VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

L5 + L6

IoT Fundamentals – ECE3501

Allen Ben Philipose – 18BIS0043

Lab Task – 5

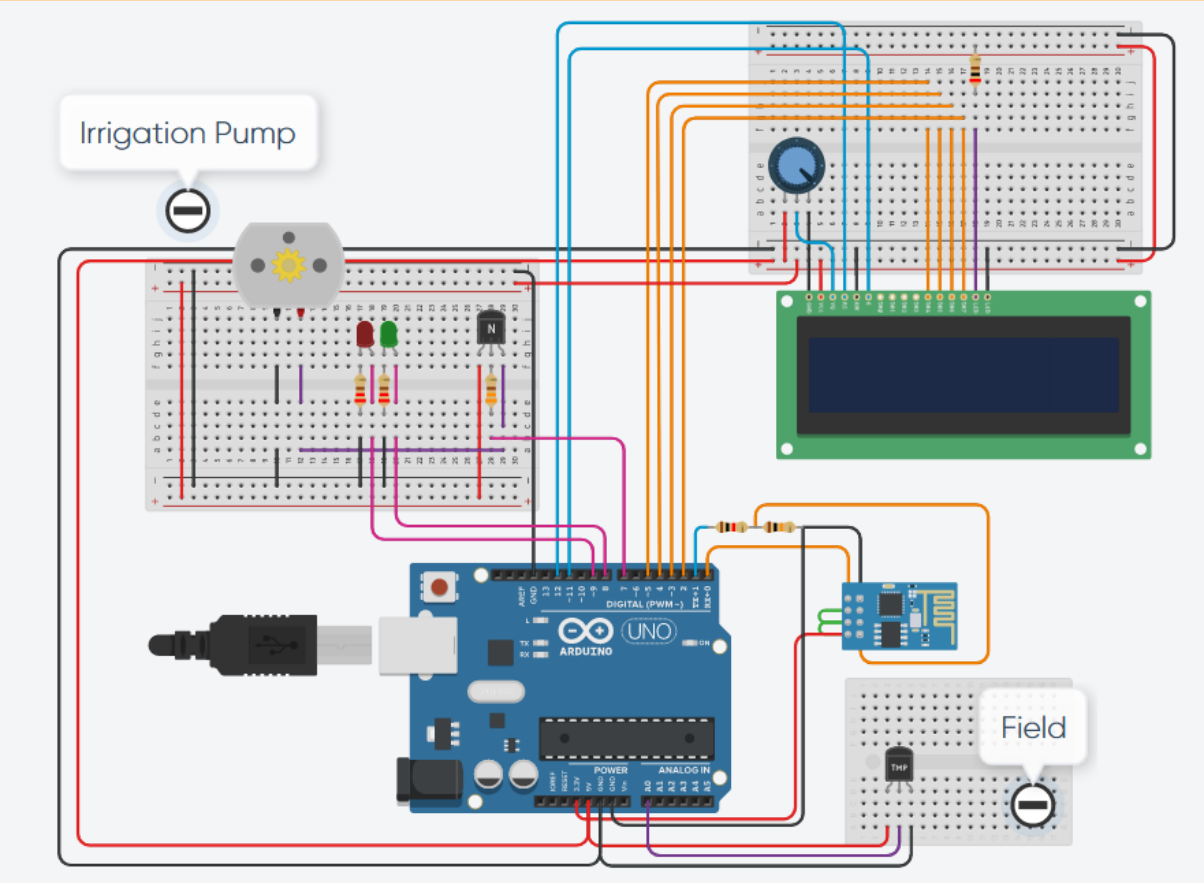
To: Prof. Suresh Chavhan

AUTOMATED WATER IRRIGATION SYSTEM

Aim

To design a circuit using Arduino for monitoring the temperature of a field and if the temperature is above a certain limit, give a command to turn on the irrigation motor. Also transmit the detected levels to ThingSpeak for further analysis.

Circuit Diagram

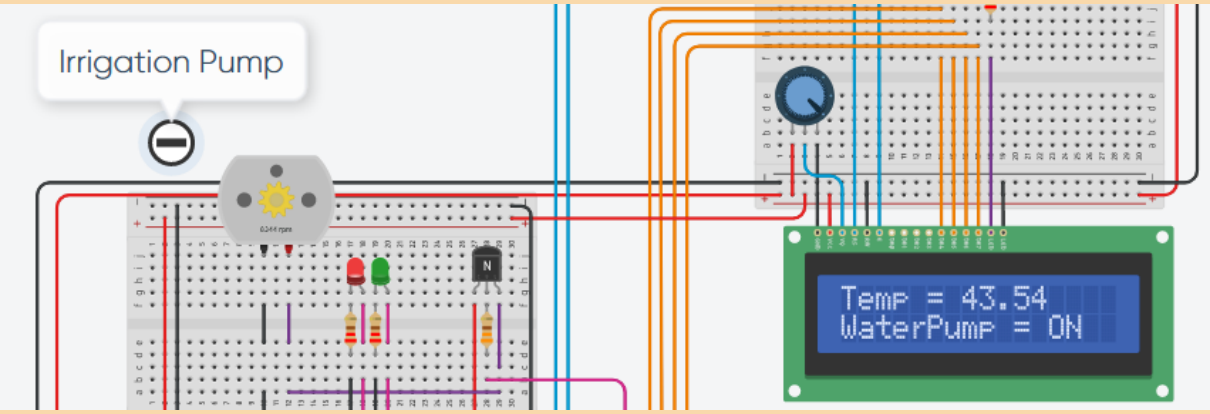


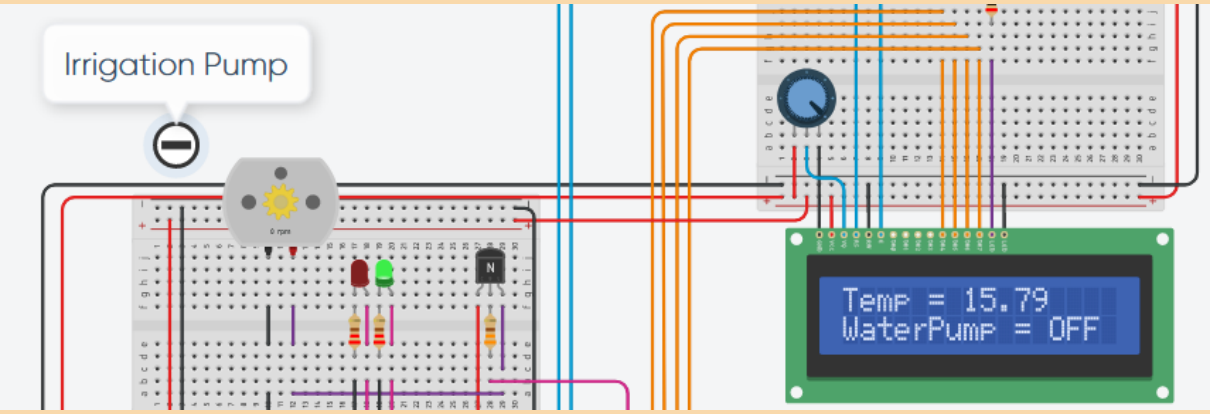
Tools Required

Tinkercad – for simulating the connection and coding of the Arduino circuit

ThingSpeak – for plotting the graph

Output from Tinkercad





Serial Monitor

```
AT+CIPSEND=98
GET /update?api_key=8XN8RYFFNZRKNYZS&field5=22&field6=0&amp HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=98
GET /update?api_key=8XN8RYFFNZRKNYZS&field5=47&field6=1&amp HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=98
GET /update?api_key=8XN8RYFFNZRKNYZS&field5=47&field6=1&amp HTTP/1.1
Host: api.thingspeak.com
```

Code

```
#include <LiquidCrystal.h>
String ssid     = "Simulator Wifi";
String password = "";
String host     = "api.thingspeak.com";
const int httpPort   = 80;
String      uri                                 =
"/update?api_key=8XN8RYFFNZRKNYZS&field5=";
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int setupESP8266(void) {
  // Start our ESP8266 Serial Communication
  Serial.begin(115200);
  Serial.println("AT");
  delay(10);
  if (!Serial.find("OK")) return 1;
  Serial.println("AT+CWJAP=\"" + ssid + "\",\""
+ password + "\"");
  delay(10);
```

```
if (!Serial.find("OK")) return 2;

  // Open TCP connection to the host:

  Serial.println("AT+CIPSTART=\"TCP\",\""     +
host + "\"," + httpPort);

  delay(50);        // Wait a little for the ESP
to respond


if (!Serial.find("OK")) return 3;

  return 0;

}


void anydata(int t1, int t2) {

  int temp1 = map(t1,0,1000,0,1000);

  int temp2 = map(t2,0,1,0,1);

  String  httpPacket  =  "GET  "  +  uri  +
String(temp1)  +  "&field6="  +  String(temp2)
+"&amp"  +  "  HTTP/1.1\r\nHost:  "  +  host  +
"\r\n\r\n";

  int length = httpPacket.length();

  Serial.print("AT+CIPSEND=");

  Serial.println(length);

  delay(10);
```

```
  Serial.print(httpPacket);
  delay(10);
  if (!Serial.find("SEND OK\r\n")) return;
}

void setup() {
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);

  lcd.begin(16, 2);
  lcd.print("Irrigation ");
  lcd.setCursor(0,1);
  lcd.print("System");
  delay(1000);
  lcd.clear();
  lcd.print("Temp = ");
  lcd.setCursor(0,1);
  lcd.print("WaterPump = ");
  setupESP8266();
}
```

```
void loop() {
  lcd.setCursor(0, 1);
  int temp1 = analogRead(A0);
  float temp = ((temp1 * 170.0/1023.0)-3.32);
  lcd.setCursor(7,0);
  lcd.print("       ");
  lcd.setCursor(7,0);
  lcd.print(temp);
  int t2 = 0;
  lcd.setCursor(12,1);
  // Upper limit of the temperature is 56 degrees
  // And the lower limit is 0 degrees
  // To simulate a real life scenario

  if (temp > 35){
    digitalWrite(7, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(8, LOW);
    lcd.print("ON ");
    t2 = 1;
  }
```

```
  else {
    digitalWrite(7, LOW);
    digitalWrite(9, LOW);
    digitalWrite(8, HIGH);
    lcd.print("OFF");
    t2 = 0;
  }
  anydata(temp, t2);
}
```

Observations

Program working as expected – minimal latency and accuracy of detecting temperature makes effective for analysis and for proactive response of the irrigation pump, in a real-world scenario.

Conclusion

Therefore, by using Tinkercad, we simulated a circuit for detecting the temperature of a field and updated the live situation in ThingSpeak for further analysis and storage. This system also allows the control for irrigation pump automatically based on the temperature detected and the values have been adjusted correspondingly for considering real-world situations.

Output from ThingSpeak