ENVIRONMENTAL MONITORING USING IOT

PHASE 3

Development part 1

INTRODUCTION:

Environmental monitoring using IoT provides data in real-time, operations and IT managers can proactively keep tabs on their equipment and processes — regardless of their location.

There are four essential components for IoT-based environmental monitoring to support critical insights and decision making:

- **Monitor the Environment:** Environmental condition monitors across fields, industrial sites and water management systems require installed sensors as well as an information delivery system, such as Digi XBee wireless communication modules and sensor connectivity gateways. These connected devices gather and deliver critical information exactly where it is needed.
- **Measure Data:** To measure environmental impact, these systems must make it possible to evaluate key data points that can indicate everything from water and chemical leaks to critical equipment failures. This data can be used by industrial operators and municipalities to measure their environmental footprint and take action to reduce waste, increase sustainability, manage valuable resources like water, and prevent environmental disasters.
- **Catalog Data:** The massive amounts of data collected from environmental monitoring stations around the globe cannot be overstated. There are global databases that catalog an enormous range of environmental data, such as the Microsoft Planetary Computer. Industrial sites and other enterprises, similarly, must utilize cloud and data center storage to catalog the gathered data for accessibility by business applications.
- **Provide Actionable Insights from the Data and Analysis:** The critical end game is actionable insights from data. Digi's IoT solutions, integrated with cloud applications like Microsoft Azure and Amazon Web Services, deliver data into complex software systems that enable personnel to gain those insights, get alerts and notifications, and take action.

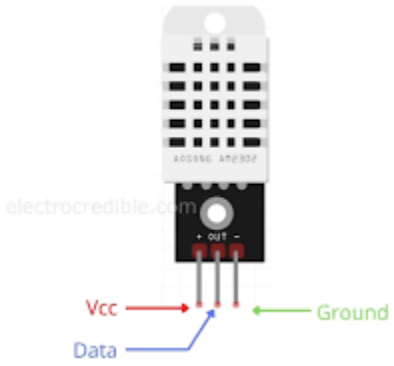
WOKWI:

Wokwi is an online Electronics simulator. You can use it to simulate Arduino, ESP32, STM32, and many other popular boards, parts and sensors.

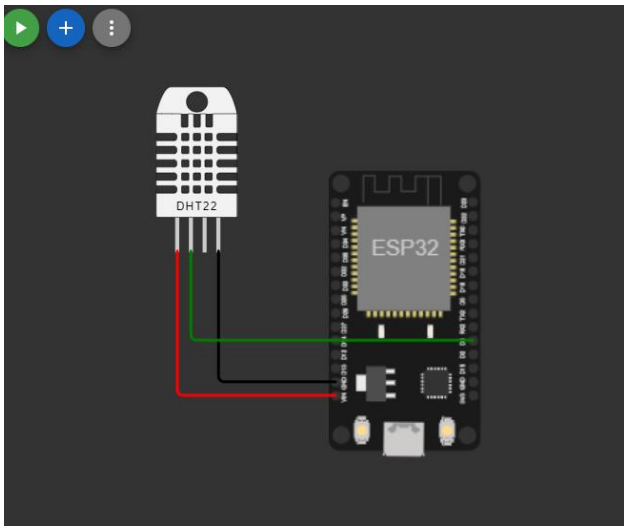
1. **Select the ESP32:**

    - In the "Select a board" section, type "ESP32" in the search bar and choose an ESP32 board model (e.g., ESP32 Dev Module).

2. **Add the DHT22 Sensor:**
    - In the components panel on the left, search for "DHT22"
    - Drag and drop the DHT22 component onto the breadboard area.



3. **Connect the Components:**
    - Connect the DHT22 sensor to the ESP32 as per your wiring instructions. You can click on the components to connect wires.
    - Connect the sensor's VCC and GND pins to the ESP32's 3.3V and GND, respectively.
    - Connect the sensor's data pin to a GPIO pin on the ESP32 (e.g., GPIO



4). Ensure the connections match your real-world wiring.

Write the Arduino Code:

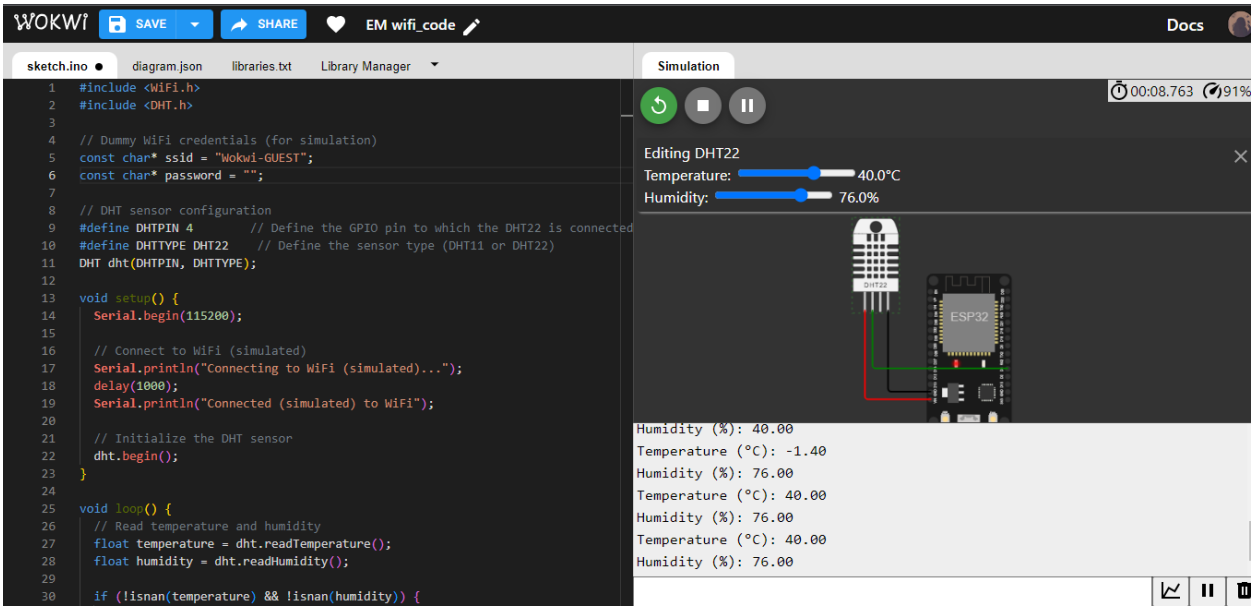
Code for Simulation of Wifi

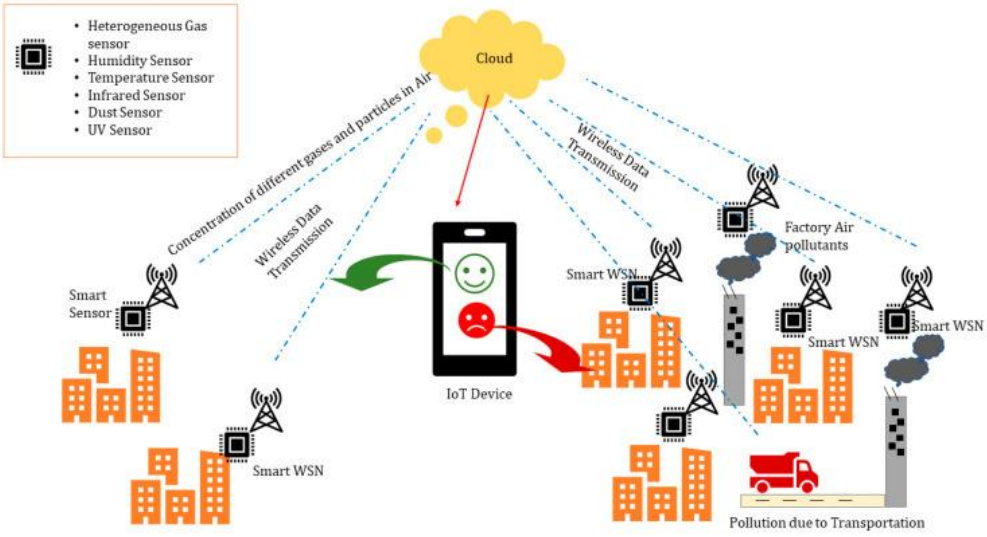
```
#include <WiFi.h>

#include <DHT.h>

// Dummy WiFi credentials (for simulation)
const char* ssid = "your_network_name";
const char* password = "your_network_password";

// DHT sensor configuration
#define DHTPIN 4        // Define the GPIO pin to which the DHT22 is connected
#define DHTTYPE DHT22    // Define the sensor type (DHT11 or DHT22)
DHT dht(DHTPIN, DHTTYPE);

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

  // Connect to WiFi (simulated)
  Serial.println("Connecting to WiFi (simulated)...");
  delay(1000);
  Serial.println("Connected (simulated) to WiFi");

  // Initialize the DHT sensor
  dht.begin();
}

void loop() {
  // Read temperature and humidity
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

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

  delay(2000); // Delay for 2 seconds before the next reading (adjust as needed)
}
```

OUTPUT ;

7. Run the Simulation:

- Click the "Run Simulation" button to start the simulation.
- The ESP32 code will run, and you'll see the temperature and humidity values printed in the Wokwi Serial Monitor.





CONCLUSION

This System monitors the changes happening over the environment and provides the sufficient ways for the users to access the information from anywhere through cloud. The temperature and humidity sensor will monitor and gives the details about the changes happening over the climate. The gas and sound sensor is used for monitoring the pollution over environment. The Monitored condition will be updated in the cloud.