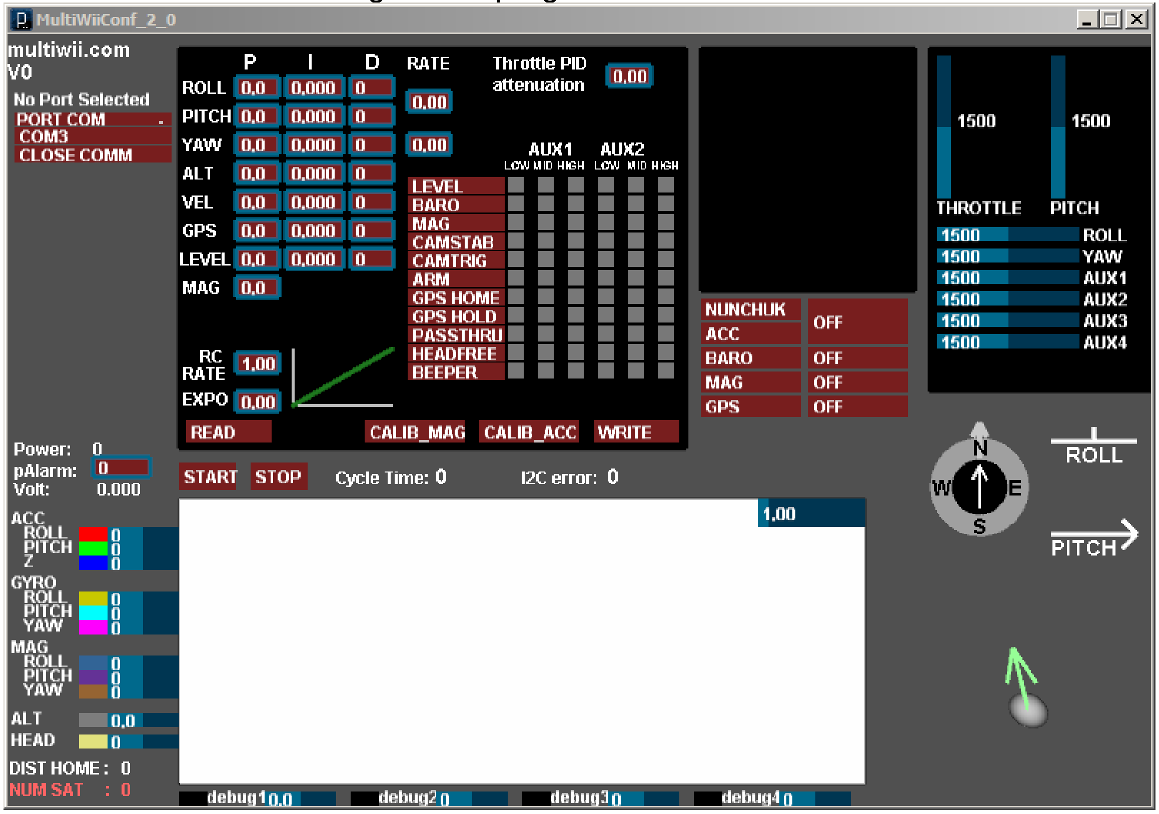
NEWBIES GUIDE to setting up and using the MultiWii Board Software

1. Download and unzip the required software:
-Arduino Compiler and Board Flashing Environment:
http://arduino.cc/en/Main/Software
-use the IDE 1.0 version
-Latest MultiWii Software including Configuration Program:
https://multiwii.googlecode.com/files/MultiWii_2_1.zip

2. 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.
http://www.ftdichip.com/Drivers/CDM/CDM20814_WHQL_Certified.zip

4. Start the MultiWii Configuration program.



	P	I	D	RATE
ROLL	0,0	0,000	0	0,00
PITCH	0,0	0,000	0	
YAW	0,0	0,000	0	0,00
ALT	0,0	0,000	0	
VEL	0,0	0,000	0	
GPS	0,0	0,000	0	
LEVEL	0,0	0,000	0	
MAG	0,0			

Throttle PID attenuation 0,00

RC RATE 1,00
EXPO 0,00

Power: 0
pAlarm: 0
Volt: 0.000

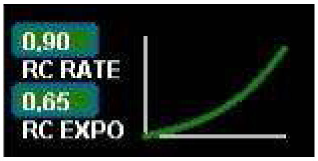
Cycle Time: 0
I2C error: 0

DIST HOME : 0
NUM SAT : 0

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):

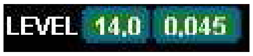
RC rate and RC expo



0,90
RC RATE
0,65
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 PITC 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 suported by the TX.
It's better to support them directly in Multiwii because the resolution is better this way.

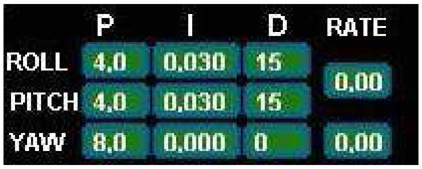
P and I Level



LEVEL 14,0 0,045

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.

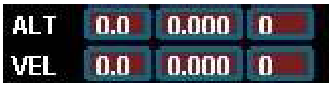
PITCH/ROLL/YAW PID and rate *(see tuning section later)*



	P	I	D	RATE
ROLL	4,0	0,030	15	0,00
PITCH	4,0	0,030	15	
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 and VEL PID



ALT	0.0	0.000	0
VEL	0.0	0.000	0

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 term deals with vertical velocity changes and helps to smooth out the altitude hold function. Think of VEL as trying to eliminate vertical movement, while ALT tries to hold the absolute altitude at the value it was set at. The values shown above are NOT the default values. Once activated altitude hold is adjusted by throttle: above neutral increases alttitude set point, below lowers it.

MAG

MAG 0.0

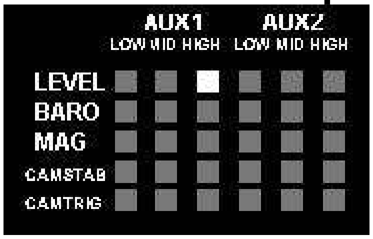
This is the P term for the compass heading hold mode. Once this mode is activated, the compass heading will be held as long as
rudder stick is at neutral. Changing the rudder stick will change the heading set point.

THROTTLE rate

Throttle PID attenuation 0.00

This boxe 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)

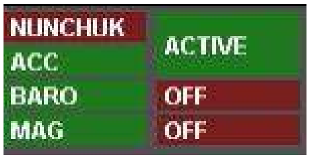
How to activate options



	AUX1			AUX2		
	LOW	MID	HIGH	LOW	MID	HIGH
LEVEL	□	□	■	□	□	□
BARO	□	□	□	□	□	□
MAG	□	□	□	□	□	□
CAMSTAB	□	□	□	□	□	□
CAMTRIG	□	□	□	□	□	□

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)

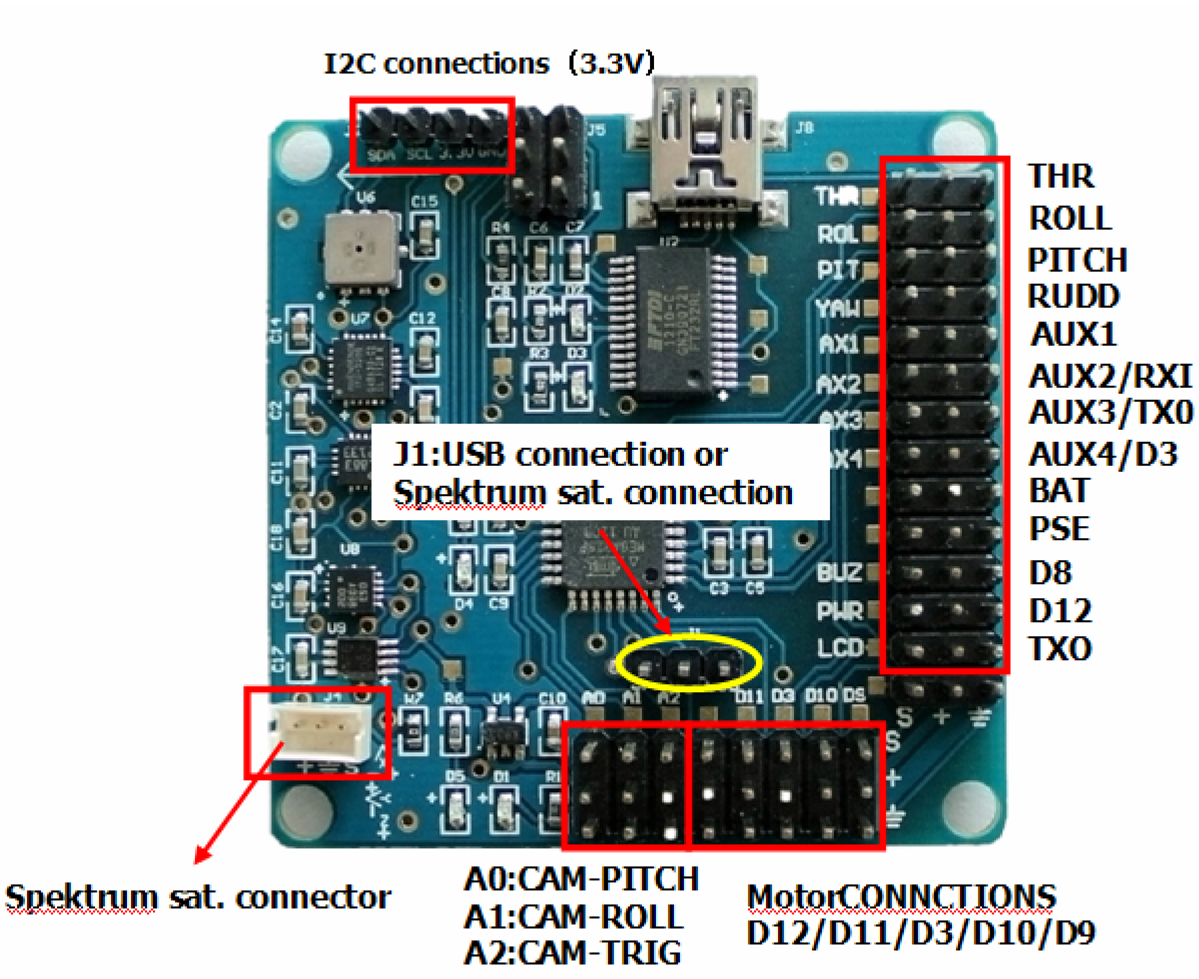
Sensor detection / usage



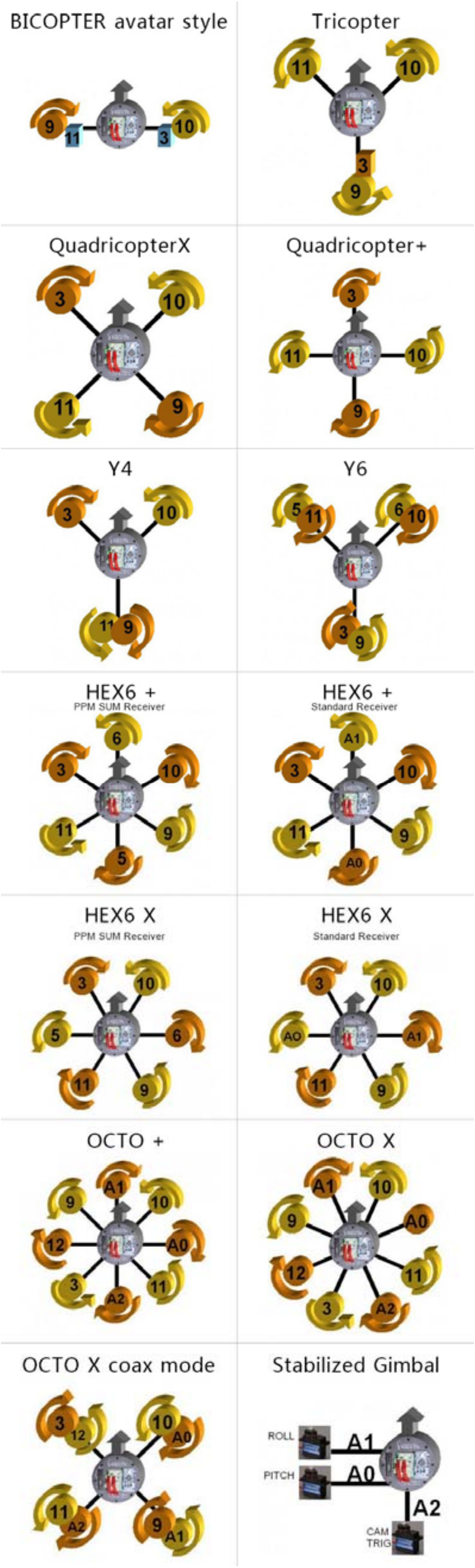
NUNCHUK	ACTIVE
ACC	
BARO	OFF
MAG	OFF

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:



Motor connections and rotations are shown below:



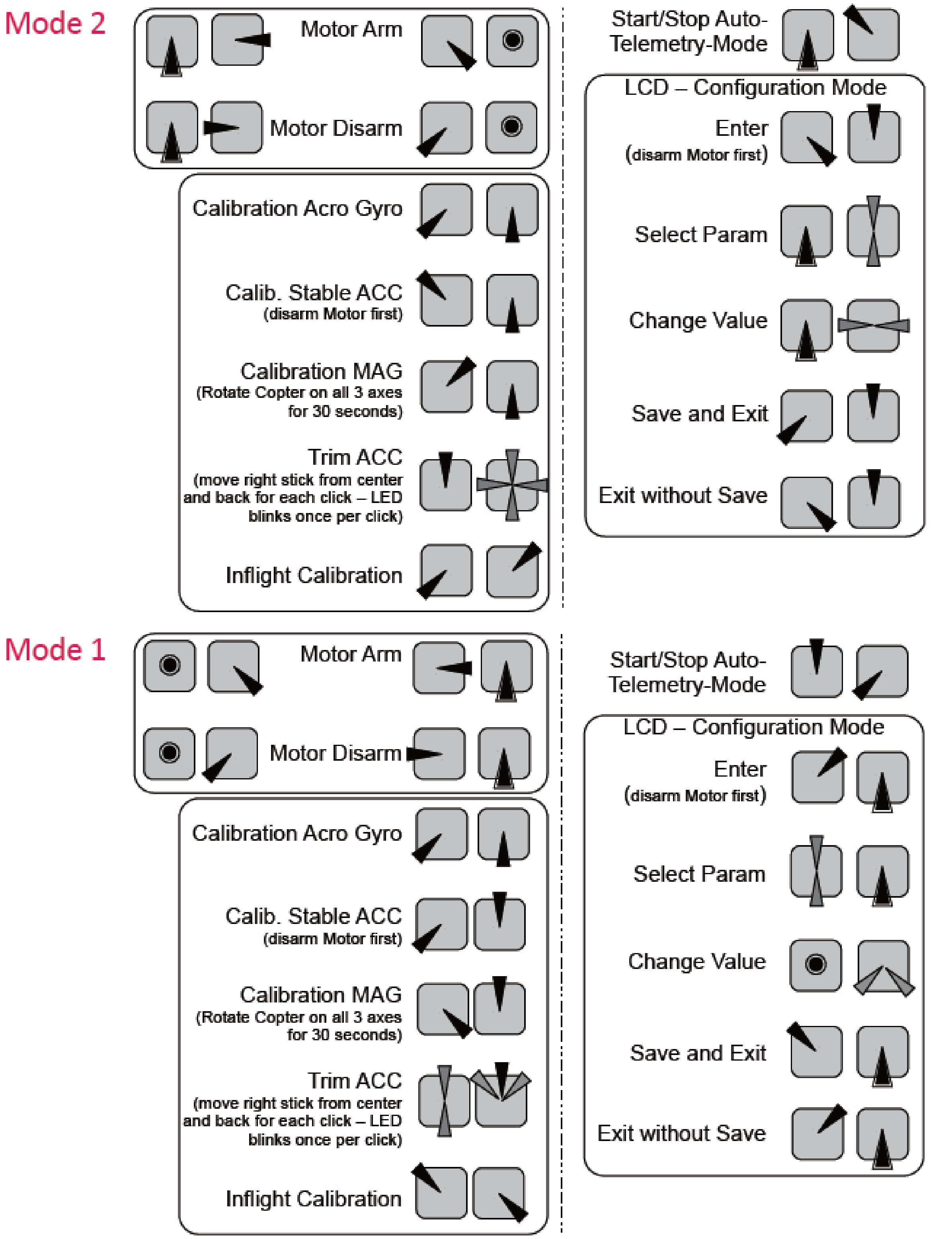
Numbers refer to pins D3,9,10,11,A0,and A1 the Motor1-6 ESC outputs.
Note: if the ESC uses a LINEAR BEC, you can connect all to the motor outputs without modification. If the BECs are switching types, you should use only one BEC output and remove the red wires from the other ESC connectors.
Alternatively, all can be removed and an External BEC connected to any unused three pin connection.

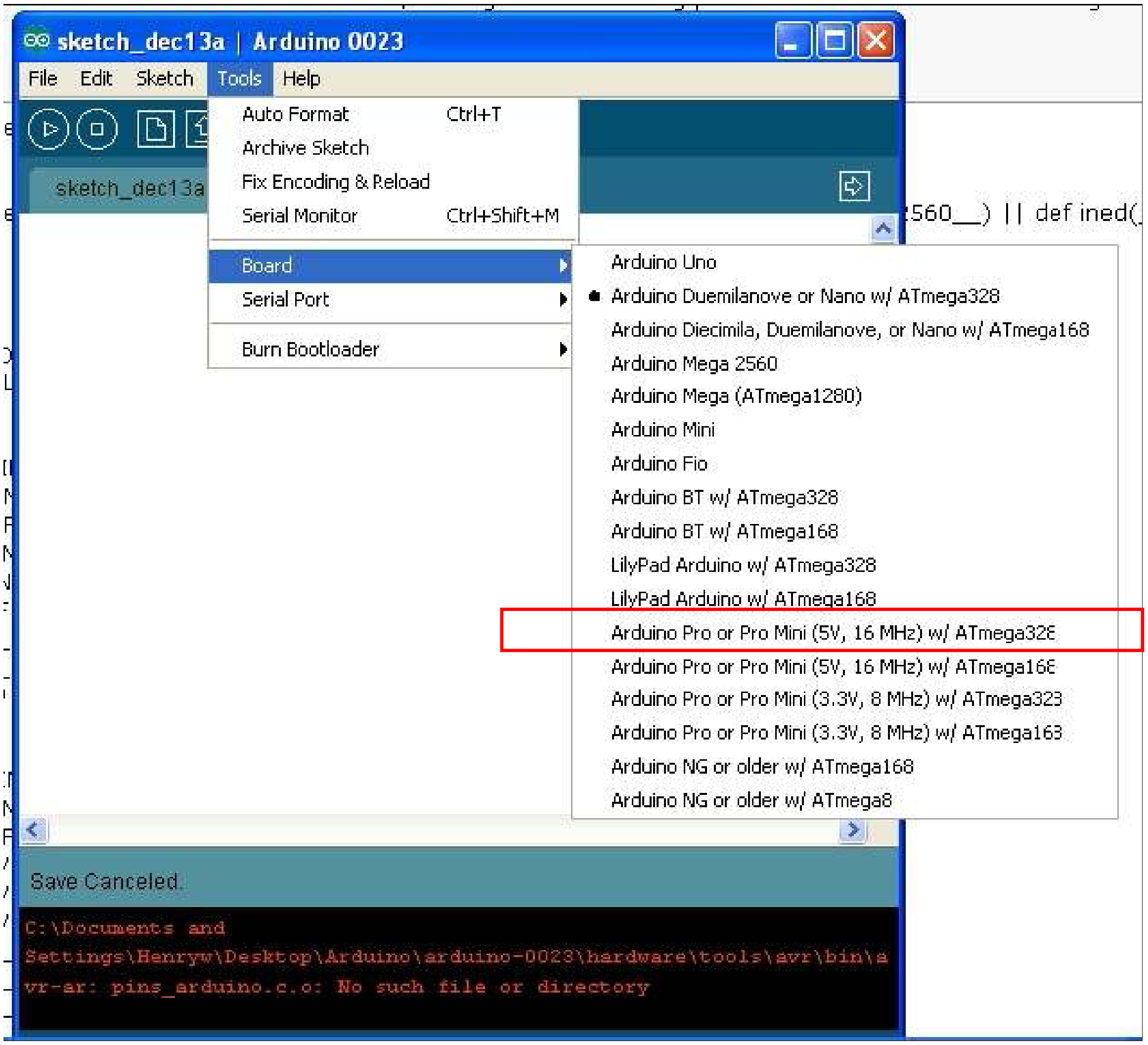
1. BALANCE PROPS and MOTORS!
2. Check that all TX channels are centered at 1500 and run from 1000 to 2000.
3. Check that all channels are correct direction...ie right runs right on roll, etc., if not reverse them.
4. Calibrate Gyro (board not moving!)
5. Calibrate Accelerometers (board level!)
6. Calibrate 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 eprom.
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.

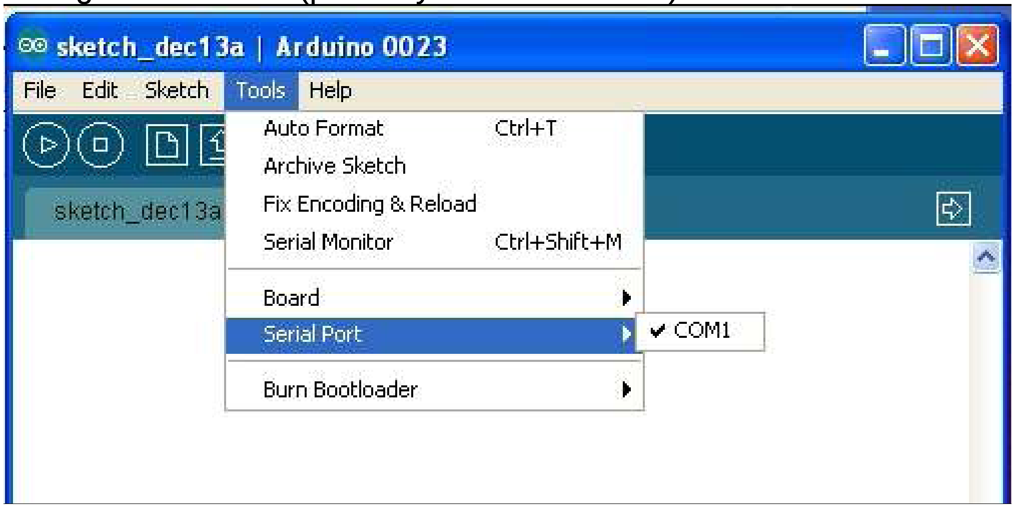
Control Stick Configuration Guide (valid for V2.0 also) :
http://multiwii.googlecode.com/svn/branches/Hamburger/MultiWii-StickConfiguration-18_v0-57721.pdf



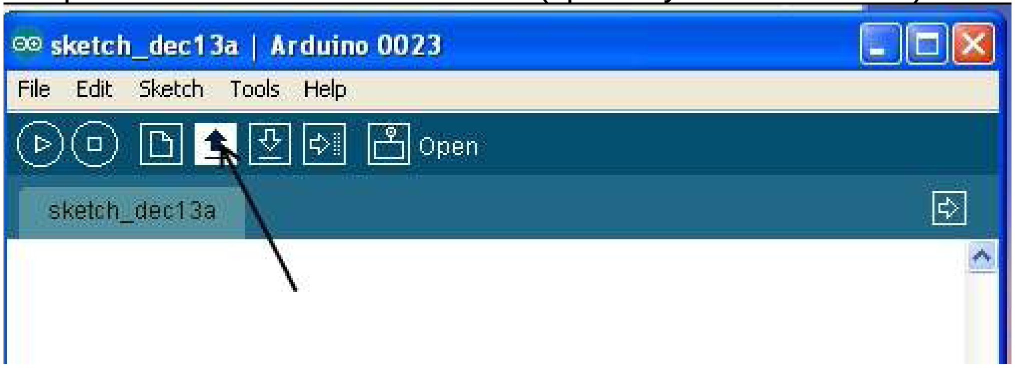

sketch_dec13a | Arduino 0023
File
Edit
Sketch
Tools
Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Burn Bootloader
sketch_dec13a
Arduino Uno
● Arduino Duemilanove or Nano w/ ATmega328
Arduino Diecimila, Duemilanove, or Nano w/ ATmega168
Arduino Mega 2560
Arduino Mega (ATmega1280)
Arduino Mini
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
560__) || def ined(

```
Save Canceled.
C:\Documents and
Settings\Henryw\Desktop\Arduino\arduino-0023\hardware\tools\avr\bin\a
vr-ar: pins_arduino.c.o: No such file or directory
```

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

d. Use : #define I2C_SPEED 400000L

e. Use: #define INTERNAL_I2C_PULLUPS

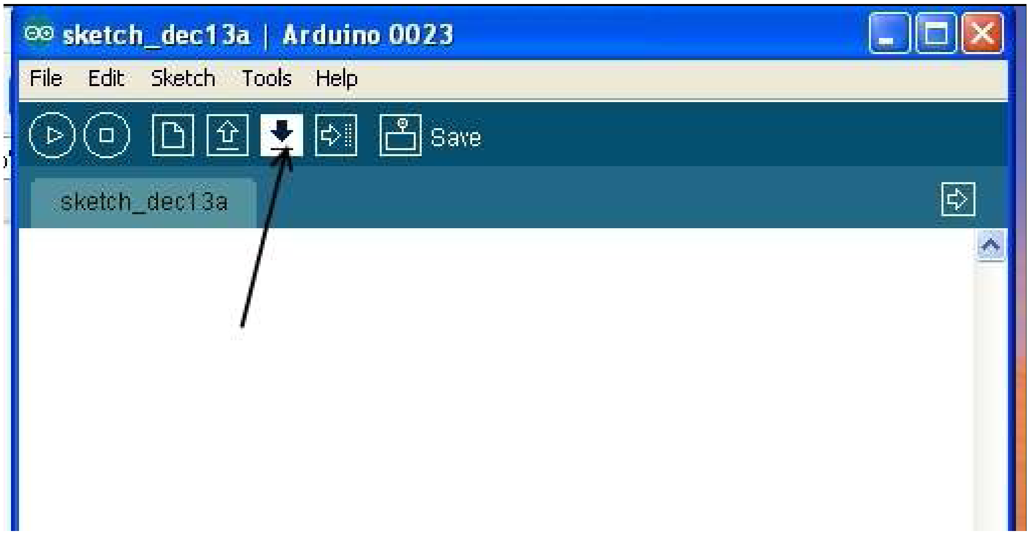
f. Use: #define RCAUXPIN8 or #define RCAUXPIN12 if you want to use AUX2 for mode selection in addition to AUX1

g. Use: #define FREEIMUv035_BMP

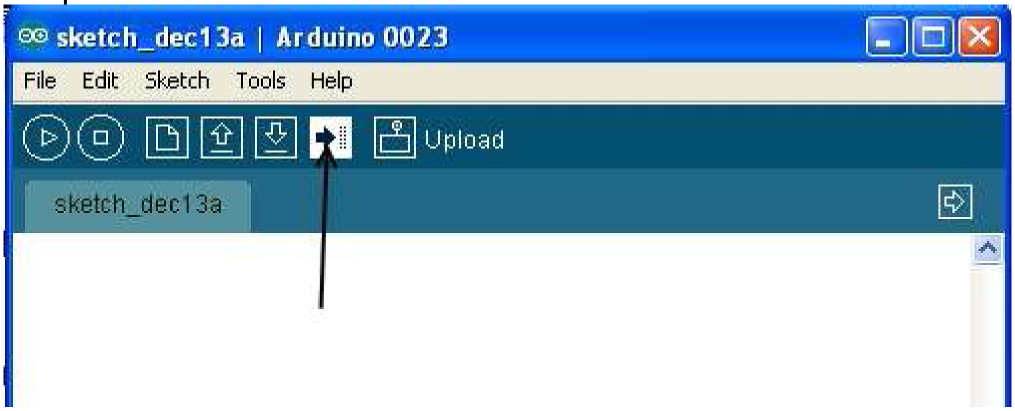
h. Use: #define MAG_ORIENTATION(X, Y, Z) {magADC[ROLL] = -Y; magADC[PITCH] = X; magADC[YAW] = Z;} note this has to be entered manually, not already there but commented out…else compass heading is off 90deg

Everything else is left as is….

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.

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 until I got the correct channel order working.