Understanding sensor behind things

1. Sensor choosed: Flex Sensor 2.2

This sensor changes its resistance when bent

Uses and application: It is used to measure bending and is commonly applied in robotics, medical devices, and interactive systems.

2.Specifications:

- Operating Voltage: 0V - 5V
- Resistance: ~10kΩ (flat) to higher values when bent
- Length: 2.2 inches
- Output Type: Variable resistance

3. Working Principle:

The flex sensor operates on the principle of **resistive variation**. As the sensor bends, its resistance increases proportionally to the bend angle. This change in resistance is converted into voltage variations, which can be measured.

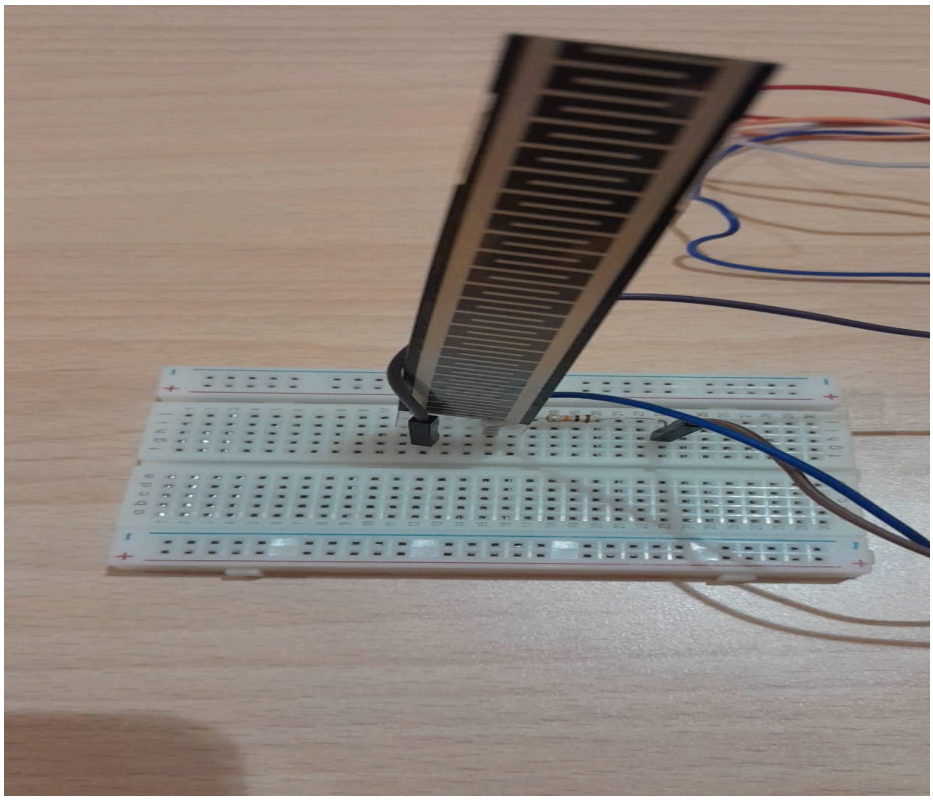
4. Capturing Analog Readings:

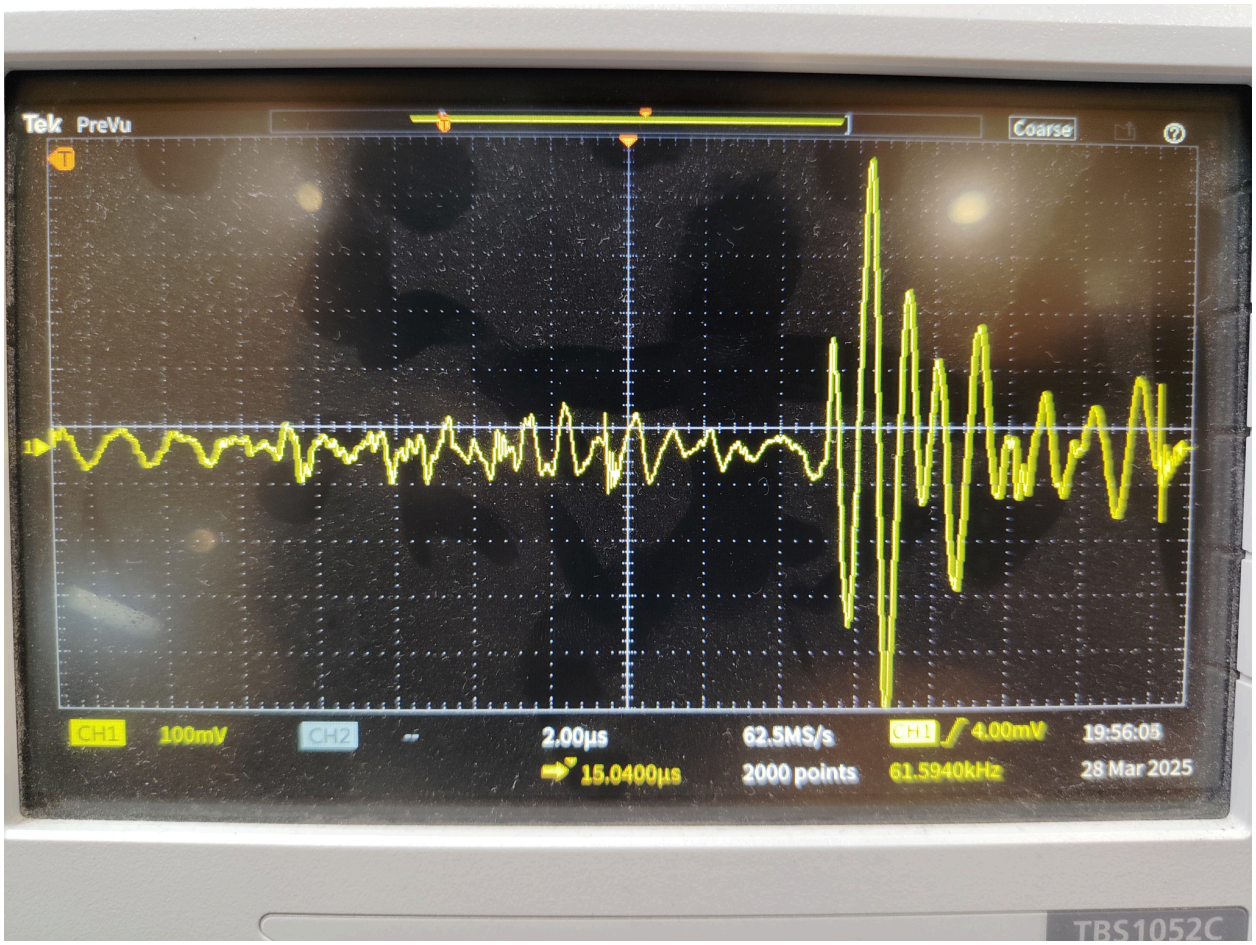
Components needed: Flex sensor, 10k ohm resistor, DSO, Power source, bread board, jumper wires (male to male)

The voltage across the sensor was observed, revealing a **sine wave output** corresponding to the bending motion.

Circuit connection: Connect one end of the flex sensor to 5V, the other end of the flex sensor to one leg of the 10kΩ resistor.Probe + of the DSO**.**The other end of the 10kΩ resistor should go to GND

Probe + (Positive Lead): Connect to the junction between the flex sensor and resistor.Probe - (Ground Lead): Connect to GND.





This picture shows the output when 5V supply is given to the circuit.

Observation:

When the sensor is bent the output signal varies.

5. Integration with ESP8266:

The flex sensor was interfaced with the **ESP8266MOD** to collect real-time data.

Components required: Flex sensor, 10k ohm resistor, DSO, Power source, bread board, jumper wires (male to male),ESP8266MOD, cable,PC with Arduino IDE

Circuit connection: Flex Sensor to ESP8266MOD

1.One end of the flex sensor to 3.3V.
2.Other end of the flex sensor to A0 (Analog Pin) on ESP8266.
3.Connect a 10kΩ resistor between the other end of the flex sensor and GND.

Code:

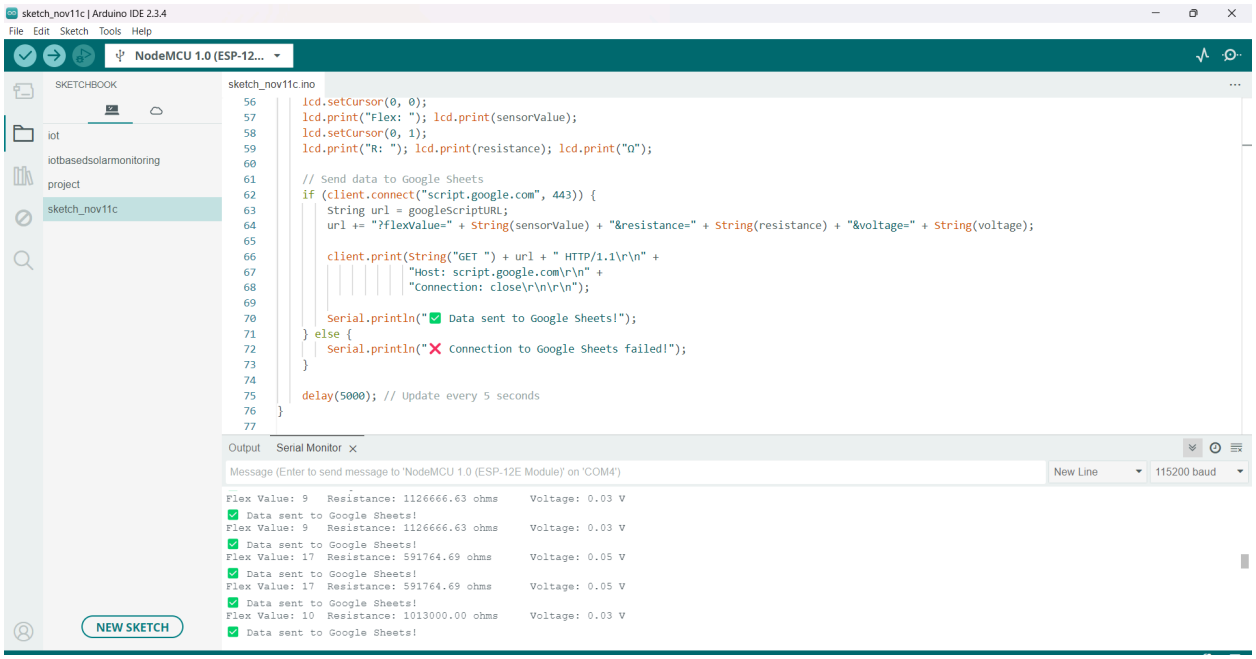
```
#include <Wire.h>

#include <LiquidCrystal_PCF8574.h>  // ✅ Correct library for ESP8266 LCD

#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>



// WiFi Credentials

const char* ssid = "Galaxy A13";

const char* password = "123abc456";



// Google Script URL

const char* googleScriptURL =
"https://github.com/electronicsguy/ESP8266/tree/master/HTTPSRedirect";



// LCD Initialization (Use I2C scanner to find correct address, usually 0x27 or 0x3F)

LiquidCrystal_PCF8574 lcd(0x27);  // Try 0x3F if 0x27 doesn't work



const int flexPin = A0; // Flex sensor connected to A0

const float R_FIXED = 10000.0; // Fixed resistor in voltage divider



WiFiClientSecure client;
```

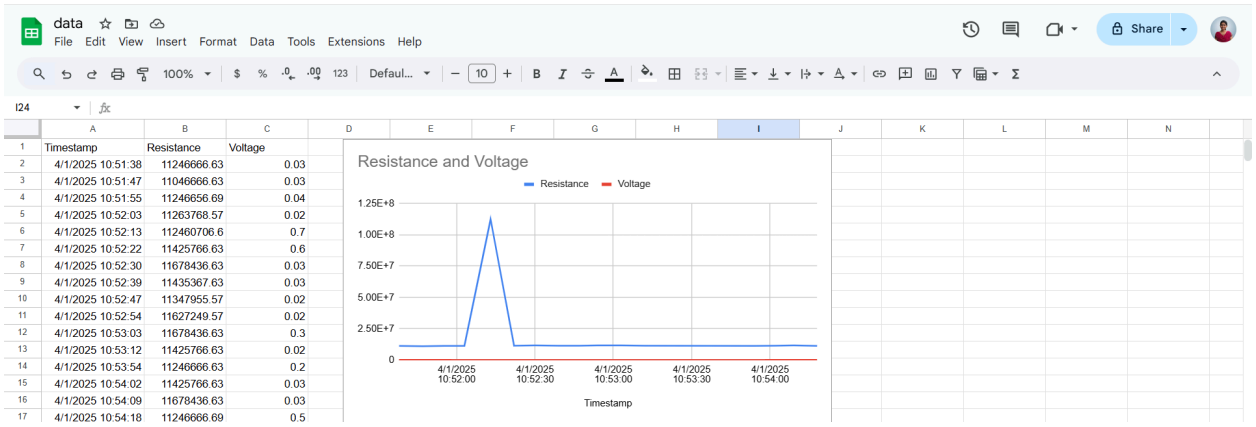
```
void setup() {
    Serial.begin(115200);
    Wire.begin();  // Initialize I2C for ESP8266

    // Initialize LCD
    lcd.begin(16, 2);
    lcd.setBacklight(255);  // ✅ Turn on LCD backlight

    // Connect to WiFi
    WiFi.begin(ssid, password);
    Serial.print("Connecting to WiFi");

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nConnected to WiFi!");

    client.setInsecure(); // Allows HTTPS connection without SSL certificate verification
}

void loop() {
    // Read sensor values
    int sensorValue = analogRead(flexPin);
```

```
    float voltage = sensorValue * (3.3 / 1023.0);
    float resistance = (R_FIXED * (3.3 - voltage)) / voltage;

    // Print to Serial Monitor
    Serial.print("Flex Value: "); Serial.print(sensorValue);
    Serial.print("\tResistance: "); Serial.print(resistance);
    Serial.print(" ohms\tVoltage: "); Serial.print(voltage);
    Serial.println(" V");

    // Display on LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Flex: "); lcd.print(sensorValue);
    lcd.setCursor(0, 1);
    lcd.print("R: "); lcd.print(resistance); lcd.print("Ω");

    // Send data to Google Sheets
    if (client.connect("script.google.com", 443)) {
        String url = googleScriptURL;
        url += "?flexValue=" + String(sensorValue) + "&resistance=" + String(resistance) +
"&voltage=" + String(voltage);

        client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                     "Host: script.google.com\r\n" +
                     "Connection: close\r\n\r\n");
```

```
        Serial.println("✅ Data sent to Google Sheets!");
    } else {
        Serial.println("❌ Connection to Google Sheets failed!");
    }

    delay(5000); // Update every 5 seconds
}
```

Serial monitor output:





Timestamp	Resistance	Voltage
4/1/2025 10:51:38	11246666.63	0.03
4/1/2025 10:51:47	11046666.63	0.03
4/1/2025 10:51:55	11246656.69	0.04
4/1/2025 10:52:03	11263768.57	0.02
4/1/2025 10:52:13	112460706.6	0.7
4/1/2025 10:52:22	11425766.63	0.6
4/1/2025 10:52:30	11678436.63	0.03
4/1/2025 10:52:39	11435367.63	0.03
4/1/2025 10:52:47	11347955.57	0.02
4/1/2025 10:52:54	11627249.57	0.02
4/1/2025 10:53:03	11678436.63	0.3
4/1/2025 10:53:12	11425766.63	0.02
4/1/2025 10:53:54	11246666.63	0.2
4/1/2025 10:54:02	11425766.63	0.03
4/1/2025 10:54:09	11678436.63	0.03
4/1/2025 10:54:18	11246666.69	0.5

6. Challenges faced:

Searching for the correct library in the Arduino IDE.
Fixing the fatal error.
Running the code in App Script.

Formula: R_flex = (R_fixed*(VCC-V_ out))/V_out

Float voltage = sensor value*(VCC/ 1023.0)