

Spectrometer Assembly

Connect raspberry pi to the screen:

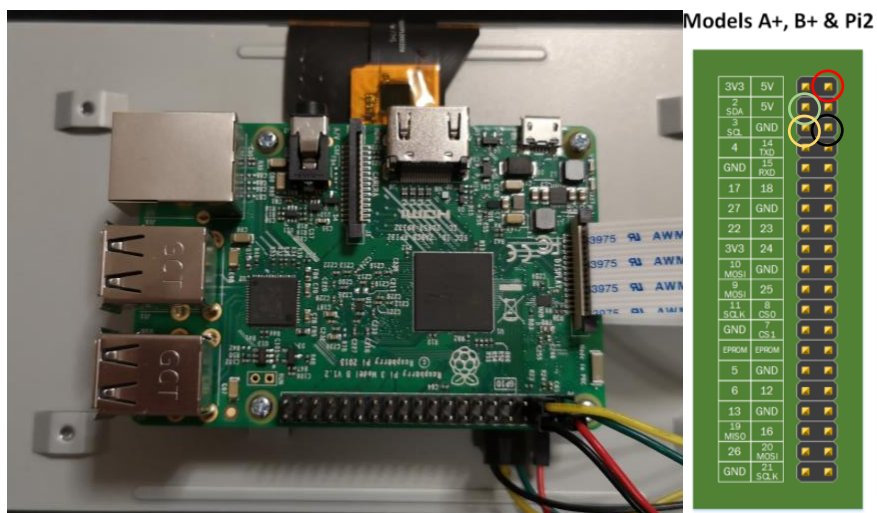
Connect pins on the touchscreen using the jumper cables, and the connector using the white ribbon cable. The blue strip of the ribbon cable should face the black tab on the connector. Be gentle with the ribbon cable and the tab as they can be quite delicate.



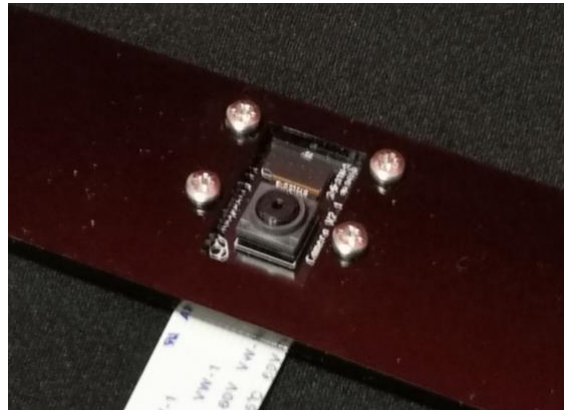
Insert the microSD card containing the operating system into the slot on the raspberry pi (located below the ribbon tab on the short edge).

Screw the raspberry pi onto the standoffs on the touchscreen.

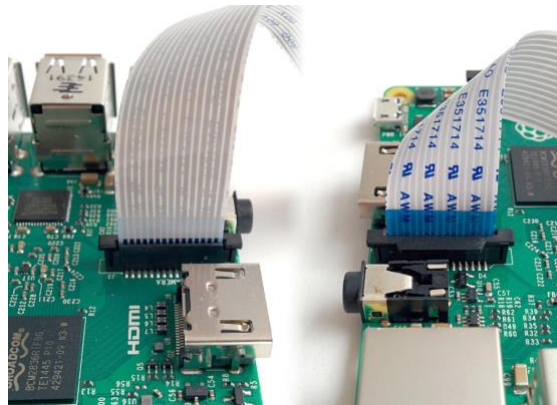
Plug the jumper cables into the corresponding pins on the raspberry pi.



Attach the camera to the mount using four 2.5 mm screws. If you have a new camera you will need to cut a thread into the camera mounting holes using a 2.5x0.45 mm tap.



Connect the raspberry pi NoIR camera using the white ribbon cable.



Connect your keyboard.

Place the camera at the viewport of the spectrometer.

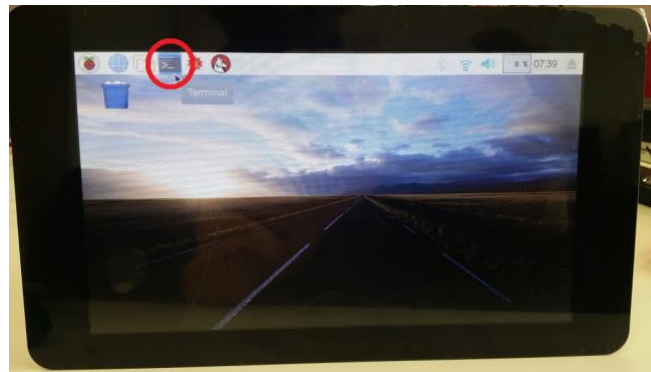


ICTP Winter College on
Applications of Optics and Photonics in Food Science

Connect the raspberry pi to power to switch the device on. The device works best if both the microUSB on the raspberry pi and the screen are plugged into a power supply, but will work with the raspberry pi connected to power only.



To use the raspberry pi to acquire a spectrum, open the terminal.



You can stream the output from the camera as a video with the following command

```
raspivid -t 5000
```

The number after -t can be adjusted to stream for different length of time (in ms: so 5000 = 5 s.) If the time you entered is too long use "CTRL" + "C" to cancel.

The python scripts for running the spectroscope are located in /home/pi/Documents/ictp and they can be located by typing the following into the terminal and then pressing enter.

```
cd /home/pi/Documents/ictp
```

If you want to see the files in a directory use the "ls" command

```
ls
```

Example: Capturing A Spectrum and Running PCA Script

The first script is an example captures an image and plots RGB intensity as a function of pixel position, producing a simple spectrum. To run it, simply type the following and press enter.

```
python picam_test.py
```

The second script is an example of a PCA analysis. To run this, you need to have 65 spectra files named in the following format "HH-MM-SS.csv" (H=hours, M=minutes, S=seconds). You can modify the scripts to change the number of spectra required, or the format the data is required to be in. The following section of the script controls how the file is read:

```
for file in os.listdir():  
    if ".csv" in file and "-" in file:#filter out other files  
        # print(file)  
        data = pd.read_csv(file).values  
        X.append(data[:,0])
```

Changing red parts above to blue below

```
for file in os.listdir("./"):  
    if ".csv" in file and ":" in file:#filter out other files  
        # print(file)  
        data = pd.read_csv(file).values  
        X.append(data[:,0])
```

will point the script to the present working directory to look for csv files in the format "HH:MM:SS".

Change the number of components in red to be less than the number of spectra you are analysing before running the script:

```
print("X_SHAPE:",X.shape)  
pca = PCA(n_components=65)  
pca.fit(X)
```

You can run this in the same way as the other script by typing

```
python pca_attempt3.py
```

ICTP Winter College on
Applications of Optics and Photonics in Food Science

Or if you want to see the script use

```
cat picam_test.py
```

```
cat pca_attempt3.py
```

The scripts are only provided as an example of what the device is capable of.

Running `picam_test.py` creates .csv files in the present working directory

```
nb-20-32:ictp andywang$ pwd
/Users/andywang/Desktop/ICTP/data/ictp
nb-20-32:ictp andywang$ ls
14-04-33.csv          14-57-29.csv          loadings.csv
14-05-04.csv          15-00-31.csv          pca_attempt3.py
14-05-46.csv          15-05-59.csv          picam_test.py
14-06-05.csv          PC1 vs PC2 Scatter.png scores.csv
14-12-54.csv          PC1>Loading_Plot.png  scratch
14-17-16.csv          PC2>Loading_Plot.png  try1
14-18-01.csv          PC3>Loading_Plot.png  yellow-red
14-52-38.csv          emission              yrb
nb-20-32:ictp andywang$
```

With suitable modifications to `pca_attempt3.py` you can run it to find the principal components of the data in this directory.

```
nb-20-32:ictp andywang$ pwd
/Users/andywang/Desktop/ICTP/data/ictp
nb-20-32:ictp andywang$ ls
14-04-33.csv          14-57-29.csv          loadings.csv
14-05-04.csv          15-00-31.csv          pca_attempt3.py
14-05-46.csv          15-05-59.csv          picam_test.py
14-06-05.csv          PC1 vs PC2 Scatter.png scores.csv
14-12-54.csv          PC1>Loading_Plot.png  scratch
14-17-16.csv          PC2>Loading_Plot.png  try1
14-18-01.csv          PC3>Loading_Plot.png  yellow-red
14-52-38.csv          emission              yrb
nb-20-32:ictp andywang$ python pca_attempt3.py
('X_SHAPE:', (11, 179))
[0.60910138 0.24665864 0.07166912]
('Score Shape: ', (11, 3))
nb-20-32:ictp andywang$
```

You can see that the script prints

```
('X_SHAPE:', (11, 179))
```

```
[0.60910138 0.24665864 0.07166912]
```

```
('Score Shape: ', (11, 3))
```

The shape of the data matrix in the first line; the % variance explained by each principal component in the second line; and the shape of the score matrix in the third line.

It also produces two files containing the loadings and scores of the principal components:

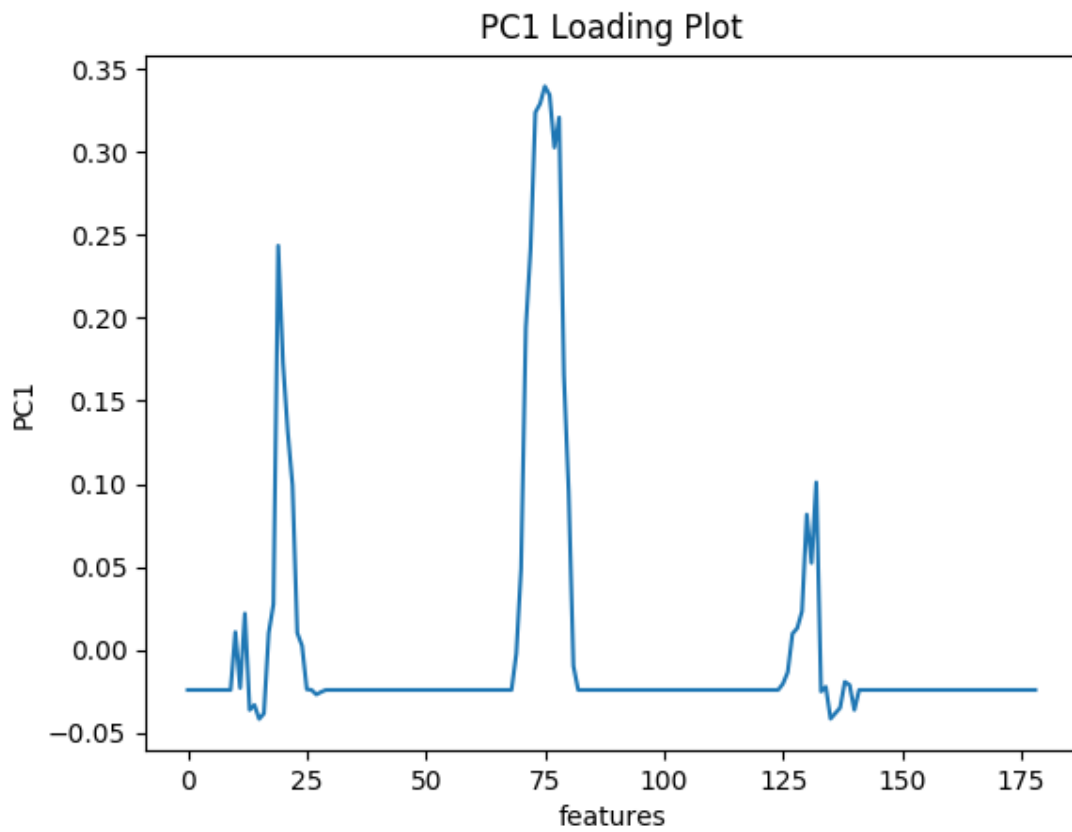
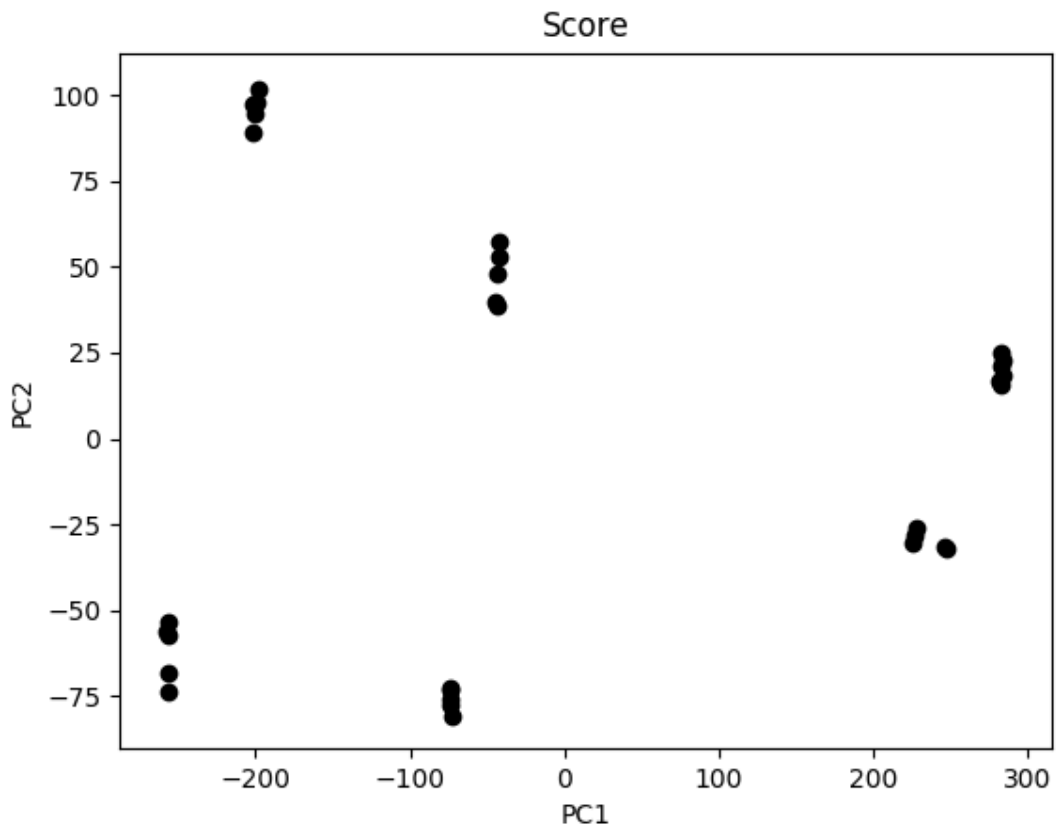


Xindi Wang
Email: xwan232@aucklanduni.ac.nz

2019-2-19

loadings.csv
scores.csv

And two plots



Troubleshooting

Required libraries:

The following libraries are required to run the example scripts

- Scipy
- Gi-cairo
- Pandas
- Sklearn (scikit-learn)
- Matplotlib

You can install libraries (if your raspberry pi is connected to the internet) using the following command

```
sudo apt-get install python-[library name]
```

if prompted for a password, **the password is “ictp2019”**.

If picam_test.py fails:

Use

```
raspivid -t 5000
```

For a 5s stream from the camera instead of plotting the curve.

If trying to run the camera produces a whole bunch of random errors:

The camera may have become loose or unplugged. Reinsert the tab, make sure to lock it down. Also, check that the camera is enabled by entering the configuration menu using the following command:

```
sudo raspi-config
```

Then navigate to item 5 “Interfaces” to find the option for enabling the camera.

Do not disconnect the ribbon cable for the camera while power is connected. This can damage the camera board and break the camera.