# CRIUS ALL IN ONE PRO v1.0 Multi Rotor Flight Controller

## Manual rev 1.3

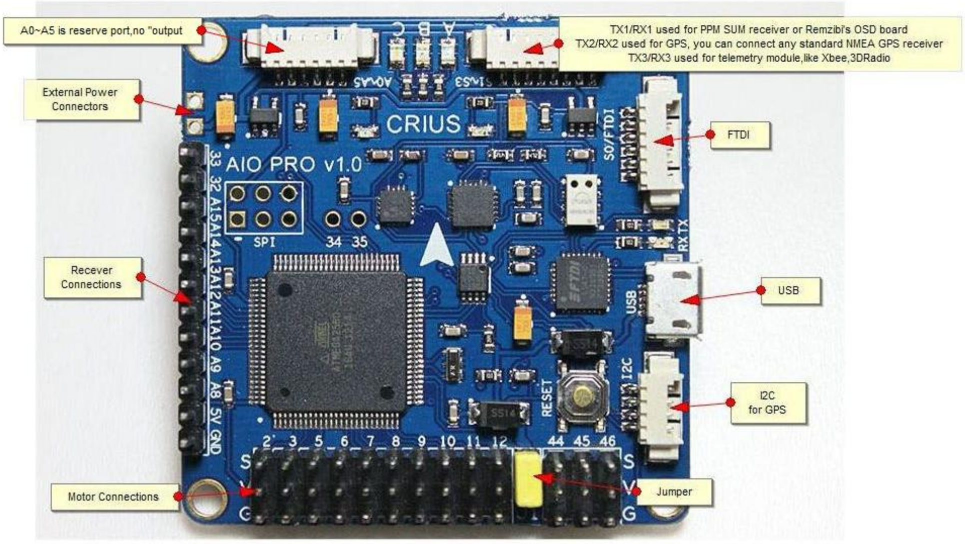
## By Quadframes

**Features:**
·Supported MegaPirateNG and MultiWii firmware
·Up to 8-axis motor output
·8 input channels for standard receiver
·4 serial ports for debug/Bluetooth Module/OSD/GPS/telemetry
·2 servos output for PITCH and ROLL gimbal system
·A servos output to trigger a camera button
·6 Analog output for extend device
·A I2C port for extend sensor or device
·Separate 3.3V and 5V LDO voltage regulator
·ATMega 2560 Microcontroller
·MPU6050 6 axis gyro/accel with Motion Processing Unit
·HMC5883L 3-axis digital magnetometer
·MS5611-01BA01 high precision altimeter
·FT232RQ USB-UART chip and Micro USB receptacle
·On board logic level converter
·Match the standard of RoHS

**Flight mode for Multiwii**
·One of the following basic mode
- Acro
- Level
- Alt Hold
- Head Lock

·Optional mode
- HeadFree (CareFree)
- GPS Hold (Need GPS receiver or Extend Board)
- GPS Back to home position (Need GPS receiver or Extend Board)

**Flight mode for MegaPirate**
·Acro
·Alt Hold
·Simple
·Loiter (uses GPS)
·Guided (uses GPS)
·Position (uses GPS)
·Circle (uses GPS)
·RTL (uses GPS)
·Auto(uses GPS)
- Follow Me(uses GPS)


A0~A5 is reserve port,no "output
TX1/RX1 used for PPM SUM receiver or Remzibi's OSD board
TX2/RX2 used for GPS, you can connect any standard NMEA GPS receiver
TX3/RX3 used for telemetry module,like Xbee,3DRadio
External Power Connectors
FTDI
Recever Connections
USB
I2C for GPS
Motor Connections
Jumper
CRIUS
AIO PRO v1.0
SPI
34 35
S0/FTDI
USB
I2C
RESET
RX TX
33 32 A15 A14 A13 A12 A11 A10 A9 A8 5V GND
2 3 5 6 7 8 9 10 11 12
44 45 46
A0~A5
S1~S3
A B C
S V G

**Motor connection**

-MultiWii : D2/D3/D5/D6/D7/D8/D9/D10
-MegaPirateNG: D2/D3/D5/D6/D7/D8/D11/D12

External Power: if you don't power it by the BEC from the ESC Also needed to power Expansion Board.

-cam pitch = input from RX
-cam roll = same
-aux4= same (aux port 1-4 u can see in the MultiWii config tool on PC)
-aux3= same (aux port 1-4 u can see in the MultiWii config tool on PC)
-aux2= same (aux port 1-4 u can see in the MultiWii config tool on PC)
-aux1= same (aux port 1-4 u can see in the MultiWii config tool on PC)
-yaw = input from RX
-pitch = input from RX
-roll = input from RX
-throttle = input from RX
-5v = so the RX gets power
-gnd= ground

-trigger 9 = to trigger the cam if needed
-echo 10 = optional echo ultrasonic sensor (??)
-2x Motor out 11, 12 (where do i put which motor????)
-5v bridge, jumper (if you power it by the BEC from the ESC this must be closes)
-pitch 44 = for the CAM Gimbal system
-roll 45 = for the CAM Gimbal system
-trigger 46 = another trigger
D9/D10 is sonar sensor port when you use MegaPirateNG firmware

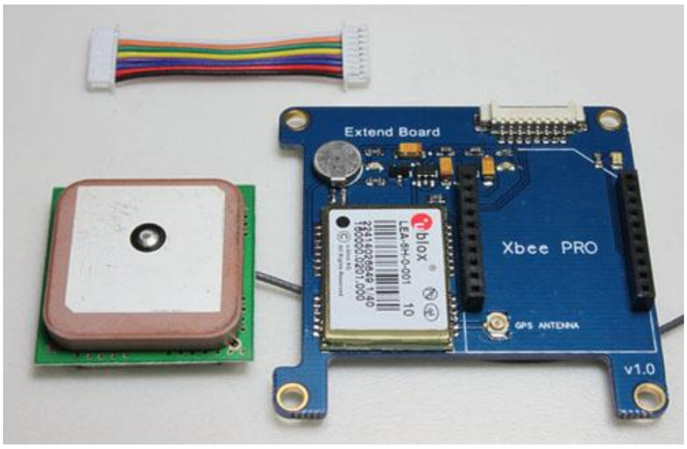
**I2C port**
(slc, sda, 5v, gnd) = this is the for expanding like Gps.

**S0/FTDI**
(gnd, gnd, 5v, rx0, tx0, dtr) = can be connected a bluetooth module or standalone FTDI cable(if the USB on board is broken)

**S1-S3**
(tx1, rx1, tx2, rx2 tx3, rx3, vcc, gnd) = RX1/TX1 used for PPM SUM receiver or Remzibi's OSD board
TX2/RX2 used for GPS, you can connect any standard NMEA GPS receiver
TX3/RX3 used for telemetry module,like Xbee,3DRadio etc

**A0-A5**
A0~A5 is reserve port,no "output"

# Extend Board for ALL IN ONE PRO



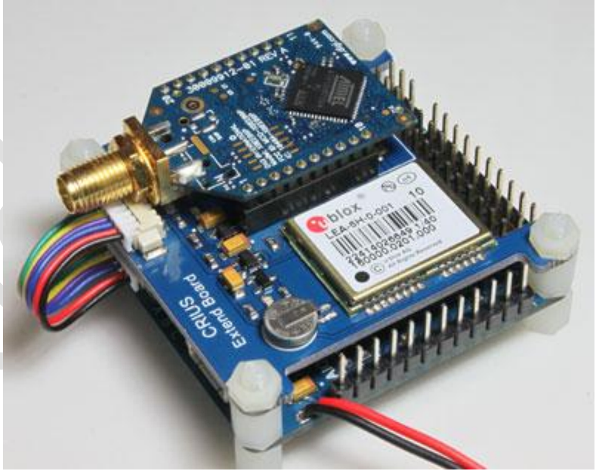
**Features:**

·Included U-blox LEA-5H GPS Module and Xbee socket. ( Been advise this will now be LEA-6H GPS module)

·Working with AIO PRO FC, provides full UAV functionality.

·2 separated 3.3V LDO voltage regulator

·25X25mm active GPS antenna with 200mm cable



Extend Board connects to S1-S3 port

**Please note:**

The extended board must be powered via external power pins on the main board and the yellow jumper is to be removed.

# Software code changes for Multiwii and Megapirate

## Multiwii

Add this code to def.h in MultiWii 2.0 or later verson
after

```
/**************************************************************************************/
/***************            IMU Orientations and Sensor definitions            ********************/
/**************************************************************************************/

//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
```

*then add this to "boards and sensor definitions" in config.h*

```
#define CRIUS_AIO_PRO_V1 // Crius Multiwii AIO PRO v1.0
```

For the GPS, scroll down to the GPS section in config.h and set the following:

```
#define GPS_SERIAL 2
#define GPS_BAUD 115200
```

## Megapirate

These settings are all in APM_Config.h you will see this tab when you load the Attitude.pde file in the MegaPirateNG/ArduCopter folder.

```
// 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
*/


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



// 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
*/
```

# Crius All In One GPS and Installation

By fpvcentral on Jul 10, 2012

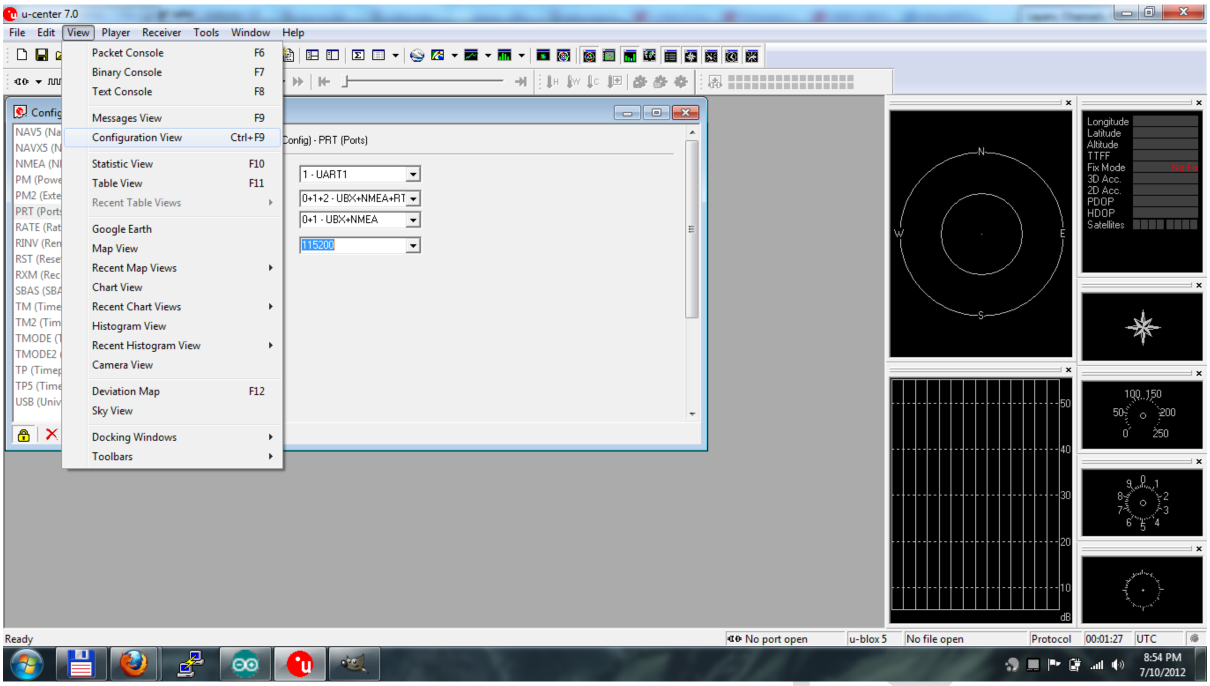
## Configuring the GPS

The first order of business is to configure the u-box based GPS module from its default 9600 baud serial speed and  1Hz update rate to a more useful 115200 baud at 10 Hz update rate. To do this, you will need to connect the GPS module to a USB-to-Serial adapter, for example the little USB FTDI interface we used to set up the Crius Lite. Connect GND to GND, 5V to 5V, RX to TX and TX to RX (obviously).

After that, download the u-blox configuration software at http://www.u-blox.com/images/Support/Support_Products/EvaluationSoftware/u-centersetup-6.2.0.0.zip

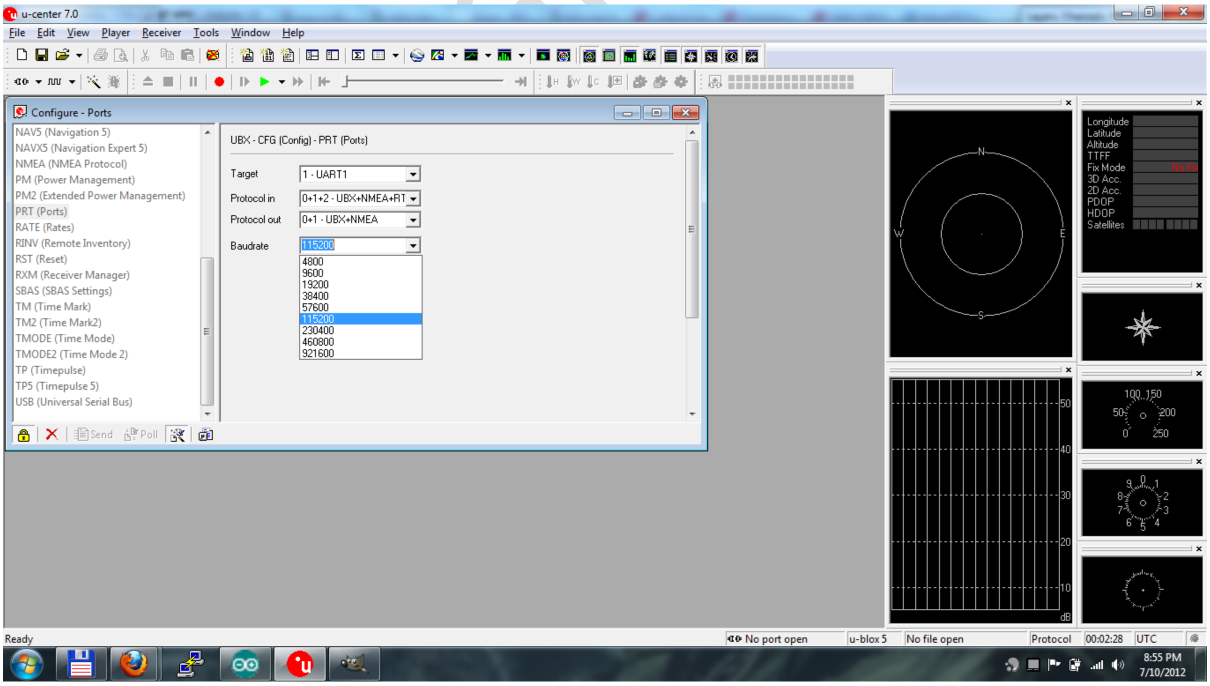
This program allows you to connect to the serial port of your USB to serial interface. Set it to 9600 baud, 8 bit, No parity, no flow control for now, as that’s what the u-blox module uses by default. You don’t need to change anything but the baud rate really. If that goes well, you should see GPS information streaming in, the little monitoring windows should jump to life. If that doesn’t work, try power cycling the GPS.

Once you have the GPS working with your PC, you need to go into the configuration window, set the baud rate and update speed, and save the configuration into the GPS memory. You’ll notice that there’s a small button cell on the GPS module, this actually powers the configuration memory because otherwise the u-blox would revert to default configuration at every power loss.



The Configuration Menu can be accessed via “View”, “Configuration View”

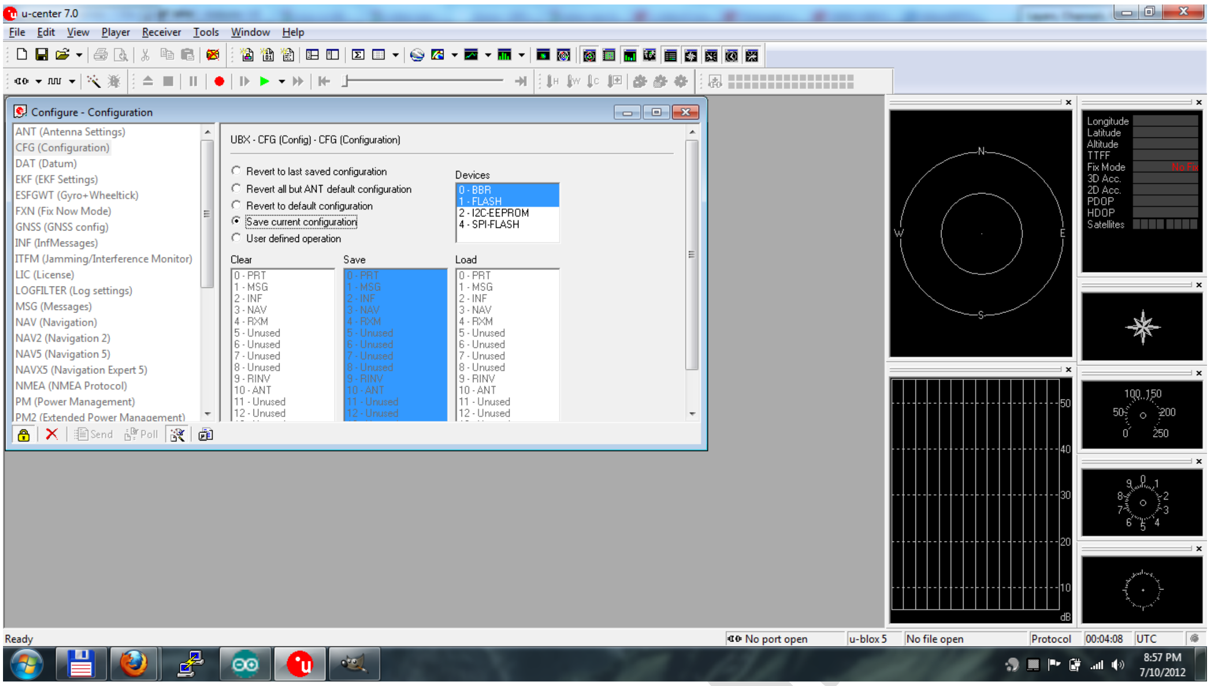
Now select **PRT (Ports)** from the list on the left, the Target is UART1, and the baud rate is 115200. Leave the protocol options unchanged. Now **Press the SEND button below the list on the left.** If you do not press SEND, the values will not be changed! (Note that the send button is on the left side, below the list of configuration categories. In these screenshots it is grayed out.)



Target: 1 - UART1
Protocol in: 0+1+2 - UBX+NMEA+RT
Protocol out: 0+1 - UBX+NMEA
Baudrate: 115200

Now scroll the left hand list down to **RATE (Rates)** and enter 100 ms for the Measurement Period, 10.00 Hz Measurement Frequency, 2 cycles Navigation Rate, 5.00 Hz Navigation

Frequency. Again, do not forget to press SEND when done. Now all that's left is to write the configuration to the configuration memory.



UBX - CFG (Config) - CFG (Configuration)

○ Revert to last saved configuration
○ Revert all but ANT default configuration
○ Revert to default configuration
● Save current configuration
○ User defined operation

Devices: 0 - BBR, 1 - FLASH (selected); 2 - I2C-EEPROM; 4 - SPI-FLASH

Clear: 0 - PRT, 1 - MSG, 2 - INF, 3 - NAV, 4 - RXM, 5 - Unused, 6 - Unused, 7 - Unused, 8 - Unused, 9 - RINV, 10 - ANT, 11 - Unused, 12 - Unused

Save (all selected): 0 - PRT, 1 - MSG, 2 - INF, 3 - NAV, 4 - RXM, 5 - Unused, 6 - Unused, 7 - Unused, 8 - Unused, 9 - RINV, 10 - ANT, 11 - Unused, 12 - Unused

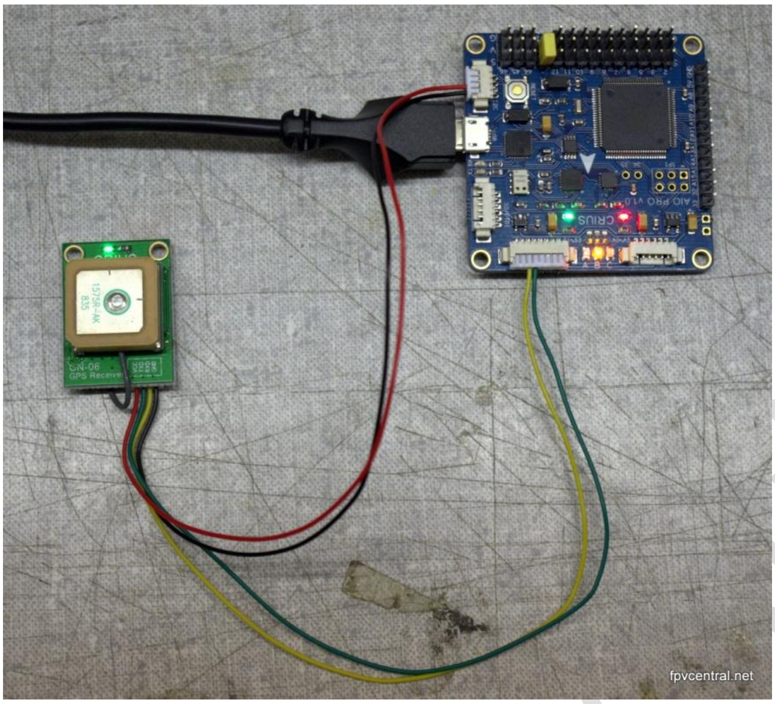
Load: 0 - PRT, 1 - MSG, 2 - INF, 3 - NAV, 4 - RXM, 5 - Unused, 6 - Unused, 7 - Unused, 8 - Unused, 9 - RINV, 10 - ANT, 11 - Unused, 12 - Unused

Scroll the list up to CFG (Configuration), select "Save current configuration", and press the SEND button.

Now the GPS module is ready for action.

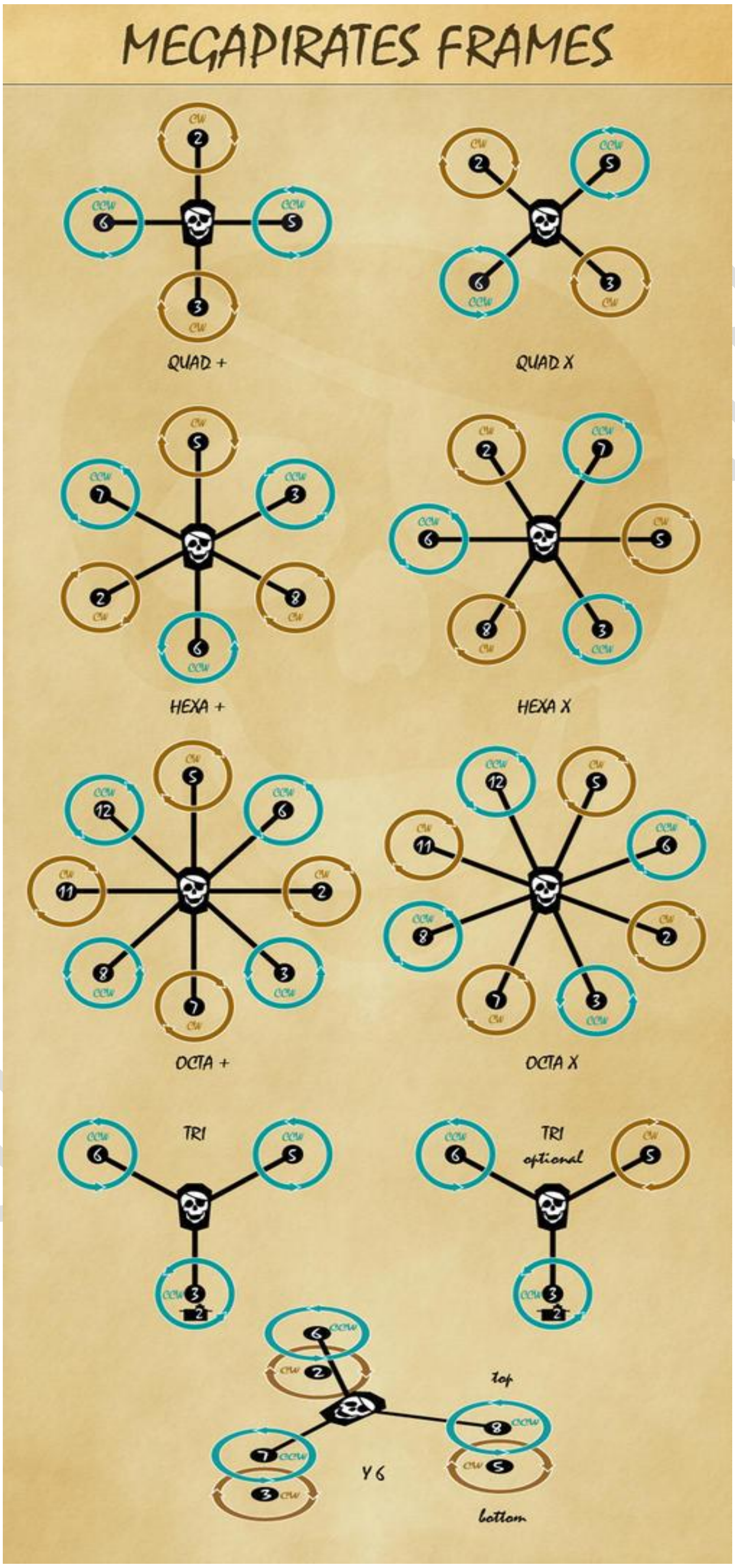
## Wiring the GPS module

The GPS needs to be connected to the RX2 and TX2 ports on the Crius AIO controller. At the same time, 5V power has to be supplied as well, but the serial ports 1, 2 and 3 on that connector don't offer Vcc (they can if you power the board externally and remove the yellow jumper, but that's for another tutorial) – so the easiest way is to just use the 5V and GND pins of the I2C port. This is simple because Crius was so nice to supply all the molex connectors needed with the board – all you have to do is switch around the pins a bit.



This is the simple solution to get a serial GPS working: Use 5V and GND from the I2C port, and connect the GPS to the RX2 and TX2 ports on the Crius AIO

# Frame and Motor Layout


MEGAPIRATES FRAMES
CW
CCW
QUAD +
QUAD X
HEXA +
HEXA X
OCTA +
OCTA X
TRI
TRI
optional
Y6
top
bottom