## **Project: Microcontroller-based Earthquake Alert System**

1. Problem Statement

The goal is to develop a cost-effective and efficient earthquake alert system that detects seismic vibrations and provides immediate alerts. This project is designed to enhance safety in industrial settings by enabling timely response to potential earthquake incidents, thus minimizing risks and reducing possible damages.

2. Scope of the Solution

- Early Warning System: The system monitors seismic activity and triggers alerts when detecting vibrations above a preset threshold.

- Industrial Safety\*\*: Targeted for use in industrial environments where immediate alerts are essential to prevent injuries and damage.

- Prototyping and Scalability\*\*: This project serves as a prototype but is scalable to cover larger areas with stronger sensors and more advanced microcontrollers.

- Data Analysis\*\*: Real-time monitoring and alerting with vibration levels displayed on an LCD, giving personnel a clear understanding of seismic activity intensity.

3. Components Required

- Microcontroller\*\*: Arduino Uno R3

- Vibration Sensor\*\*: SW-420 vibration module or ADXL335 accelerometer sensor for detecting seismic waves

- Buzzer\*\*: Piezo buzzer to emit a sound alarm on detection

- Display Module\*\*: 16x2 LCD (Liquid Crystal Display) to show vibration status and threshold levels

- LEDs\*\*: Green and Red LEDs to indicate normal and alert states

- Additional Components\*\*: Resistors (220Ω for LEDs, 10KΩ for LCD), potentiometer (for LCD brightness adjustment), and breadboard/wires for circuit setup

- Power Source\*\*: 5V from Arduino or an external 9V battery for portability

- Software & IDE\*\*:

- IDE\*\*: Arduino IDE for programming the microcontroller

- Simulation Software\*\*: TinkerCad or Fritzing for pre-assembly circuit testing

- PCB Design Software\*\*: KiCad or EasyEDA for creating the Gerber files required for PCB manufacturing

---

4. Project Design and Development

4.1 Circuit Design

- Connect the \*\*SW-420 Vibration Sensor\*\* to Arduino:

- VCC of SW-420 to Arduino 5V

- GND of SW-420 to Arduino GND

- Digital OUT of SW-420 to Arduino digital pin 2

- Connect the \*\*Buzzer\*\*:

- Positive terminal of the buzzer to digital pin 9 of Arduino

- Negative terminal to GND

- Connect the \*\*LEDs\*\*:

- Green LED: Connected to digital pin 10 with a 220Ω resistor in series

- Red LED: Connected to digital pin 11 with a 220Ω resistor in series

- Connect the \*\*16x2 LCD Display\*\*:

- LCD VCC to 5V, GND to GND

- Data pins of LCD to digital pins 4, 5, 6, 7 of Arduino

- Connect a potentiometer between LCD V0 and GND for contrast control.

4.2 Code for Earthquake Detection

Here’s sample code for configuring the vibration sensor, alert system, and LCD display. The vibration threshold can be adjusted based on the sensitivity requirement.

###### ```cpp

#include<LiquidCrystal.h> // lcd Header

LiquidCrystal lcd(12,11,5,4,3,2); // pins for LCD Connection

#define buzzer 10 // buzzer pin

#define led 13 //led pin

#define z A2 // z\_out pin of Accelerometer/\*variables\*/

int zsample=0;

long start;

int buz=0;

/\*Macros\*/

#define samples 50

#define maxVal 20 // max change limit

#define minVal -20 // min change limit

#define buzTime 5000 // buzzer on time

void setup()

{

lcd.begin(16,2); //initializing lcd

Serial.begin(9600); // initializing serial

delay(1000);

lcd.print("EarthQuake ");

lcd.setCursor(0,1);

lcd.print("Detector ");

delay(2000);

lcd.clear();

lcd.print("Calibrating.....");

lcd.setCursor(0,1);

lcd.print("Please wait...");pinMode(buzzer, OUTPUT);

pinMode(led, OUTPUT);

buz=0;

digitalWrite(buzzer, buz);

digitalWrite(led, buz);

for(int i=0;i<samples;i++) // taking samples for calibration

{

zsample+=analogRead(z);

}

zsample/=samples; // taking avg for z

delay(3000);

lcd.clear();

lcd.print("Calibrated");

delay(1000);

lcd.clear();

lcd.print("Device Ready");

delay(1000);

lcd.clear();

lcd.print(" Z ");

}

void loop(){

int value3=analogRead(z); //reading z out

int zValue=zsample-value3; // finding change in z

/\*displying change in x,y and z axis values over lcd\*/

lcd.setCursor(12,1);

lcd.print(zValue);

delay(100);

/\* comparing change with predefined limits\*/

if(zValue < minVal || zValue > maxVal)

{

if(buz == 0)

start=millis(); // timer start

buz=1; // buzzer / led flag activated

}

else if(buz == 1) // buzzer flag activated then alerting earthquake

{

lcd.setCursor(0,0);

lcd.print("Earthquake Alert ");

if(millis()>= start+buzTime)

buz=0;}

else

{

lcd.clear();

lcd.print("Z");

}

digitalWrite(buzzer, buz); // buzzer on and off command

digitalWrite(led, buz); // led on and off command

/\*sending values to processing for plot over the graph\*/

Serial.print("z=");

Serial.println(zValue);

Serial.println(" $");

}