EARTHQUAKE ALERT SYSTEM

B. Sam Niran
Vit Chennai

Aim:

To develop a prototype system which can detect and alert the earth quake.

Scope:

Building an earthquake detection system entails real-time monitoring of ground movement, offering early warnings for saving lives, contributing to earthquake research and prediction, safeguarding critical infrastructure, ensuring public safety through alerts, customizing the system for specific regions or global networks, developing data analysis algorithms, deploying sensors and hardware, integrating data into a centralized system, adhering to regulatory requirements, securing funding and collaborations, and maintaining and updating the system regularly.

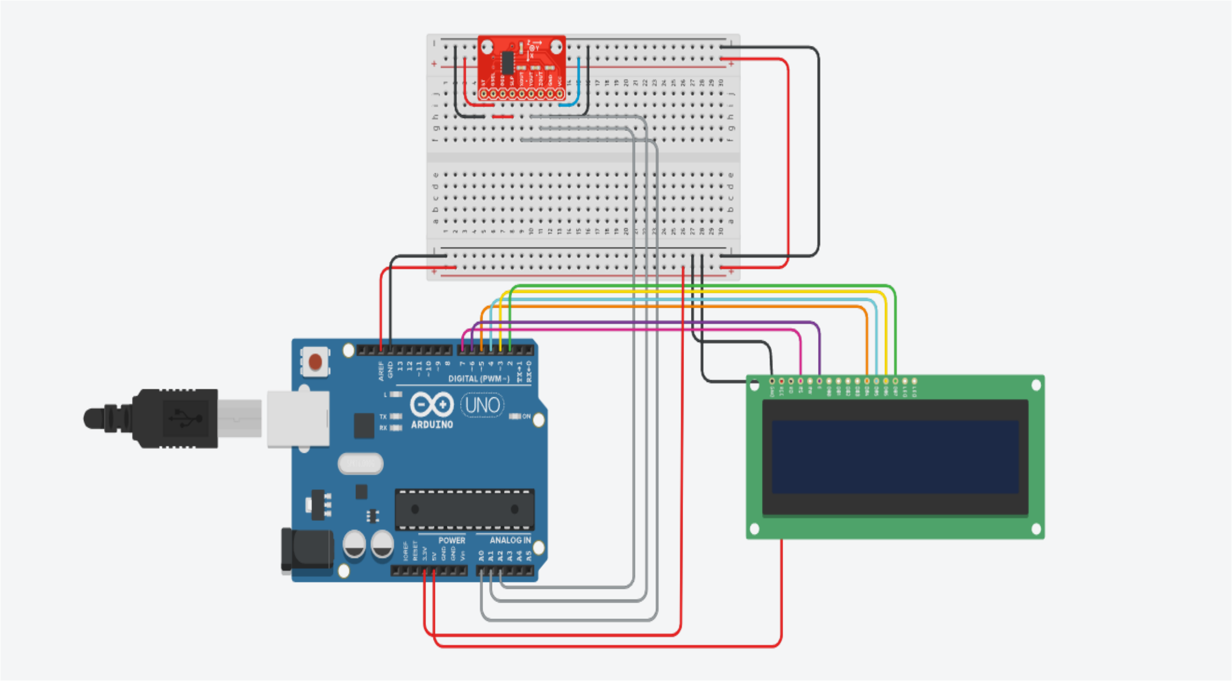
Components Required:

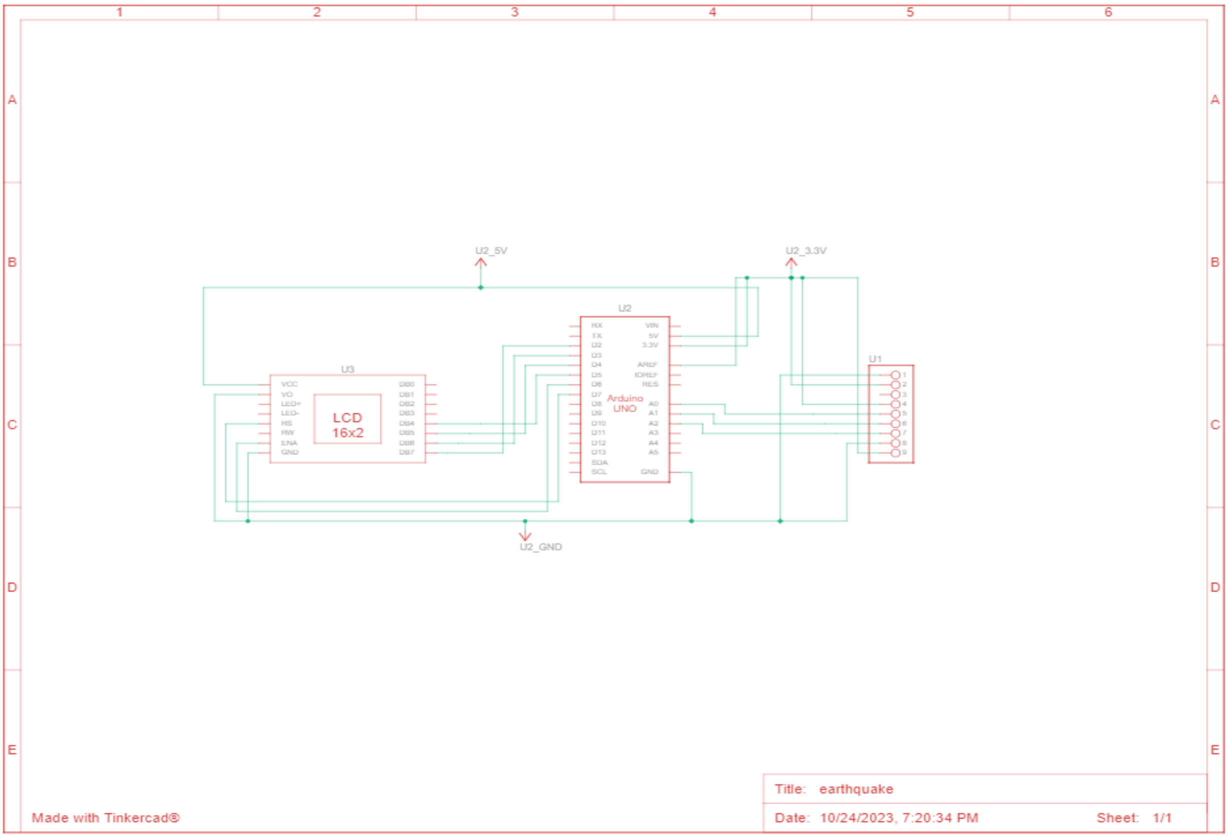
- Arduino UNO
- Accelerometer ADXL335
- 16x2 LCD
- Power Supply 9v/12v
- Berg sticks male/female
- Breadboard

Software Used:

- TinkerCAD
- EasyEDA

Simulated Circuit:





Working:

- Accelerometers measure acceleration in three axes (X, Y, and Z).But we will focus on one of these axes.
- We define a threshold value for detecting earthquake vibrations. When the accelerometer's reading exceeds this threshold, consider it an earthquake event.
- When the system detects an earthquake event (exceeding the threshold), display a message on the LCD indicating that an earthquake is detected. You can also include the vibration intensity or duration.
- To ensure accurate earthquake detection, you may need to calibrate your accelerometer and adjust the threshold accordingly.
- Once the system is ready, deploy it in a location where earthquake detection is needed.

Outcome:

The outcome of this project is a earthquake detection system that utilizes an Arduino, an accelerometer, and a 16x2 LCD display. When an earthquake event occurs and the accelerometer detects vibrations exceeding a predefined threshold, the system displays an alert on the LCD, indicating the presence of seismic activity. This project serves as an educational and experimental model for understanding the basic principles of earthquake detection and can be used for small-scale applications. However, for real-world earthquake monitoring and alerting, more advanced and comprehensive algorithms are typically employed.

Code:

```
#include <LiquidCrystal.h>


LiquidCrystal lcd( 7, 6, 5, 4, 3, 2);
unsigned char Character1[8] = { 0x04, 0x1F, 0x11, 0x11, 0x1F, 0x1F, 0x1F, 0x1F };

unsigned char Character2[8] = { 0x01, 0x03, 0x07, 0x1F, 0x1F, 0x07, 0x03, 0x01 };
//Analog read pins
const int xPin = A2;
const int yPin = A1;
const int zPin = A0;
int minVal = 265;
int maxVal = 402;
//to hold the caculated values
double x;
double y;
double z;
void setup()
   {
    Serial.begin(9600);
          lcd.begin(16,2);
   }//end setup()



void loop()
   {
    speeder();
   }
```

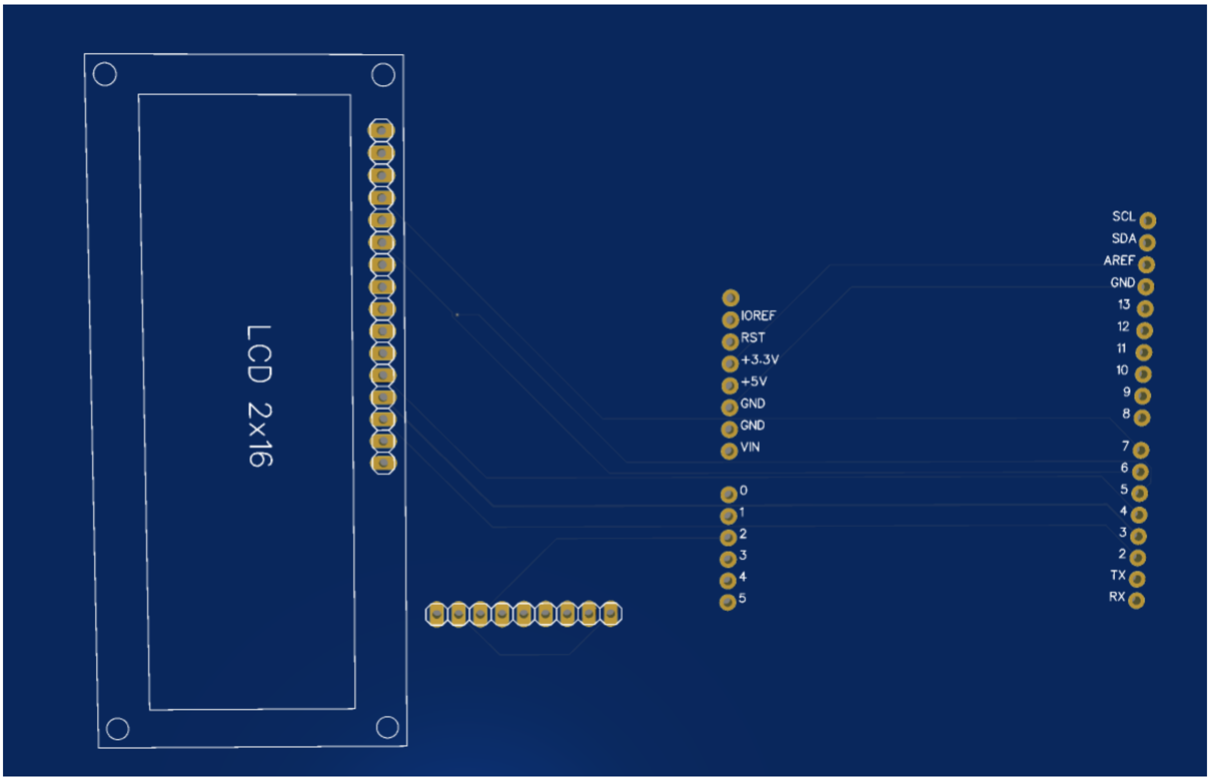
```
void speeder()
  {
   delay(3000);
   lcd.clear();
   //read the analog values from the accelerometer
   int xRead = analogRead(xPin);
   int yRead = analogRead(yPin);
   int zRead = analogRead(zPin);
   int xAng = map(xRead, minVal, maxVal, -90, 90);
   int yAng = map(yRead, minVal, maxVal, -90, 90);
   int zAng = map(zRead, minVal, maxVal, -90, 90);

   x = RAD_TO_DEG * (atan2(-yAng, -zAng) + PI);
   y = RAD_TO_DEG * (atan2(-xAng, -zAng) + PI);
   z = RAD_TO_DEG * (atan2(-yAng, -xAng) + PI);
        //lcd print
        if(x<=90){
                lcd.setCursor(2,0);
                lcd.print("x");
                lcd.setCursor(0,1);
                lcd.print(x);
                lcd.setCursor(9,0);
                lcd.print("y");
                lcd.setCursor(8,1);
        lcd.print(y);}
        else{
        lcd.setCursor(0,0);
        lcd.print("Earthquake Alert");
   }
```

Explanation:

This code is designed for an Arduino project that employs an accelerometer to gauge orientation and, under specific conditions, activate an "Earthquake Alert" message on a connected 16x2 LCD. The code begins by setting up the necessary libraries and initializing the LCD. Two custom characters are defined for potential LCD display enhancements. It then establishes connections to analog pins corresponding to the accelerometer and defines minimum and maximum values derived from the accelerometer's stationary state to aid calibration. The setup function initializes the Arduino board and LCD, while the loop function is the main execution loop. Within the user-defined "speeder" function, the LCD is cleared, and analog values from the accelerometer for the X, Y, and Z axes are read. These readings are subsequently converted to degrees and used in trigonometric calculations to determine the accelerometer's orientation in degrees for each axis. If the orientation on the X-axis exceeds 90 degrees, the code displays "Earthquake Alert" on the LCD, signifying a potential seismic event. In the event of readings within the specified range, the code displays the orientation values for X and Y on the LCD. Additionally, it provides optional serial output for debugging and data analysis. Please note that this is a basic example and may require calibration for accurate earthquake detection.

PCB Design:



Results:

Thus we have deployed an earthquake alert system using Arduino UNO, Accelerometer and LCD .

GitHub Link:

https://github.com/Sam-Niran/MPMCProject.git