Principles of Measurement &Instrumentation I Laboratory
PHYS417

Experiment 4-Communication Protocols: I2C & SPI Compatible Sensors

Author Batuhan Dereli

1851690

Experiment Date:30.10.2019

Submission Date:6.11.2019

Contents

Objectives

Briefly, this experiment was very important because SPI communication and also signal amplifier basics are very important. First mention for the signal gain amplifier is the necessary for the our analog transducers also called sensors have week output signal voltage ,generally , we are prefer before read with microcontroller using analog the digital hardware but before using this hardware we need to availed and readable signal from transducer , how it be possible because of this amplification we can prefer many analog circuit designs in the laboratory we preferred basic op-amp amplifier circuit end of the result when we apply to week 1 volt signal output was 7-8 volts actually there is no significant number because of the circuit and other components have resistance voltage little bit drop after that measurement .

Another point of this experiment using SPI communication with addressing signals actually not detailed used this protocol but that's mean SCA and SDA pins on the Arduino you can addressed many devices and used somehow SPI library protocol rules and at the same time you can drive two LCD screen.

Also, first three experiment we used to serial monitor and measurements generally plot or write on the txt or monitored at the IDE Arduino or PYTHON software. That experiment independently we used to LCD screen so sensor temperature or any value of the DATA measurement get meaningful for the third moderator users or who is making experiment with low delay second experienced to observe that.

Last point for this experiment , generally we were write the software without using libraries , but in this experiment with microcontroller how to drive the other devices properly we learned as a results in libraries depends on the pin out and pin in diagrams also inside chip integrated board differently and properly make some coding ways helps to end of the last users or without write thousands rows nor you can use any device with availed software's .

Introduction

A **programmable-gain amplifier** (**PGA**) is an electronic amplifier (typically an operational amplifier) whose gain can be controlled by external digital or analog signals.

The gain can be set from less than 1 V/V to over 100 V/V. Examples for the external digital signals can be SPI, I²C while the latest PGAs can also be programmed for offset voltage trimming, as well as active output filters. Popular applications for these products are motor control, signal and sensor conditioning. [1]

The PGA implements an opamp-based, non-inverting amplifier with user-programmable gain. This amplifier has high input impedance, wide bandwidth and selectable input voltage reference. It is derived from the switched capacitor/continuous time (SC/CT) block.[2]

In the analog systems controls are generally with hands but in the digital system generally exact values and specific values are possible the write and control the circuits.

SPI communication is the another case in this experiment ,Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.[3]

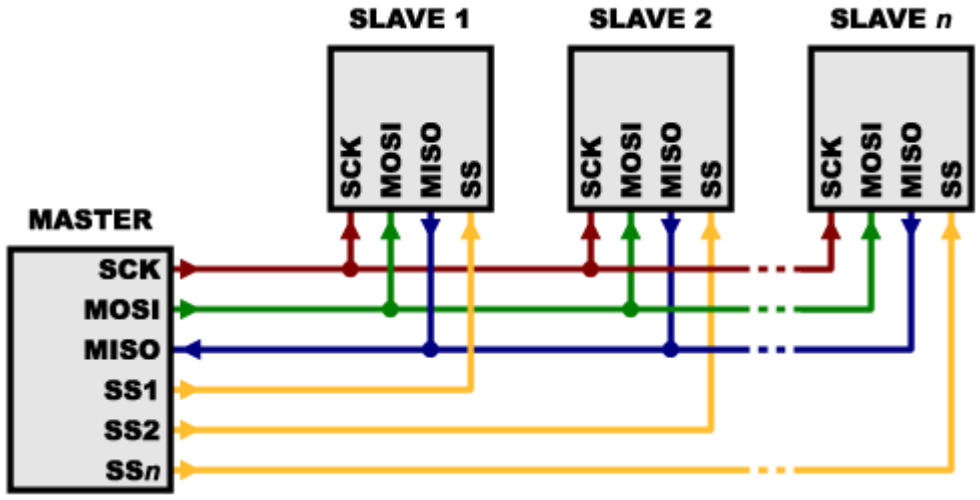


Figure 1 MASTER -SLAVE RELATIONSHIP

With SPI library make possible the connect many slaves or give the example LCD or sensor on the same cable lines. But we have to change address. Also 5 years ago or before old times in the marketing no I2C connection hardware so much, when we buy the LCD 16 pins we had to connect to Arduino and inside the Arduino pins immediately finished but I2C connection thanks to we are just using two wires also 5v and ground connection solve this problem.

Equipment

- LM741 Op-Amp
- Function Generator
- Oscilloscope
- Arduino UNO
- Mechanical Potentiometer of 100 KΩ
- X9C104 digital potentiometer module
- MAX6675 temperature sensor module
- LSM303D accelerometer module
- Lighter
- Liquid Crystal Display LCD with I2C Serial Interface Board Module

Procedure

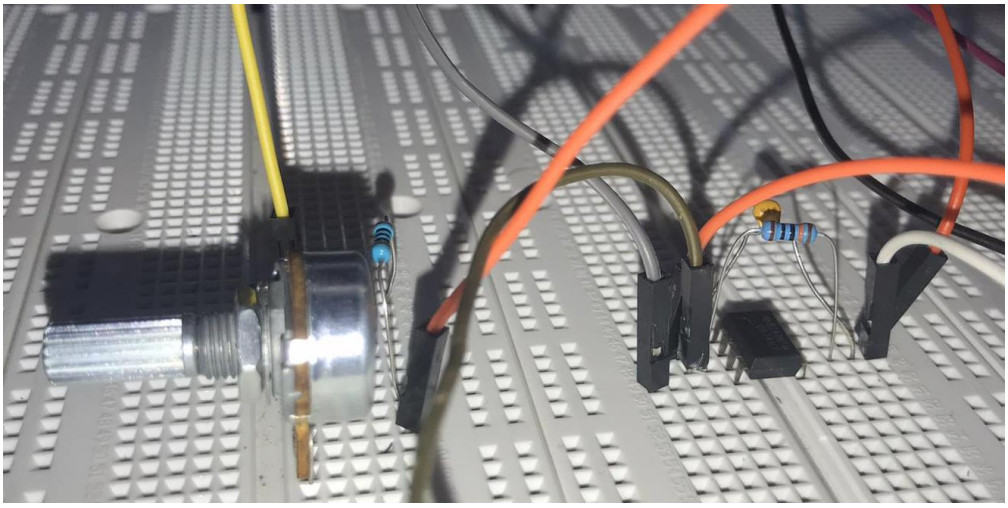


Figure 2 Gain Amplifier

First part of the experiment try to increase the gain with old style because of that we used op-amp circuit and drive this circuit with potentiometer.

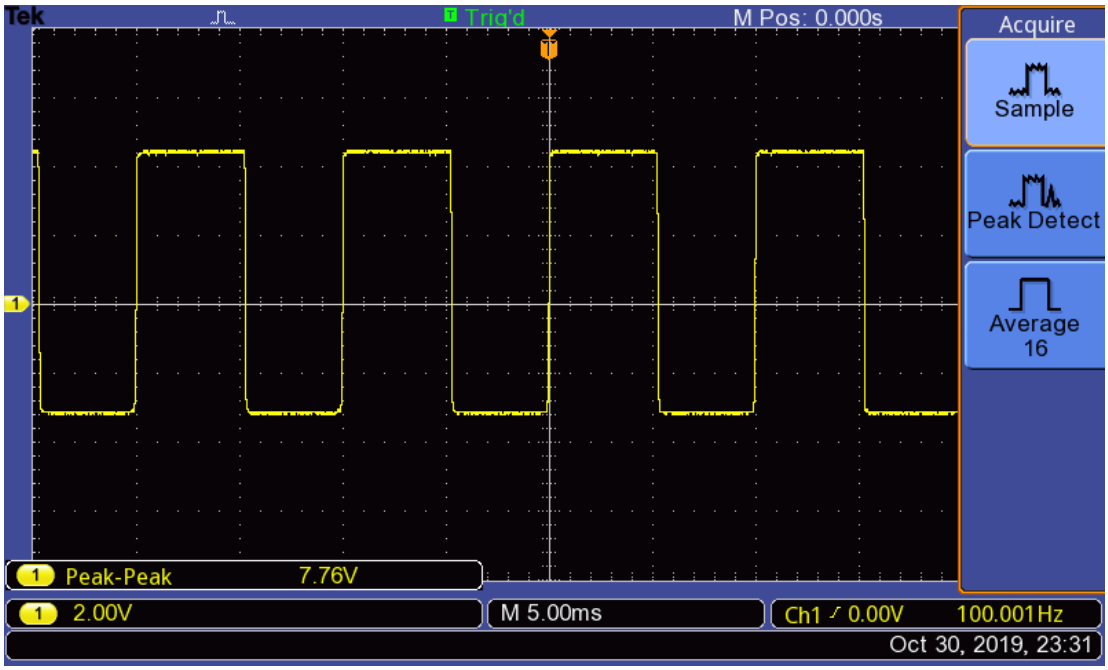


Figure 3WITH POT - MAX TURN AFTER 1VOLT - WE GAIN 7.76 VOLT

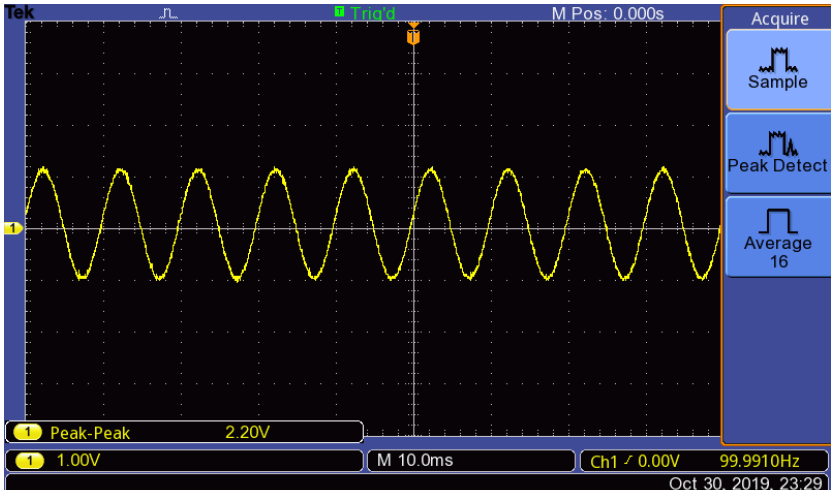


Figure 4ORJINAL SIGNAL

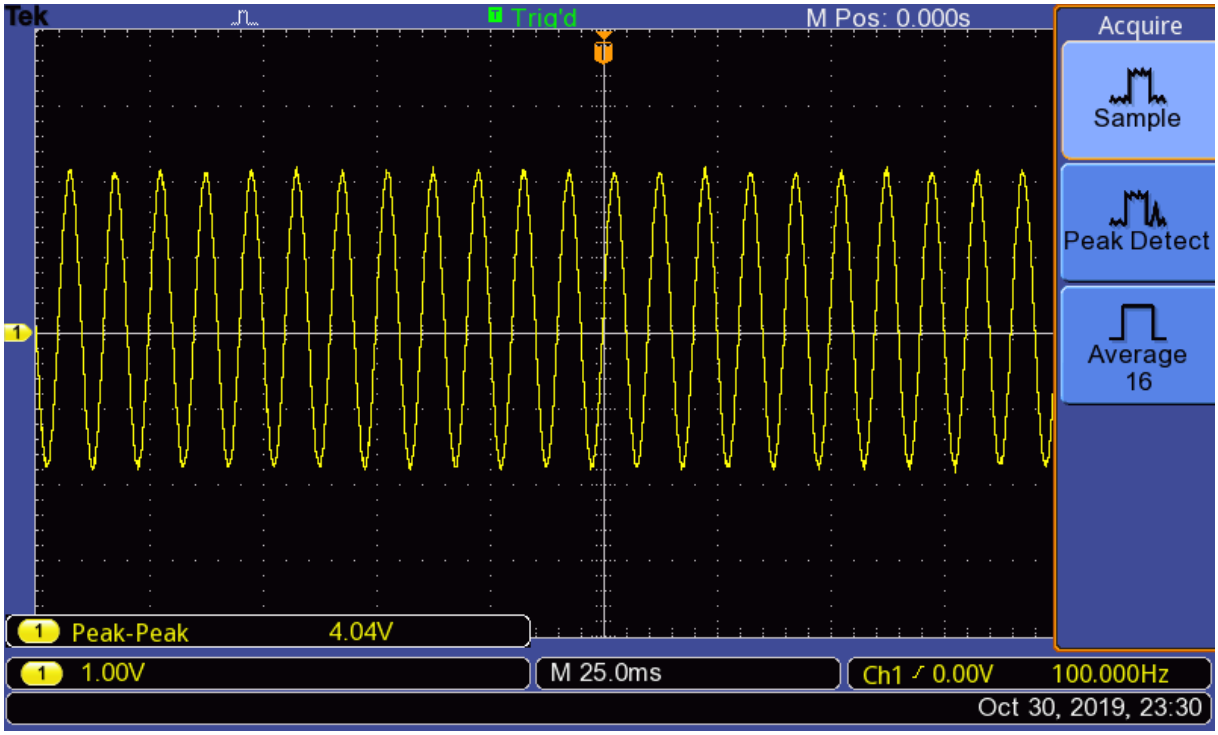
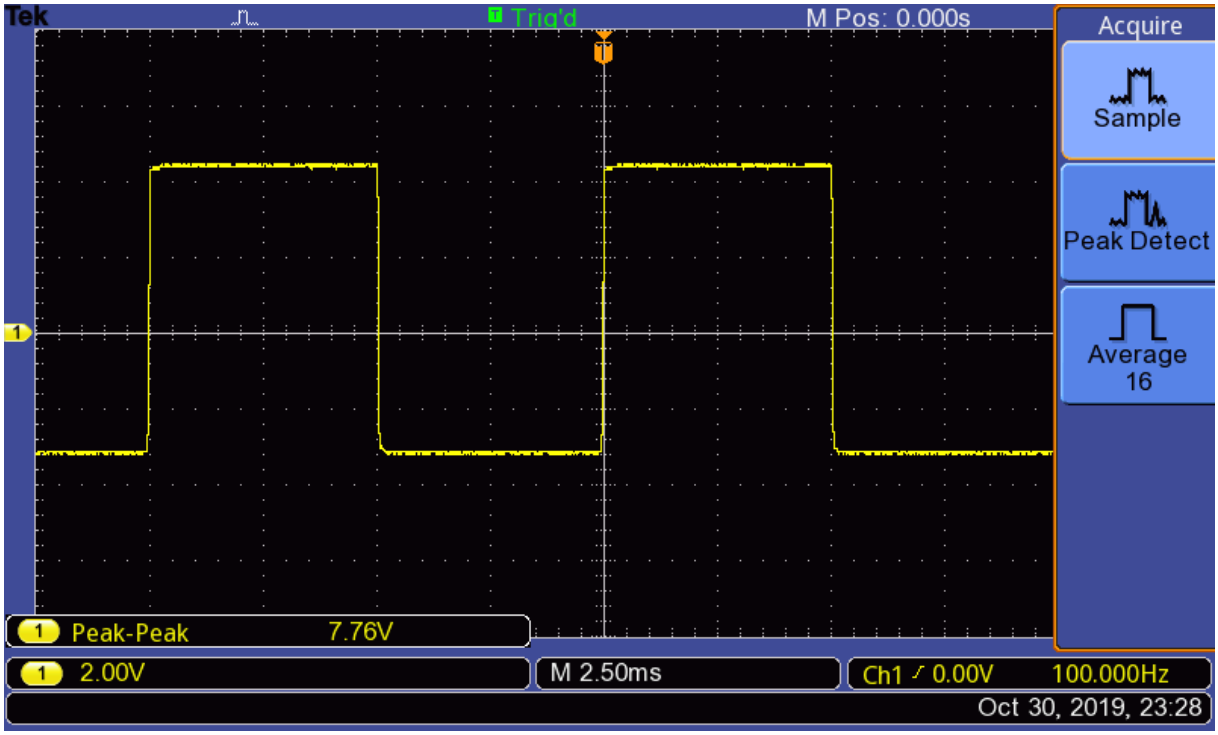


Figure 5HALF TURN



Without capacitor frequency ,

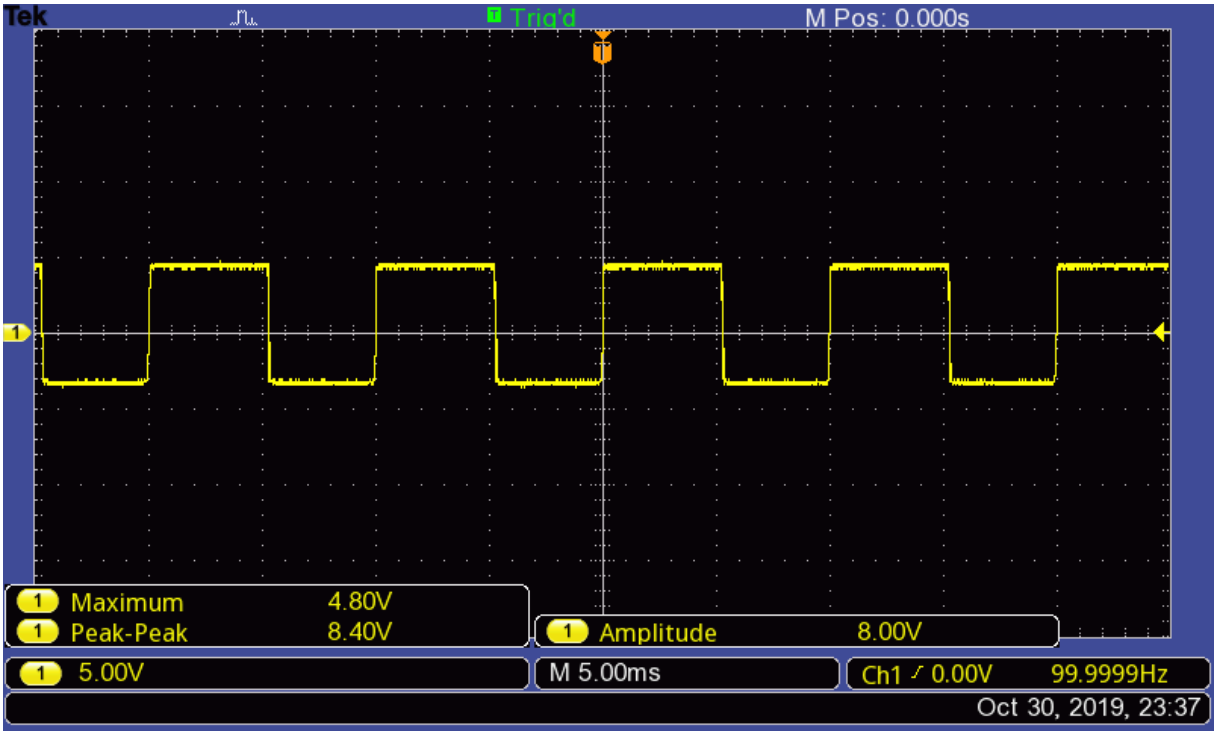


Figure 6Without capacitor

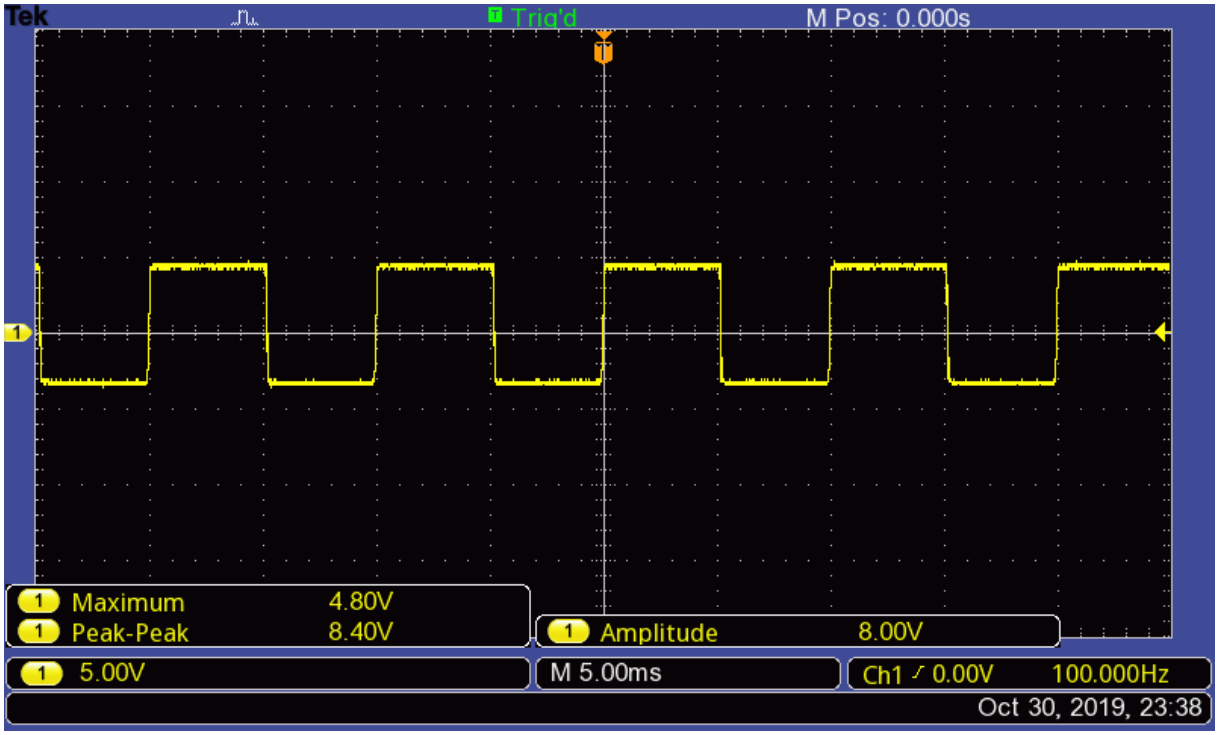
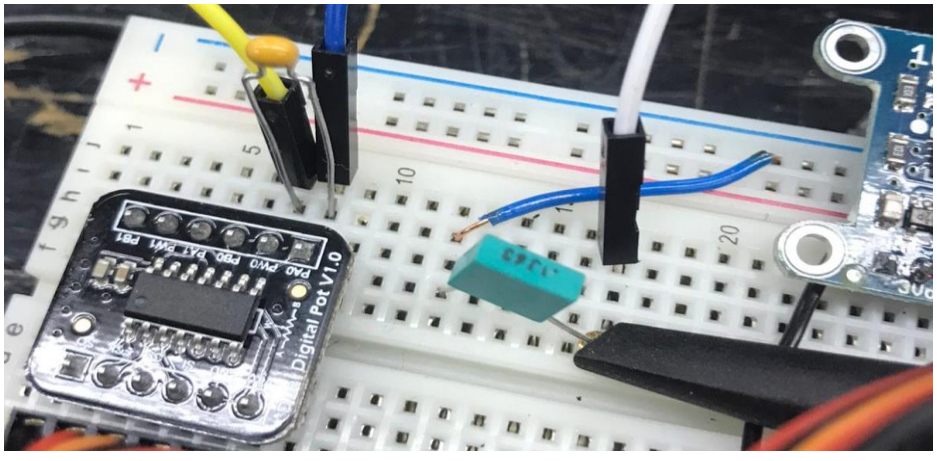


Figure 7 with capacitor

Input signal frequency did not effect to my output signals . But resistor and capacitor values affected my output signal and also Input peak to peak voltage changes affected my output voltages of course opamp 741 limited maximum I observed 17 volts .

BONUS



With this capacitor solved the more different shapes graphs during the experiment (100nf) Pls use the this capacitor end of the circuit output between ground. Another trick used 33 pf capacitor PA0 to PW0 Between.

Figure 8CAPACITOR TRICKS

RESULT

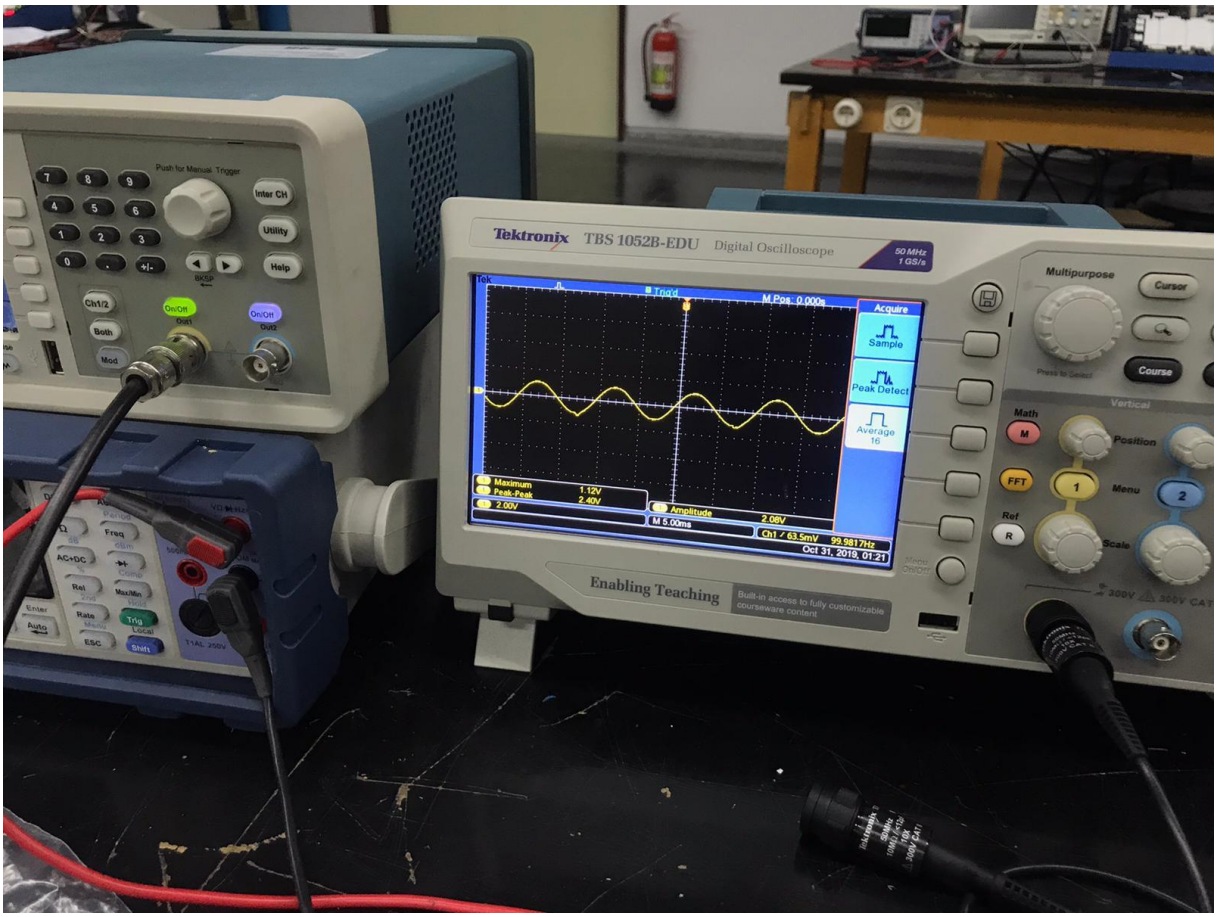


Figure 9 CAPACITORS AFTER USED FIX THE SIGNAL

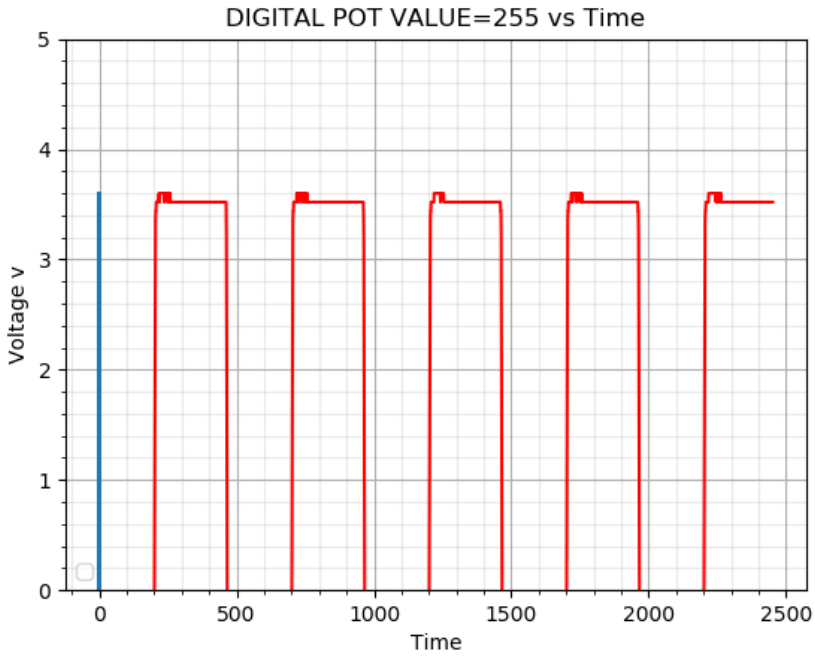


Figure 11Digital pot value 255

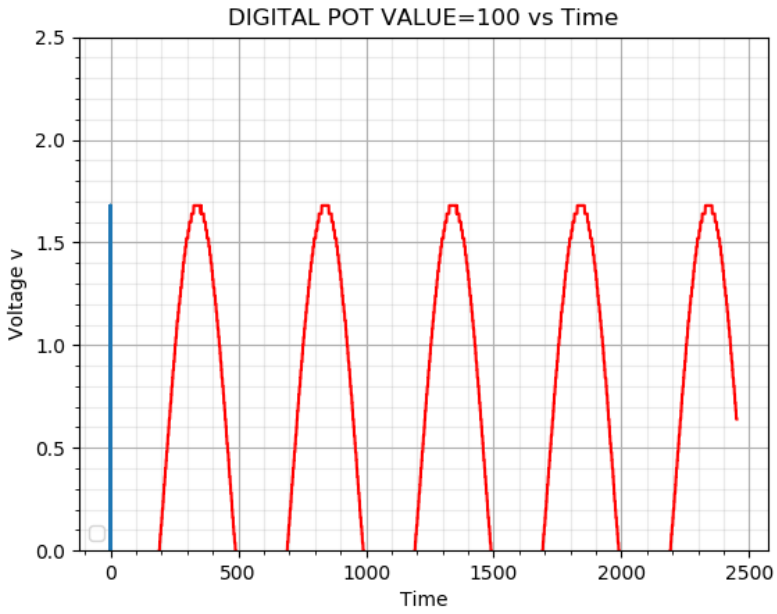


Figure 10 Digiral pot value

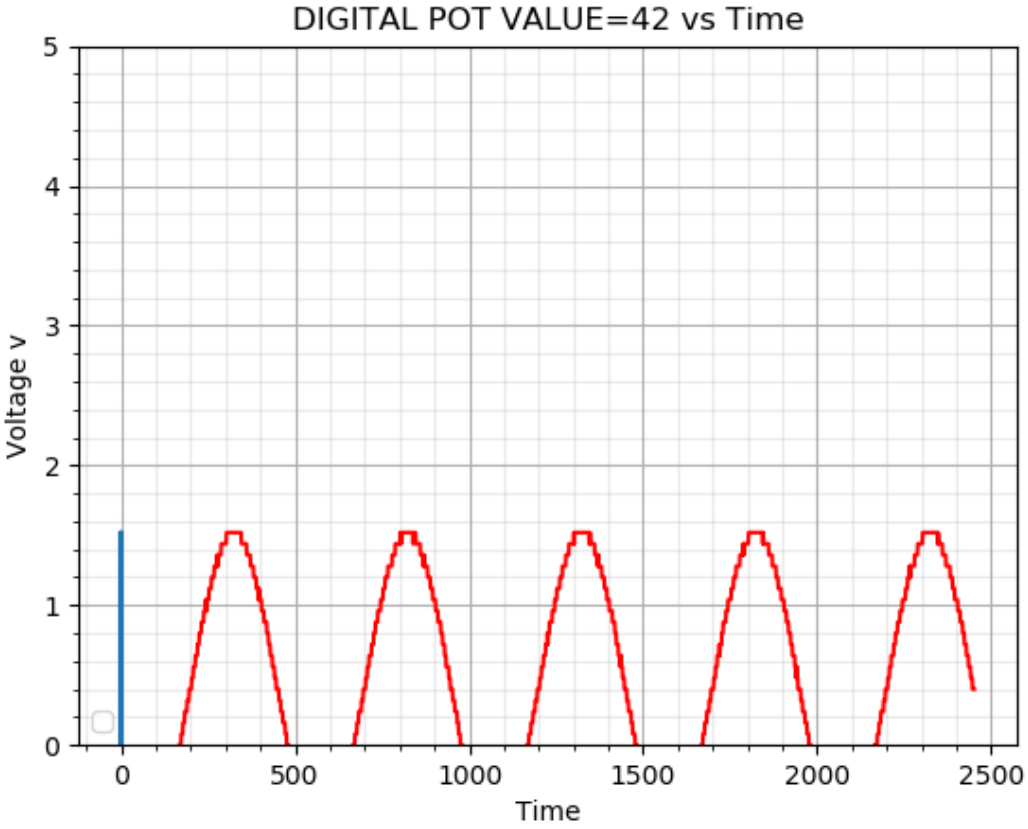


Figure 12Digital Pot values

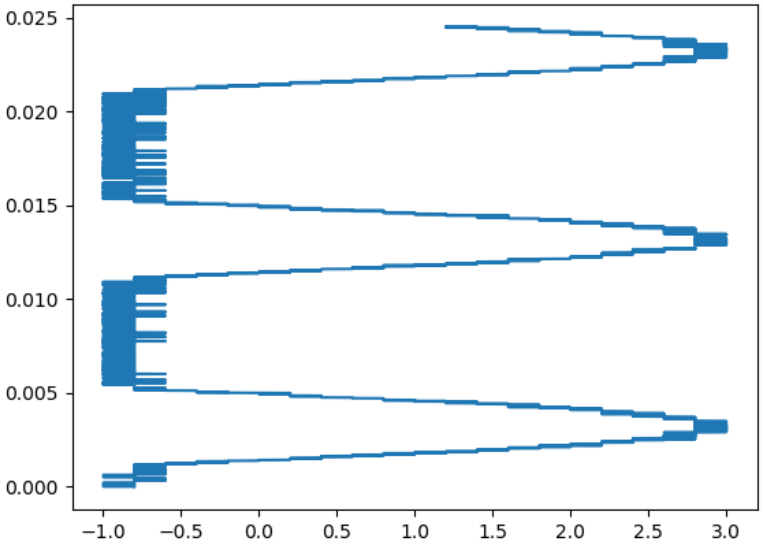


Figure 13Digital Pot values capctured with python

Thermocuple Sensor

Figure 14Thermocouple part

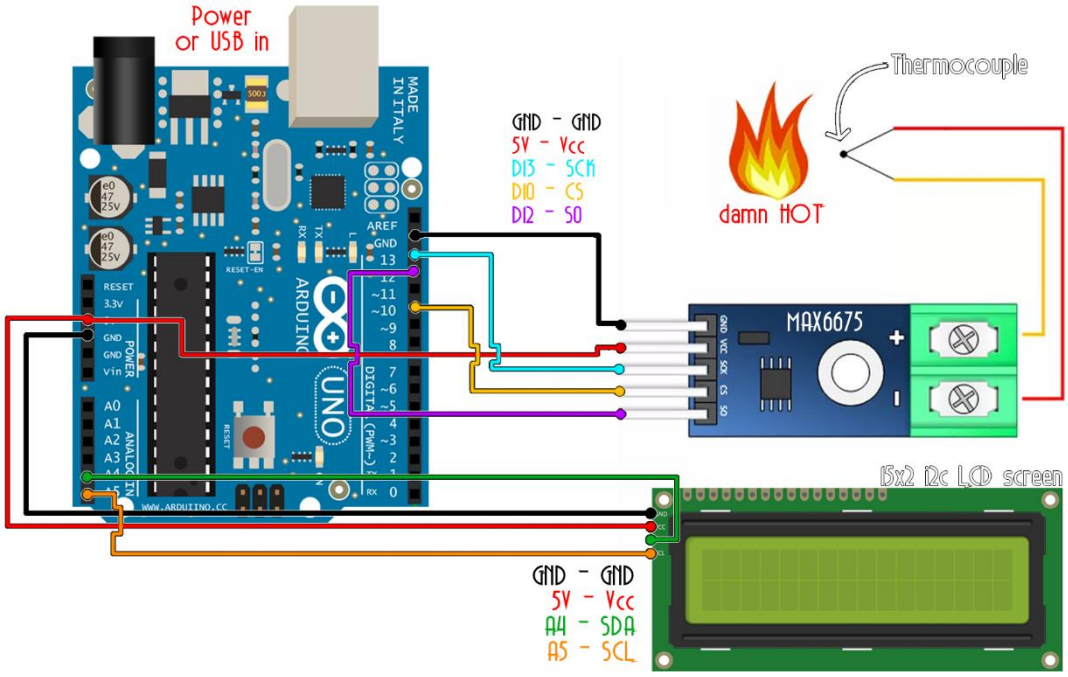


Figure 15 thermocouple circuit

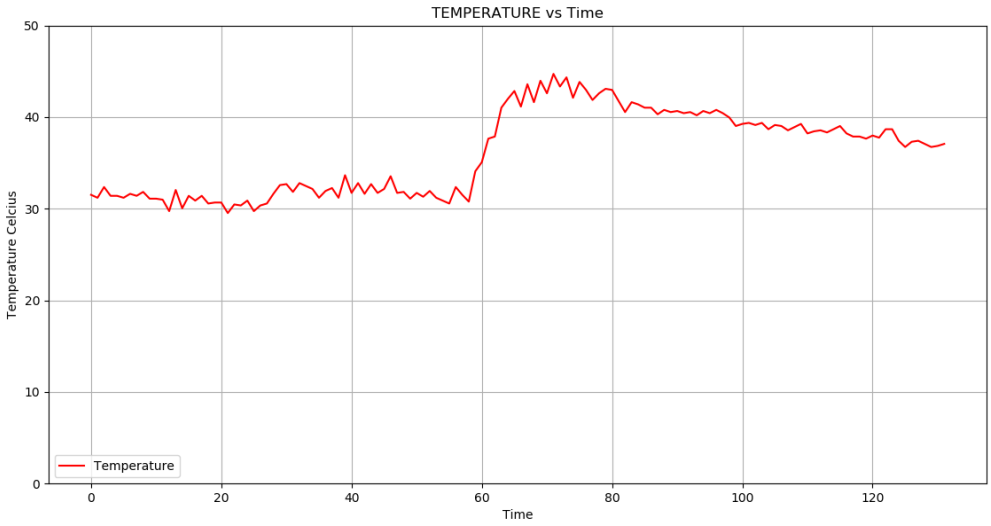


Figure 16Temperature vs Time Real Time Plot

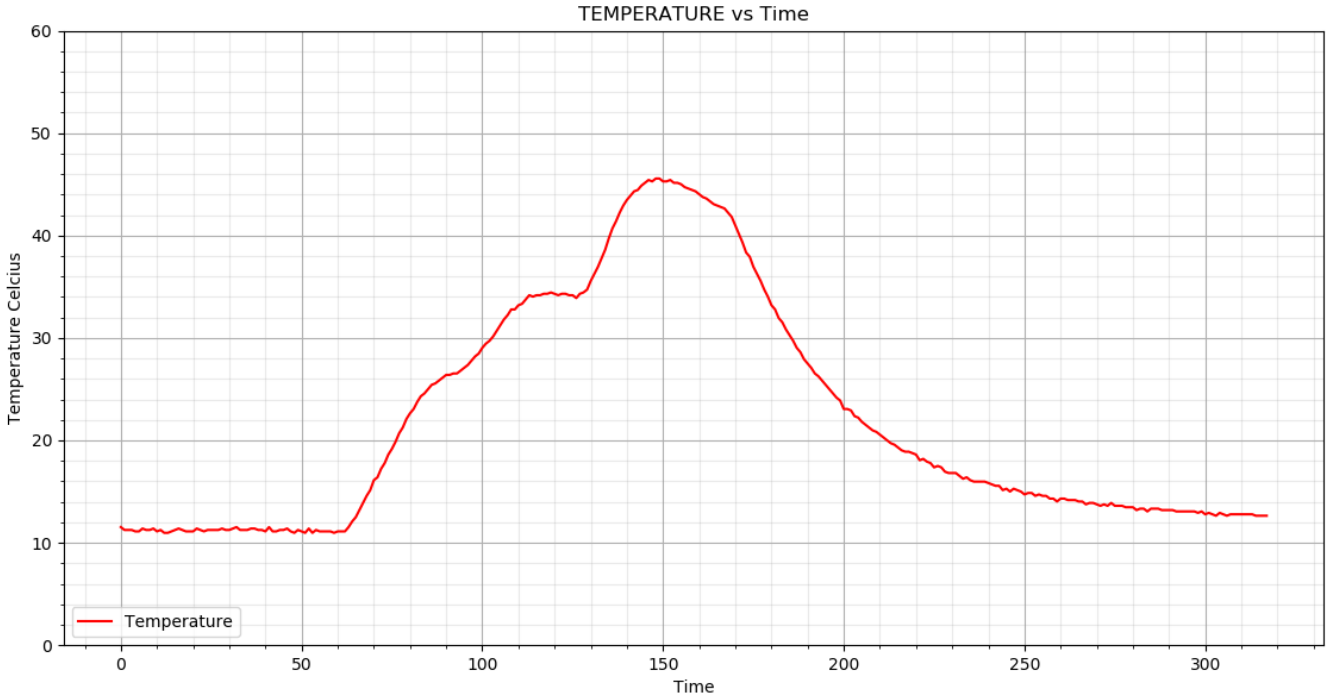


Figure 17K type thermocouple Data

3D Digital Accelerometer

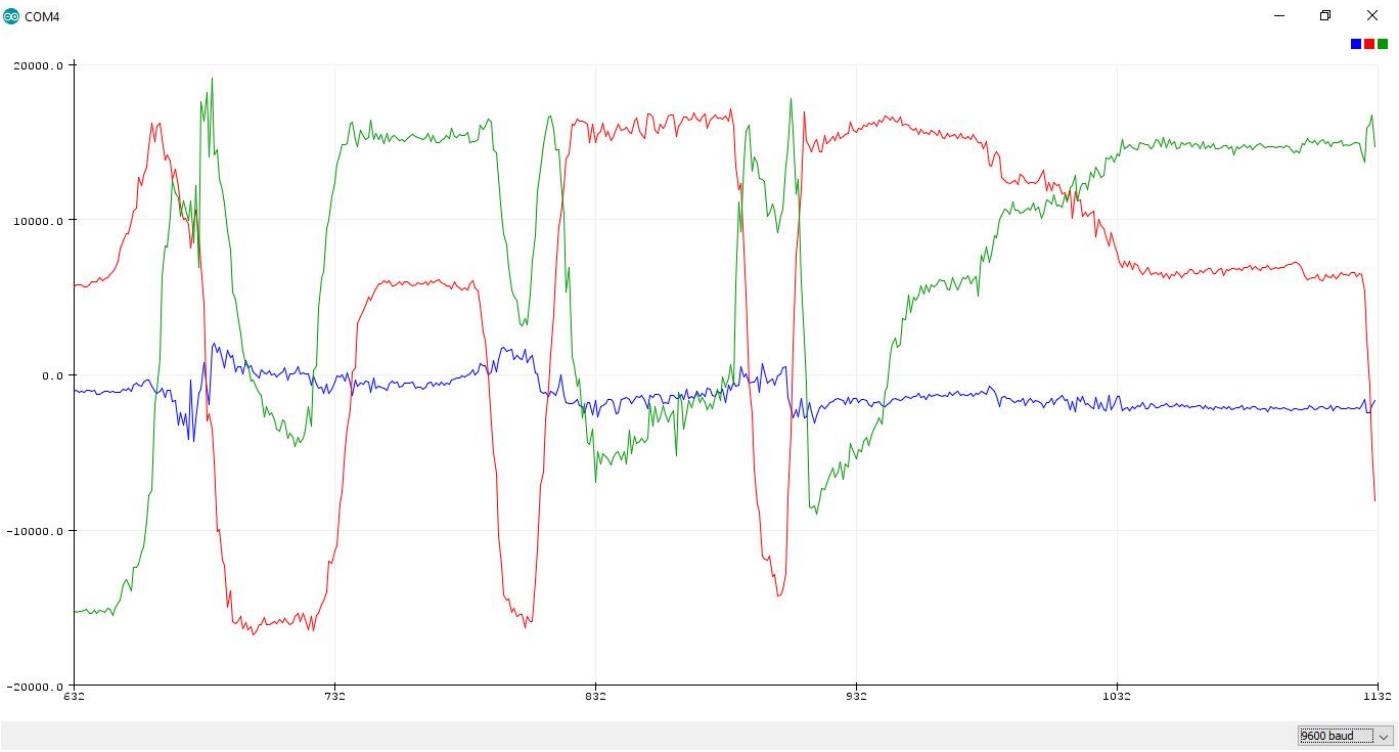


Figure 18 accelerometer data

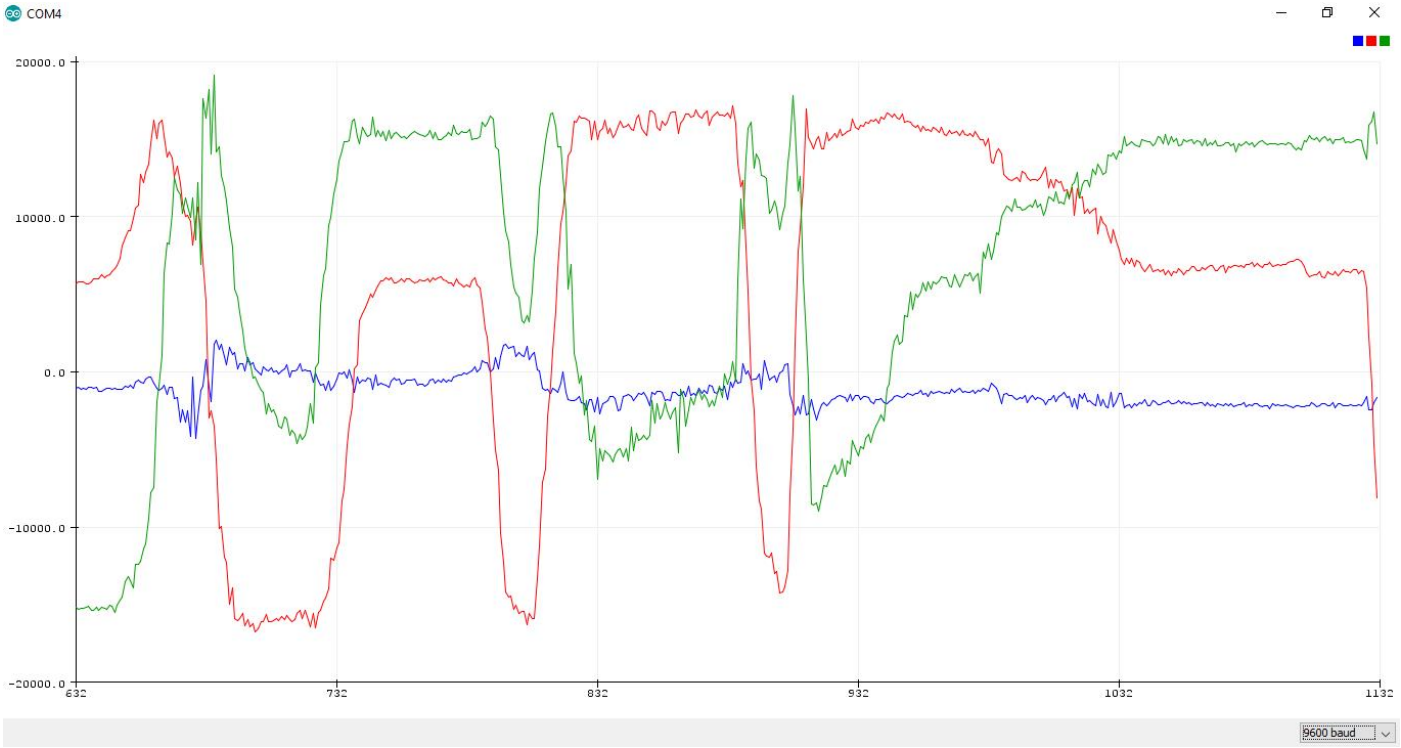


Figure 20 x axis

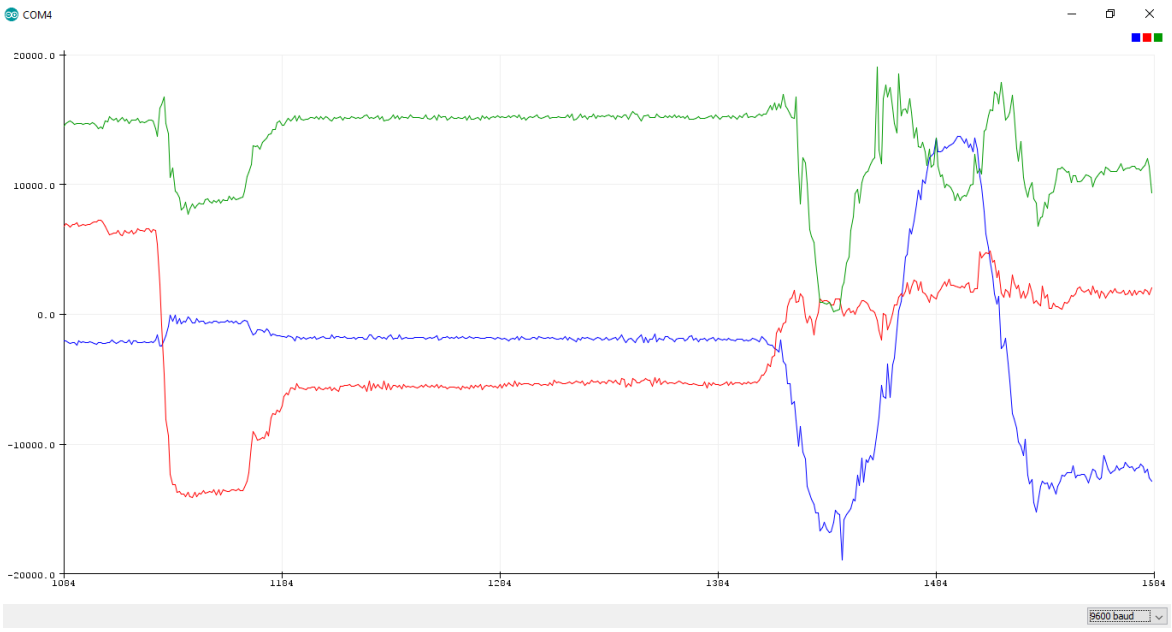


Figure 19 Y axis

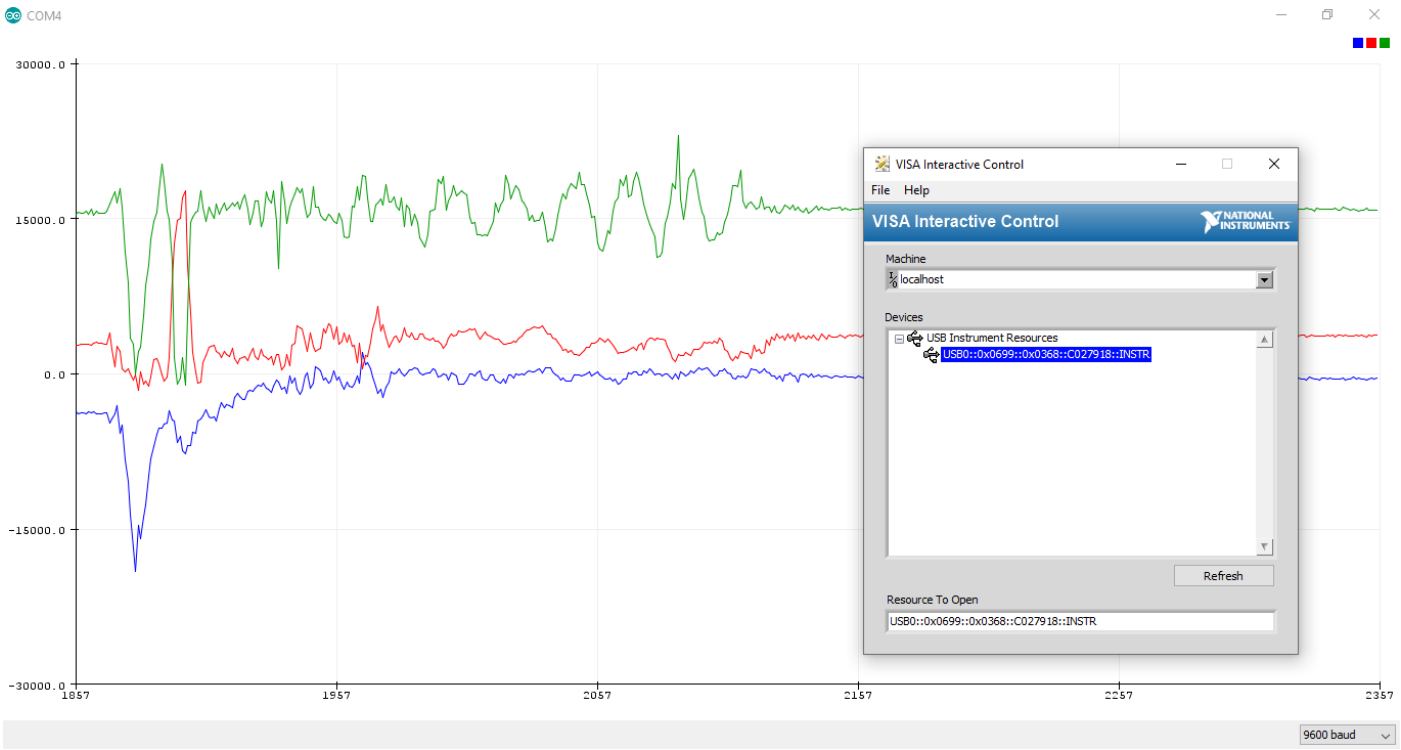


Figure 21 Z axis

Results and Discuss

This experiment remind us to how important libraries for using the sensors or other devices and also I observe that very small amount k type thermocouple values on the oscilloscope and I try to use gain amplifier and observe them .This experiment rereport I wrote everything more clearly if you can read both of them it will more under stable .

Codes

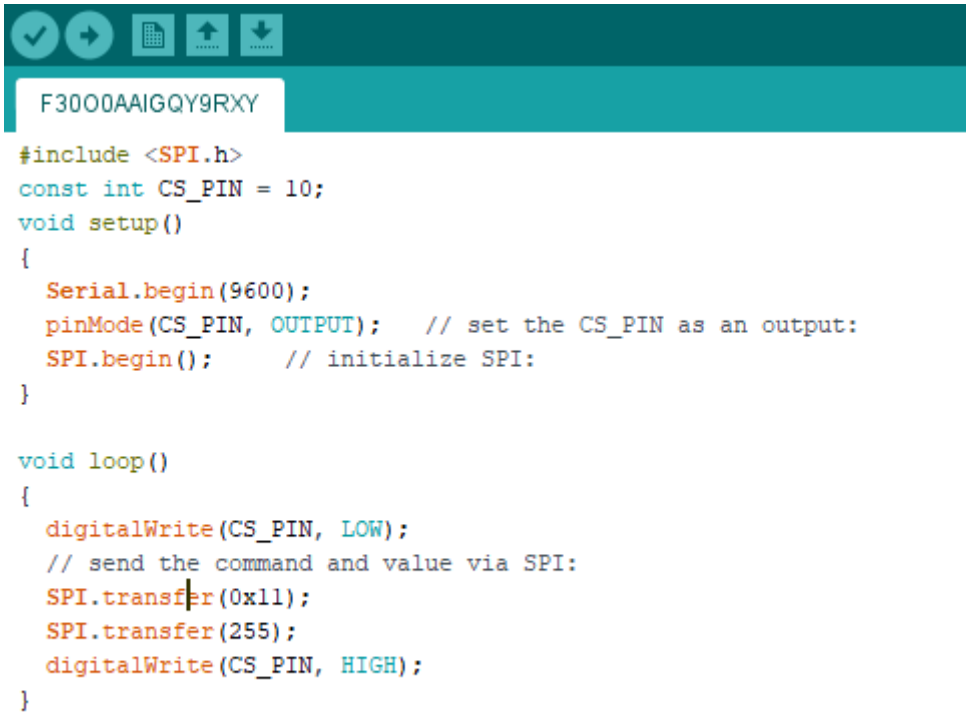


Figure 22 DIGITAL SET VALUE CODE

ACCO

```
#include <Wire.h>
#include <LSM303.h>

LSM303 compass;

char report[80];

void setup()
{
  Serial.begin(9600);
  Wire.begin();
  compass.init();
  compass.enableDefault();
}

void loop()
{
  compass.read();

  snprintf(report, sizeof(report), "A: %6d %6d %6d    M: %6d %6d %6d",
    compass.a.x, compass.a.y, compass.a.z,
    compass.m.x, compass.m.y, compass.m.z);
  Serial.println(report);

  delay(500);
}
```

Figure 23 DIGITAL ACCO CODE

thermocouple_arduino_lab_exp_4a

```
/*    Max6675 Module  ==>   Arduino
 *    CS              ==>     D10
 *    SO              ==>     D12
 *    SCK             ==>     D13
 *    Vcc             ==>     Vcc (5v)
 *    Gnd             ==>     Gnd      */

//LCD config
#include <Wire.h>
#include <LiquidCrystal_I2C.h>    //If you don't have the LiquidCrystal_I2C library, download it and install it
LiquidCrystal_I2C lcd(0x27,20,4);  //sometimes the adress is not 0x3f. Change to 0x27 if it dosn't work.

/*    i2c LCD Module  ==>   Arduino
 *    SCL             ==>     A5
 *    SDA             ==>     A4
 *    Vcc             ==>     Vcc (5v)
 *    Gnd             ==>     Gnd      */

#include <SPI.h>

#define MAX6675_CS   10
#define MAX6675_SO   12
#define MAX6675_SCK  13

int temperature_read;

char userInput;
void setup() {
  lcd.init();
```

Figure 24 K TYPE THERMOCOUPLE-1

thermocouple_arduino_lab_exp_4a

```
void setup() {
  lcd.init();
  lcd.backlight();
 Serial.begin(9600);
}

void loop() {
  float temperature_read = readThermocouple();
  lcd.setCursor(0,0);
  lcd.print("TEMPERATURE");
  lcd.setCursor(7,1);
  lcd.print(temperature_read*0.55555,1);
  Serial.println(temperature_read*0.55555);


  delay(300);
}




double readThermocouple() {

  uint16_t v;
  pinMode(MAX6675_CS, OUTPUT);
  pinMode(MAX6675_SO, INPUT);
  pinMode(MAX6675_SCK, OUTPUT);

  digitalWrite(MAX6675 CS, LOW);
```

Figure 25 THERMOCOUPLE 2

thermocouple_arduino_lab_exp_4a

```
 pinMode(MAX6675_SO, INPUT);
 pinMode(MAX6675_SCK, OUTPUT);

 digitalWrite(MAX6675_CS, LOW);
 delay(1);

 // Read in 16 bits,
 //  15    = 0 always
 //  14..2 = 0.25 degree counts MSB First
 //  2     = 1 if thermocouple is open circuit
 //  1..0  = uninteresting status

 v = shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
 v <<= 8;
 v |= shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);

 digitalWrite(MAX6675_CS, HIGH);
 if (v & 0x4)
 {
   // Bit 2 indicates if the thermocouple is disconnected
   return NAN;
 }

 // The lower three bits (0,1,2) are discarded status bits
 v >>= 3;

 // The remaining bits are the number of 0.25 degree (C) counts
 return v*0.25;
}
```

Figure 26 THERMOCOUPLE -3