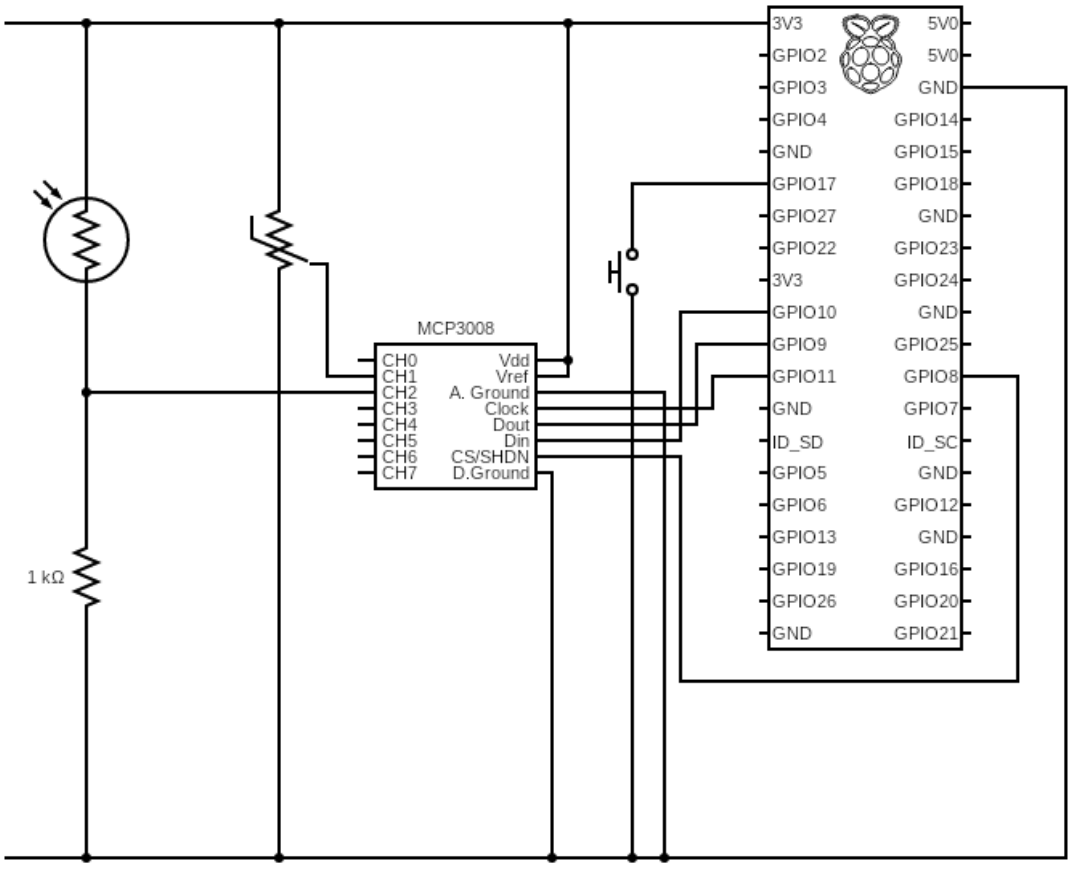
Prac4_LNGANG002_BTJMAL001

October 8, 2021

1 EEE3096S Practical 4

1.1 Authors:

1.1.1 Angus Longmore (LNGANG002) and Malcolm Baatjies (BTJMAL001)



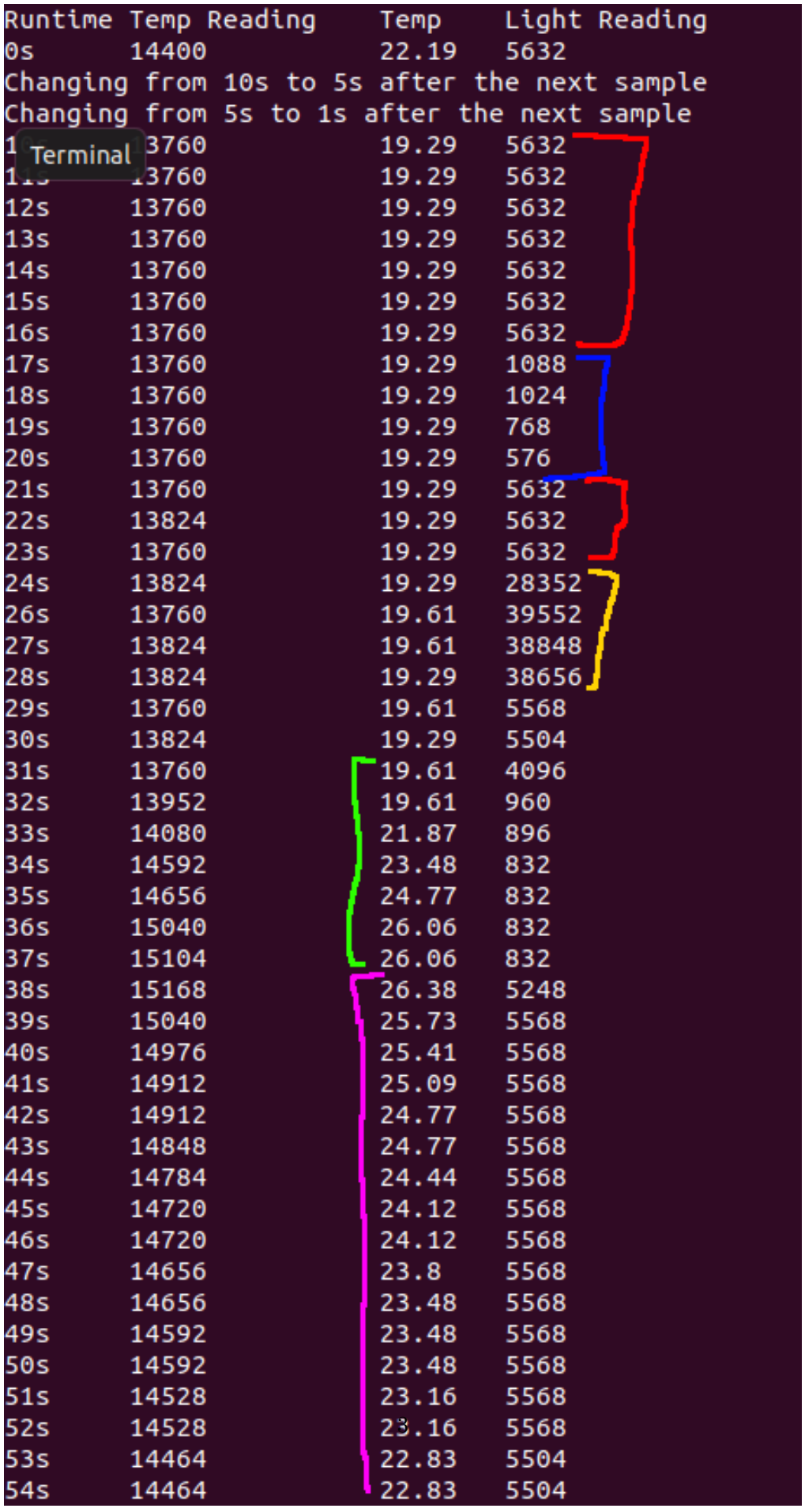
1.2 Validation and Testing

The first tests for the ldr involved measuring the readings at ambient, then covering the light, observing the readings decrease, shining a torch on the light and observing the readings increase. To test for the Thermistor, the ADC readings were converted to Celcius, then compared to the value of a mercury thermometer. The readings were seperated by a degree and half.

The Pi was then restarted and the tests redone to validate the results. We obtained very similar readings for ambient temperature and light, the increase/decreases were also consistent with our initial testing.

The image below is a sample of some of the testing. The sections outlined in red are ambient readings.The section in blue is the light sensor being covered. The section inn yellow is the light sensor having a torch shined at it. The section in green is the thermistor being held by fingers (increasing the temperature), notice the ramp up as the thermistor warms. You'll also notice during this section the ldr readings drop, this is due to the ldr being close to the thermistor, and the hand of the operator blocking the light to the ldr. The pink section is when the thermistor has been released and is returning to ambient slowly.

We will demo our program in person next week, as indicated we can do on the discord server, so no video is linked.


Runtime Temp Reading Temp Light Reading
0s 14400 22.19 5632
Changing from 10s to 5s after the next sample
Changing from 5s to 1s after the next sample
Terminal
12s 13760 19.29 5632
13s 13760 19.29 5632
14s 13760 19.29 5632
15s 13760 19.29 5632
16s 13760 19.29 5632
17s 13760 19.29 1088
18s 13760 19.29 1024
19s 13760 19.29 768
20s 13760 19.29 576
21s 13760 19.29 5632
22s 13824 19.29 5632
23s 13760 19.29 5632
24s 13824 19.29 28352
26s 13760 19.61 39552
27s 13824 19.61 38848
28s 13824 19.29 38656
29s 13760 19.61 5568
30s 13824 19.29 5504
31s 13760 19.61 4096
32s 13952 19.61 960
33s 14080 21.87 896
34s 14592 23.48 832
35s 14656 24.77 832
36s 15040 26.06 832
37s 15104 26.06 832
38s 15168 26.38 5248
39s 15040 25.73 5568
40s 14976 25.41 5568
41s 14912 25.09 5568
42s 14912 24.77 5568
43s 14848 24.77 5568
44s 14784 24.44 5568
45s 14720 24.12 5568
46s 14720 24.12 5568
47s 14656 23.8 5568
48s 14656 23.48 5568
49s 14592 23.48 5568
50s 14592 23.48 5568
51s 14528 23.16 5568
52s 14528 23.16 5568
53s 14464 22.83 5504
54s 14464 22.83 5504

```
[ ]: import busio
     import digitalio
     import board
     import threading
     import RPi.GPIO as GPIO
     from time import time, sleep
     import adafruit_mcp3xxx.mcp3008 as MCP
     from adafruit_mcp3xxx.analog_in import AnalogIn

     def print_sensor_vals(start_time, temp, ldr):
         print(f"{round(time()-start_time)}s\t{temp.value}\t \t{round(((temp.
      ↪voltage-0.5)/0.01), 2)}\t{round(ldr.value, 2)}")


     def cycle_sample_time(channel):
         global current_sampling_time_index, sampling_times
         print(f"Changing from {sampling_times[current_sampling_time_index]}s to␣
      ↪{sampling_times[(current_sampling_time_index+1)%3]}s after the next sample")
         current_sampling_time_index = (current_sampling_time_index +1) %3

     sampling_times = [10, 5, 1]
     current_sampling_time_index = 0

     #create the spi bus
     spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

     # create the cs (chip select)
     cs = digitalio.DigitalInOut(board.D5)


     # create the mcp object
     mcp = MCP.MCP3008(spi, cs)

     # Add button callback to change sampling rate
     GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
     GPIO.add_event_detect(17, GPIO.FALLING, callback=cycle_sample_time,␣
      ↪bouncetime=200)

     start_time = time()
     ldr = AnalogIn(mcp, MCP.P2)
     temp = AnalogIn(mcp, MCP.P1)
     print("Runtime\tTemp Reading\tTemp\tLight Reading")
     x = threading.Thread(target=print_sensor_vals, args=(start_time, temp, ldr))
     while True:
         x.start()
         x.join()
         wait_time = sampling_times[current_sampling_time_index]
```

```
    sleep(wait_time)
```