

NOOBIES GUIDE to setting up and using the CRIUS MWC MultiWii SE Board Software By ATXHELI

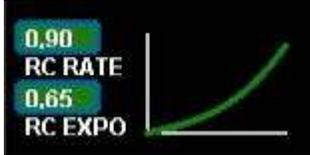
1. Download and unzip the required software:
 - Arduino Compiler and Board Flashing Environment:
<http://arduino.cc/en/Main/Software>
 - use the 0023 IDE not the 1.0 version
 - Latest MultiWii Software including Configuration Program:
http://multiwii.googlecode.com/files/MultiWii_1_9.zip
2. Attach the FTDI USB board to the MultiWii SE board and plug in USB cable to PC
3. New Hardware dialogs will appear and point to the \drivers\ftdi usb folder in the Arduino folder for the driver. Driver should install and new device be recognized.
4. Start the MultiWii Configuration program.



5. See these tutorial videos on YouTube for operating the program:
<http://www.youtube.com/watch?v=qT3MlyRvy9w>
- He has several other videos on the Configuration program that might be of interest also on his YouTube channel.

Some items in the program that might not be obvious (taken from Multiwii.com site):

RC rate and RC expo



RC rate: defines the sensibility of the multicopter for PITCH and ROLL RC stick. If you feel your multi too reactive, decrease it. If you want to improve its responsiveness, increase this parameter.

RC expo: defines a smoother zone at the center of PITCH and ROLL RC stick. With this function, it's possible to have both a good precision and a good amplitude.

0 = no expo

1 = full expo

If you don't understand really those parameters, I encourage you to read your TX documentation, as those functions are also generally supported by the TX.

It's better to support them directly in Multiwii because the resolution is better this way.

P and I Level



Defines the acc influence in stable mode only (this parameter has no effect if the stable mode is disable) If the multi is not stable once activated, reduce P.

PITCH/ROLL/YAW PID and rate (see tuning section later)

	P	I	D	RATE
ROLL	4.0	0.030	15	0.00
PITCH	4.0	0.030	15	0.00
YAW	8.0	0.000	0	0.00

For a very stable multicopter with a solid attitude, the PID settings must be set high. But if you want to do some acrobatics with these settings, the multi starts to wobble in fast translation or when you decide to shake it to much. One solution is to decrease the PID, but it is to the detriment of static stability.

So instead of implementing a switch between 2 PID settings, there are now 2 options in the GUI to define the way we should decrease PID, depending on ROLL/PITCH/YAW stick deviation

The two boxes on the right defines the rate of cancellation of the nominal PID (the one which is used on neutral ROLL/PITCH/YAW stick position) in relation with ROLL/STICK/YAW deviation. In fact only P and D parameters are impacted in the transformation. The purpose of this setting is not really to gain more stability, but to gain more maneuverability. 0 = soft rate (for FPV or beginners); 0.4 = soft acro ; 0.7 = fast acro ; 1 = insane rate

If you are not familiar with this, just keep the default values (0)

THROTTLE rate



This box defines the rate of cancellation of the nominal PID (the one which is used on neutral ROLL/PITCH stick position) in relation with Throttle stick.

The purpose of this curve is to gain more stability when you are in a situation to use more throttle than needed to just maintain a lift. This is typically the case in fast translation.

If you are not familiar with this, just keep the default values (0)

How to activate options



Depending on your configuration, you can easily activate/deactivate options via 2 switches on your RC TX. The 2 switches correspond to (AUX1), typically RX ch5 and, RX channel 6 (AUX2).

LEVEL is the autolevel feature (require an ACC) which will attempt to hold the board level if no Pitch or Roll inputs are received...sometime called "stable" mode

BARO tries to keep a constant altitude once activated (require a barometer)

MAG tries to keep a perfect heading direction once activated (require a magnetometer)

CAMSTAB: if activated, the camera's PITCH&ROLL servo output will counteract the inclination of the multi (require an ACC + the code activation via #define statement)

CAMSTRIG: if activated, a servo output will trigger repetitively a sequence in which we can define the duration on the HIGH/LOW position.

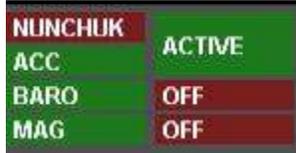
To activate a function, you must check the white box at the intersection of the required option (row) and at the switch state you want to enable (column).

It's possible to make a complex mix.

If you want to force an option permanently, it's possible to check all the white boxes, even without a RC channel connected.

(=> this way, you don't need a 5 or 6 channel RC receiver)

Sensor detection/usage



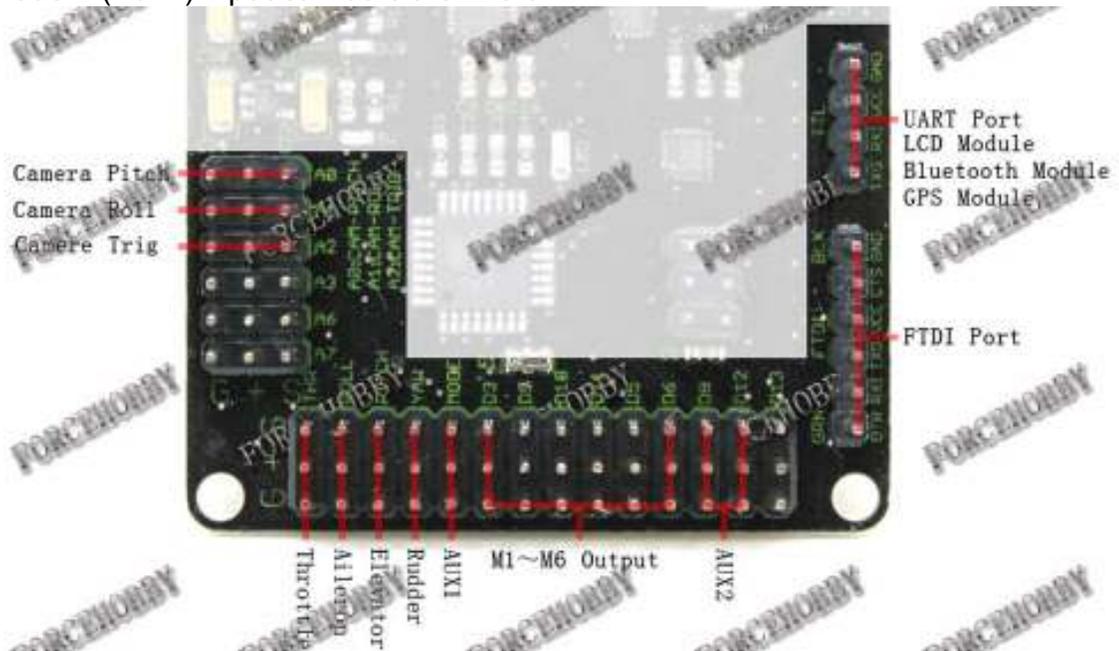
On this screen, the left column shows the detection of sensors on the board.

The right column shows if the sensor is currently use or not (depending on the checked options and AUX switch positions)

Basic Hardware Setup:

Connection Diagram:

Mode 2 (Aux2) input can be either D8 or D12.



Motor connections and rotations are shown below:



Numbers refer to pins D3,9,10,11,5,and D6 above...the Motor1-6 ESC outputs.

Note: if the ESC uses a LINEAR BEC, you can connect all to the motor outputs without modification. If the BECs are switching types, you should use only one BEC output and remove the red wires from the other ESC connectors. Alternatively, all can be removed and an External BEC connected to D13, A6, or A7 or any unused three pin connection.

1. **BALANCE PROPS and MOTORS!**
2. Check that all TX channels are centered at 1500 and run from 1000 to 2000.
3. Check that all channels are correct direction...ie right runs right on roll, etc., if not reverse them.
4. Calibrate Gyro (board not moving!)
5. Calibrate Accelerometers (board level!)
6. Calibrate Magnetometer:
press Calib_Mag - the LED blinks rapidly -**you have 30secs to rotate the board 360° degress in all axis X, Y and Z** - when the LED stops blinking it will save all the MAG tilt errors to eeprom.
7. Set ESC ranges:
Individually connect escs to RX throttle channel and follow ESC manual directions for setting min/max throttle range.
8. Fly, trim with autolevel off, adjust PIDs IF necessary...small changes, one at a time!
9. Set autolevel on and trim using stick commands to get drift free autolevel; adjust PIDS, if necessary.
10. Set altitude on and adjust PIDS for heading hold, if necessary

PID Tuning (extracted from reference at end):

the default PID values should get almost all normal sized multi copters up & flying pretty good.

P: 4
I: 0.035
D: -15

Further tuning is mostly done by just lowering / raising the P value.
In order to fine tune, increment/decrement the values with the following steps: P:0.2 ; I:0.005 ; D:1

"P" fights all motion... increase until you get a feedback wobble, reduce by ~20%

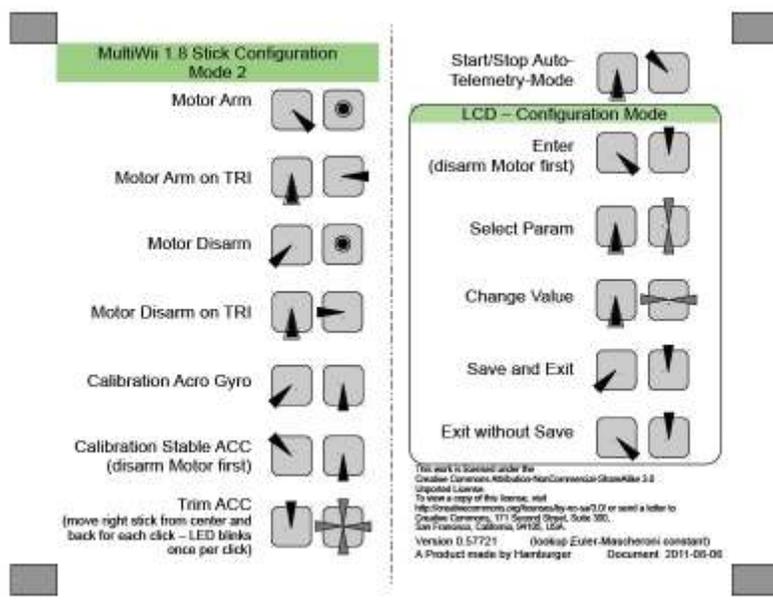
"I" is heading hold.... increase until you get a bounce back when you release a big stick input.

*"D" the bigger the better from what I can tell this tries to smooth out any rotation. The higher you run "D" it appears you can further increase "P"
"D" the bigger the better... to smooth out any rotation. The farther from zero you run "D" it appears you can further increase "P"*

Some recommend tuning P first, then D, then I.

Control Stick Configuration Guide (valid for V1.9 also) :

http://multiwii.googlecode.com/svn/branches/Hamburger/MultiWii-StickConfiguration-18_v0-57721.pdf



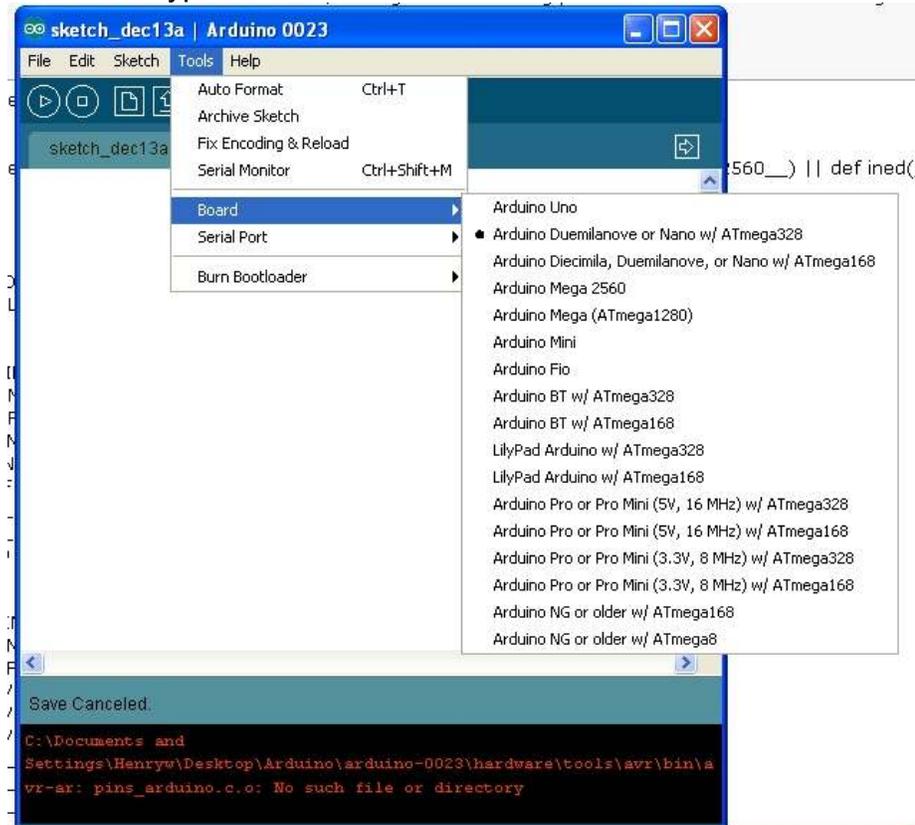
Additional Resources: A good Multiwii info source:

<http://www.rcgroups.com/forums/showthread.php?t=1340771#post16587954>

Changing the Copter Configuration: ie to Tricopter or other configuration:

Not required if using Quad X as shipped!

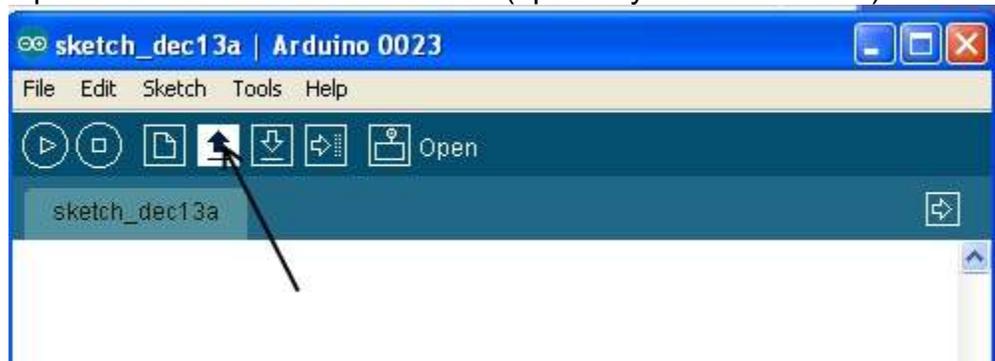
1. With FTDI USB connected, start Arduino.exe
2. Set Board type to :



3. Set Com port to the one the FTDI is on...the one used by MultiWii Configuration above (probably won't be COM1!).

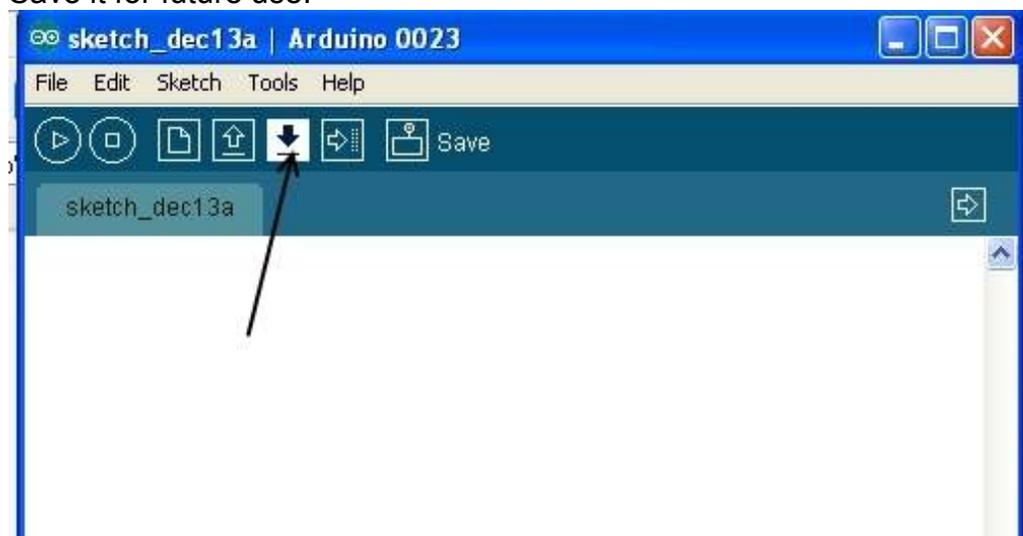


4. Open the MultiWii 1.9 source code (open any file in the folder)

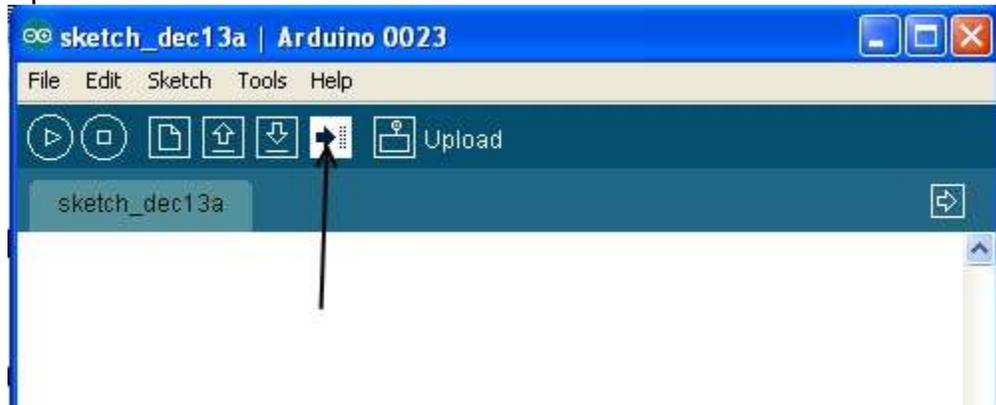


5. Select the config.h tab and uncomment (remove "//") the parameters as below, be sure to comment ("//") the unused ones in each group:
- Select the Min throttle for your ESCs..this is the lowest value the controller sends to any motor during flight...you want them still spinning at this value or you run out of control.
 - Choose the copter configuration
 - Reverse the yaw sense if needed
 - Use : `#define I2C_SPEED 400000L`
 - Use: `#define INTERNAL_I2C_PULLUPS`
 - Use: `#define RCAUXPIN8` or `#define RCAUXPIN12` if you want to use AUX2 for mode selection in addition to AUX1
 - Use: `#define ITG3200`
 - Use: `#define BMA180`
 - Use: `#define BMP085`
 - Use: `#define HMC5883`
- Everything else is left as is....

6. Save it for future use.



7. Upload the code to board.



8. Configure and Go FLY!