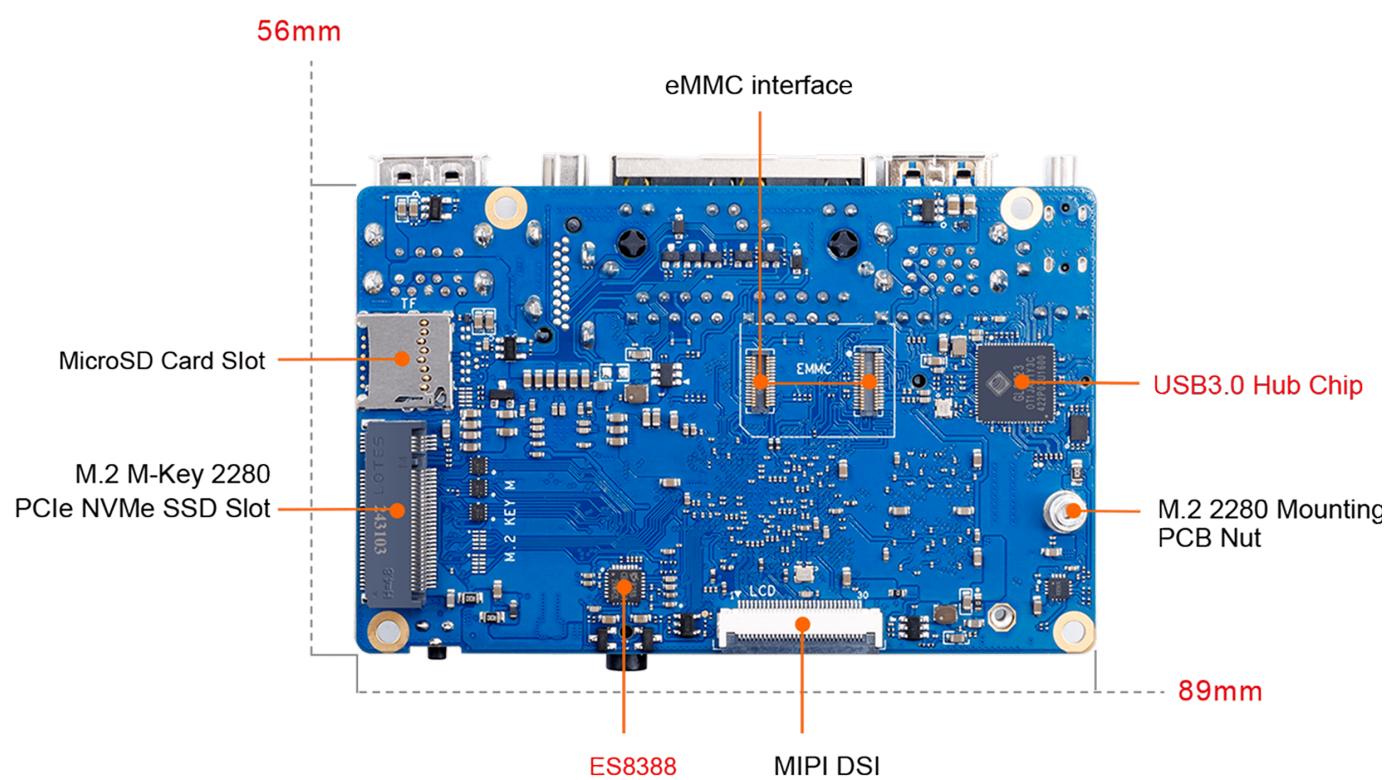
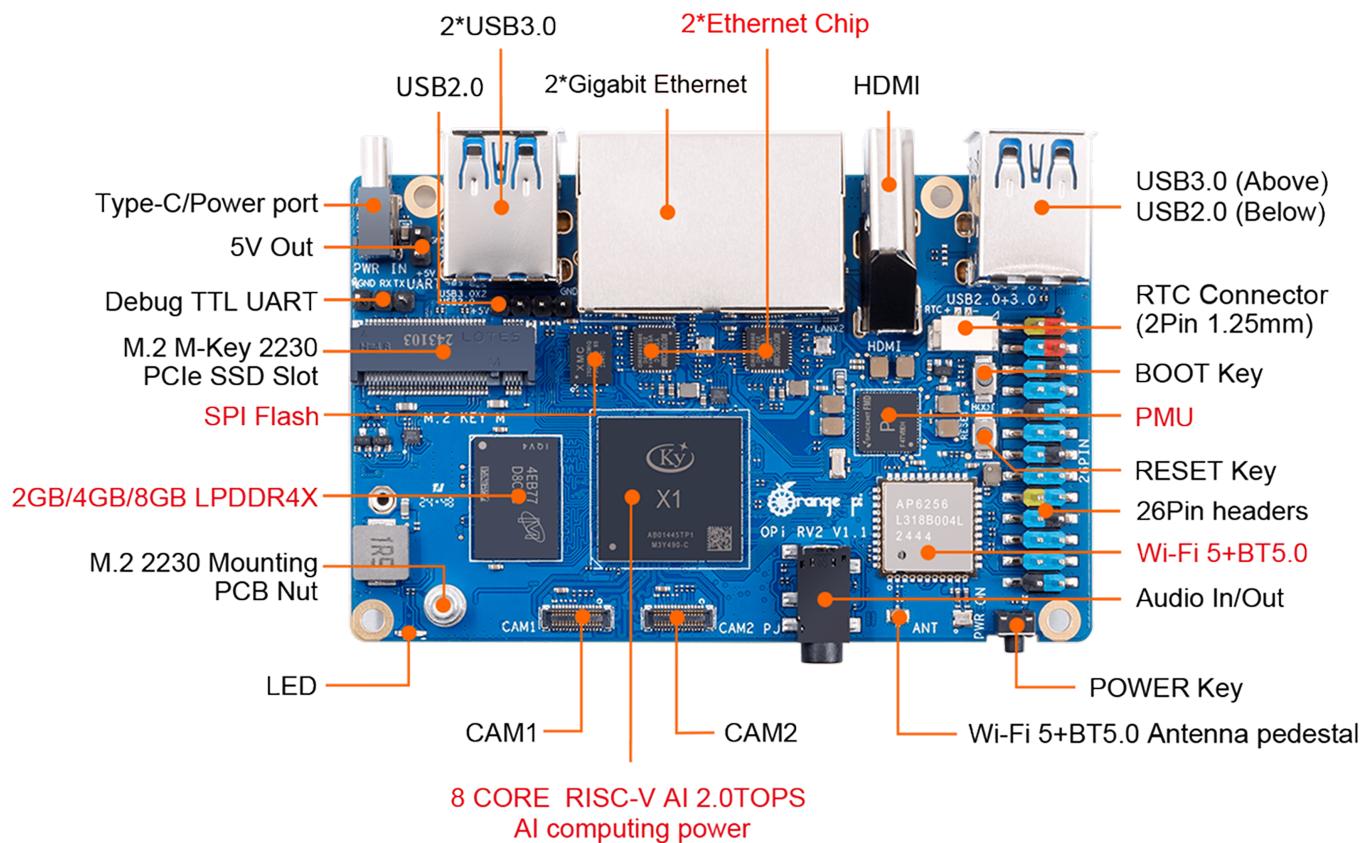


Orange Pi RV2: Complete Installation & First Boot Tutorial



Hardware Parameters

CPU	Ky X1 8-core RISC-V AI CPU 2TOPS AI power
Memory	LPDDR4X: 2GB/4GB/ 8GB
Storage	eMMC module (optional): 16GB/32GB/64GB/128GB optional SPI FLASH: 128Mb (default), 256Mb optional 2*M.2 M-Key SOCKET: PCIe2.0 2 Lane NVMe SSD TF Card: Supports SDIO3.0 TF Card
Wi-Fi + Bluetooth	Wi-Fi + Bluetooth 2-in-1 Module: AP6256 Wi-Fi5.0+BT 5.0, BLE
Ethernet interface	2* Gigabit Ethernet, PHY Chip: YT8531C-CA
Display	Supports dual-screen heterodyne display, up to 1920x1440@60fPS 1 HDMI2.0 1* 4-lane MIPI DSI
Camera	2x 4-lane MIPI CSI camera interface
USB	1USB2.0 HOST or DEVICE 3USB3.0 HOST 1*USB2.0 HOST (4Pin header)
Audio	3.5mm headphone jack, audio input/output, HDMI audio output
Buttons	1* BOOT, 1*RESET, 1 *PWR ON
RTC	2Pin spare battery connector (Pitch=1.25mm)
26Pin	26Pin function expansion interface, supports the following interface types: GPIO, UART, I2C, SPI, PWM, etc.
DEBUG	3Pin debug serial port, 3.3V level
Power input/output	Power connector: 2-pin. Maximum 5V, 1A output, suitable for connecting a fan.
Power supply	Type-C 5V 5A DC IN
PCB size	89x61.6 mm
Supported OS	Ubuntu 24.04

1. What You Need

- Orange Pi RV2 board
 - MicroSD card (16GB or larger, high quality recommended)
 - SD card reader for your PC
 - Power supply (5V/5A USB-C, non-PD)
 - HDMI monitor, keyboard, and mouse (for desktop setup)
 - USB to TTL adapter (for serial debugging, optional)
 - Ethernet cable or Wi-Fi access
 - (Optional) M.2 SSD for storage or OS boot
-

2. Download the Operating System Image

- Go to the [Orange Pi RV2 downloads page](#) or the official GitHub/Google Drive links.
 - Download the image:
 - **Ubuntu Noble Desktop** (with GUI)
-

3. Flash the SD Card

- Insert your SD card into your PC.
 - Use [Balena Etcher](#) or Raspberry Pi Imager to flash the downloaded `.img` file onto the SD card.
 - Wait for the process to complete and safely eject the SD card.
-

4. First Boot & Hardware Setup

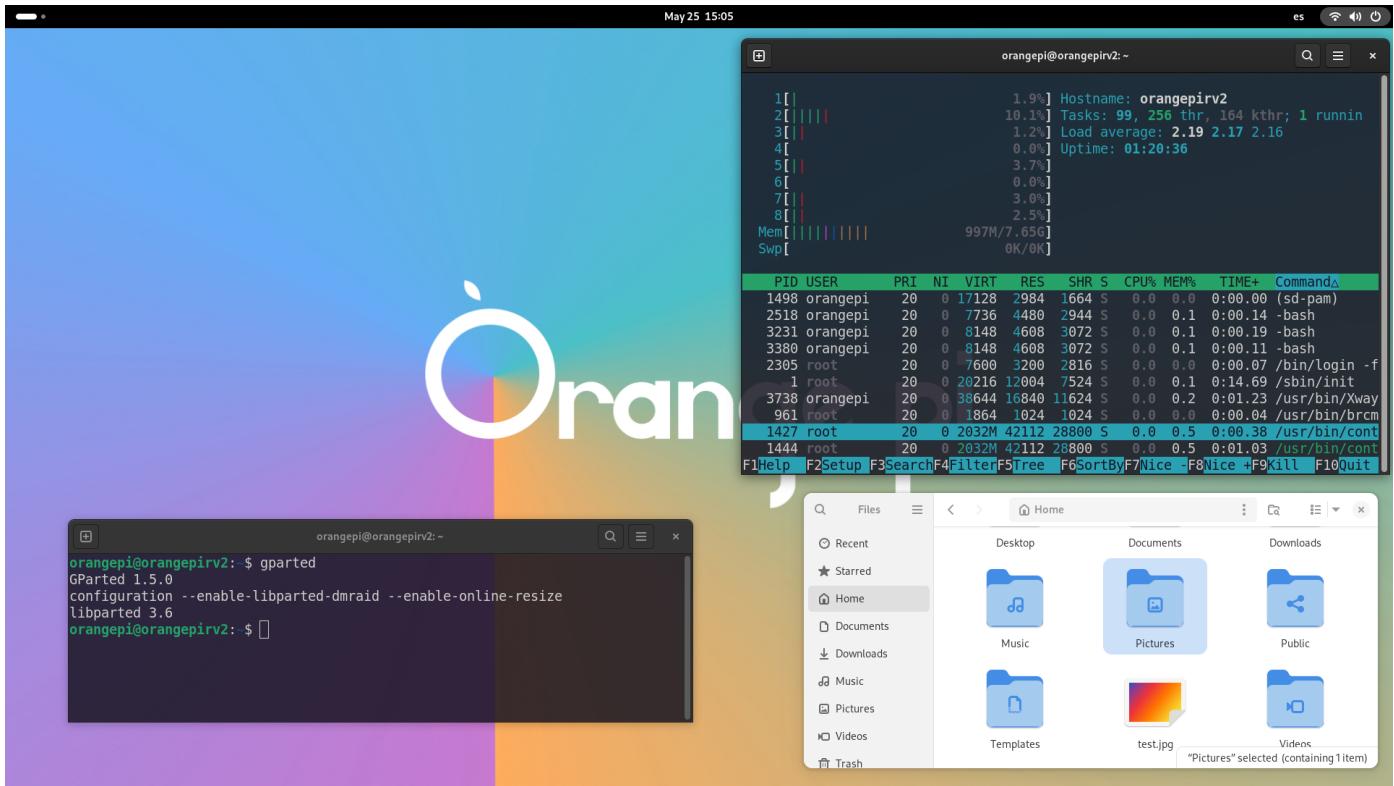
- Insert the SD card into the Orange Pi RV2.
 - Connect monitor, keyboard, and mouse (for desktop), or prepare your **USB-to-TTL adapter** for serial debugging (for headless/server).
 - Plug in Ethernet or prepare for Wi-Fi setup.
 - Connect the USB-C power supply (5V/5A, fixed voltage, not PD negotiation).
-

5. Accessing the System

A. Desktop Version

- The system will boot to the Ubuntu desktop.
- Log in with the default credentials (usually `orangepi / orangepi`).

- You can now use the graphical interface, connect to Wi-Fi via the GUI, or open a terminal for further setup.



B. Server Version (Headless)

- Connect the USB-to-TTL adapter to the UART pins (GND, RX, TX) on the board and your PC.
- Use a serial terminal, such as PuTTY or minicom, at a baud rate of 115200.
- Log in with the default credentials.
- Alternatively, connect via Ethernet and use SSH once the board is on the network.

6. Network Configuration

A. Wi-Fi (via terminal)

```
nmcli device wifi list
nmcli device wifi connect "YOUR_SSID" password "YOUR_PASSWORD"
```

- Check your IP address:

```
ip a
```

B. Ethernet

- Plug in the cable; DHCP will assign an IP automatically.

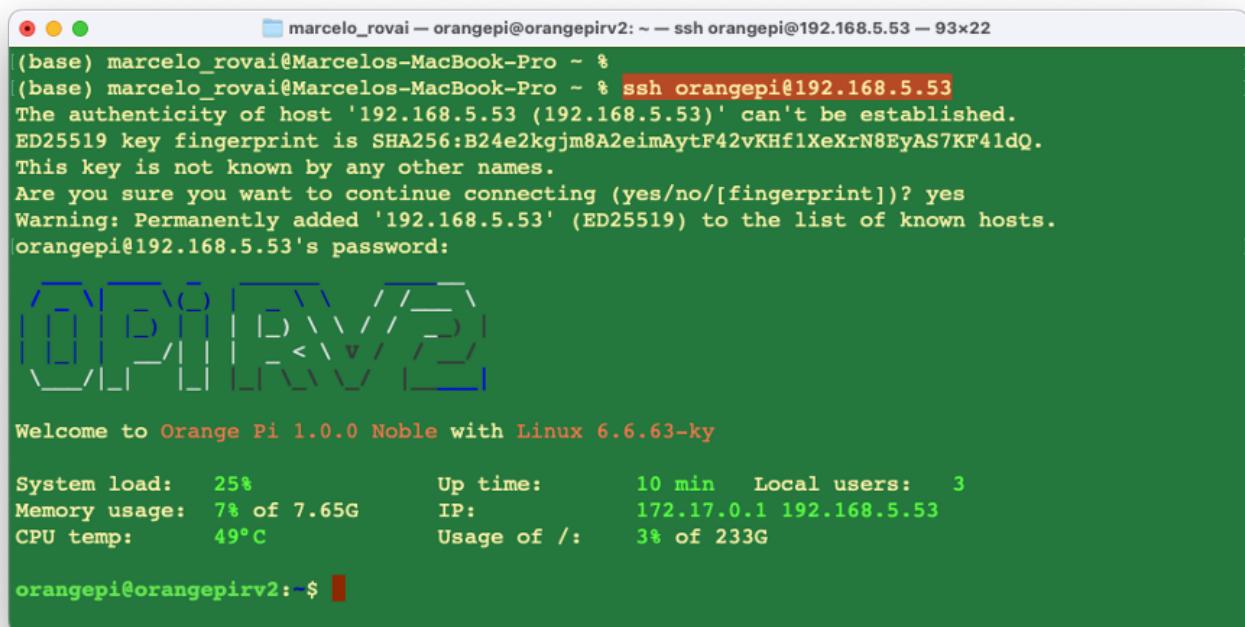
7. Enable SSH for Remote Access

- SSH is usually enabled by default. If not:

```
sudo systemctl enable ssh  
sudo systemctl start ssh
```

- Find your Orange Pi's IP address and connect from your PC:

```
ssh orangepi@<IP_ADDRESS>
```



The screenshot shows a macOS terminal window with three tabs open. The active tab is titled "marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 93x22". The command entered was "ssh orangepi@192.168.5.53". The response indicates that the host's authenticity cannot be established, and it asks if the user wants to continue connecting. The user responds with "yes". It then adds the host to the list of known hosts and prompts for the password. Below this, the terminal shows the Orange Pi's welcome message and system status information.

```
(base) marcelo_rovai@Marcelos-MacBook-Pro ~ %  
(base) marcelo_rovai@Marcelos-MacBook-Pro ~ % ssh orangepi@192.168.5.53  
The authenticity of host '192.168.5.53 (192.168.5.53)' can't be established.  
ED25519 key fingerprint is SHA256:B24e2kgjm8A2eimAytF42vKHf1XeXrN8EyAS7KF41dQ.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.5.53' (ED25519) to the list of known hosts.  
orangepi@192.168.5.53's password:  
  
Welcome to Orange Pi 1.0.0 Noble with Linux 6.6.63-ky  
  
System load: 25% Up time: 10 min Local users: 3  
Memory usage: 7% of 7.65G IP: 172.17.0.1 192.168.5.53  
CPU temp: 49°C Usage of /: 3% of 233G  
  
orangepi@orangepirv2:~$
```

8. (Optional) Install and Use M.2 SSD

- Power off the board, insert the SSD into the M.2 slot, and power back on.
- Check SSD detection:

```
lsblk
```

(The below image is after SSD installation.)

```
marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 73x14
[orangepi@orangepirv2:~]$ lsblk
[orangepi@orangepirv2:~]$ lsblk
  NAME      MAJ:MIN RM    SIZE RO TYPE MOUNTPOINTS
  mtdblock0   31:0    0     64K  0 disk
  mtdblock1   31:1    0     64K  0 disk
  mtdblock2   31:2    0    256K  0 disk
  mtdblock3   31:3    0     64K  0 disk
  mtdblock4   31:4    0    192K  0 disk
  mtdblock5   31:5    0   15.4M  0 disk
  nvme0n1    259:0   0  238.5G  0 disk
└─nvme0n1p1 259:1   0   236G  0 part /var/log.hdd
/
[orangepi@orangepirv2:~]$
```

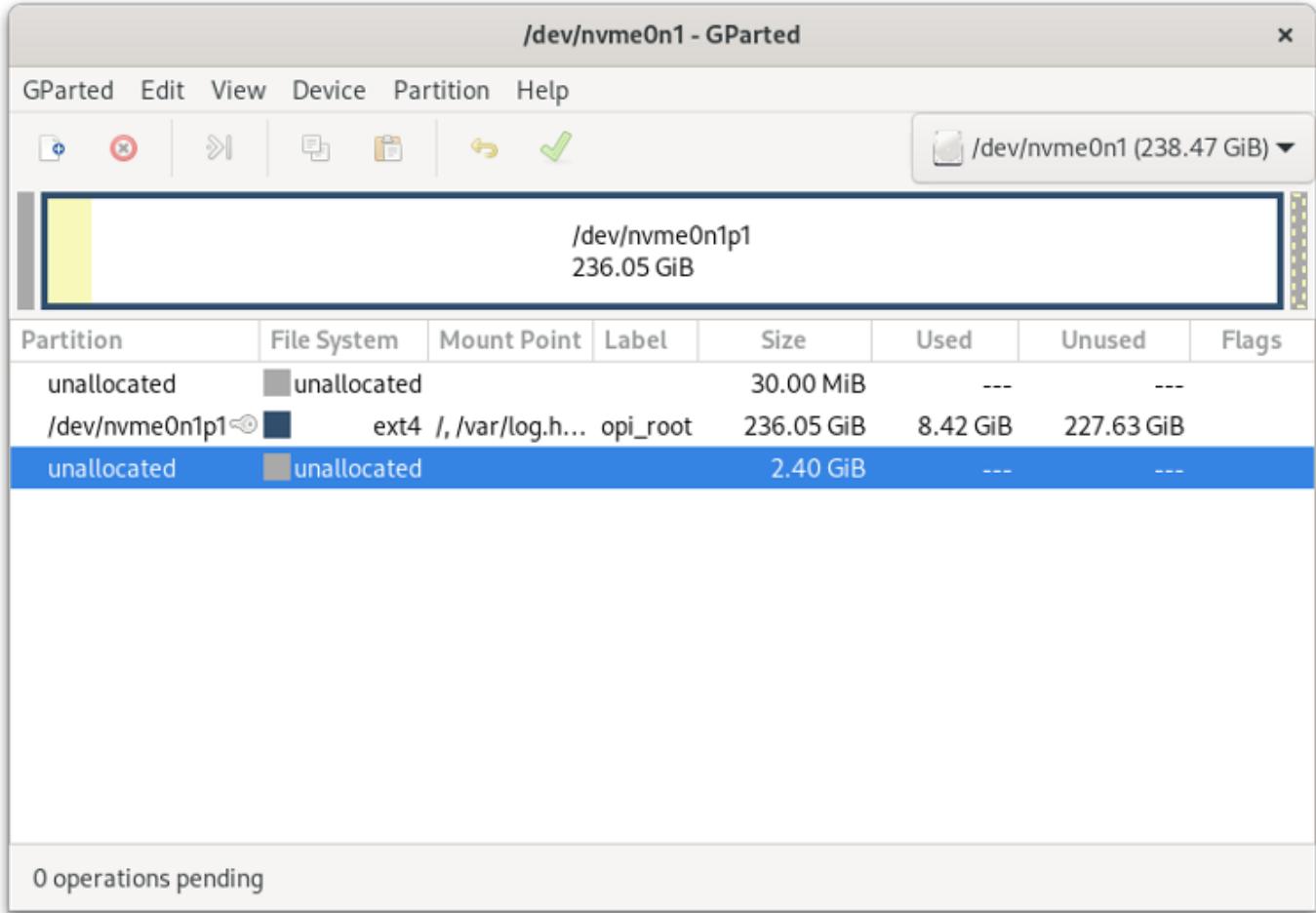
- Partition and format the SSD:

```
sudo fdisk /dev/nvme0n1
sudo mkfs.ext4 /dev/nvme0n1p1
sudo mkdir /mnt/ssd
sudo mount /dev/nvme0n1p1 /mnt/ssd
```

- To boot from SSD, flash the OS image to the SSD and update the SPI flash bootloader using `orangepi-config` (see [reference](#)).

For Storage visualization, install and run GParted

```
sudo apt install gparted
gparted
```



9. Useful First Commands

- To open a new Terminal window:

`Shift + Ctrl + N`

- Check CPU info:

```
lscpu
```

```
[marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 73x17]
[orangepi@orangepirv2:~$ lscpu
Architecture:          riscv64
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:  0-7
Model name:           Ky(R) X1
Thread(s) per core:   1
Core(s) per socket:   8
Socket(s):            1
CPU(s) scaling MHz:  100%
CPU max MHz:          1600.0000
CPU min MHz:          614.4000
Caches (sum of all):
  L1d:                 256 KiB (8 instances)
  L1i:                 256 KiB (8 instances)
  L2:                  1 MiB (2 instances)
orangepi@orangepirv2:~$ ]
```

- Check RAM:

```
free -h
```

```
[marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 94x5]
[orangepi@orangepirv2:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:       7.7Gi       973Mi      5.9Gi      96Mi       1.0Gi      6.7Gi
Swap:          0B          0B          0B
orangepi@orangepirv2:~$ ]
```

- Check storage:

```
df -h
lsblk
```

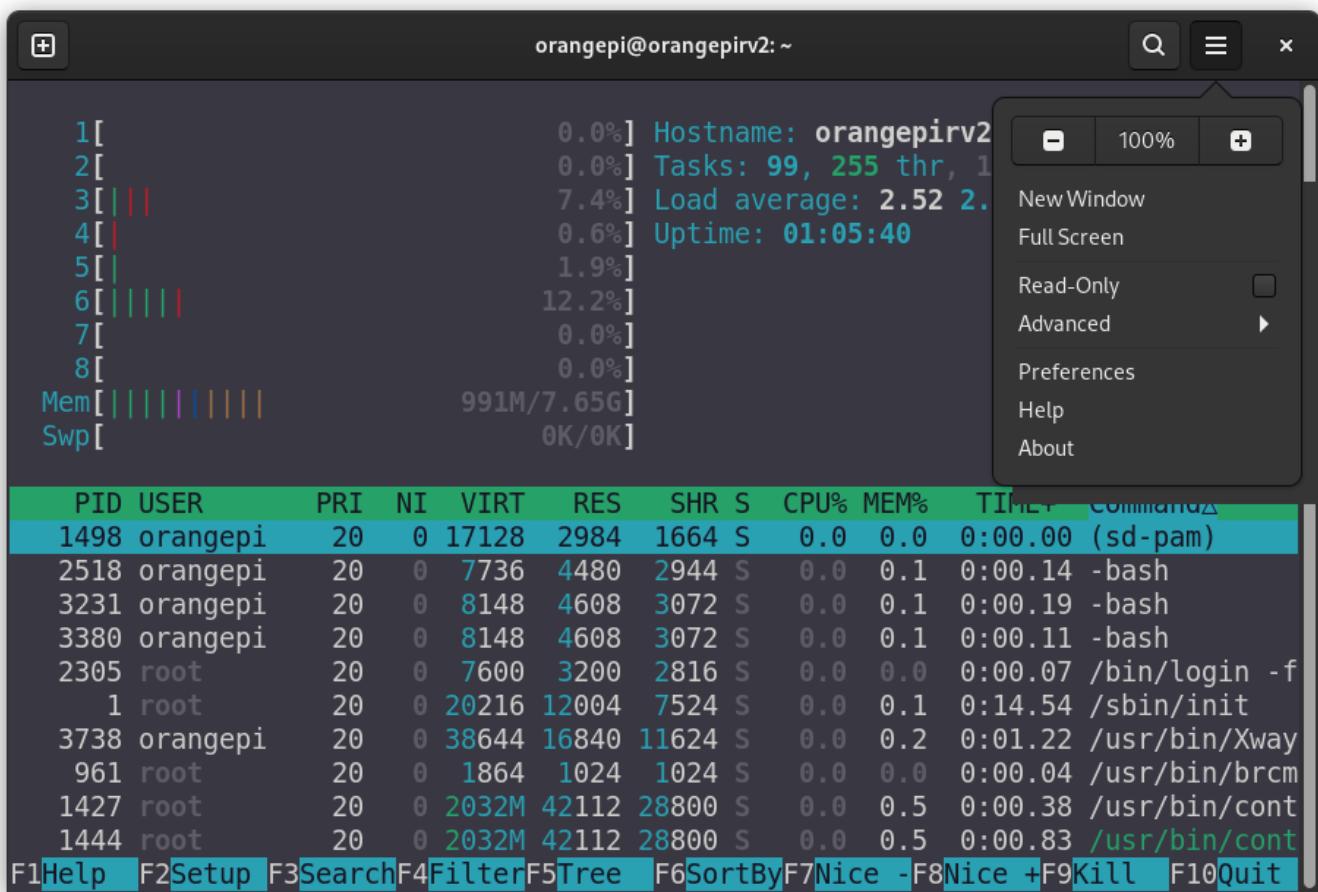
```

marcelo_rovai — orangeipi@orangepirv2: ~ — ssh orangeipi@192.168.5.53 — 63x10
orangeipi@orangepirv2:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           784M   11M  774M   2% /run
/dev/nvme0n1p1  233G  4.7G  226G   3% /
tmpfs            3.9G    0  3.9G   0% /dev/shm
tmpfs            5.0M   12K  5.0M   1% /run/lock
tmpfs            3.9G  8.0K  3.9G   1% /tmp
orangeipi-ramlog 50M  2.2M  48M   5% /var/log
tmpfs           784M   78M  707M  10% /run/user/1001
orangeipi@orangepirv2:~$ 

```

- Use htop for system monitoring

htop



10. Additional Tips

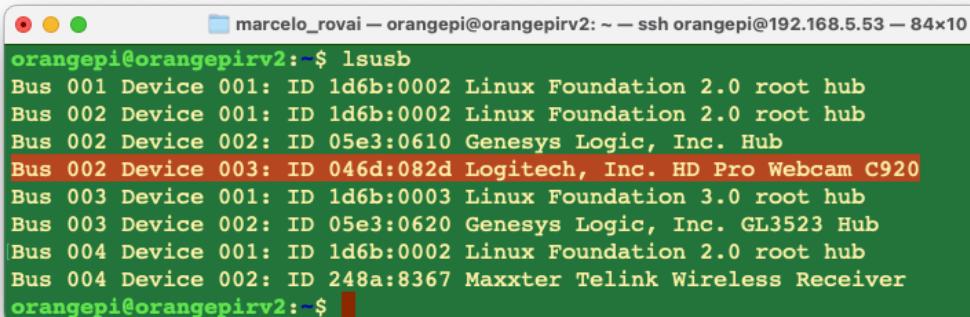
- To install software:

```
sudo apt update  
sudo apt install <package>
```

- For a webcam:

Check the camera:

```
lsusb
```

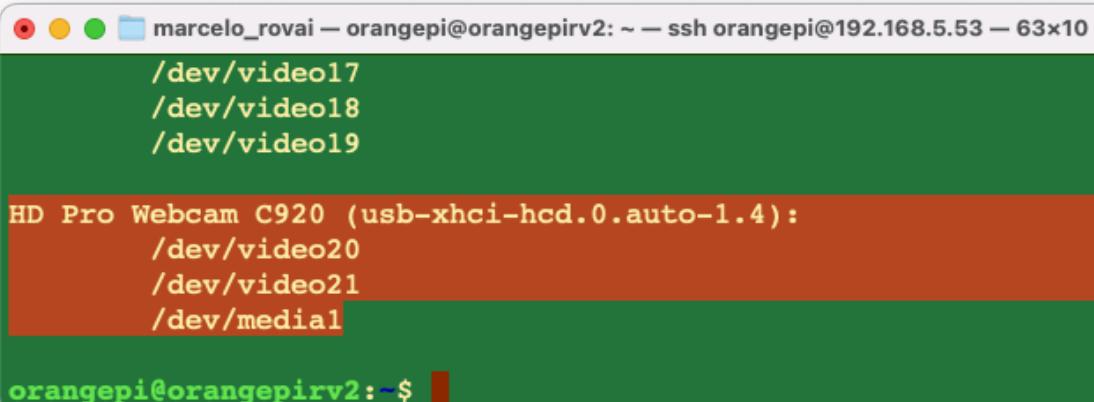


A terminal window titled "marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 84x10". The command "lsusb" is run, and the output shows various USB devices. The Logitech HD Pro Webcam C920 is highlighted in red.

```
marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 84x10  
orangepi@orangepirv2:~$ lsusb  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 002 Device 002: ID 05e3:0610 Genesys Logic, Inc. Hub  
Bus 002 Device 003: ID 046d:082d Logitech, Inc. HD Pro Webcam C920  
Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 003 Device 002: ID 05e3:0620 Genesys Logic, Inc. GL3523 Hub  
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 004 Device 002: ID 248a:8367 Maxxter Telink Wireless Receiver  
orangepi@orangepirv2:~$
```

Check the video device:

```
v4l2-ctl --list-devices
```



A terminal window titled "marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 63x10". The command "v4l2-ctl --list-devices" is run, and the output lists video devices. The HD Pro Webcam C920 is highlighted in red.

```
marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 63x10  
/dev/video17  
/dev/video18  
/dev/video19  
  
HD Pro Webcam C920 (usb-xhci-hcd.0.auto-1.4):  
/dev/video20  
/dev/video21  
/dev/media1  
  
orangepi@orangepirv2:~$
```

```
sudo apt install fswebcam  
fswebcam -d /dev/videoX --no-banner -S 5 test.jpg
```

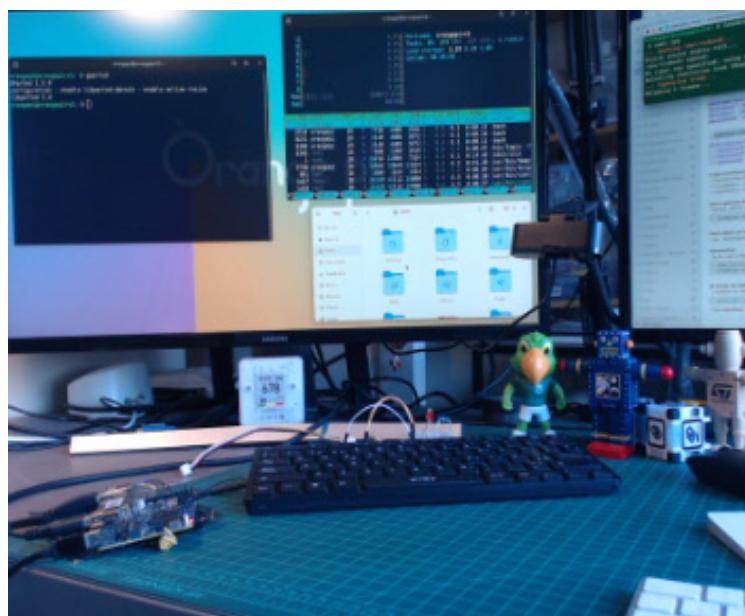
(Replace `/dev/videoX` with your actual video device.)

```

marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 84x13
orangepi@orangepirv2:~$ fswebcam -d /dev/video20 --no-banner -S 5 test.jpg
--- Opening /dev/video20...
Trying source module v4l2...
/dev/video20 opened.
No input was specified, using the first.
Adjusting resolution from 384x288 to 352x288.
[--- Capturing frame...
Skipping 5 frames...
Capturing 1 frames...
Captured 6 frames in 0.17 seconds. (35 fps)
--- Processing captured image...
Disabling banner.
Writing JPEG image to 'test.jpg'.

```

The result:



- For CPU temperature check:

```
sensors
```

```

marcelo_rovai — orangepi@orangepirv2: ~ — ssh orangepi@192.168.5.53 — 84x10
orangepi@orangepirv2:~$ sensors
cluster0_thermal-virtual-0
Adapter: Virtual device
temp1:      +53.0°C

cluster1_thermal-virtual-0
Adapter: Virtual device
temp1:      +53.0°C

orangepi@orangepirv2:~$ 

```

- For automatic monitoring (each 2 seconds:)

```
watch sensors  
wa
```

Ollama Installation

Official Ollama instalation for Linux is based on ARM and will not work for Risc-V.

For installation, follow the instructions on this website:

<https://www.jeffgeerling.com/blog/2025/how-build-ollama-run-langs-on-risc-v-linux>

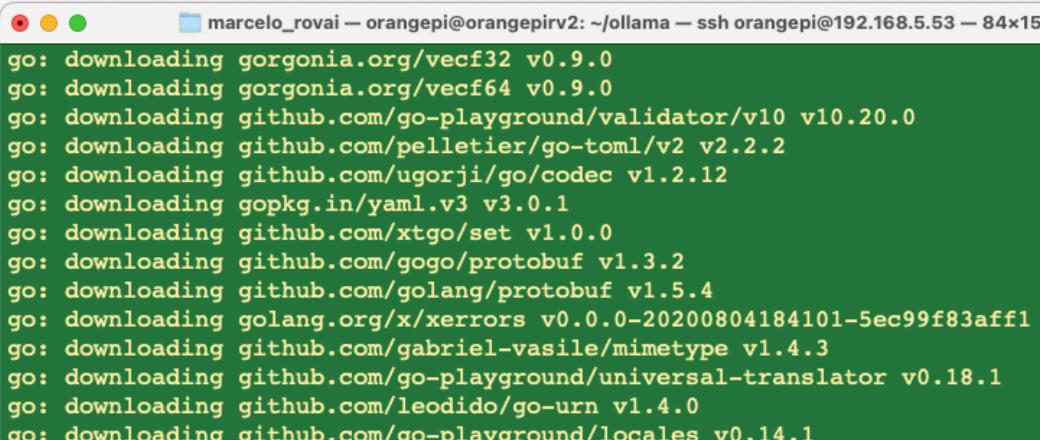
```
# Install Go (download link is under 'Unstable versions')  
wget https://go.dev/dl/gol.24rc3.linux-riscv64.tar.gz  
sudo rm -rf /usr/local/go && sudo tar -C /usr/local -xzf gol.24rc3.linux-riscv64.tar.gz
```

```
# check the version  
go version
```

```
go version gol.24rc3 linux/riscv64
```

```
# Build ollama (This is an old version)  
git clone --recurse-submodules https://github.com/mengzhuo/ollama.git  
cd ollama  
go build .
```

After downloding files, the terminal will stay several minutes (more than 10) building Ollama, but there is no indication. **Wait for its end.**



A screenshot of a terminal window titled "marcelo_rovai — orangepi@orangepirv2: ~/ollama — ssh orangepi@192.168.5.53 — 84x15". The window shows a list of Go modules being downloaded from GitHub, including gorgonia.org/vecf32, gorgonia.org/vecf64, github.com/go-playground/validator, github.com/pelletier/go-toml, github.com/ugorji/go/codec, gopkg.in/yaml.v3, github.com/xtgo/set, github.com/gogo/protobuf, github.com/golang/protobuf, golang.org/x/errors, github.com/gabriel-vasile/mimetype, github.com/go-playground/universal-translator, github.com/leodido/go-urn, and github.com/go-playground/locales. The output is in green text on a dark background.

The temperature went to aprox. 67oC during instalation, with all 8 cores running (part of the time).

```
sudo ln -s `pwd`/ollama /usr/local/bin/ollama
```

Run Ollama

```
# Run ollama and test a tiny model (approx. 400 MB)
ollama serve > /dev/null 2>&1 &
```

Check if it is running, for example listing the models (should be empty):

```
Ollama list
```

```
marcelo_rovai — orangepi@orangepirv2: ~/ollama — ssh orangepi@192.168.5.53 — 84x5
[orangepi@orangepirv2:~/ollama$ ollama serve > /dev/null 2>&1 &
[1] 12630
[orangepi@orangepirv2:~/ollama$ ollama list
NAME      ID      SIZE      MODIFIED
orangepi@orangepirv2:~/ollama$ ]
```

Let's pull a small model for testing:

```
ollama run qwen:0.5b
```

```
marcelo_rovai — orangepi@orangepirv2: ~/ollama — ssh orangepi@192.168.5.53 — 84x24
[orangepi@orangepirv2:~/ollama$ ollama run qwen:0.5b
pulling manifest
pulling fad2a06e4cc7... 100% [██████████] 394 MB
pulling 41c2cf8c272f... 100% [██████████] 7.3 KB
pulling 1da0581fd4ce... 100% [██████████] 130 B
pulling f02dd72bb242... 100% [██████████] 59 B
pulling ea0a531a015b... 100% [██████████] 485 B
verifying sha256 digest
writing manifest
success
|>>> /bye
[orangepi@orangepirv2:~/ollama$ ollama run qwen:0.5b --verbose
|>>> what is your name?
I am Qwen, a large language model developed by Alibaba Cloud.

total duration:      8.391354128s
load duration:      145.093819ms
prompt eval count:    13 token(s)
prompt eval duration: 2.588s
prompt eval rate:      5.02 tokens/s
eval count:          15 token(s)
eval duration:        5.653s
eval rate:            2.65 tokens/s
|>>> Send a message (/? for help)
```

It is important to constantly monitor the temperature. Running a SLM model, the temperature can easily go over 90°C (without Heatsink and fan). So, installing heatsinks and a fan **is mandatory** to run SLMs.

This Ollama version is not up-to-date, so some of recent models could not work.

Troubleshooting

- If you see no display, check the power, the SD card, and the monitor connections.
 - For Wi-Fi issues, check `nmcli` output and ensure your SSID/password are correct.
 - For SSD issues, ensure the SSD is inserted correctly and supported by the kernel.
-

References

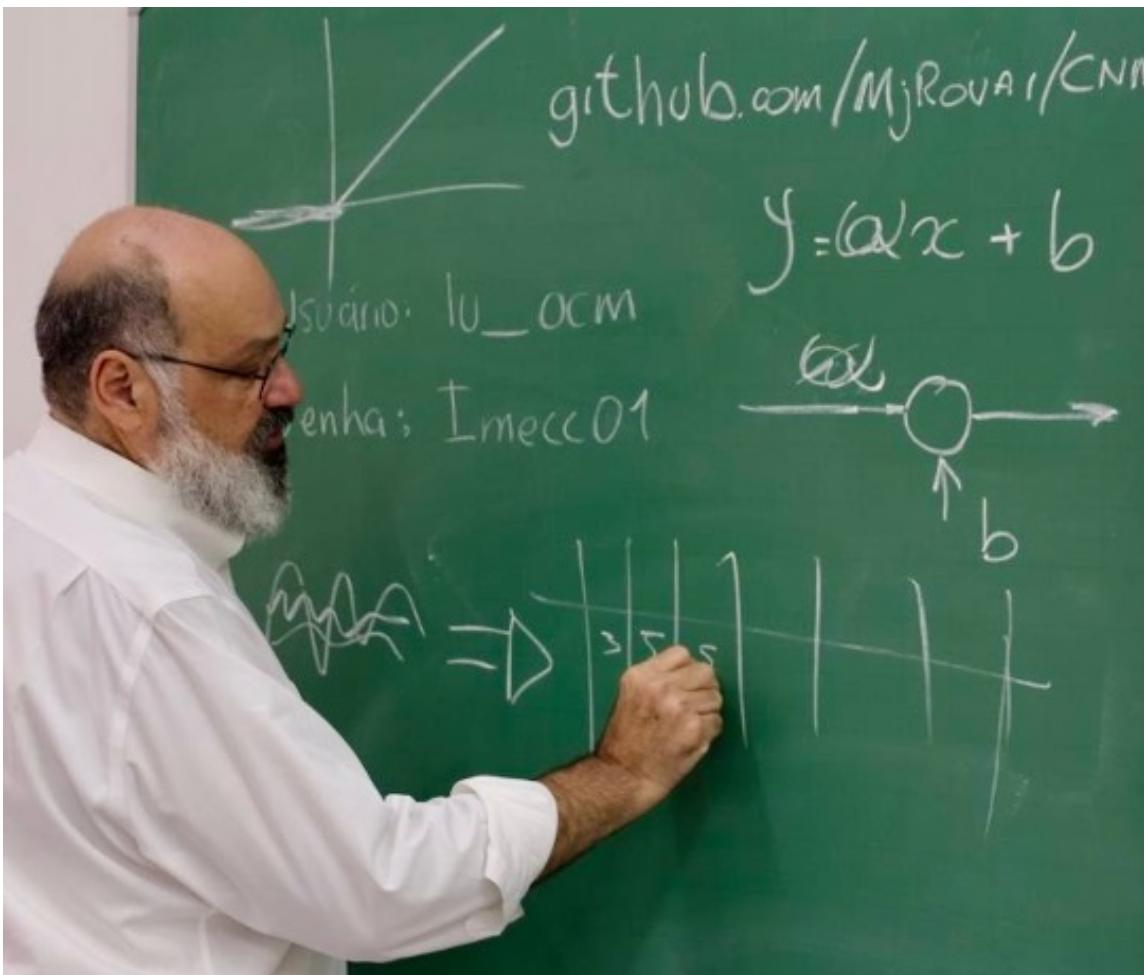
- [Getting Started Video Guide](#)
 - [Orange Pi RV2 Ubuntu Desktop Setup](#)
 - [Orange Pi 5 SSD Boot Reference](#)
 - [Orange Pi RV2 Unboxing](#)
-

This guide covers the whole process from flashing the OS to first boot, networking, remote access, and using additional hardware like SSDs and webcams. For any advanced configurations or troubleshooting, refer to the official Orange Pi forums and documentation.

Sources

- [1] Getting Started with Orange Pi RV2 | Flashing Ubuntu Noble Server <https://www.youtube.com/watch?v=gAnhCtJ5Hpl>
- [2] Getting Started with Orange Pi RV2 | Flash Ubuntu Noble Desktop ... <https://www.youtube.com/watch?v=25alsHlkOig>
- [3] Orange Pi 5 - Easy Setup and Overview! - YouTube <https://www.youtube.com/watch?v=cBqV4QWj0IE>
- [4] OrangePi RV2 unboxing - getting started with RISC/V Linux - YouTube https://www.youtube.com/watch?v=ICgpn_SE22o
- [5] Orange Pi RV2 - 8 Núcleos RISC-V // Todo lo que necesitas saber https://www.youtube.com/watch?v=UoW8b_yrFdY
- [6] OrangePi RV2 <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-RV2.html>
- [7] Getting Started with Orange Pi RV2 | Flash Ubuntu Noble Desktop ... https://www.reddit.com/r/OrangePi/comments/1jrs7sp/getting_started_with_orange_pi_rv2_flash_ubuntu/
- [8] My first impression of the Orange Pi RV2 with Ubuntu - Reddit https://www.reddit.com/r/RISCV/comment/s1k267ts/my_first_impression_of_the_orange_pi_rv2_with/
- [9] Orange Pi RV2 Review: A Complete Detailed Look - AndroidPIMP <https://www.androidpimp.com/embedded/orange-pi-rv2/>
- [10] Xunlong Orange Pi RV2 (xunlong-orangepi-rv2) - postmarketOS Wiki [https://wiki.postmarketos.org/wiki/Xunlong_Orange_Pi_RV2_\(xunlong-orangepi-rv2\)](https://wiki.postmarketos.org/wiki/Xunlong_Orange_Pi_RV2_(xunlong-orangepi-rv2))

About the author



Marcelo Rovai, a Brazilian living in Chile, is an educator and professional in the field of engineering and technology, holding the title of Professor Honoris Causa from the Federal University of Itajubá (UNIFEI), Brazil. His educational background includes an Engineering degree from UNIFEI and a specialization from the Polytechnic School of São Paulo University (POLI/USP). Further enhancing his expertise, he earned an MBA from IBMEC (INSPER) and a Master's in Data Science from the Universidad del Desarrollo (UDD) in Chile.

With a career spanning several high-profile technology companies such as AVIBRAS Airspace, AT&T, NCR, and IGT, where he served as Vice President for Latin America, he brings industry experience to his academic endeavors. He is a prolific writer on electronics-related topics and shares his knowledge through open platforms like Hackster.io.

In addition to his professional pursuits, he is dedicated to educational outreach, serving as a volunteer professor at UNIFEI and engaging with the [TinyML4D group](#) and the [EDGE AIP](#) – the Academia-Industry Partnership of EDGEAI Foundation as a Co-Chair, promoting TinyML education in developing countries. His work underscores a commitment to leveraging technology for societal advancement.

LinkedIn profile: <https://www.linkedin.com/in/marcelo-jose-rovai-brazil-chile/>

Lectures, books, papers, and tutorials: <https://github.com/Mjrovai/TinyML4D>