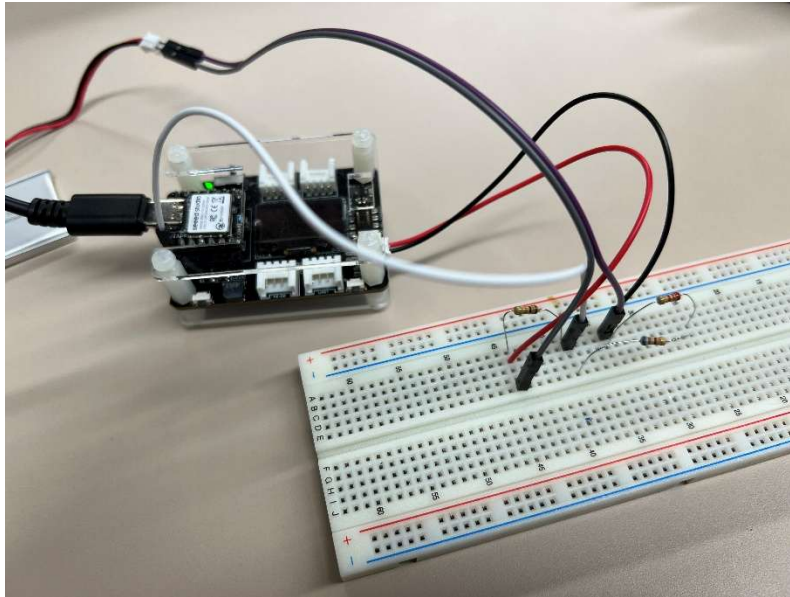


1. Voltage Divider Implementation

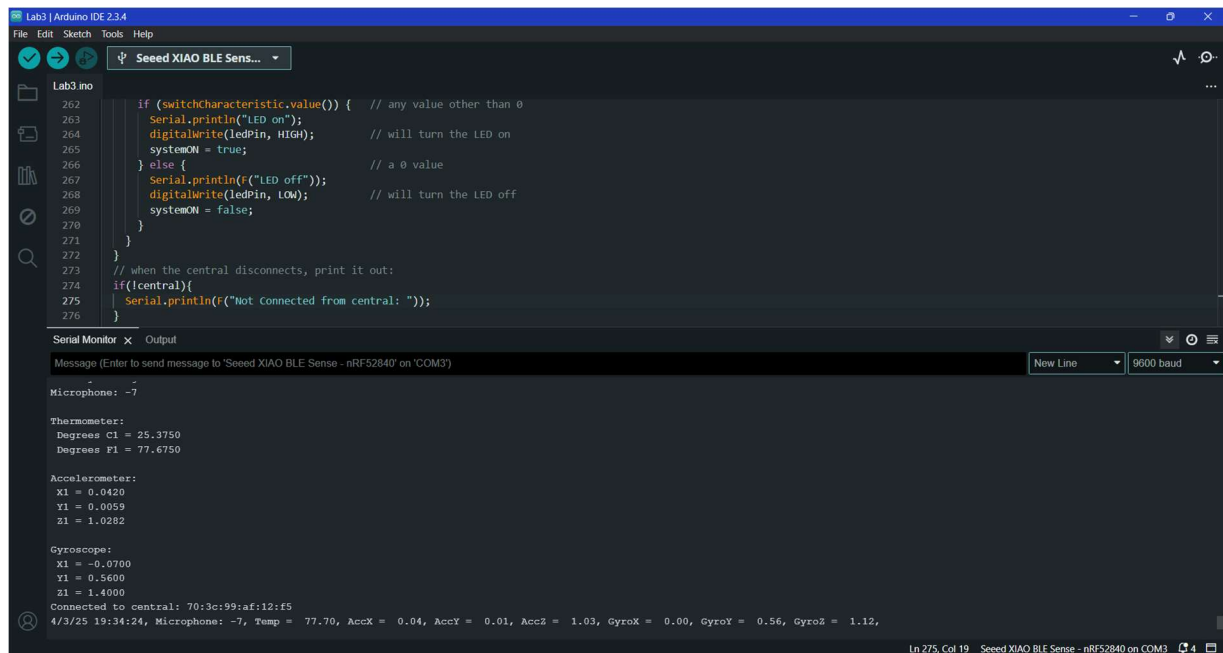


We used the following resistors for our voltage divider

```
//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

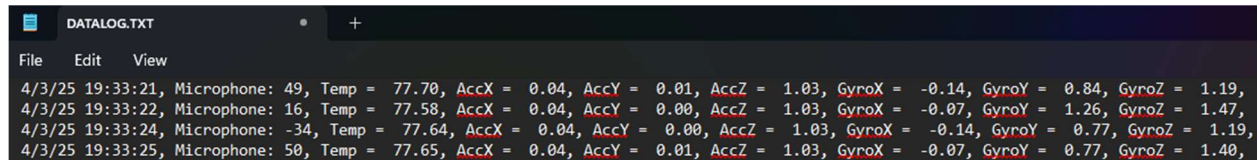


The screenshot shows the Arduino IDE interface. The top pane displays the code for Lab3.ino, which includes a switch characteristic that controls an LED and logs data when the central device disconnects. The bottom pane shows the Serial Monitor output, which displays a series of data points including a microphone reading, temperature in Celsius and Fahrenheit, accelerometer readings (X, Y, Z), and gyroscope readings (X, Y, Z). The data is logged at regular intervals, indicating a 1-second sampling rate.

```
Lab3.ino
262 if (switchCharacteristic.value()) { // any value other than 0
263   Serial.println("LED on");
264   digitalWrite(ledPin, HIGH); // will turn the LED on
265   systemON = true;
266 } else { // a 0 value
267   Serial.println(F("LED off"));
268   digitalWrite(ledPin, LOW); // will turn the LED off
269   systemON = false;
270 }
271 }
272 // when the central disconnects, print it out:
273 if(!central){
274   Serial.println(F("Not Connected from central: "));
275 }
276 }
```

Serial Monitor x Output
Message (Enter to send message to 'Seeed XIAO BLE Sense - nRF52840' on 'COM3')
New Line 9600 baud
Microphone: -7
Thermometer:
Degrees C1 = 25.3750
Degrees F1 = 77.6750
Accelerometer:
X1 = 0.0420
Y1 = 0.0059
Z1 = 1.0282
Gyroscope:
X1 = -0.0700
Y1 = 0.5600
Z1 = 1.4000
Connected to central: 70:3c:99:af:12:f5
4/3/25 19:34:24, Microphone: -7, Temp = 77.70, AccX = 0.04, AccY = 0.01, AccZ = 1.03, GyroX = 0.00, GyroY = 0.56, GyroZ = 1.12,

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.



The screenshot shows a text file named DATALOG.TXT. It contains a series of data points recorded at regular intervals. Each line represents a timestamp followed by a microphone reading, temperature, and accelerometer/gyroscope readings. The data is formatted as follows:

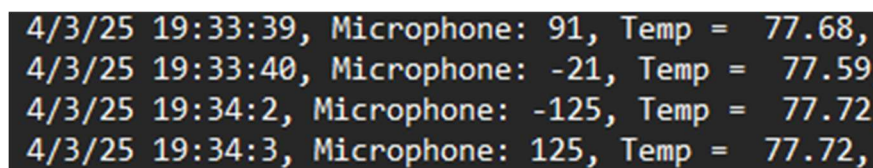
```
DATALOG.TXT
File Edit View
4/3/25 19:33:21, Microphone: 49, Temp = 77.70, AccX = 0.04, AccY = 0.01, AccZ = 1.03, GyroX = -0.14, GyroY = 0.84, GyroZ = 1.19,
4/3/25 19:33:22, Microphone: 16, Temp = 77.58, AccX = 0.04, AccY = 0.00, AccZ = 1.03, GyroX = -0.07, GyroY = 1.26, GyroZ = 1.47,
4/3/25 19:33:24, Microphone: -34, Temp = 77.64, AccX = 0.04, AccY = 0.00, AccZ = 1.03, GyroX = -0.14, GyroY = 0.77, GyroZ = 1.19,
4/3/25 19:33:25, Microphone: 50, Temp = 77.65, AccX = 0.04, AccY = 0.01, AccZ = 1.03, GyroX = -0.07, GyroY = 0.77, GyroZ = 1.40,
```

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.



The screenshot shows a code snippet from the Lab3.ino file. It defines a switch characteristic that controls an LED and logs data when the central device disconnects. The code is as follows:

```
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;
  }
}
```



The screenshot shows a text file named DATALOG.TXT. It contains a series of data points recorded at regular intervals. Each line represents a timestamp followed by a microphone reading, temperature, and accelerometer/gyroscope readings. The data is formatted as follows:

```
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.