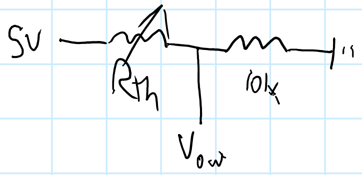
HW4 Simon Ng

Thursday, April 23, 2020 23:30

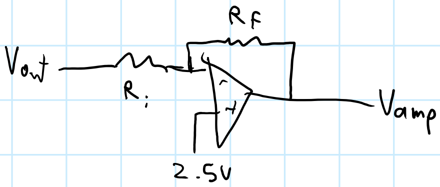
1. The thermistor is 10k, so I used a 10k resistor. I put my thermistor closer to the 5V end since it seems to make the math a bit simpler.



$$\Rightarrow \frac{5\cdot 10k}{10k+R_{th}} = V_{out}$$

$$R_{th} = \frac{5\cdot 10k}{V_{out}} - 10k$$

2. I used a 1k resistor for Ri and a 10k resistor for Rf to give the 10x inverting amplifier and used a 2.5V V+ to make sure the Arduino never gets a negative voltage.



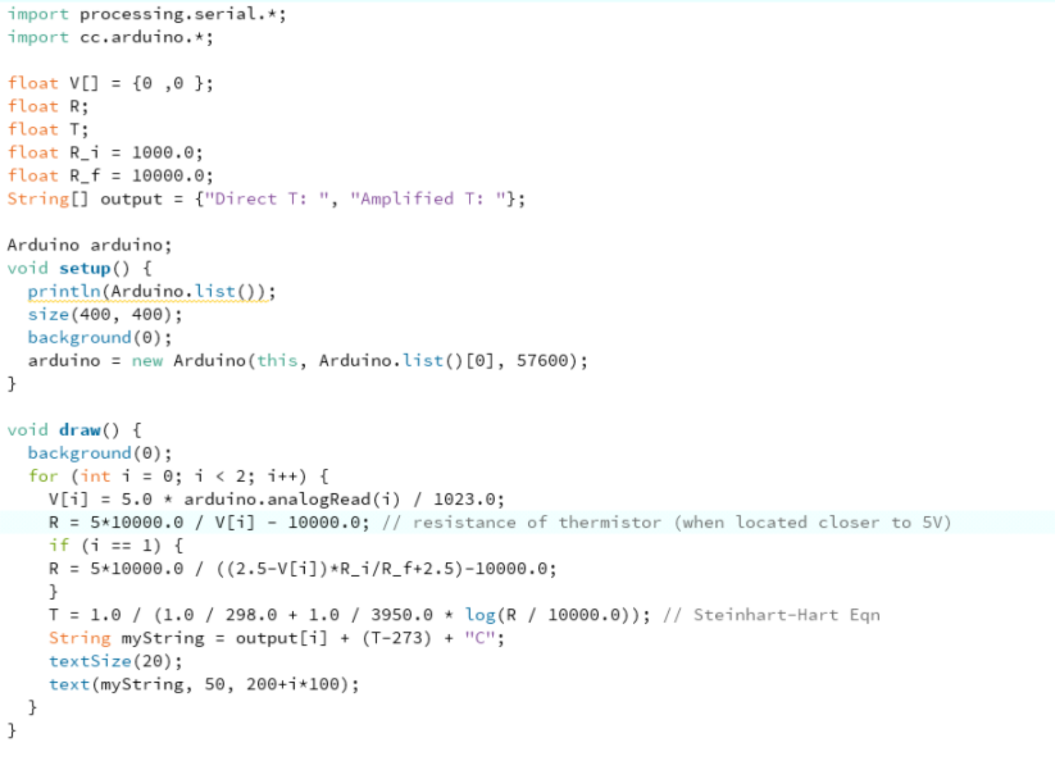
$$\frac{V_{out}-2.5}{R_i} = I = \frac{2.5-V_{amp}}{R_f}$$

$$V_{amp} = 2.5 - \frac{R_f}{R_i}\left(V_{out}-2.5\right)$$

$$R_{th} = \frac{5\cdot 10K}{(2.5-V_{amp})\frac{R_i}{R_f}+2.5} - 10k$$

$$= 2.5 - \frac{R_f}{R_i}\left(\frac{5\cdot 10k}{10k+R_{th}} - 2.5\right)$$

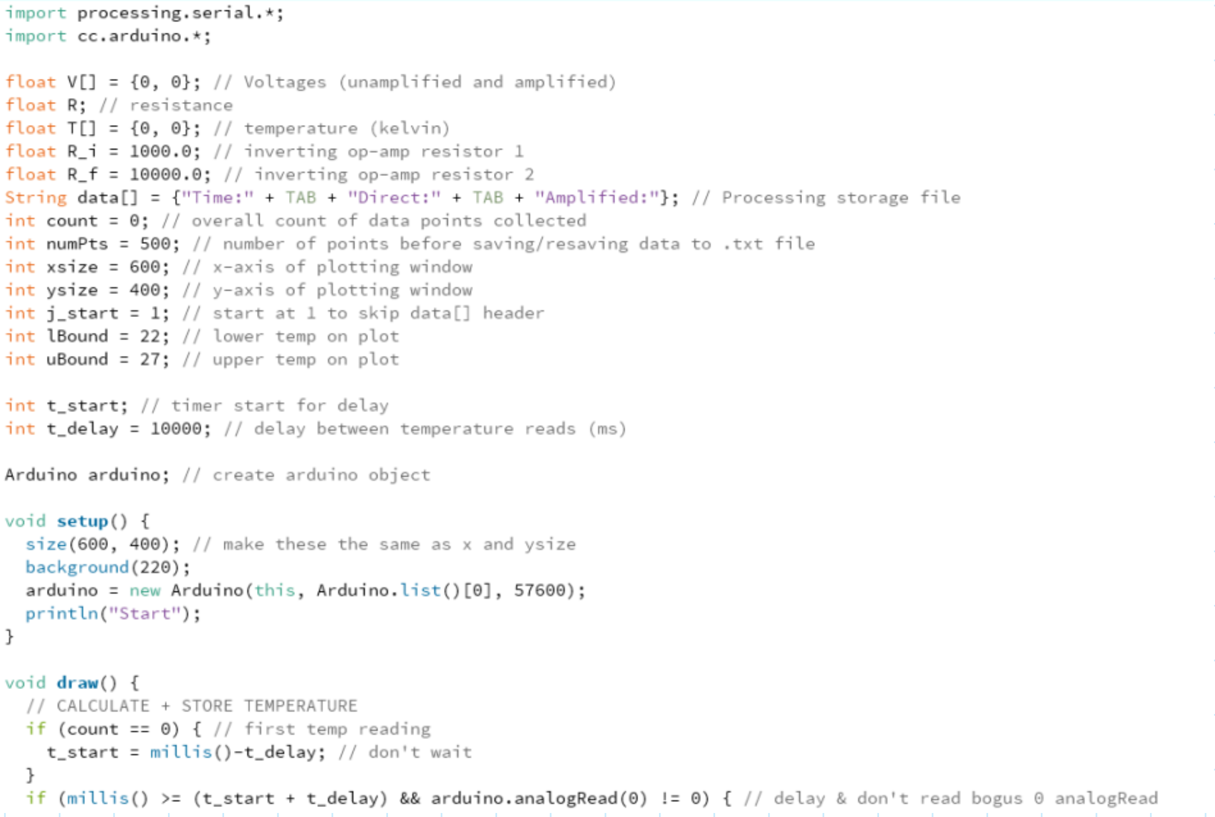
3. I struggled for a while to get the temperature to print on the screen correctly, and then it appeared as a jumble until I rem embered I have to clear the screen with background();



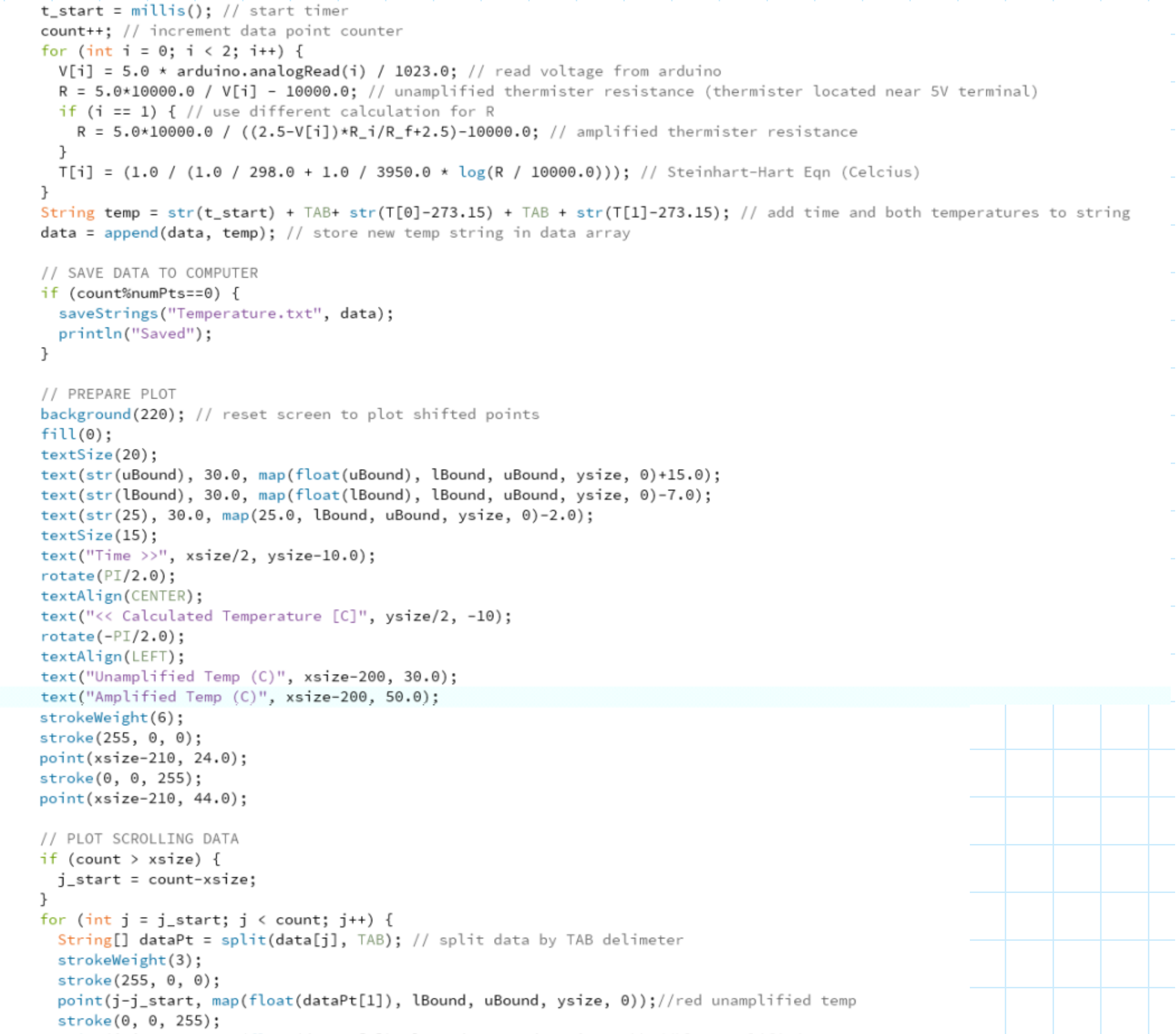
```
import processing.serial.*;
import cc.arduino.*;

float V[] = {0 ,0 };
float R;
float T;
float R_i = 1000.0;
float R_f = 10000.0;
String[] output = {"Direct T: ", "Amplified T: "};

Arduino arduino;
void setup() {
  println(Arduino.list());
  size(400, 400);
  background(0);
  arduino = new Arduino(this, Arduino.list()[0], 57600);
}

void draw() {
  background(0);
  for (int i = 0; i < 2; i++) {
    V[i] = 5.0 * arduino.analogRead(i) / 1023.0;
    R = 5*10000.0 / V[i] - 10000.0; // resistance of thermistor (when located closer to 5V)
    if (i == 1) {
    R = 5*10000.0 / ((2.5-V[i])*R_i/R_f+2.5)-10000.0;
    }
    T = 1.0 / (1.0 / 298.0 + 1.0 / 3950.0 * log(R / 10000.0)); // Steinhart-Hart Eqn
    String myString = output[i] + (T-273) + "C";
    textSize(20);
    text(myString, 50, 200+i*100);
  }
}
```

4. My unamplified reading fluxuates beyond the ones place, while my amplified reading fluxuates beyond the tenths place. It make s intuitive sense that my resolution should increase 10x since I built a 10x amplifier.
5. For some reason, occasionally, my first data point has a voltage of 0. AnalogRead is actually outputting 0. There is some con dition which gives AnalogRead a value of 0 for about 1 second. It doesn't happen every run, even when the code isn't changing. I think it must b e some time-based condition that lines up just right sometimes. I added a condition to not print the temperature if AnalogRead is 0.
   To make a scrolling graph, I used the x-axis pixel number to select and display only the most recent datapoints.
6. I was briefly concerned that using int for my millis() variables might not be enough storage space, but then I found that pro cessing ints are 32 bits, so it should last me around 25 days, so I'm not concerned anymore. My unamplified temperature sits about 2 degrees below my a mplified temperature until I touch it, when it gains approximately the same value as the amplified temperature. After more exploration , this problem seems

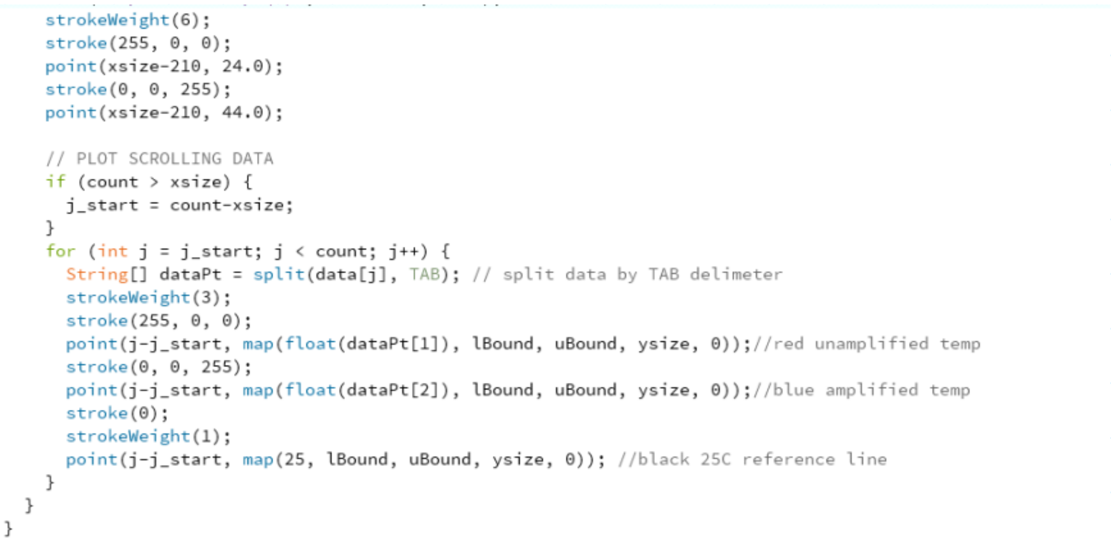
to be due to poor breadboard connection. After that, my code and hardware seemed to work well. More analysis is with two plot s below the code.



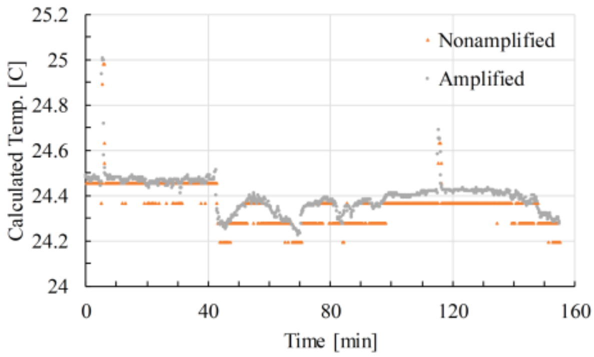
```
import processing.serial.*;
import cc.arduino.*;

float V[] = {0, 0}; // Voltages (unamplified and amplified)
float R; // resistance
float T[] = {0, 0}; // temperature (kelvin)
float R_i = 1000.0; // inverting op-amp resistor 1
float R_f = 10000.0; // inverting op-amp resistor 2
String data[] = {"Time:" + TAB + "Direct:" + TAB + "Amplified:"}; // Processing storage file
int count = 0; // overall count of data points collected
int numPts = 500; // number of points before saving/resaving data to .txt file
int xsize = 600; // x-axis of plotting window
int ysize = 400; // y-axis of plotting window
int j_start = 1; // start at 1 to skip data[] header
int lBound = 22; // lower temp on plot
int uBound = 27; // upper temp on plot

int t_start; // timer start for delay
int t_delay = 10000; // delay between temperature reads (ms)

Arduino arduino; // create arduino object

void setup() {
  size(600, 400); // make these the same as x and ysize
  background(220);
  arduino = new Arduino(this, Arduino.list()[0], 57600);
  println("Start");
}

void draw() {
  // CALCULATE + STORE TEMPERATURE
  if (count == 0) { // first temp reading
    t_start = millis()-t_delay; // don't wait
  }
  if (millis() >= (t_start + t_delay) && arduino.analogRead(0) != 0) { // delay & don't read bogus 0 analogRead
```



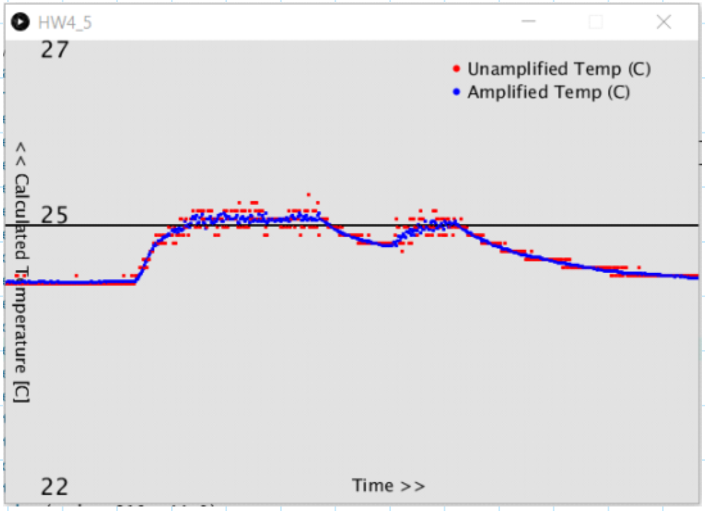
```
    t_start = millis(); // start timer
    count++; // increment data point counter
    for (int i = 0; i < 2; i++) {
      V[i] = 5.0 * arduino.analogRead(i) / 1023.0; // read voltage from arduino
      R = 5.0*10000.0 / V[i] - 10000.0; // unamplified thermister resistance (thermister located near 5V terminal)
      if (i == 1) { // use different calculation for R
        R = 5.0*10000.0 / ((2.5-V[i])*R_i/R_f+2.5)-10000.0; // amplified thermister resistance
      }
      T[i] = (1.0 / (1.0 / 298.0 + 1.0 / 3950.0 * log(R / 10000.0))); // Steinhart-Hart Eqn (Celcius)
    }
    String temp = str(t_start) + TAB+ str(T[0]-273.15) + TAB + str(T[1]-273.15); // add time and both temperatures to string
    data = append(data, temp); // store new temp string in data array

    // SAVE DATA TO COMPUTER
    if (count%numPts==0) {
      saveStrings("Temperature.txt", data);
      println("Saved");
    }

    // PREPARE PLOT
    background(220); // reset screen to plot shifted points
    fill(0);
    textSize(20);
    text(str(uBound), 30.0, map(float(uBound), lBound, uBound, ysize, 0)+15.0);
    text(str(lBound), 30.0, map(float(lBound), lBound, uBound, ysize, 0)-7.0);
    text(str(25), 30.0, map(25.0, lBound, uBound, ysize, 0)-2.0);
    textSize(15);
    text("Time >>", xsize/2, ysize-10.0);
    rotate(PI/2.0);
    textAlign(CENTER);
    text("<< Calculated Temperature [C]", ysize/2, -10);
    rotate(-PI/2.0);
    textAlign(LEFT);
    text("Unamplified Temp (C)", xsize-200, 30.0);
    text("Amplified Temp (C)", xsize-200, 50.0);
    strokeWeight(6);
    stroke(255, 0, 0);
    point(xsize-210, 24.0);
    stroke(0, 0, 255);
    point(xsize-210, 44.0);

    // PLOT SCROLLING DATA
    if (count > xsize) {
      j_start = count-xsize;
    }
    for (int j = j_start; j < count; j++) {
      String[] dataPt = split(data[j], TAB); // split data by TAB delimeter
      strokeWeight(3);
      stroke(255, 0, 0);
      point(j-j_start, map(float(dataPt[1]), lBound, uBound, ysize, 0));//red unamplified temp
      stroke(0, 0, 255);
```



```
    strokeWeight(6);
    stroke(255, 0, 0);
    point(xsize-210, 24.0);
    stroke(0, 0, 255);
    point(xsize-210, 44.0);

    // PLOT SCROLLING DATA
    if (count > xsize) {
      j_start = count-xsize;
    }
    for (int j = j_start; j < count; j++) {
      String[] dataPt = split(data[j], TAB); // split data by TAB delimeter
      strokeWeight(3);
      stroke(255, 0, 0);
      point(j-j_start, map(float(dataPt[1]), lBound, uBound, ysize, 0));//red unamplified temp
      stroke(0, 0, 255);
      point(j-j_start, map(float(dataPt[2]), lBound, uBound, ysize, 0));//blue amplified temp
      stroke(0);
      strokeWeight(1);
      point(j-j_start, map(25, lBound, uBound, ysize, 0)); //black 25C reference line
    }
  }
}
```



When I read the voltage every 10 seconds, I produced the above plot in Excel and also plotted it in real time on Processing.
The large spikes are times I held the thermistor for a minute or so, and the large dip around 45 minutes is when I moved the sensor to my windowsill, which is colder than the rest of the room. After I recorded the live, scrolling version, I wanted a legend and axes labels, so I rerecorded the scrolling Processing plot at 100 ms/sample below. The resolution difference is easier seen in these graphs, where the unamplified temperature clearly has larger discrete levels which it can occupy, indicative of the lower resolution, while the amplified temperature shows more of a gradient in between the unamplified temperature's discrete levels. From the Excel plot, it appears that the unamplified temperature can sense differences of slightly below 0.2 C.


HW4_5
27
25
22
Unamplified Temp (C)
Amplified Temp (C)
<< Calculated Temperature [C]
Time >>