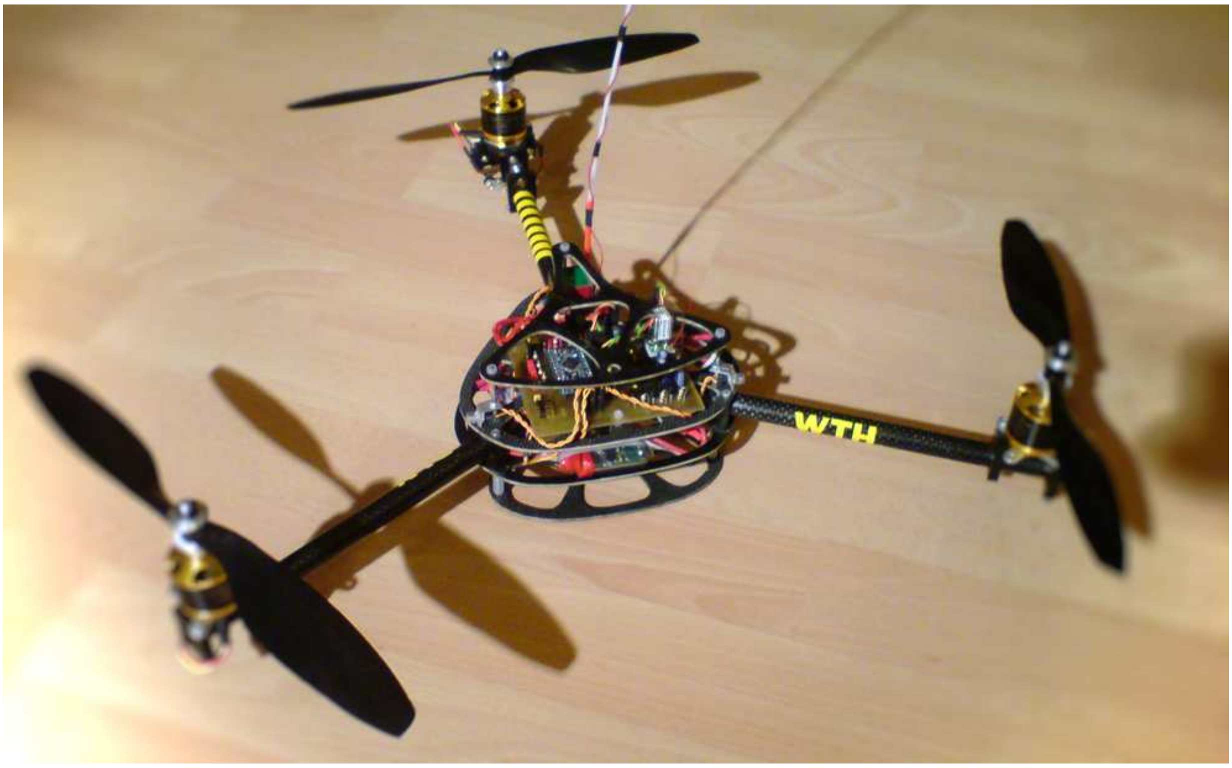
modified 02. Jun. 2010

Shrediquette

a tricopter by William Thielicke (2009)



Please note: If you never flew a radio controlled airplane or helicopter before, practice on a RC-simulator first! Even if the copter is auto-stable, there are always situations where you quickly have to react in the right way. You can only learn that by practicing (or by paying a lot of money for spare parts...). This is not a toy, although it is a lot of fun to play with it...

Use at your own risk in an environment and using safety precautions where you and other people will not get hurt.

I won't take any responsibility for accidents or damages that result from building and/ or flying a Shrediquette tricopter.

Introduction

A tricopter/ tri-rotor is a flying thing that uses three motors + propellers (arranged in a horizontal triangle) to stay airborne.

The speed of each of these motors can be controlled independently. Additionally, the motor at the rear can be rotated left and right via a servo. One propeller turns clockwise; two propellers turn counter-clockwise to minimize torque. Gyroscopes (measuring angular velocity) and accelerometers (measuring accelerations, like gravitation) are used to stabilize the inherently unstable system. The pilot can control the tricopter with a remote control. Height, nick, roll and yaw are controlled via 2 sticks on the remote control. A 3-position switch turns on the motors and selects control mode. Hence a 5-channel transmitter & receiver is needed.

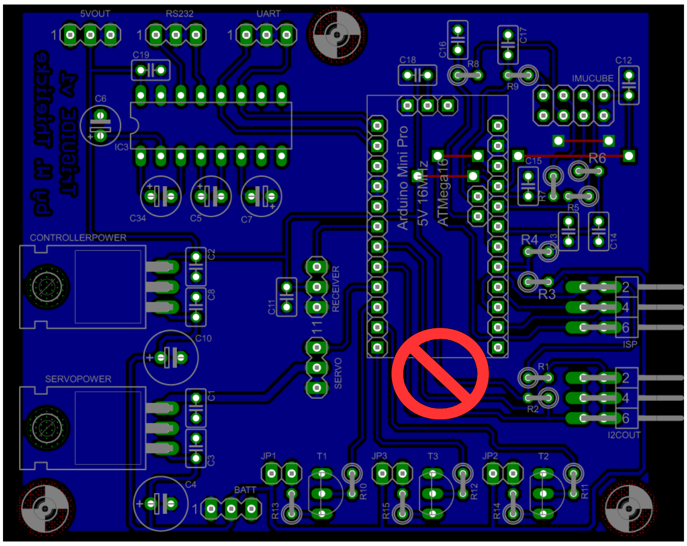
This specific tricopter uses two different control modes: Firstly, "hover mode", where the pilot controls the angle of the tricopter. Moving the sticks 10 degrees tilts the tricopter ~10 degrees. The copter will begin to accelerate in the desired direction. If you release the sticks, the copter will automatically level. But it will keep the velocity it had before (if we disregard aerodynamic drag). The tricopter can hover on its own in this flight mode if there is no wind and if there is some space.

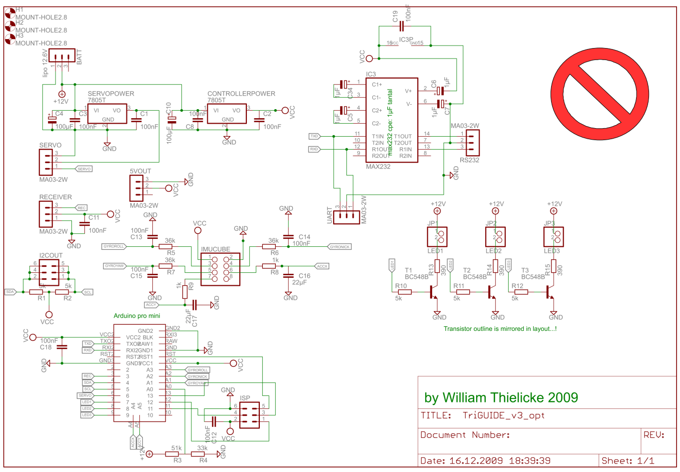
Secondly, "aerobatic mode", where the pilot controls the angular velocity of the copter. Moving the sticks 10 degrees will result in slow rotation of the tricopter (maybe 10 degrees per second). Moving the sticks further will result in faster rotation. The copter will always keep the angle it had if you release the sticks. This mode is very useful when doing aerobatics or flying around for fun. The tricopter flies like a regular RC helicopter in this mode. Yaw control is always "heading-hold" = angular velocity control.

The propellers that you are using for your tricopter should match your motors. They should also be as light as possible, as this allows for an efficient stabilization. Maxxprod has some very light propellers. APC Slowfly propellers are too heavy.

Electronics

First, you should put the main controller (TriGUIDE) of the tricopter together. **The Eagle files (schematics & layout) can be found in the blog.**





Screenshot, showing the layout and the circuit. Note: **This is just an example**, layout may change without notice. The most recent layout, source code, GUI and manual is always available in my blog at the top right corner!

The red lines in the layout are wire jumpers, don't forget to add them! Also don't forget to connect A4 and A5 with the PCB. These pins (on the Arduino pro mini) are pretty easy to forget. The following I²C addresses are used for the copter: Address1 (&H52): Motor at the rear, address2 (&H54): Left motor, address3 (&H56): right motor.

List of electronic components

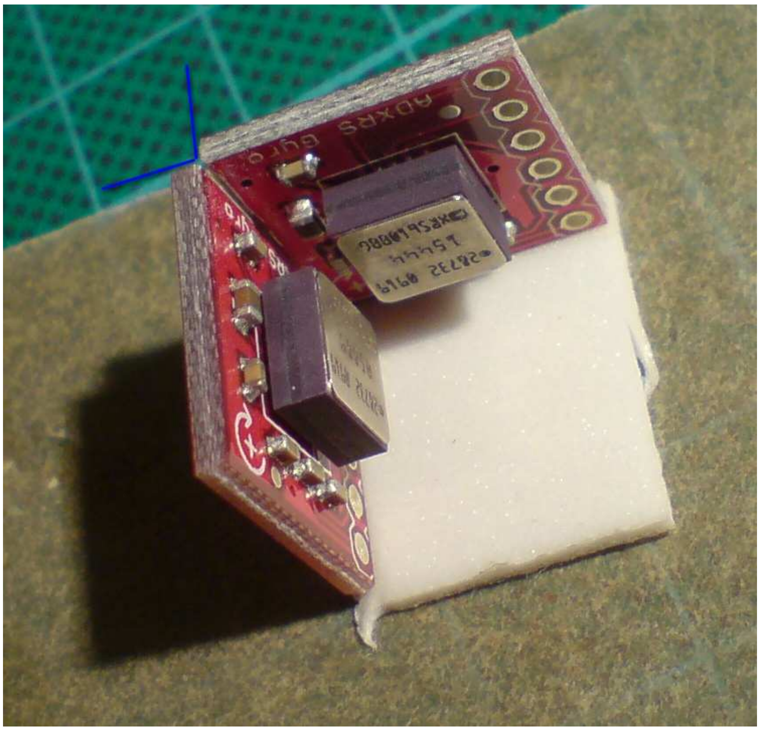
Amount	Value	Device	Used in
2	Pinheader	MA07-2W	I2COUT, ISP
1	Pinheader	PINHD-2X4	IMUCUBE
4	1µF	CPOL-EUE2.5-5	C5, C6, C7, C34
2	1µF	C-EU	C16, C17
5	5k	R-EU_0207/2V	R1, R2, R10, R11, R12
2	10k	R-EU_0207/2V	R8, R9
1	33k	R-EU_0207/2V	R4
3	36k	R-EU_0207/2V	R5, R6, R7
1	51k	R-EU_0207/2V	R3
2	100µF	CPOL-EUE2.5-7	C4, C10
11	100nF	C-EU025-025X050	C1, C2, C3, C8, C11, C12, C13, C14, C15, C18, C19
3	390 ohm	R-EU_0207/2V	R13, R14, R15
2	7805T	7805T	CONTROLLERPOWER, SERVOPOWER
1	m328p, 16MHz, 5V	ARDUINOMINIPRO	U$1
3	BC548B	BC548B-NPN-TO92-EBC	T1, T2, T3
3	Pinheader	PINHD-1X2	JP1, JP2, JP3 = connectors for LED1,2,3
6	Pinheader	MA03-1	5VOUT, RECEIVER, RS232, SERVO, UART, BATT
1	MAX232 CPE	MAX232	IC3

For the mini version: Open schematics in Eagle, run ULP script BOM.ulp. This will generate a list like the one above.

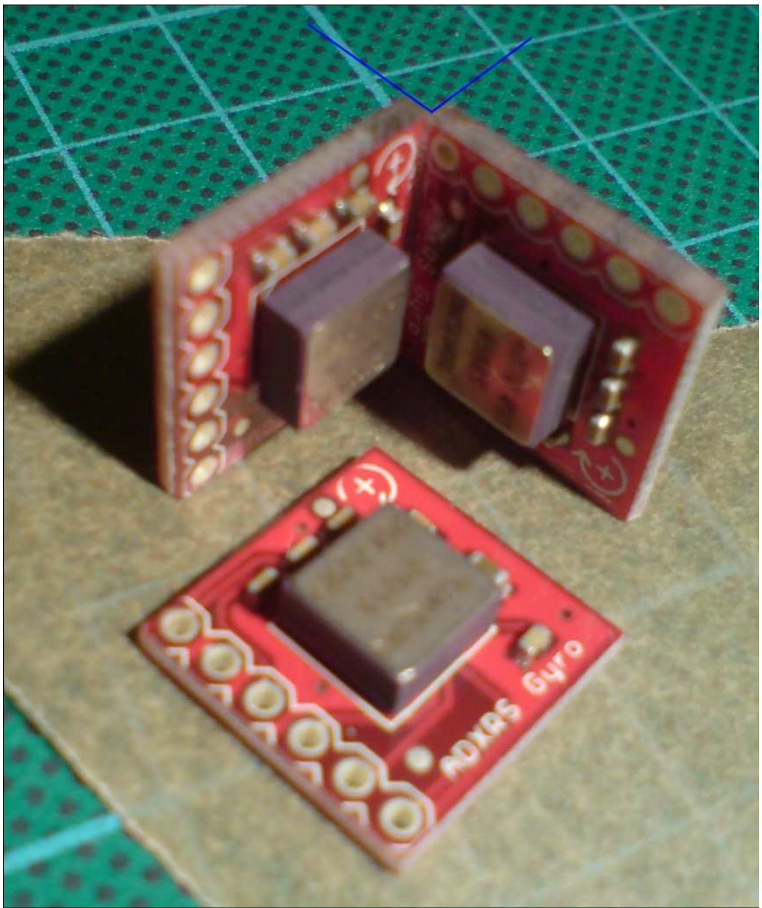
IMU (inertial measurement unit) Cube

The next step is to make the IMU cube. It consists of three gyroscopes (ADXRS610, Sparkfun breakout board) and a two axes accelerometer (ADXL322, Sparkfun breakout board). The sensors are not put directly onto TriGUIDE so that you can use these (expensive) sensors for other projects too. Additionally, this enables a better isolation against vibrations.

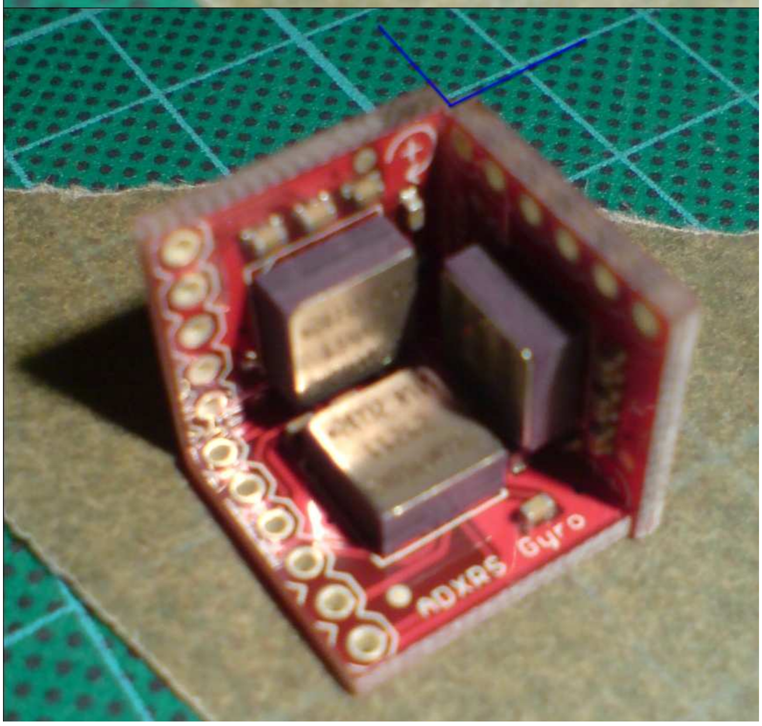
The individual sensors are glued together using cyano. The edges where you put the glue should be slightly sanded before. Please excuse the quality of the images. I have to use a mobile phone to make the pictures.



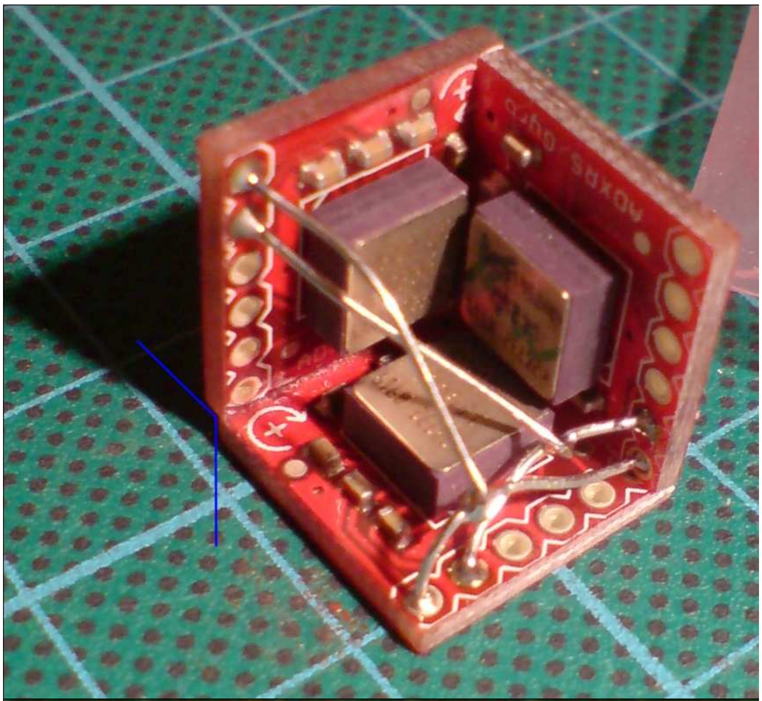
1: Glue two of the gyroscopes together. Note the blue lines and the positions of the soldering pads.



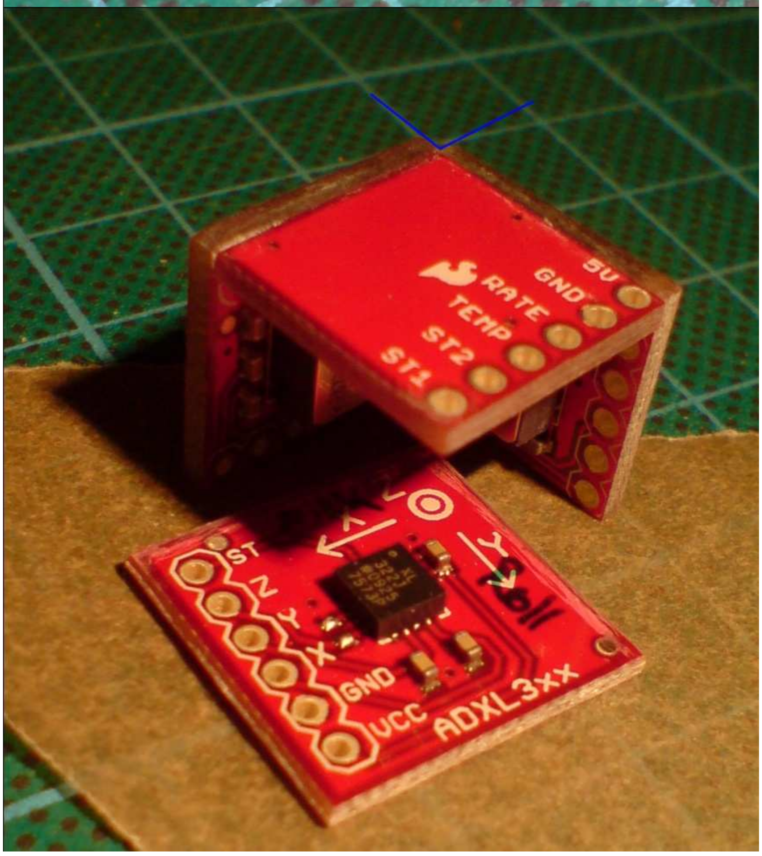
2: The third gyroscope is added like this. Note the blue lines and the position of the soldering pads.



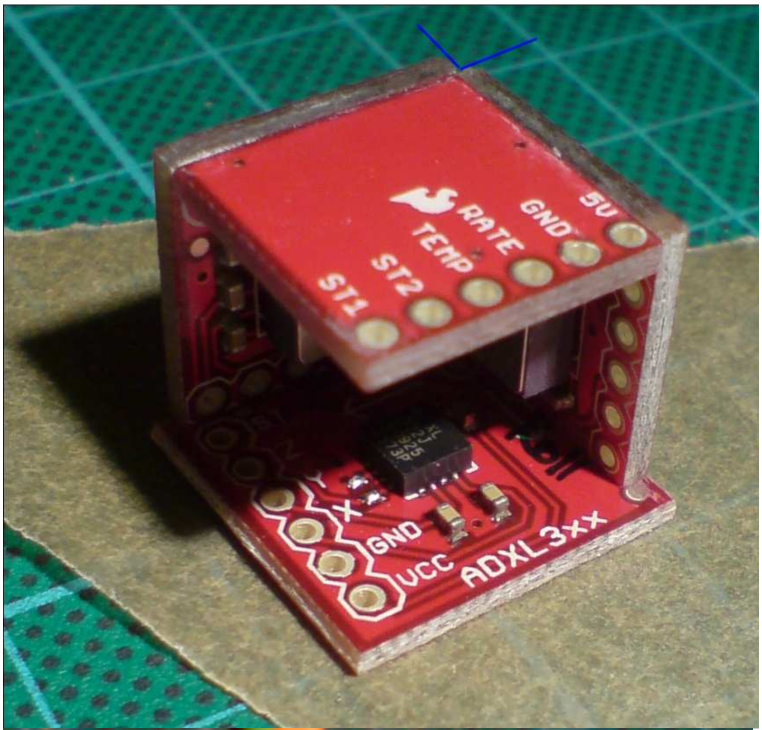
3: Glue the third gyroscope. Note the blue lines and the position of the soldering pads.



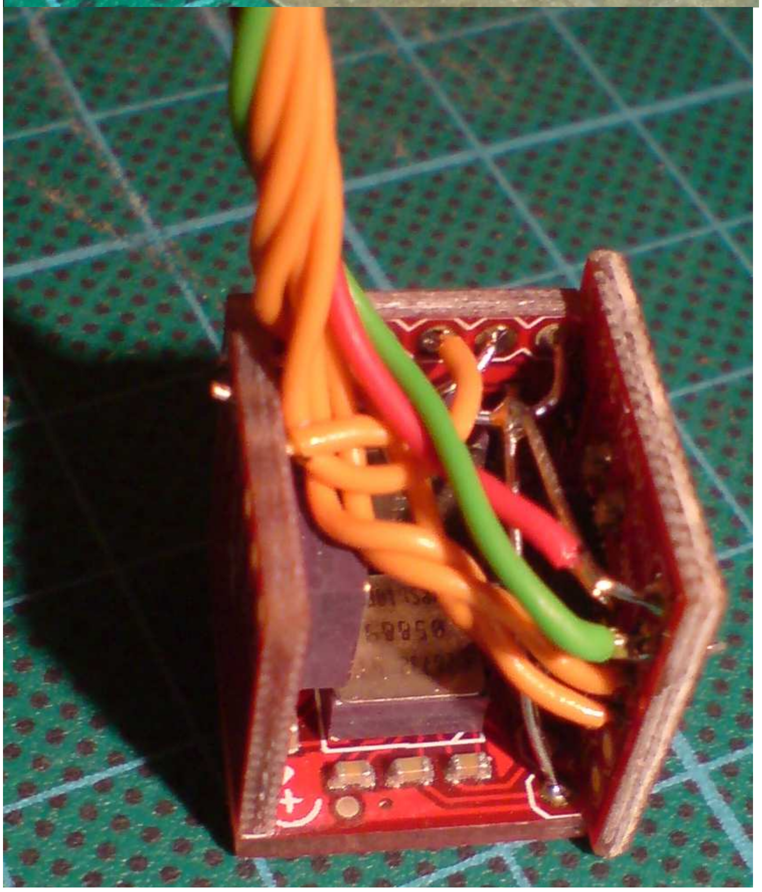
4: Connect GND and +5V of the individual boards. Pay extra attention that the wires don't touch anything (including the sensor elements).



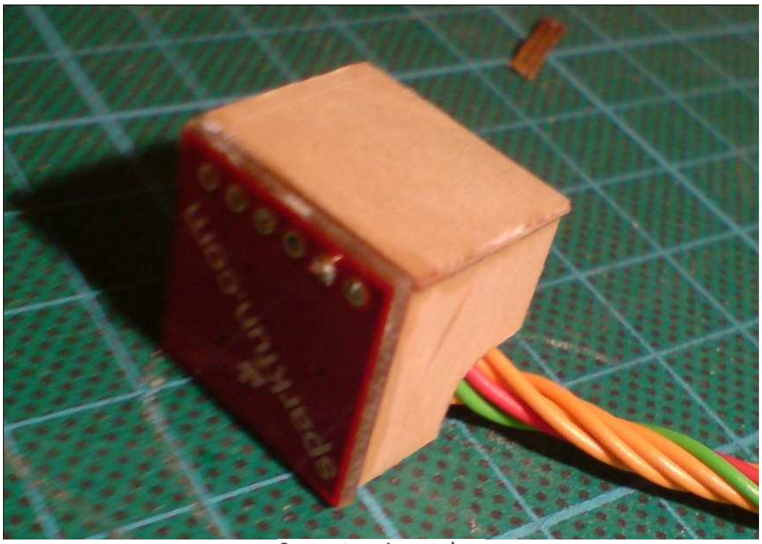
5: The accelerometer is attached like this. X is nick and Y is roll. I recommend marking these axes somewhere on the outside of the cube too.



6: Glue the accelerometer. Note the blue lines and the position of the soldering pads. Connect +5V and GND of the accelerometer.

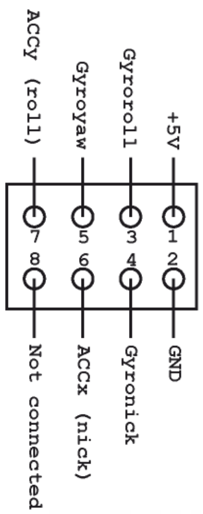


7: Attach wires to +5V and GND, as well as to each rate output of the gyroscopes and the X & Y output of the accelerometer (7 wires in total).

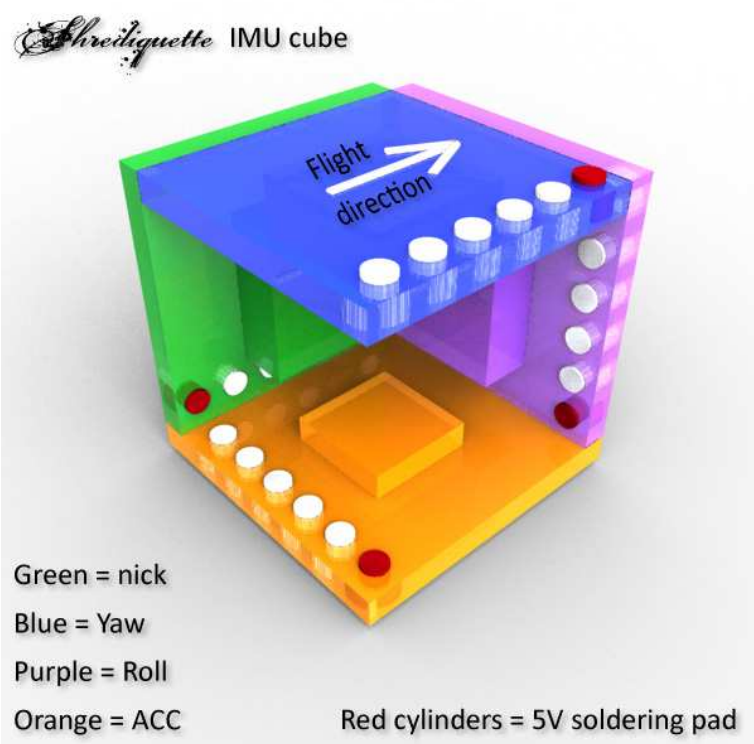


8: Close the cube with some pieces of plywood or CFK or whatever you have.

Connector, view at the top of the TriGUIDE PCB

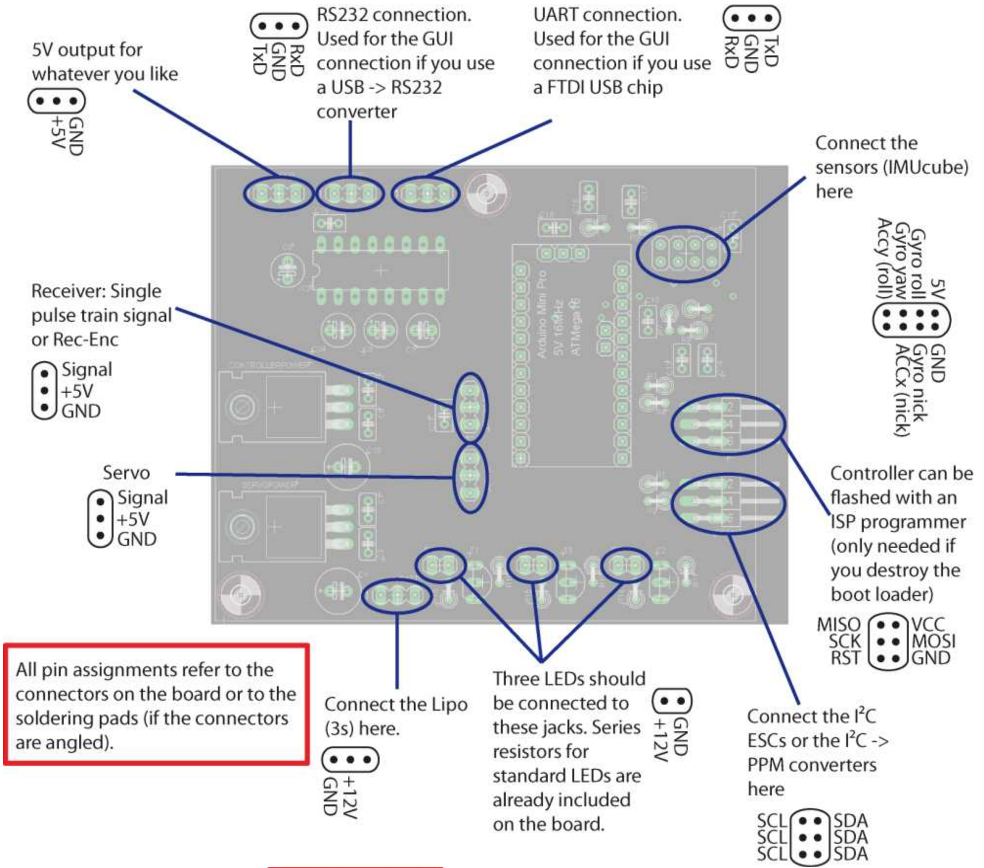


9: This is how the wires have to be connected to the jack. Note that this is a view on top of the TriGUIDE PCB.

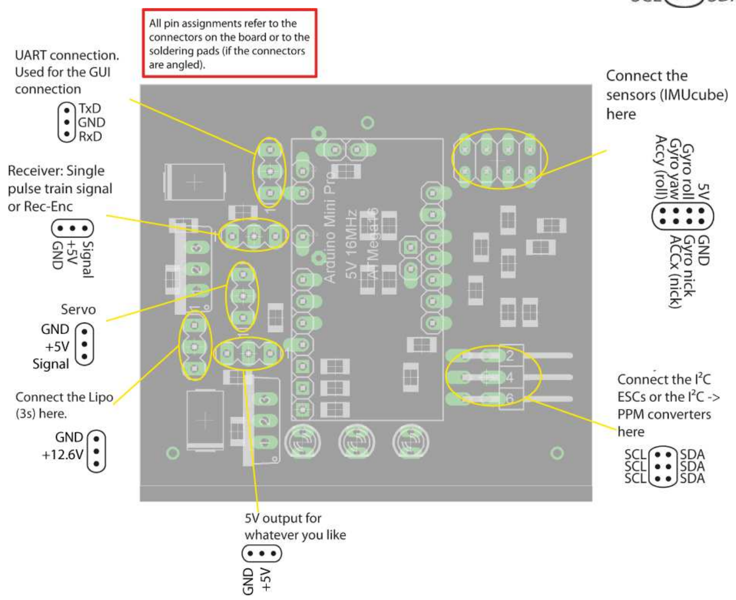


3D sketch of the IMU cube

Connector description



Here, you'll see the description and pin assignment of all connectors on the board.
TOP VIEW



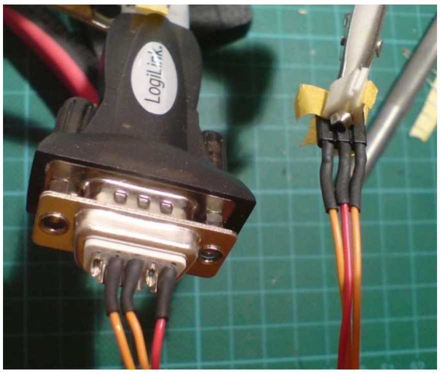
Mini version
TOP VIEW

Serial adapter

In order to program the Arduino pro mini, you'll need a serial connection to your PC. There are quite cheap USB to RS232 converters available. These converters work with +-12V, therefore it is extremely important that you connect them to the correct connector (RS232 connection)! You can also use a

Sparkfun FTDI USB to serial breakout board, in this case you have to connect it to the UART connection. For the mini version you have to use a FTDI USB to serial board. If in any of the above mentioned cases the communication doesn't work, try to turn the connectors 180 degrees first.

Here's a picture that shows how to make a cable for the USB -> RS232 converter.



Pin 2 & 3 are RxD & TxD, pin 5 is ground.

Receiver wiring

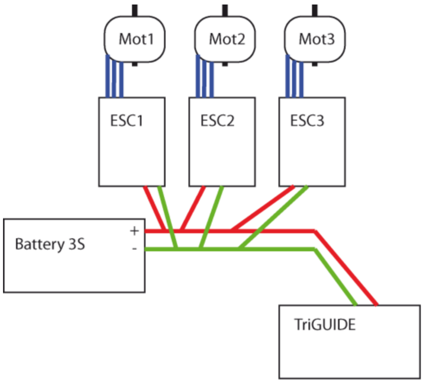
You can either use a receiver that delivers a single-pulse-train signal (buy one or convert an existing receiver), or you can use a thing called “Rec-Enc” (tt-copter.de), that generates a single-pulse-train signal out of any standard receiver. I noticed that there seem to be many definitions of the expression "single pulse train". If you are not sure about this, or if you would like to avoid potential problems, use a Rec-Enc. Using a Rec-Enc might result in a short time delay between your commands on the transmitter and the reaction of the tricopter.

This picture shows how to wire the Rec-Enc. +5V, GND and Signal go from the TriGUIDE to Rec-Enc. Channel 1 of the receiver is connected to 5V, GND and the first channel @ Rec-Enc, channel 2 of the receiver is connected to channel 2 @ Rec-Enc etc.

Please note: *You have to connect six channels of your receiver with the Rec-Enc! That is not yet shown in the picture on the left but I strongly recommend this!*

Power wiring

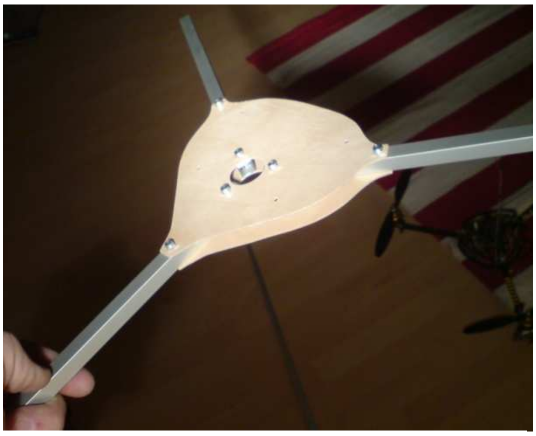
Each of the ESCs and the TriGUIDE PCB have to be connected to the battery. I recommend not using a power switch for safety reasons.



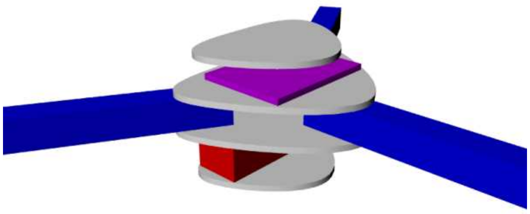
ESCs and TriGUIDE are all connected to the same 3S lipo battery.

Chassis

A cheap way to build a chassis is to use plywood (1mm thick, this doesn't seem to be very stiff at first, but wait until you built the sandwich structure…) and square aluminium profiles (10x10mm). But everyone is free to build more sophisticated chassis'. The chassis consists of 4 plywood layers: The top layer sits on top of the TriGUIDE PCB and protects the pcb, or can be used to attach a camera etc. The middle two layers sandwich three aluminium profiles. The bottom layer holds the lipo. The length of the arms doesn't really matter. Try to build something between 8cm and 30 cm (distance motor shaft -> centre of gravity).



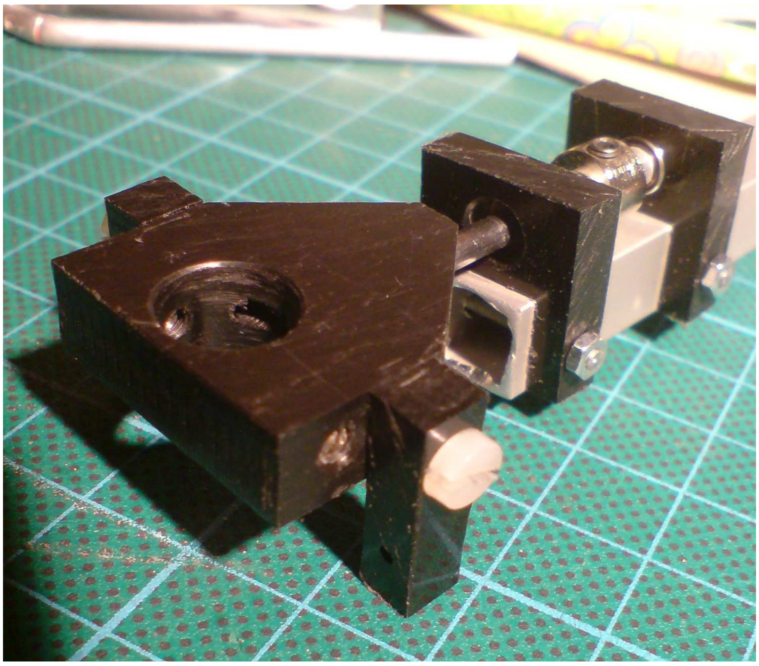
A simple chassis (middle two layers) made from plywood and square aluminium profiles.



Layer structure: The gray things are plywood etc., blue = aluminium profile, purple = TriGUIDE, red = battery. The ESCs should be placed between the middle two layers next to the aluminium profiles.

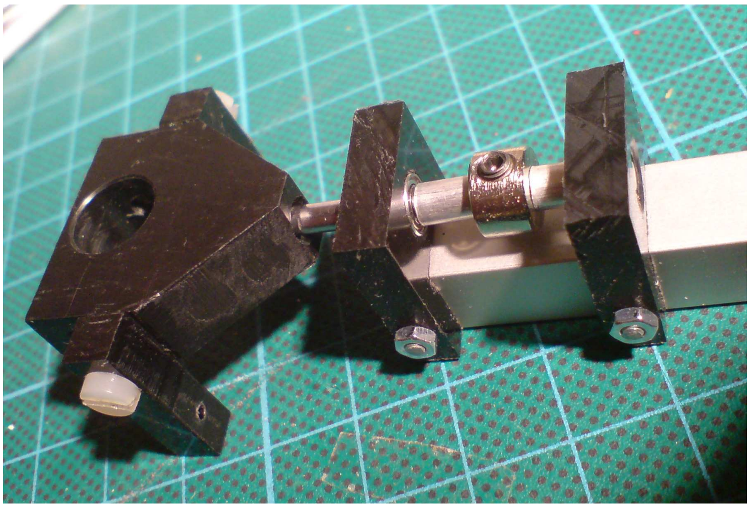
Rear servo

Principle: The motor is mounted to a block of Delrin (POM – or something else). A servo is mounted below the motor. This Delrin block is attached to the aluminium profile with two ball bearings. You should aim for an angle of about +- 30 degrees. ***Important note:*** *I just found out that in order to use a digital servo, you have to put a resistor in series with the signal line (that is already included on the TriGUIDE mini)! Otherwise the servo will vibrate slightly. Use a resistor of about 1 kilo Ohm. I have no explanation for this behavior; I noticed that by chance… Might be a good idea to put a resistor too when using analog servos. The "best" servo I found till today is the Bluebird BMS-371. Although I like high quality components, like Futaba digital micro servos, this low-budget servo gave the best results until now. If you have to order them, make sure to order at least three. If you crash, they easily break (but 6 Euros isn't too much for a servo, right…?).*

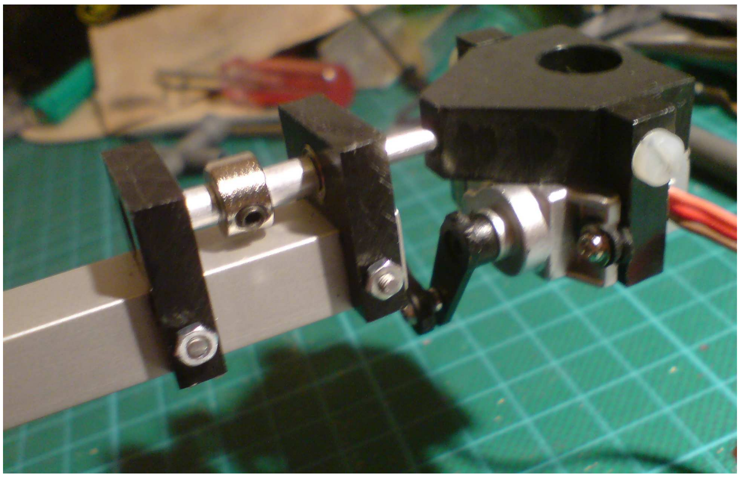


Motor mount (front) and ball bearing brackets. Again, this is just an example. Be creative to adapt your own motor mount. I can imagine that it would also work without any ball bearings.

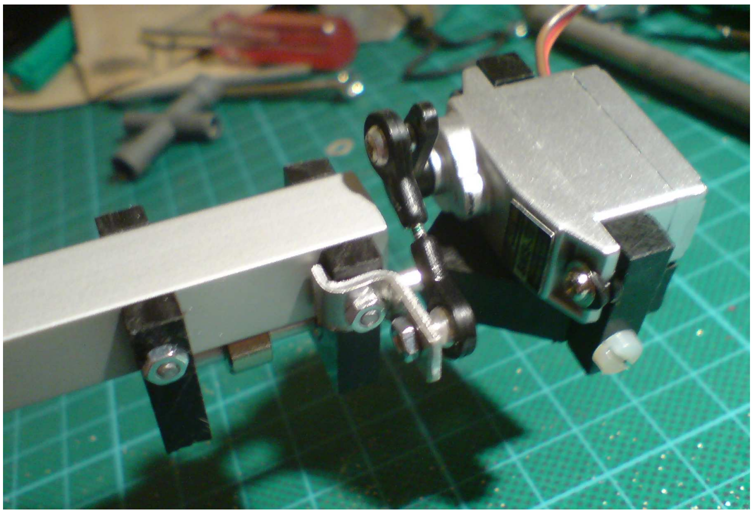
But make sure that you have only very little play in the linkage.



An additional view on the mount.



Servo in place. Top view.



Bottom view.

Your servo has to be mechanically adjusted for the right position during flight. If you switch from acro to hover mode during flight, the tail shouldn't move. If it does, that means that your servo is not set up correctly (in neutral position during flight).

Radio settings

Servo throw on the transmitter for roll, nick, yaw and throttle should be +-150% (Graupner/ JR) or +-130% (Robbe/Futaba) on all channels. **No** mixers must be activated, all trims must be **zero**.
Concerning channel 5: You have to connect a 3-stages switch to this channel. It must be configured as a regular channel. That means, if you would connect a servo to channel 5 of your receiver, it should move in one direction when the switch is in position 1 (motors off); the servo should be centered when the switch is in position 2 (acro mode), and it should go to the other direction when your switch is in position 3 (hover mode). If you aren't sure about your radio settings, please flash the file "Tricopter_ReceiverCheck.hex" to the arduino and check your channels. And read the readme called "ReadMe_ReceiverCheck.txt".

Transferring the firmware

First of all, you have to install the GUI. Get the latest version from my blog. You'll also need the latest firmware. Extract the GUI to a folder of your choice and put the firmware (.hex) into the same folder. The hex file has to have the name "Tricopter_m328p_11.hex". Connect your USB to serial converter.

Then find out which COM port it uses. In Vista this can e.g. be found out by going to the control panel, then click device manager, and then search for your serial converter. It should be somewhere in interfaces (COM & LPT) or something similar (sorry, I only have a german Windows release…). It will tell you which COM port it uses. Before attempting to connect to the GUI, install the receiver, the IMU-cube and switch on your transmitter. Remove any propellers from the tricopter.

Then, open up TriGUI, select the com port of your converter and click *connect + read out*. The GUI should say something like "*Searching…*". It won't find the tricopter, because the tricopter doesn't know how to answer the connection request (no firmware on it yet…). Go to the tab *Firmware* update and click Flash *new firmware*. Read the message boxes… They tell you that you have to press the reset button on the Arduino the same moment that you press ok on the second message box. It might be that this doesn't work the first time you try it, but after maximally three trials it should work. Otherwise try to turn around your RS232 cable. If that step was finished successfully, power off your tricopter and power it on again. Now it's important that you mounted the LEDs. They will tell you several things about the tricopter:

LED code	**Message**
Two LEDs flashing simultaneously after the battery was connected **VIDEO**	Throttle stick and/ or channel 5 (the 3-stages switch) are not in idle position. Shrediquette **will not initialize** until these two are in the correct position (throttle and switch at minimum).
LED1 flashing very fast, then LED2 flashing, then LED3 flashing **VIDEO**	Shrediquette is measuring the offset of the gyroscopes. Do NOT move the tricopter during this calibration. *Note: in the next firmware, this will be changed. The tricopter will automatically detect if it is not moving and it will then do the calibration. If it is however moved, LED2 will flash with around 2Hz and it will NOT initialize.*
Green LED on the Arduino is green 95% of the time, LED1,2,3 flashing two times slightly time shifted, then pause. **VIDEO**	Shrediquette is in hover or in acro mode
LED1,2,3 flashing simultaneously 3 times, then pause. Green LED on the Arduino flashes together with the other three LEDs **VIDEO**	**Shrediquette is ready for the connection with the GUI. Only now you can connect to the GUI.**
All LEDs flash simultaneously and fast, green LED off **VIDEO**	Battery is empty (or voltage is below low voltage threshold)

If the firmware was flashed successfully and Shrediquette is ready for a connection with the GUI, go back to the tab *Connection* and press *connect + read out*. TriGUI should find the tricopter. You can now write the standard parameters to the tricopter: Click *load file and write to µC* and select ***Defaults.shr***. (Note: if you didn't transfer the standard parameters, nothing will work as expected. So do this first). Now you should check your transmitter settings. First, select the correct channels in the tab *Radio settings* under *RX/TX channel assignment*. Write these parameters to the tricopter (tab *Connection*, press *Write all parameters to µC*). Now have a look if your channels work in the correct direction. Go to the tab *Realtime data* and have a look at all the values. The sliders should move according to your sticks. Otherwise enable servo reverse in your transmitter (note that the mot on/ mot off slider might not work correctly if the checkbox *disable motors* is unchecked, because data transfer from the tricopter is stopped if motors are running). Also the gyro's should react if you move the copter. When you tilt the copter to the left, the sliders should move to the left too. When you tilt the copter forward, the sliders should move up (= forward) etc. If they work in the wrong direction, go to *PID loop* and reverse the gyros/ ACC. If you did some changes, **write the parameters to the µC** and check again in *Realtime data*. If everything seems to work fine, you can enable the motors (tab *radio settings*, uncheck *deactivate motors*). **DO THIS ONLY IF YOU'RE 100% SURE THAT YOUR TRANSMITTER WORKS CORRECTLY!** Otherwise you won't be able to reconnect to the GUI without erasing the EEprom.

You could now try to start the motors with the 3-stages switch. But do not mount the propellers yet! Check if the tricopter reacts the right way.

GUI parameterization

Parameter	Effect
ACRO: Stick sensitivity roll & nick	Sets the agility of the tricopter in Acro mode.
Hover: Stick sensitivity roll & nick	Sets the agility of the tricopter in Hover mode.
Yaw sensitivity	Sets yaw agility for both flight modes.
ACRO: Exponential agility boost	If you plan to fly loopings etc., this might be handy: When the stick is far out of the centre, the agility of roll and nick get an extra boost.
Minimum throttle	Minimum throttle of the motors. Should be low enough not to take off with the throttle stick at minimum, and high enough to enable a smooth start of the motors.
Controls (RX/TX channel assignment)	Assign your transmitter controls here
Deactivate motors	When this is checked, the motors won't start. At least they shouldn't. If something goes really wrong, they could eventually still start. So always be careful and always dismount the props when trying something new.
Roll/ Nick/ Yaw gyro reverse	Does what it says
X/ Y acc reverse	Does what it says (surprise...!)

Parameter	Effect
ACC influence	How much the angle estimation from the gyros is affected by the accelerometer. Shouldn't be changed unless you use a different acc/ gyro.
X/Y acc scale	The gain of the acc's. ACC reaction to tilting the tricopter should be similar to gyro reaction. Shouldn't be changed unless you use a different acc/ gyro.
X/Y acc offset	**This is a pretty important parameter:** You'll have to find the correct values in order to make the copter level correctly in hover mode. If your copter flies to the left/right/front/back, you'll have to adjust these values in small steps. Once you found the right offset, your copter will exactly know when it is leveled and only then it can hover in place. You will also not need to trim the tricopter anymore.
Hover mode P	Control loop reaction to errors in angle. This parameter has to be adapted to your setup.
Hover mode I	Reaction to a prolonged error in angle. This parameter has to be adapted to your setup.
Hover mode D	Reaction to errors in angular velocity. This parameter has the most noticeable effect. Too high and the copter will swing up (deadly for the copter…), too low and it will be imprecise. This parameter has to be adapted to your setup.
Hover mode D²	Reactions to angular acceleration. This parameters makes it possible to increase Hover mode D further without oscillation. If D² is too high, the tricopter will be very “noisy” in the air. This parameter has to be adapted to your setup.
Acro mode P	Reaction to errors in angular velocity
Acro mode I	Reaction to errors in angle. This parameter has to be adapted to your setup.
Acro mode D	Reactions to angular acceleration. This parameters makes it possible to increase Acro mode P further without oscillation. If D is too high, the tricopter will be very “noisy” in the air. This parameter has to be adapted to your setup.
Yaw P	Reaction to errors in angular velocity. If too high, the servo will be very noisy. This parameter has to be adapted to your setup.
Yaw I	Reaction to errors in angle. If too high, the servo will be very noisy. This parameter has to be adapted to your setup.
Voltage warning	If the voltage drops below this value, all three LEDs will begin to flash fast. It's time to land…! This value depends very much on the battery you are using. I recommend anyway to use a timer in your transmitter.
Nr. of receiver problems	Counts the number of "errors" that were detected.

Things to consider before takeoff

When thinking about the first take-off of the tricopter you have to consider some specialties (I think these apply to all multicopters):

When starting the motors, Shrediquette instantly tries to control its attitude. If the floor is not 100% leveled (this will always be the case…), the tricopter thinks that it is not in level flight. It will try to correct for that, but the poor thing can't! It's still on the ground and leveling is only possible in flight. The integral parts of the PID control loop are responsible for this behavior. The measured error sums up and becomes bigger and bigger. Therefore the tricopter tries to correct more and more. There are two work-arounds: You can start the motors and immediately take-off, or you can simply get used to this behavior and accept that the tricopter won't level perfectly during the first 1-3 seconds of flight. The latter is what I do.

When switching from hover mode to acro mode and vice versa, the integrals are deleted (or replaced with the angle estimation from the accelerometer). So if you think that Shrediquette was desperately trying to level without succeeding, switch to acro mode and back to hover mode to let it try again.

If you want to turn off the motors, bring the 3-stages switch to the off position. Your motors won't turn off until you also brought the throttle stick to a position close to minimum. That’s a safety feature, because I already managed to accidentally turn off the motors while I was flying pretty high. Shrediquette is not a good glider I must admit.

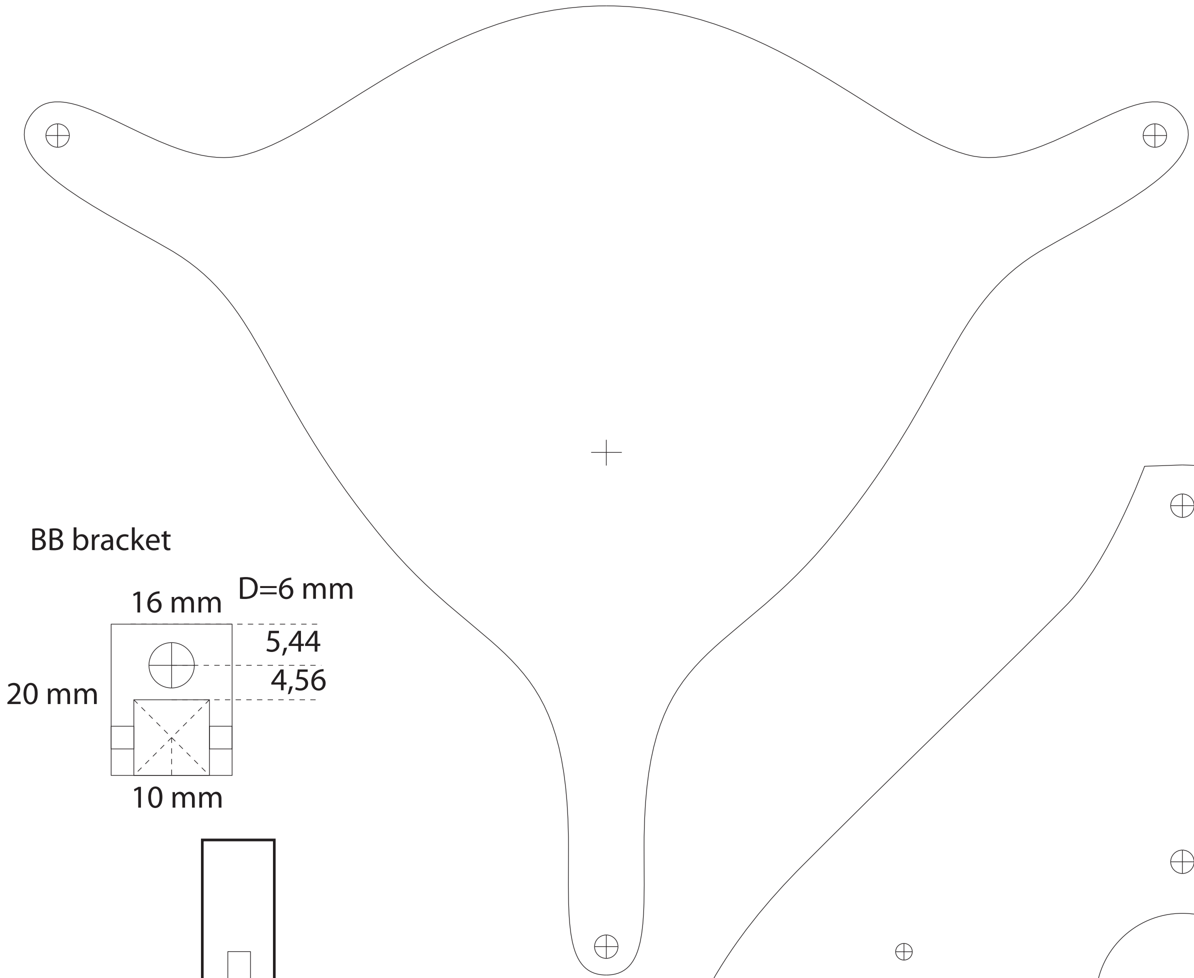
First flight should be done in hover mode by holding the tricopter in your hands and increasing throttle slowly. Keep in mind: Don't grab it too hard, the tricopter must be fully able to control its attitude in all axes; otherwise "strange" things might happen.

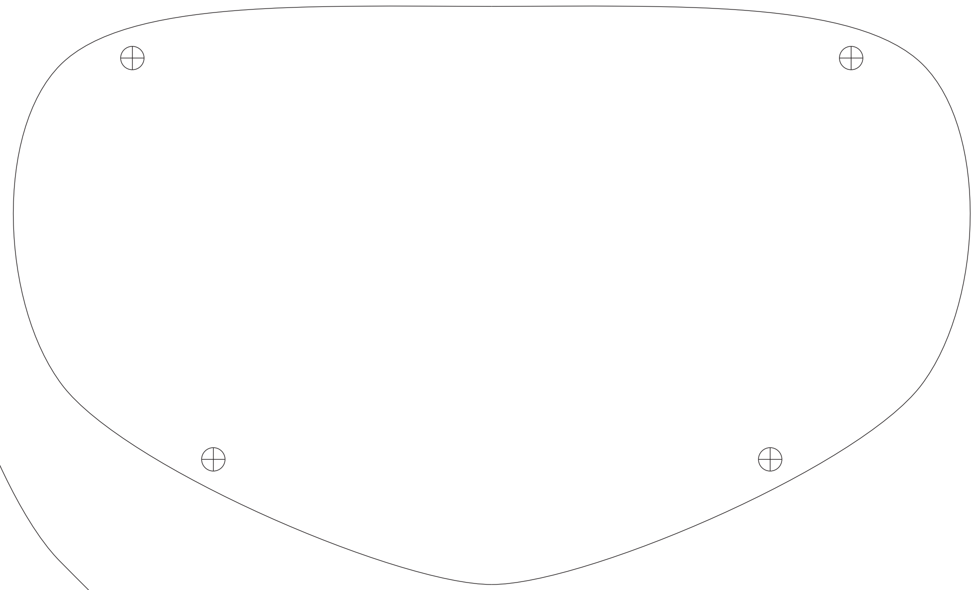
Have fun with your tricopter and please always give me some feedback or videos & pictures…!

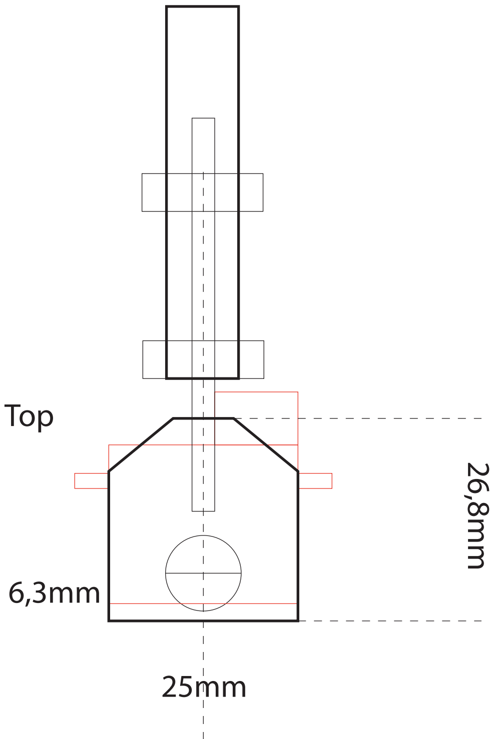
All the best,

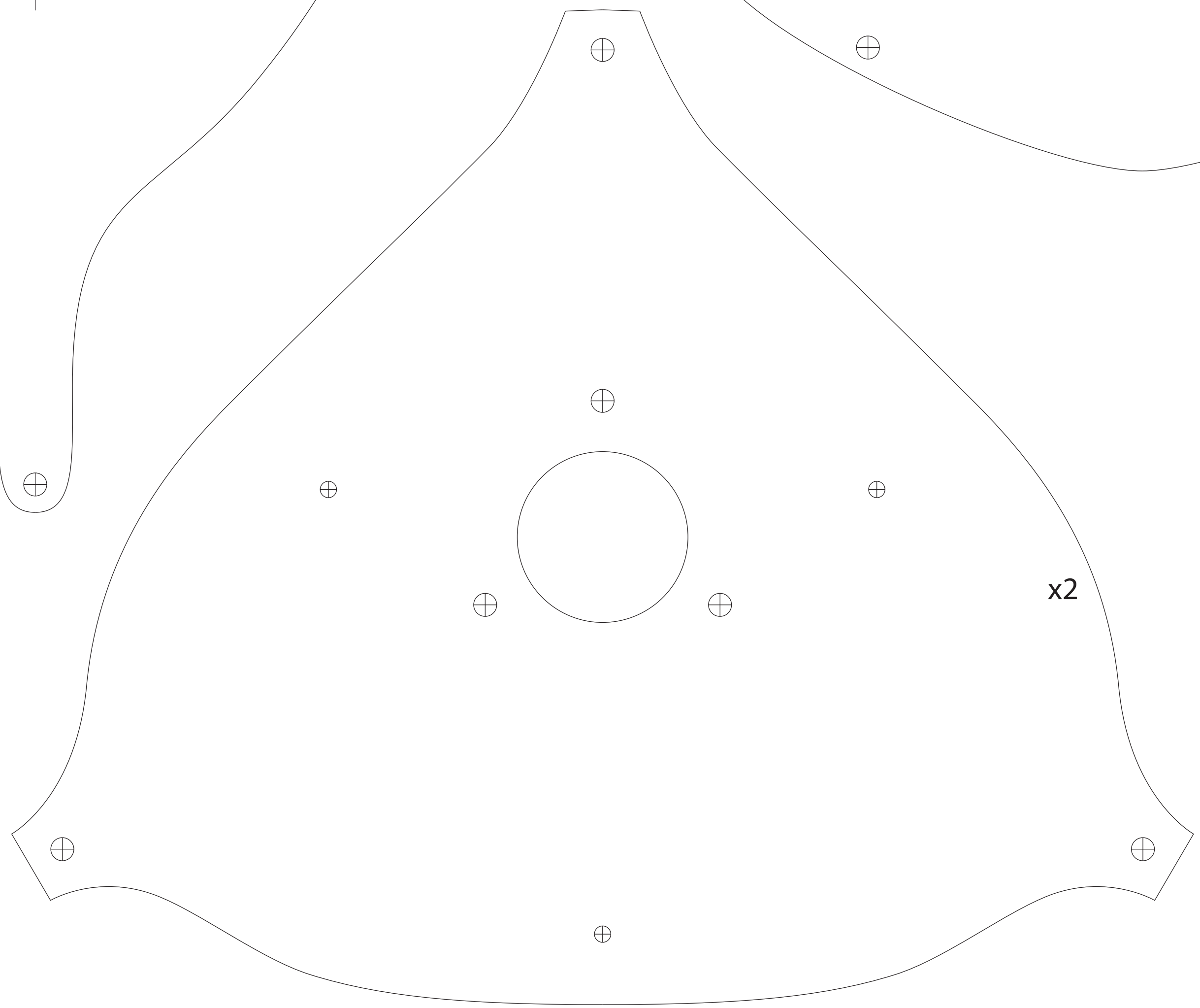
William

Appendix


BB bracket
16 mm
D=6 mm
5,44
4,56
20 mm
10 mm




Top
26,8mm
6,3mm
25mm


x2