

```

#include<Servo.h>
//PARAMETER INITIALIZATION
Servo servol;
int pos = 0;
float val, voltage, temp;
String ssid      = "Simulator Wifi"; // SSID to connect to
String password = ""; //virtual wifi has no password
String host      = "api.thingspeak.com"; // Open Weather Map API
const int httpPort = 80;
String url       = "/update?api_key=S1P4ZR5IY9GWKRW2&field1=";
int temp1 = 0;
float duration, distance;
const int analogIn = A0;
int humiditysensorOutput = 0;
int RawValue= 0;
ble Voltage = 0;
double tempC = 0;
double tempF = 0;
float humidity = 0;

int inches = 0;
int cm = 0;

//ESP8266 INITIALIZATION
void setupESP8266(void){
    // Start our ESP8266 Serial Communication
    Serial.begin(115200); // Serial connection over USB to computer
    Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266
    delay(10); // Wait a little for the ESP to respond
    if (Serial.find("OK"))
        Serial.println("ESP8266 OK!!!");
    // Connect to Simulator Wifi
    Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
    delay(10); // Wait a little for the ESP to respond
    if (Serial.find("OK"))
        Serial.println("Connected to WiFi!!!");
    // Open TCP connection to the host:
    //ESP8266 connects to the server as a TCP client.
    Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\",\" + httpPort);
    delay(50); // Wait a little for the ESP to respond
    if (Serial.find("OK"))
        Serial.println("ESP8266 Connected to server!!!") ;
}

// For Ultrasonics
long readUltrasonicDistance(int triggerPin, int echoPin)
{
    pinMode(triggerPin, OUTPUT); // Clear the trigger
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    // Sets the trigger pin to HIGH state for 10 microseconds
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);

```

```

    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
    // Reads the echo pin, and returns the sound wave travel time in
microseconds
    return pulseIn(echoPin, HIGH);
}

//VOID SETUP
void setup()
{
    Serial.begin(9600);
    pinMode(A1, INPUT);
    setupESP8266();
    servol.attach(13);
    pinMode(3, OUTPUT);
}

//VOID LOOP
void loop()
{
    anydata();
    delay(7000);
}

//SENSOR DATA PROCESS AND TRANSMIT
void anydata(void)
{
    //Ultrasonic Sensor
    // measure the ping time in cm
    cm = 0.01723 * readUltrasonicDistance(7, 7);
    // convert to inches by dividing by 2.54
    inches = (cm / 2.54);
    delay(100); // Wait for 100 millisecond(s)

    // LED
    if (inches <= 10)
    {
        // turn the LED on (HIGH is the voltage level)
        digitalWrite(3, HIGH);
    }
    if (inches > 10)
    {
        // turn the LED off (LOW is the voltage level)
        digitalWrite(3, LOW);
    }

    // Humidity
    humiditysensorOutput = analogRead(A1);

    //Temperature Sensor
    val=analogRead(A0);
    voltage=val*0.0048828125;
    temp = (voltage - 0.5) * 100.0;

```

```

//Display the data
Serial.print("\nDistance(inches): ");
Serial.print(inches);
Serial.println("in");
Serial.print("Distance(cm): ");
Serial.print(cm);
Serial.println("cm");
Serial.print("Temperature: ");
Serial.println(temp);
Serial.print("Humidity : ");
Serial.print(map(humiditysensorOutput, 0, 1023, 10, 70));
Serial.println("%");
Serial.println(" ");

// Construct our HTTP call
if (temp1 == 15){
    temp1 = 0;
}
String httpPacket = "GET " + url + String(temp) + "&field2=" +
String(map(humiditysensorOutput, 0, 1023, 10, 70)) + "&field3=" +
String(inches) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
temp1 = temp1 + 1;
int length = httpPacket.length();
// Send our message length
Serial.print(String(temp) +String(humidity) +String(humidity));
Serial.print("AT+CIPSEND=");
Serial.println(length);
delay(10); // Wait a little for the ESP to respond if
(!Serial.find(">")) return -1;
// Send our http request
Serial.print(httpPacket);
delay(10); // Wait a little for the ESP to respond
if (Serial.find("SEND OK\r\n"))
    Serial.println("ESP8266 sends data to the server");
    Serial.println(" ");
}

```