ENVIRONMENTAL MONITORING

Phase-3 Project

IMPLEMENTATION OF THE PROJECT

COMPONENTS USED:

- NodeMCU ESP32: This will be our microcontroller.
- DHT22: Sense the temperature and humidity.
- Wokwi Virtual Components: These are virtual components you can add in Wokwi for the web interface and simulation like button, resistor,LED bulb.

LIBRARIES USED:

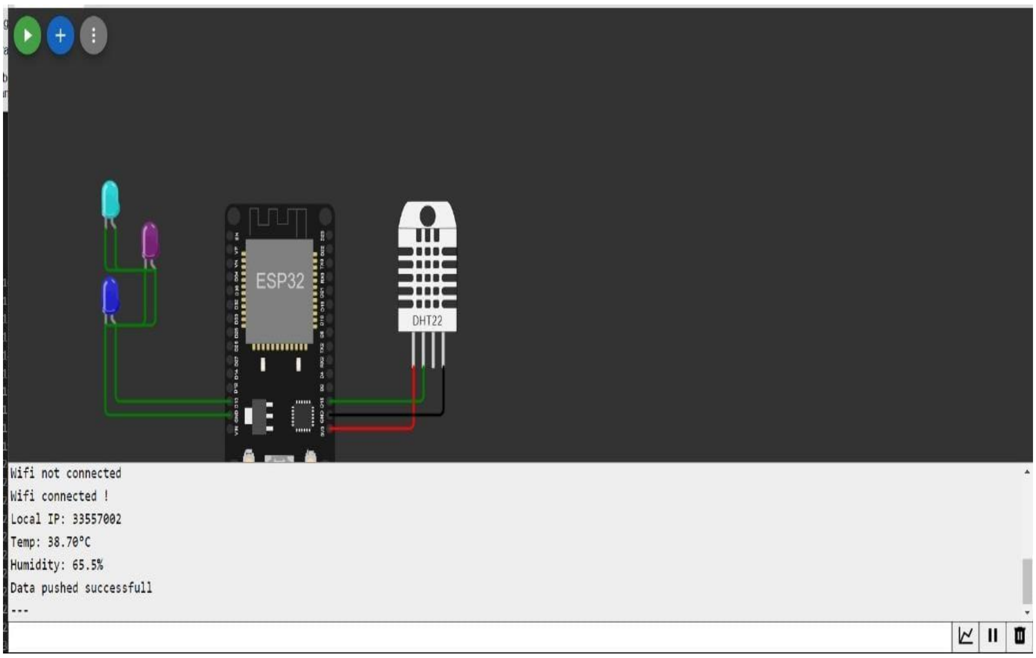
- DHT sensor library for ESPx
- PubSubClient

ARDUINO CODE:

(applied to ‘esp32’ to sense the temperature and humidity of the environment)

```
 #include <WiFi.h>
#include "DHTesp.h"
#include "ThingSpeak.h"

const int DHT_PIN = 15;
const int LED_PIN = 13;
const char* WIFI_NAME = "Wokwi-GUEST";
const char* WIFI_PASSWORD = "";
const int myChannelNumber =2307358 ;
const char* myApiKey = "1U2N21SZEGP74GFZ";
const char* server = "api.thingspeak.com";

DHTesp dhtSensor;
WiFiClient client;

void setup()  {

  Serial.begin(115200);
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  pinMode(LED_PIN, OUTPUT);
  WiFi.begin(WIFI_NAME, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED){
    delay(1000);
    Serial.println("Wifi not connected");
  }

  Serial.println("Wifi connected !");
  Serial.println("Local IP: " + String(WiFi.localIP()));
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
}
void loop() {
  TempAndHumidity  data = dhtSensor.getTempAndHumidity();
  ThingSpeak.setField(1,data.temperature);
  ThingSpeak.setField(2,data.humidity);
  if (data.temperature > 35 || data.temperature < 12 || data.humidity > 70 || data.humidity < 40) {
    digitalWrite(LED_PIN, HIGH);
  }else{
    digitalWrite(LED_PIN, LOW);
  }
    int x = ThingSpeak.writeFields(myChannelNumber,myApiKey);
    Serial.println("Temp: " + String(data.temperature, 2) + "°C");
  Serial.println("Humidity: " + String(data.humidity, 1) + "%");
    if(x == 200){
    Serial.println("Data pushed successfull");
  }else{
    Serial.println("Push error" + String(x));
  }
  Serial.println("---");
  delay(10000);
}
```

OUTPUT:



```
Wifi not connected
Wifi connected !
Local IP: 33557002
Temp: 38.70°C
Humidity: 65.5%
Data pushed successfull
---
```

CODE FOR CONNECTING THE ULTRASONIC SENSOR (HC-SP04) TO THE ARDUINO UNO BOARD

- Used to detect the number of people entered the environment we’ve provided certainly.

```
#include <NewPing.h>

#define TRIGGER_PIN 9
#define ECHO_PIN 10
#define MAX_DISTANCE 200 // Maximum distance we want to detect in centimeters (cm)

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum
distance.

void setup() {
  Serial.begin(9600); // Open a serial connection for debugging purposes.
}

void loop() {
  delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay
between pings.
  unsigned int distance = sonar.ping_cm(); // Send ping, get distance in centimeters.

  if (distance <= 200 && distance >= 2) { // Check if the distance is within the valid range.
    Serial.print("Number of people detected: ");
    Serial.println(distance); // Print the distance to the serial monitor.
  } else {
Serial.println("No people detected!"); // Print a message indicating no people are detected.
  }
  delay(1000); // Wait for a second before taking the next reading.
}
```