Step By Step

Compiling MegaPirateNG and Multiwii

by

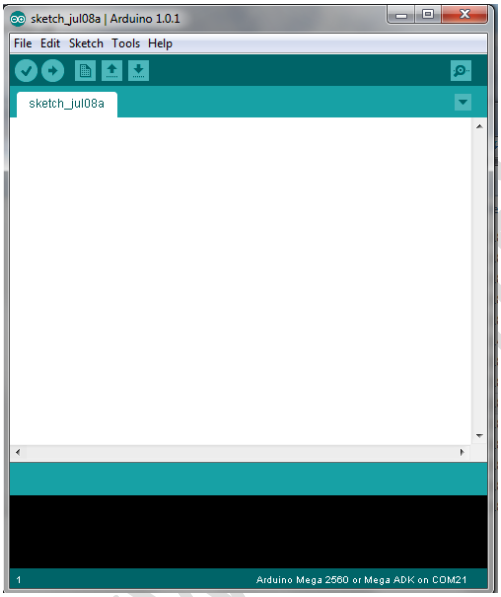
Quadframes.co.uk

Some of the programs that you will need and were to get them.

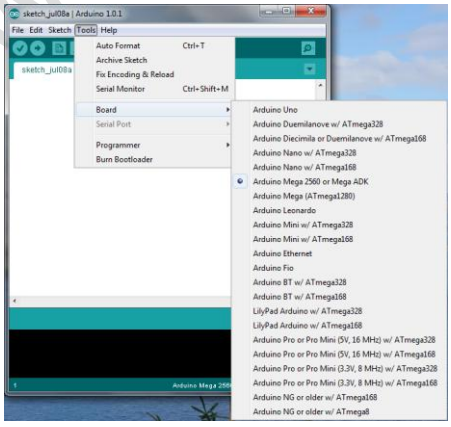
1. Arduino you can download from here
2. MegaPirateNG you can download from here
3. MultiWii you can download from here

These are the programs that you will need if using MegaPirateNG or MultiWii on your CRIUS ALL IN ONE PRO.

once you have downloaded and installed Arduino when you open it will look like this

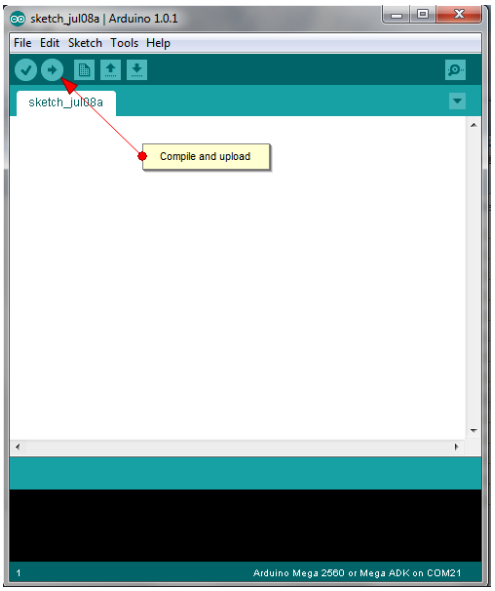


please make sure that you have selected the correct board type as shown here under *Tools-Boards*

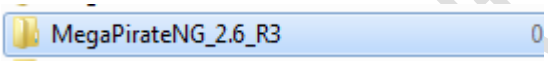


Now select the serial port this is normally selected automatically for you.

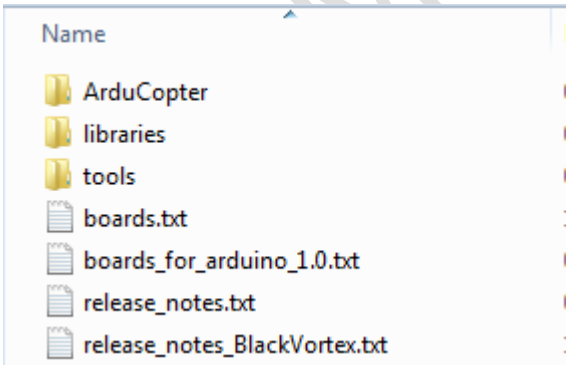
Now the compile button, when you have finished the edits to the code this is the button to click to get you code compiled and uploaded to you board



Ok now that we have got Arduino sorted let's move onto Compiling MegaPirateNG. The way that I do this is make a folder called MegaPirateNG download the latest copy of MegaPirateNG into this folder, next unzip MegaPirateNG_ 2.6 r3 and you will have a folder like this

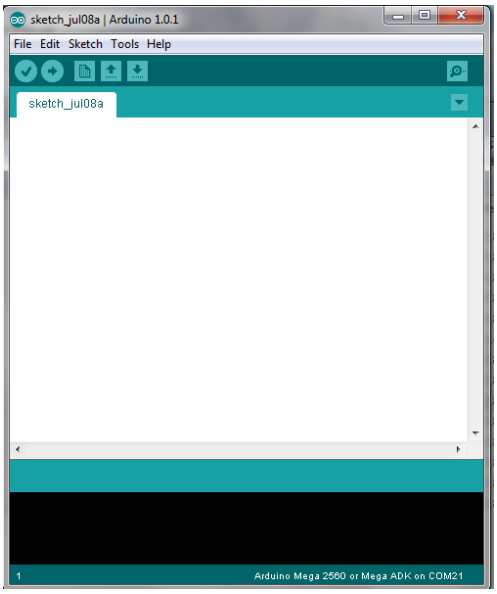


inside this folder looks like this

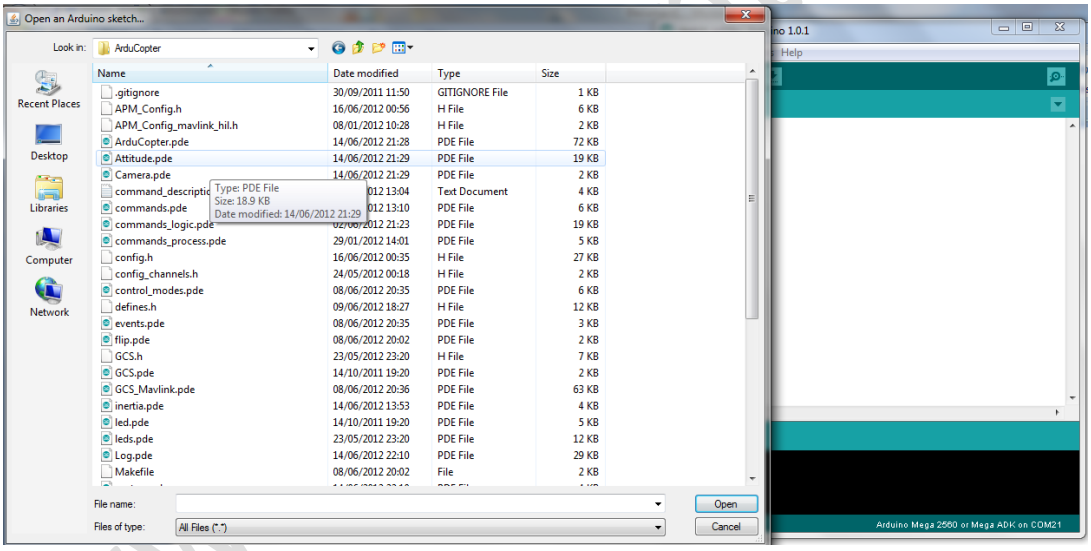


ok now we have to select the Libraries folder and copy this to the folder that we have Arduino in replacing the files when asked. Now that we have done this we can now move onto the editing the code so that we can upload to the board.

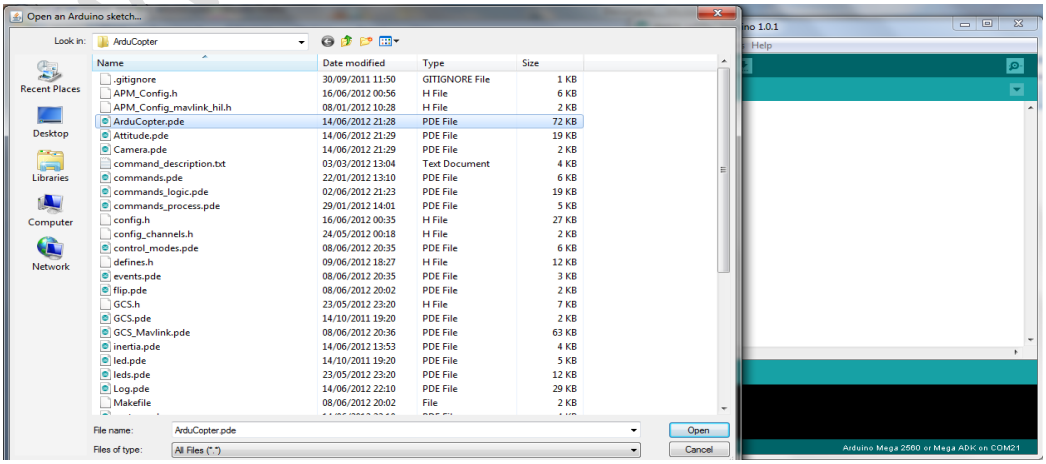
Open Arduino



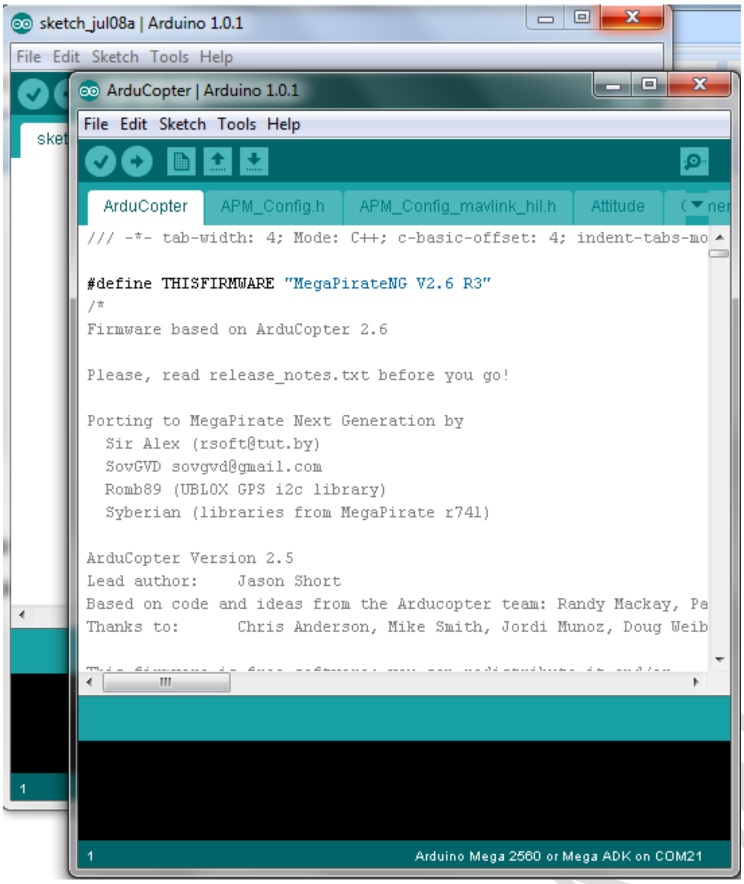
then go to file open and direct it to open the MegaPirateNG as seen here



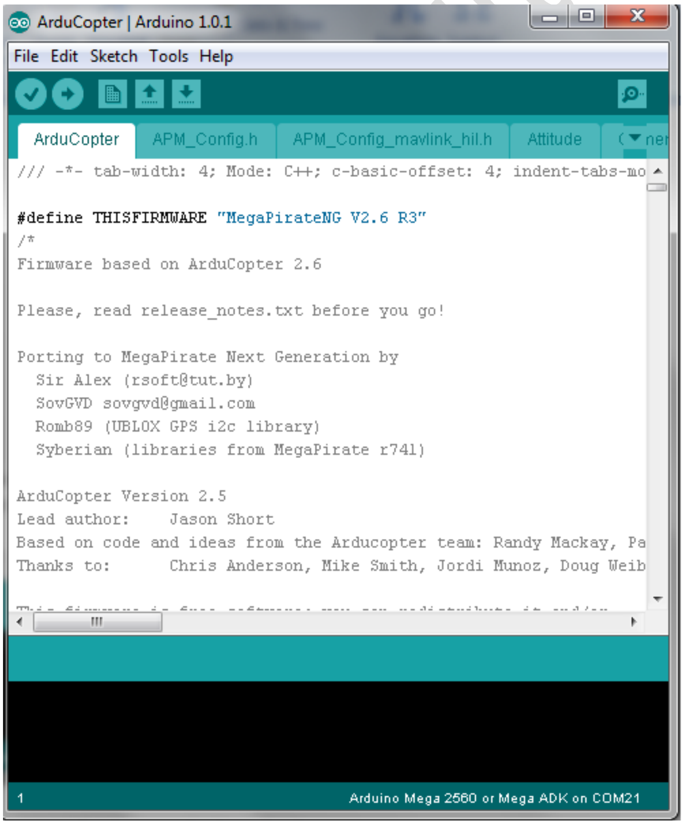
now select the ArduCopter.pde file as see here.



you should now have this screen



close the Arduino behind so that you just have the one with the code loaded.



you will notice that there are tab's along the top the only one that we have to make changes to is the APM_Config.h and as you can see buy the picture above that this is the second tab along just click on this to edit it.

These are the parts of the code that we will need to change (sensor board)

```
// Select your sensor board
#define PIRATES_SENSOR_BOARD PIRATES_ALLINONE
/*
        PIRATES_ALLINONE
        PIRATES_FFIMU
        PIRATES_FREEIMU
        PIRATES_BLACKVORTEX
        PIRATES_FREEIMU_4                                   // New FreeIMU 0.4.1 with MPU6000, MS5611 and 5883L
        PIRATES_DROTEK_10DOF_MPU                // MPU6000, MS5611 and 5883L
*/
```

edit to look like this

```
// Select your sensor board
#define PIRATES_SENSOR_BOARD PIRATES_FREEIMU_4
/*
        PIRATES_ALLINONE
        PIRATES_FFIMU
        PIRATES_FREEIMU
        PIRATES_BLACKVORTEX
        PIRATES_FREEIMU_4                                   // New FreeIMU 0.4.1 with MPU6000, MS5611 and 5883L
        PIRATES_DROTEK_10DOF_MPU                // MPU6000, MS5611 and 5883L
*/
```

next is to change the (baro sensor)

```
// Select your baro sensor
#define CONFIG_BARO AP_BARO_BMP085_PIRATES
/*
        AP_BARO_BMP085_PIRATES
        AP_BARO_MS5611_I2C
*/
```

edit to look like this

```
// Select your baro sensor
#define CONFIG_BARO AP_BARO_MS5611_I2C
/*
        AP_BARO_BMP085_PIRATES
        AP_BARO_MS5611_I2C
*/
```

next is the code changes for if you are using a GPS

```
// For BlackVortex, just set PIRATES_SENSOR_BOARD as PIRATES_BLACKVORTEX, GPS will select automatically
#define GPS_PROTOCOL GPS_PROTOCOL_NONE
/*
        GPS_PROTOCOL_NONE       without GPS
        GPS_PROTOCOL_NMEA
        GPS_PROTOCOL_SIRF
        GPS_PROTOCOL_UBLOX
        GPS_PROTOCOL_IMU
        GPS_PROTOCOL_MTK
        GPS_PROTOCOL_HIL
        GPS_PROTOCOL_MTK16
        GPS_PROTOCOL_AUTO       auto select GPS
        GPS_PROTOCOL_UBLOX_I2C
        GPS_PROTOCOL_BLACKVORTEX
*/
```

if you are not sure what to do here always try this one 1st

```
// For BlackVortex, just set PIRATES_SENSOR_BOARD as PIRATES_BLACKVORTEX, GPS will select automatically
#define GPS_PROTOCOL GPS_PROTOCOL_AUTO
/*
        GPS_PROTOCOL_NONE       without GPS
        GPS_PROTOCOL_NMEA
        GPS_PROTOCOL_SIRF
        GPS_PROTOCOL_UBLOX
        GPS_PROTOCOL_IMU
        GPS_PROTOCOL_MTK
        GPS_PROTOCOL_HIL
        GPS_PROTOCOL_MTK16
        GPS_PROTOCOL_AUTO       auto select GPS
        GPS_PROTOCOL_UBLOX_I2C
        GPS_PROTOCOL_BLACKVORTEX
*/
```

if you know what GPS you are using then make the changes to suit your GPS.

Next changes are the frame config as seen here

```
#define FRAME_CONFIG QUAD_FRAME
/*
        QUAD_FRAME
        TRI_FRAME
        HEXA_FRAME
        Y6_FRAME
        OCTA_FRAME
        OCTA_QUAD_FRAME
        HELI_FRAME
*/

#define FRAME_ORIENTATION X_FRAME
/*
        PLUS_FRAME
        X_FRAME
        V_FRAME
*/

# define CH7_OPTION              CH7_DO_NOTHING
```

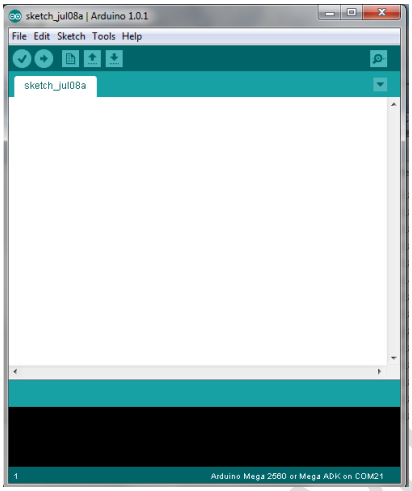
as seen in the picture above the default is QUAD_FAME and its Orientation is X_FRAME if you what a diff just change the code to suit, I hope that this helps in understanding the code a bit more.

If using Multiwii code

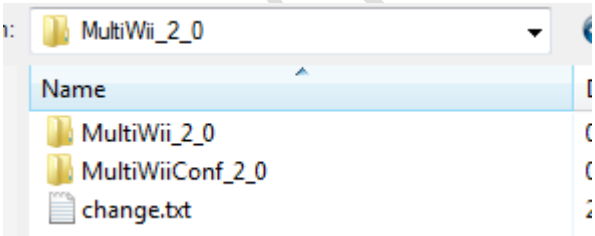
Code changes and compiling Multiwii

download Multiwii from the link above and unzip

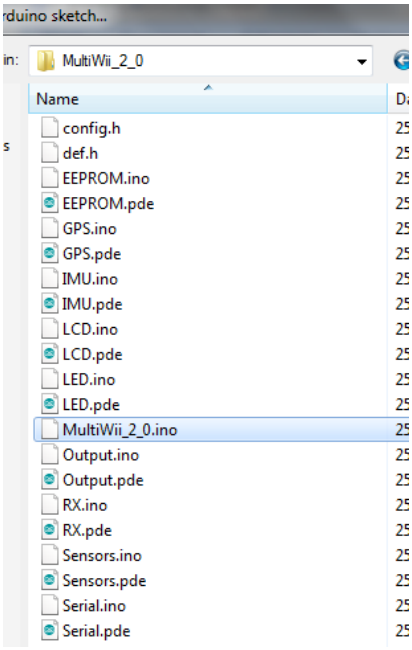
Open Arduino



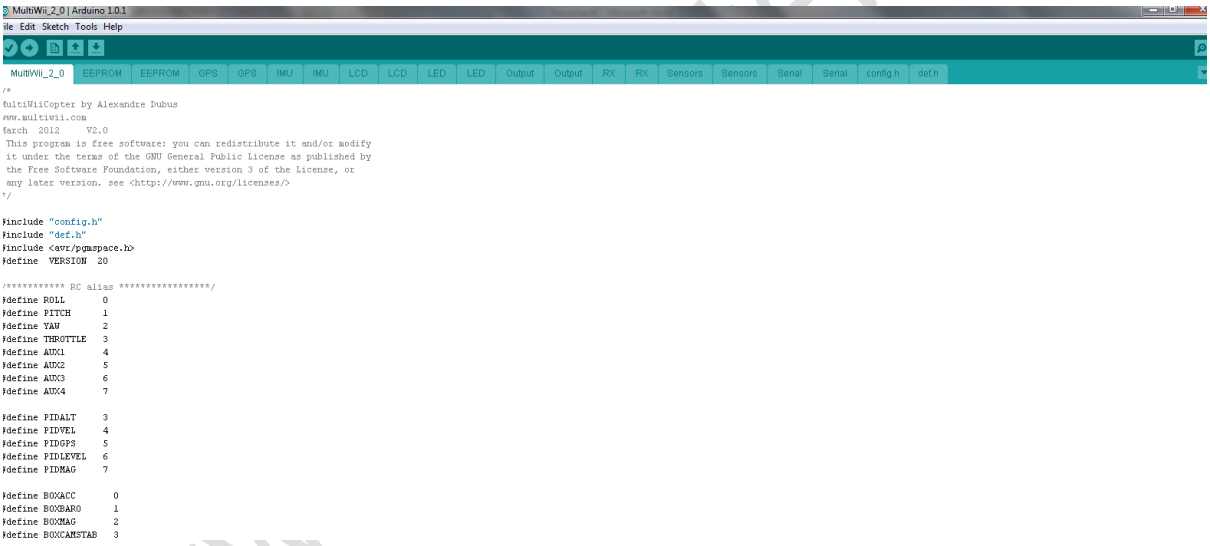
Then file open, and select the folder that you have unzipped your Multiwii as seen here



select MultiWii_2_0 and you will see this, please select the file called MultiWii_2_0.ino (your version could be different)



close the opened Arduino and then you should have the following



please note the tabs along the top. 1st change that we are going to make are in the def.h tab the last one at the top on the right. when opended you need to scroll down until you see this

```
//please submit any correction to this list.
#if defined(FFIMUv1)
  #define ITG3200
  #define BMA180
  #define BMP085
  #define HMC5843
  #define ACC_ORIENTATION(X, Y, Z)  {accADC[ROLL]  = -X; accADC[PITCH]  = -Y; accADC[YAW]  =  Z;}
  #define GYRO_ORIENTATION(X, Y, Z) {gyroADC[ROLL] =  Y; gyroADC[PITCH] = -X; gyroADC[YAW] = -Z;}
  #define MAG_ORIENTATION(X, Y, Z)  {magADC[ROLL]  =  X; magADC[PITCH]  =  Y; magADC[YAW]  = -Z;}
  #define BMA180_ADDRESS 0x80
  #define ITG3200_ADDRESS 0XD0
#endif

#if defined(FFIMUv2)
  #define ITG3200
  #define BMA180
  #define BMP085
  #define HMC5883
  #define ACC_ORIENTATION(X, Y, Z)  {accADC[ROLL]  = -X; accADC[PITCH]  = -Y; accADC[YAW]  =  Z;}
  #define GYRO_ORIENTATION(X, Y, Z) {gyroADC[ROLL] =  Y; gyroADC[PITCH] = -X; gyroADC[YAW] = -Z;}
  #define MAG_ORIENTATION(X, Y, Z)  {magADC[ROLL]  =  Y; magADC[PITCH]  = -X; magADC[YAW]  = -Z;}
  #define BMA180_ADDRESS 0x80
```

you need to add this code to under ***//please submit any correction to this list.*** to look like this

```
//please submit any correction to this list.
#if defined(CRIUS_AIO_PRO_V1)
#define MPU6050
#define HMC5883
#define MS561101BA
#define ACC_ORIENTATION(X, Y, Z) {accADC[ROLL] = -X; accADC[PITCH] = -Y; accADC[YAW] = Z;}
#define GYRO_ORIENTATION(X, Y, Z) {gyroADC[ROLL] = Y; gyroADC[PITCH] = -X; gyroADC[YAW] = -Z;}
#define MAG_ORIENTATION(X, Y, Z) {magADC[ROLL] = X; magADC[PITCH] = Y; magADC[YAW] = -Z;}
#define MPU6050_EN_I2C_BYPASS // MAG connected to the AUX I2C bus of MPU6050
#undef INTERNAL_I2C_PULLUPS
#endif

#if defined(FFIMUv1)
  #define ITG3200
  #define BMA180
  #define BMP085
  #define HMC5843
  #define ACC_ORIENTATION(X, Y, Z)  {accADC[ROLL]  = -X; accADC[PITCH]  = -Y; accADC[YAW]  =  Z;}
  #define GYRO_ORIENTATION(X, Y, Z) {gyroADC[ROLL] =  Y; gyroADC[PITCH] = -X; gyroADC[YAW] = -Z;}
  #define MAG_ORIENTATION(X, Y, Z)  {magADC[ROLL]  =  X; magADC[PITCH]  =  Y; magADC[YAW]  = -Z;}
  #define BMA180_ADDRESS 0x80
  #define ITG3200_ADDRESS 0xD0
```

Now we need to open the config.h tab and add the following code *#define CRIUS_AIO_PRO_V1 // Crius Multiwii AIO PRO v1.0*

Please add this to the bottom of this list under *//#define CRIUS_SE // Crius MultiWii SE*

```
//#define ALLINONE        // full FC board or standalone 9DOF+baro board from CSG_EU
//#define AEROQUADSHIELDv2
//#define ATAVRSBIN1      // Atmel 9DOF (Contribution by EOSBandi). requires 3.3V power.
//#define SIRIUS          // Sirius Navigator IMU                                    <- confirmed by Alex
//#define SIRIUS600       // Sirius Navigator IMU  using the WMP for the gyro
//#define MINIWII         // Jussi's MiniWii Flight Controller
//#define CITRUSv2_1      // CITRUS from qcrc.ca
//#define CHERRY6DOFv1_0
//#define DROTEK_10DOF    // Drotek 10DOF with ITG3200, BMA180, HMC5883, BMP085, w or w/o LLC
//#define DROTEK_10DOF_MS // Drotek 10DOF with ITG3200, BMA180, HMC5883, MS5611, LLC
//#define DROTEK_6DOFv2   // Drotek 6DOF v2
//#define DROTEK_6DOF_MPU // Drotek 6DOF with MPU6050
//#define MONGOOSE1_0     // mongoose 1.0    http://www.fuzzydrone.org/
//#define CRIUS_LITE      // Crius MultiWii Lite
//#define CRIUS_SE        // Crius MultiWii SE
```

as seen here

```
//#define ATAVRSBIN1      // Atmel 9DOF (Contribution by EOSBandi). requires 3.3V power.
//#define SIRIUS          // Sirius Navigator IMU                                    <- confirmed by Alex
//#define SIRIUS600       // Sirius Navigator IMU  using the WMP for the gyro
//#define MINIWII         // Jussi's MiniWii Flight Controller
//#define CITRUSv2_1      // CITRUS from qcrc.ca
//#define CHERRY6DOFv1_0
//#define DROTEK_10DOF    // Drotek 10DOF with ITG3200, BMA180, HMC5883, BMP085, w or w/o LLC
//#define DROTEK_10DOF_MS // Drotek 10DOF with ITG3200, BMA180, HMC5883, MS5611, LLC
//#define DROTEK_6DOFv2   // Drotek 6DOF v2
//#define DROTEK_6DOF_MPU // Drotek 6DOF with MPU6050
//#define MONGOOSE1_0     // mongoose 1.0    http://www.fuzzydrone.org/
//#define CRIUS_LITE      // Crius MultiWii Lite
//#define CRIUS_SE        // Crius MultiWii SE
#define CRIUS_AIO_PRO_V1 // Crius Multiwii AIO PRO v1.0
```

all done.

now you just have to make the changes to the code to suit your board config ect,ect.