

## Raspberry PI Wfi HD FPV Quad Table of content

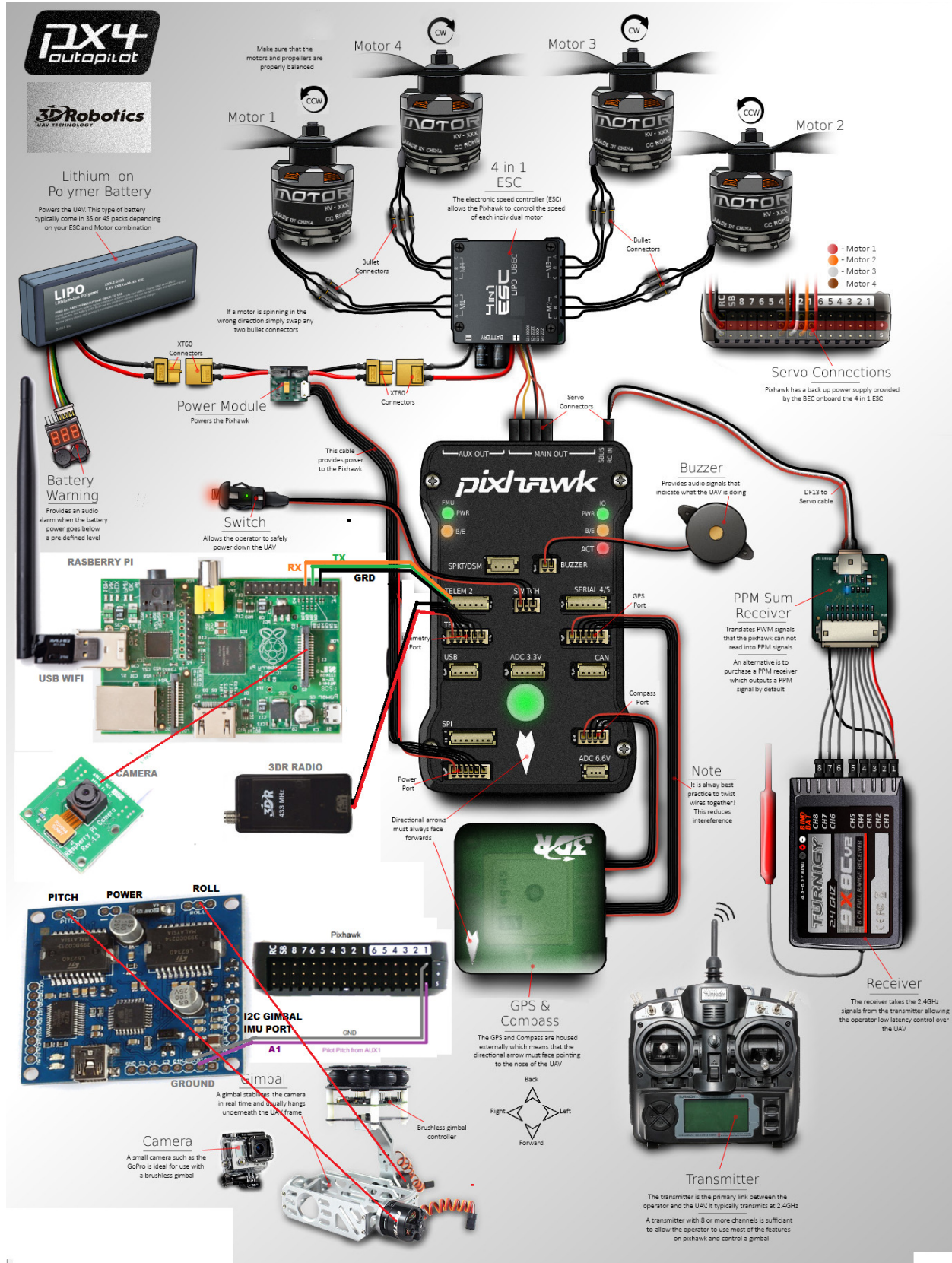
1. Over View:	Page No. 1-3	
2. Connecting RC receiver to the PIX Hawk:		Page No. 4-8
3. Installing APM firmware on PIXHAWK FC:		Page No. 8
4. Motor mounting & Direction:		Page No.
5. Calibrate compass:	Page No.	
6. Calibrate gyro:	Page No.	
7. Accelerometer calibration:	Page No.	
8. Calibrate power sensor:	Page No.	
9. Calibrate RC radio:	Page No.	

### Computer Setup

10. Calibrate 3DR radio:	Page No.	
11. Flying with a Joystick/Gamepad:		Page No.
12. Calibrate Gimbal:	Page No.	
13. Disable pre arm checks:	Page No.	
14. PID Tuning:	Page No.	
15. Raspberry PI setup:	Page No.	
16. Raspberry PI Net Work:	Page No.	
17. Connecting to copter HD Video:		Page No.
18. Mode of Operation set up:	Page No.	
19. Frame & RC TX Setup	Page No.	



# Connection Modification for PI :



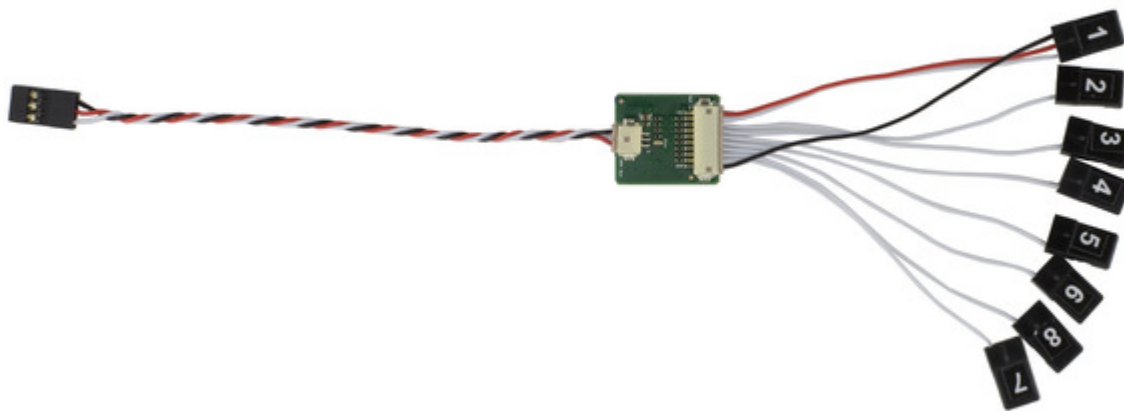
Quattro ESC 2S to 6S lipo battery motors max power is 5S battery will work with 3S to 5S lipo battery. Telemetry port 2 is used to connect to Raspberry PI serial port. Telemetry port 1 used for 3DR radio. Pi is powered by mini USB port. Assemble the parts before mounting on the frame to test, mount all components on the frame make sure all wires are covered no exposed wires carbon fiber is conductive will cause a short. Recommend 8x4 or 9x4 props. Upgrades can include larger motors and prop combinations copter for longer flight times, arms will need to be lengthened. Core components have no need to be upgraded motors power up to 6S lipo. Stable PH flight with no GPS connection requires optical flow & laser range finder optional components.

---

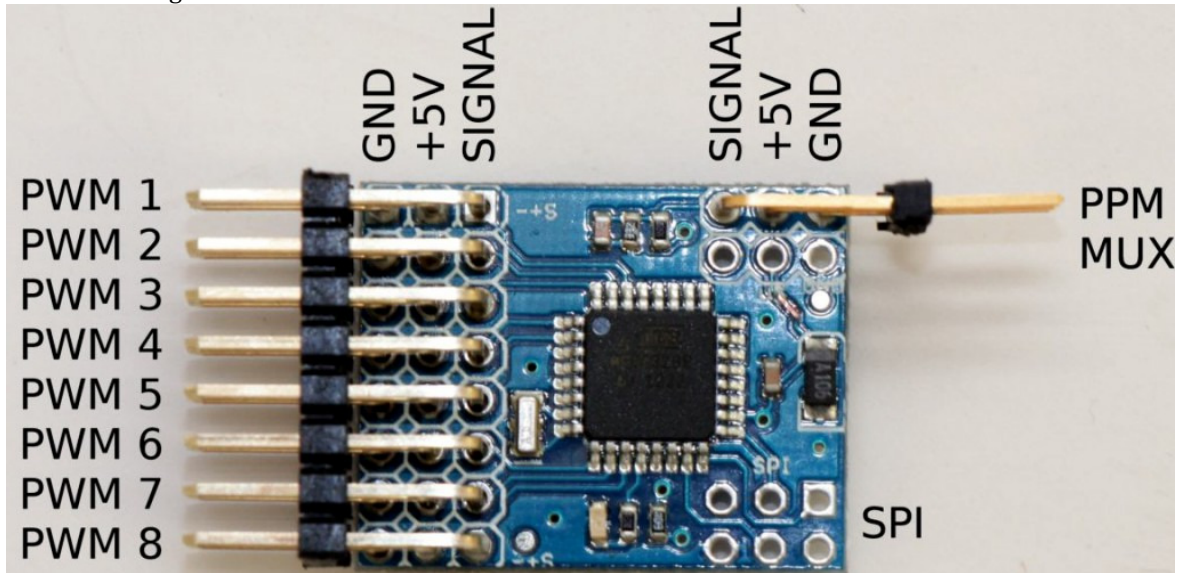
## Connecting RC receiver to the PIX Hawk:

### PX4FMU plus PX4IO PIX HAWK with 3DR 8 channel PPM encoder and 3DR telemetry radio

- The 3DR PPM Encoder will allow you to use any Conventional RC receiver but there are some downsides.
- The 3DR PPM Encoder does require quite a bit of additional wiring to the receiver.
- It uses quite a bit of power making it likely you will need to plug in your battery while doing radio setup with USB cable in Mission Planner.
- The 3DR encoder also costs as much or more than several of the available PPM-Sum receivers including the FRSky Delta 8 below.
- The Newest version of the **3DR PPM-Sum encoder** is here: ( **we are using S bus** )



- And the original 3DR PPM-Sum encoder is here:



**Using the 3DR 8 channel PPM Sum encoder in a system:**

The Original 3DR PPM-Sum encoder is shown, but the new encoder installs in the same fashion.



**Connect Pixhawk to computer**

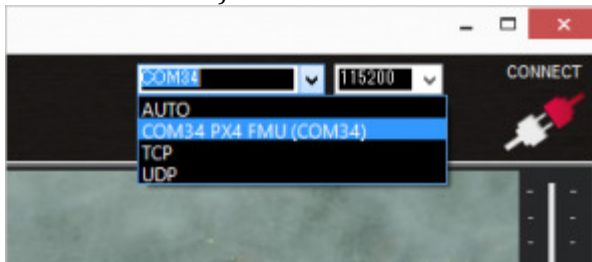
Once you've installed the Mission Planner onto your computer, connect the Pixhawk to your computer using the micro USB cable. Use a direct USB port on your computer, not a USB hub. *Click on images to enlarge.*



Windows should automatically detect and install the correct driver software. If you get a message stating “driver not found” or “Failed to install driver”, follow the instructions at the bottom of this page to reinstall the mission planner and drivers.

### Select the COM port

Open the Mission Planner and select the com port drop-down on the upper-right corner of the screen (near the Connect button). Select **AUTO** or **PX4 FMU**. Don't hit **Connect** just yet.



### Install firmware

On the Mission Planner's Initial Setup >> Install Firmware screen select the appropriate icon that matches your frame (i.e. Quad, Hexa). Answer “Yes” when it asks you “Are you sure?”.



After the Mission Planner detects which board you are using (i.e. Pixhawk) it will ask you to unplug the board, plug it back in and then press OK within a few seconds (during this brief period the bootloader accepts requests to upload new firmware).



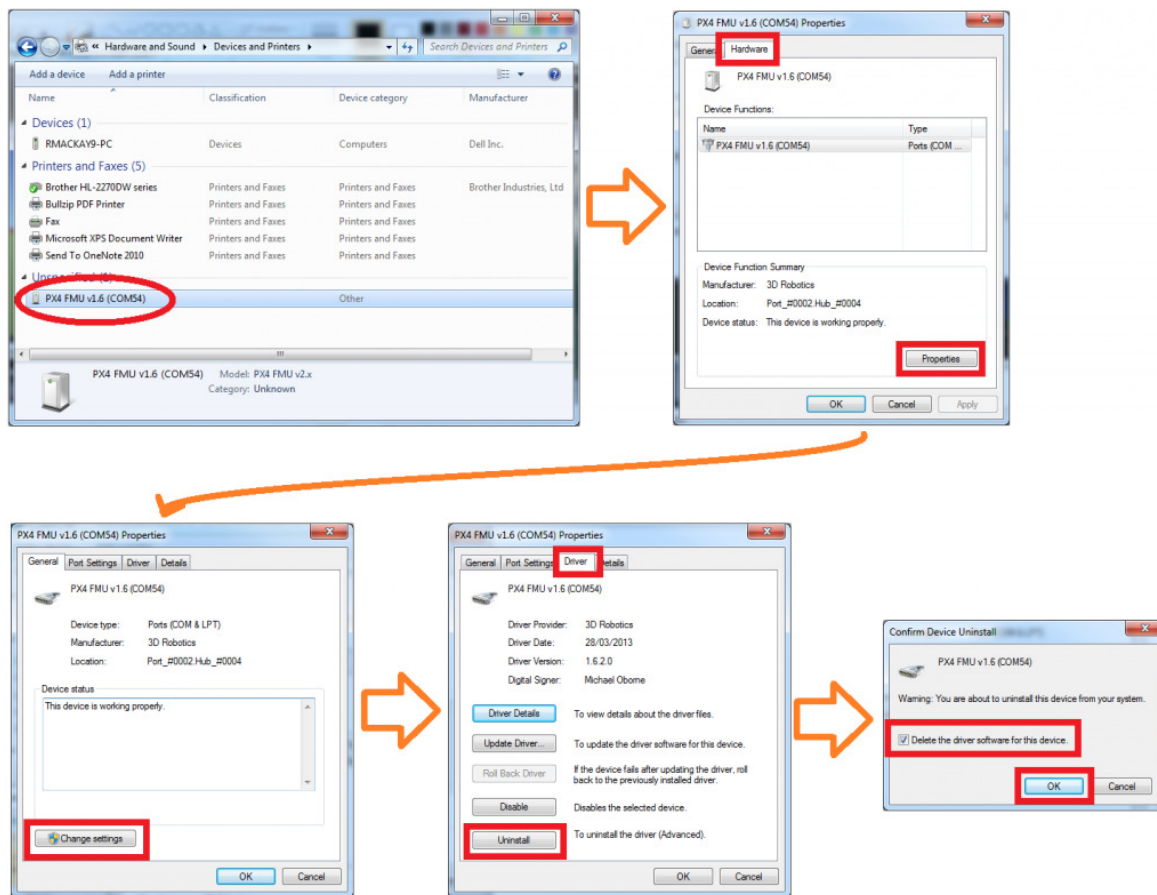
If all goes well you will see some status appear on the bottom right including the words, “erase...”, “program...”, “verify..” and “Upload Done”. The firmware has been successfully uploaded to the board.

You can test the firmware is basically working by switching to the Mission Planner’s Flight Data screen and pressing the “Connect” button. The HUD should update as you tilt the board.

### Re-Installing the Mission Planner and Pixhawk Drivers

If the driver failed to install correctly when the Pixhawk was first plugged in or you received the message “Error: No response from board” when uploading follow these instructions which may resolve the problem.

1. Backup the Mission Planner’s logs directory if there are dataflash or t logs from earlier flights you would like to keep. Normally the **logs** directory can be found in “C:\Program Files (x86)\APM Planner\logs” or “C:\Program Files (x86)\Mission Planner\logs”.
2. Uninstall the Mission Planner using the “Uninstall Mission Planner” application which can normally be found in your Start menu >> All Programs >> Mission Planner (or APM Planner).
3. Open a file explorer and delete the “C:\Program Files (x86)\Mission Planner” or “C:\Program Files (x86)\APM Planner” directories if they exist
4. Uninstall the Pixhawk driver:
  1. Open the Start >> All Programs >> Accessories >> System Tools >> Control Panel >> Devices and Printers
  2. while the Pixhawk is still plugged in right-mouse-button click on the PX4 FMU device and select Properties, the select the Hardware tab and select Properties
  3. On the General tab select Change Settings, on the Driver tab select Uninstall
  4. On the Confirm Device Uninstall window click the “Delete the driver software for this device” and press “OK”



5. Re-install the mission planner from the latest MSI. This should install the latest mission planner and also automatically reinstall the latest Pixhawk drivers onto your computer.

## Installing APM firmware on PIXHAWK FC:

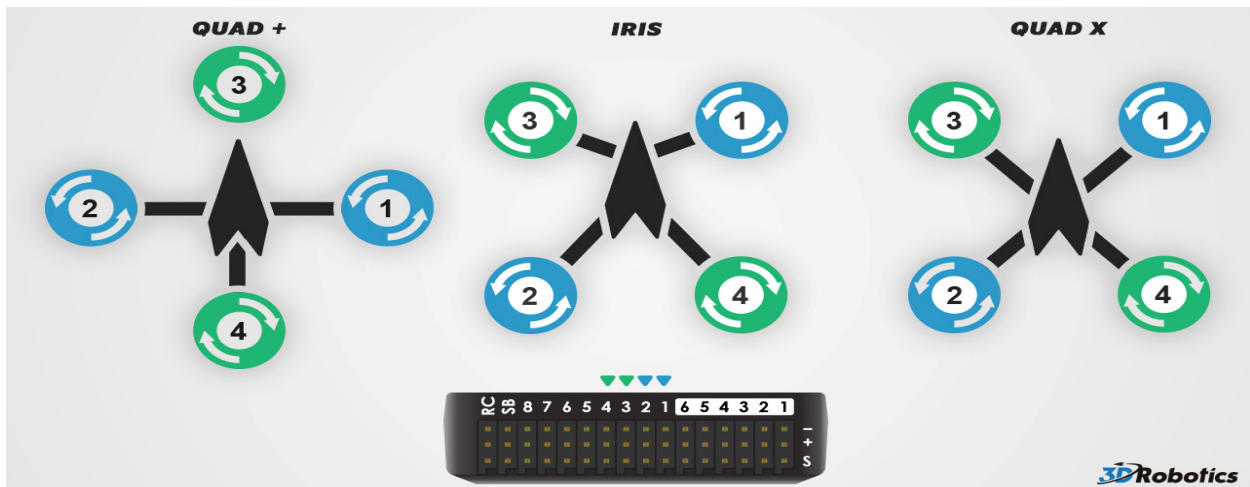
1. Install Mission Planner (Windows, Mac) or APM Planner (Windows, Mac, Linux) on your desktop or laptop computer which will be used to install Copter on your flight controller and to perform the required calibration and configuration.
2. Mounting the flight controller to the frame and Vibration Dampening
3. Flight Controller specific instructions for Connecting the APM2, Pixhawk and PX4 to the radio receiver, ESCs, etc
  - Powering the Pixhawk, PX4 or APM2
  - Mounting the GPS with Compass module
  - Connecting the radio receiver



- Connect the ESCs/Motors
4. Load Copter Firmware onto APM, Pixhawk or PX4: Load your autopilot with the latest copter firmware.
  5. Hardware Configuration: Use Mission Planner to configure radio control, calibrate the compass and accelerometer, and select your copter's orientation.
  6. ESC Calibration: ESCs must be calibrated before they can be used. This section guides you through calibrating ESCs automatically and manually.
  7. Motor Setup: This page shows you how to test the spin directions of the motors, set them correctly, and attach propellers.

Done all the above? Then you're ready for the First Flight.

## Motor mounting & Direction:



Motors are unidirectional in their design if motor is mounted the wrong way the prop will come loose. Motor are mounted R for turning right, this will self tighten the prop when turning R CW. Motor are mounted L for turning Left L this will self tighten the prop when turning L CCW. To change motor direction switches the 2 outer weir positions eg.

Motor connected to ESC.



Blue and black motor wires switched so motor spins in opposite direction



Motor direction also corresponds to prop type used with motor. There are 2 different types, pusher props usually labelled with a P or R belongs on CW motor or R motor. Normal props CCW or L belong on L motor.

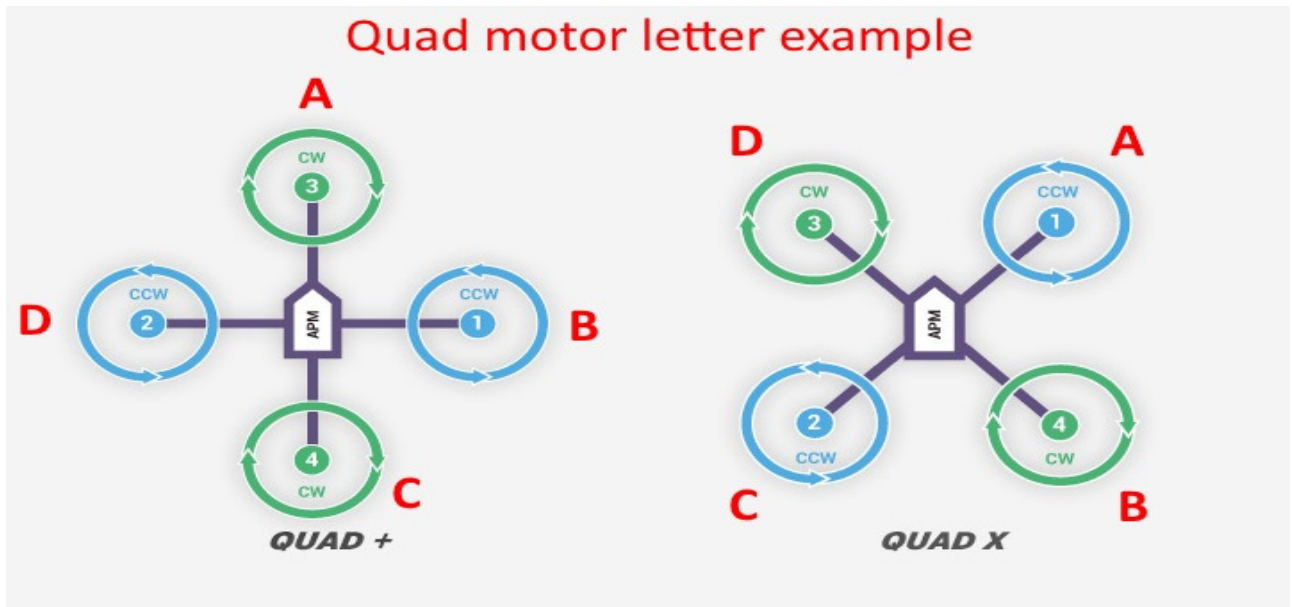
### Checking the motor numbering with the Mission Planner Motor test

An alternative way to check that the motors have been hooked up correctly is to use the “Motors” test in the Mission Planner Initial Setup menu. When connected to the vehicle via MAVLink, you can click on the green buttons shown below and the corresponding motor should spin for five seconds. Letters correspond to motor numbers as shown in the example below.

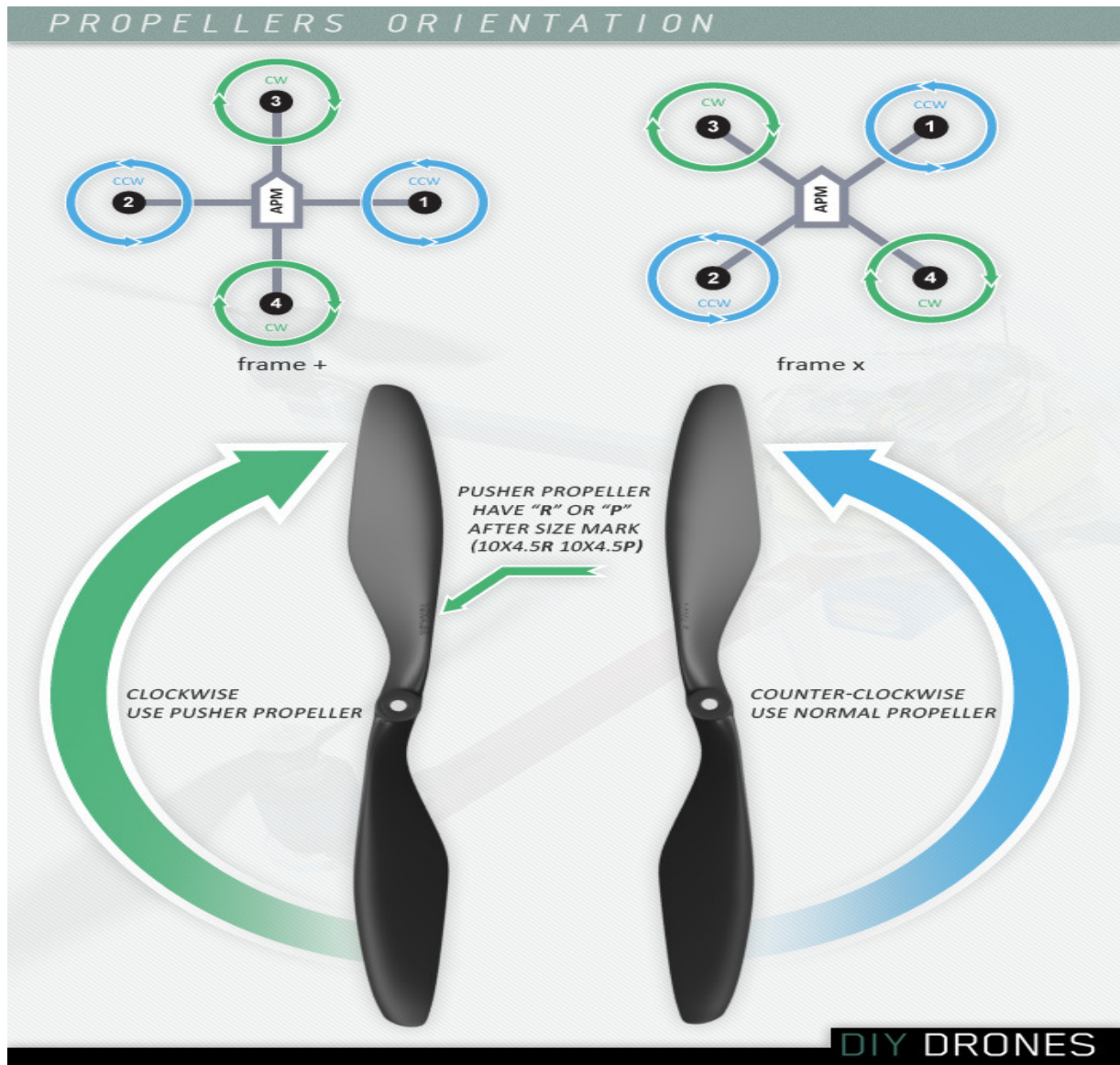
- Take off your props first!
- If no motors turn, raise the “Throttle %” to 10% and try again. If that doesn’t work, try 15%



The first motor to spin will be the one located directly forward in the case of + configuration, or the first motor to the right of straight forward in the case of X configuration. The motor test will then proceed in a clockwise rotation.



Quad is labelled accordingly :



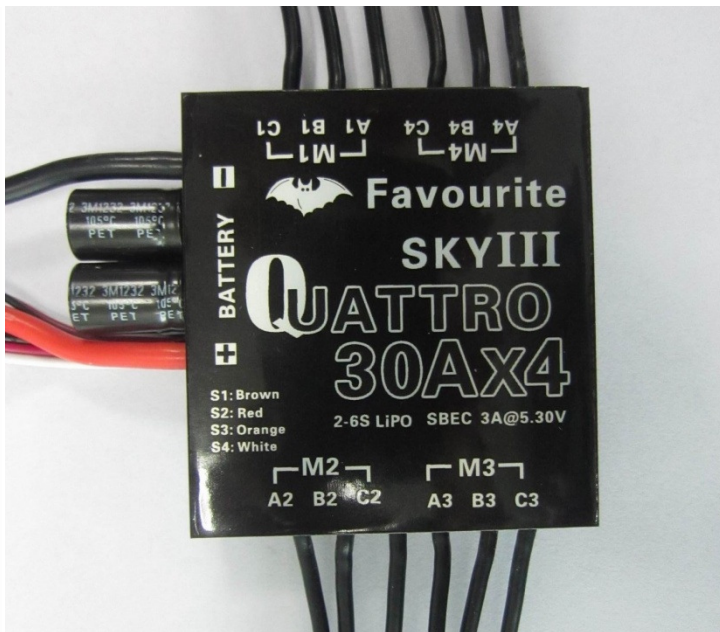
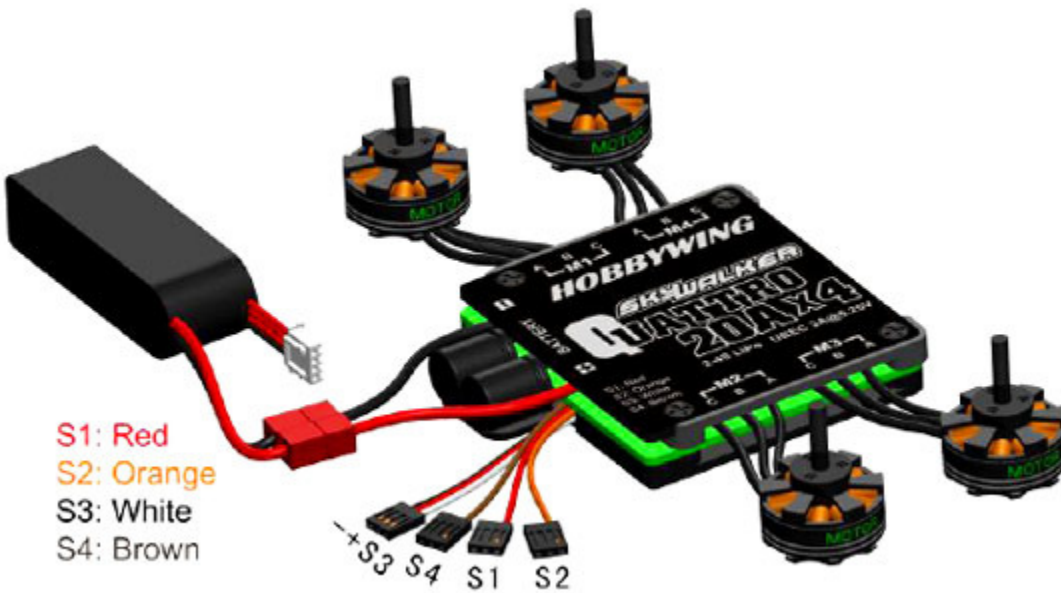
Frame lay out for a quad you have 2 choices plus or X. X mode is preferred because better visibility. Please do not mount the props at this point.

### Calibrate Quattro ESC:

**DISABLE SAFTY SWITCH SET TO (0 BRD\_SAFETYENABLE T) or you will have issues with ESC setup!  
Found in Mission planned advanced parameters list section.**

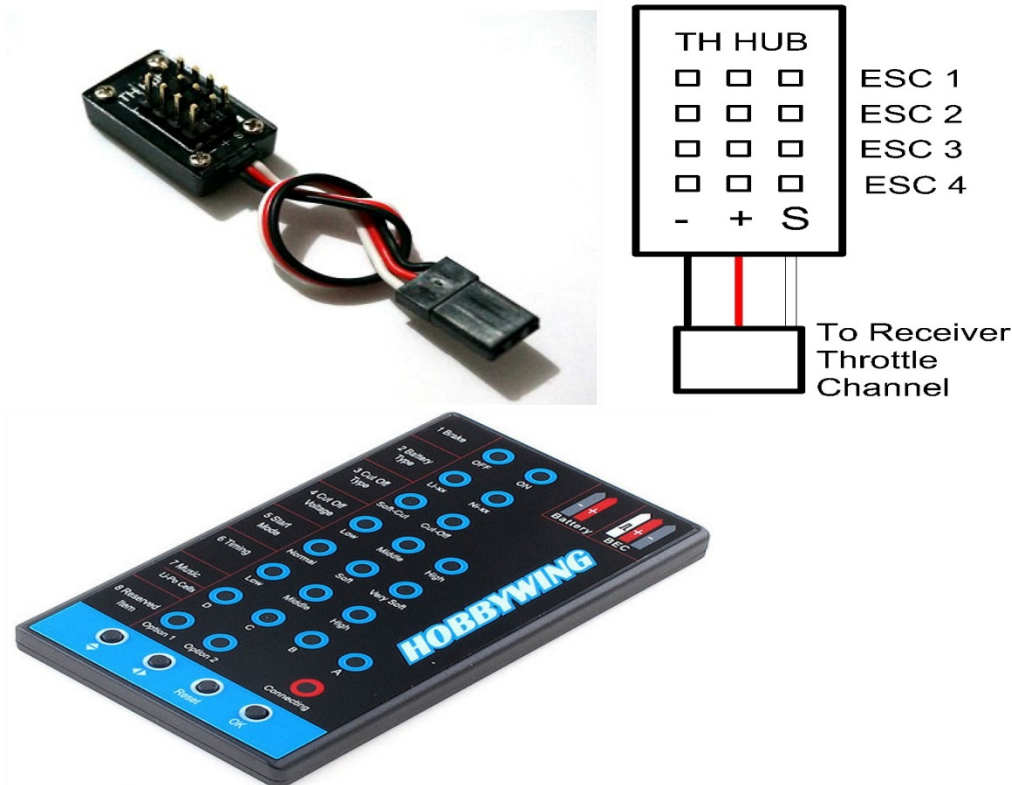
After the craft is assembled plug in the battery to Calibrate Quattro ESC. Power on the FC while your RC stick is full throttle you should hear fue tones then set the throttle to 0, Then set the throttle to full the motors should spin up this only need to be done once.

The Four In One ESC is a great option for Quadcopters, it has several advantages over individual ESCs.



- **Four in One "Quattro" ESCs are available in 20 and 25 amp 2 to 4 cell versions and a 30 amp 2 to 6 cell version.**
- The 4 in 1 ESC centralizes all your power wiring and eliminates the need for a power distribution board or block.
- The thick aluminum plate on the front of the ESC serves as a heat sink which is more than adequate for normal use.

- The design eliminates DC power wiring to individual ESCs reducing DC magnetic field and interference to magnetometer.
- The ESC comes with a good length of quality high flex motor wire which terminates in standard bullet connectors.
- For use in a quad copter with a controller which has a built in magnetometer the ESC should be mounted aluminum plate side up..
- The Four in one ESC is great for building new quadcopters or for retrofitting to old ones and it is easy to install.
- **Using the available ESC "Throttle Signal Hub" Below all 4 internal ESCs can be programmed and calibrated at one time.**
- On the site I link you to above, the "**Hub**" comes with the more expensive version of the 20 and 25 amp ESCs. **(Get The "Hub")**.
- I also (VERY) highly recommend the **Hobbywing ESC programmer** on the right below for much easier programming.



Calibration and Programming Below is for 20 and 25 AMP ESCs, [30 Amp](#)

For a PX4 or Pixhawk the receiver needs to be set to standard (not PPM-Sum) mode.

(Optional!) Four In One ESC Calibration using all at a time "Hub"

- **Optional! Calibrate all four ESCs at once using the "hub" and your RC transmitter and receiver.**
- Later it is necessary to calibrate the ESC while connected to the APM or Pixhawk, so this step probably has no benefit.
- At this time for an APM, PX4 or Pixhawk I recommend you skip this procedure and go to "Programming" the ESC below.
- If you are going to perform this calibration the receiver will need to be temporarily set to standard (not PPM-Sum) mode.
- The ESC should be installed in your copter and connected to the motors but with props off, motor direction does not matter now.
- Solder the ESC's main power and ground wires to your battery power input connector (Do not connect the battery at this time).
- Connect all 4 ESC signal cables to the "**Hub**" block in any order but be sure black wires are the - and white are on the S side.
- Connect the "**Hubs**" 3 wire cable to your RC receivers throttle channel ensuring signal and ground are on correct sides.
- Turn on your RC Transmitter and put the throttle stick all the way up to full throttle.
- **Connect the battery to the main power connector**, (This is the step to return to for a repeat attempt on any failure below).
- You should hear 3 short beeps followed within 6 seconds by another two beeps (If not disconnect the battery and try again).
- If above is OK move the throttle all the way down: Two more beeps within 3 seconds indicates calibration is done.
- If it did not produce the final two beeps when the throttle was lowered, repeat the procedure from the beginning.
- Once it has produced the final 2 beeps move the throttle up just a bit to see if all 4 motors start at the same time.
- If the motors start at significantly different times, repeat the procedure, otherwise calibration is complete.
- When you are satisfied with the calibration disconnect the main battery and turn off your RC transmitter.

### **Connect the Four In One ESC to the Quadcopter**

- **Mount the ESC with 2 strips of Velcro tape in your quadcopter with the Aluminum plate on top.**
- Note in the image below left the 4 in 1 ESC is tucked between the two upper frame plates on my FLIP Sport quadcopter.
- Connect the Motor leads to the four motors based on convenient wire connection (we will remap them as needed).

- Write down the motor (M1, M2, M3, M4) connected to each ESC signal (S1 - Red, S2 - Orange, S3 - White, S4 - Brown).
- Plug S1, S2, S3 and S4 ESC leads into the APM, PX4 or Pixhawks M1, M2, M3 and M4 outputs according to the data above.
- Make sure the ESC control signal wires are on the correct edge of your autopilot's servo bus, APM and Pixhawk are opposite.
- **If you are not using the ESC's BEC to power the Autopilot Servo Bus, disconnect the red power BEC wire from the ESC.**
- The ESC's BEC power lead is the center (red) wire of the S3 BEC Signal cable between black - ground and white - signal.
- If you need to disconnect BEC power remove the pin from the connector and tape it so you can reconnect it for programming.
- For the PX4 you normally use its built-in power supply and for the Pixhawk you normally use its supplied power module.
- However, both of these also permit isolating the servo bus power to allow separate power to be provided for external servos.
- If you are going to use additional external servos you should isolate the bus and leave the ESC's BEC red power wire connected.
- If you are using an APM without an additional power supply, leave the red power wire from the ESC's BEC connected.

### **Configure the Autopilot with the ESC**

- **Connect the APM, PX4 or Pixhawk Autopilot to Mission Planner with the USB cable.**
- Also ensure there are no propellers installed and connect the flight battery.
- Place the copter on a table in front of you with the rear of the copter nearest you.
- In Mission Planner, select the Terminal Mode tab and select test mode then select Test, Motors.
- The motors should each rotate in a clockwise sequence beginning with the right motor furthest from you.
- Based on APM notation right front is motor 1 and rotates first then motor 4 then motor 2 then motor 3 as in image above right.
- If the motor order is not correct move the ESC's signal cables on the autopilot servo bus until motor order is correct.
- It is important that each of the motors has a similar performance when it is powered up.
- You must ensure that the direction each motor turns is correct as indicated in the above right image.
- If a motor is not turning in the correct direction simply exchange any 2 motor / ESC wire connections for that motor.



- At this point you can also recalibrate the ESC through the APM and Mission Planner if you feel you need to do so.
  - Disconnect the Battery from the copter and the USB cable from the AutoPilot.
- 

## Calibrate compass:

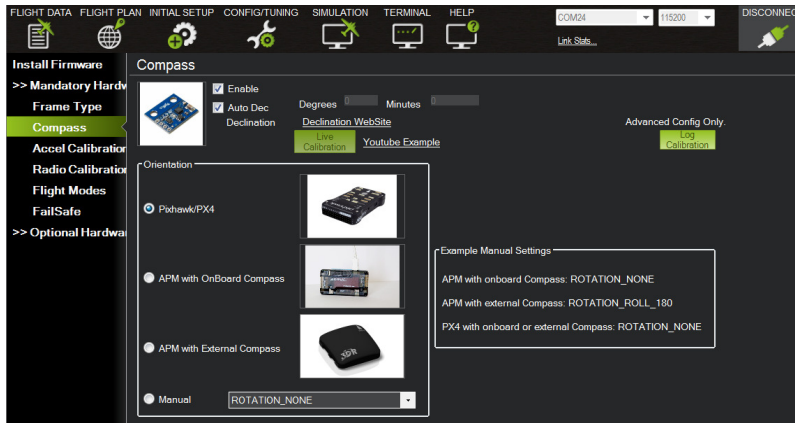
This stage of the calibration should be performed:

<https://youtu.be/UVMKn697ws4>

At good distance from large metal objects (including table frames) or strong magnetic fields (electrical lines)

UAV should be fully assembled and everything connected to get a correct offset

- On a non-moving surface click the Mag button, 1 beep will be heard and a message will appear saying “starting mag cal”.
- Without moving the FMU / IO wait for 1 beep to be heard and a message saying “rotate around X-axis” pick up the FMU / IO and roll it around the forward facing axis until 1 beep is heard.
- The message appears saying “rotate around y-axis”, pitch the FMU / IO around the sideways axis until 1 beep is heard.
- The message appears saying “rotate around z-axis”, now yaw the FMU / IO around the vertical axis until 3 beeps are heard and the message appears saying “mag calibration done”.
- The results are automatically written to the parameters file.
- If you are using an FMUv1 or the external 3DR mag it will tell you to hold still at first, and only after a beep the moving part begins. If you are using a Pix hawk (FMUv2), the moving calibration part begins right away.
- To perform the basic compass calibration you should:
- Under **Initial Setup > Mandatory Hardware** select **Compass**. Select the correct orientation for your setup.
- Ensure the **Enable** and **AutoDec** check boxes are checked
- Click the “Live Calibration” button
- Next select your autopilot configuration, for Pix hawk and PX4, select the option Pix hawk/PX4 with the image of the Pix hawk. For APM 2.6, select APM with External Compass with the image of the GPS with Compass module. These options will automatically enter the correct orientation for the board. Ensure that you have mounted the GPS with Compass with the arrow facing toward the front of the vehicle and in the same direction as the arrow on the autopilot.



- A window should pop-up informing you that you have 60 seconds in which to rotate the APM/PX4 around all axis, press “OK”
- A countdown window should appear to show that the mission planner is collecting compass data
- During these next 60 seconds you should hold the copter in the air and rotate it slowly so that each side (front, back, left, right, top and bottom) points down towards the earth for a few seconds in turn.

---

## Calibrate gyro done at boot by system now:

- On a level surface click the GYRO button and, if using a buzzer, 1 beep will be heard and a message will appear in the top right hand corner of the QGC screen saying “starting gyro cal”.
- Without moving the FMU / IO wait for 3 beeps and the message “gyro calibration done”.
- The results are automatically written to the parameters file.

---

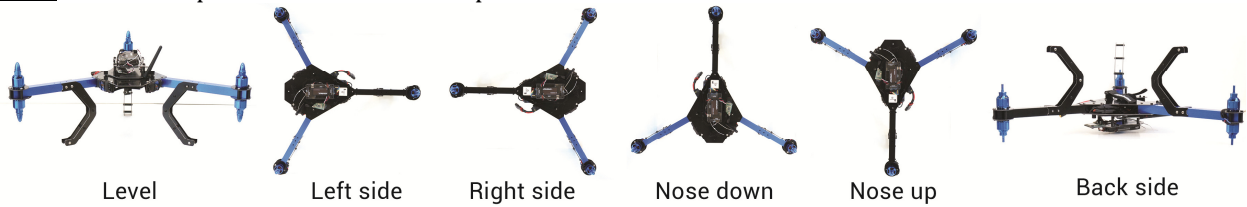
## Accelerometer calibration:

Accelerometer calibration procedure requires measurement for 6 vehicle orientations (level, on back, on left side, on right side, nose up, nose down). It's necessary to estimate accelerometer offset and scale for each axis.

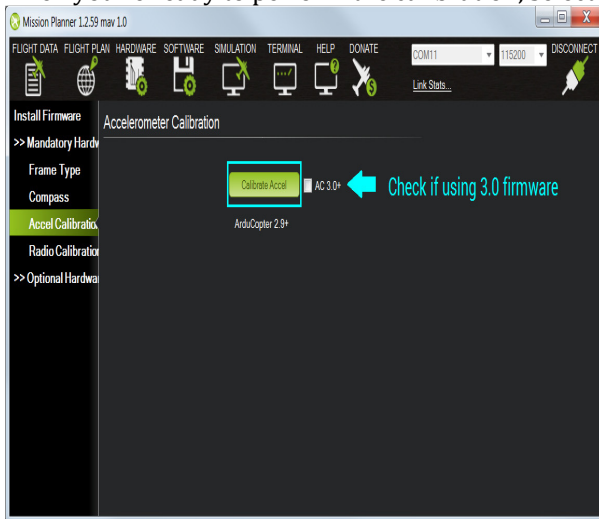
### Calibrate accelerometer

Under Initial Setup, select Accel Calibration from the left-side menu. If you are using Copter Code 3.0, make sure you check the AC 3.0+ box. If you are using 2.9.1b (or earlier), leave the box unchecked. This process will require you to place your copter in the positions shown below in sequence. The level position is the most important to get right as this will be the attitude that your controller considers level while flying.

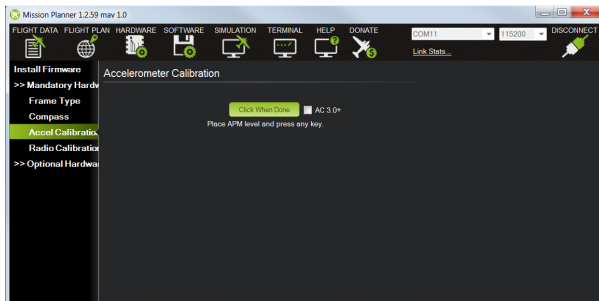
It is important not to move the copter immediately after pressing the key for each step. Check out [this video](#) for an example of how to hold the copter.



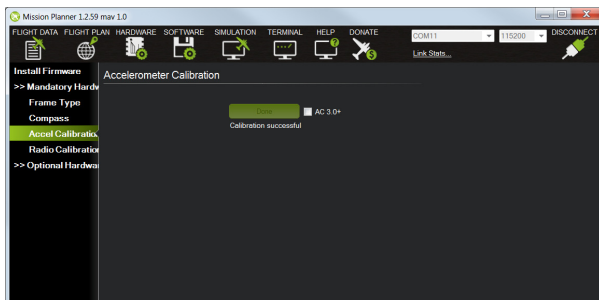
When you're ready to perform the calibration, select **Calibrate Accel**.



Mission Planner will prompt you to place the copter in each calibration position and press any key. Proceed through the required positions. (It is not necessary to use the Click When Done button.)



When you've completed the calibration process, Mission Planner will display "Calibration Successful!" as shown below.



- Find two surfaces: level and vertical (e.g. floor and wall is good choice).
- Click the Accel button, 1 beep will be heard and the message “accel calibration started” will be printed.
- Now place vehicle in one of 6 orientations and fix it. Measurement will be started after vehicle fixed for 3-4 seconds. It will not start while vehicle is moving. 1 beep indicates that measurement for this orientation complete.
- If vehicle is moved during 30s, calibration aborts with “timeout” message.
- Repeat measurements for each of 6 orientations. Message will tell which measurements are not collected yet, e.g. “keep vehicle still: y+ y-” means that “on right side” and “on left side” not yet measured.
- Triple beep will be heard when all measurements are completed.
- The results are automatically written to the parameters file.
- **MAGneto 4 Step Calibration Maneuver**
- UAV Still on a flat surface
- **Step1 : Press the “MAG ...” button on Qground ctrl-**
- 1Beep- (no beep on Pixhawk V2) message : “Starting Mag Cal” Keep UAV Still
- 1 Beep-( no beep on PX V1.2 and Pixhawk) message : “roll around X-axis”
- **-Step2 : Roll UAV around Forward facing axis.**
- 1 Beep-( no beep on PX V1.2 and Pixhawk) message : “Pitch around Y-axis”
- **-Step3 : Pitch UAV around sideways axis**
- 1 Beep-( no beep on PX V1.2 and Pixhawk) message : “rotate around Z axis”
- **-Step4 : YAw UAV around the verticale axis.**
- 3 Beep- message : “mag calibration done”

---

## Calibrate power sensor:

Pix hawk comes standard with three (redundant) ways to powered it up:

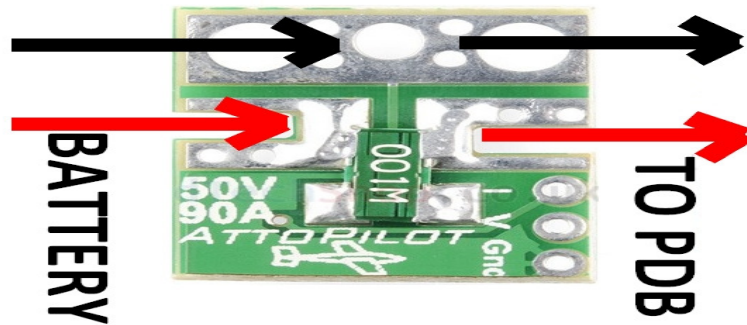
1-USB : not used to fly obviously; just useful on the ground for connection on a ground station software.

2-The power module port accepting a maximum input voltage of 5.7volts (and will not get destroyed up to 20 Volts input)

3-The RC input pins. Will accept a maximum voltage of 5.7volts also (and is also protected up to 20 Volts)

This guide assumes a use of Pix hawk's power module port which provides not only a way to power the board but also the pins to measure current and voltage values of the main battery.

This guide assumes a use of a 6S battery in combination with a Autopilot current & Voltage sensor board. This Autopilots "power module" replaces the 3DR 4S limited power module. The Autopilot board comes in three



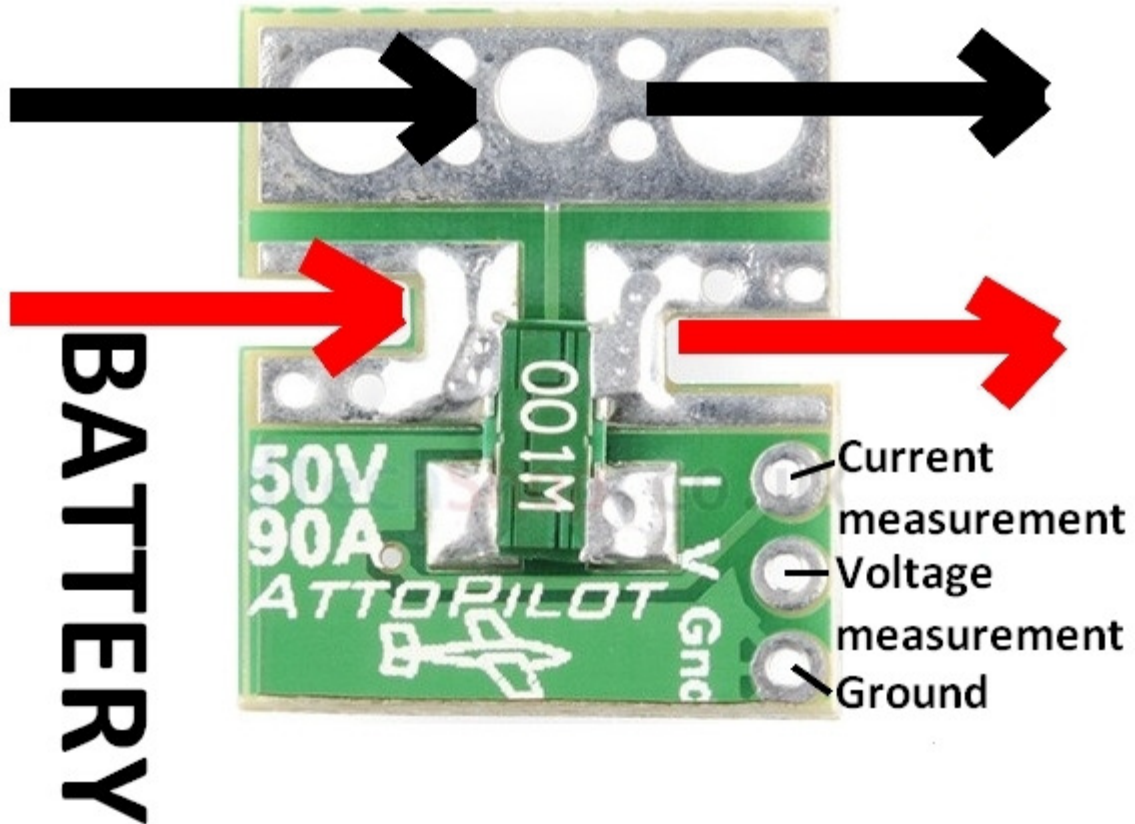
flavors:

45 amps, 90 amps or 180 amps.

The choice of the right Autopilot board (45A, 90A or 180 A) will depend on your motor/props combination: take the Autopilot version that has the smallest amps capacity above your max multi-copter current consumption. However we will introduce in this guide a way to use the 90 amps Autopilot board to measure up to 150 amps, still using Pix hawk's power module port.

**Autopilot description:**

An Autopilot board provides three wire soldering pads to solder: a current measurement wire, a voltage measurement wire and a ground wire. See picture below:



The Autopilot 90A support up to 50Volts for a maximum of 90A. However the resistor specifications exceed the 90A limitation which makes it possible to use it for measuring 150 amps (we will take this as a assumed max current as our example for the rest of the explanation).

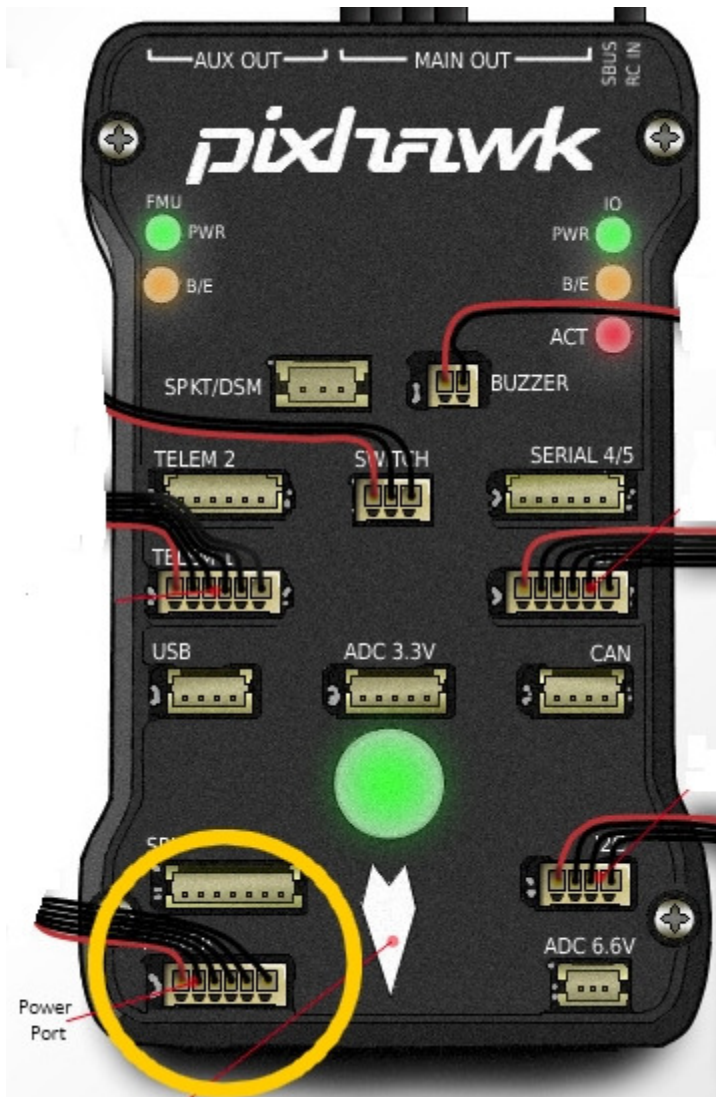
The datasheet of Autopilot specifies that the Voltage measurement wire outputs an analog voltage of 63,69 milli Volt per Volt. Similarly the current measurement wire outputs an analog voltage of 36,60 milli Volt per Volt.

So for a 6S battery the maximum analog voltage values will be:

-For voltage measurement: [min 0V – max 1.6V]

-For current measurement: [min 0V – max 3,3 V]

Pix hawk power port description (pinout):



Reusing the excellent pixhawk infographics published in the wiki, the image shows circled in yellow where the power port is on the pixhawk board.

The power port is a so-called DF13 connector with 6 pins.

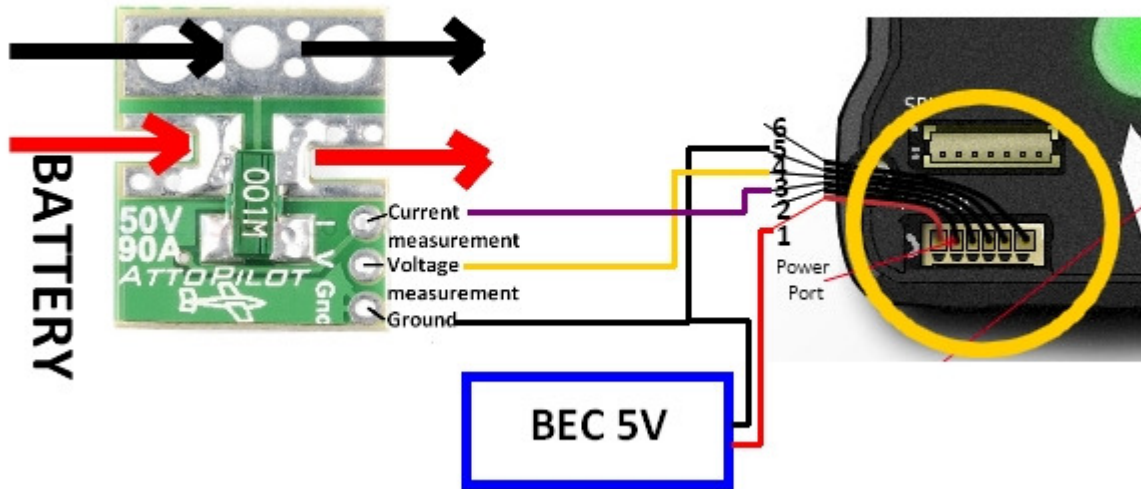
The six pins of this connector are assigned in the following order, starting by the red wire on the leftmost pin:

**Power Port Pinout Description:**

- 1- Vcc (5V input)
- 2- Vcc (5V input)
- 3- I (Battery current measurement analog voltage input)
- 4- V (Battery voltage measurement analog voltage input)
- 5- Ground
- 6- Ground

**Wiring Case 1 : to measure up to a maximum of 90 amps**

The connections between Autopilots and Pix hawk are shown in the illustration below:



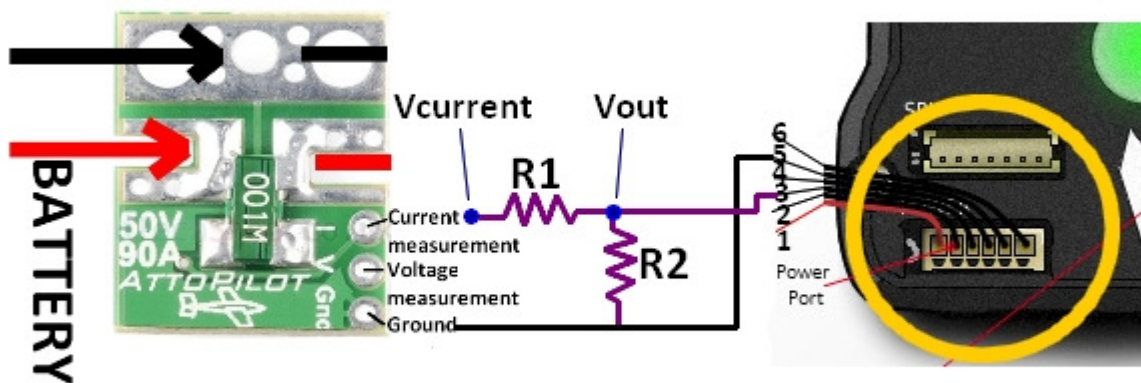
We have added an optional BEC in the illustration that would be connected to the Vcc and Ground wires of the power module. It is optional as Pix hawk could alternatively be powered via the RC inputs.

**Wiring Case 2 : to measure up to a maximum of 150 amps**

The connections between Autopilot and Pix hawk will integrate resistors to be able to measure up to 150 amps.

Indeed the ADC of this power port on Pix hawk has a range of 0-3.3V. This means that for the maximum true current of 150 amps, we want the current analog wire of Autopilots to output maximum 3.3Volts (as it is the case in case1 for 90 amps max without additional resistors).

So we will build a small resistors divider on wire 3 (current measurement) & wire 5 or 6 (Ground) as follows:





The equation that gives the value of  $V_{out}$  in function of the resistors  $R1$  and  $R2$  is:

$$V_{out} = \frac{R2}{R1 + R2} * V_{current}$$

What we want is to get  $V_{out} = 3,3$  when Current =150 amps or a Volt of :

$$3.3 * \frac{150}{90} = 5,5V$$

When current is 90 amps versus 150 amps, it requires a reduction in voltage of :

$$Voltage\ reduction = \frac{3.3}{5.5} = 0.6$$

So using the above equation :

$$\frac{R2}{R1 + R2} = 0.6$$

For  $R1=10kOhms$ ,

$$R2 = \frac{0.6}{0.4} * 10 = 15kOhms$$

So, for  $R1=10kOhms$ ,  $R2 =15kOhms$ .

You can choose other values of  $R1$  and calculate the resulting  $R2$  resistor value. You can choose also another max amp output (but I would not advise higher than 150 amps with the 90 amps Autopilot, otherwise use the 180 amps version instead) and calculate the resulting  $R1$ ,  $R2$  resistors.

As a result, when the current is 150amps, Volt will have value of 3.3Volts.

**Mission planner / parameters configuration in battery monitor screen:**

In the battery monitor parameters screen, you can manually select which current and voltage sensor you are using. In the present case, you will select the power module and modify the following parameters to make the mission planner voltage and current display match the real values (measured using a wattmeter for example).

The explanation below is an extract from the Copter parameters list.

**Battery monitoring (BATT\_MONITOR)**

Controls enabling monitoring of the battery's voltage and current

Value	Meaning
0	Disabled

3	Voltage Only
4	Voltage and Current

**Battery Voltage sensing pin (BATT\_VOLT\_PIN)**

Setting this to 0 ~ 13 will enable battery current sensing on pins A0 ~ A13. For the 3DR power brick on APM2.5 it should be set to 13. On the PX4 it should be set to 100. **On the Pixhawk powered from the PM connector it should be set to 2.**

Value	Meaning
-1	Disabled
0	A0
1	A1
2	Pixhawk
13	A13
100	PX4

**Battery Current sensing pin (BATT\_CURR\_PIN)**

Setting this to 0 ~ 13 will enable battery current sensing on pins A0 ~ A13. For the 3DR power brick on APM2.5 it should be set to 12. On the PX4 it should be set to 101. **On the Pixhawk powered from the PM connector it should be set to 3.**

Value	Meaning
-1	Disabled
1	A1
2	A2
3	Pixhawk

12	A12
101	PX4

### Voltage Multiplier (BATT\_VOLT\_MULT)

Used to convert the voltage of the voltage sensing pin (BATT\_VOLT\_PIN) to the actual battery's voltage ( $\text{pin\_voltage} * \text{VOLT\_MULT}$ ). For the 3DR Power brick on APM2 or Pixhawk, this should be set to 10.1. For the Pixhawk with the 3DR 4in1 ESC this should be 12.02. For the PX4 using the PX4IO power supply this should be set to 1.

***This is a parameter to adjust to match the real Voltage value with the displayed mission planner value.***

### Amps per volt (BATT\_AMP\_PERVOLT)

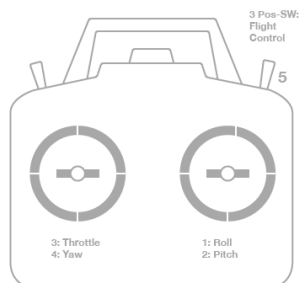
Number of amps that a 1V reading on the current sensor corresponds to. On the APM2 or Pixhawk using the 3DR Power brick this should be set to 17. For the Pixhawk with the 3DR 4in1 ESC this should be 17. Units: A/V.

***This is a parameter to adjust to match the real Voltage value with the displayed mission planner value.***

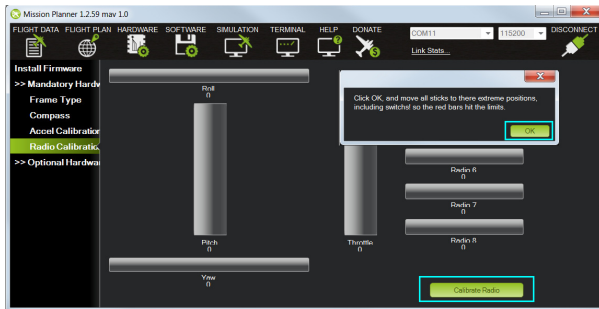
## Calibrate RC radio:

### Calibrate radio control

Turn on transmitter. Verify that the transmitter is in Airplane mode (APM needs airplane mode regardless of the platform type being piloted) and all trims are centered.



For Mode 1 transmitters, the left stick will control pitch and yaw, the right stick will control throttle and roll. For Mode 2 transmitters, the left stick will control throttle and yaw; the right stick will control pitch and roll. For either type of transmitter, the transmitter's three-position switch should be attached to channel 5 and will control flight modes. Optionally the transmitter's tuning knob should control channel 6 for in flight tuning. Channel 7 and Channel 8 switches can be used for controlling auxiliary functions. In Mission Planner, click on the green "Calibrate Radio" button in the lower right of the window. Mission Planner will call a dialog window to ensure radio control equipment is on, battery is not connected, and propellers are not attached. Select OK.



Select OK; move the control sticks and toggle switches on your transmitter to their limits of travel and observe the results on the radio calibration bars. Red lines will appear across the calibration bars to indicate maximum and minimum values. Move the Ch 5 and 6 toggle switches through their range of positions (Ch 7 and 8 are not used for basic operations).

Your transmitter should cause the following control changes:

Channel 1: low = roll left, high = roll right.

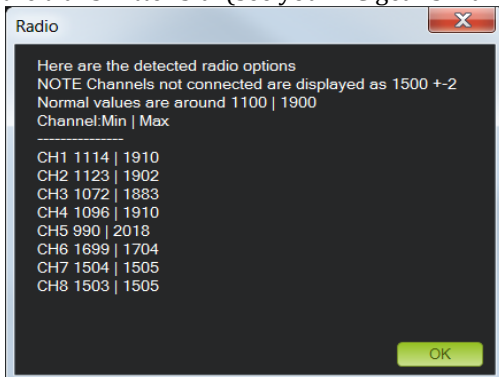
Channel 2: low = pitch forward, high=pitch back.

Channel 3: low = throttle down (off), high = throttle up.

Channel 4: low = yaw left, high = yaw right.



When the red bars for roll, pitch, throttle, yaw, and radio 5 (and optionally radio 6, 7 and 8) are set at the minimum and maximum positions, select **Click when Done**. Mission Planner will show a summary of the calibration data. Normal values are around 1100 for minimums and 1900 for maximums. **If the bars go in the opposite direction of the direction you're moving the sticks, that means that the channel is reversed on your RC transmitter side.** Use your RC transmitter's channel-reverse function to reverse it at the transmitter side. (see your RC gear's manual for that, if you're not familiar with the process)



---

## Calibrate 3DR radio:

The [3DR Radio](#) is the easiest way to setup a telemetry connection between your APM/Pixhawk and a ground station. It is small, light and inexpensive, and typically allows ranges of better than 300m “out of the box” (the range can be extended to several kilometres with the use of a patch antenna on the ground). The radio uses [open source firmware](#) which has been specially designed to work well with MAVLink packets and to be integrated with the Mission Planner, Copter, Rover and Plane.

The radios can be either 915Mhz or 433Mhz and you should purchase the model which is [appropriate for your country/region](#).

The main features of the 3DR Radio are listed below (see the [Advanced Configuration](#) for more information):

- Very small size
  - Light weight (under 4 grams without antenna)
  - Available in 900MHz or 433MHz (v2 only) variants
  - Receiver sensitivity to -121 dBm
  - Transmit power up to 20dBm (100mW)
  - Transparent serial link
  - Air data rates up to 250kbps
  - MAVLink protocol framing and status reporting
  - Frequency hopping spread spectrum (FHSS)
  - Adaptive time division multiplexing (TDM)
  - Support for LBT and AFA
  - Configurable duty cycle
  - Built-in error correcting code (can correct up to 25% data bit errors)
  - Demonstrated range of several kilometres with a small omni antenna
  - Can be used with a bi-directional amplifier for even more range
  - Open source firmware
  - AT commands for radio configuration
  - RT commands for remote radio configuration
  - Adaptive flow control when used with APM
  - Based on HM-TRP radio modules, with Si1000 8051 micro-controller and Si4432 radio module
- Status LEDs

The 3DR Radios have 2 status LEDs, one red and one green. The meaning of the different LED states is:

- Green LED blinking – searching for another radio
- Green LED solid – link is established with another radio
- Red LED flashing – transmitting data

- Red LED solid – in firmware update mode

**Connecting the radios**

The 3DR Radio v2 has interchangeable air and ground modules, meaning that you use them as a pair but it does not matter which one goes on the vehicle and which remains on the ground.

The radio has a micro-USB port, and a DF13 six-position port. The following sections explain how to connect these to flight controllers and ground stations.



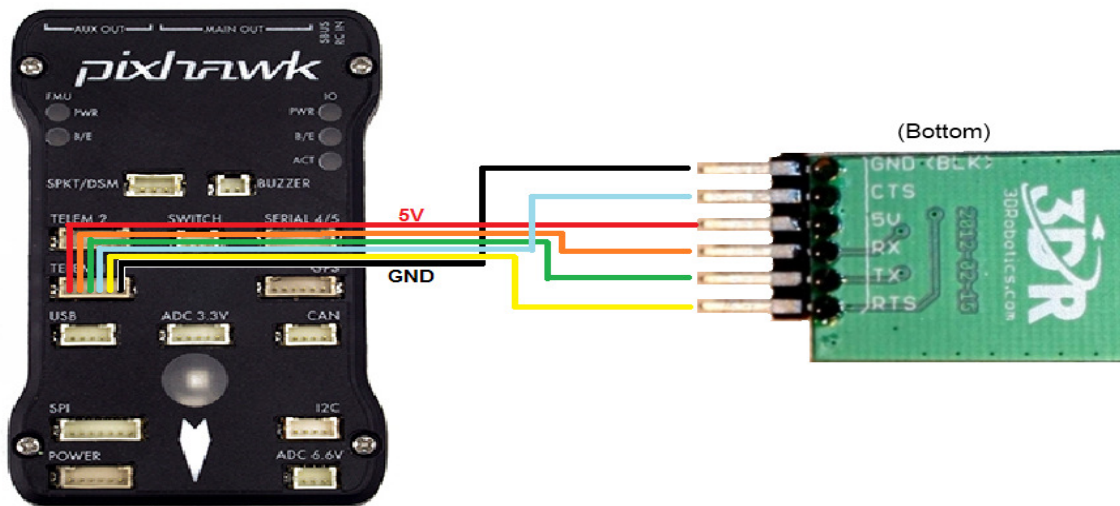
**3DR Radio V2**  
Pin-out Description

- Ground 6
- RTS\* (output) 5
- CTS\*\* (input) 4
- Autopilot receiver (radio transmitter) 3
- Autopilot transmitter (radio receiver) 2
- Power (+5 V) 1



\*RTS (request to send)  
\*\*CTS (clear to send)

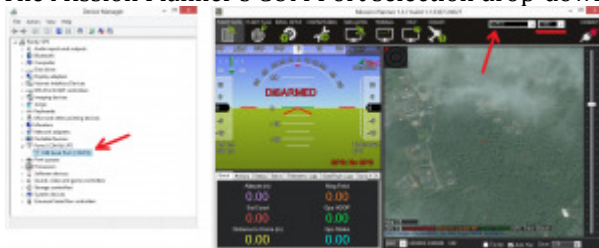
Connecting to Pix hawk Use the 6 pin DF13 connector that should have come with the radio to connect the 3DR radio to your Pix hawk's "Telem 1" ("Telem 2" or "Serial 4/5" can also be used but the default recommendation is "Telem1").





### Connecting to a PC

Connecting the 3DR radio to your Windows PC is as simple as connecting the micro USB cable (which should have been included with the radio) to your PC. The necessary drivers should be installed automatically and the radio will appear as a new “USB Serial Port” in the Windows Device Manager under Ports (COM & LPT). The Mission Planner’s COM Port selection drop-down should also contain the same new COM port.



### To connect the radios:

- Select the new COM port, set the baud rate drop down (which appears between the COM port and Connect buttons) to 57600.
- Press the **Connect** button and if the two radios connect successfully you should be able to lean your vehicle left and right and see it’s attitude update on the MP’s Flight Data screen’s artificial horizon.

### Connecting to an Android tablet

Connecting the 3DR radio to an Android tablet using the L shaped micro USB cable (which should have been included with the radio). Then follow the directions in your preferred Ground Station app.



## Configuring using the Mission Planner

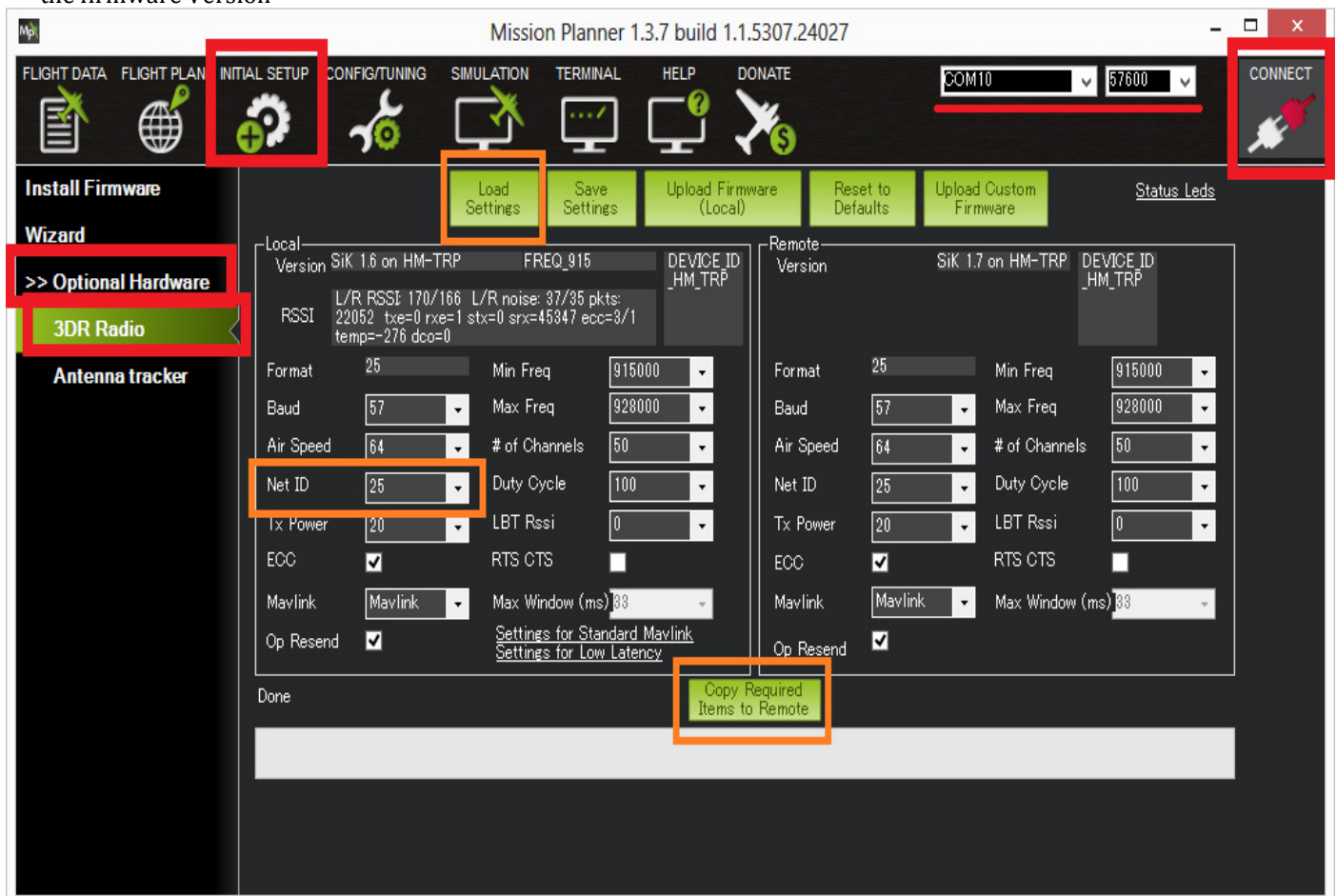
APM Mission Planner supports configuring your 3DR radios using a simple GUI interface.

Tip:

Many users will not need to configure their radios! One case where you might do so is when you use your vehicle with others — in which case you will need to specify different radio channels (Net ID).

To make changes:

- Connect one of the 3DR radios to your computer using the micro USB cable.
- Power the radio attached to the vehicle by plugging in the vehicle's battery.
- Open the Mission Planner and go to the **Initial Setup >> Optional Hardware >> 3DR Radio** page.
- Select the correct COM port and set the baud rate to 57600. Ensure the “Connect” button is in a disconnected state as shown in the image below..
- Press the **Load Settings** button and both the *Local* and *Remote* areas should fill in with values including the firmware Version



- The most common entry to change is the Net ID. The default is 25 for all 3DR radios but if you plan to fly in an area with other pilots who may be using the same radio it is best to change this to another number.



- After making changes, click the **Copy Required Items to Remote** and press **Save Settings**

Other parameters you may choose to update:

- **Baud** (default 57) : the rate at which the mission planner or vehicle communicates with the local radio. "57" = 57600 bits per second.
- **Air Speed** (default 64) : the rate at which the two radios will communicate with each other. "64" = 64kbps (kilobits per second). Setting a lower rate will increase the range of the radio but reduce the rate of data (i.e. the amount of data that can be sent in a given time) across the link.
- **ECC** (default is "on"): controls whether error correction is on or off. When on "12/24 Golay error correcting code" is used which involves sending a 16 bit CRC byte along with the data to ensure that bad data is thrown away. Unfortunately this also halves the data rate across the link but it is recommended to leave ECC on especially when the vehicle is far from home base because transmission errors increase greatly with distance.
- **MAVlink** (default is "MAVLink") : this controls whether the transmission is optimised for MAVLink packets or not. Set to "Low Latency" if using a joystick or an android tablet's virtual joystick to manually fly the vehicle. Note that information on the radio signal strength (rssi) and error rate is only sent if this parameter is set to it's default, "MAVLink".
- **Tx Power** (default 20) : the transmission power where 1=1.3milliWats, 2=1.5mW, 5=3.2mW, 8=6.3mW, 11=12.5mW, 14=25mW, 17=50mW, 20=100mW. This should be set to conform with your local regulations. [Some per-country information is below.](#)
- **Duty Cycle** (default 100) : the maximum percentage of time that the radio will transmit packets. Some regions of the world allow for higher transmit power or more frequencies if you have a duty cycle below a given threshold. So for example in Europe you can transmit on a wider range of frequencies in the 433 band if your duty cycle is below 10%. Telemetry traffic is quite 'bursty', so the average transmit time is not generally high. When you set a duty cycle below 100% then your available bandwidth will be reduced, so you will find it will only work well for telemetry at higher air speeds. A radio can be set to receive only by setting it's Duty Cycle to zero.
- **Max Window** (default 33) : ensure the GCS can send a packet to the vehicle ever 33msec. This should be kept as a low number (like 33) when the "MAVLink" setting is "Low Latency"
- **LBT Rssi** (default 0) : holds the threshold used for "listen before talk" which allows it to comply with some country's regulatory requirements. When non-zero the radio listens for quiet period of time where no signals from other radios are received before transmitting. This parameter holds the receiver signal strength below which the airwaves are considered "quiet". If this param is set to zero then LBT is disabled. Setting this to 25 (the min) is -121 dBm. Each increment above 25 raises the threshold by about 0.5dB so for example 40 equals a signal strength of 7.5dB. The full formula is:  $\text{signal\_dBm} = (\text{RSSI} / 1.9) - 127$ . The LBT implementation in the 3DR radio uses a minimum listen time of 5ms, plus randomised listen time as per the European 9.2.2.2 rules. Note that in many regions you need to implement LBT in conjunction with AFA (Adaptive Frequency Agility). The 3DR Radio implements AFA as long as you have NUM\_CHANNELS set to more than 1.

### Expected range

Typical range achieved with the 3DR radios using standard configuration and antenna is around 500m, but the range varies a lot depending on noise sources and antenna setup.

---

## Flying with a Joystick/Gamepad

Copter supports “RC overrides” which allows ground stations (including the Mission Planner) to let you fly your vehicle with a computer joystick or gamepad instead of an RC controller. Follow the instructions below to set it up.



Note:

We recommend that even if flying with a joystick, you should keep a regular transmitter/receiver connected and ready for use as a backup. In future versions of Copter, after more failsafe testing has been completed we may change this recommendation.

### What you will need

- You will need the following equipment:
- USB joystick or Gamepad such as the [Logitech F310](#) (shown above)
- a [telemetry connection](#) (such as the [3DR radio](#)) between your ground station and vehicle
- a laptop computer at the field running Mission Planner (other GCSs may also support “RC overrides” but this wiki page only covers the mission planner method)

### Setup with the Mission Planner

- Connect your USB joystick/gamepad to the laptop computer
- open the Mission Planner’s and on the Flight Data screen’s “Actions” tab push the Joystick button
- Ensure the Logitech joystick appears in the drop-down

- The “Enable” button, once pushed, will tell the mission planner to start sending commands to the vehicle so for the initial setup do not push it.
- On the “Roll” row, click the “Auto Detect” button and then waggle the control you wish to use for roll, left and right.
- The channel can be reversed with the “Reverse” checkbox
- The Expo number should be in the range of -100 to 100
- 0 = no expo
- 100 = low response around the middle, very fast response at the edges
- -100 = very fast response around the middle, less response at the edges (very few people use negative expo)
- Repeat for “Pitch”, “Throttle” and “Rudder” and push the Save button



- You will likely want to set-up other buttons for “Arm” and “Disarm”
- Instead of setting up a stick or button for ch5 (the flight mode channel), it is better to set individual buttons to initiate a flight mode by selecting “Change Mode” and then push the “Settings” button and select the flight mode from the drop-down (see pic above)
- When done, push the save button

We recommend that you continue to have your RC gear connected because it is possible to lose the telemetry connection in flight so you will need to be able to fall back to regular RC control or switch to “hardware manual” in the case of an autopilot failure. However, in most cases you can use the joystick/gamepad for all manual flying, never touching the RC transmitter.

To set it up, do the following: Once the controller is plugged in, just select “Joystick” (circled in red below) and it will open a screen that will allow you to assign functions to different joystick/gamepad buttons.

### **Testing the controls before flying**

Before flying for the first time you should test that all features work well.

To check the above controls move in the correct direction:

- Click the “Enable” button on the Joystick setup screen to enable sending messages to the vehicle
- Connect your flight controller to the computer with a USB cable
- Push the Mission Planner’s Connect button
- Go to the Initial Setup >> Mandatory Hardware >> Radio Calibration screen and ensure the green bars all move in the correct direction. Remember the bars move in the same direction as the sticks except for Pitch which moves in the opposite direction.
- If any controls are reversed, check the “Reverse” checkbox on the Joystick setup screen.

Next test you are able to arm, disarm and switch into the various flight modes (not need to connect the battery)

### **Testing the failsafe (fail safe setup to return to launch on loss of signal RTL)**

All of these tests should be performed on the ground with the battery disconnected or at least with the propellers off the vehicle.

**Simulate disabling the joystick in flight with transmitter off** (i.e. no failover to regular transmitter):

1. Turn regular transmitter off
2. Connect with the mission planner, push Joystick window’s “Enable” button and ensure RC overrides are being sent to the vehicle (use Radio Calibration screen) or Flight Data’s status screen.
3. Arm vehicle, switch to Stabilize or Loiter mode and raise the throttle
4. Ensure the motors are spinning by checking the Flight Data screen’s “ch1out” ~ “ch4out”
5. Push Joystick screen’s “Disable” button
6. “Failsafe” should appear on the HUD and the vehicle should switch to “LAND” or “RTL”
7. Repeat the above test but at step #5 actually disconnect the joystick from the computer. The results should be the same.



### Simulate failing over to the regular transmitter/receiver

1. Turn the regular transmitter on and ensure you can control the vehicle (perhaps by checking the Radio calibration page or the Flight Data screen's Status tab's "ch1in" ~ "ch8in").
2. With the regular transmitter leave the vehicle in AltHold mode and raise the throttle to mid
3. On the Joystick screen push the "Enable" button
4. Arm the vehicle in Stabilize mode and raise throttle to full (with the Joystick).
5. Check the throttle is at full in the Flight Data screen's Status tab by checking "ch3in" is very high (around 1900 usually)
6. On the Joystick screen push the "Disable" button and check the "ch3in" has dropped to a mid value (around 1500)
7. The vehicle should remain in it's current flight mode (Stabilize) but controls have been returned to the transmitter. The pilot's inputs should be reflected in the "ch1in" ~ "ch8in" values. Switch the vehicle to AltHold mode by moving the flight mode switch.

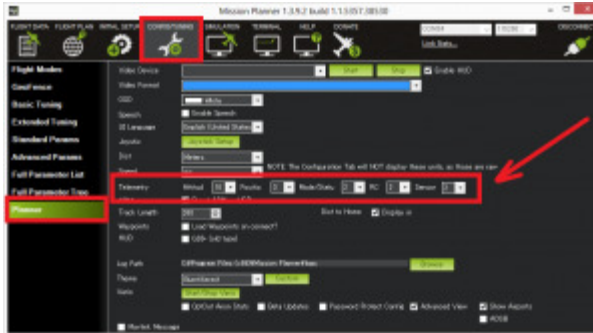
### Simulate loss of radio contact:

If radio contact is lost, the vehicle should respond as if the Joystick was "Disabled" or disconnected from the PC. You can test this by repeating the test above but instead of pushing the "Disable" button on the Joystick screen, disconnect the radio. In order to see the status of the vehicle you will need to connect two Mission Planner's separately (only possible on Pixhawk). The Mission Planner with the Joystick should be connected through telemetry and the 2nd Mission Planner should be connected through a USB connection.

### Reducing lag in the controls

It is nearly impossible to make the Joystick as responsive as a regular transmitter but lag can be reduced by:

- reducing the rate of other telemetry data being sent between the vehicle and GCS from the MP's Config/Tuning >> Planner page



- If using a 3DR radio or RFD900 radio these can be put into Low Latency mode as described [here](#).

## Calibrate Gimbal:

### Driver Updates - Windows Method

This How-To is assuming you have a Windows PC. To be successful your PC will need the correct FTDI driver installed. The driver which has worked best for my boards and set-ups is the FTDI COM driver 2.8.24.0. If you have 2.8.28.0, 2.8.30.0 or higher these likely **will not** work.

**NOTE: Check that the cable you are using actually works! I have run into the bad cable blues myself and nothing can be more frustrating than trying to fix a driver problem with a bad cable!!**

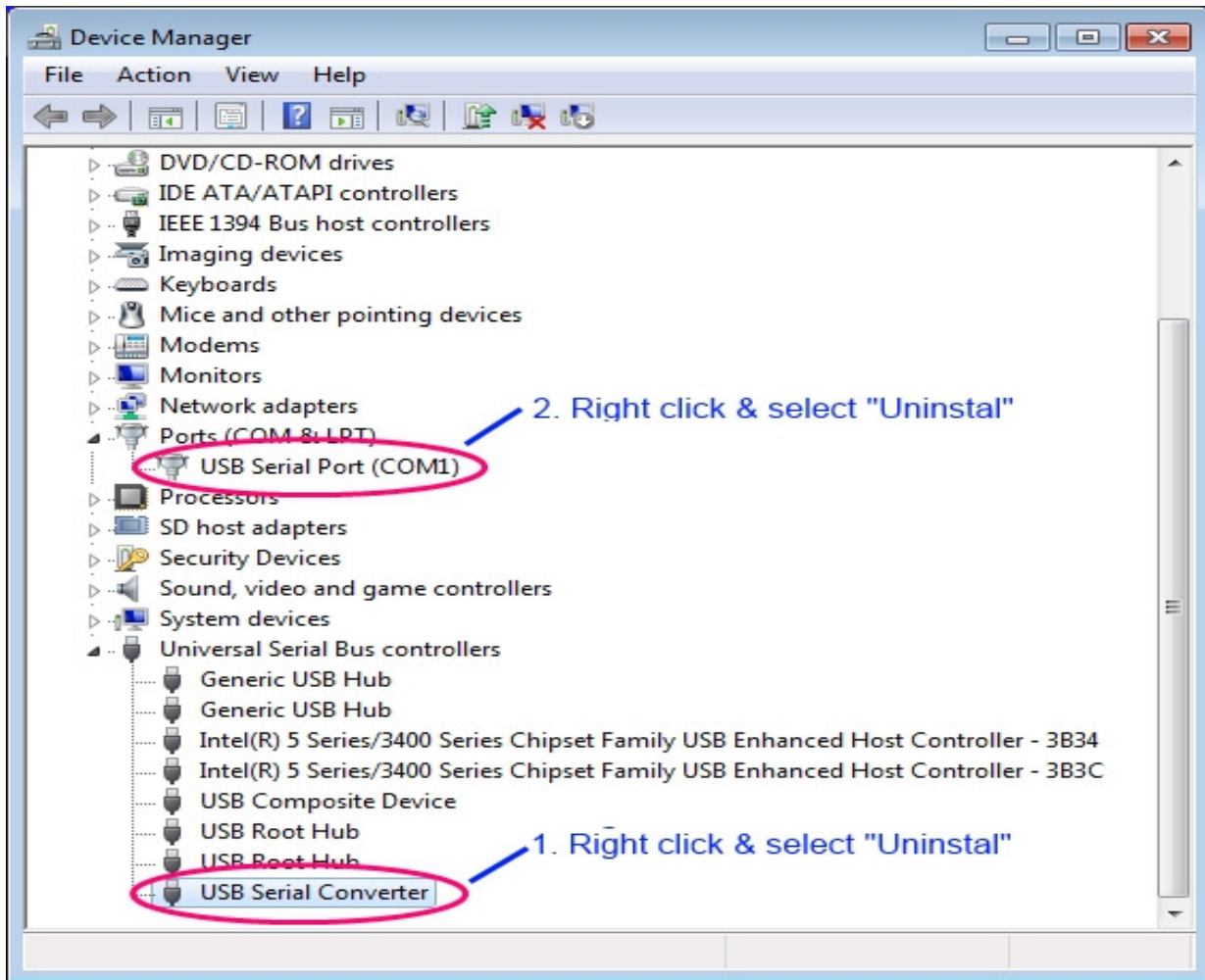
If for some reason 2.8.24.0 still doesn't connect you might try 2.8.14.0 which some have reported success with. Here are the links for these driver files:

[2.8.24 FTDI Driver Package](#)

[2.8.14 FTDI Driver Package](#)

To load either of these older drivers follow these steps **exactly** in order:

1. Disconnect your internet to prevent Windows from auto fetching the latest driver  
(Keep it disconnected from the internet until the board is working correctly)
2. Plug in your board
3. Go to device manager
4. Go to View and select "show all the hidden devices"
5. Expand "Universal Serial Bus Controllers" and "Ports (COM & LPT)"



6. Right click "USB Serial Converter" and select "Uninstall"

a. **IMPORTANT!** Also check the box "Delete the Driver software for this device"



7. Right click "USB Serial Port (COMx)" and select "Uninstall"

a. **IMPORTANT!** Also check the box "Delete the Driver software for this device"



8. Close all windows and re-start your system

9. Connect the board to USB and look for the warning that there is no driver for this device

10. In Device Manager, update with the driver that you have downloaded

(On some computers you may need to reboot the system)

11. Plug in the board, go to Device Manager and check to see the correct driver has been installed and the COM port is a number below 9

Once you have the driver installed you are ready to begin upgrading and tuning.

#### Driver Clean-Up - 3rd Party Tool

Many have reported having driver issues "no matter what they try" and having experienced this myself on one of my computers I can see why. Drivers are funny things and if you are not careful they are easy to corrupt or mis-install. Add to these Windows automation routines which constantly try and keep things up-to-date even when you may not want them to and you have a recipe for driver chaos!

On my laptop I had unsuccessfully tried to install the 2.8.24 driver a couple of times before I realized I had to check the box to uninstall the original driver firmware (2.8.28). So, I went back, checked the box and \*.24 seemed to install just fine. Then on one of my boards I could get the GUI to work but Arduino would not allow an upload. I kept getting the dreaded AVRdude error. When I checked the driver file in Windows Device Manager, it showed the correct \*.24 driver and the settings were correct. This caused some serious hours of cursing and fiddling (and eventually learning :)).

In my search for "WHY" this was happening I happened on a set of free tools from Nirsoft, a company which provides free utilities (links below). Three of these utilities, DevManView, USBDeview and DriverView are awesome tools for viewing exactly what is installed in your system and for acting on the ones you want to change or delete without having to go through all the vague screens provided in Windows.

[http://www.nirsoft.net/utills/usb\\_devices\\_view.html](http://www.nirsoft.net/utills/usb_devices_view.html)

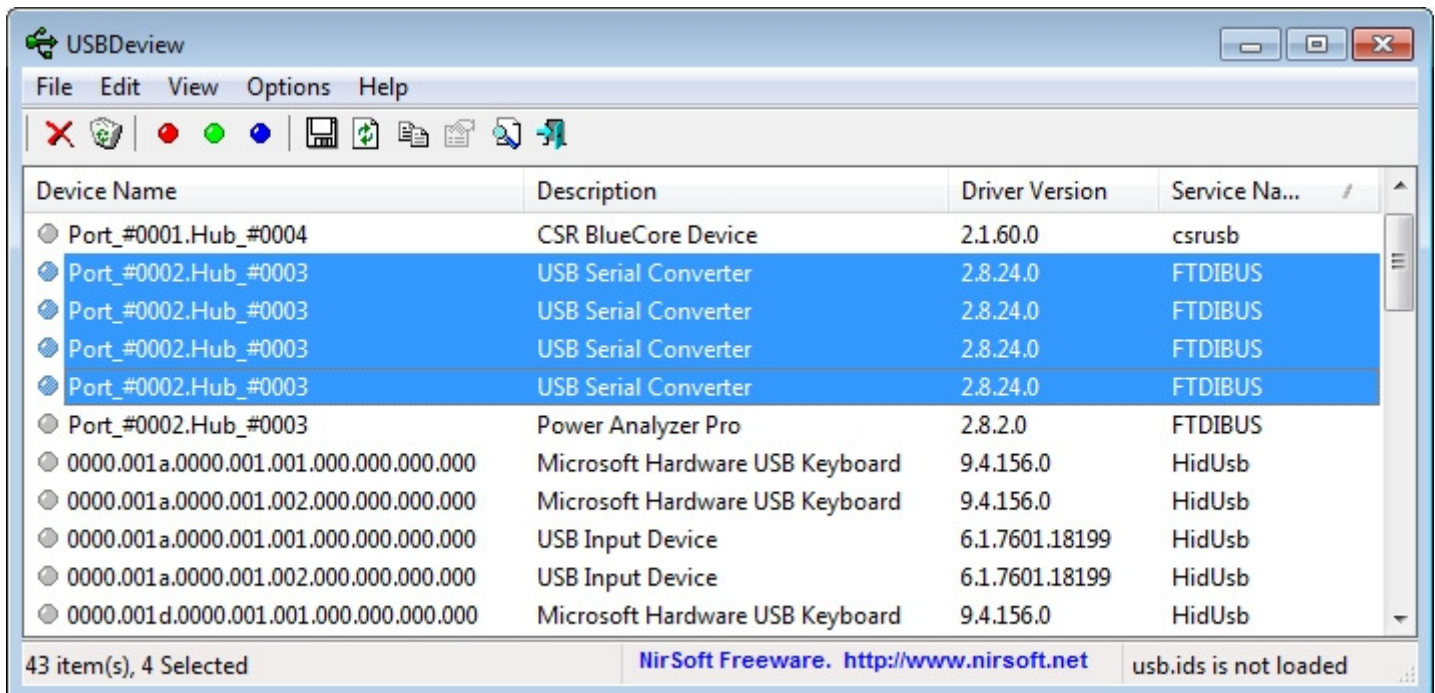


I encourage you to download all three of these and have a look at what you can see. I suspect you will be surprised at how many instances of some of these drivers and devices exist in your system.

To alleviate my driver problem I used the utility "USBDeview" which shows all the USB devices Windows has a record of. I immediately saw I had multiple instances of the \*.24, \*.28 and \*.30 FTDI drivers which didn't seem right as I thought I had uninstalled the \*.28 and \*.30 drivers. I also saw that when I plugged in one of my Martinez v3 boards the USBDeview tool reported it was using \*.30 but Windows was reporting that it was using \*.24!

To fix this using USBDeview

- Disconnect from the internet!
- Open USBDeview using "Run as Administrator" by right clicking the USBDeview.exe file and selecting this option.
- Go to "View/Choose Columns" and de-select everything but Device name, Description, Service name, and Driver Version.
- Click the column heading for "Service Name" to sort.



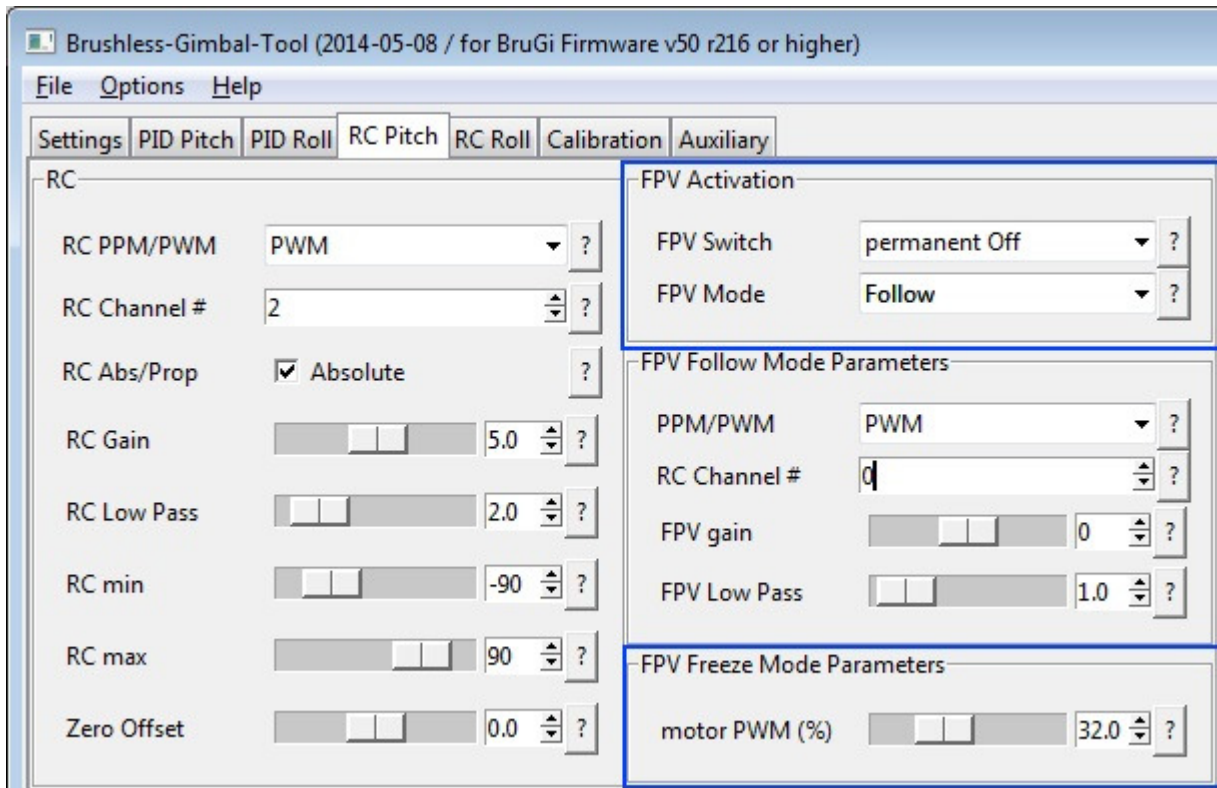
You will now see all the FTDI drivers stacked together and you can select them as a group. Select all the \*.24, \*.28 and \*.30 drivers, right click and uninstall them. Now, with your internet access still turned off, load the \*.24 driver.

Now, go back to #8 in the section above and complete the driver install. Done!

**NOTE 1:** You need to have at least enough tuning done for the Gimbal to be stable (not shaking) for this to be successful.

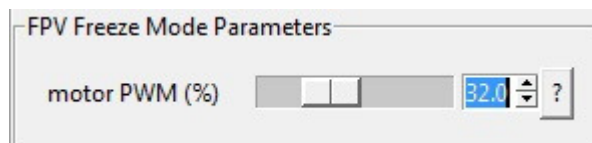
**NOTE 2:** The method described here applies to all firmware which have the FPV Follow and Freeze modes available. These functions were made available on all 050 firmware. To check motor direction on earlier firmwares please go to Motor Direction Check - Early BruGi Firmware.

1. With the board powered and the GUI active change to the RC Pitch tab. In this tab you will be making changes to "FPV Activation" and "FPV Freeze Mode Parameters"



RC Pitch Tab/FPV menus

2. In the "FPV Freeze Mode Parameters" menu, set the value equal to or greater than the PWM value you have set for Pitch



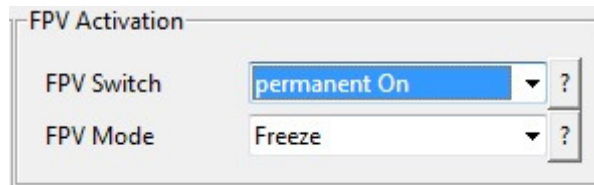
FPV Freeze Mode PWM value

3. Next, go to the "FPV Activation" menu and change "FPV Mode" to **Freeze**



FPV Mode setting

4. You will now select the menu option **Permanent On** in the "FPV Activation" menu selection "FPV Switch" and watch how the Gimbal reacts



FPV Switch setting

If the Gimbal has no reaction, the Pitch motor direction is correct, go to step 5

If the Gimbal immediately drops or moves to an alternate position then the motor direction is incorrect

To correct, go to the Pitch tab and select or de-select the "Reverse" check box and then re-test

5. Repeat for the Roll function

6. Make sure you have re-selected "Permanent Off" and save to eprom!

To finish this off, it's a good practice after doing this to mark the motor connections so that they are always plugged in exactly the same way to the board every time.

### **PWM & BGC PID Parameter Tuning**

This tuning method should work on any BruGi software/board providing you understand there may be differences in the values and/or percentages.

### **From the BruGi project - Documentation**

**NOTE:** Make sure you use the GUI that came with the firmware version you are using

1. Live tuning start-up sequence:

- a. With camera installed and balanced, connect battery power to the Gimbal
- b. Connect the USB
- c. Start the GUI and click connect
2. Zero out your max PWM and PID values in **both** Pitch and Roll
3. Beginning with **Roll**, start with a value of 5 on P, I & D (on v048, start with P=5, I=0.1, & D=5)
4. Raise the PWM value until the Roll motor begins to vibrate then back-off enough to completely stop the vibration
5. Raise the P value until the motor just begins to vibrate
6. Raise the D value until the motor becomes quiet
7. Continue going back and forth between P and D until you have reached the highest P value you can
8. Add the values for P and D together and divide by 2 to get a starting value for I (049 and up firmware, for 048, change I to 0.5)
9. Give the gimbal a light tap on one side while watching the GUI trace to make sure it immediately recovers
  - a. If it starts to hum or takes a long time to settle down, I is too high, reduce by 10%
  - b. If it returns immediately turn I up by 10% and recheck. The goal is the highest value of I without causing it to vibrate after a "tap"
10. **Roll** Tuning should be complete! Calibrate the Gyro and save these settings to your gimbal. THIS IS VERY IMPORTANT. If you are on firmware r161 or lower follow these 2 steps:
  1. Click on the options tab and then click "Save to Board"
  2. Click on options tab and then click "Save to Flash".If you are on a newer firmware than r161, simply click on options and "Save to flash".
11. Now, go to Pitch settings and repeat steps 3 through 10.

Time for a test flight!

Make sure you write down your final settings each time you tune as you may have to go back and make adjustments for balance or equipment changes.

Version Settings

The following are examples of settings I have used in various versions of the firmware. (Rounded to the nearest whole number)

**BruGi Settings by Firmware**

Pitch	048	049_r77	049_r161	049_r171	049_r187	049_r188	049_r189	050_r191	050_r207
P	12	10	10	18	25	20	19	29	22
I	0.05	45	16	64	57	15	30	40	87
D	2	15	28	31	40	40	30	51	26
PWM	46	25	22	20	13	20	19	22	28
Roll									
P	60	30	30	28	35	38	31	41	36
I	0.05	40	8	30	28	55	45	51	183
D	18	45	40	35	50	70	50	60	78
PWM	80	31	72	30	50	70	66	70	77

**A Note on Board Operation LED's**

There have been many instances where the LED lights do "funny things". I don't claim to know precisely what they mean as there are different LED configurations on different boards. Here is a quote I found on the subject direct out of one of the SBGC manuals and it primarily refers to the SBGC architecture.

There are 2 LEDs onboard. Red led lights when power is connected. Green/blue LED signals actual state of the system:

- \* LED is off — pause before calibration, to take hands off or to level gimbal.
- \* LED blinks slowly – Calibration in action. Freeze gimbal during this process.
- \* LED blinks fast — system error, stabilization cannot be performed. To check error description, connect to GUI.
- \* LED is on — normal operation mode.
- \* LED is on, but blinks irregularly – I2C errors appears.

**Do compass mot:**

**CompassMot – compensation for interference from the power wires, ESCs and motors**

This is recommended for vehicles that have only an internal compass and on vehicles where there is significant interference on the compass from the motors, power wires, etc. Compass Mot only works well if you have [a battery current monitor](#) because the magnetic interference is linear with current drawn. It is technically possible to set-up CompassMot using throttle but this is not recommended.

Please follow these instructions:

- Enable the current monitor (aka [Power Module](#))

- Disconnect your props, flip them over and rotate them one position around the frame. In this configuration they should push the copter down into the ground when the throttle is raised
- Secure the copter (perhaps with tape) so that it does not move
- Turn on your transmitter and keep throttle at zero
- Connect your vehicle's Lipo battery
- Connect your flight controller to your computer with the usb cable
- If using AC3.2:
  - Open the Initial Setup >> Optional Hardware >> Compass/Motor Calib screen
  - Press the Start button



- If using AC3.1.5:
  - open the Terminal screen, press the “Connect APM” or “Connect PX4” button
  - At the prompt type *setup* and then *compassmot*.
  - If the Power Module/Current monitor is enabled you should see “measuring compass vs CURRENT”



- You should hear your ESCs arming beep
- Raise the throttle slowly to between 50% ~ 75% (the props will spin!) for 5 ~ 10 seconds
- Quickly bring the throttle back down to zero
- Press the Finish button (AC3.2) or Press enter (AC3.1.5) to complete the calibration
- Check the % of interference displayed. If it is less than 30% then your compass interference is acceptable and you should see good Loiter, RTL and AUTO performance. If it is 31% ~ 60% then the interference is in the “grey zone” where it may be ok (some users are fine, some are not). If it is higher than 60% you should try moving your APM/PX further up and away from the sources of interference or consider purchasing an external compass or GPS+compass module.

---

## Disable pre arm checks:

### Recognising which Pre-Arm Check has failed using the GCS

The pilot will notice a pre-arm check failure because he/she will be unable to arm the copter and the LED will be flashing yellow. To determine exactly which check has failed:

1. Connect the Flight Controller to the ground station using a USB cable or Telemetry.
2. Ensure the GCS is connected to the vehicle (i.e. on Mission Planner and push the “Connect” button on the upper right).
3. Turn on your radio transmitter and attempt to arm the vehicle (regular procedure using throttle down, yaw right)
4. The first cause of the Pre-Arm Check failure will be displayed in red on the HUD window

## Failure messages

RC failures (i.e. transmitter/receiver failures):

**RC not calibrated** : the radio calibration has not been performed. RC3\_MIN and RC3\_MAX must have been changed from their default values (1100 and 1900) and for channels 1 to 4, the MIN must be less than 1300 and the MAX greater than 1700.

Barometer failures:

**Baro not healthy** : the barometer sensor is reporting that it is unhealthy which is normally a sign of a hardware failure.

**Alt disparity** : the barometer altitude disagrees with the inertial navigation (i.e. Baro + Accelerometer) altitude estimate by more than 2 meters. This message is normally short-lived and can occur when the flight controller is first plugged in or if it receives a hard jolt (i.e. dropped suddenly). If it does not clear the accelerometers may need to be calibrated or there may be a barometer hardware issue.

Compass failures:

**Compass not healthy**: the compass sensor is reporting that it is unhealthy which is a sign of a hardware failure.

**Compass not calibrated** : the compass(es) has not been calibrated. the COMPASS\_OFS\_X, Y, Z parameters are zero or the number or type of compasses connected has been changed since the last compass calibration was performed.

**Compass offsets too high** : the primary compass's offsets length (i.e.  $\sqrt{x^2+y^2+z^2}$ ) are larger than 500. This can be caused by metal objects being placed too close to the compass. If only an internal compass is being used (not recommended), it may simply be the metal in the board that is causing the large offsets and this may not actually be a problem in which case you may wish to disable the compass check.

**Check mag field** : the sensed magnetic field in the area is 35% higher or lower than the expected value. The expected length is 530 so it's  $> 874$  or  $< 185$ . Magnetic field strength varies around the world but these wide limits mean it's more likely the compass calibration has not calculated good offsets and should be repeated.

**Compasses inconsistent** : the internal and external compasses are pointing in different directions (off by  $>45$  degrees). This is normally caused by the external compasses orientation (i.e. COMPASS\_ORIENT parameter) being set incorrectly.

GPS related failures:

**GPS Glitch** : the GPS is glitching and the vehicle is in a flight mode that requires GPS (i.e. Loiter, Position Hold, etc) and/or the circular fence is enabled.

**Need 3D Fix** : the GPS does not have a 3D fix and the vehicle is in a flight mode that requires the GPS and/or the circular fence is enabled.



**Bad Velocity** : the vehicle's velocity (according to inertial navigation system) is above 50cm/s. Issues that could lead to this include the vehicle actually moving or being dropped, bad accelerometer calibration, GPS updating at below the expected 5hz.

**High GPS HDOP** : the GPS's HDOP value (a measure of the position accuracy) is above 2.0 and the vehicle is in a flight mode that requires GPS and/or the circular fence is enabled. This may be resolved by simply waiting a few minutes, moving to a location with a better view of the sky or checking sources of GPS interference (i.e. FPV equipment) are moved further from the GPS. Alternatively the check can be relaxed by increasing the GPS\_HDOP\_GOOD parameter to 2.2 or 2.5. Worst case the pilot may disable the fence and take-off in a mode that does not require the GPS (i.e. Stabilize, Alt Hold) and switch into Loiter after arming but this is not recommended.

Note: the GPS HDOP can be readily viewed through the Mission Planner's Quick tab as shown below.



INS checks (i.e. Accelerometer and Gyro checks):

**INS not calibrated** : some or all of the accelerometer's offsets are zero. The accelerometers need to be calibrated.

**Accels not healthy** : one of the accelerometers is reporting it is not healthy which could be a hardware issue. This can also occur immediately after a firmware update before the board has been restarted.

**Accels inconsistent** : the accelerometers are reporting accelerations which are different by at least 1m/s/s. The accelerometers need to be re-calibrated or there is a hardware issue.

**Gyros not healthy** : one of the gyroscopes is reporting it is unhealthy which is likely a hardware issue. This can also occur immediately after a firmware update before the board has been restarted.

**Gyro cal failed** : the gyro calibration failed to capture offsets. This is most often caused by the vehicle being moved during the gyro calibration (when red and blue lights are flashing) in which case unplugging the battery and plugging it in again while being careful not to jostle the vehicle will likely resolve the issue. Sensors hardware failures (i.e. spikes) can also cause this failure.

**Gyros inconsistent** : two gyroscopes are reporting vehicle rotation rates that differ by more than 20deg/sec. This is likely a hardware failure or caused by a bad gyro calibration.

Board Voltage checks:

**Check Board Voltage** : the board's internal voltage is below 4.3 Volts or above 5.8 Volts.

If powered through a USB cable (i.e. while on the bench) this can be caused by the desktop computer being unable to provide sufficient current to the flight controller – try replacing the USB cable. If powered from a battery this is a serious problem and the power system (i.e. Power Module, battery, etc) should be carefully checked before flying.

### Parameter checks:

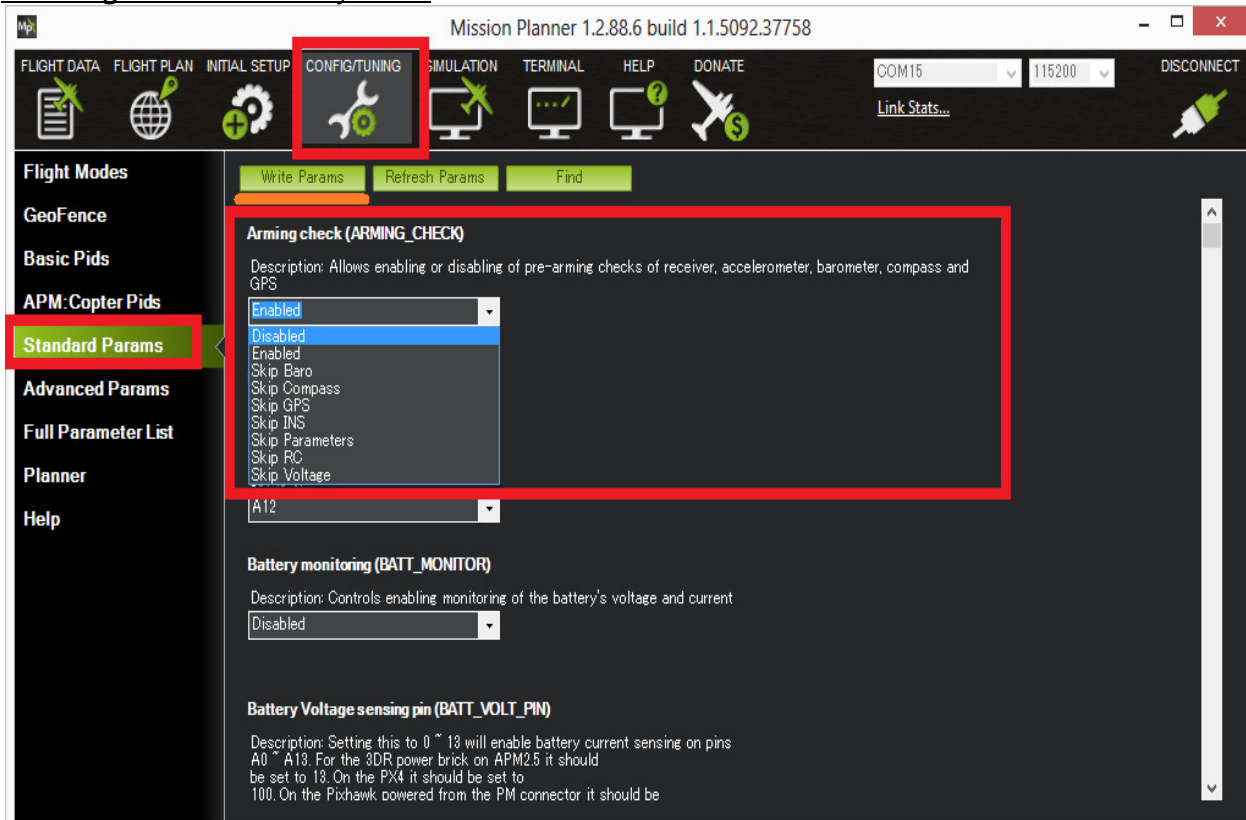
**Ch7&Ch8 Opt cannot be same** : Ch7 and Ch8 auxiliary switches are set to the same option which is not permitted because it could lead to confusion.

**Check FS\_THR\_VALUE** : the radio failsafe pwm value has been set too close to the throttle channels (i.e. ch3) minimum.

**Check ANGLE\_MAX** : the ANGLE\_MAX parameter which controls the vehicle's maximum lean angle has been set below 10 degrees (i.e. 1000) or above 80 degrees (i.e. 8000).

**ACRO\_BAL\_ROLL/PITCH** : the ACRO\_BAL\_ROLL parameter is higher than the Stabilize Roll P and/or ACRO\_BAL\_PITCH parameter is higher than the Stabilize Pitch P value. This could lead to the pilot being unable to control the lean angle in ACRO mode because the Acro Trainer stabilization would overpower the pilot's input.

### Disabling the Pre-arm Safety Check



If you are confident that the pre-arm check failure is not a real problem you can disable the checks by:

- Connecting your Flight Controller to the Mission Planner
- Go to Mission Planner's Config/Tuning >> Standard Params screen
- set the Arming Check drop-down to "Disabled" or one of the "Skip" options which more effectively skips the item causing the failure.
- Push the "Write Params" button

Ideally however you should determine the cause of the pre-arm failure and if it can be resolved, return the Arming Check parameter back to "Enabled"

---

## PID Tuning:

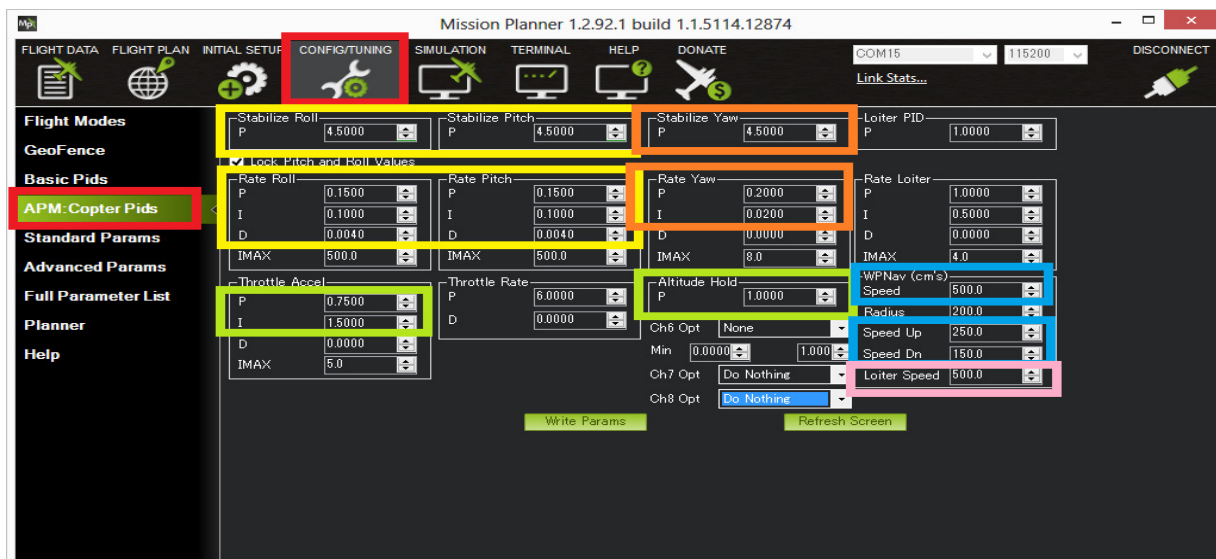
Multi-rotor PID Tuning Guide <https://youtu.be/jg3FVSL94qw>

You must run the MAGNETOMETER CALIBRATION before trying to tune the attitude controller. The attitude estimator works much more precisely with a properly calibrated mag!

NEVER do Multi-rotor tuning with carbon fiber or carbon fiber reinforced blades. NEVER use damaged blades.

For SAFETY reason, the default gains are set to small value. You HAVE TO increase the gains before you can expect any control responses.

### Overview



The default [PID gains](#) are meant for the [3DR quad copter](#) although they work on a wide variety of frames. Still to get optimal performance you may need to adjust them which can be done through the Mission Planner's Config/Tuning >> Copter Pids screen. The screen shot above shows the most important parameters for Roll/Pitch (yellow), Yaw (orange), [Altitude hold](#) (green), [Loiter](#) (pink) and [Waypoint navigation](#) (blue). The most important parameter to get right is the Rate Roll P (and by default Rate Pitch P is locked to the same value) as discussed [here](#).

Normally it's best to start by tuning the Rate Roll/Pitch P in Stabilize mode then move onto tuning altitude hold in AltHold mode, then Loiter (which often needs no tuning) and finally the waypoint navigation performance in Auto mode.

Note:

[Dave C's AC2.8.1 tuning guide](#) has good information for tuning for rate roll and pitch but altitude hold, Loiter and navigation has changed dramatically since AC2.8.1 so those sections are no longer valid.

### **Roll/Pitch tuning**

The Stabilize Roll/Pitch and Rate Roll/Pitch parameters, highlighted in yellow in the screen shot above control the roll pitch response.

The Rate parameters which convert the desired rotation rate into a motor output are the most important.

The [Rate Roll and Pitch P tuning page](#) has the most information on tuning them.

The Stabilize Roll/Pitch P converts the desired angle into a desired rotation rate which is then fed to the Rate controller.

- A higher value will make the copter more responsive to roll/pitch inputs, a lower value will make it smoother
- If set too high, the copter will oscillate on the roll and/or pitch axis
- If set too low the copter will become sluggish to inputs

More information on tuning the roll and pitch can be found on the [Stabilize mode page's Tuning section](#).

An objective view of the overall Roll and Pitch performance can be seen by graphing the [dataflash log's](#) ATT message's Roll-In vs Roll and Pitch-In vs Pitch. The "Roll" (i.e. actual roll) should closely follow the "Roll-In" while in Stabilize or AltHold modes. Pitch should similarly closely follow Pitch-In.

Alternatively you may wish to try tuning both the rate and stabilize (i.e. angular) parameters using the [AutoTune feature](#).

### **Yaw tuning**

The Stabilize Yaw and Rate Yaw parameters, highlighted in orange in the screen shot above control the yaw response. It's rare that the yaw requires much tuning.

Similar to roll and pitch if either Stabilize Yaw P or Rate Yaw P is too high the copter's heading will oscillate. If they are too low the copter may be unable to maintain it's heading.

As mentioned on the [Stabilize mode's tuning section](#), the ACRO\_YAW\_P parameter controls how quickly copter rotates based on a pilot's yaw input. The default of 4.5 commands a 200 deg/sec rate of rotation when the yaw stick is held fully left or right. Higher values will make it rotate more quickly.

### **Altitude Tuning**

The Altitude hold related tuning parameters are highlighted in green in the screen shot above.

The Altitude Hold P is used to convert the altitude error (the difference between the desired altitude and the actual altitude) to a desired climb or descent rate. A higher rate will make it more aggressively attempt to maintain it's altitude but if set too high leads to a jerky throttle response.

The Throttle Rate (which normally requires no tuning) converts the desired climb or descent rate into a desired acceleration up or down.

The Throttle Accel PID gains convert the acceleration error (i.e the difference between the desired acceleration and the actual acceleration) into a motor output. The 1:2 ratio of P to I (i.e. I is twice the size of P) should be maintained if you modify these parameters. These values should never be increased but for very powerful copters you may get better response by reducing both by 50% (i.e P to 0.5, I to 1.0).

See the [Altitude Hold flight mode page](#) for more information.

### **Loiter Tuning**

Generally if Roll and Pitch are tuned correctly, the [GPS](#) and [compass](#) are set-up and performing well and [vibration levels](#) are acceptable, Loiter does not require much tuning but please see the [Loiter Mode](#) page for more details on tunable parameters including the horizontal speed.

### **In-flight tuning**

A single parameter's value can be tuned in flight using the transmitter's Ch6 tuning knob. A specific example for tuning the Rate Roll/Pitch P values can be found on the [Rate Roll and Pitch P tuning wiki page](#). Please see the Ch6 Opt drop-down on the Mission Planner's Config/Tuning > Copter Pids's screen for a full list of parameters that can be tuned.

After setting the Ch6 Opt value, the Min and Max should also be set to reasonable values (i.e. non-zero and also not unreasonably high) and then the "Refresh screen" button should be pushed by tuning the Ch6 knob to ensure that the parameter is updating correctly.

This tutorial is valid for all multi rotor setups (AR.Drone, PWM Quads / Hexa / Octo setups). Proportional, Integral, Derivative controllers are the most widespread control technique. There are substantially better performing control techniques (LQR/LQG) from the model predictive control, since these techniques require a more or less accurate model of the system, they not as widely used. The goal of all PX4 control infrastructure is move as soon as possible on MPC, since not for all supported systems models are available, PID tuning is very relevant (and PID control sufficient for many cases).

### **In depth parameters tuning:**

The PX4 multirotor\_att\_control app executes an outer loop of orientation controller, controlled by parameters:

- Roll control (MC\_ROLL\_P)
- Pitch control (MC\_PITCH\_P)

- Yaw control (MC\_YAW\_P)

And an inner loop with three independent PID controllers to control the attitude rates:

- Roll rate control (MC\_ROLLRATE\_P, MC\_ROLLRATE\_I, MC\_ROLLRATE\_D)
- Pitch rate control (MC\_PITCHRATE\_P, MC\_PITCHRATE\_I, MC\_PITCHRATE\_D)
- Yaw rate control (MC\_YAWRATE\_P, MC\_YAWRATE\_I, MC\_YAWRATE\_D)

The outer loop's output are desired body rates (e.g. if the multirotor should be level but currently has 30 degrees roll, the control output will be e.g. a rotation speed of 60 degrees per second). The inner rate control loop changes the rotor motor outputs so that the copter rotates with the desired angular speed.

The gains actually have an intuitive meaning, e.g.: if the MC\_ROLL\_P gain is 6.0, the copter will try to compensate 0.5 radian offset in attitude (~30 degrees) with 6 times the angular speed, i.e. 3 radians/s or ~170 degrees/s. Then if gain for the inner loop MC\_ROLLRATE\_P is 0.1 then thrust control output for roll will be  $3 * 0.1 = 0.3$ . This means that it will lower the speed of rotors on one side by 30% and increase the speed on the other one to induce angular momentum in order to go back to level.

There is also MC\_YAW\_FF parameter that controls how much of user input need to feed forward to yaw rate controller. 0 means very slow control, controller will start to move yaw only when sees yaw position error, 1 means very responsive control, but with some overshoot, controller will move yaw immediately, always keeping yaw error near zero.

### **Motor Band / Limiting**

As the above example illustrates, under certain conditions it would be possible that one motor gets an input higher than its maximum speed and another gets an input lower than zero. If this happens, the forces created by the motors violate the control model and the multi rotor will likely flip. To prevent this, the multi rotor mixers on PX4 include a band-limit. If one of the rotors leaves this safety band, the total thrust of the system is lowered so that the relative percentage that the controller did output can be satisfied. As a result the multi rotor might not climb or loose altitude a bit, but it will never flip over. The same for lower side, even if commanded roll is large, it will be scaled to not exceed commanded summary thrust and copter will not flip on takeoff at near-zero thrust.

## Step 1: Preparation

First of all set all parameters to initial values:

1. Set all MC\_XXX\_P to zero (ROLL, PITCH, YAW)
2. Set all MC\_XXXRATE\_P, MC\_XXXRATE\_I, MC\_XXXRATE\_D to zero, except MC\_ROLLRATE\_P and MC\_PITCHRATE\_P
3. Set MC\_ROLLRATE\_P and MC\_PITCHRATE\_P to a small value, e.g. 0.02
4. Set MC\_YAW\_FF to 0.5

All gains should be increased very slowly, by 20%-30% per iteration, and even 10% for final fine tuning. Note, that too large gain (even only 1.5-2 times more than optimal!) may cause very dangerous oscillations!

## Step 2: Stabilize Roll and Pitch Rates

### P Gain Tuning

Parameters: MC\_ROLLRATE\_P, MC\_PITCHRATE\_P.

If copter is symmetrical, then values for ROLL and PITCH should be equal, if not - then tune it separately.

Keep the multi rotor in your hand and increase the thrust to about 50%, so that the weight is virtually zero. Tilt it in roll or pitch direction, and observe the response. It should mildly fight the motion, but it will NOT try to go back to level. If it oscillates, tune down RATE\_P. Once the control response is slow but correct, increase RATE\_P until it starts to oscillate. Cut back RATE\_P until it does only mildly oscillate or not oscillate any more (about 10% cutback), just over-shoots. Typical value is around 0.1.

### **D Gain Tuning**

Parameters: MC\_ROLLRATE\_D, MC\_PITCHRATE\_D.

Assuming the gains are in a state where the multi rotor oscillated and RATE\_P was slightly reduced. Slowly increase RATE\_D, starting from 0.01. Increase RATE\_D to stop the last bit of oscillation. If the motors become twitchy, the RATE\_D is too large, cut it back. By playing with the magnitudes of RATE\_P and RATE\_D the response can be fine-tuned. Typical value is around 0.01...0.02.

In QGroundControl you can plot roll and pitch rates (ATTITUDE.rollspeed/pitchspeed). It must not oscillate, but some overshoot (10-20%) is ok.

### **I Gain Tuning**

If the roll and pitch rates never reach the setpoint but have an offset, add MC\_ROLLRATE\_I and MC\_PITCHRATE\_I gains, starting at 5-10% of the MC\_ROLLRATE\_P gain value.

Step 3: Stabilize Roll and Pitch Angles

### **P Gain Tuning**

Parameters: MC\_RATE\_P, MC\_RATE\_P.

1. Set MC\_ROLL\_P and MC\_PITCH\_P to a small value, e.g. 3

Keep the multi rotor in your hand and increase the thrust to about 50%, so that the weight is virtually zero. Tilt it in roll or pitch direction, and observe the response. It should go slowly back to level. If it oscillates, tune down P. Once the control response is slow but correct, increase P until it starts to oscillate. Optimal response is some overshoot (~10-20%). After getting stable response fine tune RATE\_P, RATE\_D again.

In QGroundControl you can plot roll and pitch (ATTITUDE.roll/pitch) and control (ctrl0, ctrl1). Attitude angles overshoot should be not more than 10-20%.

Step 4: Stabilize Yaw Rate

### **P Gain Tuning**

Parameters: MC\_YAWRATE\_P.

1. Set MC\_YAWRATE\_P to small value, e.g. 0.1

Keep the multi rotor in your hand and increase the thrust to about 50%, so that the weight is virtually zero. Turn it around its yaw axis, observe the response. The motor sound should change and the system should fight the yaw rotation. The response will be substantially weaker than roll and pitch, which is fine. If it oscillates or becomes twitchy, tune down RATE\_P. If response is very large even on small movements (full throttle spinning vs idle spinning propellers) reduce RATE\_P. Typical value is around 0.2...0.3.



The yaw rate control, if very strong or oscillating can deteriorate the roll and pitch response. Check the total response by turning around, roll, pitch and yaw.

Step 5: Stabilize Yaw Angle

### **P Gain Tuning**

Parameters: MC\_YAW\_P.

1. Set MC\_YAW\_P to a low value, e.g. 1

Keep the multi rotor in your hand and increase the thrust to about 50%, so that the weight is virtually zero. Rotate it around yaw, and observe the response. It should go slowly back to the initial heading. If it oscillates, tune down P. Once the control response is slow but correct, increase P until the response is firm, but it does not oscillate. Typical value is around 2...3.

Look at ATTITUDE.yaw in QGroundControl. Yaw overshoot should be not more than 2-5% (less than for attitude).

### **Feed Forward Tuning**

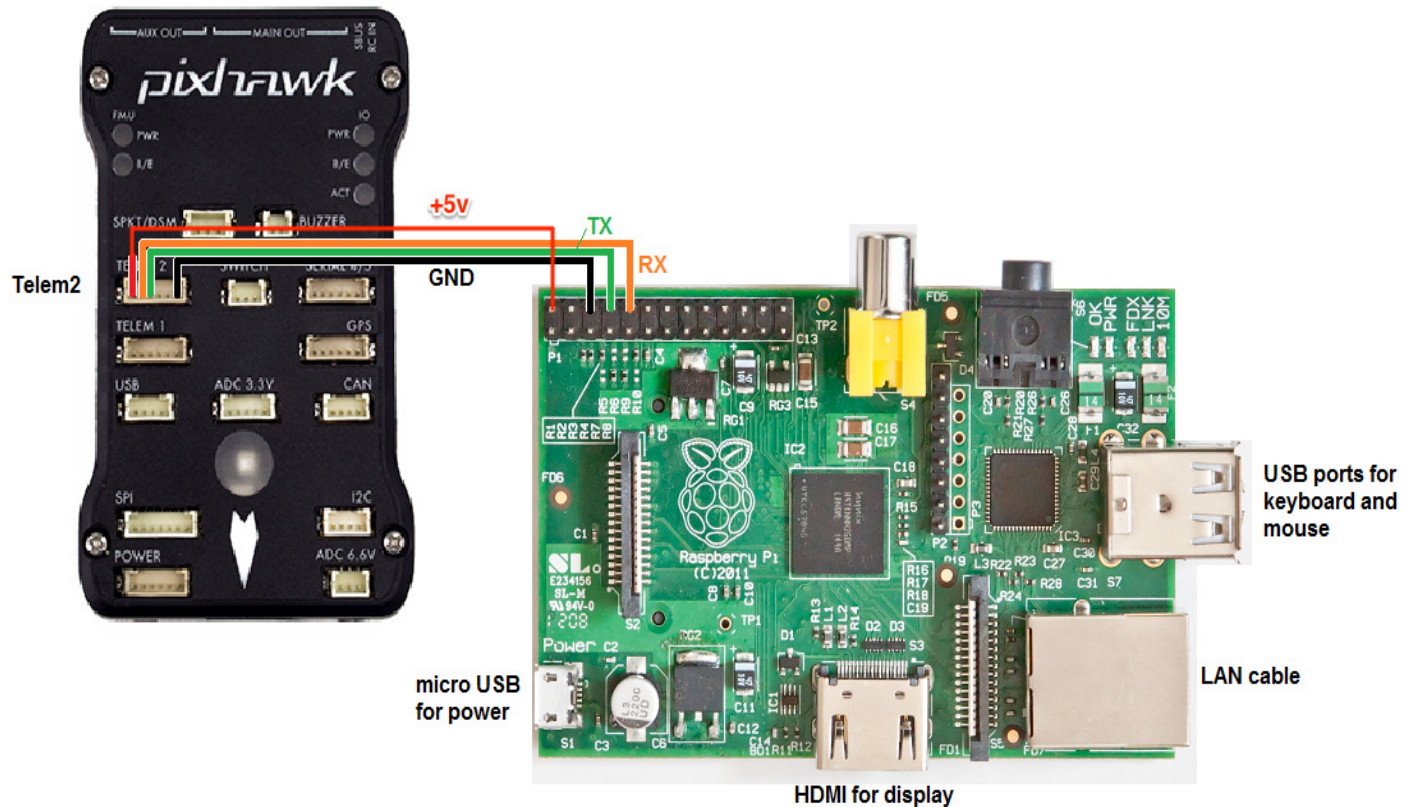
Parameters: MC\_YAW\_FF.

This parameter is not critical and can be tuned in flight, in worst case yaw responses will be sluggish or too fast. Play with FF parameter to get comfortable responses. Valid range is 0...1. Typical value is 0.8...0.9. (For aerial video optimal value may be much smaller to get smooth response.) Look at ATTITUDE. yaw in QGroundControl. Yaw overshoot should be not more than 2-5% (less than for attitude).

---

## **Raspberry PI setup:**

Connecting the Pixhawk and RP I, do not connect the +5V to pixy hawk!



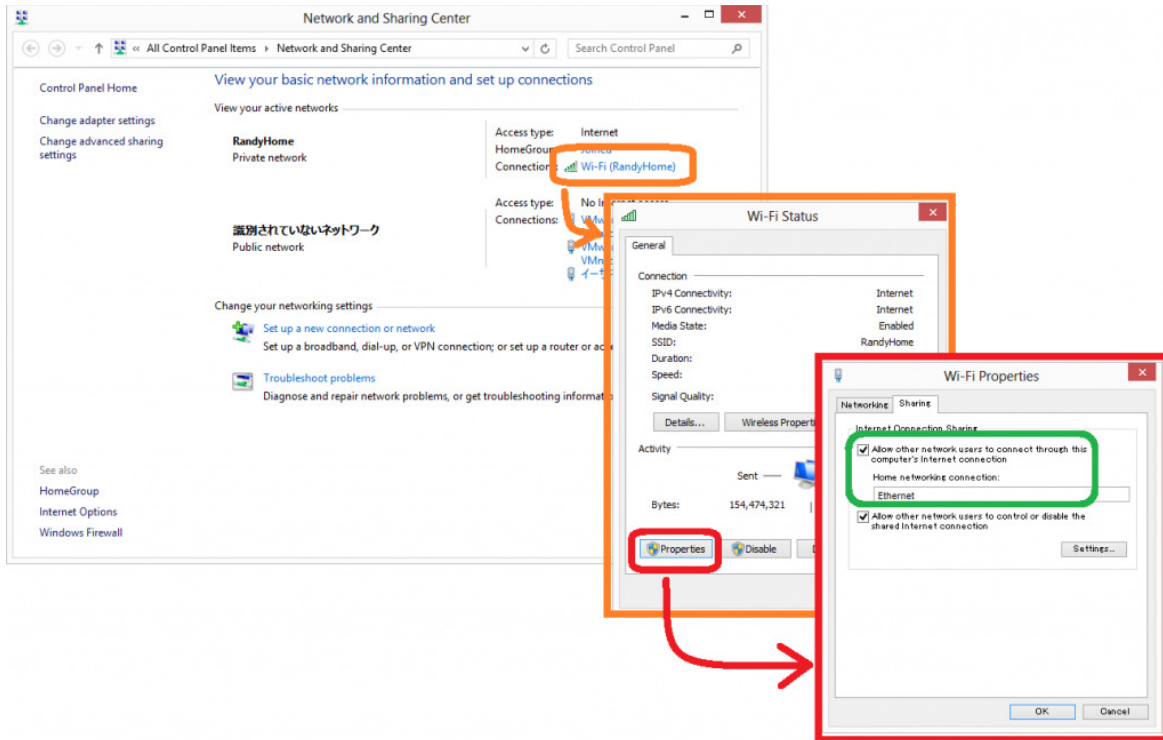
Connect the Pixhawk's TELEM2 port to the RPi's Ground, TX and RX pins as shown in the image above. The red V+ cable can go to the RPi's +5v pin, which . More details on the individual RPi's pin functions can be found [here](#).

### Connecting to RPi with an SSH/Telnet client

Note: These steps assume that you have already set-up your RPi so that it is running Raspbian.

To avoid the requirement to plug a keyboard, mouse and HDMI screen into your RPi it is convenient to be able to connect from your Desktop/Laptop computer to your RPi using an SSH/Telnet client such as PuTTY.  
Step 1: connect the RPi to your local network by one of the following methods:

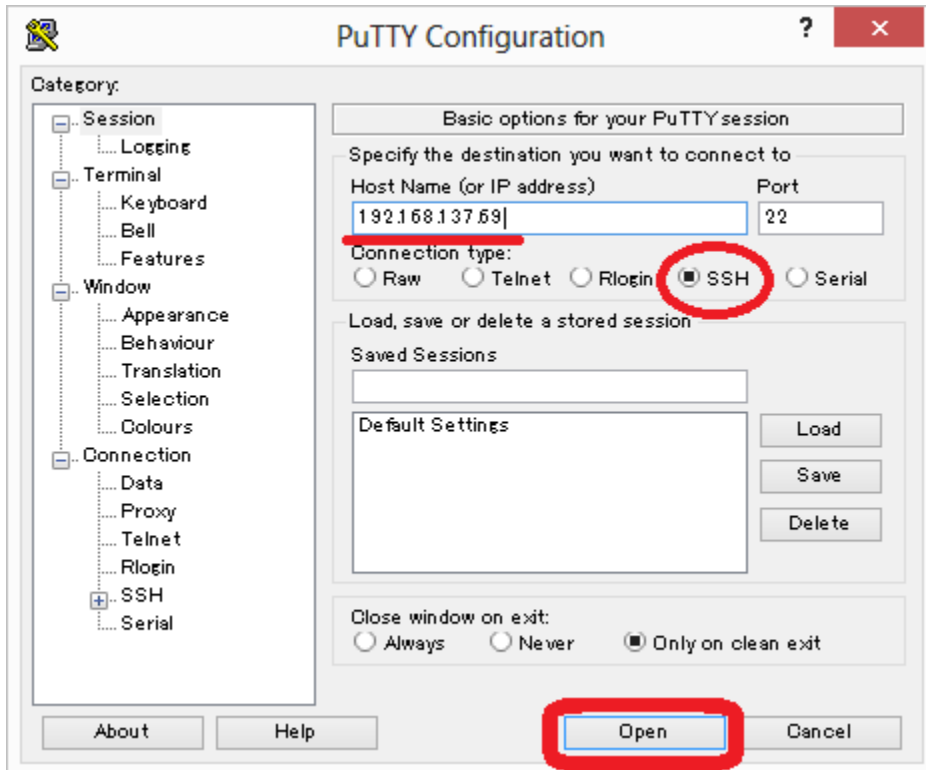
- a) connecting an Ethernet LAN cable from the RPi board to your Ethernet router
- or b) use a USB dongle to connect your RPi to the local wireless network
- or c) connect the ether net LAN cable between the RPi and your desktop/laptop computer. Open the control panel's Network and Sharing Center, click on the network connection through which your desktop/laptop is connected to the internet, select properties and then in the sharing tab, select "Allow other networks to connect through this computer's Internet connection"



Step 2: determine the RPi's IP address:

- a) if you have access to the RPi's terminal screen (i.e. you have a screen, keyboard, mouse connected) you can use the `/sbin/ifconfig` command.
- b) if your ethernet router has a web interface it may show you the IP address of all connected devices

Step 3: connect with Putty



If all goes well you should be presented with the regular login prompt to which you can enter the username/password which defaults to pi/raspberry

### Install the required packages on the Raspberry Pi

Log into the RPi board (default username password is pi/raspberry) and check that its connection to the internet is working:

ping google.com OR ping 173.194.126.196

If the first fails but the second succeeds then there is a problem with the DNS server that your RPi is attempting to use. Please edit the /etc/resolv.conf file and add the IP address of a nearby DNS server. During the creation of this wiki, the first two parts of the desktop machine's IP address plus ".1.1" worked. To stop other processes from later updating this file you may wish to run the `chattr +i /etc/resolv.conf` command (this can be undone later with `chattr -i /etc/resolv.conf`). That sets the "immutable" bit on resolv.conf to prevent other software from updating it.





Reboot the Raspberry Pi when you are done.

### Testing the connection

To test the RPi and Pixhawk are able to communicate with each other first ensure the RPi and Pixhawk are powered, then in a console on the RPi type:

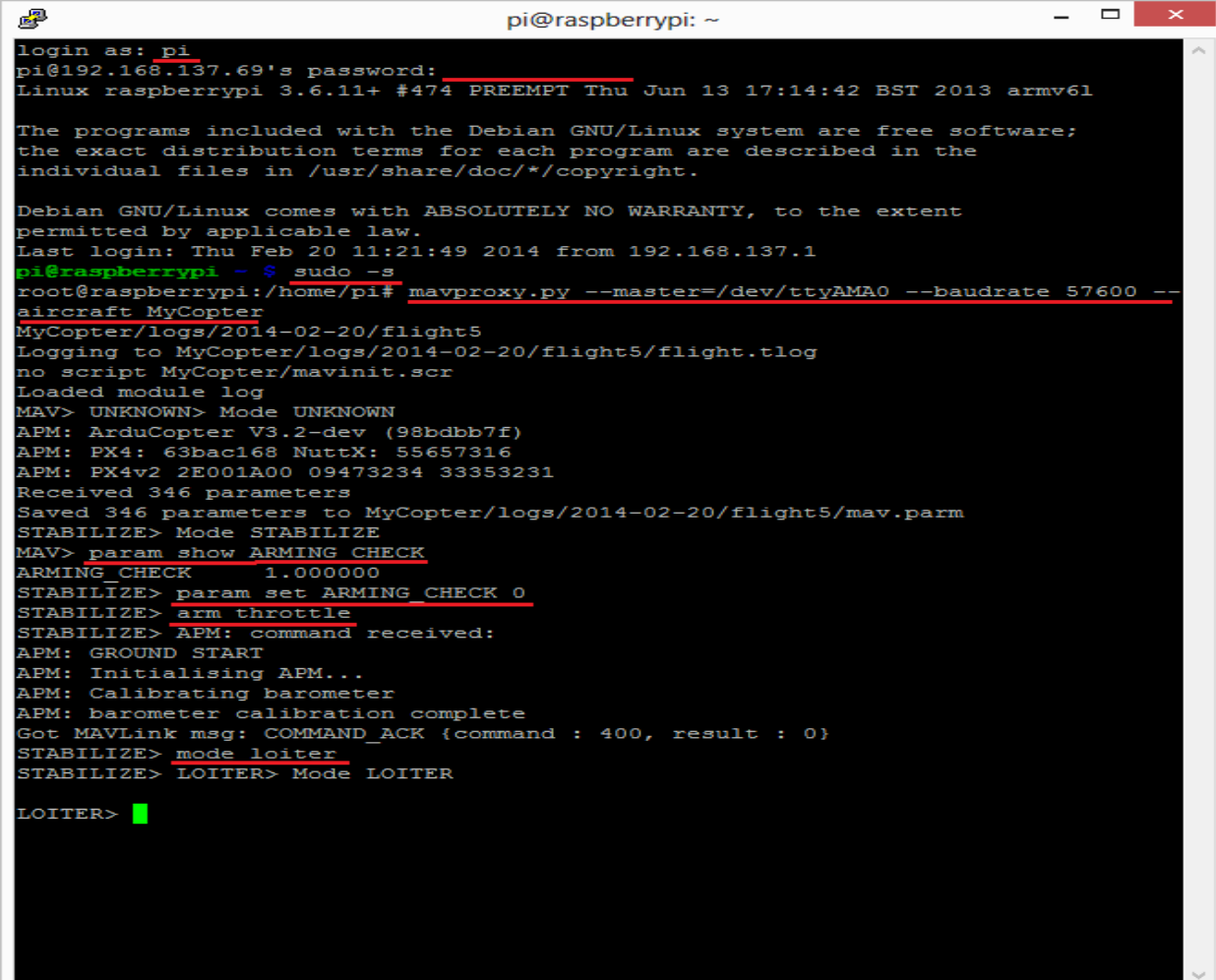
```
sudo -s
```

```
mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --aircraft MyCopter
```

once mavproxy has started you should be able to type in the following command to display the

ARMING\_CHECK parameters value param show ARMING\_CHECK

```
param set ARMING_CHECK 0
```



```
pi@raspberrypi: ~
login as: pi
pi@192.168.137.69's password:
Linux raspberrypi 3.6.11+ #474 PREEMPT Thu Jun 13 17:14:42 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 20 11:21:49 2014 from 192.168.137.1
pi@raspberrypi ~ $ sudo -s
root@raspberrypi:/home/pi# mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --
aircraft MyCopter
MyCopter/logs/2014-02-20/flight5
Logging to MyCopter/logs/2014-02-20/flight5/flight.tlog
no script MyCopter/mavinit.scr
Loaded module log
MAV> UNKNOWN> Mode UNKNOWN
APM: ArduCopter V3.2-dev (98bdbb7f)
APM: PX4: 63bac168 NuttX: 55657316
APM: PX4v2 2E001A00 09473234 33353231
Received 346 parameters
Saved 346 parameters to MyCopter/logs/2014-02-20/flight5/mav.parm
STABILIZE> Mode STABILIZE
MAV> param show ARMING CHECK
ARMING_CHECK 1.000000
STABILIZE> param set ARMING CHECK 0
STABILIZE> arm throttle
STABILIZE> APM: command received:
APM: GROUND START
APM: Initialising APM...
APM: Calibrating barometer
APM: barometer calibration complete
Got MAVLink msg: COMMAND_ACK {command : 400, result : 0}
STABILIZE> mode loiter
STABILIZE> LOITER> Mode LOITER

LOITER> █
```

To arm hold throttle Right until light change hear arming tone.

**Note:** if you get an error about not being able to find log files or if this example otherwise doesn't run properly, make sure that you haven't accidentally assigned these files to another username, such as Root. Entering the following at the Linux command line will ensure that all files belong to the standard Pi login account:

```
sudo chown -R pi /home/pi
```

**Configure mavproxy to always run ( you can skip this you do not need mavproxy to run on start)  
Mavproxy is used to control & issue commands to Pixy Hawk from the PI CLI consul.**

To setup mavproxy to start whenever the RPi is restarted open a terminal window and edit the `/etc/rc.local` file, adding the following lines just before the final "exit 0" line:

```
(  
date  
echo $PATH  
PATH=$PATH:/bin:/sbin:/usr/bin:/usr/local/bin  
export PATH  
cd /home/pi  
screen -d -m -s /bin/bash mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --aircraft MyCopter  
) > /tmp/rc.log 2>&1  
exit 0
```

Whenever the RPi connects to the Pixhawk, three files will be created in the `/home/pi/MyCopter/logs/YYYY-MM-DD` directory:

**mav.parm** : a text file with all the parameter values from the Pixhawk

**flight.tlog** : a telemetry log including the vehicles altitude, attitude, etc which can be opened using the mission planner (and a number of other tools)

**flight.tlog.raw** : all the data in the .tlog mentioned above plus any other serial data received from the Pixhawk which might include non-mavlink formatted messages like startup strings or debug output

If you wish to connect to the mavproxy application that has been automatically started you can log into the RPi and type:

```
screen -x
```

To learn more about using MAVProxy please read the [MAVProxy documentation](#).

It is also worth noting that MAVProxy can do a lot more than just provide access to your Pixhawk. By writing python extension modules for MAVProxy you can add sophisticated autonomous behaviour to your vehicle.

MAVProxy module has access to all of the sensor information that your Pixhawk has, and can control all aspects of the flight. To get started with MAVProxy modules please have a look at the [existing modules](#) in the MAVProxy source code.

Auto start mavproxy on boot download this file Go to the [DroneAPI Github file mavgateway](#)

```
nano mavgateway
```

To setup the correct parameters to start MavProxy find the following line:

```
DAEMON_ARGS="--master=/dev/ttyMFD1,115200 --out=udpin:0.0.0.0:14550 --daemon"
```

and change it to this:

```
DAEMON_ARGS="--master=/dev/ttyAMA0,57600 --out=udpin:0.0.0.0:14550 --daemon"
```

Next we need to change the user that starts MavProxy, find this line:

```
start-stop-daemon --start --background --make-pidfile --chuid edison --chdir /tmp --quiet --pidfile $PIDFILE --exec $DAEMON -- \
```

And change it to:

```
start-stop-daemon --start --background --make-pidfile --chuid pi --chdir /tmp --quiet --pidfile $PIDFILE --exec $DAEMON -- \
```

Save the file by typing in Control-X then Y then return

```
sudo chown root:root /etc/init.d/mavgateway
```

```
sudo chmod 755 /etc/init.d/mavgateway
```

```
update-rc.d /etc/init.d/mavgateway defaults
```

### **Connecting with the Mission Planner**

The Pixhawk will respond to mavlink commands received through Telemetry 1 and Telemetry 2 ports (see image at top of this page) meaning that both the RPi and the regular ground station (i.e. Mission planner, etc) can be connected. In addition it is possible to connect the Mission Planner to the mavproxy application running on the RPi [similar to how it is done for SITL](#).

Primarily this means adding an --out <ipaddress>:14550 to the mavproxy startup command with the being the address of the PC running the mission planner. On windows the ipconfig can be used to determine that IP address. On the computer used to write this wiki page the mavproxy command became:

```
mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --out 192.168.0.11:14550 --aircraft MyCopter
```



Connecting with the mission planner is shown below (you will use TCP IP of the Raspberry Pi & port 2000 )



### Preparing your Raspberry Pi for first boot

Follow the instructions at <http://www.raspberrypi.org/wp-content/uploads/2012/04/quick-start-g...>  
Install the prepared SD card in the Pi and boot.

### Setting up your Pi

Connect the Pi to your router with a network cable.

On Start-up it will resize the FAT partition and present you with a menu.

Set your language, and keyboard layout.

Select Raspbian... then click install.

After this has extracted (will take a while....) it will reboot into the configuration screen (again will take a while for this first boot.)

The important things to change here are

1. Enable the camera
2. In advance options
  1. Set the host name (camera, for the camera end, receiver, for the viewing end)
  2. Memory split, set the memory for the GPU to 256

3. Enable SSH ( will come in handy later, as you may need to talk to the Pi in the air.....

Then finish and reboot.

First login

*Username: pi*

*Password: superuser*

Setting up the required programs for video streaming

Install the dependencies by running the following in a terminal:

```
sudo apt-get install mplayer netcat
```

```
cd /opt/vc/src/hello_pi
```

```
make -C libs/ilclient
```

```
make -C libs/vgfont
```

```
cd /opt/vc/src/hello_pi/hello_video
```

```
make
```

```
cd ~
```

Now repeat this for the other Pi

Streaming

First set up the receiver

Ensure the receiver is connected to your network and run

```
ifconfig
```

after you press enter, you can find your ip Address. Note this down.

Then run the following.

```
mkfifo buffer
```

```
nc -p 5001 -l > buffer | /opt/vc/src/hello_pi/hello_video/hello_video.bin buffer
```

the Pi will now wait for the feed.

On the Camera Pi

Ensure camera is connected to the Pi

Ensure Pi is connected to the network (you can confirm this with ifconfig)

(see instructions at <http://www.raspberrypi.org/camera> for how to connect the camera)

In the following command, replace the ip address with the one you just noted down.

```
raspivid -t 0 -fps 15 -o - | nc 192.168.1.85 5001
```

if all goes well you should be streaming 1080P video at 15fps with less than 0.5seconds of delay..

now add your wireless bridge between the two, and away you go

---

## PI network:

SSH Login address # as : PI

Password: superuser

```
pi@QuadHDFPVBOT ~ $ sudo nano /etc/network/interfaces
```

```
cat /etc/network/interfaces
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
allow-hotplug eth0
```

```
iface eth0 inet manual
```

```
auto wlan0
```

```
allow-hotplug wlan0
```

```
iface wlan0 inet dhcp
```

```
    wpa-ssid "SSID"
```

```
    wpa-psk "password"
```

```
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

```
auto wlan1
```

```
allow-hotplug wlan1
```

```
iface wlan1 inet manual
```

```
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

```
sudo ifdown wlan0
```

```
sudo ifup wlan0
```

```
sudo apt-get install python-dev
```

```
sudo usermod -a -G tty pi
sudo usermod -a -G dialout pi
sleep infinity >/dev/ttyAMA0 &
```

### **TEST SERIAL CONECTION**

```
ps aux | grep ttyAMA0
screen /dev/ttyAMA0 57600 TO EXIT CTL + A K
```

**PROXY BETWEEN SERIAL AND NETWORK: COMPILE & RUN MAVLINK CONECTION DO NOT USE BOTH!**  
Compiling This is not practical slows down PI installing programming packages I used ser2net.

[https://github.com/mavlink/c\\_uart\\_interface\\_example](https://github.com/mavlink/c_uart_interface_example) [other Example Edison](#) or recommend INSALL.

Consideration camera not starting on boot can be caused by other program starting at the same time do not add any more /etc/init.d/ scripts that start at boot will kill the camera on boot use “crontab -e “ and nice command to auto start applications.

```
sudo apt-get install ser2net
```

```
sudo nano /etc/ser2net.conf
```

add to the bottom of file.

```
2000:raw:600:/dev/ttyAMA0:57600 8DATABITS NONE 1STOPBIT banner
```

```
sudo /etc/init.d/ser2net restart
```

to remove scrip from start up:

```
update-rc.d NAME_OF_INIT_SCRIPT remove
```

auto start /etc/init.d/ service

```
update-rc.d NAME_OF_INIT_SCRIPT defaults
```

**Removing the login prompt from serial interface:**

```
sudo nano /etc/inittab
```

Now go to the bottom of the file and look for the following lines:

```
#Spawn a getty on Raspberry Pi serial line
```

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

and comment out the line like this:

```
#Spawn a getty on Raspberry Pi serial line
```

```
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Save the file by typing in Control-X then Y then return

## ADDING THE NETWORK DETAILS TO THE RASPBERRY PI WPA

### **Static IP:**

```
iface eth0 inet static
address 192.168.0.12
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
```

Scan for SSID:

```
sudo iwlist wlan0 scan
```

WEP set up: Edit interfaces file

```
sudo nano /etc/network/interfaces
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
allow-hotplug eth0
```

```
iface eth0 inet dhcp
```

```
auto wlan0
```

```
allow-hotplug wlan0
```

```
iface wlan0 inet dhcp
```

```
    wireless-essid SSID
```

```
    wireless-key password
```

```
#wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

```
auto wlan1
```

```
allow-hotplug wlan1
```

```
iface wlan1 inet dhcp
```

```
    wpa-ssid SSID
```

```
wpa-psk password
```

```
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

#### **WPA:**

Open the wpa-supplciant configuration file in nano:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Go to the bottom of the file and add the following:

```
network={  
    ssid="The_ESSID_from_earlier"  
    psk="Your_wifi_password"  
}
```

In the case of the example network, we would enter:

```
network={  
    ssid="testing"  
    psk="testingPassword" }
```

Now save the file by pressing ctrl+x then y, then finally press enter.

At this point, wpa-supplciant will normally notice a change has occurred within a few seconds, and it will try and connect to the network. If it does not, either manually restart the interface with `sudo ifdown wlan0` and `sudo ifup wlan0`, or reboot your Raspberry Pi with `sudo reboot`.

You can verify if it has successfully connected using `ifconfig wlan0`. If the internet address field has an address beside it, the Pi has connected to the network. If not, check your password and ESSID are correct.

**Fixing WiFi Dropout Issues** If you find your module 'drops out' from time to time, you can fix it fairly easily with a command line fix (thanks perseus286 (<http://adafru.it/cL5>)!) Create and edit a new file in `/etc/modprobe.d/8192cu.conf` `sudo nano /etc/modprobe.d/8192cu.conf` and paste the following in `# Disable power saving options 8192cu rtw_power_mgnt=0 rtw_enusbss=1 rtw_ips_mode=1` Then reboot with `sudo reboot`

---

## **Connecting to copter using PI camera:**

Check to see if the camera is working <http://PI.IP.ADDRESS>.

**raspistill -v** Test the camera is up and running all streaming apps are installed, using raspistill module and VLC to display video on port 8554. Also 8081 8080 should work, recording can be done from the client end VLC. Http server is also running on PI you can access video there if IP changes edit ip on page.

**Run command:** /etc/default/motion turn on motion demon to edit config /etc/motion/motion.conf to start service sudo /etc/init.d/motion start (disable motion from starting at boot)

```
sudo apt-get install vlc
```

Start video stream run the following commands:

```
raspivid -o - -t 0 -n -w 720 -h 480 -fps 25 | cvlc -vvv stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554/}' :demux=h264
```

rtsp://\${RASPBEERRY\_PI\_IP:8554/} works in VLC or Quick time

“Motion -n” “motion”

```
raspivid -w 1080 -h 720 -fps 20 -b 10000000 -t 0 -o | gst-launch-1.0 -v fdsrc ! h264parse ! rtpH264payconfig-interval=10 pt=96 ! udpsink host=192.168.0.77 port=9000
```

```
360 -fps 25 | cvlc -vvv stream:///dev/video0 --sout '#rtp{sdp=rtsp://:8554/}' :demux=h264
```

```
raspivid -o - -t 0 -n -w 1080 -h 720 -fps 25 | cvlc -vvv stream:///stdin/video0 --sout '#rtp{sdp=rtsp://:8554/}' :demux=h264
```

```
raspivid -n -o - -t 9999999 -rot 180 -w 720 -h 480 -b 1000000 -fps 25|tee /home/pi/video-`date +%Y%h%d-%H%M`|nc -l 443
```

This command to a script by using the following

```
sudo nano sudo nano raspistill.sh
```

```
sudo chmod +x myscript.sh
```

If you want to start the stream automatically you have to add the script to crontab. To make this work you need to make another script run by cron ( **crontab -e will not work for the PI to start video stream if you memory is not 128mb for video** ), try to edit /etc/rc.local file add your script there or create an /etc/init.d/ scrip with proper run level to enabe at boot run `sudo update-rc.d SERVICE-TO-START-AT-BOOT enable` (VLC can't be run as sudo so make sure you're in the right cron). sudo nano FPV1.sh:

```
#!/bin/bash
```

```
/path/to/ raspistill.sh
```

Then:

```
sudo chmod +x raspistill.sh
```

```
crontab -e
```

add to crontab file this will run the script file at boot

```
@reboot /home/pi/raspivid.sh
```

Or python script

```
@reboot python /home/pi/MyScript.py &
```

To watch the video stream, open VLC on a computer on the same network as the raspberry pi you are using for streaming. Press Media -> Open Network stream and paste the following in the field:

```
rtsp://[IP].[TO].[THE].[PI]:8554/
```

Take photo command: `raspistill -o cam.jpg raspistill -v -o image.jpg`

## BASIC USAGE OF RASPIVID

With the camera module [connected and enabled](#), record a video using the following command:

```
raspivid -o vid.h264
```

Remember to use `-hf` and `-vf` to flip the image if required, like with [raspistill](#)

This will save a 5 second video file to the path given here as `vid.h264` (default length of time).

## SPECIFY LENGTH OF VIDEO

To specify the length of the video taken, pass in the `-t` flag with a number of milliseconds. For example:

```
raspivid -o video.h264 -t 10000
```

This will record 10 seconds of video.

## MORE OPTIONS

For a full list of possible options, run `raspivid` with no arguments, or pipe this command through `less` and scroll through:

```
raspivid 2>&1 | less
```

Use the arrow keys to scroll and type `q` to exit.

## FULL DOCUMENTATION

Full documentation of the camera can be found at [hardware/camera](#).

`raspiyuv` has the same set of features as `raspistill` but instead of outputting standard image files such as `.jpgs`, it generates raw unprocessed image files from the camera.



## [FULL DOCUMENTATION](#)

Full documentation of the camera can be found at [hardware/camera streaming from camera](#).

---

## Mode of operation:

1. Discover IP address of your Raspberry pi (`arp -a`) (192.168.42.21-19-12) login by ssh `user:pi password:superuser` recommend you set static ip on eth0 (but PI does not handle this well DHCP is more reliable) you will need to change steaming ip's at [http://PI.IP.Address](#) html code. Set static address by DHCP server, you can add the PI name & IP to you PC host file and enter the host name instead of IP address HTML code (`sudo nano /var/www/index.html` & `index1.html`) file location for web server. `sudo Screen -x test mavproxy` is running To exit GNU screen, type `Control-A k`.
2. Telnet to the pi if red light is off on the camera or camera crashes run camera script `./home/pi/raspivid.sh` recommend you stop broken service **first ! sudo /etc/init.d/raspivid stop** `sudo iwlist wlan0 scan` for wifi scan
3. join wireless network (scan for ssid ) by editing **sudo ifdown wlan0 sudo nano /etc/network/interfaces** enter the access point information for "wlan0" ssid & password if you are not using DHCP IP address **then save reboot/** set ip right away by **sudo ifconfig wlan0 IP-address** if hostapd is running you can edit **sudo nano /etc/hostapd/hostapd.conf**
4. Start stream from your pc enter in to VLC `rtsp://[IP].[TO].[THE].[PI]:8554/`
5. Connect to the PI on TCP port ip:2000 for serial connection to PIXHawk FC. Mavconn is not installed. If you use usb not recommended use TCP port 2001.
6. **Use command screen /dev/ttyAMA0 57600 to see if serial port is working device attached to serial port of the PI. USB port screen /dev/ttyACM0 57600**
7. Make sure you mavproxy scrip is running if you are using it `sudo /etc/init.d/mavproxy ststus` & pointing to the IP default 192.168.0.14-70 of you PC edit mavproxy script `sudo nano /etc/init.d/mavproxy ststus` start up script `sudo nano /etc/rc.local`
8. open up mission planner and select network TCP/IP port (2000) ip address of PI connect
9. To arm hold the throttle stick to the right or YAW right.
10. Save video or use GUI `vlc rtsp://192.168.0.12:8554/ --sout=file/ps:go.mp`  
Sudo rute -n then Network issues `sudo route add default gw 192.168.0.1`

---

## Frame Setup:

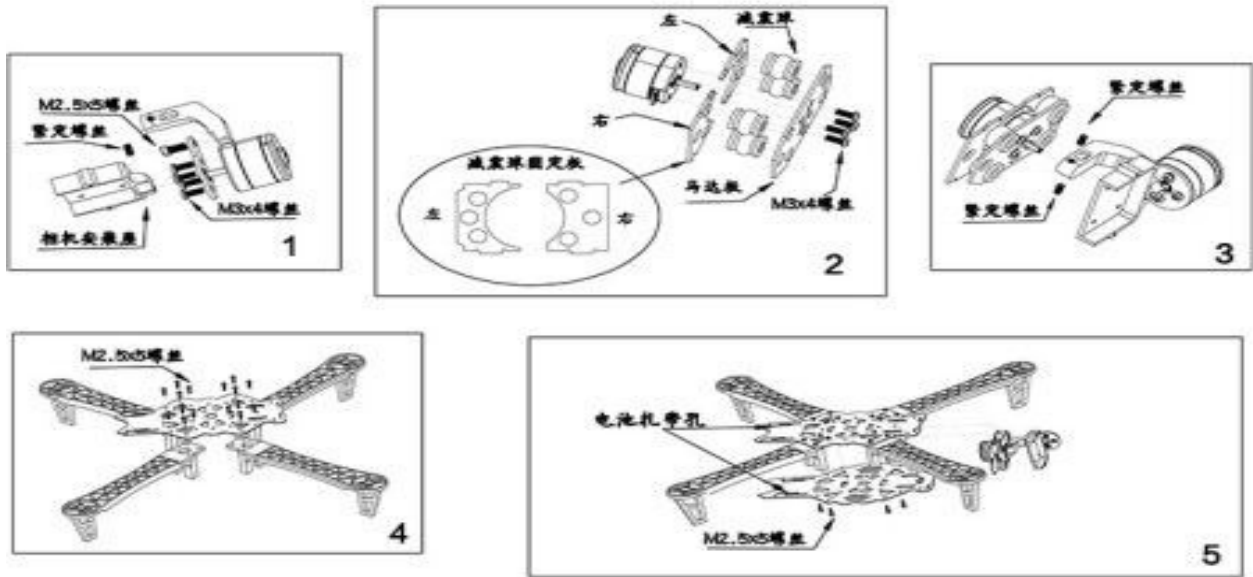




Assembly is very simple the CF plates are ambidextrous but the larger plate goes on the bottom the arm attach the bottom plate with 2 M3 bolts per arm the top plate attaches with 3 M3 bolts per arm. The gimbal assembly is straight forward visual below. See figure 4 & 5 below gimbal mount is figure 1-3 . RECOMMEND : please GPS on mass more away from the frame.



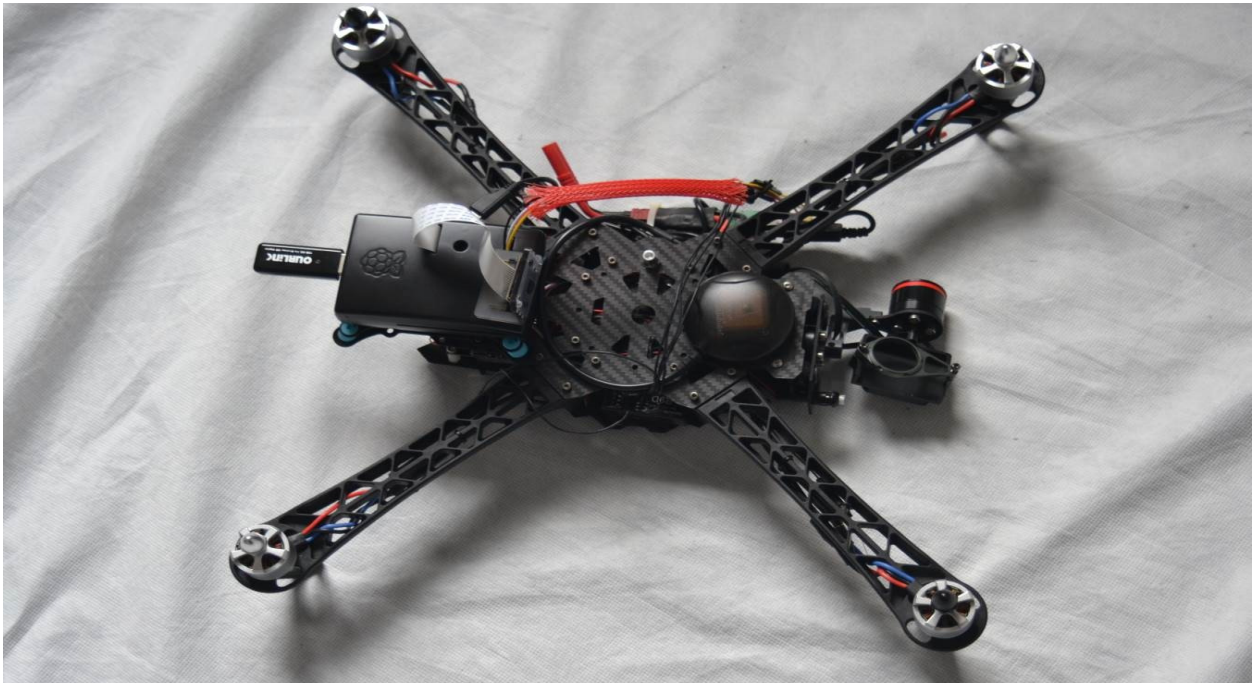
## 追风450机架装配图

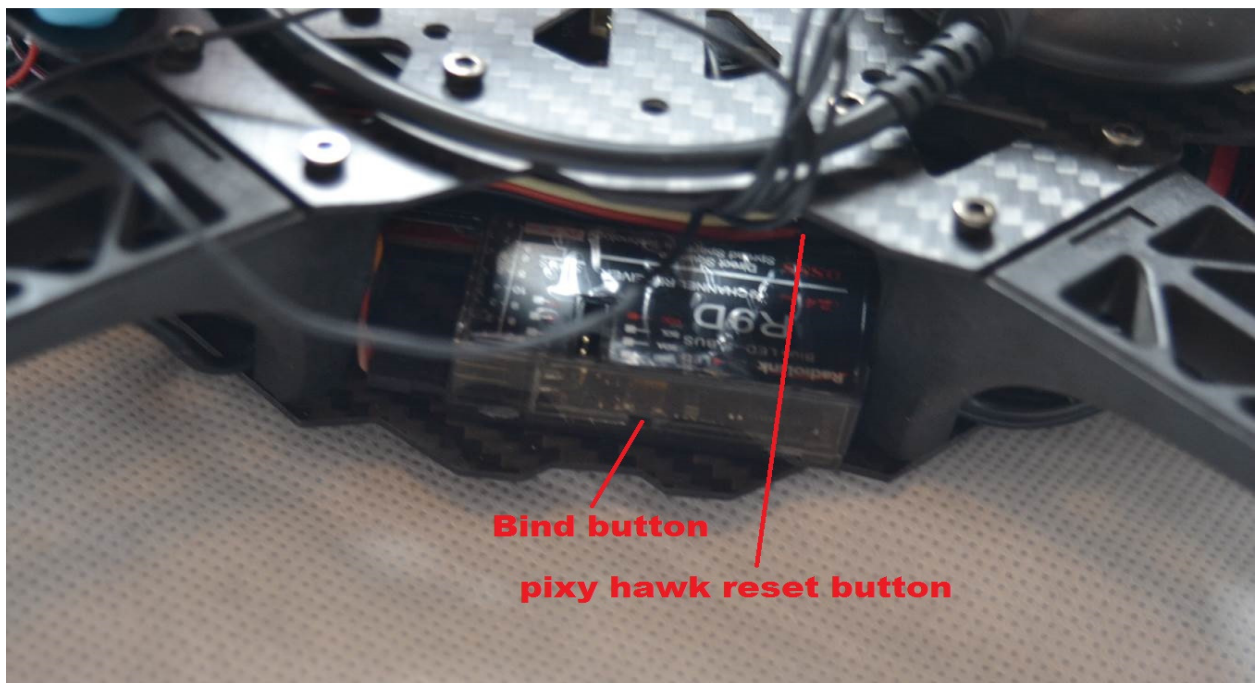
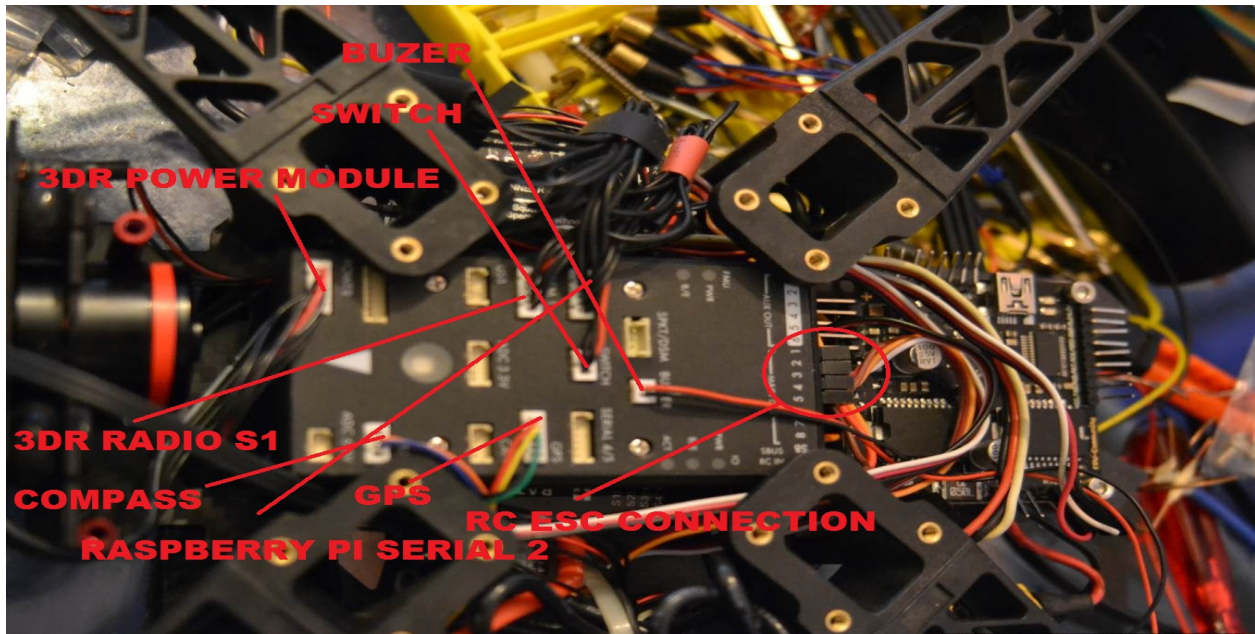


注：马达安装请使用M3x8内六角螺丝。

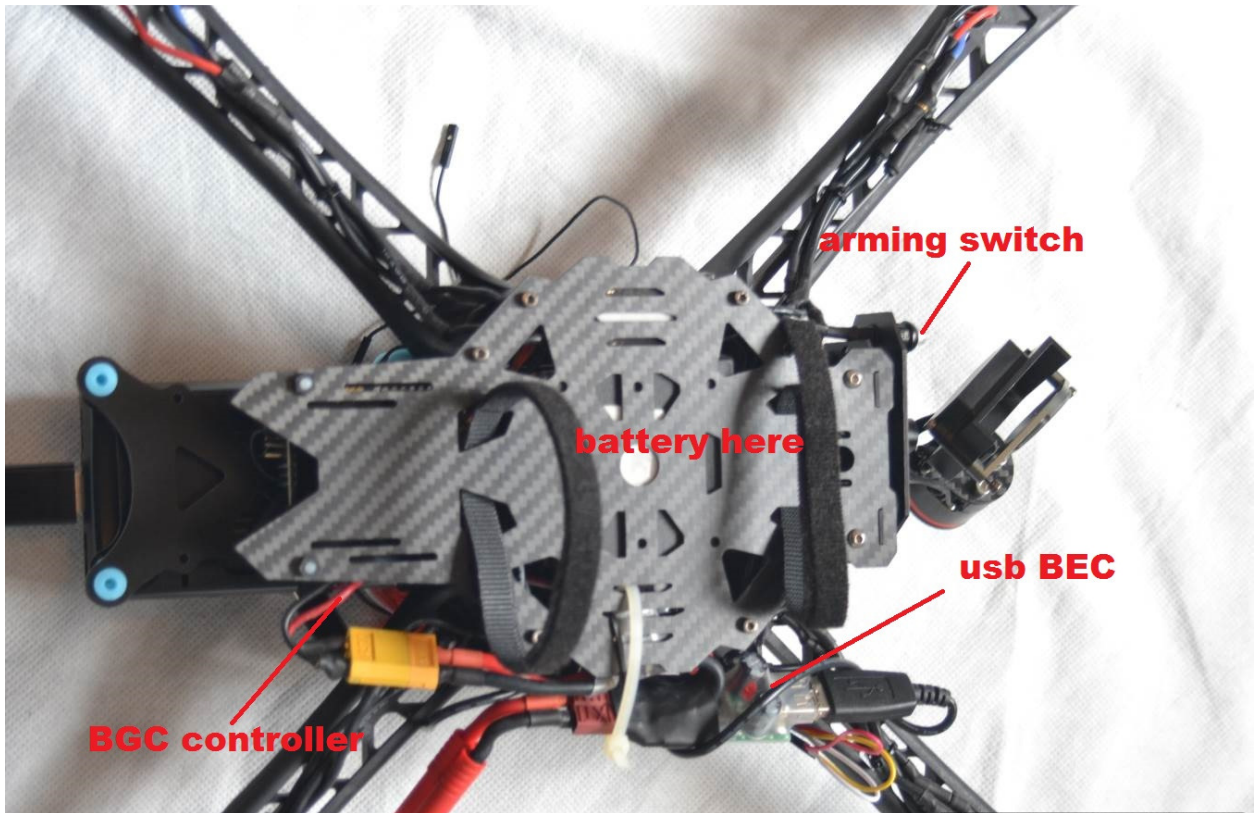
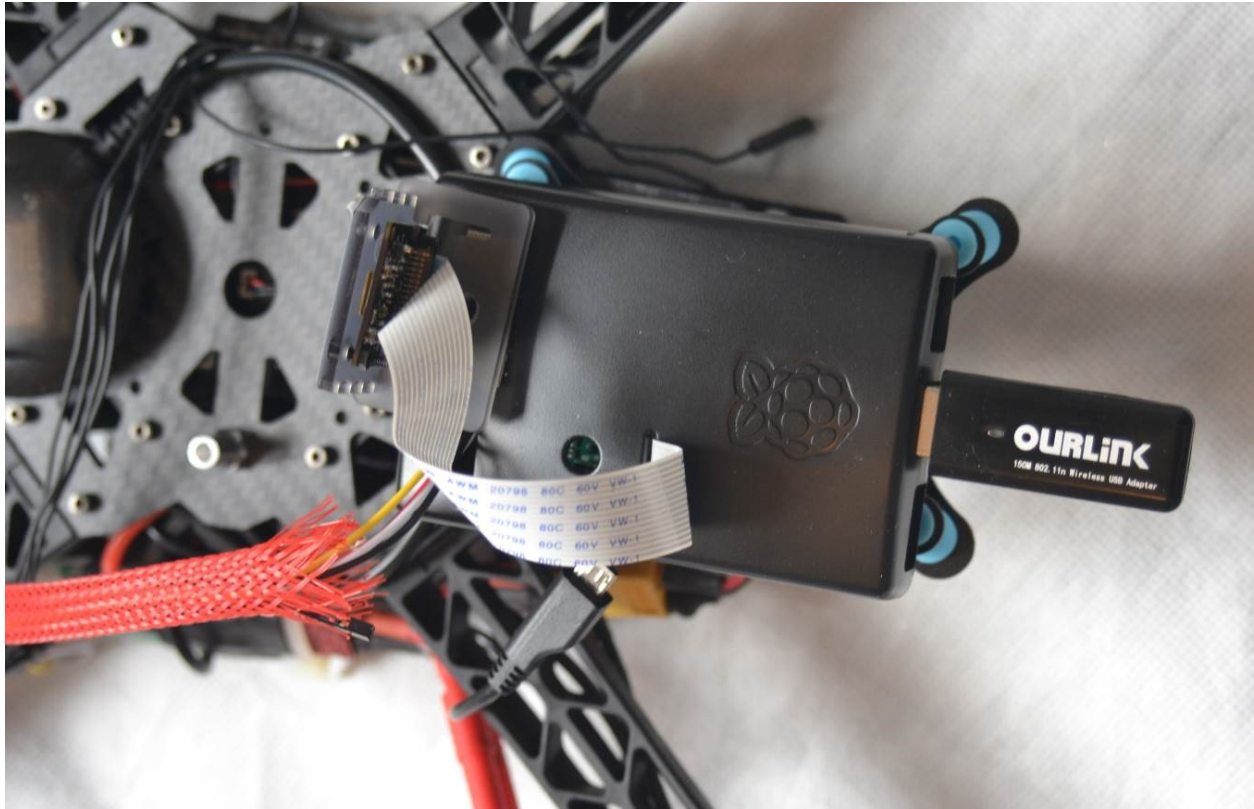
Attaching motor to the frame if there is too much vibration it is good practice to place rubber or foam under the motors as it is in this case to reduce vibration. Gasket can be made for each joint between arm, top and bottom plate if efforts are not enough. Additional standoff can also be used to make the frame more rigid as well. Please make sure no metal or wires are exposed note carbon fiber is conductive loose bare wire can cause fire or a bad short. Standard flaming wheel compatible arms can be purchased for \$15 any hobby store if broken. The motors a bit special because of capacity size is same as DJI phantom motor group with ability to handle up to 5S lipo safely. This makes easy to repair and very powerful please take care in it's operation when using supplied 4S battery. You need to assemble the parts and make sure they are in working order

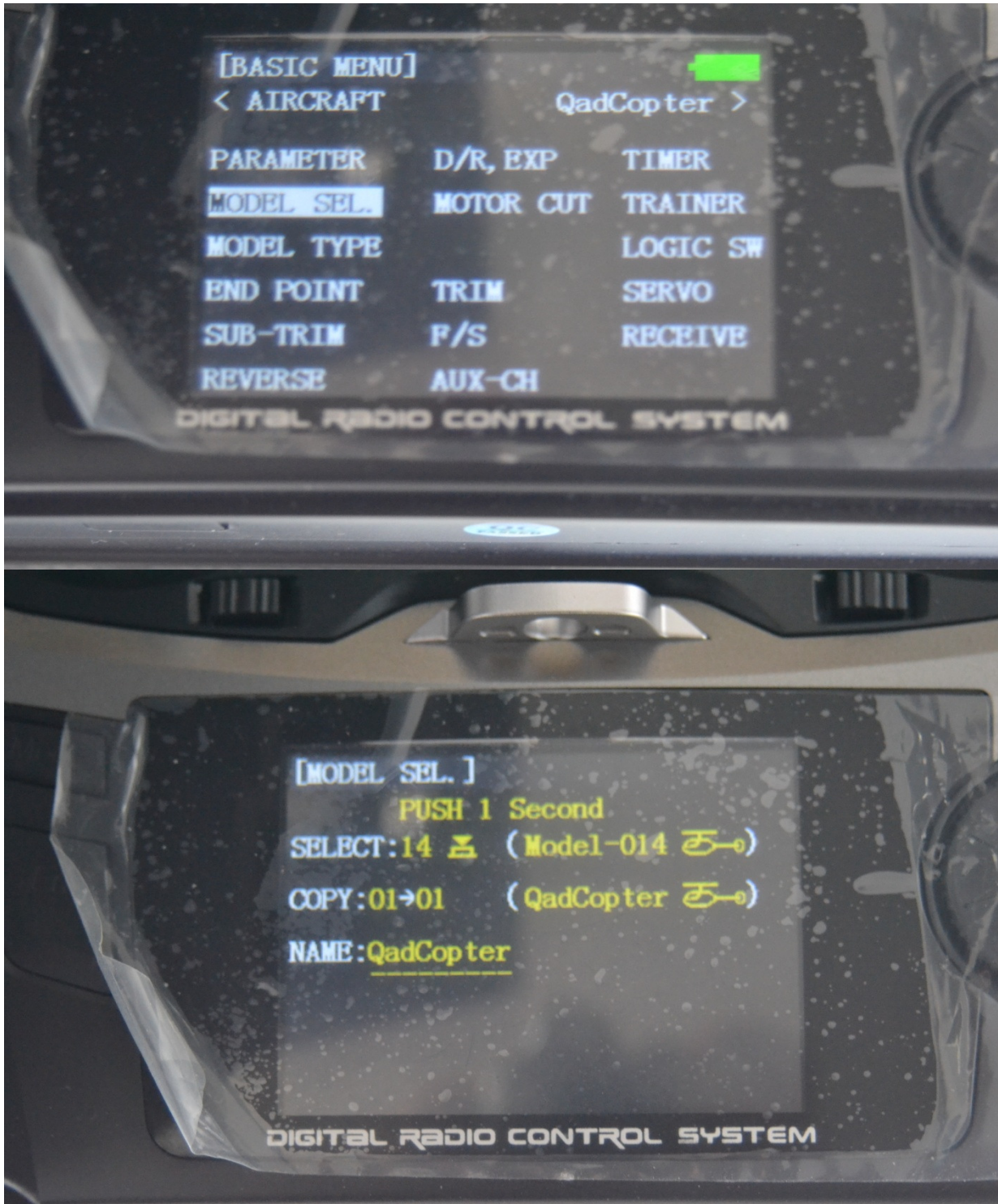
before attaching any electronics to the frame. All the parts that need to fit inside the frame.



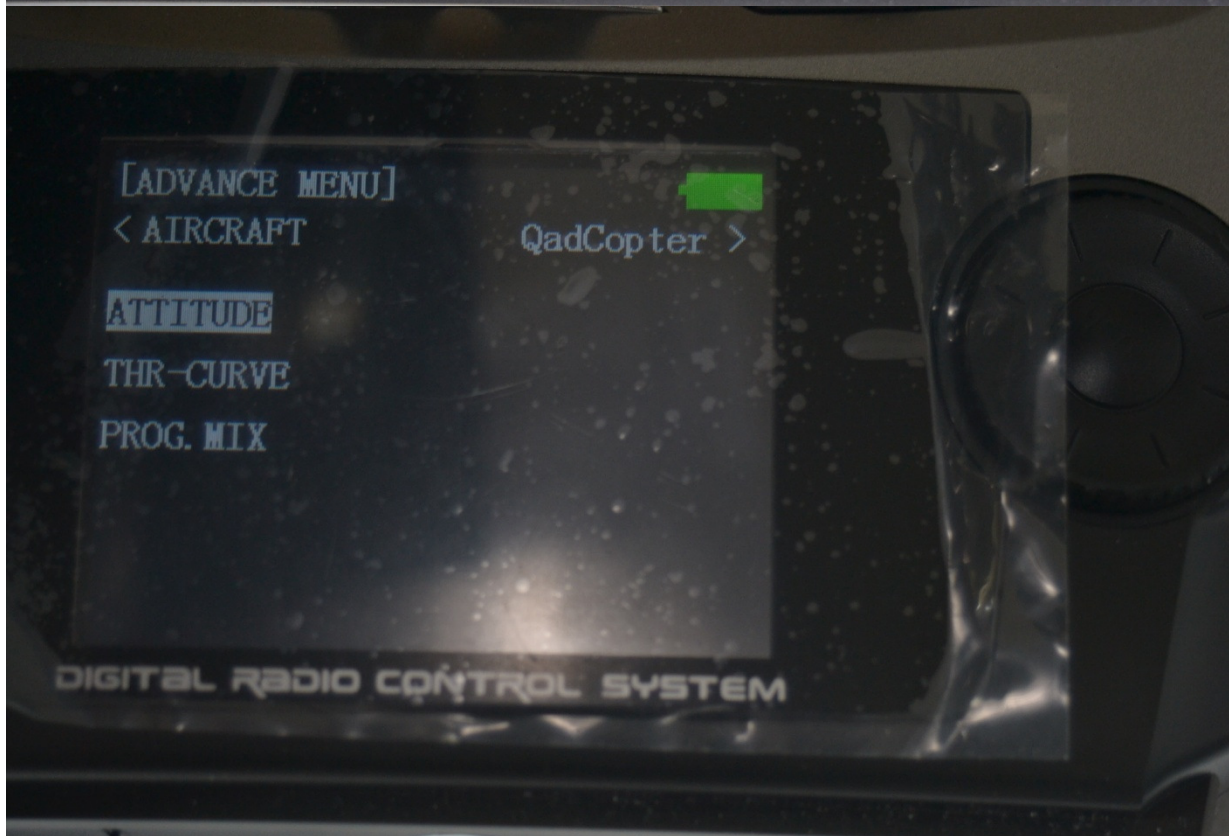
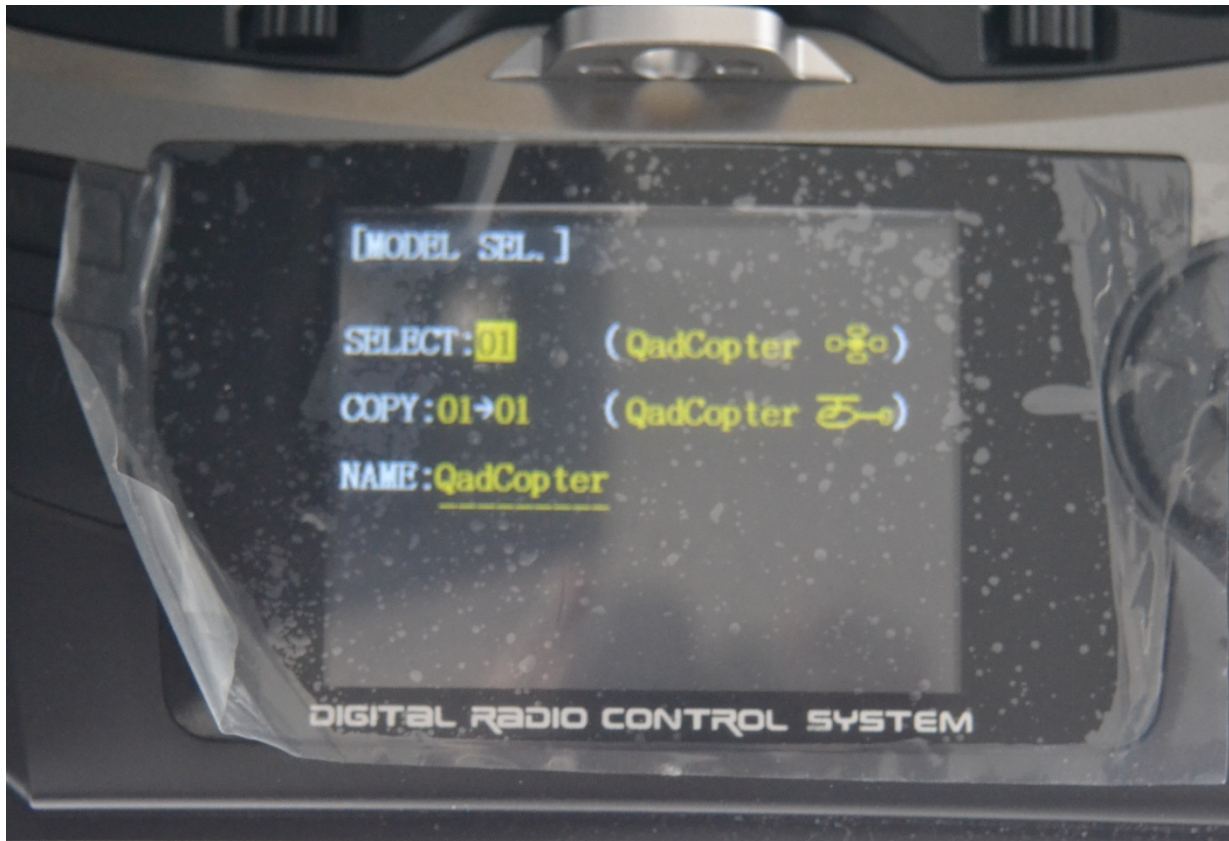


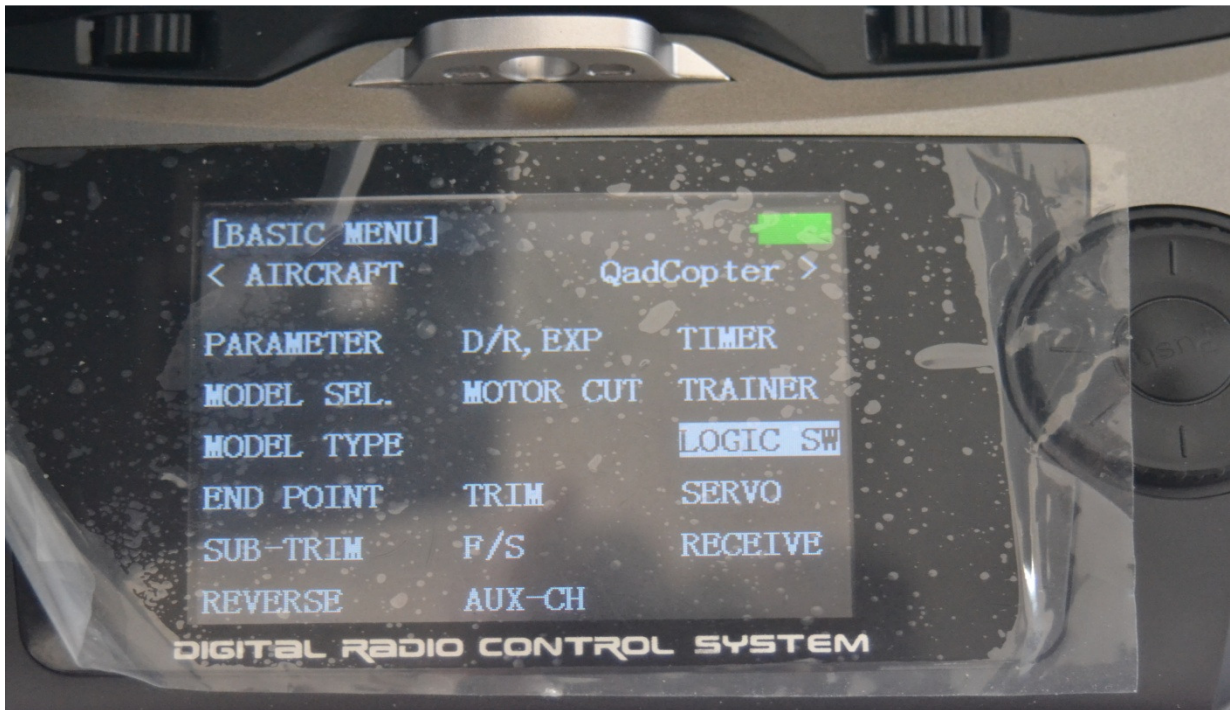
Tx Transmitter set up: change to multi copter then ,need to add switch to atti mode , them open up mode screen Mission planner adjust the endpoints until it lands on the mode you would like for the switch. You can only add 1, 3 position switch and 1, 2 position switch. You can assign a 3<sup>rd</sup> switch to RTL giving you 4 flighr modes. In this demonstration only three modes are set up.

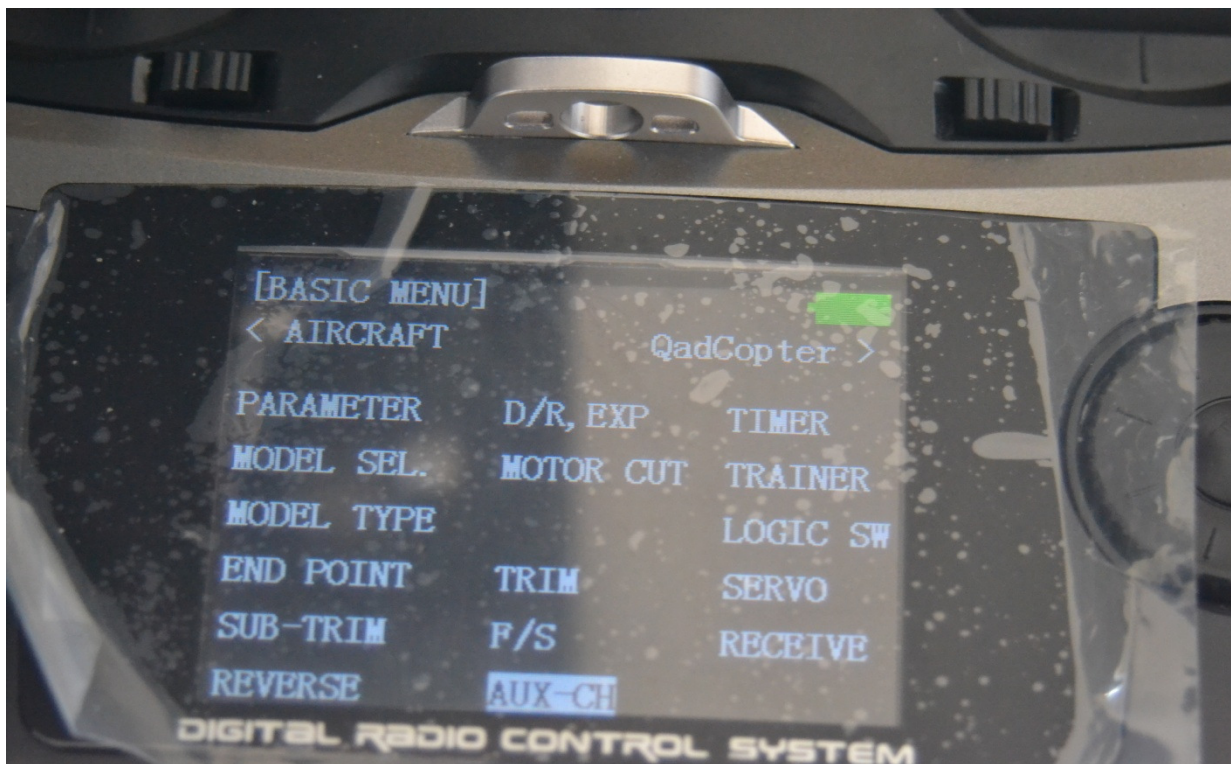


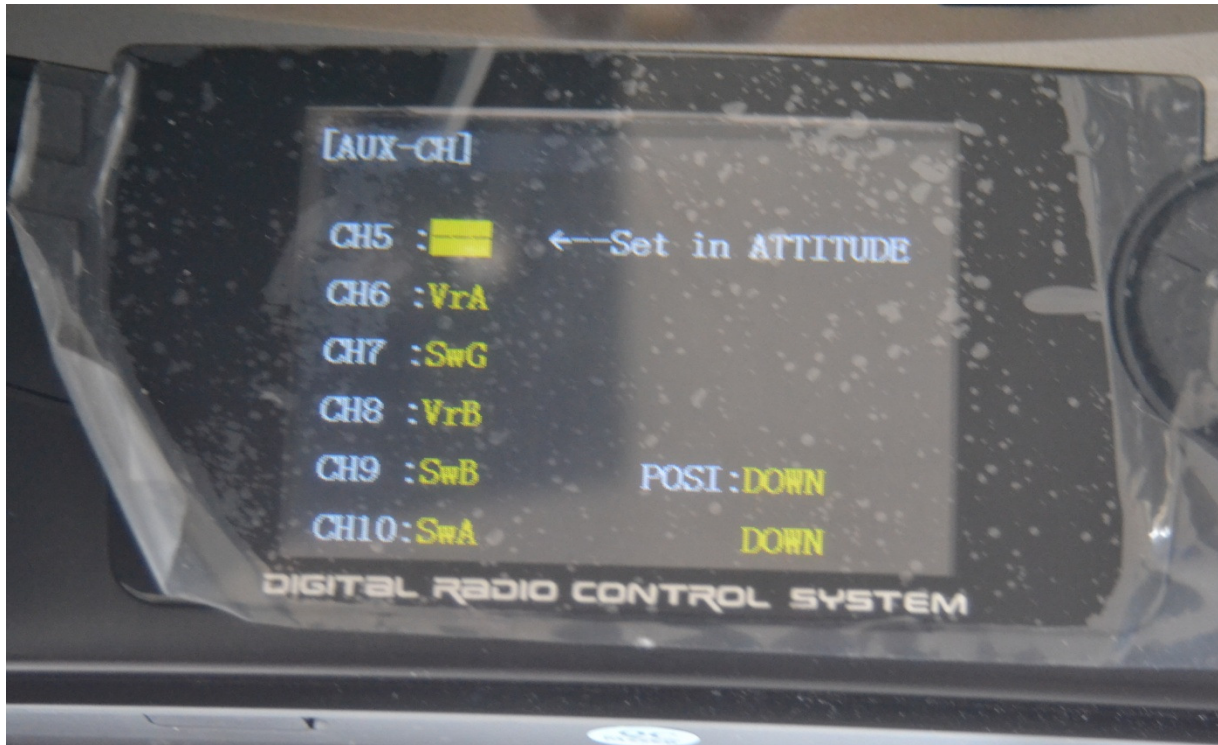


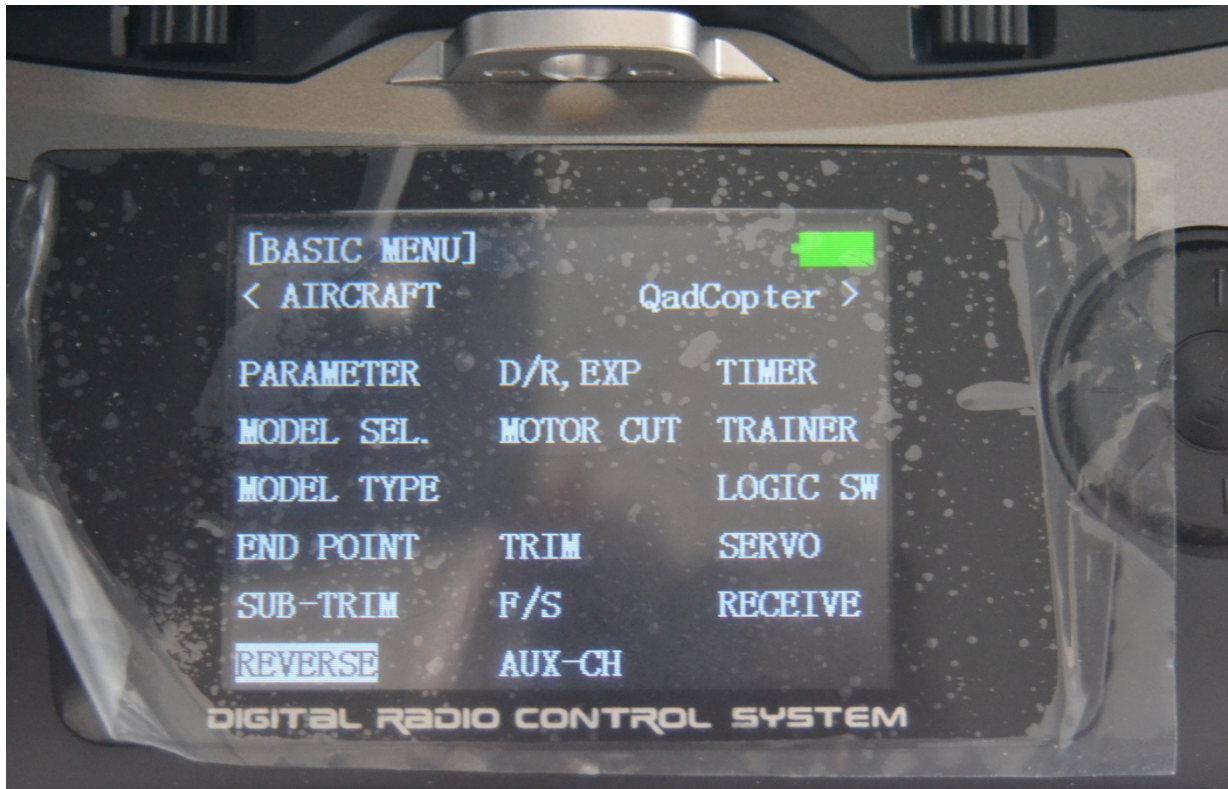


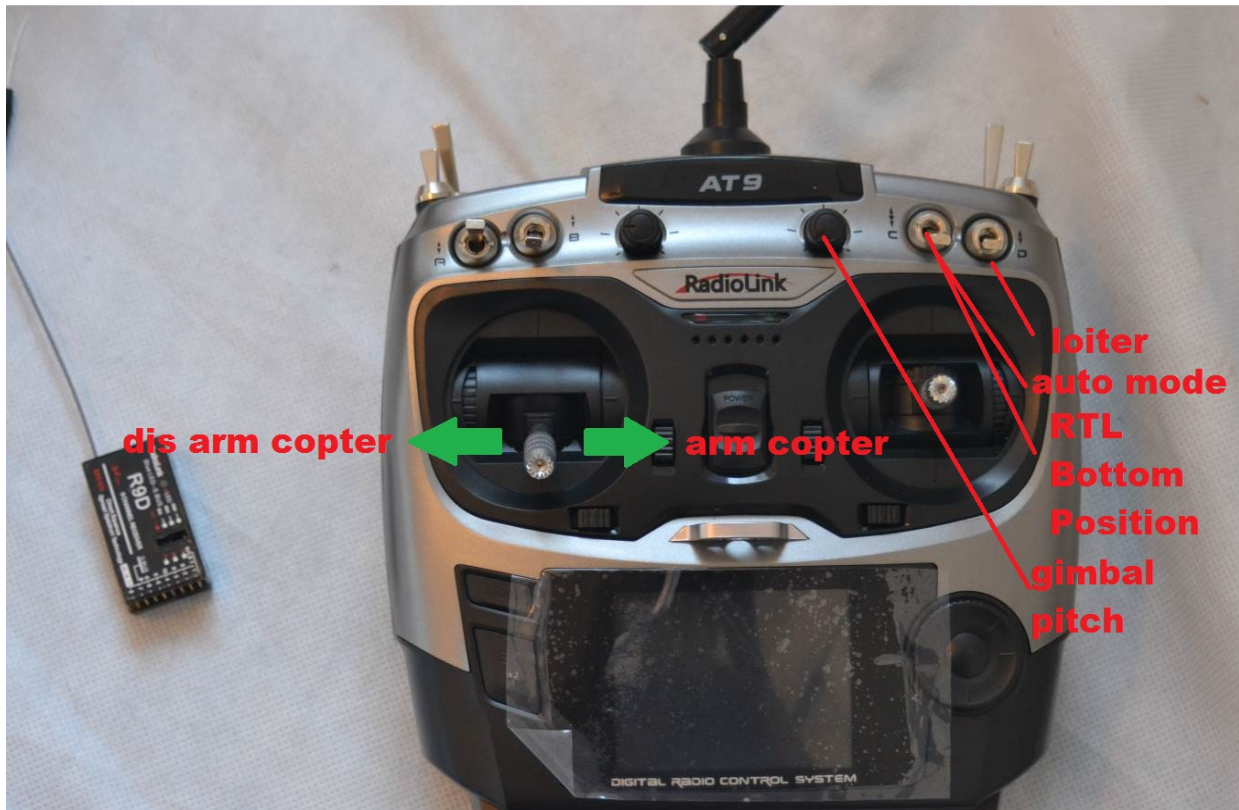












Press the mode button 1 times use wheel to scroll to model SEL press the wheel to select item END button to exit. Please select quad copters with select wheel, Exit button select mode button 1 times scroll down to model type select Air Craft quad copter. Now need to find an unused switch exit button select mode button 1 times scroll down to Logic SW. Now you need to edit Aux-CH Channel 5 mode switch already set up for DJI we need to do some minor changes to ATTITUDE, select CH 5. Now add switch 2 you found free switch to SW2 you can only use one 3 position switch SW3 one 2 position switch SW2. While in the ATTI screen open up mission planner, then navigate to mode setup screen. Note move the new switch to the active position. The pois on the screen will show up (ON). Navigate to the switch and change the rate value up or down until it triggers desired mode. This can be done for each switch in this screen. Channel 6 left pot, channel 8 right pot, channel 7 3 position switch G, channel 5 switch C, D. Open up mission planner scroll to first time setup radio calibration, move radio controls. If you're using S bus you may find some of your channels are reversed Press the mode button 1 times use wheel to scroll to reverse, reverses any channel that needs it Elevator & throttle in most cases.