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:

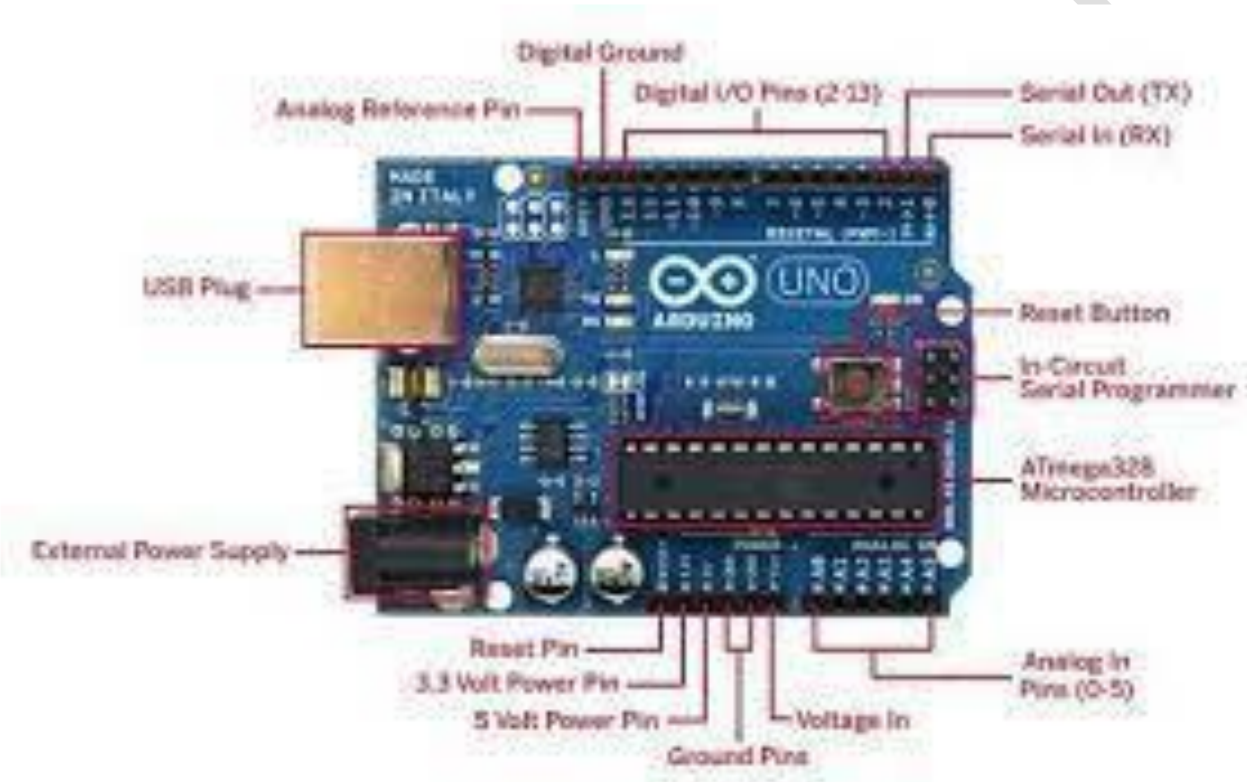
ESP8266 NodeMCU Board
Microphone sensor
16*2 LCD Module
Breadboard
Connecting wires

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.

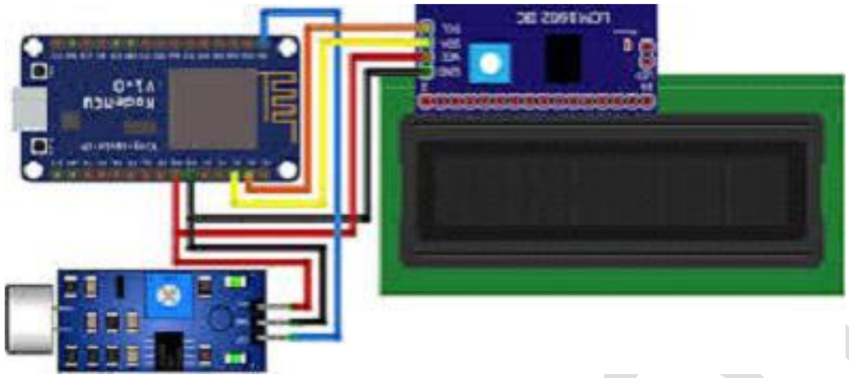
FEATURE:

• 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.