Lab Report 3: Hotplate Temperature Sensing

Max Huggins – UCA Department of Physics Astronomy

February 9, 2021

Abstract

To learn about interfacing with the Raspberry Pi, an experiment was done to measure the temperature of a hotplate with an array of 25 temperature sensors.

1 Introduction

Data collected with multiple sensors can lead to some interesting challenges and data processing. In this project, 4 MCP3008 analog to digital converter (ADC) chips were used to record data from 25 MCP9700A temperature sensors. The hotplate was a 6" $*$ 6" $*$ $\frac{3}{4}$" aluminum slab with a heating element through it's center. A python script was written to record temperature and time data with the general flow outlined in figure 1. The actual script is available in the Appendix.
To estimate the time the slab will take to reach the desired temperature, a series of calculations were made (shown below.)

$$Q = mC\Delta T \tag{1}$$

$$Q = Pt \tag{2}$$

$$m = \rho V \tag{3}$$

Therefore,

$$t = \frac{\rho V C \Delta T}{P} \tag{4}$$

Where C is the heat capacity of aluminum ($900\frac{J}{kg}$), P is the power output of the heater ($300W$), ρ is the density of aluminum ($2700\frac{kg}{m^3}$), V is the volume of the slab ($4.4 * 10^{-4} m^3$), and ΔT is the change in temperature ($75^o C$.) This comes out to a time of approximately $4.5 mins$.

2 Analysis

Two methods were used to process this data. The first was a simple temperature versus time graph with all 25 sensors. The Python script is available in the Appendix and the plots from it are shown in figure 2. It can clearly be seen that as time increases, temperature increases.

Unfortunately, this is not telling us anything about the position of the sensor and its relation to the hotplate. In an attempt to create a more descriptive image of how the temperature is changing with respect to the position of the sensor, a 5x5 heat map was created with each block representing the temperature of the sensor with a red-yellow color scale. Six snaps in time are shown in figure 3 going left to right, top to bottom as time increases. A .gif file type was made out of a composite of 248 data points to see the overall trend of the hotplate. Refer to the Appendix for the Python script.

3 Experimental Setup

Here are several figures showing schematics, plots, and charts for the experiment.

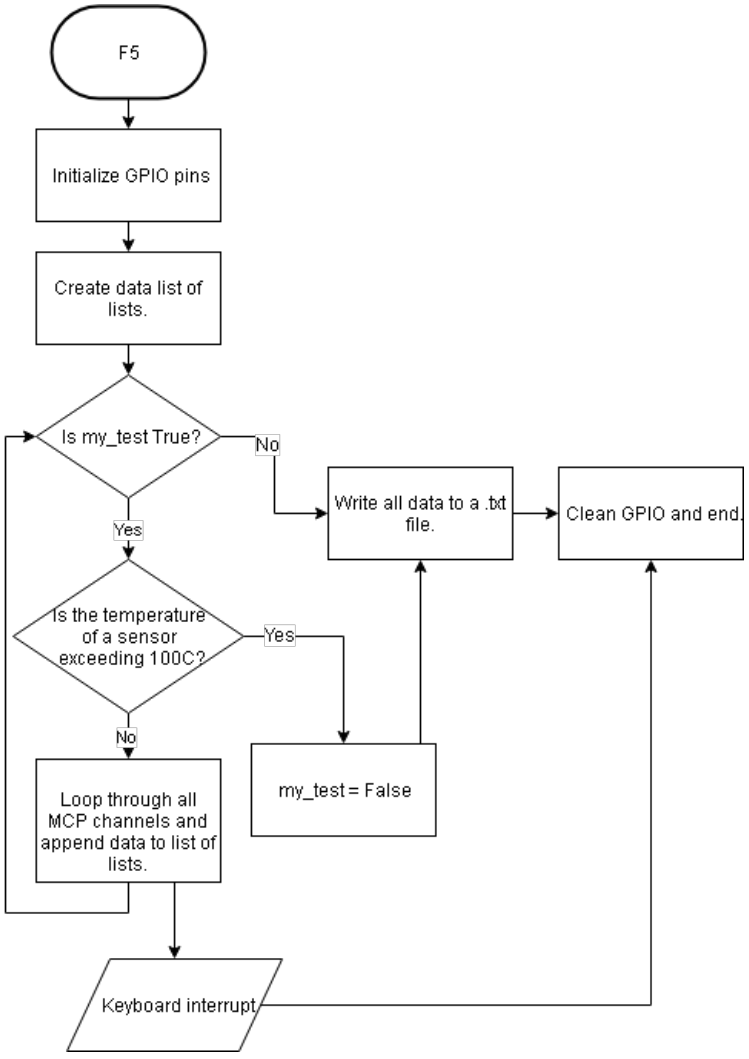

Figure 1: This is the general flow of the data collection script in Python.

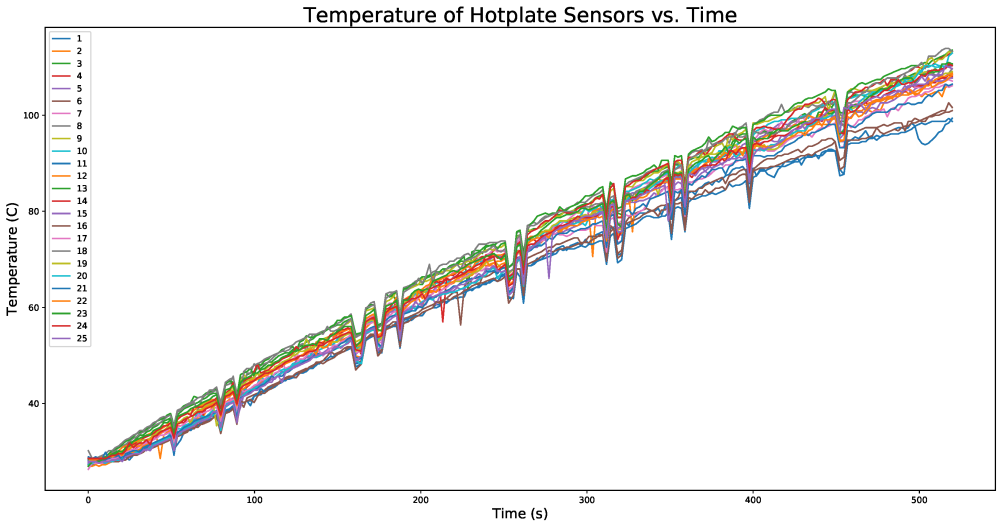

Figure 2: These are the plots of the temperature vs time data for each sensor.

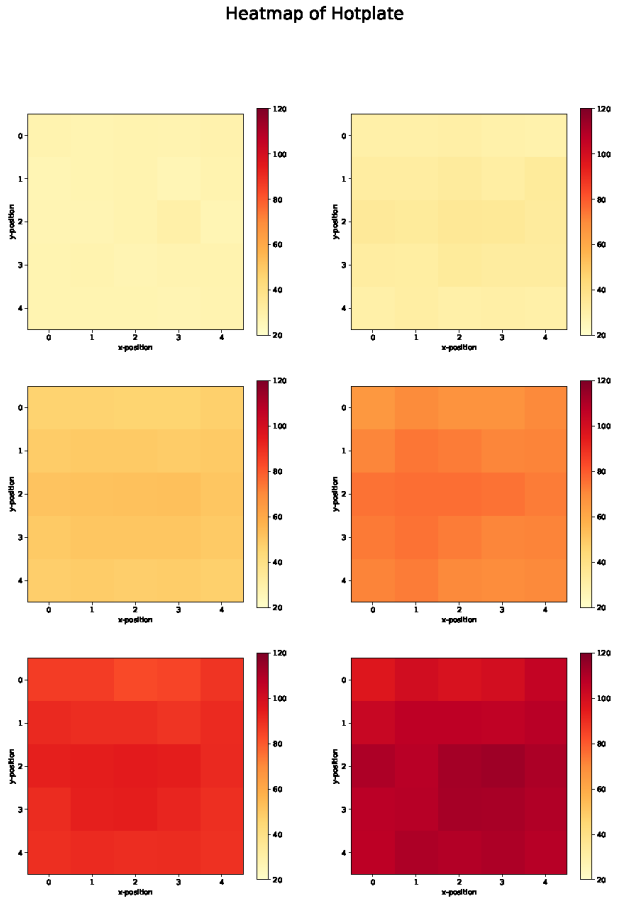

Figure 3: These are the plots of the temperature vs position data for each sensor at six different times.

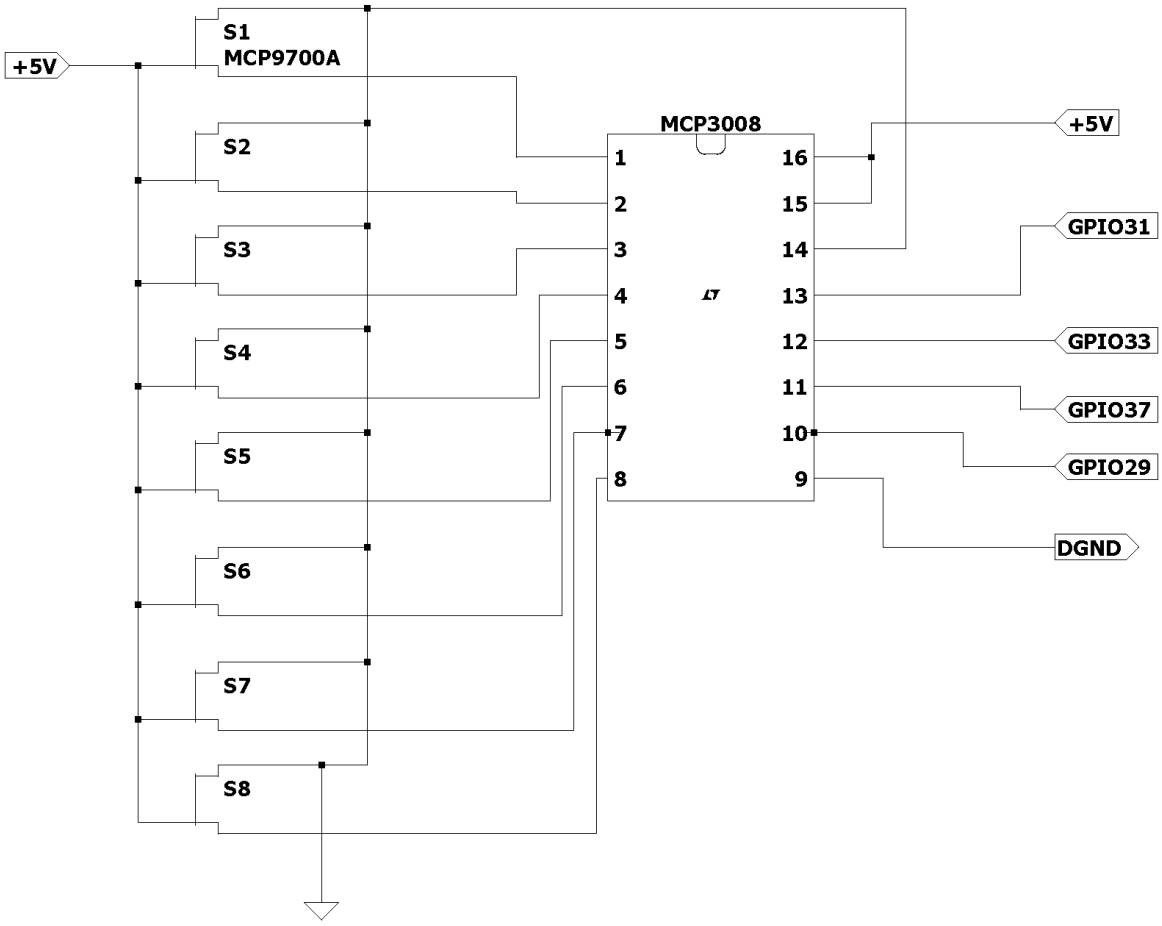

Figure 4: This is an example of how one of the MCP converter chips are set up.

4 Conclusion

This experiment was a good start to temperature sensing with an array of sensors. Next steps should include a more stable power source for the sensors. Large dips can be seen from figure 2 in the temperature data; which I contribute to instability in power supply voltages. Also, once this data is collected interpolating functions from them would be helpful in increasing the resolution of the heat map in figure 3. These results only scratch the surface regarding the scope of this project; more analysis is needed to expand on them.

5 Appendix

Here are several scripts that control the data collecting, plotting, and heat map plotting.

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar  1 15:00:27 2019

@author: maxhu
"""

import RPi.GPIO as GPIO
import time
import uControllersDataAcquisition as DtA
#============================================================#
#This segment of code deals with setting up GPIO pins for the MCP
GPIO.setmode(GPIO.BOARD)

CS = [29, 32, 18, 8]
CLK = [31, 36, 22, 10]
DOUT = [33, 38, 24, 12]
DIN = [37, 40, 26, 16]

for i in range(0,4):
    GPIO.setup(CS[i], GPIO.OUT)
    GPIO.setup(CLK[i], GPIO.OUT)
    GPIO.setup(DOUT[i], GPIO.IN)
    GPIO.setup(DIN[i], GPIO.OUT)
#============================================================#
#This creates two lists of lists that can hold 25 different sets of
#data
TEMP = [[] for i in range(0,25)]
TIME = [[] for i in range(0,25)]
#============================================================#
#Here is where the magic happens... **--*-***---*
try:
    my_test = True
    start_time = time.time()
    while my_test == True:
        #This is in charge of ending the program in case a sensor
        #exceeds 100C and ensures data is making sense for user.
        test = DtA.calc_tempMCPBudgetLM34(DtA.readMCP(0, CS[0], CLK[0],
        DOUT[0], DIN[0]))
        print(test)
        test = float(test)
        if test > 100:
            break
        #These two loops go through each MCP(i) and each MCP channel (n)
        for i in range(0,4):
            for n in range(0,8):
                if i == 3 and n == 1: #last channel of the last MCP that
                    #needs data
                    break
                #It reads from a function defined in data acquisition file
                d = DtA.readMCP(n, CS[i], CLK[i], DOUT[i], DIN[i])
                TIME[8 * i + n].append(time.time() - start_time)
                t = DtA.calc_tempMCPBudgetLM34(d)
                TEMP[8 * i + n].append(t)
```

```
55 #=========================================================================#
56 #This writes data to a file
57         file = open('./Data/HOTPLATETESTING.txt', 'w')
58         for i in range(0,25):
59             for n in range(0,len(TIME[i])):
60                 file.write(str(TIME[i][n]) + ',' + TEMP[i][n] + '\n')
61         file.close()
62 
63 except KeyboardInterrupt:
64     print("It's fried, dude.")
65 
66 finally:
67     GPIO.cleanup()
68     print('Isaac cleaned the oven')
```

```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb 28 22:03:09 2019

@author: maxhu
"""

#Plotting HotPlate Stuff
#max huggins
#2/20/19

import matplotlib.pyplot as plt
#=============================================================================#
#define data arrays
TempDataLists = [[] for i in range(0,25)]
TimeDataLists = [[] for i in range(0,25)]
#=============================================================================#
#read in data and assign to data arrays.
for i in range(1,26):
    with open('./NewDocs/TempData{}.txt'.format(i), 'r') as f:
        lines = []
        lines = f.readlines()
        temps = []
        times = []
        for n in range(len(lines)):
            stuff = lines[n].split(',')
            #only every 100th data point is used
            if n % 100 == 0:
                times.append(float(stuff[0].strip()))
                temps.append(float(stuff[1].strip()))
        for s in range(0,len(times)):
            TempDataLists[i-1].append(temps[s])
            TimeDataLists[i-1].append(times[s])
#=============================================================================#
#This handles plotting, labeling, and saving.
fig = plt.figure(figsize=(20,10))
ax = fig.add_subplot(1,1,1)

for i in range(0,25):
        plt.plot(TimeDataLists[i], TempDataLists[i], label = i+1)

ax.set_xlabel('Time (s)')
ax.set_ylabel('Temperature (C)')
ax.set_title('Temperature of Hotplate Sensors vs. Time')
plt.legend(loc = 'best')

#saving as a pdf for maximum resolution
plt.savefig('hotplateMultiple.pdf', format='pdf')
```

```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb 28 22:03:09 2019

@author: maxhu
"""

#Plotting HotPlate Stuff
#max huggins
#2/20/19
import numpy as np
import matplotlib.pyplot as plt
#=============================================================================#
#define data arrays
TempDataLists = [[] for i in range(0,25)]
TimeDataLists = [[] for i in range(0,25)]
#=============================================================================#
#read in data and assign to data arrays.
for i in range(1,26):
    with open('./NewDocs/TempData{}.txt'.format(i), 'r') as f:
        lines = []
        lines = f.readlines()
        temps = []
        times = []
        for n in range(len(lines)):
            stuff = lines[n].split(',')
            #only every 100th data point is used
            if n % 100 == 0:
                times.append(float(stuff[0].strip()))
                temps.append(float(stuff[1].strip()))
        for s in range(0,len(times)):
            TempDataLists[i-1].append(temps[s])
            TimeDataLists[i-1].append(times[s])
#=============================================================================#
map_list = []

for t in range(len(TimeDataLists[0])):
    temp_points = np.zeros([5,5])
    for q in range(0,5):
        y = q
        for w in range(0,5):
            x = w
            temp_points[x][y] =  TempDataLists[x + 5*y][t]
    map_list.append(temp_points)

plt.figure(1)
hmap = plt.imshow(map_list[0], vmin=20, vmax=120)
plt.colorbar(hmap)


file_name = './Heatmaps/{:03d}_hotplate_heatmap.jpg'

for i in range(len(map_list)):

```

```
    fig = plt.figure()
    my_fig = fig.add_subplot(111,aspect='equal')
    hmap = plt.imshow(map_list[i], vmin=20, vmax=120)
    plt.colorbar(hmap)
    my_fig.set_xlabel('x-position')
    my_fig.set_ylabel('y-position')
    my_fig.set_title('Heatmap of Hotplate')
    plt.savefig(file_name. format(i))
    plt.close()
```