**AIR QUALITY MONITORING USING IOT**

**PHASE 3**

**SENSOR DESIGN SIMULATION USING WOKWI**

**INTRODUCTION:**

In phase 3, we have interfaced the sensor with the microcontroller (ESP31). The simulation of the circuit was done using WOKWI simulator. Our project primarily focuses on Air quality and pc02 monitoring in the environment. The output is displayed using LCD display.

**SENSOR USED:**

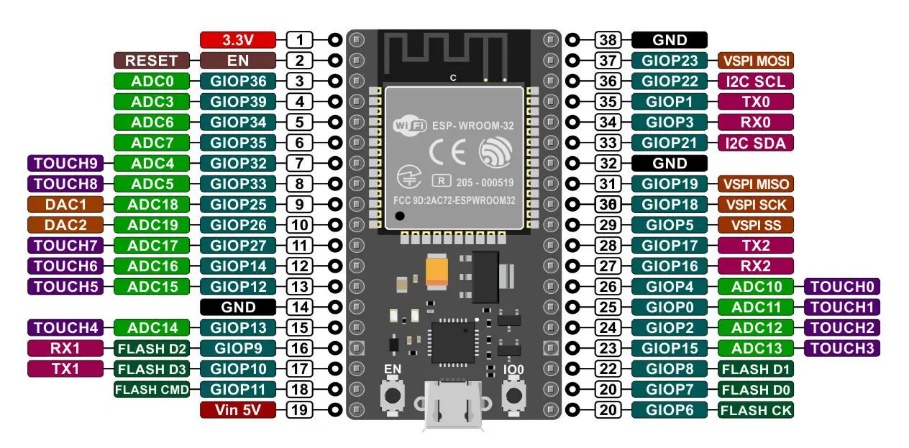
* **DHT22 SENSOR**
  + **DHT-** Digital Humidity and Temperature
  + It is low-cost digital sensor for sensing both pco2 and humidity.



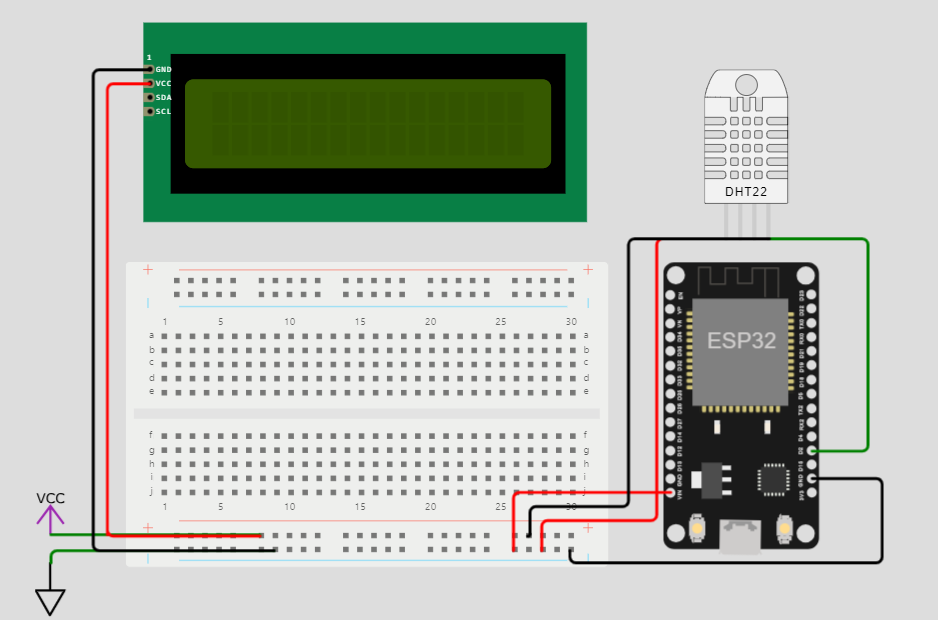
**MICRO-CONTROLLER USED:**

* **ESP32**
  + ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth.

**ESP32 PIN DIAGRAM:**



**CIRCUIT DESIGN:**

****

**PROGRAM: (**Using C)

#define BLYNK\_TEMPLATE\_ID "TMPLgwKssgggsnFXp"

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

#define BLYNK\_AUTH\_TOKEN "k03gT6nJosdsfsffesrJV\_S5SXEAdgdsdghhgPZvXEwSKDfj"

#define BLYNK\_PRINT **Serial**

#include <WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

//#include <Wire.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[] = "";  // type your wifi name

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

BlynkTimer timer;

int gas = A0;

int sensorThreshold = 100;

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

#define DHTTYPE DHT22

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();

//  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();

  if(gasValue<600)

  {

    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>600)

  {

    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 > 600){

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

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

  }

 }

**EXPLANATION:**

* The code includes two libraries, "DHTesp.h" for working with the DHT22 sensor and " BlynkSimpleEsp8266.h " for interfacing with the I2C-based LCD.
* The DHT\_PIN constant specifies the digital pin to which the DHT22 sensor is connected, and the **dhtSensor** object is an instance of the DHTesp library for interacting with the sensor.
* An LCD object is created, specifying the I2C address and the dimensions of the LCD.
* In the **setup** function:
  + It initializes the serial communication at a baud rate of 115,200 for debugging and monitoring.
  + It sets up the DHT22 sensor on the specified pin.
  + It initializes the LCD and turns on the backlight.
* In the **loop** function:
  + It reads temperature and humidity data from the DHT22 sensor using dhtSensor.getTempAndHumidity() and stores the values in the data object.
  + It prints the temperature and humidity to the serial monitor with one decimal place and degree Celsius symbol (°C).
  + It updates the LCD display with the temperature and humidity readings and adds "Wokwi Online IoT" text on the second line.
  + It then introduces a 1-second delay before repeating the loop.

The code continuously reads and displays pco2 level and po2 level data on the LCD while providing serial output for monitoring. The DHT22 sensor provides temperature in Celsius and relative humidity, and the LCD display shows this data along with some additional text.

**OUTPUT:**

