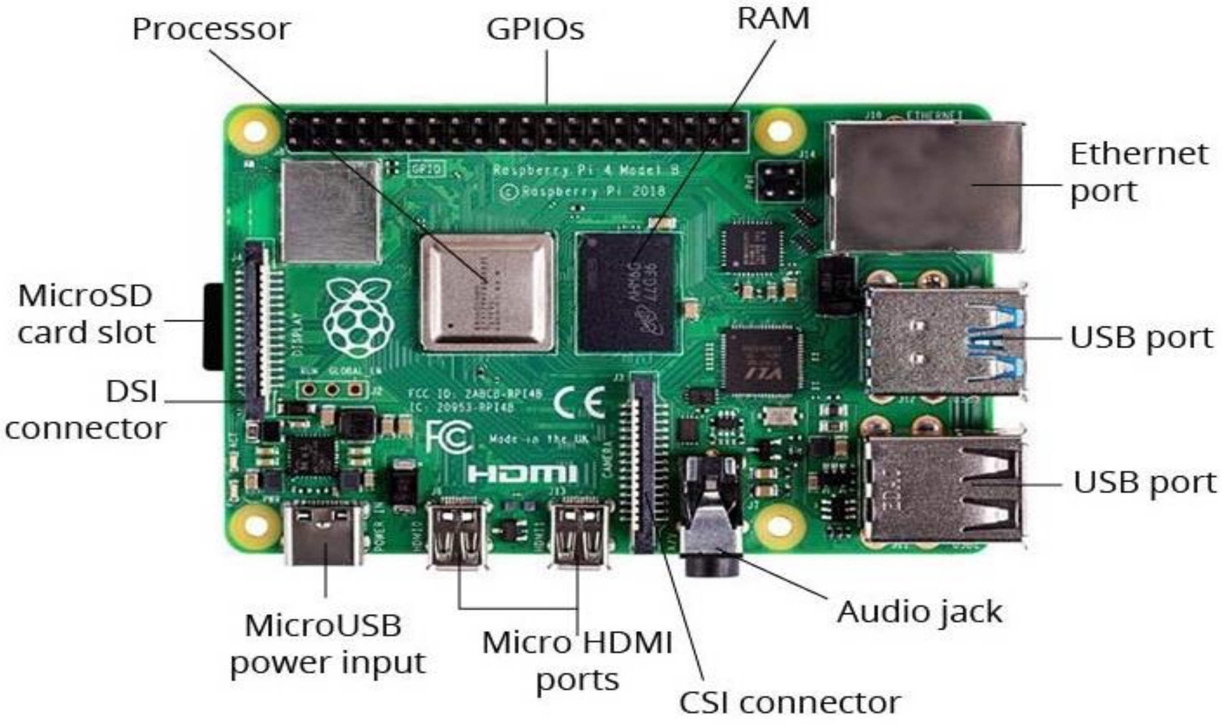
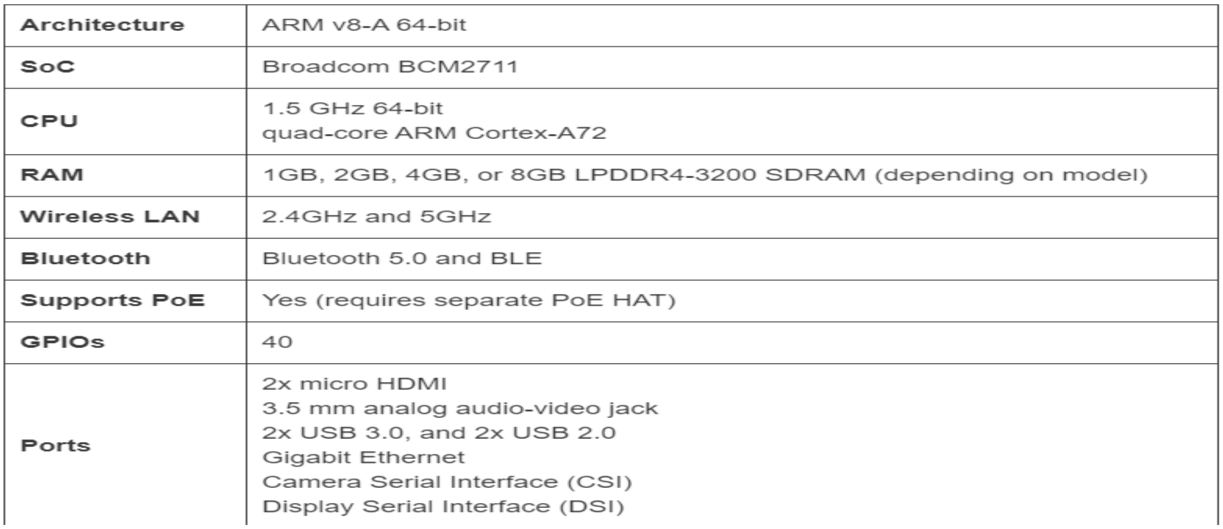
1. Introducing the Raspberry Pi

Raspberry Pi ek credit card size jitna ek computer hai. Ise united state me develop kiya gaya tha taki school me bacho ko basic computer science pdhaya ja sake kyuki us samay sabke pass laptop ya computer lene ke paise nahi hote the. Ise raspberry pi foundation ke through banaya gaya tha. 2012 se 2022 tak lagbhag 16 million raspberry pi sold ho chuke hai

RASPBERRY PI 4 Features





Architecture	ARM v8-A 64-bit
SoC	Broadcom BCM2711
CPU	1.5 GHz 64-bit quad-core ARM Cortex-A72
RAM	1GB, 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM (depending on model)
Wireless LAN	2.4GHz and 5GHz
Bluetooth	Bluetooth 5.0 and BLE
Supports PoE	Yes (requires separate PoE HAT)
GPIOs	40
Ports	2x micro HDMI 3.5 mm analog audio-video jack 2x USB 3.0, and 2x USB 2.0 Gigabit Ethernet Camera Serial Interface (CSI) Display Serial Interface (DSI)

Here’s a glance at what each component of the Raspberry Pi does:

- **USB ports**: to connect a mouse, a keyboard, or other peripherals. It comes with two USB 3.0 and two USB 2.0 ports;
- **Ethernet port**: to connect to the internet using an Ethernet cable;
- **Audio jack**: to connect an audio device;
- **CSI connector**: to connect a camera with a CSI ribbon;
- **HDMI connector**: to connect a monitor or TV;
- **Processor**: is the brain of the Raspberry Pi;
- **MicroSD card slot**: to insert a microSD card to store your files and your operating system;
- **MicroUSB power input**: to power up your Pi;
- **DSI connector**: to connect DSI-compatible displays;
- **Antenna**: picks up wireless LAN and Bluetooth signals;
- **GPIOs** (general purpose input output pins): connect devices to interact with the outside world like sensors and outputs like LEDs and motors

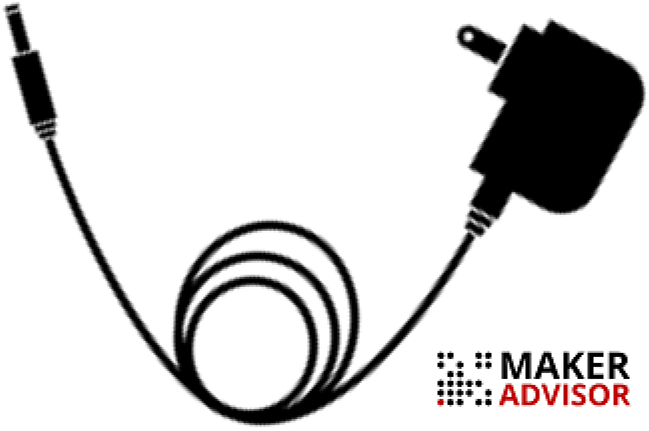
Raspberry Pi (Desktop Computer): raspberry pi ko hum kuch tarah se ek desktop computer ki tarah dekh sakte hai kyuki isme processor hota hai, RAM hota hai, USB ports hota hai keyboard aur mouse ko plugin karne ke liye aur hdmi port bhi hote hai taki hum display or TV ko connect kar sake. Aur to aur hum ise isme internet bhi connect kar sakte hai. use karna chahte hai to humare pass kya kya chiz pahle se honi chahiye.Tum isme wo sari chize lagbhag kar sakte ho jo ek regular computer me karte hai jaise web browsing, document editing, playing games, coding aur bhi bhut kuch. Niche ek photo me aap raspberry pi 3 ko ek desktop computer ki tarah setup kiya gaya hai.

STEPS TO SETUP RASPBERRY PI

1. MicroSD Card (with card reader so that we can connect this sd card to our computer)
2. Installing Raspberry Pi OS
3. Connecting the Raspberry Pi to the Internet
4. Enabling SSH
5. Connecting via SSH to the Raspberry Pi

Accessories You Need To Get Started

When you buy a Raspberry Pi board, you only get a bare electronic board that doesn’t do much on its own. You need several accessories to get started. There are a lot of accessories for the Raspberry Pi, but you need at least a microSD card and a power supply. Without these accessories your Raspberry Pi is useless.



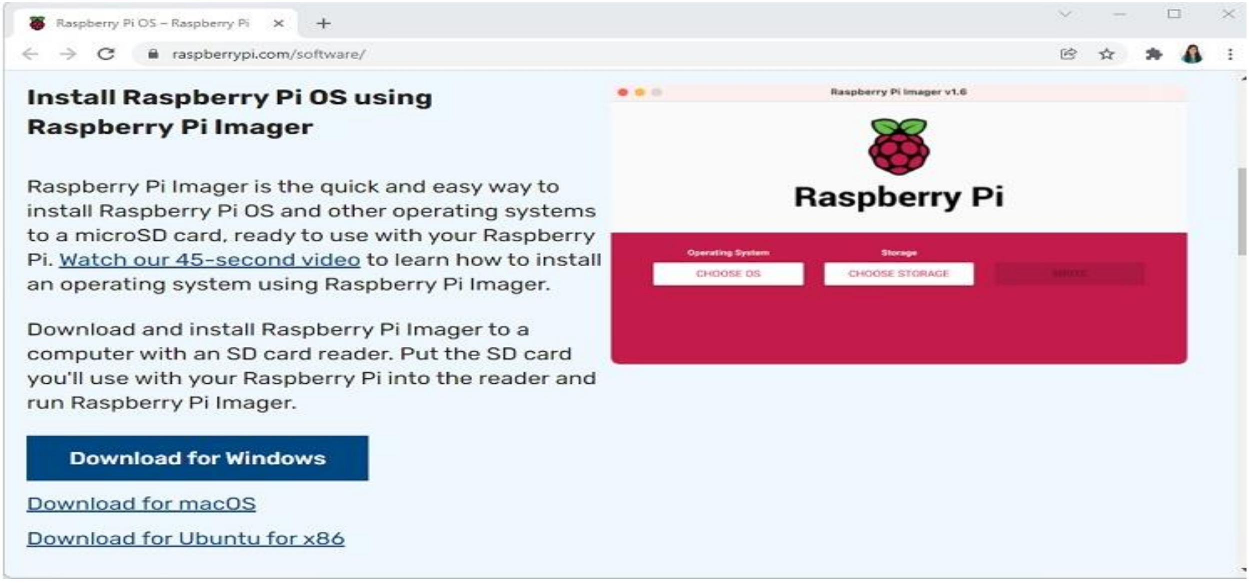
- **Power supply**: you need a power adapter that provides 5V 2.5A
- **MicroSD card**: we recommend getting a microSD card with at least 16GB, class 10. You need a microSD card to store your files and the Pi’s operating system. The Pi doesn’t have a hard drive*, so everything you do on your Pi is saved on the microSD card, even the operating system. You can get a microSD card with the operating system preloaded or install the operating system yourself (which we recommend).

Getting a Raspberry Pi Starter Kit

Installing the Operating System

There are several operating systems suitable for the Pi. The official distribution for the Raspberry Pi is Raspberry Pi OS and that's the one we recommend you install (specially if you're a beginner, because it is the most supported).

1) Start by connecting the microSD card to your computer.
2) Go to the [Raspberry Pi Software page](https://www.raspberrypi.com/software/).
3) Select and download the Raspberry Pi Imager (a tool to flash the OS on the microSD card) for your computer's operating system.



Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our 45-second video to learn how to install an operating system using Raspberry Pi Imager.

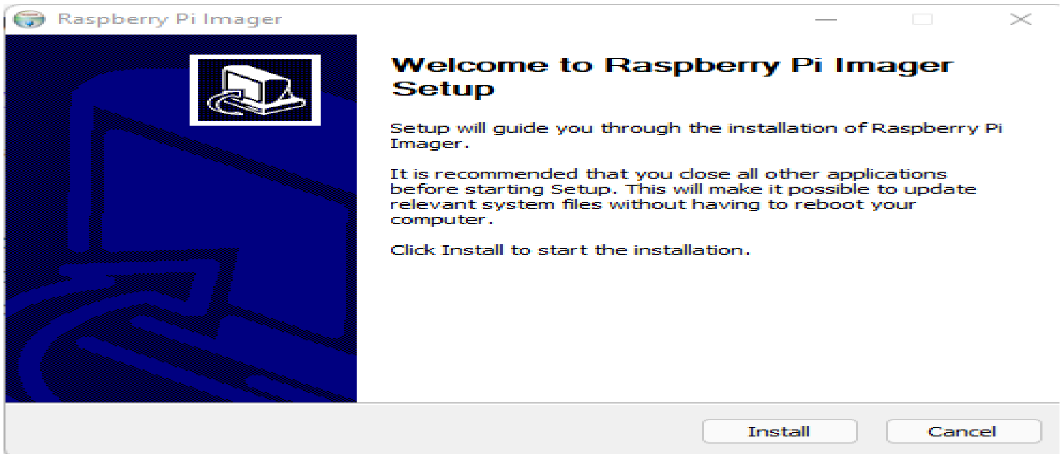
Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

Download for Windows

Download for macOS

Download for Ubuntu for x86

4) Click on the downloaded file to install the Raspberry Pi Imager.



Raspberry Pi Imager

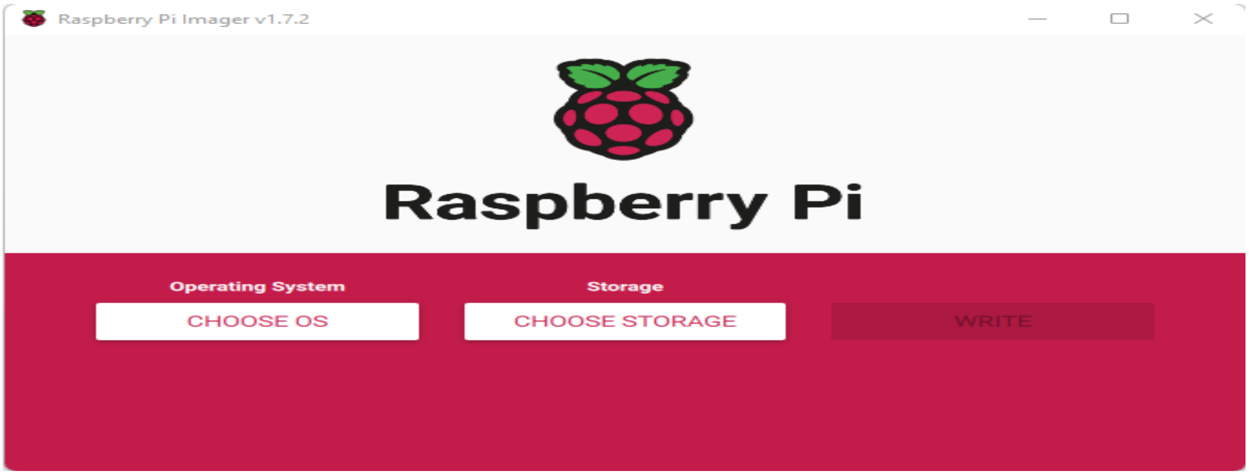
Welcome to Raspberry Pi Imager Setup

Setup will guide you through the installation of Raspberry Pi Imager.

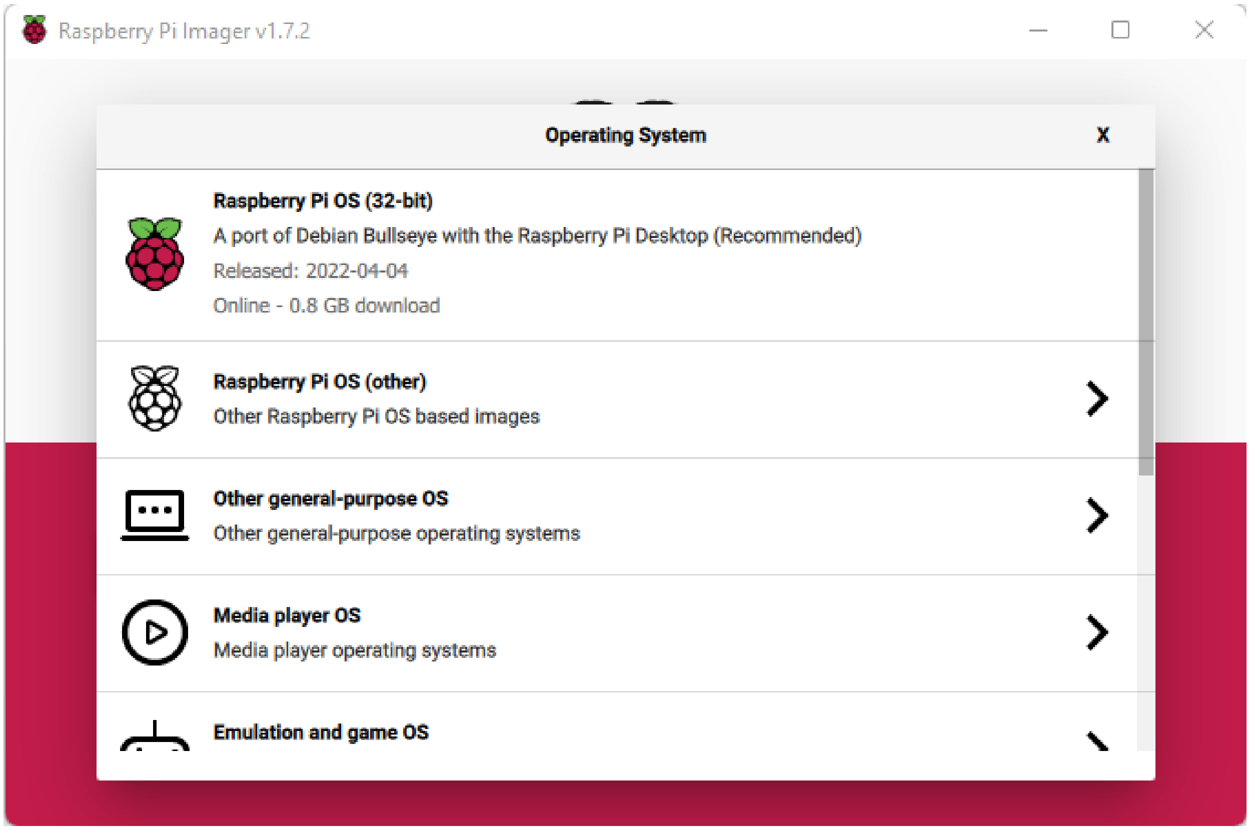
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Install to start the installation.

5) When the installation is complete, the Raspberry Pi Imager will open.



6) Click on **Choose OS** to select the Operating System. Select the **Raspberry Pi OS**, or the **Raspberry Pi OS 64-bit** under the **Raspberry Pi OS (other)** option.



Raspberry Pi Imager v1.7.2

Operating System

Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Released: 2022-04-04
Online - 0.8 GB download

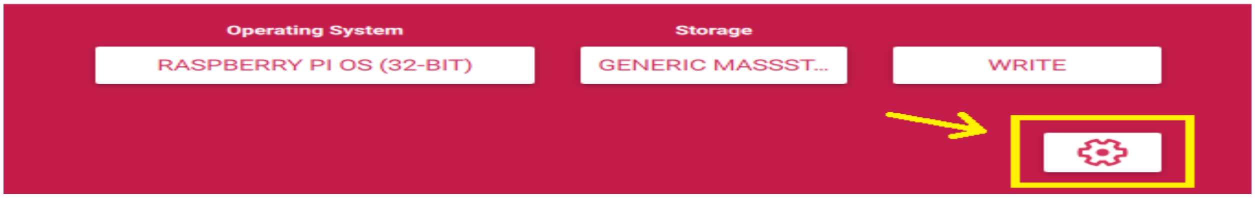
Raspberry Pi OS (other)
Other Raspberry Pi OS based images

Other general-purpose OS
Other general-purpose operating systems

Media player OS
Media player operating systems

Emulation and game OS

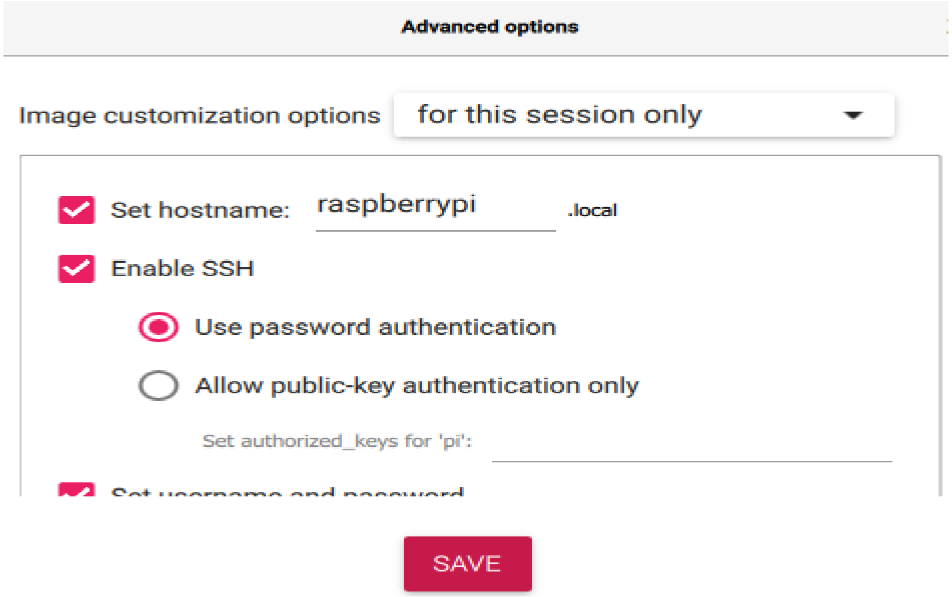
8) The Raspberry Pi Imager allows you to access advanced settings to configure hostname, SSH, Wi-Fi, among others—click on the gear icon to open advanced settings. If your window doesn't show the gear icon, press **Ctrl–Shift–X** to open the advanced setting window.



Operating System: RASPBERRY PI OS (32-BIT)

Storage: GENERIC MASSST...

9) You can set hostname (the default will be `raspberrypi`), enable SSH, and set a password for SSH connection. If you want to use the Raspberry Pi as a desktop computer, you don't need to enable SSH.



Advanced options

Image customization options: for this session only

☑ Set hostname: raspberrypi .local

☑ Enable SSH

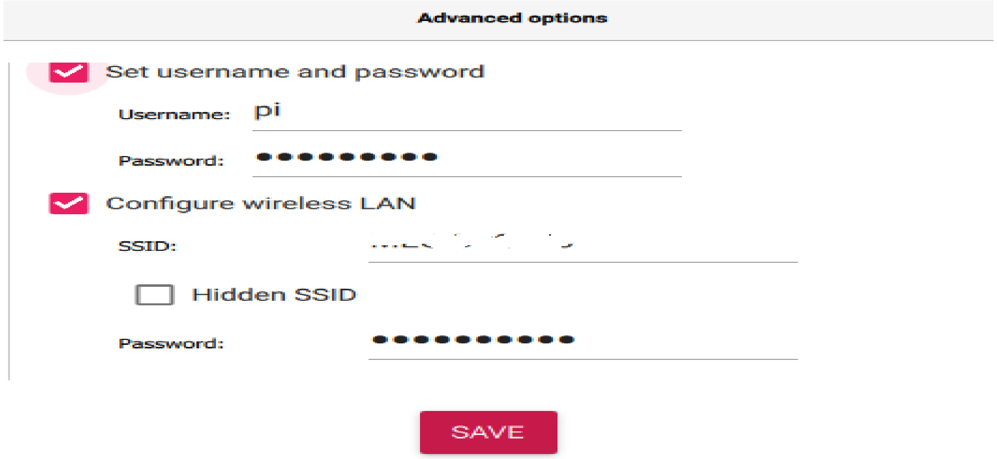
● Use password authentication

○ Allow public-key authentication only

Set authorized_keys for 'pi':

☑ Set username and password

10) Additionally, set up Wi-Fi with your local network credentials, so that you can connect to your Raspberry Pi using Wi-Fi later on.



Advanced options

☑ Set username and password

Username: pi

Password: ••••••••••

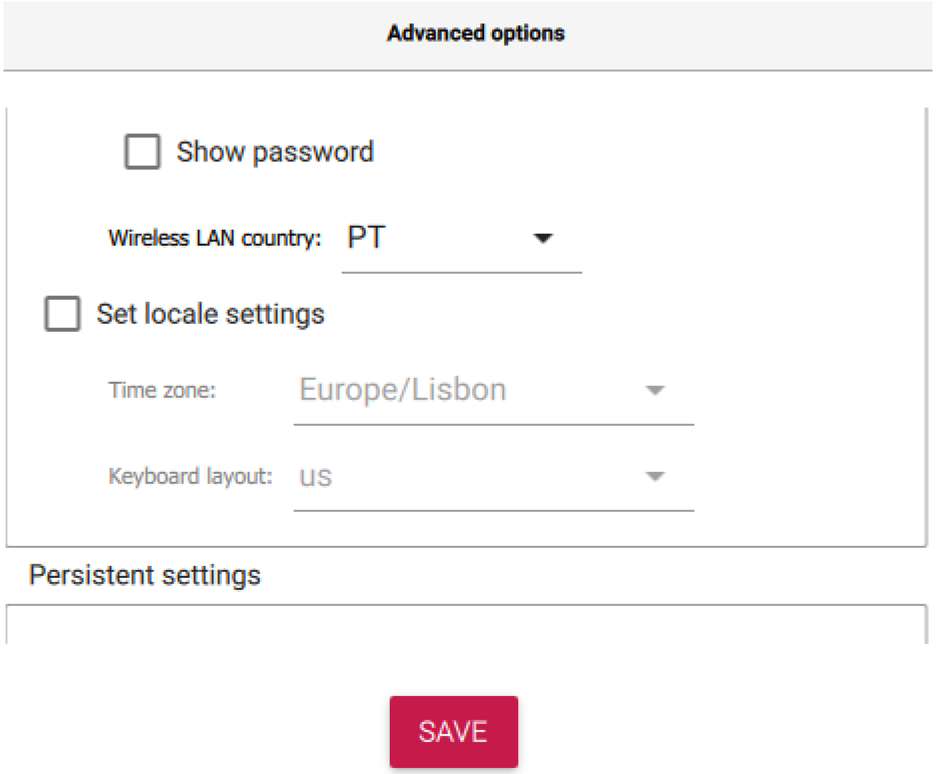
☑ Configure wireless LAN

SSID: [illegible]

□ Hidden SSID

Password: •••••••••••

11) Set up your country and time zone. Finally, click **Save**.



Advanced options

□ Show password

Wireless LAN country: PT

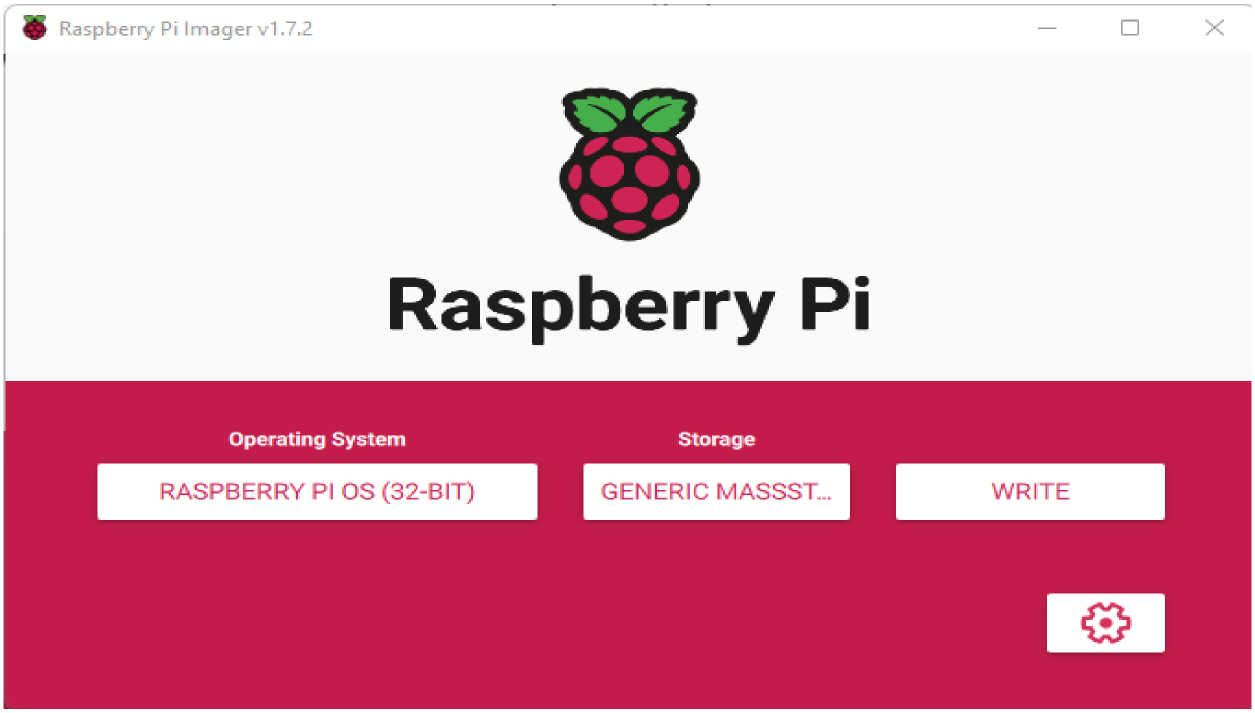
□ Set locale settings

Time zone: Europe/Lisbon

Keyboard layout: us

Persistent settings

12) After selecting the operating system, storage, and advanced settings, click on **write** to start installing the operating system on the microSD card.

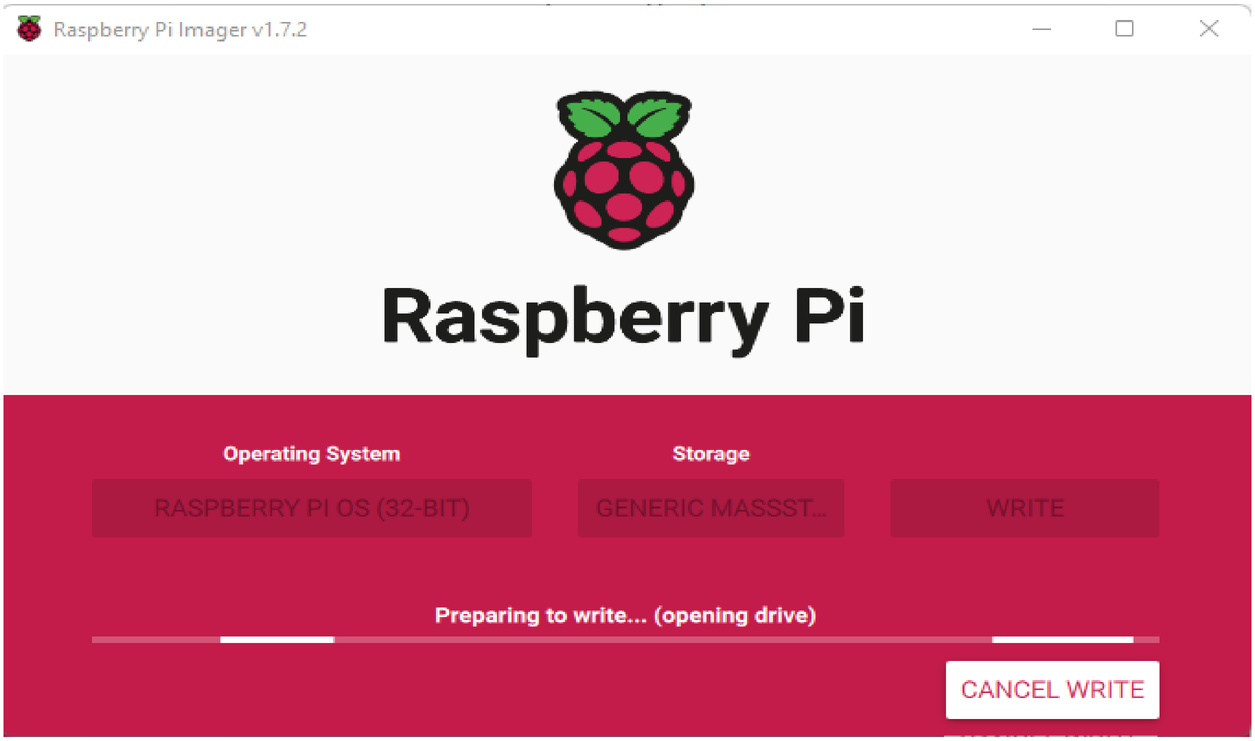


Raspberry Pi Imager v1.7.2

Operating System: RASPBERRY PI OS (32-BIT)

Storage: GENERIC MASSST...

13) Wait a few seconds while it installs the Operating System.

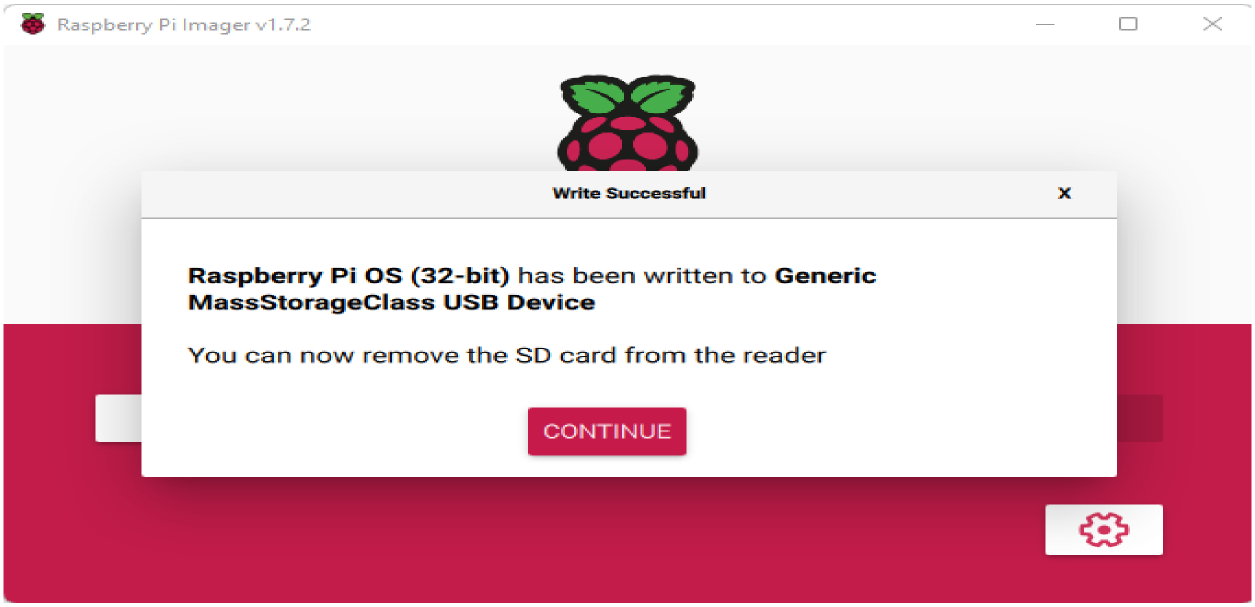


Operating System: RASPBERRY PI OS (32-BIT)

Storage: GENERIC MASSST...

Preparing to write... (opening drive)

14) When the installation is complete click on **Continue**. It will eject the microSD card safely.



Raspberry Pi OS (32-bit) has been written to **Generic MassStorageClass USB Device**

You can now remove the SD card from the reader

15) Now, remove the card from your computer and insert it into your Raspberry Pi. Then, apply power to the Raspberry Pi to start it.

OS SETTINGS:

image customization option: for this session only
set hostname: raspberry.local

Enable SSH
use password authentication

set username and password
username: pi
password: raspberry

configure wireless lan // apne wifi ka password hume set karke rakhna hai taki jab bhi raspberry pi on ho wo humare wifi network ke through connect ho sake
ssid: Houston system_5g
password: Cifyit@1212

wireless LAN country: IN

set location settings
time zone: Asia/culcutta
keyboard layout: US

persistent settings
audio sound
eject media when finished
enable telemetry

NOTE: isme hum dekh sakte hai ki hum ek raspberry pi ke andar SD card me operating system ko kaise install karte hai. Ye ek same process hai jaise hum apne computer me operating system jo ki ISO file hota hai usko usb drive ke through install karte hai. Us process me bhi hum ek software download karne ki jarurat padhti hai jisse ki hum apne pendrive ko bootable banate hai taki wo humara pendrive ISO file ko boot kar sake operating system ke image file ko. jaise ki 'RUFUS ','ETCHER', ye software USB ko bootable banana ke liye hota hai taki operating system ke image file ko usb me dal sake. Aur jab hum apne usb ko laptop me lgate hai aur keyboard press karte hai to humare laptop me windows install hone lag jata hai.

Thik isi tarah raspberry pi ke OS ko SD card me install karne ke liye hume apne SD card ko bootable karna hoga jisse ki SD card me OS ka file ja sake. So hum sabse pahle hum apne laptop ya computer me Raspberry Pi Imager (a tool to flash the OS on the microSD card) ko download karenge yah tool jo hai microSD card me OS ko flash karega(a tool to flash the OS on the

microSD card). Jab yeh tool download ho jayega to iske through hi hum apne operating system ko select karenge phir apne storage device ko select karenge jo ki humara SD card hi hai. itna karne ke bad hum advanced option me jayenge jaha hum raspberry pi ke liye username aur password set kar sakte hai. Ssh ko enable kar sakte hai taki humara raspberry pi headless mode means monitor, keyboard, mouse ke bina bhi kaam kar sake wirelessly humare laptop se connect karke. Aur wifi ki setting bhi kar sakte hai ssid name aur password dal denge hum apne wifi ka.
Itna sab karne ke bad me hum WRITE kar denge jisse ki OS humare SD card me install ho sake.

FINAL STEP:

1. SD card ko raspberry pi me laga de. Raspberry pi ka power adaptor se connect karke usko on karde. Humara adaptor 5v, 2.5A ka hona chahiye.
2. Humare raspberry pi me 2 usb port hai ek ko keyboard se connect karenge aur dusre ko mouse se connect karenge.
3. Humare raspberry pi me ek HDMI port bhi diya hua hai us port se hum apne monitor display ko connect kar denge. Display hum alag se bhi manga sakte hai yaa hum apne laptop ke HDMI port me laga kar apne laptop ke display ko use kar sakte hai.

2. RASPBERRY PI AS A HEADLESS MODE

humne apne SD card me raspberry pi ka operating system install kar diya tha aur yah perfectly work bhi kar rha hai. Lekin ab hum chahte hai ki hum kaise apne laptop se raspberry pi ko control kar sakte hai ya means hum raspberry pi ko headlessly (bina monitor, keyboard aur mouse ke) kaise operate kar sakte hai. Aur hum apne laptop me putty software ko download karke raspberry pi ke full screen ko apne laptop se control kar sakte hai.
So hum is lecture me sikhenge ki hum kaise raspberry pi ko heedlessly use kar sakte hai aur iska setup hum kis tarah se karenge.

1. Install the OS file on SD card
2. **Connecting the Raspberry Pi to the Internet:** If you’ve already set up the network credentials in the previous instructions, you can skip this section.

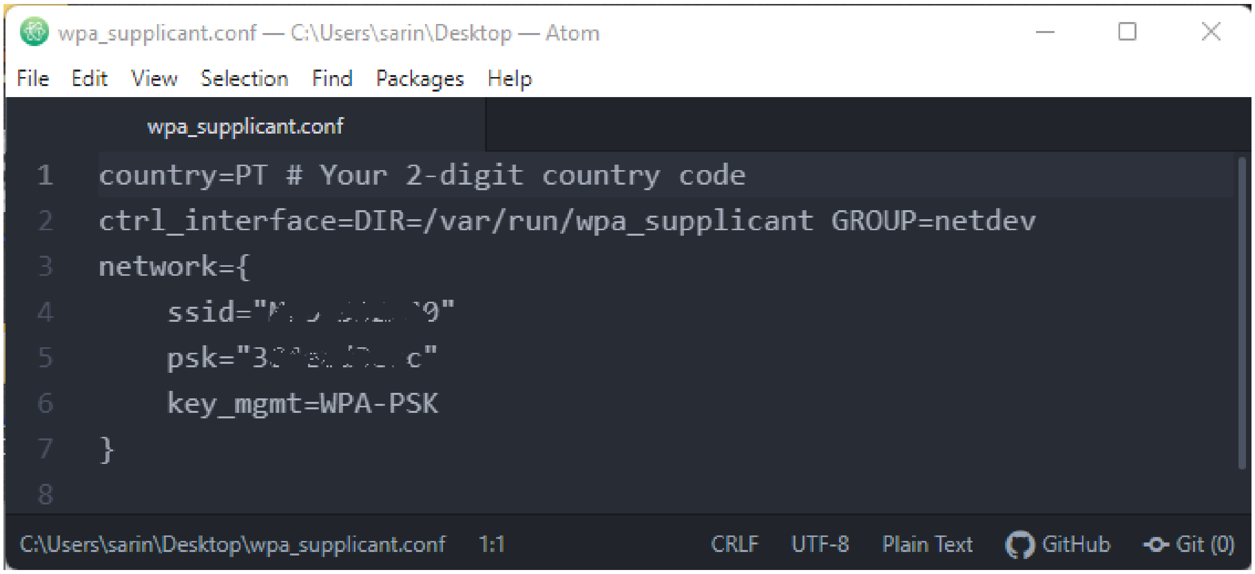
In this section, we'll set up the Raspberry Pi to connect to the internet (your local network) when it first starts, without the need to connect a monitor, keyboard, or mouse (this is called *headless*).
Remove the microSD card from your computer and insert it again so that your computer can recognize the microSD card again.

1) Create a new file called `wpa_supplicant.conf` and paste the following.

```
country=PT # Your 2-digit country code
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="YOUR_NETWORK_NAME"
    psk="YOUR_PASSWORD"
    key_mgmt=WPA-PSK
}
```

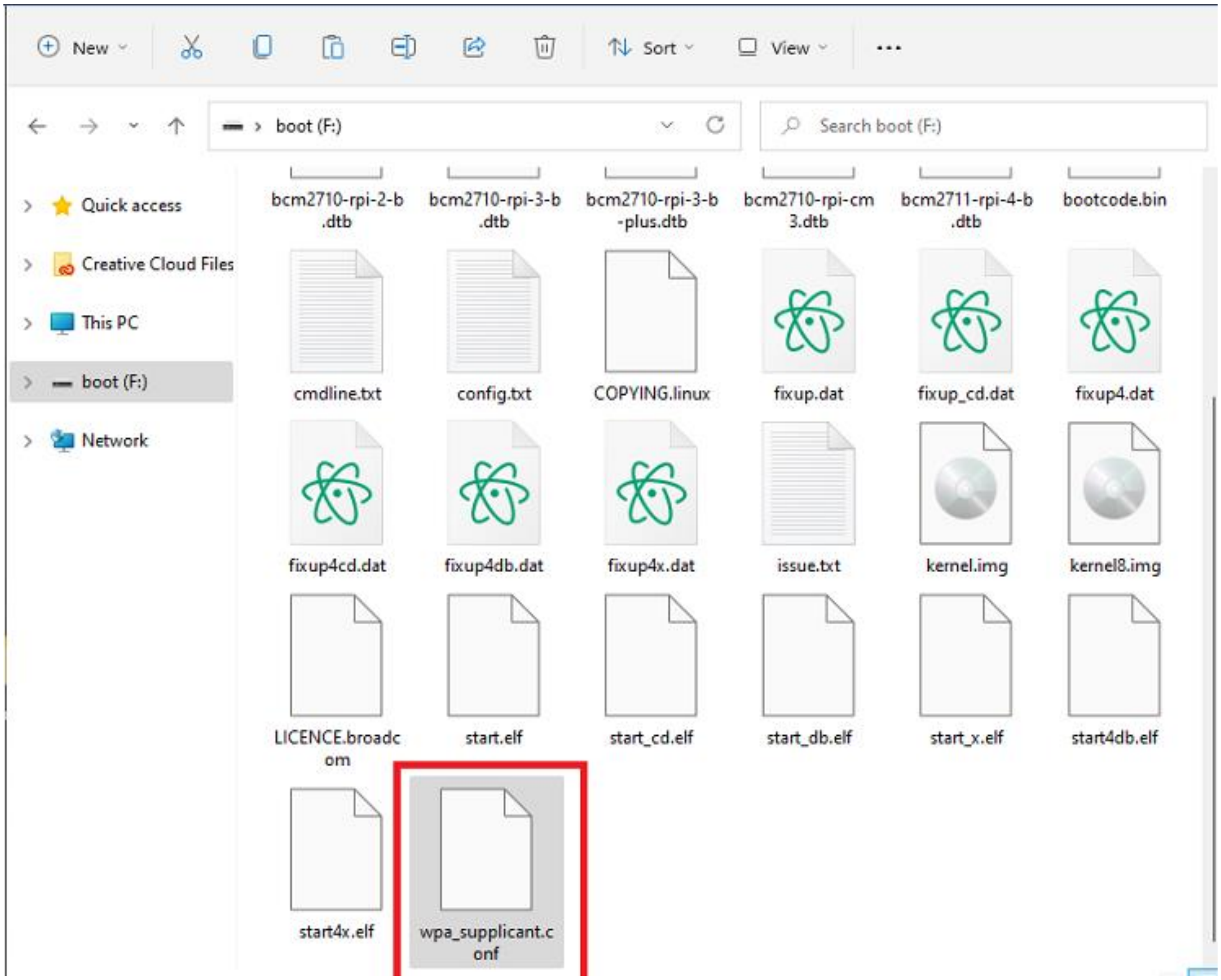
```
country=IN # Your 2-digit country code
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
           ssid=" Houston system_5G "    # YOUR_NETWORK_NAME
           psk=" Cifyit@1212"           # wifi ka password yaha likhenge
           key_mgmt=WPA-PSK
         }
```

Insert your 2-digit country code like usme PT likha hai to hum uski jagah india ka IN likh denge and the network credentials (SSID and password) of the network that you want the Raspberry Pi to connect to.


wpa_supplicant.conf — C:\Users\sarin\Desktop — Atom
File
Edit
View
Selection
Find
Packages
Help
wpa_supplicant.conf
C:\Users\sarin\Desktop\wpa_supplicant.conf
1:1
CRLF
UTF-8
Plain Text
GitHub
Git (0)

```
country=PT # Your 2-digit country code
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="[redacted]"
    psk="[redacted]"
    key_mgmt=WPA-PSK
}
```

2) Open the microSD card folder and move the 'wpa_supplicant.conf' file there.



3) Enabling SSH:

If you've already enabled SSH in the previous instructions, you can skip this section.

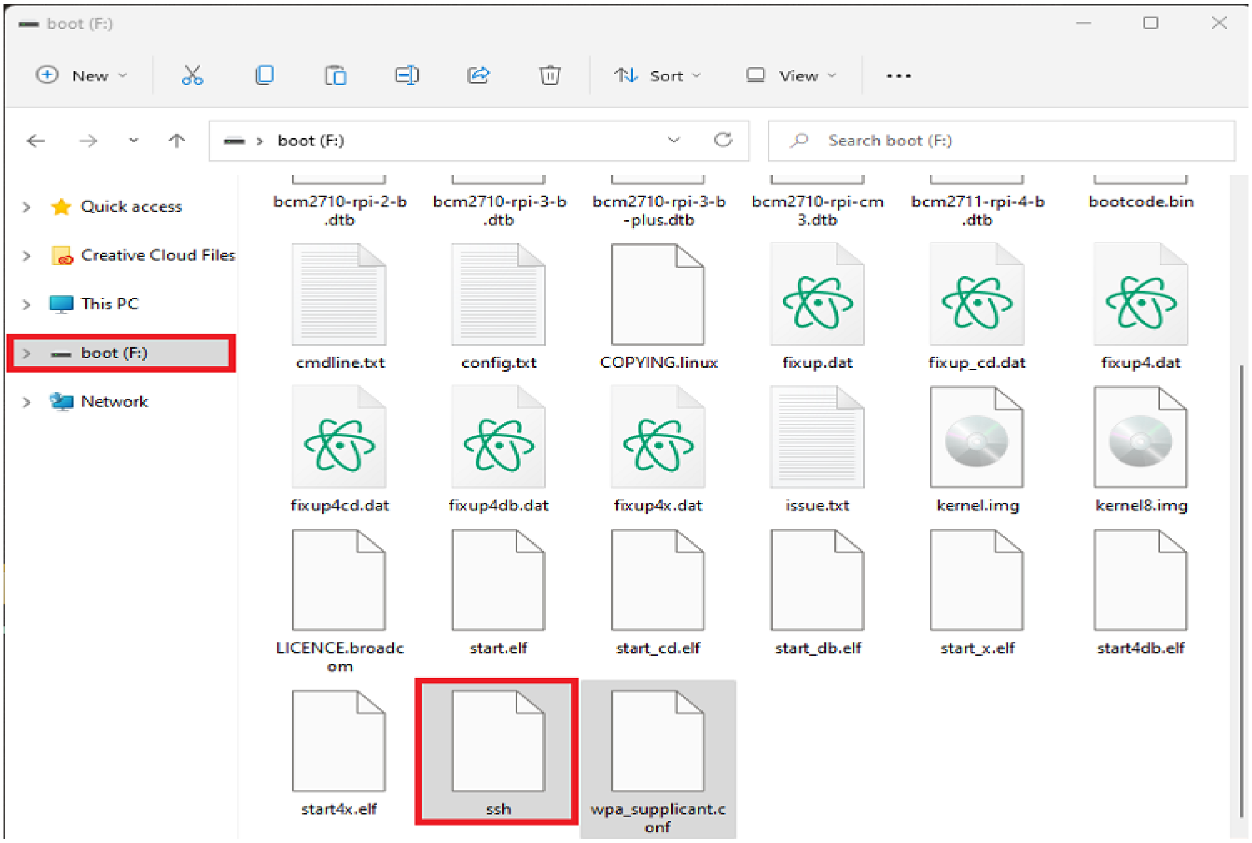
To access your Raspberry Pi remotely (via a computer on your local network) so you don't need to connect a monitor, keyboard, or mouse to the Raspberry Pi, you need to enable SSH. SSH allows you to connect to your Raspberry Pi remotely from another machine and access the command line.

To enable SSH, you need to create a file called `ssh` (**without any extensions**) on the `boot` folder of the microSD card.

1) You can click here to download the ssh file.

2) Move that file to the `boot` folder of the microSD card.

Note: ek SSH file banana hai empty jiske andar likha hai echo is on.



When the Raspberry Pi starts up, the OS will find the `ssh` file and will automatically activate SSH.
Now, safely eject the microSD card from your computer and insert it on the Raspberry Pi. Then, power up the Pi using a 5V DC power source.

NOTE: hum in above steps ko jo wifi aur ssh enable karne ka hai ise skip bhi kar sakte hai kyuki ye sari chize jab hum OS ko install karte hai to us samay par hume dalna padhta hai. Agar hum us samay par ye sari details nahi dalte hai tb hume in process ko karne ki jarurat padhti hai. So hume yad hai ki hum apne wifi ka ssid aur password us samay hi dal diye the aur ssh ko bhi enable kar diye the so ab hume phir se ye sab chize karne ki jarurat nahi hai.

OS ko SD card me dalne ke bad agar hum apne raspberry pi ko heedlessly chalana chahte hai to hum eek to putty software download karna padhega kyuki iski madad se hi hum raspberry pi ke command line ya terminal ke through raspberry pi ka jo VNC hai usko enable kar sakte hai. Raspberry pi ke VNC ko enable karne ke bad hum apne laptop me bhi VNC viewer ko download kar lenge aur raspberry pi ka hostname, username aur password fill karne ke bad ye hume access de dega ki hum apne raspberry pi ke desktop ko apne laptop par dekh sake.

4) Connecting via SSH to the Raspberry Pi

SSH (which stands for secure shell) is a method of establishing a communication with another computer securely. All data sent via SSH is encrypted. SSH is based on a Unix shell, so it allows you to access your Raspberry Pi files from a remote machine by using terminal commands.

This part is divided into two sections:

- A) Windows
- B) Mac OS X/Linux.

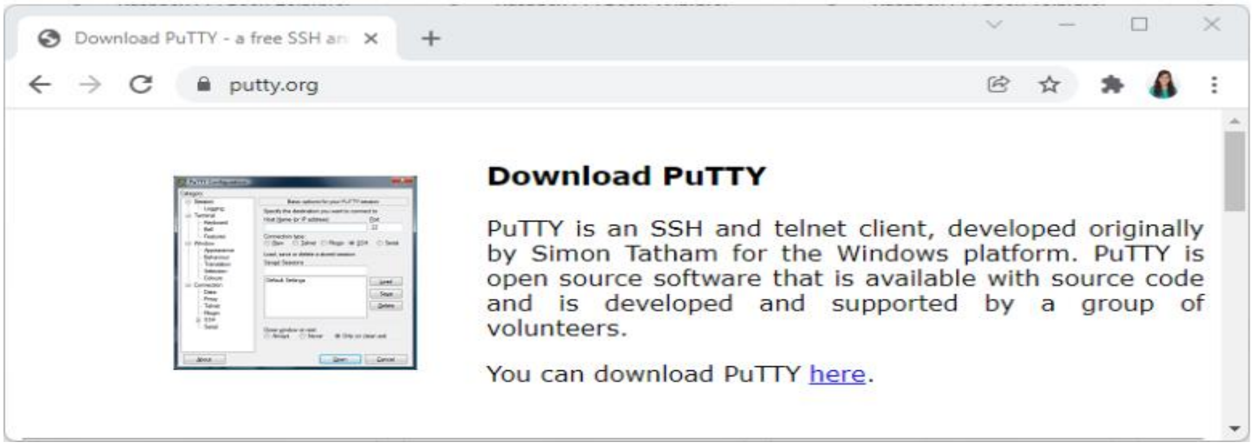
A) Windows

To communicate with the Raspberry Pi via SSH, you need a software to handle SSH communication. We'll use PuTTY. You need to install it on your computer if you haven't already.

Downloading PuTTY Or Download VNC viewer

If you use Windows, you need to download and install a free application called PuTTY. Here's how to install it:

1) Open your web browser and go to www.putty.org
2) Download PuTTY. We recommend downloading the ***putty.exe*** file.

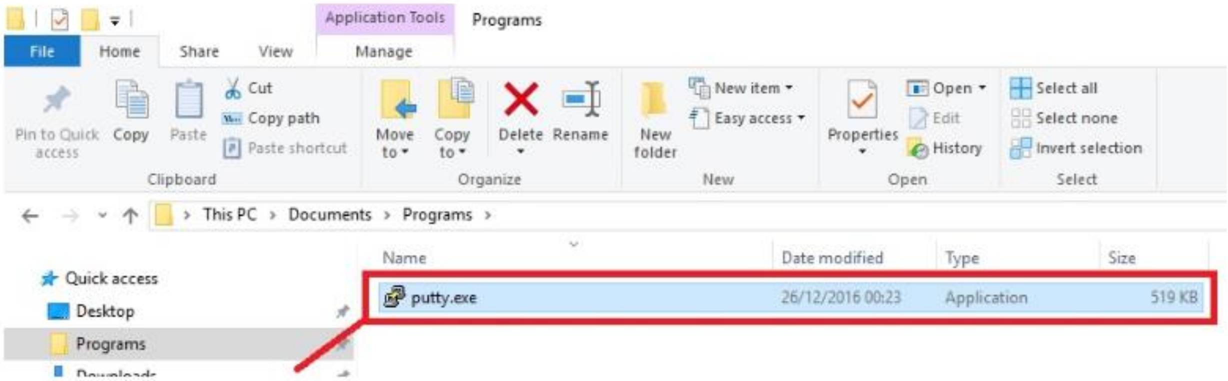


Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

You can download PuTTY here.

3) **Run the** *putty.exe* **file to execute the software.**



Name	Date modified	Type	Size
putty.exe	26/12/2016 00:23	Application	519 KB

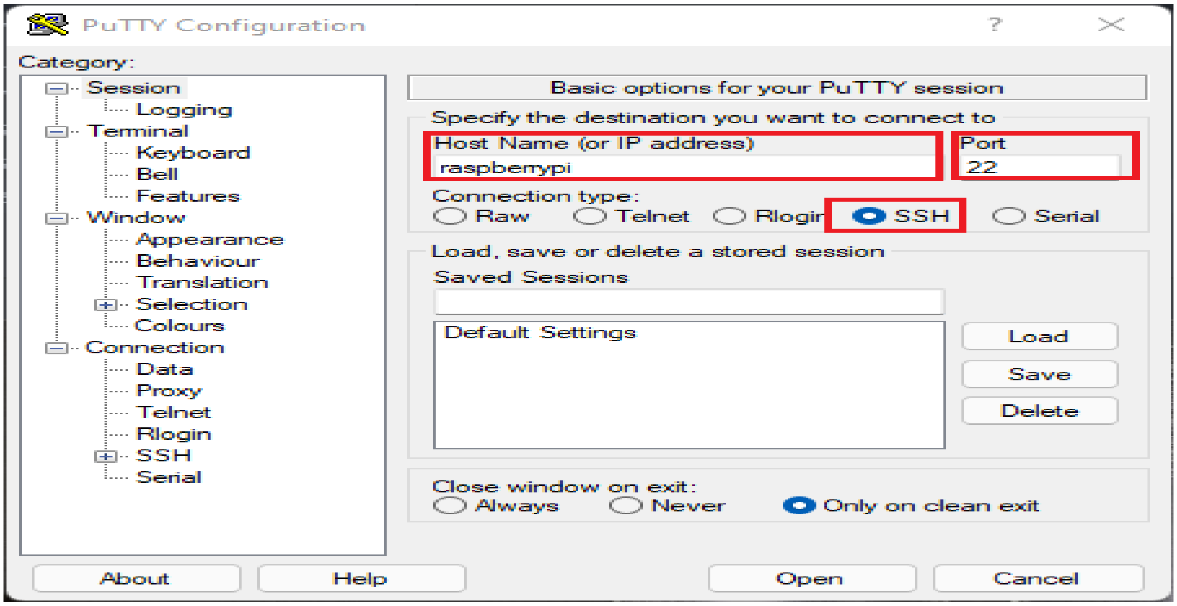
Connecting to the Raspberry Pi via SSH

With PuTTY installed, power up your Raspberry Pi and follow these steps:

1) Open PuTTY.
2) Select/enter the following options:

- Host Name: **raspberrypi.local**
- Port: **22**
- Connection type: **SSH**

* **raspberrypi** is the default hostname. If you’ve inserted a different hostname in the installation process, you should use that instead.



Specify the destination you want to connect to

Host Name (or IP address): raspberrypi

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

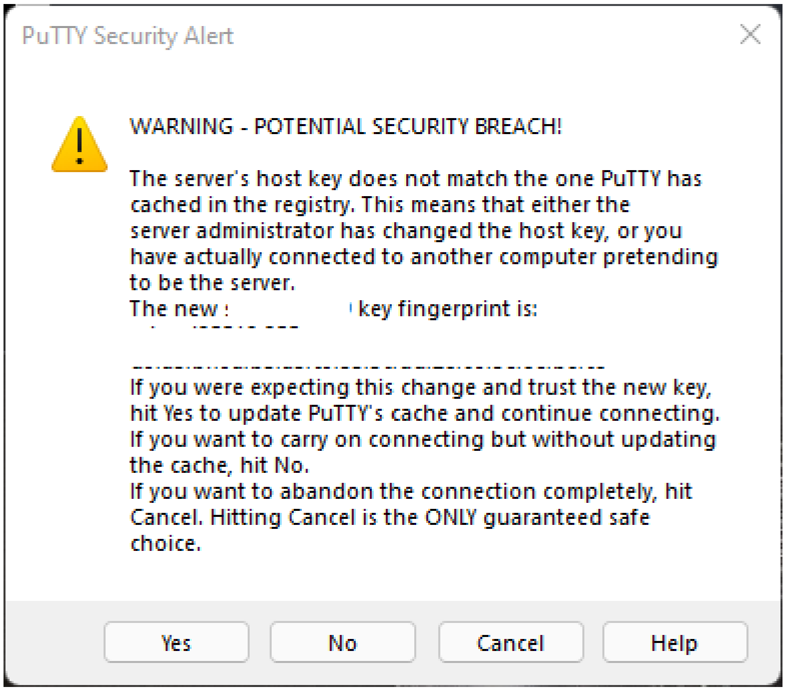
Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

3) Click Open.

4) When you connect to your Raspberry Pi for the first time, you’re prompted by a message warning you that you’re attempting to establish a connection with an unknown host. Simply click **No** to proceed.



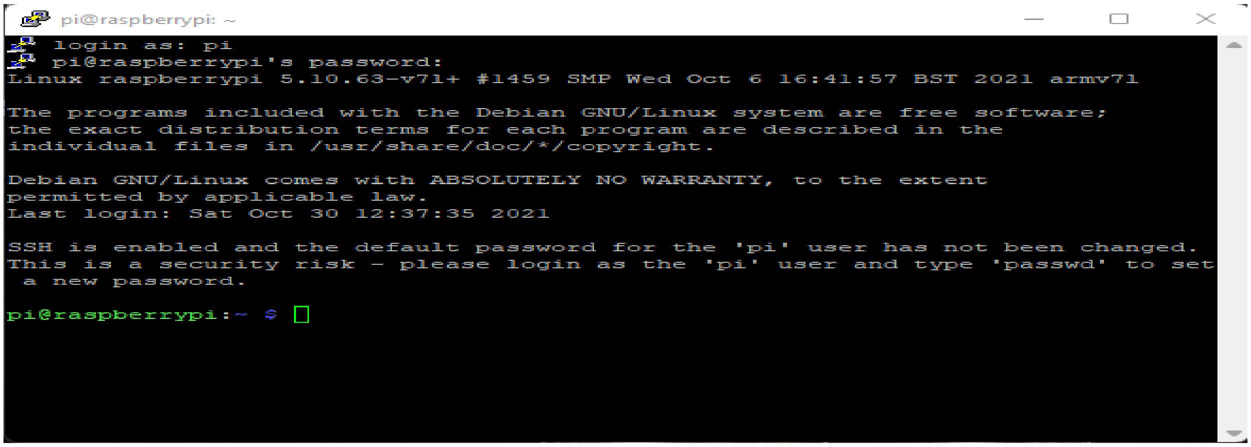
WARNING - POTENTIAL SECURITY BREACH!

The server's host key does not match the one PuTTY has cached in the registry. This means that either the server administrator has changed the host key, or you have actually connected to another computer pretending to be the server.
The new : key fingerprint is:

If you were expecting this change and trust the new key, hit Yes to update PuTTY's cache and continue connecting.
If you want to carry on connecting but without updating the cache, hit No.
If you want to abandon the connection completely, hit Cancel. Hitting Cancel is the ONLY guaranteed safe choice.

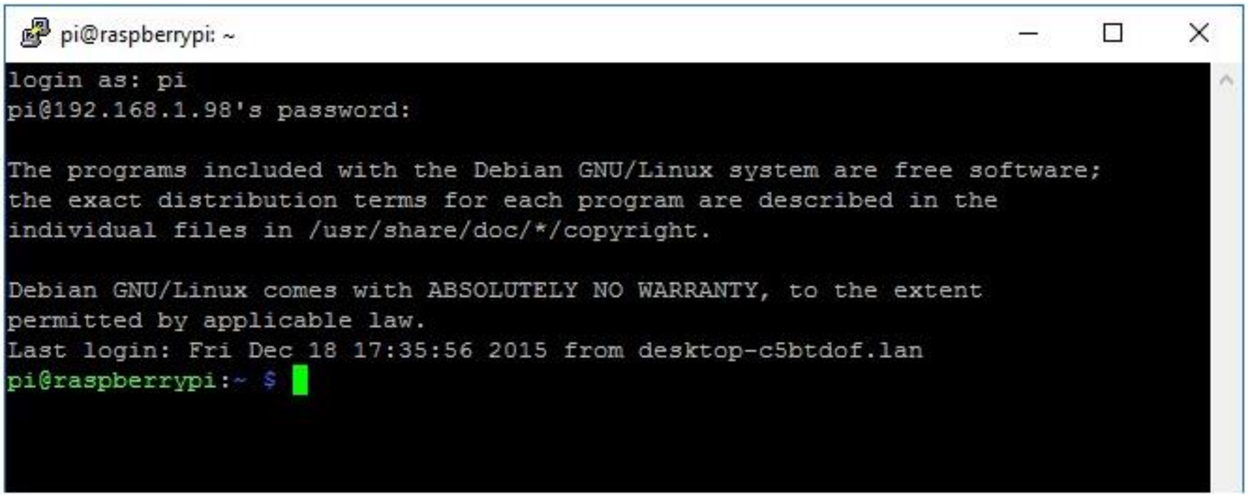
5) Now, you need to login to your Raspberry Pi using the username and password you set on the installation process.
On the new window that opens, type your username and hit Enter.

6) Then, enter your password and hit Enter. You won’t see any characters showing up on the window while you type the password.



```
login as: pi
pi@raspberrypi's password:
Linux raspberrypi 5.10.63-v71+ #1459 SMP Wed Oct 6 16:41:57 BST 2021 armv71

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: Sat Oct 30 12:37:35 2021

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $
```

Now you have an SSH communication established with your Raspberry Pi. This will be useful to install software on your Pi, run your programs, create folders or files, etc…



```
login as: pi
pi@192.168.1.98's password:

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: Fri Dec 18 17:35:56 2015 from desktop-c5btdof.lan
pi@raspberrypi:~ $
```

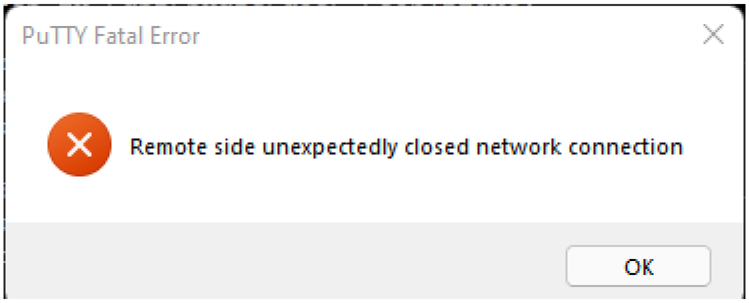
Shutting Down

To shut down your Raspberry Pi, simply type this command on the command line:

```
pi@raspberry:~ $ sudo poweroff
```

The SSH connection will be shut down right after

.



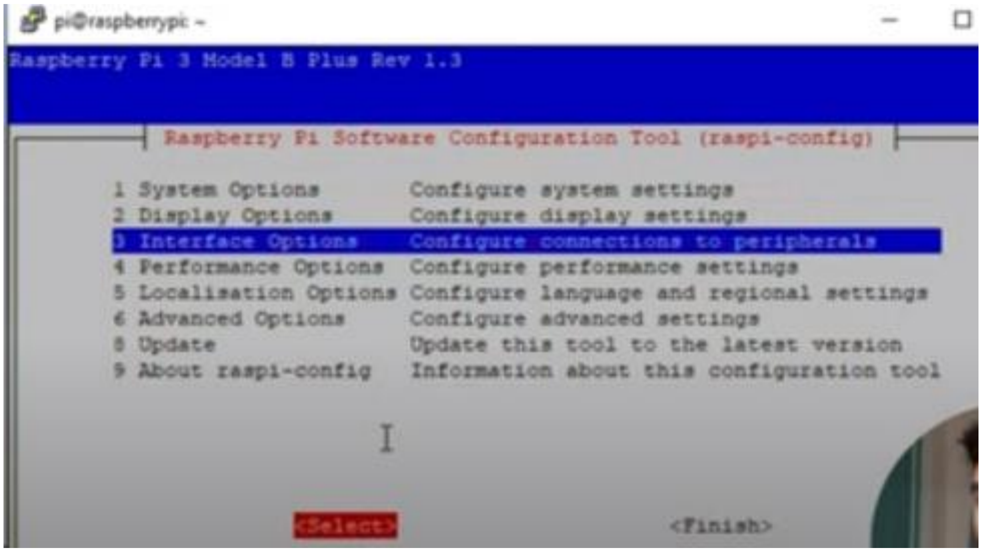
Remote side unexpectedly closed network connection

Then, if you're running the Raspberry Pi headless, you can establish an SSH connection with your Pi to install software, run programs, create folders, and files, etc.

NOTE: **raspberry pi ko headless mode me chalane ke liye hume putty software ko download karna padhta hai. Putty software installed hone ke bad jab hum ise open karenge to hume hostname: raspberrypi.local, port number: 22 aur SSH ko select karna padhta hai. Iske bad jab hum enter press karte hai to ek windows open hota hai jisme hume apna raspberry pi ka username aur password dalna padhta hai jo humne OS ko installed karte samay dala tha. Jo ki humare case me username: pi hai aur password: raspberry hai. So ye dalne ke bad hum enter press karenge. Aur hum apne raspberry pi ke andar successfully login ho chuke hai. Means raspberry pi ka jo terminal hota hai jo command line hota hai usko mai yaha se control kar sakta hu. Lekin agar hum chahte hai ki hum raspberry pi ke desktop monitor ko gui ke through control karna apne laptop se hi to hum is command line ke through raspberry pi me apne VNC ko install kar denge aur ek aisa hi VNC hum apne laptop me bhi install kar lenge. Aur apne laptop ke VNC se raspberry pi ke andar ke VNC se connection banakar hum log apne raspberry pi ke desktop ko access ya control karenge.**

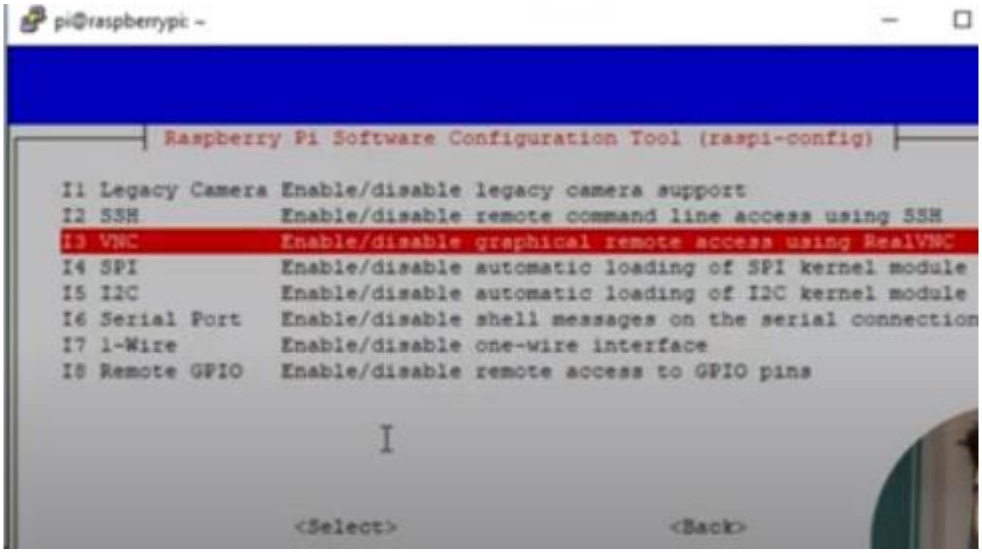
1. **Ab hum dekhte hai ki putty software ke through command line se hum raspberry pi ke andar VNC ko kaise install karte hai.**

2. **sudo raspi-config command enter karenge. Jaise hi hum is command ko enter karenge to humare pass ek interface open hoga.**



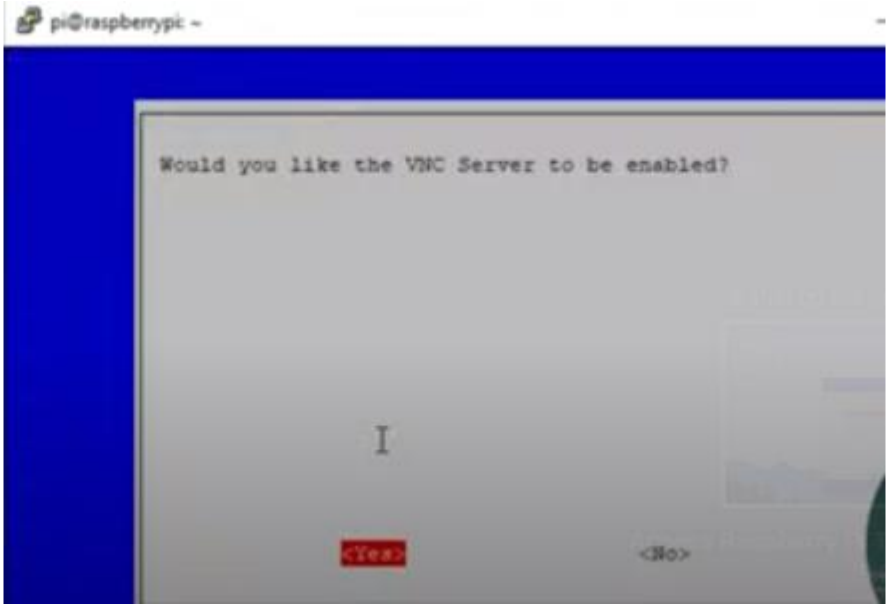
```
pi@raspberrypi: ~
Raspberry Pi 3 Model B Plus Rev 1.3

┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

    1 System Options       Configure system settings
    2 Display Options      Configure display settings
    3 Interface Options    Configure connections to peripherals
    4 Performance Options  Configure performance settings
    5 Localisation Options Configure language and regional settings
    6 Advanced Options     Configure advanced settings
    8 Update               Update this tool to the latest version
    9 About raspi-config   Information about this configuration tool

              <Select>                 <Finish>
```

3. So hum 3rd option jo hai interface option usko right arrow key se select krenge phir select button par enter karenge. Ab hum eek aur interface dikhayi dega kuch is tarah ka



```
pi@raspberrypi: ~

┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

  I1 Legacy Camera Enable/disable legacy camera support
  I2 SSH           Enable/disable remote command line access using SSH
  I3 VNC           Enable/disable graphical remote access using RealVNC
  I4 SPI           Enable/disable automatic loading of SPI kernel module
  I5 I2C           Enable/disable automatic loading of I2C kernel module
  I6 Serial Port   Enable/disable shell messages on the serial connection
  I7 1-Wire        Enable/disable one-wire interface
  I8 Remote GPIO   Enable/disable remote access to GPIO pins

              <Select>                 <Back>
```

4. Ab hum apne VNC ko right arrow se select karke select par enter karenge. Ab yah humse puchega ki VNC ko ON karna hai ya nahi. So hum on par click kar denge. Iske bad hum right arrow ko do bar press karenge finish par jane ke liye phir finish ko select karke enter kar denge.

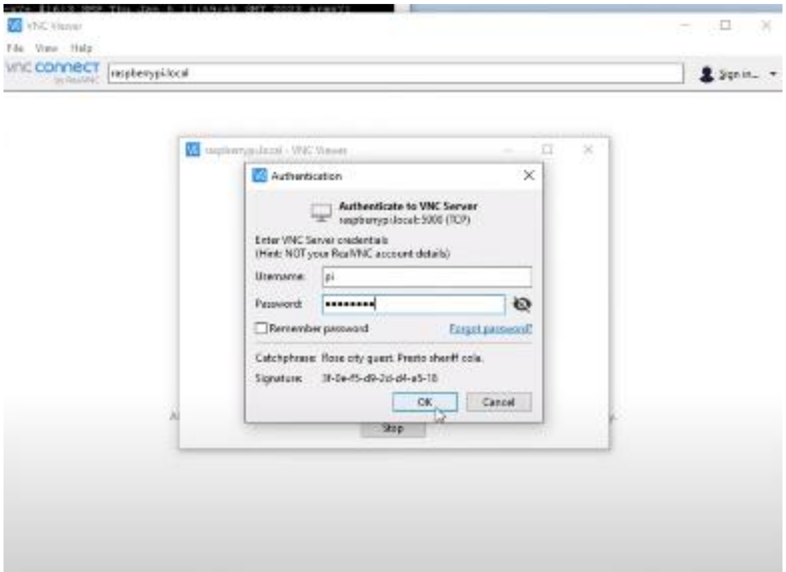


```
Would you like the VNC Server to be enabled?
```

5. Finish karne ke bad hum phir se apne putty ke terminal par pahuch jayenge

6. Iske bad hum apne raspberry pi ko ek bar restart karenge iske liye hum command line par likhenge: sudo reboot aur humara raspberry pi restart ho jayega.

7. Ab hum apne laptop me VNC viewer ko download kar lenge. Vnc viewer ko download karne ke bad upar hum apne raspberry pi ka hostname likhenge jo ki humare case me raspberrypi.local hai. Itna likhne ke bad jab hum enter karenge to ek windows pop up hoga. Isme hume apne raspberry pi ka username aur password likhna hoga so username: pi hai aur password: raspberry hai humare case me. Yeh sab chiz OS install karte samay humse puchha jata hai.



raspberrypi.local

Authenticate to VNC Server
raspberrypi.local:5900 (TCP)

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

Username: pi

Password: ••••••••

☐ Remember password Forgot password?

Catchphrase: Rose city guest. Presto sheriff cola.

Signature: 3f-0e-f5-d9-2d-d4-a5-18

OK Cancel

8. Ab hum apne raspberry pi ke screen ko apne laptop par dekh sakte hai vnc ke through.

9. Agar hum apne raspberry pi ke display ki settings ko change karna chahte hai to hum raspberry pi ke icon> preferences>raspberry pi configurations>display>headless resolution ko 1920*1080 par set kar denge.

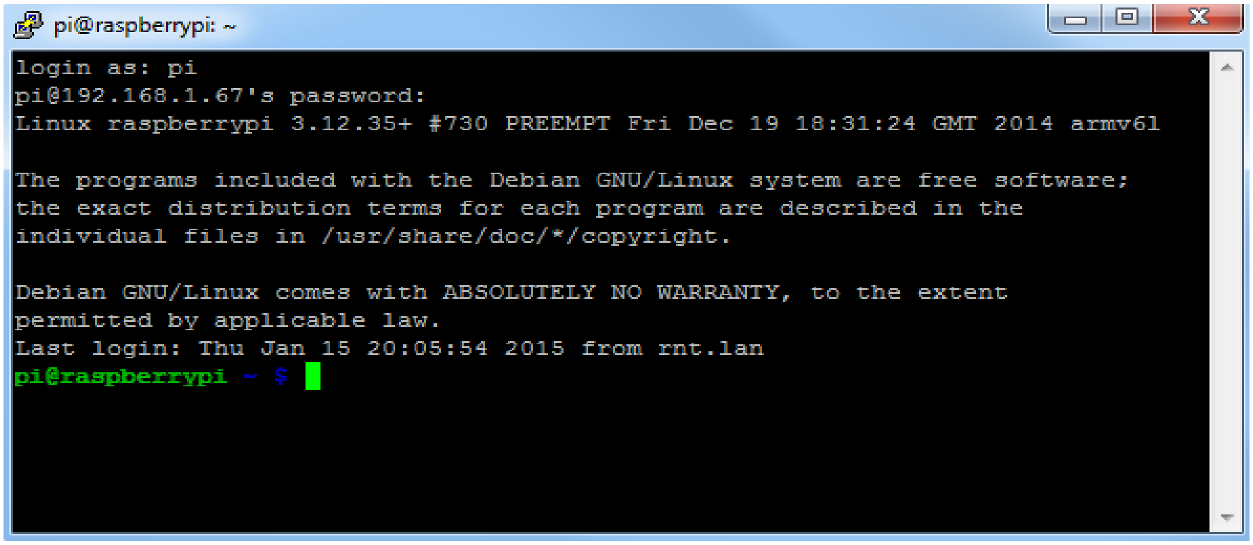
3. How to Configure WiFi on Your Raspberry Pi

In this post, you'll configure the WiFi on your Raspberry Pi. If it's your first time using the RPi, first make sure you read my Getting Started with Raspberry Pi guide.
Setting up your WiFi via the command line is the best method, since you don't necessary need access to the GUI (Graphical User Interface) everything you need is already right there (in your command line).

1) Accessing Your Command Line

Boot your Raspberry pi with the WiFi adapter plugged in. You can access your command line using one of the following methods:

- Having an Ethernet connection ensures that you can open an SSH client like PuTTY to establish an SSH communication
- Using the Raspberry Pi GUI to open your terminal window



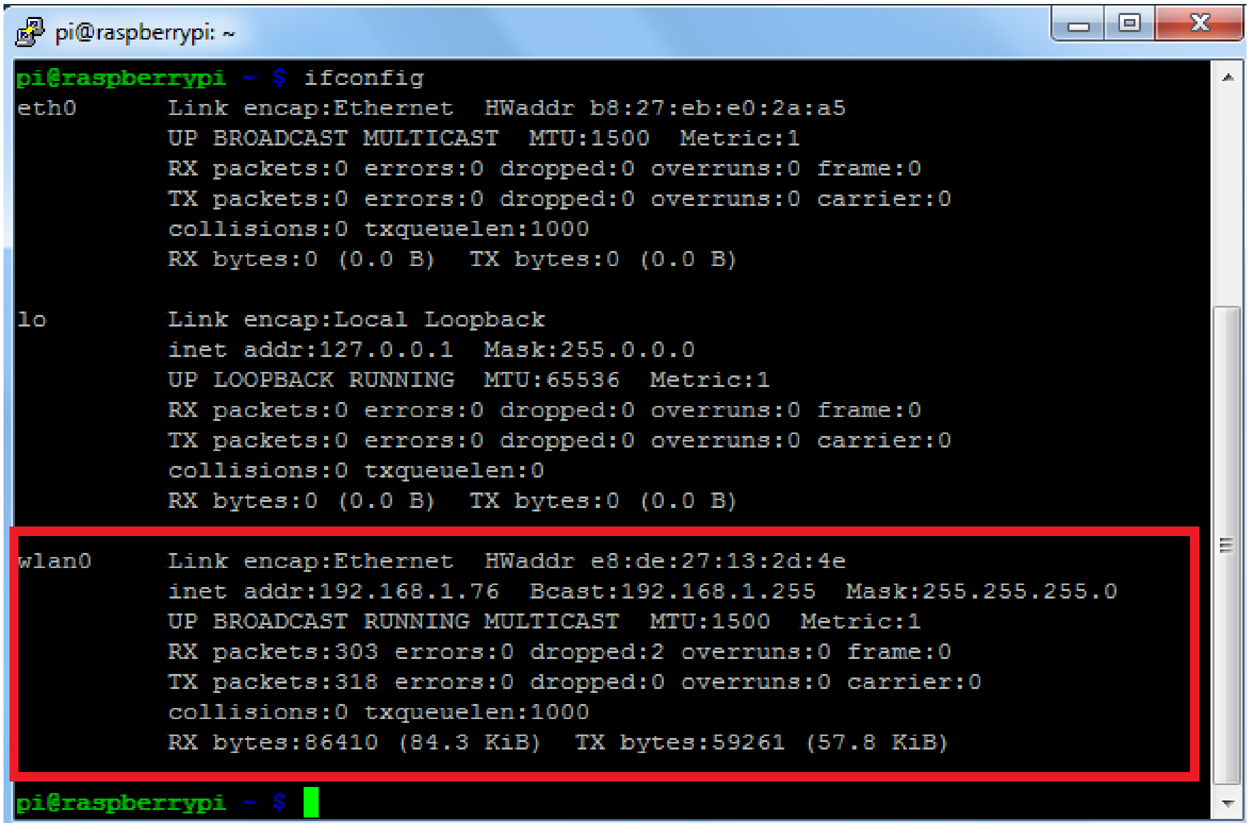
```
login as: pi
pi@192.168.1.67's password:
Linux raspberrypi 3.12.35+ #730 PREEMPT Fri Dec 19 18:31:24 GMT 2014 armv6l

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: Thu Jan 15 20:05:54 2015 from rnt.lan
pi@raspberrypi ~ $
```

3. Checking if your RPi recognizes your WiFi adapter:

There are several ways to check if your WiFi adapter has been recognized. You can type:

```
ifconfig
```

And your wireless adapter named as *wlan0* should appear as shown in the Figure below.



```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:e0:2a:a5
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr e8:de:27:13:2d:4e
          inet addr:192.168.1.76  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:303 errors:0 dropped:2 overruns:0 frame:0
          TX packets:318 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:86410 (84.3 KiB)  TX bytes:59261 (57.8 KiB)

pi@raspberrypi ~ $
```

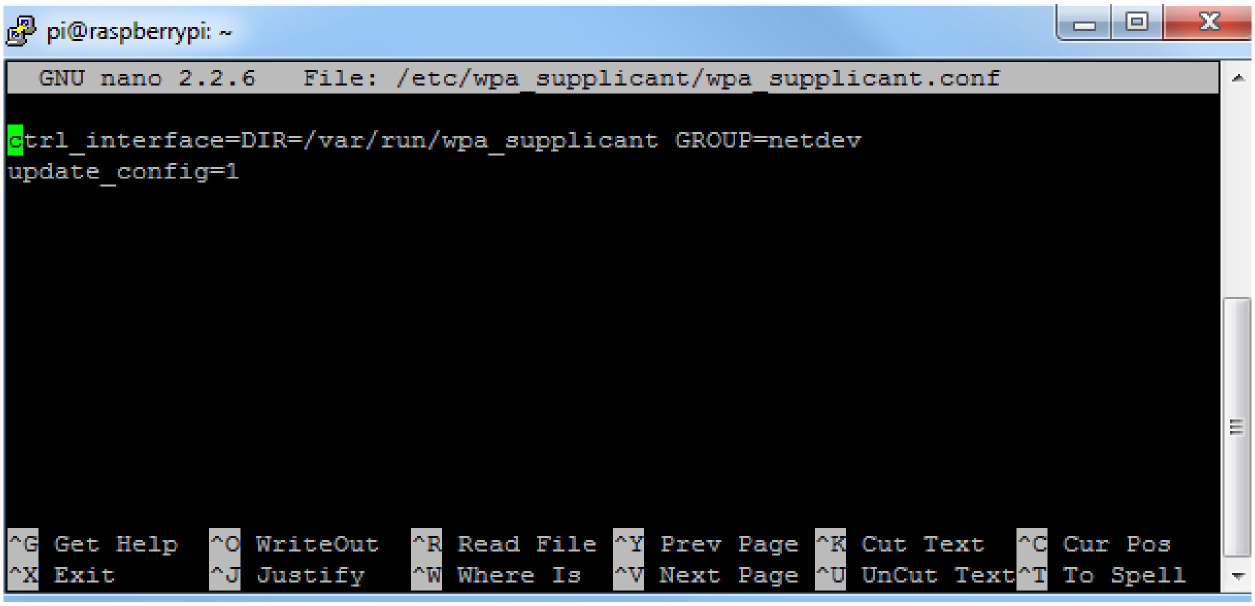
If you don’t see your WiFi adapter listed, you might have to install drivers for your particular WiFi adapter. I’m using the TL-WN725N and by default Raspbian doesn’t support my WiFi adapter. So I’ve followed this thread to install my drivers.

If you don’t have a WiFi adapter yet, I highly recommend purchasing the Edimax EW 7811UN. This is a good option, because Raspbian comes with its drivers installed out of the box, that ensures that your RPi recognizes that WiFi adapter.

3) Opening configuration file

Type the following command to open your configuration file:

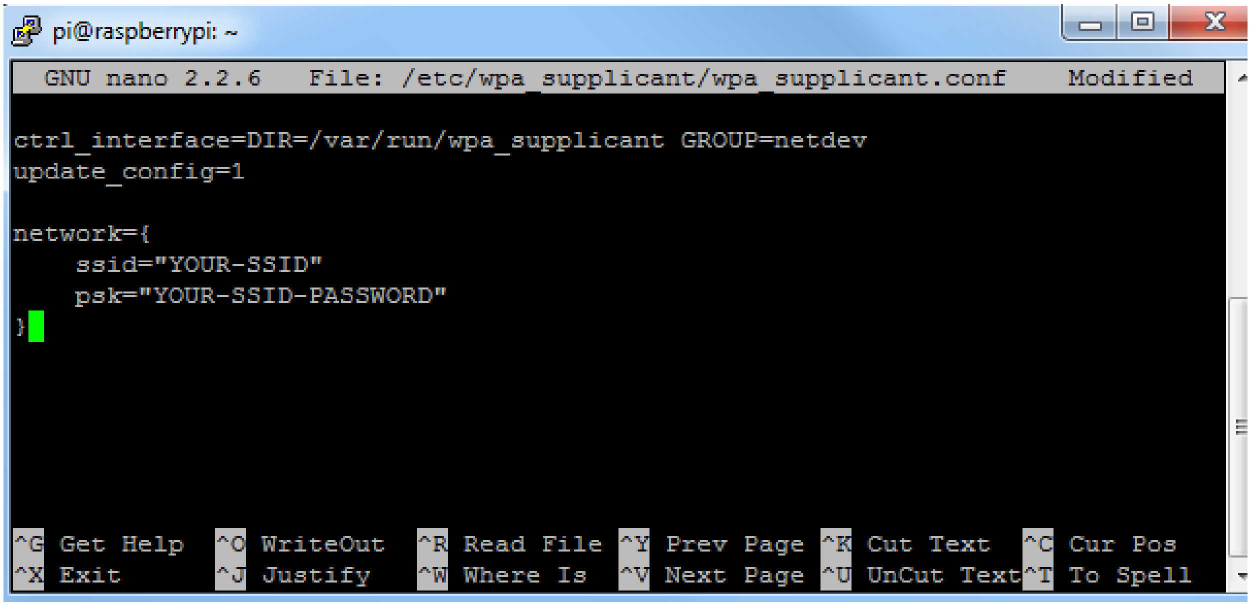
```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```



```
  GNU nano 2.2.6    File: /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

4) Adding your network details

Go to the bottom of your configuration file *wpa_supplicant.conf* and add your network details as shown below. Replace “**YOUR-SSID**” and “**YOUR-SSID-PASSWORD**” with the details of your WiFi connection.

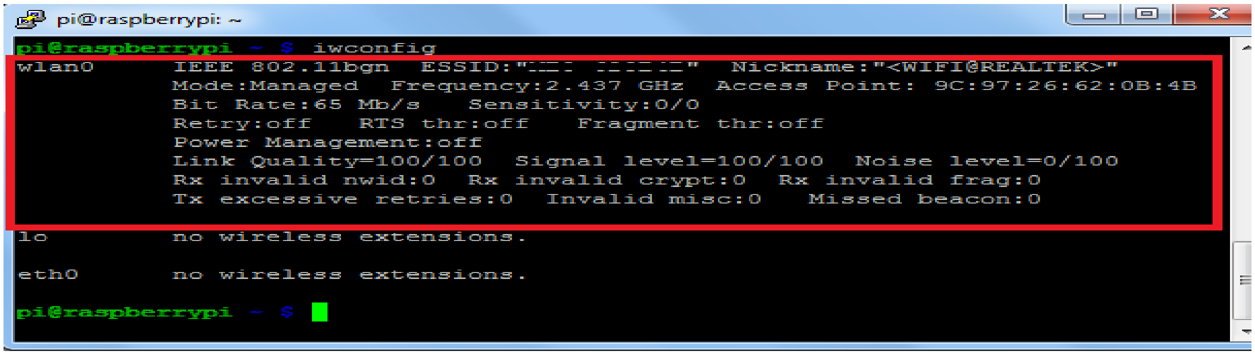


```
  GNU nano 2.2.6    File: /etc/wpa_supplicant/wpa_supplicant.conf    Modified

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="YOUR-SSID"
    psk="YOUR-SSID-PASSWORD"
}

^G Get Help   ^O WriteOut   ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Now save your file by pressing **Ctrl+x** then **y**, then finally press **Enter**.

5) Testing your connection

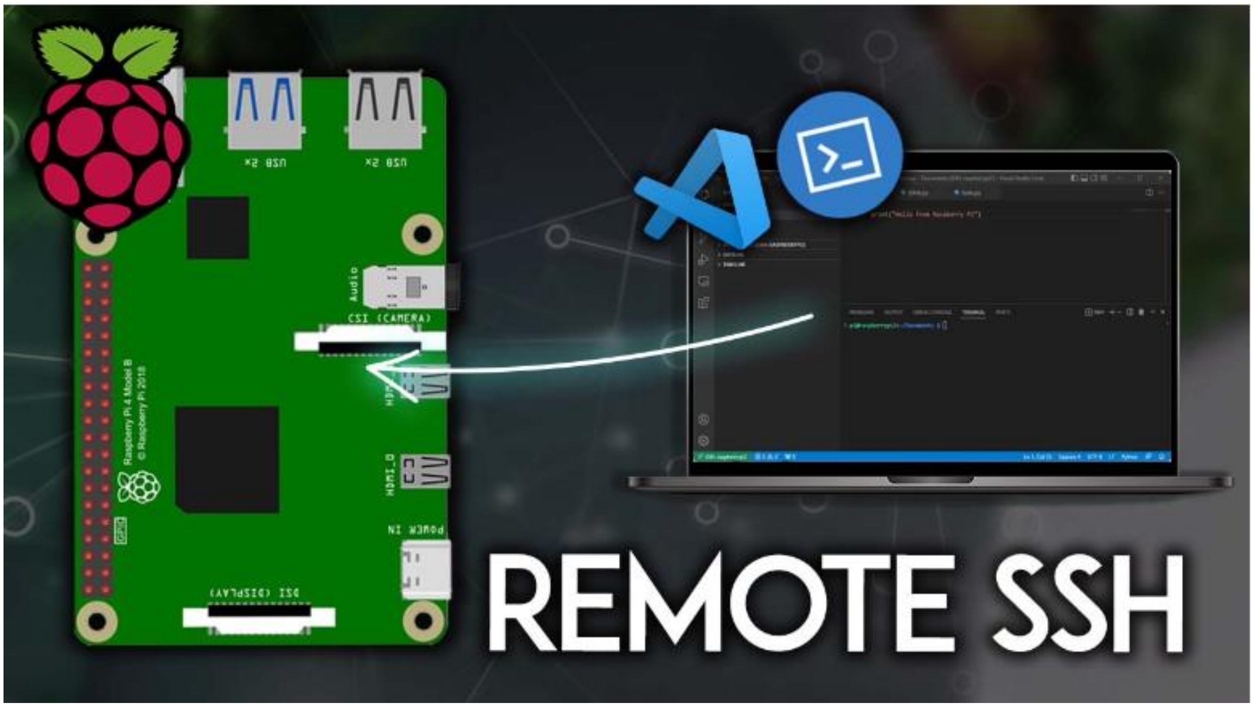
You can check the status of the wireless connection using *ifconfig* (to see if *wlan0* has acquired an IP address) and *iwconfig* to check which network the wireless adapter is using.



```
pi@raspberrypi ~ $ iwconfig
wlan0     IEEE 802.11bgn  ESSID:"[redacted]"  Nickname:"<WIFI@REALTEK>"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 9C:97:26:62:0B:4B
          Bit Rate:65 Mb/s   Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality=100/100  Signal level=100/100  Noise level=0/100
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

lo        no wireless extensions.

eth0      no wireless extensions.

pi@raspberrypi ~ $
```

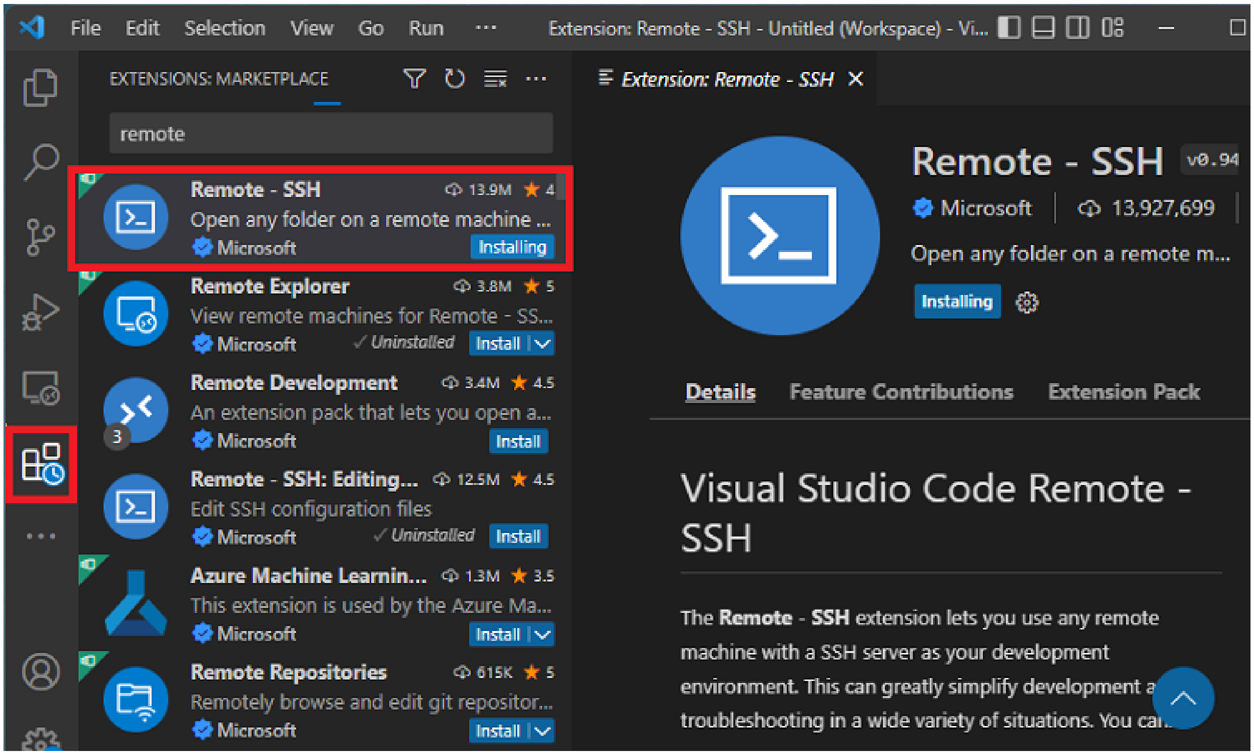
4. Programming Raspberry Pi Remotely using VS Code (Remote-SSH)

In this tutorial, we'll show you a practical way to create and execute files remotely on your Raspberry Pi using VS Code on your computer. All you have to do is to install the Remote-SSH extension on VS Code. This extension allows you to establish an SSH connection with your Pi, create files, write code and execute it directly on your Raspberry Pi board from your computer using VS Code interface. This is the perfect solution if your Raspberry Pi is running headless.



Installing Remote – SSH Extension on VS Code

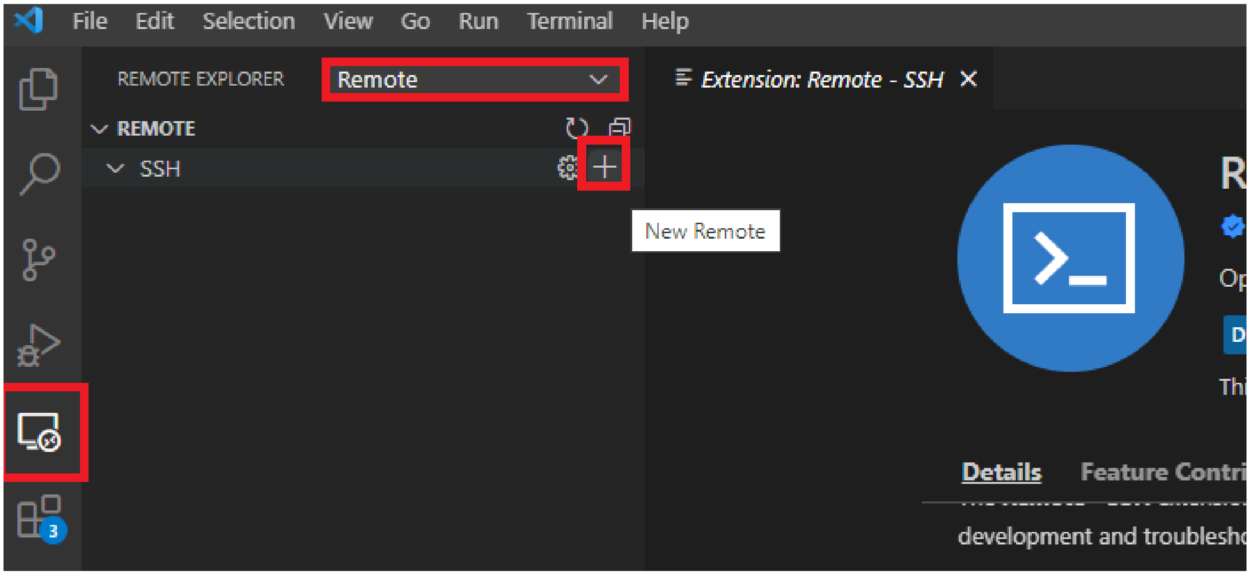
First, you need to install the **Remote – SSH** extension on VS Code. Click on the **Extensions** icon on the left sidebar, search for *remote*, and install the **Remote – SSH** extension.


File Edit Selection View Go Run ···
Extension: Remote - SSH - Untitled (Workspace) - Vi...
EXTENSIONS: MARKETPLACE
remote
Remote - SSH 13.9M ★ 4
Open any folder on a remote machine ...
Microsoft Installing
Remote Explorer 3.8M ★ 5
View remote machines for Remote - SS...
Microsoft ✓ Uninstalled Install
Remote Development 3.4M ★ 4.5
An extension pack that lets you open a...
Microsoft Install
Remote - SSH: Editing... 12.5M ★ 4.5
Edit SSH configuration files
Microsoft ✓ Uninstalled Install
Azure Machine Learnin... 1.3M ★ 3.5
This extension is used by the Azure Ma...
Microsoft Install
Remote Repositories 615K ★ 5
Remotely browse and edit git repositor...
Microsoft Install
Extension: Remote - SSH
Remote - SSH v0.94
Microsoft 13,927,699
Open any folder on a remote m...
Installing
Details Feature Contributions Extension Pack

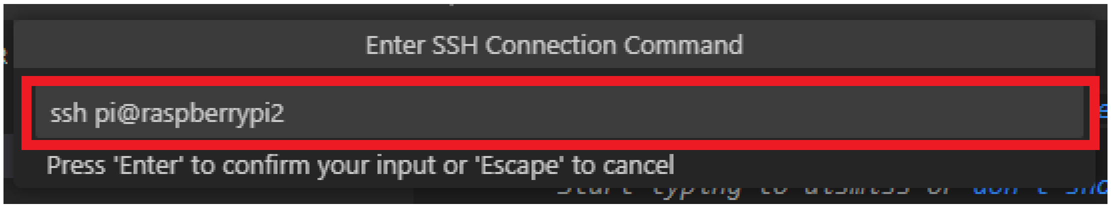
Visual Studio Code Remote - SSH

The **Remote** - **SSH** extension lets you use any remote machine with a SSH server as your development environment. This can greatly simplify development a troubleshooting in a wide variety of situations. You ca

Create a Remote Connection with the Raspberry Pi on VS Code



Then, you'll be asked to run the SSH command to establish a communication with your host.



Enter SSH Connection Command

ssh pi@raspberrypi2

Press 'Enter' to confirm your input or 'Escape' to cancel

At this point, you need to know your Raspberry pi localhost name or IP address and username. For example, in my case:

- localhost: **raspberrypi2**
- IP address: **192.168.1.106**
- username: **pi**

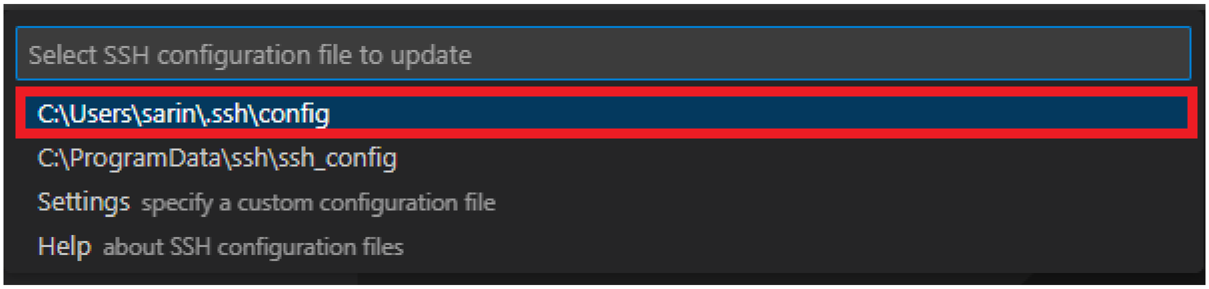
So, I need to write the following command on that field:

```
ssh pi@raspberrypi2
```

Or, if I wanted to use the IP address instead:

```
ssh pi@192.168.1.106
```

Then, press **Enter**. You'll be asked to select an SSH configuration file where VS code will save the SSH settings for that host. You can choose the one under the *Users* folder.



Select SSH configuration file to update

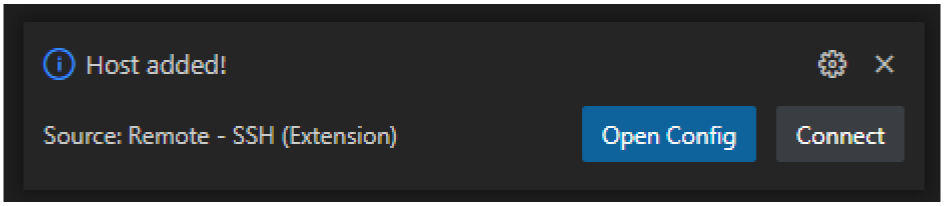
C:\Users\sarin\.ssh\config

C:\ProgramData\ssh\ssh_config

Settings specify a custom configuration file

Help about SSH configuration files

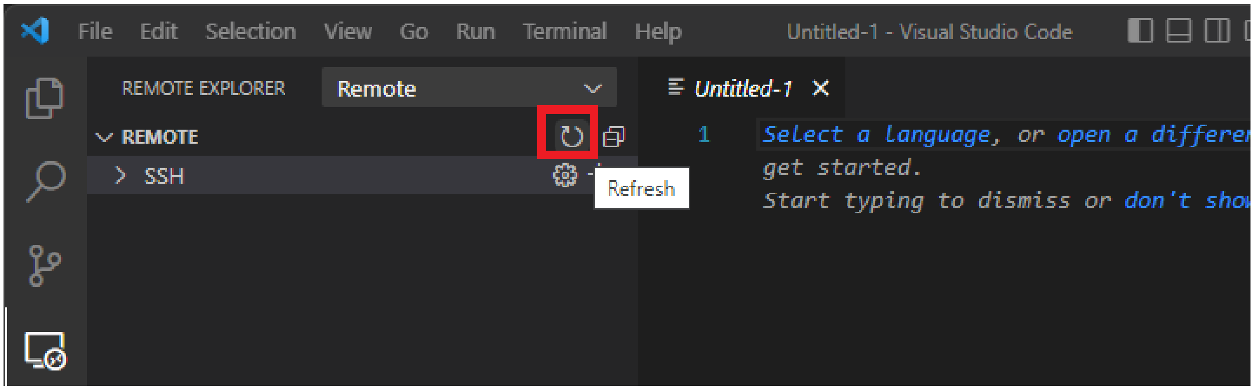
You'll get a message saying the host was added.



Host added!

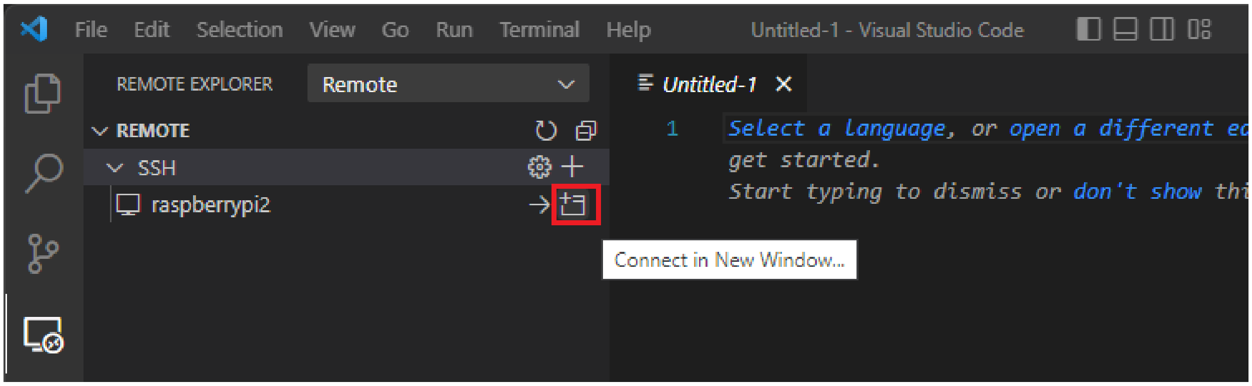
Source: Remote - SSH (Extension)

On the top left corner, click on the **Refresh** icon next to **Remote**.



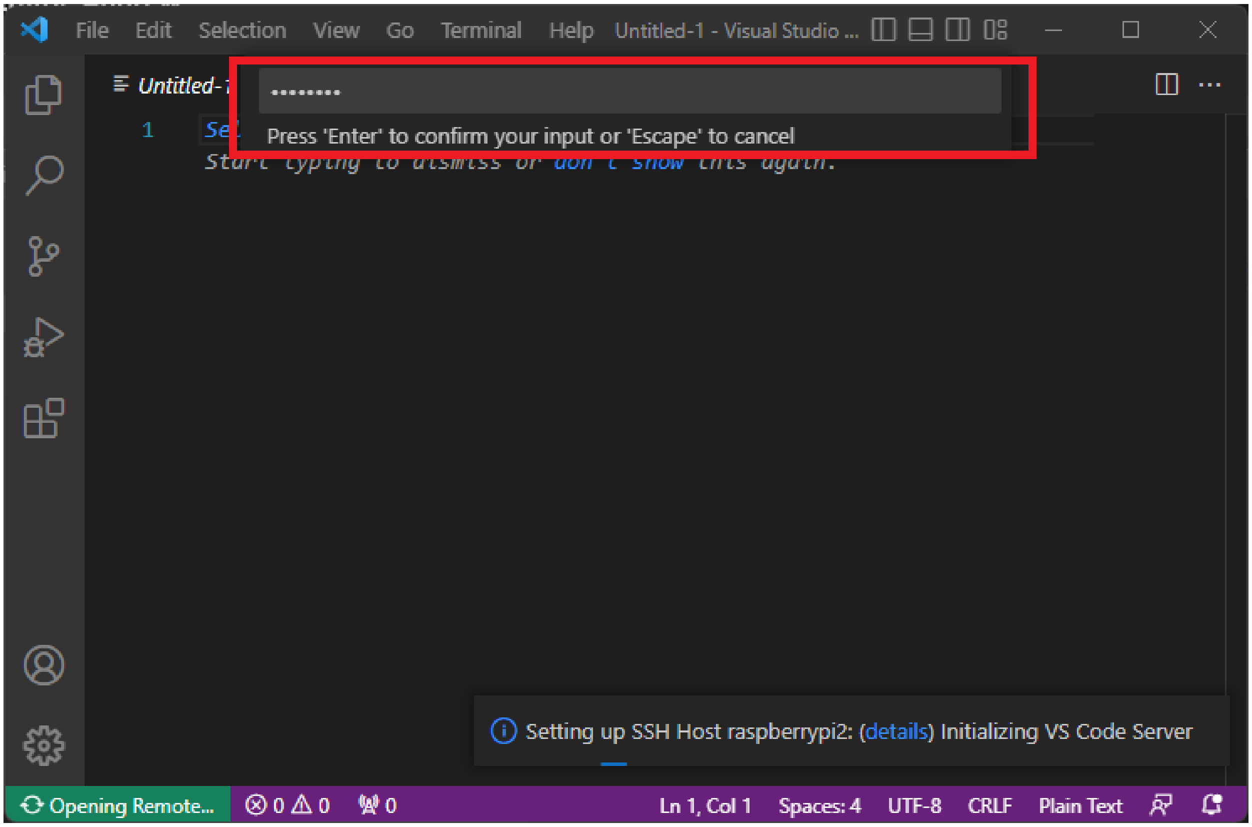
```
Select a language, or open a differen
get started.
Start typing to dismiss or don't show
```

The Raspberry Pi remote host will show up under the **SSH** menu. Then, click on the icon to connect in a new window as shown below.



```
Select a language, or open a different ed
get started.
Start typing to dismiss or don't show thi
```

This will open a new window in VS Code dedicated to that remote machine (the Raspberry Pi). You'll be asked to enter the Raspberry Pi password.



••••••••

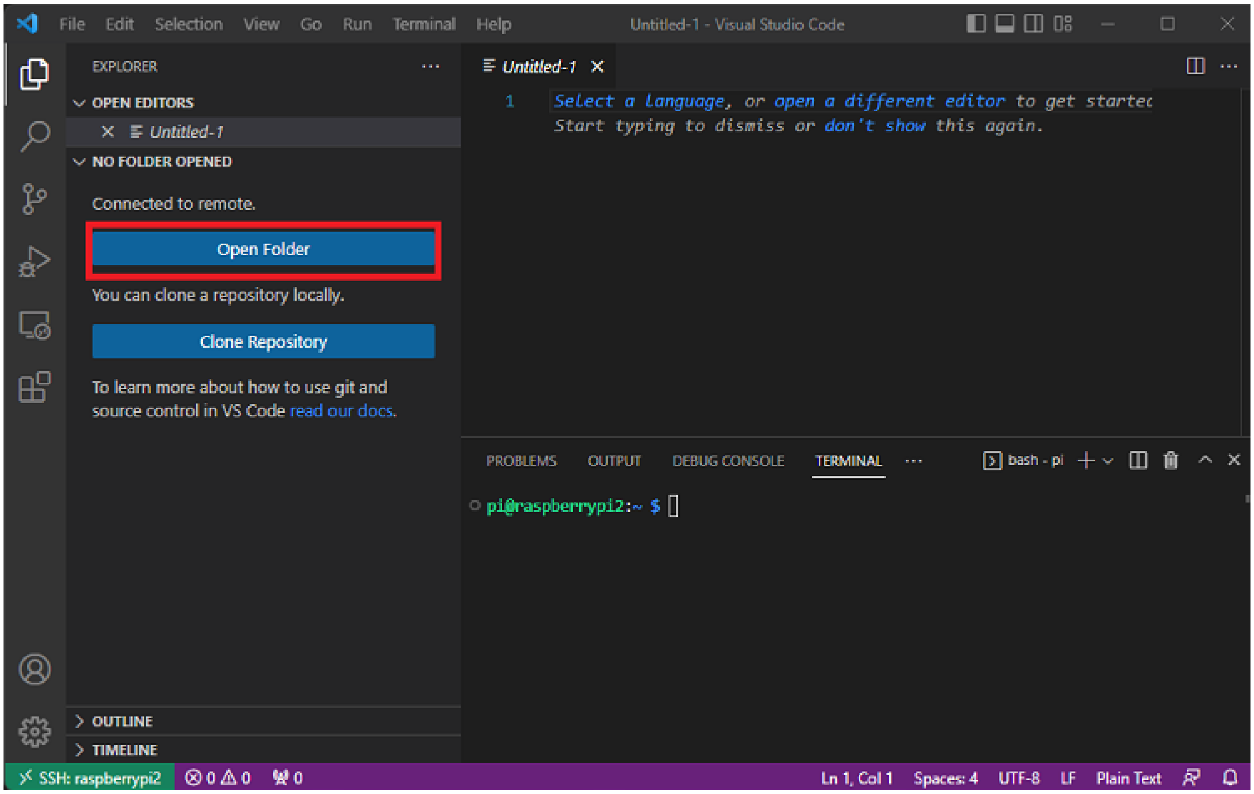
Press 'Enter' to confirm your input or 'Escape' to cancel

Start typing to dismiss or don't show this again.

Setting up SSH Host raspberrypi2: (details) Initializing VS Code Server

Now, you have a remote connection established with your Raspberry Pi. Notice that the Terminal window at the bottom is connected to your Raspberry Pi. Any commands that you run on that Terminal window, will be run on the Raspberry Pi.

You can open Raspberry Pi folders on VS Code to start creating and writing files. Click on the **Open Folder** button to open a folder on your Raspberry Pi where you want to start creating or editing your files.



Connected to remote.

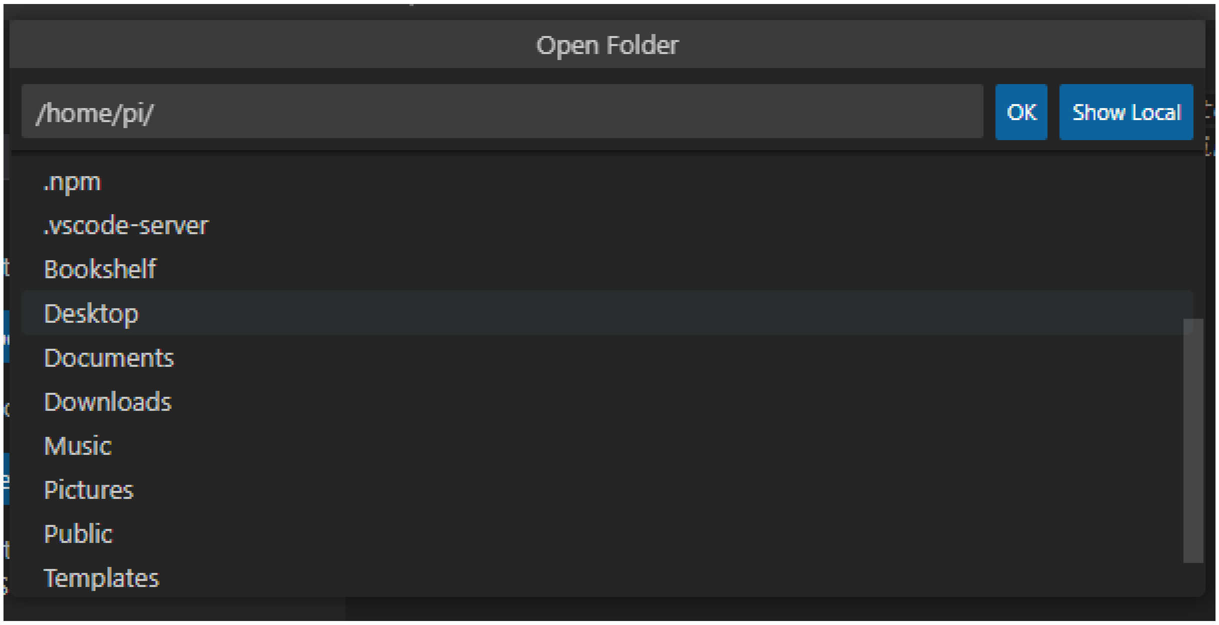
You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

```
Select a language, or open a different editor to get startec
Start typing to dismiss or don't show this again.
```

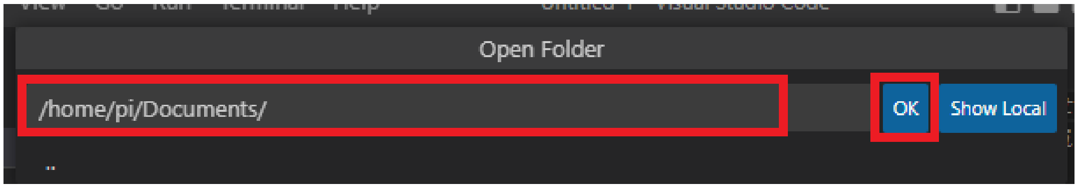
```
pi@raspberrypi2:~ $
```

Choose a folder on your Raspberry Pi where you want to start creating your files.



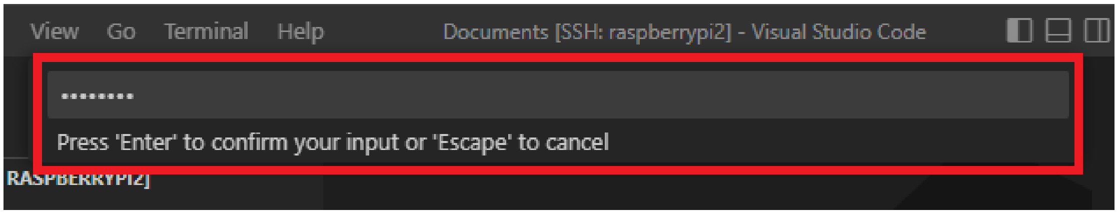
/home/pi/

I chose the *Documents* folder, but you can choose any other location on the Raspberry Pi. The *Desktop* folder might also be a convenient location. After selecting the folder, click **OK**.



/home/pi/Documents/

Insert your Raspberry Pi password again.

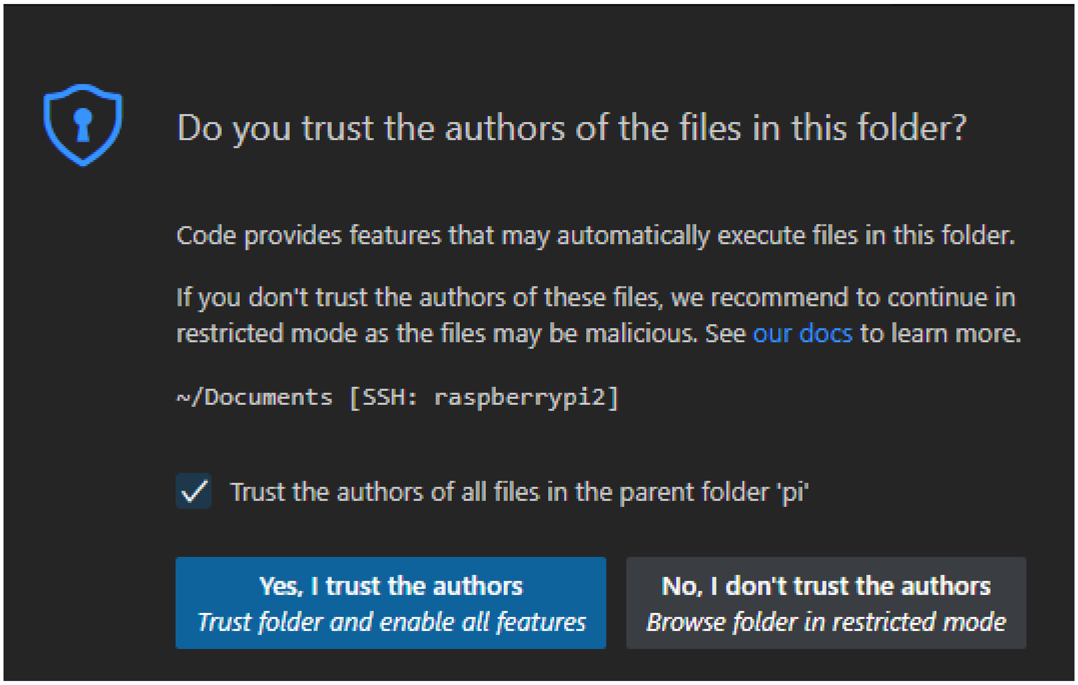


Documents [SSH: raspberrypi2] - Visual Studio Code

••••••••

Press 'Enter' to confirm your input or 'Escape' to cancel

You'll be asked if you trust the authors of the folder. Tick the option *Trust the authors of all files in the parent folder 'pi'*, and click **Yes, I trust the authors** to proceed


Yes, I trust the authors
Trust folder and enable all features
No, I don't trust the authors
Browse folder in restricted mode

Do you trust the authors of the files in this folder?

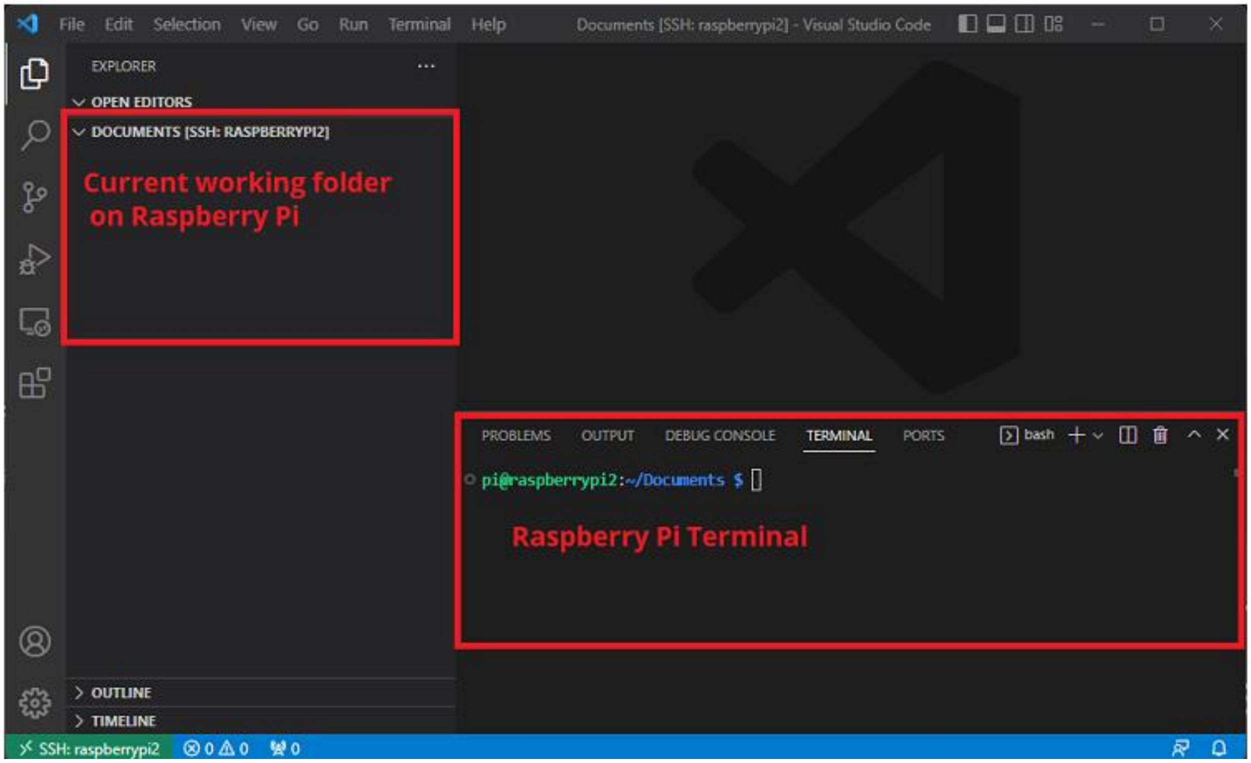
Code provides features that may automatically execute files in this folder.

If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.

```
~/Documents [SSH: raspberrypi2]
```

☑ Trust the authors of all files in the parent folder 'pi'

It will open the selected folder on the left sidebar. Now, you can use the VS Code menus to create new files and/or folders under that folder. To open the Terminal window again, go to **Terminal** > **New Terminal**.


File Edit Selection View Go Run Terminal Help
Documents [SSH: raspberrypi2] - Visual Studio Code
EXPLORER
OPEN EDITORS
DOCUMENTS [SSH: RASPBERRYPI2]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
OUTLINE
TIMELINE
SSH: raspberrypi2
⊗ 0 ⚠ 0
0

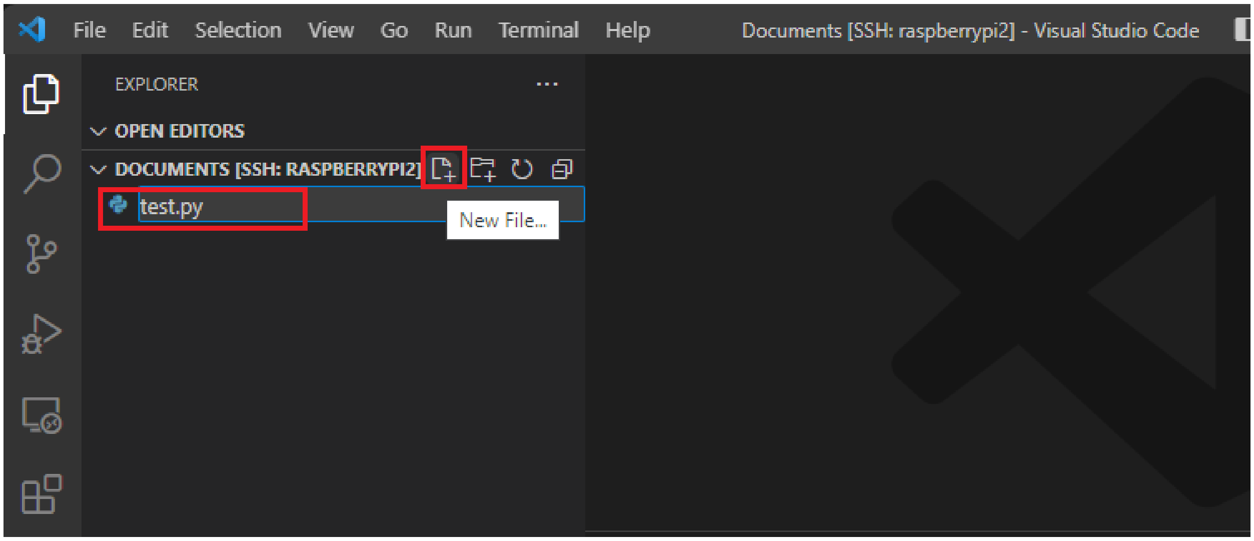
Current working folder on Raspberry Pi

```
pi@raspberrypi2:~/Documents $
```

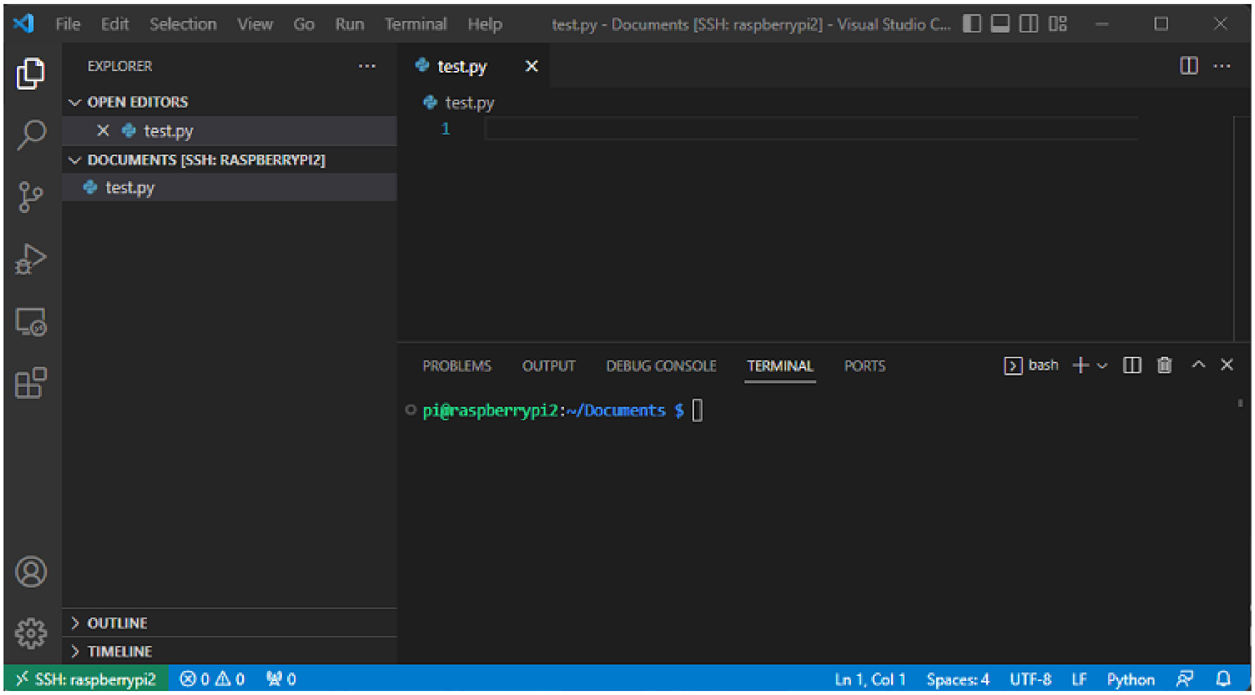
Raspberry Pi Terminal

Create a New Python File in the Raspberry Pi Remotely using VS Code

Once you've selected a working folder, you can create and write a new file in that folder. Hover your mouse over the "**Documents**" tab and click on the **New File…** button to create a new file. For demonstration purposes, you can create a file called *test.py*.


File Edit Selection View Go Run Terminal Help
Documents [SSH: raspberrypi2] - Visual Studio Code
EXPLORER
OPEN EDITORS
DOCUMENTS [SSH: RASPBERRYPI2]
test.py
New File...

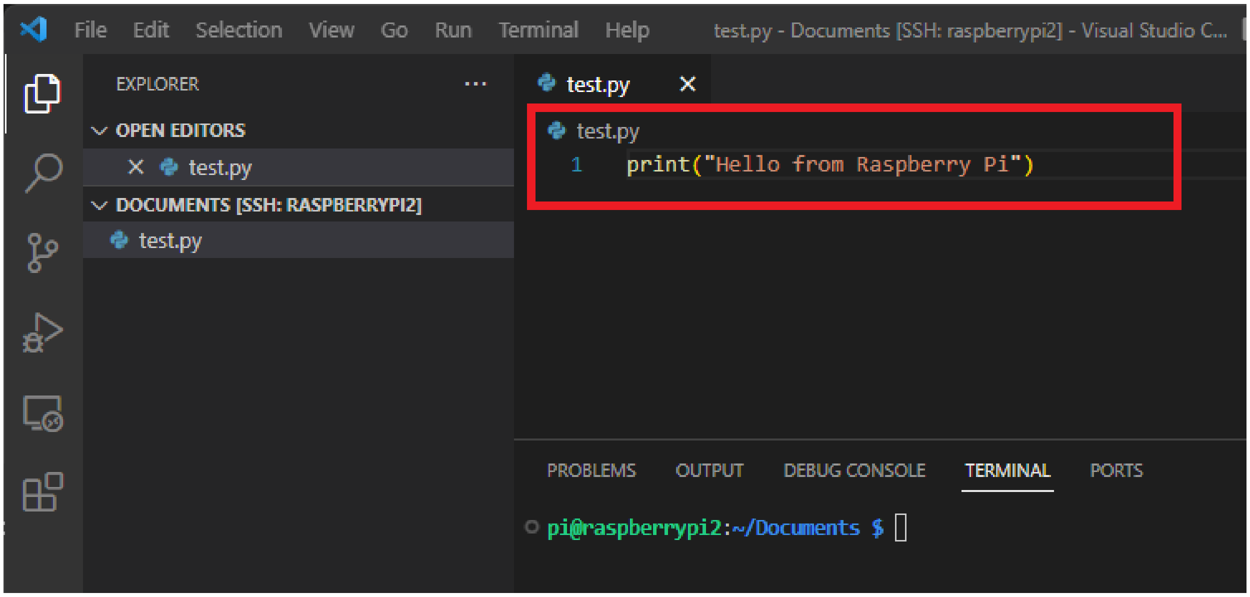
It will create and open a new Python file called *test.py* on the *Documents* folder of the Raspberry Pi filesystem. You can write to that folder using VS Code environment. You can also delete or move that folder from VS Code.


File Edit Selection View Go Run Terminal Help
test.py - Documents [SSH: raspberrypi2] - Visual Studio C...
EXPLORER
OPEN EDITORS
test.py
DOCUMENTS [SSH: RASPBERRYPI2]
test.py
OUTLINE
TIMELINE
test.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
SSH: raspberrypi2
0 0 0
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python

```
pi@raspberrypi2:~/Documents $
```

At this point, you can write your Python code on that file. For demonstration purposes, copy the following line to that file. This will simply print a message on the Terminal window when executed.

```
print("Hello from Raspberry Pi")
```



```
print("Hello from Raspberry Pi")
```

```
pi@raspberrypi2:~/Documents $
```

Save your file. To save your file you can simply press **Ctrl+S** or **Cmd+S**. Now, you have a file called *test.py* on your Raspberry Pi *Documents* folder with that line of code.

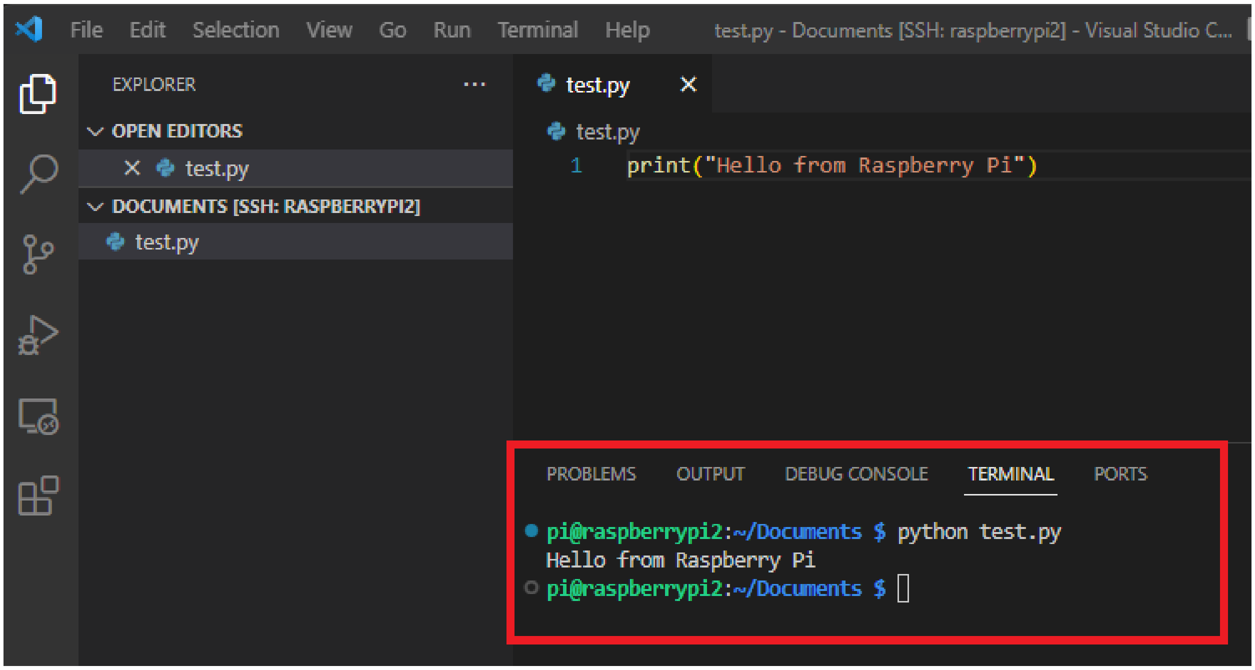
Running a Python File on the Raspberry Pi Remotely using VS Code (SSH)

To run that file simply call the command `python` followed by the name of the file on the terminal window. In this case:

```
python test.py
```

Press **Enter** after typing the command.

You should get “`Hello from Raspberry Pi`” on the Terminal window, which is what that files does (prints a message).

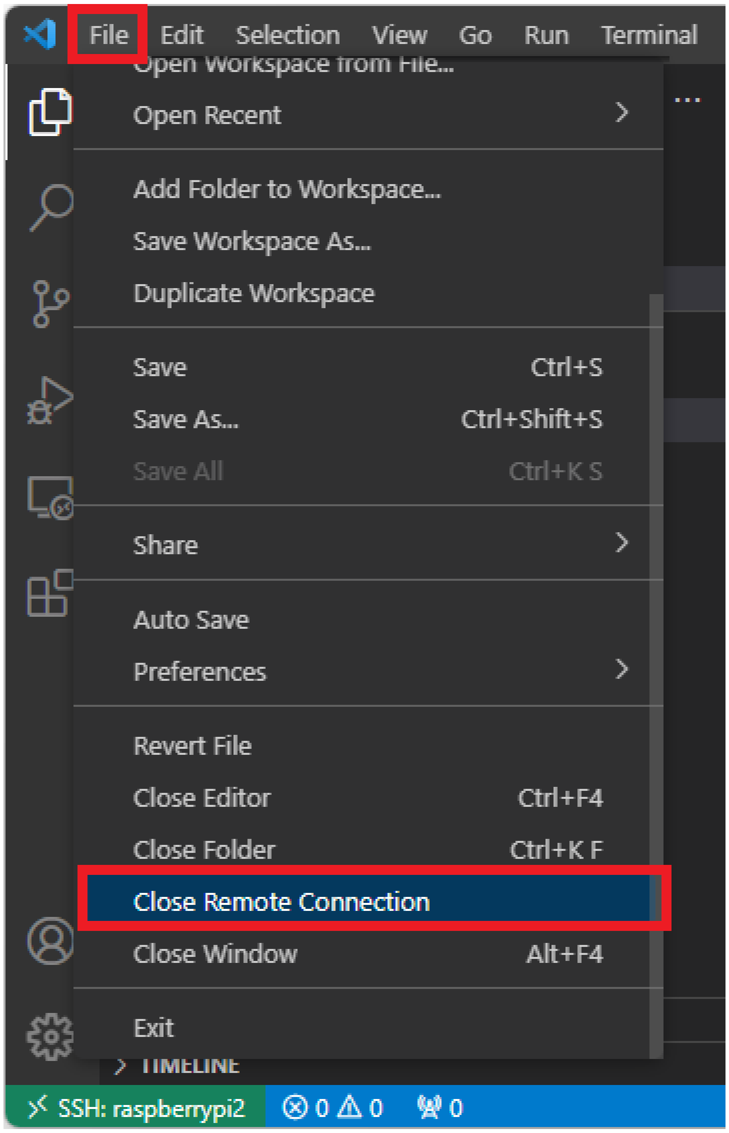


```
print("Hello from Raspberry Pi")
```

```
pi@raspberrypi2:~/Documents $ python test.py
Hello from Raspberry Pi
pi@raspberrypi2:~/Documents $
```

Closing the Remote Connection with Raspberry Pi on VS Code

To close the remote session, go to **File** > **Close Remote Connection**.



Congratulations! Now, you know how to code on Raspberry Pi remotely using VS Code on your computer using the **Remote – SSH** extension.

In this tutorial, you learned a very convenient, practical and easy way to program on your Raspberry Pi remotely via SSH using VS Code on your computer. Using the Remote – SSH extension you can create new files and folders on the Pi, and write and execute code using the command line.

This is our favorite method to program the Raspberry Pi, because you don’t have to manually transfer the files from your computer to the Pi. Using this method, you’re already writing the code on the Pi. Additionally, the Terminal window allows you to execute commands remotely via SSH on the Pi, which is very practical to run your code.

HOW TO OPEN ALL THE UART PORTS IN RASPBERRY PI

Environment

1. Hardware : Raspberry Pi 4 Model B (BCM2177*) [ref]
2. Operation System : Respbian Linux 11
3. Arch : armv7l

There are 2 types of UART interface existed on Raspberry Pi, `mini UART` & `PL011 UART`.
By default, `/dev/ttyS0` maps to `mini UART`; on the other hand, `/dev/ttyAMA0` maps to `PL011 UART`.

Software Setup

Open a terminal and change the file `config.txt` in `/boot` directory with root privilege.

```
$ sudo su
# vim /boot/config.txt
```

add following lines at the end of the file

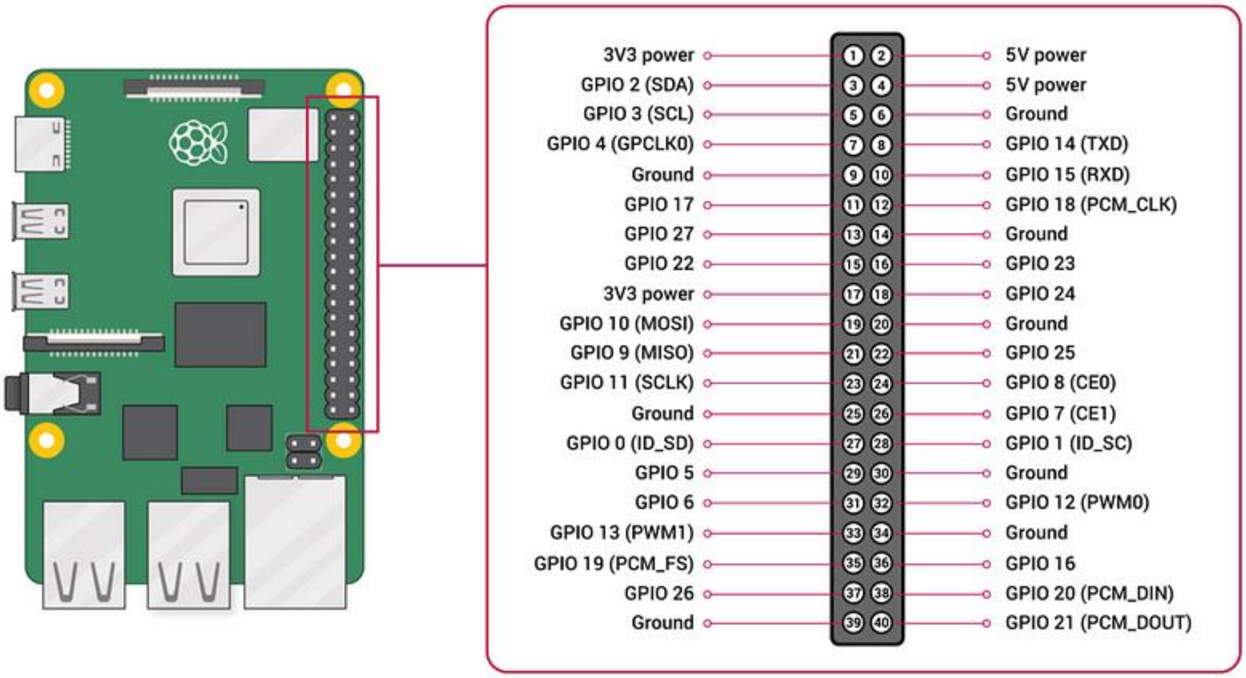
```
# enable serial interface
enable_uart=1
dtoverlay=uart0
dtoverlay=uart1
dtoverlay=uart2
dtoverlay=uart3
dtoverlay=uart4
dtoverlay=uart5
```

save & exit, then reboot !

Check system open the serial by command `$ ls -al /dev/ttyAMA*`

```
$ ls -al /dev/ttyAMA*
crw-rw---- 1 root dialout 204, 64 Dec 16 16:01 /dev/ttyAMA0
crw-rw---- 1 root dialout 204, 65 Dec 16 16:01 /dev/ttyAMA1
crw-rw---- 1 root dialout 204, 66 Dec 16 16:01 /dev/ttyAMA2
crw-rw---- 1 root dialout 204, 67 Dec 16 16:01 /dev/ttyAMA3
crw-rw---- 1 root dialout 204, 68 Dec 16 16:01 /dev/ttyAMA4
```

Let’s take a look into the Raspberry Pi 4 Model B GPIO Pinout:


3V3 power 1 2 5V power
GPIO 2 (SDA) 3 4 5V power
GPIO 3 (SCL) 5 6 Ground
GPIO 4 (GPCLK0) 7 8 GPIO 14 (TXD)
Ground 9 10 GPIO 15 (RXD)
GPIO 17 11 12 GPIO 18 (PCM_CLK)
GPIO 27 13 14 Ground
GPIO 22 15 16 GPIO 23
3V3 power 17 18 GPIO 24
GPIO 10 (MOSI) 19 20 Ground
GPIO 9 (MISO) 21 22 GPIO 25
GPIO 11 (SCLK) 23 24 GPIO 8 (CE0)
Ground 25 26 GPIO 7 (CE1)
GPIO 0 (ID_SD) 27 28 GPIO 1 (ID_SC)
GPIO 5 29 30 Ground
GPIO 6 31 32 GPIO 12 (PWM0)
GPIO 13 (PWM1) 33 34 Ground
GPIO 19 (PCM_FS) 35 36 GPIO 16
GPIO 26 37 38 GPIO 20 (PCM_DIN)
Ground 39 40 GPIO 21 (PCM_DOUT)

GPIO Pinout : https://www.raspberrypi.com/documentation/computers/os.html#gpio-and-the-40-pin-header

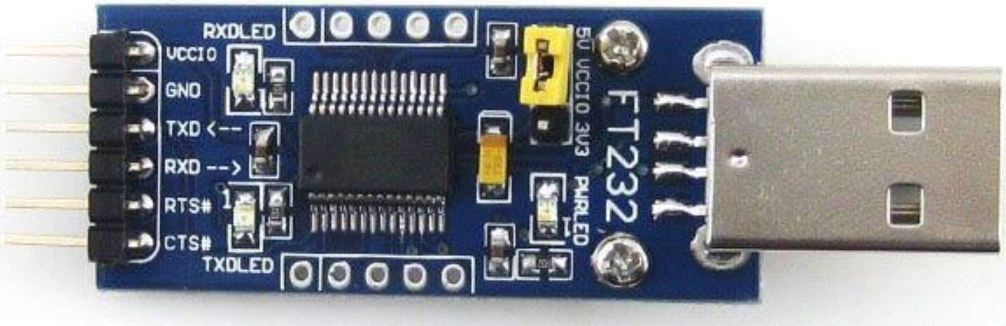
Hardware Setup

Raspberry Pi Pin pair with uart :

```
           TXD        RXD        Communication Port
uart0 :  GPIO 14    GPIO 15        /dev/ttyAMA0
uart1 :  GPIO 0     GPIO 1         /dev/ttyAMA1
uart2 :  GPIO 4     GPIO 5         /dev/ttyAMA2
```

```
uart3 :  GPIO 8     GPIO 9          /dev/ttyAMA3
uart4 :  GPIO 12    GPIO 13         /dev/ttyAMA4
```

PC setup with USB-to-UART Adapter :



FT232 : https://www.waveshare.com/ft232-usb-uart-board-type-a.htm

Please make sure **GND** is connected correctly !

USB-to-UART `TXD` connect to Raspberry Pi uart `RXD` GPIO Pin

USB-to-UART `RXD` connect to Raspberry Pi uart `TXD` GPIO Pin

Take uart3 for example

```
USB-to-UART TXD <--> GPIO 8
USB-to-UART RXD <--> GPIO 9
USB-to-UART GND <--> GND
```

Implementation

Once finish setup, open a serial console on PC (ex. Putty) with correct COM Port (ex. COM3) & Baud Rate set to `115200`

Then open a terminal in Raspberry Pi, send message from CLI to PC :

```
$ echo 'hello' > /dev/ttyAMA3
```

This command should send the string `hello` to PC via `/dev/ttyAMA3` , then the string will be parsed by PC, finally display on serial console.

Hope this article can help those makers connect more devices via multiple UART interfaces !! Have Fun !!

Please feel free to contact me by mail anytime if there is anything I could help!
Email : jason19970210@gmail.com

Raspberry Pi

Uart

HOW TO SET THE STATIC IP IN RASPBERRY PI

Once in a while, your device will benefit or will need to setup a static IP address. Here is a short how-to. Tested on Jessie, but should work for other Raspbian releases as well.

This recipe assumes you are already connected to you Pi over SSH, but if you wish the use the Terminal from your desktop, go ahead - same thing.

It is a good idea to plan your network settings so we also assume you have a fresh setup.

Set Static IP Address

Start by editing the **dhcpcd.conf** configuration file:

```
sudo nano /etc/dhcpcd.conf
```

Scroll to the end of the file and add one or both configuration sections. The first section sets the static address for a wired connection - **eth0**. The second section for wireless - **wlan0**. Choose one or both options. You need to of course modify your network addresses.

wired connection

```
interface eth0
```

```
static ip_address=192.168.1.10/24
```

```
static routers=192.168.0.1
```

```
static domain_name_servers=192.168.1.1
```

wireless connection

```
interface wlan0
```

```
static ip_address=192.168.1.10/24
```

```
static routers=192.168.1.1
```

```
static domain_name_servers=192.168.1.1
```

What do these names mean?

- interface = This defines which network interface you are setting the configuration for.

- static ip_address = This is the IP address that you want to set your device to. (Make sure you leave the /24 at the end)
- static routers = This is the IP address of your gateway (probably the IP address or your router)
- static domain_name_servers = This is the IP address of your DNS (probably the IP address of your router). You can add multiple IP addresses here separated with a single space.

Finally, close the editor and save changes by pressing **ctrl-X**, then typing **Y** and hitting

Reboot the PI:

```
sudo Reboot
```

Tho check your settings once reconnected, type

```
ifconfig
```