**Vibration Monitoring System**

**📌 Project Title:**

**Vibration Monitoring and Alert System using Arduino**

**📦 Components Required:**

| **Component** | **Quantity** | **Description** |
| --- | --- | --- |
| Arduino Uno | 1 | Microcontroller board |
| Vibration Sensor | 1 | SW-420 or similar analog vibration module |
| 16x2 I2C LCD | 1 | For displaying vibration level and system status |
| Red LED | 1 | Indicates danger |
| Green LED | 1 | Indicates safe condition |
| Buzzer | 1 | Gives alert on high vibration |
| 220Ω Resistor | 2 | For LEDs |
| Breadboard & Wires | As needed | For circuit connections |
| USB Cable | 1 | To upload code and power Arduino |

**🧠 Project Overview:**

This project continuously monitors vibration levels using a vibration sensor. The system displays the vibration level on a 16x2 I2C LCD. Based on a predefined threshold, the system classifies the state as either **SAFE** or **DANGER**:

* When vibration is low → Green LED ON, "Status: SAFE"
* When vibration crosses the threshold → Red LED blinks, buzzer turns ON, "Status: DANGER!!"

Additionally, values are printed to the **Serial Monitor** and visualized using **Serial Plotter** for live debugging.

**⚙️ Circuit Connections:**

**(URL:-** [**https://app.cirkitdesigner.com/project/452dff6a-d60a-4d25-b512-400400966ec8**](https://app.cirkitdesigner.com/project/452dff6a-d60a-4d25-b512-400400966ec8)**)**

**1. Vibration Sensor (Analog output model like SW-420)**

| **Sensor Pin** | **Arduino** |
| --- | --- |
| VCC | 5V |
| GND | GND |
| OUT | A0 |

**2. LEDs & Buzzer**

| **Component** | **Arduino Pin** | **Notes** |
| --- | --- | --- |
| Red LED | 8 | Danger alert |
| Green LED | 9 | Safe mode |
| Buzzer | 10 | Alarm |

Use 220Ω resistors with the LEDs.

**3. I2C LCD (16x2)**

| **LCD Pin** | **Arduino Pin** |
| --- | --- |
| VCC | 5V |
| GND | GND |
| SDA | A4 (Uno) |
| SCL | A5 (Uno) |

**🧾 Libraries You Need**

* LiquidCrystal\_I2C

***Install them via the Arduino Library Manager (Sketch → Include Library → Manage Libraries).***

**📜 Arduino Code:**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

const int vibrationPin = A0;

const int redLedPin = 8;

const int greenLedPin = 9;

const int buzzerPin = 10;

int dangerThreshold = 400; // Adjust this as needed

unsigned long dangerHoldTime = 2000;

unsigned long lastDangerTime = 0;

bool dangerMode = false;

void setup() {

  pinMode(vibrationPin, INPUT);

  pinMode(redLedPin, OUTPUT);

  pinMode(greenLedPin, OUTPUT);

  pinMode(buzzerPin, OUTPUT);

  lcd.init();

  lcd.backlight();

  lcd.setCursor(0, 0);

  lcd.print("Vibration Monitor");

  delay(2000);

  lcd.clear();

  Serial.begin(9600); // Start serial communication

  Serial.println("System Ready...");

}

void loop() {

  int vibrationValue = analogRead(vibrationPin);

  unsigned long currentTime = millis();

  // Print vibration value to Serial Monitor

  Serial.print("Vibration Value: ");

  Serial.println(vibrationValue);

  lcd.setCursor(0, 0);

  lcd.print("Vib Level: ");

  lcd.print(vibrationValue);

  lcd.print("   ");

  // Check for danger level

  if (vibrationValue >= dangerThreshold) {

    dangerMode = true;

    lastDangerTime = currentTime;

  }

  // Maintain danger mode for at least 2 seconds

  if (dangerMode && (currentTime - lastDangerTime > dangerHoldTime)) {

    dangerMode = false;

  }

  if (dangerMode) {

    // Blink red LED every 300ms

    if (currentTime % 600 < 300) {

      digitalWrite(redLedPin, HIGH);

    } else {

      digitalWrite(redLedPin, LOW);

    }

    digitalWrite(greenLedPin, LOW);

    digitalWrite(buzzerPin, HIGH);

    lcd.setCursor(0, 1);

    lcd.print("Status: DANGER!! ");

  } else {

    digitalWrite(redLedPin, LOW);

    digitalWrite(greenLedPin, HIGH);

    digitalWrite(buzzerPin, LOW);

    lcd.setCursor(0, 1);

    lcd.print("Status: SAFE     ");

  }

  delay(100); // Increased delay for better serial readability

}

**📜 Arduino Code Explanation:**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

Includes libraries for I2C and LCD.

LiquidCrystal\_I2C lcd(0x27, 16, 2);

Creates an LCD object with I2C address 0x27.

const int vibrationPin = A0;

const int redLedPin = 8;

const int greenLedPin = 9;

const int buzzerPin = 10;

Defines pins for the vibration sensor and indicators.

int dangerThreshold = 400;

unsigned long dangerHoldTime = 2000;

* dangerThreshold: Minimum analog value to trigger a danger.
* dangerHoldTime: Keeps system in danger mode for at least 2 seconds.

bool dangerMode = false;

Used to track the current system state.

**🧾 Program Logic (Loop):**

1. Read analog value from vibration sensor.
2. Print value to LCD and Serial Monitor.
3. If vibration ≥ dangerThreshold, enter dangerMode.
4. Blink red LED and turn on buzzer during danger mode.
5. Hold danger mode for 2 seconds even if vibration drops.
6. If safe, turn off buzzer and red LED, turn on green LED.

**🖥 Serial Monitor Output:**

System Ready...

Vibration Value: 322

Vibration Value: 365

Vibration Value: 452

**📊 Serial Plotter Support:**

To visualize the sensor readings:

1. Go to **Tools > Serial Plotter** in the Arduino IDE.
2. Set baud rate to 9600.
3. You'll see live graphs of vibration values.

To improve plotting, add these lines (optional):

Serial.print("Vibration=");

Serial.println(vibrationValue);

**📈 Expected Behavior:**

| **Condition** | **Red LED** | **Green LED** | **Buzzer** | **LCD Status** |
| --- | --- | --- | --- | --- |
| Low vibration (<400) | OFF | ON | OFF | Status: SAFE |
| High vibration (≥400) | BLINK | OFF | ON | Status: DANGER!! |