SOIL MOISTURE & AUTOMATIC PLANT WATERING SYSTEM

Group D: Munashe Chabvuta(24603)

1. Introduction

Soil Moisture refers to the amount of water contained or held within the soils. It plays a crucial role in plant development and performance, hence is regarded a measure of soil health. It is a key climate variable affected by precipitation, temperature, soil characteristics and many other variables which play part in the hydrological cycle. The availability of fertile soils only, does not guarantee healthy thriving plants, if the soil moisture present does not meet the requirements of the plant to optimize its daily operations. According to an article by Greenway Biotech, the optimum soil moisture level which plants need for survival ranges between 20% to 60%(Greenway Biotech 2021).Greenway Biotech

Soil moisture is major variable in plant functions. It acts as a solvent where nutrients and minerals are broken down for easy absorption via plant roots. It provides turgidity which gives the plant true stability, positioning and further controls the exchange of heat energy and water between the land surface and the atmosphere through evapotranspiration. Furthermore soil moisture plays a crucial role in the development of weather patterns and the formation of precipitation. As the climate and weather patterns change, moisture availability is becoming more variable affecting plant functions. As a result soil moisture data can be taken into consideration for reservoir management, early warning of droughts, irrigation scheduling and crop yield forecasting.

With the importance of soil moisture data at hand, the main aim of the Arduino based project is to monitor moisture availability in a regular house plant in real time over the internet of things through the use of microcontrollers and software. The system has to maintain the moisture content of the soil between the values of 40% and 60% which is the optimum moisture of functionality of the plant under investigation. When the moisture content is below 40% the system has to increase the moisture content by turning on the pump and a fulfilling the moisture requirements of the plant. Once the moisture percentage reaches a value of 60% the water pump switches off. The whole Arduino Based system is brought to life by the following materials and methods.

2. Materials and Methods

- ESP32 Wrover-B Module
- Capacitive Soil Moisture Sensor
- 5V water submersible pump
- 2N2222 NPN Transistor
- Breadboard
- Jump Wires

2.1 ESP32 Wrover-B Module

With the capabilities of executing as a standalone system, the ESP32 Wrover-B Module offers vast advantages at a lower price promoting the reduction of communication stack overhead, as compared to the Arduino R3 microcontroller. It can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI/SDIO OR I2C/UART interfaces which makes it a suitable candidate as the core microcontroller for the real time soil moisture monitoring, by establishing communication with the Internet of things and related applications. Besides qualifying for real time monitoring, the module is equipped with vast other advantages such as its capability of operating in industrial environments within temperatures between -40 to +125 degrees Celsius, giving leeway to environmental monitoring in hard weather environments. The module can be programmed using the Arduino IDE which establishes its capacity of interfacing with various environmental measuring sensors, hence transmitting the acquired data over the internet of things.

Fig.1: The ESP32 WROVER-B MODULE

2.2 Capacitive Soil Moisture Sensor

A capacitive moisture sensor has the ability to measure the change in capacitance which is contributed by the changes in the dielectric created between the soil and water. The sensor rather measures the ions that are dissolved in the moisture and does not take into account the measurement of moisture directly. Furthermore, the ion concentration can be affected by various

factors, such as fertilizers which decrease the resistance of the soil hence unreliable moisture readings. It is manufactured from corrosion resistant material which is a major advantage over the resistive soil moisture sensor, allowing an outstanding service life. It is equipped with a voltage regulator which allowing it to operate at voltage ranging from 3.3 to 5.5V.

Fig2.Capacitive soil moisture sensor.

2.3 Transistor 2N2222

It is a Negative-Positive-Negative(NPN) bipolar junction transistor that makes use of both electrons and electron holes as charge carriers. It permits a trivial current administered at one of its terminals to control a much larger current flowing between the the two terminals resulting in the switching of the 5V submersible pump used in the project. The transistor consists of three pins which have different purposes and connections to deliver its intended use. These pin are referred to as the collector, base and emitter.

• Collector: this is the pin where the power flows in.

• Base: this is the trigger pin which is connected to the ESP32-WROVER-B Module and which will be used as an output pin.

• Emitter: this is the ground side of the transistor.

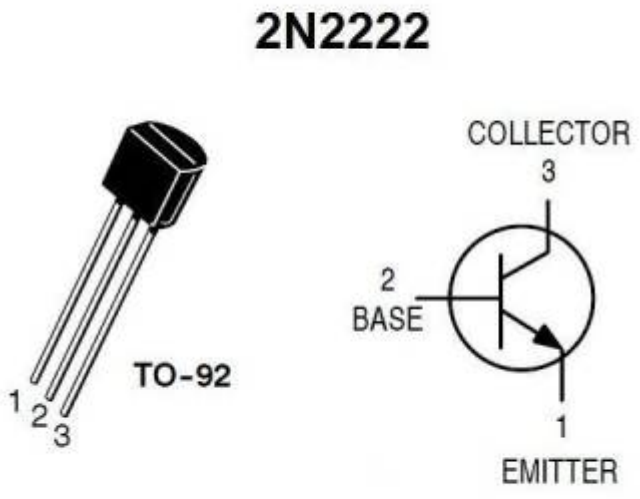


fig3: Shows current flowing from the Collector to the Emitter

2.4 IoT (Internet of Things)

It refers to a network of devices know as “things” that are embedded with sensors, software and other technologies, used to exchange and provide sensor data to cloud storage and cloud computing resources, which provide analysis and processing of data for more deep insights. For the soil moisture monitoring project “ThingSpeak” was considered for visualizations of the soil moisture data. ThingSpeak is an IoT analytics platform that offers the service to aggregate, visualize and analyze live stream via channel streams in the cloud. The ESP32 Module will be transmitting real time data from the soil moisture to the internet of things after every 15-30 seconds for a good visualization of the moisture present in the plant as a percentage and this can be observed remotely on a mobile device.

2.5 Connections Overview of Hardware

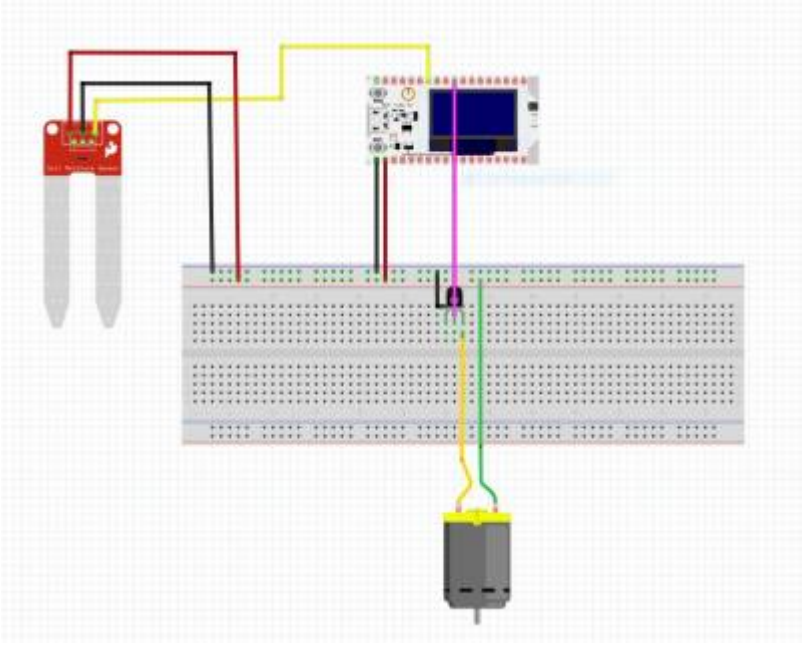


Fig4: Shows the Hardware connections, The components used in the diagram are for illustration.

The above diagram was produced in fritzing and shows how the soil moisture sensor, the 5V water pump and the ESP32 Module, are connected and to which pins. The capacitive soil moisture sensor consist of three pins, the GND, VCC and the Analogue pin. The Ground and VCC are connected to the ground and 5V of the ESP32 module respectively via the breadboard. The third pin of the sensor is connected to GPIO 39 of the microcontroller. The pump consist of two pins where one leg is connected to 5V of the microcontroller and the other leg is connected to the collector of the transistor . The base pin of the transistor is connected to GPIO32 of the microcontroller, where the pin mode is set to output.

The Overview of the experiment set up can further be supported by by the following schematic show below

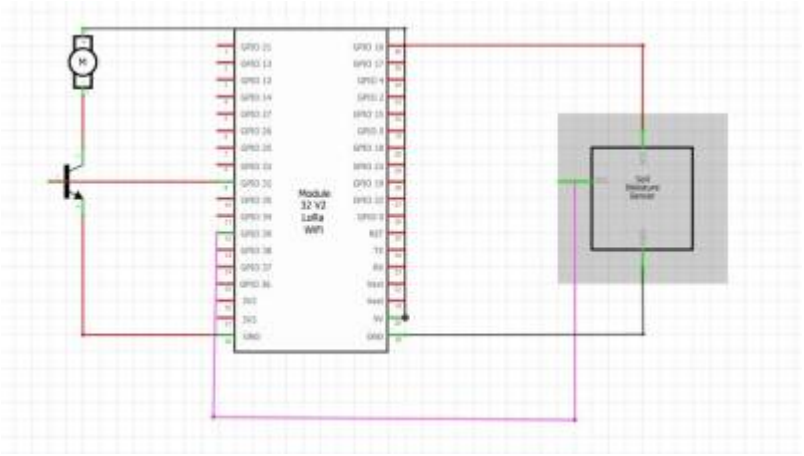


Fig5: Schematic between the ESP32 & Module parts.

3.Results

This section exhibits the final code which drives the whole soil moisture monitoring system after dry running the individual module codes for a better understanding.

3.1 Code

```
// WiFi for esp32 - Version: Latest
#include <ETH.h>
#include <WiFi.h>
#include <WiFiAP.h>
#include <WiFiClient.h>
#include <WiFiGeneric.h>
#include <WiFiMulti.h>
#include <WiFiSTA.h>
#include <WiFiScan.h>
#include <WiFiServer.h>
#include <WiFiType.h>
#include <WiFiUdp.h>


#include <WiFi.h>

String api_key = "FZN1AWGPKWQ968IH";     // This is a unique key obtained from the ThingSpeak after a creating a channel.It allows users to write data to a channel or read data from a private channel.
const char *ssid =  "BH1919";     // Service Set Identifier, Its a unique name which identifies my WI-Fi network.
const char *pass =  "Schnorre nicht mein Wlan du Knecht"; //This is the password of the Wi-Fi network.
const char* server = "api.thingspeak.com"; // Allows connection to the thingSpeak channel


WiFiClient client; // Intialiazation of the Client Library.

int analogPin = 39;
int moisture;
int moisture_percentage;

int map_low = 3455; // This is the dry air value of the moisture sensor
int map_high = 1625;// this the water value of the moisture sensor
int water_pump = 32;

void setup()
{
       Serial.begin(9600); // Starts serial commuincation baud rate set to 9600
       delay(100);
       pinMode(32, OUTPUT);// Sets the pump pin as output connected to the base of the transistor
       digitalWrite(32, 0);
       Serial.println("Connecting to ");
       Serial.println(ssid);
```

```
       WiFi.begin(ssid, pass);

      while (WiFi.status() != WL_CONNECTED)
     {
            delay(500);
            Serial.print(".");
     }
      Serial.println("");
      Serial.println("WiFi connected");

}

void loop()
{
 moisture = analogRead(analogPin);// allows the variable moisture to store
analogue values from the moisture sensor
  moisture_percentage = map(moisture, map_low, map_high, 0, 100);// fuction
map is used to map the analogue values from the moisture sensor to a
percentage

   if(moisture_percentage <= 40){
      digitalWrite(water_pump, HIGH);// if the moisture is less than or
equal to 40  the pump pin is High
      delay(5000);
   }
   if(moisture_percentage >= 60){
     digitalWrite(water_pump, LOW);//if the moisture is greater or equal to
60 the pump turns off
     delay(5000);
   }


               if (client.connect(server,80))   //   "184.106.153.149" or
api.thingspeak.com
            {
                  String data_to_send = api_key;
                   data_to_send += "&field1=";
                   data_to_send += moisture_percentage;

                   data_to_send += "\r\n\r\n";

                  client.print("POST /update HTTP/1.1\n");
                  client.print("Host: api.thingspeak.com\n");
                  client.print("Connection: close\n");
                  client.print("X-THINGSPEAKAPIKEY: " + api_key + "\n");
                  client.print("Content-Type: application/x-www-form-
urlencoded\n");
                  client.print("Content-Length: ");
                  client.print(data_to_send.length());
                  client.print("\n\n");
                  client.print(data_to_send);
```

```
                    delay(1000);

                    Serial.println(" Send to Thingspeak.");
                }
           client.stop();

           Serial.println("Waiting...");

  // thingspeak needs minimum 15 sec delay between updates, i've set it to
30 seconds
  delay(10000);
}
```

3.2 Overview & Explanation of the Code

Below is an explanation of some of the key features including functions used, which are vital for the operation of the soil moisture monitoring system.

• As the main aim of the project suggest, real time soil moisture monitoring system provides the analysis of moisture data over the internet of things. The ESP32- Wrover B module was used as the backbone of the whole operation. To enable the ESP32 to establish a communication channel with the Internet of Things certain libraries such as the Wi-Fi library had to be included which is shown in the code as "# Include <WiFi.h>. The library is responsible for configuring the ethernet library and network settings.

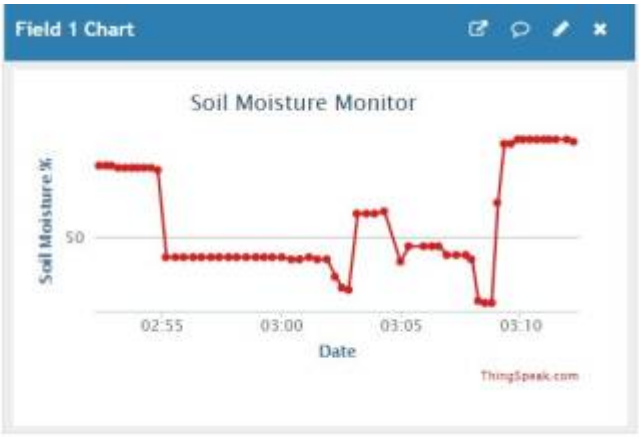
• The data pins of the moisture sensor were further defined and variables declared which are going to be holding values received from the soil moisture sensor.

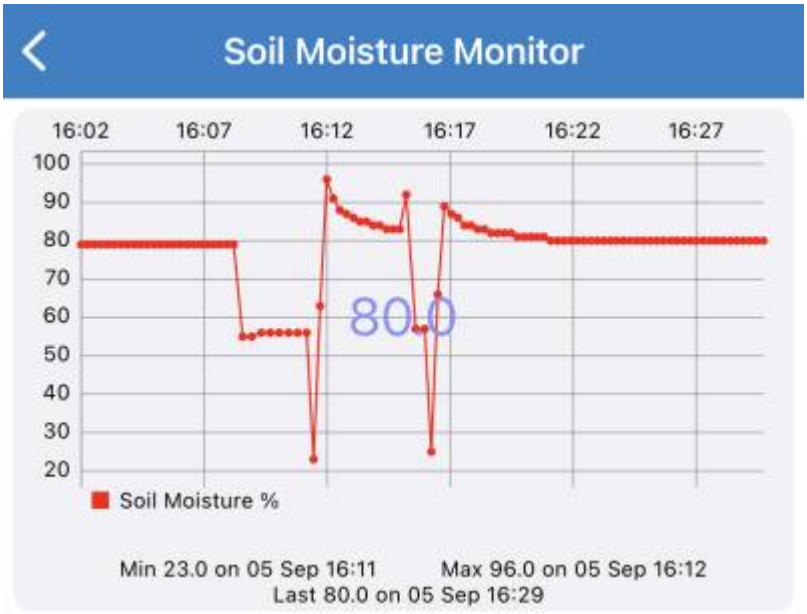
• The ESP32 can measure varying voltages between 0V and 3,3V. The voltage measured is then assigned to values between 0 and 4095. 0V corresponds to 0 and 3,3 V to 4095. With the capacitive soil moisture sensor connected to GPIO 39 of the ESP32, it measured the dry air of which it was assigned the value 3455 which became the lower limit of the measurement. Further more the water value of the moisture sensor was assigned the value 1625 which became the upper limit of the measurement. A map function was implemented in the code to map the analogue values obtained from the moisture sensor to a percentage for easier visualization in the thingspeak channel. This allows us to map the percentage of soil moisture which is responsible for switching the pump on and off when the conditions are met.

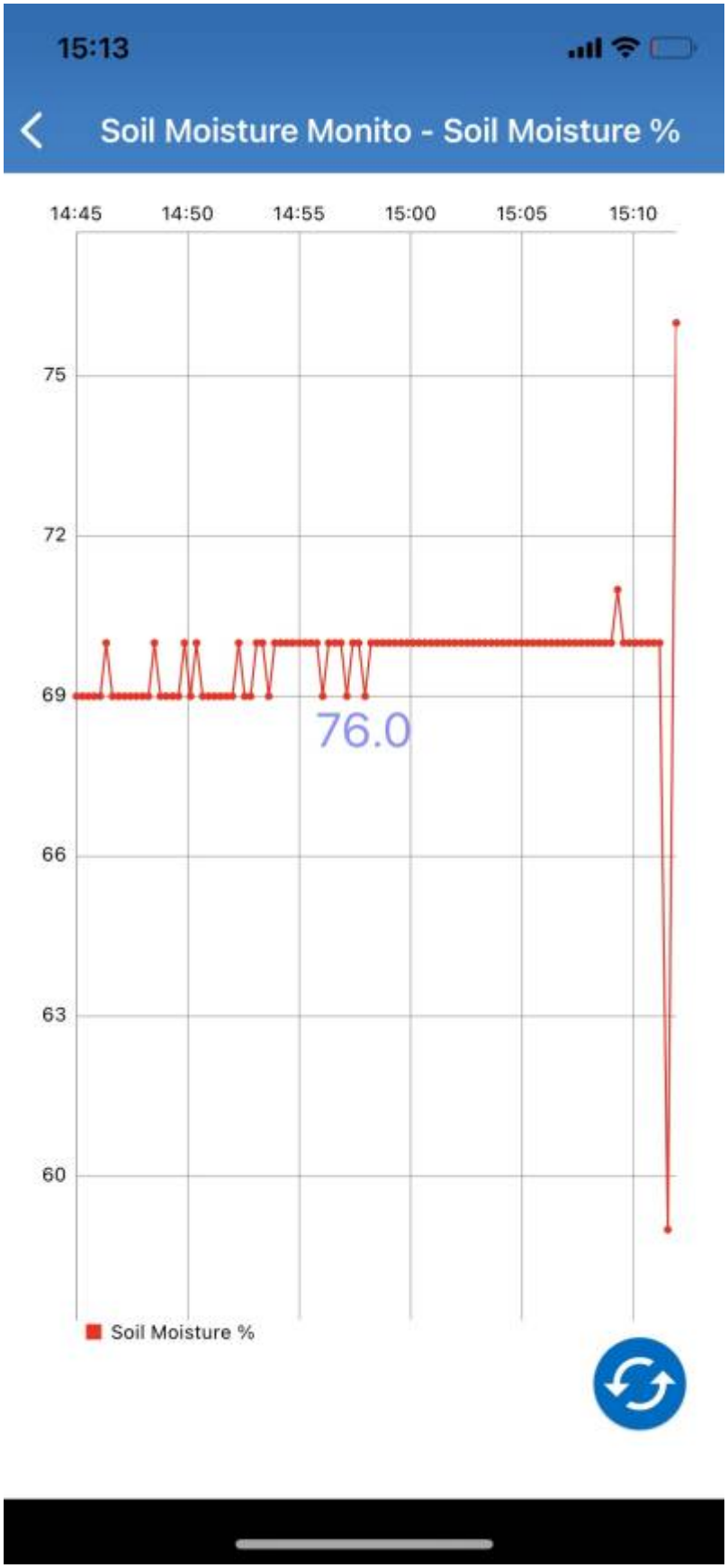
• The WIFI Client is responsible for establishing a connection to a specified IP address and port.

• Once this is established the serial monitor will confirm each time it updates values to the thingspeak and its usually a 15 sec delay but in the code it was set to 30 sec.

3.3 Results From Things Speak


Field 1 Chart
Soil Moisture Monitor
Soil Moisture %
50
02:55
03:00
03:05
03:10
Date
ThingSpeak.com


Soil Moisture Monitor
16:02
16:07
16:12
16:17
16:22
16:27
100
90
80
70
60
50
40
30
20
80.0
Soil Moisture %
Min 23.0 on 05 Sep 16:11
Max 96.0 on 05 Sep 16:12
Last 80.0 on 05 Sep 16:29



4. Discussion and Improvements

5. Reference