



Senior Design Project

ECE 595

Weekly Journal

Shaima Hussien

Table of Contents

| | |
|---------------------------------------|----------|
| <i>Week (4-5) Feb 10th:</i> | 3 |
| <i>Goals of the week:</i> | 3 |
| <i>General Notes:</i> | 3 |
| <i>Results and Conclusions:</i> | 5 |
| <i>Next Step:</i> | 5 |
| <i>Week (6-7) Feb 28th:</i> | 6 |
| <i>Goals of the week:</i> | 6 |
| <i>General Notes:</i> | 6 |
| <i>Results and Conclusions:</i> | 9 |
| <i>Next Step:</i> | 9 |

Weekly Journal:**Week (4-5) Feb 10th:****Goals of the week:**

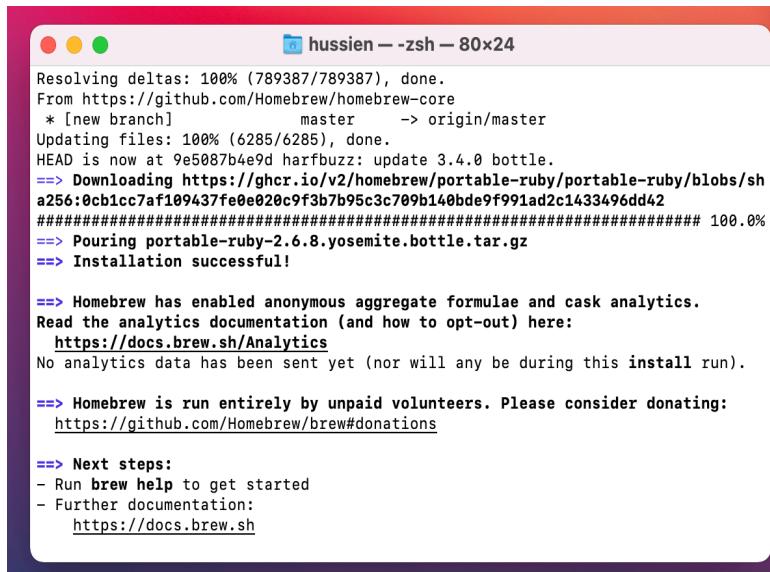
- Learn about open CV that will be used for face detection.
- install open CV
- Calculate how much power is needed to run the product.
- Try to connect the thermal camera to raspberry pi.

General Notes:

Open CV or as it known as computer vision is an open-source library for the computer vision machine learning and image processing that it can be used to identify different object and even faces.

First step is install Open cv on my computer:

1- install Home brew which is open-source software package management system that simplifies the installation of software on Apple's operating system, macOS



```

hussien -- zsh -- 80x24

Resolving deltas: 100% (789387/789387), done.
From https://github.com/Homebrew/homebrew-core
 * [new branch] master -> origin/master
Updating files: 100% (6285/6285), done.
HEAD is now at 9e5087b4e9d harfbuzz: update 3.4.0 bottle.
=> Downloading https://ghcr.io/v2/homebrew/portable-ruby/portable-ruby/blobs/sha256:0cb1cc7af109437fe0e020c9f3b7b95c3c709b140bde9f991ad2c1433496dd42
#####
=> Pouring portable-ruby-2.6.8.yosemite.bottle.tar.gz
=> Installation successful!

=> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

=> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

=> Next steps:
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh

```

2- Install open cv

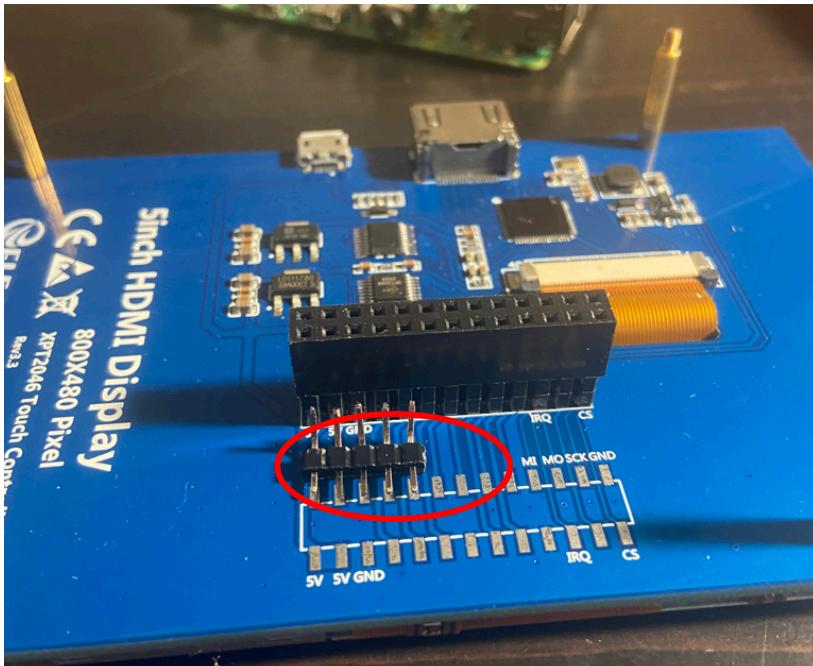
```

hussien -- zsh -- 80x24
--> Pouring netcdf--4.8.1.big_sur.bottle.tar.gz
  /usr/local/Cellar/netcdf/4.8.1: 89 files, 6.3MB
--> Installing opencv dependency: pugixml
--> Pouring pugixml--1.12.big_sur.bottle.tar.gz
  /usr/local/Cellar/pugixml/1.12: 12 files, 268.5KB
--> Installing opencv dependency: qt@5
--> Pouring qt@5--5.15.2.1.big_sur.bottle.tar.gz
  /usr/local/Cellar/qt@5/5.15.2.1: 10,688 files, 365.9MB
--> Installing opencv dependency: pyqt@5
--> Pouring pyqt@5--5.15.6.big_sur.bottle.tar.gz
  /usr/local/Cellar/pyqt@5/5.15.6: 1,327 files, 31.6MB
--> Installing opencv dependency: utf8cpp
--> Pouring utf8cpp--3.2.1.all.bottle.tar.gz
  /usr/local/Cellar/utf8cpp/3.2.1: 11 files, 103.8KB
--> Installing opencv dependency: vtk
--> Pouring vtk--9.1.0.1.big_sur.bottle.tar.gz
  /usr/local/Cellar/vtk/9.1.0.1: 3,807 files, 157.7MB
--> Installing opencv
--> Pouring opencv--4.5.4_4.big_sur.bottle.tar.gz
  /usr/local/Cellar/opencv/4.5.4_4: 852 files, 237.5MB
--> Running brew cleanup opencv...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
hussien@Shaimas-MacBook-Air ~ %

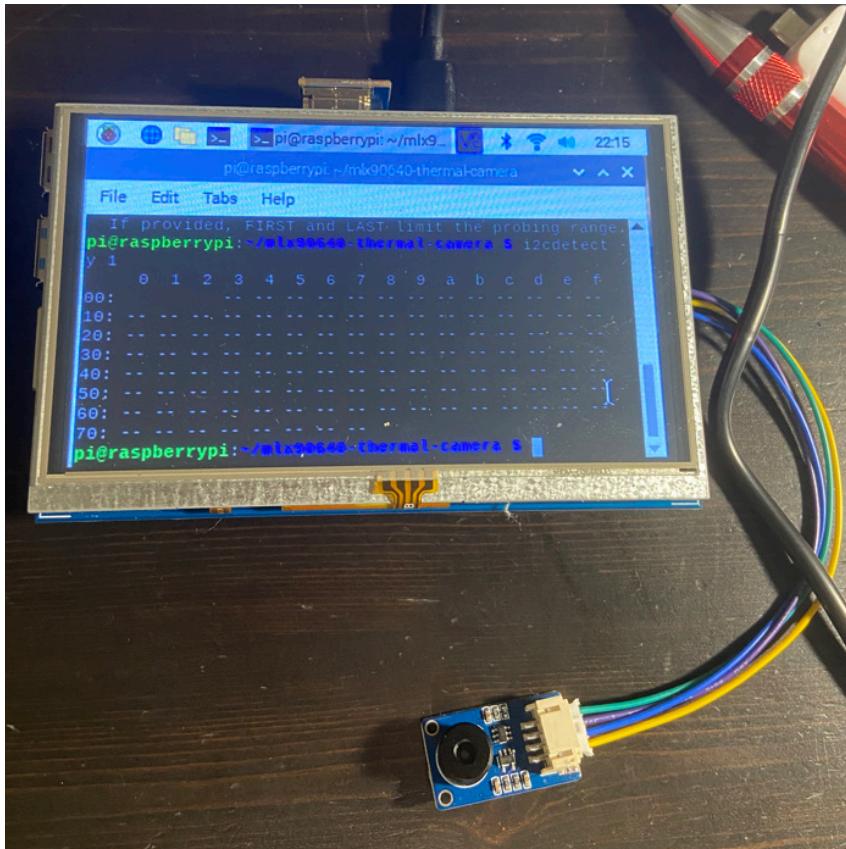
```

I am trying to connect the thermal camera on raspberry pi to measure the temperatures the issue I have is my raspberry pi is not picking the address of the thermal camera.

1- I sorted 5 pins on HDMI screen to plug the thermal camera.



2- Attached the raspberry 4 on HDMI by using HDMI bridge and the thermal camera.



Results and Conclusions:

Raspberry pi is not picking the right address for the thermal camera

Next Step:

- Continue working on the raspberry pi and thermal camera.
- Set up Open CV.

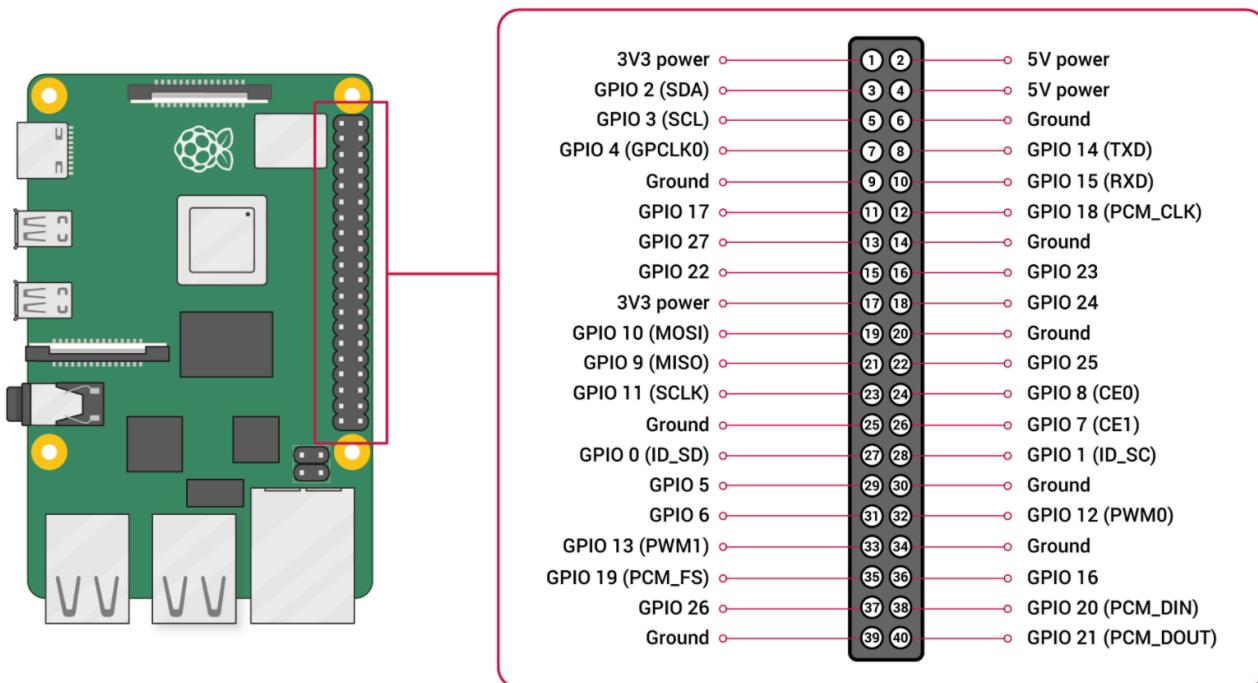
Week (6-7) Feb 28th:**Goals of the week:**

- Connect the thermal camera to the Raspberry pi.
- Figure out the right address for the Pi.
- Install the needed libraries on python to run the thermal camera.
- Understand what each pin of the pi is

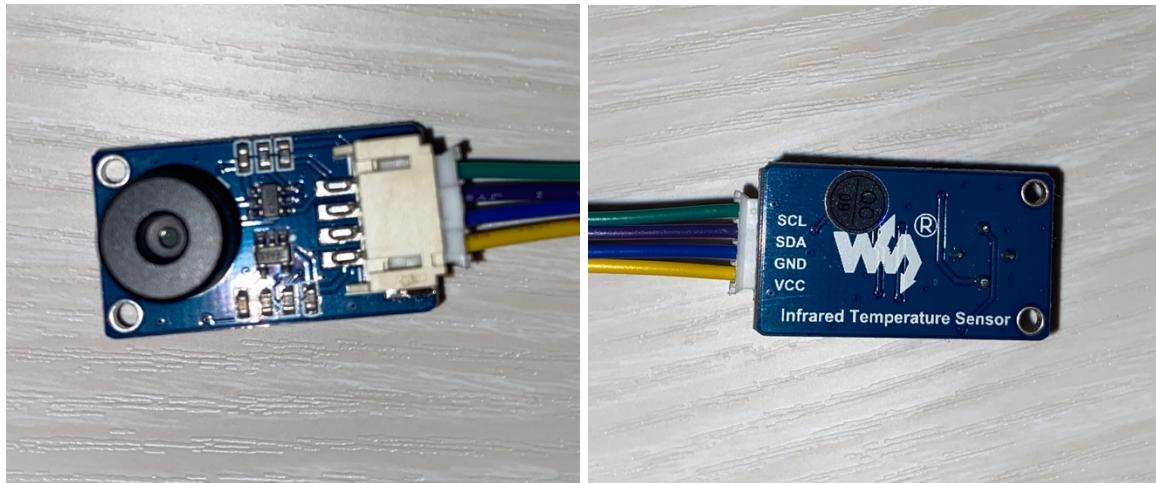
General Notes:

Some of the challenges I had to deal with was to connect the thermal camera to the pi correctly and make sure that it is picking the right address, the root cause of this problem was not connecting the pi the 5-inch HDMI lcd correctly.

to expand my knowledge about our product, I needed to understand what each pin is corresponded to in the pi.



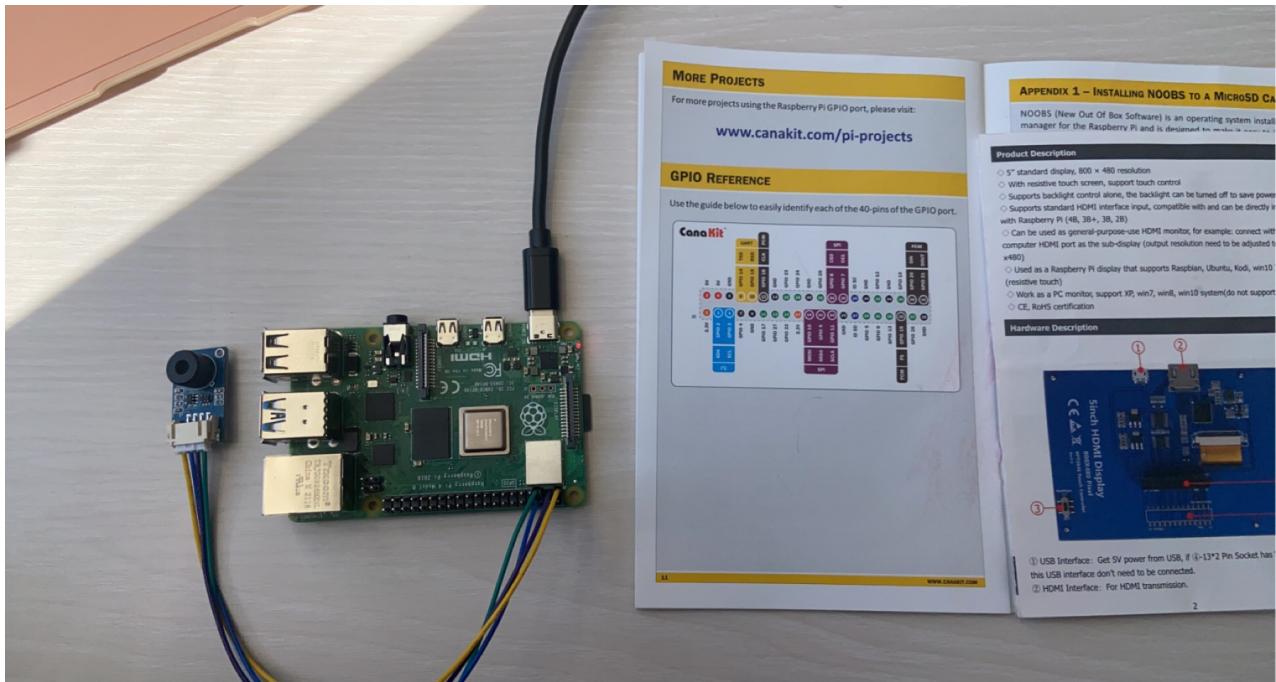
Source → <https://www.raspberrypi.com/documentation/computers/os.html>



Integration between Raspberry Pi and Adafruit MLX90640:

The pin out connections from the Adafruit MLX90640 are straightforward. For this project only 4 pins will be used:

- **Vin** → It is the power pin. The Adafruit sensor uses 3.3V it will be connected to corresponding 3.3V pin which is pin 1 on the Raspberry Pi
- **GND** → this is common ground for power and logic. I connect it to ground pin (pin 9) on the Pi4. There are more than 1 ground pin on the Pi4.
- **SCL** → this is the I2C clock pin, and it will be connected it to the corresponding SCL pin on the Pi4 which is pin 5.
- **SDA** → this is the I2C data pin, and it will be connected it to the corresponding SDA pin on the Pi4, which is pin 3.

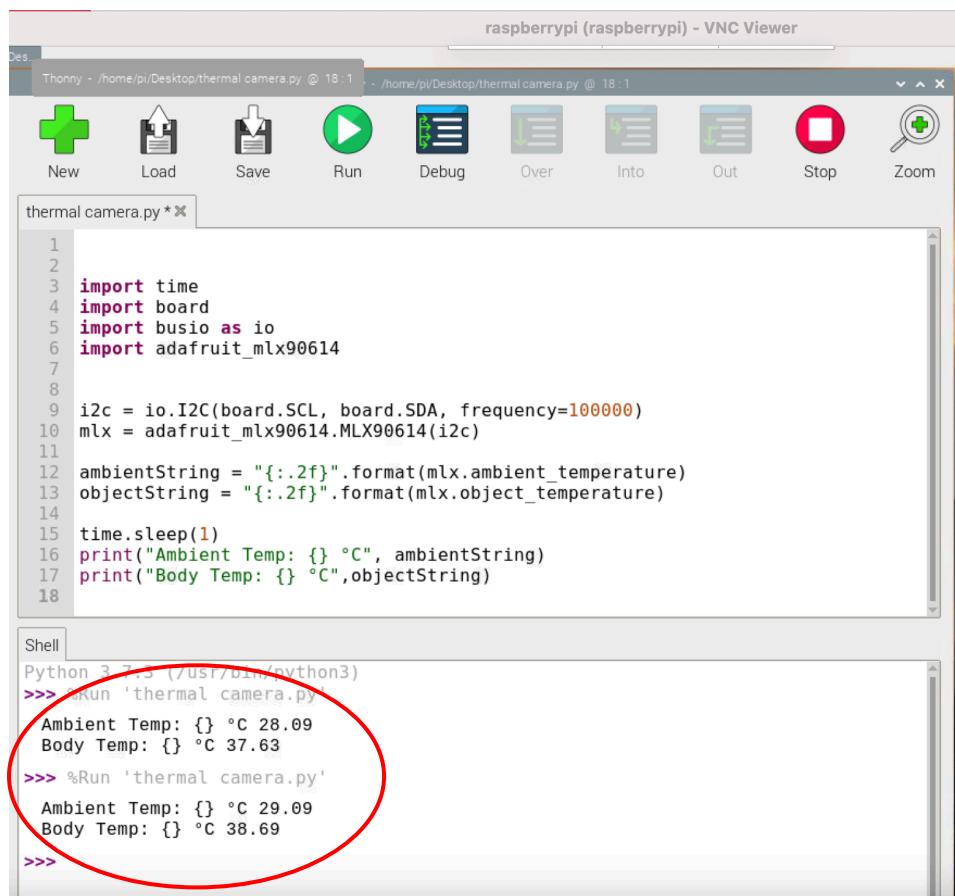


after connect the thermal camera to the right pins on the pi and install the i2c library .. finally I can see that the it getting an address.

```

pi@raspberrypi:~/mlx90640-thermal-camera $ i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -
10: -
20: -
30: -
40: -
50: -
60: -
70: -
pi@raspberrypi:~/mlx90640-thermal-camera $
  
```

now it's the time to run the code. <https://tutorial.cytron.io/2020/12/09/send-data-to-firebase-using-raspberry-pi/>



The screenshot shows the Thonny IDE interface on a Raspberry Pi VNC viewer. The top menu bar says "raspberrypi (raspberrypi) - VNC Viewer". The main window has a toolbar with icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, and Zoom. Below the toolbar is a code editor window titled "thermal camera.py *". The code in the editor is:

```

1
2
3 import time
4 import board
5 import busio as io
6 import adafruit_mlx90614
7
8
9 i2c = io.I2C(board.SCL, board.SDA, frequency=100000)
10 mlx = adafruit_mlx90614.MLX90614(i2c)
11
12 ambientString = "{:.2f}".format(mlx.ambient_temperature)
13 objectString = "{:.2f}".format(mlx.object_temperature)
14
15 time.sleep(1)
16 print("Ambient Temp: {} °C", ambientString)
17 print("Body Temp: {} °C", objectString)
18

```

Below the code editor is a "Shell" window showing the execution of the script. A red circle highlights the second run command and its output. The output is:

```

Shell
Python 3.7.3 (v3.7.3:ef537d8, Mar 29 2019, 13:14:21) [GCC 7.3.0] on linux
>>> %Run 'thermal camera.py'
Ambient Temp: {} °C 28.09
Body Temp: {} °C 37.63
>>> %Run 'thermal camera.py'
Ambient Temp: {} °C 29.09
Body Temp: {} °C 38.69
>>>

```

Results and Conclusions:

So now the thermal camera can detect a body temperature and ambient temperature.

Next Step:

- Continue working on open cv.
- Develop some python skills.
- Work on the calculations and calculate how much power needed to run the whole unit.

Hussien(V625S887)