

SOURCE CODE

```
#include <ESP8266WiFi.h>
#include "DHT.h"
// WiFi Credentials
const char* ssid = "SMART"; // SSID of your Wi-Fi
const char* password = "123456789"; // Password of your Wi-Fi
WiFiServer server(80);
// Pin definitions
int gas_pin = A0;
int gas_val = 0;
int m1 = D0, m2 = D1, m3 = D2, m4 = D3;
int relay1 = D4, relay2 = D8;
const int trigPin = D6;
const int echoPin = D5;
// Sensor variables
#define SOUND_VELOCITY 0.034
#define CM_TO_INCH 0.393701
long duration;
float distanceCm;
float distanceInch;
// DHT Sensor setup
#define DHTPIN D7
#define DHTTYPE DHT11
33DHT dht(DHTPIN, DHTTYPE);
float h = 0, t = 0, f = 0;
// State variables
int val = 0, val1 = 1, val2 = 1;
String msg = "STOP", msg1 = "RELAY1 OFF", msg2 = "RELAY2 OFF";
void setup() {
  dht.begin();
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(m1, OUTPUT);
  pinMode(m2, OUTPUT);
  pinMode(m3, OUTPUT);
  pinMode(m4, OUTPUT);
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  Serial.begin(9600);
  Serial.println();
```

```

Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password); // Connect to Wi-Fi
while (WiFi.status() != WL_CONNECTED)
{
  // Wait until connected
  delay(500);
  34Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
server.begin();
Serial.println("Server started");
Serial.println(WiFi.localIP()); // Print the IP address
delay(3000);
}
void loop() {
  // Robot Movement Control
  if (val == 1) {
    Serial.println("Moving forward");
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
  } else if (val == 0) {
    Serial.println("Stopped");
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
  }
  35// Read DHT sensor
  h = dht.readHumidity();
  t = dht.readTemperature();
  f = dht.readTemperature(true);
  // Read Gas Sensor
  gas_val = analogRead(gas_pin);
  // Print sensor data to serial monitor
  Serial.print("Humidity: "); Serial.println(h);
  Serial.print("Temperature (C): "); Serial.println(t);
  Serial.print("Temperature (F): "); Serial.println(f);
  Serial.print("Gas Value: "); Serial.println(gas_val);
}

```

```

// Read Ultrasonic Distance Sensor
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distanceCm = duration * SOUND_VELOCITY / 2;
distanceInch = distanceCm * CM_TO_INCH;

// Print distance
Serial.print("Distance (cm): "); Serial.println(distanceCm);
Serial.print("Distance (inches): "); Serial.println(distanceInch);

36// Handle client request
WiFiClient client = server.available();
if (!client) {
  return;
}
String req = client.readStringUntil('\r');
Serial.println(F("Request: "));
Serial.println(req);

// Handle GPIO requests
if (req.indexOf(F("/gpio/0")) != -1) {
  val = 0;
  msg = "STOP";
} else if (req.indexOf(F("/gpio/1")) != -1) {
  val = 1;
  msg = "FORWARD";
} else if (req.indexOf(F("/gpio/3")) != -1) {
  Serial.println("Relay 1 ON");
  digitalWrite(relay1, HIGH);
  val1 = 0;
  msg1 = "RELAY1 ON";
} else if (req.indexOf(F("/gpio/4")) != -1) {
  Serial.println("Relay 1 OFF");
  digitalWrite(relay1, LOW);
  val1 = 1;
37msg1 = "RELAY1 OFF";
} else if (req.indexOf(F("/gpio/5")) != -1) {
  Serial.println("Relay 2 ON");
  digitalWrite(relay2, HIGH);
  val2 = 0;
  msg2 = "RELAY2 ON";
}

```

```

} else if (req.indexOf(F("/gpio/6")) != -1) {
Serial.println("Relay 2 OFF");
digitalWrite(relay2, LOW);
val2 = 1;
msg2 = "RELAY2 OFF";
} else {
Serial.println(F("Invalid request"));
}

// Prepare response
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
html><html><head><title>Environment Monitoring</title><style>";
s += "a:link {background-color: YELLOW; text-decoration: none;} table, th, td {text-align:
center; padding: 8px;} </style>";
s += "<meta http-equiv='refresh' content='3'></head><body bgcolor='lightgreen'>";
s += "<h1 ALIGN=CENTER>ENVIRONMENT MONITORING</h1><h2
ALIGN=CENTER>Sensor Details</h2><table border=1 ALIGN=CENTER>";
// Display sensor data in a table
s += "<tr><th>Humidity (%)</th><td><h2>" + String(h) + "</h2></td></tr>";
38s += "<tr><th>Temperature (C)</th><td><h2>" + String(t) + "</h2></td></tr>";
s += "<tr><th>Speed</th><td><h2>" + String(gas_val) + "</h2></td></tr>";
s += "<tr><th>Level (cm)</th><td><h2>" + String(distanceCm) + "</h2></td></tr>";
s += "</table><br>";
// Buttons for controlling robot and relays
s += "<a href='http://" + WiFi.localIP().toString() + "/gpio/" + String(1 - val) + "'><h1>ROBO
- " + msg + "</h1></a>";
s += "<a href='http://" + WiFi.localIP().toString() + "/gpio/" + String(3 + (1 - val1)) +
"'><h1>" + msg1 + "</h1></a>";
s += "<a href='http://" + WiFi.localIP().toString() + "/gpio/" + String(5 + (1 - val2)) +
"'><h1>" + msg2 + "</h1></a>";
s += "</body></html>";
// Send the response to the client
client.print(s);
delay(100);
}

```