

Course work #1

Introduction of Raspberry pi

By

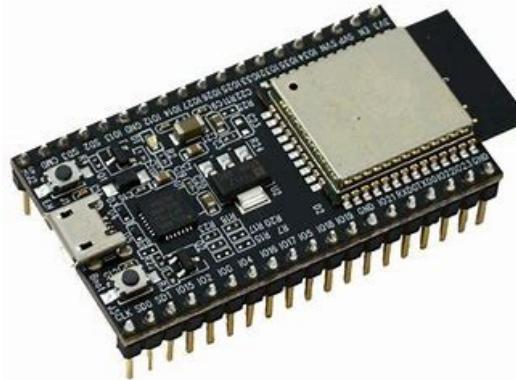
Prof. Sakaguchi and Prof. Kiyasu

Raspberry pi & Arduino (ESP32)

	Arduino (ESP32)	Raspberry pi
Operating System	None	Linux (Raspbian)
Programing Lang.	IDE	Python, C, bash
Network	None (Wifi, BT)	Wi-Fi, Ethernet, BT



Arduino Uno



ESP32



Raspberry pi

Agenda

1. Install Raspbian OS (32bit Full)
2. LED flashing
3. Servo motor
4. Camera module
5. Cron daemon
6. OpenCV and Face recognition Demo

Handout



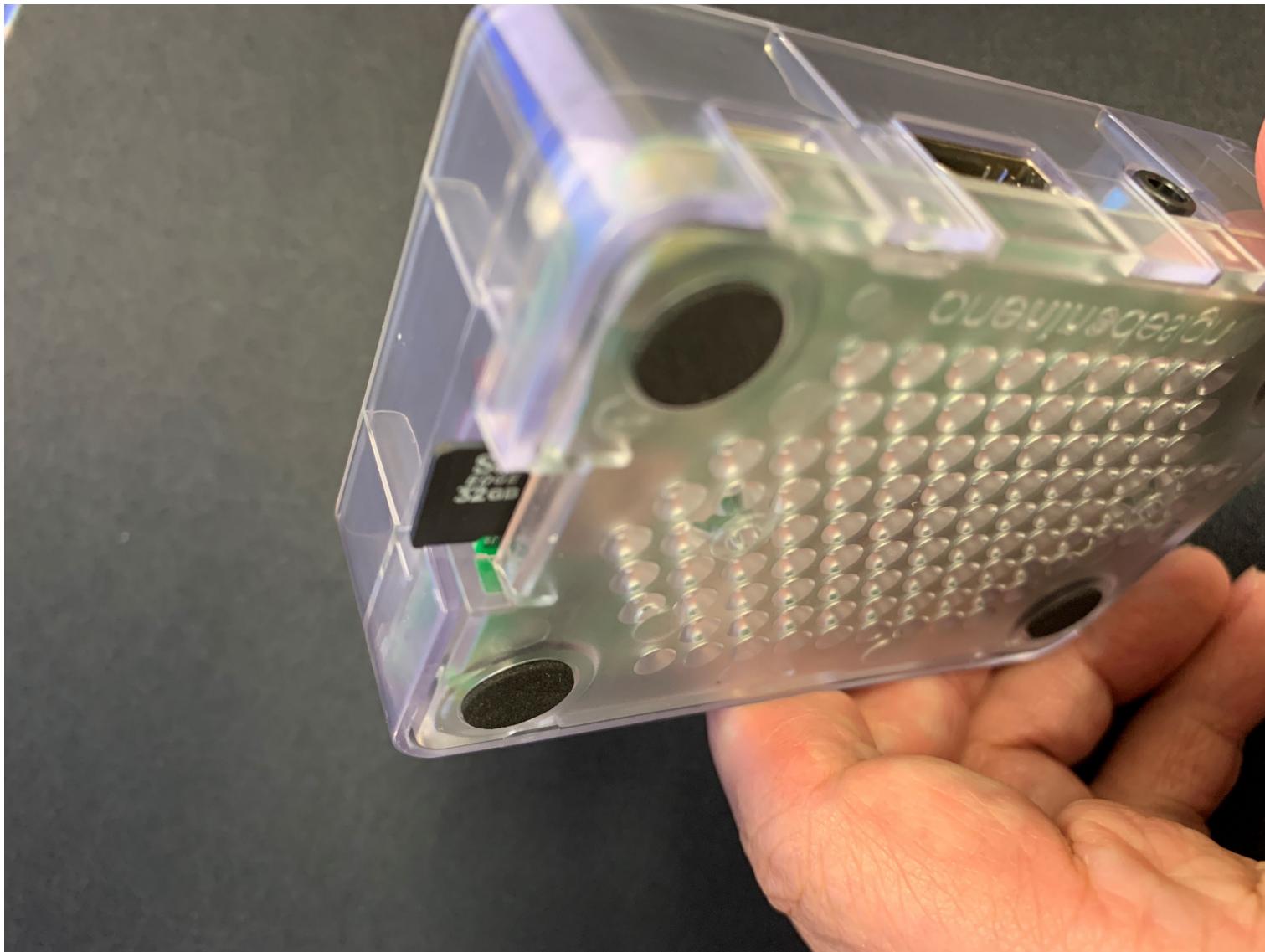
Handout



Insert the SD card



Insert the SD card



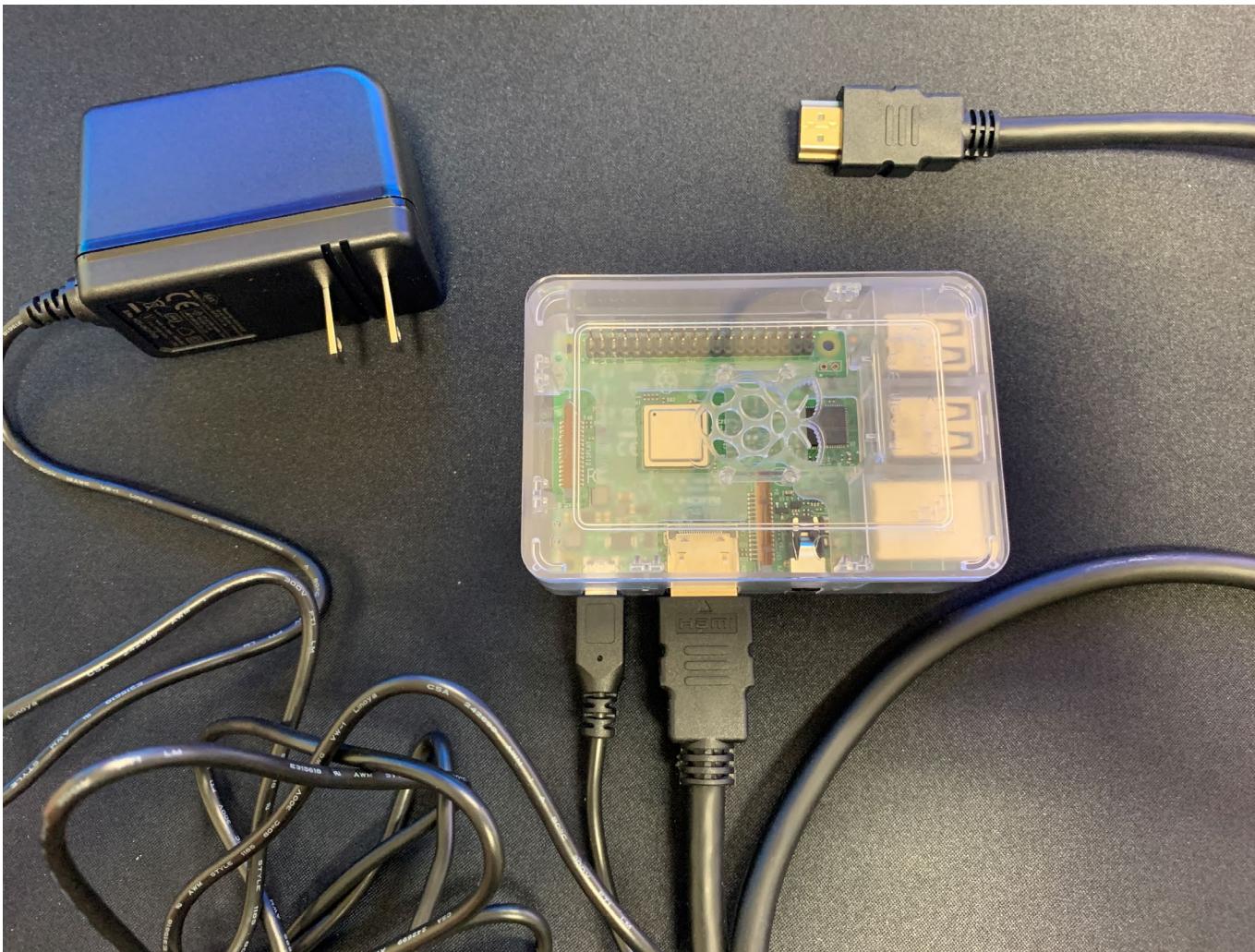
AC adapter



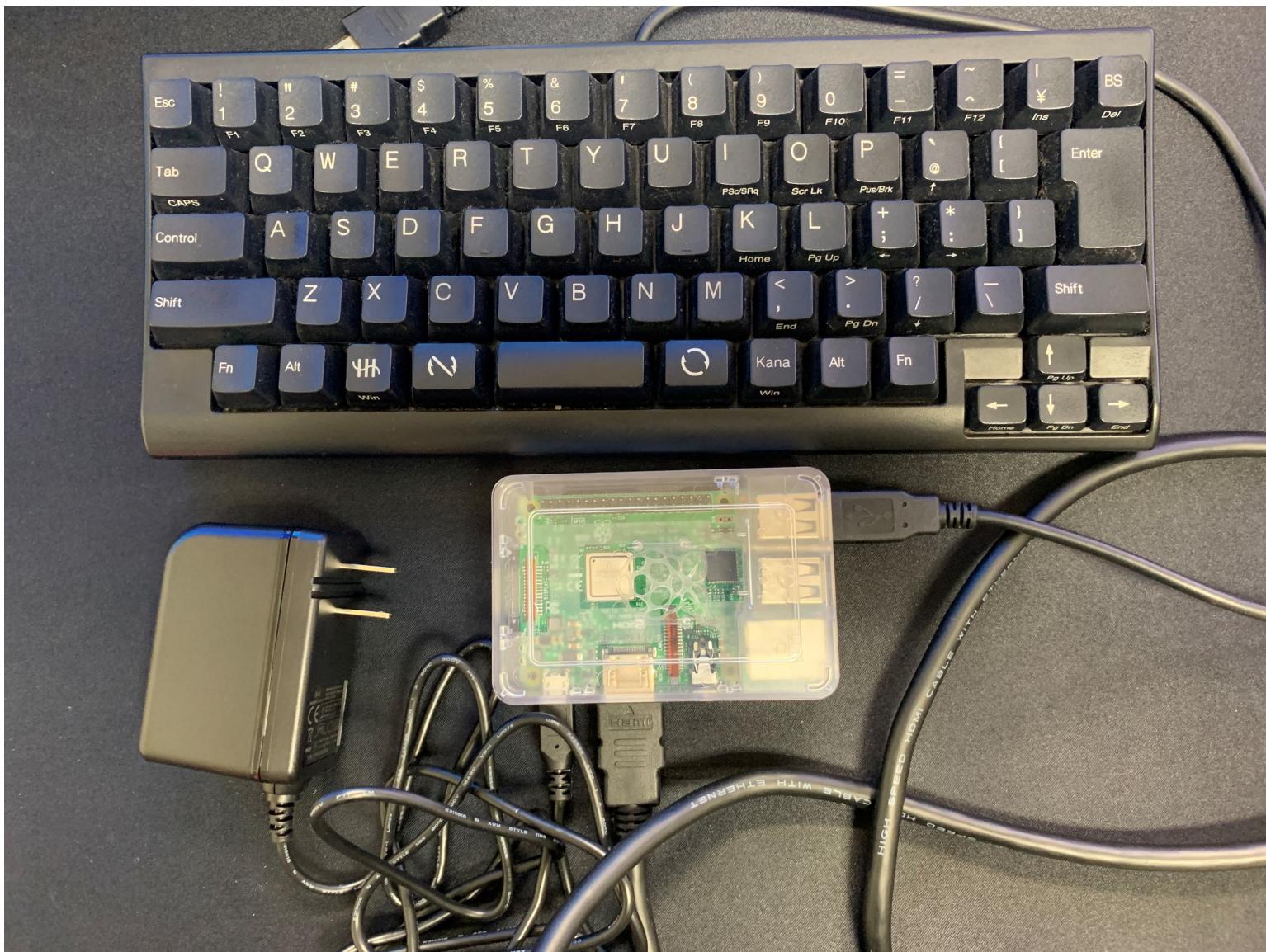
AC adapter



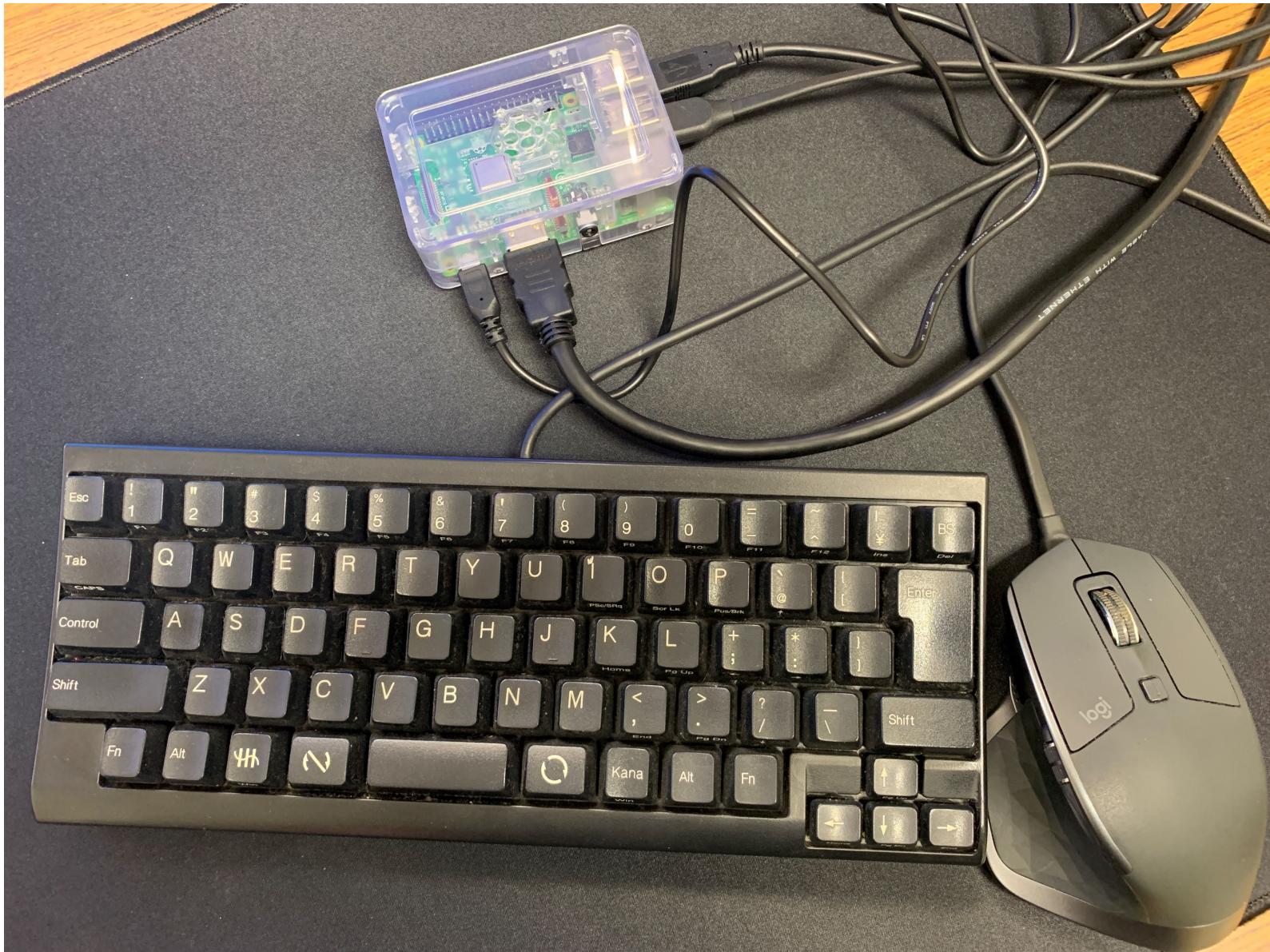
HDMI cable



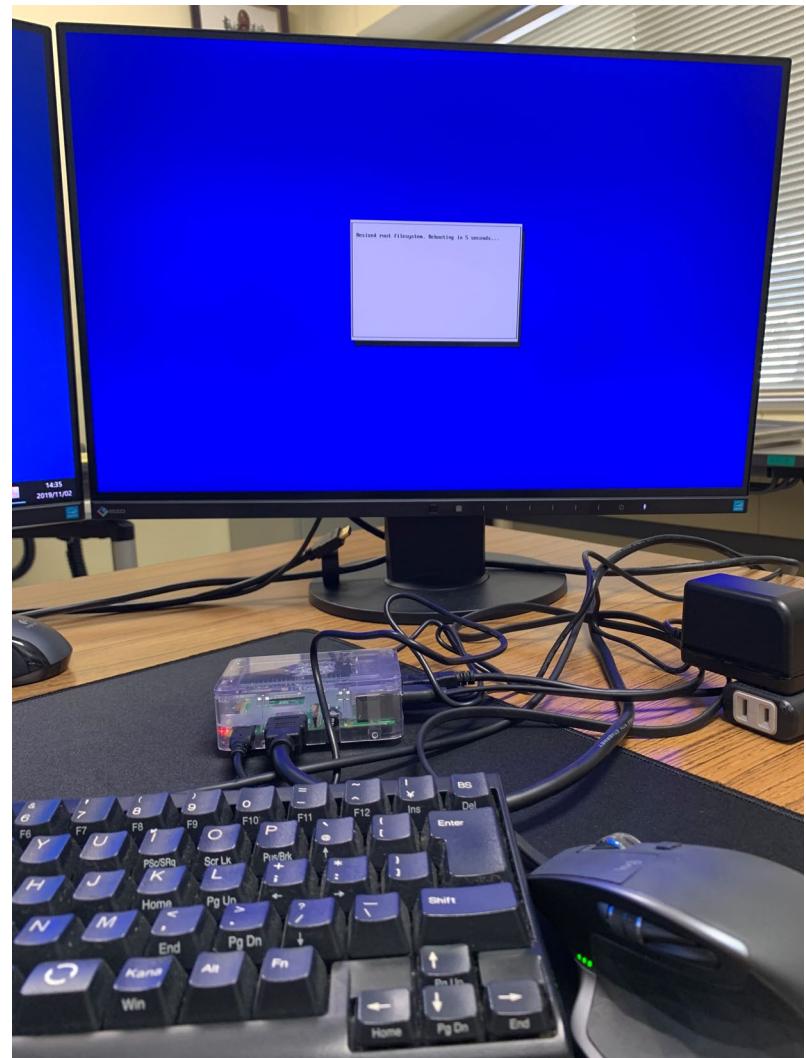
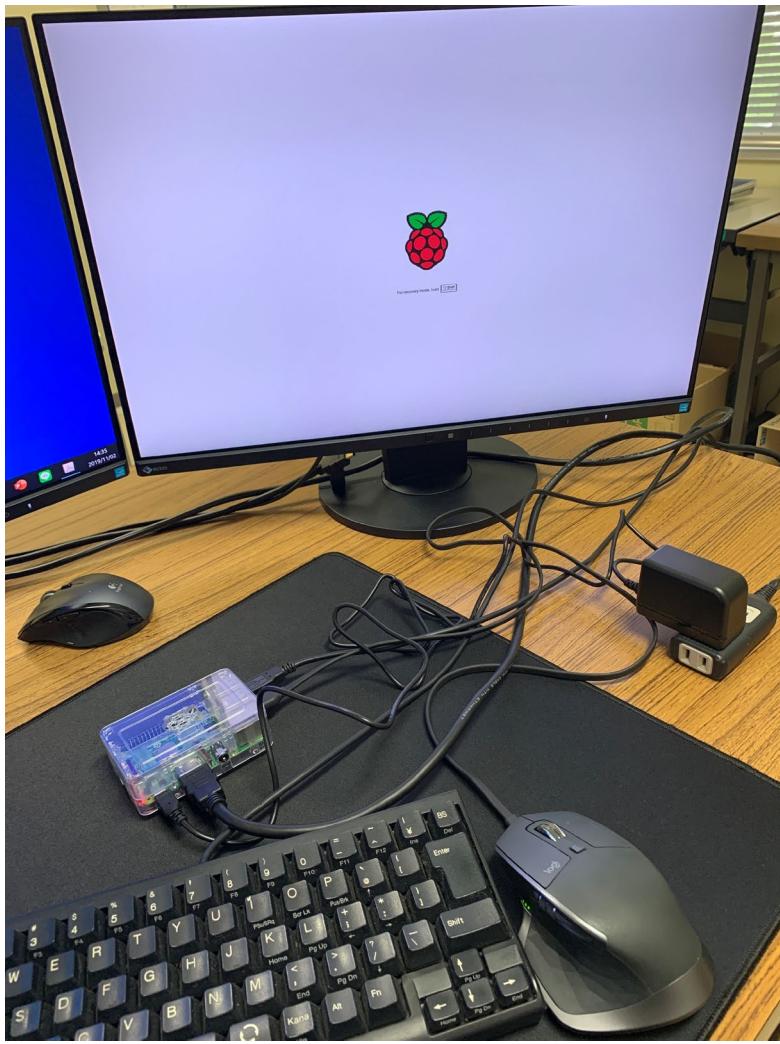
USB Mouse and Keyboard



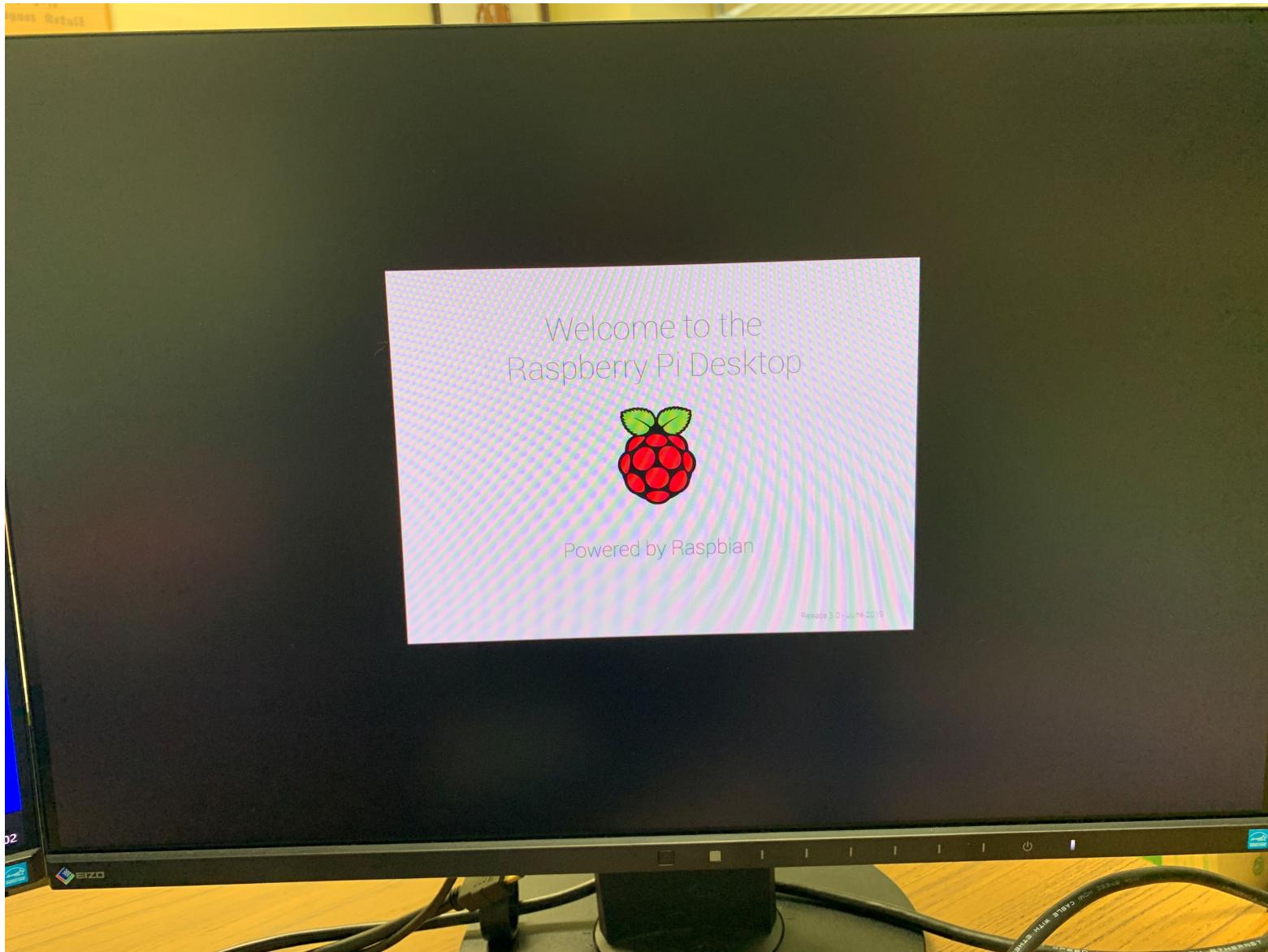
USB Mouse and Keyboard



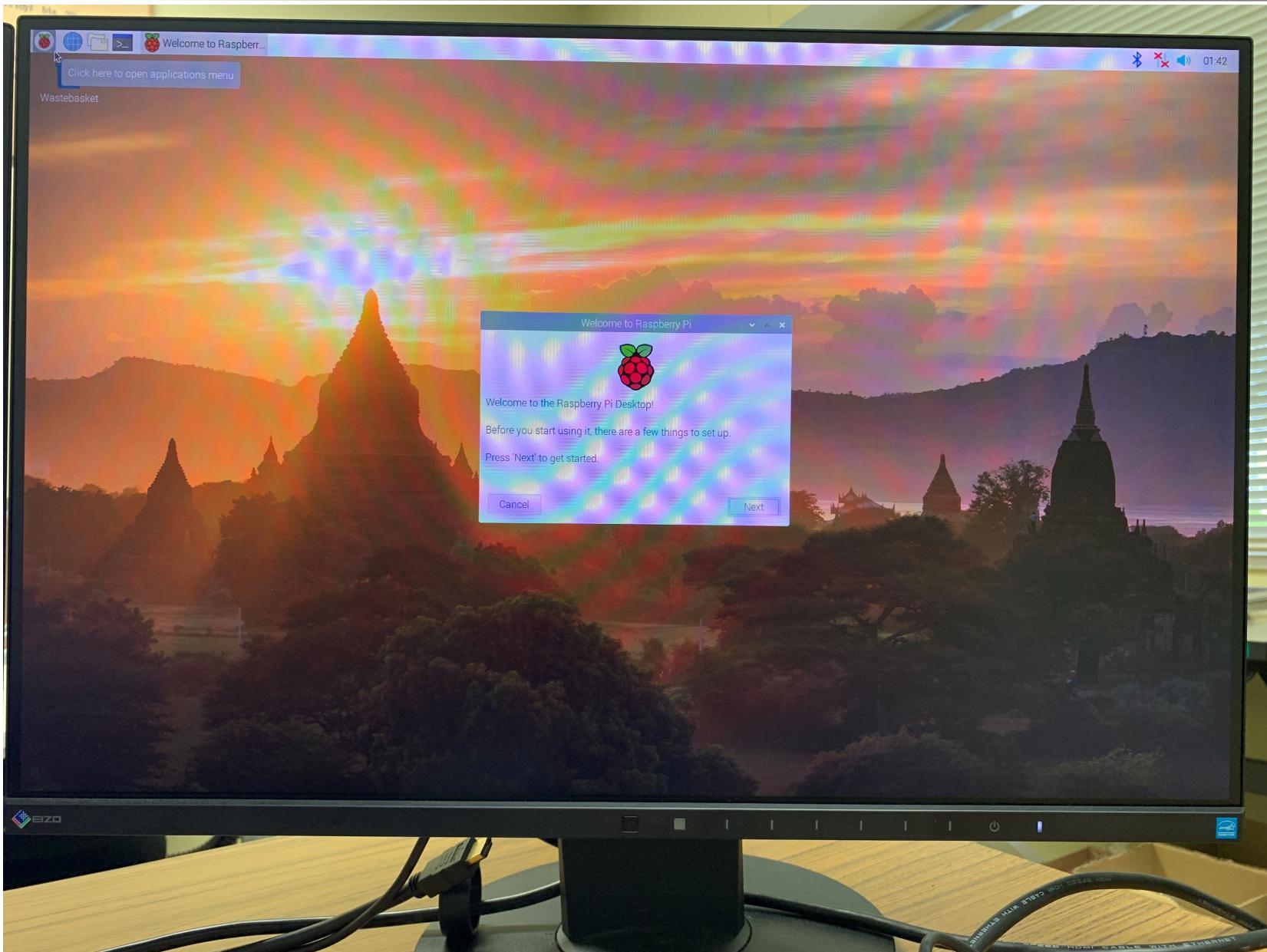
Power On



Boot up



DESKTOP



VNC Connection



You can play your R.P. with your laptop.

Install VNC viewer (Free)

Download VNC Viewer | VNC® C X +

https://www.realvnc.com/en/connect/download/viewer/

REALVNC Products Company Contact us EN Sign in

VNC CONNECT Discover Pricing Download Support Partners Try Buy

VNC® Connect consists of VNC® Viewer and VNC® Server

Download VNC® Viewer to the device you want to control from, below. Make sure you've [installed VNC® Server](#) on the computer you want to control.

 Windows  macOS  Linux  Raspberry Pi  iOS  Android  Solaris  HP-UX  AIX

[Download VNC Viewer](#)

SHA-256: 6d2637db19c0c57d9375dddc15d24dd72e461a45a47fd5c017d529b8b7135599

EXE x86/x64 ▾

Looking for VNC® Server?

ヘルプ

SSID : dojo

Password : dojodojo

DHCP : 192.168.110.2~

if you have an error VNC viewer 「Cannot currently show the desktop」
you are needed to edit the config.txt file.

```
ssh pi@jkuat-pi01.local  
sudo nano /boot/config.txt  
#hdmi_force_hotplug=1  
hdmi_force_hotplug=1
```

←remove # and save & reboot

sudo shutdown -r now

User: pi

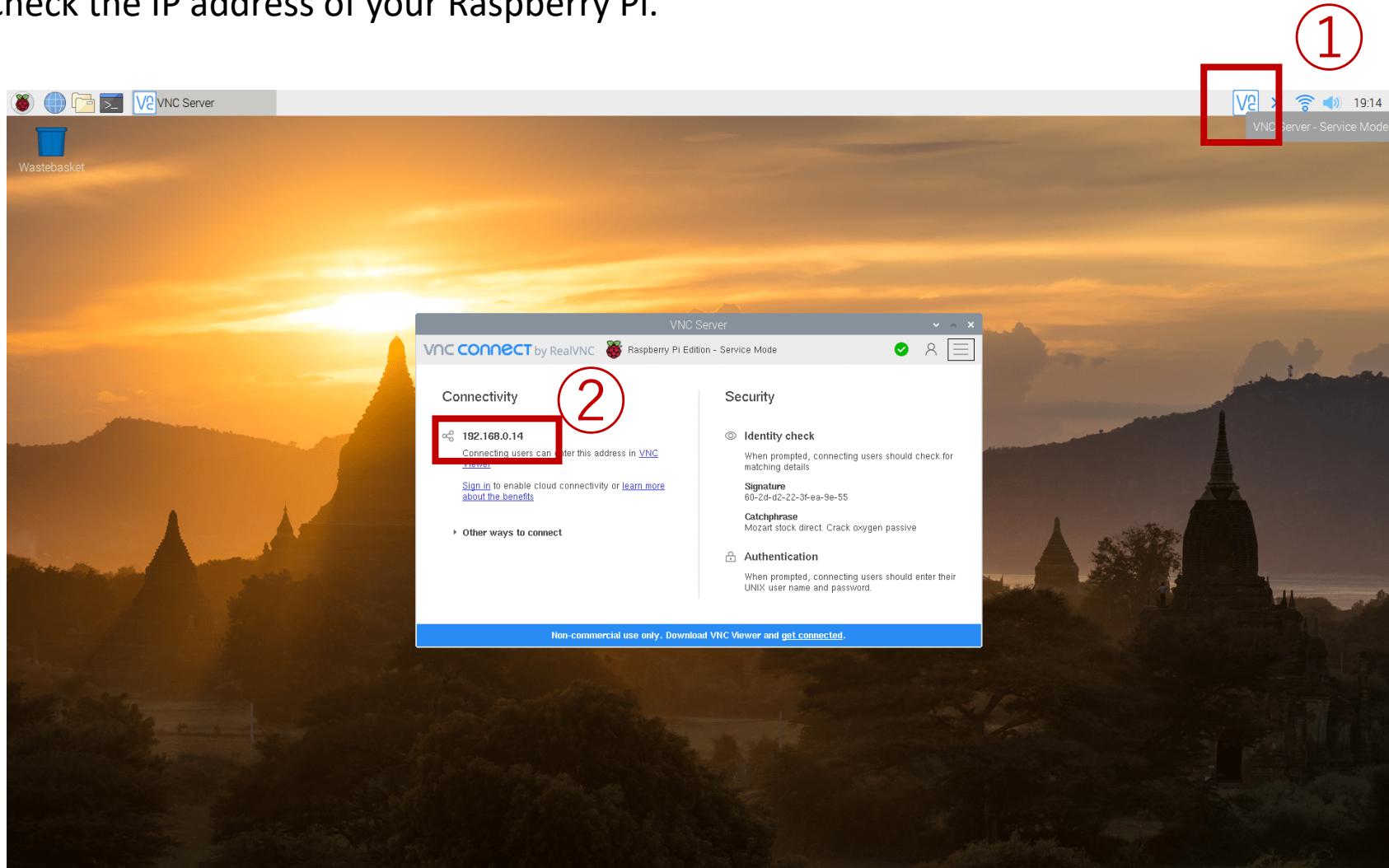
Pw: raspberry

Machine Name:

jkuat-pi01.local ~ jkuat-pi08.local

VNC Connection

Check the IP address of your Raspberry Pi.



VNC connect



Username and Password

V2 Authentication X

 Authenticate to VNC Server
192.168.110.2:5900 (TCP)

Enter VNC Server credentials
(Hint: NOT your RealVNC account details)

Username:

Password: 

Remember password [Forgot password?](#)

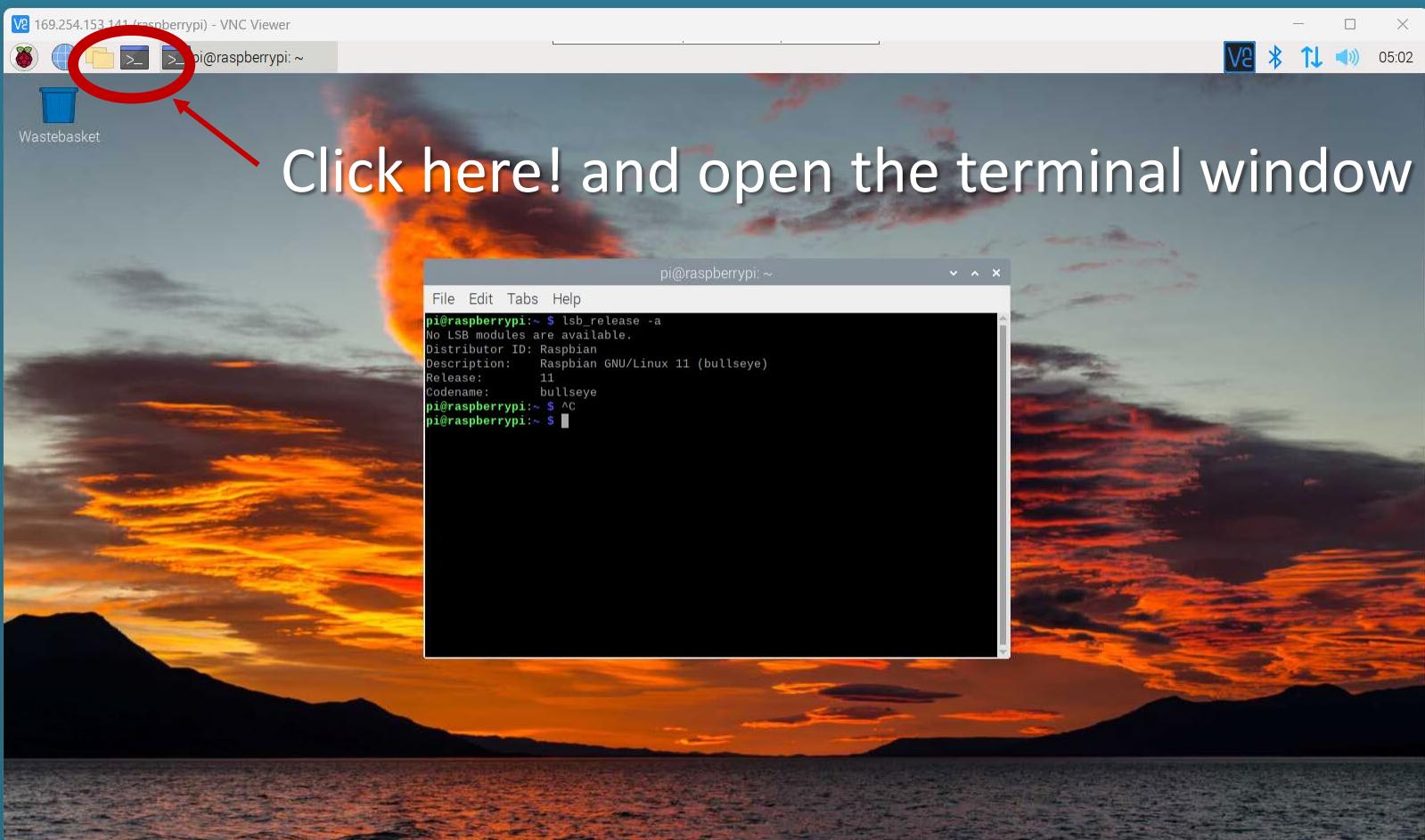
Catchphrase: Miami hand verbal. Desert gong fruit.

Signature: c4-10-f4-ff-ee-c7-7a-ae

OK Cancel

Name:jkuat-pi01.local
User:pi
Pw:raspberry

Connected via VNC viewer



Raspbian OS version : lsb_release -a

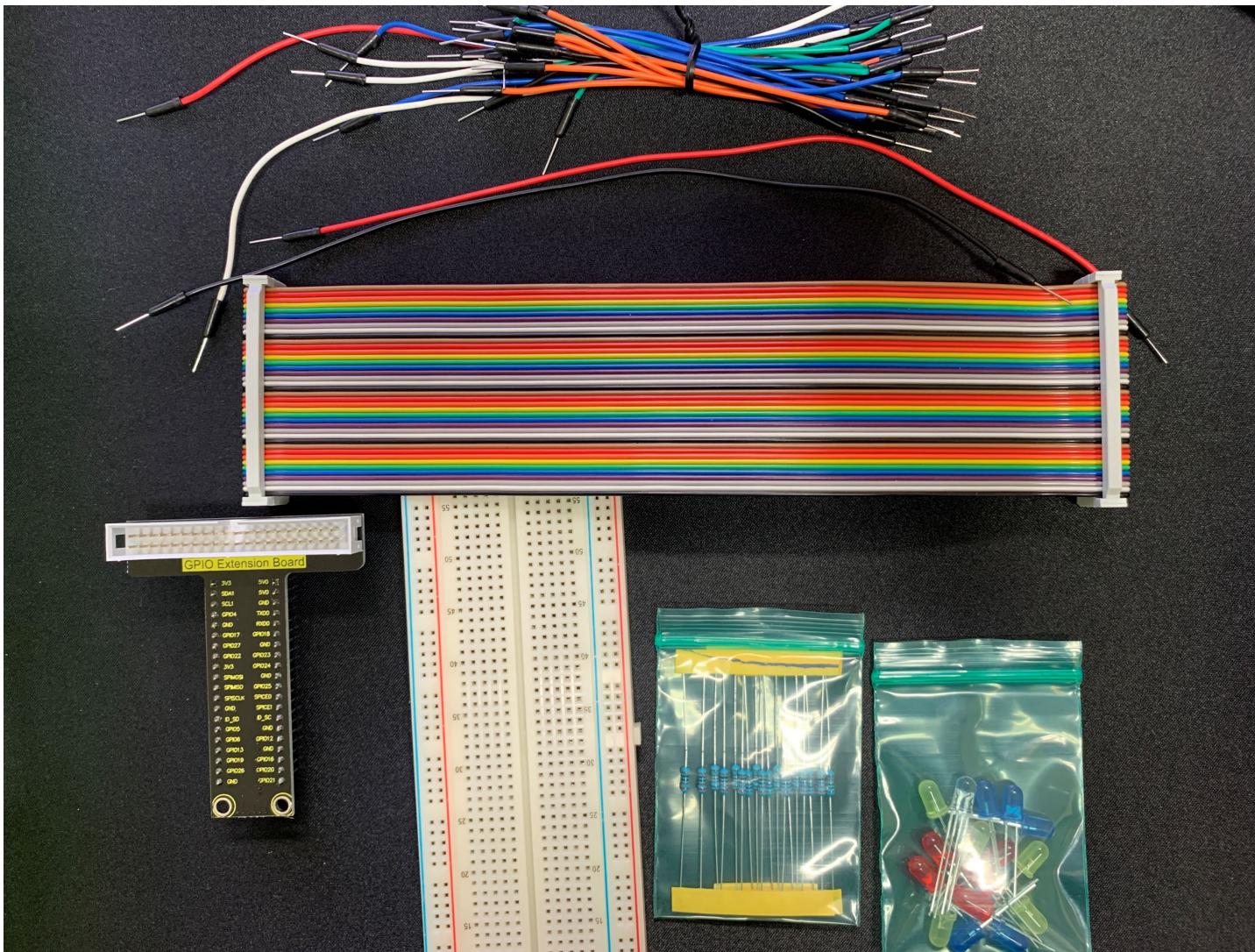
```
pi@raspberrypi:~ $ lsb_release -a
```

```
pi@raspberrypi:~ $ lsb_release -a
No LSB modules are available.
Distributor ID: Raspbian
Description:    Raspbian GNU/Linux 11
(bullseye)
Release:    11
Codename:   bullseye
```

LED flashing

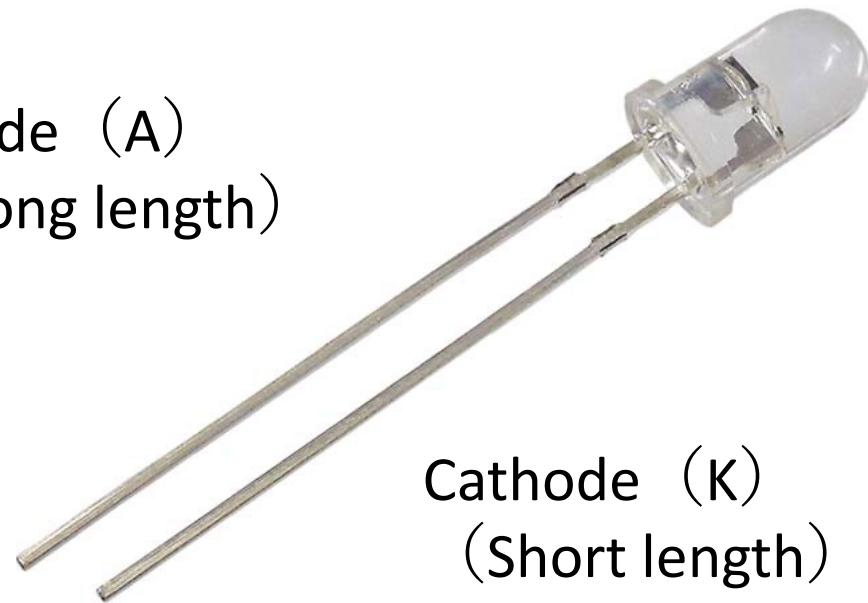


Confirm your handout

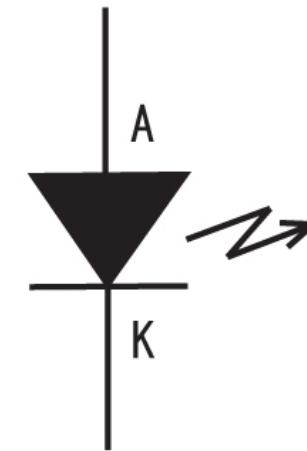


LED Electrode

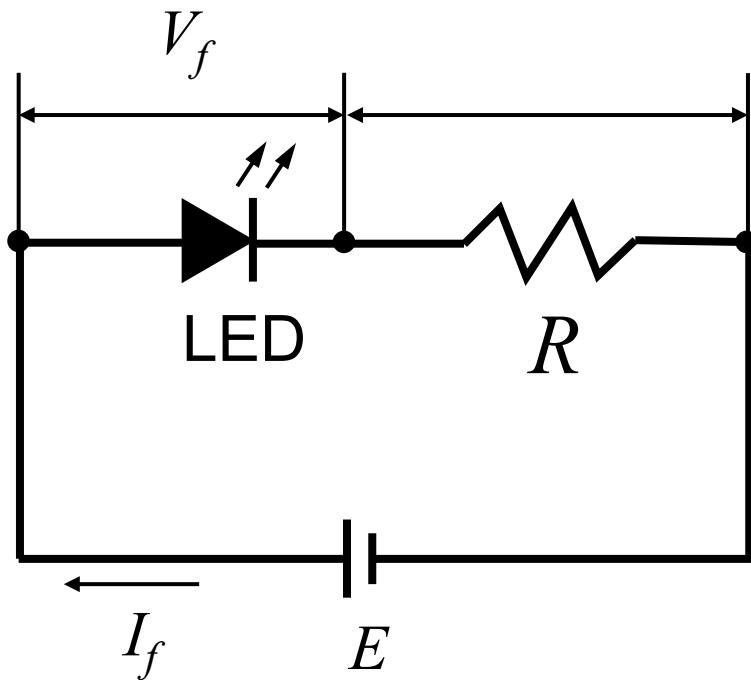
Anode (A)
(Long length)



Cathode (K)
(Short length)



Electric Circuit



<Estimation of Resistance>

$$R = \frac{E - V_f}{I_f}$$

R : Resistance

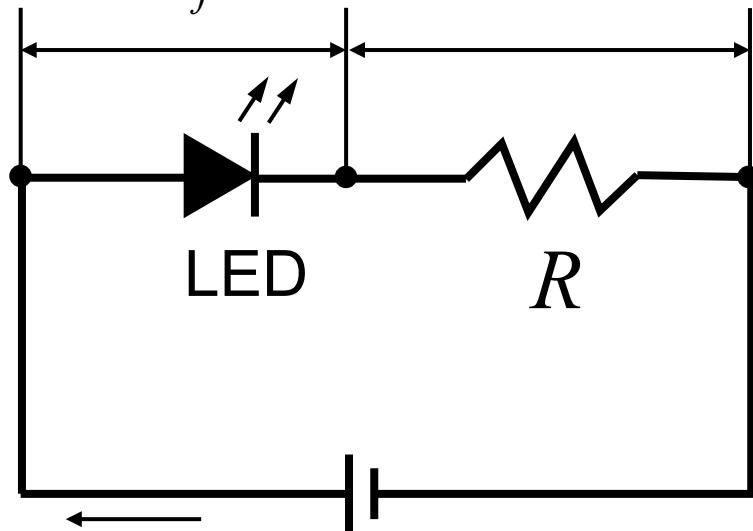
E : Power Supply Voltage

V_f : Forward Voltage

I_f : Forward Current

Electric Circuit

1.85 V
 V_f
1.45 V
(=3.3-1.85)



MAX.
16 mA

<Estimation of Resistance>

$$R = \frac{E - V_f}{I_f}$$

R : Resistance

E : Power Supply Voltage

V_f : Forward Voltage

I_f : Forward Current

If $R = 100 \Omega$ then $I_f = 14.5 \text{ mA}$...O.K. (l.t.16mA)

If $R = 220 \Omega$ then $I_f = 6.6 \text{ mA}$...O.K. (l.t.16mA)

If $R = 1K \Omega$ then $I_f = 1.45 \text{ mA}$...O.K. (l.t.16mA)

Resistor Color Codes

How to Read Resistor Color Codes

6-Band

= $274 \Omega \pm 2\%$, 250 ppm/K

Color	1st Digit	2nd Digit	3rd Digit	Multiplier	Tolerance	Temperature Coefficient
Black	0	0	0	1Ω		250 ppm/K
Brown	1	1	1	10Ω	$\pm 1\%$	100 ppm/K
Red	2	2	2	100Ω	$\pm 2\%$	50 ppm/K
Orange	3	3	3	$1k \Omega$		15 ppm/K
Yellow	4	4	4	$10k \Omega$		25 ppm/K
Green	5	5	5	$100k \Omega$	$\pm 0.5\%$	20 ppm/K
Blue	6	6	6	$1M \Omega$	$\pm 0.25\%$	10 ppm/K
Violet	7	7	7		$\pm 0.1\%$	5 ppm/K
Grey	8	8	8			
White	9	9	9	0.1Ω	$\pm 5\%$	
Gold				0.01Ω	$\pm 10\%$	
Silver						

4-Band

$12 \times 10^5 \pm 5\%$

= $1,200 \text{ k}\Omega \pm 5\%$

5-Band

$100 \times 10^2 \pm 1\%$

= $10,000 \Omega \pm 1\%$

220 Ω

Color	1st Digit	2nd Digit	3rd Digit	Multiplier	Tolerance	Temperature Coefficient
Black	0	0	0	1 Ω		250 ppm/K
Brown	1	1	1	10 Ω	$\pm 1\%$	100 ppm/K
Red	2	2	2	100 Ω	$\pm 2\%$	50 ppm/K
Orange	3	3	3	1k Ω		15 ppm/K
Yellow	4	4	4	10k Ω		25 ppm/K
Green	5	5	5	100k Ω	$\pm 0.5\%$	20 ppm/K
Blue	6	6	6	1M Ω	$\pm 0.25\%$	10 ppm/K
Violet	7	7	7		$\pm 0.1\%$	5 ppm/K
Grey	8	8	8			1 ppm/K
White	9	9	9			
Gold				0.1 Ω	$\pm 5\%$	
Silver				0.01 Ω	$\pm 10\%$	

×



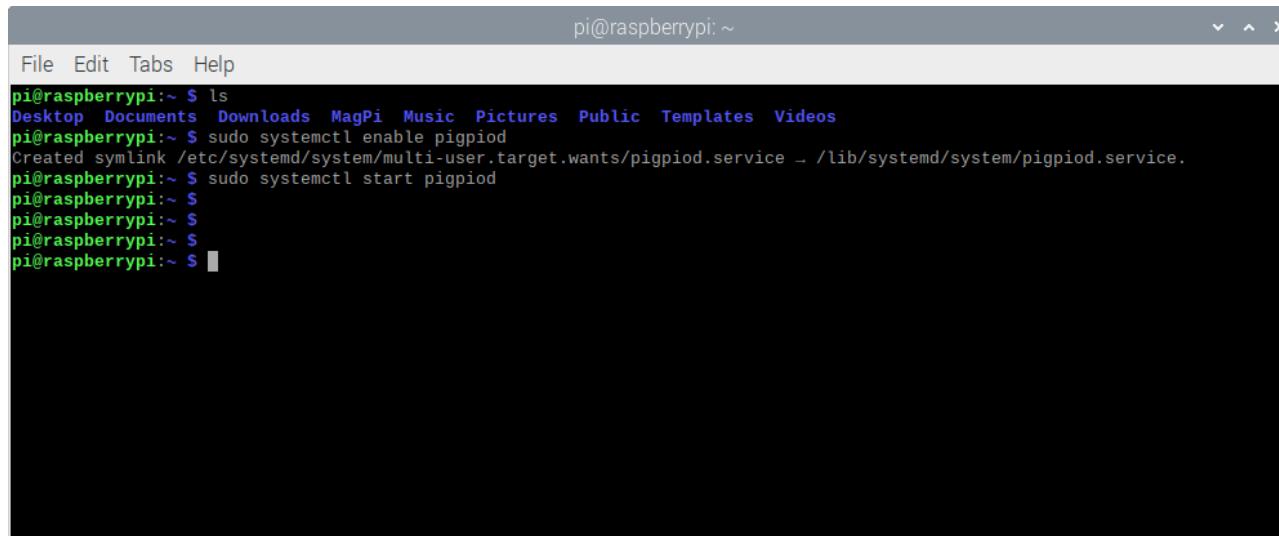
Python Library

Python Libraries : Rpi.GPIO, WiringPi, **pigpio**, gpiozero

Install **pigpio** library

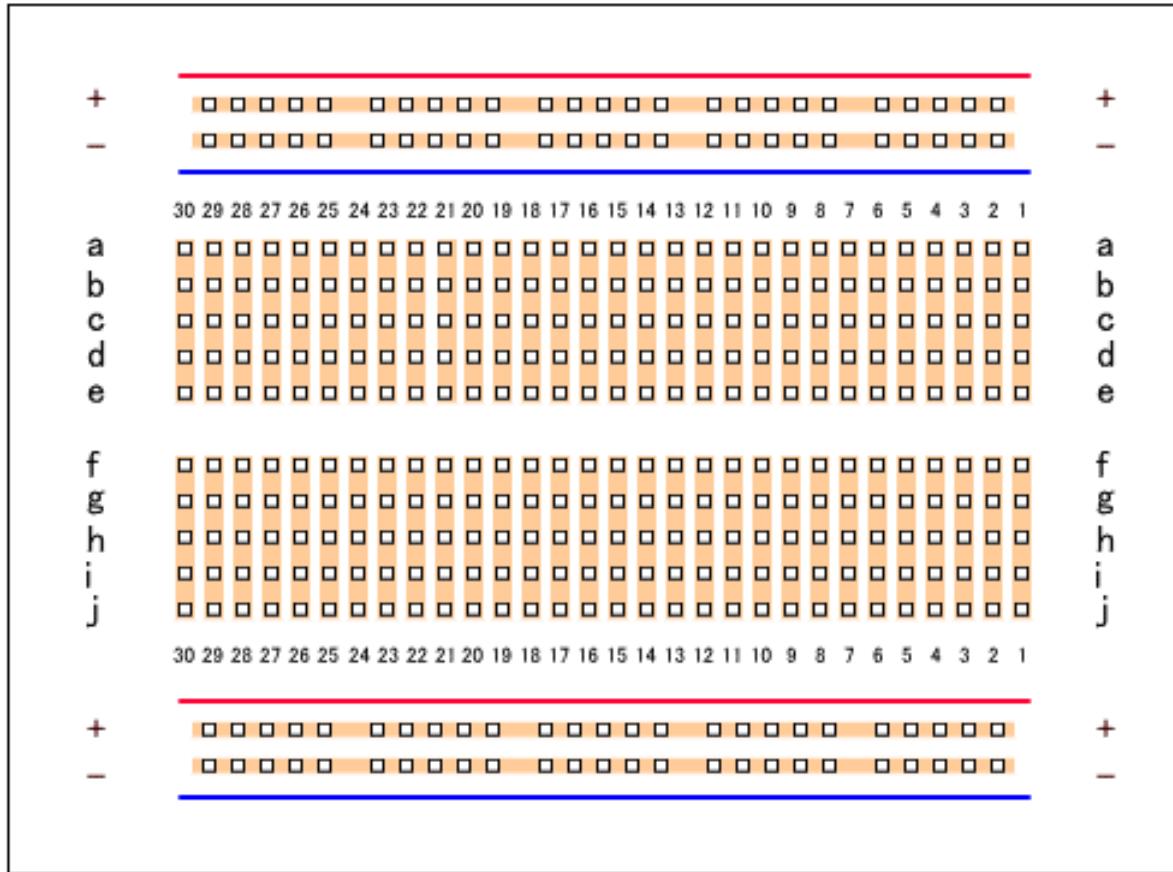
```
pi@raspberrypi:~ $ sudo systemctl enable pigpiod  
pi@raspberrypi:~ $ sudo systemctl start pigpiod
```

You can use the **pigpio** library automatically when you reboot your Raspberry Pi.



```
pi@raspberrypi:~ $ ls  
Desktop Documents Downloads MagPi Music Pictures Public Templates Videos  
pi@raspberrypi:~ $ sudo systemctl enable pigpiod  
Created symlink /etc/systemd/system/multi-user.target.wants/pigpiod.service → /lib/systemd/system/pigpiod.service.  
pi@raspberrypi:~ $ sudo systemctl start pigpiod  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $
```

Breadboard



Edit python program

“nano” is simple editor for raspberry pi.

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ nano ledon.py
```

Python program : ledon.py

```
pi@raspberrypi: ~/TrainingNU
File Edit Tabs Help
GNU nano 3.2                      ledon.py

import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.HIGH )
```

save : **ctrl + s**

quite : **ctrl + x**

Linux command : pwd, ls, more

“more” is a linux command to see a text file

```
pi@raspberrypi:~/TrainingNU
File Edit Tabs Help
pi@raspberrypi:~/TrainingNU $ pwd
/home/pi/TrainingNU
pi@raspberrypi:~/TrainingNU $ ls
ledoff.py ledon.py ledpwm.py rpicamera.sh servo.py
pi@raspberrypi:~/TrainingNU $ more ledon.py
import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.HIGH )

pi@raspberrypi:~/TrainingNU $
```

Linux command : ls -al

```
pi@raspberrypi:~/TrainingNU $ ls -al
total 28
drwxr-xr-x  2 pi pi 4096 Nov 17 14:57 .
drwxr-xr-x 22 pi pi 4096 Nov 17 12:35 ..
-rw-r--r--  1 pi pi  118 Nov  4 10:55 ledoff.py
-rw-r--r--  1 pi pi  119 Nov  4 10:54 ledon.py
-rw-r--r--  1 pi pi  186 Nov  4 12:31 ledpwm.py
-rwxr-xr-x  1 pi pi   87 Nov  4 15:19 rpicamera.sh
-rw-r--r--  1 pi pi  100 Nov  4 13:46 servo.py
pi@raspberrypi:~/TrainingNU $
```

Run your program : python3 ledon.py

```
pi@jkuat-pi01:~/CourseWork $ more ledon.py
import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.HIGH)
pi@jkuat-pi01:~/CourseWork $ python3 ledon.py
```

Python program : ledoff.py

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

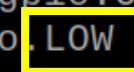
GNU nano 3.2

leddoff.py

```
import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.LOW )
```



[Read 9 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Python program : ledoff.py

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

GNU nano 3.2

leddoff.py

```
import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.LOW )
```

File Name to Write: leddoff.py

^G Get Help

^C Cancel

M-D DOS Format

M-M Mac Format

M-A Append

M-P Prepend

M-B Backup File

^T To Files

Python program : ledoff.py

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

```
pi@raspberrypi:~/TrainingNU $ more ledon.py
import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.HIGH )
```

```
pi@raspberrypi:~/TrainingNU $ nano ledon.py
pi@raspberrypi:~/TrainingNU $ more ledoff.py
import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.LOW )
```

```
pi@raspberrypi:~/TrainingNU $ █
```

Linux command : less

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

pi@raspberrypi:~/TrainingNU \$ less ledon.py

Linux command : less

```
pi@raspberrypi: ~/TrainingNU
File Edit Tabs Help

import pigpio

LED_PIN = 27
pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT )
pi.write( LED_PIN, pigpio.HIGH )

ledon.py (END)
```

quite : **ctrl + c**

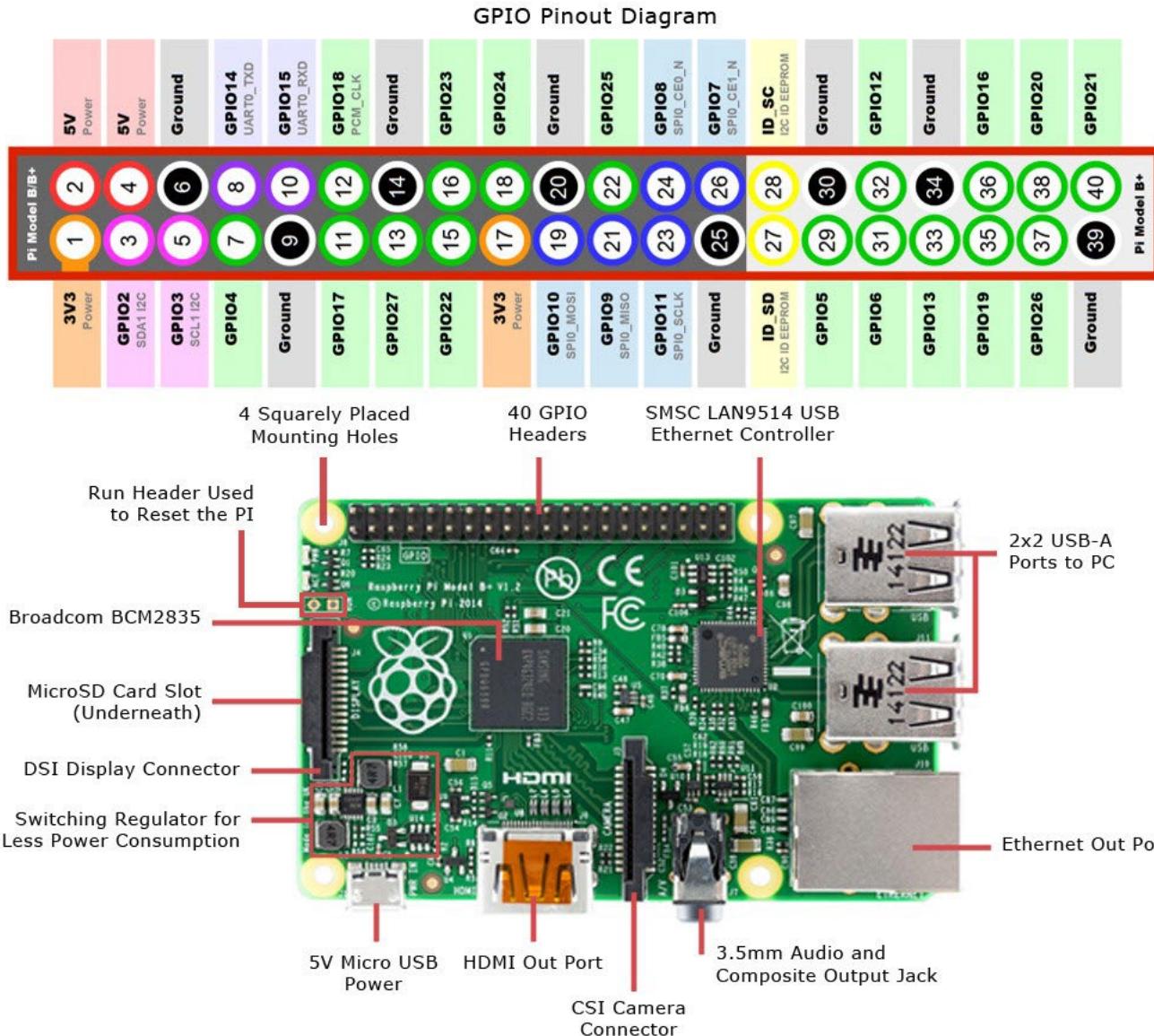
Linux command : pinout

```
pi@raspberrypi:~ $ pinout
| 00000000000000000000 J8
| 10000000000000000000
|
| Pi Model 3B V1.2
| [D] |SoC |
| [S] |
| [I] +---+
|     |C| [S]
| pwr |HDMI|[I]|A| +---+ Net
|     |V| +---+
|
Revision          : a32082
SoC              : BCM2837
RAM              : 1024Mb
Storage           : MicroSD
USB ports         : 4 (excluding power)
Ethernet ports   : 1
Wi-fi             : True
Bluetooth         : True
Camera ports (CSI): 1
Display ports (DSI): 1

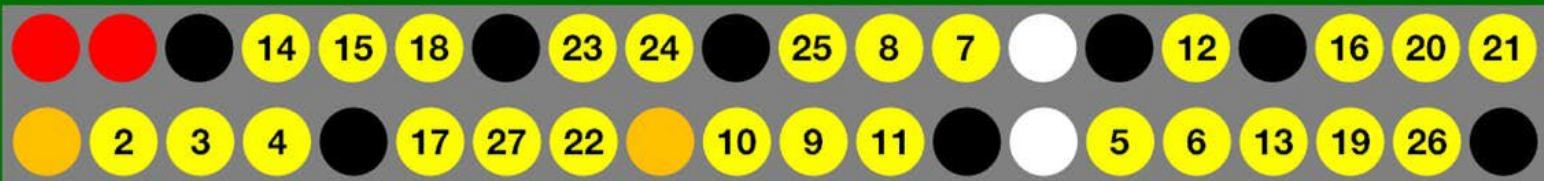
J8:
  3V3  (1) (2)  5V
  GPIO2 (3) (4)  5V
  GPIO3 (5) (6) GND
  GPIO4 (7) (8) GPIO14
  GND  (9) (10) GPIO15
  GPIO17 (11) (12) GPIO18
  GPIO27 (13) (14) GND
  GPIO22 (15) (16) GPIO23
  3V3 (17) (18) GPIO24
  GPIO10 (19) (20) GND
  GPIO9  (21) (22) GPIO25
  GPIO11 (23) (24) GPIO08
  GND  (25) (26) GPIO07
  GPIO08 (27) (28) GPIO01
  GPIO05 (29) (30) GND
  GPIO06 (31) (32) GPIO012
  GPIO13 (33) (34) GND
  GPIO19 (35) (36) GPIO16
  GPIO26 (37) (38) GPIO020
  GND (39) (40) GPIO21

For further information, please refer to https://pinout.xyz/
```

Raspberry pi GPIO



Raspberry pi GPIO



Raspberry Pi A+ / B+ and Raspberry Pi 2 GPIO pins



GPIO



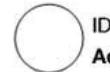
Ground



3.3v

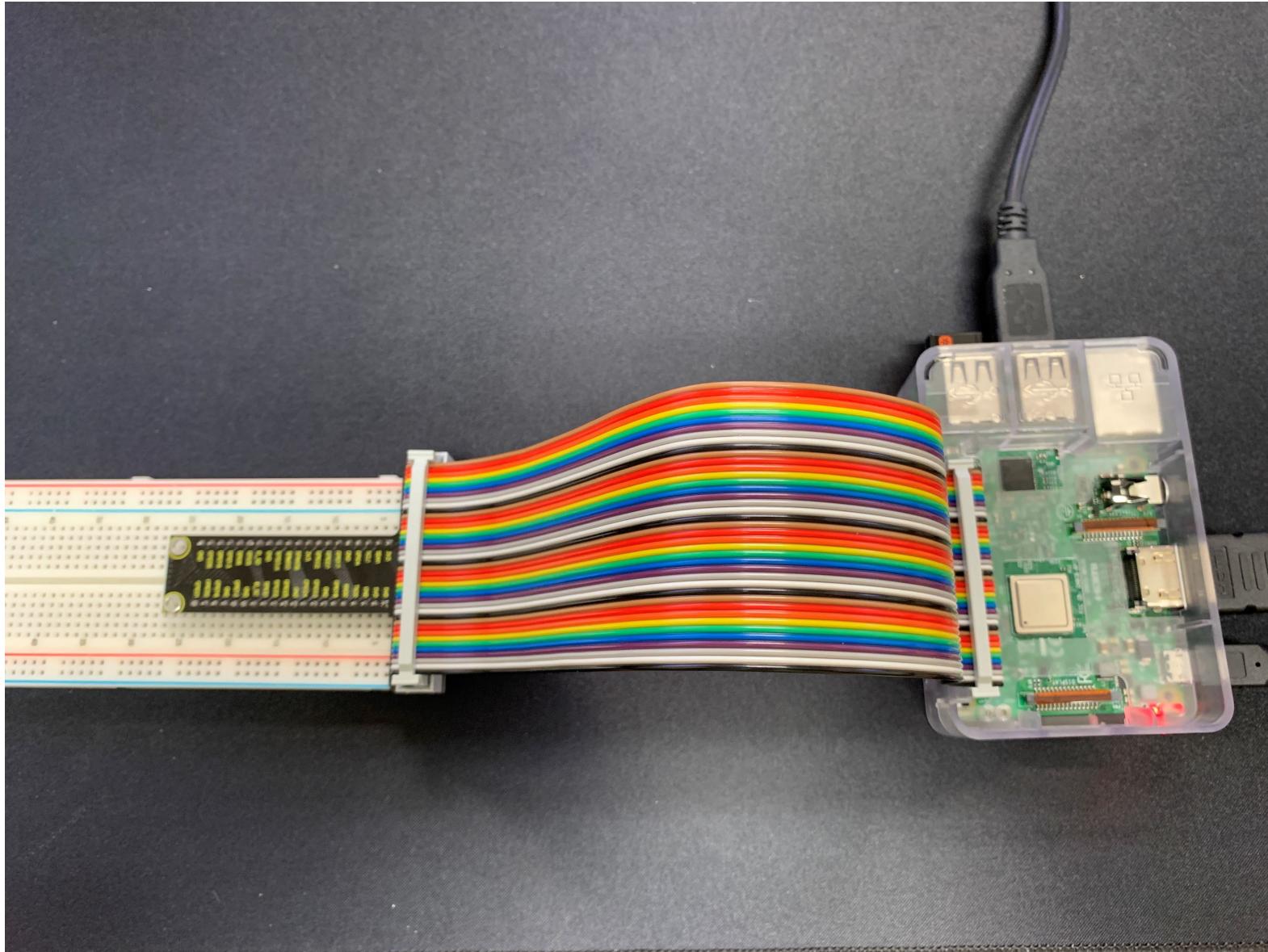


5v

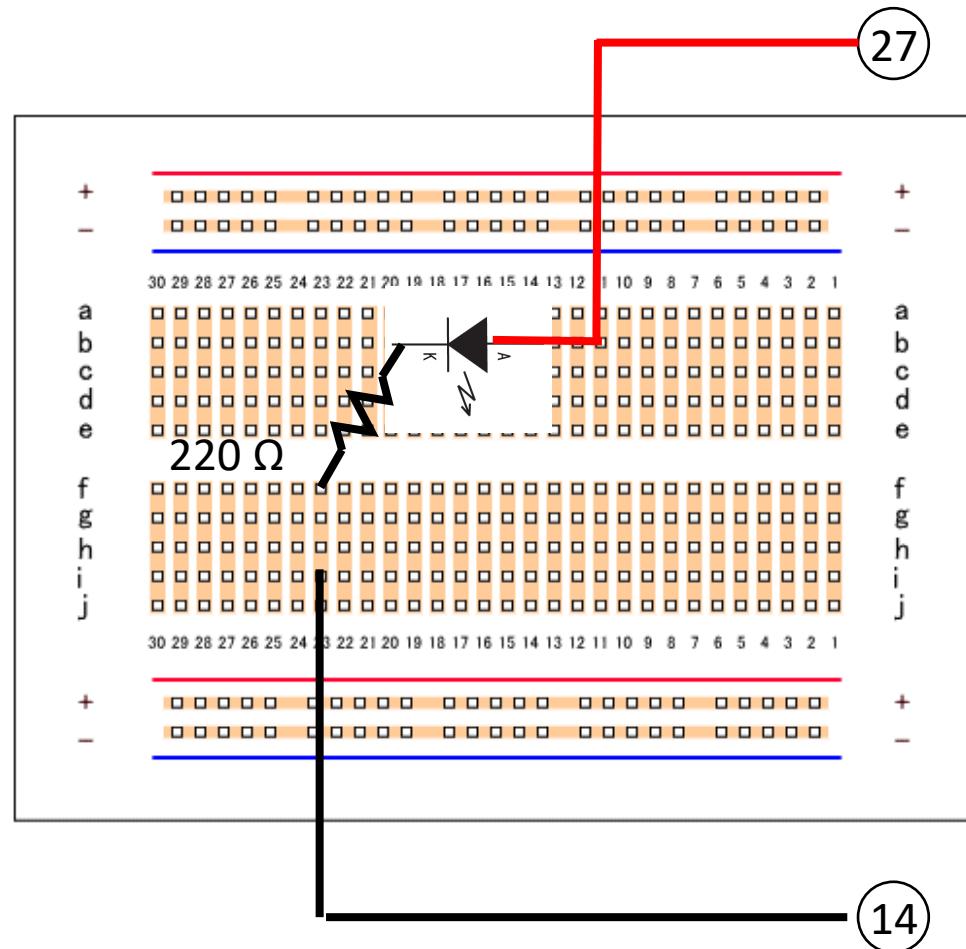
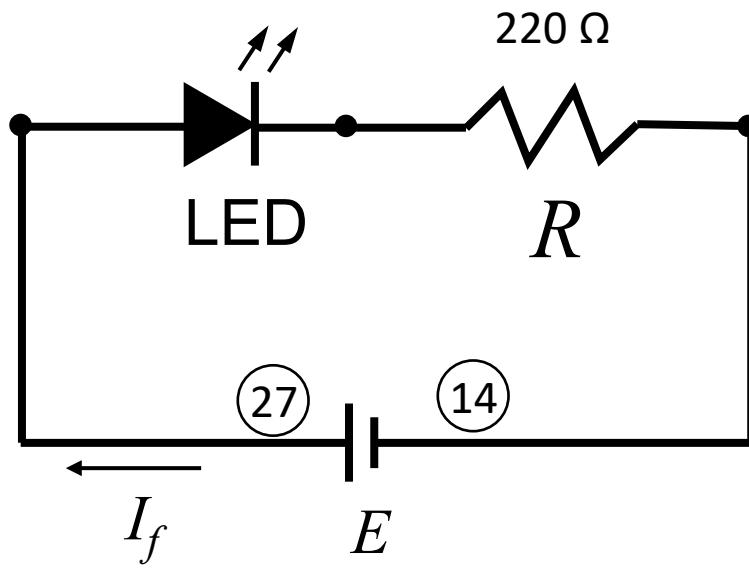
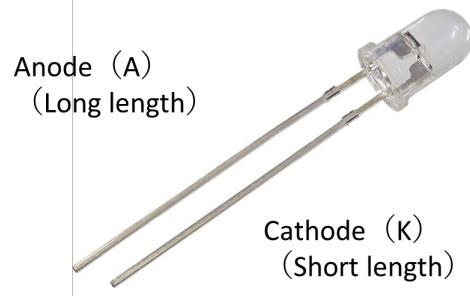


ID EEPROM
Advanced use only!

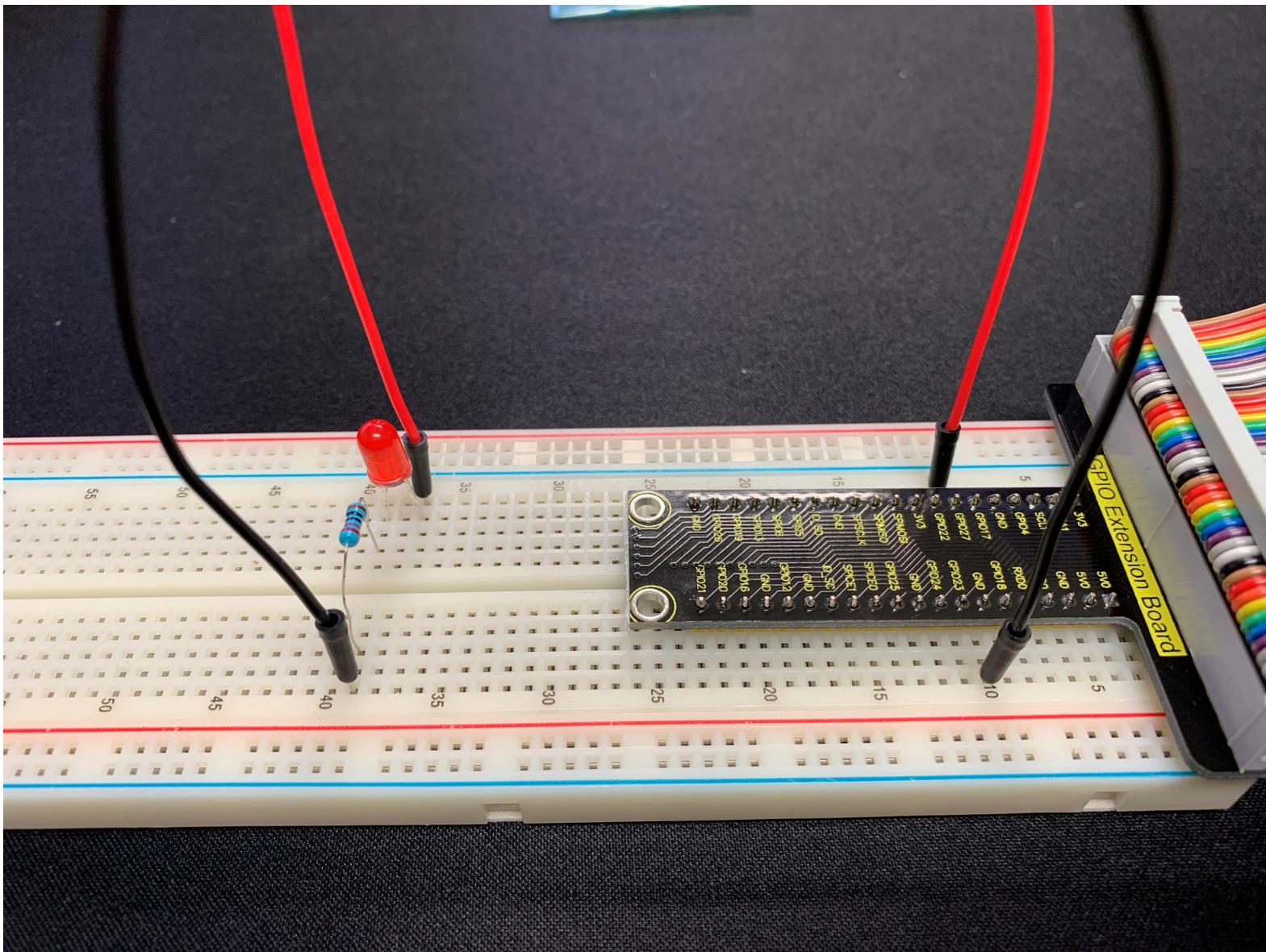
40pin flat cable



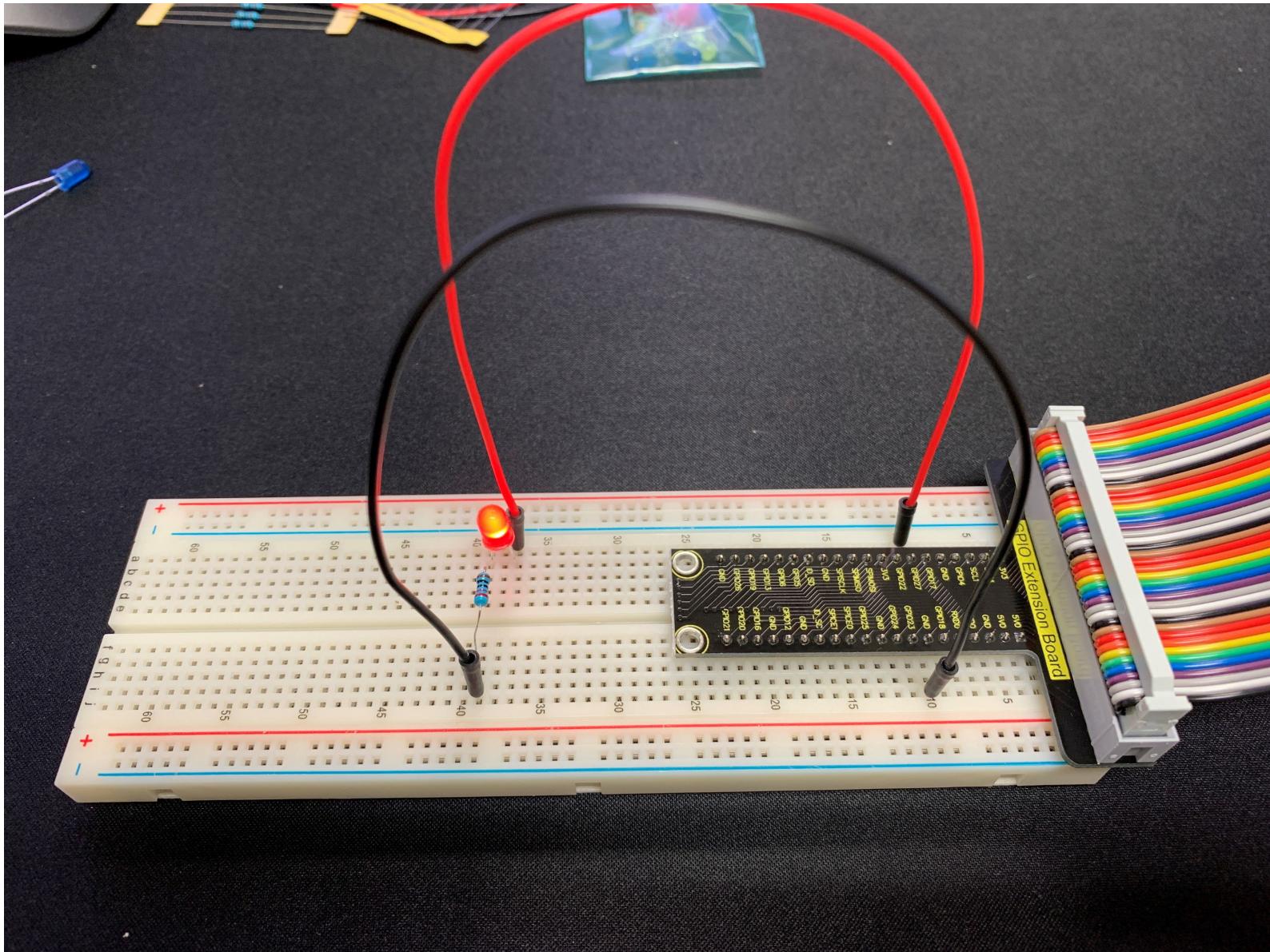
Circuit layout for LED



Circuit layout for LED



Turn on your LED



Python program : ledpwm.py

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

```
pi@raspberrypi:~/TrainingNU $ nano ledpwm.py
```

Python program : ledpwm.py

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

GNU nano 3.2

ledpwm.py

```
import pigpio
LED_PIN = 27

pi = pigpio.pi()

pi.set_mode( LED_PIN, pigpio.OUTPUT)
pi.set_PWM_frequency( LED_PIN, 1)
pi.set_PWM_range( LED_PIN, 100)

pi.set_PWM_dutycycle( LED_PIN, 50)
```

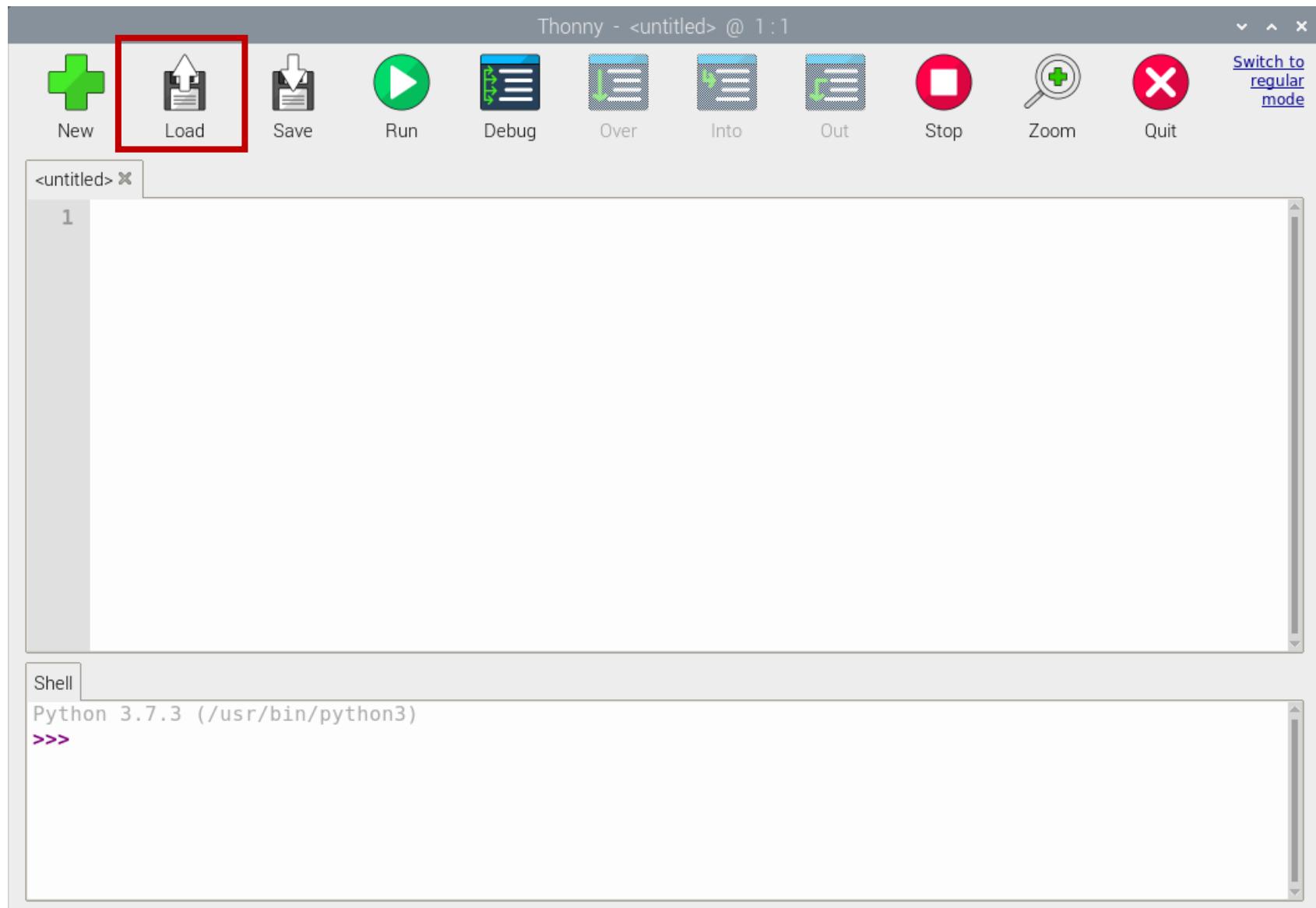
[Read 11 lines]

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos
^X Exit **^R** Read File **^** Replace **^U** Uncut Text **^T** To Spell **^_** Go To Line

Programming Environment : Thonny Python IDE



Thonny Python IDE



Thonny Python IDE

Thonny - /home/pi/TrainingNU/ledpwm.py @ 5:1

New Load Save Run Debug Over Into Out Stop Zoom Quit Switch to regular mode

ledpwm.py ledoff.py

```
1 import pigpio
2 LED_PIN = 27
3
4 pi = pigpio.pi()
5
6 pi.set_mode( LED_PIN, pigpio.OUTPUT)
7 pi.set_PWM_frequency( LED_PIN, 2)
8 pi.set_PWM_range( LED_PIN, 100)
9
10 pi.set_PWM_dutycycle( LED_PIN, 50)
11
12
```

Shell

```
>>>
Python 3.7.3 (/usr/bin/python3)
>>> %Run ledpwm.py
>>> %Run ledoff.py
>>> %Run ledpwm.py
>>> %Run ledoff.py
>>>
```

Thonny Python IDE

Thonny - /home/pi/TrainingNU/ledpwm.py @ 5:1

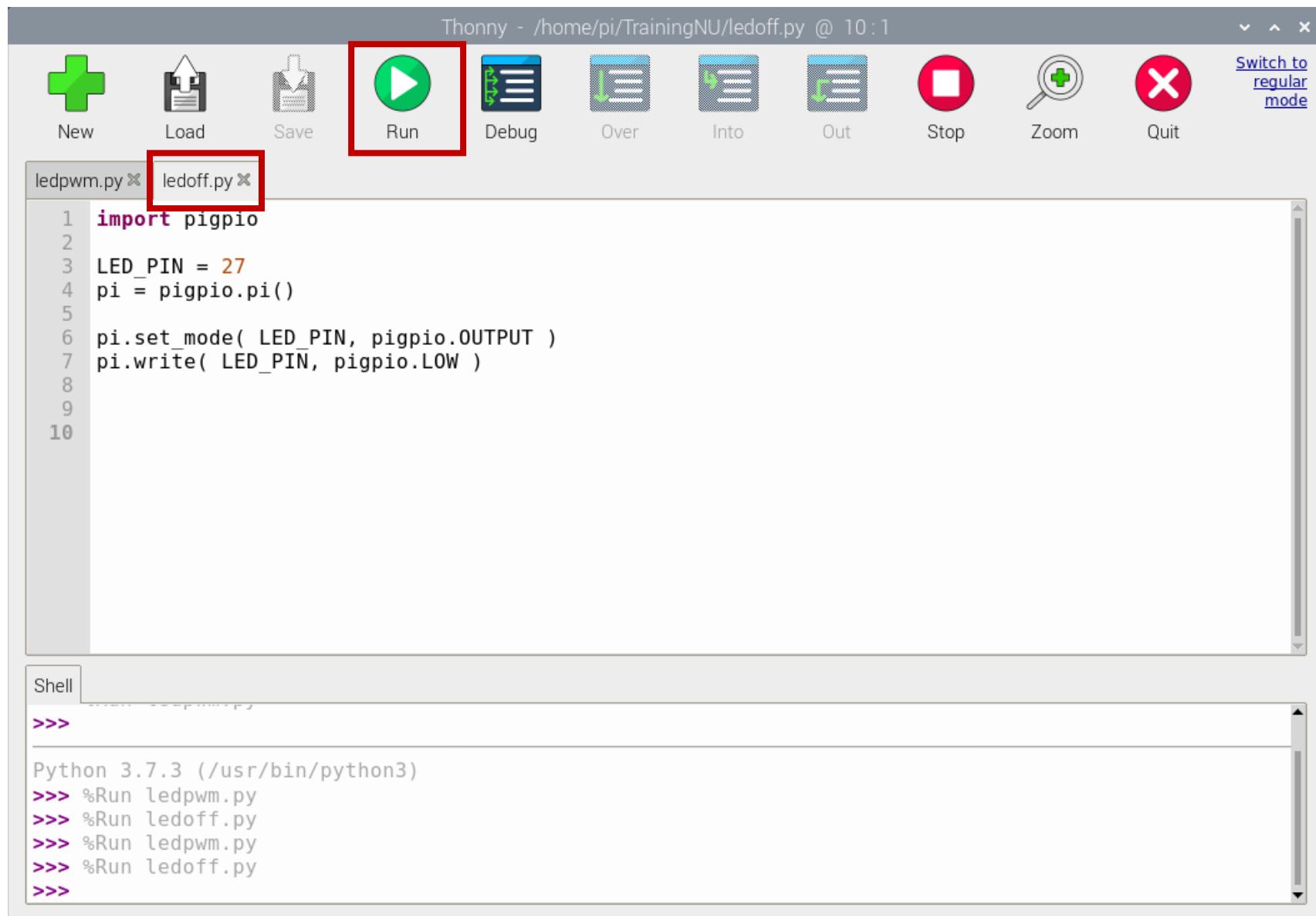
The screenshot shows the Thonny Python IDE interface. At the top is a menu bar with icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. The 'Run' icon is highlighted with a red box. To the right of the menu is a link to 'Switch to regular mode'. Below the menu is a toolbar with tabs for 'ledpwm.py' (selected) and 'ledoff.py'. The main area contains the following Python code:

```
1 import pigpio
2 LED_PIN = 27
3
4 pi = pigpio.pi()
5
6 pi.set_mode( LED_PIN, pigpio.OUTPUT)
7 pi.set_PWM_frequency( LED_PIN, 2)
8 pi.set_PWM_range( LED_PIN, 100)
9
10 pi.set_PWM_dutycycle( LED_PIN, 50)
11
12
```

Below the code editor is a 'Shell' tab. The shell window displays the following Python session:

```
>>>
Python 3.7.3 (/usr/bin/python3)
>>> %Run ledpwm.py
>>> %Run ledoff.py
>>> %Run ledpwm.py
>>> %Run ledoff.py
>>>
```

Thonny Python IDE



Thonny Python IDE

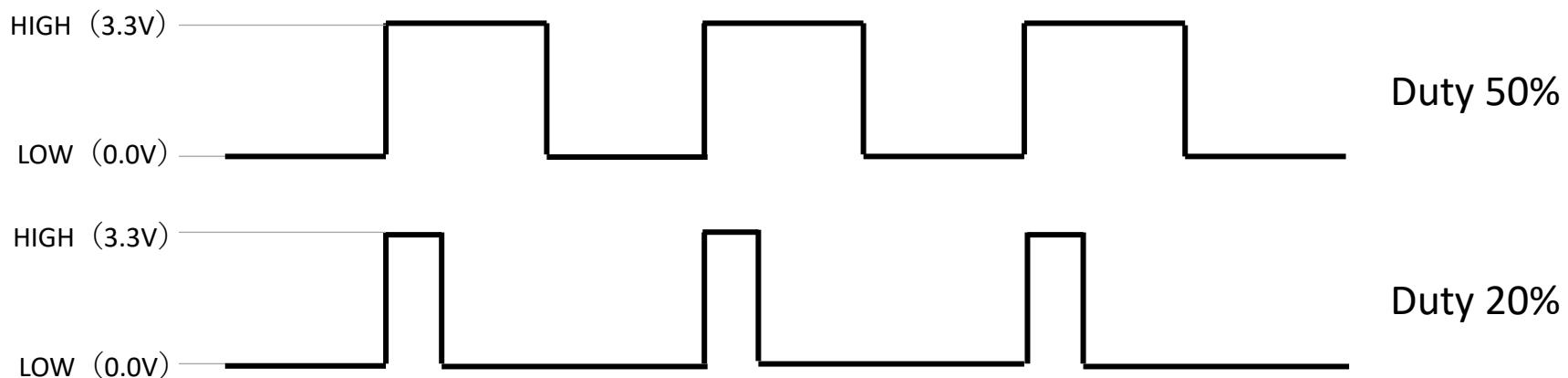
Thonny - /home/pi/TrainingNU/ledpwm.py @ 5:1

New Load Save Run Debug Over Into Out Stop Zoom Quit Switch to regular mode

ledpwm.py ledoff.py

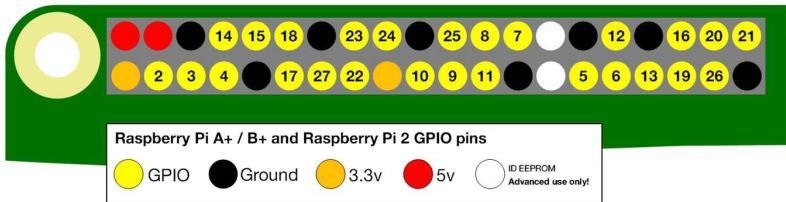
```
1 import pigpio
2 LED_PIN = 27
3
4 pi = pigpio.pi()
5
6 pi.set_mode( LED_PIN, pigpio.OUTPUT)
7 pi.set_PWM_frequency( LED_PIN, 2)
8 pi.set_PWM_range( LED_PIN, 100)
9
10 pi.set_PWM_dutycycle( LED_PIN, 50)
11
12
```

PWM : Pulse Width Modulation



Servo motor

Servo Motor SG90

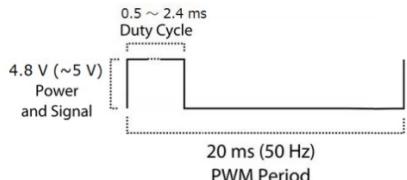


Specifications

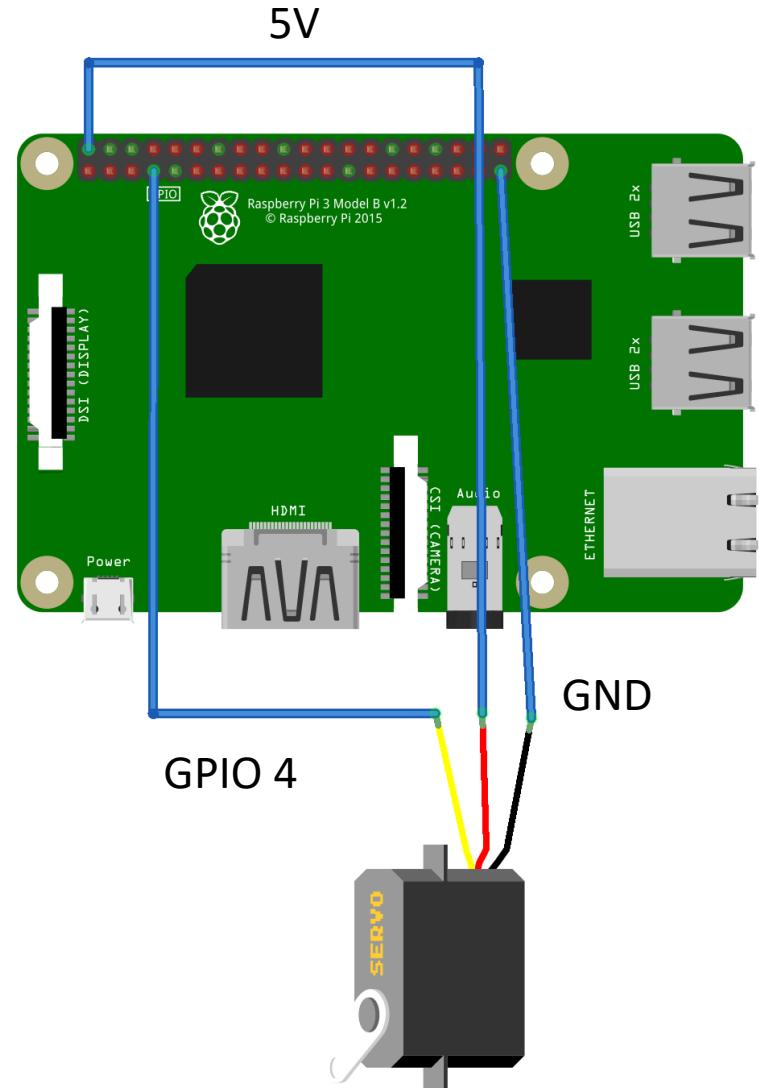
- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 μ s
- Temperature range: 0 °C – 55 °C



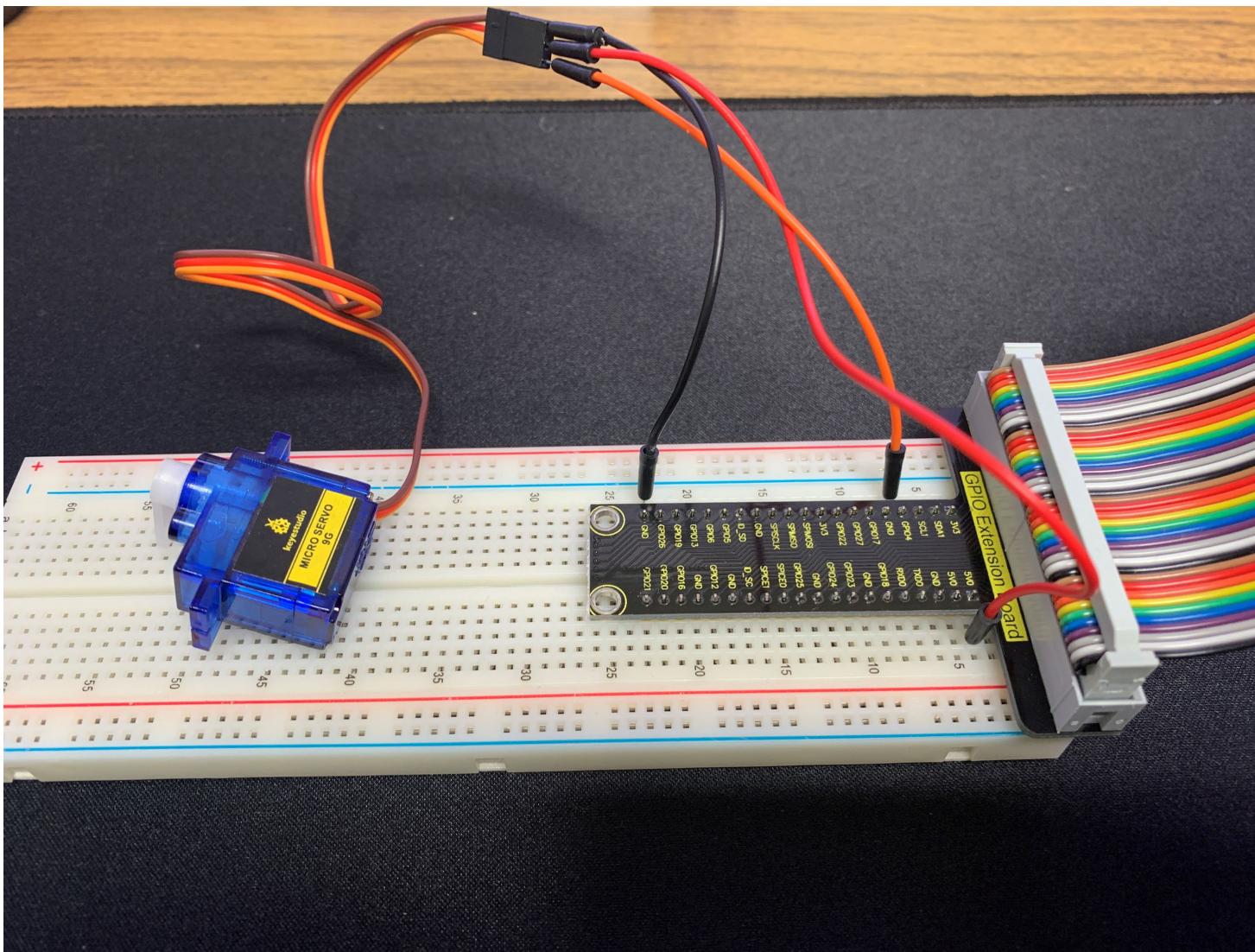
PWM=Orange (□□)
Vcc = Red (+)
Ground=Brown (-)



Position "0" (1.45 ms pulse) is middle, "90" (~2.4 ms pulse) is all the way to the right,
"-90" (~0.5 ms pulse) is all the way left.



Servo Motor SG90



servo.py

Thonny - /home/pi/TrainingNU/servo.py @ 15 : 36

New Load Save Run Debug Over Into Out Stop Zoom Quit

[Switch to regular mode](#)

servo.py

```
1 import pigpio
2 import time
3
4 servo = pigpio.pi()
5 PIN = 4
6
7 try:
8     while True:
9         servo.set_servo_pulsewidth(PIN, 1450 ) # 0 deg.
10        time.sleep(1)
11        servo.set_servo_pulsewidth(PIN, 2400 ) # +90 deg.
12        time.sleep(1)
13        servo.set_servo_pulsewidth(PIN, 1450 ) # 0 deg.
14        time.sleep(1)
15        servo.set_servo_pulsewidth(PIN, 550 ) # -90 deg.
16        time.sleep(1)
17
18 except KeyboardInterrupt:      # stop by CTRL+C
19     pass
20
21 servo.set_PWM_frequency( PIN, 0)
22 servo = pigpio.pi()
23
24
```

Servo Motor SG90

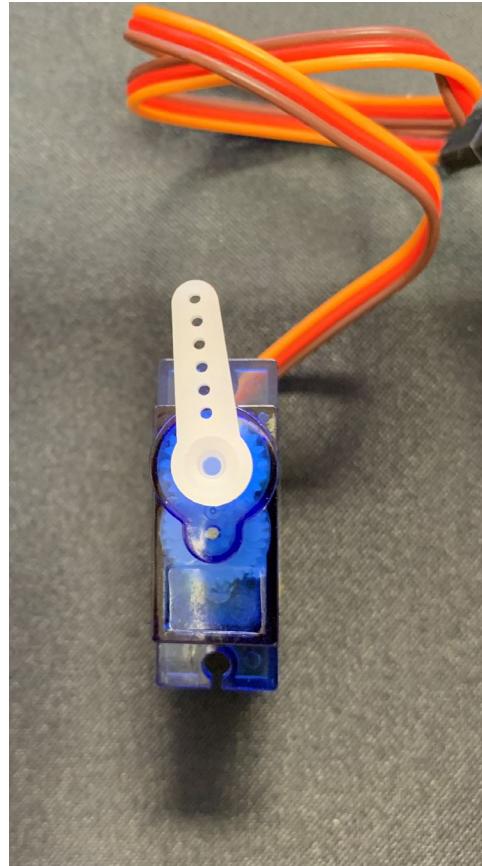
Pulse width = 2400us

Angle = 90 deg.



Pulse width = 1450us

Angle = 0 deg.



Pulse width = 500us

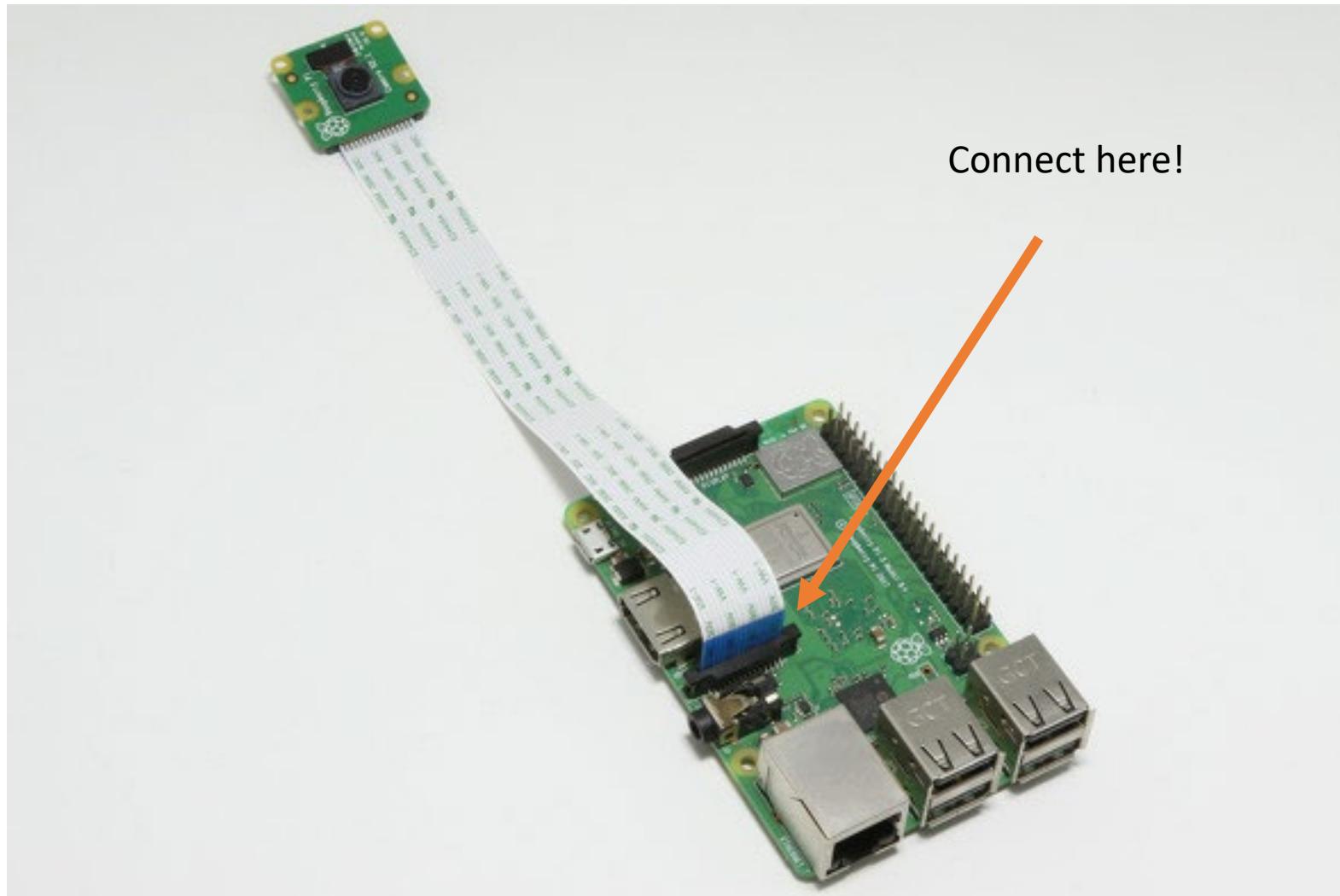
Angle = -90 deg.



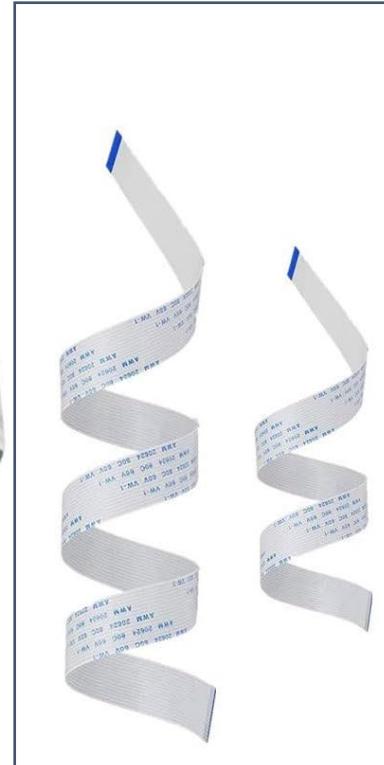
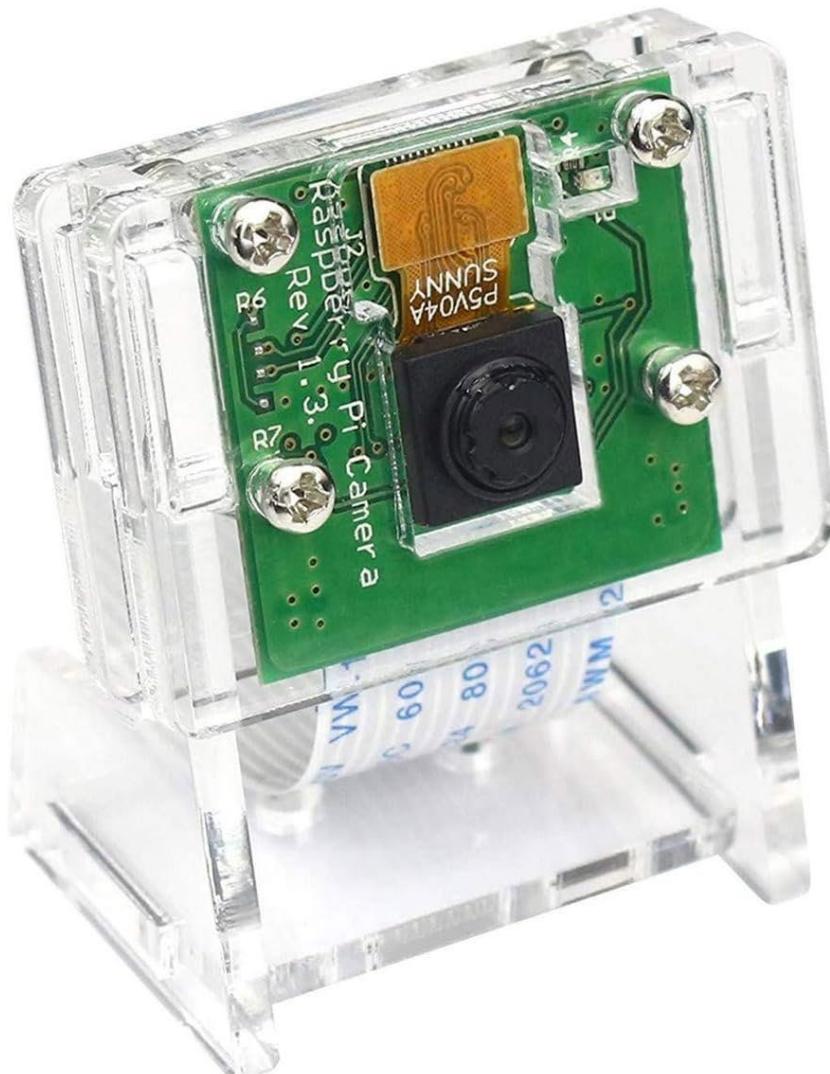
What happen if you input the pulse width = 3000 ?

Camera module

RasTech Raspberry Pi Camera Module



RasTech Raspberry Pi Camera Module v1 (ov5647)



For pi3/pi4

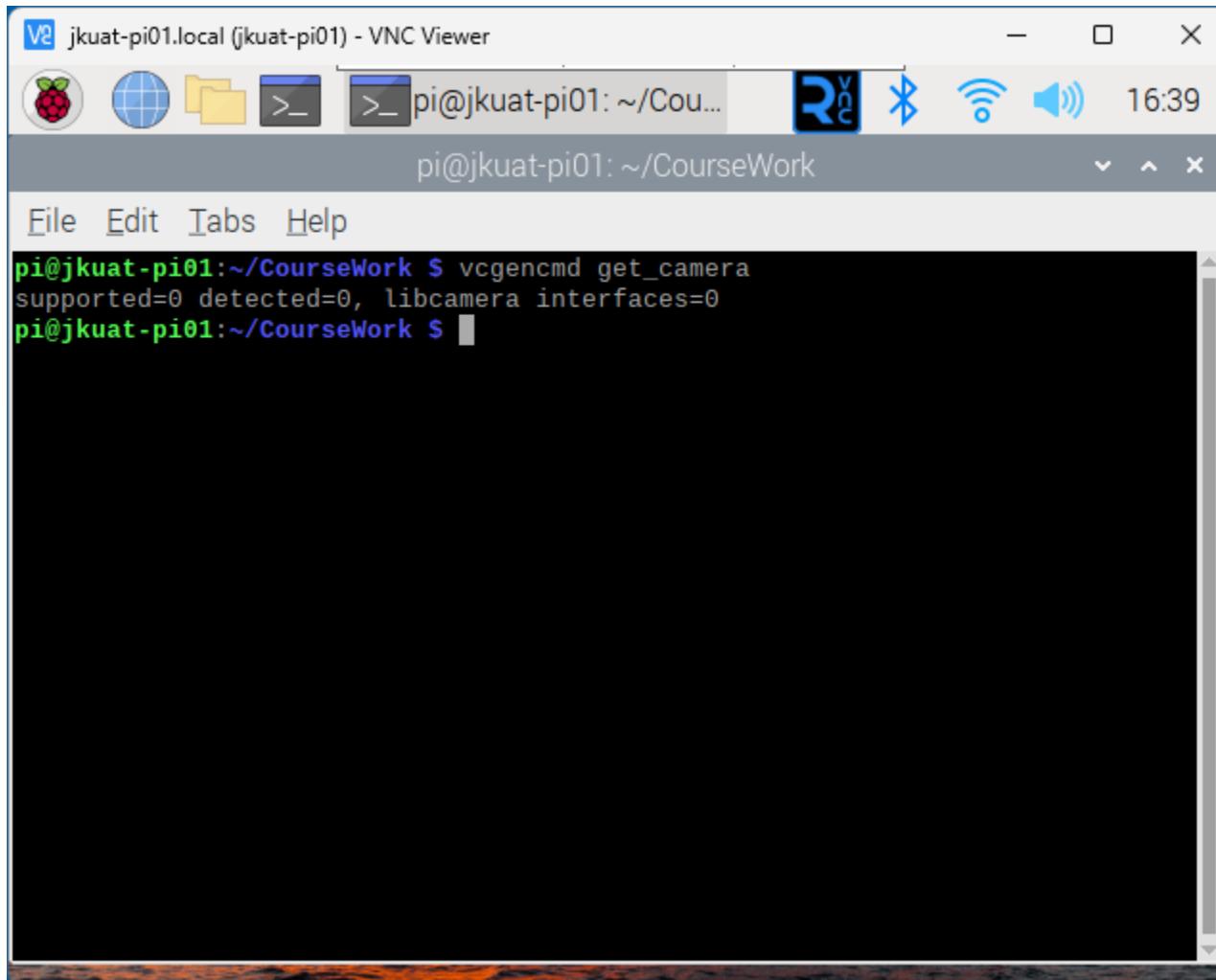


For pi zero



vcgencmd get_camera

vcgencmd get_camera



A screenshot of a VNC viewer window titled "jkuat-pi01.local (jkuat-pi01) - VNC Viewer". The window shows a terminal session on a Raspberry Pi. The terminal prompt is "pi@jkuat-pi01: ~/CourseWork \$". The user has run the command "vcgencmd get_camera", which outputs:

```
pi@jkuat-pi01:~/CourseWork $ vcgencmd get_camera
supported=0 detected=0, libcamera interfaces=0
pi@jkuat-pi01:~/CourseWork $
```

libcamera-apps

```
sudo raspi-config  
vcgencmd get_camera
```

libcamera-hello // check the camera

```
libcamera-hello -t 0  
//stop with ctrl+C
```

libcamera-jpeg // save as a jpeg format

```
libcamera-jpeg -o test1.jpg  
libcamera-jpge -o test2.jpg -t 2000 –width 640 –height 480 //preview 2sec  
libcamera-jpg –h // view all of options
```

libcamera-still //

```
libcamera-still -o test3.jpg  
libcamera-still –o %Y-%m-%d_%H%M.jpg
```

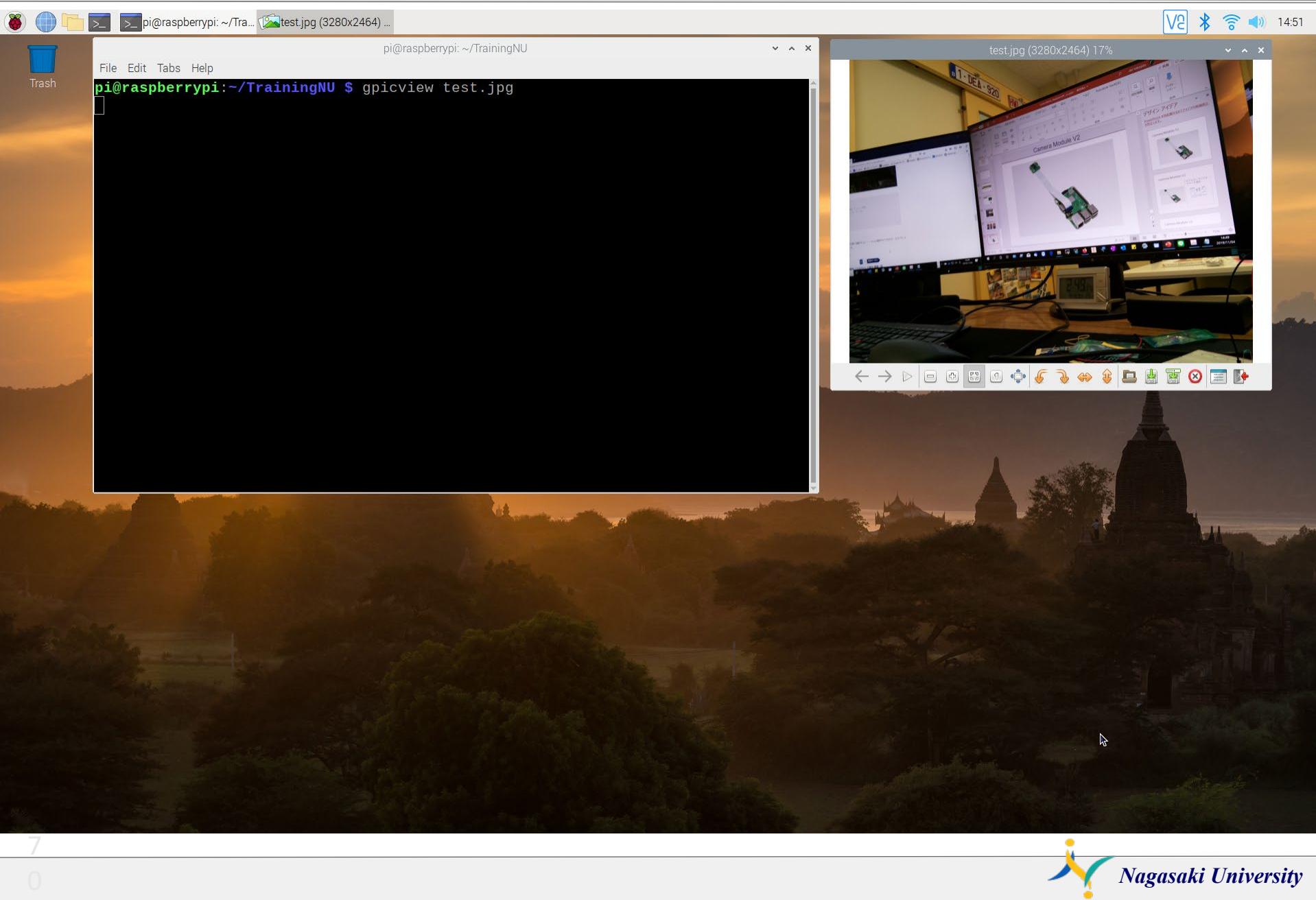
Linux command : gpicview

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

```
pi@raspberrypi:~/TrainingNU $ gpicview test.jpg
```

Linux command : gpicview



Shell program : rpicamera.sh

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

```
pi@raspberrypi:~/TrainingNU $ nano rpicamera.sh |
```

Shell program : rpicamera.sh

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

GNU nano 3.2

rpicamera.sh

```
#!/bin/bash
DATE=$(date "+%Y-%m-%d_%H%M")
libcamera-still -o $DATE.jpg
```

[Read 4 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Linux command : ls -al

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

```
pi@raspberrypi:~/TrainingNU $ nano rpicamera.sh
pi@raspberrypi:~/TrainingNU $ ./rpicamera.sh
pi@raspberrypi:~/TrainingNU $ ls -al
total 17192
drwxr-xr-x  2 pi pi    4096 Nov  4 15:21 .
drwxr-xr-x 20 pi pi    4096 Nov  4 14:42 ..
-rw-r--r--  1 pi pi 4209097 Nov  4 15:21 2019-11-04_1521.jpg
-rw-r--r--  1 pi pi 4807345 Nov  4 15:14 .jpg
-rw-r--r--  1 pi pi     118 Nov  4 10:55 ledoff.py
-rw-r--r--  1 pi pi     119 Nov  4 10:54 ledon.py
-rw-r--r--  1 pi pi    186 Nov  4 12:31 ledpwm.py
-rwxr-xr-x  1 pi pi     87 Nov  4 15:19 rpicamera.sh
-rw-r--r--  1 pi pi    100 Nov  4 13:46 servo.py
-rw-r--r--  1 pi pi 4249279 Nov  4 14:53 test2.jpg
-rw-r--r--  1 pi pi 4303484 Nov  4 14:49 test.jpg
pi@raspberrypi:~/TrainingNU $
```

Cron daemon

Linux command : crontab -e

```
pi@raspberrypi: ~/TrainingNU
File Edit Tabs Help
pi@raspberrypi:~/TrainingNU $ crontab -e
```

```
pi@raspberrypi: ~/TrainingNU
File Edit Tabs Help
pi@raspberrypi:~/TrainingNU $ crontab -e
no crontab for pi - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]: 1
```

`*/1 * * * * /home/pi/rpicamera.sh`

Every 1 minute execute the program rpicamera.sh

Linux command : ls -al

pi@raspberrypi: ~/TrainingNU

File Edit Tabs Help

```
2. /usr/bin/vim.tiny  
3. /bin/ed
```

Choose 1-3 [1]: 1

crontab: installing new crontab

pi@raspberrypi:~/TrainingNU \$ crontab -e

crontab: installing new crontab

pi@raspberrypi:~/TrainingNU \$ ls -al

total 29192

```
drwxr-xr-x  2 pi pi    4096 Nov  4 15:27 .
drwxr-xr-x 20 pi pi    4096 Nov  4 15:22 ..
-rw-r--r--  1 pi pi 4209097 Nov  4 15:21 2019-11-04_1521.jpg
-rw-r--r--  1 pi pi 3490959 Nov  4 15:25 2019-11-04_1525.jpg
-rw-r--r--  1 pi pi 4445823 Nov  4 15:26 2019-11-04_1526.jpg
-rw-r--r--  1 pi pi 4345377 Nov  4 15:27 2019-11-04_1527.jpg
-rw-r--r--  1 pi pi 4807345 Nov  4 15:14 .jpg
-rw-r--r--  1 pi pi    118 Nov  4 10:55 ledoff.py
-rw-r--r--  1 pi pi    119 Nov  4 10:54 ledon.py
-rw-r--r--  1 pi pi    186 Nov  4 12:31 ledpwm.py
-rwxr-xr-x  1 pi pi     87 Nov  4 15:19 rpicamera.sh
-rw-r--r--  1 pi pi    100 Nov  4 13:46 servo.py
-rw-r--r--  1 pi pi 4249279 Nov  4 14:53 test2.jpg
-rw-r--r--  1 pi pi 4303484 Nov  4 14:49 test.jpg
```

pi@raspberrypi:~/TrainingNU \$

Stop the crond

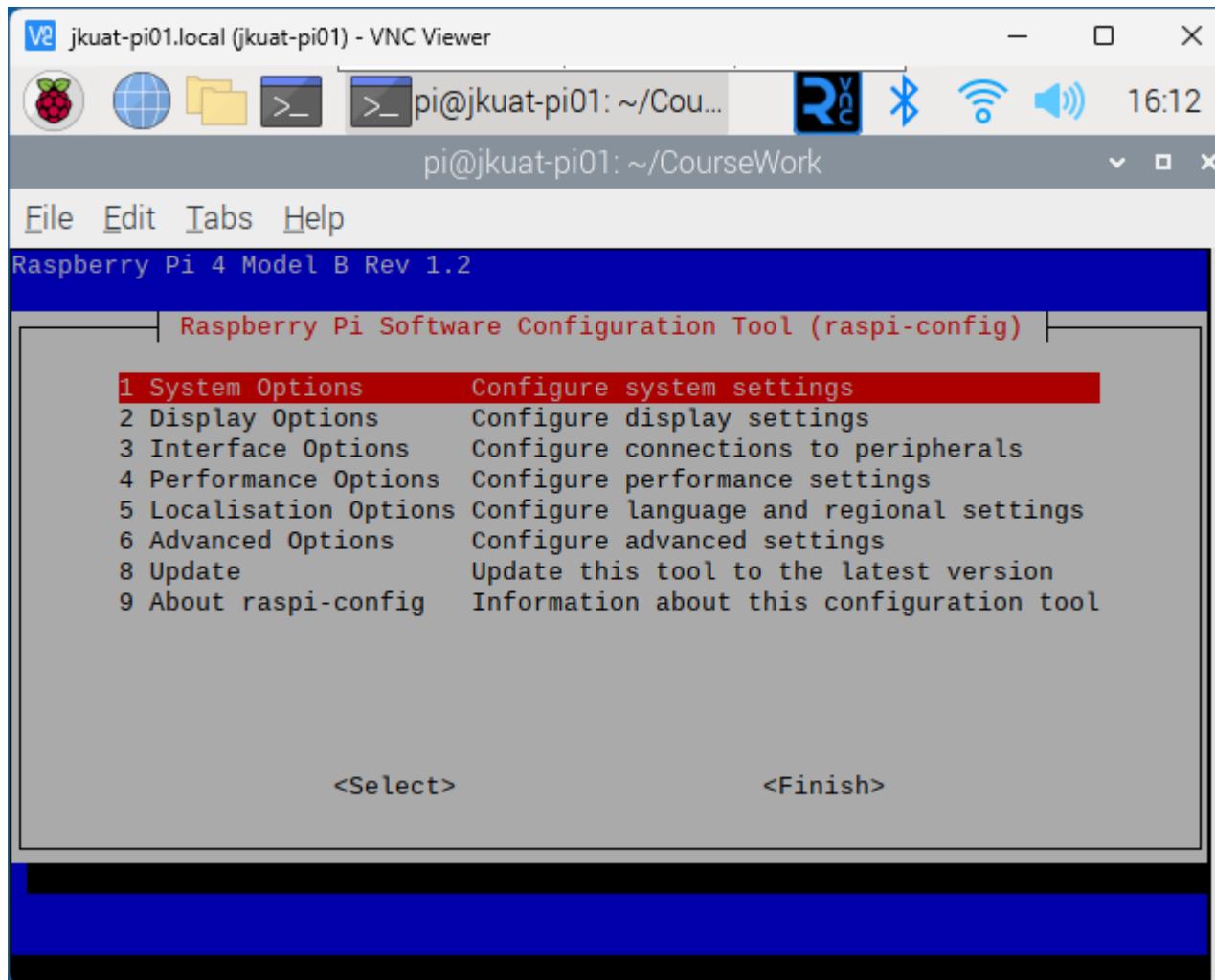
```
# */1 * * * * /home/pi/rpicamera.sh
```

means comment out

OpenCV

sudo raspi-config

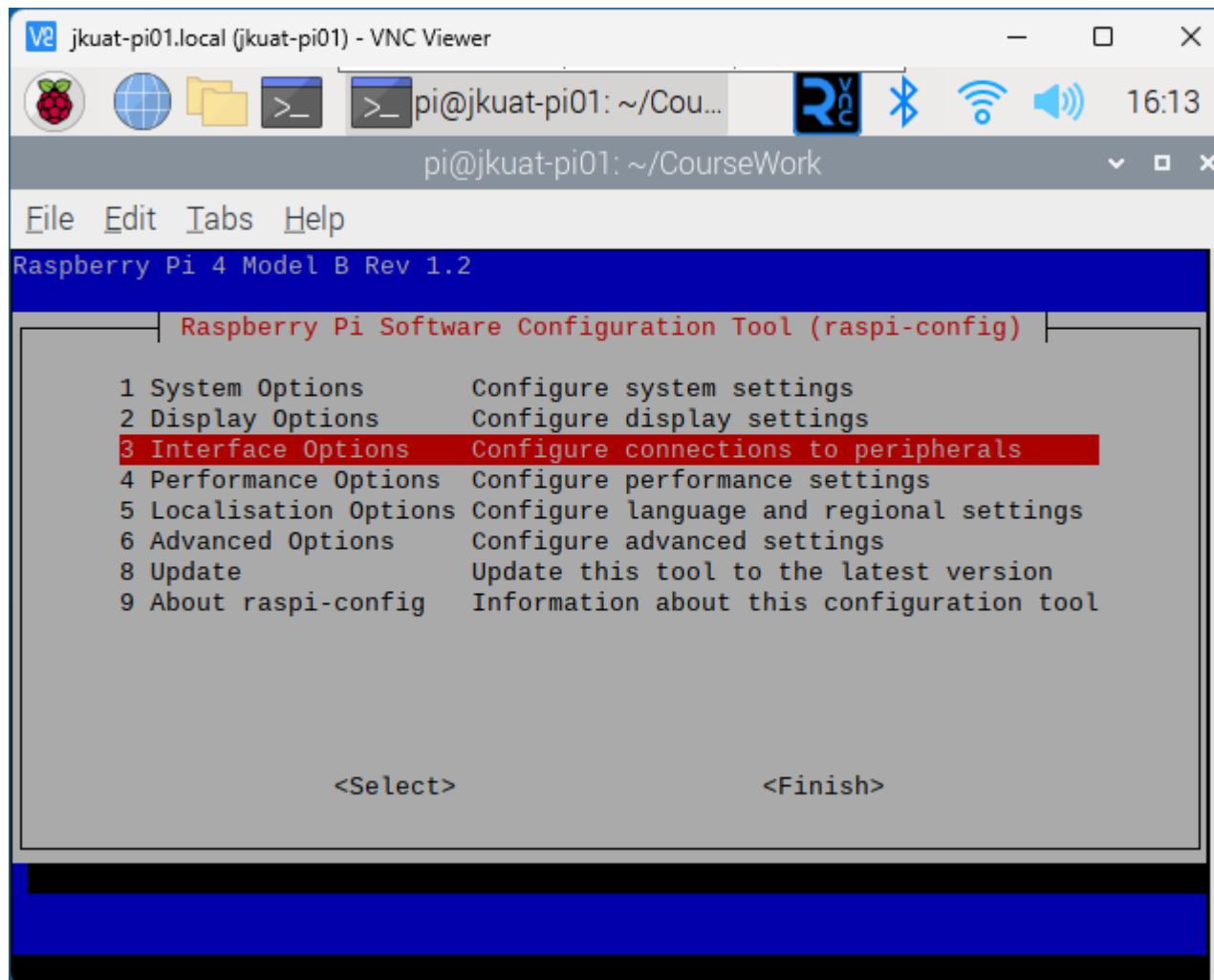
sudo raspi-config



sudo raspi-config

Select

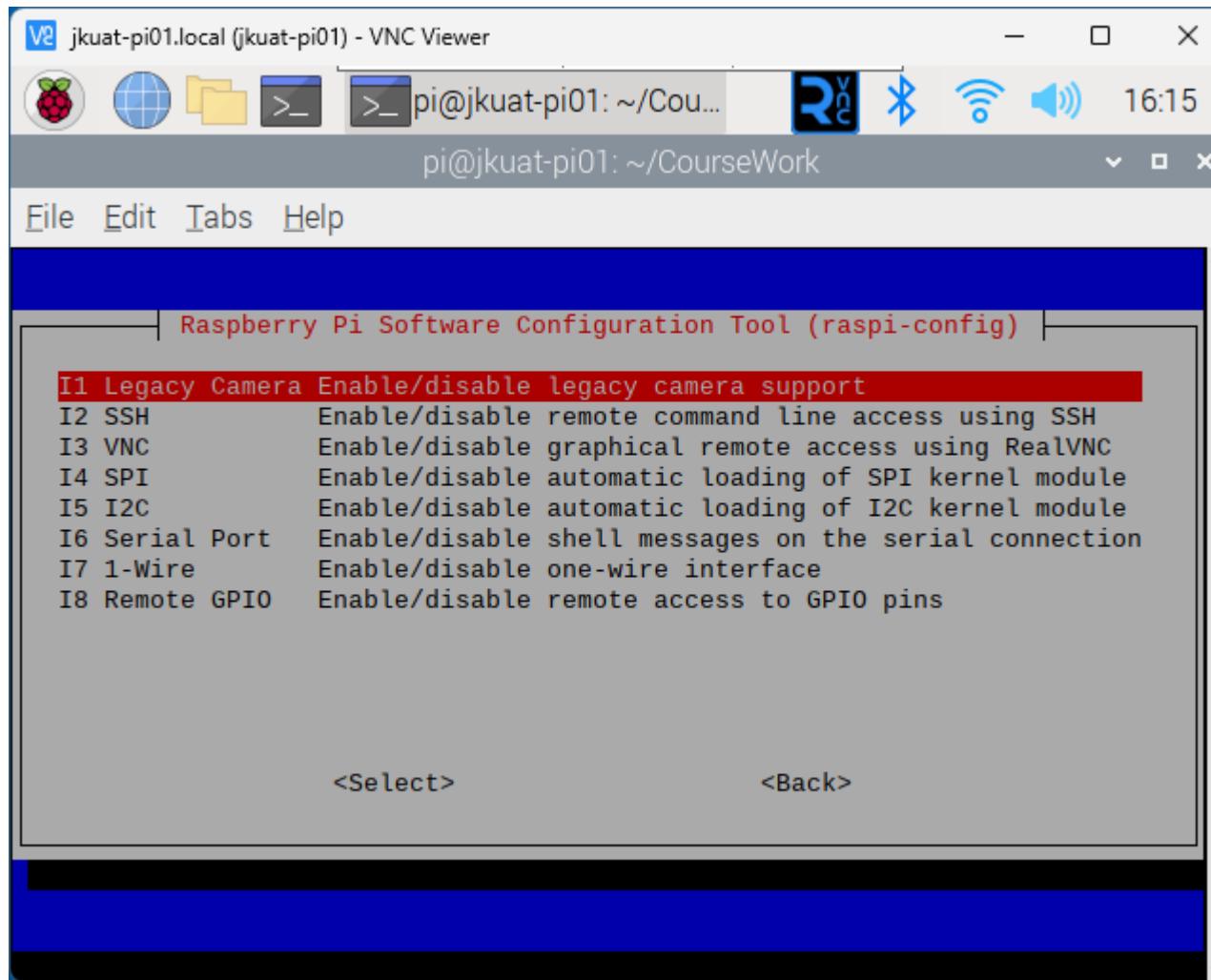
3 Interface Options



sudo raspi-config

Select

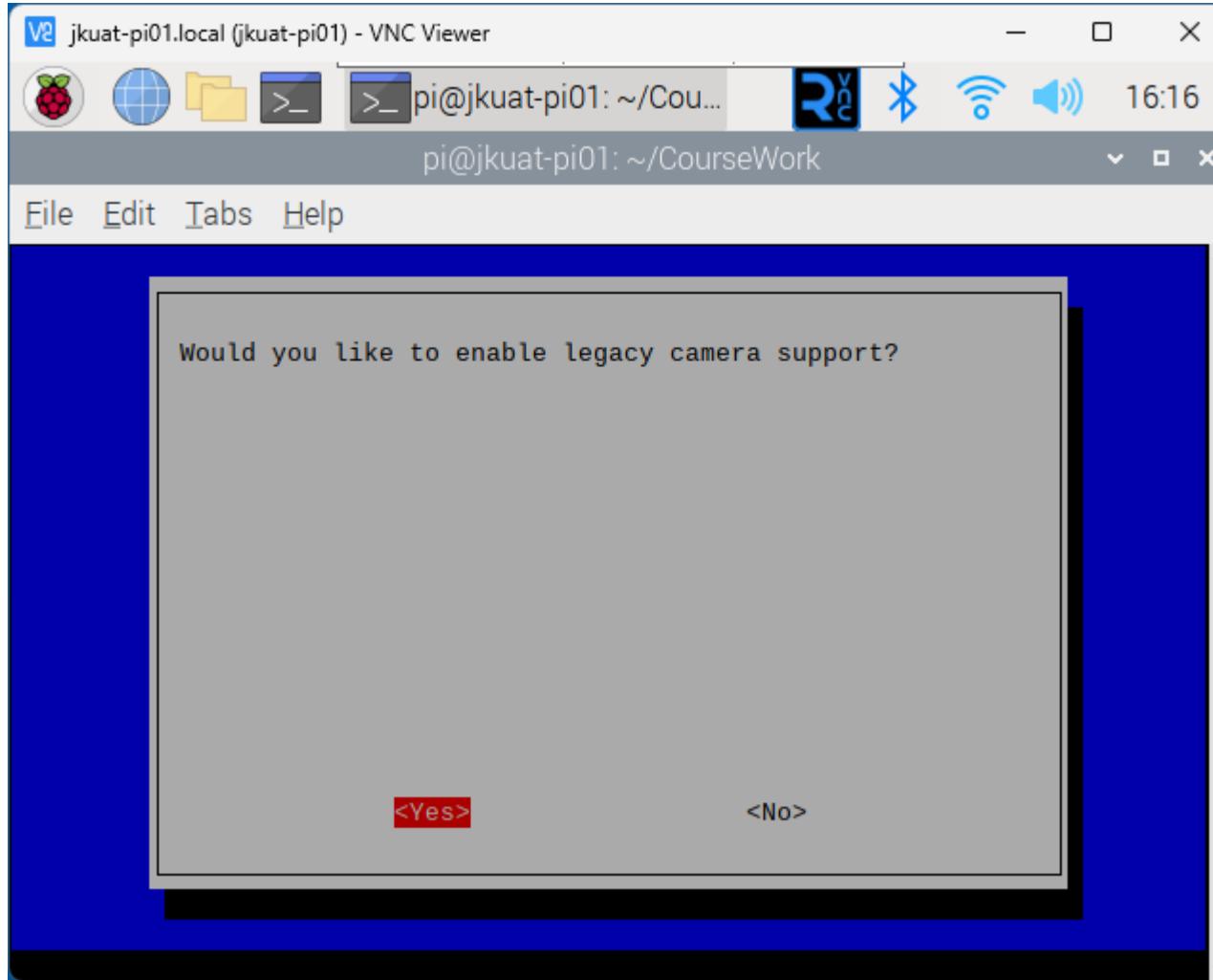
I1 Legacy Camera Enable/disable legacy camera support



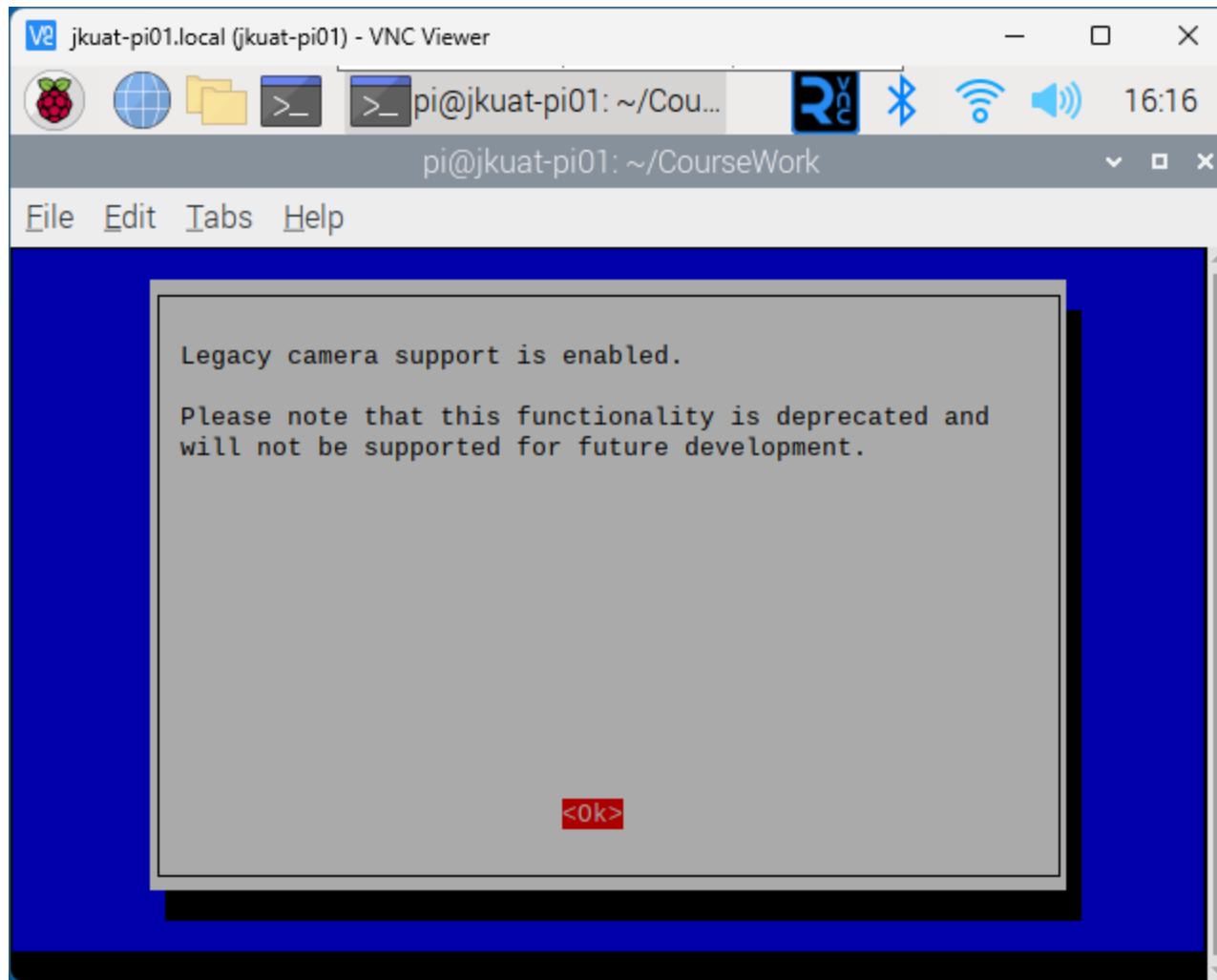
sudo raspi-config

Select

Yes

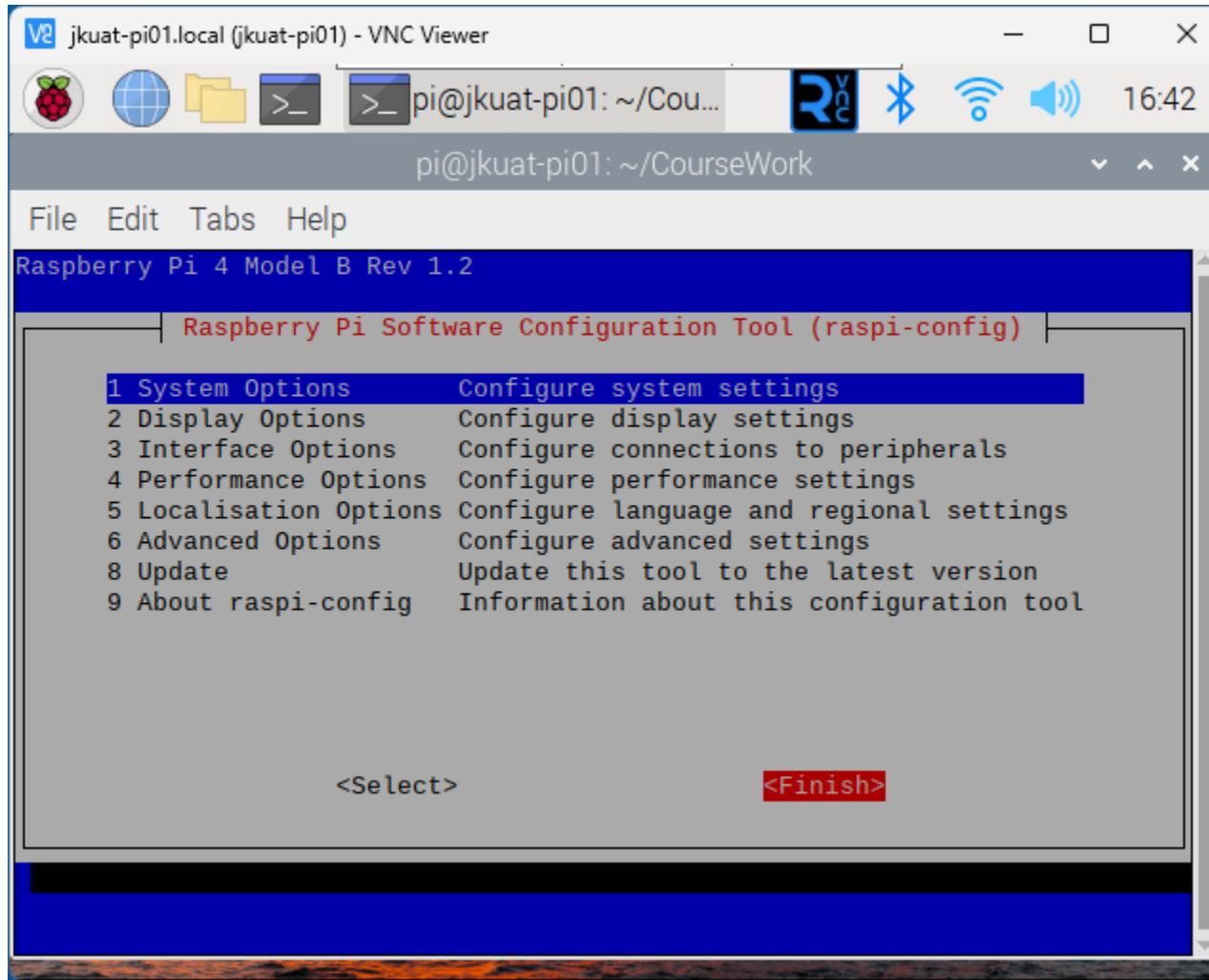


sudo raspi-config



sudo raspi-config

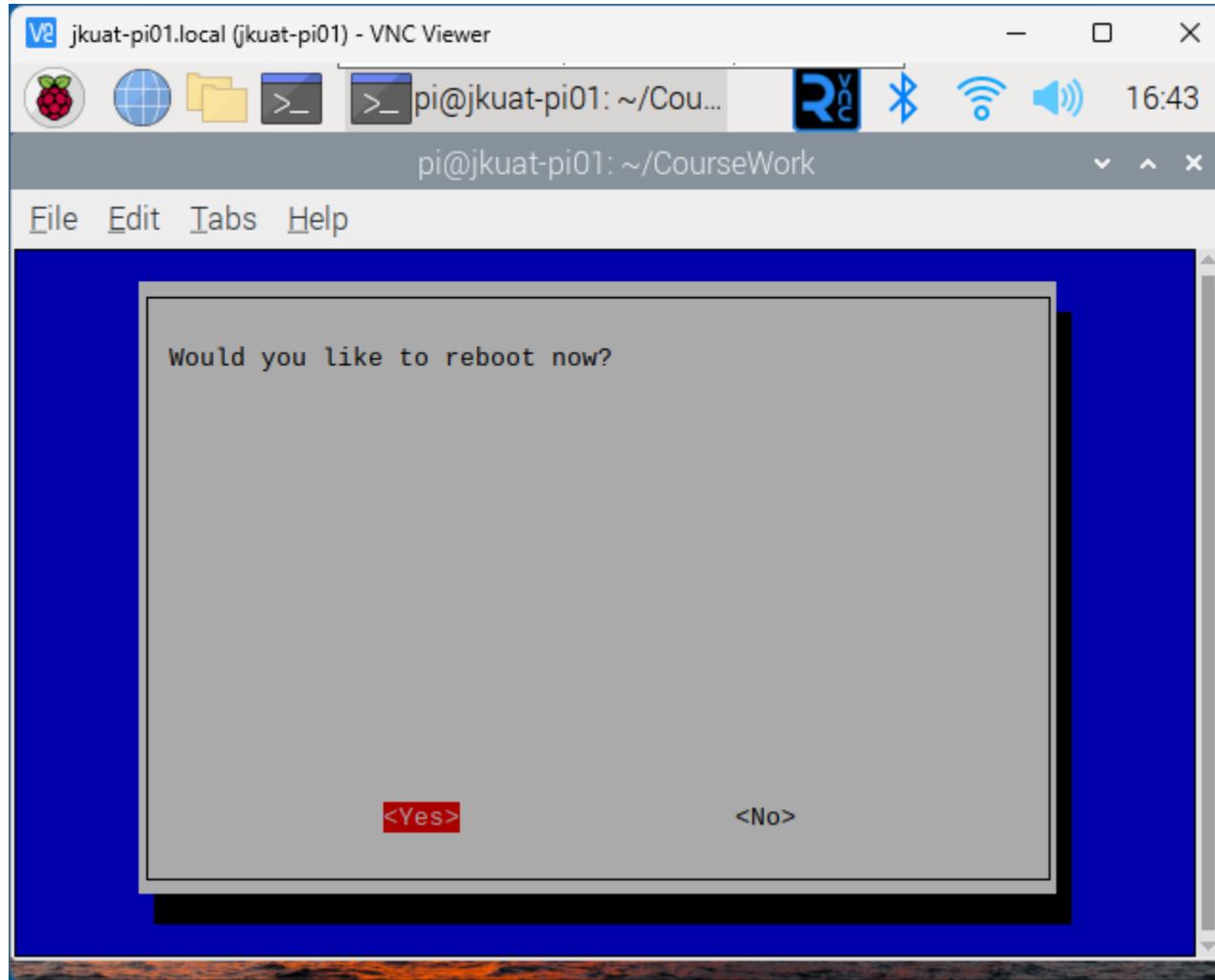
Select
Finish



sudo raspi-config

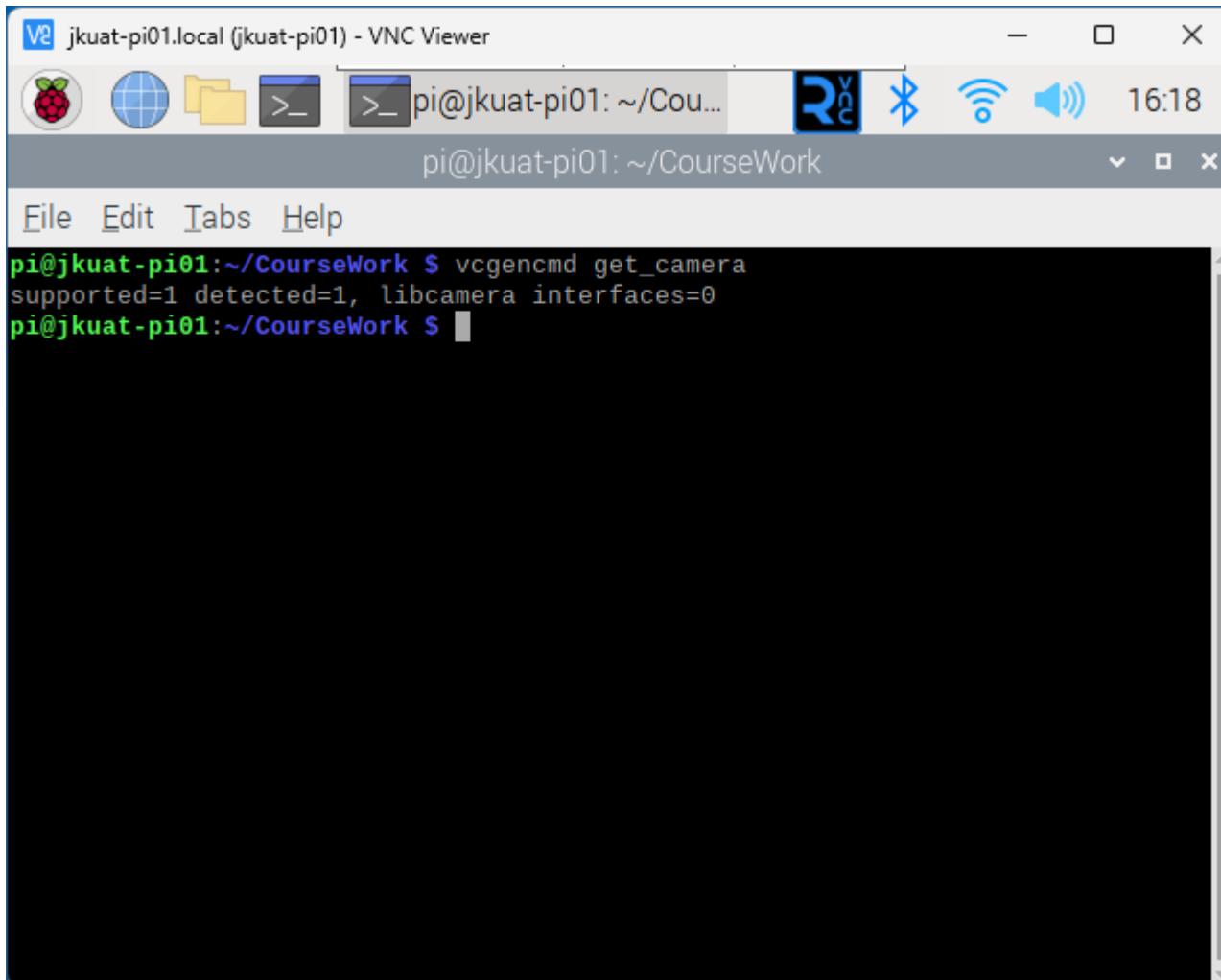
Select

Yes



vcgencmd get_camera

vcgencmd get_camera



The screenshot shows a VNC session titled "jkuat-pi01.local (jkuat-pi01) - VNC Viewer". The terminal window displays the command "vcgencmd get_camera" being run in a directory named "CourseWork". The output indicates that the camera is supported and detected.

```
pi@jkuat-pi01:~/CourseWork $ vcgencmd get_camera
supported=1 detected=1, libcamera interfaces=0
pi@jkuat-pi01:~/CourseWork $
```

Install OpenCV for facial recognition

```
# download from github
```

```
git clone https://github.com/kotamorishi/installOpenCV
```

```
cd installOpenCV
```

```
./installOpenCV.sh      #-----2 hours for installing-----
```

```
sudo python -m pip install --upgrade pip
```

```
pip install --upgrade opencv-python==4.6.0.66 # not use the latest version 4.8.0
```

```
sudo pip install numpy==1.24.0          # not use the latest version 1.26.0
```

```
#Check
```

```
python3
```

```
>>> import cv2    # please check no error message  
>>> CTRL+d
```

setup OpenCV for facial recognition

#-----run OpenCV-----

cd/facial_recognition

python3 headshots.py

more than 10 shoots for your face and stop by CTRL+c

Step #1

python3 train_model.py

training #generate encodings.pickle

Step #2

python3 facial_req.py

facial recognition

Step #3

q

force to quit

```
pi@jkuat-pi01:~/facial_recognition
File Edit Tabs Help
pi@jkuat-pi01:~/facial_recognition $ python3 headshots.py
Type your name: ProfessorSakagchi
dataset/ProfessorSakagchi/image_0.jpg written!
dataset/ProfessorSakagchi/image_1.jpg written!
dataset/ProfessorSakagchi/image_2.jpg written!
dataset/ProfessorSakagchi/image_3.jpg written!
dataset/ProfessorSakagchi/image_4.jpg written!
dataset/ProfessorSakagchi/image_5.jpg written!
dataset/ProfessorSakagchi/image_6.jpg written!
dataset/ProfessorSakagchi/image_7.jpg written!
dataset/ProfessorSakagchi/image_8.jpg written!
dataset/ProfessorSakagchi/image_9.jpg written!
^CTraceback (most recent call last):
  File "/home/pi/facial_recognition/headshots.py", line 21, in <module>
    ret, frame = cam.read()
KeyboardInterrupt
pi@jkuat-pi01:~/facial_recognition $
```

```
pi@jkuat-pi01:~/facial_recognition
File Edit Tabs Help
pi@jkuat-pi01:~/facial_recognition $ python3 train_model.py
[INFO] start processing faces...
[INFO] processing image 1/11
[INFO] processing image 2/11
[INFO] processing image 3/11
[INFO] processing image 4/11
[INFO] processing image 5/11
[INFO] processing image 6/11
[INFO] processing image 7/11
[INFO] processing image 8/11
[INFO] processing image 9/11
[INFO] processing image 10/11
[INFO] processing image 11/11
[INFO] serializing encodings...
pi@jkuat-pi01:~/facial_recognition $
```

1st step : python3 headshots.py

python3 headshots.py # more than 10 shoots for your face and stop by CTRL+c

```
pi@jkuat-pi01:~/facial_recognition
File Edit Tabs Help
pi@jkuat-pi01:~/facial_recognition $ python3 headshots.py
Type your name: ProfessorSakagchi
dataset/ProfessorSakagchi/image_0.jpg written!
dataset/ProfessorSakagchi/image_1.jpg written!
dataset/ProfessorSakagchi/image_2.jpg written!
dataset/ProfessorSakagchi/image_3.jpg written!
dataset/ProfessorSakagchi/image_4.jpg written!
dataset/ProfessorSakagchi/image_5.jpg written!
dataset/ProfessorSakagchi/image_6.jpg written!
dataset/ProfessorSakagchi/image_7.jpg written!
dataset/ProfessorSakagchi/image_8.jpg written!
dataset/ProfessorSakagchi/image_9.jpg written!
^CTraceback (most recent call last):
  File "/home/pi/facial_recognition/headshots.py", line 21, in <module>
    ret, frame = cam.read()
KeyboardInterrupt
pi@jkuat-pi01:~/facial_recognition $
```

Put your name!

Try shoot your face!
More than 10 shoot.

Stop by CTR+c

2nd step : python3 train_model.py

```
python3 train_model.py      # training      #generate encodings.pickle
```

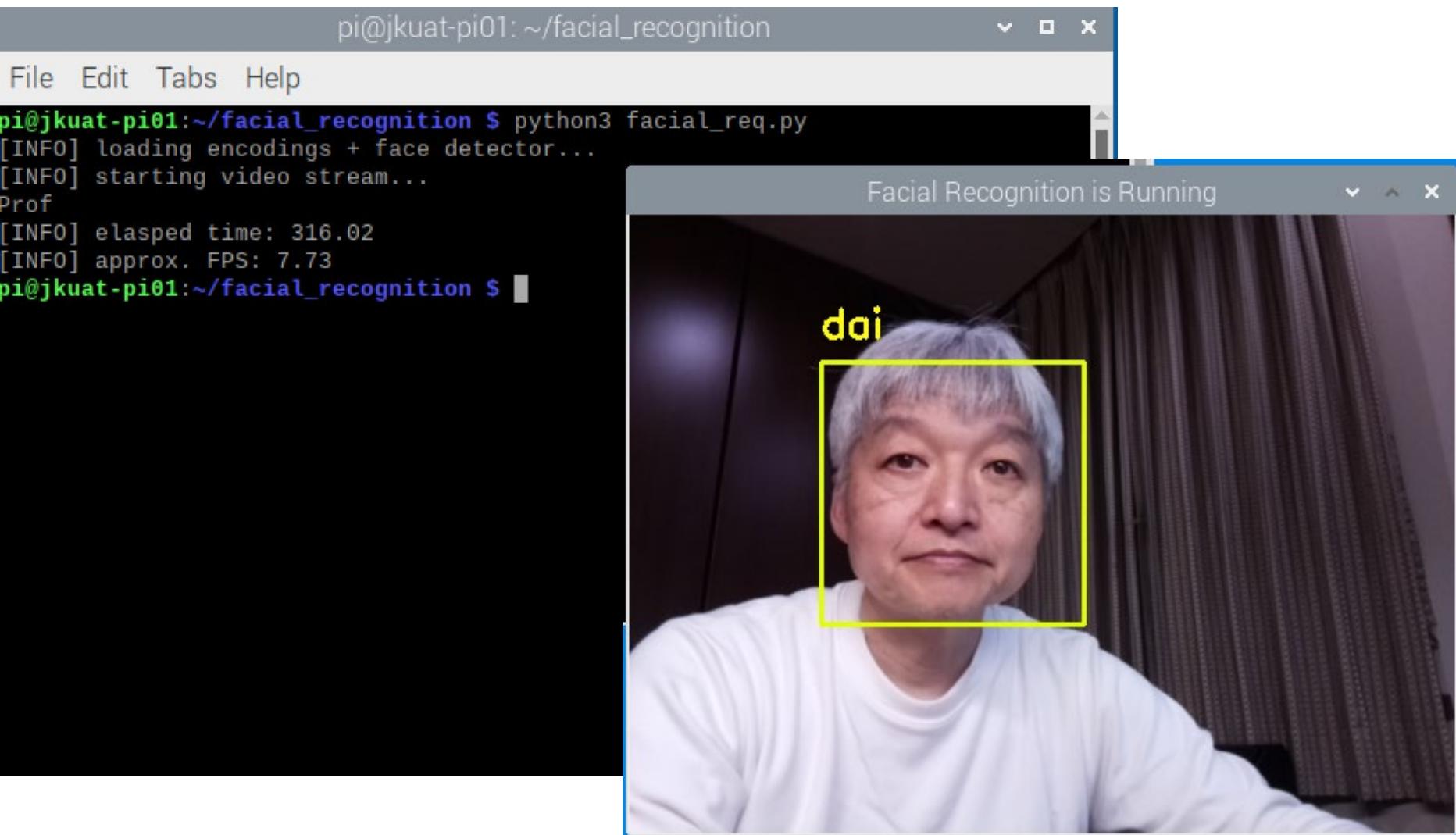
The screenshot shows a terminal window titled "pi@jkuat-pi01: ~/facial_recognition". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal prompt is "pi@jkuat-pi01:~/facial_recognition \$". The user has typed "python3 headshots.py" and is prompted to "Type your name: ProfessorSakagchi". A red box highlights the input "ProfessorSakagchi". To the right of the input, yellow text says "Put your name!". The terminal then lists ten files in the "dataset/ProfessorSakagchi/" directory, each ending in ".jpg" and labeled "written!". Below this, it shows a stack trace for a KeyboardInterrupt exception, indicating the file is "/home/pi/facial_recognition/headshots.py" and the line is 21. Finally, the prompt "pi@jkuat-pi01:~/facial_recognition \$" is shown again.

```
pi@jkuat-pi01:~/facial_recognition $ python3 headshots.py
Type your name: ProfessorSakagchi
Put your name!
dataset/ProfessorSakagchi/image_0.jpg written!
dataset/ProfessorSakagchi/image_1.jpg written!
dataset/ProfessorSakagchi/image_2.jpg written!
dataset/ProfessorSakagchi/image_3.jpg written!
dataset/ProfessorSakagchi/image_4.jpg written!
dataset/ProfessorSakagchi/image_5.jpg written!
dataset/ProfessorSakagchi/image_6.jpg written!
dataset/ProfessorSakagchi/image_7.jpg written!
dataset/ProfessorSakagchi/image_8.jpg written!
dataset/ProfessorSakagchi/image_9.jpg written!
^CTraceback (most recent call last):
  File "/home/pi/facial_recognition/headshots.py", line 21, in <module>
    ret, frame = cam.read()
KeyboardInterrupt

pi@jkuat-pi01:~/facial_recognition $
```

3rd step : python3 facial_req.py

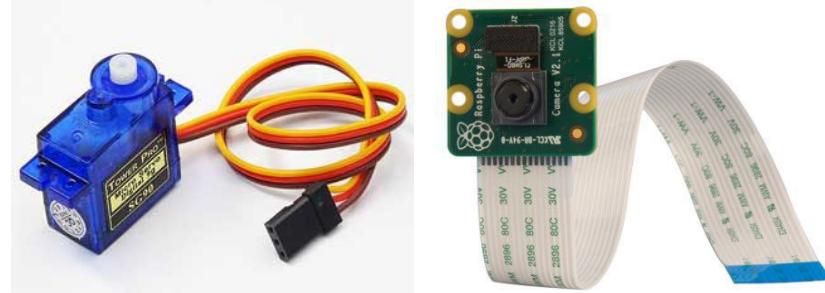
```
python3 facial_req.py      # facial recognition
```



Wrap up



1. LED flashing
2. Servo motor
3. Camera module
4. Cron daemon
5. OpenCV and Face recognition Demo

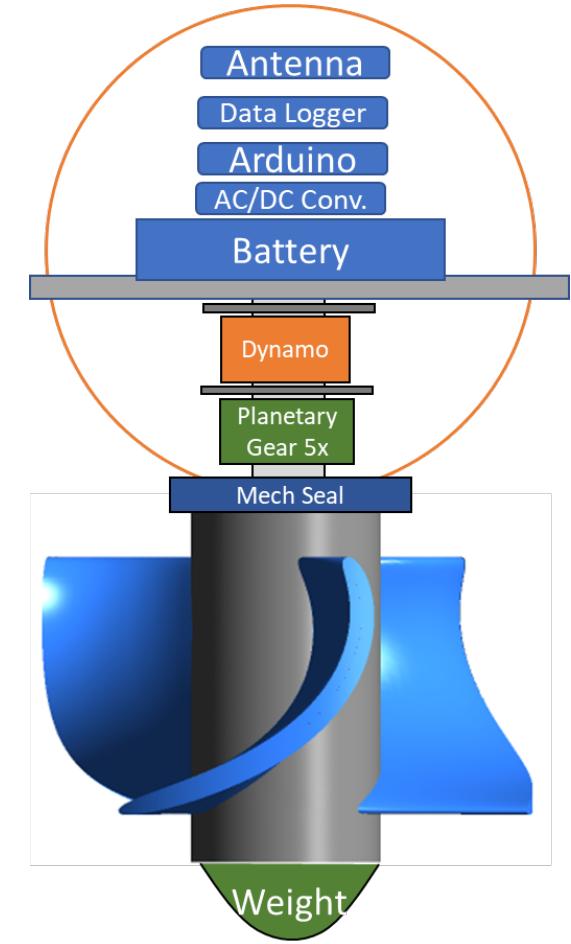
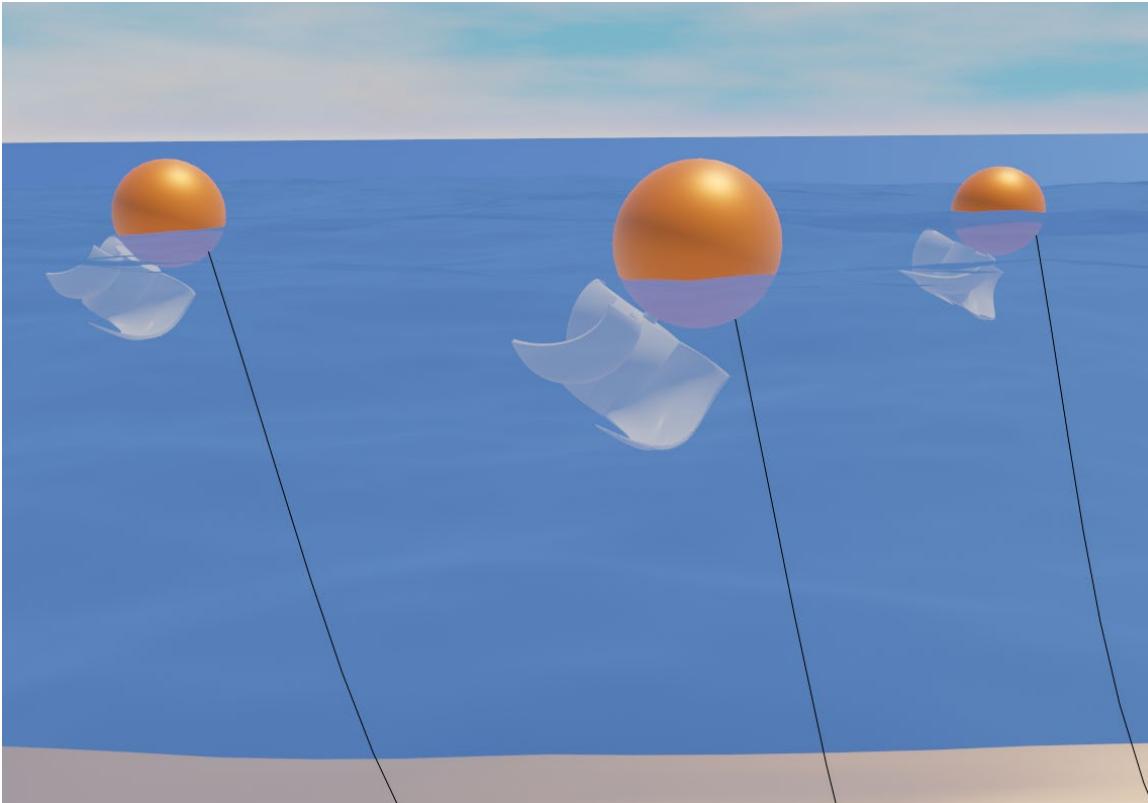


Next challenge

1. Control the Arduino Uno by Raspberry pi via I2C connection



Smart Buoy

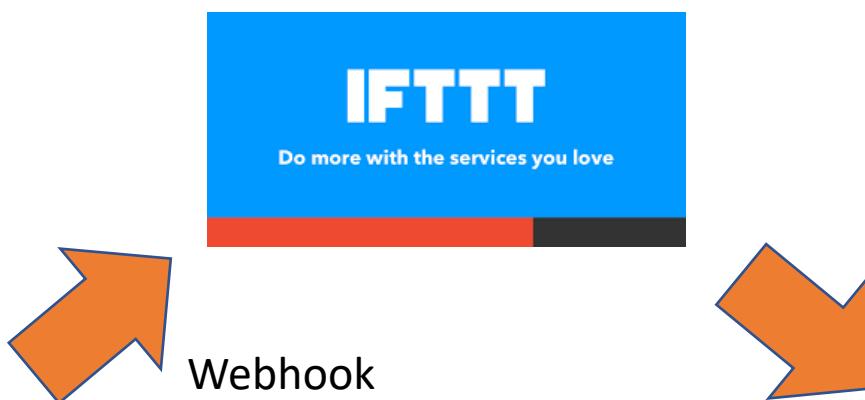


スマートブイ

IF + This Then That



Data sensing
Upload daemon by crond



Webhook
Receive the request and transfer
the data to GoogleSheet



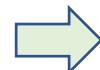
Google Drive

visualize your data by
Google sheet

LoggerIFTTT.py

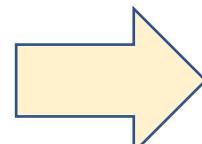
```
from sense_hat import SenseHat  
from datetime import datetime  
import time  
import requests
```

```
sense = SenseHat()
```



Initialization SenseHat

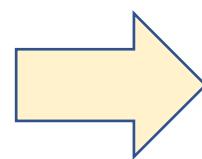
```
#sense.show_message("Start ")
```



```
temp = round(sense.get_temperature(),1)  
press = round(sense.get_pressure(),1)  
hum = round(sense.get_humidity(),1)
```

Sampling : Temp, Press, Humidity

```
orientation = sense.get_orientation_degrees()  
p=round(orientation["pitch"],1)  
r=round(orientation["roll"],1)  
y=round(orientation["yaw"],1)
```



Sampling : Angle of Yaw, Pitch and Roll

```
IFTTT_URL_GoogleSheets = 'https://maker.ifttt.com/trigger/REPORT1/with/key/'
```

```
IFTTT_KEY = 'm8yEHDspFvkF6o_OyujXRU6uFsqaGLR0EcjjirEs1Ed'
```

```
requests.post(IFTTT_URL_GoogleSheets+IFTTT_KEY, json={ 'value1': temp, 'value2' : press, 'value3' : hum})
```



Report

```
IFTTT_URL_GoogleSheets = 'https://maker.ifttt.com/trigger/REPORT2/with/key/'
```

```
IFTTT_KEY = 'm8yEHDspFvkF6o_OyujXRU6uFsqaGLR0EcjjirEs1Ed'
```

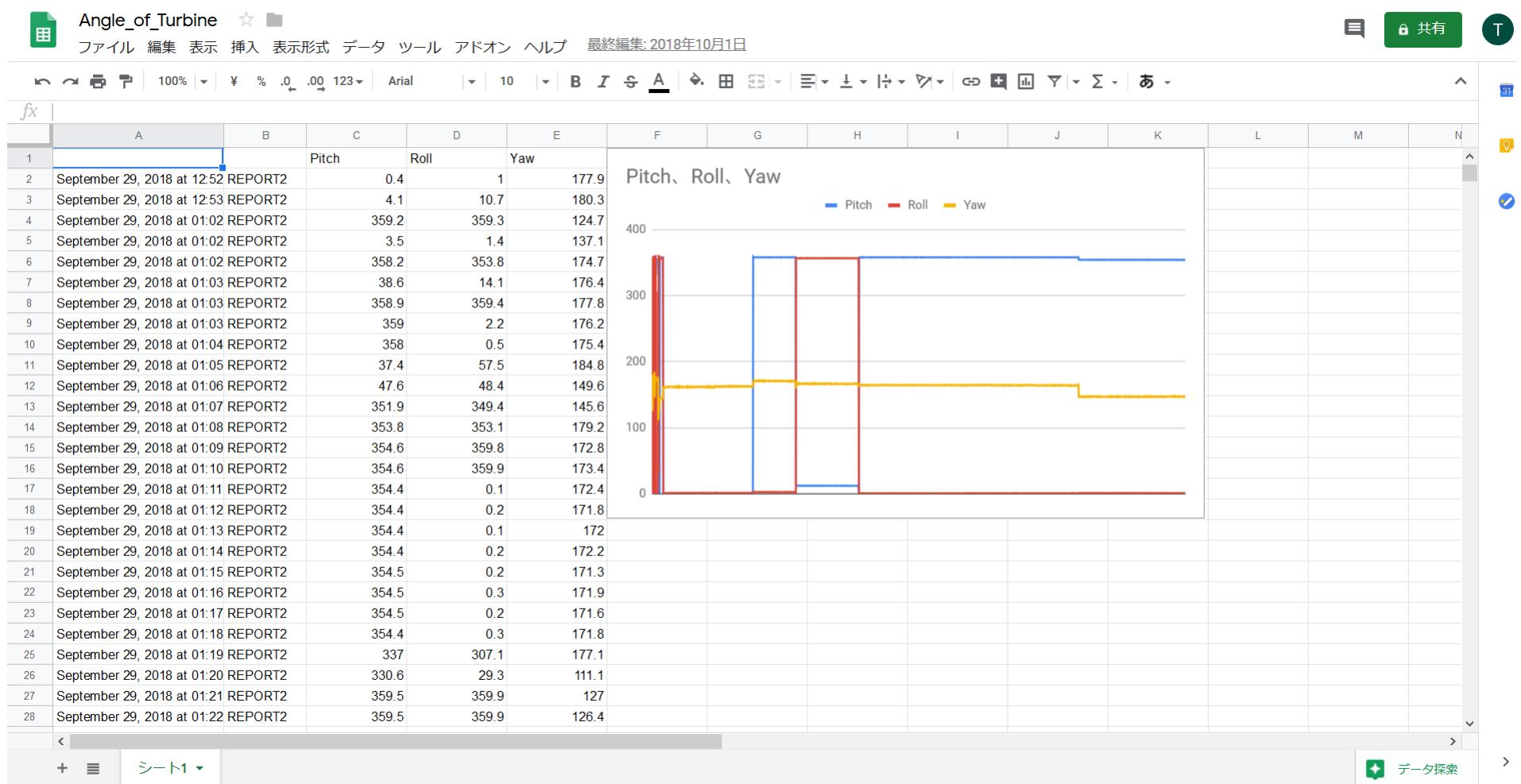
```
requests.post(IFTTT_URL_GoogleSheets+IFTTT_KEY, json={ 'value1': p, 'value2' : r, 'value3' : y})
```



Report2

```
sense.clear(255,255,255)  
time.sleep(1)  
sense.clear()
```

Data Plotting by Google spread sheet



Smart Mirror

Date Clock & Weather forecast



Raspberry Pi



Agenda

1. Install Raspbian OS (32bit Full)
2. LED flashing
3. Servo motor
4. Camera module
5. Cron daemon
6. OpenCV and Face Recognition Demo

Thank you

Contact info.:

Professor Daisaku Sakaguchi
e-mail : daisaku@nagasaki-u.ac.jp



Mechanical Science, Division of System Science
Graduate School of Engineering, Nagasaki University
1-14 Bunkyo-machi Nagasaki 852-8521 JAPAN