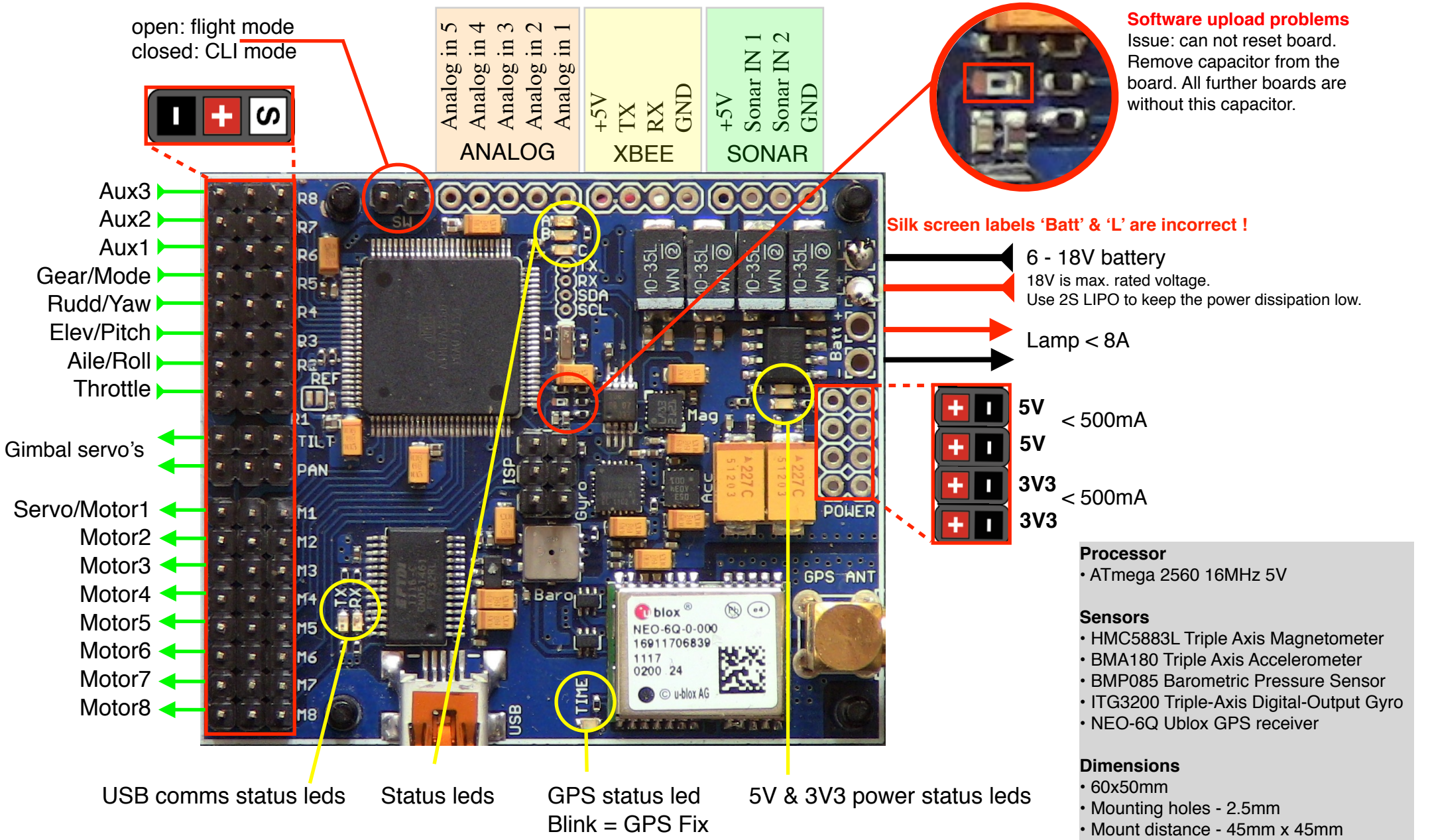


Black Vortex Board (All-In-One)

MegaPirate and MultiWii code compatible (rev. build r739)

No need to tweak the code, just flash, pass the setup and fly. GPS is working.

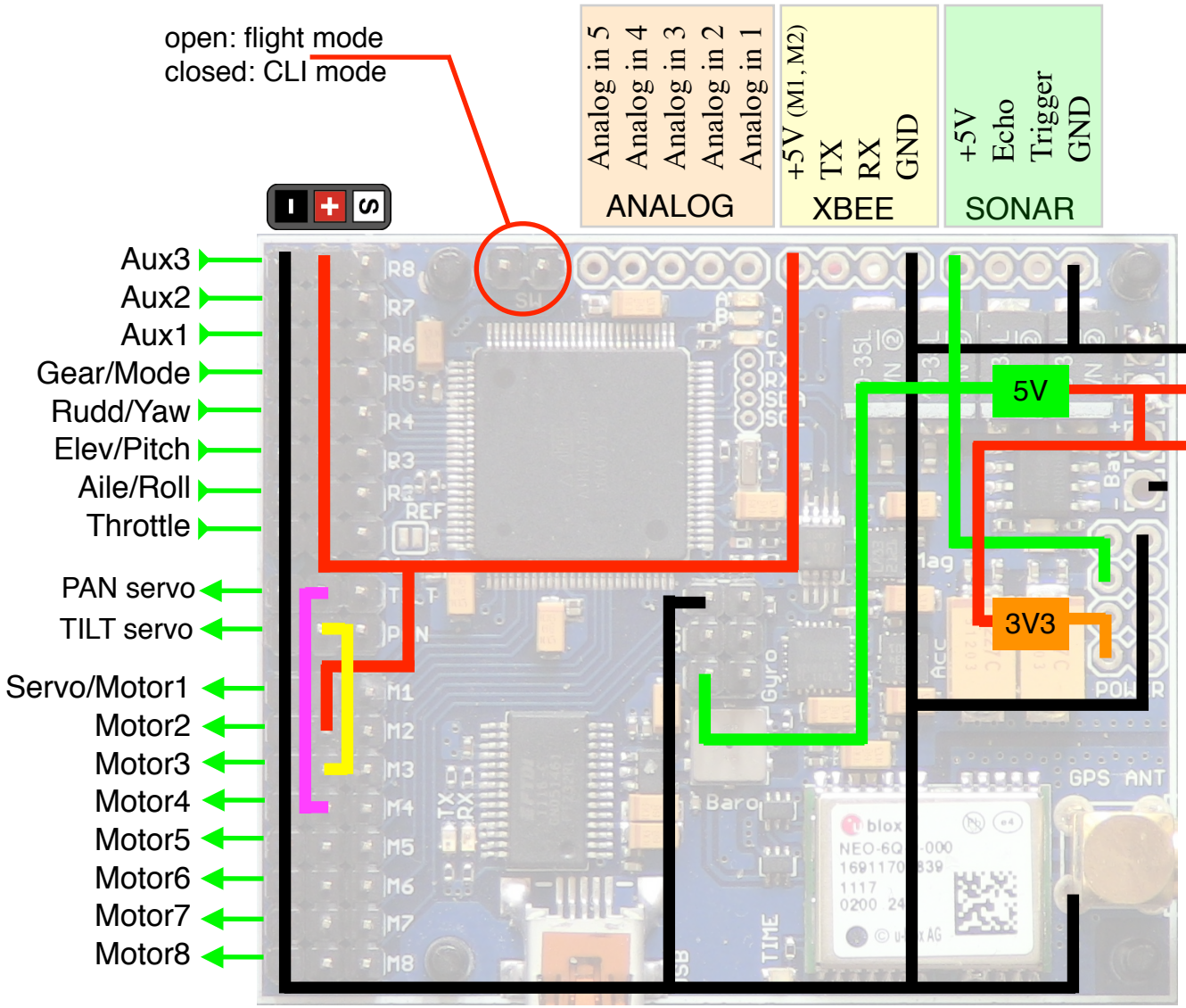
Processor
- ATmega 2560 16MHz 5V

Sensors
- HMC5883L Triple Axis Magnetometer
- BMA180 Triple Axis Accelerometer
- BMP085 Barometric Pressure Sensor
- ITG3200 Triple-Axis Digital-Output Gyro
- NEO-6Q Ublox GPS receiver

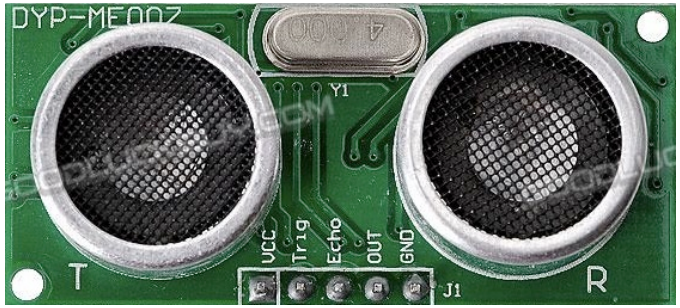
Dimensions
- 60x50mm
- Mounting holes - 2.5mm
- Mount distance - 45mm x 45mm

Links
- RC Groups
- MegaPirate BV B8 Quad-X
- Black Vortex bootloader

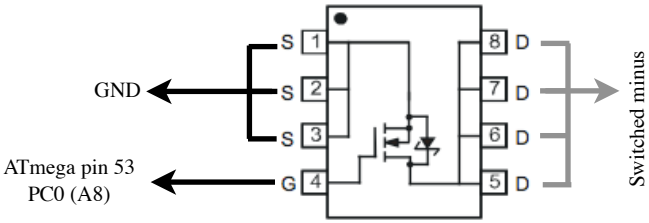
Black Vortex Board (All-In-One)



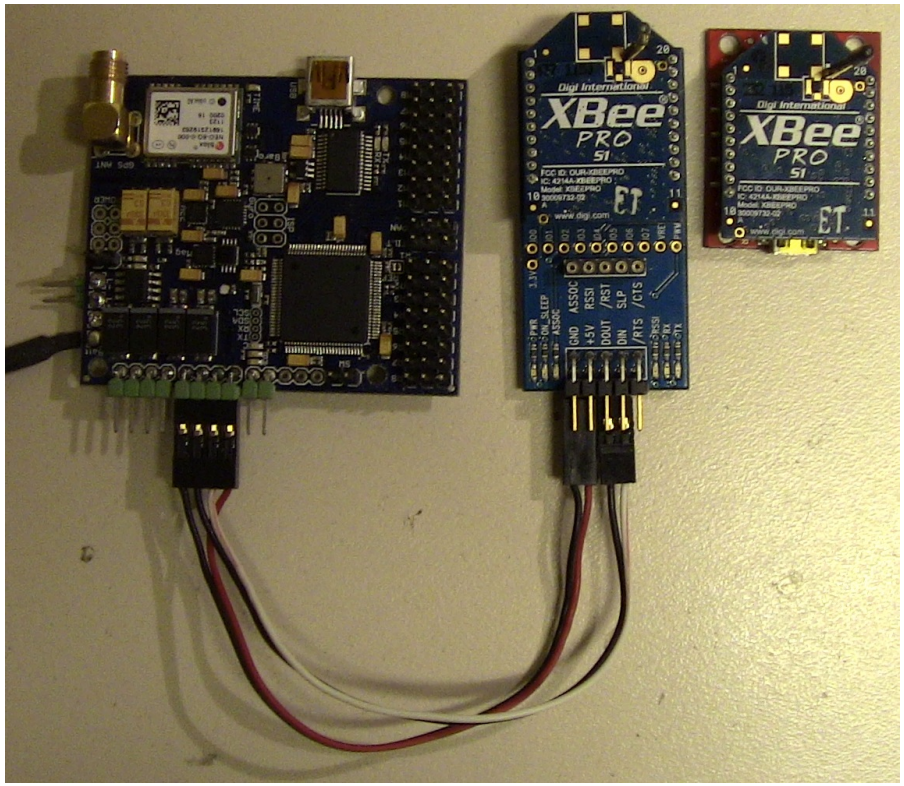
DYP-ME007 *Sonar works from v2.0.42 and up



DMS3015SSS-13



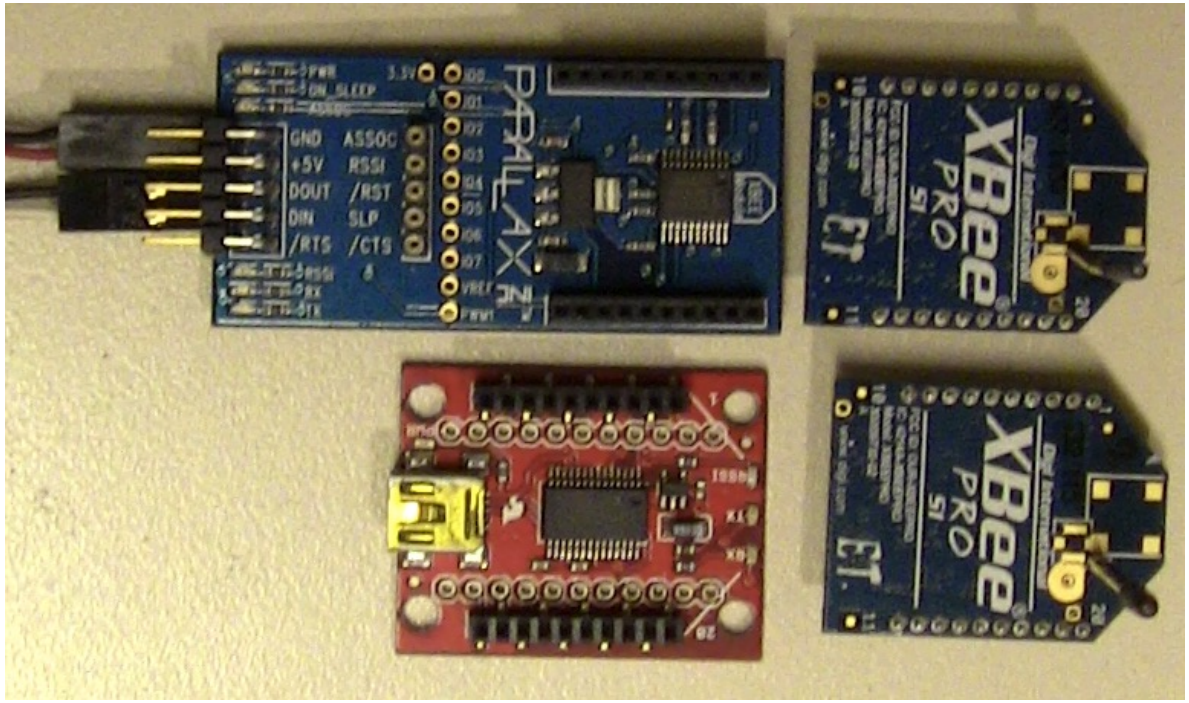
	Valid for MegaPirate B8+
LED A	
Solid lit:	system ready and motors are disarmed safe to move quad
Blink slow:	motors armed, Stable flight mode
Blink rapid:	motors armed, Acro flight mode
LED B	
Solid lit:	GPS position hold on
Off:	No GPS hold
LED C	
Solid lit:	2D or 3D fix lock on GPS
Off:	No reliable GPS info available



[Parallax XBee SIP Adapter]

Features:

- Onboard 3.3 V regulator
- 5V to 3.3 V logic translator buffers common I/O pins
- Six status indicator LEDs for Power, Tx, Rx, RSSI, Associate and mode (Sleep/ON)
- Small footprint dual SIP header provides support and allows easy interfacing to DOUT (TX), DIN (RX), RTS, 5 V supply and ground
- 5-pin female header connections provides interfacing to other XBee pins such as sleep, reset and associate
- A row of 10 plated through-holes with 0.1” spacing allows the option of soldering jumper wires or a header (not included) for access to the remaining XBee pins in advanced applications
- An additional plated through-hole gives access to 3.3 V output for ADC reference (VREF) when required



[Sparkfun XBee Explorer USB]

This is a simple to use, USB to serial base unit for the XBee line. This unit works with all XBee modules including the Series 1 and Series 2.5, standard and Pro version. Plug the unit into the XBee Explorer, attach a mini USB cable, and you will have direct access to the serial and programming pins on the XBee unit.

Note: As of August 2010, all new boards now include a MIC5219 voltage regulator which is good for 500mA.

EU **XBee 1mW Wire Antenna - Series 1 (2.4GHZ)**
XBee Pro 60mW Wire Antenna - Series 1 (2.4GHz)

Note: Don’t choose the XBee Pro series 2

XBee manufacture (Digi) page is here, XTCU software is here

APC220 can be obtained from e.g. GoodLuckBuy.com

Specification
- Working frequency: 431 MHz to 478 MHz
- Power: 3.3-5.5V
- Current: <25-35mA
- Working temperature: -20°C~+70°C
- Range (line of sight):
  - 1200m @1200 bps
  - 1000m @ 9600 bps
- Interface: UART/TTL
- Baud rate: 1200-19200 bps
- Baud rate (air): 1200-19200 bps
- Receive Buffer: 256 bytes
- Size: 37mm × 17 mm × 6.6mm
- Weight: 30g

Configuration software RF-Magic v12

USB interface CP210x (Silicon Laboratories)
Drivers for Windows, Mac and Linux can be found here

Howto by Nosepo can be found here (PDF)

APC220 & MAVlink

Syberian (Oleg) found out that APC220 chokes on the amount of telemetry data send send by MAVlink.

One has to lower the MAVlink update rate. This is done in the Mission Planner Configuration by setting the 'telemetry rates' to 3,1,1,0.

> The telemetry data amount was too high to fit in the 19200 baud aerial speed of this module. Since the modules are semi-duplex i.e. working sequentially on the same frequency, there is no space left for incoming ground commands. Also there were massive packet drops and skips on the board side.

To drastically reduce the amount of data send, comment out the lines below in `gcs_mavlink.pde`

```
// send_message(MSG_GPS_RAW);
// send_message(MSG_NAV_CONTROLLER_OUTPUT);
// send_message(MSG_RAW_IMU1);
// send_message(MSG_RAW_IMU2);
// send_message(MSG_RAW_IMU3);
```

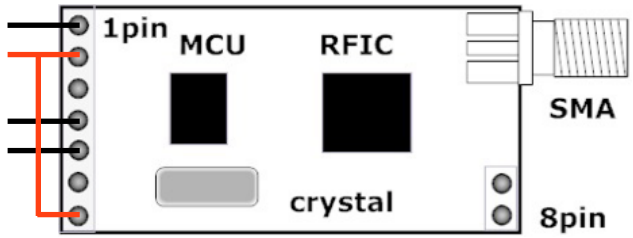
If Mission Planner becomes unresponsive (hangs) while editing the board settings thru the APC220, change the line below in `gcs_mavlink.pde`:

```
if (NULL != _queued_parameter && (requested_interface ==
    (unsigned)chan) && mavdelay > 1) {
```

to

```
if (NULL != _queued_parameter && (requested_interface ==
    (unsigned)chan) && mavdelay > 3) {
```

Pin	Definition	Detail
1	GND	0V Ground
2	VCC	3.3V-5.5V Power
3	EN	Enable the device when leave it disconnected or apply >1.6V Disable the device when apply <0.5V
4	RXD	UART RX
5	TXD	UART TX
6	AUX	UART Signal- Receive (low) Transmit (high)
7	SET	Set parameters (low)



Getting started

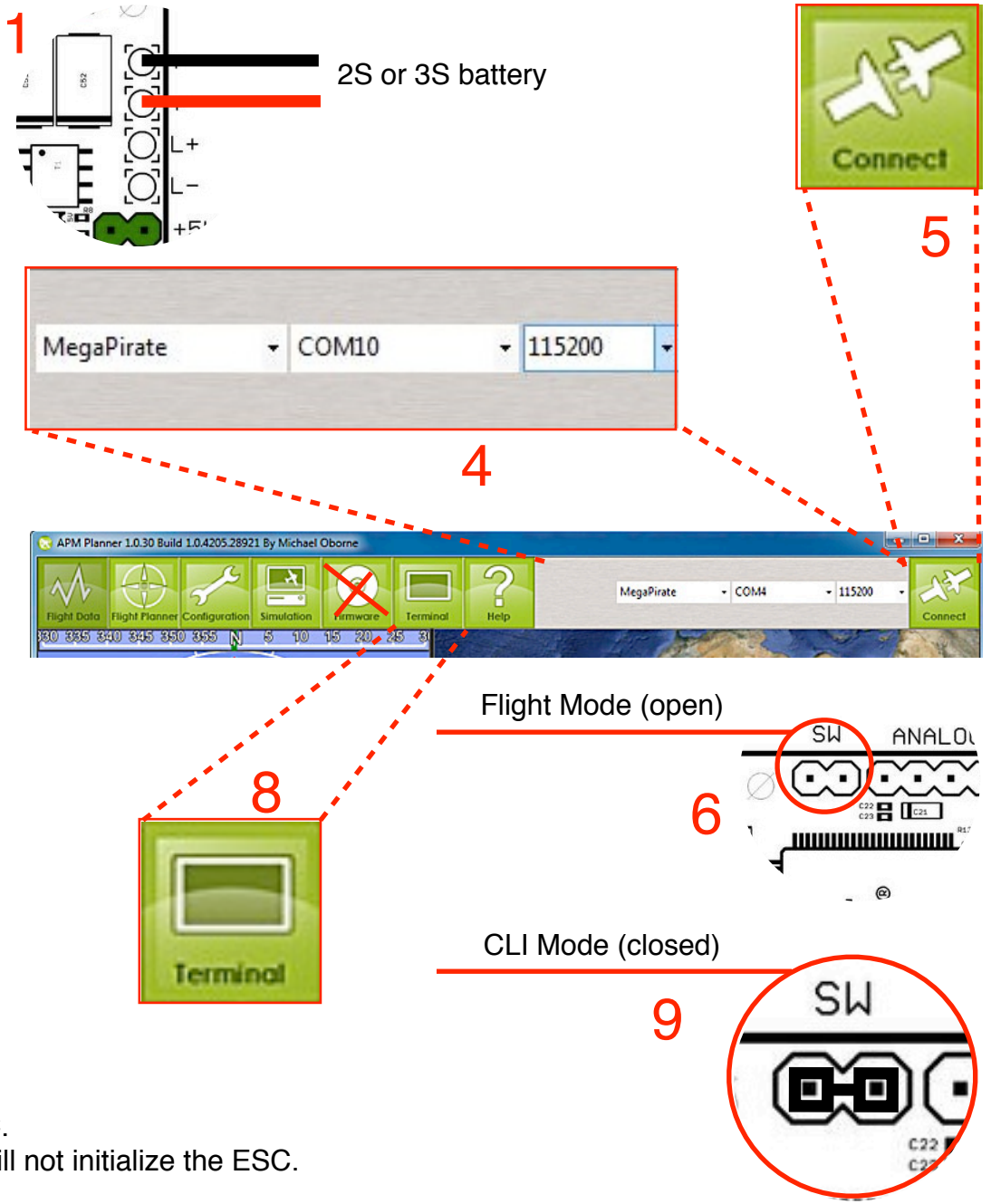
The Black Vortex Board comes pre-loaded with the MegaPirate B8+ (build r739) and is setup for a Quad in X configuration.

Download the following software:
- MegaPirates B8 QuadX archive from the RCgroups
  More on MegaPirates can be found on RCgroups here.
- Download Arduino 0022 or 0023 here
- Download the MegaPiratePlanner (MPP) here

If you have an different frame than Quad X, you need to install Arduino, add the MegaPirate B8 QuadX archive in the Arduino tree. Furthermore you have to change the APM_config.h to reflect your frame and orientation. Don't forget to add the Black Vortex Board to the Arduino, see '**Adding the Black Vortex board to Arduino**'.

DON'T USE THE FIRMWARE BUTTON in MPP !!!

Setup:
1. Connect an 2S or 3S battery to board
2. Plugin the USB
3. Start the MegaPiratePlanner
4. Set the serial communication port. Use 'MegaPirate' and 115200 !
5. Select the connect button
6. Notification popup to set the APM slider switch in the Flight Mode.
7. It takes about 20 seconds on average to establish an connection
8. Select the Terminal button
9. An popup tells to set the APM slider switch to CLI Mode
10. enter the command 'setup' at the prompt
11. enter the command 'radio' at the prompt and follow the instructions
12. repeat this for all the entries in the setup menu



Remark 1: when calibrating radio and/or motor, supply power to the ESC.
Remark 2: power up first the board and then the ESC. Simultaneously will not initialize the ESC.

Excellent PID tuning howto by Joebarteam can be found here. He also wrote build and configuration blog here.
For the declination value, visit the website Magnetic-Declination.com

Radio setup:

ch1 = Throttle	ch5 = mode switch - use your 3 position switch
ch2 = Aile / Roll	ch6 = used for in-air tuning - see AP_Config.h for options
ch3 = Elev / Pitch	ch7 = use to set throttle hold value while hovering (quick toggle), hold to trim in air values - don't use your radio's trims!
ch4 = yaw / Rudder	ch8 = NOT used!!! - this is the hardware manual - it's dangerous to use for quads BEWARE!!!

CLI interactive setup - You must go through each item and set the values to match your hardware

setup menu:

erase	- when installing ACM for the first time, run this to erases bad values from EEPROMS – just in case
reset	- Performs factory reset and initialization of EEPROM values
radio	- records the limits of ALL radio channels - very important!!!
pid	- restores default PID values - - only needed if you have changed them in flight with CGS, not for setup.
frame	- sets your frame config: [x, +, tri, hexax, hexa+, y6]
motors	- interactive setup of your ESC and motors, enter this mode, then plug-in battery, point at motors to make them spin throttle will output full range to each motor - this is good for esc calibration
level	- sets initial value of accelerometers - hold copter level
modes	- sets the flight modes assigned to each switch position (you have 5 available)
current	- enables an Attopilot current sensor: [on, off, milliamp hours]
compass	- enables the compass [on, off]
declination	- sets your local declination value – lookup online for accuracy [decimal degrees]
sonar	- Sonar hooks to the "pitot" port which is an analog input [on, off]
show	- a formatted output of all the settings

test menu:

pwm	- outputs the pwm values of all 8 radio channels
radio	- outputs the control values of all 8 radio channels in degrees * 100 or other value (see radio.pde)
gps	- outputs GPS data
rawgps	- outputs raw, unparsed GPS data
adc	- outputs raw adc values
imu	- outputs euler angles
battery	- outputs voltage readings to analog in 0-3
current	- outputs voltage and current from an AttoPilot current sensor
relay	- toggles the relay
sonar	- outputs sonar data in cm
waypoints	- dumps stored waypoint commands
airpressure	- raw output of absolute pressure sensor
compass	- outputs compass angles in degrees (0 = north)
xbee	- outputs an XBEE sequence used for range testing
mission	- writes a default mission to EEPROM [null, 'wp']. Choosing 'wp' option will send the copter 15 meters North and back again.
eedump	- raw output of bytes in eeprom

logs Menu:

See the APM wiki to better understand how to dump logs and how to set the types of data you want to record.

ACM Flight modes
Set these up in 'setup'/'modes'. Use your three position switch (channel 5) to select. Change the setting with your roll (Aileron) stick. Hit enter to save.
All of the modes allow the user to control the copter with the normal controls.
You can get yourself out of a jam sometimes by simply nudging the copter while in AUTO or LOITER modes.

Options:

ACRO	- rate control only. not for beginners
STABILIZE	- copter will hold -45 to 45° angle, throttle is manual.
SIMPLE	- Remembers the orientation of the copter when first entering SIMPLE mode, allowing the user to fly more intuitively. Manual Throttle.
ALT_HOLD	- altitude is controlled by the throttle lever. Middle is hold, high = rise, low = fall.
LOITER	- When selected, it will hold the current altitude, position and yaw. Yaw is user controllable. roll and pitch can be overridden temporarily with the radio. altitude is controlled by the throttle lever. Middle is hold, high = rise, low = fall.
RTL	- Will try and fly back to home at the current altitude.
AUTO	- Will fly the mission loaded by the Waypoint writer
GCS_AUTO	- A future mode where the copter can be flown interactively from the GCS

Special note: The props will NOT spin in stabilize when throttle is in the off position, even when armed.
Arming is Yaw right for 1 sec, disarm is yaw left for 1 sec. Just give it some juice to confirm arming.

Auto modes will NOT engage until the throttle is above neutral.
So if you put the control switch to position hold while it's on the ground, it will no spin up. Or at least it shouldn't ;)

Adding the Black Vortex board to Arduino

Open the file .\arduino-0022\hardware\arduino\boards.txt
Add the following lines:

```
vortex.name=Black Vortex (ATmega2560)

vortex.upload.protocol=stk500
vortex.upload.maximum_size=258048
vortex.upload.speed=57600

vortex.bootloader.low_fuses=0xFF
vortex.bootloader.high_fuses=0xD8
vortex.bootloader.extended_fuses=0xFD
vortex.bootloader.path=atmega
vortex.bootloader.file=BOOT_mega2560.hex
vortex.bootloader.unlock_bits=0x3F
vortex.bootloader.lock_bits=0x0F

vortex.build.mcu=atmega2560
vortex.build.f_cpu=16000000L
vortex.build.core=arduino
```

Bootloader programming

1. The Arduino boards.txt must contain the vortex definition (see above)
2. add the file **BOOT_mega2560.hex** to the folder \arduino-0022\hardware\arduino\bootloaders
3. Use Arduino to upload the bootloader.

Tricopter

To reverse yaw servo, open the file **motors_tri.pde**
Change the following (line #28)

```
motor_out[CH_3]= g.rc_4.radio_out;
```

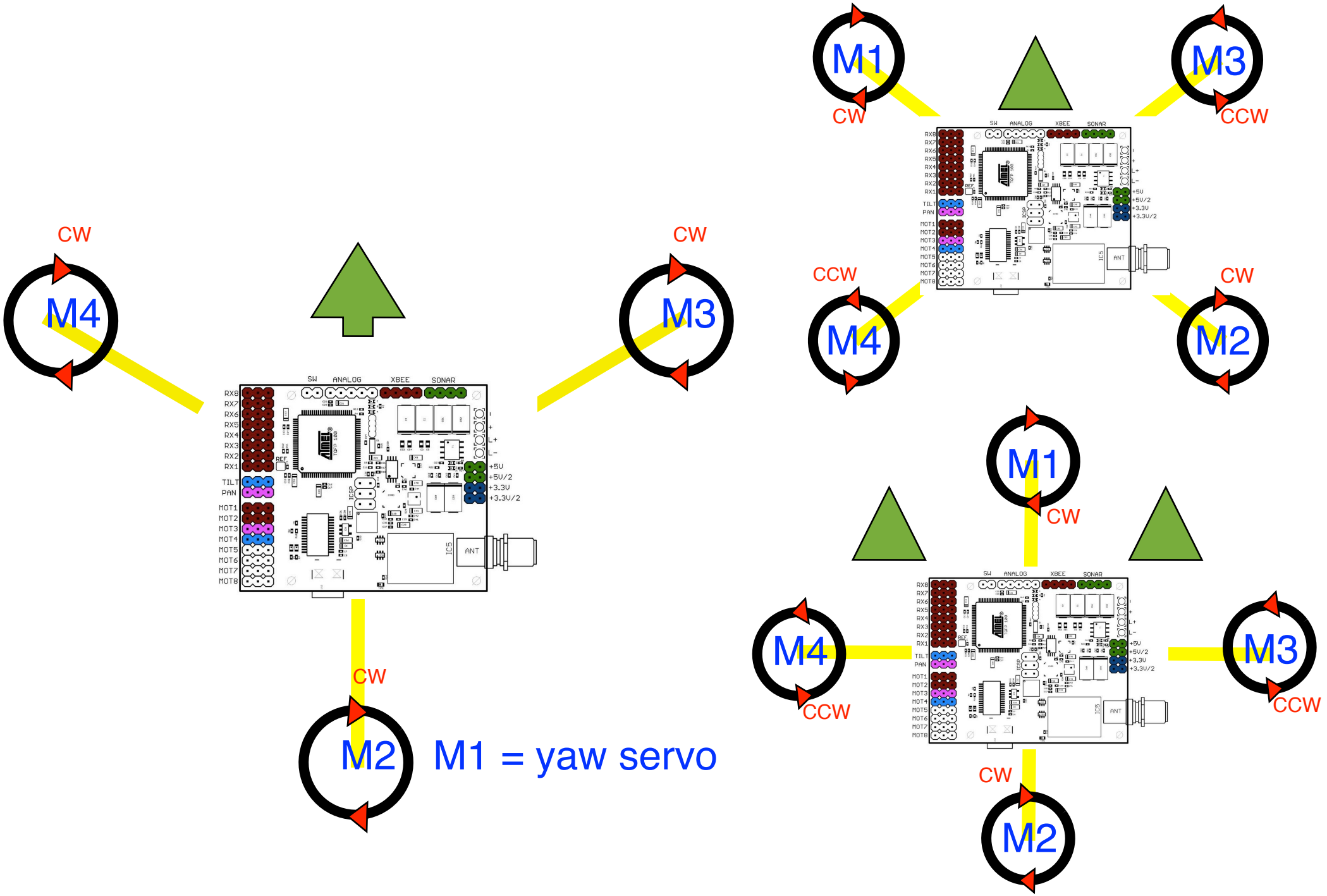
to

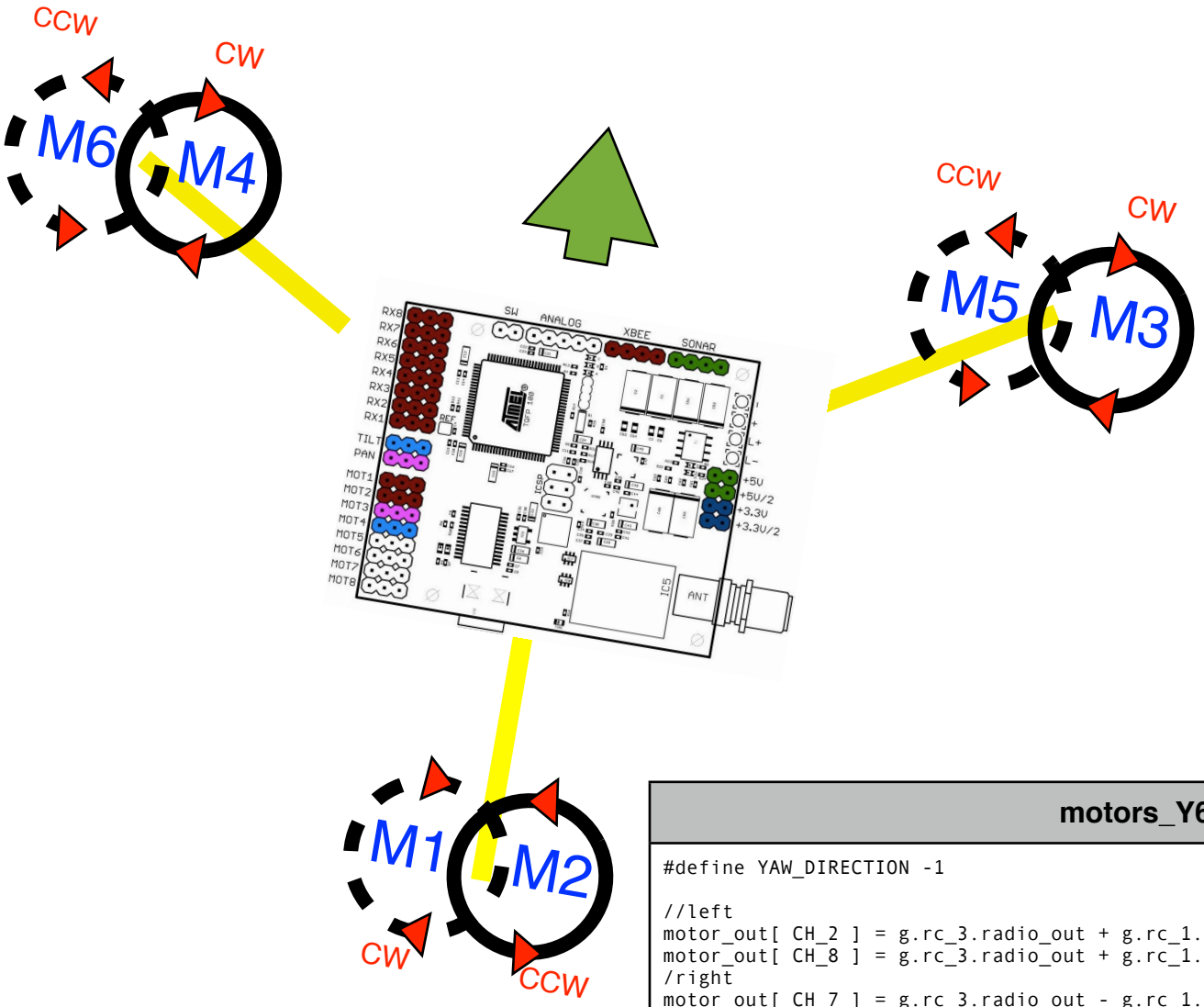
```
motor_out[CH_3]= 1500-g.rc_4.pwm_out;
```

Y6

For the Y6 frame, changes the following in de APM_Config.h

#define FRAME_CONFIG ~~QUAD_FRAME~~ Y6_FRAME
#define FRAME_ORIENTATION ~~X_FRAME~~ V_FRAME


M1
CW
M3
CCW
M4
CCW
M2
CW
M4
CW
M3
CW
M2
M1 = yaw servo
M1
CW
M4
CCW
M3
CCW
CW
M2



MP	MultiWii

`#define YAW_DIRECTION 1`
 	
`//left`
`motor_out[CH_2]`
`motor_out[CH_3]`
`//right`
`motor_out[CH_7]`
`motor_out[CH_1]`
`//back`
`motor_out[CH_8]`
`motor_out[CH_4]`
 	
`//left`
`motor_out[CH_2]`
`motor_out[CH_3]`
`//right`
`motor_out[CH_7]`
`motor_out[CH_1]`
`//back`
`motor_out[CH_8]`
`motor_out[CH_4]`
	`#define YAW_DIRECTION -1`
	
	`//left`
	`motor_out[CH_2]`
	`motor_out[CH_8]`
	`/right`
	`motor_out[CH_7]`
	`motor_out[CH_1]`
	`//back`
	`motor_out[CH_3]`
	`motor_out[CH_4]`
	
	`//left`
	`motor_out[CH_8]`
	`motor_out[CH_2]`
	`//right`
	`motor_out[CH_7]`
	`motor_out[CH_1]`
	`//back`
	`motor_out[CH_3]`
	`motor_out[CH_4]`

Black Vortex Y6 setup like MultiWii Y6 setup.
The “motors_Y6.pde” must be changed!
See outline on the right.

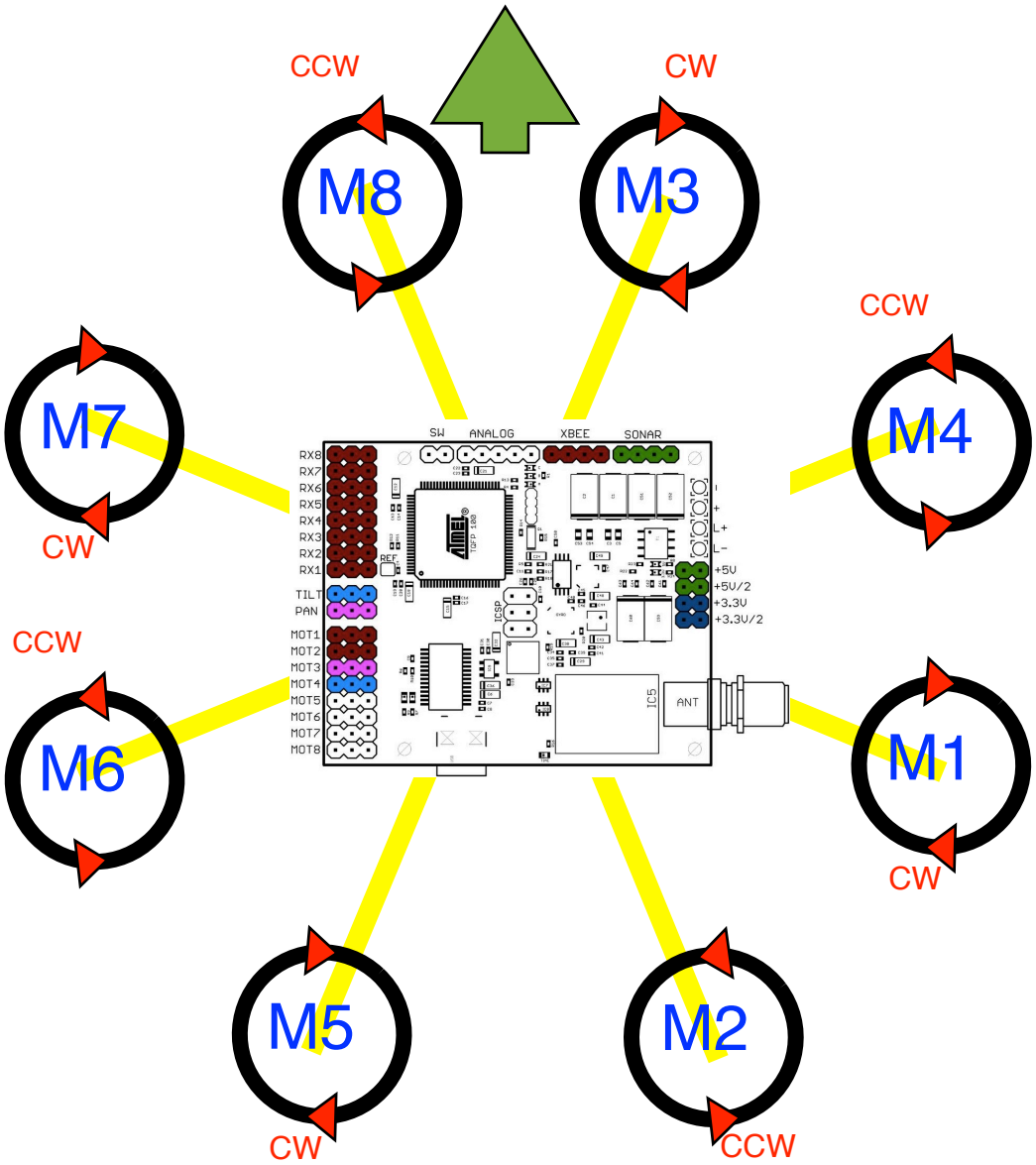
Remark:
- M1, M5 & M6 are at the bottom
- M2, M3 & M4 are at the top

motors_Y6.pde (MultiWii config)

```
#define YAW_DIRECTION -1

//left
motor_out[ CH_2 ] = g.rc_3.radio_out + g.rc_1.pwm_out + (g.rc_2.pwm_out * 2 / 3); // LEFT TOP - CW
motor_out[ CH_8 ] = g.rc_3.radio_out + g.rc_1.pwm_out + (g.rc_2.pwm_out * 2 / 3); // BOTTOM_LEFT - CCW
/right
motor_out[ CH_7 ] = g.rc_3.radio_out - g.rc_1.pwm_out + (g.rc_2.pwm_out * 2 / 3); // RIGHT TOP - CW
motor_out[ CH_1 ] = g.rc_3.radio_out - g.rc_1.pwm_out + (g.rc_2.pwm_out * 2 / 3); // BOTTOM_RIGHT - CCW
//back
motor_out[ CH_3 ] = g.rc_3.radio_out - (g.rc_2.pwm_out * 4 / 3); // REAR TOP - CCW
motor_out[ CH_4 ] = g.rc_3.radio_out - (g.rc_2.pwm_out * 4 / 3); //BOTTOM_REAR - CW

//left
motor_out[ CH_8 ] -= YAW_DIRECTION * g.rc_4.pwm_out; // LEFT TOP - CW
motor_out[ CH_2 ] += YAW_DIRECTION * g.rc_4.pwm_out; // LEFT BOTTOM - CCW
//right
motor_out[ CH_7 ] -= YAW_DIRECTION * g.rc_4.pwm_out; // RIGHT TOP - CW
motor_out[ CH_1 ] += YAW_DIRECTION * g.rc_4.pwm_out; // RIGHT BOTTOM - CCW
//back
motor_out[ CH_3 ] += YAW_DIRECTION * g.rc_4.pwm_out; // REAR TOP - CCW
motor_out[ CH_4 ] -= YAW_DIRECTION * g.rc_4.pwm_out; // REAR BOTTOM - CW
```


CCW
CW
M8
M3
M7
M4
CCW
CW
CCW
M6
M1
CW
M5
M2
CW
CCW
SW
ANALOG
XBEE
SONAR
RX8
RX7
RX6
RX5
RX4
RX3
RX2
RX1
TILT
PAN
MOT1
MOT2
MOT3
MOT4
MOT5
MOT6
MOT7
MOT8
REF
ICSP
ANT
+5V
+5V/2
+3.3V
+3.3V/2