

## Lab16: Differential and Single-ended signals

### Objective:

1. The goal of **Lab 16** is to understand the **difference between single-ended and differential measurements** and to observe how **ground noise** affects analog signals.
2. With the Temperature sensor and ADC(Analog to Digital Converter), the effect of ground bounce on single-ended and differential signals is analyzed using the I<sup>2</sup>C interface of Arduino Uno.

### Components Required:

1. **TMP36** → temperature sensor providing an analog voltage
2. **ADS1115** → 16-bit ADC that can measure both single-ended and differential signals
3. **Arduino Uno** → powers everything and communicates via I<sup>2</sup>C to the ADS1115
4. Solderless Breadboard
5. Function generator

### Overview:

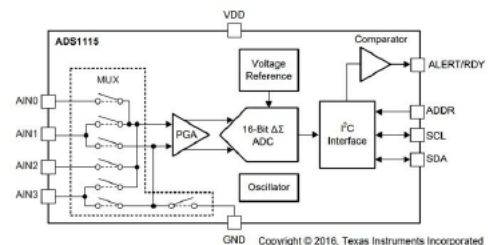
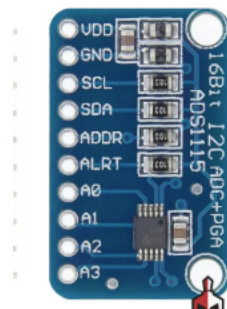
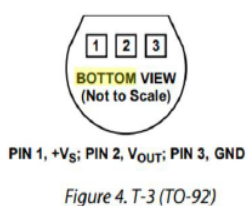


Figure 22. ADS1115 Block Diagram

FIGURE 1: TMP36

FIGURE 2: ADS1115

FIGURE 3: AD51115 Block diagram

Figure 1 and Figure 2 represent the pinouts of the Temperature sensor and ADS1115 respectively. With Arduino Uno, the 5V is supplied. With the help of code, single-ended and differential measurements are taken.

The 16-bit ADC was set to a full-scale range of  $\pm 1.024\text{V}$  which corresponds to  $31.25\mu\text{V}/\text{div}$  ( $2 \times 1.024 / 2^{16}$ ). The baud rate was set to 2000000 and the results are averaged for 100ms which are monitored and plotted using the I2C interface between Arduino Uno and ADS1115 module (internal pull-up resistor = 40Kohms)

**Connections:**

Component	Pin	Connect To	Description
<b>TMP36</b>	+Vs (pin 1)	5 V	Power input
	Vout (pin 2)	AIN0 on ADS1115	Signal output
	GND (pin 3)	Common GND	Sensor ground
<b>ADS1115</b>	VDD	5 V	Power
	GND	Common GND	Shared ground
	SDA	A4 (Arduino)	I <sup>2</sup> C data
	SCL	A5 (Arduino)	I <sup>2</sup> C clock

**I compare:**

1. **Single-ended** (AIN0 to GND) — measures the signal with respect to ground.
2. **Differential** (AIN0 – AIN1) — measures the difference between two points, canceling common-mode (ground) noise.

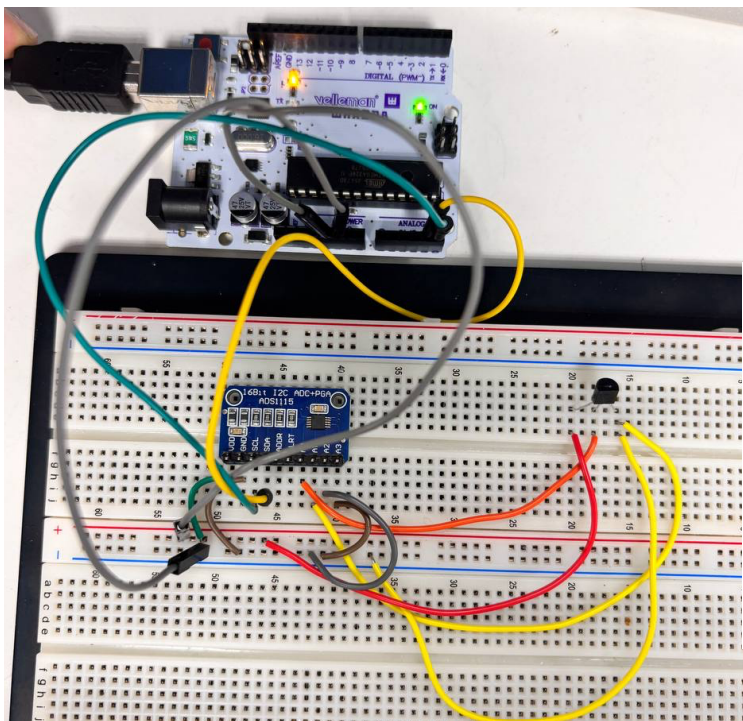


FIGURE 4: Setup to measure SE and DE signals

**Table 3. Full-Scale Range and Corresponding LSB Size**

FSR	LSB SIZE
$\pm 6.144\text{ V}^{(1)}$	187.5 $\mu\text{V}$
$\pm 4.096\text{ V}^{(1)}$	125 $\mu\text{V}$
$\pm 2.048\text{ V}$	62.5 $\mu\text{V}$
$\pm 1.024\text{ V}$	31.25 $\mu\text{V}$
$\pm 0.512\text{ V}$	15.625 $\mu\text{V}$
$\pm 0.256\text{ V}$	7.8125 $\mu\text{V}$

(1) This parameter expresses the full-scale range of the ADC scaling. Do not apply more than VDD + 0.3 V to the analog inputs of the device.

```

#include <Wire.h>
#include <Adafruit_ADS1X15.h>
Adafruit_ADS1115 ads; /* Use this for the 16-bit version */
int iTime_ave_msec = 100; //averaging time per point
long iTime_stop_msec;
int iCounter1 = 0; //used in counting number of points float V_diff_mV; //used for
diff voltage
float V_SE_mV; // used for SE measurement
float V_diff_mV;
void setup(void) { Serial.begin(2000000);
// ADS1015 ADS1115
// -----
// ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V 1 bit = 3mV 0.1875mV (default)
// ads.setGain(GAIN_ONE); // 1x gain +/- 4.096V 1 bit = 2mV 0.125mV
// ads.setGain(GAIN_TWO); // 2x gain +/- 2.048V 1 bit = 1mV 0.0625mV
ads.setGain(GAIN_FOUR); // 4x gain +/- 1.024V 1 bit = 0.5mV 0.03125mV
// ads.setGain(GAIN_EIGHT); // 8x gain +/- 0.512V 1 bit = 0.25mV 0.015625mV
// ads.setGain(GAIN_SIXTEEN); // 16x gain +/- 0.256V 1 bit = 0.125mV 0.0078125mV
ads.begin(); //with the ADR pin set to gnd.
ads.setDataRate(RATE_ADS1115_860SPS);

} void loop(void) {

V_diff_mV = 0.0;
V_SE_mV = 0.0;
iCounter1 = 0;
iTime_stop_msec = micros() / 1000 + iTime_ave_msec; while (micros() / 1000 <=
iTime_stop_msec) {
V_diff_mV = V_diff_mV + ads.readADC_Differential_0_1() * 0.0312; V_SE_mV = V_SE_mV +
ads.readADC_SingleEnded(0) * 0.0312;
iCounter1++;
}
V_diff_mV = V_diff_mV / iCounter1;
V_SE_mV = V_SE_mV / iCounter1;
Serial.print(720);
Serial.print(", ");
Serial.print(760);
Serial.print(", ");
Serial.print(V_diff_mV);
Serial.print(", ");
Serial.println(V_SE_mV);
}

```

## Measurements:

The green (Value 3 – 3rd column on serial monitor) and orange (Value 4 - 4th column on serial monitor) represent the Differential signal and single-ended signal respectively. As shown in Figure 5, with no ground noise, the differential signal as well as the single-ended signals measure: 0.734V. This converts to 23.4 degC which is the room temperature and also according to the formula (datasheet) given below.

$$T [\text{degC}] = V [\text{volts}] \times 100 [\text{degC} / \text{V}] - 50 [\text{degC}]$$

and

$$V [\text{Volts}] = T [\text{degC}] \times 0.01 [\text{Volts} / \text{degC}] + 0.5 [\text{Volts}]$$

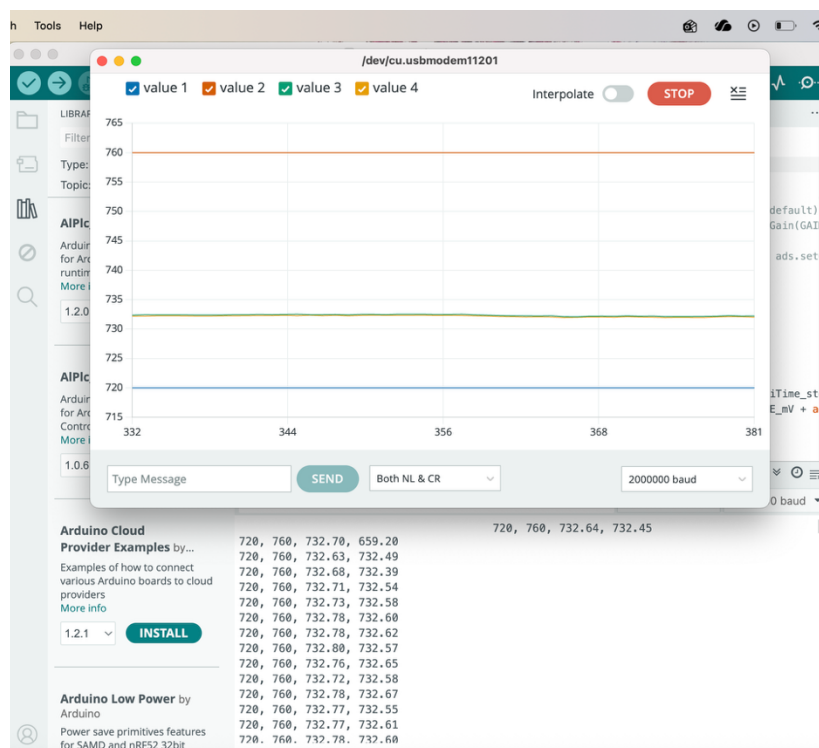


FIGURE 5: without ground noise

With the function generator (50 ohms Thevenin resistance), a  $di/dt$  was generated across the return path (0.4ohms on my SBB, using 4-wire resistance method) of TMP36 and ADS1115 using 20Vp-p, 1Hz Square wave signal (+/- 10v/50ohm= +/- 200mA). Since the differential signal measures the difference of voltages on the sensor location itself, it did not carry any change in measurement (3 rd column in Figure 6). Whereas the signal ended signal carries additional voltage drops due to local ground reference (790 and 697mV = 93mVp-p noise) which corresponds to the erroneous temperature of alternating 29degC and 19.7degC.

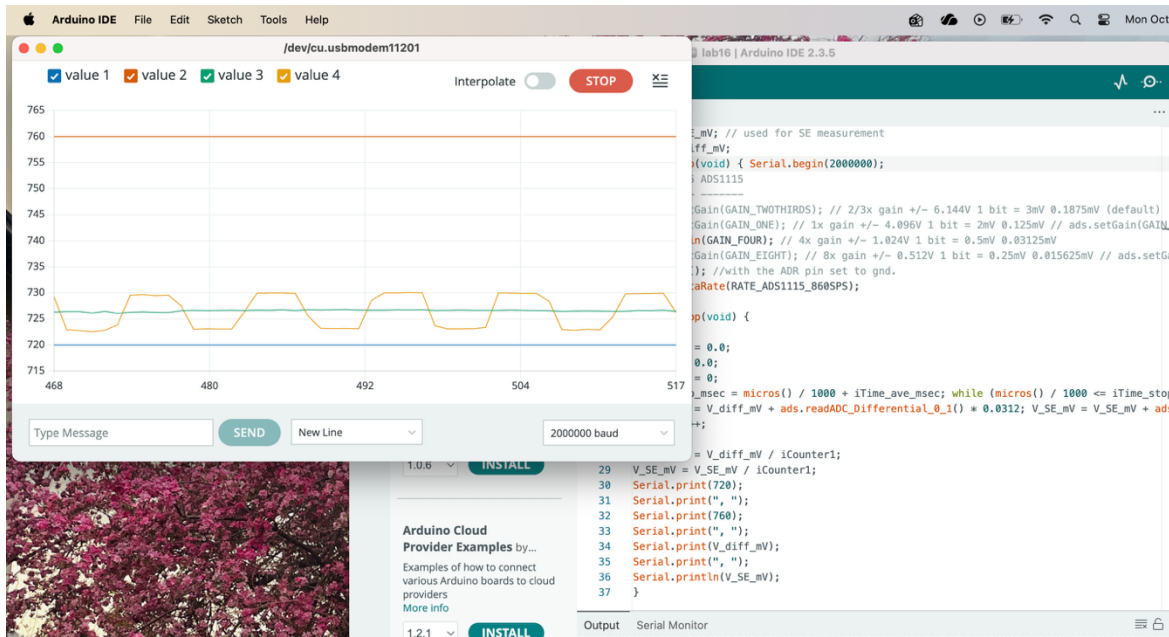


Figure 6: with ground noise

Conclusion: In a noisy ground, a differential signal gives correct measurement as it does not consider the return path voltage in its measurement and hence it is insensitive to ground bounce.

### Lab 17:

The main goal of Lab 17 is to explore and understand how the I<sup>2</sup>C communication bus operates, specifically,

-How signals behave on the SCL (clock) and SDA (data) lines, -How different pull-up resistor values and bus capacitance affect the signal timing, rise/fall times, and data integrity.

#### Rise Time Equation (10–90%)

$$t_r = 2.2 \times R \times C$$

R= pull-up resistance,  
 C= total line capacitance. (30PF)

#### I<sup>2</sup>C Bus Specification

- Standard Mode: 100 kHz → Rise time ≤ 1 μs
- Fast Mode: 400 kHz → Rise time ≤ 0.3 μs

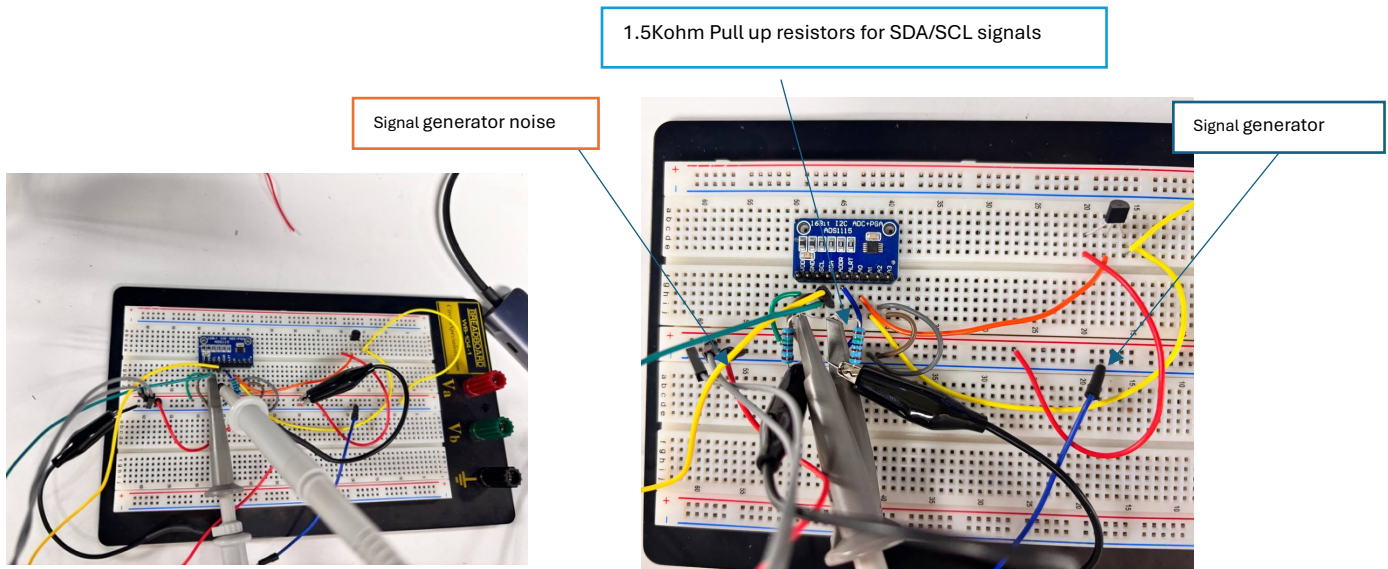


FIGURE 8: Before adding pull-ups

### Figure 8 (Before Pull-Up):

The waveforms rise slowly and have rounded tops. This happens because the I<sup>2</sup>C lines rely only on the weak internal pull-up resistors of the Arduino (about 40 kΩ). These resistors cannot charge the line capacitance quickly, causing slow rise times, distorted logic levels, and potential timing errors during communication.

### Figure 7 (After Pull-Up):

After adding external 1.5 kΩ pull-up resistors, the signal edges become much sharper and cleaner. The lines now reach the full logic-high level faster, reducing the effect of noise and improving timing accuracy. The faster rise times also make the I<sup>2</sup>C bus more reliable at 100 kHz speed.

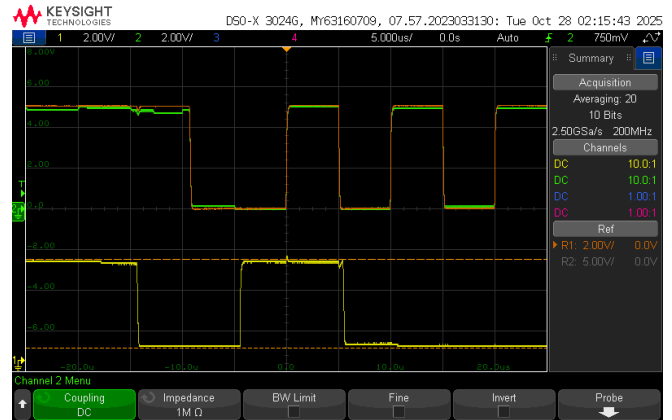


FIGURE 7: after adding pull-ups

**Conclusion:**

When no external pull-ups are used, the I<sup>2</sup>C signals (SCL and SDA) rise slowly and can pick up noise because of the weak internal resistors.

After adding 10 kΩ pull-up resistors, the lines charge faster, producing sharp, clean transitions and improving communication reliability.

For my Golden Arduino I will use 1-1.5kohm pull up resistor.