SYNERDUINO
KWAD SHIELD

V1.1

SYNERFLIGHT

INTRODUCTION TO ARDUINO MULTIROTOR DRONES

- PREPARATION OF MATERIALS AND TOOLS NEEDED
- INSTALLATION NECESSARY APPLICATION FOR CONFIGURING THE MICRO CONTROLLER
- HOW DRONES OPERATE AND FLY
- PARTS AND COMPONENTS
- TYPES OF MULTICOPTER ARRANGEMENTS
- RELATIONSHIP OF CODES TO OUTPUT OF THE MICRO CONTROLLER
- RC CONTROL MAPPING
- CONFIGURING THE IMU SENSORS
- UNDERSTANDING PROPELLERS, MOTORS AND MOTOR SPEED CONTROLLERS
- SETTING UP BATTERY SENSOR
- UNDERSTANDING BATTERY USAGE AND SAFETY
- PID TUNING FOR FLIGHT
- PID FOR STABILIZE MODE
- CONFIGURING TELEMETRY
- LEARN TO FLY BASICS
- PID FOR ALTITUDE HOLD
- CONFIGURING GPS **(ADVANCE LEVEL) 2560 MEGA REQUIRED**
- PID FOR GPS MODES **(ADVANCE LEVEL) 2560 MEGA REQUIRED**
- GPS NAVIGATION **(ADVANCE LEVEL) 2560 MEGA REQUIRED**

Anatomy of a DIY Drone

RF Transmitter

RF Receiver

GPS Receiver

Magnetometer

Flight Controller

Brushless DC Motor

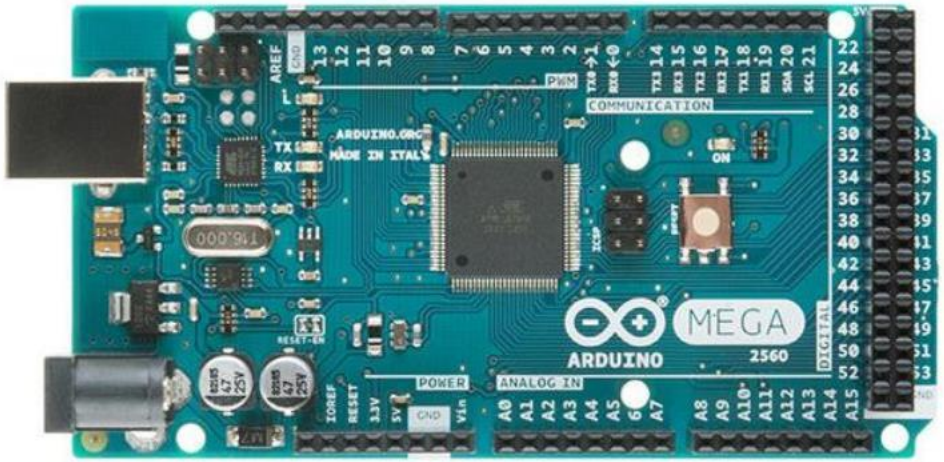
Accelerometer

Gyroscope

Barometer

Lithium Polymer Battery

Electronic Speed Controls (ESC)



SYNERDUINO
KWAD SHIELD KIT

Board Capabilities

Features / Flight Modes	UNO 328 (SMD)	Mega 2560	Mega 2560 + GPS
Heading hold	X	X	X
Altitude Hold	X	X	X
Headless / Course Lock	X	X	X
Telemetry	X	X	X
Payload Trigger		X	X
Camera Gimbal		X	X
RTH			X
Position Hold			X
Missions			X

Tools needed

Screwdriver / Hex set

Cutter Knife

Soldering Set

PVA Glue

Thread Lock , Purple

Tape Double sided and Electrical

Zip Ties

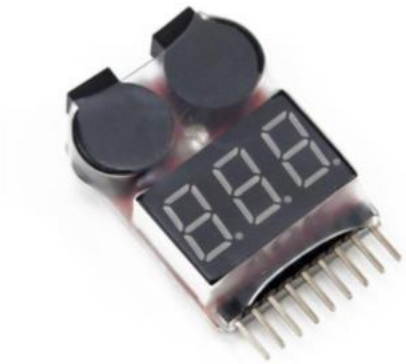
Pliers

Tools Needed

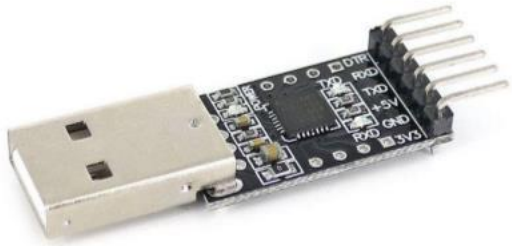
Lipo Charger 5A

Lipo Battery 3s 1300mah

Voltage alarm



USB TTL FTDI



Hardware

Synerduino Kit + 250mm frame setup

5x4.5 3 or 5x4 3 bladed propellers (Stiff Plastic)

5x45x3 (5045 3) or 5x40x3 (5040 3)

Counter rotation CCW & CW

Bluetooth HC05 and GPS

XT60 Plug

Servo Wires

5V 3A UBEC – for those who needs to run extra servos or sensor components

Recommend 2200 - 2300 Size 2000Kv-2400KV motors those with 22mm Stator size average rate of 1000g thrust

ESC OPTO Standard PWM or BL Heli or BL Heli-S 20A-30A 3s-4s

ESC With BEC if you needed to run extra servos ,sensors or accessories

Synerduino Kwad Shield BETA GY801

Note : surface mount your solder ESC wire make sure it doesn't penetrate to the bottom of the board

ESC is Solder on Top side only

ESC / Servo PWM Out

Serial Pins

ESC is Solder on Top side only

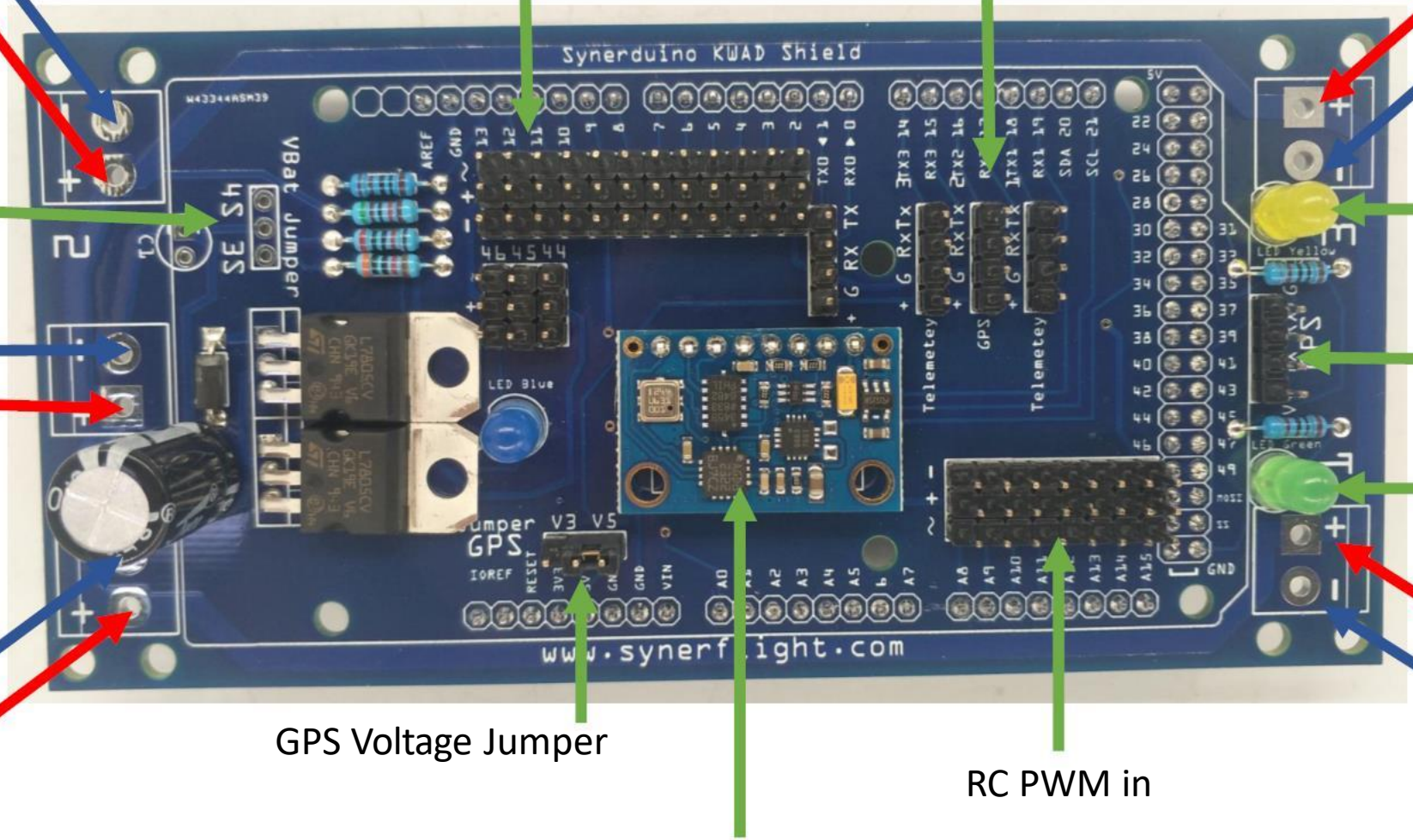
Set jumper to the Battery Cell count (Solded) 3s – 4s

GPS LED

Power input
3s 11.1V
4s 14.8V

GPS Serial Pins

Status LED



ESC is Solder on Top side only

GPS Voltage Jumper

RC PWM in

ESC is Solder on Top side only

For improve performance IMU must be protected from the Environment

IMU : L3G4200D Gyro / ADXL345 Accelerometer / BMP180 – 85 Baro / MMC5883 Mag

Synerduino Kwad Shield V1.1 GY91

Note : surface mount your solder ESC wire make sure it doesn't penetrate to the bottom of the board

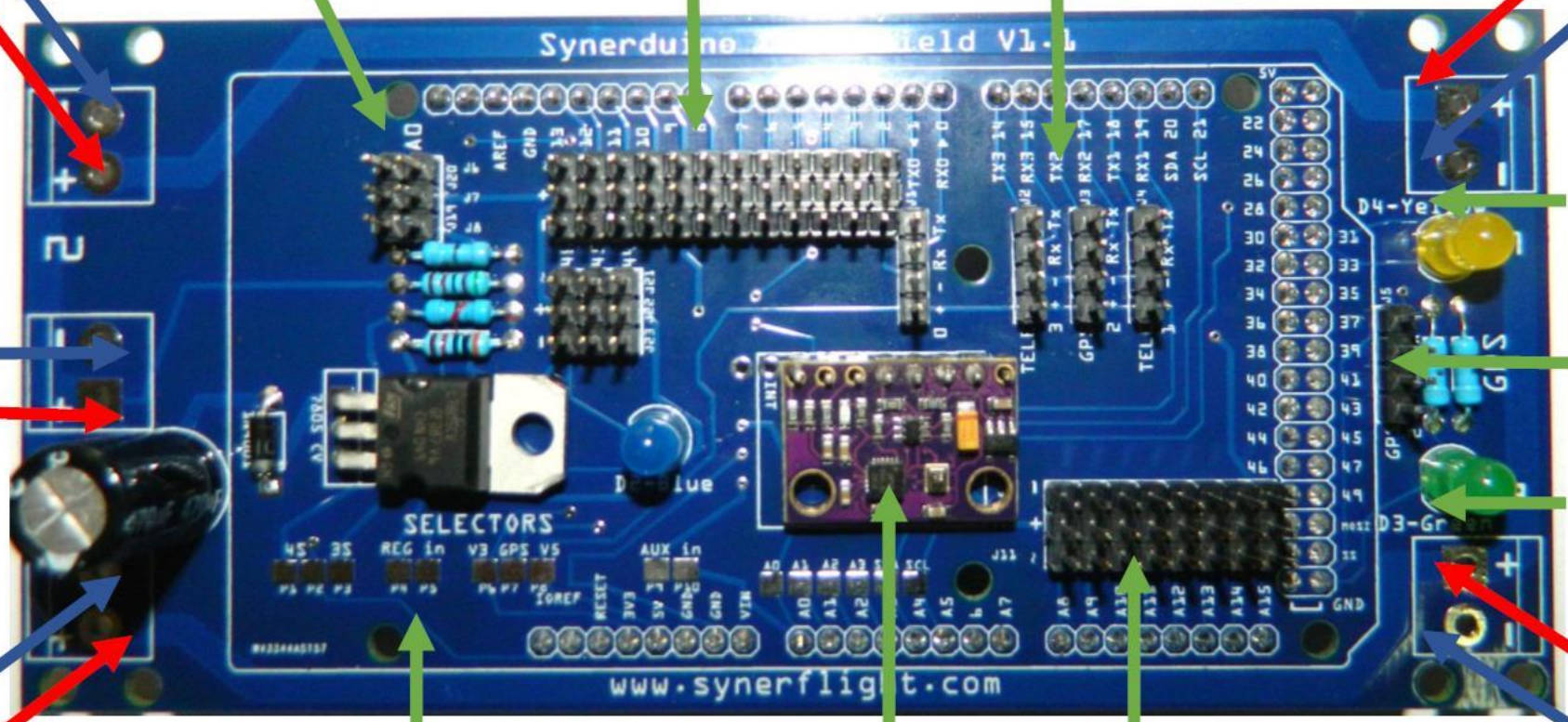
ESC is Solder on Top side only

Aux ADC in

ESC / Servo PWM Out

Serial Pins

ESC is Solder on Top side only



GPS LED

Power input 3s 11.1V

GPS Serial Pins

Status LED

ESC is Solder on Top side only

Jumper Pads Selector Zone

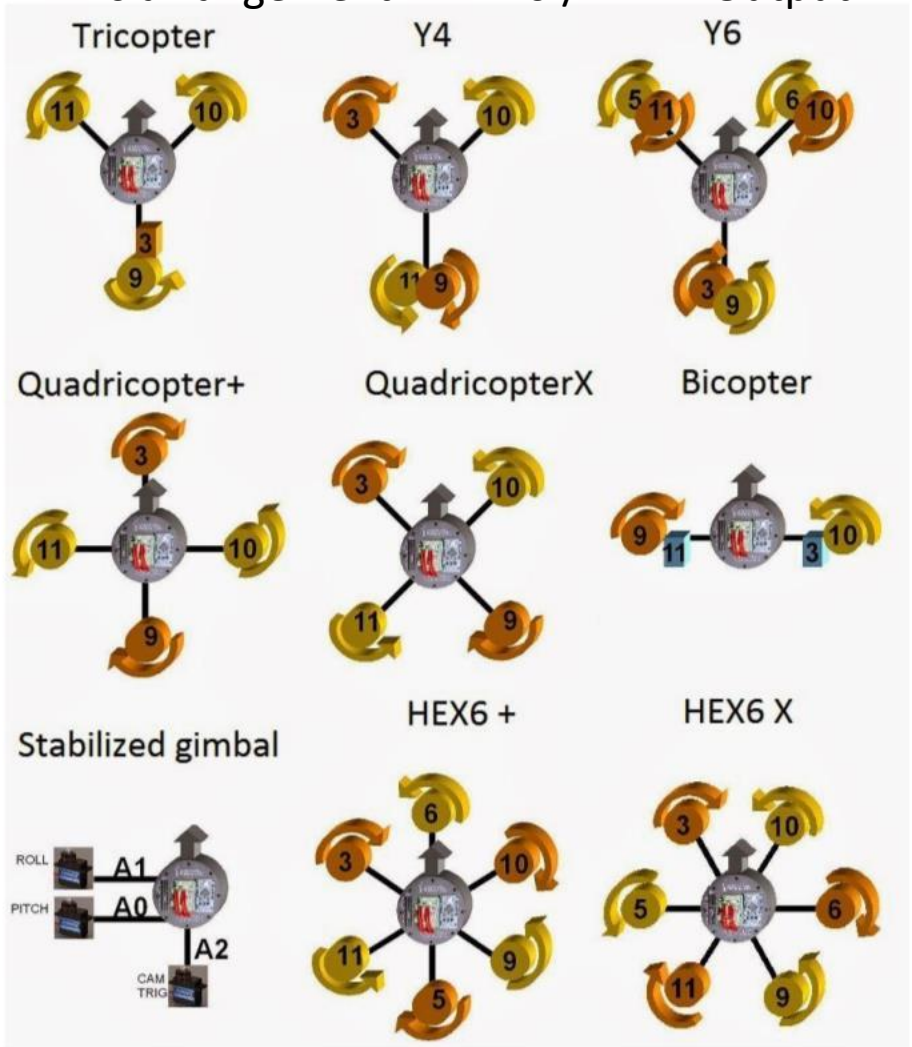
RC PWM in

ESC is Solder on Top side only

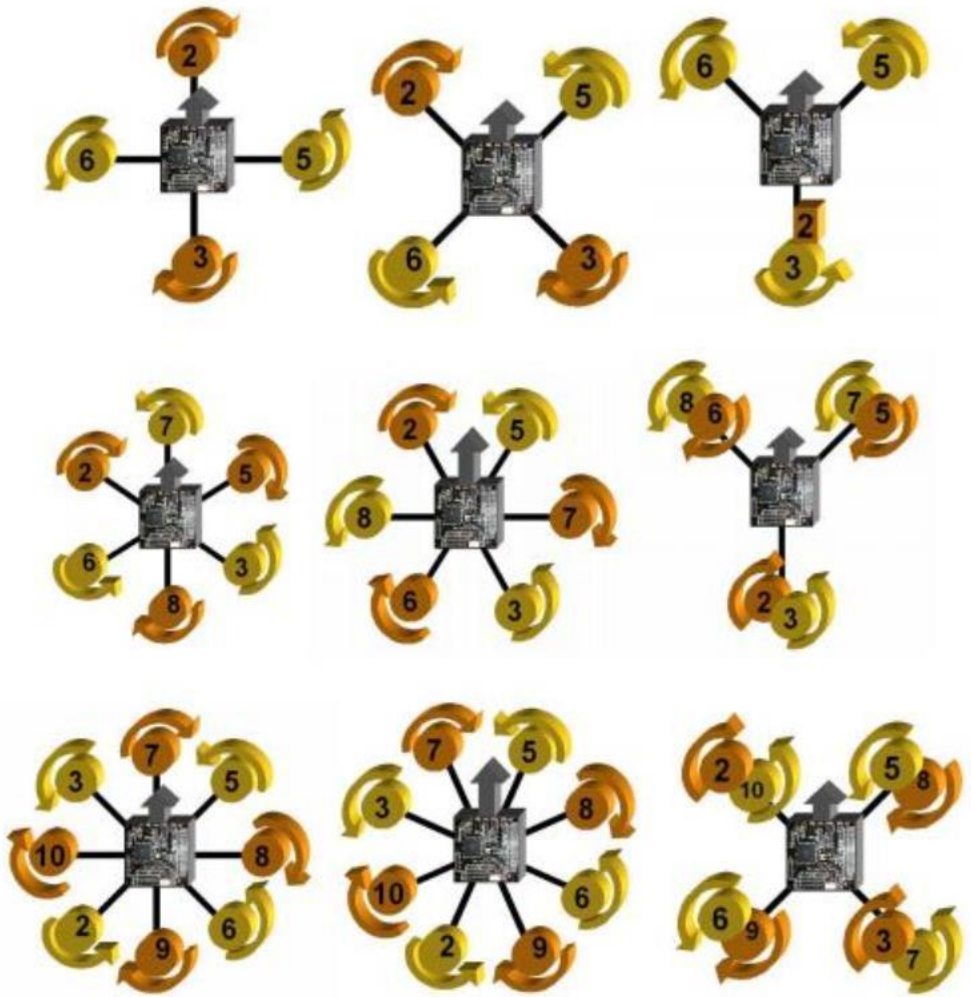
For improve performance IMU must be protected from the Environment

IMU : MPU-9250 & BMP280

PWM Pins arrangement D2-D10 / PWM Output



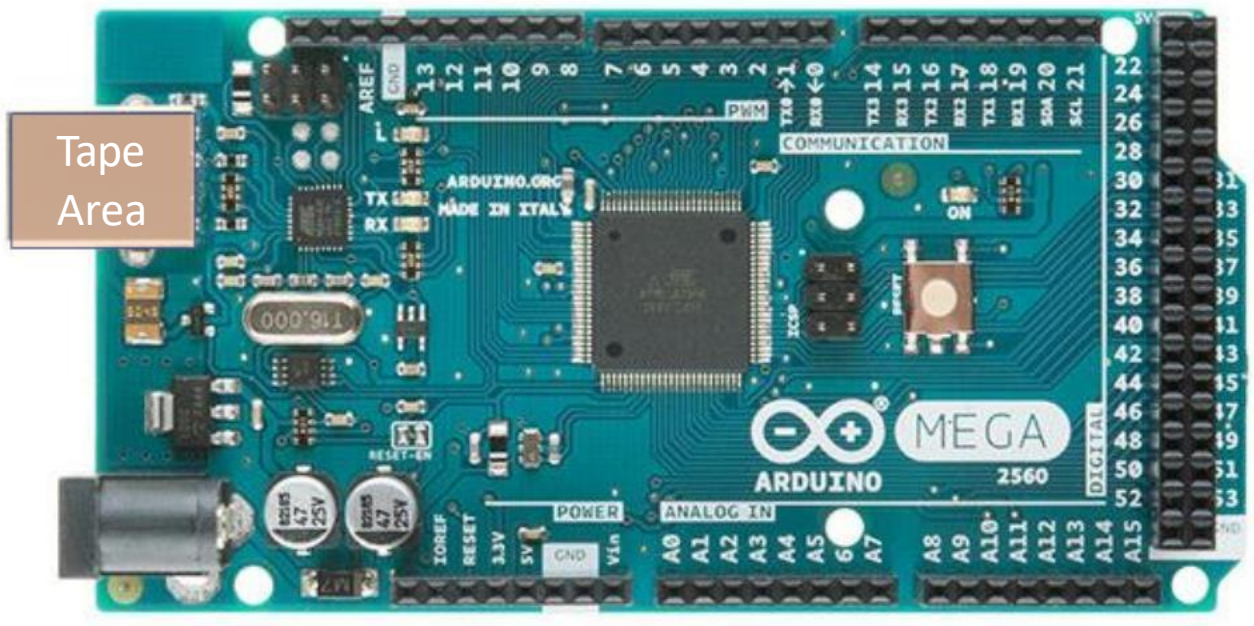
UNO 328



MEGA 2560

Arduino Board Preparation

Ensure insulation from the Arduino board add tape on these areas



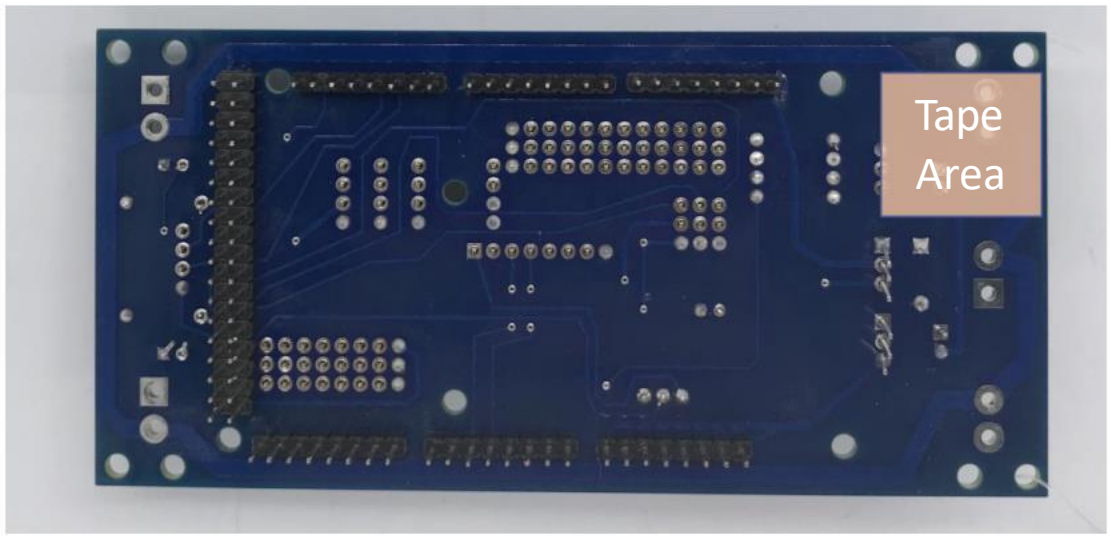
2560 MEGA

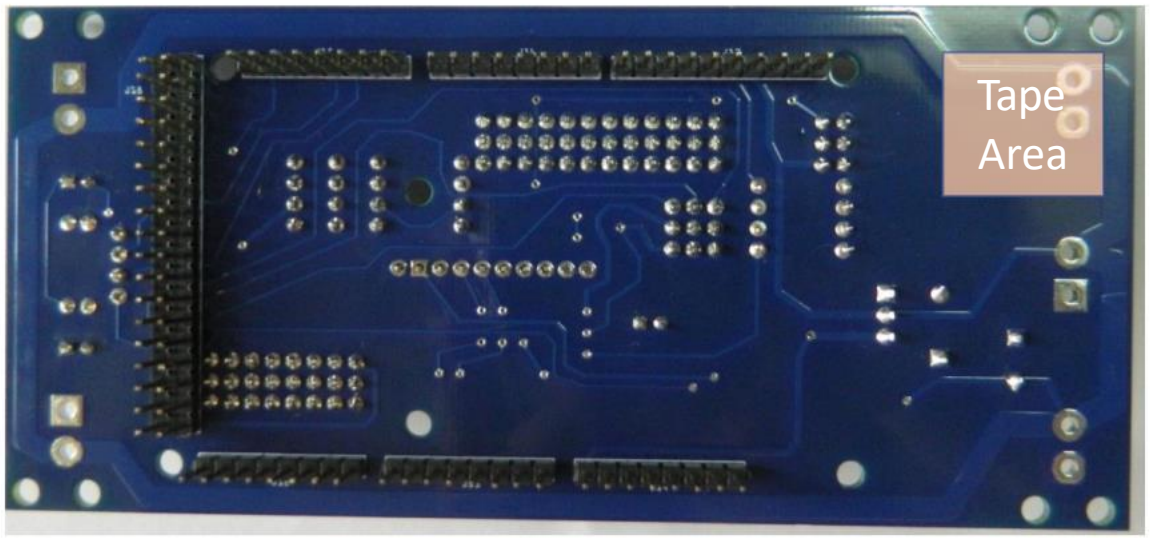
SMD UNO 328

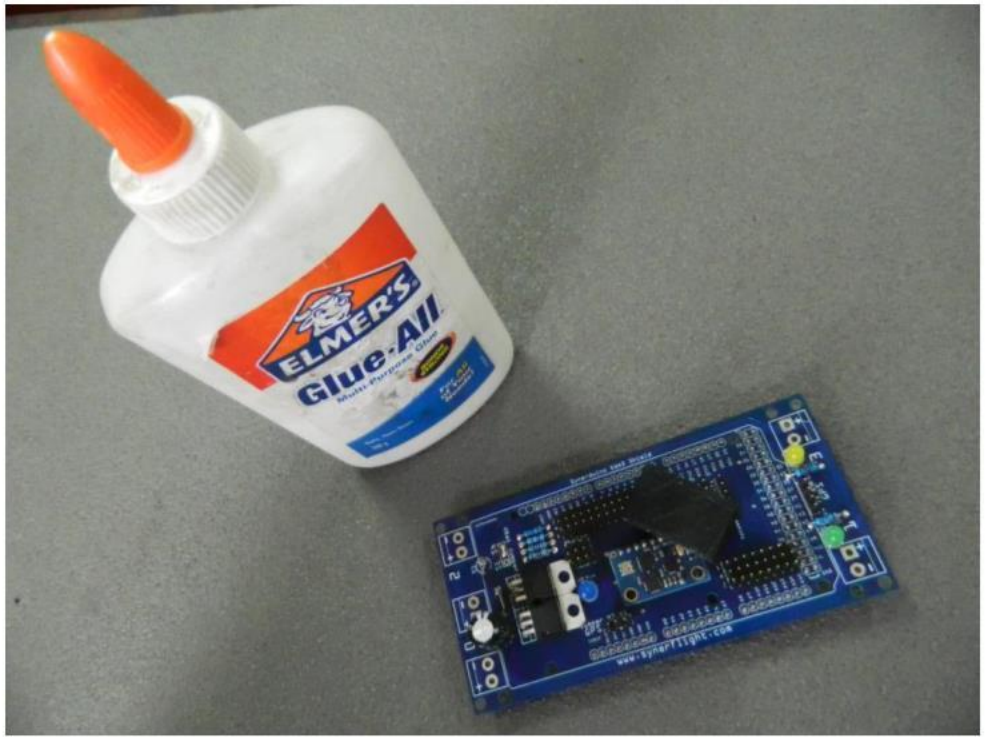
Synerduino Kwad Shield Preparation

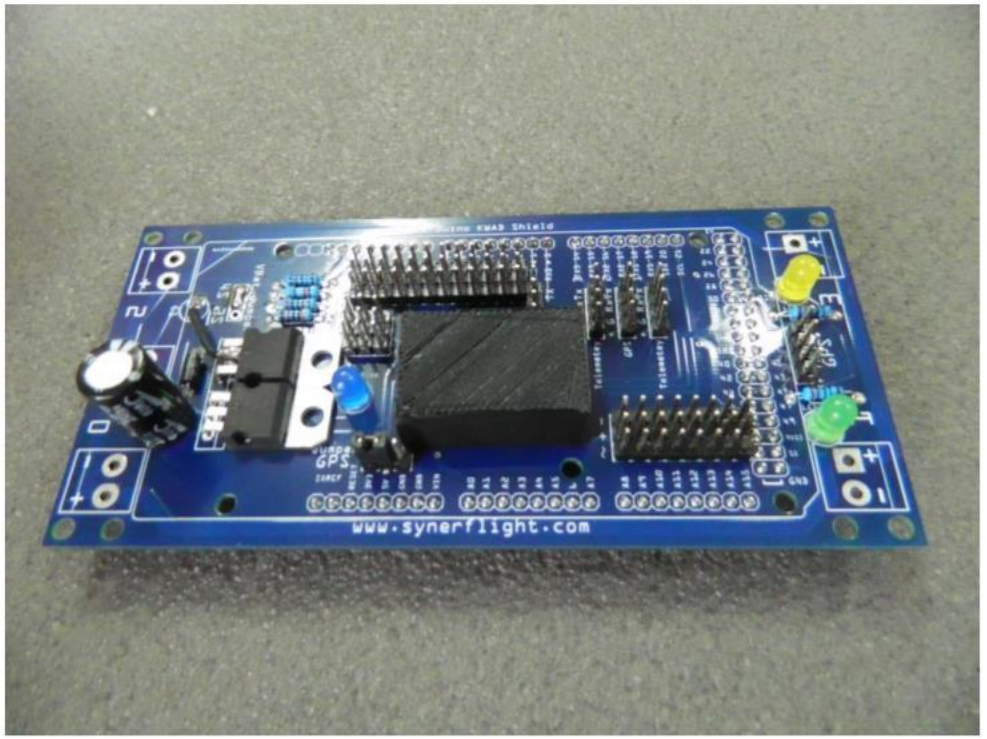
Ensure insulation from the Arduino board add tape on these areas

BETA GY801 & V1.1 GY91

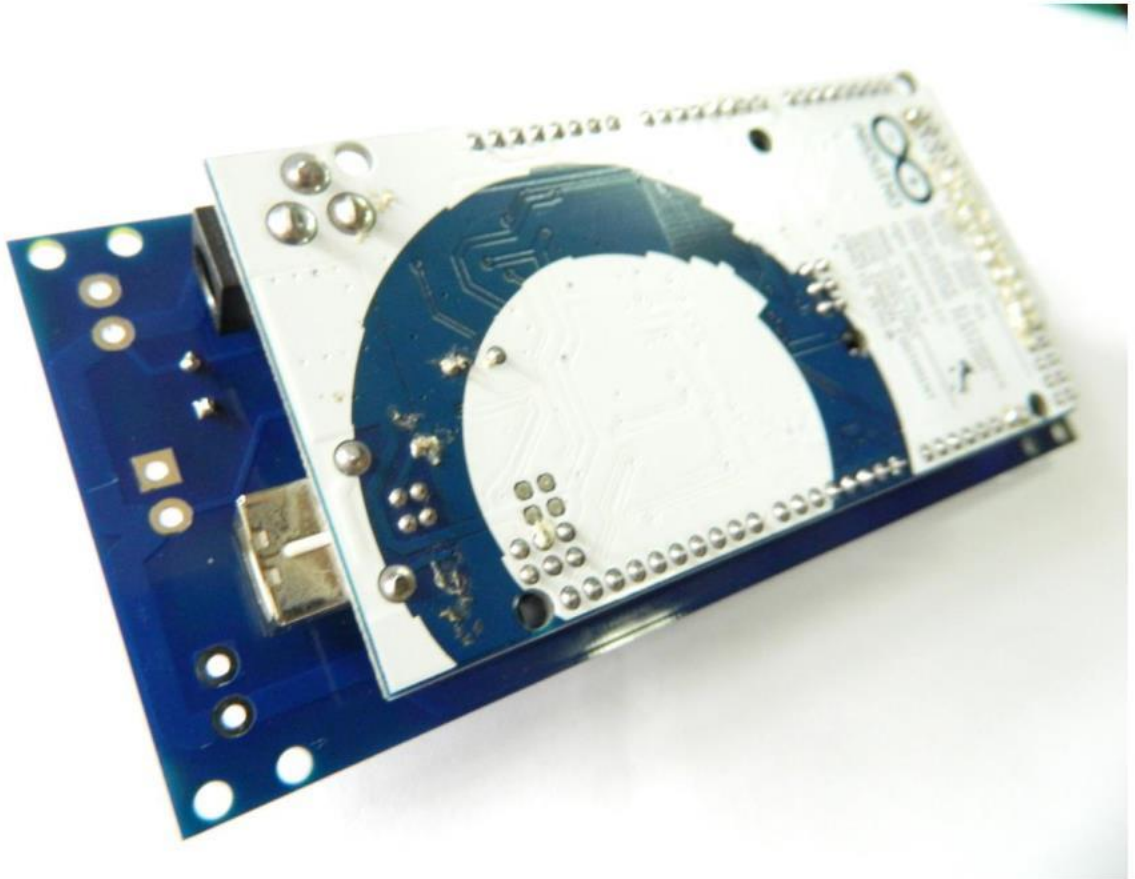








Use PVA to glue the cover to the IMU sensors seal the rim of the cover all the way

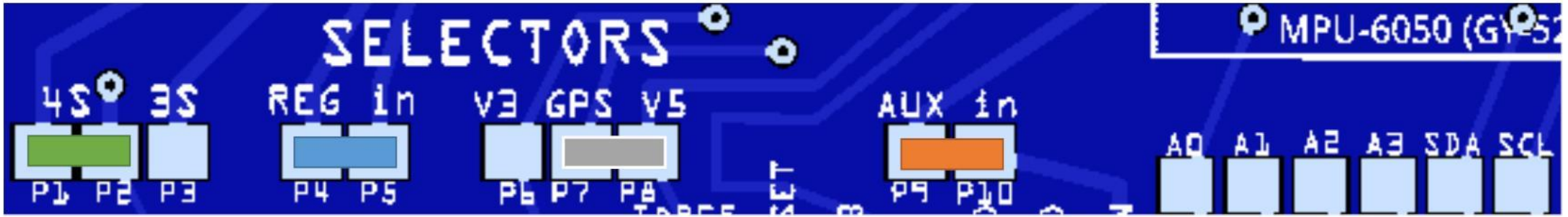


Synerduino Kwad shield V1.1

Update is 1cm longer board to clear the USB port from the solder pin

Synerduino Kwad Shield V1.1 (GY91) POWER Selector

Added the Selector Jumper Pads to the main board



Battery cell monitoring 4s or 3s

5V Regulator from battery

GPS Pins V+ voltage in front of the board

Analog 0 pin Auxin / Battery monitor

Analog 0-3 & I2C external sensors

To use onboard battery monitoring with Aux in Set to 3s if your running 1s-3s battery / set to 4s if your running 4s battery / Leave it open when using Aux in as External sensors or using 5s to 6s

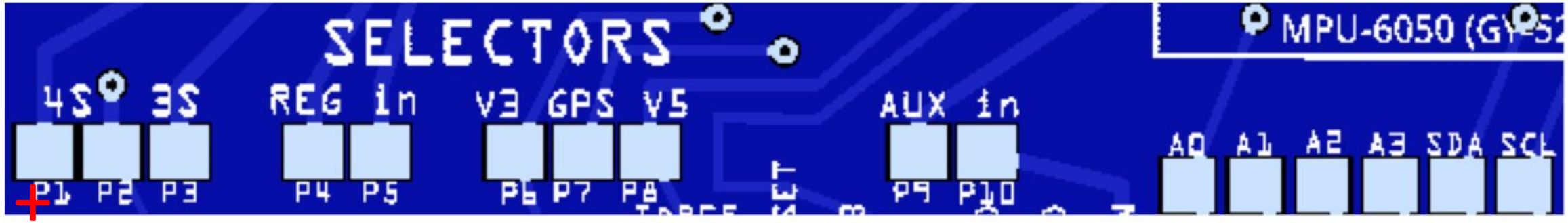
Reg In – short the Pads for using regulator to power and build in power distributor the synerduino and Arduino board

2nd GPS pin with voltage selector 5V for Regular GPS / 3V for external I2c sensor such as Magnetometer

Aux in- leave it open for utilizing the A0 Pins for External ADC sensors / Short the Pads to use build in battery monitoring . Cell Selector must be set to 4s or 3s

SDA SLC - I2C input for external sensors such as GPS with build in Magnetometer

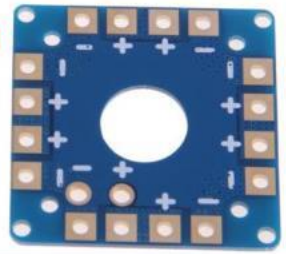
Default - A0 External ADC sensor , Require 5v UBEC applied to the 5V Serial pins & for OPTO ESC



\+

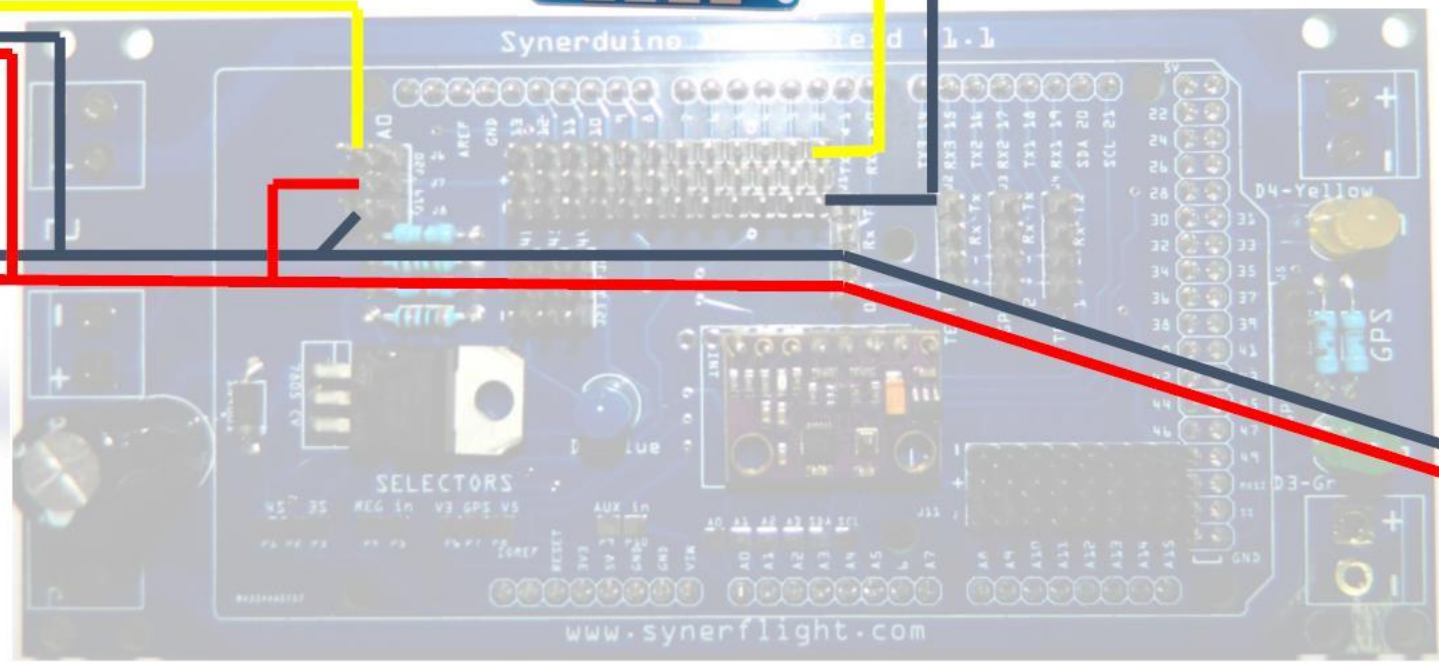
\-

External Power Distribution board



ESCs OPTO

External ADC Sensor

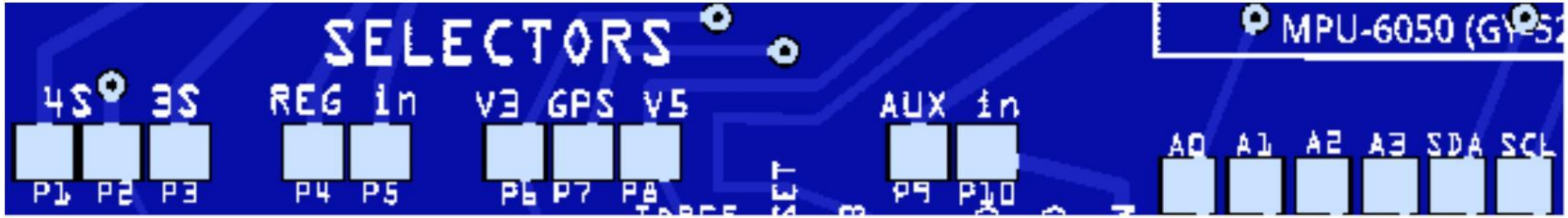


Setup for 2s - 6s Lipo
Required External Power distribution board / build in power distribution not in use

FPV camera

method also use for Servo , FPV and External sensor intended for Current intensive setups

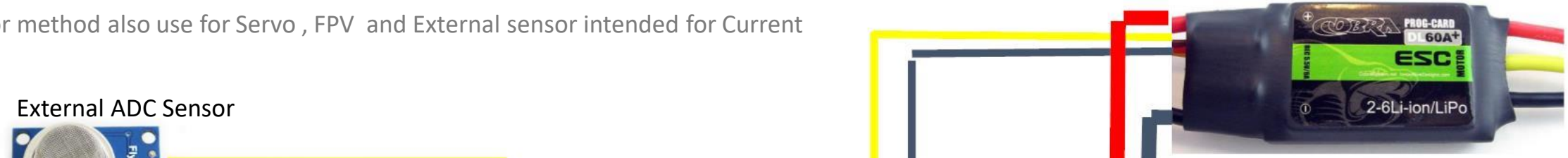
Default - A0 External ADC sensor , Require 5v UBEC applied to the 5V Serial pins & for OPTO ESC



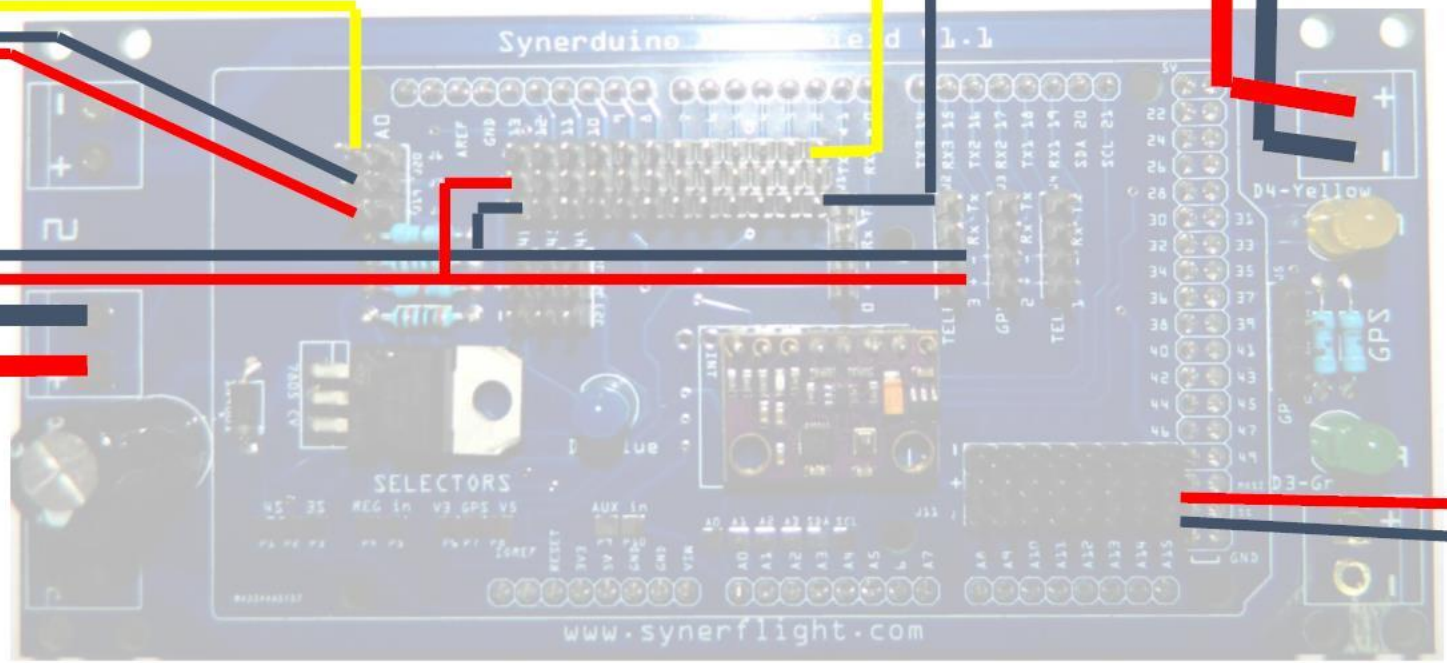
Bypass Regulator method also use for Servo , FPV and External sensor intended for Current intensive setups

External ADC Sensor

5V BEC



ESCs OPTO



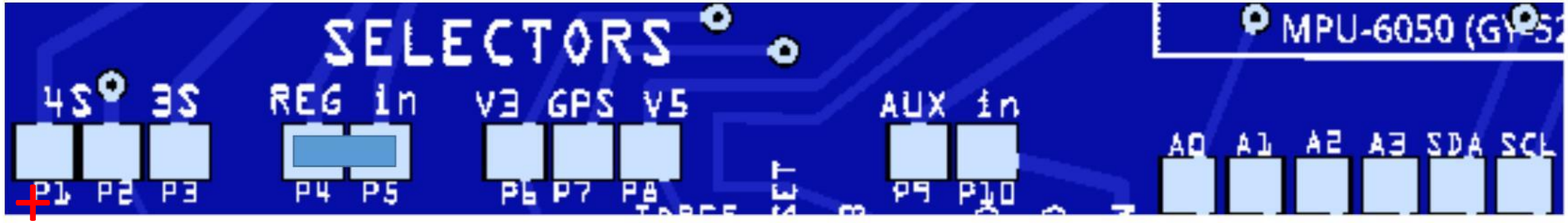
FPV camera

\-

\+

Setup for 2s - 4s Lipo
Required External Power distribution board / build in power distribution not in use

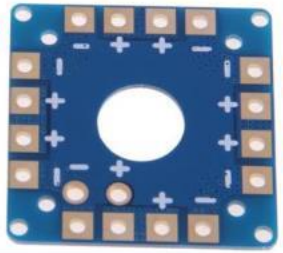
Reg in only - A0 External ADC sensor , Require ESC with UBEC applied to the 5V PWM pins



\+

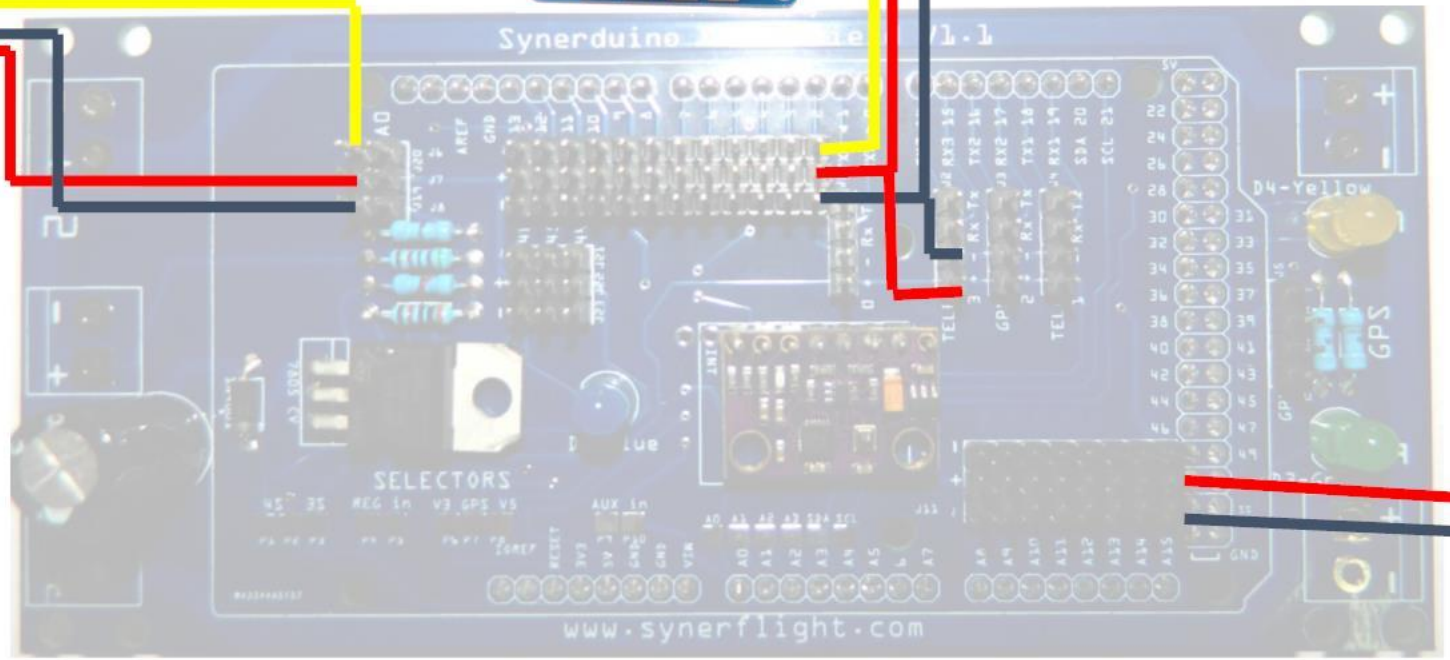
\-

External Power Distribution board



ESCs with BEC

External ADC Sensor

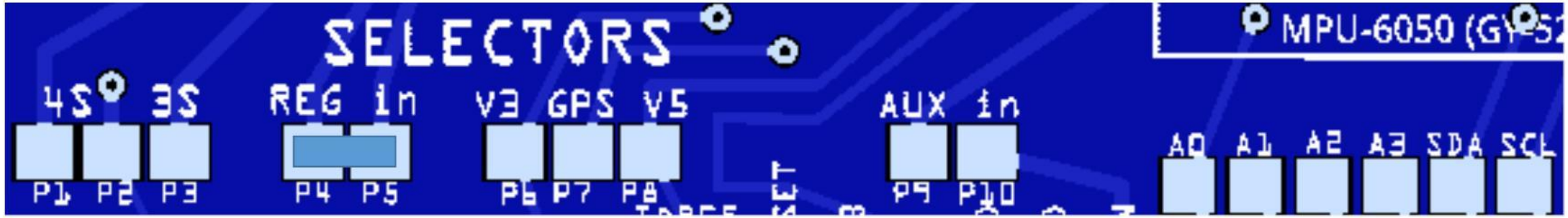


Setup works for 2s - 6s Lipo Required

External Power distribution board / build in power distribution not in use

FPV camera

Reg in only - A0 External ADC sensor , ESC BEC or OPTO applied to the 5V PWM pins



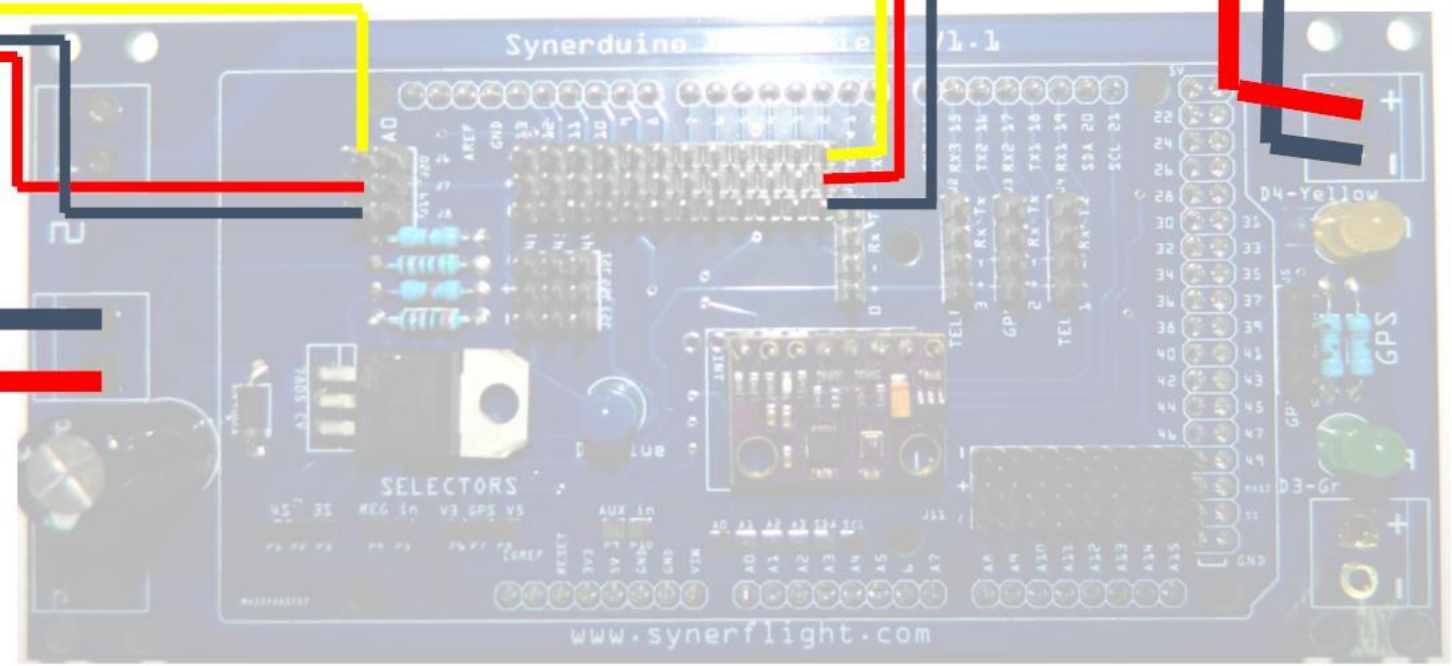
AO

External ADC Sensor

ESCs with BEC or Opto

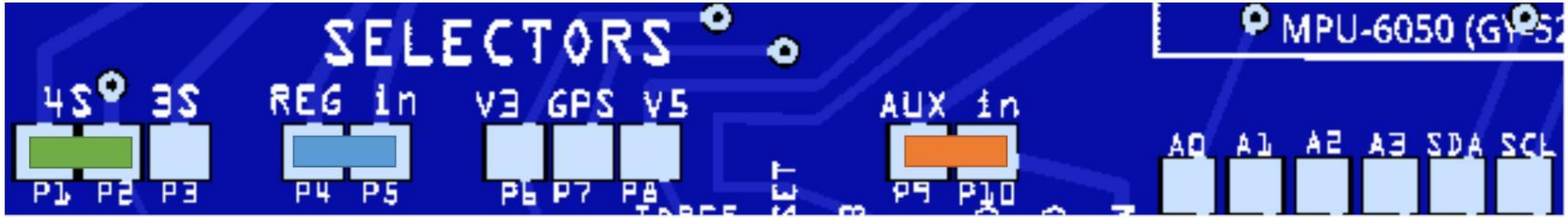
2s to 4s Lipo

Build in Power distribution



Recommended setup for beginner

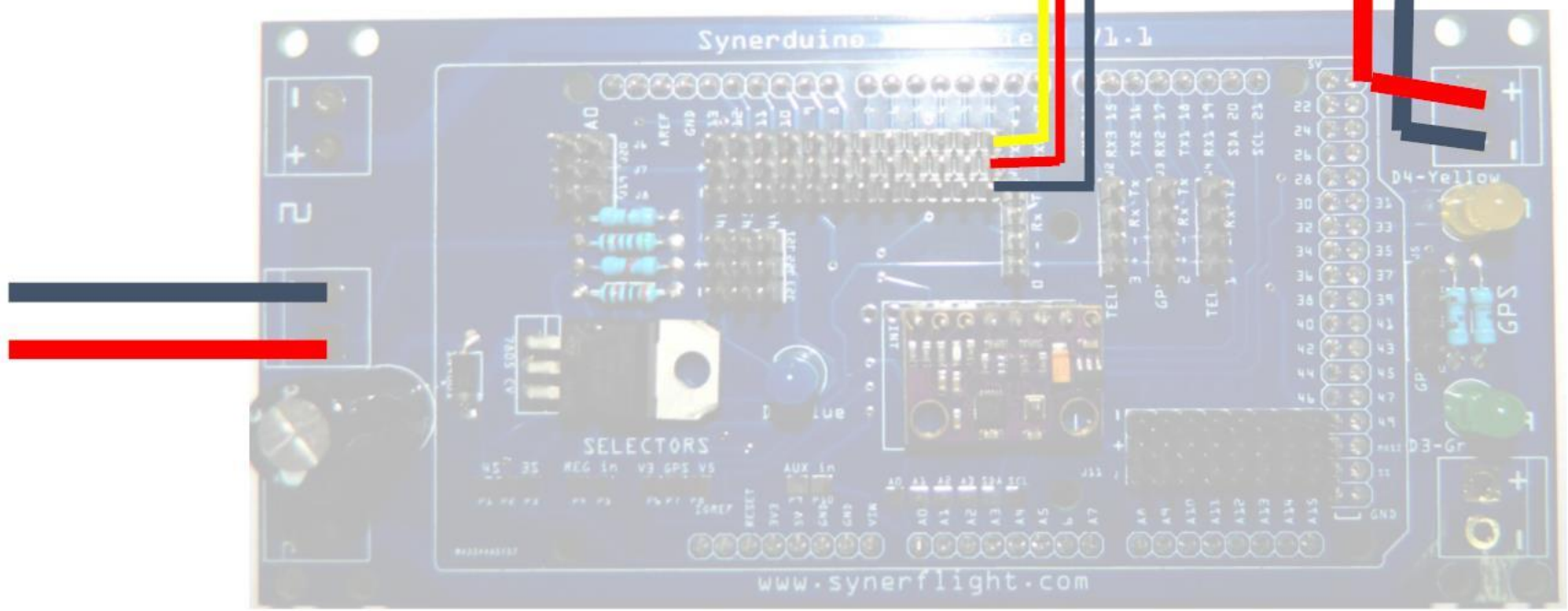
Reg & Vbat - A0 as Battery voltage monitor , ESC BEC or OPTO applied to the 5V PWM pins



For those who would use the build in battery monitoring circuit upto 4s lipo ensure the Cell Count and Aux in is jumped before powering up

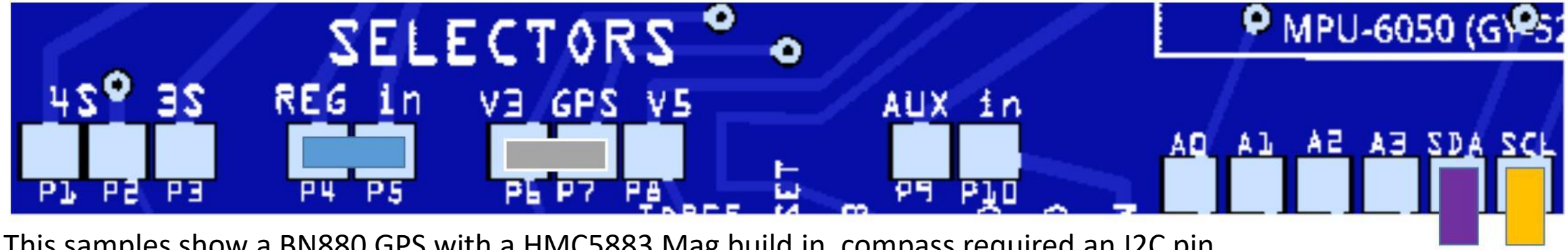
ESCs with BEC or Opto

2s to 4s Lipo
Build in Power distribution

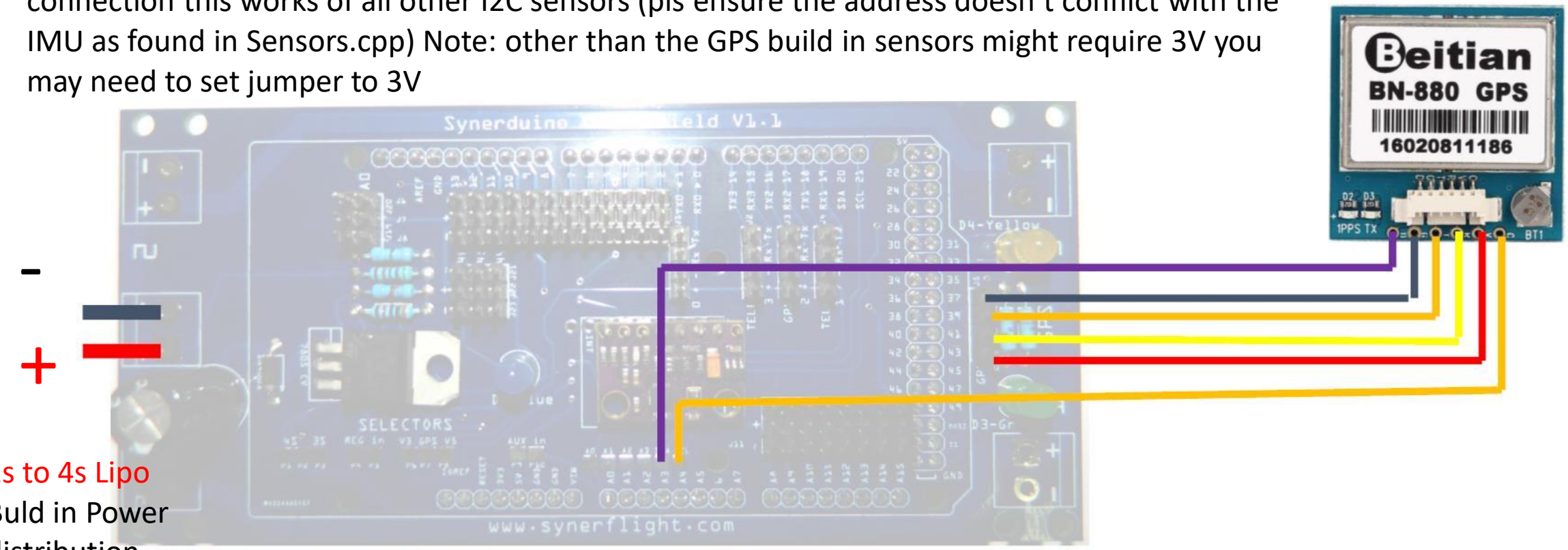


Recommended setup for beginner

Reg & External Sensors



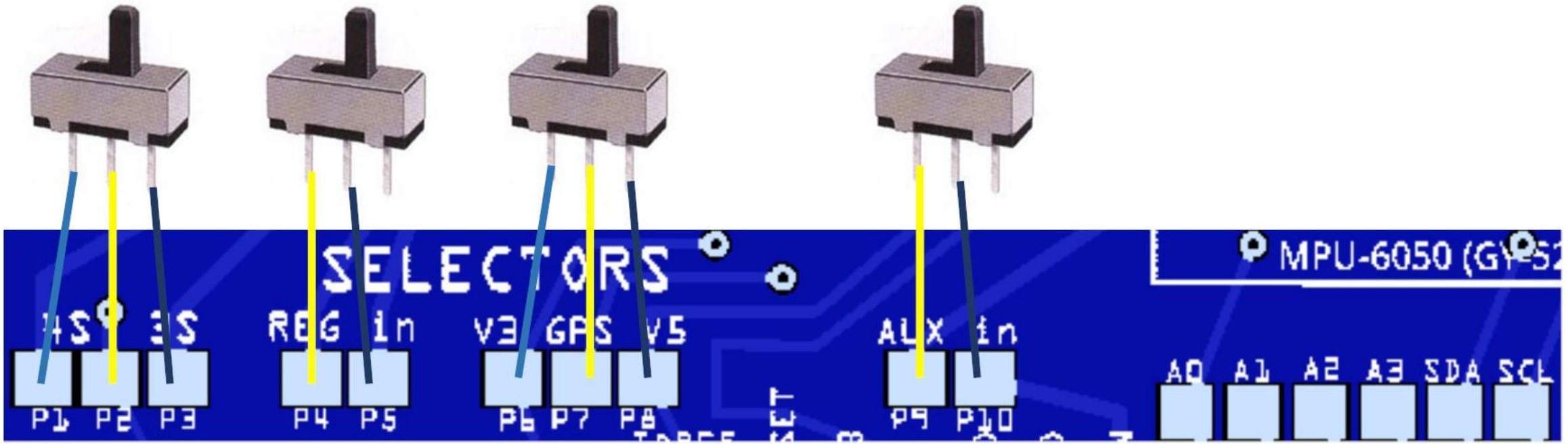
This samples show a BN880 GPS with a HMC5883 Mag build in compass required an I2C pin connection this works of all other I2C sensors (pls ensure the address doesn't conflict with the IMU as found in Sensors.cpp) Note: other than the GPS build in sensors might require 3V you may need to set jumper to 3V



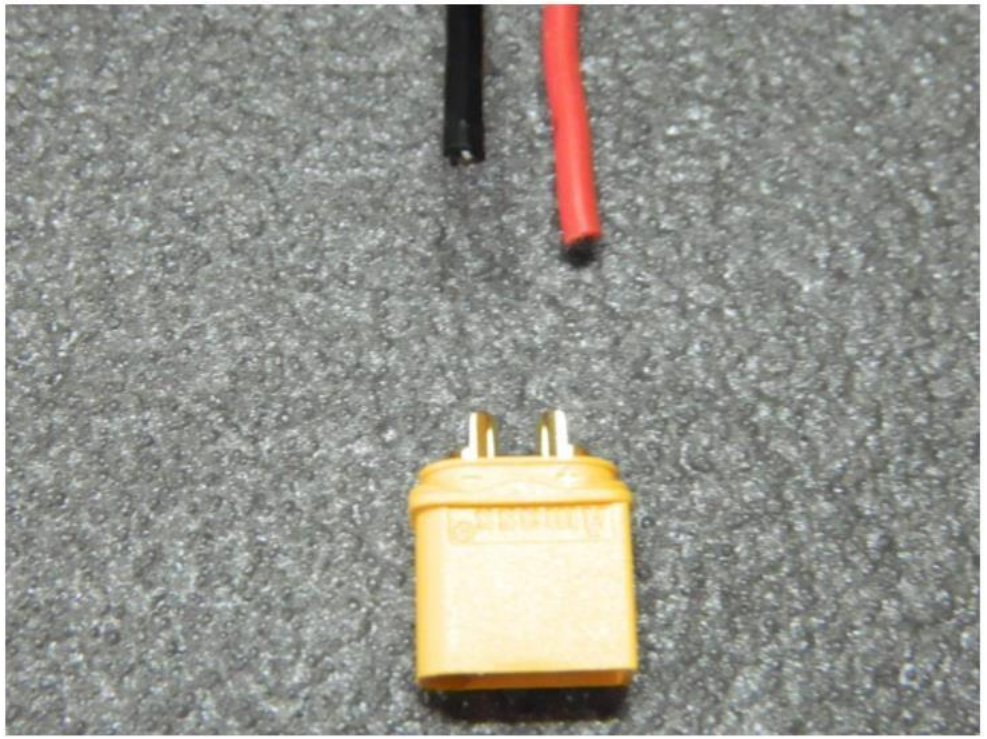
2s to 4s Lipo

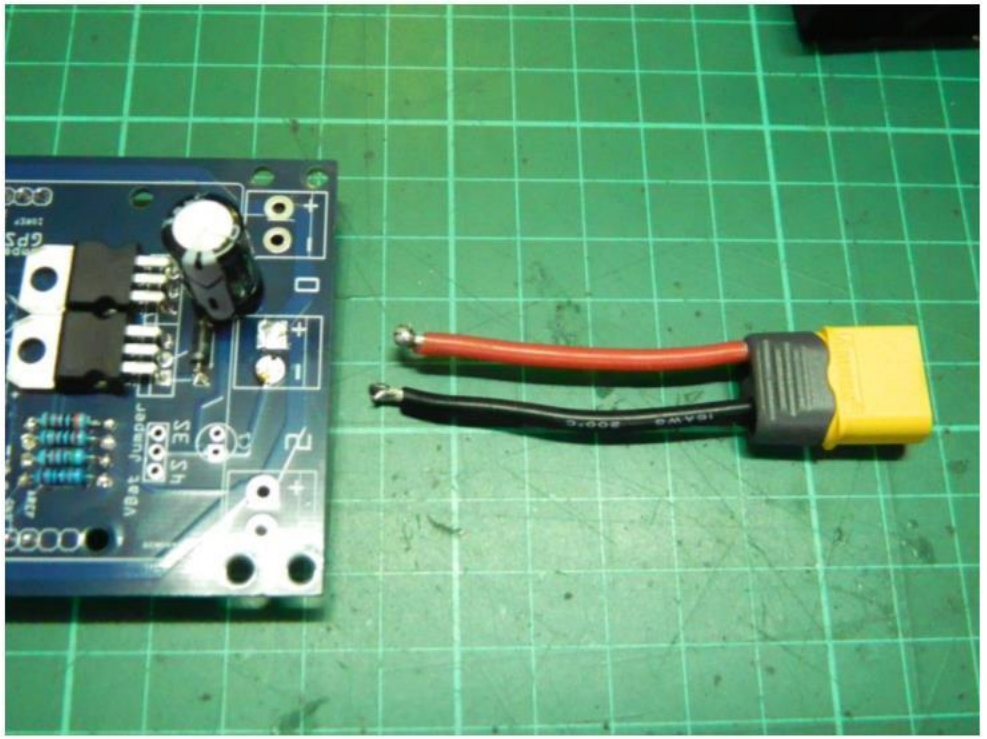
Buld in Power distribution

For those who thinks they need to change setup from time to time



For those who would use the build in battery monitoring circuit upto 4s lipo ensure the Cell Count and Aux in is jumped before powering up





Pls check the polarity of the XT60Plugs to match up with the board’s polarity

For those running power higher than 4s Lipo advice to use a external power distribution board and UBEC

Electronic Speed Controller

Motor type	The voltage (V)	Paddle size	current (A)	thrust (G)	power (W)	efficiency (G/W)	speed (RPM)
RS2205-2300KV	12	HQ5045 BN	1	62	12.00	5.17	6400
			3	162	36.00	4.50	10080
			5	236	60.00	3.93	12070
			7	311	84.00	3.70	13730
			9.1	374	109.20	3.42	15100
			11	439	132.00	3.33	16320
			13	490	156.00	3.14	17350
			15.3	548	183.60	2.98	18350
			17.3	611	207.60	2.94	19210
			20.7	712	248.40	2.87	20080

To start from the very beginning, ESC are of course electronic speed controller and these are the little devices that control the speed of your brushless motors. They come with different amperage and the size of ESC that you need depends on your multi-rotor setup

The Amperage should match the Motors Amperage Draw as found on the Tech specs of the Motor

2nd is the Type of ESC

- OPTO ESC is isolated meaning only two lines of the RC input are visible the PWM and Ground

- BEC ESC on the other hand includes the 5V in to power your electronics and it would be stated on the label /Data sheet of the ESC on how may Amps it would support in powering your RC electronics (Receiver, Flight Controller, Servos ,etc)

ESC are to be calibrated on first run as calibration process are required

Electronic Speed Controller Compatibility and Protocol

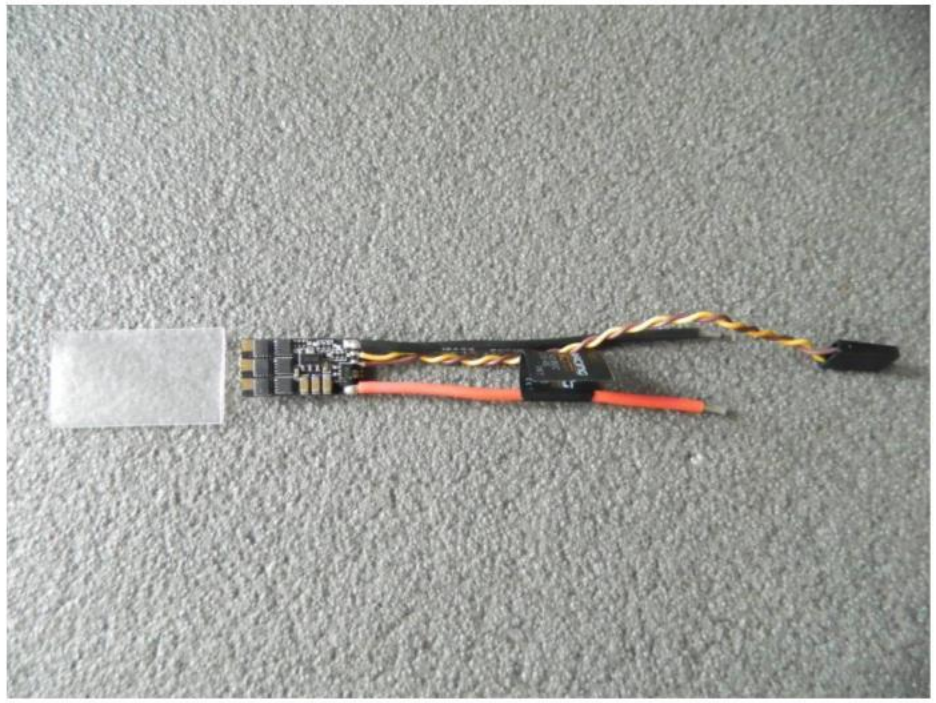
There are several types of ESC protocol and ESC firmware available for quadcopter

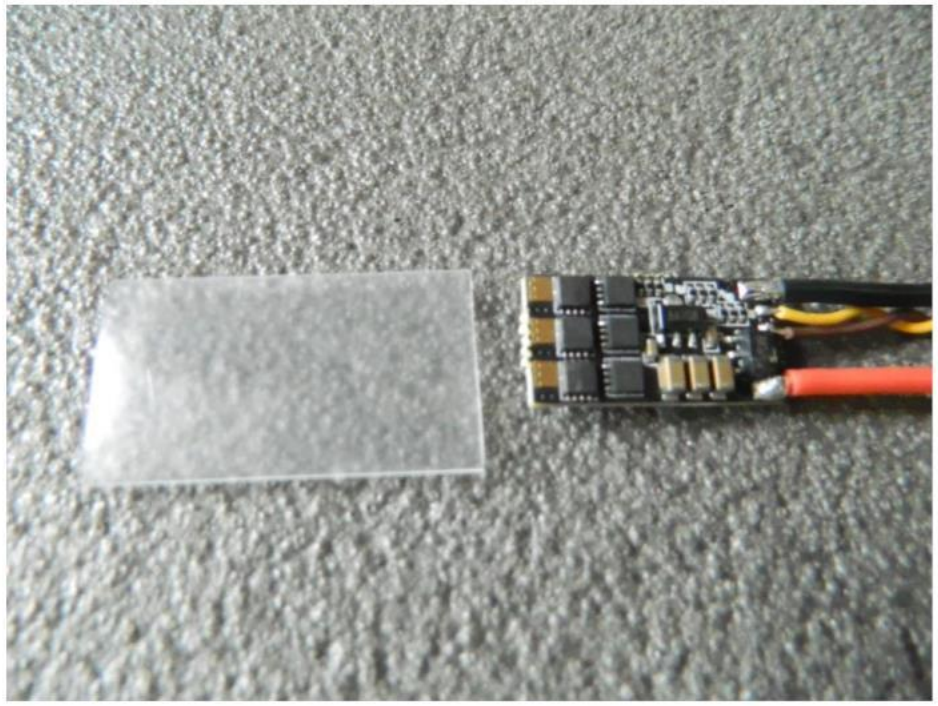
Because Synerduino is a shield base of the 8bit Arduino platform the speed of the PWM can be restrictive

The Synerduino would largely run off Standard PWM Protocol (1000us – 2000us)

Compatible ESC Firmware	Protocol
SimonK	Standard PWM (1000us – 2000us)
BL Heli	Standard PWM (1000us – 2000us) Oneshot125 (125us – 250us)
BL Hei-S	Standard PWM (1000us – 2000us) Oneshot125 (125us – 250us) OneShot 42

BLHeli_32 and Dshot ESC may not be compatible with the Arduino due to its speed requirments

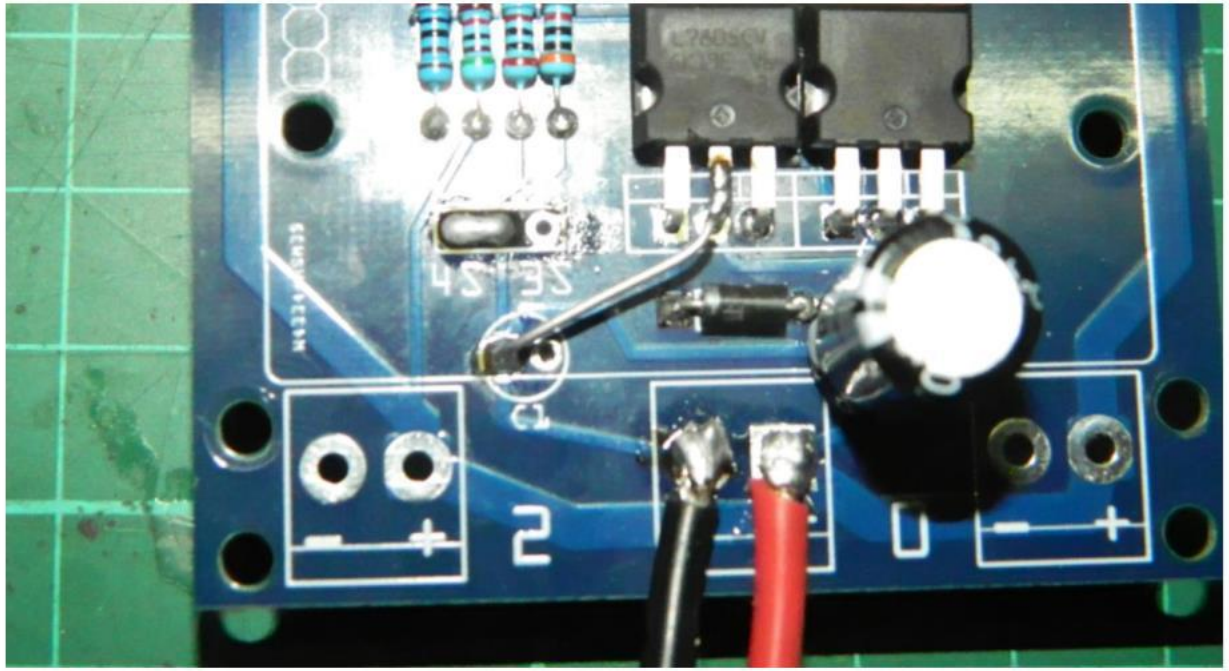


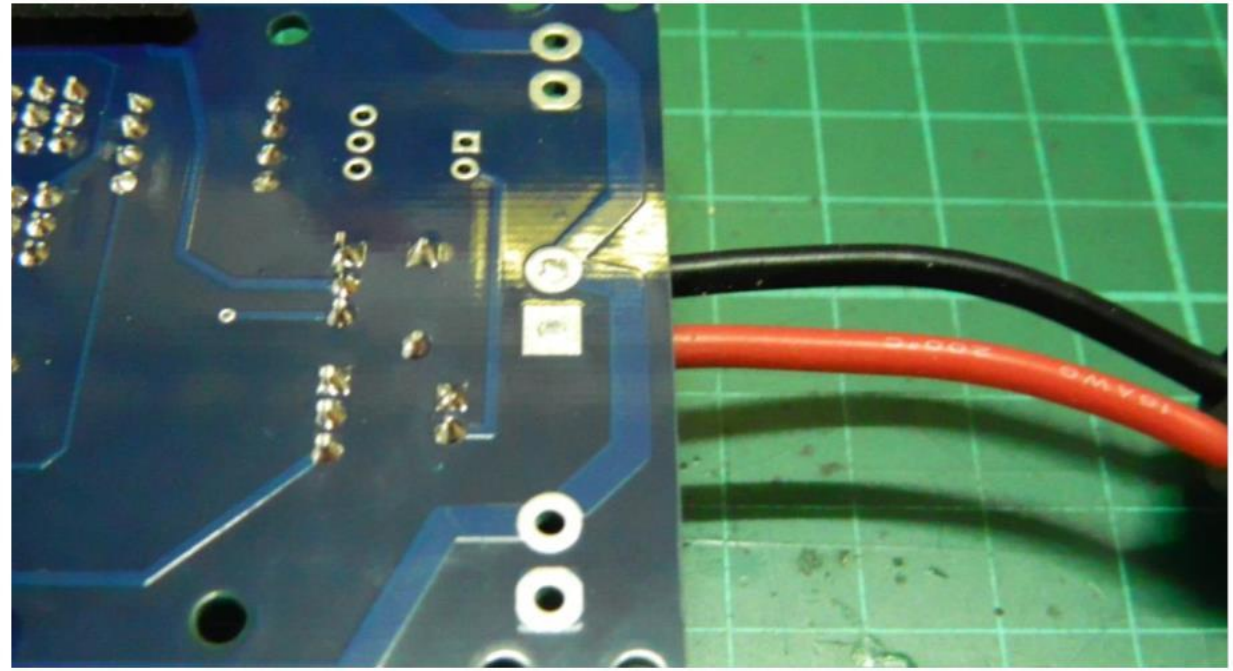


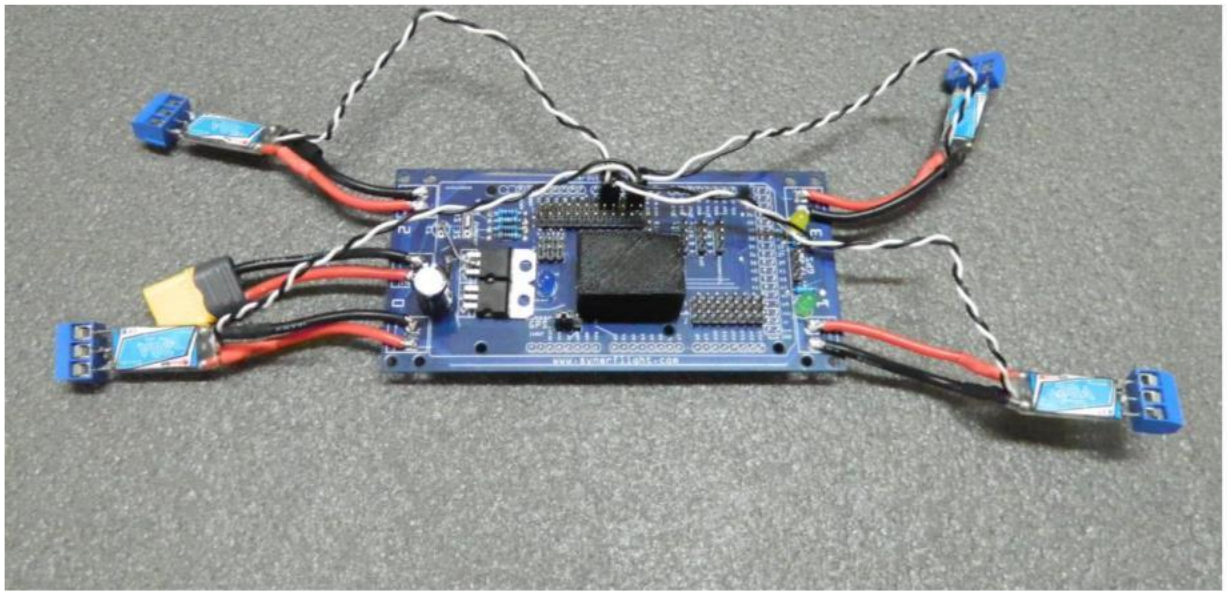
For Small Race ESCs solder the motor lines in first before applying the shrink tube

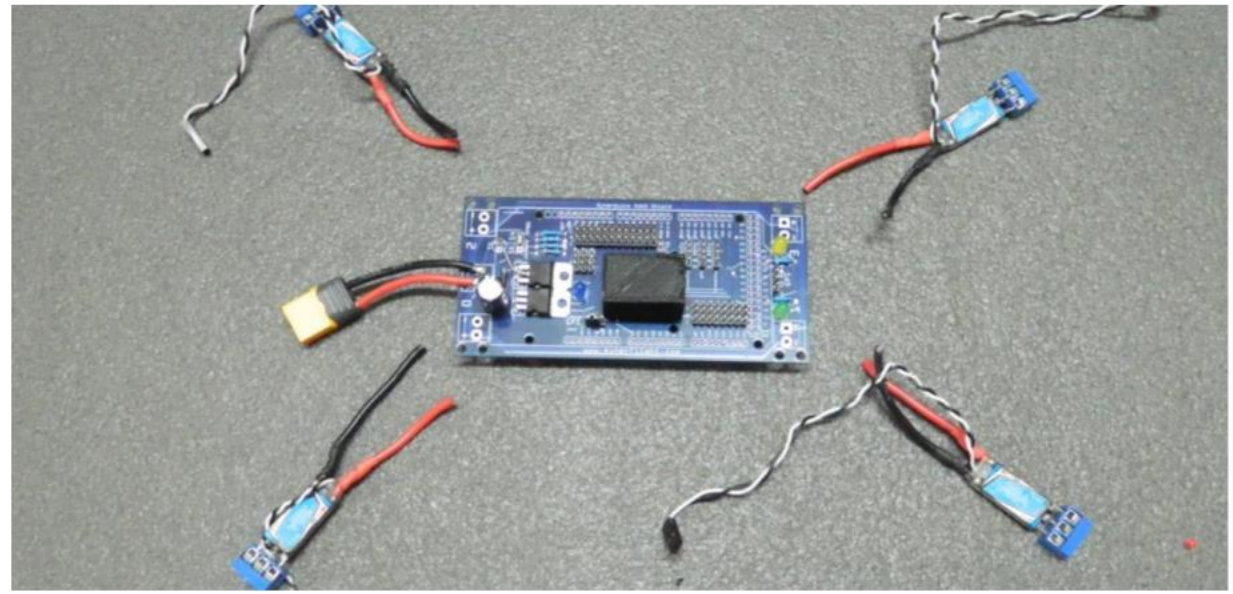
BEC “Universal Battery Elimination Circuit”

OPTO 'opto-coupled', throttle signal is coupled into the ESC is isolated





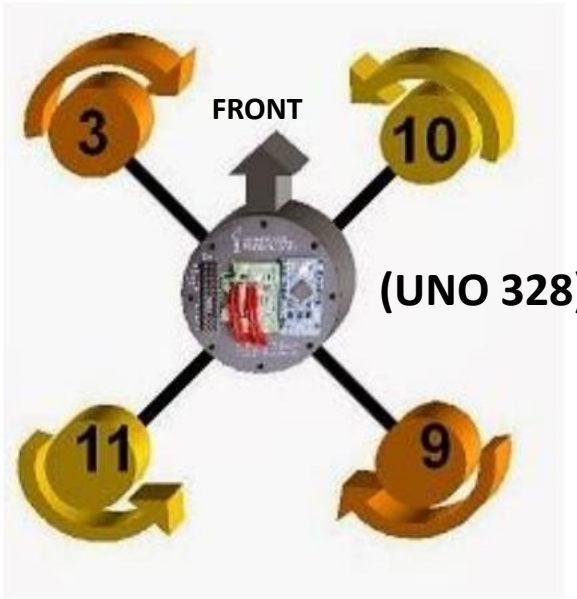




Electronic Speed Controller

Motor [3]

Motor [1]



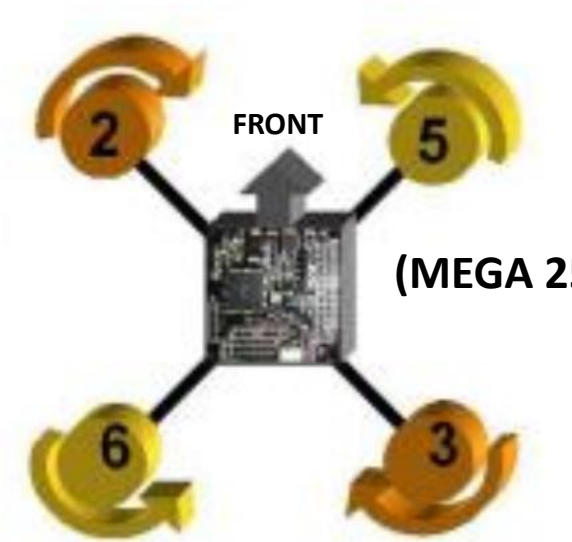
(UNO 328)

Motor [2]

Motor [0]

Motor [3]

Motor [1]



(MEGA 2560)

Motor [2]

Motor [0]

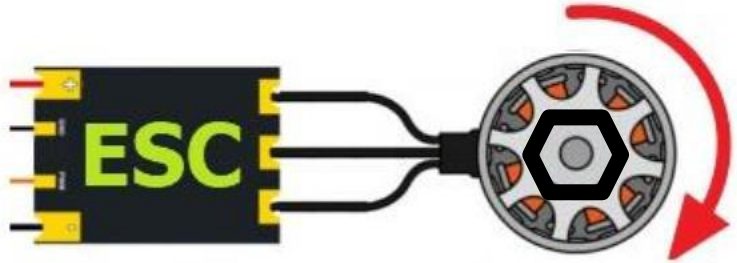
Note : you can pre solder the motor to the board and check for rotation before installing the propeller to insure all motor rotations are correct

Note : on some brands of motor they may came in two different prop nuts color (Known as self tightening nuts)

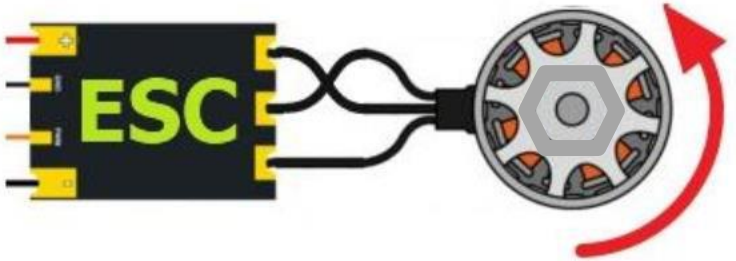
Black for Clockwise Motor

Silver for counter clockwise Motor

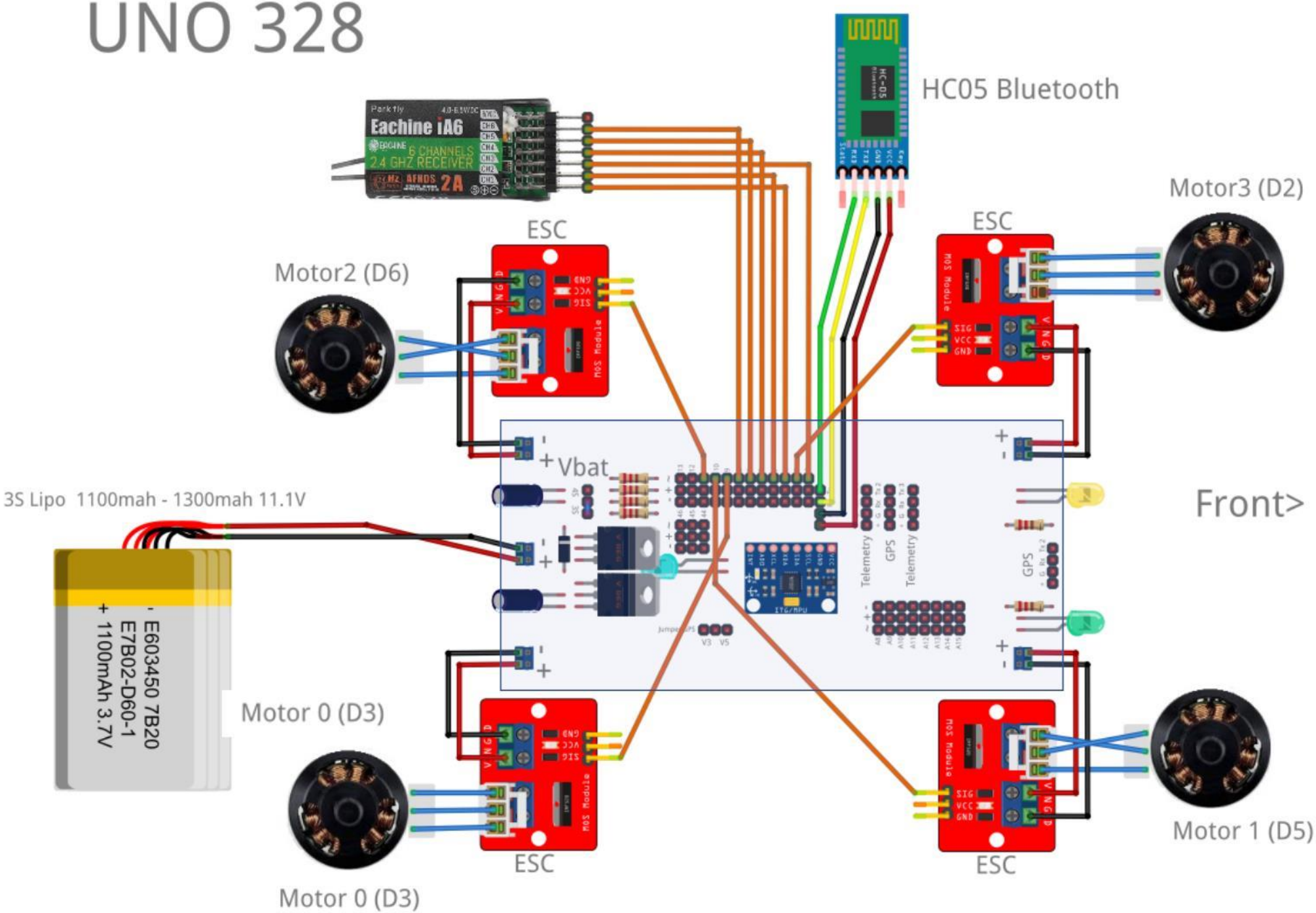
Parallel

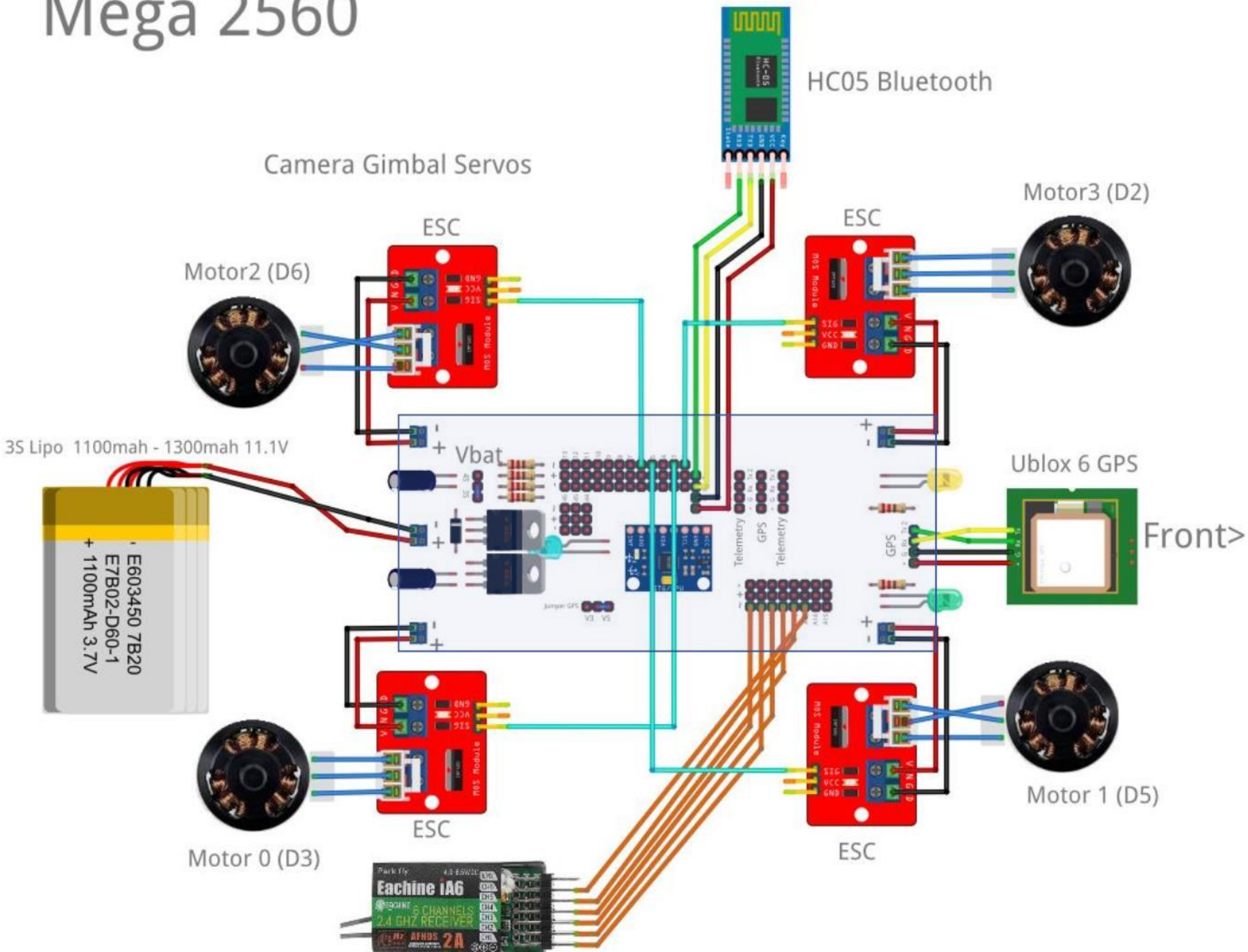


Crossover



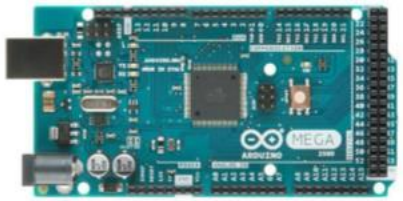
UNO 328


HC05 Bluetooth
Eachine iA6
6 CHANNELS
2.4 GHZ RECEIVER
ESC
Motor2 (D6)
Motor3 (D2)
Vbat
3S Lipo 1100mah - 1300mah 11.1V
E603450 7B20
E7B02-D60-1
1100mAh 3.7V
Front>
Motor 0 (D3)
Motor 0 (D3)
Motor 1 (D5)
Telemetry
GPS
Telemetry
GPS
MOS Module
SIG
VCC
GND


Mega 2560
HC05 Bluetooth
Camera Gimbal Servos
ESC
Motor2 (D6)
Motor3 (D2)
3S Lipo 1100mah - 1300mah 11.1V
Vbat
Ublox 6 GPS
Front>
Motor 1 (D5)
Motor 0 (D3)
ESC
Eachine iA6

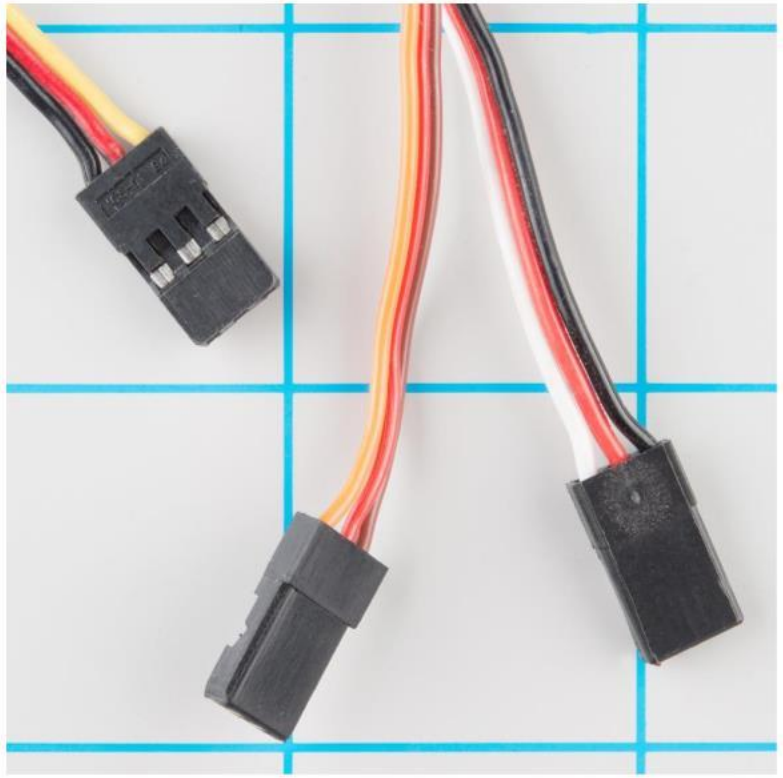
PWM INPUT Assignment

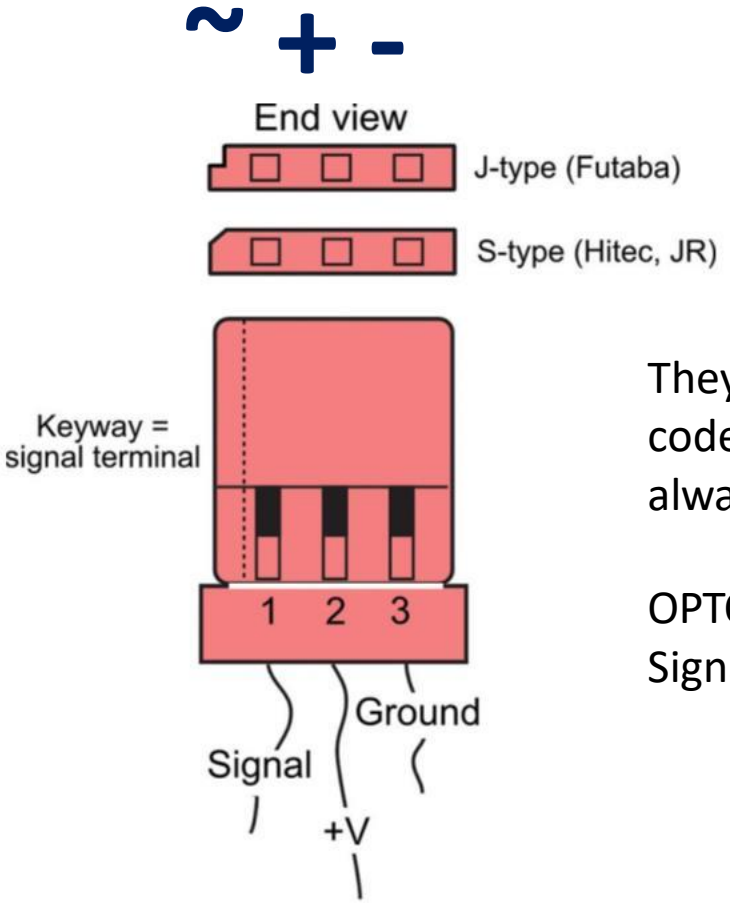
Pls Check the output pin from your Radio Rx manual



RX > Arduino / PWM in	Futaba Format	JR Format	Walkera Format	UNO 328 Input	Mega 2560 Input
Throttle	Ch3	Ch1	Ch3	D2	A8
Aileron	Ch1	Ch2	Ch2	D4	A9
Elevator	Ch2	Ch3	Ch1	D5	A10
Rudder	Ch4	Ch4	Ch4	D6	A11
Aux1	Ch5	Ch5	Ch5	D7	A12
Aux2	Ch6	Ch6	Ch6	D8	A13
Aux3	Ch7	Ch7	Ch7	N/A	A14
Aux4	Ch8	Ch8	Ch8	N/A	A15

SERVO HEADER





They may come with different coded wire but layout are always same

OPTO Wires may only have Signal and Negative Wires o

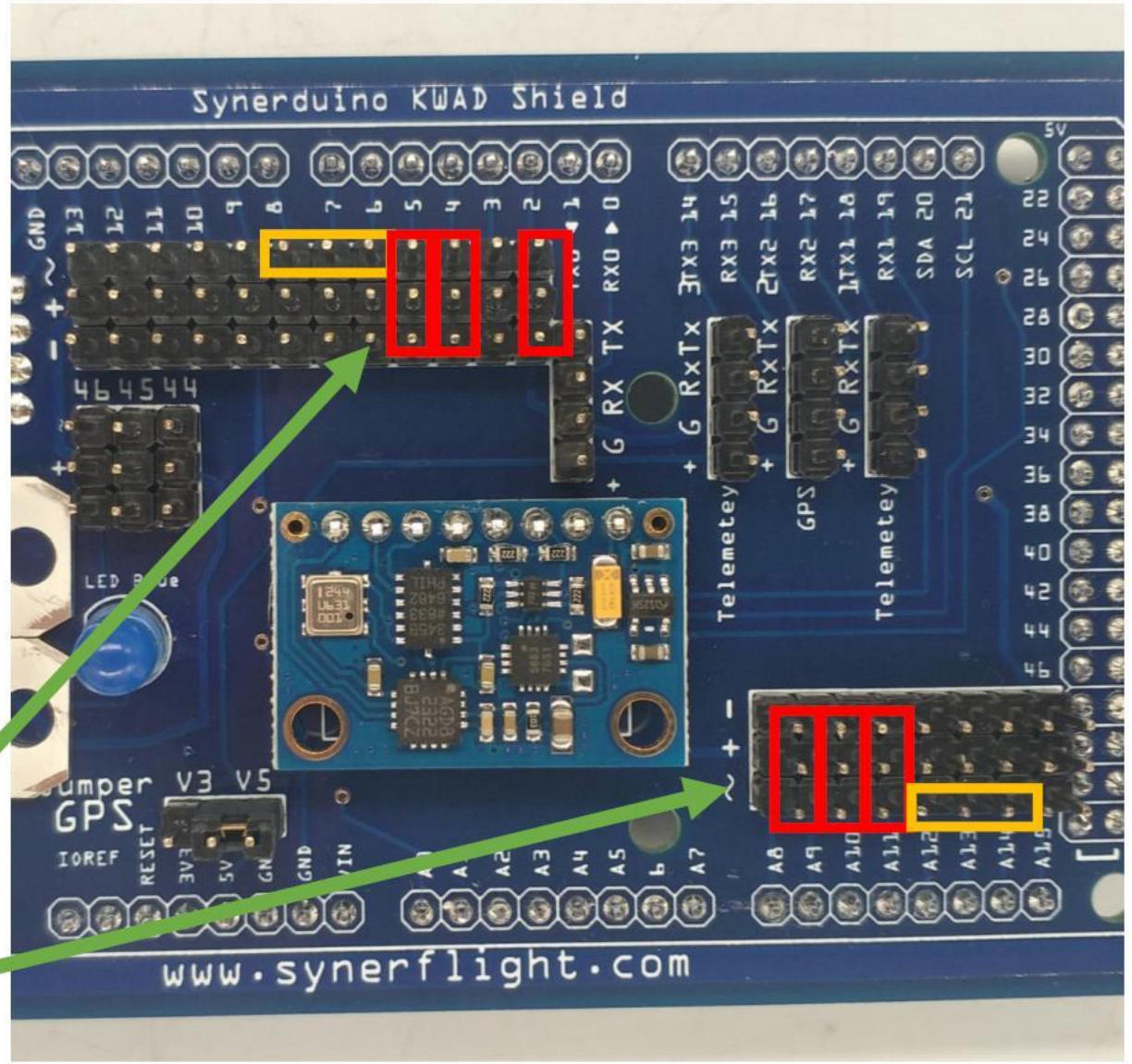
PWM RECEIVER

INCASE YOU'D ASK WHY THE SERVO CONNECTORS WERE DONE THIS WAY , ITS SIMPLY YOU DON'T NEED TO PLUG ALL THE PWM POWER RAILS ON ALL CHANNELS YOU JUST NEED THE PWM SIGNAL PIN ALONE

MOSTLY RUDDER AUX1 AND AUX2

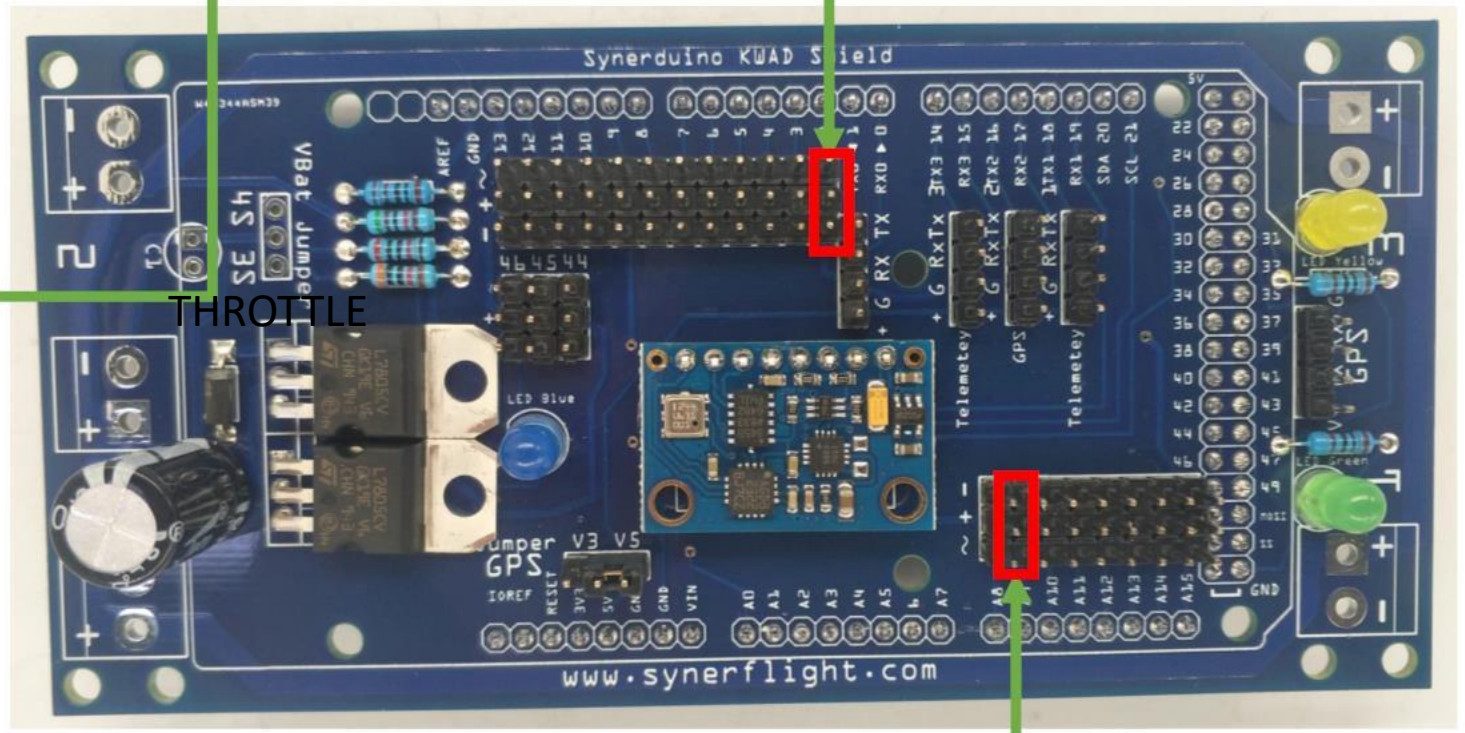
UNO PWM IN

MEGA PWM IN



PPM Receiver

THROTTLE

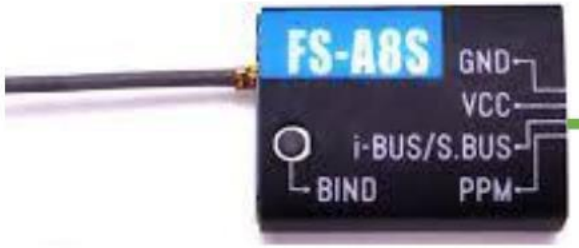


Pin D2 for Uno

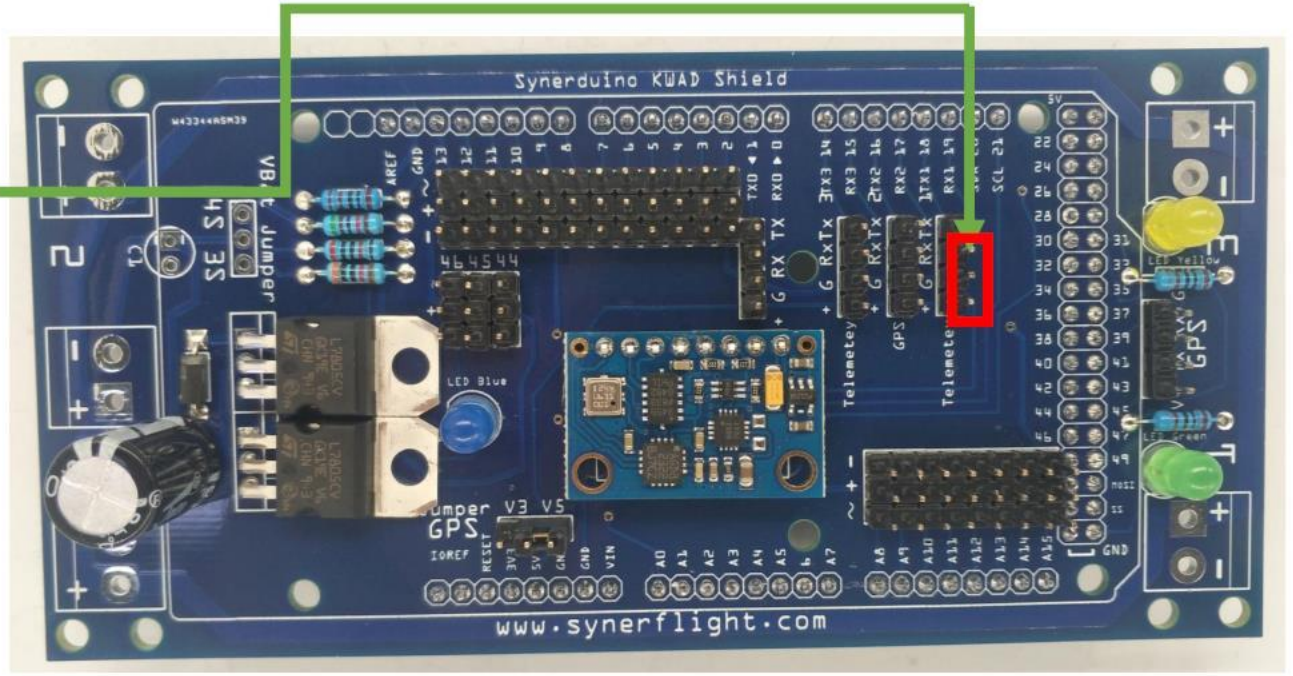
Pin A8 for Mega

SBUS Receiver

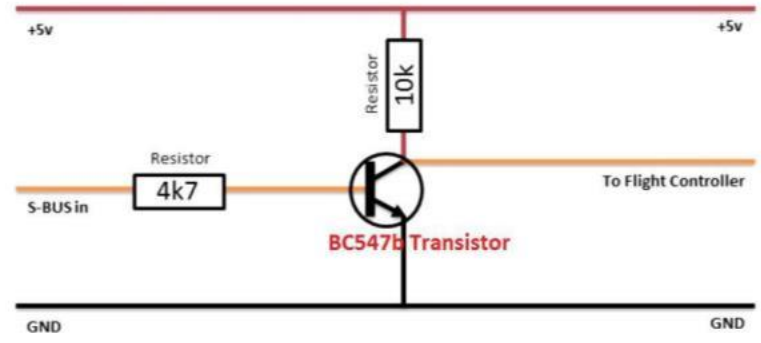
Most modern Receivers now comes with Serial Protocol as they are faster than the old PWM or PPM standard and its now the Modern defacto for Receiver to Flight Control Board communication

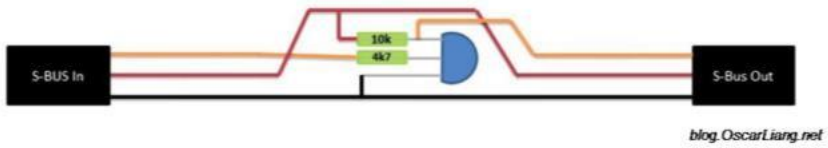


RX 1 Telemetry



The SBUS system uses Futaba protocol and should be compatible with Most SBUS Receivers





SBUS Inverter

Should there be issues in Signal Inversion an SBUS inverter may be use

You can tell if the RC control is not being read

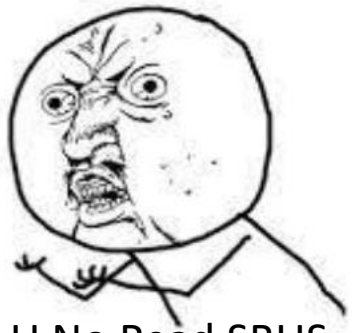
SBUS Receiver Inversion

We all get confused sometimes we plug the receiver or PPM/PWM/SBUS Converter in and it suppose to work but it doesn't

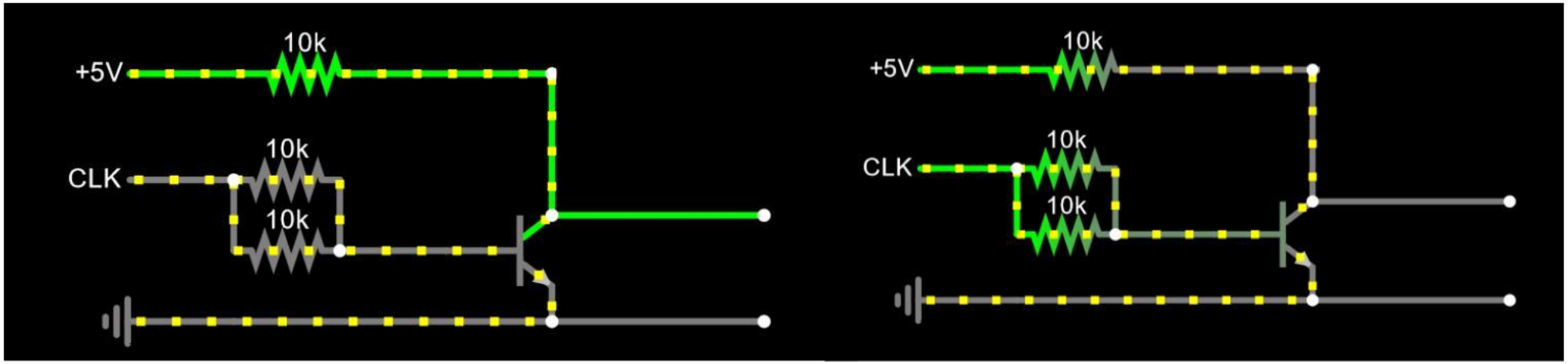
SBUS inversion depending on the Brand of Receiver or the PPM/PWM/SBUS Converter you have the SBUS signal can come as Forward Signal or Reverse Signal . This is crucial in getting a Good Receiver connection to the Synerduino STM

Normal SBUS

Inverted SBUS



Y U No Read SBUS



TELEMETRY RADIO

38400 OR 57600 FOR SIK RADIO DEPENDING IF USES 433MHZ OR 900MHZ

38400 FOR XBEE RADIO

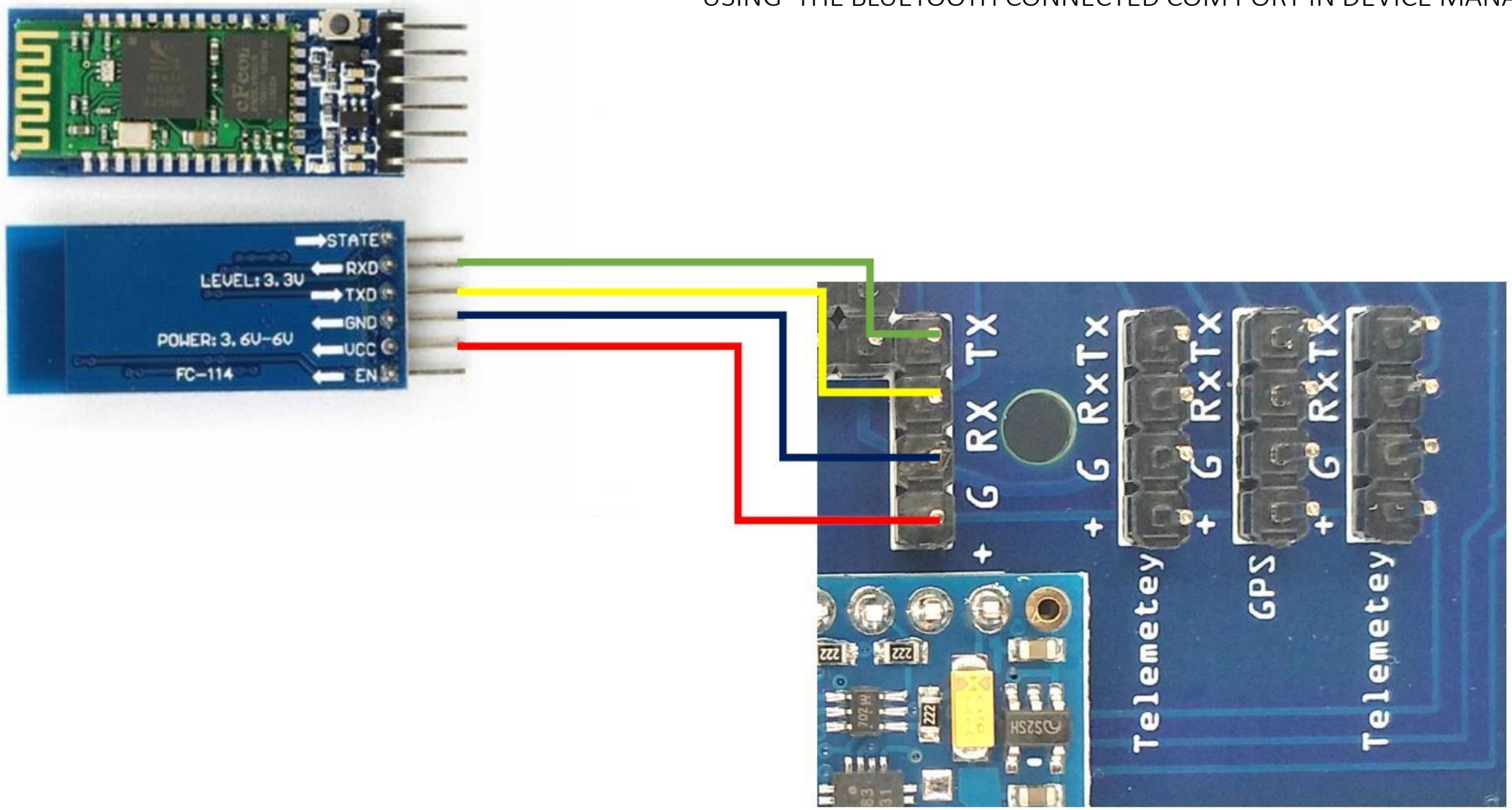
115200 FOR BLUETOOTH HC-05

STANDARD FOR ALL DRONES TO USE SERIAL LINK AS TELEMETRY MAINLY ON YOUR TX RX SERIAL PORTS , NOTE: THE LOWER THE FREQUENCY OF THE RADIO THE LOWER THE BAUD IS NEEDED

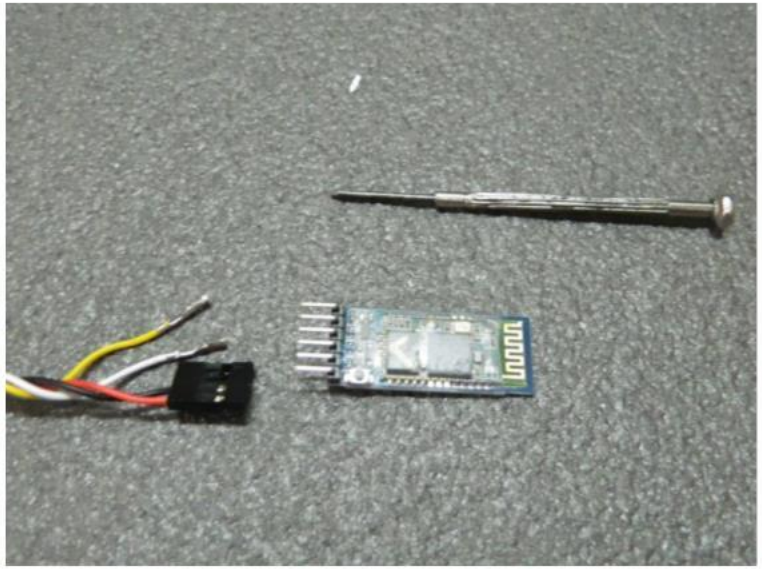
PROTOCOL IS MSP RAW OR MAVLINK

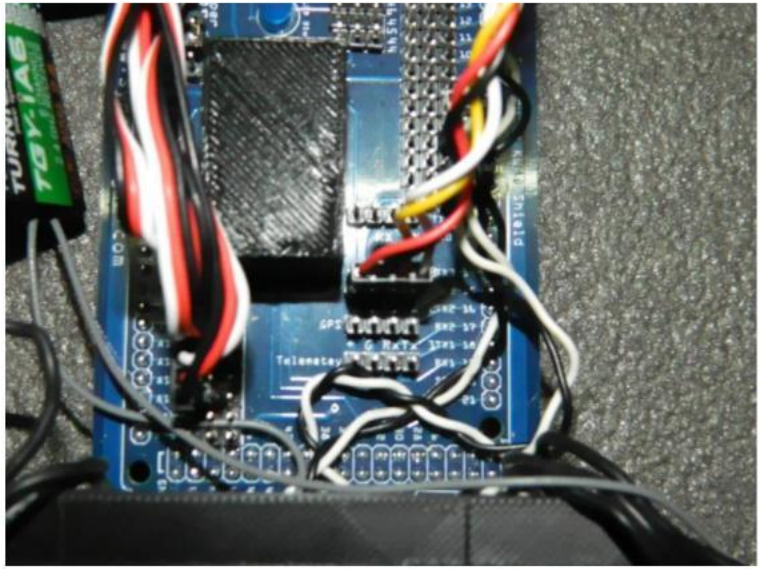
Bluetooth

NOTE: USING 0 AT BAUD115200 APPLICABLE BOTH UNO AND MEGA BOARDS REQUIRES THE BLUETOOTH ONLY TO BE PLUG IN AFTER THE SKETCH/FIRMWARE HAS BE UPLOADED TO THE BOARD . AND PAIRING USING THE BLUETOOTH CONNECTED COM PORT IN DEVICE MANAGER



Bluetooth





BLUETOOTH PLUG INTO SERIAL 1 OR SERIAL 3

115200 FOR BLUETOOTH HC-05

ATTENTION:

YOU MAY NEED TO REARRANGE THE HEADERS TO CONNECT THE BLUETOOTH MODULE TO THE SHIELD BOARD ACCORDINGLY

VCC >> +

GND >> G

TX >> RX

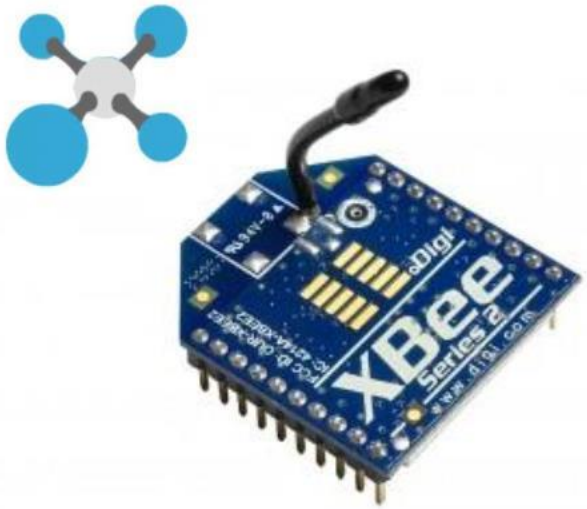
RX >>TX

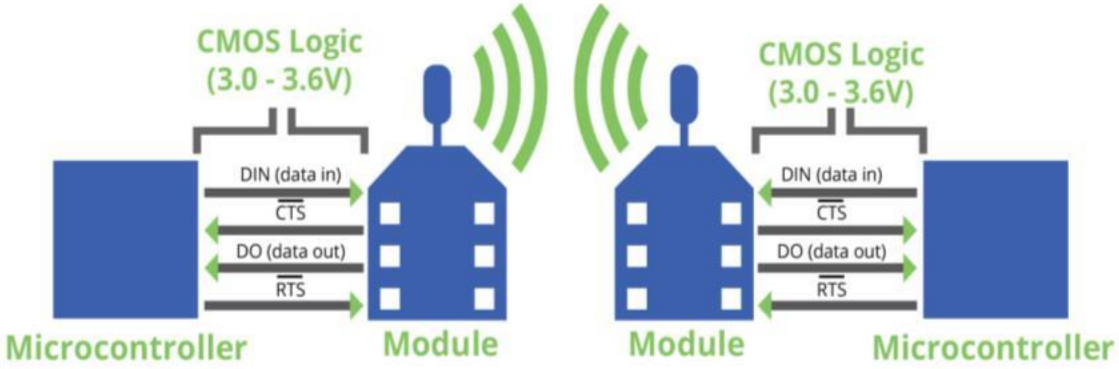
SEE TO IT THE WIRES COLOR CODE MATCHES THE MARKINGS

IMPROPER INSTALLATION MAY CAUSE DAMAGE TO THE ARDUINO BOARD AND SHIELD DUE TO REVERSE POLARITY

NOTE: WE PRESET THE BLUETOOTH FOR YOUR CONVENIENCE TO THE PROPER SETUP BUT SHOULD YOU WISH TO CHANGE THE SETTING ON YOUR DIGRESSION

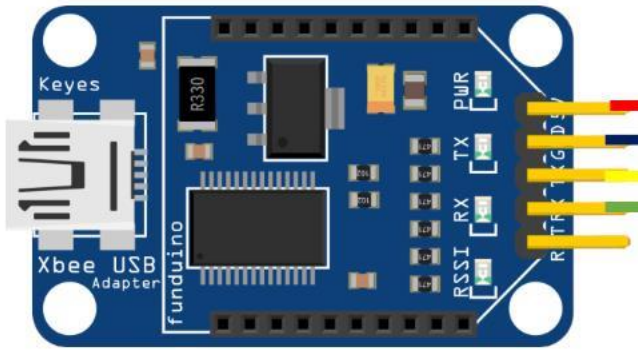
XBEE RADIO





38400 FOR XBEE RADIO

GET THE USB MODULE WITH BOOT AND RESET BUTTON AS YOU MAY NEED TO RESET THE XBEE WHEN UPDATING FIRMWARE

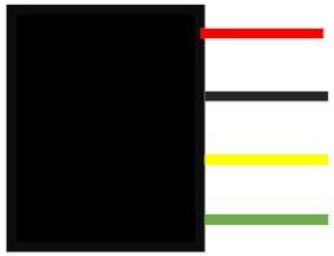


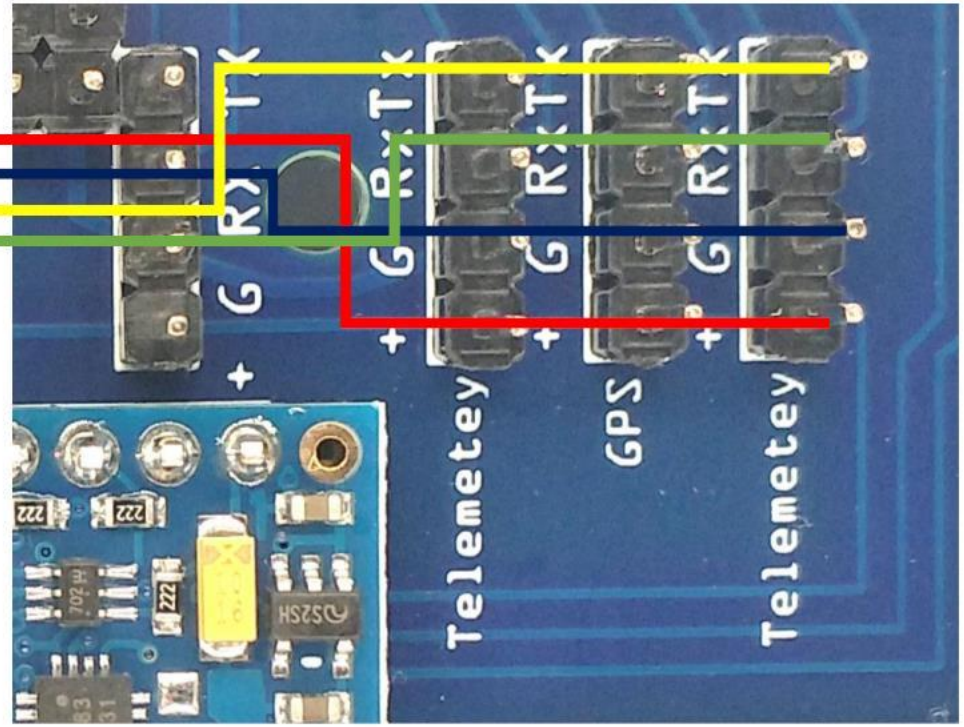
VCC >> +

GND >> G

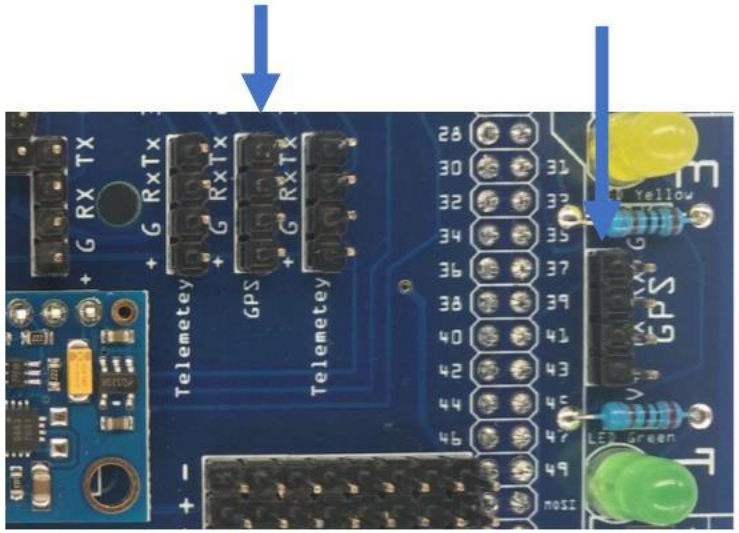
TX >> TX

RX >>RX





GPS



U BLOX NEO 6

PLUG IN TO SERIAL TX 2 RX 2 ON THE DRONE SHIELD BOARD

ATTENTION:

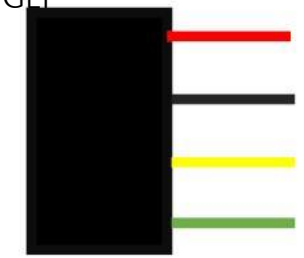
YOU MAY NEED TO REARRANGE THE HEADERS TO CONNECT THE GPS MODULE TO THE SHIELD BOARD ACCORDINGLY

VCC >> +

GND >> G

TX >> RX

RX >>TX



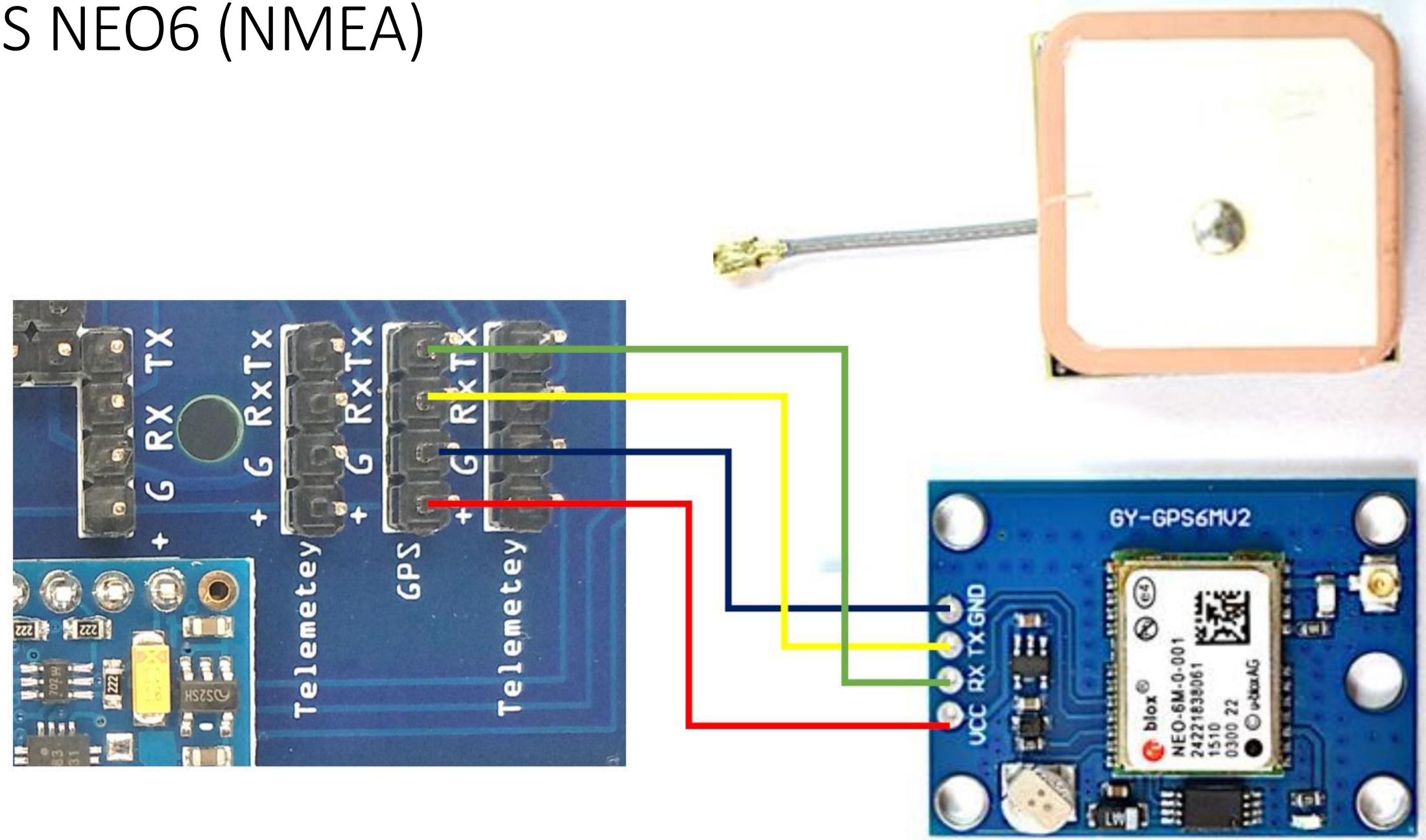
SEE TO IT THE WIRES COLOR CODE MATCHES THE MARKINGS

IMPROPER INSTALLATION MAY CAUSE DAMAGE TO THE ARDUINO BOARD AND SHIELD DUE TO REVERSE POLARITY

NOTE: YOU MAY NEED TO RE-SECURE THE GPS ANTENNA PATCH AGAIN WITH DOUBLE SIDED TAPE WHEN NECESSARY AS THE MODULE CAME IN WITH A TEMPORARY TAPE

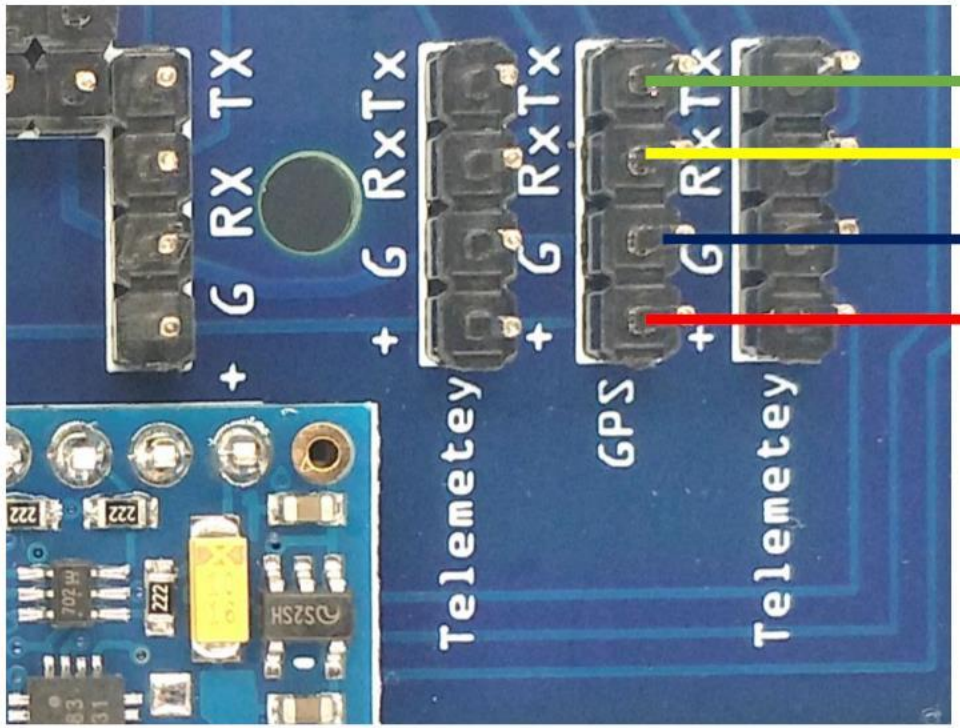
NOTE: WE PRESET THE GPS FOR YOUR CONVENIENCE TO THE PROPER SETUP BUT SHOULD YOU WISH TO CHANGE THE SETTING ON YOUR DIGRESSION

GPS NEO6 (NMEA)

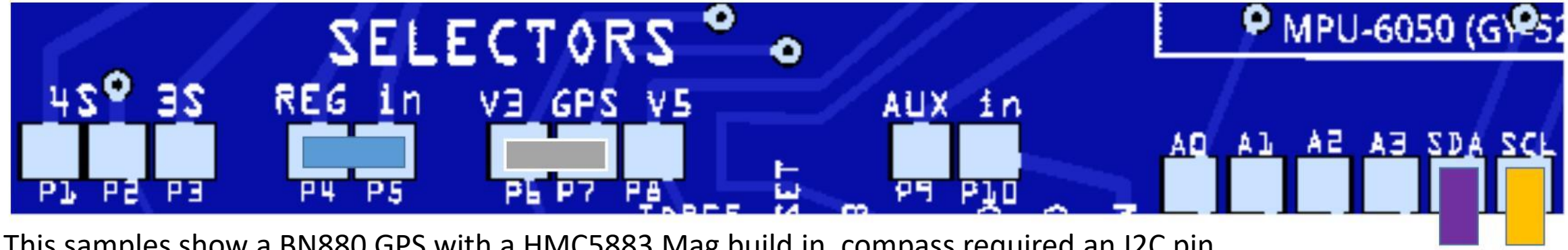


GPS BEITIAN (UBLOX)

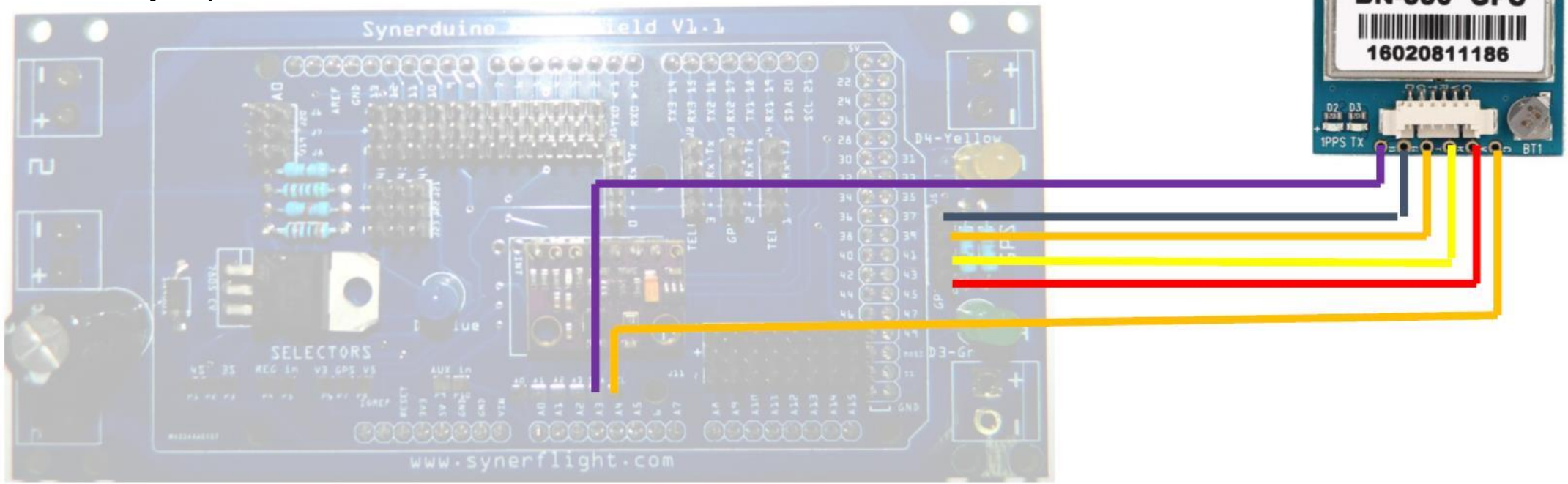
PIN	PIN Name	I/O	Description
1	GND	G	Ground
2	TX	O	Serial Data Output.
3	RX	I	Serial Data Input.
4	VCC	I	DC 3.0V - 5.5V supply input,Typical: 5.0V



External Sensors



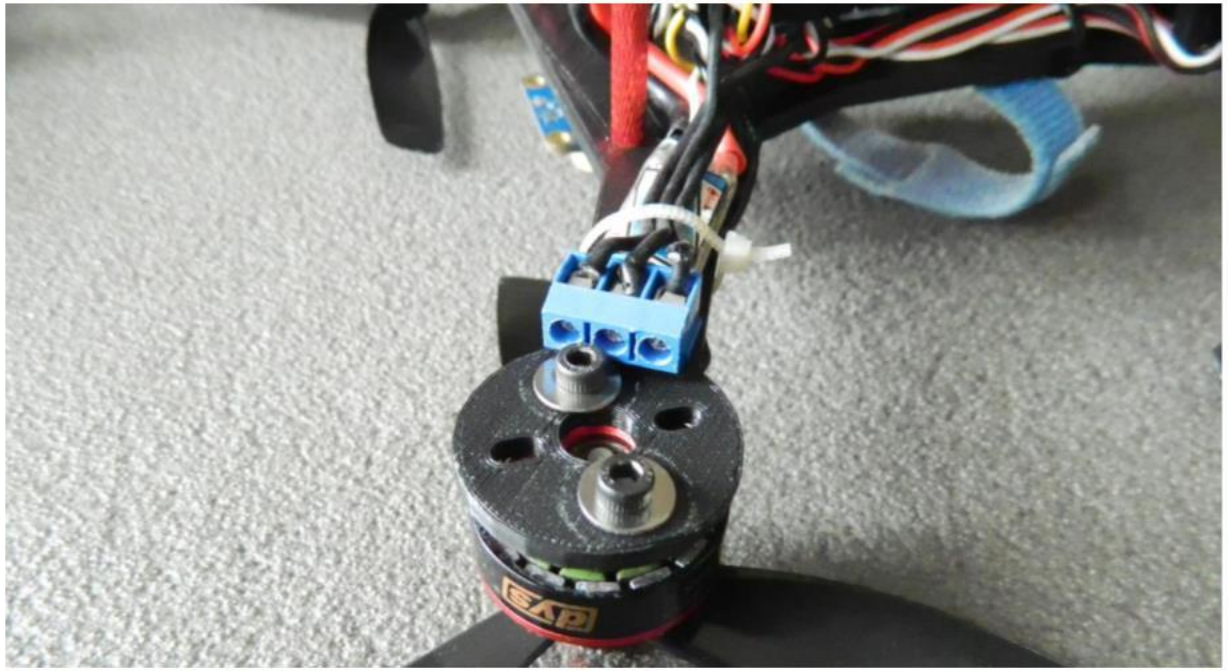
This samples show a BN880 GPS with a HMC5883 Mag build in compass required an I2C pin connection this works of all other I2C sensors (pls ensure the address doesn't conflict with the IMU as found in Sensors.cpp) Note: other than the GPS build in sensors might require 3V you may need to set jumper to 3V



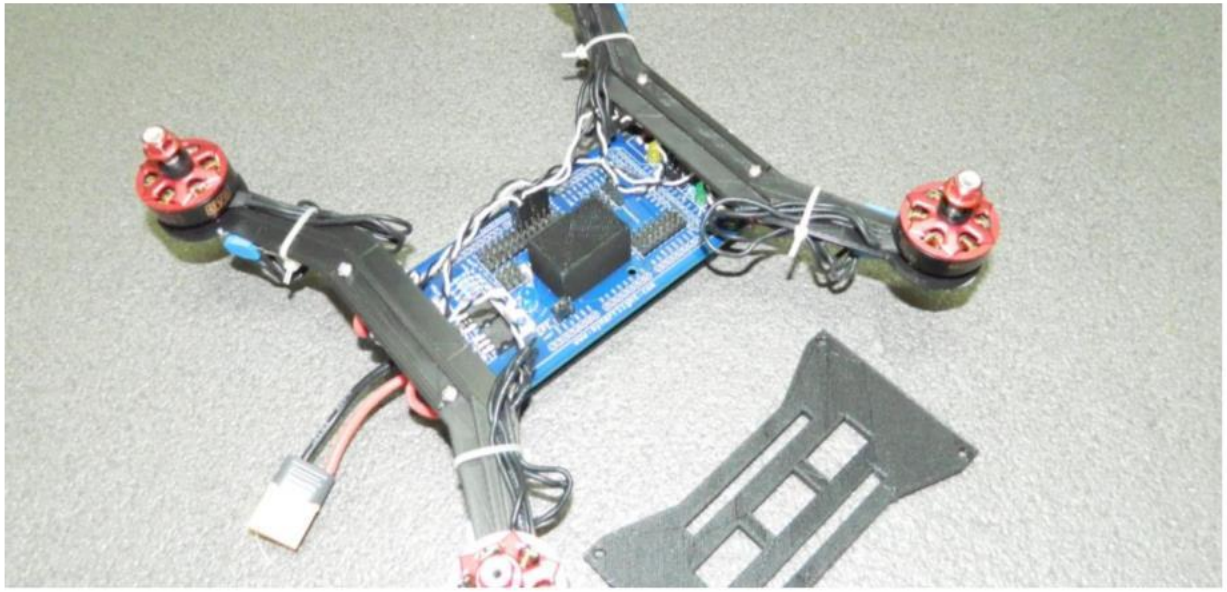
Arduino Board Preparation motor installation

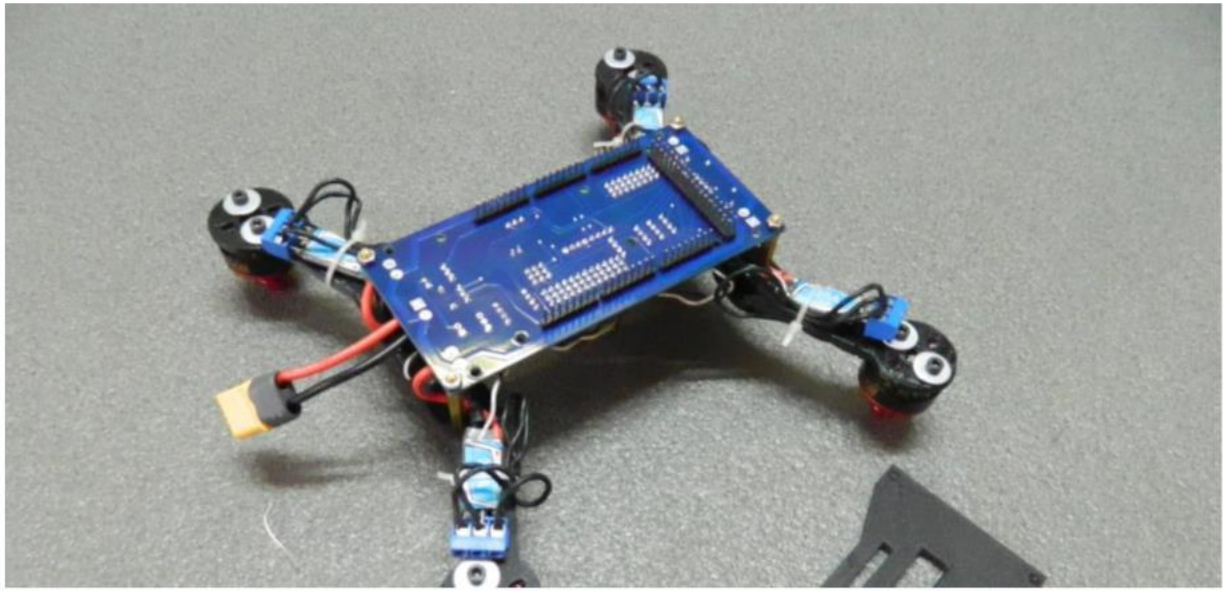
2560 MEGA

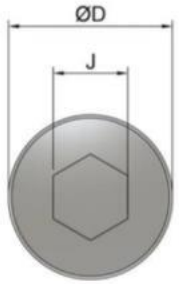
UNO 328

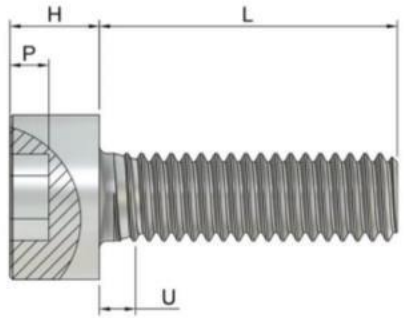


Use the 6mm M3 Bolt provided (short one) and install with washers



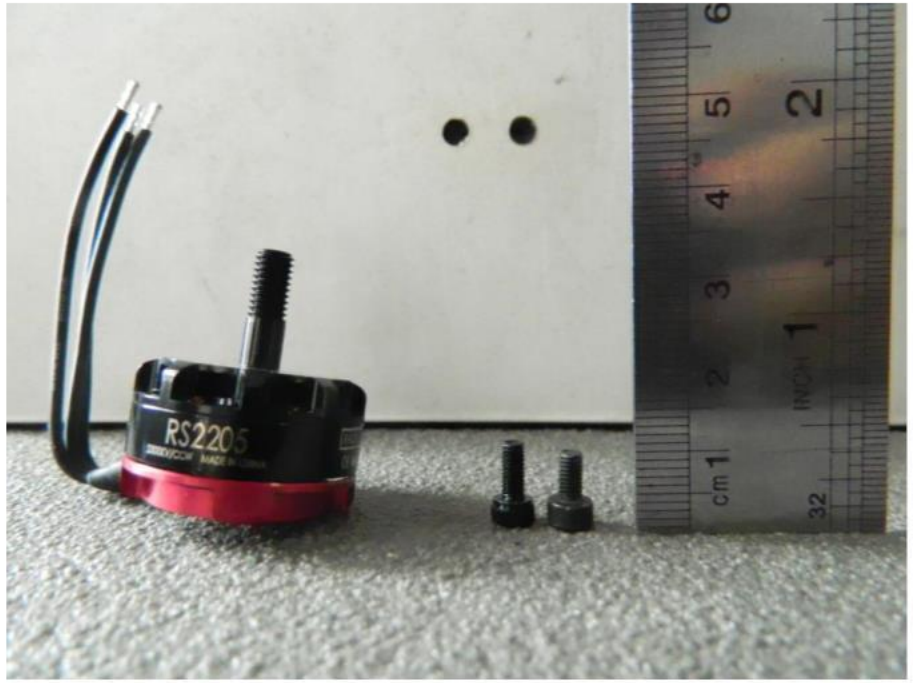


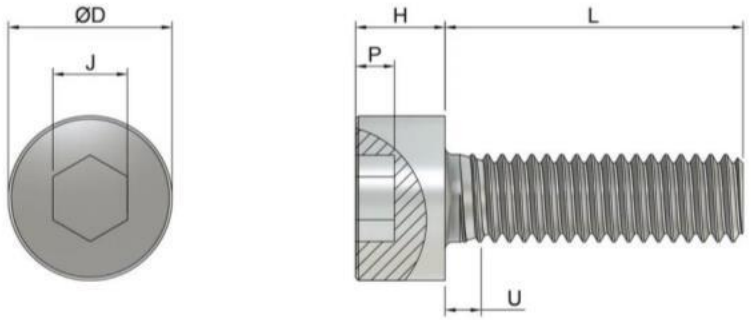




Thread Lock Purple or PVA Glue

Vibrations can cause bolts/Screws to come loose especially when bolts are on smooth plastic surface
Low strength thread lock or PVA glue are suitable in holding the bolts in place yet temporary enough that can still be remove by hand tools . Secure motors , PCB board and frame by applying a few drops enough to coat the thread of the screw and slowly tighten it in





Note : this frame is design to use M3 bolts with thread length of 6mm with washer as fittings included in the set (use the shortest bolt that came with the motor set)

Ensure that the bolt thread does not touch the inner wire coils of the motor

Use small amount of PVA white Glue to Thread Lock the bolt in place Preventing it from going loose

How to Choose Drone Motors

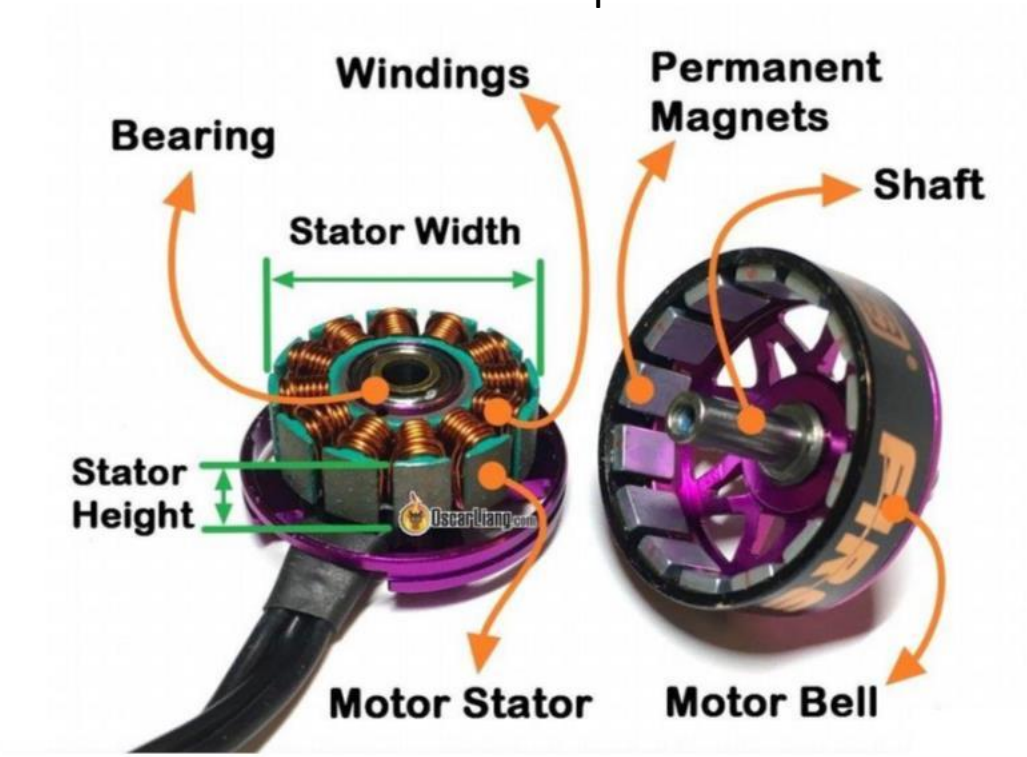
Before choosing motors, you should at least have a rough idea of the size and weight of the drone you want to build. I will explain the process of determining motor size based on drone size, Here is a rough chart of what to expect for a given frame size

Frame Size	Prop Size	Motor Size	KV
150mm or smaller	3″ or smaller	1105 -1306 or smaller	3000KV and higher
180mm	4″	1806, 2204	2600KV – 3000KV
210mm	5″	2205-2208, 2305-2306	2300KV-2600KV
250mm	6″	2206-2208, 2306	2000KV-2300KV
350mm	7″	2506-2508	1200KV-1600KV
450mm	8″, 9″, 10″ or larger	26XX and larger	1200KV and lower

Motor Size Explained

The size of brushless motors in RC is normally indicated by a 4-digit number – AABB. “AA” is the **stator width** (or stator diameter) while “BB” is the **stator height**, both are measured in millimeter.

KV is RPM Per Volt in this sample its 2300kv



Brand Name Rotation Arrow (CW)

RPM/Volt Rotation

2300KV/CW BR2205 Motor Model Number Stator Width Stator Hight

Motors Current and Voltage

It’s important to understand that voltage has a large impact on your motor and propeller choice. Your motor will try to spin faster with a higher voltage, thus draw a higher current. Ensure you are aware of how much thrust your motors produce and how much current they will draw.

Quadcopter flight time =(Battery Capacity * Battery Discharge /Average Amp Draw)*60

1300Mah 11.1V Lipo
80% Discharge

Sample base on motor datasheet max rating :
Round off 5A per motor
(1300mah/1000mah)*80% / 20Amps*60 Min

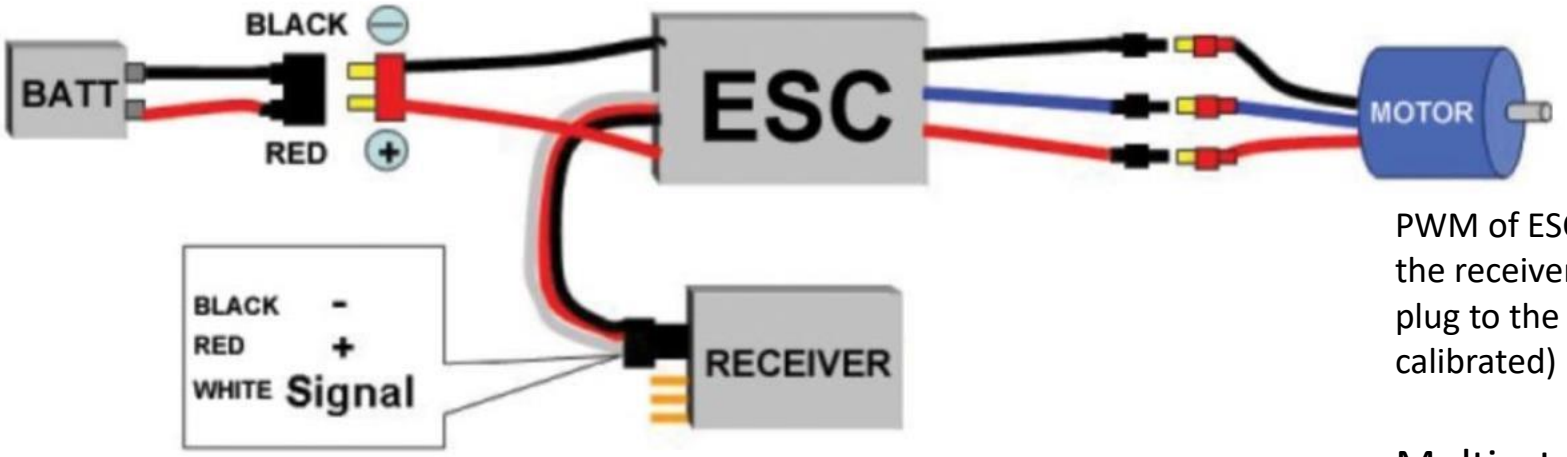
= 3.12min (Note motor chars specify Max amp)

MOTOR TECHNICAL DATA:

	NO LOAD		ON LOAD				LOAD TYPE
VOLTAGE	CURRENT	SPEED	CURRENT	Pull	Power	EEP	Battery/prop
V	A	rpm	A	g	W	%	
12	1.2	30000	5.3	200	63.6	3.1	LiPox3/4045x3
			13.3	400	159.6	2.5	
			22.7	580	272.4	2.1	
			5.7	250	68.4	3.7	LiPox3/5045x4
			14.2	500	170.4	2.9	
			31.6	800	379.2	2.1	
			5.1	250	61.2	4.1	LiPox3/5040x3
			12.2	500	146.4	3.4	
			21.6	720	259.2	2.8	
			5.2	250	62.4	4.0	LiPox3/5040x3
			12.9	500	154.8	3.2	
			23.8	750	285.6	2.6	
			4.9	250	58.8	4.3	LiPox3/5045
			11.9	500	142.8	3.5	
			19.6	700	235.2	3.0	

Note : most motor technical Data would only present maximum current draw rating of the motor and given your not always on full throttle all the time in flying so we can give the current draw a little less than what is shown here

Electronic Speed Controller CALIBRATION



Propellers are removed during this process

PWM of ESC is directly hook up to the receiver Throttle pin . Ensure the receiver is getting power thru the Aux PWM pin which remains plug to the synerduino board (process is repeated till all ESCs are calibrated)

ESC calibration will vary based on what brand of ESC you are using, so always refer to the documentation for the brand of ESC you are using for specific information (such as tones). “All at once” calibration works well for most ESCs, so it is good idea to attempt it first and if that fails try the “Manual ESC-by-ESC” method.

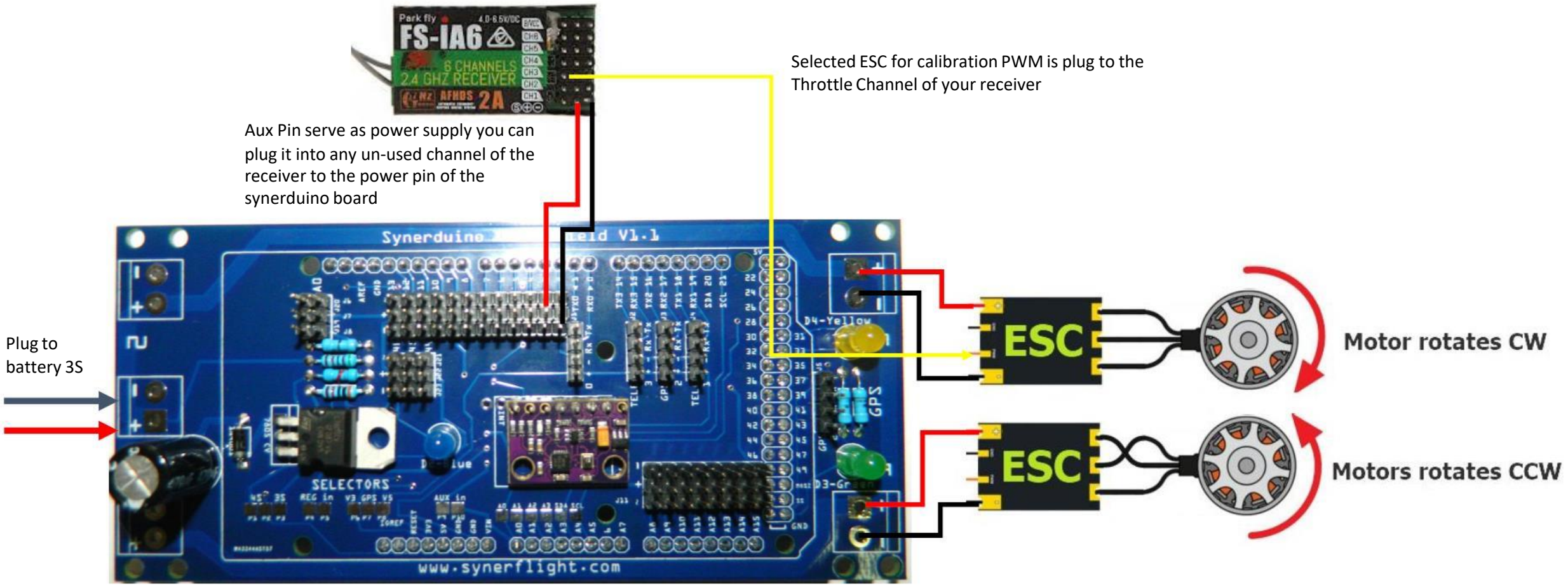
Multirotors must have all ESCs calibrated similarly to ensure reliable operations

Motor must be plug in at this point w/o the propeller. As it will serve several purpose

- An Speaker to listen to calibration tone of the ESC
- Identify motor rotation should it needs to be corrected
- Test full speed range

If your ESC happens to be an OPTO. The Synerduino board can provide as power supply for both RC Receiver and ESCs when soldered in . Get the PWM Pin of the ESC you want to calibrate and plug it into the Throttle Channel of your Receiver

Individual ESC calibration method , for individual ESC at a time



Electronic Speed Controller CALIBRATION

1.

Turn transmitter on.
Set throttle to maximum.

You have the option to put your trim center or lowest trim setting using the trim buttons

2.

Connect battery to power module.
/ Synerduino Shield board

3.

Wait for your ESCs/Motor to emit the musical tone, the regular number of beeps indicating your battery's cell count (i.e. 3 for 3S, 4 for 4S) and then an additional two beeps to indicate that the maximum throttle has been captured.

4.

Set throttle to minimum.

Some ESC will give a tone to identify the calibration mode is exited

5.

6.

Disconnect battery.

7.

Connect battery to power module.

Reconnect the battery with the throttle to minimu to initiated booth up then test the throttle. This Process is repeated till all ESC are calibrated

Synerduino Multiwii ESC calibration method , for All ESC at Once

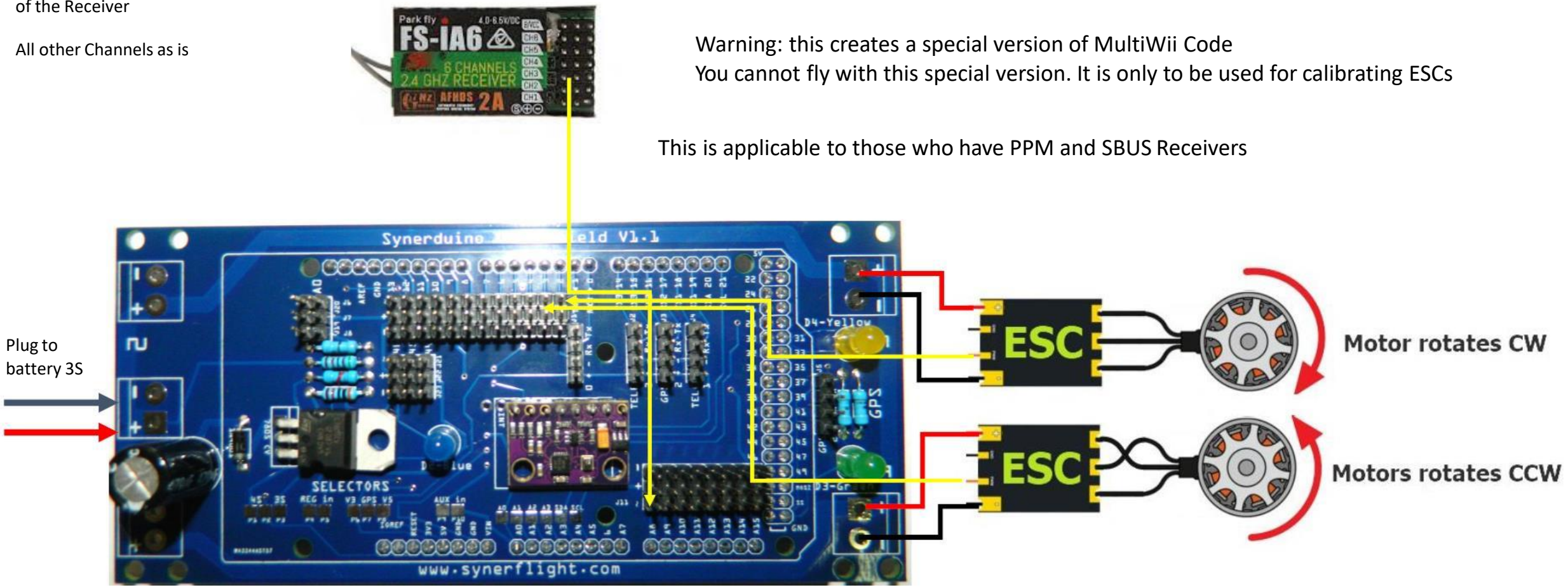
Throttle Pin is Connected to the Throttle Channel of the Receiver

All other Channels as is

to calibrate all ESCs connected to MWii at the same time (useful to avoid unplugging/re-plugging each ESC)

Warning: this creates a special version of MultiWii Code
You cannot fly with this special version. It is only to be used for calibrating ESCs

This is applicable to those who have PPM and SBUS Receivers



MULTIWII.INO

Multiwii Sketch

source code for Synerduino KWAD Shield: compile this using the arduino IDE then flash it to your Arduino board

(Arduino 1.8.16) SynerduinoKwad4-GY801-GY91-1.8.16 Download

()Arduino 1.8.16 – 2.0.0 SynerduinoGY91-1.8.6-2.0.0 Download

(For UNO boards)Synerduino-Uno-GY91-801 Download

http://synerflight.com/kwad-documentation/3-software/

The Calibration sketch is accessible on these firmware config.h Tab

This houses the bulk of that the Arduino Drones codes

CONFIG.H

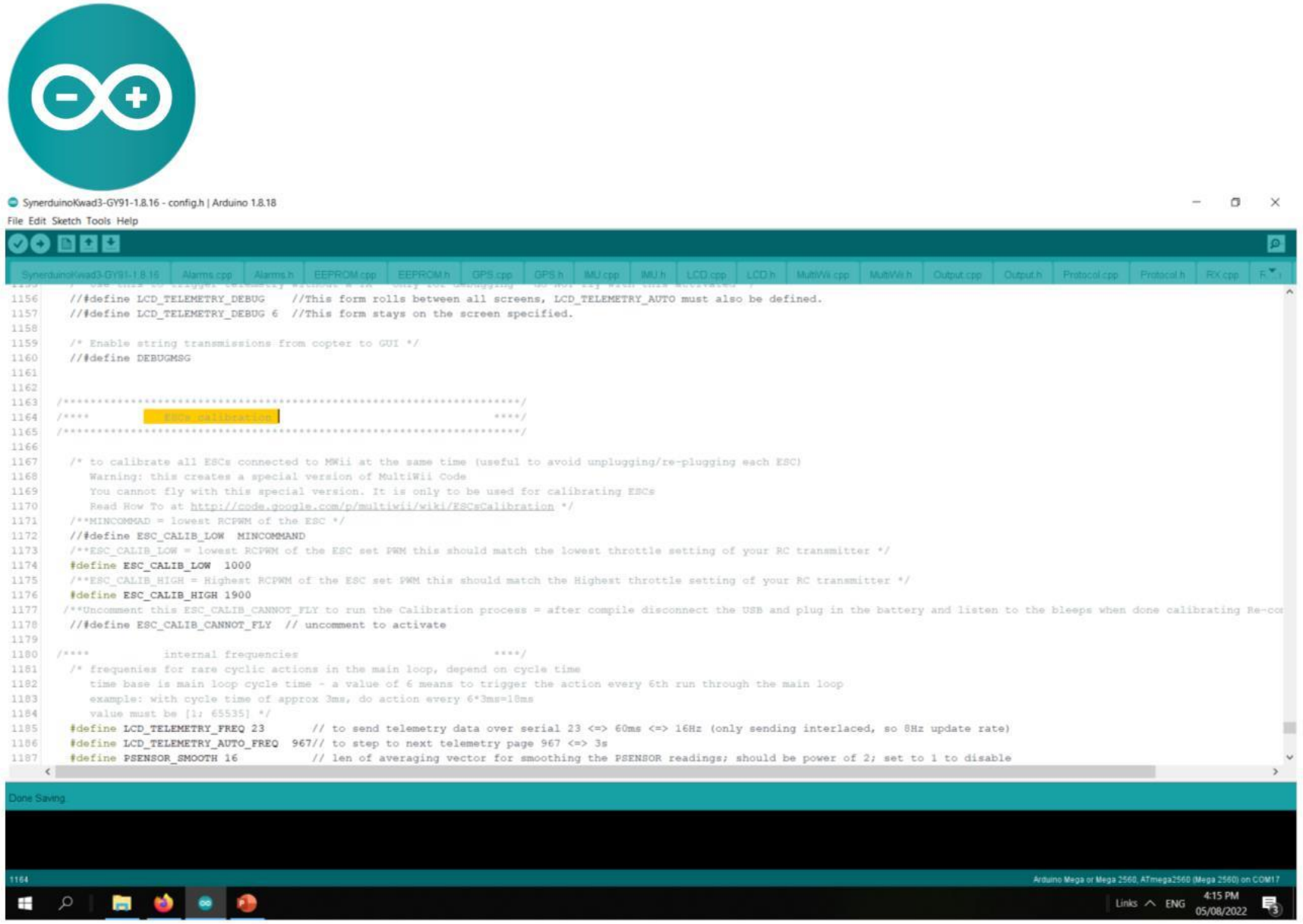
In Config.h Tab

Look for ESC Calibration Menu

Uncomment Define ESC_Calib _Cannot _Fly

This disable all other functions of the drone other than the Throttle input from the receiver as Pass thru.

Remember to Re-comment this menu after Calibration Process is complete



Electronic Speed Controller CALIBRATION

1.remove props or tie copter down.

2.setup all options in config.h to whatever suits your copter

3.activate the *define ESC_CALIB_CANNOT_FLY*, possibly set high and low values for ESC calibration, if you know what you are doing

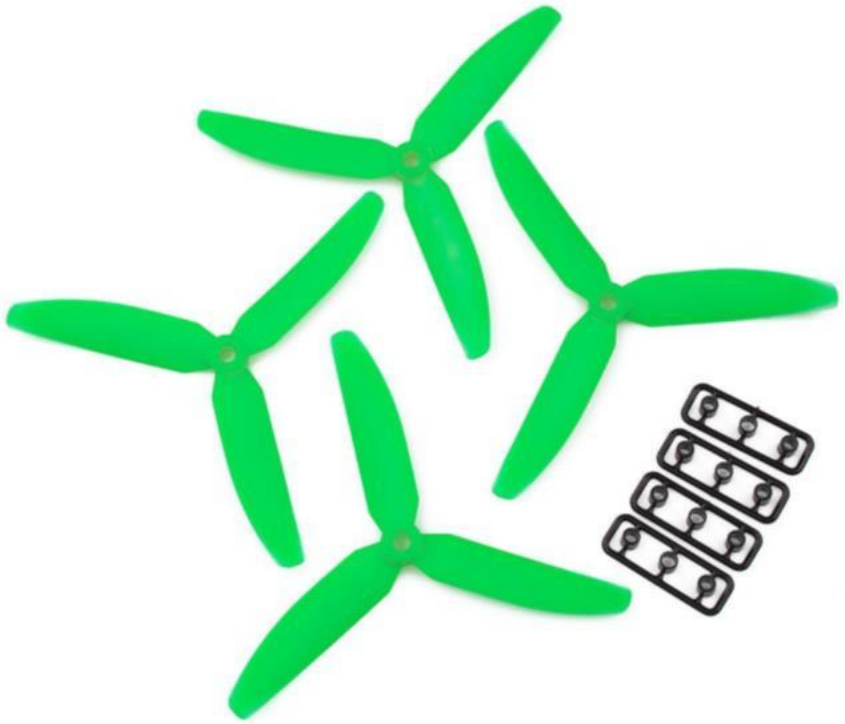
4.compile, upload, run --- cannot fly and will use Buzzer/LEDs to indicate finished calibration (after approx 10 seconds)

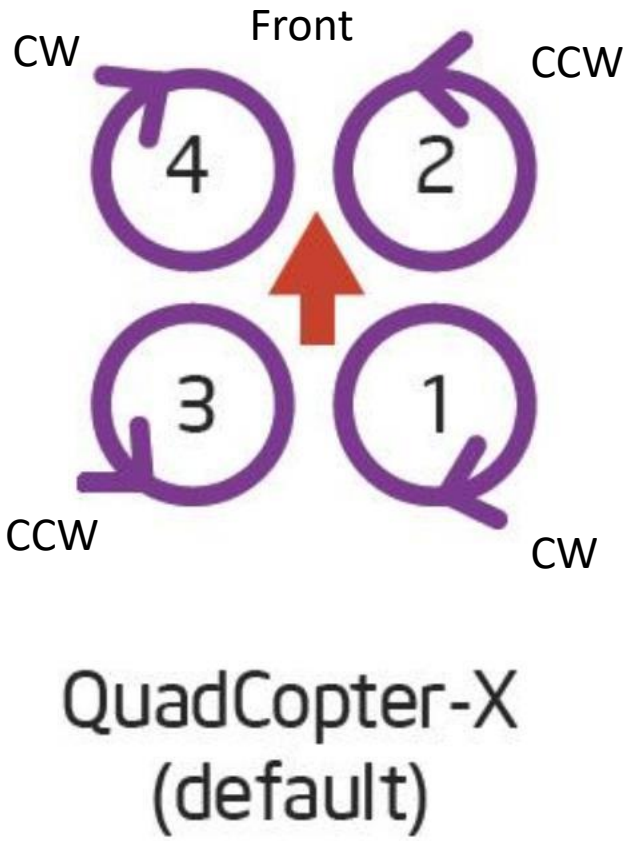
5.comment the define again, compile, upload

6.test carefully with your ESCs calibrated, fly and have fun

Propellers

Sometimes even the most prominent features can be overlooked yet this is a critical element on what makes a multirotor fly





Note: ensure the props are well balance
With no blade damage if you want to ensure a good Stability in GPS and Altitude hold modes.

Vibration in the frame can cause the sensors to register noise making flights unstable

With the Synerduino 250mm we recommend the 5x45x3 (5045 3) or 5x40x3 (5040 3) Prop type to match with the 2300kv motor the Props have to be stiff too to accommodate the extra load of the quad

Size

Propeller size directly influences thrust and responsiveness.

Smaller propellers are lighter and work against less air. This makes them more responsive; they're easy to stop and speed up.

They require high KV motors to operate due to their RPM

Larger propellers consume more power and create a bigger pressure difference. Put simply, they generate more thrust.

They require low KV motors as they need more torque to move the large blades

Pitch

The pitch is a crucial factor to consider as it's an important par of propeller specifications.

It's defined as the travel distance of the propeller per revolution and is generally measured in inches.

Propellers with a lower pitch spin faster which facilitates higher acceleration. At the same time, less current is drawn.

A higher pitch, on the other hand, means more thrust will be generated. This results in higher with the drawback of more power consumption and additional torque required to drive it. Also high pitch can result on prop stall fluttering which can be dangerous at certain RPM use with caution

Again, the ideal pitch length depends on your drone's application.

50453 or 5x45x3 Prop size in Inch Pitch Angle Number of Blades

Motor & Propeller Selection

For beginners its easy to mismatch these components that can result in unwanted behavior or poor performance of the drone we have a few recommendation what we were tested to work
this basic over view chart can show clearly shows the relation ship of Motor to Propeller ratio

Motor KV	Propeller Size and Pitch Hot Under Load Good	Drone Weight & Size Range Type Quad
850KV on 3s	10x45 (1045)	2kg-2.5kg 450mm-550mm
950KV on 3s	10x45 (1045)	2kg-2.5kg 450mm-550mm
920KV on 4s	10x45 (1045) 9x45 (9450) 8x45 (8045 or 8450)	2kg-2.5kg 450mm-550mm
920KV on 3s	10x45 (1045) 9x45 (9450) 8x45 (8045 or 8450)	2kg-2.5kg 450mm-550mm
1300kV on 3s	8x45 (8045 or 8450) 8x3.8 (8038 or 8380)	600g-1kg 330mm-450mm
1700kV on 3s	8x45 (8045 or 8450) 8x3.8 (8038 or 8380)	600g-1kg 330mm-450mm
2000Kv on 3s	6x45x3 (6045 3) 5x45x3 (5045 3)	300g-500g 300mm-330mm
2300kv on 3s	6x45x3 (6045 3) 5x45x3 (5045 3) 5x40x3 (5040 3)	300g-500g 250mm-330mm
2300Kv on 4s	5x45x3(5045 3) 5x45x3 (5040 3)	300g-500g 250mm-330mm
2400KV - 2500kv on 3s	5x45x3 (5045 3) 5x45x3 (5040 3) 4x45x3 (4045 3)	300g-500g 250mm-330mm

Note : Synerduino Drone tend to be slightly heavier than a similar size drone requiring stiffer Props to perform better We recommend 5045 3 or 5040 3

Once all the hardware been calibrated tune and setup you may zip tie the ESC and Tape insulate all the Solder Pads to prevent shorting

Application Needed

https://www.arduino.cc/en/main/software

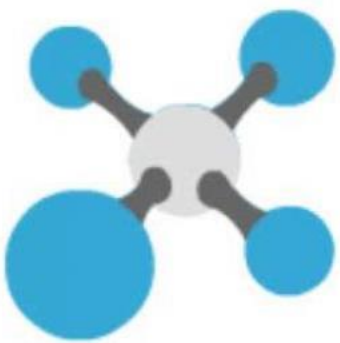
ARDUINO IDE

http://synerflight.com/flywiigui/

FLYWII GUI & Arduino Drone Multiwii firmware

https://www.u-blox.com/en/product/u-center

UBLOX Configuration Platform



https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu

XCTU Configuration Platform

MULTIWII.INO

Multiwii Sketch

source code for Synerduino KWAD Shield: compile this using the arduino IDE then flash it to your Arduino board

(Arduino 1.8.16) SynerduinoKwad4-GY801-GY91-1.8.16 Download

()Arduino 1.8.16 – 2.0.0 SynerduinoGY91-1.8.6-2.0.0 Download

(For UNO boards)Synerduino-Uno-GY91-801 Download

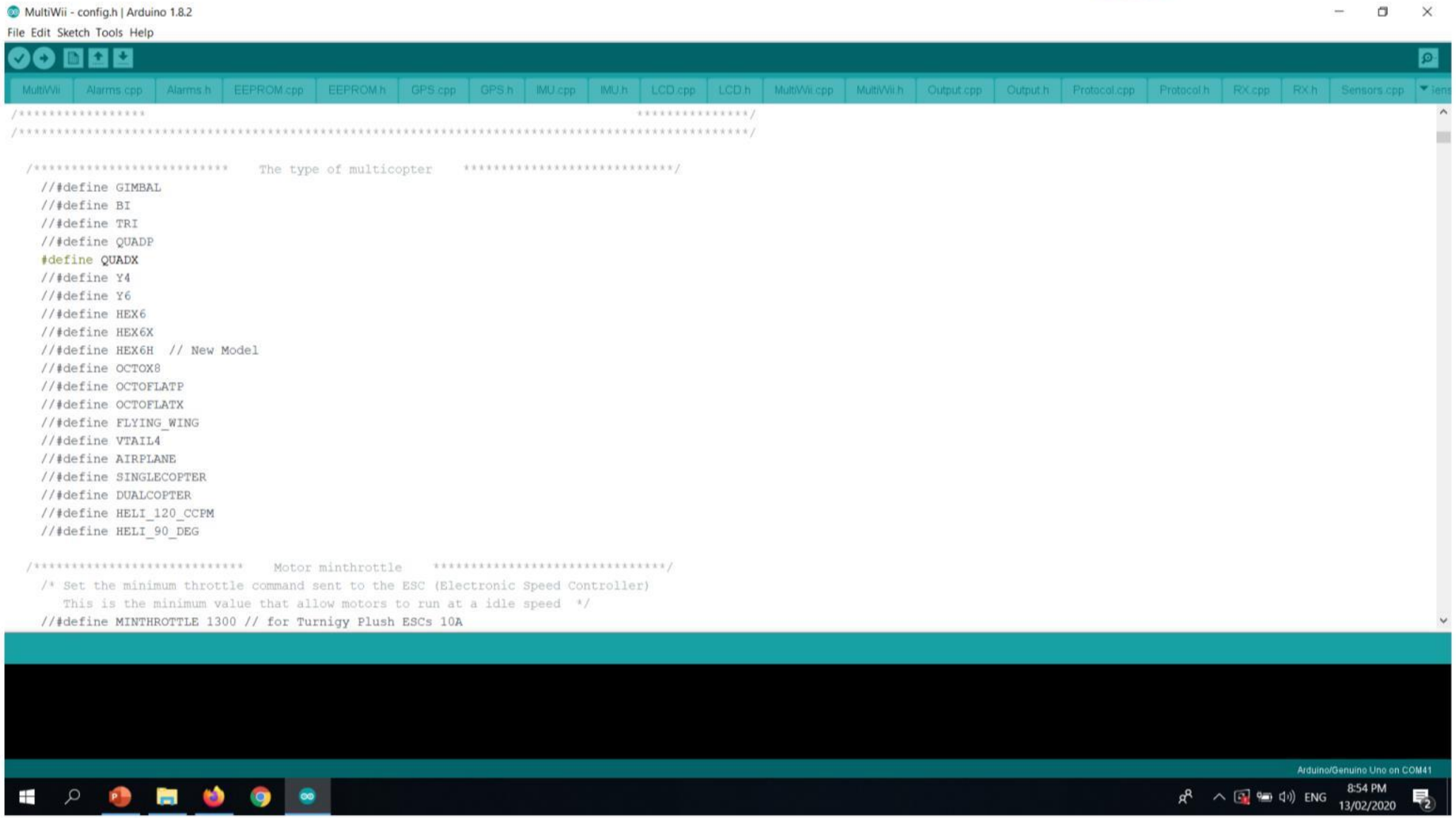
http://synerflight.com/kwad-documentation/3-software/

This is going to be the major bulk of this project down load and open the Multiwii.Ino as in Arduino IDE

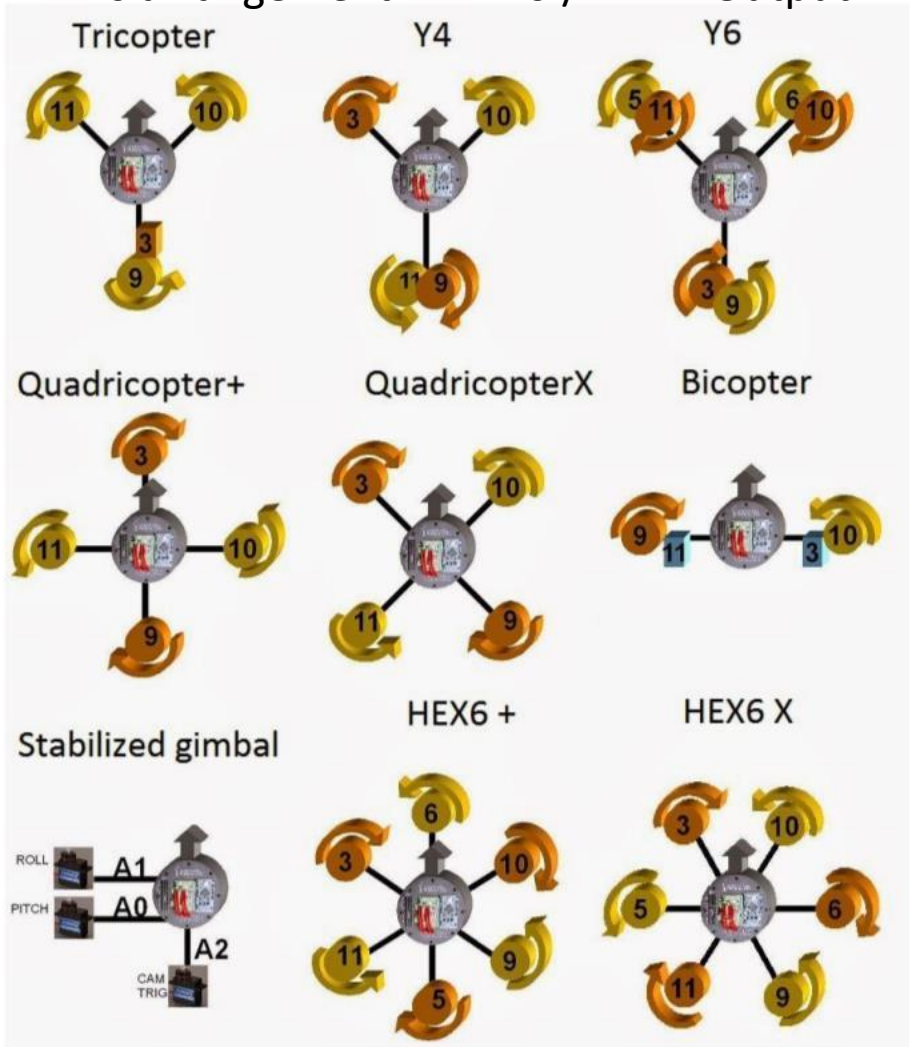
This houses the bulk of that the Arduino Drones codes

TYPE OF MULTICOPTER

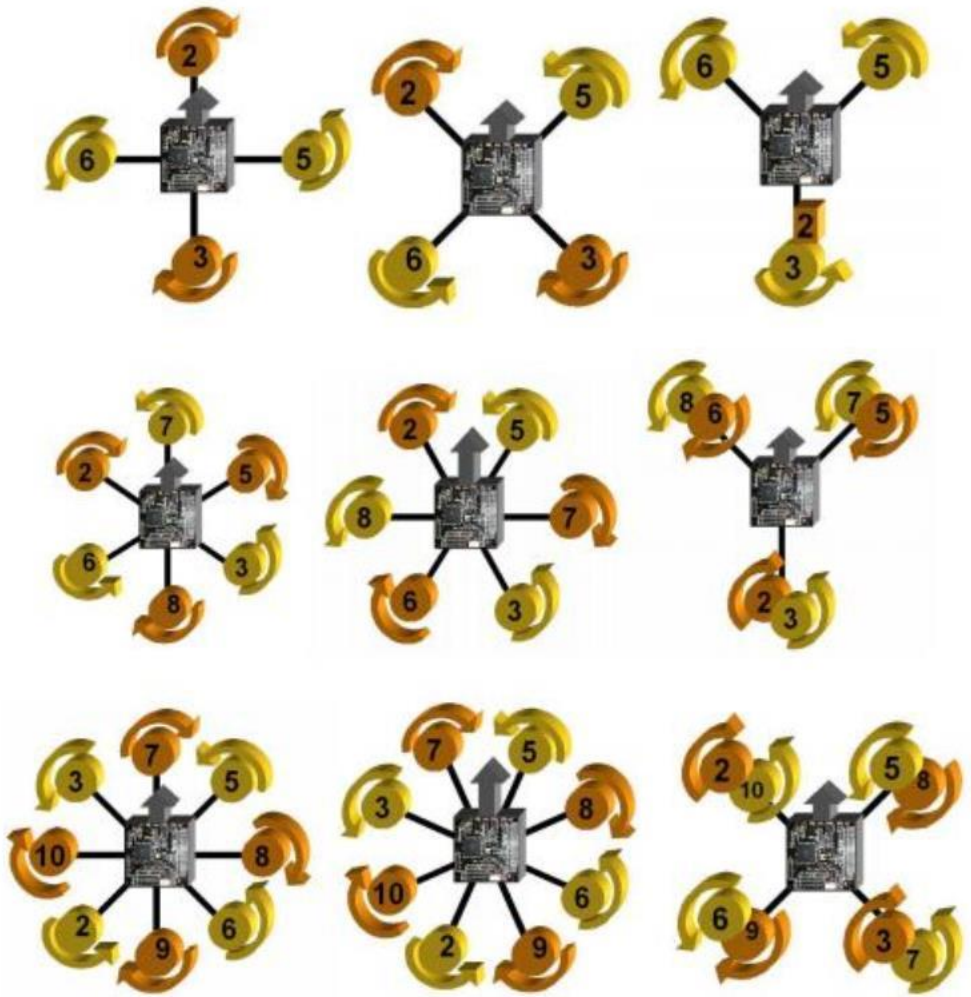
CONFIG.H



PWM Pins arrangement D2-D10 / PWM Output



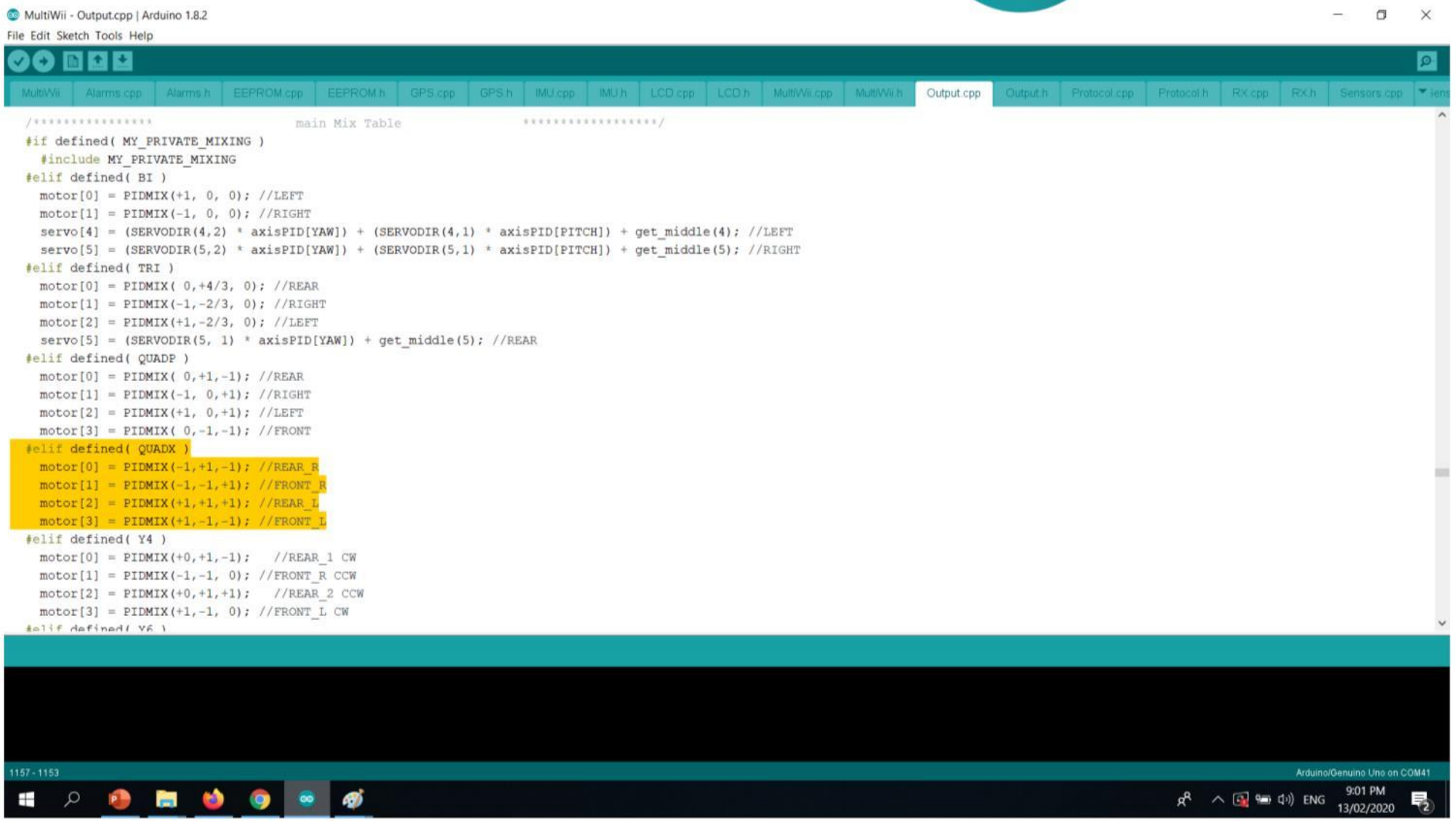
UNO 328



MEGA 2560

MIX TABLE

OUTPUT.CPP



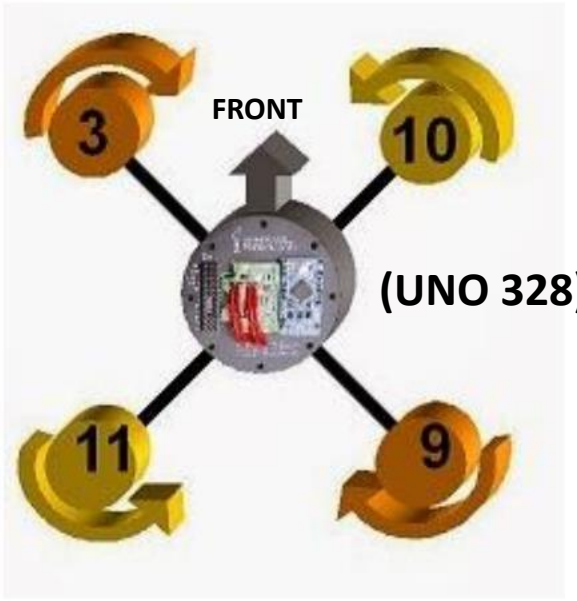
Mix Table shows the motors setup on input this is also helpful for those custom airframe designs require special mixing

OUTPUT.CPP

Roll Right Pitch Forward Yaw Right

Motor [3]

Motor [1]



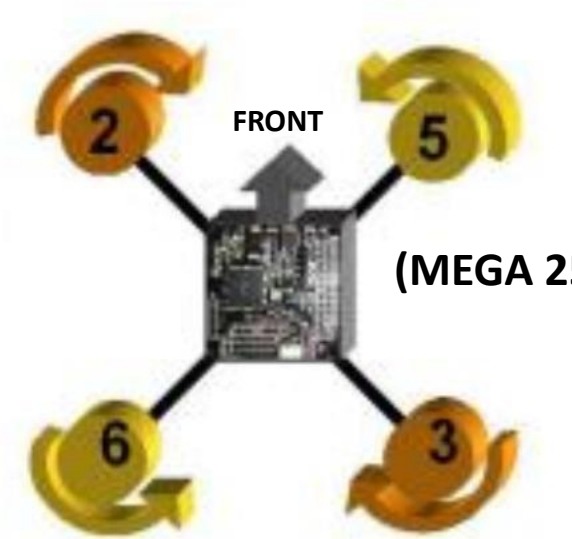
(UNO 328)

Motor [2]

Motor [0]

Motor [3]

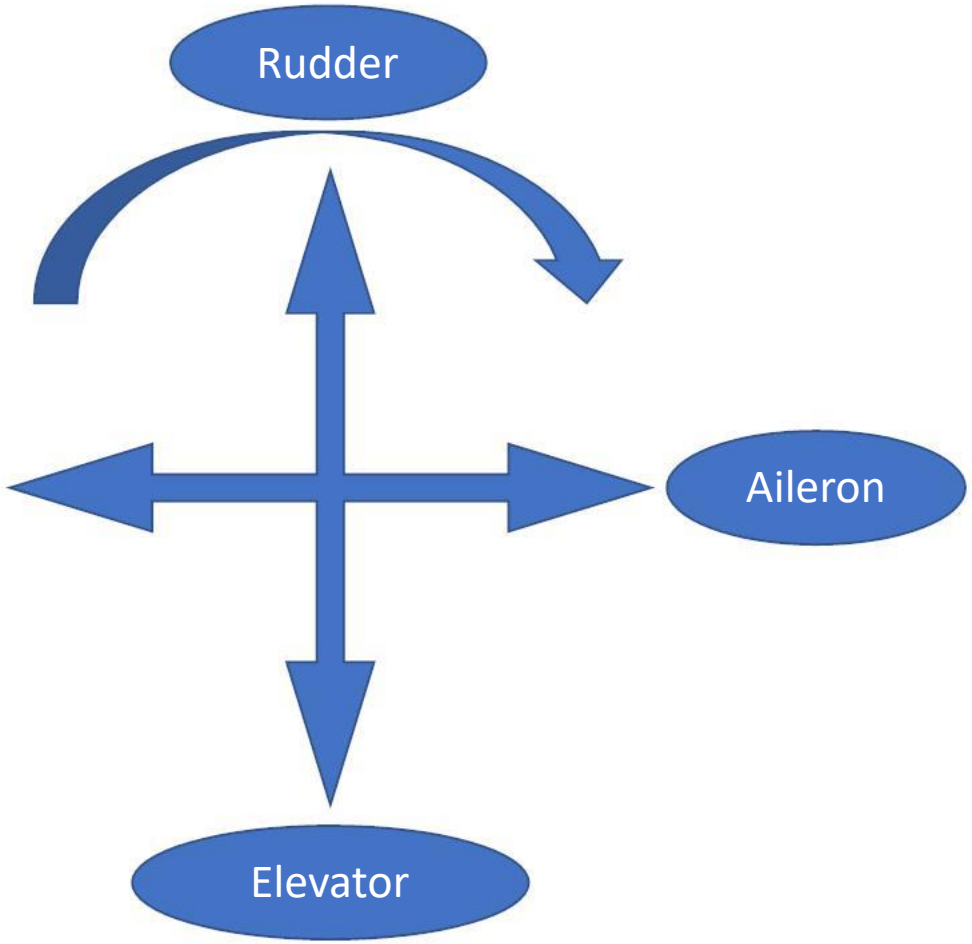
Motor [1]



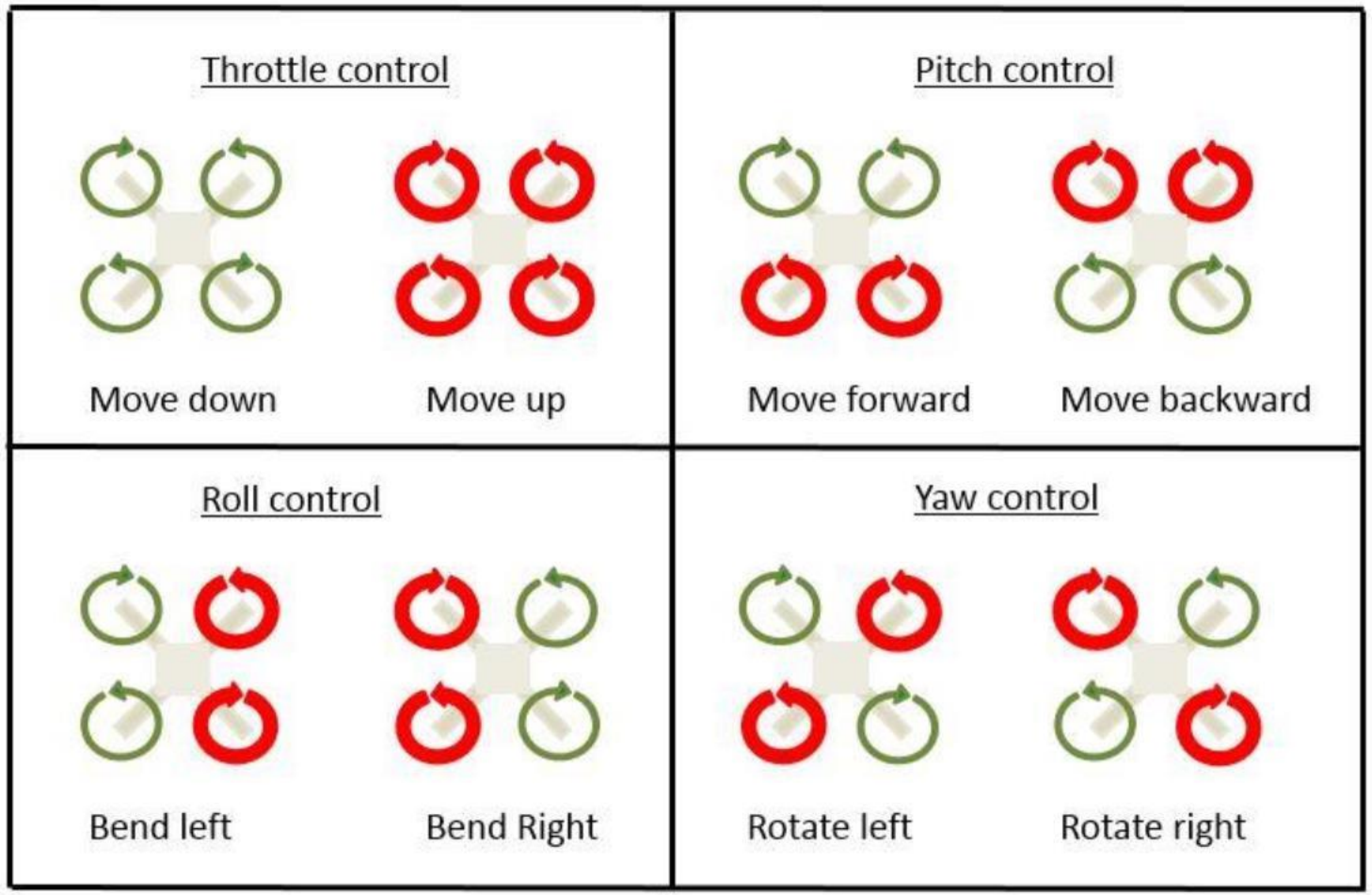
(MEGA 2560)

Motor [2]

Motor [0]

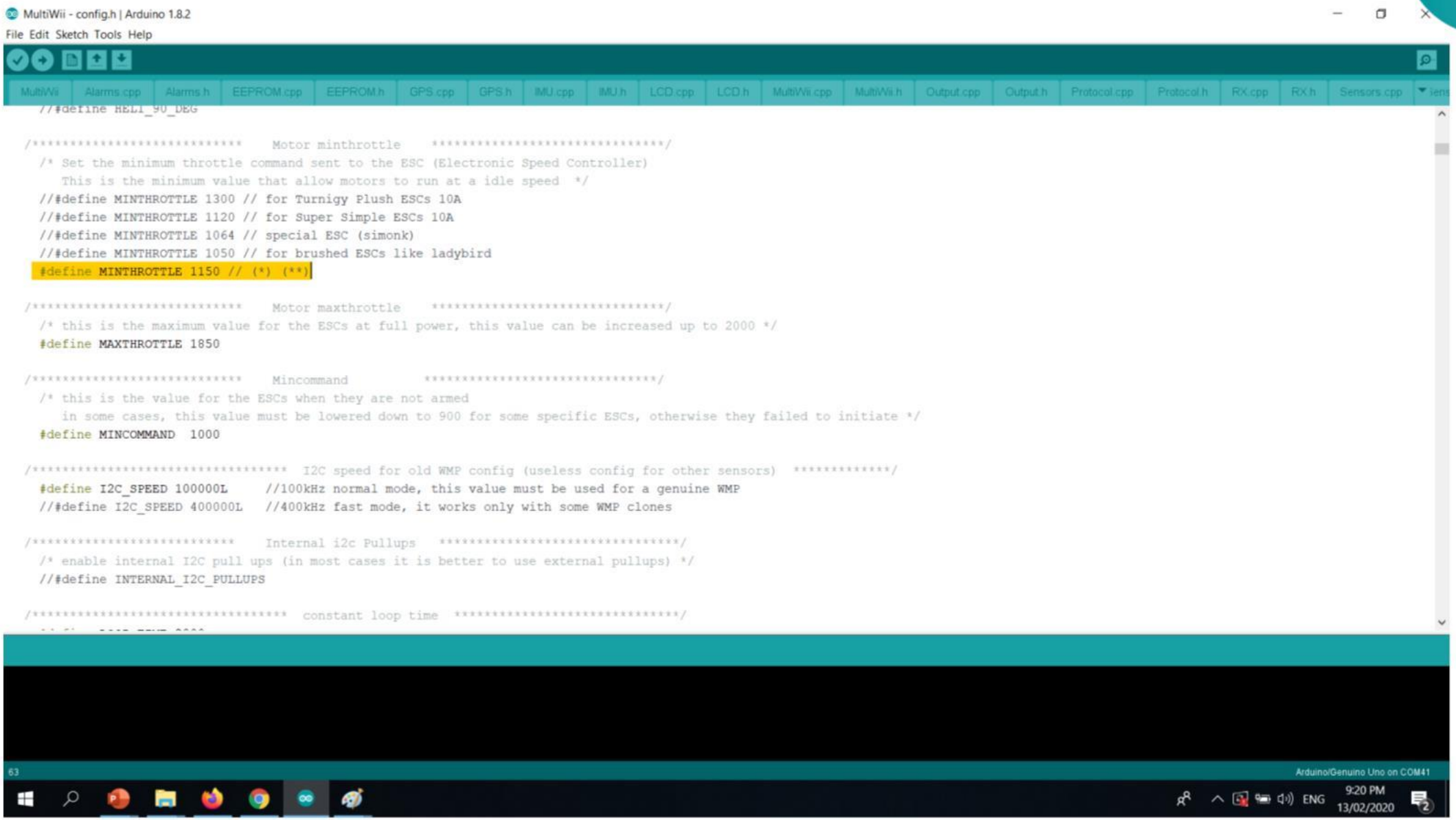


MOTOR MIX FOR QUAD



Motor Min Throttle

CONFIG.H



Motor Min Throttle

Electronic Speed Controller

Brushless Motor

1500

Idle Speed

MINTHROTTLE 1150

PWM Frequency

MAXTHROTTLE 1850

MINCOMMAND 800 - 1100

1900 – 2000

RECEIVER TYPES

PPM RECEIVER

SBUS RECEIVER

PWM RECEIVER

RECEIVER TYPES

CONFIG.H

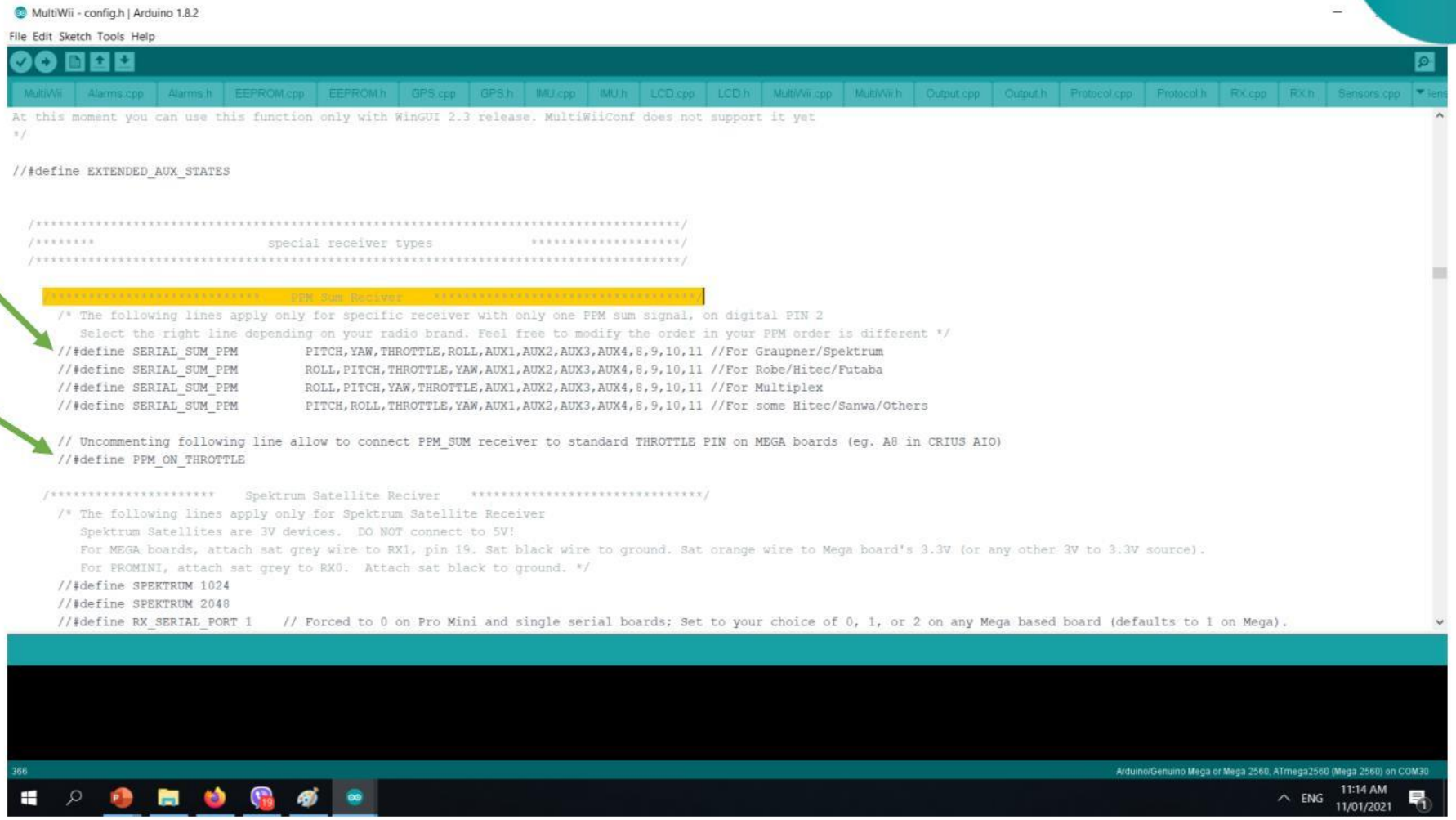
For PPM Receiver

Channel Mapping

Uncomment PPM on Throttle

Pin A8 for Mega
Pin D2 for Uno

You may need to uncomment and change the ordering of your channel depending on your Transmitter's model and specification



RECEIVER TYPES

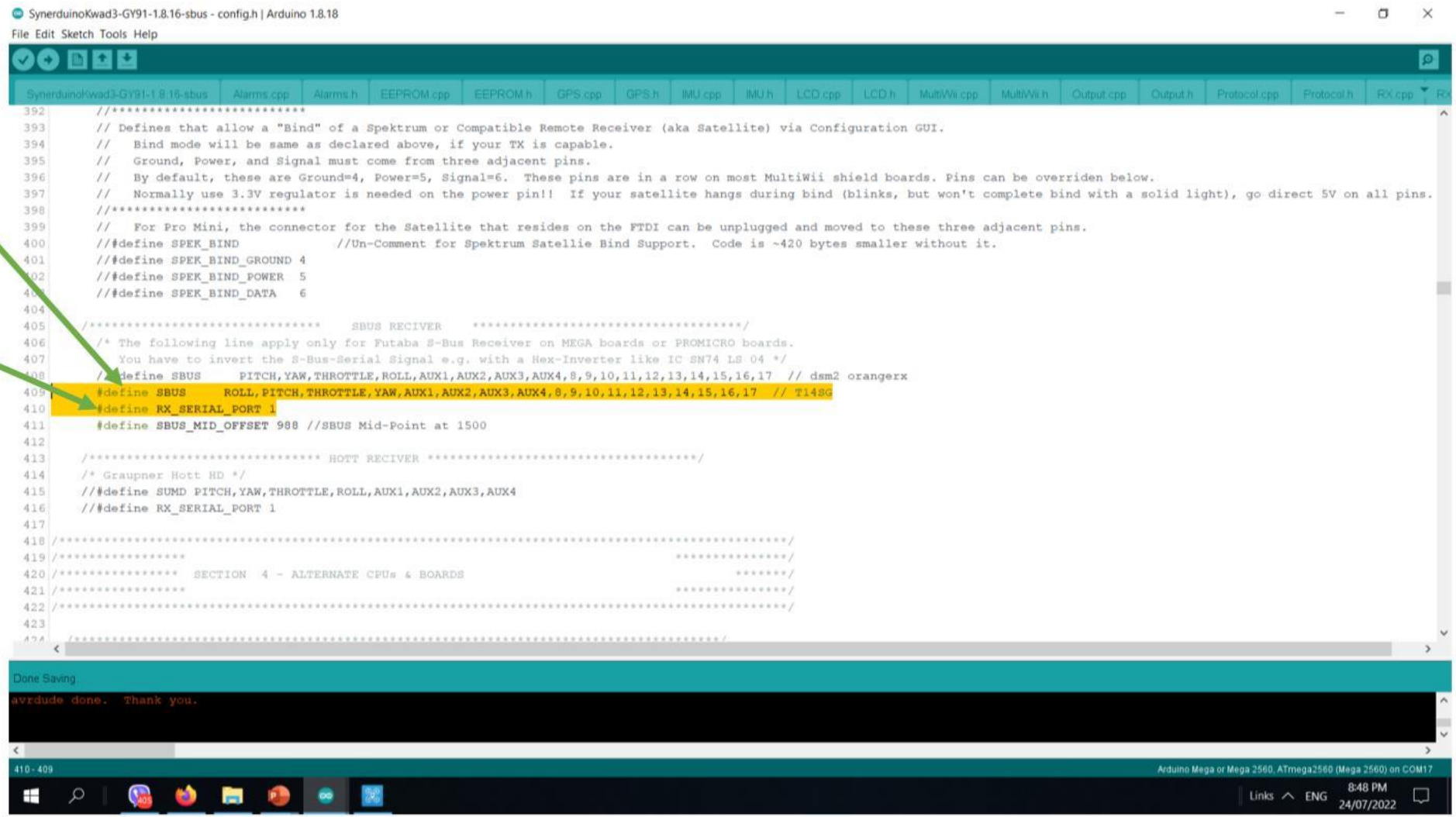
CONFIG.H

For SBUS Receiver

Channel Mapping

Uncomment SBUS on RX Serial Port 1 (Telemetry 1)

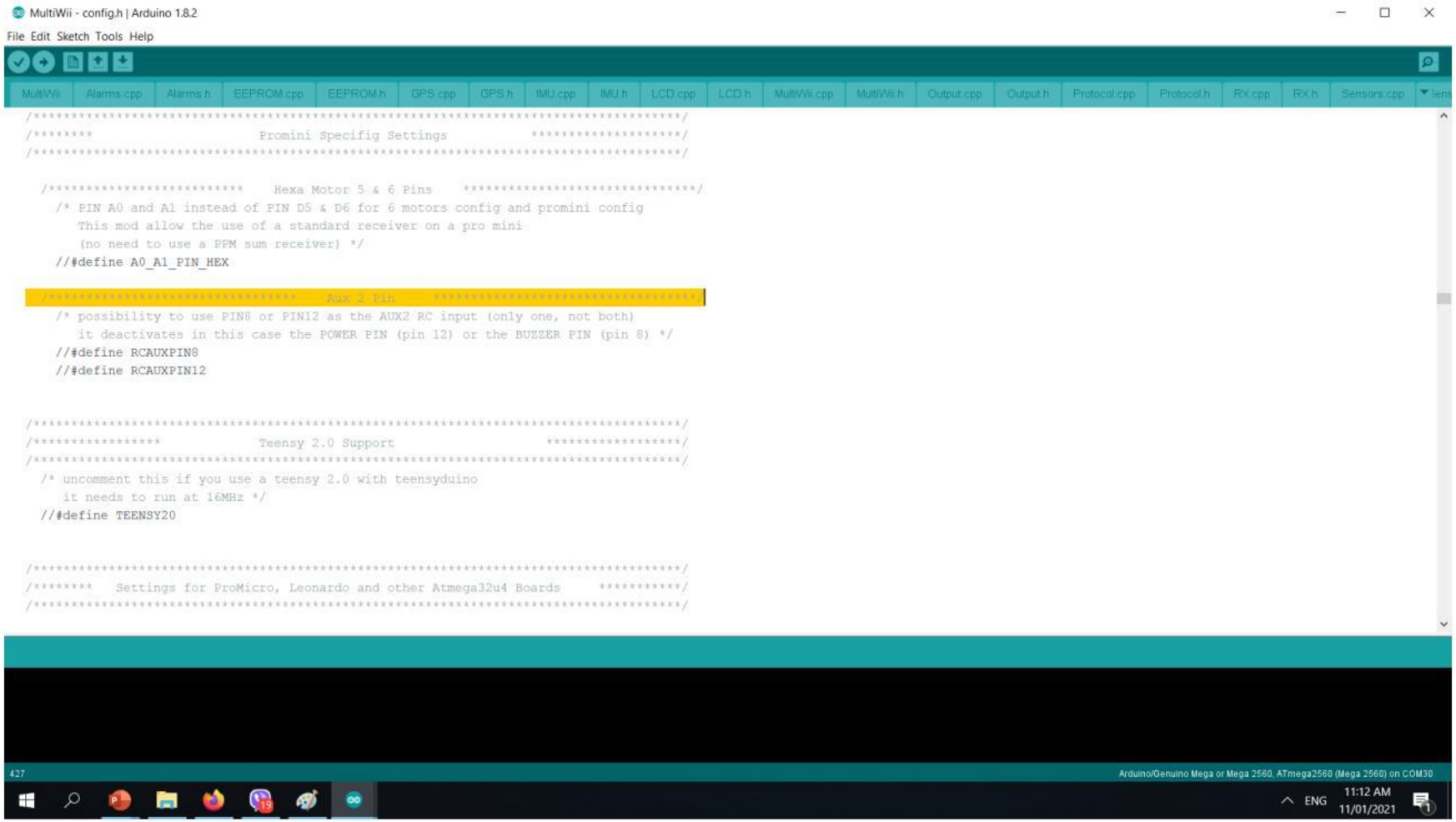
You may need to uncomment and change the ordering of your channel depending on your Transmitter's model and specification



AUX PIN

CONFIG.H

For UNO you need to define your Aux 2 Pin



INERTIAL MEASURING UNIT MEASURING UNIT

Pls see the Board Specs Data sheets for the installed IMUs onboard

Magnetometer

Barometer

This is the heart of every flight controller AKA the Main 4 ,

Gyro – stabilization on Roll Pitch Yaw Axis
Acc - Horizontal and Vertical stabilization XYZ
Baro – Altitude hold control
Mag – Heading and Compass

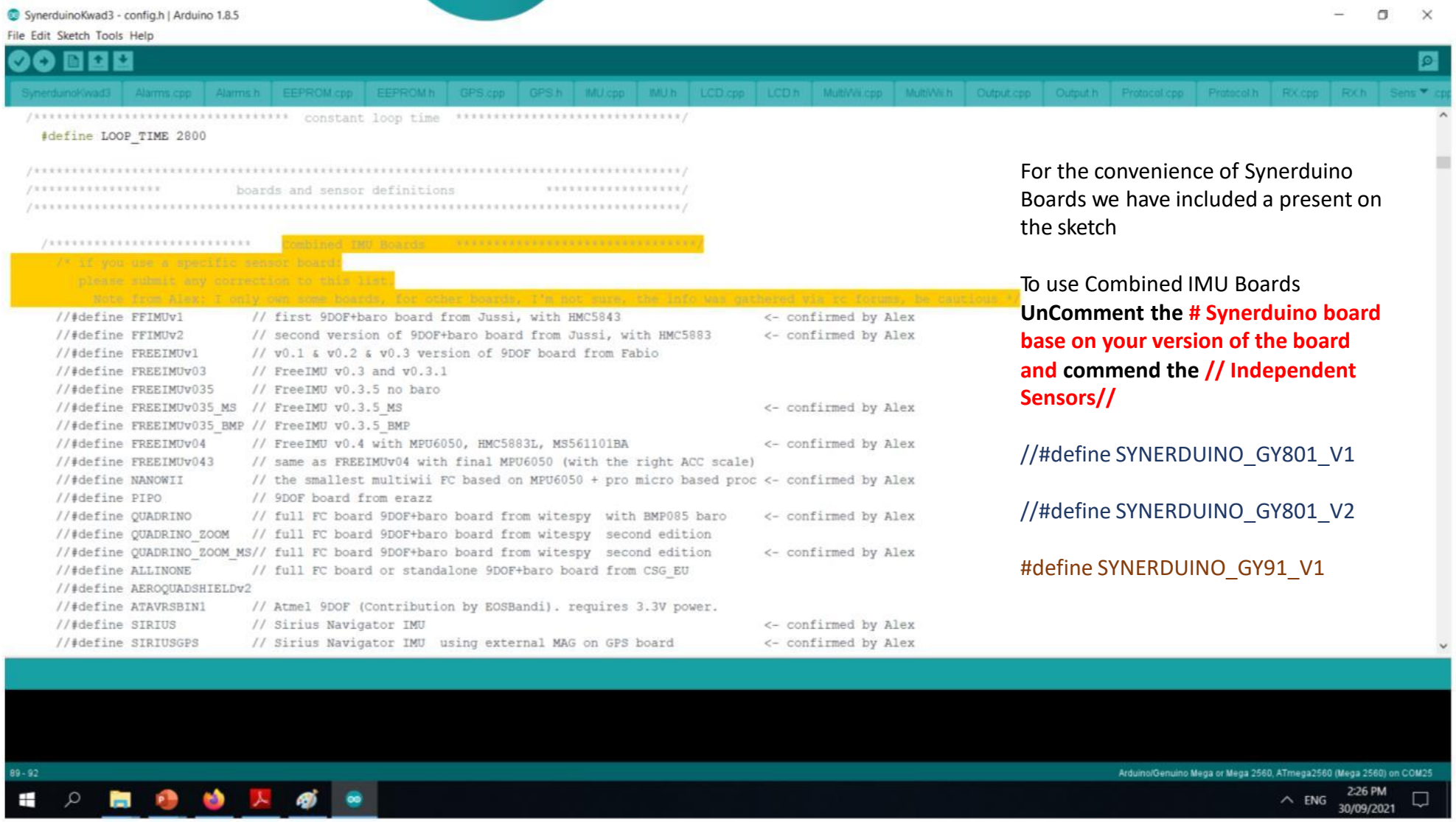
Accelerometer

Gyroscope

Each sensor has a corresponding address registry set by manufacturer

You can find it on sensors.ccp tab

CONFIG.H



For the convenience of Synerduino Boards we have included a present on the sketch

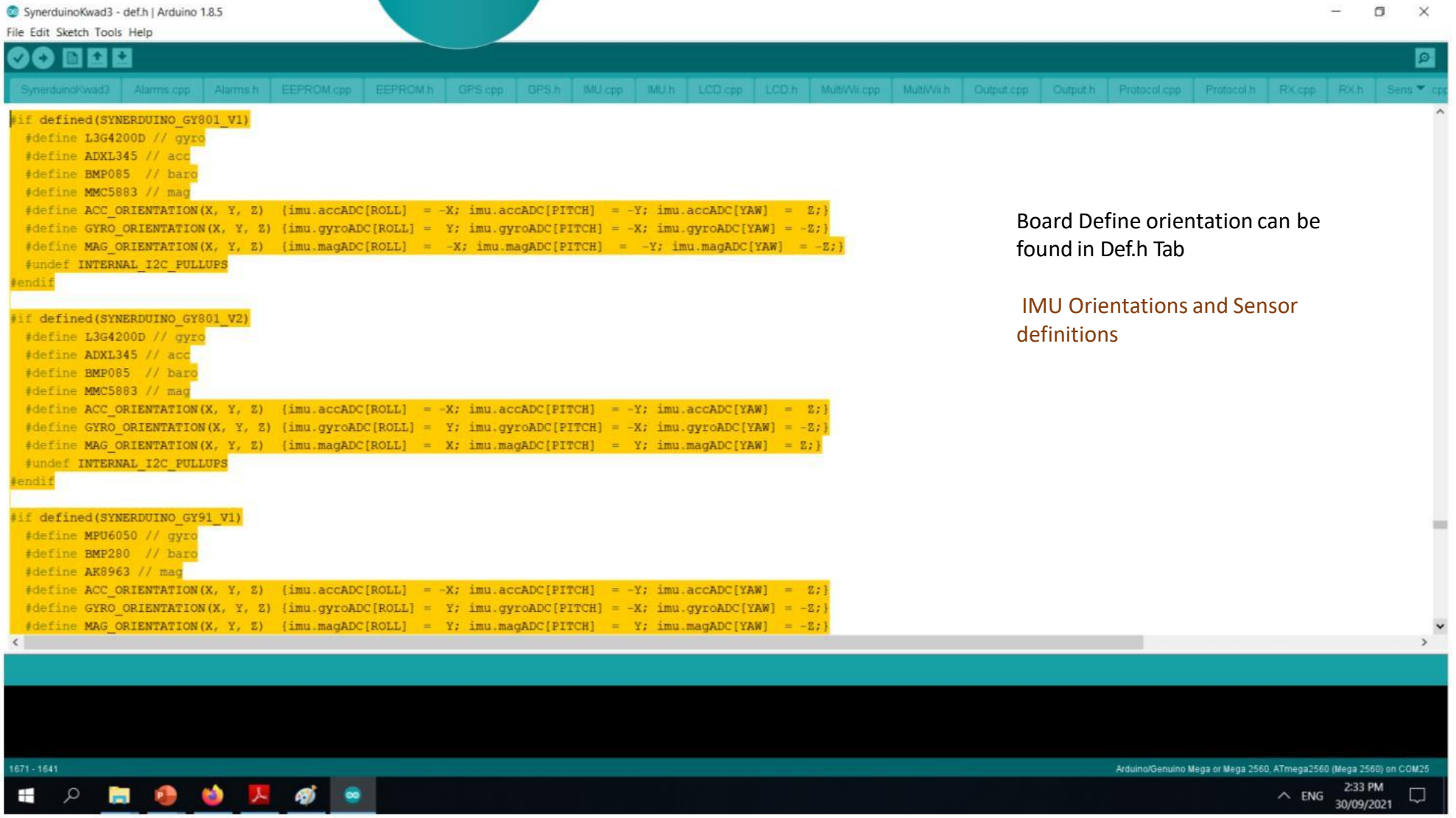
To use Combined IMU Boards **UnComment the # Synerduino board base on your version of the board and commend the // Independent Sensors//**

//#define SYNERDUINO_GY801_V1

//#define SYNERDUINO_GY801_V2

#define SYNERDUINO_GY91_V1

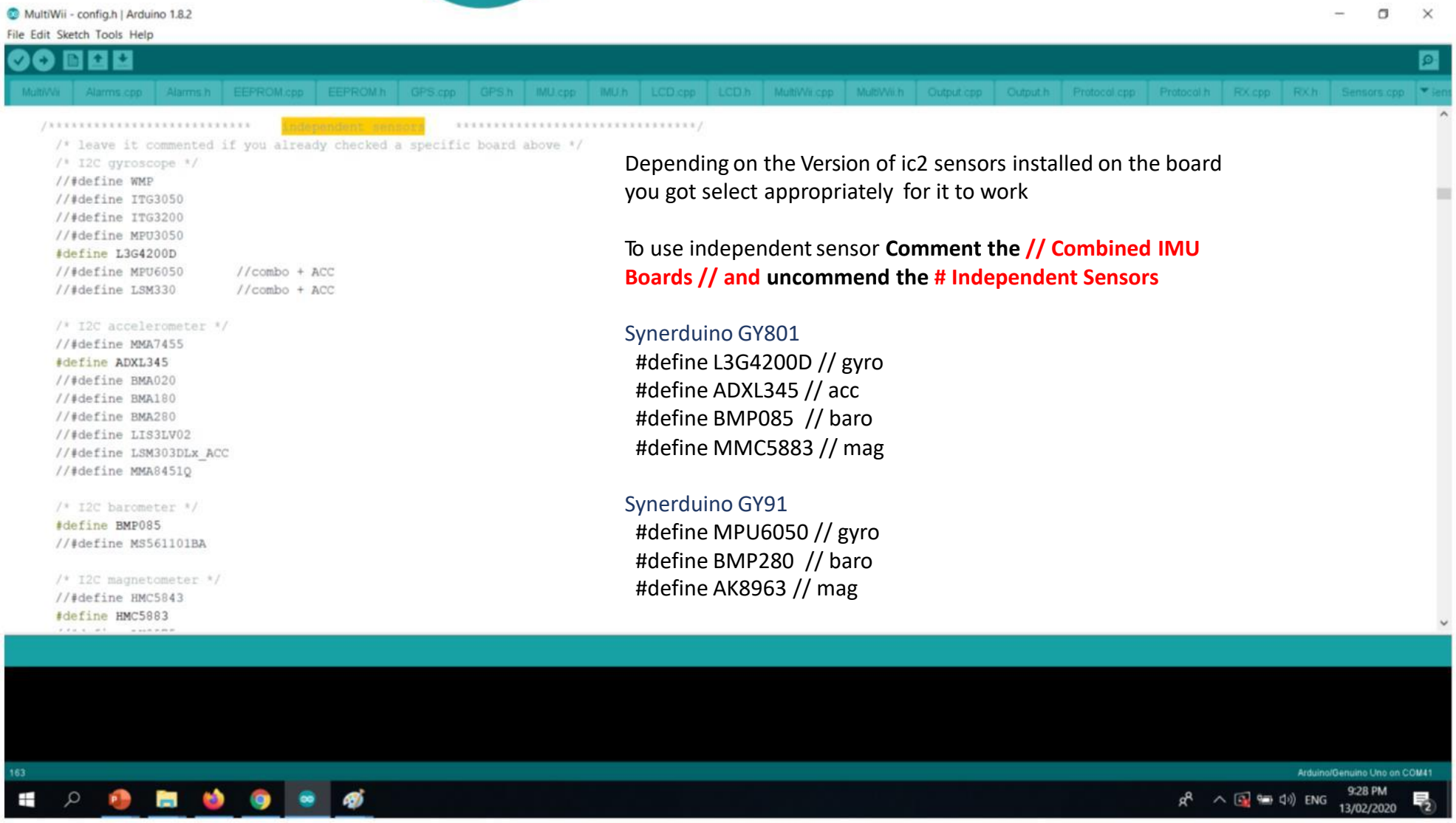
DEF.H



Board Define orientation can be found in Def.h Tab

IMU Orientations and Sensor definitions

CONFIG.H



Depending on the Version of ic2 sensors installed on the board you got select appropriately for it to work

To use independent sensor **Comment the // Combined IMU Boards // and uncommend the # Independent Sensors**

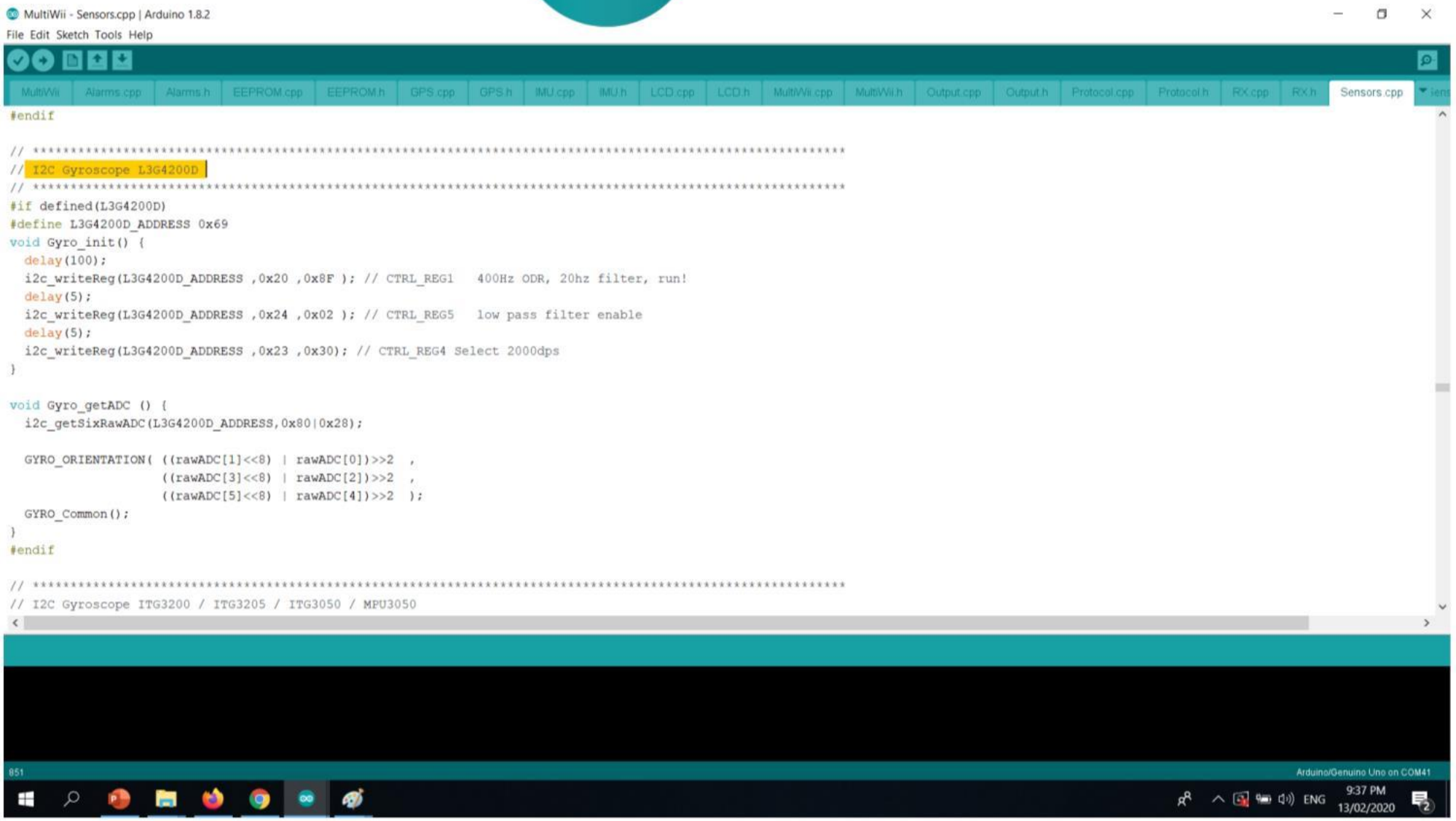
Synerduino GY801

```
#define L3G4200D // gyro
#define ADXL345 // acc
#define BMP085  // baro
#define MMC5883 // mag
```

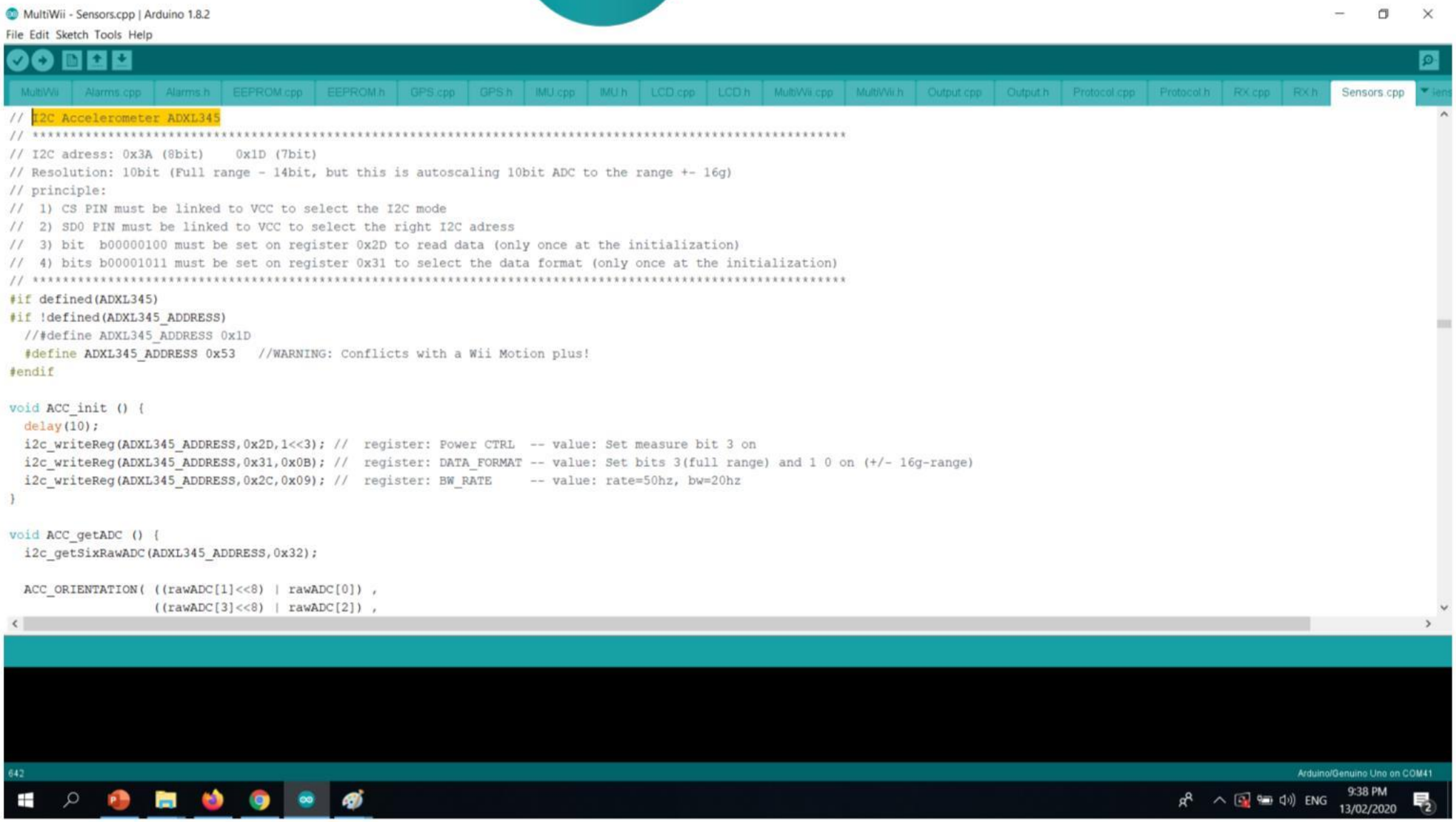
Synerduino GY91

```
#define MPU6050 // gyro
#define BMP280  // baro
#define AK8963 // mag
```

SENSORS.CPP


MultiWii - Sensors.cpp | Arduino 1.8.2
File Edit Sketch Tools Help
MultiWii
Alarms.cpp
Alarms.h
EEPROM.cpp
EEPROM.h
GPS.cpp
GPS.h
IMU.cpp
IMU.h
LCD.cpp
LCD.h
MultiWii.cpp
MultiWii.h
Output.cpp
Output.h
Protocol.cpp
Protocol.h
RX.cpp
RX.h
Sensors.cpp
#endif
// I2C Gyroscope L3G4200D
#if defined(L3G4200D)
#define L3G4200D_ADDRESS 0x69
void Gyro_init() {
delay(100);
i2c_writeReg(L3G4200D_ADDRESS ,0x20 ,0x8F); // CTRL_REG1 400Hz ODR, 20hz filter, run!
delay(5);
i2c_writeReg(L3G4200D_ADDRESS ,0x24 ,0x02); // CTRL_REG5 low pass filter enable
delay(5);
i2c_writeReg(L3G4200D_ADDRESS ,0x23 ,0x30); // CTRL_REG4 Select 2000dps
}
void Gyro_getADC () {
i2c_getSixRawADC(L3G4200D_ADDRESS,0x80|0x28);
GYRO_ORIENTATION(((rawADC[1]<<8) | rawADC[0])>>2 ,
((rawADC[3]<<8) | rawADC[2])>>2 ,
((rawADC[5]<<8) | rawADC[4])>>2);
GYRO_Common();
}
#endif
// I2C Gyroscope ITG3200 / ITG3205 / ITG3050 / MPU3050
851
Arduino/Genuino Uno on COM41
ENG
9:37 PM
13/02/2020

SENSORS.CPP

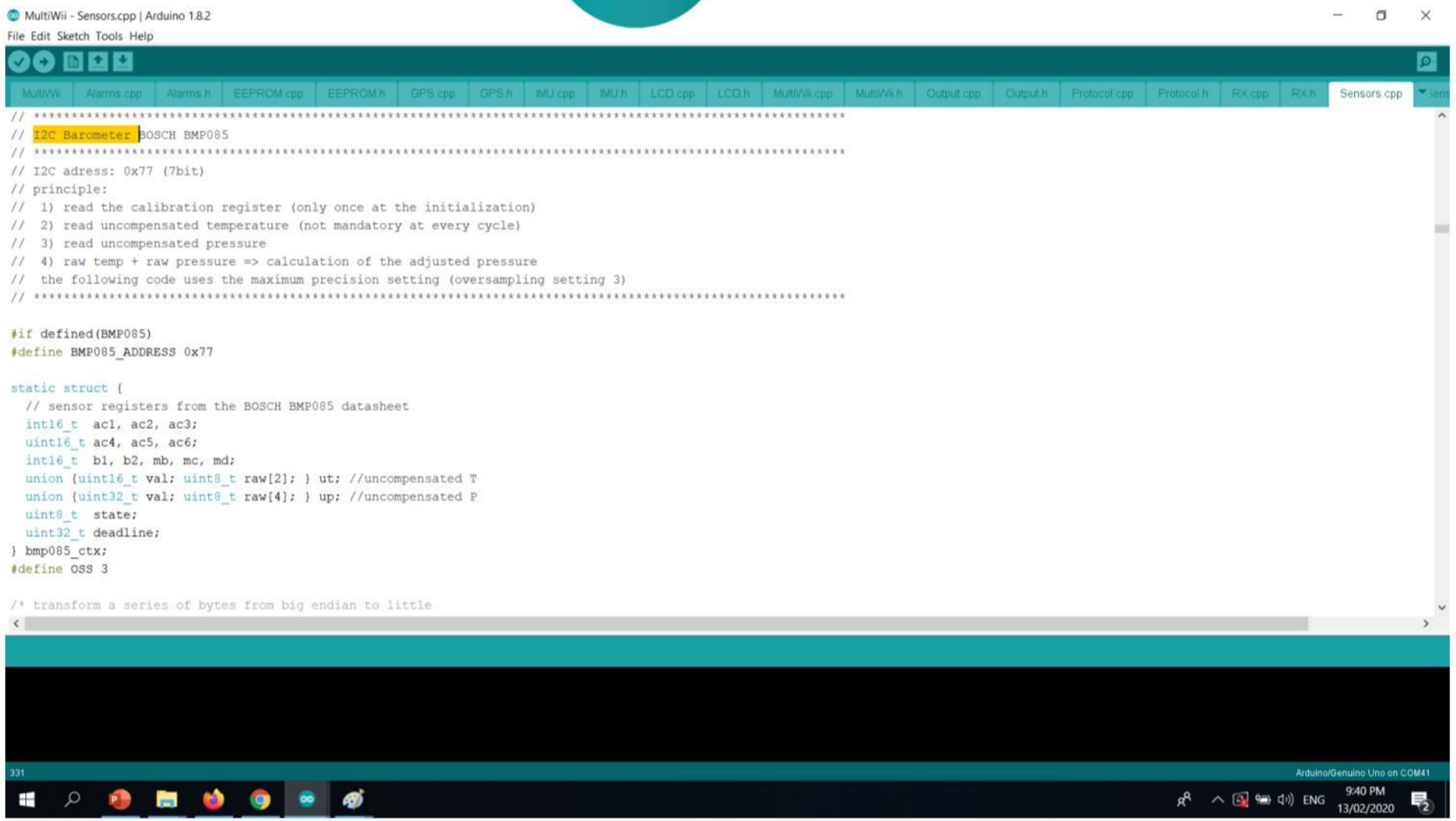

MultiWii - Sensors.cpp | Arduino 1.8.2
File Edit Sketch Tools Help
MultiWii
Alarms.cpp
Alarms.h
EEPROM.cpp
EEPROM.h
GPS.cpp
GPS.h
IMU.cpp
IMU.h
LCD.cpp
LCD.h
MultiWii.cpp
MultiWii.h
Output.cpp
Output.h
Protocol.cpp
Protocol.h
RX.cpp
RX.h
Sensors.cpp
// I2C Accelerometer ADXL345
// **
// I2C adress: 0x3A (8bit)   0x1D (7bit)
// Resolution: 10bit (Full range - 14bit, but this is autoscaling 10bit ADC to the range +- 16g)
// principle:
// 1) CS PIN must be linked to VCC to select the I2C mode
// 2) SDO PIN must be linked to VCC to select the right I2C adress
// 3) bit b00000100 must be set on register 0x2D to read data (only once at the initialization)
// 4) bits b00001011 must be set on register 0x31 to select the data format (only once at the initialization)
// **
#if defined(ADXL345)
#if !defined(ADXL345_ADDRESS)
  //#define ADXL345_ADDRESS 0x1D
  #define ADXL345_ADDRESS 0x53   //WARNING: Conflicts with a Wii Motion plus!
#endif

void ACC_init () {
  delay(10);
  i2c_writeReg(ADXL345_ADDRESS,0x2D,1<<3); // register: Power CTRL -- value: Set measure bit 3 on
  i2c_writeReg(ADXL345_ADDRESS,0x31,0x0B); // register: DATA_FORMAT -- value: Set bits 3(full range) and 1 0 on (+/- 16g-range)
  i2c_writeReg(ADXL345_ADDRESS,0x2C,0x09); // register: BW_RATE   -- value: rate=50hz, bw=20hz
}

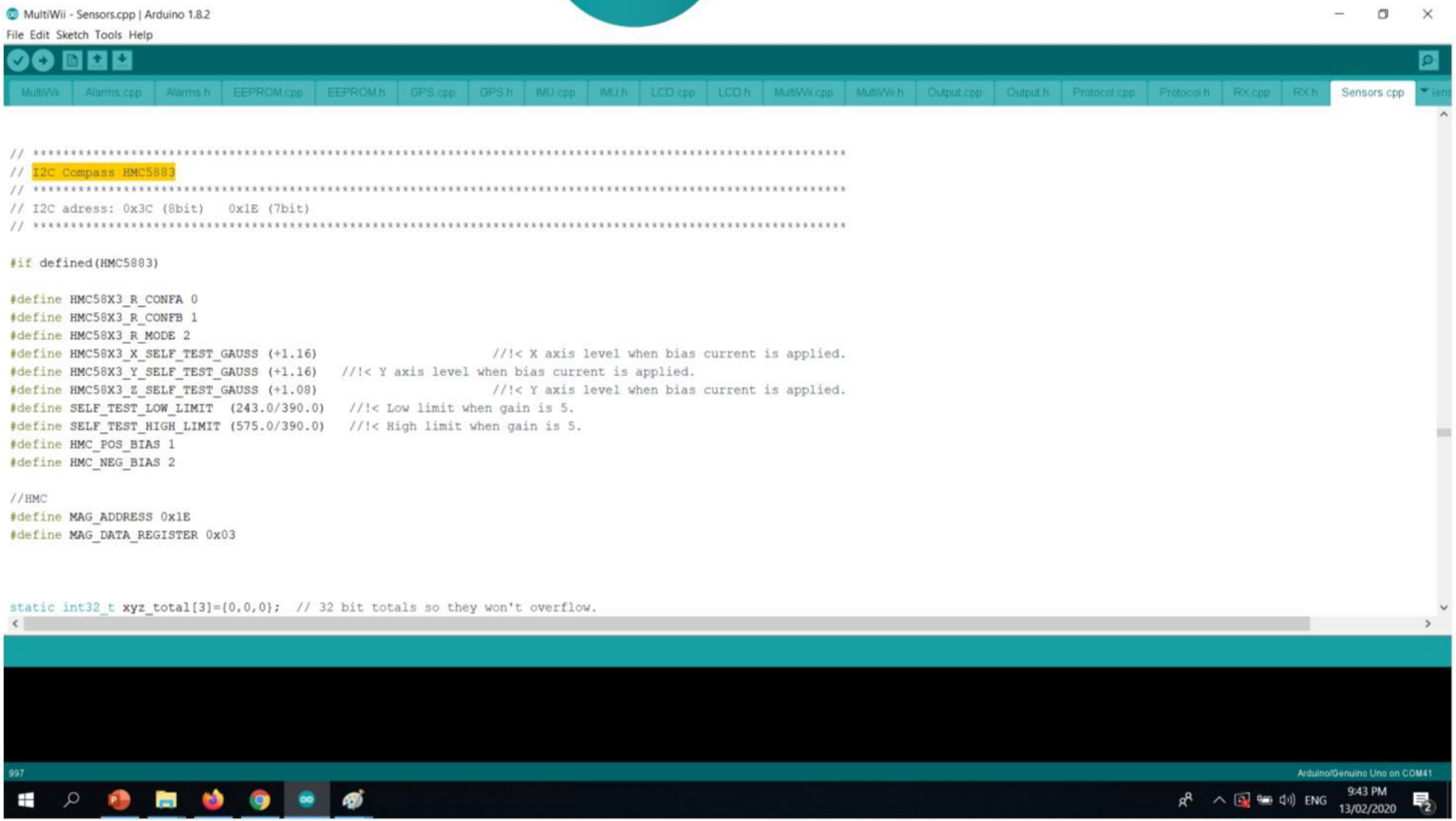
void ACC_getADC () {
  i2c_getSixRawADC(ADXL345_ADDRESS,0x32);

  ACC_ORIENTATION(((rawADC[1]<<8) | rawADC[0]) ,
                   ((rawADC[3]<<8) | rawADC[2]) ,
642
Arduino/Genuino Uno on COM41
ENG
9:38 PM
13/02/2020

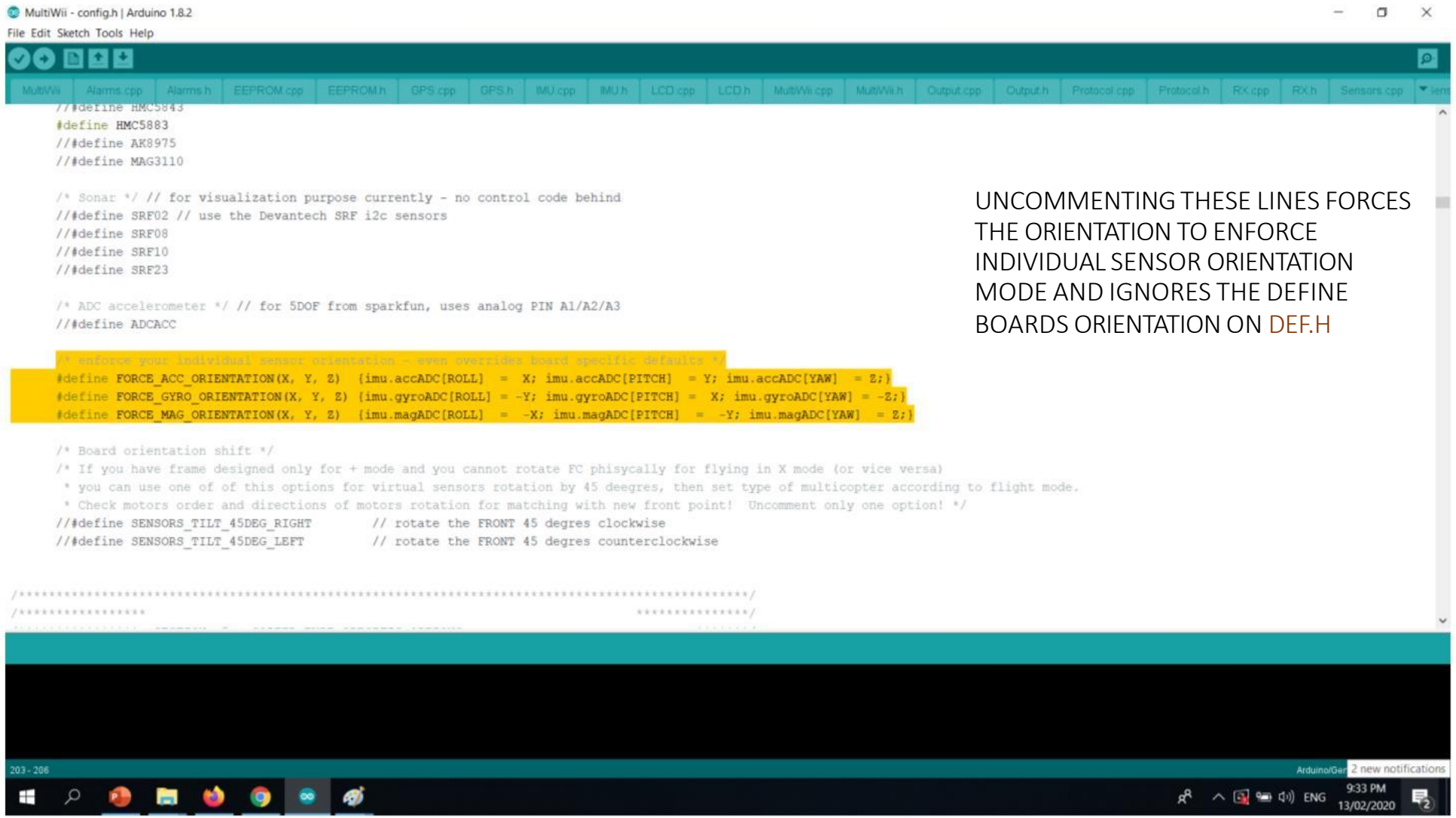
SENSORS.CPP


MultiWii - Sensors.cpp | Arduino 1.8.2
File Edit Sketch Tools Help
MultiWii
Alarms.cpp
Alarms.h
EEPROM.cpp
EEPROM.h
GPS.cpp
GPS.h
IMU.cpp
IMU.h
LCD.cpp
LCD.h
MultiWii.cpp
MultiWii.h
Output.cpp
Output.h
Protocol.cpp
Protocol.h
RX.cpp
RX.h
Sensors.cpp
// I2C Barometer BOSCH BMP085
// I2C adress: 0x77 (7bit)
// principle:
// 1) read the calibration register (only once at the initialization)
// 2) read uncompensated temperature (not mandatory at every cycle)
// 3) read uncompensated pressure
// 4) raw temp + raw pressure => calculation of the adjusted pressure
// the following code uses the maximum precision setting (oversampling setting 3)
#if defined(BMP085)
#define BMP085_ADDRESS 0x77
static struct {
// sensor registers from the BOSCH BMP085 datasheet
int16_t ac1, ac2, ac3;
uint16_t ac4, ac5, ac6;
int16_t b1, b2, mb, mc, md;
union {uint16_t val; uint8_t raw[2]; } ut; //uncompensated T
union {uint32_t val; uint8_t raw[4]; } up; //uncompensated P
uint8_t state;
uint32_t deadline;
} bmp085_ctx;
#define OSS 3
/* transform a series of bytes from big endian to little
331
Arduino/Genuino Uno on COM41
ENG
9:40 PM
13/02/2020

SENSORS.CPP


MultiWii - Sensors.cpp | Arduino 1.8.2
File Edit Sketch Tools Help
MultiWii
Alarms.cpp
Alarms.h
EEPROM.cpp
EEPROM.h
GPS.cpp
GPS.h
IMU.cpp
IMU.h
LCD.cpp
LCD.h
MultiWii.cpp
MultiWii.h
Output.cpp
Output.h
Protocol.cpp
Protocol.h
RX.cpp
RX.h
Sensors.cpp
// I2C Compass HMC5883
// I2C adress: 0x3C (8bit) 0x1E (7bit)
#if defined(HMC5883)
#define HMC58X3_R_CONFA 0
#define HMC58X3_R_CONFB 1
#define HMC58X3_R_MODE 2
#define HMC58X3_X_SELF_TEST_GAUSS (+1.16) //!< X axis level when bias current is applied.
#define HMC58X3_Y_SELF_TEST_GAUSS (+1.16) //!< Y axis level when bias current is applied.
#define HMC58X3_Z_SELF_TEST_GAUSS (+1.08) //!< Y axis level when bias current is applied.
#define SELF_TEST_LOW_LIMIT (243.0/390.0) //!< Low limit when gain is 5.
#define SELF_TEST_HIGH_LIMIT (575.0/390.0) //!< High limit when gain is 5.
#define HMC_POS_BIAS 1
#define HMC_NEG_BIAS 2
//HMC
#define MAG_ADDRESS 0x1E
#define MAG_DATA_REGISTER 0x03
static int32_t xyz_total[3]={0,0,0}; // 32 bit totals so they won't overflow.
997
Arduino/Genuino Uno on COM41
ENG
9:43 PM
13/02/2020

SENSOR ORIENTATION CONFIG.H



UNCOMMENTING THESE LINES FORCES THE ORIENTATION TO ENFORCE INDIVIDUAL SENSOR ORIENTATION MODE AND IGNORES THE DEFINE BOARDS ORIENTATION ON DEF.H

SENSORS IMU ORIENTATION IS IMPORTANT SEE TO IT THE ACC GYRO AND MAG ALL COMPLIMENT EACH OTHER

SERIAL COM

CONFIG.H

SERIAL BAUD RATE

WILL DEPEND ON WHAT BAUD YOUR TELEMETRY MODULE ARE SET INTO YOU CAN CHANGE IT TO SUITE THE PORT YOUR DEVICE IS CONNECTED TO.

SERIAL 0 CAN BE USE FOR TELEMETRY GIVEN NOTHING IS CONNECTED TO THE USB AT THIS POINT. AND FIRMWARE MUST BE FLISH PRIOR TO HOOKING UP ANYTHING TO THIS PINS

SERIAL 1 , 3 IS RESERVE FOR TELEMETRY

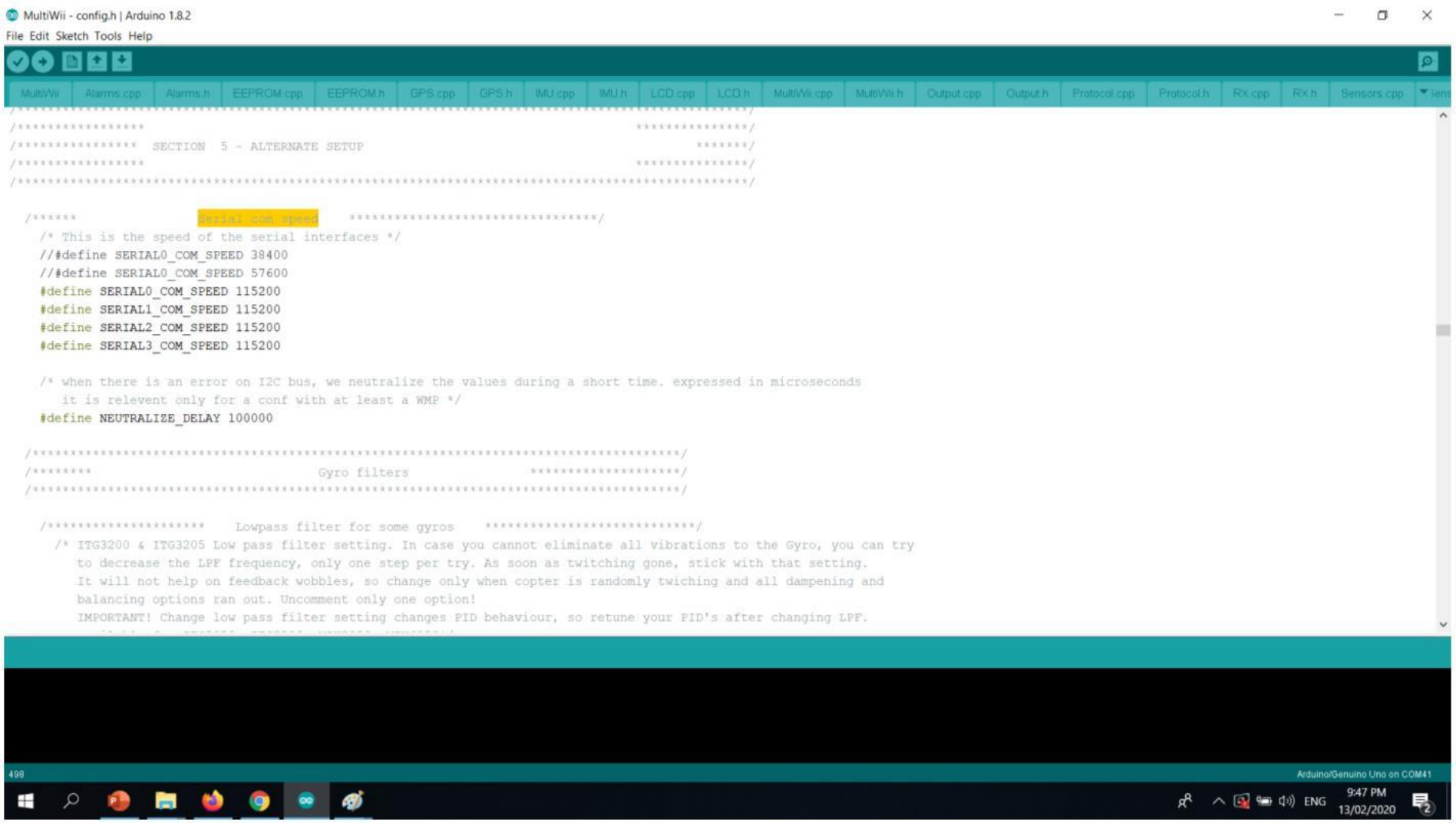
115200 FOR BLUETOOTH HC-05

38400 FOR XBEE RADIO

57600 for SIK RADIO

SERIAL 2 IS RESERVE FOR GPS

NMEA Baud 57600



BLUETOOTH

Bluetooth setup with the USB TTL and Arduino IDE

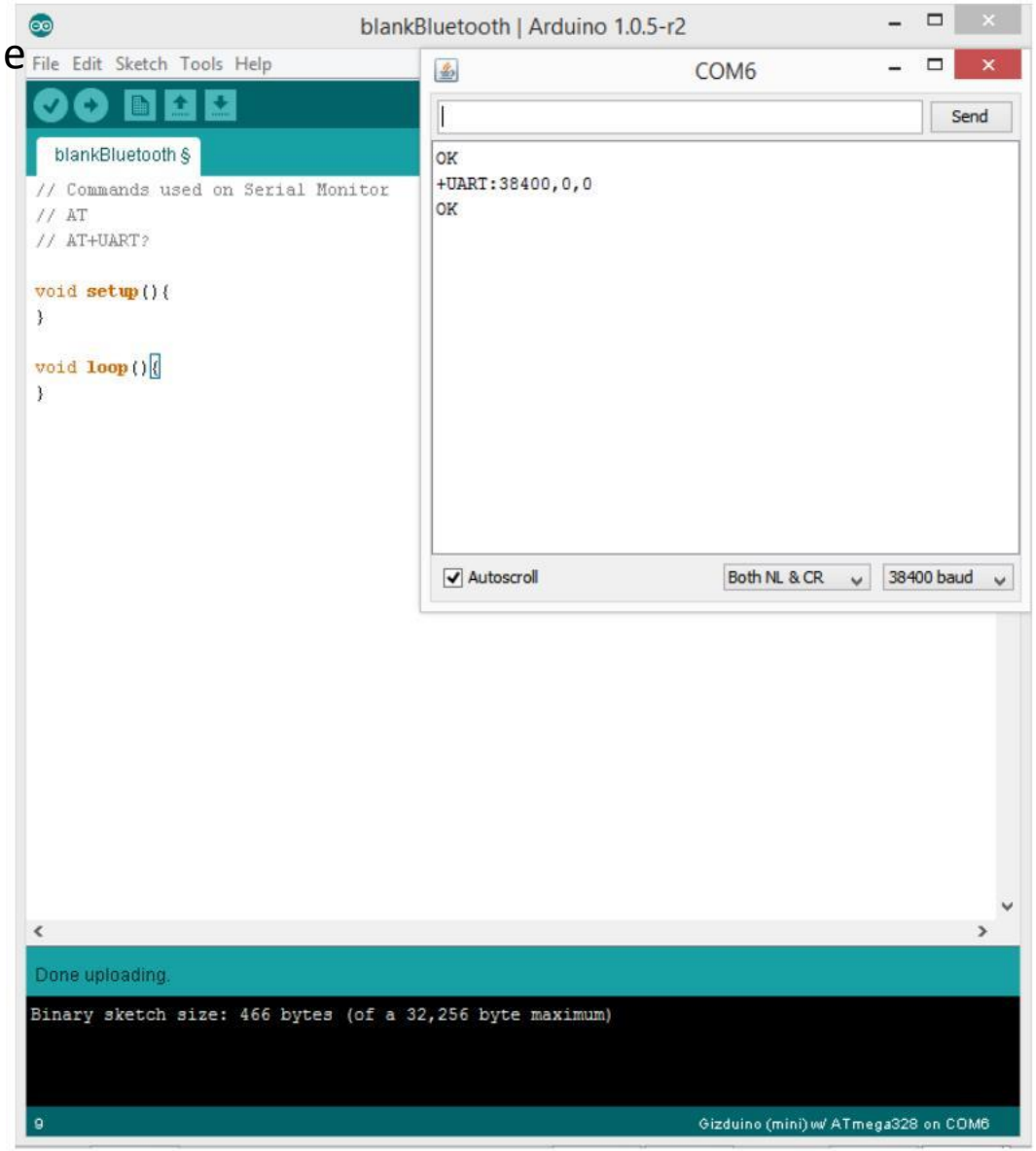
Arduino IDE>Tools>Serial Monitor (Push Button Before Connecting the USB) Set (Baud 38400) (Both NL & CR)

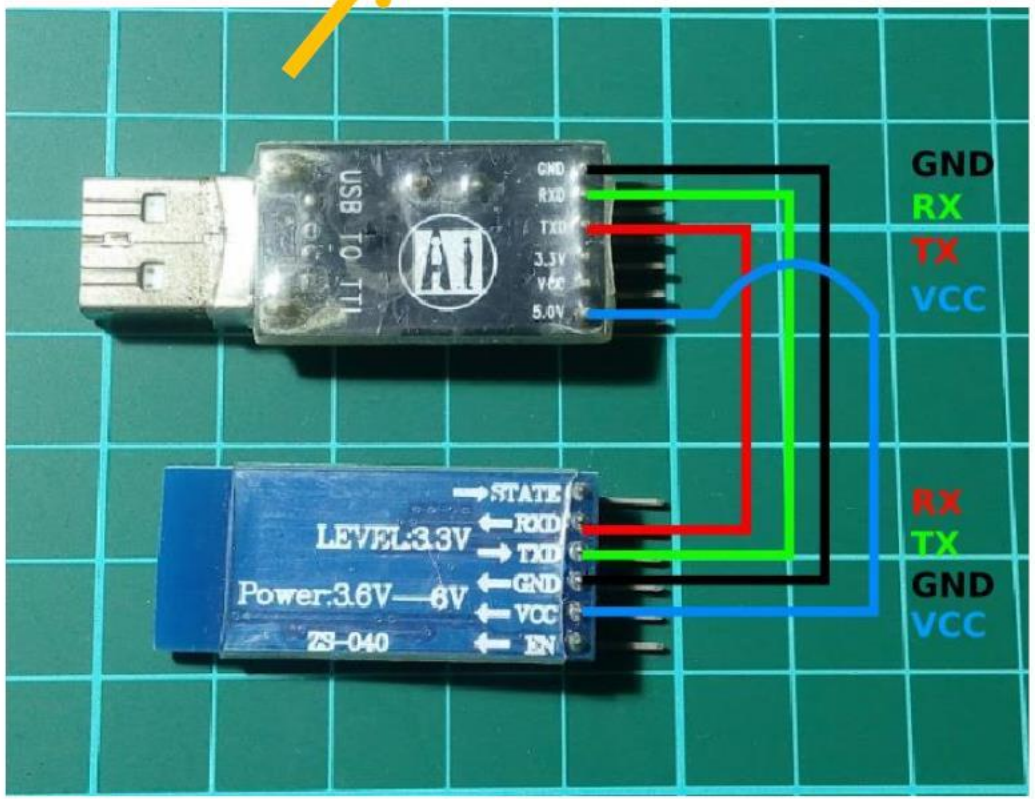
AT : check the connection
AT+VERSION : Check Version

HC-05 (Recommended)

AT+NAME=ArduinoDrone
AT+PSWD=1234 (Version 2)
AT+PSWD="1234" (Version 3)
AT+UART=115200,1,0

HC-05 (Recommended)





Bluetooth setup with the USB TTL and Arduino IDE

Arduino IDE>Tools>Serial Monitor (hold the Button while Plugging USB) to go programming mode Set (Baud 38400) (Both NL & CR)

Hold Press when sending AT command (Version 5)

AT : check the connection
AT+VERSION : Check Version

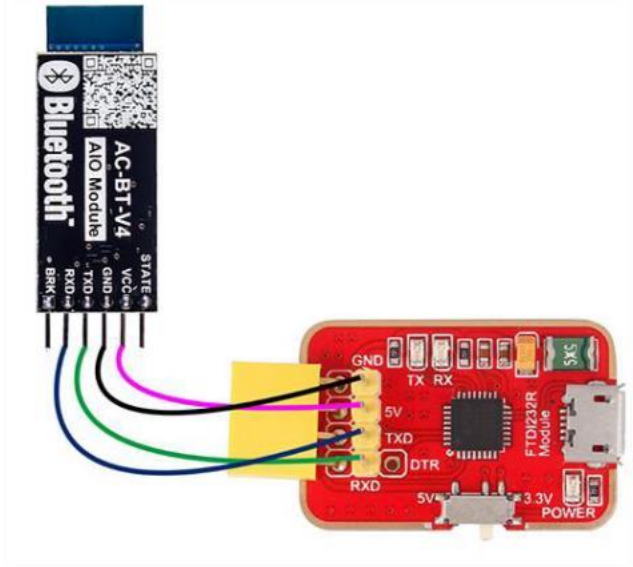
HC-05 (Recommended)

AT+NAME=Change name (Synerduino)
AT+PSWD=1234 (Version 2)
AT+PSWD="1234" (Version 3)
AT+PIN"1234" (Possibly works on Version 5)
AT+UART=115200,1,0 (115200 FOR BLUETOOTH)

HC-06

AT+NAME: Change name
AT+PIN: change pin, xxxx is the pin, again, no space.
AT+BAUDX, where X=1 to 9
1 set to 1200bps, 2 set to 2400bps, 3 set to 4800bps
4 set to 9600bps (Default)
5 set to 19200bps,6 set to 38400bps, 7 set to 57600bps
8 set to 115200bps

HM-10 Bluetooth



Setup with FTDI + Arduino Serial Monitor + AT Command

AT+NAME? (Query name)
AT+ADDR? ((Query Mac address)

First you will need to Query the native MAC address using AT Command **AT+ADDR?**
You will get something like this 20C38FF61DA1, each BLE has a unique MAC address.

Use **AT+CON[param1]** and **AT+ROLE[param1]** to pair to another device.

Example
BLE A has Mac Address 11C11FF11DA1, I used **AT+ADDR?** to figure it out BLE B has Mac Address 22C22FF22DA2, I used **AT+ADDR?** to figure it out
Send **AT+CON**22C22FF22DA2 to BLE A Send **AT+CON**11C11FF11DA1 to BLE B (Send the B address to A, A address to B)
Send **AT+ROLE0** to BLE ASend **AT+ROLE1** to BLE B (Doesn't matter which one)
Now it's ready to use on you ATMEGA 328P, Arduino or Attiny. **The red light will stay solid after the connection has been made on both BLE. This should take less than a second.**

HM-10 (Original)

AT (Check if new configuration is working)

AT+NAME (Query name)

AT+ADDR (Query Mac address)

AT+BAUD (Query Baud)

AT+PASS (Query current Pincode)

AT+PIN **(Query current Pincode on some BL module)**

AT+TYPE (Query authentication mode)

AT+ROLE (Query Peripheral (Slave) or Central (Master) mode)

AT+TYPE

0:Not need PIN Code

1:Auth not need PIN

2:Auth with PIN

3:Auth and bond

AT+BAUD

0 – 9600:

1 – 19200

2 – 38400

3 – 57600 (Some BL its 4800)

4 – 115200

5 – 4800

6 – 2400

7 – 1200

8 – 230400 (Some BL its 115200)

AT+NAMEArduinoDrone

AT+BAUD4 set baud to 115200 (we want this for high speed)

AT+BAUD8 set baud to 115200 (on some BL module)

AT+PASS123456 Set password to 123456

AT+PIN123456 Set password to 123456 (on some BL module)

AT+ROLE

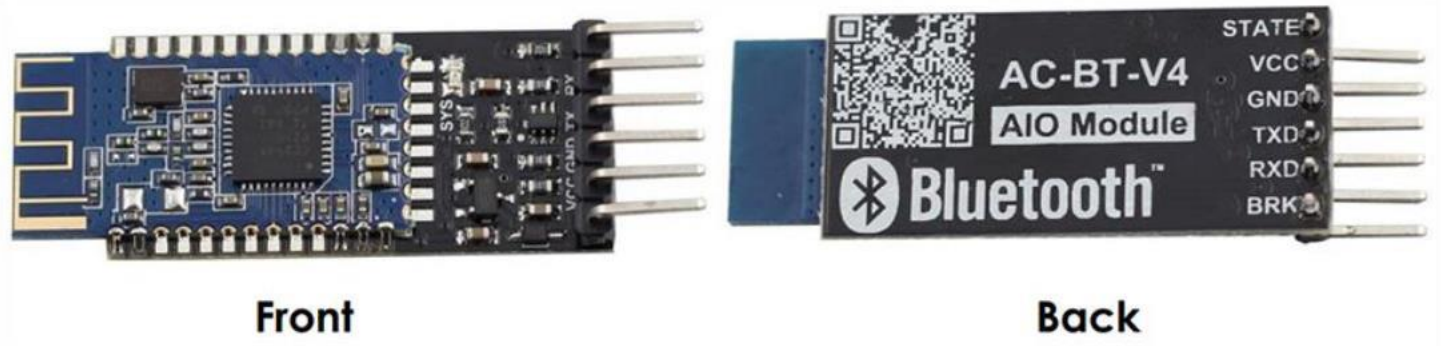
0 = Slave or Peripheral

1 = Master or Central.

AT+TYPE2

AT+TYPE1 (on Some BL modules)

AT+ROLE0



Front Back

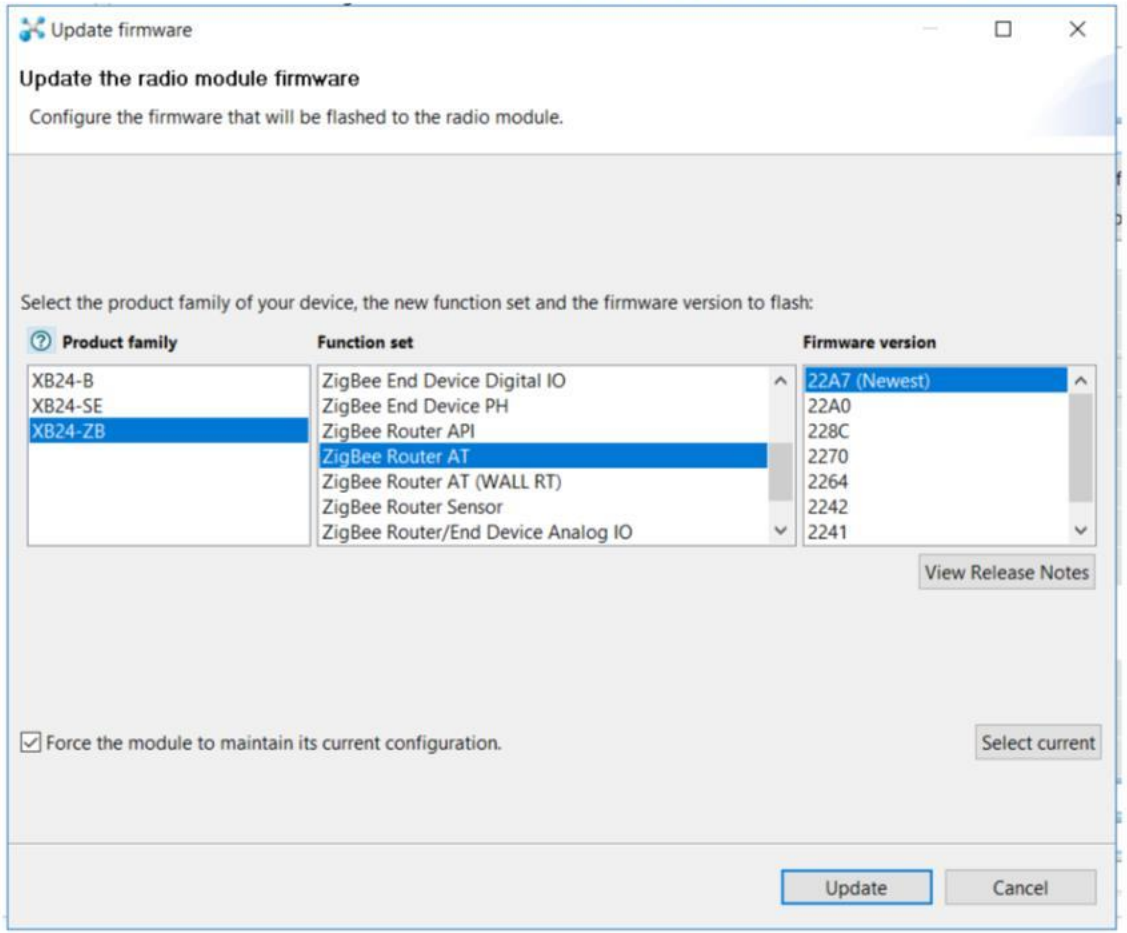
Note : there are several clones of this type in the market that can be very difficult to setup

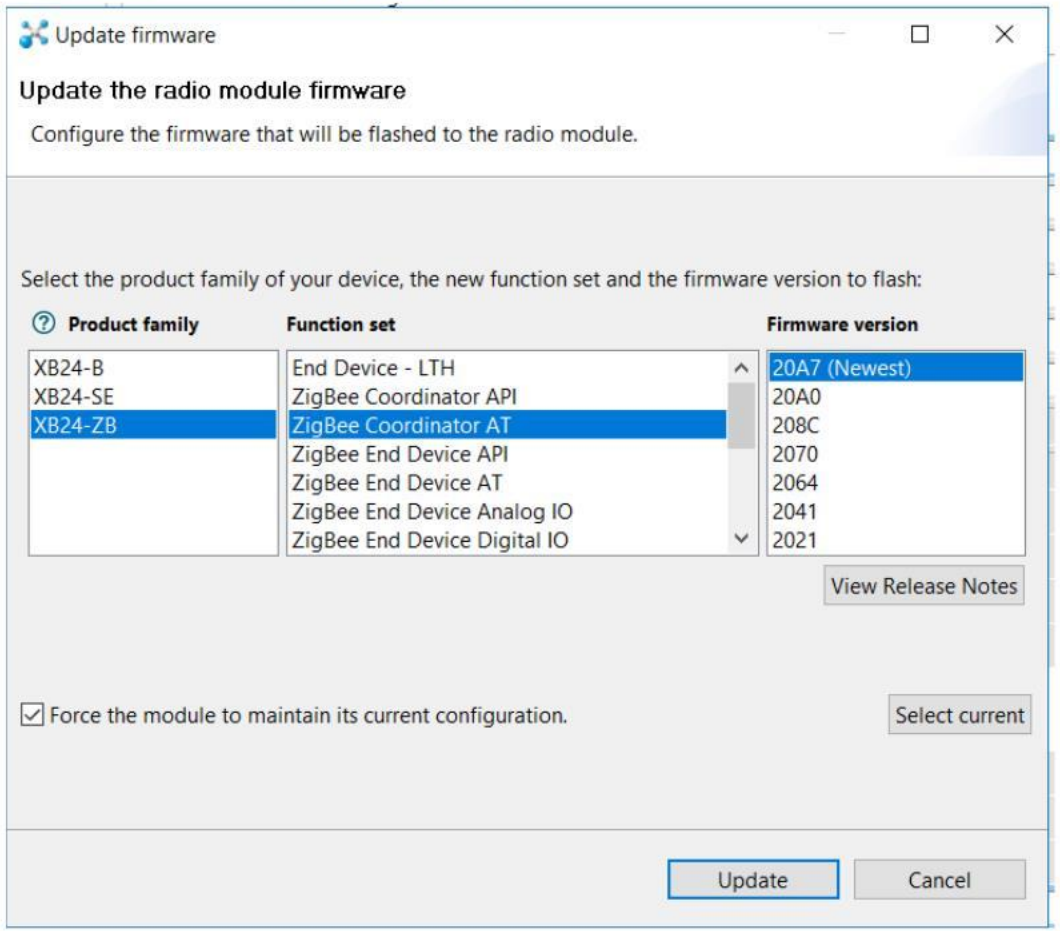
Download [40003026_CXCTU-1.zip](#)

XBEE RADIO

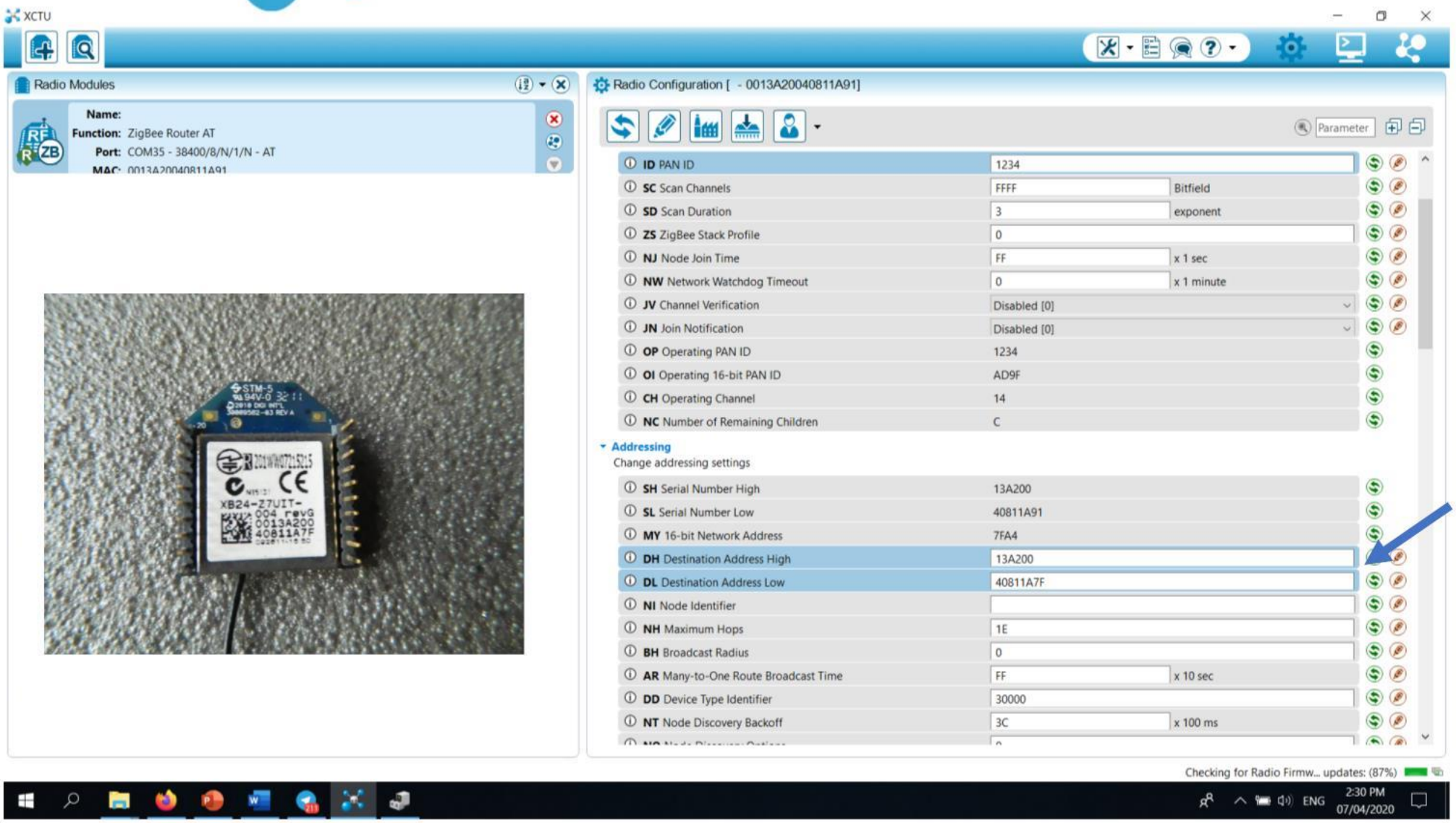
GROUND STATION ROUTER
38400 8/N/1/N - AT

AIRCRAFT COORDINATOR 38400 8/N/1/N - AT

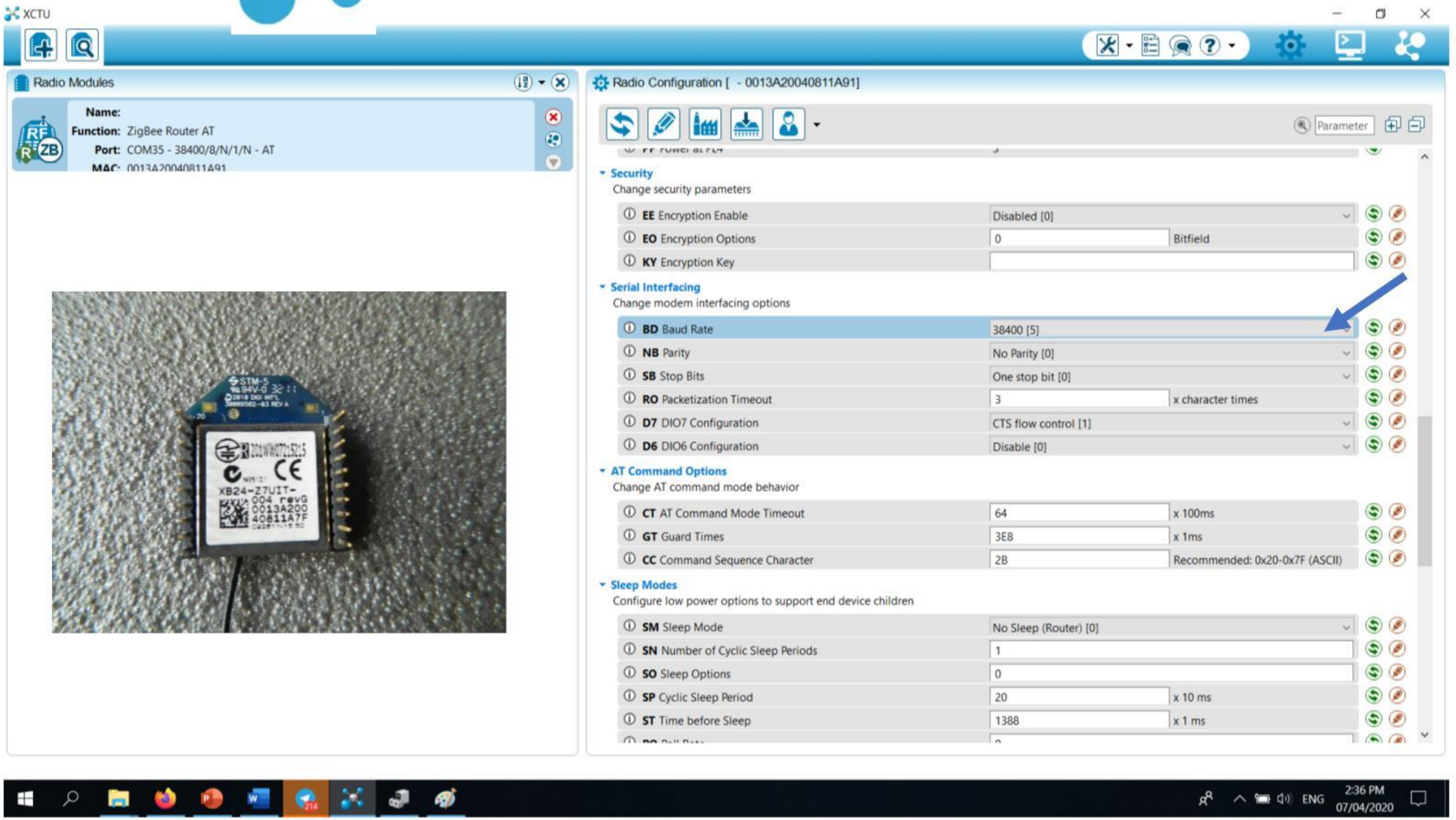




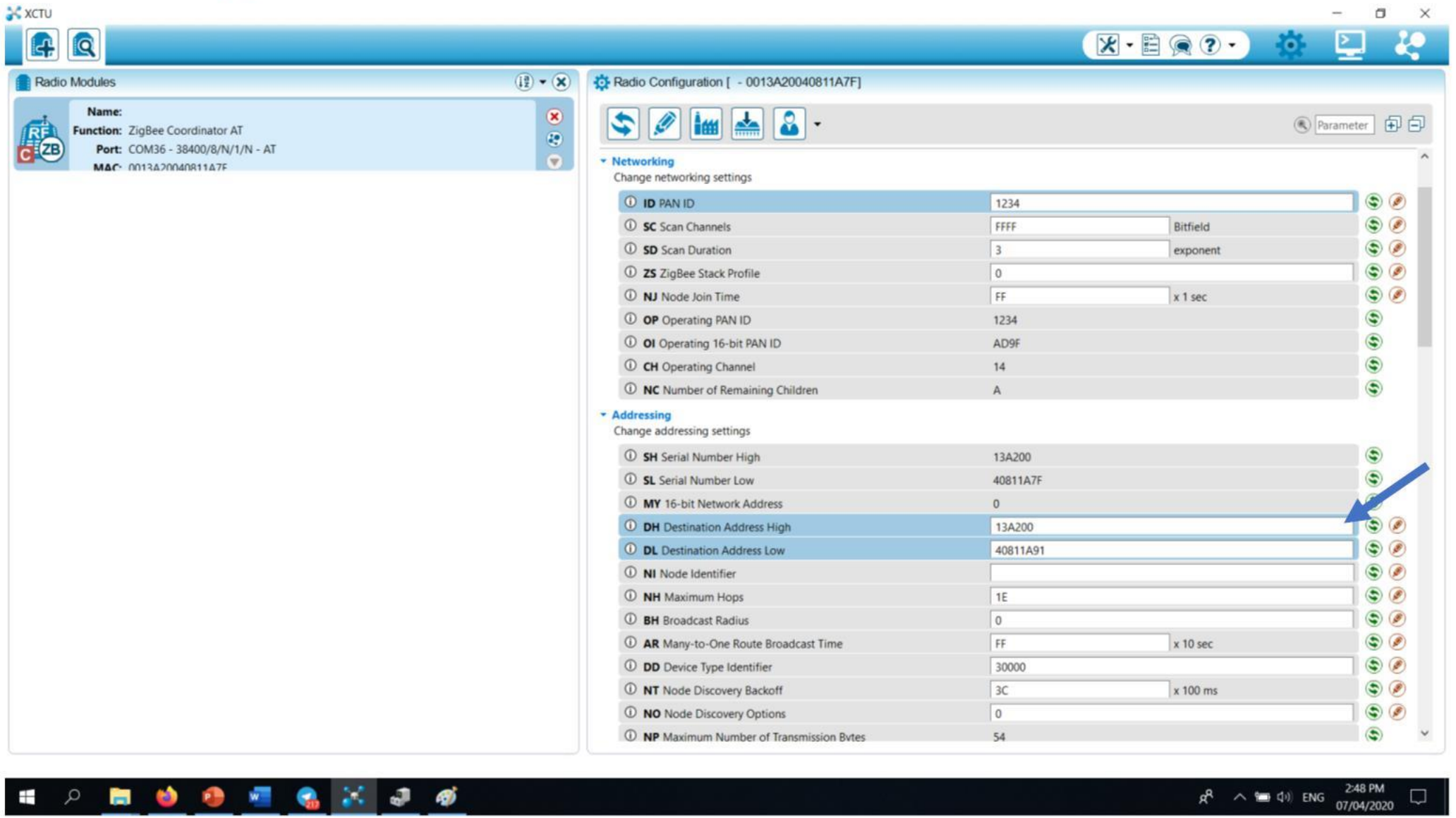
GROUND STATION



GROUND STATION



AIRCRAFT

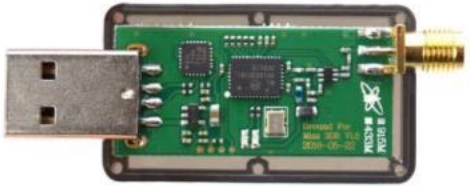

XCTU
Radio Modules
Name:
Function: ZigBee Coordinator AT
Port: COM36 - 38400/8/N/1/N - AT
MAC: 0013A20040811A7F
Radio Configuration [- 0013A20040811A7F]
Parameter
Networking
Change networking settings
ID PAN ID 1234
SC Scan Channels FFFF Bitfield
SD Scan Duration 3 exponent
ZS ZigBee Stack Profile 0
NJ Node Join Time FF x 1 sec
OP Operating PAN ID 1234
OI Operating 16-bit PAN ID AD9F
CH Operating Channel 14
NC Number of Remaining Children A
Addressing
Change addressing settings
SH Serial Number High 13A200
SL Serial Number Low 40811A7F
MY 16-bit Network Address 0
DH Destination Address High 13A200
DL Destination Address Low 40811A91
NI Node Identifier
NH Maximum Hops 1E
BH Broadcast Radius 0
AR Many-to-One Route Broadcast Time FF x 10 sec
DD Device Type Identifier 30000
NT Node Discovery Backoff 3C x 100 ms
NO Node Discovery Options 0
NP Maximum Number of Transmission Bytes 54
ENG
2:48 PM
07/04/2020

SIK SERIAL RADIO

38400 OR 57600 FOR SIK RADIO DEPENDING IF USES 433MHZ OR 900MHZ

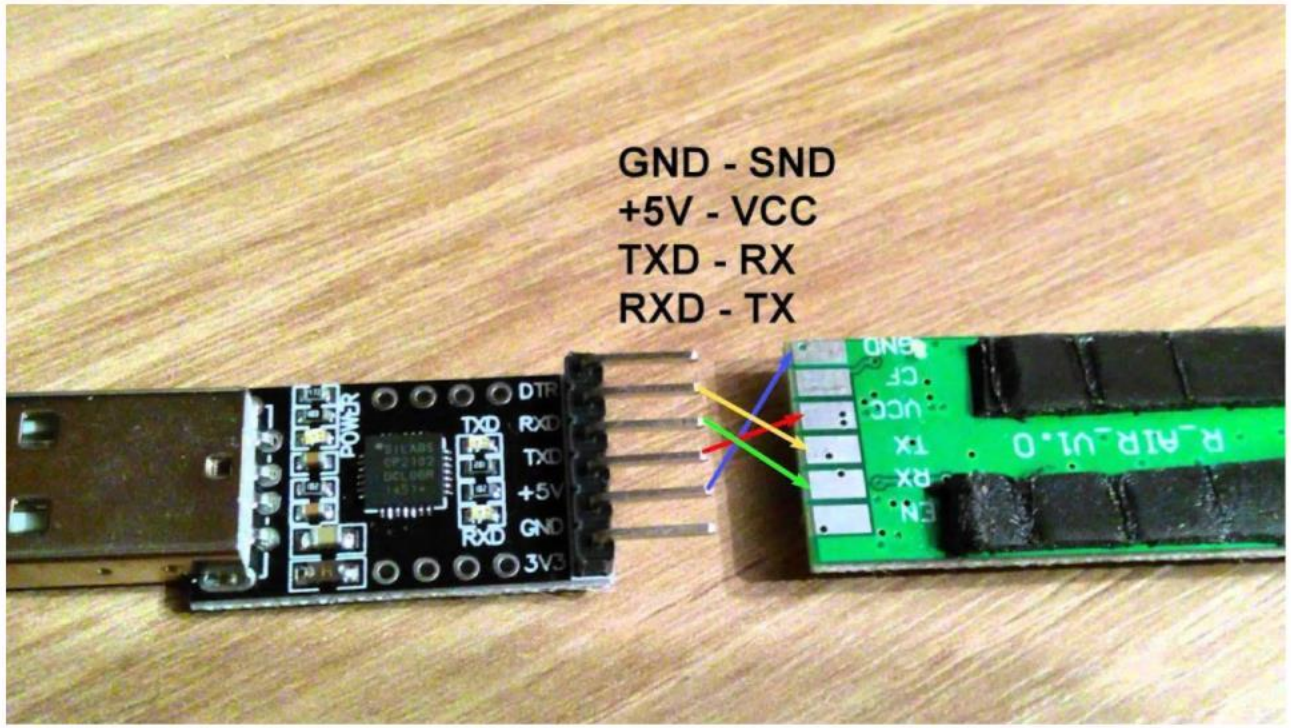
3drradioconfig Download

RadioTelemetry Air Module



RadioTelemetry Ground Module

Again to setup you require an USB-TTL module to connect to the serial port to configure both the module how ever mostlikely you only need to do this for the vehicle unit as the ground unit has an USB build into



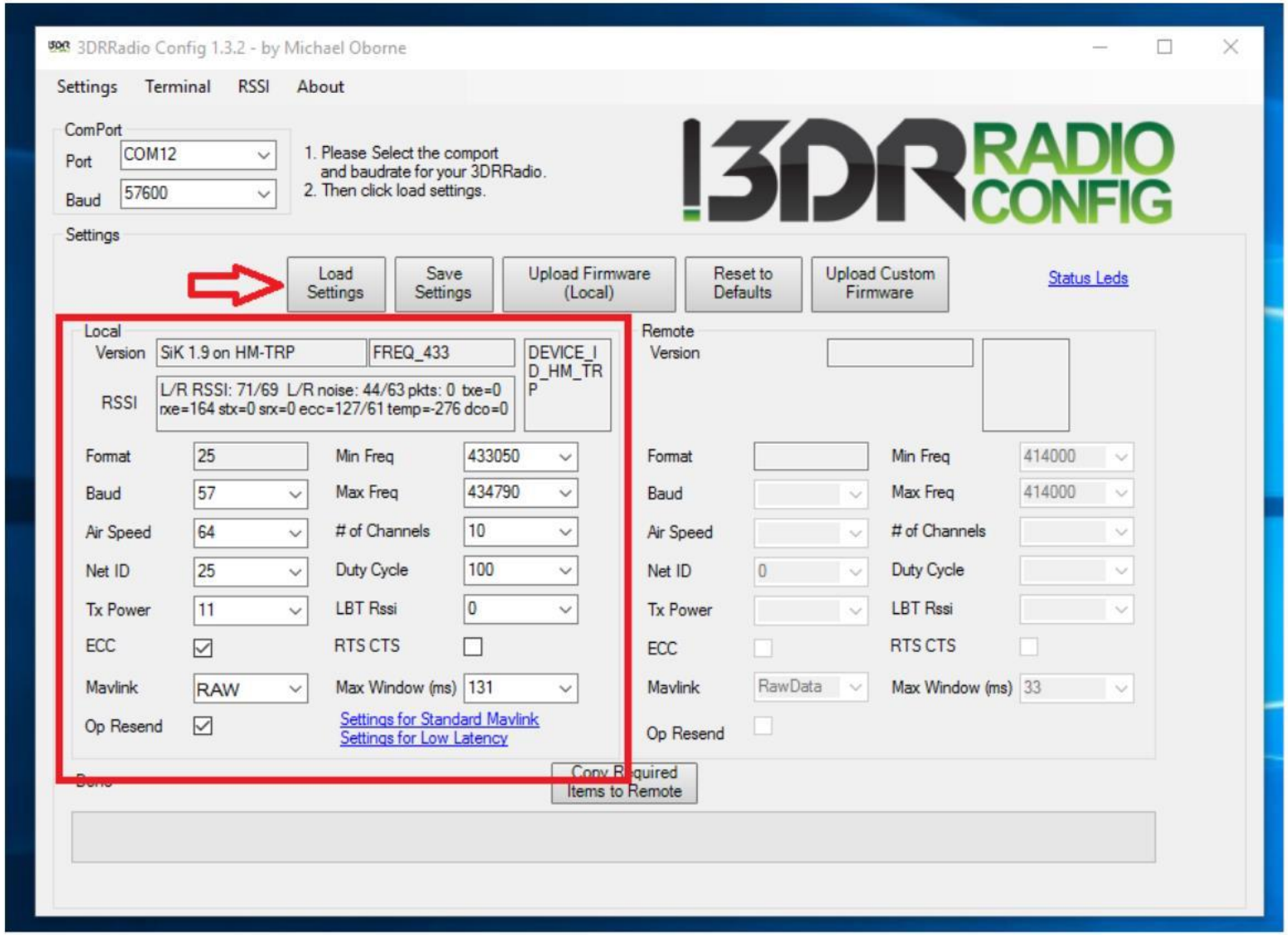
Manually configuring the telemetry kit for Synerduino uses the 3DR radio Config

http://vps.oborne.me/3drradioconfig.zip

Also available in the synerduino page

Both Vehicle and Ground station unit must have similar in the following

- Versions
- Frequency
- Baud (38400 or 57600 ensure)
- Airspeed
- Net ID (in cases you need to assign multiple drones each having their own ID)
- Tx power
- Mavlink (RAW –Synerduino uses Format)



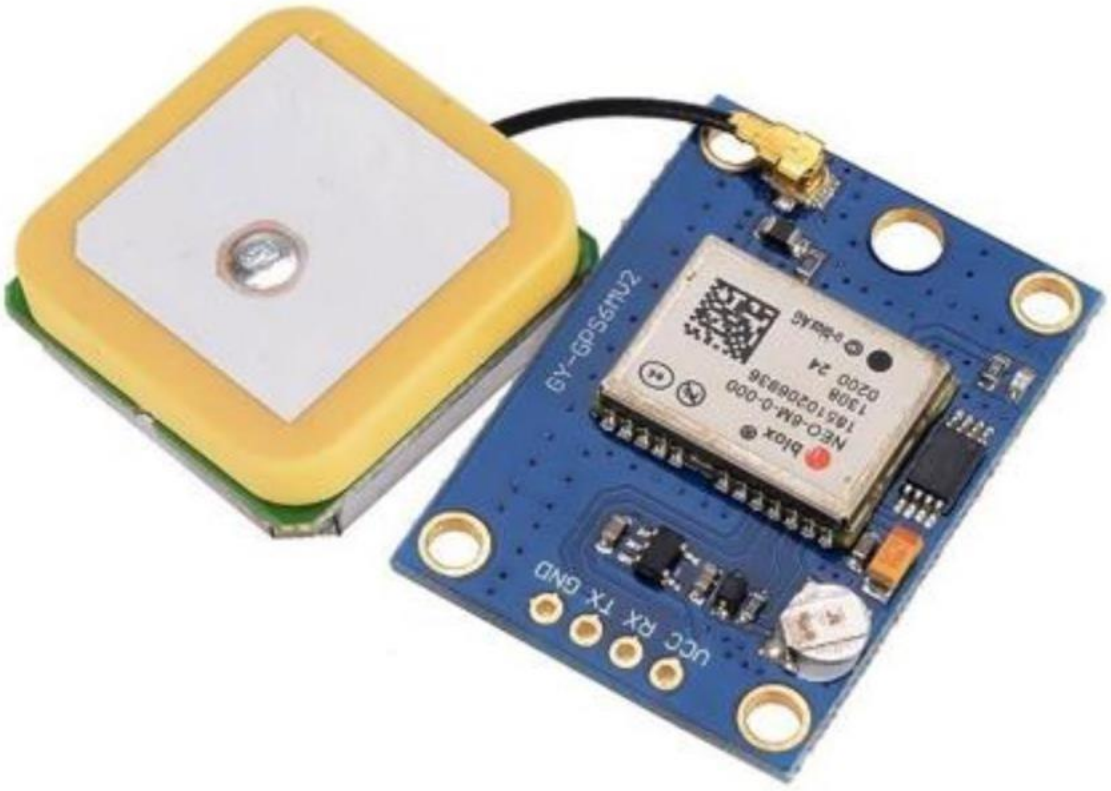
GPS

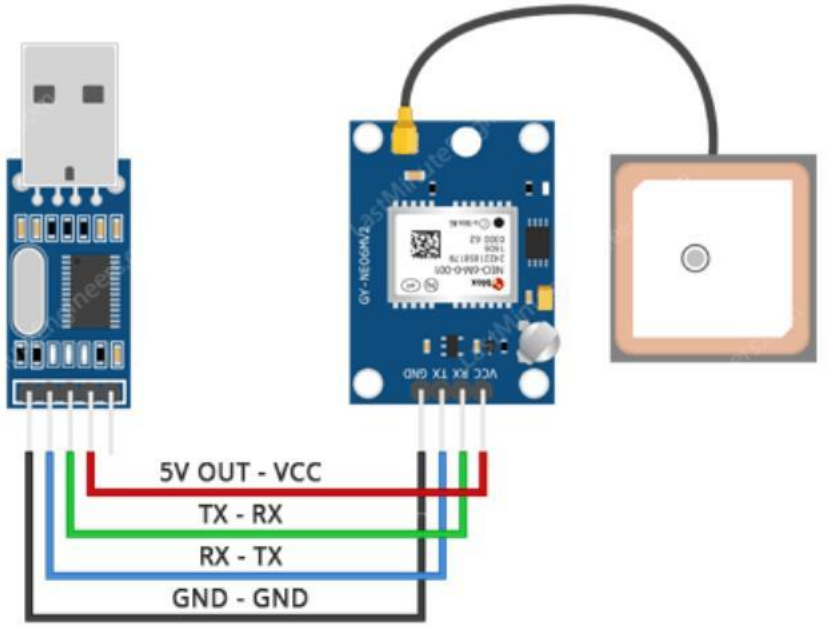
UBLOX Protocol

NMEA Protocol

NOTE: GPS CONFIGURING ONLY WORKS WHEN GPS MODULE COMES WITH EEPROM OR FLASH MEMORY. AS YOUR SELLER IF IT COMES WITH THOSE FUNCTIONS

GPS CONFIGURING





U BLOX NEO 6

PLUG IN TO SERIAL TX 2 RX 2

USB TTL TO PROGRAM THE GPS

THIS GOES SAME ON THE DRONE SHIELD

GPS CONFIGURING

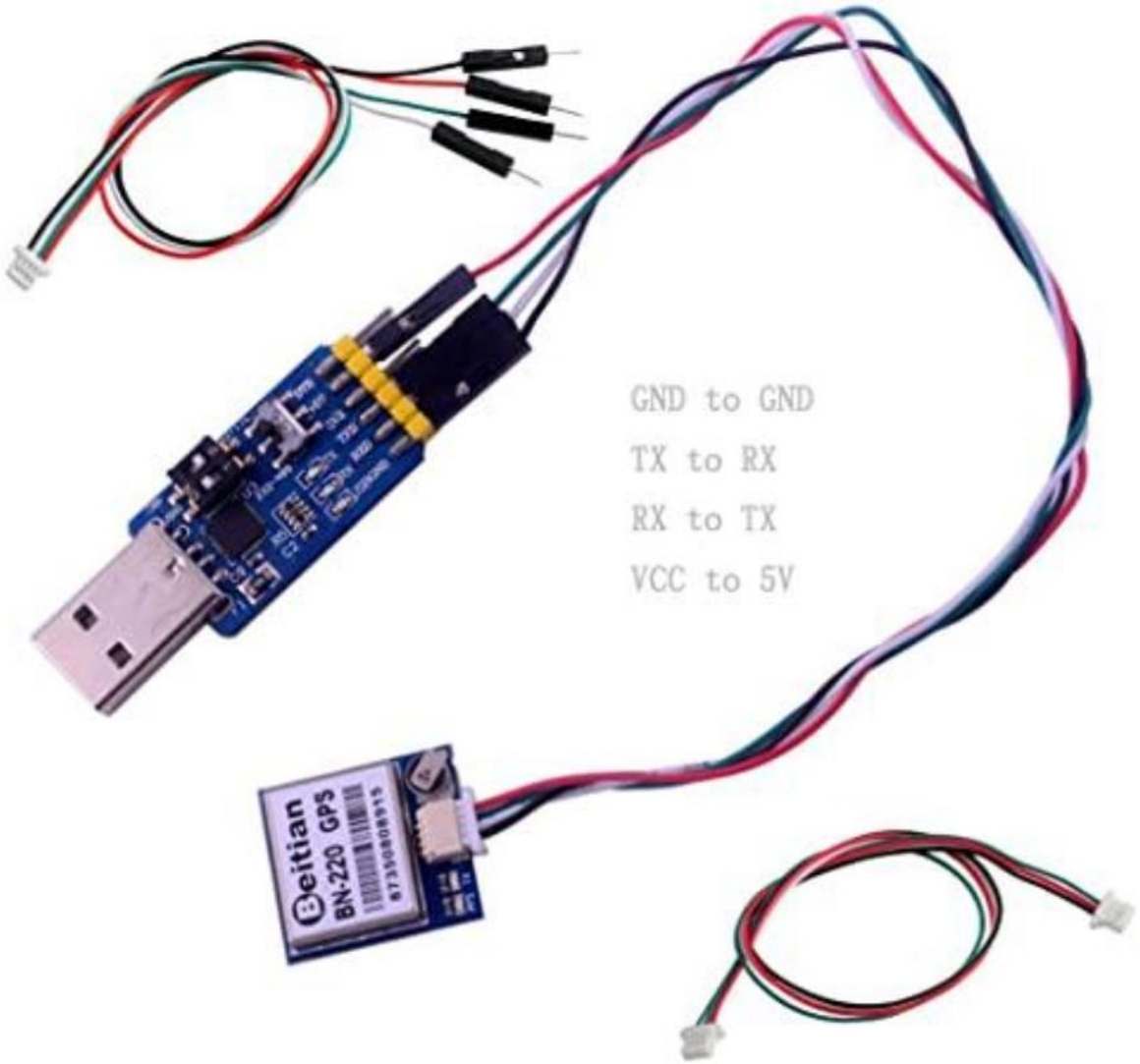
PIN	PIN Name	I/O	Description
1	GND	G	Ground
2	TX	O	Serial Data Output.
3	RX	I	Serial Data Input.
4	VCC	I	DC 3.0V - 5.5V supply input,Typical: 5.0V

Pin Description:

PIN	PIN Name	I/O	Description
1	SDA	O	Compass SDA
2	GND	G	Ground
3	TX	O	Serial Data Output.
4	RX	I	Serial Data input.
5	VCC	I	3.0V~ 5.5V supply input,Typical: 5.0V
6	SCL	I	Compass SCL

GPS ONLY MODELS & GPS WITH COMPASS MODEL

GPS CONFIGURING



BEITIAN UBLOX

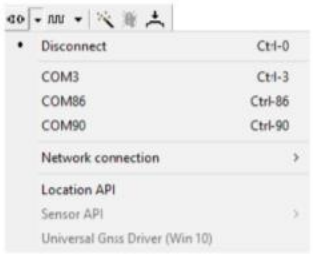
PLUG IN TO SERIAL TX 2 RX 2

USB TTL TO PROGRAM THE GPS

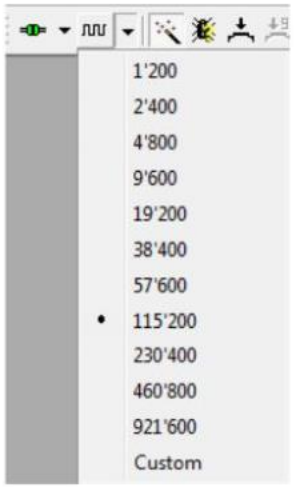
THIS GOES SAME ON THE DRONE SHIELD

U CENTER

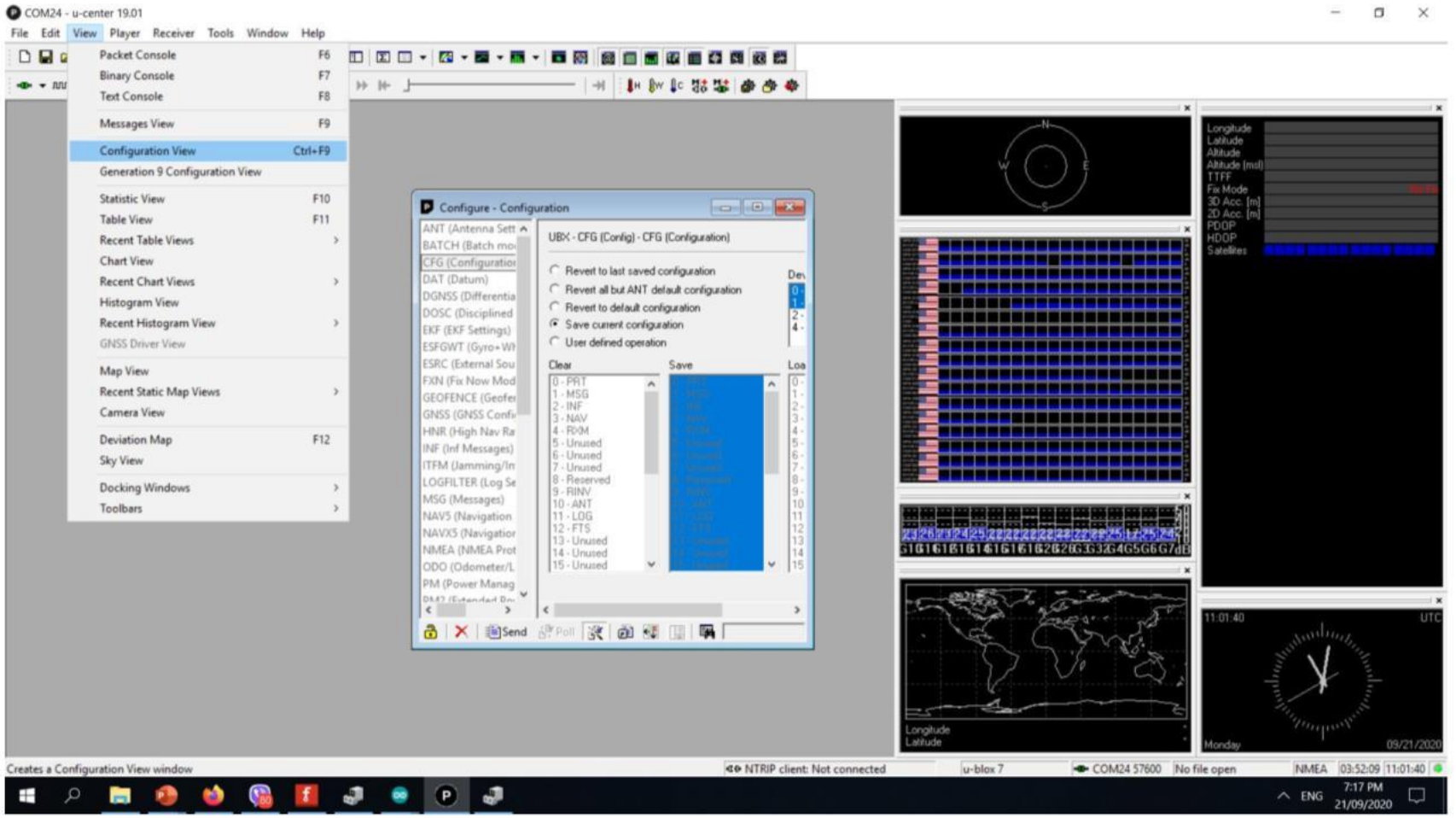
Connect to the device. Select the COM port your GPS and USBTTL is connected to



Connect to the baud rate your GPS is set to Default (9600) Setup we wanted (57600)

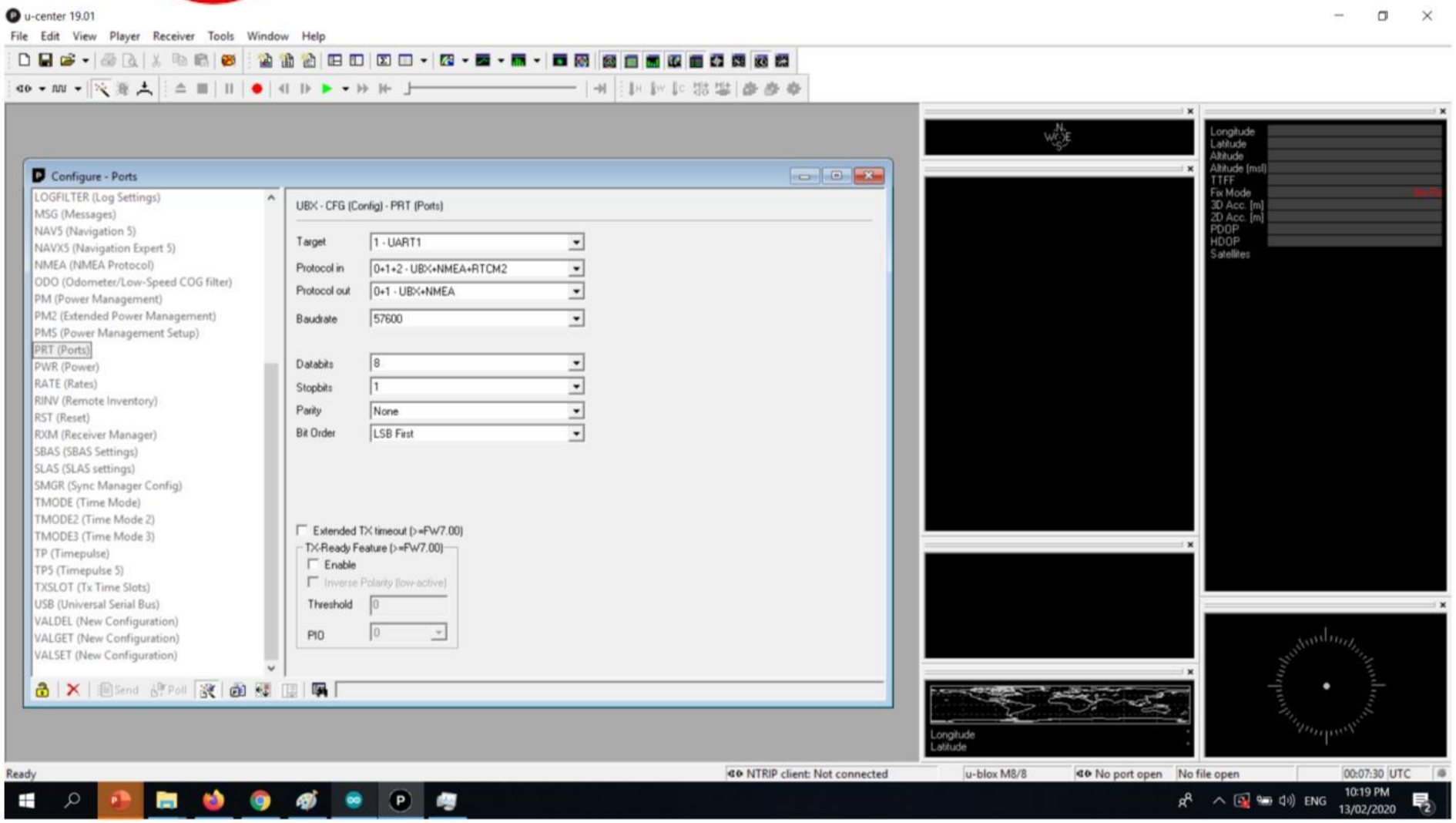


Go to View> Configuration View



U CENTER

1. Connect to the device.
2. Open View / Messages View
3. Select UBX-CFG-PRT.
4. Poll the current configuration from the receiver Change the setting to the desired baud rate. (57600)
5. Click the lock icon on the lower left and Send the message to the receiver

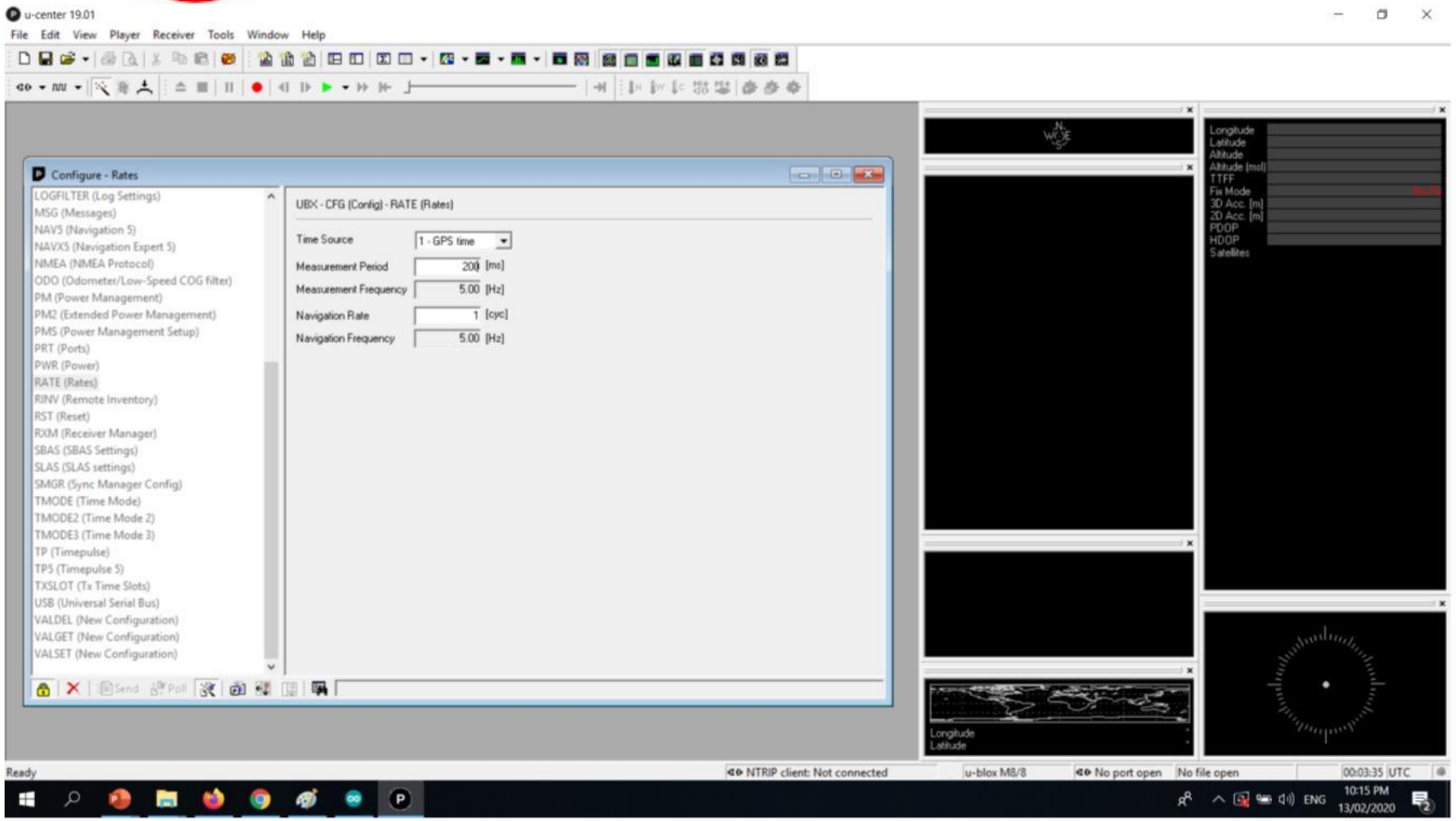


U CENTER

Rate 200ms
Frequency 5hz

Note after updating the settings click o the Lock icon on the left and click the send button

Exit the configuration window this will promp you to the save parameters . Hit “Yes”



U CENTER

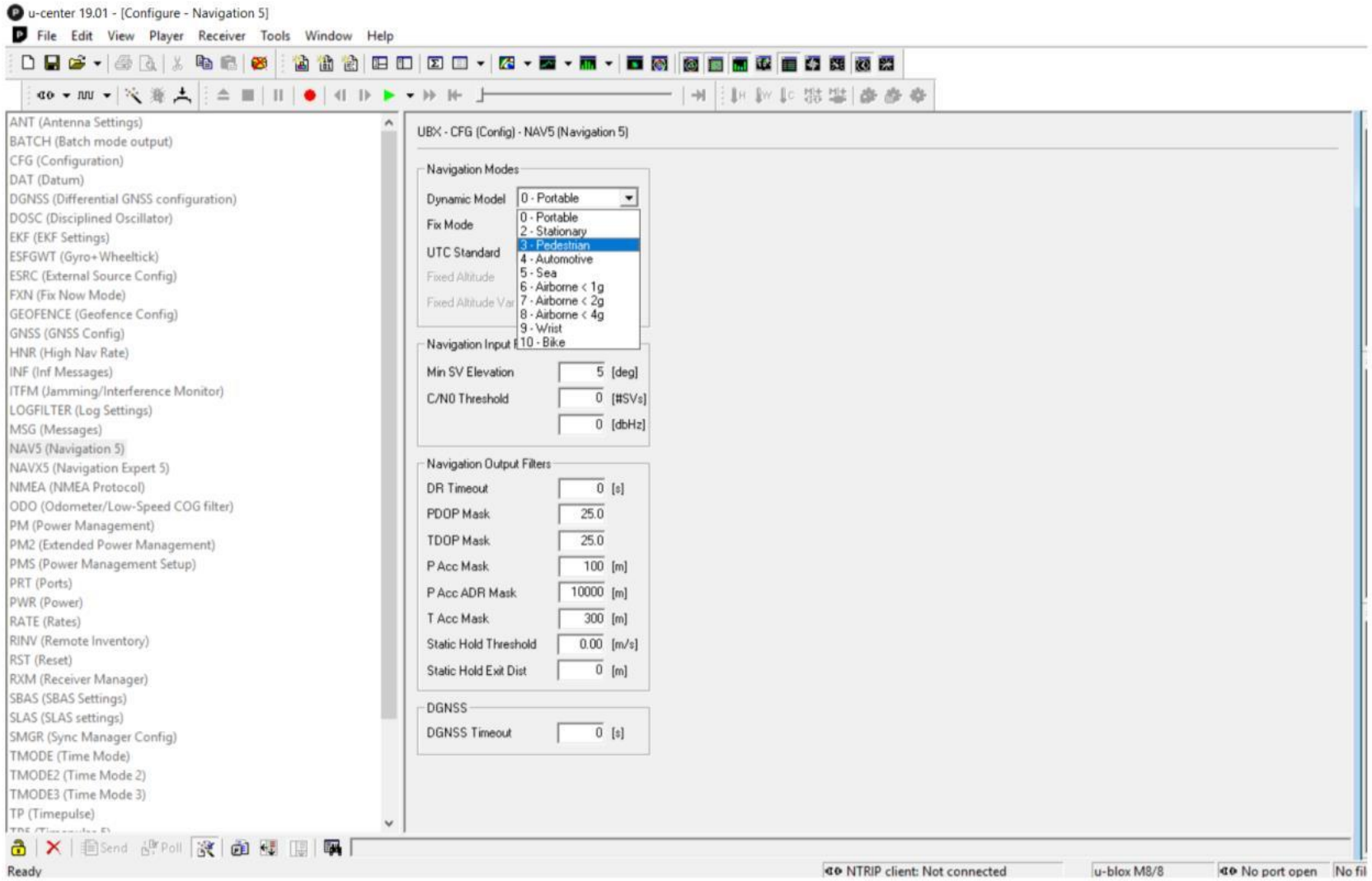
Nav5 (Navigation 5)

Dynamic Model

3. Pedestrian – better position hold

6-8 Airborne – ideal for Navigation , this reduces filtering

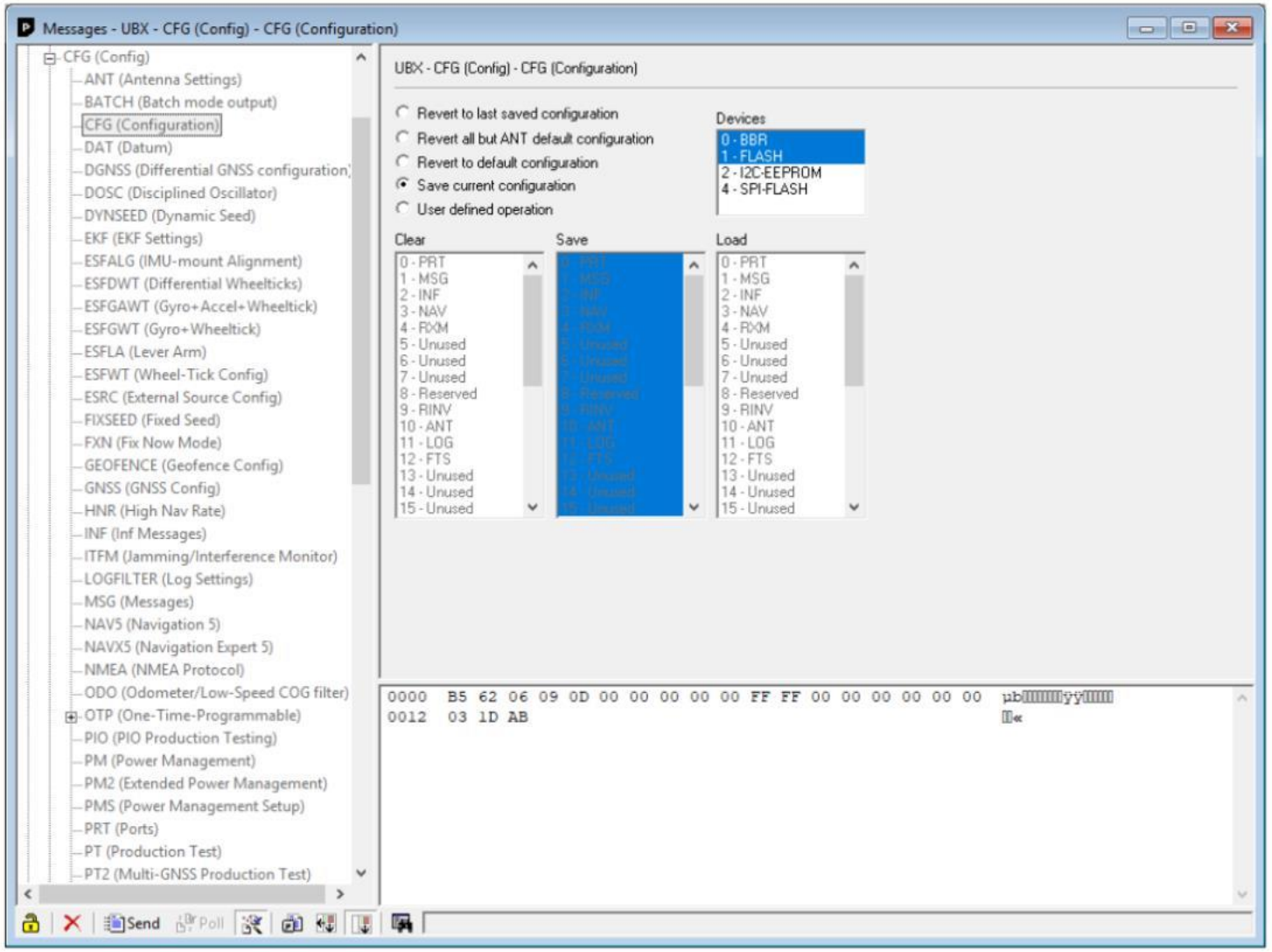
Hit Send icon to save



U CENTER

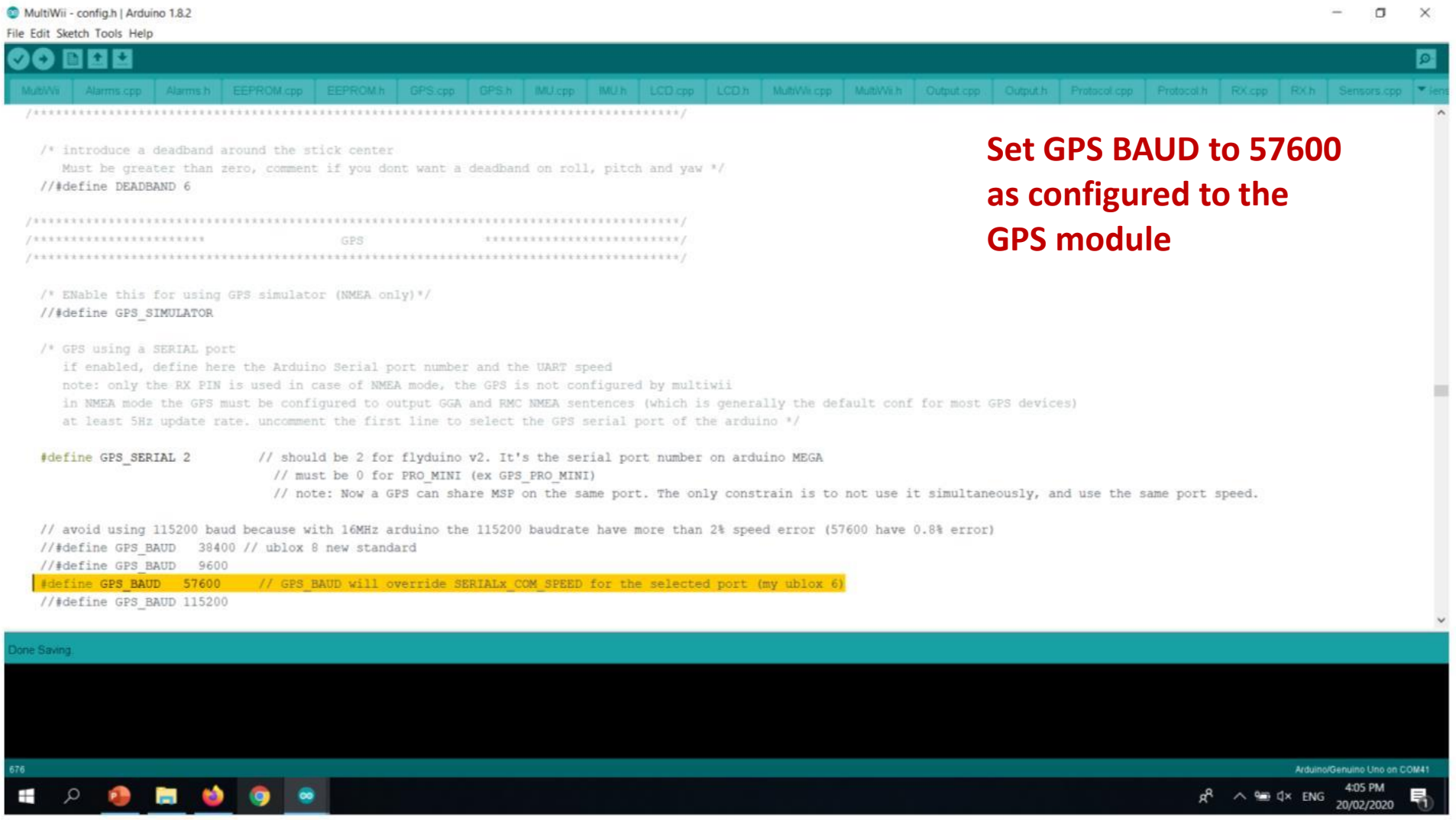
1. Connect to the device.
2. Open View / Messages View
3. Select UBX-CFG-CFG.
4. Select “save current configuration”
5. Send the message to the receiver hit the Lock icon then send icon in the bottom

Disconnect / Reconnect to the GPS again using the 57600 baud to see if the parameters are save



GPS

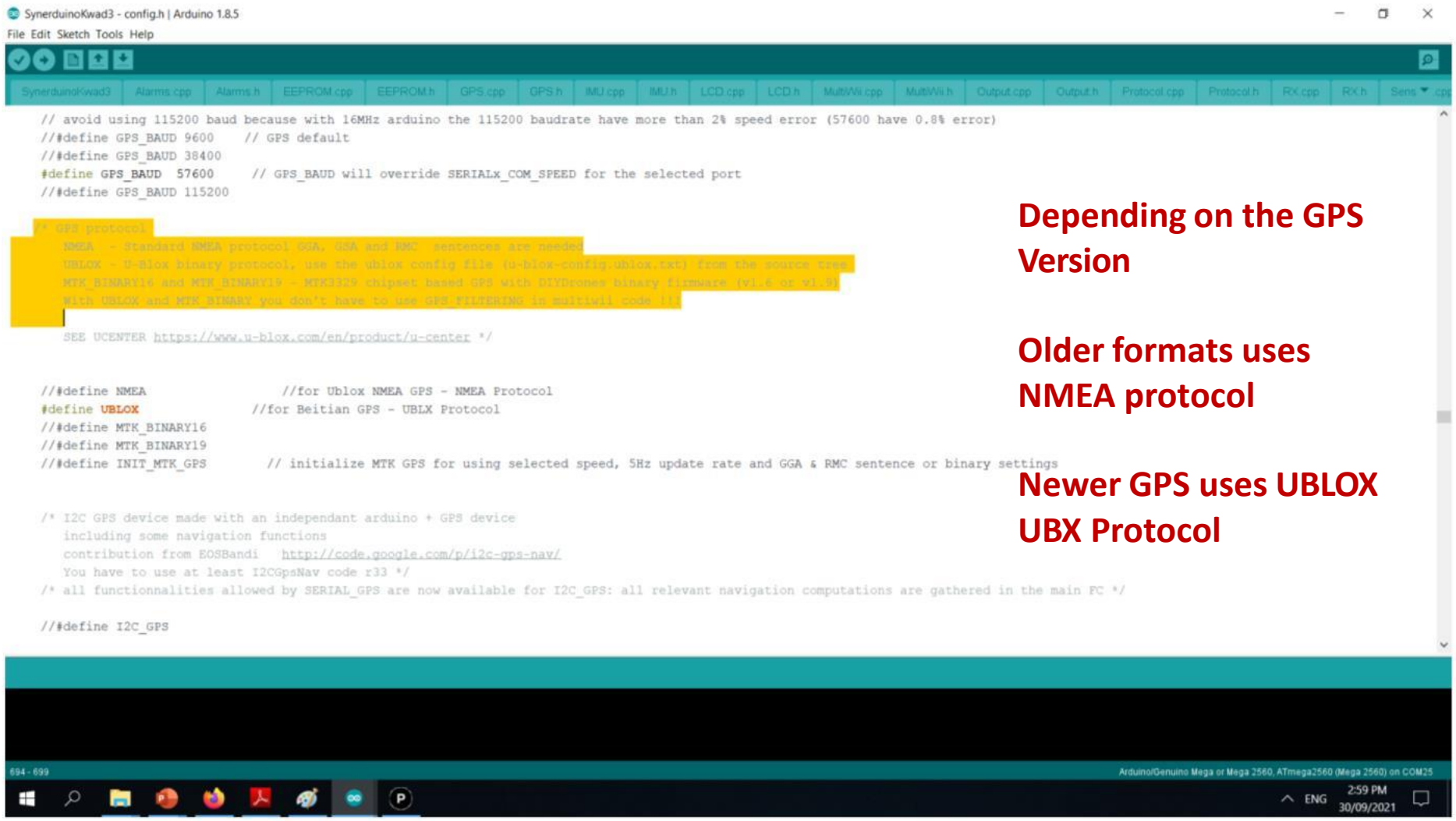
CONFIG.H



Set GPS BAUD to 57600 as configured to the GPS module

GPS

CONFIG.H

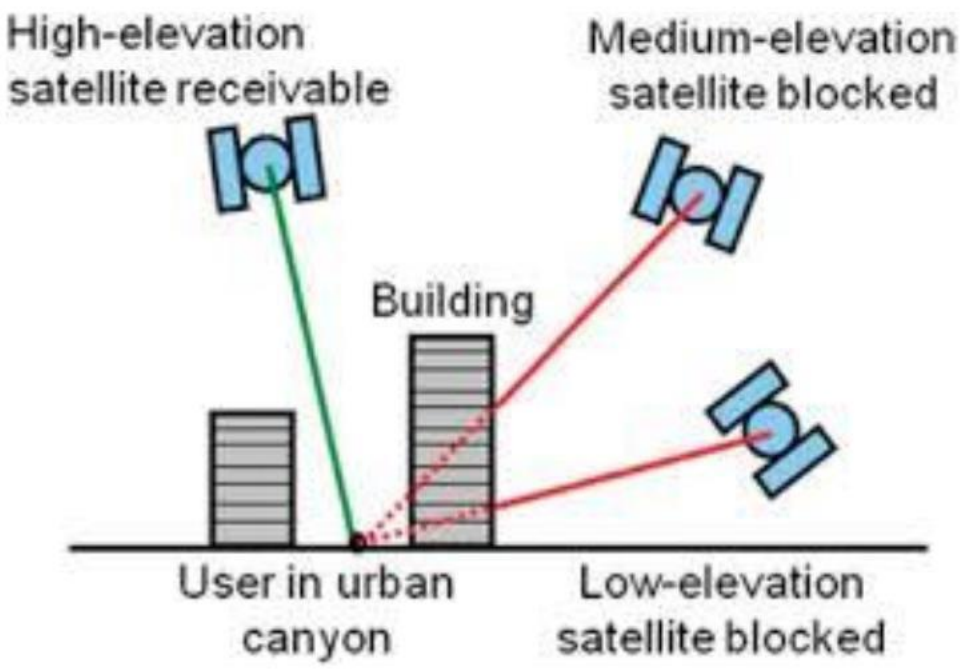


Depending on the GPS Version

Older formats uses NMEA protocol

Newer GPS uses UBLOX UBX Protocol

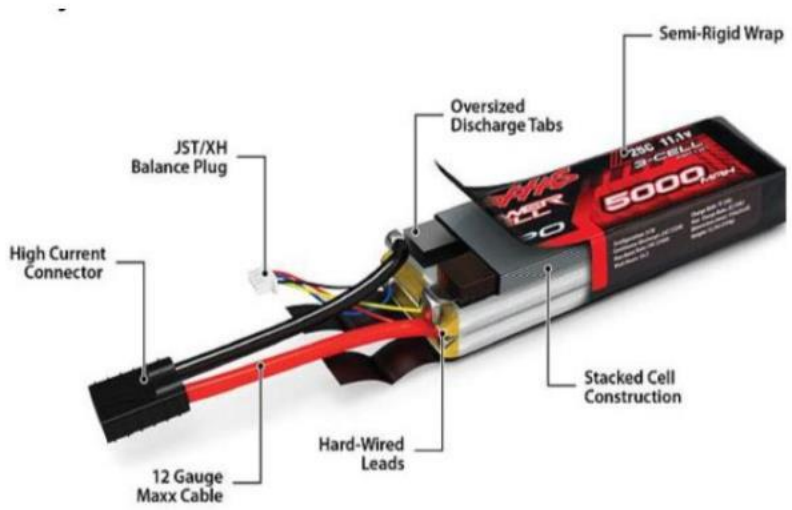
GPS



Note : GPS require a clear open area to get a proper fix and accuracy minimum 7 satellites but 10+ are Ideal

Flying next to a building can distort satellite signal deteriorating accuracy

Which in this case its better to not use GPS modes and fly Manual



BATTERY

BATTERY CARE – ONLY CHARGE AT 1C OR THE RECOMMENDED THE CHARGE RATE ON THE LABEL

USE BALANCE CHARGER TO SET THE CURRENT IN THE SAMPLE IS 5A

BATTERY STORAGE MODE IS 3.8V PER CELL

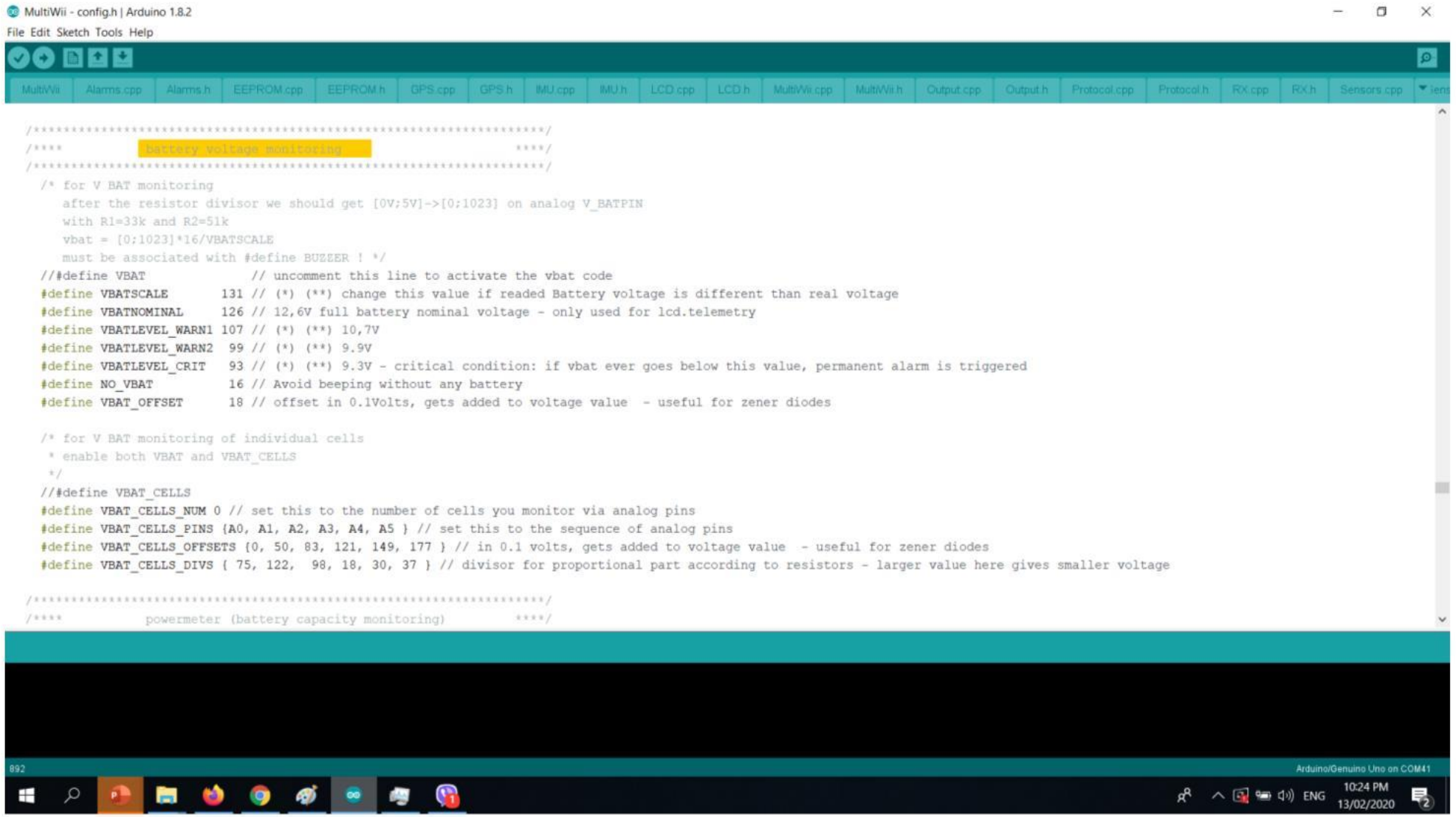
BATTERY DISCHARGE IS 3.6V **PER CELL**

BATTERY FULL CHARGE IS 4.2V **PER CELL**

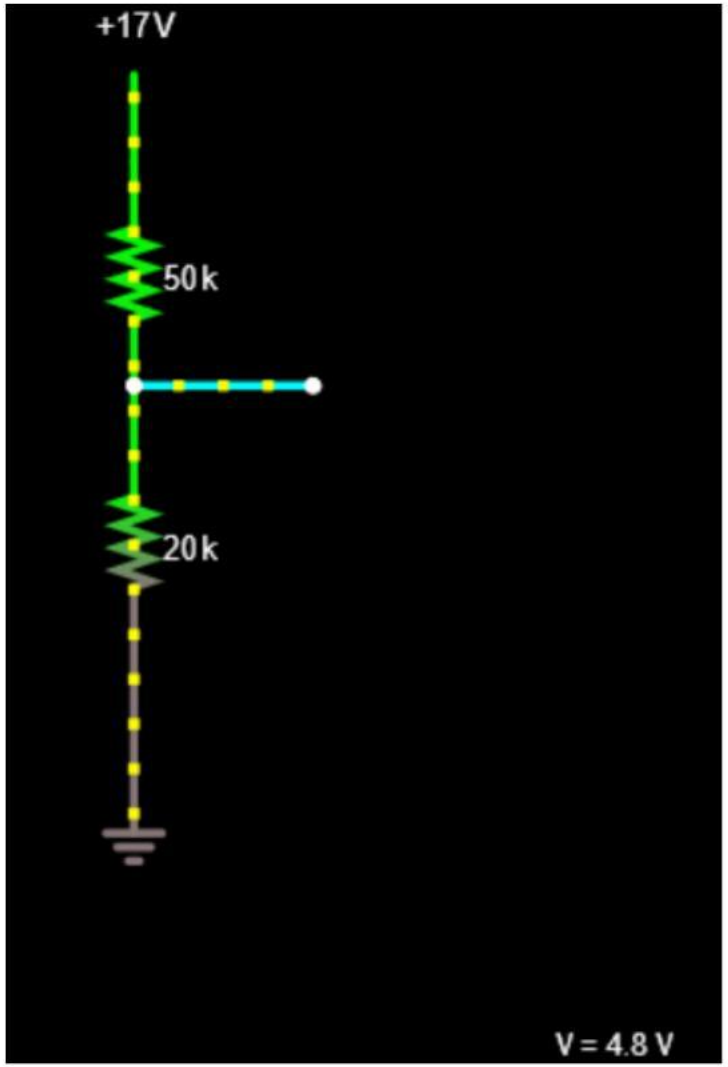
ITS RECOMMEND TO USE THE VOLTAGE ALARM TO MONITOR THE BATTERY VOLTAGE WHILE IN USE

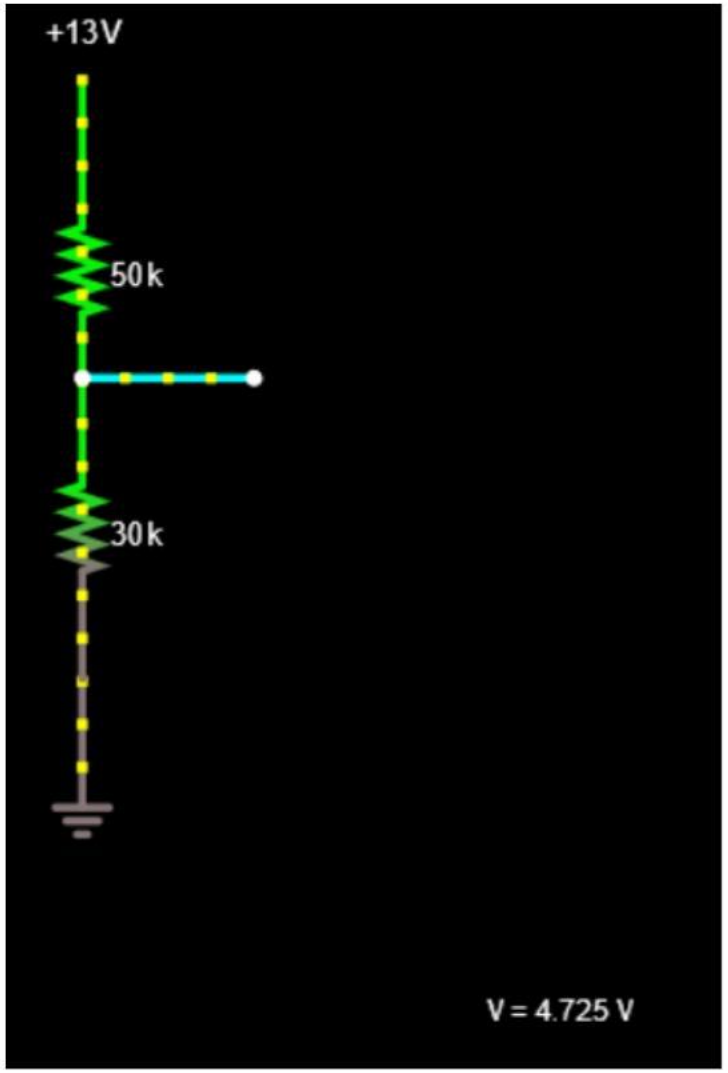
BATTERY MONITORING

CONFIG.H



4S 16.8V 3S 12.6V





VOLTAGE READING

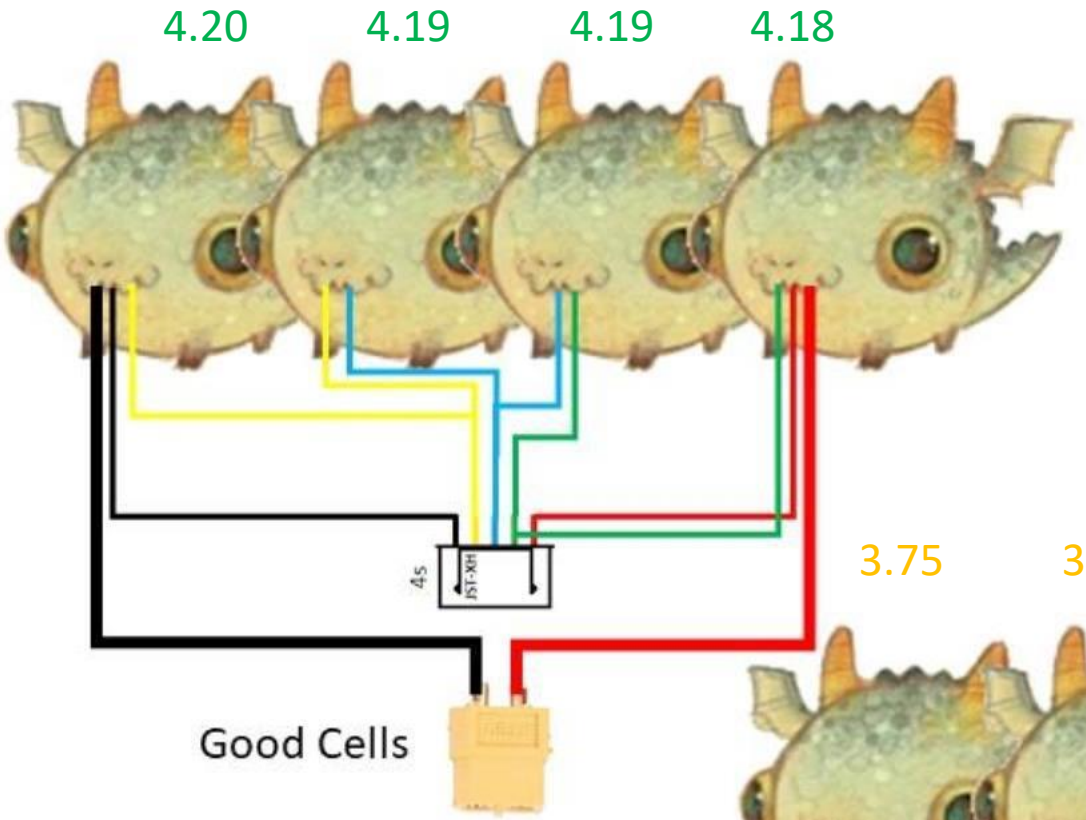
HOW MUCH POWER IN YOUR BATTERY

VOLTAGE DIVIDER

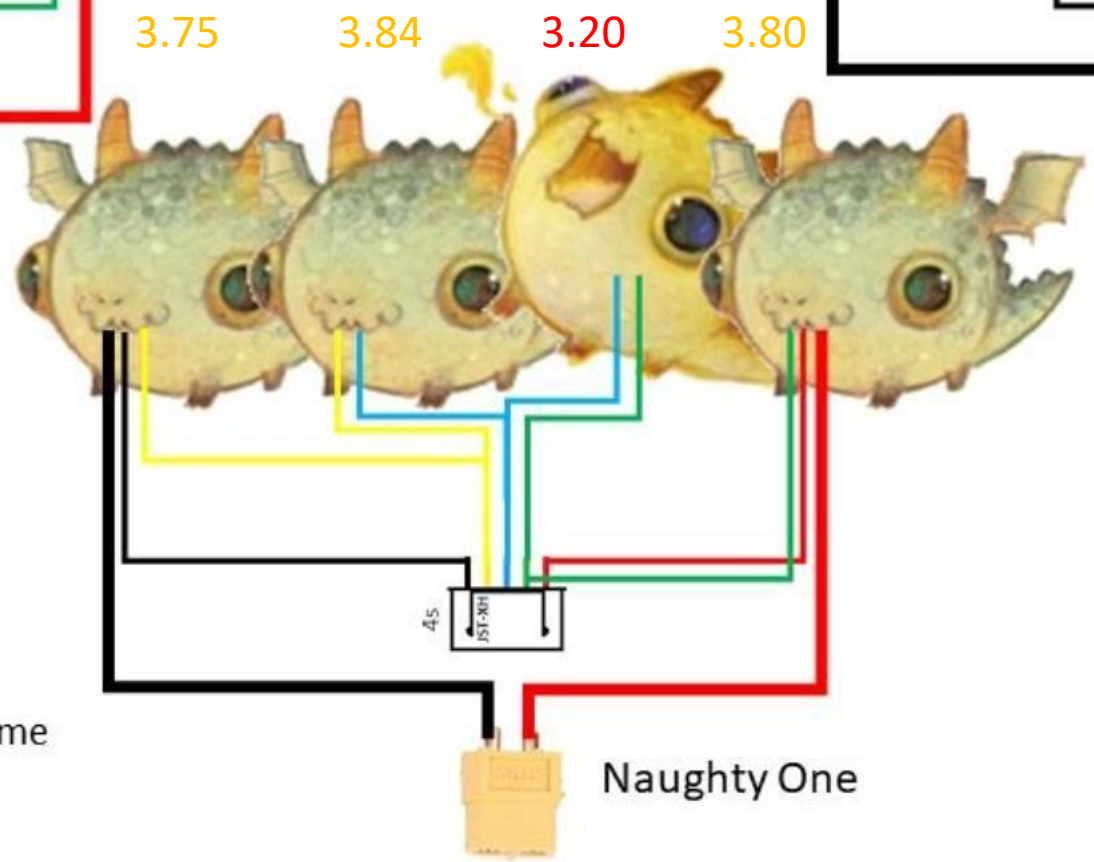
THIS ALLOWS THE 3V-5V TO BE INPUTTED TO THE A0 ANALOG PIN OF THE ARDUINO TO READ THE BATTERY VOLTAGE

SWITCH THE VBAT JUMPER ACCORDINGLY TO THE BATTER CELLS YOUR USING

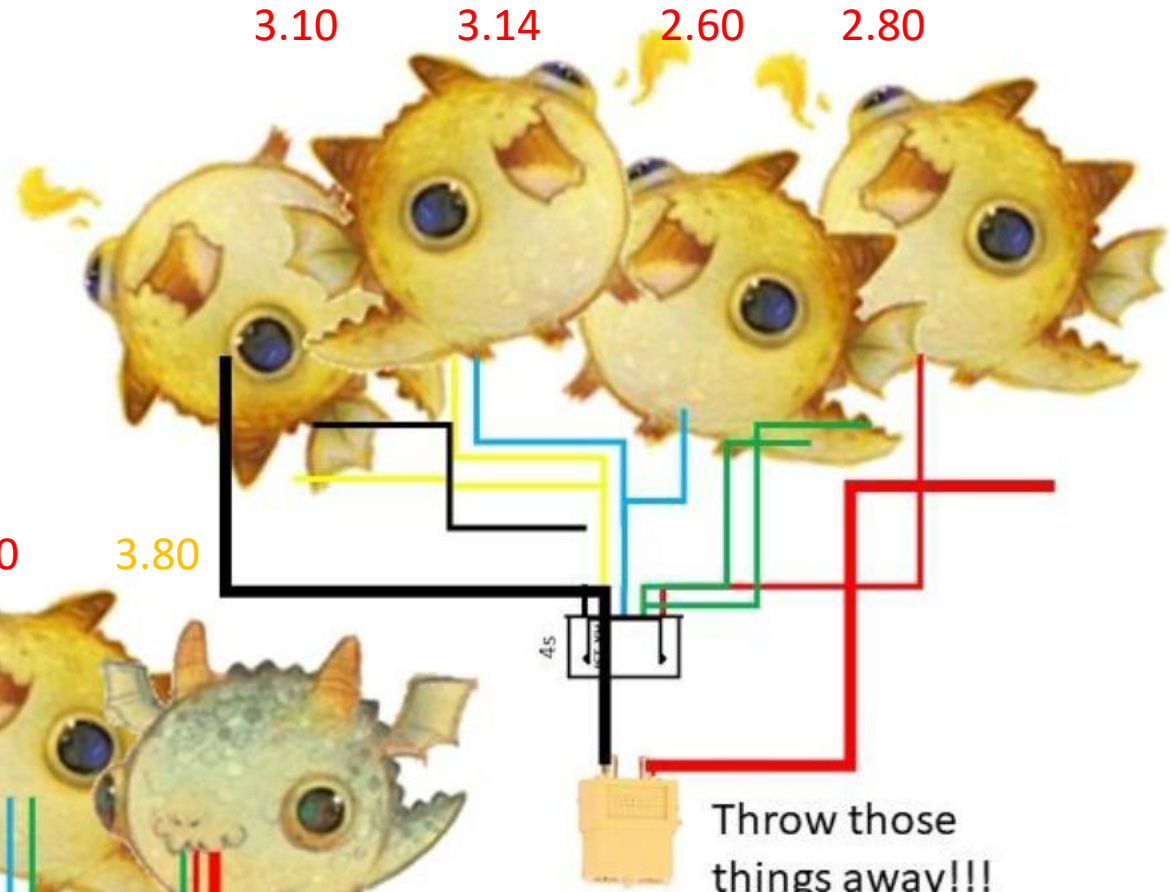
3S OR 4S



Good Cells



Naughty One



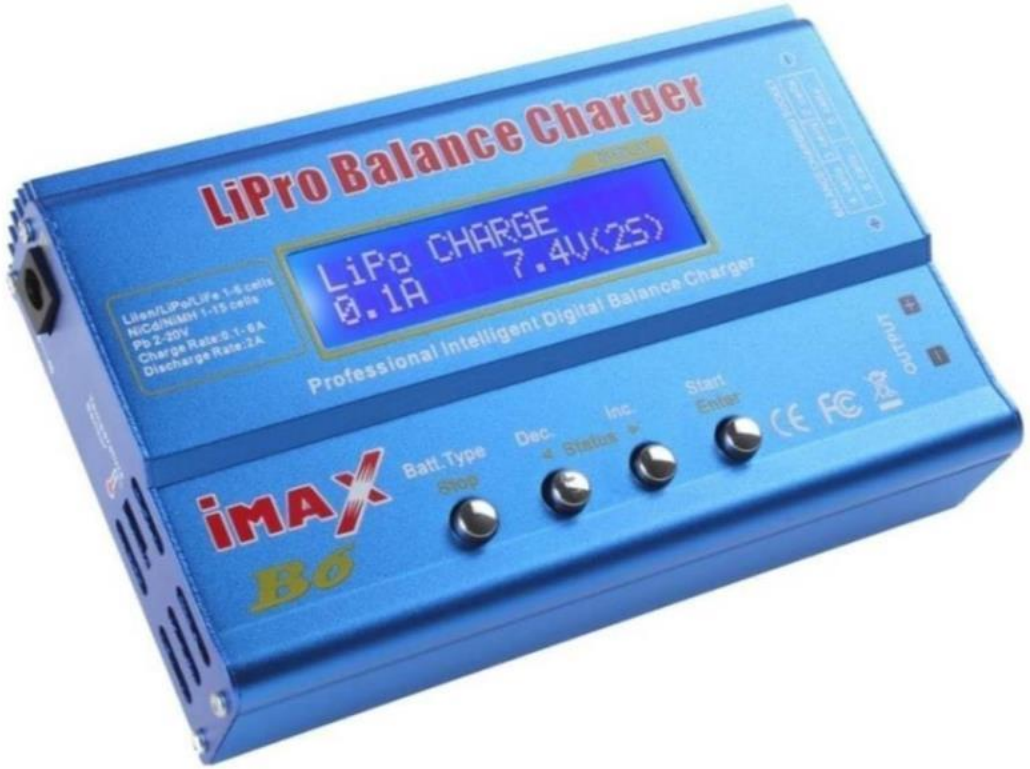
Throw those things away!!!

I always tell newbies to pretend they were Orb Dragons

Treat them right and they will last a long time

Abuse them and your asking for trouble

BATTERY CHARGER



Standard Lipo Balance Charger
Nimh/Nicad/Lipo/Pb Mode 1s to 6s 5A

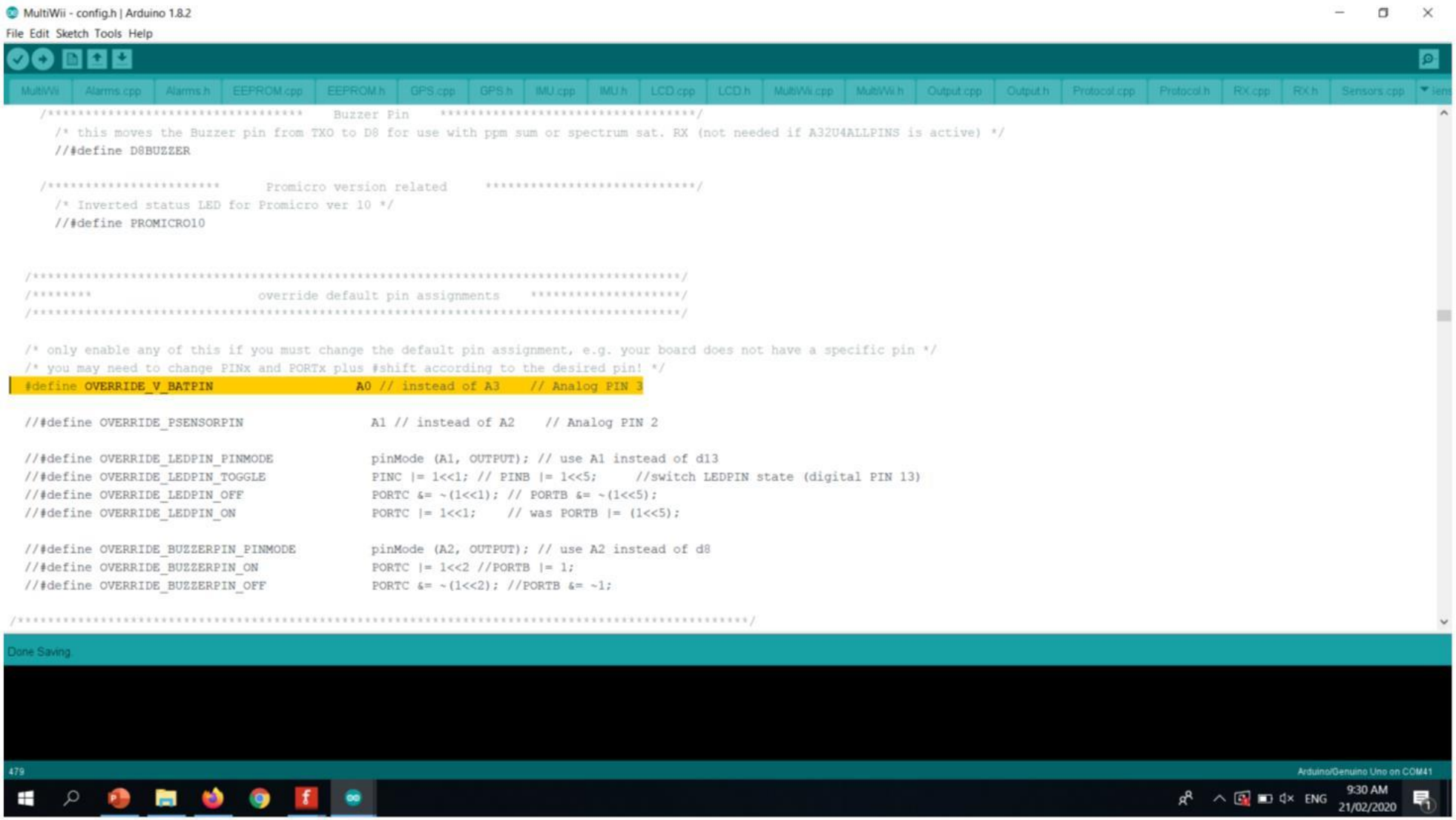
The Advantage of this is the ability to monitor your charge as well as resurrecting batteries that were over drain by 1V from drain

Cheap Balance Charger but no way users can monitor their battery condition

May post risk as you cannot monitor or emergency charge your battery

PIN ASSIGNMENTS

CONFIG.H



CAMERA STABILIZATION

CONFIG.H

Photo for illustration purpose only.

MultiWii - config.h | Arduino 1.8.2

File Edit Sketch Tools Help

MultiWii | Alarms.cpp | Alarms.h | EEPROM.cpp | EEPROM.h | GPS.cpp | GPS.h | IMU.cpp | IMU.h | LCD.cpp | LCD.h | MultiWii.cpp | MultiWii.h | Output.cpp | Output.h | Protocol.cpp | Protocol.h | RX.cpp | RX.h | Sensors.cpp

```
 */
//#define DISABLE_SERVOS_WHEN_UNARMED


/* if you want to preset min/middle/max values for servos right after flashing, because of limited physical
 * room for servo travel, then you must enable and set all three following options */
 //#define SERVO_MIN  {1020, 1020, 1020, 1020, 1020, 1020, 1020, 1020}
 //#define  SERVO_MAX {2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000}
 //#define  SERVO_MID {1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500} // (*)
 //#define FORCE_SERVO_RATES      {30,30,100,100,100,100,100,100} // 0 = normal, 1= reverse

/************************          Cam Stabilisation          ************************/
/* The following lines apply only for a pitch/roll tilt stabilization system. Uncomment the first or second line to activate it */
#define SERVO_MIX_TILT
//#define SERVO_TILT

/* camera trigger function : activated via Rc Options in the GUI, servo output=A2 on promini */
// trigger interval can be changed via (*GUI*) or via AUX channel
#define CAMTRIG
#define CAM_TIME_HIGH 1000   // the duration of HIGH state servo expressed in ms

/************************          Airplane          ************************/
//#define USE_THROTTLESERVO // For use of standard 50Hz servo on throttle.

//#define FLAPPERONS    AUX4          // Mix Flaps with Aileroins.
#define FLAPPERON_EP   { 1500, 1700 } // Endpooints for flaps on a 2 way switch else set {1020,2000} and program in radio.
#define FLAPPERON_INVERT { -1, 1 }    // Change direction om flapperons { Wing1, Wing2 }

//#define FLAPS                       // Traditional Flaps on SERVO3.
```

259

Generic ESP8266 Module, 80 MHz, Flash, ck, 26 MHz, 40MHz, QIO, 512K (no SPIFFS), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM24

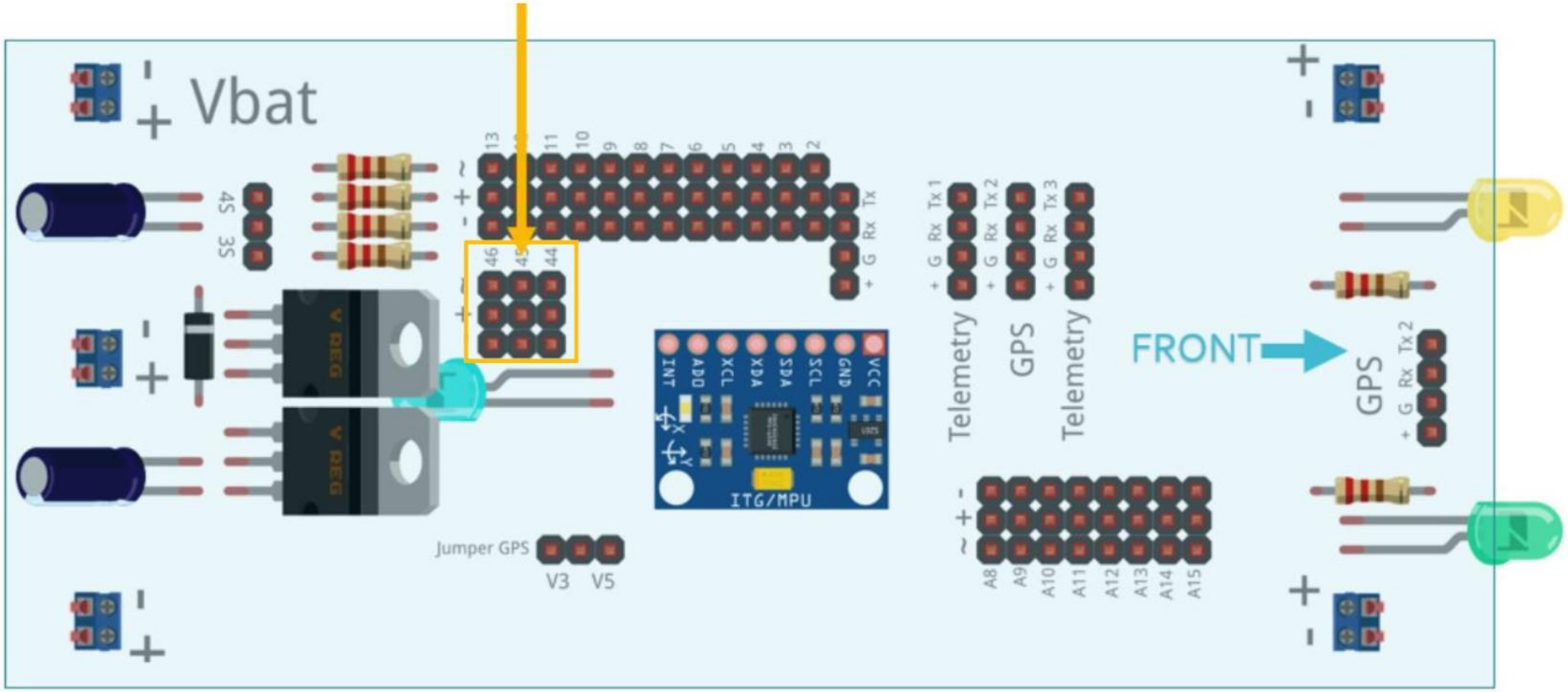
ENG 6:16 PM 11/08/2020

#define SERVO_MIX_TILT //D44 and D45 (120 Mix)

#define SERVO_TILT // D44 (Pitch), D45 (Roll)

#define CAMTRIG // Activate function D46 Trigger PWM

#define CAM_TIME_HIGH 1000 // HIGH state servo expressed in ms



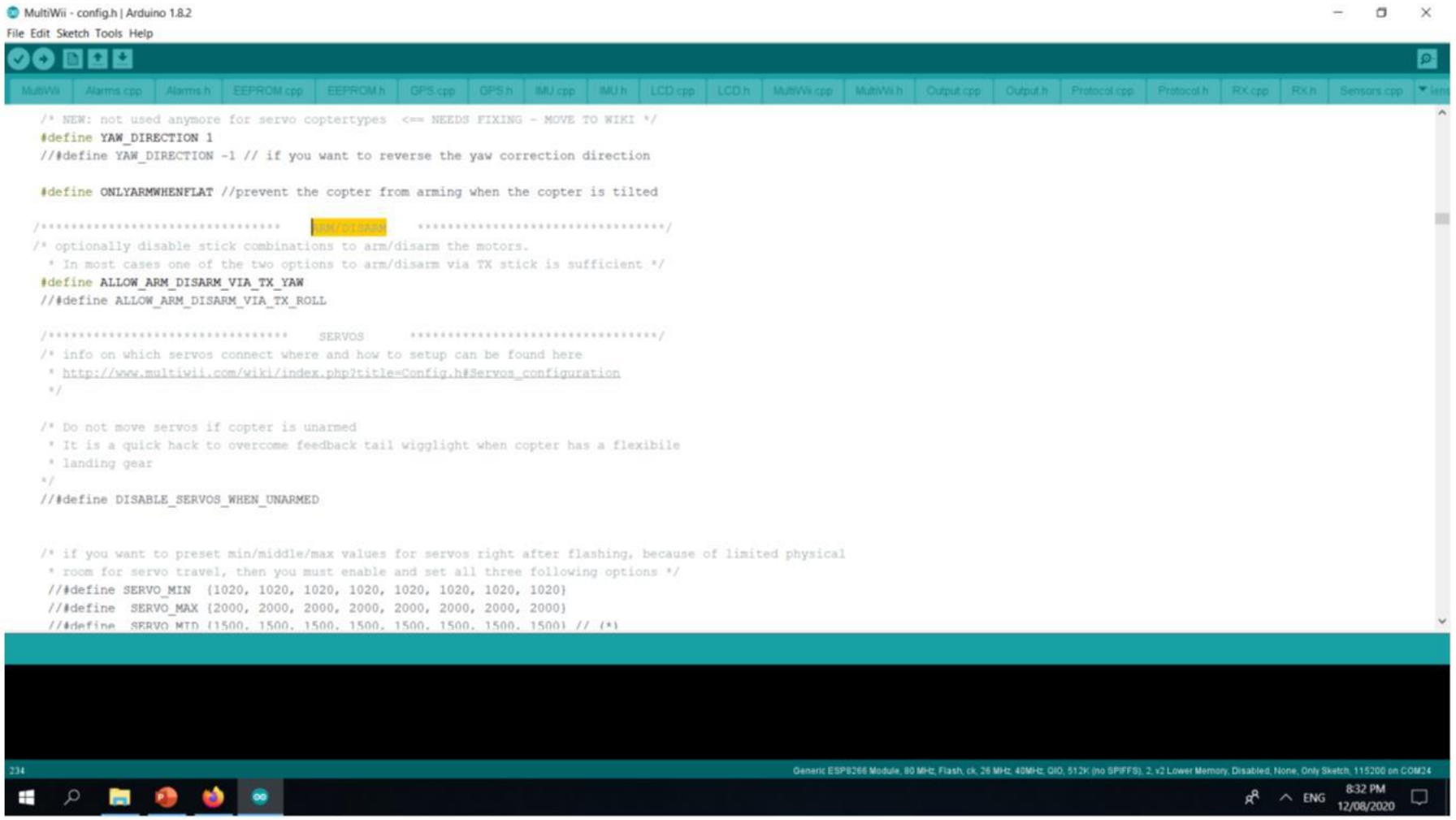
Arm/DisArm

CONFIG.H

Arm/DisArm

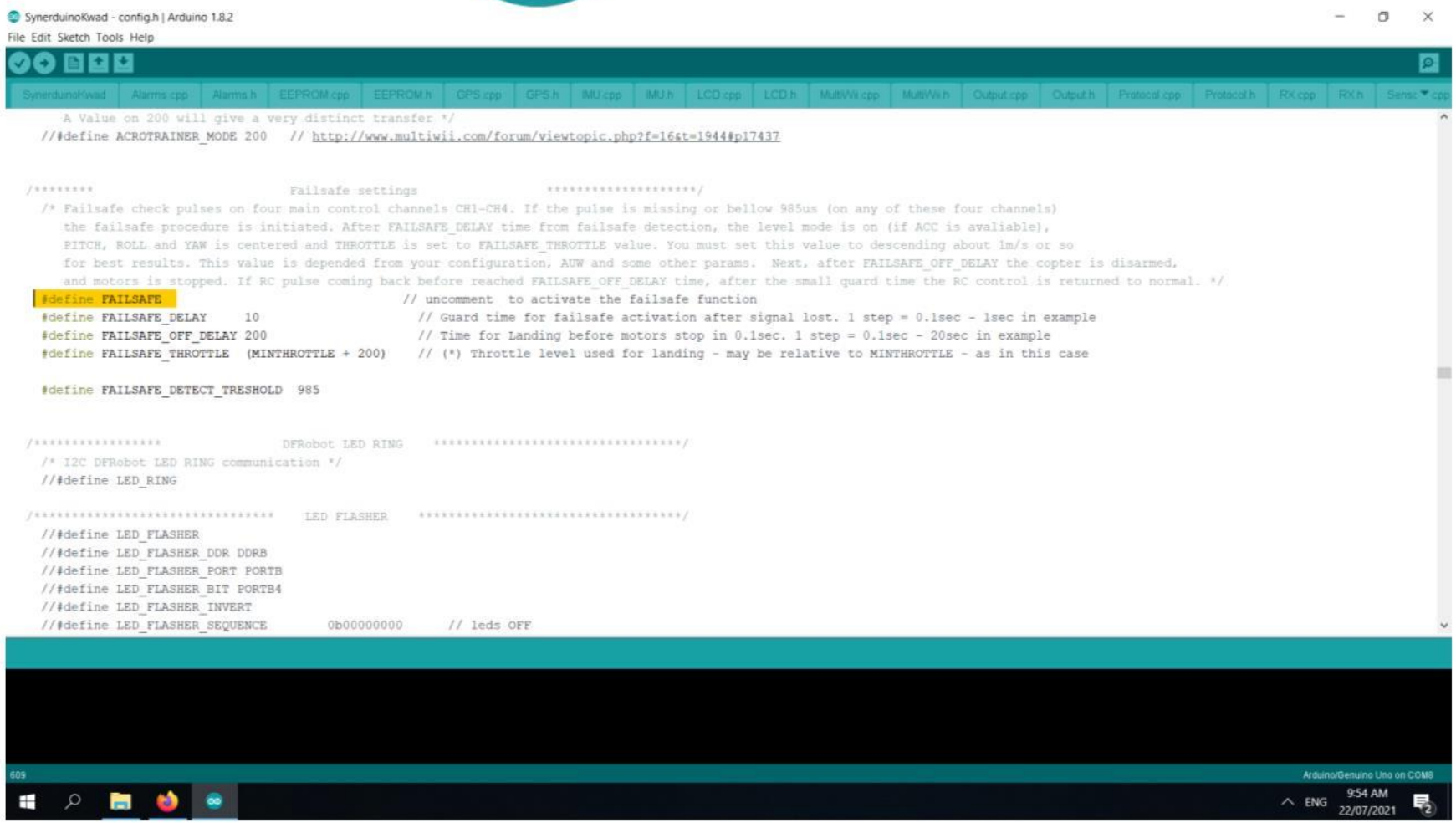
Option for combination stick command to start and stop the drone

Arm Only When Flat
Is a safety option not to arm when the drone is not level prevent starting up when on a slope or when moving



Works for Multitotors Example : Rudder Stick Left to Rudder Arm Stick Right to Disarm
Note: motors will spool up to Idle Speed

CONFIG.H



THROTTLE FAILSAFE –WHAT THE THROTTLE INPUT WOULD BE IF SIGNAL IS LOST BETWEEN THE SYNERDUINO BOARD , RECEIVER AND TRANSMITTER

FLYWIIGUIGROUND STATION

Download the FlyWiiGUI groundstation and open FlywiiGUI.exe

Name	Date modified	Type	Size
210130-0301	30/04/2021 3:01 PM	File	3 KB
210814-0408	14/09/2021 4:08 PM	File	3 KB
212812-0428	12/06/2021 4:28 PM	File	3 KB
214012-0340	12/06/2021 3:40 PM	File	3 KB
AForge.Controls.dll	25/01/2015 1:15 PM	Application extens...	44 KB
AForge.dll	25/01/2015 1:15 PM	Application extens...	17 KB
AForge.Imaging.dll	25/01/2015 1:15 PM	Application extens...	248 KB
AForge.Math.dll	25/01/2015 1:15 PM	Application extens...	67 KB
AForge.Video.DirectShow.dll	25/01/2015 1:15 PM	Application extens...	52 KB
AForge.Video.dll	25/01/2015 1:15 PM	Application extens...	19 KB
AForge.Video.FFMPEG.dll	25/01/2015 1:15 PM	Application extens...	60 KB
avcodec-53.dll	25/01/2015 1:15 PM	Application extens...	13,181 KB
avdevice-53.dll	25/01/2015 1:15 PM	Application extens...	342 KB
avfilter-2.dll	25/01/2015 1:15 PM	Application extens...	870 KB
avformat-53.dll	25/01/2015 1:15 PM	Application extens...	2,405 KB
avutil-51.dll	25/01/2015 1:15 PM	Application extens...	135 KB
FlyWiiGUI.exe	30/10/2021 11:41 ...	Application	6,945 KB
FlyWiiGUI.exe.config	28/02/2017 5:31 PM	CONFIG File	1 KB
FlyWiiGUI.exe.manifest	30/10/2021 11:41 ...	MANIFEST File	30 KB

The FlyWii GUI is a free updated version of the MultiWii WinGUI. It serves as the ground control station for the MultiWii 2.4 controller software.

FlyWii GUI is currently only supported for Windows 7/8/10

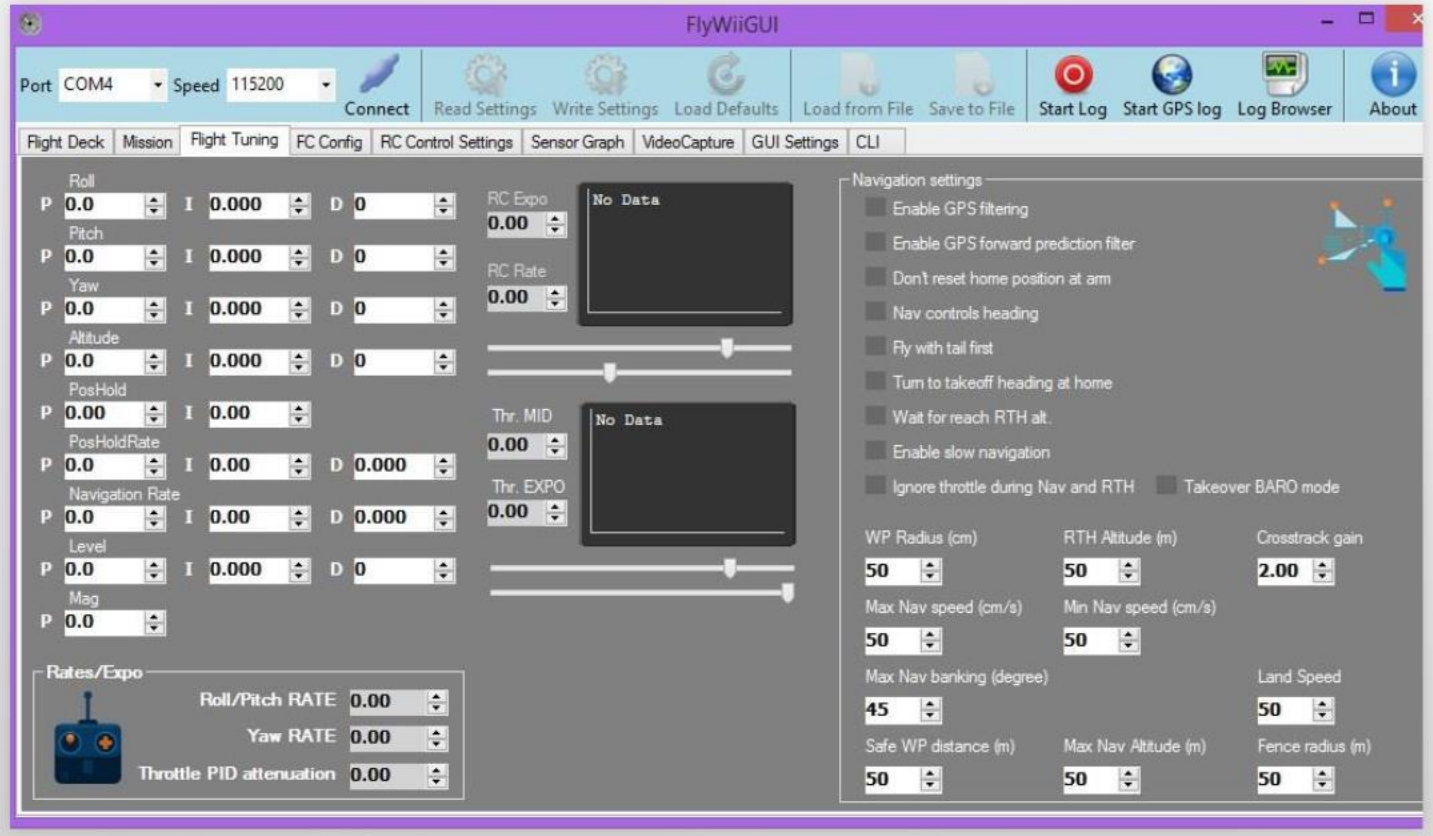
Download

Latest Release

FlwiiGUI Ground Station Software .EXE*

FlyWiiGUI20 Download

FLYWII GUI



DEDICATED USER INTERFACE AS CONFIGURATOR AND GROUND STATION FOR THE ARDUINO DRONE

FLIGHT DECK – IF THIS DOESN'T LOOK RIGHT CHECK YOUR SENSORS ORIENTATION AGAIN USING THE SENSOR GRAPH

TELEMETRY CONNECTION SEE YOUR CHECK YOUR BLUETOOTH RADIO OR USB ON WHERE IS THE VIRTUAL COM PORT IS

COMPASS (MAG)

PWM OUTPUT INDICATOR

PWM INPUT INDICATOR

CALIBRATION ACC /MAG

SAVE CONFIG

FLIGHT & GPS LOGS

ALTITUDE (BARO)

ATTITUDE (ARTIFICIAL HORIZON)
(GYRO XYZ AND ACC XYZ)

GPS SATELLITE COUNT
(4 SATS FOR 3D FIX – IDEAL 7 SATS)

VERTICAL SPEED INDICATOR
(ACC Z AXIS)

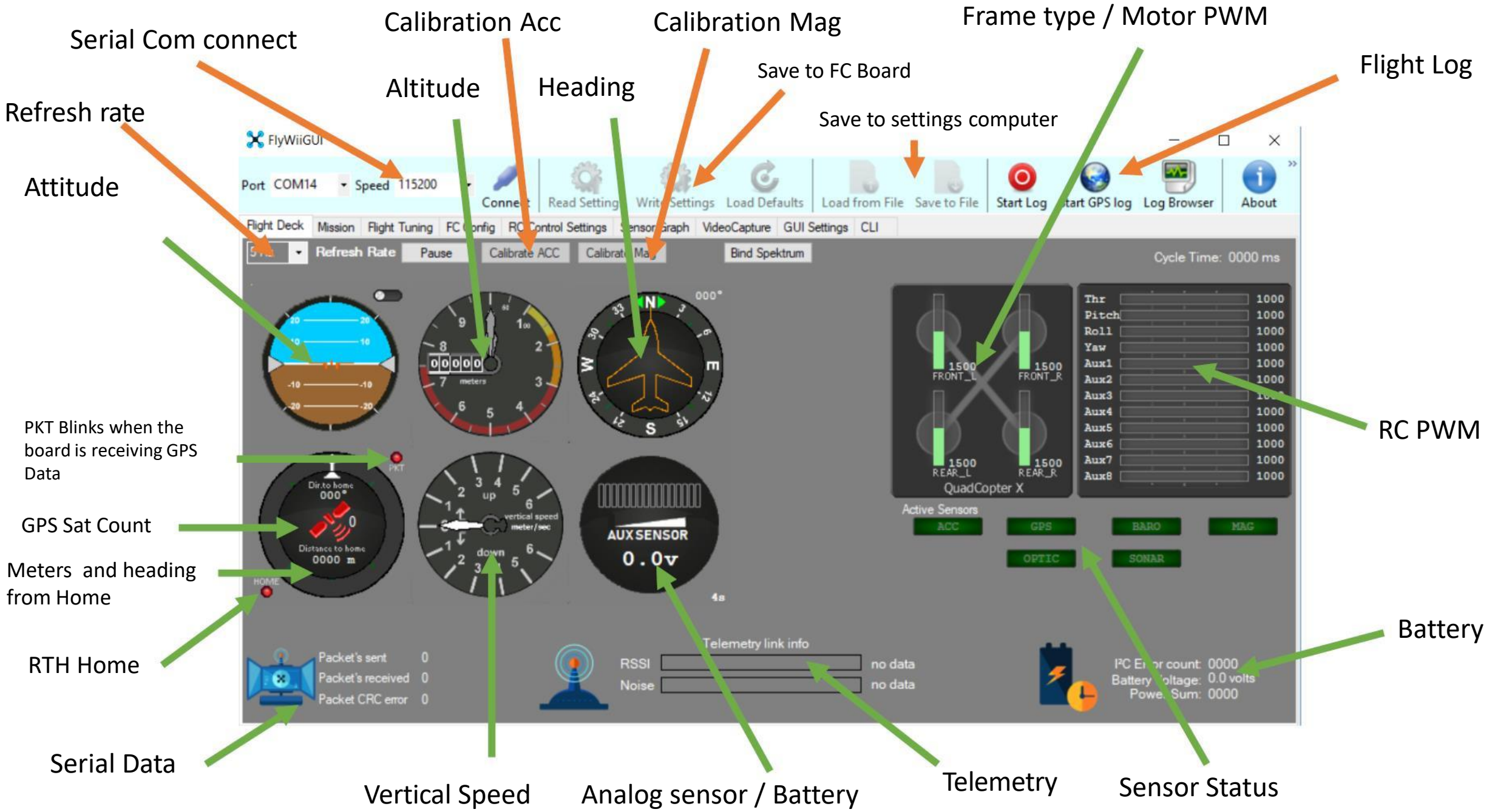
PACKETS STATUS

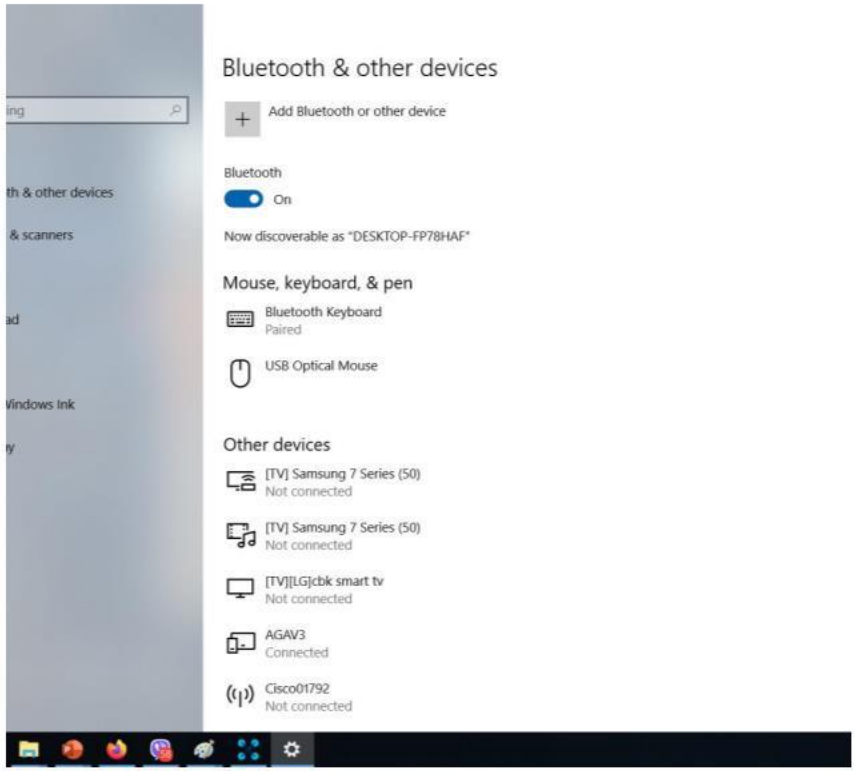
(IF THE ERROR NUMBERS ARE HIGH PLS CHECK YOUR TELEMETRY CONFIG)

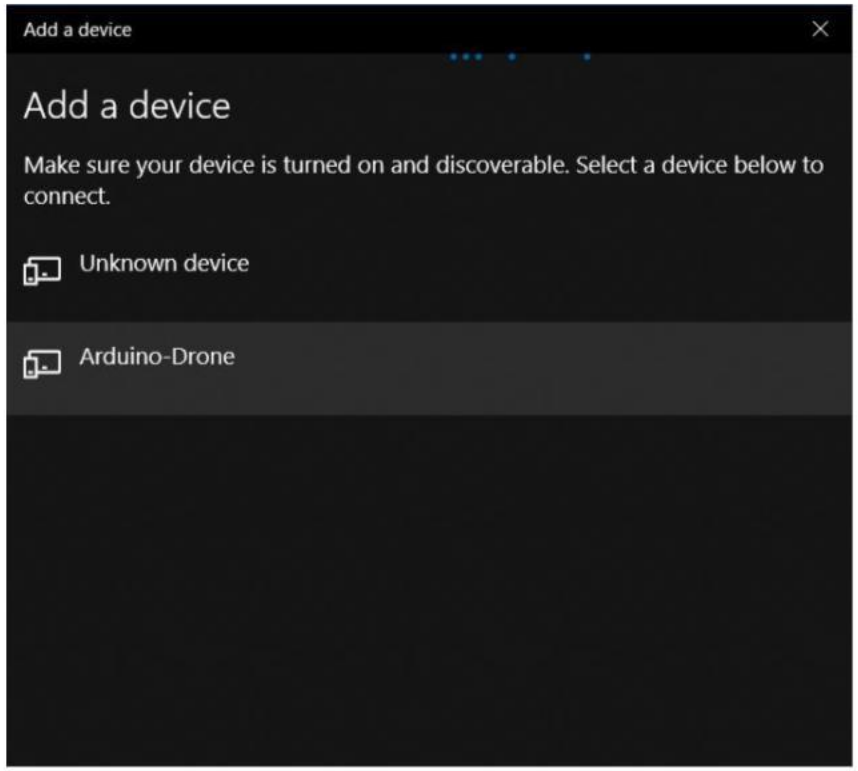
POWER STATUS / AUX SENSOR

ALSO KNOWN AS FUEL GAUGE

(VBAT)

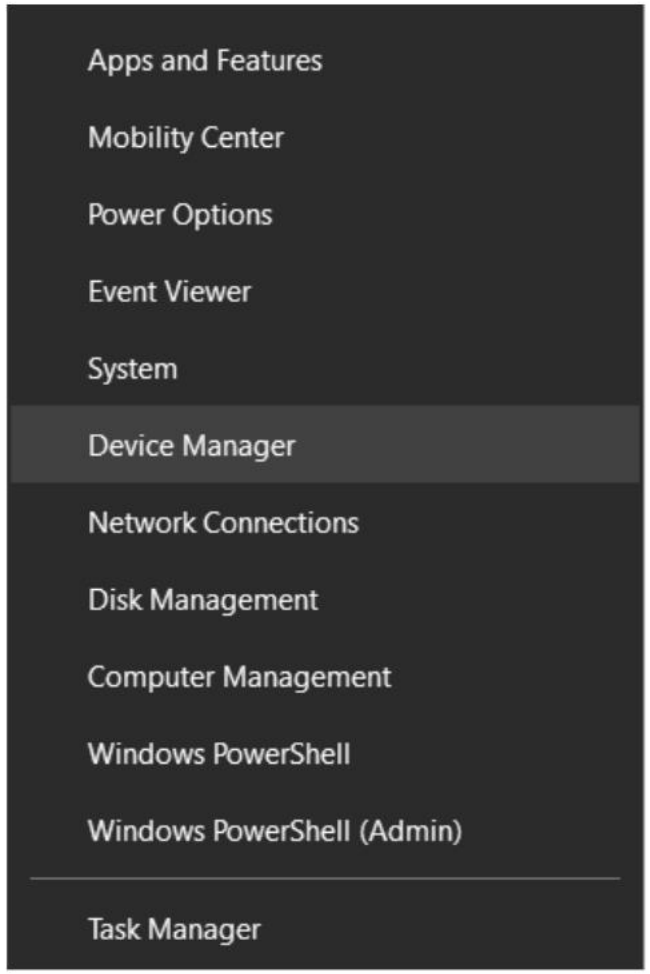

Serial Com connect
Calibration Acc
Calibration Mag
Frame type / Motor PWM
Flight Log
Altitude
Heading
Save to FC Board
Refresh rate
Save to settings computer
Attitude
FlyWiiGUI
Port COM14
Speed 115200
Connect
Read Setting
Write Settings
Load Defaults
Load from File
Save to File
Start Log
Start GPS log
Log Browser
About
Flight Deck
Mission
Flight Tuning
FC Config
RC Control Settings
Sensor Graph
VideoCapture
GUI Settings
CLI
Refresh Rate
Pause
Calibrate ACC
Calibrate Mag
Bind Spektrum
Cycle Time: 0000 ms
PKT Blinks when the board is receiving GPS Data
PKT
GPS Sat Count
Dir.to home
Distance to home
0000 m
Meters and heading from Home
HOME
RTH Home
Serial Data
Vertical Speed
up
down
vertical speed
meter/sec
AUXSENSOR
0.0v
Analog sensor / Battery
Telemetry
Sensor Status
RC PWM
Battery
FRONT_L
FRONT_R
REAR_L
REAR_R
QuadCopter X
Thr
Pitch
Roll
Yaw
Aux1
Aux2
Aux3
Aux4
Aux5
Aux6
Aux7
Aux8
Active Sensors
ACC
GPS
BARO
MAG
OPTIC
SONAR
Telemetry link info
RSSI
Noise
no data
Packet's sent
Packet's received
Packet CRC error
Battery Voltage: 0.0 volts

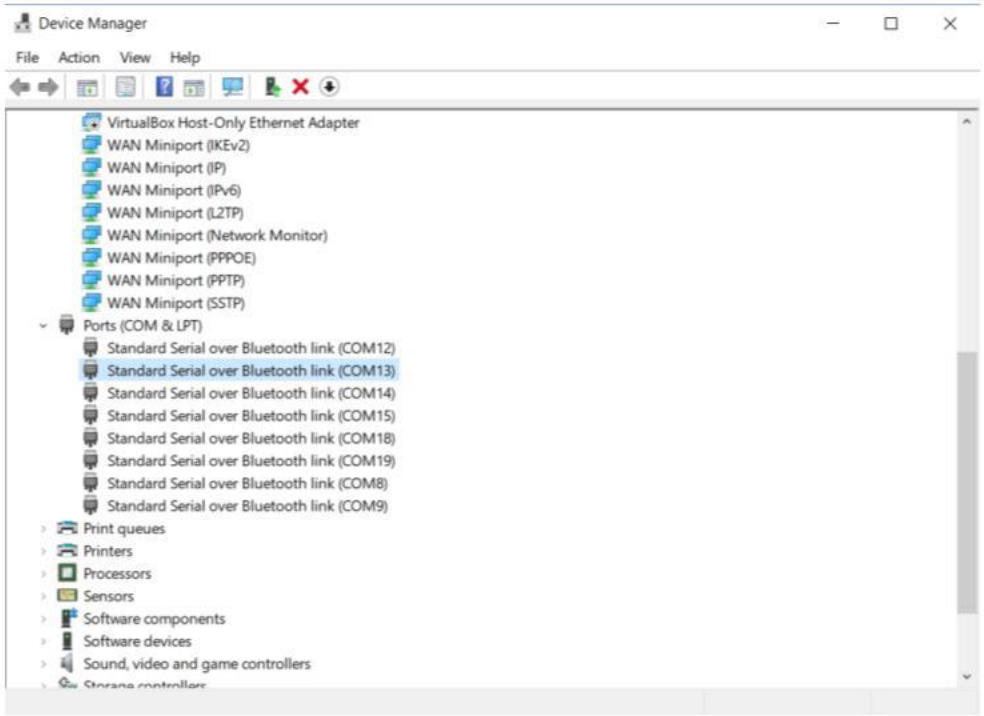




Adding Bluetooth on Windows Device Manager look for Arduino-Drone BT device

Take note on which Serial Com port its added to in Device Manager





in Device Manager Located in COM & LPT

Select the com port your Bluetooth is connected to .

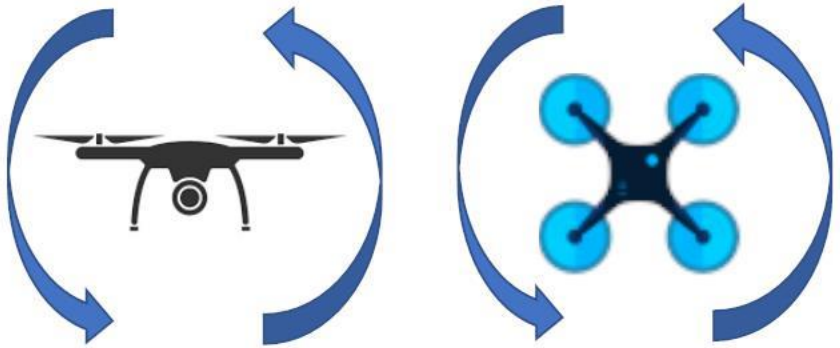
At this point Disconnect your Physical USB and your drone should be running on batteries using only the Bluetooth to communicate

Connect to the Drone with the associated COM port and Baud as found in your device manager

Refresh Rate . Telemetry update speed

Acc Calibration . Set the drone down on a level surface . Away from any metal objects for 30 secs.

Mag Calibration . Move the drone 360 degrees in all axis within 1 min. while the blue Led flashes

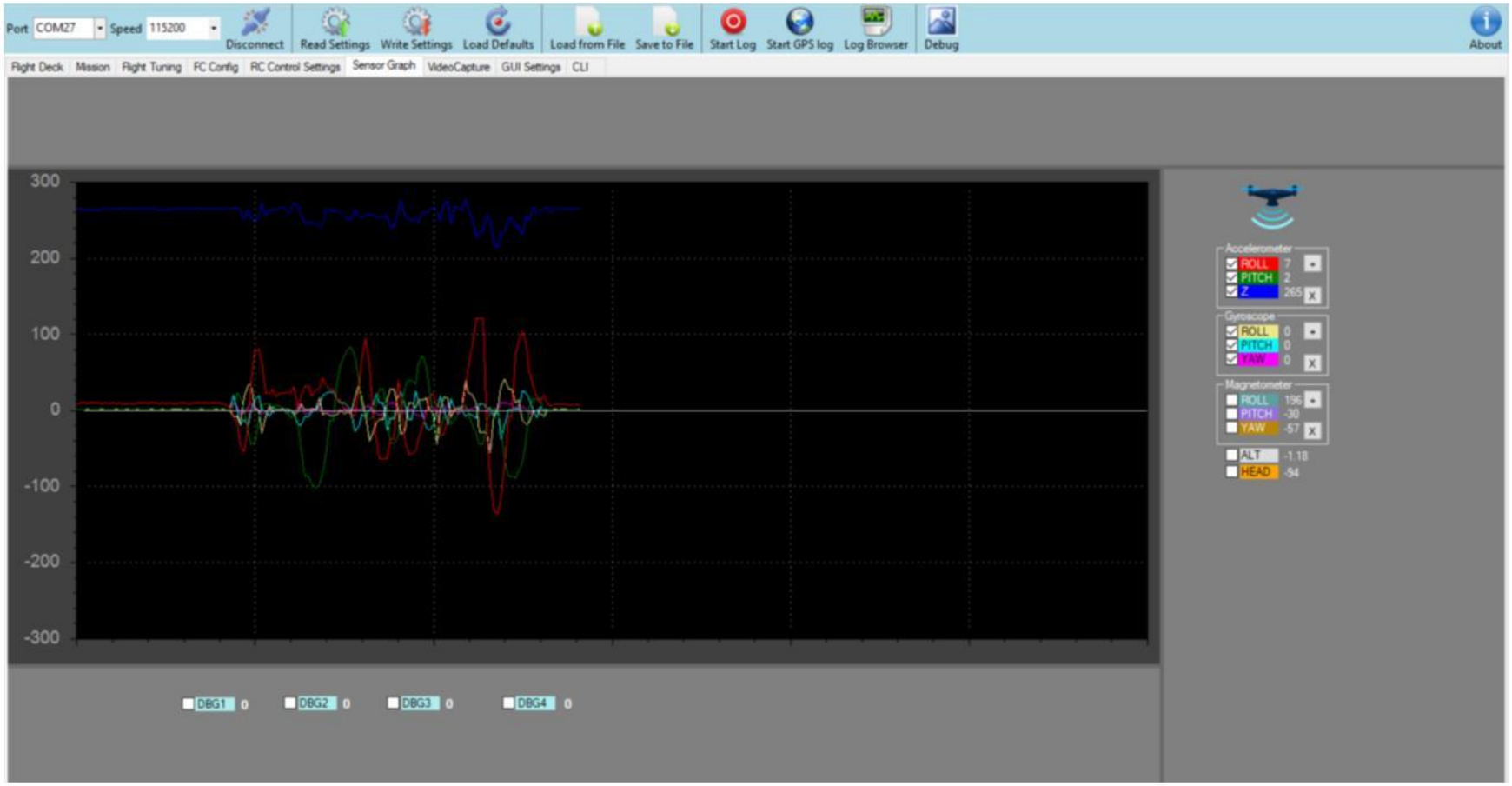


Mag Calibration must be perform when launching your drone in a new location for the first time. Pls verified the Compass if the drone heading matches your compass app in your phone.

These Calibration must be perform after Parameter updates after Flashing the firmware
Blue LED would flash during these calibration processes

Graphs and Sensors

Upload the sketch to the Arduino attach to the drone shield and open the FlywiiGUI sensor Graphs tab and hit connect to the appropriate COM your drone is connected to



Example : if roll the drone to the right the Accelerometer and Gyroscope graphs would show positive numbers and to the Left Negative numbers

If Lift the drone up Vertically the accelerometer Z axis should shows positive numbers and altitude should show a climb in meters , and if you hold the drone upsidedown Z axis would show Negative numbers

the correct orientation

ACC

Roll Right + no#

Pitch nose down + No#

Z up + No#

GYRO

Roll Right + no#

Pitch nose down + No#

Yaw Right +No#

MAG

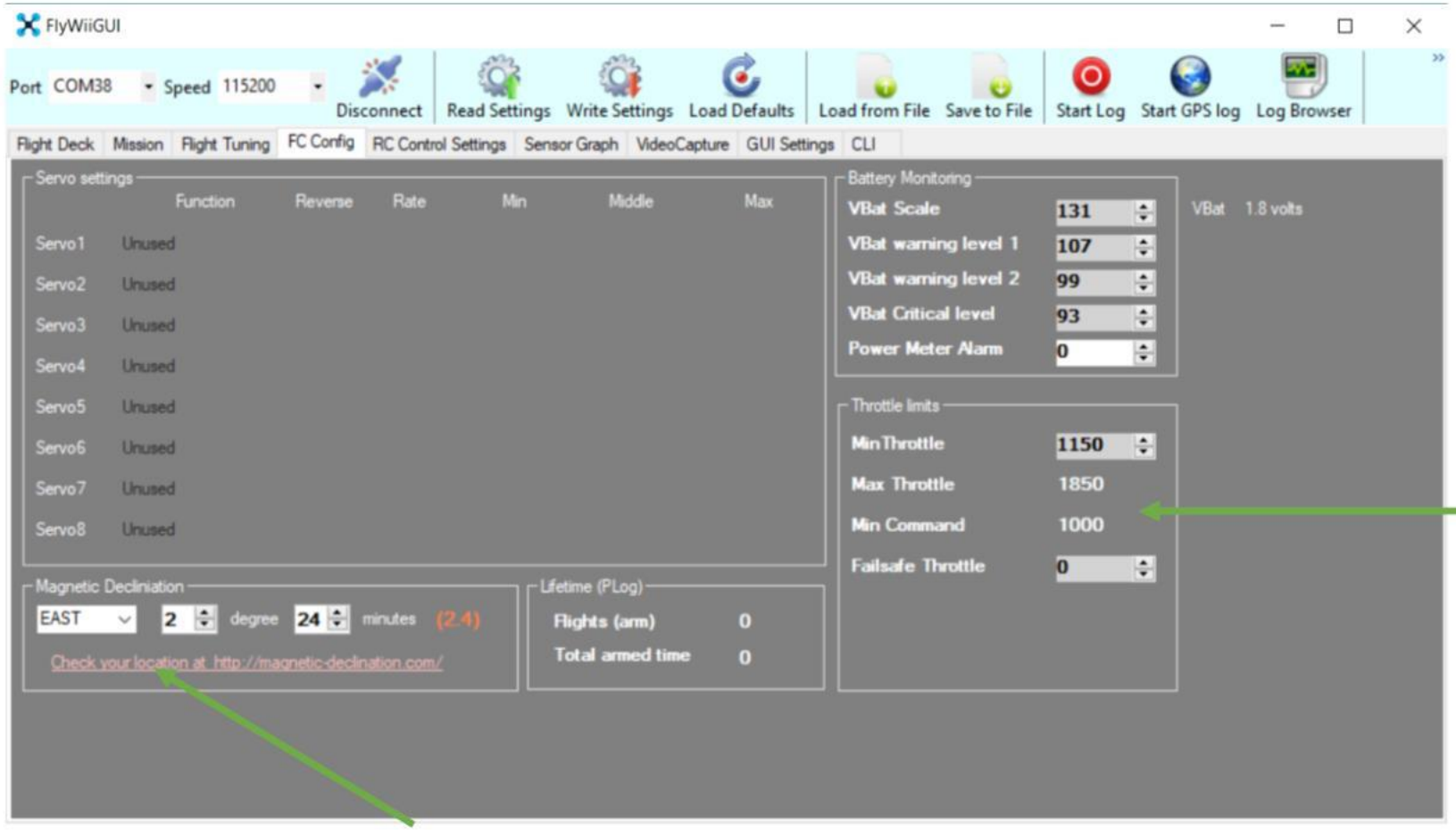
Mag & HEAD degrees corresponds to the compass

(0 degrees = North)

BARO

Alt up +no#

FC Config Functions



MOTOR THROTTLE RANGE PWM TO THE MOTOR

THIS ALSO CONTROLS THE MOTOR IDLE SPEED ON ARM

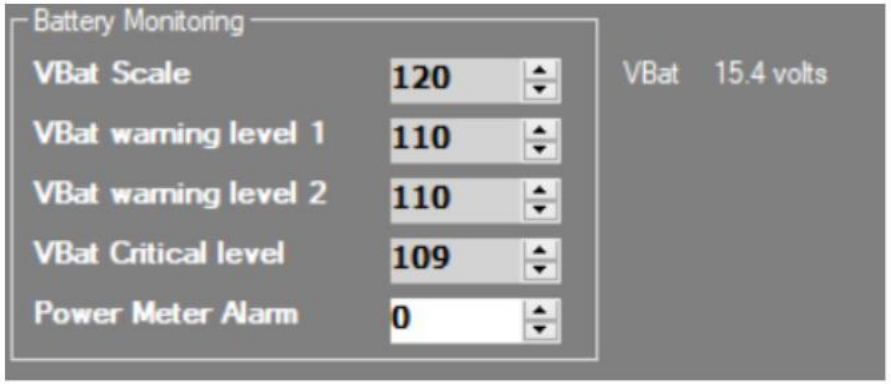
IMPORTANT TO KNOW THE MAGNETICDECLINATION OF YOUR REGION

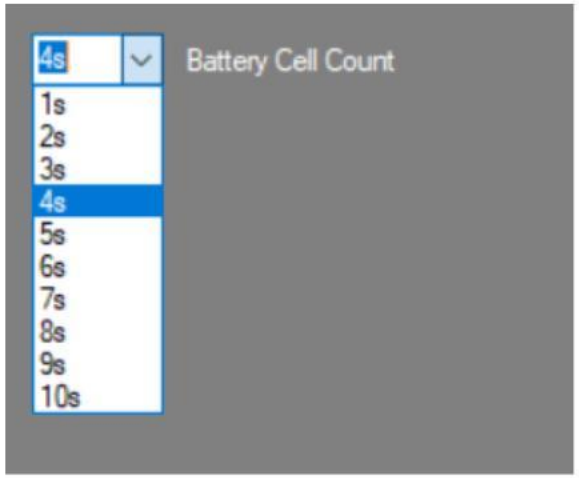
THIS AID ANY AUTONOMOUS FUNCTION THAT REQUIRES COMPASS

- HEADING HOLD
- GPS HOLD
- RTH
- MISSION

CALIBRATE COMPASS AT THE FLIGHT DECK TAB AFTER SETTING THIS UP

BATTERY VOLTAGE CALIBRATION





(FC CONFIG TAB)

BATTERY MONITORING

VBAT SCALE - ADJUST THIS TO MATCH THE BATTERY VOLTAGE OUTPUT USING THE VOLTAGE ALARM INDICATOR

VBAT WARNING LEVEL – IDENTIFY THE NOTICE WHEN THE BATTERY DROPS TO THIS VOLTAGE

(GUI SETTINGS TAB)

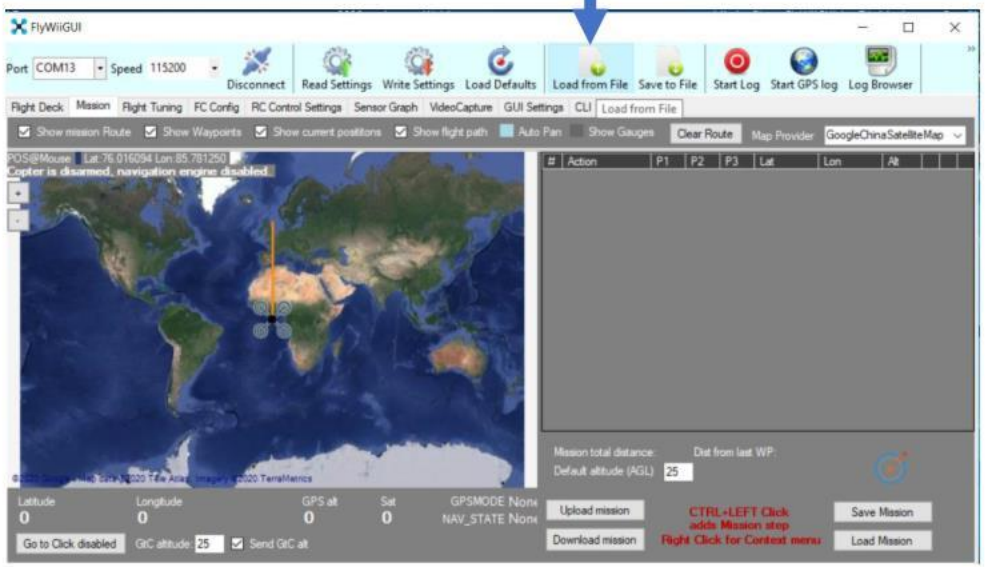
BATTERY CELL COUNT- ADJUST THIS DEPENDING ON THE NUMBER OF CELLS

THIS BOARD SUPPORTS 2S-4S BATTERY

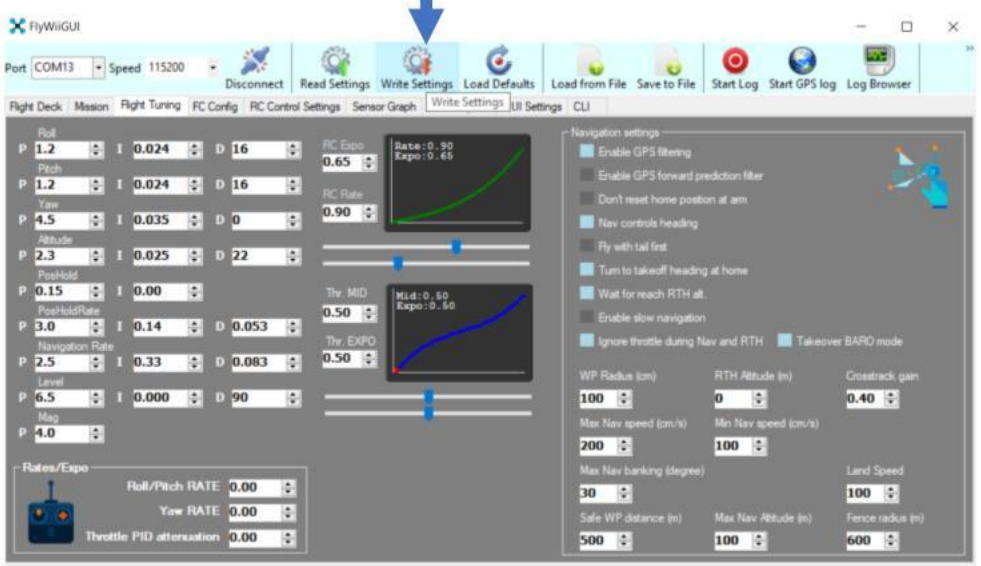
BATTERY

Download PID Parameters Preset Unzip and Load the PID file and Write settings after changes made in any of the parameters

Load the PID file



Write PID Parameter to drone



PID Parameters Preset*

PID 250mm v1.1 Slow Rate* Download

250mm v1.1 Kwad - slow rate PID.mws

FLIGHT TUNING

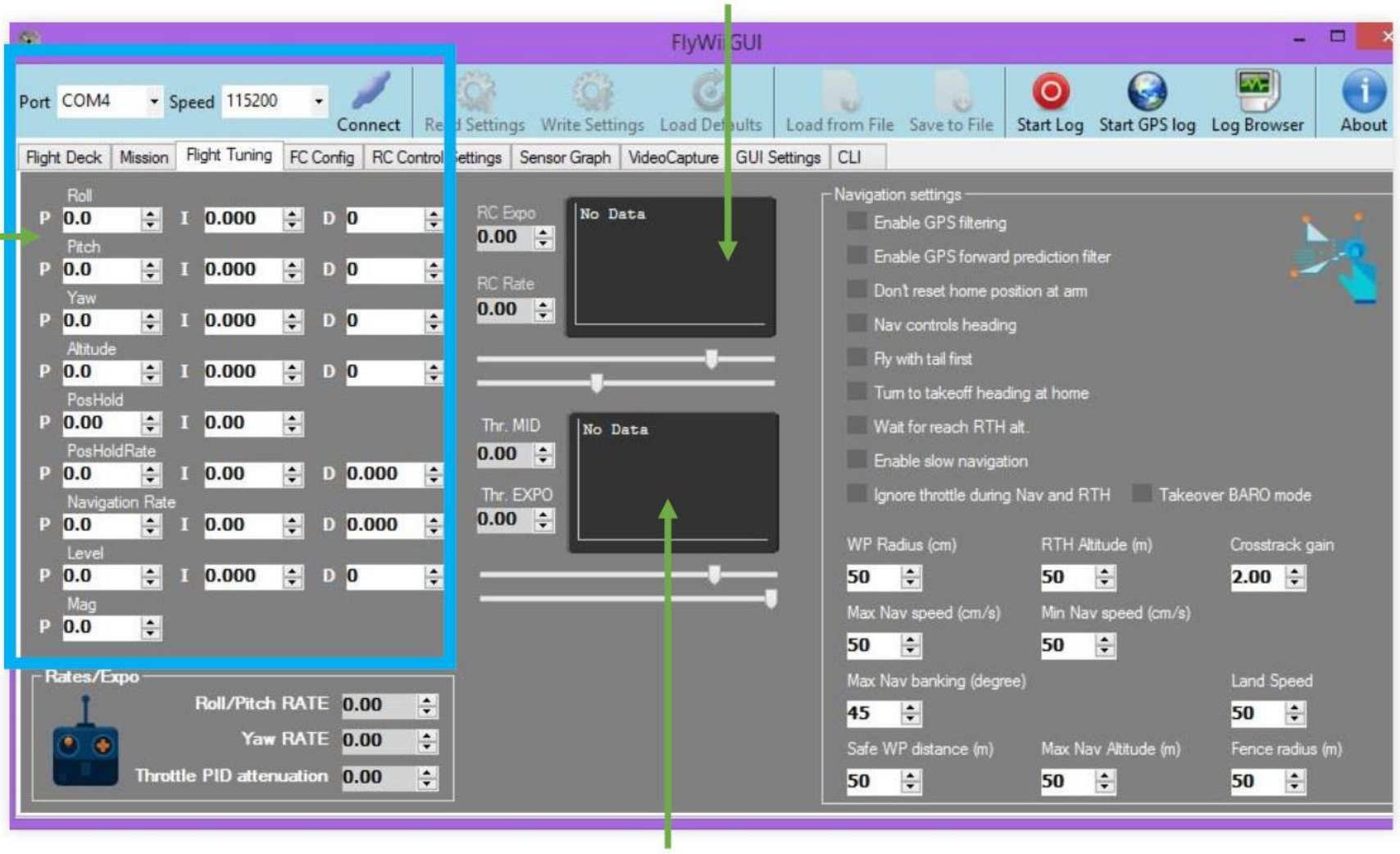
RC RADIO EXPO RATE CONTROL , THIS CONTROL HOW RESPONSIVE YOUR DRONE RESPOND TO YOUR STICK INPUT

FLIGHT TUNING FOR STABILITY AND CONTROL (PID)

PROPORTION

INTEGRAL

DERIVATIVE



THROTTLE CURVE EXPO THIS ALSO CONTROLS THE THROTTLE RESPONSIVENESS AND THE DEAD ZONE FOR ALTITUDE HOLD

Understanding PID and its relation to stability and Controls

PID : Level of your Gyro

PID : X ,Y Axis Level of your Accelerometer

PID : heading of your magnetometer Calibration

PID : Barometer and Z accelerometer

PID : sensitivity of GPS position reaction

Understanding impact of P, I and D

P : this is the amount of corrective force applied to return the MultiRotor back to its initial position

The amount of force is proportional to a combination of the the deviation from initial position minus any command to change direction from the controller input. A higher P value will create a stronger force to resist any attempts to change it's position. If the P value is too high, on the return to initial position, it will overshoot and then opposite force is needed to compensate. This creates an oscillating effect until stability is eventually reached or in severe cases becomes completely destabilised.

I : this is the time period for which the angular change is sampled and averaged

The amount of force applied to return to initial position is increased by the I factor the longer the deviation exists until a maximum force value is reached. A higher I will increase the angular hold capability.

D : this is the speed at which the MultiRotor is returned to its original position

Increasing value for D: Improves the speed at which deviations are recovered

With fast recovery speed comes a higher probability of overshooting and oscillations
Will also increase the effect of P

P – proportional

P provides a proportional amount of corrective force based upon the angle of error from desired position. The larger the deviation, the larger the corrective force.

A higher P value will create a stronger force to return to desired position. If the P value is too high, on the return to initial position, it will overshoot and then opposite force is needed to compensate. This creates an oscillating effect until stability is eventually reached or in severe cases, the overshoot becomes amplified and the multi-rotor becomes completely destabilised.

Increasing value for P :It will become more solid/stable until P is too high where it starts to oscillate and lose control. You will notice a very strong resistive force to any attempts to move the Multi-Rotor

Decreasing value for P: It will start to drift in control until P is too low when it becomes very unstable. Will be less resistive to any attempts to change orientation

Aerobatic flight: Requires a slightly higher P

Gentle smooth flight: Requires a slightly lower P

I – Integral

“I” gain provides a variable amount of corrective force based upon the angle of error from desired position.

The larger the deviation and / or the longer the deviation exists, the larger the corrective force. It is limited to prevent becoming excessively high.

A higher I will increase the heading hold capability

Increasing value for I: Increase the ability to hold overall position, reduce drift due to unbalanced frames etc

Decreasing value for I:Will improve reaction to changes, but increase drift and reduce ability to hold position

D- Divide / Derivative

This moderates the speed at which the Multi-Rotor is returned to its original position.

A lower D will mean the Multi-Rotor will snap back to its initial position very quickly

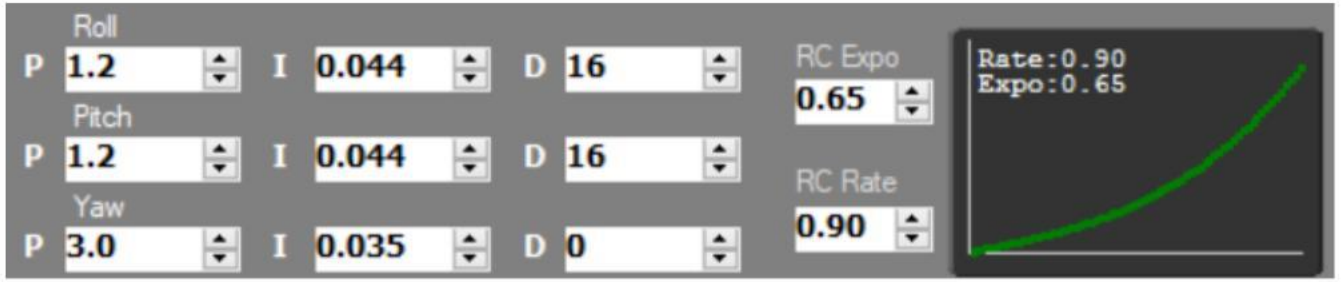
Increasing value for D: Dampens changes. Slower to react to fast changes

Decreasing value for D: Less dampening to changes. Reacts faster to changes

Aerobatic flight: Lower D

Gentle smooth flight: Increase D

Basic PID Tuning – on the ground



1 - Set PID to the designers default recommended settings.

2 - Hold the Multi-Rotor securely and safely in the air.

3 - Increase throttle to the hover point where it starts to feel light.

4 - Try to lean the Multi-Rotor down onto each motor axis. You should feel a reaction against your pressure for each axis.

5 - Change P until it is difficult to move against the reaction. Without stabilisation you will feel it allow you to move over a period of time. That is OK

6 - Now try rocking the Multi-Rotor. Increase P until it starts to oscillate and then reduce a touch.

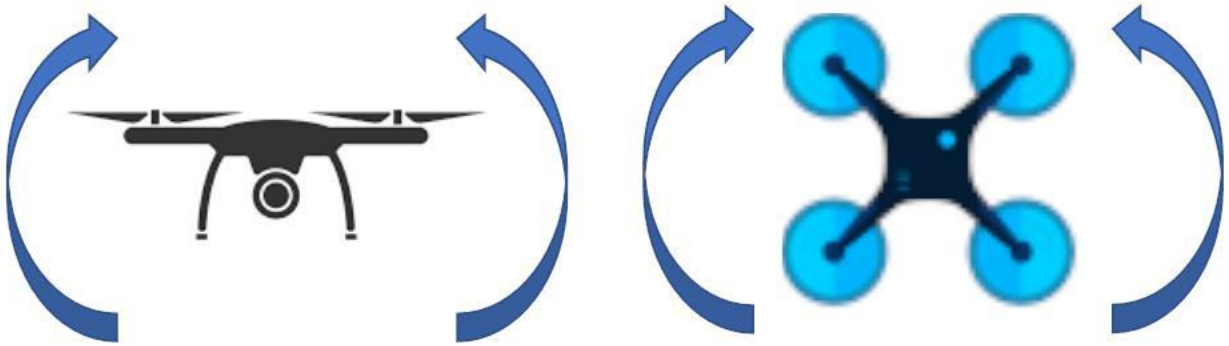
7 - Repeat for Yaw Axis. Your settings should now be suitable for flight tuning.

You will have to accept a compromise of optimal settings for stable hover and your typical mode of flying.

Obviously factor it towards your most common style.

Other factors affecting PID Taking known good PID values from an identical configuration will get you close, but bear in mind no two Multi-Rotors will have the same flying characteristics and the following items will have an impact on actual PID values:

1 - Frame weight /size / material / stiffness
2 - Motors - power / torque /momentum
3 - Position - Motor-->motor distance (I.E. frame size)
4 -ESC / TX - power curves
5 - Prop - diameter / pitch / material
6 - BALANCING
7 - Pilot skills



Advanced Tuning - practical implementation

For Aerobatic flying: Increase value for P until oscillations start, then back of slightly

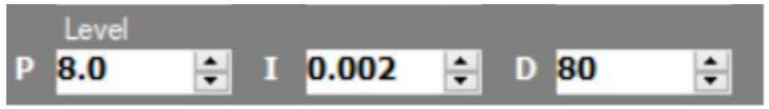
Change value for I until wobble is unacceptable, then decrease slightly

Decrease value for D until recovery from dramatic control changes results in unacceptable recovery oscillations, then increase D slightly Repeat above steps

For stable flying (RC): Increase value for P until oscillations start, then back of quite a bit

Decrease value for I until it feels too loose /unstable then increase slightly Increase value for D

PID : Level

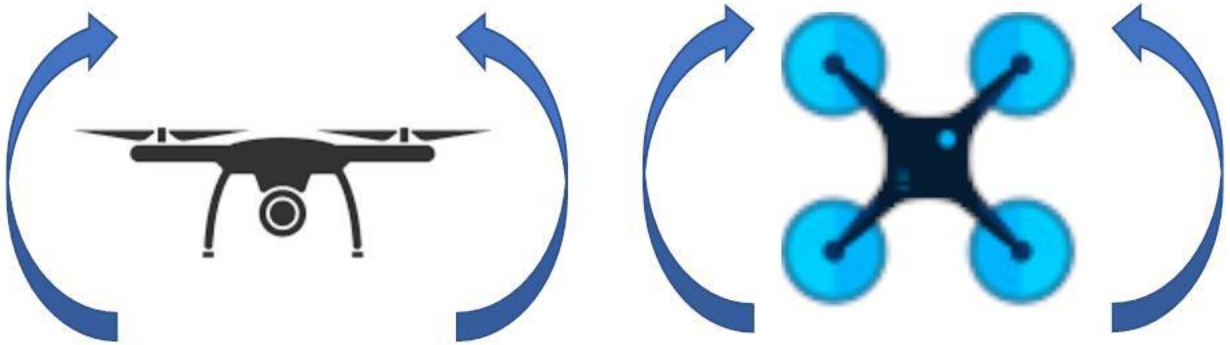


This will influence the flight characteristic with an accelerometer : this is Level Mode

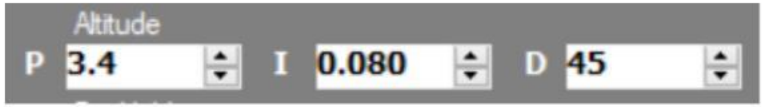
P is the dominant part of autolevel mode.
I will tell how much force must be applied when the mesured angle error persit
D is used to clamp the maximum correction for autolevel mode

Increase value for P will make the autolevel mode stronger

For smooth operation the sum of P axis + P level should stay near the default value : if you decrease P for Roll and Pitch axis you can increase P Level



ALT Hold



The Barometer sensor is used to detect the altitude of your multirotor aircraft and is used for altitude hold mode. As the barometer sensor is not very precise and is quite noisy, detection of small up and down movements is impossible.

So small up and down movements are detected by the accelerometer Z axis.
Combination of these two sensors gives good altitude hold.
PID settings for ALT works like this:

P - Is how much the multirotor should rely on the barometer sensor. The higher the value is the stronger the multirotor relies on the Barometer reading.

I - Is used to compensate for drift caused by battery voltage drop during the flight. The higher the value is more the multirotor will react to voltage drops (or other varying factors over time).

D - Is how strong the multirotor should react to data from the accelerometer Z axis. It is used to react to small up and down movements that the barometer cannot accurately sense. The higher the value is the stronger the multirotor will react to small altitude changes.

1. So set the P and I to 0

2. Start to play with D value only. To high D may cause yoyo effect (up and down oscillations). With to low D copter will be not able to react strong/fast enough to hold altitude. Your goal here is to set D to the value when copter don';t oscillate up and down and also holds altitude quite well for a not very long period of time. Copter will not hold altitude perfectly at this point during long periods. It will slowly drift up or down, but altitude should be quite stable in short periods.

3. Start to increase P to the point where copter holds altitude over long time period. If the value is to small the copter will drift slowly up and down. If the value is to high yoyo effect may appear. Goal here is to set it to the point where copter holds altitude for quite some time. Copter will still go slowly down due to battery voltage drop over time.

4. "I" is used to compensate the voltage drop. So start to increase the "I" value slowly until you get a perfect position hold during a very long time.
Now your altitude hold should be good enough.

For Mega 2560 + GPS Pos Rate PID controller & Pos Rate PID Tuning

Pos Rate PID controller
Pos Rate PID Tuning

The Pos Rate PID controller takes the commanded speed output from the Pos PI controller and commands an attitude in order to maintain the position hold location. This PID controller should be tuned before adjusting the Pos PI controller.

The Pos Rate PID settings control how the attitude of the multirotor is changed in order to move towards the desired hold location.

The speed of movement is controlled by the Pos PI controller, while the attitude of the multirotor is controlled by Pos Rate.

When the multirotor is within the defined distance of the hold location or waypoint (set by GPS_WP_RADIUS in config.h) the Pos Rate PID is used, when further away from the location the Nav Rate PID is used to return to within the defined waypoint radius.

For Mega 2560 + GPS Pos Rate PID controller & Pos Rate PID Tuning

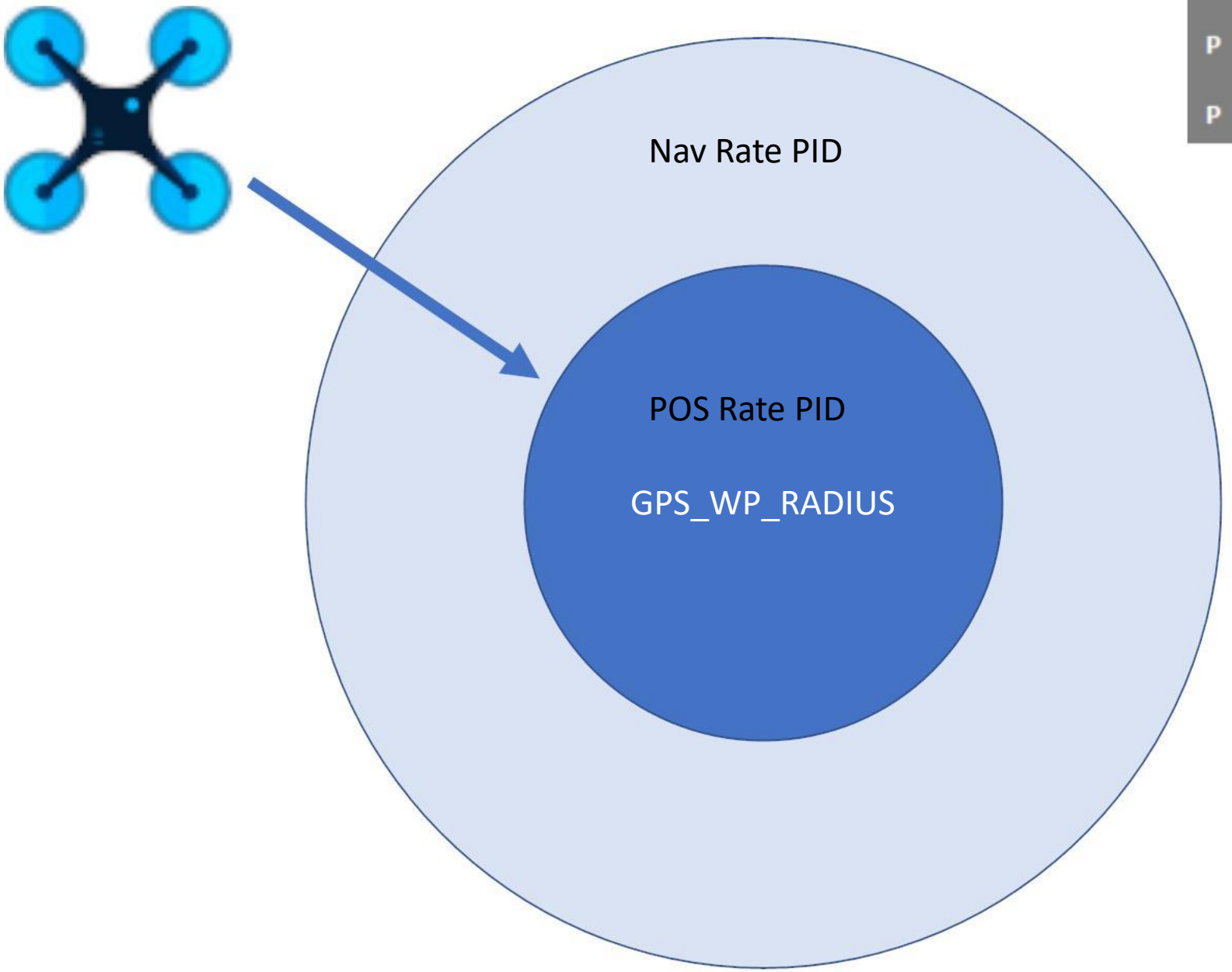
The Pos Rate PID controller takes the commanded speed output from the Pos PI controller and commands an attitude in order to maintain the position hold location. This PID controller should be tuned before adjusting the Pos PI controller.

The Pos Rate PID settings control how the attitude of the multirotor is changed in order to move towards the desired hold location.

The speed of movement is controlled by the Pos PI controller, while the attitude of the multirotor is controlled by Pos Rate.

When the multirotor is within the defined distance of the hold location or waypoint (set by GPS_WP_RADIUS in config.h) the Pos Rate PID is used, when further away from the location the Nav Rate PID is used to return to within the defined waypoint radius.

To tune the Pos Rate PID, initially set P, I and D values to 0.
Gradually increase P until the multirotor begins to position hold with some drift.
Gradually increase D until the multirotor responds more quickly to undesired changes in attitude caused by the wind. If this value is set too high you will see oscillations or sudden jerking in pitch and roll motion.
If needed, gradually increase I value to allow the PID controller to compensate for long lasting error, ie if it is being blown by the wind.



PosHold
P 0.15 I 0.00

PosHoldRate
P 3.4 I 0.14 D 0.053

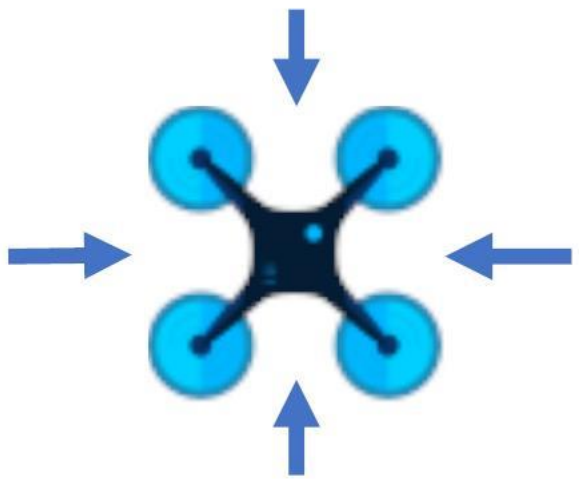
Navigation Rate
P 2.5 I 0.33 D 0.083

To tune the Pos Rate PID, initially set P, I and D values to 0.

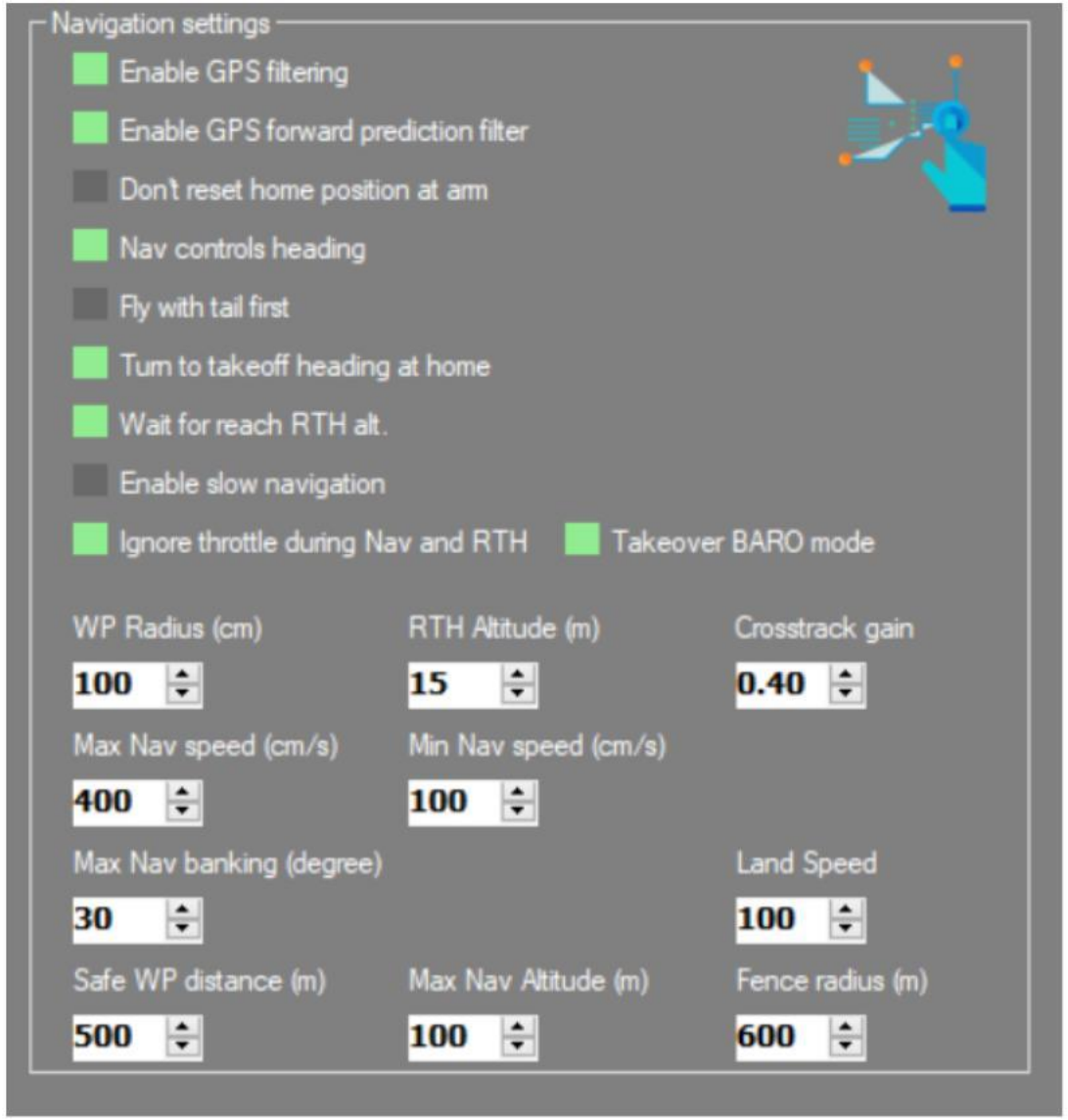
Gradually increase P until the multirotor begins to position hold with some drift.

Gradually increase D until the multirotor responds more quickly to undesired changes in attitude caused by the wind. If this value is set too high you will see oscillations or sudden jerking in pitch and roll motion.

If needed, gradually increase I value to allow the PID controller to compensate for long lasting error, ie if it is being blown by the wind.



Other Navigation Functions



WP Radius – the radius of the area the Pos PID with trigger it has reach the waypoint

Max Nav Speed – Maximum speed the drone travel between waypoints (too fast and you likely over shoot your target**)** ***for first mission flight test Nav speed of 100cm/s with ("Enable Slow Navigation "Active)***

Min Nav Speed – the speed the drone travel when with in the WP Radius

RTH Altitude – Altitude the drone will climb to when its below the altitude in relation to its home point when the RTH is trigger set this to 0 to RTH at current altitude

Max Nav Banking – the max allowable pitch and roll the drone will be set too while traveling between waypoints (tune this along with Max Nav Speed to take account with Environment conditions)

Max Nav Altitude – Max altitude the drone is cap to fly at

Land Speed – speed of descending for Landing cm/s

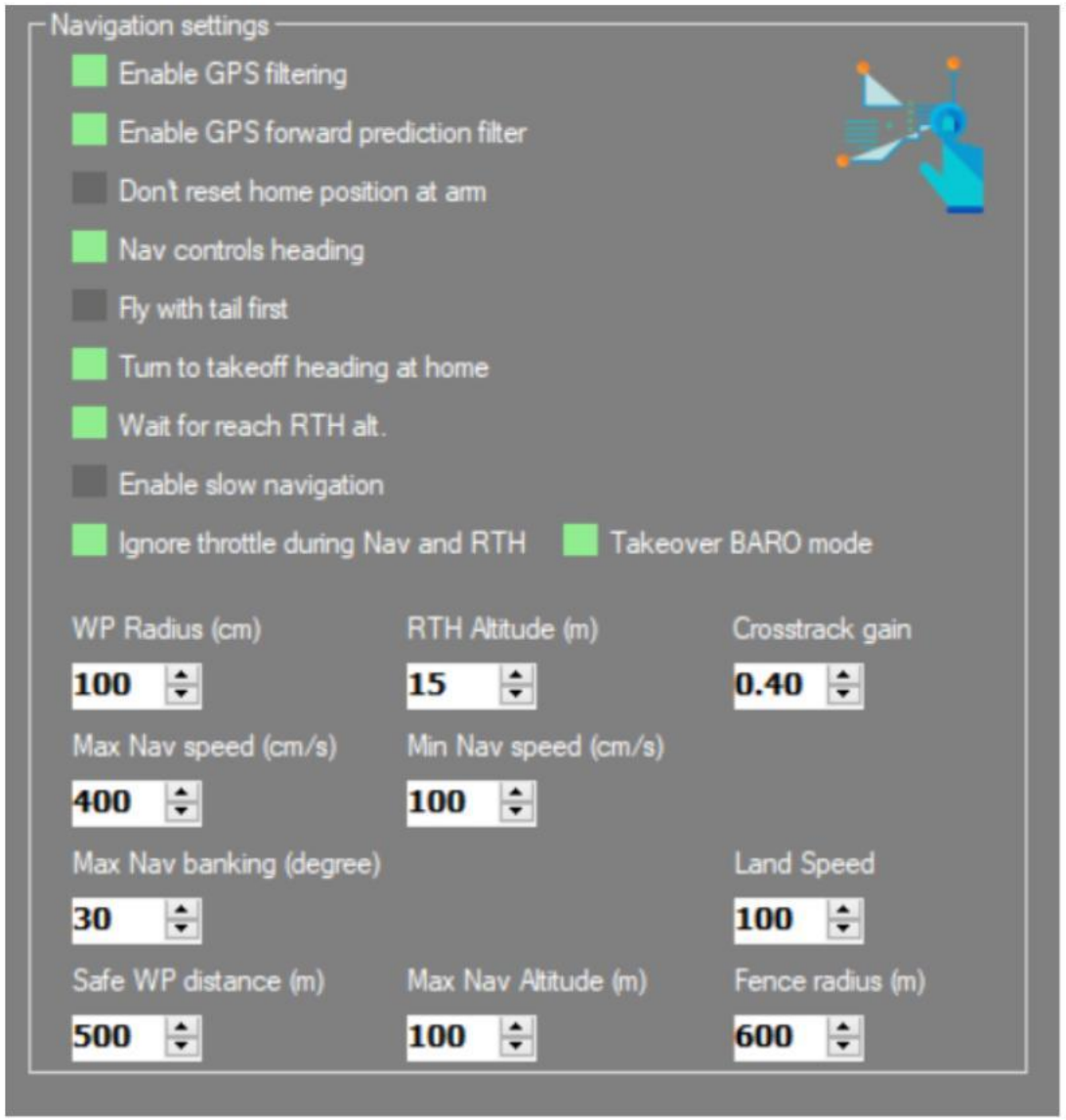
Safe WP Distance – max distance between waypoint before its null out

Fence Radius – Geo Fence to keep the drone with in the perimeter in relation to home position

CrossTrack gain - this tune the GPS and Nav sensitivity

GPS Filtering – use to enhance GPS accuracy

GPS Forward Prediction Filter – predicting the drones location and to compensate for lag . (optional) – not necessary for most application



Don't Reset Home position at Arm – this retains the home position where you first plug power on your drone

Nav Controls Heading – this points the drone to its next waypoint

Fly tail first – makes the drone fly reverse (don't use unless it's a camera pull out shot)

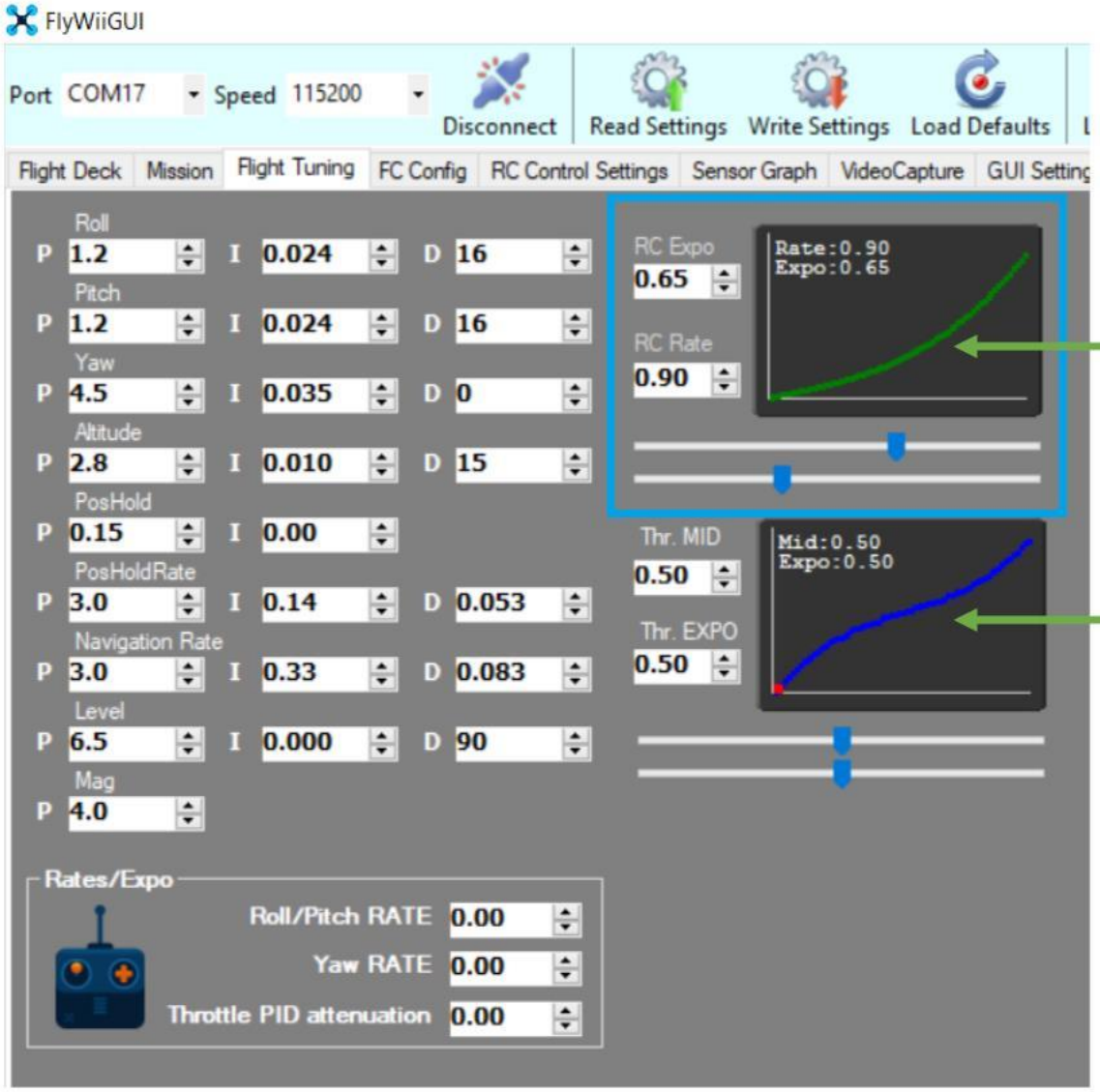
Turn take off heading at Home – when drone arrives at home position it orientates to its heading right after arming

Wait to reach RTH - this works with RTH altitude command which the drone would climb to the said altitude before initiating the flight to home position

Enable slow navigation – this works with keeping the drone to its **Min Nav speed**

Ignore throttle and Take over Baro – as the name suggest disable throttle stick command from the controller when the drone is on mission mode

FLIGHT TUNING



RC RADIO EXPO RATE CONTROL , THIS CONTROL HOW RESPONSIVE YOUR DRONE RESPOND TO YOUR STICK INPUT

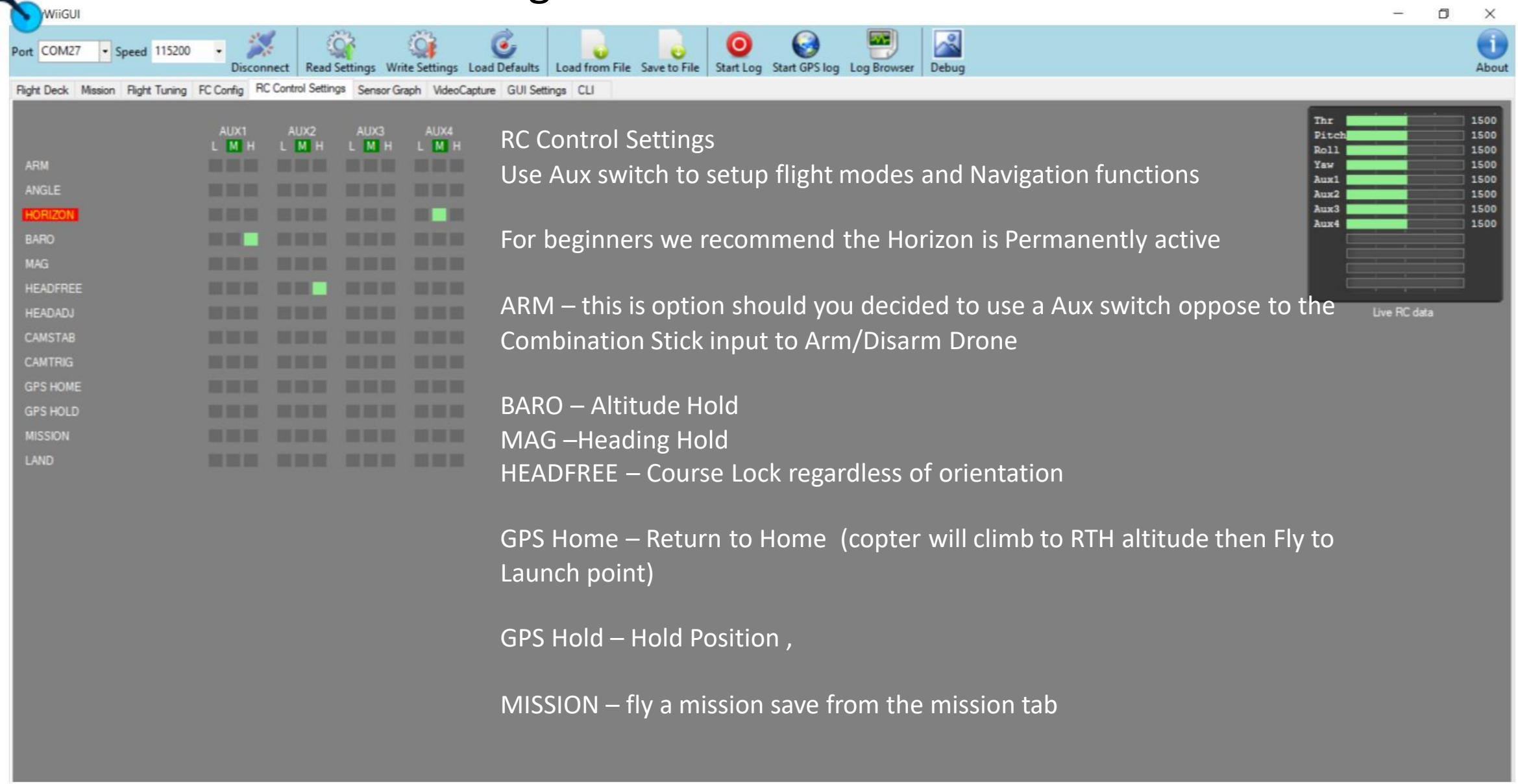
HIGHER NUMBER MEANS MORE RESPONSIVENESS THE DRONE TO STICK INPUTS

FOR BEGINNERS WE RECOMMEND RC RATE AT .50-.60 THIS OFFER A MORE SLUGGISH RESPONDS OF THE DRONE AND NOT ADVICE ABLE FOR STRONG WINDS CONDITION

THROTTLE CURVE EXPO THIS ALSO CONTROLS THE THROTTLE RESPONSIVENESS AND THE DEAD ZONE FOR ALTITUDE HOLD

ADJUST THIS FOR YOUR THROTTLE RESPOND RATE AND WHERE YOUR CENTER STICK DEADBAND FOR ALTITUDE HOLD IS

RC Control Settings



RC Control Settings
Use Aux switch to setup flight modes and Navigation functions

For beginners we recommend the Horizon is Permanently active

ARM – this is option should you decided to use a Aux switch oppose to the Combination Stick input to Arm/Disarm Drone

BARO – Altitude Hold
MAG –Heading Hold
HEADFREE – Course Lock regardless of orientation

GPS Home – Return to Home (copter will climb to RTH altitude then Fly to Launch point)

GPS Hold – Hold Position ,

MISSION – fly a mission save from the mission tab

Missions

Note: Only functional for Mega 2560 Boards with GPS

Waypoint – the drone with travel between those points

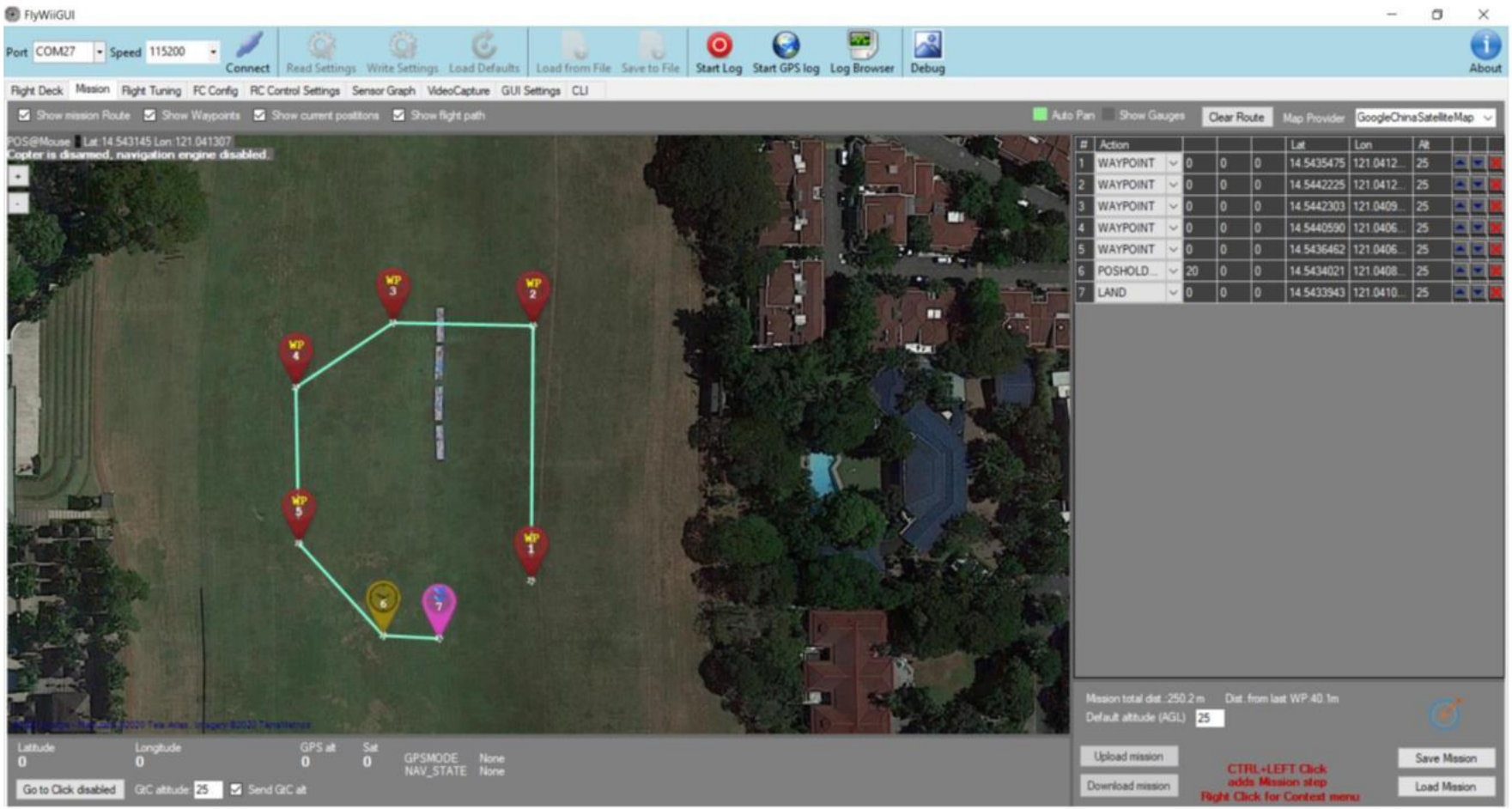
Time PosHold – Drone will wait X number of 00:00:00 then move to the next waypoint

Unlimited PosHold – once the drone reach this point it will hover and wait till you switch out of Mission mode

Land – the drone will land once it has reach this point (Must be place at the end of the mission)

RTH – the Drone will fly back to home position (Must be place at the end of the mission)

Default Alt – Altitude in meters (for first Mission test waypoint with altitude 2m-3m Above Ground Level) And set missions with 2m-3m altitude with Nav speed of 100cm/s .



RC Control Setting Tab – activate Baro , Mag , Mission

To start mission takeoff aircraft in stabilize mode up to 1-2meter altitude then switch the aux switch to mission mode .

Any time you can switch out of it on hold or stabilize mode

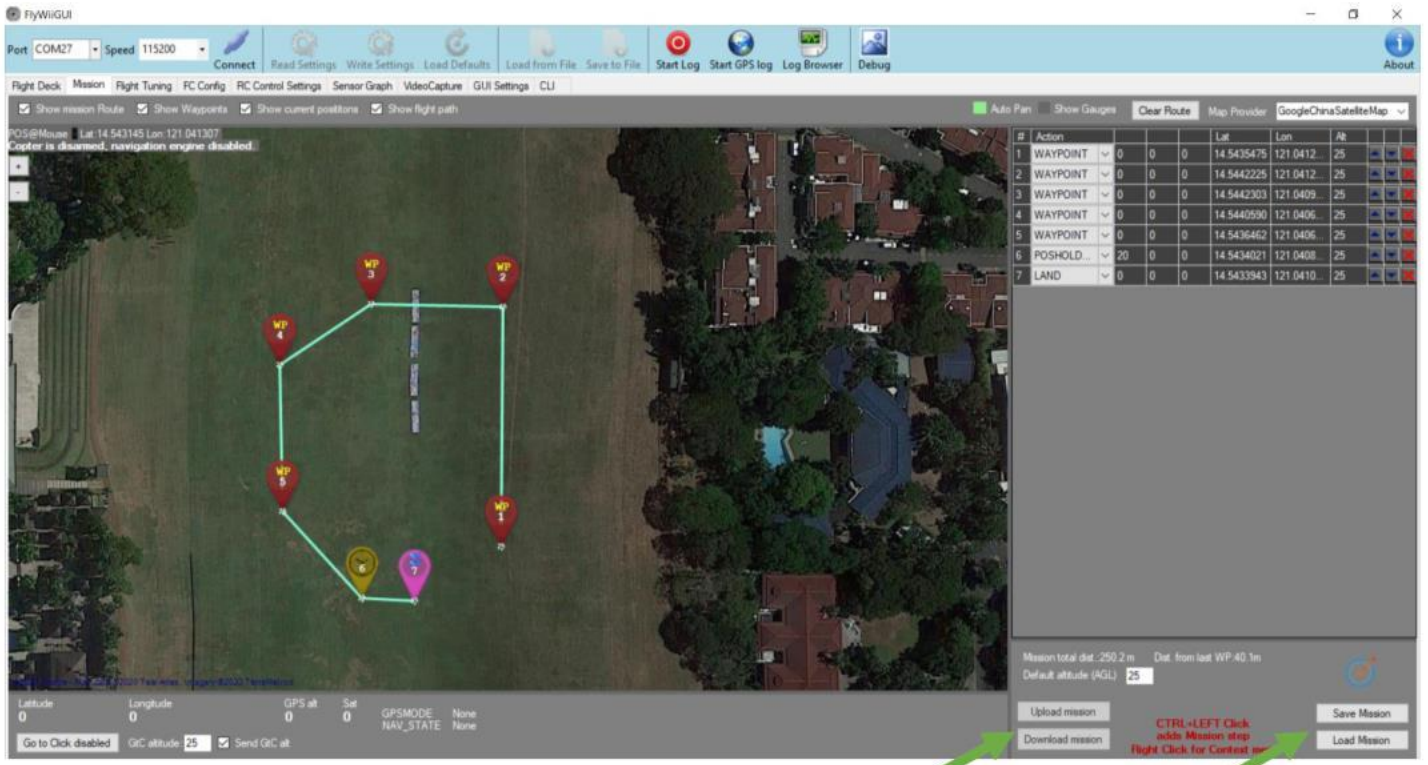
Missions

Prerequisite and process for a good mission , Points to test before performing a mission

1. Drone is flying stable in horizon and Alt hold mode , holding altitude consistently less than 1m variation over 1 minute period . Tune PID and altitude PID when necessary.

2. Drone is flying stable and holding position in GPSHold mode and Alt Mode not deviating with in a 1 x1 Meter Imaginary box , tune PosHold Rate PID when necessary

3. RTH – set RTH Altitude to 0 , Max Nav speed to 100cm/s , set aux switch to RTH ,Baro , Mag and write settings ,Fly the drone 5 Meters away from the Launch site and activate the RTH Aux switch ,see if the drone returns back to home position and holds position when arrive . Tune Navigation Rate PID when necessary

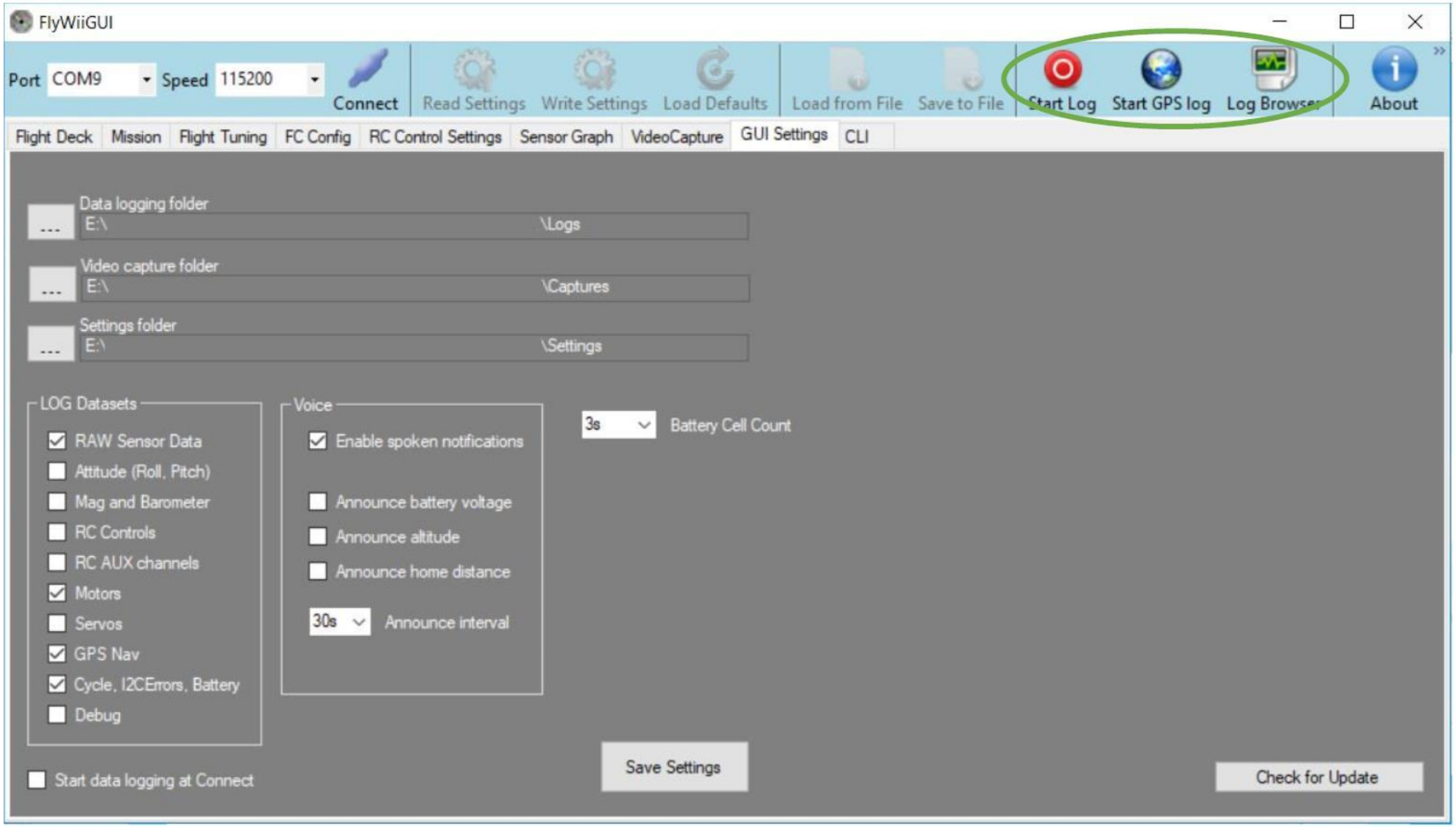


Mission upload to /download from Drone

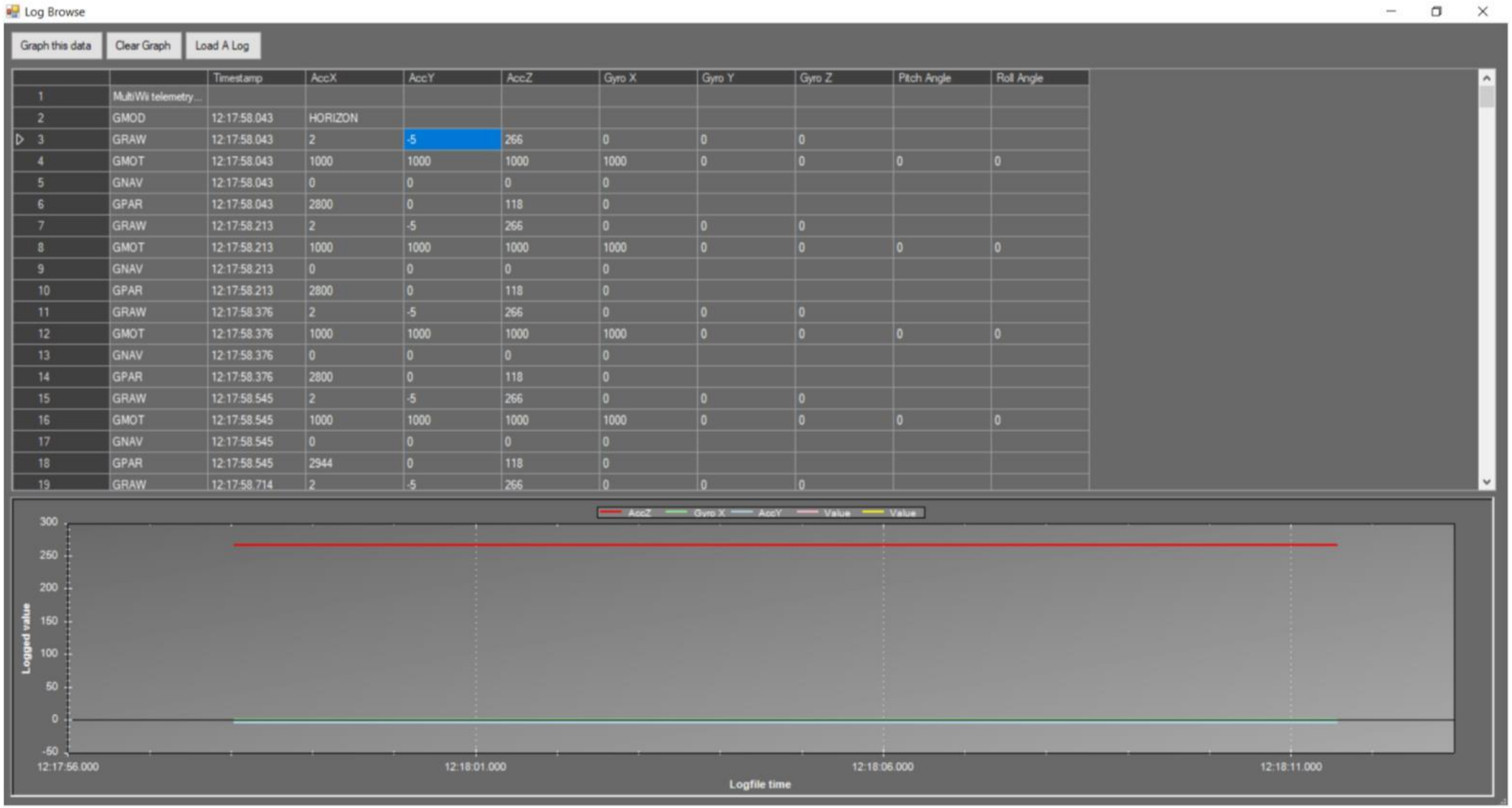
Mission Save to /Open from File

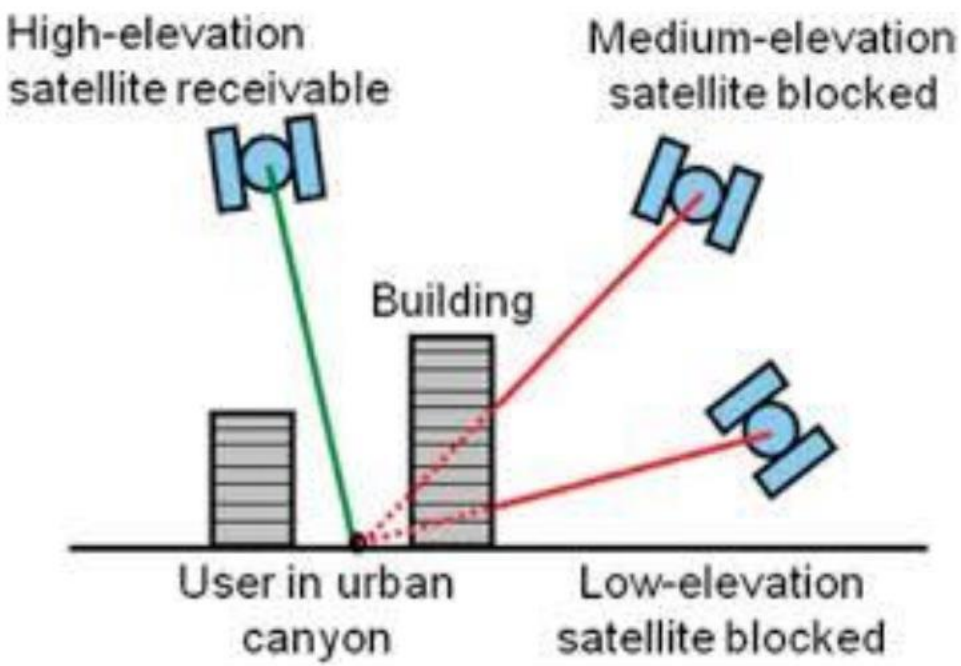
Data Logging

GUI Settings (where you save your PID ,Flight Logs and Video Logs)



Log Browser


Log Browse
Graph this data
Clear Graph
Load A Log
Timestamp
AccX
AccY
AccZ
Gyro X
Gyro Y
Gyro Z
Pitch Angle
Roll Angle
1 MultiWii telemetry...
2 GMOD 12:17:58.043 HORIZON
3 GRAW 12:17:58.043 2 -5 266 0 0 0
4 GMOT 12:17:58.043 1000 1000 1000 1000 0 0 0 0
5 GNAV 12:17:58.043 0 0 0 0
6 GPAR 12:17:58.043 2800 0 118 0
7 GRAW 12:17:58.213 2 -5 266 0 0 0
8 GMOT 12:17:58.213 1000 1000 1000 1000 0 0 0 0
9 GNAV 12:17:58.213 0 0 0 0
10 GPAR 12:17:58.213 2800 0 118 0
11 GRAW 12:17:58.376 2 -5 266 0 0 0
12 GMOT 12:17:58.376 1000 1000 1000 1000 0 0 0 0
13 GNAV 12:17:58.376 0 0 0 0
14 GPAR 12:17:58.376 2800 0 118 0
15 GRAW 12:17:58.545 2 -5 266 0 0 0
16 GMOT 12:17:58.545 1000 1000 1000 1000 0 0 0 0
17 GNAV 12:17:58.545 0 0 0 0
18 GPAR 12:17:58.545 2944 0 118 0
19 GRAW 12:17:58.714 2 -5 266 0 0 0
AccZ
Gyro X
AccY
Value
Value
300
250
200
150
100
50
0
-50
Logged value
12:17:56.000
12:18:01.000
12:18:06.000
12:18:11.000
Logfile time

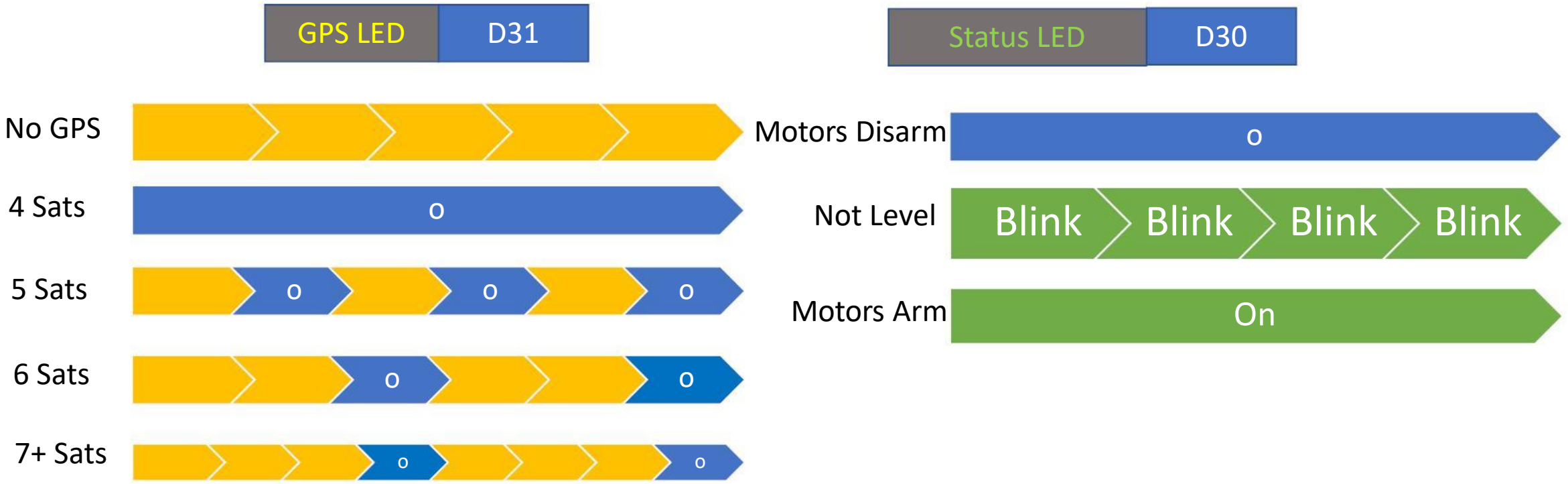


Note : GPS require a clear open area to get a proper fix and accuracy minimum 7 satellites but 10+ are Ideal

Flying next to a building can distort satellite signal deteriorating accuracy

Which in this case its better to not use GPS modes and fly Manual

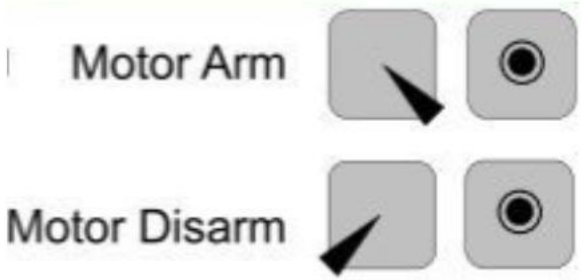
LED Indicator



indicate a valid GPS fix by flashing the LED
- led work as sat number indicator
- No GPS FIX -> LED blinks constant speed
- Fix and sat no. below 5 -> LED off
- Fix and sat no. >= 5 -> LED blinks, one blink for 5 sat, two blinks for 6 sat, three for 7 +

And your much Done on your setup

For Mode 2 Hold 2 seconds



Cannot Arm Motors

when on GPS Home , GPS Hold , Mission Flight modes & when USB is plugged in . (pls use Bluetooth telemetry)

Tests motors with Props off

Baro and Mag preferably switch off when Arming

Pls calibrate ACC and Mag in the Dashboard

Before your First Flight one Final procedure known as pre flight and Tuning tests

Before your First Flight one Final procedure
known as pre flight and Tuning tests

Note: You may need to build a suspended Test Rig
to test your drone’s

RC mapping
EXPO rates
PID Tuning
Motor Rotation & Propeller installation