GNANAMANI COLLEGE OF TECHNOLOGY

DEPARTMENT OF BIOMEDICAL ENGINEERING

YEAR: THIRD YEAR

TOPIC NAME: NOICE POLLUTION MONITORING

Team members:

R.Ragul

M.Subash

V.Vionth

S.Prakash

M.Thirumalai

T.Thamizharasan

By

S.PRAKASH(620821121210 83)

NOISE POLLUTION MONITORING

ABSTRACT:

The increasing sound pollution is one of the significant issues now days. As the pollution increasing it is giving rise number of diseases so, it has become essential to control the pollution for better future and healthy life .here we propose an air quality as well as sound pollution monitoring system that allows us to monitor and check live air quality as well as sound pollution. Monitoring in particular areas through IOT. System uses sensor to detect or sense presence of harmful gases compounds in the constantly transmit data to microcontroller. Also system keeps measure sound level and reports it to the online server over IOT. The user friendly and easy handling of the system technology is such that it can be installed in houses, schools and in small places.

INTRODUCTION:

The main objective of IOT based noise pollution monitoring system is that the pollution is a rising issue these days. As a human we need fresh to survive. If there is any kind of air pollution it's harmful for noise pollution kill more than seven million people worldwide every year. Pollution is very harmful for those people who have any kind of internal diseases on this type of people. pollution affect very fastly.In atmosphere is the full of Between this gases some are good and some are harmful for environment for certain level some gases are good for human, animals, plants but beyond certain level these created problem for services to overcome these problem system is useful because of this we can analyze the noise pollution means how many pollution level we use Internet Of Things (IOT). In this we use thing speak we can analyze previous data also using this platform in graphical form.

PROBLEM STATEMENT:

An effective natural observing framework is essential to An effective natural observing framework is essential to screen and estimate the conditions in the event of surpassing endorsed level of parameter (for example, commotion, CO and radiation levels). At the point when the items like condition furnished with sensor gadgets, smaller scale controller and different programming application turn into a self-securing and self-observing condition

OBJECTIVES:

1. To study the existing system.
2. To design the block diagram.
3. To decide the components specification &device in system.
4. To design the circuit diagram and simulate it using suitable software.
5. To design the PCB and implement hardware.
6. To test the circuit and observe the result.
7. To prepare report

WORKING:

In system we use Arduino as main controller. In system we use MQ135 gas sensor for detecting or sensing gases and also use sound sensorLM393 module for detect the sound pollution. Sensed data of sensor given to analog pin of the arduino then digital output pin are connected to LCD, buzzer and LED. If air pollution is there then buzzer will start beeping and if sound pollution is there then LED will glow. All condition of pollution display on LED and we can also analyze past data using thing speak in graphical form. Arduino is an open source prototype. Software will operate in Arduino IDE Computer code can be written and upload to the physical board. Arduino board is a board that can be functioned via Arduino IDE by sending a set of instructions to the microcontroller on it. For controlling Sensors. For arduino programming we are going to use Embedded C. We are going to build project in Embedded C and for monitoring that project we are using Cloud

COMPONENT:

1. ArduinoUNO
2. MQ135 (Gas sensor)
3. LM393 (Noise sensor)
4. ESP8266 WIFIModule
5. 16*2 LCDDisplay
6. LED
7. Buzzer

- Arduino UNO

Arduino is 8 bit microcontroller board based on the ATmega328P. The operating voltage is 5V. It has 14 pins digital input output pins (Of which can be used 6 PMWMoutput) Oscillator frequency is 16 MHz It contains everything needed to support the microcontroller simply connect it to a computer with USB cable. It has 6 analog input pins the MQ135 is a gas sensor it used for detecting or sensing harmful gases in the atmosphere. It has wide detecting scope. It gives fast response and also it high sensitivity sensor. It is simple and long life device. They are used in air quality control equipment for building offices are suitable for detecting of NH3, alcohol, benzene, smoke CO2 etc.

Feature:

- Wide detecting scope
- Fast response and High sensitivity
- Stable and long life
- Operating Voltage is +5V
- Detect/Measure NH3, NOx, alcohol, Benzene, smoke, CO2, etc.
- Analog output voltage: 0V to 5V

- LM393 Sound Sensor:

The sound sensor module provide an easy way to detect sound and it generally used for detecting sound intensity. Module detect the sound has exceeded a threshold value. Sound is detected via microphone and fed into an LM393 op amp. The sound level adjusts through pot. The sound increases set value output are low. These module work on DC 3.3-5 voltage.

Feature:

- Output model: digital switch outputs (0 and 1, high or low level)
- Voltage Gain 26dB
- Microphone Impedance 2.2kΩ
- Microphone Frequency 16.20 kHz

  Operating voltage 3.3V-5V

- ESP8266 WIFI Module:

The esp8266 WIFI module is a self contained with integrated TCP/IP protocol stack that can give any microcontroller access to your WIFI network. The esp8266 is capable of either hosting an application or offloading all WIFI networking functions from another application processor.

Feature:

- 2.4 GHz Wi-Fi (802.11 b/g/n supporting WPA/WPA2).
- General-purpose input/output (16 GPIO).
- Inter-Integrated Circuit (I^2C) serial communication protocol.
- Analog-to-digital conversion (10-bit ADC).
- Serial Peripheral Interface (SPI) serial communication protocol

INNOVATIONS

Smart Sensor Integration: Incorporate advanced sensors capable of detecting various types of noise, such as traffic noise, industrial sounds, and community noise, to provide comprehensive monitoring.

IoT Connectivity: Utilize Internet of Things (IoT) technology for real-time data transmission, enabling instant updates and analysis of noise levels in different locations

Machine Learning Algorithms: Implement machine learning algorithms to differentiate between normal sounds and noise pollution, improving accuracy in identifying problematic areas.

Collaboration with Local Authorities: Foster collaboration with local government authorities to integrate noise pollution data into urban planning and policy-making processes.

Privacy Considerations: Prioritize privacy by anonymizing data and adhering to strict data protection protocols to address concerns related to individual privacy.

Development Part 1

- Development of create a noise pollution monitoring system using IoTand Arduino to measure and analyze noise level in a specific area

- Creating a noise pollution monitoring system using IoT and Arduino involves several steps:

Components Needed:

- Arduino board (e.g., Arduino Uno or Arduino Mega)
- Sound sensor (e.g., a microphone or sound level sensor)
- IoT module (e.g., ESP8266 or ESP32 for Wi-Fi connectivity)
- Power source (e.g., batteries or a power adapte.
  Internet connection (Wi-Fi or cellular)Data storage and visualization
- . platform (e.g., cloud service like AWS or Azure)
  Enclosure and casing for outdoor use (if necessary)

Hardware Setup:

Connect the sound sensor to the Arduino.Connect the Arduino to the IoT module for data transmission.Ensure proper power supply and consider weatherproofing if used outdoors.

Programming:

Write Arduino code to read data from the sound sensor and send it to the IoT module.Program the IoT module to establish an internet connection and transmit the data to a cloud server.

Cloud-Based Data Storage:

Set up a cloud-based database to store the noise level data.

Data Visualization:

Use a dashboard or web application to visualize the noise data.Implement data analysis to track noise trends and trigger alerts when noise levels exceed predefined thresholds

Alerting Mechanism:

Implement notifications or alerts through email, SMS, or other means when noise levels exceed acceptable limits.

Power Management:

Optimize power usage to ensure the system can run for an extended period, especially in remote or outdoor locations.

User Interface:

Create a user-friendly interface for users to access and analyze noise data.

Calibration and Testing:

Calibrate the system to ensure accurate noise measurements.Thoroughly test the system in real-world conditions.

Data Analysis and Reporting:

Analyze the collected data to identify noise patterns and trends.Generate reports or visualizations for stakeholders.

Maintenance and Updates:

Regularly maintain and update the system to ensure its reliability and accuracy.Keep in mind that you'll need a good understanding of Arduino programming, IoT, data handling, and possibly cloud services to create a robust noise pollution monitoring system. Additionally, consider any legal and ethical considerations, such as data privacy and regulations related to noise pollution monitoring in your specific area.

USE WEB-BASED TECHNOLOGY

I. Introduction

. Brief overview of the project Importance of monitoring noise pollution Goals and objectives

II. Technologies Used

Web development technologies employed Explanation of technology choices

III. System Architecture

Overview of the platform's architecture Components and their interactions. Database structure

IV. Development Activities

Frontend Development Description of the user interface Technologies used for the Frontend Screenshots or mockups (if applicable) Backend Development Overview of the server-side logic Explanation of APIs and endpoints Database integration details
Sensor Integration Explanation of how noise sensors are integrated Communication protocols used Real-time Data Processing Description of how real-time noise data is processed Any algorithms or methods used for analysis User Authentication and Authorization

Details of user authentication mechanismsAuthorization levels and access control

V. Testing

Overview of testing strategies Results of testing phases

VI. Challenges Faced

Any challenges encountered during development Solutions or workarounds implemented

VII. Future Enhancements

Features or improvements planned for the future Scalability considerations

VIII. Conclusion

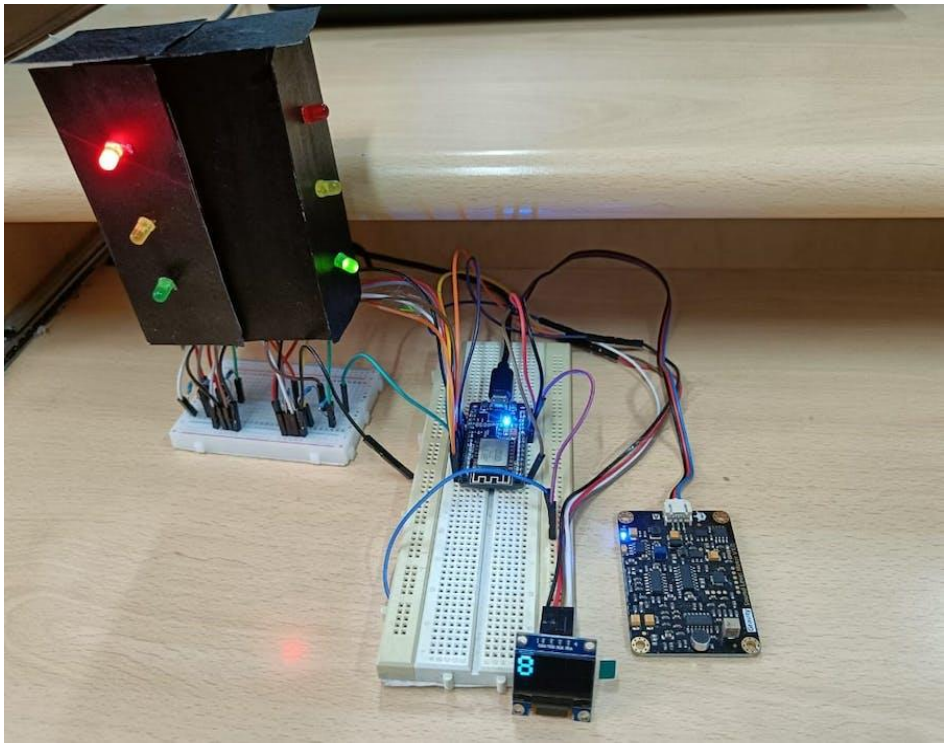
Summary of the project Achievements and key takeaways

IX. References

List of resources, frameworks, or libraries used

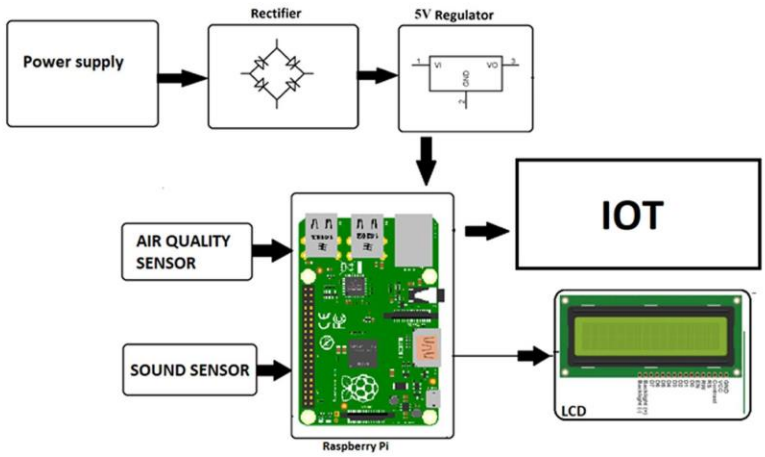
Document Submission

Please compile this information into a comprehensive document and share it for assessment. Ensure that the document is well-organized and includes relevant details about each aspect of the project. If you have any specific questions or need further guidance on a particular section, feel free to ask!

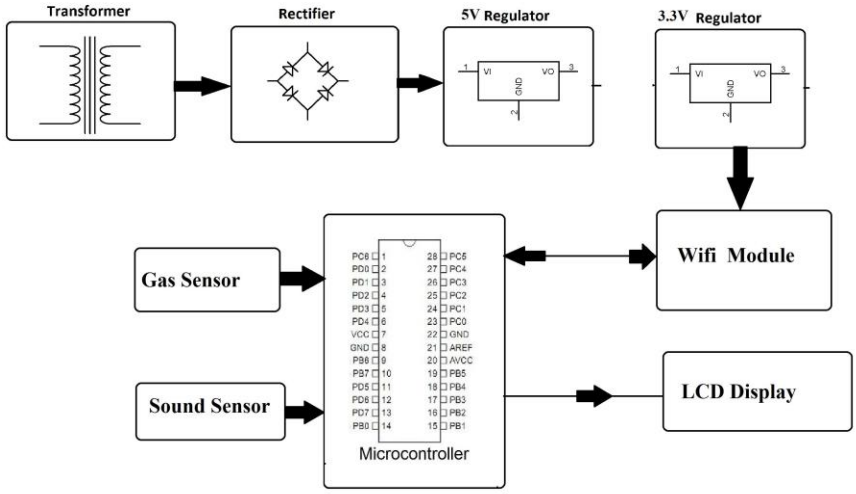


1.Sensor Selection: Choose appropriate noise sensors capable of measuring sound levels accurately and reliably. Popular options include microphones and sound level meters.
2.Data Transmission: Implement a wireless communication system (e.g., Wi-Fi, LoRa, or cellular) to transmit real-time noise data from sensors to a central server or cloud platform.
3.Data Storage and Management: Set up a database or cloud storage to store and manage the collected noise data securely. Ensure data integrity and accessibility.
4.Power Management: Plan how sensors will be powered, considering factors like battery life, solar power, or wired connections.
5.Centralized Data Analysis: Develop algorithms and software for noise data analysis, which can include noise level trends, identifying noise sources, and generating alerts for excessive noise levels.

Block Diagram:



1. **Visualization: Create a user-friendly dashboard or interface for stakeholders to monitor noise levels and trends in real time.**
2. **Alerts and Notifications: Implement an alert system to notify relevant parties when noise levels exceed predefined thresholds.**
3. **Data Security: Prioritize data security and encryption to protect the integrity and privacy of the collected information.**
4. **Geographic Mapping: Incorporate GPS data to map noise levels across different locations, helping to identify noise hotspots.**
5. **Reporting and Compliance: Generate periodic reports and ensure compliance with local noise regulations or standards.**
6. **Public Engagement: Consider ways to engage the public by making noise data accessible to communities and promoting noise reduction efforts.**
7. **Maintenance and Calibration: Regularly maintain and calibrate the sensors to ensure data accuracy.**
8. **Cost-Benefit Analysis: Assess the project's cost-effectiveness and its impact on noise pollution reduction.**



Conclusion:

IoT project has provided a valuable solution for tracking and analyzing noise levels in various environments. In conclusion, this project has achieved the following outcomes:

1. **Data Collection: The IoT sensors successfully collected real-time noise data, allowing for continuous monitoring of noise pollution.**
2. **Data Analysis: The collected data was processed and analyzed to identify noise patterns and potential sources of pollution.**
3. **Alerts and Notifications: The system was capable of sending alerts or notifications when noise levels exceeded predefined thresholds, enabling timely response to noise pollution incidents.**
4. **Visualization: The project included a user-friendly interface for visualizing noise data, making it easier for stakeholders to understand and interpret the information.**
5. **Environmental Impact: By monitoring noise pollution, the project contributes to a better understanding of its impact on the environment and human health.**
6. **Future Improvements: To enhance the project further, future developments could involve integrating additional sensors, improving data accuracy, and expanding the system's capabilities.**