Phase 3 Development part 1
ENVIRONMENT MONITORING IN PARKS

DESCRIPTION:

This document is describing the process of developing the project with the mentioned components in phase 2 innovation segment. For our project we are going to use DHT22 Sensor and ESP32 module for indicating weather values to public.

COMPONENTS REQUIRED:

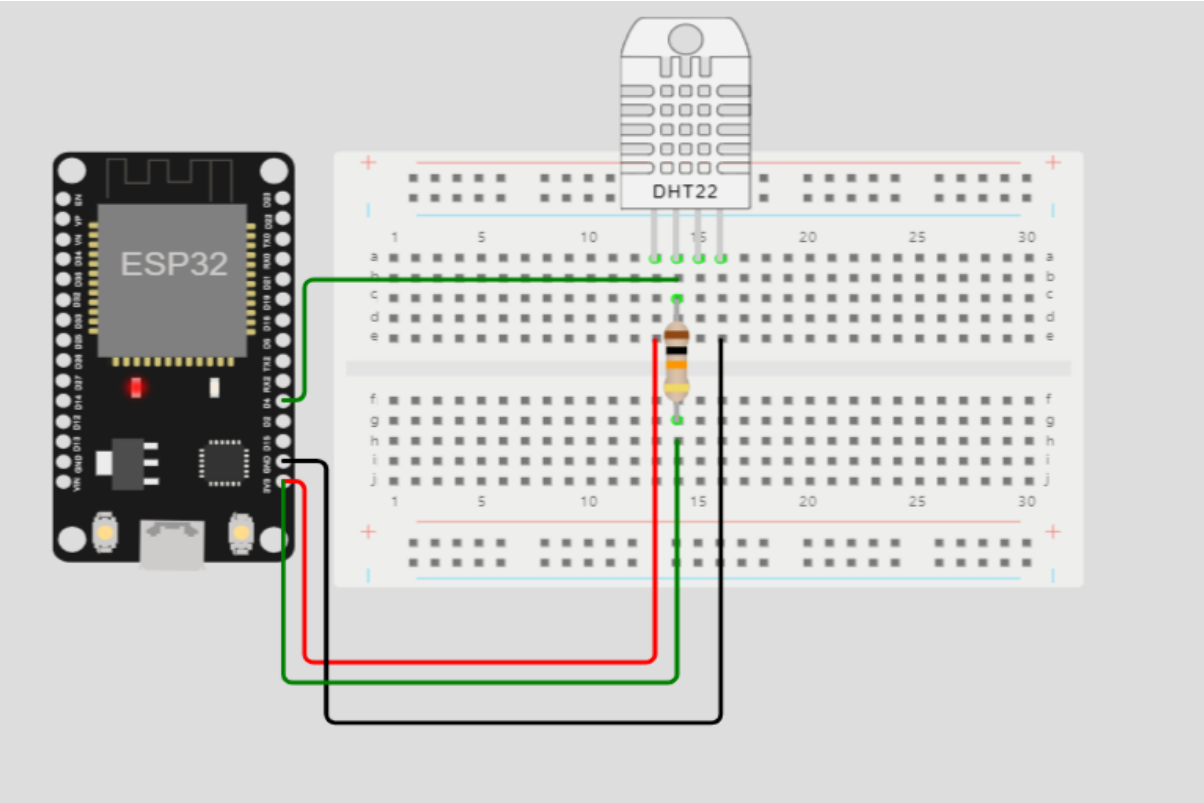
- ESP32
- DHT22 Sensor.
- Arduino IDE
- Power Supply.
- Basic components like Bread board, Wires, LED's.

WORKING DESCRIPTION:

The Sensing of values of temperature and humidity values by DHT22 sensor can be done in the basis of IOT enabled system. First, the ESP32 microcontroller is chosen for the simulation available in the wowki platform. Then the DHT22 sensor is connected to the microcontroller with the corresponding pins and checked for installation. Then the required wiring can be done and it will given to the power supply as the source of the elements. Then the circuit is checked for operation.

The DHT22 sensor operates on the basis of resistance changes with the variations in temperature and humidity values. The ESP32 microcontroller sends a signal to the sensor to read the data and processed the data to produce the output. The LED is placed in a circuit for the indication of changes received from the sensor. The program code was written for the operation of the sensor in the ESP32 Controller for reading the values. Then it is sent to the cloud using communication protocols and it is made available for the public using our concept of creating API with the access to the stored data in the cloud.

CIRCUIT DIAGRAM:



Here is the .json program for the above diagram as follows:

```
{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-breadboard-half", "id": "bb1", "top": -70.2, "left": -
102.8, "attrs": {} },
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -72.1, "left": -
225.8, "attrs": {} },
    { "type": "wokwi-dht22", "id": "dht1", "top": -134.1, "left": 23.4,
"attrs": {} },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 24,
      "left": 18.65,
      "rotate": 90,
      "attrs": { "value": "10000" }
    }
  ],
```

```
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "bb1:13t.e", "esp:3V3", "red", [ "v153.6", "h-153.6", "v-86.4" ] ],
    [ "bb1:16t.e", "esp:GND.1", "black", [ "v182.4", "h-172.8", "v-124.8",
"h0" ] ],
    [ "esp:3V3", "bb1:14b.h", "green", [ "v96", "h162.9" ] ],
    [ "bb1:14t.b", "esp:D4", "green", [ "v0", "h-163.2", "v57.6" ] ],
    [ "dht1:VCC", "bb1:13t.a", "", [ "$bb" ] ],
    [ "dht1:SDA", "bb1:14t.a", "", [ "$bb" ] ],
    [ "dht1:NC", "bb1:15t.a", "", [ "$bb" ] ],
    [ "dht1:GND", "bb1:16t.a", "", [ "$bb" ] ],
    [ "r1:1", "bb1:14t.c", "", [ "$bb" ] ],
    [ "r1:2", "bb1:14b.g", "", [ "$bb" ] ]
  ],
  "dependencies": {}
}
```

CONNECTION PROCEDURE:

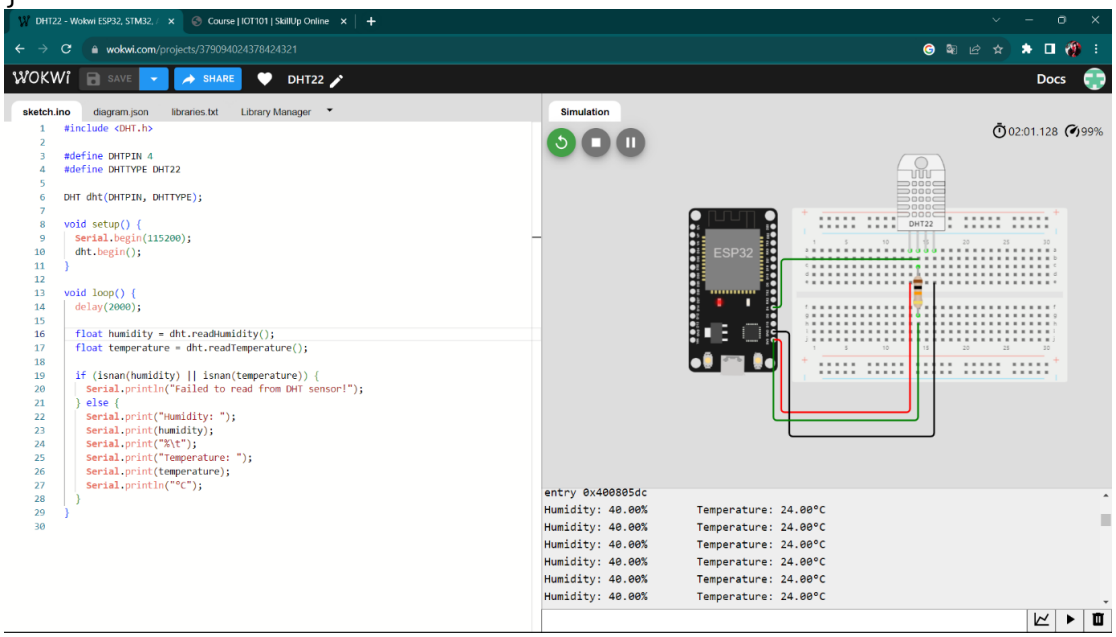
1. **Connection to ESP32:**
   - Connect the 3v3 pin of the ESP32 to the VCC pin of the DHT22 sensor and the GND pin of the both with the bread board in the positive and negative rails.
2. **Pull up resistor:**
   - Connect the pull up resistor of 10K ohm to the DHT22 sensor's digital data pin and connect it with the 3v3 pin of the ESP32 board for more accurate readings and transmission.

PROGRAM FOR SIMULATION:

As our project is based on IOT, we need to code the Instructions to the controller for performing our desired function. Here is the code for simulating our project. Before that, we have to ensure the required libraries are installed in wowki platform (DHT22)

```
#include <DHT.h>

#define DHTPIN 4
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
```

```
  dht.begin();
}

void loop() {
  delay(2000);

  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.print("%\t");
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.println("°C");
  }
}
```



Then we develop the transmission of the data to the cloud in phase 4.