```
========================================
Aduino uno code

========================================

#include <ArduinoJson.h>   // json library
#include <dht.h>   //  DHT11 temparature and humidity sensor library
#include <OneWire.h>   // Soil temp sensor
#include <DallasTemperature.h>  // Soil temp sensor

// json library
String message = "";
bool messageReady = false;
// json library


dht DHT;  //Environment Humidity and temparature sensor
#define DHT11_PIN 9   // Environment Humidity and temparature sensor


// Resgister 4.7k  must be conected with data and VCC conected
#define ONE_WIRE_BUS 7 // soil temp sensor data digital pin 7
OneWire oneWire(ONE_WIRE_BUS);  // Setup a oneWire instance to communicate
with any OneWire devices - soil temp sensor
DallasTemperature sensorsSoil(&oneWire); // Pass our oneWire reference to soil
temp sensor
int SoilTemp; // soil temp


// sun light detection -----------
int light;  // light sensor output in serial
// ------------------- setup --------------------


// water level sensor -----start------
int trigPin = 11;    // Trigger -  water sensor
int echoPin = 12;    // Echo    -  water sensor
long duration;
float cm;
int WaterLevel;
// water level sensor--------End-----


// Soil-Moisture sensor -----Start---

const int AirValue = 590;   //you need to replace this value with Value_1
const int WaterValue = 310;  //you need to replace this value with Value_2
int soilMoistureValue = 0;
int soilmoisturepercent=0;
int SoilMoisture;   // soil moisture percentage

// Soil-Moisture sensor -----End-----
```

```
void setup() {
  //Serial Port begin
  Serial.begin(9600);


 // water level sensor ----------start-------
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  // water level sensor ----------end-------


pinMode (8,OUTPUT);  // confirm the sun light

  sensorsSoil.begin();  // soil temp sensor start
}


void loop() {


// Environment Humidity and temparature sensor ------------start-----

    int chk = DHT.read11(DHT11_PIN);

// Environment Humidity and temparature sensor------------end-------


// Water level sensor ---------start-------------------

// The sensor is triggered by a HIGH pulse of 10 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  // Convert the time into a distance - water level sensor
  cm = (duration/2) / 29.1;     // Divide by 29.1 or multiply by 0.0343

  if (cm > 2 && cm < 300 ){       // water level display; sensor 2-300 cm;
    WaterLevel = cm;
  }else{
  WaterLevel = 0;
```

```
  }

// water level sensor ---------End---------------------


// soil temp -------------start----------
sensorsSoil.requestTemperatures(); // Send the command to get temperature
readings
 //Serial.print("Soil Temp : ");
 SoilTemp = sensorsSoil.getTempCByIndex(0)- 2 ;  // Error mesurement (-)
// soil temp -------------End-------


// SoilMoisture sensor ---------start-------------
soilMoistureValue = analogRead(A0);  //put Sensor insert into soil
soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
if(soilmoisturepercent >= 100)
{
  SoilMoisture = 100;
}
else if(soilmoisturepercent <=0)
{
  SoilMoisture = 0;
}
else if(soilmoisturepercent >0 && soilmoisturepercent < 100)
{
  SoilMoisture = soilmoisturepercent;
}
// SoilMoisture sensor -------------End---------

// sunlight -----------start----------
  light = analogRead(A3);
if (light < 250){                 // sensor output high in sun
  digitalWrite(8, HIGH);
}else{
  digitalWrite(8,LOW);
}
//Serial.print(light);  //output high when 250 and low  in sun

// sunlight -----------End----------



  // Monitor serial communication with D1 mini
  while(Serial.available()) {
    message = Serial.readString();
    messageReady = true;
  }
  // Only process message if there's one
  if(messageReady) {
    // The only messages we'll parse will be formatted in JSON
    DynamicJsonDocument doc(1024); // ArduinoJson version 6+
    // Attempt to deserialize the message
```

```
    DeserializationError error = deserializeJson(doc,message);
    if(error) {
      Serial.print(F("deserializeJson() failed: "));
      Serial.println(error.c_str());
      messageReady = false;
      return;
    }
    if(doc["type"] == "request") {
      doc["type"] = "response";
      // Get data from analog sensors

      doc["Temperature"] = DHT.temperature;    // Environment temparature
      doc["Humidity"] = DHT.humidity;          // Environment Humidity
      doc["Water Level"] = WaterLevel;         // Water level
      doc["Soil Temparature"] = SoilTemp;      // Soil temparature
      doc["Soil Moisture"] = SoilMoisture;      // Soil Moisture
      doc["Sun light"] = light;                // sun light if 250 and high

      serializeJson(doc,Serial);
    }
    messageReady = false;
  }

//delay(1000);

}


================================================
D1 mini code
================================================


#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <ArduinoJson.h>

// ThingSpeak ----------------start---
#include <ThingSpeak.h>
#define CHANNEL_ID 1526074
#define CHALLEL_API_KEY "---------"
WiFiClient client;
// ThingSpeak ------------------End --



// display ---------------start--
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
```

```
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// display ----------------End--


  //Declar the veriable
  int Temperature = 0;
  int Humidity = 0;
  int WaterLevel = 0;
  int SoilTemp = 0;
  int SoilMoisture = 0;
  int light = 0;


// ledPin refers to D1 GPIO 2
int ledPin = 2;


ESP8266WebServer server;
char* ssid = "--------";
char* password = "-------";

void setup()
{
  WiFi.begin(ssid,password);
  Serial.begin(9600);
  while(WiFi.status()!=WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  server.on("/",handleIndex);
  server.begin();

  ThingSpeak.begin(client);

// Display --------------strat---------------------------

 // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
// Display ---------------End----------------------------
```

```
  // initialize digital pin ledPin as an output.
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  server.handleClient();

  delay(1000);
  handleIndex();

  delay(1000);
  LcdDispay();

  LEDlight();


 ThingSpeak.setField(1,Temperature);  // insert data
 ThingSpeak.setField(2,Humidity);
 ThingSpeak.setField(3,WaterLevel);
 ThingSpeak.setField(4,SoilTemp);
 ThingSpeak.setField(5,SoilMoisture);
 ThingSpeak.setField(6,light);

 ThingSpeak.writeFields(CHANNEL_ID, CHALLEL_API_KEY);   // send data

 delay (1000);

}

void handleIndex(void)
{
  // Send a JSON-formatted request with key "type" and value "request"
  // then parse the JSON-formatted response with keys "gas" and "distance"
  DynamicJsonDocument doc(1024);

  // Sending the request
  doc["type"] = "request";
  serializeJson(doc,Serial);
  // Reading the response
  boolean messageReady = false;
  String message = "";
  while(messageReady == false) { // blocking but that's ok
    if(Serial.available()) {
      message = Serial.readString();
      messageReady = true;
    }
  }
  // Attempt to deserialize the JSON-formatted message
  DeserializationError error = deserializeJson(doc,message);
  if(error) {
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.c_str());
```

```
    return;
  }

  Temperature = doc["Temperature"];
  Humidity = doc["Humidity"];
  WaterLevel =  doc["Water Level"];
  SoilTemp = doc["Soil Temparature"];
  SoilMoisture = doc["Soil Moisture"];
  light = doc["Sun light"];

  // Prepare the data for serving it over HTTP
  String output = "Room Temperature: " + String(Temperature) + "\n";
  output += "Room Humidity: " + String(Humidity)+ "\n";
  output += "Water Level: " + String(WaterLevel)+ "\n";
  output += "Soil Temparature: " + String(SoilTemp)+ "\n";
  output += "Soil Moisture: " + String(SoilMoisture)+ "\n";
  output += "Sun light: " + String(light)+ "\n";

    // Serve the data as plain text, for example
  server.send(200,"text/plain",output);
}

void LcdDispay(void)
{
    // Clear the display
  display.clearDisplay();
  //Set the color - always use white despite actual display color
  display.setTextColor(WHITE);
  //Set the font size
  display.setTextSize(1);
  //Set the cursor coordinates

  display.setCursor(0,0);
  display.print("Room Temp.:  ");
  display.print(Temperature);
  display.print(" C");

  display.setCursor(0,10);
  display.print("Humidity:    ");
  display.print(Humidity);
  display.print(" %");

  display.setCursor(0,20);
  display.print("Sun light:   ");
  display.print(light);
  display.print("  ADC ");

  display.setCursor(0,26);
  display.print("--------------------");

  display.setCursor(0,33);
  display.print("Soil Mois.:  ");
  display.print(SoilMoisture);
```

```
  display.print("  %");

  display.setCursor(0,43);
  display.print("Soil Temp.:  ");
  display.print(SoilTemp);
  display.print(" C");

  display.setCursor(0,53);
  display.print("Water dis:   ");
  display.print(WaterLevel);
  display.print(" cm");

  display.setCursor(0,60);
  display.print("--------------------");

  display.display();
}

void LEDlight(void)
{
  digitalWrite(ledPin, LOW);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, HIGH);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```