

# NEWBIES GUIDE to setting up and using the CRIUS MWC MultiWii SE Board Software

By ATXHELI *rev. 04/22/2012*

## 1. Download and unzip the required software:

-Arduino Compiler and Board Flashing Environment:

<http://arduino.cc/en/Main/Software>

-use the 1.0 version

For Windows: <http://arduino.googlecode.com/files/arduino-1.0-windows.zip>

For Mac: <http://arduino.googlecode.com/files/arduino-1.0-macosx.zip>

For Linux 32bit: <http://arduino.googlecode.com/files/arduino-1.0-linux.tgz>

For Linux 64bit: <http://arduino.googlecode.com/files/arduino-1.0-linux64.tgz>

&

-Latest MultiWii Software including Configuration Program:

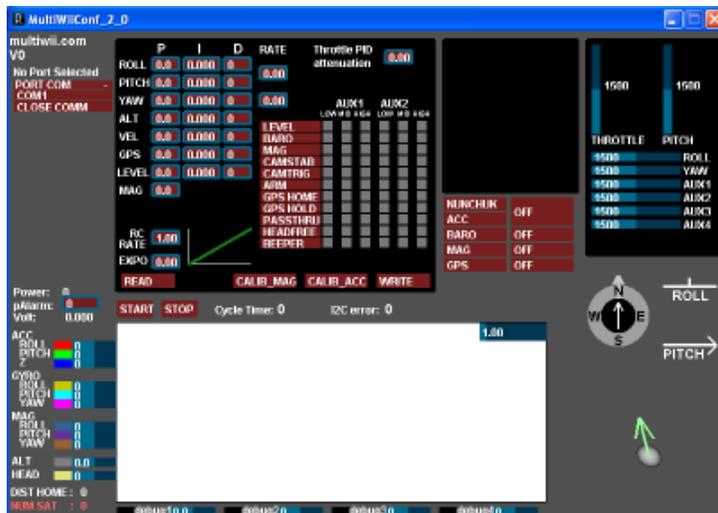
[http://code.google.com/p/multiwii/downloads/detail?name=MultiWii\\_2\\_0.zip&can=2&q=](http://code.google.com/p/multiwii/downloads/detail?name=MultiWii_2_0.zip&can=2&q=)

(Note: if the board has V1.9 preloaded, then use this software package [http://code.google.com/p/multiwii/downloads/detail?name=MultiWii\\_1\\_9.zip&can=2&q=](http://code.google.com/p/multiwii/downloads/detail?name=MultiWii_1_9.zip&can=2&q=) to get the Configuration program for V1.9, unless you want to upgrade to V2.0 which has better level and altitude hold operation, then you will need to compile and load the V2.0 code, see the last section of this guide)

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.



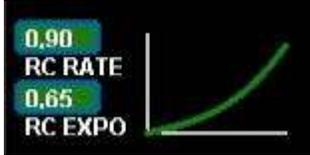
5. See these tutorial videos on YouTube for operating the program:

<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

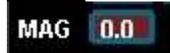
If you are not familiar with this, just keep the default values (0)

### ALT PID



These values adjust the altitude hold function. P,I,D are the same definitions as for the Pitch/Roll/Yaw loops above in terms of their effect on the Altitude hold loop. The VEL PID is not currently used any longer. The values shown above are NOT the default values. Once activated altitude hold is adjusted by throttle: above neutral increases altitude set point, below lowers it.

## MAG



This is the P term for the compass heading hold mode. Once this mode is activated, the compass heading will be held as long as rudder stick is at neutral. Changing the rudder stick will change the heading set point.

## 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). In Version 2.0 and later, two additional channels (AUX3 & 4) may also be used if you are using PPM composite receiver (see Other Neat Stuff section below)

**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.

**ARM**: allows arming in addition to via stick (see Stick Table below)

**GPS**: Not available with this board

**PASSTHRU**: Bypasses IMU for flying wings

**HEADFREE**: Heading is stored at power up from MAG, TX AIL and ELEsticks will move the copter in direction relative to initial heading no matter what actual heading is. Like CAREFREE mode in other controllers.

**BEEPER**: A buzzer can be attached to D8 for low voltage warning. This box enables it. Requires code options to be set also which are beyond the scope of this guide. Study the source.

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

NUNCHUK	ACTIVE
ACC	ACTIVE
BARO	OFF
MAG	OFF

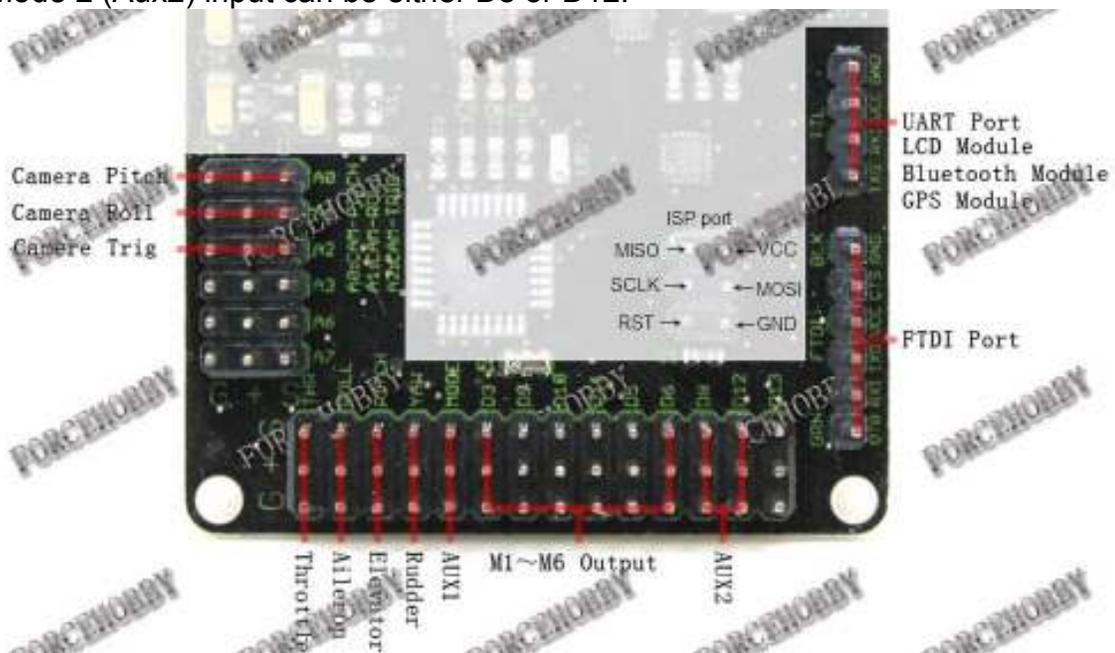
On this screen, the left column shows the detection of sensors on the board.

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 that on this board Motor 5 and 6 (HEXcopters) should actually be connected to A0 pin and A1 pins, respectively. There is an option in the code to use the D5 and D6 pins on a HEX, but a PPM-SUM type receiver must be used since RX Pitch and Yaw pins are the same connection on the board as D5 and D6. See Other Neat Stuff section at the end.

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

See this reference for more on trimming:

<http://www.multiwii.com/faq#How to trim your Copter: version 1.7 and more>

### **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 V2.0 also) :

**Note:**

This work is licensed under the  
 Creative Commons Attribution-NonCommercial-ShareAlike 3.0  
 Unported License.  
 To view a copy of this license, visit  
<http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to  
 Creative Commons, 171 Second Street, Suite 300,  
 San Francisco, California, 94105, USA.  
 Version 0.57721 (lookup Euler-Mascheroni constant)  
 A Product made by Hamburger Document 2011-08-06

MODE 1		MODE 2	
<b>MultiWii Stick Configuration</b> Motor ARM Motor ARM on TRI Motor Disarm Motor Disarm on TRI Calibration ACRO Gyro Calibration Stable ACC (Disarm motor first) Trim ACC (Move stick from center and back for each click - LED blinks once per click)	<b>Telemetry</b> Start/Stop Auto Telemetry mode  <b>LCD Configuration mode</b> Enter (Disarm Motor first) Select Parameter Change Value Save and Exit Exit without Save	<b>MultiWii Stick Configuration</b> Motor ARM Motor ARM on TRI Motor Disarm Motor Disarm on TRI Calibration ACRO Gyro Calibration Stable ACC (Disarm motor first) Trim ACC (Move stick from center and back for each click - LED blinks once per click)	<b>Telemetry</b> Start/Stop Auto Telemetry mode  <b>LCD Configuration mode</b> Enter (Disarm Motor first) Select Parameter Change Value Save and Exit Exit without Save
MODE 3		MODE 4	
<b>MultiWii Stick Configuration</b> Motor ARM Motor ARM on TRI Motor Disarm Motor Disarm on TRI Calibration ACRO Gyro Calibration Stable ACC (Disarm motor first) Trim ACC (Move stick from center and back for each click - LED blinks once per click)	<b>Telemetry</b> Start/Stop Auto Telemetry mode  <b>LCD Configuration mode</b> Enter (Disarm Motor first) Select Parameter Change Value Save and Exit Exit without Save	<b>MultiWii Stick Configuration</b> Motor ARM Motor ARM on TRI Motor Disarm Motor Disarm on TRI Calibration ACRO Gyro Calibration Stable ACC (Disarm motor first) Trim ACC (Move stick from center and back for each click - LED blinks once per click)	<b>Telemetry</b> Start/Stop Auto Telemetry mode  <b>LCD Configuration mode</b> Enter (Disarm Motor first) Select Parameter Change Value Save and Exit Exit without Save

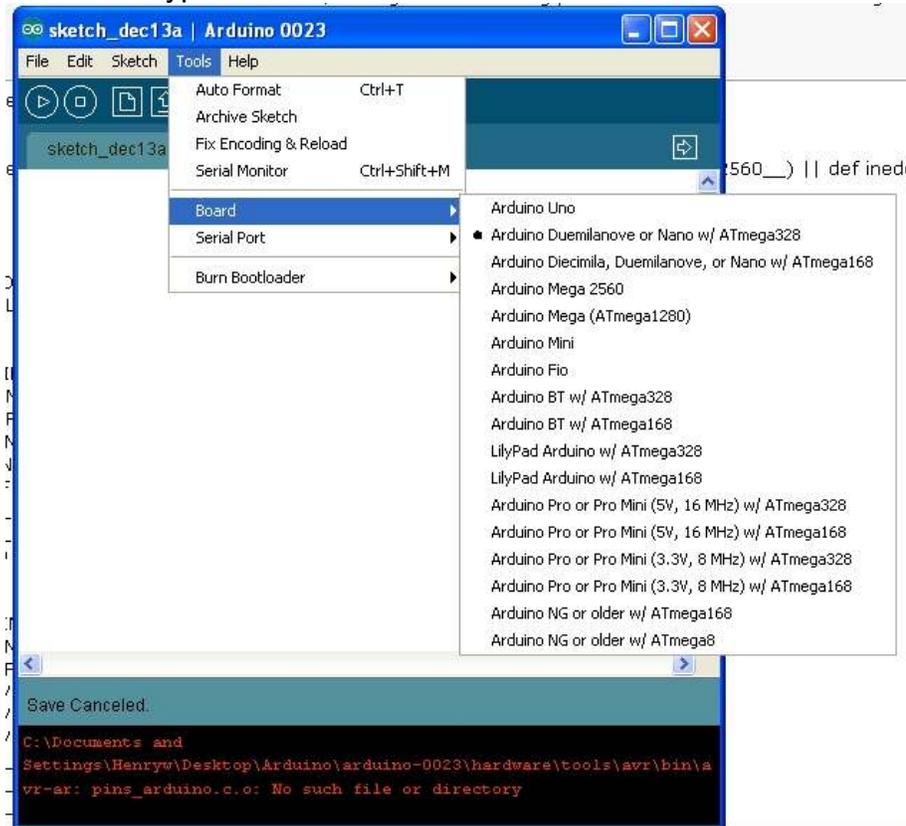
**Additional Resources: A good Multiwii info source:**

<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! However, you must use MultiWii Conf1.9 since that is the version presently shipped. Will no doubt change to 2.0 shortly.*

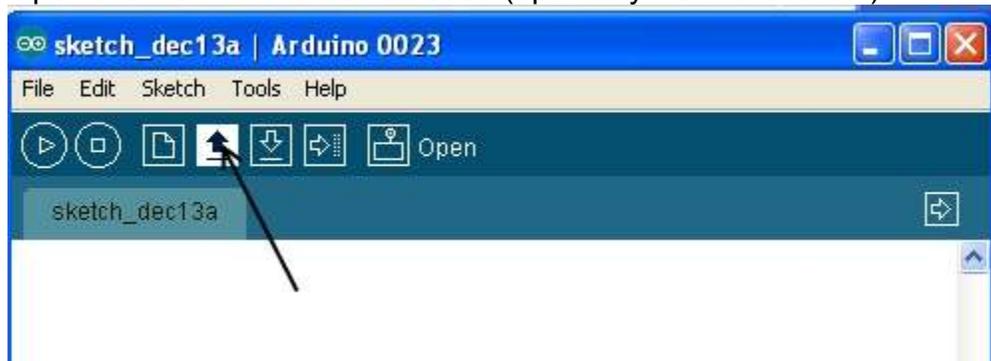
1. With FTDI USB connected, start Arduino.exe ( use Ver 1.0 for MultiWii code V2.0 or later, not 0023 as shown below)
2. Set Board type to :



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 2.0 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:
  - a. 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.
  - b. Choose the copter configuration
  - c. Reverse the yaw sense if needed  
Use `#define CRIUS_SE` or `#define CRIUS_LITE` , as appropriate
  - d. For Tricopter builds, your yaw servo may be reversed, so you need to uncomment: `#define YAW_DIRECTION -1` and comment out the default one.

For HEX6 builds, you must enable `#define A0_A1_PIN_HEX` and use pins A0 and A1 instead of D5 and D6 for motor connections, unless you are using a PPM-SUM receiver which only connects to the first RX pin on the board. See Other Neat Stuff at the end of this section.

Everything else is left as is....

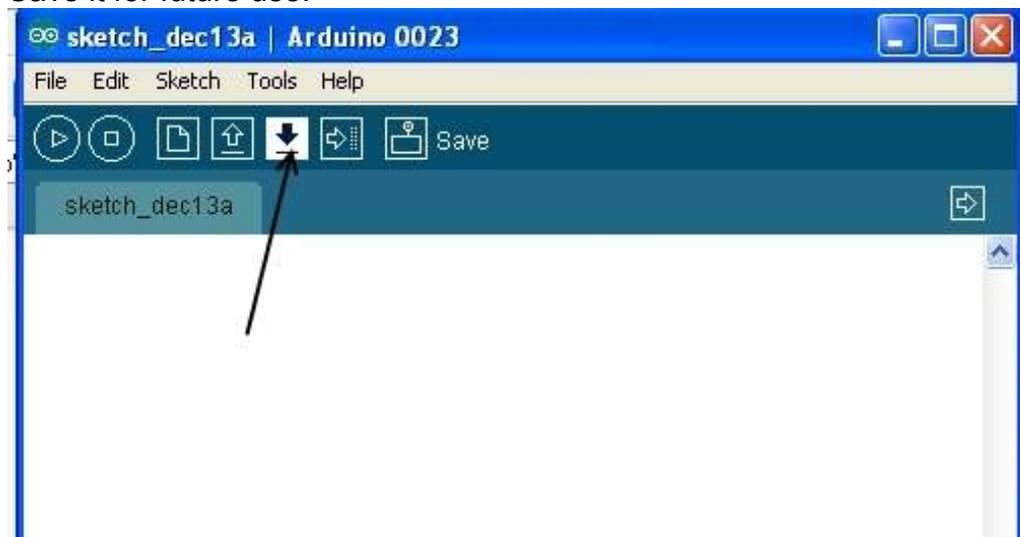
**Optional:**

If you set the throttle range for 995 to 2000, like I did, you will have to change two parameters:

```
#define MINCOMMAND 1000....I changed it to 980 to allow my  
ESCs to power up quicker  
#define MAXTHROTTLE 1850....I changed it 2000...otherwise I got  
only about 70% throttle at max...
```

**Note also for HEX6, you must not set MINCOMMAND below 1000 nor MAXTHROTTLE above 1910, or unreliable operation may result!**

6. Save it for future use.



7. Upload the code to board.



8. Configure and Go FLY!

## Other Neat Stuff

### LCD Configuration

It is possible to attach an LCD panel for field adjustment of PID values without the need for a PC. Stick movements are used to change values, while the LCD shows the variable being changed and by how much.

Search Ebay for “Multiwii LCD” to obtain one

Installation: just plug it into the TTL output pins beside the FTDI connector...be sure that you get power and gnd correctly oriented!

Operation: see the LCD section of this page: <http://www.multiwii.com/software>

### Bluetooth Configuration

Another option for field configuration without a PC is to use an Android smartphone and a Bluetooth module. The module connects to the TTL port pins as the LCD does, but uses all four pins since the communication is bi-directional in this case with the board.

The application is free. Here is a link to the app and sources for appropriate modules:

<http://www.androidpit.com/en/android/market/apps/app/net.xrotor.andmultiwiiconf/Android-MultiWii-Configuration>

Note that after you add this module, if you get a USB Bluetooth dongle for your PC you can remove the FTDI module since you can communicate with the PC over Bluetooth now instead of USB for code updates, configuration, etc.

Note: You cannot have both the BT module and FTDI active at the same time. Use one or the other at a time. You do not use the #BTSERIAL define! This define is for remote control of the copter using BT instead of a RC transmitter and uses a different phone app!

### Using RC receivers with a PPM composite output

Some RC receivers (like the FrSky and some Hitec) have an output in which all the channels are output in a single stream. This allows you to use only one servo wire connection between the RX and the MultiWii, eliminating all other connectons and cutting down on wiring mess. Attach it to the first RX input pin on the board, THROTTLE.

In order to use it, you must enable(uncomment) one of the *“#define SERIAL\_SUM\_PPM”* defines in the config.h file appropriate for your receiver. Recompile and download to the board (See previous section). In my case, it wasn't listed and I just experimented changing the order in the define until I got the correct channel order working.