A step-by-step guidance to build a drone from scratch using Ardupilot APM Navio2 Flight controller

Prepared by

Sim Kai Sheng
Yew Chang Chern

Table of Contents

Step-by-step guidance to build a drone from scratch using Ardupilot APM Navio2 Flight controller

Ardupilot is a widely used open source unmanned vehicle autopilot software that is capable of performing many functions. Documentations and various sources have provided us with the basic knowledge of the setups and use each separate component of a drone but none has provided a detailed guide on how to put them together to build the drone's hardware with guided steps of component setup and assembly from scratch, together with all necessary configuration and setups.

This article is written to combine all scattered information into one piece, to provide a step-by-step guidance from head to tail of how to build and assemble a quadcopter from scratch and how to perform all setup configurations required using the Navio2 Ardupilot flight controller. Other flight controllers from APM or others can also partly rely on this guide.

First let us select and purchase all the components that are required. The components can be purchased from different sites or sources but a basic requirement guide for all components will be provided as shown. A summarised list of the required components and the links to purchase each of the components are as shown in the PDF file named Drone BOM List (All pricing are converted into Malaysia Ringgit MYR or RM). Those components are the low-cost, high quality components that I have searched through the internet. Most of the components are from Chinese companies. I have tested them I believe they have similar qualities with their western counterparts but at much lower prices. However, you need to be extra careful when you want to purchase components from Alibaba because strangely cheap components from untrusted sites may have compromised qualities.

Full Component List

Purchase the following components online from links in PDF file named 'drone BOM list' attached below. Components from other sites can be used but bear in mind that the rule of thumb of buying components from Alibaba is to avoid buying components that are cheap. The more you pay, the better the quality you will get.

1.1 Navio2 Emlid Flight Controller

The main function of the Flight Controller is to provide control for the Electronic Speed Controller (ESC) to direct the rpm of motors based on inputs from the transmitter.

Navio2 Emlid Flight Controller

One of the advantages of using the Navio2 flight controller is because it has most of the components built-in so we do not need to buy them separately and integrate them by ourselves. The components that comes together in the Navio2 flight controller include

1. Dual Inertial Measurement Unit (IMU)
2. Barometers measure altitude accurate to 10 cm of resolution.
3. GNSS receiver tracks satellites from all over the world.
4. Compass or magnetometer determines the direction of heading
5. Extension ports allocate for additional sensors and radio telemetry.
6. Radio Communication co-processors accept PPM or SBUS inputs from receiver, and output 14 PWM channels for motors and servos at the servo rails

The link to the official website to purchase Navio2 Flight Controller is as follows https://emlid.com/navio/

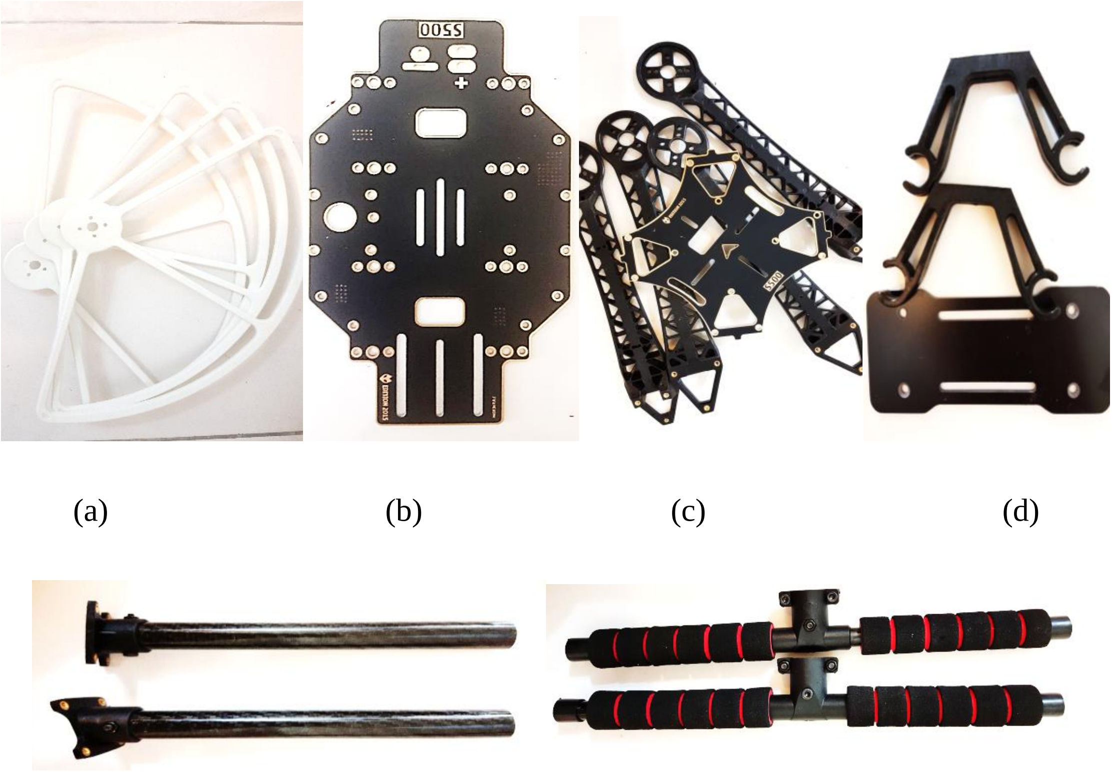
1.2 Raspberry Pi3 Model B

Raspberry Pi is a micro-computer at the size of an average credit card for data processing. With the General Purpose Input Output (GPIO) pins, the Raspberry Pi has high processing capability to run functions like flying a drone. Of the 40 pins of extended GPIOs, Navio2 has utilised 37 of them for flight control, leaving only 3 free GPIOs for additional components, that is GPIO 17 (Pin 11), GPIO18 (Pin 12), and GPIO 26 (Pin 37).

1.3 Airframe

The airframe makes up the body of the quadcopter with arms, landing gears, mounts, power distribution boards and blade protectors. The X frame is chosen because of good symmetry, simpler design, more flexibility and suitable for abrupt tilting. The selected frame size has a frame of 485 mm. The arrangement of the airframe size has to match with those of the batteries, propellers, as well as motors, as shown in table

Frame Size/mm	*Prop size/in*	*Motor size*	*Motor speed/KV*	*Li-Po size/mAh*
120/smaller	3	1104-1105	4000+	80-800 1S/2S
150-160	3-4	1306-1407	3000+	600-900 2S/3S
180	4	1806-2204	2600+	1000-1300 3S/4S
210	5	2204-2206	2300-2700	1000-1300 3S/4S
250	6	2204-2208	2000-2300	1300-1800 3S/4S
330-350	7-8	2208-2212	1500-1600	2200-3200 3S/4S
450-500	9-12	2212-2216	800-1000	3300+ 3S-5S



(a) (b) (c) (d)

(e)

Quadcopter airframe setup (a) Propeller protectors, (b) Power Distribution Board, (c) Rotor arms and central cross arm, (d) Battery or gimbal holder, (e) Carbon-fibre landing gear

To tighten the screws connecting the different parts of the airframe, you will need Alan keys of suitable sizes. Assemble the airframe tightly using suitable sizes of Alan keys (Vibrations may cause the screws to fall apart), according to guides provided by your supplier.

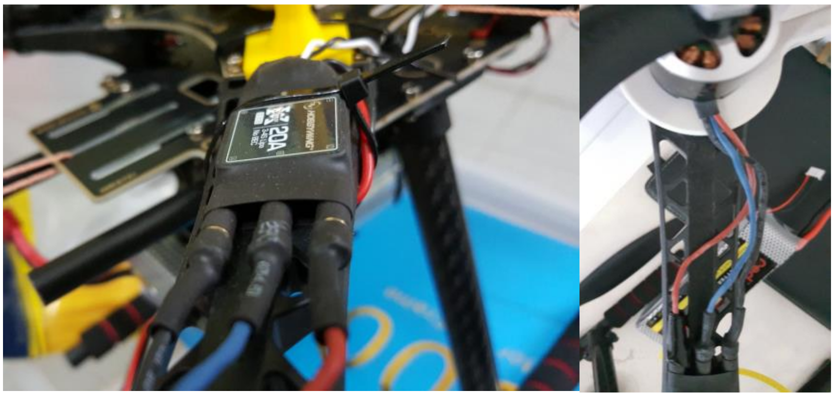
1.4 **Motors 2216/950KV**

The selection of the motor type depends on the size of the propellers or drone. A larger propeller should be powered by rotors of lower rpm because of the higher air resistance. A 950 KV motor means the motor has a 950 rpm of speed for each volt it is connected across. 2216 specifies the motor size.

It is recommended to buy a few spares of the motor caps. A motor cap covers and holds the propellers in place, and can falls off easily if the cap is not tightened enough (It is highly recommended to tighten it fully using a spanar). It can be troublesome to look for fallen motor caps especially in grassy areas. If the motor cap falls off, it usually will be somewhere near to your quadcopter. Look for it around and not far from the the drone.

Try to buy motors of different colours to differentiate the direction of rotations. 2 motors will be rotated clockwise and 2 motors anti-clockwise, depending on how you will connect it to the 3-phase ESC output, which will be discussed in the Hardware setup section later on.

1.5 Electronic Speed Controller (ESC)



Electronic Speed Controller (ESC)

An Electronic Speed Controller (ESC) draws current to drive the rotor based on output from the Flight Controller's servo rail. The ESC is graded with the maximum amount of current that it allows to pass through. Therefore, the selection of ESC is supposed to be such that the ESC rating is 1.2 to 1.5 times of the maximum rating of the rotor. For example, since the selected motor draws a maximum of 15 Amps (look for the maximum current draw from the motor datasheet or current rating specifications provided by the supplier) 20 Amps of ESC for each rotor would suffice without causing any overheating or burnouts. According to equation 2.1,

$$\frac{ESC\ rating}{Maximum\ rotor\ rating} = \frac{20A}{18A} = 1.333$$

It is best that the above equation yields values within the range of 1.2 to 1.5. And since the maximum current draw is rare, 20 A of ESC will suffice so long the ESC exceeds that of the motor current rating.

It is recommended to use ESCs without a Battery Eliminator Circuit (BEC). The navio2 flight controller can provide power supply to the ESC circuit through the servo rail. BECs are therefore not required.

1.6 Propellers

The propeller has to be selected based on the intended reliability and the sized according to the motor speed. Carbon fibre propellers are of higher quality but may not be cost-worthy. After all, carbon fibre propellers also will break under similar circumstances of impact. A softer propellers, on the other hand, may even be more flexible and less fragile. They are also less stiffer in resisting air, thus may be better at reducing possibilities of motor heating in large drones. With relatively large airframe of 485 mm (580 mm including the propeller protectors), the rotor speed selected is lower at 950 KV, and the propeller size is correspondingly larger at 12 inches or 30.48 cm. Propellers are balanced on a propeller balancer using additional elecrical tapes to make sure both blades are of equal mass. This minimises vibrations that could arise from unbalanced propellers. Mount the propellers on the propeller balancer, turn off your fan, and then tape the end of a lighter propeller blade until balance. You can shift the tape across the length of the blade to find the spot for perfect balancing.

Balanced propellers

1.7 GNSS receiver with antenna

Figure 0.1: GNSS receiver and antenna

A Global Navigational Satellite System (GNSS) receiver is the built-in GPS (and other satellites') receiver in Navio2. It requires an antenna that is raised high up away from the flight controller to avoid magnetic interferences generated from the components leading to GPS glitches or inaccuracy of satellite signal reception. A GPS stand is where the antenna is being mounted to, while the other end connects to the flight controller's GNSS receiver through a micro coaxial (MCX) connector. The micro co-axial connector will come with the GPS antenna set purchased, but the GPS stand has to be purchased separately, or 3D-printed online. It is recommended to have it purchased online because those are designed to be bent when you want to place your drone upside-down during calibration.

1.8 Transmitter

Radio transmitter

A high grade transmitter is what required to enable good control of a flying vehicle. This is crucial because otherwise, fatal injuries could be the result following an impaired communication. The Radiolink AT10II series is by far the most advanced transmitter the company has developed, offering hybrid of Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) radio communication protocol with the receiver, which then communicates with the flight controller via Serial Bus (SBUS) single communication cable. The transmitter can be powered with a small 2200 mAh Li-Po battery provided by the supplier to save battery replacement cost. The transmitter will also come with the receiver in a pair.

1.9 Receiver

Figure 0.2: Receiver

It is highly recommended to use a transceiver pair whose receiver can communicate with Navio2 through SBUS communication protocol. The R12DS receiver I used receives signals from the transmitter and then relay the signal to the flight controller via SBUS communication protocol. SBUS is a new single-wire communication protocol (saves a lot of messy wirings) that is supported by Navio2. One signal wire transmit all radio values of all 12 channels from the receiver (which is in turn transmitted wirelessly from the transmitter) to the flight controller. Navio2 has built-in decoder to decode SBUS RC input values before passing them to the Ardupilot. To connect the receiver to the flight controller, you will need three female-to-female jumpers to connect the signal, positive and negative terminals of the SBUS line from the reveiver to the respective signal, positive and negative terminals in the servo rail of the Navio2 flight controller. Use shorter jumpers for better tidiness.

1.10 **MicroSD card**

Similar to a computer that requires Hard Disk Drive (HDD) or the latest Solid State Drive (SSD) as storage, the Raspberry Pi uses a microSD card in which the operating system (Raspbian) is being installed. It contain crucial files that store system data such as connection IP addresses, file directories with codings for different features, etc.

1.11 **Lithium-Polymer (Li-Po) Battery**

Lithium battery has one of the highest power-to-weight ratio, making it one of the best choices for powering UAVs. However, Li-Po battery has to be taken care of thoroughly and have its voltage monitored so that each cell stays at least 3V (3.4V to be safe) before it is being recharged.

The Li-Po battery used is a 5200 mAh battery with 25 C rating. To determine if the discharge limit, C is sufficient for the motors used, the maximum discharge current is calculated from.

$$I_{max} = C \times Ah = 25 \times 5.2 = 130A$$

This selection is in line with the ESC and battery size, with each ESC rating being 20 A, and a maximum of 80 A (for quadcopter with 4 motors) will be drawn from the battery at a time. Occasions where full throttle are applied are rare and therefore 130 A from a 25 C battery is more than enough.

Li-Po battery size selection has to be made by taking into account the size of the copter for optimum flying duration. If a large drone is powered by relatively smaller sized battery, the flight time will be limited. If a small drone is powered by large batteries out of the optimum range, the battery drains even faster due to the extra weight allocated. The selection of Li-Po battery are best to be as according to the table to achieve maximum available flight time.

A T-type connector battery is used. If this type of connector is selected, the Li-Po charger and the power module's type T connector head should also be selected accordingly. A professional Li-Po charger is recommended because it can monitor the charging voltage and current flow throughout the charging to avoid overcharging. It can also trigger alarms when fault occurs.

Li-Po battery size based on quadcopter size

Quadcopter size/mm	*Li-Po battery size/mAh*	*Number of cells/S*
Mini Quad	80-800	1 or 2
180	1000-1300	3 or 4
210	1000-1300	3 or 4
250	1300-1800	3
280-290	1500-3300	4
330-360	2200-3200	4
400	3200-3300	4
450	3300	4
500	3300-5000	4
540	5000-5200	4
550-750	5000-8000	4 or 5 or 6
800 or bigger	8000-30000	6

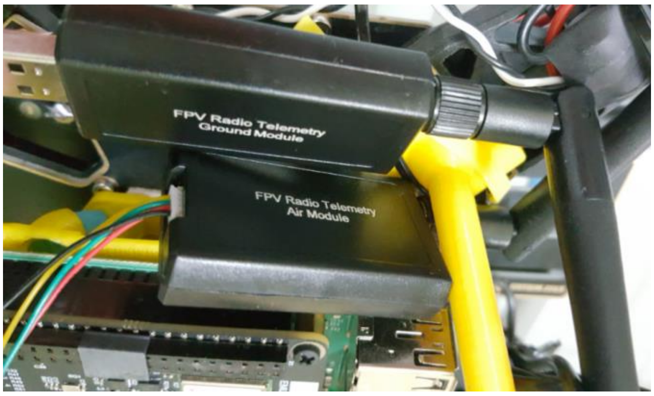
1.12 BB Alarm

To protect the Li-Po battery from possibilities of over-discharging, a BB alarm – a loud warning alarm when battery voltage of any cells of the Li-Po dropped below set values – is in place to monitor the battery voltage from time to time. As studied, if the battery is allowed to discharge under 3 V per cell, it can be permanently damaged and never be able to recharge back again.

1.13 Power Module

Because the same Li-Po battery supplies both the Raspberry Pi and the motors, a power module controls the voltage across the two according to their optimal requirements. A power module splits the battery current to the power distribution board (to the ESCs) and the power port in the navio2 flight controller. The power module should be carefully selected such that the connector matches that of the battery. A power module with the T-connector head is used if the battery purchased also has a T-connector.

1.14 Telemetry



Radio telemetry

A radio telemetry consists of a Universal Serial Bus (USB) ground module to be connected to the Ground Control Station (GCS) which is your laptop, and a Universal Asynchronous Receiver-Transmitter (UART) air module to be connected to the navio2 flight controller. The GCS's purpose is to plan flight path and monitor flight data. If the UART port's serial lines have to be used for other purposes such as for obstacle avoidance, or the GPIO 17 of the UART port is used for installation of turn-off switch, the Air Module of the telemetry can be interchanged with that of the Ground Module, with the Ground Module's USB head connected to the RPi3. The Air Module's UART requires an additional USB-TTL CP2012 converter to communicate with the GCS, or computer through the Silicon Labs CP210X USB-UART Bridge driver (can be downloaded online). The link to purchase CP2012 converter:

https://item.taobao.com/item.htm?id=528554518951

Thrust-to-weight ratio

The calculation of thrust-to-weight ratio of a quadcopter is essential to make sure the quadcopter can take off and fly stably with desired payloads.

Calculating the "unladen" weight of the quadcopter

Components	*No. of quantity*	*Mass per quantity (g)*	*Total mass (g)*
X Airframe	1	460	460
Battery 3S (11.1V)	1	420	420
Motor	4	52	208
Raspberry Pi	1	45	45
Navio2	1	23	23
ESC	4	14	56
Telemetry Air Module	1	12	12
BB Alarm	1	11	11
Propeller protector	4	11	44
Propeller	4	8	32
Power Module and cables	1	25	25
		Grand total	1336

The total unladen mass of the quadcopter without any payloads is around 1.336 kg. For the drone to be able to take off, the thrust that the rotors produce in total has to exceed that value, or twice of that.

$$Total\ thrust = 2 \times Total\ Weight\ of\ quadcopter$$

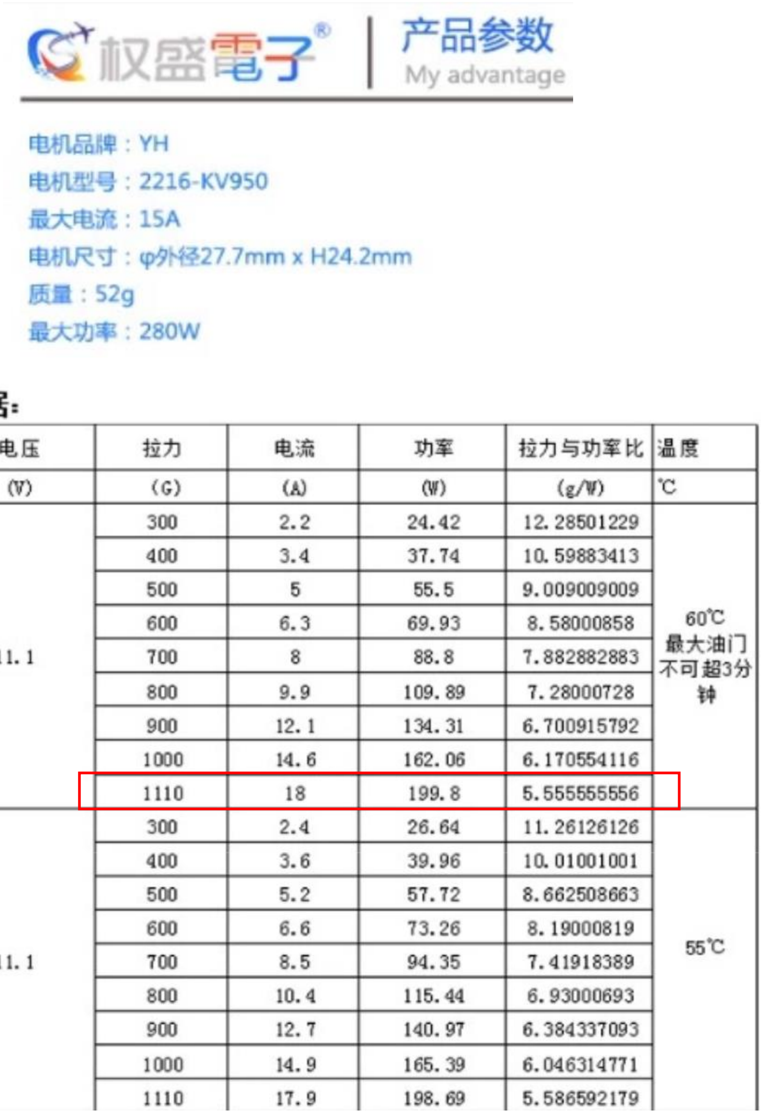
The generated thrust of a quadcopter can be calculated from the equation

$$m = \frac{T}{g} = \frac{\sqrt[3]{\frac{\pi}{2} D^2 \rho P^2}}{g}$$

Where,
T = Thrust (N)
D = Propeller diameter (m)
ρ = Density of air (1.225 kg/m^3)
P = Power of rotor (W)
m = Equivalent mass of thrust
g = Gravitational acceleration (9.81 m/s^2)

Since the calculation may be complicated, look from the datasheet provided by the rotors' supplier,

Datasheet of rotor provided by supplier



权盛電子® | 产品参数
My advantage

电机品牌：YH
电机型号：2216-KV950
最大电流：15A
电机尺寸：φ外径27.7mm x H24.2mm
质量：52g
最大功率：280W

螺旋桨测试数据：

螺旋桨规格	电压	拉力	电流	功率	拉力与功率比	温度
（inch)	(V)	(G)	(A)	(W)	(g/W)	℃
1238碳纤桨	11.1	300	2.2	24.42	12.28501229	60℃ 最大油门不可超3分钟
		400	3.4	37.74	10.59883413	
		500	5	55.5	9.009009009	
		600	6.3	69.93	8.58000858	
		700	8	88.8	7.882882883	
		800	9.9	109.89	7.28000728	
		900	12.1	134.31	6.700915792	
		1000	14.6	162.06	6.170554116	
		1110	18	199.8	5.555555556	
11*4.7	11.1	300	2.4	26.64	11.26126126	55℃
		400	3.6	39.96	10.01001001	
		500	5.2	57.72	8.662508663	
		600	6.6	73.26	8.19000819	
		700	8.5	94.35	7.41918389	
		800	10.4	115.44	6.93000693	
		900	12.7	140.97	6.384337093	
		1000	14.9	165.39	6.046314771	
		1110	17.9	198.69	5.586592179	

The thrust of each motor can also be obtained directly from the motor datasheet. As shown in the above table the equivalent mass of the thrust given is 1110 g for a maximum of 18 A current drawn to each of the rotor. From here, the total equivalent mass of the thrust can be calculated by simply multiplying the thrust by 4 rotors, giving $m = 4 \times 1.11 = 4.44\ kg$. The thrust to power ratio decreases as the power rises faster at higher throttle, with minimum of 5.55 g/W at the highest throttle.

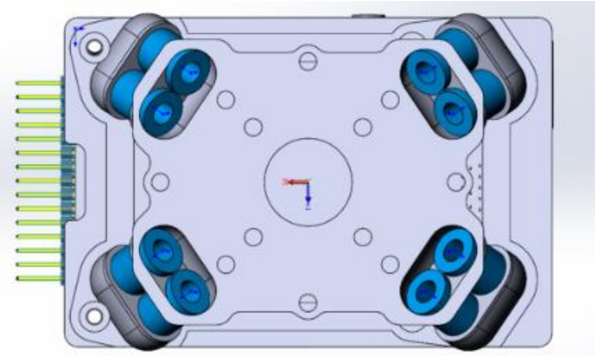
The thrust to weight ratio of the quadcopter without any payloads

$$ratio = \frac{equivalent\ mass\ of\ thrust}{flight\ mass\ of\ quadcopter} = \frac{4.44}{1.34} = 3.4$$

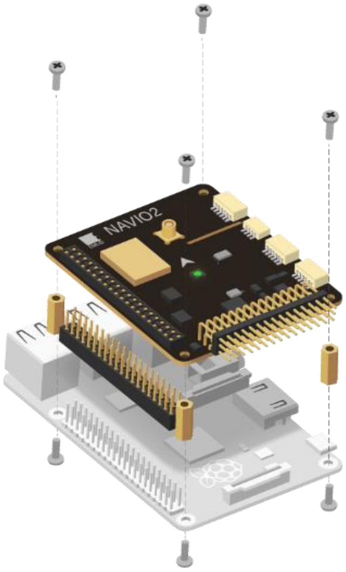
Navio2 hardware setup

The first step was to set up RPi3 for Ardupilot. The Navio2 shield was stacked onto the RPi3 and screws were in place to bind the two securely. Navio2 together with the RPi3 would act as the FC of the quadcopter built. Meanwhile, a 3D model holder through which the FC will be secured onto the quadcopter's airframe was printed. Two parts of the holder were bonded together through 8 shock-absorbing balls. The link provided for the full component list in the PDF file "Drone BOM list" also included these balls.

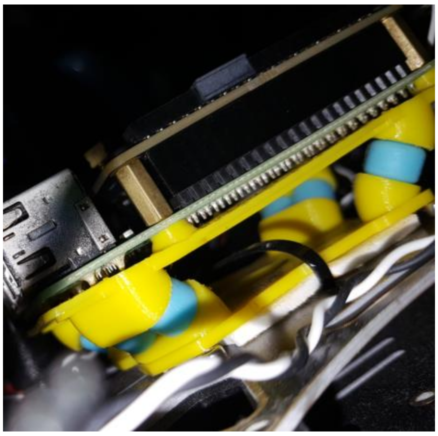
The link to the documented Navio2 hardware setup is as follows:
https://docs.emlid.com/navio2/ardupilot/hardware-setup/



Link of Holder parts to 3D-print
https://github.com/emlid/hardware/blob/master/VibroNavio2top_rev_A.STL
https://github.com/emlid/hardware/blob/master/VibroNavio2bot_rev_A.STL



To insert the shock-absorbing balls to hold the 2 printed holder parts together, do not be too gentle to the balls or it will take forever to complete the task. Note that these balls are made to be flexible and stretchable. Utilise thin tools like a small Allen key to push the tips of a ball so that they get stuck at the necks into the holes between the 2 printed parts, which have been correctly oriented as shown in the figure above. Repeat for another 7 balls, 2 at each corner. Then, attach the back of your Raspberry Pi onto the upper surface of the printed holder (the one with the larger surface area) using the screws provided.



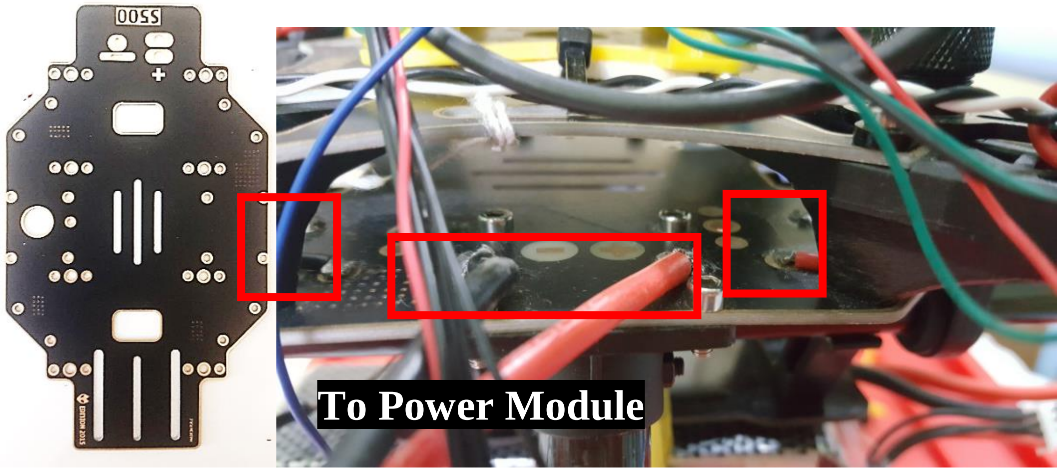
Drone Hardware Setups

After the previous step is done, leave the Raspberry Pi (stacked with the Navio2 flight controller) with the holder and then proceed with setting up the airframe.

You can try assemble your airframe first if you would love to try out, but you will have to disassemble it for soldering purposes after your motors and ESCs arrive.

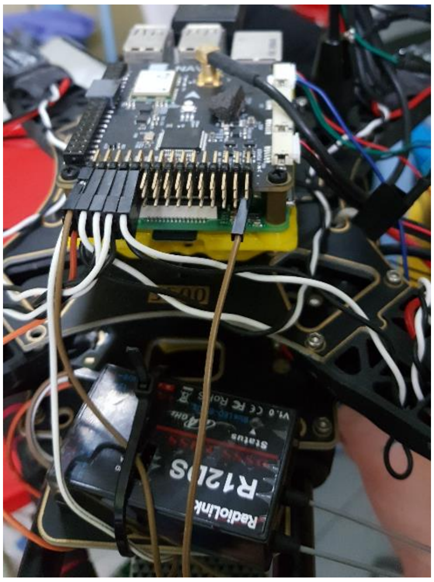
Your airframe should consist of a power distribution board. Find the surface of the board that has circular copper terminals marked with positive and negative signs that are meant for soldering of your components. Use a multimeter’s connectivity test to test that all the positive terminals are connected, so do all the negative terminals. I use the soldered surface to be facing up, so I need to make sure that when I solder the cables, they will not obstruct any screw openings connected to the top central cross arm connector. Find appropriate positive and ground terminals on the power distribution board to solder your power module and all 4 of your ESCs to each corners of the power distribution board. All 4 ESCs and motors are the same and can be connected to anywhere so you need not worry which of them to be connected to which corner. Solder the red cables to the positive terminals on the board and the black wires to the negative terminals on the board. The central cable(s) (usually) is the signal cable to be connected to the navio2 flight controller.

Make sure the positive and negative terminals do not short and test the connectivity again after the solder. Secure the connection to the board with more solders in place and test the connection by slightly pulling the cables. Upon completion, the power distribution board should look like a board with hanging threads connecting securely, 4 ESCs and 1 power module. Each of these components is connecetd by 2 wires – red for positive and black for negative.



Soldering Power Module output and ESC outputs onto PDB

Then, assemble the top central cross arm connector with the quadcopter arms using suitably sized Allen keys, and then connect the combined structure with the power distribution board. Adjust the wirings of the ESCs so that they will not obstruct the connection points, if you decided to have the soldered surface facing upward (less possibility of short circuiting with other wirings but needs to diassemble the airframe for resoldering, which can be extremely rare). Now, mount the holder containing the Raspberry Pi 3 and Navio2 onto the central cross arm connector using thick double-sided tapes.

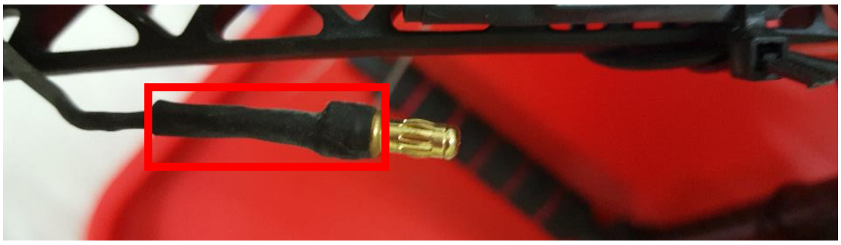


Mount the motor on the end of each of the quadcopter's arm together with the propeller protector (optional), with the 3-phase wires facing inward parallel to each quadcopter's arm. Secure them tightly in place using the provided screws.

Then, if the motors do not have readily soldered bullet connectors, solder the bullet connector to the end of each of the 3-phase wires of the motors. To do this, use a thin head plyer to secure the bullet head connector (may be very hot during soldering), then melt solders into the cylindrical opening. Before the solder hardens, push the wire from the motor into the bullet head connector.

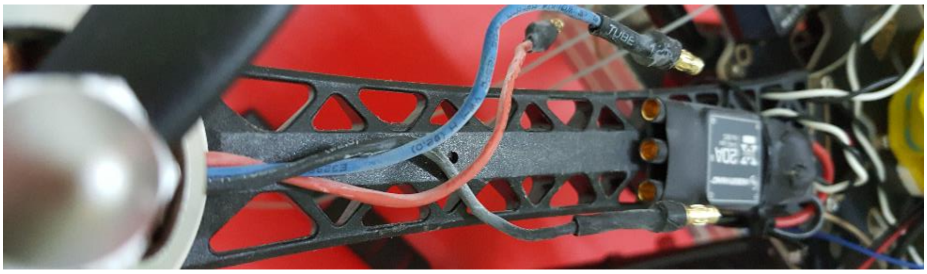
Soldering bullet head connector to the motor input wires

Hold for a few seconds for the solder to harden completely before you move. Try apply force to test the strength of connection. Loose connections may cause motor failure in the mid flight leading to crashes. If the supplier provided the heat-shrink tubes for insulation, cut a sufficient amount of the tube and lay it over the exposed metal part (with solder connected) under the bullet head, and then apply heat from your soldering gun or a lighter to shrink and have it covered. Otherwise, use an electrical insulation tape.



Metal connector after covered with heat-shrink insulator

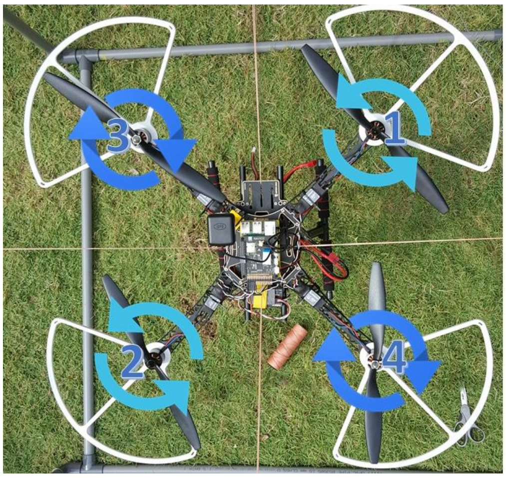
Next, secure the each ESC on the top (or bottom) surface of each arm of the quadcopter using cable ties. Then, for now, **randomly** connect the bullet head connector (may be other types of connectors) of the 3-phase wires of the Brushless DC motors to the 3 outputs of the ESCs by applying a decent amount of strength. The three-phase quasi AC input to the BLDC motor will magnetise coils in the excitable coils in the stator alternatedly such that they repel the permanent magnets (rotor) to turn. The sequence of the 3-phase input to the motor will determine the direction of the rotation of motor. To make sure that the ESCs output a correct sequence of AC current to turn the motor in required directions, you will need to test it out later on.



ESC with unconnected bullet heads

When you have your quadcopter armed for the first time (after you have performed all steps that follows), with all propellers removed, you need to make sure that the direction of motor rotation for each motor is as follows. If any of the motor(s) do not rotate in the direction as required, simply interchange any 2 wires that you have randomly connected from the motor (which has wrong direction of rotation) to its powering ESC. Use your finger to feel all motors rotating in the correction direction as in

1. Front right – Anti-clockwise
2. Rear left – Anti-clockwise
3. Front left – Clockwise
4. Rear right – Clockwise



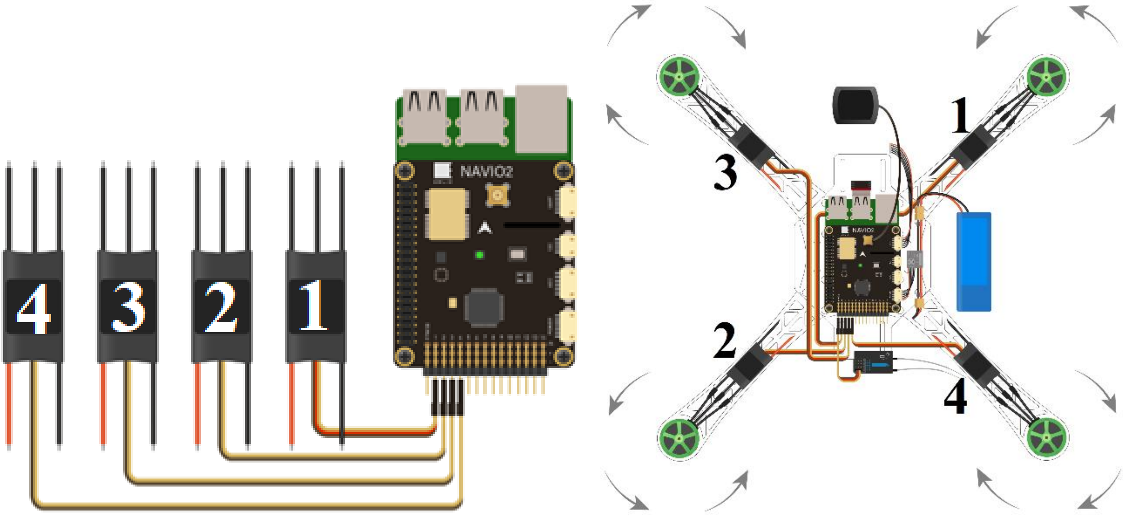
Motor rotation direction

But that is only one of the last steps you would do before which there are still many to be done. Keep this section toward the end (you will be reminded about this once again after that) and proceed with the following.

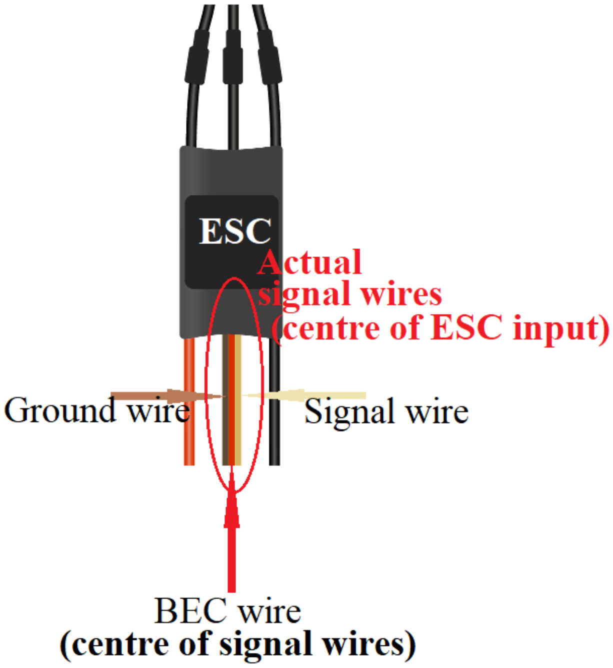
Before that, you will need to connect the ESC to the servo rails of the flight controller as shown in the figure below. Refer to the Emlid documentation in the link:
https://docs.emlid.com/navio2/ardupilot/hardware-setup/

You should now have your ESCs positive and negative inputs soldered onto the Power Distribution board, the central signal cable should be those to connect to the servo rail. Now, the connection of each ESC must be **specific**. Looking from the left of the servo rail, the leftmost column is the SBUS communication lines (we will deal with it later). The second column, third, forth and fifth column, is meant for the first, second, third and forth ESCs respectively. Please connect them according to the diagram provided by Emlid below.

Column	ESC numbering	Position of ESC
2	1	Top Right
3	2	Rear Left
4	3	Top Left
5	4	Rear Right

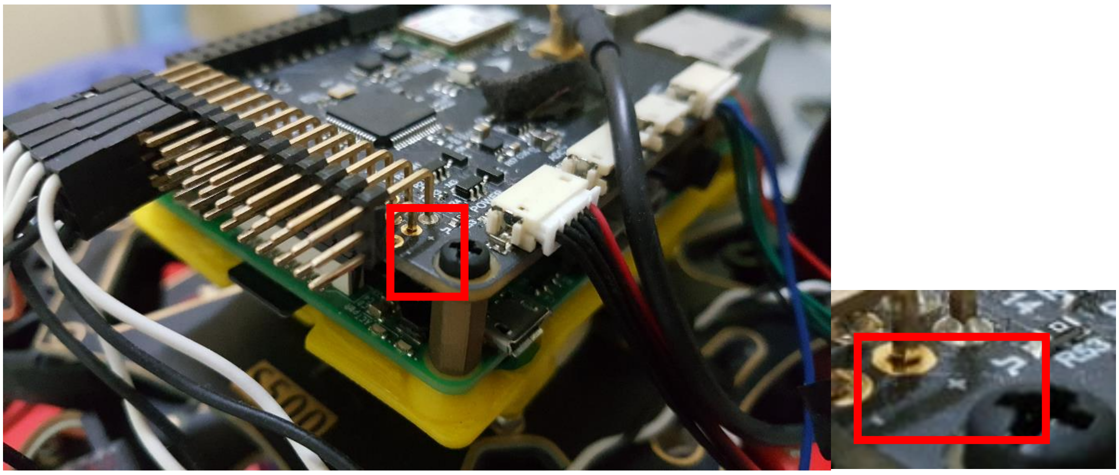


Let’s name the central wire of our ESC as the signal wires. The signal wires consist of the actual signal wire, the BEC wire and the ground wire.



The signal wire (central of ESC) consists of 2 wires (for ESCs without BEC) and 3 wires (for ESCs with BEC). If your ESC is without a BEC, it will not have a central BEC wire. The central of the signal wires is empty. If your ESC has a BEC, the BEC wire is usually ‘the centre of the centre’. It is usually the central wire of your 3 signal wires, which is located in the centre of an ESC input.

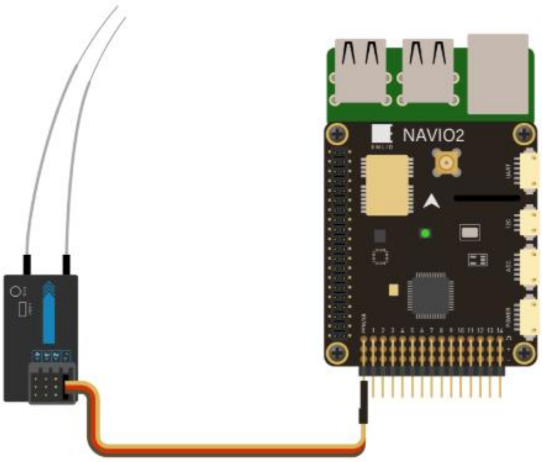
If your ESC has a Battery Eliminator Circuit (BEC), connect only the BEC wire of your first ESC (can be any other ESCs as long as only one), as shown in the diagram above, on the left, as the orange wire extending from the first ESC. For other ESCs the BEC cable is cut off and insulated (don’t cut wrong- only “the **centre of the centre**”). This means that only one of your ESC’s central wire (BEC wire) of the central signal wires is connected to the servo rail. The other BEC wires from the other 3 ESCs are removed. This leaves only the first ESC with 3 signal wires (with BEC wire, actual signal wire and ground wire) and the rest of the ESCs with 2 signal wires (with the actual signal wire and ground wire). We know that usually the “central of the central” is the BEC wire, but to differentiate between the actual signal wire and the ground wire, you have to test it out. Again, do this toward the end after you arm your quadcopter. If your motors don’t work as according to the transmitters command (you can hear the change in speed), you have had it connected wrongly. As a rule of thumb, more often than not, the black or dark coloured wire is the ground wire. On the servo rail side, the signal, positive and negative terminals are clearly labelled as the top, middle and bottom row, respectively. The middle (positive) row is where the remaining one BEC wire (if you have one) should connect to, while the actual signal wire should be connected to the first row (of the respective column) marked with the square-wave signal icon. Lastly, the ground wires should be connected to the respective columns in the last row of the servo rail.



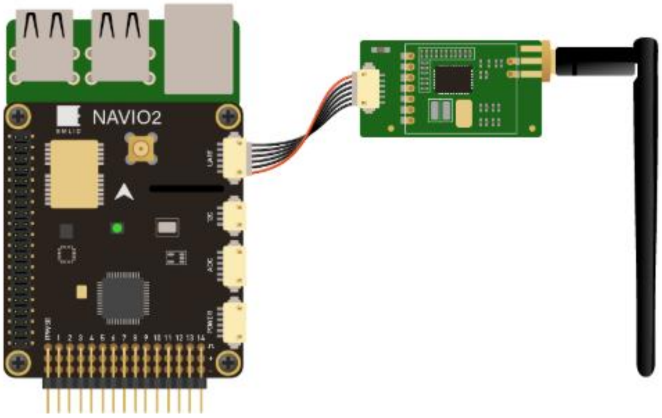
The servo rail from the side with label of signal, positive and negative icon

If your ESCs do not have a BEC, you will not need to remove or add anything.

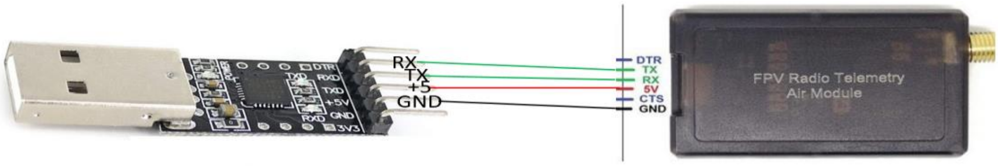
To connect the receiver to the flight controller, you will need three female-to-female jumpers to connect the signal, positive and negative terminals of the SBUS line from the reveiver to the respective signal, positive and negative terminals in the servo rail of the Navio2 flight controller. The first or the leftmost column labelled as ‘PPM/SB’ provides the signal, positive and negative terminal for the SBUS communication on the first, second and third line respectively. Use shorter jumpers (for better tidiness) to connect those to their counterparts in the receiver. Secure the receiver with a cable tie.



After the connection of the ESCs, motors, power module and receiver with the airframe, we are left with the telemetry module, GPS antenna and the battery. The telemetry module comes in a pair – one air module and another ground module. The air module usually comes with a UART micro-header pin connection to be connected to the navio2's UART port as shown below, whereas the ground module comes with USB connector to connect to your PC.

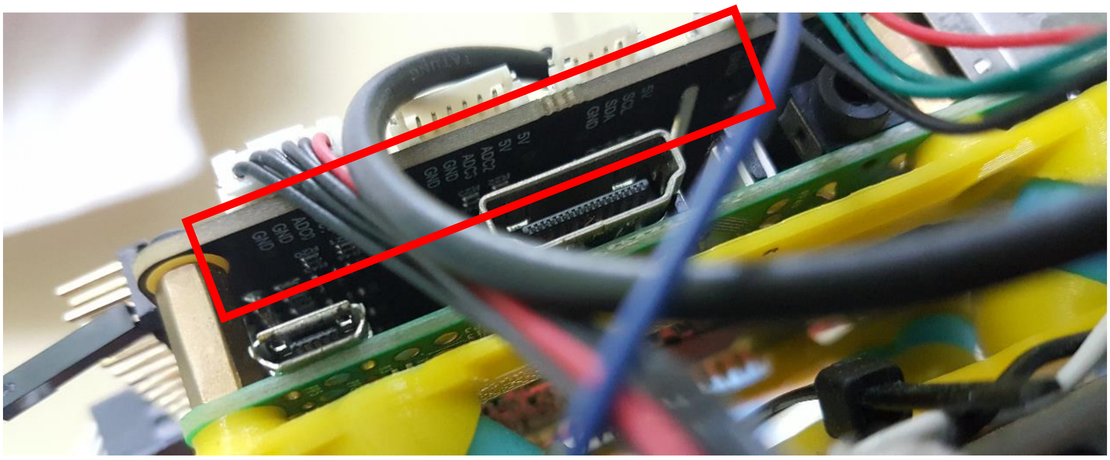


(Skip the following section if you are connecting it the normal way) Nonetheless, you can interchange the air and ground module by connecting the ground module (with USB connector) to the Raspberry Pi, and the air module (with UART connector) to your PC. The latter will require an additional USB-TTL CP2102 converter in order that the air module can be connected to the COM port of your PC. The connection of the air module to the converter required soldering, and is as shown in the figure below.



USB to TTL conversion for telemetry air module

The label on the USB-TTL converter is clear and correct, but on the air module the label within the cover may not be accurate. Maintain the micro-header pin on the telemetry air module side, and cut off the micro-header pin connecting to the UART port. Judging from their connection to the UART port of Navio2, we can know exactly which of the wires are TX, RX, 5V and GND. The port label can be found underneath the Navio2 board or as shown below. The wire that connects to RX on the UART port leads to TX terminal on the air module, while the wire that connects to TX on the UART port leads to RX terminal on the air module. Based on the similar working principle, TX terminal on the air module should be soldered to a jumper connecting to RX of the USB-TTL converter, and vice versa.



Label of ports underneath Navio2

USB-TTL converter soldered to the air module

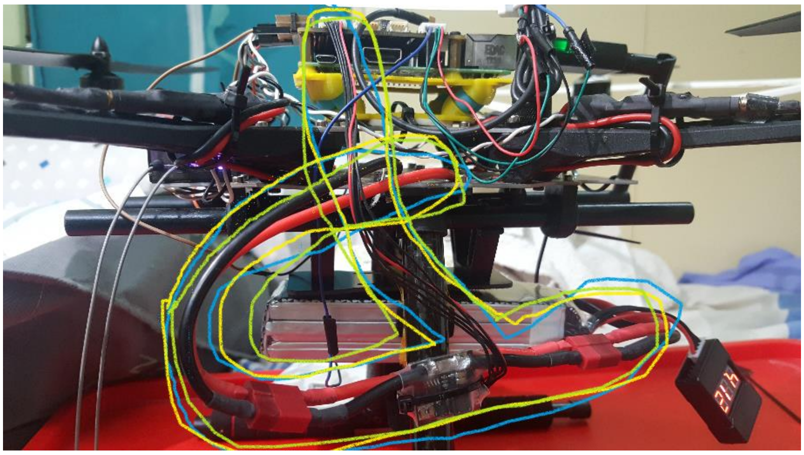
With the 5V and Ground pins connected, the LEDs on the air module will start to blink, signifying the complete circuit in powering up the air module. Make sure that the RX pin of the converter was connected to the TX of the telemetry air module, while the TX pin of the converter was connected to the RX of the air module. Completed connection will show solid light for LED when both the air and ground module are communicating with one another, but that did not imply the telemetry can work in terms of transmitting data for GCS. To achieve this, minor settings have to be done over the RPi3 by logging into the microcomputer's OS.

From the "etc/default/arducopter" file, the connection configuration was updated to allow this. Details of this is shown in the later steps when you will go into setting up in the section "Connecting to your Ground Control Station (GCS)".

A GPS stand is where the antenna is being mounted to, while the other end connects to the flight controller's GNSS receiver through a micro coaxial (MCX) connector.

GPS stand and GPS antenna

The battery is tied securely under the battery holder using the connecting tie/strip for battery and connected to the Power Module as shown below. One of the outputs of the power module should now been soldered to the power distribution board. The other output, a 6-pin micro-header branches out from the power module, is now connected to the power port of your navio2 flight controller as shown in the diagram below.

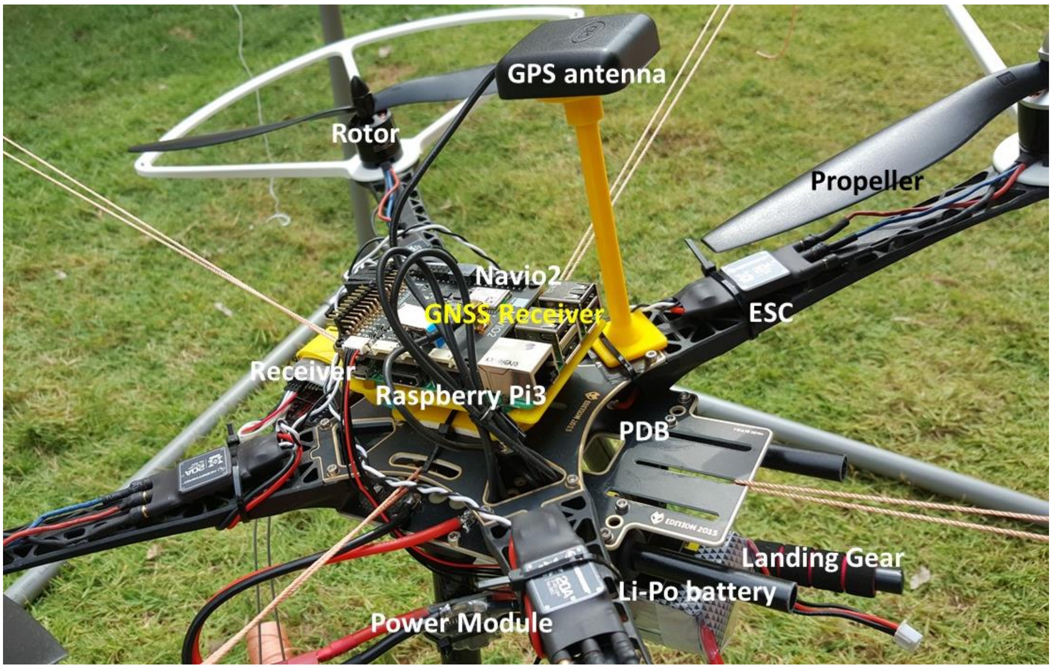


The power from battery splits through the power module to the power distribution board and Navio2

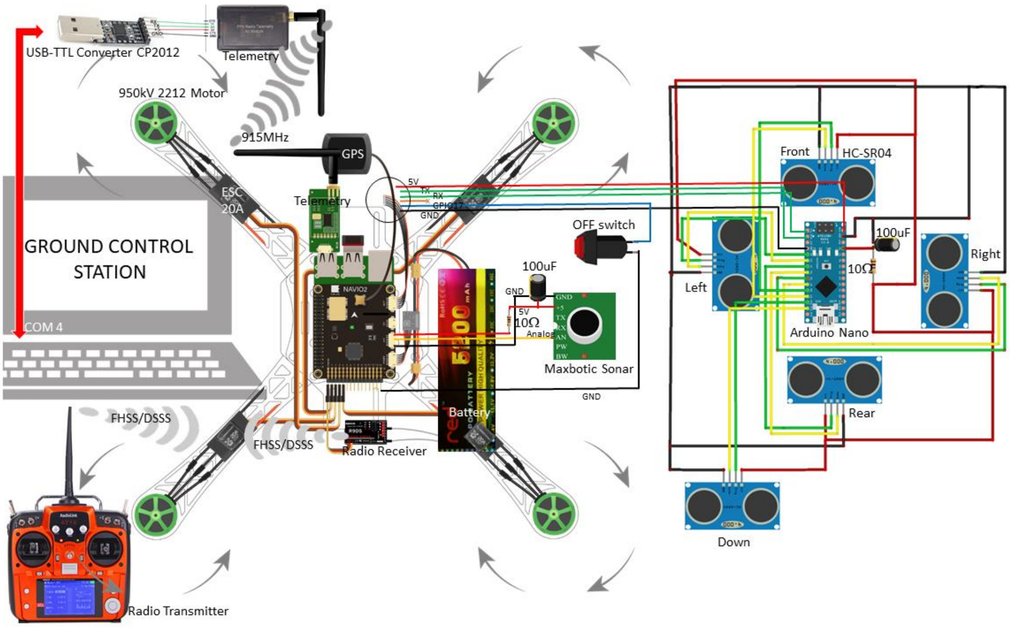
With the battery connected, you will be powering up your Raspberry Pi, Navio2, receiver, and your telemetry. Navio2 will have blinking LEDs whose colours will determine the different conditions of the flight controller. The telemetry module has blinking LED that turns solid when connection is established with its ground counterparts. The receiver should also have LEDs indicative of its supply to the power.

The assembled quadcopter is as shown below. Remember to tie all loose components securely using a cable tie. Then, after you have done with all the hardware setups, you will need to perform a series of software setups, network configurations and flight calibrations before you can set it for its maiden flight.

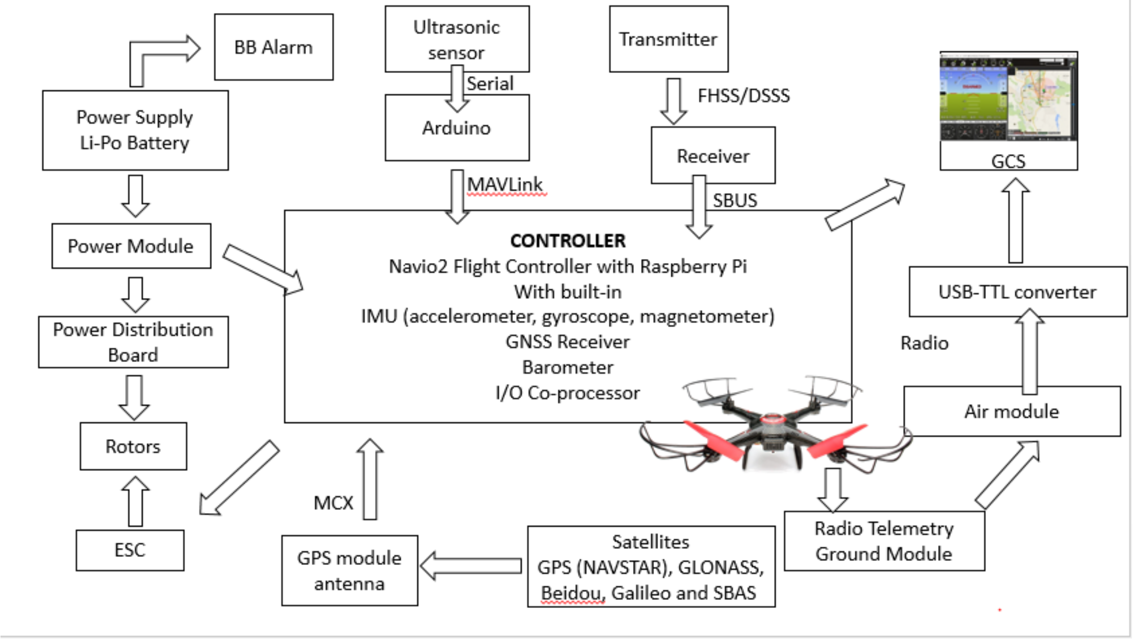
Assembled quadcopter



Quadcopter after assembly



The overall circuit connection of the quadcopter

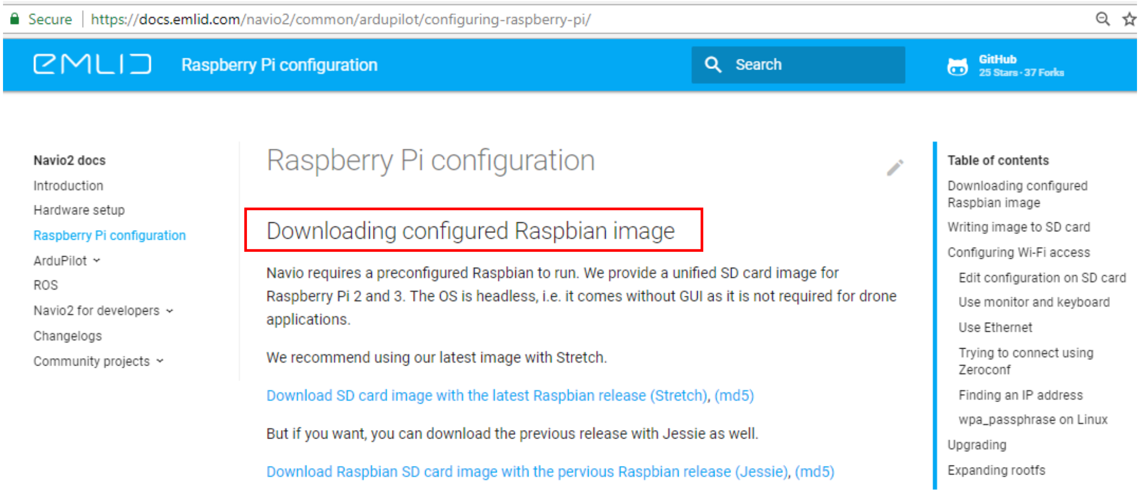


System Architecture

Flashing OS into microSd card

A microSD card which will be used as the storage for the micro-computer, RPi3, has to have the Raspbian Operating system (OS) installed. Choose a suitable size of 16GB or 32GB. Visit the Navio2 developer's documentation at

https://docs.emlid.com/navio2/common/ardupilot/configuring-raspberry-pi/ and download the OS image to your computer. Remember where you store the file because this zipped file downloaded will be "flashed" to your microSD card later on.



Raspberry Pi configuration

Downloading configured Raspbian image

Navio requires a preconfigured Raspbian to run. We provide a unified SD card image for Raspberry Pi 2 and 3. The OS is headless, i.e. it comes without GUI as it is not required for drone applications.

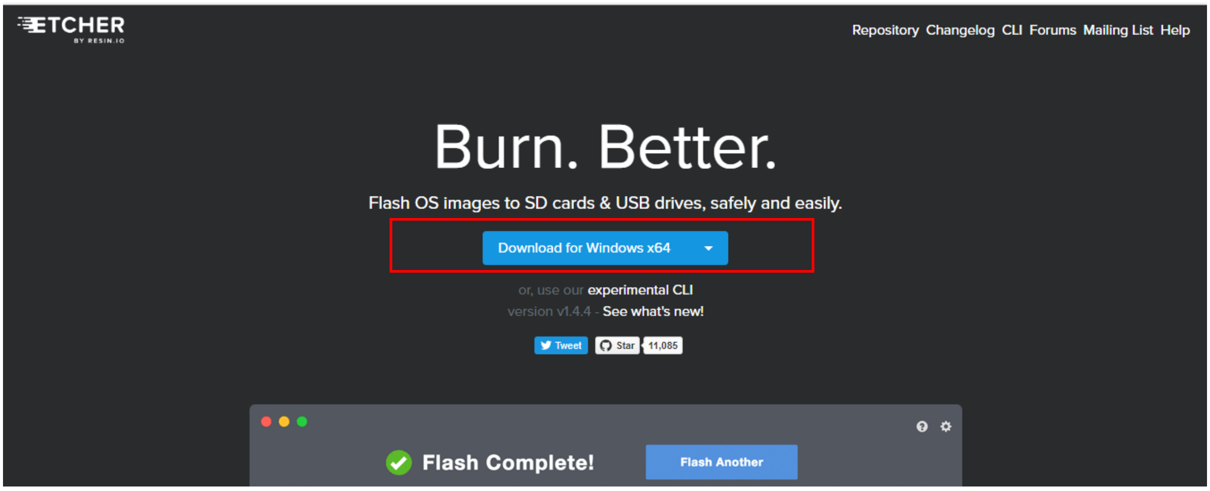
We recommend using our latest image with Stretch.

Download SD card image with the latest Raspbian release (Stretch), (md5)

But if you want, you can download the previous release with Jessie as well.

Download Raspbian SD card image with the pervious Raspbian release (Jessie), (md5)

Then, we need a software to flash the downloaded OS into your microSD card. Visit https://etcher.io/ to download and install Etcher. Etcher file is extracted and run on your PC.



Burn. Better.

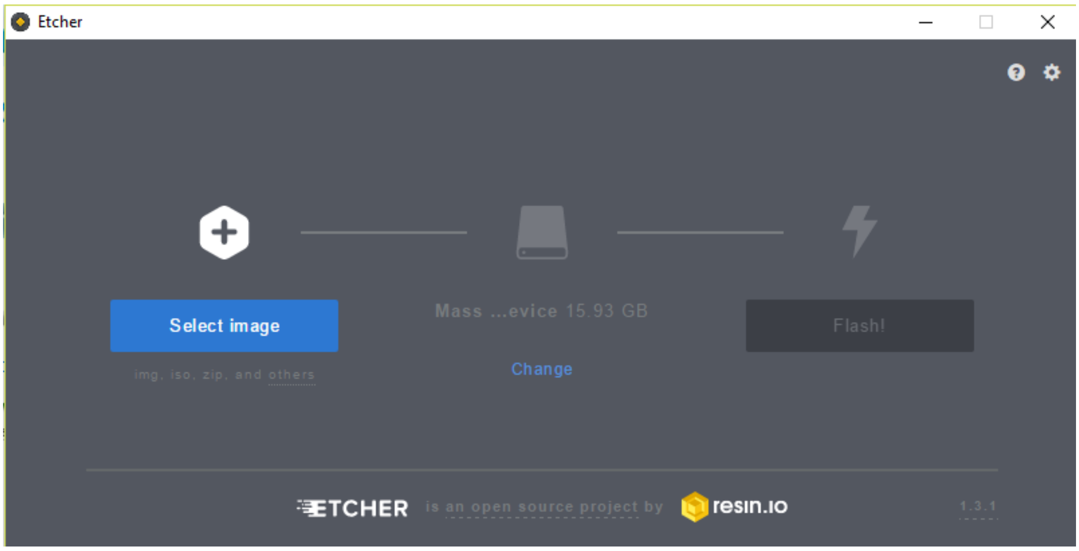
Flash OS images to SD cards & USB drives, safely and easily.

or, use our experimental CLI

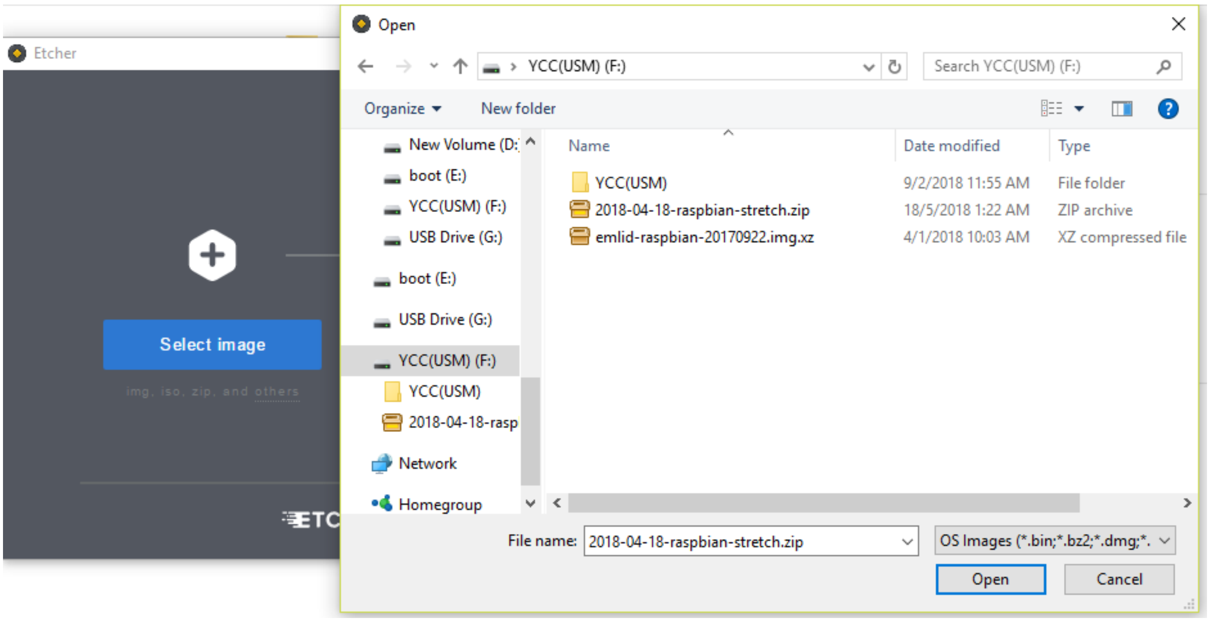
version v1.4.4 - See what's new!

After we get Etcher running, the OS image in the archive file selected is ‘flashed’ into the microSD card. It was found that this process may not be compatible with some PCs, which required other PCs to be used instead.

Plug in your new or newly formatted microSD card to your PC using a suitable card reader or a microUSB adapter. Usually Etcher will automatically select the microSD card as the device to flash. Otherwise change the device to your microSD card before selecting the OS image.

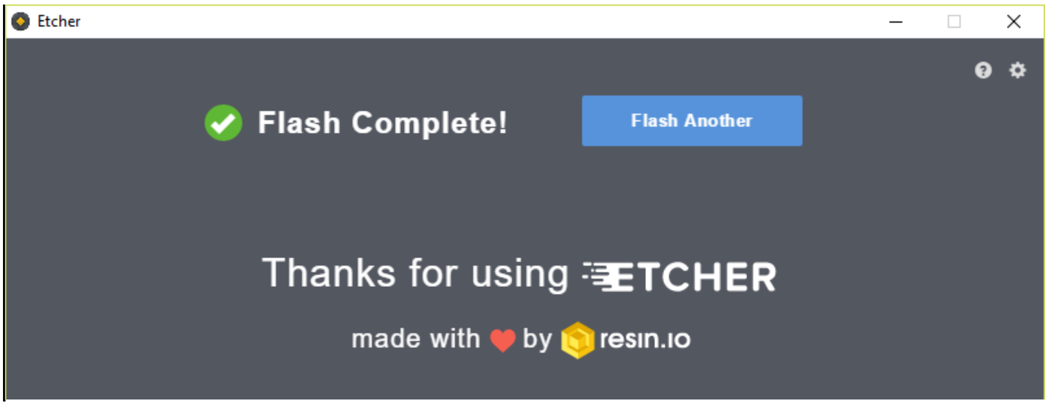


Select the OS image zip file from where you had it downloaded. Then click flash after it is illuminated!



Name	Date modified	Type
YCC(USM)	9/2/2018 11:55 AM	File folder
2018-04-18-raspbian-stretch.zip	18/5/2018 1:22 AM	ZIP archive
emlid-raspbian-20170922.img.xz	4/1/2018 10:03 AM	XZ compressed file

File name: 2018-04-18-raspbian-stretch.zip



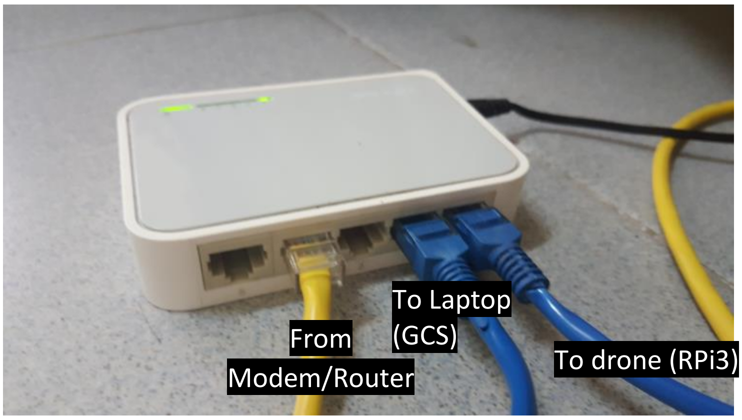
This is all required to flash the OS image to your microSD card. The OS image by Ardupilot is a simple one without fancy GUI, and is simple to use. Note that your microSD card has now been partitioned for use of the Navio2 Flight Controller. If you wish to format it (if somehow it corrupts after flashing or after several times of boots), remember to delete the partitions after your formatting. Softwares such as AOMEI partition manager can be used for this purpose.

After the "flash", your microSD card can now be inserted into your Raspberry Pi3, and then begin network configuration settings, followed by secure shell (SSH) your Raspberry Pi to your laptop using PuTTY.

Raspberry Pi Wi-Fi network configuration setting

To secure shell your Raspberry Pi to your laptop, you need to connect both your Raspberry Pi and your laptop to the same LAN network – be it through wireless (Wi-Fi), or with cable (Ethernet) connection.

Let's start with connecting your Raspberry Pi to the internet. It is easier if you are using LAN (or Ethernet) cable. Connect both the RPi and your laptop to the same LAN cable port, and this requires you to have a LAN cable splitter. Split the LAN port to 2 LAN cables. One connect to your laptop and the other to your RPi. Just like on your PC, if you are connecting to the internet using the Ethernet cable, you would not need to perform WPA network configuration (security) on your Raspberry Pi micro-computer so you can skip the step below



A splitter splits LAN into 2: one connects to laptop the other to your drone

Otherwise, if you do not have a splitter or if you are not using a LAN cable port, then follow through the wireless configuration step below, just like you would set up connection of your PC to wireless networks – it is only slightly more complicated than how you would normally connect your daily devices to Wi-Fi.

Please note that this network configuration setting is just to connect Raspberry Pi to Wi-Fi network so that it shares with your PC the same network you would connect your PC to. It is different from the network configuration that you will perform for Ground Control Station (GCS) communication via a telemetry module, which will be discussed later on.

As aforementioned, if you are using the Ethernet cable, you can skip the following step and directly go to “SSH Raspberry Pi on laptop”.

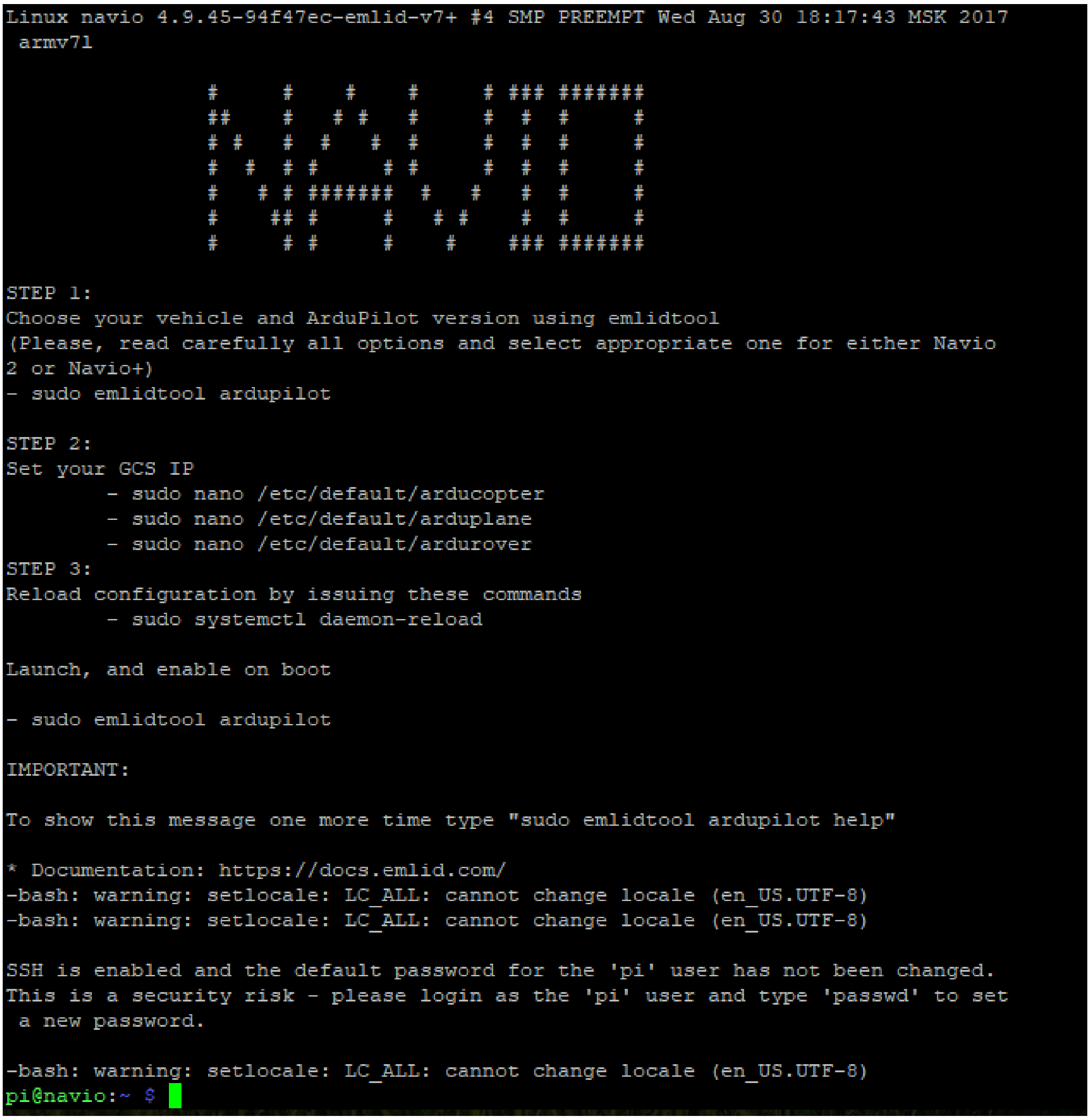
Otherwise, network configuration setting is required. Because this step is to enable SSH Raspberry Pi onto your laptop, before that can be done, you can only login Raspberry Pi using an external HDMI monitor and a keyboard. Only for the first time of connection setup (unless you prefer not to use SSH on your laptop but always use a monitor instead), connect your Raspberry Pi to a HDMI monitor or use a HDMI-to-VGA converter to connect to a VGA monitor.

Power up the Raspberry Pi either by connecting a microUSB charge cable, or by connecting it to the Li-Po battery through the power module. Note that if you are using the Li-Po battery, always attach a BB alarm to monitor the battery voltage to avoid the irrepairable damage if the battery voltage falls below 3 V per cell.

Then, by looking at the monitor screen, login to the OS by entering the default username and password. The default username is ‘pi’ and the password is ‘raspberry’. The password may not show up but anyways hit enter for each command entered.

```
login as: pi
pi@navio.local's password:
```

The moment when you login to the OS, you will be greeted with the page as follows

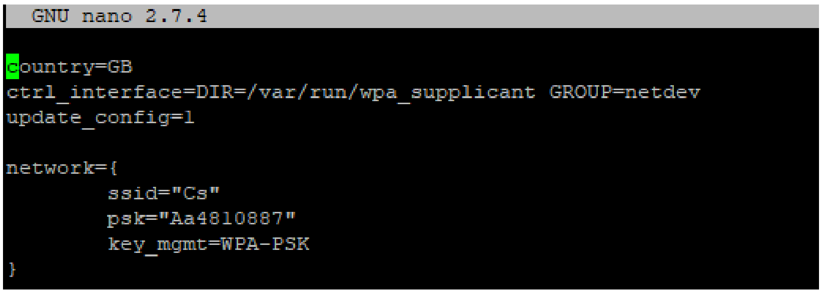


```
Linux navio 4.9.45-94f47ec-emlid-v7+ #4 SMP PREEMPT Wed Aug 30 18:17:43 MSK 2017
 armv7l

                #     #    #    #     # ### #######
                ##    #   # #   #     #  #  #     #
                # #   #  #   #  #     #  #  #     #
                #  #  # #     # #     #  #  #     #
                #   # # #######  #   #   #  #     #
                #    ## #     #   # #    #  #     #
                #     # #     #    #    ### #######

STEP 1:
Choose your vehicle and ArduPilot version using emlidtool
(Please, read carefully all options and select appropriate one for either Navio
2 or Navio+)
- sudo emlidtool ardupilot

STEP 2:
Set your GCS IP
        - sudo nano /etc/default/arducopter
        - sudo nano /etc/default/arduplane
        - sudo nano /etc/default/ardurover
STEP 3:
Reload configuration by issuing these commands
        - sudo systemctl daemon-reload

Launch, and enable on boot

- sudo emlidtool ardupilot

IMPORTANT:

To show this message one more time type "sudo emlidtool ardupilot help"

* Documentation: https://docs.emlid.com/
-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
pi@navio:~ $ 
```

Copy and paste the command:

sudo nano /boot/wpa_supplicant.conf

as in  `pi@navio:~ $ sudo nano /boot/wpa_supplicant.conf` and hit enter to go to the file to configure Wi-Fi setting. Password and username of the network over which both the RPi3 and your laptop (with PuTTY) would connect to, is entered as shown. Fill your network name between the “ ” as in “*NETWORK NAME*” for ssid line and the password of your Wi-Fi network for psk line. If your local network is not password-protected, just leave the psk’s “” empty. These are nothing but the network name and password that you would normally enter to connect your devices to the Wi-Fi. To move the green text insertion cursor, use the arrow keys.



```
  GNU nano 2.7.4

country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
        ssid="Cs"
        psk="Aa4810887"
        key_mgmt=WPA-PSK
}
```

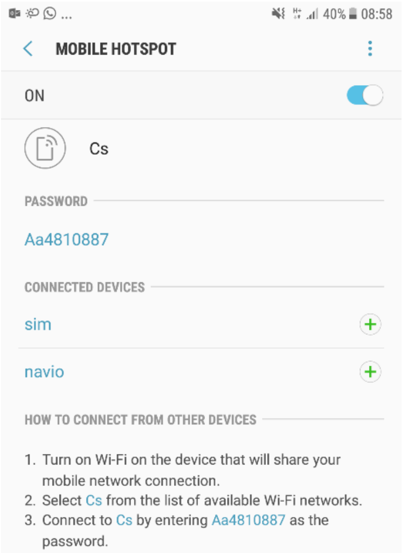
Wi-Fi network configuration setting of RPi3

After that, press “Ctrl+X” to quit and hit key “Y” followed by “Enter” to save what you have editted on the network configuration file. Reboot the ardupilot by typing the command sudo reboot.

If you do not have a monitor or a keyboard, the whole editing process can also be done by editing the text file named wpa_supplicant.conf in the microSD card. To do this, just remove the micro SD card from your Raspberry Pi, plug it into your laptop or PC, and then look for the text file to edit the ssid and psk setting just as you would edit it using the monitor. This time, right-click the file and open with suitable softwares such as WordPad. Save the txt file after editting.

Then, if you are on the monitor display, reboot your Raspberry Pi by typing the command sudo reboot. After device reboot, your Raspberry Pi will then be able to automatically connect to your set Wi-Fi network when it is booted (though it may take some time for the connections). The connection of both your Raspberry Pi and your laptop to the same LAN (either Wi-Fi or Ethernet) is necessary for SSH connection in the following step.

Note: Using mobile or PC hotspot is also the same as using ordinary Wi-Fi. You can monitor from your phone or computer the devices that are connected to your network. If it cannot connect you may restart the RPi3 or recheck the network configuration name and password you have edited in your text file (note that they are case-sensitive and there should not be additional spacing when there is actually none), and wait until it connects (both navio and your PC) and is shown.



ON

Cs

PASSWORD

Aa4810887

CONNECTED DEVICES

sim

navio

HOW TO CONNECT FROM OTHER DEVICES

1. Turn on Wi-Fi on the device that will share your mobile network connection.
2. Select Cs from the list of available Wi-Fi networks.
3. Connect to Cs by entering Aa4810887 as the password.

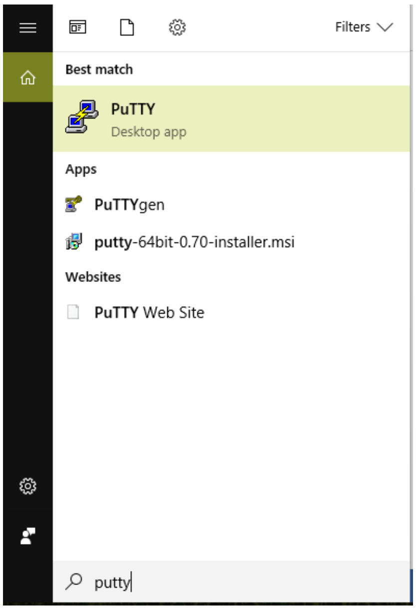
Mobile hotspot shows both navio and the laptop connected

SSH Raspberry Pi on your laptop using PuTTY

PuTTY is one of the network terminal emulator software that enables login to Raspberry Pi using your laptop. It is fine if you do not wish to SSH your Raspberry Pi with your laptop, but this will require you to connect your Raspberry Pi to a monitor and keyboard every time you want to make a configuration settings, such as setting up the Ardupilot, or setting up GCS communication. If that is the case, you don't even have to perform the Wi-Fi configuration setting and do not need any internet connection. Just directly proceed to "setting up Ardupilot" through the monitor instead of through PuTTY.

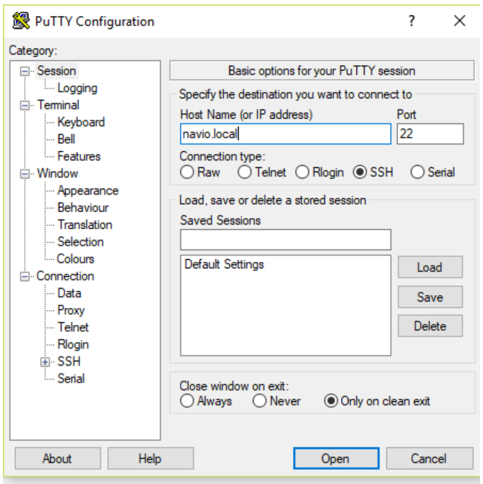
Install PuTTY from the link below and follow through all the installation steps. Then search in your PC for PuTTY desktop app and run it.

Link to PuTTY download:
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html



putty

To connect with PuTTY, type in the configuration “navio.local” as the IP address and enter (or verify) the default port 22 as shown in the figure. Restart PuTTY and try it for several times if it prompts error that local host does not exist. If the condition persists, try disconnect supply to your Raspberry Pi and then reconnect it, i.e. to reboot your Raspberry Pi.



Host Name (or IP address): navio.local

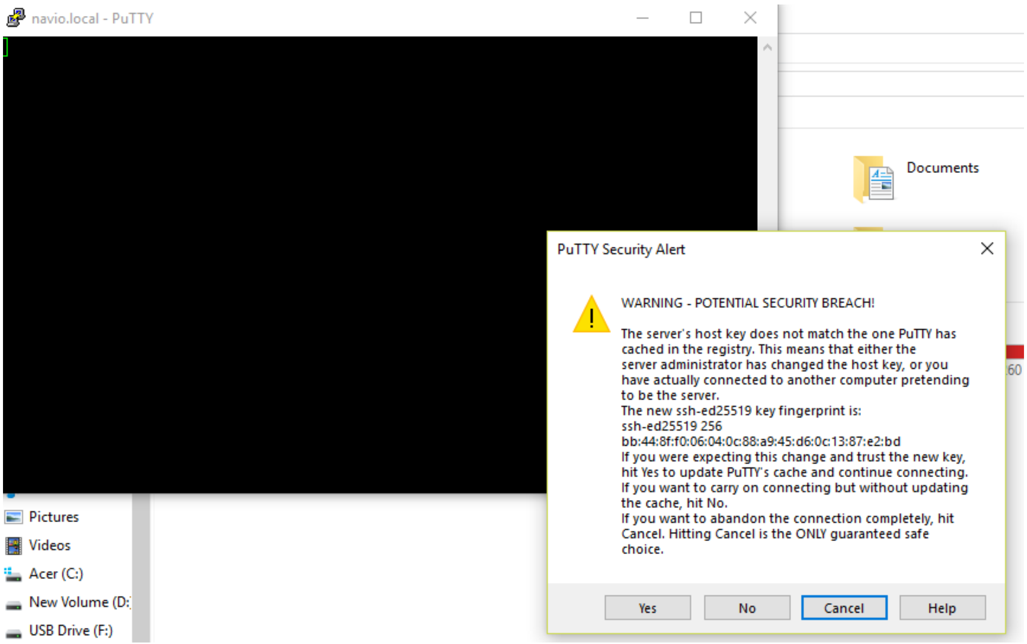
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

Repeat the action until PuTTY is started and you are prompted with a warning message from PuTTY. Allow connection by hitting “Yes”. Then, after entering the default “pi” as username and “raspberry” as password, you will be greeted with the same interface you would find by connecting your Raspberry Pi directly to a monitor, but this time is an SSH to your laptop.

If the error “Local host does not exist” persists, it means that you didn’t manage to connect **both** your laptop and your Raspberry Pi to the **same** and working network. Try using your mobile hotspot to see if both your laptop and Raspberry Pi (default name of device is navio) are in the list of connected device. If the navio device is not connected, check back to your network configuration setting.



```
WARNING - POTENTIAL SECURITY BREACH!

The server's host key does not match the one PuTTY has
cached in the registry. This means that either the
server administrator has changed the host key, or you
have actually connected to another computer pretending
to be the server.
The new ssh-ed25519 key fingerprint is:
ssh-ed25519 256
bb:44:8f:f0:06:04:0c:88:a9:45:d6:0c:13:87:e2:bd
If you were expecting this change and trust the new key,
hit Yes to update PuTTY's cache and continue connecting.
If you want to carry on connecting but without updating
the cache, hit No.
If you want to abandon the connection completely, hit
Cancel. Hitting Cancel is the ONLY guaranteed safe
choice.
```

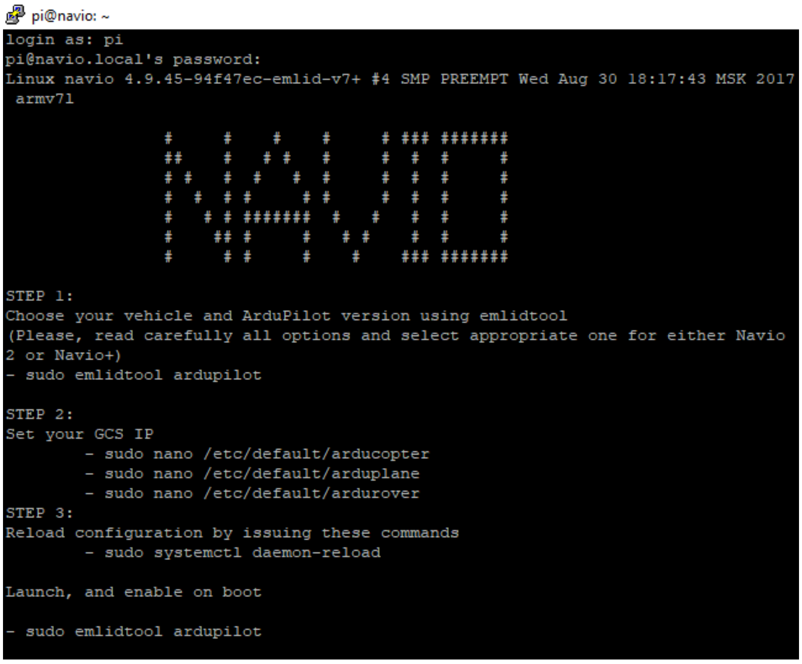
The entire setup steps can be referred from the official navio documentation: https://docs.emlid.com/navio2/common/ardupilot/installation-and-running/

Setting up Ardupilot

If you have powered your Raspberry Pi from Li-Po battery through the Power Module that also powers your ESCs, the ESC will beeps continuously until you have done calibrating your hardware. If you think the beeping sound is annoying, use a microUSB cable to power up the board instead. However, it is not recommended so in the following step because using a microUSB cable will add physical difficulties to the calibration process. Switch to battery connection during the flight calibration steps will ease things out.

Navio2 battery connection vs cabled connection

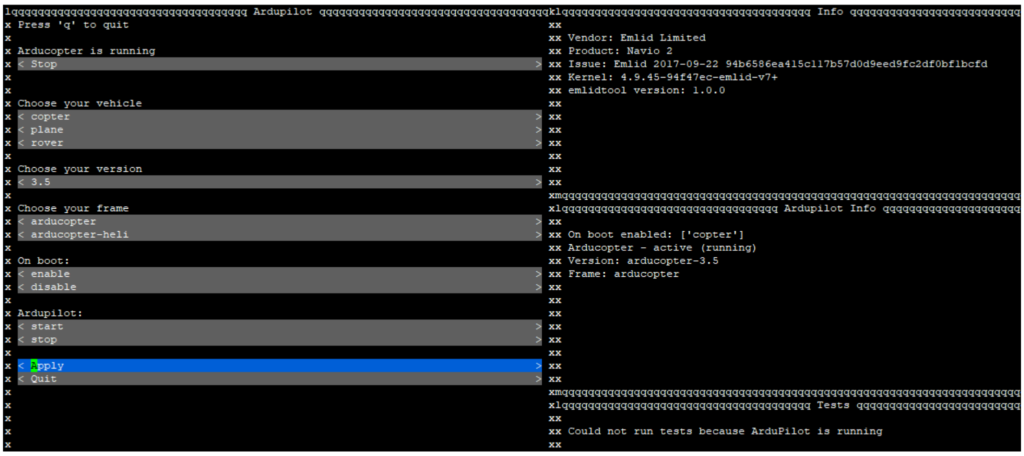
Once the connection was established, entering the default ‘pi’ and ‘raspberry’ username and password will bring forward the same simple interface as shown below. As mentioned, you can lead to this interface either through SSH it on your laptop (where you have to connect both your laptop and RPi to the same LAN) or directly through a monitor.



```
login as: pi
pi@navio.local's password:
Linux navio 4.9.45-94f47ec-emlid-v7+ #4 SMP PREEMPT Wed Aug 30 18:17:43 MSK 2017
 armv7l

                #     #    #    #     # ### #######
                ##    #   # #   #     #  #  #     #
                # #   #  #   #  #     #  #  #     #
                #  #  # #     # #     #  #  #     #
                #   # # #######  #   #   #  #     #
                #    ## #     #   # #    #  #     #
                #     # #     #    #    ### #######

STEP 1:
Choose your vehicle and ArduPilot version using emlidtool
(Please, read carefully all options and select appropriate one for either Navio
2 or Navio+)
- sudo emlidtool ardupilot

STEP 2:
Set your GCS IP
        - sudo nano /etc/default/arducopter
        - sudo nano /etc/default/arduplane
        - sudo nano /etc/default/ardurover
STEP 3:
Reload configuration by issuing these commands
        - sudo systemctl daemon-reload

Launch, and enable on boot

- sudo emlidtool ardupilot
```

In order to set up Ardupilot, the vehicle type, version and board had to be selected through the command “sudo emlidtool ardupilot”

`pi@navio:~ $ sudo emlidtool ardupilot`. Use right-click (not Ctrl-C and Ctrl-V) if you want to copy and paste, and hit enter to run the command. The interface will show as below. Follow through the selection of your vehicle type – copter, version of Ardupilot, frame – Arducopter, whether you want Ardupilot to start on boot – enable, start the Arducopter and Apply.



```
lqqqqqqqqqqqqqqq Ardupilot qqqqqqqqqqqqqqqqklqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Info qqqqqqqqqqqqqqqqqqqqqqqqqqq
x Press 'q' to quit                        xx
x                                          xx Vendor: Emlid Limited
x Arducopter is running                    xx Product: Navio 2
x < Stop                                 > xx Issue: Emlid 2017-09-22 94b6586ea415c117b57d0d9eed9fc2df0bf1bcfd
x                                          xx Kernel: 4.9.45-94f47ec-emlid-v7+
x                                          xx emlidtool version: 1.0.0
x Choose your vehicle                      xx
x < copter                               > xx
x < plane                                > xx
x < rover                                > xx
x                                          xx
x Choose your version                      xx
x < 3.5                                  > xx
x                                          xmqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
x Choose your frame                        xlqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Ardupilot Info qqqqqqqqqqqqqqqqqqqqq
x < arducopter                           > xx
x < arducopter-heli                      > xx On boot enabled: ['copter']
x                                          xx Arducopter - active (running)
x On boot:                                 xx Version: arducopter-3.5
x < enable                               > xx Frame: arducopter
x < disable                              > xx
x                                          xx
x Ardupilot:                               xx
x < start                                > xx
x < stop                                 > xx
x                                          xx
x < Apply                                > xx
x < Quit                                 > xx
x                                          xmqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
x                                          xlqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Tests qqqqqqqqqqqqqqqqqqqqqqqqq
x                                          xx
x                                          xx Could not run tests because ArduPilot is running
x                                          xx
```

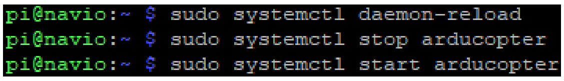
Emlid tool setting

Then, hit ‘q’ to exit. To check if the ardupilot will be started on boot of the Raspberry Pi, the command:

systemctl is-enabled arducopter as in `pi@navio:~ $ systemctl is-enabled arducopter` is entered.

The change in vehicle setting required reloading and restarting of the vehicle through the commands, one after another:

sudo systemctl stop arducopter
sudo systemctl daemon-reload
sudo systemctl start arducopter



```
pi@navio:~ $ sudo systemctl daemon-reload
pi@navio:~ $ sudo systemctl stop arducopter
pi@navio:~ $ sudo systemctl start arducopter
```

Or less preferably,
sudo reboot
to directly reboot the arducopter

Connecting to your Ground Control Station (GCS)

A Ground Control station is your laptop with a software such as Mission Planner used to monitor the flight in real-time.

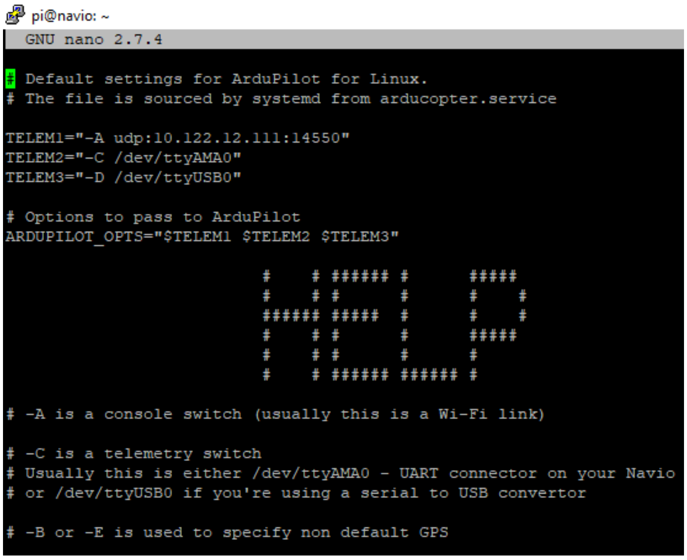
You can connect your quadcopter with your Ground Control Station (GCS) using LAN connection (udp connection), just as you would do to set up SSH communication. Udp connection is useful when you want to connect your drone to the GCS for simple tasks such as pre-flight checking, post-flight analysis or retrieving flight data. However, it is recommended that to monitor a flight, a telemetry module pair is used because they are more reliable (do not rely on network connectivity strength), and are easier to connect.

For both udp and telemetry connection, a one-time configuration setting is required for the GCS communication. Type the command to configure telemetry network communication setting:

sudo nano /etc/default/arducopter

```
pi@navio:~ $ sudo nano /etc/default/arducopter
```

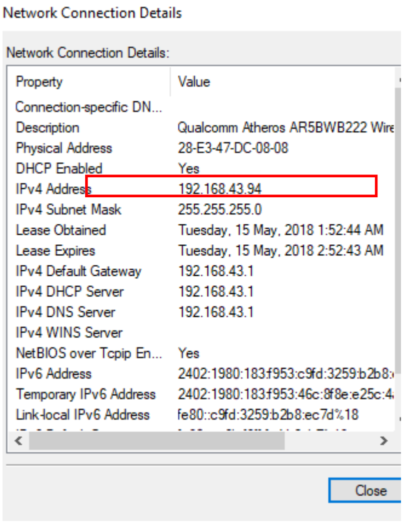
If the GCS is going to be connected to the Arducopter through udp, i.e. LAN connection, the IP address of the network, followed by the default udp baud rate, 14550 is entered.



```
pi@navio: ~
  GNU nano 2.7.4

# Default settings for ArduPilot for Linux.
# The file is sourced by systemd from arducopter.service

TELEM1="-A udp:10.122.12.111:14550"
TELEM2="-C /dev/ttyAMA0"
TELEM3="-D /dev/ttyUSB0"

# Options to pass to ArduPilot
ARDUPILOT_OPTS="$TELEM1 $TELEM2 $TELEM3"

                          #    # ###### #      #####
                          #    # #      #      #    #
                          ###### #####  #      #    #
                          #    # #      #      #####
                          #    # #      #      #
                          #    # ###### ###### #

# -A is a console switch (usually this is a Wi-Fi link)

# -C is a telemetry switch
# Usually this is either /dev/ttyAMA0 - UART connector on your Navio
# or /dev/ttyUSB0 if you're using a serial to USB convertor

# -B or -E is used to specify non default GPS
```

The network IP address was obtained from network and sharing centre shown in the figure below. Take for example this case, we will replace

TELEM1=“-A udp:10.122.12.111:14550” with
TELEM1=“-A udp:192.168.43.94:14550”

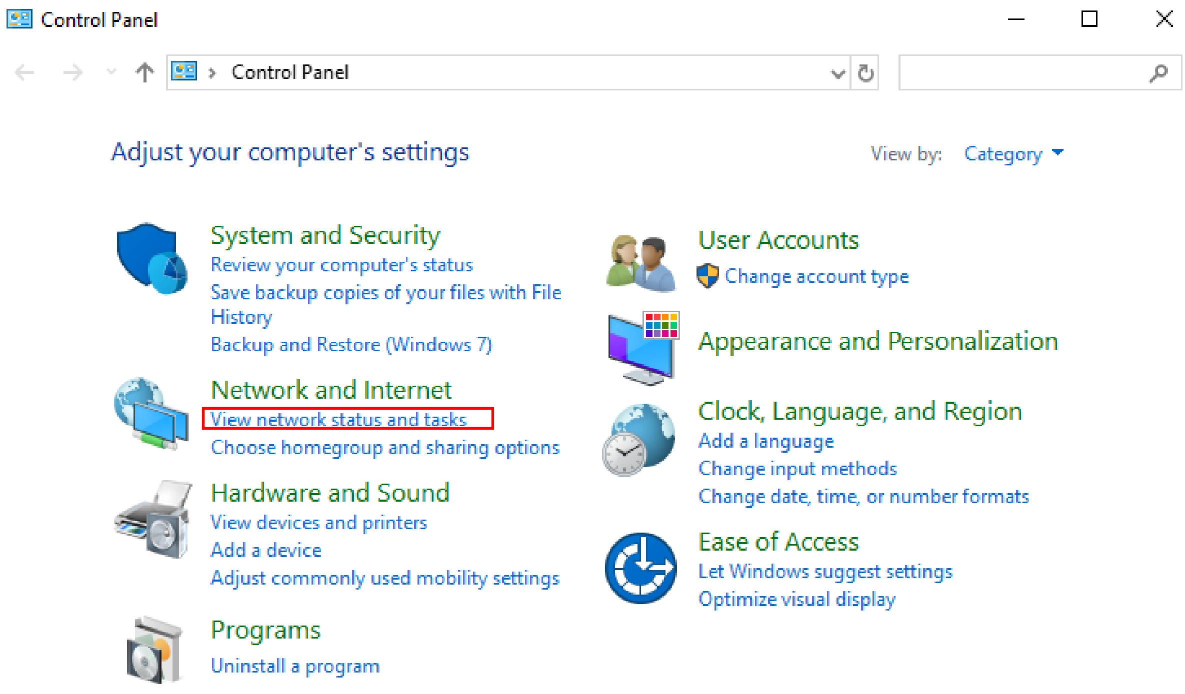


Network Connection Details

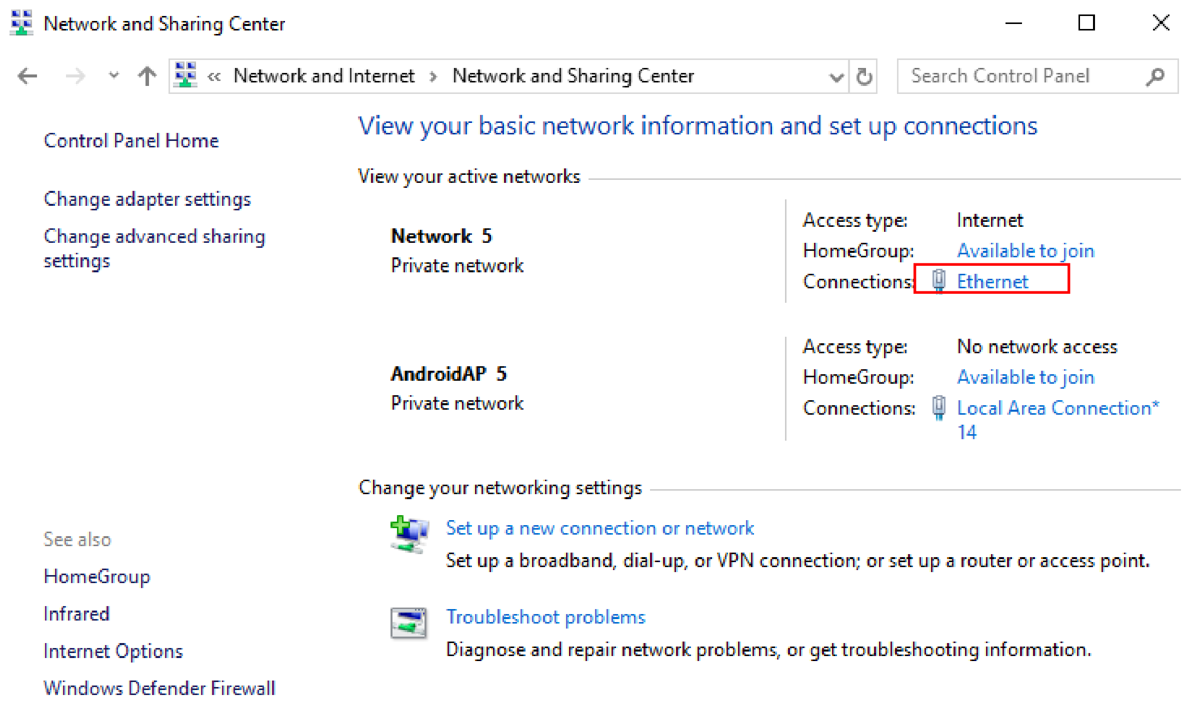
Network Connection Details:

Property	Value
Connection-specific DN...	
Description	Qualcomm Atheros AR5BWB222 Wire
Physical Address	28-E3-47-DC-08-08
DHCP Enabled	Yes
IPv4 Address	192.168.43.94
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Tuesday, 15 May, 2018 1:52:44 AM
Lease Expires	Tuesday, 15 May, 2018 2:52:43 AM
IPv4 Default Gateway	192.168.43.1
IPv4 DHCP Server	192.168.43.1
IPv4 DNS Server	192.168.43.1
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes
IPv6 Address	2402:1980:183f953:c9fd:3259:b2b8:
Temporary IPv6 Address	2402:1980:183f953:46c:8f8e:e25c:4
Link-local IPv6 Address	fe80::c9fd:3259:b2b8:ec7d%18

To open the network connection details in network and sharing centre, go to your control panel and press “view network status and task”



From the network and sharing centre, select the common LAN that you have had both your Raspberry Pi and your laptop connected to. The network address is taken from the IPV4 address shown above.



View your basic network information and set up connections

View your active networks

Network 5
Private network

Access type: Internet
HomeGroup: Available to join
Connections: Ethernet

AndroidAP 5
Private network

Access type: No network access
HomeGroup: Available to join
Connections: Local Area Connection* 14

Change your networking settings

Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

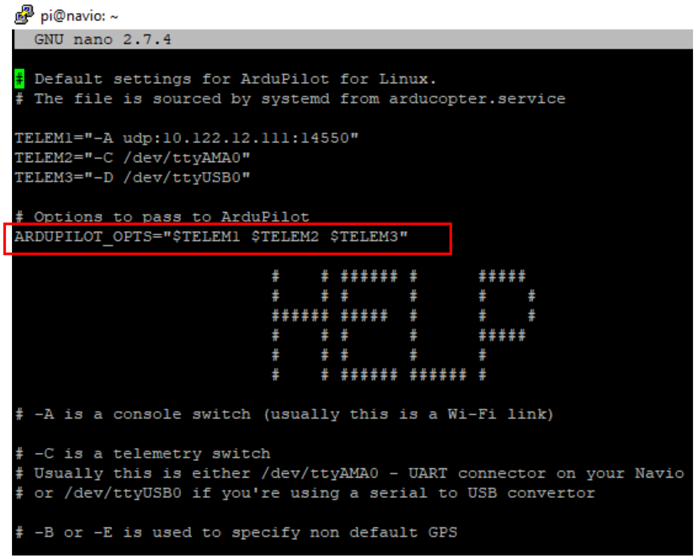
Now if the GCS is going to be connected through a pair of telemetry module, the second line with ‘TELEM2’ was updated. Simply uncomment the line by removing the # symbol in the beginning of the line

TELEM2=“-C /dev/ttyAMA0”

If a USB telemetry (usually a ground module) is used, the third line with ‘TELEM3’ was added. This time replace the “AMA” in the line with “USB” as

TELEM3=“-D /dev/ttyUSB0”

By specifying all entered ‘TELEM1’ to ‘TELEM3’ as your options, any types of connections as mentioned can be established without the need of a reconfiguration in this interface.

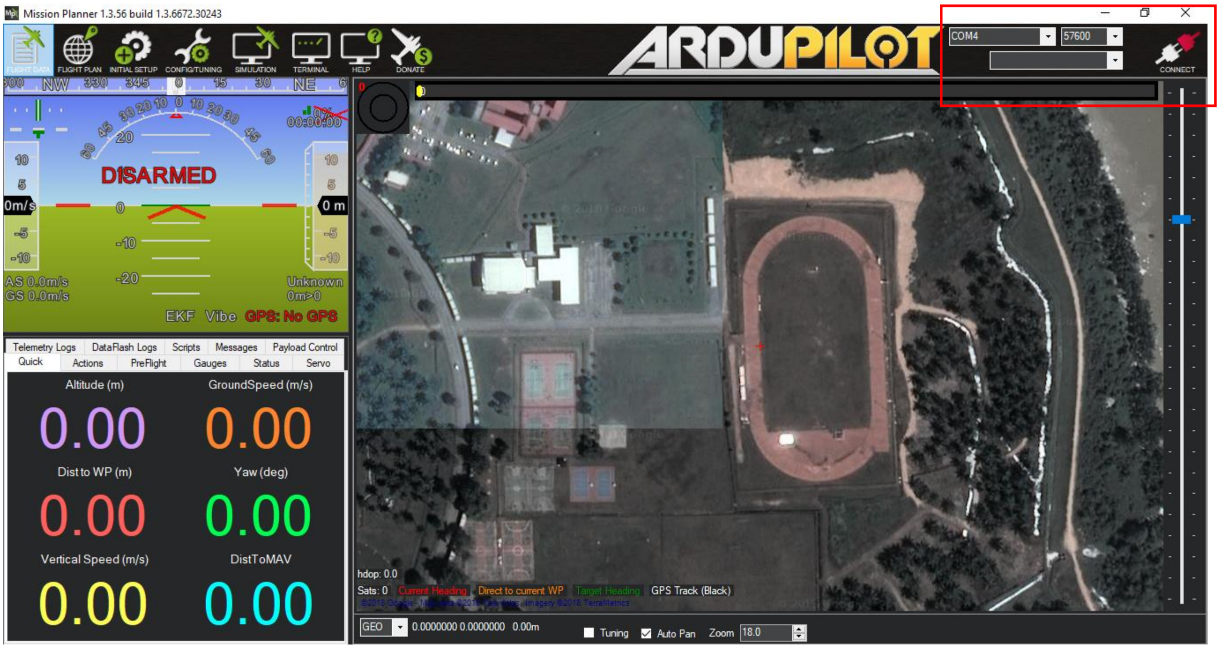


```
pi@navio: ~
  GNU nano 2.7.4

# Default settings for ArduPilot for Linux.
# The file is sourced by systemd from arducopter.service

TELEM1="-A udp:10.122.12.111:14550"
TELEM2="-C /dev/ttyAMA0"
TELEM3="-D /dev/ttyUSB0"

# Options to pass to ArduPilot
ARDUPILOT_OPTS="$TELEM1 $TELEM2 $TELEM3"

                          #    # ###### #      #####
                          #    # #      #      #    #
                          ###### #####  #      #    #
                          #    # #      #      #####
                          #    # #      #      #
                          #    # ###### ###### #

# -A is a console switch (usually this is a Wi-Fi link)

# -C is a telemetry switch
# Usually this is either /dev/ttyAMA0 - UART connector on your Navio
# or /dev/ttyUSB0 if you're using a serial to USB convertor

# -B or -E is used to specify non default GPS
```

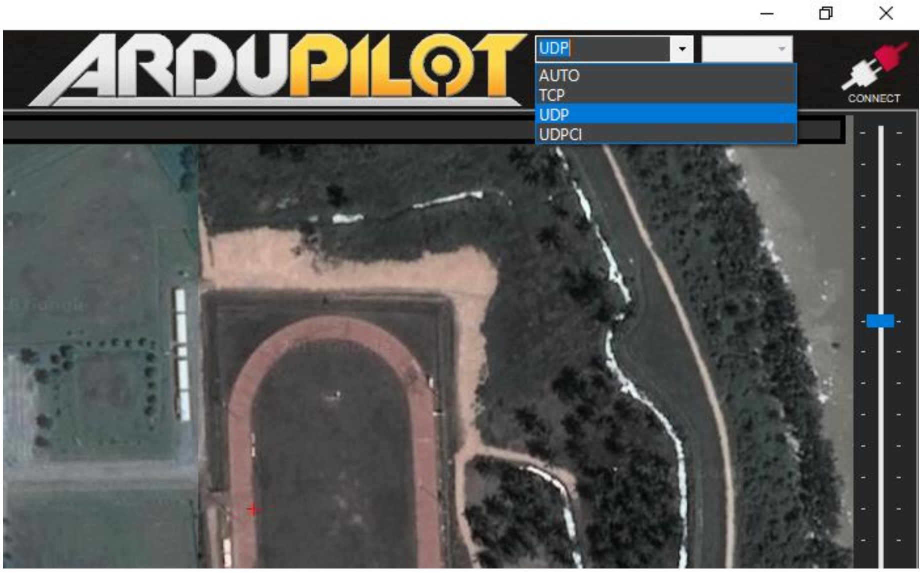
Again, press ctrl-X to leave and ‘Y’ followed by enter to save the changes. Then, reboot your Raspberry Pi through ‘sudo reboot’.

The GCS software used was Mission Planner by Michael Oborne which supports a variety of functions as discussed through easy-to-use interfaces. Download the latest version of Mission Planner from

http://ardupilot.org/planner/docs/common-install-mission-planner.html

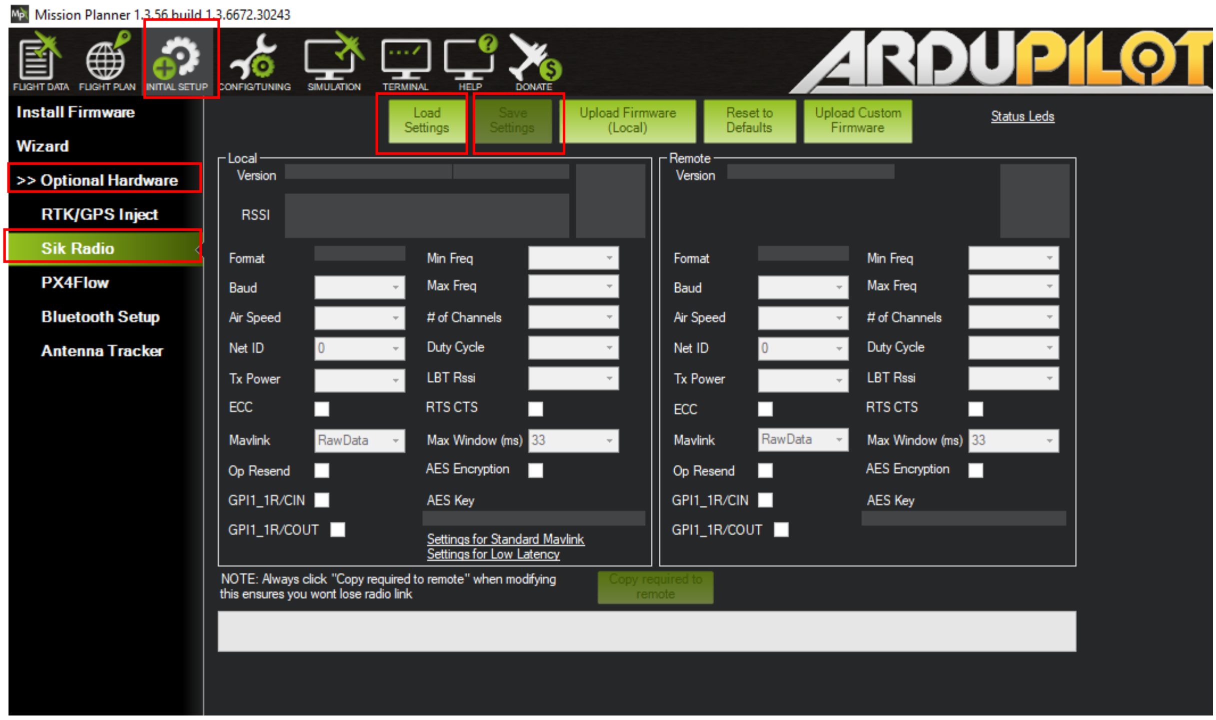


If you are using a common LAN for GCS-navio2 communication, select on the top right corner (after you editted the telemetry network configuration address) UDP connection and hit ‘connect’. The baud rate will automatically show, and parameters will be loaded soon. Changes in the values of flight data on the bottom left in Mission Planner will a signify successful connection. If mavlink connection takes too long – after the 30-second counting down – consider restarting your Raspberry Pi, through sudo reboot on PuTTY, or directly unplug and reconnect supply to the Raspberry Pi. Note that it will never connect if your IP address of your network have changed but you have not updated the telemetry network configuration in the above step. Check out the IPV4 address at your network and sharing centre!



If you are connecting using a telemetry module pair, plug in the air module to the UART port on navio2 and the ground module to your laptop through USB. Wait until the LEDs on both modules stop flashing (i.e. emitting a solid light), go to Initial setup > Optional hardware > Sik Radio on Mission Planner. To do this you will not have to establish udp or any connections before that. Then, press the button “Load setting” and wait as the parameters in the blank spaces are automatically loaded. Save the setting after process is complete. Then go to the other tab and come back to see if the paramteres are still there. If this fails, try to reboot the RPi and attempt several times.

Please note that all hardware connections are shown in the “drone hardware setup” section.



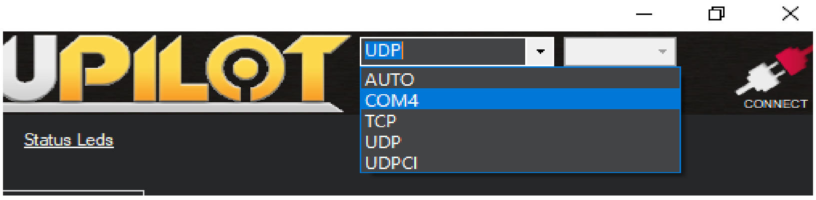
Net ID: 0

Mavlink: RawData

Max Window (ms): 33

NOTE: Always click "Copy required to remote" when modifying this ensures you wont lose radio link

Then, change the connection to COM4 (the baud rate will also be automatically loaded), and then hit ‘connect’. Telemetry connection usually takes longer time to load parameters but should also show responses within seconds. Again, retry again after rebooting the RPi if this step fails.



COM4 57600

Getting Params 1

Got param SR2_PARAMS

NOTE: Always click "Copy required to remote" when modifying this ensures you wont lose radio link

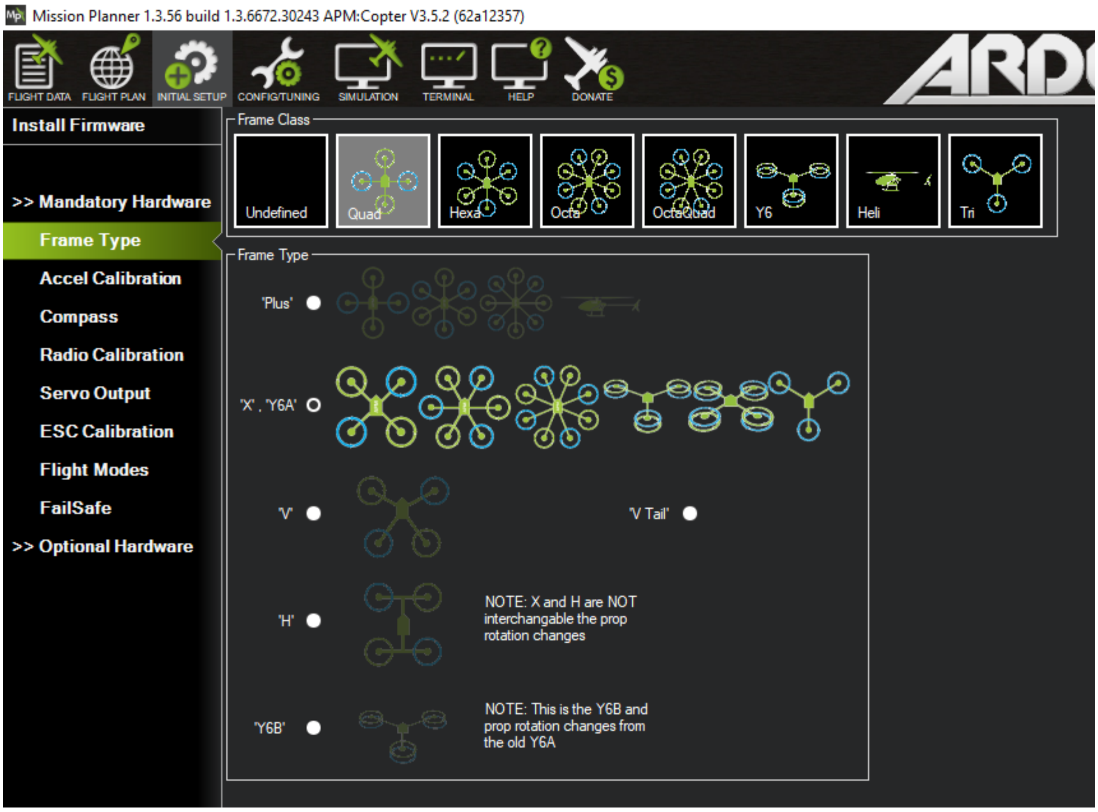
Flight calibration on Mission Planner (MP)

If we connect the power of the Raspberry Pi through the Power Module from the battery, instead of the micro-USB cable, the esc will beep continuously signifying the setup procedure is yet to complete and the esc does not recognise the flight controller. Only after all calibrations are done, the esc beeping will stop.

Flight calibration is a crucial and compulsory step to complete before the quadcopter can be armed for a flight. To proceed we first connect the Mission Planner GCS to the drone from the previous step, through udp connection, or through a pair of telemetry module (recommended).

It is not necessary to follow through the steps in the setup wizard, that is designed to guide new users through the calibration procedures. The calibration steps are as follows:

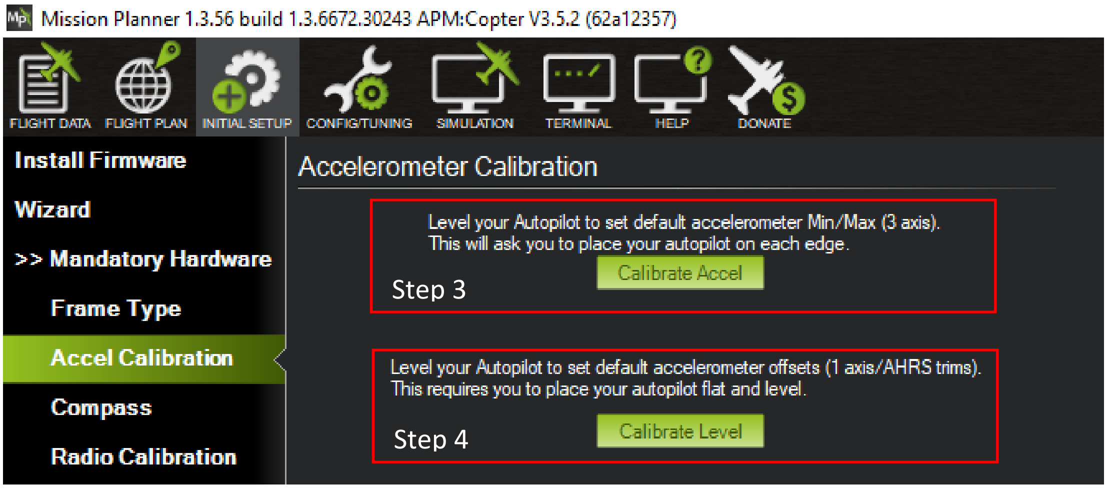
1. Go to Initial Setup tab > Mandatory Hardware > Frame type
2. The quadcopter airframe type – X-frame – is selected.



NOTE: X and H are NOT interchangable the prop rotation changes

NOTE: This is the Y6B and prop rotation changes from the old Y6A

3. Perform accelerometer calibration. Simply place the drone on flat level and then hit "Calibrate Accel".



Level your Autopilot to set default accelerometer Min/Max (3 axis).
This will ask you to place your autopilot on each edge.

Step 3

Level your Autopilot to set default accelerometer offsets (1 axis/AHRS trims).
This requires you to place your autopilot flat and level.

Step 4

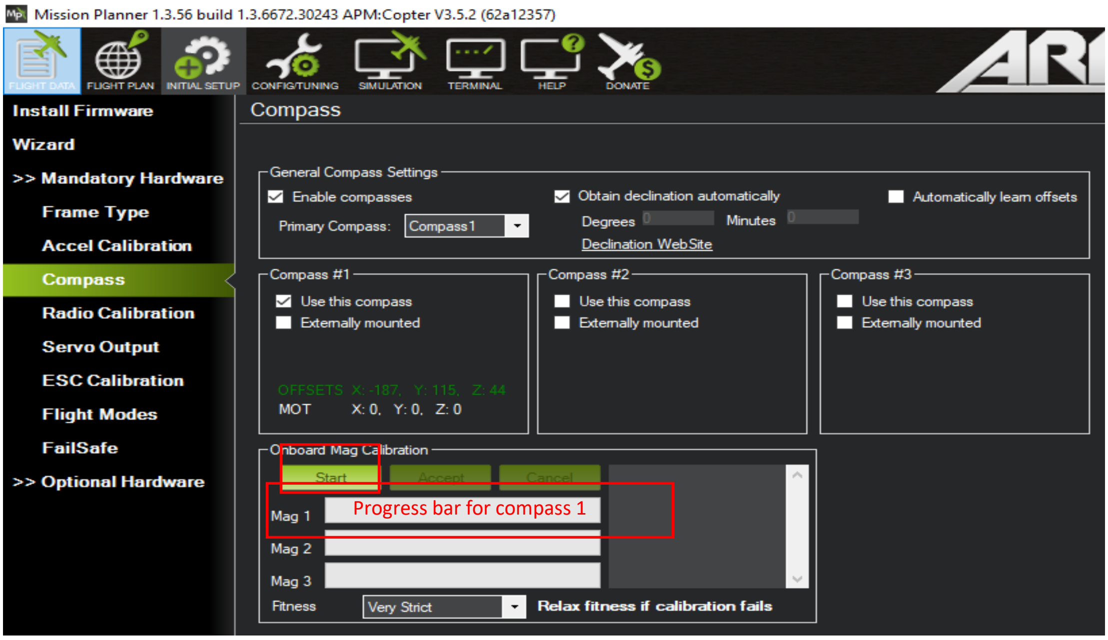
4. The quadcopter is then placed still on a flat surface for level calibration. Different orientations of tilting, as instructed: on level, on its right, on its left, nose up, nose down and on its back, is calibrated. Press the button after each position is properly adjusted. As mentioned, this calibration process is much simpler done when your RPi is connected wirelessly to the GCS through a telemetry and has power supplied through a battery, instead of using LAN cable or power from MicroUSB cable because otherwise movements of the quadcopter will be restricted by the cables.

Different positions of drone in accelerometer calibration from the Arducopter site

This step overrides the current state of the quadcopter as level and in different positions, and can be done everytime you see from the flight data that the quadcopter is not levelled properly.

Note: Telemetry's connection requires one-time configuration through Sik Radio connection in optional hardware in the Initial Setup tab, by loading the settings after both the air and ground modules' LED are in green, solid emitted state.

5. Magnetometer or compass calibration is required for the accuracy of the quadcopter's headings, and is performed by moving the quadcopter to cover all directions of the 3-axis planes. To start this, simply hit the start button as shown in the figure below. Calibration will begin collecting required sensor data and the progress of the calibration of the selected compass is shown in the green progress bar.



Compass

General Compass Settings

☑ Enable compasses

Primary Compass: Compass1

☑ Obtain declination automatically

Degrees 0 Minutes 0

☐ Automatically learn offsets

Compass #1

☑ Use this compass

☐ Externally mounted

OFFSETS X: -187, Y: 115, Z: 44

MOT X: 0, Y: 0, Z: 0

Compass #2

☐ Use this compass

☐ Externally mounted

Compass #3

☐ Use this compass

☐ Externally mounted

Onboard Mag Calibration

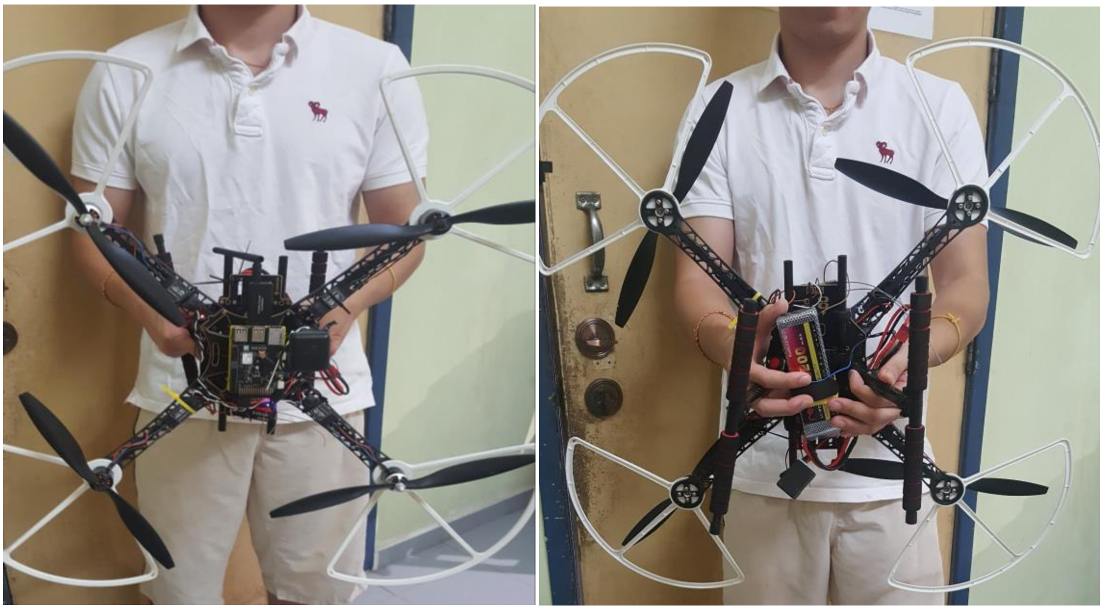
Mag 1 Progress bar for compass 1

Mag 2

Mag 3

Fitness Very Strict

To calibrate the compass, turn the drone's top toward you and move it in circle like how you would steer a car to exceed one cycle. Then repeat the actions with the drone's top facing away from you. Then, rotate the drone on its level then rotate again with it upside down. The video demonstration by Ardupilot on magnetometer calibration is as: https://youtu.be/DmsueBS0J3E

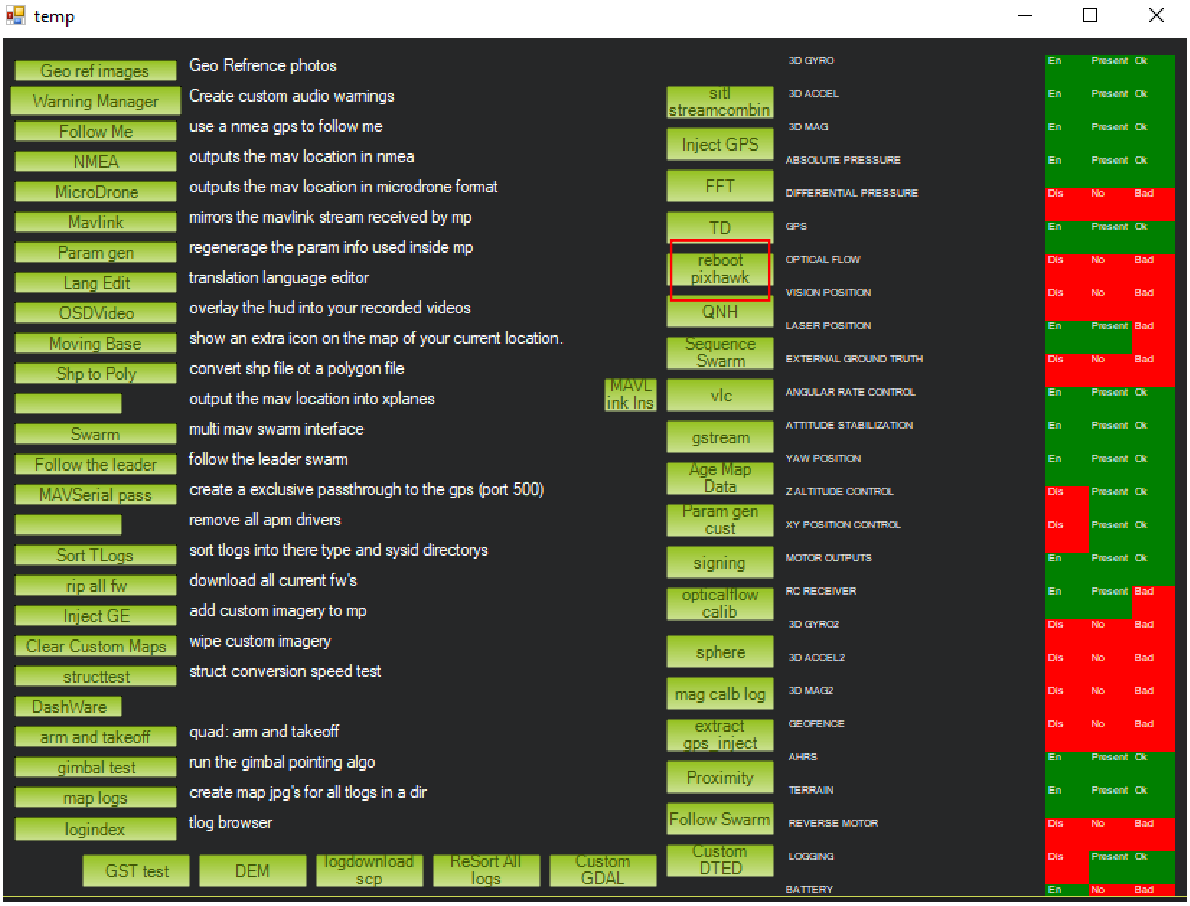


"Steering" a drone in different directions

By the end the progress bar should have reached its end and a message to tell completion and prompt restart will be shown.

There are 2 built-in compasses in the navio2 flight controller. If both compass 1 and 2 are used (checked). They both have to be calibrated. If any of the compasses keeps reload (the green progress bar keeps restart again after it progresses to the end repeatedly) when being calibrated, uncheck that problematic compass, i.e. use only compass 1 if compass 2's progress keeps reloading and use compass 2 if compass 1's progress keeps reload.

Don't use the setup wizard if there is problem reccurring in the magnetometer calibration. After the calibration is complete, press Ctrl+F to open 'temps' and reboot pixhawk as shown below.

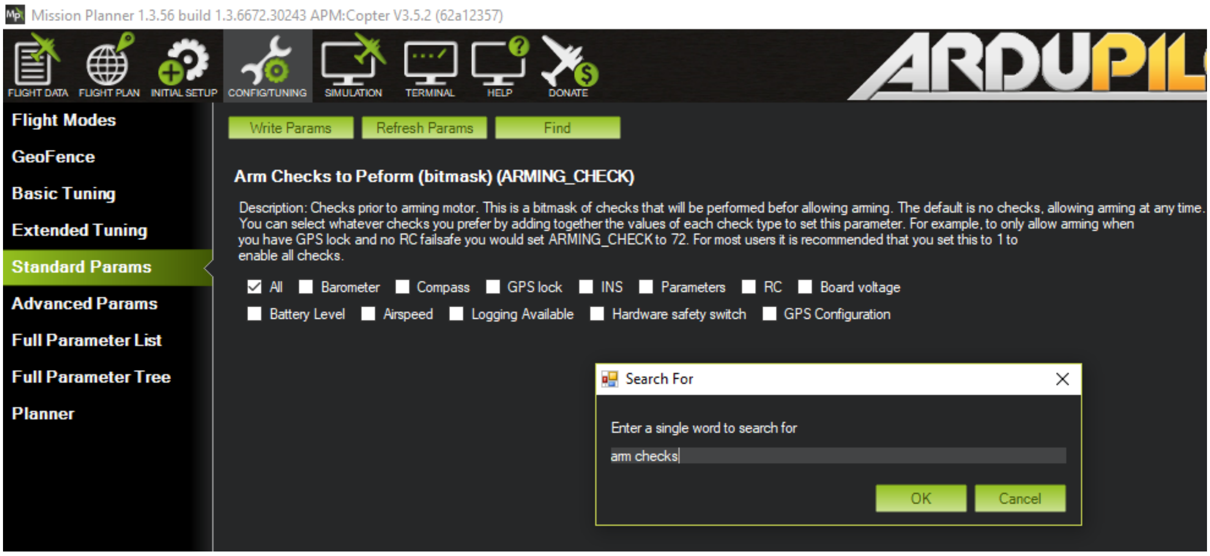


Geo ref images	Geo Refrence photos
Warning Manager	Create custom audio warnings
Follow Me	use a nmea gps to follow me
NMEA	outputs the mav location in nmea
MicroDrone	outputs the mav location in microdrone format
Mavlink	mirrors the mavlink stream received by mp
Param gen	regenerage the param info used inside mp
Lang Edit	translation language editor
OSDVideo	overlay the hud into your recorded videos
Moving Base	show an extra icon on the map of your current location.
Shp to Poly	convert shp file ot a polygon file
	output the mav location into xplanes
Swarm	multi mav swarm interface
Follow the leader	follow the leader swarm
MAVSerial pass	create a exclusive passthrough to the gps (port 500)
	remove all apm drivers
Sort TLogs	sort tlogs into there type and sysid directorys
rip all fw	download all current fw's
Inject GE	add custom imagery to mp
Clear Custom Maps	wipe custom imagery
structtest	struct conversion speed test
DashWare	
arm and takeoff	quad: arm and takeoff
gimbal test	run the gimbal pointing algo
map logs	create map jpg's for all tlogs in a dir
logindex	tlog browser

3D GYRO	En	Present	Ok
3D ACCEL	En	Present	Ok
3D MAG	En	Present	Ok
ABSOLUTE PRESSURE	En	Present	Ok
DIFFERENTIAL PRESSURE	Dis	No	Bad
GPS	En	Present	Ok
OPTICAL FLOW	Dis	No	Bad
VISION POSITION	Dis	No	Bad
LASER POSITION	En	Present	Bad
EXTERNAL GROUND TRUTH	Dis	No	Bad
ANGULAR RATE CONTROL	En	Present	Ok
ATTITUDE STABILIZATION	En	Present	Ok
YAW POSITION	En	Present	Ok
Z ALTITUDE CONTROL	Dis	Present	Ok
XY POSITION CONTROL	Dis	Present	Ok
MOTOR OUTPUTS	En	Present	Ok
RC RECEIVER	En	Present	Bad
3D GYRO2	Dis	No	Bad
3D ACCEL2	Dis	No	Bad
3D MAG2	Dis	No	Bad
GEOFENCE	Dis	No	Bad
AHRS	En	Present	Ok
TERRAIN	En	Present	Ok
REVERSE MOTOR	Dis	No	Bad
LOGGING	Dis	Present	Ok
BATTERY	En	No	Bad

After rebooting, in order to see the newly calibrated compass offset value, click to go to any other tabs and then back to the compass calibration tab. The offset values are prefereably less than 150 for all directions to be considered successful (values are shown in green). If the values are more than 150 (in yellow or in red), redo the calibration. Offsets value that are too high may cause the quadocopter to fail the pre-arm check list.

If several attempts of recalibration turns out similar results, and have ruled out interference of surrounding metal objects, uncheck the compass from the pre-arm check list if you are restricted from arming your quadcopter due to “compass offset too high” after all the calibrations are completed. This is done by going to the Configuring tab > Standard Params > click the ‘Find’ button and search for arm checks. Uncheck ‘All’ and then individually reselect all but compass and write (save) the parameters. Sometimes the failure to arm a quadcopter due to compass offset disappears even after you re-check compass as a component of the pre-arm check requirement.



Arm Checks to Peform (bitmask) (ARMING_CHECK)

Description: Checks prior to arming motor. This is a bitmask of checks that will be performed befor allowing arming. The default is no checks, allowing arming at any time. You can select whatever checks you prefer by adding together the values of each check type to set this parameter. For example, to only allow arming when you have GPS lock and no RC failsafe you would set ARMING_CHECK to 72. For most users it is recommended that you set this to 1 to enable all checks.

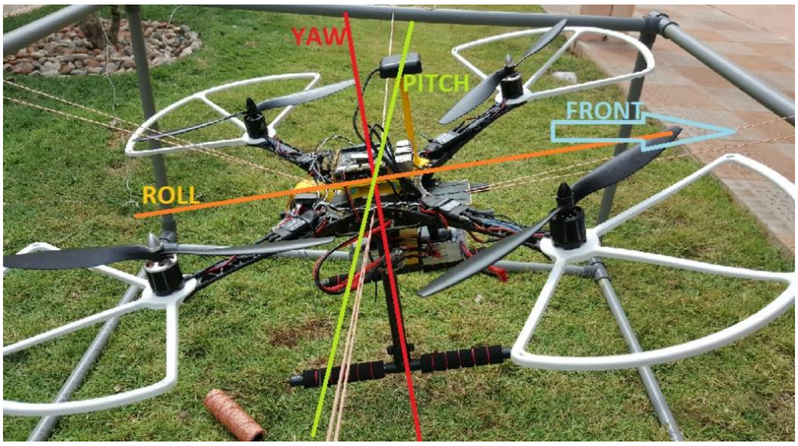
☑ All ☐ Barometer ☐ Compass ☐ GPS lock ☐ INS ☐ Parameters ☐ RC ☐ Board voltage
☐ Battery Level ☐ Airspeed ☐ Logging Available ☐ Hardware safety switch ☐ GPS Configuration

Enter a single word to search for
arm checks

6. Radio calibration was carried out to let the Flight controller (FC) know the range of PWM that each channel of radio is sent from the transmitter to the receiver before the FC can translate them into flight control actions. To begin with the calibration, hit the button “Calibrate Radio”. Then, move the left and right sticks on your trnsmitter in all directions while checking that each of them is responding to the right channel as shown below. The left stick controls the yaw (left and right) and throttle (up and down), while the right stick controls roll (left and right) and pitch (up and down). Bring each stick to their 2 extreme ends to get the full range of their PWM value recorded. As the stick is pushed to its extremes, and the red markers that specify the range of radio PWM will expand to the side. If the stick does not respond to the right channel as discussed, modification on the transmitter has to be made, but the transmitter usually would only allow modification on channels other than the first 4, which are fixed for

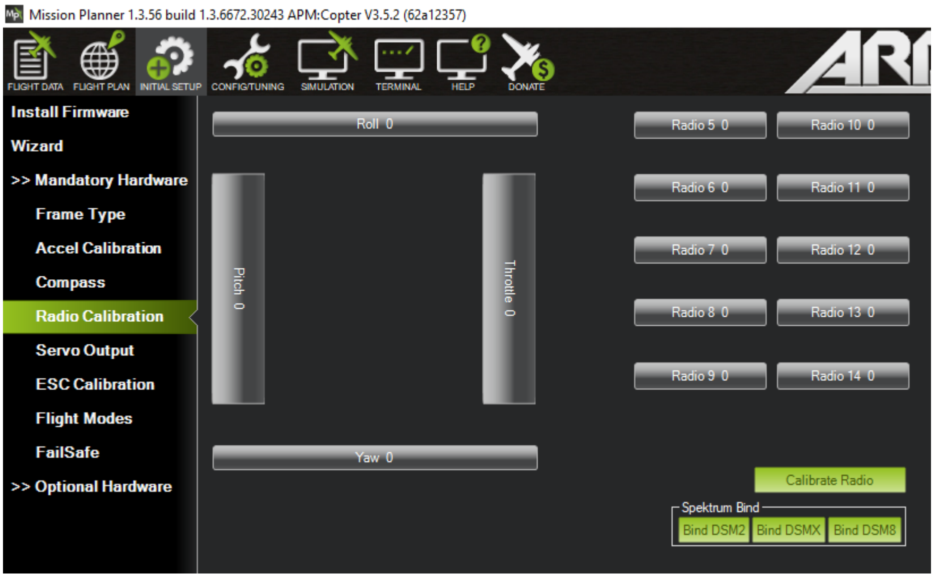
   a. Channel 1: Roll
   b. Channel 2: Pitch
   c. Channel 3: Throttle
   d. Channel 4: Yaw

Note: Please make sure that the transmitter vehicle setting is copter and not other types of vehicles.



If the direction of the response of any radio channel is inverted, reverse the channel signal either through setting on the transmitter or the Mission Planner. For example, if the pitch is shown to be positive in the radio configuration when you push the right stick up, that radio channel can be reversed. After that, all available channels have their radio calibrated by throwing each switch into different directions.

Transmitter sticks and their actions



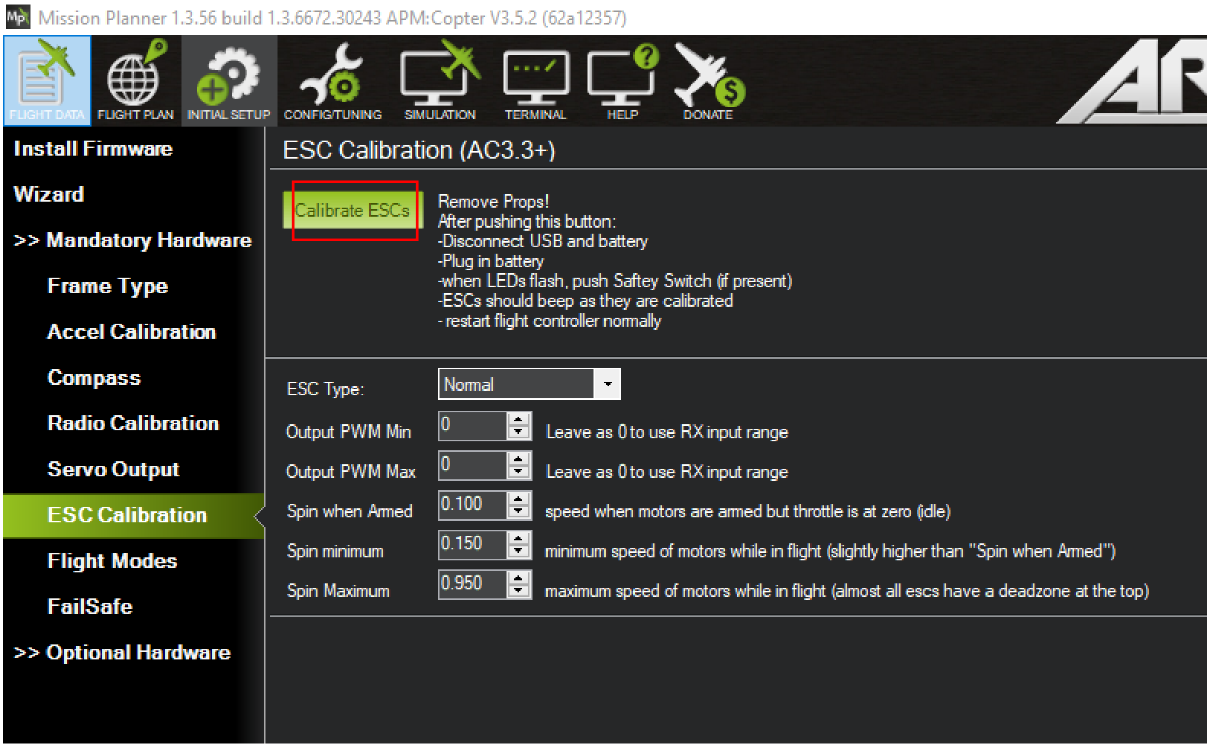
Mission Planner 1.3.56 build 1.3.6672.30243 APM:Copter V3.5.2 (62a12357)

Mission Planner 1.3.56 build 1.3.6672.30243 APM:Copter V3.5.2 (62a12357)

#	Position	Reverse	Function	Min	Trim	Max
1	1065	☐	Motor1	1100	1500	1900
2	1065	☐	Motor2	1100	1500	1900
3	1065	☐	Motor3	1100	1500	1900
4	1065	☐	Motor4	1100	1500	1900
5	0	☐	Disabled	1100	1500	1900
6	0	☐	Disabled	1100	1500	1900
7	0	☐	Disabled	1100	1500	1900
8	0	☐	Disabled	1100	1500	1900

Motor and ESC calibration is performed for better power delivery and motor rotation.

7. To perform ESC calibration, hit the calibrate ESC button. With the transmitter throttle stick (left) fully pushed up, start your flight controller. Then, after a while, pull the throttle down. The ESC calibration will complete after the long beeping sound stops. If the beeping persists forever, give up the calibration and restart your quadcopter normally with your throttle at minimum. Some ESC calibration will be performed every time they are first connected to the power supply, and is marked by long beeping sounds with the FC LED blinking in read and blue.



Mission Planner 1.3.56 build 1.3.6672.30243 APM:Copter V3.5.2 (62a12357)

ESC Calibration (AC3.3+)

Remove Props!
After pushing this button:
-Disconnect USB and battery
-Plug in battery
-when LEDs flash, push Saftey Switch (if present)
-ESCs should beep as they are calibrated
- restart flight controller normally

ESC Type: Normal
Output PWM Min 0 Leave as 0 to use RX input range
Output PWM Max 0 Leave as 0 to use RX input range
Spin when Armed 0.100 speed when motors are armed but throttle is at zero (idle)
Spin minimum 0.150 minimum speed of motors while in flight (slightly higher than "Spin when Armed")
Spin Maximum 0.950 maximum speed of motors while in flight (almost all escs have a deadzone at the top)

After all calibration the beeping sound is gone. If not then the calibration steps may not be complete. Follow through all the wizard just to make sure you covered all the mandatory calibration and setups. Check also if the selected frame type is correct.

8. Failsafe

Mission Planner 1.3.56 build 1.3.6672.30243 APM:Copter V3.5.2 (62a12357)

Stabilize
Disarmed
GPS: No Fix

Battery
Low Battery 10.5
Reserved MAH 0
Disabled

Radio
Enabled always RTL
FS Pwm 975

GCS
☑ GCS FS Enable

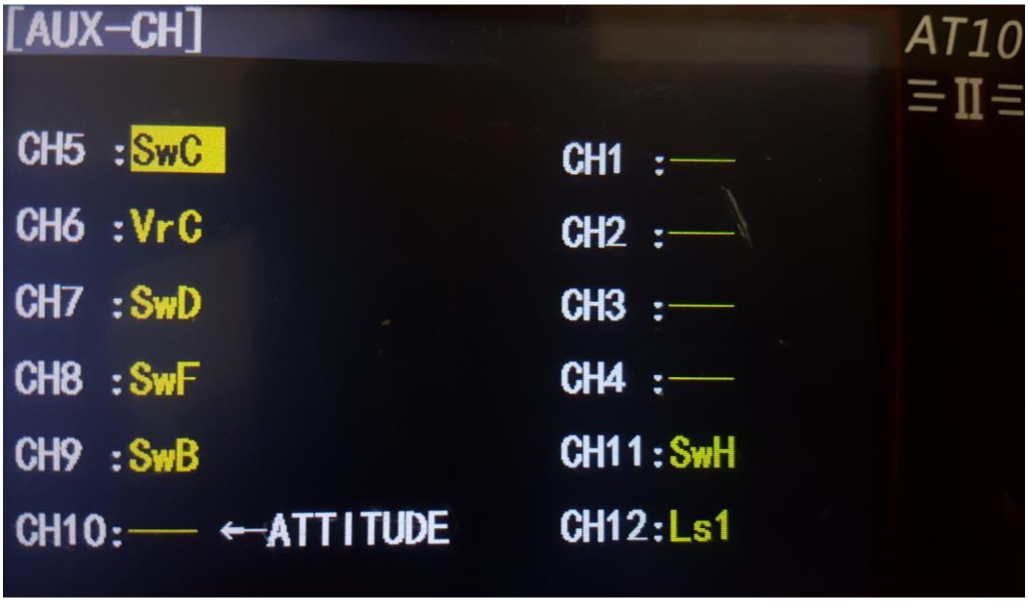
It is recommended to set failsafe for the quadcopter to land at low board voltage (unless the board voltage measurement by the FC is not accurate. It is easier and more convenient to use a BB alarm instead. The BB alarm is connected to the Li-Po battery to trigger an alarm whenever the battery voltage drops below a specified limit. The battery voltage below which the alarm is triggered can be set easily through a simple button push that can be found on the BB alarm. It is safe to set it at 3.2-3.4V where when either one of the cells drops below the threshold, you will be alarmed to land your quadcopter.

Radio communication is a crucial component whose failure can cause your quadcopter to be gone forever. Do not use components that can have interference with the radio channel frequency and always check the transmitter voltage and functionality. Safeguard your quadcopter by setting a failsafe for the radio communication by which the quadcopter will Return to Launch (RTL) whenever the radio communication is lost.

The quadcopter can only be armed after all the safety pre-arm checks are completed, which also includes all of the above flight calibration steps.

Change the flight mode on mission planner with respect to radio transmitter channel

As aforementioned, the primary channels (channel 1 to 4) are fixed for the specific type of vehicle selected. The Auxiliary channel on the other hand, can be altered according to our needs. Different transmitters have different methods of setting for the auxiliary channel. Each channel is controlled by a switch on the transmitter. All we need to do is to change the channel's switch to the specific switch that we want to use to control the specific channel.



[AUX-CH]

CH5 :SwC
CH6 :VrC
CH7 :SwD
CH8 :SwF
CH9 :SwB
CH10:—— ←ATTITUDE

CH1 :——
CH2 :——
CH3 :——
CH4 :——
CH11:SwH
CH12:Ls1

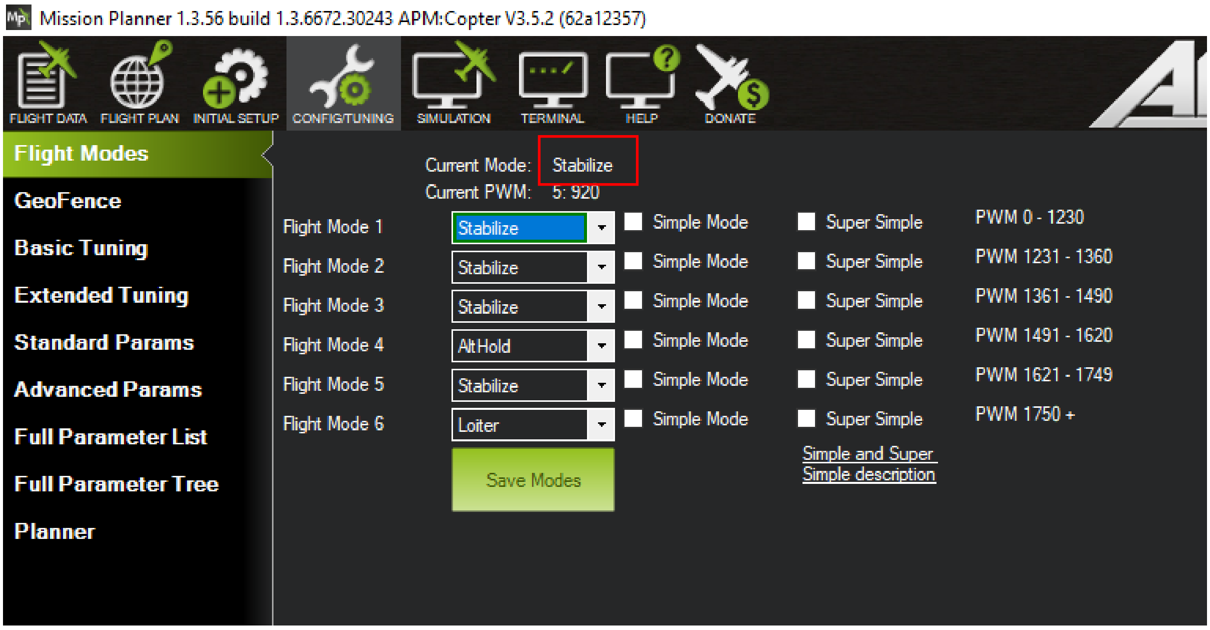
Switch selection for auxiliary channels on a transmitter

The radio channel setting on your radio transmitter has to be collaborated with the setting on the Mission Planning, and the actual mode that your quadcopter holds to will be as shown in the Mission Planner (not as shown in your Radio transmitter).

To go to the flight mode setting on Mission Planner, connect your GCS to your quadcopter (through udp or telemetry), then go to Configuring > Flight Modes. The flight mode setting in the Mission Planner uses fixed and default channel 5 from your radio. Our aim here is we want to utilise most out of the 6 flight modes switching offered by Mission Planner across the PWM range in channel 5.

Therefore, on your radio transmitter, change the auxiliary channel, Channel 5, to be controlled by a suitable 3-way switch (if you have one). Throw your switch to different positions to test which flight modes that were being engaged at different positions. My 3-way switch at channel 5 engages Flight Mode 1 (when switch position is at Bottom), Flight Mode 4 (when switch position is Mid) and Flight Mode 6 (when switch position is at top). Now change on Mission Planner the by selecting from the drop-down list the modes that we want the quadcopter to engage, i.e. for my case, selecting the modes for flight modes 1, 4 and 6.

Now try to look at the current mode shown to see the changes as you throw your switch across different positions. In my case, for example, the current mode will change from stabilise (bottom) to Altitude Hold (Mid) and Loiter (Top).



Current Mode: Stabilize
Current PWM: 5: 920

Flight Mode 1	Stabilize	☐ Simple Mode	☐ Super Simple	PWM 0 - 1230
Flight Mode 2	Stabilize	☐ Simple Mode	☐ Super Simple	PWM 1231 - 1360
Flight Mode 3	Stabilize	☐ Simple Mode	☐ Super Simple	PWM 1361 - 1490
Flight Mode 4	AltHold	☐ Simple Mode	☐ Super Simple	PWM 1491 - 1620
Flight Mode 5	Stabilize	☐ Simple Mode	☐ Super Simple	PWM 1621 - 1749
Flight Mode 6	Loiter	☐ Simple Mode	☐ Super Simple	PWM 1750 +

Need not worry if your radio transmitter only has 2-way switches. Use the 2 Flight Modes for the mode you think is most important for you. I would suggest Stabilise (the basic one), and Altitude Hold (Important if you are not pro at controlling throttle).

Landing a quadcopter is equally challenging and you may want to include that as well. I would suggest Land mode to be set from a separate channel so you can leave channel 5 alone for other modes. Simply go to Configuring > Extended Tuning for the mode settings for other channels

Stabilize Roll (Error to Rate): P 4.500
Stabilize Pitch (Error to Rate): P 4.500
Stabilize Yaw (Error to Rate): P 4.500
Position XY (Dist to Speed): P 1.000

☐ Lock Pitch and Roll Values

Rate Roll: P 0.300, I 0.090, D 0.004, IMAX 0.5, FILT 20.000
Rate Pitch: P 0.135, I 0.090, D 0.0036, IMAX 0.5, FILT 20.000
Rate Yaw: P 0.180, I 0.018, D 0.000, IMAX 0.5, FILT 2.500
Velocity XY (Vel to Accel): P 1.000, I 0.500, D 0.000, IMAX 100

Throttle Accel (Accel to motor): P 0.500, I 1.000, D 0.000, IMAX 80
Throttle Rate (VSpd to accel): P 5.000
Altitude Hold (Alt to climbrate): P 1.000
WPNav (cm's): Speed 400.000, Radius 200.000, Speed Up 250.000, Speed Dn 150.000, Loiter Speed 500.000

Ch6 Opt: None
Min 0.000 1.000
Ch7 Opt: Auto
Ch8 Opt: Land

Again, match the setting on your radio transmitter with that on Mission Planner. For example, we first made sure that we set channel 8 to be controlled by a switch (2-way) that we want to use to Land the quadcopter, and then on the Mission Planner, I set channel 8 Opt as Land. Land will be a prioritised mode over all others. This means that if my channel 5 is set as stabilise (say with switch C at bottom), and my channel 8 is set to land (say with switch F on), the quadcopter will go for Land. **Always note that the actual mode that the quadcopter engages is shown in the Current Mode shown in Configuration > Flight Modes**. From there, always test out different switching of the switches you have set to see if the quadcopter really engage the modes that you desire at all times. As shown in the above diagram, there are 2 more channels from channel 6 and channel 7 where you can use to engage other desired flight modes.

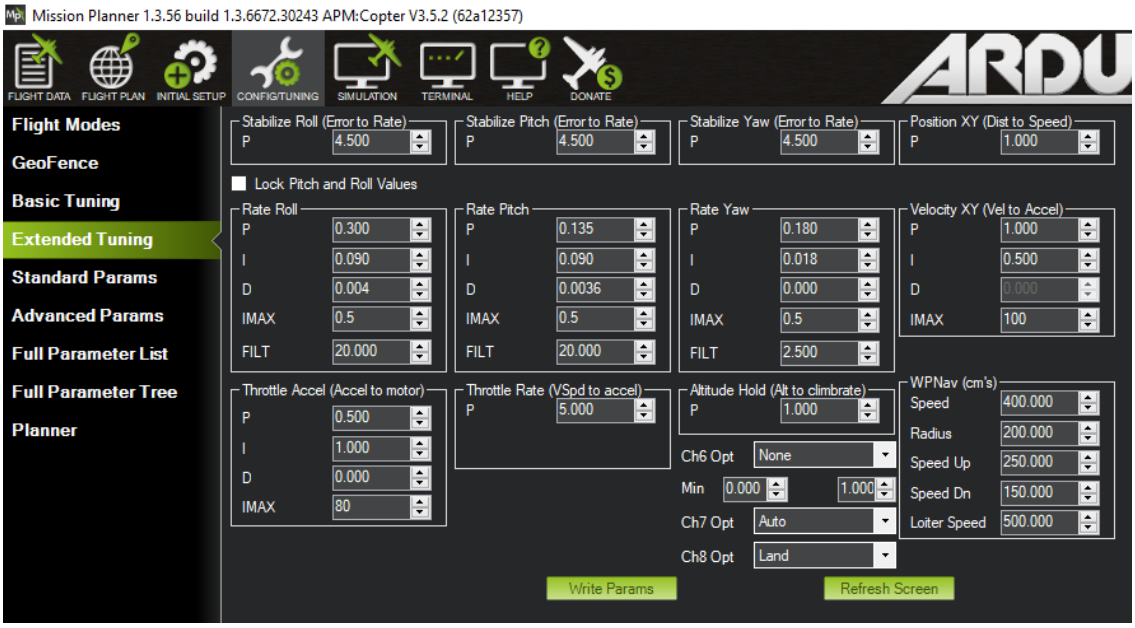
The PWM value of the channel at a particular switch position is shown below the flight mode. It is not recommended to use a knob you are confident because it is hard to gauge the position and the accurate PWM output.

Some radio transmitter allow logic control of switches to control a single channel. This means that 2 switches can be used to control, say channel 5 for more outputs. An example is shown below just for better illustration. This setting can be done on your radio transmitter (if it has this feature) under logic switch which may be different for every transmitter. Spend some time to look through how to perform the setting.

Swicth C	Switch D	PWM output
Bottom	Up	500
Bottom	Down	1200
Mid	Up	1300
Mid	Down	1400
Up	Up	1500
Up	Down	1700

However, it is not recommended to use logic switch unless you are confident because it might easily cause confusion due to the complexity of switching and can be dangerous at times of panick. To be on the safe side, always use a single 3-way switch and make clear labels on your radio transmitters for the different modes that the different switches will engage. This helps you to be able to respond immediately to emergency cases such as land the quadcopter or disengage a failing mode, than having to figure out which switch to control.

Transmitter labelled with flight modes at switches



Mission Planner 1.3.56 build 1.3.6672.30243 APM:Copter V3.5.2 (62a12357)

Stabilize Roll (Error to Rate): P 4.500

Stabilize Pitch (Error to Rate): P 4.500

Stabilize Yaw (Error to Rate): P 4.500

Position XY (Dist to Speed): P 1.000

☐ Lock Pitch and Roll Values

Rate Roll: P 0.300, I 0.090, D 0.004, IMAX 0.5, FILT 20.000

Rate Pitch: P 0.135, I 0.090, D 0.0036, IMAX 0.5, FILT 20.000

Rate Yaw: P 0.180, I 0.018, D 0.000, IMAX 0.5, FILT 2.500

Velocity XY (Vel to Accel): P 1.000, I 0.500, D 0.000, IMAX 100

Throttle Accel (Accel to motor): P 0.500, I 1.000, D 0.000, IMAX 80

Throttle Rate (VSpd to accel): P 5.000

Altitude Hold (Alt to climbrate): P 1.000

Ch6 Opt: None

Min 0.000 1.000

Ch7 Opt: Auto

Ch8 Opt: Land

WPNav (cm's): Speed 400.000, Radius 200.000, Speed Up 250.000, Speed Dn 150.000, Loiter Speed 500.000

Below shows the summary of my switch setting as an example for your reference based on the figure on MP (extended tuning) above. The 3 fundamental modes that you may consider setting are Land, Stabilise and Altitude hold.

Mode	Channel (set on MP)	Switch (set on transmitter)
Land	8 (selected from Extended Tuning)	F (2-way)
Stabilise	5 (selected from Flight Modes)	C (3-way)
Altitude Hold	5 (selected from Flight Modes)	C (3-way)
Loiter	5 (selected from Flight Modes)	C (3-way)
Auto	7 (selected from Extended Tuning)	D (2-way)

After the flight mode is set on both your transmitter and your Mission Planner (connected to your drone), you will not require connection to the MP anymore to start the flight, unless you want to monitor your flight. However, make sure you are clear of the modes you are switching into!

Summary of basic flight modes

Flight Modes	Functions
Stabilise	Allow manual control of vehicle with self-levelled roll and pitch axis. Constant adjustments of the throttle is required to maintain altitude.
Alt Hold	Copter maintains at a consistent altitude while allowing roll, pitch and yaw axes control. The vehicle will maintain at current altitude if the throttle is in the 40-60% deadzone, and will only descend or climb if the throttle is out of the deadzone. The deadband can be adjusted 0-400, with 100 means 10% above and 10% below mid throttle.
Loiter	Loiter mode maintains the location, heading and altitude. Sticks control can interrupt the mode but the copter will stop and hold as soon as the sticks are released. Altitude control is the same as that in Alt Hold mode. Loiter mode requires minimal vibrations, good compass and GPS lock functions.
Return to Launch (RTL)	When RTL mode is engaged, the copter will return to, and hover above the home position. The copter will first rise to RTL_ALT, an altitude defaulted at 15m, or maintain at any altitude higher than 15m, before returning to launch. Armed position is the home position of the copter.
Auto	Auto Mode makes the copter follow a pre-programmed mission script stored in the autopilot. The script is made up of navigational commands called waypoints. Missions should end with RTL or LAND, otherwise manual control has to take over. The speed (WPNAV_SPEED) can be adjusted.
Acro Mode	Acro Mode is useful in aerobatics, and is the hardest mode to master. RC sticks are used to control the angular velocity of the copter. Releasing the sticks will maintain its last attitude, i.e. the copter will not level.
Auto Tune	AutoTune sees the copter “twitching” in the roll and pitch axes in Alt Hold mode, to automatically tune the PID. Combination of axes (AUTOTUNE_AXES) and aggressiveness (AUTOTUNE_AGGR) of tuning can be adjusted. AutoTune should be performed where twitching is perpendicular to wind direction.
Brake	This mode stops the copter as soon as possible, using GPS lock. Once invoked, the copter will not accept any inputs from the pilot.
Circle	With the copter’s nose pointing toward the centre, it will circle around a point with radius equals to CICLE_RADIUS centimeters. Setting the value to zero is useful for panaroma shots. The speed can be adjusted at CIRCLE_RATE. The roll and pitch cannot be controlled manually. Only the throttle for altitude and yaw can be adjusted.
Drift	Drift mode allows the copter to fly as if it were a plane with built in automatic coordinated turns. The throttle is manually controlled as in stabilise mode.
Guided Mode	Guided mode enables the guidance of copter through a GCS wirelessly using a telemetry. The copter will hover after arriving at the

	commanded target location, awaiting next commands. To use this mode, the copter shall take off in stabilise mode and switched to Loiter mode at reasonable altitude.
Land	Land mode brings the copter down at 150cm/s above 10 m, and 50cm/s below 10 m. If the throttle is minimum, the copter will be disarmed upon reaching the ground. Land speed can adjusted in configuration tab, in the full parameter list. Roll and pitch is controlled with GPS lock during landing.
Position Hold	Position hold is similar to Loiter where it holds position, heading and altitude, except the pilot sticks can directly control the copter's lean angle giving a more natural feel.
Sports	Known as the "rate controlled stabilize" plus Altitude Hold. When the control sticks are released the vehicle will remain in its current attitude. The maximum lean angle (ANGLE_MAX) can be adjusted.
Throw	This risky mode allows the copter to start mid-air when it is thrown up in trajectory. This mode requires GPS, and will not accept any inputs from the pilot.
Follow me mode (GCS enabled)	This mode allows the copter to follow the pilot with a telemetry radio and a ground station. This mode works with Loiter mode after taking off to a safe altitude.
Simple and super simple modes	These modes allow the pilot to control the movement of the copter from the pilot's point of view regardless of which way the copter is facing. Simple Mode allows you to control the copter relative to the copters heading at take off. Super Simple Mode allows you to control the copter relative to its direction from home (i.e. that is the pilots location where it is armed)
Smart RTL	Smart RTL is the latest added feature that is similar to RTL where the copter returns to its armed location. The copter records the manual flight which brought it to a location, prune out circles and loops, avoid obstacles to fly in a clean and efficient path.

Installing a turn-off switch to the RPi3

To turn off the Arducopter, a sudo halt command is required. Similar to a PC, no proper shutting down of devices may corrupt the storage in the long run. Therefore, a switch is installed and the program for the switch operation to turn the device off is being executed on start of the RPi. The switch installation details were being discussed in the hardware component setup section.

The Raspberry Pi starts booting as soon as it receives power supply from the battery. However, there is slightly more trouble when it comes to shutting it down appropriately.

We all know that appropriate shutdowns of the Raspberry Pi microcomputer can be as important as shutting down your PCs appropriately. The shutting down can only be executed after logging into Raspberry Pi, either through proper connections with keyboard or mouse and monitor, or secure shell (SSH) to laptop via PuTTY. Forced shutting down by unplugging the supply may result in SD card corruption in the long run.

Therefore, one of the free GPIOs of the Raspberry Pi – that is not used up by the Navio2 Flight controller – is used to install a turn-off switch for the Raspberry Pi. Only 3 free GPIOs can be used, that is GPIO 17 (Pin 11), GPIO18 (Pin 12), and GPIO 26 (Pin 37). The idea is to write a program that runs on boot in the Raspberry Pi, that whenever GPIO 17 (Pin 11), for example, is shorted to the ground (through a momentary push button switch), the Raspberry Pi will run the command to shut itself down automatically.

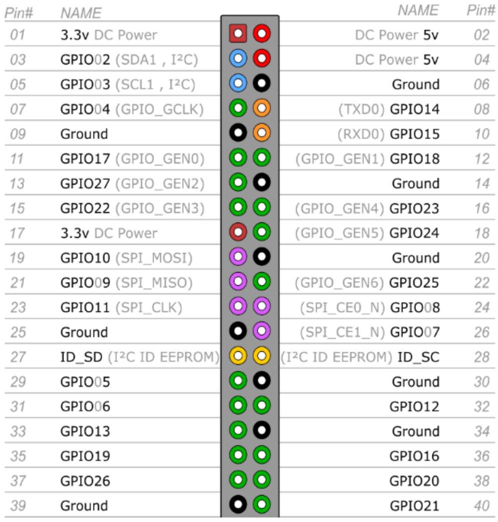
To add a turn-off button to the Raspberry Pi, simply follow through the steps in the following link by ETA Prime. It is very straightforward and easy for everyone.

https://www.youtube.com/watch?v=4nTuzIY0i3k

The RetroPie text file can be downloaded from

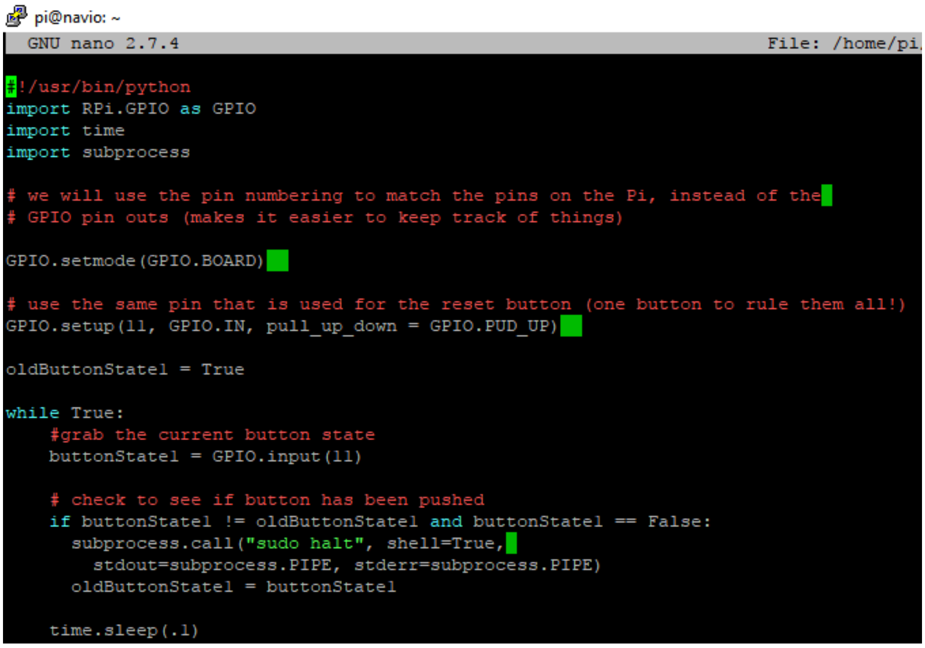
https://mega.nz/#!ppdynJjK!eclxScK_XLy0UmSQeyTK5WyzE5QfmQu0DKZp0WPX9Hc

The contents of the RetroPie text file can be opened with notepad or WordPad, and the commands in the text file are copied one after another, into the pi@navio command interface as shown below



Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Raspberry Pi3 pin connection diagram



```
#!/usr/bin/python
import RPi.GPIO as GPIO
import time
import subprocess

# we will use the pin numbering to match the pins on the Pi, instead of the
# GPIO pin outs (makes it easier to keep track of things)

GPIO.setmode(GPIO.BOARD)

# use the same pin that is used for the reset button (one button to rule them all!)
GPIO.setup(11, GPIO.IN, pull_up_down = GPIO.PUD_UP)

oldButtonState1 = True

while True:
    #grab the current button state
    buttonState1 = GPIO.input(11)

    # check to see if button has been pushed
    if buttonState1 != oldButtonState1 and buttonState1 == False:
      subprocess.call("sudo halt", shell=True,
        stdout=subprocess.PIPE, stderr=subprocess.PIPE)
      oldButtonState1 = buttonState1

    time.sleep(.1)
```

Coding for the addition of turn-off switch at free pin 11 or GPIO17 of RPi

Link to purchases of the micro-Header pins to connect to UART or other ports on Navio2 is included in the PDF file named “Drone BOM List”.

Running an autonomous mission

Please refer to manual flight first if this is the first time you arm your quadcopter.

To test out, first find a wide and empty open field with little obstructions or trees. Carry with you your laptop (Mission Planner), your drone, and all necessary equipment (such as transmitter, spare propellers, spanner for tightening the motor cap if needed, cable ties if needed, etc.) to the empty field. Connect your telemetry ground and air module to your laptop and to the flight controller respectively. Making sure the telemetry connection has been configured as in the section “Connecting to ground control station”, press ‘Connect’ on the top right corner after both the ground and air module has the same solid LED lights (not flashing).

On Mission Planner, go to the Flight Mode tab. Use a mouse to easier navigate through the map. The mouse scroll wheel can be used to zoom in and out of the map, and then use the left-click to drag the map to interactively navigate to any directions. A pop-out dialogue box will ask if you want to set the current coordinate of your drone as the Home location (the point for your drone to take-off). Press ‘Yes’ to do so. With this the quadcopter icon (your drone’s location) on the map will then merge with the ‘H’ icon (Home location) on the map.

With the GPS antenna properly connected to the MCX connector on the Flight Controller (see component setup), the Google map will load automatically the map where your drone is currently at. Note that the home button (‘H’ icon) is loaded as your current drone’s location (quadcopter icon), i.e. the ‘H’ icon must be coinciding with the quadcopter icon. Otherwise, press ‘Home location’ as shown below to do so.

GEO 5.1497814
100.4971898
SRTM 2.91m

GoogleSatelliteMa
Status: loaded tiles

Lat 5.1488738
Long 100.4990349
Alt (abs) 10.64000034

Next, on the map, perform path planning or mission planning by clicking on points (WP) where you want your drone to follow through as shown below, one WP after another.

During the path planning, a WayPoint table is automatically generated. The WP table contains the coordinates of location of each of the selected WP and the commands we want the drone to execute at specific WPs. Move your cursor to the line that separates between the map and your table and pull down to enlarge the map with respect to the table.

Distance: 0.3791 km
Prev: 183.46 m AZ: 112
Home: 188.21 m

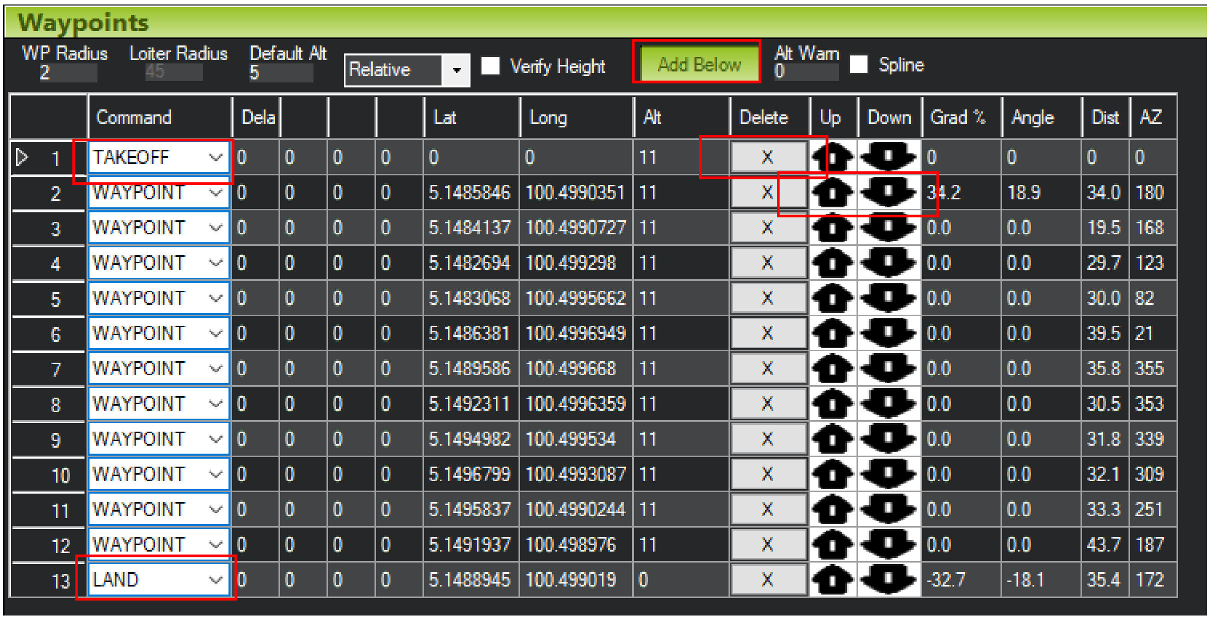
Waypoints

WP Radius 2 | Loiter Radius 45 | Default Alt 5 | Relative | ☐ Verify Height | Add Below | Alt Warn 0 | ☐ Spline

	Command	Dela				Lat	Long	Alt	Delete	Up	Down	Grad %	Angle	Dist	AZ
▷ 1	WAYPOINT	0	0	0	0	5.1491723	100.4990351	5	X	⬆	⬇	15.1	8.6	33.6	0
2	WAYPOINT	0	0	0	0	5.1494822	100.4991585	5	X	⬆	⬇	0.0	0.0	37.1	22
3	WAYPOINT	0	0	0	0	5.1495998	100.4993784	5	X	⬆	⬇	0.0	0.0	27.6	62
4	WAYPOINT	0	0	0	0	5.1495196	100.4996037	5	X	⬆	⬇	0.0	0.0	26.5	110
5	WAYPOINT	0	0	0	0	5.1491510	100.4996949	5	X	⬆	⬇	0.0	0.0	42.2	166
6	WAYPOINT	0	0	0	0	5.1488571	100.4996896	5	X	⬆	⬇	0.0	0.0	32.7	181
7	WAYPOINT	0	0	0	0	5.1486167	100.4997003	5	X	⬆	⬇	0.0	0.0	26.8	177
8	WAYPOINT	0	0	0	0	5.1484350	100.4996681	5	X	⬆	⬇	0.0	0.0	20.5	190
9	WAYPOINT	0	0	0	0	5.1482747	100.4994857	5	X	⬆	⬇	0.0	0.0	26.9	229
10	WAYPOINT	0	0	0	0	5.1482747	100.4992175	5	X	⬆	⬇	0.0	0.0	29.7	270
11	WAYPOINT	0	0	0	0	5.1484404	100.4990405	5	X	⬆	⬇	0.0	0.0	26.9	313
12	WAYPOINT	0	0	0	0	5.1486968	100.4989976	5	X	⬆	⬇	0.0	0.0	28.9	351

To enable the drone to carry out a complete mission, add a WP above the first WP and change the command for the new first WP to be ‘TakeOff’. To add a WP, click on the row below which you want to insert the WP in sequence, and then press ‘Add below’. WP will be added below the row, and press the arrow UP or DOWN to exchange the sequence of the WP. Move the ‘TakeOff’ WP to the top. Then, add below the last WP another new WP and set the new last WP command to be ‘Land’. An example of the WP table is as shown below.

To delete a WP, click ‘X’ and the entire row will be deleted.



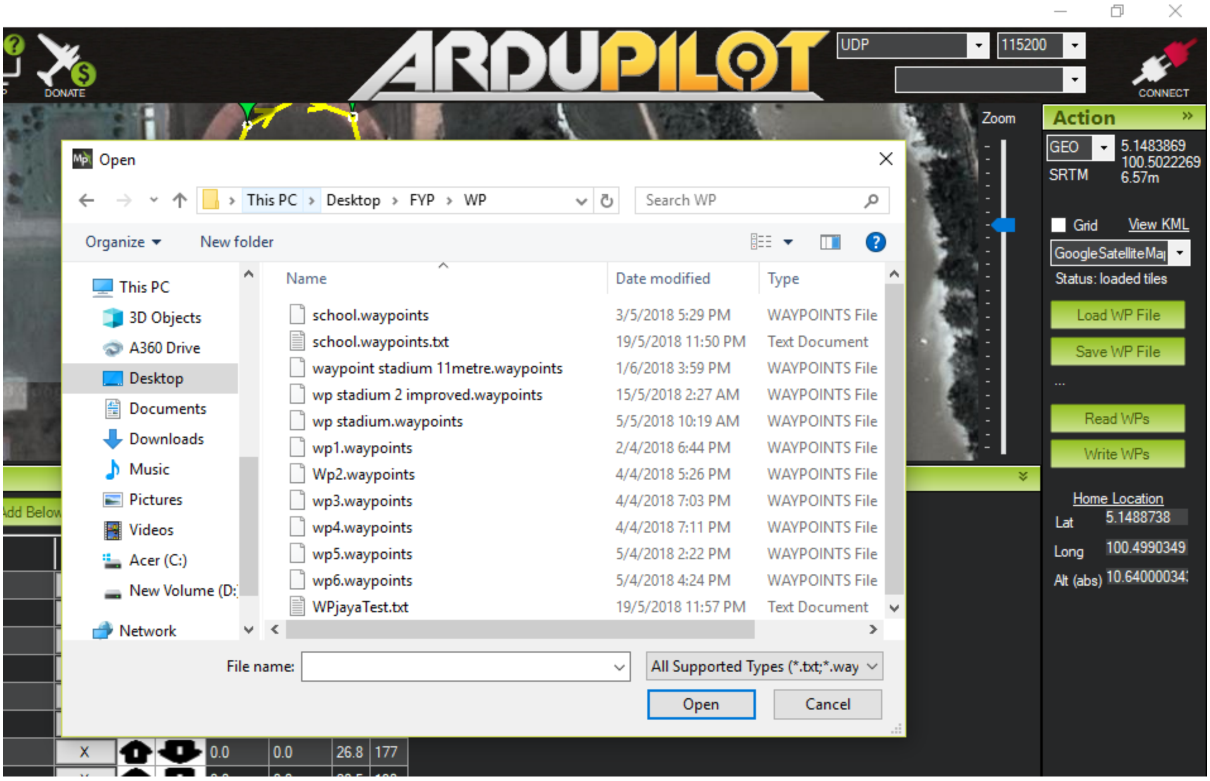
Waypoints

WP Radius 2 | Loiter Radius 45 | Default Alt 5 | Relative | ☐ Verify Height | Add Below | Alt Warn 0 | ☐ Spline

	Command	Dela				Lat	Long	Alt	Delete	Up	Down	Grad %	Angle	Dist	AZ
▷ 1	TAKEOFF	0	0	0	0	0	0	11	X	⬆	⬇	0	0	0	0
2	WAYPOINT	0	0	0	0	5.1485846	100.4990351	11	X	⬆	⬇	34.2	18.9	34.0	180
3	WAYPOINT	0	0	0	0	5.1484137	100.4990727	11	X	⬆	⬇	0.0	0.0	19.5	168
4	WAYPOINT	0	0	0	0	5.1482694	100.499298	11	X	⬆	⬇	0.0	0.0	29.7	123
5	WAYPOINT	0	0	0	0	5.1483068	100.4995662	11	X	⬆	⬇	0.0	0.0	30.0	82
6	WAYPOINT	0	0	0	0	5.1486381	100.4996949	11	X	⬆	⬇	0.0	0.0	39.5	21
7	WAYPOINT	0	0	0	0	5.1489586	100.499668	11	X	⬆	⬇	0.0	0.0	35.8	355
8	WAYPOINT	0	0	0	0	5.1492311	100.4996359	11	X	⬆	⬇	0.0	0.0	30.5	353
9	WAYPOINT	0	0	0	0	5.1494982	100.499534	11	X	⬆	⬇	0.0	0.0	31.8	339
10	WAYPOINT	0	0	0	0	5.1496799	100.4993087	11	X	⬆	⬇	0.0	0.0	32.1	309
11	WAYPOINT	0	0	0	0	5.1495837	100.4990244	11	X	⬆	⬇	0.0	0.0	33.3	251
12	WAYPOINT	0	0	0	0	5.1491937	100.498976	11	X	⬆	⬇	0.0	0.0	43.7	187
13	LAND	0	0	0	0	5.1488945	100.499019	0	X	⬆	⬇	-32.7	-18.1	35.4	172

Make sure that your drone will not hit any objects such as trees, poles and etc. that arise near the path planned by avoiding drawing any WP near to them altogether. Set the desired altitude to a suitable value. It is advisable that the autonomous flight has a ground clearance of at least 5 metres. Set it higher to overcome trees, lampposts and other objects. Make sure also that the flight path is complete and there is no WPs that is out of the zone. Be very careful in this because sometimes WPs that are not desired will appear in between and that is shown by yellow lines extending to other locations, Make sure to delete the wrong WPs before any actions are done.

On the action panel as shown below, there are 4 actions that are important, which are Save, Load, Read and Write. Any WP tables generated on Mission Planner can be saved to your PC as a text file (WP file) so that it can be loaded to the Mission Planner whenever you want to run the same mission. To load a saved WP file, press the Load button and direct to where you have saved your WP file and open it. The WP table will be loaded to your Mission Planner. Double check if the loaded mission is really the mission you intend to run. Then, press the Write button to save the WP table on Mission Planner. After that, press ‘Read’ to send the WP file to the Flight Controller so that the drone will follow through the path. This requires the drone to be connected with the GCS through a telemetry module. After the ‘Read’, Mission Planner will ask again if you want to set the Home location (Launch location) to be at the current loaded drone coordinate. Press ‘No’ to maintain the WP file. Perform the read for several times to make sure your drone is told what to do.



Name	Date modified	Type
school.waypoints	3/5/2018 5:29 PM	WAYPOINTS File
school.waypoints.txt	19/5/2018 11:50 PM	Text Document
waypoint stadium 11metre.waypoints	1/6/2018 3:59 PM	WAYPOINTS File
wp stadium 2 improved.waypoints	15/5/2018 2:27 AM	WAYPOINTS File
wp stadium.waypoints	5/5/2018 10:19 AM	WAYPOINTS File
wp1.waypoints	2/4/2018 6:44 PM	WAYPOINTS File
Wp2.waypoints	4/4/2018 5:26 PM	WAYPOINTS File
wp3.waypoints	4/4/2018 7:03 PM	WAYPOINTS File
wp4.waypoints	4/4/2018 7:11 PM	WAYPOINTS File
wp5.waypoints	5/4/2018 2:22 PM	WAYPOINTS File
wp6.waypoints	5/4/2018 4:24 PM	WAYPOINTS File
WPjayaTest.txt	19/5/2018 11:57 PM	Text Document

File name:
All Supported Types (*.txt;*.way
Open
Cancel

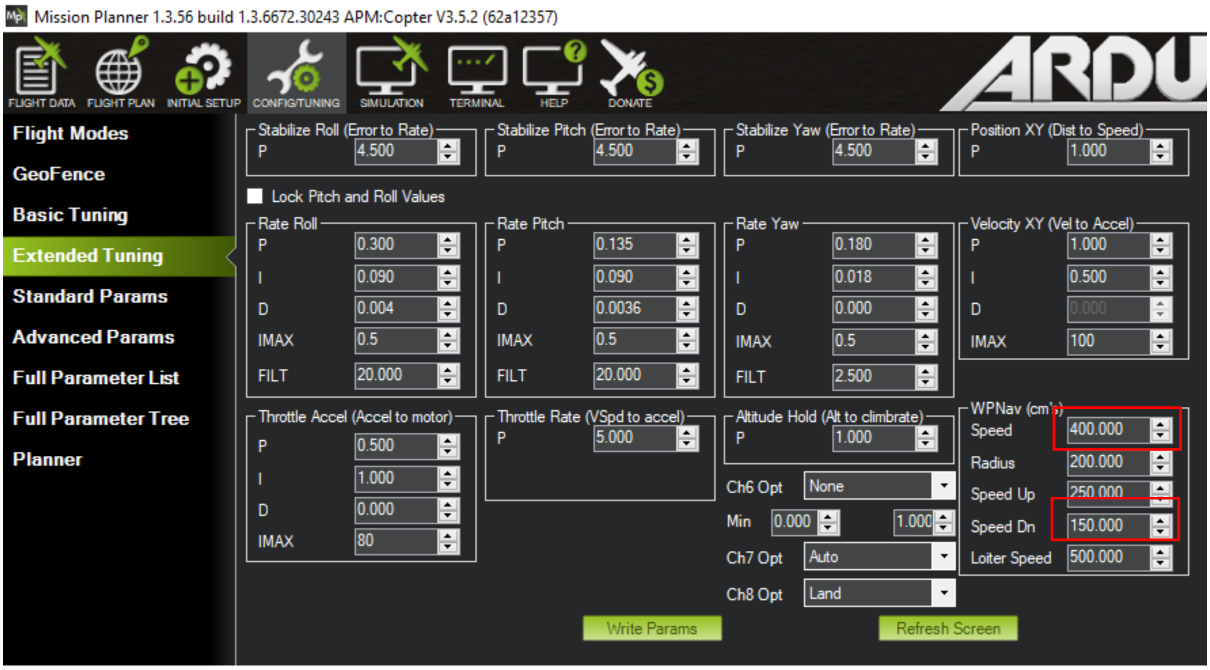
GEO 5.1483869 100.5022269
SRTM 6.57m
Status: loaded tiles

Lat 5.1488738
Long 100.4990349
Alt (abs) 10.64000034

When ‘Read’ is complete, arm your drone through your transmitter by pushing the left throttle stick (at zero throttle) fully to the right and hold. Your drone should now be placed as mentioned such that the loaded coordinates coincide with the Home Location (press the Home Location button to do so if they are at different locations on the map). Your drone will not arm in autonomous mode or land mode. Switch to stabilise mode first before you arm. For details on the switching between different flight modes, please refer to the section “change the flight mode on the Mission Planner”. Immediately after your drone is armed, engage the autonomous mode, and then give a gentle slight push to the throttle to initiate autonomous mode. The drone will take off automatically to the selected altitude and then begin moving to the first and subsequent WPs.

The speed across which your drone will fly across the WPs can be set from the Full Parameter list by searching for and editing the WPNAV_SPEED parameter. It is easier done on the Configuring tab > Extended Tuning as shown below. The Speed 400.000 means your drone is set to move horizontally at a speed of 4 metre per second. If you are not confident of your GPS accuracy, set the speed to be less than 3 metre per second.

The Landing speed is equally important. Land speed can be set in the full parameter list by searching and editing the LAND_SPEED parameter. The landing speed is the vertical final touch down speed when your drone is about to touch the ground. This value can be best set to be as low as 10 cm/s. The vertical descend speed (before final touch down) of your drone is set from WPNAV_SPEED_DN in the full parameter list. If you are not confident, or if you have barometer sensing errors, set the descend speed to be as low as 10 cm/s as well. Again, this can be easier set on the extended tuning tab, where 'Speed Dn' in the figure shown below means the quadcopter will descend at a speed of 1.5 metre/s before reducing to its final touch down speed at your set LAND_SPEED. Remember to hit the 'Write Params' button to save any changes.



Mission Planner 1.3.56 build 1.3.6672.30243 APM:Copter V3.5.2 (62a12357)

Stabilize Roll (Error to Rate)
P 4.500

Stabilize Pitch (Error to Rate)
P 4.500

Stabilize Yaw (Error to Rate)
P 4.500

Position XY (Dist to Speed)
P 1.000

☐ Lock Pitch and Roll Values

Rate Roll
P 0.300
I 0.090
D 0.004
IMAX 0.5
FILT 20.000

Rate Pitch
P 0.135
I 0.090
D 0.0036
IMAX 0.5
FILT 20.000

Rate Yaw
P 0.180
I 0.018
D 0.000
IMAX 0.5
FILT 2.500

Velocity XY (Vel to Accel)
P 1.000
I 0.500
D 0.000
IMAX 100

Throttle Accel (Accel to motor)
P 0.500
I 1.000
D 0.000
IMAX 80

Throttle Rate (VSpd to accel)
P 5.000

Altitude Hold (Alt to climbrate)
P 1.000

Ch6 Opt None
Min 0.000 1.000
Ch7 Opt Auto
Ch8 Opt Land

WPNav (cm's)
Speed 400.000
Radius 200.000
Speed Up 250.000
Speed Dn 150.000
Loiter Speed 500.000

To disengage the autonomous mode in the mid flight can be challenging. You would not want to do this if you are new and inexperienced. However, you will still need this when errors occur mid-flight such as the drone moves too slowly, or you realise you have selected a WP that may hit objects, or you did not set your drone to Land at the last WP or at a suitable spot or etc. Do not switch to Land Mode as the drone may fall from the sky! Push the throttle up above the mid throttle level before you switch to Stabilise mode. Prepare to take control over the throttle to prevent it plummeting from the sky. Fly it to safety before you land your drone. If you have pre-set Loiter mode on your transmitter, engage it to take back control from auto-mode, it is easier to control in Loiter Mode since it includes Altitude Hold.

After the path planning is complete, your drone will land at the last WP and the throttle will automatically turn off as soon as it touches the ground. If it doesn't, pull your throttle fully down and engage stabilise mode, then disarm your drone manually. Disarm your drone by pushing the throttle stick fully to the left at zero throttle position.


Mission Planner 1.3.56 build 1.3.6672.30243

Distance: 0.0000 km
Prev: 152.37 m AZ: 9
Home: 152.37 m

UDP
115200

GEO 5.1502249 100.4992551
SRTM 3.82m

☐ Grid
GoogleSatelliteMap
Status: loaded tiles

Home Location
Lat 5.1488738
Long 100.4990349
Alt (abs) 10.64000034

WP Radius 2
Loiter Radius 45
Default Alt 5
Relative
☐ Verify Height
Alt Warn 0
☐ Spline

Manual Flight control

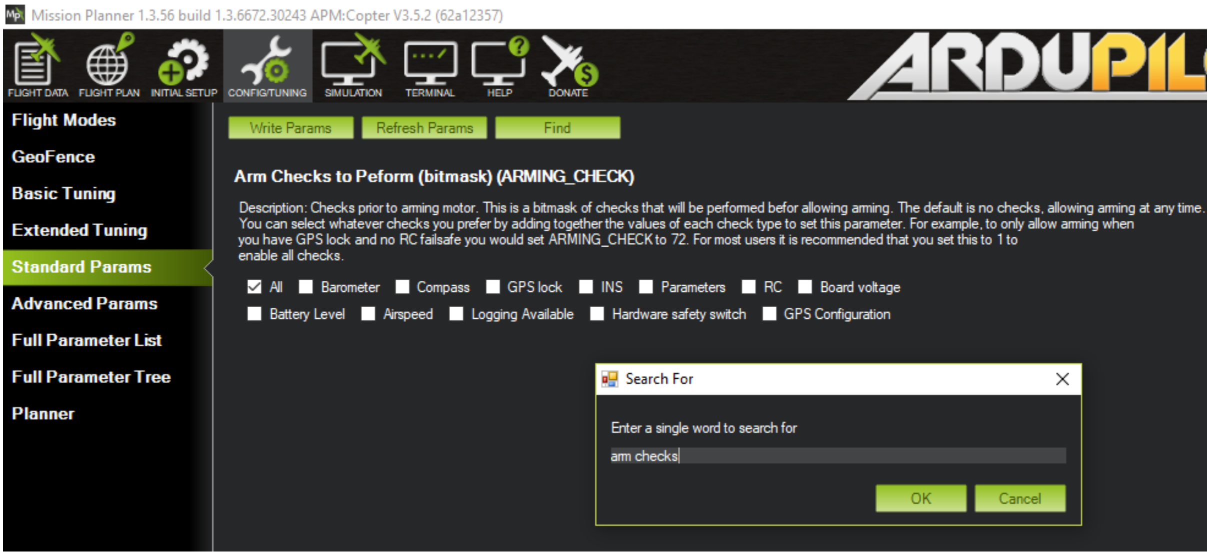
Make sure that you have done with all the calibration and setup in the previous steps. Make sure also you have set all basic required modes ready on the transmitter and on Mission Planner such as Stablise, Altitude Hold, and Land. Refer to "Change the flight mode on mission planner with respect to radio transmitter channel" for more details.

Place your quadcopter on a flat ground in a wide open field and then plug in the battery. Wait until the long beeping sound stops. Then arm your quadcopter by pushing the throttle stick at zero throttle position fully to the right. If this is the first time you arm your quadcopter, you are going to make sure that the directions of your motor rotation are correct. Arm it with your propellers off, and then feel with your finger if the direction of rotation of each motor is correct. Otherwise, interchange any 2 connections (bullet connector) between the specific ESC and motor. Please refer to motor setup in the section "Drone Hardware Setup".

Arming the quadcopter

If your quadcopter cannot arm, connect your drone to the Mission Planner to see what is causing the pre-arm check to trigger. For example, if the compass offset is too high, you need to re-calibrate your magnetometer. Google search the pre-arm check error message followed by the keyword "Ardupilot" (eg: Compass offset too high Ardupilot) to look for methods to resolve the error.

The pre-arm check's components can be selectively disabled by going to the Configuring tab > Standard Params > click the 'Find' button and search for arm checks. Uncheck 'All' and then individually reselect all but the faulty components and then press 'Write Params' to save. Make sure you do this after making sure the faulty components will not result in crashes.



Mission Planner 1.3.56 build 1.3.6672.30243 APM:Copter V3.5.2 (62a12357)

Arm Checks to Peform (bitmask) (ARMING_CHECK)

Description: Checks prior to arming motor. This is a bitmask of checks that will be performed befor allowing arming. The default is no checks, allowing arming at any time. You can select whatever checks you prefer by adding together the values of each check type to set this parameter. For example, to only allow arming when you have GPS lock and no RC failsafe you would set ARMING_CHECK to 72. For most users it is recommended that you set this to 1 to enable all checks.

☑ All ☐ Barometer ☐ Compass ☐ GPS lock ☐ INS ☐ Parameters ☐ RC ☐ Board voltage
☐ Battery Level ☐ Airspeed ☐ Logging Available ☐ Hardware safety switch ☐ GPS Configuration

Search For

Enter a single word to search for

arm checks

After your quadcopter is armed, i.e. the motors are spinning at minimum speed, push the throttle gently and slowly up until you see your drone begins to lift, give more thrust to help it lift and rise into the air. At your desired altitude, switch to Altitude Hold to remain the altitude. Switch to Loiter mode if you want the drone to stay at the current GPS location while maintaining at the same altitude. Your drone should stay at the last altitude when you switch to these modes. If it falls or descends, you can still push the throttle up to control the flight.

If you do not wish to use the Altitude Hold mode, make sure you are confident at controlling the altitude by constantly pushing the throttle stick up when it descends, and push it back down when it rises, while at the same time controlling the pitch, roll and yaw of your quadcopter. You are advised to test your pilot skill on a cheaper Chinese drone (without Altitude Hold) first before you pilot your self-built drone. Some drones has built-in Altitude Hold ability, and learning over those drones will not help you master the skill of controlling the drone's altitude manually.

You can begin moving your drone horizontally after your drone is in the air with suitable amount of ground clearance. Your drone moves by rotating around three-axes:

1. Rolling to the left and rolling to the right (Please refer to the transmitter control diagram)
2. Pitching nose down and pitching nose up
3. Yawing to the left (anti-clockwise) and yawing to the right (clockwise)

Make sure you always monitor the battery voltage using a BB alarm battery voltage monitor. Land your quadcopter as soon as the alarm triggers off. The vertical landing speed can be adjusted together with the descend speed as shown in the section "Running an autonomous mission". If you do not want to use the Land Mode, slowly descend your quadcopter to the ground by controlling the throttle stick up and down. That again, requires a lot of practices to master a perfect landing. Make sure to try this out on cheaper drones first.

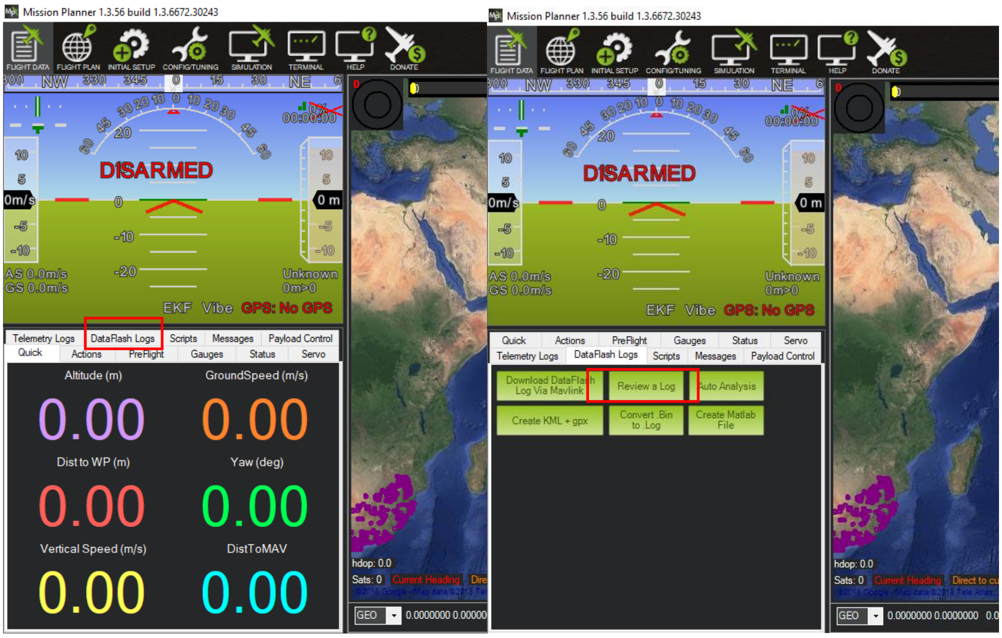
A soon as your quadcopter touches the ground, the throttle will turn off automatically. If it doesn't, pull your throttle fully down and engage the stabilise mode, then disarm your drone manually. Disarm your drone by pushing the throttle stick fully to the left at zero throttle position.

Data Flash Log Analysis

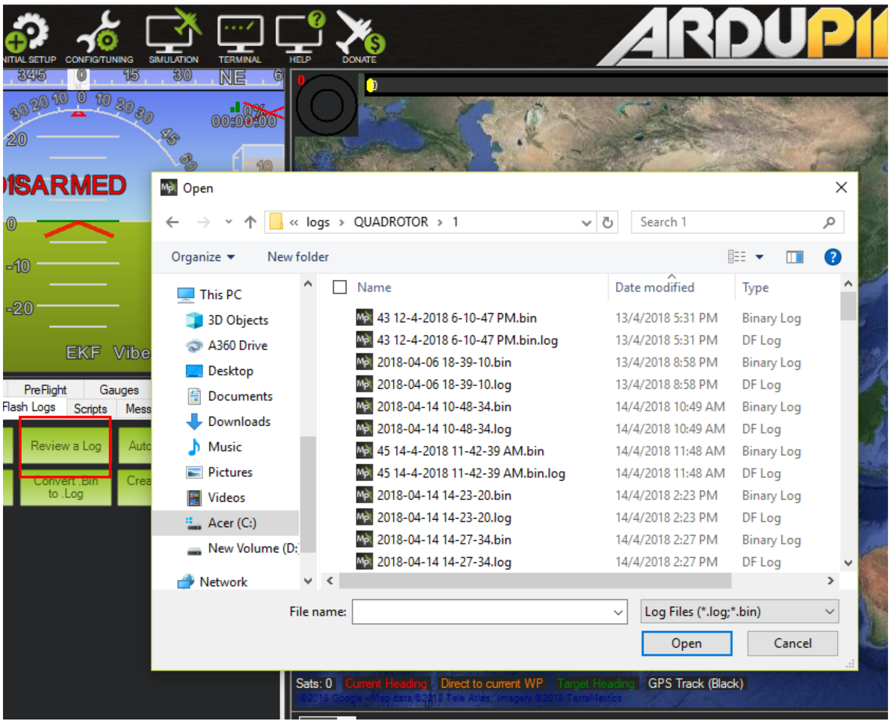
Post-flight analysis tells you everything about your flight ranging from barometer altitude, GPS locations, satellite reception, battery voltage, attitude responses, desired attitude, radio communication, speed, motor current, to 3D images of the flight path and modes.

To load the data flash log from Mission Planner, you should not connect the Mission Planner to your drone through a radio telemetry. Instead, you will need to establish udp connection by connecting both (RPi3 and your laptop) to the same LAN and perform necessary telemetry network configuration as shown in the section "Connecting to the Ground Control Station".

Next, press the data flash log tab as shown below > Download DataFlash Log via Mavlink > and then select the log files of flights (based on accurate time and date) that you want analyse > Download selected files. The downloaded log files will be stored in your PC.



The downloaded binary log files can be loaded to the Mission Planner for analyses without having to connect to the drone. Simply press 'Review a Log' and then open to load the desired flight data based on recorded time and date.



Name	Date modified	Type
43 12-4-2018 6-10-47 PM.bin	13/4/2018 5:31 PM	Binary Log
43 12-4-2018 6-10-47 PM.bin.log	13/4/2018 5:31 PM	DF Log
2018-04-06 18-39-10.bin	13/4/2018 8:58 PM	Binary Log
2018-04-06 18-39-10.log	13/4/2018 8:58 PM	DF Log
2018-04-14 10-48-34.bin	14/4/2018 10:49 AM	Binary Log
2018-04-14 10-48-34.log	14/4/2018 10:49 AM	DF Log
45 14-4-2018 11-42-39 AM.bin	14/4/2018 11:48 AM	Binary Log
45 14-4-2018 11-42-39 AM.bin.log	14/4/2018 11:48 AM	DF Log
2018-04-14 14-23-20.bin	14/4/2018 2:23 PM	Binary Log
2018-04-14 14-23-20.log	14/4/2018 2:23 PM	DF Log
2018-04-14 14-27-34.bin	14/4/2018 2:27 PM	Binary Log
2018-04-14 14-27-34.log	14/4/2018 2:27 PM	DF Log

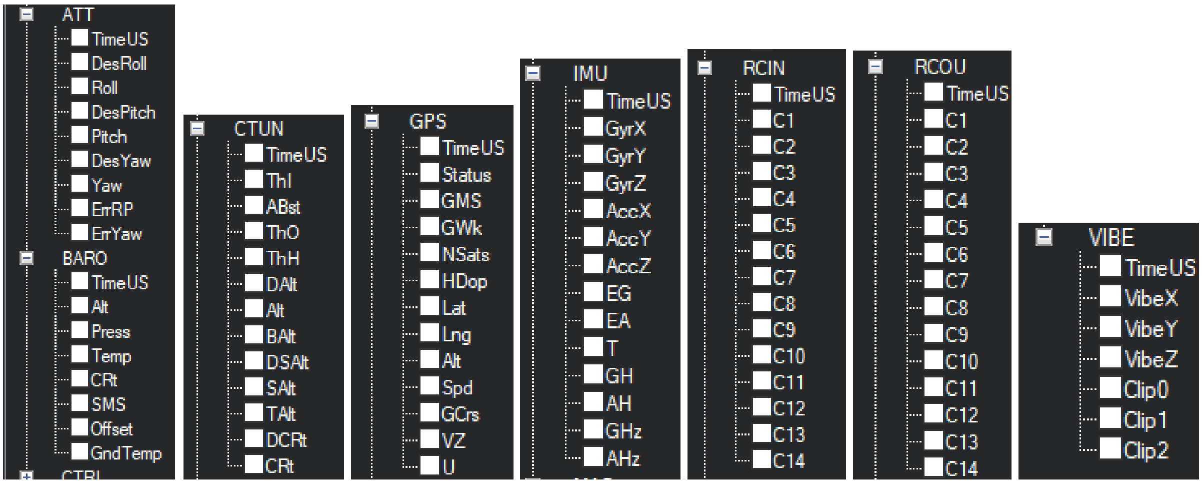
File name:

Log Files (*.log;*.bin)

You will be automatically directed to where the flash logs are stored when you want to review them. Otherwise, they can be found in

C:\Users\Username\ Mission Planner\logs\QUADROTOR\1

Double-click on the selected binary log file to open the flash log. From the flash log, some of the popular and useful parameters in the data flash logs include:



- ATT
  - ☐ TimeUS
  - ☐ DesRoll
  - ☐ Roll
  - ☐ DesPitch
  - ☐ Pitch
  - ☐ DesYaw
  - ☐ Yaw
  - ☐ ErrRP
  - ☐ ErrYaw
- BARO
  - ☐ TimeUS
  - ☐ Alt
  - ☐ Press
  - ☐ Temp
  - ☐ CRt
  - ☐ SMS
  - ☐ Offset
  - ☐ GndTemp
- CTUN
  - ☐ TimeUS
  - ☐ ThI
  - ☐ ABst
  - ☐ ThO
  - ☐ ThH
  - ☐ DAlt
  - ☐ Alt
  - ☐ BAlt
  - ☐ DSAlt
  - ☐ SAlt
  - ☐ TAlt
  - ☐ DCRt
  - ☐ CRt
- GPS
  - ☐ TimeUS
  - ☐ Status
  - ☐ GMS
  - ☐ GWk
  - ☐ NSats
  - ☐ HDop
  - ☐ Lat
  - ☐ Lng
  - ☐ Alt
  - ☐ Spd
  - ☐ GCrs
  - ☐ VZ
  - ☐ U
- IMU
  - ☐ TimeUS
  - ☐ GyrX
  - ☐ GyrY
  - ☐ GyrZ
  - ☐ AccX
  - ☐ AccY
  - ☐ AccZ
  - ☐ EG
  - ☐ EA
  - ☐ T
  - ☐ GH
  - ☐ AH
  - ☐ GHz
  - ☐ AHz
- RCIN
  - ☐ TimeUS
  - ☐ C1
  - ☐ C2
  - ☐ C3
  - ☐ C4
  - ☐ C5
  - ☐ C6
  - ☐ C7
  - ☐ C8
  - ☐ C9
  - ☐ C10
  - ☐ C11
  - ☐ C12
  - ☐ C13
  - ☐ C14
- RCOU
  - ☐ TimeUS
  - ☐ C1
  - ☐ C2
  - ☐ C3
  - ☐ C4
  - ☐ C5
  - ☐ C6
  - ☐ C7
  - ☐ C8
  - ☐ C9
  - ☐ C10
  - ☐ C11
  - ☐ C12
  - ☐ C13
  - ☐ C14
- VIBE
  - ☐ TimeUS
  - ☐ VibeX
  - ☐ VibeY
  - ☐ VibeZ
  - ☐ Clip0
  - ☐ Clip1
  - ☐ Clip2

1. Attitude
    a. Desired roll/pitch/yaw: Your intended attitude or yaw value from radio
    b. Roll/pitch/yaw: Actual attitude and yaw response measured by on-board sensors
2. BARO
    a. Alt: Barometer altitude sensing
3. ThO: Output Throttle value
4. GPS
    a. NSats: Number of satellite reception
    b. HDOP: Horizontal Dilution of Precision to tell GPS accuracy
    c. Lat: Latitudinal coordinates throughout the flight
    d. Lng: Longitudinal coordinates throughout the flight
    e. Alt: GPS measured altitude
    f. Spd: Speed
    g. GCrs: Ground Course (Heading of quadcopter)
5. IMU
    a. Gyr X/Y/Z: Gyroscope IMU data
    b. Acc X/Y/Z: Accelerometer IMU data
6. RCIN: Input radio channel value
7. RCOUT
    a. C1/C2/C3/C4: Motor current for Motor 1 to 4
8. VIBE
    a. Vibe X/Y/Z: Vibration in different axes

Simply check the box of the parameter you want to analsye on the bottom right panel and the respective graph of that parameter will be generated. It is advisable to uncheck unrelated parameters before you analyse a new parameter so that the graph's scale can be reset suitably. To increase the size of the graph, move your cursor to the double dotted lines and then drag it down to expand the graph. This takes up in turn the space below it.

If you want to save the graph, right-click and 'save image as'. Even easier, use a Snipping tool to crop wherever you want to save or copy. The link to download Snipping Tool is as follow:

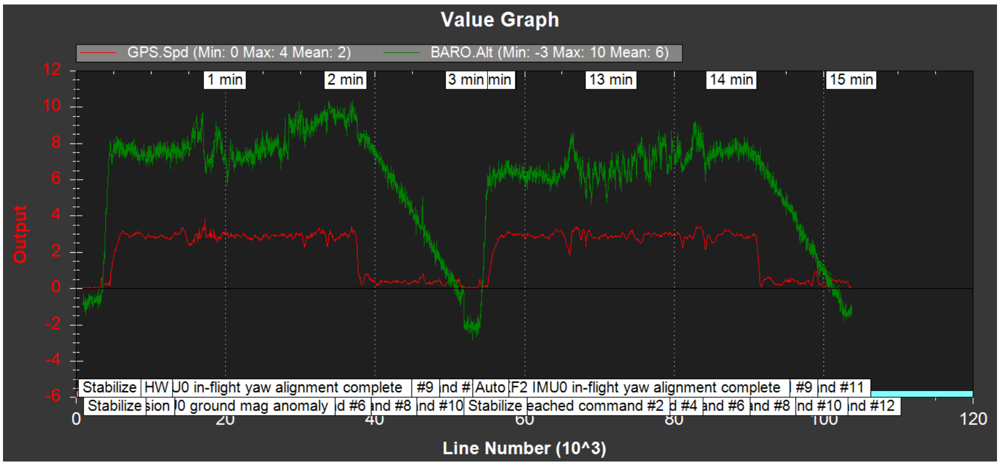
https://snipping-tool-plus-plus.en.softonic.com/download

It requires Java tools to work: The link to download Java is as follows https://www.java.com/en/

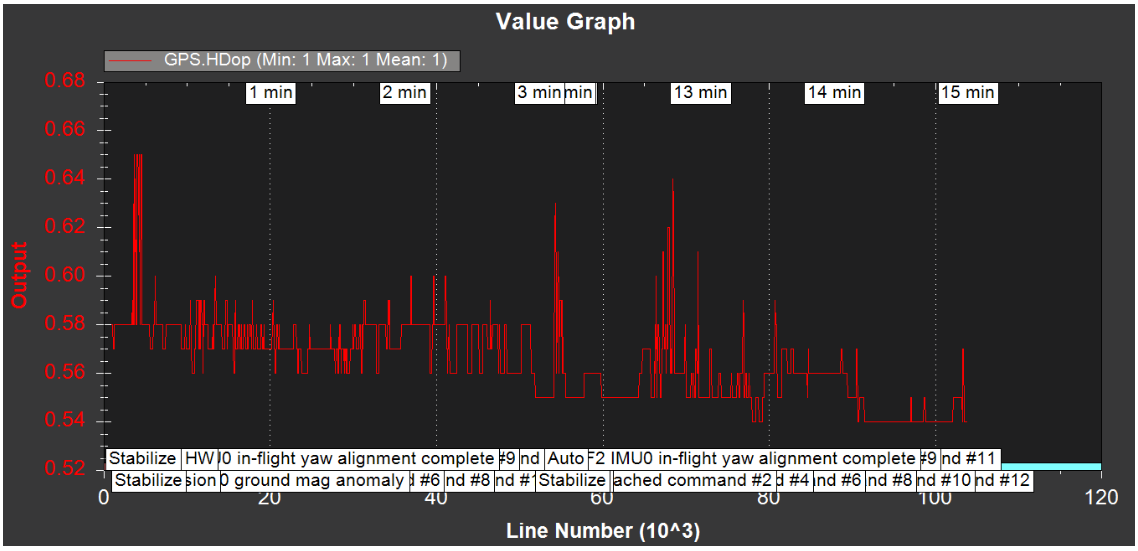
The flight data is useful to find out reasons behind crashes, malfunctionalities, or inability to carry out specific functions. For example, if the quadcopter keeps falling down in the Altitude Hold mode, look at the barometer data to see if the barometer sensing of altitude is working properly.

-2	-1	0	Type	Length	Name	Format	Columns				
10	2018-04-14 14:2...	FMT	135	39	RCIN	QHHHHHHHHH...	TimeUS	C1	C2	C3	C4
11	2018-04-14 14:2...	FMT	136	39	RCOU	QHHHHHHHHH...	TimeUS	C1	C2	C3	C4
12	2018-04-14 14:2...	FMT	137	15	RSSI	Qf	TimeUS	RXRSSI			
13	2018-04-14 14:2...	FMT	139	37	BARO	Qffcflff	TimeUS	Alt	Press	Temp	CRt
14	2018-04-14 14:2...	FMT	140	21	POWR	QffH	TimeUS	Vcc	VServo	Flags	
15	2018-04-14 14:2...	FMT	143	45	CMD	QHHHffffff	TimeUS	CTot	CNum	CId	Prm1
16	2018-04-14 14:2...	FMT	144	20	RAD	QBBBBBHH	TimeUS	RSSI	RemRSSI	TxBuf	Noise
17	2018-04-14 14:2...	FMT	146	43	CAM	QIHLLeeeccC	TimeUS	GPSTime	GPSWeek	Lat	Lng
18	2018-04-14 14:2...	FMT	232	43	TRIG	QIHLLeeeccC	TimeUS	GPSTime	GPSWeek	Lat	Lng
19	2018-04-14 14:2...	FMT	162	30	ARSP	QffcffB	TimeUS	Airspeed	DiffPress	Temp	RawPress
20	2018-04-14 14:2...	FMT	164	45	CURR	QfffcHHHHHHH...	TimeUS	Volt	Curr	CurrTot	Temp
21	2018-04-14 14:2...	FMT	165	45	CUR2	QfffcHHHHHHH...	TimeUS	Volt	Curr	CurrTot	Temp
22	2018-04-14 14:2...	FMT	163	27	ATT	QccccCCCC	TimeUS	DesRoll	Roll	DesPitch	Pitch

Other examples of data analysis

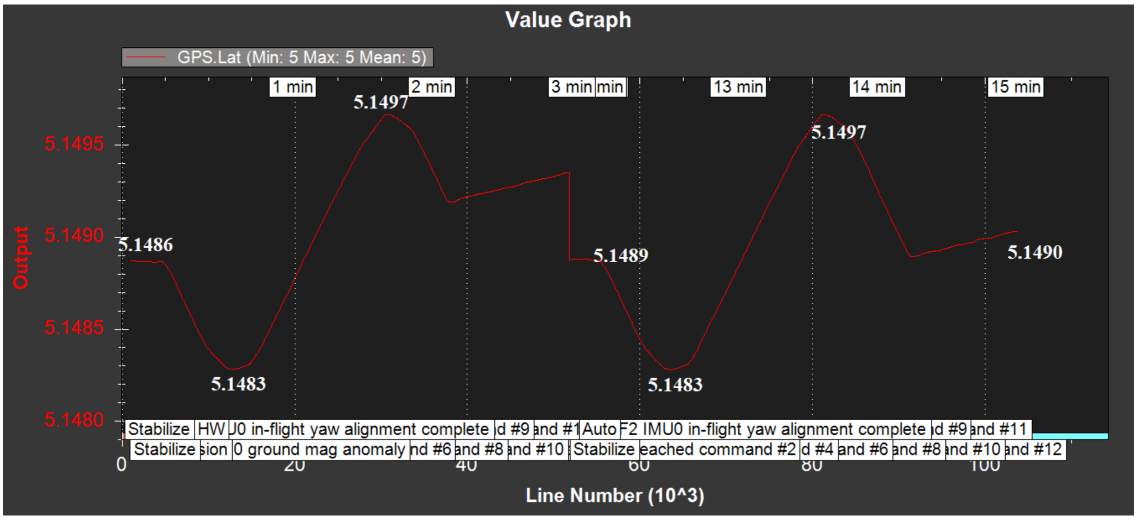


Barometer altitude and GPS speed for stable speed averaged at 3 metre/s

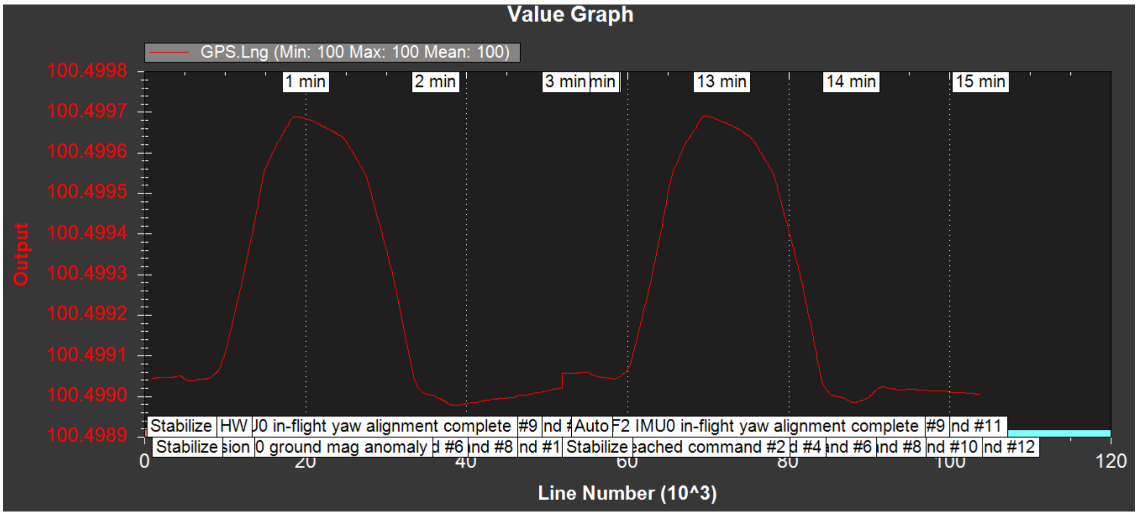


HDOP for second mission

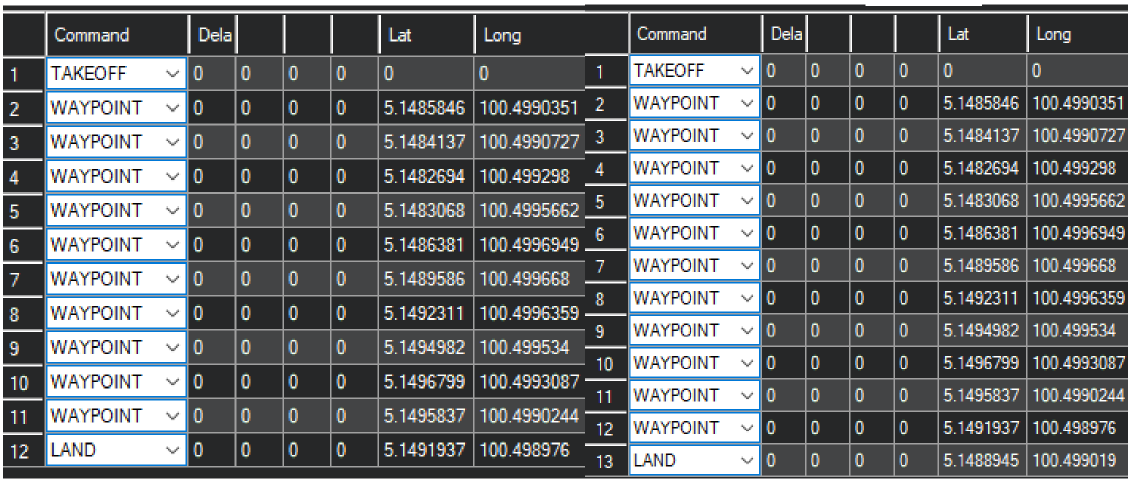
DOP Value	Rating	Description for GPS accuracy
< 1	Ideal	For applications demanding the highest possible precision at all times.
1-2	Excellent	At this confidence level, positional measurements are considered accurate enough to meet all but the most sensitive applications.
2-5	Good	Represents a level that marks the minimum appropriate for making business decisions. Positional measurements could be used to make reliable in-route navigation suggestions to the user.
5-10	Moderate	Positional measurements could be used for calculations, but the fix quality could still be improved. A more open view of the sky is recommended.
10-20	Fair	Represents a low confidence level. Positional measurements should be discarded or used only to indicate a very rough estimate of the current location.
>20	Poor	At this level, measurements are inaccurate by as much as 300 meters with a 6-meter accurate device (50 DOP × 6 meters) and should be discarded.



Plot of latitudinal coordinates throughout the flight of an autonomous mission



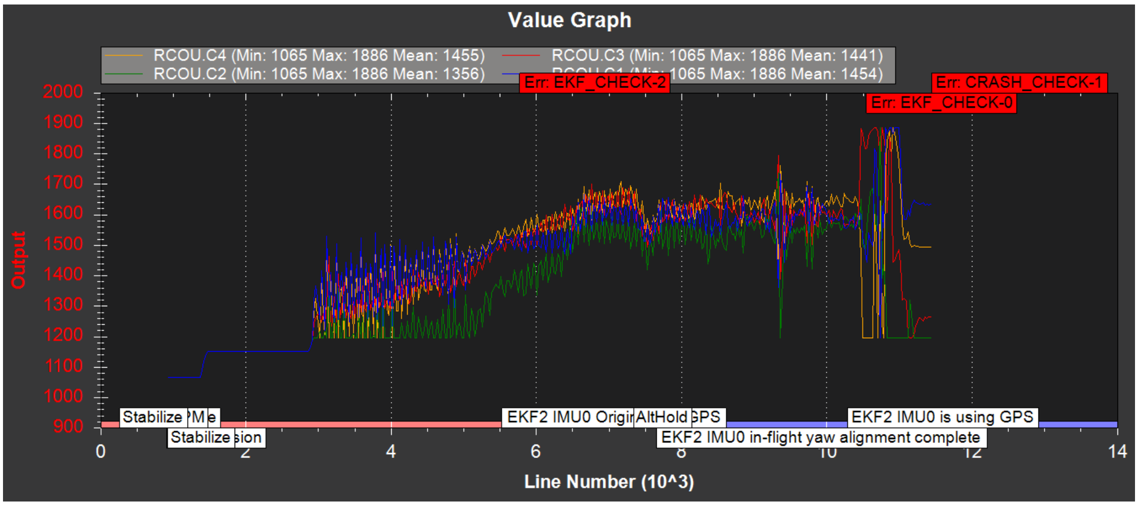
Plot of longitudinal coordinates throughout the flight of the second mission



	Command	Dela				Lat	Long
1	TAKEOFF	0	0	0	0	0	0
2	WAYPOINT	0	0	0	0	5.1485846	100.4990351
3	WAYPOINT	0	0	0	0	5.1484137	100.4990727
4	WAYPOINT	0	0	0	0	5.1482694	100.499298
5	WAYPOINT	0	0	0	0	5.1483068	100.4995662
6	WAYPOINT	0	0	0	0	5.1486381	100.4996949
7	WAYPOINT	0	0	0	0	5.1489586	100.499668
8	WAYPOINT	0	0	0	0	5.1492311	100.4996359
9	WAYPOINT	0	0	0	0	5.1494982	100.499534
10	WAYPOINT	0	0	0	0	5.1496799	100.4993087
11	WAYPOINT	0	0	0	0	5.1495837	100.4990244
12	LAND	0	0	0	0	5.1491937	100.498976

	Command	Dela				Lat	Long
1	TAKEOFF	0	0	0	0	0	0
2	WAYPOINT	0	0	0	0	5.1485846	100.4990351
3	WAYPOINT	0	0	0	0	5.1484137	100.4990727
4	WAYPOINT	0	0	0	0	5.1482694	100.499298
5	WAYPOINT	0	0	0	0	5.1483068	100.4995662
6	WAYPOINT	0	0	0	0	5.1486381	100.4996949
7	WAYPOINT	0	0	0	0	5.1489586	100.499668
8	WAYPOINT	0	0	0	0	5.1492311	100.4996359
9	WAYPOINT	0	0	0	0	5.1494982	100.499534
10	WAYPOINT	0	0	0	0	5.1496799	100.4993087
11	WAYPOINT	0	0	0	0	5.1495837	100.4990244
12	WAYPOINT	0	0	0	0	5.1491937	100.498976
13	LAND	0	0	0	0	5.1488945	100.499019

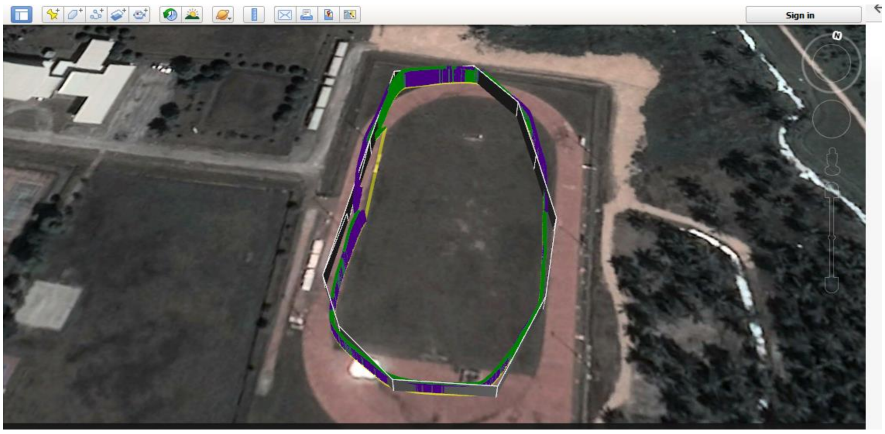
Comparing to the coordinates of autonomous mission from WP table



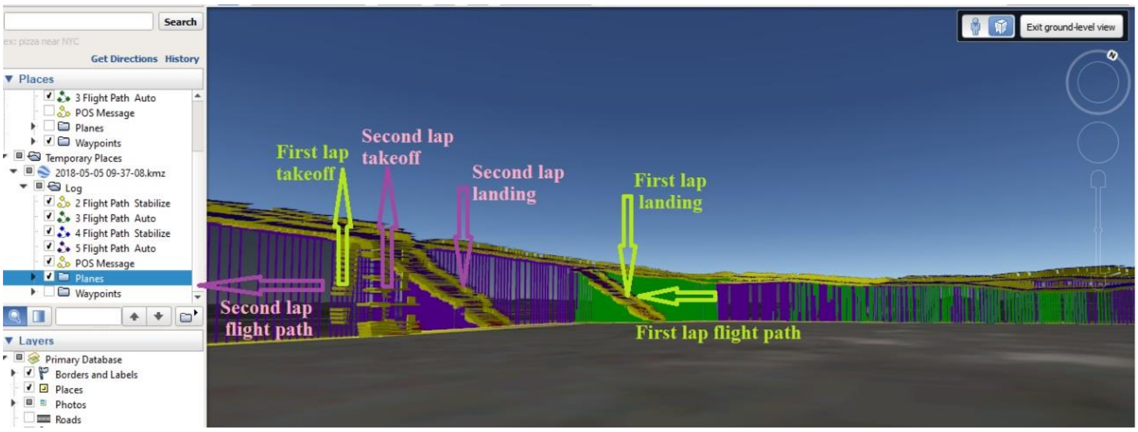
Motor current draw graph

Together with the downloaded binary log files is the KMZ file of the flight. KMZ files can be opened using the Goggle Earth pro software. Navigate in your PC to C:\Users\Username\ Mission Planner\logs\QUADROTOR\1 and then **open** the KMZ file **with** Google Earth Pro.

From Google Earth Pro, you will be able to analyse your flight in 3D street view,



The actual flight path that the quadcopter followed through from KMZ file of a mission



Places
- ☑ 3 Flight Path Auto
- ☐ POS Message
- ☐ Planes
- ☑ Waypoints
- Temporary Places
  - 2018-05-05 09-37-08.kmz
    - Log
      - ☑ 2 Flight Path Stabilize
      - ☑ 3 Flight Path Auto
      - ☑ 4 Flight Path Stabilize
      - ☑ 5 Flight Path Auto
      - ☑ POS Message
      - ☑ Planes
      - ☐ Waypoints

Layers
- Primary Database
  - ☑ Borders and Labels
  - ☑ Places
  - Photos
  - ☐ Roads

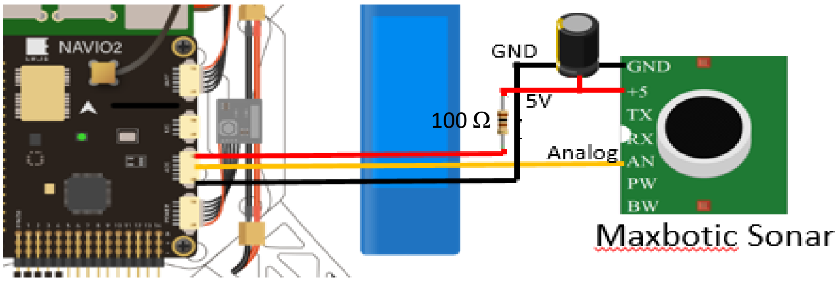
3D Street view image of the flight paths of mission two

3D street view image can be observed using Google earth pro. Different flight paths are differentiated based on the switch of flight modes as shown on the left panel above. The flight paths can be deselected from the list as shown in the left panel. Unchecked flight modes will disappear from the 3D map leaving only the checked flight path. From the diagram above, green path is the first lap of auto mode flight path the quadcopter has followed through while the overlapping purple path is the second lap of auto mode.

Horizontal distance measurement of the flight path can be done using the ruler tool in the software.

Object Avoidance Implementation

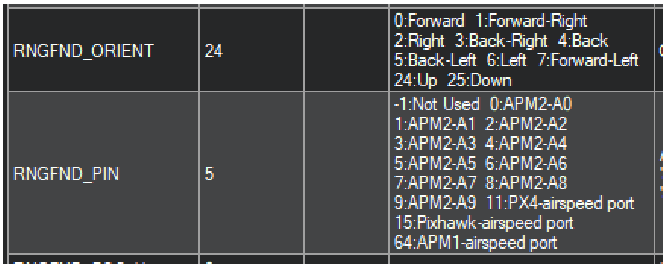
APM supports object avoidance in a maximum of 2 directions at most. Through the LV-MaxSonar-EZ0 high performance sonar by Maxbotix, the quadcopter is able to avoid obstacles up to a range of 6.45 m. This expensive EZ0 sensor is used only for object detection above the quadcopter, to avoid hitting objects when the quadcopter takes off or rises. The connection of the MaxSonar to Navio2 FC is through the analog to digital converter (ADC) pins, connecting as shown in below, the voltage, ground and analog pin from the FC to the sonar. To establish physical connection, the micro-header pin for the 6-pin ADC adopted by Navio2 had to be purchased online. The header pins will also fit the other ports so long the number of pins matches.



Maxbotix ultrasonic sensor connection diagram

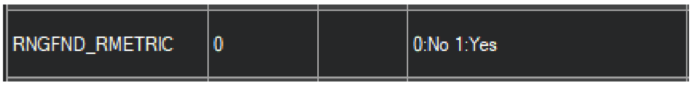
As studied, a 100uF capacitor is added in parallel to the supply to stabilise the voltage supply during times of high current consumption and a 100 Ω resistor connected in series to adjust the high logical state during idle. Together this configuration brings about a consistent values of readings from the MaxSonar sensor.

Settings have been made in the Mission Planner software’s “Full Parameter List” tab to enable the rangefinder and to specify the direction it will act against. Rangefinder orientation is set as 24 for UP orientation and the rangefinder pin is set as 5 with connection to the pin ADC 3 on Navio2 as shown below. Other required settings in the parameter list required are shown. Search for them using the “find” function.



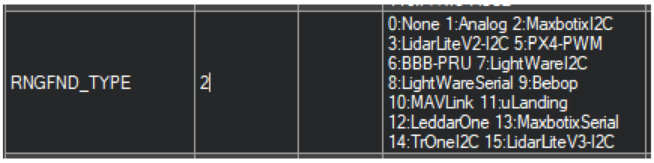
RNGFND_ORIENT	24		0:Forward 1:Forward-Right 2:Right 3:Back-Right 4:Back 5:Back-Left 6:Left 7:Forward-Left 24:Up 25:Down
RNGFND_PIN	5		-1:Not Used 0:APM2-A0 1:APM2-A1 2:APM2-A2 3:APM2-A3 4:APM2-A4 5:APM2-A5 6:APM2-A6 7:APM2-A7 8:APM2-A8 9:APM2-A9 11:PX4-airspeed port 15:Pixhawk-airspeed port 64:APM1-airspeed port

Orientation and pin setting of rangefinder



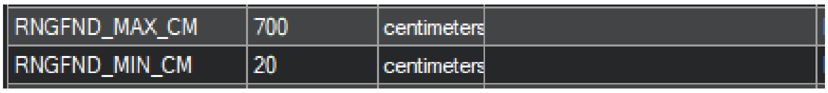
RNGFND_RMETRIC	0		0:No 1:Yes

The MaxSonar is not a ratiometric sensor



RNGFND_TYPE	2		0:None 1:Analog 2:MaxbotixI2C 3:LidarLiteV2-I2C 5:PX4-PWM 6:BBB-PRU 7:LightWareI2C 8:LightWareSerial 9:Bebop 10:MAVLink 11:uLanding 12:LeddarOne 13:MaxbotixSerial 14:TrOneI2C 15:LidarLiteV3-I2C

The type of rangefinder is selected



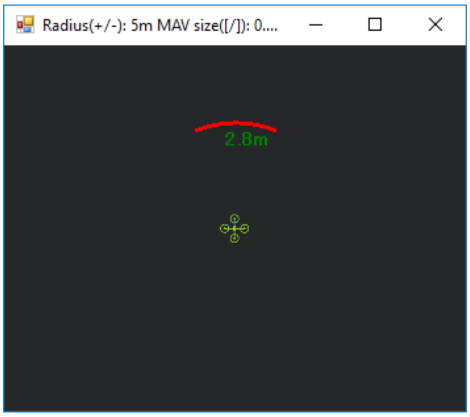
RNGFND_MAX_CM	700	centimeters	
RNGFND_MIN_CM	20	centimeters	

The maximum and minimum distance for sonar detection is specified

PRX_TYPE	4		0:None 1:LightWareSF40C 2:MAVLink 3:TeraRangerTower 4:RangeFinder

The type of proximity sensor used is selected

The real-time sensor data was not shown for all orientations other than down or 25 in the rangefinder tab in initial setup, but was shown in the radius radar initiated from the proximity button in the temps menu initiated by Ctrl-F shortcut key. Note that the 2-Dimensional radar will not show distance when the orientation is set to up, or down.



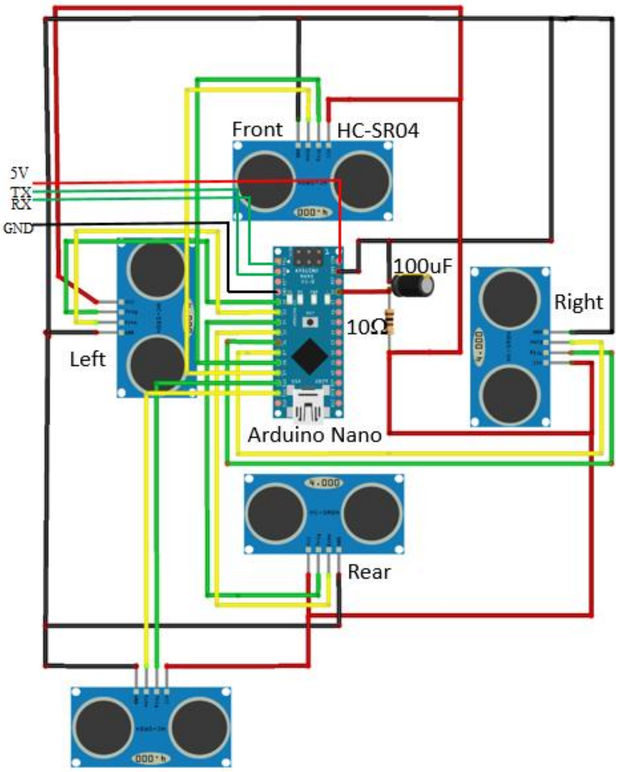
Radar window for object detection

After making sure that the sonar was working through monitoring from the radar window and testing out the data with different distances, the quadcopter was then safe for flight with obstacle avoidance. The sensor is mounted properly according to the orientation being set, making sure that no objects from the quadcopter, such as wiring or GPS stand, comes in the pings' travel path. The quadcopter is armed and then switched to only either guided or altitude hold mode for object avoidance.

Several limitations are underlined in the existing object avoidance system which is currently still under development by Emlid developers. The limitations include expensive ultrasonic sensors, limitations of the modes in which object avoidance can work, and limitation of the directions against which the object avoidance system can work.

Shao Fu from https://shaofuhw.github.io/Portfolio/ have come up with a solution of using HC-SR04 ultrasonic sensor for the implementation of object avoidance in his YouTube channel https://www.youtube.com/watch?v=kmpxkKq3zNA

As shown in the diagram below, five inexpensive ultrasonic sensors HC-SR04 were used for object detection in the front, right, rear, left and down direction. Arduino microcontroller acts as the processor for the receiving sensor data and generate responses accordingly. Arduino Nano was used because of its small size and relatively lighter weight. The Arduino connections with all sensors were sketched and the complete circuit diagram was drawn.

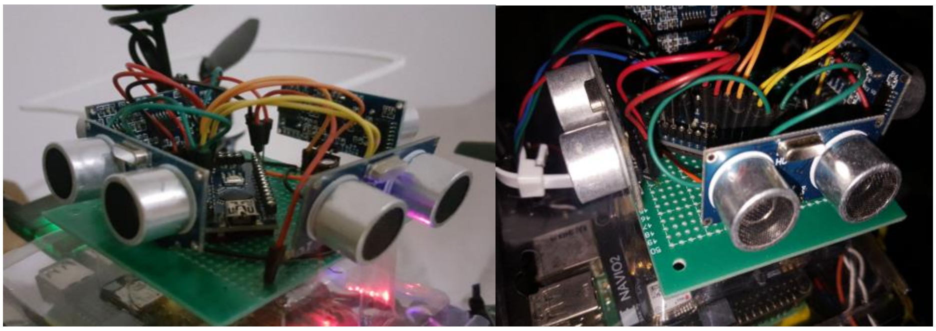


Object avoidance using HC-SR04 ultrasonic sensor

This time, the system was connected to the FC through the UART serial port. 5V and ground completes the supply circuit to power the Arduino Nano, while Transmitter (TX) pin of the FC was connected to Receiver (RX) pin of Arduino Nano. Similarly, RX pin of the FC was connected to TX pin at the Arduino. The UART port was initially used for the air module of telemetry to enable communication with the ground module in the GCS. To make way for the object avoidance system a simple modification and an extra component was required so the UART port can be spared.

The trick is to exchange the ground and air module where now the ground module was connected through USB port at the FC, and the air module was connected to the GCS. Details of this is explained in the section “Drone Hardware Setup” under the telemetry component.

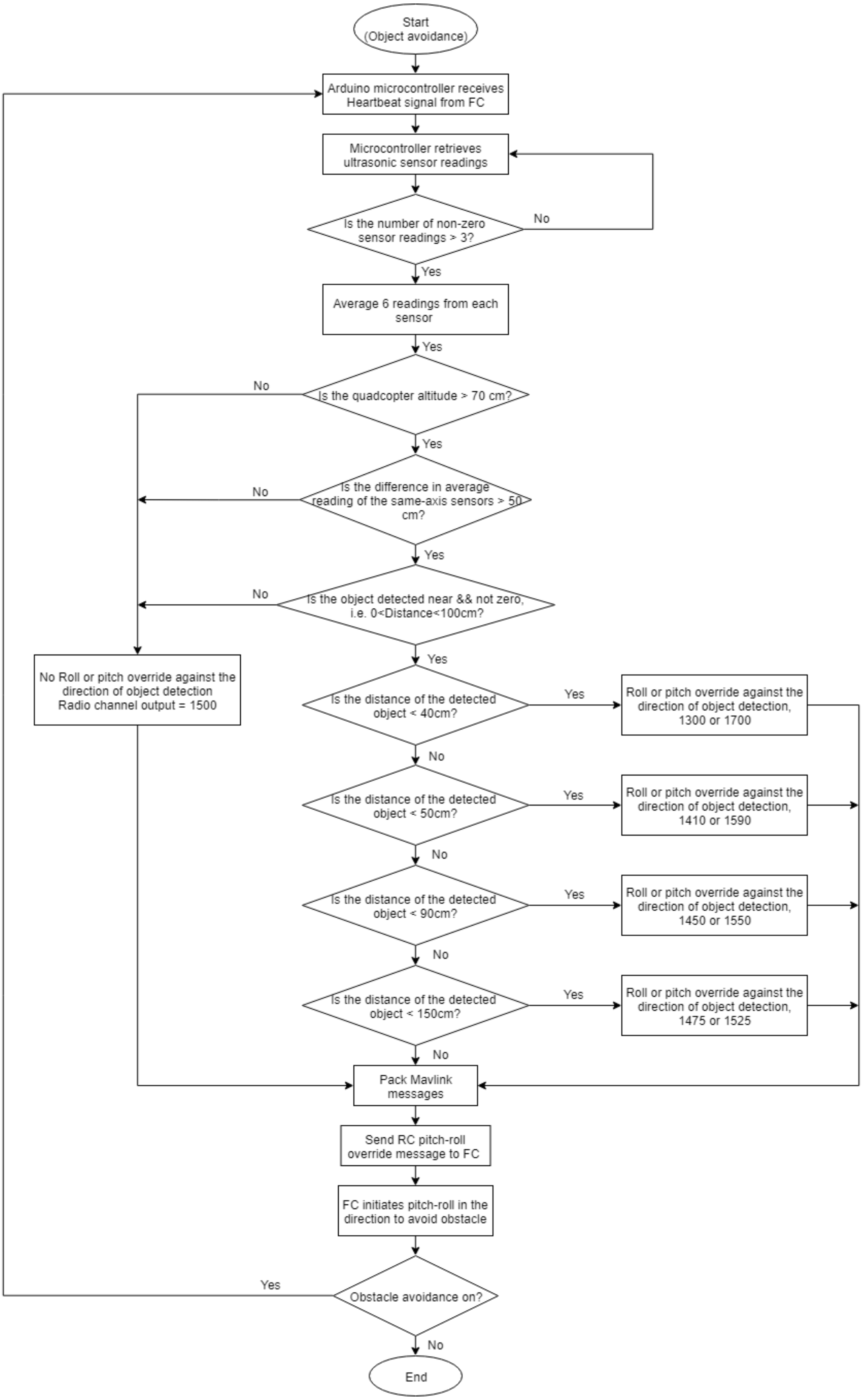
Connection points required were being precisely and tightly soldered on a copper board. Jumpers were used to ease out connections and female header pins were soldered on place for the installation of ultrasonic sensors in each direction, as shown below. Again, capacitors and resistors were added to improve consistency of each sensor.



Ultrasonic sensor model

The software of the object avoidance system was built with the Arduino Software (IDE). For distance measurement, NewPing library was used to retrieve raw sensor data directly in centimetre. To improve overall sensor data consistency, averaged value of 5 sensor readings were taken. It was learnt that sonar outputting zero distance is not uncommon. It happens when the distance of the object is out of the detection range, or when there is a directional uncertainty. To avoid the average values getting affected by such errors, only average values with at least 4 non-zero readings are taken, otherwise discarded.

Pitch and roll responses can be realised by sending mavlink pakages containing signals to override the existing roll and pitch value. From the data flash logs, the input RC channel shows a value nearing 1500 for both pitch and roll values when the quadcopter is stable. When the quadcopter pitches nose down, the value drops below 1500, whereas when pitched up, the value rises above 1500 in the pitch channel. Likewise, when the quadcopter tilts to the right, it rises above 1500, in the roll channel, and vice versa.



Object detection and avoidance flowchart

This piece of information is vital because the Mavlink messages that were sent from the Arduino microcontroller to the FC will be the messages to override those values in the respective radio channels. The Arduino software was coded in a way that the change in roll-pitch values increase when the nearest distance detected reduces.

Roll and pitch override with respect to distance of object detected

Distance/cm	Roll/Pitch override
<40	1700/1300
<50	1590/1410
<90	1550/1450
<150	1525/1475

Responses will not be initiated just as soon as the criteria stated in table above is met. Several other conditions have to be fulfilled at the same time to make sure that responses only take place when necessary and when the environment allows so.

Firstly, the quadcopter will only begin its object avoidance, in any flight modes, when the ‘down’ ultrasonic sensor shows altitude over a certain heights. This stabilises the flight during takeoffs, where unnecessary tilting can lead to parts of the quadcopter hitting the ground. Only after making sure that the quadcopter has sufficient clearance from the ground to perform as defined a pitch or roll, object avoidance will take control of the flight.

Secondly, the object detection distance of a particular sensor has to be ‘near’ as in lesser in distance than a specified value, i.e. 100 cm. ‘Near’ in its context has been defined as a non-zero value – in case obstacles are too far away – below 100 cm.

Thirdly, distance sensor along the same axis has to agree upon a response in a way that the direction for response where the quadcopter will act toward must have no obstacles. In other words, if the difference in the distance of object at the front sensor and that of the rear sensor does not exceed a safe specified value, the quadcopter will not be made to tilt to the back even if there is obstacle that appears to fulfil the first 2 conditions stated.

During the upload of sketch on Arduino Nano, TX and RX pins have to be disconnected from Arduino to the FC. Blinking LEDs on the Arduino signifies certain exchange of information between the 2 components. For instance, Arduino is coded to receive at its RX pins heartbeat signal from the FC for every passing second. In the other way round, the TX pin of Arduino will be sending overriding signal to the FC whenever the conditions stated are fulfilled for a specified set of response to be initiated.

When all conditions fulfil, RC overriding mavlink message containing new RollOut and PitchOut values are sent in a standard UART send function. Because the program runs in loop, as soon as the obstacle is distanced from the particular sensor, the response should stop and return to initialised value of zero pitch or roll, i.e. at 1500.

Tests for object avoidance were carried out by constraining 4 sides of the quadcopter to the PVC rectangular structure built as shown below.