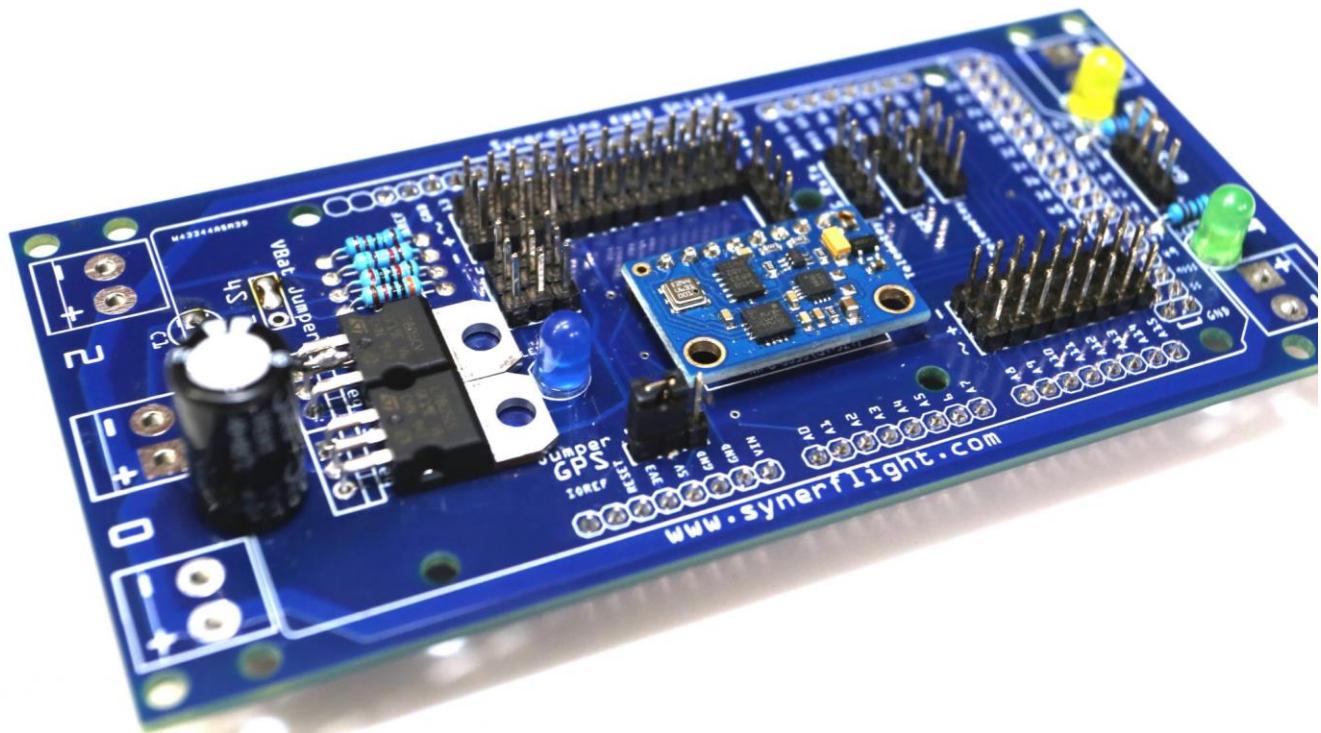


Synerduino Shield Stability and Navigation for Airplanes



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

Hardware

- Synerduino shield
- Arduino2560 MEGA / Uno 328
- Fixwing (Airplane) Airframe
- Brushed / Brushless Motor
- Micro/ Mini Servos 2-4pcs Depending on aircraft design
- Gearbox (if your motor set came with it)
- Brush / Brushless ESC for Airplane
- Ublox6 GPS
- Bluetooth

Tools needed

Screwdriver



Cutter Knife



Soldering Set



tape



Zip Ties



Pliers



Tools Needed

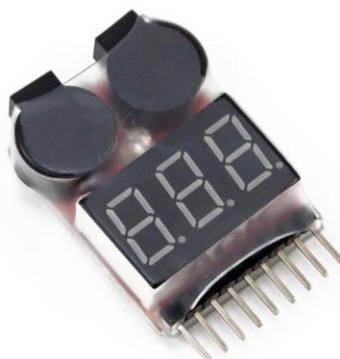
Lipo Charger 5A



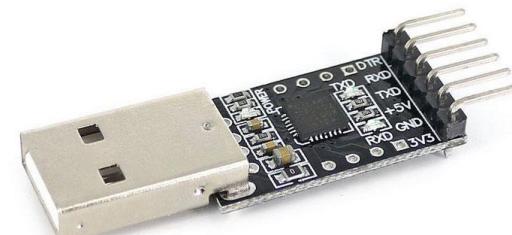
Lipo Battery 3s 1300mah



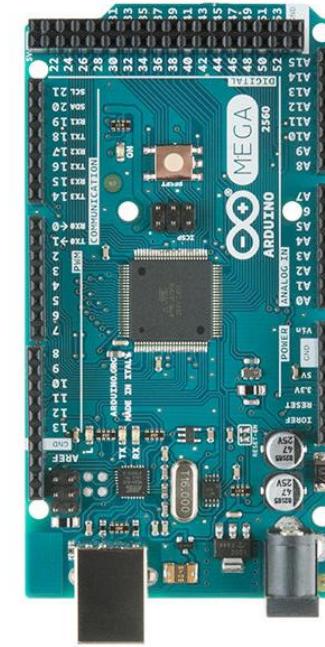
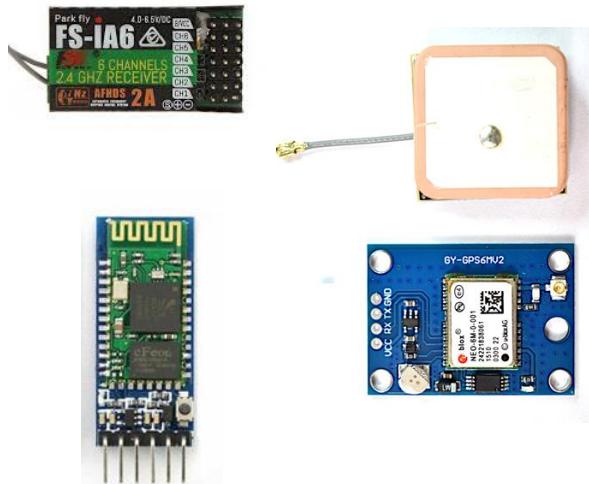
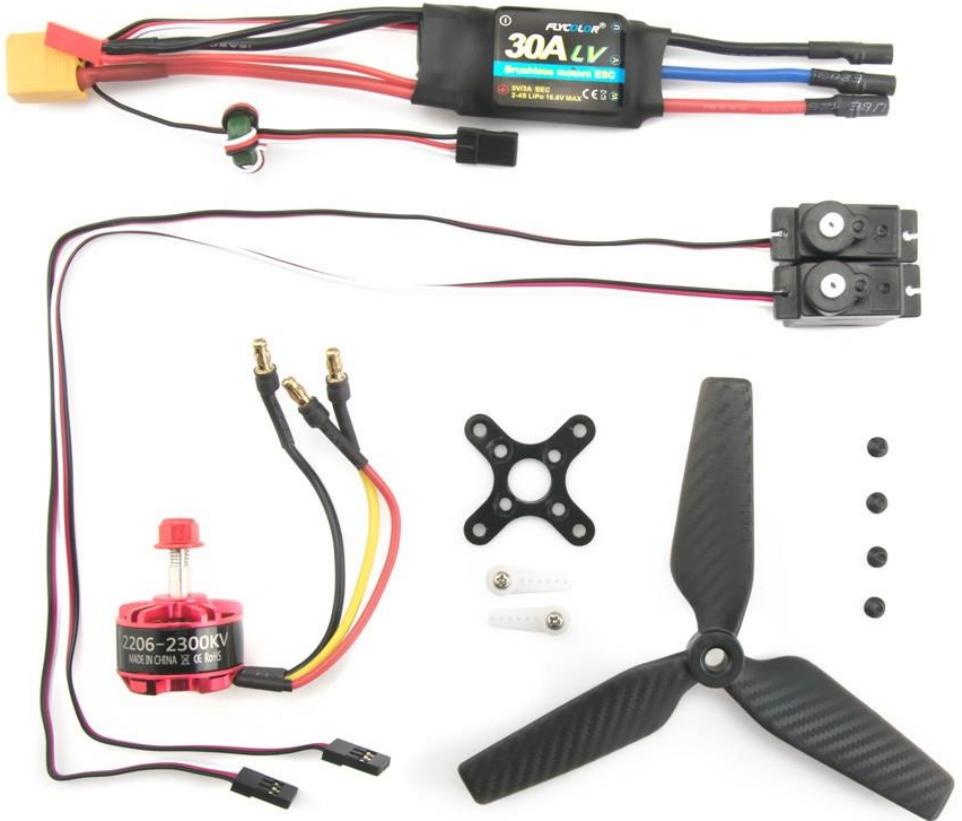
Voltage alarm



USB TTL FTDI



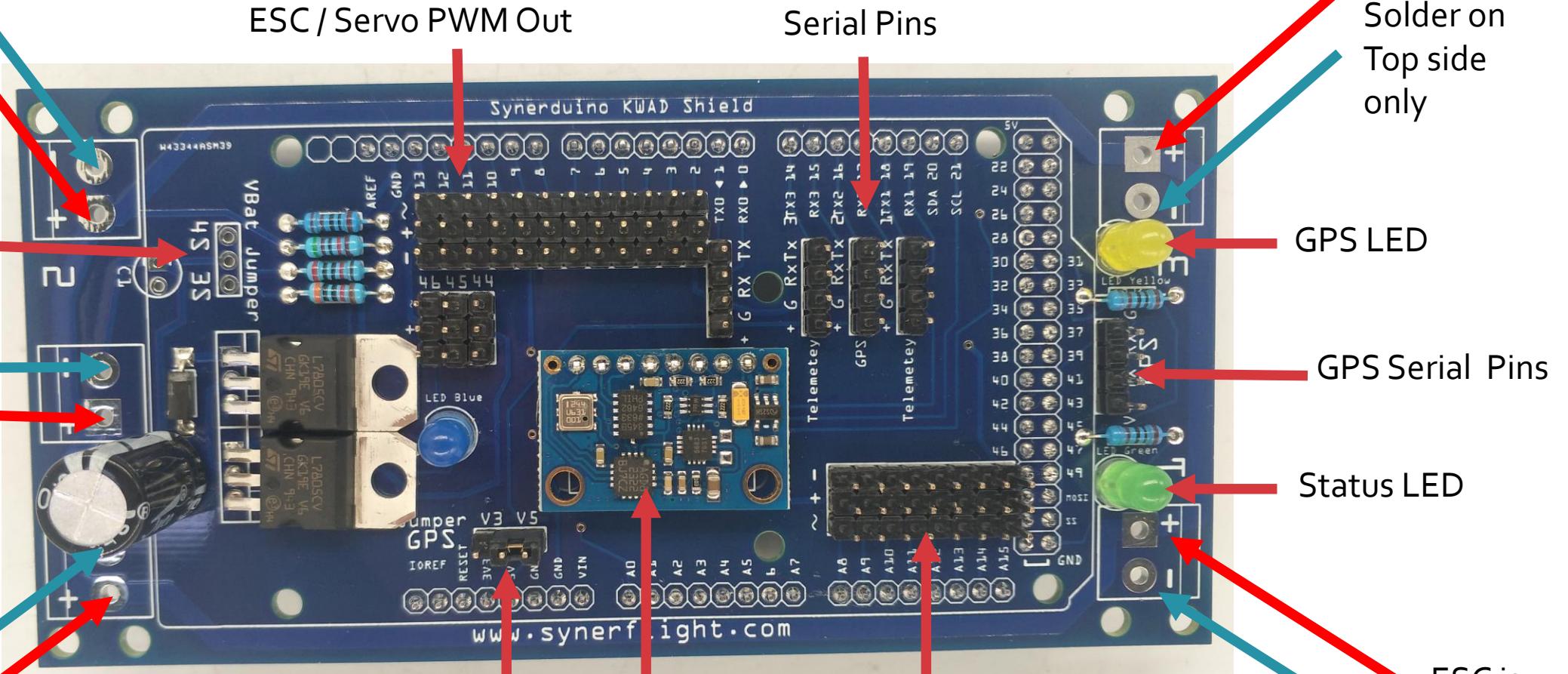
Hardware



Synerduino Kwad Shield BETA GY801

ESC is
Solder on
Top side
only

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

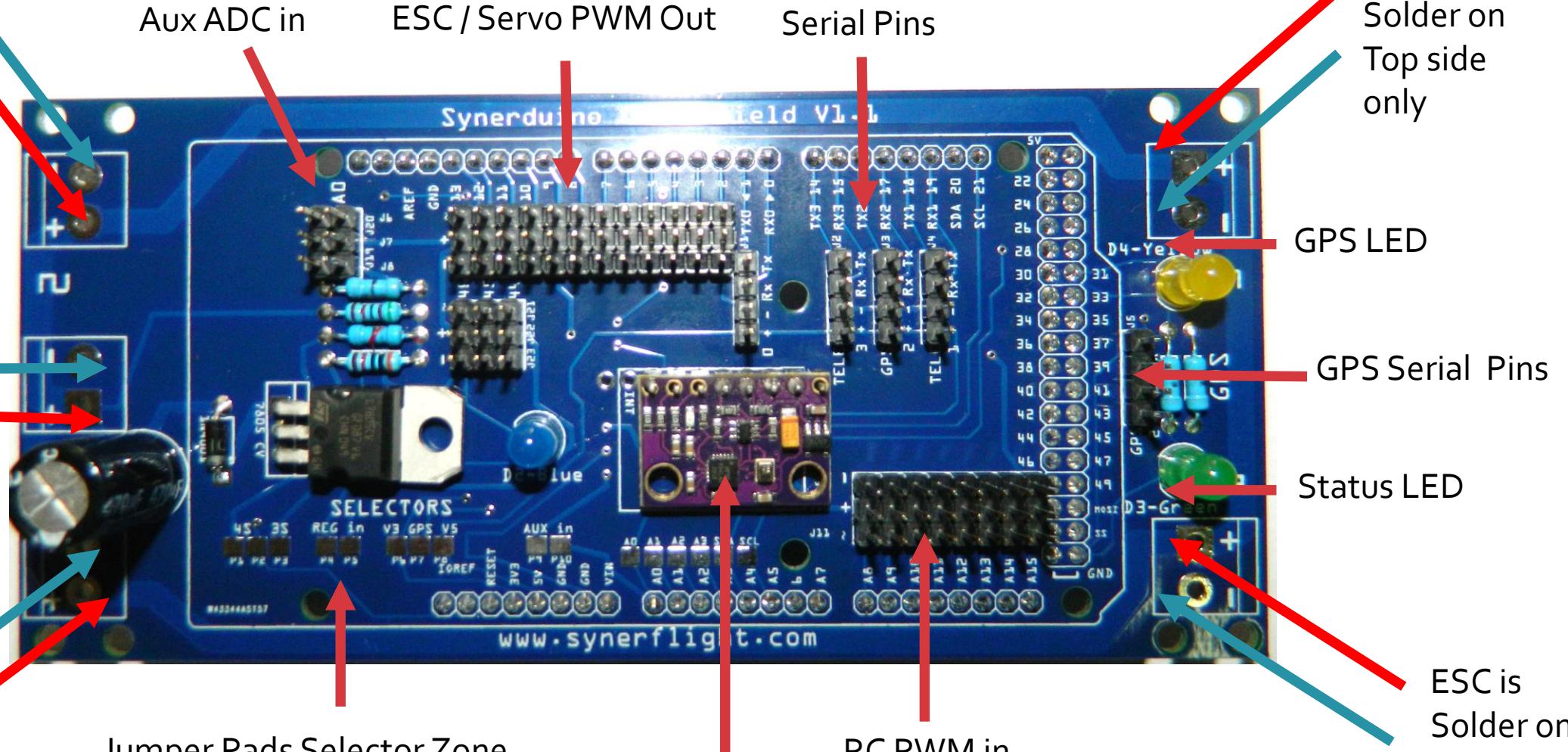


IMU : L3G4200D Gyro / ADXL345 Accelerometer / BMP180 – 85 Baro / MMC5883 Mag

Synerduino Kwad Shield V1.1 GY91

ESC is
Solder on
Top side
only

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

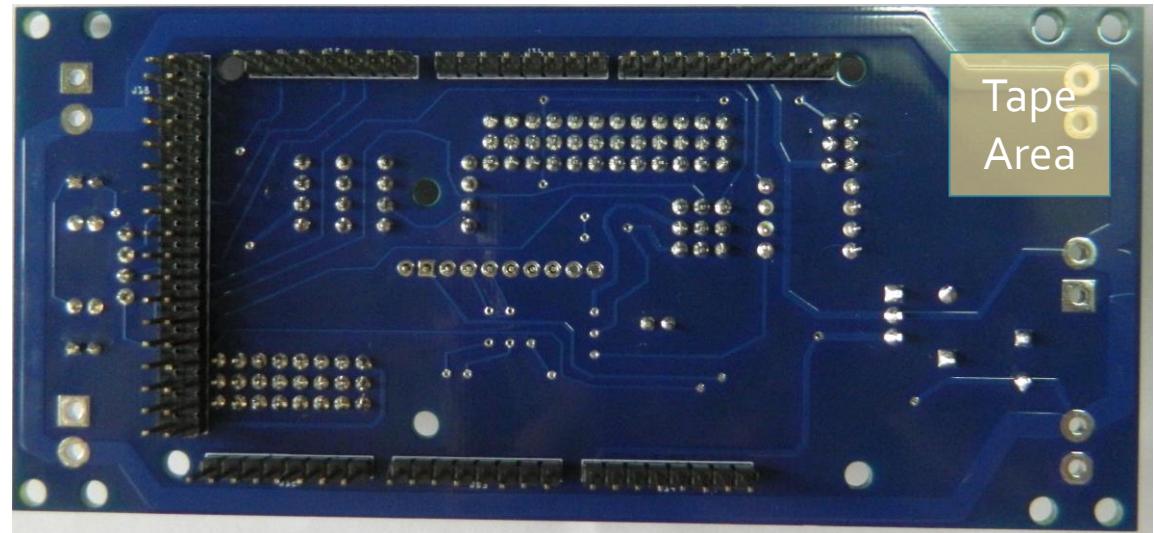
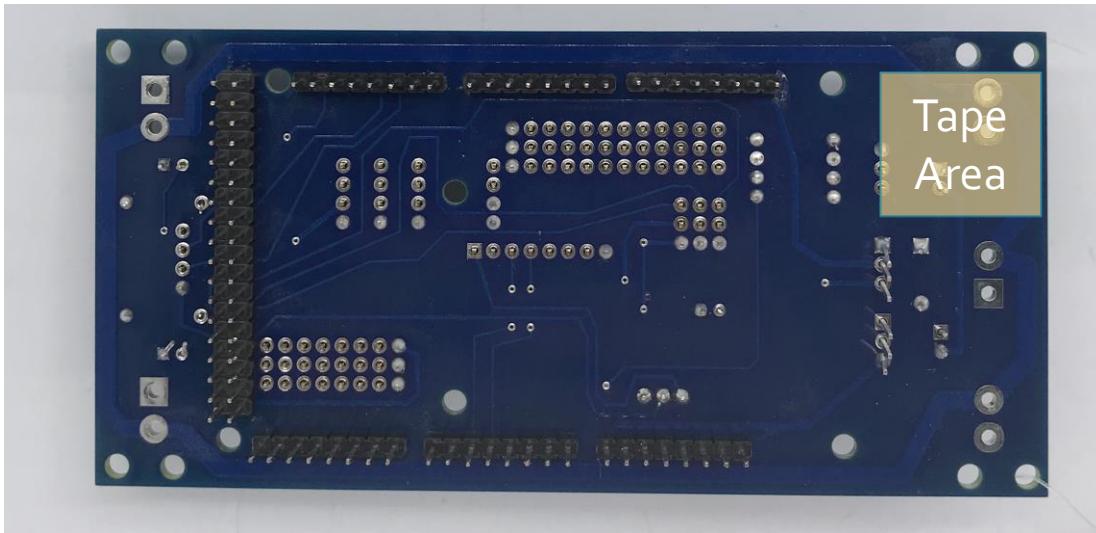


IMU : MPU-9250 & BMP280

Synerduino Kwad Shield Preparation

Ensure insulation from the Arduino board add tape on these areas

BETA GY801 & V1.1 GY91



Arduino Board Preparation

Ensure insulation from the Arduino board add tape on these areas

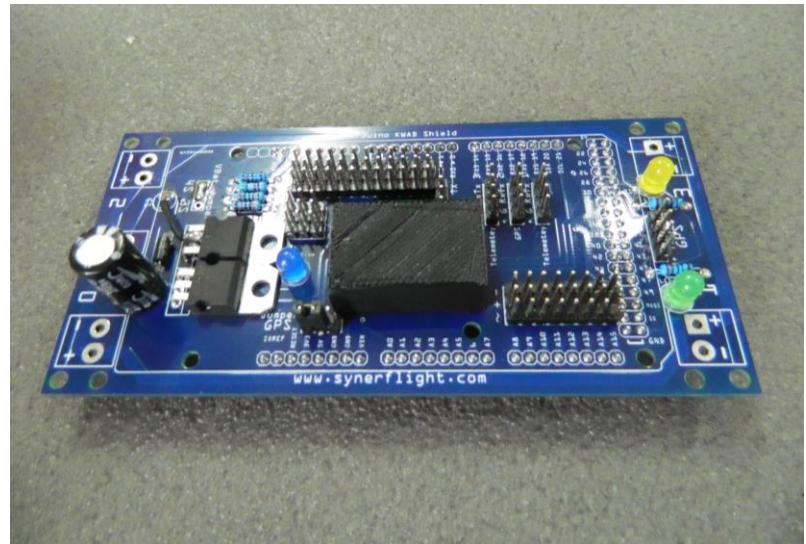


2560 MEGA



SMD UNO 328

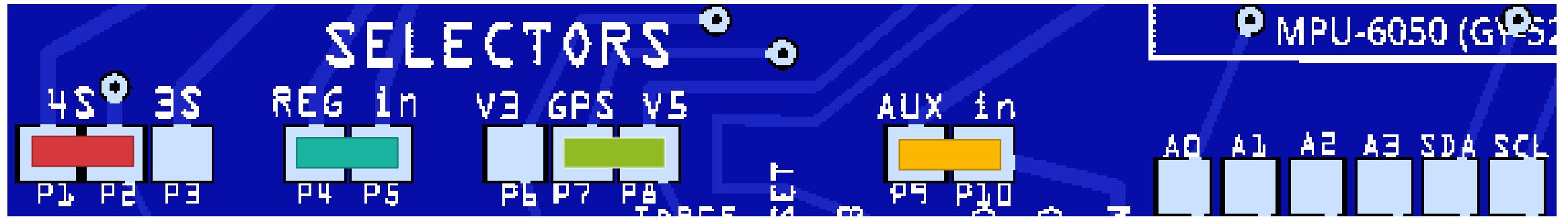
Use PVA to glue the cover to the IMU sensors seal the rim of the cover all the way



Synerduino Kwad Shield V1.1 (GY91)

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 o pin Auxin / Battery monitor

Analog o-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

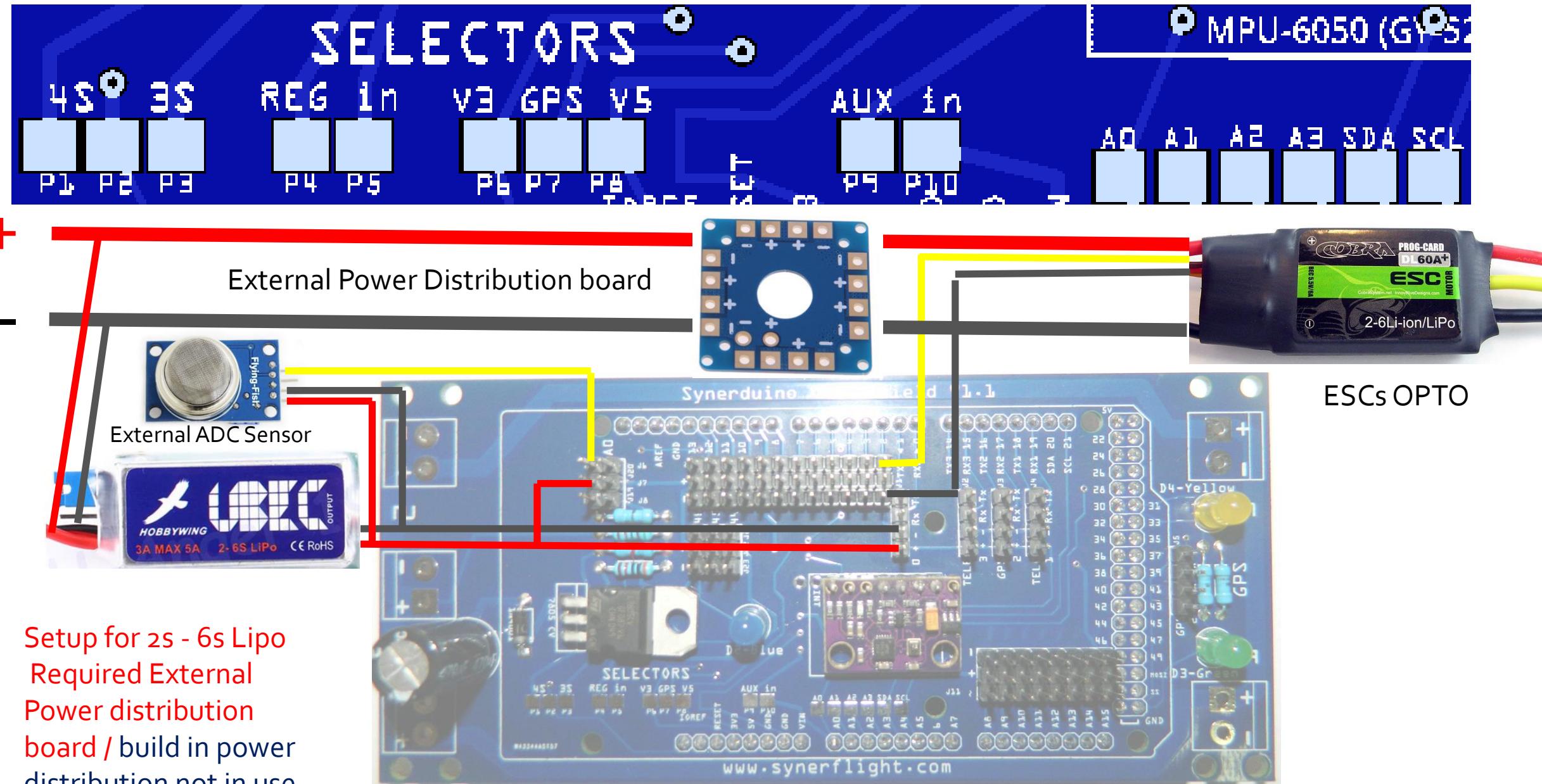
Reg In – short the Pads for using regulator to power and build in power distributor the synerduino and Arduino board

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

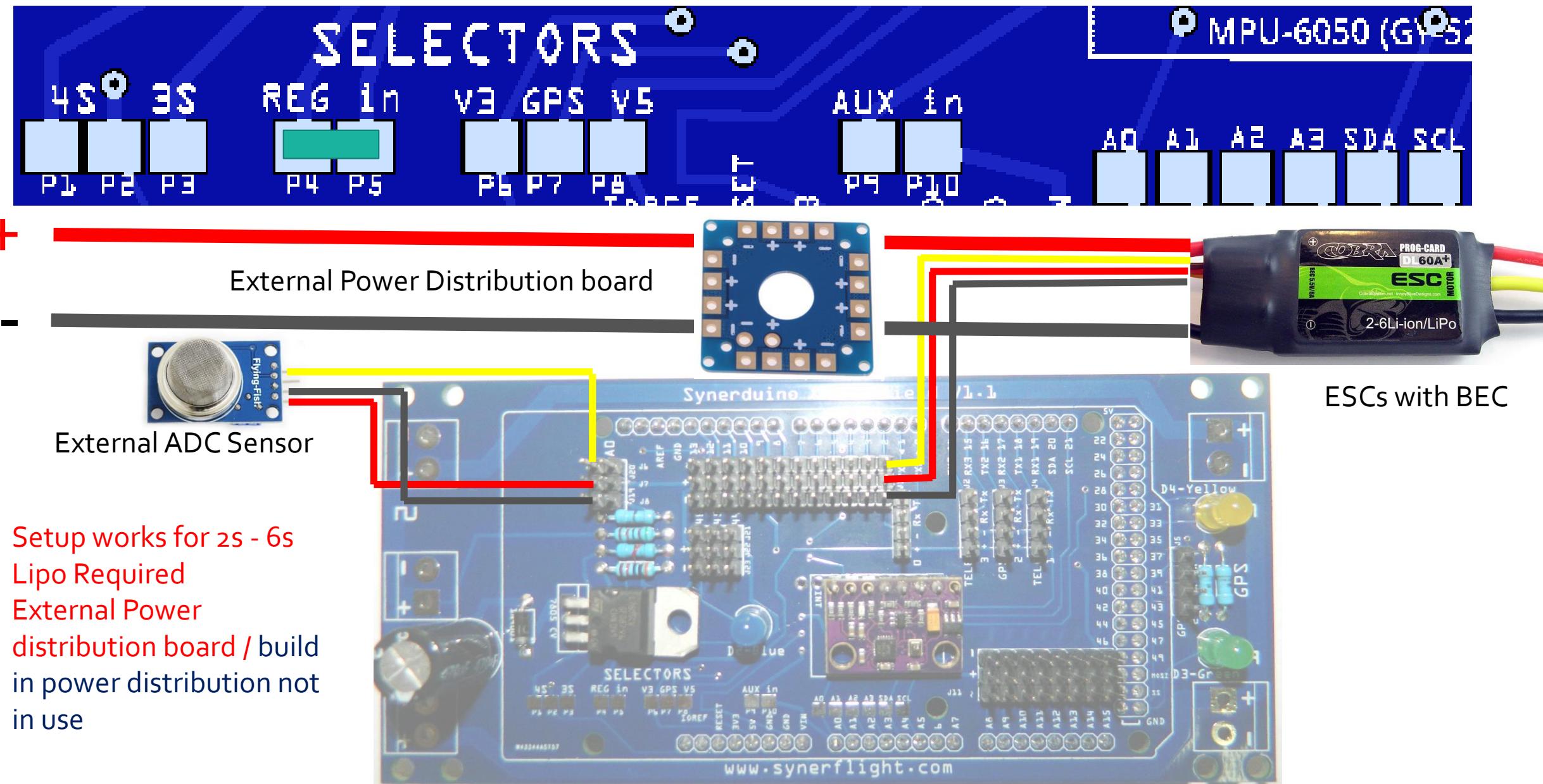
Aux in- leave it open for utilizing the Ao Pins for External ADC sensors / Short the Pads to use build in battery monitoring . Cell Selector must be set to 4s or 3s

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

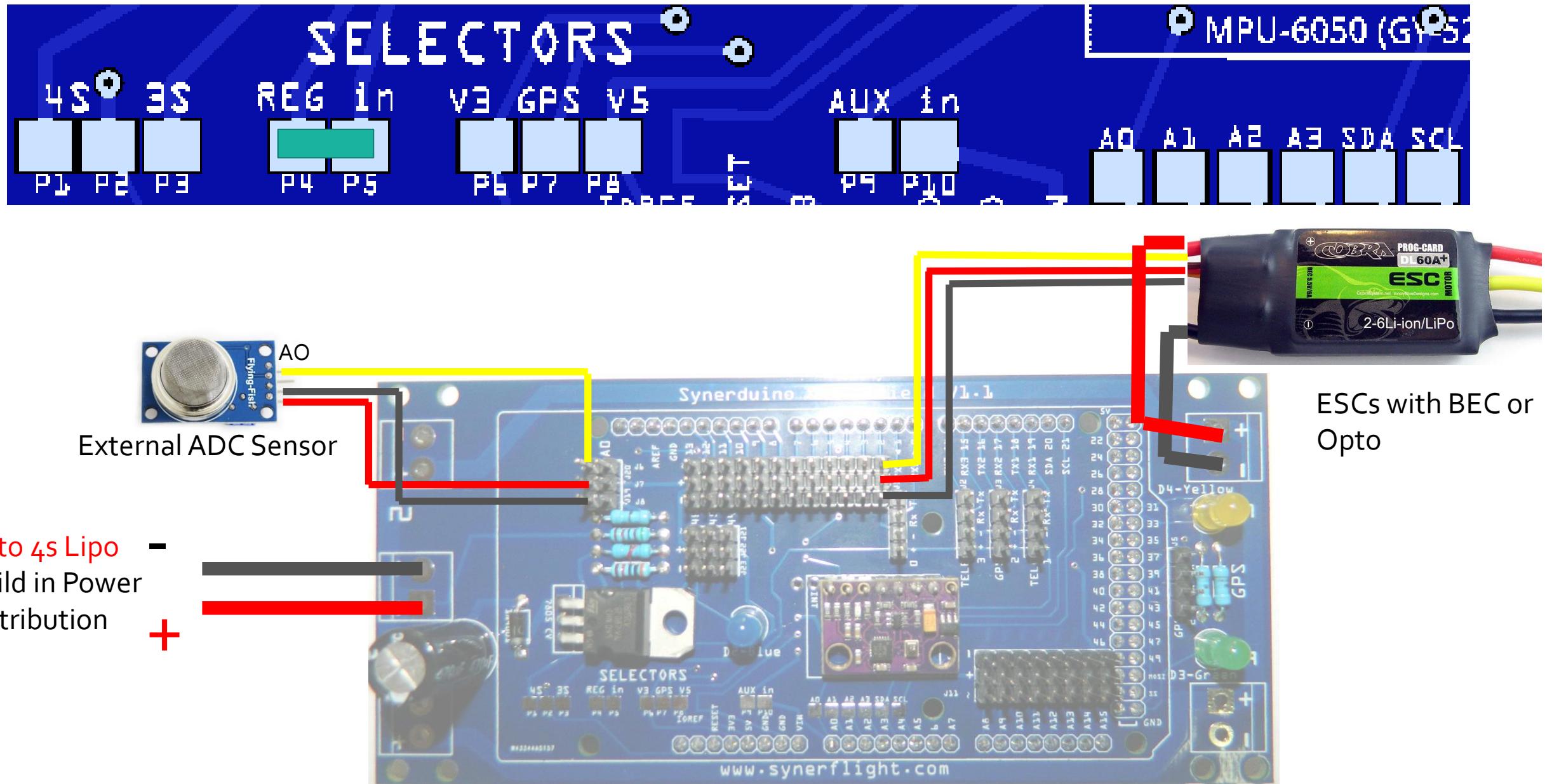
Default - Ao External ADC sensor , Require 5v UBEC applied to the 5V Serial pins & for OPTO ESC



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

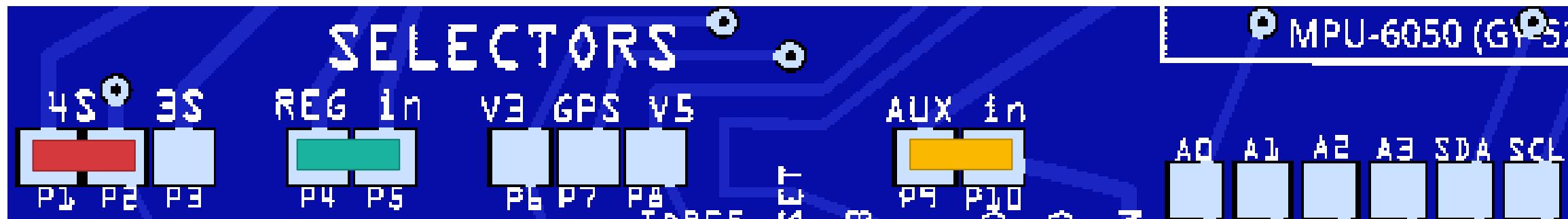


Reg in only - Ao External ADC sensor , ESC BEC or OPTO applied to the 5V PWM pins

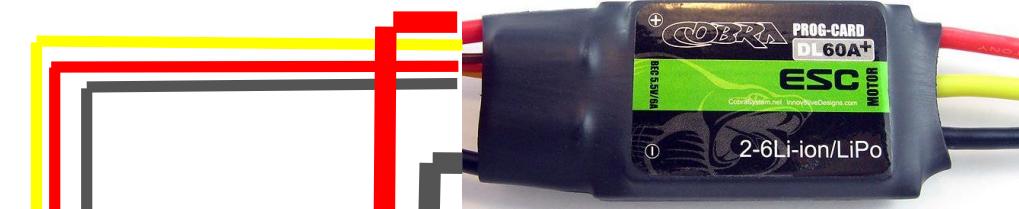


Recommended setup for beginner

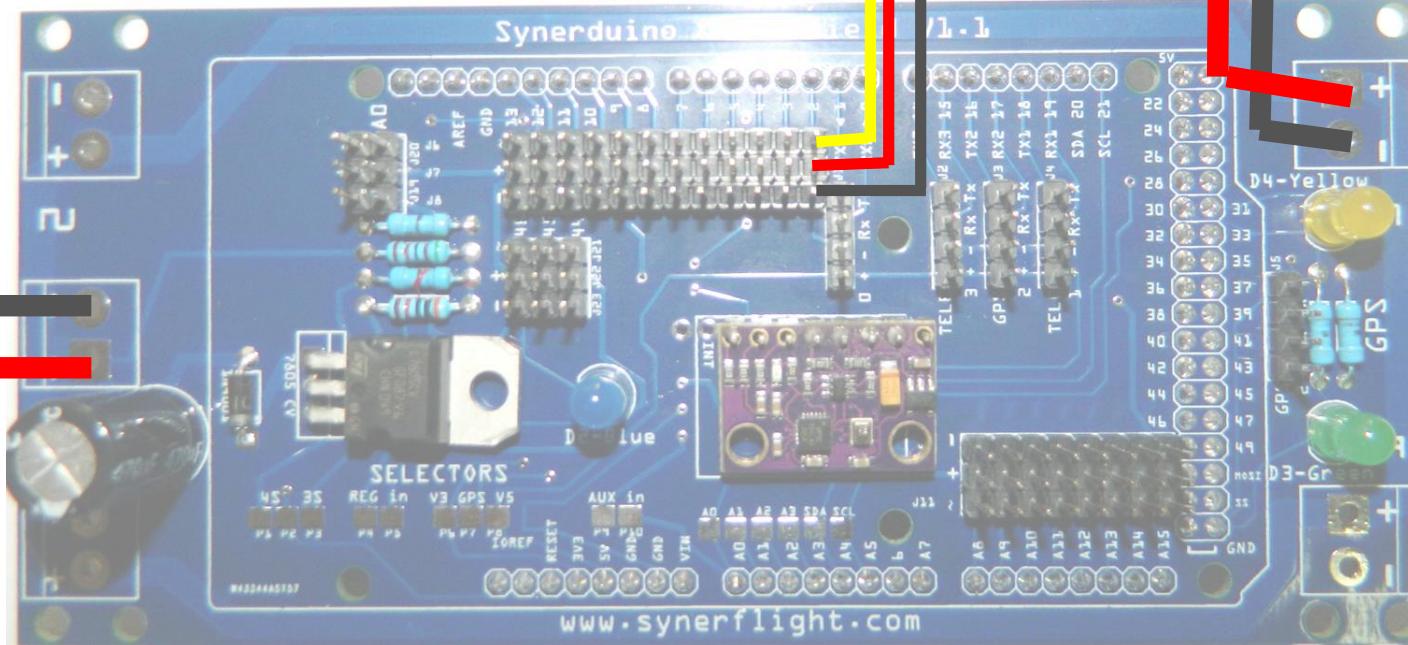
Reg & Vbat - Ao as Battery voltage monitor , ESC BEC or OPTO applied to the 5V PWM pins



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



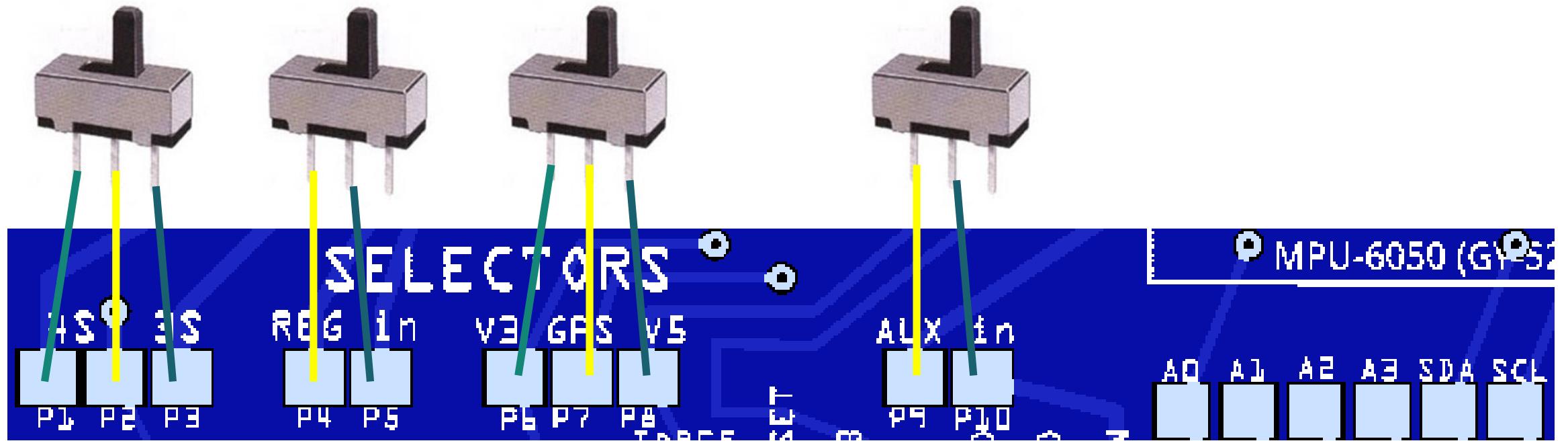
2s to 4s Lipo
Build in Power distribution

-



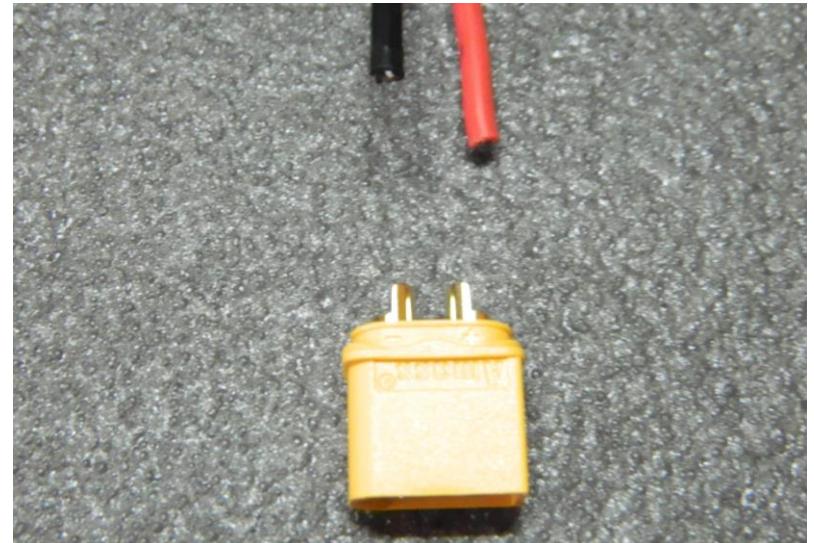
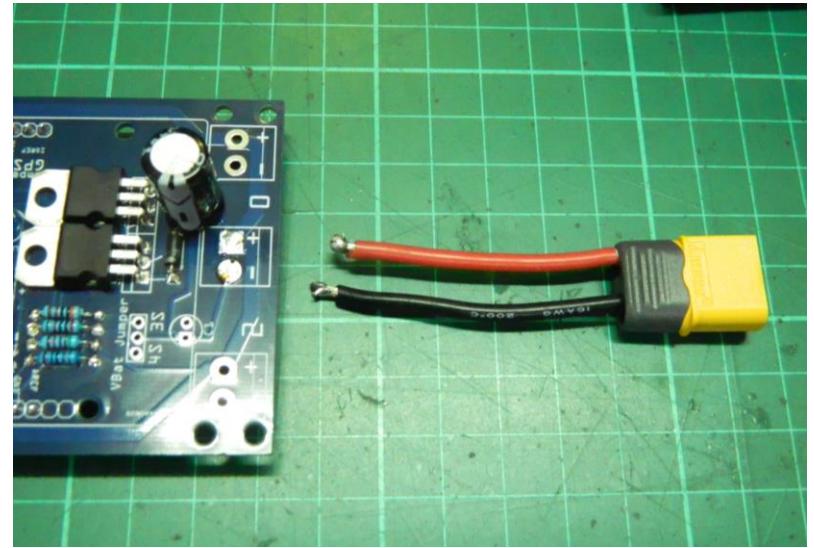
Recommended setup for beginner

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

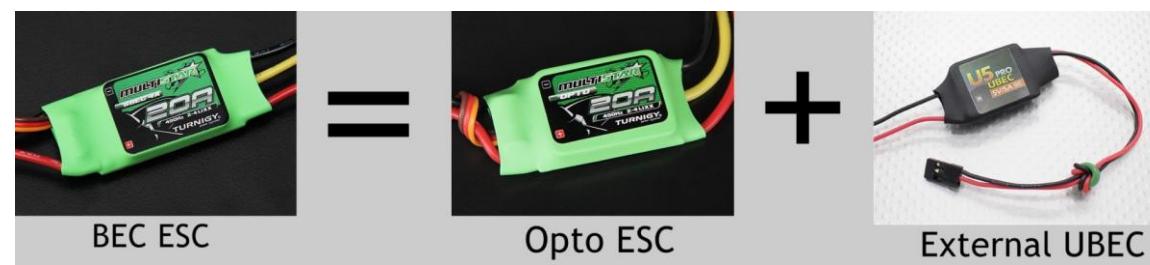


For those who would use the build in battery monitoring
circuit upto 4s lipo ensure the Cell Count and Aux in is jumped
before powering up

Pls check the polarity of the XT6oPlugs to match up with the board's polarity



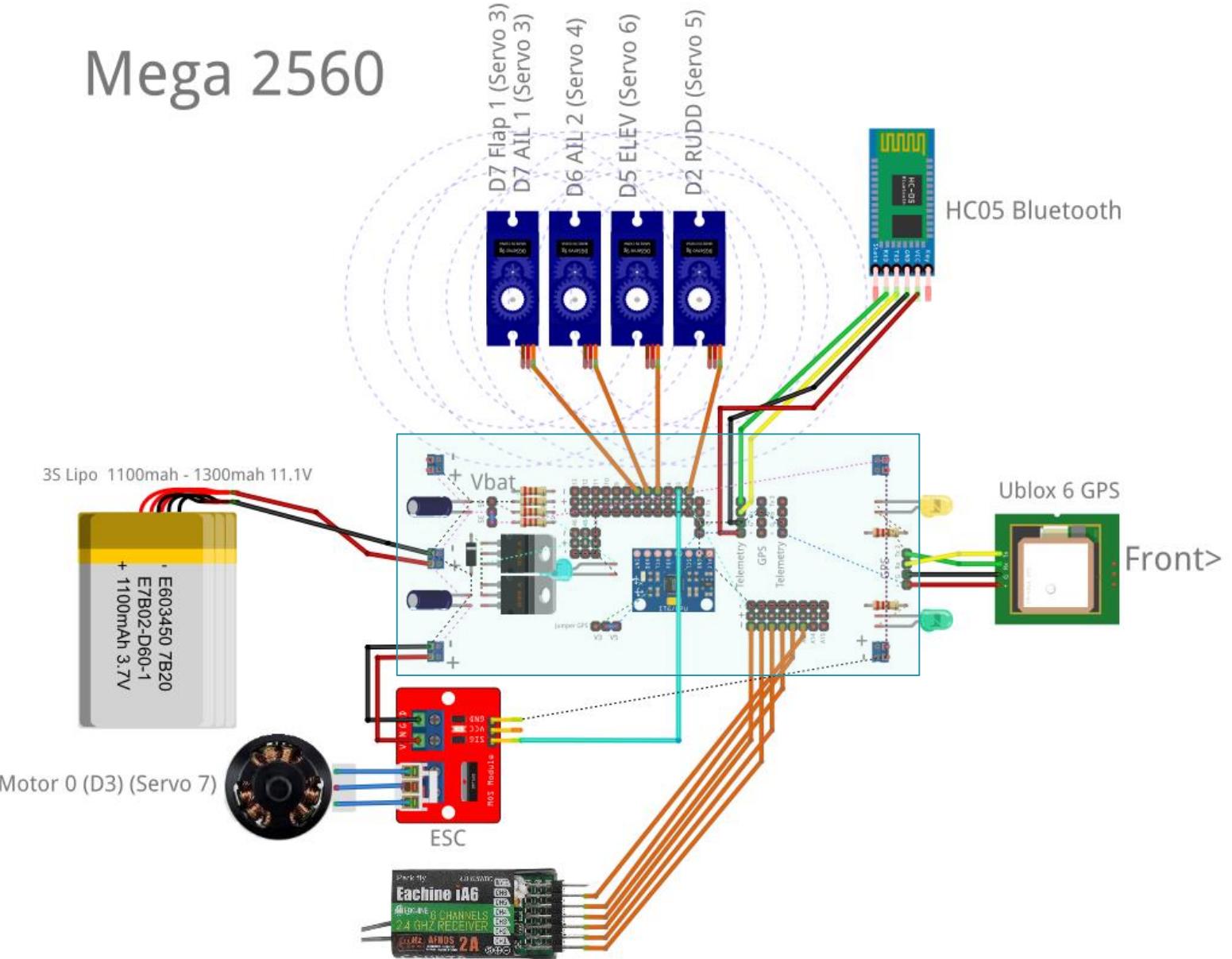
Electronic Speed Controller



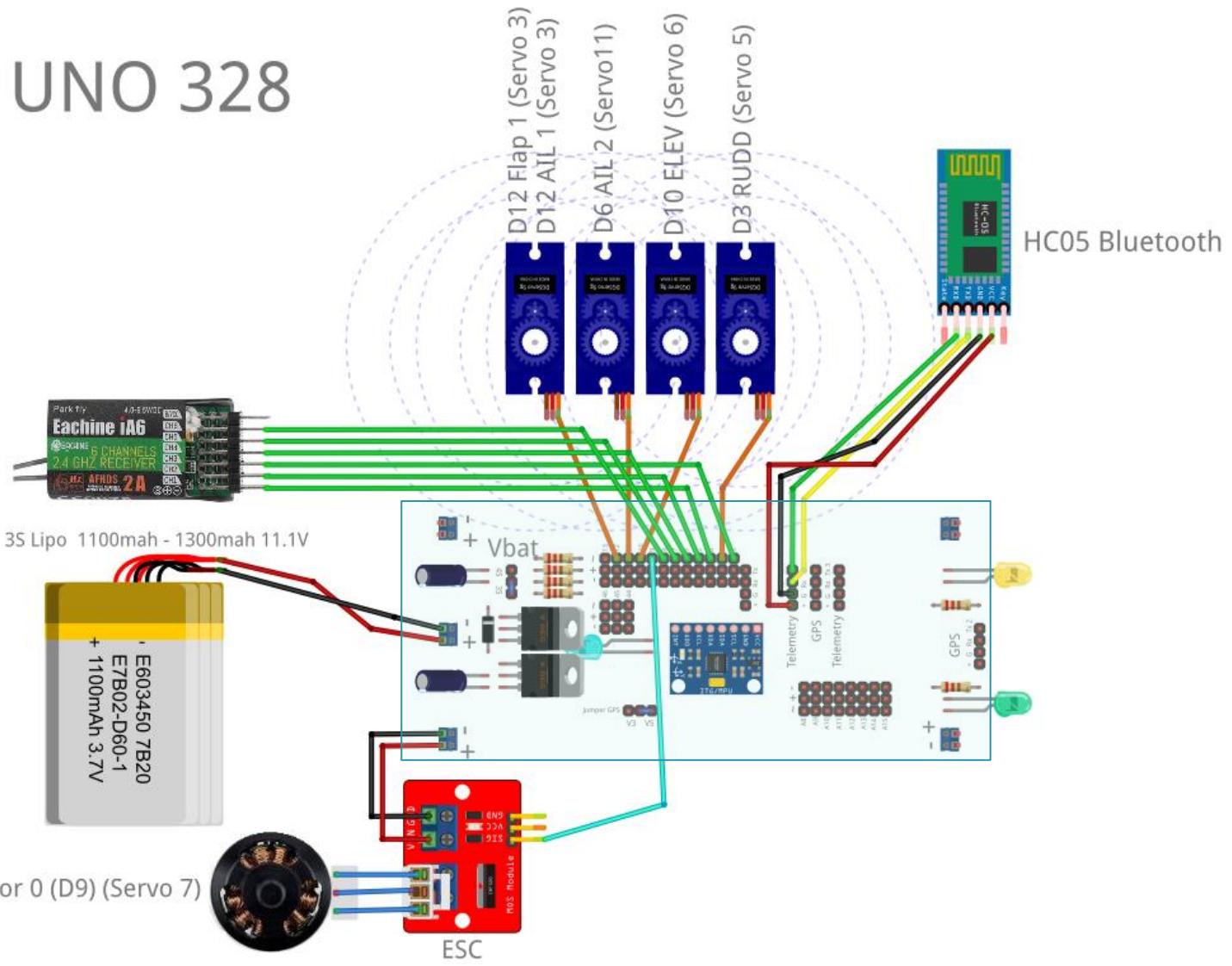
BEC "Universal Battery Elimination Circuit"

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

Mega 2560



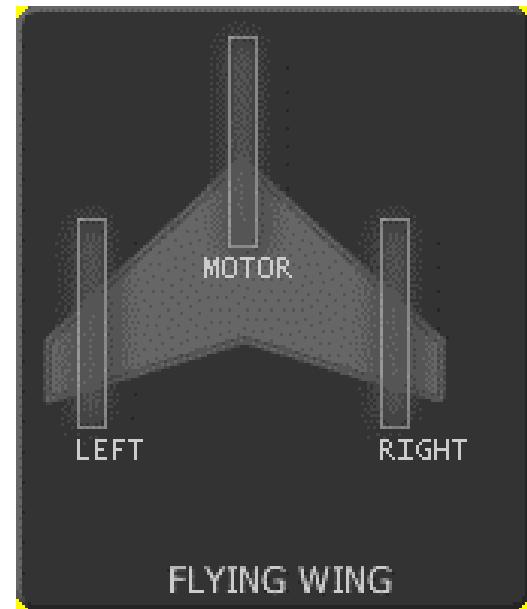
UNO 328



- ***Servos***
- Totally 8 servos is available.
- Default mix uses.
 - 2 Wing servos
 - 1 Elevator
 - 1 Rudder
 - 1 Throttle servo or esc.
- Ao & A1 can be used for a gimbal.
- A2 Is used for conventional Flaps or Camera trigger.

PWM Pins arrangement PWM Output

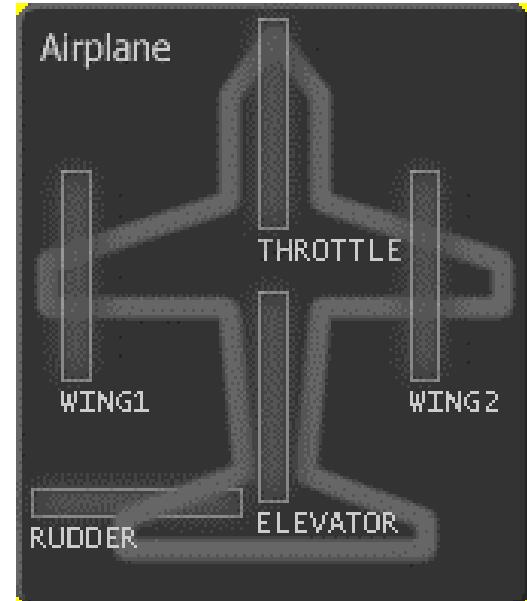
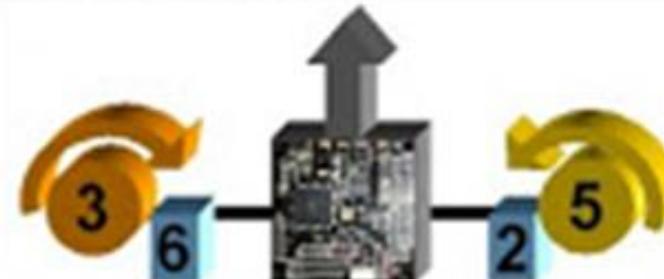
	MINI	MEGA	AirPlane	Heli-90	Heli-120	Engine nr:
servo[0] =	A0	D34/44	-	-	-	
servo[1] =	A1	D35/45	-	-	-	
servo[2] =	A2	D33/46	-	-	-	
servo[3] =	D12	D7/D37	Wing:1	Nckservo	Nckservo	
servo[4] =	D11	D6	Wing:2	Roll	Left	motor[3]
servo[5] =	D3	D2	Rudder	Tail	Tail	motor[2]
servo[6] =	D10	D5	Elev	Coll	Right	motor[1]
servo[7] =	D9	D3	Engine	Engine	Engine	motor[0]



for Arduino 328p:



for arduino mega:



PWM Pins arrangement PWM Output **use special airplane ino file** for this setup utilizing Bi copter special Plane types

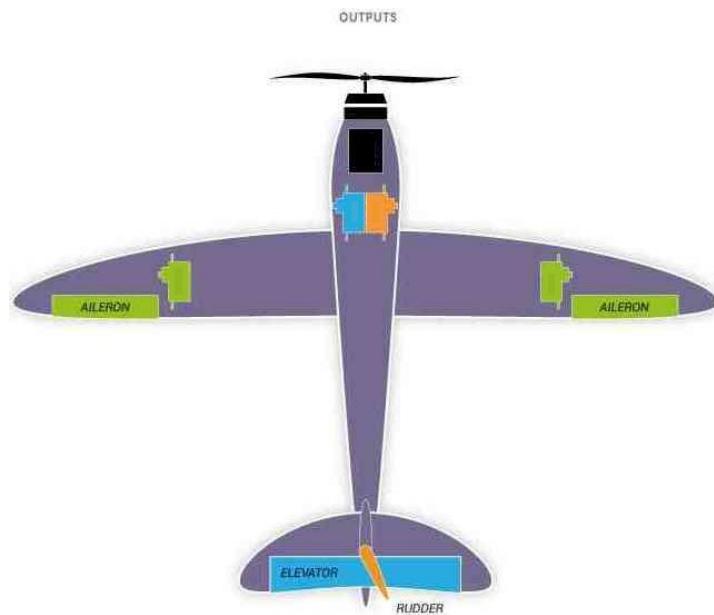
Differential Thrust Wing

Arduino UNO Plane AETR

Wing1	Wing2	Elevator	Rudder	Throttle
D12	D11	D10	D3	D9 motor

Arduino Mega Plane RET

Wing1	Wing2	Elevator	Rudder	Throttle
D7/37	D6	D5	D2	D3 motor



Flying Wings have very similar arrangement except Wing 1 and 2 are Elevons

Note: check if your RC servo needs reversing on FC config or on your Transmitter to ensure out put of differential respond correctly
Some models of transmitter may need RC output reversing to operate correctly

PWM Pins arrangement PWM Output **use special airplane ino file** for this setup utilizing Bi copter special Plane types

Differential Thrust Wing

Arduino UNO Plane RET

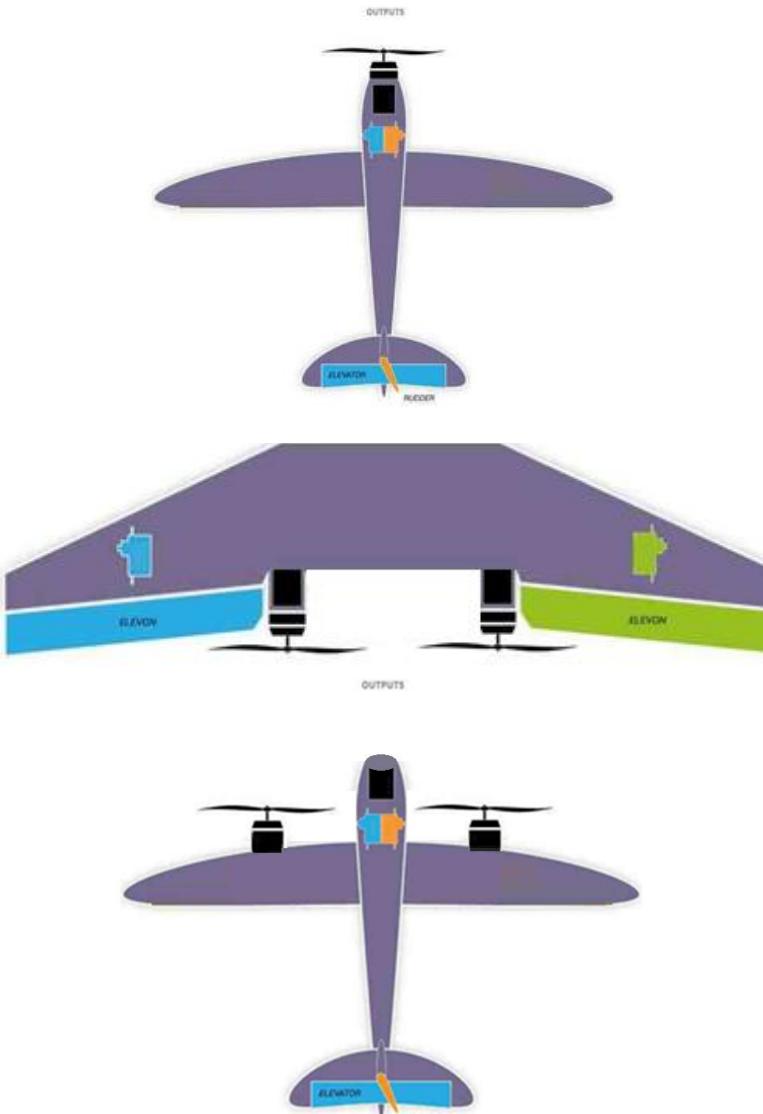
Elevator	Rudder	Throttle
D11	D3	D9 motor D10 motor

Arduino UNO Diff Thrust Wing

Wing1	Wing2	Throttle
D11	D3	D9 Left motor D10 Right motor

Arduino Uno Diff Thrust Plane

Elevator	Rudder	Throttle
D11	D3	D9 Left motor D10 Right motor



Arduino Mega Plane RET

Elevator	Rudder	Throttle
D6	D2 Aileron	D3 motor 5 motor

Arduino Mega Diff Thrust Wing

Wing1	Wing2	Throttle
D6	D2	D3 motor 5 motor

Arduino Mega Diff Thrust Plane

Elevator	Aileron	Throttle
D6	D2	D3 Left motor D5 Right motor

Note: check if your RC servo needs reversing on FC config or on your Transmitter to ensure out put of differential respond correctly
Some models of transmitter may need RC output reversing to operate correctly

Output.cpp

Main Mix table should you need to custom mix your plane (Define {Bi})

```
1129 #if defined(DYNBALANCE)
1130     return;
1131 #endif
1132 #define PIDMIX(X,Y,Z) rcCommand[THROTTLE] + axisPID[ROLL]*X + axisPID[PITCH]*Y + YAW_DIRECTION * axisPID[YAW]*Z
1133 #define SERVODIR(n,b) ((conf.servoConf[n].rate & b) ? -1 : 1)
1134
1135 //***** main Mix Table *****
1136 #if defined(MY_PRIVATE_MIXING )
1137 #include MY_PRIVATE_MIXING
1138 #elif defined( BI )
1139 //motor[0] = PIDMIX(+1, 0, 0); //LEFT
1140 //motor[1] = PIDMIX(-1, 0, 0); //RIGHT
1141 //servo[4] = (SERVODIR(4,2) * axisPID[YAW]) + (SERVODIR(4,1) * axisPID[PITCH]) + get_middle(4); //LEFT
1142 //servo[5] = (SERVODIR(5,2) * axisPID[YAW]) + (SERVODIR(5,1) * axisPID[PITCH]) + get_middle(5); //RIGHT
1143 //***** 3ch Airplane Throttle Elevator Rudder *****
1144 motor[0] = PIDMIX(0, 0, 0); //LEFT
1145 motor[1] = PIDMIX(0, 0, 0); //RIGHT
1146 servo[4] = (SERVODIR(4, 1) * axisPID[YAW]) + get_middle(4); //RUDDER
1147 servo[5] = (SERVODIR(5, 1) * axisPID[PITCH]) + get_middle(5); //ELEVATOR
1148 //***** BI Airplane Elevator Ailron *****
1149 //motor[0] = PIDMIX(0, 0, +1); //LEFT
1150 //motor[1] = PIDMIX(0, 0, -1); //RIGHT
1151 //servo[4] = (SERVODIR(4,1) * axisPID[ROLL]) + get_middle(4); //AILERON
1152 //servo[5] = (SERVODIR(5,1) * axisPID[PITCH]) + get_middle(5); //ELEVATOR
1153 //***** BI Wing Elevon *****
1154 //motor[0] = PIDMIX(0, 0, +1); //LEFT
1155 //motor[1] = PIDMIX(0, 0, -1); //RIGHT
1156 //servo[4] = (SERVODIR(4,2) * axisPID[ROLL]) + (SERVODIR(4,1) * axisPID[PITCH]) + get_middle(4); //LEFT
1157 //servo[5] = (SERVODIR(5,2) * axisPID[ROLL]) + (SERVODIR(5,1) * axisPID[PITCH]) + get_middle(5); //RIGHT
1158
```

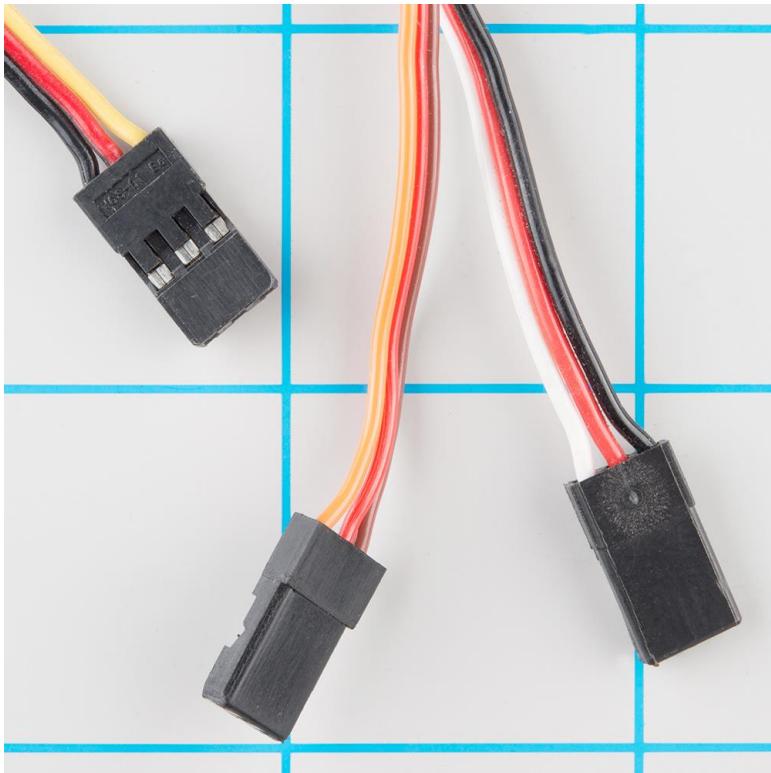
PWM INPUT Assignment

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



~ + -

End view

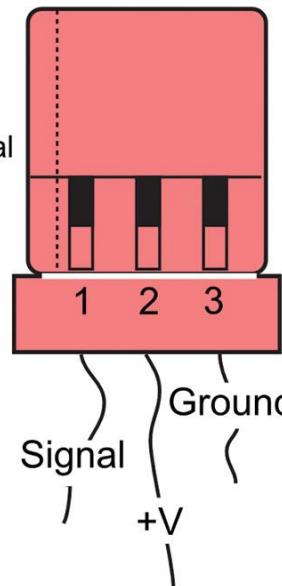


J-type (Futaba)



S-type (Hitec, JR)

Keyway =
signal terminal



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

OPTO Wires may only have
Signal and Negative Wires o



PWM RECEIVER

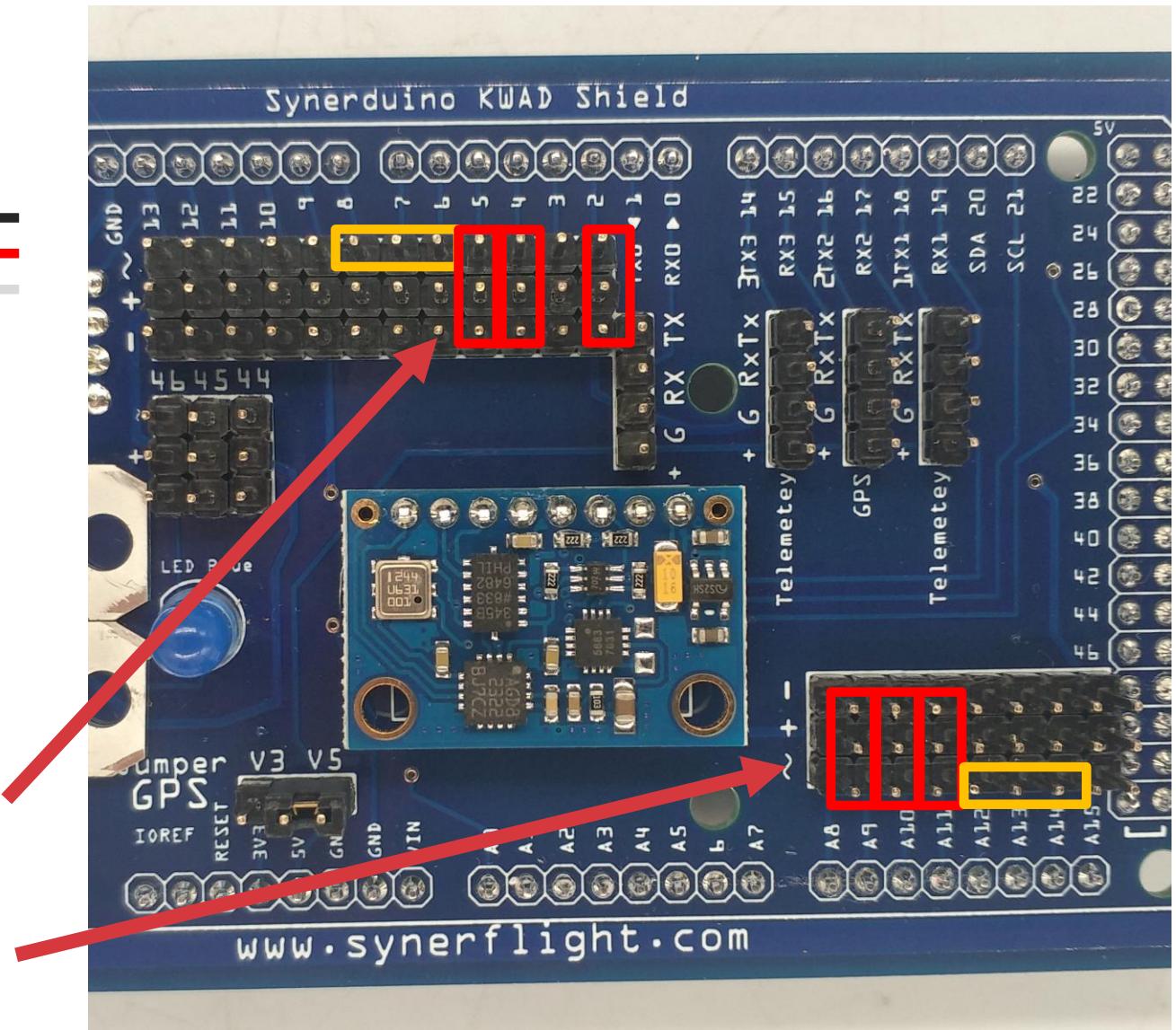


IN CASE YOU'D ASK WHY THE SERVO CONNECTORS WERE DONE THIS WAY , ITS SIMPLY YOU DON'T NEED TO PLUG ALL THE PWM POWER RAILS ON ALL CHANNELS YOU JUST NEED THE PWM SIGNAL PIN ALONE

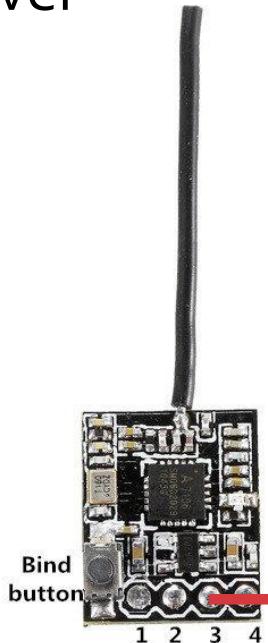
MOSTLY RUDDER AUX1 AND AUX2

UNO PWM IN

MEGA PWM IN



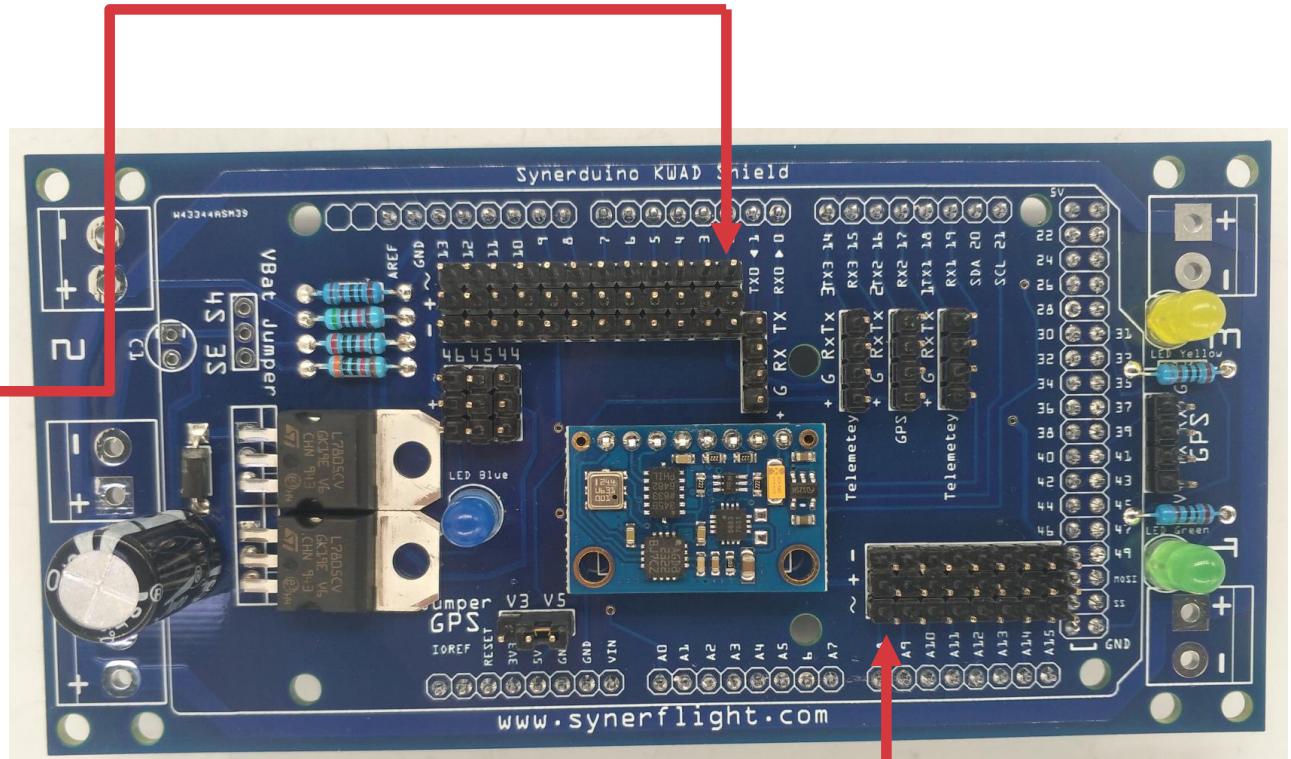
PPM Receiver



Pin D2 for Uno



Pin A8 for Mega



TELEMETRY RADIO



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



38400 FOR XBEE RADIO

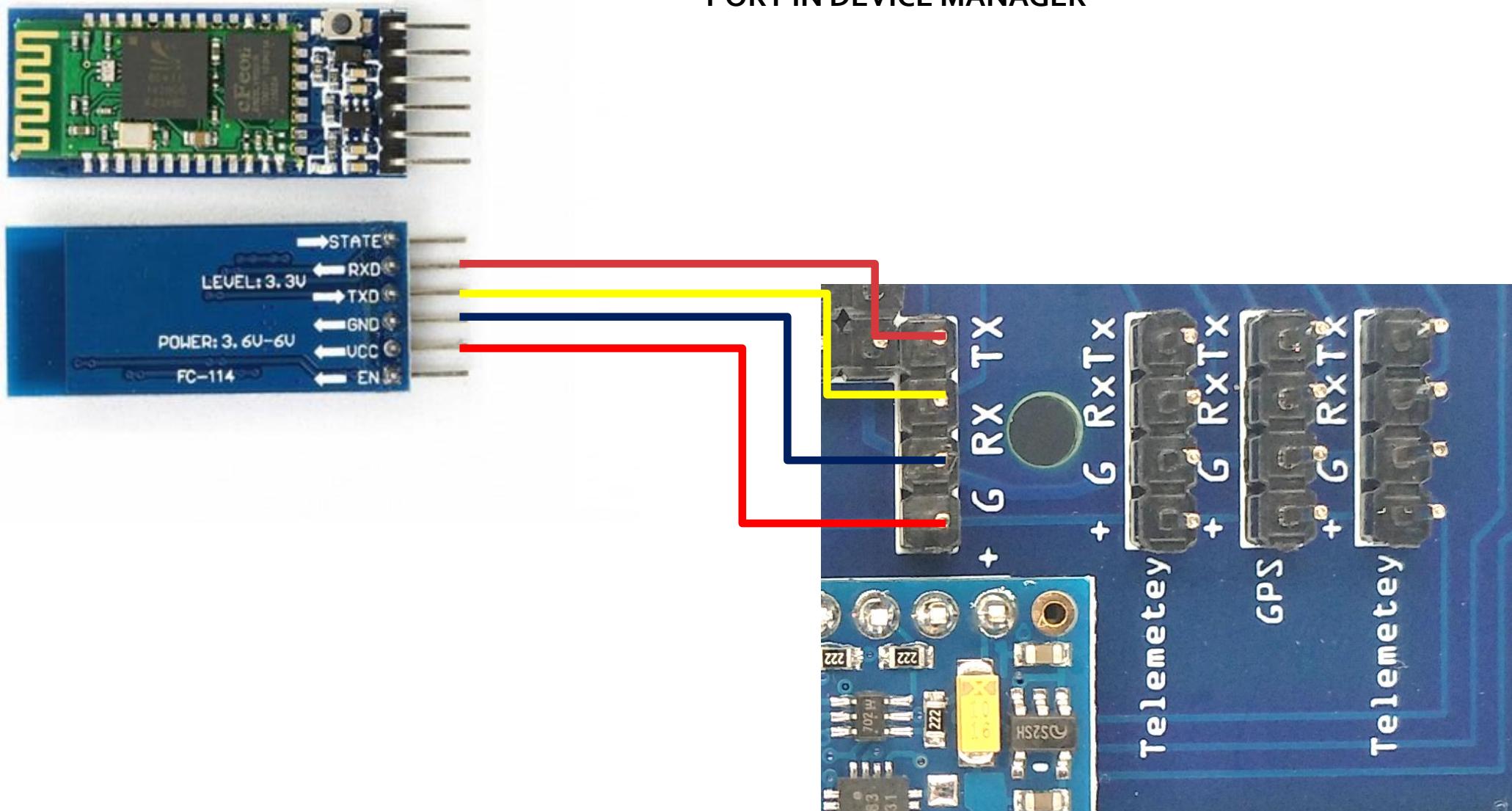


115200 FOR BLUETOOTH HC-05

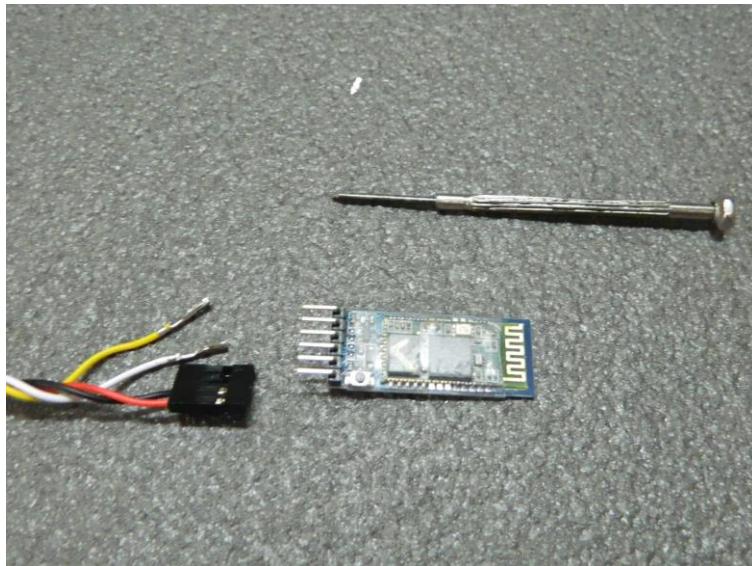
STANDARD FOR ALL DRONES TO USE SERIAL LINK AS TELEMETRY MAINLY ON YOUR TX RX SERIAL PORTS , NOTE: THE LOWER THE FREQUENCY OF THE RADIO THE LOWER THE BAUD IS NEEDED
PROTOCOL IS MSP RAW OR MAVLINK

Bluetooth

NOTE: USING o 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

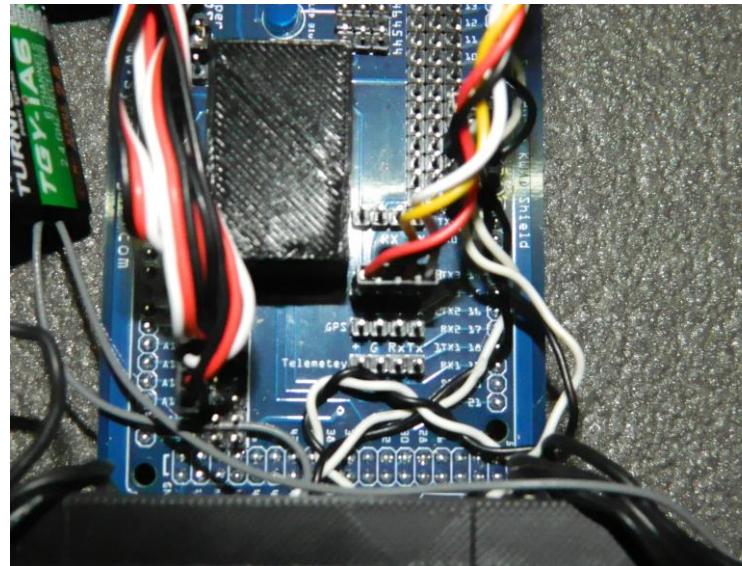


Bluetooth



**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

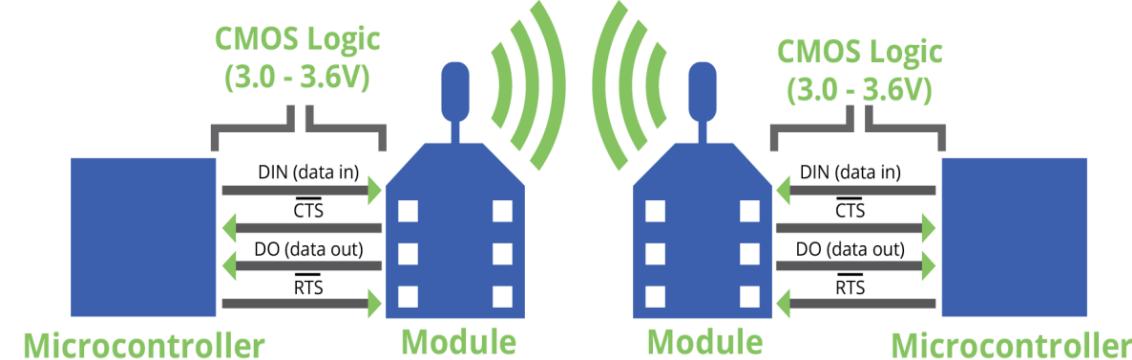
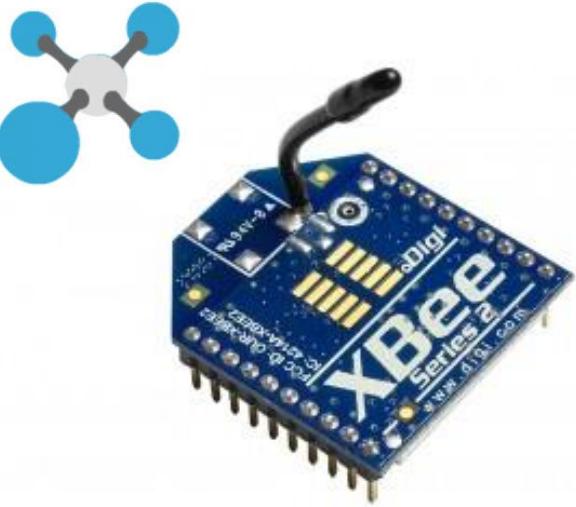


SEE TO IT THE WIRES COLOR CODE MATCHES THE
MARKINGS

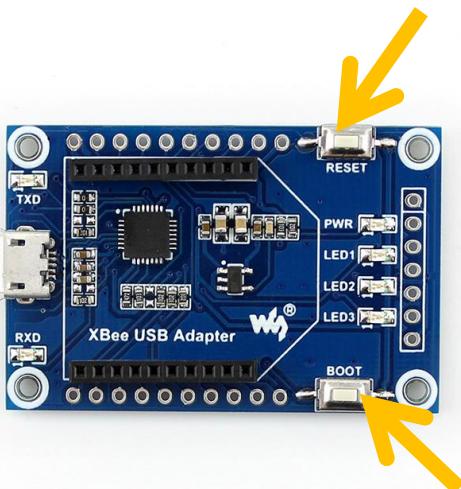
IMPROPER INSTALLATION MAY CAUSE DAMAGE
TO THE ARDUINO BOARD AND SHIELD DUE TO
REVERSE POLARITY

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

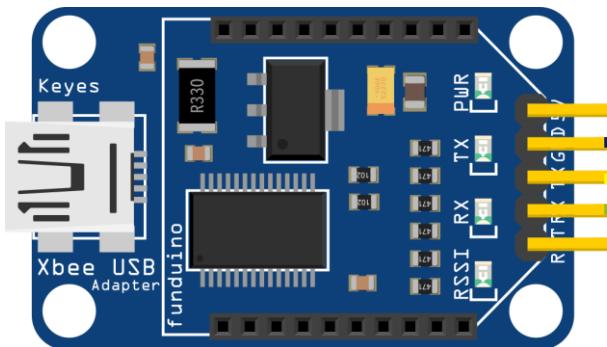
XBEE RADIO



38400 FOR XBEE RADIO



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

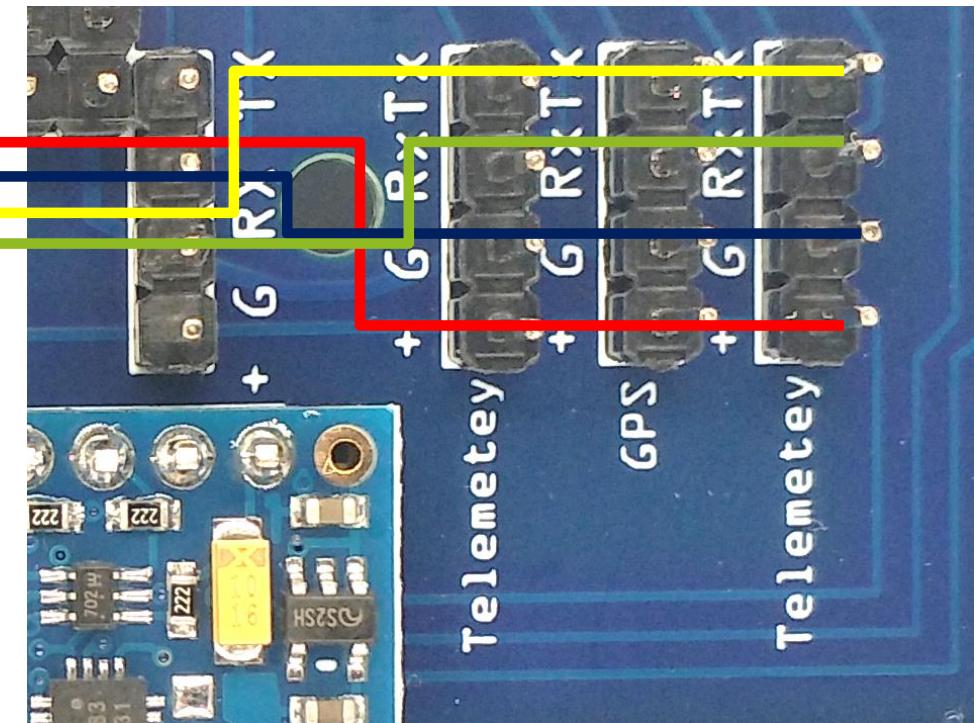


VCC >> +

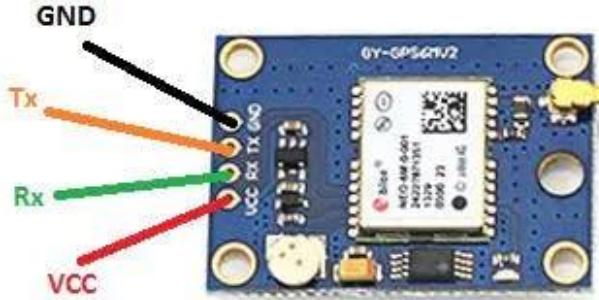
GND >> G

TX >> TX

RX >> RX



GPS



U BLOX NEO 6

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

ATTENTION:

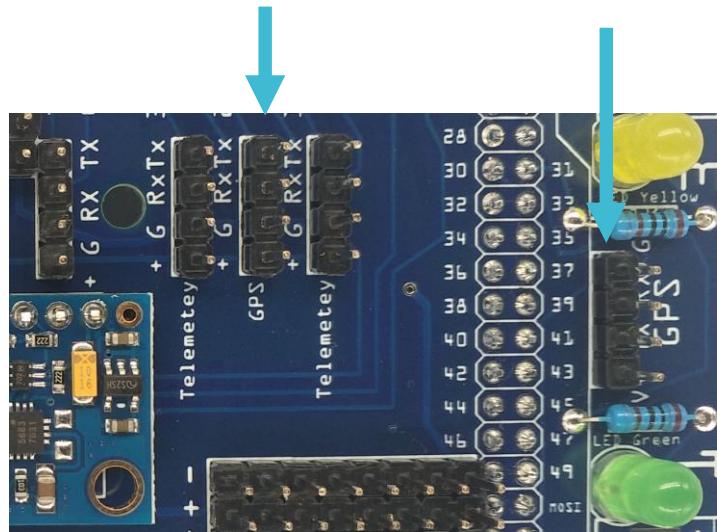
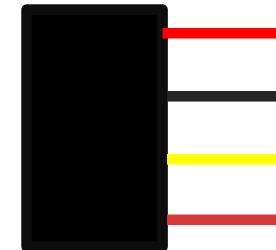
YOU MAY NEED TO REARRANGE THE HEADERS TO CONNECT THE GPS MODULE TO THE SHIELD BOARD ACCORDINGLY

VCC >> +

GND >> G

TX >> RX

RX >> TX



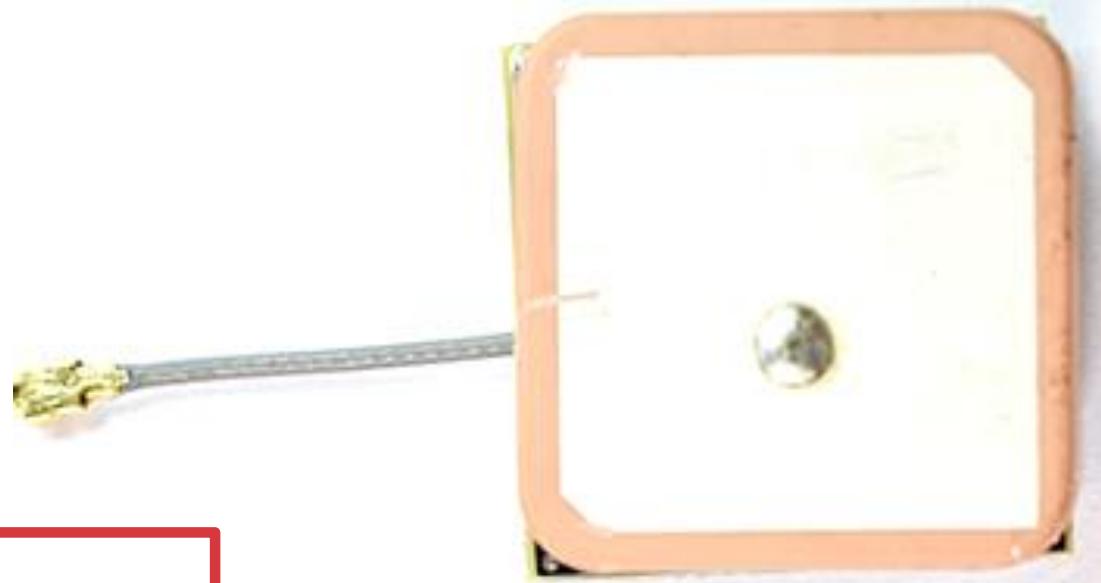
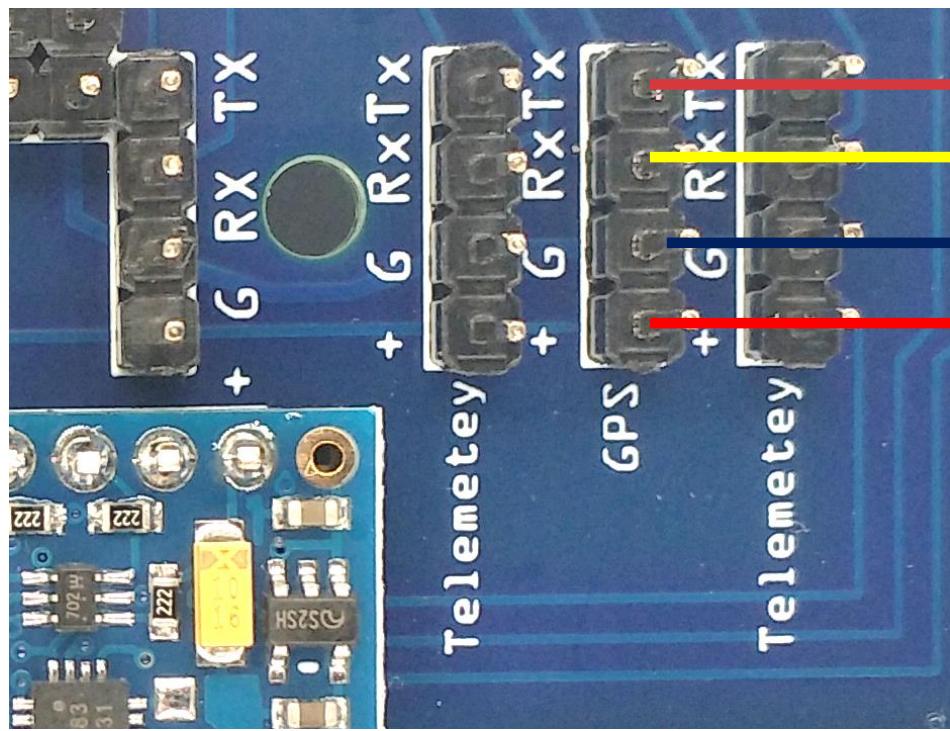
SEE TO IT THE WIRES COLOR CODE MATCHES THE MARKINGS

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

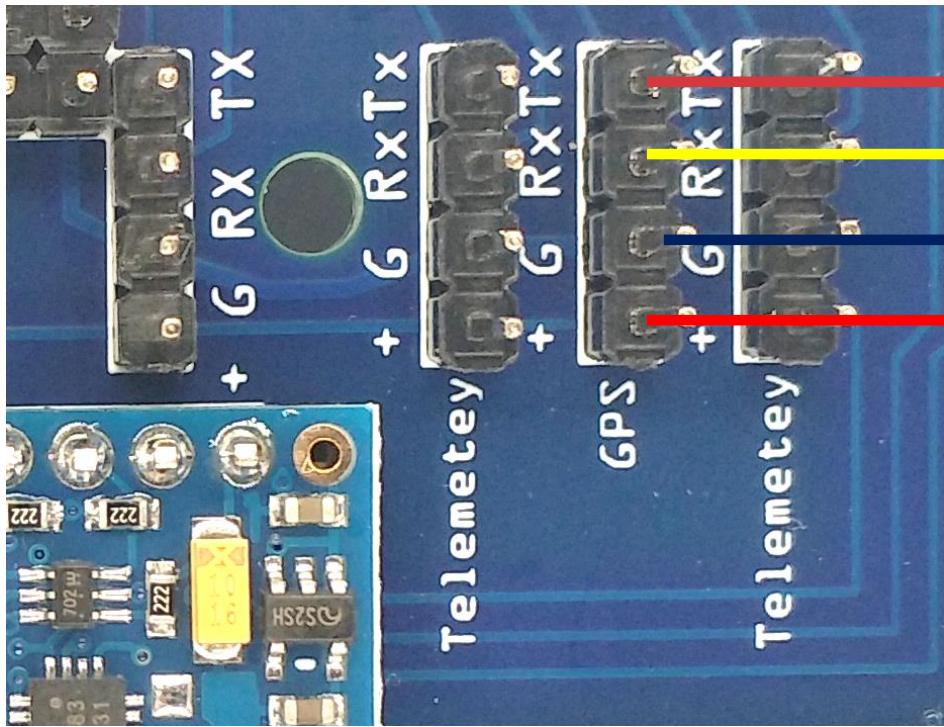
NOTE: YOU MAY NEED TO RE-SECURE THE GPS ANTENNA PATCH AGAIN WITH DOUBLE SIDED TAPE WHEN NECESSARY AS THE MODULE CAME IN WITH A TEMPORARY TAPE

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

GPS



GPS

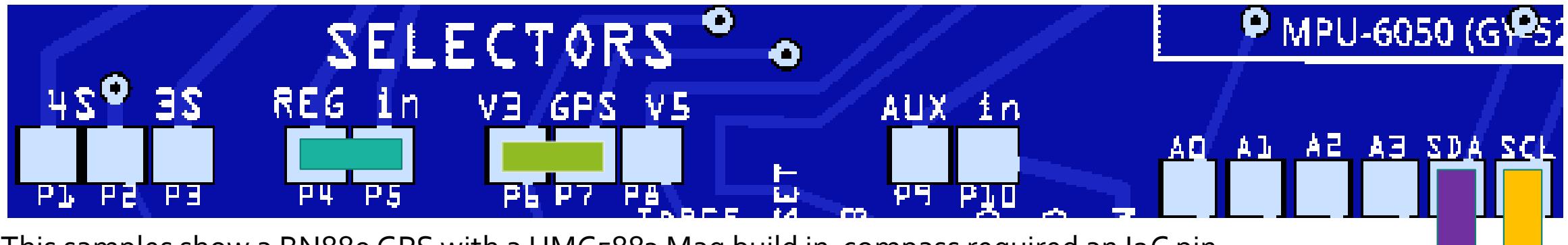


4.VCC
3.RX
2.TX
1.GND

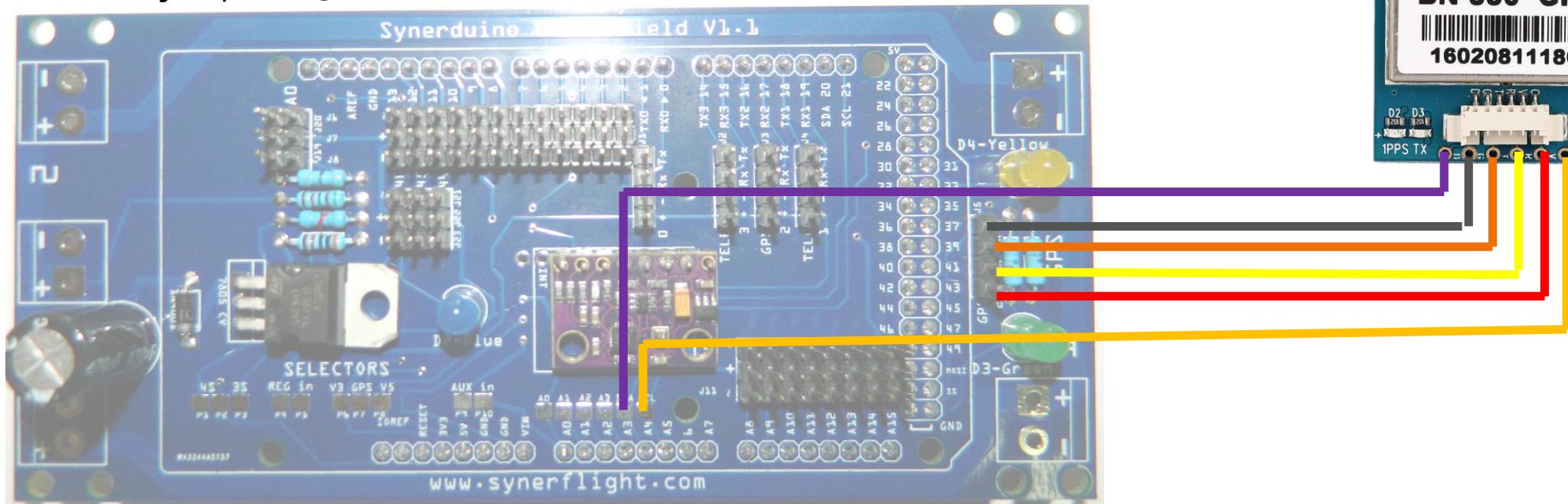
PIN	PIN Name	I/O	Description
1	GND	G	Ground
2	TX	O	Serial Data Output.
3	RX	I	Serial Data Input.
4	VCC	I	DC 3.0V - 5.5V supply input,Typical: 5.0V



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 IDE

Application Needed

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



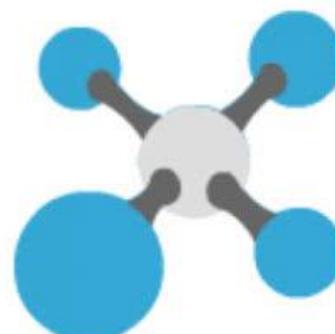
<http://synerflight.com/flywiogui/>

FLYWII GUI & Arduino Drone Multiwii firmware



UBLOX Configuration Platform

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



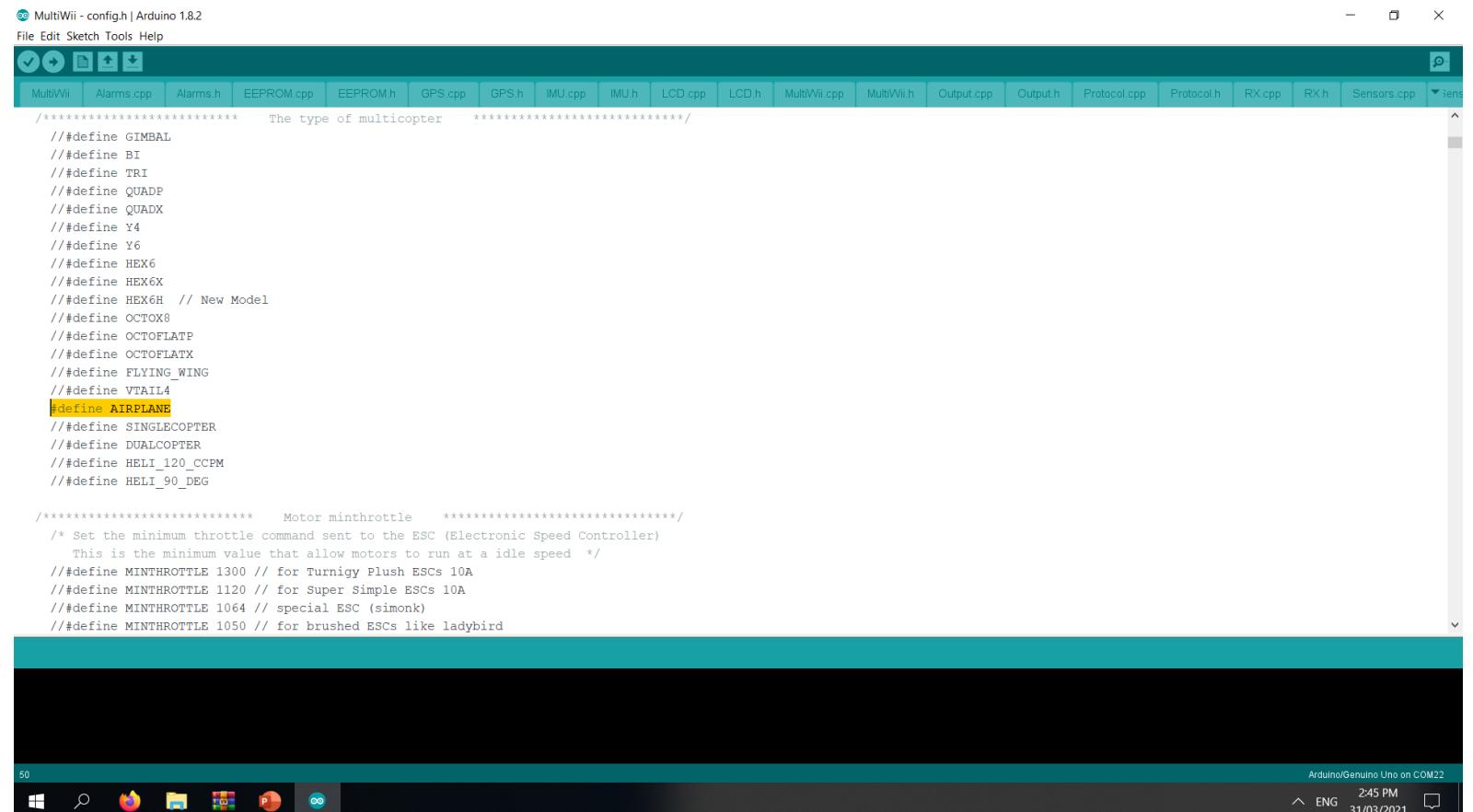
<https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu>

XCTU Configuration Platform

Config.h

Uncomment Define Airplane

This puts the vehicle into
Airplane mode



```
MultiWii - config.h | Arduino 1.8.2
File Edit Sketch Tools Help
Multivii 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 sens
// The type of multicopter
#define GIMBAL
#define BI
#define TRI
#define QUADP
#define QUADX
#define Y4
#define Y6
#define HEX6
#define HEX6X
#define HEX6H // New Model
#define OCTOX8
#define OCTOFLATP
#define OCTOFLATK
#define FLYING_WING
#define VTAIL4
#define AIRPLANE
#define SINGLECOPTER
#define DUALCOPTER
#define HELI_120_CCPM
#define HELI_90_DEG

// Motor minthrottle
/* Set the minimum throttle command sent to the ESC (Electronic Speed Controller)
   This is the minimum value that allow motors to run at a idle speed */
#define MINTHROTTLE 1300 // for Turnigy Plush ESCs 10A
#define MINTHROTTLE 1120 // for Super Simple ESCs 10A
#define MINTHROTTLE 1064 // special ESC (simonk)
#define MINTHROTTLE 1050 // for brushed ESCs like ladybird
```

The screenshot shows the Arduino IDE interface with the MultiWii - config.h sketch open. The code editor displays configuration defines for different vehicle types. The define `#define AIRPLANE` is highlighted in yellow, indicating it is selected or being edited. The status bar at the bottom right shows "Arduino/Genuino Uno on COM22", "2:45 PM", and the date "31/03/2021".

Output.cpp

Servo reversing is done thru here apart those on the flywii GUI

This also covers flaps and flapperons

Flying wing /Vtail modes are also set in this area

```
MultiWii - Output.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.h MultiWii.cpp Output.cpp Output.h Protocol.cpp Protocol.h RX.cpp RX.h Sensors.cpp sens
#ifndef AIRPLANE
#define AIRPLANE
// servo[7] is programmed with safty features to avoid motorstarts when ardu reset..
// All other servos go to center at reset.. Half throttle can be dangerous
// Only use servo[7] as motorcontrol if motor is used in the setup */
if (!f.ARMED) {
    servo[7] = MINCOMMAND; // Kill throttle when disarmed
} else {
    servo[7] = constrain(rcCommand[THROTTLE], conf.minthrottle, MAXTHROTTLE);
}
motor[0] = servo[7];

// Flapperon Controll TODO - optimilisation
int16_t flapperons[2]={0,0};
#if defined(FLAPPERONS) && defined(FLAPPERON_EP)
int8_t flapinv[2] = FLAPPERON_INVERT;
static int16_t F_Endpoint[2] = FLAPPERON_EP;
int16_t flap =MIDRC-constrain(rcData[FLAPPERONS],F_Endpoint[0],F_Endpoint[1]);
static int16_t slowFlaps= flap;
#if defined(FLAPSPEED)
    if (slowFlaps < flap ) {slowFlaps+=FLAPSPEED;}else if(slowFlaps > flap) {slowFlaps-=FLAPSPEED;}
#else
    slowFlaps = flap;
#endif
flap = MIDRC-(constrain(MIDRC-slowFlaps,F_Endpoint[0],F_Endpoint[1]));
for(i=0; i<2; i++) (flapperons[i] = flap * flapinv[i] );
#endif

// Traditional Flaps on SERVO3
```

Arduino/Genuino Uno on COM22
1241 2:48 PM 31/03/2021

Output.cpp

Servo reversing is done thru here apart those on the flywii GUI

This also covers flaps and flapperons

Flying wing /Vtail modes are also set in this area

```
MultiWii - Output.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 sens

int16_t lFlap = get_middle(2);
lFlap = constrain(lFlap, conf.servoConf[2].min, conf.servoConf[2].max);
lFlap = MIDRC - lFlap;
static int16_t slow_LFlaps= lFlap;
#ifndef FLAPSPEED
    if (slow_LFlaps < lFlap ){slow_LFlaps+=FLAPSPEED;} else if(slow_LFlaps > lFlap){slow_LFlaps-=FLAPSPEED;}
#else
    slow_LFlaps = lFlap;
#endif
servo[2] = ((int32_t)conf.servoConf[2].rate * slow_LFlaps)/100L;
servo[2] += MIDRC;
#endif

if(f.PASSTHRU_MODE){ // Direct passthru from RX
    servo[3] = rcCommand[ROLL] + flapperons[0]; // Wing 1
    servo[4] = rcCommand[ROLL] + flapperons[1]; // Wing 2
    servo[5] = rcCommand[YAW]; // Rudder
    servo[6] = rcCommand[PITCH]; // Elevator
} else{
    // Assisted modes (gyro only or gyro+acc according to AUX configuration in Gui
    servo[3] = axisPID[ROLL] + flapperons[0]; // Wing 1
    servo[4] = axisPID[ROLL] + flapperons[1]; // Wing 2
    servo[5] = axisPID[YAW]; // Rudder
    servo[6] = axisPID[PITCH]; // Elevator
}
for(i=3;i<7;i++){
    servo[i] = ((int32_t)conf.servoConf[i].rate * servo[i])/100L; // servo rates
    servo[i] += get_middle(i);
}

Wednesday, 31 March 2021
1295-1285
2:54 PM
31/03/2021
```

MOTOR MIN THROTTLE

CONFIG.H



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 Sens

```

//#define HELI_90_DEG

/****************************************** Motor minthrottle *****/
/* Set the minimum throttle command sent to the ESC (Electronic Speed Controller)
   This is the minimum value that allow motors to run at a idle speed */
#define MINTHROTTLE 1300 // for Turnigy Plush ESCs 10A
#define MINTHROTTLE 1120 // for Super Simple ESCs 10A
#define MINTHROTTLE 1064 // special ESC (simonk)
#define MINTHROTTLE 1050 // for brushed ESCs like ladybird
#define MINTHROTTLE 1150 // (*) (**) */

/****************************************** Motor maxthrottle *****/
/* this is the maximum value for the ESCs at full power, this value can be increased up to 2000 */
#define MAXTHROTTLE 1850

/****************************************** Mincommand *****/
/* this is the value for the ESCs when they are not armed
   in some cases, this value must be lowered down to 900 for some specific ESCs, otherwise they failed to initiate */
#define MINCOMMAND 1000

/****************************************** I2C speed for old WMP config (useless config for other sensors) *****/
#define I2C_SPEED 100000L //100kHz normal mode, this value must be used for a genuine WMP
//#define I2C_SPEED 400000L //400kHz fast mode, it works only with some WMP clones

/****************************************** Internal i2c Pullups *****/
/* enable internal I2C pull ups (in most cases it is better to use external pullups) */
#define INTERNAL_I2C_PULLUPS

/****************************************** constant loop time *****/
// - - - - - 2000

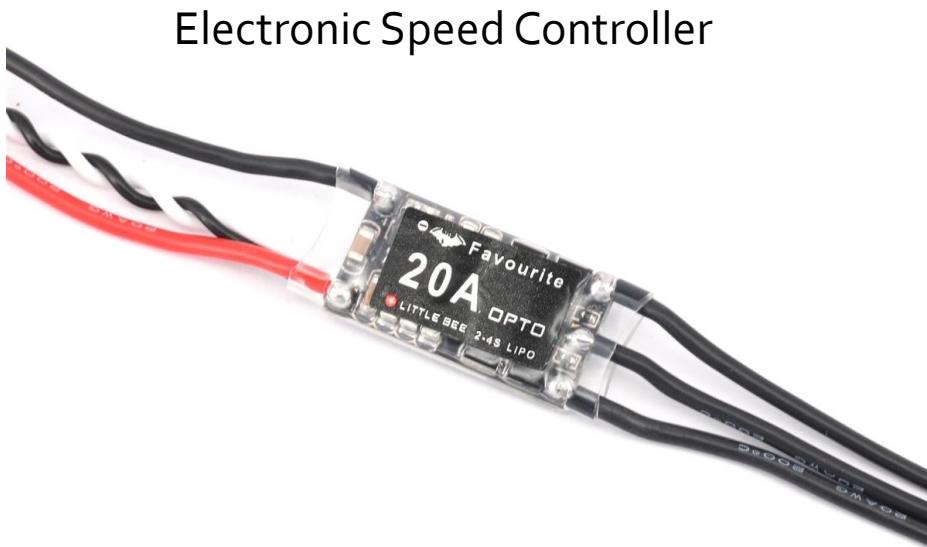
```

63 Arduino/Genuino Uno on COM41

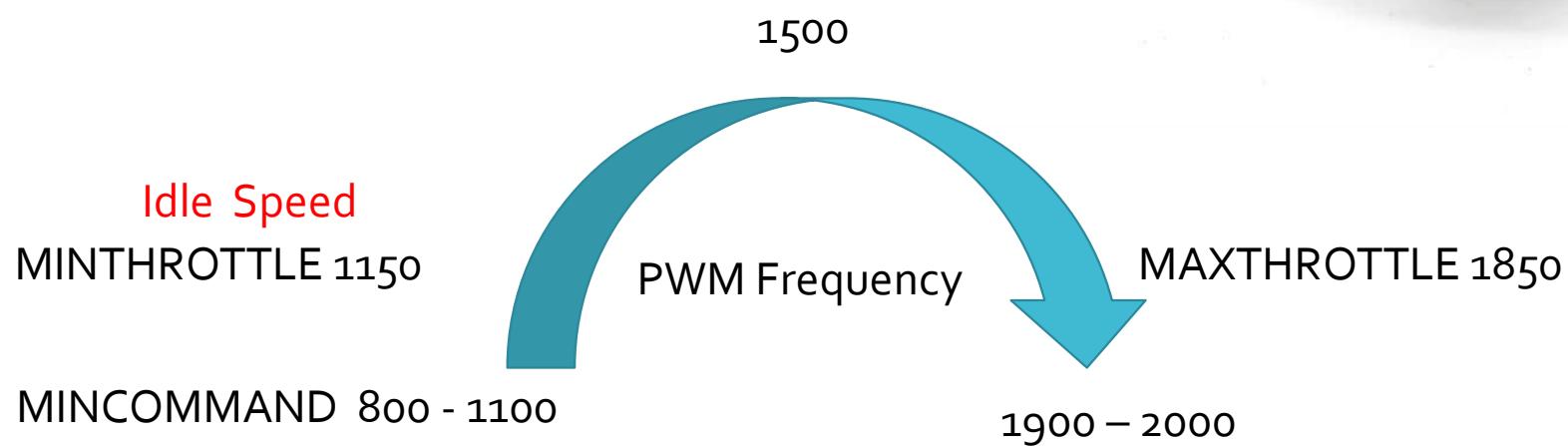
9:20 PM 13/02/2020

#define MAXTHROTTLE 2000
Because you want Full Power sometimes!

Motor Min Throttle



Brushless Motor



Output.cpp

Servo reversing is done thru here apart those on the flywii GUI

This also covers flaps and flapperons

Flying wing /Vtail modes are also set in this area

```
MultiWii - Output.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 sens
motor[0] = PIDMIX(+0,+1, +1); //REAR_R
motor[1] = PIDMIX(-1, -1, +0); //FRONT_R
motor[2] = PIDMIX(+0,+1, -1); //REAR_L
motor[3] = PIDMIX(+1, -1, -0); //FRONT_L
#elif defined( FLYING_WING )
***** FLYING WING *****
if (!f.ARMED) {
    servo[7] = MINCOMMAND; // Kill throttle when disarmed
} else {
    servo[7] = constrain(rcCommand[THROTTLE], conf.minthrottle, MAXTHROTTLE);
}
motor[0] = servo[7];
if (f.PASSTHRU_MODE) { // do not use sensors for correction, simple 2 channel mixing
    servo[3] = (SERVODIR(3,1) * rcCommand[PITCH]) + (SERVODIR(3,2) * rcCommand[ROLL]);
    servo[4] = (SERVODIR(4,1) * rcCommand[PITCH]) + (SERVODIR(4,2) * rcCommand[ROLL]);
} else {
    // use sensors to correct (gyro only or gyro+acc according to aux1/aux2 configuration
    servo[3] = (SERVODIR(3,1) * axisPID[PITCH]) + (SERVODIR(3,2) * axisPID[ROLL]);
    servo[4] = (SERVODIR(4,1) * axisPID[PITCH]) + (SERVODIR(4,2) * axisPID[ROLL]);
}
servo[3] += get_middle(3);
servo[4] += get_middle(4);
#elif defined( AIRPLANE )
***** AIRPLANE *****
// servo[7] is programmed with safty features to avoid motorstarts when ardu reset..
// All other servos go to center at reset.. Half throttle can be dangerous
// Only use servo[7] as motorcontrol if motor is used in the setup */
if (!f.ARMED) {
    servo[7] = MINCOMMAND; // Kill throttle when disarmed
} else {
```

1224 Arduino/Genuino Uno on COM22
ENG 2:50 PM 31/03/2021

Output.cpp

Flaps Settings

Conventional Flaps

Using servo[2] (A2 on ProMini)

```
//#define FLAPS // Enable Flaps .
```

Flaperons

```
//#define FLAPPERONS AUX4 //
```

Use Ailerons as Flaps .

```
#define FLAPPERON_EP { 1500, 1700 } //
```

Endpoints for flaps on a 2 way switch

```
#define FLAPPERON_INVERT { 1, -1 }
```

```
} // Change flap direction
```

Make flaps move slow Higher value is Higher Speed. Effects Both types of flaps.

```
//#define FLAPSPEED 3
```

Flaps (Flaperons)*

Select a channel to control your flaps with.

Endpoints is the preprogrammed positions for the flaperons .

Endpoints for flaps can also be programmed in the TX .

Then set FLAP_EP { 1020, 2000} in the code.

```
MultiWii - config.h | Arduino 1.8.2
File Edit Sketch Tools Help
Multivii 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 sens

/*
   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 Ailerons.
#define FLAPPERON_EP { 1500, 1700 } // Endpoints for flaps on a 2 way switch else set {1020,2000} and program in radio.
#define FLAPPERON_INVERT { -1, 1 } // Change direction on flaperons { Wing1, Wing2 }

#define FLAPS // Traditional Flaps on SERVO3.
#define FLAPSPEED 3 // Make flaps move slown Higher value is Higher Speed.

*****
   Common for Heli & Airplane
   /* Governor: attempts to maintain rpm through pitch and voltage changes
   * predictive approach: observe input signals and voltage and guess appropriate corrections.
   * (the throttle curve must leave room for the governor, so 0-50-75-80-80 is ok, 0-50-95-100-100 is _not_ ok.
   * Can be toggled via aux switch.
   */
#define GOVERNOR_P 7 // (*) proportional factor. Higher value -> higher throttle increase. Must be >=1; 0 = turn off
#define GOVERNOR_D 4 // (*) decay timing. Higher value -> takes longer to return throttle to normal. Must be >=1;

Done Saving

avrduude done. Thank you.

276 - 283
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM31
^ ENG 5:45 PM 31/03/2021
```

INERTIAL MEASURING UNIT MEASURING UNIT

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

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

Each sensor has a corresponding address registry set
by manufacturer

You can find it on sensors.ccp tab



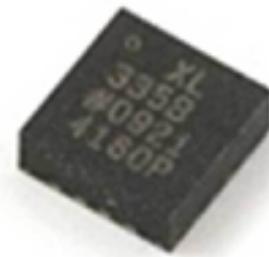
Magnetometer



Barometer



Accelerometer



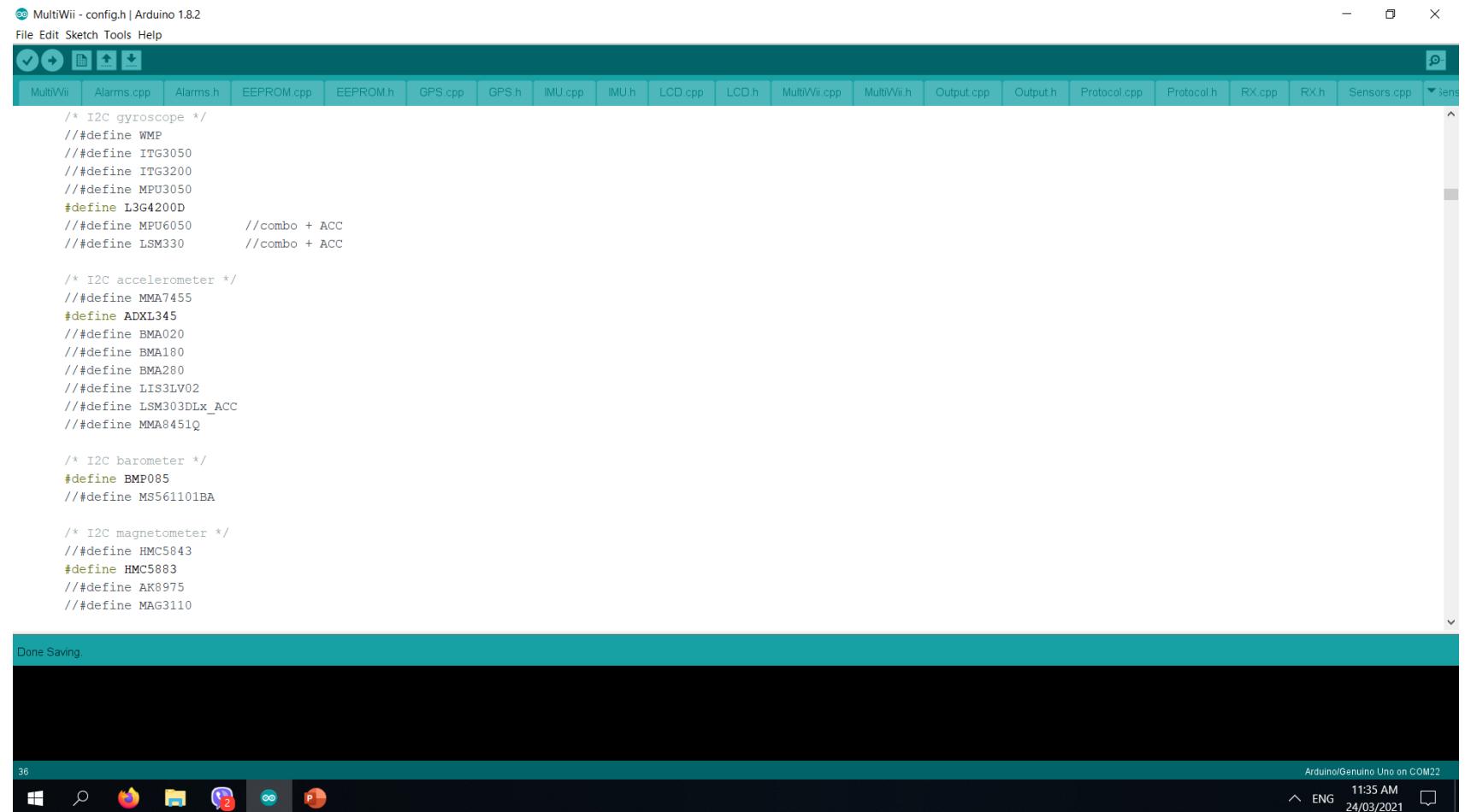
Gyroscope

**Sensors work best if mounted as close to CG
as possible.**

Config.h

Sensors

```
#define L3G4200D
#define ADXL345
#define BMP085
#define MMC5883
```



```
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 Sens
/*
 * I2C gyroscope */
#ifndef WMP
#define ITG3050
#define ITG3200
#define MPU3050
#define L3G4200D
#define MPU6050 //combo + ACC
#define LSM330 //combo + ACC

/*
 * I2C accelerometer */
#define MMA7455
#define ADXL345
#define BMA020
#define BMA180
#define BMA280
#define LIS3LV02
#define LSM303DLx_ACC
#define MMA8451Q

/*
 * I2C barometer */
#define BMP085
#define MS561101BA

/*
 * I2C magnetometer */
#define HMC5843
#define HMC5883
#define AK8975
#define MAG3110

Done Saving.
```

36

Arduino/Genuino Uno on COM22

ENG 11:35 AM 24/03/2021

CONFIG.H



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 ▾ Sens

```
***** independent sensors *****

/* leave it commented if you already checked a specific board above */

/* I2C gyroscope */
#ifndef WMP
#define ITG3050
#ifndef ITG3200
#define MPU3050
#define L3G4200D
#define MPU6050 //combo + ACC
#define LSM330 //combo + ACC

/* I2C accelerometer */
#define MMA7455
#define ADXL345
#define BMA020
#define BMA180
#define BMA280
#define LIS3LV02
#define LSM303DLX_ACC
#define MMA8451Q

/* I2C barometer */
#define BMP085
#define MS561101BA

/* I2C magnetometer */
#define HMC5843
#define HMC5883
//<--> ->-->
```

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

163 Arduino/Genuino Uno on COM41 9:28 PM 13/02/2020

SENSORS.CCP



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

9:37 PM ENG 13/02/2020

SENSORS.CCP



MultiWii - Sensors.cpp | Arduino 1.8.2

File Edit Sketch Tools Help

MultilWii 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]) ,
                    ((rawADC[5]<<8) | rawADC[4]) );
}
```

SENSORS.CCP

MultiWii - Sensors.cpp | Arduino 1.8.2

File Edit Sketch Tools Help

The image shows the Arduino IDE interface with the file "Sensors.cpp" open. The code is for a BOSCH BMP085 I2C Barometer sensor. It includes comments explaining the calibration process (steps 1-4) and defines a struct for sensor registers. The code uses union types to handle raw byte data and calculate compensated temperature and pressure. A note at the bottom indicates transforming big-endian bytes to little-endian.

```
// ****
// 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

Windows taskbar icons: File Explorer, Edge browser, Google Chrome, Paint 3D, Task View, Taskbar settings, Volume, Battery, Network, Language, Date/Time, Notifications.

SENSORS.CPP



MultiWii - Sensors.cpp | Arduino 1.8.2

File Edit Sketch Tools Help

Sensors.cpp

```
// I2C Compass MMC5883
// ****
// I2C adress: 0x30 (8bit) 0x00 (7bit)
// ****

#ifndef MMC5883_H
#define MMC5883_H

#define MAG_ADDRESS 0x30
#define MAG_DATA_REGISTER 0x00 //Read register address
//REG CONTROL
#define MMC5883MA_OUT 0x00
#define MMC5883MA_XOUT 0x00
#define MMC5883MA_XOUT_LOW 0x00
#define MMC5883MA_XOUT_HIGH 0x01
#define MMC5883MA_YOUT 0x02
#define MMC5883MA_YOUT_LOW 0x02
#define MMC5883MA_YOUT_HIGH 0x03
#define MMC5883MA_ZOUT 0x04
#define MMC5883MA_ZOUT_LOW 0x04
#define MMC5883MA_ZOUT_HIGH 0x05
#define MMC5883MA_TEMPERATURE 0x06
#define MMC5883MA_STATUS 0x07
#define MMC5883MA_INTERNAL_CONTROL_0 0x08
#define MMC5883MA_INTERNAL_CONTROL_1 0x09
#define MMC5883MA_INTERNAL_CONTROL_2 0x0A
#define MMC5883MA_X_THRESHOLD 0x0B
#define MMC5883MA_Y_THRESHOLD 0x0C
#define MMC5883MA_Z_THRESHOLD 0x0D

#endif
```

1264 - 1265

Arduino/Genuino Uno on COM22

3:22 PM 31/03/2021

SENSOR ORIENTATION

CONFIG.H



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 sens

```
//#define HMC5843
#define HMC5883
//#define AK8975
//#define MAG3110

/* Sonar */ // for visualization purpose currently - no control code behind
#ifndef SRF02 // use the Devantech SRF i2c sensors
#define SRF08
#define SRF10
#define SRF23

/* ADC accelerometer */ // for 5DOF from sparkfun, uses analog PIN A1/A2/A3
#ifndef ADCACC

/* enforce your individual sensor orientation - even overrides board specific defaults */
#define FORCE_ACC_ORIENTATION(X, Y, Z) {imu.accADC[ROLL] = X; imu.accADC[PITCH] = Y; imu.accADC[YAW] = Z;}
#define FORCE_GYRO_ORIENTATION(X, Y, Z) {imu.gyroADC[ROLL] = -Y; imu.gyroADC[PITCH] = X; imu.gyroADC[YAW] = -Z;}
#define FORCE_MAG_ORIENTATION(X, Y, Z) {imu.magADC[ROLL] = -X; imu.magADC[PITCH] = -Y; imu.magADC[YAW] = Z;}

/* Board orientation shift */
/* If you have frame designed only for + mode and you cannot rotate FC physically for flying in X mode (or vice versa)
 * you can use one of these options for virtual sensors rotation by 45 degrees, then set type of multicopter according to flight mode.
 * Check motors order and directions of motors rotation for matching with new front point! Uncomment only one option!
#ifndef SENSORS_TILT_45DEG_RIGHT      // rotate the FRONT 45 degrees clockwise
#define SENSORS_TILT_45DEG_LEFT       // rotate the FRONT 45 degrees counterclockwise

*****
```

203 - 206

Arduino/Ger 2 new notifications

9:33 PM 13/02/2020

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

MIXING

OUTPUT.CPP



SynerduinoWingPlane3-GY801 - Output.cpp | Arduino 1.8.2

File Edit Sketch Tools Help

SynerduinoWingPlane3-GY801 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 R...

```
motor[1] = PIDMIX(-1, -1, +0); //FRONT_L
motor[2] = PIDMIX(+0,+1, -1); //REAR_L
motor[3] = PIDMIX(+1, -1, -0); //FRONT_L
#endif defined( FLYING_WING )
/********************************************* FLYING WING *****/
if (!f.ARMED) {
    servo[7] = MINCOMMAND; // Kill throttle when disarmed
} else {
    servo[7] = constrain(rcCommand[THROTTLE], conf.minthrottle, MAXTHROTTLE);
}
motor[0] = servo[7];
if (f.PASSTHRU_MODE) { // do not use sensors for correction, simple 2 channel mixing
    servo[3] = (SERVODIR(3,-1) * rcCommand[PITCH]) + (SERVODIR(3,2) * rcCommand[ROLL]);
    servo[4] = (SERVODIR(4,1) * rcCommand[PITCH]) + (SERVODIR(4,2) * rcCommand[ROLL]);
} else { // use sensors to correct (gyro only or gyro+acc according to aux1/aux2 configuration
    /*With Rudder Mix*/
    servo[3] = (SERVODIR(3,-1) * axisPID[PITCH]) + (SERVODIR(3,2) * axisPID[ROLL])+(SERVODIR(3,2) * axisPID[YAW]);
    servo[4] = (SERVODIR(4,1) * axisPID[PITCH]) + (SERVODIR(4,2) * axisPID[ROLL])+(SERVODIR(4,2) * axisPID[YAW]);
    /*No Rudder Mix*/
    servo[3] = (SERVODIR(3,-1) * axisPID[PITCH]) + (SERVODIR(3,2) * axisPID[ROLL]);
    servo[4] = (SERVODIR(4,1) * axisPID[PITCH]) + (SERVODIR(4,2) * axisPID[ROLL]);
}
servo[3] += get_middle(3);
servo[4] += get_middle(4);
#endif defined( AIRPLANE )
/********************************************* AIRPLANE *****/
// servo[7] is programmed with safty features to avoid motorstarts when ardu reset..
// All other servos go to center at reset.. Half throttle can be dangerous
// Only use servo[7] as motorcontrol if motor is used in the setup */
if (!f.ARMED) {
    servo[7] = MINCOMMAND; // Kill throttle when disarmed
} else {

```

Done Saving.

1238 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM30 Links ENG 9:14 AM 29/04/2022

MIXING

OUTPUT.CPP



Airplane and Flying Wings section

```
motor[0] = servo[7];
if (f.PASSTHRU_MODE) {      // do not use sensors for correction, simple 2 channel mixing
    servo[3] = (SERVODIR(3,-1) * rcCommand[PITCH]) + (SERVODIR(3,2) * rcCommand[ROLL]);
    servo[4] = (SERVODIR(4,1) * rcCommand[PITCH]) + (SERVODIR(4,2) * rcCommand[ROLL]);
} else {                     // use sensors to correct (gyro only or gyrotacc according to aux1/aux2 configuration
                           /*With Rudder Mix*/
    servo[3] = (SERVODIR(3,-1) * axisPID[PITCH]) + (SERVODIR(3,2) * axisPID[ROLL])+(SERVODIR(3,2) * axisPID[YAW]);
    servo[4] = (SERVODIR(4,1) * axisPID[PITCH]) + (SERVODIR(4,2) * axisPID[ROLL])+(SERVODIR(4,2) * axisPID[YAW]);
                           /*No Rudder Mix*/
    servo[3] = (SERVODIR(3,-1) * axisPID[PITCH]) + (SERVODIR(3,2) * axisPID[ROLL]);
    servo[4] = (SERVODIR(4,1) * axisPID[PITCH]) + (SERVODIR(4,2) * axisPID[ROLL]);
```

Cases than none true for flying wings , or special base airplane that certain servo Mixing &
 reversing is required due to the nature of Elevon, Ruddervator or Rudderon configuration
(-1, 0, 1, 2) digits are use to determine Normal / Inactive / Reverse command
The GUI may have limitation to certain mixing for this it has to be config here.

PASSTHRU_MODE = this command when use Direct control from the RX without augmentation from the Flight control board

SERIAL COM

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 FLUSH 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

CONFIG.H



```
MultiWii - config.h | Arduino 1.8.2
File Edit Sketch Tools Help

MultiVii 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 Sensors.h

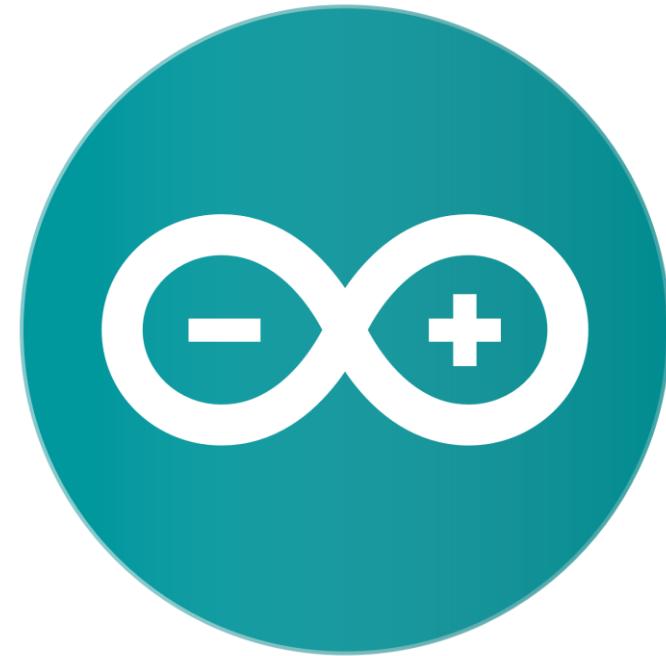
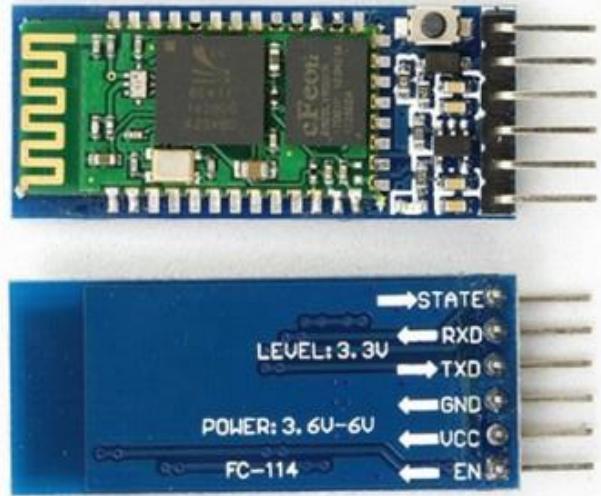
/*
*****
***** SECTION 5 - ALTERNATE SETUP *****
*****
****

***** Serial com speed *****
/* This is the speed of the serial interfaces */
#define SERIAL0_COM_SPEED 38400
#define SERIAL0_COM_SPEED 57600
#define SERIAL0_COM_SPEED 115200
#define SERIAL1_COM_SPEED 115200
#define SERIAL2_COM_SPEED 115200
#define SERIAL3_COM_SPEED 115200

/* when there is an error on I2C bus, we neutralize the values during a short time. expressed in microseconds
it is relevant only for a conf with at least a WMP */
#define NEUTRALIZE_DELAY 100000

***** Gyro filters *****
***** Lowpass filter for some gyros *****
/* ITG3200 & ITG3205 Low pass filter setting. In case you cannot eliminate all vibrations to the Gyro, you can try
to decrease the LPF frequency, only one step per try. As soon as twitching gone, stick with that setting.
It will not help on feedback wobbles, so change only when copter is randomly twitching and all dampening and
balancing options ran out. Uncomment only one option!
IMPORTANT! Change low pass filter setting changes PID behaviour, so retune your PID's after changing LPF.

498
Arduino/Genuino Uno on COM4
9:47 PM 13/02/2020
```



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=Change name

AT+PSWD=1234 (Version 2)

AT+PSWD="1234" (Version 3)

AT+UART=115200,1,0



HC-05 (Recommended)

MultiWii | Arduino 1.8.5

File Edit Sketch Tools Help

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 Firmware Updater

ArduBlock

Board: "Arduino/Genuino Uno"

Port

Get Board Info

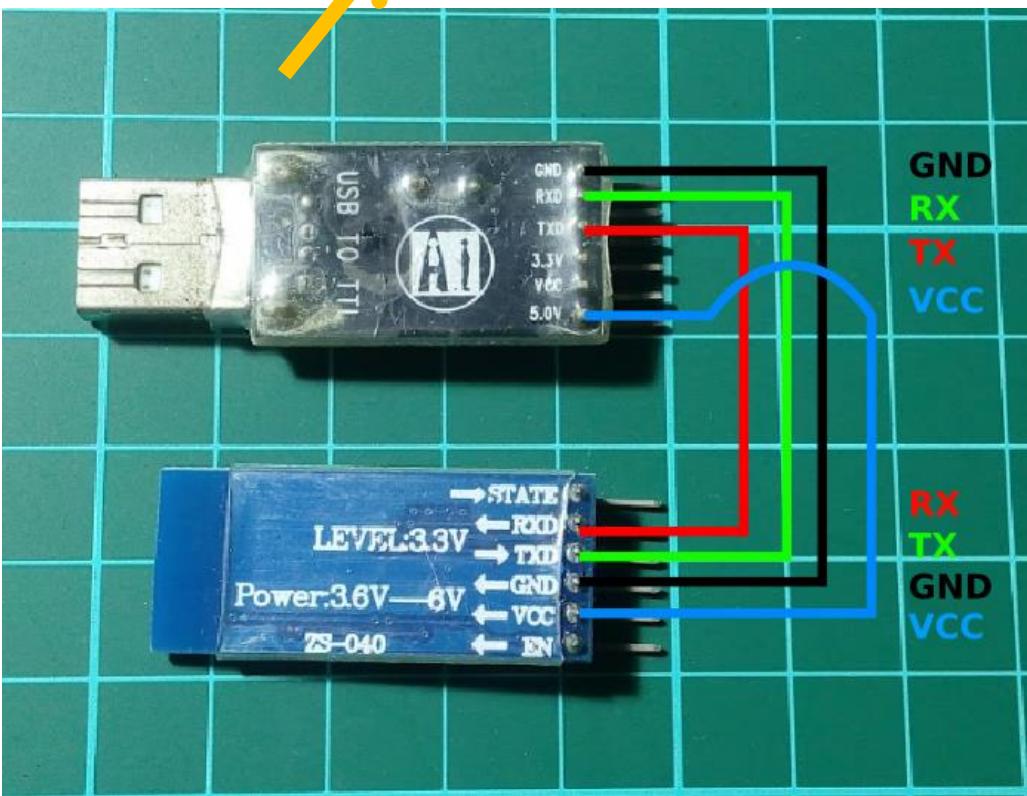
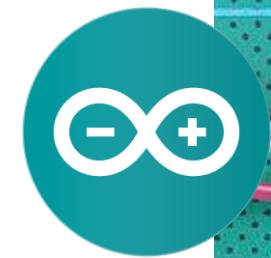
Programmer: "AVRISP mkII"

Burn Bootloader

```
/*
 * Welcome
 *
 * If you see this message, it means you are using the Arduino IDE. That is ok.
 * To get started with your copter, you must switch to the tab named 'config.h'.
 * Maybe there is a configuration file for your board or sensors and optionally various features.
 * For more information, please visit the MultiWii wiki at
 * /wiki/index.php?title=Main_Page
 *
 * Have fun!
 */

```





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)
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) (Possibly 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.

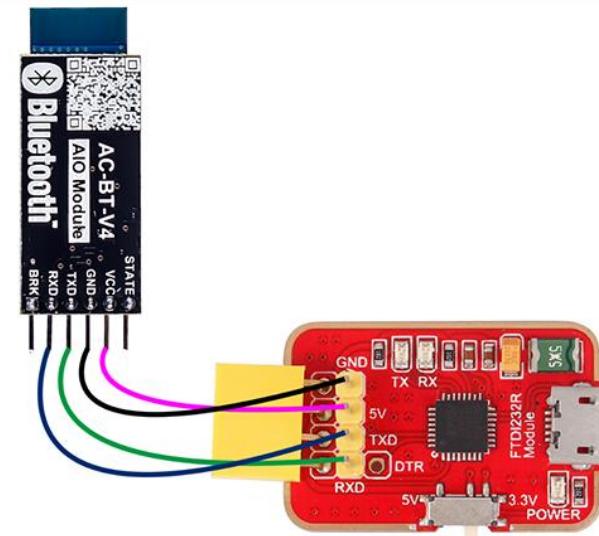
Example

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+CON22C22FF22DA2** to BLE A Send **AT+CON11C11FF11DA1** 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+NAME ArduinoDrone

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

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

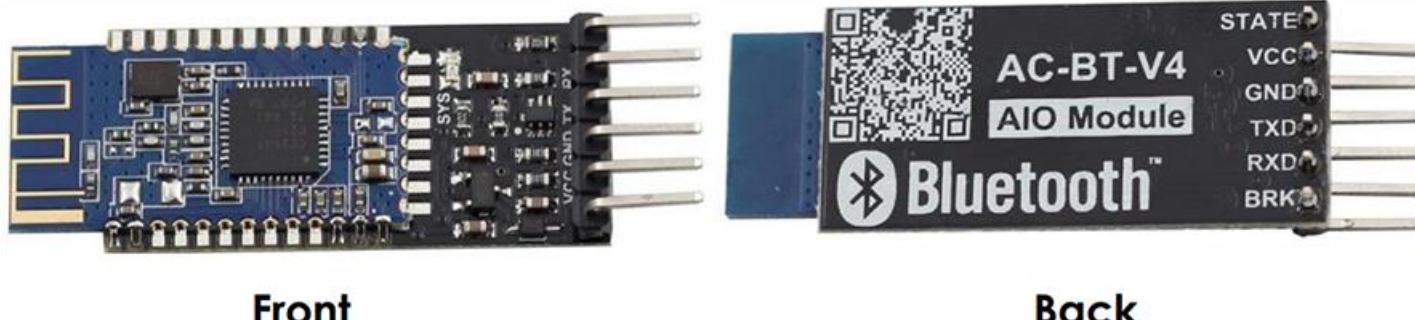
AT+PASS₁₂₃₄₅₆ Set password to 123456

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

AT+ROLE

0 = Slave or Peripheral

1 = Master or Central.



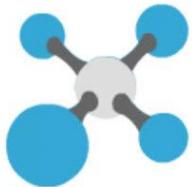
Front

Back

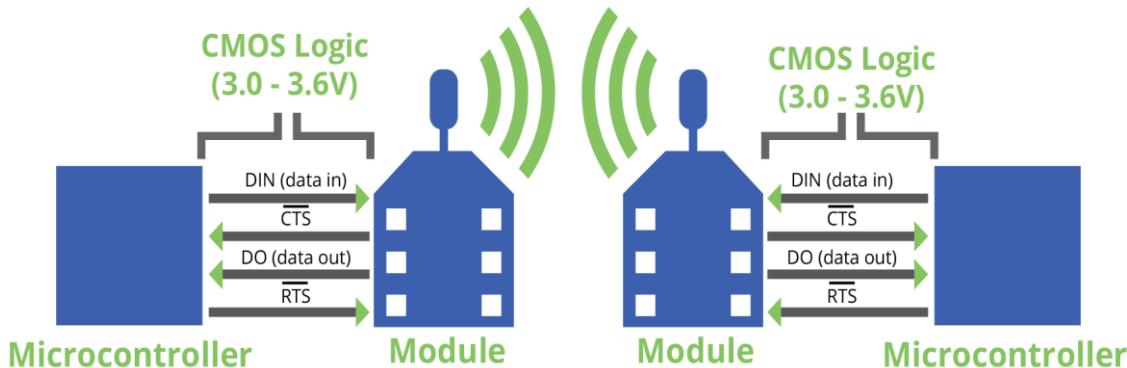
Note : there are several clones of this type in the market that can be very difficult to setup



XBEE RADIO



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



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

Update firmware

Update the radio module firmware

Configure the firmware that will be flashed to the radio module.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XB24-B	ZigBee End Device Digital IO	22A7 (Newest)
XB24-SE	ZigBee End Device PH	22A0
XB24-ZB	ZigBee Router API	228C
	ZigBee Router AT	2270
	ZigBee Router AT (WALL RT)	2264
	ZigBee Router Sensor	2242
	ZigBee Router/End Device Analog IO	2241

Force the module to maintain its current configuration.

Select current

[View Release Notes](#)

[Update](#) [Cancel](#)

Update firmware

Update the radio module firmware

Configure the firmware that will be flashed to the radio module.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XB24-B	End Device - LTH	20A7 (Newest)
XB24-SE	ZigBee Coordinator API	20A0
XB24-ZB	ZigBee Coordinator AT	208C
	ZigBee End Device API	2070
	ZigBee End Device AT	2064
	ZigBee End Device Analog IO	2041
	ZigBee End Device Digital IO	2021

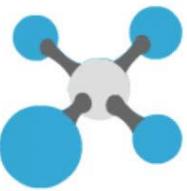
Force the module to maintain its current configuration.

Select current

[View Release Notes](#)

[Update](#) [Cancel](#)

GROUND STATION



XCTU

Radio Modules

Name: ZigBee Router AT
Function: ZigBee Router AT
Port: COM35 - 38400/8/N/1/N - AT
MAC: 0013A20040811A91

Radio Configuration [- 0013A20040811A91]

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
NW Network Watchdog Timeout: 0 x 1 minute
JV Channel Verification: Disabled [0]
JN Join Notification: Disabled [0]
OP Operating PAN ID: 1234
OI Operating 16-bit PAN ID: AD9F
CH Operating Channel: 14
NC Number of Remaining Children: C

Addressing

Change addressing settings

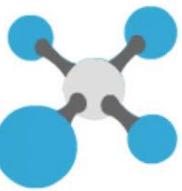
SH Serial Number High: 13A200
SL Serial Number Low: 40811A91
MY 16-bit Network Address: 7FA4
DH Destination Address High: 13A200
DL Destination Address Low: 40811A7F
NI Node Identifier:
NH Maximum Hops: 1E
BH Broadcast Radius: 0
AR Many-to-One Route Broadcast Time: FF x 10 sec
DD Device Type Identifier: 30000
NT Node Discovery Backoff: 3C x 100 ms
ND Node Discovery Outage:

Checking for Radio Firmw... updates: (87%)

2:30 PM 07/04/2020



GROUND STATION



XCTU

Radio Modules

Name: ZigBee Router AT
Function: ZigBee Router AT
Port: COM35 - 38400/8/N/1/N - AT
MAC: 0013A20040811A91

Radio Configuration [- 0013A20040811A91]

POWER ALIVE

Parameter

Security

Change security parameters

- EE Encryption Enable: Disabled [0]
- EO Encryption Options: Bitfield [0]
- KY Encryption Key: []

Serial Interfacing

Change modem interfacing options

- BD Baud Rate: 38400 [5]
- NB Parity: No Parity [0]
- SB Stop Bits: One stop bit [0]
- RO Packetization Timeout: 3 x character times
- D7 DIO7 Configuration: CTS flow control [1]
- D6 DIO6 Configuration: Disable [0]

AT Command Options

Change AT command mode behavior

- CT AT Command Mode Timeout: 64 x 100ms
- GT Guard Times: 3E8 x 1ms
- CC Command Sequence Character: 2B Recommended: 0x20-0x7F (ASCII)

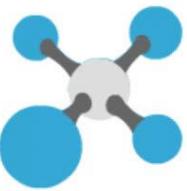
Sleep Modes

Configure low power options to support end device children

- SM Sleep Mode: No Sleep (Router) [0]
- SN Number of Cyclic Sleep Periods: 1
- SO Sleep Options: 0
- SP Cyclic Sleep Period: 20 x 10 ms
- ST Time before Sleep: 1388 x 1 ms

Image of the XB24-Z7UIT module:

AIRCRAFT



XCTU

Radio Modules

Name: ZigBee Coordinator AT
Function: ZigBee Coordinator AT
Port: COM36 - 38400/8/N/1/N - AT
MAC: 0013A20040811A7F

Radio Configuration [- 0013A20040811A7F]

Networking

Change networking settings

① ID PAN ID	1234			
① SC Scan Channels	FFFF	Bitfield		
① SD Scan Duration	3	exponent		
① ZS ZigBee Stack Profile	0			
① NJ Node Join Time	FF	x 1 sec		
① OP Operating PAN ID	1234			
① OI Operating 16-bit PAN ID	AD9F			
① CH Operating Channel	14			
① NC Number of Remaining Children	A			

Addressing

Change addressing settings

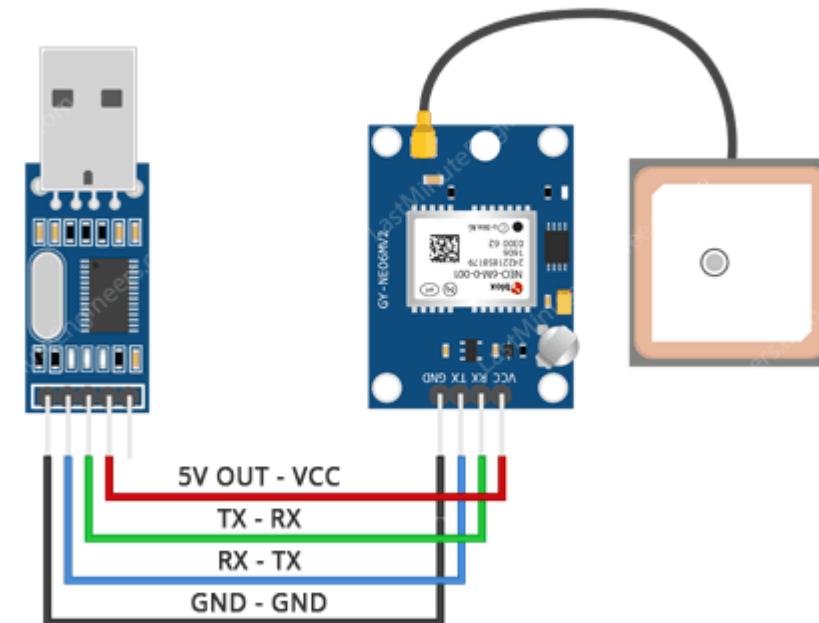
① SH Serial Number High	13A200		
① SL Serial Number Low	40811A7F		
① MY 16-bit Network Address	0		
① DH Destination Address High	13A200		
① DL Destination Address Low	40811A91		
① NI Node Identifier			
① NH Maximum Hops	1E		
① BH Broadcast Radius	0		
① AR Many-to-One Route Broadcast Time	FF x 10 sec		
① DD Device Type Identifier	30000		
① NT Node Discovery Backoff	3C x 100 ms		
① NO Node Discovery Options	0		
① NP Maximum Number of Transmission Bytes	54		



GPS



GPS CONFIGURING



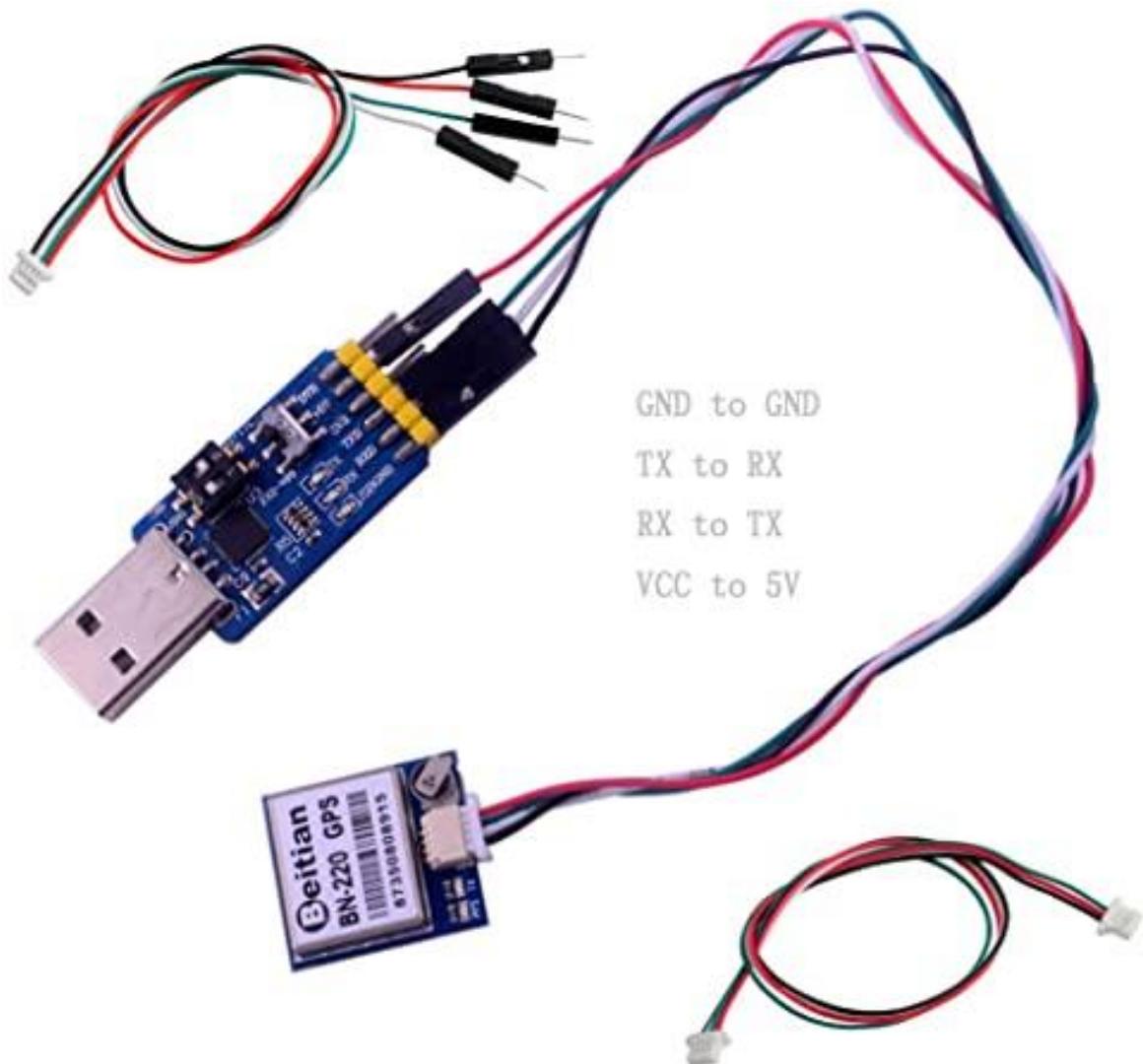
U BLOX NEO 6

PLUG IN TO SERIAL TX 2 RX 2

USB TTL TO PROGRAM THE GPS

THIS GOES SAME ON THE DRONE SHIELD

GPS CONFIGURING



BEITIAN UBLOX

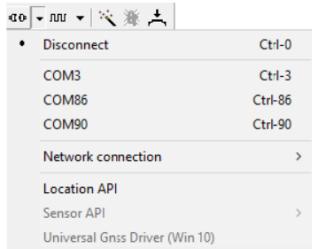
PLUG IN TO SERIAL TX 2 RX 2

USB TTL TO PROGRAM THE GPS

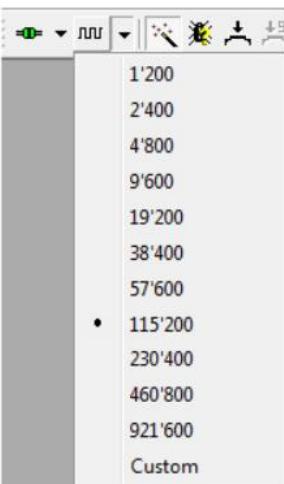
THIS GOES SAME ON THE DRONE SHIELD



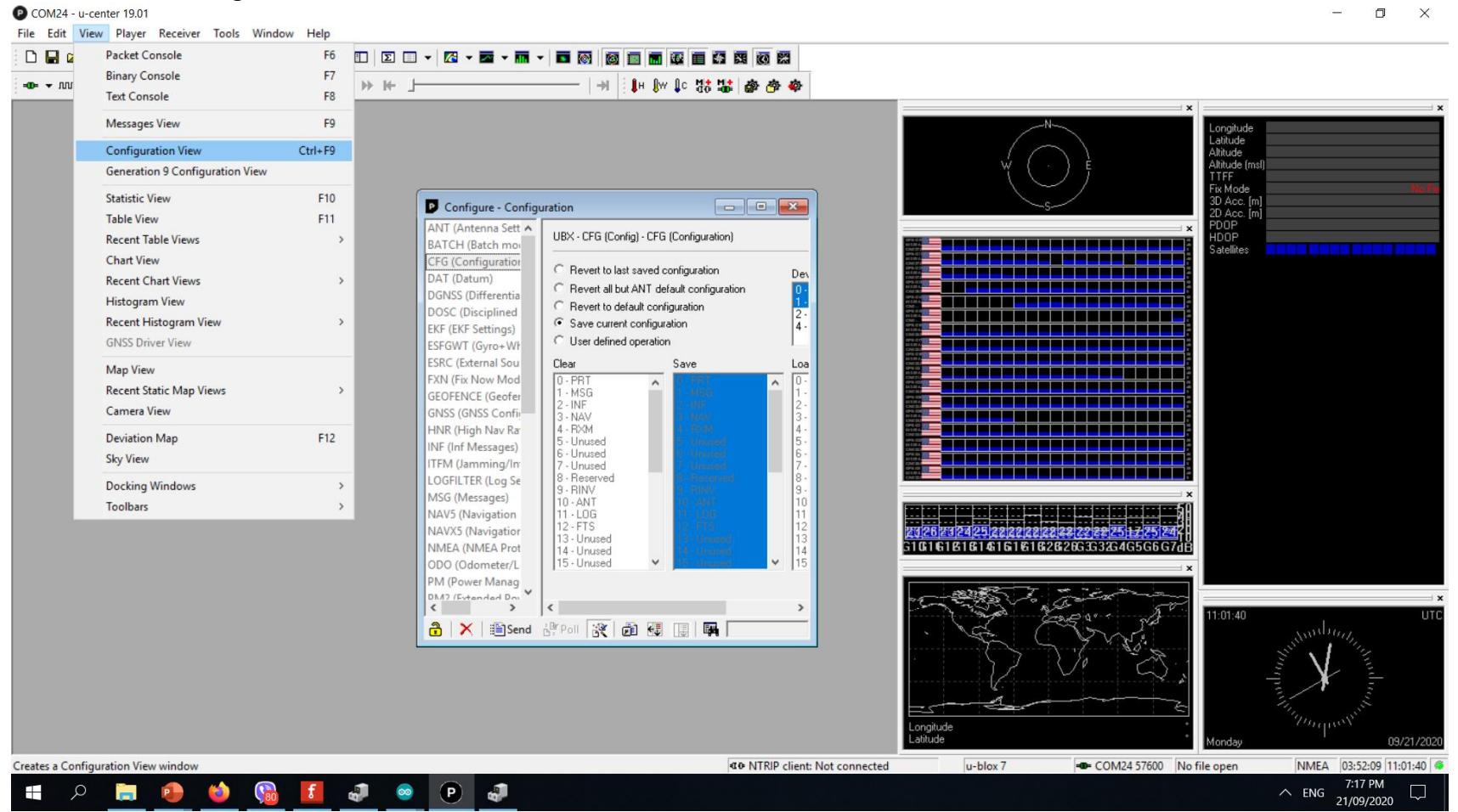
Connect to the device. Select the COM port your GPS and USBTTL is connected to



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



Go to View> Configuration View

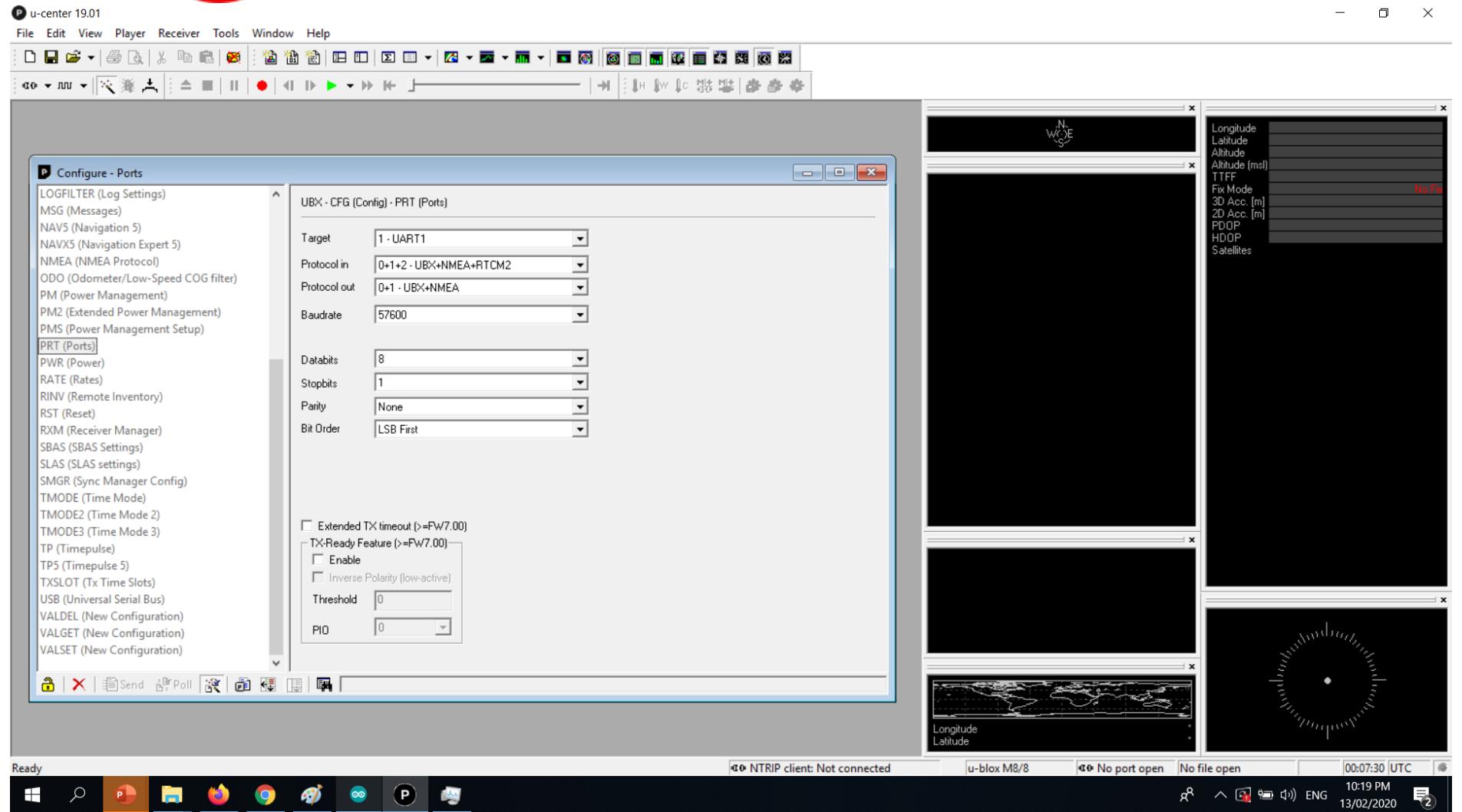


U CENTER



ublox

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

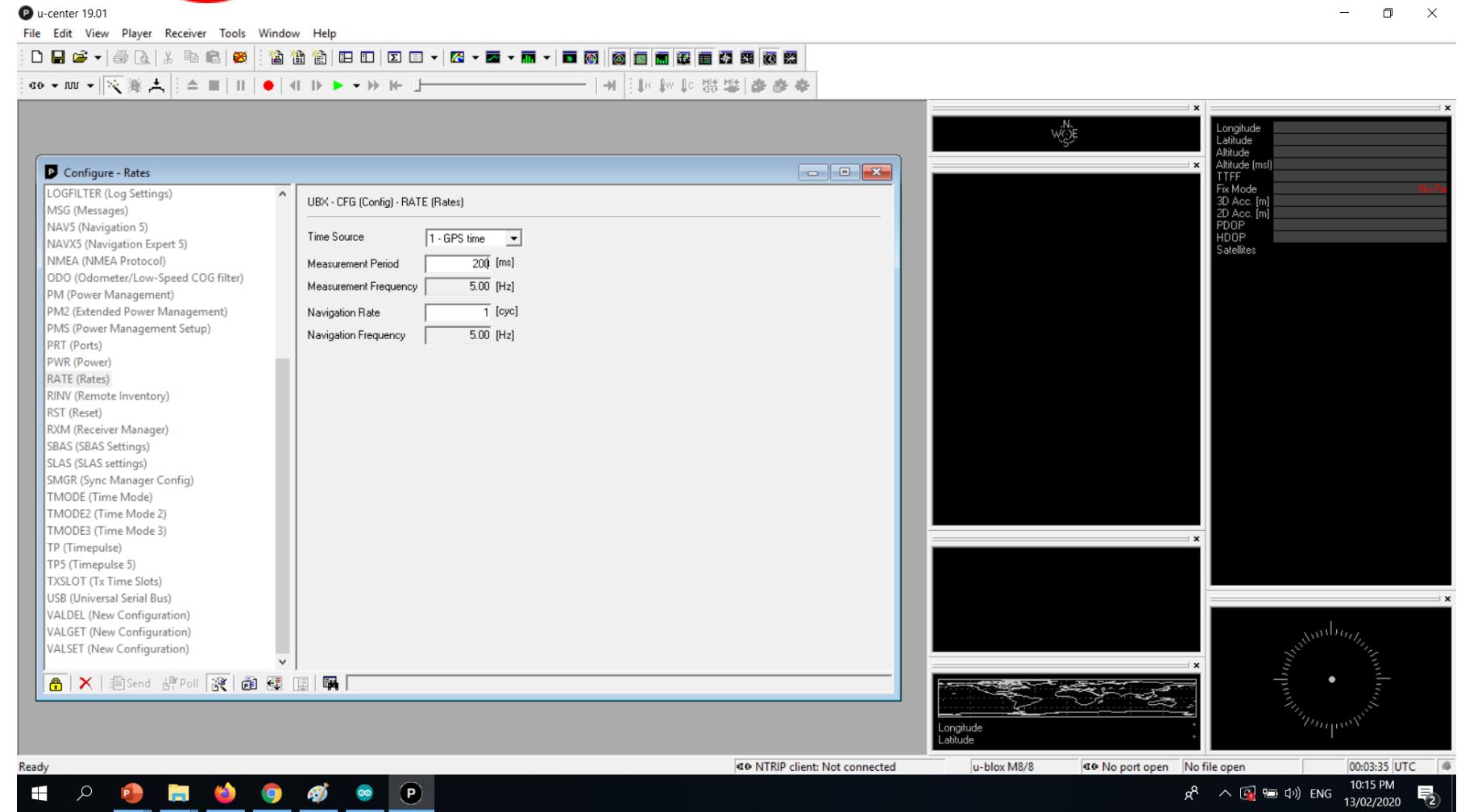




Rate zooms
Frequency 5hz

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

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



GPS

CONFIG.H



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 Sensors.h

```
/*
 * introduce a deadband around the stick center
 * Must be greater than zero, comment if you dont want a deadband on roll, pitch and yaw */
#define DEADBAND 6

/*
 * ENable this for using GPS simulator (NMEA only) */
#define GPS_SIMULATOR

/*
 * GPS using a SERIAL port
 * if enabled, define here the Arduino Serial port number and the UART speed
 * note: only the RX PIN is used in case of NMEA mode, the GPS is not configured by multiwii
 * in NMEA mode the GPS must be configured to output GGA and RMC NMEA sentences (which is generally the default conf for most GPS devices)
 * at least 5Hz update rate. uncomment the first line to select the GPS serial port of the arduino */

#define GPS_SERIAL 2      // should be 2 for flyduino v2. It's the serial port number on arduino MEGA
                        // must be 0 for PRO_MINI (ex GPS_PRO_MINI)
                        // note: Now a GPS can share MSP on the same port. The only constrain is to not use it simultaneously, and use the same port speed.

// avoid using 115200 baud because with 16MHz arduino the 115200 baudrate have more than 2% speed error (57600 have 0.8% error)
#define GPS_BAUD 38400 // ublox 8 new standard
#define GPS_BAUD 9600
#define GPS_BAUD 57600 // GPS_BAUD will override SERIALx_COM_SPEED for the selected port (my ublox 6)
#define GPS_BAUD 115200

Done Saving.
```

676 Arduino/Genuino Uno on COM41 4:05 PM 20/02/2020

Set GPS BAUD to 57600 as configured to the GPS module

GPS

CONFIG.H



SynerduinoKwad3 - config.h | Arduino 1.8.5

File Edit Sketch Tools Help

SynerduinoKwad3 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 Sens.cpp

```
// avoid using 115200 baud because with 16MHz arduino the 115200 baudrate have more than 2% speed error (57600 have 0.8% error)
//#define GPS_BAUD 9600 // GPS default
//#define GPS_BAUD 38400
#define GPS_BAUD 57600 // GPS_BAUD will override SERIALX_COM_SPEED for the selected port
//#define GPS_BAUD 115200

/* GPS protocol
NMEA - Standard NMEA protocol GGA, GSA and RMC sentences are needed
UBLOX - U-Blox binary protocol, use the ublox config file (u-blox-config.ublox.txt) from the source tree
MTK_BINARY16 and MTK_BINARY19 - MTK3329 chipset based GPS with DIYDrones binary firmware (v1.6 or v1.9)
With UBLOX and MTK Binary you don't have to use GPS FILTERING in multiwii code !!! */

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

#ifndef NMEA //for Ublox NMEA GPS - NMEA Protocol
#define UBLOX //for Beitian GPS - UBX Protocol
#define MTK_BINARY16
#define MTK_BINARY19
#define INIT_MTK_GPS // initialize MTK GPS for using selected speed, 5Hz update rate and GGA & RMC sentence or binary settings

/* I2C GPS device made with an independant arduino + GPS device
including some navigation functions
contribution from EOSBandi http://code.google.com/p/i2c-gps-nav/
You have to use at least I2CGpsNav code r33 */
/* all functionnalities allowed by SERIAL_GPS are now available for I2C_GPS: all relevant navigation computations are gathered in the main FC */

#define I2C_GPS
```

694 - 699

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM25

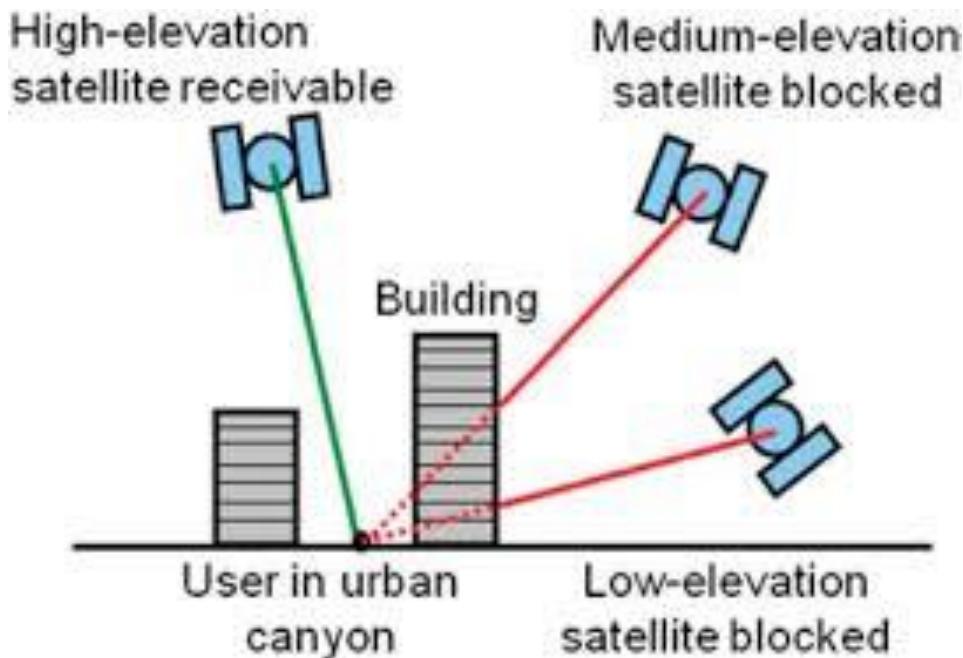
^ ENG 2:59 PM 30/09/2021

Depending on the GPS Version

Older formats uses
NMEA protocol

Newer GPS uses
UBLOX UBX Protocol

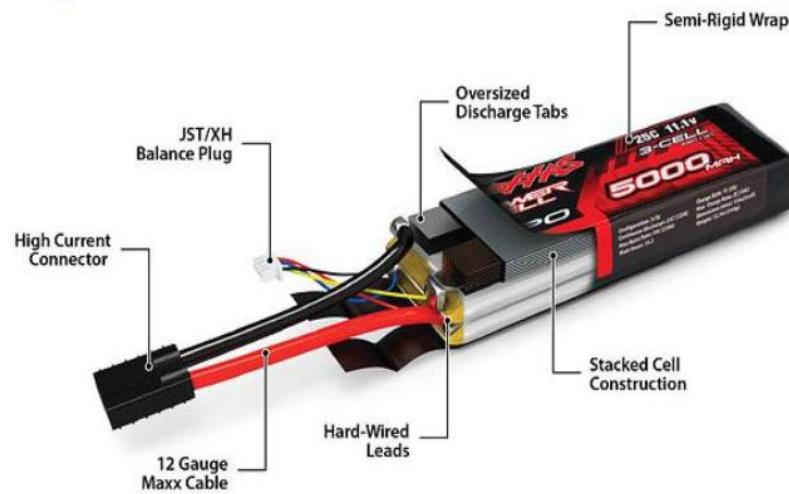
GPS



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 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**

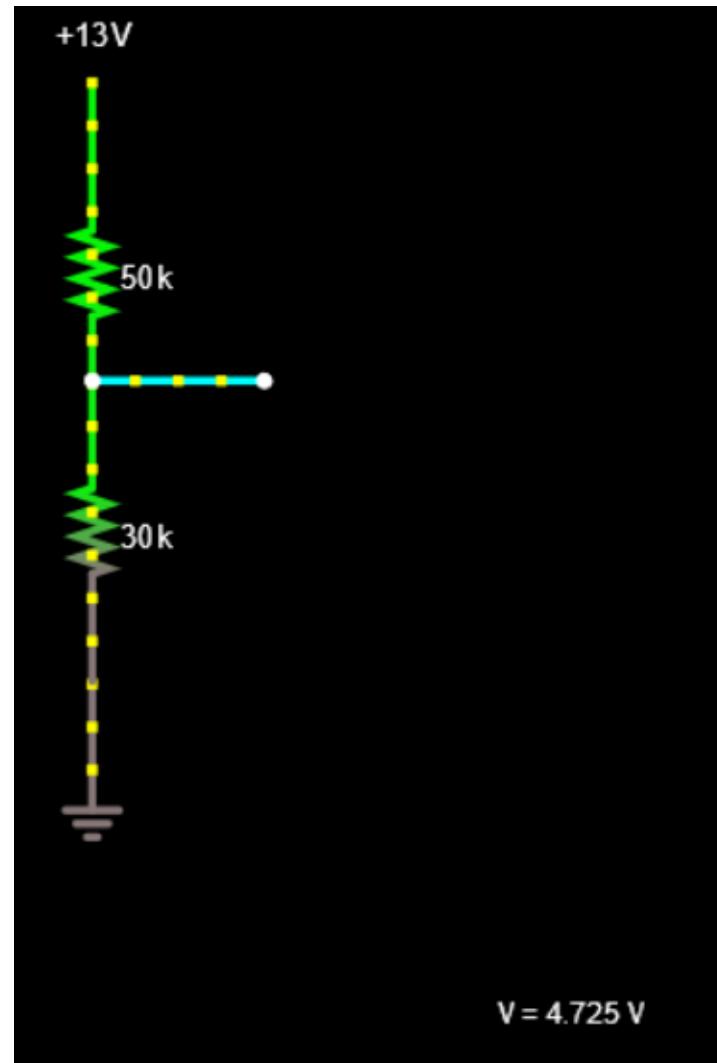
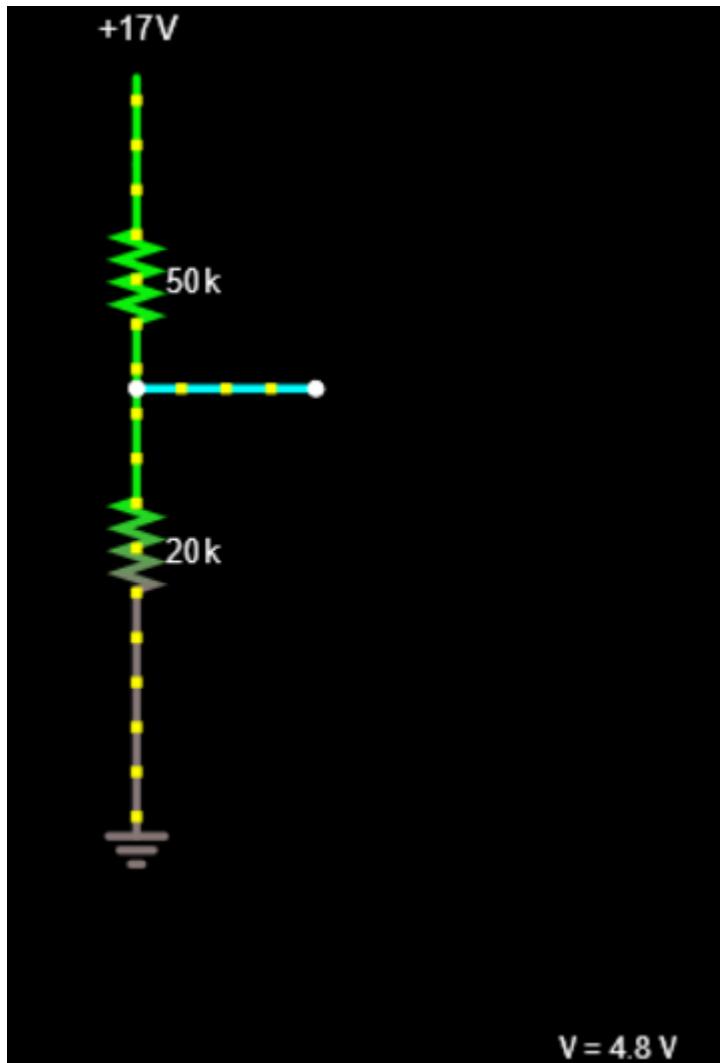


ITS RECOMMEND TO USE THE VOLTAGE ALARM TO MONITOR THE BATTERY VOLTAGE WHILE IN USE

BATTERY

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 Ao ANALOG
PIN OF THE ARDUINO TO READ
THE BATTERY VOLTAGE

SWITCH THE VBAT JUMPER
ACCORDINGLY TO THE BATTER
CELLS YOUR USING

3S OR 4S

BATTERY MONITORING

CONFIG.H



MultiWii - config.h | Arduino 1.8.2

File Edit Sketch Tools Help

Multivii 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 ▾ Sens

```
/*
***** battery_voltage_monitoring *****
*****
/* for V BAT monitoring
   after the resistor divisor we should get [0V;5V]->[0;1023] on analog V_BATPIN
   with R1=33k and R2=51k
   vbat = [0;1023]*16/VBATSCALE
   must be associated with #define BUZZER ! */
#ifndef VBAT           // uncomment this line to activate the vbat code
#define VBATSCALE      131 // (*) (**) change this value if readed Battery voltage is different than real voltage
#define VBATNOMINAL    126 // 12,6V full battery nominal voltage - only used for lcd.telemetry
#define VBATLEVEL_WARN1 107 // (*) (**) 10,7V
#define VBATLEVEL_WARN2 99 // (*) (**) 9,9V
#define VBATLEVEL_CRIT 93 // (*) (**) 9,3V - critical condition: if vbat ever goes below this value, permanent alarm is triggered
#define NO_VBAT         16 // Avoid beeping without any battery
#define VBAT_OFFSET     18 // offset in 0.1Volts, gets added to voltage value - useful for zener diodes

/* for V BAT monitoring of individual cells
 * enable both VBAT and VBAT_CELLS
 */
#ifndef VBAT_CELLS
#define VBAT_CELLS_NUM 0 // set this to the number of cells you monitor via analog pins
#define VBAT_CELLS_PINS {A0, A1, A2, A3, A4, A5 } // set this to the sequence of analog pins
#define VBAT_CELLS_OFFSETS {0, 50, 83, 121, 149, 177 } // in 0.1 volts, gets added to voltage value - useful for zener diodes
#define VBAT_CELLS_DIVS { 75, 122, 98, 18, 30, 37 } // divisor for proportional part according to resistors - larger value here gives smaller voltage

***** powermeter (battery capacity monitoring) *****

```

892

Arduino/Genuino Uno on COM4!

10:24 PM 13/02/2020

The bottom of the screen shows a Windows taskbar with several icons: Start button, File Explorer, Microsoft Edge, Google Chrome, Paint 3D, Mail, and a red notification badge icon.

PIN ASSIGNMENTS

CONFIG.H



MultiWii - config.h | Arduino 1.8.2

File Edit Sketch Tools Help

Buzzer Pin Promicro version related

```
/*
 * this moves the Buzzer pin from TXO to D8 for use with ppm sum or spectrum sat. RX (not needed if A32U4ALLPINS is active)
 //define D8BUZZER

 *****
 * Inverted status LED for Promicro ver 10
 //define PROMICRO10

 *****
 * override default pin assignments
 *****
 * only enable any of this if you must change the default pin assignment, e.g. your board does not have a specific pin
 * you may need to change PINx and PORTx plus #shift according to the desired pin!
 #define OVERRIDE_V_BATPIN      A0 // instead of A3    // Analog PIN 3

 ##define OVERRIDE_PSENSORPIN      A1 // instead of A2    // Analog PIN 2

 ##define OVERRIDE_LEDPIN_PINMODE      pinMode (A1, OUTPUT); // use A1 instead of d13
 ##define OVERRIDE_LEDPIN_TOGGLE      PINC |= 1<<1; // PINB |= 1<<5;      //switch LEDPIN state (digital PIN 13)
 ##define OVERRIDE_LEDPIN_OFF      PORTC &= ~(1<<1); // PORTB &= ~(1<<5);
 ##define OVERRIDE_LEDPIN_ON      PORTC |= 1<<1;      // was PORTB |= (1<<5);

 ##define OVERRIDE_BUZZERPIN_PINMODE      pinMode (A2, OUTPUT); // use A2 instead of d8
 ##define OVERRIDE_BUZZERPIN_ON      PORTC |= 1<<2 //PORTB |= 1;
 ##define OVERRIDE_BUZZERPIN_OFF      PORTC &= ~(1<<2); //PORTB &= ~1;

 *****
 Done Saving.
```

479 Arduino/Genuino Uno on COM41

9:30 AM 21/02/2020



CAMERA GIMBAL

CONFIG.H

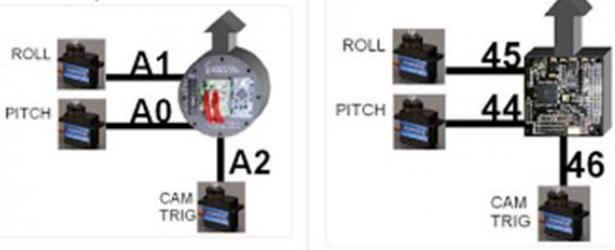


Photo for illustration purpose only.

Pure stabilized gimbal system

For this setup, you need a GYRO + ACC setup (not possible with a gyro only setup).
A RC system is optional for this setup.

for arduino mega:



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 send

/*
//#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}
#define SERVO_MAX {2000, 2000, 2000, 2000, 2000, 2000, 2000}
#define SERVO_MID {1500, 1500, 1500, 1500, 1500, 1500, 1500} // (*)
#define FORCE_SERVO_RATES {30,30,100,100,100,100,100} // 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.

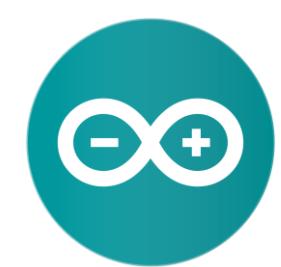
#define FLAPPERONS AUX4 // Mix Flaps with Ailerons.
#define FLAPPERON_EP { 1500, 1700 } // Endpoints for flaps on a 2 way switch else set {1020,2000} and program in radio.
#define FLAPPERON_INVERT { -1, 1 } // Change direction on flaperons { Wing1, Wing2 }

#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

8 ENG 6:16 PM
11/08/2020



CONFIG.H

Arm/DisArm

Option for combination stick command to start and stop the drone

Note: Combination stick only works on some vehicles configuration . Others uses Aux Arm switch

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

```
/* NEW: not used anymore for servo coptertypes <== NEEDS FIXING - MOVE TO WIKI */
#define YAW_DIRECTION 1
//#define YAW_DIRECTION -1 // if you want to reverse the yaw correction direction

#define ONLYARMWHENFLAT //prevent the copter from arming when the copter is tilted

/********************* ARM/DISARM *****************/
/* optionally disable stick combinations to arm/disarm the motors.
 * In most cases one of the two options to arm/disarm via TX stick is sufficient */
#define ALLOW_ARM_DISARM_VIA_TX_YAW
//#define ALLOW_ARM_DISARM_VIA_TX_ROLL

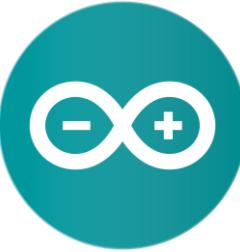
/********************* SERVOS *****************/
/* info on which servos connect where and how to setup can be found here
 * http://www.multiwii.com/wiki/index.php?title=Config.h#Servos\_configuration
 */

/* Do not move servos if copter is unarmed
 * It is a quick hack to overcome feedback tail wiggles when copter has a flexible
 * landing gear
 */
#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_MTD {1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500} // (*)
```

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

CONFIG.H



SynerduinoKwad - config.h | Arduino 1.8.2

File Edit Sketch Tools Help

A Value on 200 will give a very distinct transfer */

```
//#define ACROTRAINER_MODE 200 // http://www.multiwii.com/forum/viewtopic.php?f=16&t=1944#p17437

***** Failsafe settings *****
/* Failsafe check pulses on four main control channels CH1-CH4. If the pulse is missing or bellow 985us (on any of these four channels)
the failsafe procedure is initiated. After FAILSAFE_DELAY time from failsafe detection, the level mode is on (if ACC is available),
PITCH, ROLL and YAW is centered and THROTTLE is set to FAILSAFE_THROTTLE value. You must set this value to descending about 1m/s or so
for best results. This value is depended from your configuration, AUW and some other params. Next, after FAILSAFE_OFF_DELAY the copter is disarmed,
and motors is stopped. If RC pulse coming back before reached FAILSAFE_OFF_DELAY time, after the small guard time the RC control is returned to normal. */
#define FAILSAFE          // uncomment to activate the failsafe function
#define FAILSAFE_DELAY    10      // Guard time for failsafe activation after signal lost. 1 step = 0.1sec - 1sec in example
#define FAILSAFE_OFF_DELAY 200     // Time for Landing before motors stop in 0.1sec. 1 step = 0.1sec - 20sec in example
#define FAILSAFE_THROTTLE  (MINTHROTTLE + 200) // (*) Throttle level used for landing - may be relative to MINTHROTTLE - as in this case

#define FAILSAFE_DETECT_THRESHOLD 985

***** DFRobot LED RING *****
/* I2C DFRobot LED RING communication */
//#define LED_RING

***** LED FLASHER *****
#define LED_FLASHER
#ifndef LED_FLASHER_DDR
#define LED_FLASHER_PORT PORTB
#define LED_FLASHER_BIT PORTB4
#define LED_FLASHER_INVERT
#endif
#define LED_FLASHER_SEQUENCE 0b00000000 // leds OFF

609
```

Arduino/Genuine Uno on COM8

9:54 AM 22/07/2021 2

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

Flight Deck Mission Flight Tuning FC Config RC Control Settings Sensor Graph VideoCapture GUI Settings CLI

5 Hz Refresh Rate Pause Calibrate ACC Calibrate Mag Bind Spektrum Cycle Tim

FLYWIIGUI GROUND STATION

<http://synerflight.com/flywiogui/>

The screenshot displays the FlyWiigui Ground Station software interface. At the top, there are tabs for Flight Deck, Mission, Flight Tuning, FC Config, RC Control Settings, Sensor Graph, VideoCapture, GUI Settings, and CLI. Below the tabs are buttons for Refresh Rate, Pause, Calibrate ACC, Calibrate Mag, Bind Spektrum, and Cycle Tim. On the left, there's a large black overlay with the text "FLYWIIGUI GROUND STATION" and a URL. The main interface includes a compass rose, a battery level indicator (0.0V), and a bar chart for Active Sensors (ACC, GPS, BARO, HDG). A central panel shows a quadcopter icon with four 1500 values (FRONT, LEFT, RIGHT, REAR) and a "QuadCopter +" label. At the bottom, there's a "Telemetry link info" section with RSSI and Noise fields, and a "PC Error count" section with Battery Voltage and Power Sum indicators.

Packet's sent: 0
Packet's received: 0
Packet CRC error: 0

Telemetry link info

RSSI: no data
Noise: no data

PC Error count
Battery Voltage:
Power Sum:



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

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)

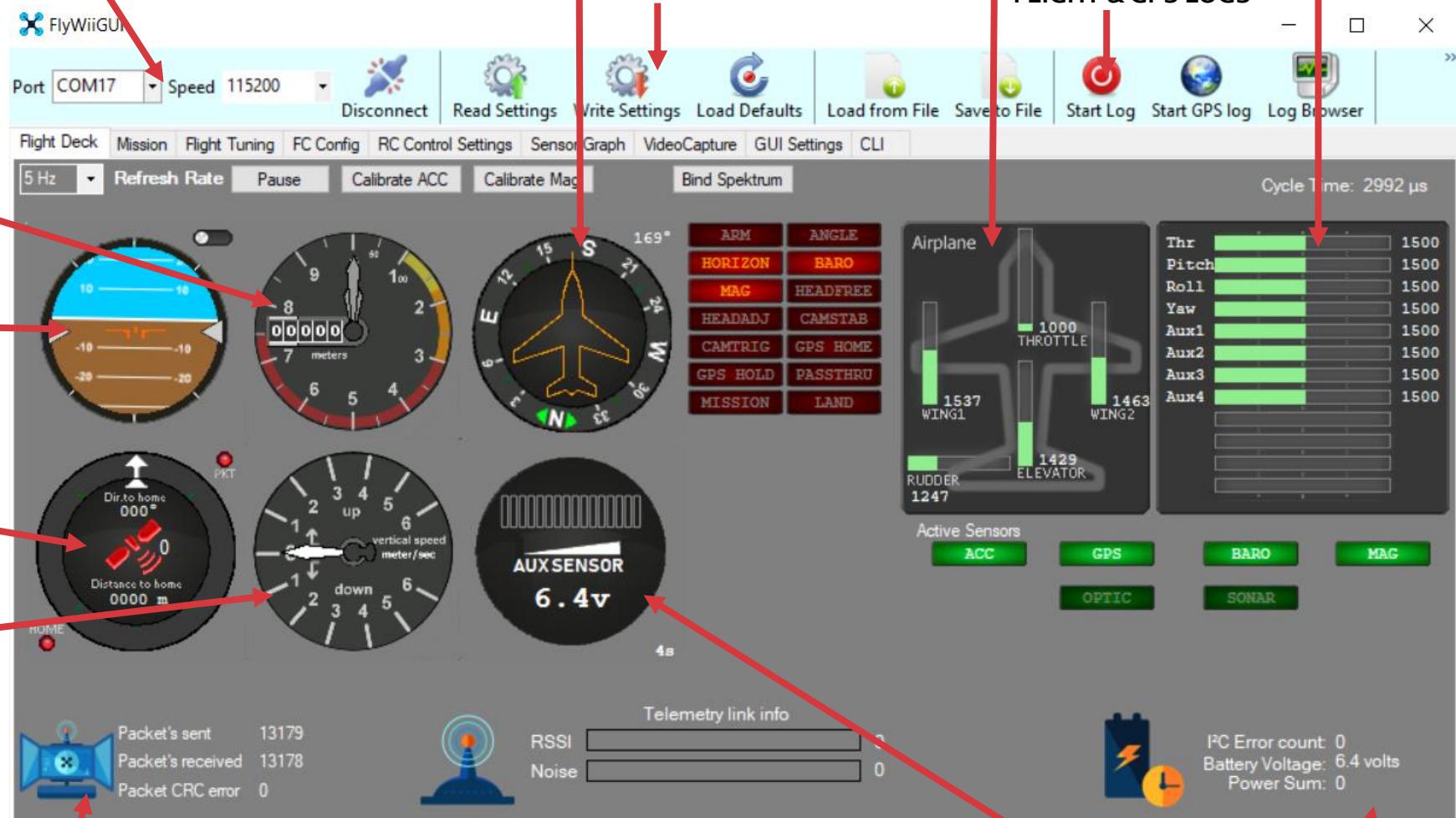
COMPASS (MAG AND GYRO)

PWM OUTPUT INDICATOR

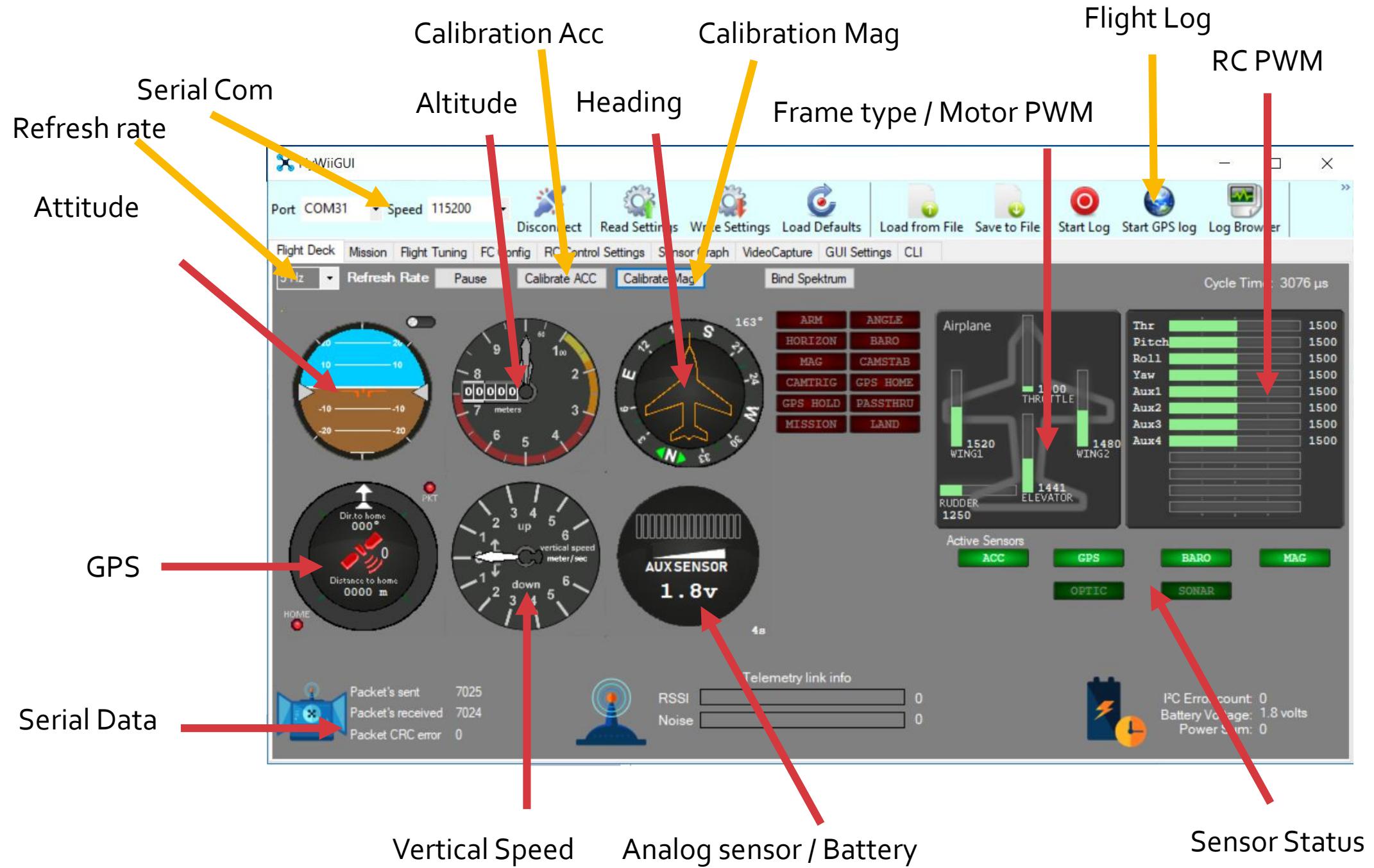
PWM INPUT INDICATOR

SAVE CONFIG

FLIGHT & GPS LOGS

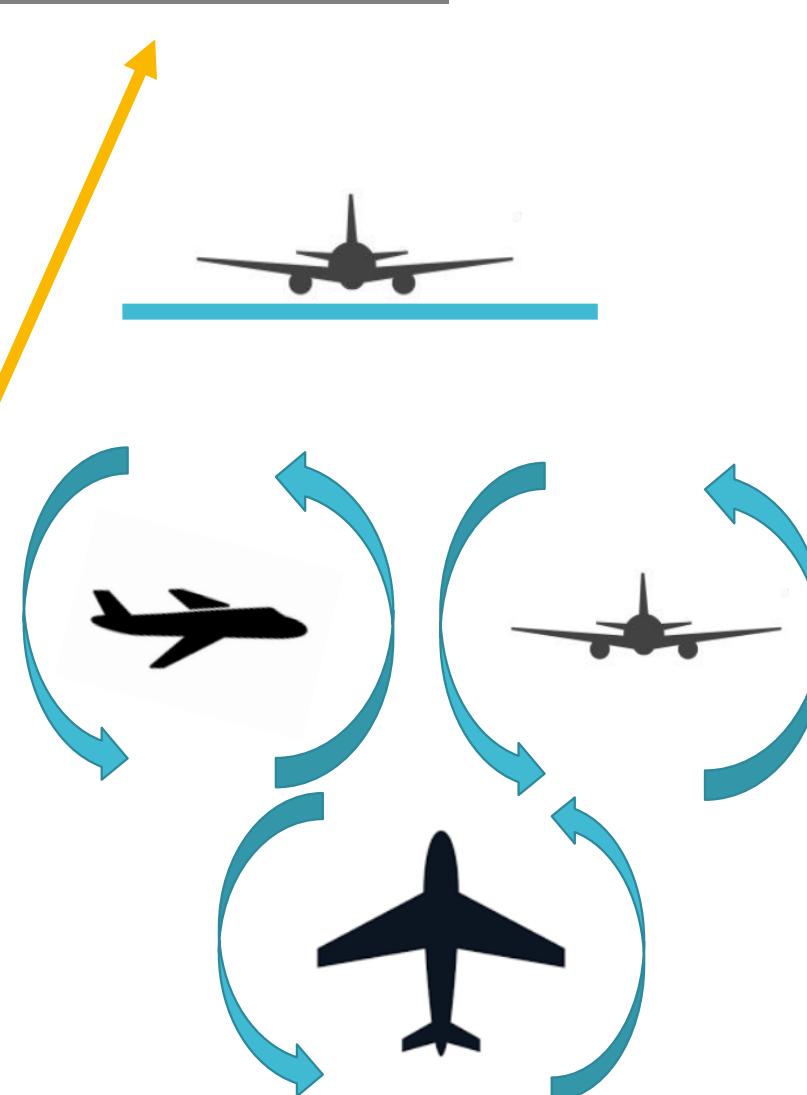


POWER STATUS / AUX SENSOR
ALSO KNOWN AS FUEL GAUGE
(VBAT)





- Refresh Rate . Telemetry update speed
- Acc Calibration . Set the drone down on a level surface . Away from any metal objects for 10 secs.
- Mag Calibration . rotate the aircraft 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



FC Config

Gyro or Acc assisted Mode.

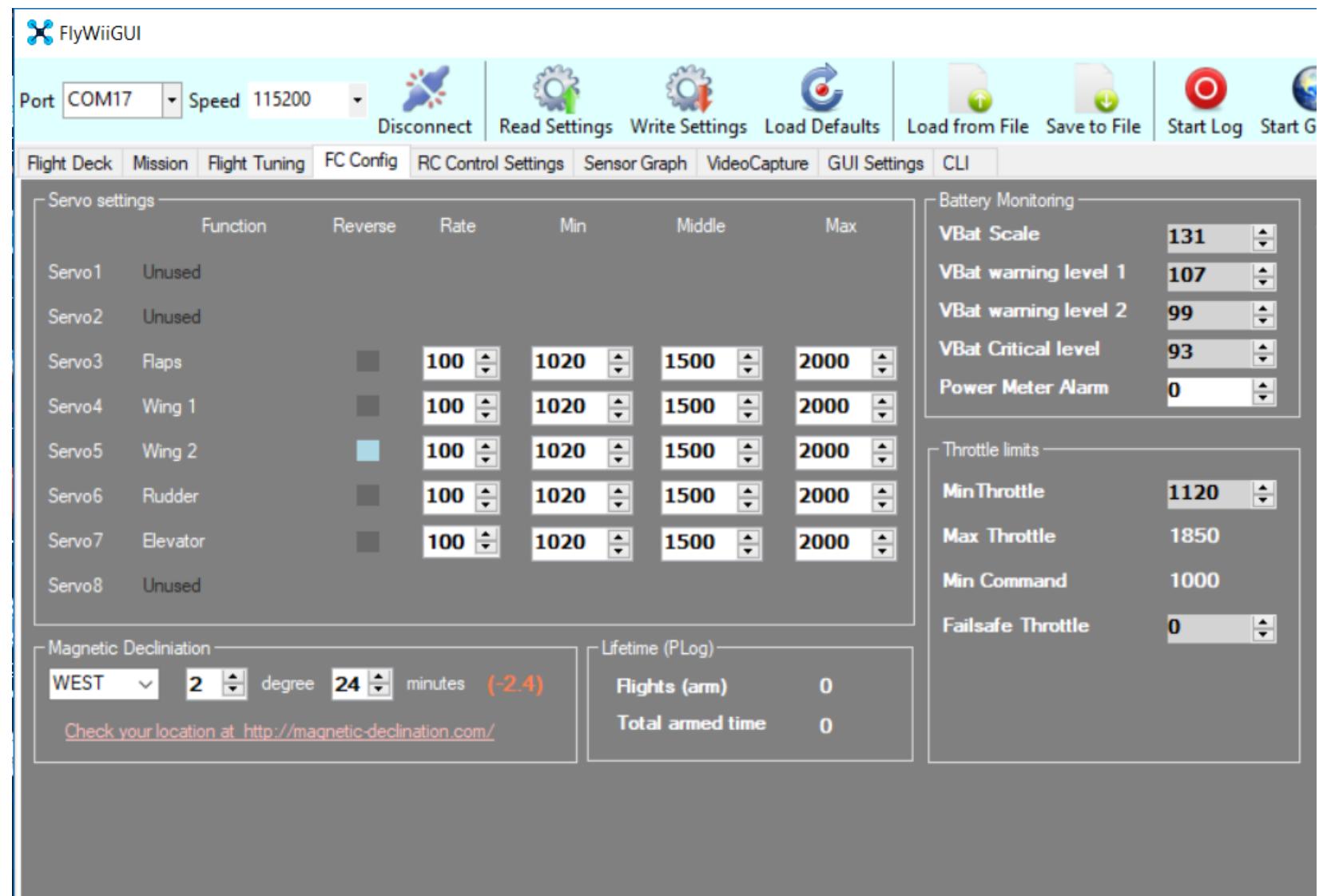
Check if Gyro move servos in right directions.

Lift a wingtip and Aileron goes up.

Lift the tail and Elevator goes up.

Rudder moves in same direction as the tail.

Use the Servo Tab in Gui to Change Servo directions.



Servos Settings

Gyro or Acc assisted Mode.

Check if Gyro move servos in right directions.

Lift a Right wingtip and Right Aileron goes up.

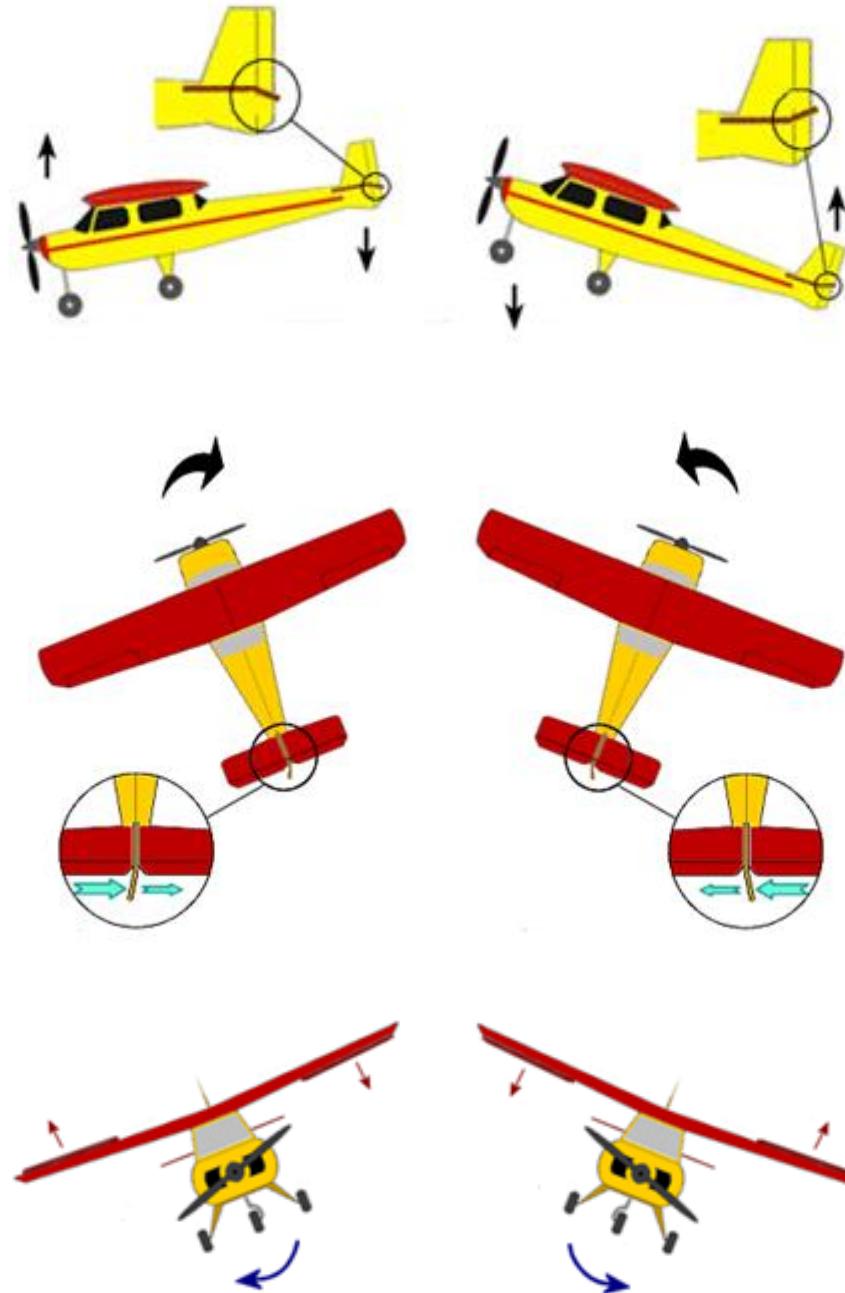
Lift the tail and Elevator goes up.

Rudder moves in same direction as the tail.

Use the Servo Tab in Gui to Change Servo directions.

The Aircraft should give the correct proportion for control attitude correction

Control surfaces should oppose the disturbance applied to the aircraft





FLYWII GUI

Battery Monitoring

VBat Scale	120
VBat warning level 1	110
VBat warning level 2	110
VBat Critical level	109
Power Meter Alarm	0

VBat 15.4 volts

Battery Cell Count

4s	▼
1s	
2s	
3s	
4s	
5s	
6s	
7s	
8s	
9s	
10s	

(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 MONITORING

ARM – Arms the main motor of the aircraft

Setup in PassThru Mode.

Set all TX trims to zero and check the value when the sticks centers.

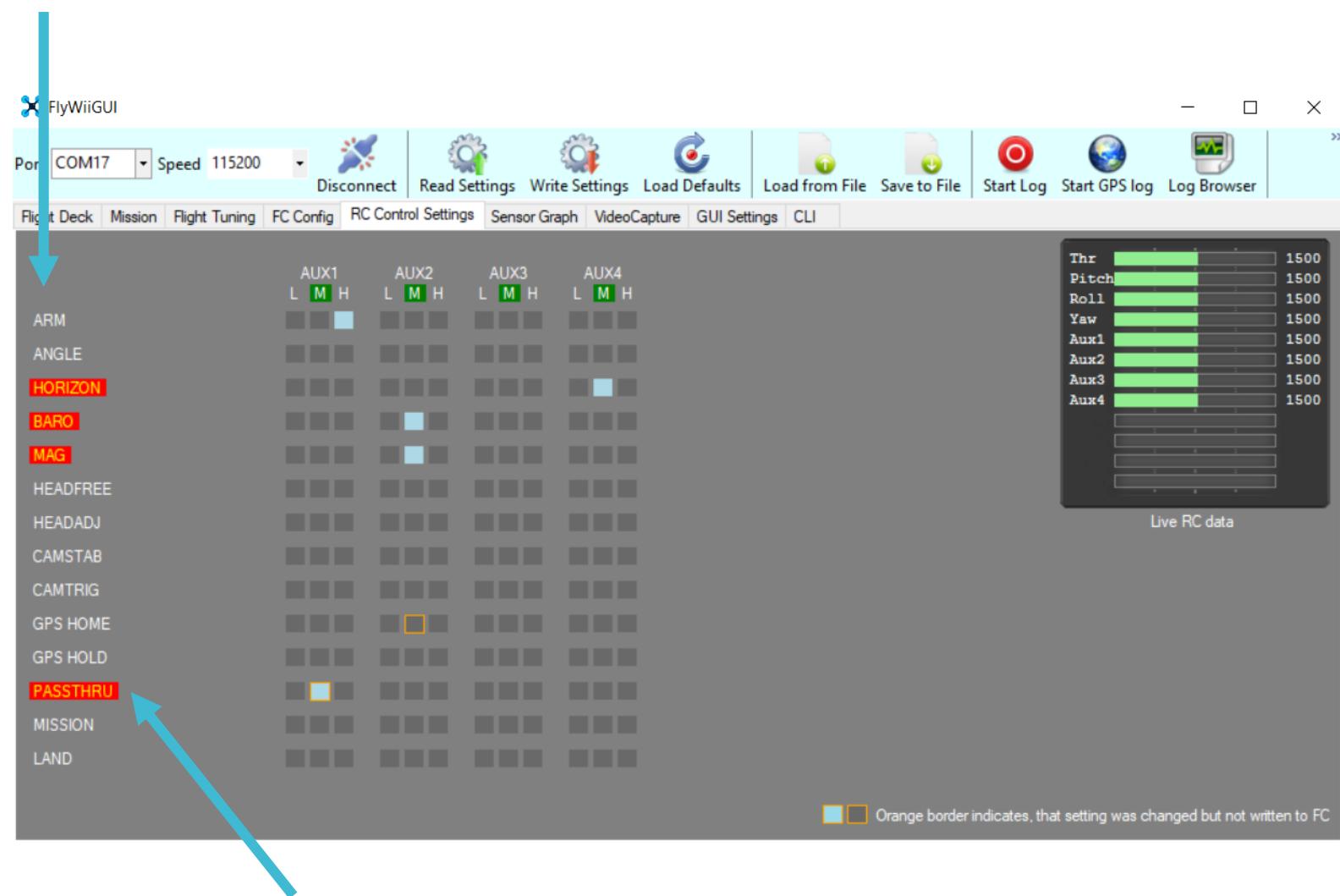
If it's not 1500 change #MIDRC to your Midrc in the code.

* Use the Servo Tab in Gui to adjust Servo Offset to center the Control Surfaces.

Adjust mechanically on the linkages to keep offsets as small as possible.

* Set the Servo Rates to maximum recommended for the model.

Remember to press Write settings button before exit!..



Pass thru - Mode disables all gyro stabilization puts the aircraft into manual flying

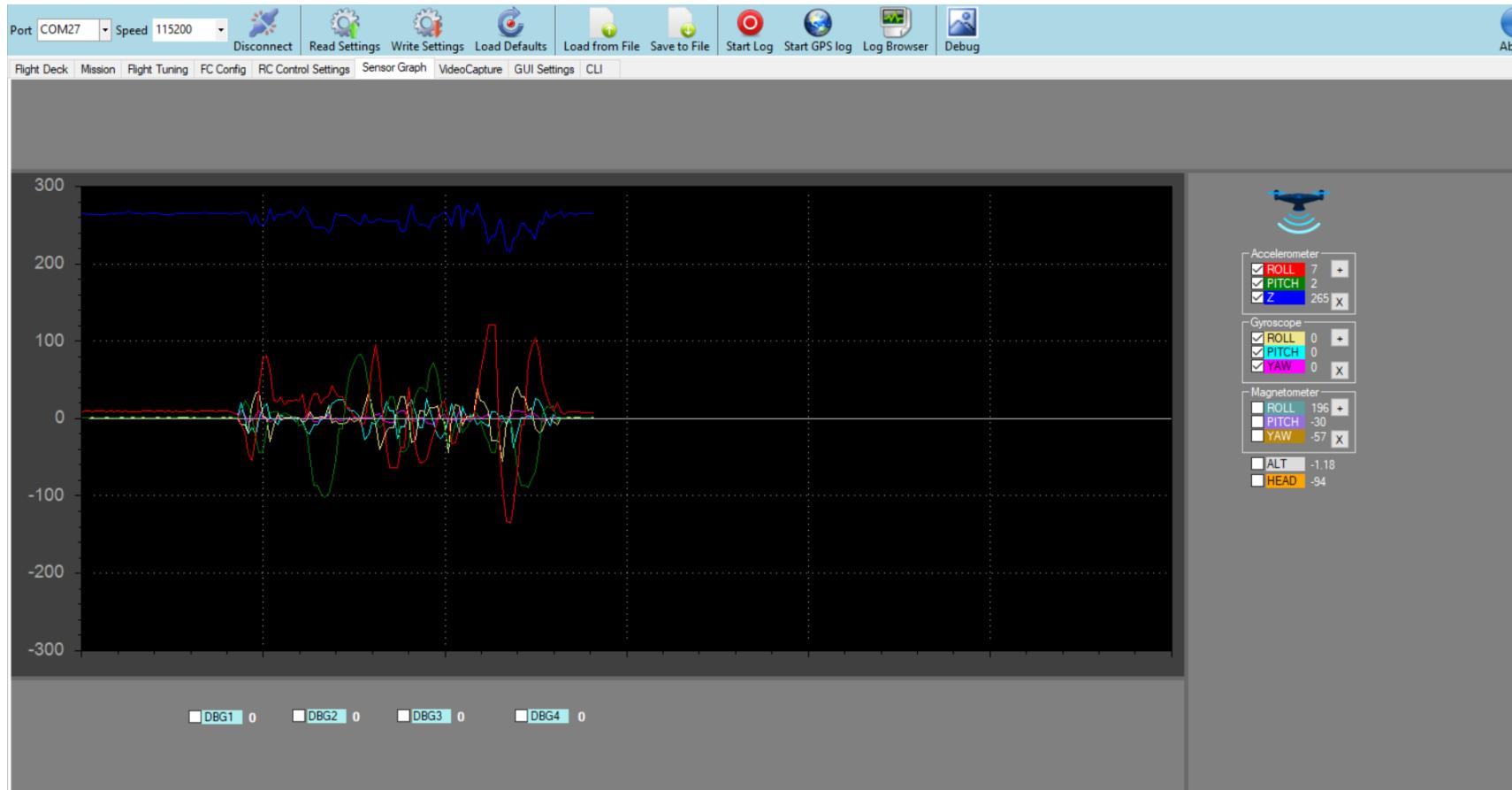
- **Acc Calibration**
 - Place the plane on a stable surface.
 - Wings Level with nose in expected Attack angle for level flight.
 - Normally a few degrees up.
 - Memorize the planes attitude in flight this should be Level for Acc.
 -
- ***Passthru***
 - Sends Rc commands direct to servos.
No influence from sensors.

- **Gyro Mode (Acro)**
 - This is "Normal" mode when nothing else is selected.
 - The plane should compensate for movements. (Wind Gusts etc)
 - The plane feels stable and locked in but still able to loop & roll.
 - Stall speed is lower and it can be necessary to "Push" it down in landings.
- **Stable Modes**
 - With the sticks centered the plane will self stabilize.
Returning to level flight from almost any situation.
 - Provided there's enough Altitude for recovery.
 -
- **Horizon Mode** Allows rolls and loops. Levels with centered sticks
This is a comfortable flight mode for FPV.
- **Angle Mode** also limits how much the plane can tilt.
Gives a Stiff feeling and is only recommended for beginners.



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



the correct orientation

Roll Right + no#

Pitch nose down + No#

Z up + No#

Roll Right + no#

Pitch nose down + No#

Yaw Right +No#

Mag & HEAD degrees
corresponds to the compass

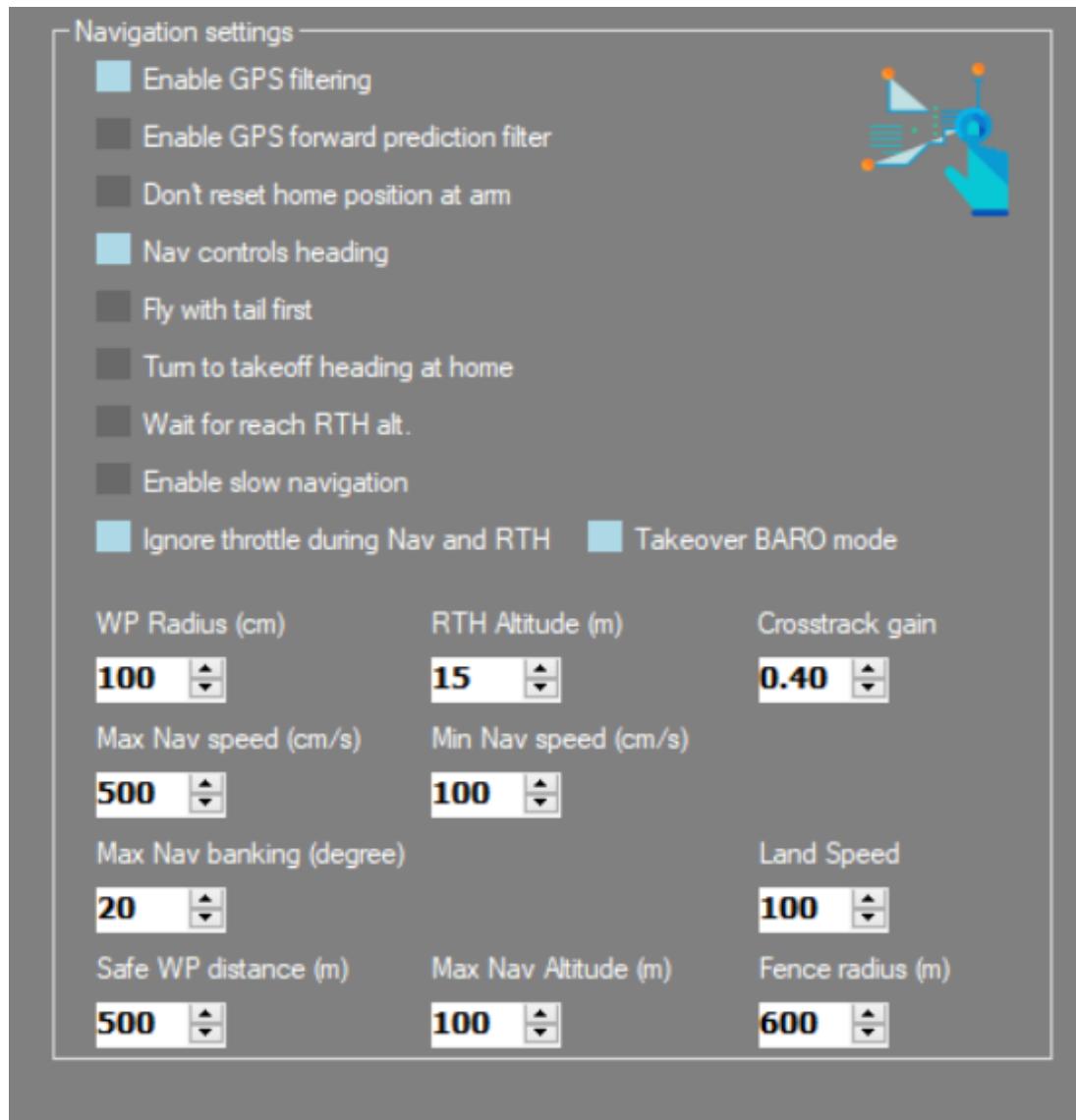
(0 degrees = North)

Alt up +no#

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

Other Navigation Functions



WP Radius – the radius of the area the Pos PID will trigger it has reached 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 within the WP Radius

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

Max Nav Banking – the max allowable pitch and roll the drone will be set to 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 capped to fly at

Land Speed – speed of descending for Landing cm/s

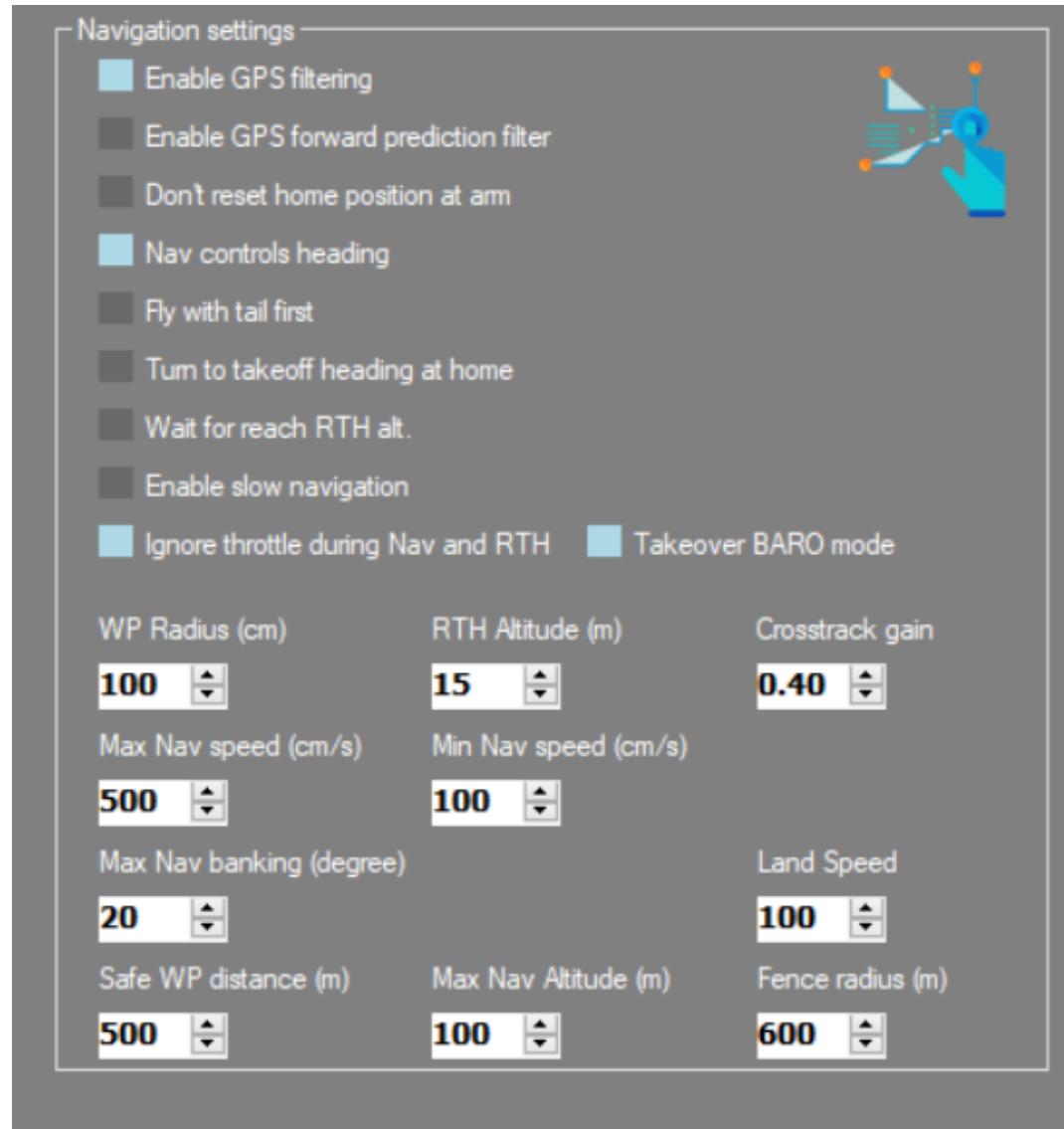
Safe WP Distance – max distance between waypoint before it's null out

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

CrossTrack gain - this tunes the GPS and Nav sensitivity

GPS Filtering – used to enhance GPS accuracy

GPS Forward Prediction Filter – predicting the drone's location and to compensate for lag . (optional) – not necessary for most applications



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 Important to fixwings

Fly tail first – Not applicable in fixwings

Turn take off heading at Home – when drone arrives at home position it orients 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

Important Navigation Settings for Fixwing

- Enable GPS Filtering ON
- Nav Control Heading ON
- Flywith Tail First OFF
- Turn Takeoff Heading OFF
- Wait for Reach RTH alt OFF
- Enable slow navigation OFF

Flight Tuning

FlyWiiGUI

Port COM18 Speed 115200

Disconnect Read Settings Write Settings Load Defaults

Load from File Save to File Log Browser Start Log Start GPS log

Flight Deck Mission Flight Tuning FC Config RC Control Settings Sensor Graph VideoCapture GUI Settings CLI

Flight Tuning

Roll
P 3.3 I 0.030 D 23
Pitch
P 3.3 I 0.030 D 23
Yaw
P 4.8 I 0.045 D 0
Altitude
P 3.0 I 0.020 D 45
PosHold
P 0.11 I 0.00
PosHoldRate
P 2.0 I 0.08 D 0.050
Navigation Rate
P 3.0 I 0.20 D 0.045
Level
P 9.0 I 0.010 D 100
Mag
P 4.0

RC Expo 0.00
RC Rate 0.90

Thr. MID 0.50
Thr. EXPO 0.50

Rates/Expo
Roll/Pitch RATE 0.00
Yaw RATE 0.00
Throttle PID attenuation 0.50

Navigation settings

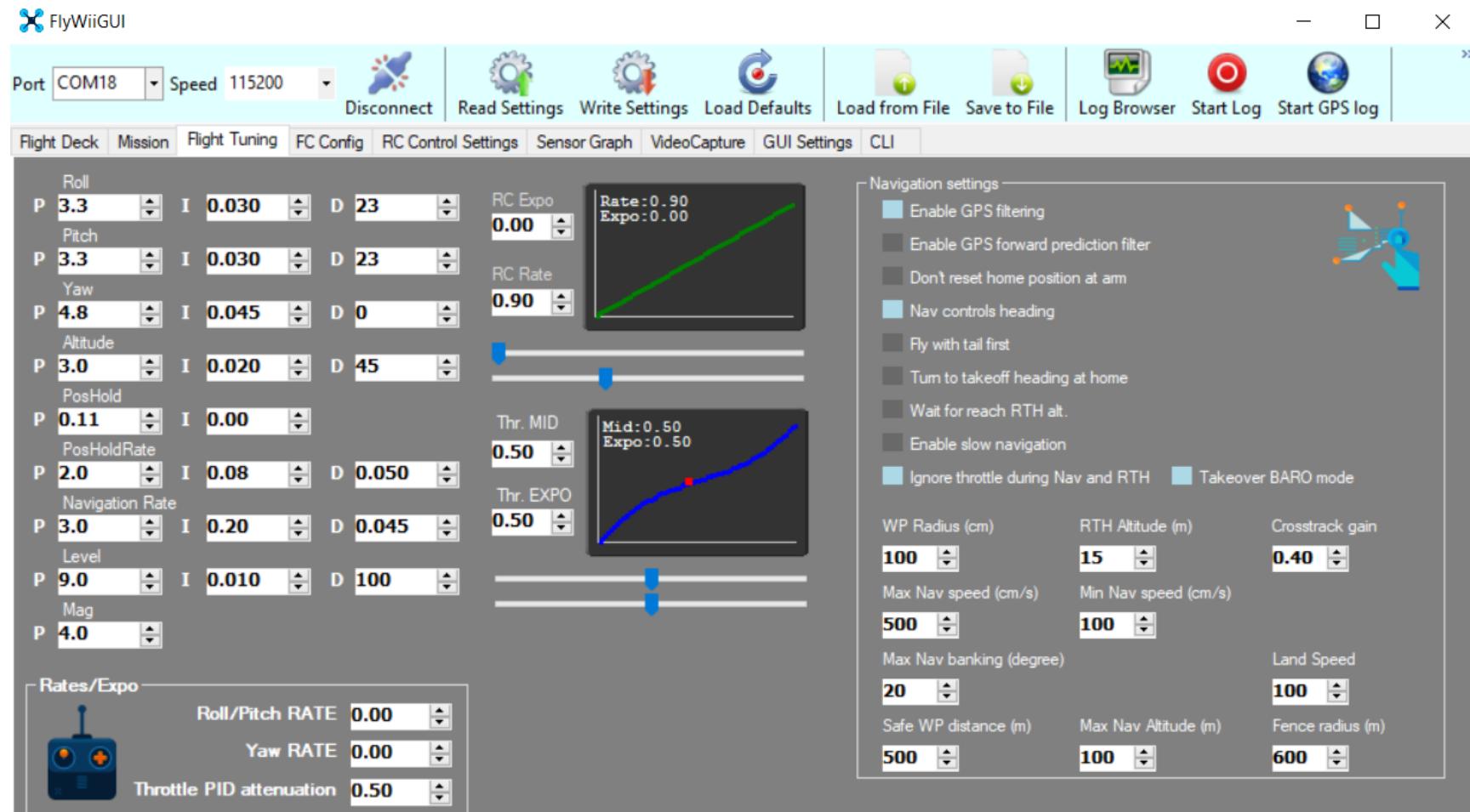
- Enable GPS filtering
- Enable GPS forward prediction filter
- Don't reset home position at arm
- Nav controls heading
- Fly with tail first
- Turn to takeoff heading at home
- Wait for reach RTH alt.
- Enable slow navigation
- Ignore throttle during Nav and RTH
- Takeover BARO mode

WP Radius (cm) 100 RTH Altitude (m) 15 Crosstrack gain 0.40

Max Nav speed (cm/s) 500 Min Nav speed (cm/s) 100

Max Nav banking (degree) 20 Land Speed 100

Safe WP distance (m) 500 Max Nav Altitude (m) 100 Fence radius (m) 600



Stability	Roll,Pitch,Yaw Gyro	PID : Level of your Gyro
Horizon / Level Mode	X,Y Accelerometer	PID : X ,Y Axis Level of your Accelerometer
Heading Lock	Compass/Mag	PID : heading of your magnetometer Calibration
Altitude Hold	Barometer / Z	PID : Barometer and Z accelerometer
Position Hold	GPS Pos	PID : sensitivity of GPS position reaction
Navigation Rate	GPS Nav	

Understanding impact of P, I and D

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

The amount of force is proportional to a combination of 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 destabilized.

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 Aircraft 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 destabilized.

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

Aerobic 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 Aircraft 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

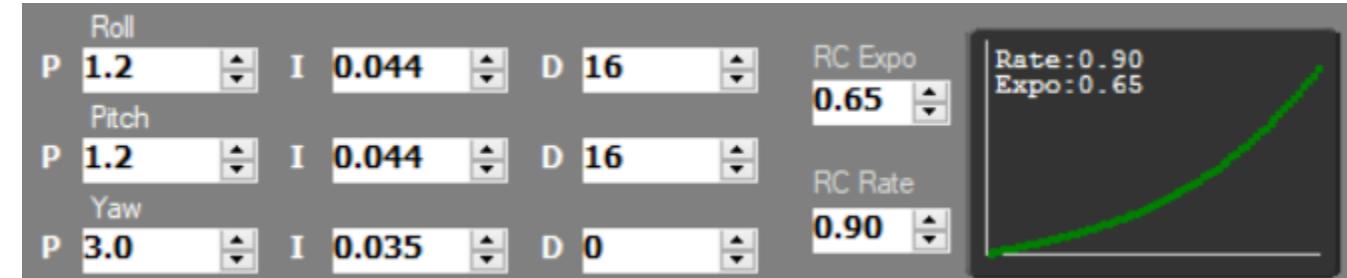
Aerobatic flight: Lower D

Gentle smooth flight: Increase D



Basic PID Tuning – on the ground

- 1 - Set PID to the designers default recommended settings.
- 2 - Hold the Aircraft securely and safely in the air.
- 3 - Increase throttle to the hover point where it starts to feel light.
- 4 - Try to lean the Aircraft 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 stabilization you will feel it allow you to move over a period of time. That is OK
- 6 - Now try rocking the aircraft. 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.



Click on Write settings
after changes made in
any of the parameters to
save

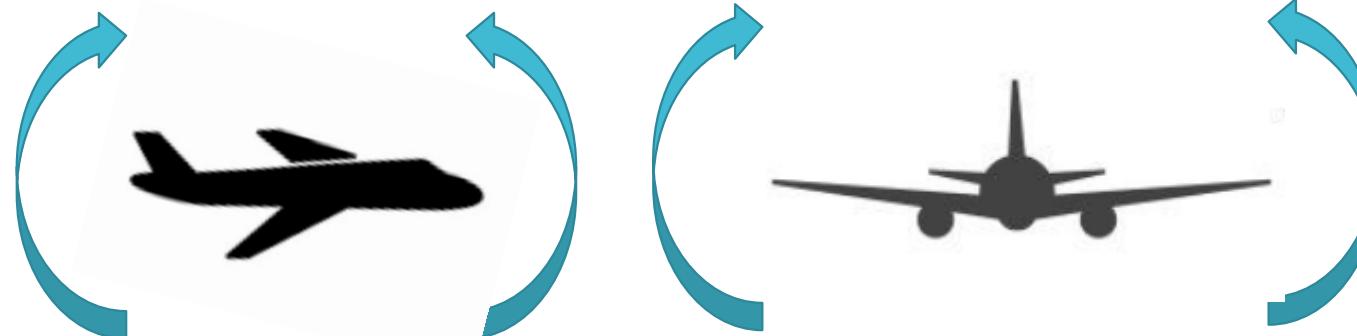


You will have to accept a compromise of optimal settings for stable hover and your typical mode of flying.

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 Aircraft 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 off 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 off quite a bit

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



Click on Write settings after changes made in any of the parameters to save

PID : Level

Level	
P	8.0
I	0.002
D	80

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 measured angle error persists

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



Click on Write settings after changes made in any of the parameters to save

The Barometer sensor is used to detect the altitude of your 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.

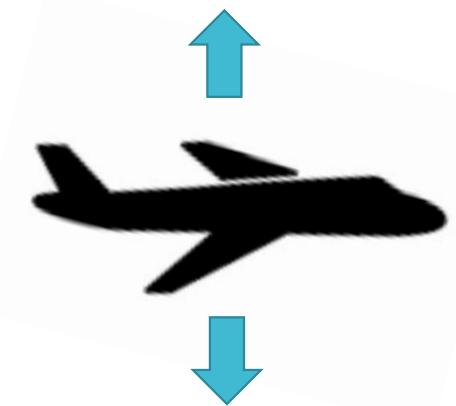
So small up and down movements are detected by the accelerometer Z axis.
Combination of these two sensors gives good altitude hold.

PID settings for ALT works like this:

P - Is how much the Aircraft 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 Aircraft 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 Aircraft will react to small altitude changes.



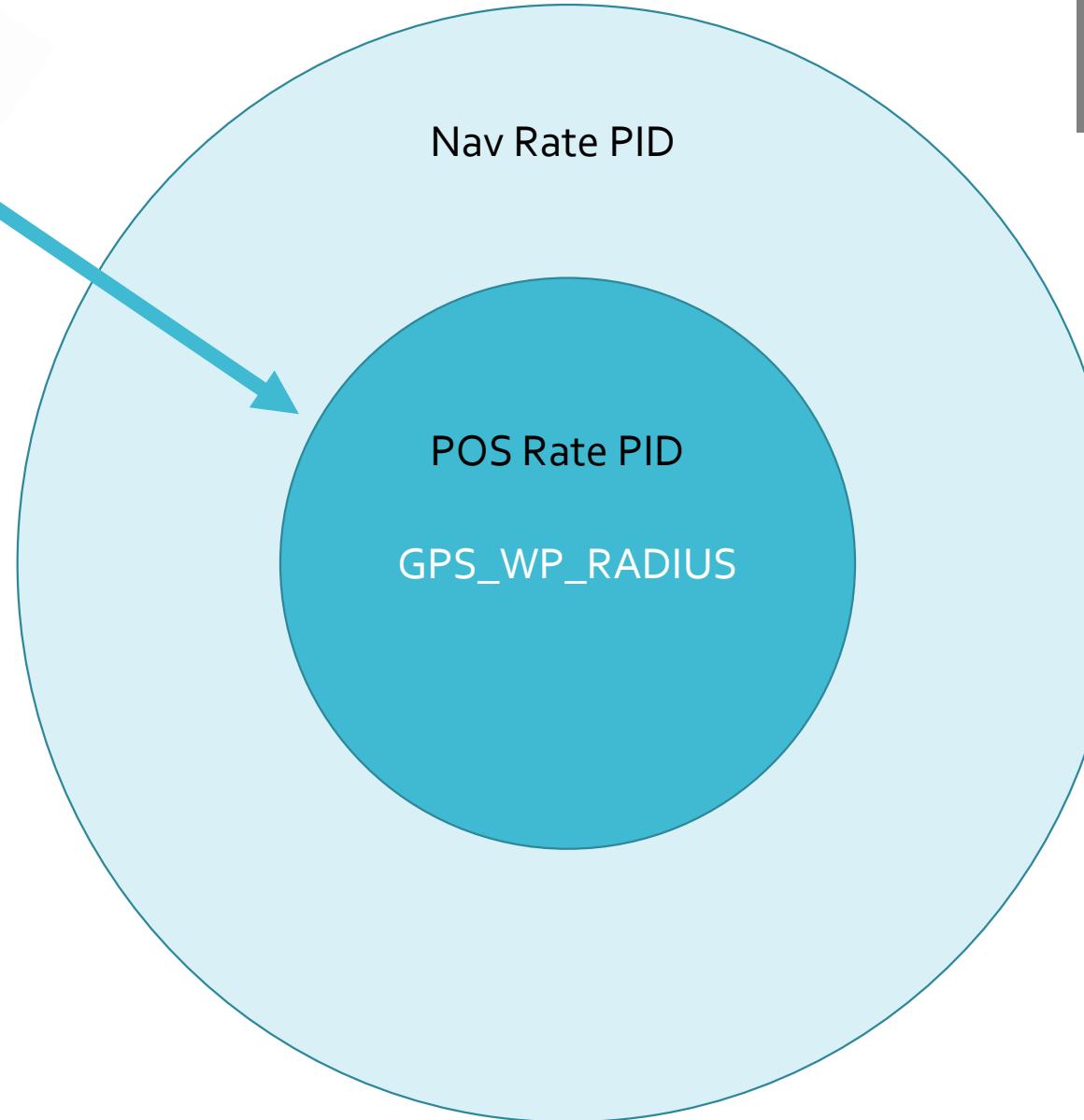
1. So set the P and I to 0
2. Start to play with D value only. To high D may cause yoyo effect (up and down oscillations). With to low D copter will be not able to react strong/fast enough to hold altitude. Your goal here is to set D to the value when copter don't oscillate up and down and also holds altitude quite well for a not very long period of time. Aircraft 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 Aircraft 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. Aircraft 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 Vehicle 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 Aircraft is controlled by Pos Rate.
- When the Aircraft 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 Vehicle 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 Aircraft is controlled by Pos Rate.
 - When the Aircraft 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 Aircraft begins to position hold with some drift.
- Gradually increase D until the Aircraft 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.

- To tune the Pos Rate PID, initially set P, I and D values to 0.
- Gradually increase P until the Aircraft begins to position hold with some drift.
- Gradually increase D until the aircraft 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

The Aircraft suppose to heading towards the waypoint

When the aircraft goes pass the waypoint

Mission – fly to next waypoint

GPS Hold – Aircraft would orbit the Waypoint

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

Enter loiter orbit mode

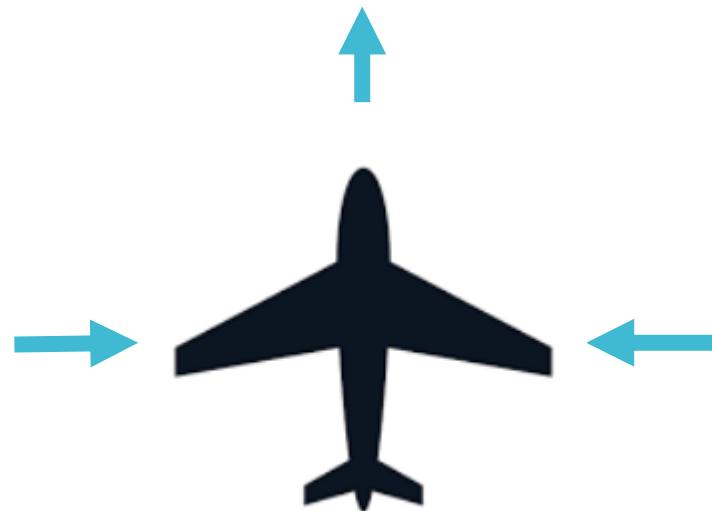


To tune the Pos Rate PID, initially set P, I and D values to 0.

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

Gradually increase D until the Aircraft 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.

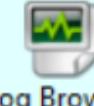
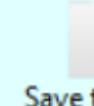




MyWiiGUI

Port COM17

Speed 115200



>>

Flight Deck

Mission

Flight Tuning

FC Config

RC Control Settings

Sensor Graph

VideoCapture

GUI Settings

CLI

ARM

	AUX1	AUX2	AUX3	AUX4
L	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
M	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
H	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

ANGLE

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

HORIZON

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

BARO

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

MAG

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

HEADFREE

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

HEADADJ

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

CAMSTAB

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

CAMTRIG

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

GPS HOME

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

GPS HOLD

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

PASSTHRU

	AUX1	AUX2	AUX3	AUX4
L	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

MISSION

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

LAND

	AUX1	AUX2	AUX3	AUX4
L	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
M	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

RC Control Settings

Use Aux switch to setup flight modes and Navigation functions

HORIZON- keeps the aircraft Level

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

BARO – Altitude Hold

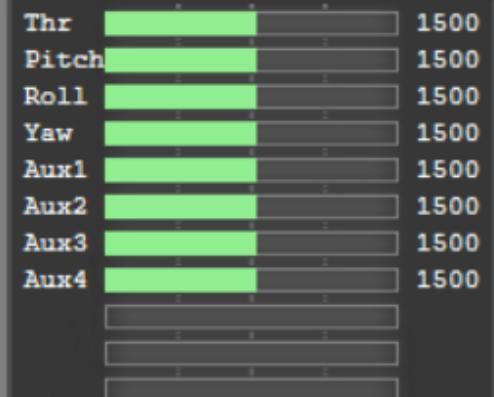
MAG –Heading Hold

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

GPS Hold – Hold Position ,

MISSION – fly a mission save from the mission tab

PASSTHRU- Disable all gyro Functions Manual



Live RC data

Orange border indicates, that setting was changed but not written to FC

- Preferably to reserve 1 Aux channel as Arm switch as combination stick could make the vehicle move prematurely
- Then the alternative Aux for the Mission or RTH mode

Passthru

Sends Rc commands direct to servos.

No influence from sensors.

Gyro Mode (Acro)

This is "Normal" mode when nothing else is selected.

The plane should compensate for movements. (Wind Gusts etc)

The plane feels stable and locked in but still able to loop & roll.

Stall speed is lower and it can be necessary to "Push" it down in landings.

Stable Modes

With the sticks centered the plane will self stabilize.

Returning to level flight from almost any situation.

Provided there's enough Altitude for recovery.

Horizon Mode Allows rolls and loops. Levels with centered sticks

This is a comfortable flight mode for FPV.

Angle Mode also limits how much the plane can tilt.

Gives a Stiff feeling and is only recommended for beginners.



Missions

Note: Only functional for Mega 2560 Boards with GPS

Waypoint – the drone with travel between those points

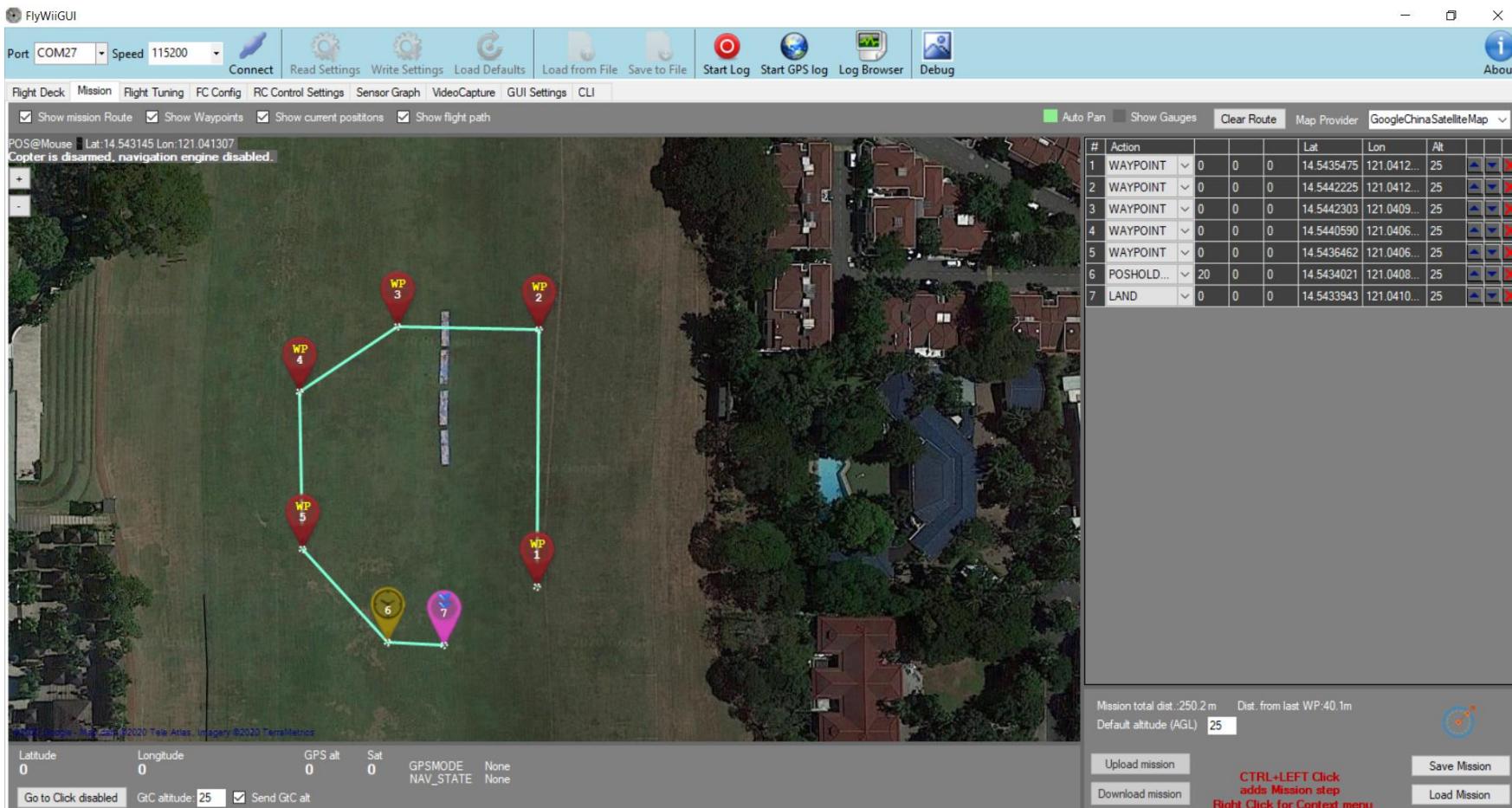
Time PosHold – Drone will wait X number of 00:00:00 then move to the next waypoint

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 .

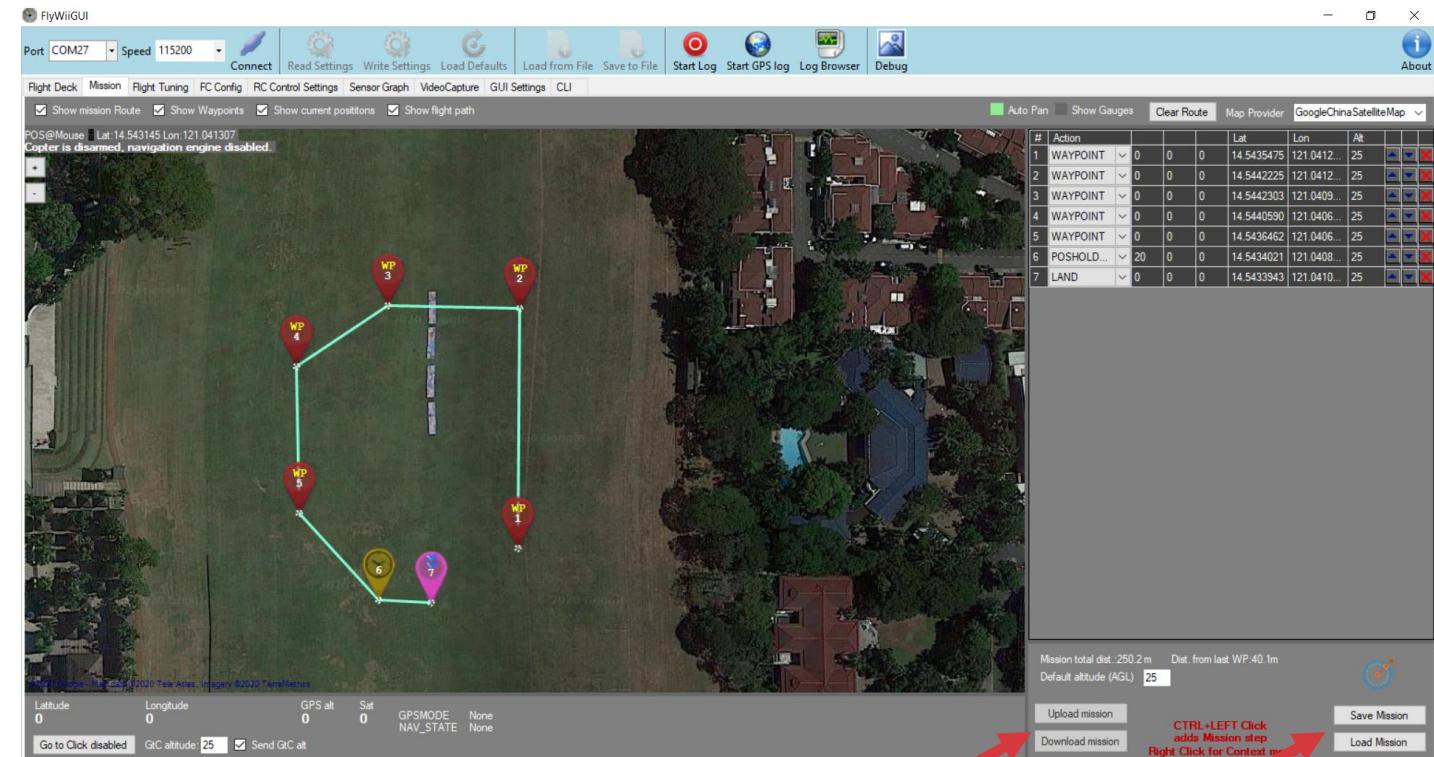
Any time you can switch out of it on hold or stabilize mode



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 ,Horizon/Angle 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

Preflight setup

After you have changed the servo Rates

you can set Dual rates and Expos in the Transmitter.

Engine must be *Armed* to prevent motorstart by accident..

It can be Armed from AXU-channel if it's setup in the gui . (recommended)

Or with stick combination min throttle & max rudder.

First Flight.

Take off in Passthru.

Switch mode on safe height.

Activate Assisted modes (Horizon /Angle) and feel the difference.

Level-P value will Reduce the maximum throw in Level-Mode.

GPS Flight Modes

Missions

Puts the aircraft into waypoint mode set Aux switch to Mission & Horizon or Angle mode

GPS-Hold

Can only be enabled With AUX switch.

Check both Angle mode and GPS Hold in Gui.

When GPS-Hold is activated The position is saved in a 3D Waypoint.

The plane will Navigate and try to "hit" the WP continuously and maintaining altitude.

No pattern is programmed and the plane fly the shortest way back.

Often in a circle or figure eight.

RTH (Return to home)

Can be enabled With AUX switch or by Failsafe.

Check both Angle mode and RTH in Gui.

When RTH is activated the plane will start Climb to reach safe Altitude.

If RTH is enabled Higher than set altitude it will start navigation and keep that altitude.

If altitude is safe the plane will start to Navigate to home Position.

Only use Angle/Horizon + Gps Home/Hold together.

Do **NOT** Activate BARO Or MAG for navigation.

It will interfere with the navigation code.

When the plane reaches **SAFE_DECSCEND_ZONE** the plane will begin descending to correct altitude.

The plane will keep flying in hold mode and continuously pass home.

If Failsafe is active at return The plane will Disarm motor and descend to a "Landing"

PID settings is made in Gui.
ALT & NavR.

Return Altitude can be set with
RTH Altitude
Scale is in Meters
15m is set as default.

WP Radius (cm)	RTH Altitude (m)	Crosstrack gain
400 <input type="button" value="▲"/> <input type="button" value="▼"/>	0 <input type="button" value="▲"/> <input type="button" value="▼"/>	0.40 <input type="button" value="▲"/> <input type="button" value="▼"/>
Max Nav speed (cm/s)	Min Nav speed (cm/s)	
400 <input type="button" value="▲"/> <input type="button" value="▼"/>	100 <input type="button" value="▲"/> <input type="button" value="▼"/>	
Max Nav banking (degree)	Land Speed	
20 <input type="button" value="▲"/> <input type="button" value="▼"/>	100 <input type="button" value="▲"/> <input type="button" value="▼"/>	
Safe WP distance (m)	Max Nav Altitude (m)	Fence radius (m)
500 <input type="button" value="▲"/> <input type="button" value="▼"/>	100 <input type="button" value="▲"/> <input type="button" value="▼"/>	600 <input type="button" value="▲"/> <input type="button" value="▼"/>

PID setup and config

FlyWiiGUI

Port COM18 Speed 115200 Disconnect Read Settings Write Settings Load Defaults Load from File Save to File Log Browser Start Log Start GPS log

Flight Deck Mission Flight Tuning FC Config RC Control Settings Sensor Graph VideoCapture GUI Settings CLI

Roll
P 3.3 I 0.030 D 23 RC Expo 0.00 Rate: 0.90 Expo: 0.00

Pitch
P 3.3 I 0.030 D 23 RC Rate 0.90

Yaw
P 4.8 I 0.045 D 0

Altitude
P 3.0 I 0.020 D 45

PosHold
P 0.11 I 0.00

PosHoldRate
P 2.0 I 0.08 D 0.050

Navigation Rate
P 3.0 I 0.20 D 0.045

Level
P 9.0 I 0.010 D 100

Mag
P 4.0

Rates/Expo
Roll/Pitch RATE 0.00
Yaw RATE 0.00
Throttle PID attenuation 0.50

Navigation settings

- Enable GPS filtering
- Enable GPS forward prediction filter
- Don't reset home position at arm
- Nav controls heading
- Fly with tail first
- Turn to takeoff heading at home
- Wait for reach RTH alt.
- Enable slow navigation
- Ignore throttle during Nav and RTH
- Takeover BARO mode

WP Radius (cm) 100 RTH Altitude (m) 15 Crosstrack gain 0.40

Max Nav speed (cm/s) 500 Min Nav speed (cm/s) 100

Max Nav banking (degree) 20 Land Speed 100

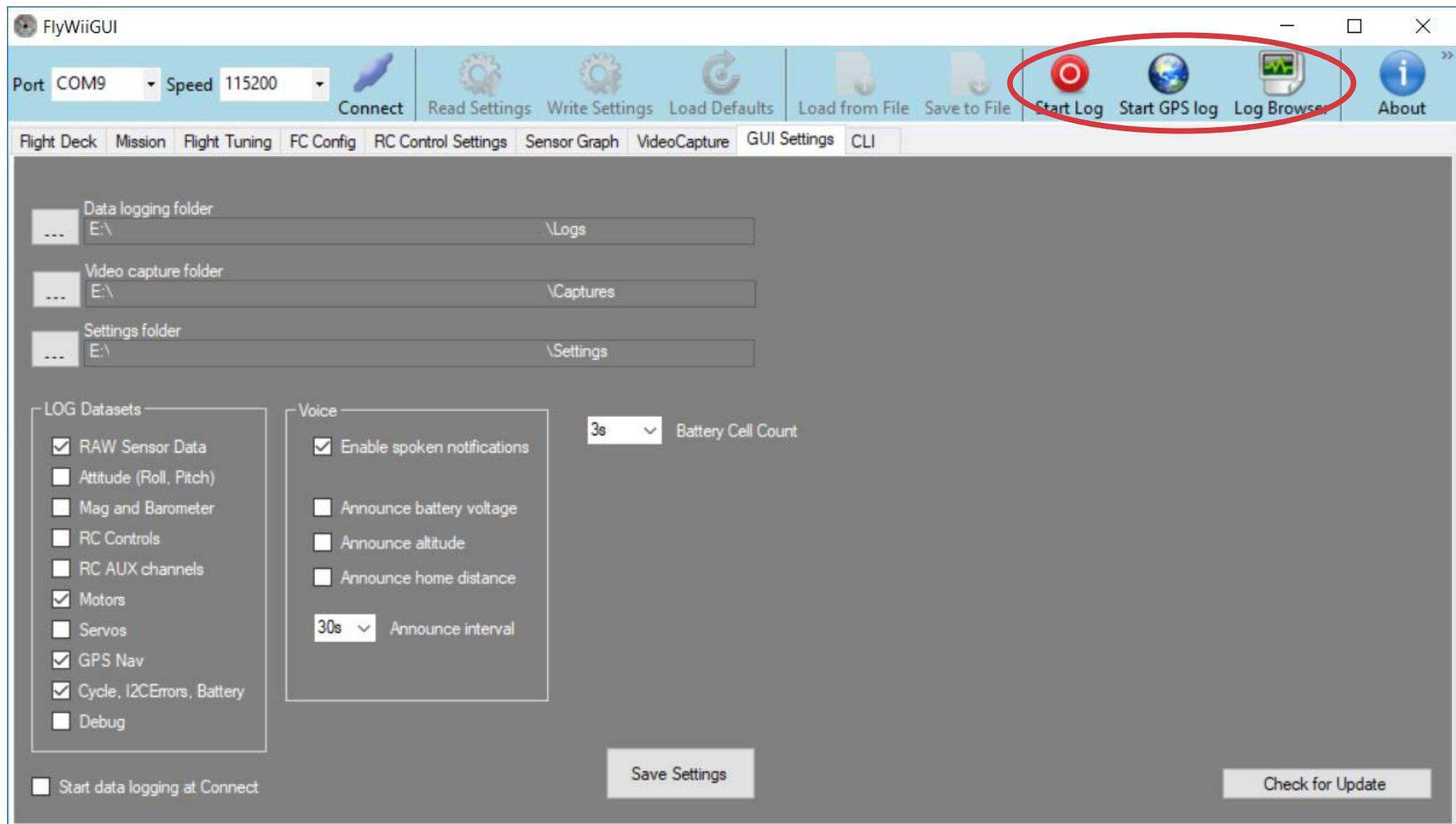
Safe WP distance (m) 500 Max Nav Altitude (m) 100 Fence radius (m) 600



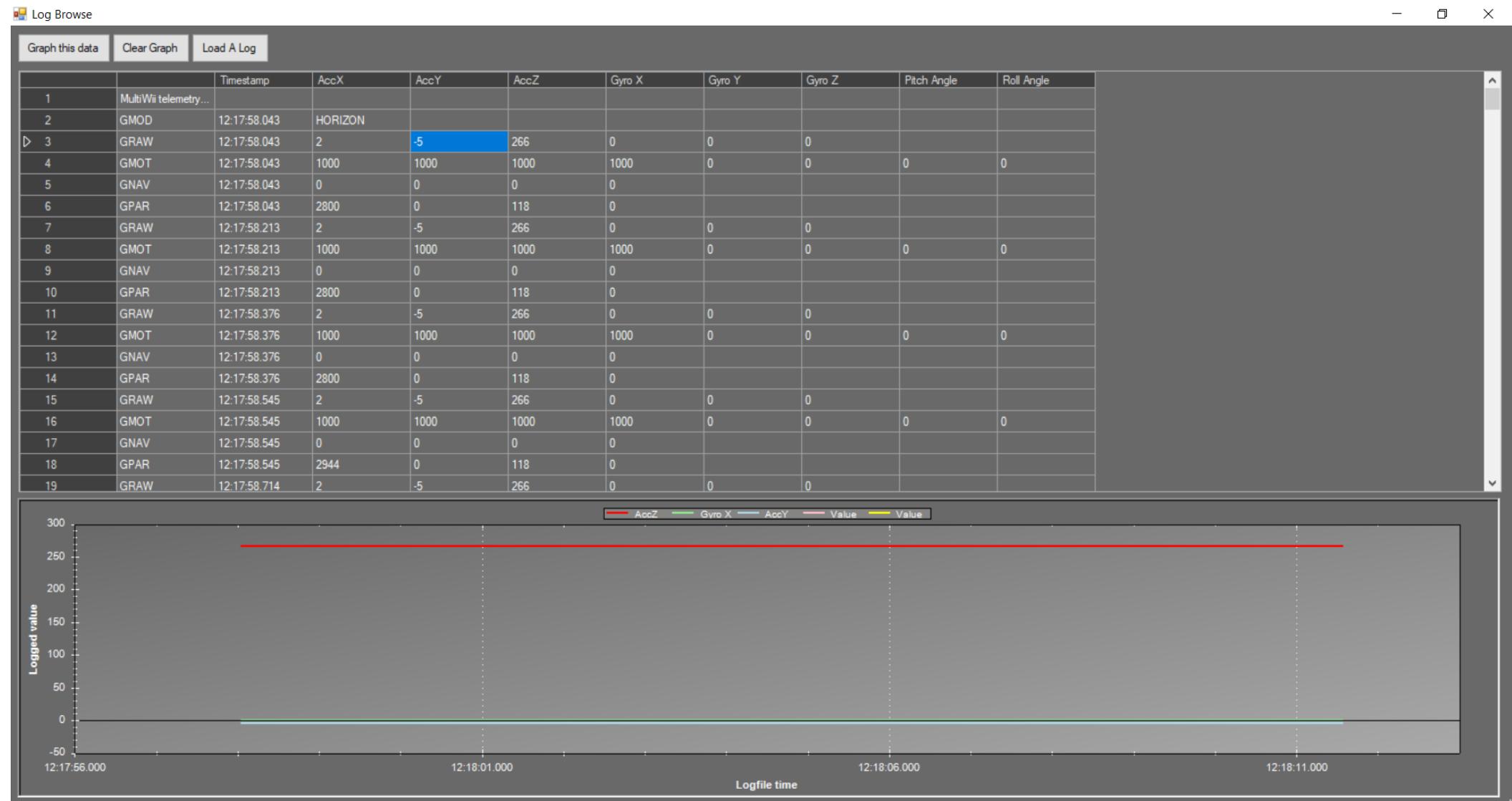
Graphs and Data Logging



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



Log Browser



Preflight setup

After you have changed the servo Rates
you can set Dual rates and Expos in the
Transmitter.

Engine must be *Armed* to prevent motorstart
by accident..

It can be Armed from AXU-channel if it's
setup in the gui . (recommended)

Or with stick combination min throttle &
max rudder.

In RC control Setting set Aux 1 as Arm

First Flight.

Take of in Passthru.

Switch mode on safe height.

Activate Assisted modes and feel the
difference.



And your much Done on your setup

Cannot Arm Motors

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

Tests motor with Props off

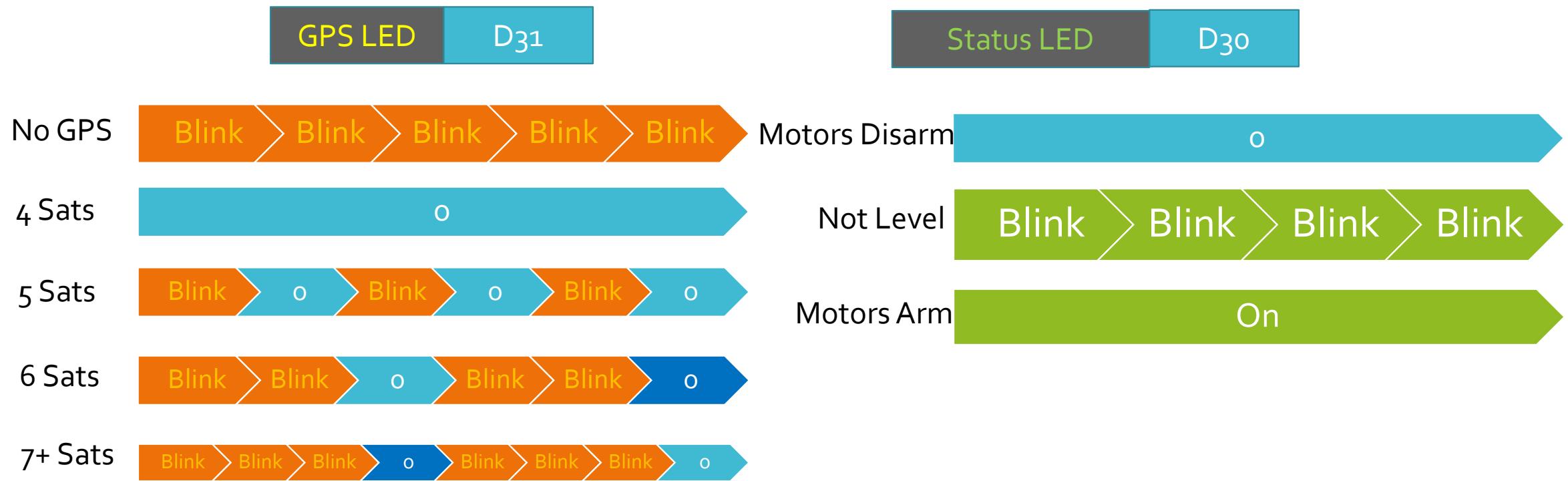
Baro and Mag preferably switch off when Arming

Pls calibrate ACC and Mag in the FlyWii GUI Dashboard

Ensure the compass is facing the correct orientation



LED Indicator



indicate a valid GPS fix by flashing the LED

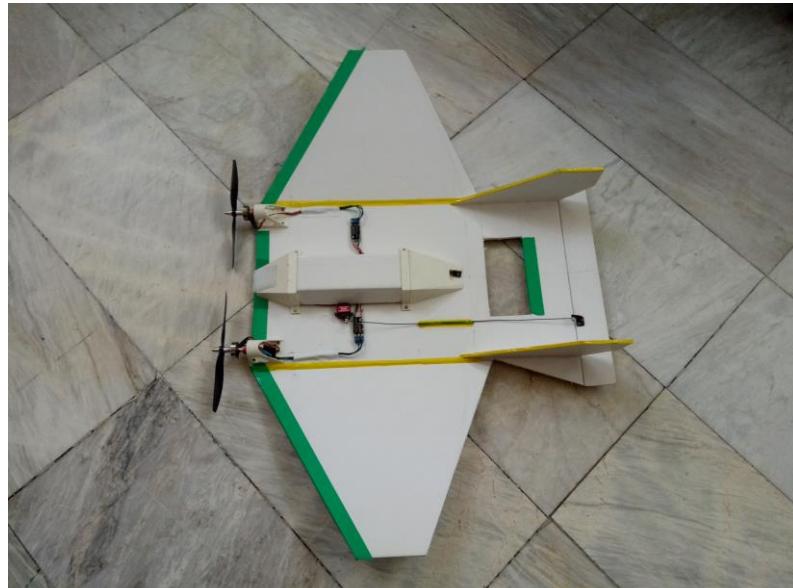
- led work as sat number indicator
- No GPS FIX -> LED blinks constant speed
- Fix and sat no. below 5 -> LED off
- Fix and sat no. >= 5 -> LED blinks, one blink for 5 sat, two blinks for 6 sat, three for 7 +



Conventional 4Ch AETR



Conventional 3Ch RET



Differential Thrust 3Ch RET



Flying Wing 3Ch

First timer for Autonomous RC Aircraft

We recommend that you choose a sizeable slow flying plane.

Either build from scratch or an available kit

Large enough to fit all your equipment and slow enough

To fly comfortably

This way its easy to tune and flown manually should needed