NOISE POLLUTION MONITORING USING IoT

TEAM MEMBER

au820421106033 : NANDHARAJAN K , B.E(ECE)

Phase 1 Document Submission

Project: Noise Pollution Monitoring

OBJECTIVE:

Define objectives such as real-time noise pollution monitoring, public awareness, noise regulation compliance, and improved quality of life.

Phase 1: Project Definition and Design Thinking

1.Project Definition

Abstract.

Presently, noise pollution has become a very big issue around the world. The adverse effects of this pollution include hearing impairment, negative social behaviour, annoyance, sleep disturbance and intelligibility to understand people's speech.

In learning context, noise can affect understanding and behaviour of people and places with high noise level are not suitable for learning and teaching process. Internet of Things (IoT) technology is one of the best choices to monitor the noise or sound intensity in the environment for the safety of human being.

The aim of this paper is to deliver a development of an IoT based noise monitoring system comprises of a sound sensor, an IoT platform called NodeMCU, LCD and LEDs. The system will provide a real-time alert if the noise exceeds the threshold noise limit set by Environmental Department of Health standard.

Equipped with an Android application, the data from the sound sensor will be transferred into the cloud server and subsequently transferred into the app for display and to enable remote monitoring.

The sound level is measured for two different days during weekend and weekday. Based on Charted Institution of Building Service Engineers (CIBSE), 60dBA is the permissible ambient level and any readings that above 60dBA can interrupt speech intelligibility. From the research, the suitable time for students to study for weekend is all day starting from morning until midnight. As for weekday, the most suitable time to study is during midnight. These justifications are made based on the readings of the sound level.

2.IoT Sensor Design

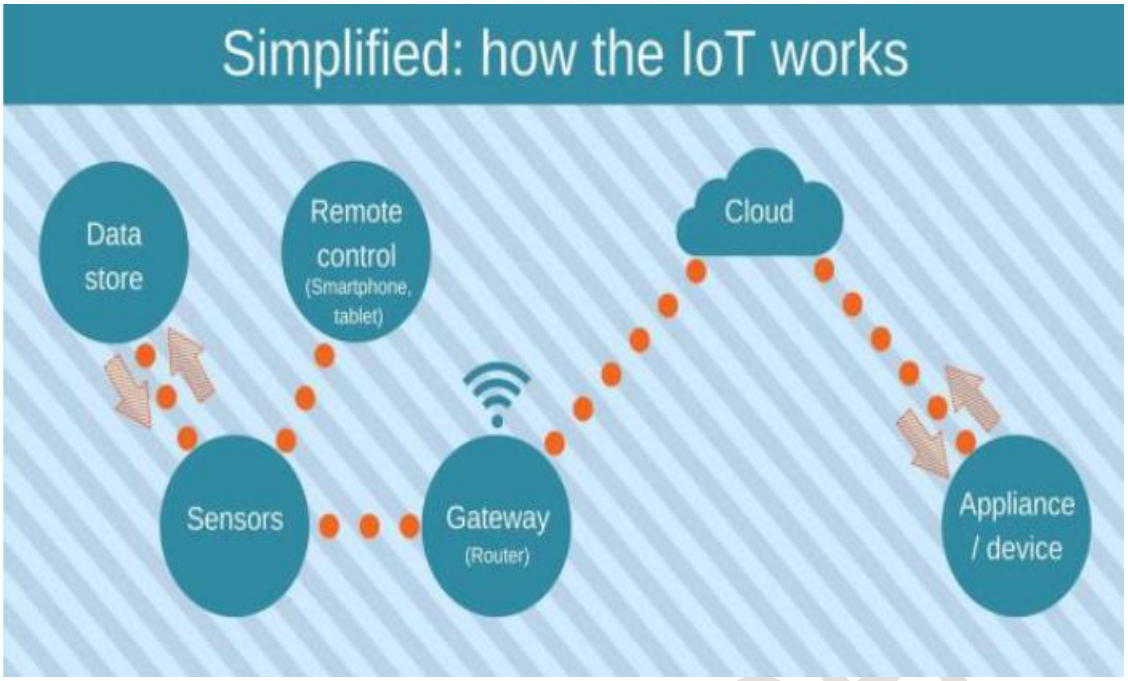


Figure 1. Working principle of IoT

For the hardware parts, LM 393 sound sensor is used to read the readings of the sound level from the environment. The reading of sound sensor is calibrated using the real sound level meter to get the accurate readings of the sound level.

The 16x2 LCD will show the values of sound level at that researched area and give the warning that says the level of sound is high when the measurement exceeds the set value. If the users could not read the readings due to poor eyesight, they can know the level of sound by using the light emitting diodes (LED) which in red, blue and green colour placed below the LCD.

LED acts as an indicator to indicate when the noise is very high. It will turn to red, blue for low noise while green for intermediate level. All these components such as sound sensor, LCD, and LEDs will be connected to the ESP8266 NodeMCU.

Figure 2. Prototype of the Project

Components Required:

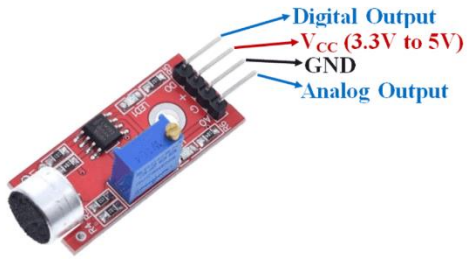


Figure 3. LM393 Noise Sensor

Figure 4. 16x2 LCD Display

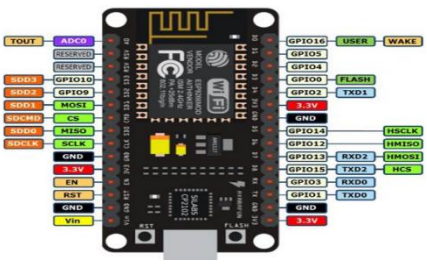


Figure 5.ESP8266NodeMCU

3.Noise Pollution Information Platform:

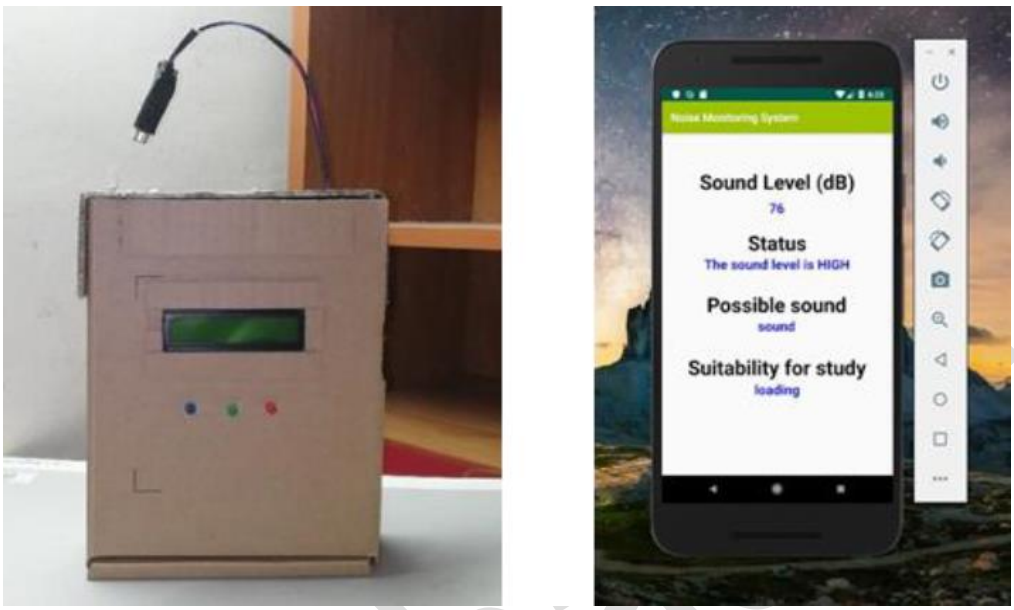


Figure 6. Prototype of the Project and interface of the app on mobile smartphone

3.1 App Development

As the app was created by using Android Studio, the app will display the data taken from the sound sensor. Android Studio is a software to create app use JAVA language to design an Android development. The app has four features which are the reading of sound level in dBA, the level of warning based on the reading of sound intensity, the possible sound that contributes to the sound level and the suitability for students to study. The app gives different level of warning such as “low”, “normal”, “high” and “very high”.

4. Integration Approach:

The prototype is calibrated using actual sound level meter to get the accurate measurement of sound level or sound intensity.
The prototype is used to measure the sound level at five different times which are during morning, afternoon, evening, night and midnight at a place where UTM students are staying. The reading is taken 30 minutes per range of time. Within that time, the lowest and highest sound intensity were recorded and from the readings, the range of sound level was determined during that specific range of time.

Table 1. Data Analysis from Prototype.

Time	Weekend (Saturday) (dBA)	Weekday (Sunday) (dBA)	Allowable Noise Level according to Environmental Department of Malaysia (dBA)	Permissible Level of Comfort according to CIBSE (dBA)
Morning 7.00am-12.00pm	47 – 60	57 – 71	55	60
Afternoon 12.01pm-14.00pm	43 – 49	54 – 69	55	60
Evening 2.01pm-7.00pm	43 – 49	62 – 69	55	60
Night 7.01pm-12.00am	42 – 59	48 – 63	55 (7.01pm–10.00pm) 45 (10.00pm– 12.00am)	60
Midnight 12.01am-6.59am	34 – 35	34 – 35	45	60

5. Conclusion

People thought that noise pollution is merely an annoyance but it is actually very important to monitor noise level because according to research, people who are exposed to noise for a long duration of time can have hearing loss, sleep disturbance, high blood pressure and injuries [16].
Besides, it can affect the learning process of people in terms of understanding and behaviour. Thus, this research investigates and subsequently proposes the suitable time for students to study by utilising the cloud server and android application to realize an IoT based noise monitoring system.
From the prototype, it also can be determined the dominant sound that increases the noise level in the researched area. The app can display the reading from the prototype successfully.
Based on the results, it can be concluded that the students can study throughout the day starting from morning until midnight during weekends because the noise level is still under the allowable standard which is 60dBA according to CIBSE.
As for weekdays, the suitable time to study is found out to be at midnight because the readings are below 60dBA for that time. The limitation of this study is that it is conducted within UTM campus only.
In addition, the app can only show the reading from only a single prototype as well as the prototype and app can only operate when there is an internet connection. This system can be improvised in future to include measurements outside UTM such as at schools and airports.

NOISE POLLUTION MONITORING USING IOT

820421106033: NANDHARAJAN K, B.E/ECE

PHASE- 4 : **DEVELOPMENT PART - 2**

INTRODUCTION:

Noise monitoring refers to the systematic process of measuring, recording, and assessing sound levels in various environments to understand the extent of noise pollution and its potential impact on human health and the surrounding ecosystem. It involves using specialised equipment to gather data, which is then analysed to make informed decisions about noise management, regulatory compliance, and mitigation strategies.

COMPONENTS REQUIRED:

ESP8266 NodeMCU Board
Microphone sensor
16*2 LCD Module
Breadboard
Connecting wires

MICROPHONE BASED SOUND DETECTOR:

The microphone based sound sensor is used to detect sound. It gives a measurement of how loud a sound is. The sound sensor module is a small board that mixes a microphone (50Hz-10kHz) and a few processing circuits to convert sound waves into electrical signals. This electrical signal is fed to on-board LM393 High Precision Comparator to digitise it and is made available at the OUT pin.

The module features a built-in potentiometer for sensitivity adjustment of the OUT signal. We will set a threshold by employing a potentiometer. So that when the amplitude of the sound exceeds the edge value, the module will output LOW, otherwise, HIGH. Apart from this, the module has two LEDs. The facility LED will illuminate when the module is powered. The Status LED will illuminate when the digital output goes LOW.

ARDUINO UNO:

Arduino is an 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 asPWM output)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.

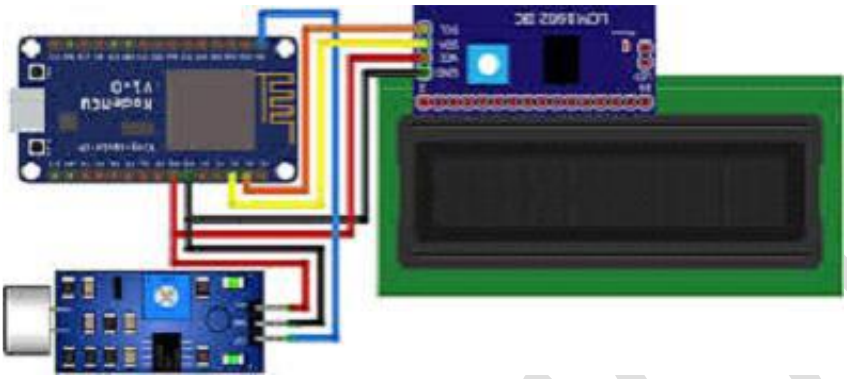
FEATURE:

- Operating voltage is5v.
- DC current per input pin is 40mA.
- Clock speed16MHz.
- DC current for the 3.3v pin is 50mA.
- SPAM 2 KB
- EEPROM 1KB

SOUND SENSOR MODULE:

The sound sensor only has three pins: VCC, GND & OUT. VCC pin supplies power for the sensor & works on 3.3V to 5V. OUT pin outputs HIGH when conditions are quiet and goes LOW when sound is detected.

CIRCUIT DIAGRAM:



PYTHON CODE FOR CREATING APPLICATION:

```
import sys
from phant import Phant
from twilio.rest import TwilioRestClient
from time import sleep

account_sid = ""

auth_token = ""

client = TwilioRestClient(account_sid, auth_token)

p = Phant(public_key='q5JMKnDJKXCMnjbYr0lG', fields=['temp'],
private_key='')

while(True):

    data = p.get()
```

```
    print("Latest Loudness Value is: {}".format(data[0]['temp']))

    if float(data[0]['temp']) > 1500:
        message = client.messages.create(body="Loudness: {0}\nD313
Room is making noise, please take action".format(data[0]['temp']),

        to="+919650055244", # Replace with your phone number

        from_="+12018905183") # Replace with your Twilio number
      print (message.sid)
    sleep(15)
```

ARDUINO CODE FOR DECIBEL METER AND LCD DISPLAY:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);

const int sampleWindow = 50;
unsigned int sample;

#define SENSOR_PIN A0
#define PIN_QUIET 3
#define PIN_MODERATE 4
#define PIN_LOUD 5

void setup ()
{
  pinMode (SENSOR_PIN, INPUT);
  pinMode(PIN_QUIET, OUTPUT);
  pinMode(PIN_MODERATE, OUTPUT);
```

```
  pinMode(PIN_LOUD, OUTPUT);
  digitalWrite(PIN_QUIET, LOW);
  digitalWrite(PIN_MODERATE, LOW);
  digitalWrite(PIN_LOUD, LOW);

  Serial.begin(115200);
  lcd.begin();

  lcd.backlight();
  lcd.clear();
}
void loop ()
{
   unsigned long startMillis= millis();
   float peakToPeak = 0;

   unsigned int signalMax = 0;

   unsigned int signalMin = 1024;

   while (millis() - startMillis < sampleWindow)
   {
      sample = analogRead(SENSOR_PIN);
      if (sample < 1024)
      {
        if (sample > signalMax)
      {
        signalMax = sample;
      }
      else if (sample < signalMin)
      {
     signalMin = sample;
      }
   }
```

```
}
peakToPeak = signalMax - signalMin;
int db = map(peakToPeak,20,900,49.5,90);

lcd.setCursor(0, 0);
lcd.print("Loudness: ");
lcd.print(db);
lcd.print("dB");

if (db <= 60)
{
   lcd.setCursor(0, 1);
   lcd.print("Level: Quite");
  digitalWrite(PIN_QUIET, HIGH);
  digitalWrite(PIN_MODERATE, LOW);
  digitalWrite(PIN_LOUD, LOW);
}
else if (db > 60 && db<85)
{
  lcd.setCursor(0, 1);
  lcd.print("Level: Moderate");
  digitalWrite(PIN_QUIET, LOW);
  digitalWrite(PIN_MODERATE, HIGH);
  digitalWrite(PIN_LOUD, LOW);
}
else if (db>=85)
{
  lcd.setCursor(0, 1);
  lcd.print("Level: High");
  digitalWrite(PIN_QUIET, LOW);
  digitalWrite(PIN_MODERATE, LOW);
  digitalWrite(PIN_LOUD, HIGH);
```

```
}
  delay(200);
  lcd.clear();

}
```

HTML CODE FOR WEB PAGE DESIGN

```
<!DOCTYPE html>
<html>
  <head>
    <!-- EXTERNAL LIBS-->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></sc
ript
>
    <script src="https://www.google.com/jsapi"></script>

    <!-- EXAMPLE SCRIPT -->
    <script>
// onload callback
function drawChart() {

  var public_key = 'q5JMKnDJKXCMnjbYr0lG';

// JSONP request
var jsonData = $.ajax({
   url: 'https://data.sparkfun.com/output/' + public_key + '.json',
   data: {page: 1},
   dataType: 'jsonp',
}).done(function (results) {

  var data = new google.visualization.DataTable();
```

```
  data.addColumn('datetime', 'Time');
  data.addColumn('number', 'Loudness');

  $.each(results.slice(0,80), function (i, row) {
     data.addRow([
     (new Date(row.timestamp)),
     parseFloat(row.temp) ]);
});

var chart = new google.visualization.LineChart($('#chart').get(0));

chart.draw(data, {
  title: 'SNU Library Noise Level Monitor'
  });

 });

}

// load chart lib
google.load('visualization', '1', {
   packages: ['corechart']
});

// call drawChart once google charts is loaded
google.setOnLoadCallback(drawChart);

</script>

</head>
<body>
  <div id="chart" style="width: 100%;"></div>
 </body>
```

</html>

CONCLUSION:

By using this project each and every variation we can analyse and inform nearby people in time. We can also analyse data from home using thingspeak.The most important factor of this system is that it is small, cost efficient and portable. Sensors are available easily anywhere. This system is fully helpful to save lives and overcome all the problems related to the environment.

NOISE POLLUTION MONITORING USING IOT

au820421106033 | NANDHARAJAN K, B.E/ECE

INTRODUCTION:

Noise monitoring refers to the systematic process of measuring, recording, and assessing sound levels in various environments to understand the extent of noise pollution and its potential impact on human health and the surrounding ecosystem. It involves using specialised equipment to gather data, which is then analysed to make informed decisions about noise management, regulatory compliance, and mitigation strategies.

COMPONENTS REQUIRED:

ESP8266 NodeMCU Board
Microphone sensor
16*2 LCD Module
Breadboard
Connecting wires

MICROPHONE BASED SOUND DETECTOR:

The microphone based sound sensor is used to detect sound. It gives a measurement of how loud a sound is. The sound sensor module is a small board that mixes a microphone (50Hz-10kHz) and a few processing circuits to convert sound waves into electrical signals. This electrical signal is fed to on-board LM393 High Precision Comparator to digitise it and is made available at the OUT pin.

The module features a built-in potentiometer for sensitivity adjustment of the OUT signal. We will set a threshold by employing a potentiometer. So that when the

amplitude of the sound exceeds the edge value, the module will output LOW, otherwise, HIGH. Apart from this, the module has two LEDs. The facility LED will illuminate when the module is powered. The Status LED will illuminate when the digital output goes LOW.

ARDUINO UNO:

Arduino is an 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 asPWM output)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.

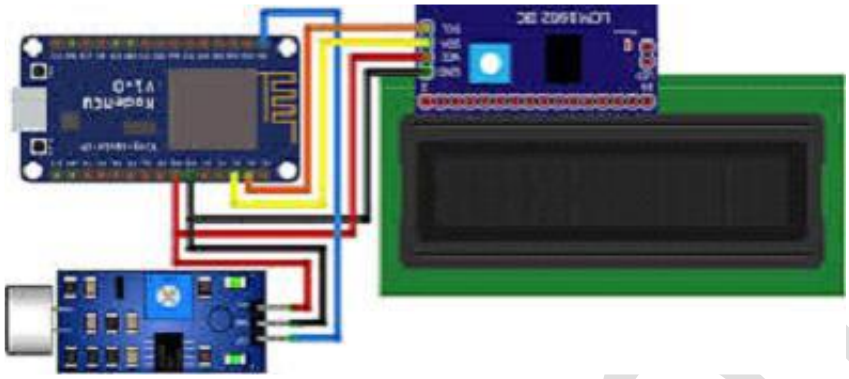
FEATURE:

- Operating voltage is5v.
- DC current per input pin is 40mA.
- Clock speed16MHz.
- DC current for the 3.3v pin is 50mA.
- SPAM 2 KB
- EEPROM 1KB

SOUND SENSOR MODULE:

The sound sensor only has three pins: VCC, GND & OUT. VCC pin supplies power for the sensor & works on 3.3V to 5V. OUT pin outputs HIGH when conditions are quiet and goes LOW when sound is detected.

CIRCUIT DIAGRAM:



PYTHON CODE FOR CREATING APPLICATION:

```
import sys
from phant import Phant
from twilio.rest import TwilioRestClient
from time import sleep

account_sid = ""

auth_token = ""

client = TwilioRestClient(account_sid, auth_token)

p = Phant(public_key='q5JMKnDJKXCMnjbYr0lG', fields=['temp'],
private_key='')

while(True):

      data = p.get()

    print("Latest Loudness Value is: {}".format(data[0]['temp']))
```

```
    if float(data[0]['temp']) > 1500:
        message = client.messages.create(body="Loudness: {0}\nD313
Room is making noise, please take action".format(data[0]['temp']),

        to="+919650055244", # Replace with your phone number

        from_="+12018905183") # Replace with your Twilio number
      print (message.sid)
    sleep(15)
```

ARDUINO CODE FOR DECIBEL METER AND LCD DISPLAY:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);

const int sampleWindow = 50;
unsigned int sample;

#define SENSOR_PIN A0
#define PIN_QUIET 3
#define PIN_MODERATE 4
#define PIN_LOUD 5

void setup ()
{
  pinMode (SENSOR_PIN, INPUT);
  pinMode(PIN_QUIET, OUTPUT);
  pinMode(PIN_MODERATE, OUTPUT);
  pinMode(PIN_LOUD, OUTPUT);
  digitalWrite(PIN_QUIET, LOW);
```

```
  digitalWrite(PIN_MODERATE, LOW);
  digitalWrite(PIN_LOUD, LOW);

  Serial.begin(115200);
  lcd.begin();

  lcd.backlight();
  lcd.clear();
}
void loop ()
{
   unsigned long startMillis= millis();
   float peakToPeak = 0;
   unsigned int signalMax = 0;
   unsigned int signalMin = 1024;
   while (millis() - startMillis < sampleWindow)
   {
      sample = analogRead(SENSOR_PIN);
      if (sample < 1024)
      {
        if (sample > signalMax)
      {
        signalMax = sample;
      }
      else if (sample < signalMin)
      {
     signalMin = sample;
      }
   }
}
peakToPeak = signalMax - signalMin;
```

```
int db = map(peakToPeak,20,900,49.5,90);

lcd.setCursor(0, 0);
lcd.print("Loudness: ");
lcd.print(db);
lcd.print("dB");

if (db <= 60)
{
   lcd.setCursor(0, 1);
   lcd.print("Level: Quite");
  digitalWrite(PIN_QUIET, HIGH);
  digitalWrite(PIN_MODERATE, LOW);
  digitalWrite(PIN_LOUD, LOW);
}
else if (db > 60 && db<85)
{
  lcd.setCursor(0, 1);
  lcd.print("Level: Moderate");
  digitalWrite(PIN_QUIET, LOW);
  digitalWrite(PIN_MODERATE, HIGH);
  digitalWrite(PIN_LOUD, LOW);
}
else if (db>=85)
{
  lcd.setCursor(0, 1);
  lcd.print("Level: High");
  digitalWrite(PIN_QUIET, LOW);
  digitalWrite(PIN_MODERATE, LOW);
  digitalWrite(PIN_LOUD, HIGH);
}
  delay(200);
```

```
  lcd.clear();

}
```

CONCLUSION:

By using this project each and every variation we can analyse and inform nearby people in time. We can also analyse data from home using thingspeak.The most important factor of this system is that it is small, cost efficient and portable. Sensors are available easily anywhere. This system is fully helpful to save lives and overcome all the problems related to the environment.

NOISE POLLUTION MONITORING USING IoT

TEAM MEMBER

au820421106033: NANDHARAJAN K

Phase-2 Document Submission

Project: Noise Pollution Monitoring

OBJECTIVE:

Real-time noise pollution monitoring, public awareness, noise regulation compliance, and improved quality of life.

Phase 2: Innovation

1.Introduction

- in a world where urbanization is on the rise and the soundscape of our surroundings
- continuously evolves, monitoring and mitigating noise pollution have become paramount
- concerns. Our project, aptly titled "Innovation in Noise Pollution Monitoring," aims to address
- this challenge by introducing an innovative system for real-time noise level assessment and
- display. This system incorporates a robust sensor network that detects ambient noise and
- promptly communicates it through an LCD display, enabling individuals to stay informed
- about their sonic environment. Beyond passive monitoring, this project takes a proactive
- approach by incorporating mechanisms to curtail sources of excessive noise, thereby
- fostering quieter, healthier urban spaces.

- The primary objective of this initiative is to provide individuals with the tools and information
- needed to make informed decisions about their surroundings. By determining the
- appropriate decibel level thresholds and developing a responsive system, our project
- facilitates not only real-time noise monitoring but also an improved quality of life. We
- envisage an urban environment where excessive noise is not only quantifiable but also
- controllable, where individuals can actively participate in noise pollution reduction by
- turning off instruments and equipment that contribute to the acoustic disturbances.
- This submission outlines the foundational elements of our innovative noise pollution

- monitoring project, from sensor deployment and data processing to the display of
- actionable information on an LCD screen. By harnessing technology and data, we strive to
- create quieter, more harmonious urban landscapes that enhance the well-being of
- individuals and communities

2.Innovation

Servo Motor Implementation:
Integrating a servo motor into your monitoring system offers a dynamic and automated way
to control noise sources. Here's how it can be implemented:

1.Device Identification: Firstly, we should need to identify the specific devices in our environment that
contribute to excessive noise. This could include sources like construction equipment, industrial machinery, or even loudspeaker systems.
Servo Control Mechanism: Connect the servo motor to the device's power source or control

2.panel in a way that allows it to physically toggle the device's power switch. The servo should
be positioned in such a way that it can engage the switch and turn off the device when
triggered.

3.Noise Threshold Detection: The noise monitoring system continuously measures and
analyzes the ambient noise levels. When the noise level exceeds a predetermined

threshold, the system triggers the servo motor to deactivate the identified noisy device.

4.Safety Considerations: Ensure that safety measures are in place to prevent unintended
device deactivation, especially if the device being controlled has critical operations.
Direct Switch Connection Implementation:
Alternatively, we can use direct switch connections to control noisy devices. This method is simpler and might be suitable for devices where a manual switch can be easily accessed and
turned off.

Here's how it can be done:
1.Device Identification: As with the servo motor implementation,
2.identify the noise-producing devices.
3.Switch Control Mechanism: Create a control mechanism to directly interface with the power switch or control panel of the noisy device. This control mechanism could be an electronically controlled switch or relay.
Triggering Logic: Implement a triggering logic in your monitoring system that, when noise
levels exceed a defined threshold, sends a signal to the switch control mechanism to turn
off the device.
4. Manual Override: Consider providing a manual override or emergency shutdown capability,
allowing individuals to deactivate the device independently if necessary.
Both approaches have their advantages and considerations. Servo motors offer more
precise and automated control, while direct switch connections are simpler and can be
retrofitted to various devices more easily. In either case, safety and reliability should be
paramount concerns, and you must ensure that any actions to deactivate devices do not

create additional hazards or disruptions. This innovative approach has the potential to
significantly reduce noise pollution by addressing its sources in real-time, ultimately
improving the quality of life in the monitored area.

- When multiple noise-emitting instruments, such as chainsaws and drilling machines,
- converge in the same vicinity and operate concurrently, the resultant cacophony can create
- a significantly amplified noise environment in the specified area. This heightened noise level
- can be particularly disruptive and detrimental to the overall well-being of individuals in the
- immediate vicinity. The cumulative effect of simultaneous noise sources not only intensifies
- the sound pressure but also compounds the potential for adverse health impacts, including
- stress, sleep disturbances, and decreased cognitive performance.
- In such scenarios, an active noise pollution monitoring system equipped with intelligent
- triggering mechanisms, as discussed previously, becomes all the more critical. It not only
- provides real-time noise level readings but also offers the capability to identify, differentiate,
- and initiate actions against multiple disruptive sources simultaneously. This proactive

- approach ensures that the monitoring system can assess the totality of the acoustic
- environment, respond promptly to noise violations, and coordinate the deactivation or
- mitigation of the various contributing devices. By addressing multiple noise sources
- efficiently, this innovative solution aims to restore a sense of tranquility and quality of life to
- those affected by the concurrent clamor of multiple instruments in a specified area.
- For the hardware parts, LM 393 sound sensor is used to read the readings of the sound level from the environment. The reading of sound sensor is calibrated using the real sound level meter to get
- the accurate readings of the sound level.

- The 16x2 LCD will show the values of sound level at that researched area and give the warning that says the level of sound is high when the measurement exceeds the set value. If the users could not read the readings due to poor eyesight, they can know the level of sound by using the light emitting diodes (LED) which in red, blue and green colour placed below the LCD.

- LED acts as an indicator to indicate when the noise is very high. It will turn to red, blue for low noise while green for intermediate level. All these components such as sound sensor, LCD, and LEDs will be connected to the ESP8266 NodeMCU.

4. Integration Approach:

The prototype is calibrated using actual sound level meter to get the accurate measurement of sound level or sound intensity.

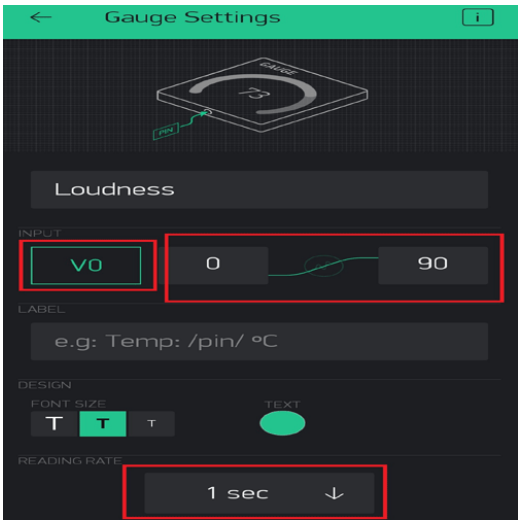
The prototype is used to measure the sound level at five different times which are during morning, afternoon, evening, night and midnight at a place where UTM students are staying. The reading is taken 30 minutes per range of time. Within that time, the lowest and highest sound intensity were recorded and from the readings, the range of sound level was determined during that specific range of time.

	ZONE: RESIDENTIAL			LOCATION: RTU, Kota			
	TIME	L_{eq} (A) SOUND PRESSURE LEVEL IN DAY TIME					
		Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
		12-07-2019	13-07-2019	14-07-2019	15-07-2019	16-07-2019	17-07-2019	
Morning	06:00 – 07:00	76.4	76.6	76.9	77	74.2	76.6	76.4
	07:00 – 08:00	73.6	74.9	77.9	75.3	76.3	77.6	76.2
	08:00 – 09:00	72.6	73.3	76.8	77.2	76.5	75.8	75.7
	09:00 – 10:00	74.9	72.1	77.5	75.8	77.4	76.9	76.1
	10:00 – 11:00	75.1	73.3	74.3	75.9	76.5	77.9	75.8
							Morning Average	76.0
Afternoon	11:00 – 12:00	79.6	79.3	79.9	78.9	79.9	80	79.6
	12:00 – 13:00	78.5	77.6	80.7	80.9	78.1	78.9	79.3
	13:00 – 14:00	79.3	79.6	79.9	80	80.1	79.9	79.8
	14:00 – 15:00	77.3	78.5	80.9	79.4	80.2	78.9	79.4
	15:00 – 16:00	79.8	79.6	79.9	80	80.1	80.9	80.1
							Afternoon Average	79.6
Evening	16:00 – 17:00	81.1	83.9	81	81.1	82.8	81.2	82.0
	17:00 – 18:00	83.1	82.9	82.7	83	83.7	82.3	83.0
	18:00 – 19:00	83.3	82.9	83.4	83.5	83.9	83.7	83.5
	19:00 – 20:00	81.1	83.5	81.5	81.3	82.9	81.1	82.0
	20:00 – 21:00	82.6	82.7	82.3	81.9	82.6	82.4	82.4
	21:00 – 22:00	81.3	81.4	82.9	83.8	81.2	81.1	82.1
							Evening Average	82.5

- The foundation of our noise pollution monitoring project lies in the extensive dataset that
- captures real-time noise levels in a specific city across various time intervals. This invaluable
- dataset provides us with a comprehensive understanding of the acoustic landscape within
- the urban environment. By analyzing this data, we aim to determine the threshold noise
- levels that are critical for noise pollution assessment. These thresholds will serve as the

- reference points against which our monitoring device will gauge the current noise levels.
- Such data-driven thresholds enable our device to distinguish between acceptable and
- excessive noise levels, ensuring that users receive timely and accurate notifications.
- The incorporation of data-driven thresholds not only enhances the precision of our noise
- monitoring system but also fosters a more proactive approach to noise pollution
- management. By recommending our device's performance parameters based on these
- thresholds, we empower the device to respond effectively and efficiently. For instance, in
- areas where noise levels consistently exceed the recommended thresholds during specific
- time periods, the device can trigger alerts or even initiate corrective actions, such as
- notifying relevant authorities or suggesting noise-reduction measures to individuals. The
- utilization of data-derived thresholds represents a crucial step toward creating a dynamic
- and intelligent noise monitoring system that is attuned to the unique noise patterns of the
- city, ensuring that it remains adaptable and responsive to the evolving soundscape of the urban environment

4.Application & interface



- Using the Blynk app's gauge option to display the noise level in our mobile app offers us a
- convenient and user-friendly way to visualize real-time noise data. We installed the Blynk
- app on our mobile device, signed in, and created a new project related to our noise
- monitoring system. We added a "Gauge" widget to the project, which visually displays the
- noise level using a dial or needle on a scale. To integrate our noise monitoring hardware, we
- obtained the unique authentication token for our project. Then, we connected our hardware
- to the Blynk platform by writing code specific to our hardware platform (e.g., using the Blynk
- library for Arduino or Raspberry Pi). We incorporated instructions in our code to send real-
- time noise data to the Blynk server, updating the gauge widget with the current noise level.
- We also customized the gauge widget properties within the app, setting minimum and
- maximum values, changing the appearance, and adding labels for a user-friendly interface.
- As our noise monitoring system collects data, the Blynk app displays the current noise level
- in real-time. We've even set up alert thresholds to notify us when noise levels exceed certain
- limits. The Blynk app provides us with a straightforward way to monitor noise levels,
- ensuring that we can stay informed about the noise environment in our surroundings using
- our mobile device.

5. Conclusion

- People thought that noise pollution is merely an annoyance but it is actually very important to monitor noise level because according to research, people who are exposed to noise for a long duration of time can have hearing loss, sleep disturbance, high blood pressure and injuries.
  Besides, it can affect the learning process of people in terms of understanding and behaviour. Thus, this research investigates and subsequently proposes the suitable time for students to study by utilising the cloud server and android application to realize an IoT based noise monitoring system.
  From the prototype, it also can be determined the dominant sound that increases the noise level in the researched area. The app can display the reading from the prototype successfully.
  Based on the results, it can be concluded that the students can study throughout the day starting from morning until midnight during weekends because the noise level is still under the allowable standard which is 60dBA according to CIBSE.
  As for weekdays, the suitable time to study is found out to be at midnight because the readings are below 60dBA for that time.
  The limitation of this study is that it is conducted within UTM campus only.
  In addition, the app can only show the reading from only a single prototype as well as the prototype and app can only operate when there is an internet connection. This system can be improvised in future to include measurements outside UTM such as at schools and airports