drip Irrigation using ESP8266 NodeMCU has been successfully developed and implemented.


boAt
EW
HIGH POWER
GENERAL PURPOSE


boAt
EW
HIGH POWER
GENERAL PURPOSE



Drip-Irrigation-System/DI.ino at
IO - Welcome Dashboard
https://io.adafruit.com/HSGSIVAK/dashboards/welcome-dashboard
Hello, Harisriguhan Sivakumar | Sign Out | My Account
HSGSIVAK / Dashboards / Welcome Dashboard
20
0
100
Type here to search
ENG
IN
4:47 PM
4/9/2019

Motor Switch OFF

soil moisture 54.64 percent