ECE 476/676 - Homework #8

Text Files, Temperature Sensors - Due Monday, March 24th

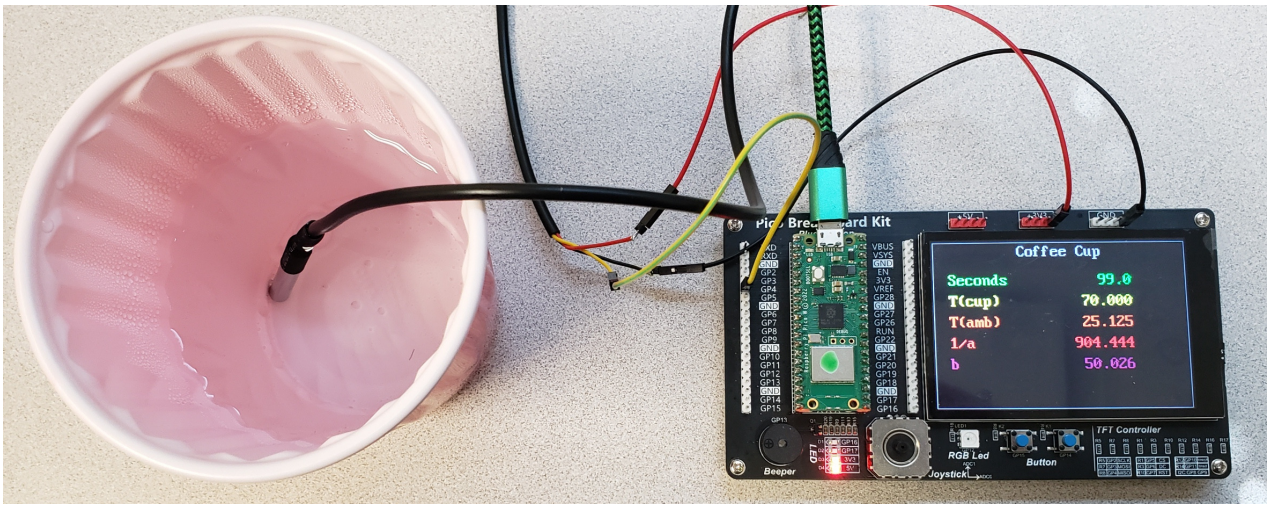
Temperature Sensor: Thermal Time Constant

1) Write a Python program which

- Records the temperature of a DS18B20 sensor every 1.0 seconds
- Displays the data on the LCD display, and
- Saves the data to a text file

Hardware:

- Connect a DS18B20 sensor to GP14
- The rest of everything is part of the 52Pi Breakout Board Plus



Software:

- Initialize the ST7796S graphics display using the LCD library on Bison Academy
- Initalize the DS18B20 temperature sensor
- Use interrupts to set the sampling rate to 1 second (1000ms)
- Read the sensor 1000 times, saving the data to a text file and displaying it on the LCD display

Code:

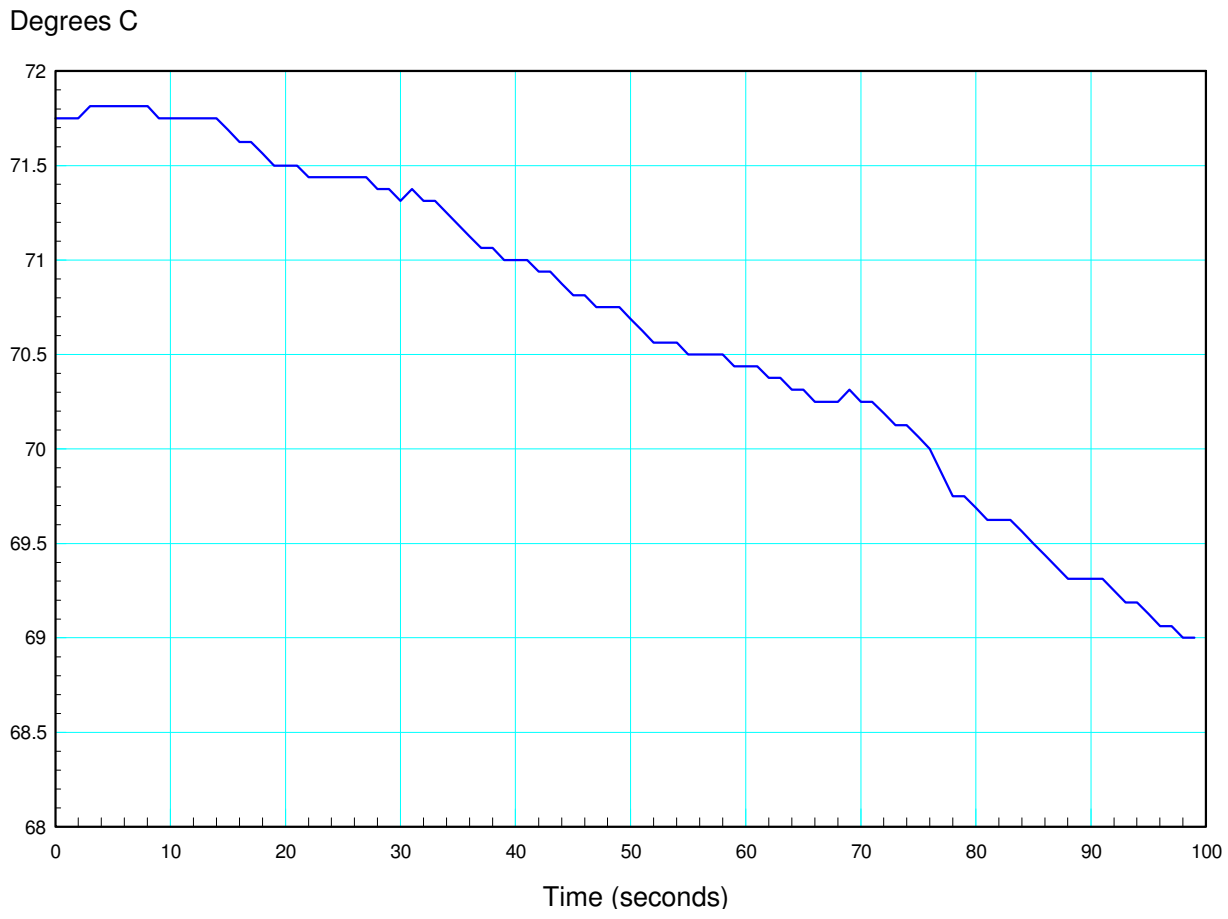
```
import onewire, ds18x20
from machine import Pin, ADC, Timer
from time import sleep, sleep_ms
import LCD

ds_pin = Pin(4)
ds_sensor = ds18x20.DS18X20(onewire.OneWire(ds_pin))

roms = ds_sensor.scan()
print('Found DS devices: ', roms)

flag = 1

def tick(timer):
    global flag
    flag = 1

Time = Timer()
Time.init(period = 1000, mode=Timer.PERIODIC, callback=tick)

file1 = open("HW8_Prob1", "a")
file1.write('----------------------\n')
file1.write('Seconds  Degrees C\n')

Navy = LCD.RGB(0,0,5)
LtGreen = LCD.RGB(100,250,50)
Yellow = LCD.RGB(250,250,0)
Orange = LCD.RGB(250,150,50)
Red    = LCD.RGB(250,50,50)
White = LCD.RGB(200,200,200)

LCD.Init()
LCD.Clear(Navy)
LCD.Box(2,2,478,318,White)
LCD.Title('Coffee Cup', White, Navy)
LCD.Text2('Seconds',30,60, LtGreen, Navy)
LCD.Text2('Degrees C',30,100,Yellow,Navy)
LCD.Text2('a',30,140,Orange,Navy)
LCD.Text2('b',30,180,Red,Navy)
time = 0

while(time < 100):
    while(flag == 0):
        pass
    flag = 0
    ds_sensor.convert_temp()
    sleep_ms(750)
    Temp = ds_sensor.read_temp(roms[0])
    print(time, Temp)

    file1.write(str('{: 4.0f}'.format(time)) + " ")
    file1.write(str('{: 7.4f}'.format(Temp)) + " ")
    file1.write("\n")

    LCD.Number2(time, 9, 1, 200, 60, LtGreen, Navy)
    LCD.Number2(Temp, 9, 3, 200, 100, Yellow, Navy)

    time += T

file1.close()
```

2) Using this program, record

- Pick something that gets hot or cold (cup of tea, light bulb when turned on, motor, etc)
- Record the temperature of this object for 1 minute or more

The data looks like...

```
----------------------
Seconds  Degrees C
   0  71.7500
   1  71.7500
   2  71.7500
   3  71.8125
   4  71.8125
   5  71.8125
   6  71.8125
   7  71.8125
   8  71.8125
   9  71.7500
  10  71.7500
```



3) Modify this program to

- Compute the thermal time constant (1/a) using recursive least squares and
- Display the temperature and thermal time constant on the LCD display

$$T = b \cdot \exp(-at) + T_{amb} \qquad \ln(T - T_{amb}) = \ln(b) - at3$$

Writing this as

$$y = ax + b$$

In matrix form

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$Y = BA$$

A is then

$$A = (B^T B)^{-1} B^T Y$$

$$B^T B = \begin{bmatrix} \Sigma x^2 & \Sigma x \\ \Sigma x & n \end{bmatrix}$$

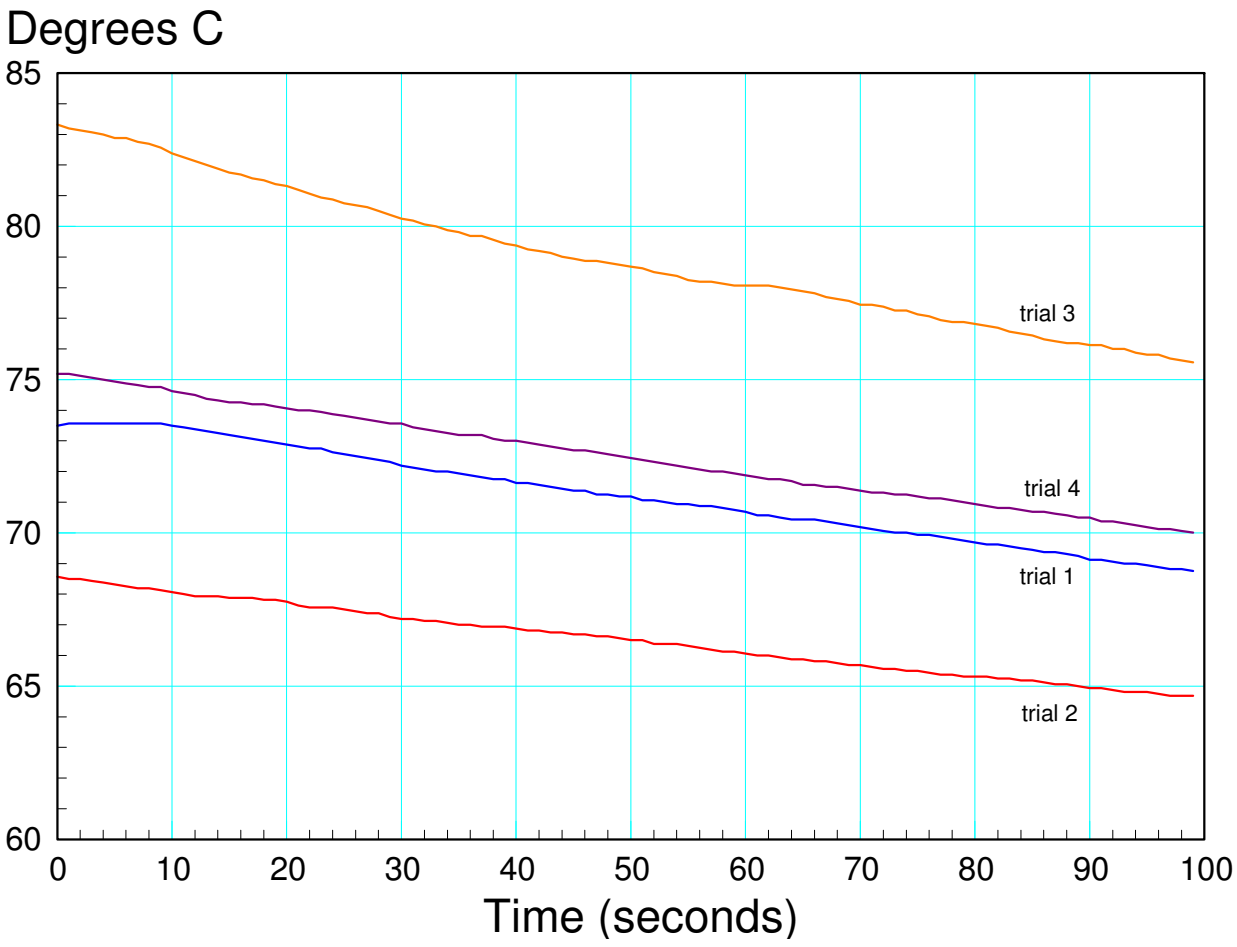
$$B^T Y = \begin{bmatrix} \Sigma xy \\ \Sigma y \end{bmatrix}$$

You likewise only need to keep track of these two matricies. In code (the main loop)

```
while(time < 100):
    while(flag == 0):
        pass
    flag = 0
    ds_sensor.convert_temp()
    sleep_ms(750)
    Temp = ds_sensor.read_temp(roms[0])

    x = time
    y = log(abs(Temp - Tamb))
    n += 1

    B = matrix.add(B, [[x**2, x], [x, 1]])
    Y = matrix.add(Y, [[x*y], [y]])

    if(n > 2):
        Bi = matrix.inv(B)
        A = matrix.mult(Bi, Y)
    if(A[0][0] != 0):
        a = -1 / A[0][0]
    else:
        a = 0
    b = exp(abs(A[1][0]))

    time += 1
file.close
```

4) Collect three sets of data (three different measurements of the thermal time constant)

Trial	1	2	3	4
Time Constant (sec)	888.6473	1051.5005	705.6663	904.4443



5) From your data, determine the 90% confidence interval for the thermal time constant (1/a)

In Matlab:

```
>> T = [888.6473     1051.5005   705.6663    904.4443 ];
>> x = mean(T)
x =  887.5646

>> s = std(T)
s =  141.7134
```

The t-score for 5% tails and 3 degrees of freedo is 2.3534

For any given trial, the 90% confidence interval is (554.056, 1221.1) seconds

```
>> x + 2.3534*s
ans =  1.2211e+003

>> x - 2.3534*s
ans =  554.0563
```

The thermal time constant of this cup is (population) with p=90% is (720.8, 1054.3) seconds

```
>> x - 2.3534*s/sqrt(4)
ans =  720.8104

>> x + 2.3534*s/sqrt(4)
ans =  1.0543e+003
```