

Department of Electrical and Computer Engineering

University of Colorado at Boulder

ECEN5730 - Practical PCB design



Lab:16, Differential and SingleEnded measurements

Report



Submitted by
Parth Thakkar

Submitted on
March 17, 2024

Contents

List of Figures	1
List of Tables	1
1 Objective / Purpose of Lab	2
2 Component listing	2
3 Explanation	2
4 Procedure	3
5 Code	4
6 Measurement	5
7 Learnings and Observation	6
8 Conclusion	7

List of Figures

1	PCB	2
2	Frequency	5
3	Differential reading	6

List of Tables

1	Component list	2
---	--------------------------	---

1 Objective / Purpose of Lab

- To distinguish between the methodologies of differential and single-ended signal measurements, pinpointing their individual merits and drawbacks. This objective was achieved through the utilization of a TMP36 temperature sensor and an ADS1115 analog-to-digital converter (ADC).
- To examine the influence of ground noise on the integrity of signals for both differential and single-ended setups.
- ADS1115 module for data capture through the I2C communication protocol. This included adjustments in gain settings, sample rates, and addressing configurations, aimed at bolstering our proficiency in employing digital communications for sensor data gathering and ADC adjustments to achieve heightened measurement precision.
- To explore the effectiveness of averaging in diminishing noise within signal measurements. This was executed by developing software routines for collecting, averaging, and evaluating multiple ADC readings.
- To implement differential measurement strategies for achieving improved signal clarity in scenarios affected by ground noise, underscoring the practical advantages of differential signaling in electronic circuit design.

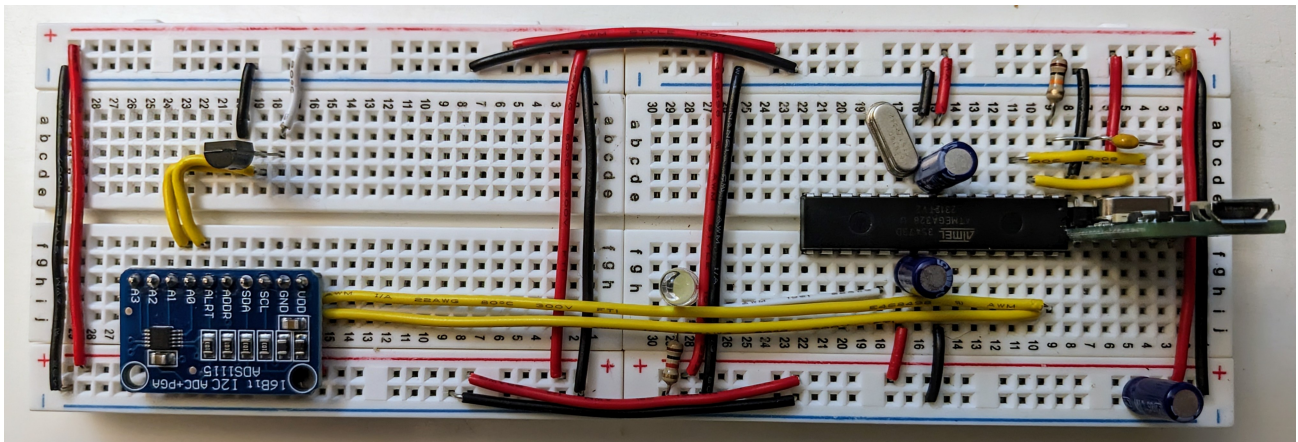


Figure 1: PCB

2 Component listing

	Component Name
1	Digital Multimeter (DMM)
2	ADS1115 ADC sensor
2	TMP36 Temperature Sensor
2	Solderless Breadboard
2	Function Generator

Table 1: Component list

3 Explanation

Single-Ended Signaling Single-ended signaling is a method of measuring or transmitting information where the signal is represented as a voltage relative to a common ground. In this setup, each signal requires

one wire plus a ground connection. The simplicity of single-ended signaling makes it appealing for many applications; however, it is inherently more susceptible to noise. Noise can be introduced into the system from various sources, including electromagnetic interference (EMI) from nearby electrical devices or inherent noise within the circuit itself. Since the signal is measured against a common ground, any fluctuation in the ground voltage due to noise directly affects the accuracy of the signal measurement.

Differential Signaling Differential signaling, in contrast, utilizes pairs of wires to transmit signals, with each wire carrying a signal that is the inverse of the other. The actual signal is the difference in voltage between these two wires, which effectively doubles the signal voltage and cancels out common-mode noise. This method is inherently more resistant to external noise because any noise that is picked up on the signal path is likely to be common to both lines and is thus canceled out when the difference is taken. Differential signaling is particularly advantageous in environments with significant electrical noise or over long transmission distances where signal integrity is crucial.

ADS1115 and Differential Measurement Support The ADS1115 module is a precision, low-power, 16-bit analog-to-digital converter (ADC) that offers high-resolution measurements and supports both single-ended and differential inputs. This capability makes the ADS1115 an excellent tool for exploring the differences between single-ended and differential measurements in a lab setting. The ADS1115 features four input channels, which can be configured either as four single-ended inputs or two differential pairs, providing flexibility in measurement setups.

The module communicates with a microcontroller, such as an Arduino, via the I2C protocol, allowing for easy integration into a variety of projects. For differential measurements, the ADS1115 can compare the voltage between two inputs (e.g., AIN0 and AIN1 for one differential pair) and convert this voltage difference into a digital value. This feature is particularly useful in experiments designed to demonstrate the efficacy of differential signaling in reducing the impact of noise on signal measurements.

4 Procedure

Arduino IDE Configuration Launch the Arduino IDE on your computer. Install the Adafruit ADS1X15 library (if it's not already installed) by going to Sketch *⌵* Include Library *⌵* Manage Libraries... , then search for "Adafruit ADS1X15" and click "Install".

Wiring the Circuit

1. **Power Connections** Connect the Arduino 5V and GND to the solderless breadboard's power rails. Use red wires for 5V and black wires for GND to maintain color consistency.
2. **TMP36 Sensor Wiring** Place the TMP36 on the breadboard. Connect its left pin (facing the flat part towards you) to 5V, the middle pin to an unused row on the breadboard (for connection to the ADS1115 later), and the right pin to GND.
3. **ADS1115 Module Connections**
 - (a) Solder the header pins to the ADS1115 module if needed.
 - (b) Connect the module's VDD to 5V and GND to ground on the breadboard.
 - (c) Connect SCL to Arduino's A5 and SDA to A4.
 - (d) For differential measurements, connect AIN0 to the TMP36 middle pin and AIN1 to the TMP36 right pin.
 - (e) For single-ended measurements, connect AIN0 to the TMP36 middle pin.
4. Connect the Arduino to your computer with a USB cable.

Programming the Arduino

1. **Load the Sketch**

Open a new sketch in the Arduino IDE. At the beginning, include the Adafruit ADS1X15 library with `#include <Adafruit_ADS1015.h>`.

2. Initialize the ADS1115

In the setup() function, initialize the ADS1115 with ads.begin();. Set the gain as necessary using ads.setGain(GAIN_TWOTHIRDS); for TMP36 voltage readings.

3. Reading data

- For differential readings, use ads.readADC_Differential_0_1(); in the loop() function.
- For single-ended readings, use ads.readADC_SingleEnded(0); in the loop() function.
- Convert the raw data to voltage by applying the correct scale factor based on your gain setting.

4. Serial Output

Print the voltage readings to the Serial Monitor for observation in both differential and single-ended setups.

5. Introducing Ground Noise

To simulate ground noise, carefully inject a low-frequency signal into the ground path between the TMP36 and ADS1115 using a function generator.

5 Code

```
1  #include <Adafruit_ADS1X15.h>
2  #include <Wire.h>
3  Adafruit_ADS1115 ads;
4  // instantiates the object
5  float mV_per_ADU;
6  long n_operations = 100;
7  long iTimer0_usec;
8  long iDummy;
9  float Time_X_usec;
10 void setup() {
11   Serial.begin(1000000);
12   delay(1000); //lets the serial port get initialized
13   ads.begin(0x48); //default address with addr pin connected to gnd
14   ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V 1 bit = 3mV
15   mV_per_ADU = 0.1875; //for ADS1115
16 }
17 void loop() {
18   //Serial.println(ads.readADC_SingleEnded(0)*mV_per_ADU);
19   iTimer0_usec = micros();
20   for (int i = 1; i <= n_operations; i++) {
21     iDummy = ads.readADC_SingleEnded(0);
22   }
23   Time_X_usec = (micros() - iTimer0_usec) / (n_operations * 1.0);
24   Serial.print(" Xtime_usec: ");
25   Serial.print(Time_X_usec, 3);
26   Serial.print(" Frequency (Hz): ");
27   Serial.print(1e6 / (Time_X_usec), 3);
28   Serial.println();
29 }
```

Output :

```
1  #include <Wire.h>
2  #include <Adafruit_ADS1X15.h>
3  Adafruit_ADS1115 ads; /* Use this for the 16-bit version */
4  int iTime_ave_msec = 100; //averaging time per point
5  long iTime_stop_msec;
6  int iCounter1 = 0; //used in counting number of points
7  float V_diff_mV;
8  //used for diff voltage
9  float V_SE_mV;
```

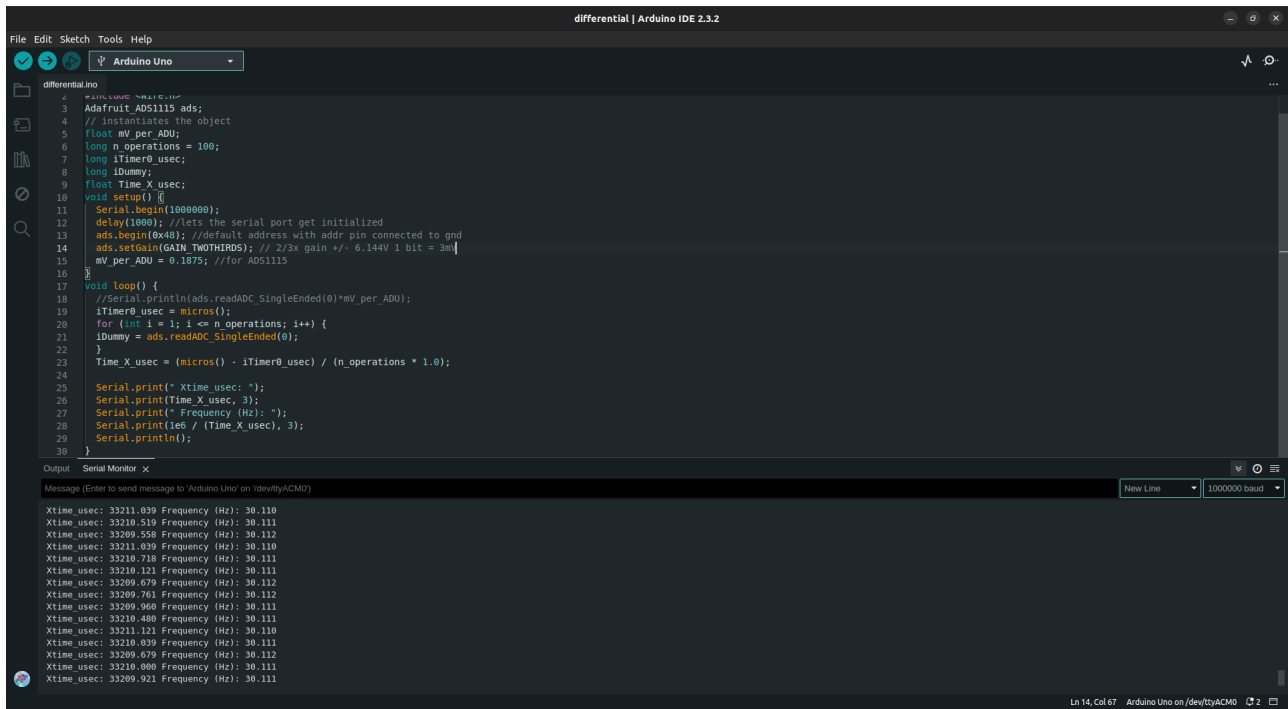


Figure 2: Frequency

```

10 // used for SE measurement
11 void setup(void) {
12   Serial.begin(2000000);
13
14   ads.setGain(GAIN_FOUR); // 4x gain
15
16   // ads.setGain(GAIN_SIXTEEN);
17   // 16x gain +/- 0.256V 1 bit = 0.125mV 0.0078125mV
18   ads.begin(); //with the ADR pin set to gnd.
19   ads.setDataRate(RATE_ADS1115_860SPS);
20 }
21 void loop(void) {
22   V_diff_mV = 0.0;
23   V_SE_mV = 0.0;
24   iCounter1 = 0;
25   iTime_stop_msec = micros() / 1000 + iTime_ave_msec;
26   while (micros() / 1000 <= iTime_stop_msec) {
27     V_diff_mV = V_diff_mV + ads.readADC_Differential_0_1() * 0.0312;
28     V_SE_mV = V_SE_mV + ads.readADC_SingleEnded(0) * 0.0312;
29     iCounter1++;
30   }
31   V_diff_mV = V_diff_mV / iCounter1;
32   V_SE_mV = V_SE_mV / iCounter1;
33   Serial.print(720);
34   Serial.print(", ");
35   Serial.print(760);
36   Serial.print(", ");
37   Serial.print(V_diff_mV);
38   Serial.print(", ");
39   Serial.println(V_SE_mV);
40 }

```

6 Measurement

Results Without Function Generator Noise: The baseline measurements showed a higher SNR in the differential configuration compared to the single-ended setup, indicating lower noise levels when differential

signaling was used.

Here we don't see much noise while there is no noise in the ground.

```
1 720, 760, 738.42, 733.62
2 720, 760, 739.49, 734.14
3 720, 760, 737.29, 736.56
4 720, 760, 738.19, 734.63
5 720, 760, 738.62, 736.36
6 720, 760, 737.39, 738.43
7 720, 760, 738.41, 741.68
8 720, 760, 739.15, 743.68
9 720, 760, 740.51, 743.33
```

With Function Generator Noise: The introduction of noise through the function generator significantly reduced the SNR in the single-ended configuration, demonstrating its susceptibility to external noise. Conversely, the differential configuration maintained a relatively high SNR, When we added noise (Sine wave of 0.1 Hz at 10V peak to peak we got this kind of signal)

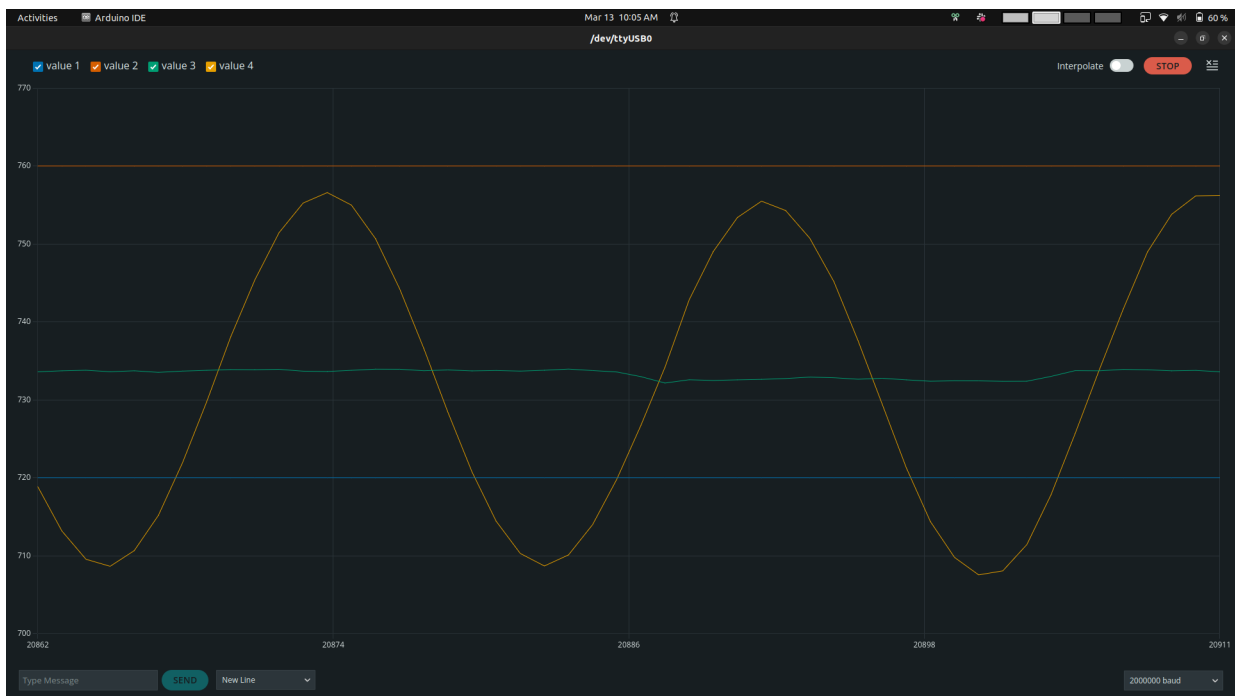


Figure 3: Differential reading

7 Learnings and Observation

1. In our lab, we looked at how outside noise, especially the kind made by a function generator, can affect the signals coming from a TMP36 temperature sensor. This test showed us that electronic signals are really sensitive to outside disturbances. We saw that noise could really lower the quality of our signal measurements, making our data less accurate and reliable.
2. We also found out that using differential signaling is much better for dealing with noise than the single-ended method. When we compared the two, the differential setup was much better at ignoring the noise that was the same in both signals. This makes differential signaling a better choice in places with a lot of noise or when it's really important to keep the signal clear.
3. The lab also made it clear how important it is to design and lay out your circuit carefully to keep noise out. It showed how choosing the right way to measure things and how you connect your wires can make your signal clearer.
4. Getting to work directly with the ADS1115 module and the TMP36 temperature sensor.

8 Conclusion

1. In this lab, we looked at how two ways of sending signals, called differential and single-ended, work differently when there's unwanted noise. We used a temperature sensor TMP36, a 16 bit ADC ADS1115, and an Arduino to gather data. Our main goal was to see how noise affects signal quality and to show that differential signaling is better at dealing with noise than single-ended signaling.
 2. This lab also taught us that while single-ended signaling might work fine in places with very little noise, differential signaling is really important in situations where it's crucial to keep the signal clear of noise. This is an important thing to know for future projects that involve collecting data from sensors, as the choice of signaling method could greatly affect how well the project works.
 3. ADS1115 can let us easily switch between the two signaling methods to compare them under the same conditions. This showed us the benefits of using differential signaling in places where there's a lot of noise.
-