

**MIT AI2 204**

# **IoT with MIT App Inventor**

**Fundamental**

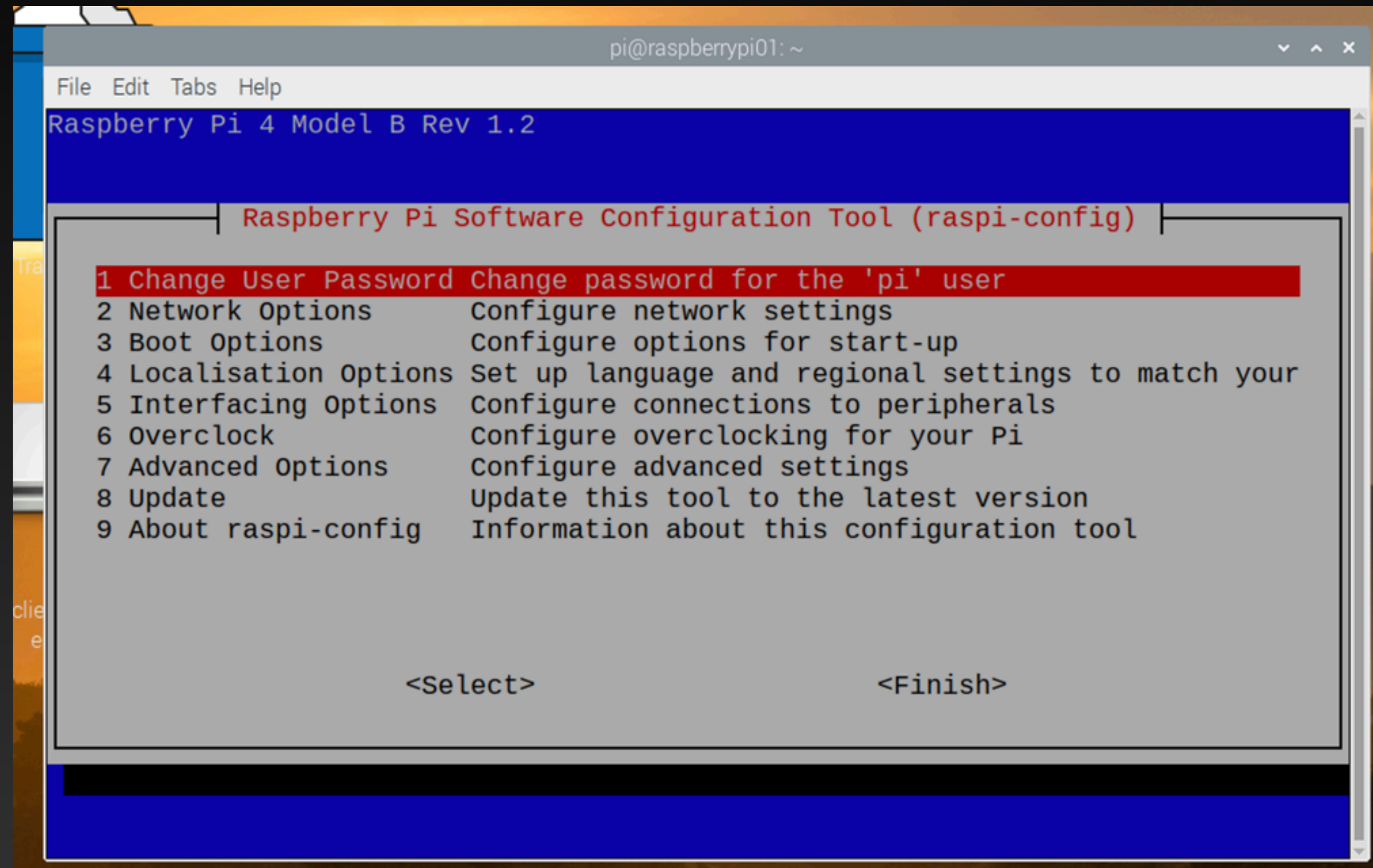
Xincheng Tang

# 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 or Mac, download <https://www.realvnc.com/en/connect/download/viewer/>

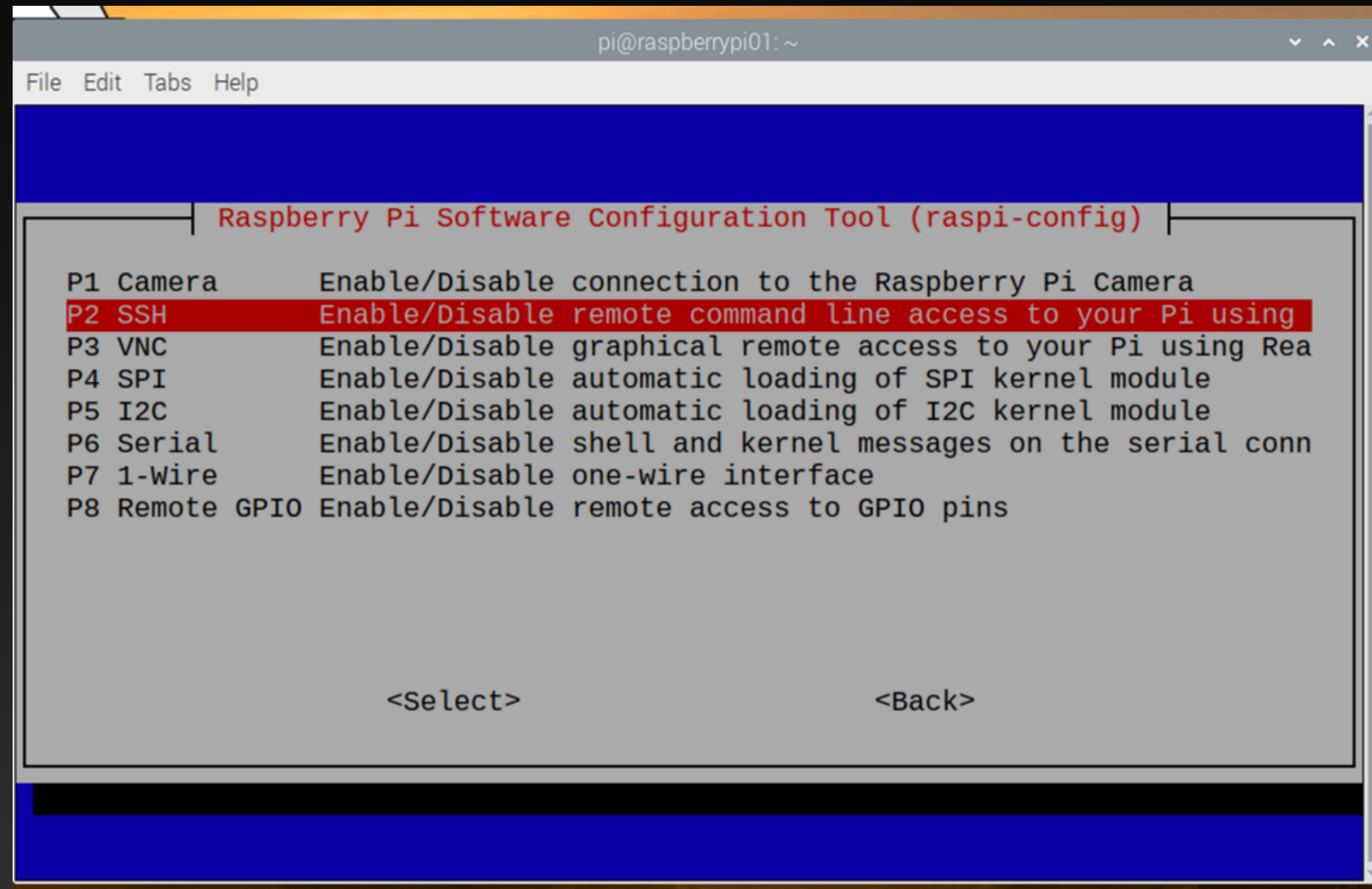
# Remote Access -Setup

- `$ sudo raspi-config`
- Go to Interfacing Options



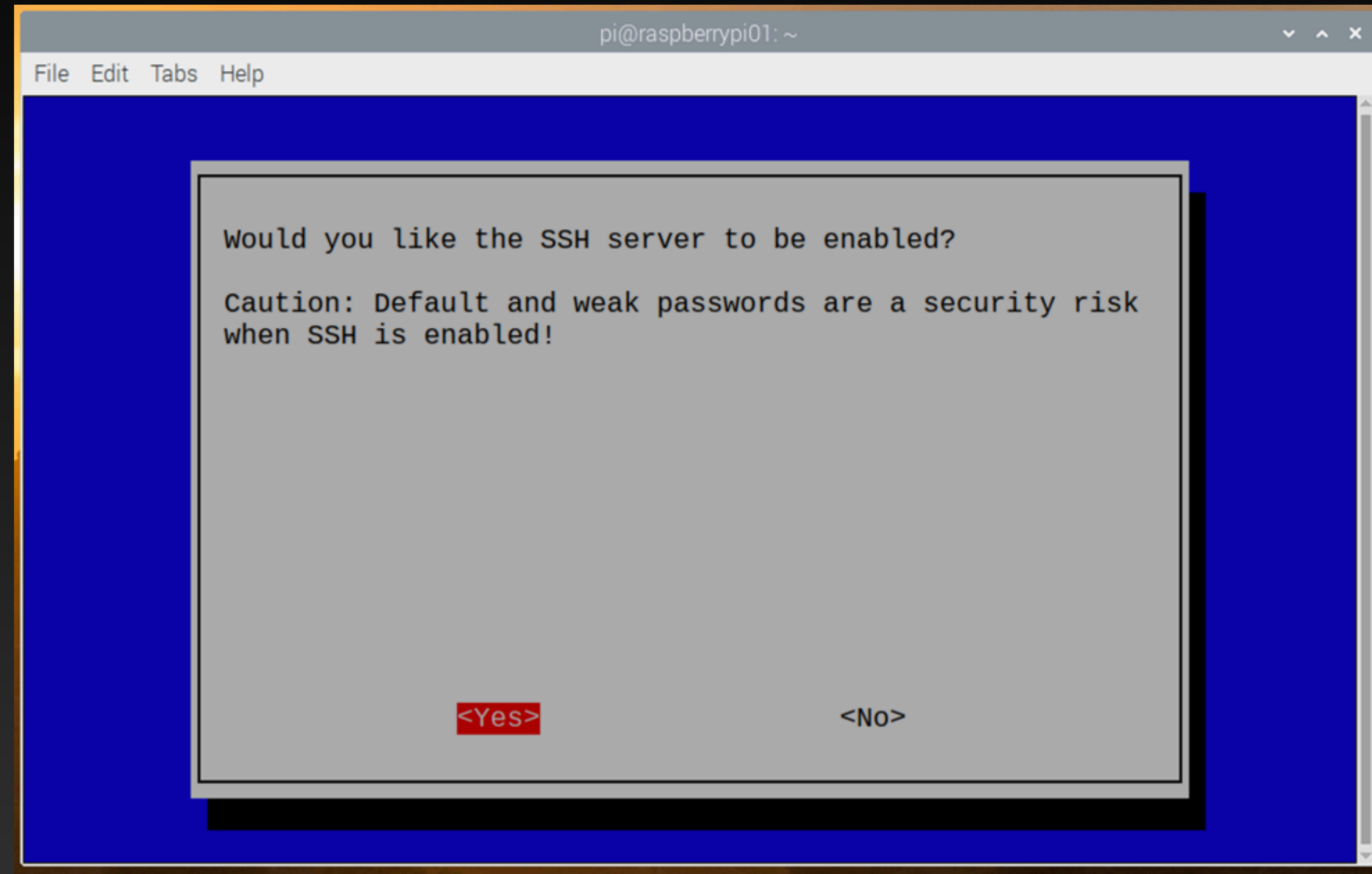
# Remote Access -Setup

- Choose P2 SSH, hit enter



# Remote Access -Setup

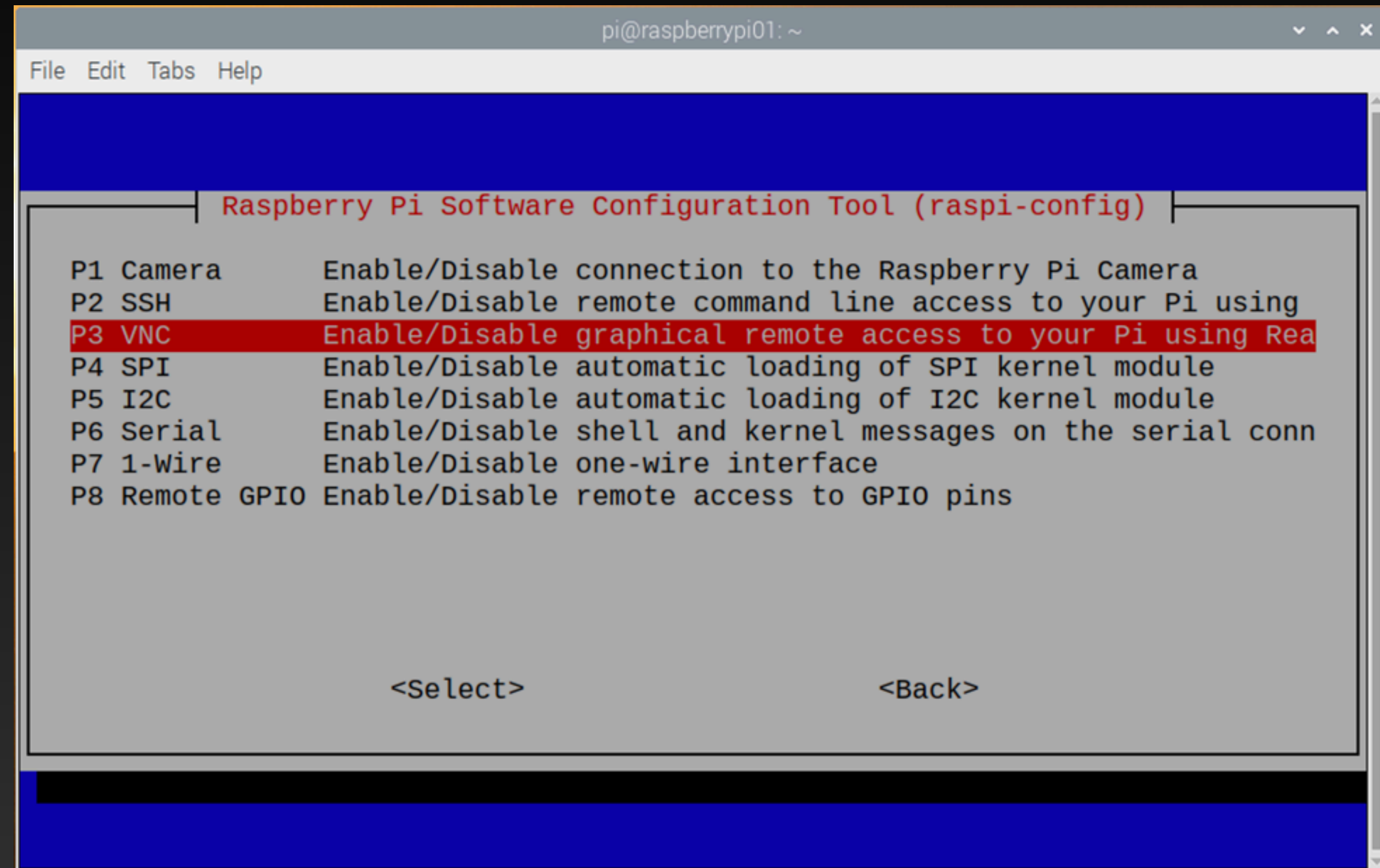
- Select Yes, hit enter





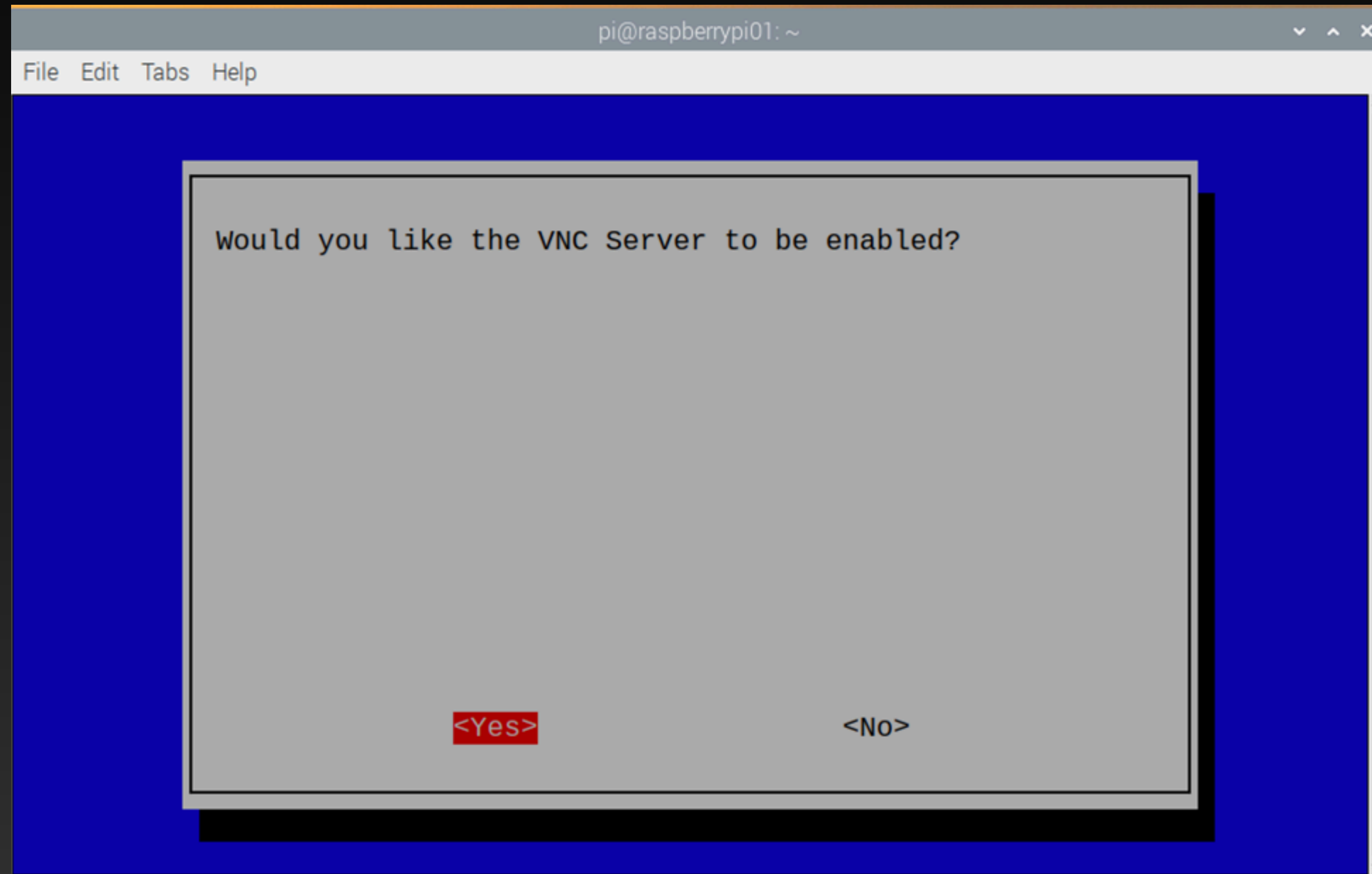
# Remote Access -Setup

- Choose P3 VNC, hit enter



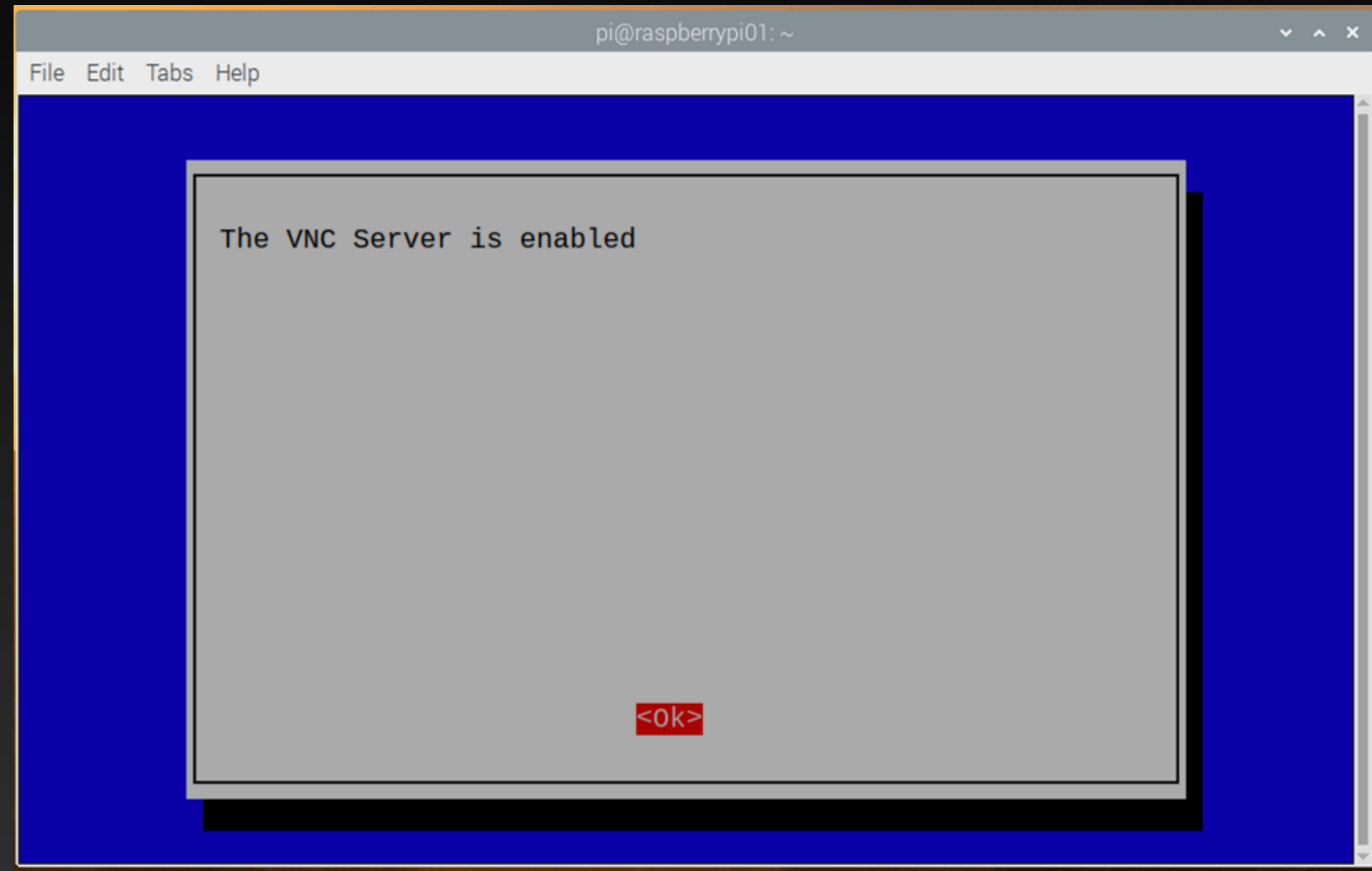
# Remote Access -Setup

- Select Yes, hit enter



# Remote Access -Setup

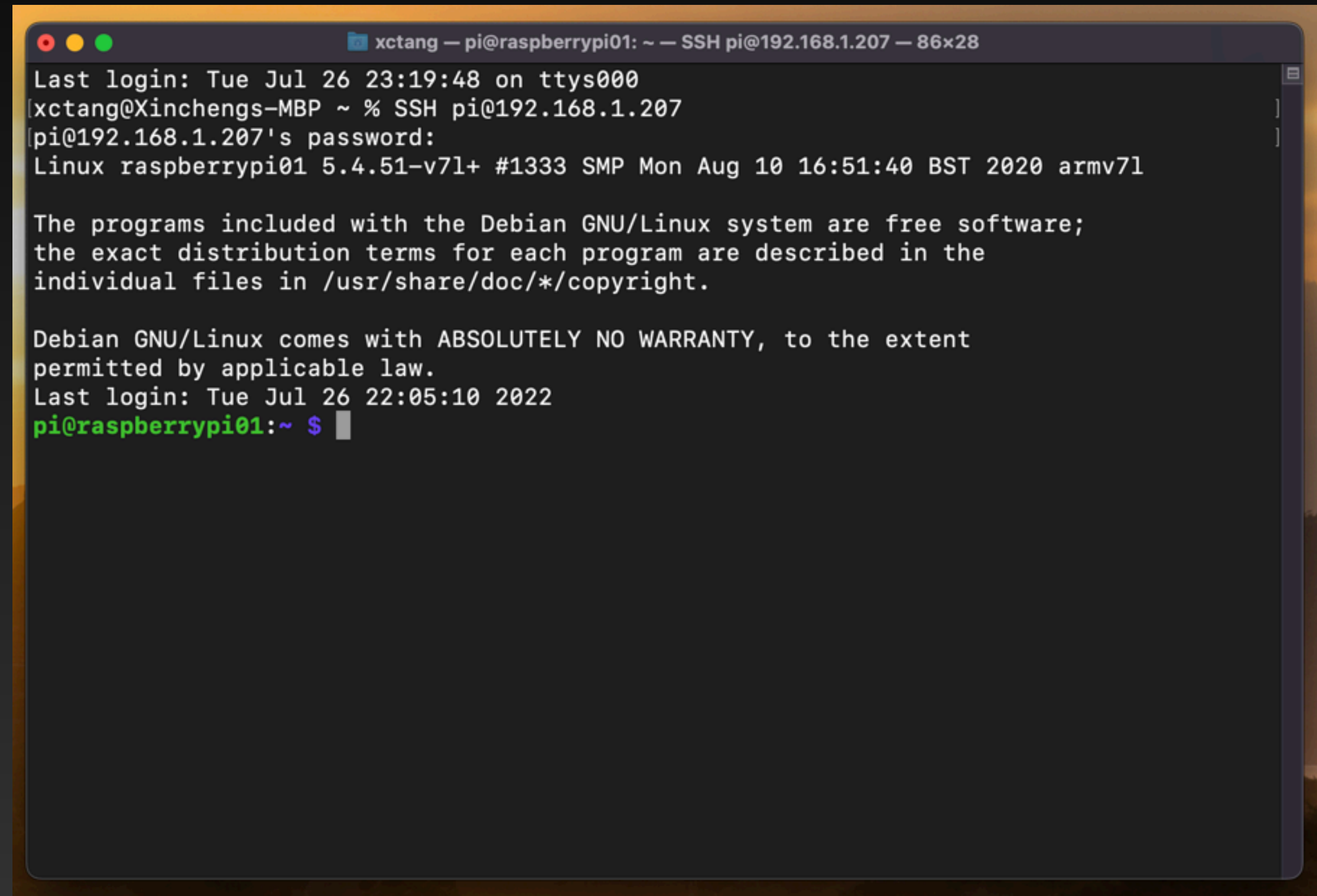
- You will see screen which confirm VNC is enabled





# Remote Access-SSH

- SSH on MAC



```
xctang — pi@raspberrypi01: ~ — SSH pi@192.168.1.207 — 86x28
Last login: Tue Jul 26 23:19:48 on ttys000
xctang@Xinchengs-MBP ~ % SSH pi@192.168.1.207
pi@192.168.1.207's password:
Linux raspberrypi01 5.4.51-v7l+ #1333 SMP Mon Aug 10 16:51:40 BST 2020 armv7l

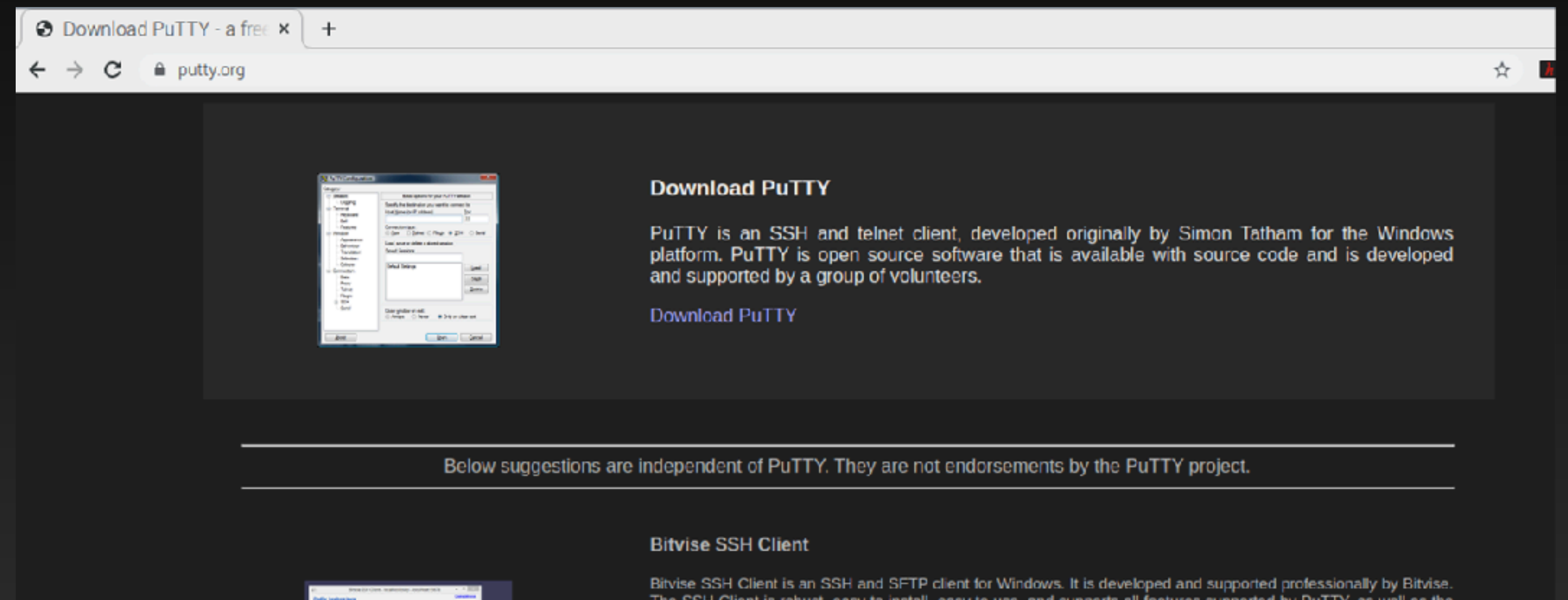
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jul 26 22:05:10 2022
pi@raspberrypi01:~ $
```

# Remote Access-SSH

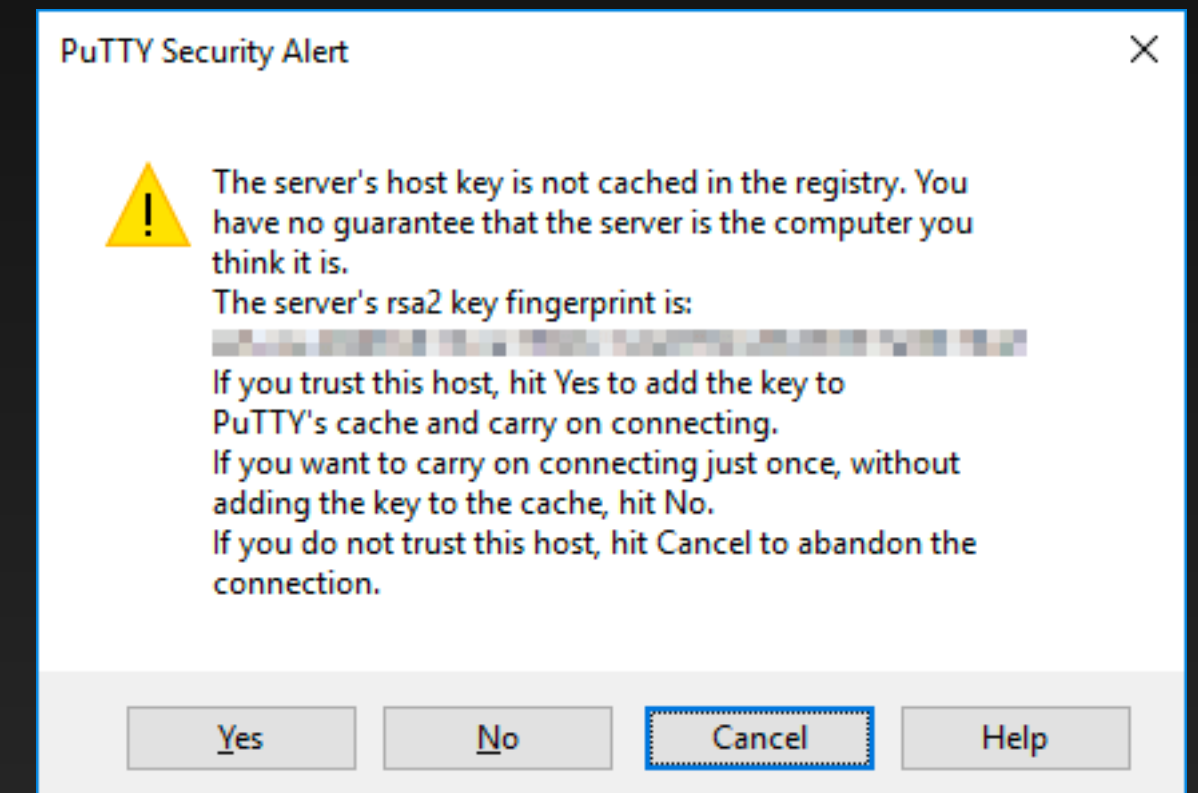
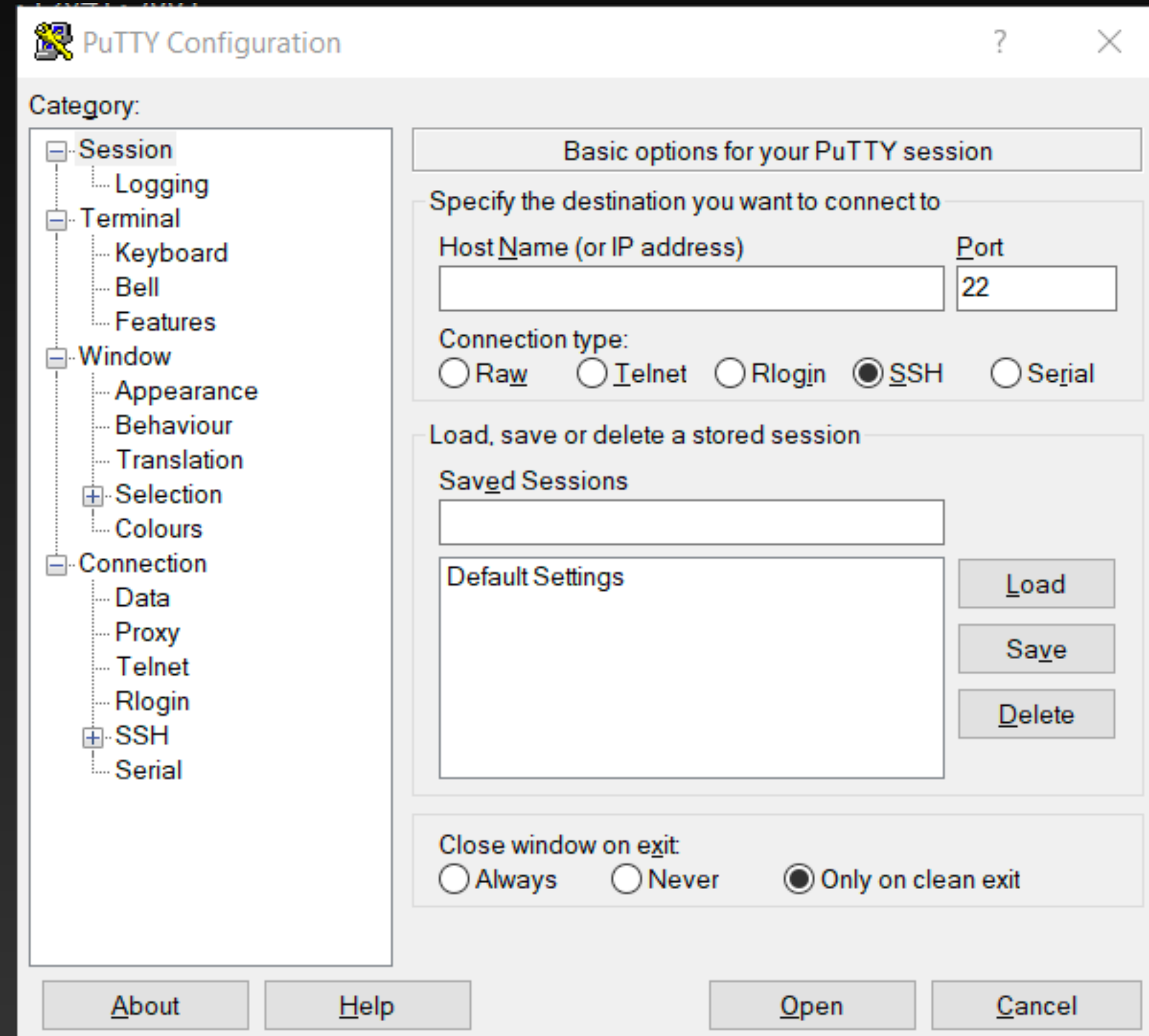
- SSH on Window Machine
- download Putty

<https://www.putty.org/>



# Remote Access-SSH

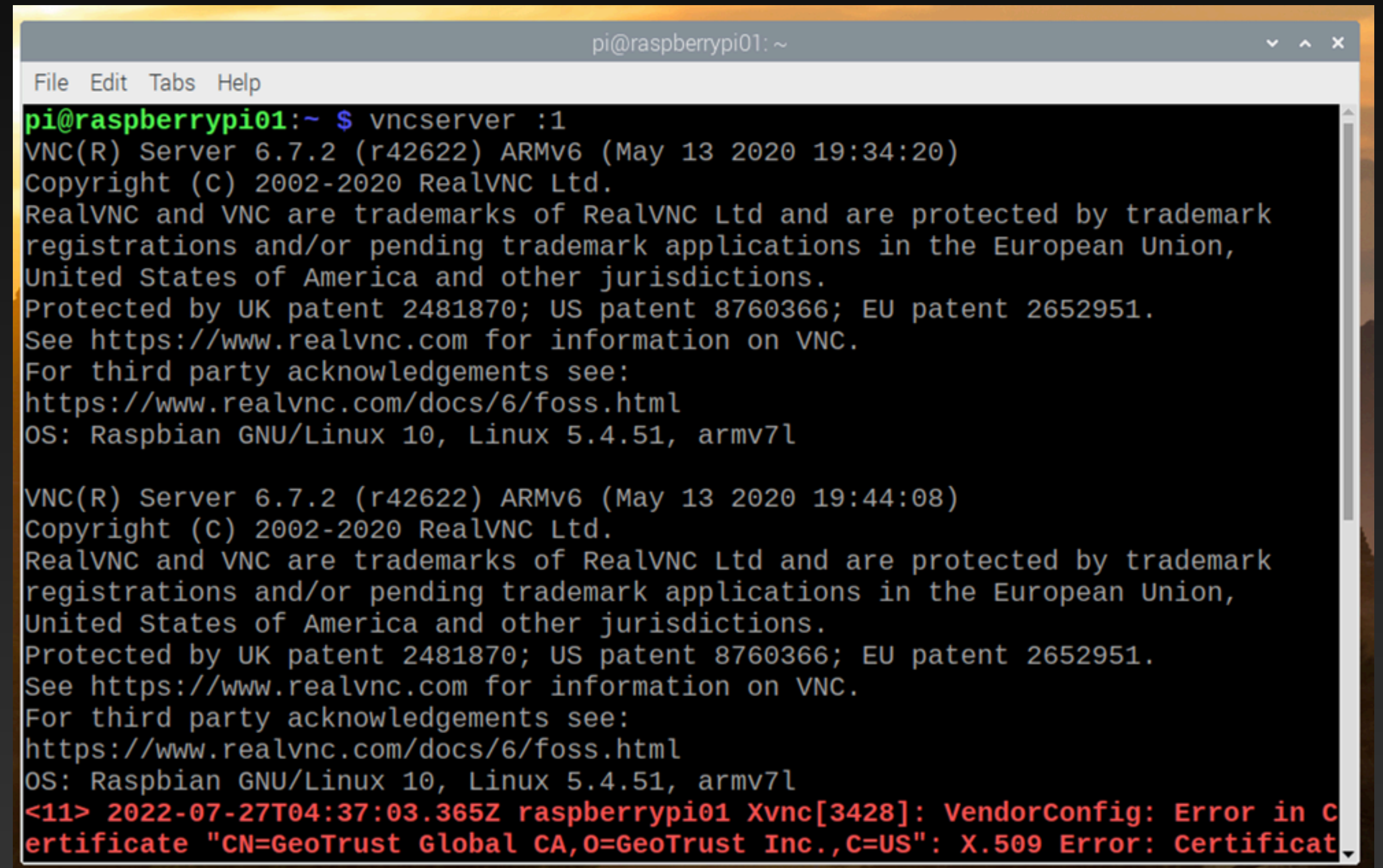
- Putty config screen
- Enter your IP, click open
- You will see security alert, Click Yes to accept
- You will prompt to enter your Pi username and password





# 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

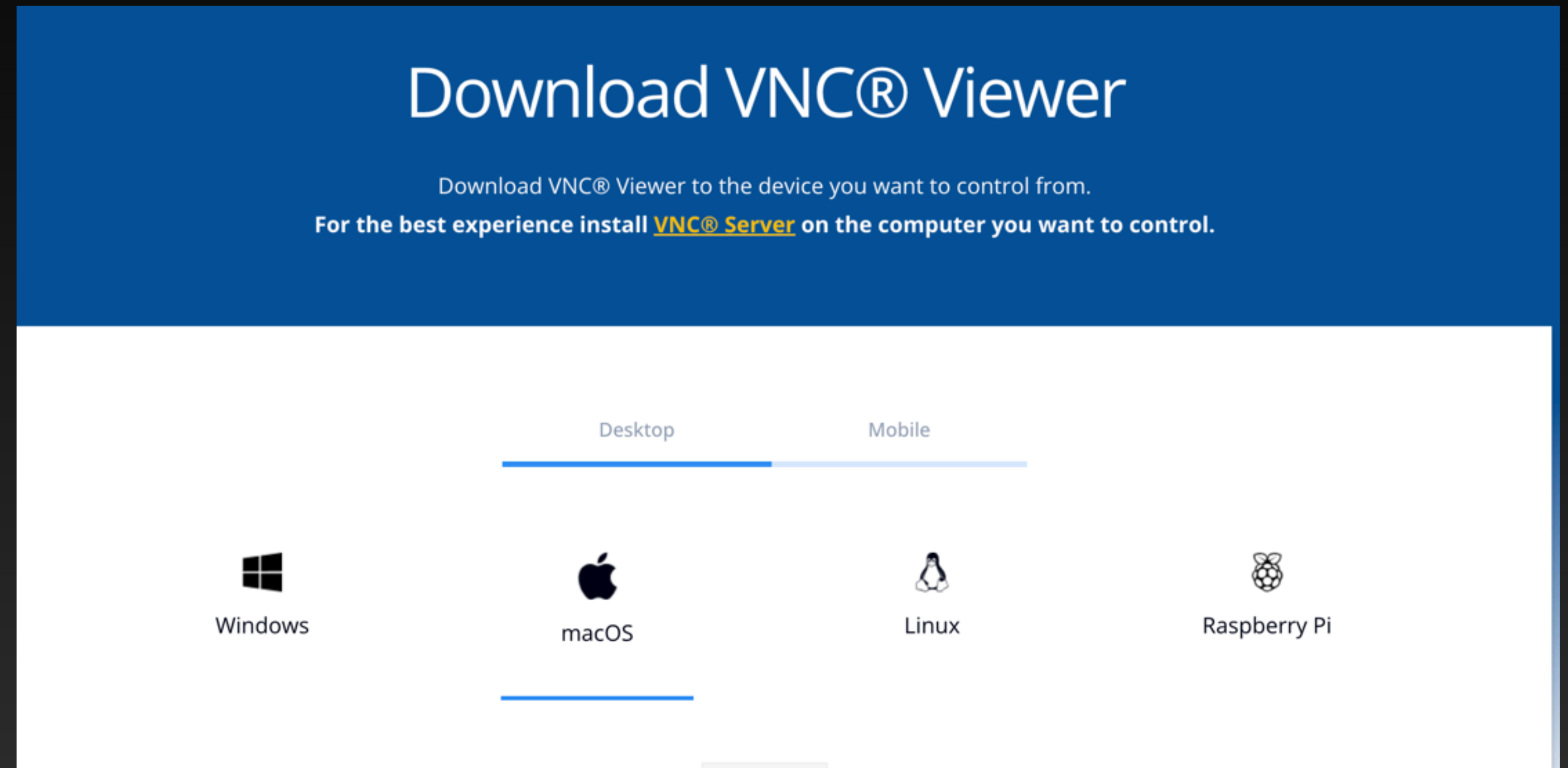


```
pi@raspberrypi01: ~
File Edit Tabs Help
pi@raspberrypi01:~ $ vncserver :1
VNC(R) Server 6.7.2 (r42622) ARMv6 (May 13 2020 19:34:20)
Copyright (C) 2002-2020 RealVNC Ltd.
RealVNC and VNC are trademarks of RealVNC Ltd and are protected by trademark
registrations and/or pending trademark applications in the European Union,
United States of America and other jurisdictions.
Protected by UK patent 2481870; US patent 8760366; EU patent 2652951.
See https://www.realvnc.com for information on VNC.
For third party acknowledgements see:
https://www.realvnc.com/docs/6/foss.html
OS: Raspbian GNU/Linux 10, Linux 5.4.51, armv7l

VNC(R) Server 6.7.2 (r42622) ARMv6 (May 13 2020 19:44:08)
Copyright (C) 2002-2020 RealVNC Ltd.
RealVNC and VNC are trademarks of RealVNC Ltd and are protected by trademark
registrations and/or pending trademark applications in the European Union,
United States of America and other jurisdictions.
Protected by UK patent 2481870; US patent 8760366; EU patent 2652951.
See https://www.realvnc.com for information on VNC.
For third party acknowledgements see:
https://www.realvnc.com/docs/6/foss.html
OS: Raspbian GNU/Linux 10, Linux 5.4.51, armv7l
<11> 2022-07-27T04:37:03.365Z raspberrypi01 Xvnc[3428]: VendorConfig: Error in C
ertificate "CN=GeoTrust Global CA,O=GeoTrust Inc.,C=US": X.509 Error: Certificat
```

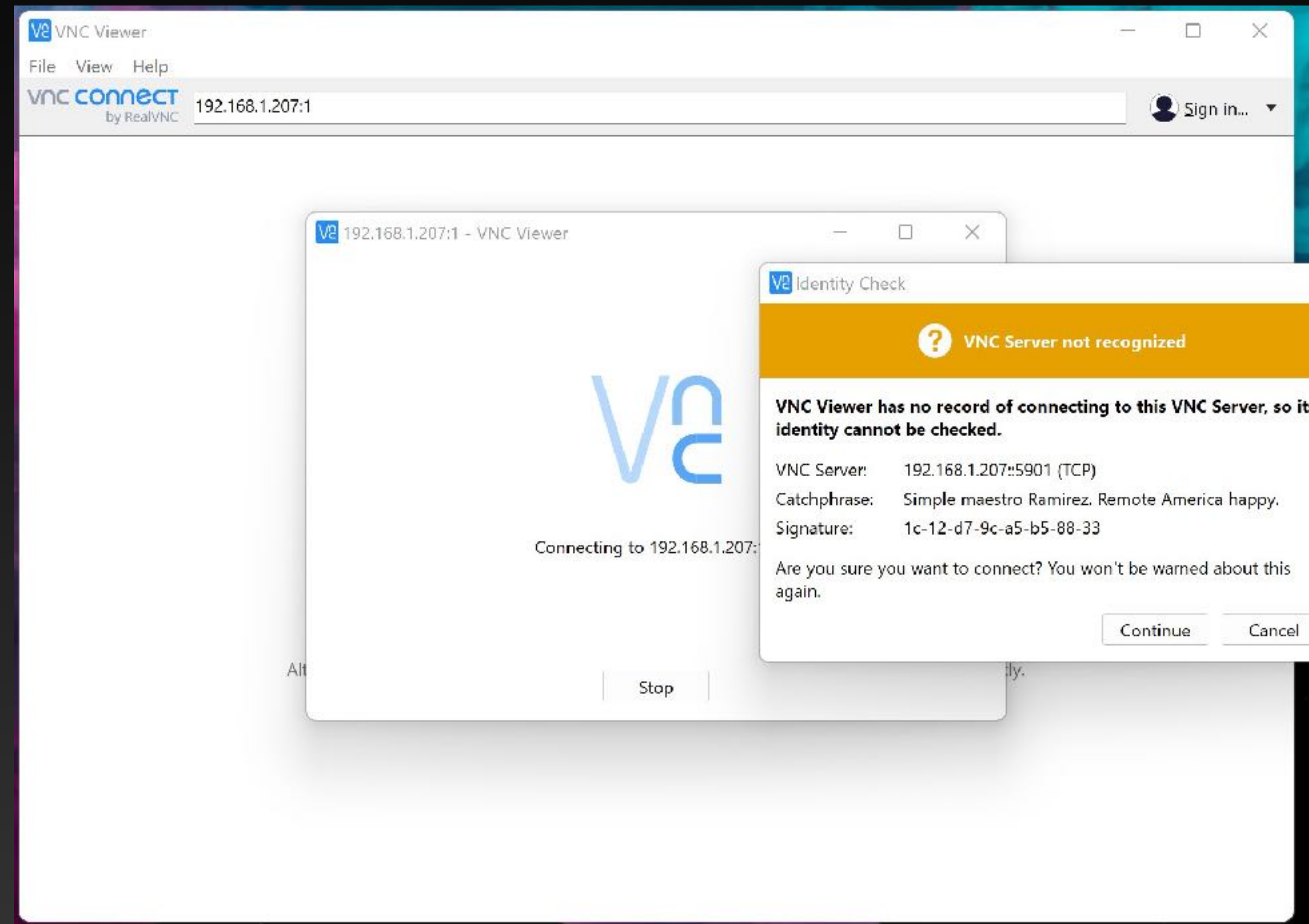
# Remote Access-VNC

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



# Remote Access-VNC

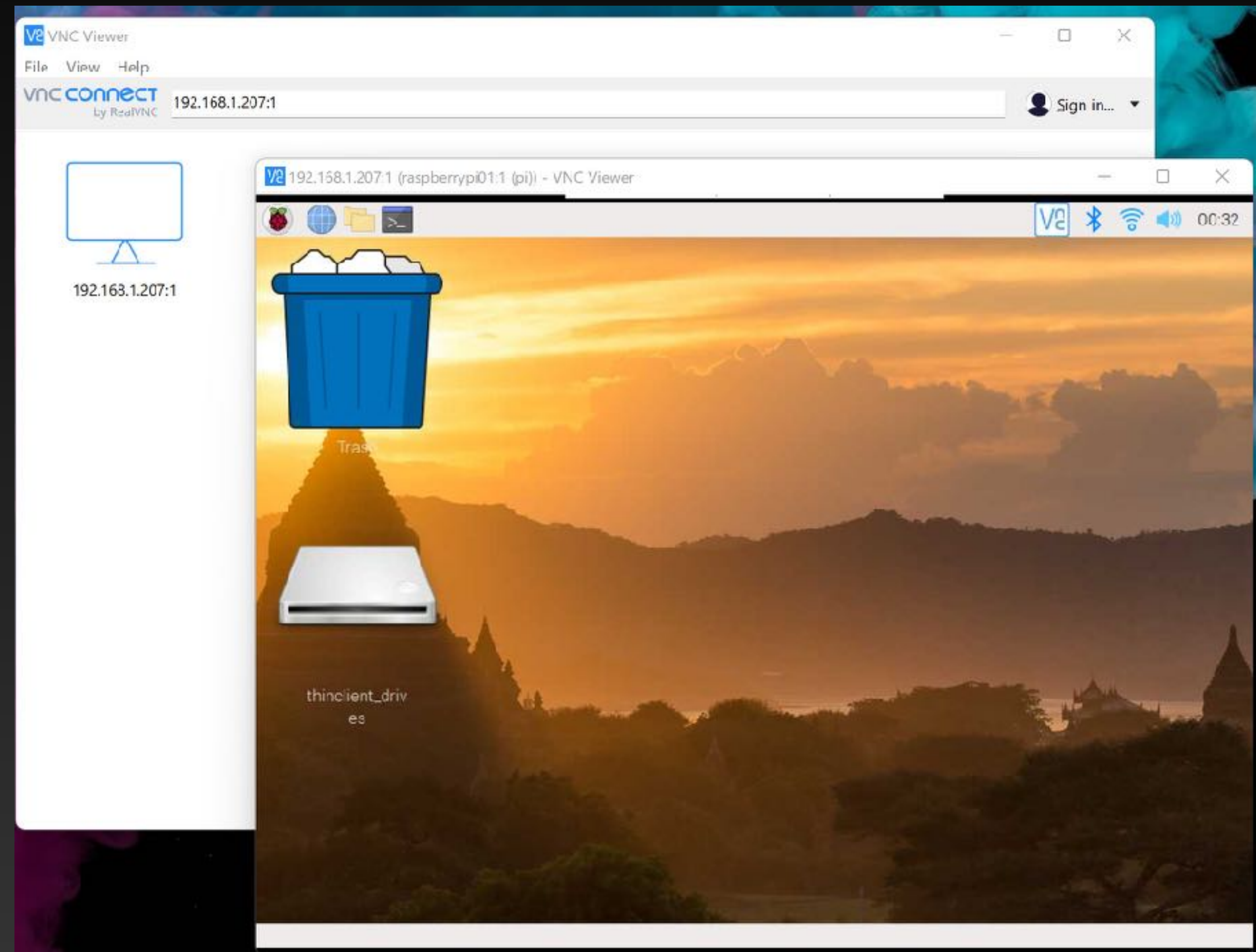
- Put your Pi IP address in the installed VNC viewer





# Remote Access-VNC

- Enter your username and password



# Sending Temperature and Humidity To your android device

Description: In this project, a temperature and humidity sensor module is connected to your Raspberry Pi. The Raspberry Pi sends the ambient temperature and humidity measurements every 10 seconds to the Android mobile phone where they are displayed on the screen.

# Sending Temperature and Humidity To your android device

Hygrothermograph is an important tool in our lives to give us data on the temperature and humidity in our environment. In this project, we will use the RPi to read Temperature and Humidity data of the DHT11 Module.

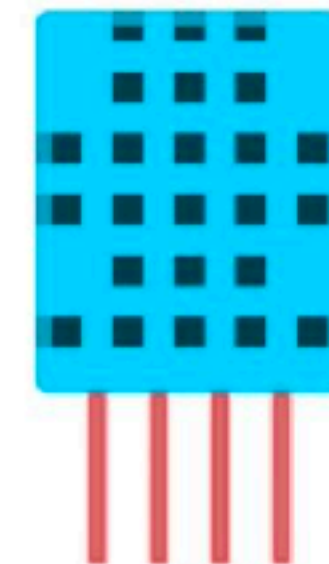
## Component List

Raspberry Pi (with 40 GPIO) x1  
GPIO Expansion Board & Ribbon Cable x1  
Breadboard x1

Jumper Wire x4



DHT11 x1

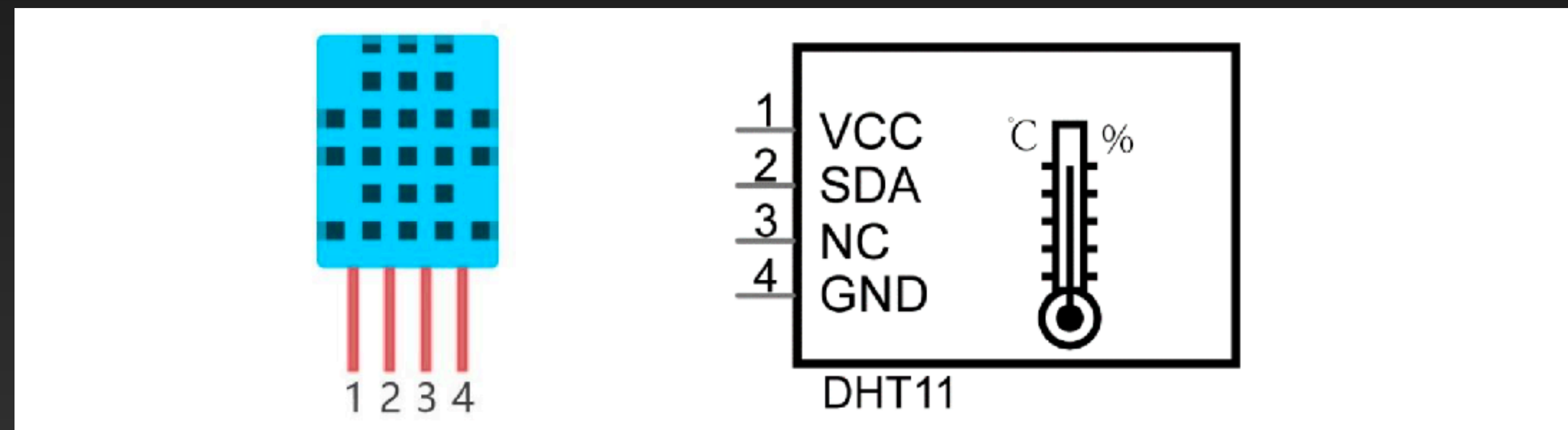


Resistor 10kΩ x1



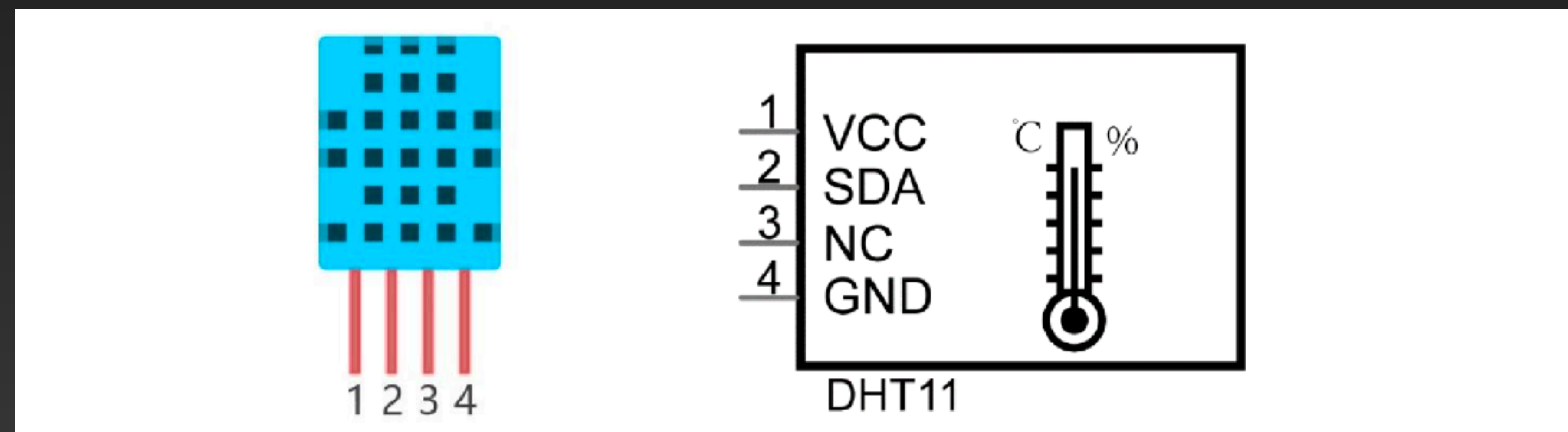
# Sending Temperature and Humidity To your android device

The Temperature & Humidity Sensor DHT11 is a compound temperature & humidity sensor, and the output digital signal has been calibrated by its manufacturer.



# Sending Temperature and Humidity To your android device

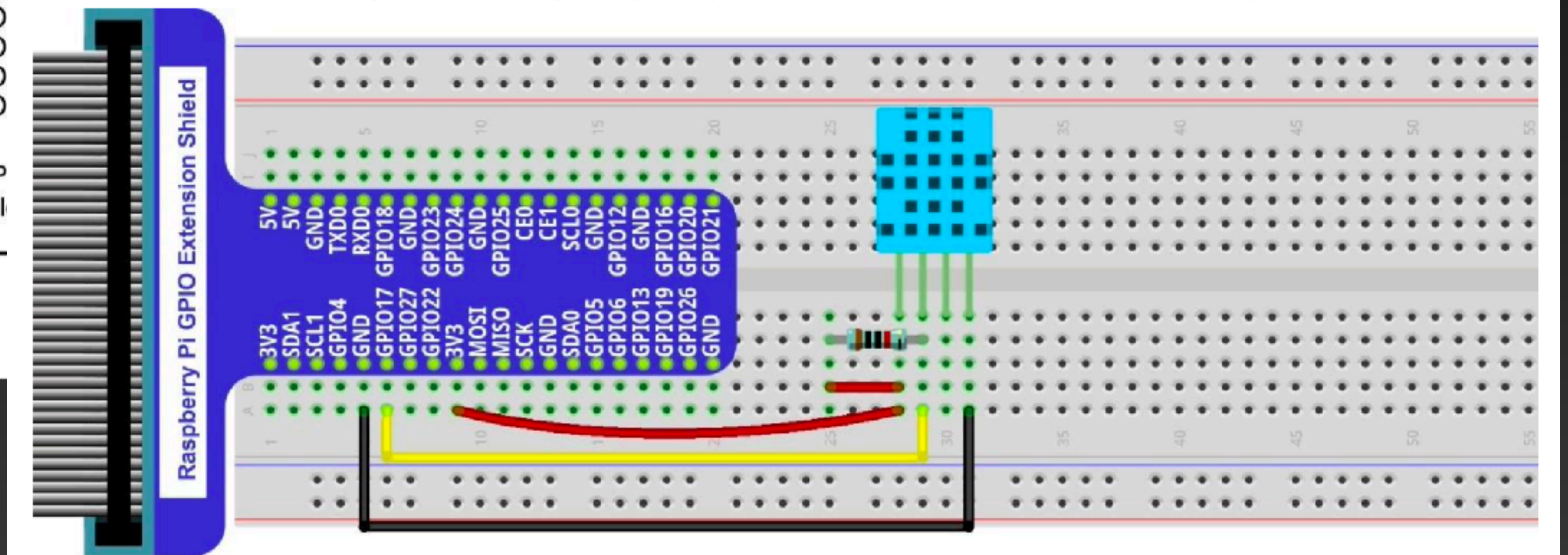
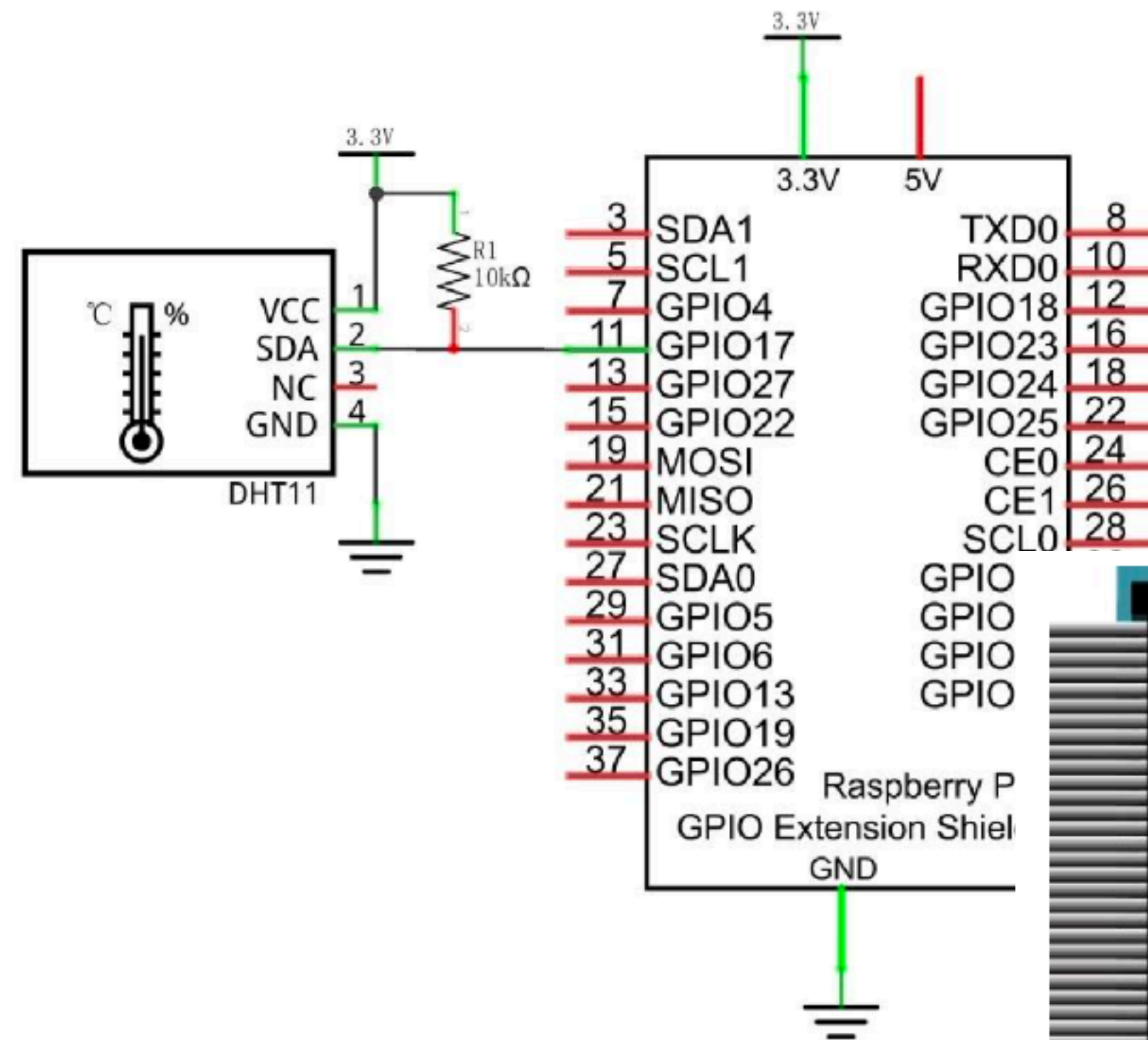
After being powered up, it will initialize in 1 second. Its operating voltage is within the range of 3.3V-5.5V. The SDA pin is a data pin, which is used to communicate with other devices. The NC pin (Not Connected Pin) are a type of pin found on various integrated circuit packages. Those pins have no functional purpose to the outside circuit (but may have an unknown functionality during manufacture and test). Those pins **should not be connected** to any of the circuit connections.





# Sending Temperature and Humidity To your android device

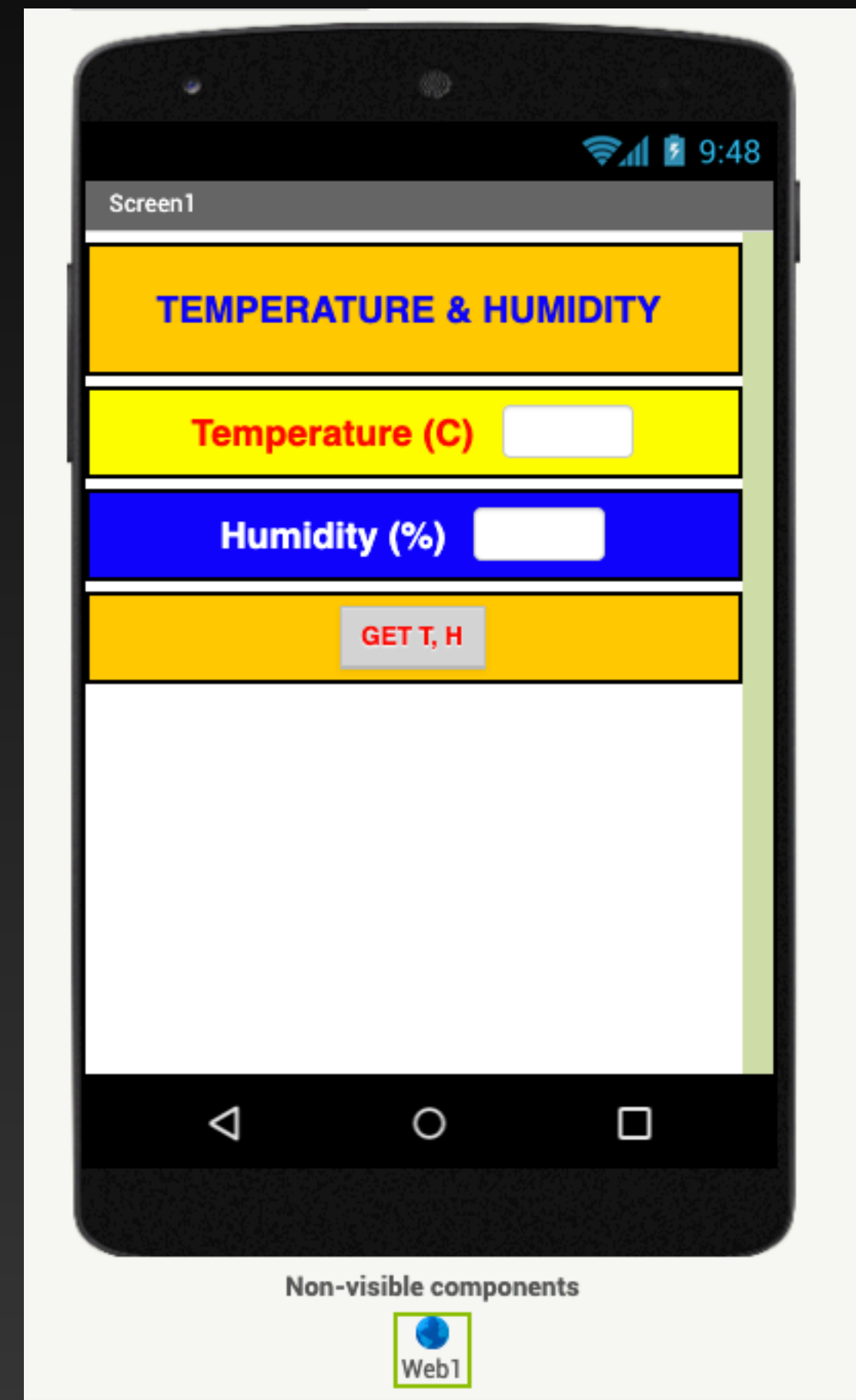
Schematic diagram





# Sending Temperature and Humidity To your android device

Build your MIT App



```
initialize global TH to ""
initialize global RaspberryPi to "http://192.168.1.xxx"

when Button1.Click
do
  set Web1.Url to get global RaspberryPi
  call Web1.Get

when Web1.GotText
  url responseCode responseType responseContent
do
  set global TH to get responseContent
  set TextBox1.Text to segment text get global TH
    start 1
    length 2
  set TextBox2.Text to segment text get global TH
    start 4
    length 2
```

# Sending Temperature and Humidity To your android device

Prepare the environment:

```
sudo pip3 install adafruit-circuitpython-dht
```

There is new updates to the library:

[https://github.com/adafruit/Adafruit\\_CircuitPython\\_DHT](https://github.com/adafruit/Adafruit_CircuitPython_DHT)

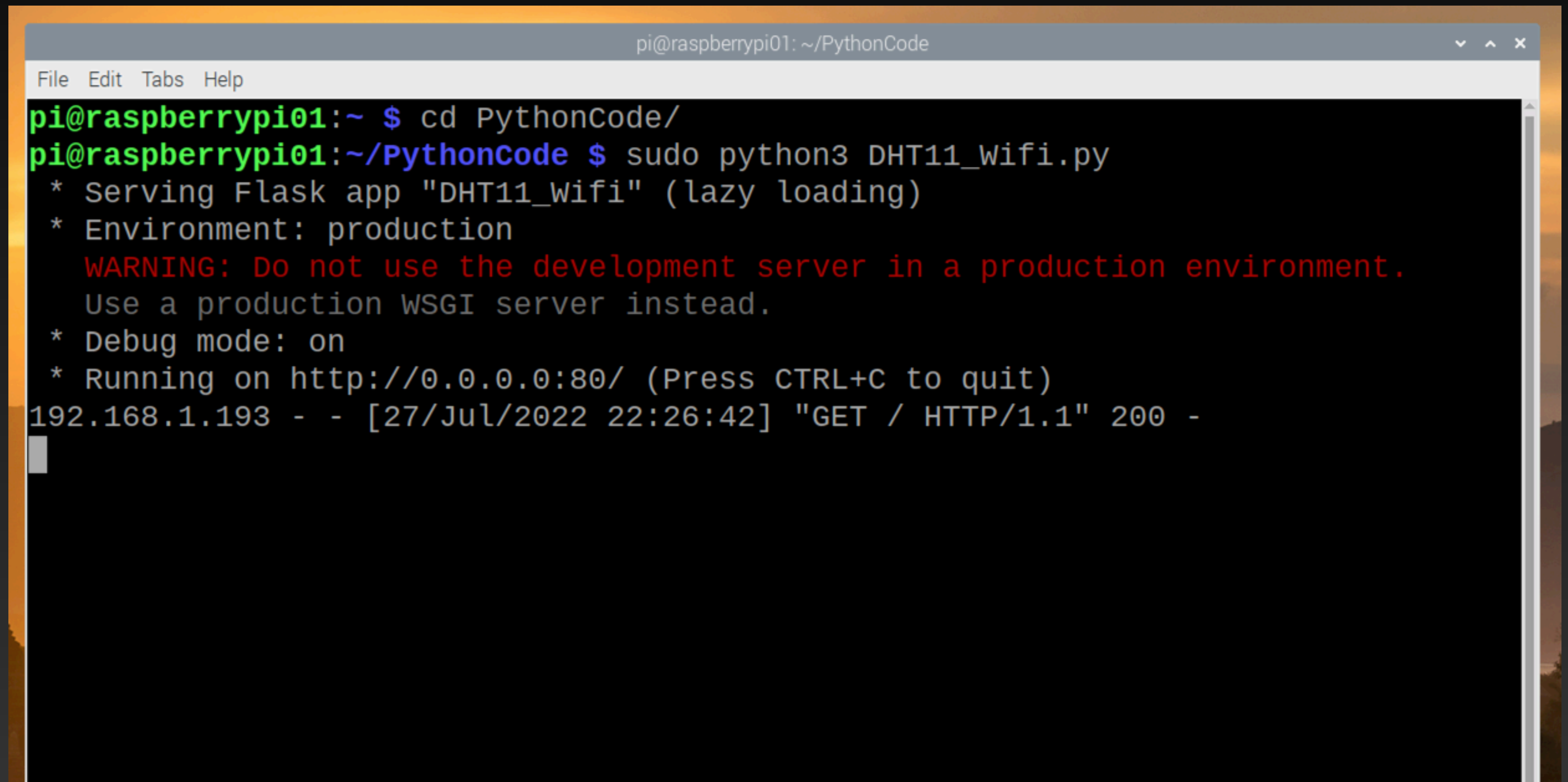
# Sending Temperature and Humidity To your android device

Write the python's  
control program

```
9 import time
10 from flask import Flask, render_template
11 import RPi.GPIO as GPIO
12 import adafruit_dht
13 from board import *
14
15 # GPIO17
16 SENSOR_PIN = D17
17
18 app = Flask(__name__)
19
20 @app.route('/', methods=['GET', 'POST'])
21 def get_data():
22
23     dht11 = adafruit_dht.DHT11(SENSOR_PIN, use_pulseio=False)
24     temp = dht11.temperature
25     hum = dht11.humidity
26     tempint = int(temp)
27     humint = int(hum)
28     if tempint < 10:
29         datat = "0" + str(tempint)
30     else:
31         datat = str(tempint)
32     datath = datat + "," + str(humint)
33     return(datath)
34
35 if __name__ == '__main__':
36     app.run(debug=True, port=80, host='0.0.0.0', use_reloader=False)
37
38
```

# Sending Temperature and Humidity To your android device

Run the program in  
terminal

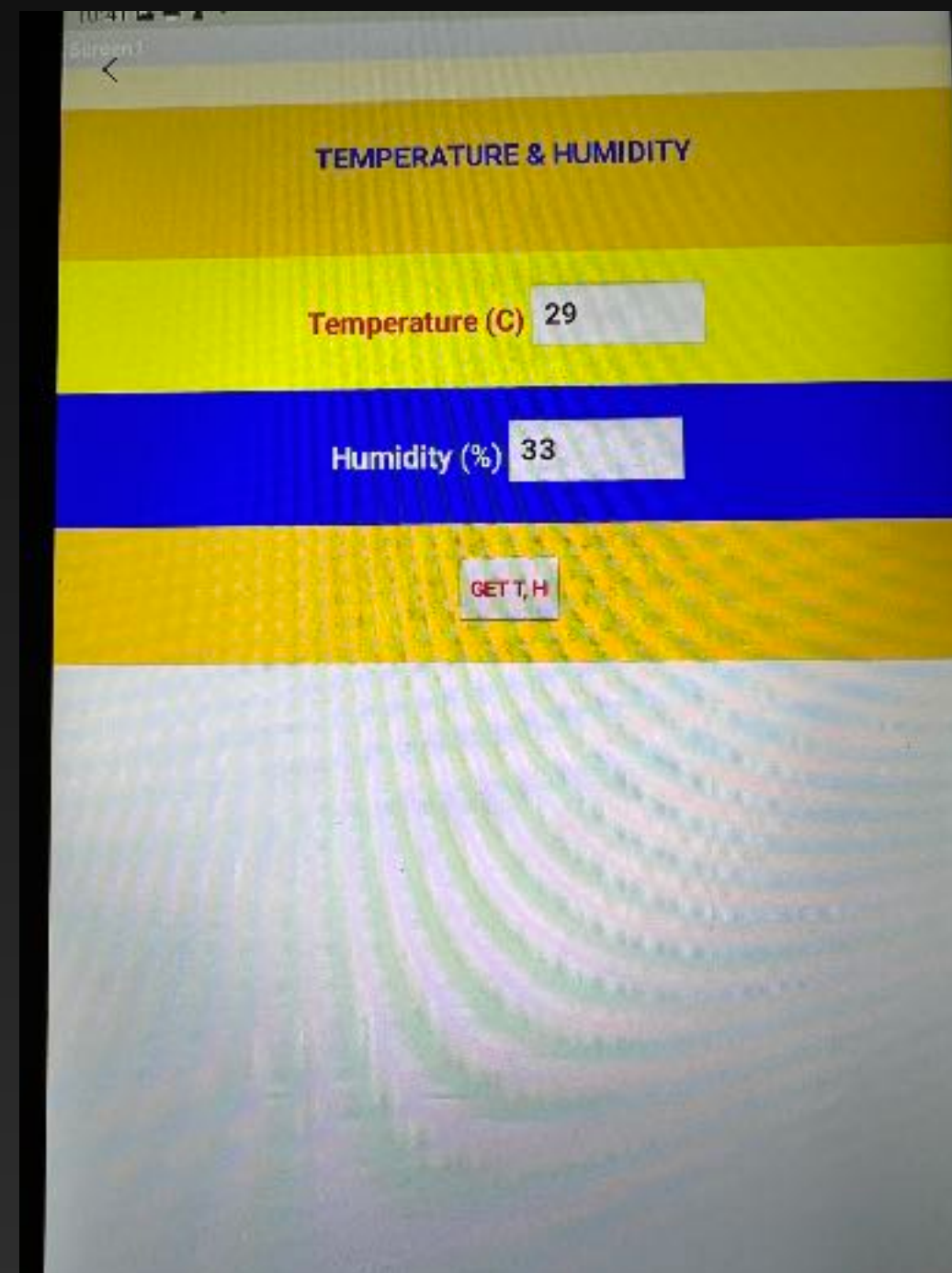


```
pi@raspberrypi01: ~/PythonCode
File Edit Tabs Help
pi@raspberrypi01:~ $ cd PythonCode/
pi@raspberrypi01:~/PythonCode $ sudo python3 DHT11_Wifi.py
* Serving Flask app "DHT11_Wifi" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
192.168.1.193 - - [27/Jul/2022 22:26:42] "GET / HTTP/1.1" 200 -
```



# Sending Temperature and Humidity To your android device

When run the APP program, will see the web server get with 200 code indicates successful connection.



```
pi@raspberrypi01: ~/PythonCode
File Edit Tabs Help
pi@raspberrypi01:~ $ cd PythonCode/
pi@raspberrypi01:~/PythonCode $ sudo python3 DHT11_Wifi.py
* Serving Flask app "DHT11_Wifi" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
192.168.1.193 - - [27/Jul/2022 22:26:42] "GET / HTTP/1.1" 200 -
```

# Homework

Finish class project to setup Temperature and Humidity sensor, and post questions you have encounter