IESTI05 – Edge AI

Machine Learning System Engineering

2. Introduction to Embedded Linux and Raspberry Pi Setup

UNIFEI

Marcelo Rovai

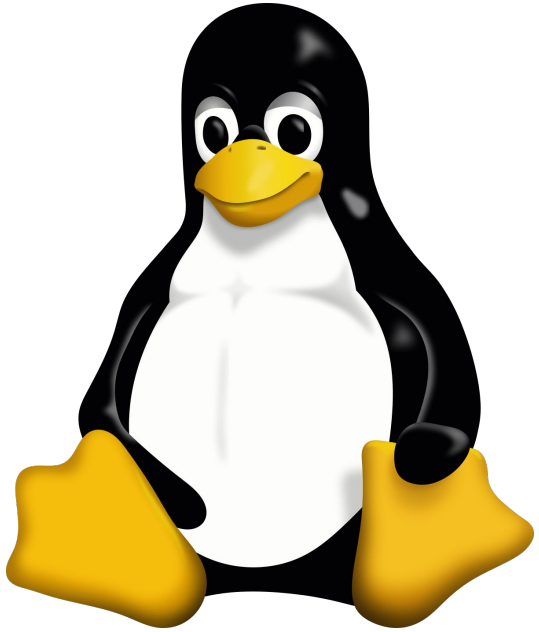
Professor Honoris Causa

Embedded Linux

The Operating System (OS)

An operating system (OS) is essential software that manages computer hardware and software resources, providing standard services for computer programs. It is the core software that runs on a computer, serving as an intermediary between hardware and application software. The OS oversees the computer's memory, processes, device drivers, files, and security protocols.

1. Key functions:
   - Process management: Allocating CPU time to different programs
   - Memory management: Allocating and freeing up memory as needed
   - File system management: Organizing and keeping track of files and directories
   - Device management: Communicating with connected hardware devices
   - User interface: Providing a way for users to interact with the computer



2. Components:
   - Kernel: The core of the OS that manages hardware resources (i.e., Linux)
   - Shell: The user interface for interacting with the OS
   - File system: Organizes and manages data storage
   - Device drivers: Software that allows the OS to communicate with hardware

What is Embedded Linux?

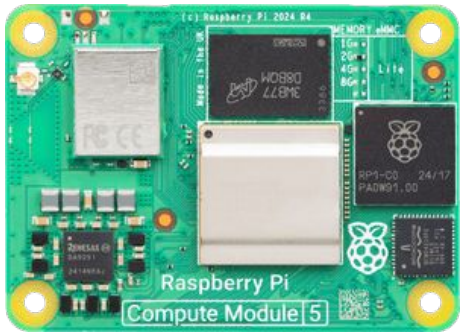
- A lightweight Linux-based OS for embedded systems
- Customizable, open-source, and widely adopted
- Used in routers, drones, SBCs, industrial systems

Raspberry Pi OS Lite (64-bit) is a minimal, headless Linux distribution for the Raspberry Pi. It is based on Debian (like the desktop version), but it excludes a graphical desktop environment and most bundled applications, providing just the core operating system and essential command-line tools

Why Use Linux in Embedded Systems?

- Open-source and free
- Highly flexible and customizable
- Large developer ecosystem
- Reliable and proven in production

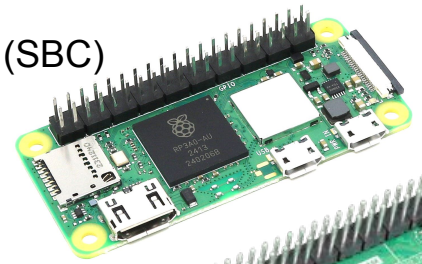
(Compute Module / SoM)



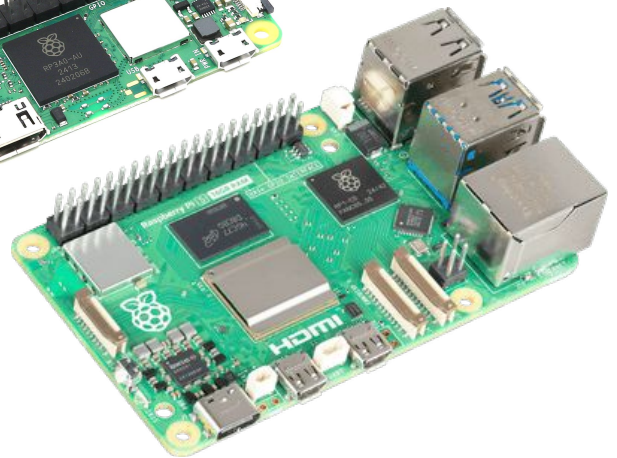
Characteristics

- UI: Usually headless
- Size: Minimal
- Hardware: Targeted SBC

(SBC)



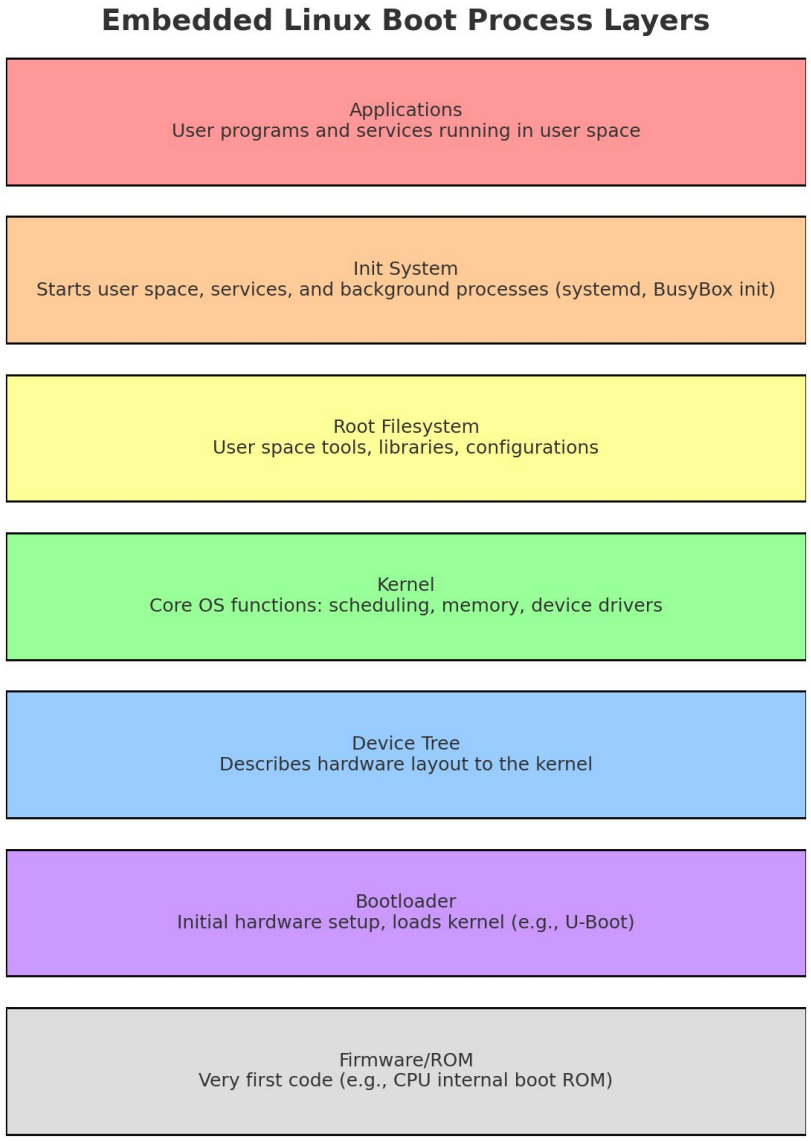
(SBC)



(Compute Module / SoM)

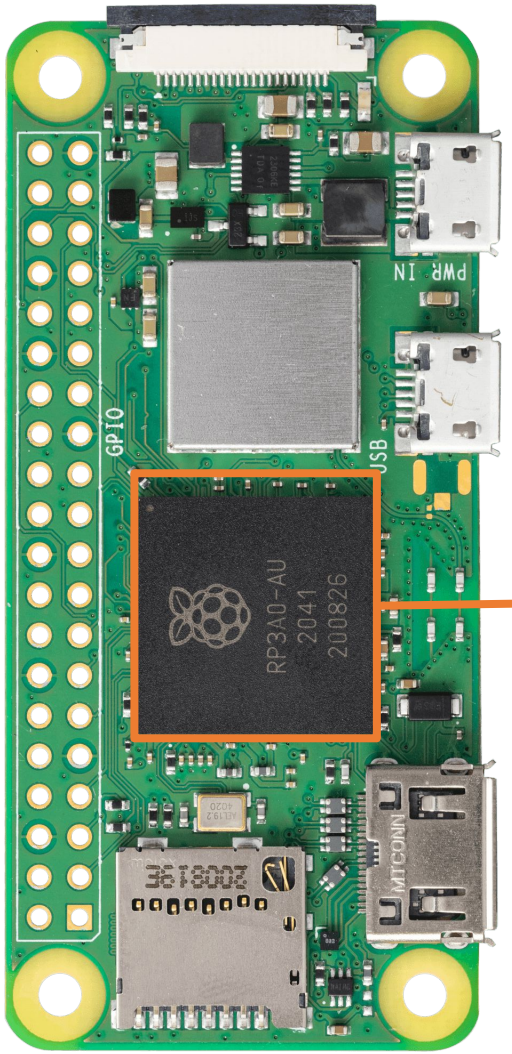
Linux System Components

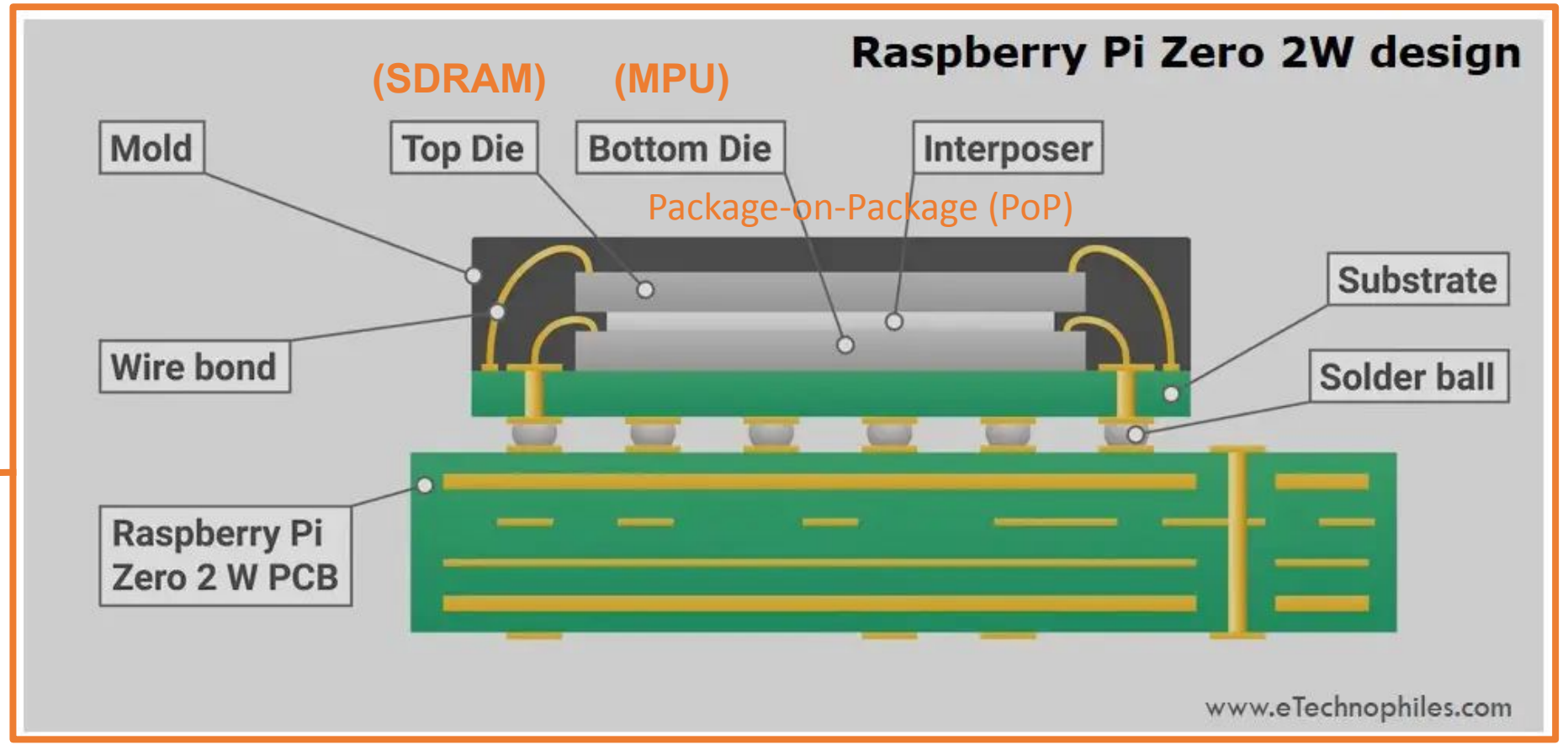
- Firmware/ROM
- Bootloader – e.g., U-Boot
- Device tree – hardware description
- Kernel – core OS functions
- Root filesystem – user space tools
- Init system – systemd or BusyBox
- Applications



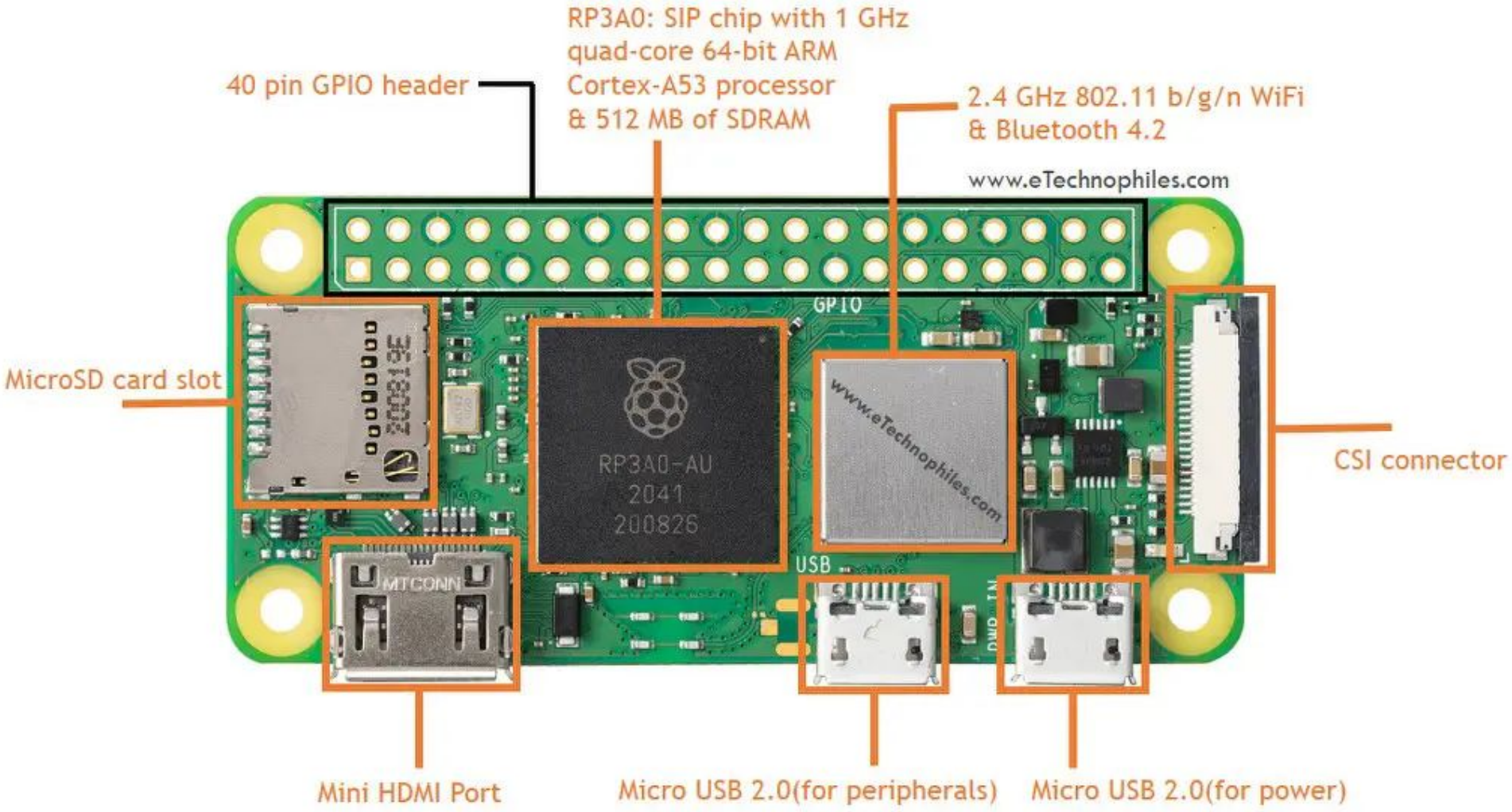
Raspberry Pi 2W

Broadcom BCM2710A1 SoC (System-on-a-chip)

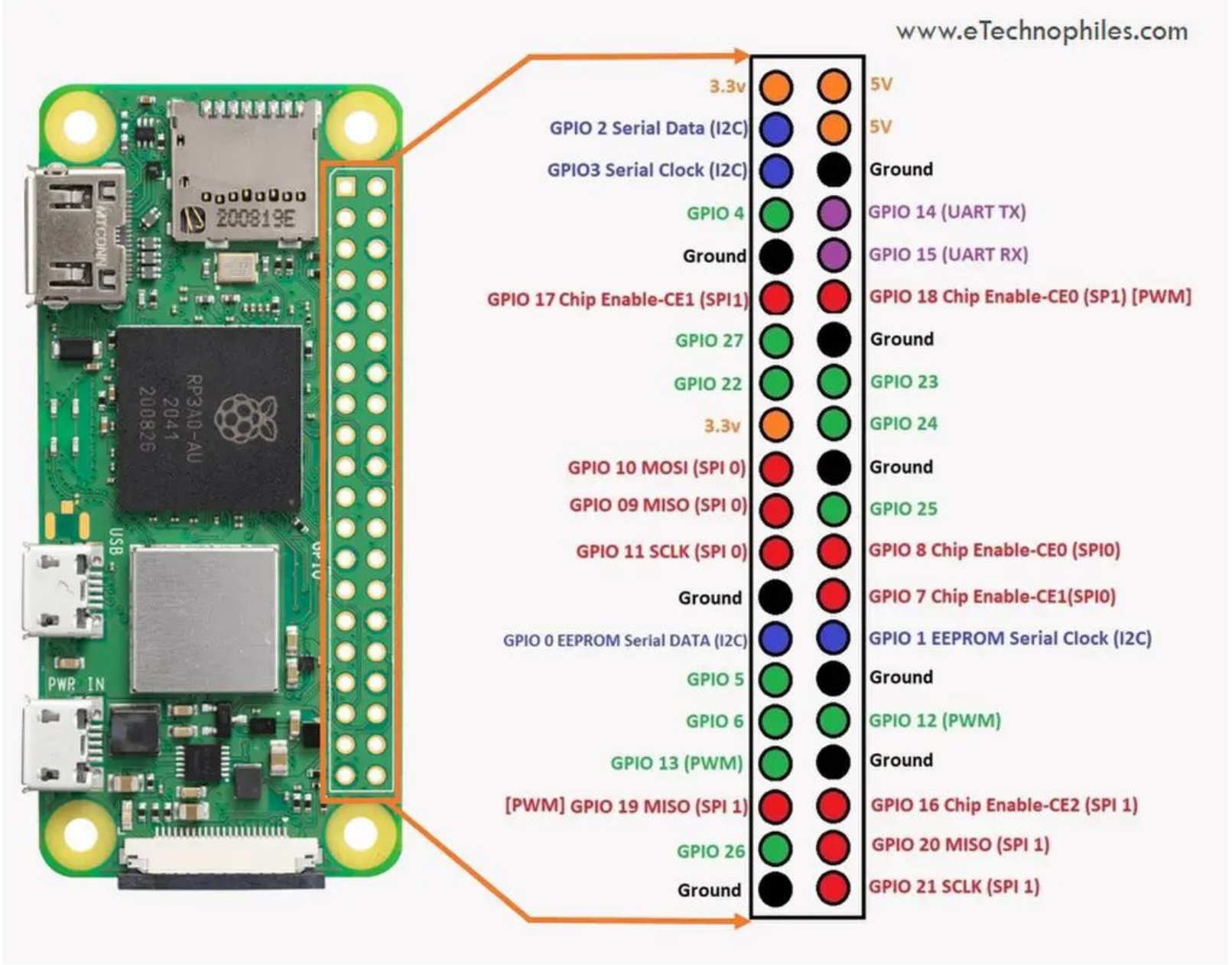




Board Layout



GPIO pinout

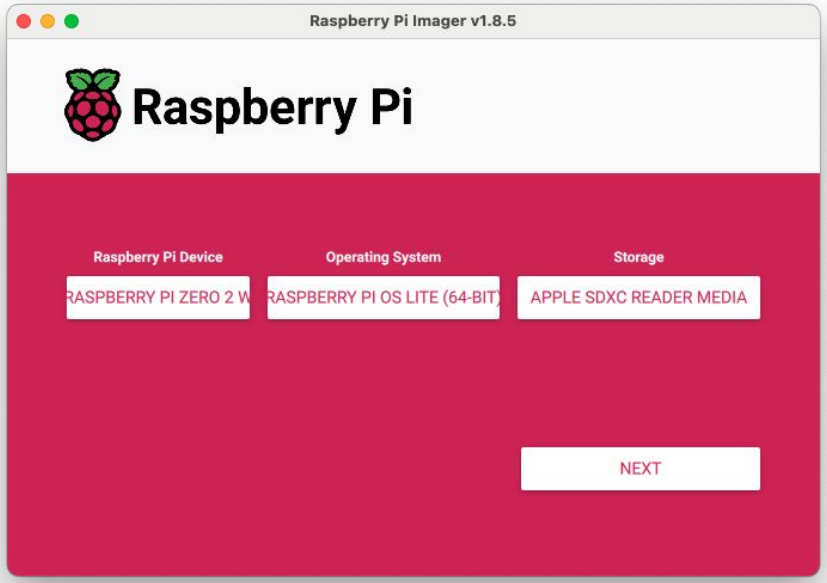

www.eTechnophiles.com
3.3v
5V
GPIO 2 Serial Data (I2C)
5V
GPIO3 Serial Clock (I2C)
Ground
GPIO 4
GPIO 14 (UART TX)
Ground
GPIO 15 (UART RX)
GPIO 17 Chip Enable-CE1 (SPI1)
GPIO 18 Chip Enable-CE0 (SP1) [PWM]
GPIO 27
Ground
GPIO 22
GPIO 23
3.3v
GPIO 24
GPIO 10 MOSI (SPI 0)
Ground
GPIO 09 MISO (SPI 0)
GPIO 25
GPIO 11 SCLK (SPI 0)
GPIO 8 Chip Enable-CE0 (SPI0)
Ground
GPIO 7 Chip Enable-CE1(SPI0)
GPIO 0 EEPROM Serial DATA (I2C)
GPIO 1 EEPROM Serial Clock (I2C)
GPIO 5
Ground
GPIO 6
GPIO 12 (PWM)
GPIO 13 (PWM)
Ground
[PWM] GPIO 19 MISO (SPI 1)
GPIO 16 Chip Enable-CE2 (SPI 1)
GPIO 26
GPIO 20 MISO (SPI 1)
Ground
GPIO 21 SCLK (SPI 1)
PWR IN
USB

Installing the OS

Use Raspberry Pi Imager
https://www.raspberrypi.com/software/

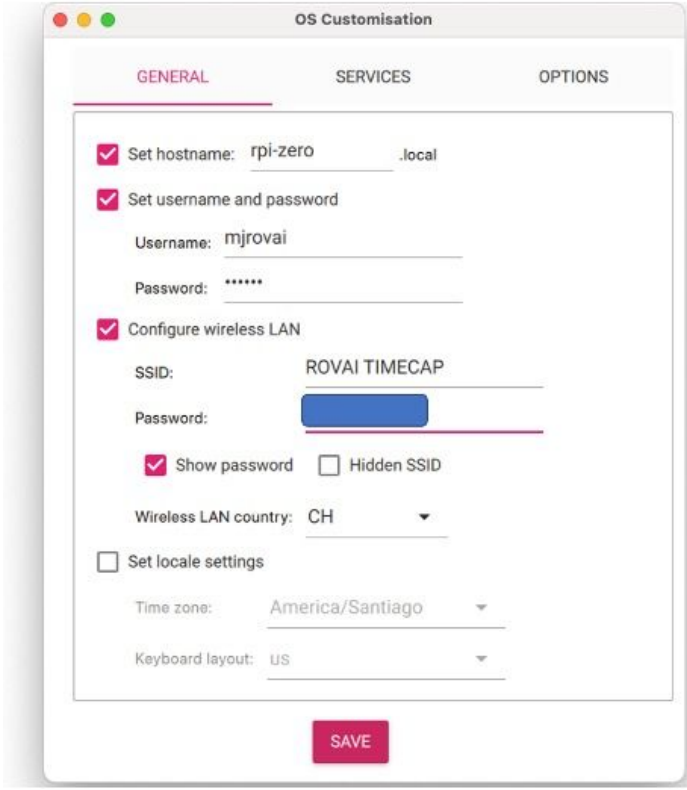
and select:

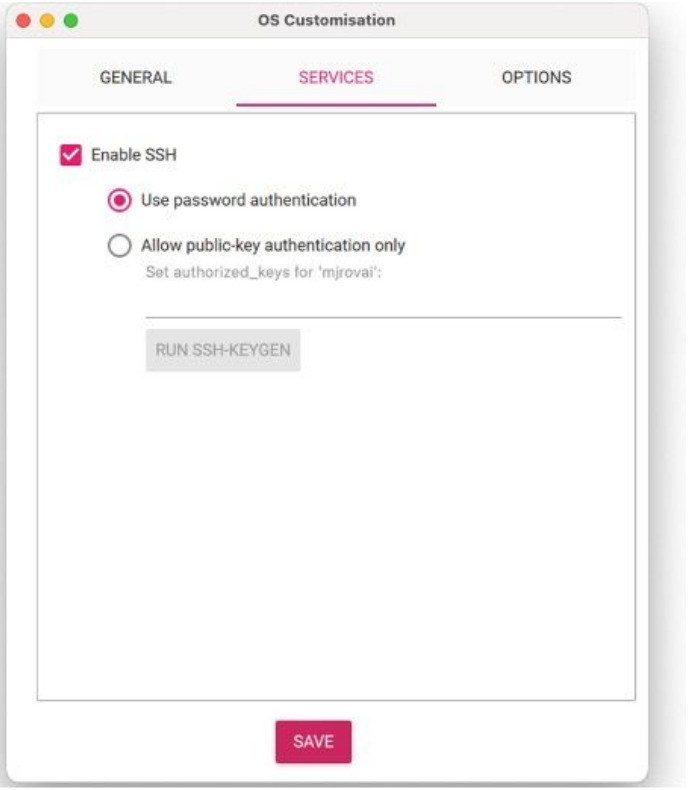
- RASPBERRY PI ZERO 2W
- RASPBERRY PI OS LITE (64-BIT)



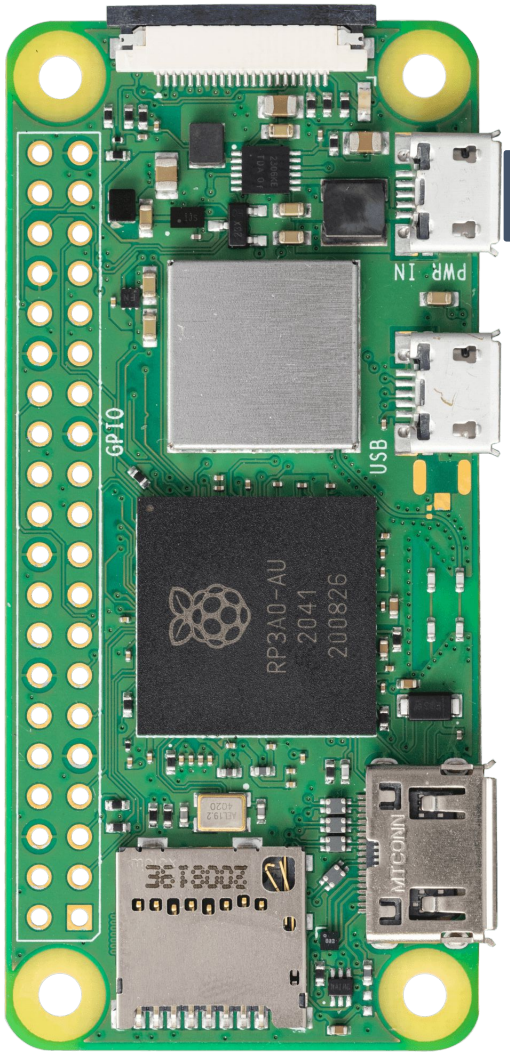
Headless setup: enable SSH, Wi-Fi config

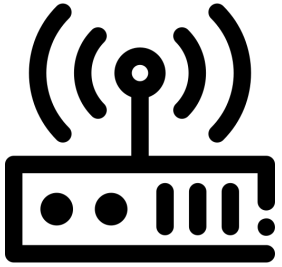
- Define hostname, username, and password





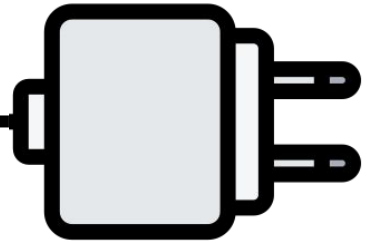
Power Supply and WiFi Network





(WiFi X)
2.4GHz

USB 2.0 – 5V/2.5A



(Option)

(WiFi X)

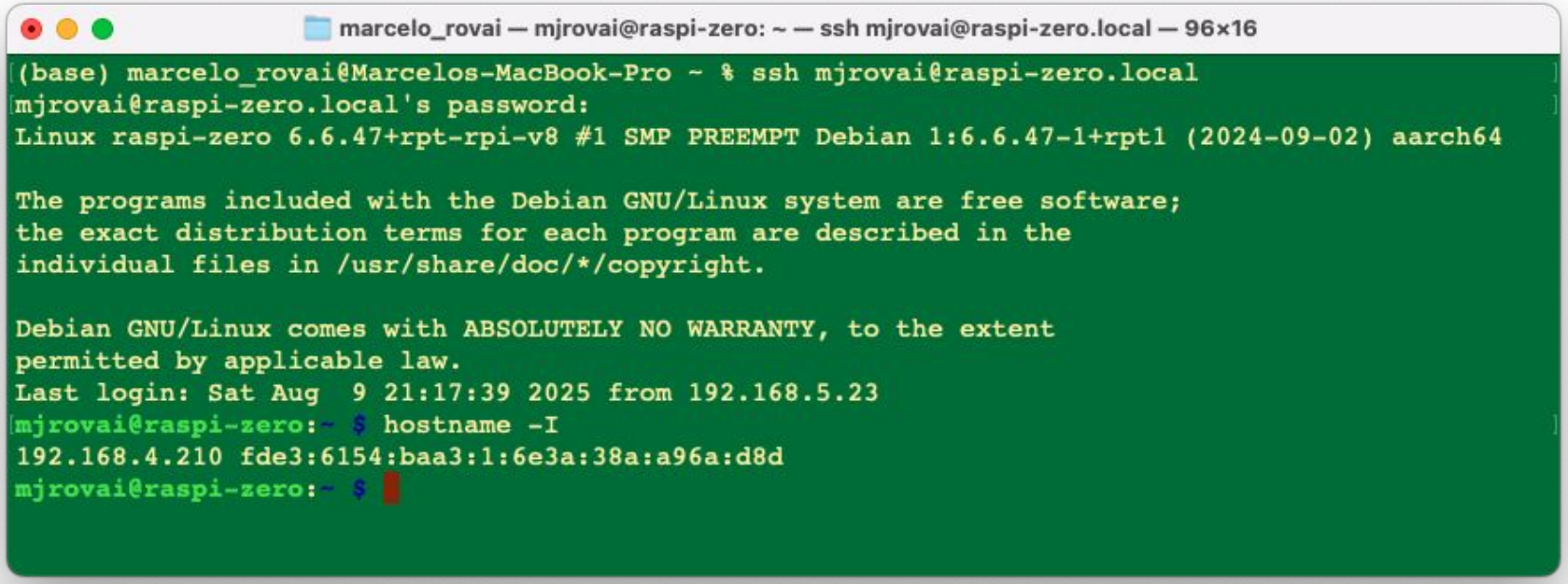
(WiFi X)

USB 2.0 – 5V/2.5A

Rasp Pi Zero and Notebook should be on the same WiFi network

Connecting to the Pi – SSH Via Terminal

- Enter with: ssh username@hostname.local (i.e., mjrovai@raspi-zero.local)
  - Once in the Raspi-Zero, use hostname –I to get the IP Address



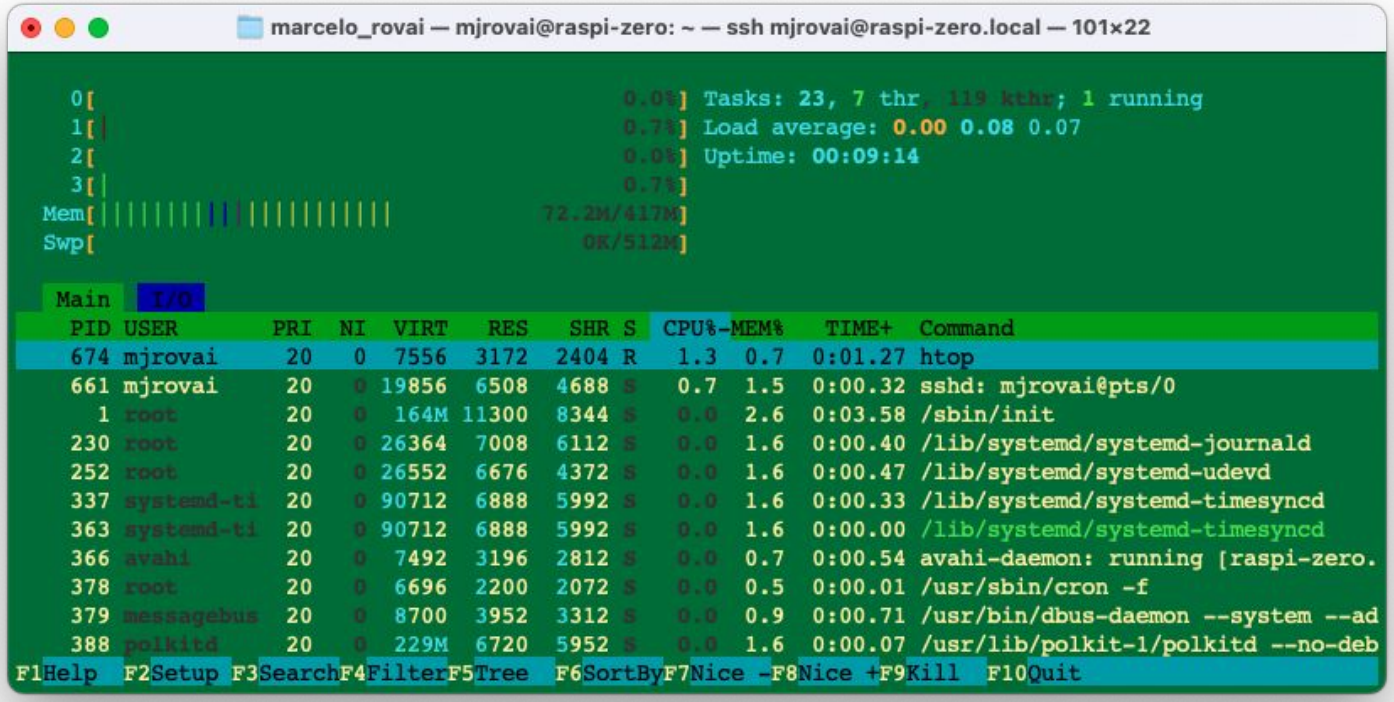
- Knowing the IP address: Enter with:

  ssh username@ip_address (i.e., mjrovai@ 192.168.4.210)

Note: On Windows, use Command Prompt (cmd) or PowerShell.

Initial Linux Commands

1. Package mgmt:
   - sudo apt update && upgrade
   - sudo reboot
2. System: (Open in a new terminal window)
   - htop



Increasing SWAP Memory

1. First, turn off the swap-file:
   sudo dphys-swapfile swapoff

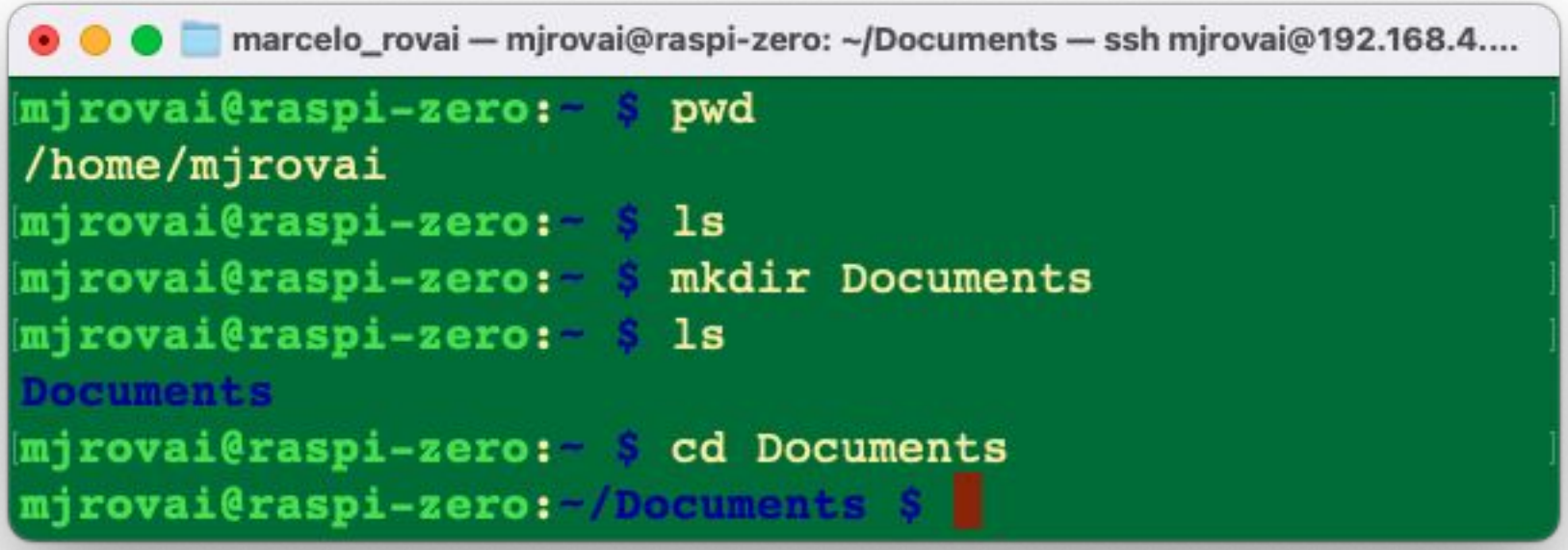
2. Next, open and modify the file /etc/dphys-swapfile. For that, we will use the **nano** text editor:
   sudo nano /etc/dphys-swapfile

   a. Search for the **CONF_SWAPSIZE** variable (default is 200) and
   b. update it to **2000**: CONF_SWAPSIZE=2000, and
      c. save the file: CTRL+X -> Y => Enter.

3. Next, turn on the swapfile again and reboot the Raspberry Pi:
   sudo dphys-swapfile setup
   sudo dphys-swapfile swapon
   sudo reboot

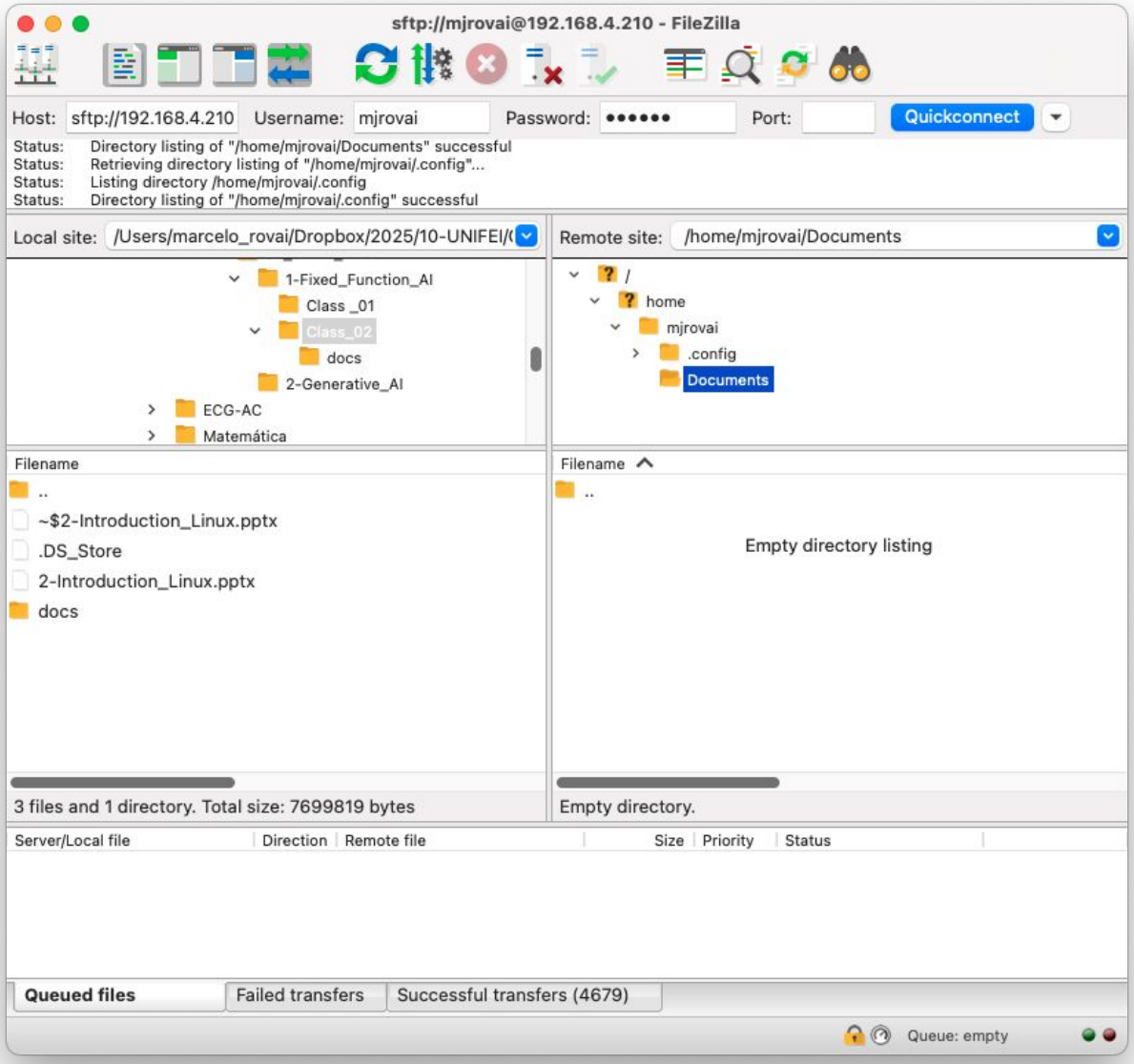
Linux: Basic Commands

- clear -> Clear the terminal
- pwd -> Show the current directory: /home/mjrovai
- ls -> Lists the current directory content: (empty)
- Mkdir <name> -> Creates a directory: mkdir Documents
- cd <dir> -> Change to a directory cd Documents



Transferring files using FTP

1. Install FileZila Client in the Desktop
   https://filezilla-project.org/download.php?type=client
2. Enter with Host Credentials
   (i.e., sftp://192.168.4.210)



Using the Camara Module

1. Install camera software (if not pre-installed):
   sudo apt-get install libcamera-apps

2. List the installed cameras:
   rpicam-hello --list-cameras

3. Capture a 640x480 JPEG image:
   rpicam-jpeg --output test_cli_camera.jpg --width 640 --height 480

4. Use the command ls to check if the image was saved in the current directory and transfer it to your desktop with FileZilla.

Tips

- Connecting the Raspberry Pi to the Computer via USB can be unstable for heavy use. Prefer a 5V/2.5 Power Supply (same as used for mobile phones)
- The WiFi Network should be 2.4GHz
- Always turn off the Raspberry Pi, using the command:
  sudo shutdown -h now
- Install packages using
  sudo apt install <package name>

Questions?

Prof. Marcelo J. Rovai

rovai@unifei.edu.br

UNIFEI