

# MHV Quadcopter Workshop 2.0

---

*July/August 2013*

## Table of Contents

Objective .....	2
What to bring to the Workshop.....	3
The hardware .....	3
The software .....	3
Ground Station.....	3
Autopilot .....	3
ESC's .....	4
Transmitter .....	4
Radios.....	4
Build Instructions .....	4
Timing.....	4
Evening 1.....	5
Introduction to Quadcopters .....	5
Components.....	5
Generating Thrust .....	6
Basic Stats .....	6
Autopilot Modes .....	6
Initial charging of the battery .....	7
Frame assembly .....	7
Evening 2.....	9
Transmitter Firmware Upgrade .....	9
Transmitter setup .....	11
ESC Firmware Flashing .....	13
Simulation .....	15
Wiring up the system –Power train .....	16
Propeller Balancing .....	17
Radio binding (and servo test) .....	18
Evening 3.....	19
Safety .....	19

Current Australian Law and Regulations.....	19
LIPO Batteries.....	20
Propellers .....	20
Radio Interference .....	21
Soldering .....	21
Wiring up the system – Flight Electronics.....	21
Testing the ground station.....	22
PX4 Calibration.....	26
Motor and ESC tests and calibration .....	28
Build Checklist.....	29
Evening 4.....	30
First steps .....	30
Trims .....	30
Longer flights.....	31
Morning 1 and 2.....	31
LOITER .....	31
RTL.....	31
ALT_HOLD .....	31
AUTO .....	31
References .....	33

## Objective

To design and build a quadcopter for a workshop to be held at the Make Hack Void Hackerspace (Canberra). It is designed to be:

- (Relatively) Cheap
- Commonly available spare parts
- Open source or hackable hardware/software where available
- Easy for beginners to build
- Strong/tough enough to withstand beginner pilots
- Able to carry a 200g payload (such as a small camera)

## What to bring to the Workshop

- Laptop (and charger!). Confirm that the APM Mission Planner software (see the Ground station software section) runs on your laptop beforehand.
- MHV will provide everything else.

Note the workshop will require basic soldering. If you haven't soldered before, we can teach you during the workshop

## The hardware

The following list includes all parts (and some tools) required to build the quadcopter. Note you may need a soldering iron for electrical work. Other hardware tools should be in your shed (or local hackerspace!)

The list can be found at <https://canberraauav.readthedocs.org/en/latest/quadcopter-workshops/quadcopter-workshop-2-0/parts-list.html> for the list of tools.

Total cost of all parts is \$730.

## The software

### Ground Station

The standard GCS is the APM Mission Planner

<http://ardupilot.com/downloads/>

This will run on Windows XP/Vista/7 or Ubuntu (or similar). Linux users need to install the mono framework.

The program does require a moderately powerful laptop/PC to run on. It will barely run on most netbooks. An active internet connection will be required to use some features.

The APM Mission Planner is frequently updated. Fortunately, it includes auto-update software.

More advanced users can use MAVProxy (a command line GCS) instead:

[http://qgroundcontrol.org/mavlink/mavproxy\\_startpage](http://qgroundcontrol.org/mavlink/mavproxy_startpage)

### Autopilot

We use the ArduPilotMega (APM):

<https://github.com/diydrones/ardupilot>

The APM mission planner includes the APM software in it. This software is being frequently updated with bug fixes, better navigation code and more features. It is recommended to update when a new version is released.

Note that the PX4 board is used for this workshop. When uploading firmware ensure the PX4 version of the APM software is selected.

**A note about terminology:** PX4 refers to the *physical board based* on the ARM chip. APM refers to the *software* running on the board. To make things more confusing, APM also refers to the physical boards based on the ATMEGA2560 chip.

Due to the 2 hardware boards, there are two builds of the APM software:

- ATMEGA board
- PX4 board

It should be emphasised that the ATMEGA and PX4 builds run the same APM software, minus the different hardware drivers. The high level navigation, positioning and other features are exactly the same.

For the purposes of this workshop, I will refer to the PX4 when referring to the board and APM when referring to the software.

## ESC's

The SimonK firmware is compatible with the ESC's used in this project.

<https://github.com/sim-/tgy>

This firmware increases the update rate of the ESC's, making the ESC (and hence motors) more responsive to commanded throttle changes.

## Transmitter

The er9x firmware is built for the Turnigy 9X transmitter.

<http://code.google.com/p/er9x/>

The er9x firmware makes the GUI far easier to navigate.

## Radios

Like the APM, the radios use open-source firmware.

<http://code.google.com/p/ardupilot-mega/wiki/3DRadio>

The firmware is not frequently updated, so we probably won't need to update this.

## Build Instructions

### Timing

The build will take several evenings. Flight testing and lessons will take an afternoon.

The timing is:

- **Evening 1** –Introduction. Flashing ESC's, Transmitters and frame assembly. Soldering of power connectors. Simulator setup
- **Evening 2** – Flight electronics installation.
- **Evening 3** – Calibration and testing. Safety Briefing
- **Evening 4** – Indoor flight at Dickson College. Simple flights
- **Afternoon 1/2** – Outdoors flight at a local oval. Training of advanced flight modes

## Evening 1

### Introduction to Quadcopters

A quadcopter is a flying vehicle possessing 4 identical rotors, evenly spaced around the central fuselage (hub).

*First, quadrotors do not require mechanical linkages to vary the rotor blade pitch angle as they spin. This simplifies the design and maintenance of the vehicle. Second, the use of four rotors allows each individual rotor to have a smaller diameter than the equivalent helicopter rotor, allowing them to possess less kinetic energy during flight. This reduces the damage caused should the rotors hit anything. For small-scale UAVs, this makes the vehicles safer for close interaction. Some small-scale quadrotors have frames that enclose the rotors, permitting flights through more challenging environments, with lower risk of damaging the vehicle or its surroundings. (Wikipedia)*



### Components

**Electronic Speed Controller (ESC):** Regulates power to the motor according to the input throttle level. It also provides +5V power for the flight electronics

**RC Receiver:** A (usually 2.4 GHz) RC radio receiver on the quadcopter that receives commands from the RC transmitter on the ground. One way link

**Telemetry Link:** A (usually 915 MHz or 433 MHz) bidirectional link between the flight controller and ground station. Provides current status to the ground station and accepts flight commands to the quadcopter.

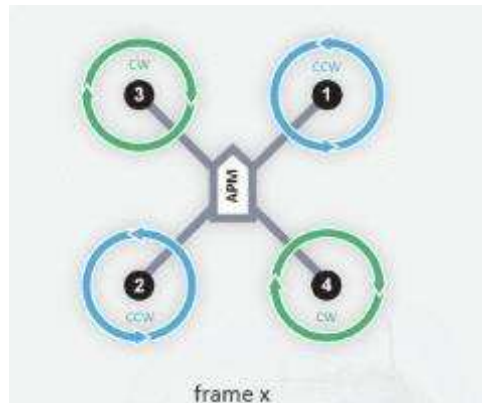
**RC Transmitter:** A (usually 2.4 GHz) RC radio transmitter used by the pilot to direct the quadcopter's direction and position.

**Hub:** The central fuselage of the quadcopter. Contains the flight electronics and battery

**Arm:** The beam that each of the motors sit on.

## Generating Thrust

The motors and propellers alternate in direction and pitch, in order to cancel out and horizontal torque. All propeller/motor combinations still produce downwards thrust:



By altering the thrust to each of the motors, the quadcopter can move and turn:

**Pitch (forward/back):** Decrease thrust to motors 1 and 3. Increase thrust to motors 2 and 4 to maintain altitude. This will pitch the quadcopter forwards, giving it a forward velocity.

**Yaw (horizontal rotation):** Decrease thrust to motors 3 and 4. Increase thrust to motors 2 and 1 to maintain altitude. This will rotate the quadcopter counter-clockwise.

**Roll (left/right):** Decrease thrust to motors 2 and 3. Increase thrust to motors 1 and 4 to maintain altitude. This will roll the quadcopter left, giving it a sideways velocity.

## Basic Stats

The basic stats are:

- Top speed: 2.5 m/s vertical, 6m/s horizontal (no payload)
- Max Payload: 400g
- Flight time: 12 min (no payload)

Note quadcopters (particular those < 1kg) are very sensitive to payloads. The payload should be as close as practical to the quadcopter's centre of mass.

## Autopilot Modes

The ArdupilotMega (2.9.1 at time of writing) has the following flight modes:

- **STABILISE** – The primary mode. Use the RC sticks to navigate the quadcopter. Otherwise, it will stay level

- **ALT\_HOLD** – It will maintain the current altitude. Otherwise, the RC sticks can be used to navigate the quadcopter
- **LOITER** – It will maintain position, heading and altitude at the current point. Altitude can be changed by changing the throttle value
- **RTL** – It will return to its takeoff (where the quadcopter was armed) point. Depending on the mode settings, it will either hover at a 5-10m altitude and wait for the pilot to land it or gradually descend until it lands.
- **AUTO** – It will run through the waypoints currently loaded. The waypoints can be a simple “go to point”, “loiter for X min”, “change velocity”, “land”. Note there is not automated take off.
- **ACRO** – Advanced Mode. The RC sticks act as rate controllers rather than position controllers. There is no automatic stabilisation
- **LAND** – lands the quadcopter on the ground at its current point

More modes may become available as new versions of the APM software are released.

### Initial charging of the battery

The 2 batteries come shipped with a nominal 60% charge. Use the battery charger to charge both the flight battery and RC transmitter battery. The settings on the battery charger should be: 3S LiPo battery type, charge rate 5A for the flight battery and 2.5A for the RC Transmitter battery.

Both the power cable and balance connector should be hooked up to the charger.

The batteries will be fully charged around the 12.6 V level. The charger will emit a series of loud beeps when it is done.

### Frame assembly

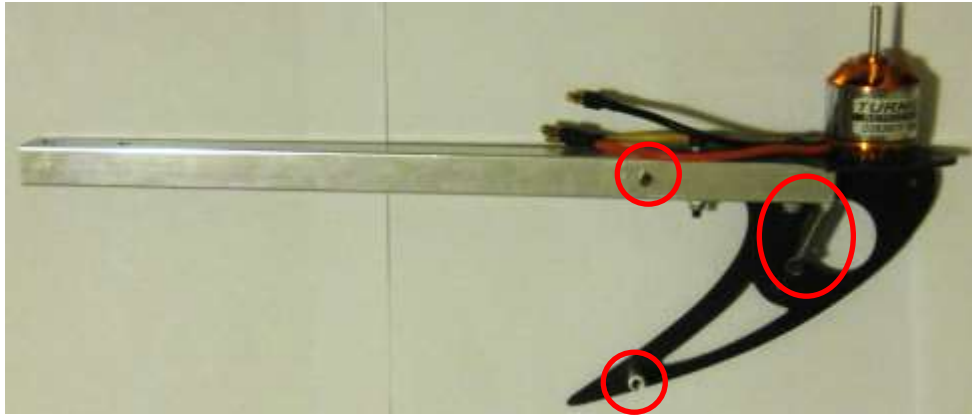
1. Attach the motor mounts to the end of each arm:



2. Use the M3 screws to attach the motors:



3. Attach 1 of the leg struts, along with the spring and screws:



4. Attach the other half of the strut:

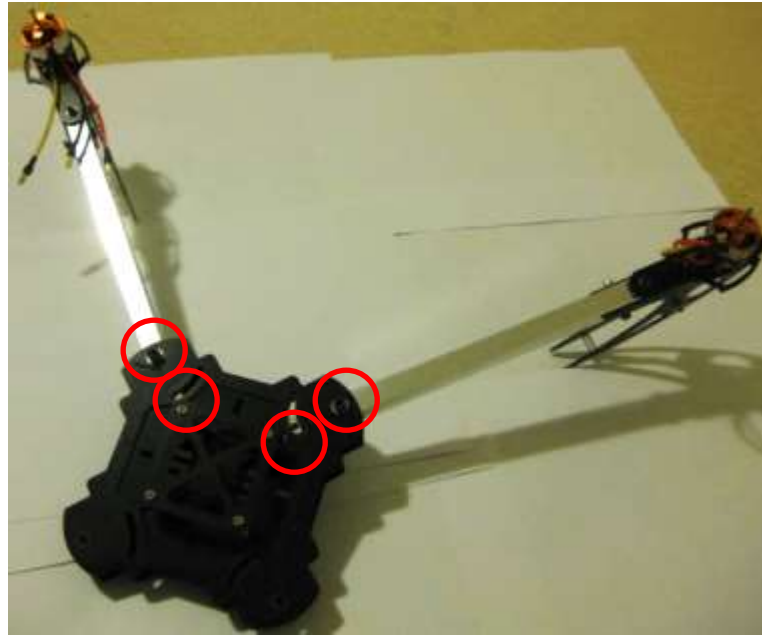


5. Assemble the top of the hub plate. Note there are 3 plates here:

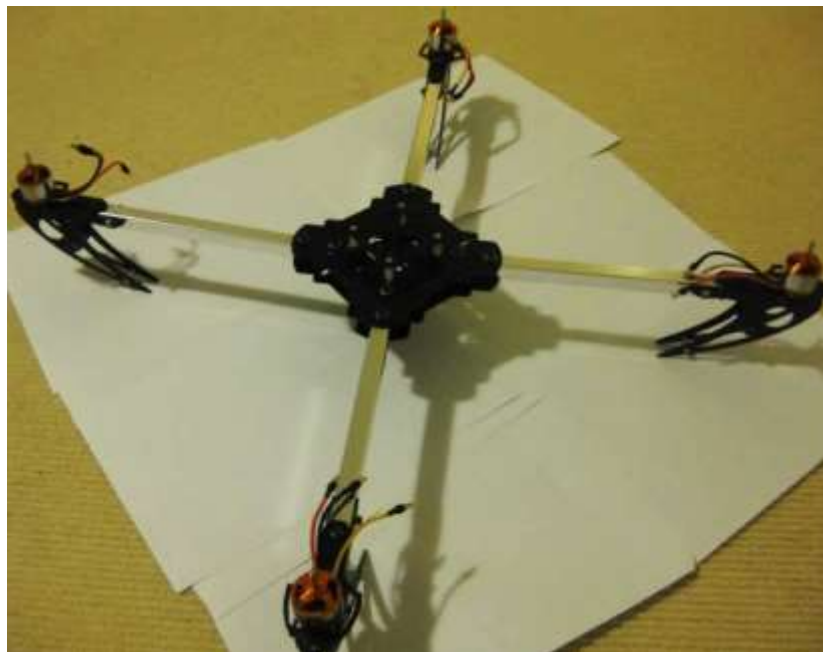


6. Along with the bottom plate, start attaching the legs:





7. Add in the rest of the legs:



Thread the small piece of Velcro in a loop through and underneath the bottom hub plate. It should form a harness for the battery.

## Evening 2

### Transmitter Firmware Upgrade

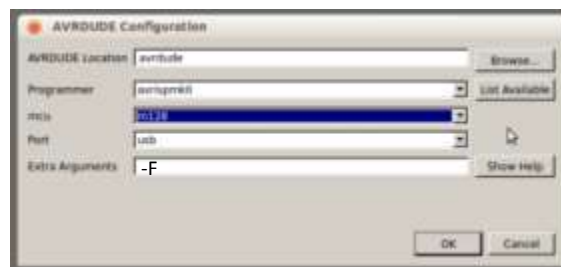
The firmware and settings file for the Turnigy 9XR can be found at <https://canberraav.readthedocs.org/en/latest/quadcopter-workshops/quadcopter-workshop-2.html>. Download these files and use then as needed in the following section. The firmware can be found at <http://code.google.com/p/er9x/source/browse/trunk/er-128.hex>

Use the AVR programmer to connect the Transmitter's 6-pin connector to the laptop's USB port:



1 - The RC Transmitter connected to the AVR Programmer

Connect the USB side to a laptop and open up Epee (a `sudo` may be required under Linux). Go to Burn -> Configure. Change the programmer to `avrispmkII`, mcu to `m128` and add a “-F” to the additional options.



2 - eePe programmer setup

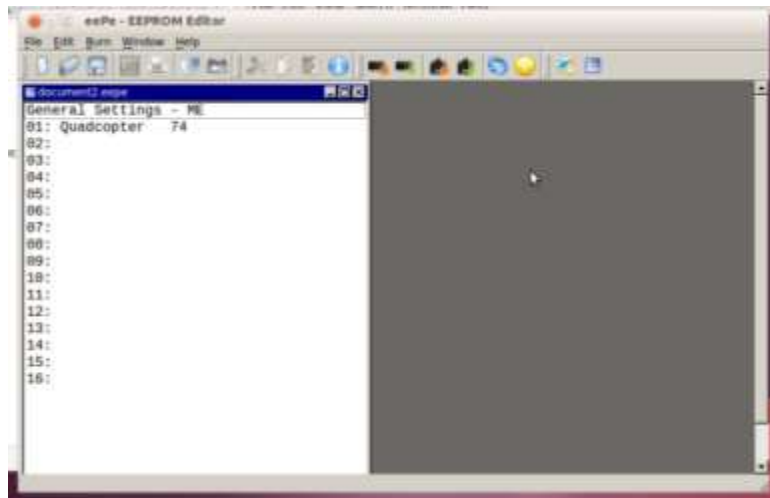
Press the “burn” button on the main screen and select the supplied firmware file to upload.



3 - The eePe software on Windows

Wait for the program to complete and confirm that there were no errors during the flashing process.

To upload the settings file, go to File -> Open and select the settings file. Under General Settings, change the owner name to something unique (like your name).



4 - eePe EEPROM editor

Then go Burn -> Write Memory to TX.

Wait for the program to complete and confirm that there were no errors during the flashing process.

After flashing, remove the programmer and confirm that the new firmware is working correctly (you may get a few errors the first time you start it up, this is normal).

## Transmitter setup

**Note this section is simply a reference for manually changing the settings of the Transmitter to match the contents of the 9XR settings file in the previous section. If you have already uploaded the settings, you can skip this section.**

Turn on the transmitter.

To go to the global settings menu, press the LEFT button

Use UP/DOWN to cycle through the settings and LEFT/RIGHT to change the selected setting.

The settings that need to be changed:

- Owner Name = <your name>
- Beeper = NoKey
- Battery Warning = 9.9 V
- Splash Screen = OFF
- Splash Name = OFF
- Throttle Warning = OFF
- Switch Warning = OFF
- Mode = 1

Go to the calibration page:

And calibrate the sticks.

Press exit to get back to the main screen.

Press RIGHT to get to the model profiles. A number of different profiles (with their own settings) can be set up here. We need to create a profile for the quadcopter.

Press RIGHT again to get into the settings for MODEL01.

The settings for it should be:

- Name = Quadcopter

Go across to the MIXER page:

The channels should be setup as such:

- CH1 = 100% AIL
- CH2 = 100% ELE
- CH3 = 100% THR
- CH4 = 100% RUD
- CH5 = 61% HALF ID0
- -76% HALF GEA
- 10% HALF ID2
- 36% HALF ID1
- CH6 = 100% FULL TRN
- CH7 = 100% FULL ELE

Go across to the LIMITS page

Set CH2 to INV

At this point, your transmitter is set up with its modes:

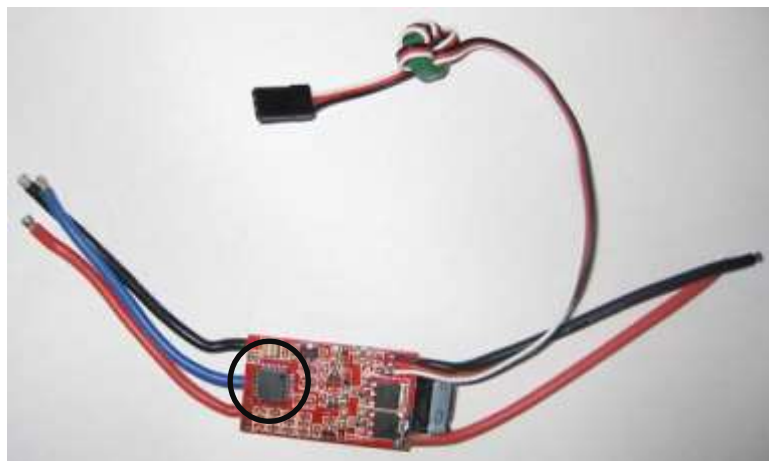


5 - Channel layout on the RC Transmitter

## ESC Firmware Flashing

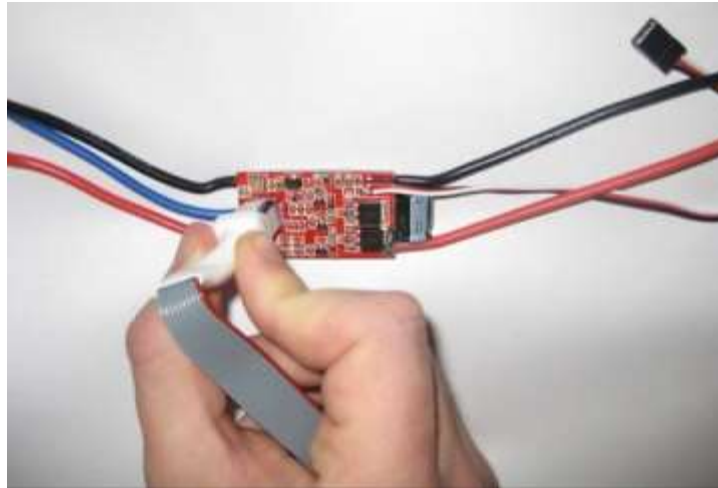
Note: For the purposes of this workshop, a computer will be setup with the flashing software ready to go.

Carefully cut off the heatshrink material, taking care not to damage the components underneath. Try cutting from the side with a knife



6 - Location of microcontroller on the ESC

For the purposes of this workshop we are using a dedicated firmware programmer device. Using this device, line it up over the microcontroller on the ESC (taking note of the correct orientation). Hook the programmer to a laptop and load the SimonK firmware.



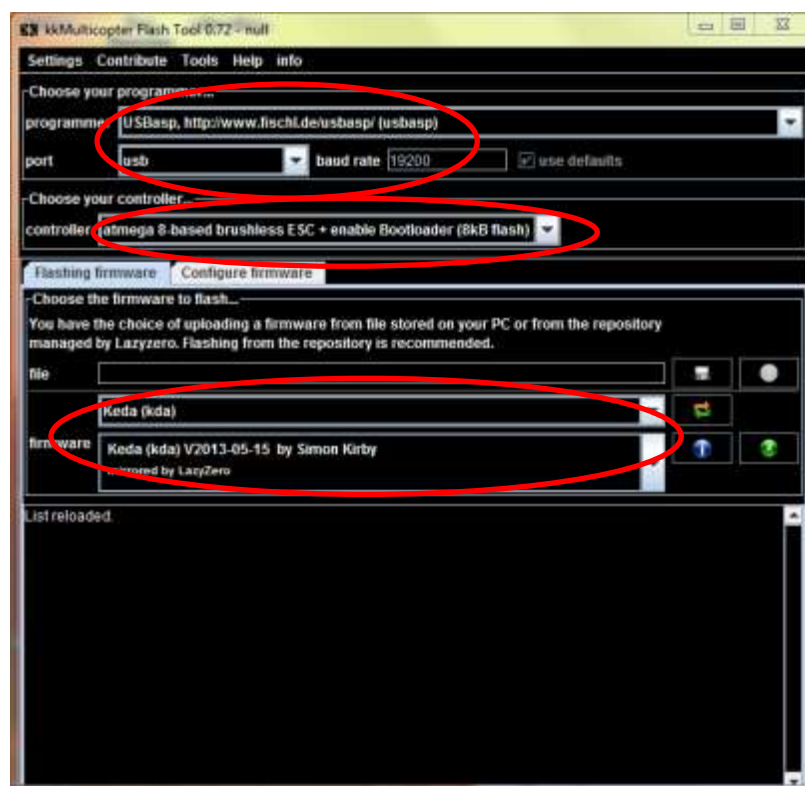
#### 7 - Using the firmware flasher on the ESC

Use the “KKMulticopter Flashtool” from <http://lazyzero.de/en/modellbau/kkmulticopterflashtool#download>.

If you are running windows, ensure to install the USBASP driver first.

The black corner of the firmware flasher aligns to the pin 1 corner (the corner on the IC with the dot).

Use the “Keda (kda)” firmware variant with the “atmega 8-based brushless ESC + enable Bootloader”



#### 8 - kkMulticopter Flash Tool setup for ESC flashing

Under Linux, grab the Github download of the SimonK firmware (<https://github.com/sim-tgy/downloads>) and look for the “kda.hex” variant. This is the correct firmware for the Multistar 20A ESC’s. Use the following shell command to upload it:

```
avrdude -c usbasp -p m8 -U flash:w:kda.hex
```

(a `sudo` may be required on some systems)

Once programmed, apply (new) heatshrink to the ESC. Repeat for all 4 ESC’s.

Ignore any errors that say “warning: unable to sck period”. The ESC will still be programmed.

## Simulation

The RC transmitter can be connected (via the supplied USB dongle) to your PC and used as a standard joystick. Combined with the (various) RC flight simulators, it can serve as a decent training module for new pilots and you can practice your flying skills in a safe environment.

Simply connect the (audio stereo) cable to the rear of your RC transmitter this cable plugs into the dongle, which in turn connects to a USB port. The TX module must be removed (the box module on the back of the transmitter) and the power switch of the RC transmitter must be off.

There are a few options for RC Simulators. The most popular are:

- Realflight (<http://www.realflight.com/>) which is commercial
- CRRCSim ([http://sourceforge.net/apps/mediawiki/crrcsim/index.php?title=Main\\_Page](http://sourceforge.net/apps/mediawiki/crrcsim/index.php?title=Main_Page)) which is open source.

Either will work well. In the case of this workshop we will be using CRRCSim.

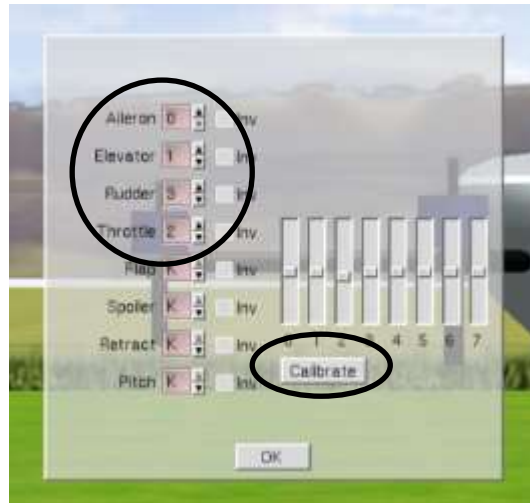
Open CRRCSim, press the `esc` key to get the main menu. Go to Options -> Controls -> Input Method and enable the RC Controller:



9 - CRRCSim Input Selection

Then go to Options -> Controls -> Configure. Select the correct channels for the sticks and calibrate the controller.





10 - CRRCSim calibration page

Go to Options-> Aircraft and ensure the x-config quadcopter (attitude controlled) is selected:



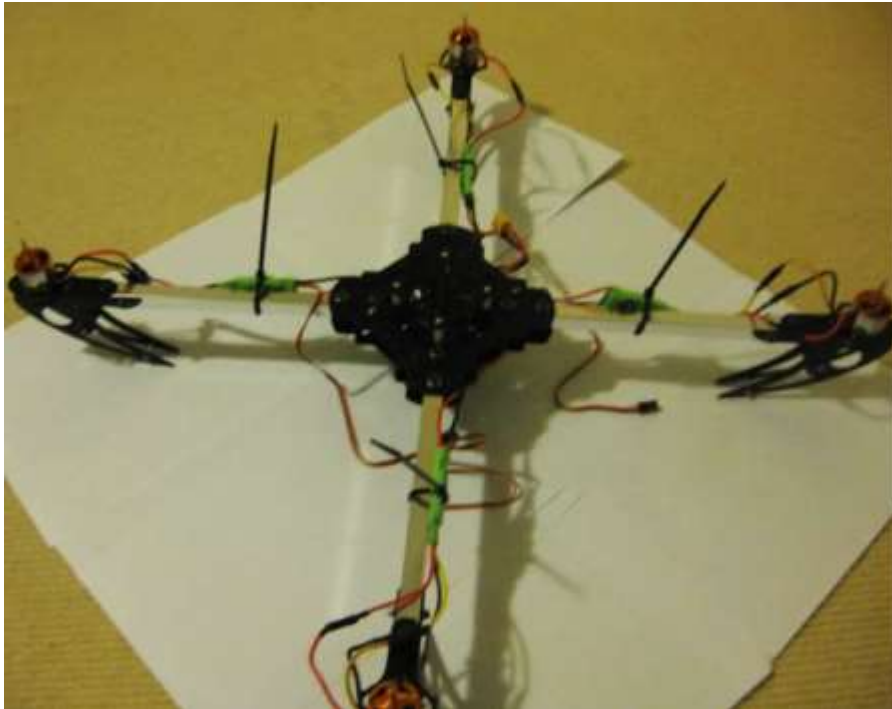
11 - CRRCSim model selection

Return to the simulation and test that the RC Transmitter controls the simulator correctly. Use this simulation to get a feel for the quadcopter's controls and movement.

## Wiring up the system –Power train

Connect the ESC's up to the motors and thread the power distributor through between the hub plates and hook it up to the ESC's. Wire ties or tape can be used to affix the ESC's to the arms. They ESC's should be set back from the hub in order to reduce electrical interference to the PX4.





The ESC <-> Motor connections should be left loose for the moment, as they may need to be changed later (to set the correct motor direction).

## Propeller Balancing

Given how fast the propellers on the quadcopter spin, it is important that the propellers are balanced (equal mass on both sides). If they are not, the quadcopter may have a slight spin or drift when it is flying.

To balance them, you need a propeller balancer:



12 - Propeller balancer

Simply place the propeller in the rod and watch for movement. If one side of the propeller is clearly heavier, use some sandpaper to (gently) rub some mass off the top (the side with the embossed writing on it) of the propeller until it is balanced.

Repeat for all 4 propellers.

## Radio binding (and servo test)

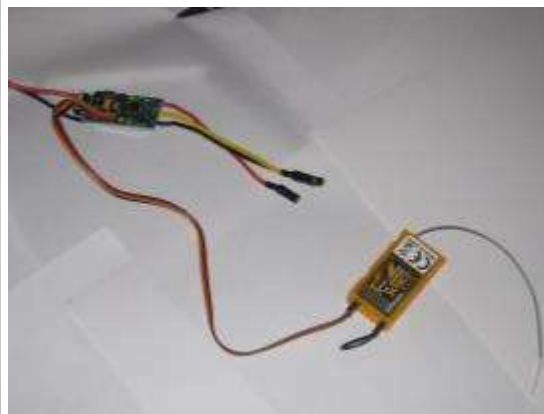
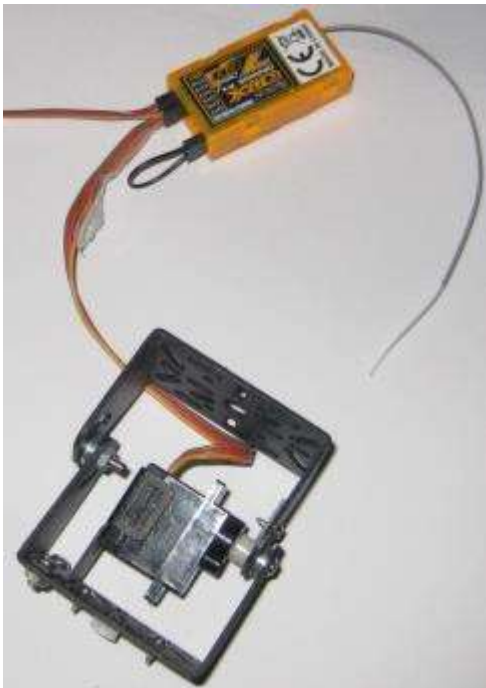
Binding is the action of syncing your transmitter and receiver. Due to the nature of the process, this can only be done one at a time.

For the servo cables: The brown cable goes to the top, red in the middle and orange on the bottom

1. Plug in the TX Module to the Transmitter and connect the antenna. NOTE: Do not turn the TX Module on without the antenna connected – you could damage the TX Module.



2. Hook up one of the ESC's to the RC receiver to provide power. Attach a servo to channel 3.



3. The binding wire (the loop) should be plugged in to the bind plug on the RC receiver.
4. Power up the RC receiver by connecting the flight battery to the main plug on the quadcopter. The RC Receiver should be flashing at a fast rate
5. Whilst holding the bind button on the underside of the RC transmitter, switch it on.
6. Wait until the light on the RC receiver stops flashing. This take around 10 seconds
7. Turn off the RC transmitter, then the RC receiver.
8. Remove the bind plug from the RC receiver

Turn the RC receiver and transmitter back on. Confirm the binding works by twiddling the sticks and watching for movement from the servo.

Perform a range test on the Transmitter by walking 15-20m from the RC Receiver and pressing the bind button (low power mode on the RC Transmitter) and confirming the servo will still respond.

## Evening 3

### Safety

#### Current Australian Law and Regulations

Any UAV (no matter the size) being used for commercial operation requires a UAV Operator's Licence (expensive!). Otherwise, the UAV is covered under the Model Aircraft rules.

The Model Aircraft Rules specify:

- Keep your model aircraft away from populated areas and the immediate vicinity of others (i.e. more than 30 metres)
- Do not operate within 3nm (5.5 km) of an aerodrome without approval
- Do not operate in controlled airspace above 400ft (120m) without approval
- Do not operate in military prohibited or restricted areas without approval
- Do not fly in poor visibility, clouds or at night

The MAAA (Model Aircraft Association of Australia) are the recognised association for model aircraft pilots. They are mostly focussed on fixed wing aircraft, but can offer training (and airfields) for the use of members.

When flying at the local oval or in public places, it is **very** important to realise that if you crash into a person/car/house/etc, **you put yourself at risk of legal action and may have to pay compensation to the affected party.**

The general rules are:

- Do not buzz or fly near people without their OK.
- Consider your skill level, nearby obstructions and weather conditions in estimating your "safe zone".
- Don't rely on the APM to get you out of trouble.
- Only fly in situations where you are confident of manual recovery.
- Maintain a line of sight to the quadcopter.
- Watch out for trees!
- Windy conditions can really toss a quadcopter around. Take this into consideration when flying.
- If flying with a camera, be mindful of people's privacy.
- Don't fly over private property without the landowner's permission.

Note that, like most countries, Australia is in the process of reforming its laws and regulations regarding UAVs. Media coverage of people performing stupid or dangerous manoeuvres with UAVs

will only encourage the authorities to make the regulations more stringent, or outright ban amateur UAVs.



13 – RC Aircraft crash (from [www.theoildrum.com](http://www.theoildrum.com))

### LIPO Batteries

The Lithium Polymer (LiPo) batteries that quadcopters use contain large amounts of energy in a small and light package.

Due to the nature of the chemistry in the batteries, they can catch on fire or explode if not maintained properly.

LiPo guidelines:

- When not in use, keep in a fire-resistant enclosure.
- Never charge a battery unattended.
- If a battery looks damaged or puffy - discharge it, cut off the leads and throw it away.
- Do not discharge the batteries below their minimum (use a battery alarm)
- Make sure you use a LiPo charger
- Keep a bucket of sand handy to put out a LiPo fire (do NOT use water to put out a LiPo fire).



14 - Results of a LiPo fire (from [www.rctech.net](http://www.rctech.net))

### Propellers

The propellers on a quadcopter spin at many 1000's RPM. They can cause serious injury if they were to hit a person.

It is thus very important that all people near an active quadcopter are aware of the quadcopters position at all times and are prepared to run/duck as needed to avoid getting hit by it.

The APM features an arm/disarm switch. When disarmed, the motors will not spin under any circumstances (barring a major software error in the APM). **The APM should only be armed at takeoff and immediately disarmed after landing.**

### Radio Interference

Due to the large numbers of radios operating during the workshop, it is important that any radio transmitters (RC Transmitters, Ground station radio and Quadcopter radio) are turned off when not in use. This will lower the chance of harmful interference to other users.

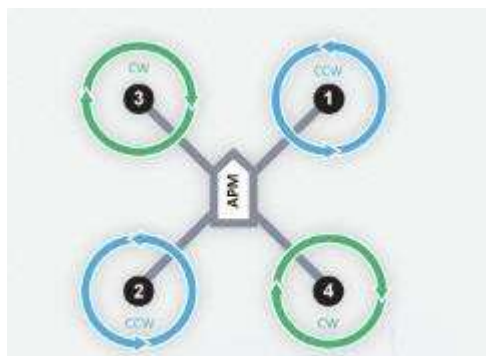
### Soldering

- Battery charger XT60 connector
- Bullet connectors to PX4 power in
- Cut power lines to ESC's (unless there's a jumper on the PX4)

### Wiring up the system – Flight Electronics

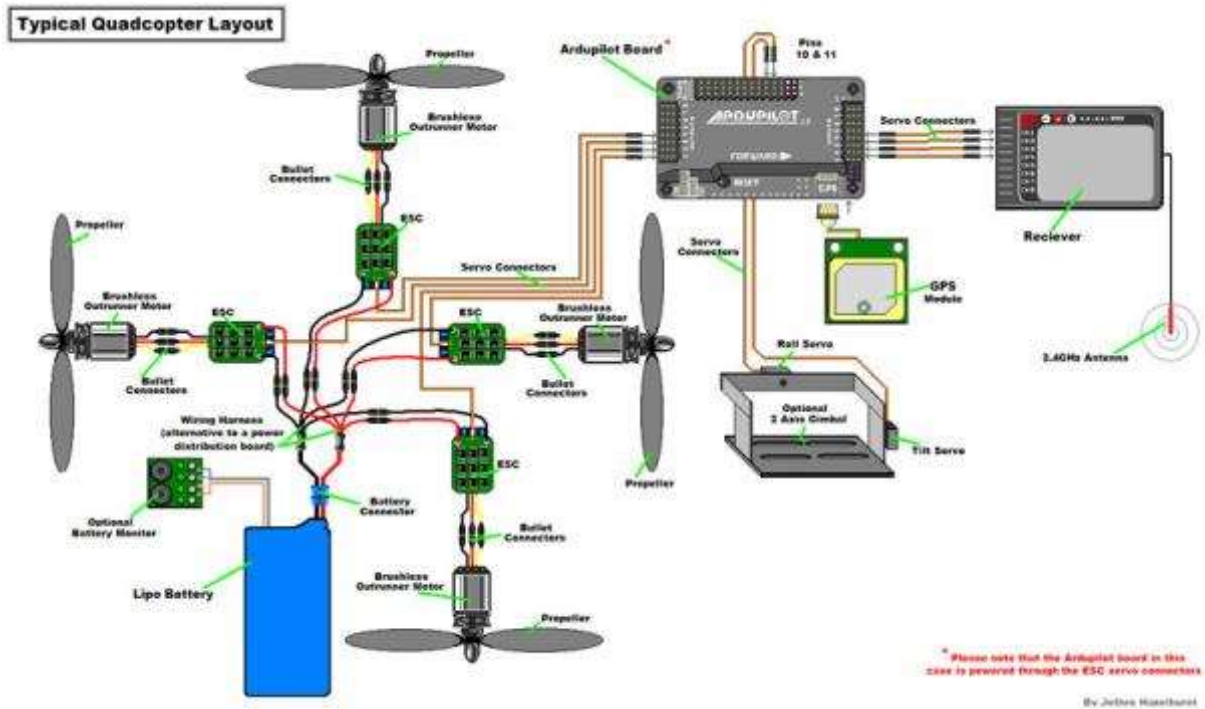
There are a number of wire connections that need to be made:

- GPS <-> PX4 – this uses a polarised cable. The GPS module should be as far away from the other radio transmitters as reasonably possible.
- Telemetry radio <-> PX4 – this uses a polarised cable
- RC Receiver -> APM (input) – use the 3cm servo cables, taking care that the signal (top/white cable) line goes between all 8 channels. Only 1 power (middle/bottom) link needs to be made:
- APM (output) -> ESC's – take careful note of which numbered motor goes to which APM output (see below). It's useful to use a marker to write the motor number and direction of motion on each arm



15 - Motor and propeller directions (from <http://code.google.com/p/arducopter/wiki>)

Generally, the power system looks like:



16 - Quadcopter power system (from <http://code.google.com/p/ardupiloter/wiki>)

## Testing the ground station

Hook up the USB cable to the APM. This will provide enough to power the flight electronics, but not enough to activate the ESC's/motors.

Note that the PX4 board and the IO board require separate firmwares.

First, update the firmware:

## APM Mission Planner



Follow the instructions in APM Mission Planner for the two firmwares. The main PX4 firmware should be upgraded first.

More info at

<http://copter.ardupilot.com/wiki/common-px4-firmware-installation-with-mission-planner/>

## Linux

As a first off, you will need to download a copy of the latest firmwares:

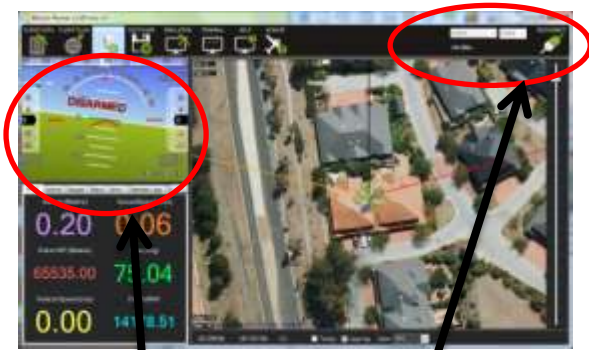
<http://firmware.diydrones.com/Copter/stable/PX4-quad/> and  
<http://firmware.diydrones.com/PX4IO/latest/PX4IO/>

<not sure what the actual process is under linux>



Then connect using the ground station. Rotate the quadcopter and check the roll/pitch values look sensible:

### APM Mission Planner



Watch for  
roll/pitch changes

Connect here. Baud  
rate is 115200 for  
USB cable, 57600  
through radio

### MAVProxy

cd to the mavproxy directory

```
mavproxy.py --baud=115200 --  
console
```

It will auto-detect the USB-Serial port  
(assuming you only have one connected).  
Otherwise use the `--master=<port>`  
argument

Look at the roll pitch angles and confirm they  
look right.



Next, the telemetry radios need to be configured (<http://code.google.com/p/arducopter/wiki/3DRadio>).

### APM Mission Planner

Make sure you are not connected to the APM.

Go to Hardware -> Optional  
Hardware -> 3DR Radio:



Change the NET ID number to a random number between 0 and 65536.

Check the below settings list to ensure all your settings are correct.

The settings should be:

- Frequency – 91500 to 928000
- # of (hopping) channels >= 20
- Tx Power 20 (dBm)
- Net ID = random unique number. Use this to differentiate yourself from all the other transmitter/receiver pairs. Anyone with the same NET ID as you can read/command your telemetry data.

Make sure the local and remote radios are both changed before pressing the “Save settings” button.

The radios will need to be rebooted before the new settings take effect. Connect to the APM via the radios and confirm that you’re getting the telemetry.

### MAVProxy

Go to a serial console (telnet, etc) and connect to the radio

Type +++ to enter command mode

ATI5 to show settings (local)

RTI5 to show settings (remote radio)

ATS4=10 to change setting **4** to value **10** (local), for example.

RTS4=10 to change setting **4** to value **10** (remote), for example.

AT&W to save settings to EEPROM (local)

RT&W to save settings to EEPROM (remote)

ATO to exit command mode

Check the below settings list to ensure all your settings are correct.

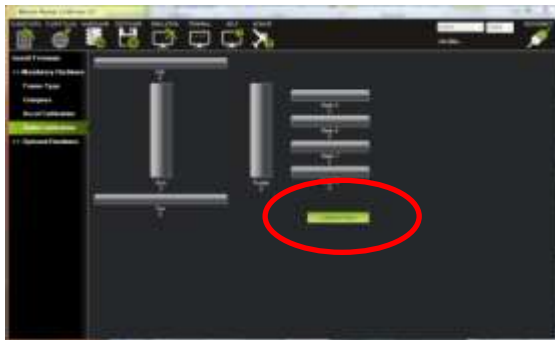
## PX4 Calibration

Connect to the PX4 and go to Hardware -> Mandatory Hardware -> Radio Calibration:

Waggle the RC sticks and buttons to confirm they all work.

Go through the “*Calibrate radio*” process so the APM knows the min/max extents of your RC controller:

### APM Mission Planner



### MAVProxy

```
mavproxy.py --baud=115200 -  
console -setup
```

```
ArduCopterMega] setup
```

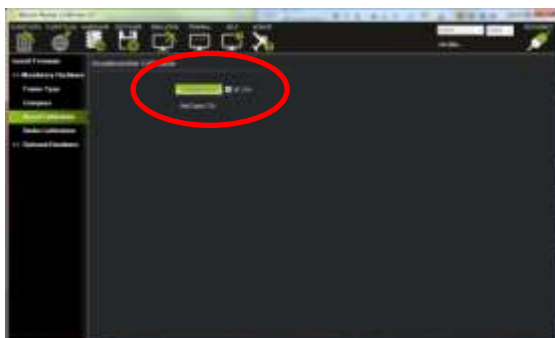
```
Setup] radio
```

<follow instructions>

Next the accelerometers need to be calibrated. Go to Hardware -> Mandatory Hardware -> Accel Calibration:

Go through the “*Calibrate Accel*” (Make sure “AC 3.0+” is selected) process so the APM can account for any accelerometer offsets:

### APM Mission Planner



### MAVProxy

```
mavproxy.py --baud=115200 -  
console -setup
```

```
ArduCopterMega] setup
```

```
Setup] accel
```

<follow instructions>

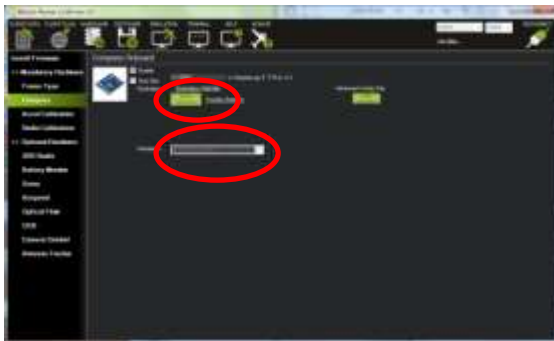
Next the compass need to be calibrated. Go to Hardware -> Mandatory Hardware -> Compass:

Untick the “Auto Dec” option and put in the declination as “12.19” for Canberra. This number will vary for other locations around the world.

In the Orientation option, select “ROTATION\_ROLL\_180”. This is because the compass sensor is on the underside of the GPS/Compass module.

Go through the “Live Calibration” procedure. When moving the Quadcopter around for the 60 seconds make sure to roll it around all axes.

### APM Mission Planner



### MAVProxy

```
mavproxy.py --baud=115200 -  
console -setup
```

```
ArduCopterMega] setup
```

```
Setup] set COMPASS_DEC 12.19
```

```
Setup] set COMPASS_ORIENT 8
```

```
Setup] set AUTODEC 0
```

```
Setup] compassmot
```

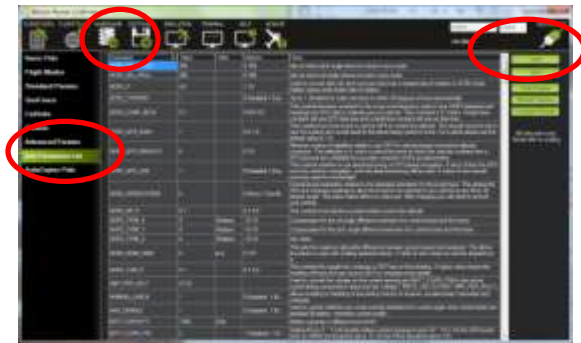
<follow instructions>

The settings for the APM are known as the parameters (or params for short). They can be exported/imported the APM via a simple text file.

To make things easier for everyone, we will be loading a set of known parameters. They are available from <https://canberraauav.readthedocs.org/en/latest/lessons/quadcopter-workshop/MHV.param>

They can be loaded by:

### APM Mission Planner



### MAVProxy

```
mavproxy.py --baud=115200 -  
console
```

```
param load MHV.param
```

- Connect to the APM
- Go to Software -> Adv Parameter List
- Press *Load* to open the parameters file
- Press *Write* to write the parameters to the APM

These parameters contain:

- Frame/engine tuning parameters
- Mode selection for the switches on the RC Transmitter
- Failsafes
  - GPS failure. The Quadcopter will not ARM if it does not have a GPS lock.
  - On loss of RC control signal, it will return to land at the ARming point.

Finally, to ensure your APM is positioning itself correctly, look at the position and orientation on the quadcopter in the main screen of the APM Mission Planner (or MAVProxy, use `module load console` and `module load map`). Check that it looks correct

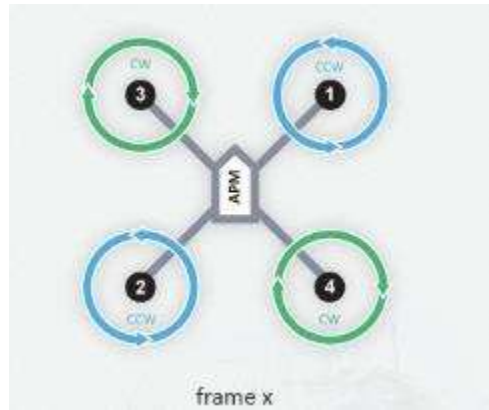
## Motor and ESC tests and calibration

**Make sure the propellers are not attached for this section**

Hook up the battery to the APM and power up the quadcopter.

Push rudder (left stick) right for 4 seconds to ARM the quadcopter. Increase the throttle to 1/5 and confirm that all the motors spin.

Carefully put your finger on the motor (or use your eye) to determine which way the motors are spinning. It should match this:



17 - Motor and propeller directions (from <http://code.google.com/p/arducopter/wiki>)

If they do not, reduce throttle to 0 and DISARM push rudder to the left for 4 seconds. Confirm the DISARM by pushing the throttle up – the motors should not engage. Swap (any) two power leads (ESC $\leftrightarrow$ motor) of the affected motors.

ARM the quadcopter and throttle up again to confirm the motors now spin correctly.

Next the ESC's need to be calibrated to match the signal levels given out by the RC transmitter. It is done by:

1. Disconnect USB and battery. The entire system must not have power
2. Turn your RC transmitter on and put the radio throttle stick at maximum.
3. Connect the battery to power the quadcopter. You will probably hear one short musical tone from the ESCs. (but NOT two beeps after the short musical tone - if you get two beeps after the short musical tone and the motors are live it's probable that your throttle needs to be reversed)
4. When the APM boots, the ABC LED's will cycle slowly and continuously
5. Leaving the throttle stick at full, disconnect the battery and reconnect it again to reboot the APM and power the quadcopter. The APM will now pass the radio signal directly to the ESCs, triggering calibration. The ABC LED's will cycle quickly - You will probably hear the short musical note and then 2 beeps. After another 1 beep there is a pause, during this pause drop the throttle to full down. You should hear confirmation/arming beeps.
6. Do not unplug, move the throttle up a little to confirm all ESCs are armed and the motors are working in sync. It is important that all motors start/stop at the same time.
7. Your ESC's are now calibrated.

## Build Checklist

Check that all of the following activities have been completed before flight

- Transmitter settings loaded and calibrate
- Receiver bound to transmitter
- Telemetry radio NET\_ID changed
- ESC's calibrated
- All propellers give an upward force when spun up
- PX4 settings loaded

- PX4 accelerometers, RC channels calibrated
- PX4 compass calibrated
- Confirm GPS lock

## Evening 4

### First steps

Activate the ground station software and connect to the PX4. Confirm that telemetry is downloading.

Add the propellers and battery to the quadcopter and give it ~20 seconds to boot. If you have your laptop handy, launch your ground station software and connect to the quadcopter via the telemetry radio links.

ARM the quadcopter by tilting the left stick to the right for 4 seconds. The lights on the APM will stop flashing when it is armed. The ground station will also show that the quadcopter is armed.

For safety reasons, the quadcopter will not ARM if the throttle is not at zero or if it does not have a GPS lock.

Raise the throttle to 25%. Gradually raise it further until the quadcopter takes off. Once it's 1m off the ground, slowly decrease the throttle for a gentle landing.

If the trims on the quadcopter are slightly off it will slip to one side.

Repeat the small hops a number of times until you are familiar with the throttle levels needed for a safe takeoff and landing.

Start making the hops a bit longer and use the throttle to maintain a safe altitude.

### Trims

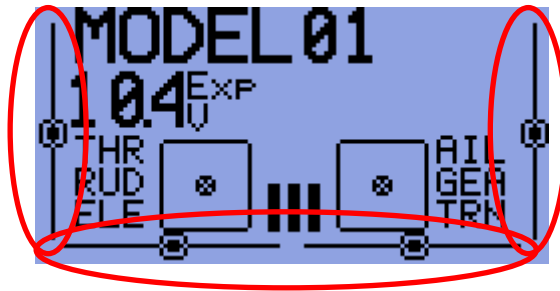
If your quadcopter is slipping to the side a bit, the trims are off. This can be fixed by teaching the APM the correct trims.

Use the trim tabs (the small buttons on the bottom/side of the sticks on the RC controller) to cancel out any side slip.

Land the quadcopter and set the throttle to 0

Set the CH7 button (top left button) to 1 for a few seconds. This will save the current trims

Reset the trims to 0 on your RC controller (there's a home screen mode that will show you your current trim positions)



18 - RC Transmitter home screen. The trim for each channel is shown as the line and dot

Takeoff and repeat this process until the quadcopter had very little/no sideslip

## Longer flights

Takeoff and fly up to 3m altitude.

Use the sticks to navigate the quadcopter around a simple box pattern.

At any time you lose control, switch the quadcopter to LAND mode (top button on right edge)

## Morning 1 and 2

### LOITER

LOITER mode will keep the quadcopter at the current lat/long/altitude (within a few metres) and will compensate for any wind.

Test this mode out and use your sticks to change the lat/long/altitude setpoint.

### RTL

On your ground control station, switch the Channel 7 function to RTL (via the Configuration -> Aducopter Pids -> Ch 7 Options).

Fly the quadcopter around for a bit and switch the Channel 7 to on. The quadcopter will then fly to 2m altitude, fly to the point at which the quadcopter was ARMED and land. This mode is very useful if you ever lose control of your quadcopter.

### ALT\_HOLD

This mode will attempt to keep the quadcopter at the current altitude, whilst giving full roll/pitch/yaw control to the pilot.

Only switch to ALT\_HOLD mode when you are level at the altitude you want to maintain. It may take up to 30 seconds for ALT\_HOLD to come into full effect.

### AUTO

This mode is fully automatic, with no input needed from the pilot.

AUTO mode requires a *mission* (set of waypoints) in the APM's memory in order to function.

This is a simple point-and-click affair on the APM Mission planner (left click on the map to add points, right click for advanced options):



In this case the quadcopter will fly to waypoints 1, 2 and then 3. The DO\_JUMP waypoint at the end tells the APM to go to waypoint 1. Thus it will repeat the set of waypoints until told otherwise. Despite what the above image indicates, it will NOT go to the home point.

There are numerous waypoint types to tell the APM to change speed/altitude, loiter at a position for a period of time, return to land and others.

Of particular note are the altitude settings (red circles). Double check them before you write the waypoints to the APM. The radius settings tell the APM how close it needs to be to a waypoint to consider it "reached".

Note that all altitudes are relative to the ARM point. Be careful when setting waypoints around hills and other obstacles. Leave plenty of clearance to ensure the quadcopter doesn't fly into any of these objects.

For the purpose of this workshop, plan a simple mission within the confines of the oval, with a Return to Land at the end. Don't forget to write the waypoints to the APM.

You should be able to takeoff and switch to AUTO once you're a metre off the ground. Watch the quadcopter to ensure that it's flying the mission correctly.

If at any time, you are concerned that the quadcopter is not doing what it's supposed to be doing, flick to RTL and let it come home and land.



## References

Arducopter Manual

<http://copter.ardupilot.com/>

APM source code:

<https://github.com/diydrones/ardupilot>

DIYDrones community site

[www.diydrones.com](http://www.diydrones.com)

MHV Quadcopter repository

<https://canberraauav.readthedocs.org/en/latest/quadcopter-workshops/quadcopter-workshop-2.html>

ER9X firmware:

<http://code.google.com/p/er9x/>

ESC Firmware:

<https://github.com/sim-/tgy>