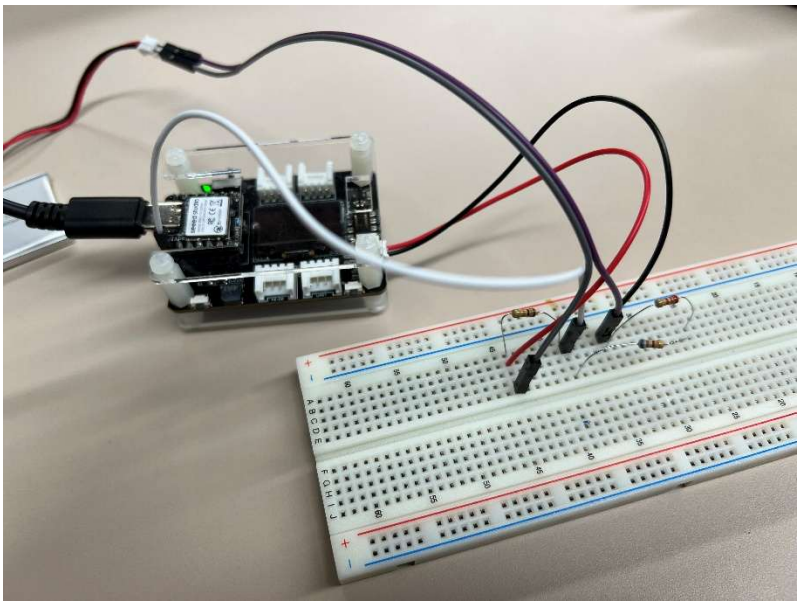
**1. Voltage Divider Implementation**

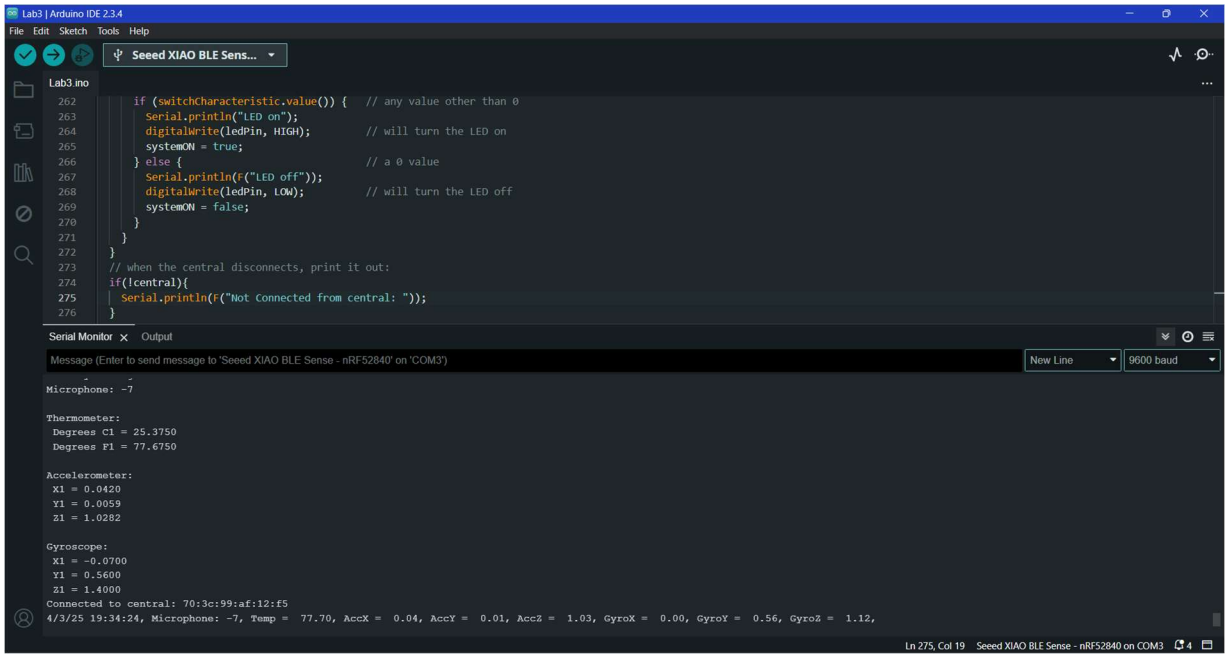


We used the following resistors for our voltage divider

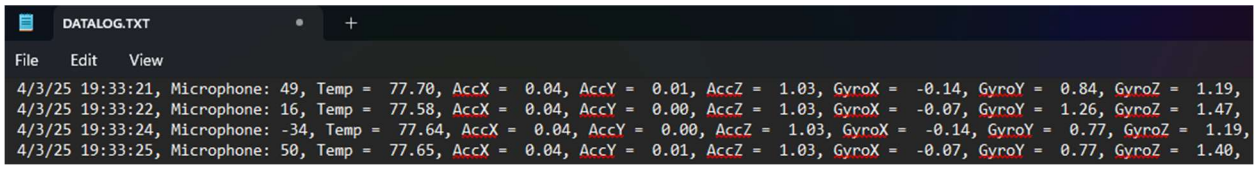
```
//battery ------------------------------------------------
// Resistor values in your voltage divider (in ohms)
const float R1 = 2880.0;  // a 2.2k and 680 resistor
const float R2 = 10000.0;  // 10kΩ resistor
```

With a $V_{in}$ of 4.2V multiplied by R2/(R1+R2) our $V_{out}$ was 3.26V which was within the 3.3V max for the ADC pin.

## 2. System Integration




Our system recorded about every 1 second: a timestamp, a microphone reading, the temperature in Fahrenheit, the accelerometer reading, and the gyroscope reading. If the system status was on it would save it to the SD card.




If the system status was off, which could be triggered via Bluetooth it would stop saving to the SD card until the status was back to on.

```
if (switchCharacteristic.written()) {
  if (switchCharacteristic.value()) {   // any value other than 0
    Serial.println("LED on");
    digitalWrite(ledPin, HIGH);         // will turn the LED on
    systemON = true;
  } else {                              // a 0 value
    Serial.println(F("LED off"));
    digitalWrite(ledPin, LOW);          // will turn the LED off
    systemON = false;
  }
```

```
4/3/25 19:33:39, Microphone: 91, Temp = 77.68,
4/3/25 19:33:40, Microphone: -21, Temp = 77.59
4/3/25 19:34:2, Microphone: -125, Temp = 77.72
4/3/25 19:34:3, Microphone: 125, Temp = 77.72,
```

**3. Battery Protection**

The system had the OLED display constantly giving a reading of the battery voltage along with the system status. If the voltage ever fell below 3.2V, the system status would turn to off and the SD card would not record.

```
if(batteryVoltage < 3.2)
  systemON = false;
```

**4. Testing and Validation (2.5 points)**

A demo video showing all features will be attached, along with our code and the DATALOG file.