Add Company Name

Information available in audio.

CATTLE EVACUATION EARTHQUAKE SENSOR USING AURDINO

Add a short description here

PRESENTED BY
k.komali venkateswari
21A31A04K9
ch.aashritha kavya
21A31A04K3

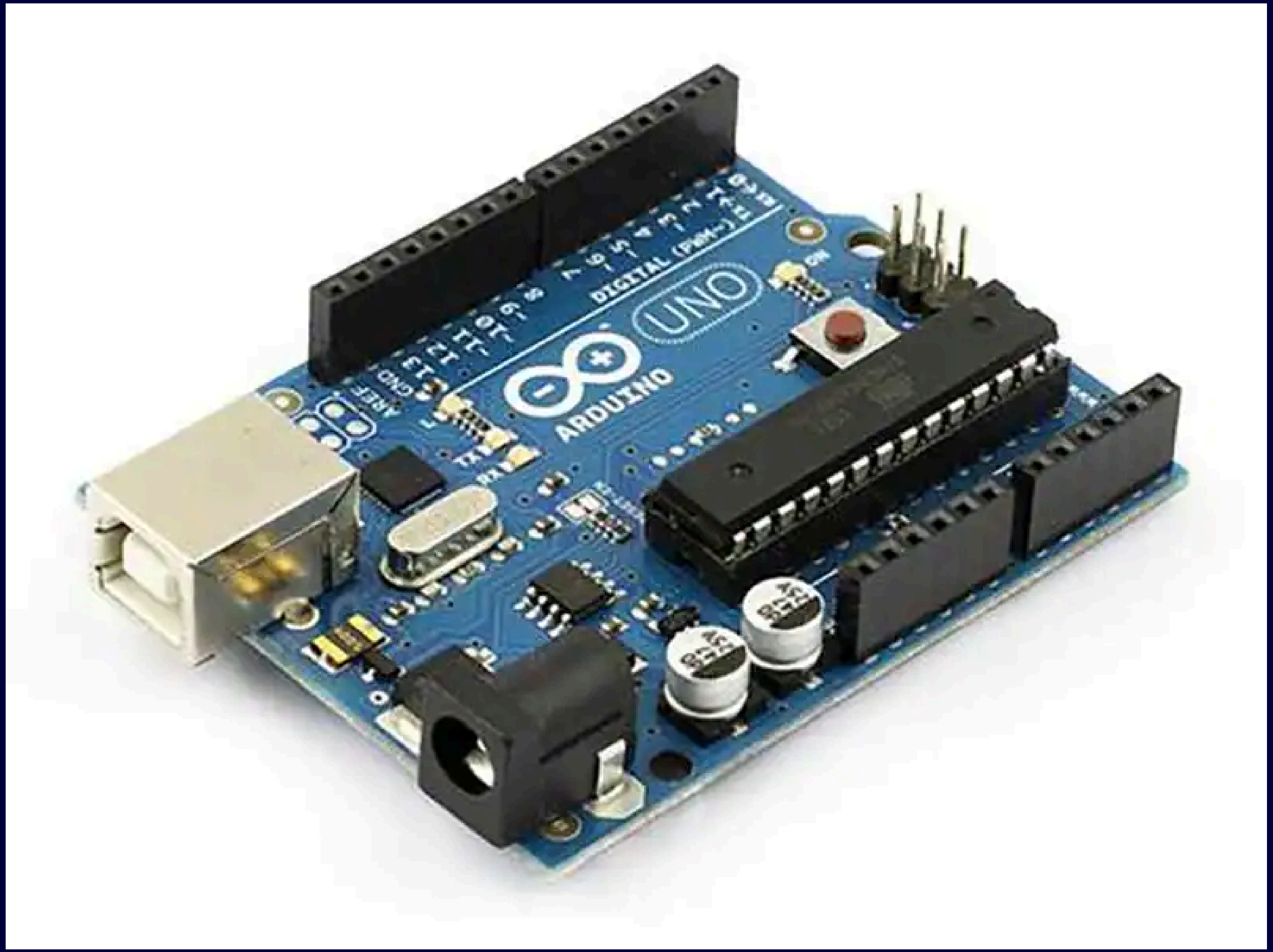
ere

problem statement

Earthquakes, hurricanes, volcanic eruptions, tsunamis, and forest fires can have devastating consequences. Many animals die, drowned or buried alive by dirt, ash, lava, or snow; crushed to death in collapsed or burnt burrows; smashed against trees and rocks, or pelted by hailstones.Add a little bit of body text

components used

- **aurdino uno board**
- **MPU6050 Accelrometer**
- 16*2 led display
- buzzer
- led

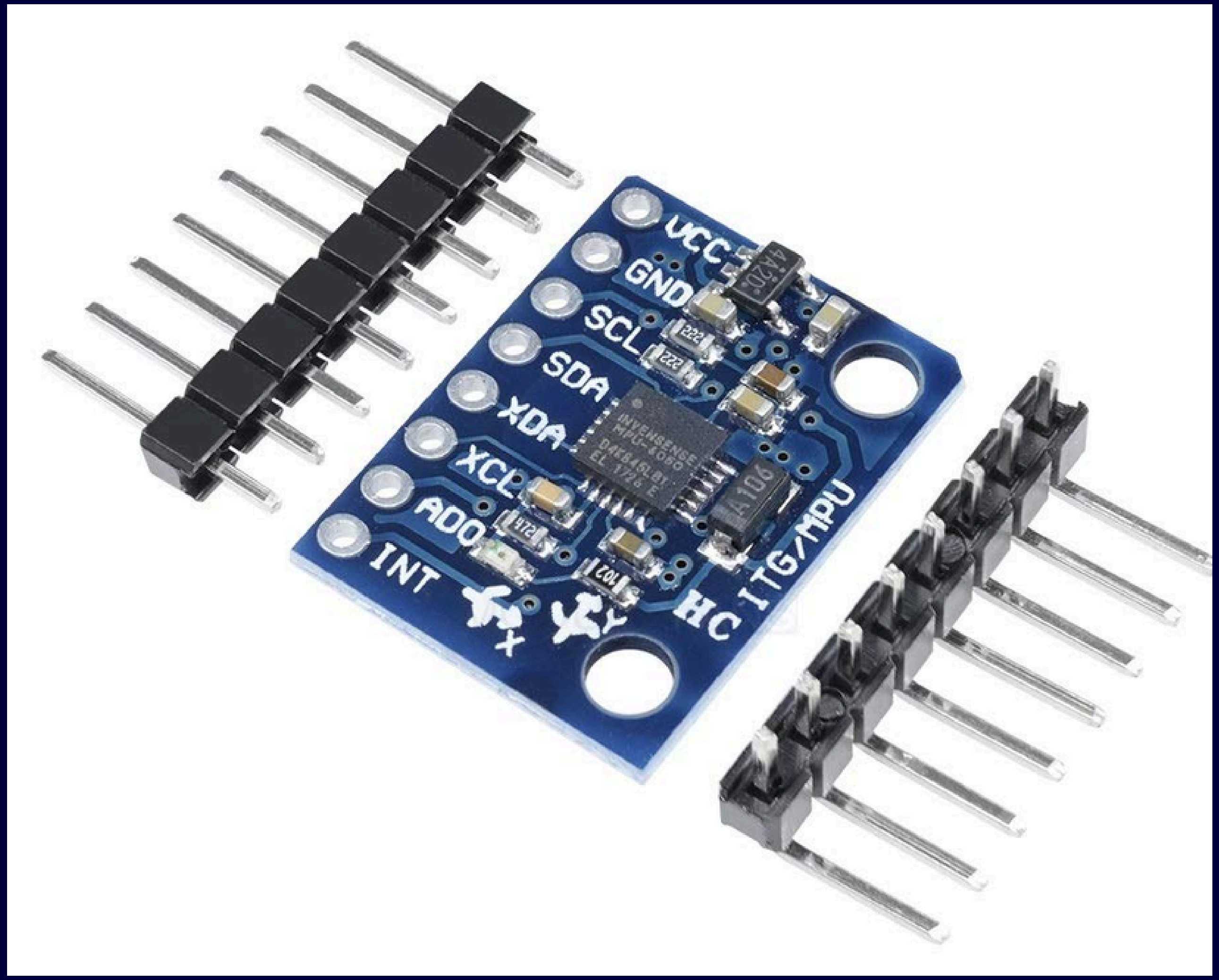


MPU6050 WORKING

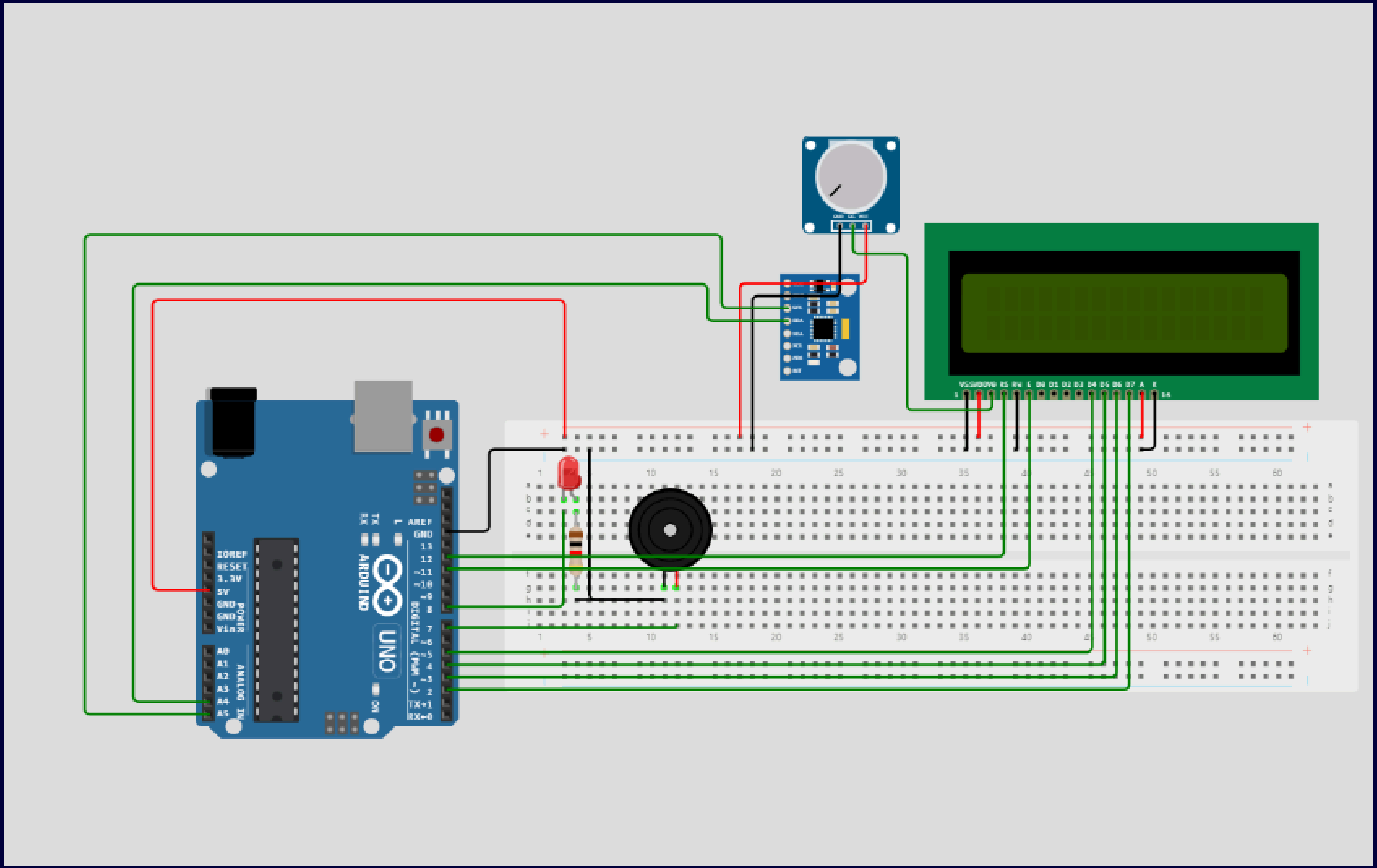
- MPU6050 is 6 Axis Accelerometer and gyroscope. It is generally a motion tracking device. It is also capable of measuring temperature. It consumes less power and very cheap in price.

Here are some feature MPU6050

- Gyroscope operating current: 3.6mA.
- Accelerometer normal operating current: 500µA.
- Standby current: 5µA.
- Operating voltages lies between 2.37v to 3.46v
- Improved low-frequency noise performance.
- Digital-output temperature sensor.



circuit


ARDUINO
UNO
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM ~)
TX
RX
L
ON

working of earhquake sensor using aurdino

The Arduino first of all initialize MPU 6050. Check the sleep mode and a clock signal from module then started reading the values. There are maximum and minimum values are declared in the code it checks if the value is greater or smaller than the desired values then it starts the buzzer, led and display message “***Earthquake***”o n the LCD. If the values are normal then it does nothing.
when it sensor detects the waves then the lock is set to open.

```
#include <LiquidCrystal.h>

#include <Wire.h>

#include <MPU6050.h>

#define minval -5

#define maxval 3

MPU6050 mpu;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()

{   lcd.begin(16, 2);

Serial.begin(115200);

pinMode(7,OUTPUT);

pinMode(8,OUTPUT);

lcd.print("   EarthQuake ");

lcd.setCursor(0, 1);

lcd.print("   Detector");

delay (2000);

lcd.clear();

// Initialize MPU6050

Serial.println("Initialize MPU6050");

while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))

{ Serial.println("Could not find a valid MPU6050 sensor, check wiring!");

delay(500);}

mpu.setThreshold(3);

// Check settings

checkSettings();

}
```

```
void checkSettings()

{

Serial.println();


Serial.print(" * Sleep Mode:       ");

Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

Serial.print(" * Clock Source:      ");

switch(mpu.getClockSource())

{case MPU6050_CLOCK_KEEP_RESET:     Serial.println("Stops the clock and keeps the timing generator in reset");
break;

case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL with external 19.2MHz reference"); break;

case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL with external 32.768kHz reference"); break;

case MPU6050_CLOCK_PLL_ZGYRO:      Serial.println("PLL with Z axis gyroscope reference"); break;

case MPU6050_CLOCK_PLL_YGYRO:      Serial.println("PLL with Y axis gyroscope reference"); break;

case MPU6050_CLOCK_PLL_XGYRO:      Serial.println("PLL with X axis gyroscope reference"); break;

case MPU6050_CLOCK_INTERNAL_8MHZ:  Serial.println("Internal 8MHz oscillator"); break;

}

Serial.print(" * Gyroscope:         ");

switch(mpu.getScale())

{case MPU6050_SCALE_2000DPS:        Serial.println("2000 dps"); break;

case MPU6050_SCALE_1000DPS:        Serial.println("1000 dps"); break;

case MPU6050_SCALE_500DPS:         Serial.println("500 dps"); break;

case MPU6050_SCALE_250DPS:         Serial.println("250 dps"); break:}

Serial.print(" * Gyroscope offsets: ");

Serial.print(mpu.getGyroOffsetX());

Serial.print(" / ");

Serial.print(mpu.getGyroOffsetY());

Serial.print(" / ");

Serial.println(mpu.getGyroOffsetZ());

Serial.println();}

void loop()

{   Vector rawGyro = mpu.readRawGyro();

Vector normGyro = mpu.readNormalizeGyro();

Serial.print(" Xraw = ");

Serial.print(rawGyro.XAxis);

Serial.print(" Yraw = ");

Serial.print(rawGyro.YAxis);

Serial.print(" Zraw = ");

Serial.println(rawGyro.ZAxis);
```

```
if(normGyro.XAxis > maxval || normGyro.XAxis < minval && normGyro.YAxis >
maxval || normGyro.YAxis  < minval && normGyro.ZAxis > maxval ||
normGyro.ZAxis  < minval)

{ digitalWrite(7,HIGH);

digitalWrite(8,HIGH);

delay(300);

digitalWrite(7,HIGH);

digitalWrite(8,HIGH);

delay(300);

lcd.clear();

lcd.print("***EarthQuake***");

delay (1000);

lcd.clear();}

else{digitalWrite(7,LOW);

digitalWrite(8,LOW);}

Serial.print(" Xnorm = ");

Serial.print(normGyro.XAxis);

Serial.print(" Ynorm = ");

Serial.print(normGyro.YAxis);

Serial.print(" Znorm = ");

Serial.println(normGyro.ZAxis);

delay(10);}
```

Thank you!