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

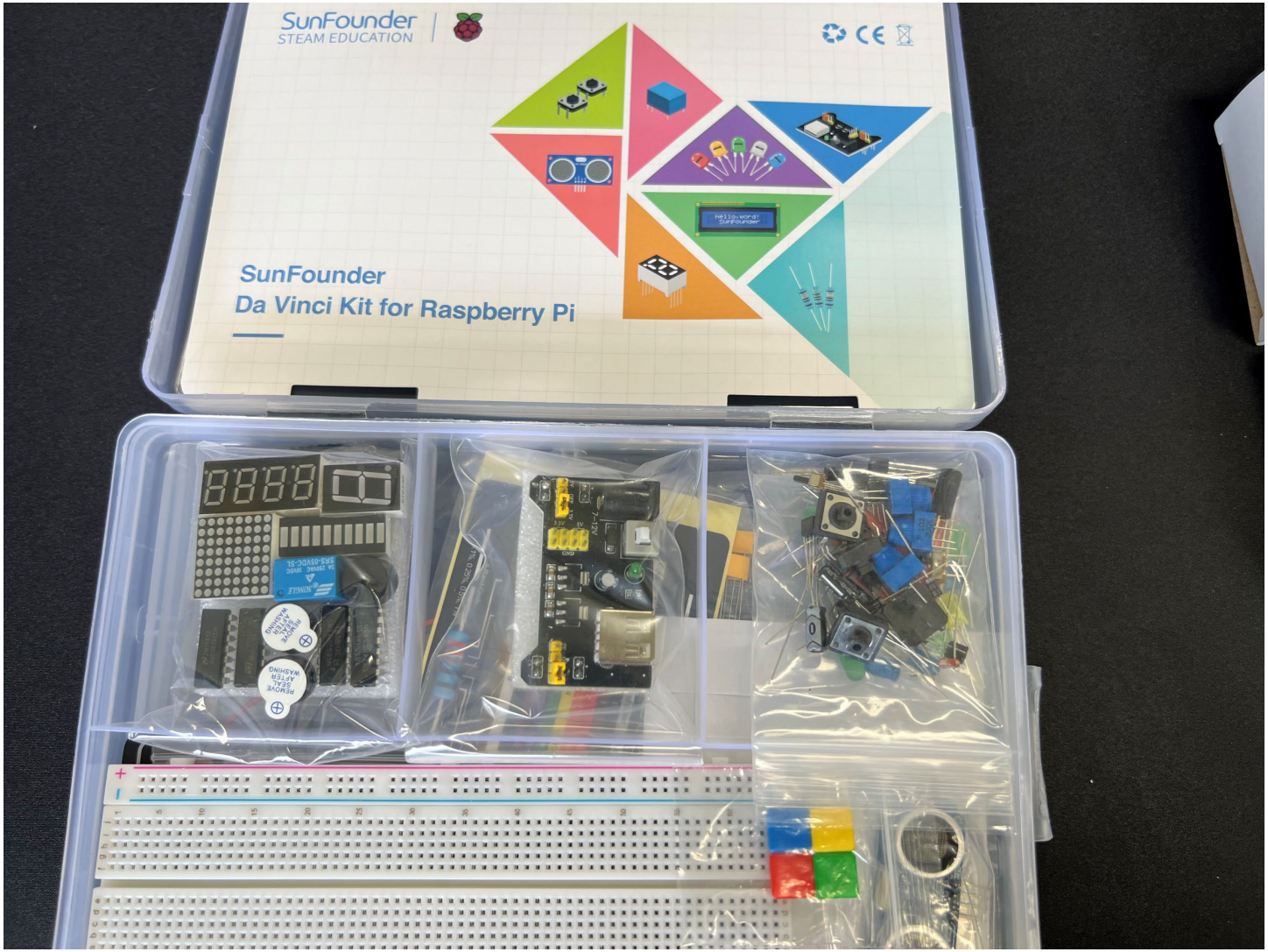
Agenda

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

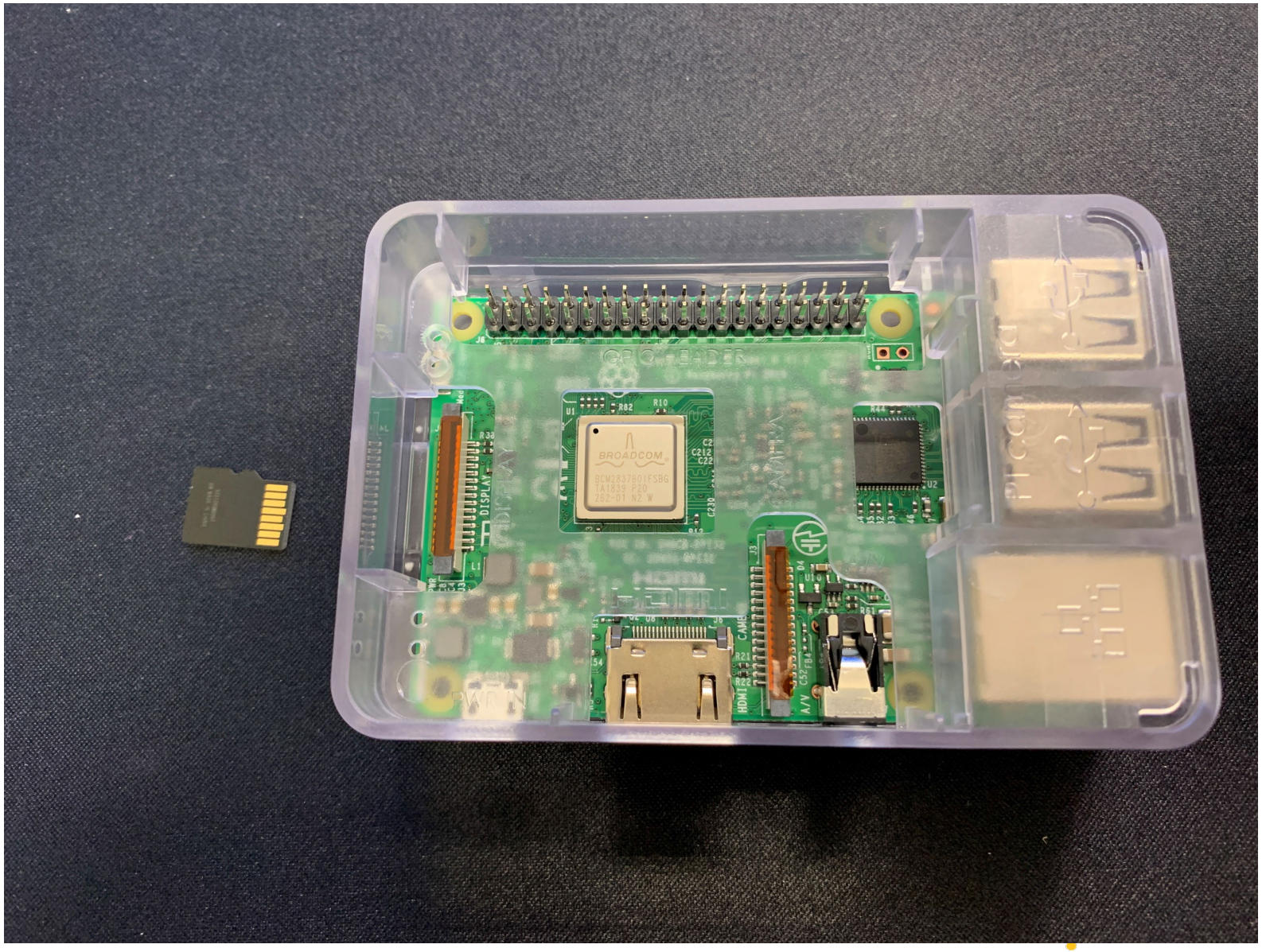
Handout

Nagasaki University

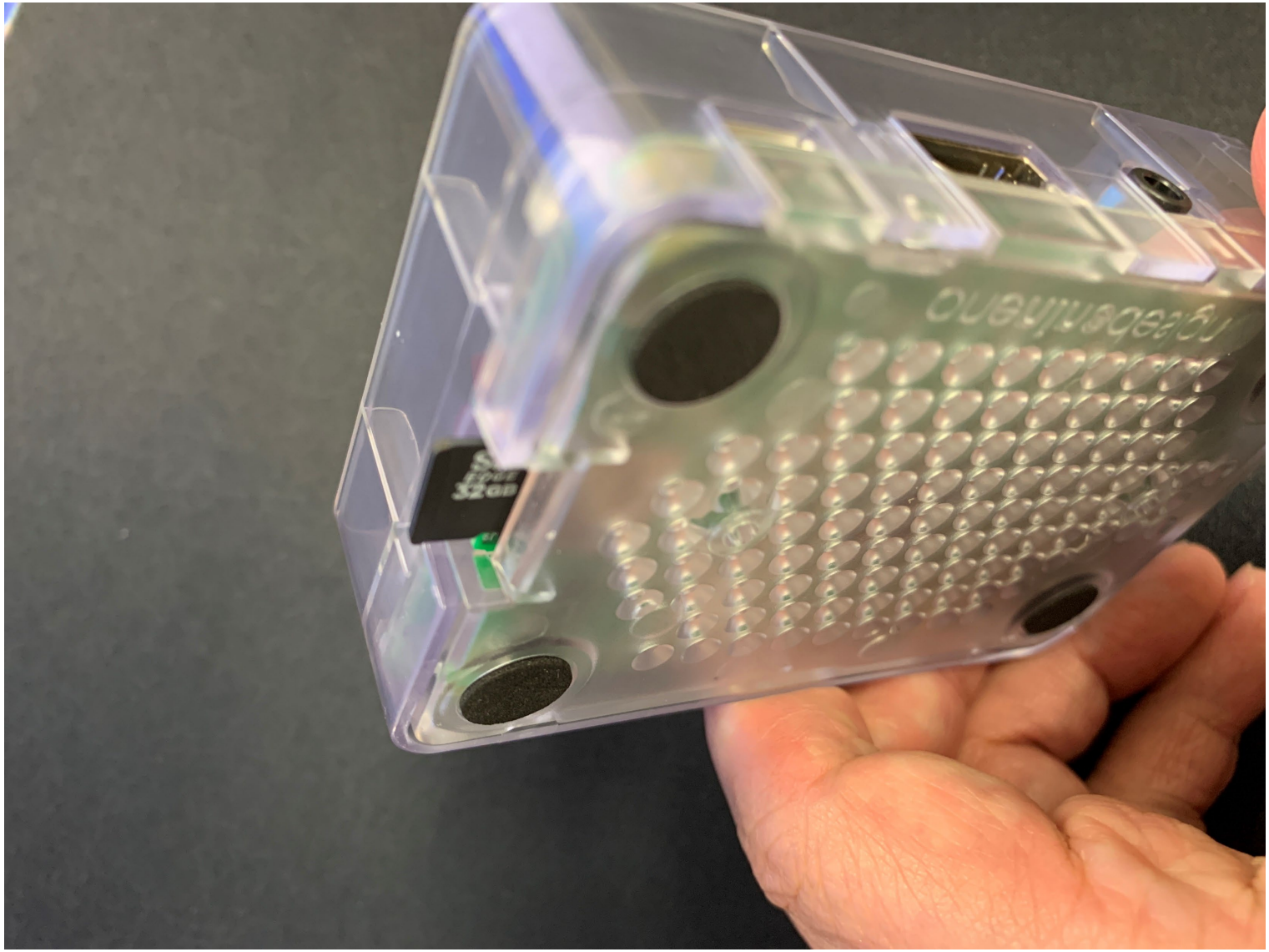
Handout


SunFounder
STEAM EDUCATION
SunFounder
Da Vinci Kit for Raspberry Pi

Insert the SD card

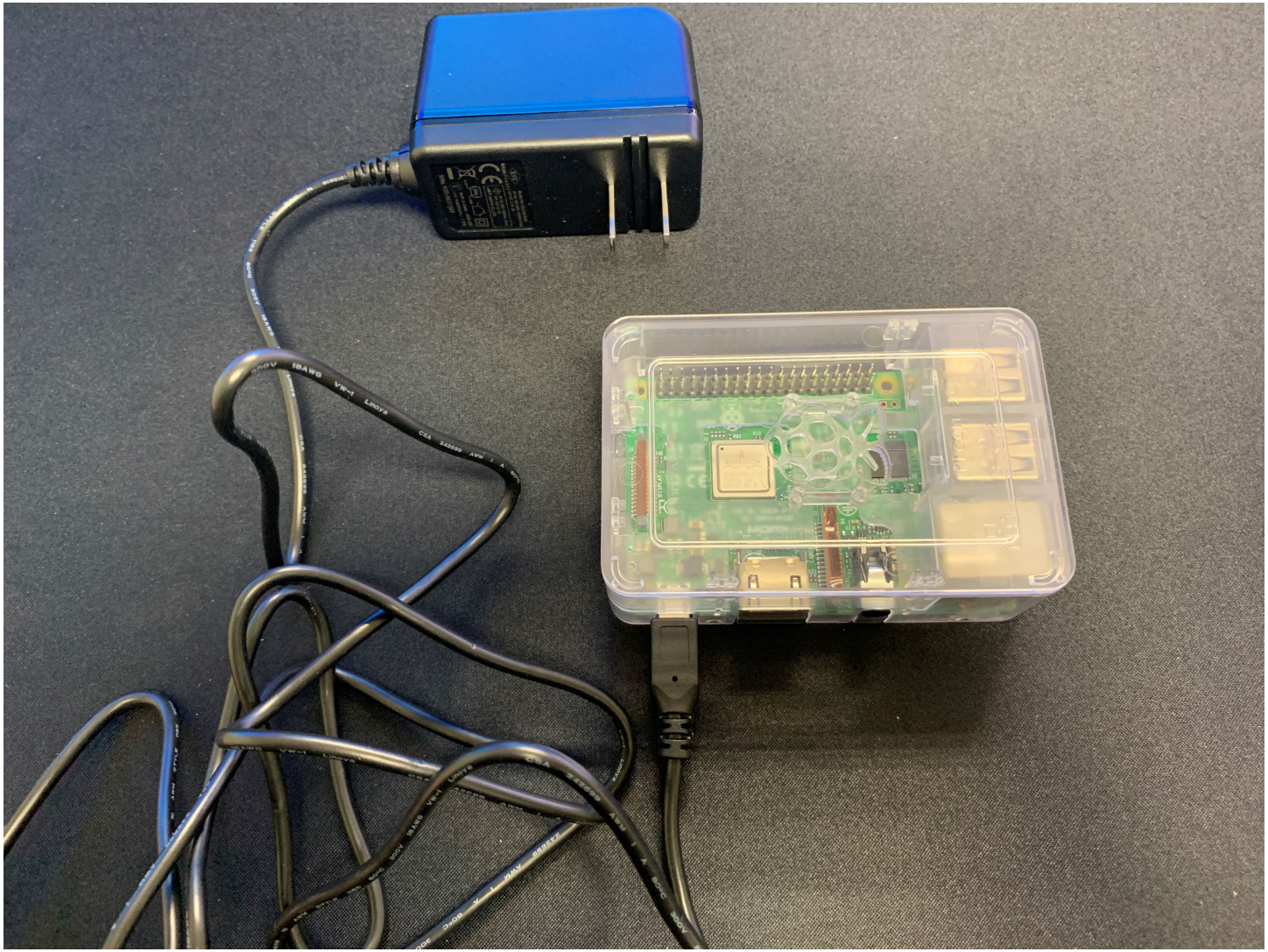


Insert the SD card

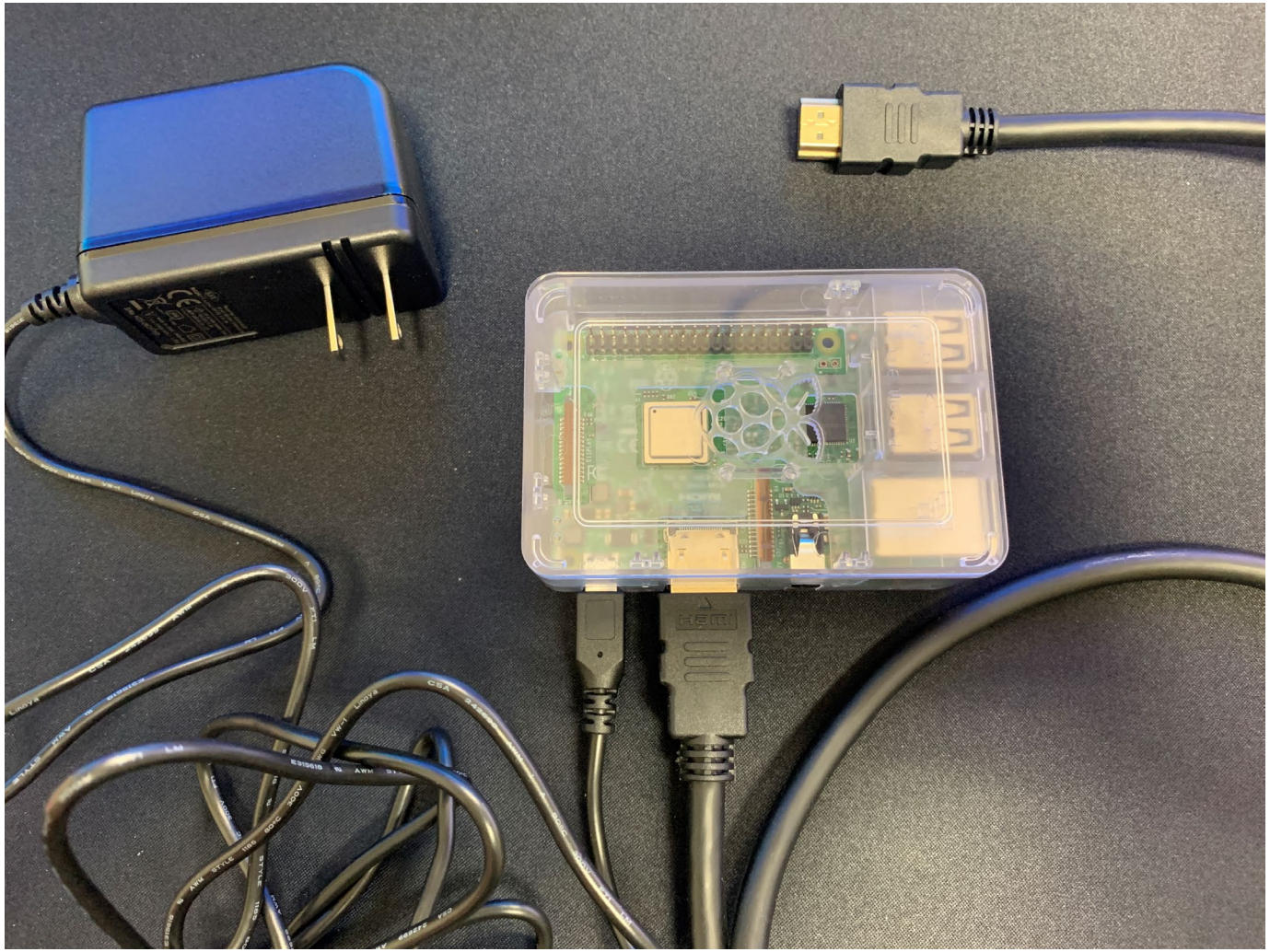


AC adapter

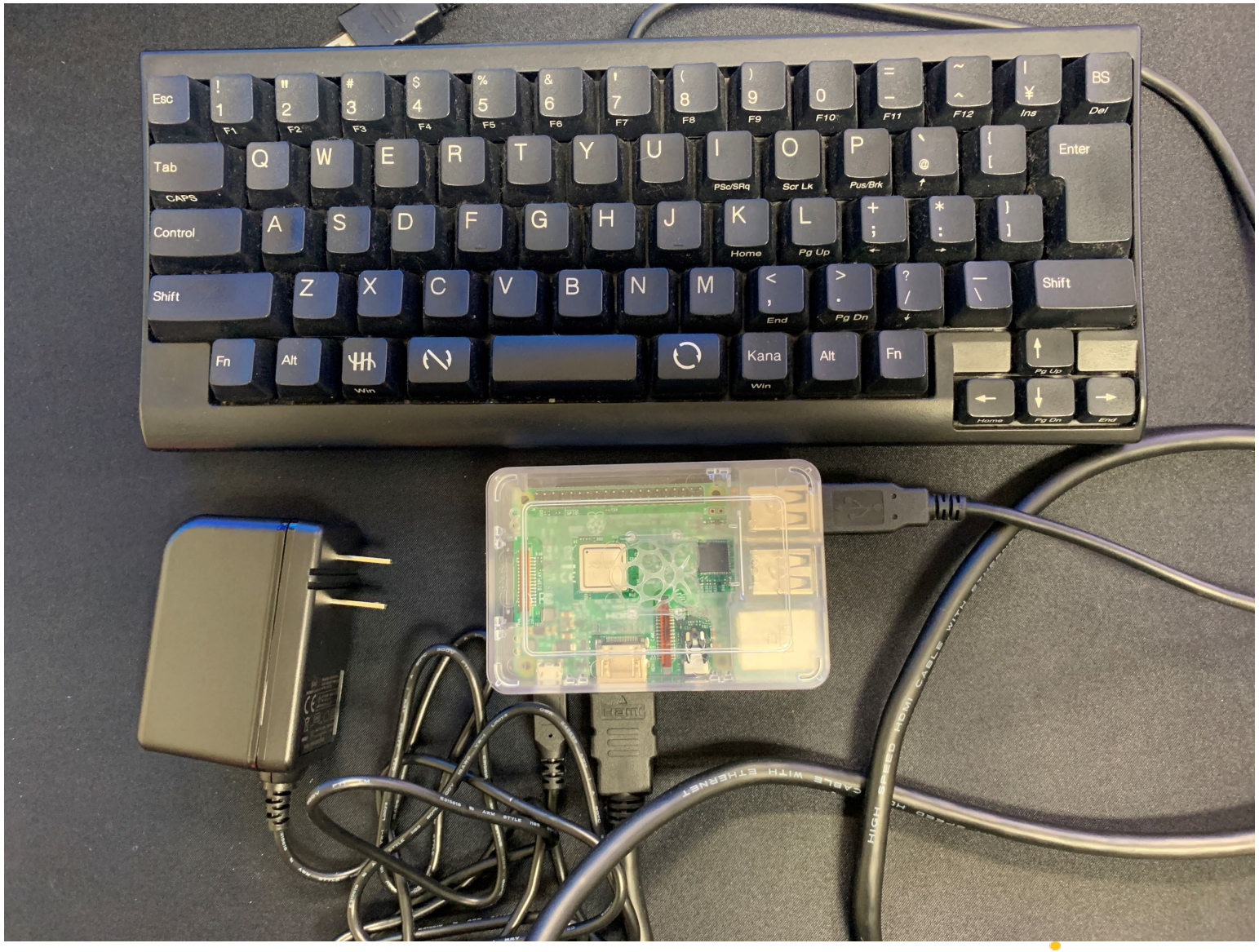
AC adapter



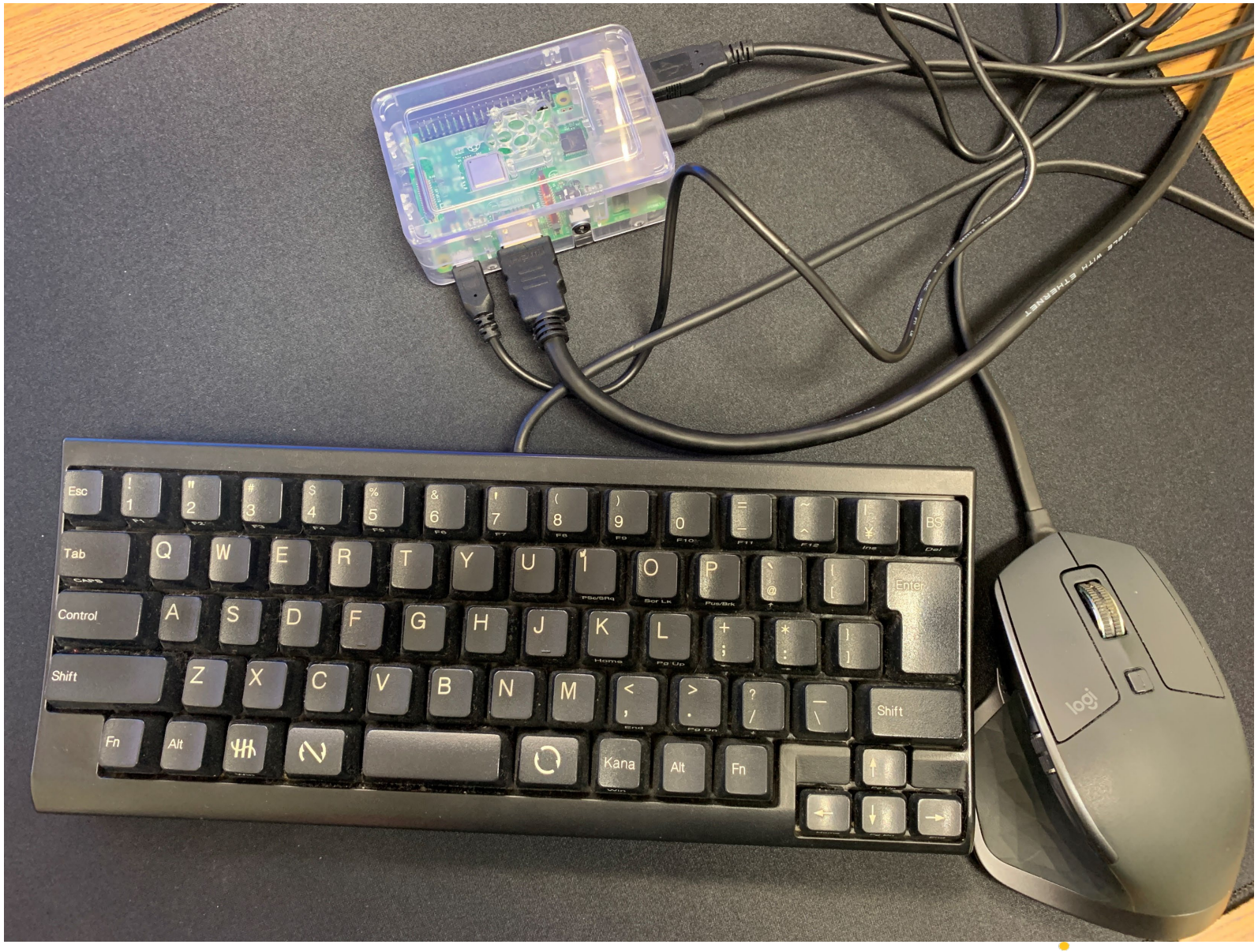
HDMI cable



USB Mouse and Keyboard

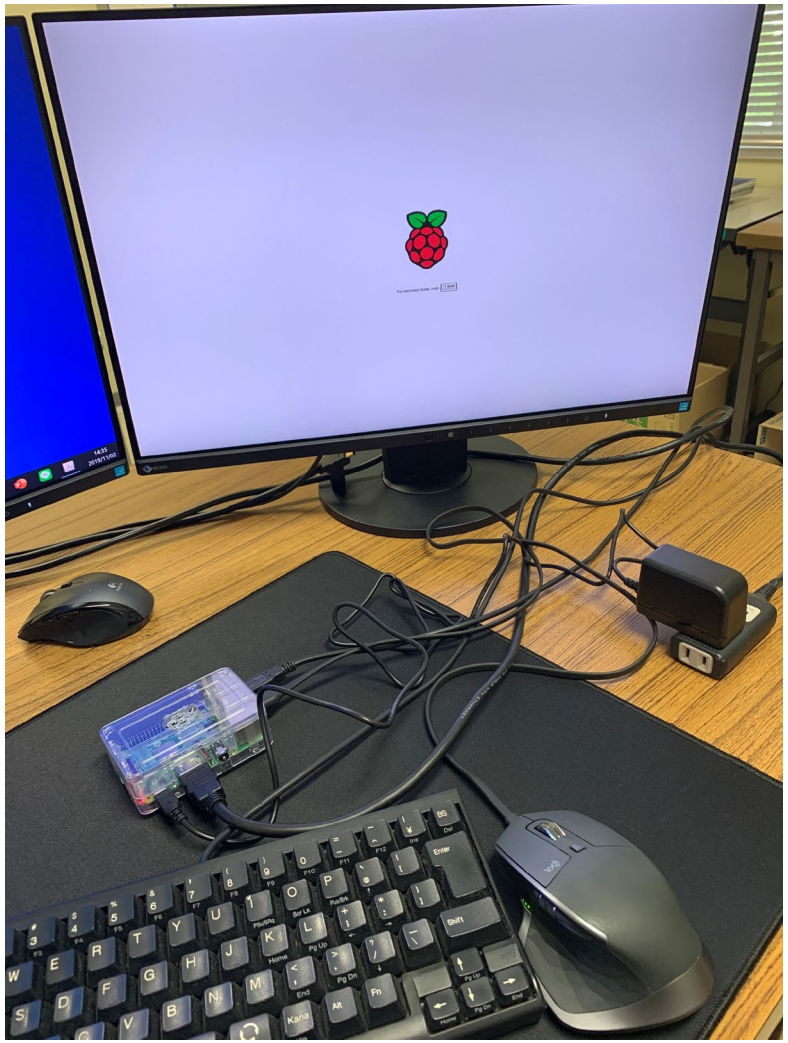


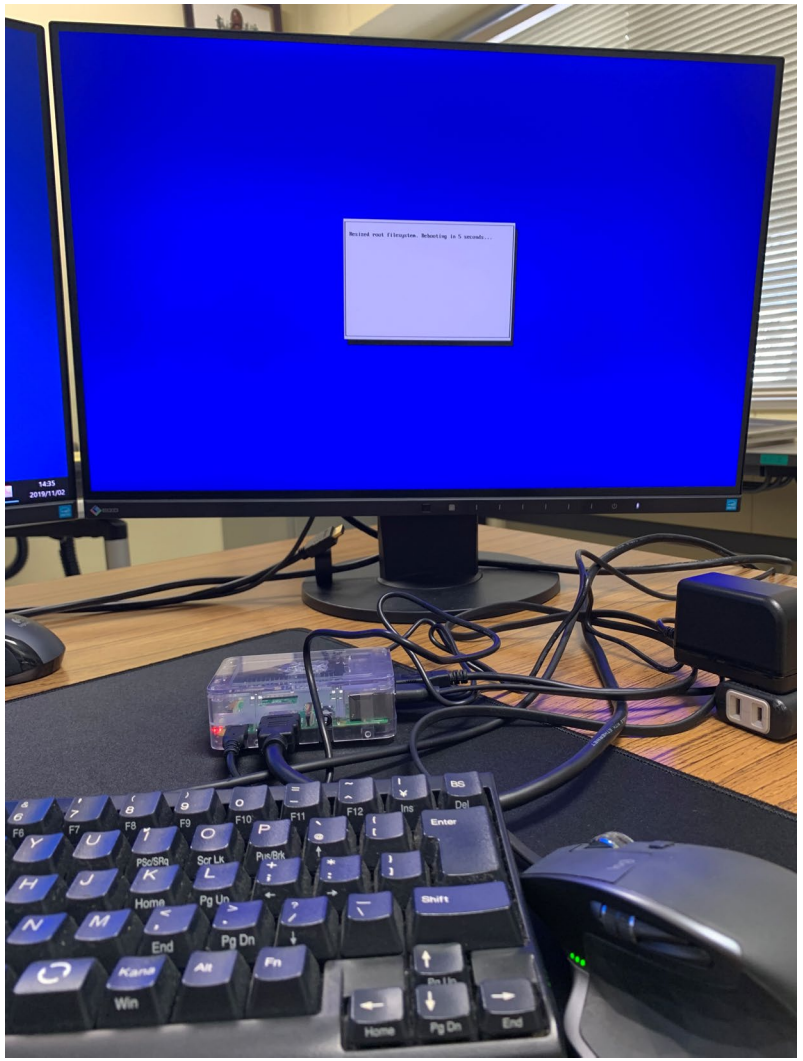
USB Mouse and Keyboard



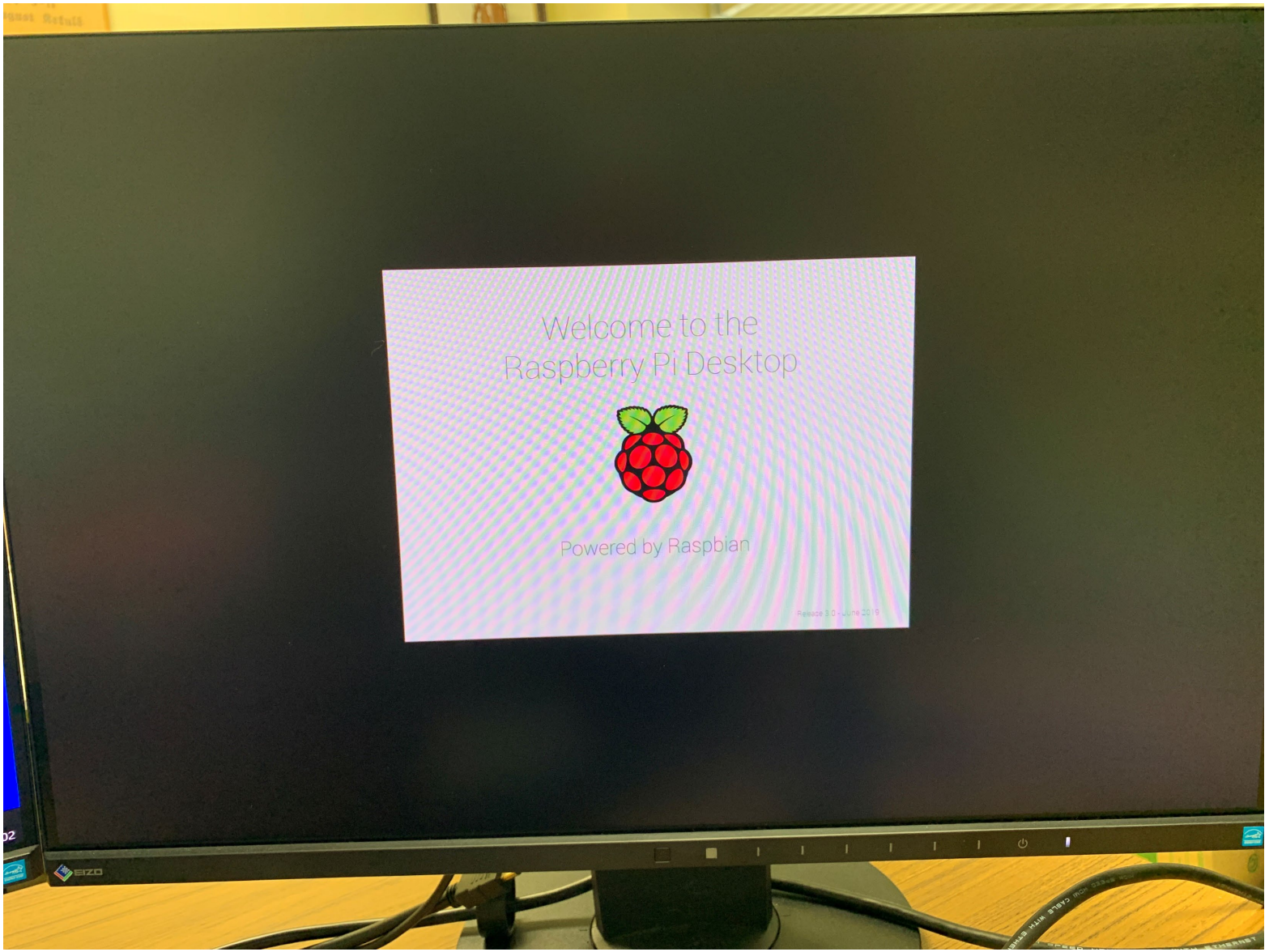
Nagasaki University

Power On

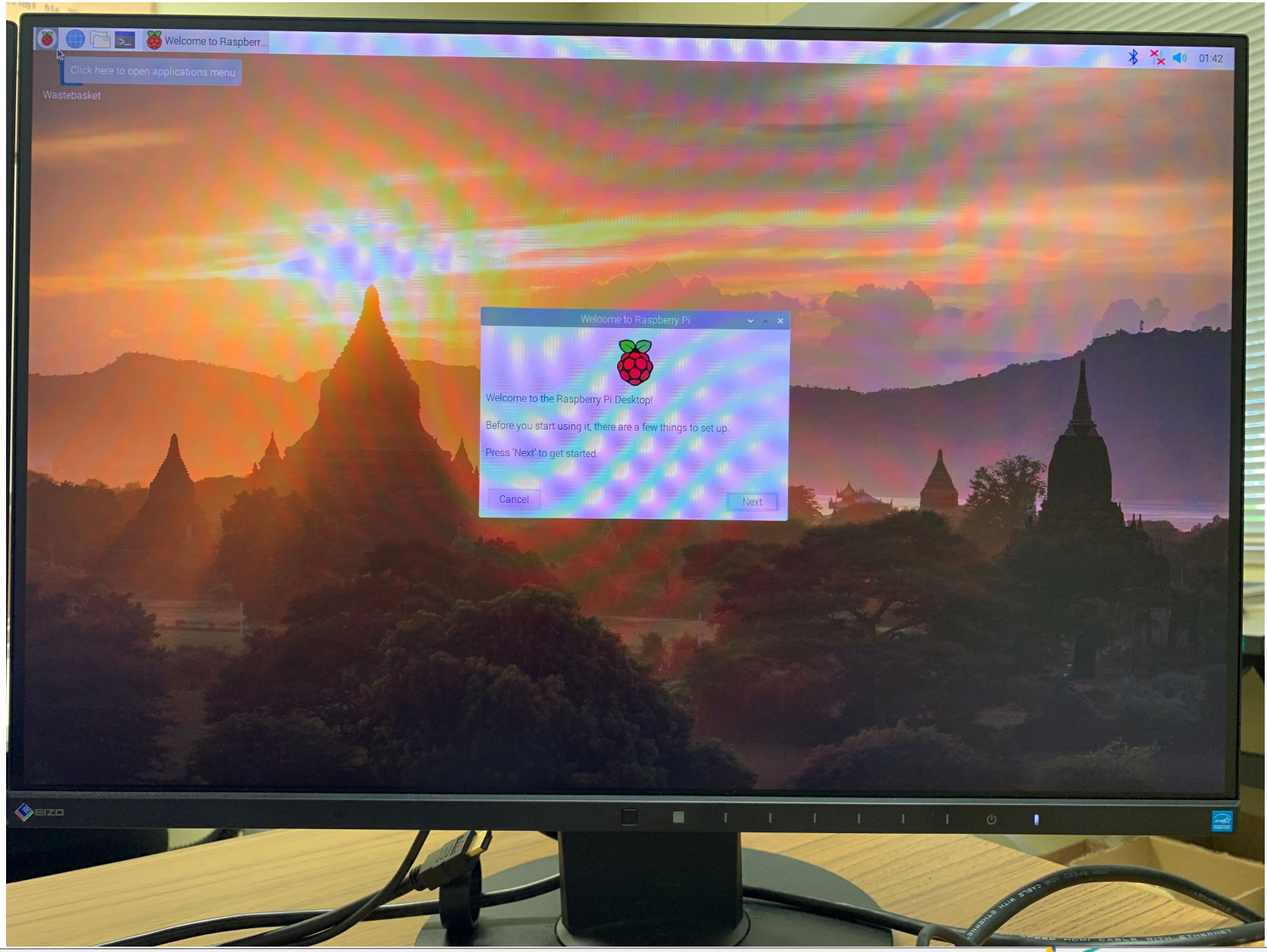




Boot up


Welcome to the
Raspberry Pi Desktop
Powered by Raspbian
EIZO

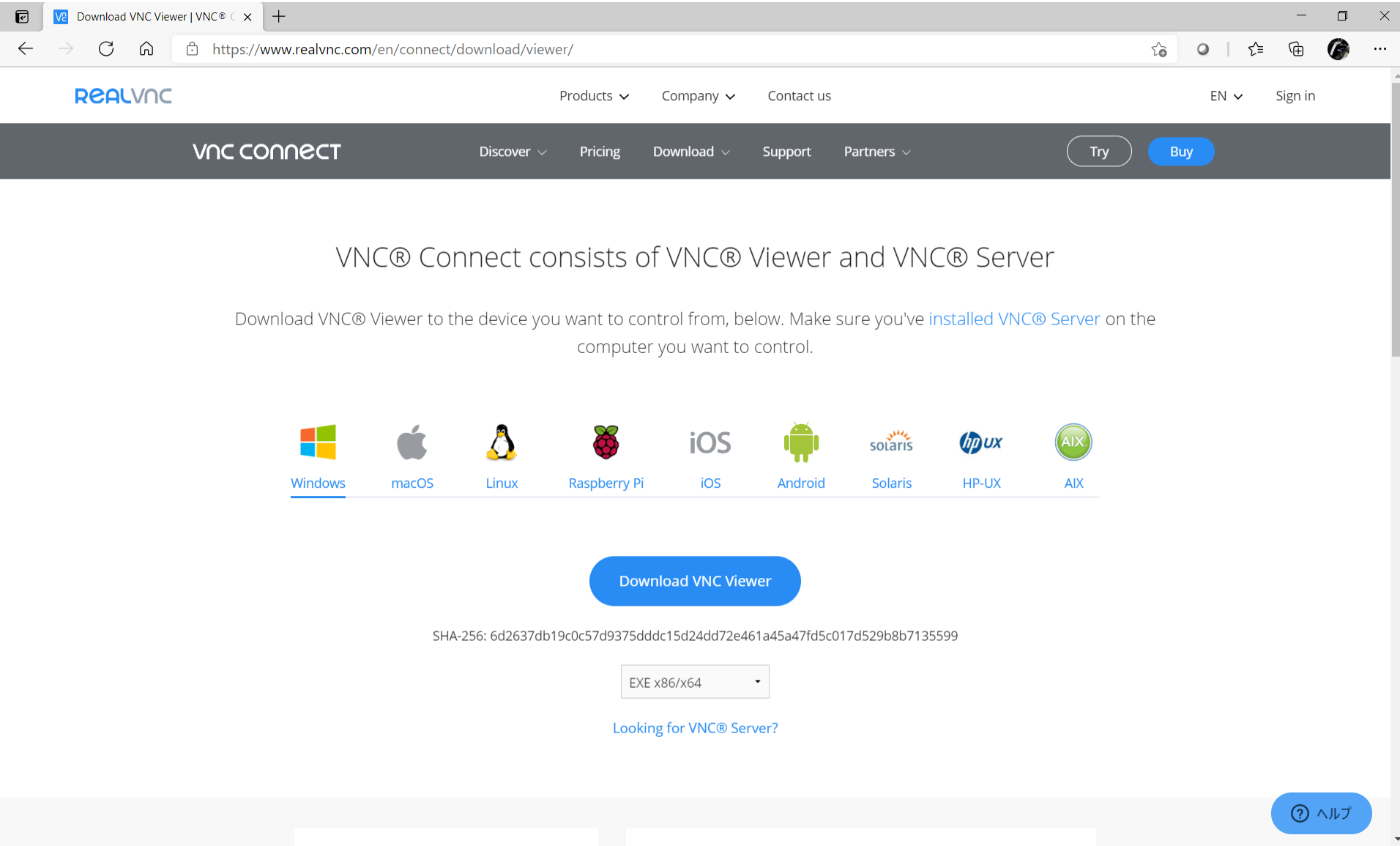
DESKTOP


Welcome to Raspberr...
Click here to open applications menu
Wastebasket
01:42
Welcome to Raspberry Pi
Welcome to the Raspberry Pi Desktop!
Before you start using it, there are a few things to set up.
Press 'Next' to get started.
Cancel
Next
EIZO

VNC Connection

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

Install VNC viewer (Free)



Nagasaki University

Wifi : Select SSID

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 ←remove # and save & reboot
hdmi_force_hotplug=1

sudo shutdown –r now

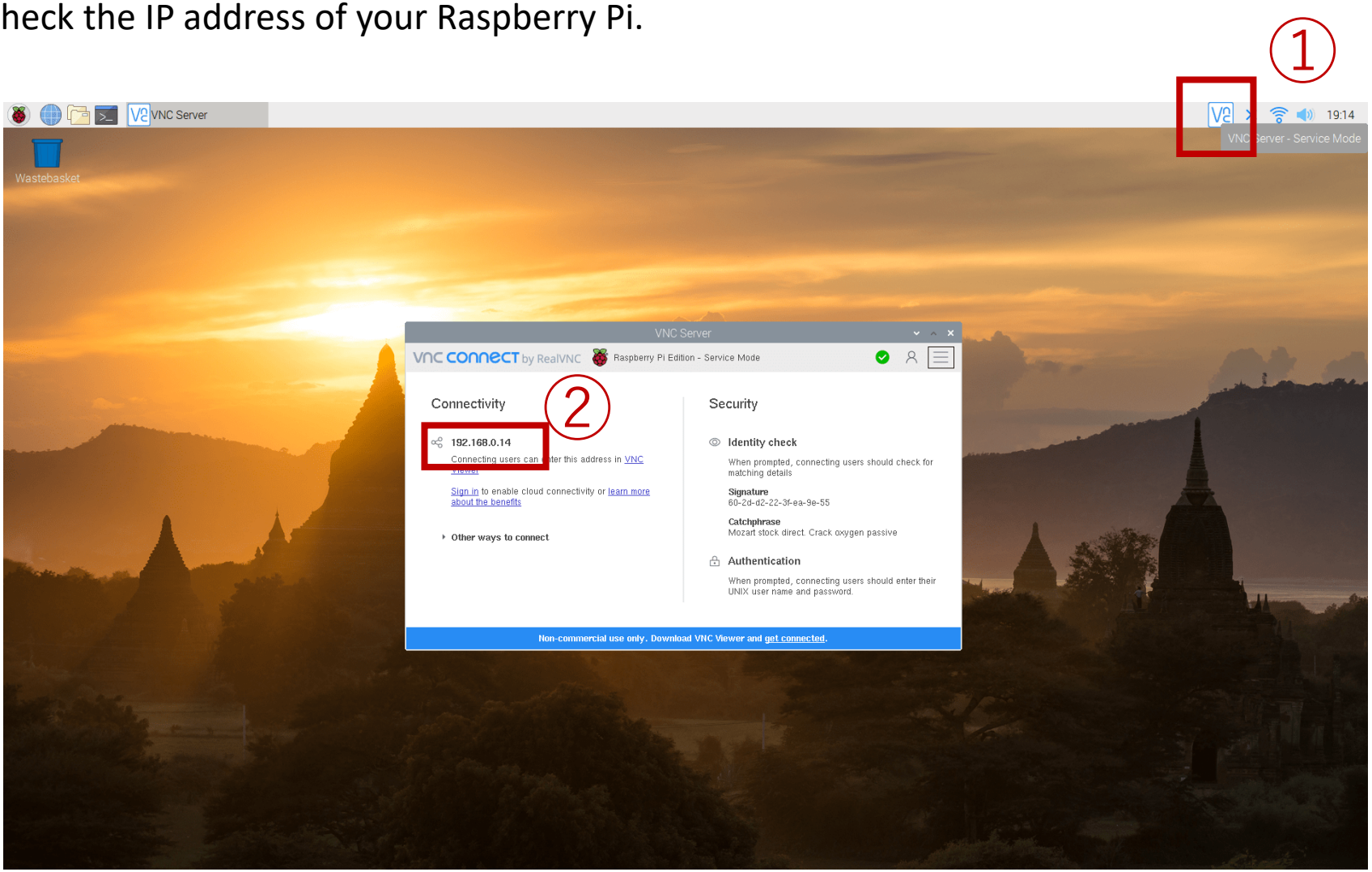
Machine Name

User: pi
Pw: raspbery

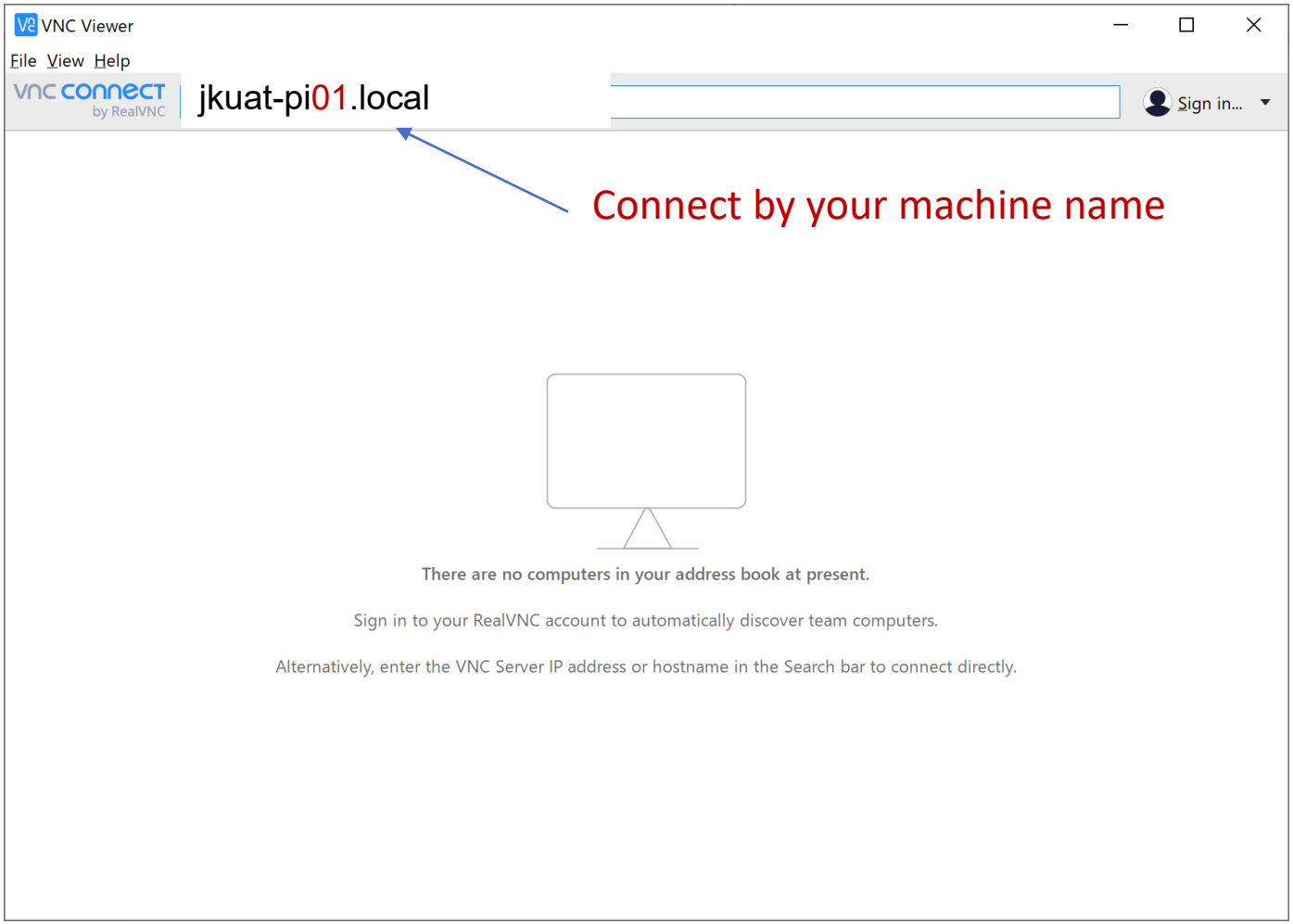
Machine Name:
jkuat-pi01.local ~ jkuat-pi08.local

VNC Connection

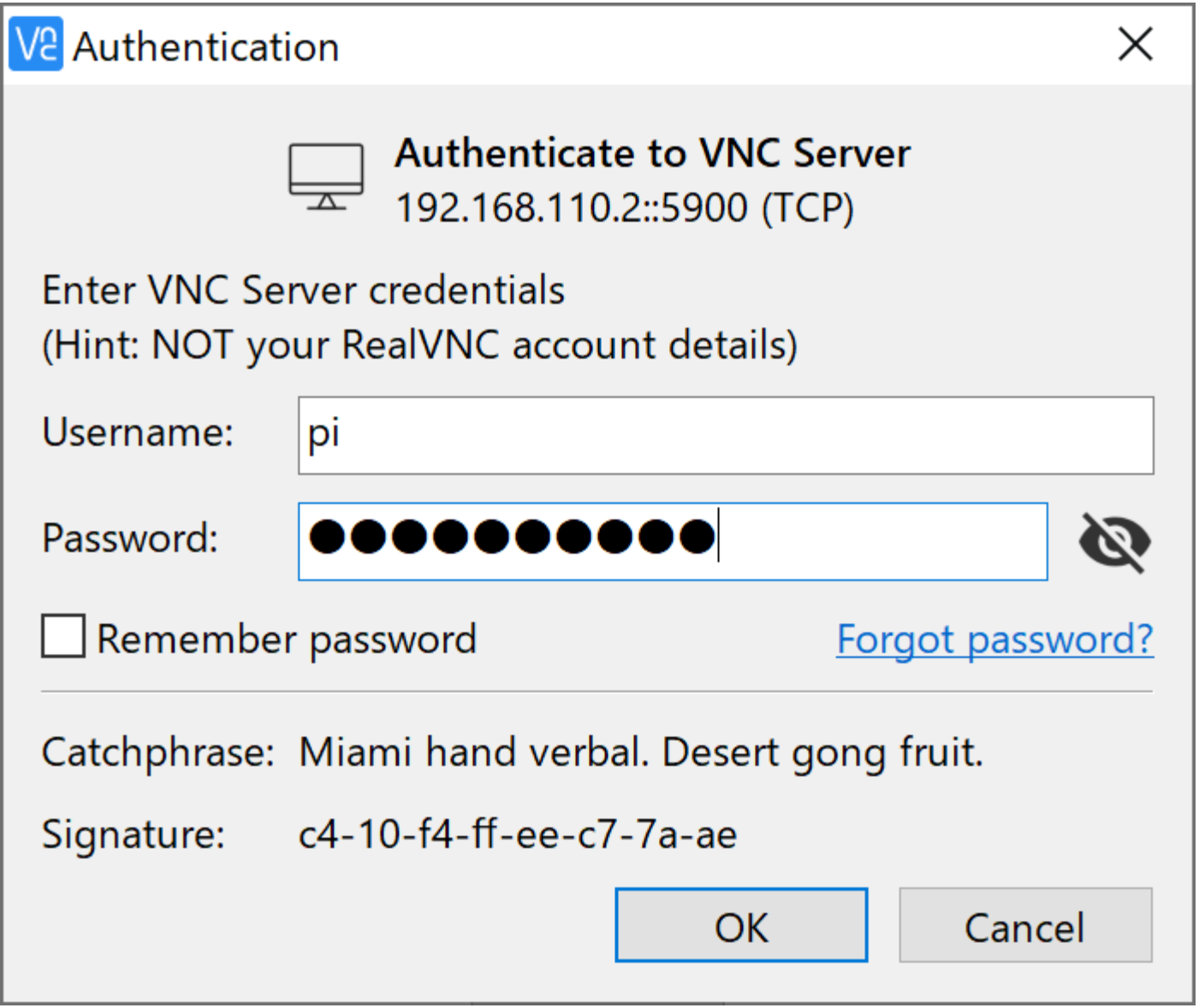
Check the IP address of your Raspberry Pi.



VNC connect

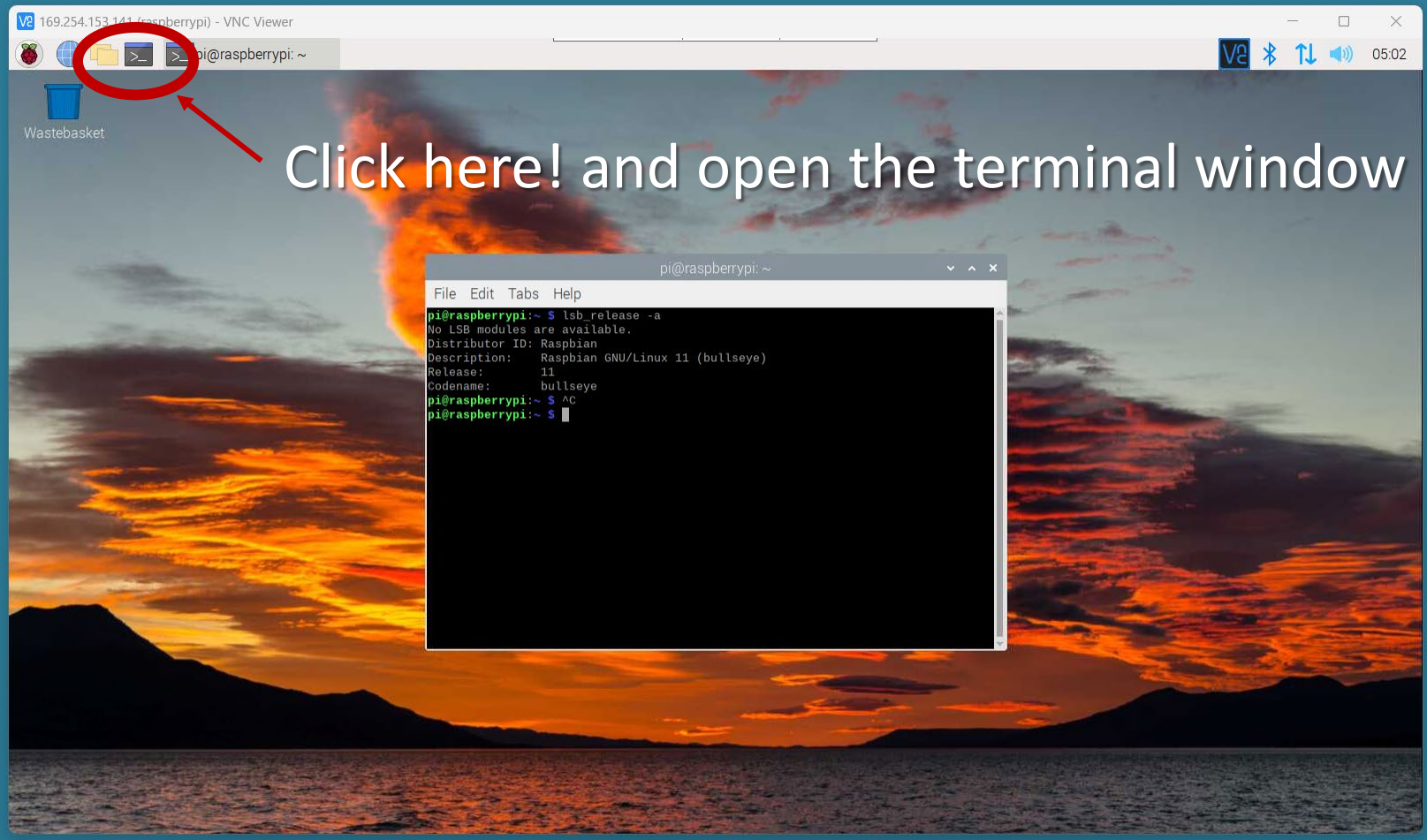

VNC Viewer
File View Help
vnc connect by RealVNC
jkuat-pi01.local
Sign in...
Connect by your machine name
There are no computers in your address book at present.
Sign in to your RealVNC account to automatically discover team computers.
Alternatively, enter the VNC Server IP address or hostname in the Search bar to connect directly.

Username and Password



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

Connected via VNC viewer



Nagasaki University

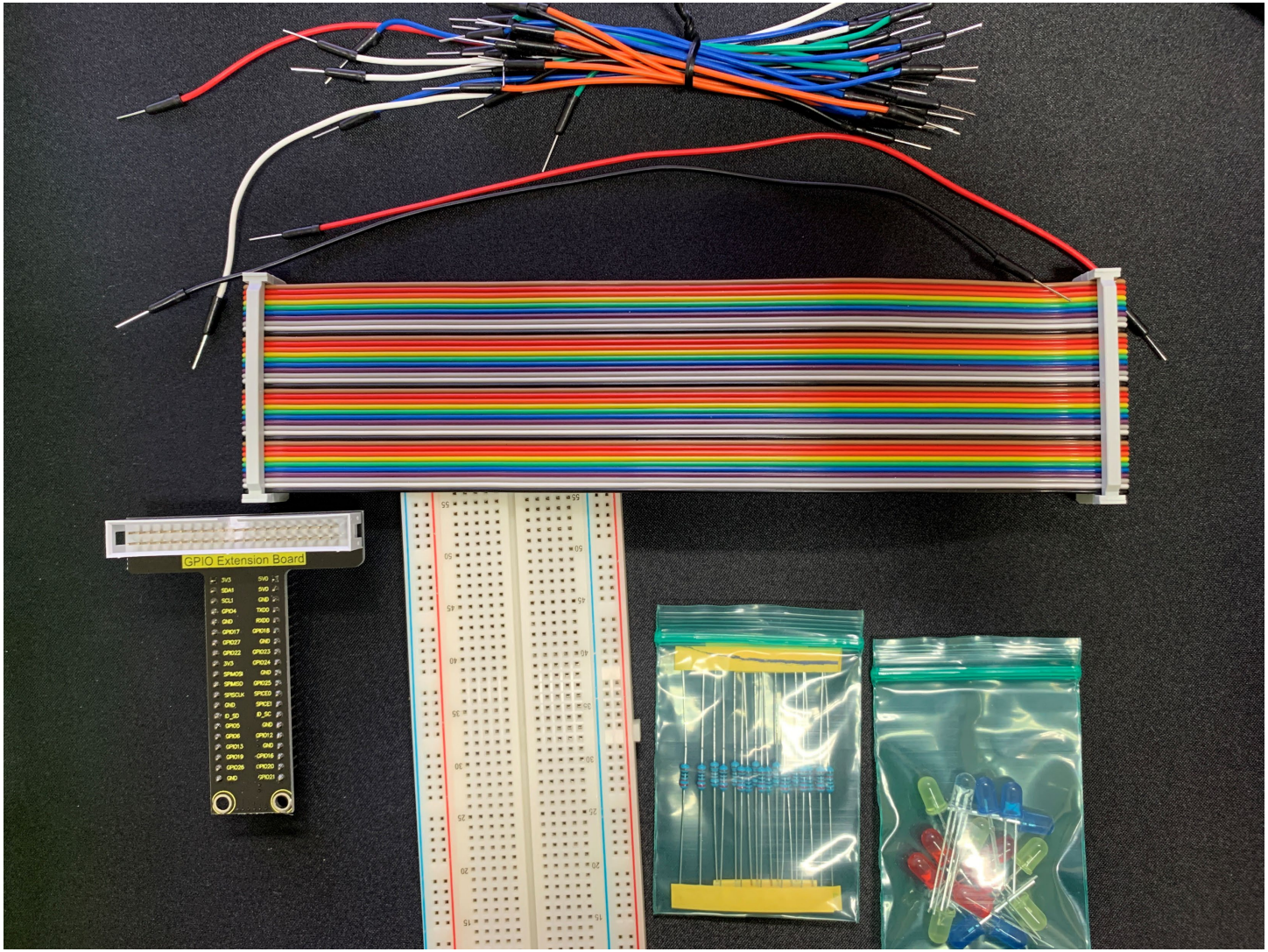
Raspbian OS version : lsb_release –a

```
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
```

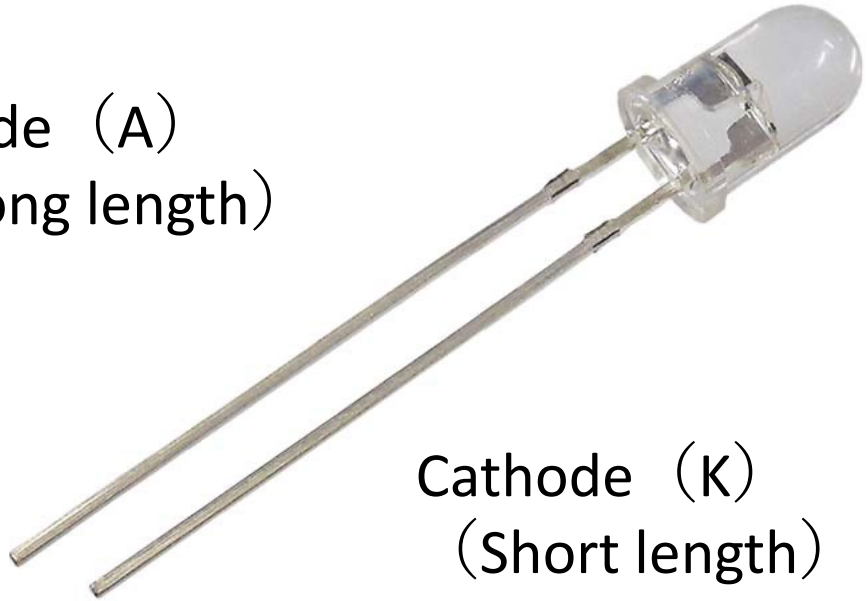
LED flashing

Confirm your handout

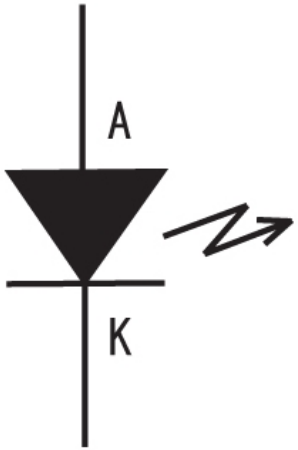


LED Electrode

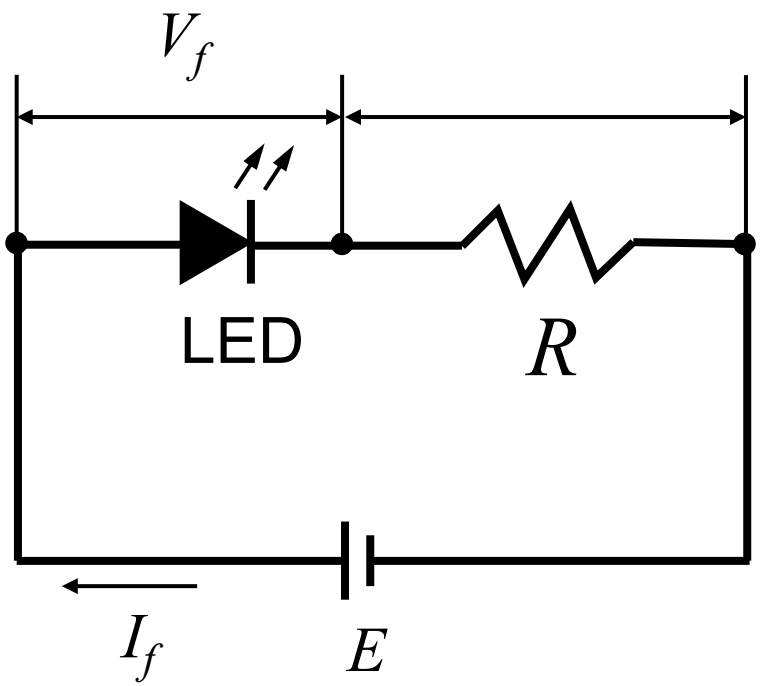
Anode（A）
（Long length）



Cathode（K）
（Short length）



Electric Circuit

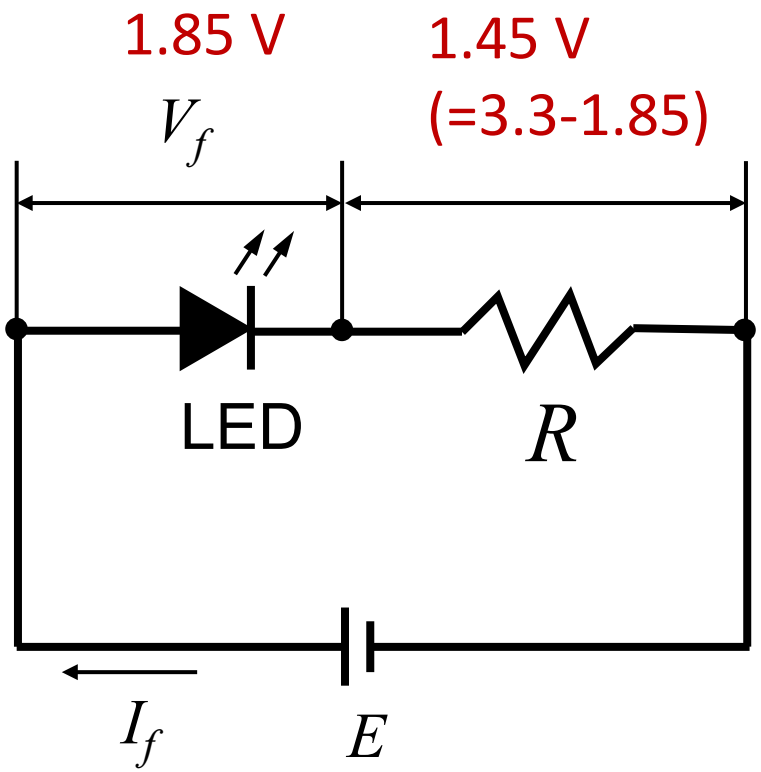


＜Estimation of Resistance＞

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

R：Resistance
E：Power Supply Voltage
V_f：Forward Voltage
I_f：Forward Current

Electric Circuit



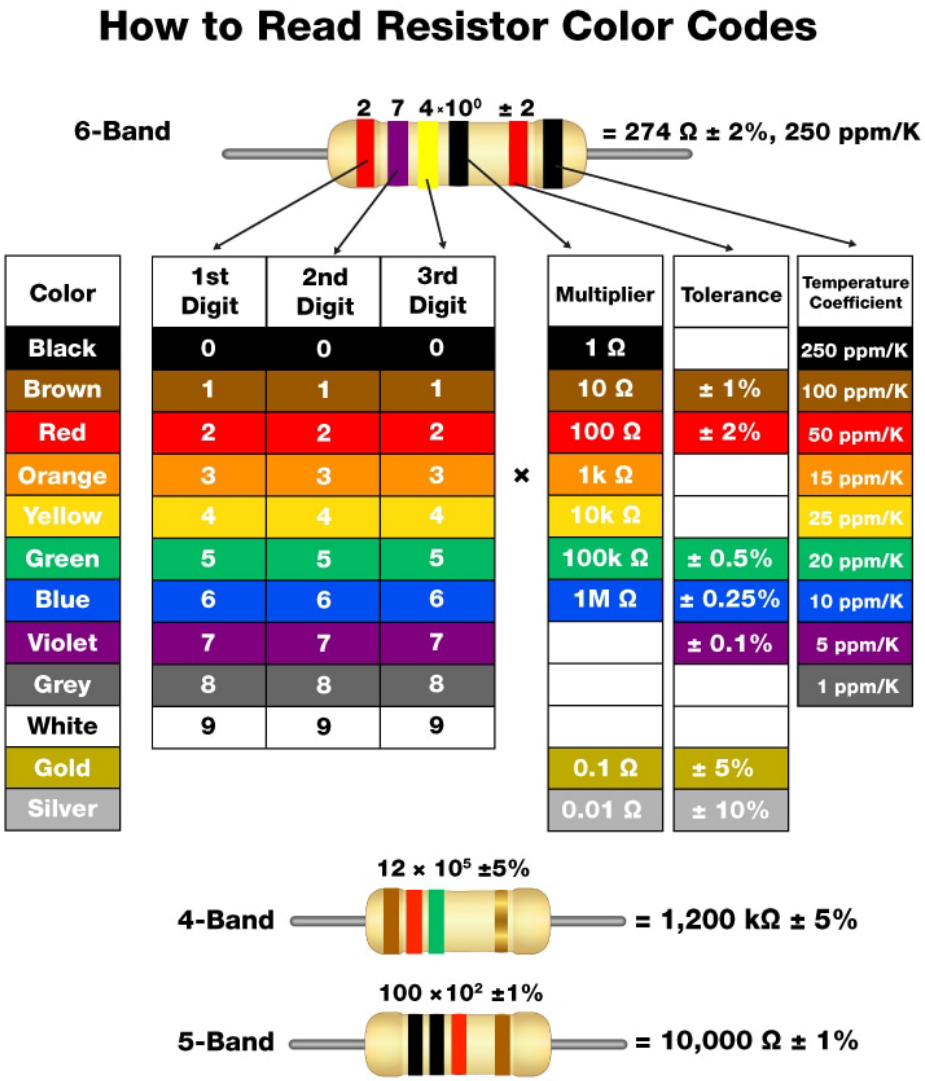
＜Estimation of Resistance＞

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

R：Resistance
E：Power Supply Voltage
V_f：Forward Voltage
I_f：Forward Current

If R = 100 Ω then I_f = 14.5 mA …O.K. (l.t.16mA)
If R = 220 Ω then I_f = 6.6 mA …O.K. (l.t.16mA)
If R = 1K Ω then I_f = 1.45 mA …O.K. (l.t.16mA)

Resistor Color Codes



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	± 1%	100 ppm/K
Red	2	2	2		100 Ω	± 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	± 0.5%	20 ppm/K
Blue	6	6	6		1M Ω	± 0.25%	10 ppm/K
Violet	7	7	7			± 0.1%	5 ppm/K
Grey	8	8	8				1 ppm/K
White	9	9	9				
Gold					0.1 Ω	± 5%	
Silver					0.01 Ω	± 10%	

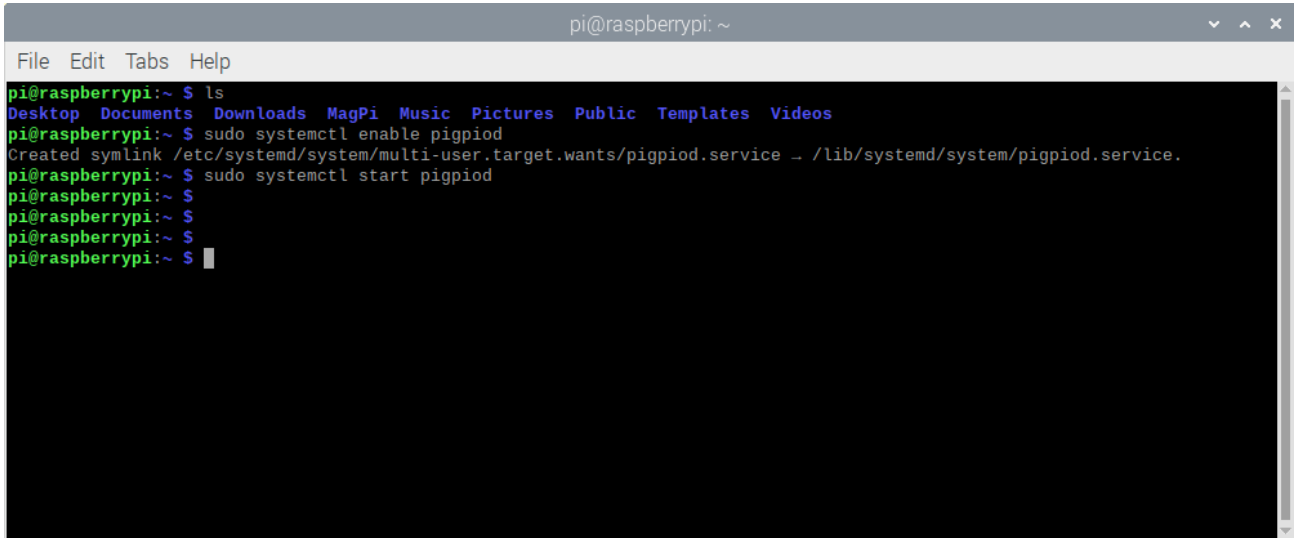
Python Library

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

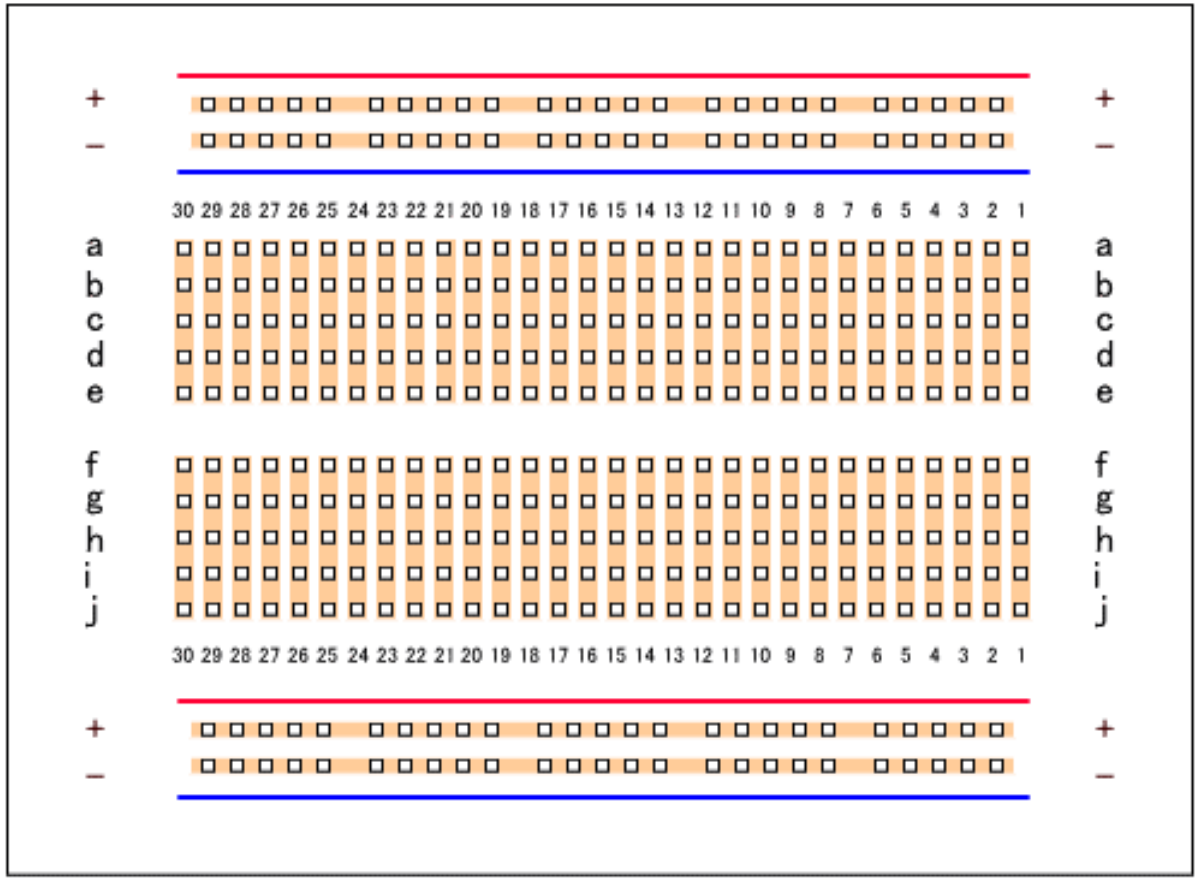
Install pigpio library

```
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.

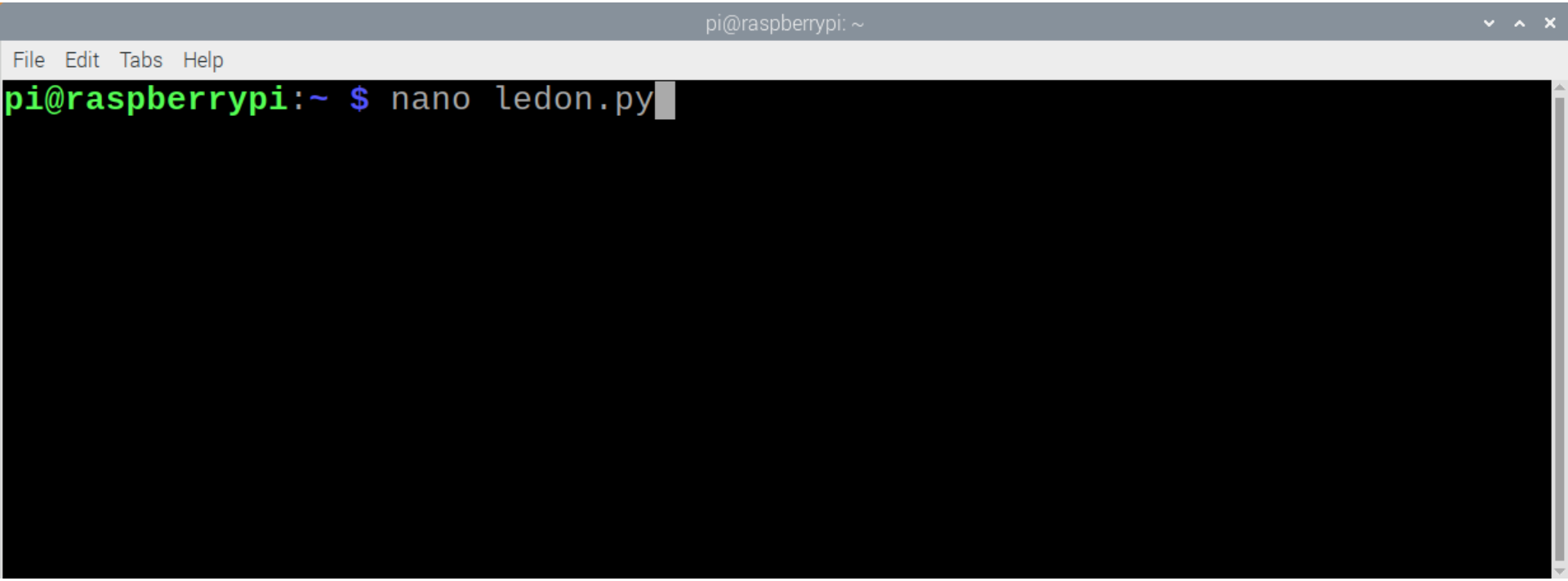


Breadboard


+
–
30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
a
b
c
d
e
f
g
h
i
j

Edit python program

“nano” is simple editor for raspberry pi.



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

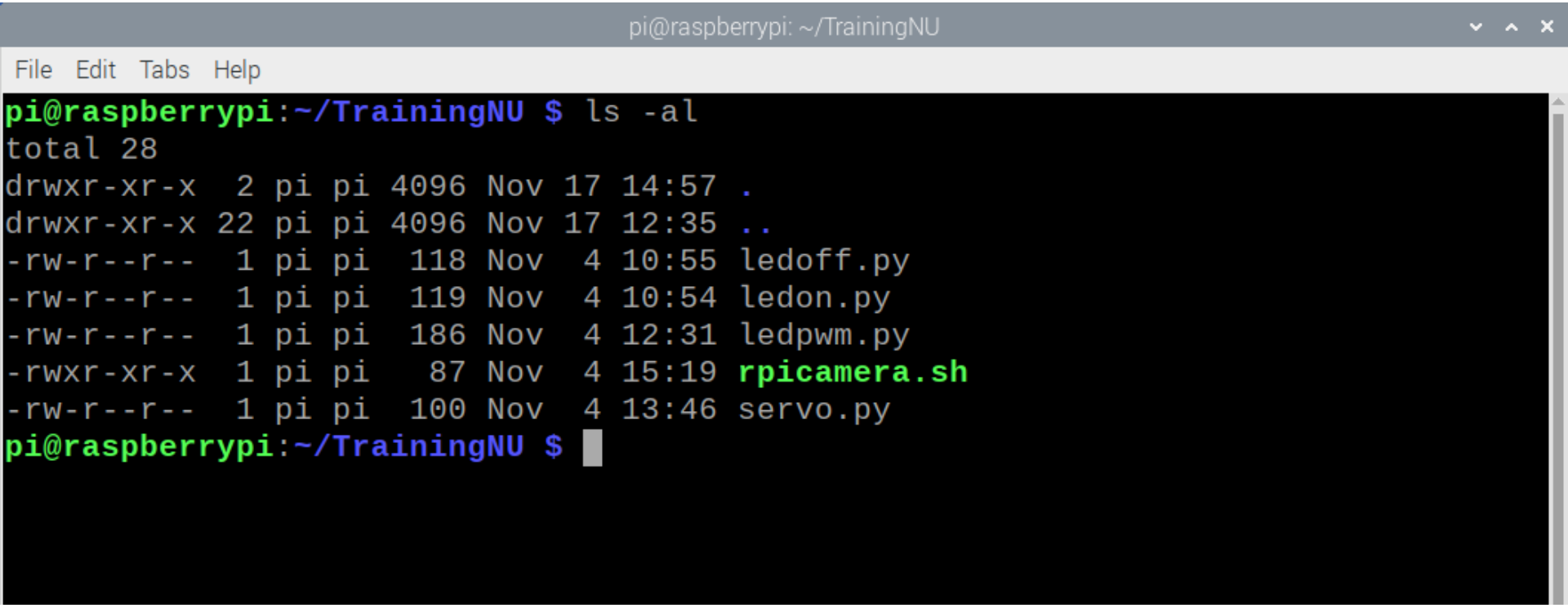
"more" is a linux command to see a text file

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



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                          ledoff.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                          ledoff.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: ledoff.py
^G Get Help       M-D DOS Format      M-A Append        M-B Backup File
^C Cancel         M-M Mac Format      M-P Prepend       ^T To Files
```

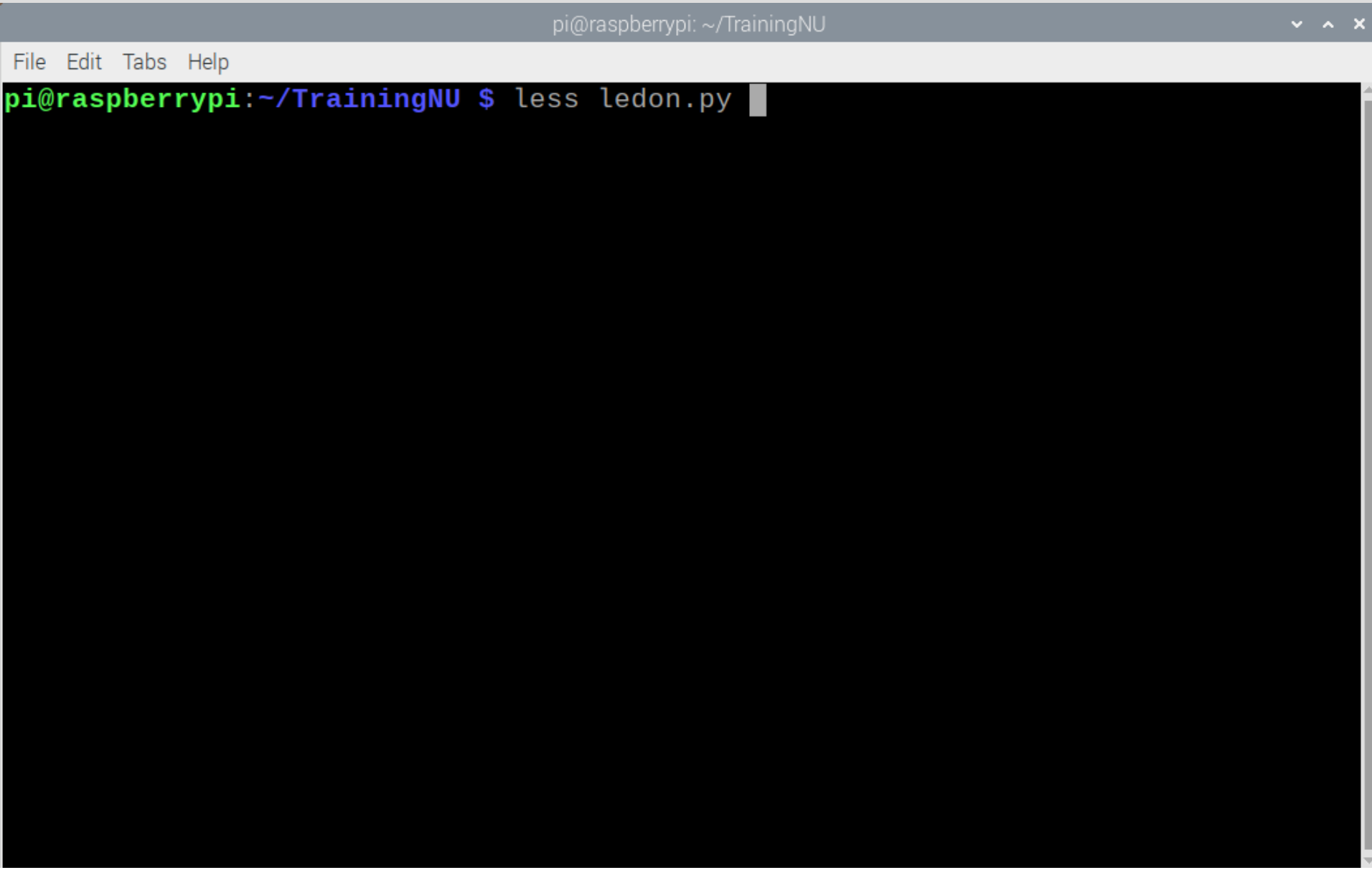
Python program : ledoff.py

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



Linux command : less

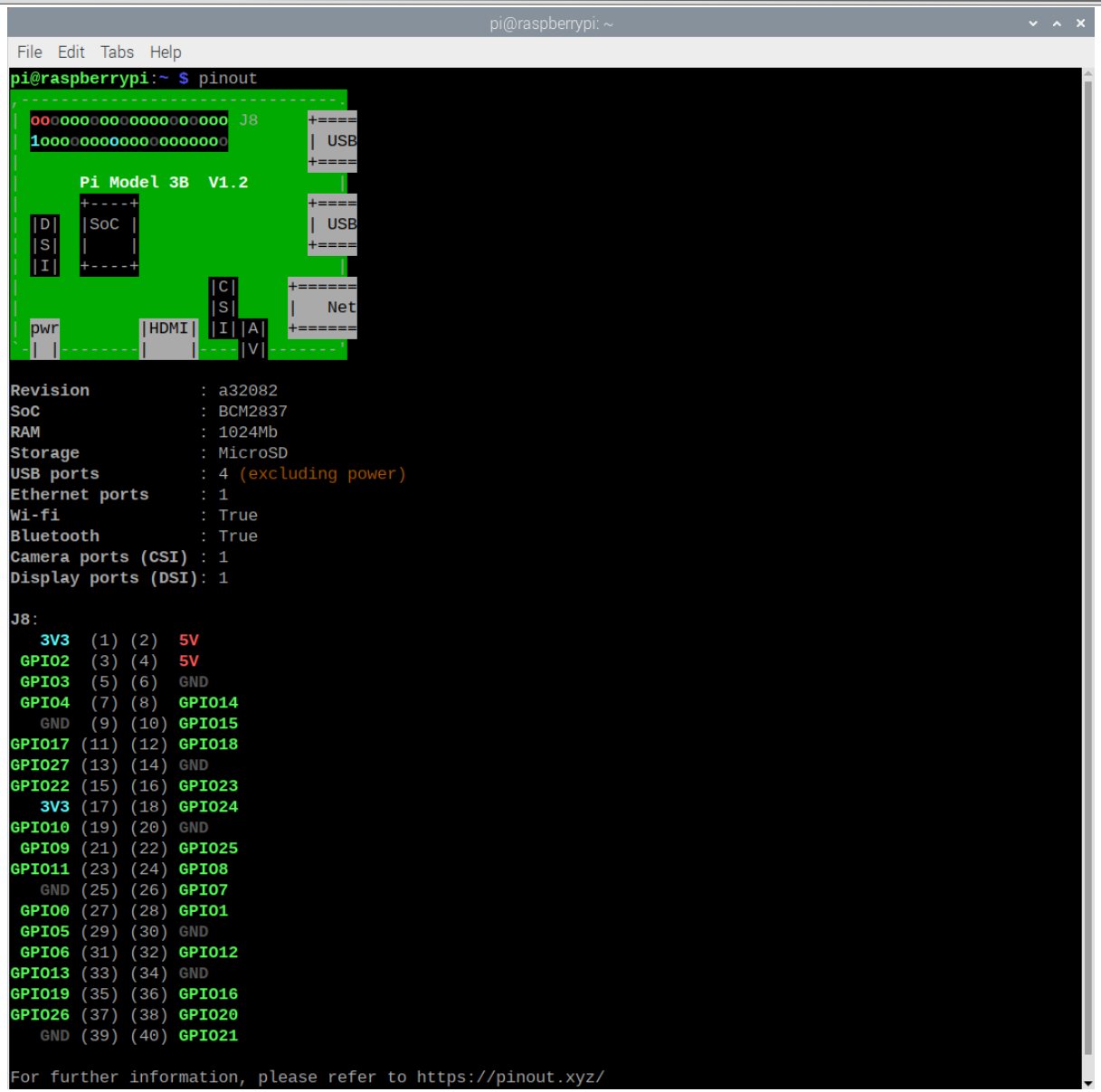
pi@raspberrypi: ~/TrainingNU

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)
```

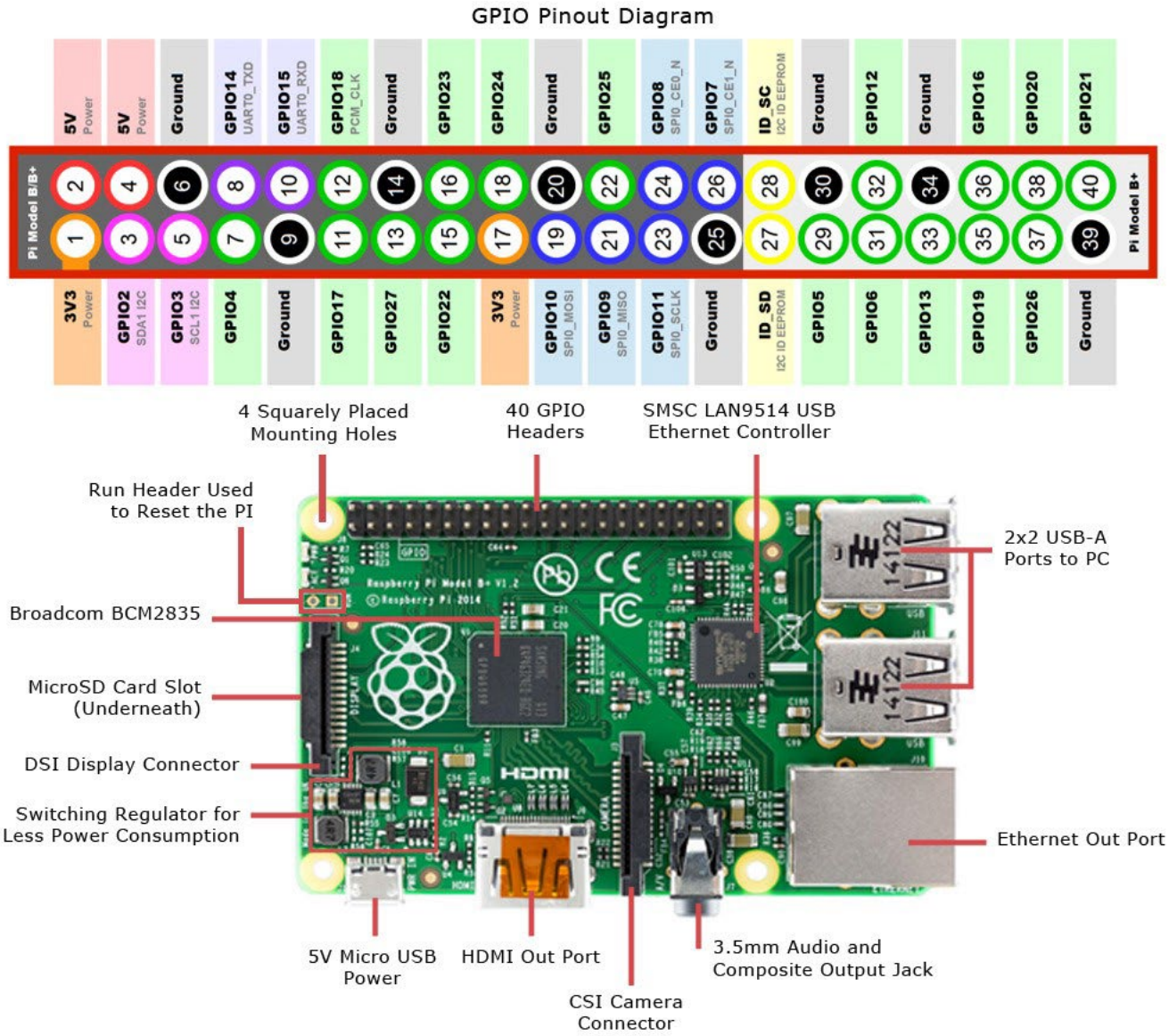
quite : ctrl + c

Linux command : pinout

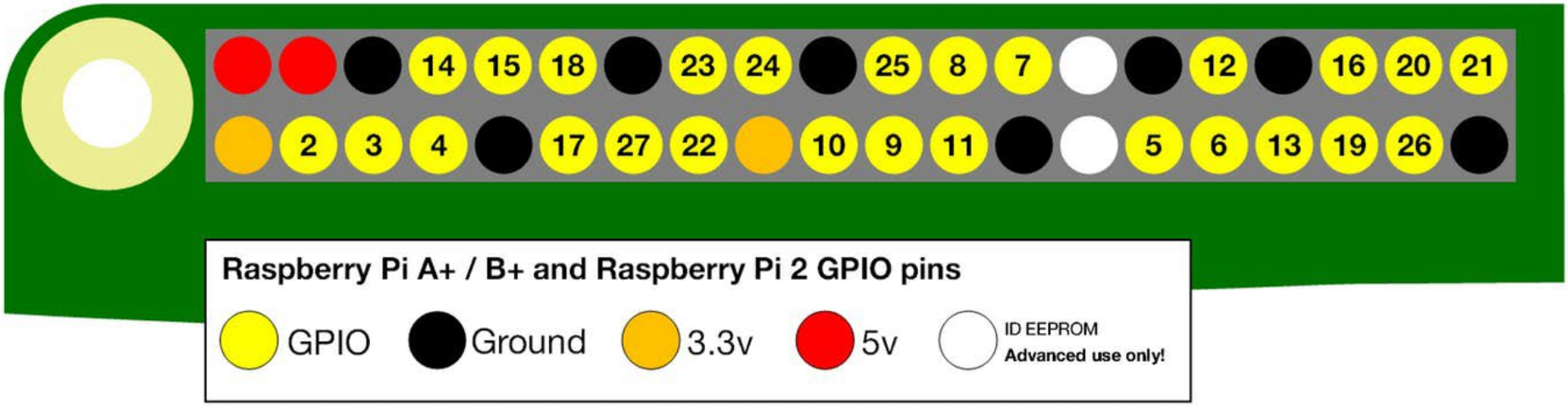


4

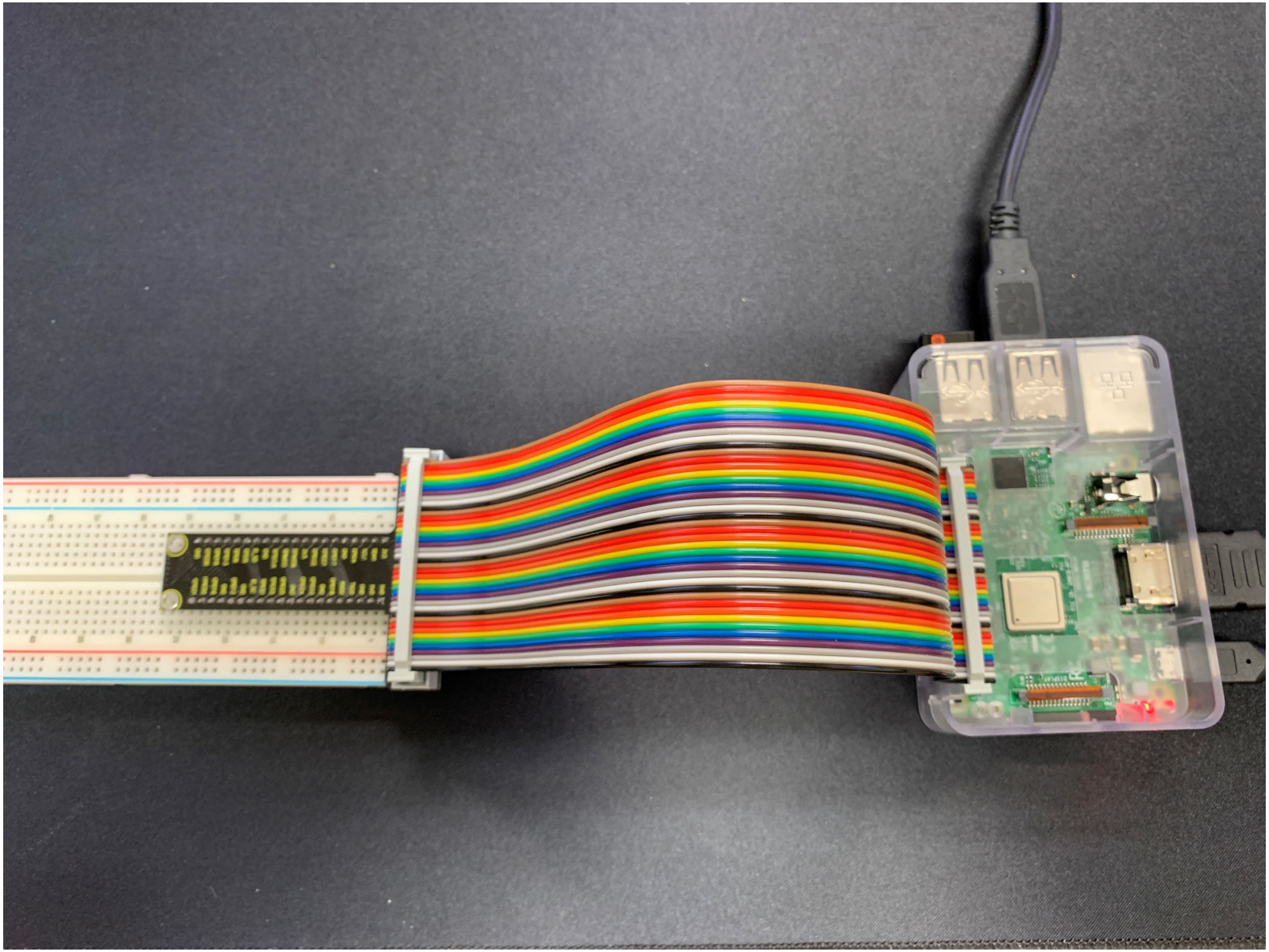
Raspberry pi GPIO



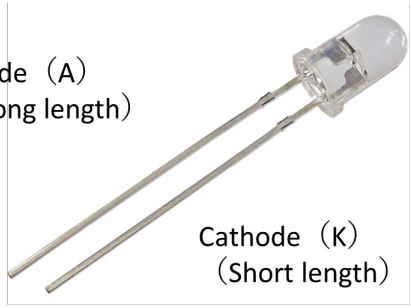
Raspberry pi GPIO

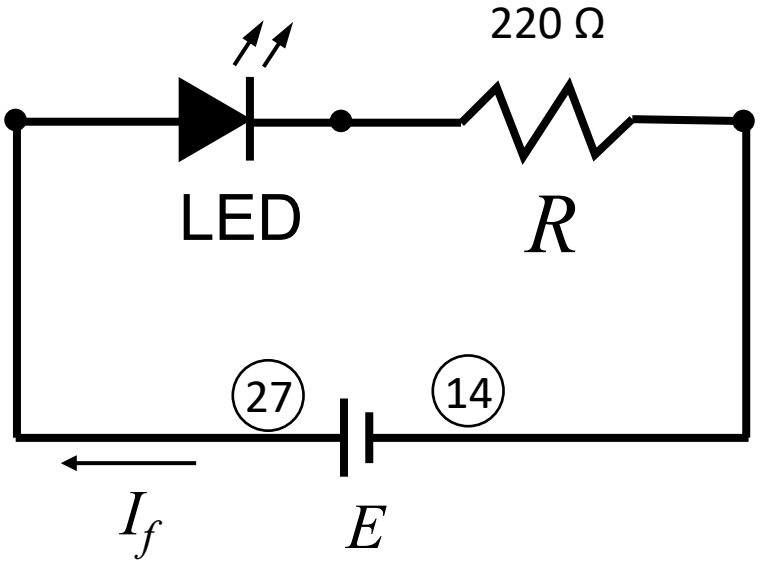


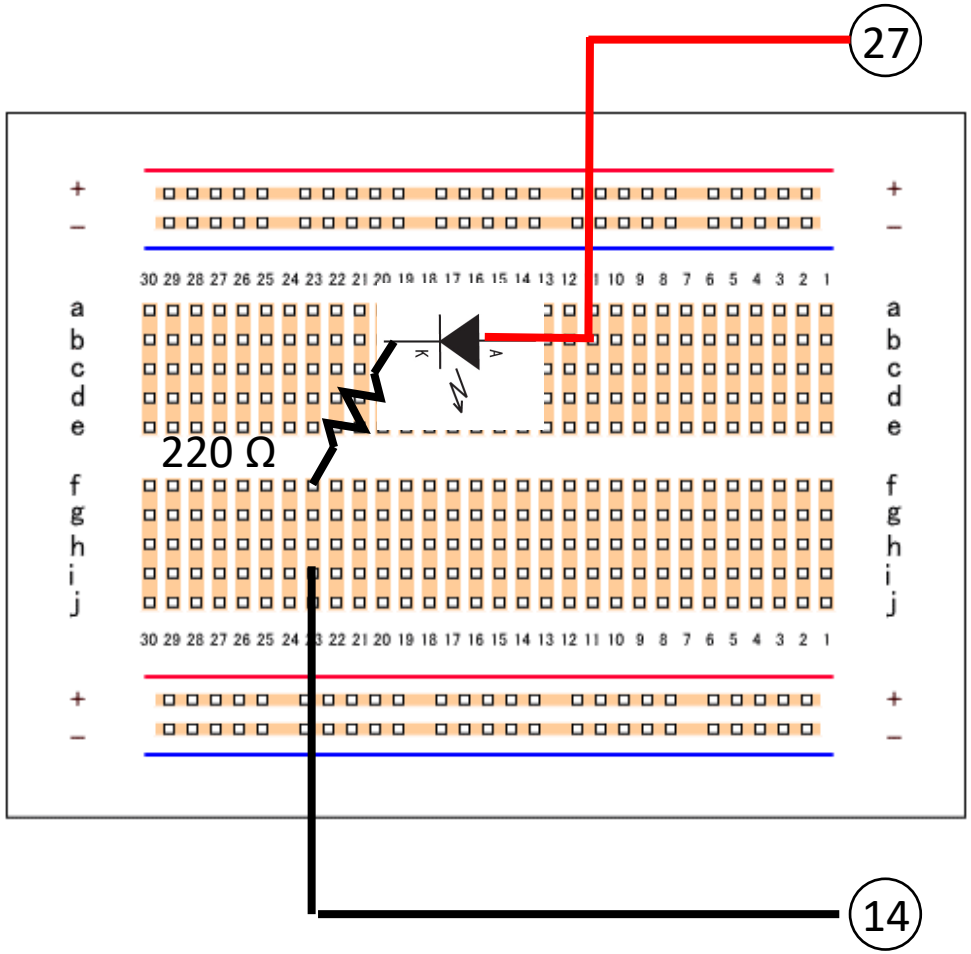
40pin flat cable



Circuit layout for LED

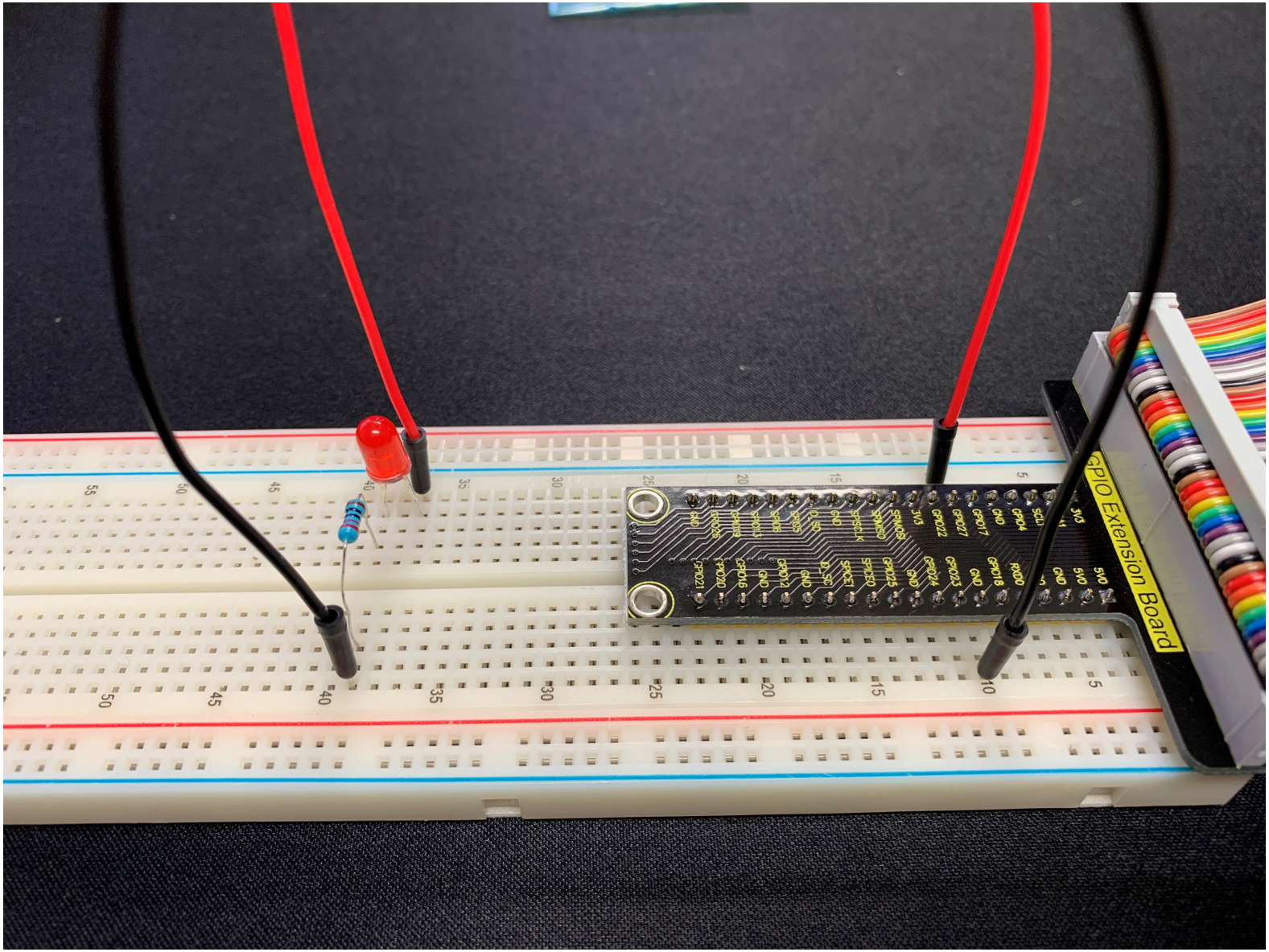




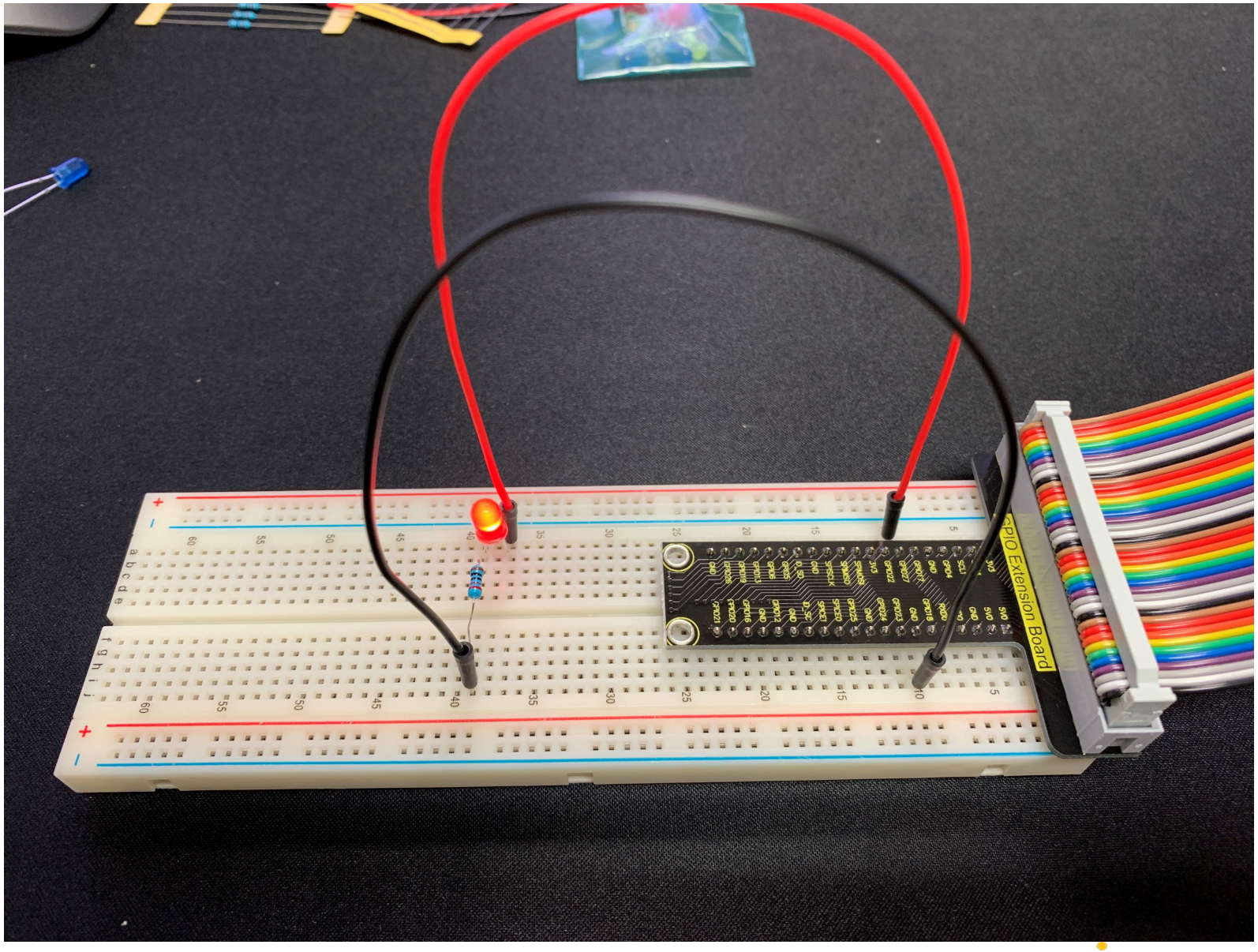


Nagasaki University

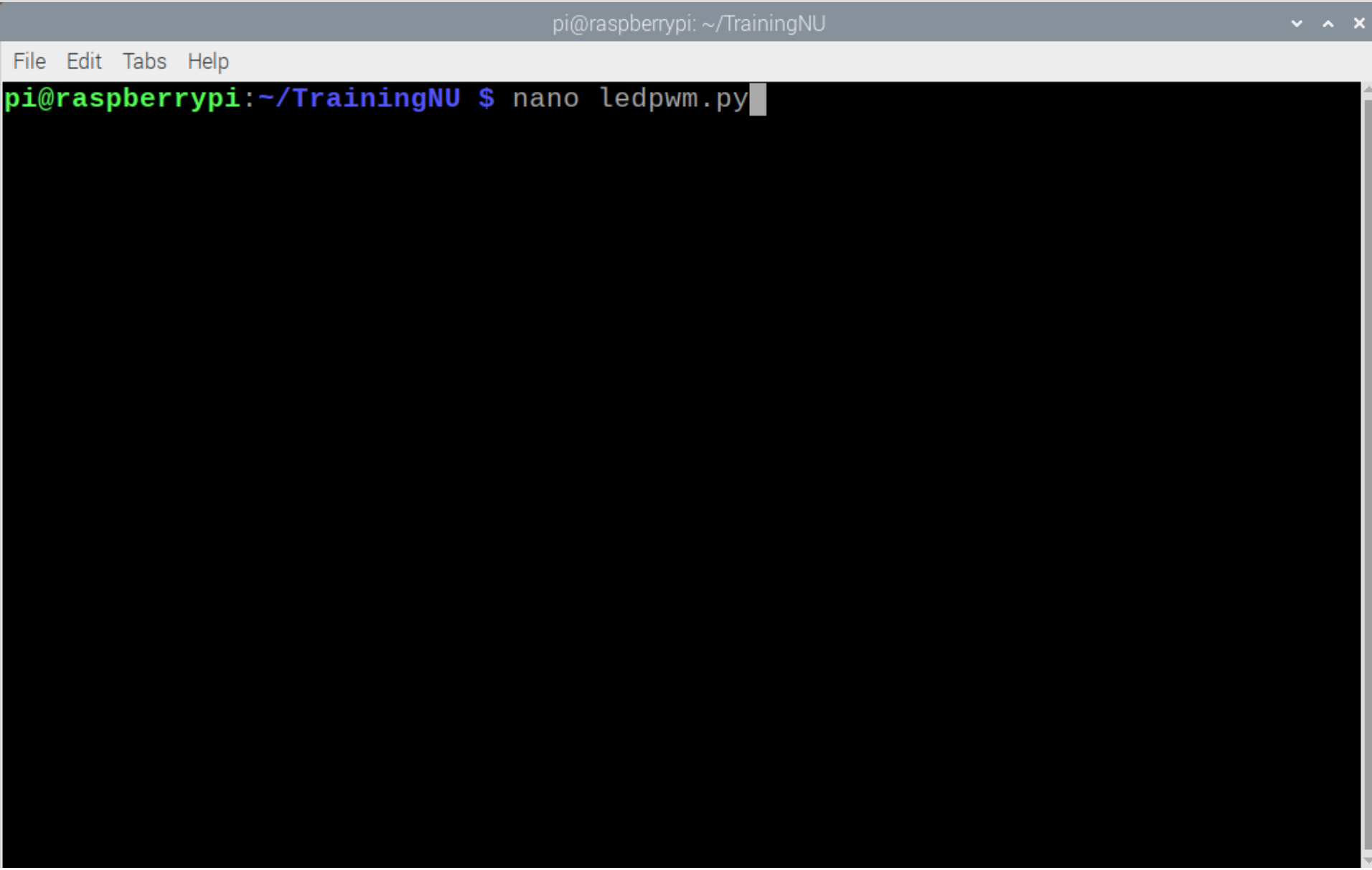
Circuit layout for LED



Turn on your LED



Python program : ledpwm.py



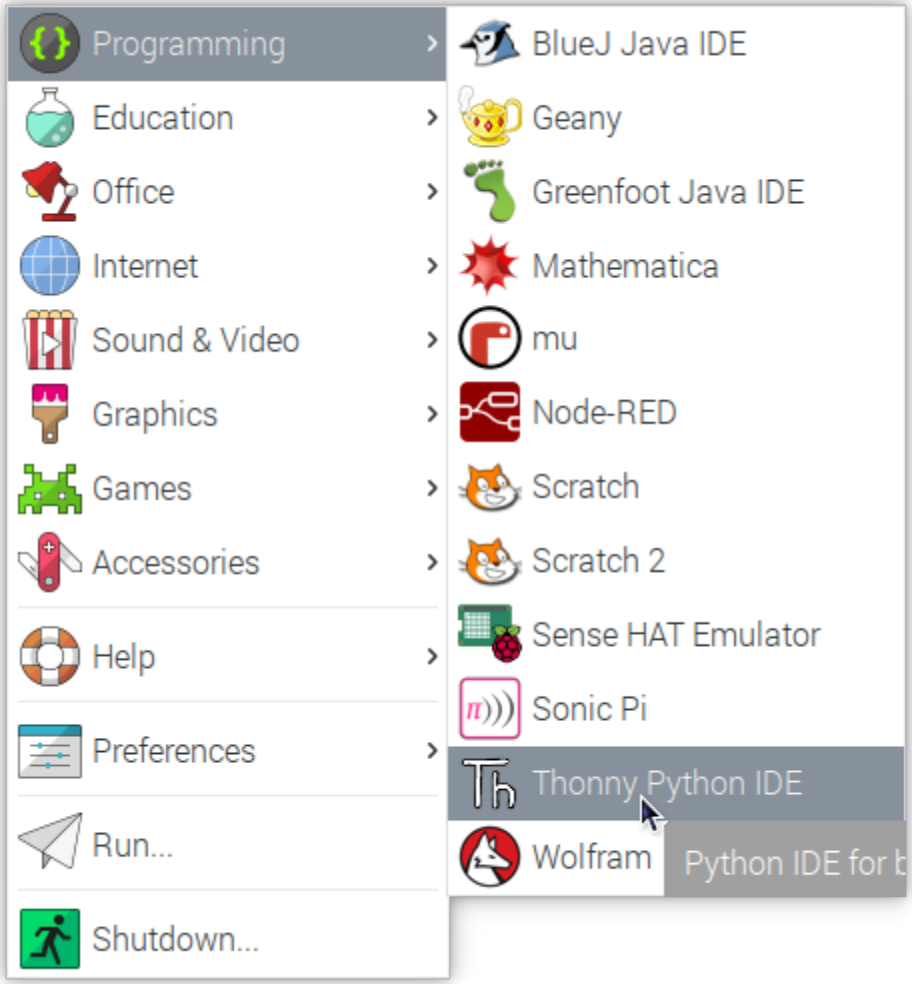
Python program : ledpwm.py

pi@raspberrypi: ~/TrainingNU

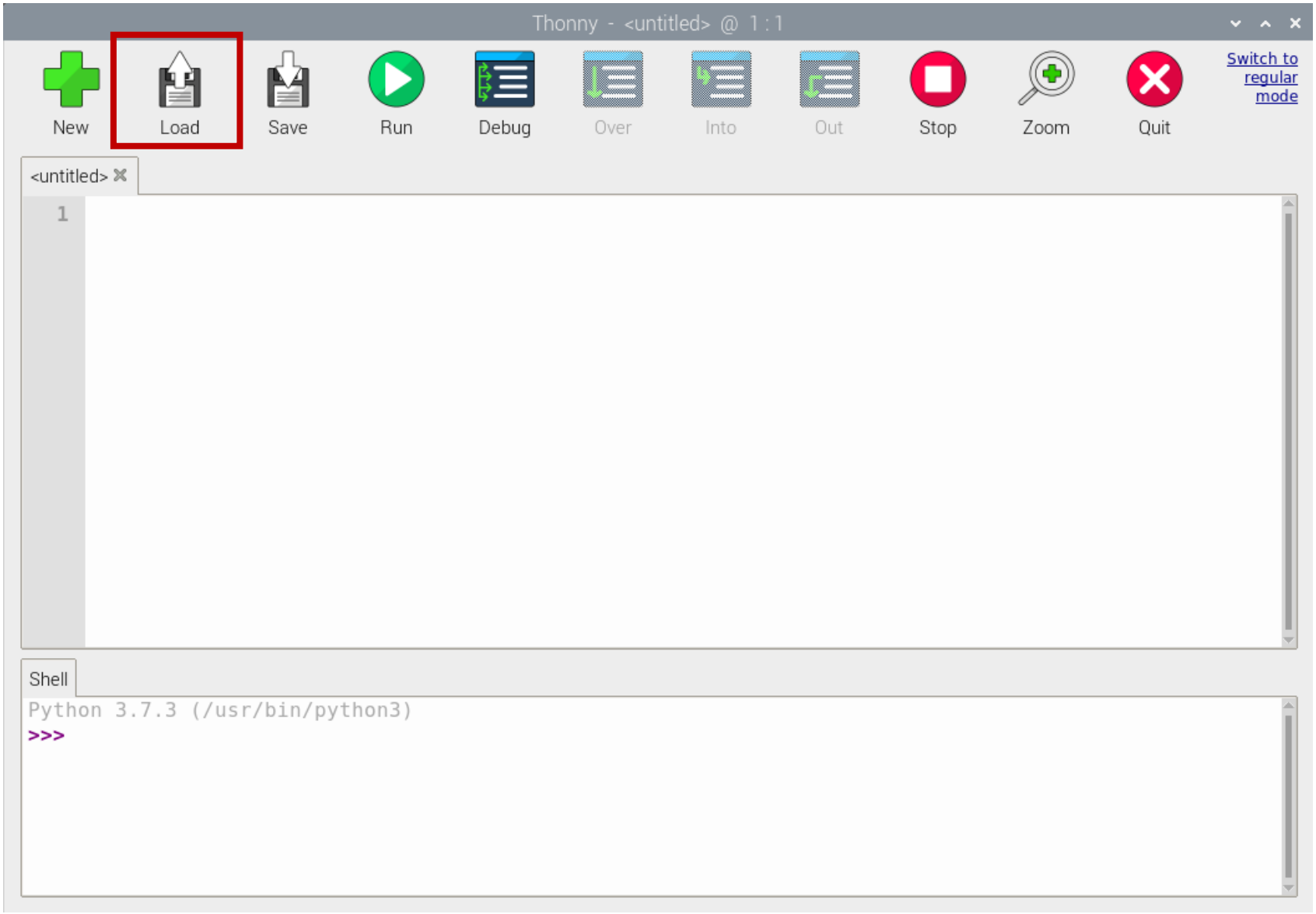
File Edit Tabs Help

```
  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
```

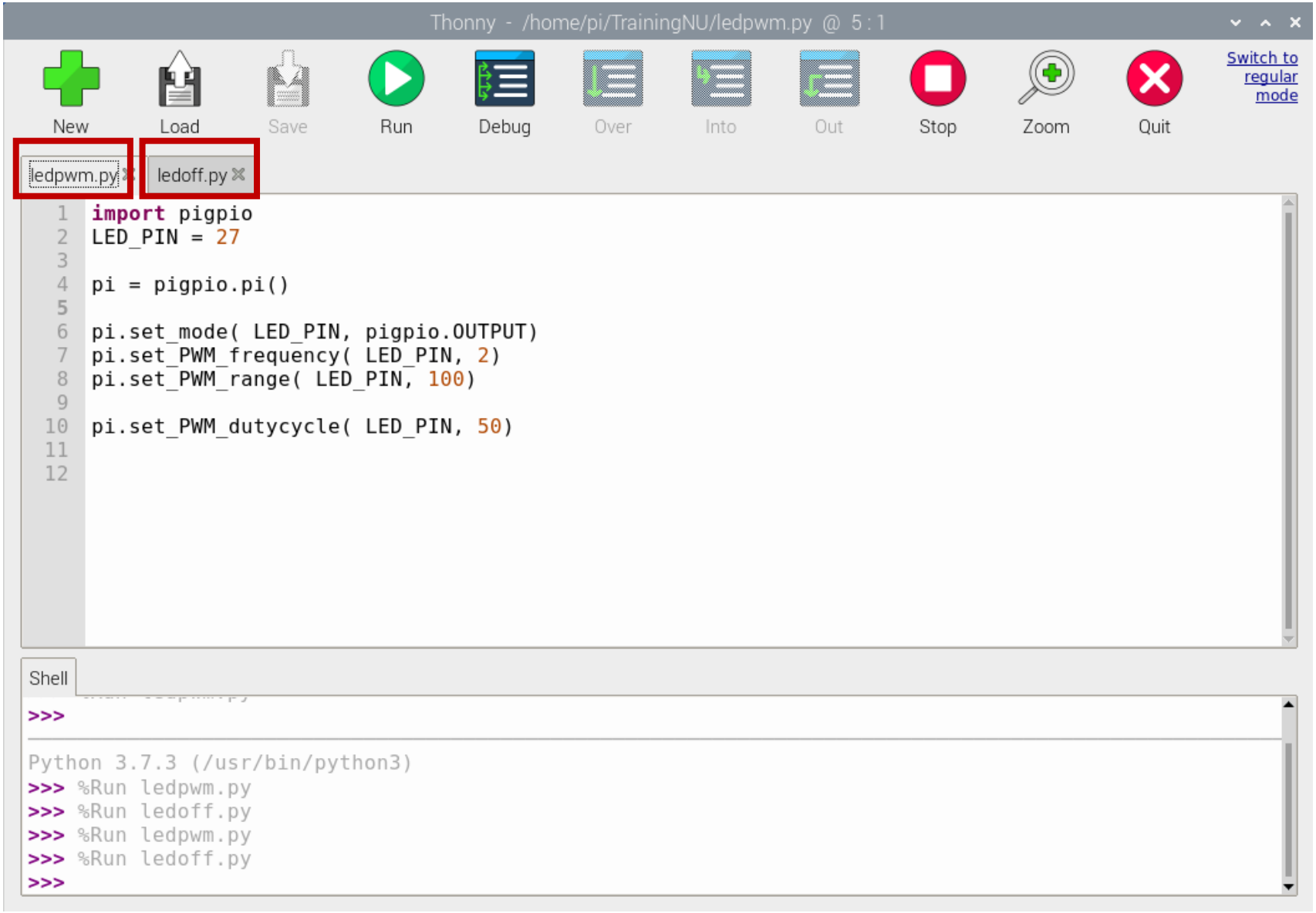
Programming Environment : Thonny Python IDE



Thonny Python IDE

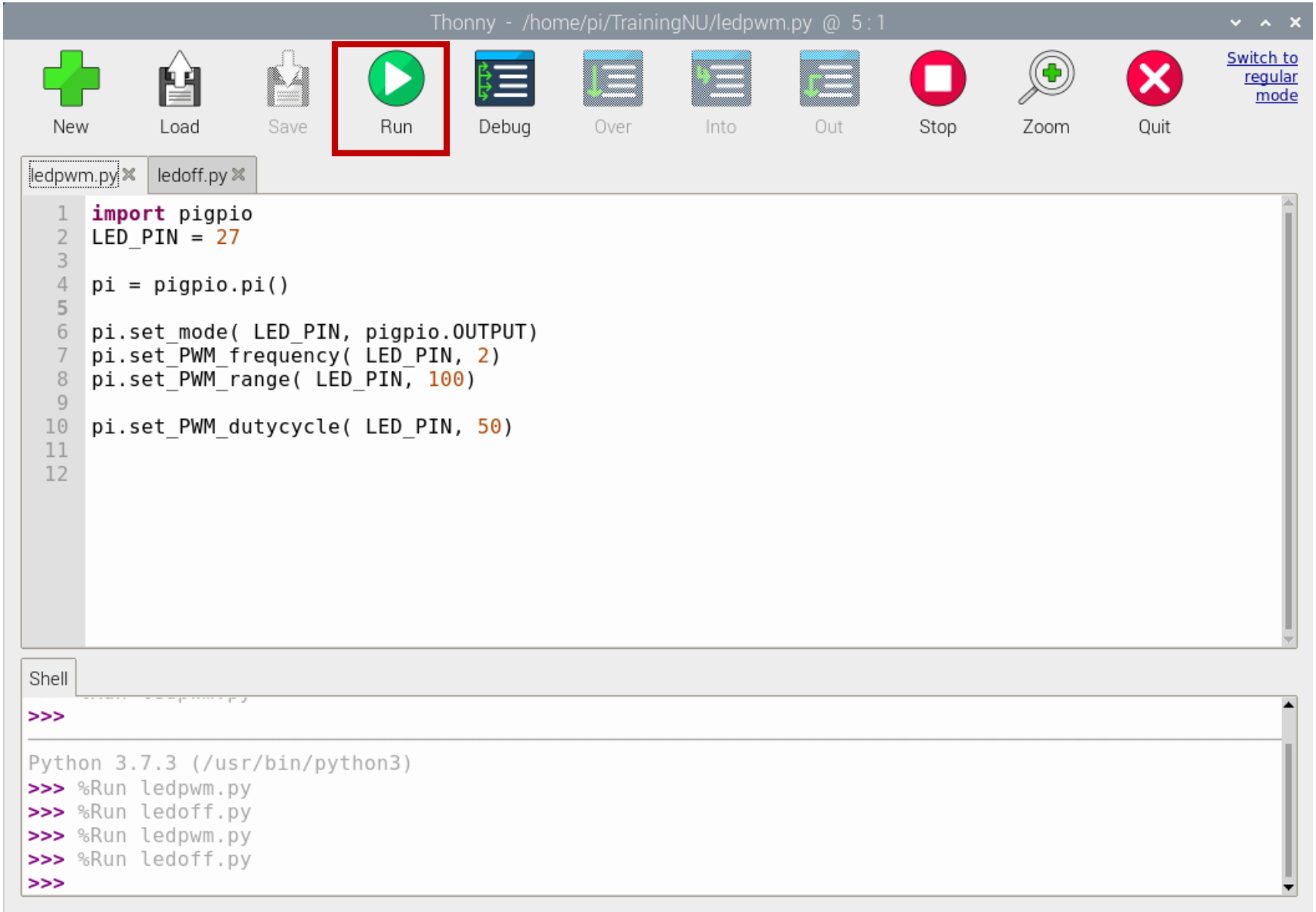


Thonny Python IDE

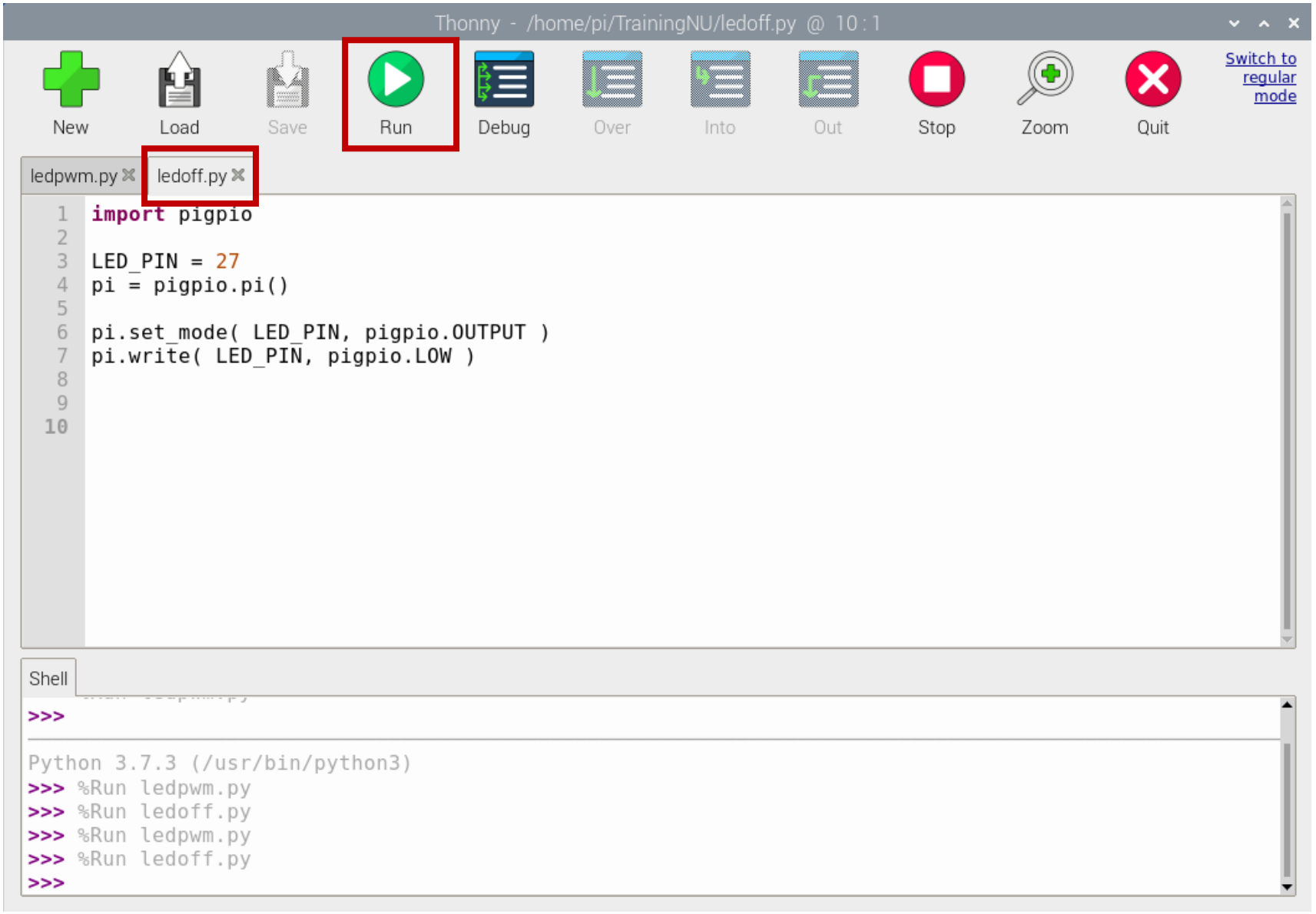


5

Thonny Python IDE



Thonny Python IDE



Thonny Python IDE

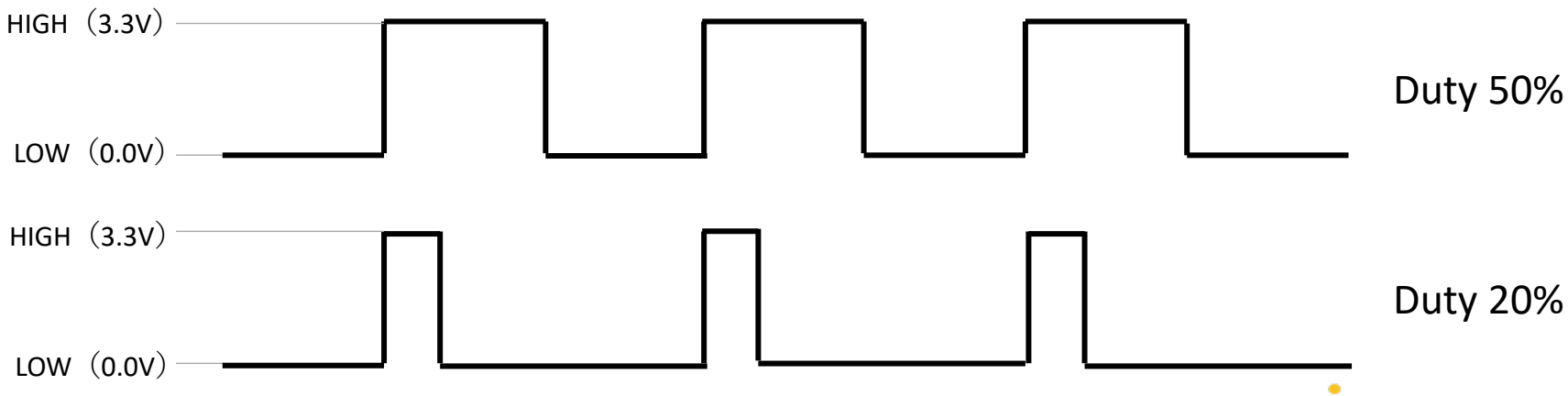
Thonny - /home/pi/TrainingNU/ledpwm.py @ 5 : 1

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

ledpwm.py ledoff.py

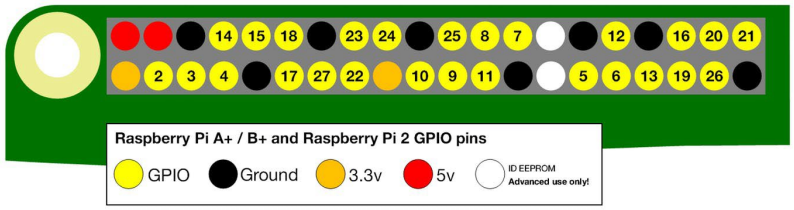
```
1  import pigpio
2  LED_PIN = 27
3
4  pi = pigpio.pi()
5
6  pi.set_mode( LED_PIN, pigpio.OUTPUT)
7  pi.set_PWM_frequency( LED_PIN, 2)
8  pi.set_PWM_range( LED_PIN, 100)
9
10 pi.set_PWM_dutycycle( LED_PIN, 50)
11
12
```

PWM：Pulse Width Modulation



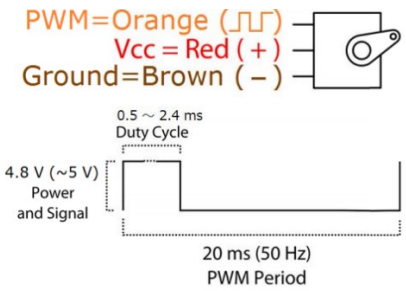
Servo motor

Servo Motor SG90

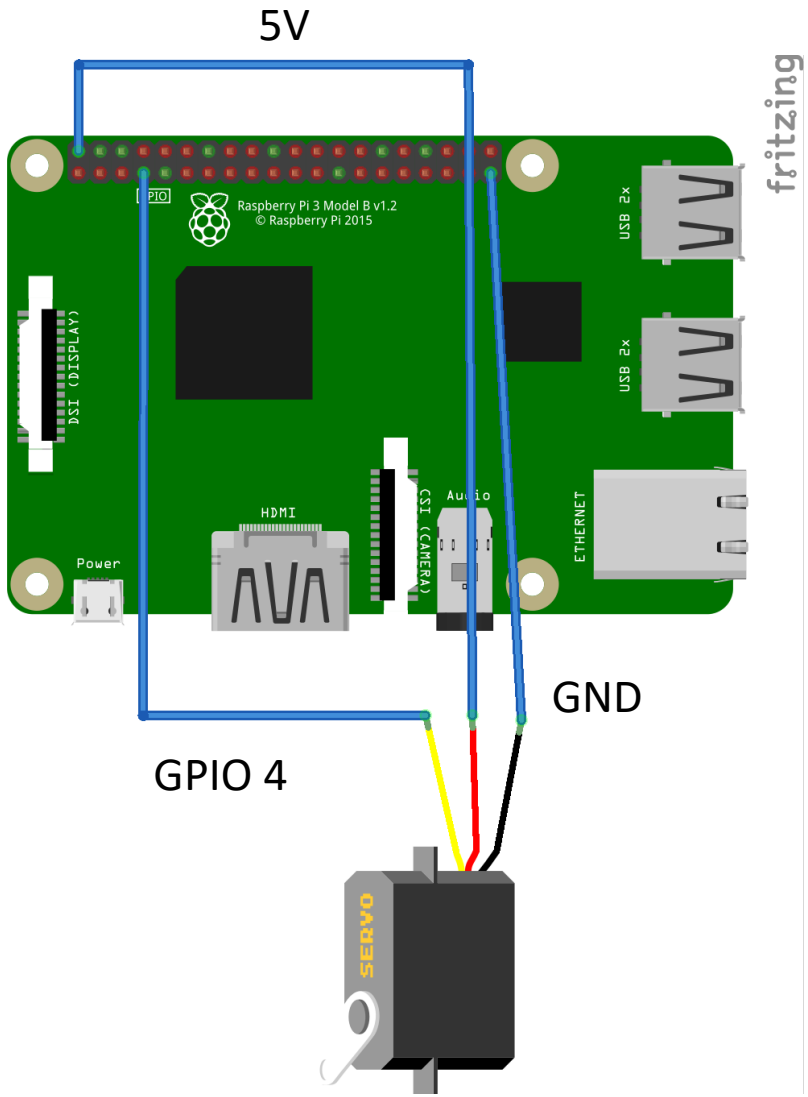


Specifications

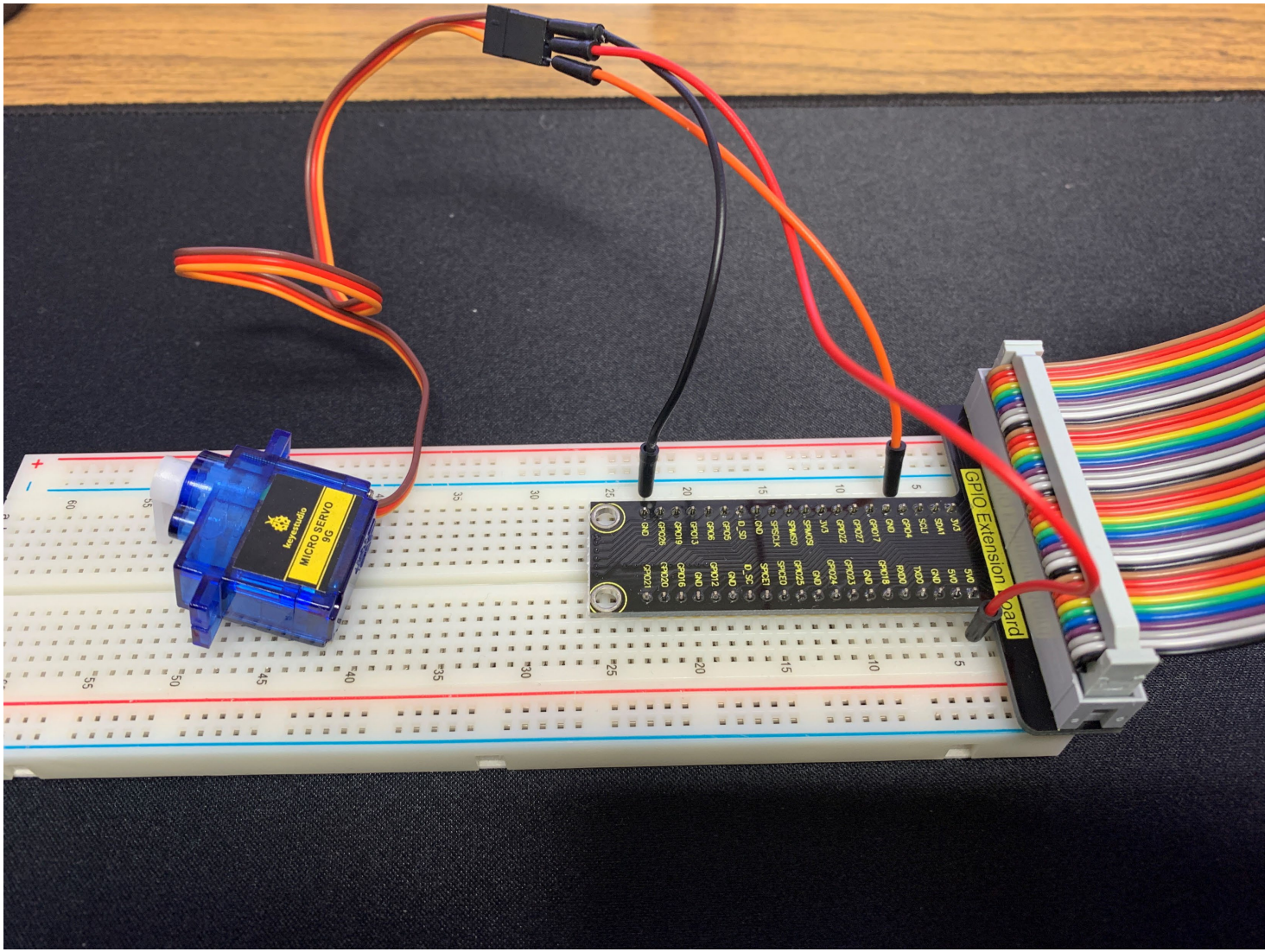
- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 µs
- Temperature range: 0 ℃ – 55 ℃



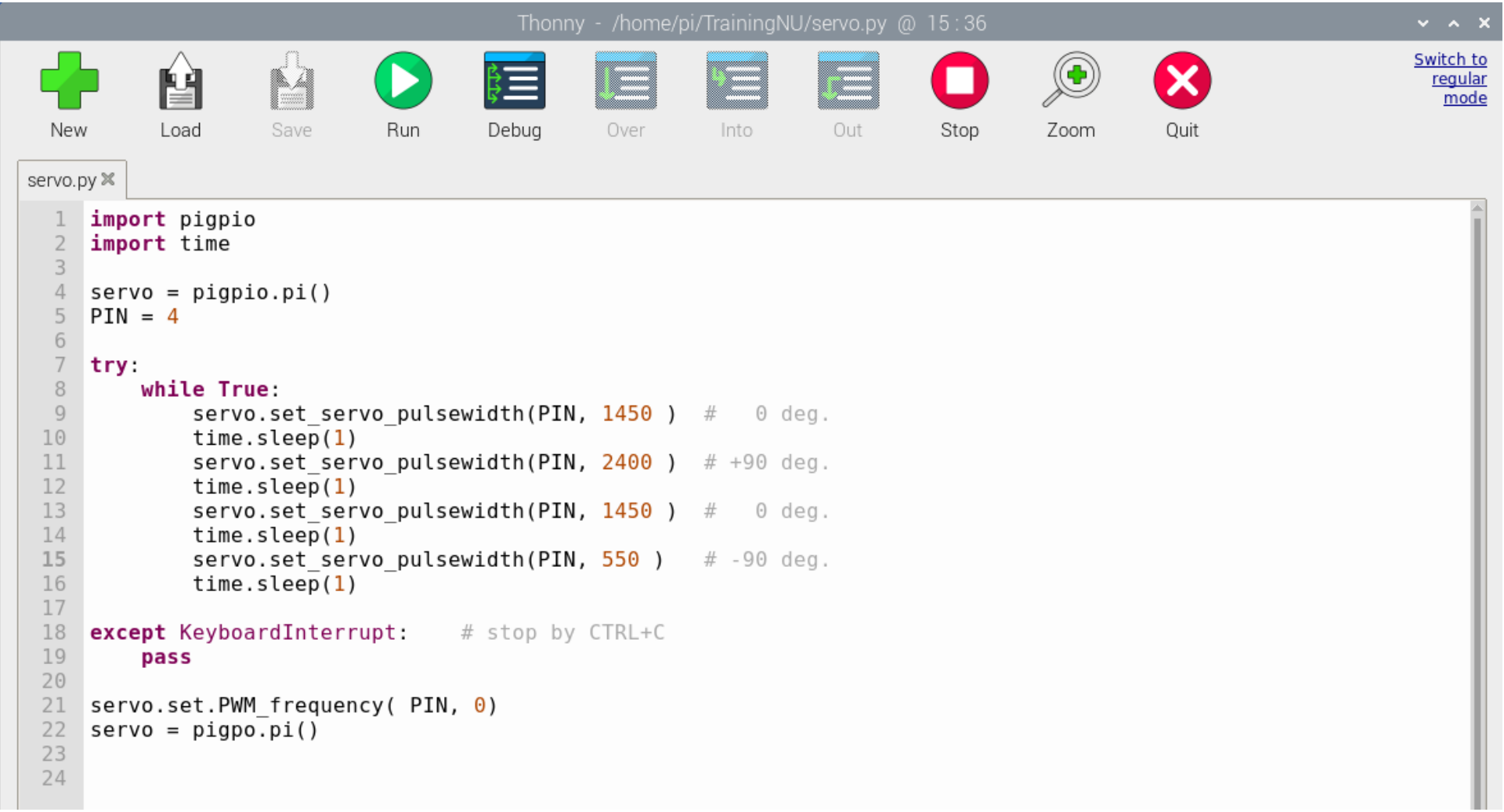
Position "0" (1.45 ms pulse) is middle, "90" (~2.4 ms pulse) is all the way to the right, "-90" (~ 0.5 ms pulse) is all the way left.



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 = pigpo.pi()
23
24

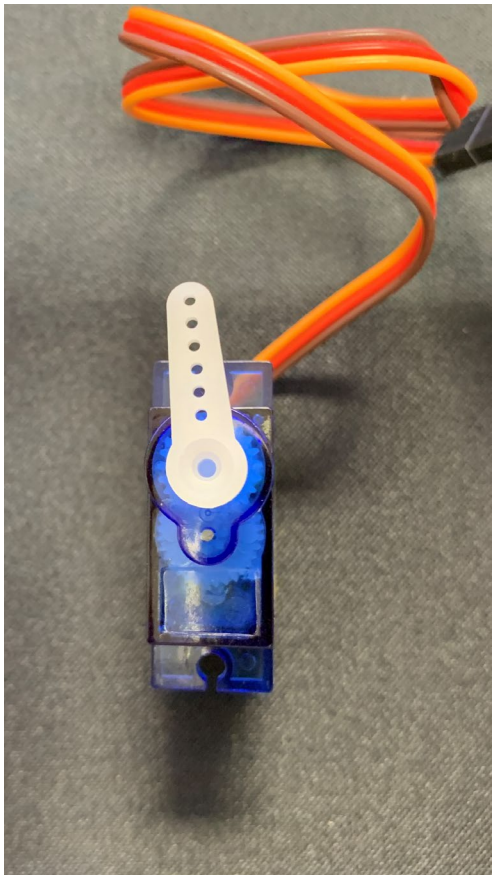
Servo Motor SG90

Pulse width = 2400us

Angle = 90 deg.

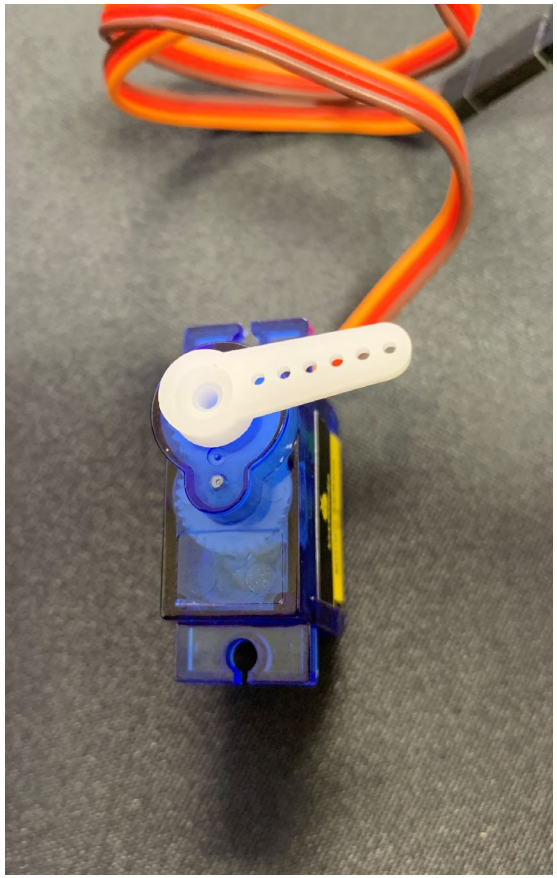
Pulse width = 1450us

Angle = 0 deg.



Pulse width = 500us

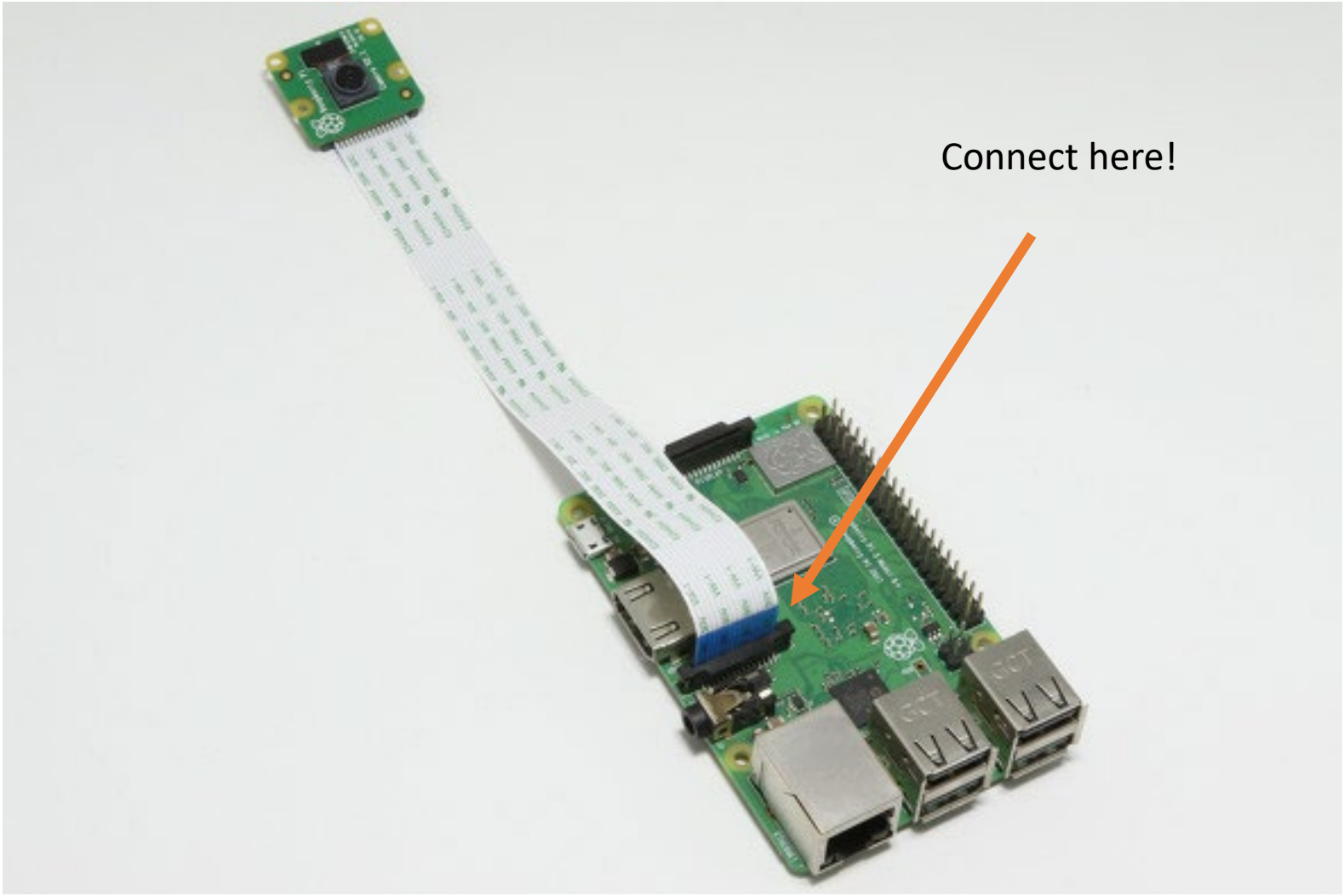
Angle = -90 deg.



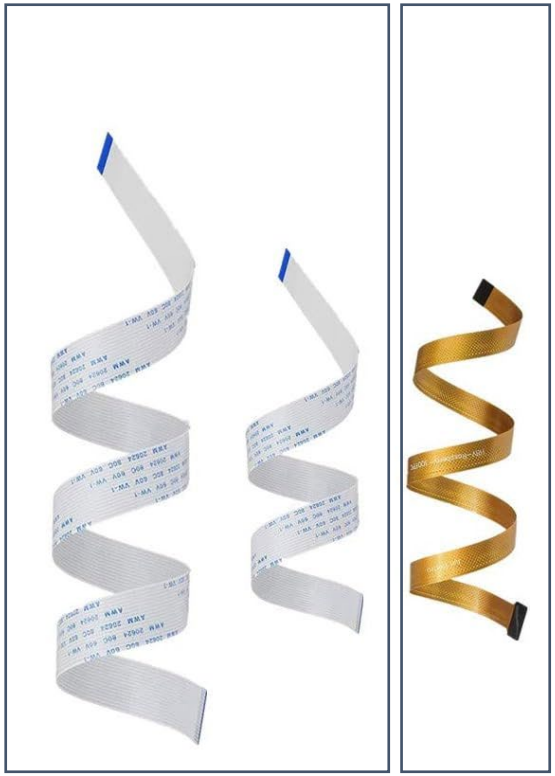
What happen if you input the pulse width = 3000 ?

Camera module

RasTech Raspberry Pi Camera Module



RasTech Raspberry Pi Camera Module v1 (ov5647)

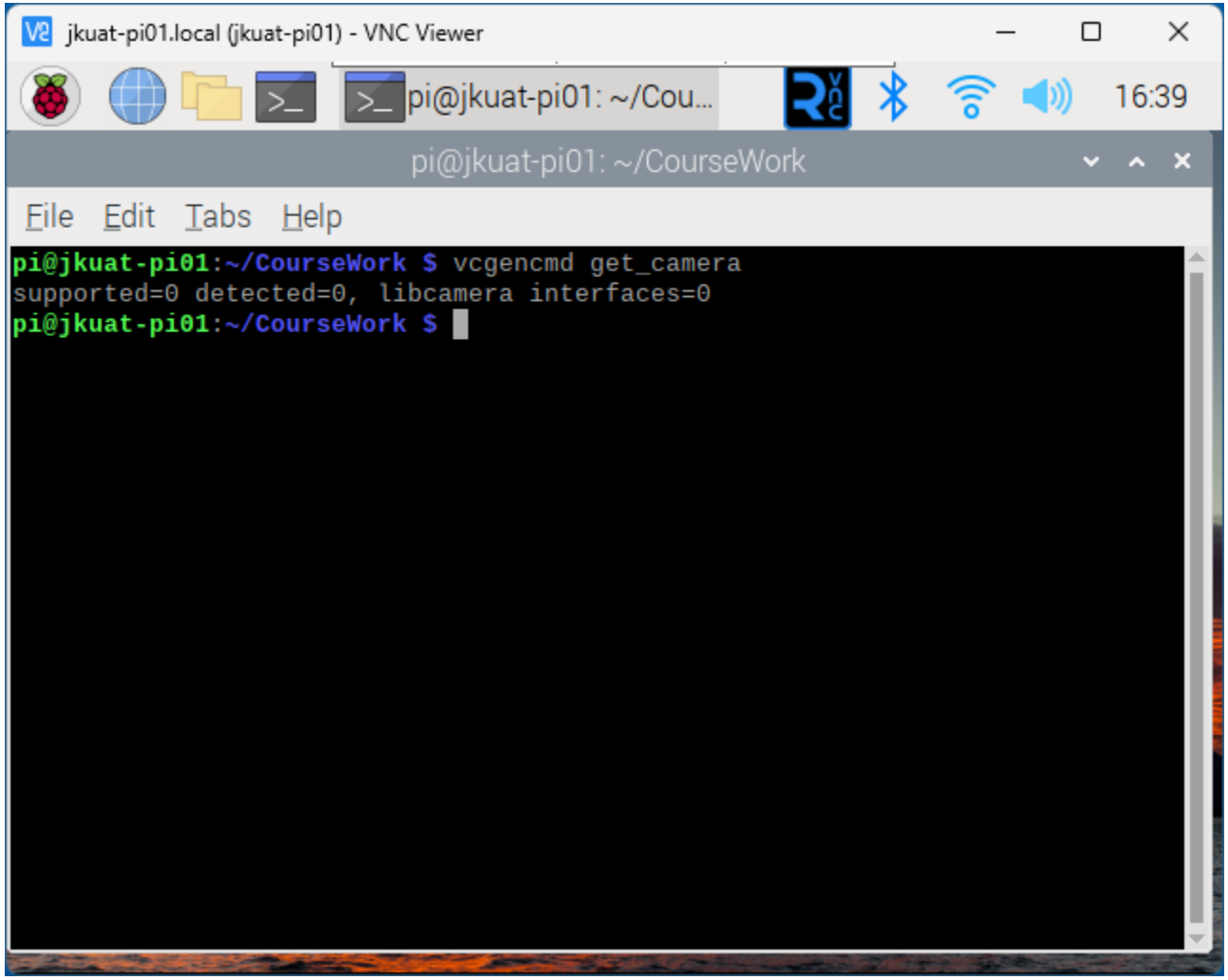


For pi3/pi4 For pi zero

6

vcgencmd get_camera

vcgencmd get_camera



libcamera-apps

sudo raspi-config
vcgencmd get_camera

libcamera-hello // check the camera

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

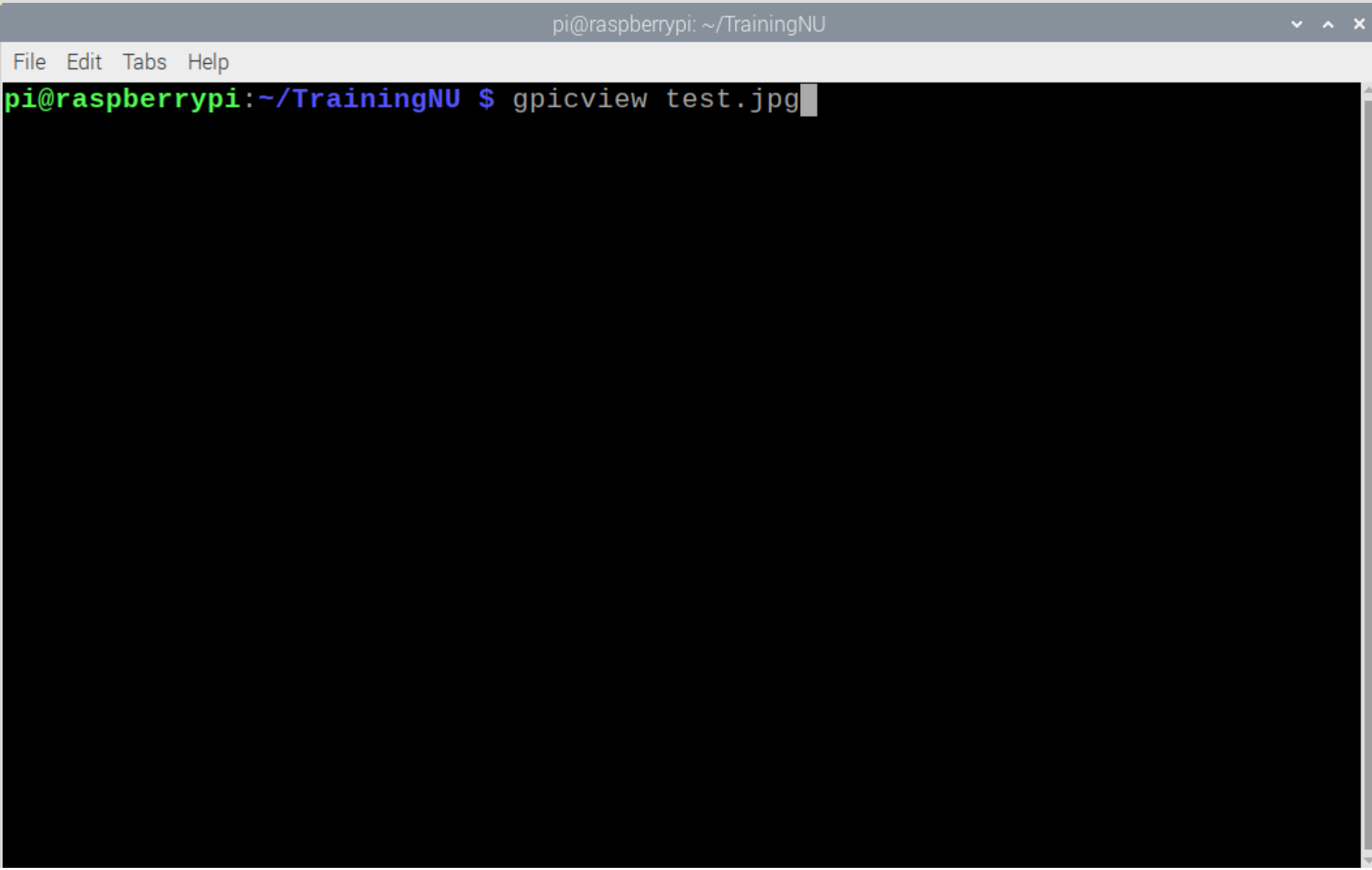
libcamera-jpeg // save as a jpeg format

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

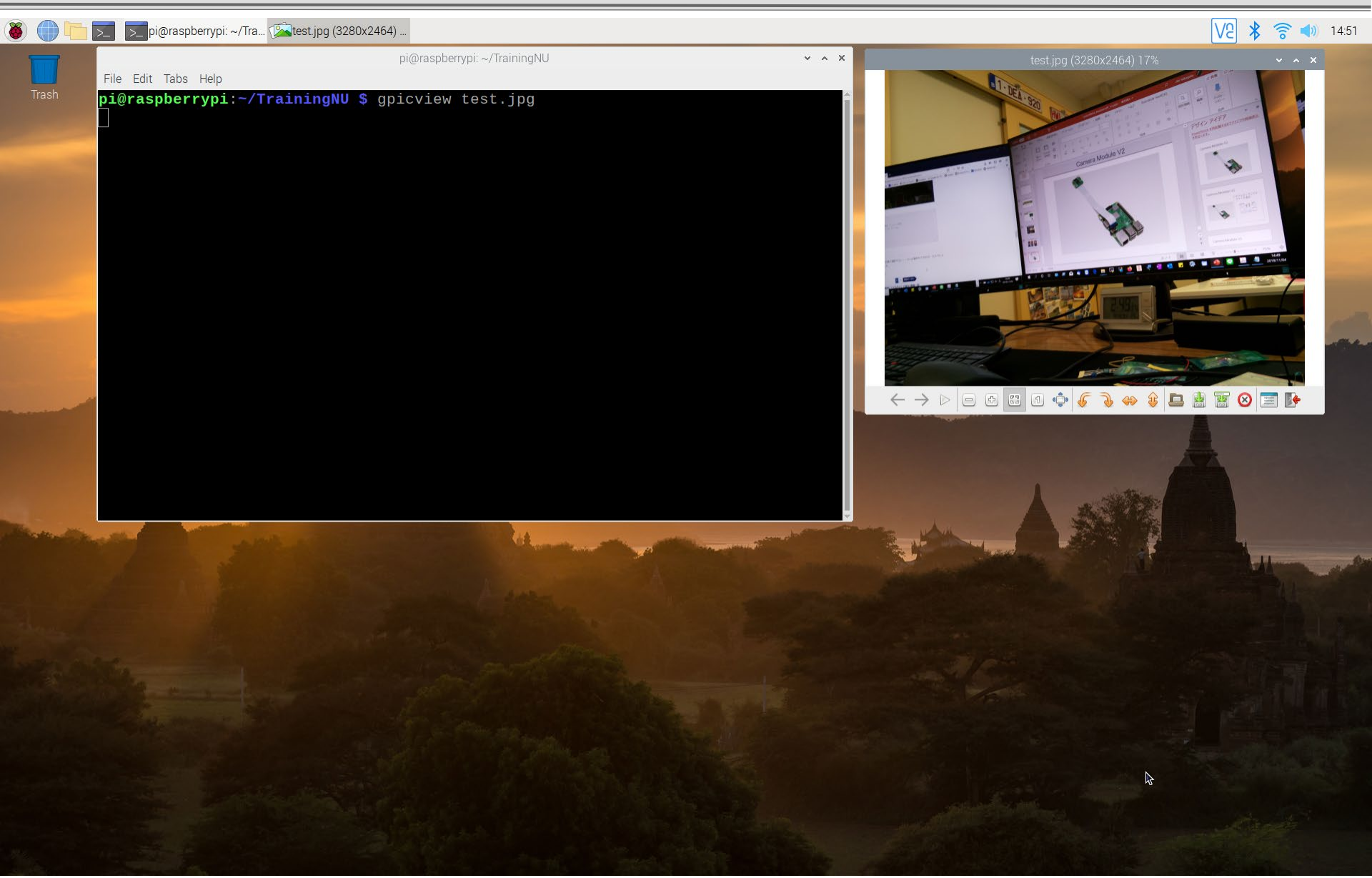
libcamera-still //

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

Linux command : gpicview

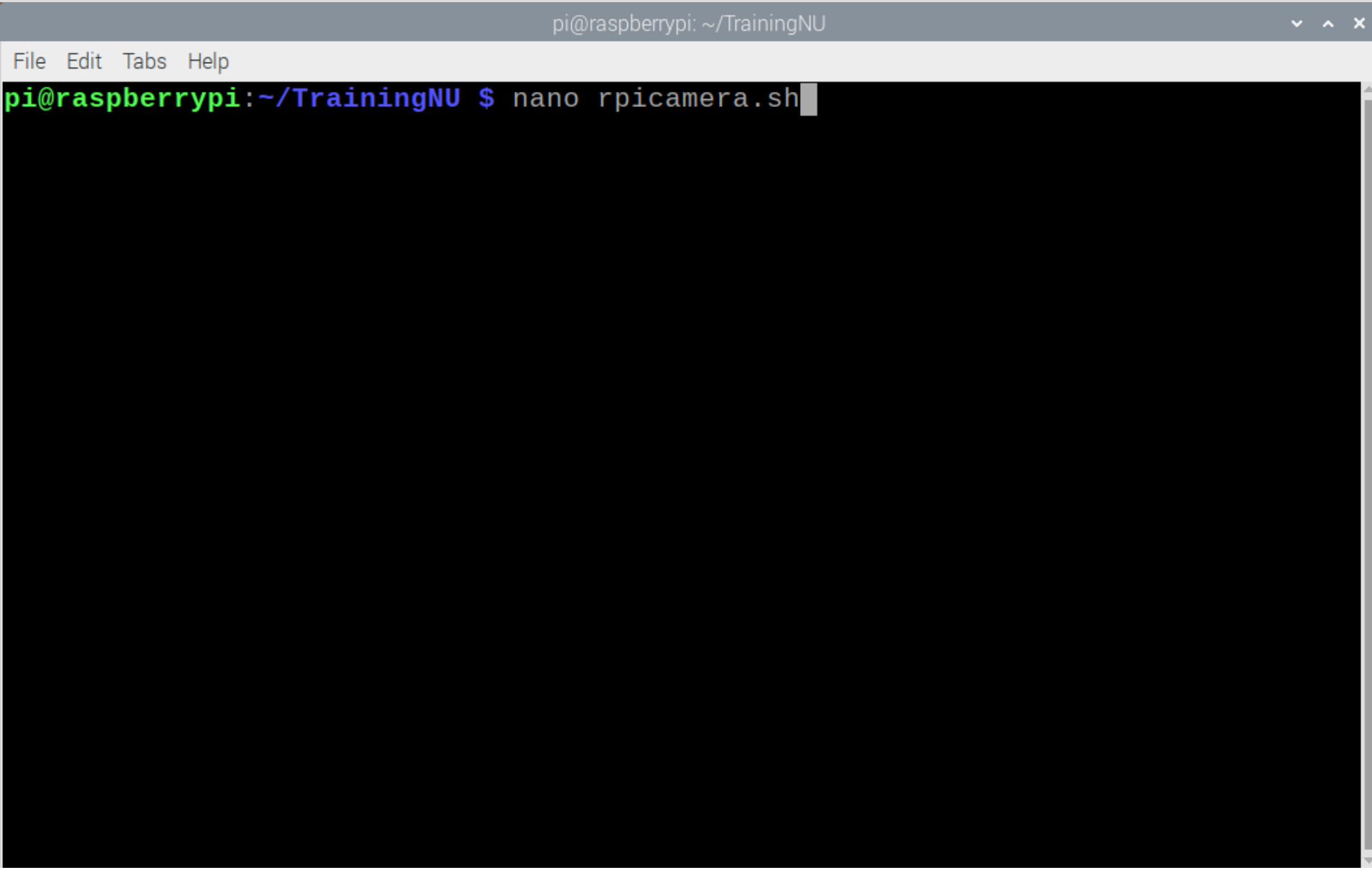


Linux command : gpicview



Nagasaki University

Shell program : 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

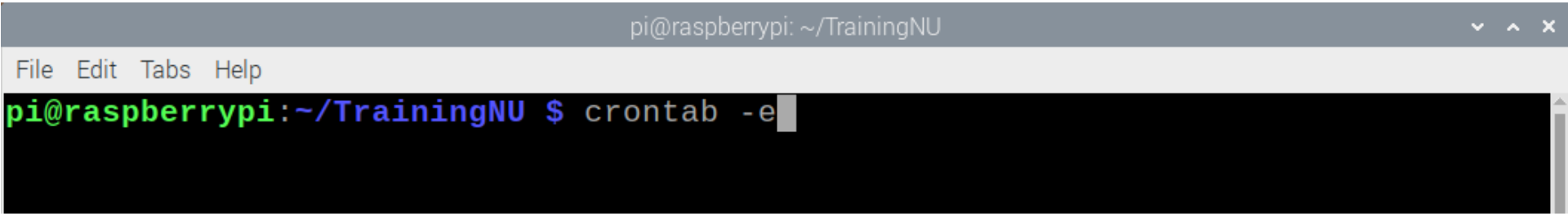
pi@raspberrypi: ~/TrainingNU

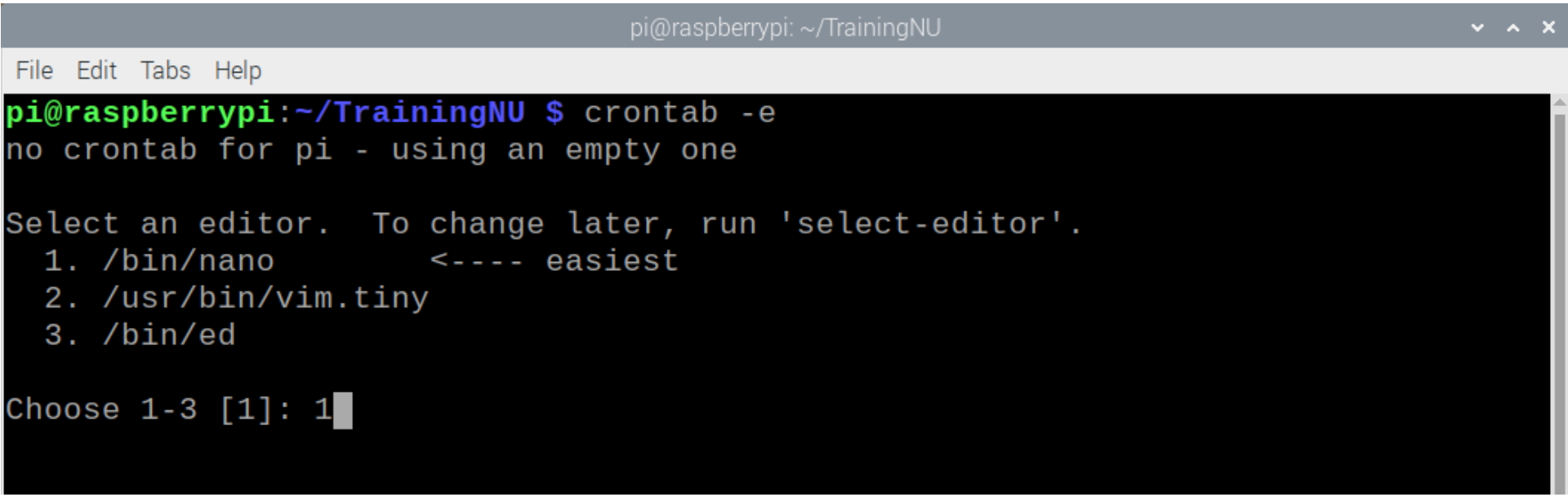
File Edit Tabs Help

```
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





*/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

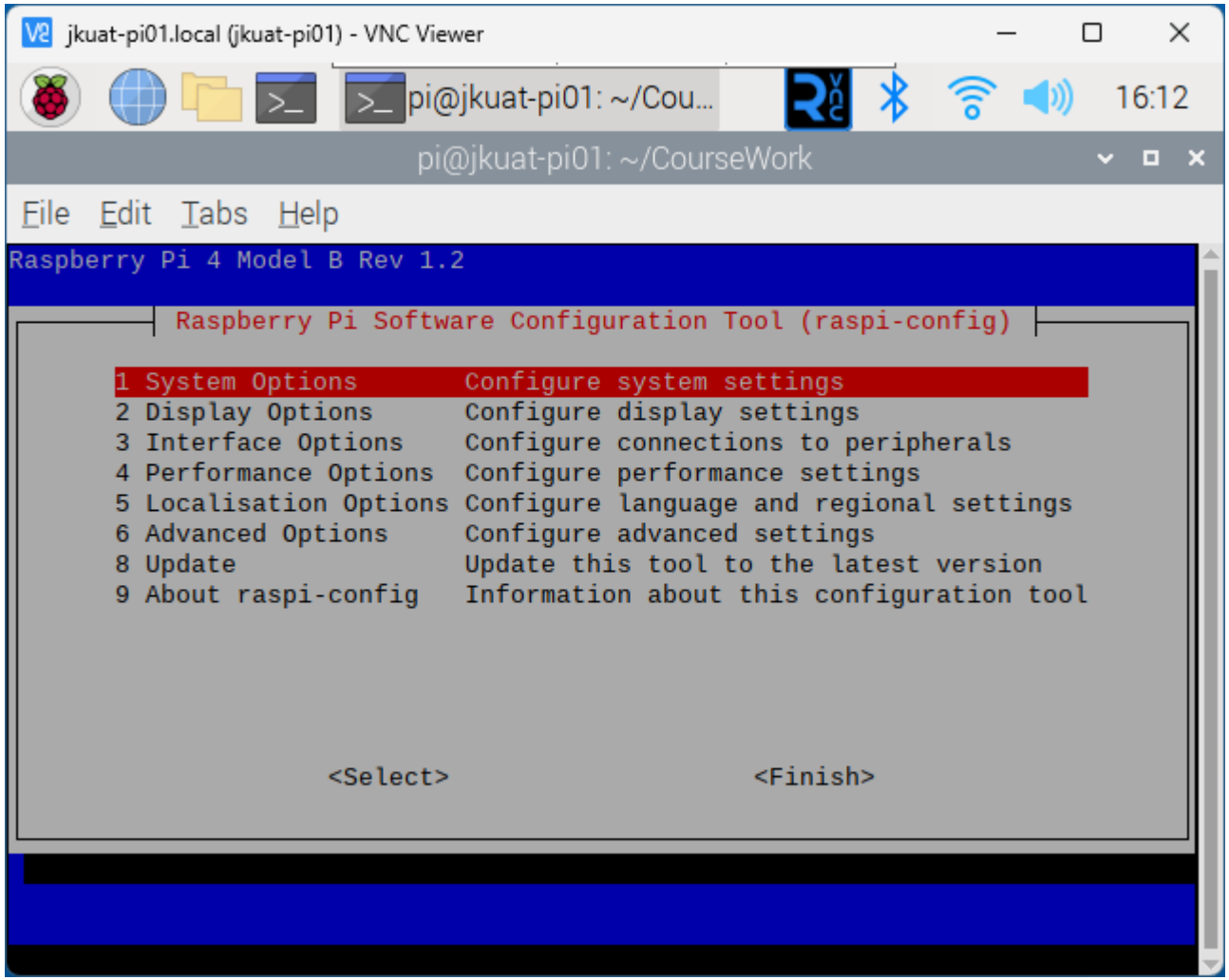
*/1 * * * * /home/pi/rpicamera.sh

means comment out

OpenCV

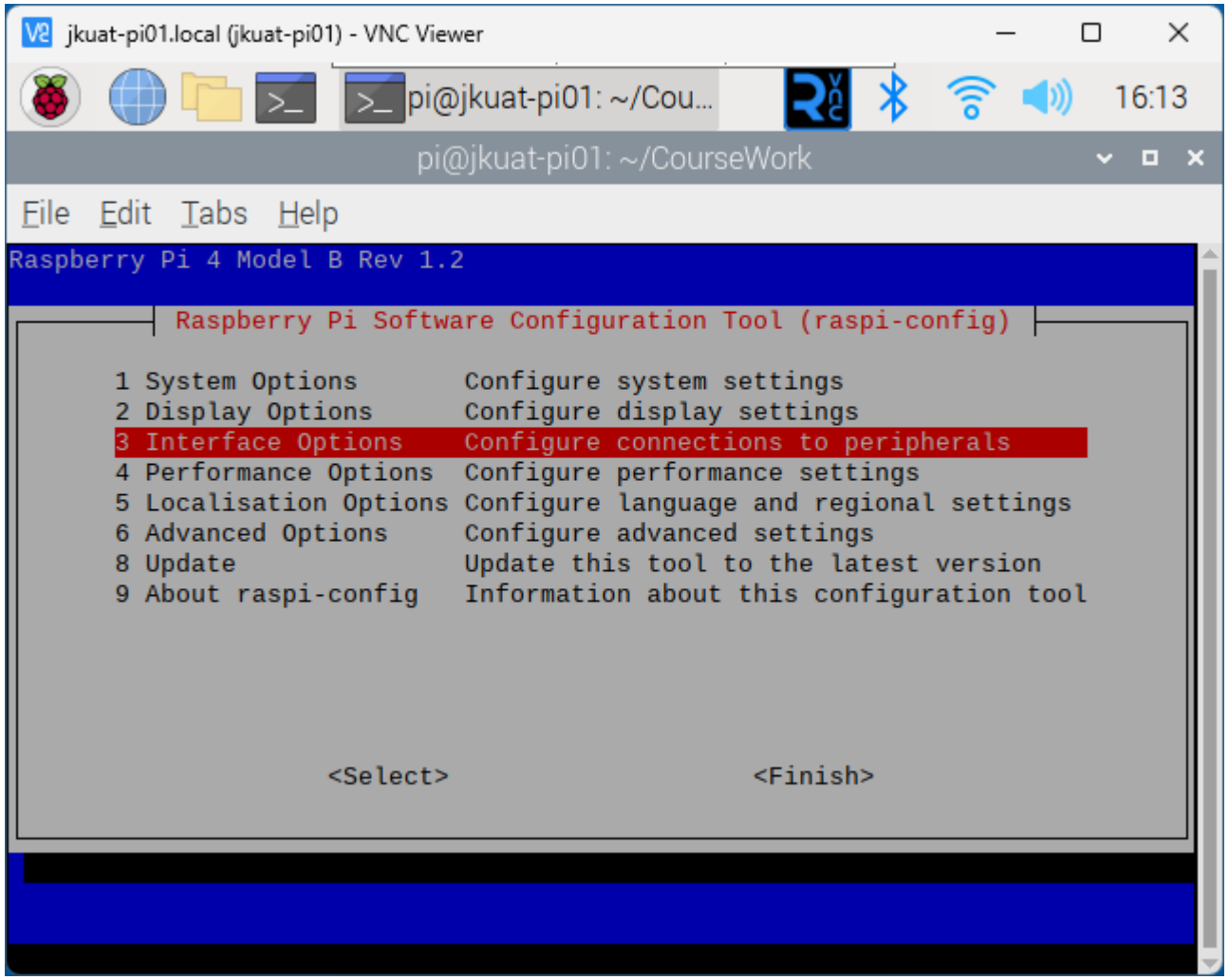
sudo raspi-config

sudo raspi-config



sudo raspi-config

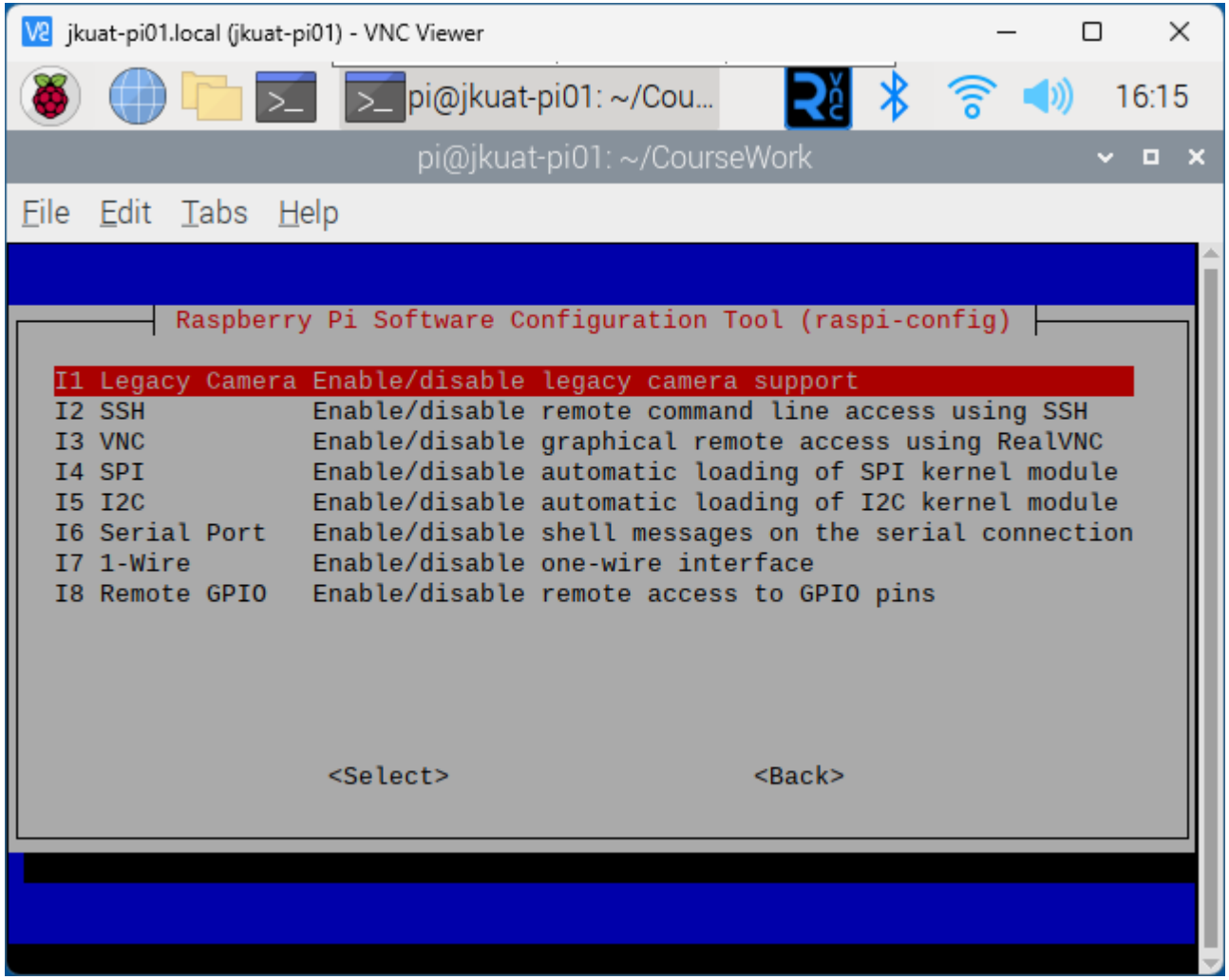
Select
3 Interface Options



sudo raspi-config

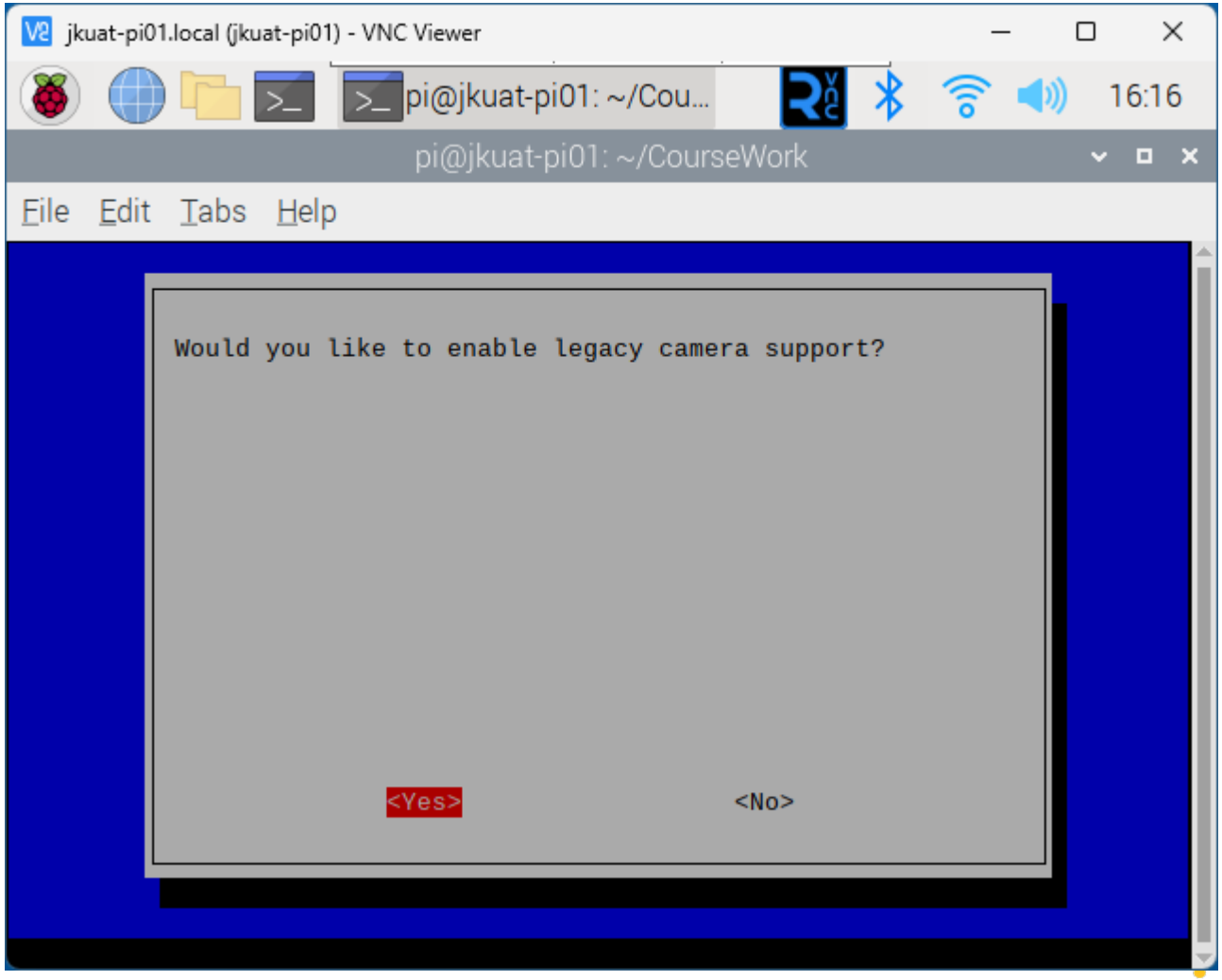
Select

I1 Legacy Camera Enable/disable legacy camera support

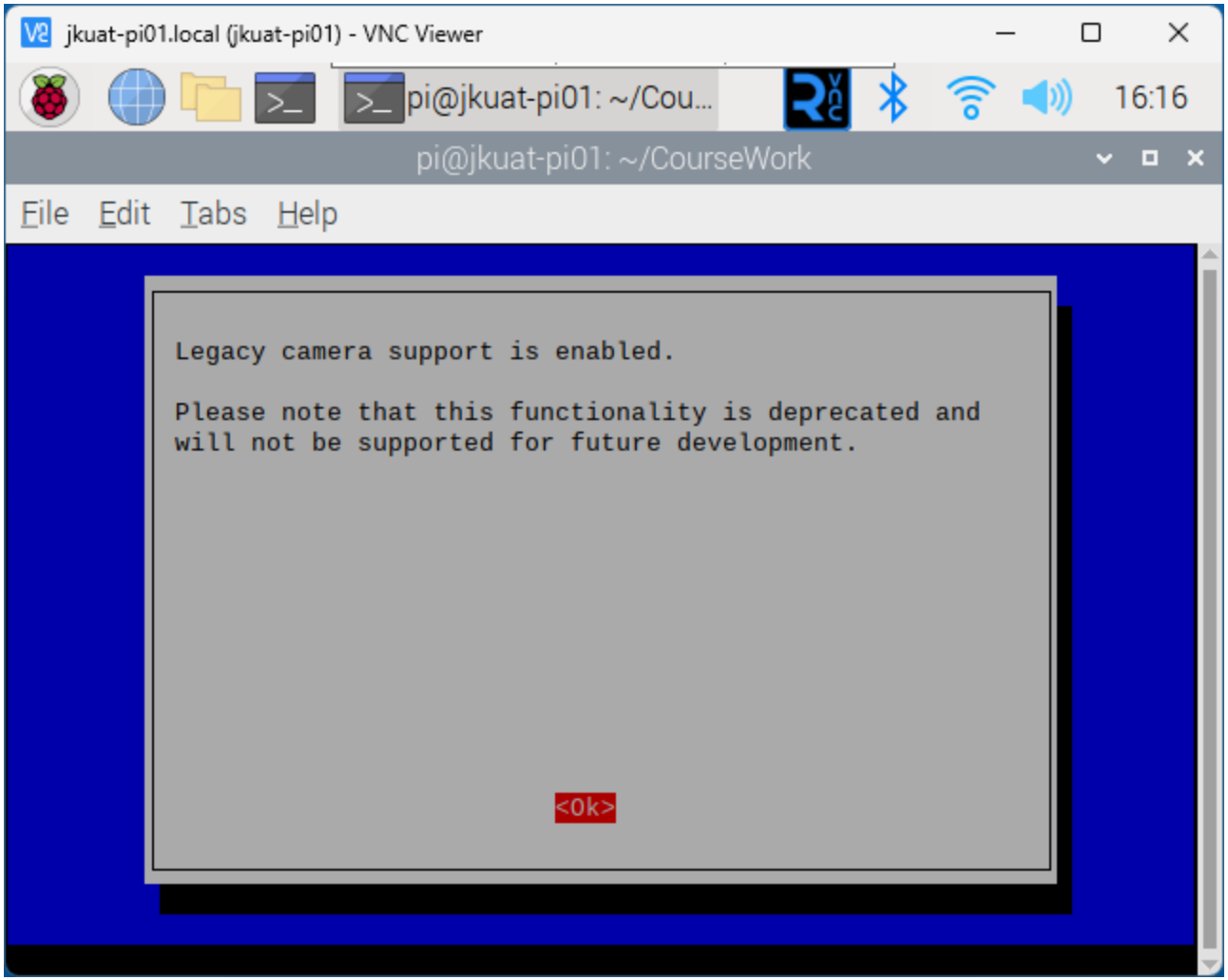


sudo raspi-config

Select
Yes

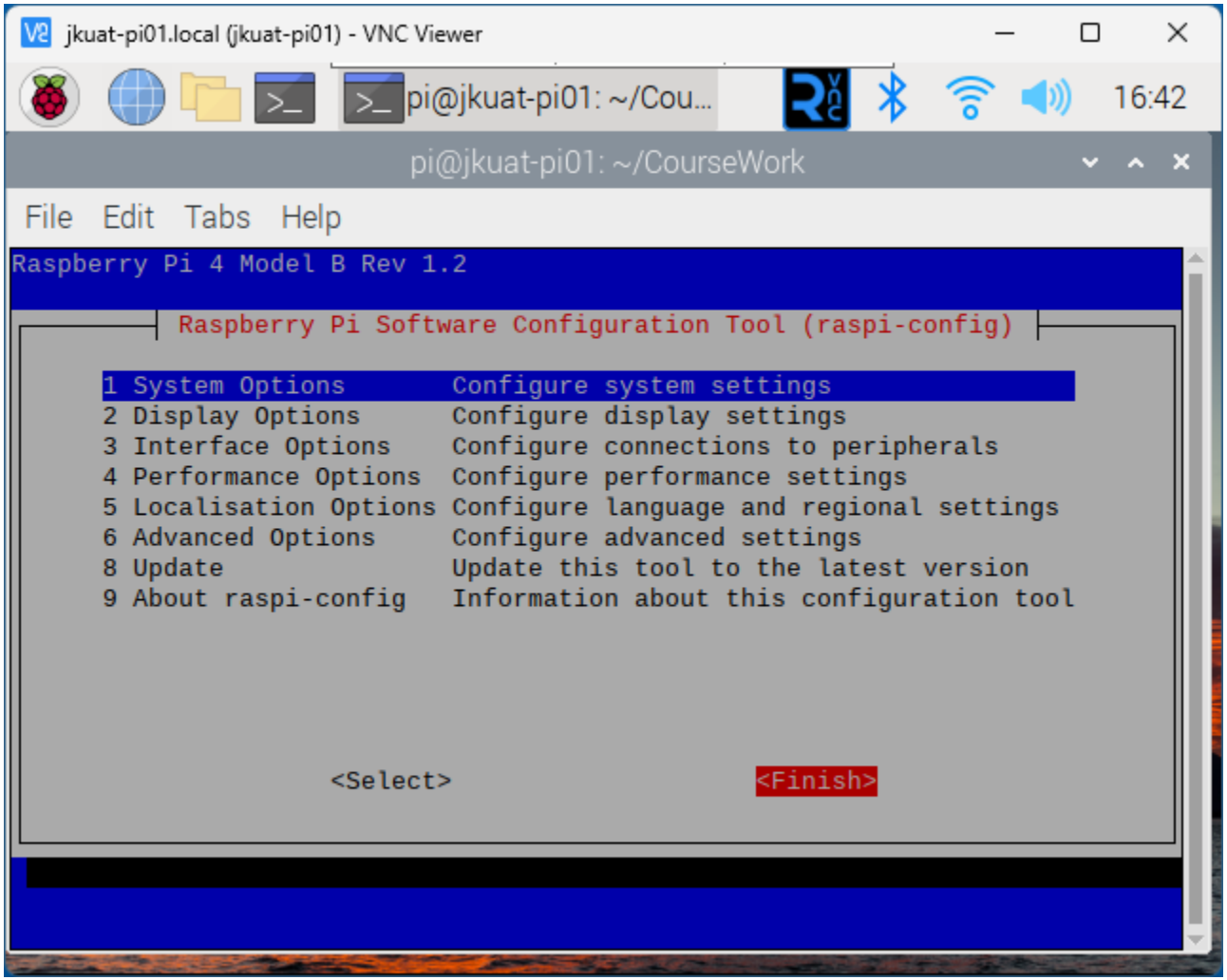


sudo raspi-config



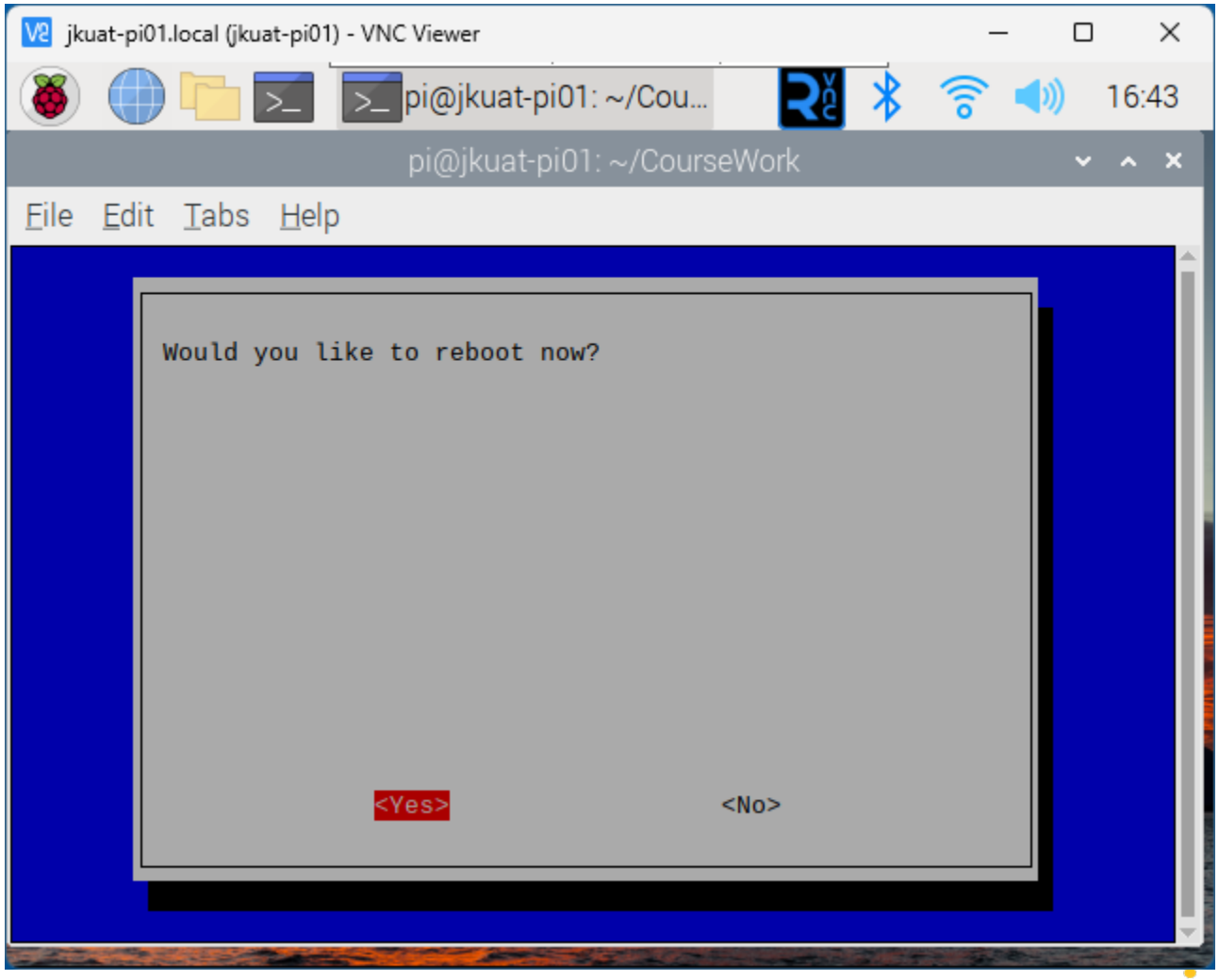
sudo raspi-config

Select
Finish



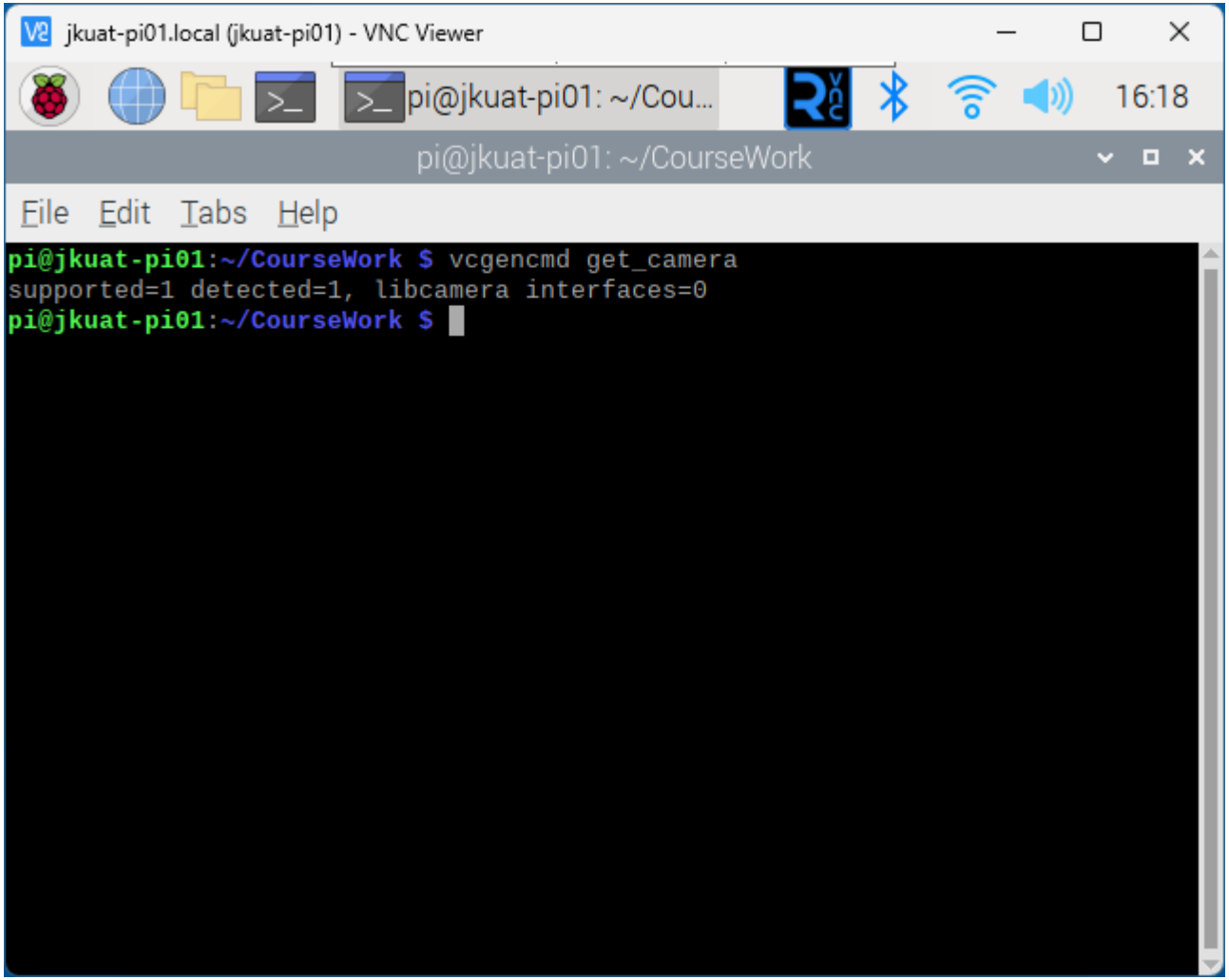
sudo raspi-config

Select
Yes



vcgencmd get_camera

vcgencmd get_camera



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

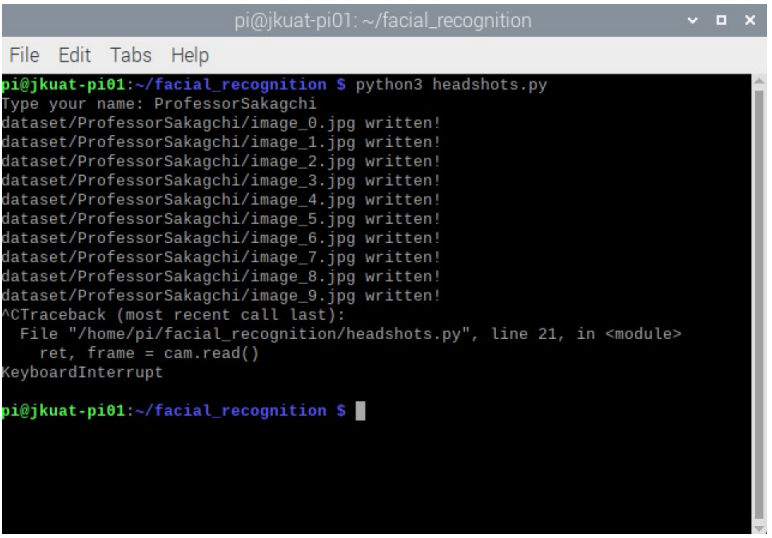
```
#Check
python3
 >>> import cv2    # please check no error message
 >>> CTRL+d
```

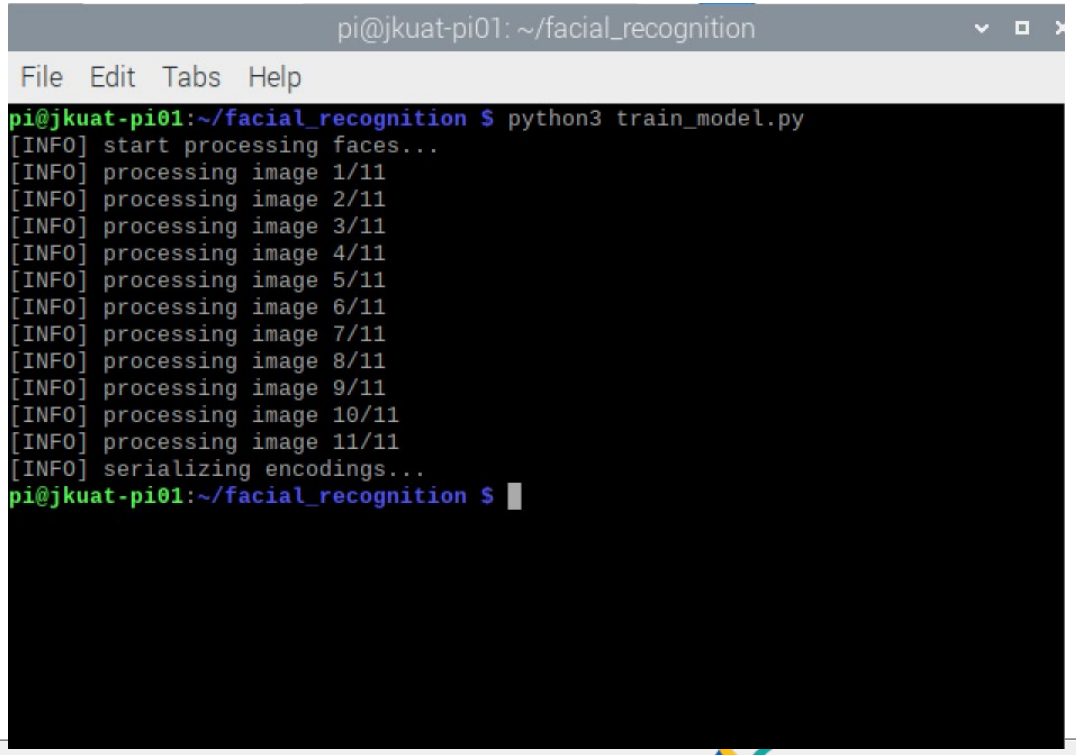
setup OpenCV for facial recognition

#----------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	

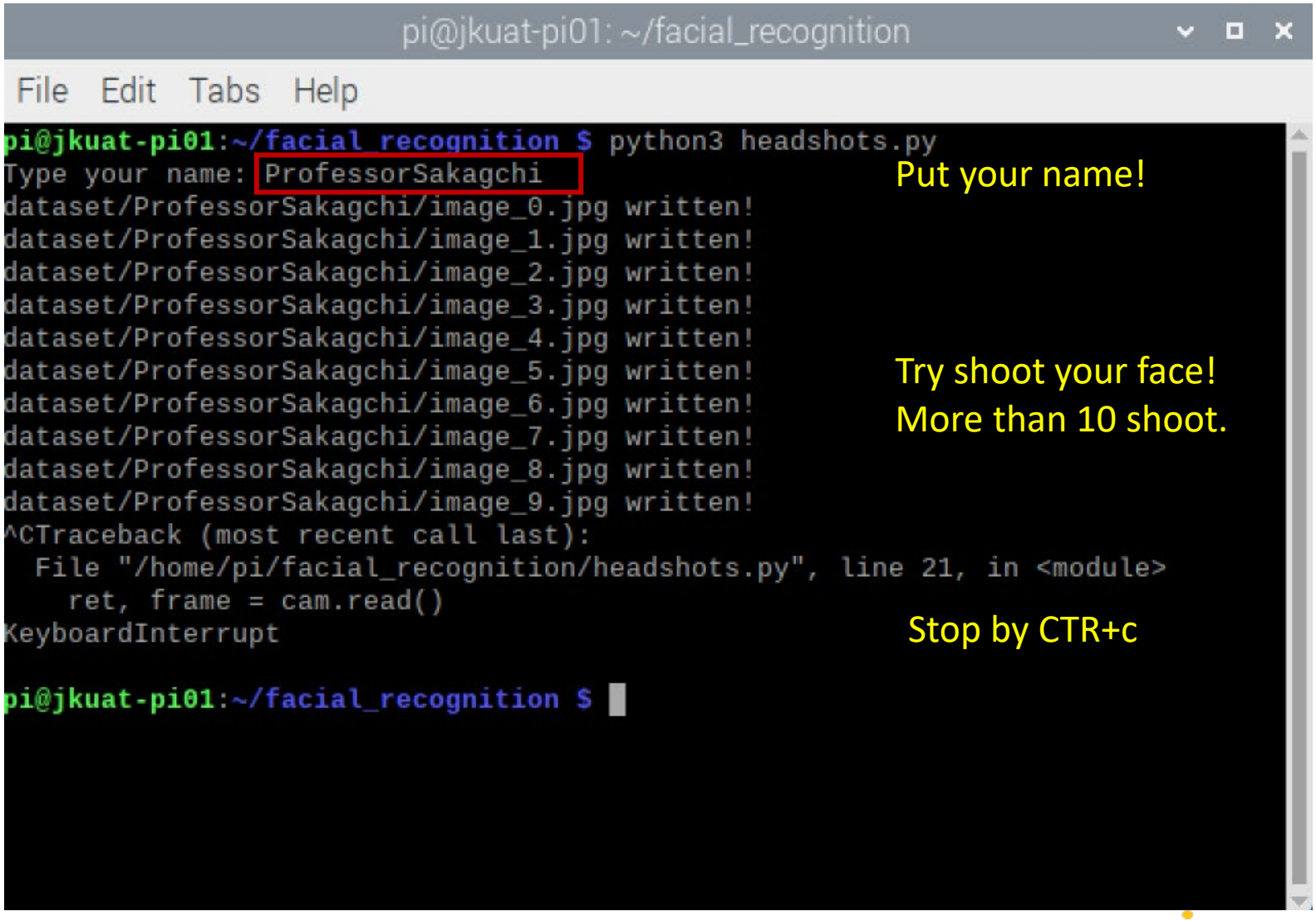




8

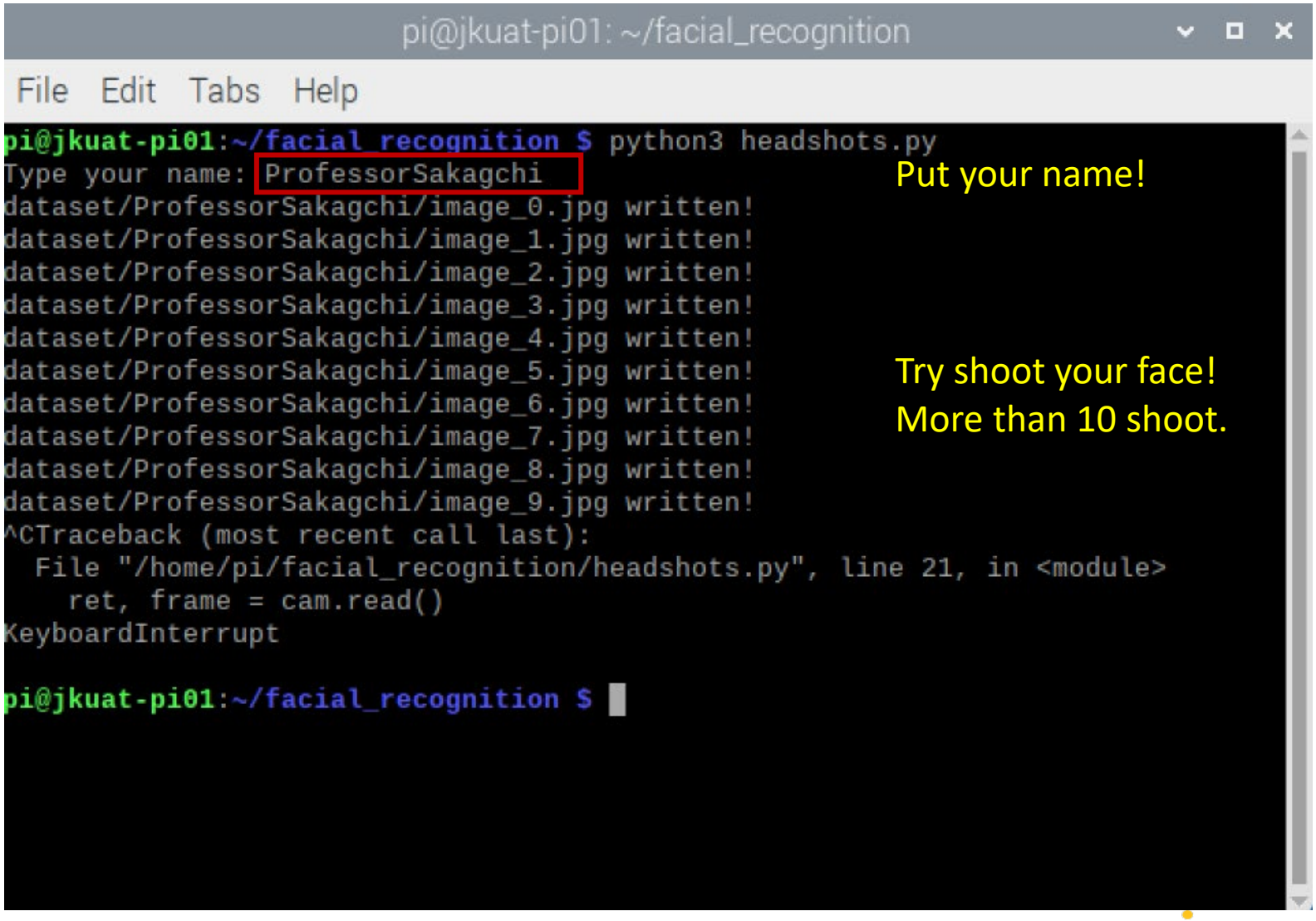
1st step : python3 headshots.py

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



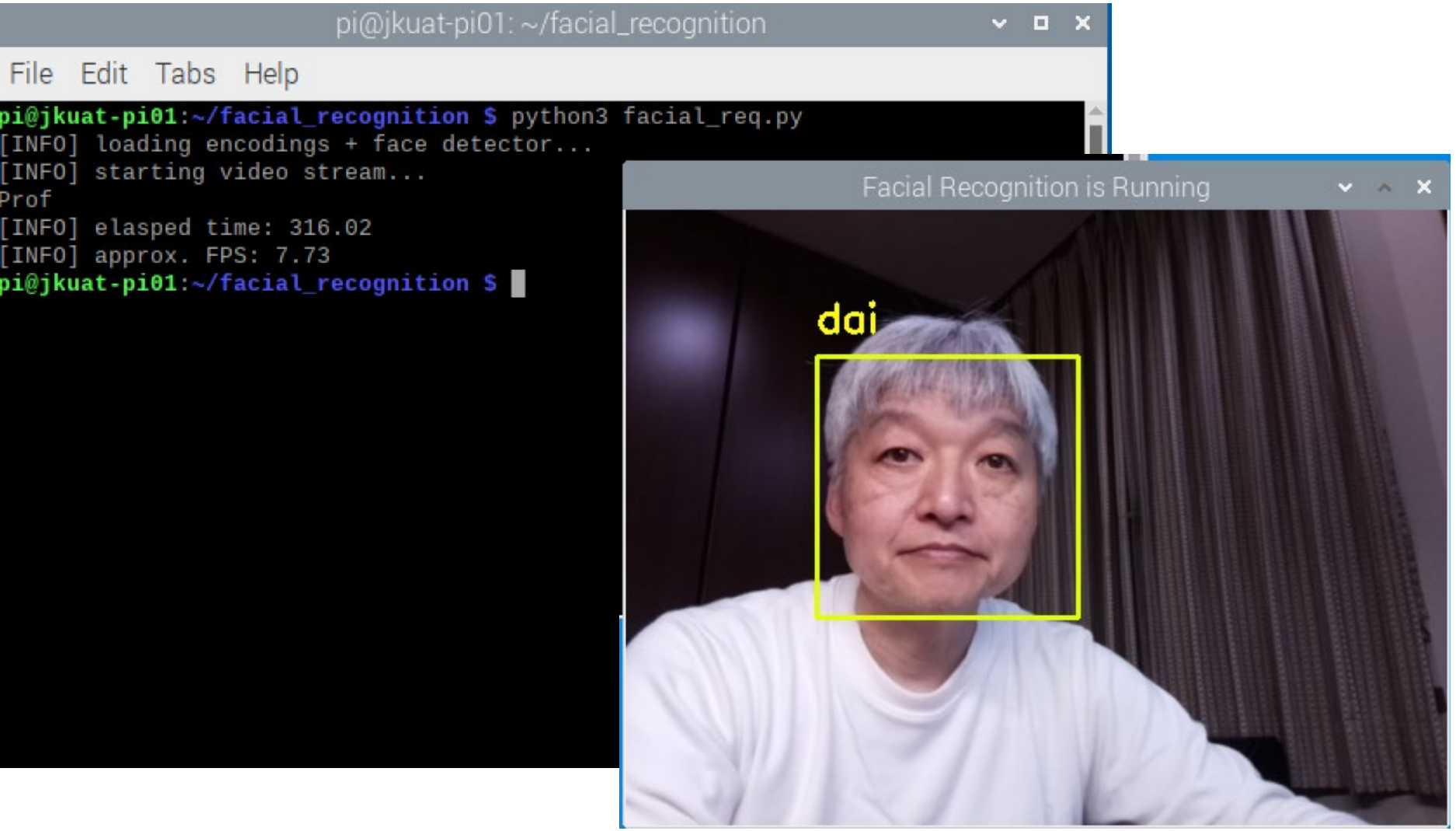
2nd step : python3 train_model.py

python3 train_model.py　　# training　　#generate encodings.pickle



3rd step : python3 facial_req.py

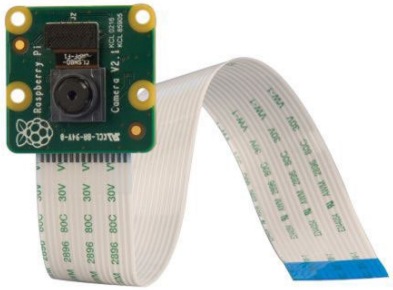
python3 facial_req.py # facial recognition



Nagasaki University

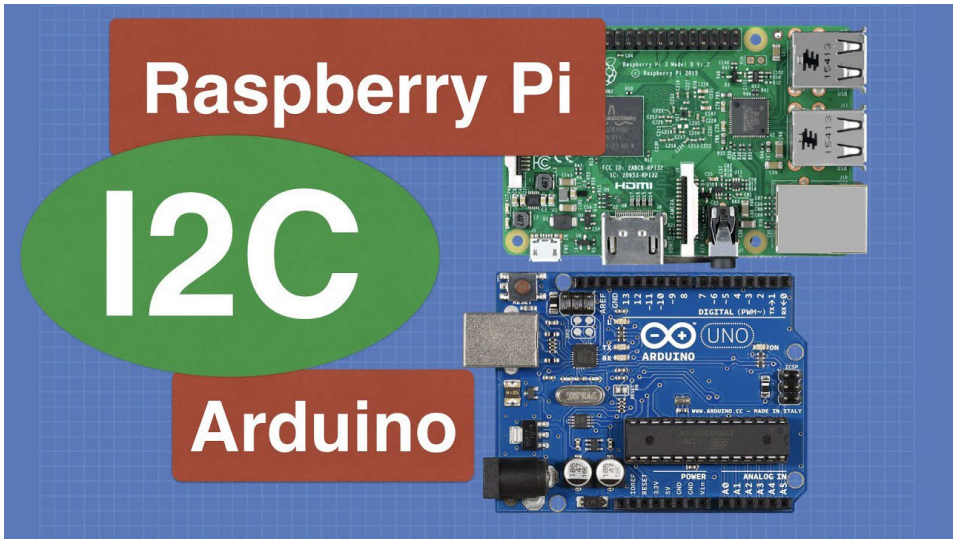
Wrap up

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

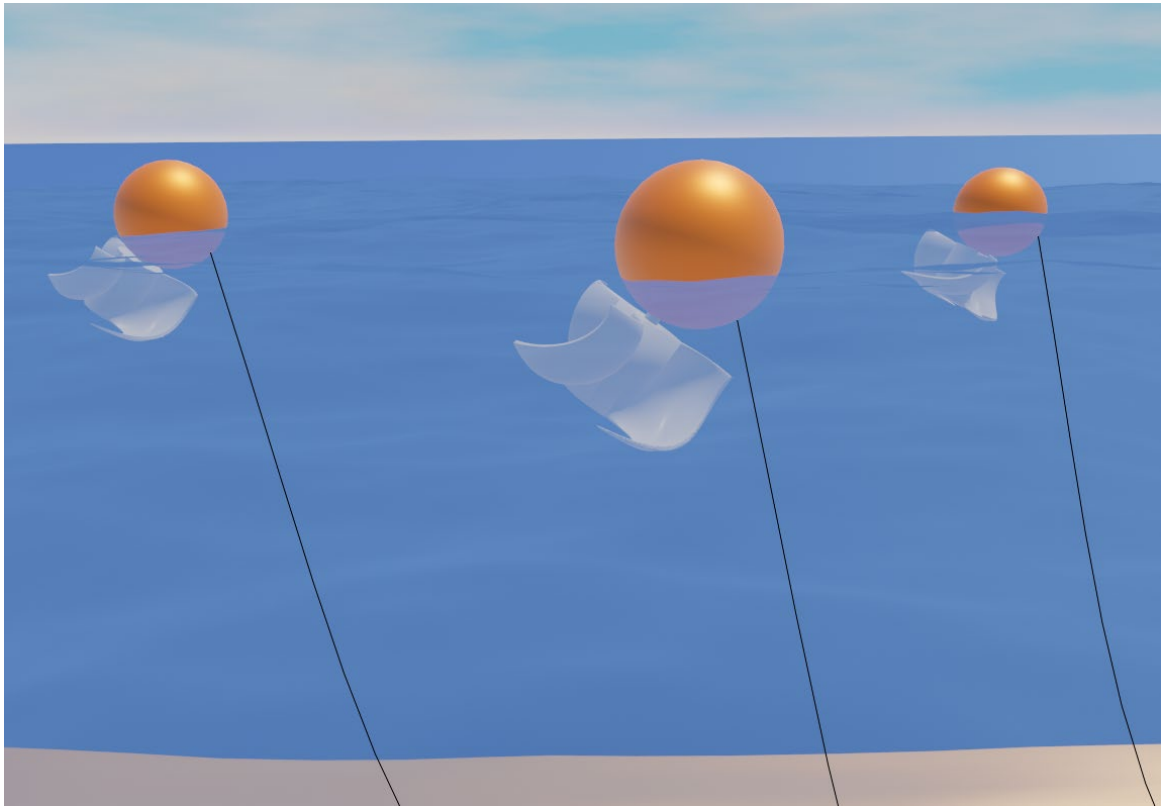


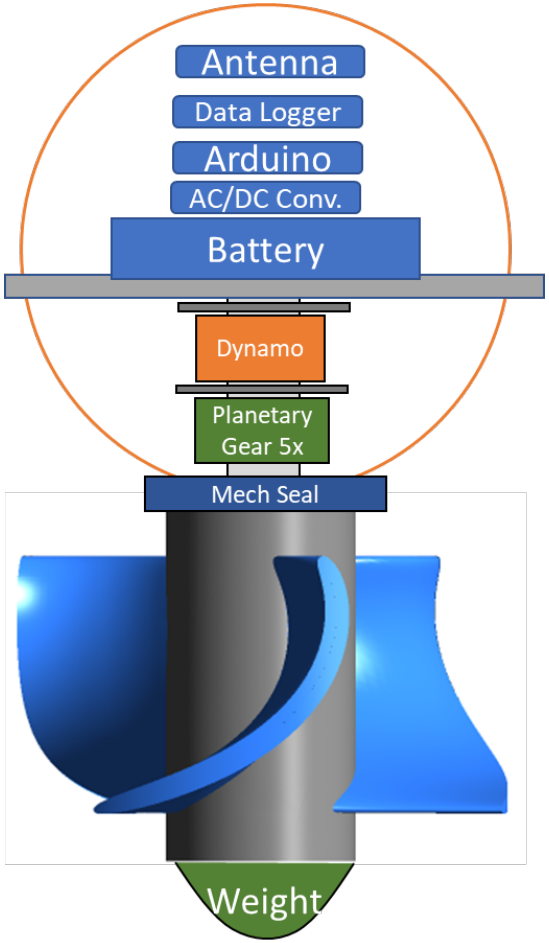
Next challenge

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



Smart Buoy

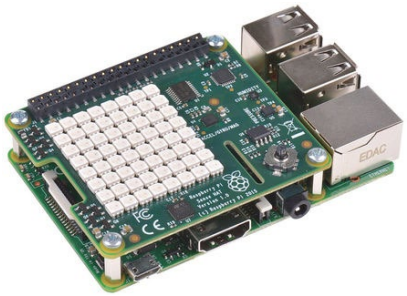




スマートブイ

Nagasaki University

IF + This Then That



Data sensing
Upload daemon by crond

IFTTT
Do more with the services you love

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()

#sense.show_message("Start ")

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

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

IFTTT_URL_GoogleSheets = 'https://maker.ifttt.com/trigger/REPORT1/with/key/'
IFTTT_KEY = 'm8yEHDspFvkF6o_0yujXRU6uFsqaGLR0EcjjirEs1Ed'
requests.post(IFTTT_URL_GoogleSheets+IFTTT_KEY, json={ 'value1':  temp, 'value2' : press, 'value3' : hum})

IFTTT_URL_GoogleSheets = 'https://maker.ifttt.com/trigger/REPORT2/with/key/'
IFTTT_KEY = 'm8yEHDspFvkF6o_0yujXRU6uFsqaGLR0EcjjirEs1Ed'
requests.post(IFTTT_URL_GoogleSheets+IFTTT_KEY, json={ 'value1':  p, 'value2' : r, 'value3' : y})

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

Initialization SenseHat

Sampling : Temp, Press, Humidity

Sampling : Angle of Yaw, Pitch and Roll

Report

Report2

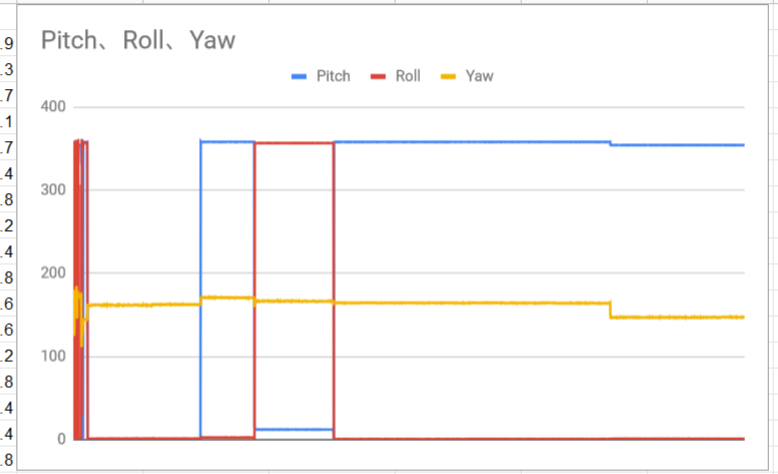
Data Plotting by Google spread sheet

Angle_of_Turbine

ファイル 編集 表示 挿入 表示形式 データ ツール アドオン ヘルプ 最終編集: 2018年10月1日

共有

	A	B	C	D	E
1			Pitch	Roll	Yaw
2	September 29, 2018 at 12:52	REPORT2	0.4	1	177.9
3	September 29, 2018 at 12:53	REPORT2	4.1	10.7	180.3
4	September 29, 2018 at 01:02	REPORT2	359.2	359.3	124.7
5	September 29, 2018 at 01:02	REPORT2	3.5	1.4	137.1
6	September 29, 2018 at 01:02	REPORT2	358.2	353.8	174.7
7	September 29, 2018 at 01:03	REPORT2	38.6	14.1	176.4
8	September 29, 2018 at 01:03	REPORT2	358.9	359.4	177.8
9	September 29, 2018 at 01:03	REPORT2	359	2.2	176.2
10	September 29, 2018 at 01:04	REPORT2	358	0.5	175.4
11	September 29, 2018 at 01:05	REPORT2	37.4	57.5	184.8
12	September 29, 2018 at 01:06	REPORT2	47.6	48.4	149.6
13	September 29, 2018 at 01:07	REPORT2	351.9	349.4	145.6
14	September 29, 2018 at 01:08	REPORT2	353.8	353.1	179.2
15	September 29, 2018 at 01:09	REPORT2	354.6	359.8	172.8
16	September 29, 2018 at 01:10	REPORT2	354.6	359.9	173.4
17	September 29, 2018 at 01:11	REPORT2	354.4	0.1	172.4
18	September 29, 2018 at 01:12	REPORT2	354.4	0.2	171.8
19	September 29, 2018 at 01:13	REPORT2	354.4	0.1	172
20	September 29, 2018 at 01:14	REPORT2	354.4	0.2	172.2
21	September 29, 2018 at 01:15	REPORT2	354.5	0.2	171.3
22	September 29, 2018 at 01:16	REPORT2	354.5	0.3	171.9
23	September 29, 2018 at 01:17	REPORT2	354.5	0.2	171.6
24	September 29, 2018 at 01:18	REPORT2	354.4	0.3	171.8
25	September 29, 2018 at 01:19	REPORT2	337	307.1	177.1
26	September 29, 2018 at 01:20	REPORT2	330.6	29.3	111.1
27	September 29, 2018 at 01:21	REPORT2	359.5	359.9	127
28	September 29, 2018 at 01:22	REPORT2	359.5	359.9	126.4



シート1

データ探索

Nagasaki University

Smart Mirror

Date Clock & Weather forecast

Raspberry Pi

Nagasaki University

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