

## BÁO CÁO TUẦN 1

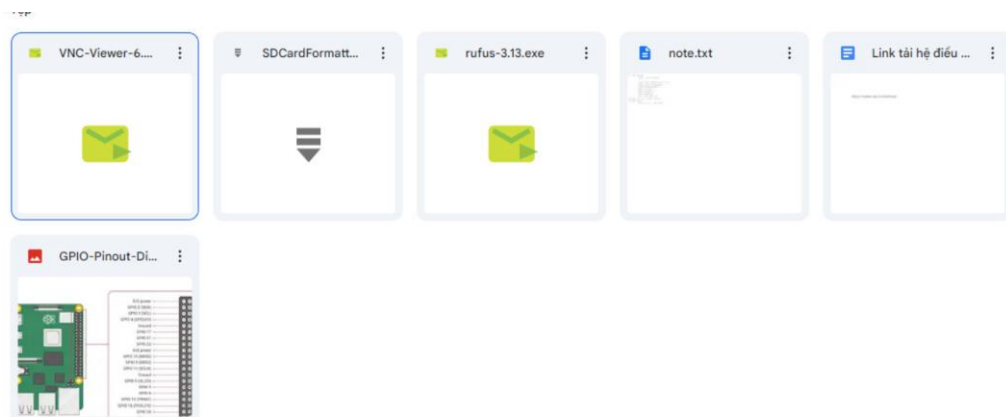
NHÓM 4	
Họ và tên	MVVS
Võ Minh Thuận	21161366
Lê Quang Thương	21161363
Trần Thị Xuân Hy	21161323

### 1. Trình bày các bước format và chép hệ điều hành vào thẻ nhớ:

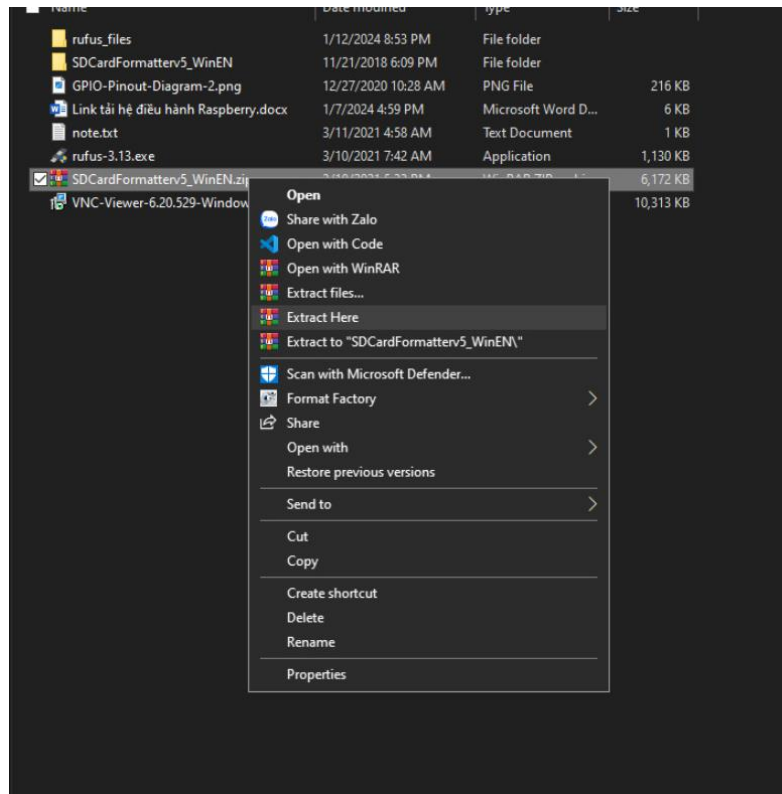
- **Bước 1:** Đăng nhập vào UTEX sau đó tải các phần mềm được đính kèm trong link đính kèm ở mục “Phần mềm học tập” về máy



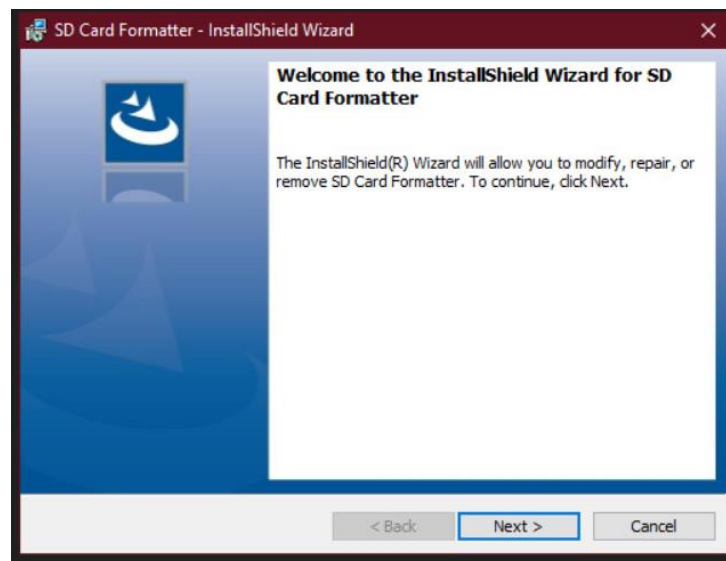
+ Nhấn vào mục “Các phần mềm hỗ trợ cho môn học”



- **Bước 2:** Giải nén các file.zip

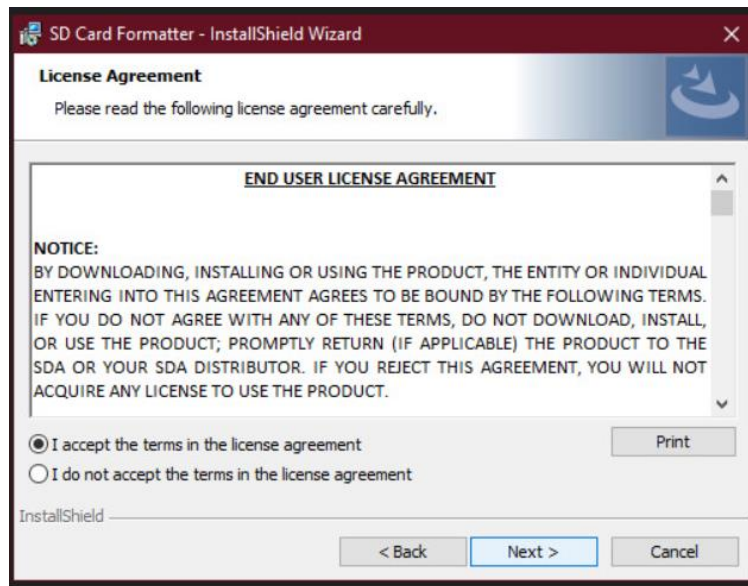


- **Bước 3:** Bắt đầu cài đặt phần mềm SD Card Formatter 5.0.1 Setup.exe



+ Chọn **Next** để tiếp tục

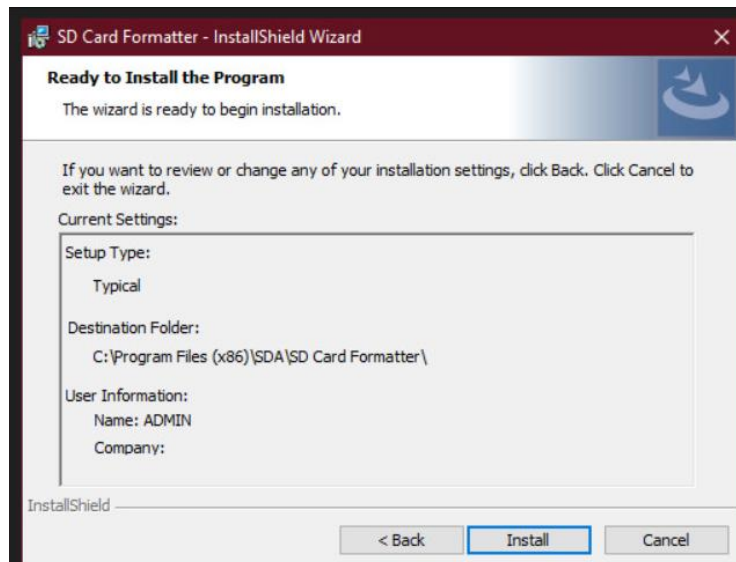
+ Click chọn ***I accept the terms in the license agreement*** sau đó chọn **Next**



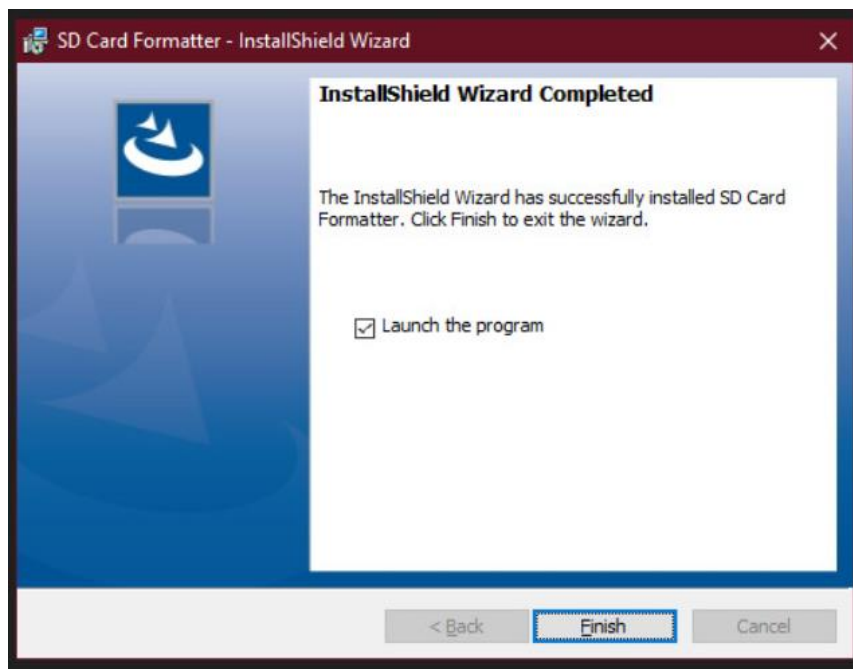
+ Chọn **Next** để tiếp tục



+ Click chọn **Install** và chờ đợi cài đặt



+ Click chọn **Finish** để hoàn tất cài đặt

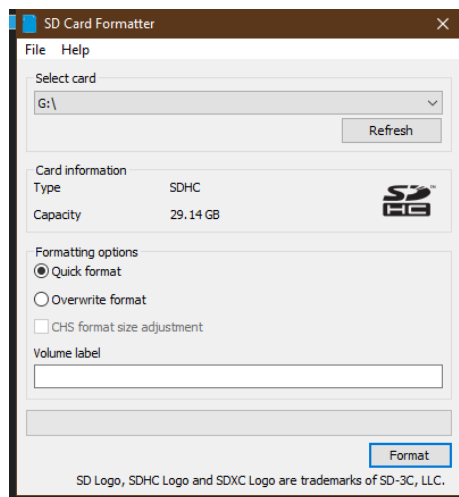


- **Bước 4:** Chuẩn bị sẵn thẻ nhớ (dạng thẻ nhớ SD class 10) để cài hệ điều hành và đầu đọc thẻ nhớ để laptop đọc và ghi dữ liệu trên thẻ nhớ đó.

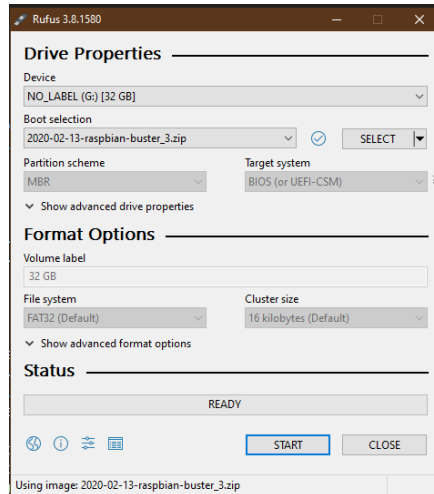


*Hình 1: Thẻ nhớ và đầu đọc thẻ nhớ*

- **Bước 5:** Cài những phần mềm cần thiết để cài hệ điều hành như: SD Card Formatter để format thẻ nhớ, Rufus để cài hệ điều hành từ file iso đã tải lúc đầu và ghi file vào thẻ nhớ SD bằng phần mềm Rufus và file iso (hệ điều hành Rasbpi os).



*Hình 2: Phần mềm SD Card Formatter*



*Hình 3: Phần mềm Rufus*

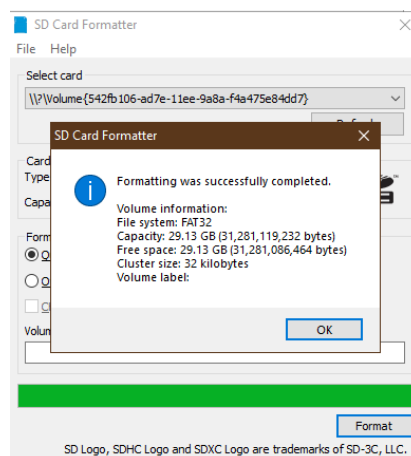
- **Bước 6:** Khi đủ những phần mềm và file hệ điều hành cần thiết rồi ta mở phần mềm Rufus và thực hiện theo các bước.

+ Chọn thẻ nhớ cần ghi file hệ điều hành (khi mở phần mềm tự tìm và chọn SD sẵn cho ta khi đã cắm thẻ nhớ vào laptop).



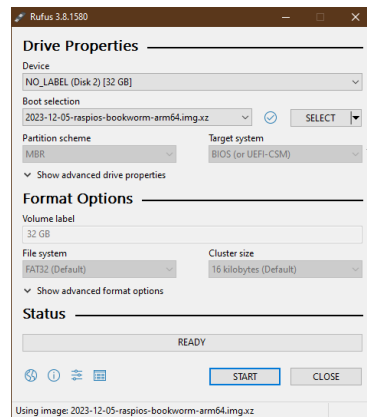
*Hình 4: Cắm thẻ nhớ vào đầu đọc thẻ và laptop*

+ Mở phần mềm SD Card Formatter và chọn phím Format để format thẻ nhớ.

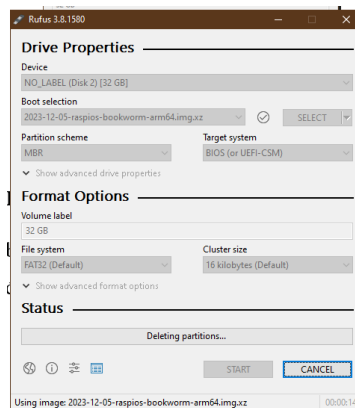


*Hình 5: Format thẻ nhớ thành công*

+ Nhấn nút SELECT ở phần Boot selection để chọn file iso (.zip,.iso,...) mà ta đã tải lúc đầu.



+ Nhấn START để bắt đầu ghi file vào thẻ nhớ.



- **Bước 7:** Đợi khi phần mềm báo Finish thì là đã ghi file thành công và ta có rút thẻ nhớ và gắn vào Rasbpi để setup hệ điều hành và sử dụng

## 2. Trình bày chi tiết từng bước (có hình ảnh cụ thể) cài đặt hệ điều hành cho Raspberry:

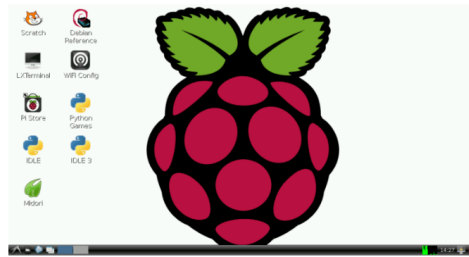
- **Bước 1:** Chuẩn bị Raspberry Pi, Màn hình, chuột, bàn phím, nguồn, dây HDMI, thẻ SD class 10.

- **Bước 2:** Tải hệ điều hành Raspberry OS (Raspbian) từ web <https://raspberrypi.vn/download>.

## 1. Raspbian

Raspbian là phiên bản hệ điều hành phổ biến nhất trên Raspberry Pi. Đa số người dùng sẽ được khuyến sử dụng phiên bản HĐH này. Nhưng tại sao?

Raspbian là một hệ điều hành dễ sử dụng và với sự hỗ trợ tốt của cộng đồng trên thế giới. Hơn thế nữa, Raspbian là phiên bản không chính thức của Debian Wheezy – một phiên bản Linux có tiếng. Raspbian rất đơn giản và quen thuộc. Nó là nền tảng rất tốt cho những người mới bắt đầu làm quen với Raspberry Pi nói riêng và Linux nói chung. Giống như một chiếc máy tin Windows, Raspbian bao gồm các ứng dụng đa phương tiện và đồ họa (xem ảnh, xem phim, soạn thảo notepad), và nếu bạn cần nhiều hơn, bạn có thể cài thêm các gói phần mềm khác như trình duyệt internet, nhận tin, bộ phần mềm văn phòng, v.v



*Tương thích:* Raspbian tương thích mọi phiên bản Raspberry Pi

*Download Raspbian:* Các bạn có thể tải phiên bản Raspbian mới nhất [TẠI ĐÂY](#).

*User:* pi

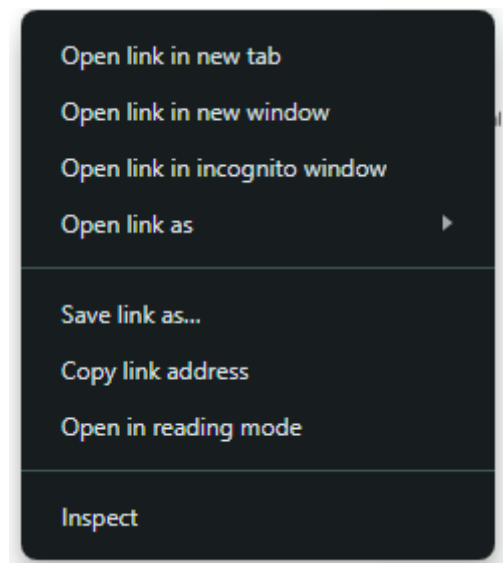
*Password:* raspberry

*Hình 6: Hình trang tải hệ điều hành*

+ Chọn nút tại đây để tải file hệ điều hành về máy.

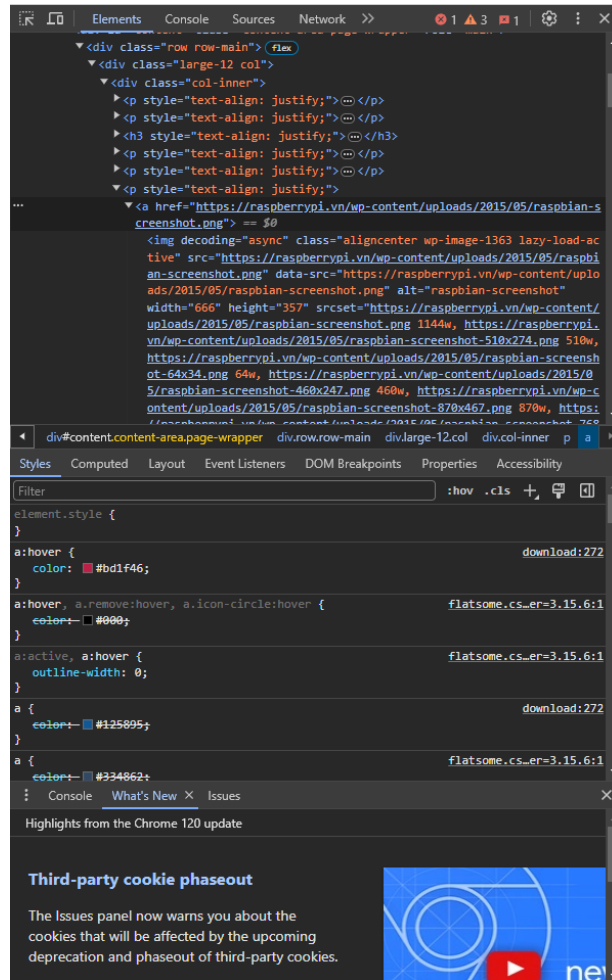
*#Lưu ý:* Nếu nhấn nút tại đây để tải mà không tải xuống được thì thực hiện theo các bước sau.

- Bước 1: Nhấn chuột phải chọn Inspect để mở cửa sổ console hoặc nhấn nút F12.



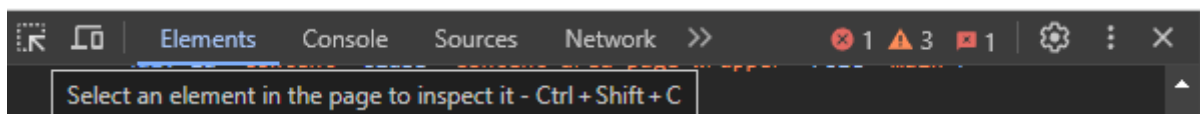
*Hình 7: Chuột phải chọn Inspect*





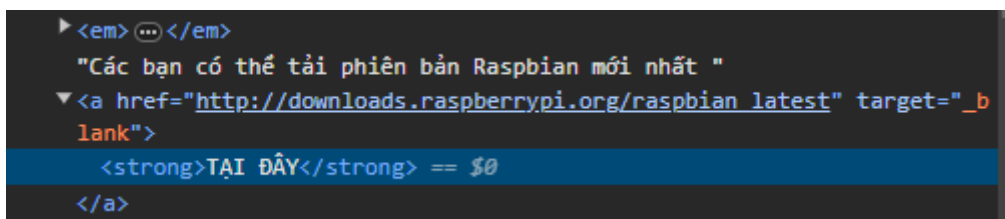
Hình 8: Cửa sổ Console

- Bước 2: Chọn hình mũi tên Scale để chọn phần tử trong trang đang hiển thị của nó (Ctrl + Shift + C)



Hình 9: Chọn biểu tượng ở ngoài cùng bên trái

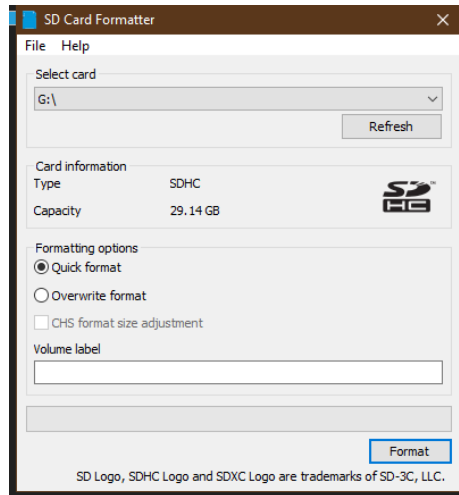
+ Sau đó di chuyển con chuột vào biểu tượng tại đây lúc ban đầu ta sẽ thấy được link tải file hệ điều hành.



Hình 10: Link tải ở trong phần href="..."

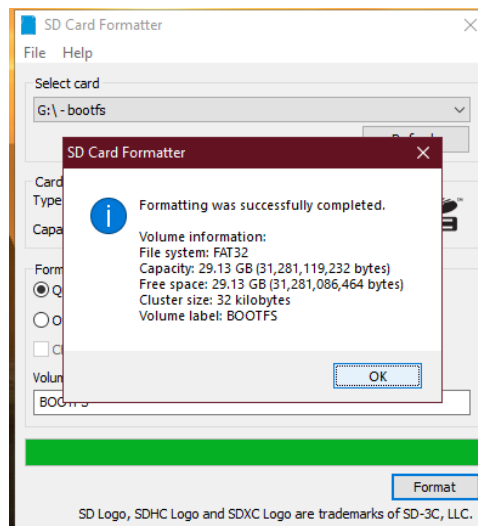
+ Ta click vào link tải và file sẽ tự động tải xuống.

- Bước 3: Sau khi đã tải file hệ điều hành xong thì ta tiếp tục cắm thẻ nhớ vào máy tính để thực hiện công việc Format bằng phần mềm Formatter. Sau đây giao diện SD Card Formatter.

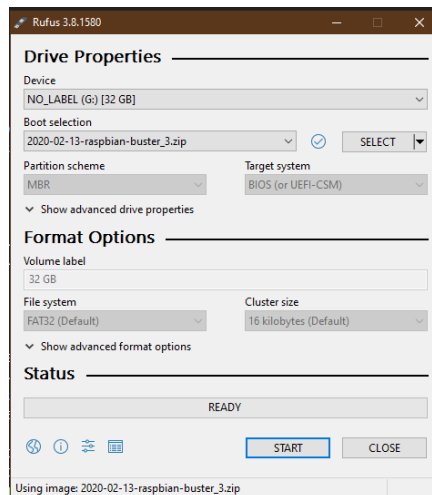


*Hình 11: Giao diện SD Card Formatter*

+ Để format ta cắm thẻ nhớ vào máy tính rồi mới mở phần mềm SD Card Formatter thì hệ thống sẽ tự động nhận diện thẻ nhớ và chọn sẵn cho ta. Ta chọn nút Format để Format thẻ.

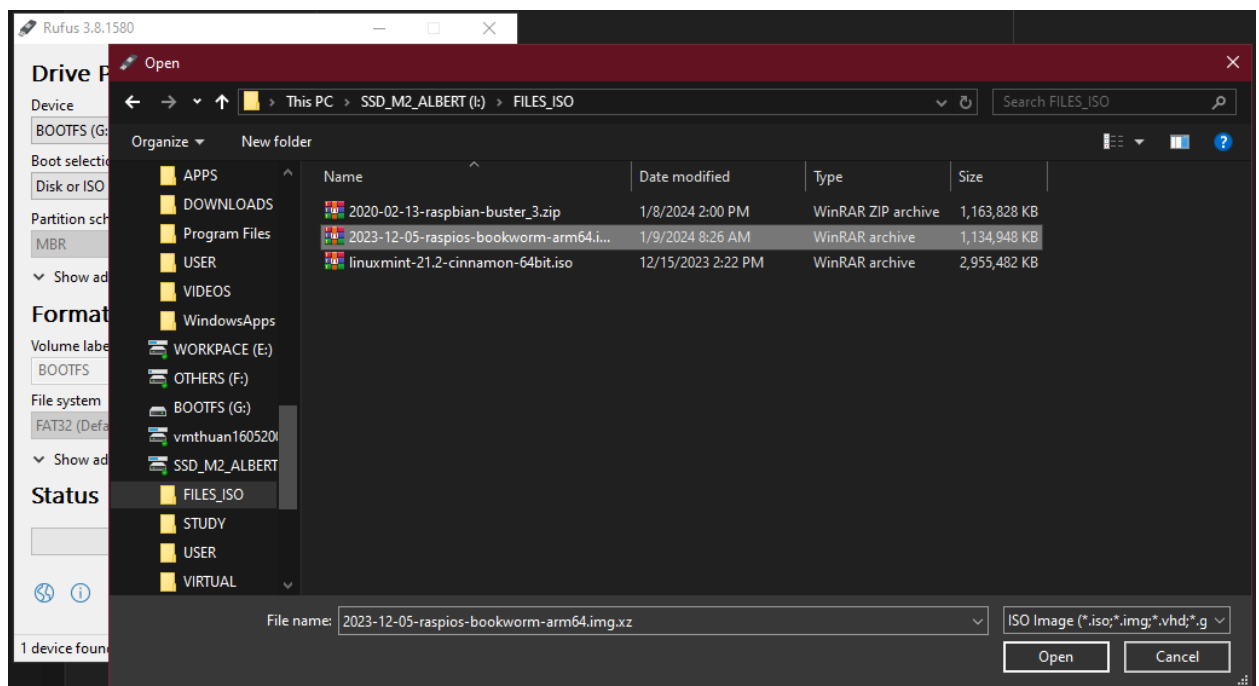


+ Sau khi Format thành công thì ta đóng phần mềm lại và mở tiếp phần mềm Rufus để giải nén hệ điều hành. Sau đây là giao diện Rufus.

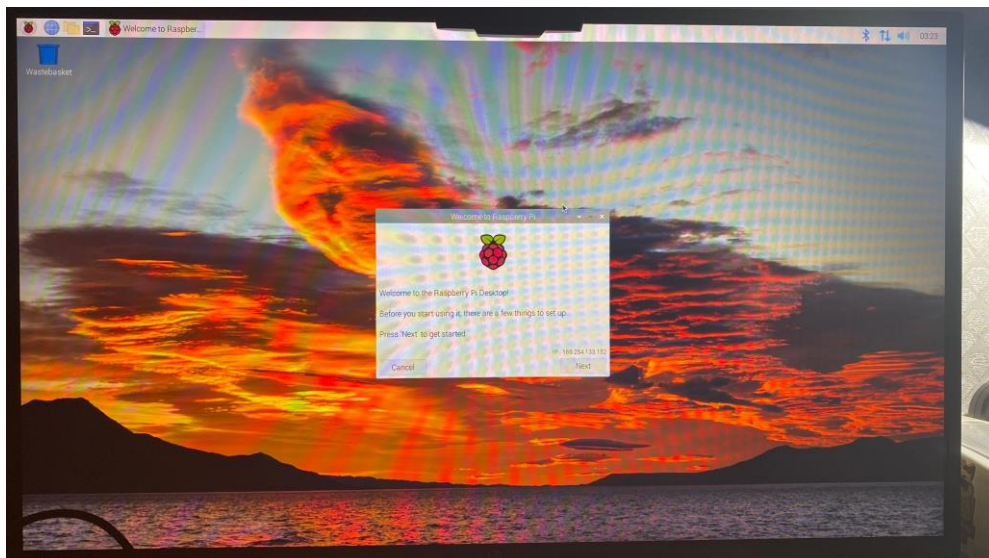


Hình 12: Giao diện Rufus

+ Khi mở phần mềm Rufus thì hệ thống cũng tự động chọn thẻ nhớ cho ta, ta chỉ cần nhấn nút SELECT để chọn file hệ điều hành mà ta mới tải xuống máy. Sau đó, ta nhấn nút START để bắt đầu quá trình ghi file hệ điều hành vào thẻ nhớ. Và chờ cho đến khi kết thúc quá trình ghi file vào thẻ nhớ.

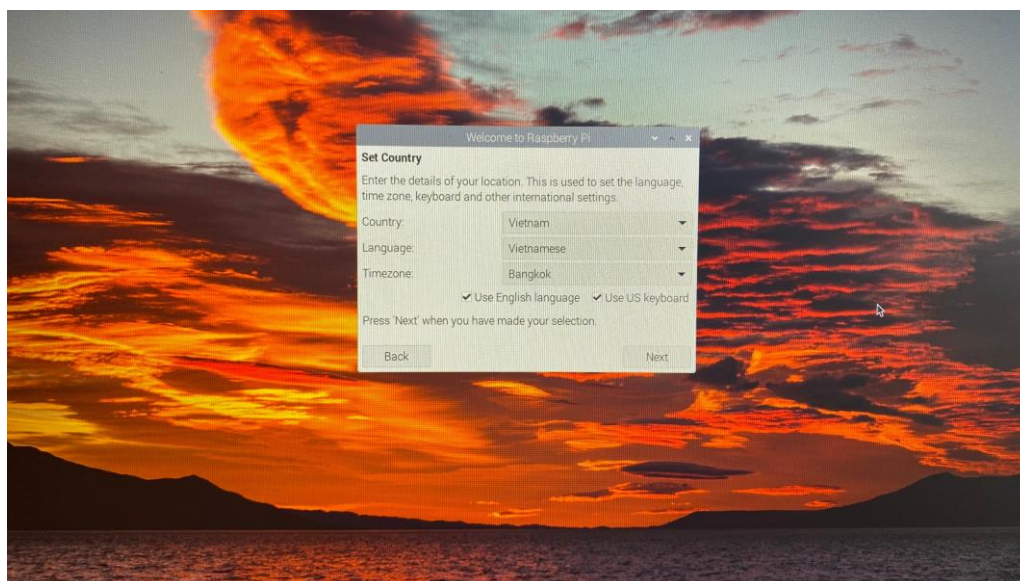


- Bước 4: Sau khi báo thành công ở phần mềm Rufus thì ta tắt phần mềm và rút thẻ nhớ cắm vào Raspberry để tiếp tục cài đặt hệ điều hành.



*Hình 13: Hình ban đầu khi mở máy*

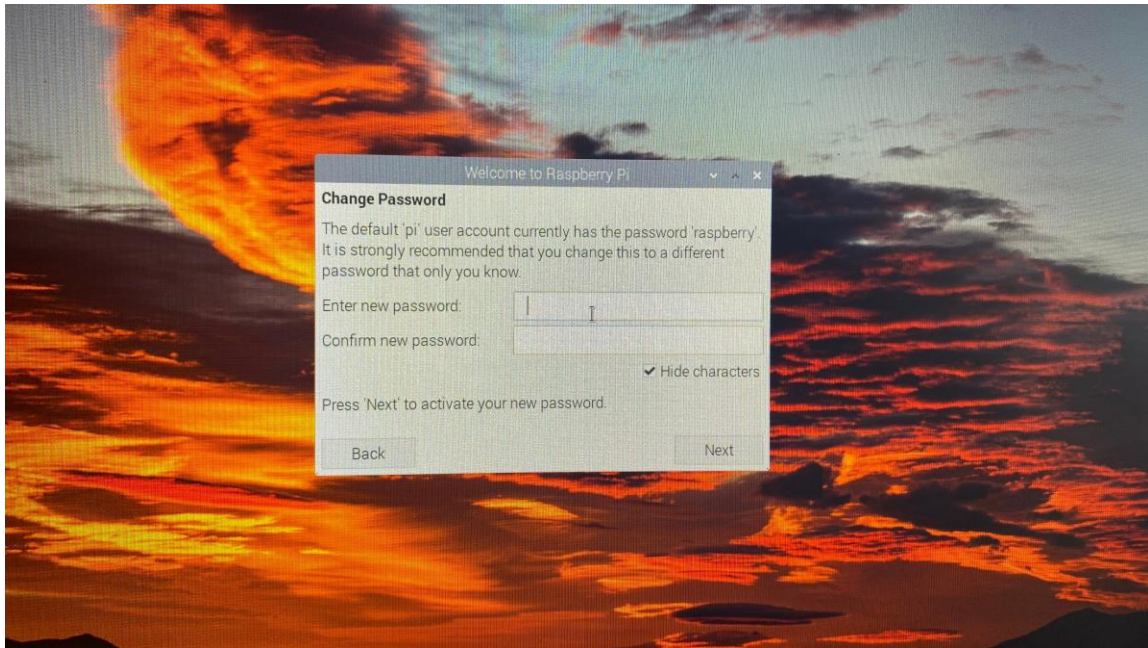
+ Chọn nút Next để thực hiện tiếp tục cài đặt hệ điều hành. Sau khi nhấn Next ta cửa sổ cài đặt thời gian sẽ mở ra như hình sau.



*Hình 14: Cửa sổ cài đặt thời gian*

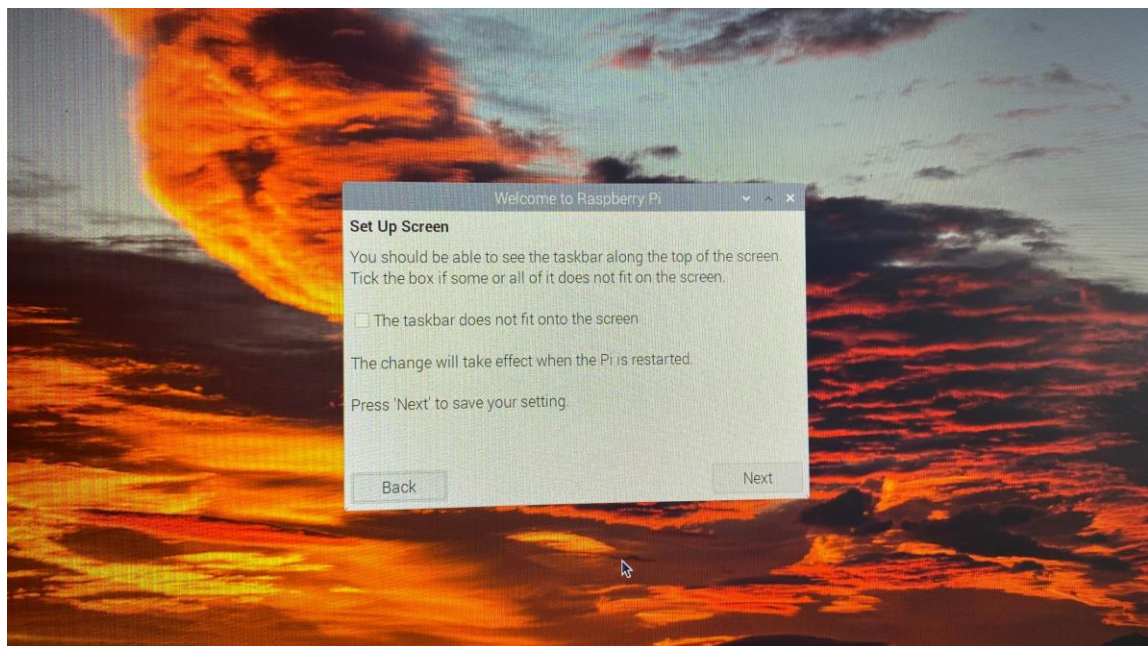
+ Ta chọn quốc gia là Việt Nam chọn tích Use English language và Use US Keyboard nếu muốn. Sau đó ta tiếp tục nhấn Next để cài đặt mật khẩu bảo mật như hình sau.





*Hình 15: Cửa sổ bảo mật mật khẩu*

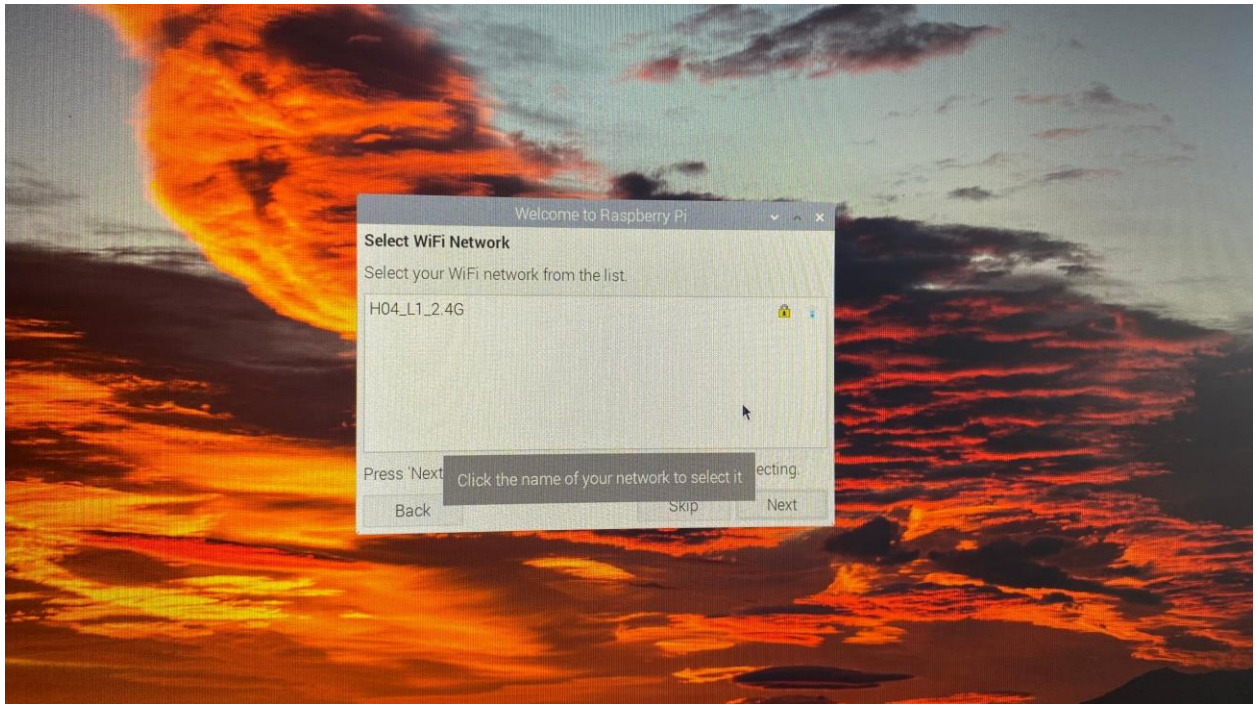
+ Ta nhập mật khẩu của mình rồi nhấn Next để mở cửa sổ cài đặt màn hình như hình sau.



*Hình 16: Cài đặt màn hình*

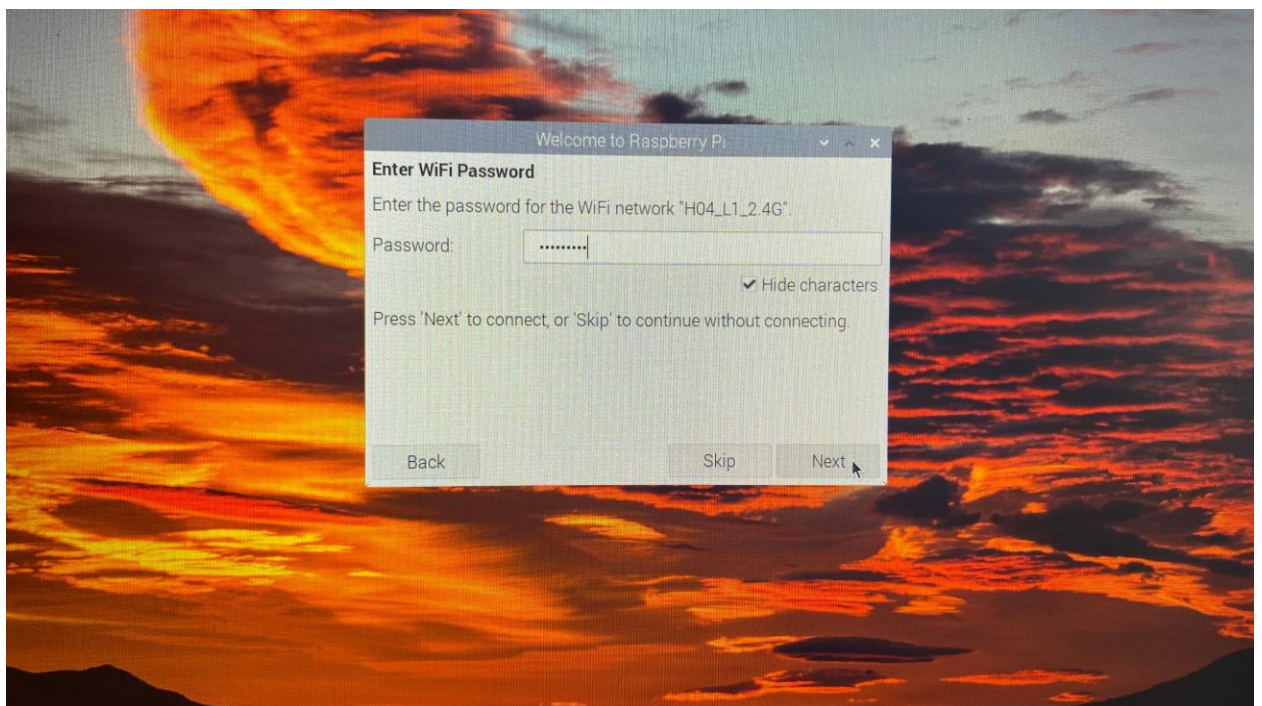
+ Chọn tích nếu muốn để thanh taskbar không cố định ở trên màn hình hiển thị. Sau đó ta nhấn nút Next để tiếp tục cài đặt.





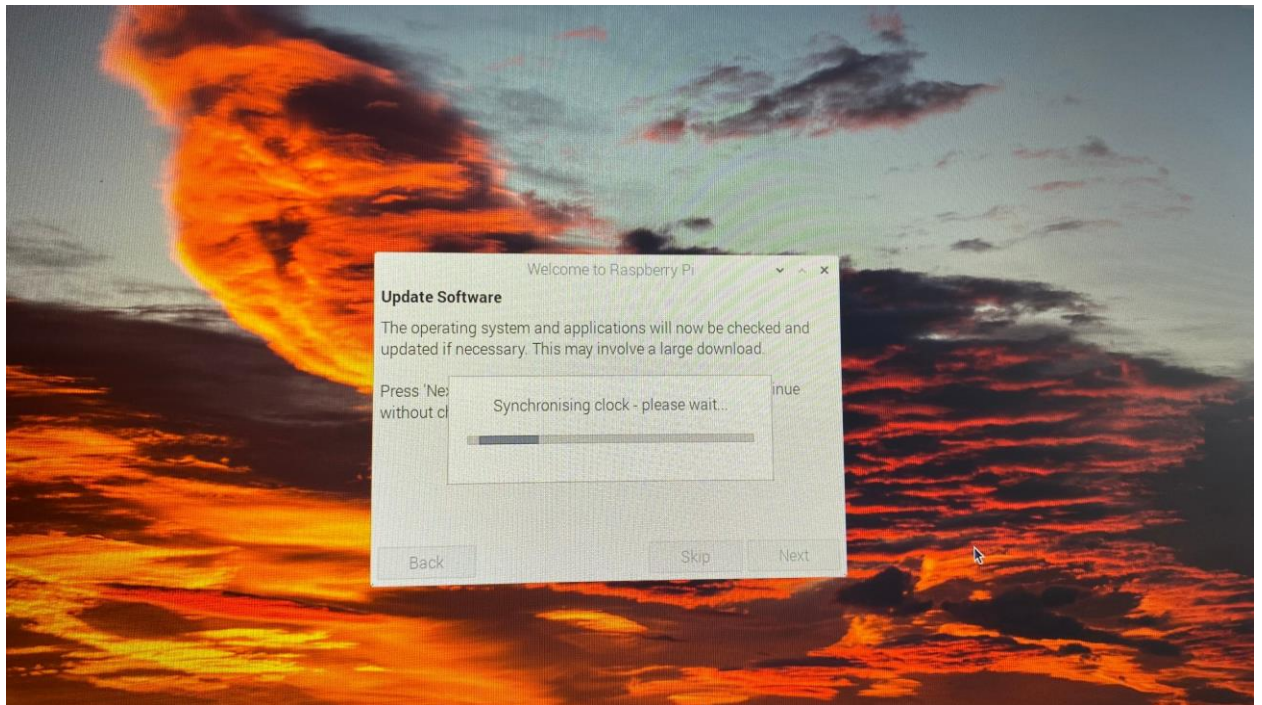
*Hình 17: Cài đặt mạng*

+ Tại đây ta chọn mạng để kết nối. Ví dụ mình chọn mạng H04\_L1\_2.4G thì mình nhấn vào mạng H04\_L1\_2.4G và chọn nút Next để nhập mật khẩu kết nối và nếu không muốn kết nối mạng ta sẽ nhấn nút Skip để bỏ qua cài đặt mạng.



*Hình 18: Nhập mật khẩu mạng kết nối*

+ Ta sẽ nhập mật khẩu của mạng mà ta muốn kết nối và sau đó nhấn nút Next để tiếp tục cài đặt.



*Hình 19: Cửa sổ cập nhật phần mềm*

+ Ta có thể muốn cập nhật lên phiên bản mới hơn không, nếu không ta nhấn Skip để bỏ qua việc cập nhật. Nếu muốn cập nhật ta sẽ nhấn Next để cho hệ thống cập nhật phiên bản mới nhất.

+ Cuối cùng cửa kết thúc cài đặt hiện lên và ta nhấn nút Finish để kết thúc quá trình cài đặt hệ điều hành cho Raspberry

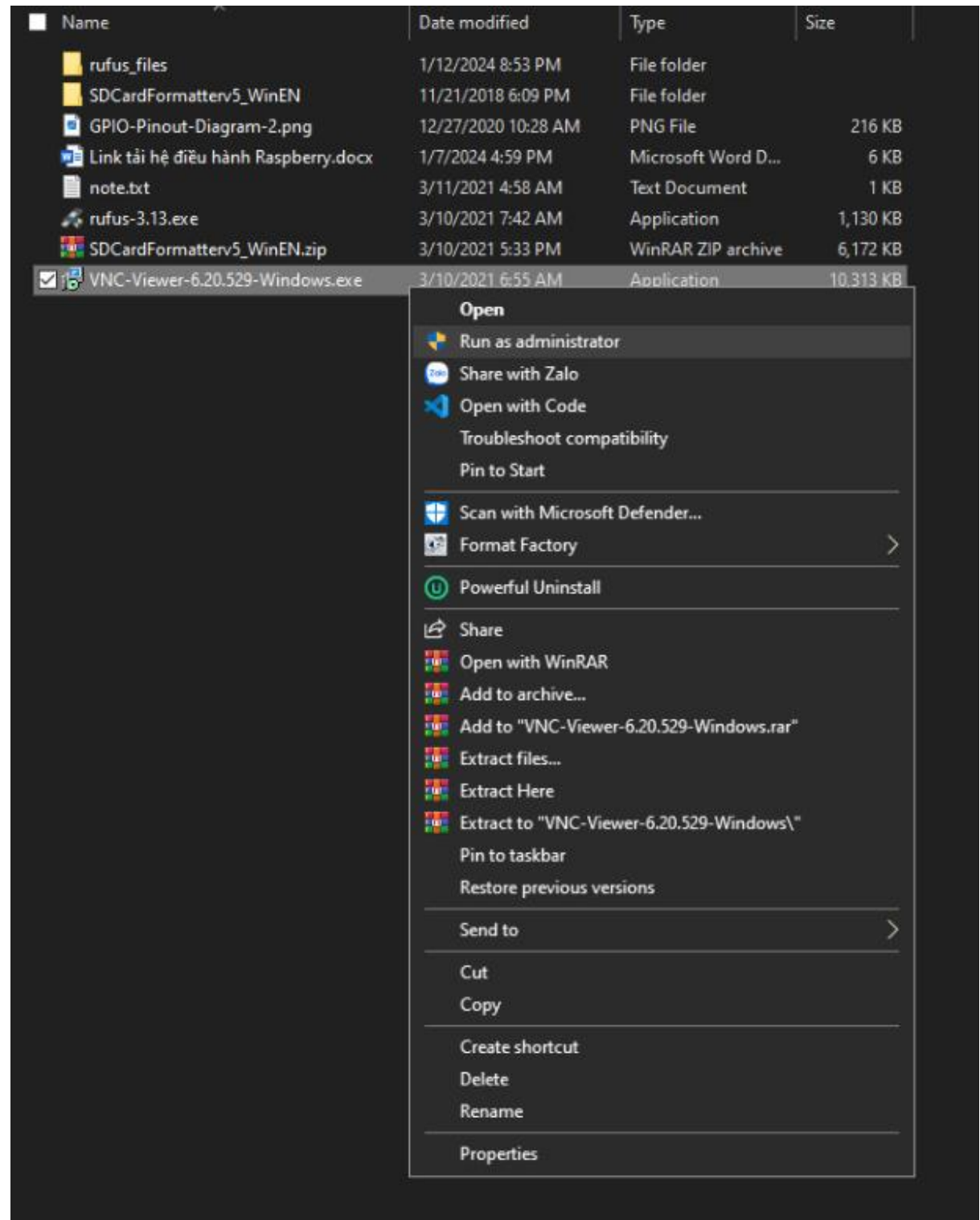
### **3. Trình bày các bước kết nối cáp để hiển thị lên màn hình**

- Bước 1: Xác định các chân của con Raspberry
- Bước 2: Cáp micro USB để cấp nguồn cho Raspberry Pi
- Bước 3: Kết nối cáp HDMI/miniHDMI để kết nối raspberry với màn hình rời.
- Bước 4: Kết nối chuột để điều khiển
- Bước 5: Sau đó bật nguồn tất cả mọi thứ.



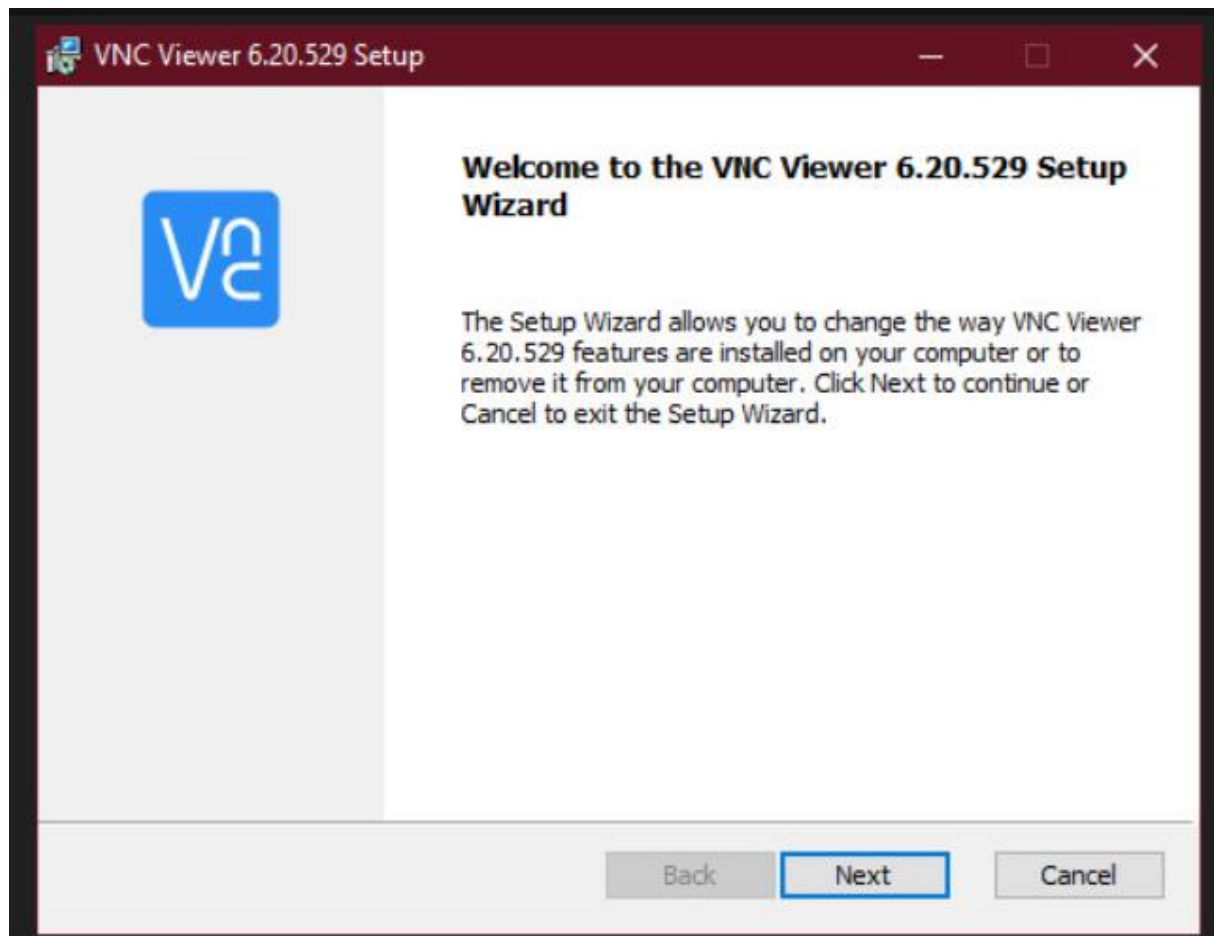
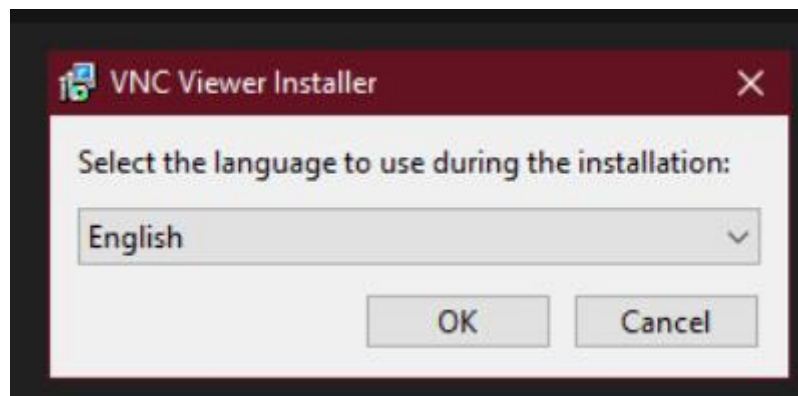
#### 4. Trình bày các bước cài đặt để hiển thị lên màn hình thông qua địa chỉ IP.

##### \*Các bước cài đặt VNC

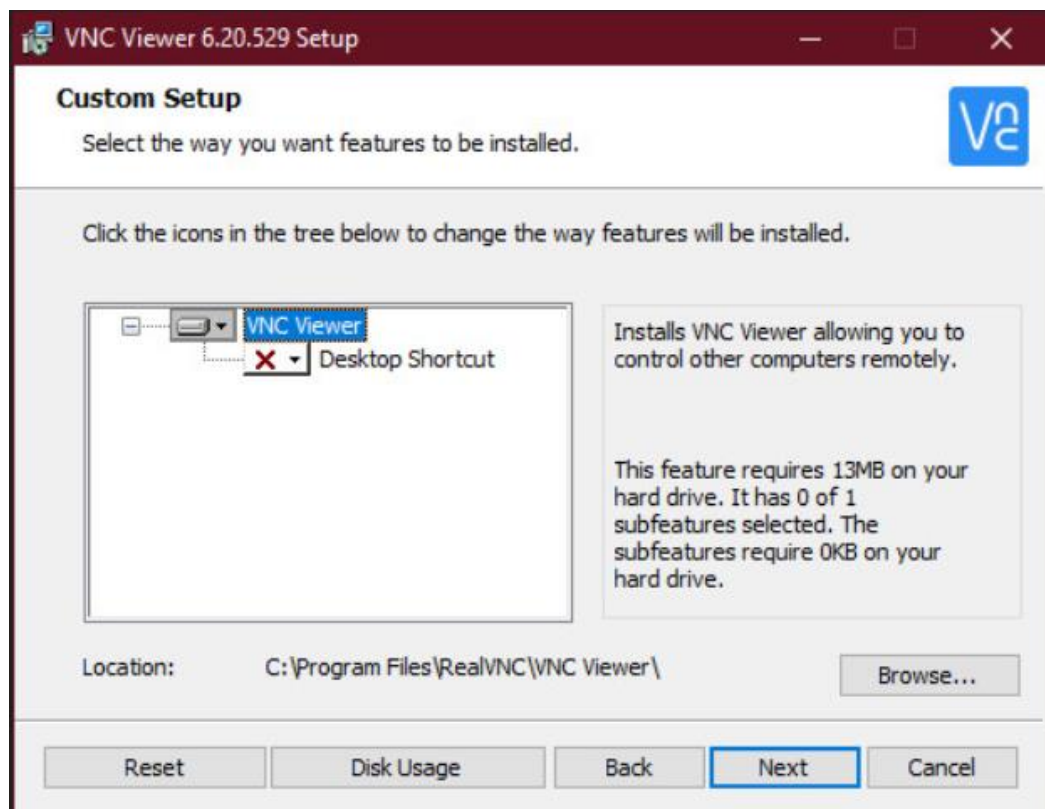
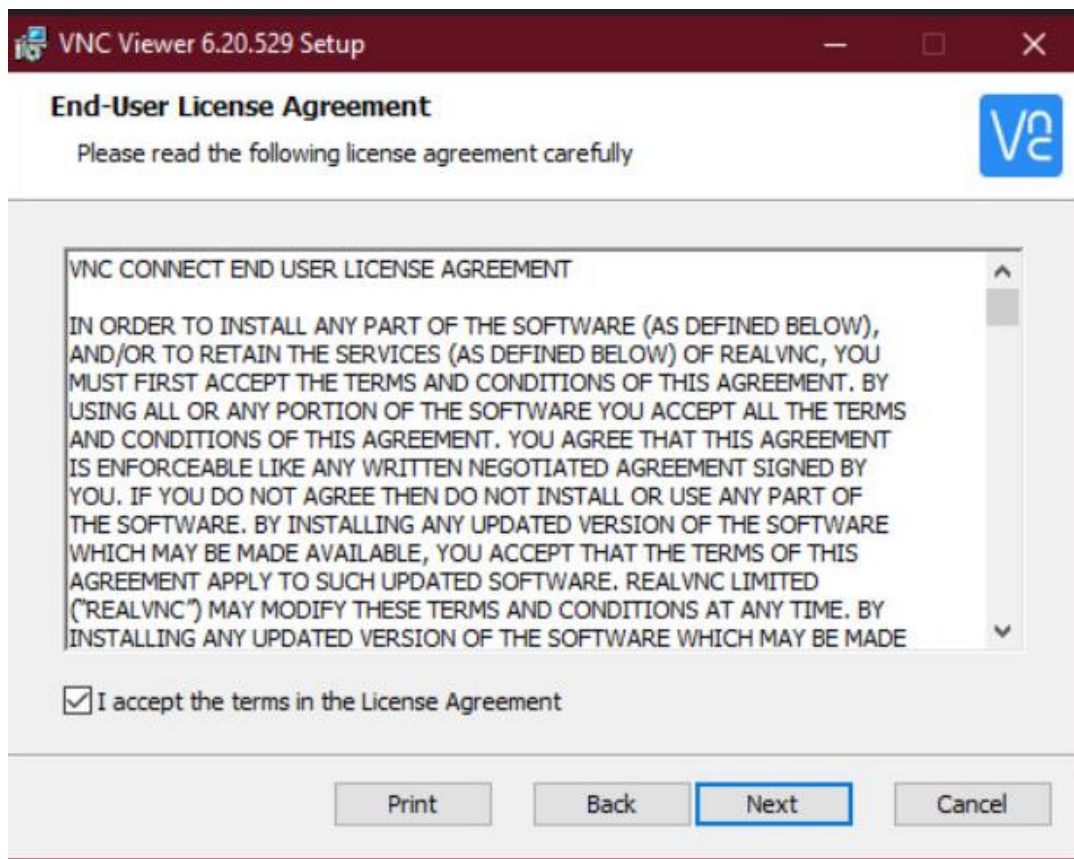


Chọn ngôn ngữ sử dụng và Click chọn Next để tiếp tục

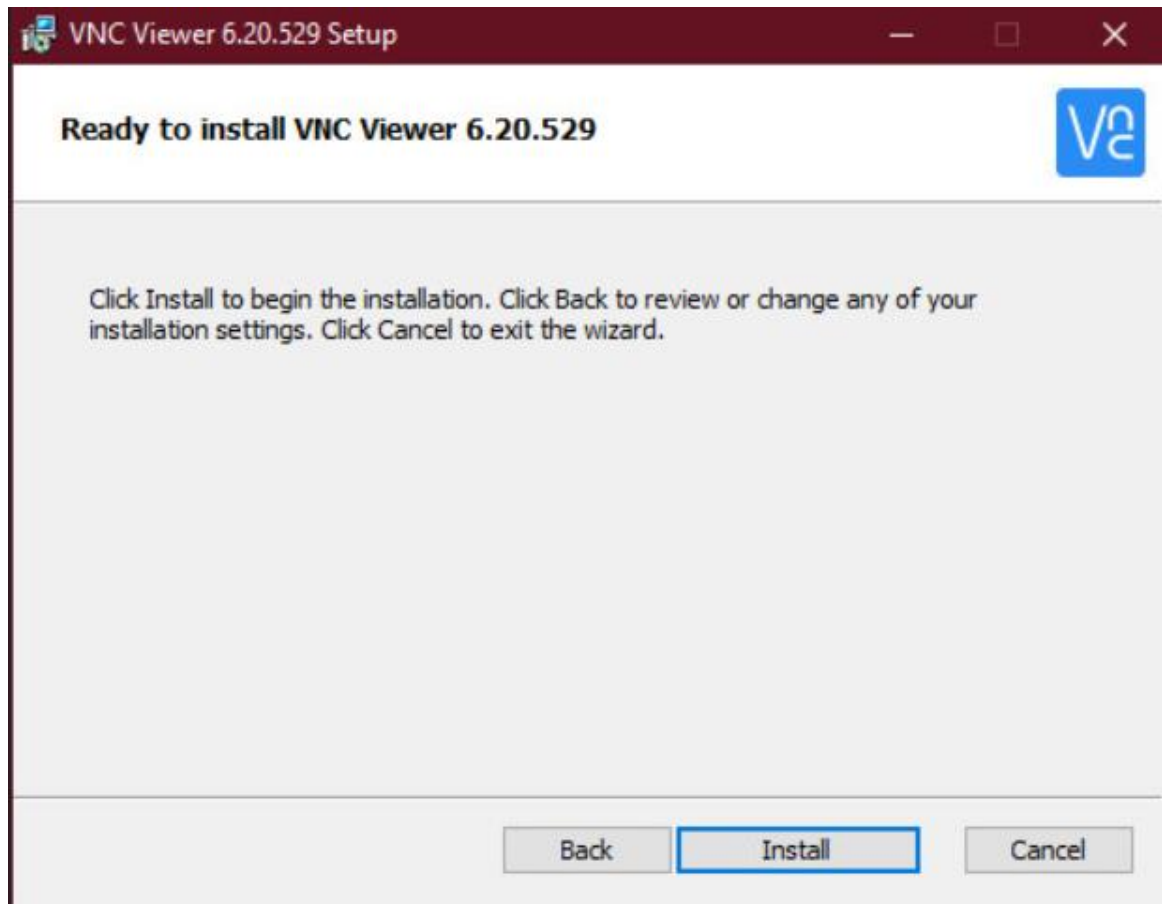




Click chọn *I accept the terms in the license agreement* sau đó chọn **Next**

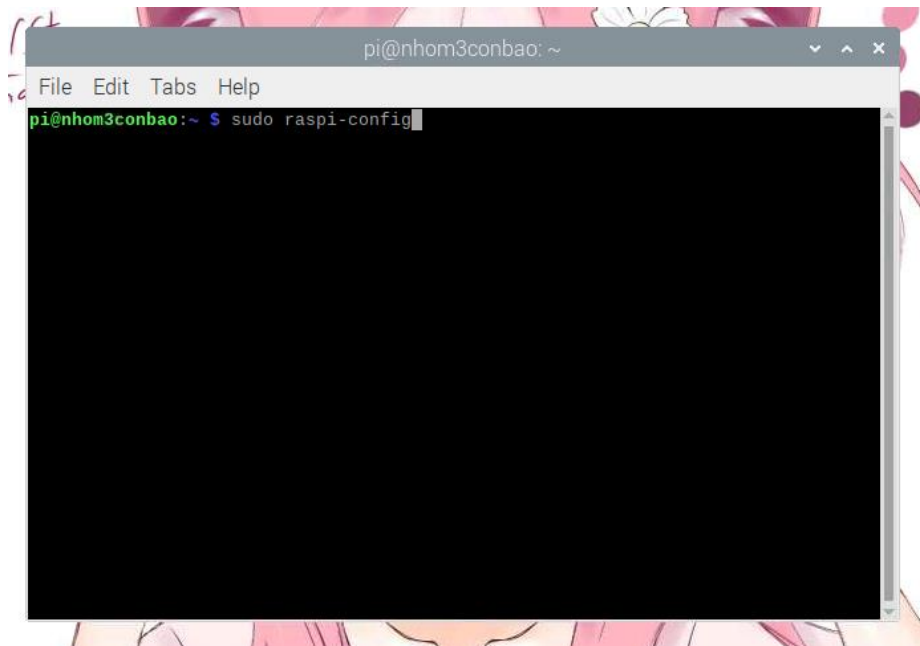


Click chọn **Next** và **Install** để hoàn tất

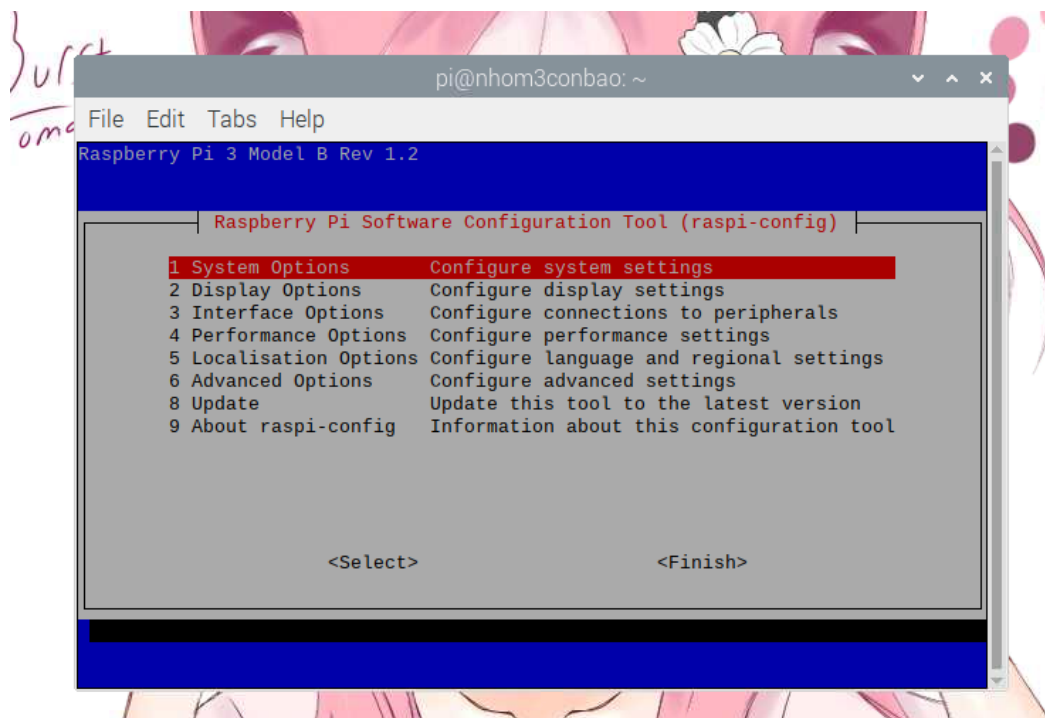


**\*Các bước thiết lập để máy tính lấy địa chỉ IP để kết nối**

- **Bước 1:** Ta phải kết nối mạng wifi cho raspberry và trùng với mạng mà máy tính ta kết nối.
- **Bước 2:** Vào Terminal của Raspberry và nhập lệnh “sudo raspi-config” để vào cài đặt cấu hình cho Raspberry.



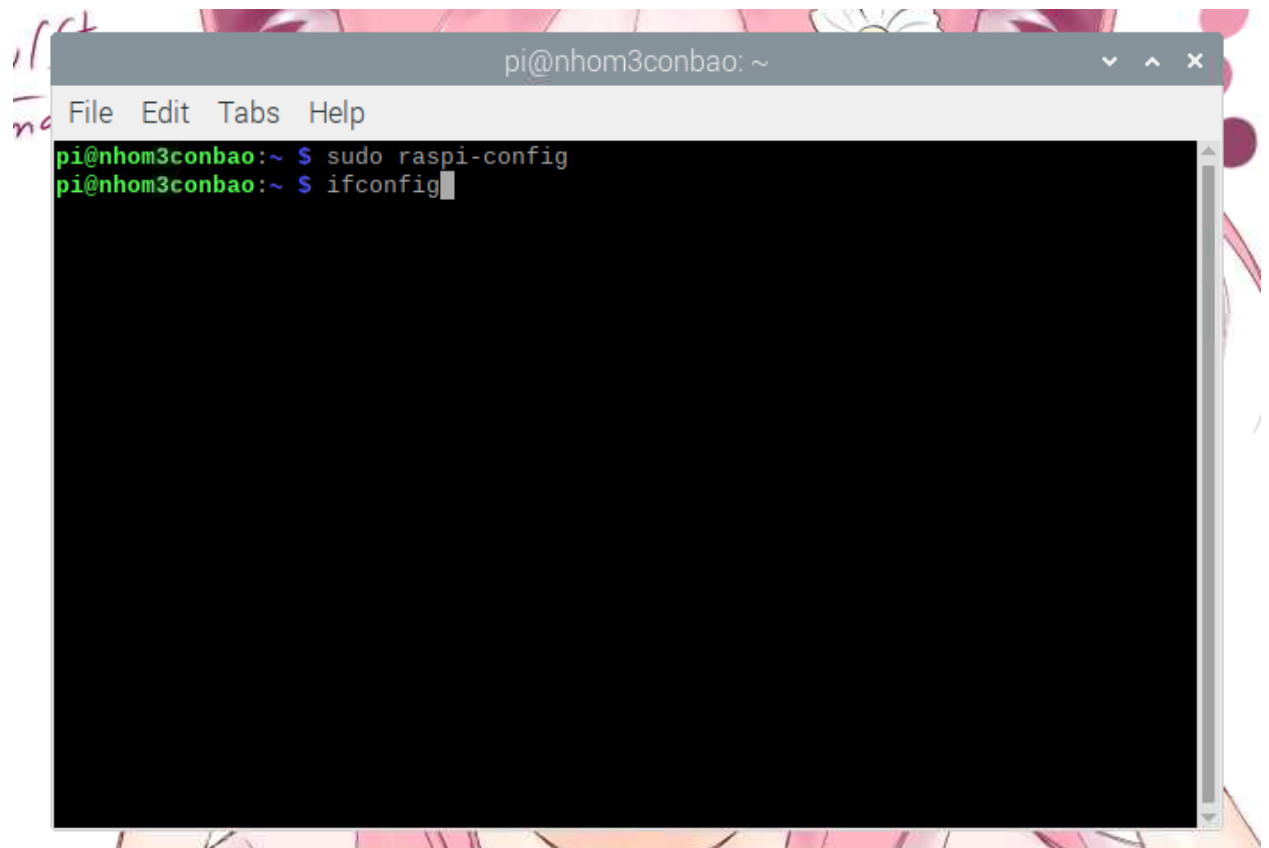
Hình 20: Nhập lệnh vào Terminal



Hình 21: Cửa sổ cấu hình cho Raspberry

+ Ta sử dụng nút lên xuống để di chuyển và nút enter để chọn nút Tab để chọn Select hoặc Finish. Ta thực hiện theo đường dẫn là Interface Options -> VNC -> Yes -> Ok.

- **Bước 3:** Nhập tiếp lệnh “ifconfig” terminal để hiển thị địa chỉ Ip của mình.



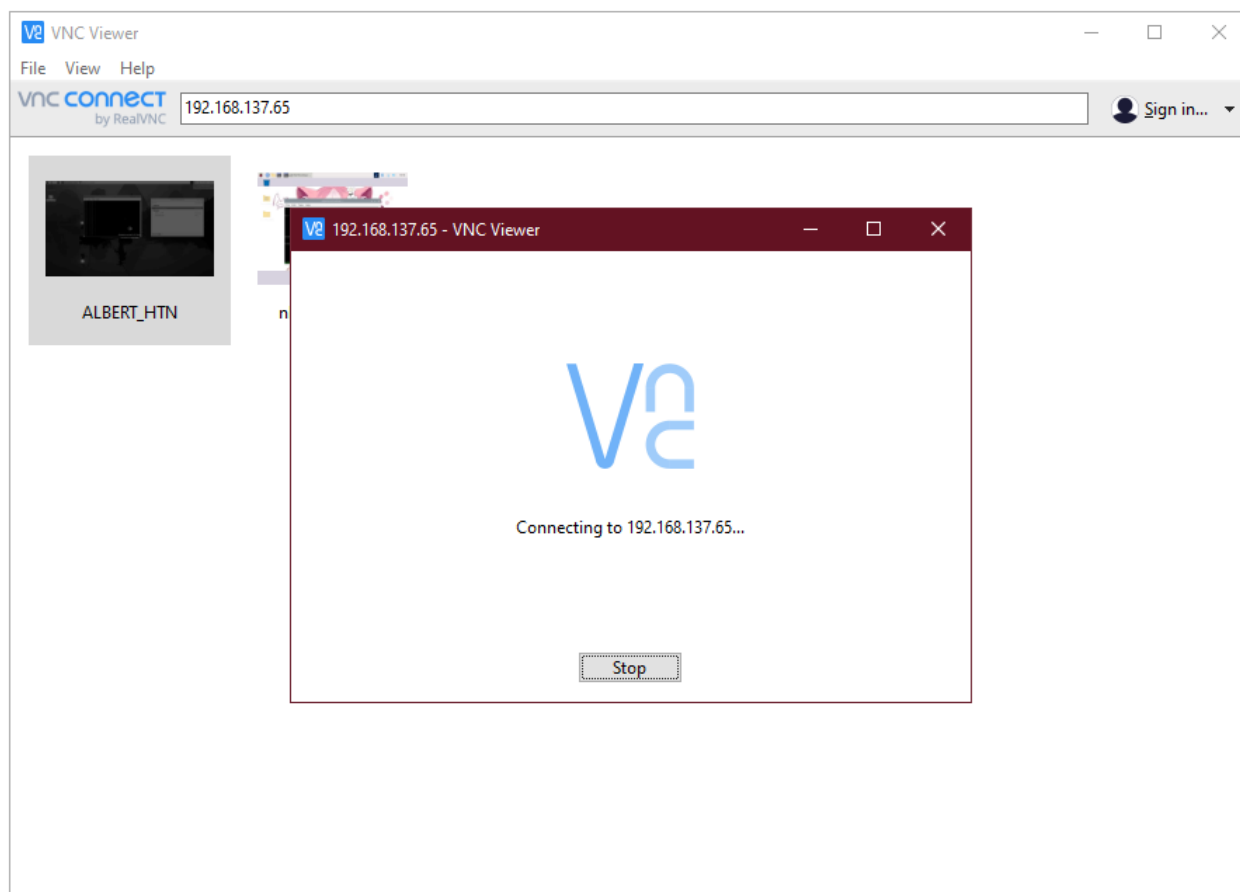
Hình 22: Nhập lệnh vào terminal

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.137.65 netmask 255.255.255.0 broadcast 192.168.137.255
    inet6 2402:800:6370:2a58:f79a:4278:57b8:253 prefixlen 64 scopeid 0x0<g
    local>
```

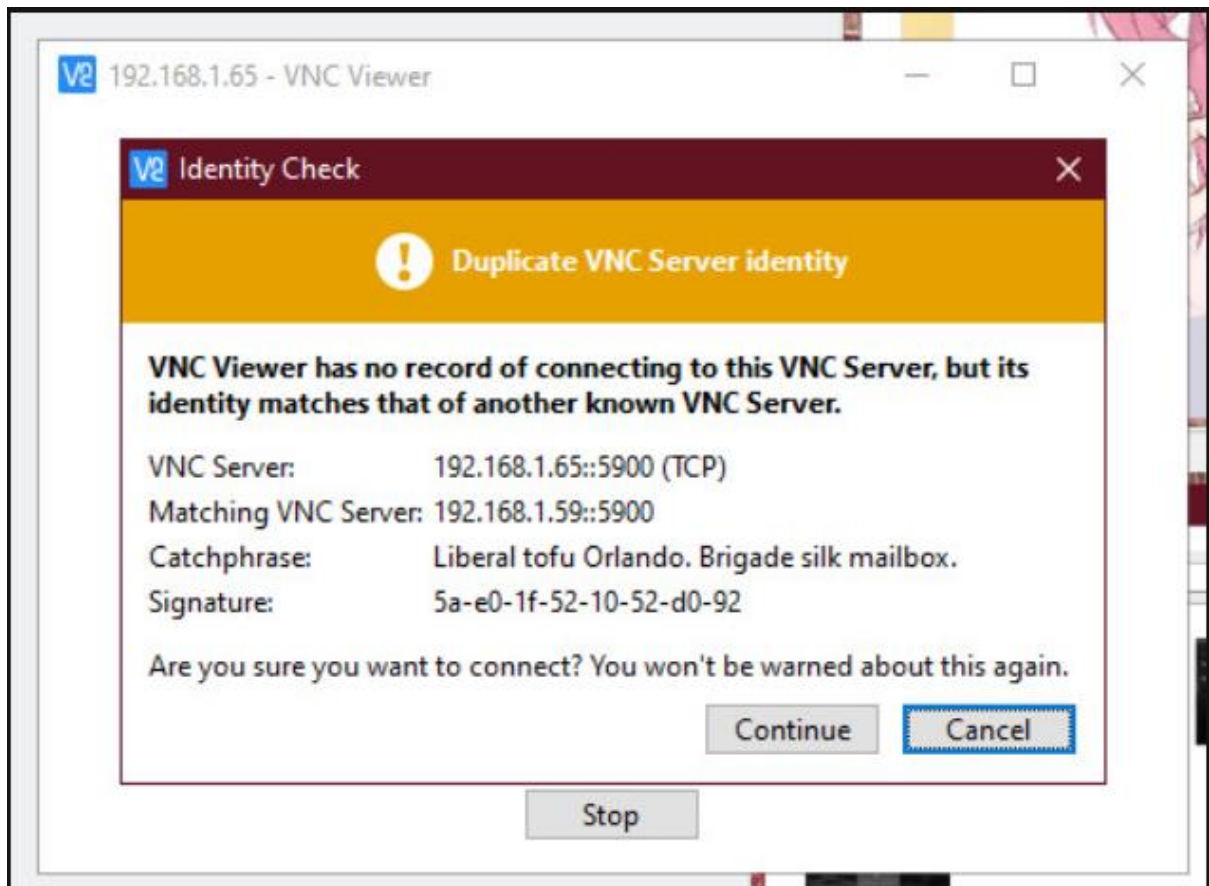
Hình 23: Địa chỉ inet

+ Ta xem địa chỉ inet 192.168.137.65 là địa chỉ ip4 của raspberry để máy tính có thể truy cập vào qua phần mềm VNC Viewer.

- **Bước 4:** Ta ở phần mềm VNC Viewer và nhập địa chỉ inet và nhấn Enter.



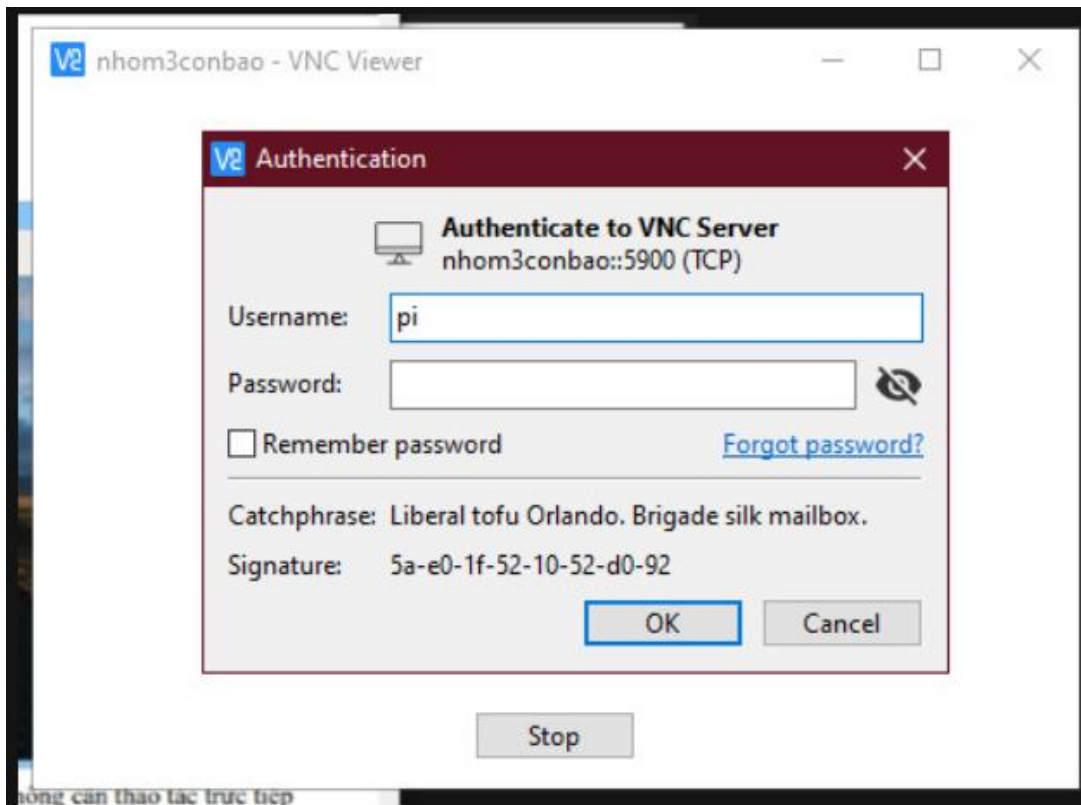
*Hình 24: Cửa sổ VNC Viewer*



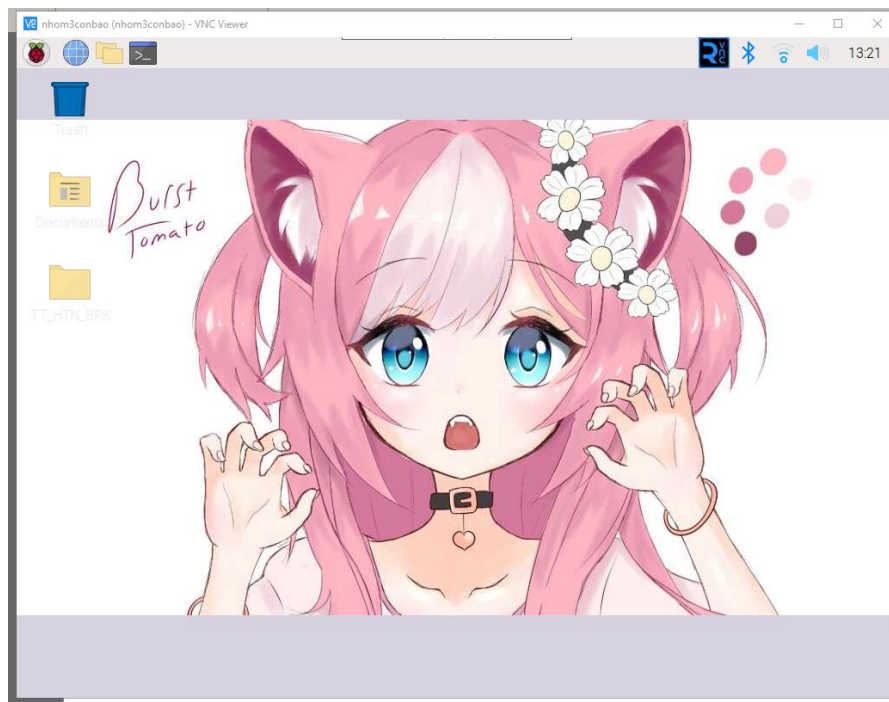
+Nhấn continue, khi thành công ta nhấn Username là pi (mặt định của Raspberry) và Password là password mà ta đặt lúc cài hệ điều hành nếu ta không cài mật khẩu lúc cài hệ điều hành thì mật khẩu mặt định là raspberry. Sau đó nhấn Connect để truy cập vào màn hình của Raspberry.

+ Sau khi hoàn tất cài đặt, vào phần mềm VNC và nhập ID ở trên vào và tiến hành nhập tên đăng nhập và mật khẩu ( username là Pi và mật khẩu là 162003)





Nhấn OK để tiếp tục



Hình 25: Cửa sổ làm việc của máy tính khi truy cập bằng địa chỉ IP

Kết nối xong ta được giao diện Raspbian nhìn từ máy tính

Như vậy, bạn đã có thể thao tác cho Raspbian trên PC mà không cần thao tác trực tiếp trên màn hình Raspberry.

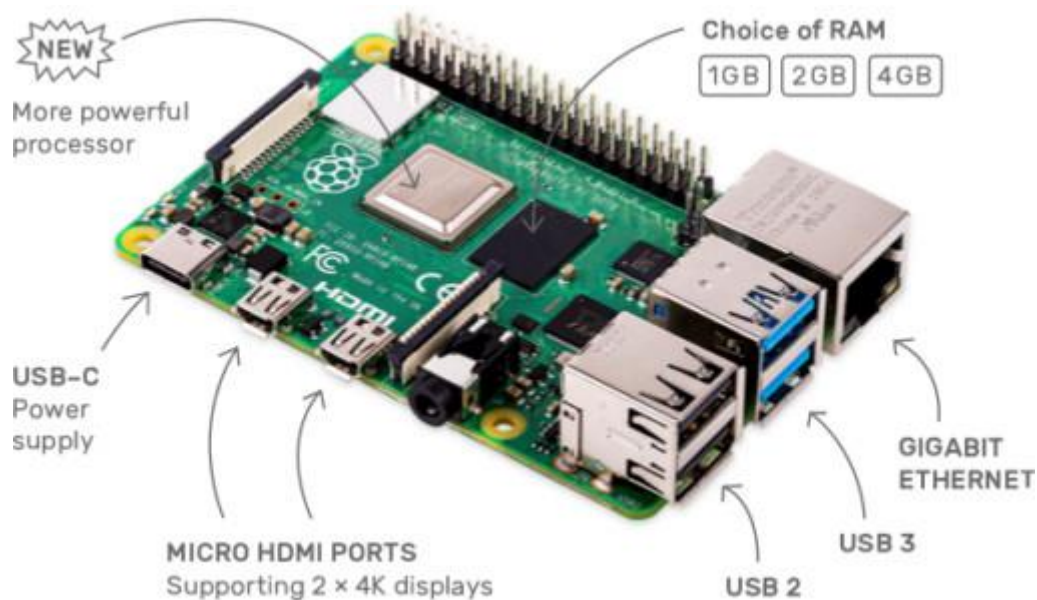


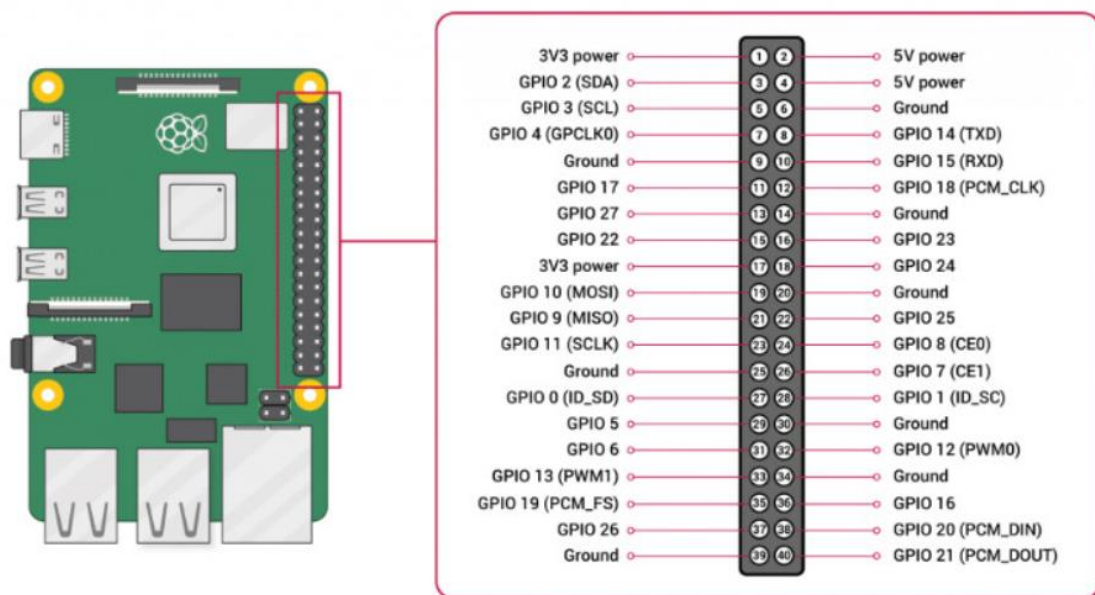
## 5. Trình bày các bước viết chương trình, cấu hình chân (giải thích) để điều khiển led đơn.

### \*Cấu hình chân

Trong 40 chân GPIO bao gồm:

- 26 chân GPIO. Khi thiết lập là input, GPIO có thể được sử dụng như chân interrupt, GPIO 14 & 15 được thiết lập sẵn là chân input.
- 1 UART, 1 I2C, 2 SPI, 1 PWM (GPIO 4)
- 2 chân nguồn 5V, 2 chân nguồn 3.3V, 8 chân GND
- 2 chân ID EEPROM Vi xử lý ARMv7 32bit quad core 900Mhz, dung lượng Ram 1G, và bộ nhớ kiểu micro SD dung lượng tùy chọn ( nên  $\geq 4G$ ). Khi một chân GPIO lên mức cao sẽ đạt điện áp 3.3V, dòng ra tối đa  $I_{max} = 5mA$ .





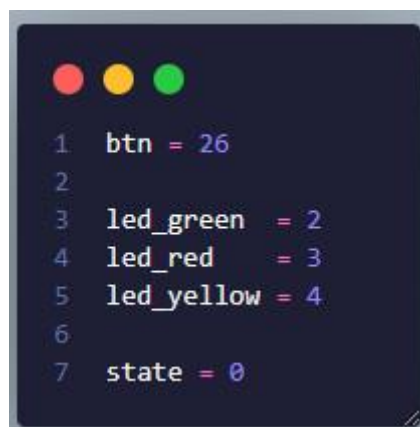
### \*Các bước viết chương trình

#### - Bước 1: Xem đề trên FHQX và xác định việc cần làm

Đề: Điều khiển LED bằng button:

- + Khi button được nhấn thì trạng thái trên GPIO-2 sẽ xuống mức thấp.
- + Lưu ý rằng, khi set chân GPIO làm input thì trạng thái của GPIO sẽ lơ lửng (float) lúc cao, lúc thấp không xác định. Để xác định mức rõ ràng cần dùng điện trở để kéo GPIO lên mức cao hoặc kéo xuống thấp. Ở đây, sử dụng điện trở trong của GPIO để kéo lên mức cao ( $R=10k\Omega$ ).

#### - Bước 2: Lựa chọn các chân để kết nối và sử dụng



Hình 26 Chọn chân sử dụng

- **Bước 3:** Khai báo các chân được sử dụng theo định dạng BCM (sử dụng BCM thay cho BOARD vì khi sử dụng BCM thì ta dùng chân nào thì nó chính là số sau ký tự GPIO, vd như GPIO13 thì ta sẽ khai báo sử dụng chân 13). Lệnh `pull_up_down = GPIO.PUD_DOWN` là sử dụng trở nội và chế độ DOWN khi nhấn thì giá trị của nút nhấn là 1 và ngược lại, khai báo các chân dùng cho led là chân out, xuất ngõ ra ban đầu là các led tắt

```
1 def setup():
2
3     state = 0
4
5     GPIO.setwarnings(False)
6     GPIO.setmode(GPIO.BCM)
7     GPIO.setup(btn, GPIO.IN, pull_up_down = GPIO.PUD_DOW
8 N)
9     GPIO.setup(led_green,GPIO.OUT)
10    GPIO.setup(led_red,GPIO.OUT)
11    GPIO.setup(led_yellow,GPIO.OUT)
12
13    GPIO.output(led_green, 0)
14    GPIO.output(led_red, 0)
15    GPIO.output(led_yellow, 0)
```

Hình 27 Thiết lập các chân tín hiệu

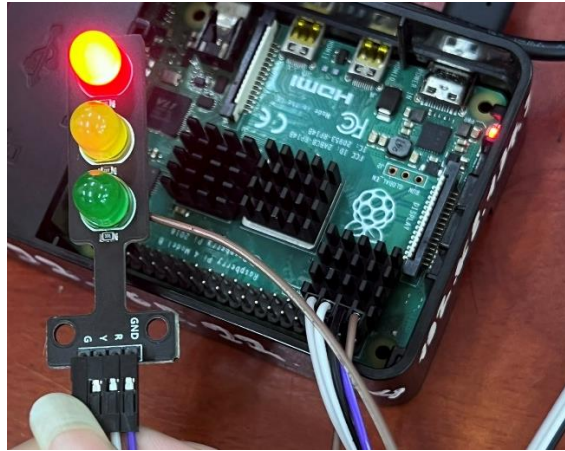
- **Bước 4:** Vào chương trình chính, lúc này biến state được gán bằng giá trị của nút nhấn. Xuất tín hiệu ngõ ra led là trạng thái của biến state và in ra terminal giá trị state là “state = ”

```
1 def main():
2     setup()
3     while True:
4         state = GPIO.input(btn)
5         GPIO.output(led_green,state)
6         GPIO.output(led_red,state)
7         GPIO.output(led_yellow,state)
8         print("state: ",state)
```

Hình 28 Xuất các tín hiệu ngõ ra cho các led dựa vào trạng thái state

- **Bước 5:** Gắn linh kiện lên Raspberry

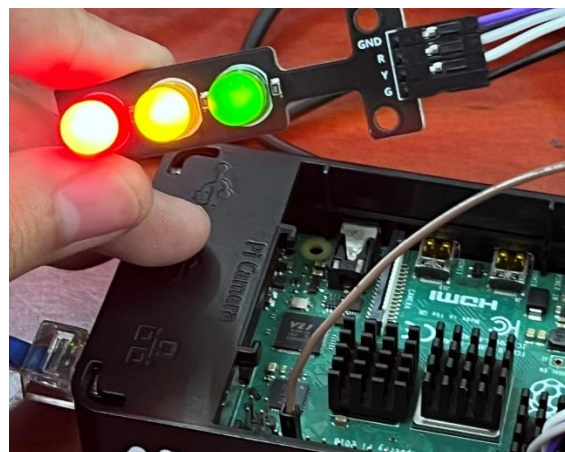
+ Sử dụng 4 chân GPIO Là 2, 3, 4, 26 ( led là chân GPIO 2, 3, 4, btn là chân GPIO 26) và 1 chân Ground



- **Bước 6:** cho chạy chương trình, quan sát led đã hoạt động đúng hay chưa



*Hình 29 Bấm đầu các led tắt*



*Hình 30 Khi nút nhấn tích cực mức cao thì 3 đèn sáng*

## **6. Điều khiển led đơn qua giao diện trên máy tính**

- **Bước 1:** Truy cập vào FHQX để xem đề bài và xác định mục tiêu cần làm

Đề:

1. Tạo nút on và off để điều khiển on/off.
2. Thêm nút inv để đảo trạng thái của led.
3. Thực hiện các bài khác với giao diện nút nhấn và led.

- **Bước 2:** thiết kế giao diện

+ from tkinter import \*: Dòng này import tất cả các thành phần từ thư viện Tkinter, một thư viện đồ họa được sử dụng để tạo giao diện người dùng đồ họa (GUI) trong Python.

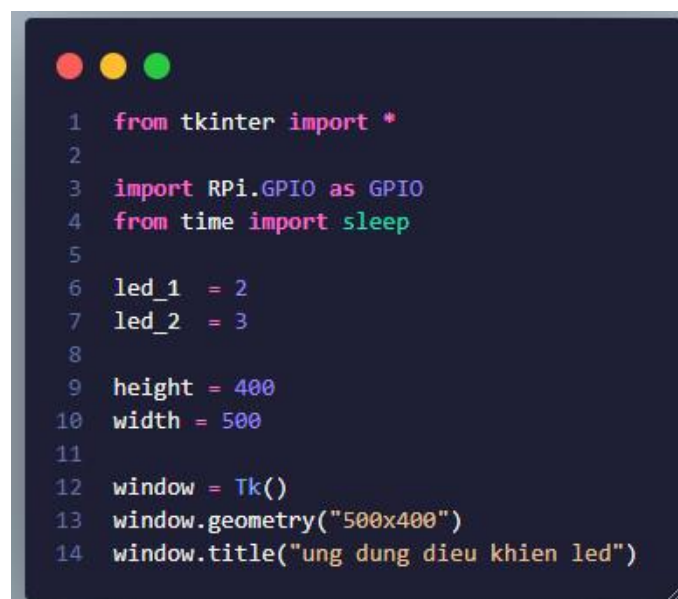
+ import RPi.GPIO as GPIO: Dòng này import thư viện RPi.GPIO và đặt tên viết tắt là GPIO. Thư viện này cung cấp các chức năng để làm việc với GPIO trên Raspberry Pi.

+ from time import sleep: Dòng này chỉ import hàm sleep từ thư viện time. Hàm sleep được sử dụng để tạm dừng thực thi chương trình trong một khoảng thời gian được chỉ định.

+ Chọn chân GPIO2 và GPIO3 để dùng cho led

+ window = Tk() được sử dụng để tạo một đối tượng cửa sổ (window) trong thư viện Tkinter của Python

+ Thiết kế màn hình có kích thước là 500x400 với tiêu đề trên màn hình là “ung dung dieu khien led”

A screenshot of a code editor with a dark background and light-colored text. The code is a Python script for a Raspberry Pi GUI. It starts with imports for tkinter, RPi.GPIO, and time. Then it defines pin numbers for two LEDs (led\_1 = 2, led\_2 = 3) and the window dimensions (height = 400, width = 500). Finally, it creates a Tk window, sets its geometry to 500x400, and sets its title to "ung dung dieu khien led".

```
1  from tkinter import *
2
3  import RPi.GPIO as GPIO
4  from time import sleep
5
6  led_1 = 2
7  led_2 = 3
8
9  height = 400
10 width = 500
11
12 window = Tk()
13 window.geometry("500x400")
14 window.title("ung dung dieu khien led")
```



*Hình 31 Thiết kế giao diện điều khiển*

- **Bước 3:** Khởi tạo các chân ngõ ra led, khai báo theo định dạng BCM, thiết lập các chân led là ngõ ra, xuất giá trị led ban đầu là tắt, in ra terminal “\_start\_” và nghỉ trong 1s

A screenshot of a terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The terminal displays a Python code snippet for setting up GPIO pins. The code is as follows:

```
1 def setup():
2     GPIO.setwarnings(False)
3     GPIO.setmode(GPIO.BCM)
4     GPIO.setup(led_1,GPIO.OUT)
5     GPIO.setup(led_2,GPIO.OUT)
6     GPIO.output(led_1, 0)
7     GPIO.output(led_2, 0)
8     print("_start_")
9     sleep(1)
```

*Hình 32 Thiết lập các chân tín hiệu*

- **Bước 4:** Điều khiển các led 1 và led 2 với hai nút nhấn on và off

+ control\_light(pin, state): Hàm này được sử dụng để điều khiển trạng thái của đèn LED được kết nối với chân GPIO được chỉ định (pin). Tham số state được sử dụng để xác định trạng thái của đèn (0 là tắt, 1 là bật).

+ control\_light\_1\_off(): Hàm này tắt đèn LED số 1 và cập nhật thông tin trạng thái đèn 1 lên giao diện người dùng.

+ control\_light\_1\_on(): Hàm này bật đèn LED số 1 và cập nhật thông tin trạng thái đèn 1 lên giao diện người dùng.

+ control\_light\_2\_off(): Hàm này tắt đèn LED số 2 và cập nhật thông tin trạng thái đèn 2 lên giao diện người dùng.

+ control\_light\_2\_on(): Hàm này bật đèn LED số 2 và cập nhật thông tin trạng thái đèn 2 lên giao diện người dùng.



```

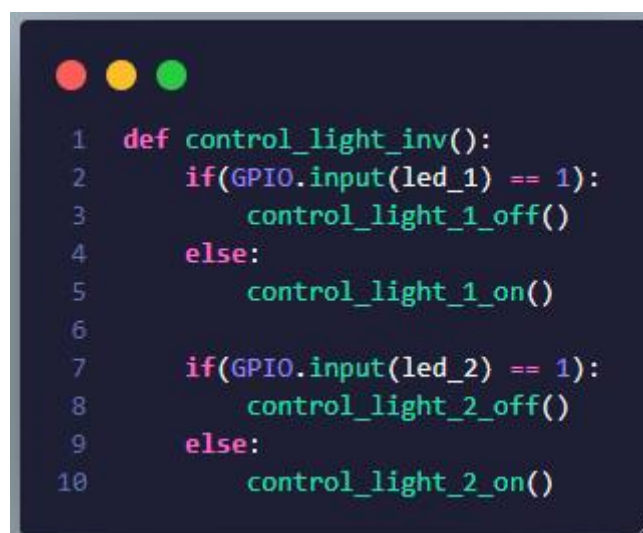
1  def control_light(pin, state):
2      GPIO.output(pin, state)
3
4  def control_light_1_off():
5      control_light(led_1, 0)
6      text_led_1 = Label(window, text = "State Led 1: " + str(GPIO.input(led_1)))
7      text_led_1.place(x=50,y=10)
8
9  def control_light_1_on():
10     control_light(led_1, 1)
11     text_led_1 = Label(window, text = "State Led 1: " + str(GPIO.input(led_1)))
12     text_led_1.place(x=50,y=10)
13
14 def control_light_2_off():
15     control_light(led_2, 0)
16     text_led_2 = Label(window, text = "State Led 2: " + str(GPIO.input(led_2)))
17     text_led_2.place(x=250,y=10)
18
19 def control_light_2_on():
20     control_light(led_2, 1)
21     text_led_2 = Label(window, text = "State Led 2: " + str(GPIO.input(led_2)))
22     text_led_2.place(x=250,y=10)

```

*Hình 33 Điều khiển led 1 và led 2*

- **Bước 5:** Điều khiển các led 1 và led 2 với nút nhấn INV đảo chiều trạng thái

+ control\_light\_inv(): Hàm này đảo trạng thái của cả hai đèn LED. Nếu led 1 đang bật, nó sẽ tắt và led 2 sẽ sáng và ngược lại, nếu cả 2 led đều đang sáng thì khi nhấn inv cả 2 đèn sẽ tắt và ngược lại. Sau đó, nó cập nhật thông tin trạng thái của cả hai đèn lên giao diện người dùng.



```

1  def control_light_inv():
2      if(GPIO.input(led_1) == 1):
3          control_light_1_off()
4      else:
5          control_light_1_on()
6
7      if(GPIO.input(led_2) == 1):
8          control_light_2_off()
9      else:
10         control_light_2_on()

```

*Hình 34 Điều khiển 2 led với nút đảo trạng thái INV*

**- Bước 6: Thiết kế giao diện cho các nhãn**

+ def style\_info(): Khai báo một hàm có tên là style\_info. Hàm này sẽ chứa mã để tạo các nhãn và thiết lập kiểu dáng cho chúng.

+ label\_1 = Label(window, fg="red",text = "Bài tập: Ứng dụng điều khiển led có đảo chiều"): Tạo một nhãn (Label) có tên label\_1. Nhãn này được đặt trong cửa sổ window và có màu chữ là đỏ (fg="red") với nội dung là "Bài tập: Ứng dụng điều khiển led có đảo chiều".

+ label\_1.place(x=50,y=200): Đặt nhãn label\_1 tại vị trí (50, 200) trên cửa sổ.

+ label\_1.config(font=("Courier", 11)): Thiết lập kiểu dáng cho label\_1. Trong trường hợp này, kiểu chữ được chọn là "Courier" với kích thước là 11.

+ label\_2 = Label(window, text = "Tên nhóm: 3 con báo"): Tạo một nhãn label\_2 khác với nội dung là "Tên nhóm: 3 con báo".

+ label\_2.place(x=50,y=250): Đặt nhãn label\_2 tại vị trí (50, 250) trên cửa sổ.

+ label\_2.config(font=("Courier", 14)): Thiết lập kiểu dáng cho label\_2. Kiểu chữ được chọn là "Courier" với kích thước là 14.

+ label\_3 = Label(window, text = "Thành viên 1: Võ Minh Thuận - MSSV: 21161366"): Tạo một nhãn label\_3 khác với thông tin về thành viên 1.

+ label\_3.place(x=50,y=300): Đặt nhãn label\_3 tại vị trí (50, 300) trên cửa sổ.

+ label\_4 = Label(window, text = "Thành viên 2: Lê Quang Thương - MSSV: 21161363"): Tạo một nhãn label\_4 khác với thông tin về thành viên 2.

+ label\_4.place(x=50,y=320): Đặt nhãn label\_4 tại vị trí (50, 320) trên cửa sổ.

+ label\_5 = Label(window, text = "Thành viên 3: Trần Thị Xuân Hy - MSSV: 21161323"): Tạo một nhãn label\_5 khác với thông tin về thành viên 3.

+ label\_5.place(x=50,y=340): Đặt nhãn label\_5 tại vị trí (50, 340) trên cửa sổ.



```
1 def style_info():
2     # height = 200 - width = 500
3     label_1 = Label(window, fg="red", text = "Bài tập: Ứng dụng điều khiển led có đảo chiều")
4     label_1.place(x=50,y=200)
5     label_1.config(font =("Courier", 11))
6     label_2 = Label(window, text = "Tên nhóm: 3 con báo")
7     label_2.place(x=50,y=250)
8     label_2.config(font =("Courier", 14))
9     label_3 = Label(window, text = "Thành viên 1: Võ Minh Thuận - MSSV: 21161366")
10    label_3.place(x=50,y=300)
11    label_4 = Label(window, text = "Thành viên 2: Lê Quang Thương - MSSV: 21161363")
12    label_4.place(x=50,y=320)
13    label_5 = Label(window, text = "Thành viên 3: Trần Thị Xuân Hy - MSSV: 21161323")
14    label_5.place(x=50,y=340)
```

*Hình 35 Thiết kế giao diện cho các nhãn*

**- Bước 7: Tạo giao diện để điều khiển các led**

+ def style\_window(): Khai báo một hàm có tên là style\_window. Hàm này chứa mã để tạo giao diện điều khiển đèn LED và hiển thị trạng thái của chúng.

+ text\_led\_1 = Label(window, text="State Led 1: " + str(GPIO.input(led\_1))): Tạo một nhãn text\_led\_1 để hiển thị trạng thái của LED 1. Thông tin trạng thái được lấy từ chân led\_1 và được hiển thị trong nhãn.

+ text\_led\_1.place(x=50, y=10): Đặt nhãn text\_led\_1 tại vị trí (50, 10) trên cửa sổ.

+ B1 = Button(window, text="LED 1 ON ", width=20, bg="green", fg="black", command=control\_light\_1\_on): Tạo một nút (Button) B1 để bật đèn LED 1. Nút này có văn bản là "LED 1 ON", màu nền là màu xanh (bg="green"), màu chữ là đen (fg="black"), và có liên kết với hàm control\_light\_1\_on khi được nhấp.

+ B1.place(x=50, y=50): Đặt nút B1 tại vị trí (50, 50) trên cửa sổ.

+ B2 = Button(window, text="LED 1 OFF", width=20, bg="red", fg="black", command=control\_light\_1\_off): Tạo một nút B2 để tắt đèn LED 1. Nút này có văn bản là "LED 1 OFF", màu nền là màu đỏ (bg="red"), màu chữ là đen (fg="black"), và có liên kết với hàm control\_light\_1\_off khi được nhấp.

+ B2.place(x=250, y=50): Đặt nút B2 tại vị trí (250, 50) trên cửa sổ.

+ text\_led\_2 = Label(window, text="State Led 2: " + str(GPIO.input(led\_1))):  
Tạo một nhãn text\_led\_2 tương tự như text\_led\_1 nhưng để hiển thị trạng thái của LED 2.

+ text\_led\_2.place(x=250, y=10): Đặt nhãn text\_led\_2 tại vị trí (250, 10) trên cửa sổ.

+ B1 = Button(window, text="LED 2 ON ", width=20, bg="green", fg="black", command=control\_light\_2\_on): Tạo một nút B1 để bật đèn LED 2.

+ B1.place(x=50, y=100): Đặt nút B1 tại vị trí (50, 100) trên cửa sổ.

+ B2 = Button(window, text="LED 2 OFF", width=20, bg="red", fg="black", command=control\_light\_2\_off): Tạo một nút B2 để tắt đèn LED 2.

+ B2.place(x=250, y=100): Đặt nút B2 tại vị trí (250, 100) trên cửa sổ.

+ B3 = Button(window, text="INV", width=20, bg="yellow", fg="black", command=control\_light\_inv): Tạo một nút B3 để đảo trạng thái của cả hai đèn LED cùng một lúc.

+ B3.place(x=50, y=150): Đặt nút B3 tại vị trí (50, 150) trên cửa sổ.



```
1 def style_window():
2     # height = 200 - width = 500
3
4     text_led_1 = Label(window, text = "State Led 1: " + str(GPIO.input(led_1)))
5     text_led_1.place(x=50,y=10)
6     B1 = Button(window, text="LED 1 ON ", width=20, bg="green", fg="black", command=control_light_1_on)
7     B1.place(x=50,y=50)
8     B2 = Button(window, text="LED 1 OFF", width=20, bg="red", fg="black", command=control_light_1_off)
9     B2.place(x=250,y=50)
10
11     text_led_2 = Label(window, text = "State Led 2: " + str(GPIO.input(led_1)))
12     text_led_2.place(x=250,y=10)
13     B1 = Button(window, text="LED 2 ON ", width=20, bg="green", fg="black", command=control_light_2_on)
14     B1.place(x=50,y=100)
15     B2 = Button(window, text="LED 2 OFF", width=20, bg="red", fg="black", command=control_light_2_off)
16     B2.place(x=250,y=100)
17
18     B3 = Button(window, text="INV", width=20, bg="yellow", fg="black", command=control_light_inv)
19     B3.place(x=50,y=150)
```

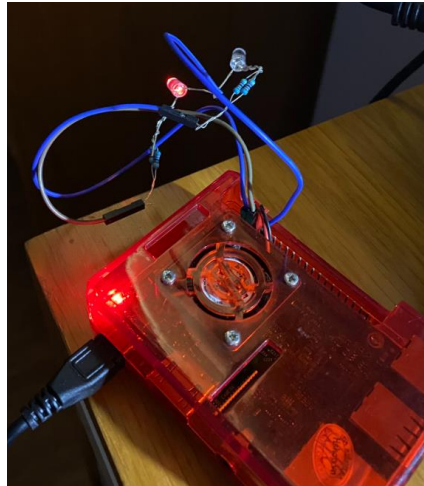
*Hình 36 Thiết lập giao diện cho các phím nhấn điều khiển các led*

- **Bước 8:** Vào chương trình chính, gọi các hàm vừa khởi tạo ở trên, khi gọi window.mainloop() chương trình sẽ dừng lại tại đây và bắt đầu xử lý các thao tác mà người dùng vừa sử dụng trên giao diện. Nếu không có dòng lệnh này, cửa sổ của ứng dụng sẽ hiển thị và biến mất ngay lập tức.

```
1  def main():
2      setup()
3      style_info()
4      style_window()
5      window.mainloop()
6
7  if __name__ == '__main__':
8      main()
```

*Hình 37 Chương trình chính*

- **Bước 9:** Kết nối linh kiện lên Raspberry

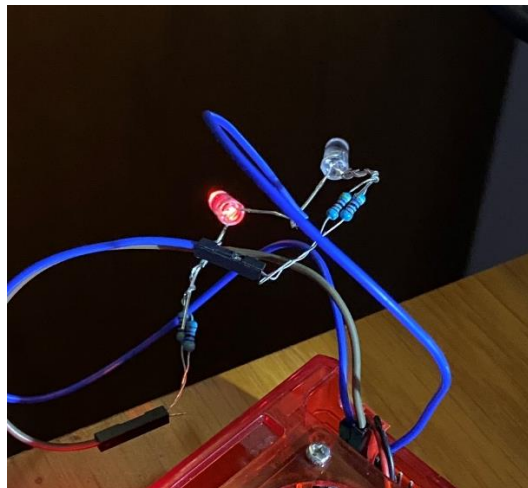


- **Bước 10:** Chạy chương trình, quan sát xem led đã hoạt động như yêu cầu hay chưa

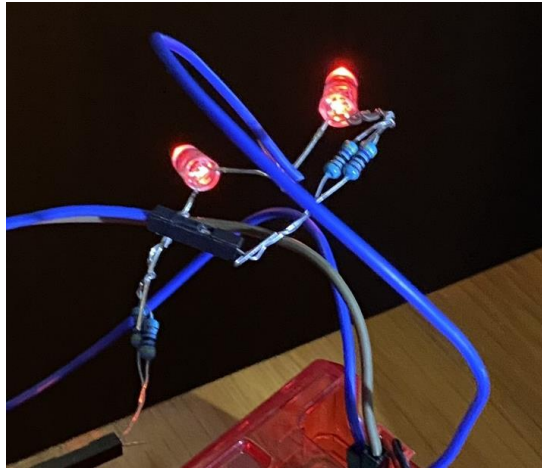


*Hình 38 Giao diện người dùng*

+ Khi nhấn nút nhấn LED 1 ON thì led 1 sẽ sáng, khi nhấn LED 1 OFF thì đèn 1 sẽ tắt, nếu trong lúc led 1 đang ở chế độ on mà ta nhấn thêm phím LED 2 ON thì cả 2 đèn sẽ sáng, nhấn LED 2 OFF thì led 2 sẽ tắt. Phím INV có chức năng đảo lại trạng thái hiện tại của cả 2 led.



*Hình 39 Led 1 sáng sau khi nhấn phím LED 1 ON*



*Hình 40 Cả 2 led đều sáng sau khi nhấn và phím LED 1 ON và LED 2 ON*