

## Practical 3

### **Q3. To write a program to detects the vibration of an object with sensor using Arduino.**

To detect vibration using an Arduino in Proteus, you can use a vibration sensor module (such as a SW-420 vibration sensor) or a piezo sensor. In this example, we'll assume you have a basic vibration sensor like the SW-420, which outputs a digital signal when vibration is detected.

#### **Hardware Setup in Proteus**

##### **1. Add Components to Proteus:**

- Arduino UNO: The main controller.
- SW-420 Vibration Sensor Module (or an equivalent): Detects vibrations and outputs a digital signal.
- LED (optional): To indicate when a vibration is detected.
- Virtual Terminal (optional): To display messages when vibration is detected.

##### **2. Wiring Connections:**

- Connect the VCC of the vibration sensor to the 5V pin on the Arduino.
- Connect the GND of the vibration sensor to the GND pin on the Arduino.
- Connect the OUT pin of the vibration sensor to Digital Pin 2 on the Arduino.
- Connect the LED anode to Digital Pin 13 on the Arduino and the cathode to GND through a 220-ohm resistor (optional for visual indication).
- Optionally, add a Virtual Terminal to TX on the Arduino for debugging messages.

## Arduino Code

Here's an Arduino code to read the vibration sensor output. The program will turn on the LED when a vibration is detected and output a message on the Serial Monitor or Virtual Terminal.

```
#define VIBRATION_SENSOR_PIN 2 // Digital pin connected to the vibration
sensor

#define LED_PIN 13           // Digital pin connected to the LED

void setup() {

    pinMode(VIBRATION_SENSOR_PIN, INPUT); // Set the vibration sensor pin as
input

    pinMode(LED_PIN, OUTPUT);           // Set the LED pin as output

    Serial.begin(9600);                // Initialize serial communication at 9600 bps

    Serial.println("Vibration Detection Initialized...");

}

void loop() {

    int vibrationState = digitalRead(VIBRATION_SENSOR_PIN); // Read the vibration
sensor state

    if (vibrationState == HIGH) {      // If vibration is detected
```

```
digitalWrite(LED_PIN, HIGH);    // Turn on the LED

Serial.println("Vibration detected!"); // Print message to Serial Monitor

} else {

    digitalWrite(LED_PIN, LOW);    // Turn off the LED

}

delay(100); // Small delay to stabilize readings

}
```

### **Code Explanation**

1. ``digitalRead(VIBRATION_SENSOR_PIN)``:

- Reads the digital output from the vibration sensor. If vibration is detected, the sensor outputs ``HIGH``; otherwise, it outputs ``LOW``.

2. ``digitalWrite(LED_PIN, HIGH/LOW)``:

- Turns on the LED when vibration is detected and turns it off when no vibration is detected.

3. ``Serial.println("Vibration detected!")``:

- Sends a message to the Serial Monitor (or Virtual Terminal in Proteus) whenever vibration is detected.

4. ``delay(100);``:

- Adds a small delay to stabilize readings.

### **Simulation Steps in Proteus**

### **1. Compile the Code in Arduino IDE:**

- Open the Arduino IDE, paste the code, and select the Arduino UNO board from Tools > Board.
- Compile the code, and the Arduino IDE will generate a `.hex` file in the temporary folder. You can find this path by enabling Show verbose output during: compilation in File > Preferences.

### **2. Load the `.hex` File into Proteus:**

- Open your Proteus project.
- Double-click the Arduino UNO module to open its properties.
- Load the compiled `.hex` file from the Arduino IDE into the Program File field.
- Click OK to save the changes.

### **3. Run the Simulation:**

- Start the simulation in Proteus.
- If the vibration sensor module detects a vibration (you may need to trigger it manually in Proteus or adjust its parameters), the LED on pin 13 should light up.
- Check the Virtual Terminal or Serial Monitor for messages indicating that vibration was detected.

### **Notes**

1. Simulation Limitation: In Proteus, actual physical vibration can't be simulated, but you may be able to set the vibration sensor to manually trigger a `HIGH` output to test the code.
2. Physical Hardware Testing: For real-world testing, this code should be uploaded to an actual Arduino with a connected SW-420 vibration sensor.