MIT AI2 204

IoT with MIT App Inventor

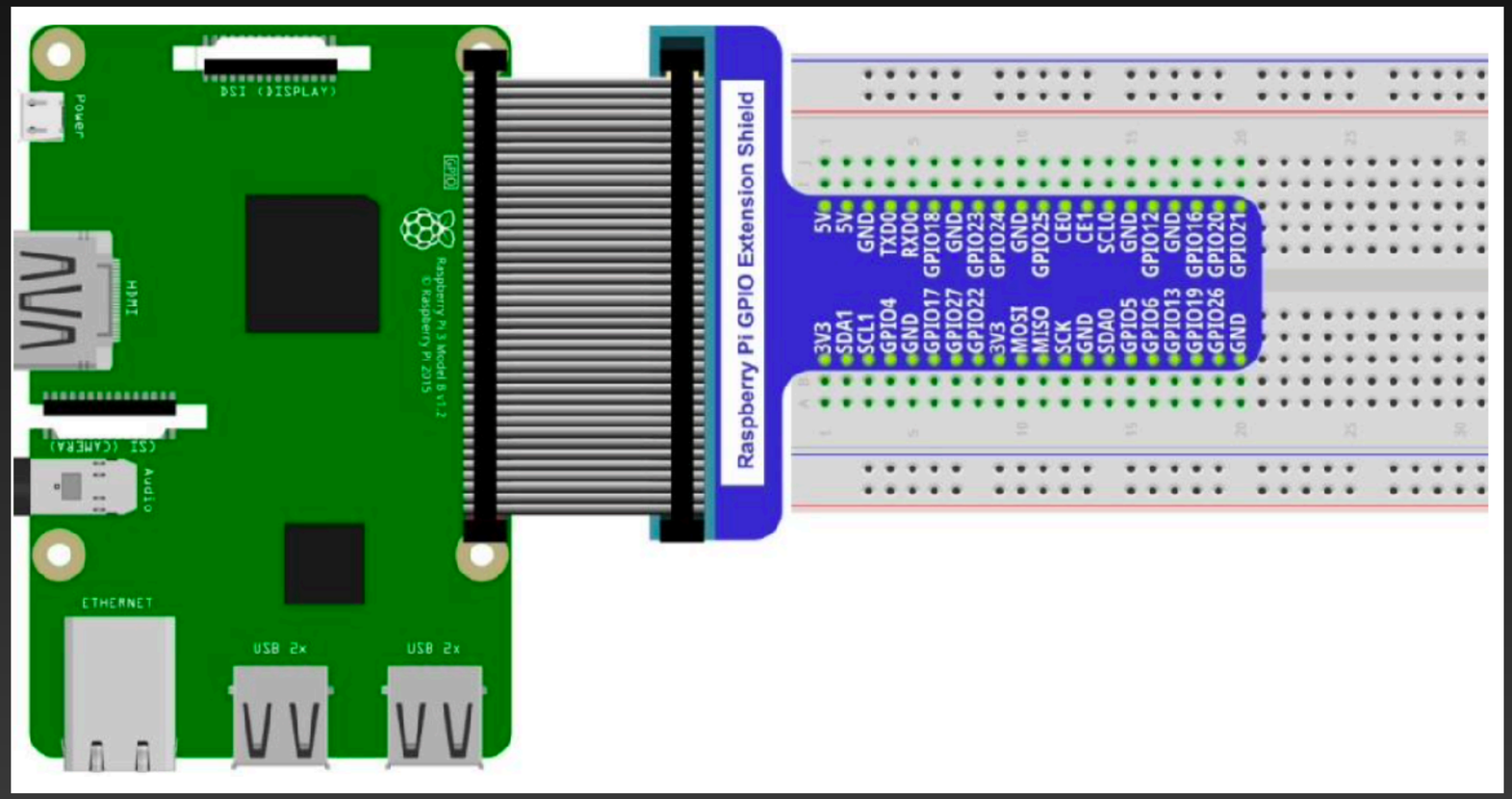
Fundamental

X. Tang

Introducing Extension Board

GPIO Extension Board

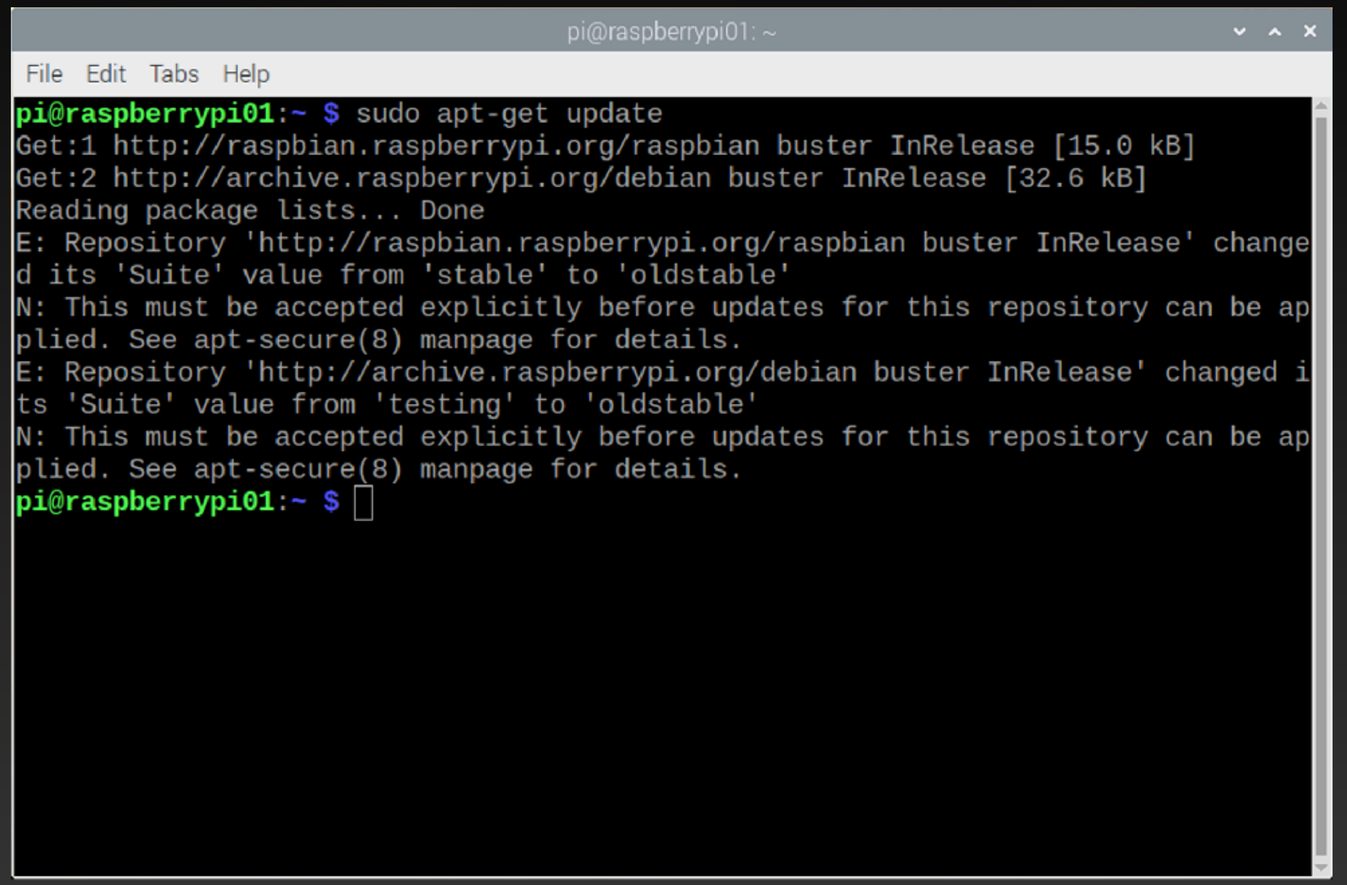
GPIO board is a convenient way to connect the RPi I/O ports to the breadboard directly. The GPIO pin sequence on Extension Board is identical to the GPIO pin sequence of RPi.



Physical Computing from Python - GPIO Library

Install GPIO Library

```
sudo apt-get update
```

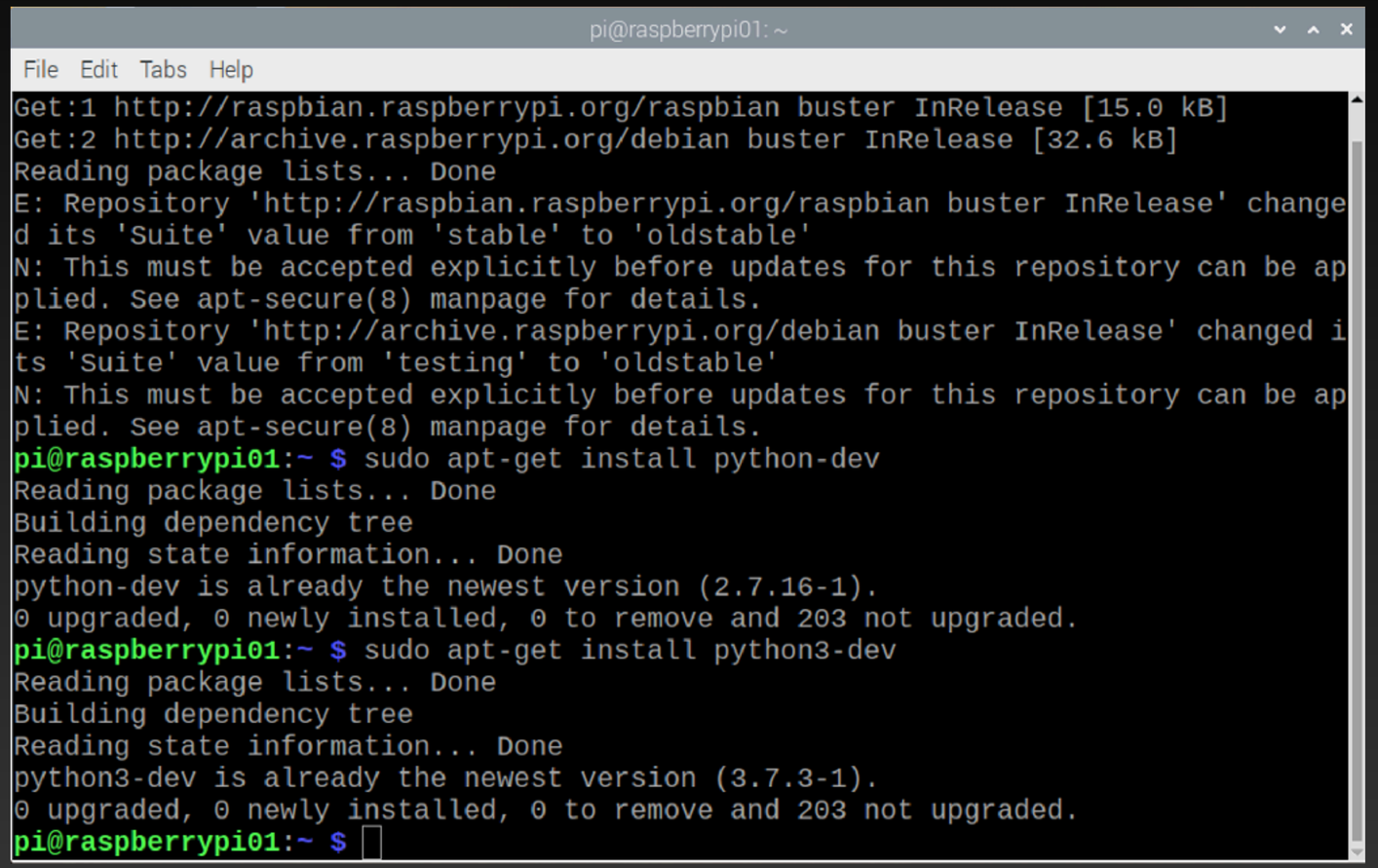


Physical Computing from Python - GPIO Library

Install GPIO Library

```
sudo apt-get install
python-dev
```

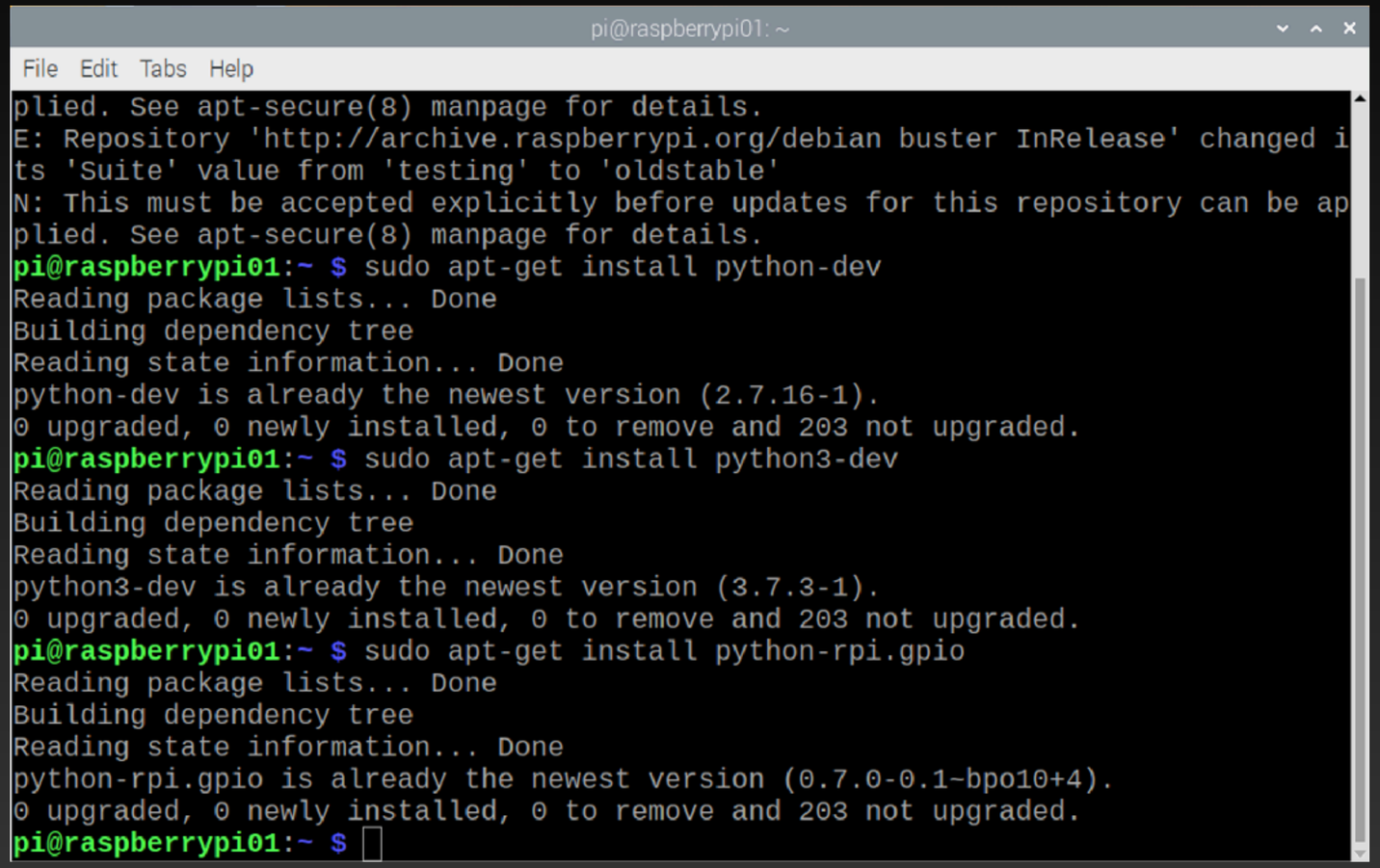
```
sudo apt-get install
python3-dev
```



Physical Computing from Python - GPIO Library

Install GPIO Library

```
sudo apt-get install
libgpiod2 libgpiod-dev
```



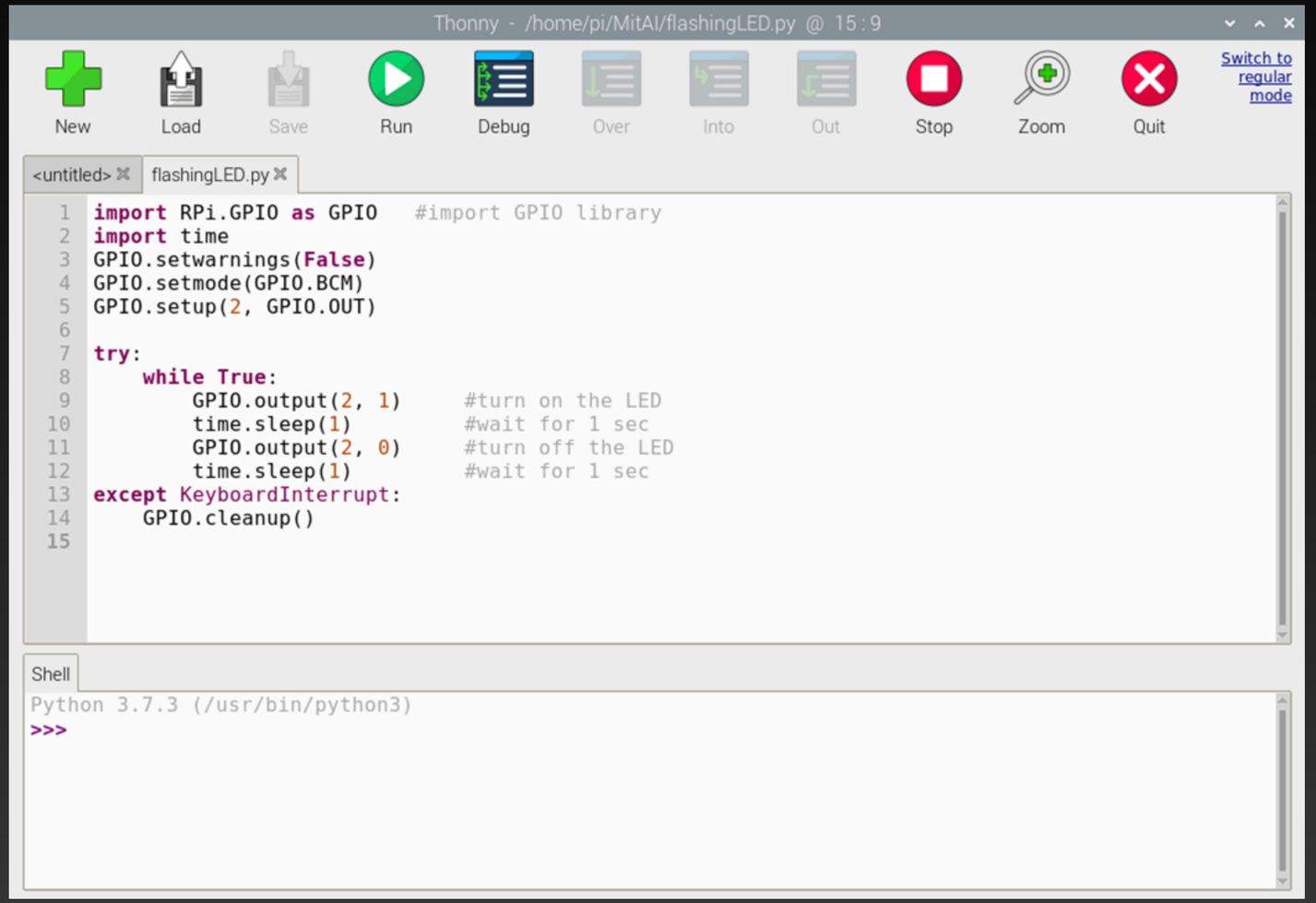
Physical Computing from Python - GPIO Library

The GPIO Library

The GPIO library is called RPi.GPIO and it should already be installed on your raspberry Pi 4/5.

When you programming in python, this library must be included in your code. The statement is:

```
import RPi.GPIO as GPIO
```



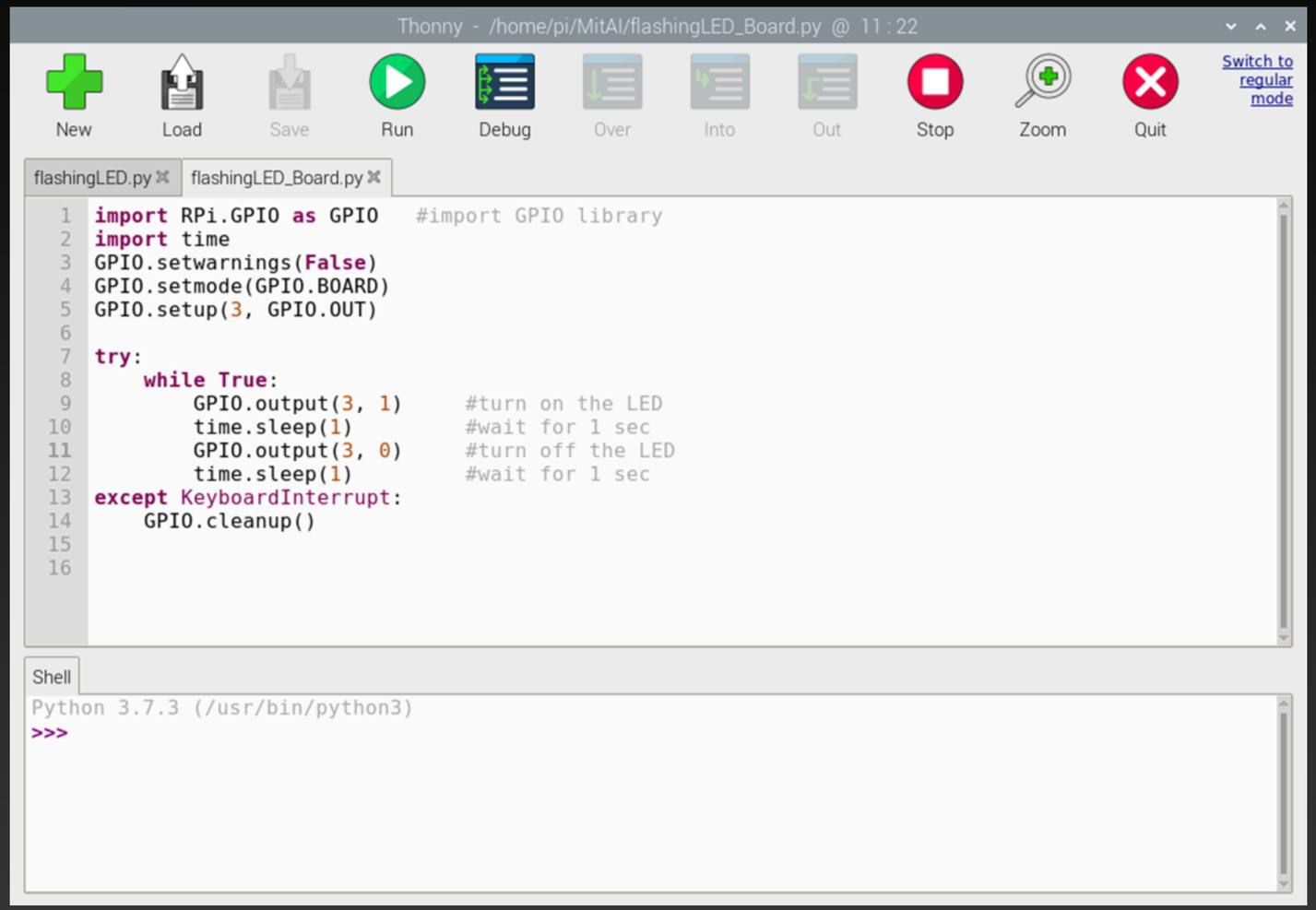
Physical Computing from Python - GPIO Library

Pin Numbering

There are two ways to refer to the GPIO pins.

1. Use the BOARD numbering, where the pin numbers on the GPIO connector of the Raspberry Pi 4/5 are used.

```
GPIO.setmode(GPIO.BOARD)
```



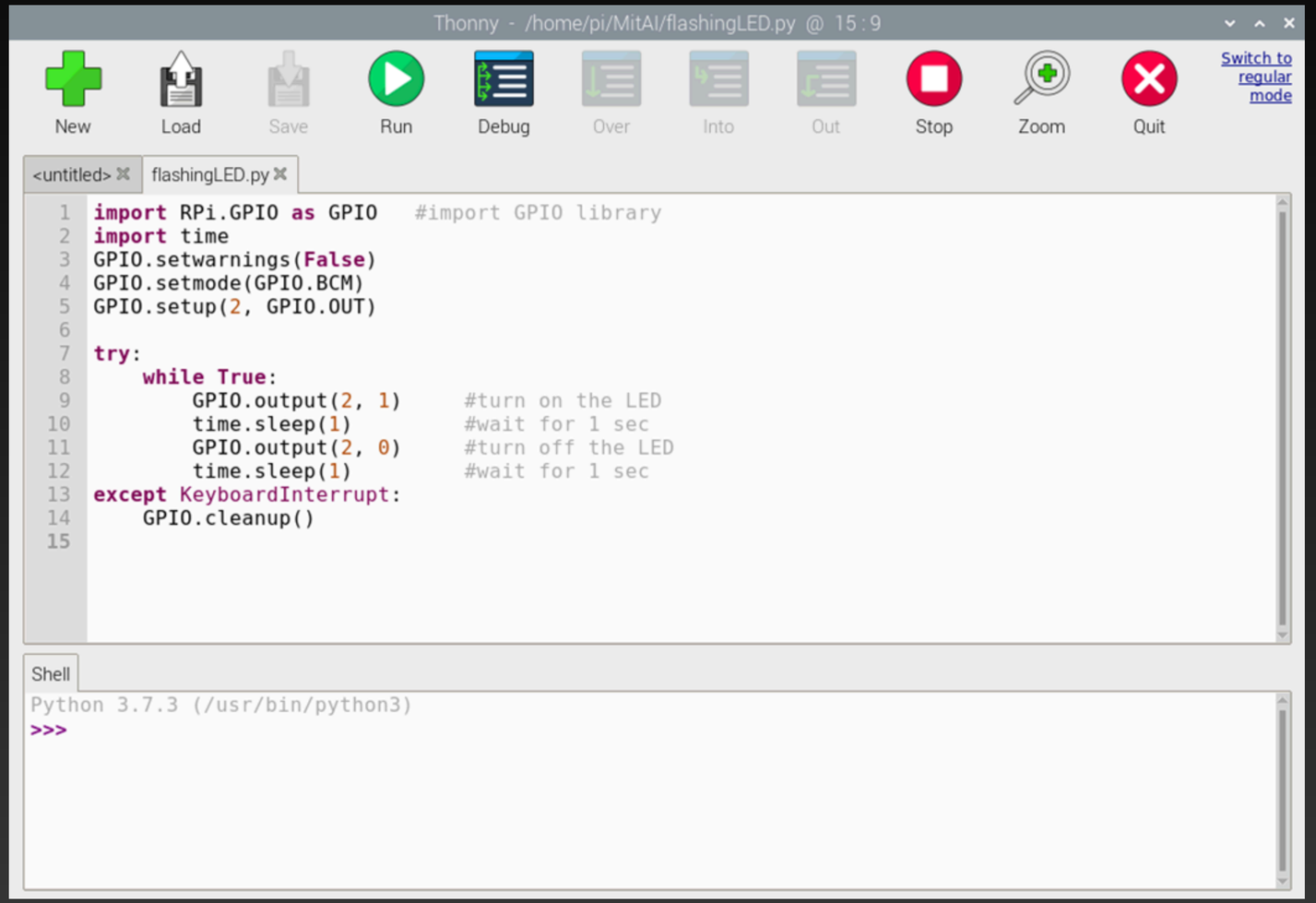
Physical Computing from Python - GPIO Library

Pin Numbering

There are two ways to refer to the GPIO pins.

2. Use the BCM, which is preferred. It uses the channel numbers refers to which pin on the board.

```
GPIO.setmode(GPIO.BCM)
```



Physical Computing from Python - GPIO Library

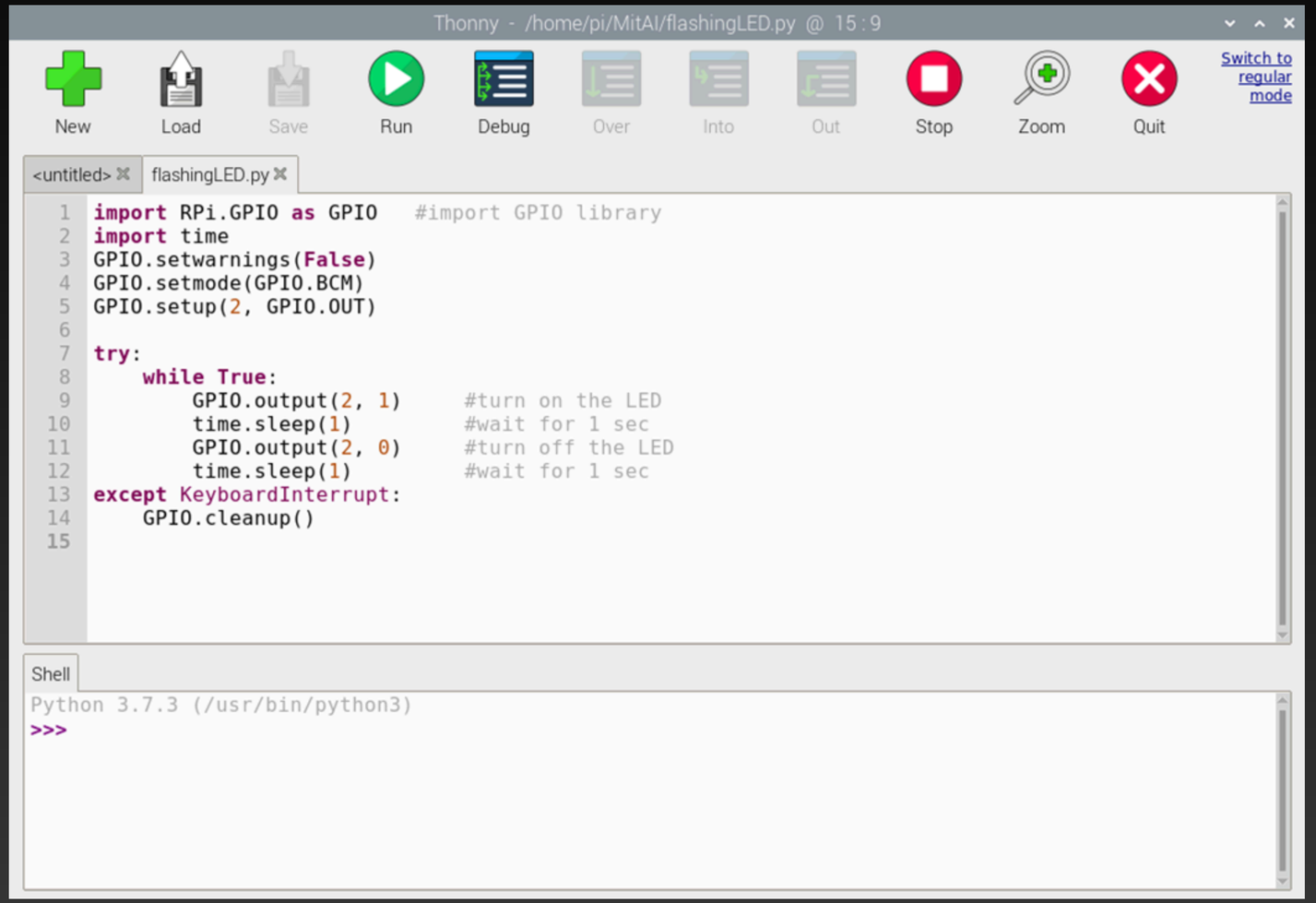
Channel(I/O port pin) Config

Input configuration

You need to configure the channels(or port pins) you are using whether they are input or output channels.

```
GPIO.setup(channel,GPIO.IN)
```

Channel refers to the channel number based on the set mode statement in previous setup.



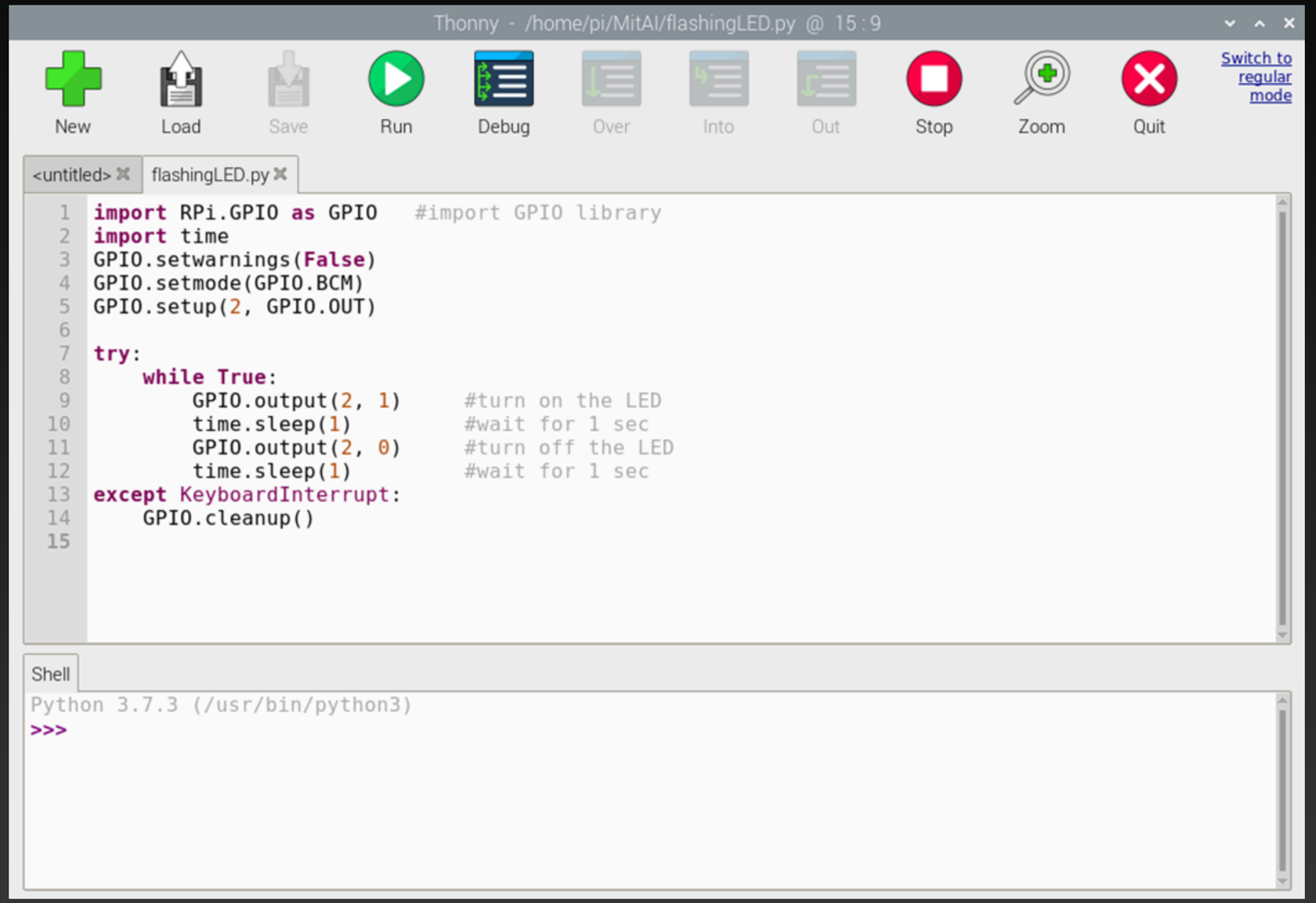
Physical Computing from Python - GPIO Library

Channel(I/O port pin) Config

Output configuration

```
GPIO.setup(channel,GPIO.OUT)
```

Channel refers to the channel number based on the set mode statement in previous setup.



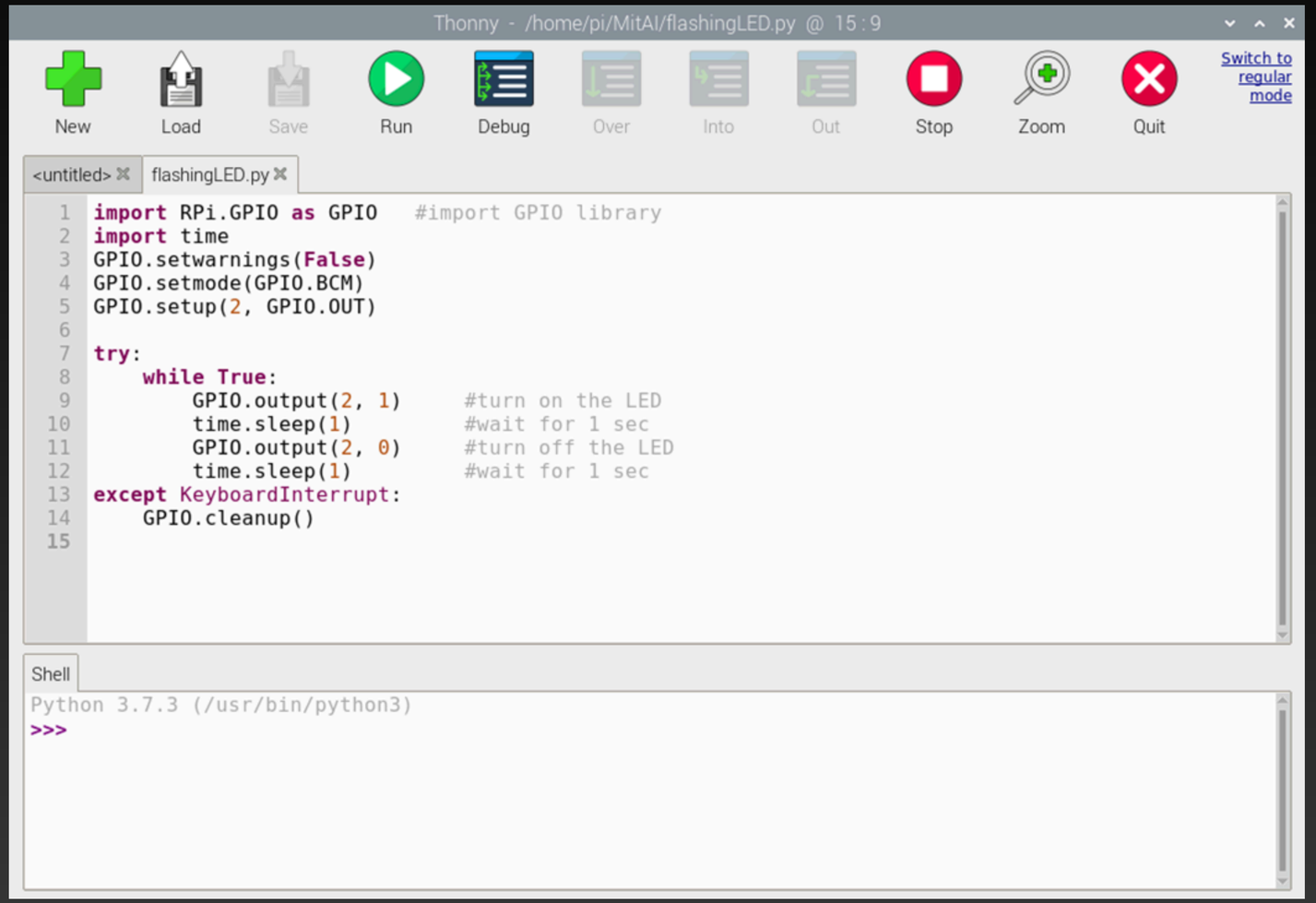
Physical Computing from Python - GPIO Library

Channel(I/O port pin) Config

Output configuration

We can specify a value for an output pin during it's setup. For example, we can config a channel as output and at the same time set its value to logic HIGH(3.3V)

```
GPIO.setup(channel,GPIO.OUT
, initial=GPIO.HIGH)
```



Physical Computing from Python - GPIO Library

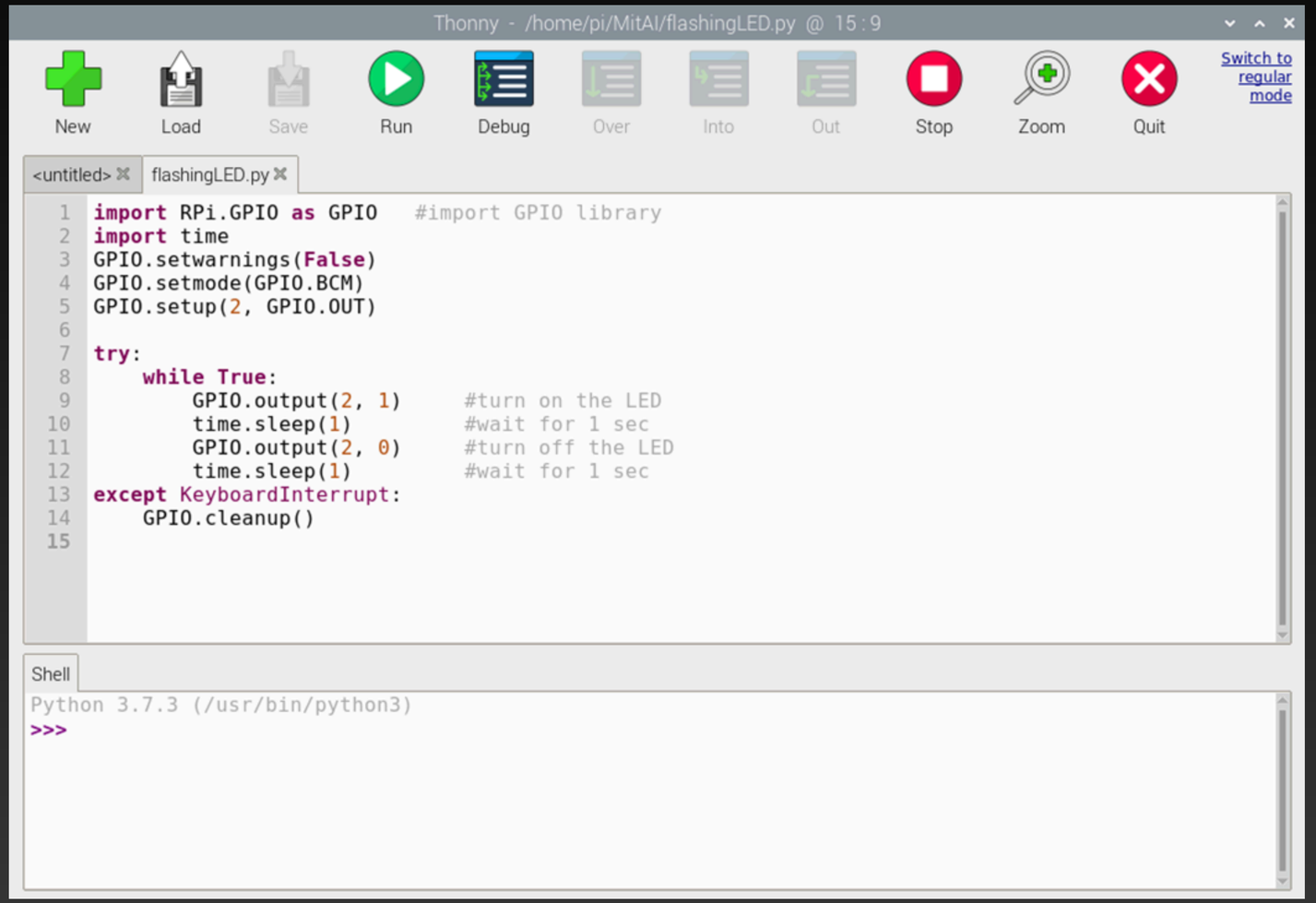
Channel(I/O port pin) Config

Output configuration

To send data to an output port pin we can use the following statement:

```
GPIO.setup(channel, value)
```

Where value can be 0(or GPIO.LOW, or False), or 1 (or GPIO.HIGH, or True)



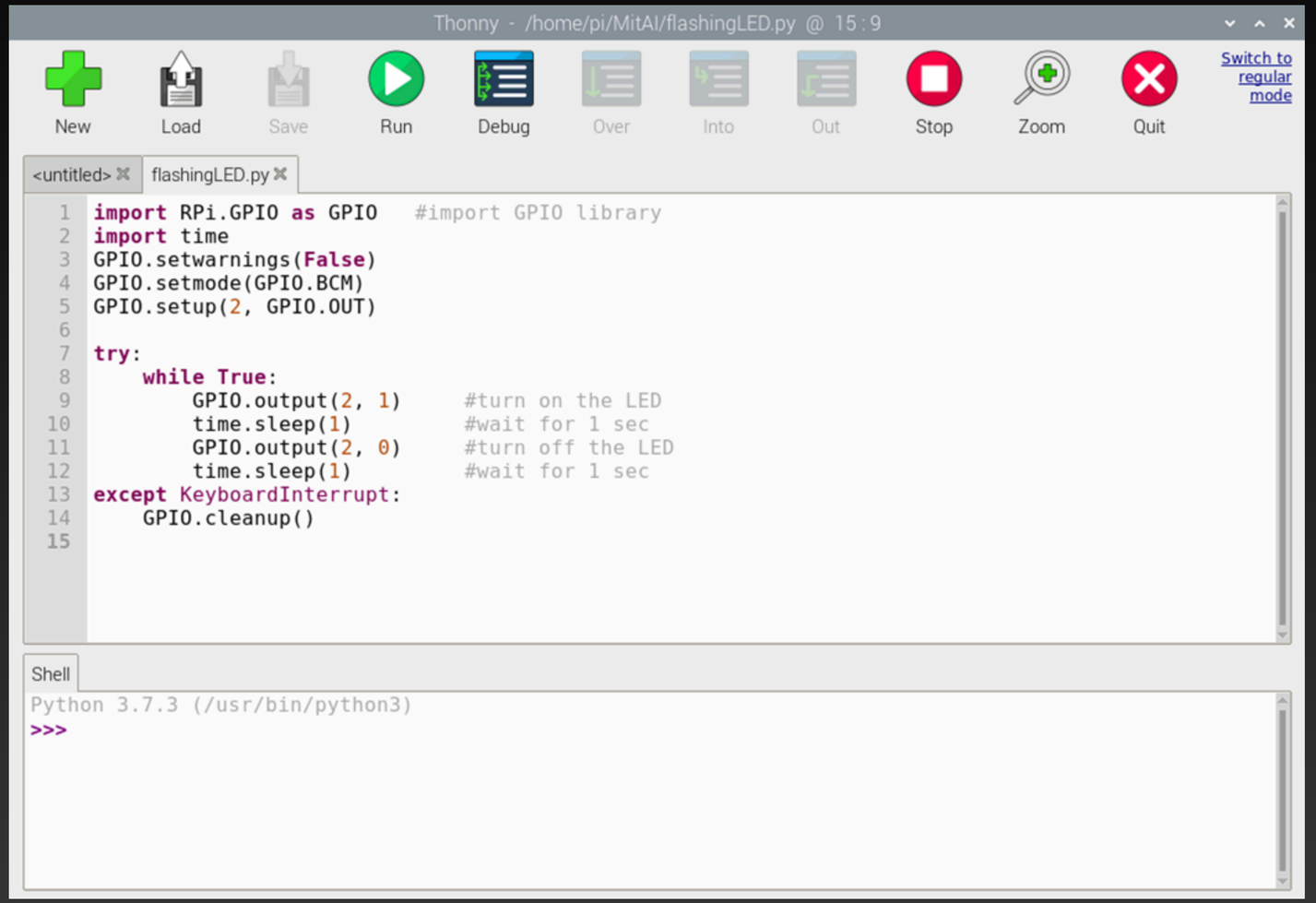
Physical Computing from Python - GPIO Library

Channel(I/O port pin) Config

Output configuration

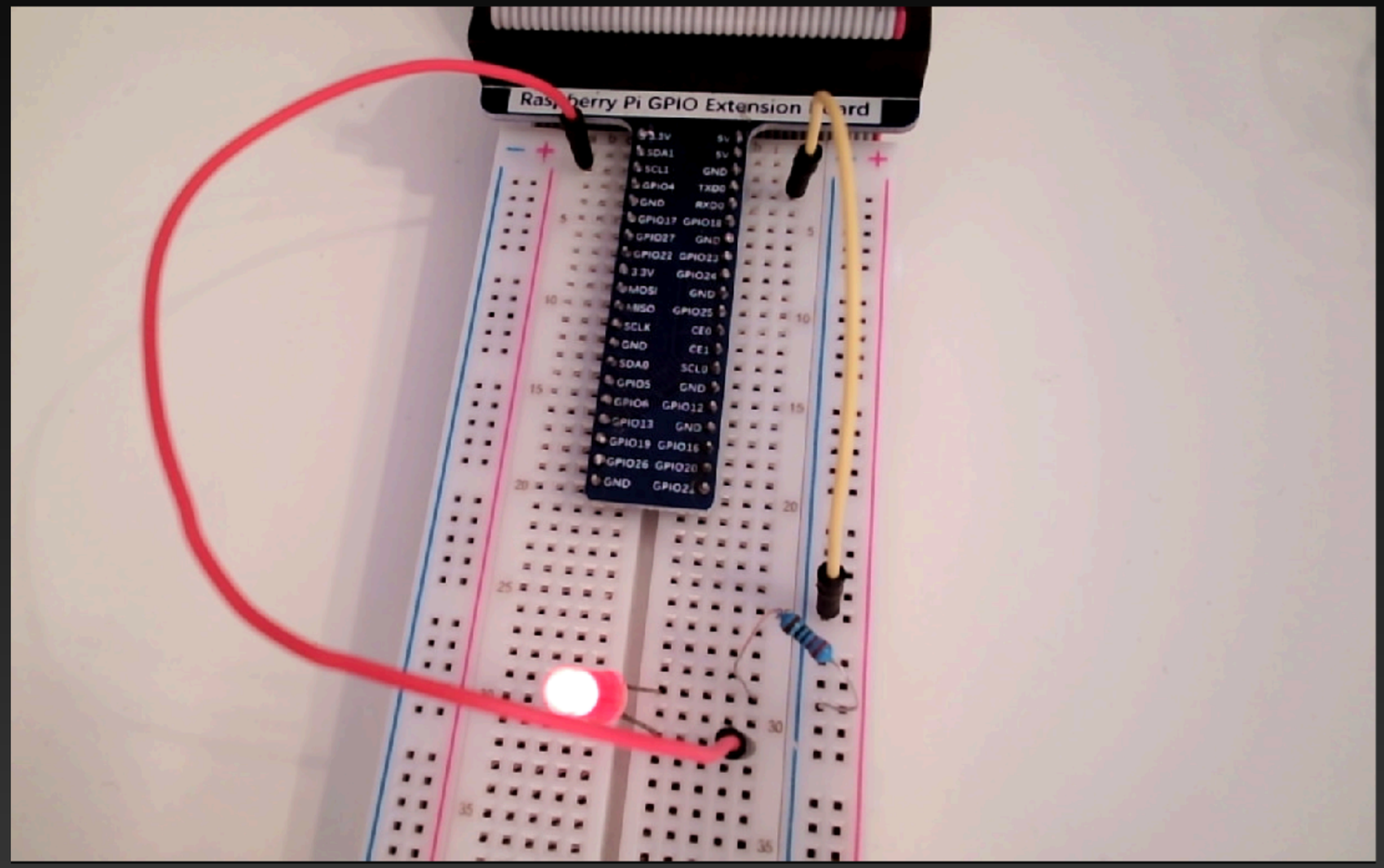
At the end of the program, we should return all the used resources to the operating system. This is done by including the following statement at the end of our program:

```
GPIO.cleanup()
```



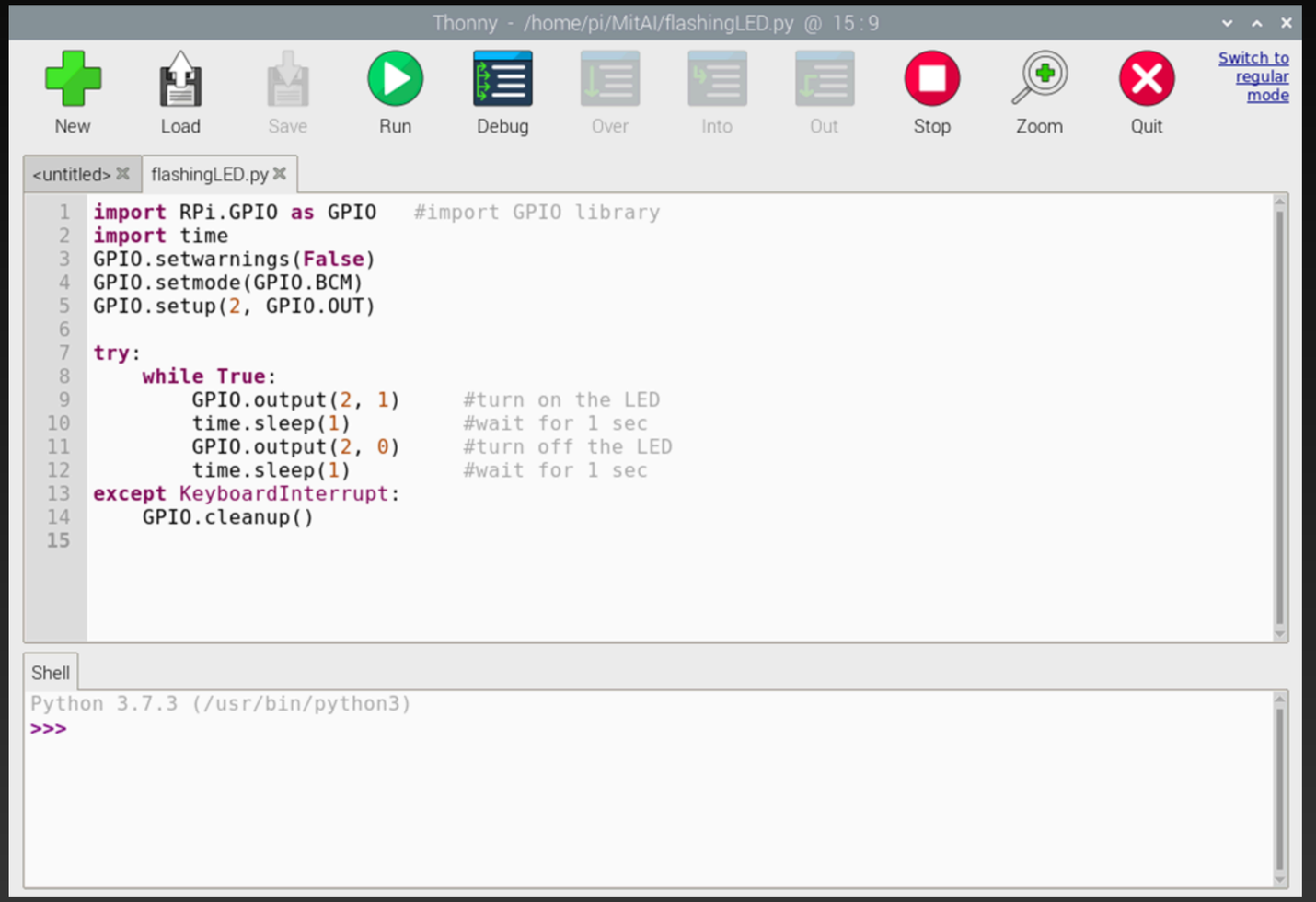
Flashing LED - Accessing the External World

Connect the wires use GPIO2 as output.



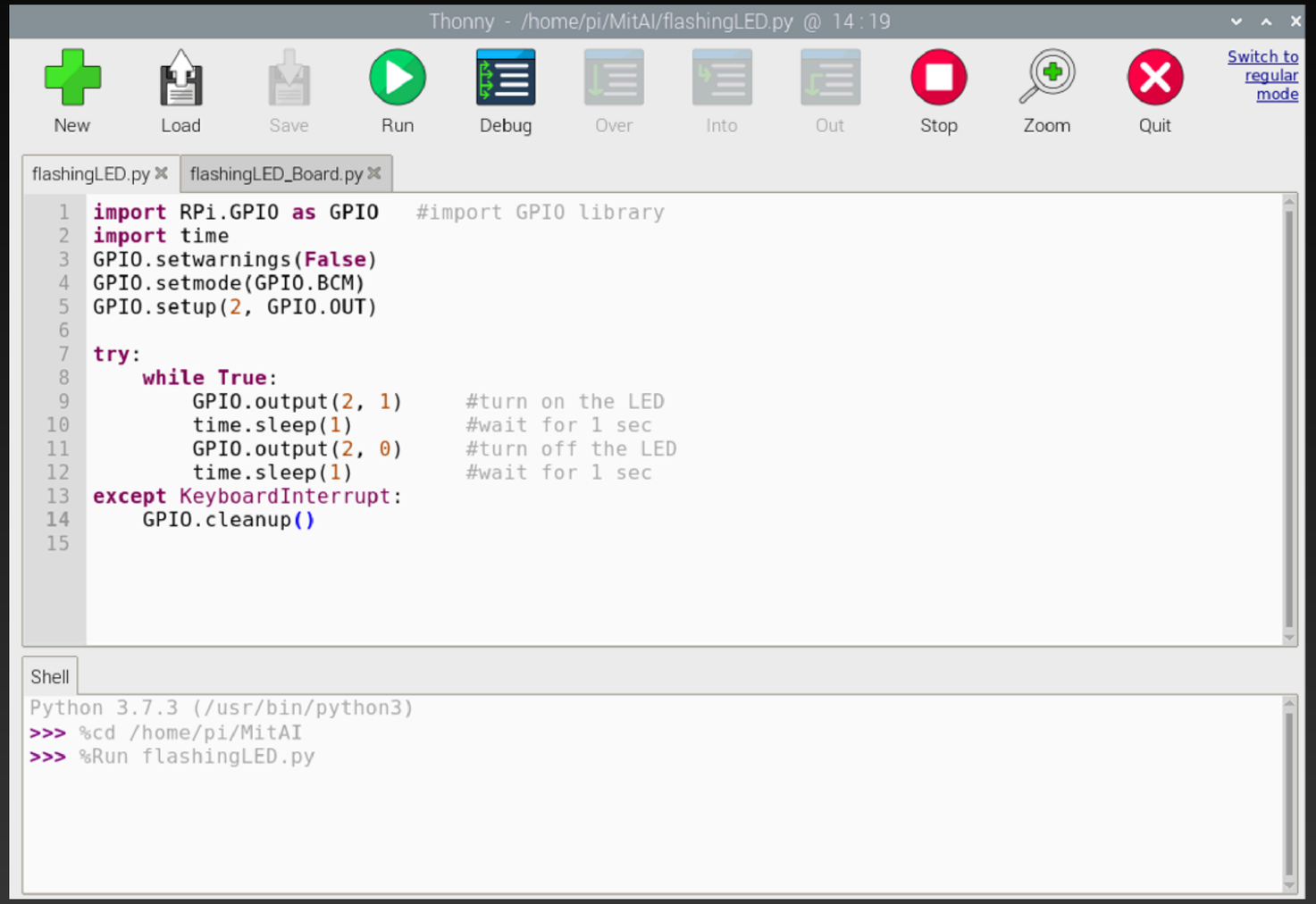
Flashing LED - Accessing the External World

Code Flashing LED program in Thonny.



Flashing LED - Accessing the External World

Run the code.



Flashing LED - Accessing the External World

What did you see?

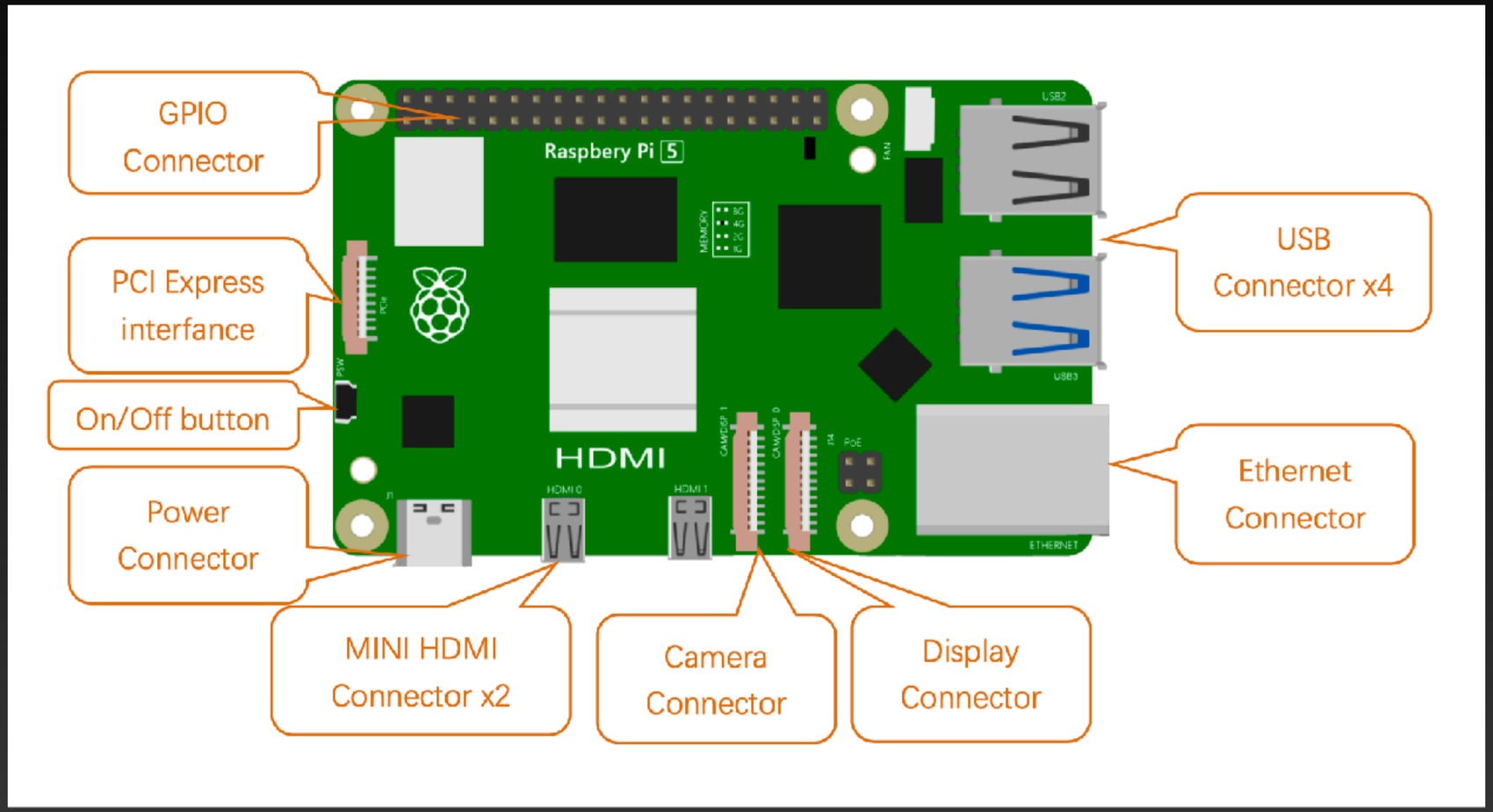
IoT with MIT App Inventor

Back at 8:12PM

Please update your name with studentID

In Zoom meeting

GPIO control in python on Raspberry Pi

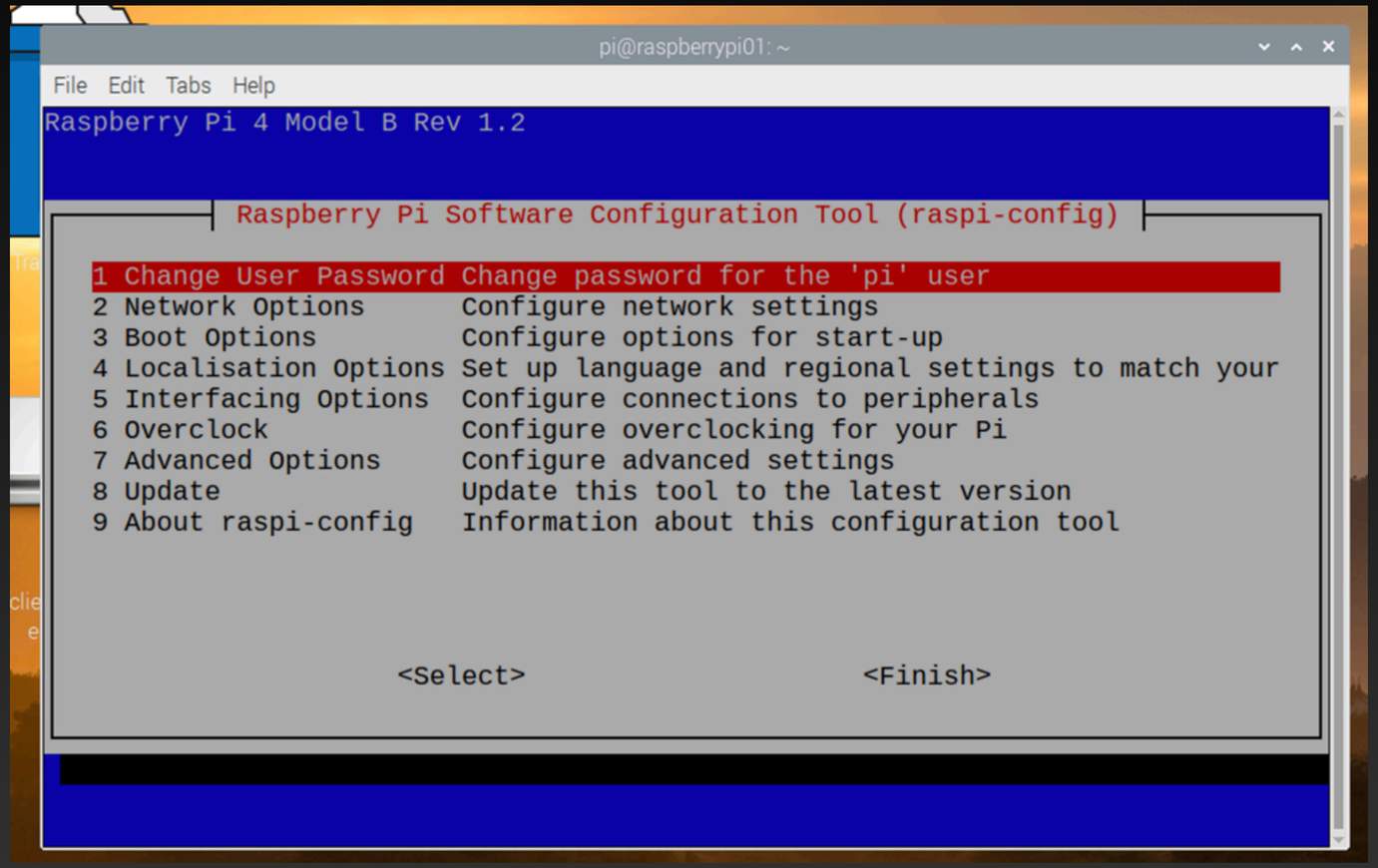


Remote Access -Setup

- $ sudo raspi-config
- config P2 SSH and P3 VNC to YES under Interface Options
- Reboot
- SSH access. Windows, download Putty. Https://www.putty.org/
- MAC, SSH pi@IP
- Windows, download https://www.realvnc.com/en/connect/download/viewer/
- MAC, app store, windows remote desktop

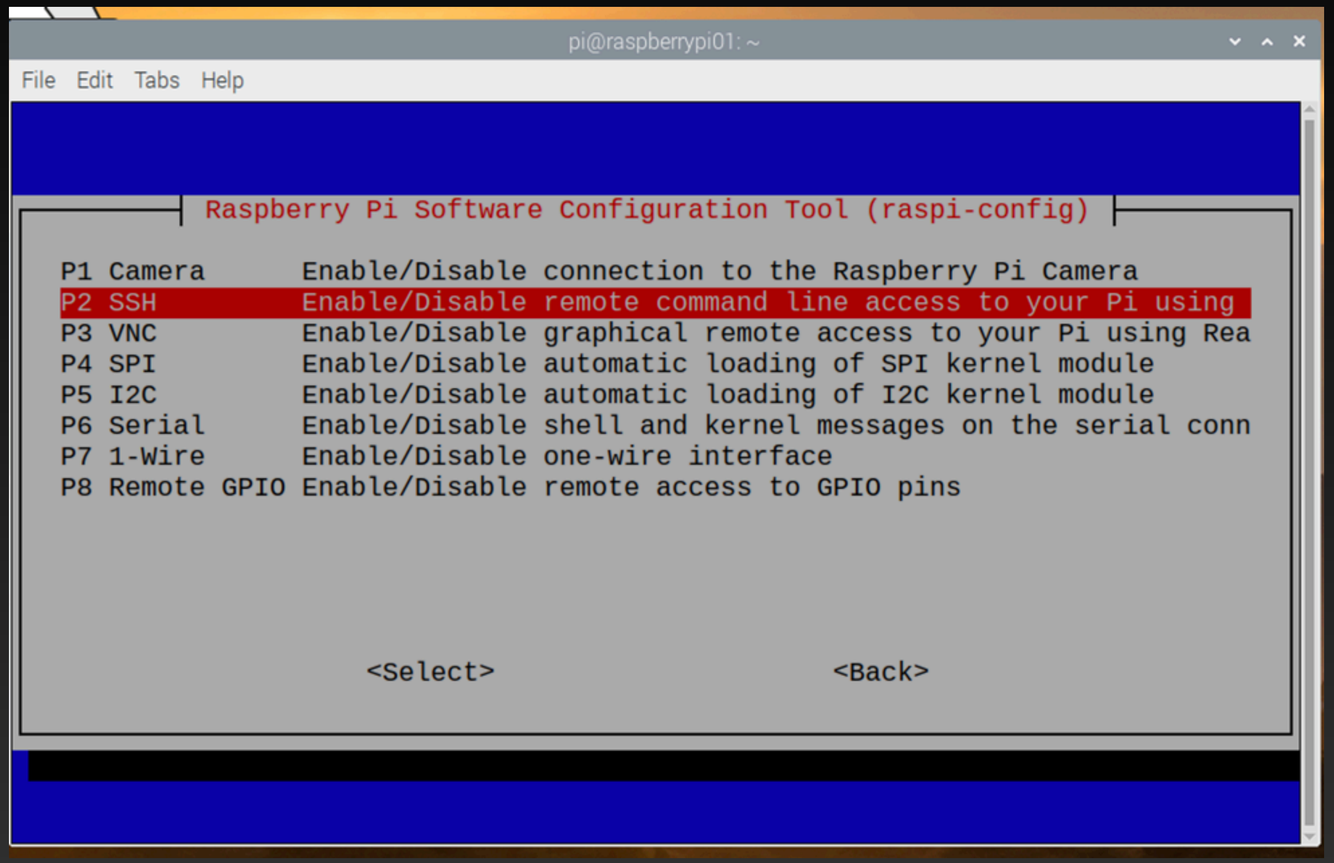
Remote Access -Setup

- $ sudo raspi-config
- Go to Interfacing Options



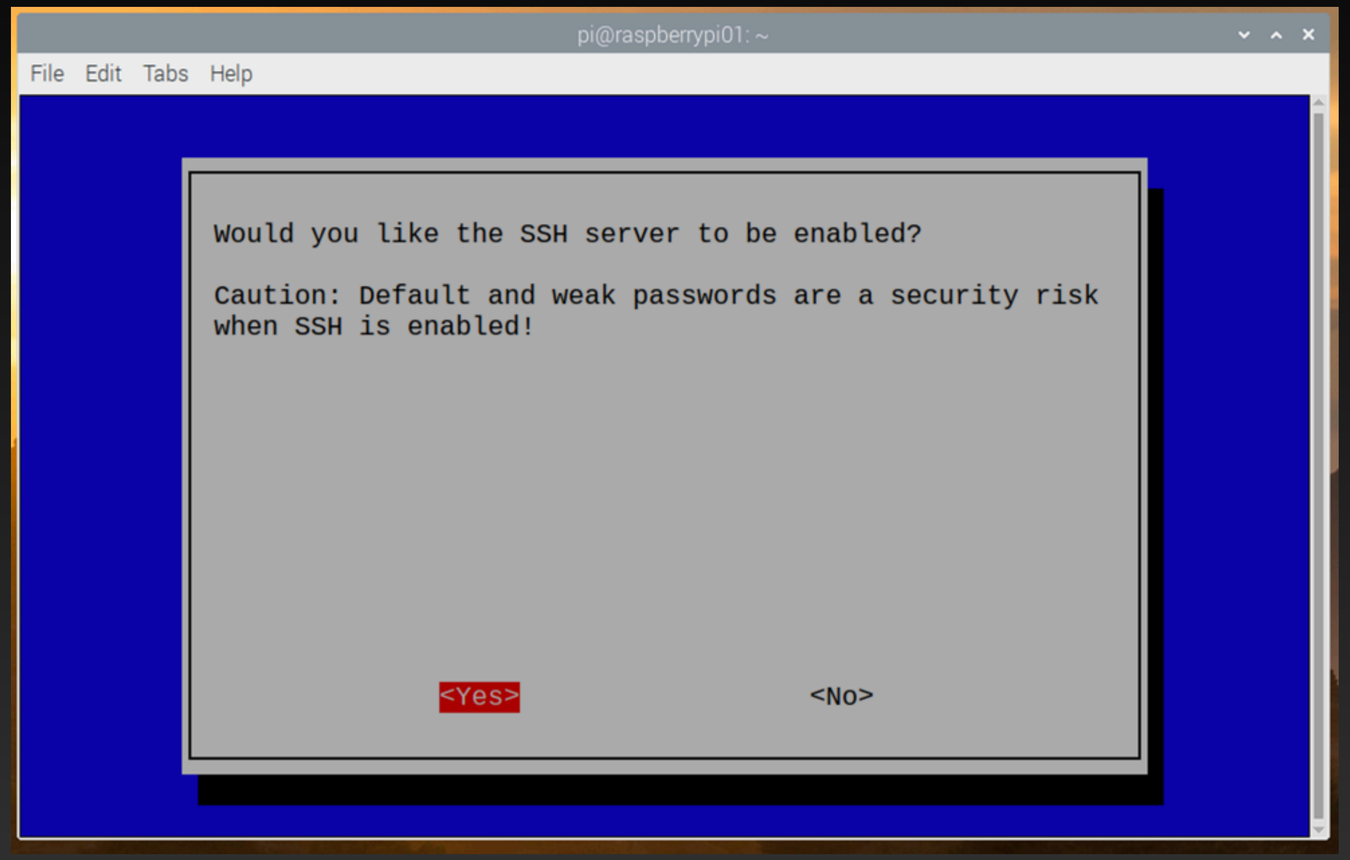
Remote Access -Setup

- Choose P2 SSH, hit enter



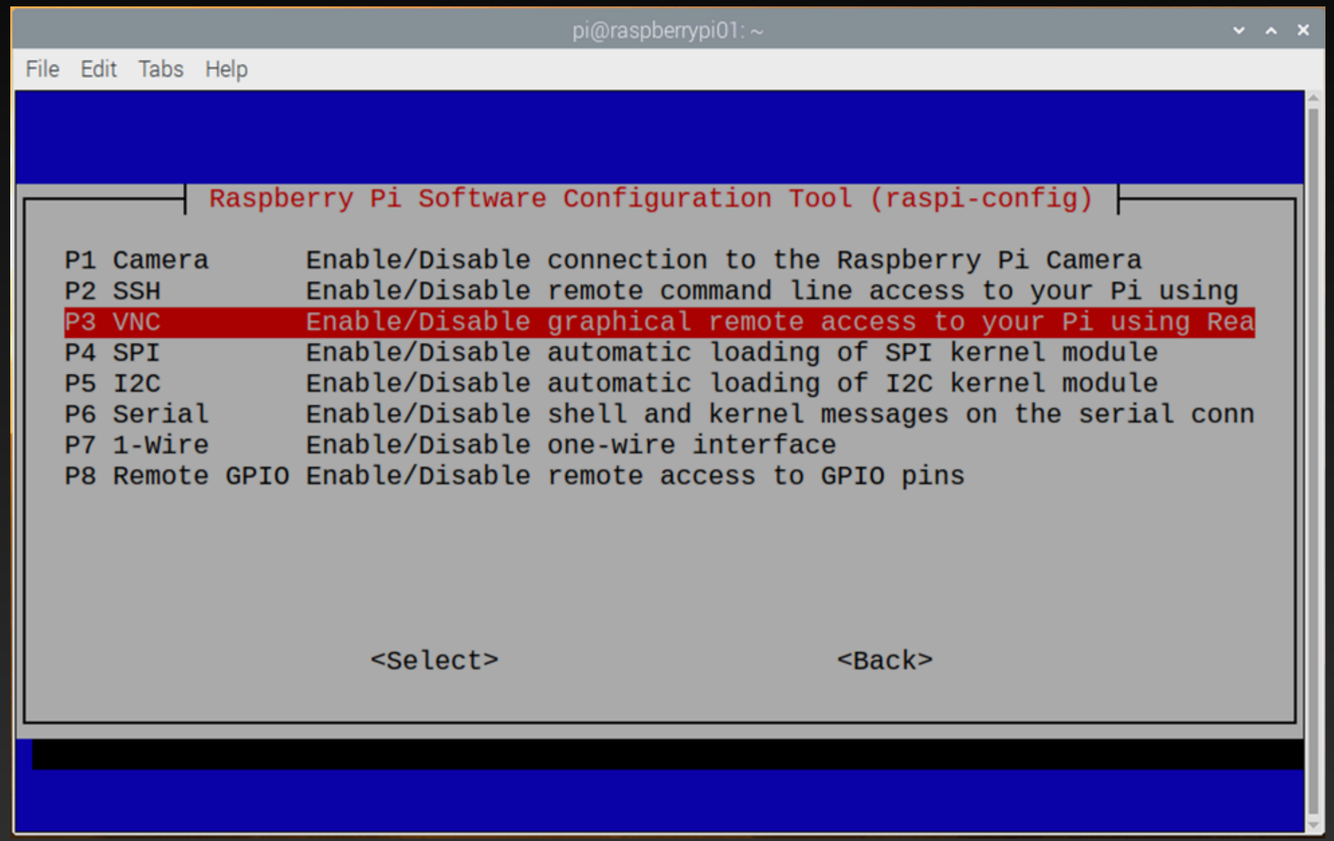
Remote Access -Setup

- Select Yes, hit enter



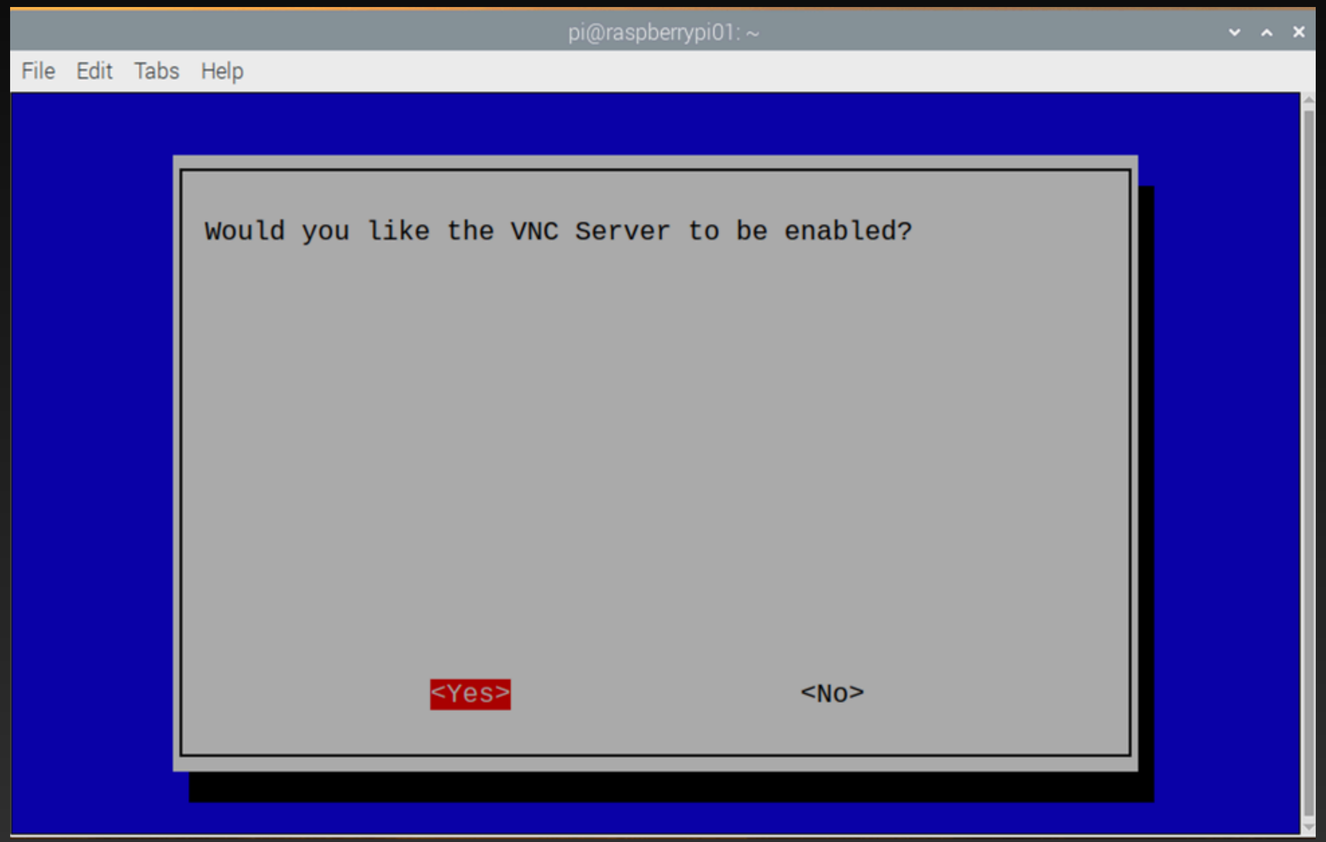
Remote Access -Setup

- Choose P3 VNC, hit enter



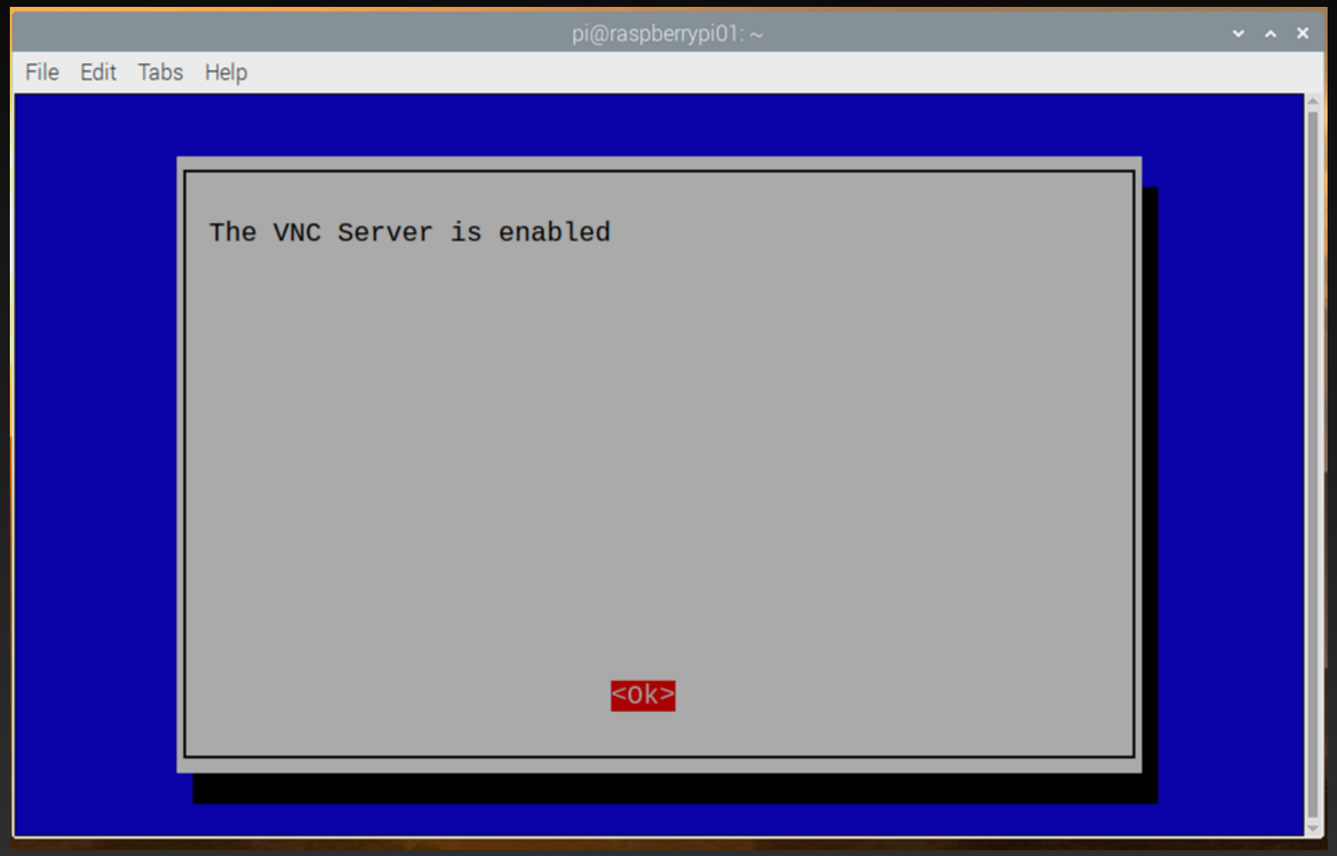
Remote Access -Setup

- Select Yes, hit enter



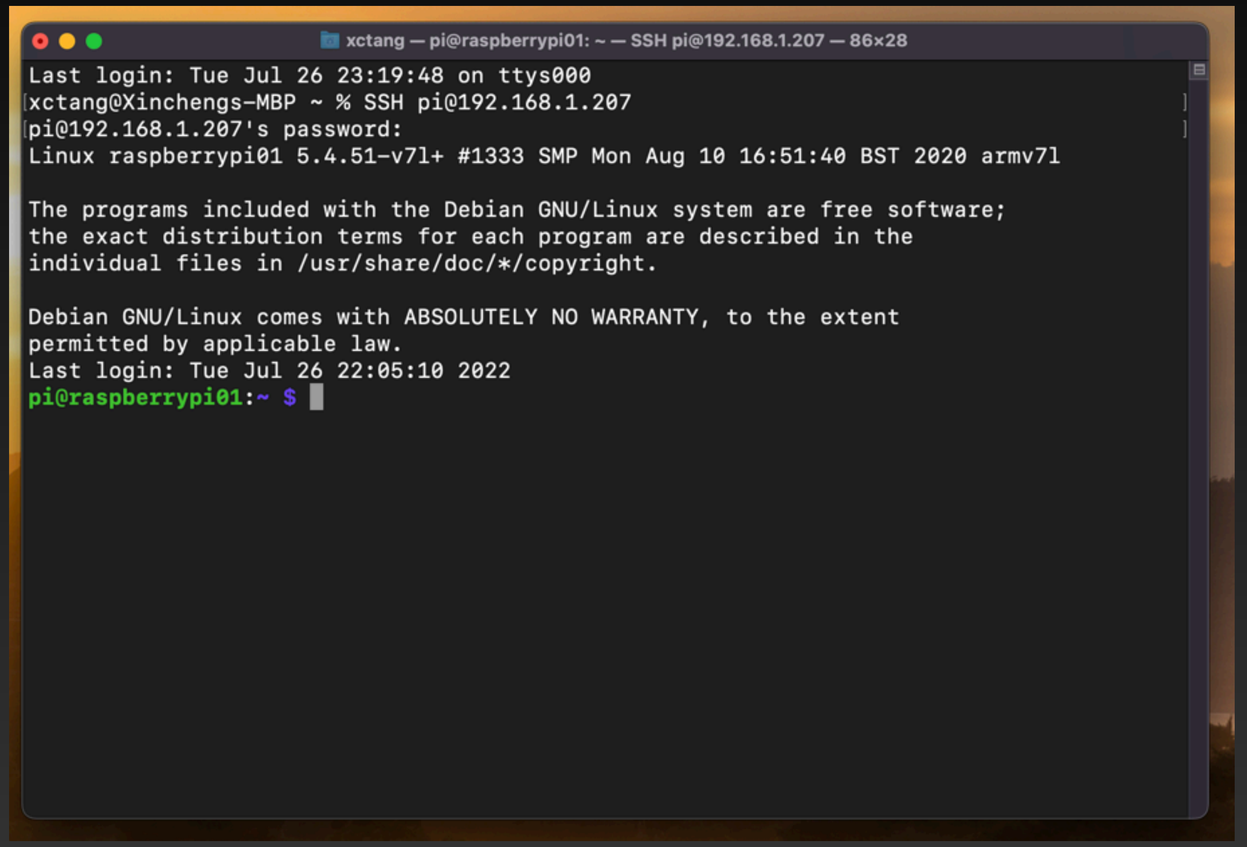
Remote Access -Setup

- You will see screen which confirm VNC is enabled



Remote Access-SSH

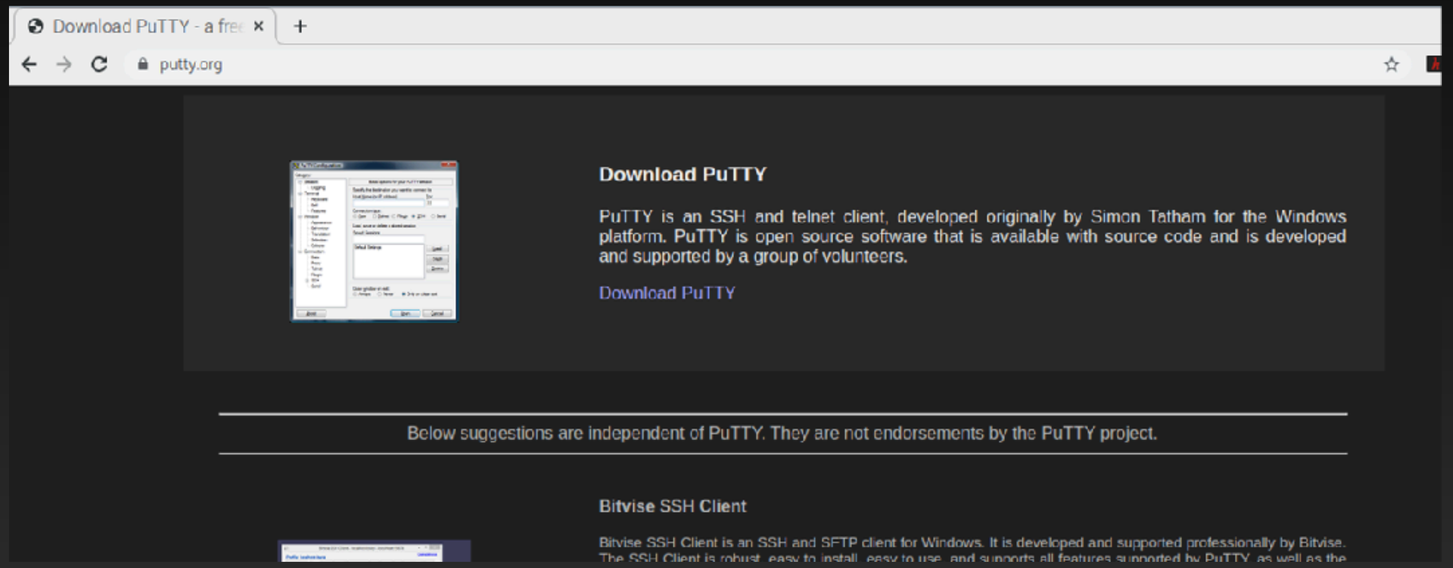
- SSH on MAC



Remote Access-SSH

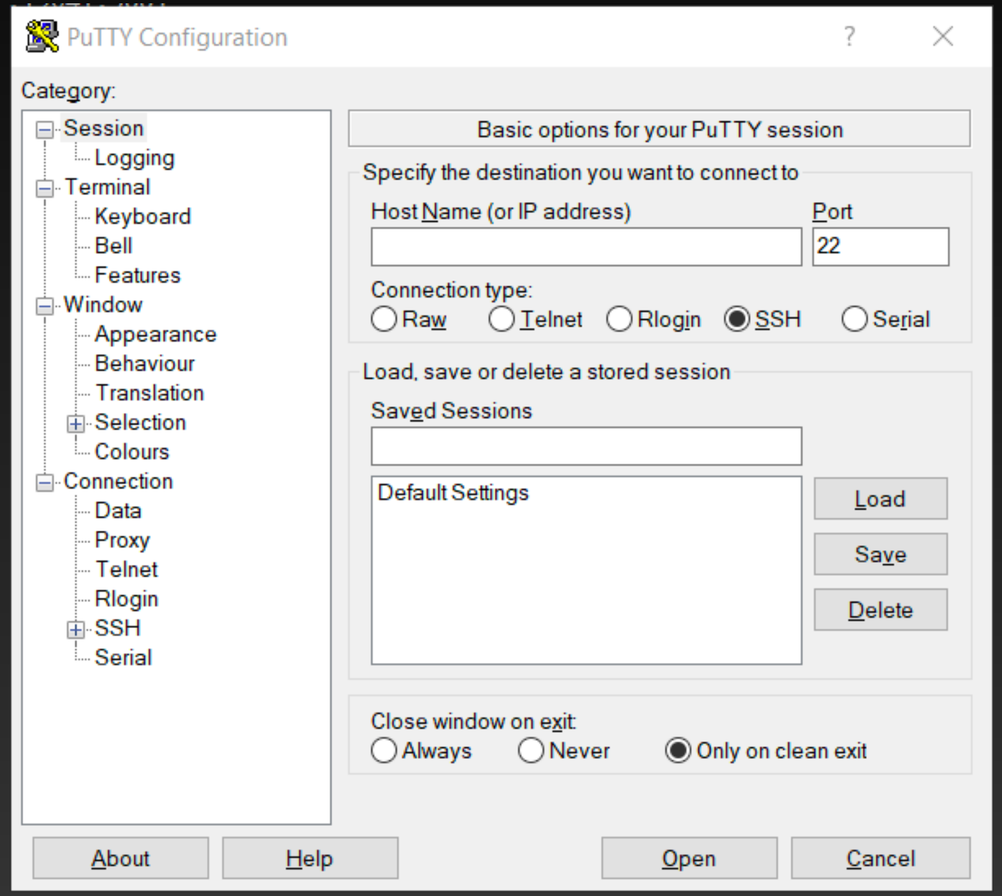
- SSH on Window Machine
- download Putty

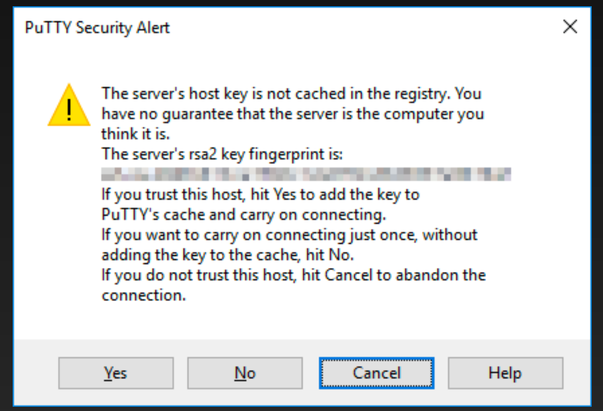
Https://www.putty.org/



Remote Access-SSH

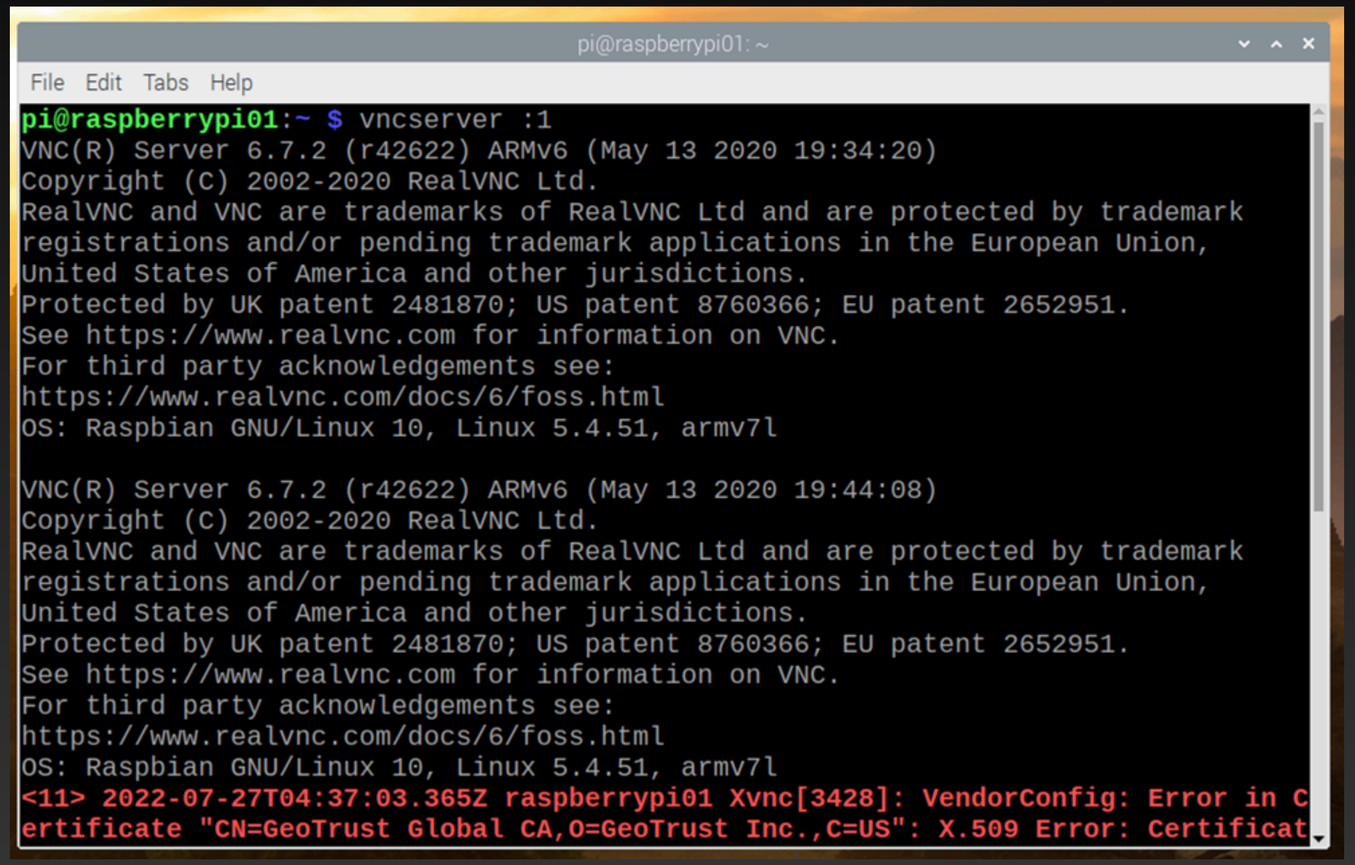
- Putty config screen
- Enter your IP, click open
- You will see security alert, Click Yes to accept
- You will prompt to enter your Pi username and password





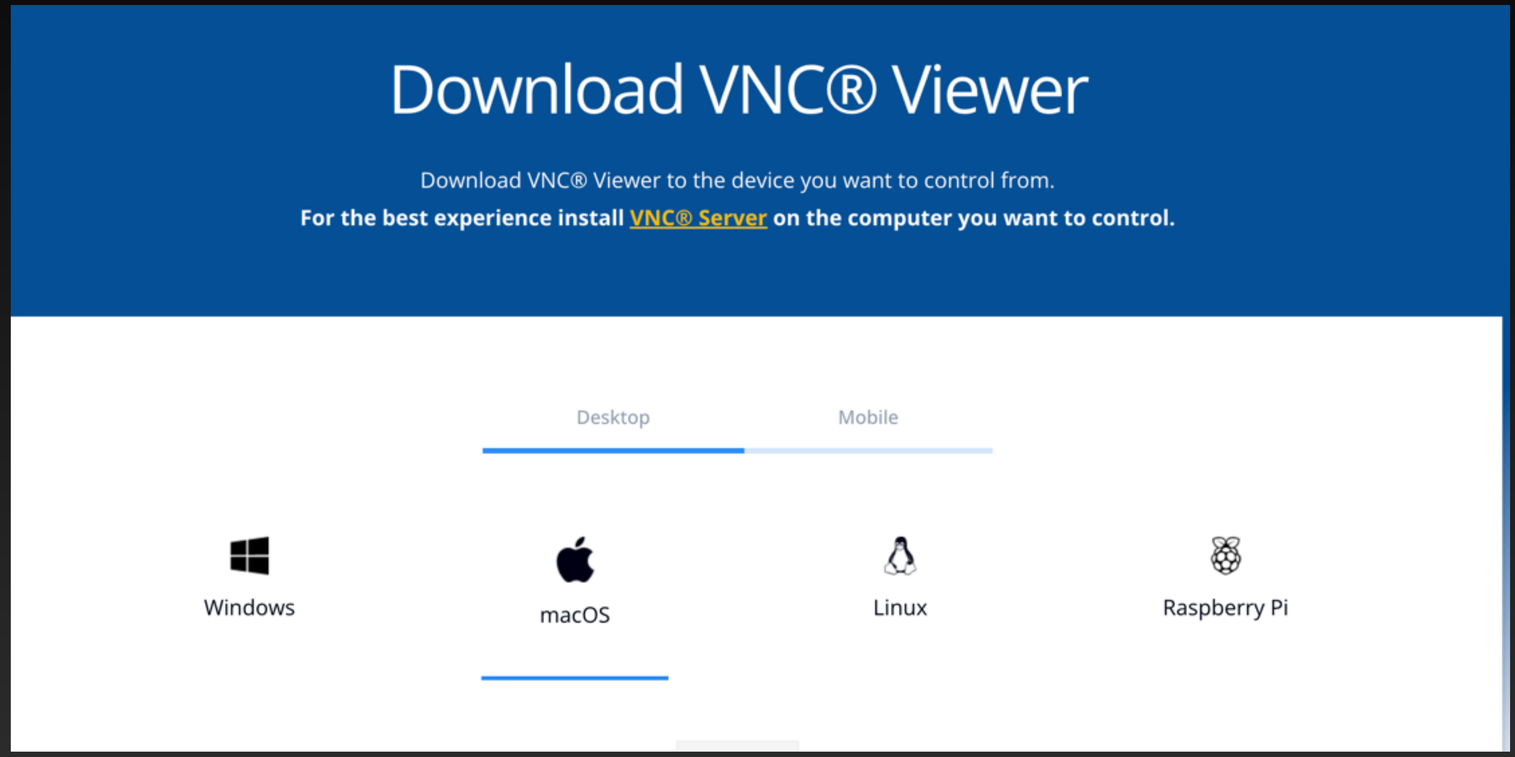
Remote Access-VNC

- VNC consists of two part, VNC server and the VNC Viewer.
- VNC server run on RaspberryPi and VNC viewer run on your Window PC



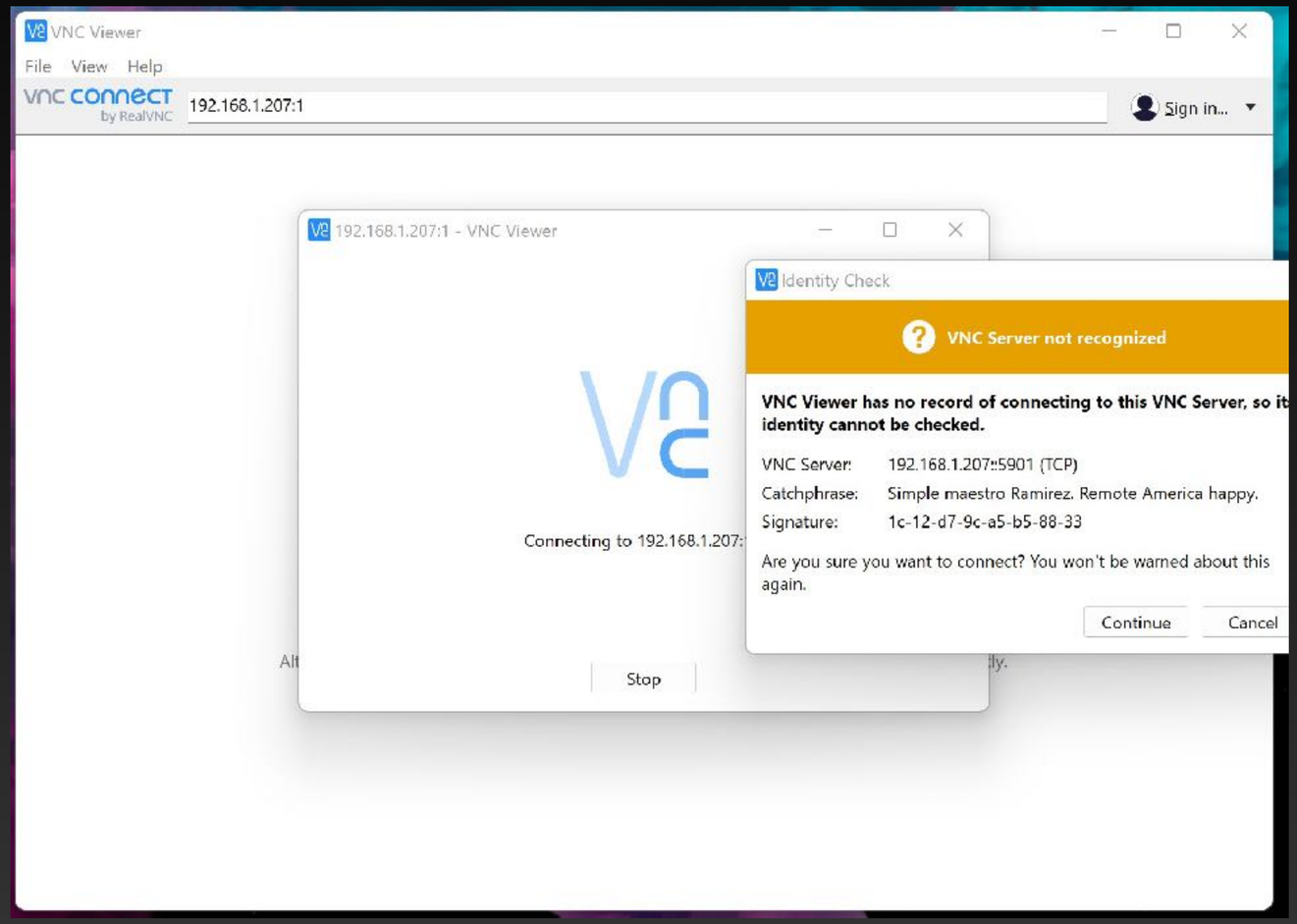
Remote Access-VNC

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



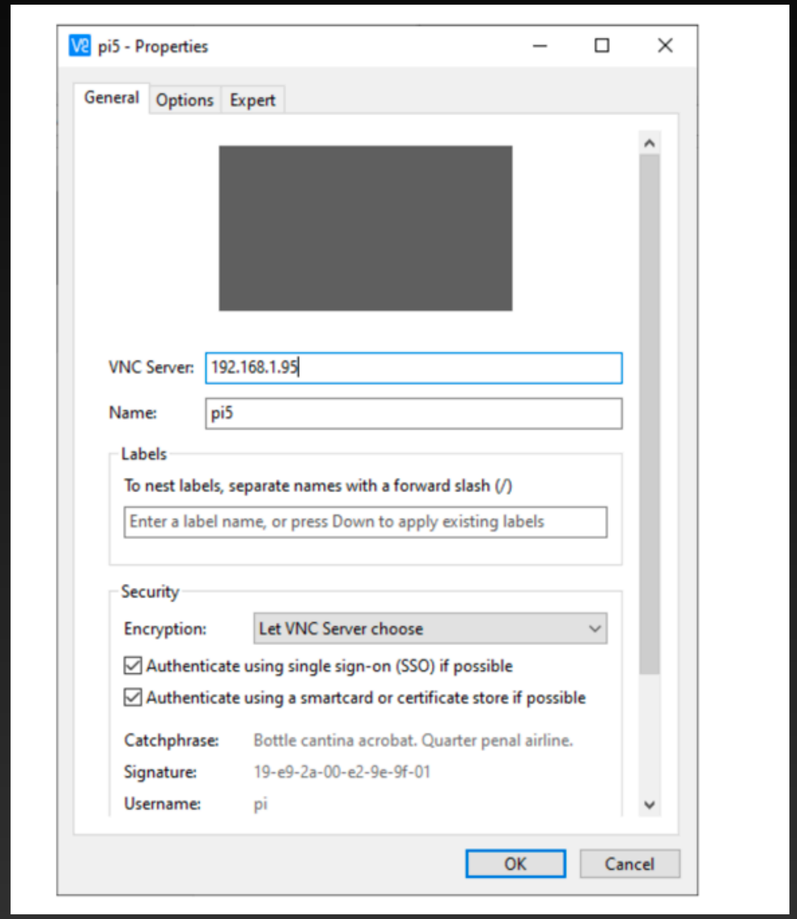
Remote Access-VNC

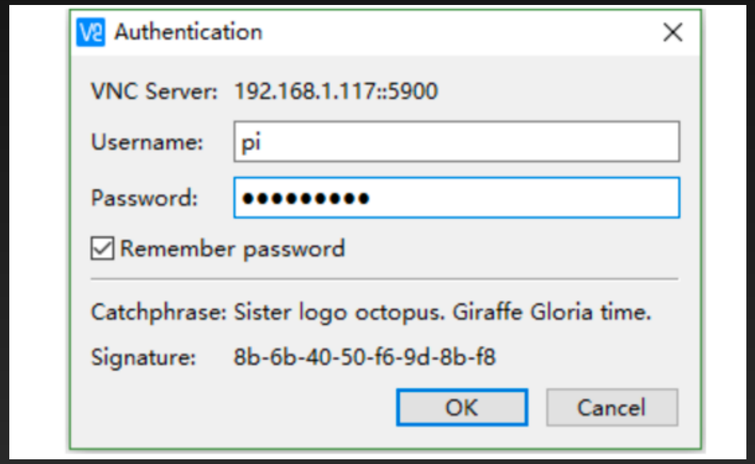
- Put your Pi IP address in the installed VNC viewer



Remote Access-VNC

- Put your Pi IP address in the installed VNC viewer





Remote Access-VNC

- Enter your username and password