Artifact ID:	Artifact Title:
GD-001	Instructions for Rpi5 Integration with PixRacer Pro and Ros 2 Jazzy
Revision: 13	Revision Date: 2025-03-04
Prepared by: Israel Zenteno	Checked by: Janie Linford

Purpose:

Instructions for initially integrating the Raspberry pi 5 with PixRacer Pro and Ros 2 Jazzy, establishing communication between the PixRacer Pro and its companion computer.

Revision History

Revision	Revised by	Checked by	Date
01	Israel Zenteno	Janie Linford	2024-10-16
02	Joshua Crookston	Isreal Zenteno	2024-10-21
03	Janie Linford	Joshua Crookston	2024-11-22
04	Israel Zenteno	Joshua Crookston	2024-11-25
05	Janie Linford	Israel Zenteno	2025-01-13
06	Janie Linford	Joshua Crookston	2025-01-14
07	Janie Linford	Israel Zenteno	2025-01-15
08	Janie Linford	Israel Zenteno	2025-01-17
09	Joshua Crookston	Janie Linford	2025-01-22
10	Joshua Crookston	Janie Linford	2025-01-23
11	Joshua Crookston	Israel Zenteno	2025-01-27
12	Janie Linford	Joshua Crookston	2025-02-04
13	Joshua Crookston	Israel Zenteno	2025-03-04
14	Joshua Crookston	Anthony Cardenas	2025-04-07

Initial Setup Guide

for Raspberry Pi 5 with PixRacer Pro and ROS 2 Jazzy

Author: Team 56 “Snowflake”

Date: 2025-03-04

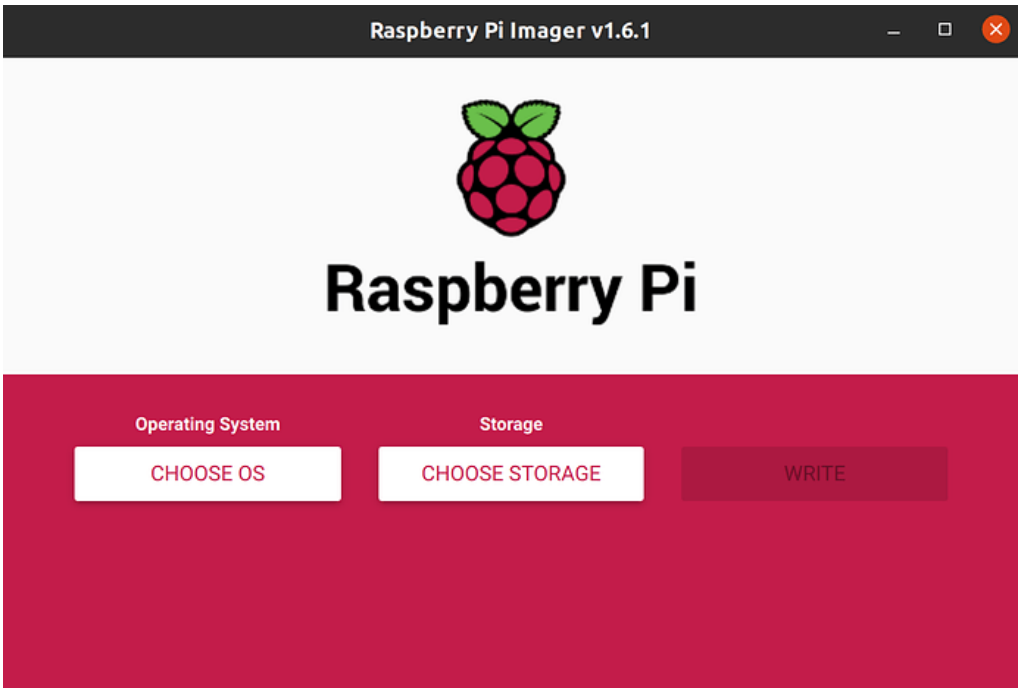
Table of Contents

Initial Setup

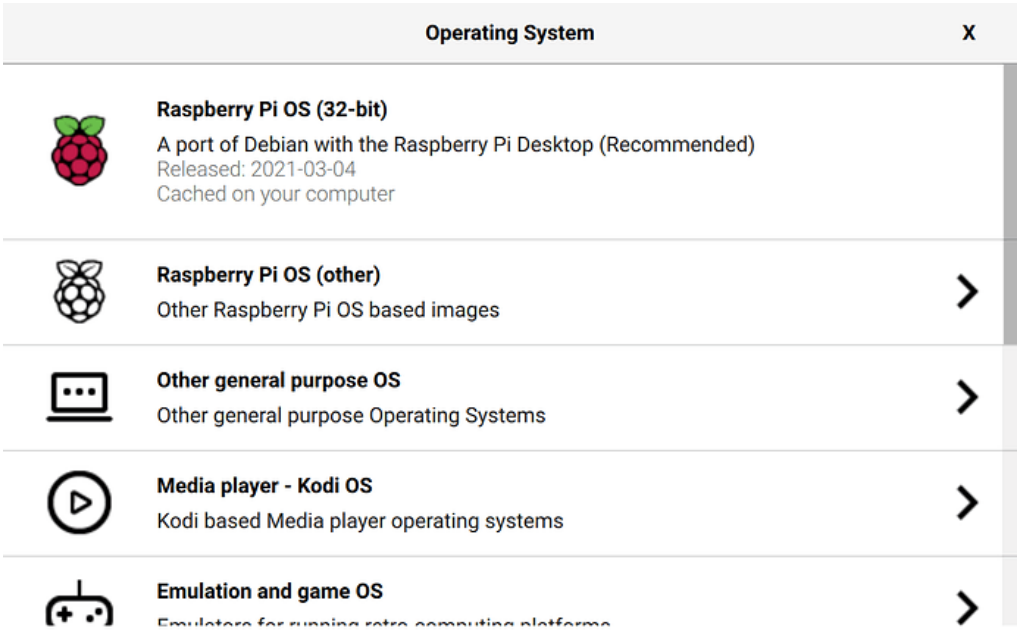
Flash SD Card with Ubuntu 24.04 Server

1. Install the right Raspberry Pi Imager for your operating system: https://www.raspberrypi.com/software/

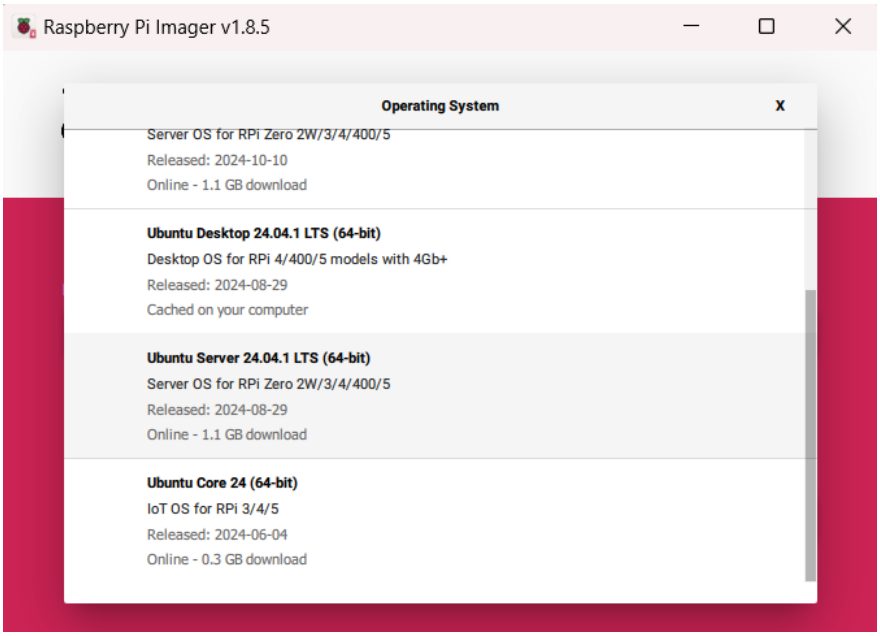
2. Start the Imager and open the “CHOOSE OS” menu



3. Scroll down the menu click “Other general-purpose OS”

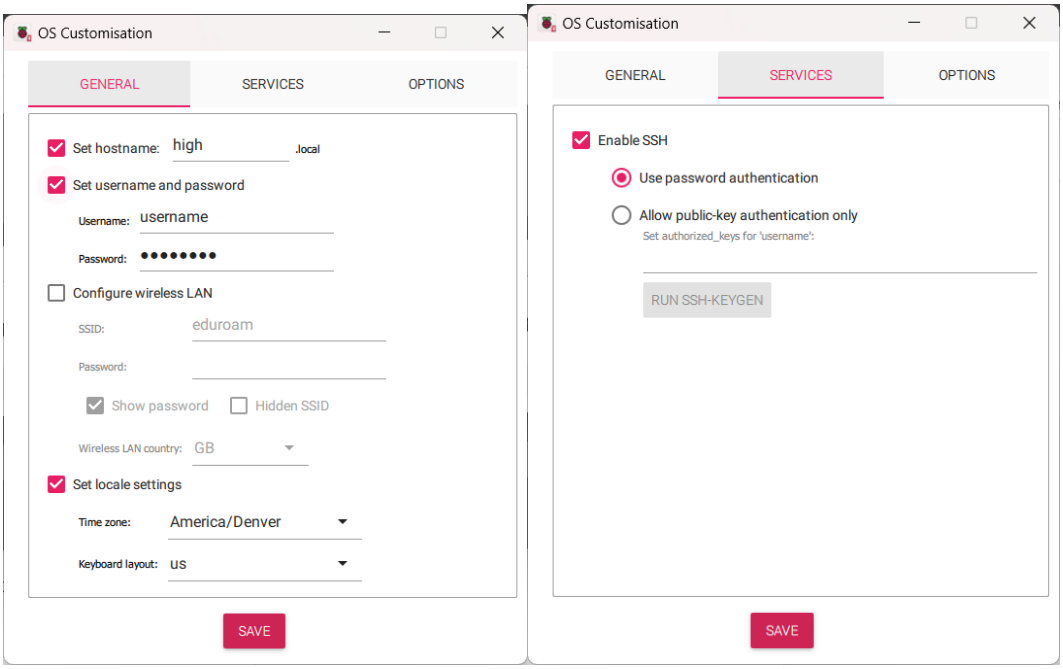


4. Select the “Ubuntu Server 24.04 (Raspberry Pi 5)” option



5. Open the “SD Card” menu and select the microSD card you have inserted, and click “WRITE”. Once complete, make sure your Pi is off and insert this SD card.

6. Go through the OS customization options to configure the OS.
   Note. you are unable to connect to eduroam this way due to the authentication it requires.



Boot Ubuntu and Set Up SSH

1. Ensure your HDMI screen and a USB keyboard are plugged into the Raspberry Pi 5 before plugging in and powering on
2. Complete the boot process and prompted setup
3. Update the system:
    i. sudo apt update -y && sudo apt upgrade -y && sudo apt full-upgrade –y
4. Install essential tools and dependencies:
    i. sudo apt install -y git curl wget build-essential openssh-server software-properties-common python3-pip
5. Enable and start ssh:
    i. sudo systemctl enable ssh
    ii. sudo systemctl start ssh
6. Reboot the system:
    i. sudo reboot
7. After reboot, check the IP address:
    i. ip addr

Enable UART0 on RPi

On Raspberry Pi 5, the primary UART appears on the Debug header. All other UARTs are disabled by default and need to be enabled. We specifically need to enable UART0.

1. Install raspi-config:
    i. sudo apt-get install raspi-config
2. Open raspi-config:
    i. sudo raspi-config
3. In raspi-config:
    i. Go to "Interface Options"
    ii. Select "Serial Port"
    iii. Choose "No" to disable serial login shell
    iv. Choose "Yes" to enable the serial interface
    v. Select "Finish" and choose to restart the Raspberry Pi
4. After reboot, edit the boot configuration file:
    i. sudo nano /boot/firmware/config.txt
5. Enable UART0 by adding the following lines of code to the bottom of the file:
    i. NOTE: Other instructions will tell you to put “`dtoverlay=disable-bt", but doing so on the Raspberry Pi 5 will not let you connect to the PixRacer Pro!`
    ii. enable_uart=1

iii. dtoverlay=uart0

6. Exit the file by typing ctrl+x, ctrl+y, and enter
7. Reboot the RPi:
    i. sudo reboot
8. Verify the serial port is available:
    i. ls /dev/ttyAMA0

The serial port should now be available as /dev/ttyAMA0 (also accessible as /dev/serial0).

ROS 2 "Jazzy" Installation

1. Set up ROS 2 Jazzy repositories on the RPi:
    a. sudo add-apt-repository universe
    b. sudo apt update && sudo apt install -y curl gnupg lsb-release
    c. sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
    d. echo "deb [arch=$(dpkg --print-architecture) 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
2. Install ROS 2 Jazzy base and development tools:
    a. sudo apt update
    b. sudo apt install -y ros-jazzy-ros-base ros-dev-tools

RESUME INSTRUCTIONS HERE

3. Install Python dependencies:
    a. sudo apt install pipx
    b. pip install --user --break-system-packages -U empy==3.3.4 pyros-genmsg setuptools
4. Install colcon build system and additional tools:
    a. sudo apt install -y python3-colcon-common-extensions python3-rosdep
    b. python3-vcstool
    c. sudo rosdep init
    d. rosdep update

ROS 2 Workspace Setup

1. Create a ROS 2 workspace and clone necessary repositories on the RPi:
    a. mkdir -p ~/ros2_ws/src
    b. cd ~/ros2_ws/src
    c. git clone https://github.com/PX4/px4_msgs.git

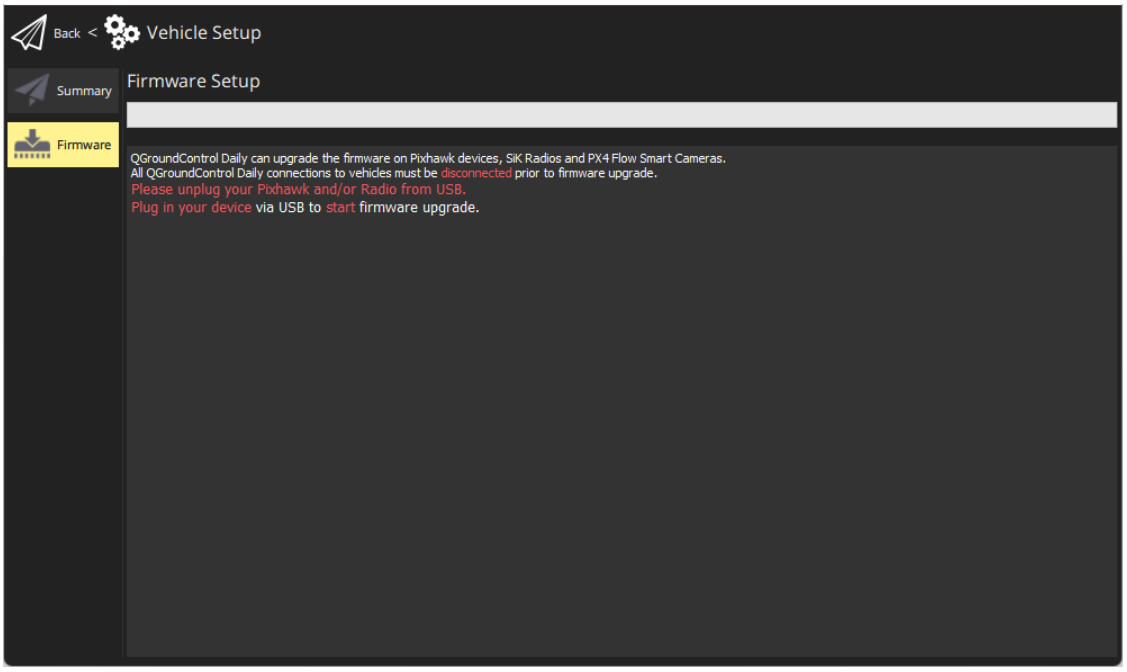
    d. git clone https://github.com/PX4/px4_ros_com.git
    e. cd ..
2. Install dependencies and build the workspace:
    a. source /opt/ros/jazzy/setup.bash
    b. sudo rosdep init
    c. rosdep update
       rosdep install --from-paths src --ignore-src -r –y
    d. export MAKEFLAGS="-j4"
    e. colcon build --parallel-workers 4
3. Set up environment to source ROS 2 and workspace on terminal startup:
    a. echo "source /opt/ros/jazzy/setup.bash" >> ~/.bashrc
    b. echo "source ~/ros2_ws/install/setup.bash" >> ~/.bashrc
    c. source ~/.bashrc
4. Verify the ROS 2 setup:
    a. echo $ROS_PACKAGE_PATH
    b. ros2 --version

PixRacer Pro Setup

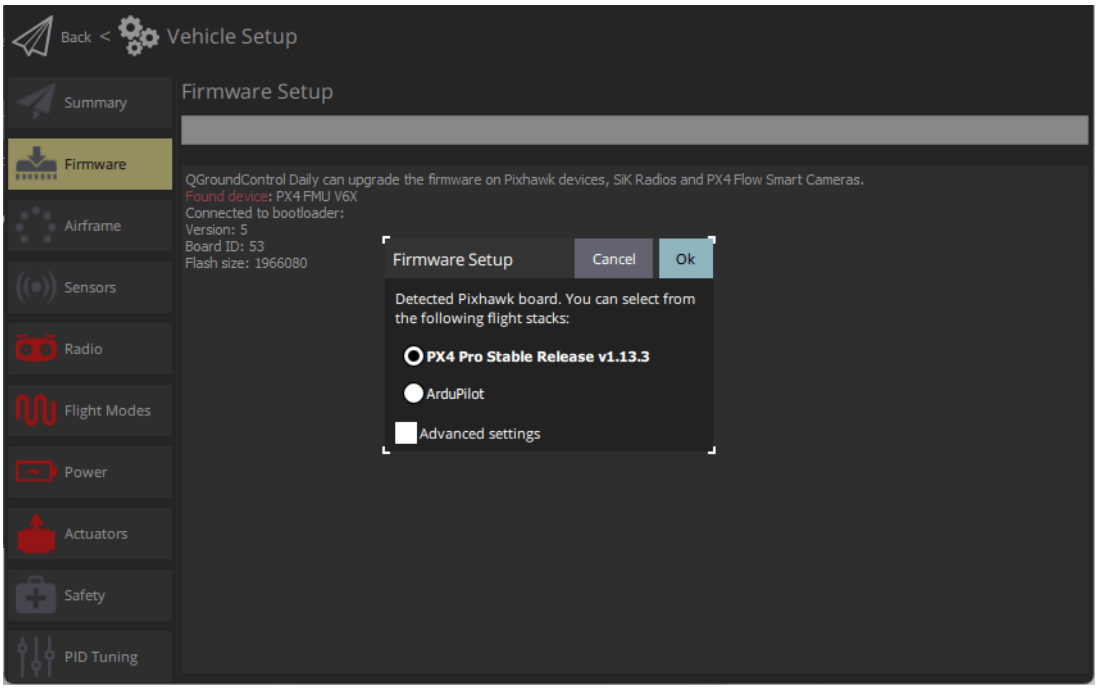
Download QGroundControl onto a computer you will use to connect to your PixRacer Pro to download the firmware: https://docs.qgroundcontrol.com/master/en/qgc-user-guide/getting_started/download_and_install.html

1. Start *QGroundControl* and connect the vehicle

2. Select **"Q" icon > Vehicle Setup > Firmware** (sidebar) to open *Firmware Setup*

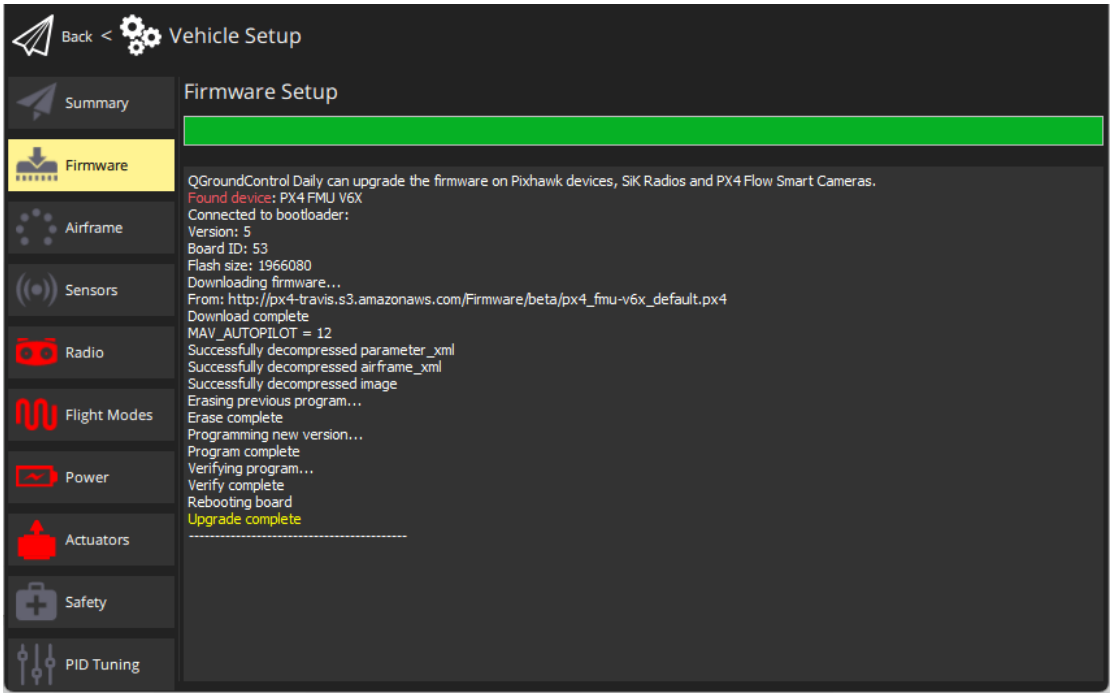


3. Connect the flight controller directly to your computer via USB
4. Select the **PX4 Pro Stable Release vX.x.x** option to install the latest stable version of PX4 *for your flight controller* (autodetected)



5. Click the **OK** button to start the update. The firmware will then proceed through a number of upgrade steps (downloading new firmware, erasing old firmware etc.). Each

step is printed to the screen and overall progress is displayed on a progress bar.



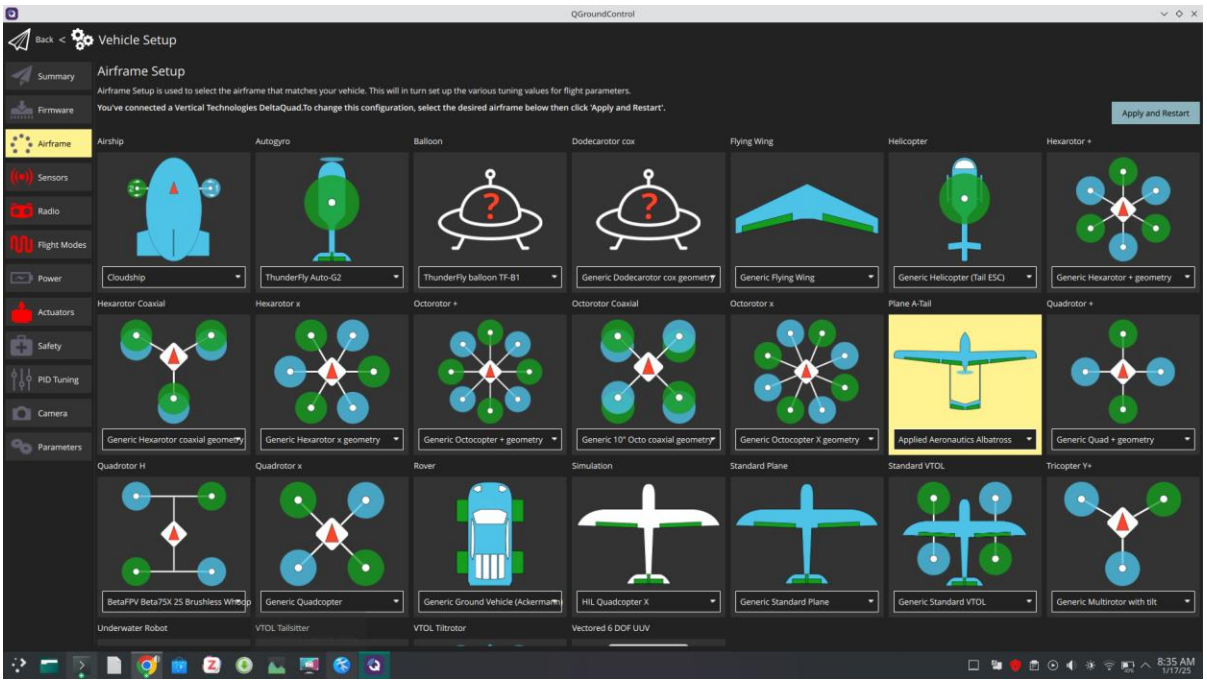
6. Once the firmware has completed loading, the device/vehicle will reboot and reconnect.

After installing the firmware you need to select a vehicle type and frame configuration. The vehicle type can be found here:
https://docs.px4.io/main/en/airframes/airframe_reference.html

Once you determine the vehicle type this can be set in Qgroundcontrol.

1. Select "Q" icon > Vehicle Setup > Airframe (sidebar) to open Airframe Setup.

2. Select the broad vehicle group/type that matches your airframe and then use the dropdown within the group to choose the airframe that best matches your vehicle.



The example above shows *Applied Aeronautics Albatross* which is what we used for our High-Altitude drone.

3. Click Apply and Restart. Click Apply in the following prompt to save the settings and restart the vehicle.

The PX4 firmware then needs to be configured to use ROS 2 instead of MAVLINK. To update parameters, follow this guide: https://docs.px4.io/main/en/advanced_config/parameters.html. The following parameters need to be changed:

```
MAV_1_CONFIG = 0 (Disabled)
UXRCE_DDS_CFG = 102 (TELEM2)
SER_TEL2_BAUD = 921600
```

MAV_1_CONFIG=0 and UXRCE_DDS_CFG=102 disable MAVLink on TELEM2 and enable the uXRCE-DDS client on TELEM2, respectively. The SER_TEL2_BAUD rate sets the comms link data rate. Check that the uxrce_dds_client module is now running. You can do this by running the following command in the QGroundControl MAVLink Console:

- uxrce_dds_client status

If the client module is not running you can start it manually in the MAVLink console:

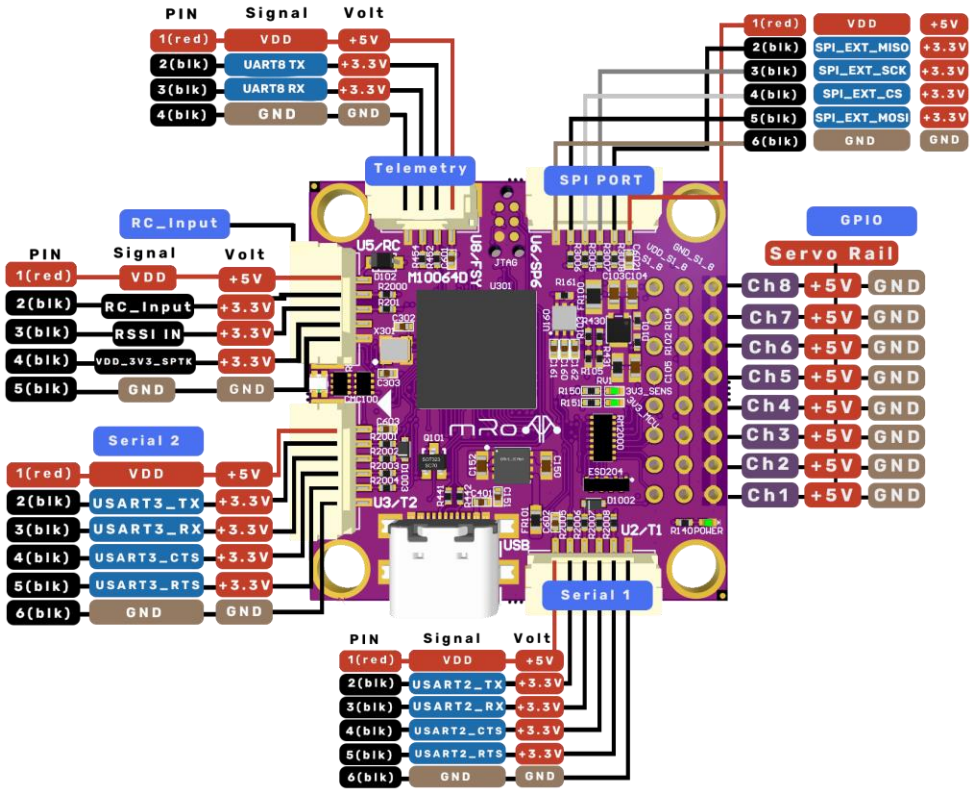
- uxrce_dds_client start -t serial -d /dev/ttyAMA0 -b 921600

RPi and PixRacer Pro Wiring

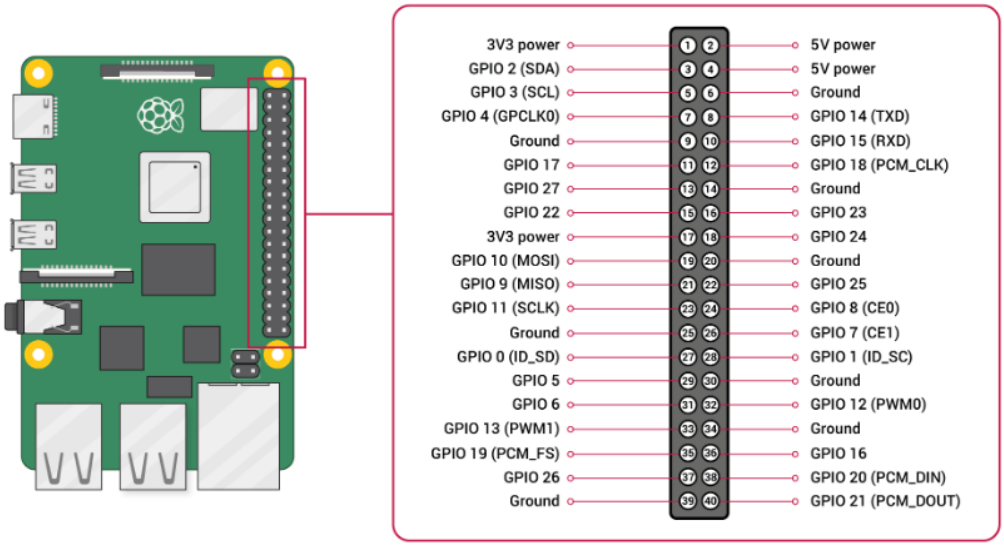
Wire up the serial connection between the RPi and PX4 that is to be used for offboard control. Connect the PixRacer Pro TELEM2 TX/RX/GND pins to the complementary RXD/TXD/Ground pins on the RPi GPIO board:

PX4 TELEM2 Pin	**RPi GPIO Pin**
UART5_TX (2)	RXD (GPIO 15 - pin 10)
UART5_RX (3)	TXD (GPIO 14 - pin 8)
GND (6)	Ground (pin 6)

The following diagram shows the top view of the PixRacer pro, the TELEM2 port pins are the “Serial 2” pins.



This diagram shows the port pins for the RPi.



uXRCE_DDS Agent Installation

On the RPi, install the Micro XRCE-DDS Agent to communicate with the PixRacer Pro.

1. Install dependencies:
    a. sudo apt install -y cmake gcc g++ git libssl-dev
2. Clone and build Micro-XRCE-DDS-Agent:
    a. cd ~
    b. git clone https://github.com/eProsima/Micro-XRCE-DDS-Agent.git
    c. cd Micro-XRCE-DDS-Agent
    d. mkdir build
    e. cd build
    f. cmake ..
    g. make -j4
    h. sudo make install -j4
    i. sudo ldconfig /usr/local/lib/
    j. cd ../..

uXRCE_DDS Agent Activation

Start the uXRCE_DDS agent in the RPi terminal. To run the Micro-XRCE-DDS Agent, use the following command on the RPi:

- sudo ~/Micro-XRCE-DDS-Agent/build/MicroXRCEAgent serial --dev /dev/ttyAMA0 -b 921600

Both the agent and client should be running, and you should see activity on both the MAVLink console and the RPi terminal. You can view the available topics using the following command on the RPi:

- source /opt/ros/jazzy/setup.bash
- ros2 topic list

Make sure to rebuild your workspace after making any changes by doing the following:

- cd ~/ros2_ws
- colcon build
- source ~/.bashrc

To automatically start the MicroXRCE Agent on Ubuntu 24.04 startup, you can create a systemd service.

1. Create a new systemd service file:
    a. sudo nano /etc/systemd/system/microxrce-agent.service
2. Add the following content to the file:
    a. NOTE: PLEASE MAKE SURE TO CHANGE THE USERNAME IN THE FILE PATH TO YOUR USER USERNAME, OR IT WILL NOT WORK

```
[Unit]
Description=MicroXRCE Agent
After=network.target

[Service]
ExecStart=/home/my_username/Micro-XRCE-DDS-Agent/build/MicroXRCEAgent serial --dev /dev/ttyAMA0 -b 921600
Restart=always
User=root

[Install]
WantedBy=multi-user.target
```

3. Reload systemd to recognize the new service:
    a. sudo systemctl daemon-reload
4. Enable the service to start on boot:
    a. sudo systemctl enable microxrce-agent.service
5. Start the service:
    a. sudo systemctl start microxrce-agent.service
6. Check the status with:
    a. sudo systemctl status microxrce-agent.service
7. Reboot the system:
    a. sudo reboot
    b. ros2 topic list

c. ros2 topic echo /fmu/out/vehicle_status

This setup ensures that the agent runs with root privileges, which may be necessary for accessing the serial device. This should allow the MicroXRCE Agent to automatically start on system boot.

SSH setup with GUI to view Images/Graphs

On Raspberry Pi:

- Install X11 Forwarding Tools
  sudo apt update && sudo apt upgrade -y
  sudo apt install x11-apps xauth xorg
- Edit the SSH Configuration File
  sudo nano /etc/ssh/sshd_config
  - Ensure the following lines are set and/or uncommented in the file
    X11Forwarding yes
    X11DisplayOffset 10
    X11UseLocalhost yes
- Restart the SSH Service

Sudo systemctl restart ssh

On Windows Laptop:

- Install an X server, such as Xming or VcXsrv.
  - Configure the X server to allow connections and start it before using PuTTY.
- **Open PuTTY**
  - Start PuTTY and enter the Ubuntu machine's hostname or IP address in the **Host Name (or IP address)** field.
- **Enable X11 Forwarding**
  - In the left-hand menu, navigate to **Connection > SSH > X11**.
  - Check the box labeled **Enable X11 forwarding**.
  - Enter localhost:0 in the **X display location** field (this is usually the default).
- **Save the Session**
  - Go back to the **Session** category, give the session a name, and click **Save** to save the settings.

Servo Control Using RPi

Install the following on the RPi:

- sudo apt install python3-gpiozero

- sudo apt-get install python3-rpi.gpio

Attach the Servo PWM pin to GPIO 25 (pin 22 on the RPi). The Servo needs to be powered by an external 5V power source because it draws too much current from the RPi. The RPi and the external power source do need to share the same ground, however. Once the Sevo has the correct wiring, run the following in a python script:

```
from gpiozero import Servo
from time import sleep

servo = Servo(25)

val = -1

try:
        while True:
                servo.value = val
                sleep(0.1)
                val = val + 0.1
                if val > 1:
                        val = -1
except KeyboardInterrupt:
        print("Program stopped")
```

Appendix

Useful links divided by section

Flash SD Card with Ubuntu 24.04

→ https://ubuntu.com/tutorials/how-to-install-ubuntu-desktop-on-raspberry-pi-4#2-prepare-the-sd-card

Boot Ubuntu Desktop and Set Up SSH

→ https://averagelinuxuser.com/how-to-install-and-use-ssh-on-linux/

Enable UART0 on RPi

→ https://www.raspberrypi.com/documentation/computers/configuration.html#cm1-cm3-cm3-and-cm4

ROS 2 “Jazzy” Installation & ROS 2 Workspace Setup

→ ROS 2 installation and workspace setup: https://docs.ros.org/en/jazzy/Installation/Ubuntu-Install-Debs.html#install-ros-2

→ ROS 2 examples: https://github.com/ros2/examples

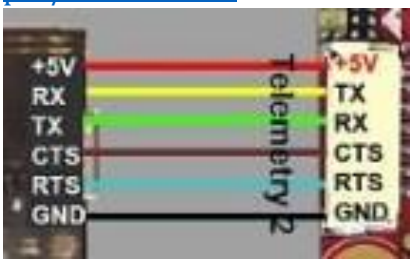
PixRacer Pro Setup

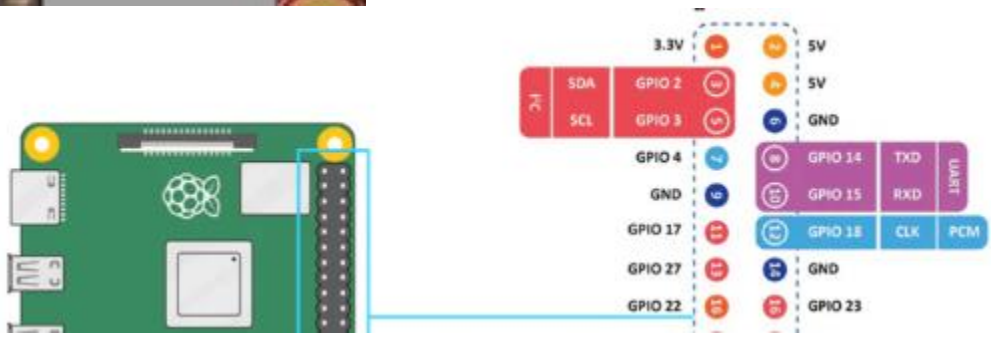
→ Qgroundcontrol firmware installation: https://docs.px4.io/main/en/config/firmware.html#install-stable-px4
→ Qgroundcontrol vehicle selection: https://docs.px4.io/main/en/config/airframe.html
→ Airframe type reference: https://docs.px4.io/main/en/airframes/airframe_reference.html
→ How to change parameters in Qgroundcontrol guide: https://docs.px4.io/main/en/advanced_config/parameters.html
→ TELEM 2 parameter config: https://docs.px4.io/main/en/companion_computer/pixhawk_rpi.html#ros-2-and-uxrce-dds

uXRCE_DDS Agent Installation & uXRCE_DDS Agent Activation

→ https://docs.px4.io/main/en/companion_computer/pixhawk_rpi.html#ros-2-and-uxrce-dds

RPi and PixRacer Pro Wiring

→ RPi and PixRacer pro wiring: https://docs.px4.io/main/en/companion_computer/pixhawk_rpi.html#wiring
→ PixRacer Pro hardware layout: https://docs.3dr.com/autopilots/pixracer-pro/#downloads
→ 

→ 

Extra links

ROS 2 “Jazzy” Integration with PX4

- ROS 2 Integration with PX4: https://docs.px4.io/main/en/ros2/
- uXRCE-DDS (PX4-ROS 2/DDS Bridge): https://docs.px4.io/main/en/middleware/uxrce_dds.html
- uORB Message Reference:
- https://docs.px4.io/main/en/msg_docs/
- ROS 2 User Guide (with PX4): https://docs.px4.io/main/en/ros2/user_guide.html
- PX4-Autopilot/IntegrationTests: https://github.com/PX4/PX4-Autopilot/tree/main/integrationtests/python_src/px4_it/mavros

PX4

- PX4 Autopilot User Guide: https://docs.px4.io/main/en/
- MAVLink Messaging: https://docs.px4.io/main/en/middleware/mavlink.html#mavlink-overview
- UAV Data Transmission and Protocols PowerPoint: https://robolabor.ee/img/cms/projektid/UAV%20Data%20Transmission%20and%20Communication%20Protocols.pdf
- General PixRacer Documentation: https://bkueng.gitbooks.io/px4-user-guide/content/en/flight_controller/pixracer.html