IOT Based Air & Sound Pollution Monitoring System Using Raspberry pi

MID COURSE PROJECT

BATCH 1(3)

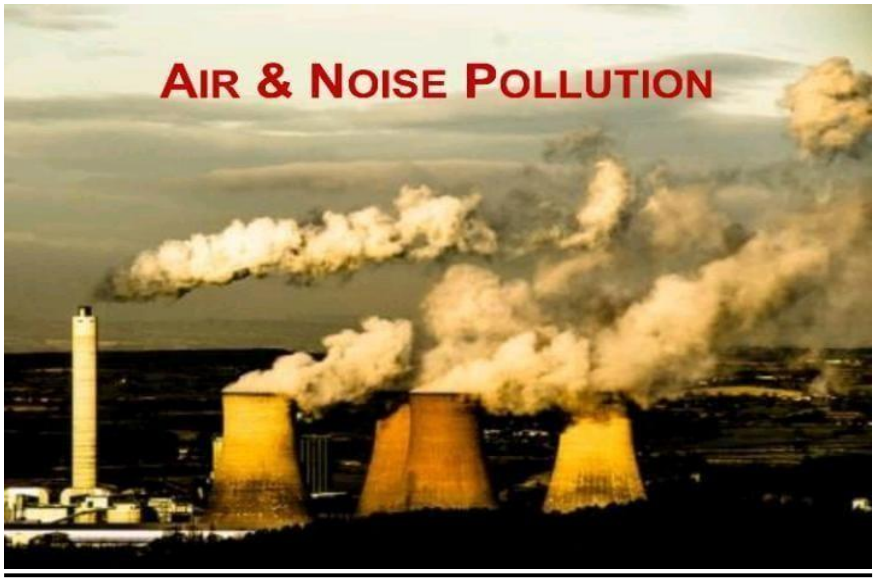
A. M. SRAVANTHI (21781A0405)

C. ABHINAYA (21781A0448)

D. SUSHMA (21781A0449)

Problem Statement:

With an ever-increasing number of industrial units and transport vehicles, the problem of air and sound pollution is becoming severe with each passing day. Therefore, it has now become extremely important to keep the air and noise pollution levels under check, especially important to keep the air and noise pollution levels under check, especially in urban centers. we have proposed Raspsberry Pi Air and Noise Pollution Monitoring System Over IOT to achieve this goal.



Scope of Solution:

To counter the problem of air and sound pollution monitoring, we propose a Raspberry Pi and IOT based air and sound pollution levels, the monitoring system project. To measure air pollution levels, the monitoring system employs a co2 sensor and a methane sensor. To measure noise pollution , the system uses a microphone. The measured sensor data is then transmitted to a remote web server via Wi-Fi connection the most significant aspect of this Raspberry Pi based Air pollution monitoring system is the IOT connectivity. It allows the remote and real-time monitoring of air and sound pollution levels. By effective monitoring of pollution levels , measure can be taken to bring those levels down

Required Components To Develop Solution:

To develop an IOT air and sound pollution monitoring system using pollution Raspberry Pi Pico for real-time data collection and remote accessibility, emphasizing accuracy, cost-effectiveness, and ease of development, following steps will guide to that development. This will ensure low-cost, real-time data transmission. Ease of development. we will focus on using MQTT protocal for communication with cloudbased IOT platform. Here is a step-by-step gide:

HARDWARE SETUP

1. Micro controller : Raspberry pi pico
2. LCD : 16*2 lcd

3.IOT module : ESP8266

4. Temp sensor : DHT11

5.Humility sensor : DHT11

6.Air quality sensor : MQ135

7.Sound sensor : ADC Mic with amplifier circuit

8. LEDs : Red and Green LED

9.Adaptor

10. Power Source : 12v 1amp DC

SOFTWARE SETUP

1.Arduino IDE

2.Proteus based circuit diagram

Description of the block diagram:

The block diagram of the Raspberrry Pi and IOT based air and sound pollution monitoring project consists of 6 sub-blocks.

1.Raspberry Pi:

It is the most important part of the IOT air and sound pollution monitoring system. All three sensors send their data to the Raspberry Pi single-board computer. The sensors are the input devices that send their data to the Raspberry Pi controllers. The output of the Raspberry Pi controller is connected to two output blocks. These output blocks are the IOT web server block and the LCD display screen block.

2.LCD display:

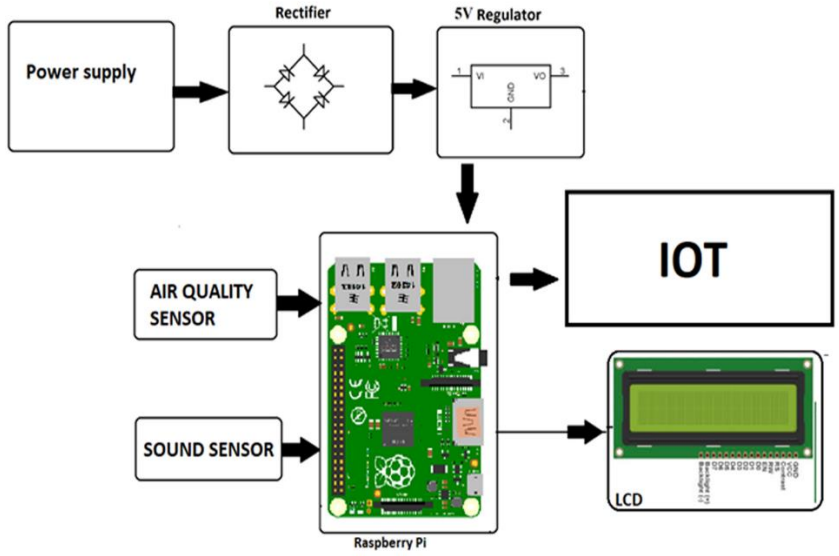
Raspberry pi processes the data from the sensors. Then it is displayed on their LCD output device.

3.IOT web server:

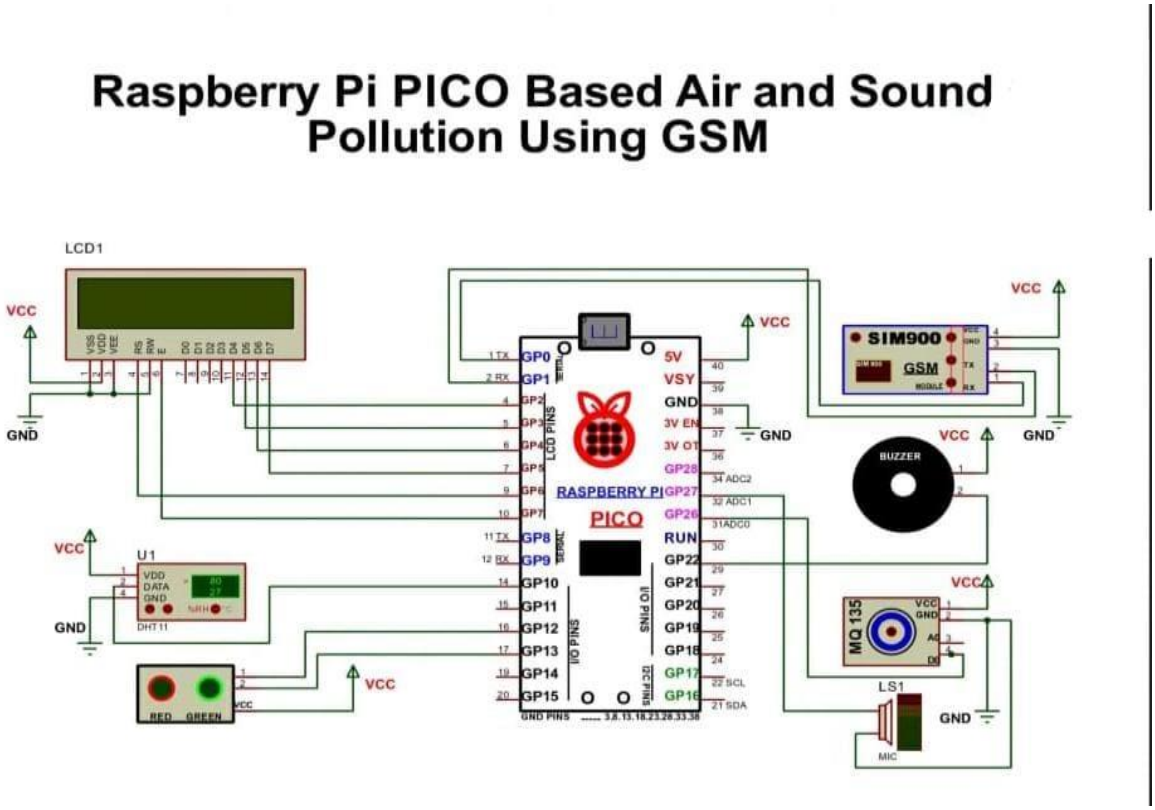
the same air pollution sensor data which is displayed on the LCD is also communicated to the remote web server. This is implemented by means of onboard Wi-Fi of the Raspberry Pi board. This sensor data can then be monitored from a remote location using the IOT website.

BLOCK DIAGRAM:

Block Diagram:



SIMULATED CIRCUIT:



CODE FOR THE SOLUTION

```
import machine import network from
umqtt.simple import MQTTClient import
time

# Define pin numbers for sensors
AIR_SENSOR_PIN = 0      # Example pin number for air pollution sensor
NOISE_SENSOR_PIN = 1 # Example pin number for noise pollution sensor
# Define Wi-Fi credentials
WIFI_SSID = "your_wifi_ssid"
WIFI_PASSWORD = "your_wifi_password"

# Define MQTT broker settings
MQTT_SERVER = "mqtt.server.com"
MQTT_PORT = 1883
MQTT_TOPIC = "environment/pollution"

# Function to connect to Wi-Fi
def connect_wifi():
```

```
    wlan =
    network.WLAN(network.STA_IF)
    wlan.active(True) if not wlan.isconnected():
        print("Connecting to WiFi...")
        wlan.connect(WIFI_SSID, WIFI_PASSWORD)
        while not wlan.isconnected():
            pass
    print("Connected to WiFi:", wlan.ifconfig())

# Function to read sensor data def
read_sensor(pin): adc =
machine.ADC(pin)
sensor_value = adc.read() return
sensor_value


# Function to connect to MQTT broker def
connect_mqtt():
    client = MQTTClient("raspberry_pico", MQTT_SERVER,
MQTT_PORT)
    client.connect()
    return client

# Main function
def main():
```

```
    connect_wifi() mqtt_client = connect_mqtt()

    while True: # Read sensor data air_pollution =
       read_sensor(AIR_SENSOR_PIN) noise_pollution =
       read_sensor(NOISE_SENSOR_PIN)

       # Create data payload
       data = {
          "air_pollution": air_pollution,
          "noise_pollution": noise_pollution
       }
       # Publish data to MQTT broker
       mqtt_client.publish(MQTT_TOPIC, str(data))
       print("Data published:", data)

       # Wait for some time before reading again (adjust as needed)
       time.sleep(60) # Read data every 60 seconds

# Run the main function if
_name_ == "_main_":
main()
```

CONCLUSION:

Using Raspberry pi pico, we will read noise and sound pollution level and updated to IOT cloud server. Along with these two sensors we used DHT11 sensor to read temperature and Humidity. Microphone used to detect sound pollution and MQ 135 used to detect air
pollution. Raspberry pi pico displays sensor data on lcd and then send to EAP8266 module. ESP 8266 send data to IOT cloud server. Arduino based firmware used to perform all required conditions.

BATCH 1(3)
A. M. SRAVANTHI (21781A0405)
C. ABHINAYA (21781A0448)
D. SUSHMA (21781A0449