UAV setup guide

The following guide describes how to set up and configure the UAV (or drone). The vehicle has been set up using PX4 Autopilot, a flight stack or firmware/operating system, which is installed on the flight controller. The behavior and flight of the vehicle are configured via QGroundControl installed on a computer and through manually configuring the Radio Controller (RC).

Hardware components

The drone uses the following hardware and electronic components:

- Flight Controller (FC) aka. Autopilot: Pixhawk 6X
- Power Distribution board: PM03D Power Module
- Companion/Onboard computer: Raspberry Pi 4 Model B, 4GB RAM
    - Power cord: 4pin JST GH to USB Type C
    - Ethernet: JST to RJ45 cable
- GPS module: M9N GPS Module
- Motor: T-Motor MS2820-7 830KV
- Propellers: APC Sport 13x4 (inches)
- 360-camera: Ricoh Theta X
    - Power cord: 5m USB-C to USB-C cable, and USB-C to USB-A adapter
- Computer vision camera: OAK-D Pro
- Radio receiver on the drone: RadioMaster R81 Receiver
- Radio Controller: Radiomaster TX12 Mark II Radio Controller
- Battery: LIPO BATTERI 4S 14.8V 11000MAH 100C EC5 KONTAKT GENS ACE
    - Connected with EC5 male/female connectors, 5 m 10AWG power cable, and a cable sock.
- Three different Electronic Speed Control (ESC)**:
    - EMAX BLHeli 20A 2-4S OneShot125
    - Aikon AK32 35A 2-4S BlHeli32 ESC
    - X500 V2-BLHeli S 20A ESC
- Drone frame (body, arms, legs, etc.) designed and printed by Bengt Erik Gustafsson, from IDT in MDU.
- Various cables and connectors (XT-30, JST, etc.) for connecting the electronic components.

***ESC cannot be mixed and matched. It is recommended to only use one kind, e.g., 4x Aikon ESCs for the quadcopter.*

The hardware can be found here:
https://studentmdh.sharepoint.com/:x:/r/sites/STAD/_layouts/15/doc2.aspx?sourcedoc=%7B692053DC-E5C1-44CB-B161-7BBEFF9722F1%7D&file=System_Specification.xlsx&action=default&mobileredirect=true

Documentation and guides

The following main guides were used to set up the UAV.

Main PX4 documentation: https://docs.px4.io/main/en/

Pixhawk 6X Assembly guide: https://docs.px4.io/main/en/assembly/quick_start_pixhawk6x.html

Standard configuration guide: https://docs.px4.io/main/en/config/

Multi-copter/Quadcopter configuration: https://docs.px4.io/main/en/config_mc/

Advanced setup: https://docs.px4.io/main/en/advanced_config/

Support Discord/Forum information: https://docs.px4.io/main/en/contribute/support.html

ArduPilot (not installed, used as a reference): https://ardupilot.org/ardupilot/

Along with YouTube, ChatGPT, google, etc.

Setup guide

These steps explain how the UAV was configured from scratch. Note that it might not be necessary to follow these steps exactly in order and that some steps will have to be continuously configured/adjusted/fine-tuned to make the vehicle more reliable and fly more stable.

Schematics

The following picture roughly explains how the hardware is connected. Note that the Ricoh Theta X camera is attached to the drone, but not connected to any electronics on the UAV. Additionally, the power to the Raspberry Pi (JST to USB-C) is missing in this picture.

Remove propellers from motors

When configuring the actuators, for safety reasons, remove the propellers. The vehicle will not take off when it is armed, i.e., when the motors spin without any throttle. However, it is not advisable to have them on when configuring the vehicle.

Install the latest QGroundControl**

Connect the computer with the installed QGroundControl software to the PixHawk Flight Controller (FC) via the USB-A to USB-C cable.

Download and installation instructions:
https://docs.qgroundcontrol.com/master/en/getting_started/download_and_install.html

It takes about 30 seconds for the connection between the computer and the FC to be established. The LEDs on the FC should blink blue/green/red, indicating that it has power over the USB cable. A green bar will fill up at the top of the QGroundControl program during the establishment of the connection. After successfully connecting to the FC, the device is accessible to be configured.

*****Note**: There are several different ground control applications. QGroundControl is just one option that PX4/Pixhawk recommends and that was used during the project.*

Load PX4's latest stable firmware** to the flight controller via QGroundControl

The firmware is open source, meaning that it is possible to modify it as needed. However, there are many security features built-in with the firmware, e.g., to avoid crashing the vehicle. Consequentially, it might be difficult to modify the source code to some extent.

*****Note**: The PixHawk Flight Controller also supports other firmware, such as ArduPilot.*

Reference: https://docs.px4.io/main/en/config/firmware.html and https://github.com/PX4/PX4-Autopilot

Airframe

Set the frame as Generic Quadcopter for the custom build drone, since we do not use any predefined model. Use the "Generic" frame type that most closely matches geometry with the number of motors and their relative position. It is possible to customize the frame type to change geometry, specify FC outputs, etc. The frame configuration file (on the FC's UNIX file system) contains all predefined (and customizable) parameters. This can be customized from QGroundControl instead of using the configuration file directly.

Reference: https://docs.px4.io/main/en/config/airframe.html and
https://docs.px4.io/main/en/airframes/airframe_reference.html#quadrotor-x

Power setup / ESC Calibration

Every time the type of ESC is changed, the ESC Calibration and ESC Pulse width modulation (PWM) Minimum and Maximum Calibration must be performed. Disconnect the battery, go to "ESC Calibration", and click on "Calibrate", connect the battery and the calibration process will automatically be performed. You should note that the speaker in the ESCs and motors play a startup sound.

ESCs may support many different protocols such as PWM, DShot, OneShot, and CAN. The current version of the drone uses Oneshot (See Actuator setup). PWM is the oldest one-way protocol, and DShot is the newest two-way protocol. Newer protocols such as Dshot support changing motor spin direction through software. Only PWM and OneShot ESCs need to be calibrated.

Reference: https://www.youtube.com/watch?v=okUrPLP_Eec and
https://docs.px4.io/main/en/advanced_config/esc_calibration.html

It might be possible to directly connect and configure the ESCs via software such as BLHeli_S, BLHeli_32, or similar. With this software, it should be theoretically possible to, e.g., remove the startup sound, lower the sound volume, change the protocol from PWM to DShot, reverse motor spin, etc. However, during the project, it was not possible to establish a connection via the COM port and configure the ESC manually.

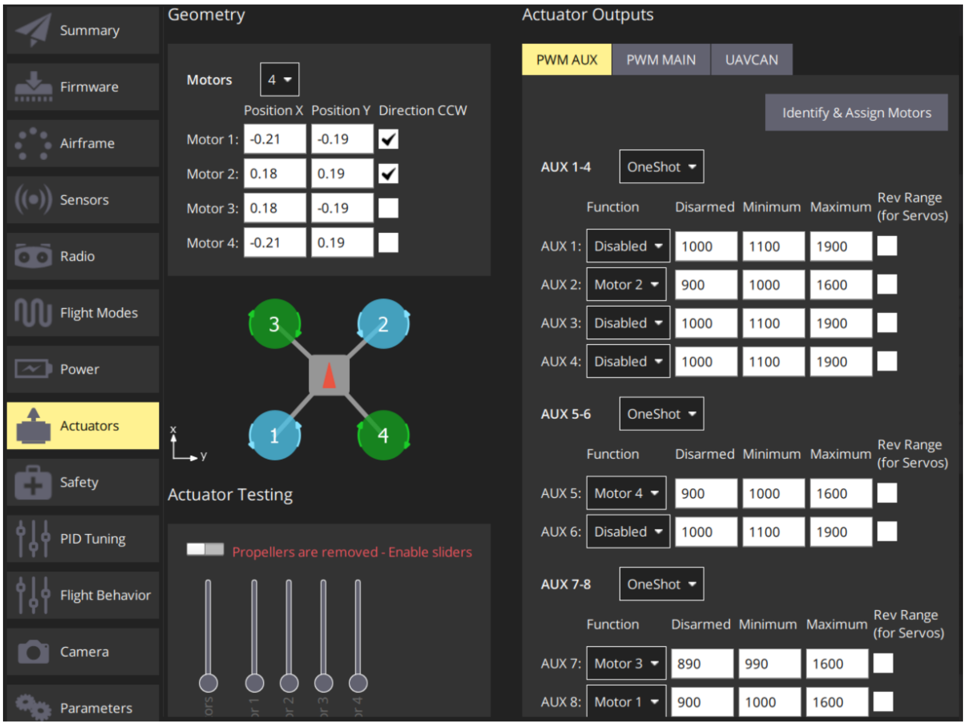
Reference: https://oscarliang.com/connect-flash-blheli-s-esc/

X500 V2-BLHeli S 20A ESC reference: https://holybro.com/products/spare-parts-qav250-kit?variant=41591153000637

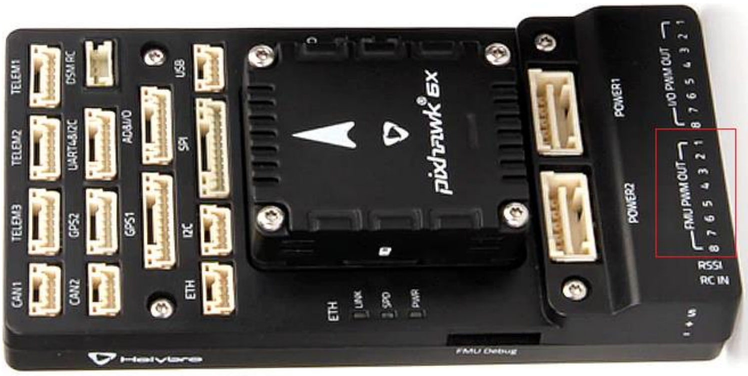
Actuator Output

To configure the direction of motor spin when using PWM or OneShot, you need to manually cut and solder two of the connecting cables. In the picture below, cables 1, 2, and 3 are connected so that the motor spins in a counterclockwise direction. To change the direction, cut cables 1 and 3 and switch their position on the motor side through soldering.

In QGroundControl, the following settings are currently in use:



Note that AUX2, AUX5, AUX7, and AUX8 in the GUI represent FMU PWM OUT pins 2, 5, 7, and 8 on the FC. Note that it was not possible with the current configuration to use AUX1, 2, 3, and 4.



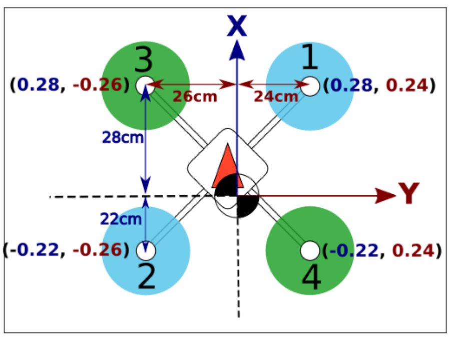
The values for “Disarmed”, “Minimum”, and “Maximum” represent the PWM duty cycle in microseconds (µs). To set the correct values, you must manually start all motors from Actuator Testing, look at each motor, and listen to the sound they produce, where the goal is to have all motors have equal Revolutions per minute (RPM) when the motors are given the same amount of output. For example, at the minimum output, all motors should spin relatively slowly, and equally, and produce the same sound. The disarmed value should be less than the minimum value and cause the motors to not spin. At maximum, they should not cause the ESC to overheat, but given enough output for liftoff. It is currently set to the minimally allowed value.

Note that, when the propellers are mounted on the motors, the motors will experience additional wind resistance which must be considered for successful liftoff.

The direction of the motor spin should be equal on the diagonal side of each motor, as seen by the two green-colored motors spinning clockwise, and the two blue motors spinning counterclockwise. It is **very important** to verify that the direction of the motor spin is correct, by manually turning the motors on and off and looking at, or feeling, the direction. The spin direction should also be set in QGroundControl. The drone will flip if this is incorrect.

Note*: If the weight or the distribution of the weight is changed at any point, you need to recalibrate the vehicle geometry.*

Define vehicle geometry, i.e., the motor's relative position to the center of gravity. It can be measured by determining the X and Y position of the center of gravity, and then measuring the X and Y distance to the center of each motor as described in the picture below. Note that, in QGroundControl, it is only possible to set the values in meters with two decimal precisions, e.g., 0.20. Consequentially, the measurements do not need to be very precise.



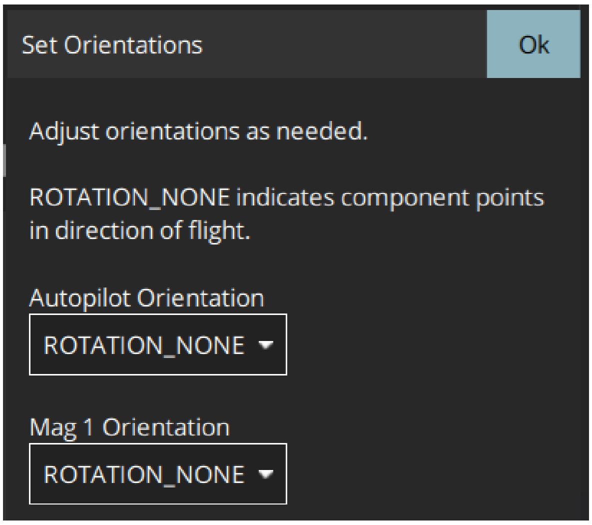
Reference: https://docs.px4.io/main/en/config/actuators.html and https://docs.px4.io/main/en/config/actuators.html#motor-geometry-multicopter

The Z-position is the axis down (towards the ground) and is difficult to measure when using a tethered drone since the cable length is variable until set at some altitude. However, if a certain altitude is determined, the center of gravity of the drone should be calculated while the tether is attached to the drone. For example, if we decide that the tether length should be a maximum of 2 m, we can try to use a ladder and attach the drone to a rope attached to the ceiling to determine the Z-position.

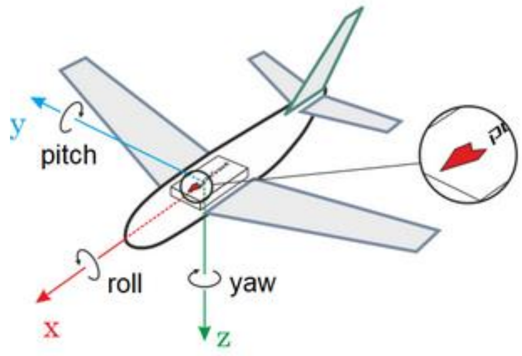
Reference: https://kth.diva-portal.org/smash/get/diva2:1530843/FULLTEXT01.pdf

Sensor orientation

In the current configuration, the FC is not rotated in any axis, meaning that the printed arrow on the FC is facing upwards (towards the sky) and is pointing toward the front of the vehicle.



To understand the aircraft flight dynamics, i.e., the orientation of the vehicle in three dimensions, the picture below can be used as a reference. The rotational angles, known as pitch, roll, and yaw are collectively known as attitude, controlled by the FC's attitude controller.



Therefore, if the FC is mounted in some other way, we need to configure this accordingly in QGroundControl under “Sensors” → “Set Orientations”. If an external compass is used, i.e., the M9N GPS Module, the module's direction also needs to be correctly configured according to the printed arrow on the GPS module.

Reference: https://docs.px4.io/main/en/config/flight_controller_orientation.html

Calibrate compass

Post initial configuration, the compass might need recalibration if:

- The GPS is exposed to strong magnetic fields and/or large metal objects. Signs of bad calibration are that the drone is flying poorly, e.g., spinning, circling, bowling, veering off-path, etc.
- Payload is changed – perform quick calibration.
- There are error messages when trying to arm the vehicle and it will not take off, as seen by pressing the “Speaker” icon here:

It is recommended to connect to the vehicle via telemetry radio, if possible, rather than USB. Note that we currently do not have any telemetry radio. The external GPS should be mounted as far away as possible from other electronics, e.g., on a plastic mast. After the configuration, verify that the compass is pointing north (approximately) from QGroundControl.

Initial configuration: In QGroundControl go to: “Sensors” → “Compass” → On “Calibrate Compass” click “OK” → Follow the instructions by rotating the UAV according to the picture until the process is complete.

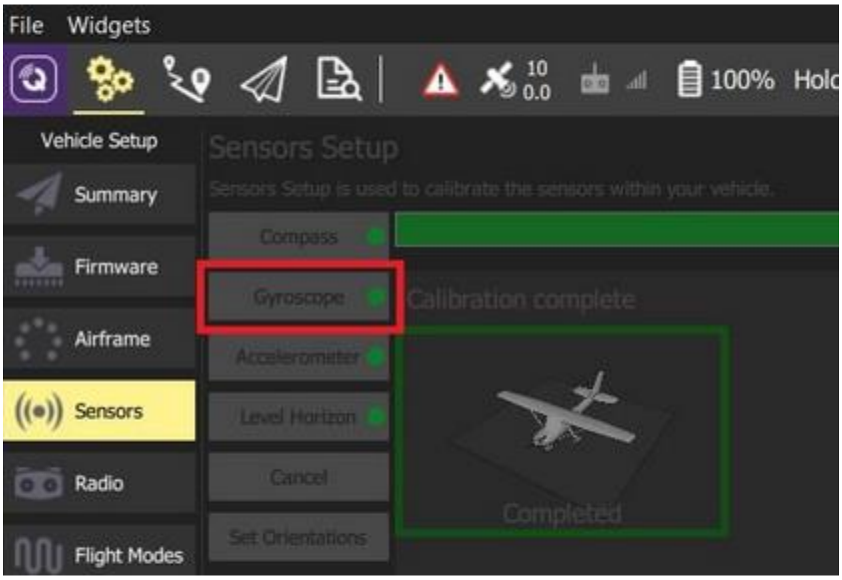
Quick configuration (post initial): When the vehicle is running and disarmed → rotate the UAV on all axes, 2-3 oscillations of ~30 degrees → wait for the heading estimate to stabilize. Verify compass rose is pointing in the correct direction, i.e., north when facing north.

Reference: https://docs.px4.io/main/en/config/compass.html

Gyroscope Calibration

Go to the “Gyroscope” setting and follow the instructions. Put the UAV on the table, click “Calibrate” and wait until the calibration is complete. Note that the vehicle is sensitive to external vibrations at this stage.

Reference: https://docs.px4.io/main/en/config/gyroscope.html

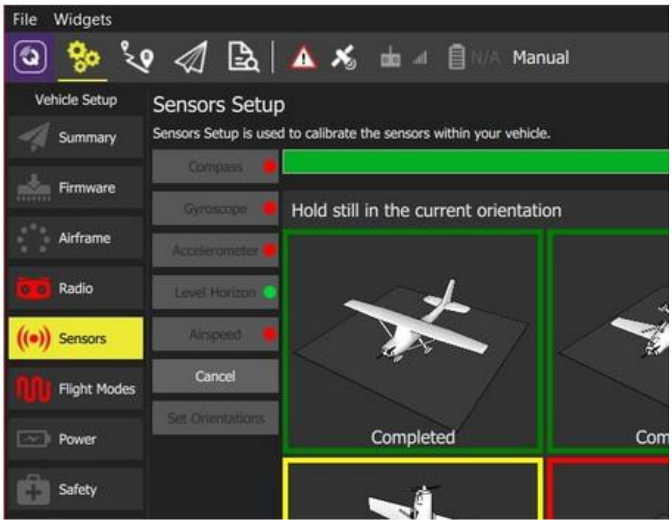


Accelerometer

Needs configuration the first time and if the FC position is changed.

Same as compass calibration, expect you to hold the vehicle still for each orientation picture that is shown. Note that it does not need to be exact.

Reference: https://docs.px4.io/main/en/config/accelerometer.html



Level Horizon

Used to fix small errors in orientation, and level the horizon of the vehicle. The artificial horizon can be seen in QGroundControl, below, where the green field represents the ground and the blue field the sky.

It is recommended to perform this calibration each time the drone is placed on the ground, especially if the ground is not level. If the drone is constantly drifting in some direction, this calibration can help in solving the issue.

Reference: https://docs.px4.io/main/en/config/level_horizon_calibration.html and https://dronesurveyservices.com/have-calibrate-my-drone-every-time-i-use/

Airspeed

This calibration can be **skipped**, since we do not use any airspeed sensor, nor do we use a Fixed-Wing vehicle (such as an airplane) or VTOL vehicle (such as a rocket, or helicopter). According to the PX4 documentation, this type of sensor may help in windy conditions, slow flight, and autonomous landing. According to the ArduPilot documentation (another flight stack, as opposed to PX4), this type of sensor is not recommended for new users.

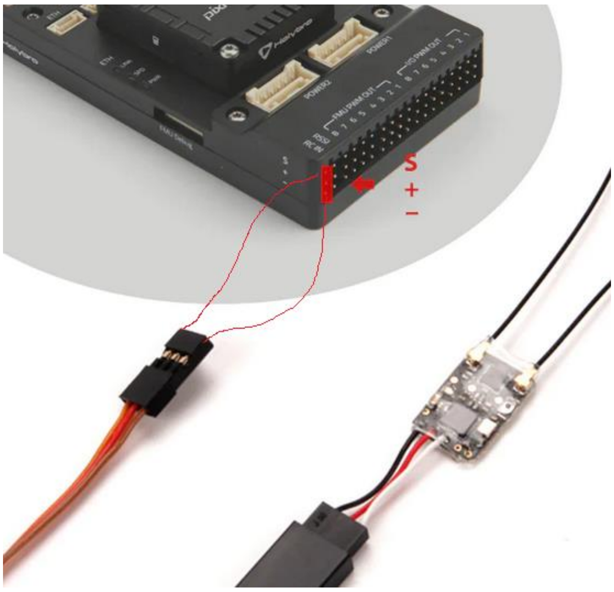
Reference: https://docs.px4.io/main/en/config/airspeed.html and https://ardupilot.org/plane/docs/airspeed.html

Radio Controller (RC) Setup & Calibration

The RadioMaster TX12MKII 2.4GHz RC can communicate over 16 radio channels, depending on the receiver.

The receiver is a RadioMaster R81 Receiver. Supports **8 channels**, 2400-2483.5 Mhz, in a FrSky D8 signal format.

Connect the Receiver on the FC as:



The first step is to bind the RC and the receiver – see specific HW instructions in the folder /HW Manuals for the files named RC_RECEIVER and RC_TRANSMITTER:

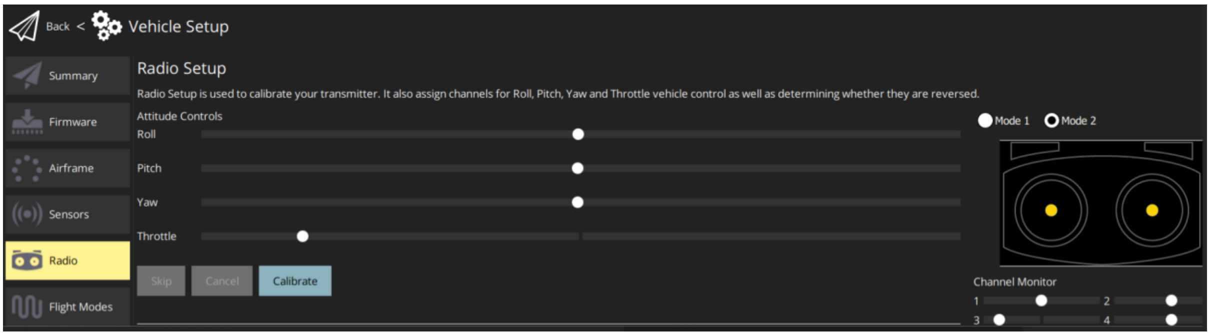
- Start RC by pressing and holding the power button. Press the MDL button → Press the PAGE > button → in SETUP, scroll down to Mode and select MULTI → in Type select FRSky D (or similar) → in Subtype select D8.
- Press and hold the BIND button on the receiver (small physical button on the circuit), then connect power to FC via USB, after 3 seconds the receiver LED will be RED, it is now in bind mode for a couple of seconds.
- On the RC, scroll to the [Bnd] option to start the binding process → The LED on the receiver will flash indicating successful binding.
- Exit bind mode on RC → Remove power from FC → Power the FC again → LED should now be a solid RED color, indicating that the receiver and transmitter are bound successfully.

It is also possible to fine-tune the RC settings (avoid loss of RC communication) according to the HW instructions documents in /HW Manuals:

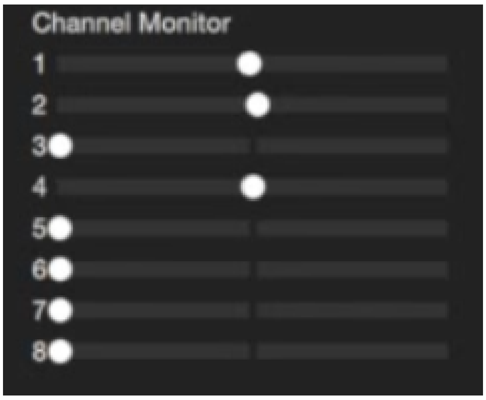
- RSSI output: Changed the frequency manually to first -57 and then +20, where it was noticed that the connection was lost between the receiver and transmitter. Then, set the Freqtune option in SETUP on the RC to $\frac{-57+20}{2} \approx -18$ to get a more stable radio connection, theoretically.
- Set a fail-safe mode on the RC receiver so that if the bind button is pressed, it doesn't accidentally cause the connection to be dropped.

In QGroundControl, configure the RC via Mode 2. This means that the left gimbal/stick is throttle, right is movement, which is standard for right-handed people.

Follow the Radio calibration guide in QGroundControl, i.e., follow the instructions and movement as shown in the picture for the left, and right gimbals, and the various switches on the RC during the calibration process. Move all the RC switches (physically marked as B, C, E, and F on the RC) through all their positions, one at a time.



Verify the channel association by moving the various controls on the RC and viewing that the correct channel is moving on the Channel monitor view. Note that if the Channel monitor is not visible, the probable cause is that the receiver is connected incorrectly, i.e., upside down. This will not damage the electronics, however.



It is also possible to limit the various controls' maximum/minimum output. This is useful for modifying the throttle, roll, pitch, or yaw. Limiting the throttle can help avoid overheating an ESC, however, enough throttle must still be available for takeoff**. This can be modified on the RC. On the RC → Press MLD → Press Page > until you reach the INPUTS page → Scroll down to and mark the setting you wish to modify, e.g., Thr (throttle) → Press and hold the scroll button → Edit → Change the weight to, for example, 70% and the offset to –30%, this will limit the maximum throttle to 40 (from 100).

******Note*** *that, in the current configuration, the throttle must be configured as 100% to takeoff in altitude mode.*

Reference: https://www.youtube.com/watch?v=Z4ozVGu3IZI and the HW documentation

Flight mode

Various flight modes can be configured to define how the vehicle responds to input from the RC and/or how the vehicle operates overall. Flight modes can be fully manual (Manual/Stabilized mode), semi-autonomous (Altitude mode), or fully autonomous (Mission mode, Takeoff mode). The different flight modes have different requirements, most requiring a GPS, whereas different modes have different degrees of difficulty to use.

Note that PX4 has some built-in safety logic, i.e., conditional statements that need to be fulfilled to change into different flight modes.

Due to the project being limited to not using a GPS (or other position fix), the available flight modes are also limited. The following list is the recommended modes to configure on the RC switches and use to operate the vehicle.

- Manual mode – Stabilizes the vehicle when the sticks are centered. Pitch and roll control the angle in the respective axis. This mode does not account for wind.
- Altitude mode – The vehicle will maintain the current altitude where the roll, pitch, and yaw control the movement at a set altitude. Note that this mode requires 62.5% throttle for takeoff and 50% throttle (centered) for holding the altitude. In windy conditions, the vehicle will drift.
- Offboard mode – The vehicle maneuvers according to some external source, such as an onboard or offboard companion computer running ROS or sending MAVLink messages.

Other configuration:

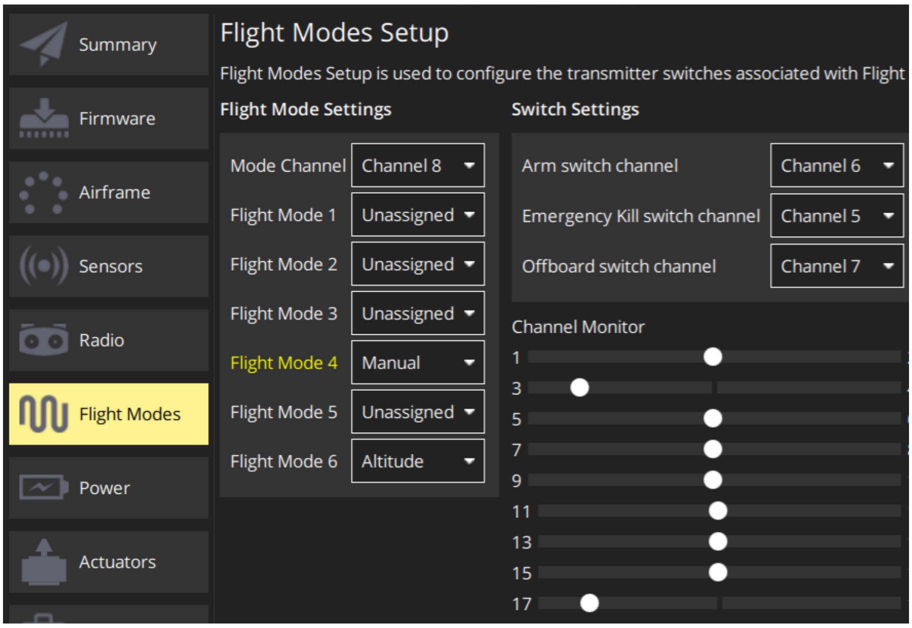
- Arm switch – Motors draw enough current to rotate, and the vehicle has full power.
- Kill switch – Immediately kill all motors, will cause the vehicle to crash, but is very useful in some situations.

Extra modes to use in the future with a GPS (or other position fix):

- Return mode – Return to launch position, as noted by GPS or marked in QGroundControl, through a safe path and land, which can be activated automatically (failsafe trigger) or via a switch on RC.
- Hold mode – Stop and hover in position (against wind forces etc.), used to pause the mission/regain control.
- Mission mode – Follow some predefined mission that is specified in QGroundControl, e.g., traveling from location A to B autonomously.
- Takeoff/Land mode – Causes takeoff or landing immediately when switched.
- Follow me mode – Autonomous tracking of something running QGroundControl or a MAVSDK application.

Reference: https://docs.px4.io/main/en/getting_started/flight_modes.html and https://docs.px4.io/main/en/flight_modes_mc/altitude.html and https://docs.px4.io/main/en/flight_modes_mc/manual_stabilized.html and https://docs.px4.io/main/en/flight_modes/offboard.html and https://docs.px4.io/main/en/advanced_config/prearm_arm_disarm.html

The RC switches configured earlier can be associated with different flight modes and behavior in QGroundControl, see the picture below for the current settings.



Using an available channel, move the switches on the RC to see what is highlighted in a yellow font (here, "Flight Mode 4") and associate this switch position with a desired flight mode or setting. Verify that the correct channel is used through the Channel Monitor, i.e., which switch is associated with the respective channels.

Power / Battery Setup

The power configuration mainly serves to provide an estimation of the remaining battery life to avoid sudden crashes in case the battery is completely discharged.

The current battery for the drone is 14.8 V, 4S2P, 11 000 mAh, and 162.8 Wh. Therefore the settings in QGroundControl should approximately reflect this, i.e.:

Number of cells (in series): 4

Empty voltage (per cell): 3.7

Full voltage (per cell): 4.2

Reference: https://docs.qgroundcontrol.com/master/en/SetupView/Power.html and https://docs.px4.io/main/en/config/battery.html

Safety Configuration (failsafe)

Through these settings, some security functions can be configured. For example, the behavior of the vehicle if the battery level is low, the behavior of the RC communication is lost, etc.

The only semi-relevant settings in the project are for the battery and the RC communication, the rest require GPS, Telemetry, etc. If the drone pilot always has a clear line of sight toward the vehicle, and the battery is charged, these settings are a non-issue.

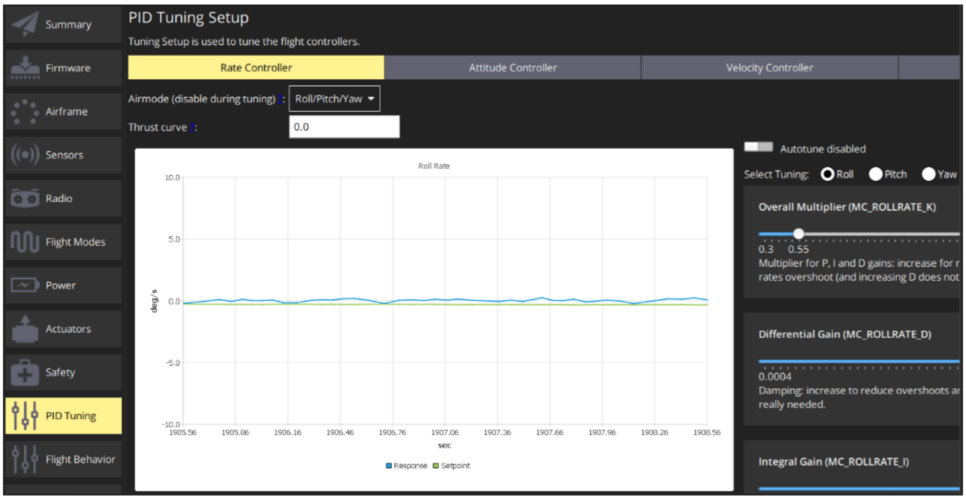
Reference: https://docs.px4.io/main/en/config/safety.html

PID Tuning Setup and flight log analysis

According to the PX4 documentation, the most important flight controllers for a stable vehicle are the **altitude** and **rate**. These will affect flight stability, such as twitching movement (oscillations), vehicle drift, vehicle responsiveness from pilot commands, and PID calculation overshoots. PX4 uses

PID controllers for rate, attitude, velocity, and position to calculate the setpoint (set from, for example, the RC) to the desired estimate.

To perform the Auto-Tuning (of controllers), some pre-requirements need to be fulfilled. For a quadcopter, the vehicle must be able to fly in a relatively stable enough manner in either Altitude or Manual/Stabilized mode. The stability of the flight is described in the guide and involves moving the vehicle on different maneuvers (rolling left and right, pitching up and down), ensuring that it stabilizes within a few oscillations. Depending on the behavior of the drone, the PID controller values might have to be manually tuned. In QGroundControl and the documentation, it is possible to read how the various controller parameters affect the stability of the vehicle.

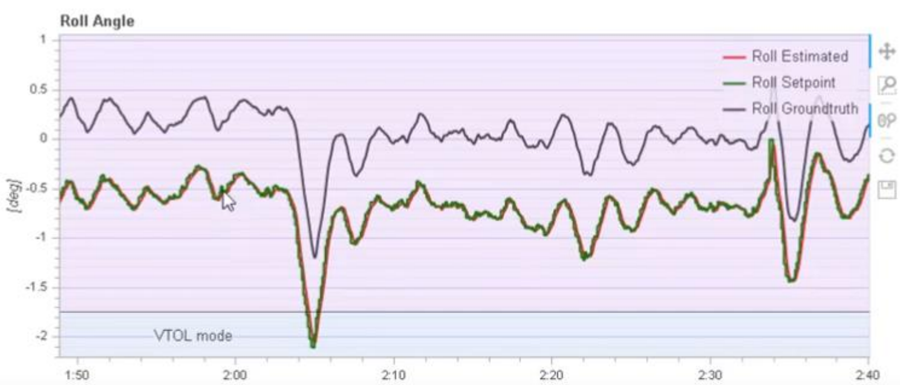


After the vehicle can hover relatively stable in the air, in one position, it is possible to use the "Autotune" procedure if the vehicle has contact via radio or WiFi telemetry or a USB cable with a laptop or other device running QGroundControl. The "Autotune" procedure will approximately take 70 s where the vehicle is automatically performing some maneuvers to adjust the controller values.

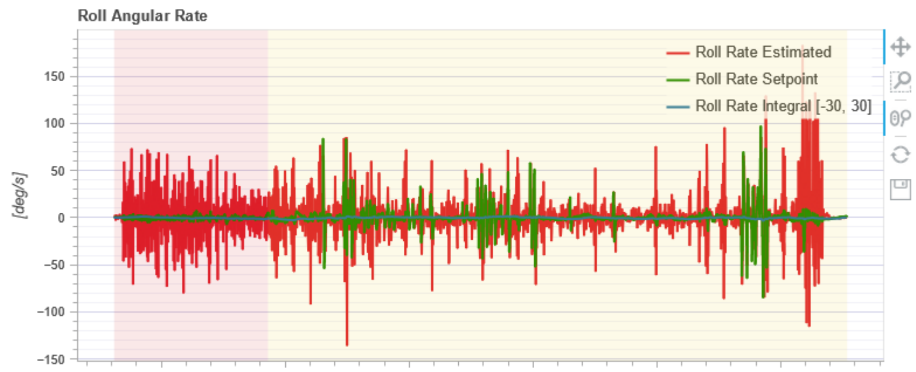
Reference: https://docs.px4.io/main/en/config/autotune.html and https://docs.px4.io/main/en/config_mc/pid_tuning_guide_multicopter_basic.html

Each time the vehicle is armed, i.e., the motors spin, a binary log file (.ulg) will be created and saved on the vehicle's file system. These log files can be analyzed in various online tools to troubleshoot, e.g., the roll setpoint in comparison to the estimate of the controllers.

An example of a well-tuned vehicle roll angle in a flight log:



An example of a poorly tuned vehicle roll angular rate in a flight log:



Reference: https://docs.px4.io/main/en/dev_log/logging.html

Parameters

In QGroundControl, you can see all the parameters and their current values, where all the parameter values marked in red color have been changed by the user.

It is possible to click on each parameter to read some description of it, and what values are recommended.

A full list of all parameters, with a description can be found here:
https://docs.px4.io/main/en/advanced_config/parameter_reference.html