Naan Mudhalvan Phase 3 Assessment

Course Name : Internet Of Things

Project Title : Environmental monitoring

Team Name : Tronixx

Team Members :

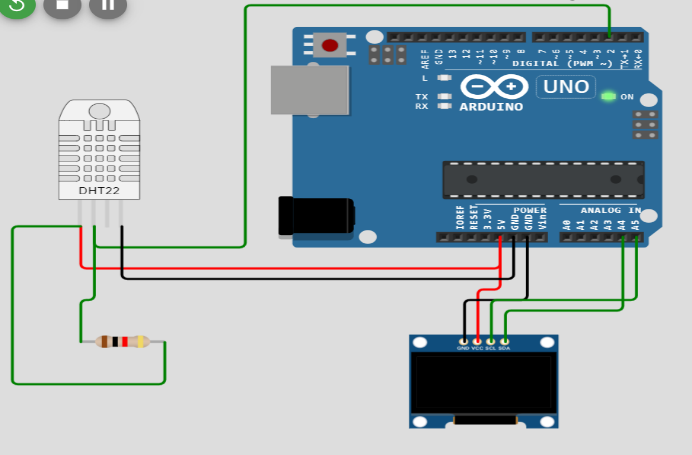
|  |  |  |
| --- | --- | --- |
| Name | Register number | Naan Mudhalvan ID |
| Thanush T | 721221106112 | au721221106112 |
| PradeepKumar M | 721221106068 | au721221106068 |
| Raja K | 721221106077 | au721221106077 |
| Ragupathi M | 721221106076 | au721221106076 |

Simulator used:  **Wowki Simulator**

Components used :

|  |  |  |
| --- | --- | --- |
| Component Name | Specifications | Quantity |
| DHT22 Sensor | - | 1 |
| Arduino UNO | - | 1 |
| OLED Display | SSD1306 | 1 |
| Resistor | 1k | 1 |

Circuit Diagram :



Working :

**Setup:**

In the setup() function, the Arduino initializes serial communication for debugging and sets up the DHT22 sensor.

It also initializes the OLED display and clears it.

**Loop:**

The loop() function continuously runs, collecting temperature and humidity data from the DHT22 sensor.

It clears the OLED display and then displays the temperature and humidity data.

The displayed information includes "Temperature" in degrees Celsius and "Humidity" as a percentage.

The data is updated every 2 seconds (delay(2000)).

**Display:**

The OLED display shows the words "Calculating..." when the program starts.

After 2 seconds, it clears the display and shows the actual temperature and humidity values.

The data is displayed on the OLED screen and updated in a loop.

**Summarization:**

**Data Collection:**

The DHT22 sensor is responsible for collecting temperature and humidity data. It offers accurate and reliable measurements, which are essential for environmental monitoring applications.

**Real-Time Display:**

The project displays real-time temperature and humidity data on the OLED screen. Users can easily observe the current environmental conditions without the need for external devices or connections.

**User-Friendly Interface:**

The OLED display provides clear and easy-to-read information. The data is presented in a user-friendly format, making it accessible to anyone without requiring specialized knowledge.

**Initialization Message:**

The project starts with an "Calculating..." message to indicate the initialization process. This brief message provides feedback to the user and ensures they are aware that the system is actively collecting data.

**Continuous Monitoring:**

The project continuously monitors and updates the displayed data at regular intervals (every 2 seconds in this case). This allows users to track changes in temperature and humidity over time.

**Simplified Circuit:**

The circuit design is straightforward, with minimal components, making it suitable for educational purposes and prototyping. The DHT22 sensor and OLED display are commonly used in environmental monitoring projects due to their ease of integration.

**Adaptability:**

The project can serve as a foundation for more advanced environmental monitoring systems. By expanding the code and hardware, additional sensors and features can be integrated to monitor other environmental parameters, such as air quality, light levels, or carbon dioxide levels.

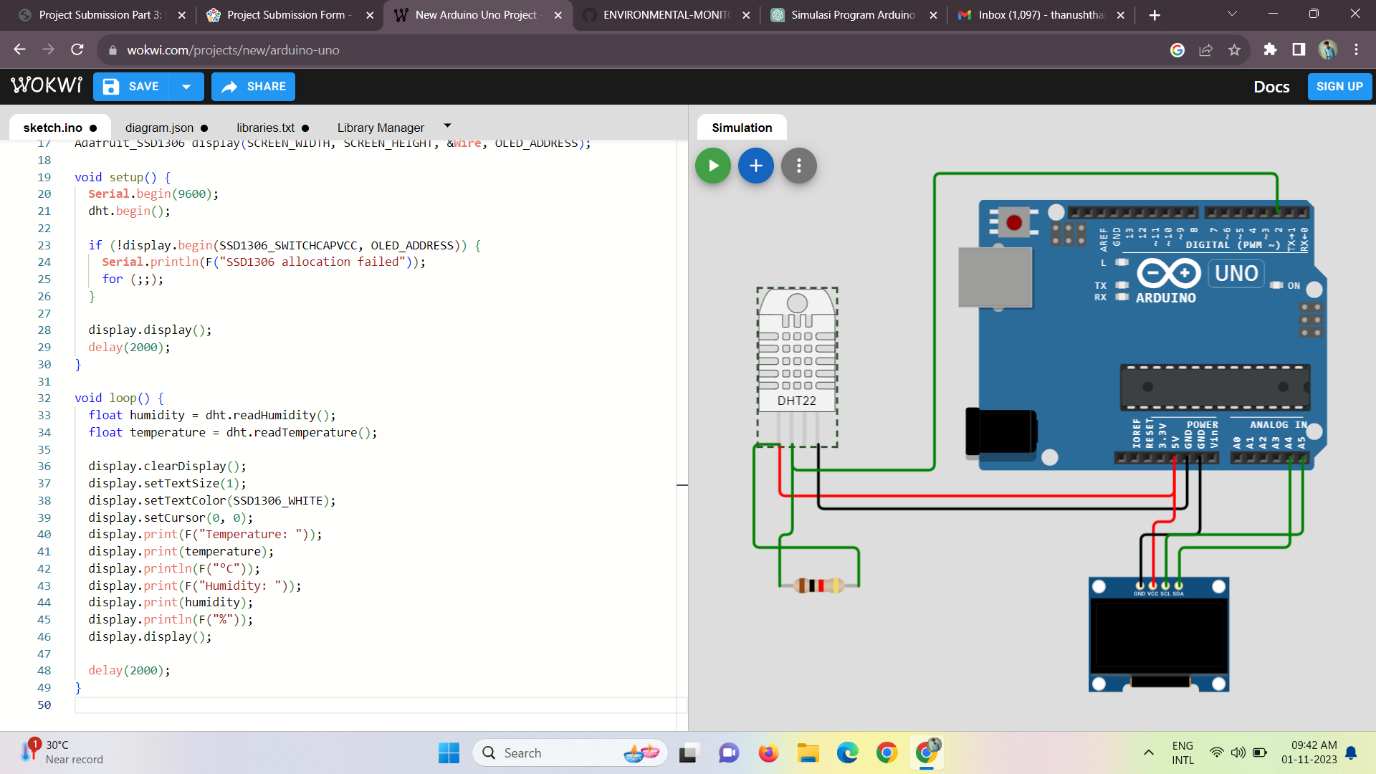
**Educational Value:**

This project is suitable for educational purposes and provides a hands-on introduction to Arduino programming, sensor usage, and data visualization on an OLED display.

**Conclusion :**

In conclusionthe environmental monitoring project offers a practical solution for monitoring and displaying temperature and humidity data. It showcases the capabilities of Arduino microcontrollers and the versatility of environmental sensors, making it a valuable learning tool for electronics and programming enthusiasts**.**

SIMULATED OUTPUT FOR ENVIRONMENTAL MONITORING USING WOWKI SOFTWARE



void loop() {

  // Display "Calculating" on the OLED display

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(SSD1306\_WHITE);

  display.setCursor(0, 0);

  display.print("Calculating...");

  display.display();

  float humidity = dht.readHumidity();

  float temperature = dht.readTemperature();

  // Display temperature and humidity

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(SSD1306\_WHITE);

  display.setCursor(0, 0);

  display.print(F("Temperature: "));

  display.print(temperature);

  display.println(F("°C"));

  display.print(F("Humidity: "));

  display.print(humidity);

  display.println(F("%"));

  display.display();

  delay(2000);

}

**Coding:**

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <DHT.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define DHTPIN 2

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// Change the I2C address to match your OLED display

#define OLED\_ADDRESS 0x3D

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &**Wire**, OLED\_ADDRESS);

void setup() {

**Serial**.begin(9600);

  dht.begin();

  if (!display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDRESS)) {

**Serial**.println(F("SSD1306 allocation failed"));

    for (;;);

  }

  display.display();

}