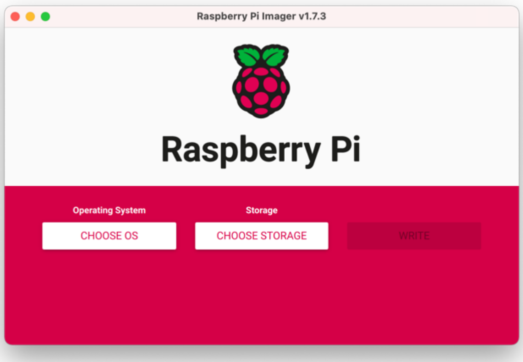
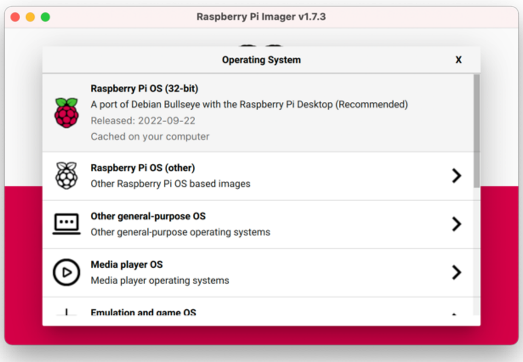
Raspberry PI installation

Connect the NIC SD to the computer.

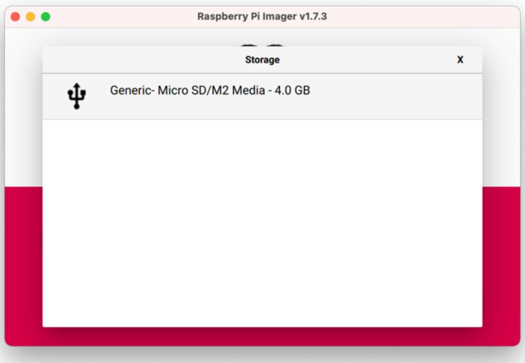
Download software installation program:



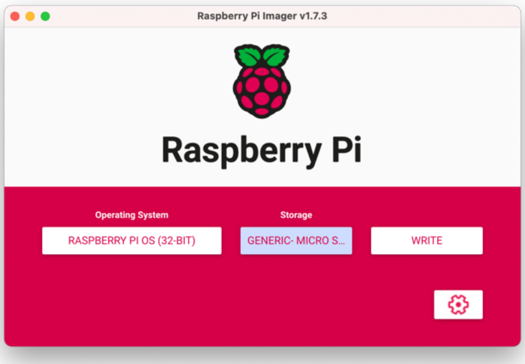
Choose the operation system:


Raspberry Pi Imager v1.7.3
Operating System
X
Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Released: 2022-09-22
Cached on your computer
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

Connect the NIC SD at the computer:

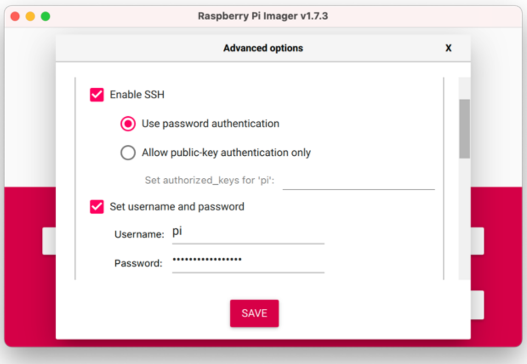

Raspberry Pi Imager v1.7.3
Storage
X
Generic- Micro SD/M2 Media - 4.0 GB

Almost ready:



Configure with ⚙ :

- hostname
- enable the SSH and configure username and password



[✓] Enable SSH

[● Use password authentication]

○ Allow public-key authentication only

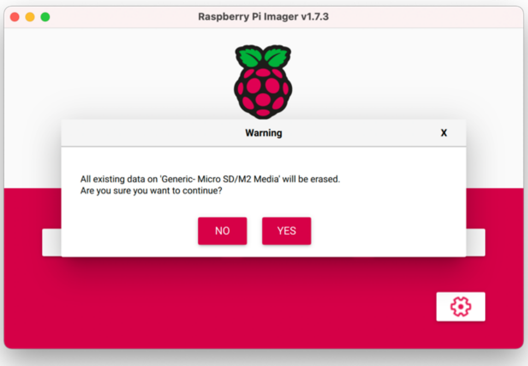
Set authorized_keys for 'pi':

[✓] Set username and password

Username: pi

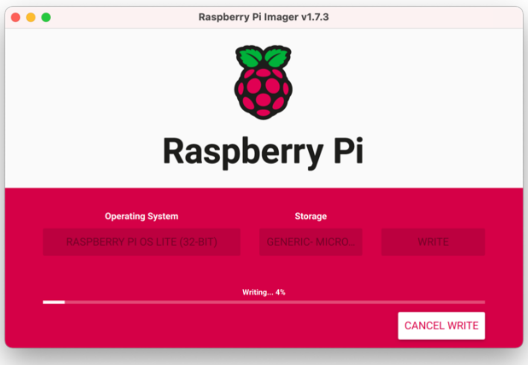
Password: ••••••••••••••••••

Save the options, click on “WRITE” and confirm.



All existing data on 'Generic- Micro SD/M2 Media' will be erased.
Are you sure you want to continue?

Wait until the end of the verification:



Now transfer the SD on the Raspberry PI, connect it to the ethernet network ed turn it on.

Wait until the OS starts steadily.

Now you can connect with a monitor and keyboard to the Raspberry PI, or remotely via SSH.
In the first case, it is possible to retrieve the IP address given by the DHCP with the following command:

ifconfig -a eth0

```
pi@raspberrypi:~ $ ifconfig -a eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.22  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::d24:1a07:7903:d7dc  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:6e:26:36  txqueuelen 1000  (Ethernet)
        RX packets 800  bytes 136232 (133.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 218  bytes 35881 (35.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

In case you do not have a monitor, to retrieve the IP address given by the DHCP look at the *leased* assigned by the DHCP server (by looking at the log files of the modem/router)

To connect:
ssh pi@192.168.0.22

```
sh-3.2$ ssh pi@192.168.0.22
pi@192.168.0.22's password:
Linux raspberrypi 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST 2022 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 Dec  6 14:25:48 2022 from 192.168.0.28
pi@raspberrypi:~ $
```

To conclude the installation, we upload the software by downloading first the packets list:
sudo apt-get update

```
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Get:2 http://archive.raspberrypi.org/debian bullseye InRelease [23.6 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf Packages [13.2 MB]
Get:4 http://archive.raspberrypi.org/debian bullseye/main armhf Packages [311 kB]
Fetched 13.6 MB in 17s (777 kB/s)
Reading package lists... Done
```

Then upload the obsoletes ones:
sudo apt-get upgrade

```
pi@raspberrypi:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  bind9-host bind9-libs bluez bluez-firmware dbus dhcpcd5 firmware-atheros firmware-brcm80211
firmware-libertas
  firmware-misc-nonfree firmware-realtek isc-dhcp-client isc-dhcp-common libbluetooth3 libc-bin libc-dev-
bin libc-devtools
  libc-l10n libc6 libc6-dbg libc6-dev libcamera-apps-lite libcamera0 libdbus-1-3 libexpat1 libgssapi-krb5-2
libk5crypto3 libkms++0
  libkrb5-3 libkrb5support0 libksba8 libntfs-3g883 libuv1 libxml2 linux-libc-dev locales ntfs-3g python3-
kms++ python3-libcamera
  python3-picamera2 raspberrypi-bootloader raspberrypi-kernel raspberrypi-sys-mods raspi-config rpi-
eeprom vcdbg
46 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 198 MB of archives.
After this operation, 69.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
...
Get:1 http://archive.raspberrypi.org/debian b
```

FIREWALL SMART HOME INSTALLATION

……….

Firstly, configure the WiFi network interface, since my Raspberry PI model only has one Ethernet network interface.
Once the WiFi USB adapter is connected it is possible to see the version with the command:
dmesg

```
pi@raspberrypi:~ $ dmesg
...
[  939.267369] usb 1-1.4: new high-speed USB device number 5 using dwc_otg
[  939.541086] usb 1-1.4: New USB device found, idVendor=148f, idProduct=2573, bcdDevice= 0.01
[  939.541132] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[  939.541151] usb 1-1.4: Product: 802.11 bg WLAN
[  939.541165] usb 1-1.4: Manufacturer: Ralink
[  939.977376] usb 1-1.4: reset high-speed USB device number 5 using dwc_otg
[  940.431886] ieee80211 phy0: rt2x00_set_chip: Info - Chipset detected - rt: 2573, rf: 0002, rev: 000a
[  940.432535] ieee80211 phy0: Selected rate control algorithm 'minstrel_ht'
[  940.441114] usbcore: registered new interface driver rt73usb
[  940.774190] ieee80211 phy0: rt2x00lib_request_firmware: Info - Loading firmware file 'rt73.bin'
[  940.775793] ieee80211 phy0: rt2x00lib_request_firmware: Info - Firmware detected - version: 1.7
```

As we can see the version is verified and supported.

Indeed the Ethernet interface is listed with the others interfaces. To verify, use the command:
ifconfig -a

```
pi@raspberrypi:~ $ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.22  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::d24:1a07:7903:d7dc  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:6e:26:36  txqueuelen 1000  (Ethernet)
        RX packets 156487  bytes 220638891 (210.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29119  bytes 2835426 (2.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 10  bytes 1600 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 1600 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:e0:4d:82:4d:3b  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

With the following command it is possible to see the functionalities of the WiFi NIC:
iw list

```
pi@raspberrypi:~ $ iw list
Wiphy phy0
        wiphy index: 0
        max # scan SSIDs: 4
...
        Supported interface modes:
                 * IBSS
                 * managed
                 * AP
```

```
                * AP/VLAN
                * monitor
                * mesh point
...
```

As we can see, the AP (Access Point) mode is supported and we need it.

Check if the NIC is not blocked with the command:
rfkill list

```
pi@raspberrypi:~ $ rfkill list
0: phy0: Wireless LAN
        Soft blocked: no
          Hard blocked: no
```

Now we can activate the AP mode via hostapd, that has to be installed with the following command:
sudo apt-get install hostapd

```
pi@raspberrypi:~ $ sudo apt-get install hostapd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  hostapd
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 665 kB of archives.
After this operation, 1,817 kB of additional disk space will be used.
Get:1 http://ftp.arnes.si/mirrors/raspbian/raspbian bullseye/main armhf hostapd armhf 2:2.9.0-21 [665 kB]
Fetched 665 kB in 3s (260 kB/s)
Selecting previously unselected package hostapd.
(Reading database ... 43650 files and directories currently installed.)
Preparing to unpack .../hostapd_2%3a2.9.0-21_armhf.deb ...
Unpacking hostapd (2:2.9.0-21) ...
Setting up hostapd (2:2.9.0-21) ...
Created symlink /etc/systemd/system/multi-user.target.wants/hostapd.service →
/lib/systemd/system/hostapd.service.
Job for hostapd.service failed because the control process exited with error code.
See "systemctl status hostapd.service" and "journalctl -xe" for details.
Created symlink /etc/systemd/system/hostapd.service → /dev/null.
Processing triggers for man-db (2.9.4-2) ...
```

Create and configure hostapd with an editor command, for example **vi** or **nano**:
sudo vi /etc/hostapd/hostapd.conf

Insert the following configuration:

```
# Band: a = 5g (a/n/ac), g = 2g (b/g/n)
hw_mode=g
# Channel
channel=1
# Country code
country_code=IT
# security
```

```
auth_algs=1
wmm_enabled=1

wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP

# WiFi interface
interface=wlan0
ssid=SmartHome
wpa_passphrase=ChangeMeIsTooSimple
```

For the different options have a look at:
/usr/share/doc/hostapd/examples/hostapd.conf

Rehabilitate the service:
sudo systemctl unmask hostapd
sudo systemctl enable hostapd

Configure the wlan0 interface's parameters with the editor (vi or others):
sudo vi /etc/network/interfaces

Add the following configuration:

```
# Physical Wireless
auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
   address 192.168.2.1
   network 192.168.2.0
   netmask 255.255.255.0
   broadcast 192.168.2.255
```

Now the DHCP server has to be configured for the SmartHome network.
For this we install and use dnsmasq:
sudo apt-get install dnsmasq

Configure the parameter with an editor:
sudo vi /etc/dnsmasq.conf

Add the following configuration:

```
#SMART HOME configuration
domain=smarthome.local
# range for dinamic IP
dhcp-range=192.168.2.100,192.168.2.199,255.255.255.0,48h
# For static  IP (MAC Address)
#dhcp-host=xx:xx:xx:xx:xx,hostname,192.160.0.x,lease-time
# other configuration
dhcp-option=option:router,192.168.2.1
Reading package lists... Done
```

We now need to enable the routing between eth0 and wlan0 interfaces.

Edit the following file with the command vi or other editor:
sudo vi /etc/sysctl.conf

Uncomment the following line:
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

Now restart the Raspberry:
sudo reboot

When it has restarted, check if the WiFi connection works.
Pay attention: the Internet connection is not working yet because we still have to configure the NAT on the eth0 interface.

Enabling the NAT

In order to let the modem/router of the main network know the new SmartHome network, we should configure one static route. Instead, we prefer to masquerade the addresses that leave the eth0 interface of our firewall.

To enable the NAT functionality, use the following command:
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Then, to allow devices connected to the SmartHome to create connections with Internet and the main network use the command:
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT

And to deny connections originated from Internet or the main network to the SmartHome network use:
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT

So now, the connection has to be initialized from the SmartHome network and all the others will be discarded.

Firewall configuration:

In order to avoid communication between Home network and IoT network we set the following rules:
sudo iptables -A FORWARD -d 192.168.0.0/24 -j DROP
sudo iptables -A FORWARD -s 192.168.0.0/24 -j DROP

Then to deny packets from outside the network with the sensor as destination:
sudo iptables -A FORWARD ! -s 192.168.2.0/24 -d 192.168.2.45/32 -j DROP

and to deny packets from the sensor to the Internet:
sudo iptables -A FORWARD -s 192.168.2.45/32 ! -d 192.168.2.0/24 -j DROP