Pixhawk

* Bill of material:

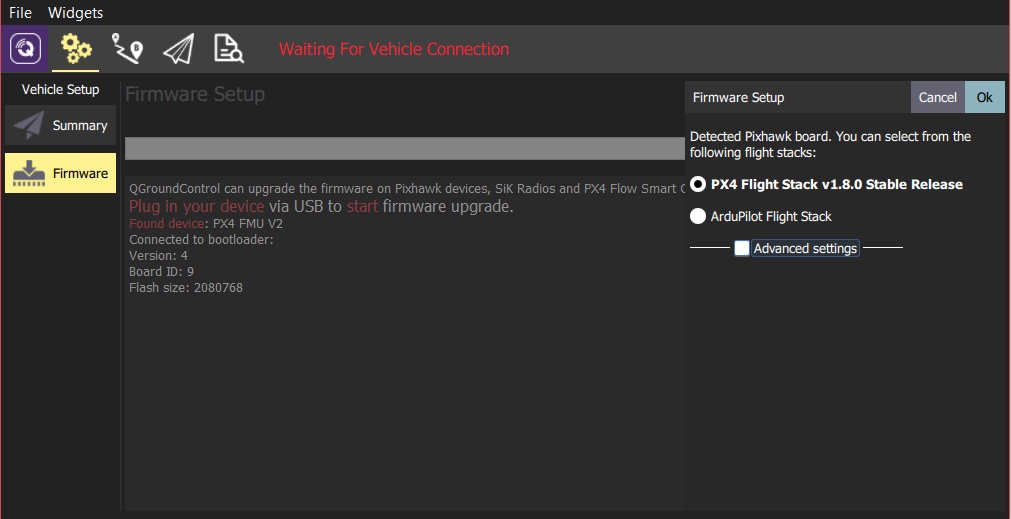
1. Pixhawk px4 2.4.8
2. UBlox GPS and compass
3. Power module of the Pixhawk
4. FrSky transmitter and receiver
5. Df 13 male to male wires

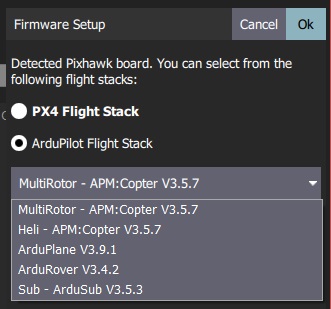
6 pins, 5 pins and 2 pins.

1. Setting up of Pixhawk.

* Initial setup

Connect the Pixhawk to the work station with the help of a usb. Start Qgroundcontrol. To set up Qgc go to firmware section and install the firmware. The firmware required for setting up the vehicle in this case is Ardurover (latest version).

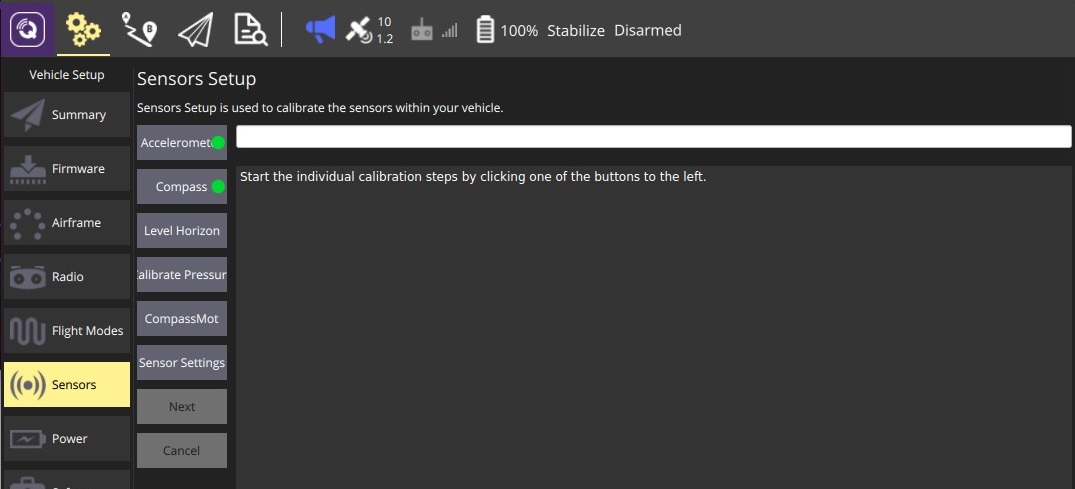




Once the firmware is installed, go to setup section and you will see red dots on some windows. This red dot will change to green by calibration of the sensors and radio. The SensorSetup section allows you to configure and calibrate the vehicle's compass, gyroscope, accelerometer and any other sensors (the available sensors will depend on the vehicle type).

Note: One must check whether he/she is using ardupilot or px4 because set of sensors are different for both. It is better to use ardupilot’s firmware.

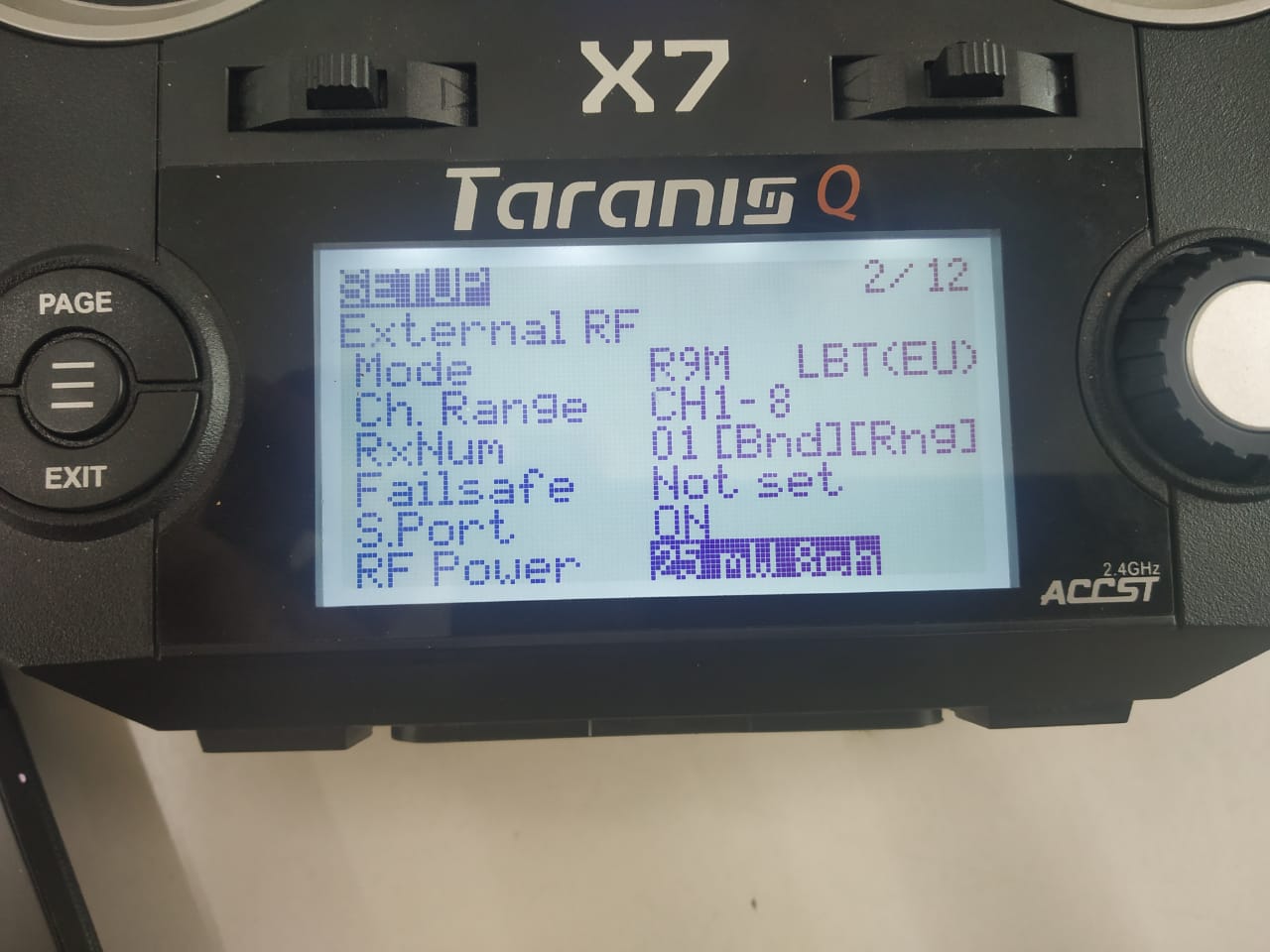
Sensors to be callibrated include accelerometer and compass. Once the sensors are setup vehicle is ready to set up rc telemetry followed by frame type and flight modes.



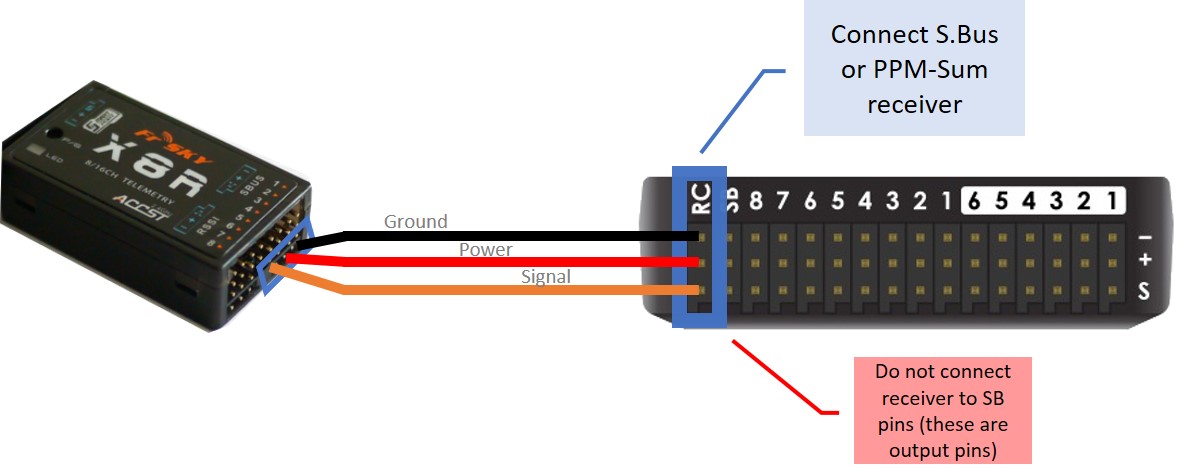
After this initial setup your vehicle is ready to get the telemetary connections. Now further steps are setting up RC transmitter and reciever, gps and vehicle frame. Now further setup of qgc will be done after setting up the radio transmitter.

1. Setting up of radio transmitter and reciever.

* Transmitter setup:

To setup the connection between transmitter and reciever create a model in Tx and setup the Rx as I was using R9 as a Rx so these are the following factors that we use to set 

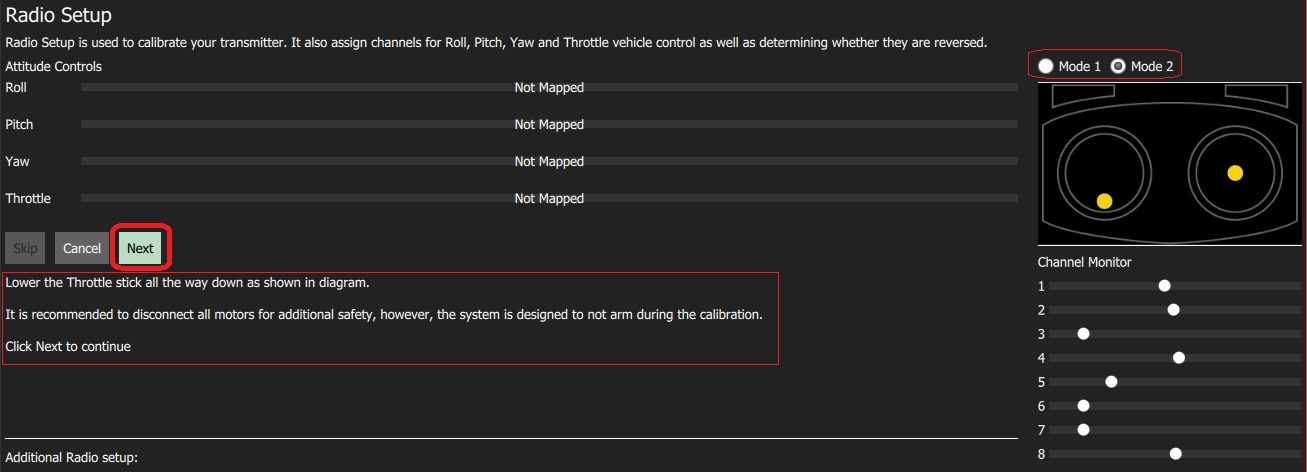
To control vehicle in manual mode we must use rc system. Pixhawk supports a very small set of RC transmitters and reciervers. You will need to select a transmitter/receiver pair that are compatible with each other. In addition, receivers have to be [compatible with PX4](https://docs.px4.io/en/getting_started/rc_transmitter_receiver.html#compatible_receivers) and the flight controller hardware. Compatible radio systems are often sold together. For example, [FrSky Taranis X9D and FrSky X8R](https://hobbyking.com/en_us/frsky-2-4ghz-accst-taranis-x9d-plus-and-x8r-combo-digital-telemetry-radio-system-mode-2.html?___store=en_us) are a popular combination. A typical setup of rc looks like this



Connect your Pixhawk to the receiver in the same way as shown in the picture above. Once the connection is established between the transmitter and receiver you can now calibrate the radio of your qgc.

To calibrate the radio:

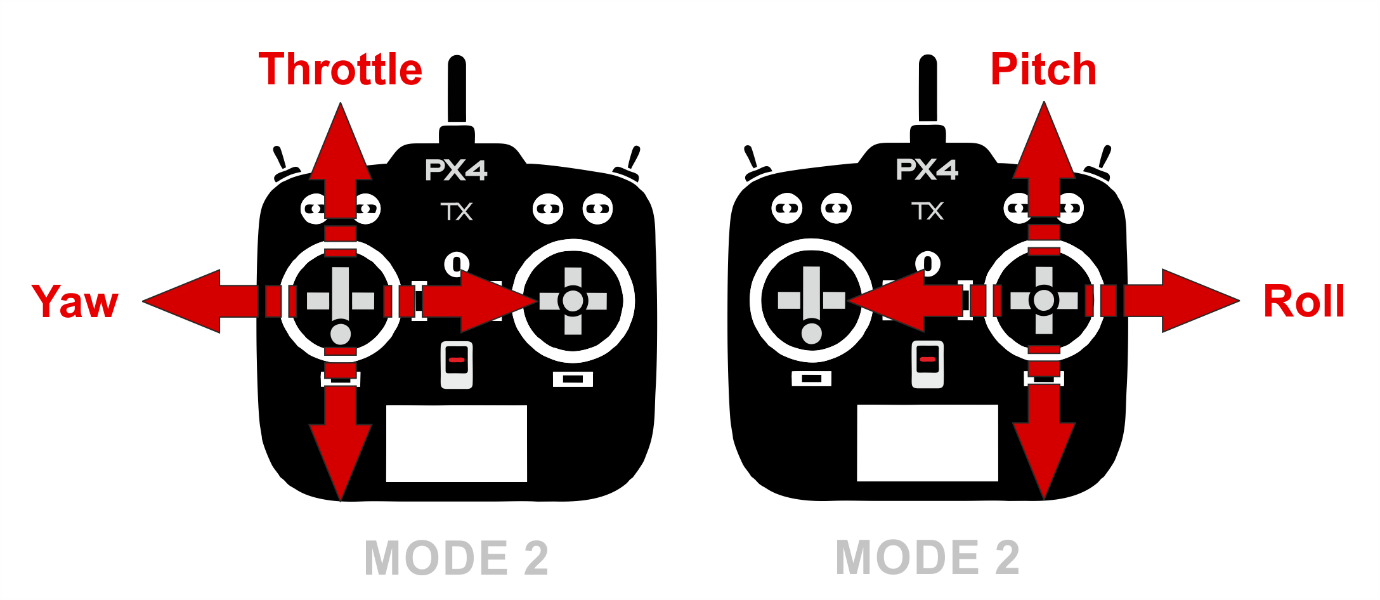
1. Select the **Gear** icon (Vehicle Setup) in the top toolbar and then **Radio** in the sidebar.
2. Turn on your RC transmitter.
3. Press **OK** to start the calibration.



1. Set the transmitter mode radio button that matches your transmitter configuration (this ensures that Qgroundcontrol displays the correct stick positions for you to follow during calibration).

Note: The choice of mode is largely one of taste (Mode 2 is more popular).

1. Move the sticks to the positions indicated in the text (and on the transmitter image). Press **Next** when the sticks are in position. Repeat for all positions.
2. When prompted, move all other switches and dials through their full range (you will be able to observe them moving on the *Channel Monitor*).
3. Press **Next** to save the settings.



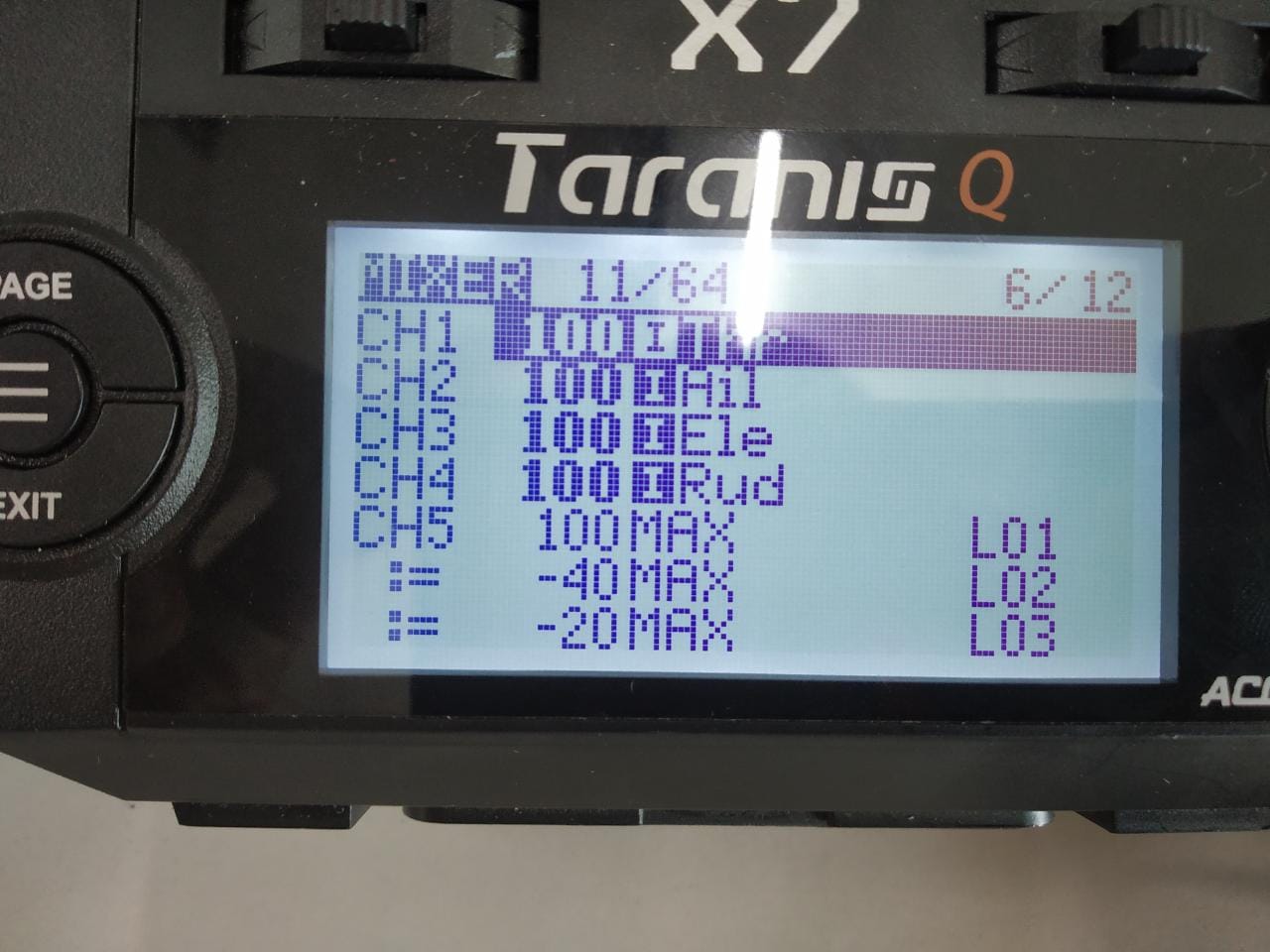
After completing the radio calibration in qgc go for setting up transmitter as follows:

1. Setting up of flight modes.

The Flight Modes section allows you to map flight modes to radio channel(s), and hence to the switches on your radio control transmitter. Both flight mode setup and the available flight modes are different in PX4 and ArduPilot (and there are some differences between ArduCopter and ArduPlane). Before we set up flight mode in qgc we must set it up in transmitter on any available channel. Usually first four channels are available for controlling of vehicle to later channels can be used for changing flight modes using switches and their different combinations. (NOTE:- use channel 5 for Flight mode change)

* Setting modes in transmitter

1. Create 6 logical switches using physical switches SA, SB, SC, SD and SE.
2. Now go to mixers and on any one channel establish switches on that channel.
3. One can create all the switches on one channel only putting different value of max weight.

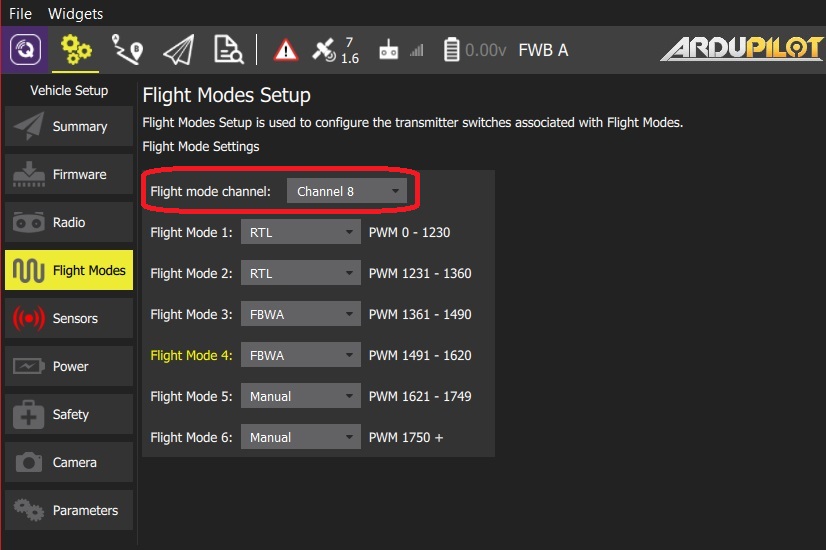




Once the switches are set up in transmitter, we can now calibrate flight modes on qgc.

* Calibrating the flight modes:

1. Turn on your RC transmitter.
2. Select the **Gear** icon (Vehicle Setup) in the top toolbar and then **Flight Modes** in the sidebar.



1. Select up to 6 flight modes in the drop downs.
2. Test that the modes are mapped to the right transmitter switches by selecting each mode switch on your transmitter in turn, and check that the desired flight mode is activated (the text turns yellow on *Qgroundcontrol* for the active mode).

* Different flight modes:

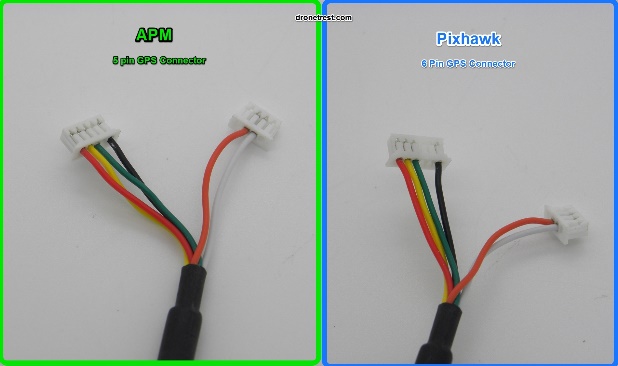
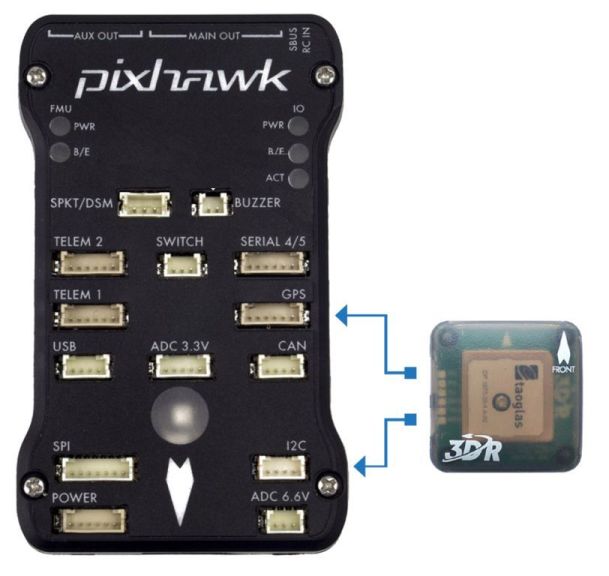
1. Manual: The control is done with the help of a RC controller.
2. Auto: If someone wants to put a mission in the Pixhawk.
3. RTL: RTL stands for return to launch. This mode will take the vehicle right to the starting point.
4. Loiter: Loiter mode allows boats to hold position in a strong current.
5. Steering: In Steering mode the user’s steering stick controls the vehicle’s lateral acceleration and the throttle stick controls the vehicle’s speed.

* Setting up of GPS:

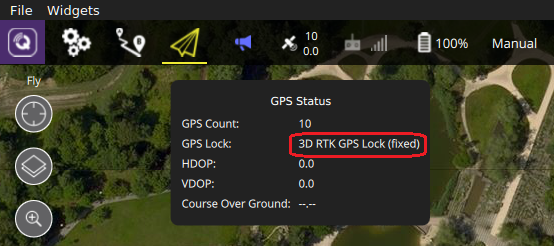
PX4 supports global navigation satellite systems (GNSS) (including GPS, GLONASS, Galileo, BeiDou, QZSS and SBAS) using receivers that communicate via the UBlox, MTK Ashtech or Emlid protocols, or via UAVCAN. It also supports [Real Time Kinematic (RTK) GPS Receivers](https://docs.px4.io/en/gps_compass/rtk_gps.html), which extend GPS systems to centimetre-level precision.



* Interface with Pixhawk:

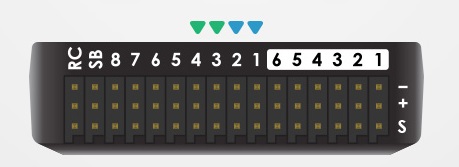
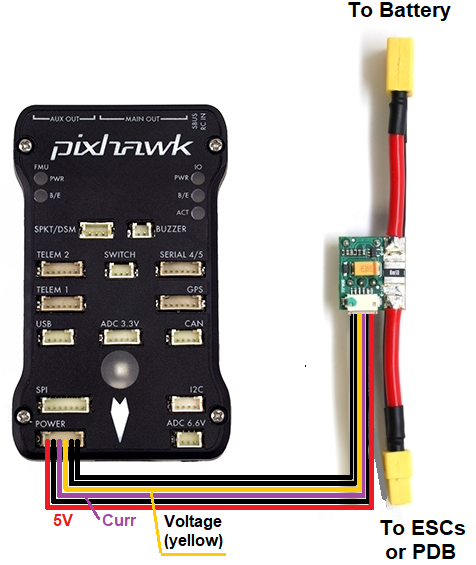
Start the vehicle and make sure it is connected to Qgroundcontrol. An RTK GPS status icon is displayed in the top icon bar while an RTK GPS device is connected (in addition to the normal GPS status icon).



Once the gps is setup we can now setup flight modes other than manual.

* Setting up a mission in Qgroundcontrol

1. Change to *Plan View*.
2. Add waypoints or commands to the mission and edit as needed.
3. Upload the mission to the vehicle.
4. Change to *Fly View* and fly the mission.
5. Powering of Pixhawk:



* Connection of esc’s with Pixhawk:

Connect the power (+), ground (-), and signal (s) wires for each ESC to the flight controller’s main output pins by motor number. Find your frame type below to determine the assigned order of the motors.

Now for motor calibration go to mission planner and then connect your vehicle.

Right after that go to mandatory hardware and to servo option and follow the picture.

Before that one might have to change a couple parameter as follows:

* [FRAME\_TYPE](http://ardupilot.org/rover/docs/parameters.html#frame-type) = 0
* FRAME\_CLASS= Boat
* [SERVO1\_FUNCTION](http://ardupilot.org/rover/docs/parameters.html#servo1-function) = ThrottleLeft
* [SERVO2\_FUNCTION](http://ardupilot.org/rover/docs/parameters.html#servo2-function) = Leave this to default
* [SERVO3\_FUNCTION](http://ardupilot.org/rover/docs/parameters.html#servo3-function) = ThrottleRight

Note: Boats can also be controlled with the standard Rover firmware. To specify that the vehicle is a boat the [FRAME\_CLASS](http://ardupilot.org/rover/docs/parameters.html#frame-class) parameter should be set to 2 (Boat).



* Precautions:

1. While calibration of sensors in Pixhawk make sure to align it properly.
2. While calibrating radio it is better to use mode 2.
3. GPS can stop working inside a room so better do the calibrate it near a window for faster results.

* Do’s

1. Repeat the calibration after the complete setup of vehicle
2. GPS should be mounted over antenna
3. GPS and Pixhawk arrow should be in same direction
4. Calibrate PIxhawk as per alignment

* Relay Switching

Create a logic in transmitter for ex SE↑ and mix a channel 7 with this logic with max value and open mission planner and edit parameter BRD\_PWM\_COUNT

The number of available Relays can be increased to a maximum of 6 by reducing the number of AUX pins used as [Servo](http://ardupilot.org/rover/docs/common-servo.html#common-servo) outputs. This can be accomplished by reducing the [BRD\_PWM\_COUNT](http://ardupilot.org/rover/docs/parameters.html#brd-pwm-count) from 4 to 2 or 0.

Then map parameter RELAY\_PIN as 54 (AUXOUT5)

Then map rc channel to relay pin output



Link for help :- <http://ardupilot.org/rover/docs/common-relay.html>

* Tuning for Auto Pilot
* [PIVOT\_TURN\_ANGLE](http://ardupilot.org/rover/docs/parameters.html#pivot-turn-angle) holds the minimum angle error that will trigger the pivot turn. For example when set to the default of “60”, a pivot turn will be triggered whenever the vehicle’s heading is at least 60 degrees off from the next waypoint. This parameter should normally not be set below 30 unless you wish to disable pivot-turns which can be done by setting this parameter to 0. The vehicle will begin towards the next waypoints once its heading is within 10 degrees of the heading to the destination.
* [ATC\_STR\_ANG\_P](http://ardupilot.org/rover/docs/parameters.html#atc-str-ang-p) converts the vehicle’s heading error into a desired turn rate. Higher values lead to a more aggressive turn towards the destination.
* [ATC\_STR\_RAT\_MAX](http://ardupilot.org/rover/docs/parameters.html#atc-str-rat-max) limits the vehicle’s maximum turn rate (in degrees per second) in all modes including during pivot turns.
* [ATC\_STR\_ACC\_MAX](http://ardupilot.org/rover/docs/parameters.html#atc-str-acc-max) limits the vehicle’s maximum rotational acceleration (in deg/sec/sec) in all modes. higher numbers lead to the vehicle getting up to its maximum turn rate more quickly

IN OUR CASE WE KEPT

PIVOT\_TRUN\_ANGLE - 35 degrees

ATC\_STR\_ANG\_P - 1.04 rad

ATC\_STR\_RAT\_MAX - 12 deg/sec

ATC\_STR\_ACC\_MAX – 2.4 deg/sec^2

* NAVL1\_PERIOD Period in seconds of L1 tracking loop. This parameter is the primary control for aggressiveness of turns in auto mode. This needs to be larger for less responsive airframes. The default of 20 is quite conservative, but for most RC aircraft will lead to reasonable flight. For smaller more agile aircraft a value closer to 15 is appropriate, or even as low as 10 for some very agile aircraft. When tuning, change this value in small increments, as a value that is much too small (say 5 or 10 below the right value) can lead to very radical turns, and a risk of stalling.
* NAVL1\_DAMPING Damping ratio for L1 control. Increase this in increments of 0.05 if you are getting overshoot in path tracking. You should not need a value below 0.7 or above 0.85.

IN OUR CASE WE KEPT

NAVL1\_PERIOD - 30 sec

NAVL1\_DAMPING – 0.785

WP\_RADIUS – 1.5 meter

WP\_OVERSHOOT – 1 meter

* Parameters changed in last test

NAVL1\_PERIOD:- 35 sec

NAVL1\_DAMPING:- 0.8

CRUISE\_SPEED:- 0.5m/s

CRUISE\_THROTTLE:- 100%

TURN RADIUS:- Radius at which the bot take 360 deg turn to find this we take the bot at a stable position and give full right or left throttle we get a approx. radius of around 0.6 meter

* NOTE:- SAVE THE CURRENT PARAMETERS SO THAT IF SOMETHING WENT WRONG YOU CAN DIRECTLY UPLOAD THESE PARAMETERS .