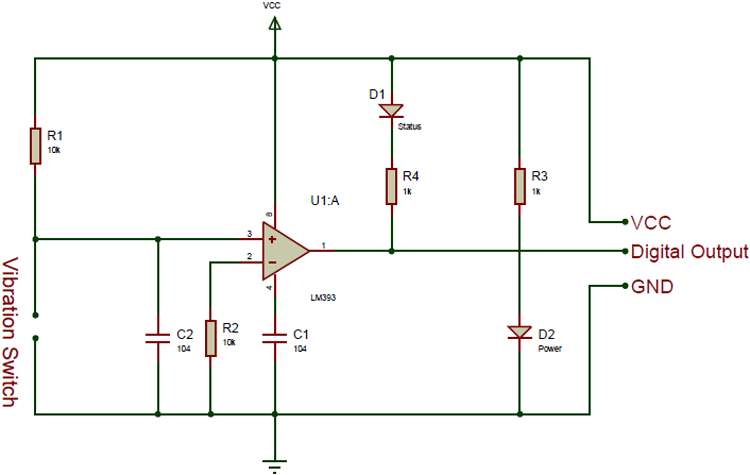
**Vibration Module**

This is an SW-420 V**ibration module**, which can work from 3.3V to the 5V. The sensor uses LM393 comparator to detect the vibration over a threshold point and provide digital data, Logic Low or Logic High, 0 or 1. **During normal operation, the sensor provides Logic Low and when the vibration is detected, the sensor provides Logic High.** There are three peripherals available in the module, two LEDs, one for the Power state and other for the sensor’s output. Additionally, a potentiometer is available which can be further used to control the threshold point of the vibration. In this project, we will use 5V to power the module.



**Vibration Sensor Module Features & Specifications**

* Operating Voltage: 3.3V to 5V DC
* Operating Current: 15mA
* Using SW-420 normally closed type vibration sensor
* LEDs indicating output and power
* LM393 based design
* Easy to use with Microcontrollers or even with normal Digital/Analog IC
* With bolt holes for easy installation
* Small, cheap and easily available



Below is the Pin Description of the Vibration sensor Module:

|  |  |
| --- | --- |
| **Pin** | **Description** |
| Vcc | 3.3 – 5V power supply |
| GND | Ground |
| Dout | Digital output |

**Components Required**

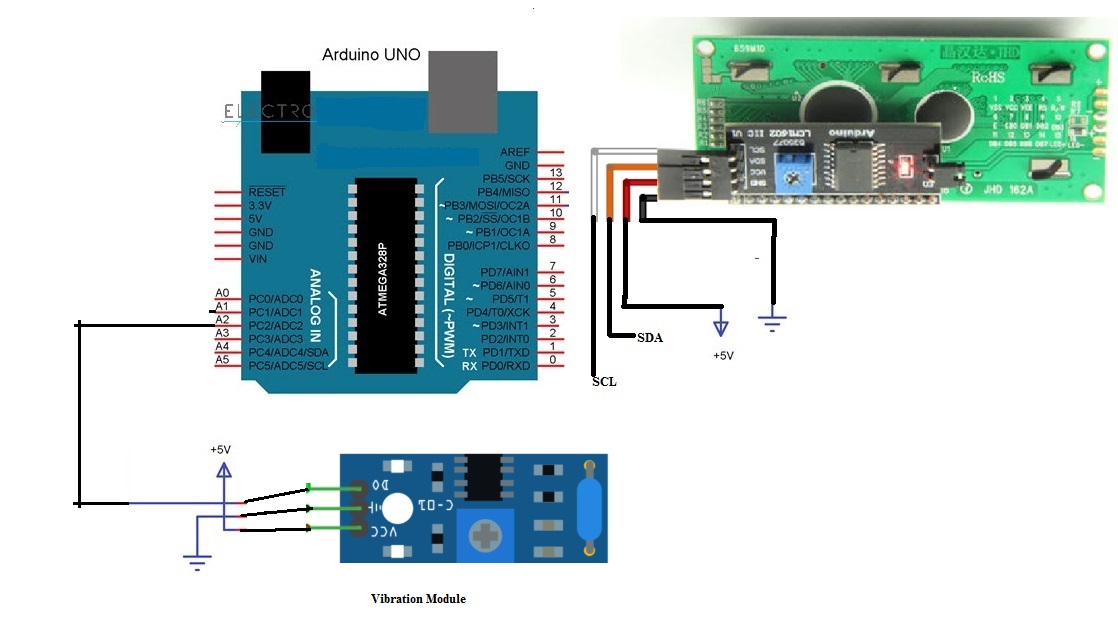
* Arduino Uno (any Arduino board can be used)
* Flame sensor module
* LED
* Buzzer
* Resistor
* Jumper wires

**Circuit Diagram of Arduino with Vibration Sensor and LCD**



[Arduino](https://circuitdigest.com/arduino-projects) Uno is a open-source microcontroller board based on the ATmega328p microcontroller. It has 14 digital pins (out of which 6 pins can be used as PWM outputs), 6 analog inputs, on-board voltage regulators etc. Arduino Uno has 32KB of flash memory, 2KB of SRAM and 1KB of EEPROM. It operates at a clock frequency of 16MHz. Arduino Uno supports Serial, I2C, SPI communication for communicating with other devices. The table below shows the technical specification of Arduino Uno.

|  |  |
| --- | --- |
| Microcontroller | ATmega328p |
| Operating voltage | 5V |
| Input Voltage | 7-12V (recommended) |
| Digital I/O pins | 14 |
| Analog pins | 6 |
| Flash memory | 32KB |
| SRAM | 2KB |
| EEPROM | 1KB |
| Clock speed | 16MHz |



I2C LCD (16X2)

**Working of Vibration Sensor with Arduino**

The **Vibration sensor detects the presence of vibration i.e. Accident**. It gives logic 1 as output if a Vibration is detected, otherwise, it gives logic 0 as output. Arduino Uno checks the logic level on the output pin of the sensor and performs further tasks such as activating the buzzer and LED, sending an alert message.

Vibration has three pins (some may have four pins): VCC, GND and DO. Connect VCC and GND to +5V and GND of the power supply (can be connected to Arduino’s +5V). the DO (short for Digital Output) is connected to Analog I/O Pin A2 of Arduino(used as Digital input with pull up resistor).

Variable “Vibration*\_val*” is used for storing the digital value read out from the Vibrtion Module. Based on this value we will detect the presence of flame.

**#define Vibration\_Pin A2**

**int Vibration\_Val ;**

**we are interfacing Vibration Module and also LCD(16X2)with Arduino. As LCD interfacing is explained ,we will concentrate on only Vibration Module interfacing.**

**void setup()**

**{**

**pinMode(Vibration\_Pin, INPUT\_PULLUP) ;**

**}**

This line of code reads the digital output from flame sensor and stores it in the variable “*Vibration\_Val*”.

**Vibration\_Val = digitalRead(Vibration\_Pin) ;**

Based on the value stored in “Vibration*\_Val*”, we have to display message on the LCD. In this part of the code, we **compare the value stored in “*Vibration\_Val*” with 0 or 1**.

**If its equal to 1**, it indicates that Vibration(Accident) has been detected. We have to show it on LCD and then send an alert.

**If its equal to 0**, then it indicates that no Vibration has been detected.This process is repeated every second to identify the presence of Vibration.

**if (Vibration\_Val == 1)**

**{**

**lcd.clear();//Clean the screen**

**lcd.setCursor(0, 0);**

**lcd.print("Accident Detected");**

**delay(1000);**

**}**

**Procedure**

1. Download and Install the Library **LiquidCrystal\_I2c** and include **wire.h library**.
2. Write the code for the Arduino UNO using Arduino IDE.
3. Connect the components as the circuit attached.
4. Edit the code as need to change the text on LCD.
5. Upload the code on Arduino UNO.
6. Now check the text on LCD and make required corrections to fit text on LCD.

**Code**

*//both LCD and Serial Interface*

*#include <Wire.h>*

*#include <LiquidCrystal\_I2C.h>*

*LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //*

*#define Vibration\_Pin A2 //Vibration module output connected to A2 pin of Arduino*

*void setup()*

*{*

*//Initialisation*

*lcd.begin(16, 2);*

*lcd.clear();*

*lcd.backlight();//Power on the back light*

*pinMode(Vibration\_Pin, INPUT\_PULLUP);*

*}*

*void loop()*

*{*

*int Vibration\_Val ;*

*Vibration\_Val=digitalRead(Vibration\_Pin);*

*if (Vibration\_Val == 1)*

*{*

*lcd.clear();//Clean the screen*

*lcd.setCursor(0, 0);*

*lcd.print("Accident Detected");*

*delay(1000);*

*}*

*delay(1000);*

*}*