**AIR QUALITY MONITORING USING IOT**

PHASE\_5 421221106302

**Design Thinking:**

* **Define:** 
  + - Clearly define the problem statement and the specific goals of the air quality monitoring system. Consider factors such as the pollutants to be measured, desired accuracy, and the target audience.
* **Ideate:**
  + - Brainstorm and generate ideas for the system's architecture, sensor selection, data communication, and user interface.
* **Prototype:** 
  + - Build a working prototype of the system, incorporating the selected sensors, IoT devices, and data communication modules. Test and refine the prototype based on user feedback and technical feasibility.
* **Test:**
  + - Deploy the prototype in real-world scenarios to gather data and evaluate its performance. Assess its accuracy, reliability, and usability. Make necessary adjustments and improvements based on the test results.
* **Iterate:**
  + - Continuously refine and enhance the system based on user feedback, emerging technologies, and changing requirements. Iterate through the design

process to ensure the system evolves and remains effective over time.

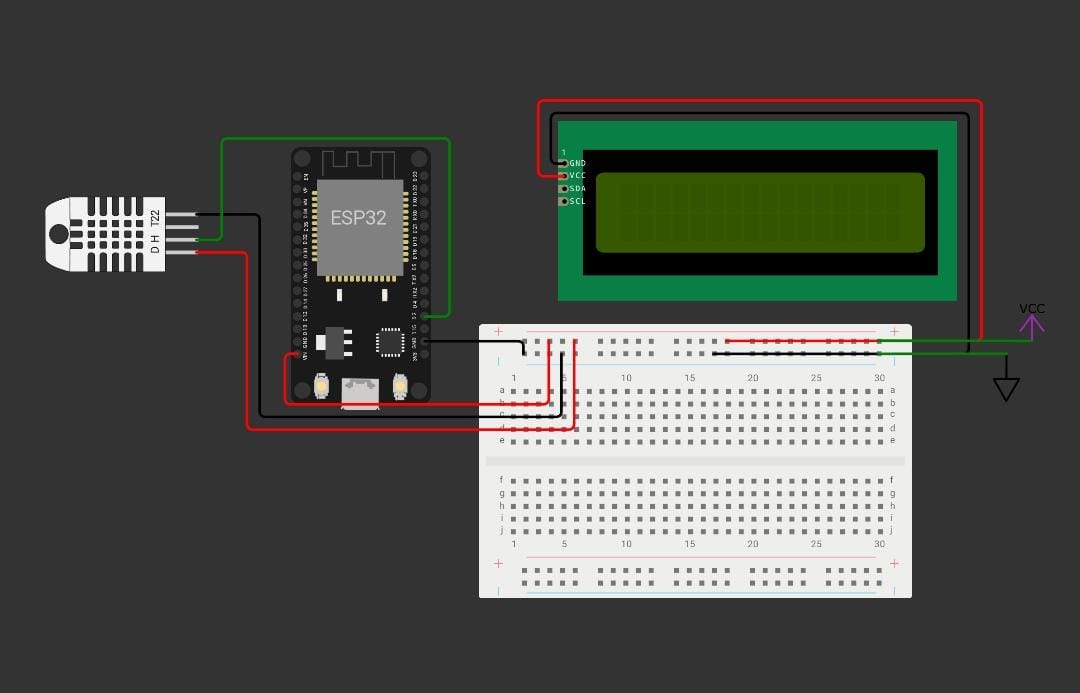
**Components:**

* **Air Quality Sensor:**
  + - Choose a sensor that can measure various air pollutant like particulate matter(PM2.5, PM10),gases (CO2),carbon (CO), (O3), (NO2),etc.
* **Microcontroller:**
  + - Select a microcontroller board like Arduino or Raspberry Pi to collect data from the air quality sensor and process it.
* **IoT Module:**
  + - Use an IoT module like ESP8266 or ESP32 to enable wireless communication and connect the system to the internet.
* **Power Supply:**
  + - Provide a stable power source for the system. You can use a battery or a power adapter depending on your requirements.
* **Display:** 
  + - Include an LCD or OLED display to show real-time air quality readings or a graphical representation of the data.
* **Connectivity**:
  + - Use Wi-Fi or Ethernet module to connect the system to the internet and enable remote monitoring and control.
* **Data Storage**:
  + - Incorporate an SD card or cloud storage service to store the collected air quality data for further analysis.
* **Enclosure:** 
  + - Design an enclosure to protect the components and provide a professional appearance.

**Process to Bulit:**

* **Choose the Sensors**: Select the appropriate sensors to measure air pollutants like PM2.5, CO2, CO, O3, NO2, etc.
* **Connect the Sensors**: Connect the sensors to a microcontroller board like Arduino or Raspberry Pi to collect data.
* **Set up IoT Connectivity:** Use an IoT module like ESP8266 or ESP32 to enable wireless connectivity.
* **Data Processing:** Program the microcontroller to process the sensor data and convert it into meaningful information.
* **Connect to the Internet:** Use Wi-Fi or Ethernet to connect the system to the internet for remote monitoring.
* **Data Transmission:** Send the processed data to a cloud platform or a server for storage and analysis.
* **Data Visualization:** Create a user-friendly interface to display real-time air quality readings and trends.
* **Alerts and Notifications:** Set up alerts and notifications to inform users about poor air quality conditions.
* **Data Analysis:** Use data analytics tools to analyse the collected data and identify patterns or trends.
* **Continuous Monitoring and Maintenance:** Regularly monitor the system's performance, calibrate the sensors if necessary, and ensure proper functioning.

Circuit Diagram:



**Steps for Coding:**

* 1. Connect the sensors to a microcontroller or single-board computer like Arduino or Raspberry Pi.
* 2. Use libraries or APIs provided by the sensor manufacturers to read data from the sensors.
* 3. Store the sensor data in a database or a file for further analysis.
* 4. Apply data processing techniques to clean and normalize the data.
* 5. Use programming languages like Python or R to analyze the data, calculate air quality indices, and generate visualizations.
* 6. Implement real-time monitoring by continuously collecting and processing data.
* 7. Display the air quality information on a web page or a mobile app for easy access.

**Program C coding:**

#define BLYNK\_TEMPLATE\_ID "TMPLwToQUqRw"

#define BLYNK\_TEMPLATE\_NAME "Air Quality Monitoring"

#define BLYNK\_AUTH\_TOKEN "C8Y7T0Fr54QF8pdfQ5dZsdfhhSdiQBFLj8mYe"

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <DHT.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

byte degree\_symbol[8] =

{

0b00111,

0b00101,

0b00111,

0b00000,

0b00000,

0b00000,

0b00000,

0b00000

};

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "SSID: Airtel-MyWiFi-AMF-311WW-943A"; // type your wifi name

char pass[] = ""; // type your wifi password

BlynkTimer timer;

int gas = 32;

int sensorThreshold = 100;

#define DHTPIN 2 //Connect Out pin to D2 in NODE MCU

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void sendSensor()

{

float h = dht.readHumidity();

float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

if (isnan(h) || isnan(t)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

int analogSensor = analogRead(gas);

Blynk.virtualWrite(V2, analogSensor);

Serial.print("Gas Value: ");

Serial.println(analogSensor);

// You can send any value at any time.

// Please don't send more that 10 values per second.

Blynk.virtualWrite(V0, t);

Blynk.virtualWrite(V1, h);

Serial.print("Temperature : ");

Serial.print(t);

Serial.print(" Humidity : ");

Serial.println(h);

}

void setup()

{

Serial.begin(115200);

//pinMode(gas, INPUT);

Blynk.begin(auth, ssid, pass);

dht.begin();

timer.setInterval(30000L, sendSensor);

//Wire.begin();

lcd.begin(16,2);

// lcd.backlight();

// lcd.clear();

lcd.setCursor(3,0);

lcd.print("Air Quality");

lcd.setCursor(3,1);

lcd.print("Monitoring");

delay(2000);

lcd.clear();

}

void loop()

{

Blynk.run();

timer.run();

float h = dht.readHumidity();

float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

int gasValue = analogRead(gas);

lcd.setCursor(0,0);

lcd.print("Temperature ");

lcd.setCursor(0,1);

lcd.print(t);

lcd.setCursor(6,1);

lcd.write(1);

lcd.createChar(1, degree\_symbol);

lcd.setCursor(7,1);

lcd.print("C");

delay(4000);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Humidity ");

lcd.print(h);

lcd.print("%");

delay(4000);

lcd.clear();

//lcd.setCursor(0,0);

// lcd.print(gasValue);

// lcd.clear();

Serial.println("Gas Value");

Serial.println(gasValue);

if(gasValue<1200)

{

lcd.setCursor(0,0);

lcd.print("Gas Value: ");

lcd.print(gasValue);

lcd.setCursor(0, 1);

lcd.print("Fresh Air");

Serial.println("Fresh Air");

delay(4000);

lcd.clear();

}

else if(gasValue>1200)

{

lcd.setCursor(0,0);

lcd.print(gasValue);

lcd.setCursor(0, 1);

lcd.print("Bad Air");

Serial.println("Bad Air");

delay(4000);

lcd.clear();

}

if(gasValue > 1200){

//Blynk.email("shameer50@gmail.com", "Alert", "Bad Air!");

Blynk.logEvent("pollution\_alert","Bad Air");

}

}

OUTPUT:

