

**Computer Systems Engineering Technology**

**CST 120 – Embedded C Programming**

|  |  |
| --- | --- |
| Lab 04\_Voltage Divider; Thermometer | Name\_\_\_\_\_\_Chris Thomas\_\_\_\_\_\_\_\_\_\_\_\_ |
| Spring 2020 | Due Date: 11:59PM Friday 5/6/2022 |
| Instructor: George Drouant | |
| Possible Points: 100 | |

# Instructions

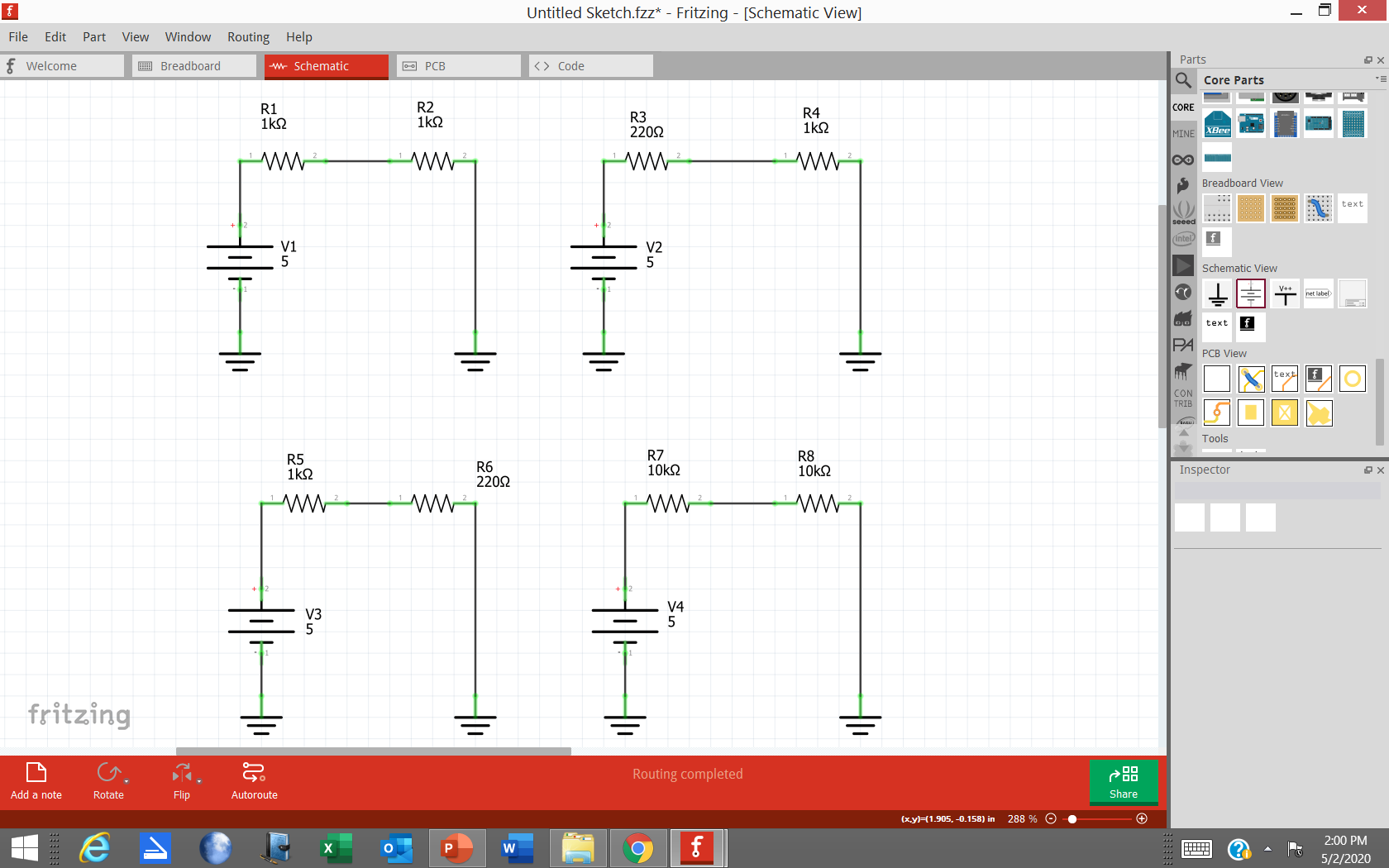
This is a two part lab. The first part will be a tutorial introducing you to the Voltage Divider Circuit that we will be using with a Thermistor to build a Thermometer.

# Part 1 – Voltage Divider

The first step will be to plug in your 5 Volt power supply into the protoboard. Connect two 1K resistor as shown in the schematic and the photo below. Plug the wallwart into the power supply after you have built and check the circuit.

A circuit board

Description automatically generated



Using the Voltage Divider Equation calculate the voltages across R1 (V\_R1) and R2 (V\_R2). Record your answers in the “Calculated” table.

V\_R1 = (R1 / (R1+R2)) \* 5V

V\_R2 = (R2 / (R1+R2)) \* 5V

**Calculated**

|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R1** | **V \_R2** |
| **5V** | **2.5** | **2.5** |

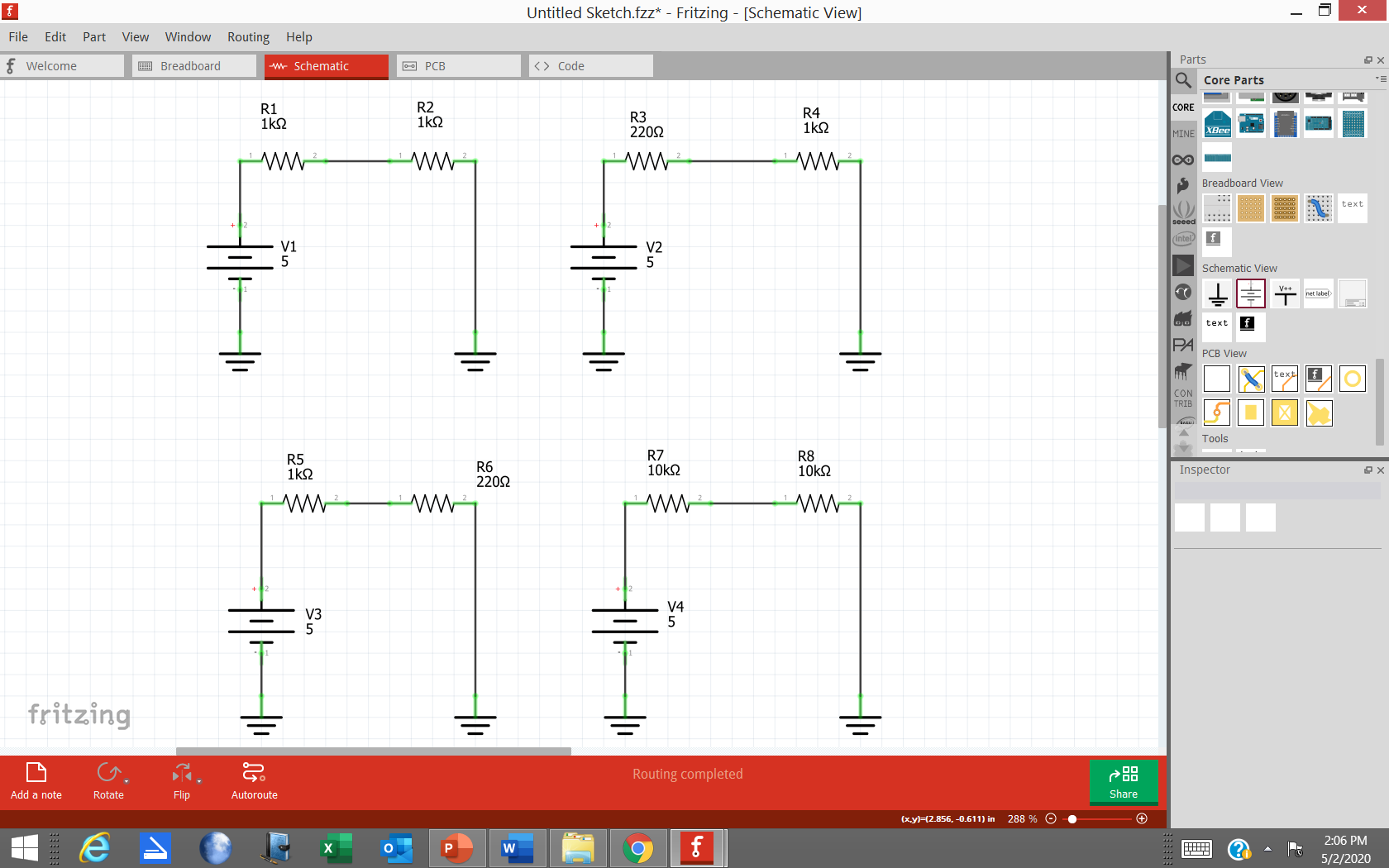
Use the Digital Multimeter to measure the voltages across R1 (V\_R1) and R2 (V\_R2). Record your observations in the “Measured” table.

**Measured**

|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R1** | **V \_R2** |
| **5V** | **2.48V** | **2.48V** |

How do the calculated and measured values compare?

They are within tolerance of each other.



Using the Voltage Divider Equation calculate the voltages across R3 (V\_R3) and R4 (V\_R4). Record your answers in the “Calculated” table.

V\_R3 = (R3 / (R3+R4)) \* 5V

V\_R4 = (R4 / (R3+R4) \* 5V

**Calculated**

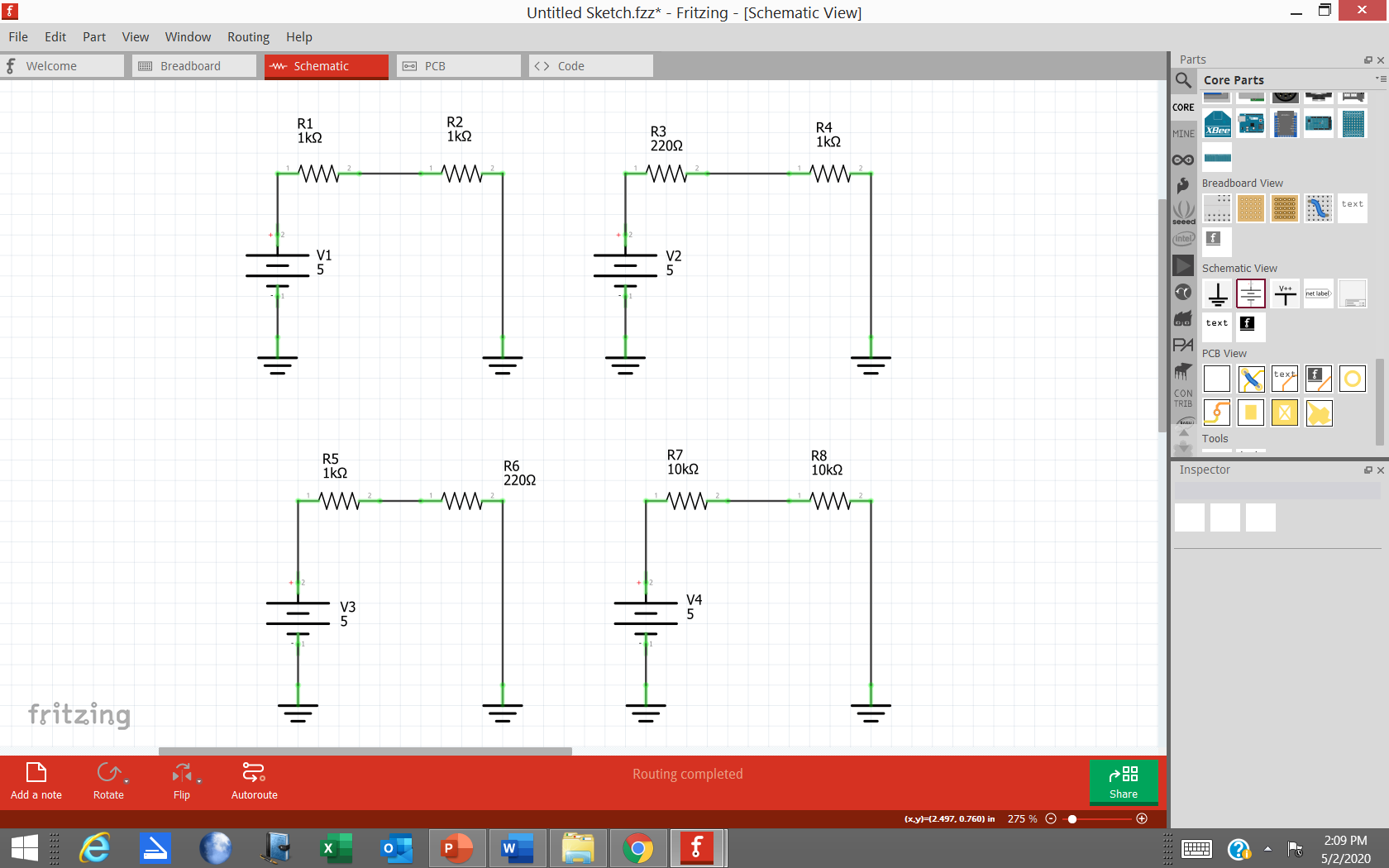
|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R3** | **V \_R4** |
| **5V** | **901mV** | **4.09V** |

**Measured**

|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R3** | **V \_R4** |
| **5V** | **890mV** | **4.06V** |

How do the calculated and measured values compare?

They are withing tolerance of each other



Using the Voltage Divider Equation calculate the voltages across R5 (V\_R5) and R6 (V\_R6). Record your answers in the “Calculated” table.

V\_R5 = (R5 / (R5+R6)) \* 5V

V\_R6 = (R6 / (R5+R6)) \* 5V

**Calculated**

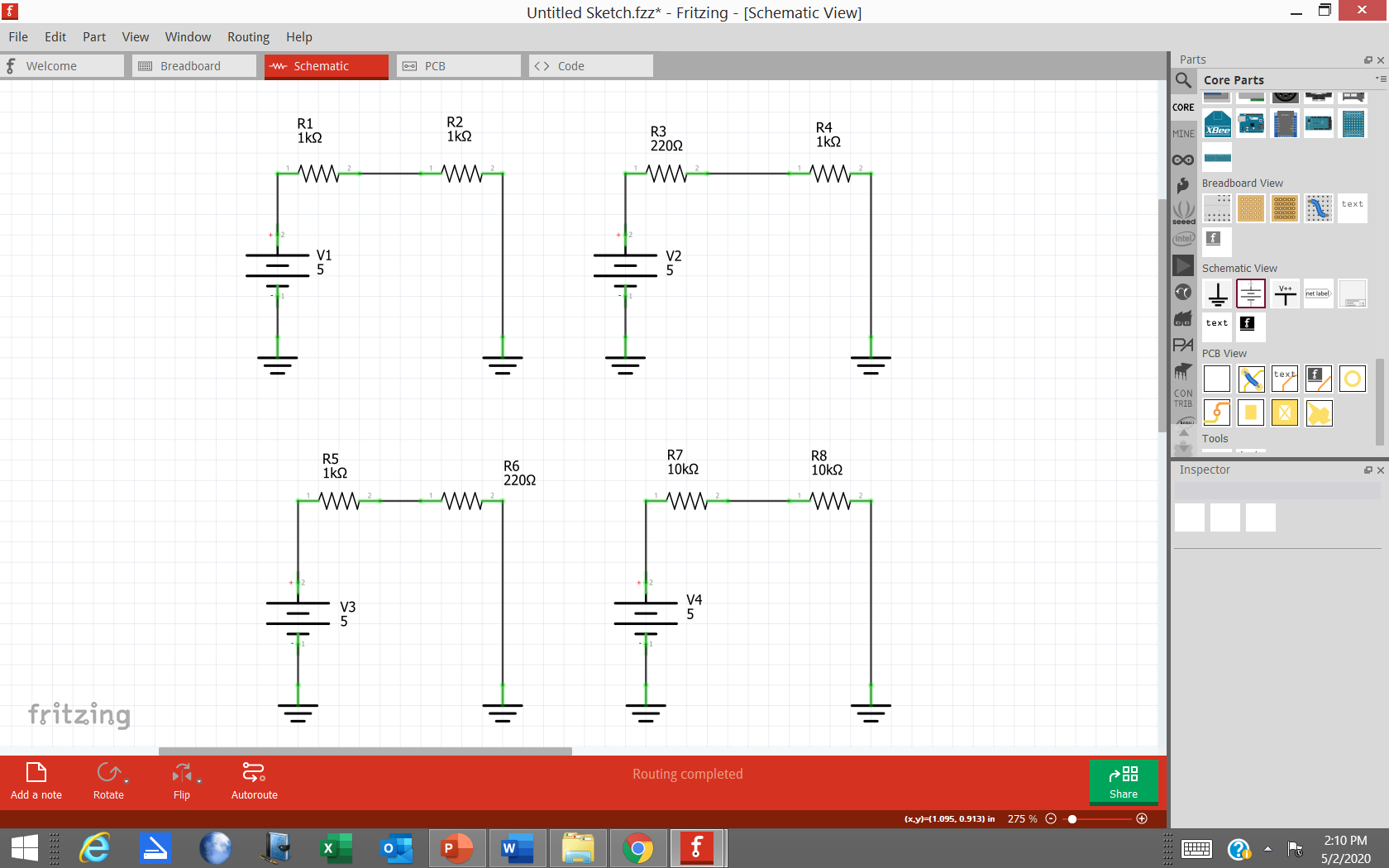
|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R5** | **V \_R6** |
| **5V** | **4.09V** | **901mV** |

**Measured**

|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R5** | **V \_R6** |
| **5V** | **4.06V** | **890mV** |

How do the calculated and measured values compare?

**They are within tolerance of each other**



Using the Voltage Divider Equation calculate the voltages across R7 (V\_R7) and R8 (V\_R8). Record your answers in the “Calculated” table. **Use a 10K Resistor for R7. Use the 10K Thermistor for R8.**

V\_R7 = (R7 / (R7+R8)) \* 5V

V\_R8 = (R8 / (R7+R8)) \* 5V

**Calculated**

|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R7** | **V \_R8** |
| **5V** | **2.5V** | **2.5V** |

**Measured**

|  |  |  |
| --- | --- | --- |
| **Vsource** | **V\_R7** | **V \_R8** |
| **5V** | **2.37V** | **2.58V** |

How do the calculated and measured values compare?

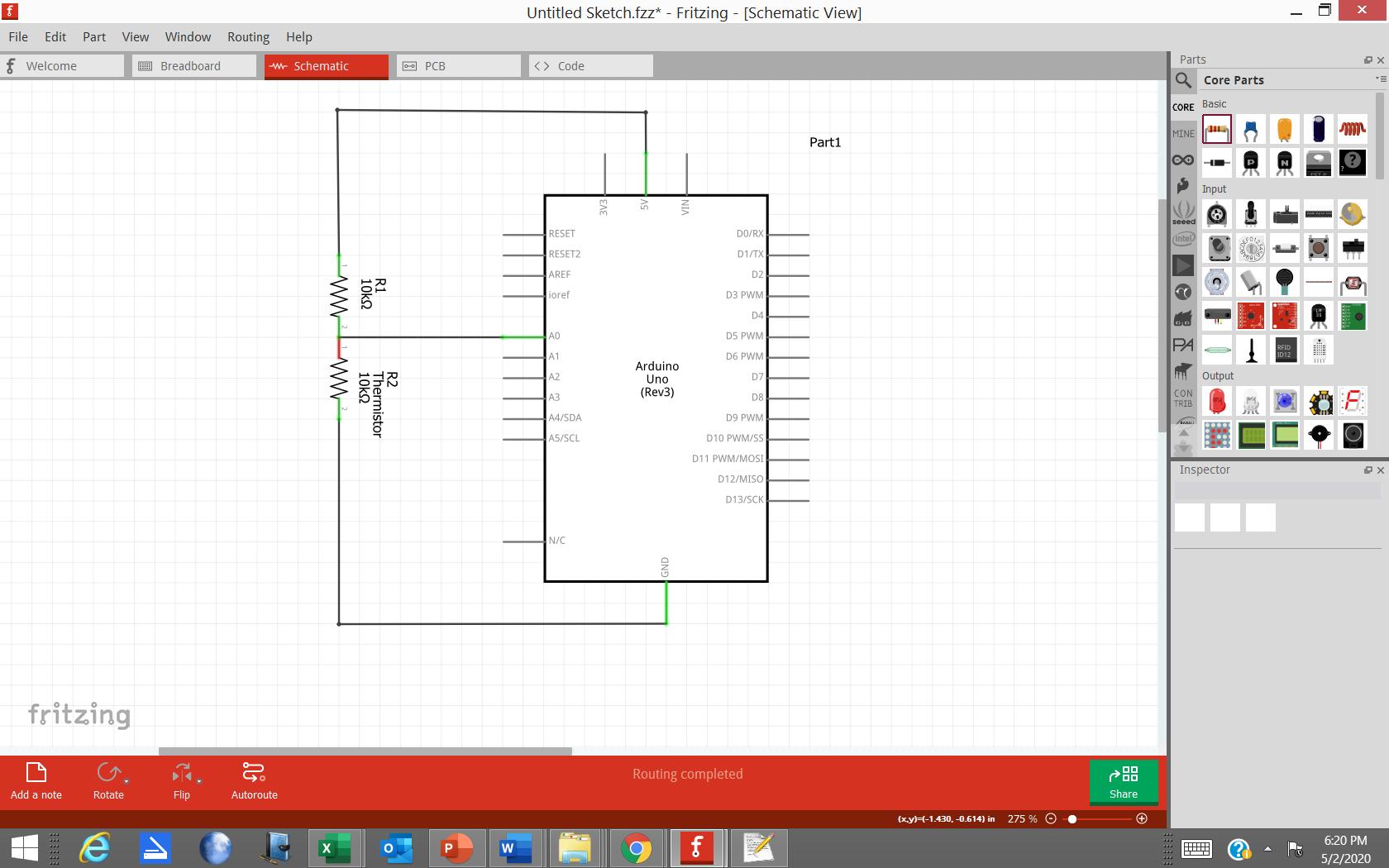
They are within tolerance of each other

Sign Off \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PART TWO - THERMOMETER

Build a thermometer using the information in the following program and graph as well as the 10K Resistor and 10K Thermistor used in the previous experiment.

With the UNO board disconnected from the USB cable build the circuit shown in the schematic and picture below. Build it on your protoboard ***after you have removed the 5V power supply used in the previous experiment***. After you have carefully checked your circuit – making certain that 5V and GND from the UNO are NOT Shorted together - you may reconnect the USB cable.



A circuit board

Description automatically generated

Load the code given below into Atmel Studio. The stdio\_setup.c and stdio\_setup.h files must be added to the project so that the Uart in Atmega328P chip will be able to send temperature data to the host computer for display. Those files can be downloaded from Canvas.

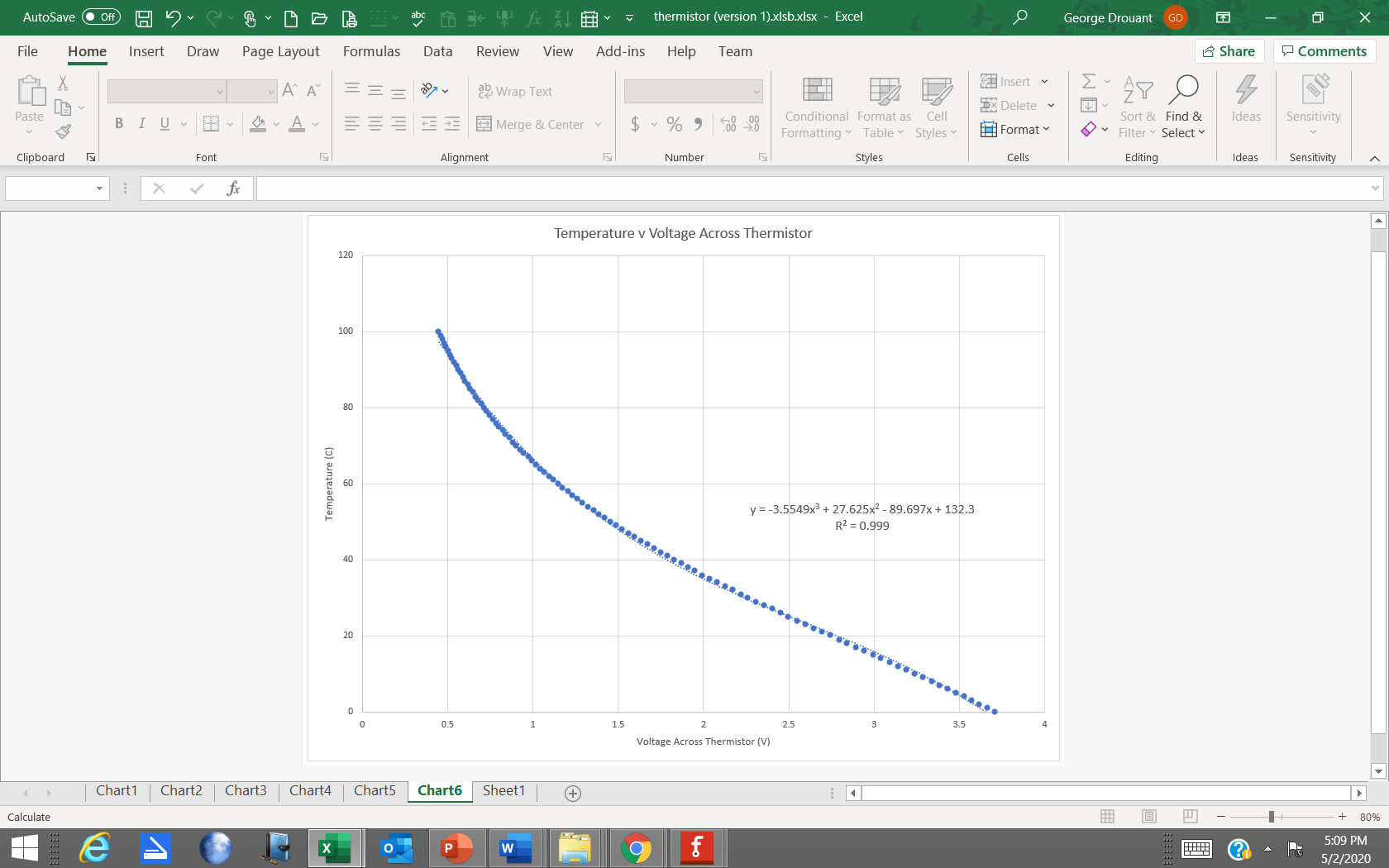
In addition to loading the files above, the following flags must be loaded into the “Other Linker Flags” drop down box in the Miscellaneous section of the AVR/GNU Linker:

-Wl,-u,vfprintf -lprintf\_flt -lscanf\_flt

These flags are required so that avr C will print floating point values. The code to print floating point values takes up “a lot of” space, so avr C usually doesn’t load it. In order to load the flags, right click on the project’s name in Studio’s Solution Explorer box on the right hand side of the computer screen and select “Properties.” Go to the left side of the screen and select “Toolchain.” The “Miscellaneous” option will be located under the “AVR/GNU Linker” selection. Load the flags into the drop down box.

The equation in the graph of *Temperature v Voltage Across Thermistor* will be used to calculate the Temperature given the Voltage Measured Across the Thermistor. Modify the code so that the A/D Voltage is used to calculate the Temperature. If you would like you can use the “pow” function (covered in CST116) to implement the equation shown in the plot.

Don’t forget to #include <math.h> so the pow function will work.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REGISTERS** | **BITS** |  |  |  |  |
| ADMUX | *1<<REFS0* |  |  |  |  |
| DIDR0 | *1<<ADC0D* |  |  |  |  |
| ADCSRA | *1<<ADEN* | *1<<ADSC* | *1<<ADPS2* | *1<<ADPS1* | *1<<ADPS0* |

Use the following template as an aid in writing your code:

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <math.h> //math.h included for pow function

#include "stdio\_setup.h"

int main(void)

{

unsigned int ADC\_data = 0;

float ADC\_voltage = 0.0;

float Temperature = 0.0;

ADMUX = //AVcc (5V supply) used for Vref; Analog Channel 0 (ADC0) used as input; ADC result //is right justified

DIDR0 = //Disables digital input buffer circuit of the ADC0 pin to reduce power consumption

ADCSRA = //Enable ADC; Start ADC; Prescaler Value = 128; ADC Clock = 125 KHz

UartInit();

while (1)

{

if (!(ADCSRA & (1<< ADSC))); //used if to check ADSC in register ADCSRA; ADSC will be 1 //as long as conversion in progress

{

ADC\_data = ADC; //reads the "number" produced by the ADC; reads all 10 bits right justified

ADC\_voltage = //calculate the Analog Voltage (float) read by the ADC

Temperature = //calculate the Temperature (float) from the Analog Voltage and the equation //shown on graph in lab handout

printf("ADC\_Output\_Code = %4d Hex Value = 0x%03x Voltage = %03f Temperature = %04f\r\n",ADC\_data, ADC\_data, ADC\_voltage, Temperature);

ADCSRA |= (1 << ADSC);//start another ADC conversion

\_delay\_ms(5000);

}

}

}

Run and troubleshoot the code if necessary. When the system is working record:

Voltage Measured across thermistor with voltmeter = \_\_\_\_\_\_\_\_\_\_\_

ADC\_Output\_Code = \_\_\_\_\_539\_\_\_\_\_\_\_

ADC\_Voltage = \_\_\_\_\_\_2.622\_\_\_\_\_\_

Temperature = \_\_\_\_\_\_22.95\_\_\_\_\_\_\_

Sign Off \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please turn in your Lab Report to the Canvas Lab website – Assignments Section. Send in the Lab Handout with calculations and observations, and your Thermometer Project Code. Don’t zip your files. Send in the lab handout Word file and a .txt containing your C code.