Complete Step-by-Step Setup for Raspberry Pi 4B (8GB)

This guide covers the complete setup process for your Raspberry Pi 4B (8GB), including Ubuntu installation, GUI and VNC configuration, ROS 2 Humble setup, Slamtech C1 LiDAR integration, USB camera configuration, and PS4 controller-based mecanum drive control.

# 1. Ubuntu 22.04.05 LTS Installation

Works well with ROS2

* Download the image from https://ubuntu.com/download/raspberry-pi.
* Use Raspberry Pi Imager or Balena Etcher to flash the image to a 32GB+ microSD card.
* Enable SSH by creating a blank file named `ssh` in the boot partition.
* Optionally configure Wi-Fi by editing `network-config`.
* Insert the microSD into the Pi, power on, and SSH in (user: ubuntu, password: ubuntu).

# 2. System Update and GUI Installation

GUI graphical user interface

* Run system update:  
   sudo apt update && sudo apt upgrade -y
* If prompted to update the kernel, choose 'yes' and keep default selections.
* Install XFCE and LightDM:  
   sudo apt install xfce4 lightdm -y
* Reboot:  
   sudo reboot

# 3. Set Up Remote Access with Tailscale

Tailscale is a software-defined network tool that simplifies creating secure and private networks between your devices. (google)

* Install Tailscale:  
   curl -fsSL https://tailscale.com/install.sh | sh
* Start the Tailscale daemon:  
   sudo tailscale up
* Complete authentication at the URL provided in the terminal.
* Get your Pi’s IP address:  
   tailscale ip -4

# 4. Set Up VNC

VNC, or Virtual Network Computing, is a remote desktop-sharing system that allows users to control and view the graphical interface of a computer from another device over a network (google)

* Install VNC:  
   sudo apt install x11vnc -y
* Optional: Auto-login setup:  
   sudo nano /etc/lightdm/lightdm.conf
* Paste (replace 'ubuntu' with your username):  
   [Seat:\*]  
   autologin-user=ubuntu  
   autologin-user-timeout=0  
   user-session=xfce
* Save with Ctrl+O, Enter, then Ctrl+X.
* Start VNC server:  
   x11vnc -usepw -forever -display :0
* Note: Connect from another device on the same Wi-Fi network using your Pi's IP address.

# 5. Install ROS 2 Humble

* Update system and configure locale:  
   sudo apt update && sudo apt upgrade -y  
   sudo apt install locales -y  
   sudo locale-gen en\_US en\_US.UTF-8  
   sudo update-locale LC\_ALL=en\_US.UTF-8 LANG=en\_US.UTF-8

ROS2 was used because it is a dedicated operated system that is for the purpose of Robot Operation

* Add ROS 2 sources and install:  
   sudo apt install curl gnupg lsb-release -y  
   sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
* Add source list:  
   echo "deb [signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU\_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
* Update and install:  
   sudo apt update  
   sudo apt install ros-humble-desktop -y
* Source ROS in your shell:  
   echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc  
   source ~/.bashrc
* Test installation:  
   ros2 run demo\_nodes\_cpp talker

# 6. Create ROS 2 Workspace

ROS2 workspace serves as a dedicated directory for organizing and developing your robot’s software.

* Install colcon:  
   sudo apt install python3-colcon-common-extensions -y
* Create and build workspace:  
   mkdir -p ~/ros2\_ws/src  
   cd ~/ros2\_ws  
   colcon build  
   source install/setup.bash

# 7. Configure Mecanum Drive Node with PS4 Controller

* Create the package:  
   ros2 pkg create --build-type ament\_python ps4\_mecanum --dependencies rclpy sensor\_msgs

This is where you create a NODE that is dedicated to the software that will be controlling the movement of your wheels

* Add `mecanum\_drive\_node.py` with your motor control logic.
* Edit setup.py to include correct entry points.
* Create `\_\_init\_\_.py`:  
   touch ~/ros2\_ws/src/ps4\_mecanum/ps4\_mecanum/\_\_init\_\_.py
* Build the package:  
   cd ~/ros2\_ws  
   colcon build  
   source install/setup.bash
* Run joy and mecanum nodes:  
   ros2 run joy joy\_node  
   ros2 run ps4\_mecanum mecanum\_drive\_node
* If you see 'RPi' import errors:  
   sudo apt install python3-rpi.gpio -y

# 8. Connect and Pair PS4 Controller

* Reset Bluetooth:  
   sudo rfkill unblock Bluetooth  
   sudo systemctl restart Bluetooth  
   bluetoothctl

ROS2 was used because it is a dedicated operated system that is for the purpose of Robot Operation

* Inside bluetoothctl:  
   power on  
   agent on  
   default-agent  
   scan on
* Hold PS+Share until controller blinks rapidly. Then pair:  
   pair <MAC>  
   trust <MAC>  
   connect <MAC>
* Confirm controller connected:  
   ls /dev/input/ → should show `js0`  
   jstest /dev/input/js0

# 9. Enable Headless Boot & Dummy HDMI

* Enable SSH:  
   sudo touch /boot/firmware/ssh

This should be done so that you can boot the Raspberry Pi and it load completely without the need of having it connected to a monitor

* Edit display config:  
   sudo nano /boot/firmware/config.txt
* Add:  
   hdmi\_force\_hotplug=1  
   hdmi\_group=2  
   hdmi\_mode=82  
   hdmi\_drive=2  
   dtoverlay=imx708
* Disable network wait:  
   sudo systemctl disable systemd-networkd-wait-online.service

# 10. Camera Setup (Arducam or USB Camera)

* Install dependencies for libcamera:  
   sudo apt install -y cmake ... ninja-build  
   pip3 install --user meson

The setup of your camera

* For USB cameras:  
   sudo apt install cheese ffmpeg  
   ffplay /dev/video0
* List available resolutions:  
   v4l2-ctl --list-formats-ext -d /dev/video0
* Install ROS 2 USB cam node:  
   sudo apt install ros-humble-usb-cam
* Create config files: `usb\_cam.yaml`, `camera\_launch.py`, `package.xml`, and `setup.py`.
* Build the workspace:  
   cd ~/ros2\_ws  
   colcon build  
   source install/setup.bash
* Run and visualize:  
   ros2 launch my\_camera\_pkg camera\_launch.py  
   rqt\_image\_view

# 11. Final Notes

* Typical launch sequence:
* Terminal 1:  
   ros2 launch my\_camera\_pkg robot\_system\_launch.py
* Terminal 2:  
   ros2 run joy joy\_node && ros2 run ps4\_mecanum mecanum\_drive\_node
* Make setup easier by editing your ~/.bashrc:
* source /opt/ros/humble/setup.bash  
   source ~/ros2\_ws/install/setup.bash

# 12. Motor HAT Power and Control Setup

* The Cokoino 4WD Motor HAT uses two DRV8833 motor drivers to control four DC motors.
* It accepts a 7V–12.6V input on VIN (we used 18650 9900mAh batteries 3.7V actual ~4V each).
* It steps down voltage to 5V to safely power the Raspberry Pi via GPIO header.
* ⚠️ Ensure the battery supplies enough current for the Pi, camera, LiDAR, and all motors.
* Power and GPIO Overview:  
   - VIN: 7V–12.6V input  
   - 5V: Output to Pi  
   - IN1–IN4 and ENA/ENB: Control GPIO  
   - PWM: Controlled via RPi.GPIO

Information on the Motor Hat and information on a Power Source

* Motor wiring:  
   - Motor 1 = Front Left  
   - Motor 2 = Front Right  
   - Motor 3 = Rear Left  
   - Motor 4 = Rear Right
* Each driver uses two GPIOs per motor for direction, and PWM for speed control.