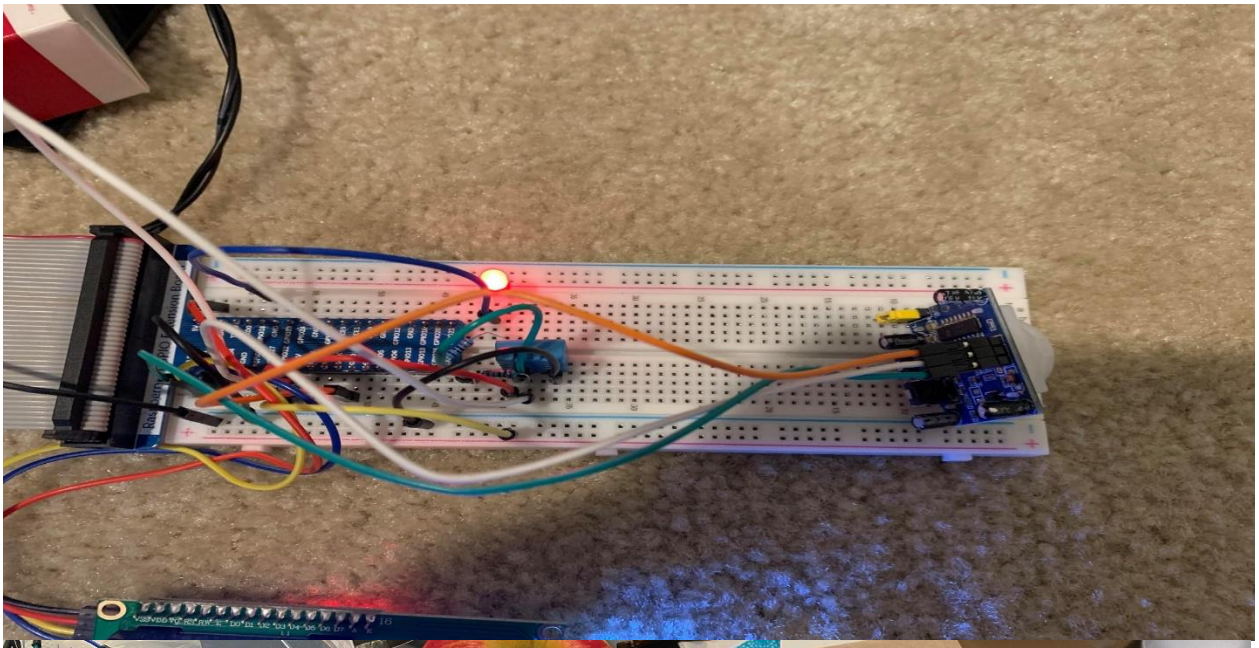
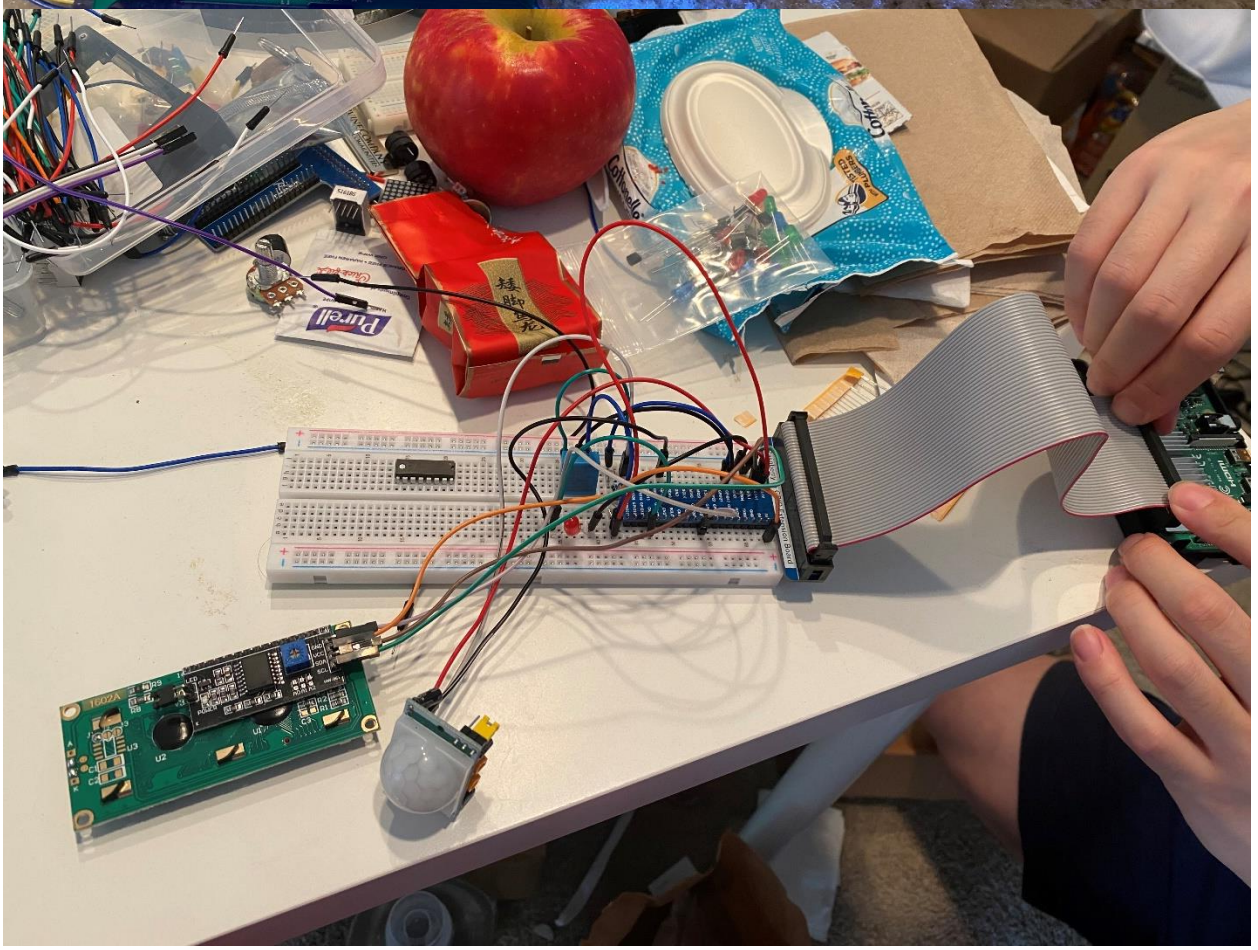
EECS 113
Final Project Report

Liyuan Zhao (43307320)

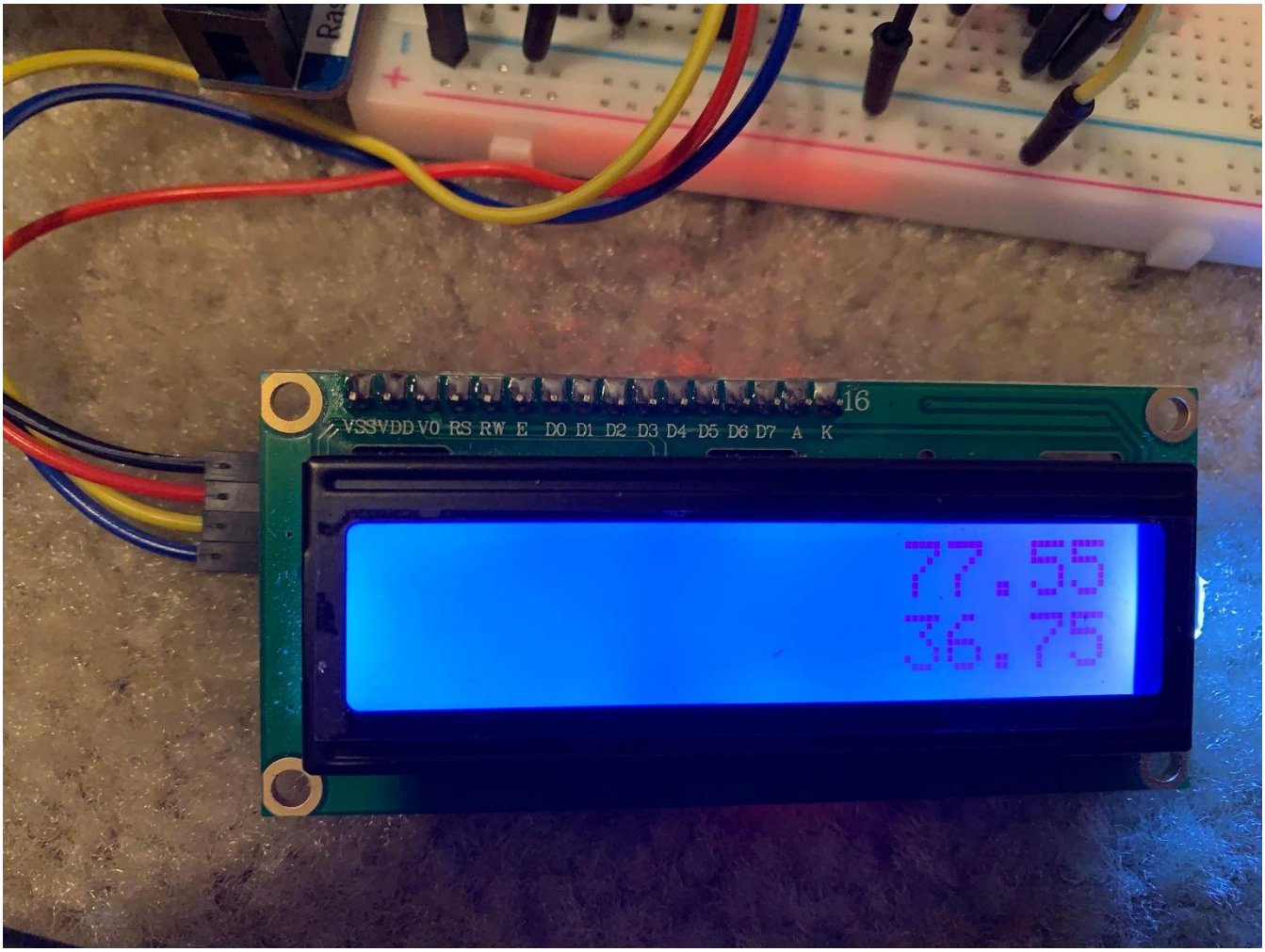
2020-6-12

1 Board and hardware setup





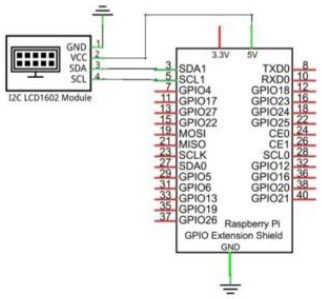
This is my complete board setup including LCD, HT sensor and motion Sensor.



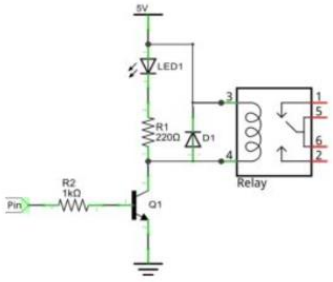
This is my pin connection and LCD connection.

Pin connections:

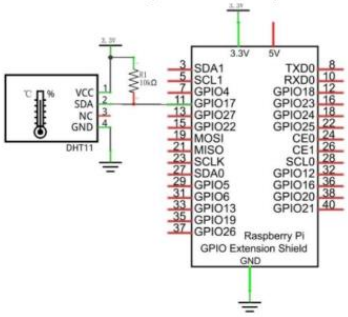
LCD



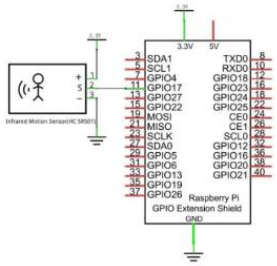
Relay (GPIO 13)

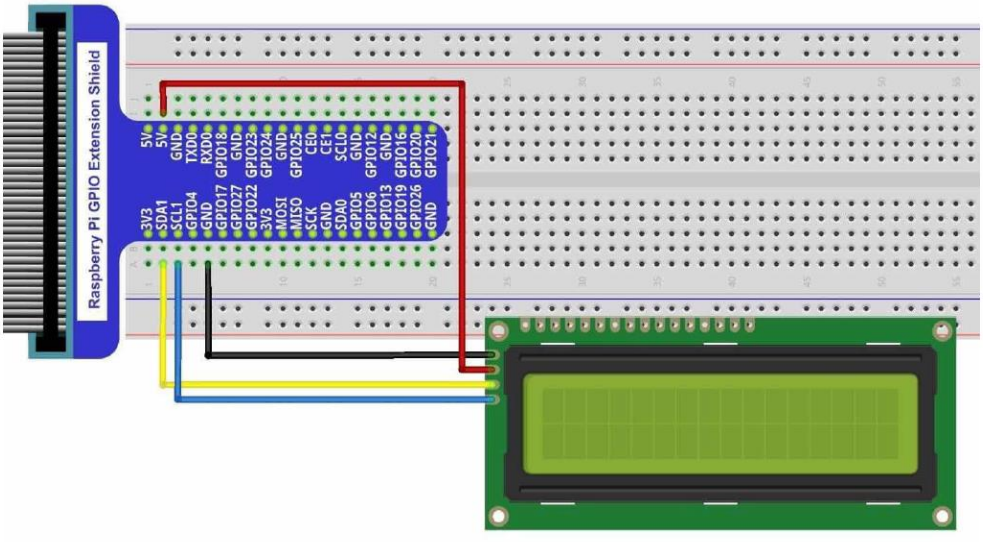


DHT (GPIO 17)

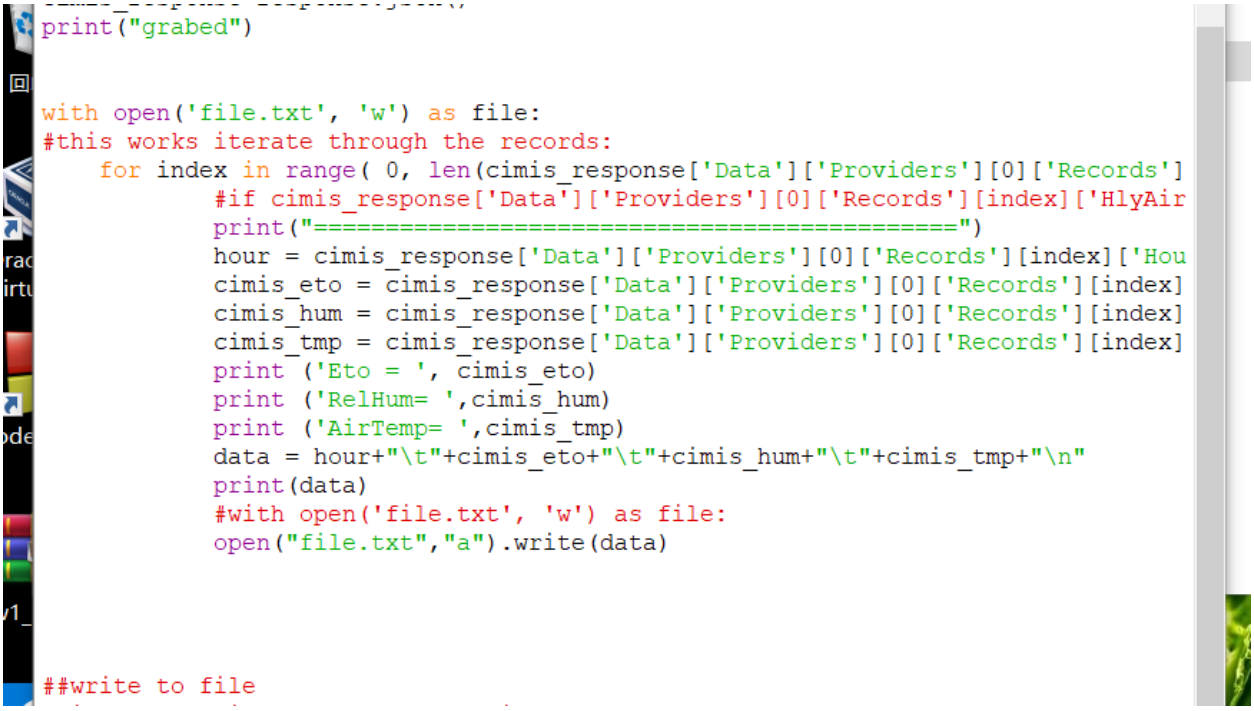


PIR (GPIO 25)



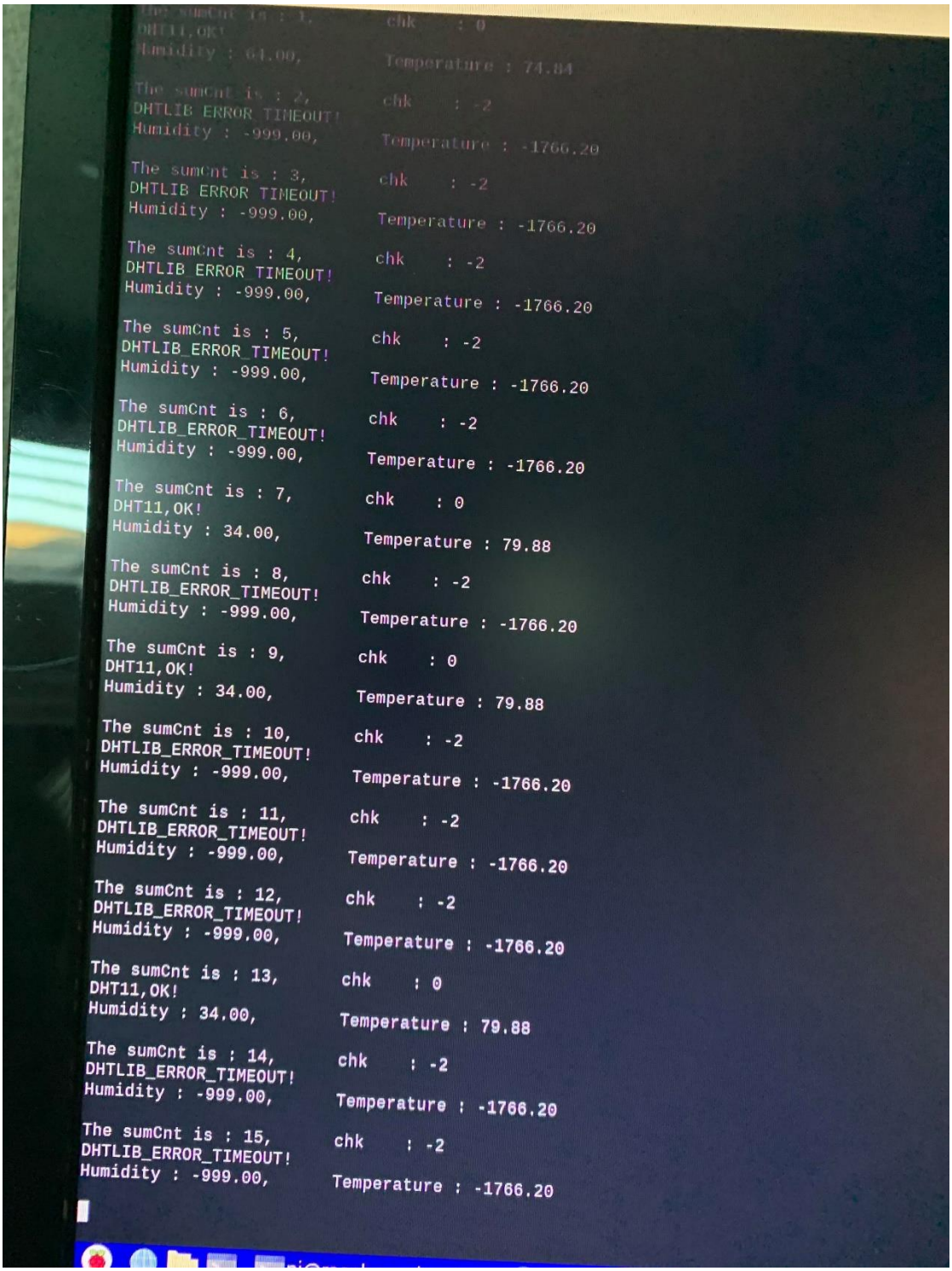


2. Steps and Code

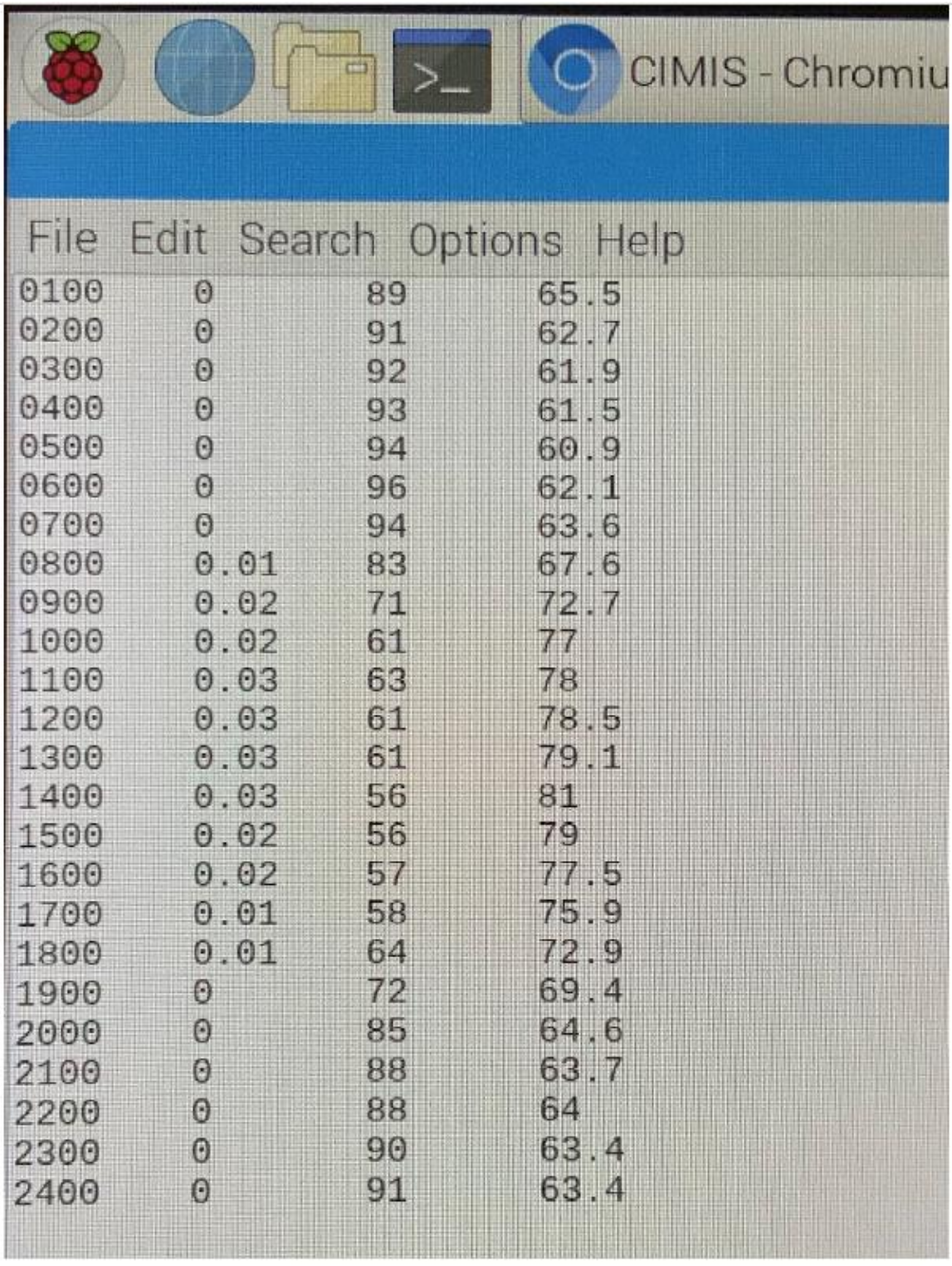


```
print("grabed")


with open('file.txt', 'w') as file:
#this works iterate through the records:
    for index in range( 0, len(cimis_response['Data']['Providers'][0]['Records']
            #if cimis_response['Data']['Providers'][0]['Records'][index]['HlyAir
            print("============================================")
            hour = cimis_response['Data']['Providers'][0]['Records'][index]['Hou
            cimis_eto = cimis_response['Data']['Providers'][0]['Records'][index]
            cimis_hum = cimis_response['Data']['Providers'][0]['Records'][index]
            cimis_tmp = cimis_response['Data']['Providers'][0]['Records'][index]
            print ('Eto = ', cimis_eto)
            print ('RelHum= ',cimis_hum)
            print ('AirTemp= ',cimis_tmp)
            data = hour+"\t"+cimis_eto+"\t"+cimis_hum+"\t"+cimis_tmp+"\n"
            print(data)
            #with open('file.txt', 'w') as file:
            open("file.txt","a").write(data)



##write to file
```

This test code is used to test CIMIS data pulling and output. The url in this test code is fixed but in the complete code the url is able to change according to the current date.



This picture shows the output of the CIMIS test case shown above. It prints eto, humidity and temperature of every hour pulled from CIMIS and writes the data into an output file. The data will also form a table shown below:


CIMIS - Chromiu
File Edit Search Options Help
0100 0 89 65.5
0200 0 91 62.7
0300 0 92 61.9
0400 0 93 61.5
0500 0 94 60.9
0600 0 96 62.1
0700 0 94 63.6
0800 0.01 83 67.6
0900 0.02 71 72.7
1000 0.02 61 77
1100 0.03 63 78
1200 0.03 61 78.5
1300 0.03 61 79.1
1400 0.03 56 81
1500 0.02 56 79
1600 0.02 57 77.5
1700 0.01 58 75.9
1800 0.01 64 72.9
1900 0 72 69.4
2000 0 85 64.6
2100 0 88 63.7
2200 0 88 64
2300 0 90 63.4
2400 0 91 63.4

```
    for cnt_hour in range (6):
        if(cimis_tmp == None):
            try:
                global data
                data=requests.get(url).json()
            except HTTPError as http_err:
                print("HTTP error occurred:",http_err)
            except Exception as err:
                print("Other error occurred:",err)
            else:
                cimis_tmp=data['Data']['Providers'][0]['Records'][now.hour-2]['H
                print("cimis success")


        for i in range(10):
            sumCnt += 1         #counting number of reading times
            chk = dht.readDHT11()     #read DHT11 and get a return value. Then d
            print ("The sumCnt is : %d, \t chk    : %d"%(sumCnt,chk))
            if (chk is dht.DHTLIB_OK):      #read DHT11 and get a return value.
                print("DHT11,OK!")
                global result_hum, result_temp, cnt_hum, cnt_temp, hourtemp_val,
                result_hum += dht.humidity
                result_temp += dht.temperature
                cnt_hum+=1
                cnt_temp+=1
            elif(chk is dht.DHTLIB_ERROR_CHECKSUM): #data check has errors
                print("DHTLIB_ERROR_CHECKSUM!!")
            elif(chk is dht.DHTLIB_ERROR_TIMEOUT):  #reading DHT times out
                print("DHTLIB_ERROR_TIMEOUT!")
            else:               #other errors
                print("Other error!")

            print("Humidity : %.2f, \t Temperature : %.2f \n"%(dht.humidity,dht.


            sleep(1)
            lcd.DisplayLeft()

            i+=1
```

```
while(True):

    now = datetime.now()
    if now.hour == 0 or now.hour == 1 or now.hour == 2:
        now_list = [str(now.year), str(now.month), str(now.day-1)]
        hour = now.hour+22
    else:
        now_list = [str(now.year), str(now.month), str(now.day)]
        hour = now.hour-2
    #now_list = [str(now.year), str(now.month), str(now.day)]
    date = now_list[0]+"-"+now_list[1]+"-"+now_list[2]
    print(date)
    print(hour)
    url = "http://et.water.ca.gov/api/data?appKey=55a1b0c5-298b-4fd5-9c42-25

    loop()
    print("Humidity : %.2f, \t Temperature : %.2f \n"%(result_hum/cnt_hum, r
    lcd.clear()
    lcd.setCursor(0,0)

     now = datetime.now()
```

These codes shown above is used to measure the humidity and temperature every 1 second and the LCD display function. There is a large chance that we get timeout error and checksum error, so we calculate the average of the valid measurements every minute. Average hourly measurements are calculated from these minutes average.

```
PCF8574_address = 0x27  # I2C address of the PCF8574 chip.
PCF8574A_address = 0x3F  # I2C address of the PCF8574A chip.

t1 = HT_sensor()
t2 = Motor()



try:
    try:
        mcp = PCF8574_GPIO(PCF8574_address)
    except:
        try:
            mcp = PCF8574_GPIO(PCF8574A_address)
        except:
            print ('I2C Address Error for HT_sensor!')
            exit(1)

    # Create LCD, passing in MCP GPIO adapter.
    t1.lcd = Adafruit_CharLCD(pin_rs=0, pin_e=2, pins_db=[4,5,6,7], GPIO=mcp)
    mcp.output(3,1)     # turn on LCD backlight
    t1.lcd.begin(16,2)     # set number of LCD lines and columns
    t1.lcd.setCursor(0,0)  # set cursor position
    t2.lcd = t1.lcd

    t1.start()
    sleep(3600)  # the first hour of the day, we don't need to irrigate
    t2.start()

    t1.join()
    t2.join()

except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the child program destroy
    t1.destroy()
    t2.destroy()
    exit(1)
```

Complete code is turned in along with video explanation. The complete version can pull CIMIS data according to current time and handle website outage; both local and CIMIS humidity and temperature data on
LCD; store the data into a file… The features are demonstrated in the video.

For the extra credit, I discuss with my other friend for the motor detection and I write several functions for the motor. However, the GPIO.setup(RelayPin, GPIO.IN) always come out with error kept stopping while testing. So I did not

systematically test for the Motor. So I guess the motor sensor works in some fashion. Here is the functionality for motor.

```
class Motor(Thread):



    def setup(self):
        #print ('Program is starting...')
        GPIO.setmode(GPIO.BOARD)       # Numbers GPIOs by physical location
        GPIO.setup(RelayPin, GPIO.OUT)   # Set ledPin's mode is output
        GPIO.setup(sensorPin, GPIO.IN)    # Set sensorPin's mode is input
        GPIO.output(RelayPin, True)

    def loop(self):
        global PrevS
        global CurrS
        global text
        global watertime

        for i in range(watertime):
            i+=1
            GPIO.output(RelayPin,False)

            detect = GPIO.input(sensorPin)
            if (detect == 1):
                CurrS = False
                if CurrS != PrevS:
```

3 Result:

The below is the output table recording 24 hours of time, local humidity, temperature, and eto data and these data pulled from CIMIS website.
The other functionalities are demonstrated in the second half of the video.

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Date	Hour	LocalHum	LocalTmp	LocalEto	CIMISHum	CIMISTmp	CIMISEto
2020-6-11	1100	56.05	73.12	0.00	72.0	63.1	0.0
2020-6-11	1200	54.22	73.08	0.00	71.0	64.2	0.0
2020-6-11	1300	55.48	73.87	0.00	74.0	63.9	0.0
2020-6-11	1400	56.22	74.45	0.00	75.0	64.2	0.0
2020-6-11	1500	55.63	75.67	0.03	73.0	66.1	0.02
2020-6-11	1600	54.96	74.99	0.02	61.0	71.9	0.02
2020-6-11	1700	55.46	74.57	0.02	63.0	70.4	0.01
2020-6-11	1800	55.03	75.09	0.01	66.0	68.3	0.01
2020-6-11	1900	55.43	75.49	0.01	73.0	64.8	0.0
2020-6-11	2000	56.52	75.86	0.00	78.0	63.1	0.0
2020-6-11	2100	57.35	76.43	0.00	79.0	62.9	0.0
2020-6-11	2200	56.49	75.13	0.00	79.0	63.1	0.0
2020-6-11	2300	54.00	76.24	0.00	78.0	63.1	0.0
2020-6-11	2400	54.25	75.97	0.00	78.0	62.8	0.0
2020-6-12	0100	55.03	75.70	0.00	79.0	62.2	0.0
2020-6-12	0200	55.03	75.89	0.00	80.0	61.7	0.0
2020-6-12	0300	55.03	75.69	0.00	81.0	61.4	0.0
2020-6-12	0400	55.03	76.89	0.00	84.0	61.5	0.0
2020-6-12	0500	55.03	75.12	0.00	80.0	62.9	0.0
2020-6-12	0600	55.03	75.75	0.00	79.0	62.2	0.0
2020-6-12	0700	55.03	75.49	0.00	80.0	61.7	0.0
2020-6-12	0800	55.03	75.69	0.00	81.0	61.4	0.0
2020-6-12	0900	55.03	76.89	0.00	84.0	61.5	0.0
2020-6-12	1000	54.42	75.12	0.00	80.0	62.9	0.0