ArduCopter Based on ArduPilot Mega

ArduCopter Quad Introduction

ArduCopter Quad is the first drone created with the ArduCopter UAV Platform, supported and created by DIY Drones and it's community. The platform is based on the ArduPilot Pro Mega (APM) and the APM sensor board currently nicknamed the "APM Shield/Oil Pan".

The ArduCopter Quad joins two really great multirotor projects and create one even better. These two projects was known as AeroQuad and ArduIMU Quad III.

Here's an initial feature list and software road map.

Details

ArduCopter Feature List

- 6 Degree of Freedom IMU stabilized control
- Gyro stabilized flight mode enabling acrobatics (loops and barrel rolls)
- GPS for position hold
- Magnetometer for heading determination
- Barometer for altitude hold
- IR sensor integration for obstacle avoidance
- Sonar sensor for automated takeoff and landing capability
- Automated waypoint navigation
- Motor control using low cost standard PWM Electronics Speed Controllers (ESC's)

- On board flight telemetery data storage
- Mounted camera stabilization capability
- Wireless command & telemetry for long distance communication
- Capability to fly in "+", "x", hexa and octo configurations
- Battery level detection
- User configurable LED flight pattern
- Capability to use any R/C receiver
- ArduCopter Configuration and Ground Control Software
- Realtime graphs of flight data
- GUI for configuration of PID and other flight parameters
- On Screen Display integration
- Waypoint programming using Google Maps
- Mixertable view to auto configure "+", "x", hexa and octo configurations

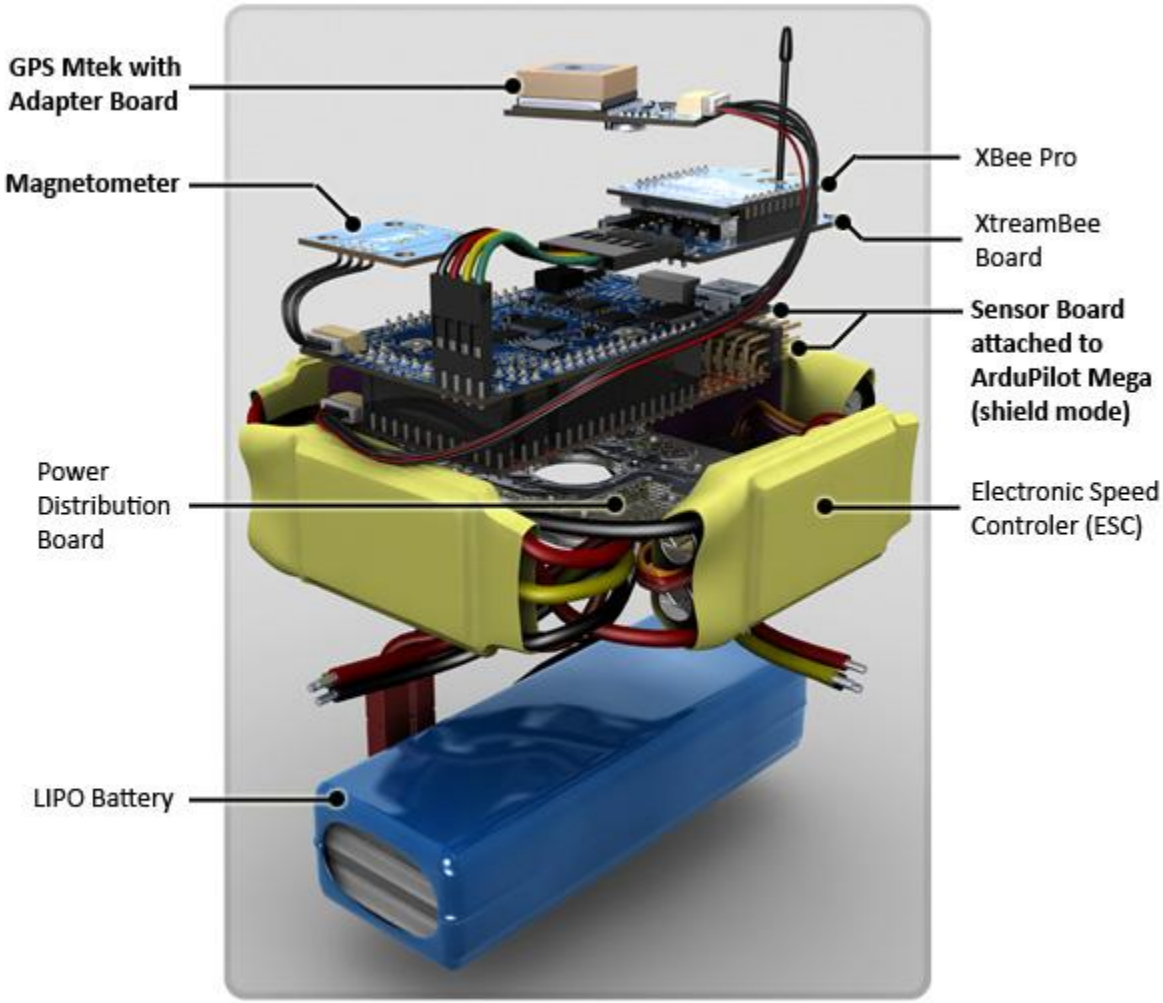
Software Roadmap

- Initial baseline using Jose Julio's v3 software
    - Provides absolute angle PID flight control
    - Obstacle avoidance
    - Waypoint navigation
- Generalize basic ArduCopter functions (ie. Separate PPM receiver input and motor control functions into separate libraries. Allows future coding of PWM vs. I2C ESC's)
- Emphasis on developing new capability into easy to use C++ libraries
- Integrate user defined EEPROM storage capability
- Develop/optimize AeroQuad serial real-time command/telemetry for ArduCopter
- Integrate AeroQuad Configurator for external software configuration of ArduCopter
- Rename Configurator to Ground Control Station and integrate graphical programming of waypoint navigation
- Integrate AeroQuad rate PID control
- Integrate mixertable configuration for multicopter configurations
- Integrate AeroQuad camera stabilization
- Integrate I2C motor control
- Develop capability to wirelessly control ArduCopter directly from Ground Control Station (USB joystick controller from laptop or through waypoint programming)

ArduCopter_Electronics

ArduCopter Electronics

The ArduCopter UAV Platform is based on ArduPilotMega Board, APM Sensor Shield/Oilpan, i2c Magnetometer HC5843 and the GPS MTek (optionally you can add wireless connection using the XBee Pro and XtreamBee Board). The drones based on the ArduCopter UAV Platform could add more electronics accord to their characteristics. See below the ArduCopter Quad example (main components in bold):



Main components list

ArduPilot Mega (flight controller): Buy here.

ArduPilot Mega IMU Shield/OilPan V1.4 (sensor board): Buy here.

Pin Headers (you need 2): Buy here.

Con Headers (you need 2): Buy here.

Servo cable (4-8pcs. needed, recently 6 - to connect your rx): Buy here.

Magnetometer Buy here. Cable: Buy here.

GPS Buy here. Cable: Buy here.

You can also purchase most parts from the AeroQuad store, Sparkfun in the USA, Phi Fun Shop in Asia, Coolcomponents in the UK or Lipoly in Germany (if available).

Quad_GettingStarted…….

ArduCopter Parts List

The ArduCopter UAV is a platform that supports any rotary-wing drone.

Arduino

In order to do any work with the ArduPilot Mega, we need to have Arduino installed.

Go here and on the right side of the screen, there is a green box called ***Featured Downloads***. The ***Featured Downloads*** section will always contain the latest version of Arduino in Mac OS X *(.dmg)*, Linux *(.tgz)*, and Windows *(.zip)*. Alternatively, you can click the ***Downloads*** tab on the top of the page and see all of the previous versions. Once it is downloaded, you should install or extract it to your computer.

The Brain - ArduPilot Mega and IMU Sensor Shield

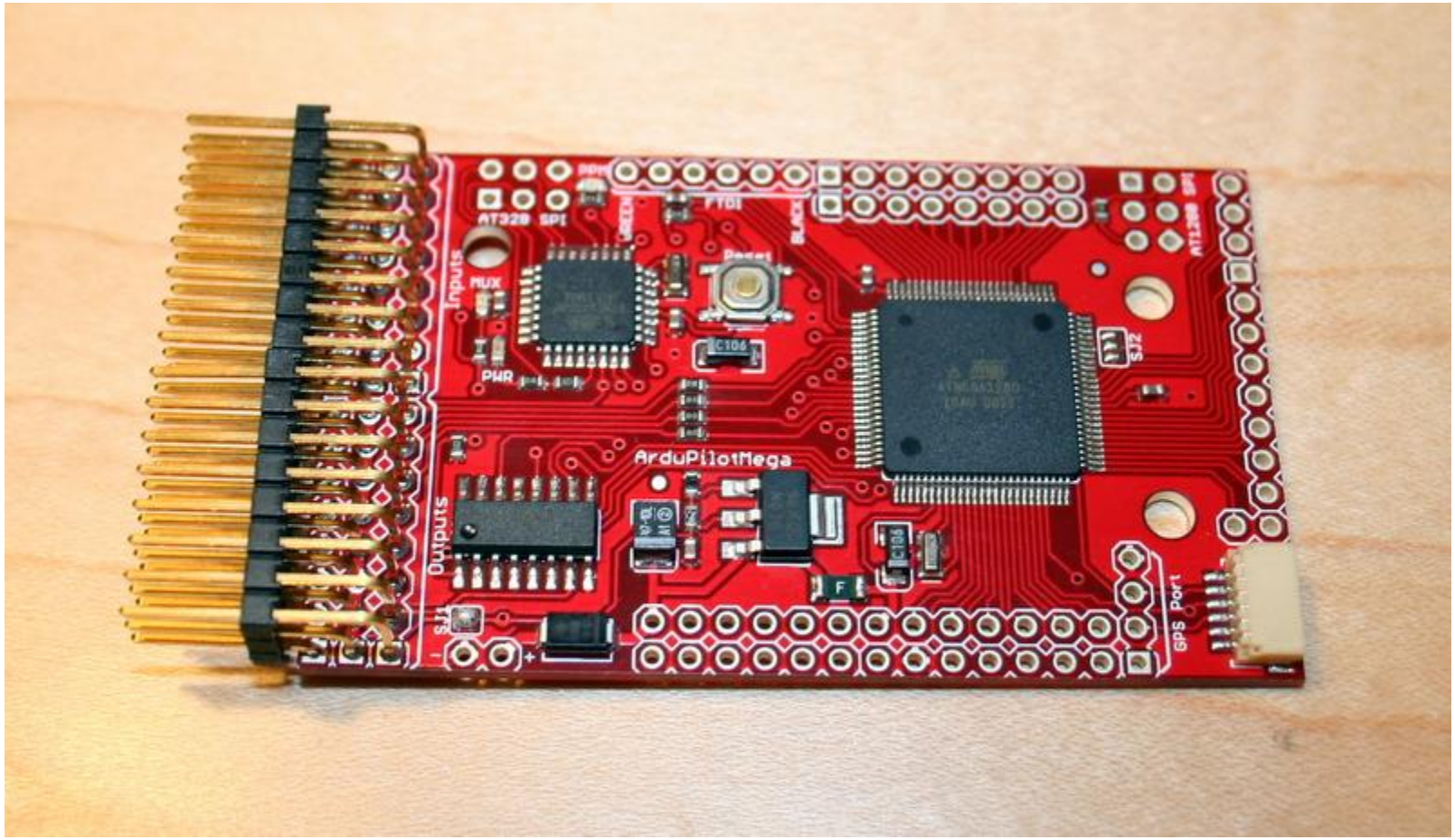
You will need to assemble the ArduPilot Mega *(APM)* and the IMU Sensor Board *(APM Shield/Oilpan)*. The ArduPilot Mega can be purchased here from SparkFun and the IMU can be purchased here from the DIYDrones Store.

You will also need the following:

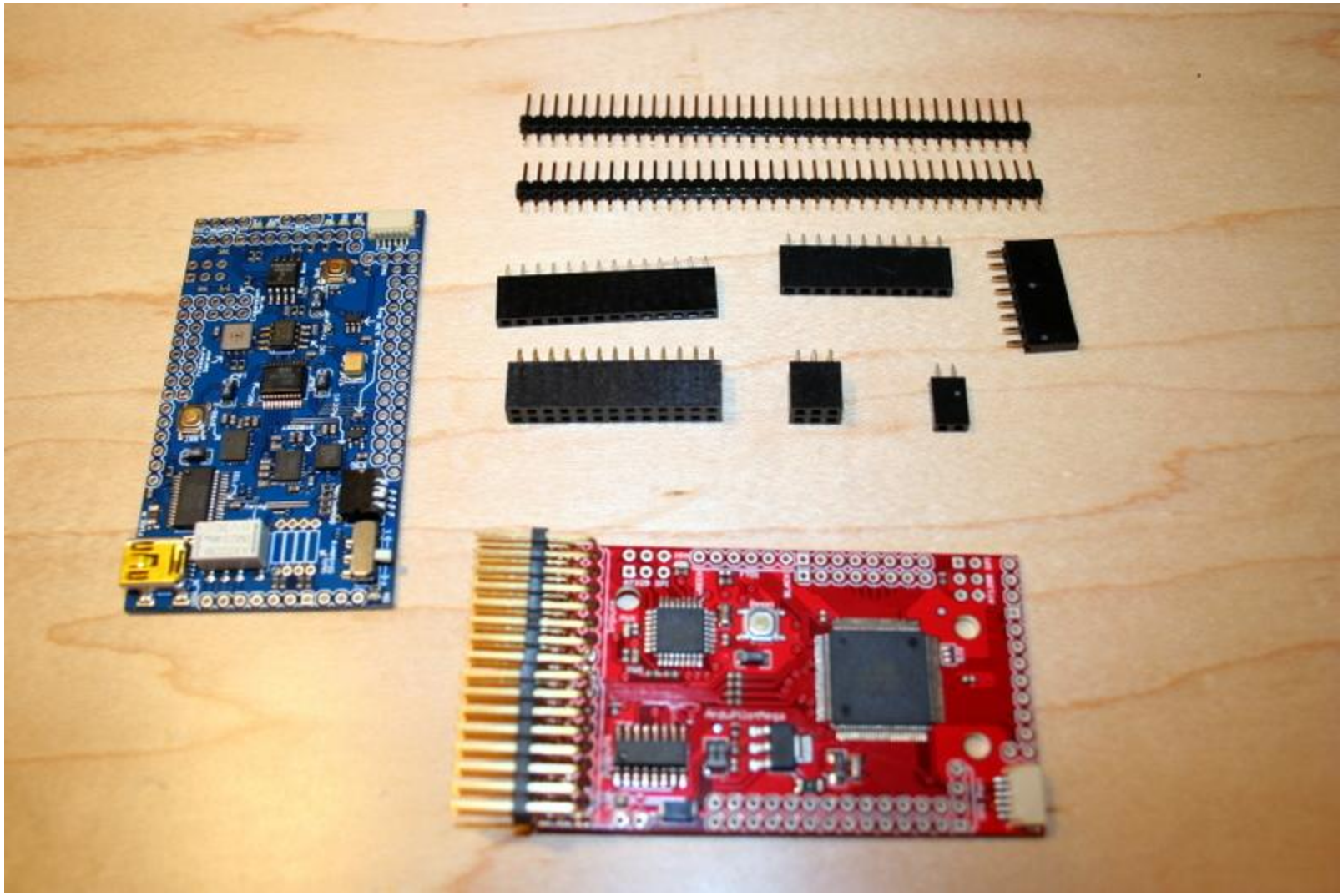
- Two 3x8 right angle headers
- A fine-point soldering iron. The Weller WES51 is the one we use, but you can get cheaper ones at a hardware store or Radio Shack. Just make sure it has a fine tip.
- You'll need some solder, too.
- A GPS module and cable *(we recommend the MediaTek and a cable)*.

Assembly:

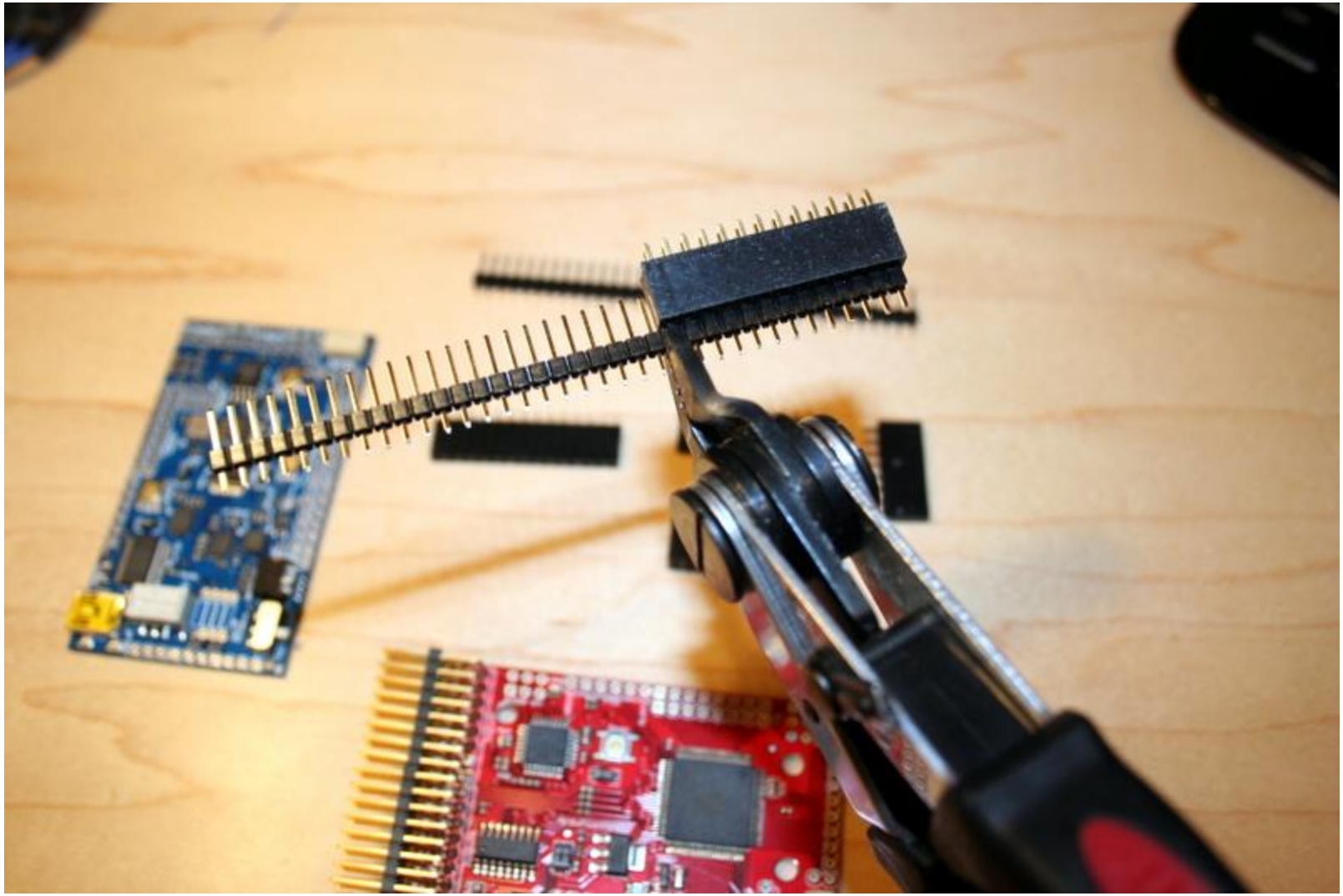
First, solder on the two 3x8 right angle headers for the RC connectors.



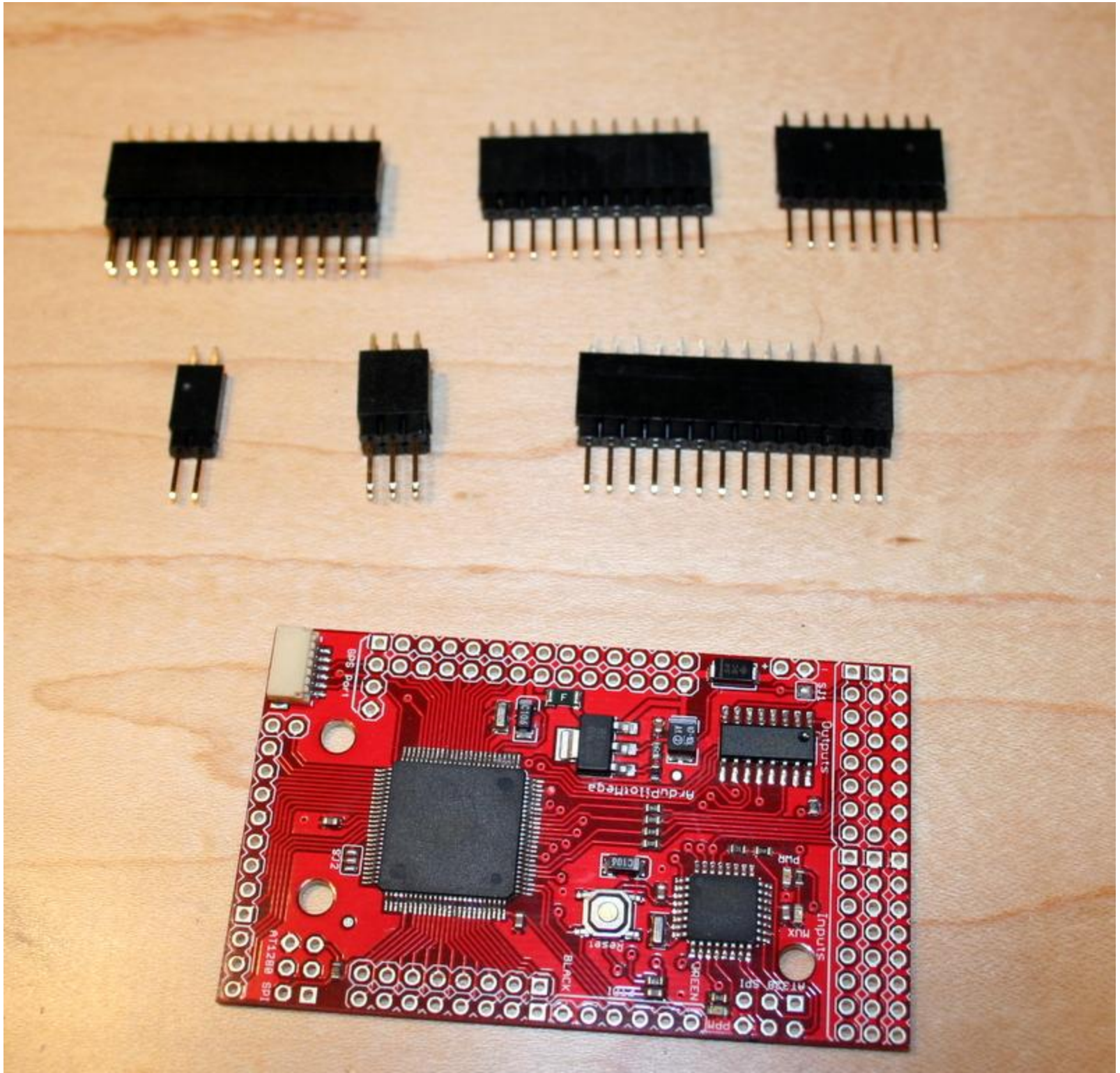
Next, we'll cut the headers to attach the IMU shield. Here are the headers that come with the shield



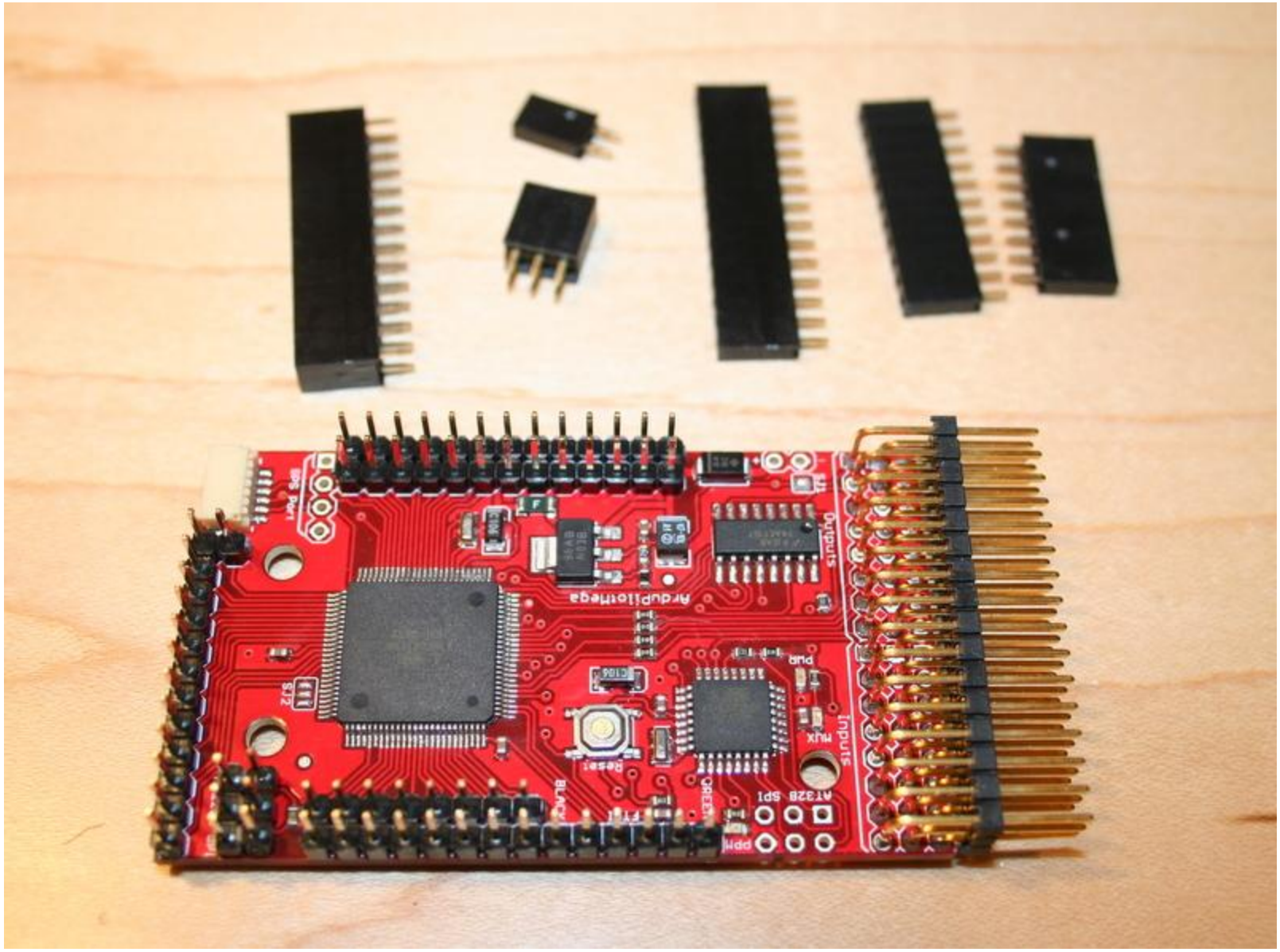
Push a strip of male headers into each row of female headers, then snip off the remaining parts:



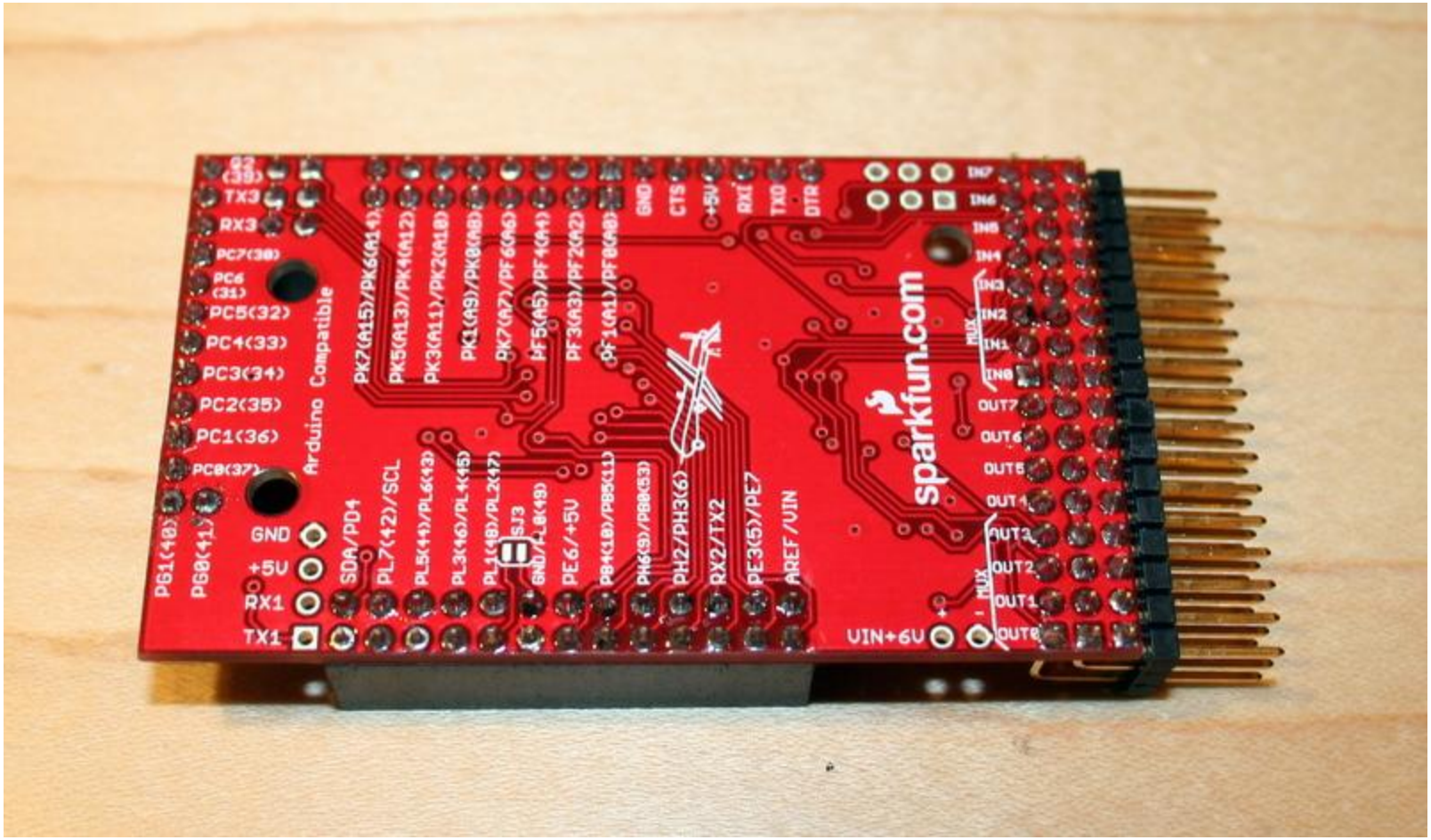
When you're done, you should have one male header for each female header:


ArduPilotMega
Outputs
Inputs
MUX
Reset
PWR
BLACK
SPI

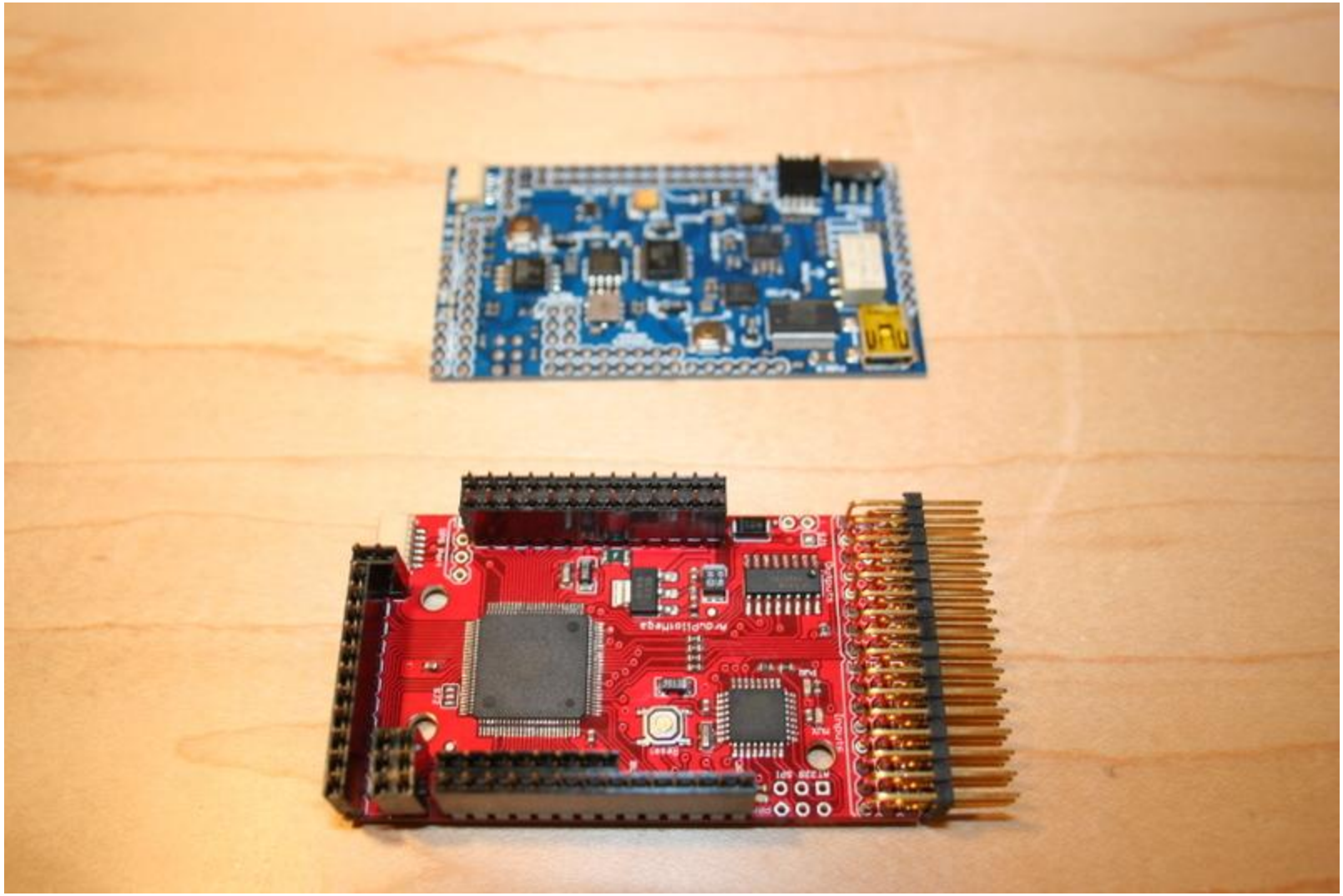
Now remove the male headers and place them in the holes of the ArduPilot Mega board as shown here:



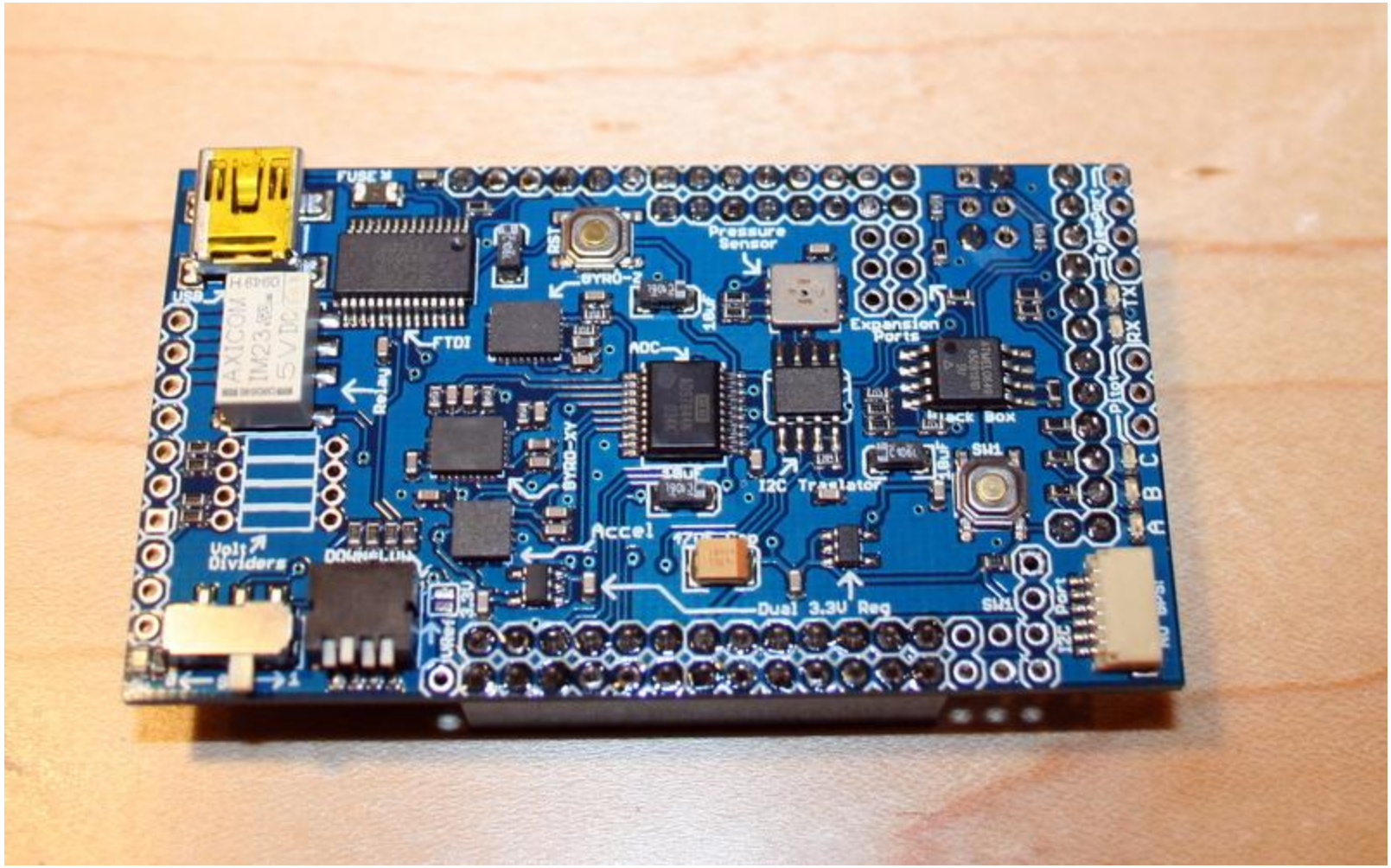
Place a paper notebook on top of the pins and then, holding the board to the notebook, turn it upside down so you can solder the pins on the back of the board.



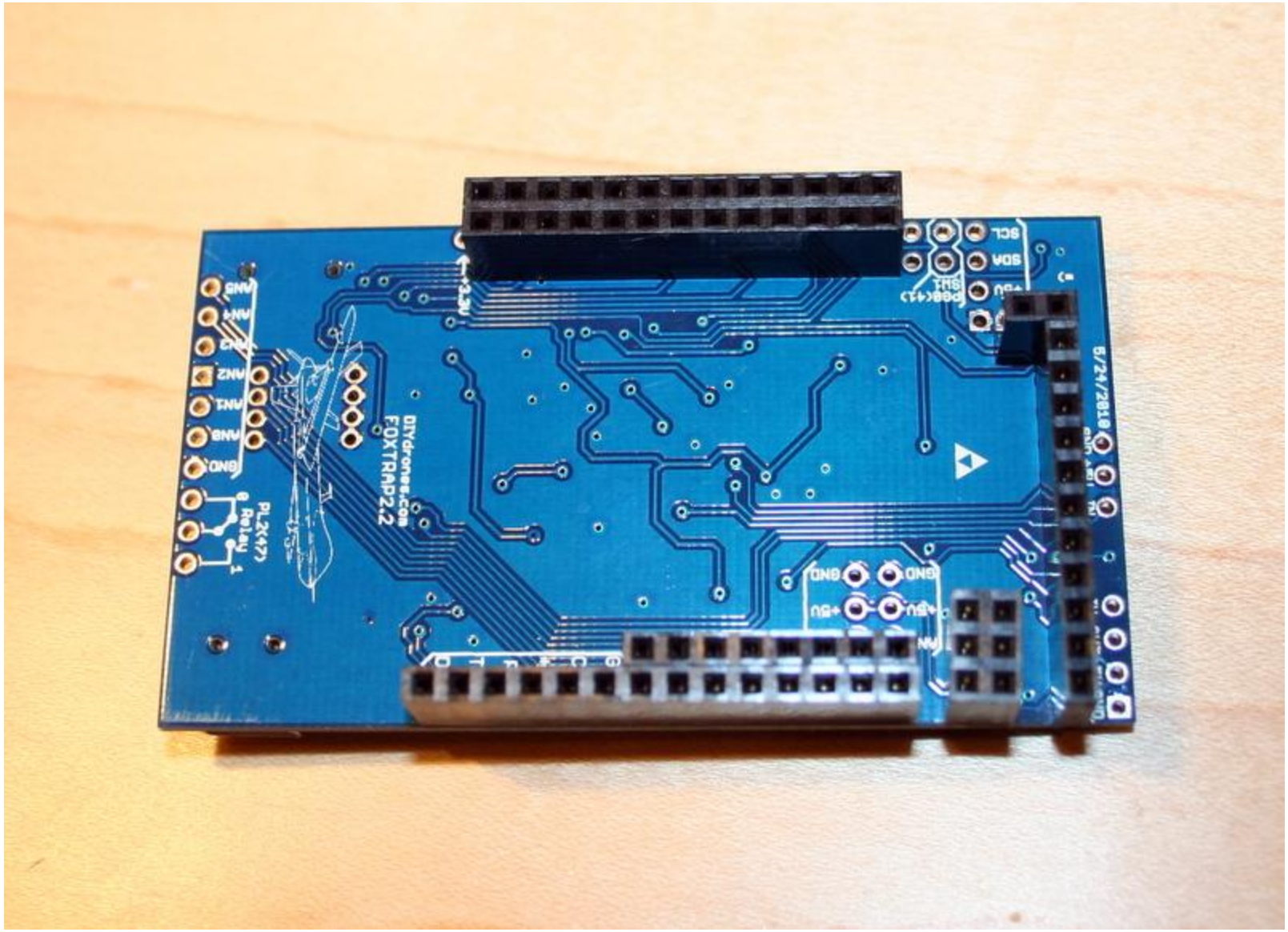
Once you've soldered on all the pins, turn the board back over and place the matching female headers on each row of male headers:



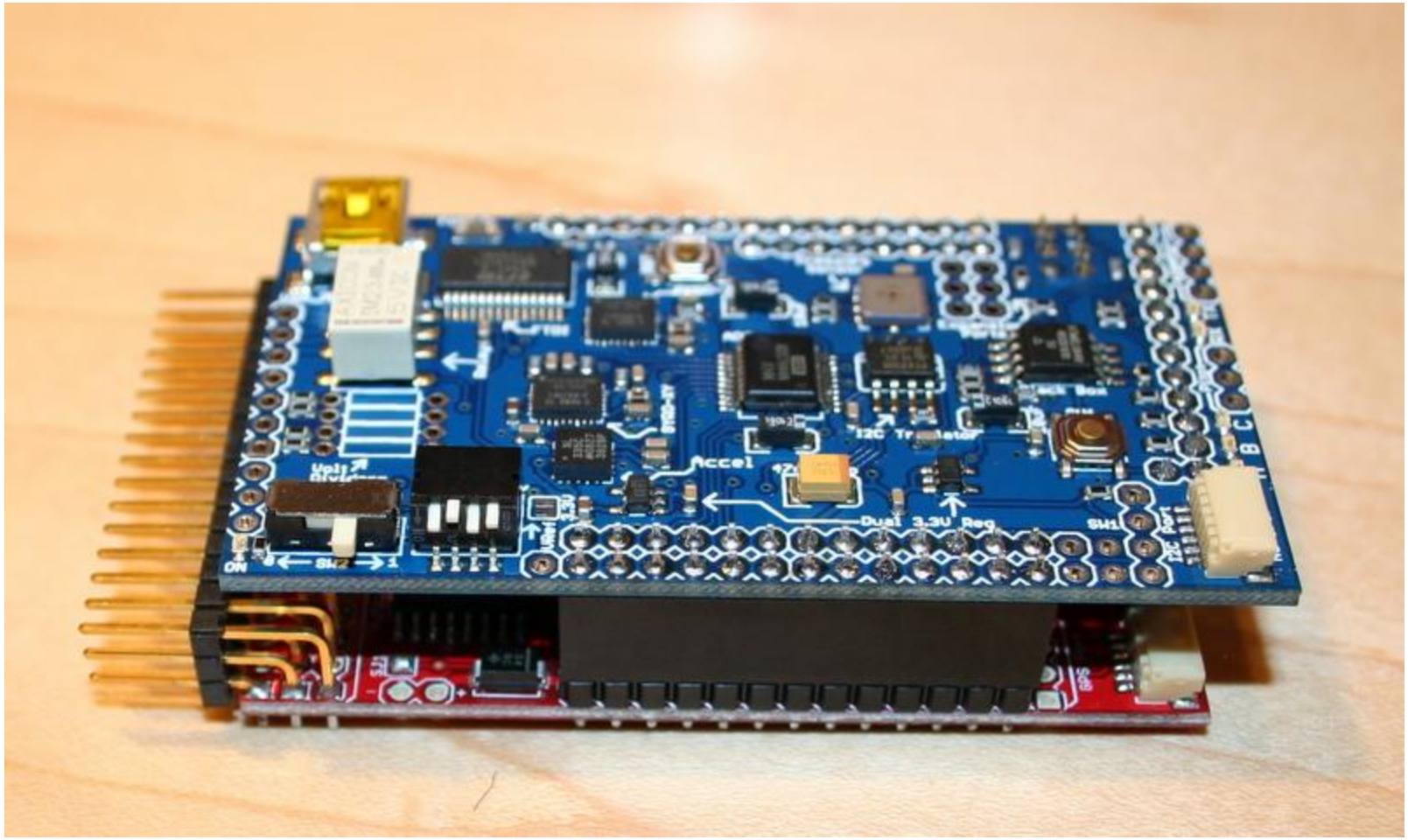
Now carefully place the shield on top, ensuring that all the female header pins line up with the right holes, as shown here:



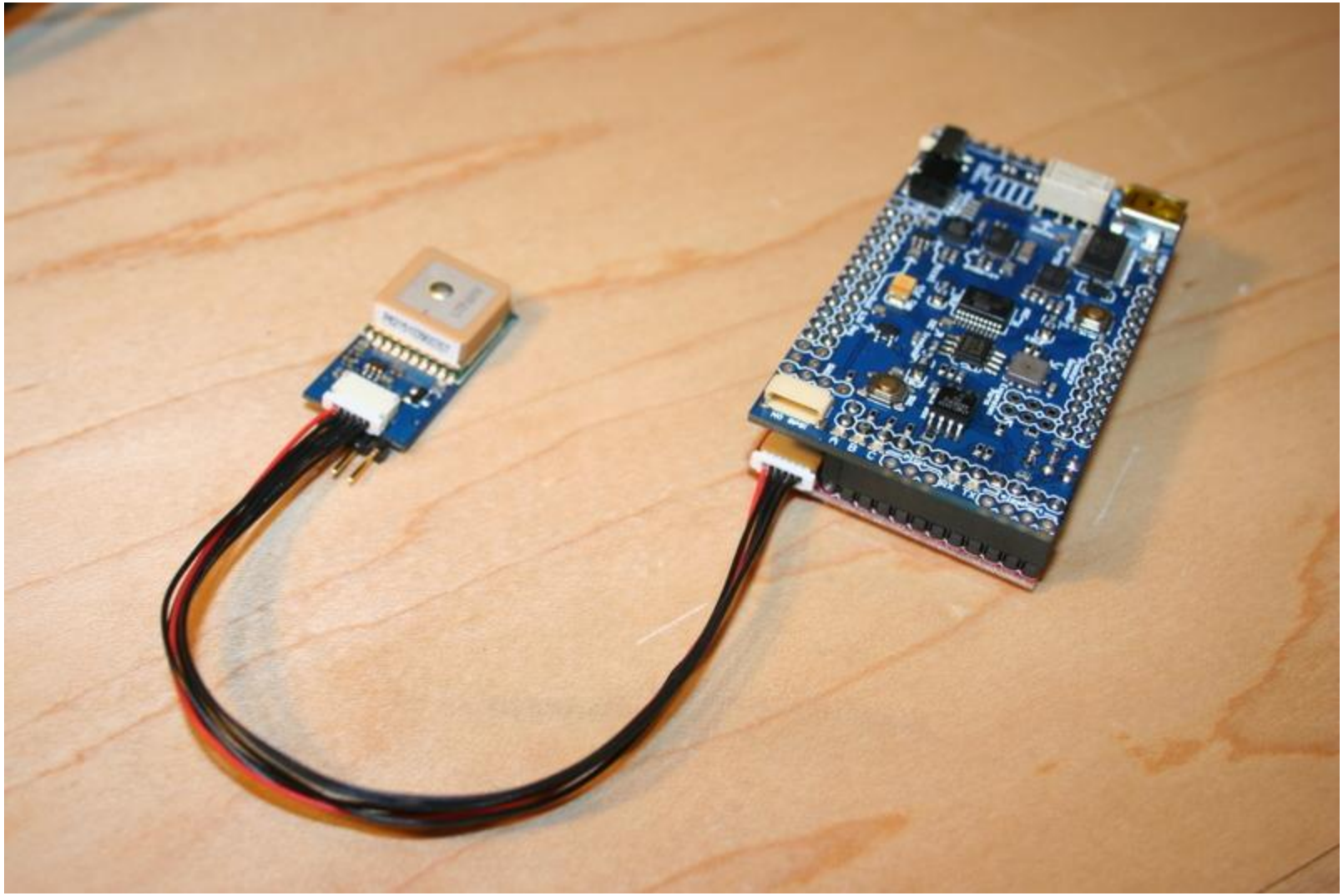
Solder a few pins on each row to hold everything in place, then gently remove the shield. Once it's off, finish soldering all the remaining pins. When you're finished and turn the shield over, it should look like this:



Now you can gently put the shield back on the ArduPilot Mega board. The paired boards should look like this:



Finally, attach your GPS module as shown below (MediaTek module shown). It goes in the connector on the APM board, **not** the similar one on the IMU shield (that one, which says "No GPS!", is an I2C connector for the optional magnetometer or other I2C sensor).



You're done with connecting your APM and IMU!

Quad frame

Kit status: Proto stage, initial frames under test

#1 - Main Frame & Carrier

ITEM	QTY	NOTES
Main Frame Plate	2	Center plates of whole quad
Carrier Board	2	Electronics carrier plates


DRONES
ArduCopter
Main Frame
Carrier Plate
#1
ArduCopter
DIYDrones

#2 - Aluminum arms

ITEM	QTY	NOTES
Aluminum Arm (21.5cm or 28cm)	4	Motor arms

21.5 cm:

28 cm:

#3 - Power Distribution

ITEM	QTY	NOTES
Power Distribution PCB	1	Board the connect everything together and carries the power
Battery Cable	1	For connecting your battery to the P-PCB
40 pin right angle header	1	Cut into groups of 3 pins to connect ESC signal cable to P-PCB

24 AWG Black Jumper Cable	1	Connecting 2 ESC grounds and/or I2C jumpering
18 AWG Black Jumper Cable	2	Bridging the negative traces of the P-PCB
24 AWG Connector Cable	6	APM connection, ESC signals and power
100 mm Ziptie	2	
M3 Nylon Nut	4	Mounting Nut
6 mm M3 Nylon Screw	4	Mounting Screw
12 mm Male-Female M3 Nylon Spacer	4	Hexagonal spacer
1 Row 4 Pin Connector Casing	1	For connection cables between APM and P-PCB
1 Row 2 Pin Connector Casing	1	For connection cables between APM and P-PCB
Shrink Tubes	4	


DRONES ArduCopter
Power Distribution
PCB
#3
Rev.B
DIYDrones
ArduCopter

#4 - Battery Mount

ITEM	QTY	NOTES
Battery Plate	1	Plate that mounts under main frame
10 mm Female-Female M3 Nylon Spacer	4	Hexagonal spacer allows distance for velcro strap
6 mm M3 Nylon Screw	4	Mounting screws, nylon


DIY DRONES ArduCopter
Battery Mount
#4

#5 - Landing Gear & Protective Dome

ITEM	QTY	NOTES
FIN	4	Landing gear fin
ARCH	4	Arches for dome
20 mm M3 Plastic Screws	12	20mm long Poly/Plastic screws, clear
M3 Plastic Nut	12	M3 Nut, Poly/Plastic, clear


Arducopter
DRONES
Langing Gear
Protection Dome
#5

#6 - Dome Center

ITEM	QTY	NOTES
Center Upper	1	Dome Center, upper plate
Center Lower	1	Dome Center, lower plate
20 mm M3 Plastic Screw	2	20mm long Poly/Plastic screw, clear
M3 Plastic Nut	4	M3 Nut, Poly/Plastic, clear


DRONES ArduCopter
Dome Center
#6

#7 - Motor Mount (A), Upper

ITEM	QTY	NOTES
Motor Mount	4	Motor Mount upper plate, thickness 5mm


DRONES ArduCopter
MotorMount (A)
#7

#8 - Motor Mount (B), Lower

ITEM	QTY	NOTES
Motor Mount	4	Motor Mount lower plate, thickness 3mm
25 mm M3 Plastic Screw	8	25mm long Poly/Plastic screw, clear
M3 Plastic Nut	8	M3 Nut, Poly/Plastic, clear
25 mm M3 Steel Screw	4	M3 x 25mm steel screw
M3 Steel Washer	4	M3 Washer, steel
M3 Nylon Washer	4	M3 Washer, nylon


DIY DRONES ArduCopter
MotorMount (B)
#8

#9 - General Pack

ITEM	QTY	NOTES
Battery Strap	1	Securing battery to mount
Velcro Sticker	1	Securing receiver
100 mm Ziptie	10	
25 mm Male-Female M3 Nylon Spacer	8	
20 mm M3 Nylon Screw	8	
M3 Nylon Nut	18	
6 mm M3 Nylon Screw	4	
10 mm M3 Nylon Screw	3	
M3 Nylon Washer	3	
20 mm M3 Steel Screw	4	
M3 Steel Nut	4	


DRONES ArduCopter
General Pack
#9

Hexa frame

Kit status: Proto stage, initial frames under test

ITEM	QTY	NOTES
Round Hexa/Quad plate	x 2	Arms can be mounted as + or x and in hexa or quad formation
Square Carrier plate	x 2	
Aluminum arms	x 6	

To be added more, spacers, screws, nuts, washers

Power distribution PCB

For Quad setup

Part status: On proto stage, testing

ITEM	QTY	NOTES
Power Distribution PCB	x 1	
3 Pin right angle connector	x 4	

For Hexa setup

Part status: On design stage, protos planned at end of June

ITEM	QTY	NOTES

Quad_Soldering_PDistro

Soldering the Power Distribution Board

For soldering Power Distribution board you need to have some basic soldering tools.

- Soldering iron, normal 30W iron is ok
- Soldering wire (Some documentation talks about paste but wire is best)
- Cutting pliers for cutting wires and peeling cover

On Bag #3 you should find following items:

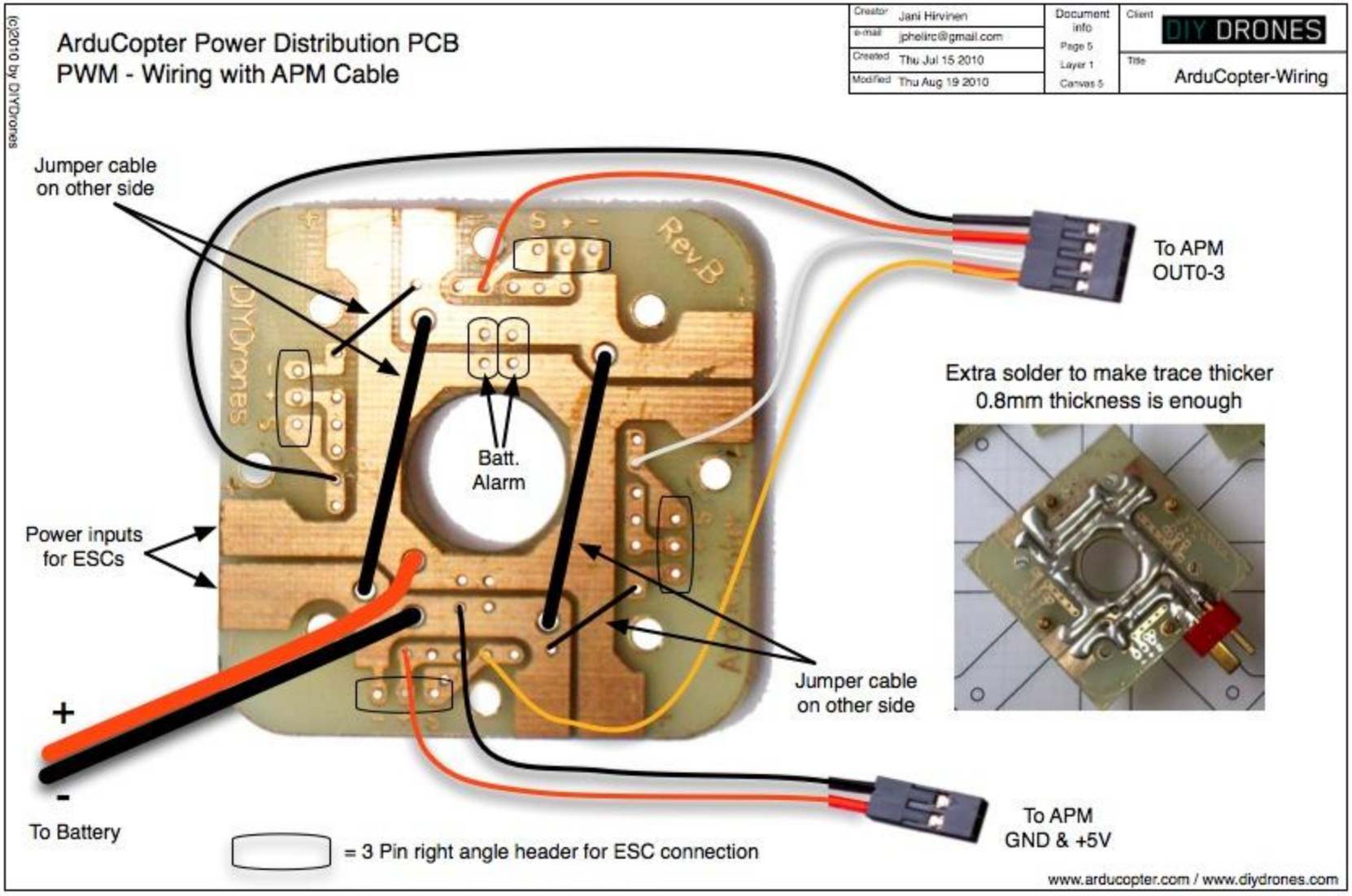
- 6 x 24awg wire for P-PCB - APM connection, ESC signals and power
- 1 x 2 pin flat connector, power connector to APM from one of your ESCs
- 1 x 4 pin flat connector, ESC signals from APM (OUT0-3)
- 1 x 40 pin right angle header
- 1 x 15cm 14AWG battery cable, fine copper
- 1 x 20cm 20AWG jumper cable, connecting 2 ESC grounds and/or I2C jumpering
- 2 x 4cm 18AWG jumper cable
- 1 x Power distribution PCB (Rev.B)
- 4 x M3x12mm Nylon Hexagonal spacer, Male-Female
- 4 x M3 Nylon nut
- Shrink tubes

This instruction is only for Rev.B model PCB. which is still beta PCB and having traces on only one side.

Retail packets will have Rev.C board and it will have plated trough holes for easier soldering from both sides.

Power Distribution wiring

Note: all wires should come up front the bottom (the side without the traces). Solder all the wires first, then lay down a layer of solder on the trace side.

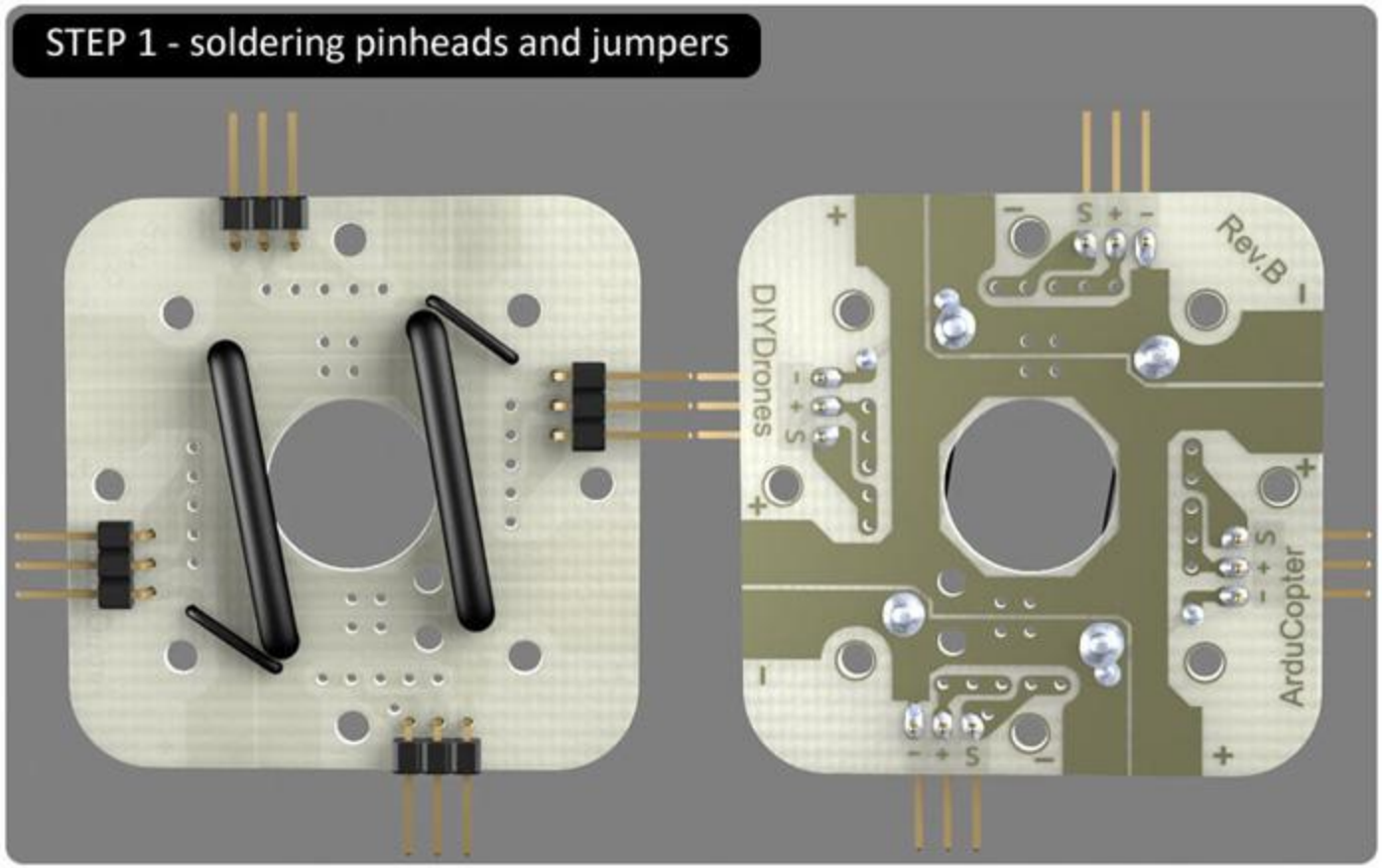


Step by Step

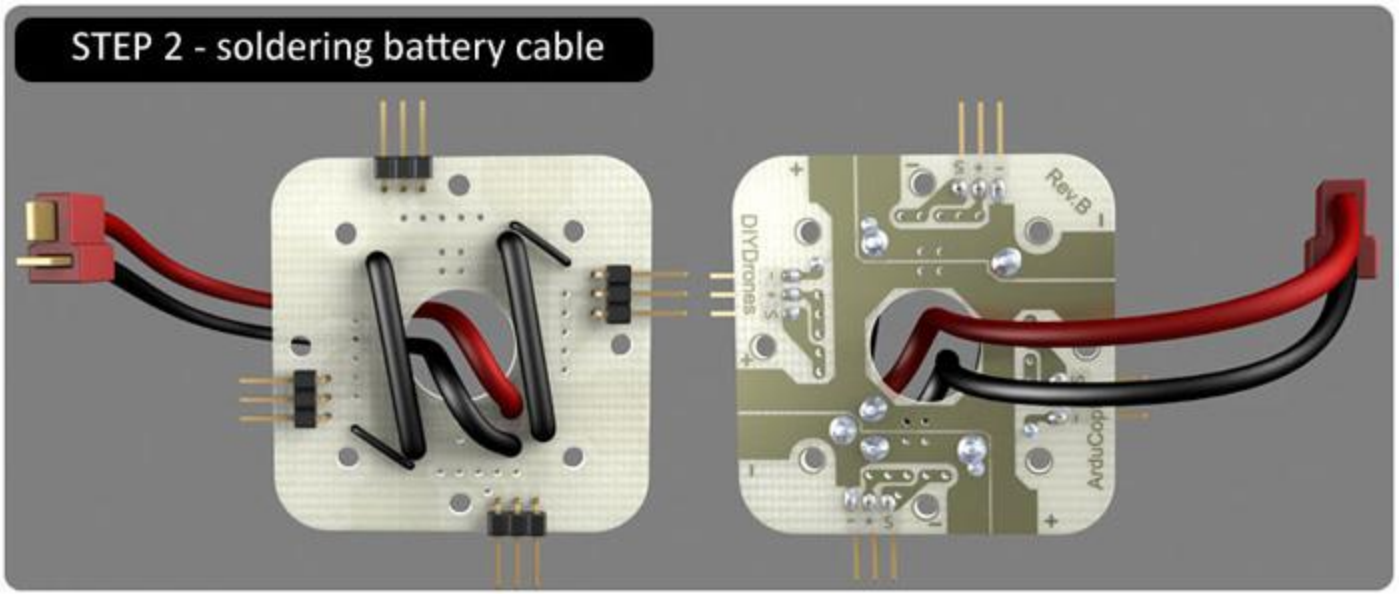
In Bag #3, you will find a 40 pin right angle header. This is used to connect the signal cables from each ESC to the Power Distribution PCB. Each signal cable needs 3 pins, so cut the 40 pin right angle header into sections of 3 pins. Look at the Power Distribution PCB and notice the three small holes labeled *(-), (+), and (S)*. Now, take your 3 pin right angle header that you just made and put the shorter pins through those holes from the bottom of the board (the side without the

traces and the writing). Then, solder each pin from the top of the board being careful not to short together any of the pins. Do this for all four of the 3 pin right angle headers.

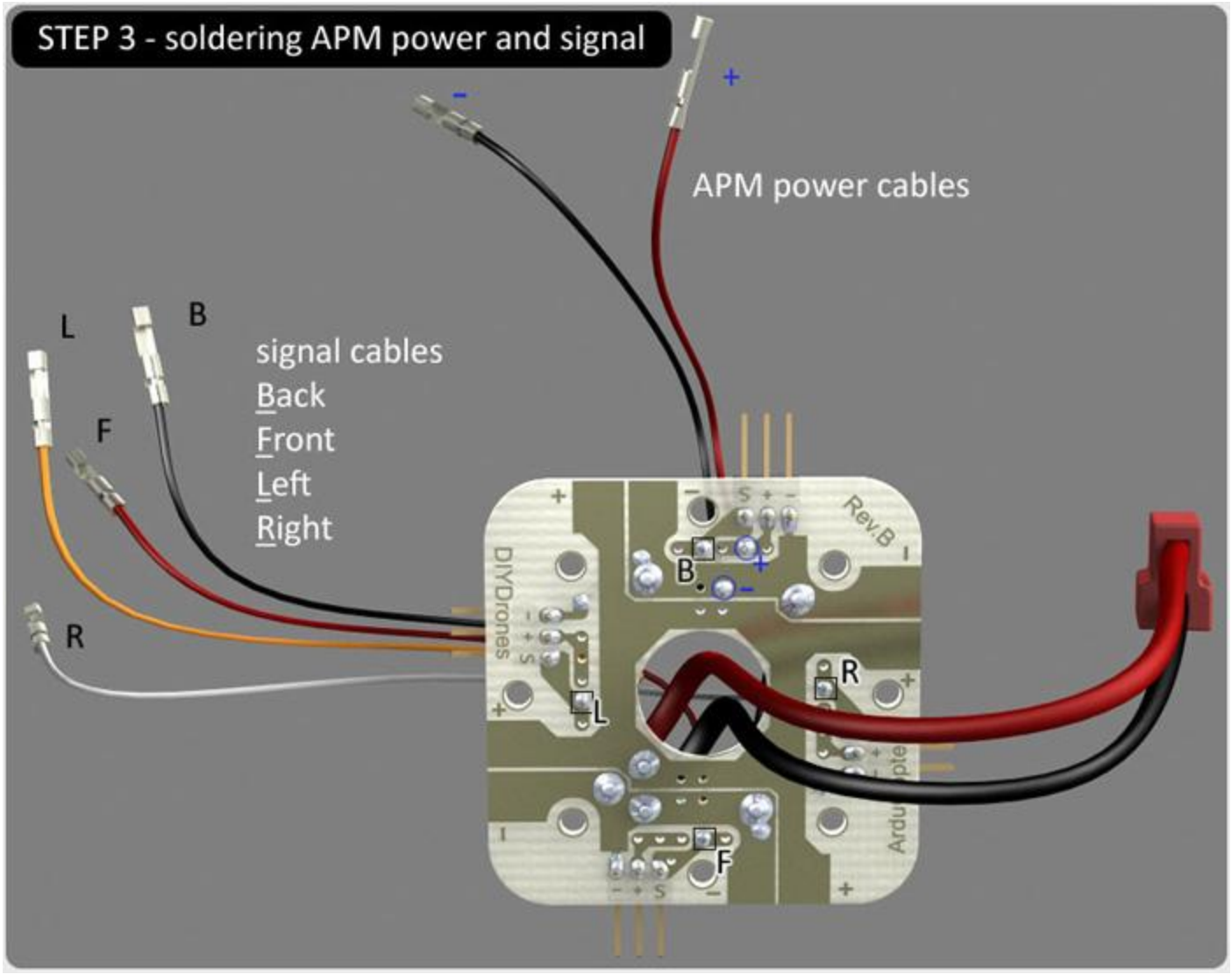
You will also find 2 - 18 AWG black jumper cable in Bag #3. These go into two holes on the negative traces of the board. Jumper these negative traces by soldering the cables on the back of the board. The same concept applies for the 24 AWG black jumper cable. Like shown in the picture below, cut the supplied cable into 2 pieces that allow for a connection from the negative pin of the ESC connectors to the negative power trace and solder them.



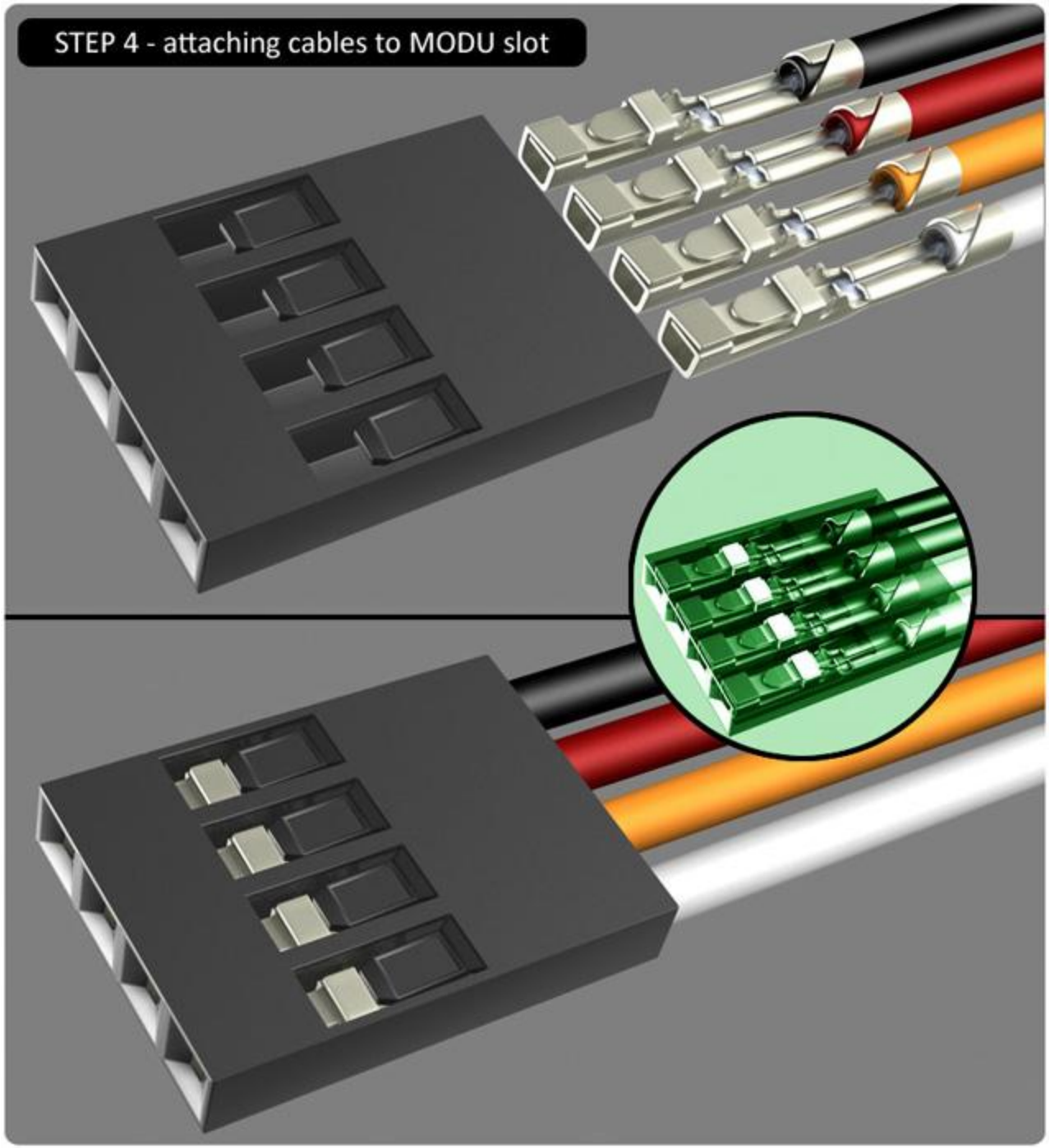
A battery cable is supplied. In the picture, a Deans connector is soldered to one end of the cable. Depending on your battery, you may choose to solder a different connectors such as a XT60 or EC5. Slide the battery cable through the bottom of the board into the positive and negative holes for the battery and solder.



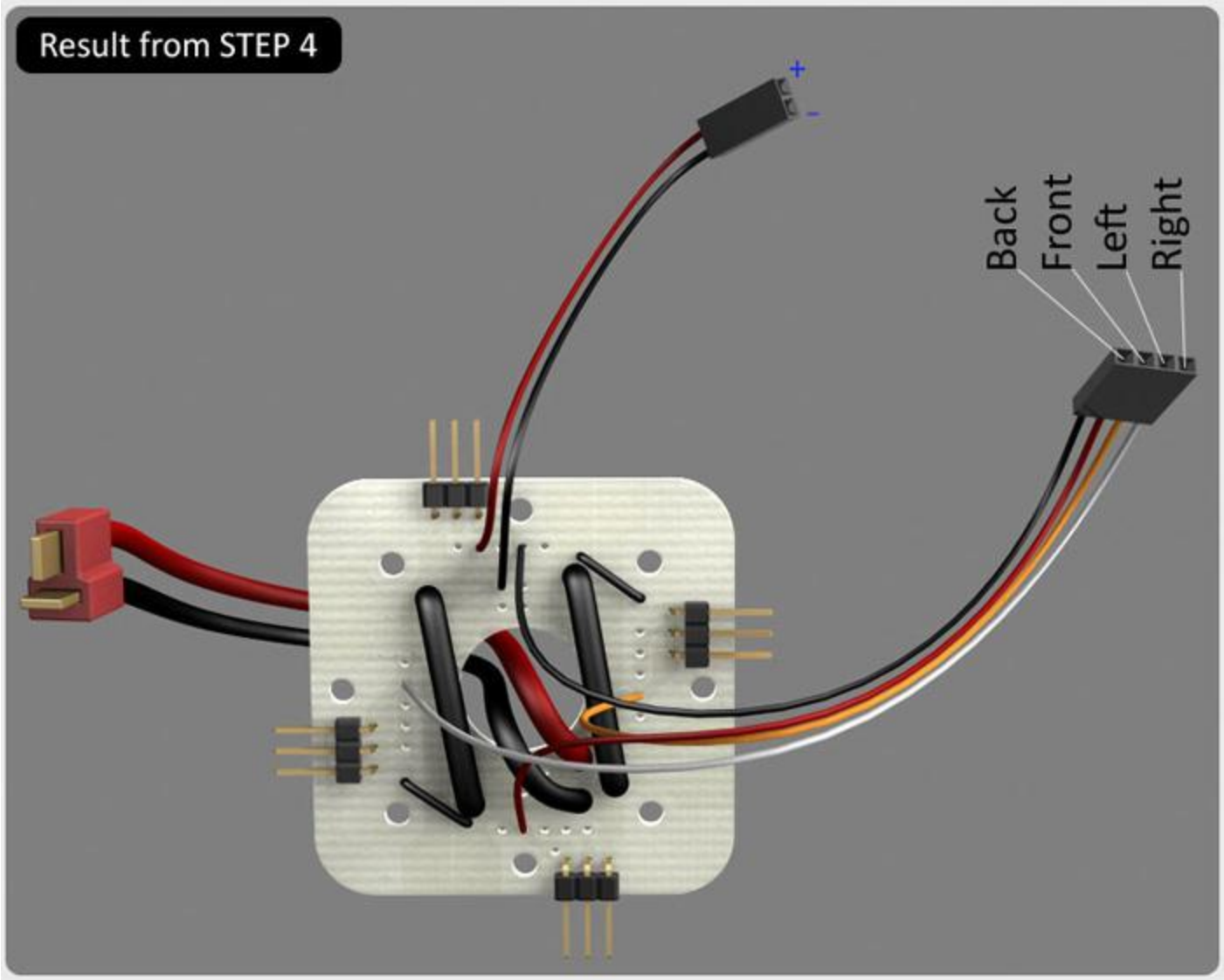
Take out the 6 - 25 AWG Connector Cables. There should be two black, two red, one yellow, and one white. Four of these are used to supply the signals to the ESCs and two will be used to supply power to the APM. Look at the diagram below and solder the corresponding connector cables to the board.



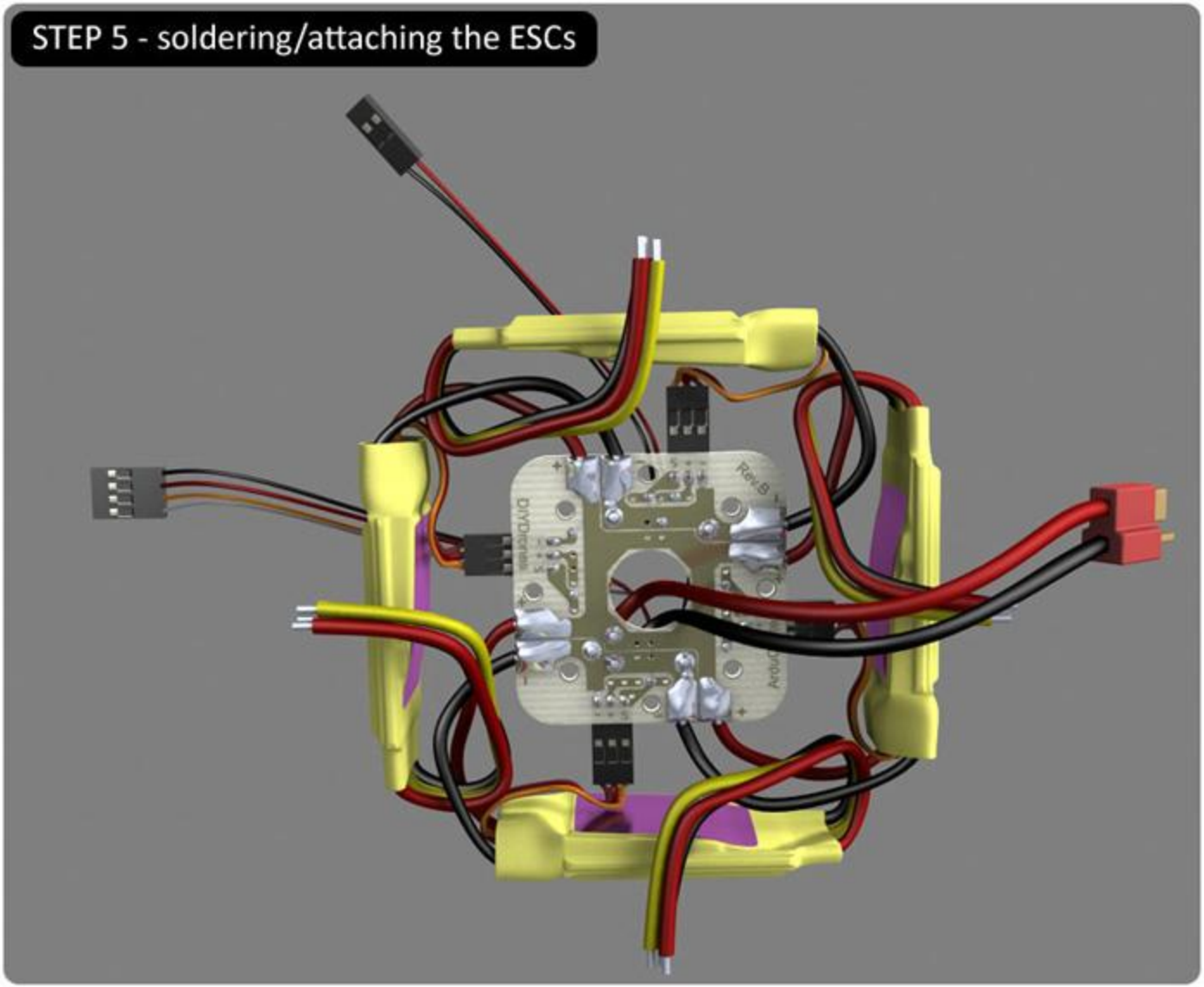
Now, take out the 4 pin connector casing. The four connector cables for the ESC signals that you just soldered will snap into this casing. Take note of the correct color order.


STEP 4 - attaching cables to MODU slot

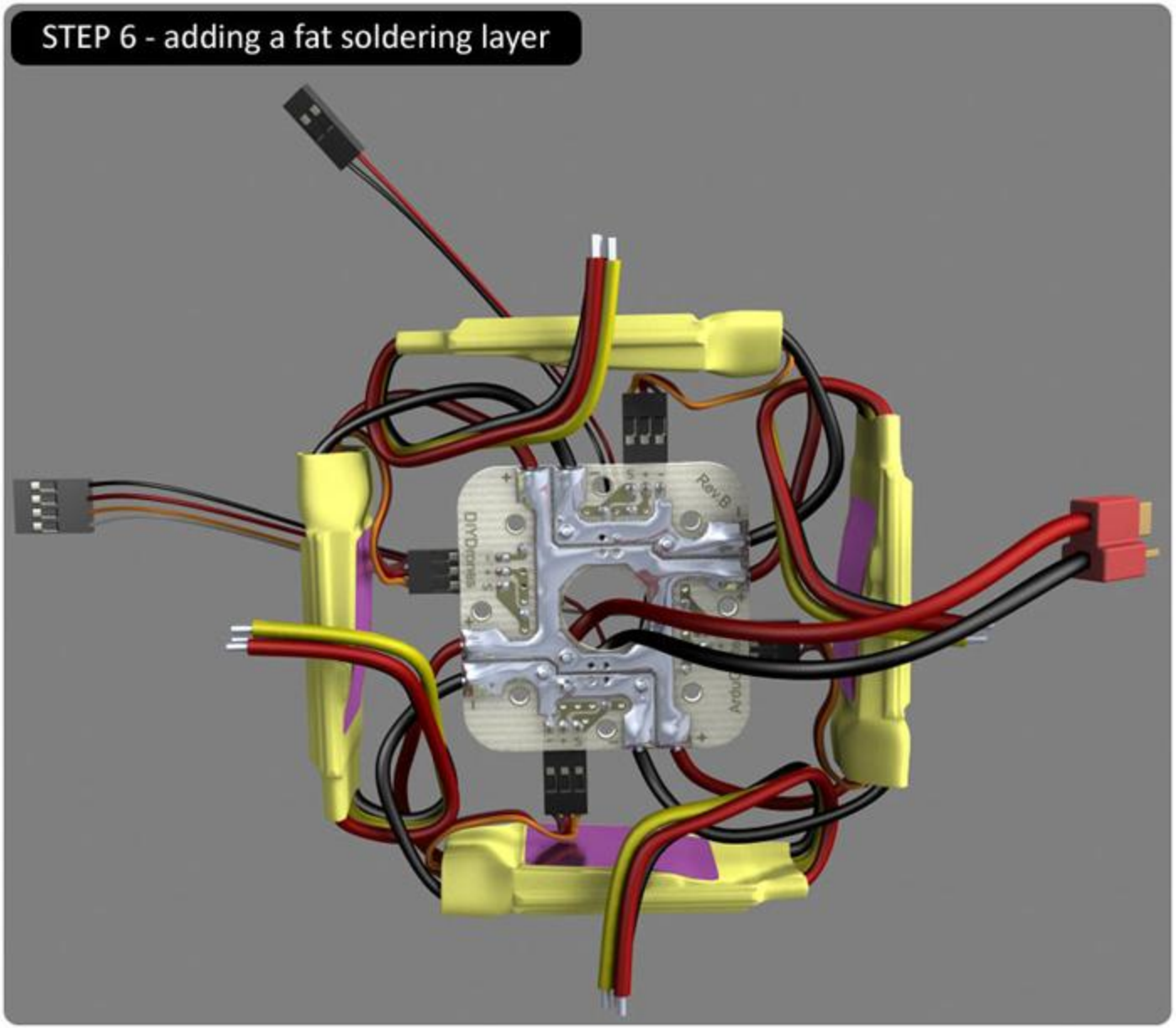
Now, do the same thing with the other two connector cables and the 2 pin connector casing. Your P-PCB should now look like this:



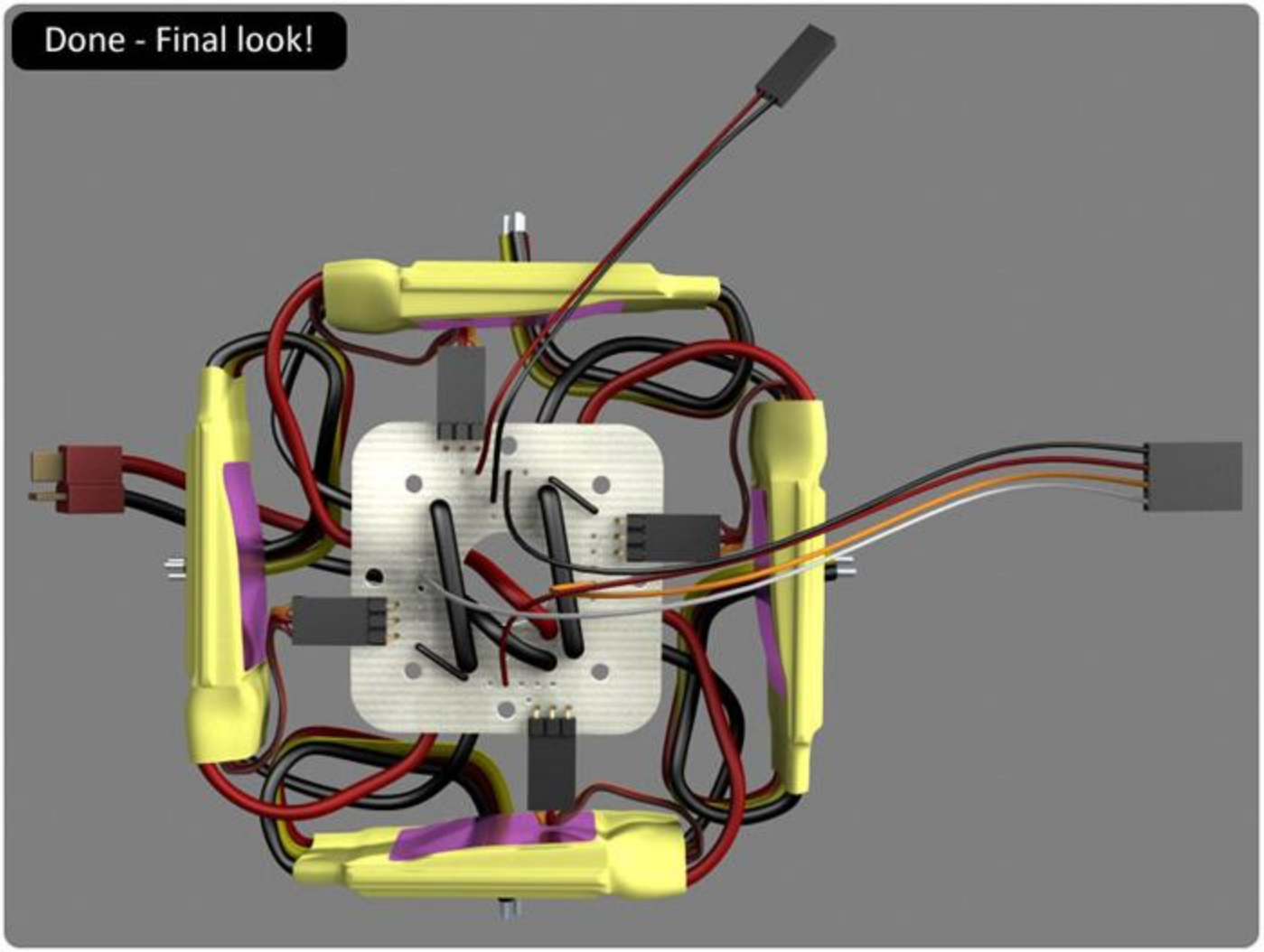
For all four of the ESCs, strip the end of the positive and negative wires so that they can be soldered to the P-PCB. Take note of the labeling on the P-PCB and remember that red wires should be positive and black wires should be negative. Securely attach them with a layer of solder. The other connectors on your ESCs can be attached to the 3 pin right angle headers now. When connecting these, take note of the signal pin. Signal is usually white or yellow.



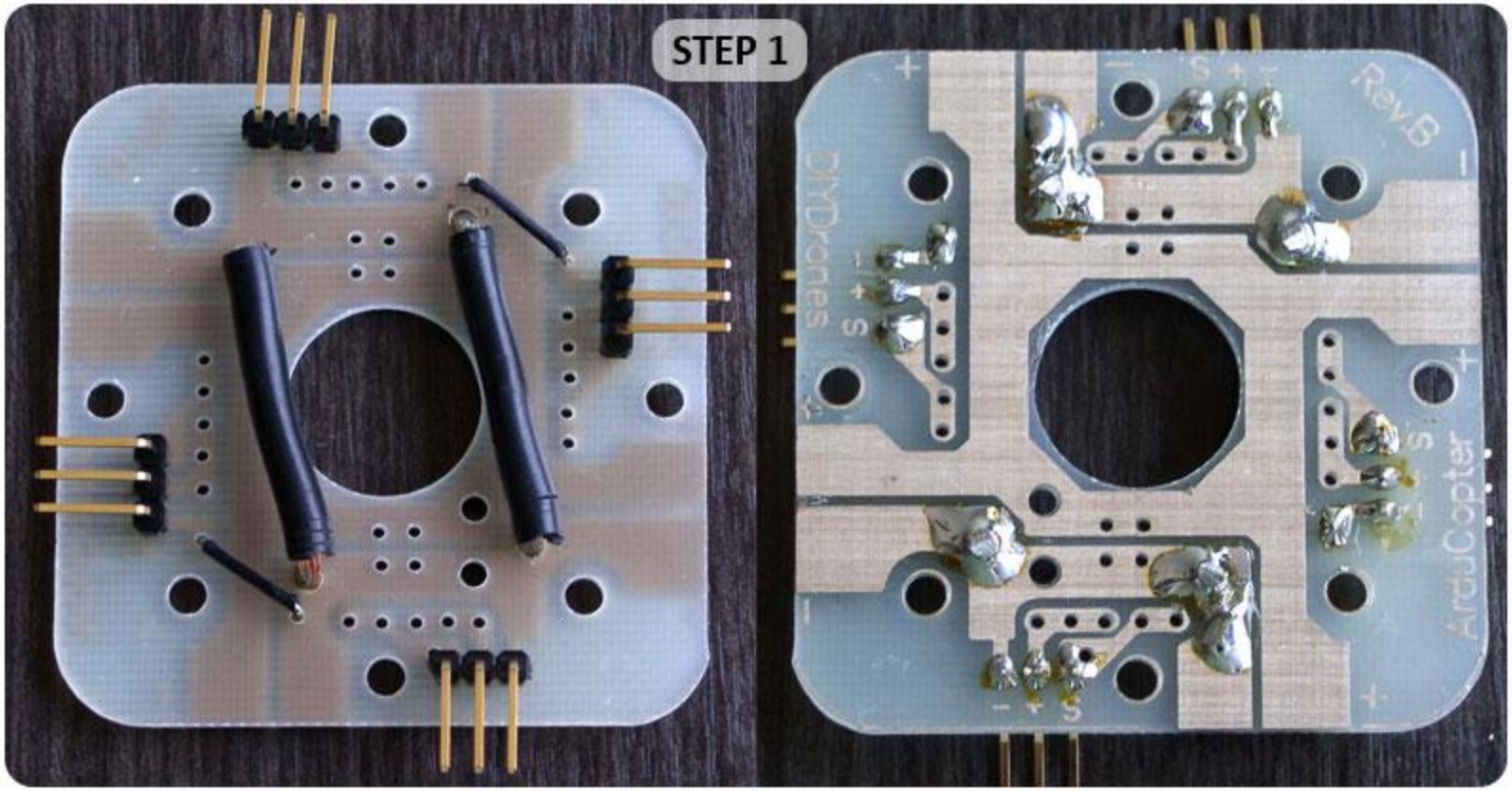
Now that all of the components are soldered on to the P-PCB, an extra layer of solder needs to be added to the traces in order to accommodate the current. A thickness of about 0.8 mm will be enough.

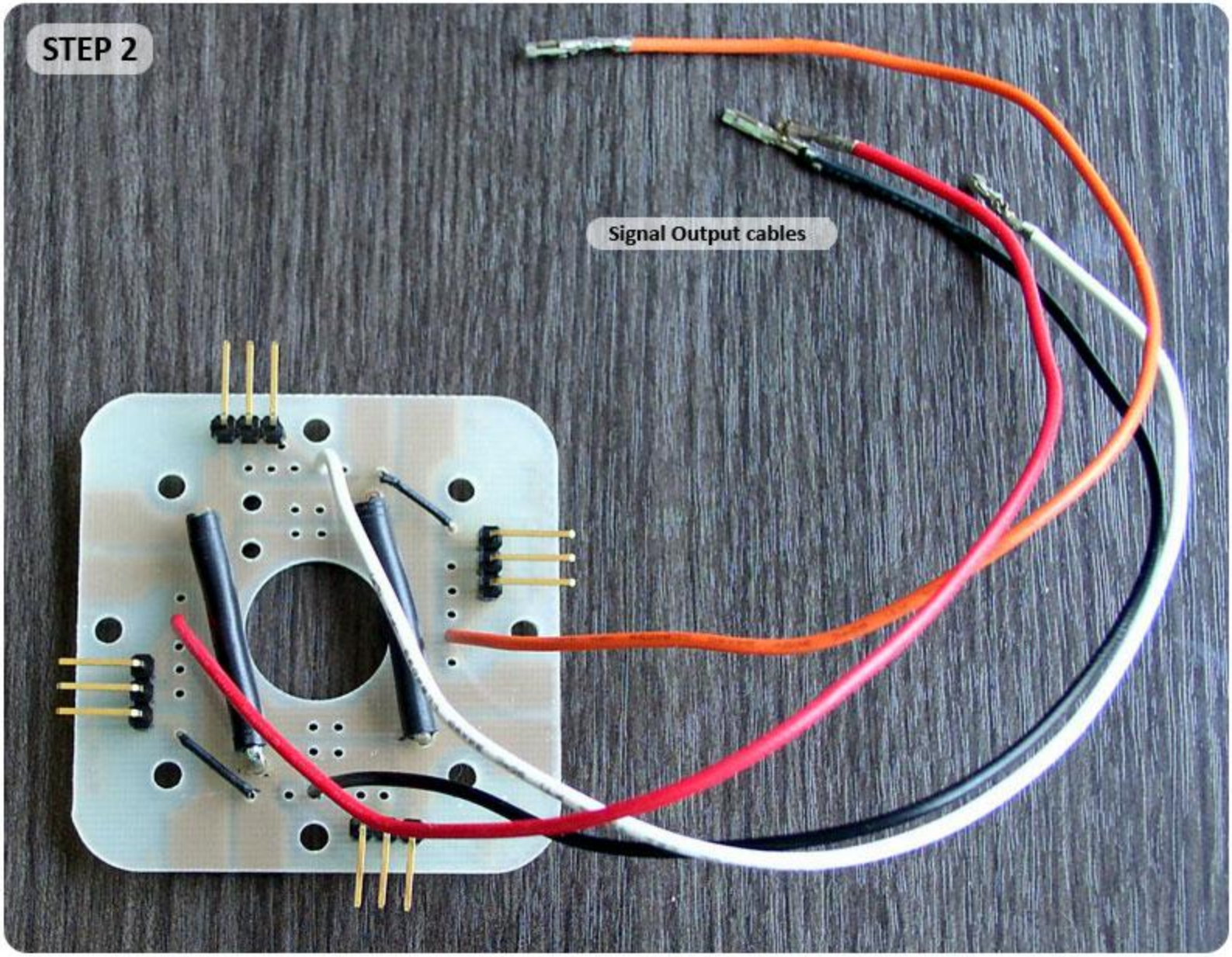


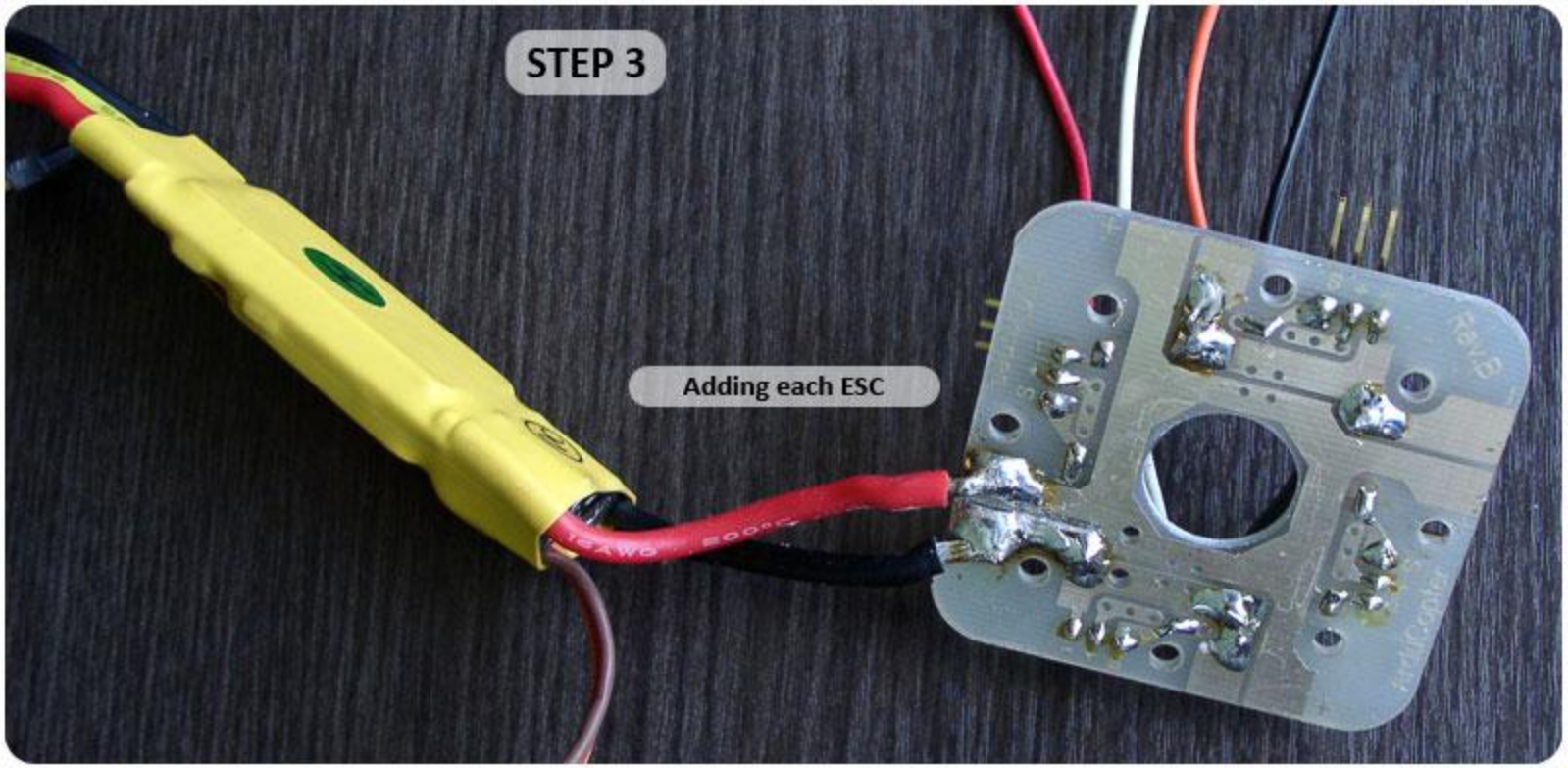
The P-PCB is all done! It should look like this:

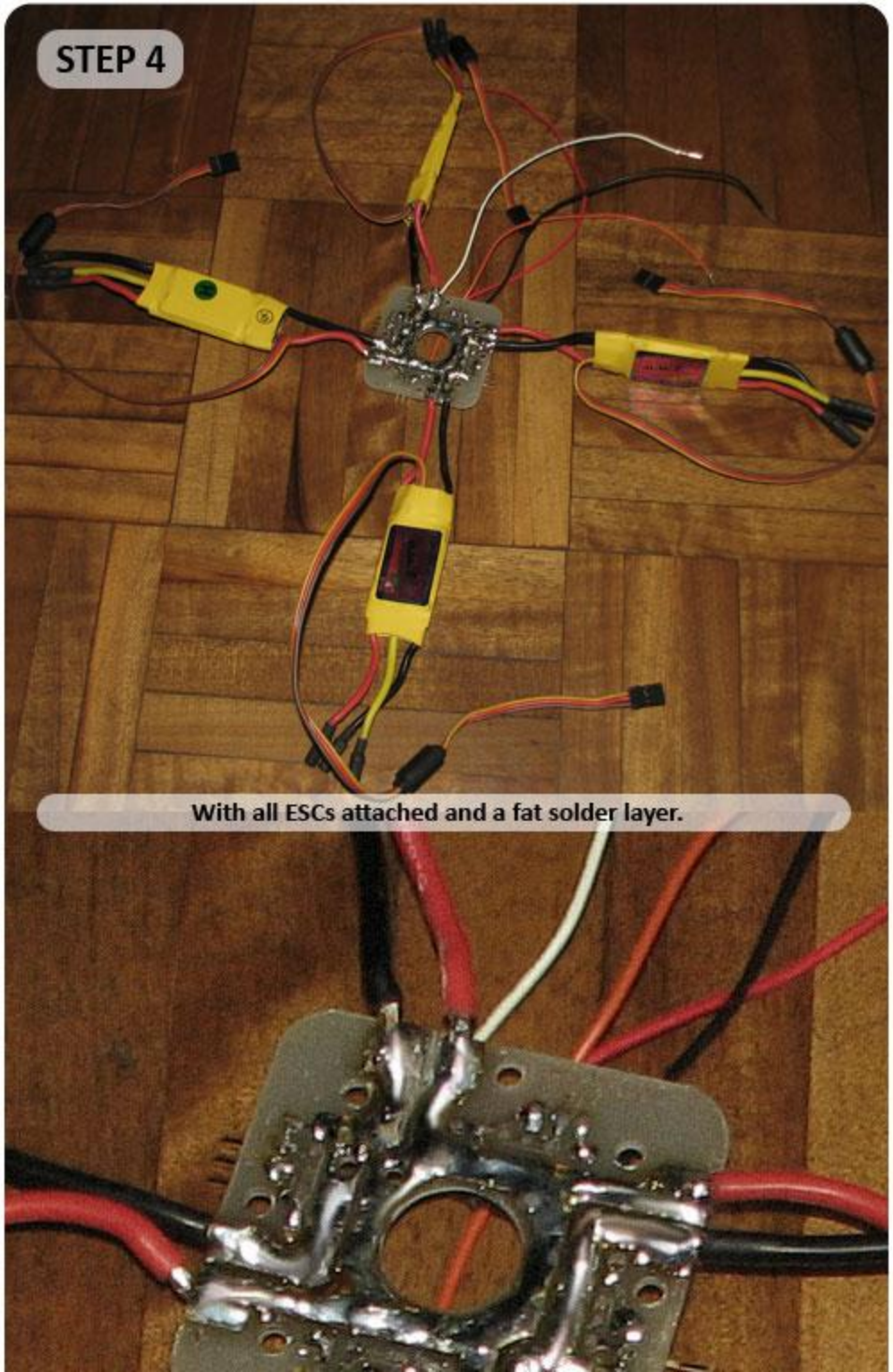


Additional Images:


STEP 1
RevB
DIYDrones
ArduCopter


STEP 2
Signal Output cables


STEP 3
Adding each ESC

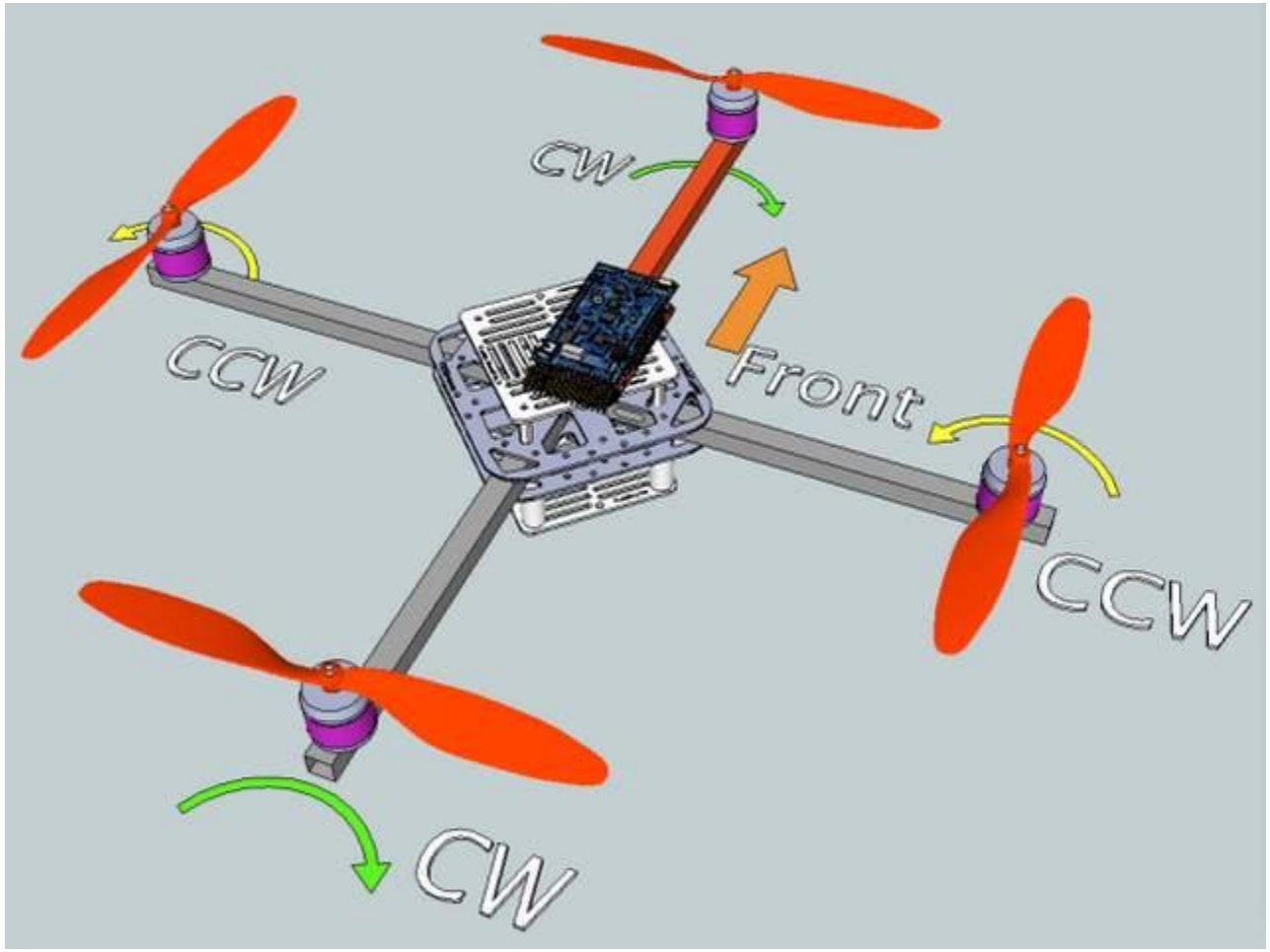

STEP 4
With all ESCs attached and a fat solder layer.

Quad_Frame_Assembly

Setting up your airframe

This is quick assembly instructions on how to assemble ArduCopter Frame 1.0. We are constantly updating these pages so come back often to see if there is any new information.

From these pages you will find instruction guides and also assembly videos.



You'll need two kinds of props (puller and pusher). Pusher props go clockwise; regular props go counterclockwise. If you get the full kit with motors (not yet available), these are included.

Content of Jani's "Magic Box" with all frame parts:

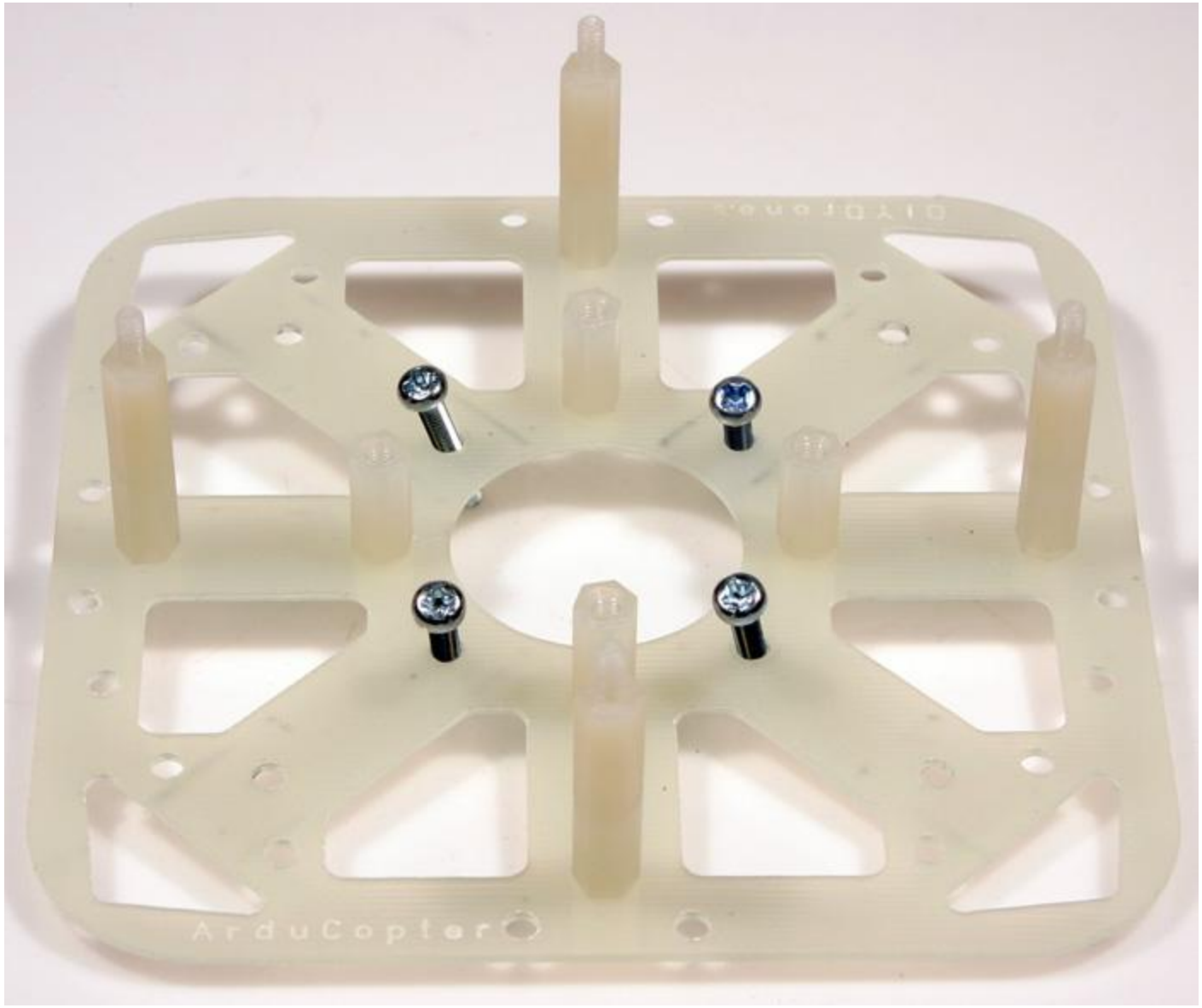
Below are instructions in video format followed by an instructional picture guide. In the videos, we assemble the basic frame with all spacers, aluminum arms, carrier plates, motor mounts, protective dome, landing fins, and battery mount.

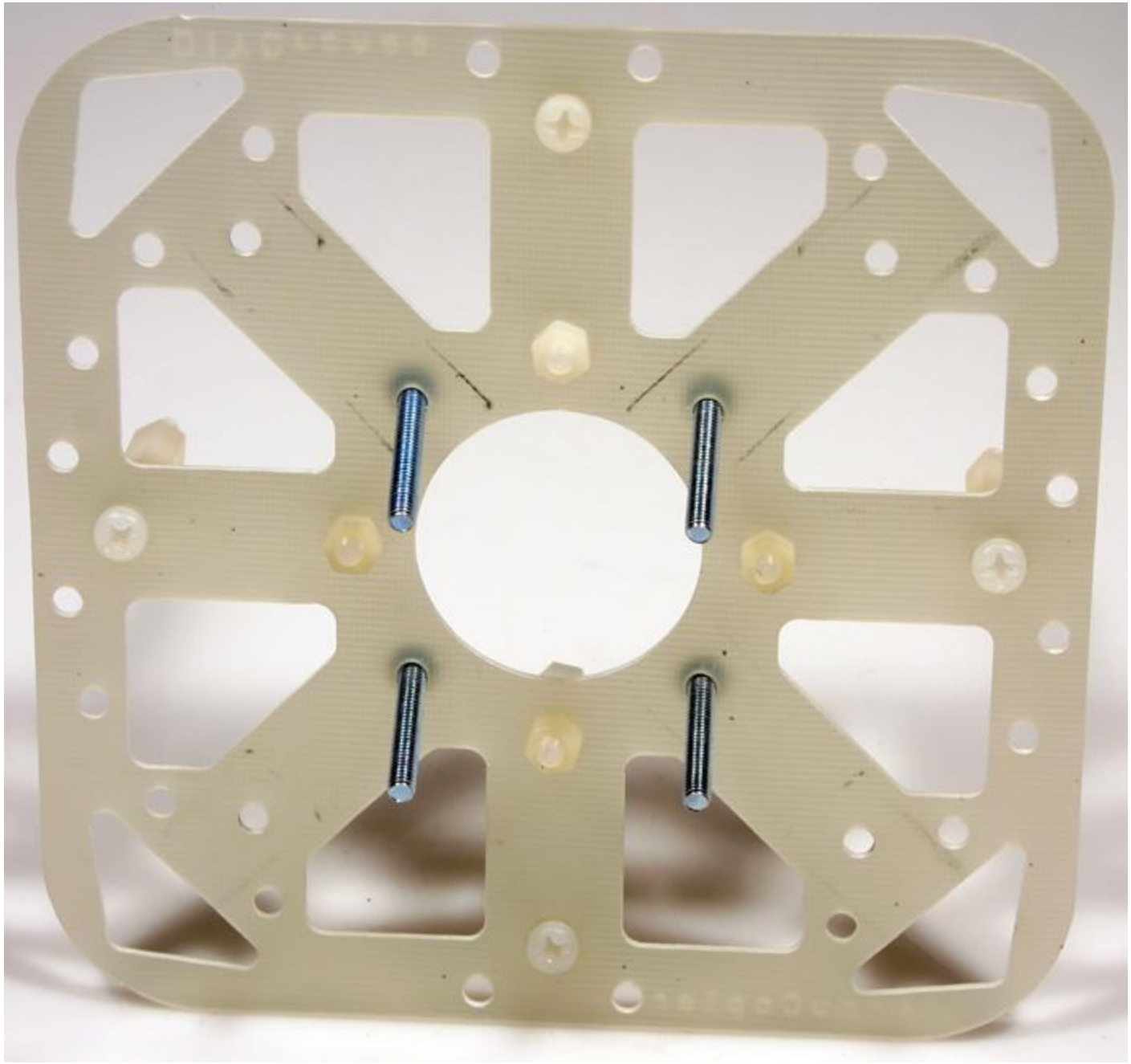
ArduCopter Frame 1.0 Guide

NOTE *This is the first write-up of assembling the beta frame. Methods are subject to change.* ***In the RC1 release of ArduCopter code, there is a code definition to switch between (+) and (x) mode. Therefore, assemble your frame for the (+) configuration instead of the (x) shown below.*** *This guide will be updated with the new frame and new orientation in the future.*

Open Bag #1 and take out 1 of the Main Frame boards. You'll also need 4 of the M3 x 25mm Male-Female Nylon Spacers and 4 of the M3 x 6mm Nylon screws that are in Bag #9. The main frame is supposed to support both '+' and 'X' configurations. Take each of the spacers and put them over the hole closest to the center of the three holes near the outside edge of the Main Frame board with the female side down. Secure each one with a screw from the bottom of the board. The spacers should NOT be installed over where the aluminum arms will be which depends on if you have a '+' or 'X' configuration. Now, take 4 - M3 x 20mm steel screws out of Bag #9 and place them through the innermost hole. These will end up going through your aluminum arms, so place them accordingly.

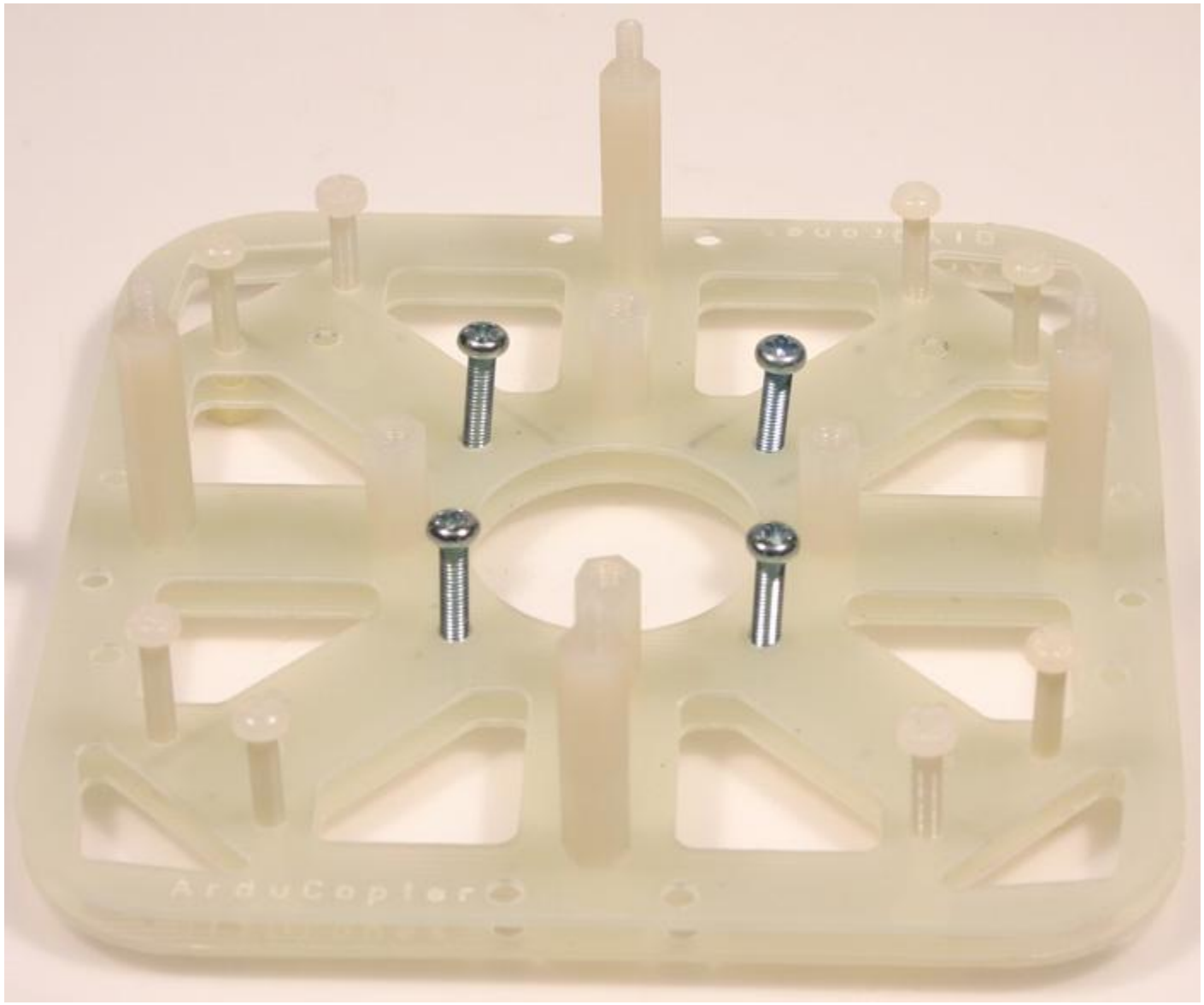
In anticipation of installing the Power Distribution Board and to make it easier for ourselves when we do, it is best to install the spacers right now. Open Bag #3 and take out the 4 - M3 x 12mm Male-Female Nylon spacers and the 4 - M3 Nylon Nuts. Place the spacers through the innermost holes with the female side up that are not being used by the steel screws. These spacers should line up with the longer spacers already installed. Secure them from the bottom side of the board with a nut.

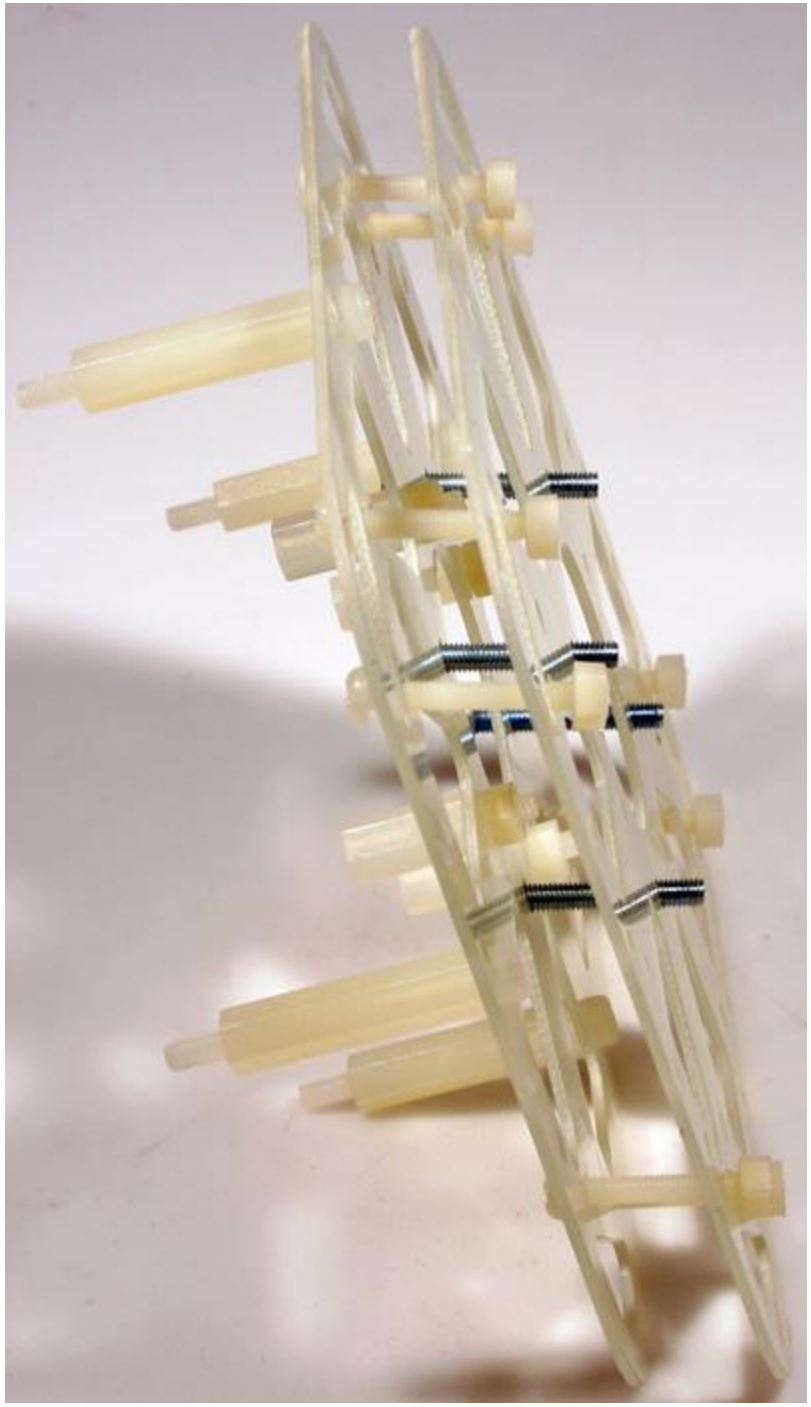

ArduCopter

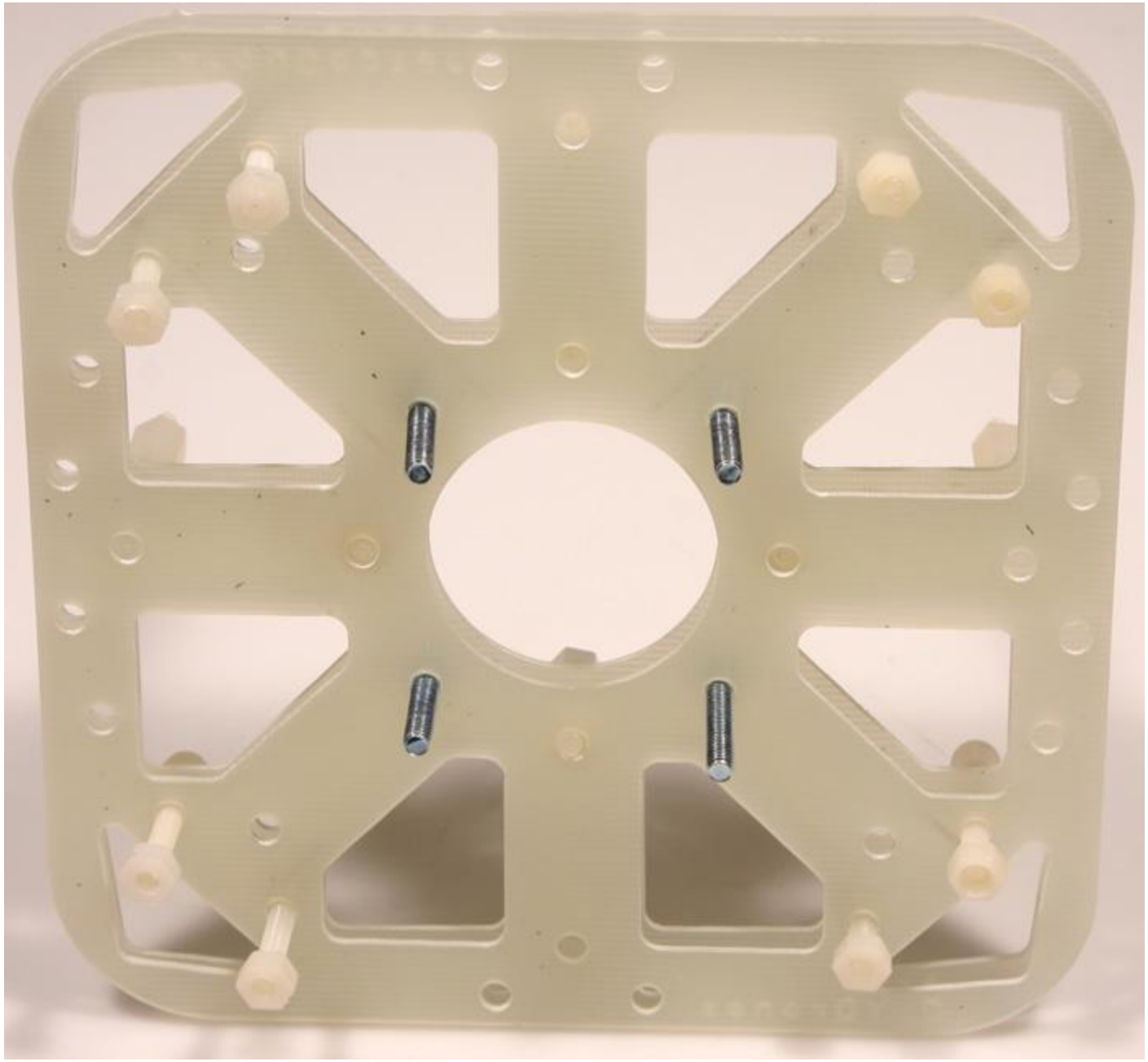


Now, remove the second Main Frame board and line it up with the first one. These two boards will sandwich the aluminum arms. Take out the 8 - M3 x 20mm Nylon Screws and 8 - M3 Nylon Nuts from Bag #9. Two of the nylon screws will be on either side of an aluminum arm and the steel screw will pass through the

arm. Place the nylon screws through both Main Frame boards in the outermost holes of where your arms will be. Place a nut just barely on the other end; we will tighten it later.

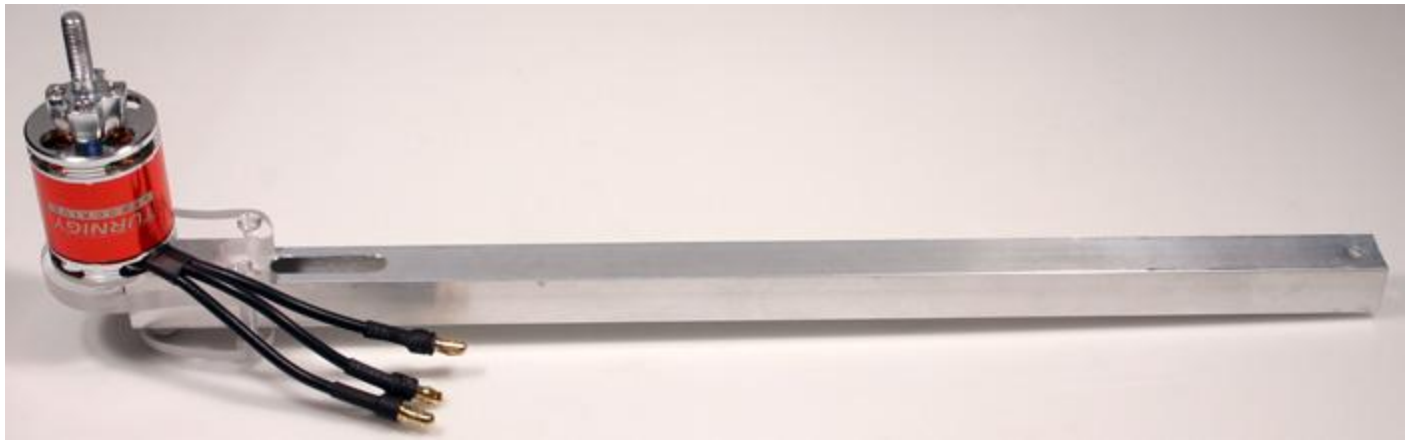


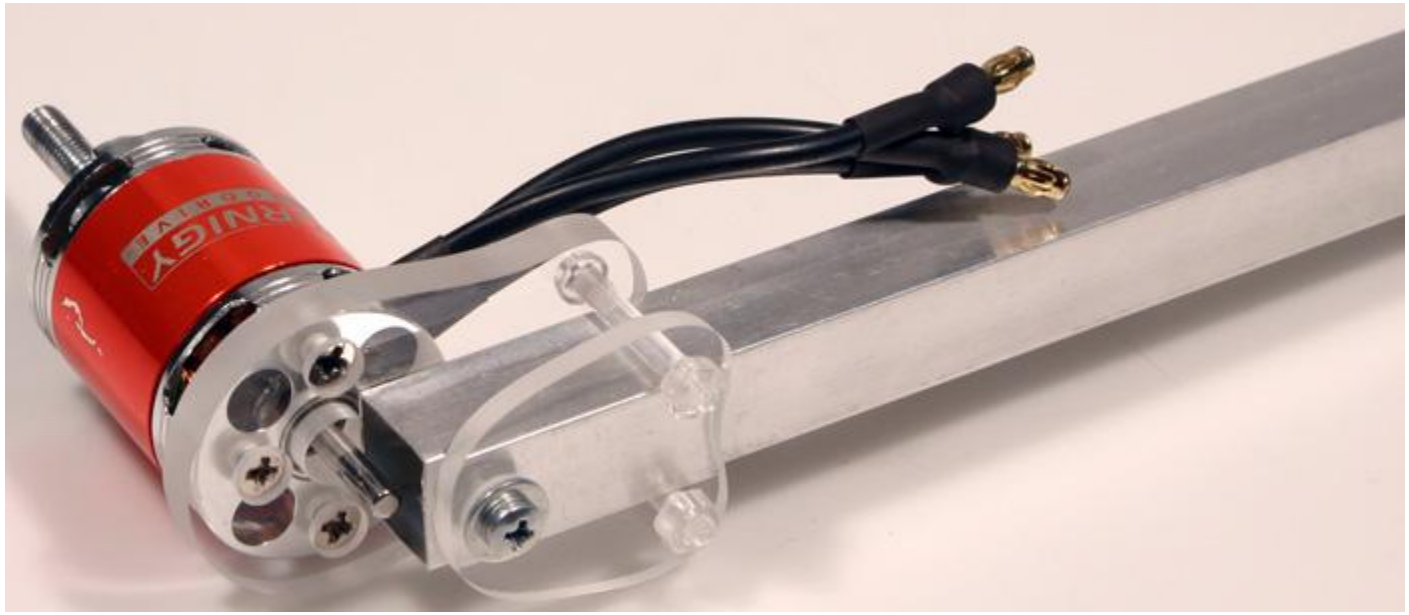




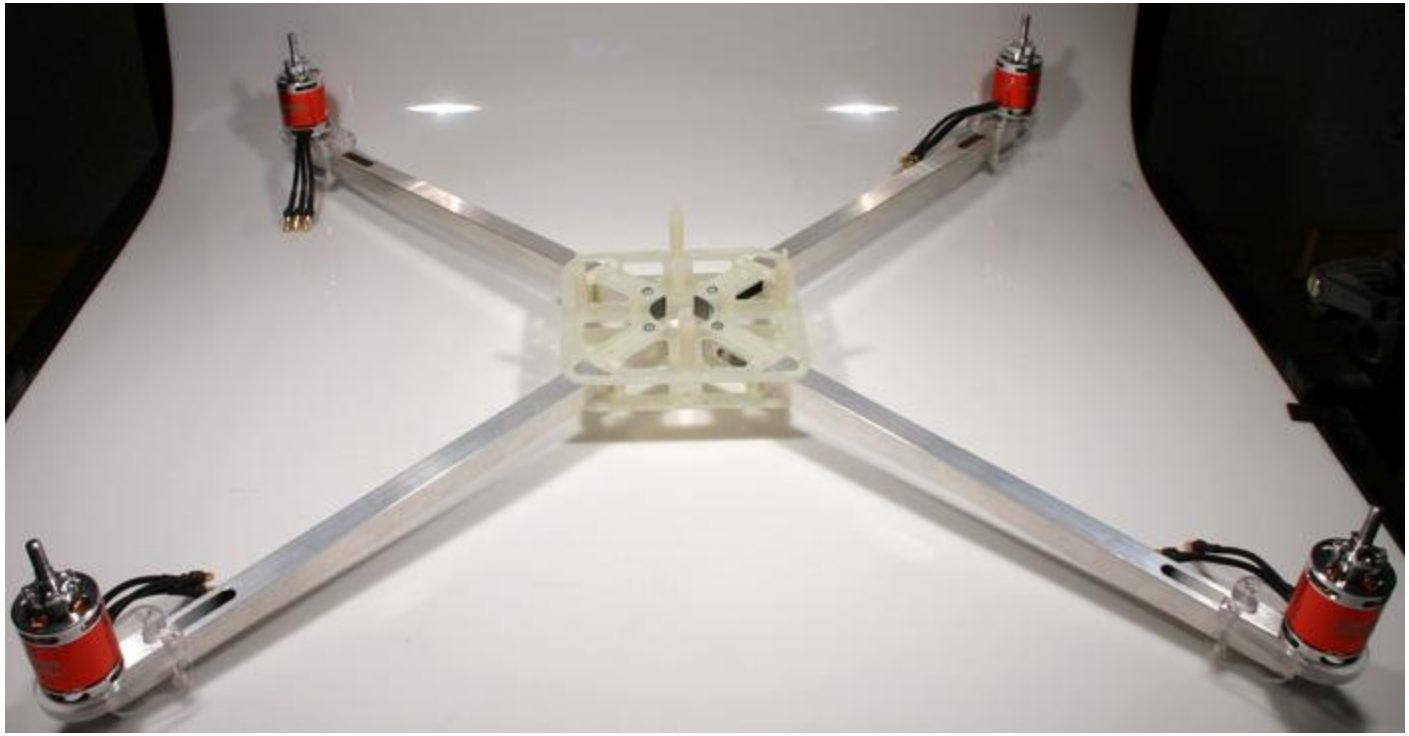
Now, choose the size of aluminum arm you want to use. If you look at it, you should an elongated hole on both ends. Find the side where the hole is furthest from the end of the arm. This is the side that should face up and the end that your motor will mount to. The motor mount parts are located in Bag #7 and #8. The upper part of the motor mount is the larger piece. Take a M3 x 25mm Steel screw and place a steel washer followed by a nylon washer on it. Push it through the lower

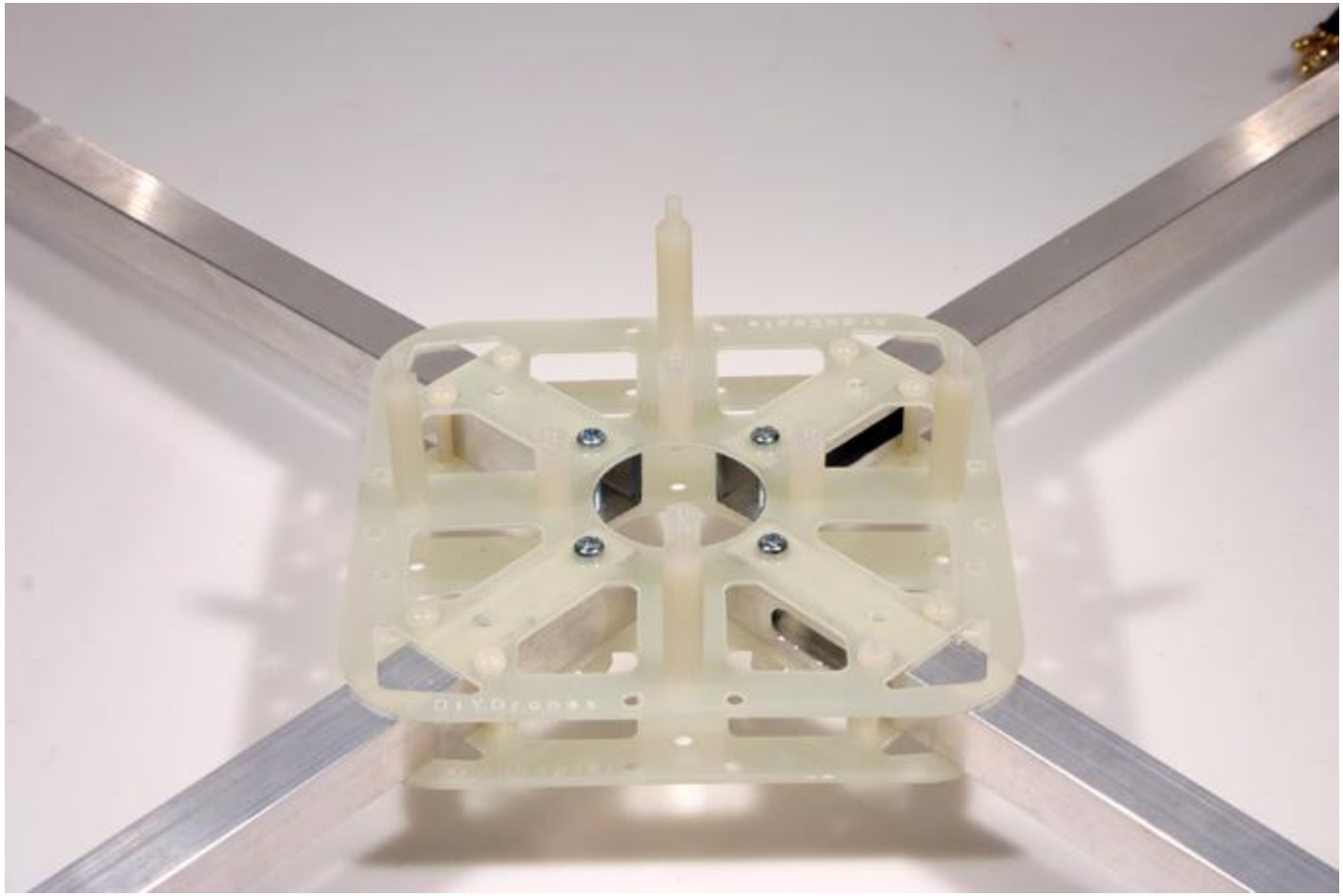
motor mount, the aluminum arm, and the upper motor mount. Take your 16/19 mm spacing motor and line it up with the holes on the end of the upper motor mount and the steel screw you just put it. Tighten the steel screw and use the mounting screws that came with your motor for the remaining three holes. Find 2 - M3 x 25mm plastic screws and nuts and place them through the remaining holes of the motor mount. These should press the motor mount pieces tightly to the aluminum arm and brace both sides of the arm. Do this for all four of the arms.

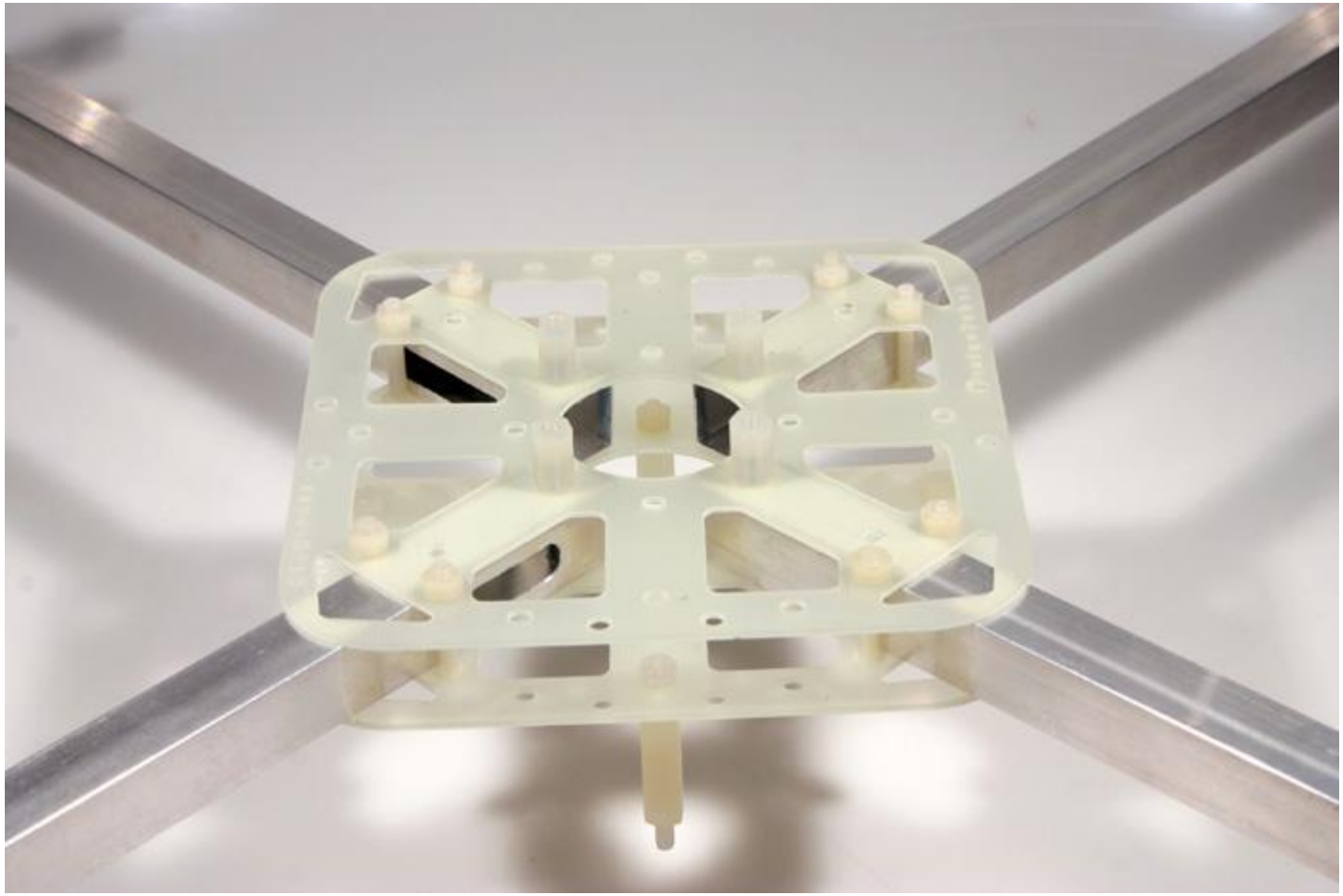




Slide each aluminum arm between the nylon screws of the Main Frame boards and pass the steel screw through them. Secure the steel screws on the bottom with the 4 - M3 x 10mm Female-Female Nylon Spacers found in Bag #4 and tighten the nylon screws so that the Main Frame boards are snug with the aluminum arms.

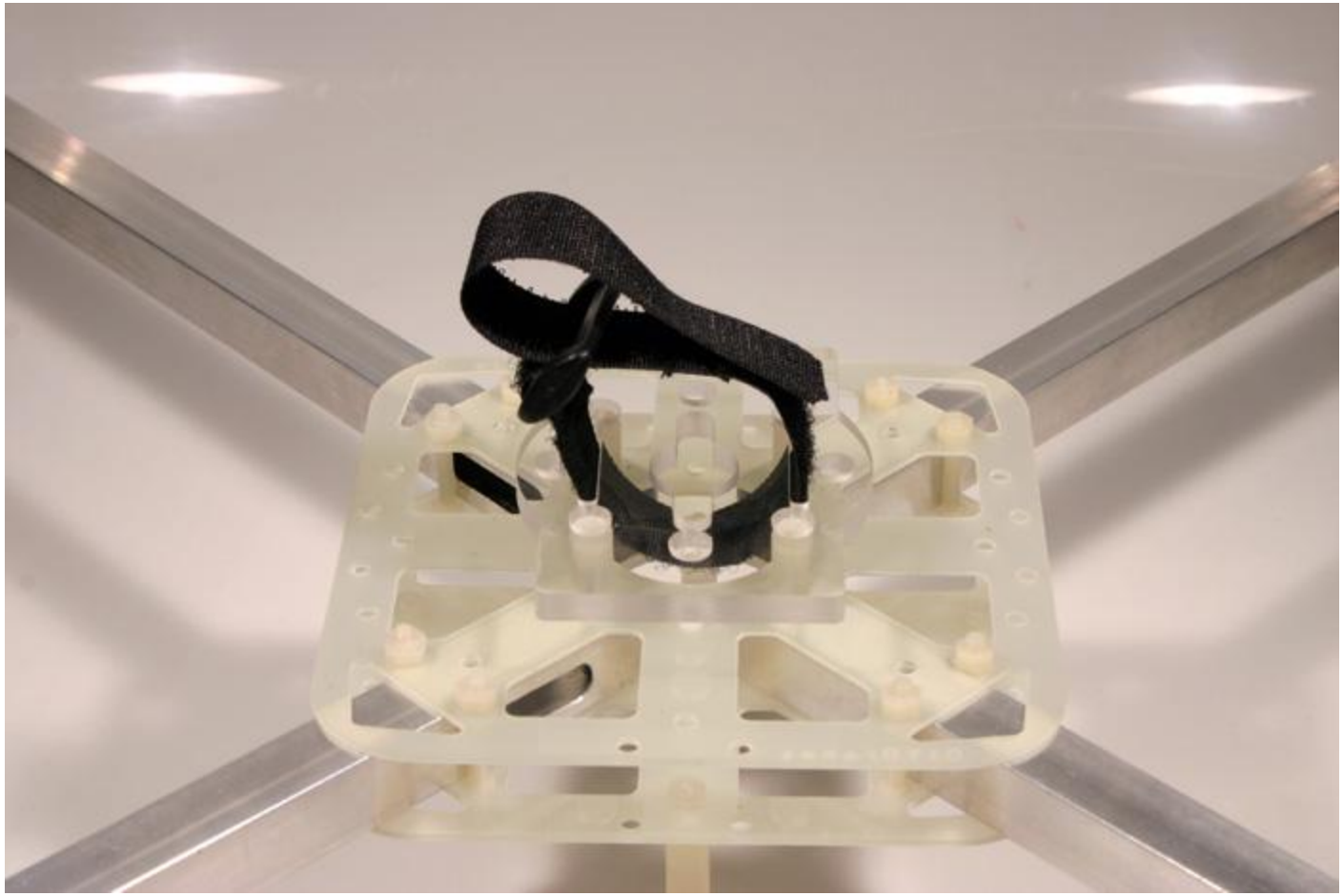


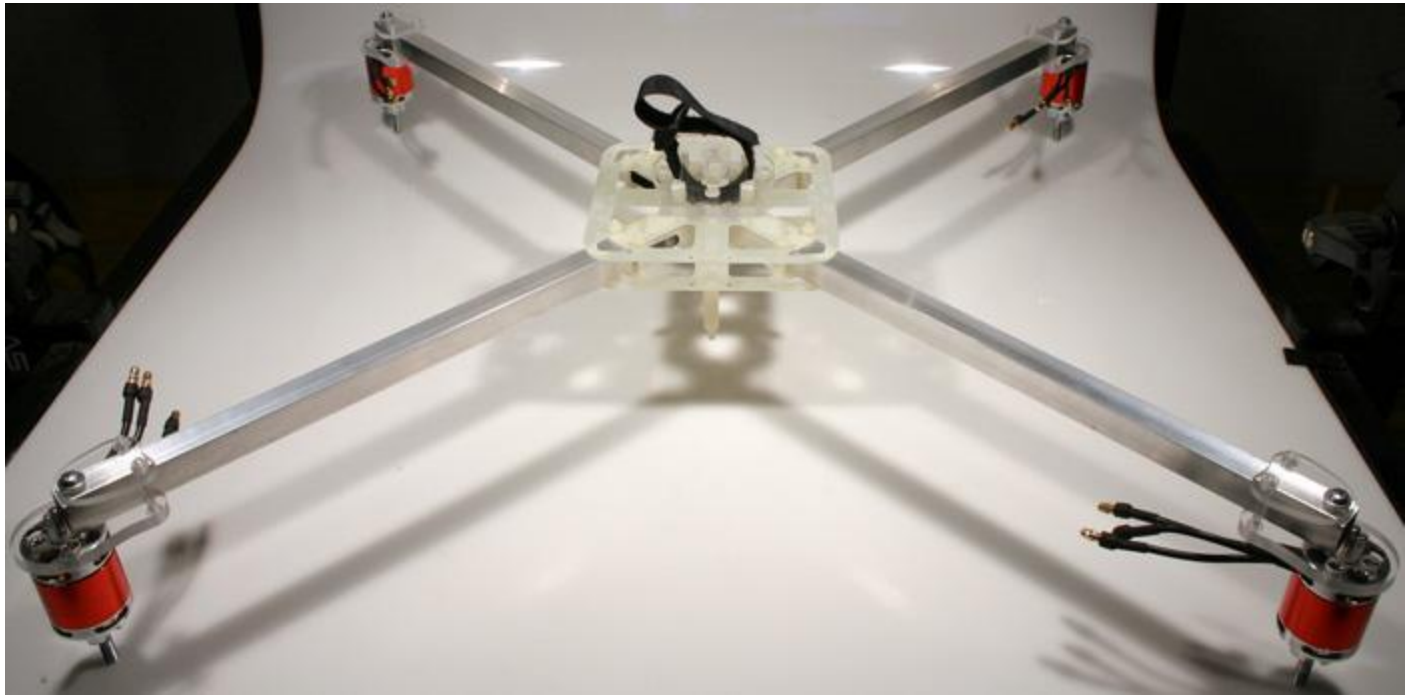




Take out the Battery Mounting Plate and the 4 - M3 x 6mm Nylon Screws from Bag #4 along with the velcro battery strap from Bag #9. Thread the velcro strap through the batter mounting plate ensuring that you have the correct side of the strap so that it will stick to itself. Make sure the recessed screw holes are also on that side.

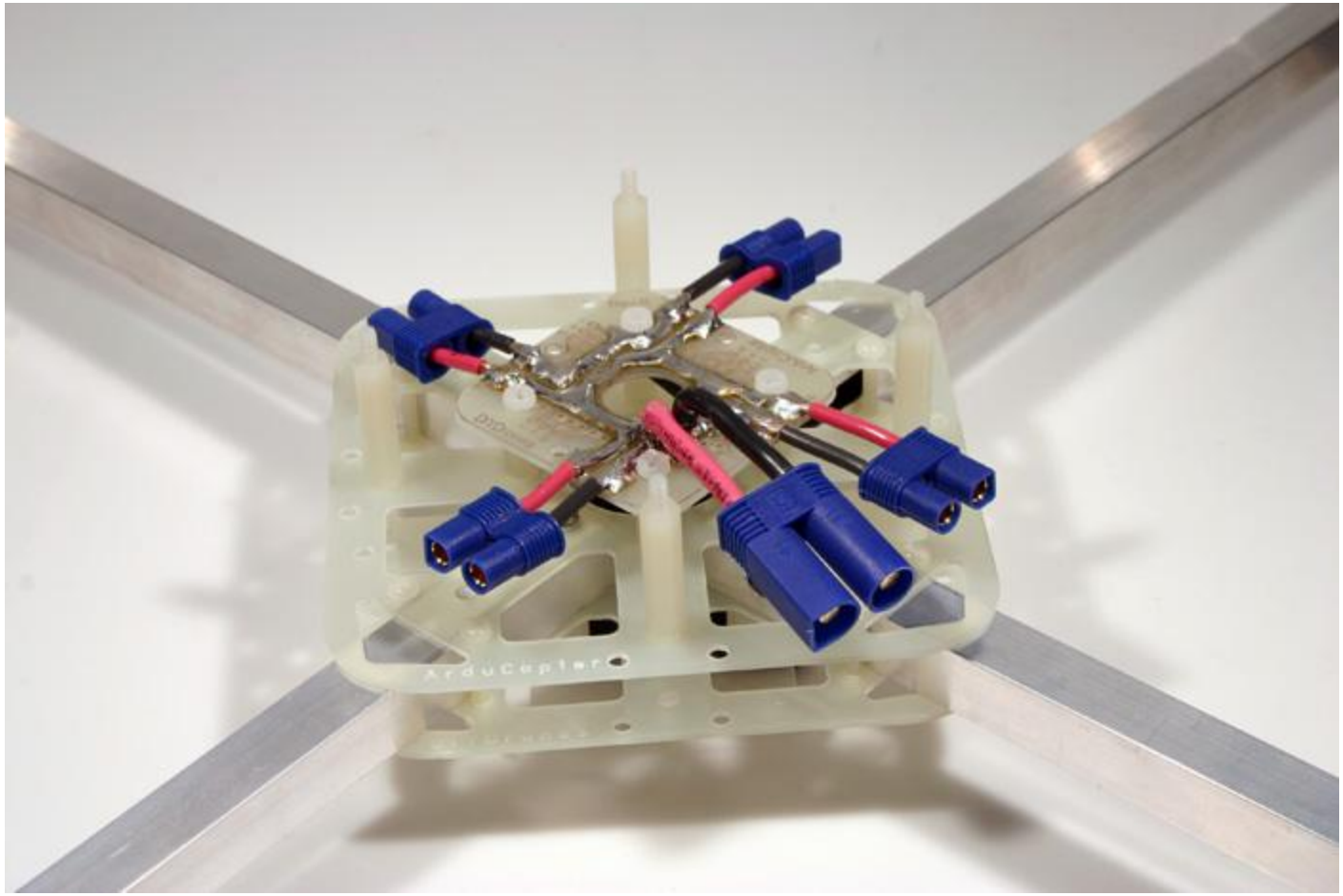
Line the screw holes up with the nylon spacers on the bottom of the Main Frame board taking note of which direction you'll want the battery to be. In my 'X' configuration, I want the length of the battery to be facing straight forward and aft. Secure it with the 4 nylon screws.

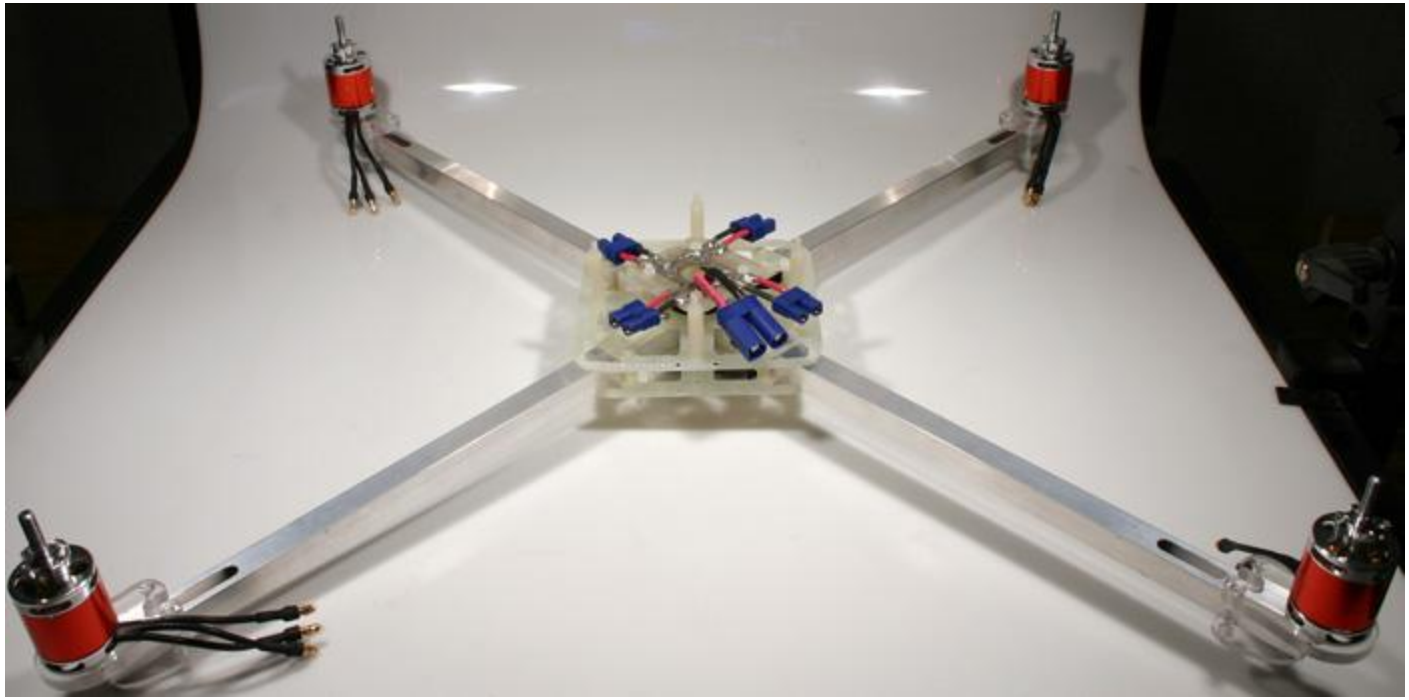




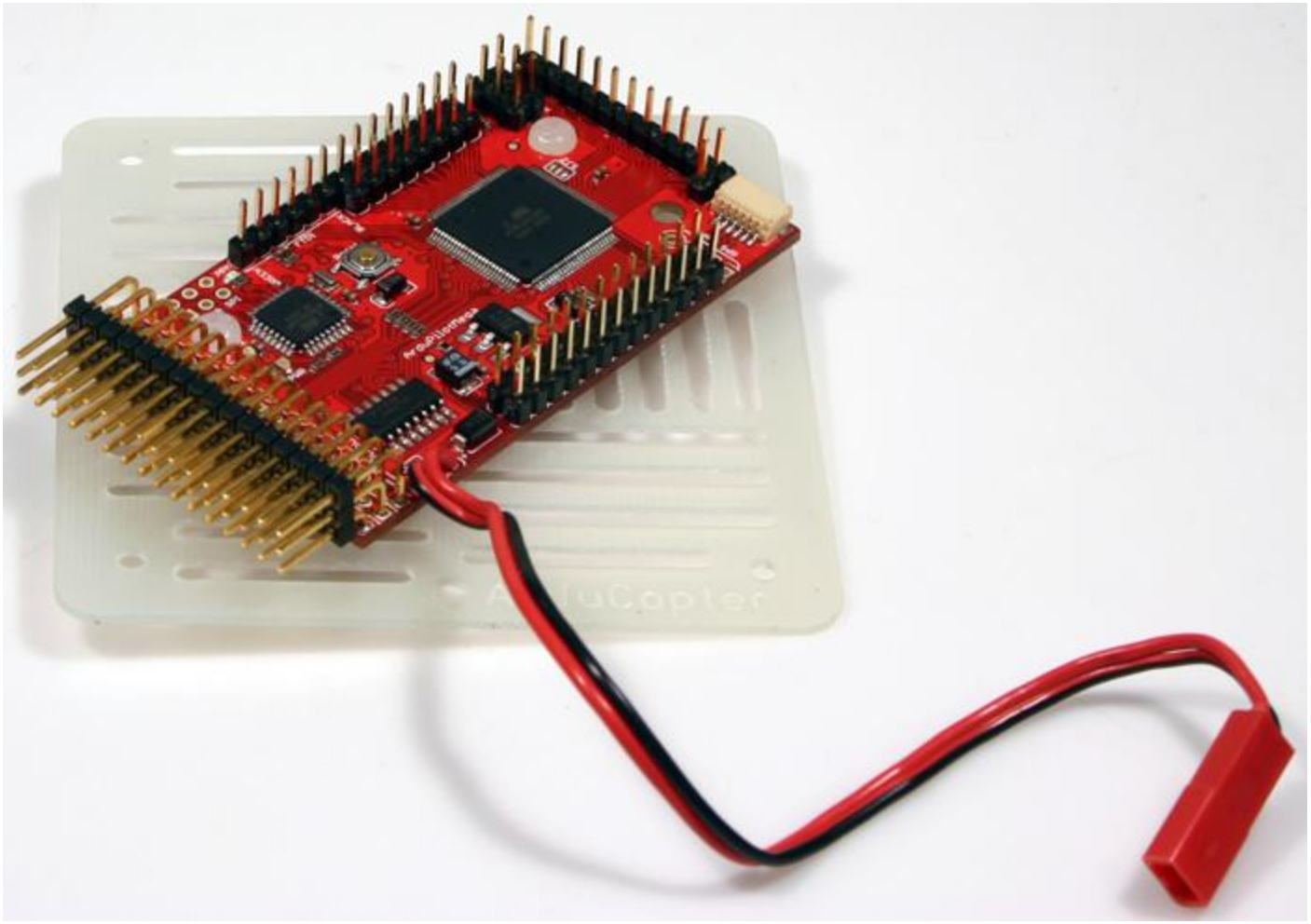
If you haven't finished your P-PCB yet, follow the steps here before preceding.

Place the PCB into the 12mm nylon spacers and secure it with 4 - M3 x 6mm Nylon Screws from Bag #3.

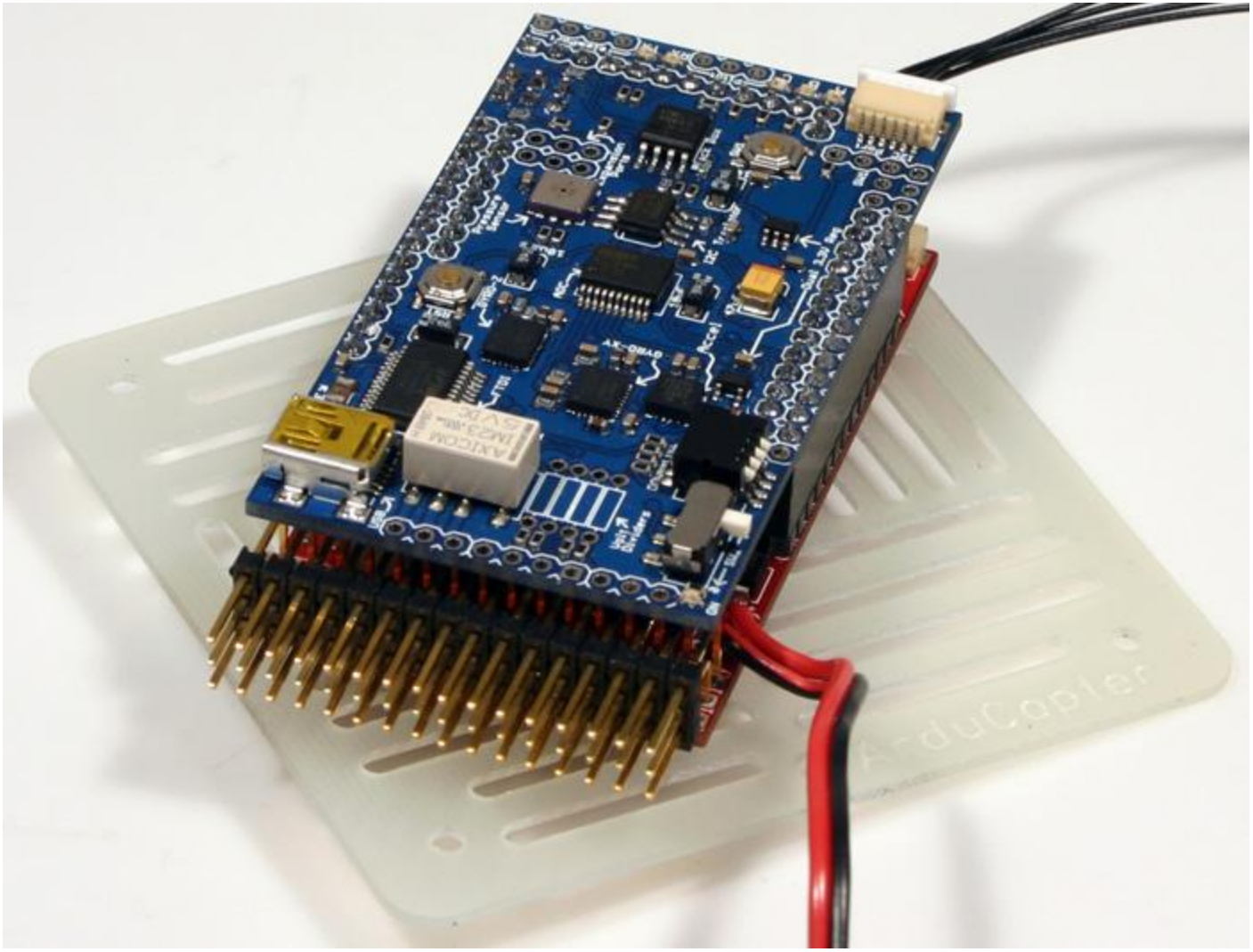




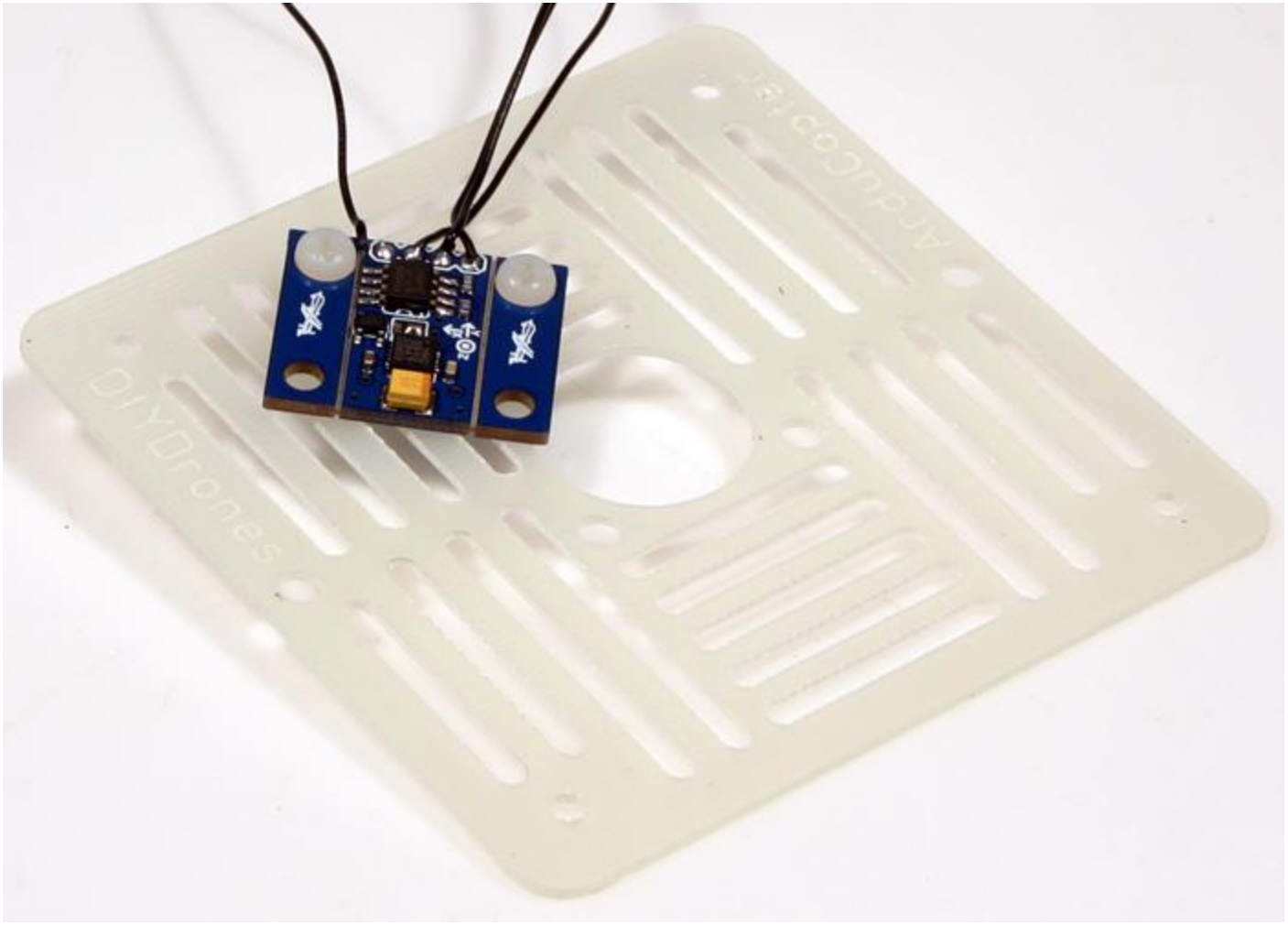
Now, take out a carrier board from Bag #1. I used some nylon screws from the extra parts bag to mount my ArduPilot Mega board. ~~Normally, you should mount the board so that the edges are parallel with the side of the carrier board, but mine is mounted at a 45 degree angle. I'll explain later in my thoughts section.~~ **Mount the board so that the edges are parallel with the side of the carrier board. This is different than seen in the pictures below. *The pictures will be updated in the future***. Keep in mind when attaching your board that the edge opposite the right-angle headers must be the front of the ArduCopter.



Place the IMU shield on top of the APM.

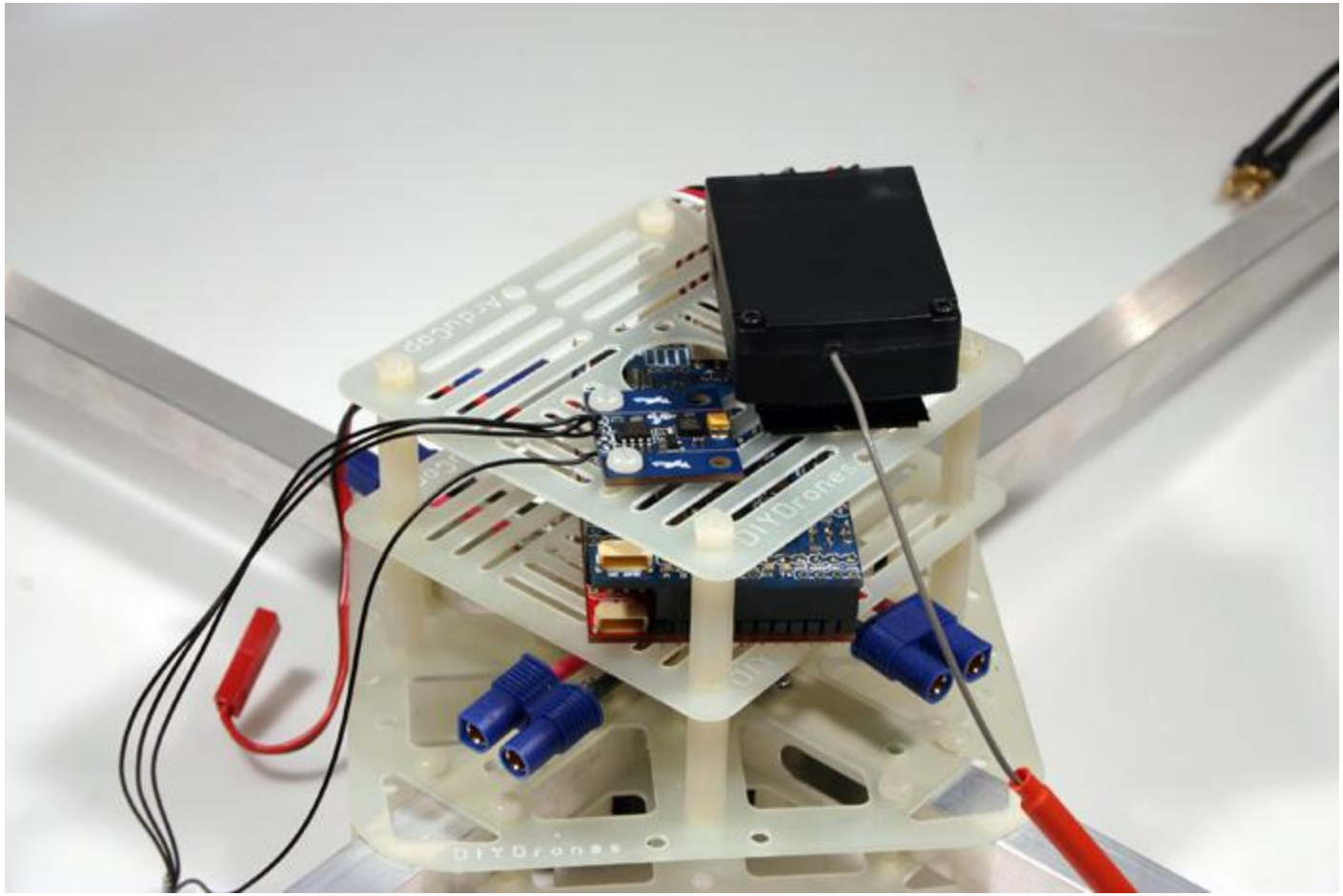


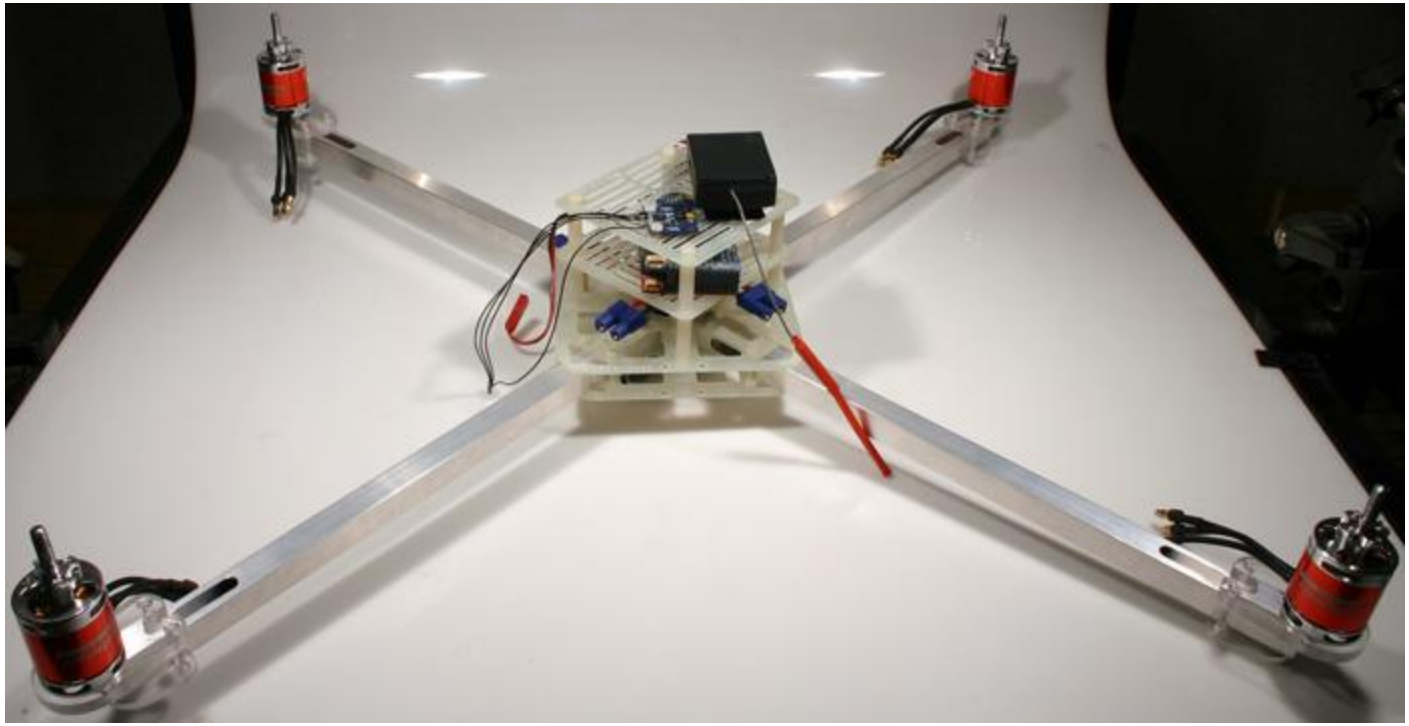
Take the second carrier board and attach the magnetometer (if you have one) as well as the receiver from your transmitter. If you take a look at the magnetometer, there are two arrows labeled 'x' and 'y'. ~~The coordinate frame for aircraft designates the x-axis out the nose and the y-axis out the right wing. So, in the ArduCopter, the x-axis should be pointing to the front and the y-axis to the right.~~ **For default mounting, the pins of the magnetometer should face the rear of the ArduCopter.** ***Note that the pictures below are wrong.***



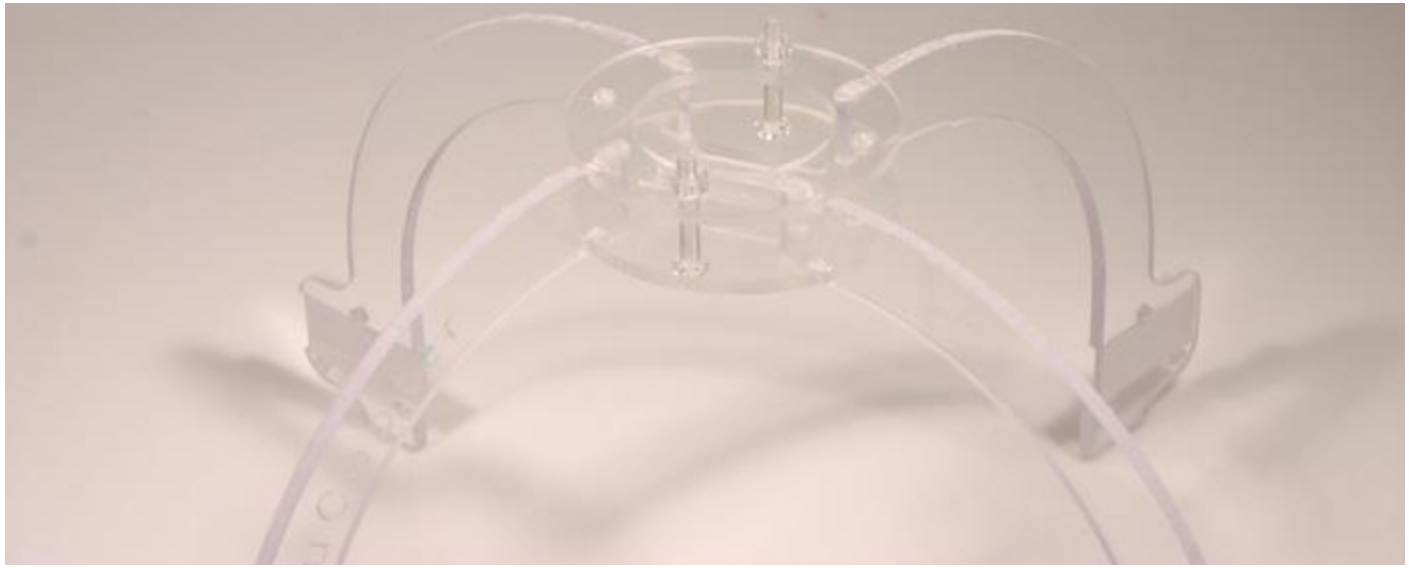
A velcro sticker is supplied in Bag #9 for mounting the receiver, but you may not be able to use it. I later found out the velcro adds too much thickness to the point where the protective dome doesn't fit, so I have mine secured instead with zip ties.

Place the carrier board that has the APM and IMU on top of the 25 mm Nylon spacers already on your frame. Take the other 4 - M3 x 25mm Male-Female Nylon spacers out of Bag #9 and screw them on top of those. Place the second carrier board with your magnetometer and receiver on top of those spacers and secure with a M3 nylon nut.


DIYDrones

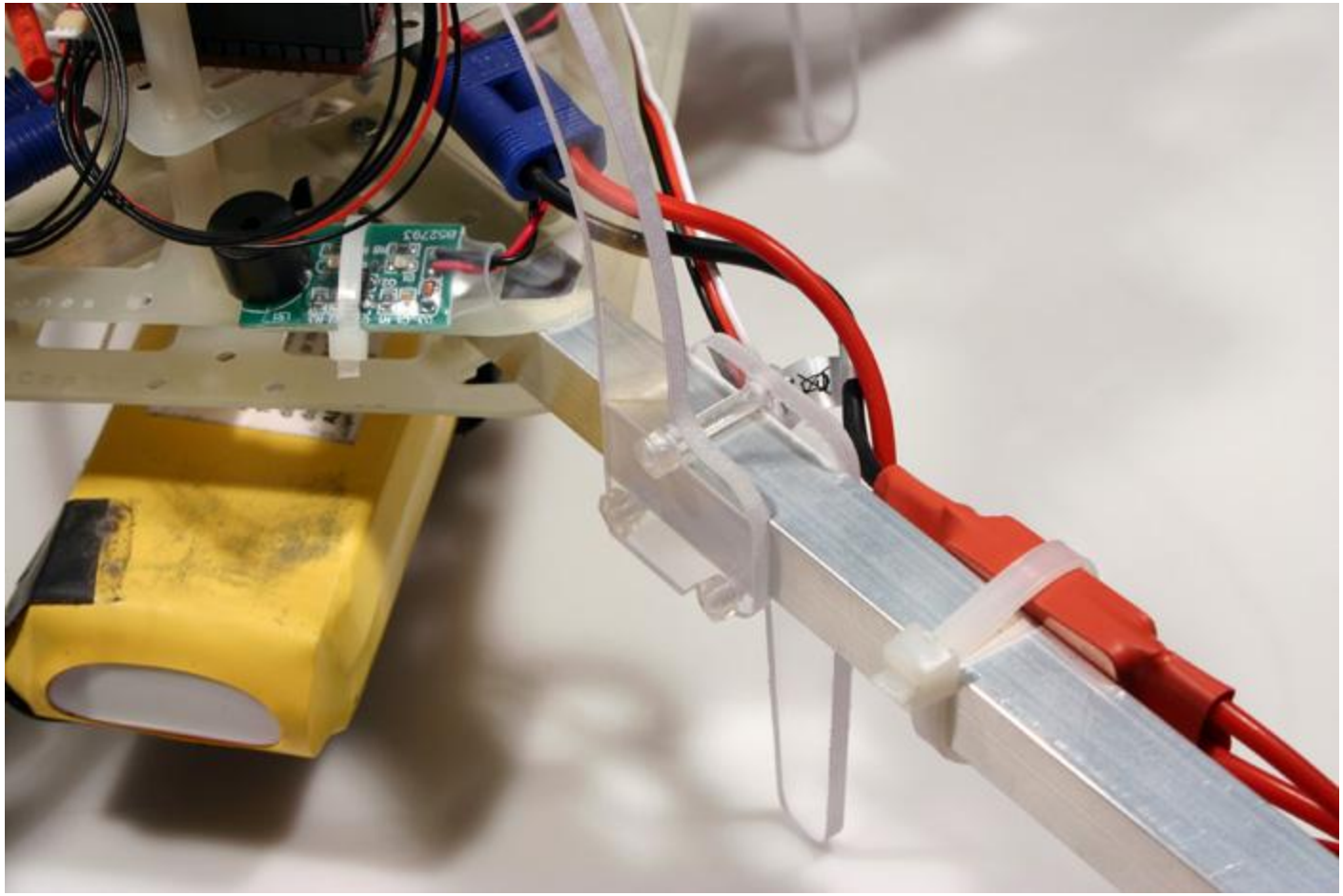


All of the parts for the protective dome and landing gear are in bags #5 and #6. The upper and lower dome center have elongated slots in them that the protective dome arches slide into. Secure with 2 - M3 x 20mm Plastic screws and two M3 plastic nuts. The top carrier plate will have more room if you hold the dome centers with the nuts being placed on top and the screw head on the bottom. Two extra nuts are included for both sides of the dome center, but I don't think they are necessary as the dome arches are supporting it.

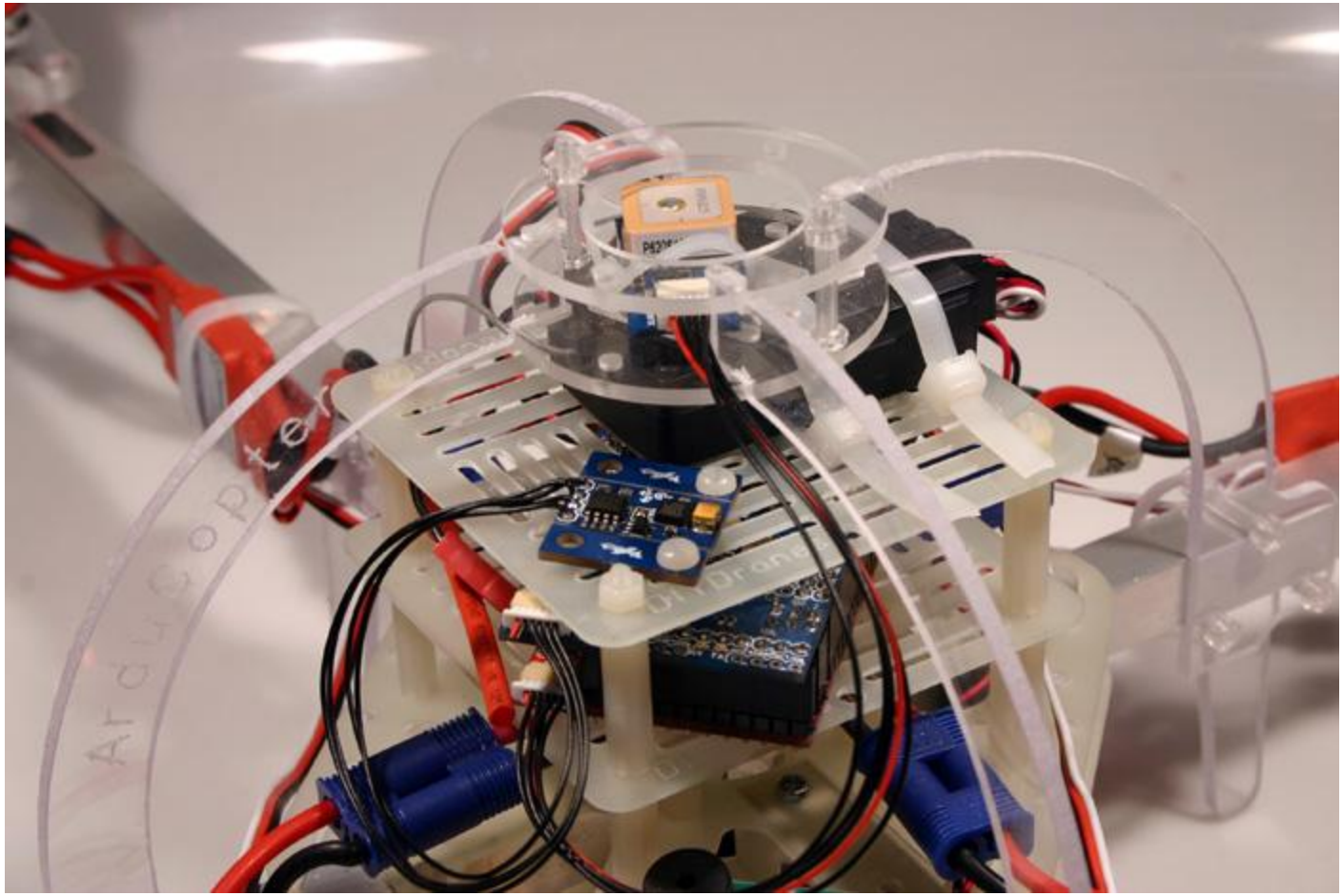


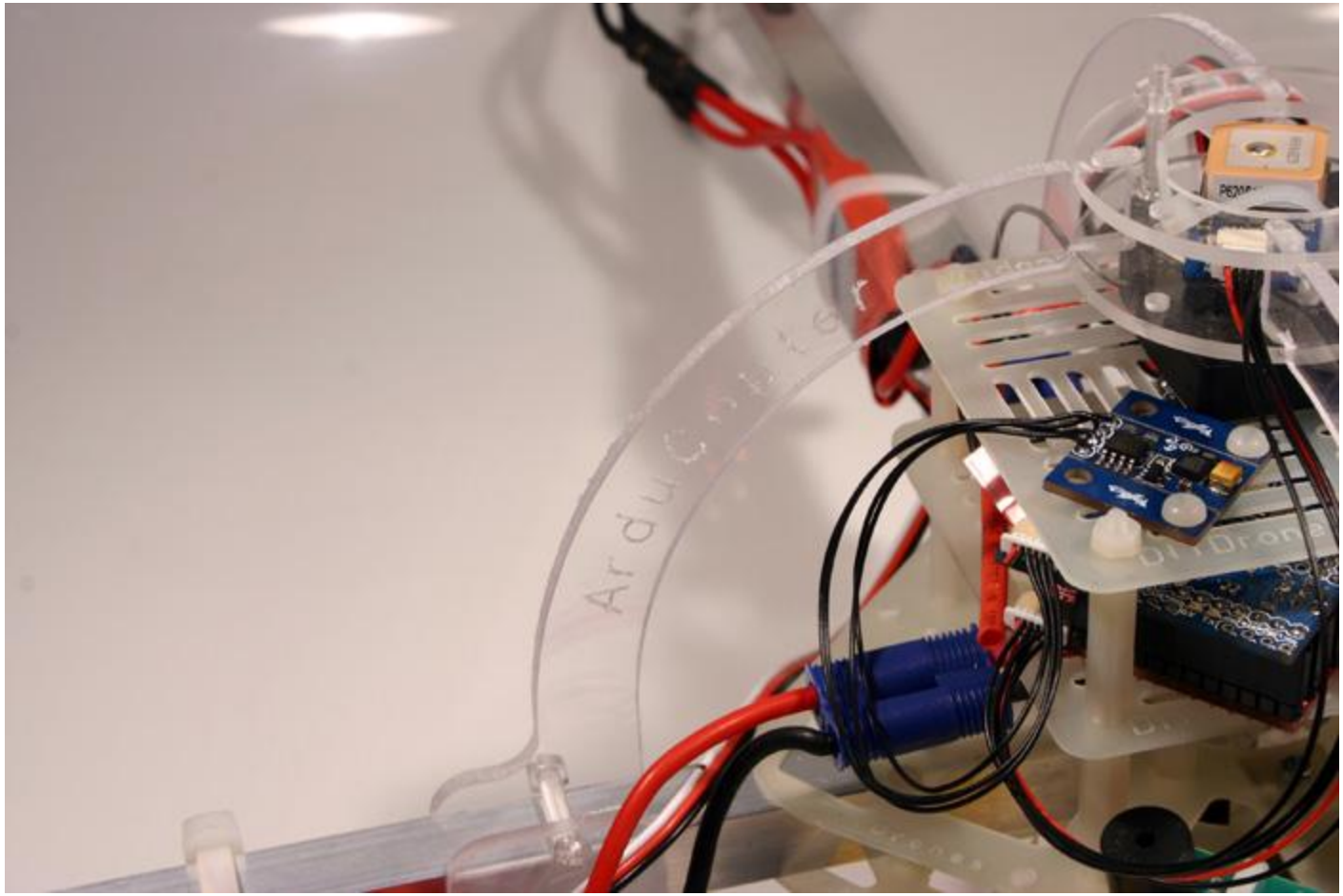
If you look at the end of the arches, notice that there is material removed from one side of it between the 3 screw holes. This is the side that is against the aluminum arm.

The Landing Gear Fins also have material removed on one side and they attach with 3 - M3 x 20mm Plastic Screws and Nuts. Keep the nuts loose until all of the landing fins are on so that you can center the entire Protective Dome on the Arducopter. Once it is centered, tighten the nuts. (Some people may find it easier to attach the Protective Dome Arches and Landing Fins to the aluminum arms and then snap the Dome Centers in place)



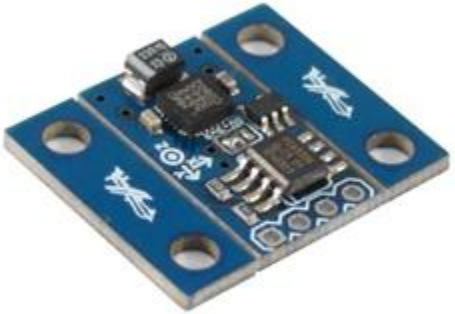
The GPS is placed in the top dome and is secured with a zip tie.





Quad_Magnetos

Using a magnetometer

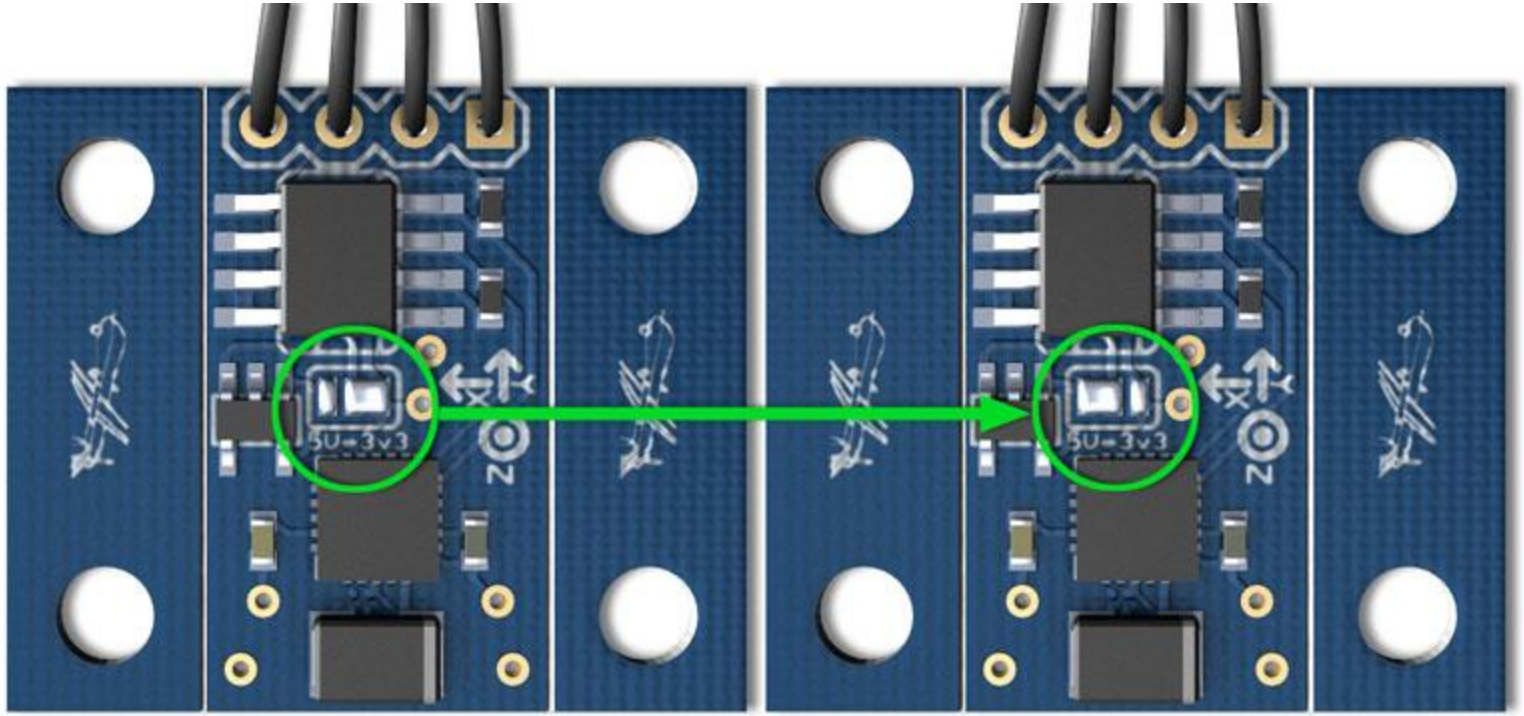


If you're flying quadcopters or helis, which are often hovering in one position, the GPS (which can only calculate a directional vector when it's in forward motion) will not be able to consistently correct the drift in the yaw gyro. For those applications, you may want to add a magnetometer, which can correct the yaw even when the vehicle is not moving.

You can attach the HMC5843 magnometer to the APM IMU shield's I2C sensor port, which looks like a GPS connector but says "No GPS".

The best way to do this is to modify a GPS cable, cutting off the connector on one side and soldering the wires to the right pads on the magnetometer board. The pins are even in the correct order but if there's any doubt, you can look at the bottom of the shield where you should see "SCL", "SDA", "+5V", "GND" written. These should be matched up to the pins on the magnometer.

Ensure that the solder blob jumper on the magnetometer is set to 5V before connecting it. You can see the instruction bellow:



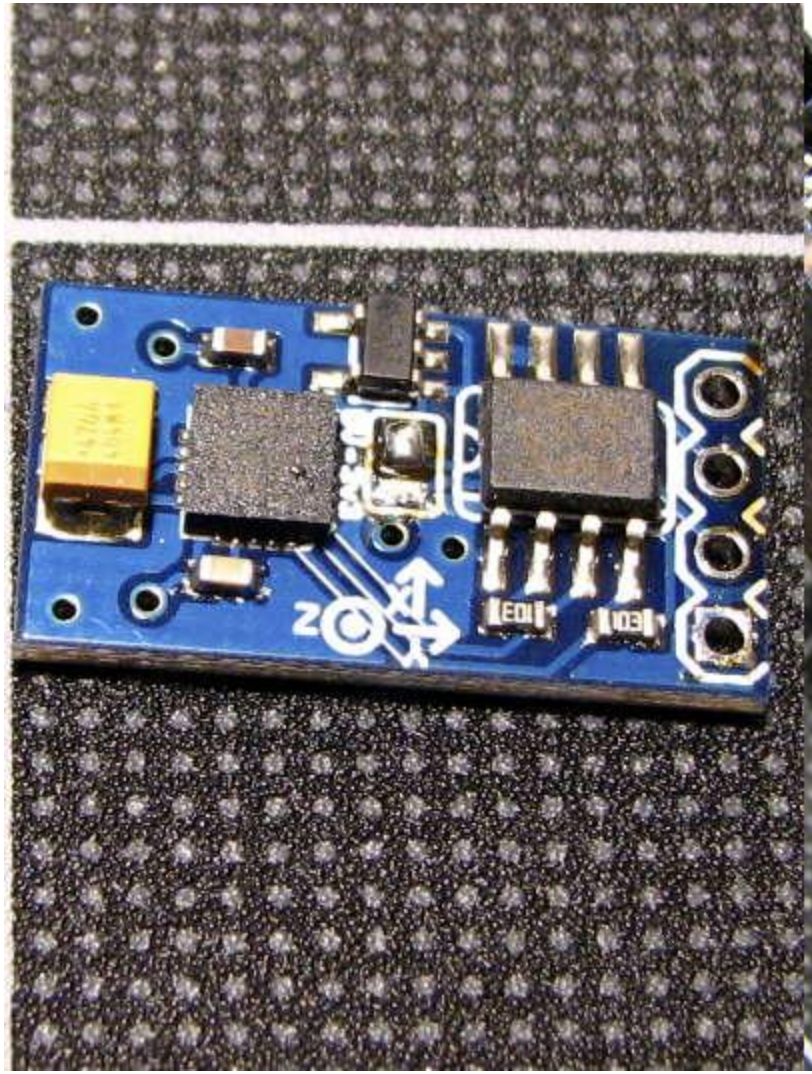
Change the jumper from 3V3 to 5V, before attach it on IMU Shield/Oilpan

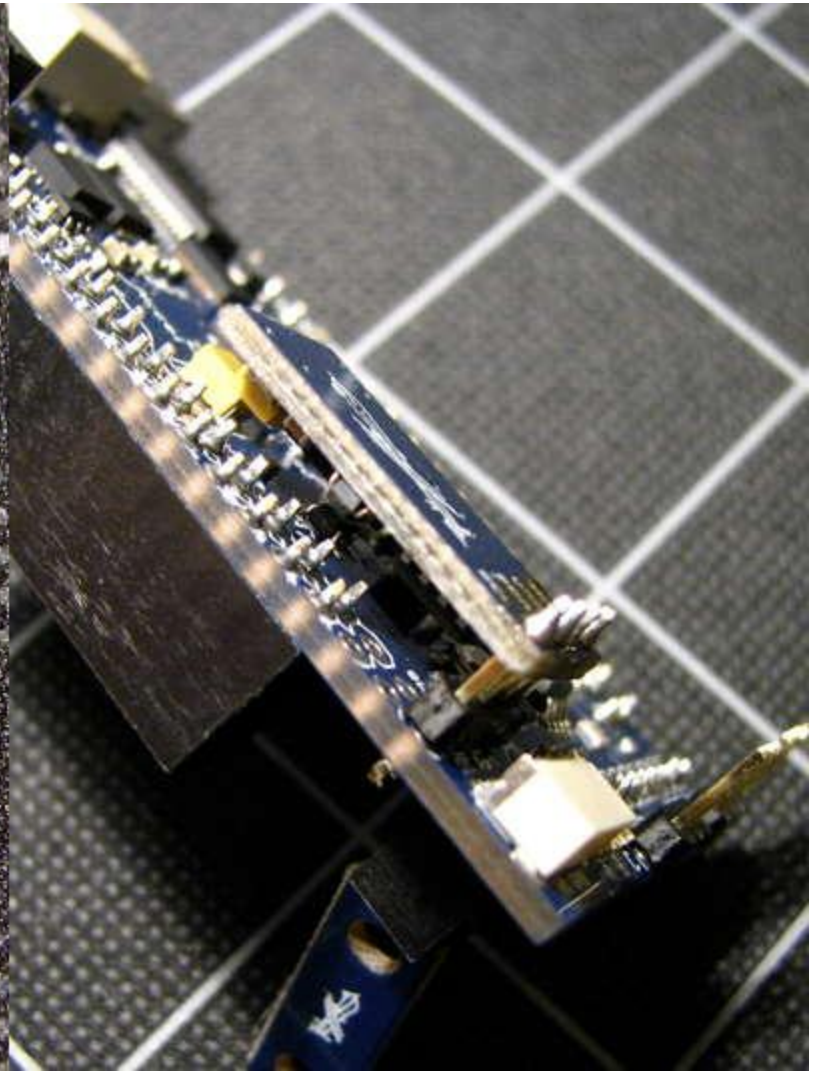
Alternative mounting option

If ou don't plan on using the 4-pin I2C connector on the shield for anything else, you can mount the magnetometer directly on top. Check the 3.3/5V solder jumper on the magnetometer first - some ship set to 3.3V, but the shield supplies 5V to this header. In the image below, the jumper has been shifted to the 5V setting.

Once the input voltage is selected correctly, you can use a 4 pin header as a standoff. Note that using pin sockets on the magnetometer board is not recommended, as they don't hold it very securely and without some other form of mechanical support it will flop around.

Take care to align the magnetometer and shield edges. Precision here will reduce the amount of trimming you have to do later. Note that to achieve correct signal routing you can mount the magnetometer as shown here, or with the component side up but overhanging the edge of the shield.


LED1
LED3



Quad_GPS

GPS

MediaTek GPS module

The state-of-the-art 66 channels MediaTek MT3329 GPS Engine

High sensitivity: Up to -165dBm tracking, superior urban performance

USB/UART Interface

Build-in patch antenna for optimal sensitivity

DGPS(WAAS, EGNOS, MSAS) support (optional by firmware)

Maximum update rate : up to 10Hz (optional by firmware)

RoHS compliant

Note that the new MediaTek has custom and exclusive "DIYDrones" firmware that allows the unit to output an efficient and very compressed binary protocol. You can still change between NMEA and Binary protocol with standard MTK messages, and switch the refresh rate between 1hz to 10hz, or set any standard serial baud rate (by default is set to 38400 bps and custom binary protocol).

Arduino example for decoding the custom binary protocol is provided. Remember to install the library folder "GPS_MTK" inside the Arduino IDE "libraries" folder "\arduino-00XX\libraries\\". The library and the source code is located in the same ZIP file, click here to download.

Features:

-Based on MediaTek Single Chip Architecture.

-Dimension：16mm x 16mm x 6mm

-L1 Frequency, C/A code, 66 channels

-High Sensitivity：Up to -165dBm tracking, superior urban performances

-Position Accuracy：< 3m CEP (50%) without SA (horizontal)

-Cold Start is under 35 seconds (Typical)

-Warm Start is under 34 seconds (Typical)

-Hot Start is under 1 second (Typical)

-Low Power Consumption：48mA @ acquisition, 37mA @ tracking

-Low shut-down current consumption：15uA, typical

-DGPS(WAAS, EGNOS, MSAS) support (optional by firmware)

-USB/UART Interface

-Support AGPS function (Offline mode : EPO valid up to 14 days)

UBlox GPS module

The powerful GPS based on Ublox 5 chip-set and Saratel helix antenna.

It comes pre-programmed for ArduPilot and Paparazzi UAV.

for the UAV DevBoard, select the "Default Factory Settings"(Note: It is a free service and it's not guaranteed)

You can choose to add the uBlox adapter.

Features

u-Blox 5H chipset Sarantel omni-directional Geo-helix S-type active antenna Real 2Hz Refresh rate

Can be used up to 4Hz Fifty channels Supports UBX, NMEA and USB&NMEA High immunity to RF interference Firmware upgradable

Quad_TelemetryPage

(Optional) Telemetry

Hooking up your Xbee wireless modules

Adding wireless telemetry is not difficult and can extend the capabilities of your UAV immensely. We recommend using Xbee wireless modules, which have a range of more than a mile.

The first thing to keep in mind is that you should use Xbee modules in a different frequency range than your RC equipment.

If you have 72Mhz RC gear, you can use 2.4Ghz Xbee modules. In that configuration, we use these Xbee Pro wireless modules. But please note that the only reason to use 2.4Ghz Xbee equipment is if you also want to use 900 Mhz video transmission; otherwise, it's better to use 900Mhz Xbees (see next paragraph), which have longer range.

If you have 2.4Ghz RC gear, you should use 900Mhz Xbee modules. In that case, we use this Xbee Pro with the wire antenna for the aircraft, and this Xbee Pro with a SMA antenna connector (and a good 900Mhz antenna) on the ground.

All Xbee modules need adapters to work with APM. You have two choices:

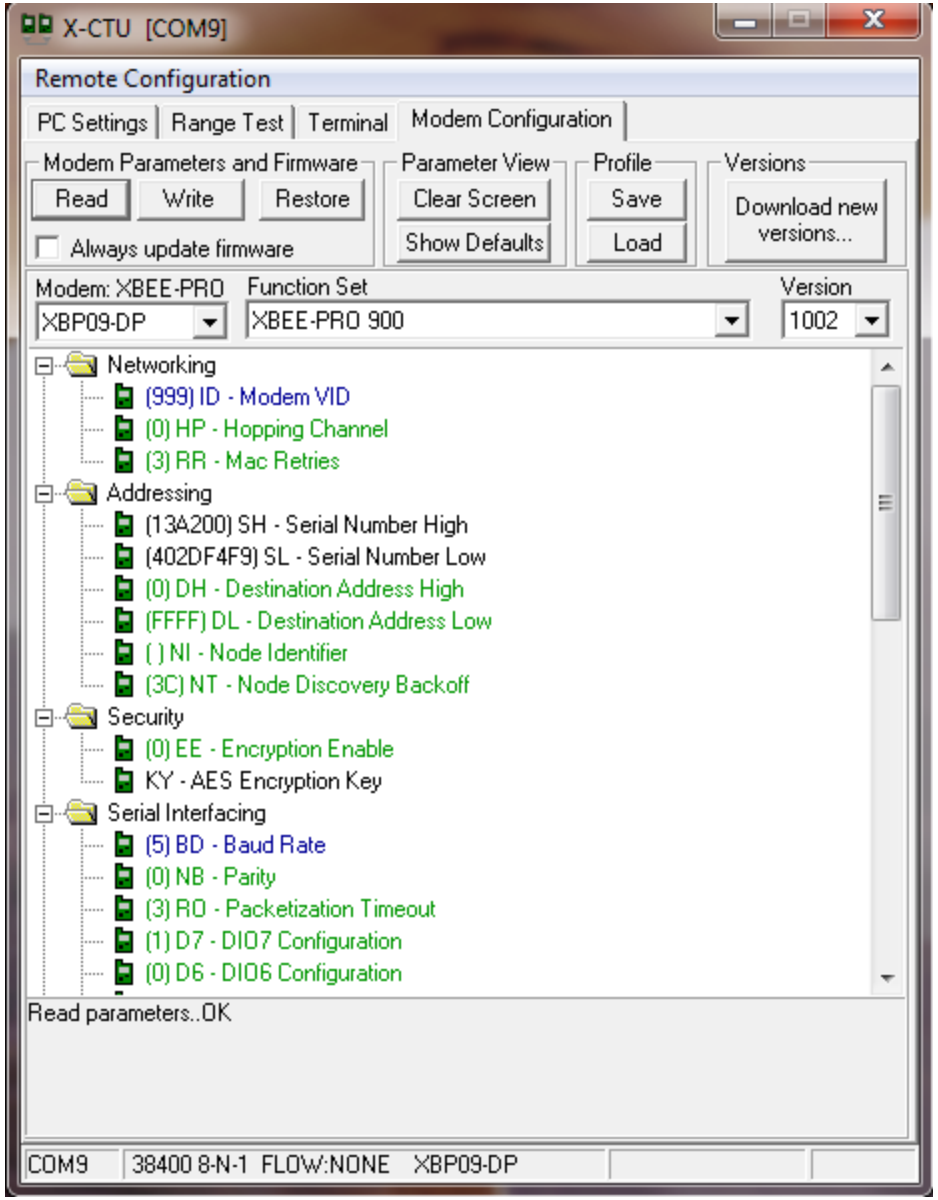
- An Adafruit adapter board on the aircraft side, and a Sparkfun USB adapter board on the ground/laptop side with a USB cable.
- Two DIY Drones XtreemBee adapters, with a FTDI cable on the ground.

Setting up the Xbee modules

The Xbee modules ship with a default of 9600bps, which you must change to match the APM's serial speed of 115200bps; set your Xbee modules to match this speed.

Connect each one of the them to the Sparkfun USB adapter board, plug the USB cable into your PC, and use Digi's X-CTU utility to select the right serial port and communicate with them. Remember to initially set the utility to 9600bps to contact the new Xbee modules, and than after you've changed the speed, change the utility's serial speed accordingly. You should also give the modules unique Network IDs (VIDs) so they will be paired. Just use any 3-digit number, and just make sure you have set it the same on both modules. (Note: If you will be flying near other UAV planes make sure to verify the Network IDs are unique and not used by others in your vicinity.)

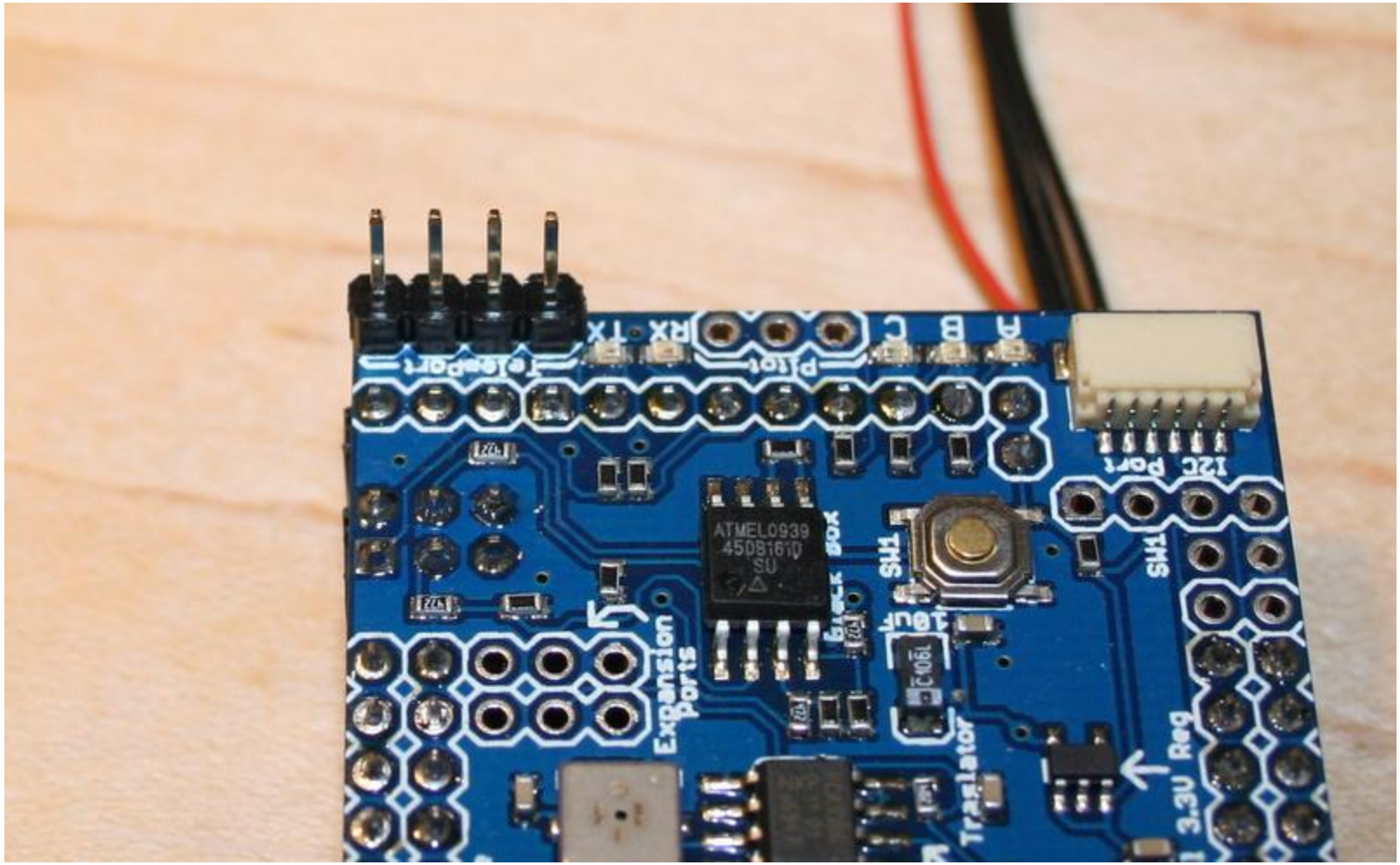
This is what the setting should look like when you click "Read" in Modem Configuration tab of X-CTU (we're using 999 as the VID here as an example):



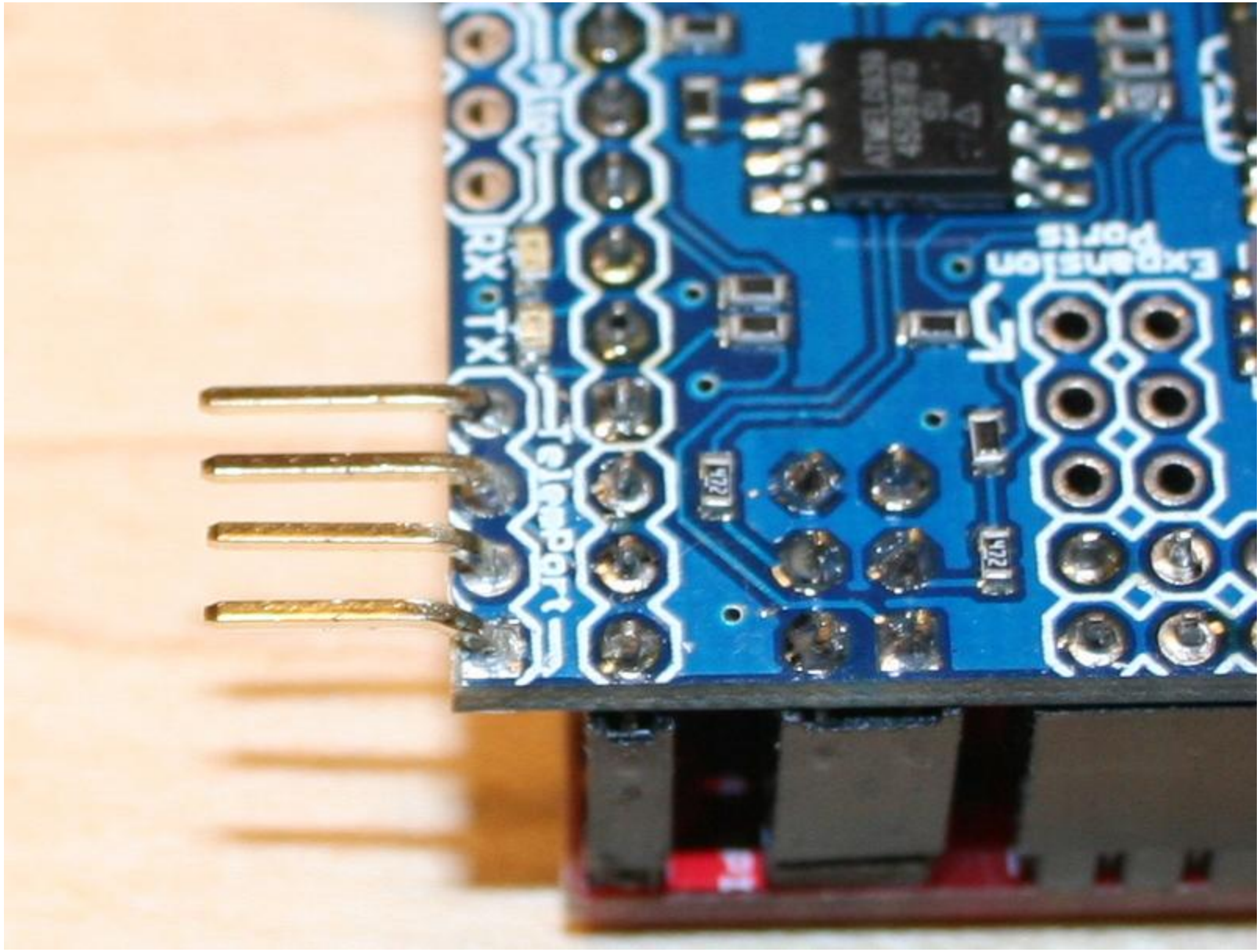
Connecting your airborne Xbee to APM

Hardware connection:

Solder four breakaway header pins in the IMU shield's "Telecom" port as shown below:

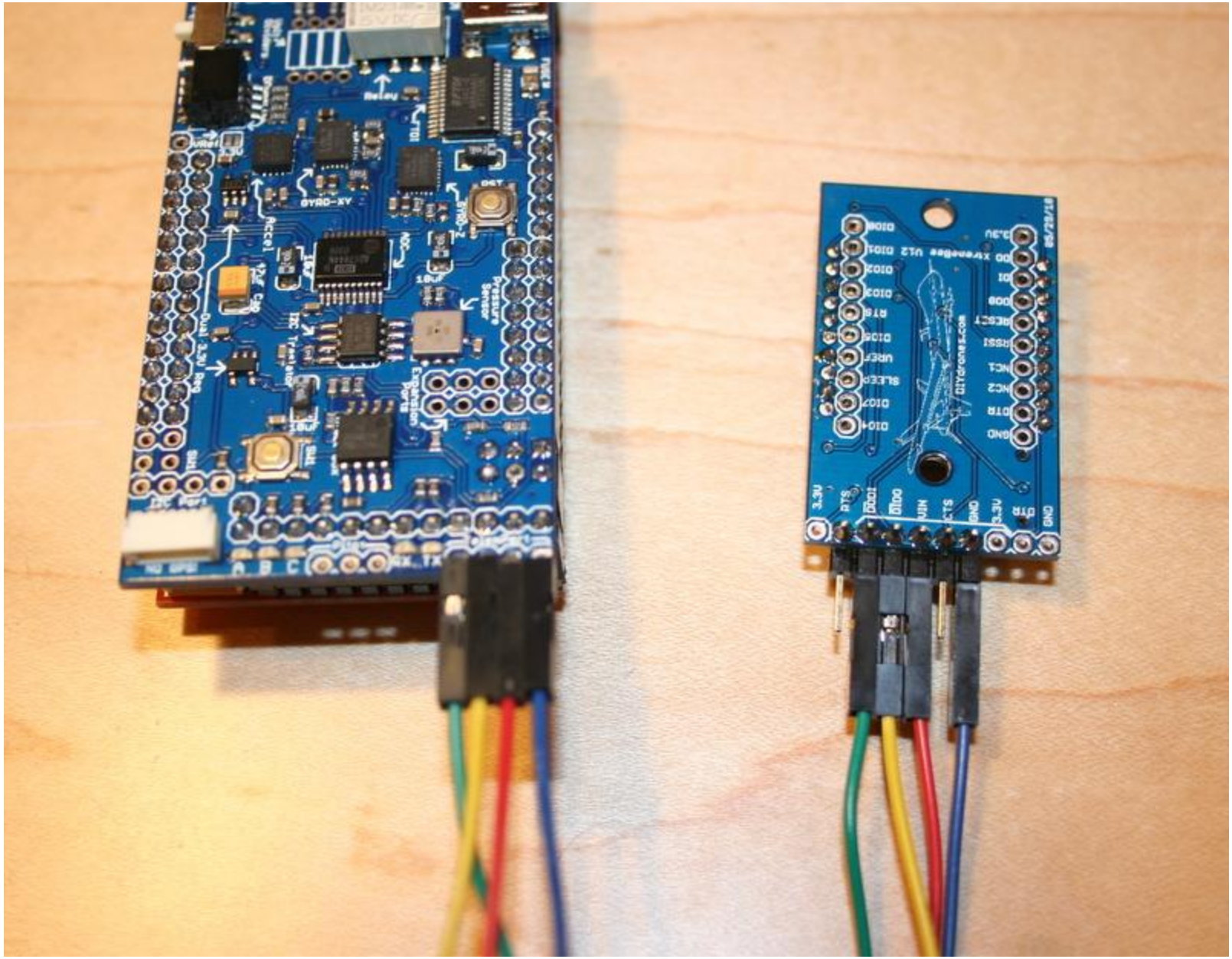


Now pull the black plastic strip off the pins (it's a little tight, but wiggling with a pair of pliers will do it), and bend them over 90 degrees as shown below:

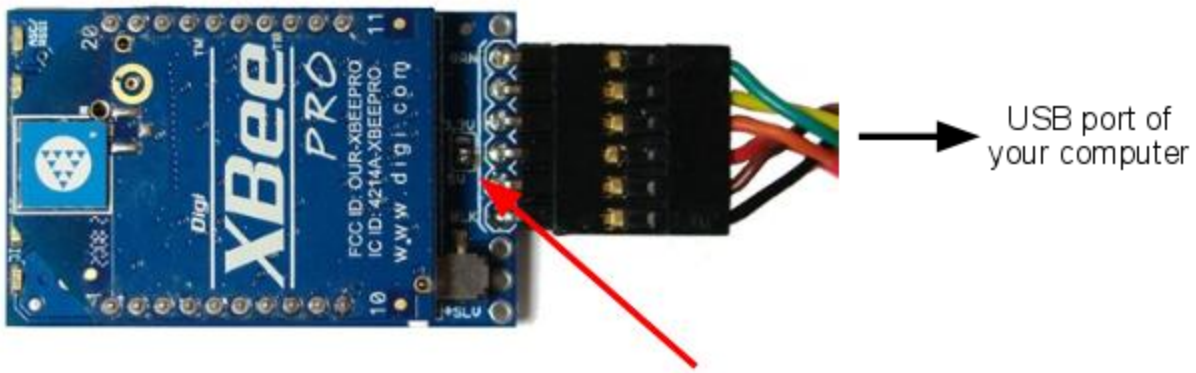


If you're using the DIY Drones XtreemBee adapters

Connect the adapter (with the Xbee plugged in) to the APM shield pins as shown below via four individual connector wires. Your adapter should be in "Master" mode. ("Master" and "Slave" just reverse the TX and RX pins).



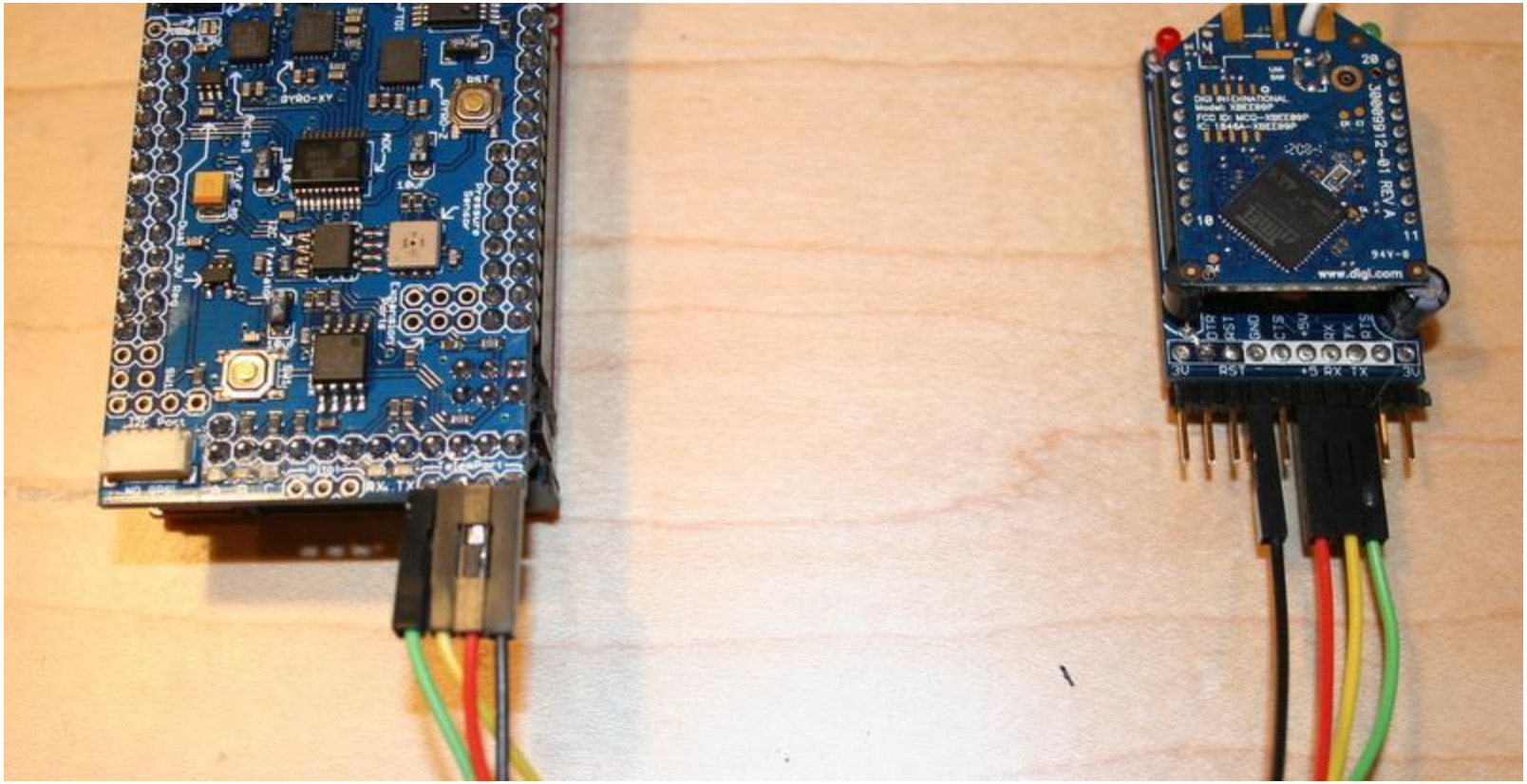
On the ground side, connect the other XtreemBee adapter/Xbee to a FTDI cable as shown below and plug that into your USB port. The adapter should also be in Master mode.



If you're using Adafruit/Sparkfun adapters

1. The XBee with adapter board is connected to the IMU/shield via four individual connector wires.
2. Connect the TelemPort's Out pin to the Rx pin of the Adapter board. Connect the TelemPort's In pin to the Tx pin of the Adapter board. Connect the TelemPort's 5V pin to the +5V pin of the Adaptor board.Connect the TelemPort's Gnd pin to the Gnd pin of the Adaptor board.

This is what the plane side XBee unit looks like finished and connected.



On the ground side, connect the other Xbee module via the Sparkfun adapter to your laptop with a USB cable.

Testing the connection

If you open up a terminal program on your laptop (you can use the Arduino IDE's serial monitor for this, too), select the correct serial port, and set the baud rate to whatever you set the Xbee modules to above. Once you do this, you should see ArduPilot telemetry coming in. Anytime there is a "Serial.println" in the code, that data will be sent through the Xbees to the ground. You can record any data you want, and even datalog from the ground! You can also open the Ground Station software, setting the right port and baud speed) and it should begin to show ArduPilot data.

Additionally, if you want to test the range of your Xbee link, connect the plane-side Xbee module's RX and TX pins together to create a loopback circuit and use the X-CTU utility's range test function. For the modules we are using you should get around a mile.

Test code

ArduPilot Mega has four serial ports so all the usual Arduino serial commands now take a specifier to say which port you want to read from or write to. For example: Serial1.print(), Serial2.print(). The port connected to the USB/FDTI connector is Serial0. The port connected to the Telecom pins is Serial3.

Here's a quick demo that will print to all four ports so you can check to see that your Xbee connection is working. Just set your Arduino IDE to the Com port you've connected your "ground" Xbee to, and set the serial speed to 115200. You should see "Port 3" repeated in the serial monitor.

Unbricking an Xbee

IMPORTANT NOTE: Sometimes Xbee modules get corrupted due to signal coming in before power on bootup. To avoid this, ALWAYS disconnect the signal wire (the blue one in the photos above) to the onboard Xbee adapter before powering up. Only reconnect them after the rest of the UAV has power.

If you're finding that yours stops working (green LED on Adafruit adapter doesn't come on), instructions to reload the firmware follow:

Using the Sparkfun USB explorer board:

1. Take the module out of the interface board.
2. Connect the interface board to the computer.
3. Open X-CTU make sure Baud Rate is set to 9600
4. Go to "Modem Configuration"
5. Put a check in the "Always update firmware" box
6. Select proper modem from drop down menu,
7. Select proper function set and firmware version from the drop down menus (the 900Mhz ones recommended above are "XBP09-DP").
8. Click on the "Write" button. After a few seconds of trying to read the modem, you will get an Info box that says Action Needed. At this point, CAREFULLY insert the module into the interface board. After a few seconds, this should trigger a reloading of the firmware.
9. You may get the info box again a short while after; if so just use the reset button on the interface board, or if you board doesn't have a reset button connect the reset pin to ground.
10. Once you've confirmed that it's working again, make you sure you reset its baud rate and ID number to match your other module.

(Thanks to Doug Barnett for these tips)

Quad_LoadingPage

Loading the code

Software/Firmware:

Depending on your OS you may need some drivers for the USB-port of the shield.

Software needed to upload the firmware to the board: Download here

Arduino troubleshouting:

APM-Libraries and firmware: Download here

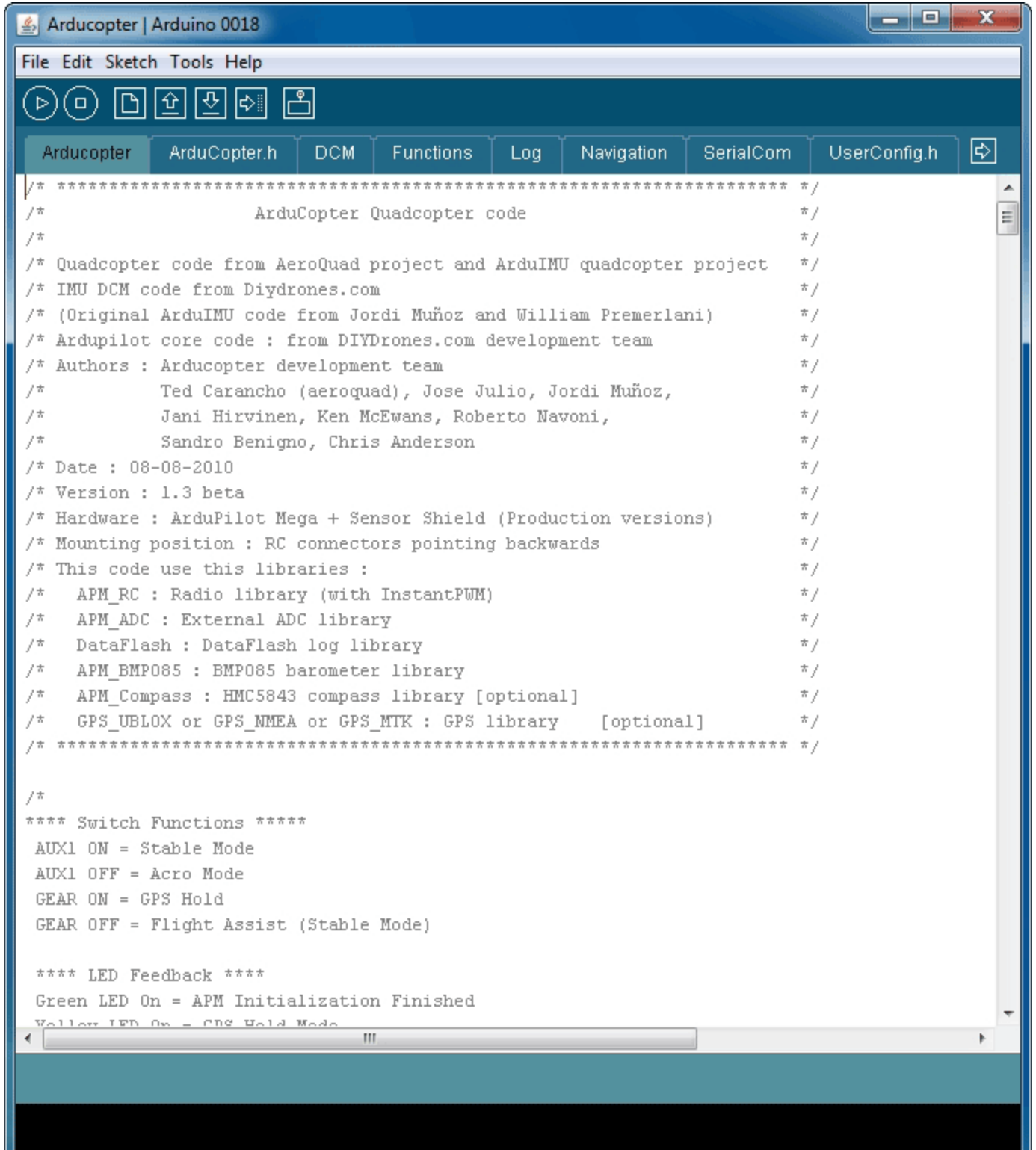
Installation instructions for the libraries:

ArduCopter Configurator: Download here

Important Install notes

PLEASE READ THE WARNINGS AT THE END OF THIS FILE!!!

Recommended read: The ArduCopter Wiki "Programming ArduPilot Mega"


Arducopter | Arduino 0018
File Edit Sketch Tools Help
Arducopter
ArduCopter.h
DCM
Functions
Log
Navigation
SerialCom
UserConfig.h
/* ** */
/*   ArduCopter Quadcopter code   */
/*   */
/* Quadcopter code from AeroQuad project and ArduIMU quadcopter project   */
/* IMU DCM code from Diydrones.com   */
/* (Original ArduIMU code from Jordi Muñoz and William Premerlani)   */
/* Ardupilot core code : from DIYDrones.com development team   */
/* Authors : Arducopter development team   */
/*   Ted Carancho (aeroquad), Jose Julio, Jordi Muñoz,   */
/*   Jani Hirvinen, Ken McEwans, Roberto Navoni,   */
/*   Sandro Benigno, Chris Anderson   */
/* Date : 08-08-2010   */
/* Version : 1.3 beta   */
/* Hardware : ArduPilot Mega + Sensor Shield (Production versions)   */
/* Mounting position : RC connectors pointing backwards   */
/* This code use this libraries :   */
/*   APM_RC : Radio library (with InstantPWM)   */
/*   APM_ADC : External ADC library   */
/*   DataFlash : DataFlash log library   */
/*   APM_BMP085 : BMP085 barometer library   */
/*   APM_Compass : HMC5843 compass library [optional]   */
/*   GPS_UBLOX or GPS_NMEA or GPS_MTK : GPS library   [optional]   */
/* ** */
/*
**** Switch Functions *****
AUX1 ON = Stable Mode
AUX1 OFF = Acro Mode
GEAR ON = GPS Hold
GEAR OFF = Flight Assist (Stable Mode)
**** LED Feedback ****
Green LED On = APM Initialization Finished

1. Copy the arduino-0020 folder onto your harddisk.

Follow the setup instructions from the Arduino-0020-README.txt

Some necessary basic libraries are already included.

In case of problems, consult the Arduino Trouble Shooting

2. Install the USB Serial Drivers if necessary. When you plug in the Megashield (oilpan), you might be asked for the drivers (not needed on Win7 and Mac OS 10.5x).

Point the installer to the "USB Serial Drivers" folder.

3. User Definable Modules
In the ArduCopter RC1 code, there are a series of user definable modules near the top of the Arducopter_alpha_RC1 file. By default, the only one not commented out is the `define IsAM`. Depending on what you have connected, uncomment the appropriate lines by deleting two slashes (`//`). *Note that the Magnetometer also needs to be enabled in the configurator.* Scroll down about 15 lines and you'll see two lines of code defining what flight mode your ArduCopter is in. New with the RC1 release is a code-based frame orientation. Your APM+IMU should be installed in the (+) configuration; the software handles the rotation of the sensors for flying in the (x) configuration. If you are flying in the (+) configuration, you do not need to change anything here. If flying in the (x) configuration, make sure the lines are:

```
//#define FLIGHT_MODE_+ // Traditional "one arm as nose" frame configuration
#define FLIGHT_MODE_X // Frame orientation 45 deg to CCW, nose between two arms (front and right)
```

4. Upload the ArduCopter firmware to the APM.

- plug the board into the USB

- start arduino.exe from the arduino-0020 folder

- go to -> File -> Preferences and set the location for your sketches. Choose the folder that resides above the "ArduCopter-firmware" folder

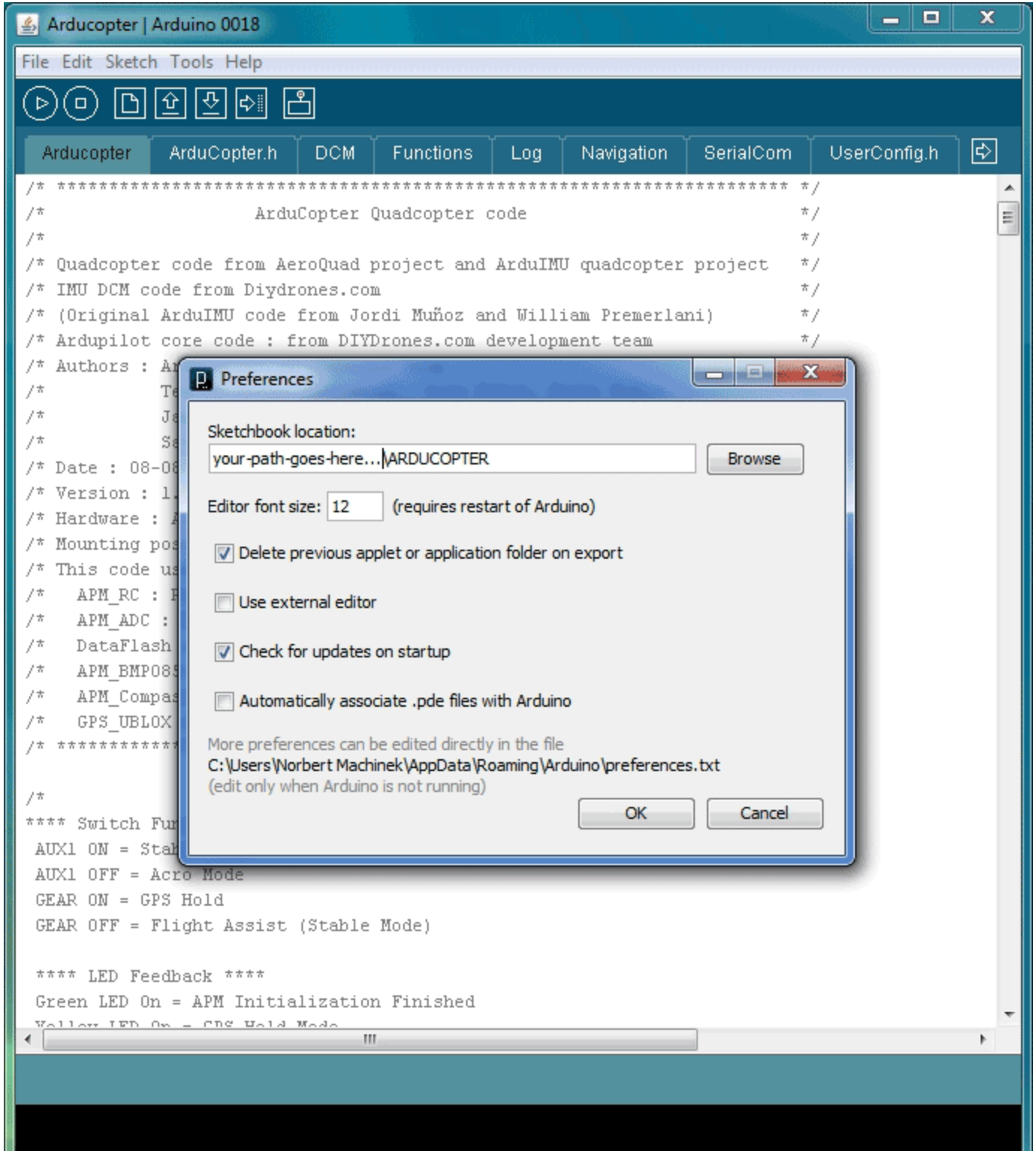
- go to -> Tools -> Board and choose the "Arduino Mega" as your board

- go to -> Tools -> Serial Port and choose the port that was previously set up after you installed the USB Serial Driver (usually COM3)

- go to -> Tools -> Serial Monitor and set the port speed to 115200 baud

- go to -> File -> Sketchbook -> ArduCopter

- click on the "Upload"-button. Then the arduino software compiles the firmware and uploads it afterwards. You should get the message "Done Uploading"


Arducopter | Arduino 0018
File Edit Sketch Tools Help
Arducopter
ArduCopter.h
DCM
Functions
Log
Navigation
SerialCom
UserConfig.h
/* ArduCopter Quadcopter code */
/* Quadcopter code from AeroQuad project and ArduIMU quadcopter project */
/* IMU DCM code from Diydrones.com */
/* (Original ArduIMU code from Jordi Muñoz and William Premerlani) */
/* Ardupilot core code : from DIYDrones.com development team */
Preferences
Sketchbook location:
your-path-goes-here...\ARDUCOPTER
Browse
Editor font size: 12
(requires restart of Arduino)
Delete previous applet or application folder on export
Use external editor
Check for updates on startup
Automatically associate .pde files with Arduino
More preferences can be edited directly in the file
C:\Users\Norbert Machinek\AppData\Roaming\Arduino\preferences.txt
(edit only when Arduino is not running)
OK
Cancel
AUX1 ON = Stab
AUX1 OFF = Acro Mode
GEAR ON = GPS Hold
GEAR OFF = Flight Assist (Stable Mode)
**** LED Feedback ****
Green LED On = APM Initialization Finished

5. Install the ArduCopter Configurator by starting the setup.exe

After installation start the ArduCopter Configurator software

- choose COM3 als serial port (if it's the desired port - see above)

- type 115200 baud for the "Baud Rate", set the timeout to 30 sec

- click on Connect

6. Configure your copter

- go to "Initial Setup" and click on "Initialize EEPROM"

- in the status bar it should say "Initializing done"

- Go to "Transmitter Adjustment" and check if your commands are displayed correctly

A NOTE OF WARNING! The firm- and software is at **alpha stage**. That means there are still several bugs in the code that have to be ironed out in the next few weeks.

ANOTHER IMPORTANT WARNING!!! When you plug in the USB make sure you don't have your ESCs and motors connected. Otherwise you will blow your outputs! Reference

Quad_ESC

ESC Calibration

CALIBRATING ESCS and getting the MOTORS working

Usually all calibration will be done through the configurator. For now you can only initialize the EEPROM and calibrate your tx/rx within the version 1.2

So you'll have to do the calibration manually: Plug each ESC attached to the motors into the throttle channel of your rx. Switch on your tx and go to full throttle. Plug in the lipo for the motors. You'll here 2 series of beeps confirming the correct calibration. Then pull your throttle back to zero and wait for the beep code. When done unplug the lipo, switch the tx off, plug the lipos again and switch on the tx again.

Do this for each ESC you want to calibrate.

After the calibration is done you can plug all cables in place and try to arm the motors by pushing the throttle stick to the lower right position. To disarm the motors push it to the lower left.

Note: There's a new built in delay when arming/disarming the motors. Depending on the version you use it'll take up to 2,5 sec. for the motors to react.

ArduCopter Quad uses 4 motors in total and they are named as:

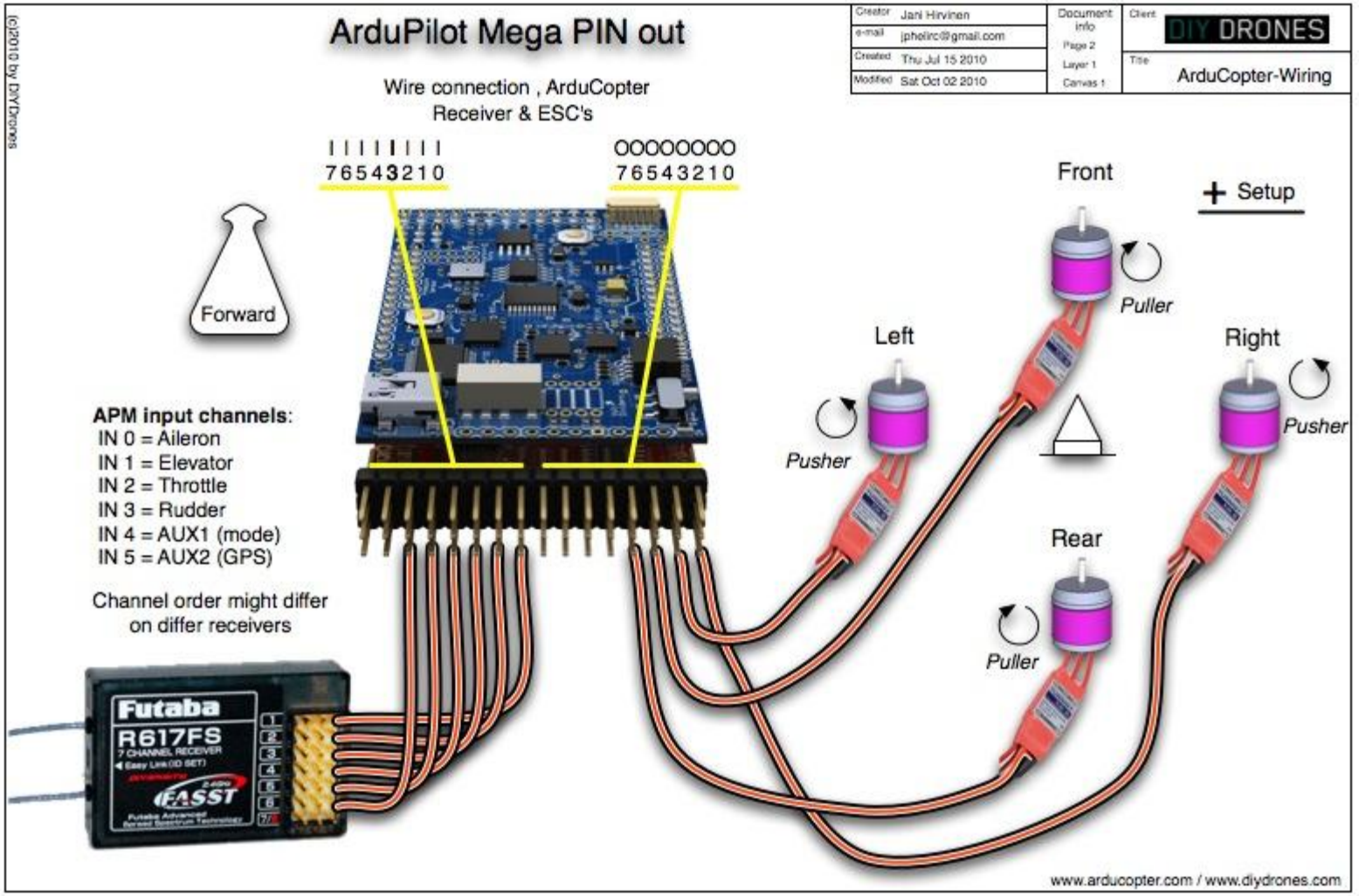
+ Configuration

- OUT0 = Right motor (CCW)
- OUT1 = Left motor (CCW)
- OUT2 = Front motor (CW)
- OUT3 = Rear motor (CW)

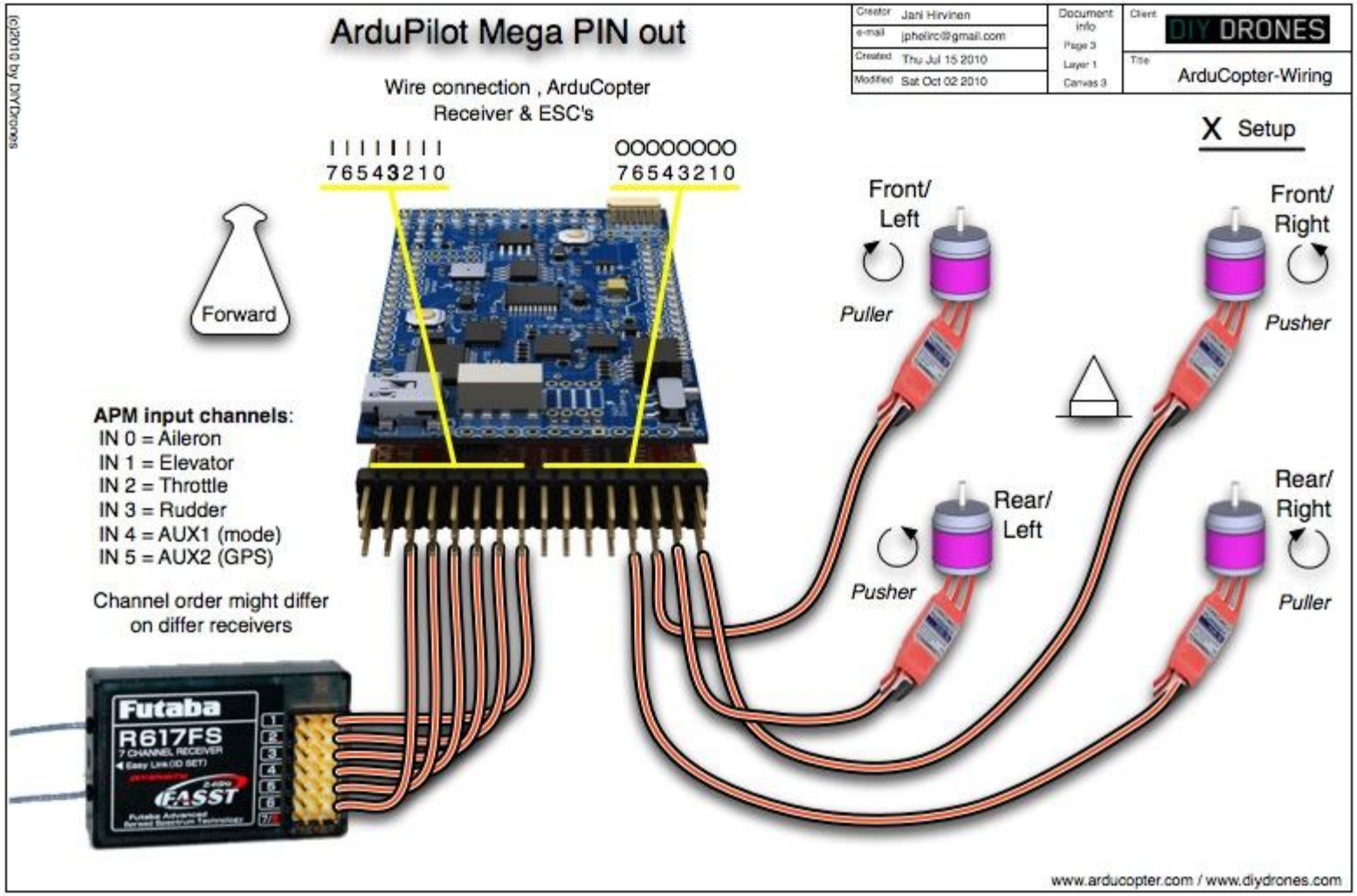
X Configuration

- OUT0 = Front/Right motor (CCW)
- OUT1 = Back/Left motor (CCW)
- OUT2 = Front/Left motor (CW)
- OUT3 = Back/Right motor (CW)

CCW = Counter Clock Wise rotation (eg backwards) CS = Clock Wise (eg forward like clock does)



PlusConfiguration



X-Configuration

Comment by project member analoguedevices, Sep 18, 2010

On the "X" configuration diagram, "front", "right", "rear" and "left" unclear. They should be marked "right front", "right rear", "left front" and "left rear".

Also should the pusher props go on the clockwise or counterclockwise motors?

Comment by shaice90, Sep 26 (6 days ago)

Hi, If I want use 3 motor (tricopter), what the configuration look like?

Comment by project member jphelirc, Sep 26 (6 days ago)

Currently ArduCopter does not support Tricopter setups. Tri also need special hardware like tail motor tilt system, servos to control it etc.

Hmm yep picture could be a bit more clear on this issue. Will be fixed now that we also have working software based orientation system on code.

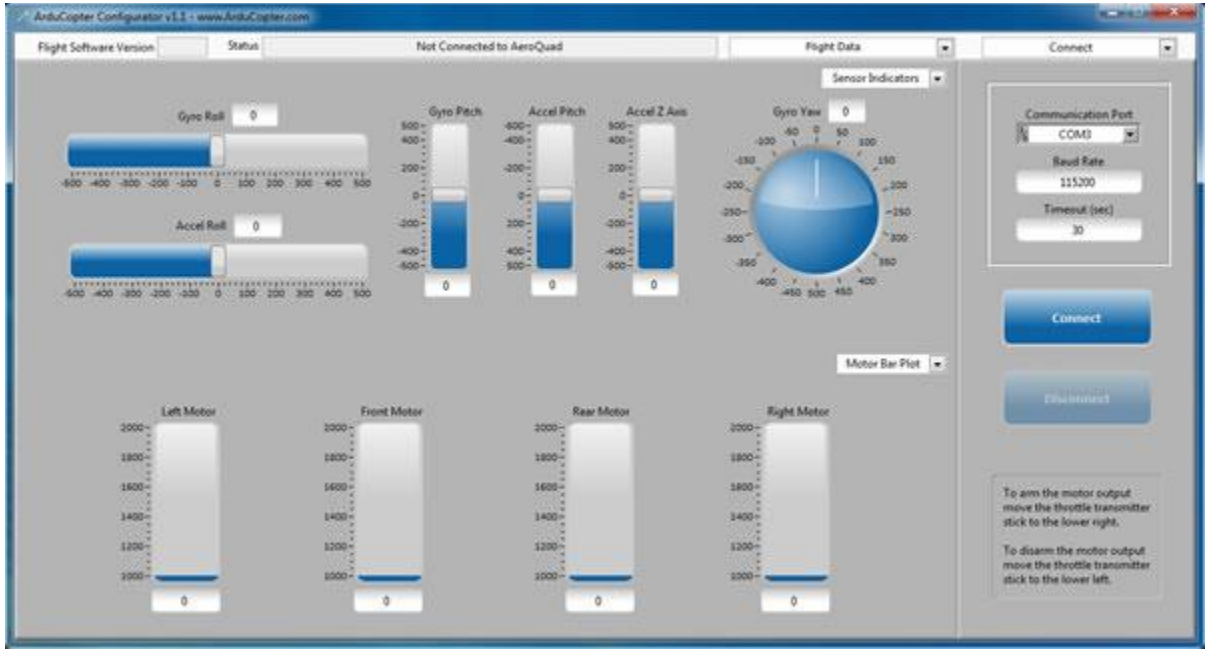
Quad_ConfiguratorTool

Configurator Tool

What is it? The purpose of this tool is to allow the user to setup the ArduCopter before it's first flight and to quickly adjust settings for desired flight characteristics. The user will be able to graphically observe correct operation of the sensors, transmitter commands and motor control of the quadrocopter. Additionally there are user programmable values such as PID control loop values and the Transmitter Factor that can be adjusted and stored to the ArduCopter's EEPROM.

If you haven't already downloaded it, it can be found here. Remember that you must also load the LabVIEW drivers if you have not already done so.

It is highly recommended to first checkout the ArduCopter without motors connected (or powered on) with the Configurator.



The accel values will not be correct until you save your configuration to EEPROM, which will normalize them.

Once you have configured the sensors, you can reconnect the motors and test them, ensuring that the props are turning in the correct directions. If a prop is turning the wrong way, just swap two of the three wires going to that motor.

Quad_Radio

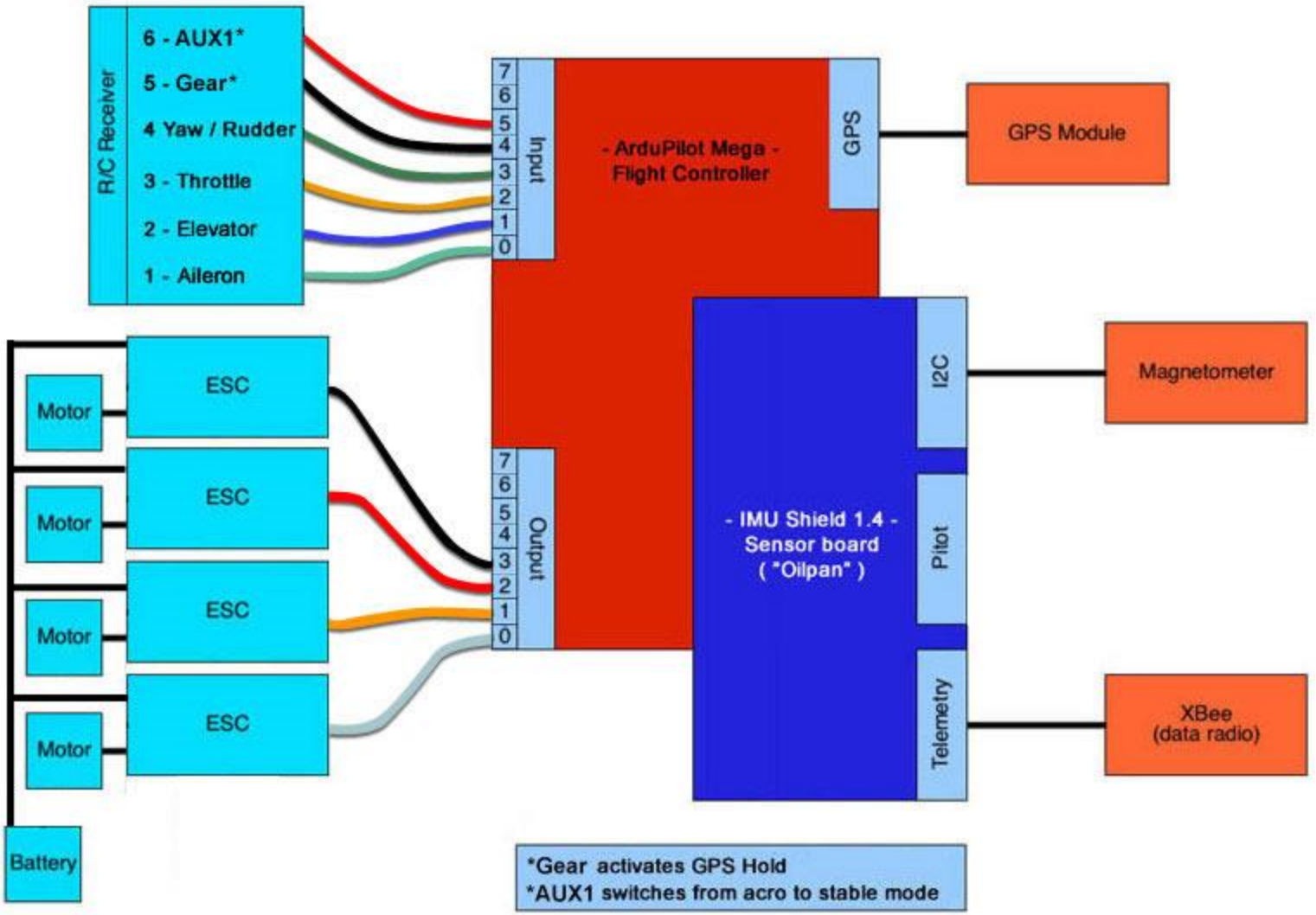
Tuning Radio

Connecting your RC equipment

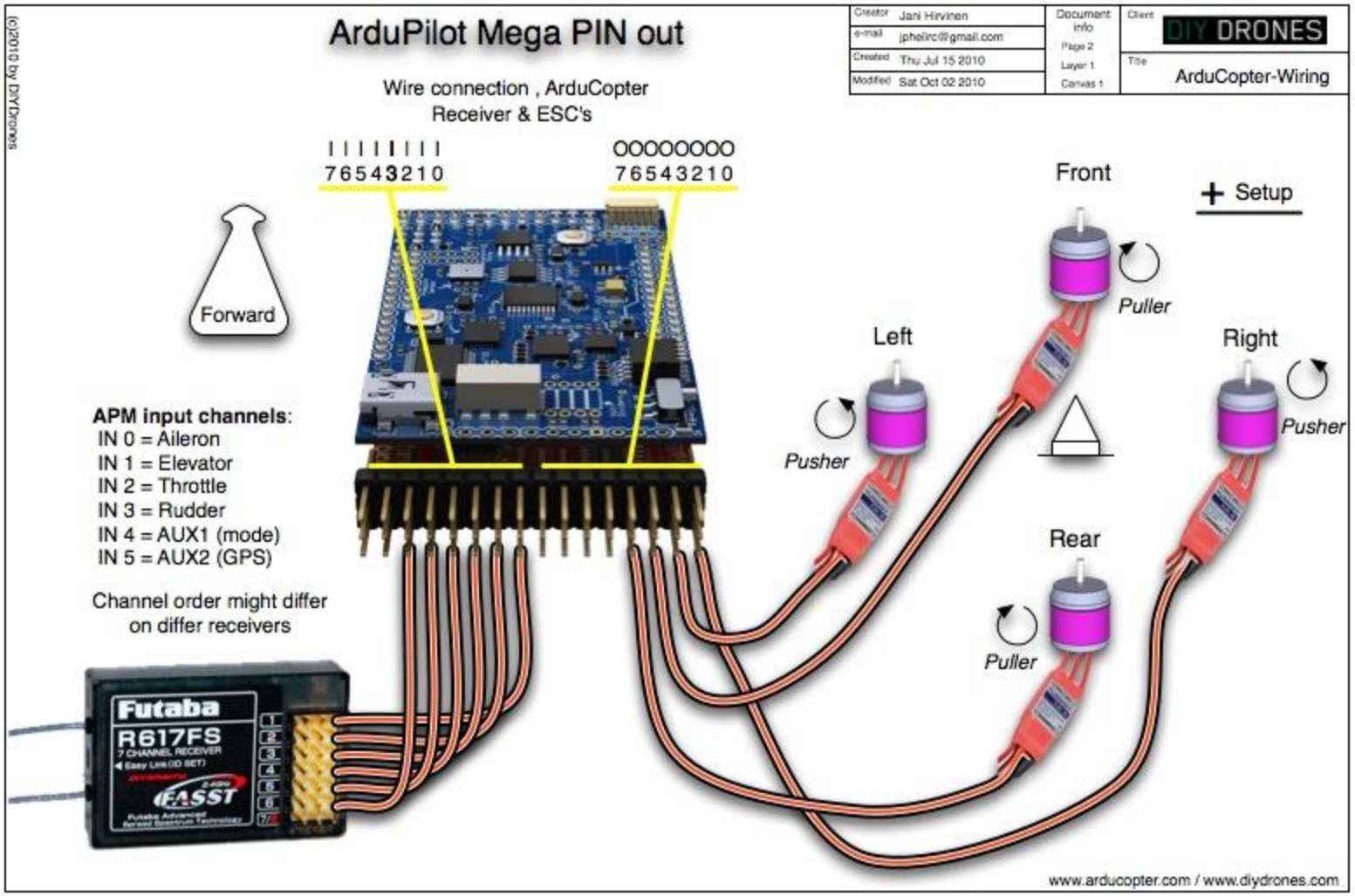
What you'll need:

- At least a 5-channel RC unit. 7 channels or more is highly recommended
- Female-to female cables for each channel you'll be using. We use these short ones to minimize cable clutter.
- A power source. For electric aircraft, this is usually the ESC. For gas/nitro powered planes, your radio will need its own battery. ArduPilot Mega gets its power from the RC system.

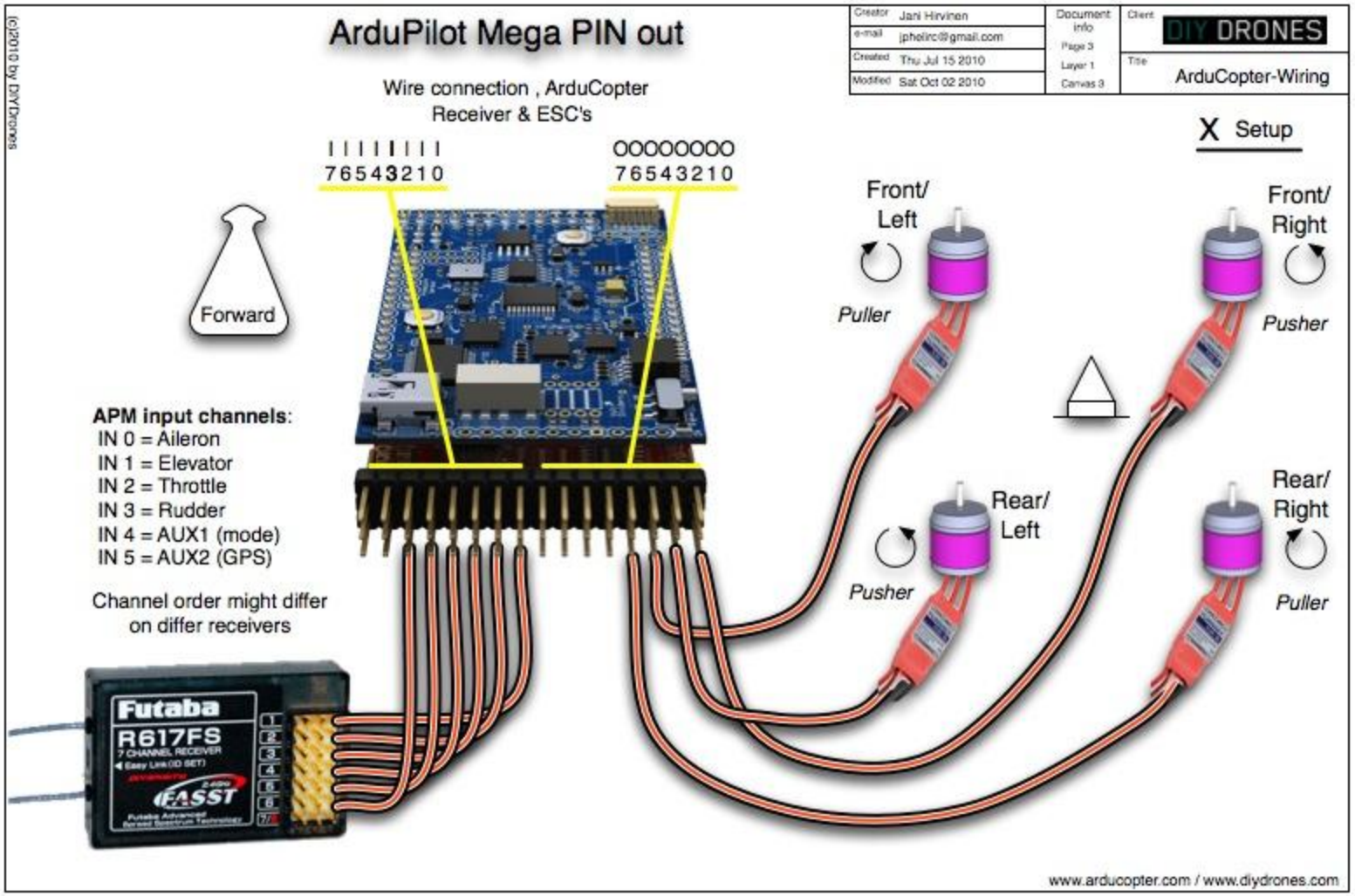
Instructions



Wiring diagram for + config

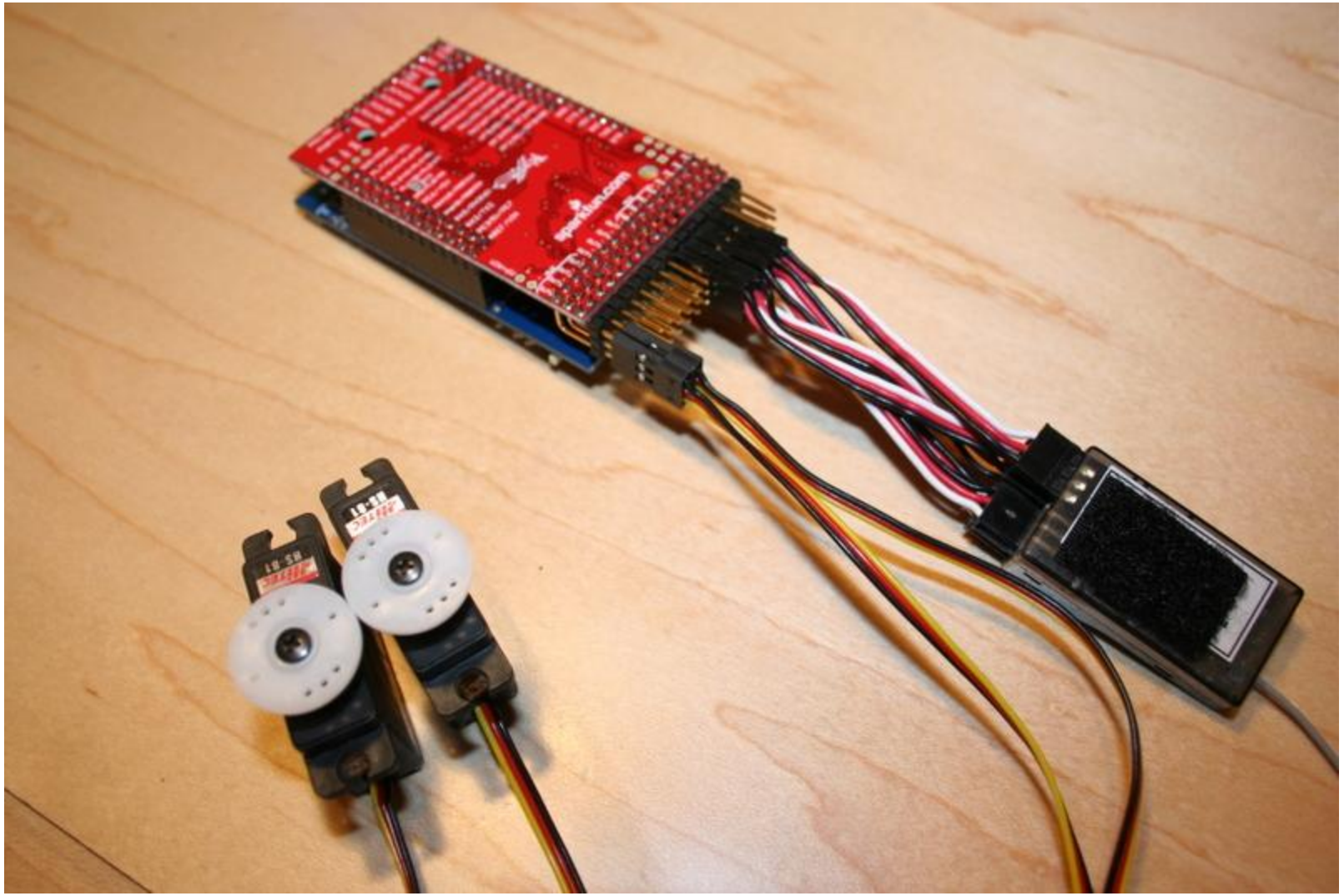


Wiring diagram for x config



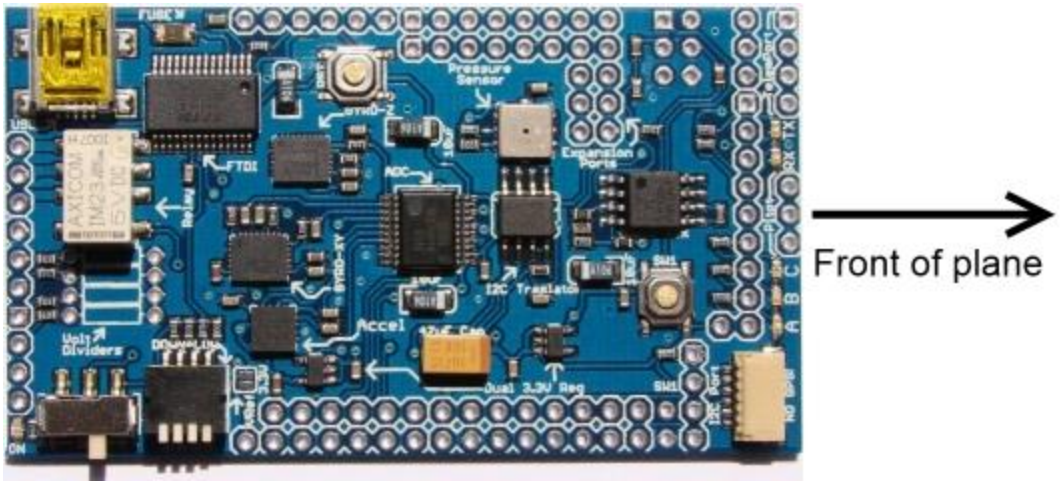
Connect your RC receiver to ArduPilot Mega (APM) with female-to-female cables. Each channel that you want APM to control should be connected to an Input on the APM board. When you follow the right-angle connector, you'll see that the signal pins are those that connect to the board furthest from the edge; the ground pins are closest to the edge. Usually signal is white or yellow; ground is black.

Plug your servos and other devices you want APM to control into the matching Output connectors. In this picture, I've connected 5 channels in and two servos out.



The first four channels are controlled by APM's multiplexer. To use this feature you should connect the channel on your RC receiver that you want to use to select the autopilot mode to the last channel input (marked as 7 on the board) on ArduPilot Mega.

When you place APM in your aircraft, it is very important that it face the right way. The GPS connector should face forward, and the servo cables face back. Like this (note: there's a little arrow on the bottom of the shield that point to the front, too, in case you need a reminder at the field):



Flight modes

You can set your RC system up so that you can choose between either three modes or six modes using the mode selection channel. (The available modes are described here.) Selecting between three modes is easily accomplished if your transmitter has a three position switch. Generally you don't have to do anything and the RC system's default settings are fine.

Note: The toggle switch settings are the reverse of what they were on the original ArduPilot. So if you manual setting was the toggle being the position closest to you, now it's furthest away. If you prefer it the way it originally was, just reverse that channel on your RC transmitter.

If you want to have six modes, you'll probably have to configure your RC transmitter to do this. That's usually accomplished by mixing a two position switch and a three position switch on your transmitter. Set your switch(es) to (ideally) produce PWM pulse widths of 1165, 1425, and 1815 microseconds or 1165, 1295, 1425, 1555, 1685, and 1815 milliseconds. (You can also do this with an analog dial, if you have one, but it's hard to reliably turn it to just the right position for six distinct settings)

How can you know what the PWM pulse widths are? Use the radio test debug mode.

Failsafe function

When the PWM pulse width on the last channel exceeds 1750 microseconds the multiplexor switches to manual mode so that the lower 4 channels are controlled by the RC inputs rather than by the autopilot. That means that even if the APM code crashes, you will always be able to switch them back into manual control. You can override this feature by using some other channel for your mode selection and leaving the highest channel input unused. In this case you can still have a manual mode, but it will be dependent on software. There is a parameter in the autopilot header file that you use to tell the autopilot which channel input you want to use for mode selection. You can read more about APM's failsafe functions here.

Also, you can set up your system to provide a failsafe in case of radio signal loss. The first method is using a receiver which allows you to set the default PWM values that it will output if the rf signal is lost. Spektrum receivers, for example, have this feature. In this case set up the receiver so that the channel you are using for mode selection outputs a value corresponding to a switch position you will use for RTL (or AUTO). If you are using a receiver for which you can only program

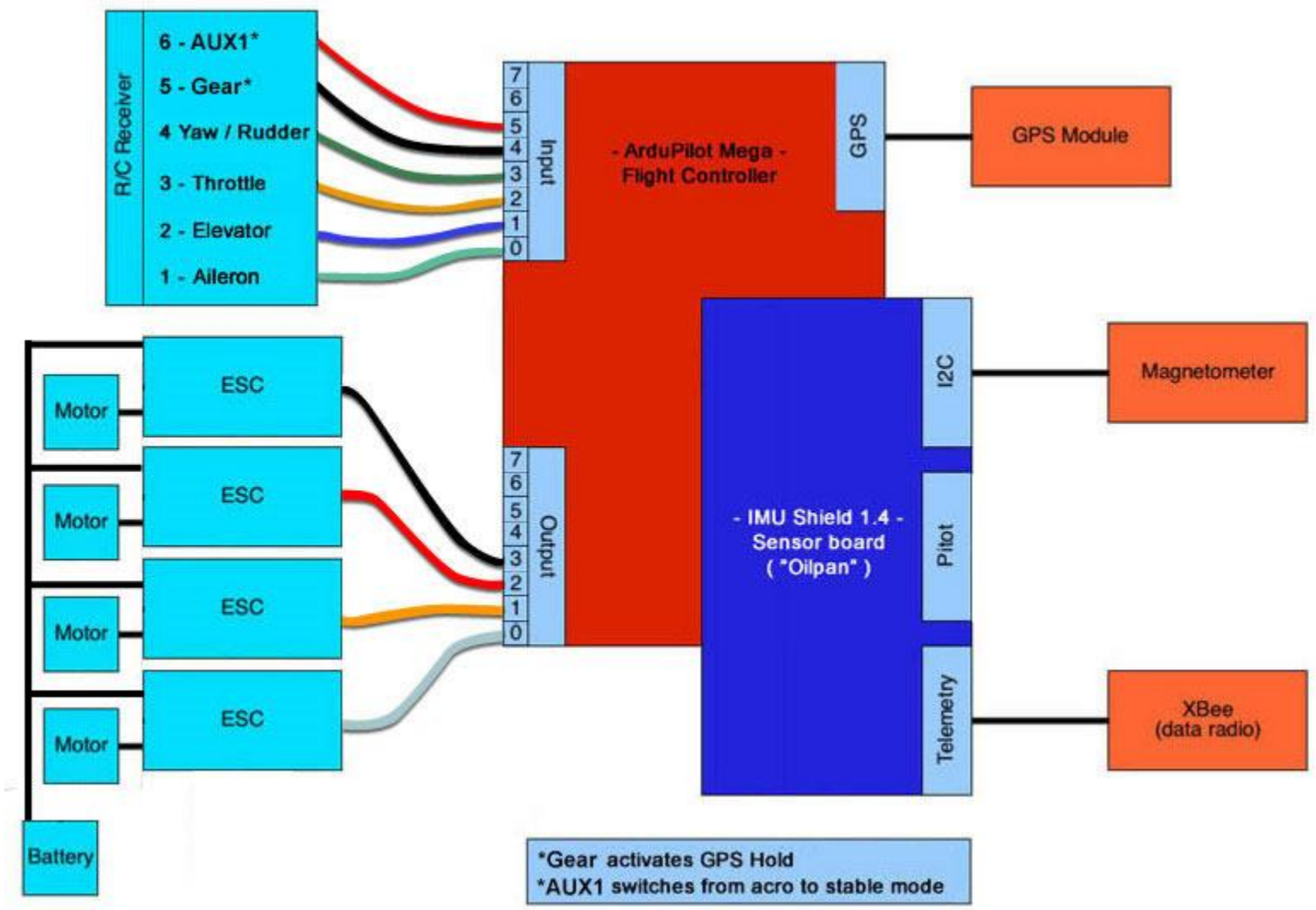
the throttle channel for radio loss failsafe, program it to output a value below 1000 milliseconds and enable the throttle failsafe in the header file.

Comment by project member analoguedevices, Sep 18, 2010

This should be edited to delete "plane" references

Comment by emile.castelnuovo, Sep 20, 2010

The connection diagram is not correct... The input and motor connections are inverted. Please update as many would make mistakes. Use this image taken from r



Quad_Serial_Commands

Serial Commands

Note: We make a best attempt at keeping all the documentation up to date when possible. If you are not getting expected command/telemetry responses please look through SerialCom.pde in the ArduCopter source code to make sure the command is coded as expected.

(A) Send roll and pitch gyro PID values for Stable mode

Send an 'A' followed by numeric values to transmit user defined roll and pitch PID values for gyro stabilized flight control. Each value is separated by a semi-colon in the form:

```
A[KP Quad Roll];[KI Quad Roll];[KP RATE ROLL];[KP Quad Pitch];[KI Quad Pitch];[KP RATE PITCH];[KP Quad Yaw];[KI Quad Yaw];[KP Rate Yaw];[KP Rate];[Magneto]
```

(B) Read roll and pitch gyro PID values

Send a 'B' to receive user defined roll and pitch PID values for gyro stabilized flight control. The ArduCopter will then send a string response in the form:

```
[KP Quad Roll];[KI Quad Roll];[KP RATE ROLL];[KP Quad Pitch];[KI Quad Pitch];[KP RATE PITCH];[KP Quad Yaw];[KI Quad Yaw];[KP Rate Yaw];[KP Rate];[Magneto]
```

When you verify that the values you sent match the values in the ArduCopter, send a 'W' to write these values to the EEPROM. It is recommended to set all desired user defined values first, then send a 'W' to write all values to EEPROM at once. Example string received: 2.5,0,-4,3.1,0,-3.8,0

(C) GPS PID Values

Send an 'C' followed by numeric values to transmit user defined roll and pitch PID values for gyro stabilized flight control. Each value is separated by a semi-colon in the form:

```
C[P GPS ROLL];[I GPS ROLL];[D GPS ROLL];[P GPS PITCH];[I GPS PITCH];[D GPS PITCH];[GPS MAX ANGLE];[GEOG Correction factor]
```

(D) Read GPS PID values

Send a 'D' to receive user defined GPS PID values. The ArduCopter will then send a string response in the form:

```
[P GPS ROLL];[I GPS ROLL];[D GPS ROLL];[P GPS PITCH];[I GPS PITCH];[D GPS PITCH];[GPS MAX ANGLE];[GEOG Correction factor]
```

When you verify that the values you sent match the values in the ArduCopter, send a 'W' to write these values to the EEPROM. It is recommended to set all desired user defined values first, then send a 'W' to write all values to EEPROM at once.

Example string received:

10,0,0

(K) Spare
(L) Spare
(N) Spare
(M) Receive debug Motor commands

Example string received; 1040,1040,1040,1040,0

(Q) Read Sensor Data

Send a 'Q' to initiate reading back of gyro, accel, and calculated absolute roll/pitch values. After a 'Q' is transmitted, the following comma delimited string ending with a carriage return and line feed will be continuously sent from the ArduCopter:

```
[Roll Gyro Rate],[Pitch Gyro Rate],[Yaw Gyro Rate],[Roll Accel],[Pitch Accel],[Z Accel],[Roll Angle],[Pitch Angle]\r\n
```

Roll Gyro Rate = measured ADC value of roll gyro axis centered around zero Pitch Gyro Rate = measured ADC value of pitch gyro axis centered around zero Yaw Gyro Rate = measured ADC value of yaw gyro axis centered around zero Roll Accelerometer Position = measured ADC value of roll accel axis centered around zero Pitch Accelerometer Position = measured ADC value of pitch accel axis centered around zero Z Accelerometer Position = measured ADC value of Z-axis accel axis centered around zero Roll Angle = absolute roll angle experienced by the quad as calculated by the 2nd order complementary filter Pitch Angle = absolute pitch angle experienced by the quad as calculated by the 2nd order complementary filter

(R) Read Raw Sensor Values

Send an 'R' to initiate reading back of all sensor values. After an 'R' is transmitted, the following comma delimited string ending with a carriage return and line feed will be continously sent from the ArduCopter:

Example string:

(S) Read All Flight Values

Send an 'S' to initiate reading back of all flight values. After an 'S' is transmitted, the following comma delimited string ending with a carriage return and line feed will be sent continuously from the ArduCopter:

```
[Loop Time],[Roll Gyro Rate],[Pitch Gyro Rate],[Yaw Gyro Rate],[Throttle Output],[Roll PID Output],[Pitch PID
Output],[Yaw PID Output],[Front Motor Command],[Rear Motor Command],[Right Motor Command],[Left Motor Command]\r\n
```

Loop Time = time in milliseconds it takes the ArduCopter to execute one iteration in code

Roll Gyro Rate = measured ADC value of roll gyro axis centered around zero

Pitch Gyro Rate = measured ADC value of pitch gyro axis centered around zero

Yaw Gyro Rate = measured ADC value of yaw gyro axis centered around zero

Throttle Output = measured PWM output of throttle (ranges from 1000-2000)

Roll PID Output = PID output for roll axis (ranges from 1000-2000)

Pitch PID Output = PID output for pitch axis (ranges from 1000-2000)

Yaw PID Output = PID output for yaw axis (ranges from 1000-2000)

Front Motor Command = PWM output sent to front motor (ranges from 1000-2000)

Rear Motor Command = PWM output sent to rear motor (ranges from 1000-2000)

Right Motor Command = PWM output sent to right motor (ranges from 1000-2000)

Left Motor Command = PWM output sent to left motor (ranges from 1000-2000) Example string that can be transmitted:

2,-10,3,-2,1011,1012,1002,1000,1001,1003,1002,1004

The above example string specifies:

Loop Time = 2

Roll Gyro Rate = -10

Pitch Gyro Rate = 3

Yaw Gyro Rate = -2

Throttle Output = 1011

Roll PID Output = 1012

Pitch PID Output = 1002

Yaw PID Output 1000

Front Motor Command = 1001

Rear Motor Command 1003

Right Motor Command = 1002

Left Motor Command = 1004

(T) Spare
(U) Read Receiver Values

Send an 'U' to initiate reading back of R/C receiver values. After an 'U' is transmitted, the following comma delimited string ending with a carriage return and line feed will be continuously sent from the ArduCopter:

```
[Loop Time],[Throttle],[Roll],[Pitch],[Yaw],[Gear],[Aux]
```

Loop Time = time in milliseconds it takes the ArduCopter to execute one iteration in code

Roll = PWM output from roll channel (ranges from 1000-2000)

Pitch = PWM output from pitch channel (ranges from 1000-2000)

Yaw = PWM output from yaw channel (ranges from 1000-2000)

Throttle = PWM output throttle channel (ranges from 1000-2000)

Gear = PWM output from gear channel (ranges from 1000-2000)

Aux = PWM output from auxilliary channel (ranges from 1000-2000)

Example string that can be transmitted:

1,1403,1620,1523,1501,1900,1950

The above example command specifies receiver values of:

Loop Time = 1

Roll = 1403

Pitch = 1620

Yaw = 1523

Throttle = 1501

Gear = 1900

Aux = 1950

(W) Write user defined values to EEPROM
(X) Stop sending continuous data

Quad_Testing

Testing

Pre-Flight Checkout

Before your first flight, it is very important that you follow this pre-flight checkout! Use the Configurator plots to visually view the ArduCopter response graphically.

The propellers should be taken off to prevent injury. Be careful! Spinning propellers can cause serious injury!!!

- Arm the motor output by moving the throttle stick to the lower right. Increase the throttle to 50%. (No need to apply Lipo power to the ArduCopter)
- By hand, roll the ArduCopter to the left. The left motor command should increase. The right motor command should decrease.
- Roll the ArduCopter to the right. The right motor command should increase. The left motor command should decrease.
- Pitch the ArduCopter down (the front motor should be lower in position than the rear motor). The front motor command should increase. The rear motor command should decrease.
- Pitch the ArduCopter up (the front motor should be higher than the rear motor). The rear motor command should increase. The front motor command should decrease.
- Rotate the ArduCopter clockwise. The front and rear motor commands should increase (assuming the motors are wired to rotate in the clockwise direction).
- Rotate the ArduCopter counter-clockwise. The left and right motor commands should increase in value (assuming the motors are wired to rotate in the counter-clockwise direction).

- Using the transmitter, move the roll stick to the left. The right motor command should increase. The left motor command should decrease.
- Move the roll stick to the right. The left motor command should increase. The right motor command should decrease.
- Move the pitch stick forward. The rear motor command should increase. The front motor command should decrease.
- Move the pitch stick back. The front motor command should increase. The rear motor command should decrease.
- Move the yaw stick to the left. The front and rear motor commands should increase.
- Move the yaw stick to the right. The left and right motor commands should increase.

Quad_ModeSwitch

- # Mode Switch
- By using the mode switching abilities of the ArduCopter you can change the flight behavior of your drone either by programming fixed values or simply by additional channels of your Rx/Tx. I.e. you could use
- - AUX1 (channel 4) for enabeling the GPS Position Hold mode
- - AUX2 (channel 5) to switch between acrobatic and stable mode

Quad_FlightTips

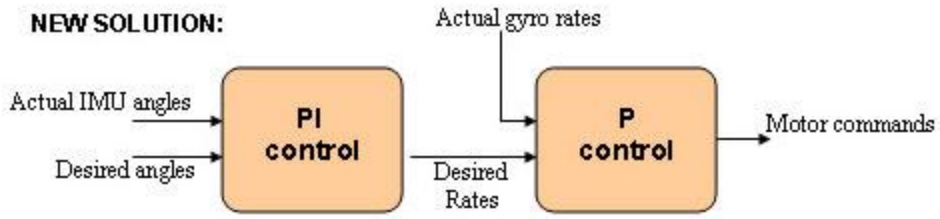
Flight Tuning tips

Acrobatic Mode

Acrobatic mode is a `PID` rate control. The `I` term is not needed (usually 0). `D` should be a low value (start with `D=0`). The `P` term is the main parameter. You should start by increasing `P` until the ArduCopter starts to oscillate and then reduce the `P` parameter a little bit.

Stable Mode

On the current version of stable mode, there are two controls. One is an inner control that is a `P` rate control (*In the configurator, this is labeled as `D`. It will be changed to `P_rate`*), and the other is an outer control that is a `PI` absolute angle control.



This approach utilizing two controls instead of the normal `PID` has proved to be much better. Because of this, the `D`, or `P_rate`, parameter is important and shouldn't be zero.

Control Adjustment Guidelines

If you haven't adjusted your Acrobatic mode parameters, just use the default. The `P_rate` value in Stable mode should be a little lower than the `P` value in Acrobatic mode. *For example, a `P` rate in Acrobatic mode could be 1.9 and the `P_rate` in Stable mode could be 1.2.* If the `P_rate` is too high, you will see quick oscillations on the ArduCopter. If the values are too low, it will be unstable. The `I` term can be thought of like the trim on your radio; it is needed to account for differences in motors, ESCs, etc. Its value will be low, likely between 0.1 and 0.4. The `P` parameter is the main authority of the outer control. If its value is too low, the quad will be lazy and slow to respond to stick movements. If it is too high, instability will develop and the ArduCopter will oscillate.

Since control tuning is difficult, it is recommended that you start with the default values and only change one value at a time.

Quad_GroundStation

Setting up the Ground Station

We support both a software ground station running on your PC or, if you don't want to take your laptop the field, a small custom hardware ground station called ArduStation, which is available from the DIY Drones store.

These instructions will be for the software ground station, which runs on Windows. If you are using the hardware-based ArduStation instead of this software, please see the operating instruction for that here.

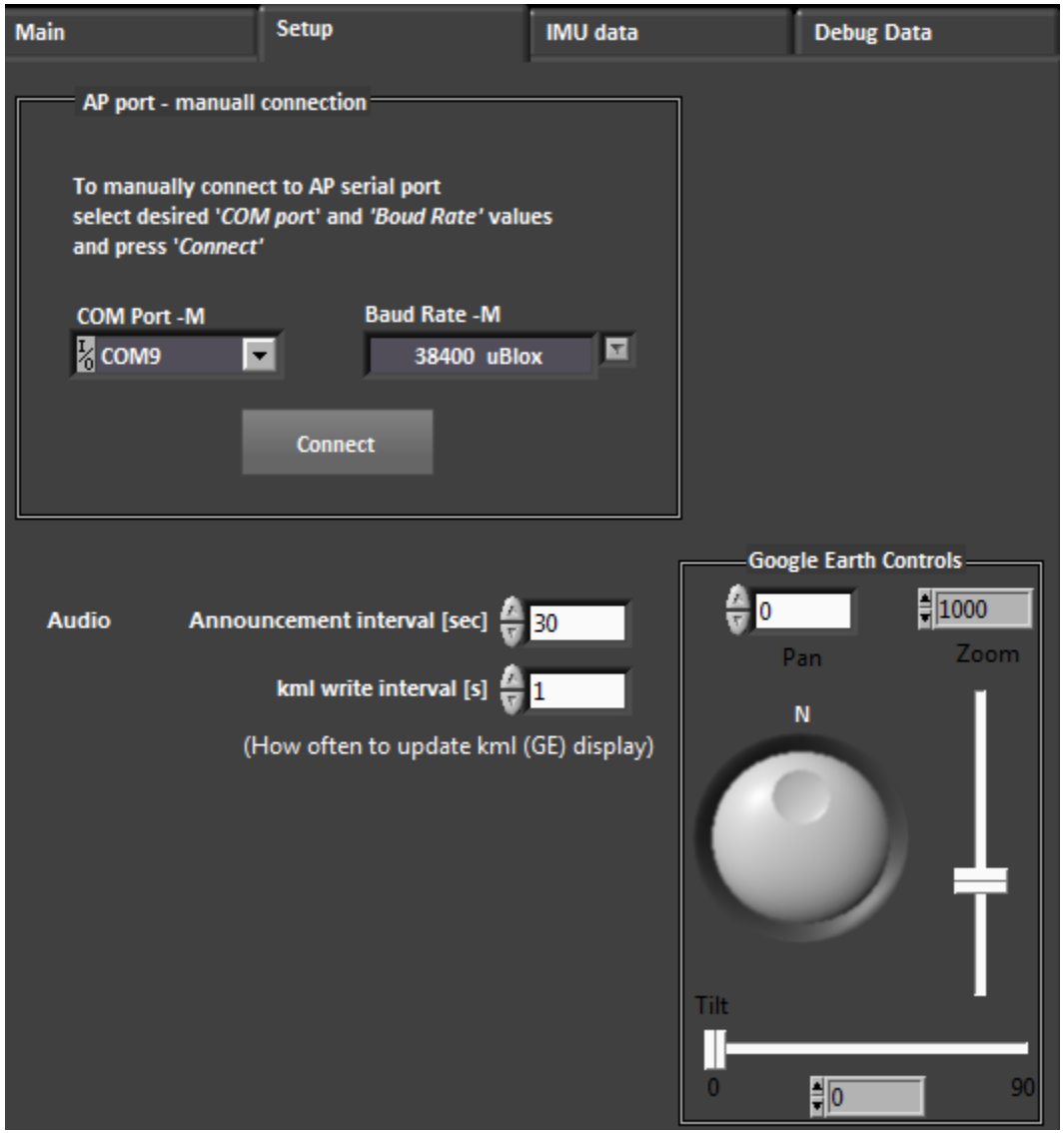
Once you have the Xbee telemetry set up, you can use the ground station. Download the executable code from the Google Code repository. Unpack the directory to your desktop.

You'll also need the LabView runtime engine and serial drivers. Download the runtime engine here and the drivers here. Install them. You'll also need to install Google Earth if you don't already have it installed on your PC.

Now you can run ArduStation.exe in the groundstation folder. After telling Windows to allow the software access to the Internet, you should see a screen like the following:

Ensure that your ground-based Xbee and the USB adapter board are connected and plugged into your PC. In the Setup tab, select the correct serial port in the Ground Station software and set the serial speed to match the speed you set your Xbee modules to (which are, in turn, determined by your choice of GPS

module). If you are using the EM406 GPS, please set it to 57600. If you are using the uBlox, set it to 38400 as shown below. The com port is assigned when you plug in your FTDI cable. Check your Windows Device Manager if you're not sure which Com port your cable was assigned to.



If ArduPilot is running, you should start to see the data displayed in real time on the ground station. You will not get GPS data until you have GPS lock.

If you turn on your speakers, you'll hear the ArduPilot status information spoken with voice synthesis. Very handy for the field, when you don't want to take your eyes off the plane!

Note: If you live in a European country that uses commas instead of decimals in numbers, you'll need to change your PC's settings. Go to the Windows Control Panel/Regional and Language Options. Click on Personalizing. In Symbol Decimal, select replace comma with decimal. Click on "Apply" and then restart your PC.

Additional instructions for this Ground Control Station can be found in the software's documentation, which you can find here.