The prelab portion of this lab has to be completed at home and presented to a TA at the BEGINNING of your lab section. The prelab may contain written assignments and/or assignments to implement using your own Arduino kit. Before starting each lab, present your work papers and/or your working implementations to a TA and get their signature and keep this sheet for your records. Prelabs are individual assignments. The lab portions may be worked on in groups of two.

Lab 2 - Prelab/Home Portion

1) Explain the following: *resolution*, *reference voltage*, *quantization error*, and *step size*
2) What is the resolution of the Arduino Uno's analog to digital converter? Provide the relationship between resolution and step size
3) Convert the following voltages into their corresponding values: 1.33V; 2.71V; 2.75V; 3.99V
4) Convert the following values into their corresponding voltages: 125; 400; 938; 20
5) Plot (line graph) the relationship between the angle of a potentiometer (as described in class and textbook) on the x-axis and the raw reading of the ADC on the y-axis. Derive the equation.
6) Connect a thermistor as shown in part 1 below. By following the instructions through part c, write a program to read the ADC channel connected to the thermistor and print the RAW ADC value to the terminal program.(To be done **at home** as part of the prelab. **Demo at lab start**)

Lab 2 - Lab Portion

Components:

- 1x Thermistor
- 1x LED
- 1x 10 kΩ resistor (for the thermistor circuit)
- 1x 220 Ω resistors (for the LED)

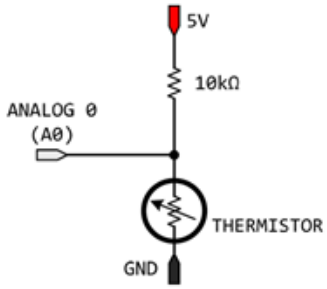
A thermistor is a device whose resistance changes with the temperature applied to it, and your kit contains a small black bead with two pins. In order to read the temperature applied to the thermistor, we use the Steinhart-Hart equation – this equation is used to find the resistance of a semiconductor at a temperature, and the Arduino implementation is included further on.

Objectives:

Part 1. *Write a program that implements a temperature sensor*

a. The program should use the thermistor to detect a temperature value.

b. The schematic for wiring the thermistor is included below:



c. Read the raw value of the thermistor, the output should be printed using the **Serial Monitor**.

i. In your **setup** function, you need to initialize the serial port using the **Serial.begin** function

ii. To print information over the serial port, use the **Serial.print** and **Serial.println** functions

iii. More information on these functions can be found in the textbook and online

d. Use the function provided below to convert the raw data into temperature, and change the **Serial Monitor** to display temperature.

e. Wire up an LED to pin 9, as done in lab 1. The brightness of the LED should be set according to the following table:

Temperature	Raw (LED Brightness)
T<25	0 (OFF)
25<=T<=35	0 ~ 255
T>35	255 (ON)

The code used to read the thermistor is included below:

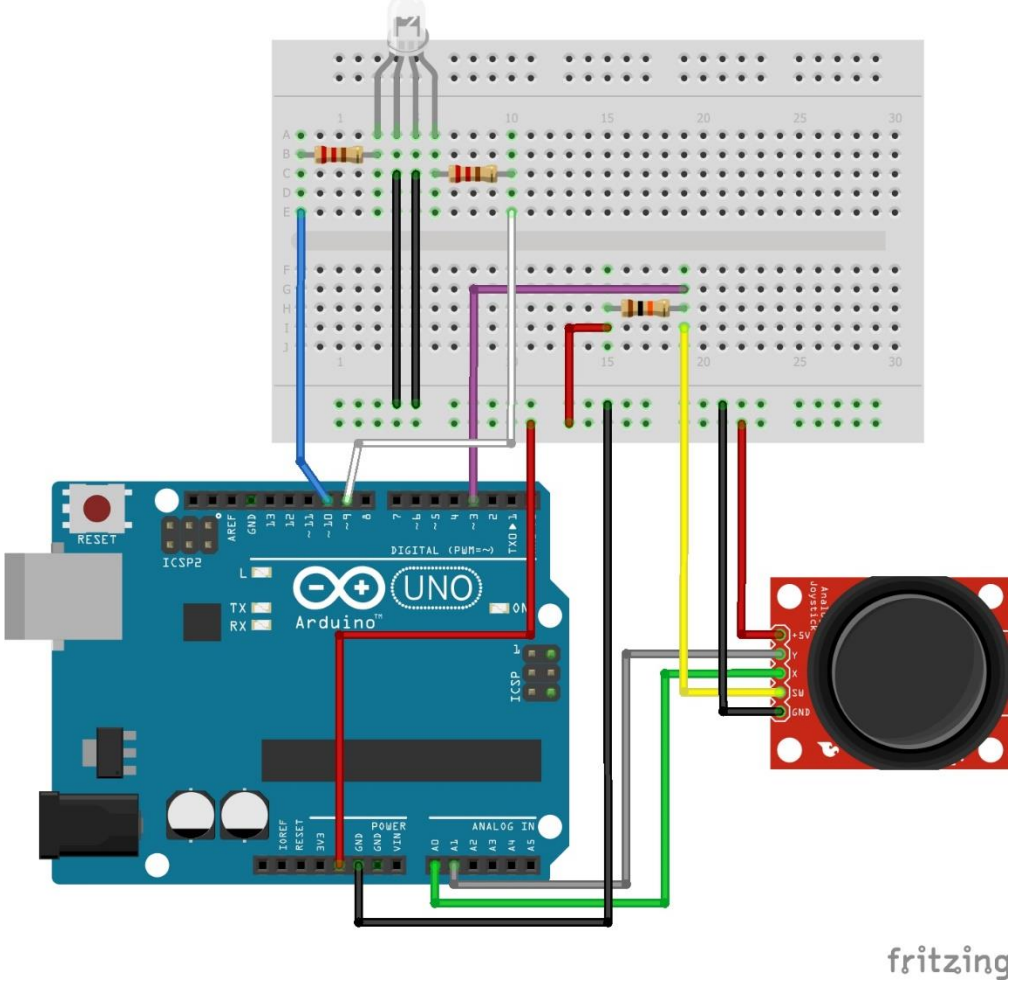
```
double readThermistor(int pin)
{
double result = 0.00;
double adc= 0.00;
int index = 0;
for(index = 0; index <10; index++)
{
adc+= analogRead(pin);
delay(10);
}
adc= adc/ 10.00;
adc= (1023.00 / adc) -1.00;
adc= 10000.0 / adc;
result= adc/ 10000.00;
result= log(result);
result= result / 3950;
result+= 1.0 / (25.00 + 273.15);
result= 1.0 / result;
result-= 273.15;
return result;
}
```

Part 2. *Write a program that implements a simple color selector using the RGB LED*

Components:

- 1x Joystick
- 1x RGB LED
- 1x 10 kΩ resistor (for joystick pushbutton)
- 2x 220 Ω resistors (for the LEDs)

a. Connect RGB LED and joystick as follows:



The joystick is simply two potentiometers, one for the X axis and one for the Y axis, set up as 0-5V voltage dividers. According to the diagram above, the x and y outputs are connected to ADC inputs 0 and 1. The joystick also has a pushbutton that we are using in part c.

b. Write a program to use the joystick's displacement to gradually select the brightness (i.e., change the PWM duty cycle) of the Blue and Red LED. Green is not used in this circuit.
   i. Blue is controlled with the X axis of the joystick.
   ii. Red is controlled with the Y axis of the joystick.
   iii. When the joystick is centered, both LEDs are 50% on. When in top right position, both are fully on. In the lower left corner, both LEDs are off.

c. Then edit the program. It should use the joystick's pushbutton to hold the LED color and print its blue and red color details to the Serial monitor. While the button is pushed, the color will not change (even if the joystick is moved), and the serial monitor will continuously print the color details.