# Soil humidity and temperature monitoring system using thingspeak

A soil humidity and temperature monitoring system using Arduino and ThingSpeak is a practical and efficient way to keep track of environmental conditions in agricultural or gardening settings. This system combines hardware components like sensors and an Arduino microcontroller with cloud-based data storage and visualization through the ThingSpeak platform. Here's a detailed description of such a system:

**Components Required:**

1. **Arduino Board:** You can use an Arduino Uno, Arduino Nano, or any compatible board.

2. **Soil Moisture Sensor:** This sensor measures the moisture content in the soil. It typically has two probes that you insert into the soil.

3. **Temperature Sensor:** A temperature sensor like the DHT11 or DHT22 can be used to measure soil temperature.

4. **Wi-Fi Module:** An ESP8266-based Wi-Fi module (e.g., ESP-01, ESP-12) to connect the Arduino to the internet.

5. **Breadboard and Jumper Wires:** These are used for connecting the components on a breadboard.

6. **Power Supply:** Depending on your setup, you may need a power supply to power the Arduino and sensors.

**System Description:**

1. **Hardware Setup:**
   - Connect the soil moisture sensor and temperature sensor to the Arduino using jumper wires.
   - Power the Arduino using a USB cable or an external power supply.
   - Connect the Wi-Fi module to the Arduino if it's not integrated (e.g., in the case of an Arduino Uno + ESP8266 setup).

2. **Arduino Code:**
   - Write an Arduino sketch (code) to read data from the soil moisture and temperature sensors.
   - Use appropriate libraries for sensor data acquisition (e.g., DHT library for the DHT temperature sensor).
   - Create variables to store the sensor readings.

3. **ThingSpeak Setup:**
   - Sign up for a ThingSpeak account if you don't already have one.
   - Create a new ThingSpeak channel where you can store and visualize your sensor data.

4. **Arduino to ThingSpeak Integration:**
   - Use the ThingSpeak API and Arduino libraries to send sensor data to your ThingSpeak channel.
   - You'll need the Write API Key from your ThingSpeak channel to enable data posting.

5. **Data Visualization:**

```
   - ThingSpeak provides built-in tools for visualizing your data, such as line ch
arts, gauges, and maps.
   - Set up charts to display soil moisture and temperature trends over time.

6. **Alerts and Notifications (Optional):**
   - Configure ThingSpeak to send you alerts or notifications when sensor readings
fall outside desired ranges. For example, if soil moisture is too low, you could r
eceive an email or SMS alert.

7. **Remote Monitoring:**
   - With your system in place, you can now monitor soil humidity and temperature
remotely through the ThingSpeak website or mobile app.

Source code:

#include"dht.h"                      // Including library for dht

#include<LiquidCrystal.h>

LiquidCrystal lcd(14,15,16,17,18,19);

#include<Timer.h>

Timer t;

#include <SoftwareSerial.h>

SoftwareSerial Serial1(2, 3);


#define dht_dpin 12

#define heart 13

dht DHT;

char *api_key="SIWOYBX26OXQ1WMS";     // Enter your Write API key from ThingSpeak

static char postUrl[150];

int humi,tem;

void httpGet(String ip, String path, int port=80);


void setup()

{

 lcd.begin(16, 2);

 lcd.clear();

 lcd.print("   Humidity   ");

 lcd.setCursor(0,1);

 lcd.print("  Measurement ");
```

```
 delay(2000);
 lcd.clear();
 lcd.print("Circuit Digest ");
 lcd.setCursor(0,1);
 lcd.print("Welcomes You");
 delay(2000);
 Serial1.begin(9600);
 Serial.begin(9600);
 lcd.clear();
 lcd.print("WIFI Connecting");
 lcd.setCursor(0,1);
 lcd.print("Please wait....");
 Serial.println("Connecting Wifi....");
 connect_wifi("AT",1000);
 connect_wifi("AT+CWMODE=1",1000);
 connect_wifi("AT+CWQAP",1000);
 connect_wifi("AT+RST",5000);
 connect_wifi("AT+CWJAP=\"1st floor\",\"muda1884\"",10000);
 Serial.println("Wifi Connected");
 lcd.clear();
 lcd.print("WIFI Connected.");
 pinMode(heart, OUTPUT);
 delay(2000);
 t.oscillate(heart, 1000, LOW);
 t.every(20000, send2server);
}
void loop()
{
  DHT.read11(dht_dpin);
```

```
  lcd.setCursor(0,0);

  lcd.print("Humidity: ");

  humi=DHT.humidity;

  lcd.print(humi);   // printing Humidity on LCD

  lcd.print(" %    ");

  lcd.setCursor(0,1);

  lcd.print("Temperature:");

  tem=DHT.temperature;

  lcd.print(tem);   // Printing temperature on LCD

  lcd.write(1);

  lcd.print("C   ");

  delay(1000);

  t.update();

}

void send2server()

{

  char tempStr[8];

  char humidStr[8];

  dtostrf(tem, 5, 3, tempStr);

  dtostrf(humi, 5, 3, humidStr);

  sprintf(postUrl, "update?api_key=%s&field1=%s&field2=%s",api_key,humidStr,tempSt
r);

  httpGet("api.thingspeak.com", postUrl, 80);

}
```

//GET https://api.thingspeak.com/update?api_key=SIWOYBX26OXQ1WMS&field1=0

```
void httpGet(String ip, String path, int port)

{

  int resp;

  String atHttpGetCmd = "GET /"+path+" HTTP/1.0\r\n\r\n";
```

```
  //AT+CIPSTART="TCP","192.168.20.200",80
  String atTcpPortConnectCmd = "AT+CIPSTART=\"TCP\",\""+ip+"\","+port+"";
  connect_wifi(atTcpPortConnectCmd,1000);
  int len = atHttpGetCmd.length();
  String atSendCmd = "AT+CIPSEND=";
  atSendCmd+=len;
  connect_wifi(atSendCmd,1000);
  connect_wifi(atHttpGetCmd,1000);
}
void connect_wifi(String cmd, int t)
{
  int temp=0,i=0;
  while(1)
  {
    lcd.clear();
    lcd.print(cmd);
    Serial.println(cmd);
    Serial1.println(cmd);
    while(Serial1.available())
    {
      if(Serial1.find("OK"))
      i=8;
    }
    delay(t);
    if(i>5)
    break;
    i++;
  }
  if(i==8)
```

```
  {
   Serial.println("OK");
         lcd.setCursor(0,1);
       lcd.print("OK");
  }
  else
  {
   Serial.println("Error");
          lcd.setCursor(0,1);
       lcd.print("Error");
  }
}
```