**Temperature, Humidity and Soil Moisture Sensor with Nodemcu**

**Project Overview:**

The Crop Monitoring and Controlling System is a comprehensive project designed to facilitate efficient monitoring and management of crop conditions. The project is divided into different parts, and this specific part focuses on integrating temperature, humidity, and soil moisture sensors with NodeMCU ESP32 (also compatible with ESP8266) using the Arduino IDE. The gathered data is then visualized in real-time on the Asksensor platform and can be conveniently downloaded as a Google Sheet for record-keeping.

**Components:**

* NodeMCU ESP32/ESP8266:

The NodeMCU ESP32/ESP8266 serves as the central microcontroller, connecting and processing data from various sensors.

Compatible with the Arduino IDE for easy programming and customization.

* Temperature Sensor:

Measures the ambient temperature in the crop environment.

Provides crucial data for assessing optimal temperature conditions for different crops.

Humidity Sensor:

* Monitors the humidity levels in the air.

Essential for understanding the moisture content in the atmosphere, influencing crop growth.

* Soil Moisture Sensor:

Measures the moisture content in the soil.

Helps in determining when and how much to irrigate, ensuring optimal soil conditions for crop growth.

* **Asksensor Platform:**

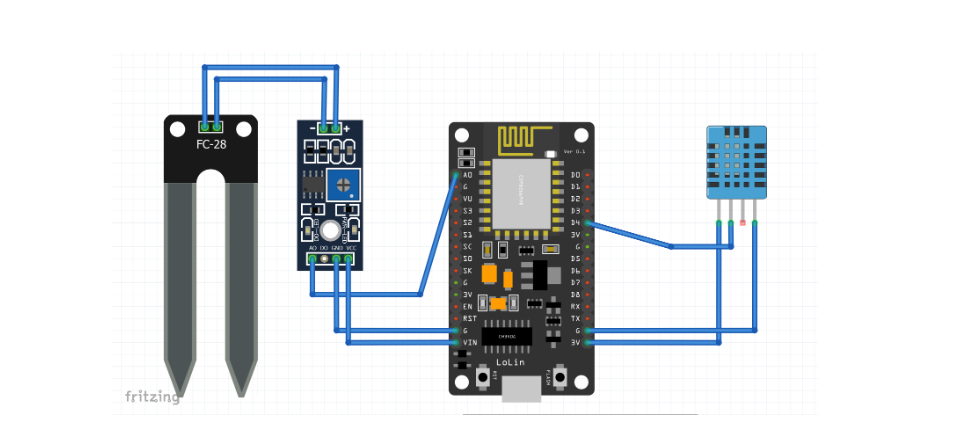
Real-time data visualization platform for monitoring and tracking sensor data.

Provides a user-friendly interface for easy access to crop-related information.

* Google Sheets Integration:

Data collected by the sensors is automatically logged into a Google Sheet.

Enables users to maintain a historical record of temperature, humidity, and soil moisture levels over time.



To run the provided Arduino code successfully, you need to install the following libraries:

1. \*\*WiFi Library:\*\*

- This library is used for connecting to a Wi-Fi network.

- You can install it through the Arduino Library Manager.

- Go to Arduino IDE -> Sketch -> Include Library -> Manage Libraries.

- Search for "WiFi" and install the library.

2. \*\*WiFiMulti Library:\*\*

- This library is used to manage multiple Wi-Fi connections.

- Install it through the Arduino Library Manager.

- Go to Arduino IDE -> Sketch -> Include Library -> Manage Libraries.

- Search for "WiFiManager" and install the library.

3. \*\*HTTPClient Library:\*\*

- This library is used to make HTTP requests.

- Install it through the Arduino Library Manager.

- Go to Arduino IDE -> Sketch -> Include Library -> Manage Libraries.

- Search for "HTTPClient" and install the library.

4. \*\*Adafruit Unified Sensor Library:\*\*

- This library is used for unified sensor management.

- Install it through the Arduino Library Manager.

- Go to Arduino IDE -> Sketch -> Include Library -> Manage Libraries.

- Search for "Adafruit Unified Sensor" and install the library.

5. \*\*DHT Sensor Library:\*\*

- This library is specifically for DHT sensors (temperature and humidity).

- Install it through the Arduino Library Manager.

- Go to Arduino IDE -> Sketch -> Include Library -> Manage Libraries.

- Search for "DHT sensor library" and install the library.

After installing these libraries, you should be able to compile and upload the provided code to your NodeMCU ESP32 or ESP8266 board. If you encounter any issues during compilation, make sure that all libraries are installed, and there are no errors in the code or missing dependencies.

Code :  [**Download**](https://github.com/vishal-ravi/projects/blob/main/humidity)

#include <WiFi.h>

#include <WiFiMulti.h>

#include <HTTPClient.h>

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

WiFiMulti WiFiMulti;

HTTPClient ask;

// user config: TODO

#define MOISTURE\_THRESHOLD 55 // moisture alert threshold

const char\* ssid = "\*\*\*"; // enter wifi user name

const char\* password = "\*\*\*"; // enter wifi password

const char\* apiKeyIn = "E5IiZ2835YBOgiiQwKjEErx7gaiaVIbw"; // Write your api key given from ask sensor

const unsigned int writeInterval = 25000; // write interval (in ms)

// ASKSENSORS config.

const char\* host = "api.asksensors.com"; // ASKSENSORS API host name

const int httpPort = 80; // port

// DHT config.

#define DHTPIN 4 ////d4 Pin which is connected to the DHT sensor.

// Uncomment the type of sensor in use:

#define DHTTYPE DHT11 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302)

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

DHT\_Unified dht(DHTPIN, DHTTYPE);

uint32\_t delayMS;

int status = WL\_IDLE\_STATUS;

float myTemperature = 0, myHumidity = 0;

int moisture\_Pin= A0; // Soil Moisture Sensor input at Analog PIN vp

int moisture\_value= 0, moisture\_state = 0xFF;

// create ASKSENSORS client

//

void setup() {

// open serial

Serial.begin(115200);

Serial.println("Wait for WiFi... ");

// connecting to the WiFi network

WiFiMulti.addAP(ssid, password);

while (WiFiMulti.run() != WL\_CONNECTED) {

Serial.print(".");

delay(500);

}

// connected

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

// Initialize device.

dht.begin();

Serial.println("DHTxx Unified Sensor Example");

// Print temperature sensor details.

sensor\_t sensor;

dht.temperature().getSensor(&sensor);

Serial.println("------------------------------------");

Serial.println("Temperature");

Serial.print ("Sensor: "); Serial.println(sensor.name);

Serial.print ("Driver Ver: "); Serial.println(sensor.version);

Serial.print ("Unique ID: "); Serial.println(sensor.sensor\_id);

Serial.print ("Max Value: "); Serial.print(sensor.max\_value); Serial.println(" C");

Serial.print ("Min Value: "); Serial.print(sensor.min\_value); Serial.println(" C");

Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println(" C");

Serial.println("------------------------------------");

// Print humidity sensor details.

dht.humidity().getSensor(&sensor);

Serial.println("------------------------------------");

Serial.println("Humidity");

Serial.print ("Sensor: "); Serial.println(sensor.name);

Serial.print ("Driver Ver: "); Serial.println(sensor.version);

Serial.print ("Unique ID: "); Serial.println(sensor.sensor\_id);

Serial.print ("Max Value: "); Serial.print(sensor.max\_value); Serial.println("%");

Serial.print ("Min Value: "); Serial.print(sensor.min\_value); Serial.println("%");

Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println("%");

Serial.println("------------------------------------");

// Set delay between sensor readings based on sensor details.

delayMS = sensor.min\_delay / 1000;

}

void loop() {

Serial.println(" Temperature, Humidity and Soil Moisture sensors ");

Serial.print("Moisture level : ");

moisture\_value= analogRead(moisture\_Pin);

moisture\_value= moisture\_value/10;

Serial.println(moisture\_value);

if(moisture\_value > MOISTURE\_THRESHOLD) moisture\_state = 0;

else moisture\_state = 1;

// Read data from DHT

// Delay between measurements.

delay(delayMS);

// Get temperature event and print its value.

sensors\_event\_t event;

dht.temperature().getEvent(&event);

if (isnan(event.temperature)) {

Serial.println("Error reading temperature!");

}

else {

// Update temperature and humidity

myTemperature = (float)event.temperature;

Serial.print("Temperature: ");

Serial.print(myTemperature);

Serial.println(" C");

}

// Get humidity event and print its value.

dht.humidity().getEvent(&event);

if (isnan(event.relative\_humidity)) {

Serial.println("Error reading humidity!");

}

else {

myHumidity = (float)event.relative\_humidity;

Serial.print("Humidity: ");

Serial.print(myHumidity);

Serial.println("%");

}

// Use WiFiClient class to create TCP connections

WiFiClient client;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;

}else {

// Create a URL for the request

String url = "https://api.asksensors.com/write/";

url += apiKeyIn;

url += "?module1=";

url += myTemperature;

url += "&module2=";

url += myHumidity;

url += "&module3=";

url += moisture\_value;

url += "&module4=";

url += moisture\_state;

Serial.print(" requesting URL: ");

Serial.println(url);

ask.begin(url); //Specify the URL

Serial.println("Temperature, Humidity and Soil moisture data sent to Asksensors");

//Check for the returning code

int httpCode = ask.GET();

if (httpCode > 0) {

String payload = ask.getString();

Serial.println(httpCode);

Serial.println(payload);

} else {

Serial.println("Error on HTTP request");

}

ask.end(); //End

Serial.println(" End ");

}

client.stop(); // stop client

delay(writeInterval); // delay

}