Principles of Measurement & Instrumentation I Laboratory

PHYS417

Laboratory Report

Abdullah Burkan BEREKETOĞLU

Experiment 4 - Communication Protocols: I2C & SPI Compatible Sensors

Submitted By: Abdullah Burkan BEREKETOĞLU

Submitted to: Deniz Orhun BOZTEMUR & Enver BULUR

Student Id: 2355170

Experiment Date: 16.1.2022 (Make-up)

Submission Date: 19.1.2022

OBJECTIVES

Aim of this experiment is to build a controllable-gain amplifier circuitry, then measure temperature using I^2C or SPI compatible sensor module. Lastly, we want to finish with measuring the acceleration of the motion we do in the system, in further studies being able to know the basics of I^2C/SPI Communication Protocols.

INTRODUCTION

Communication networks consist of protocols which are some sets of instructions and conventions which regulate the data exchange between the ports. All Internet of Things devices have the binary protocols that regulate their data delivery system, and as known binary is only 0s and 1s.

Communication networks are also can be separated into two with their clock synchronization type, some devices in these systems have asynchronous clocks which will get the time from one of the delivery pipelines, but some of these devices may need cables or bluetooth to calibrate their system time. Some devices in these systems can have higher authorizations, with an hierarchical level, but one device that is the admin/master, has the authority to communicate with all devices and these devices who are below the master is slave for the communication networks master. This master can be chosen by the end-user or developer and it's duty is to determine the information exchanges, and the communications at all levels. One can think this as main branch of the GitHub and there are only certain authorizations for the mergers due to not make a collision, which can also happen if the lower hierarchies deliver data on their own.

The speed of the communication that is made by the authorization of master is determined by the physical properties of the environment and the system such as resistance, humidity, capacitance, inductance, etc.) also there can be effects of the resistivity of the cable, air and so as said environmental effects. So for to reach high speeds the transmission needed to be done in cables that can carry high-frequency signals or the data may be lost in the middle of the delivery from the pipeline. Most systems that work at high speed such as HPC systems are tuned and expensive materials are used to prevent this package loss at the delivery.

EQUIPMENT

- Function Generator
- Oscilloscope
- Arduino UNO
- Mechanical Potentiometer of 100K**Ω**
- X9C104 Aether digital potentiometer of 100K**Ω**
- MAX6675 temperature sensor module
- GY-521 MPU6050 3-Axis Acceleration Gyroscope 6DOF Module (regular one that is given by the supplier doesn't work since other students tried I was assigned to use GY - 521)
- Lighter
- LCD with I^2C Serial Interface Board Module
-

PROCEDURE

1-) In the first part of the experiment it is wanted from the experimenter (us), that to make a controllable gain amplifier circuit with a mechanical potentiometer except the capacitor in the system, to be able to clearly see the difference for why do we have capacitor, also what to do when we add digital potentiometer.

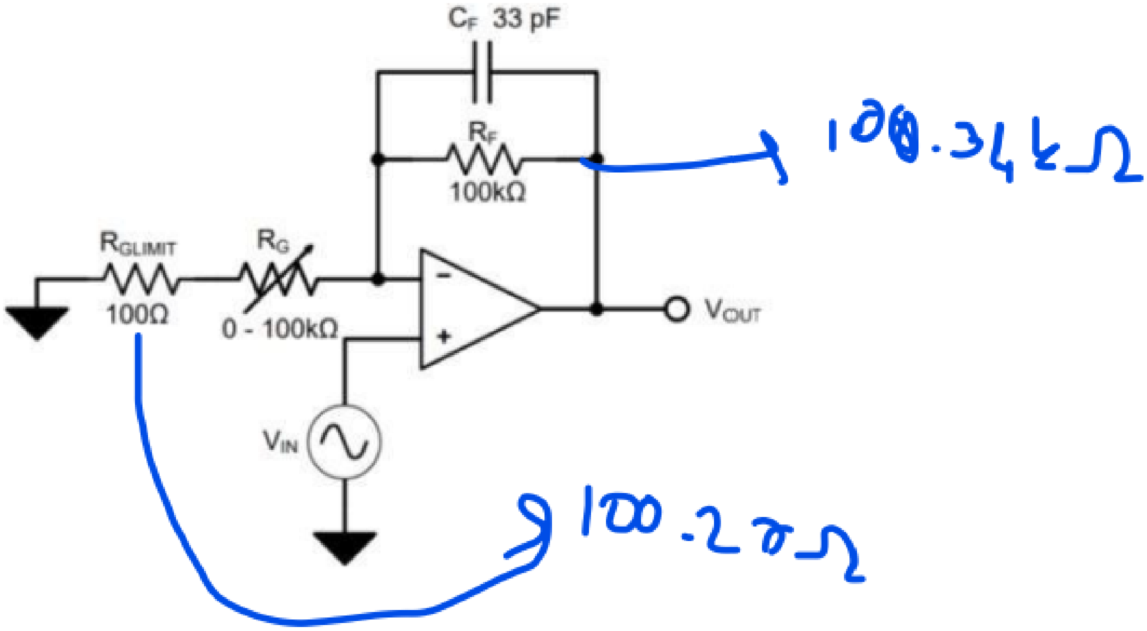

Figure 1. Controllable gain amplifier

First, we did without adding the capacitor as said above to see the shapes and the issue faster and without filtering. The input signal was lowest for my case with 1kHz and Vpp = changed often due to an error that we couldn't fix at the digital potentiometer part of the controllable gain amplifier can accept it (2mV ~ 4mV), but Vcc was +5 to -5, since Vpp at minimum we can't clearly say that it is giving 2mV or 4mV.

The result without a capacitor at the bottom is 735 mV and the top voltage is 3.63 V

With adding the capacitor we have only one picture attained from the oscilloscope which shows that since it had higher DC/offset voltage and there is capacitor gain was high. We do use capacitors to prevent DC/offset to mix with AC due to its mechanistic effects capacitors act as filters or isolators in amplifier design. Due to this filtering harmonic waves will be produced if the capacitor is put as given in figure 1.

By the calculations from the formula that is given for RG = Min, we can get 3.63 which is due to the potentiometer and Arduino also consumes some voltage which deduces from the total voltage. It is seen in figure 3. Also in figure 1, we can see that at RG = MAX voltage is still have some gain, but it looks more than the theoretical value since the Vpp changed too often too high and low values while taking a screenshot by python, we cannot directly say that gain is really really higher tan the expected value, which is not true for this case.

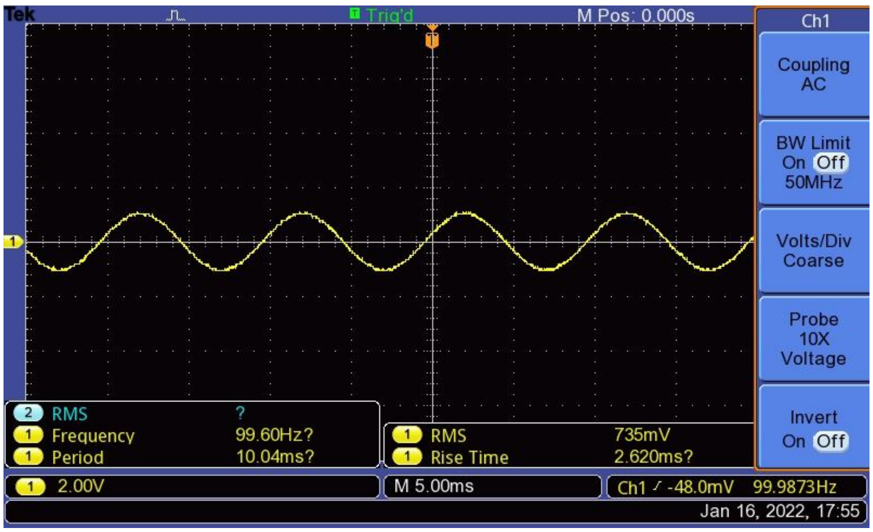

Figure 2. RG=MAX oscillation and gain finding no capacitor

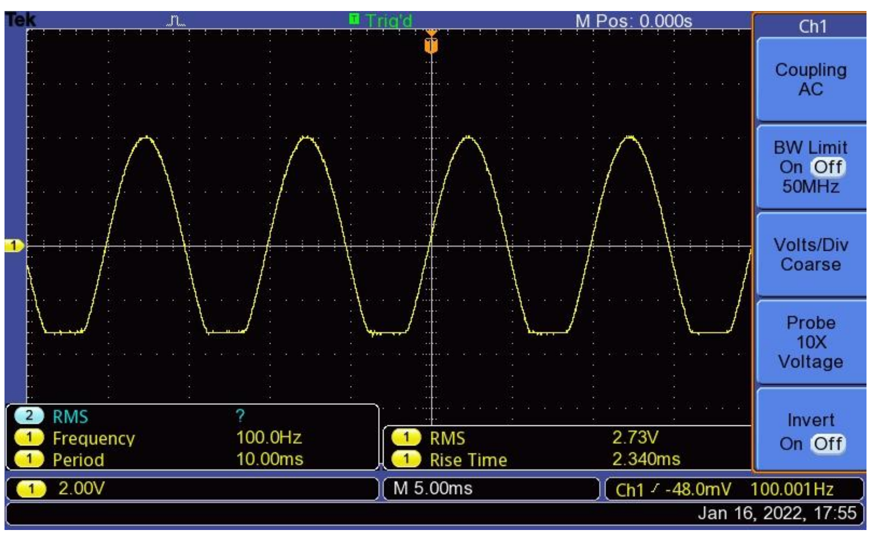

Figure 3. Between RG=MAX and RG = LOW oscillation and gain finding no capacitor

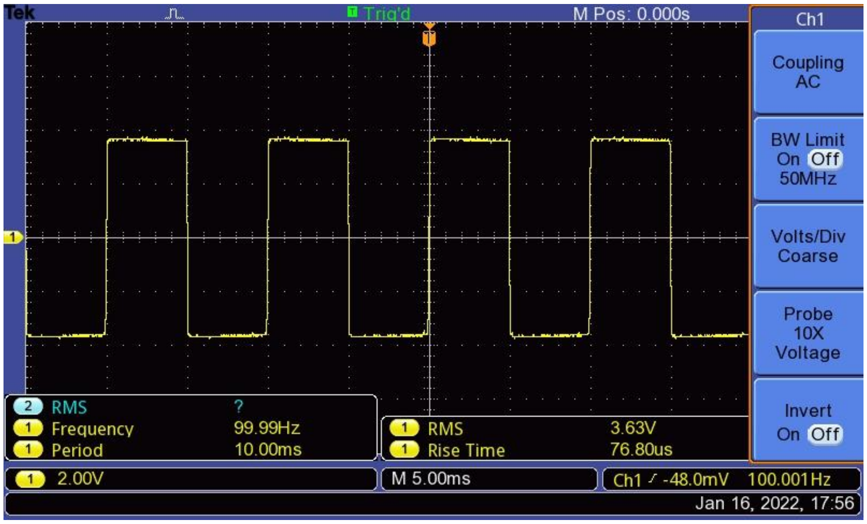

Figure 4. RG = LOW no capacitor

The three photos that are collected from the oscilloscope shows us that without the capacitor in the circuit we have high gain, what I mean by that is when the signal turns into square RMS is now 3.63 at 2 mV function generator voltage, which is 1815 times amplified voltage or 1815 gain. When it turns into a square that means the potentiometer is now at full resistance, and when it goes sinusoidal waveform, that means the potentiometer now has high resistance or near 100k and the gain lowers. As it can be seen from figure 2 on page 3 it is similar to a sinusoidal waveform hence we can deduce that the resistance is high at the potentiometer, also that corresponds to gain is less in figure 2.

1-B) Here we will look at the Programmable Controllable-Gain Amplifier with a digital potentiometer

In figure 5 that is shown below in the figure of when we switched the mechanical potentiometer with the digital one and added the capacitor later on to program the digital pot we used an Arduino code and used FastX9CXXX.h

We had many issues with the digital potentiometer it sometimes worked nicely but sometimes didn't. Still, we could see the gain increase with offset increasing also the Vpp.

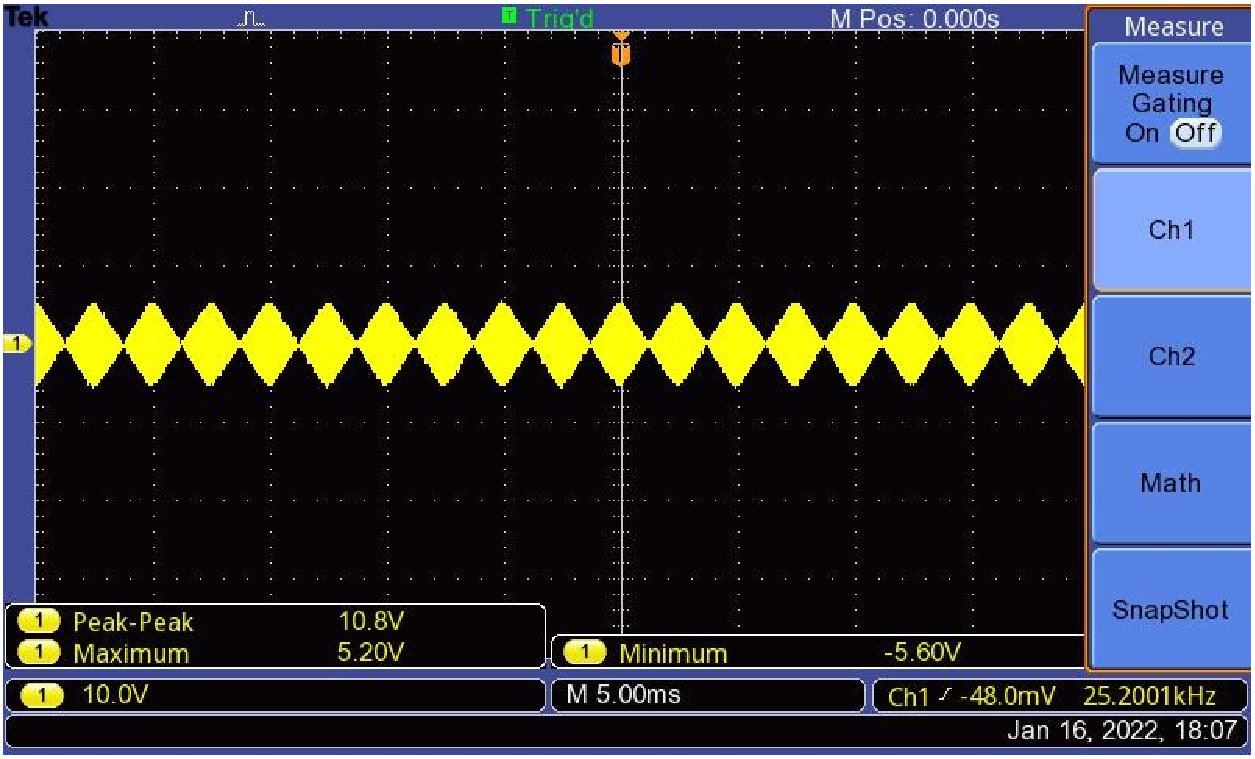

Figure 5 with capacitor and digital potentiometer results

```
///
/// Example project to control a X9C102 digital potentiometer.
///void JumpToStep(const uint8 step, const bool store = false)
///void Reset()
///void Down(const bool store = false)
///void Up(const bool store = false)
///void Store()
///GetStep()
///

#include <FastX9CXXX.h>

#define X9_CS_PIN 3
#define X9_UD_PIN 4
#define X9_INC_PIN 5

FastX9C104 Potentiometer;
int StepNow;

void setup() {
  Serial.begin(9600);
  Serial.println();
  Serial.print(F("X9C102 Digital Potentiometer setup..."));

  Potentiometer.Setup(X9_CS_PIN,X9_UD_PIN,X9_INC_PIN);

  // Reset potentiometer to known position (Step == 0).
  Potentiometer.Reset();

  Serial.println(F("complete."));
}

void loop() {
 ///
 Potentiometer.JumpToStep(random(X9CXXX::X9_STEPS));

 StepNow = Potentiometer.GetStep();
 Serial.print(F("Potentiometer current resistance: "));

 Serial.println(StepNow);
 Potentiometer.JumpToStep(50);
 delay(10);  // wait for a second
}
```

Figure 6 Arduino Code for Digital Potentiometer 0 - 100 steps

2-) Design of Digital Thermometer Using a Thermocouple Sensor

Here in this part, we used the MAX6675 thermocouple sensor and its Arduino library to measure temperature from a liquid crystal display LCD. We used max6675.h library, liquidcrystal library that one of which made it possible to connect the I^2C protocol and LCD. also Wire.h

```
#include <LiquidCrystal.h>

#include <max6675.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// Set the LCD address to 0x3F for a 16 chars and 2 line display

LiquidCrystal_I2C lcd(0x27, 16, 2);
// end of settings for LCD1602 with I2C

int soPin = 8;// SO=Serial Out
int csPin = 9;// CS = chip select CS pin
int sckPin = 10;// SCK = Serial Clock pin

MAX6675 thermocouple(sckPin, csPin, soPin);

void setup() {

  lcd.init();
  lcd.backlight();// turn the backlight ON for the LCD
  lcd.print("S.A.");
  lcd.setCursor(0, 1);
  lcd.print("Thermocouple");

  Serial.begin(9600);// initialize serial monitor with 9600 baud
  Serial.println("S.A.");

  delay(3000);

}

void loop() {


  Serial.print("C = ");
  Serial.println(thermocouple.readCelsius());


  lcd.clear();// clear previous values from screen
  lcd.setCursor(0, 0); // set cursor at character 0, line 0
  lcd.print("Temperature");


  lcd.setCursor(0, 1); // set cursor at character 0, line 1
  lcd.print(thermocouple.readCelsius());
  lcd.setCursor(5, 1); // set cursor at character 9, line 1
  lcd.print((char)223);
  lcd.setCursor(6, 1); // set cursor at character 9, line 1
  lcd.print("C");


  delay(1000);
```

Figure 7 Thermocouple and Display code for Arduino

The only extra thing that we needed to do is to adjust the contrast since I couldn't see the screen letters in the LCD I thought there are no words and the code is broken, but the contrast was too bright thanks to Deniz Boz hocam he fixed contrast issue from the behind of the LCD display, also showed me where to change the contrast if ever needed.

The program has clearly shown us the temperature change in Celsius, as it is said the only issue was the contrast in this part. From the lcd.print(thermocouple.readCelsius()); we read the temperature value then add C to the end.

3-) Measurement of Mechanical Vibration using 3D Digital Accelerometer

Here in this part since the Accelerometer that is needed to be used is changed we used GY-521 which didn't need the LSM303.h library to visualize and read the accelerometer value. Since LSM303.h is set for LSM303D, there is no need for that.

After I ran the code that is below and shook the accelerometer on my hand on every axis I got disruptions on the plot as it will be seen in the figures 9 - 10.

```
#include<Wire.h>
const int MPU=0x68;
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

void setup(){
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
}
void loop(){
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,12,true);
  AcX=Wire.read()<<8|Wire.read();
  AcY=Wire.read()<<8|Wire.read();
  AcZ=Wire.read()<<8|Wire.read();
  GyX=Wire.read()<<8|Wire.read();
  GyY=Wire.read()<<8|Wire.read();
  GyZ=Wire.read()<<8|Wire.read();

  Serial.print("Accelerometer: ");
  Serial.print("X = "); Serial.print(AcX);
  Serial.print(" | Y = "); Serial.print(AcY);
  Serial.print(" | Z = "); Serial.println(AcZ);

  Serial.print("Gyroscope: ");
  Serial.print("X = "); Serial.print(GyX);
  Serial.print(" | Y = "); Serial.print(GyY);
  Serial.print(" | Z = "); Serial.println(GyZ);
  Serial.println(" ");
  delay(333);
}
```

Figure 8 Accelerometer GY-521 code for Arduino

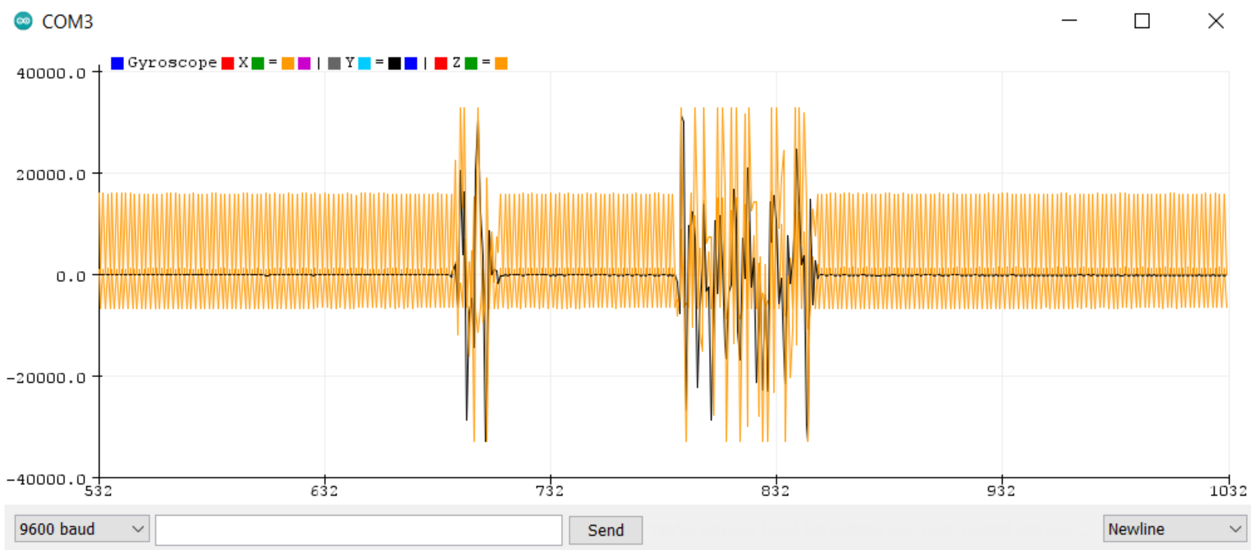

Figure 8 Accelerometer GY-521 result with x difference

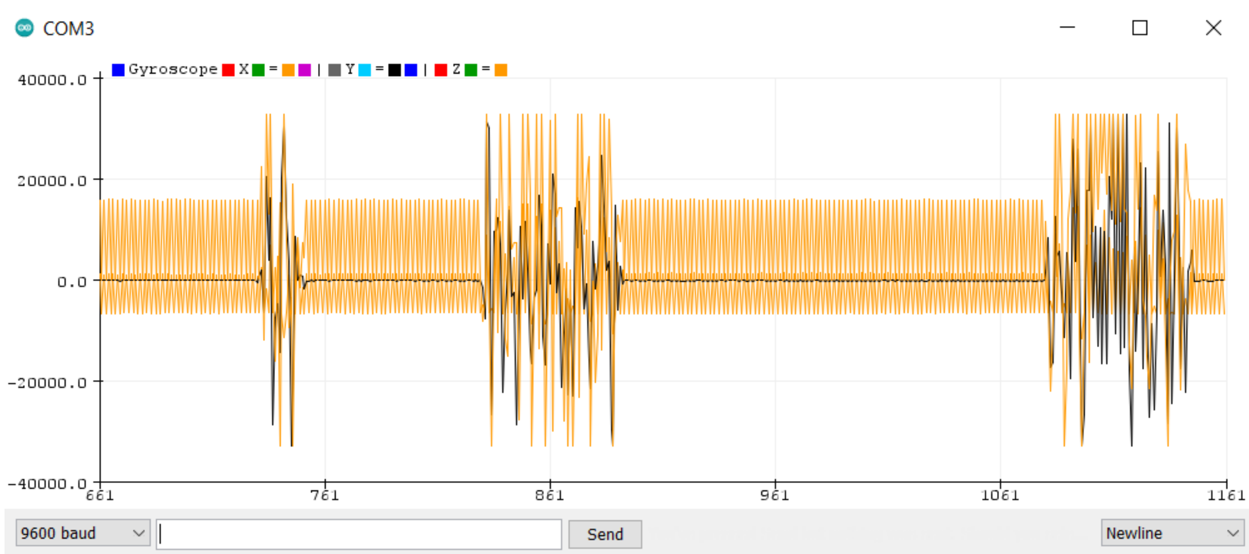

Figure 10 Accelerometer GY-521 result with rotational movement

Sensor doesn't get difference for rotational or one direction shook.

CALCULATIONS

For this lab, one should've understood the concept of gain and op-amp negative feedback mechanism beforehand and solved some problems on it to understand and malfunctioning or issues related with part 1. Also one should read the libraries in Arduino beforehand for the technical issues in the display or so and that may even lead to thermocouple resistance calculation since there might be some issues with it from wrong power connections and so if not set well.

DISCUSSION & COMMENTS

In this lab, we mostly learned about the display, gain, more Arduino, and how to visualize our data in different ways in Arduino. I believe this experiment carried the lab sessions with combined exp 6 to high levels. With two together we can make a user interface and make them see it in a display these basics can help one to solve their issues easily, if they want to build simple systems from Arduino and its components. What I understand after this lab which is the end lab for me, we should learn more coding in an object-oriented way, and for myself, since I know Java and C# I will try to find implementations that I can do with them to the Arduino systems for next semesters. If there exist connections from C# or Java, I can easily make a database and real-time data acquisition system with a great user interface.

REFERENCES

GY-521 MPU6050 3-Axis Acceleration Gyroscope 6DOF Module [GY-521] - US $1.40 : HAOYU Electronics : Make Engineers Job Easier (hotmcu.com)