

## PROJECT SUBMITTED BY:

NAME: NITHYANANDAM. K

REGISTER NO: 713921106034

TOPIC: NOISE POLLUTION MONITORING

MAIL ID: nithyanandamnithi600@gmail.com

NM ID:au713921106034

COLLEGE CODE: 7139

## DEVELOPMENT OF NOISE POLLUTON MONITERING DEVICE USING IOT

### Requirements:

- Determine the specific goals of your noise pollution monitoring system (e.g., measuring sound levels in a specific area, identifying noise sources).
- Choose appropriate sound sensors (microphones or sound level sensors) based on your requirements.
- Decide on the communication protocol (Wi-Fi, LoRa, GSM) and the platform for data storage and analysis.

### Hardware:

- **Sensors:** Choose sound level sensors capable of capturing the range of noise levels you want to monitor. Calibrate the sensors for accurate readings.
- **Microcontroller:** Use a microcontroller board like Arduino or Raspberry Pi to interface with the sensors and transmit data.
- **Communication Module:** Select a communication module compatible with your microcontroller and the desired range of data transmission.
- **Power Supply:** Design a power supply system (battery or mains power) based on the deployment location.

## PYTHON SCRIPT

```
import spidev

import time

import requests

SPI_PORT = 0

SPI_DEVICE = 0

API_ENDPOINT = "https://example.com/api/noise-level"


def read_noise_level(channel):

    spi = spidev.SpiDev()

    spi.open(SPI_PORT, SPI_DEVICE)

    adc_data = spi.xfer2([1, (8 + channel) << 4, 0])

    noise_level = ((adc_data[1] & 3) << 8) + adc_data[2]

    spi.close()

    return noise_level


# Main function

def main():

    try:

        while True:

            # Read noise level from MCP3008 ADC (channel 0 in this case)

            noise_level = read_noise_level(channel=0)

            print("Noise Level: {}".format(noise_level))

            payload = {

                "noise_level": noise_level
```

```

    }

    response = requests.post(API_ENDPOINT, json=payload)

    if response.status_code == 200:

        print("Data sent successfully!")

    else:

        print("Failed to send data!")

    time.sleep(60)

except KeyboardInterrupt:

    print("Monitoring stopped by the user.")

if __name__ == "__main__":

    main()

```

## Conclusion and Future Improvements:

- **Benefits:** IoT-based noise pollution monitoring systems offer valuable insights for urban planning, environmental research, and public health, aiding in noise pollution mitigation strategies.
- **Continuous Improvement:** Regular calibration, software updates, and feedback-driven enhancements are essential to maintain the accuracy and reliability of the monitoring device.
- **Community Engagement:** Involving communities and local authorities can lead to collaborative efforts in addressing noise pollution issues and implementing effective solutions.