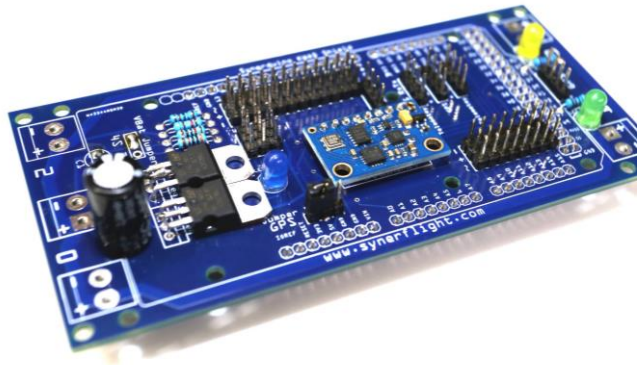
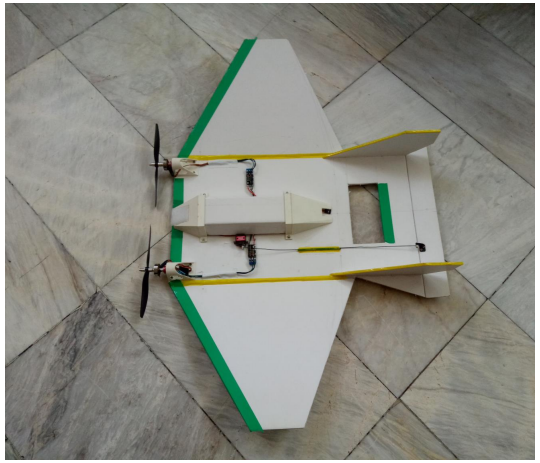
Synerduino Special Airplane



This segment Discussing setting up a Differential thrust plane with the Synerduino board

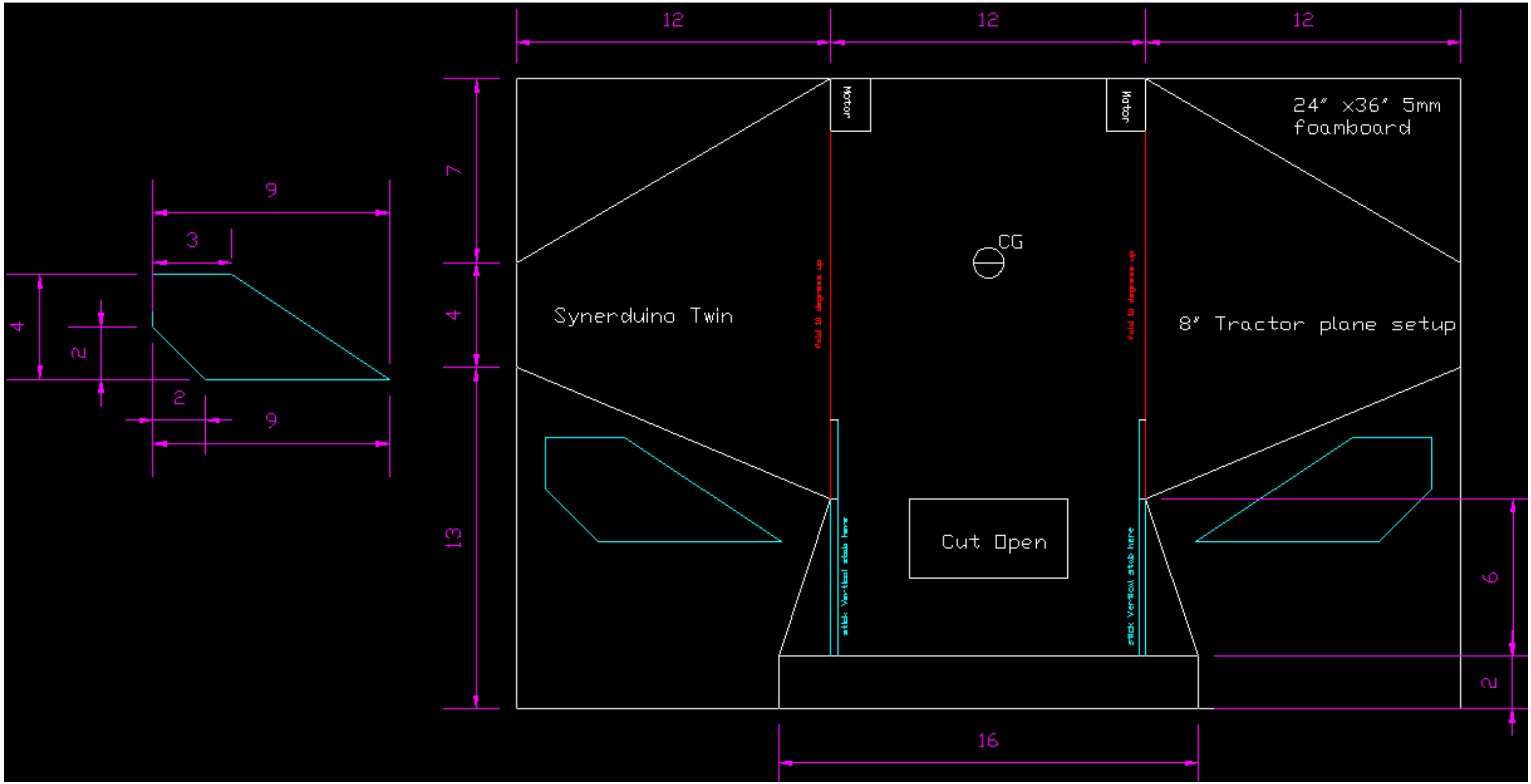
a Differential thrust Utilizing two motors to provide Yaw control and stability



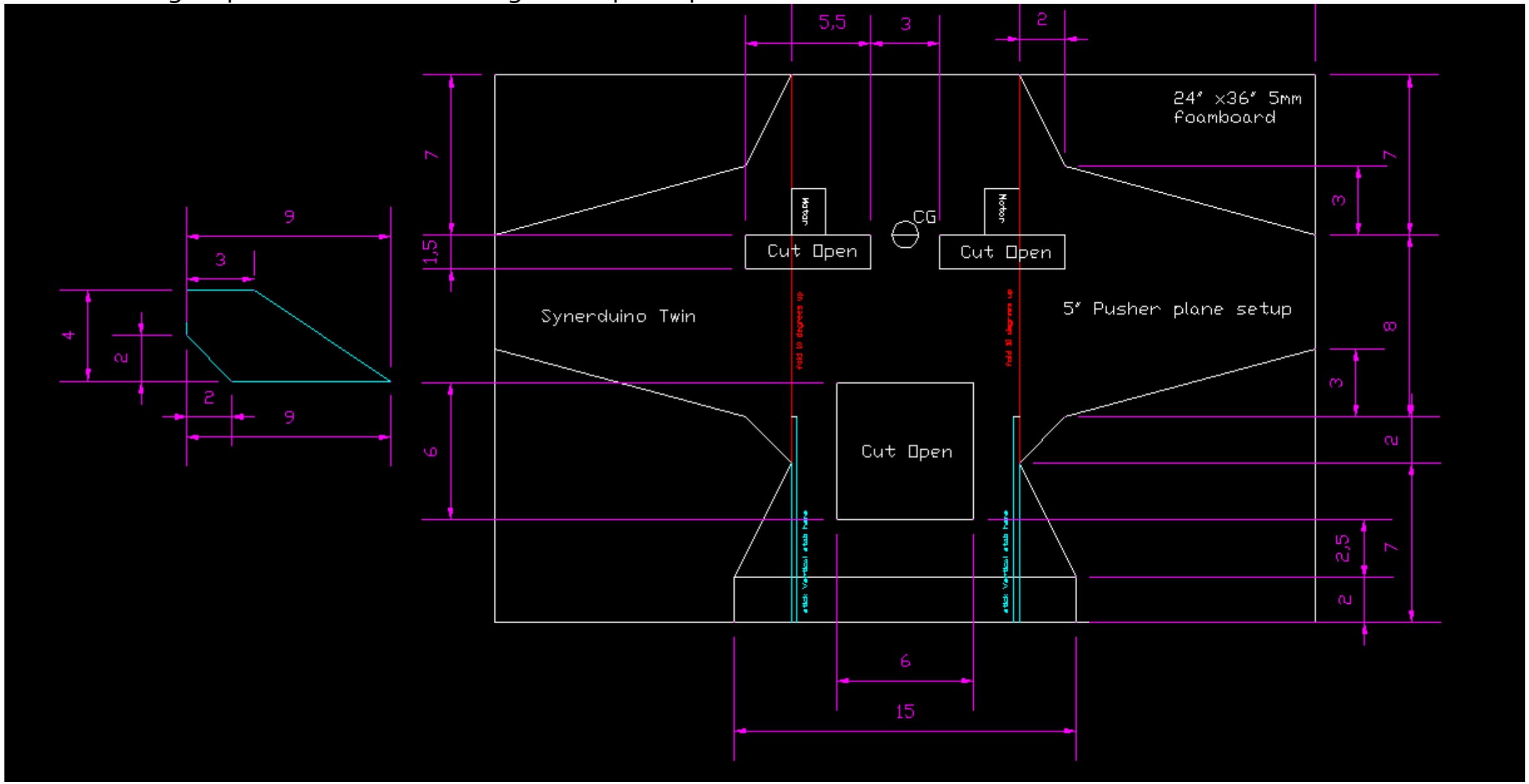
Hardware

- Synerduino shield
- Arduino2560 MEGA / Uno 328
- Fixwing (Airplane) Synerduino Plane Airframe
- 2x Brushless Motor 1500kv - 2300kv
- 2x ESCs 30A 2s-4s
- 2x Props 5045 or 8045 depending on setup
- 1x 9g Micro Servos and control Rod & horn set
- GPS
- Bluetooth

Synerduino Dart Design is base of the FT Flyer require 5mm 24”x36” foam board
Wings have a dihedral of 15-20 degrees at most



Hole in slot Wings is possible with such design as required pusher



Tools needed

Screwdriver

Cutter Knife

Soldering Set

tape

Zip Ties

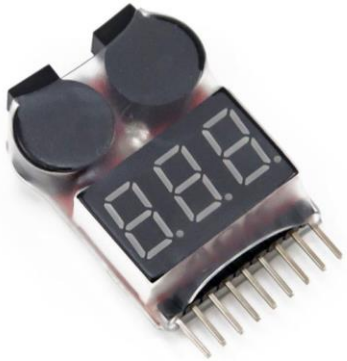
Pliers

Tools Needed

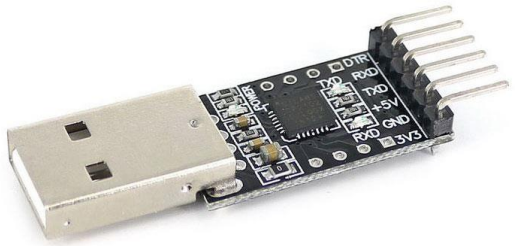
Lipo Charger 5A

Lipo Battery 3s 1300mah

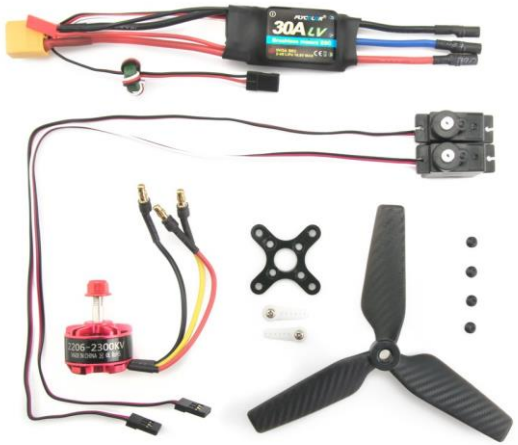
Voltage alarm

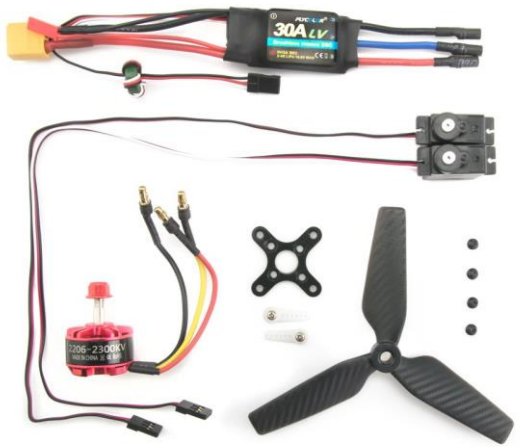


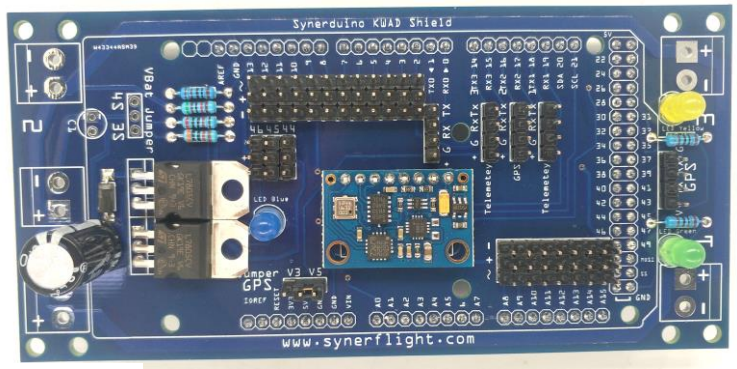
USB TTL FTDI



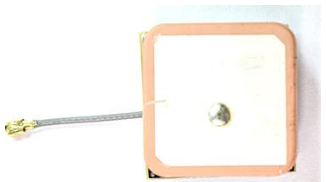
Hardware


30A LV
2206-2300KV

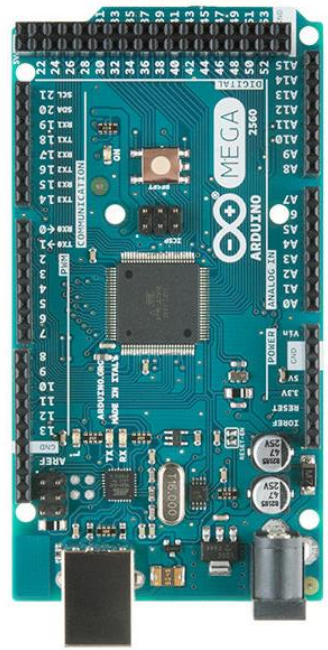

30A LV
2206-2300KV


SynerduIno KWAD Shield
www.synerflight.com
GPS


Park fly
FS-IA6
6 CHANNELS
2.4 GHZ RECEIVER
AFHDS 2A




GY-GPS6MV2
VCC RX TX GND


MEGA
ARDUINO

Synerduino Kwad Shield BETA GY801

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

ESC is Solder on Top side only

ESC / Servo PWM Out

Serial Pins

ESC is Solder on Top side only

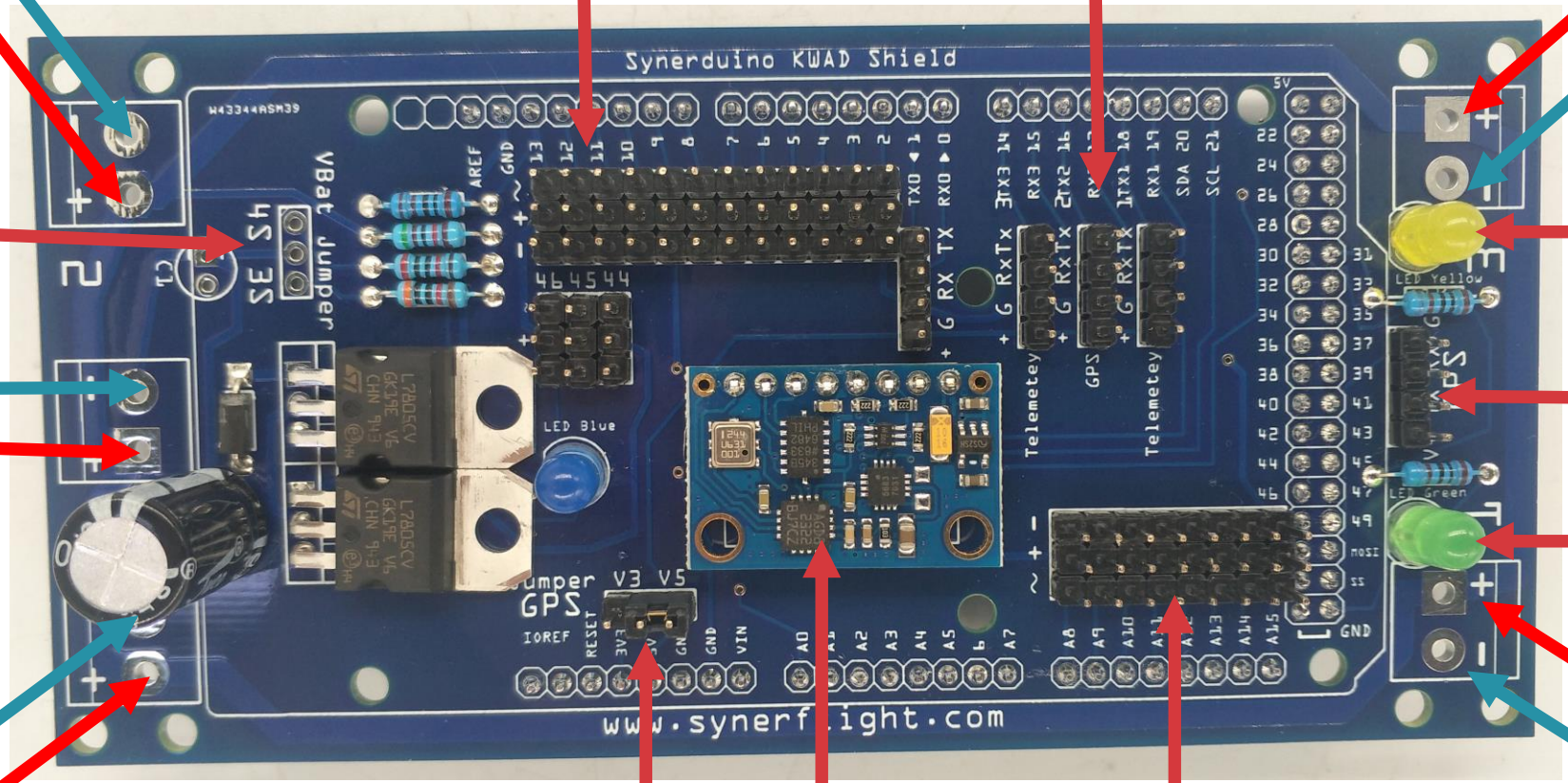
Set jumper to the Battery Cell count (Solded) 3s – 4s

GPS LED

Power input
3s 11.1V
4s 14.8V

GPS Serial Pins

Status LED



ESC is Solder on Top side only

GPS Voltage Jumper

RC PWM in

ESC is Solder on Top side only

For improve performance IMU must be protected from the Environment

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

Synerduino Kwad Shield V1.1 GY91

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

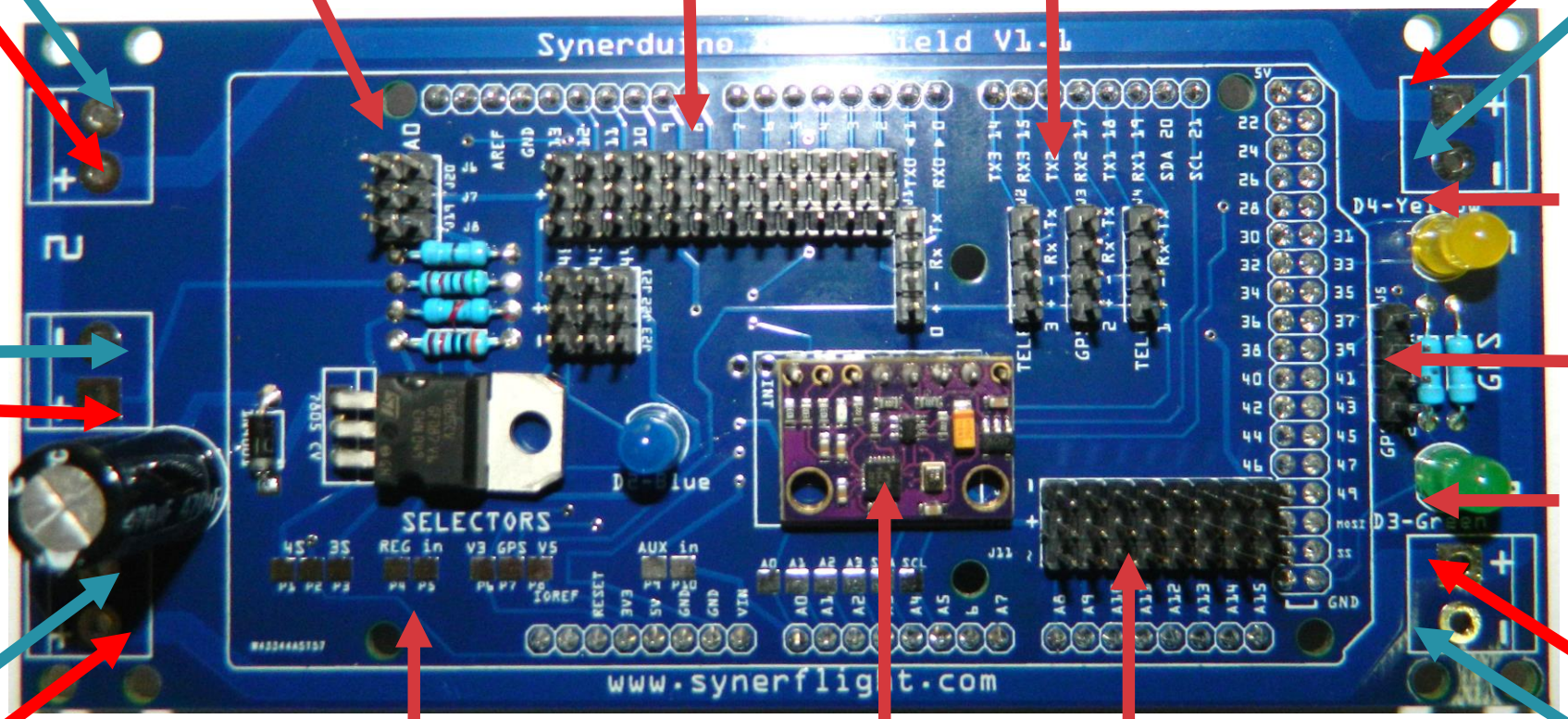
ESC is Solder on Top side only

Aux ADC in

ESC / Servo PWM Out

Serial Pins

ESC is Solder on Top side only



GPS LED

Power input 3s 11.1V

GPS Serial Pins

Status LED

ESC is Solder on Top side only

Jumper Pads Selector Zone

RC PWM in

ESC is Solder on Top side only

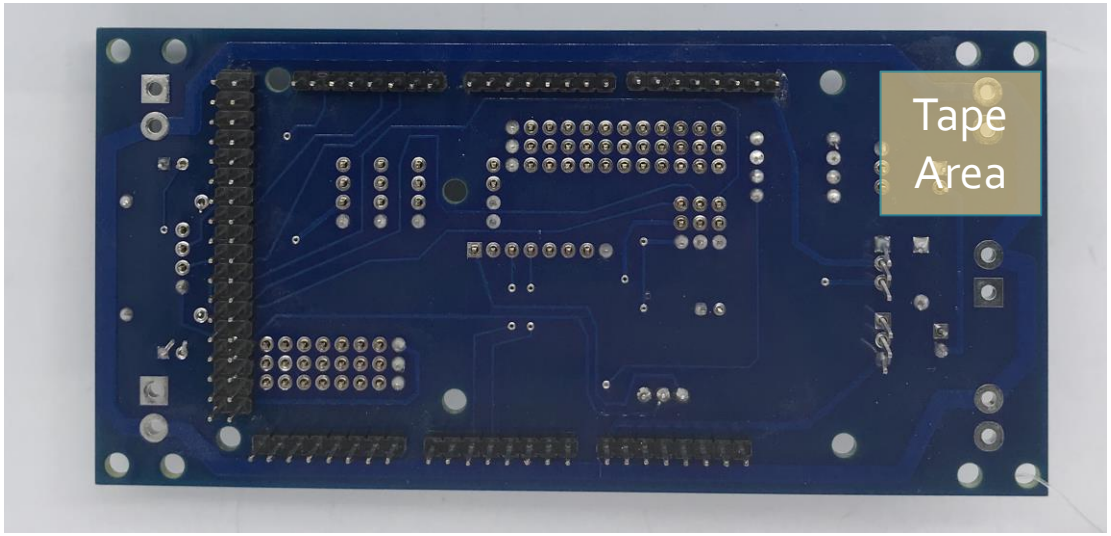
For improve performance IMU must be protected from the Environment

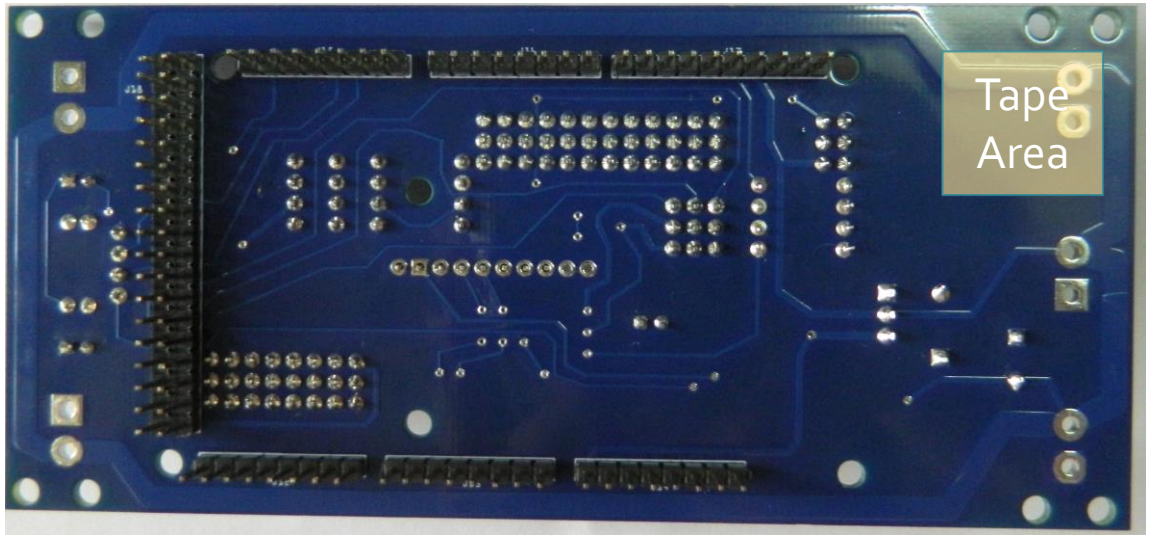
IMU : MPU-9250 & BMP280

Synerduino Kwad Shield Preparation

Ensure insulation from the Arduino board add tape on these areas

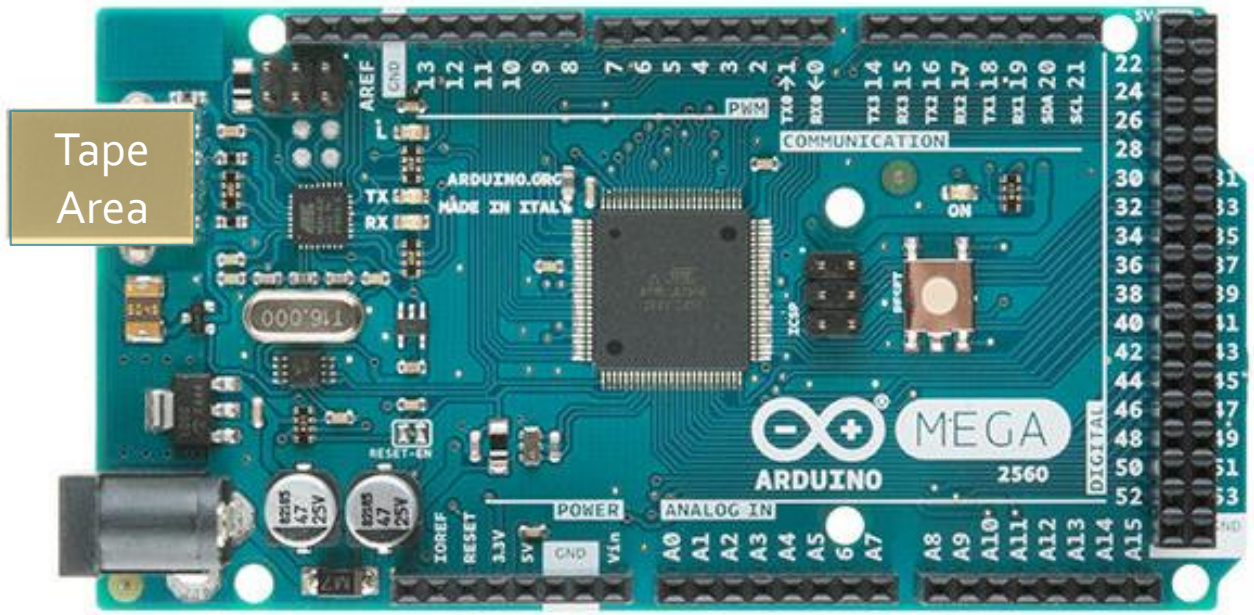
BETA GY801 & V1.1 GY91





Arduino Board Preparation

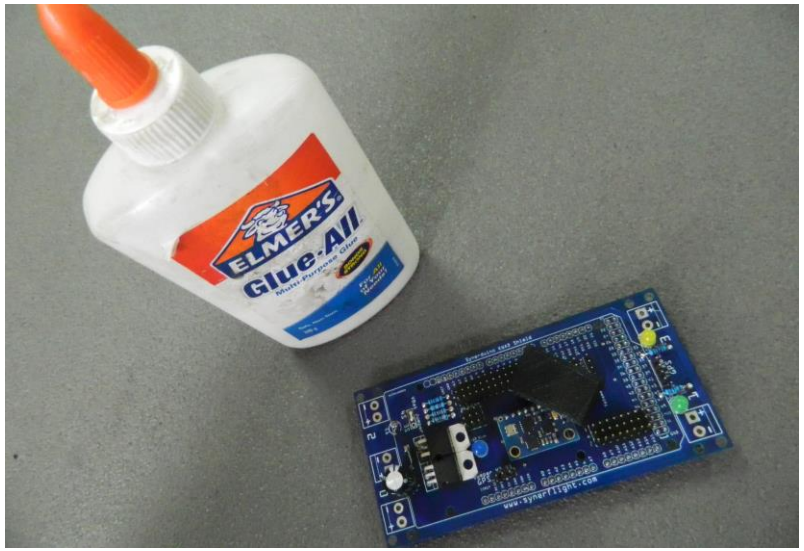
Ensure insulation from the Arduino board add tape on these areas

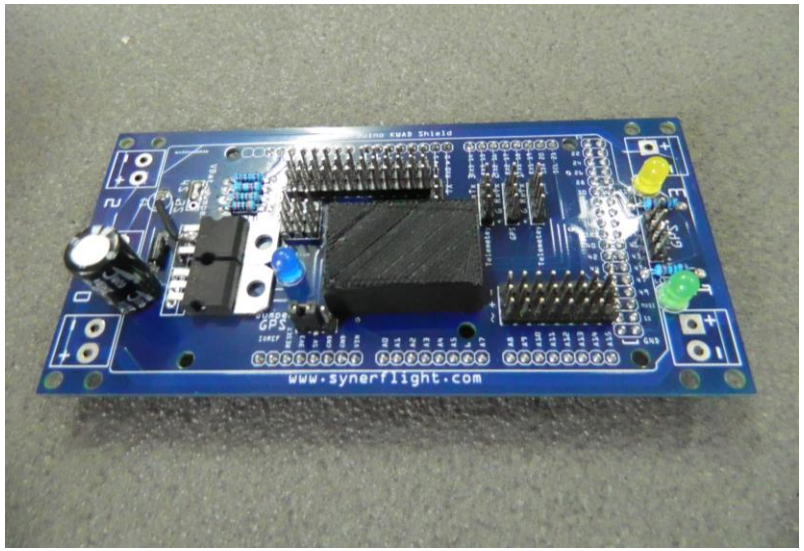


2560 MEGA

SMD UNO 328

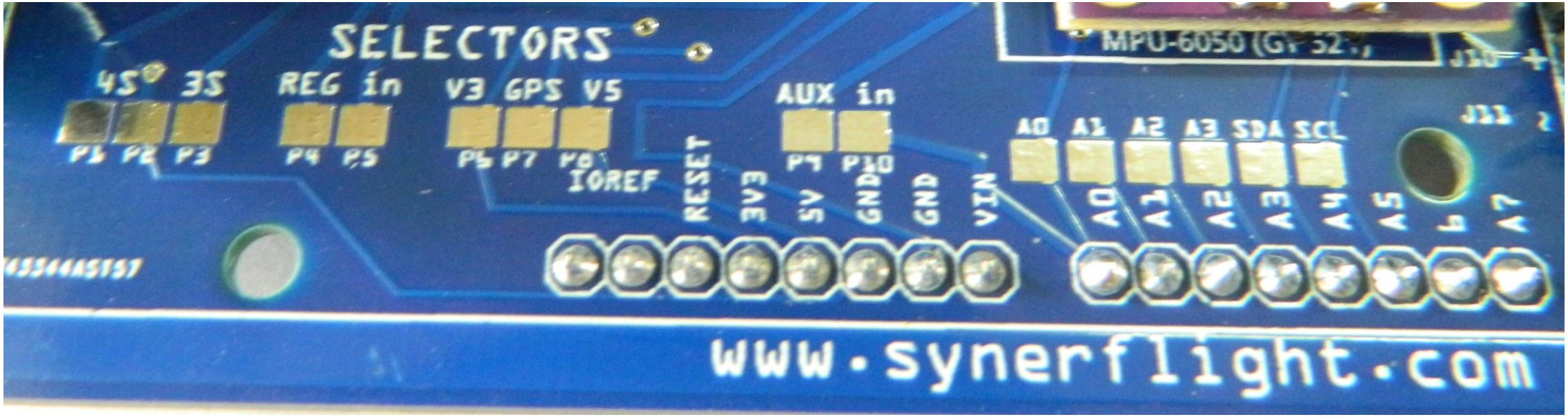
Use PVA to glue the cover to the IMU sensors seal the rim of the cover all the way

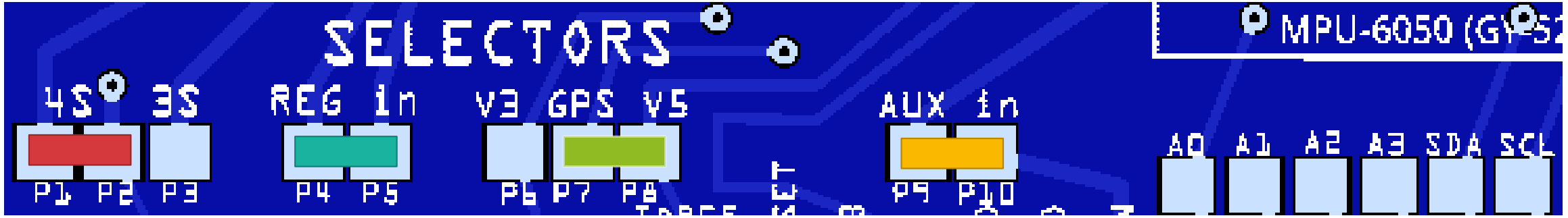




Synerduino Kwad Shield V1.1 (GY91)

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 0 pin Auxin / Battery monitor

Analog 0-3 & I2C external sensors

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

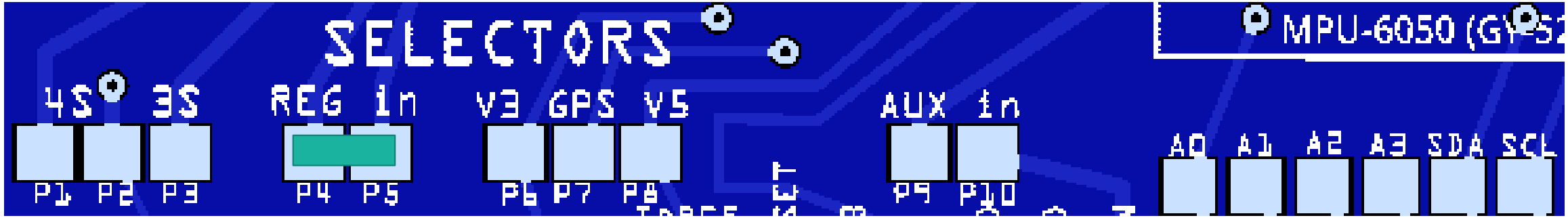
Reg In – short the Pads for using regulator to power and build in power distributor the synerduino and Arduino board

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

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

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

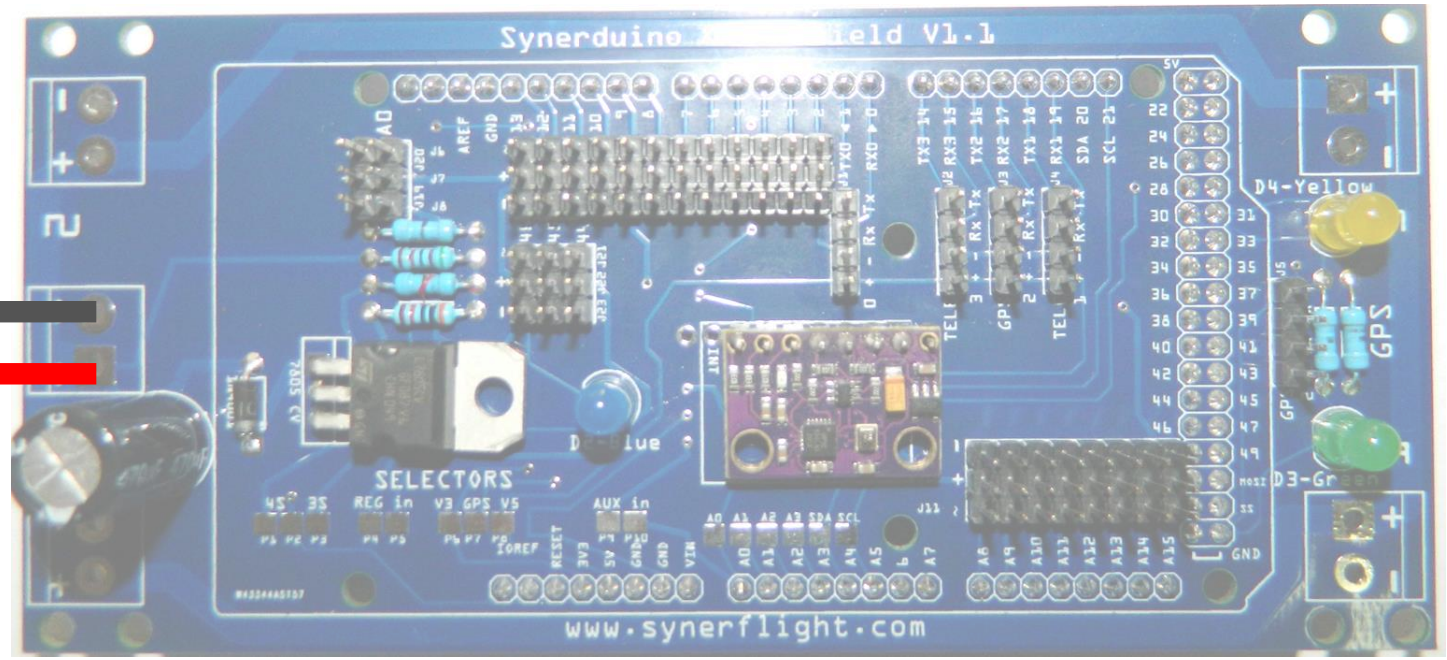
Reg in only - Ao External ADC sensor , ESC BEC or OPTO applied to the 5V PWM pins



2s to 4s Lipo -

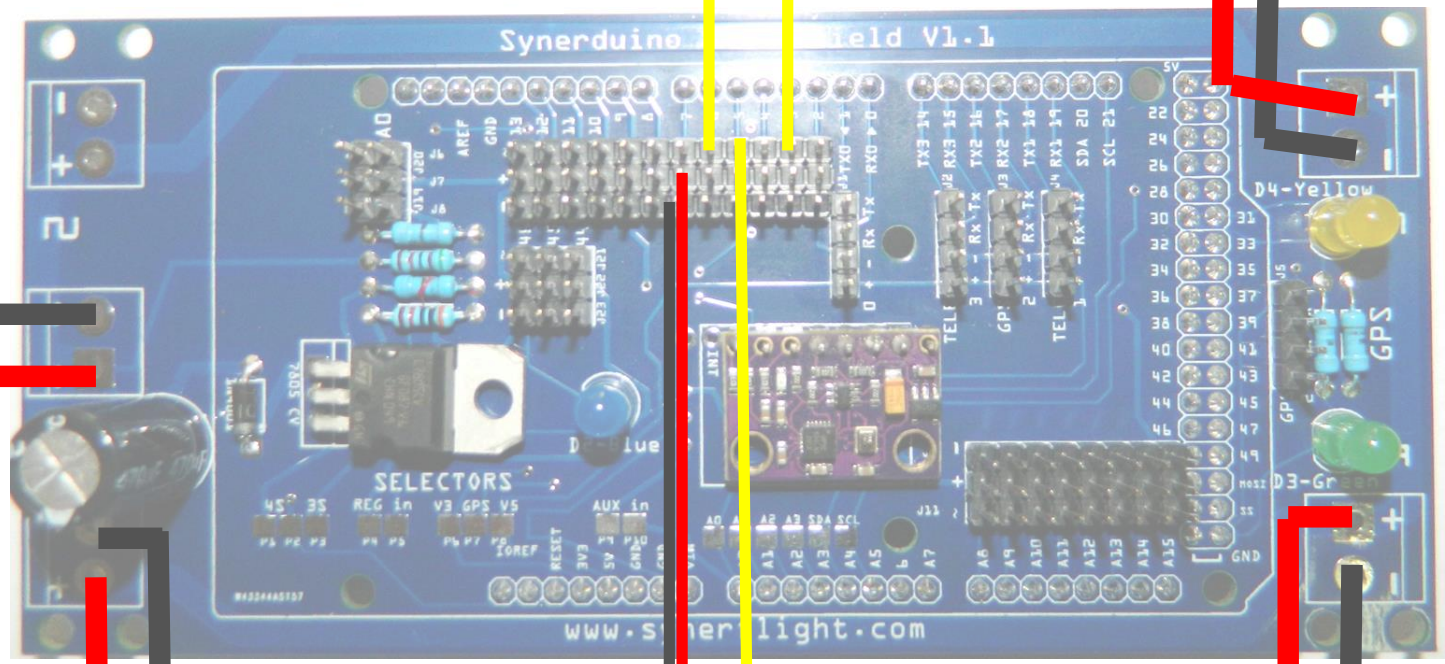
Build in Power distribution

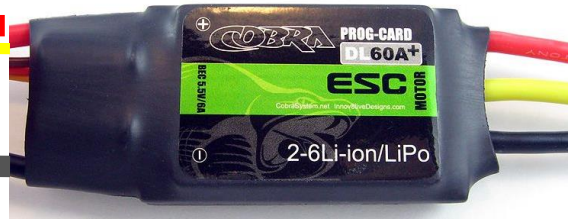
\+



Recommended setup for beginner

Recommended setup for beginner



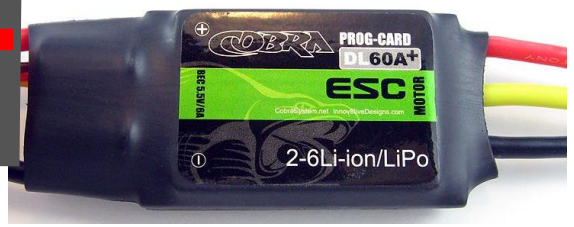


ESCs D3

2s to 4s Lipo

Build in Power distribution

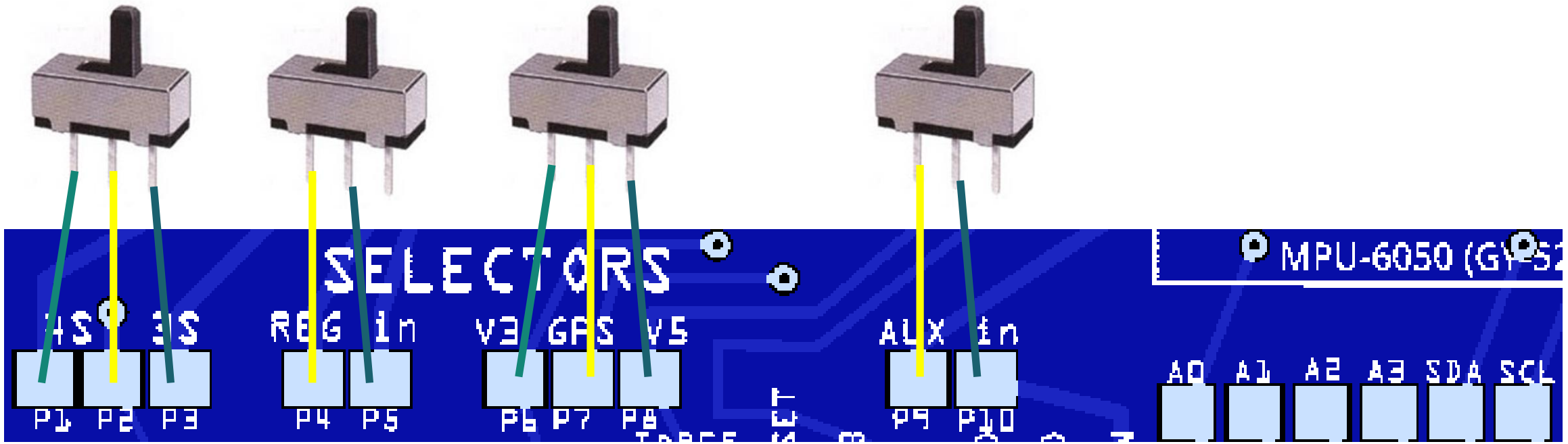
ESCs D5



Ubec Provide extra current to the servo

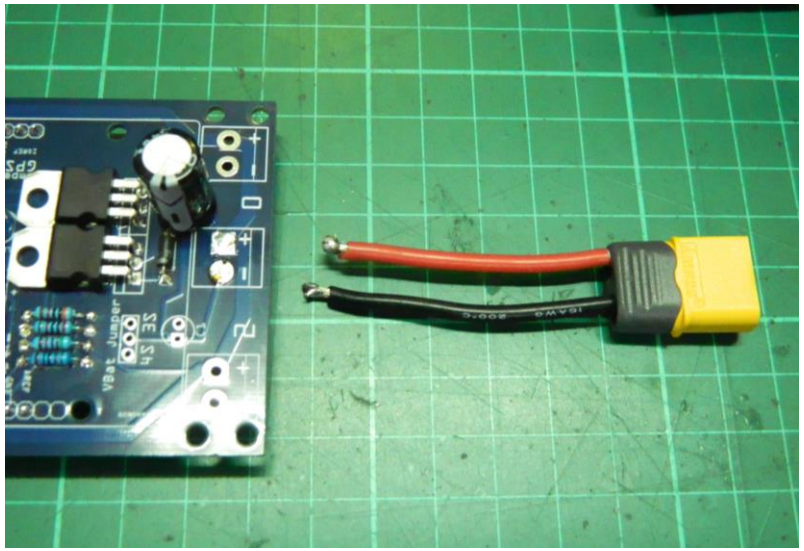
See Page 20 RC connection

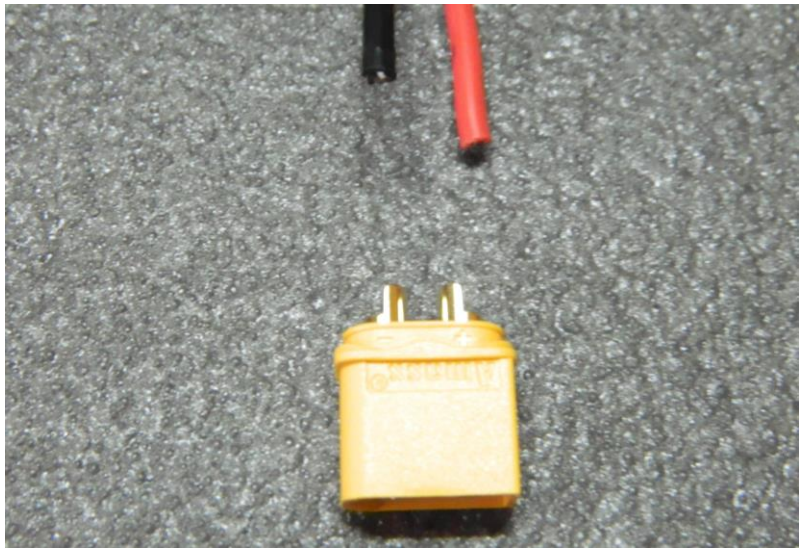
For those who thinks they need to change setup from time to time



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

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





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

Differential Thrust Wing

Arduino Mega Diff Thrust Plane

Elevator	Rudder	Aileron	Throttle
D6	D2	D7	D3 Left motor D5 Right motor

Arduino Uno Diff Thrust Plane

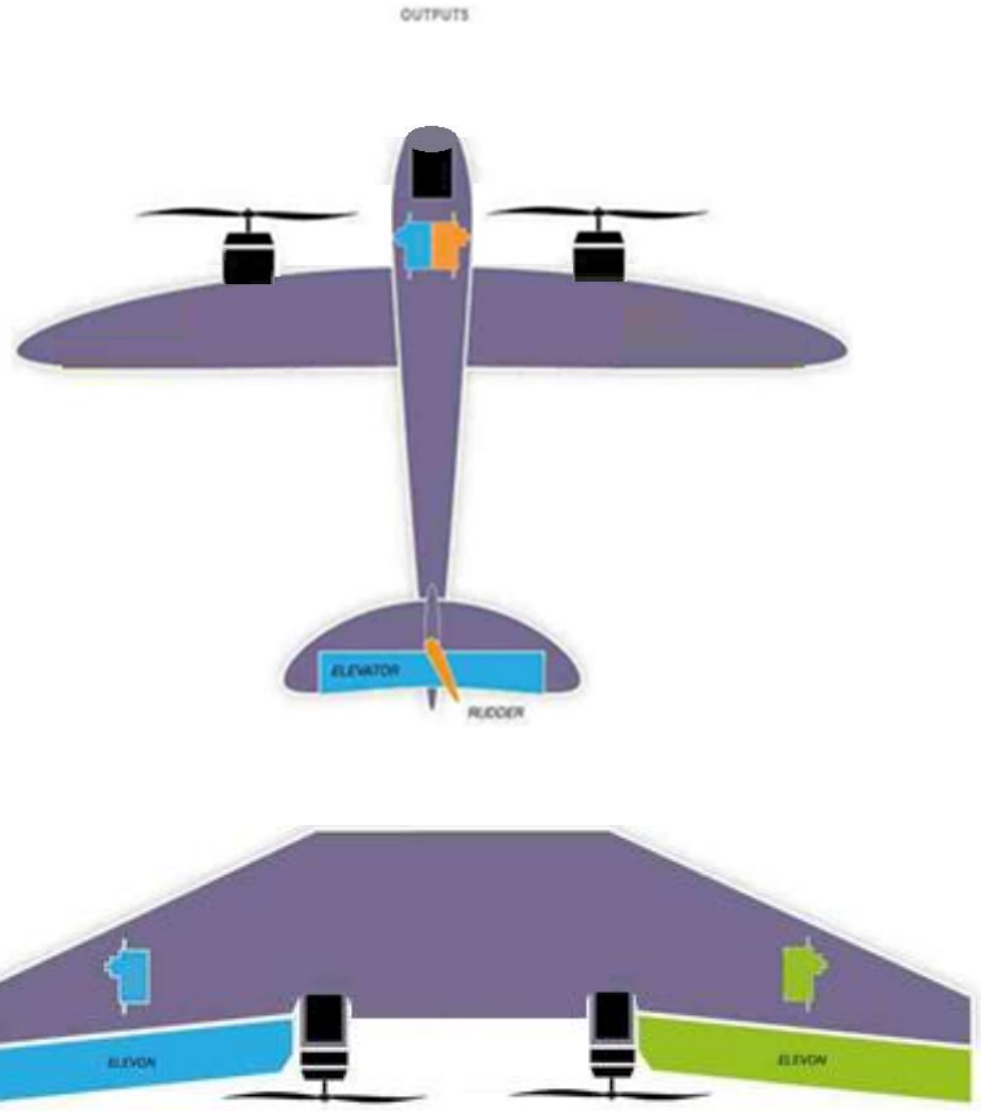
Elevator	Rudder	Aileron	Throttle
D11	D3	D12	D9 Left motor D10 Right motor

Arduino Mega Wing

Elevon	Elevon	Rudder	Throttle
D6 L	D2 R	D7	D3 Left motor D5 Right motor

Arduino Uno Wing

Elevon	Elevon	Rudder	Throttle
D11 L	D3 R	D12	D9 Left motor D10 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

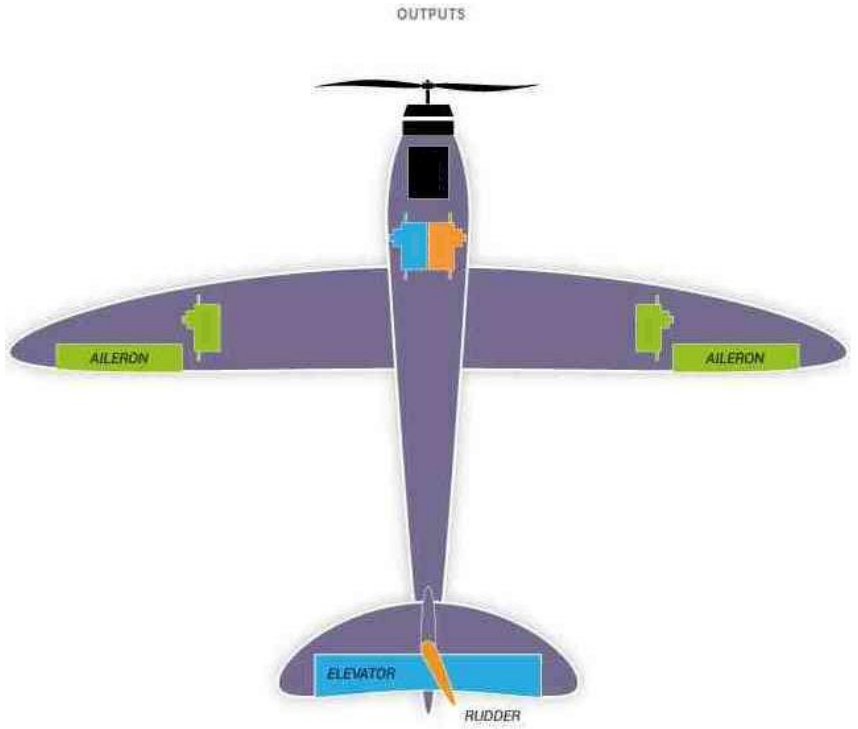
Some models of transmitter may need RC output reversing to operate correctly

Arduino Mega Airplane

Elevator	Rudder	Aileron	Throttle
D6	D2	D7	D3 D5 motor

Arduino Uno Airplane

Elevator	Rudder	Aileron	Throttle
D11	D3	D12	D9 D10 motor



Arduino Mega Wing

Elevon	Elevon	Aileron	Throttle
D6 L	D2 R	D7	D3 D5 motor

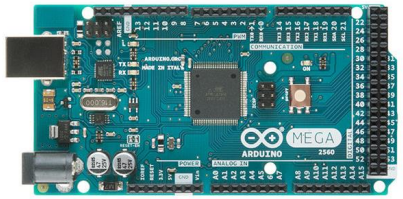
Arduino Uno Wing

Elevon	Elevon	Rudder	Throttle
D11 L	D3 R	D12	D9 D10 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

PWM INPUT Assignment

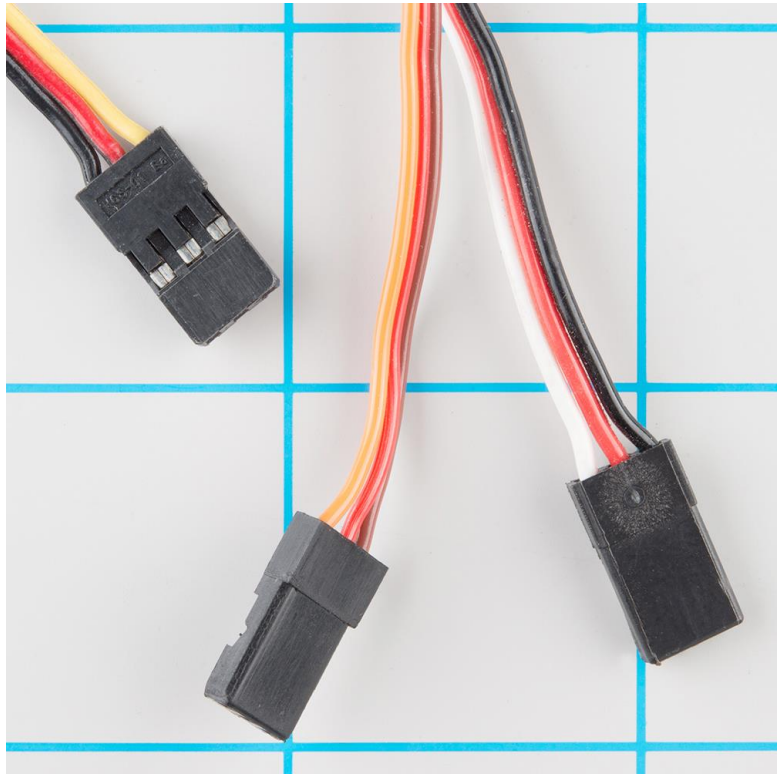
Pls Check the output pin from your Radio Rx manual

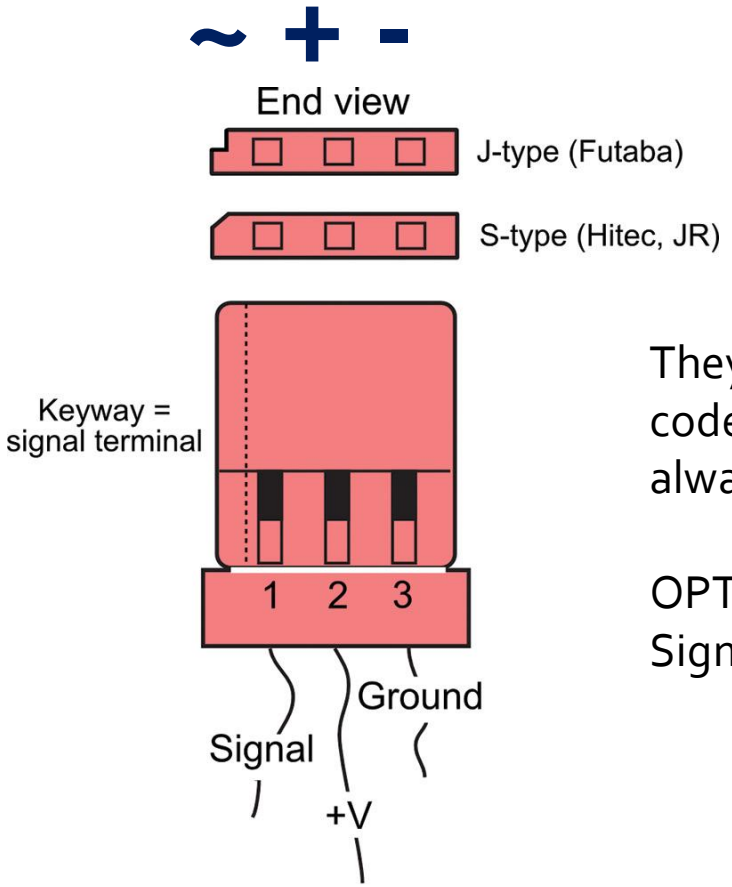


RX > Arduino / PWM in	Futaba Format	JR Format	Walkera Format	UNO 328 Input	Mega 2560 Input
Throttle	Ch3	Ch1	Ch3	D2	A8
Aileron	Ch4	Ch4	Ch4	D4	A9
Elevator	Ch2	Ch3	Ch1	D5	A10
Rudder	Ch1	Ch2	Ch2	D6	A11
Aux1	Ch5	Ch5	Ch5	D7	A12
Aux2	Ch6	Ch6	Ch6	D8	A13
Aux3	Ch7	Ch7	Ch7	N/A	A14
Aux4	Ch8	Ch8	Ch8	N/A	A15

Because the plane is classed as 3ch setup RET the Aileron and Rudder pins are intentional swap turning your mode 2 right stick as Elevator Rudder control

SERVO HEADER





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

OPTO Wires may only have Signal and Negative Wires o

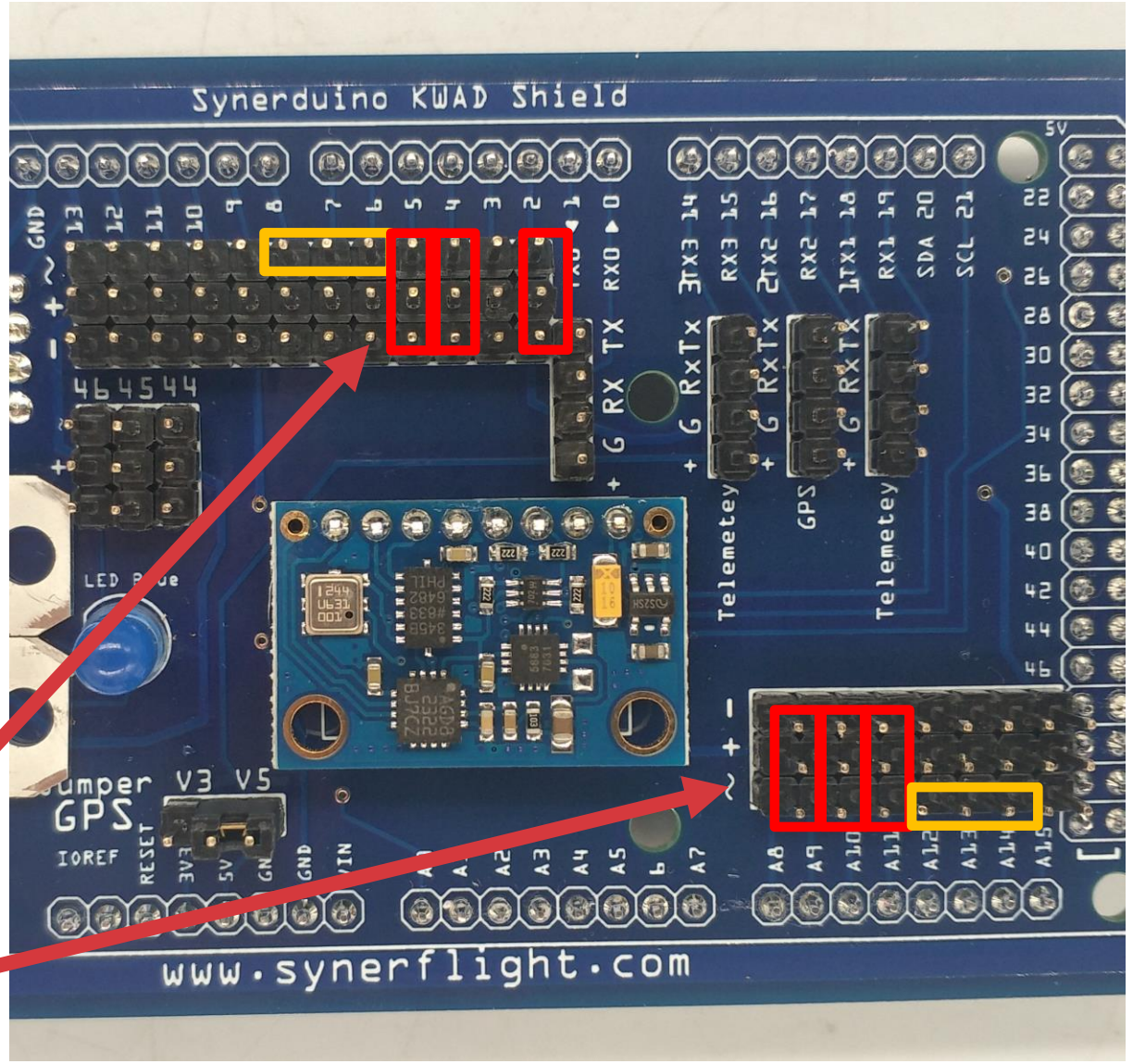
PWM RECEIVER

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

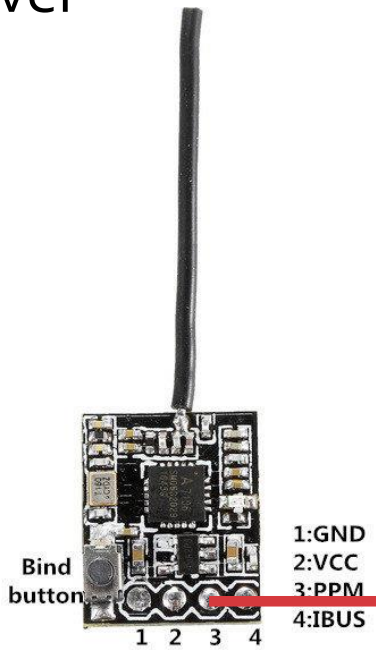
MOSTLY RUDDER AUX1 AND AUX2

UNO PWM IN

MEGA PWM IN

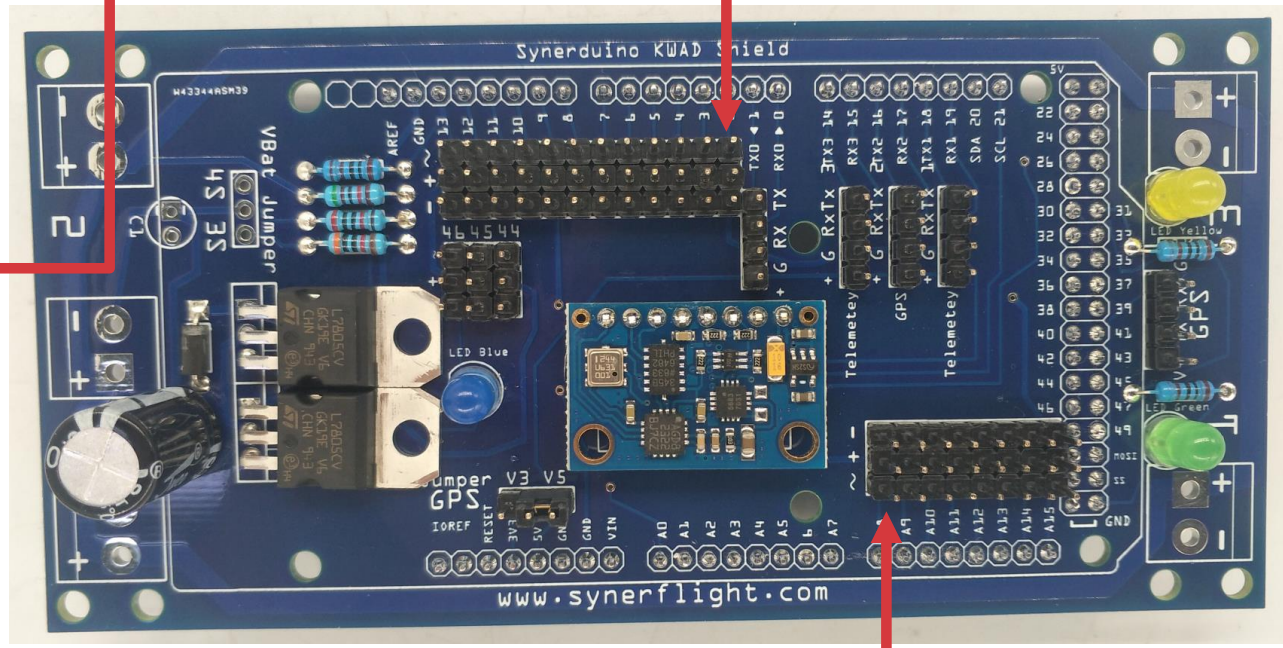


PPM Receiver



Pin D2 for Uno

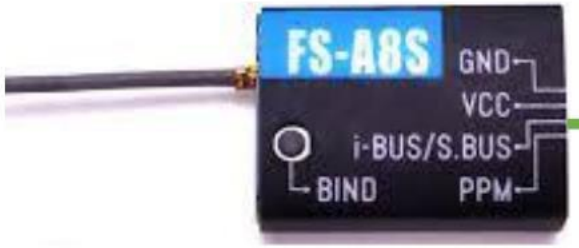
Pin A8 for Mega



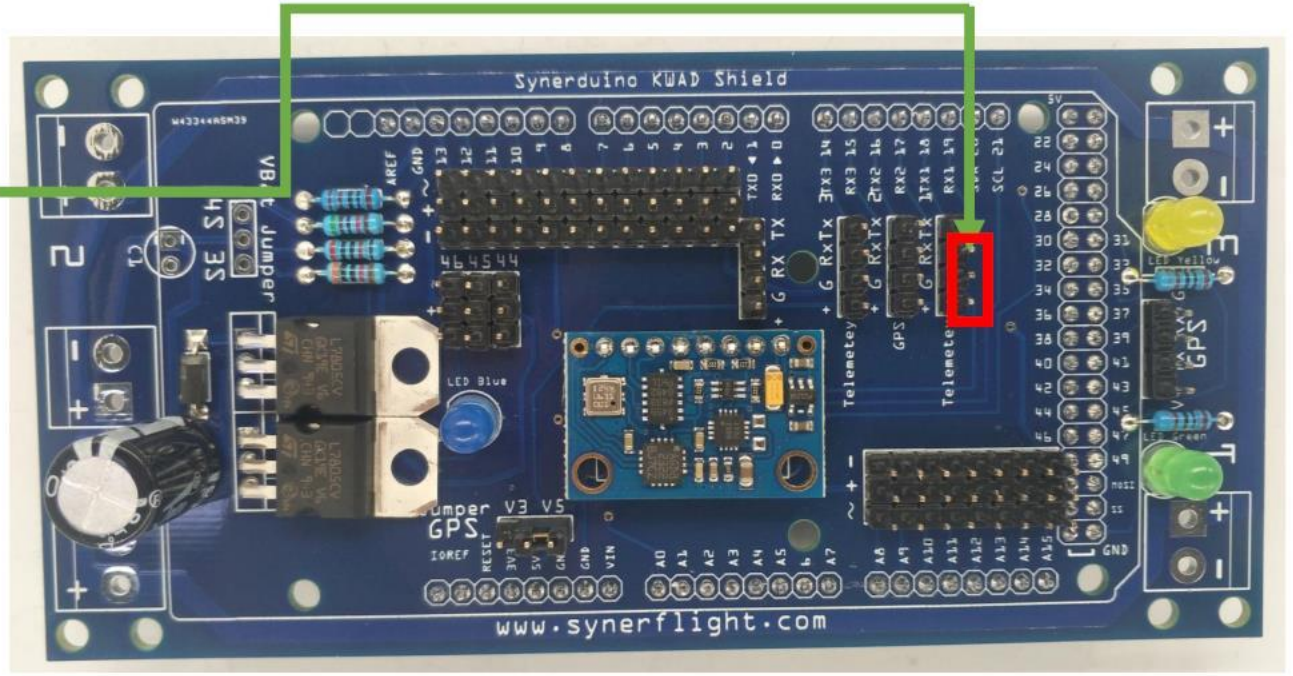
PPM and SBUS reassignment for RET setup see on

SBUS Receiver

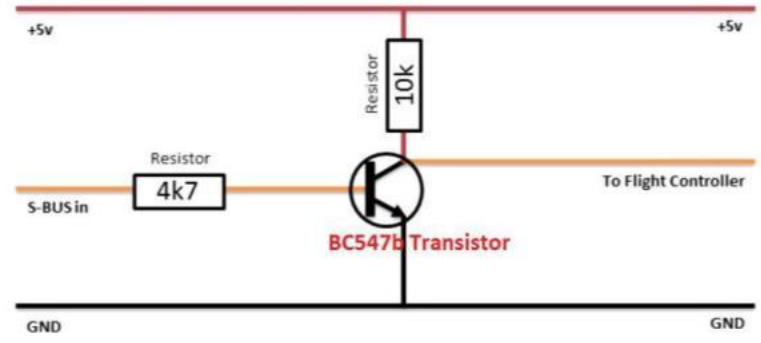
Most modern Receivers now comes with Serial Protocol as they are faster than the old PWM or PPM standard and its now the Modern defacto for Receiver to Flight Control Board communication

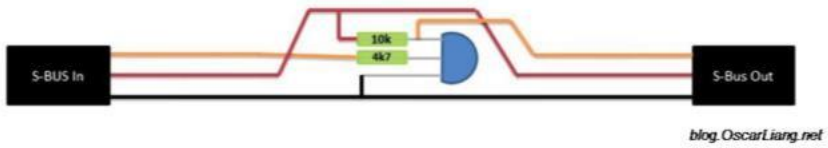


RX 1 Telemetry



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





SBUS Inverter

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

You can tell if the RC control is not being read

RECEIVER TYPES

CONFIG.H

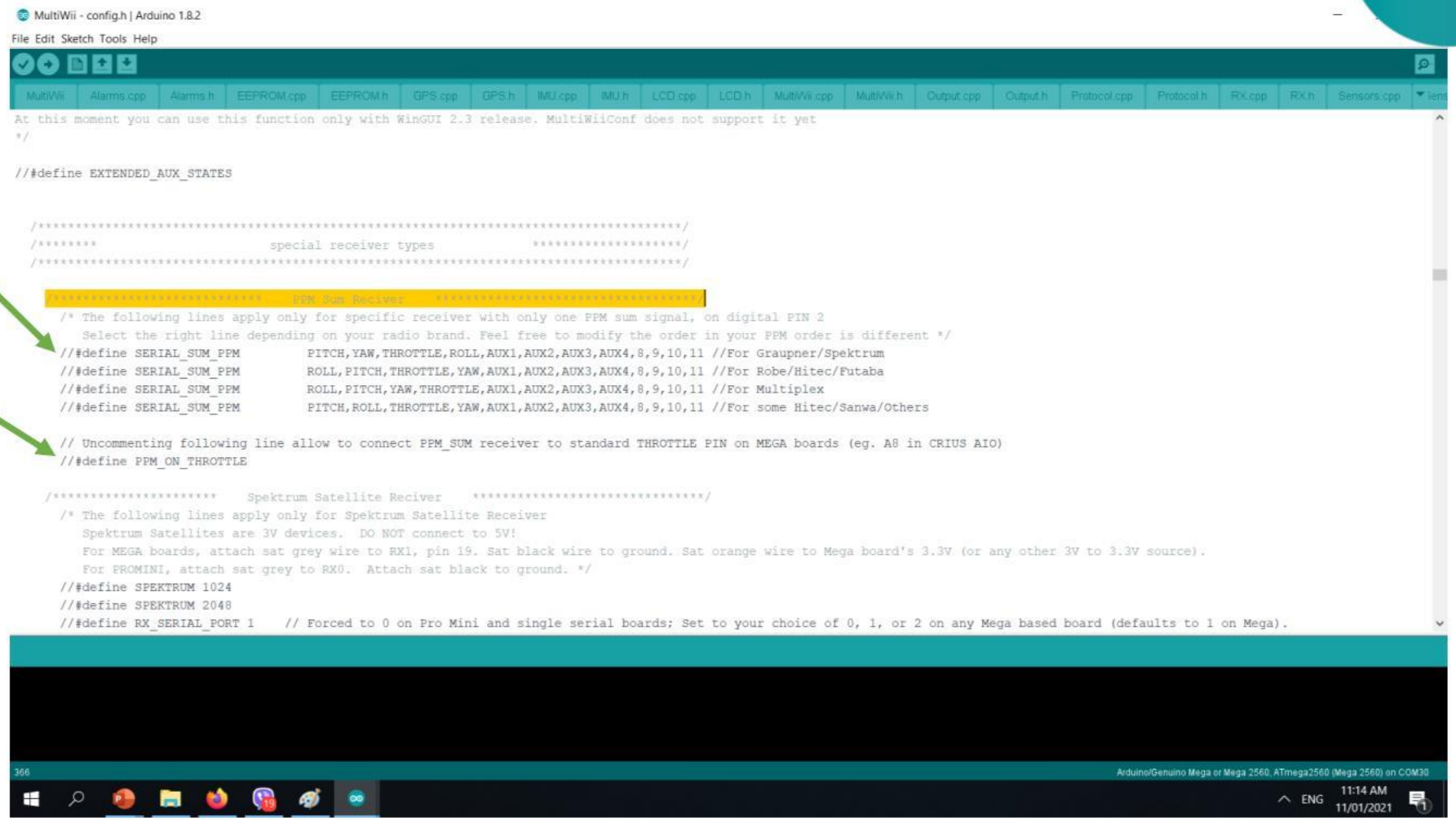
For PPM Receiver

Channel Mapping

Uncomment PPM on Throttle

Pin A8 for Mega
Pin D2 for Uno

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



You may need to rearrange Define SERIAL_SUM _PPM YAW, PITCH, THROTTLE, ROLL

RECEIVER TYPES

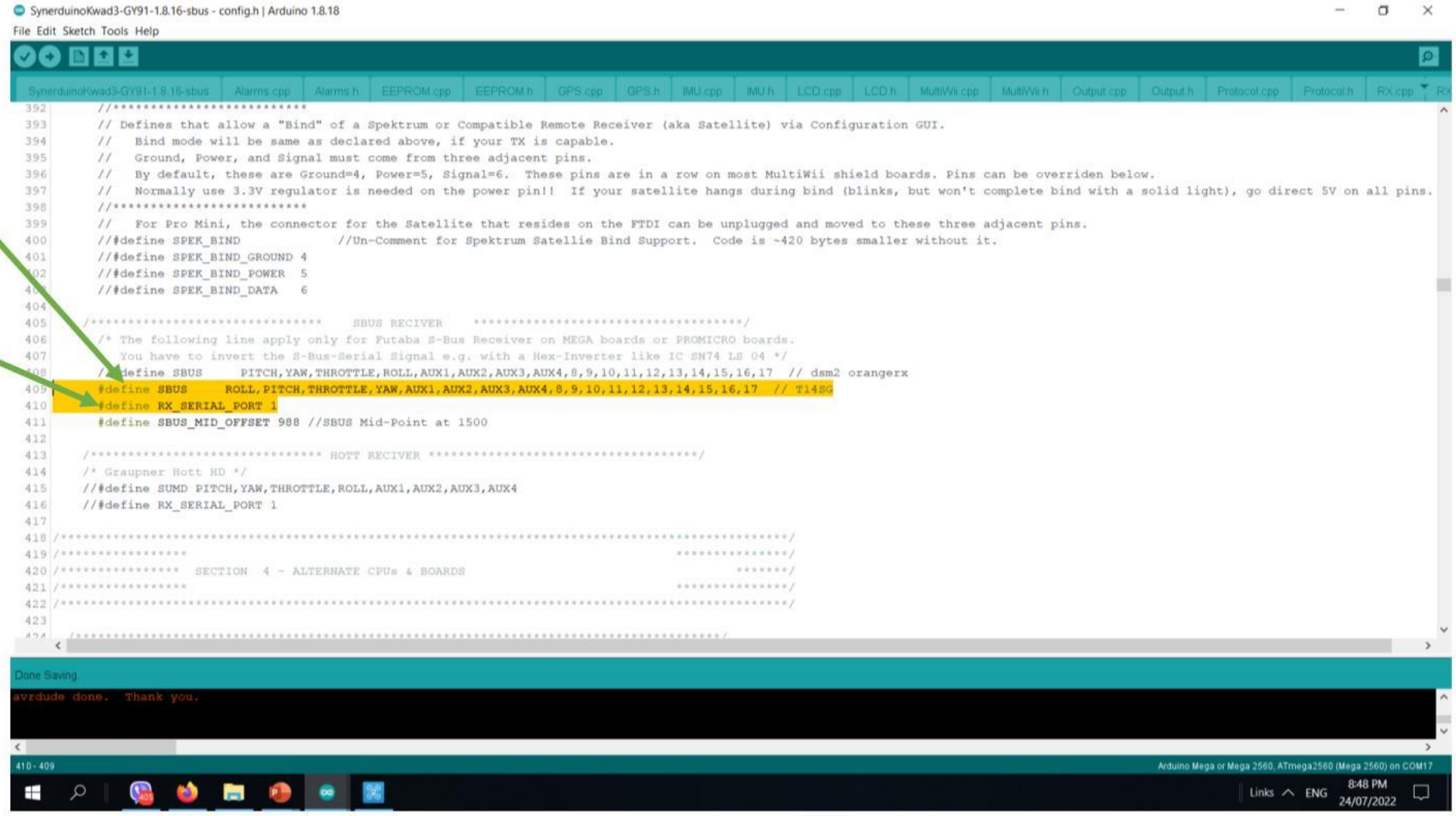
CONFIG.H

For SBUS Receiver

Channel Mapping

Uncomment SBUS on RX Serial Port 1 (Telemetry 1)

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

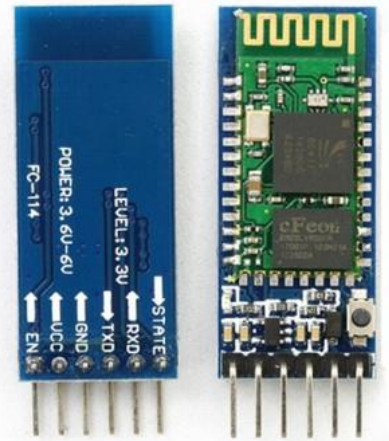


You may need to rearrange Define SERIAL_SUM _PPM YAW, PITCH, THROTTLE, ROLL

TELEMETRY RADIO

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

38400 FOR XBEE RADIO



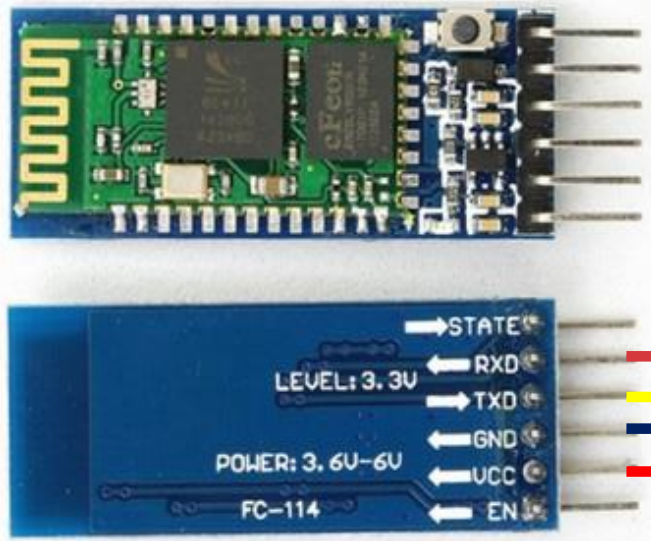
115200 FOR BLUETOOTH HC-05

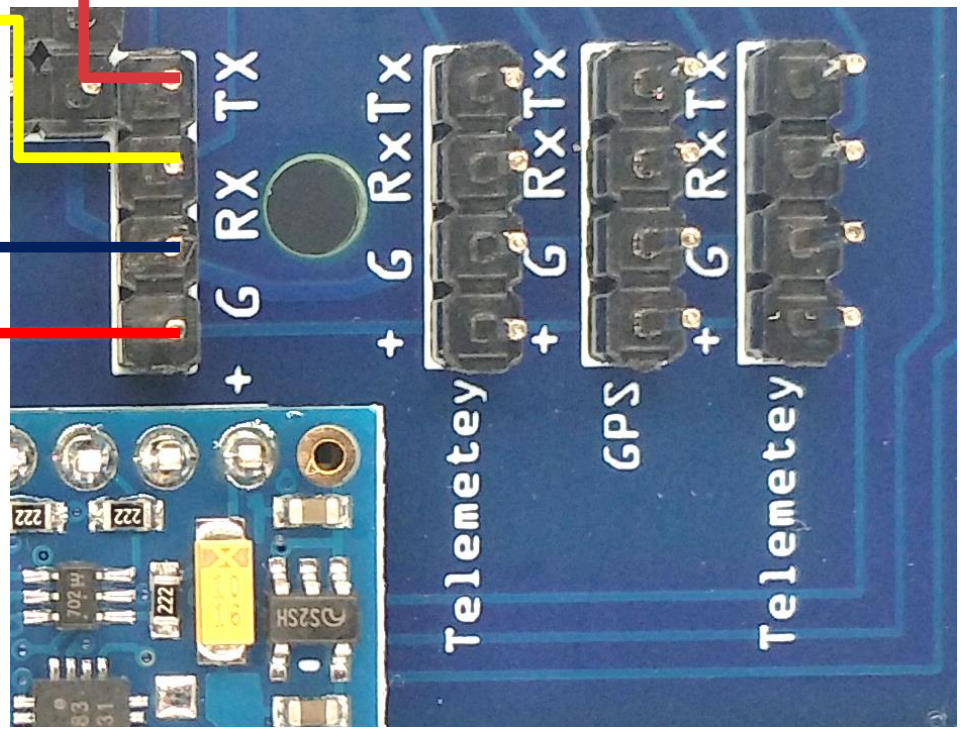
STANDARD FOR ALL DRONES TO USE SERIAL LINK AS TELEMETRY MAINLY ON YOUR TX RX SERIAL PORTS , NOTE: THE LOWER THE FREQUENCY OF THE RADIO THE LOWER THE BAUD IS NEEDED

PROTOCOL IS MSP RAW OR MAVLINK

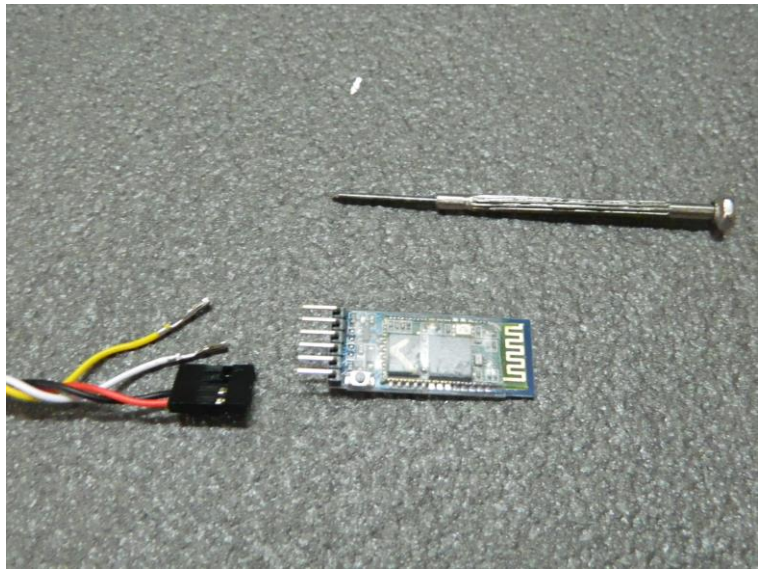
Bluetooth

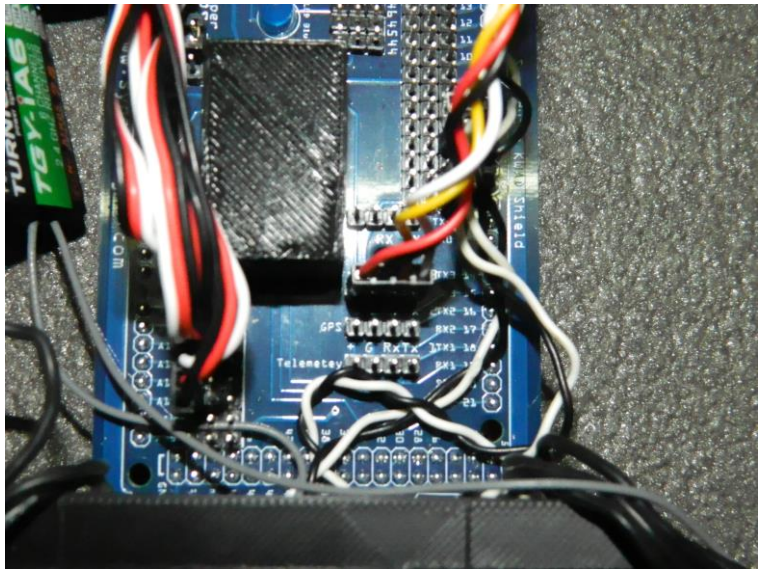
NOTE: USING 0 AT BAUD115200 APPLICABLE BOTH UNO AND MEGA BOARDS REQUIRES THE BLUETOOTH ONLY TO BE PLUG IN AFTER THE SKETCH/FIRMWARE HAS BE UPLOADED TO THE BOARD . AND PAIRING USING THE BLUETOOTH CONNECTED COM PORT IN DEVICE MANAGER





Bluetooth





BLUETOOTH PLUG INTO SERIAL 1 OR SERIAL 3

115200 FOR BLUETOOTH HC-05

ATTENTION:

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

VCC >> +

GND >> G

TX >> RX

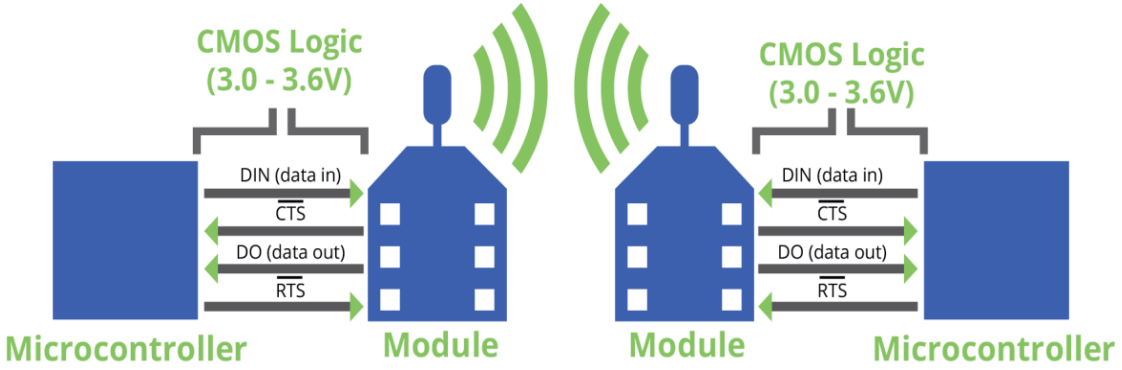
RX >>TX

SEE TO IT THE WIRES COLOR CODE MATCHES THE MARKINGS

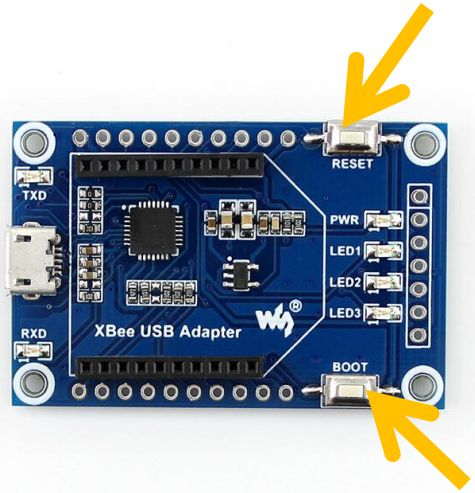
IMPROPER INSTALLATION MAY CAUSE DAMAGE TO THE ARDUINO BOARD AND SHIELD DUE TO REVERSE POLARITY

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

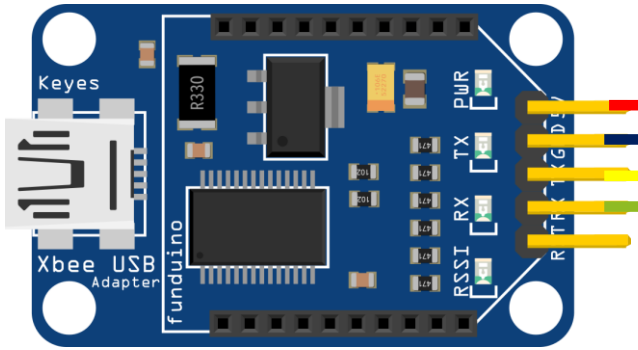
XBEE RADIO



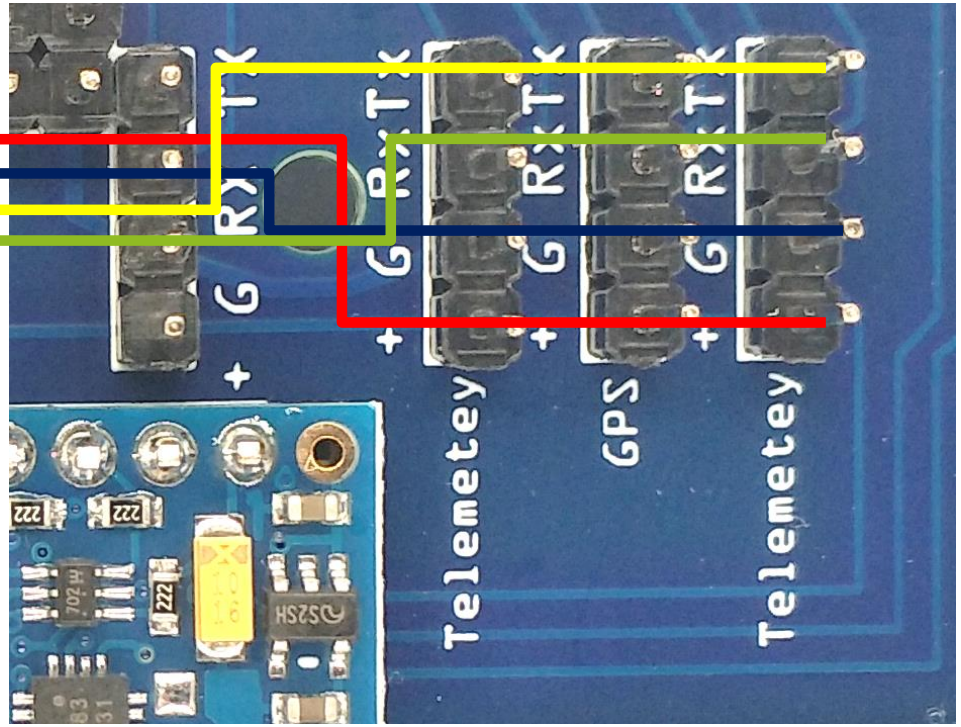
38400 FOR XBEE RADIO



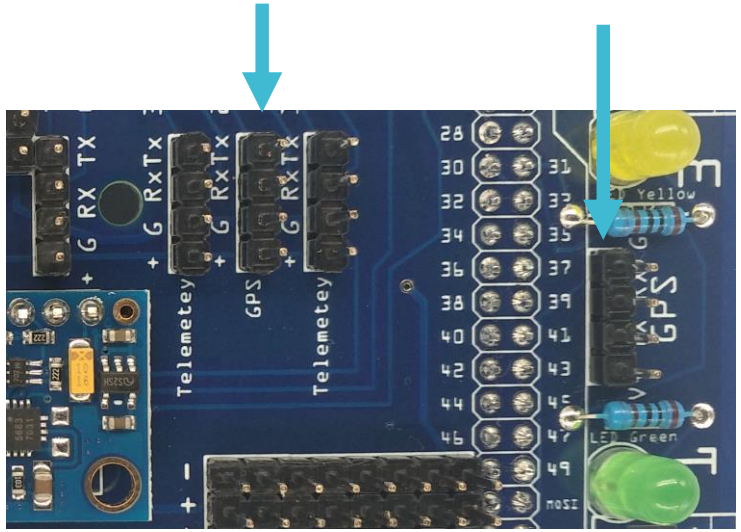
GET THE USB MODULE WITH BOOT AND RESET BUTTON AS YOU MAY NEED TO RESET THE XBEE WHEN UPDATING FIRMWARE



VCC >> +
GND >> G
TX >> TX
RX >>RX



GPS



U BLOX NEO 6

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

ATTENTION:

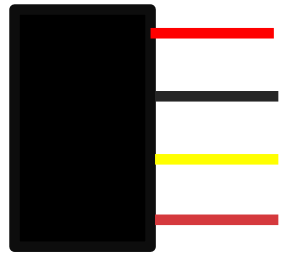
YOU MAY NEED TO REARRANGE THE HEADERS TO CONNECT THE GPS MODULE TO THE SHIELD BOARD ACCORDINGLY

VCC >> +

GND >> G

TX >> RX

RX >>TX



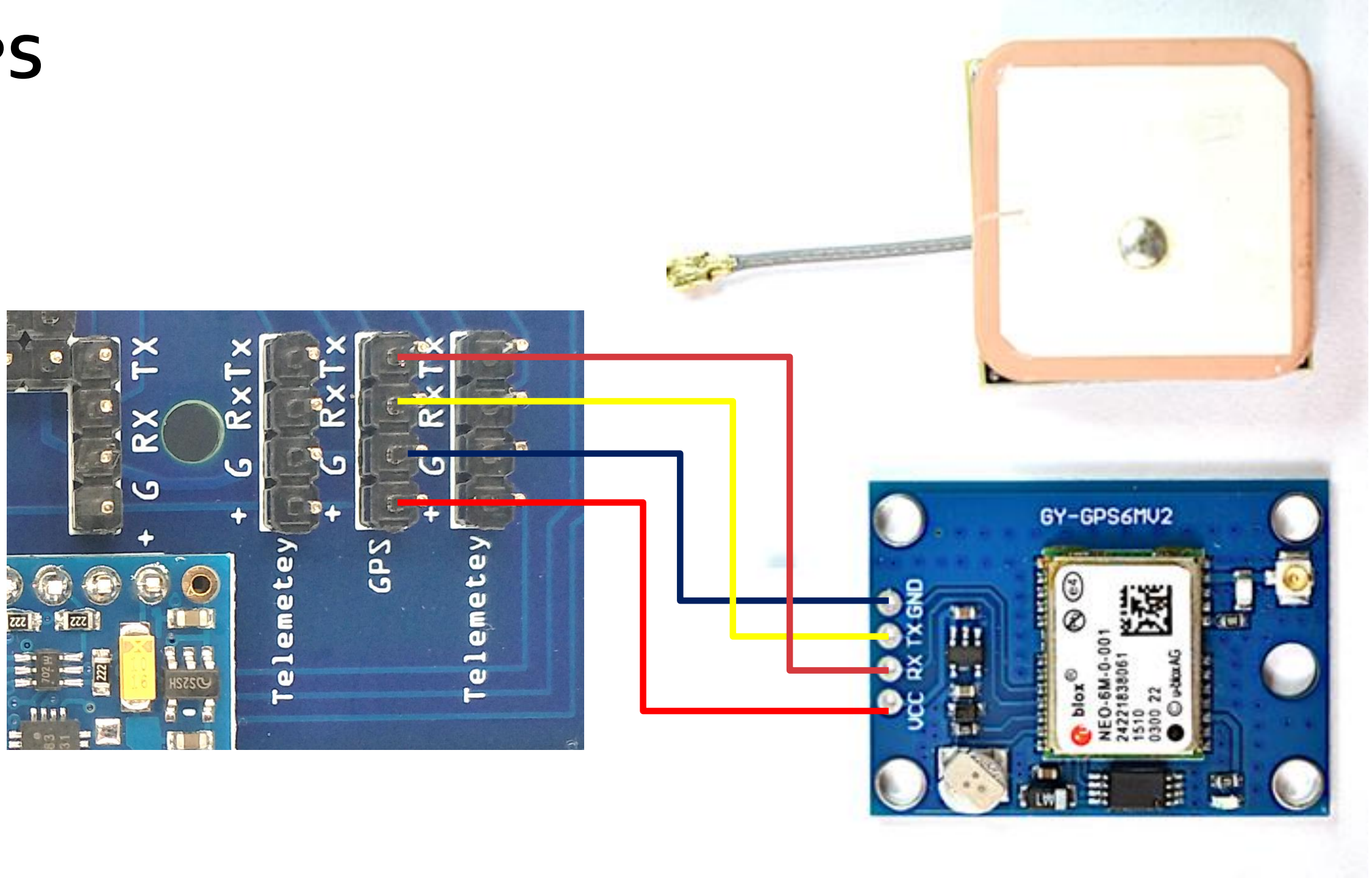
SEE TO IT THE WIRES COLOR CODE MATCHES THE MARKINGS

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

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

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

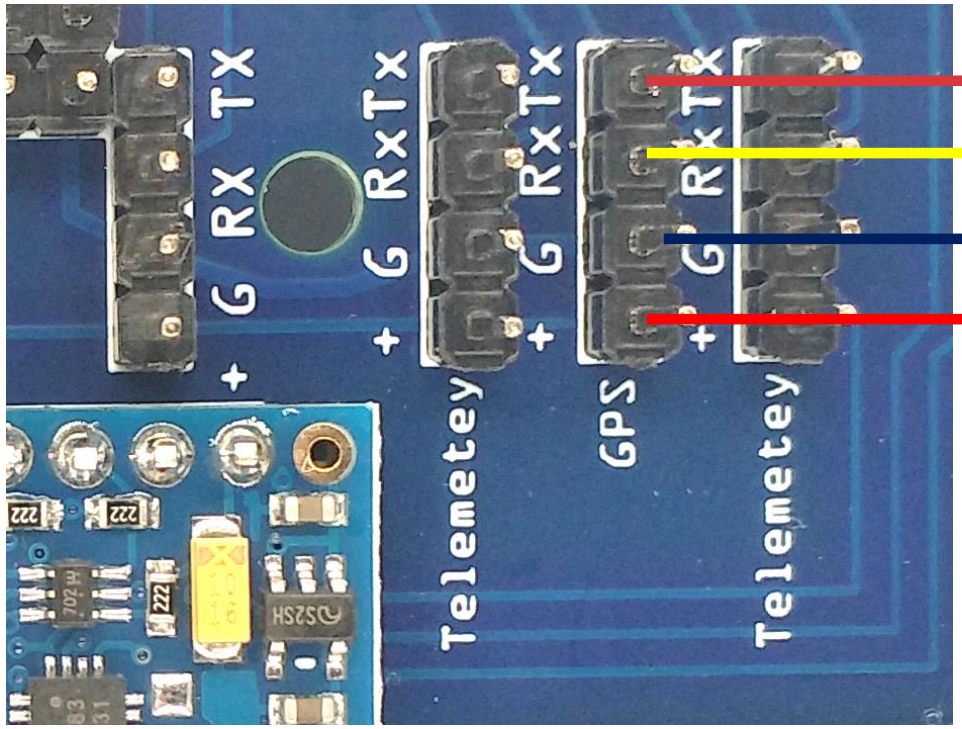
GPS


+ G RX TX
+ G RxTx
+ G RxTx
+ G RxTx
Telemetey
GPS
Telemetey
GY-GPS6MV2
VCC RX TX GND
NEO-6M-0-001
u-blox AG

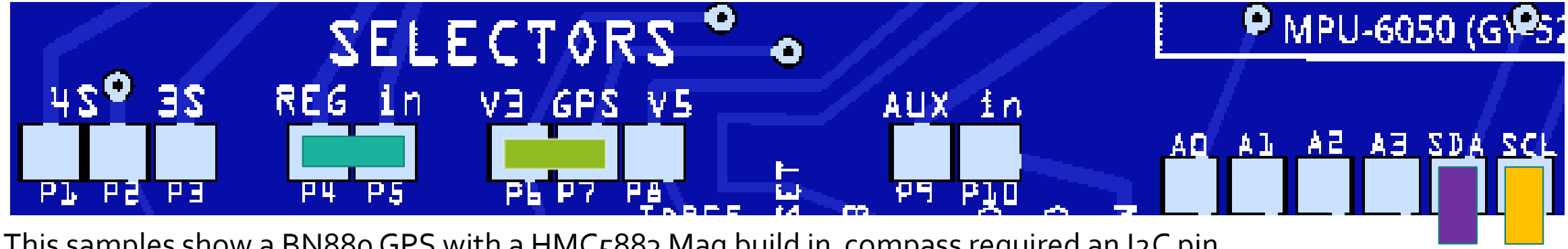
GPS

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

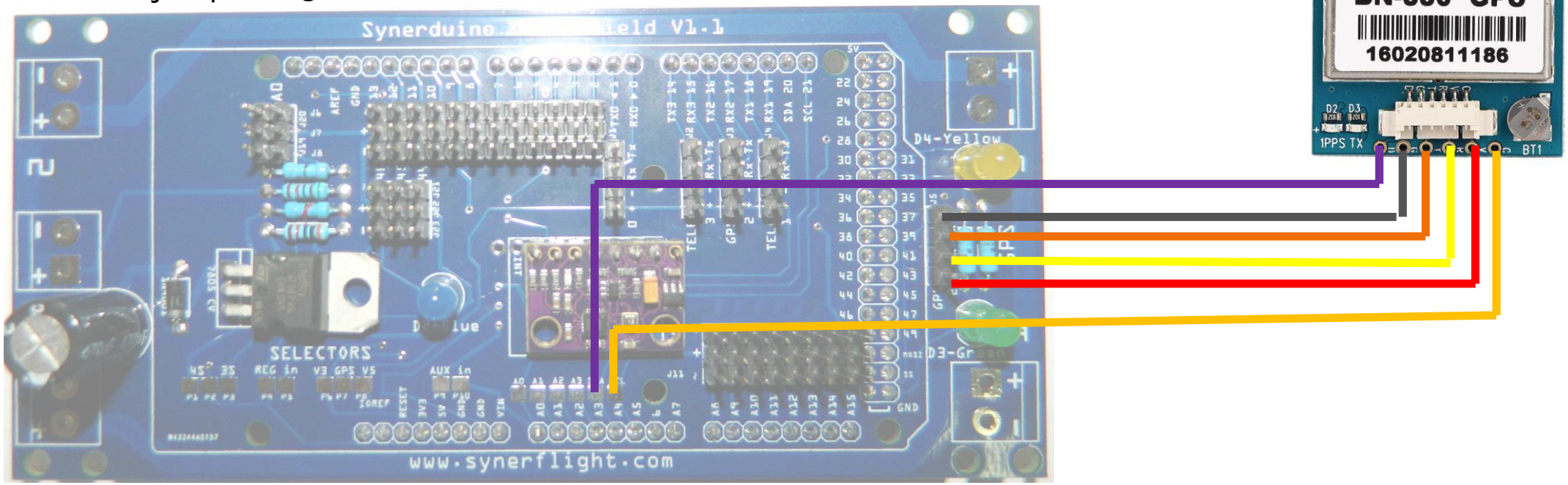
PIN	PIN Name	I/O	Description
1	GND	G	Ground
2	TX	O	Serial Data Output.
3	RX	I	Serial Data Input.
4	VCC	I	DC 3.0V - 5.5V supply input,Typical: 5.0V



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



Application Needed

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

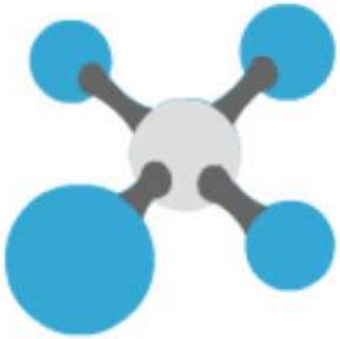
ARDUINO IDE

http://synerflight.com/flywiigui/

FLYWII GUI & Arduino Drone Multiwii firmware

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

UBLOX Configuration Platform



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

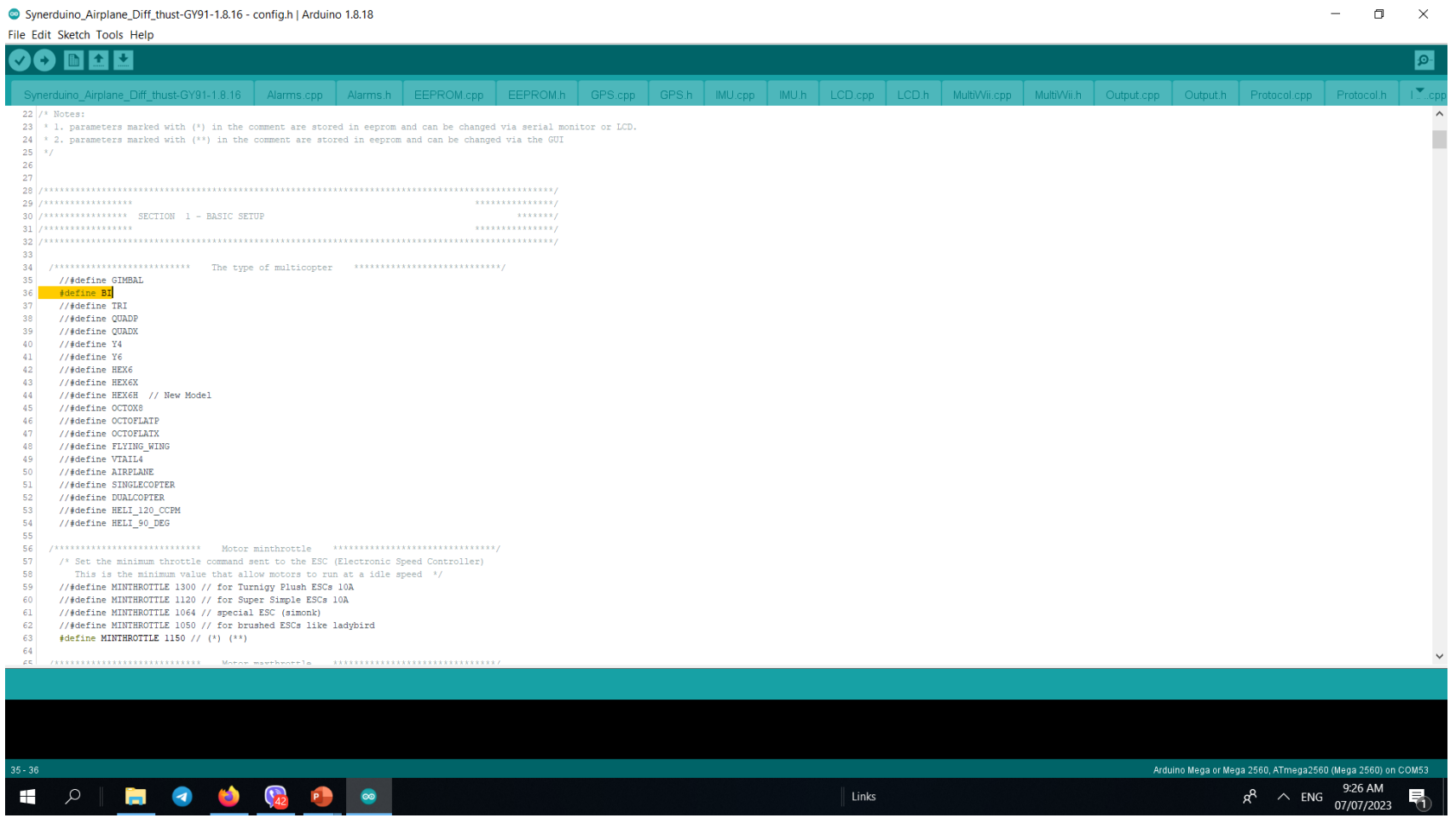
XCTU Configuration Platform

Config.h

Uncomment Define Bi

This puts the vehicle into Bi diff thrust mode mode

Only applicable on special Airplane ino skech

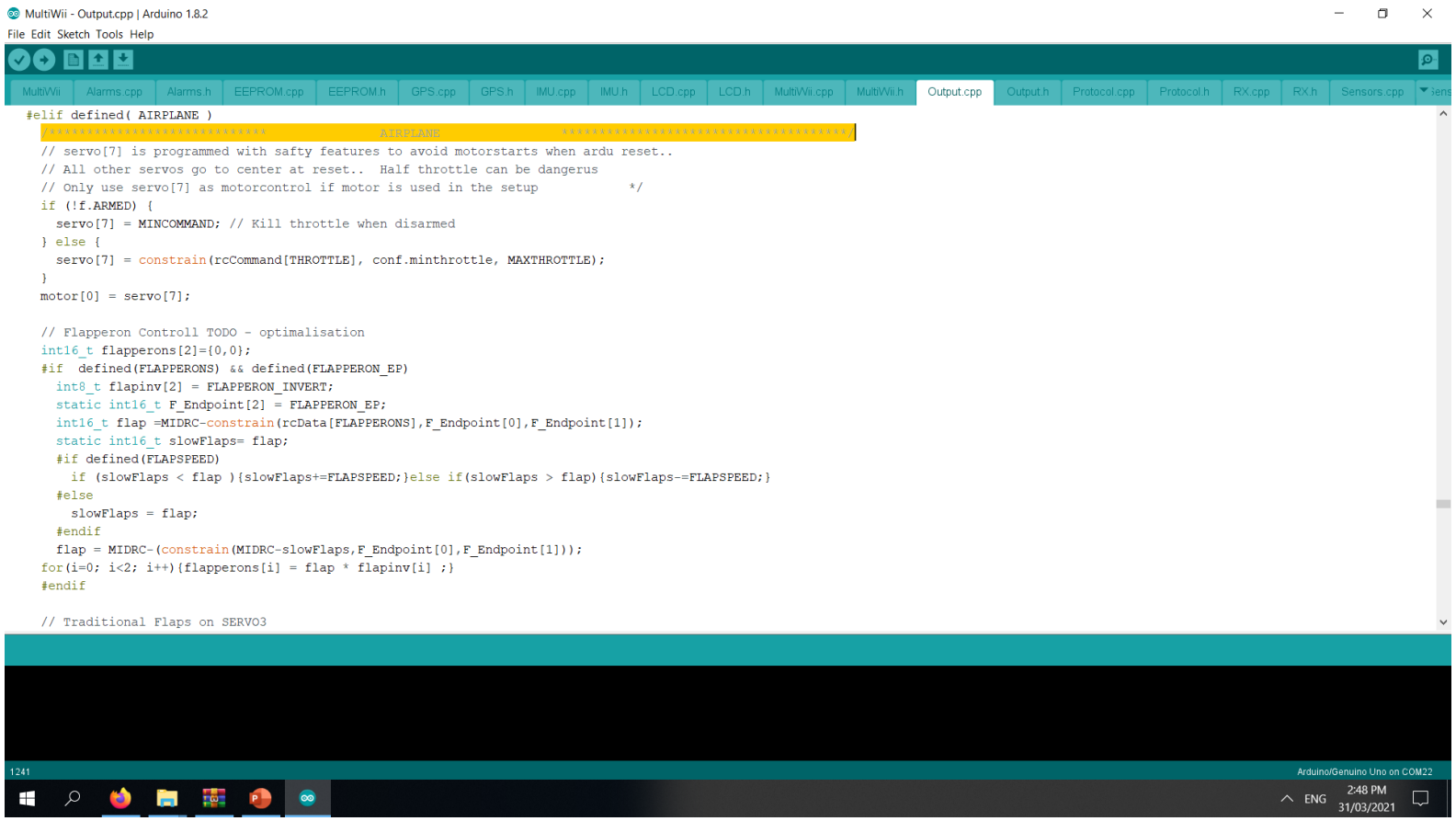


Output.cpp

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

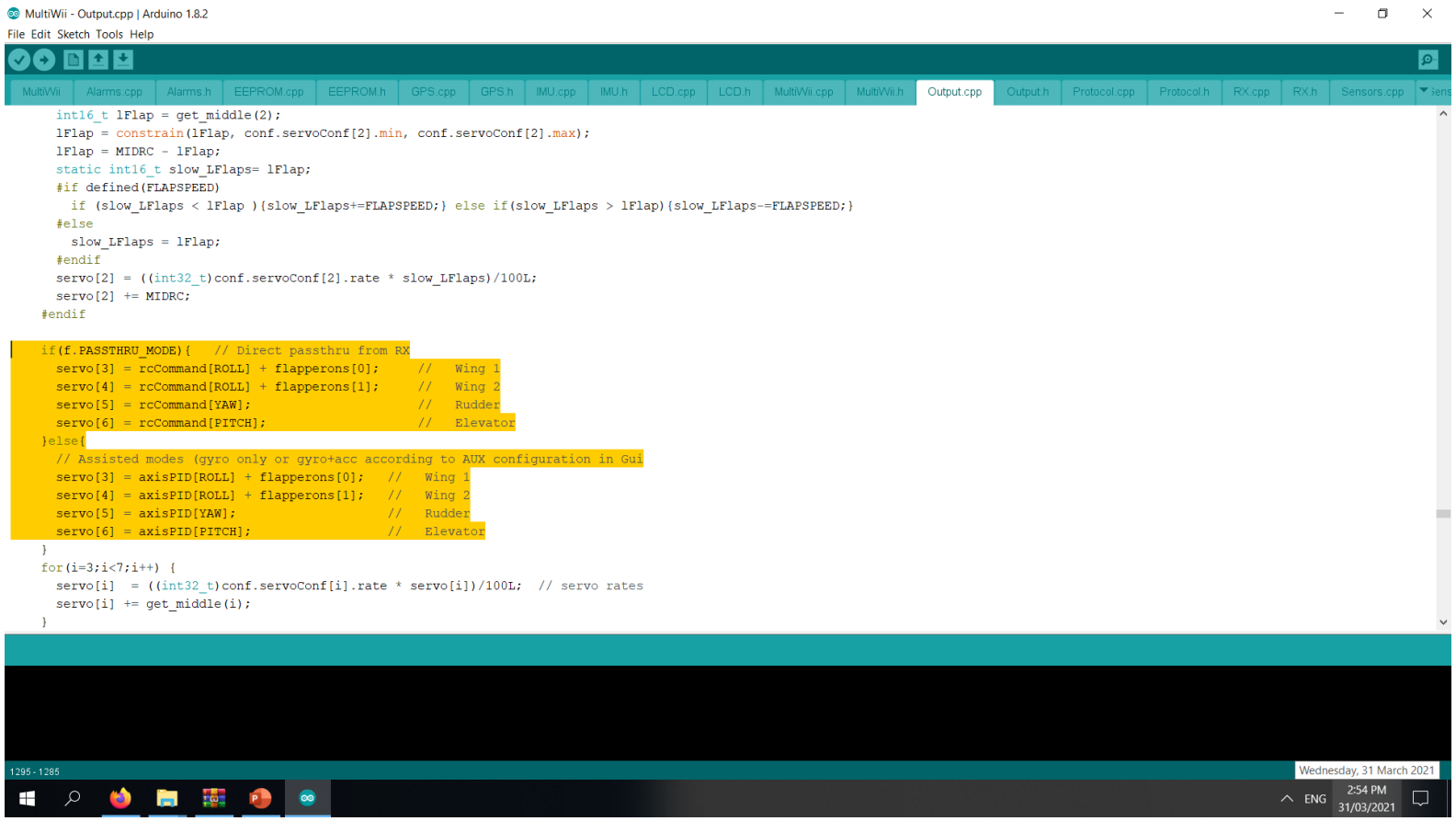


Output.cpp

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

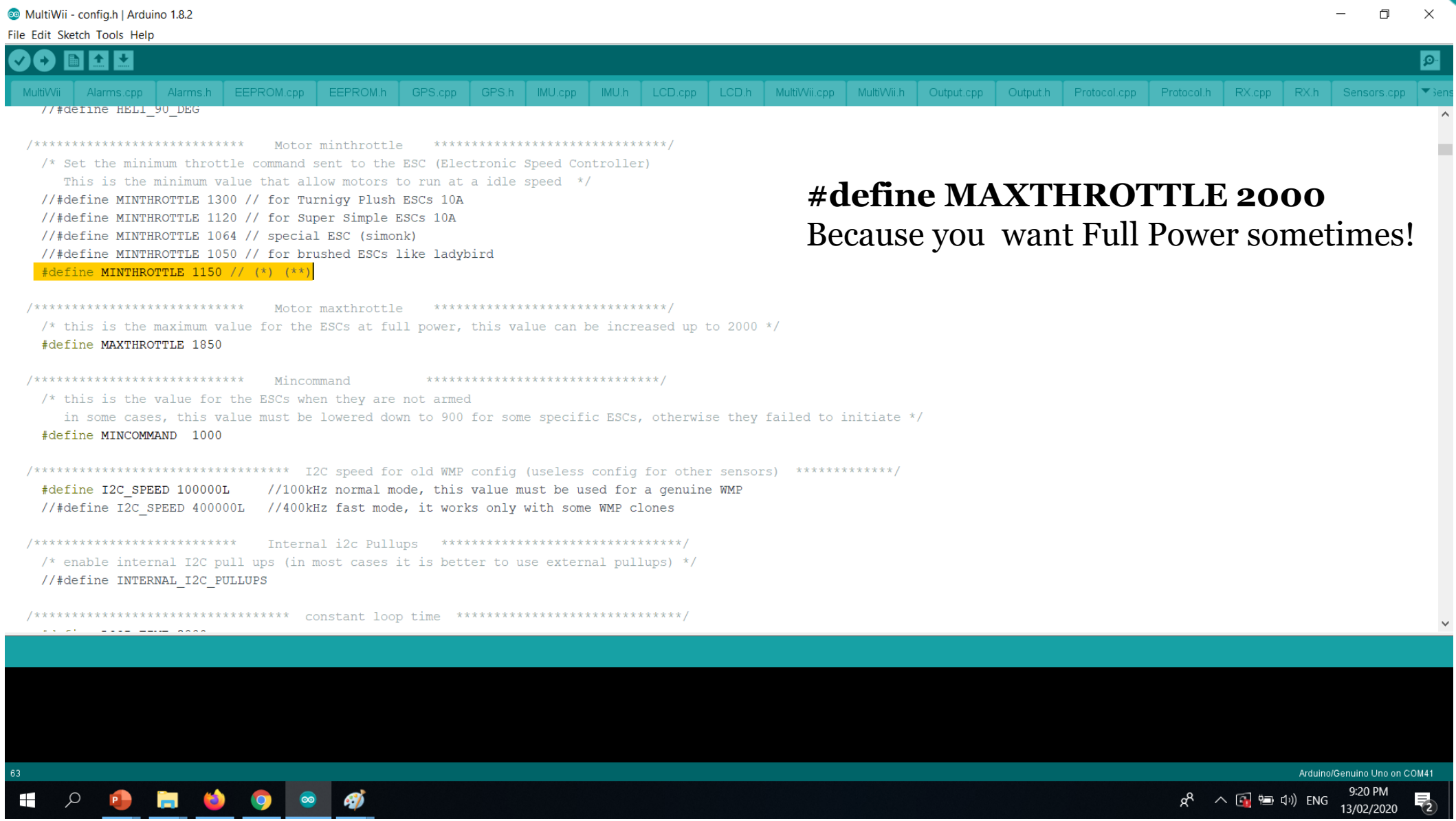
This also covers flaps and flapperons

Flying wing / Vtail modes are also set in this area



MOTOR MIN THROTTLE

CONFIG.H

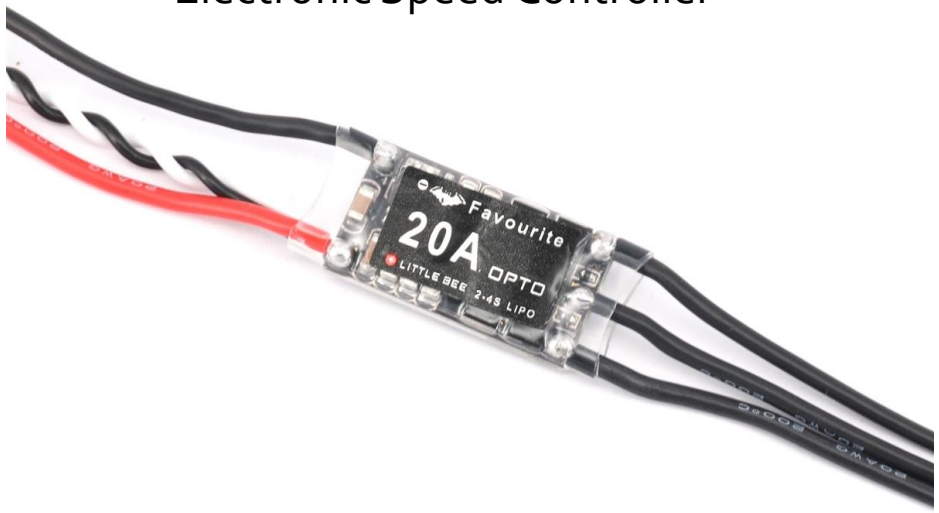


#define MAXTHROTTLE 2000

Because you want Full Power sometimes!

Motor Min Throttle

Electronic Speed Controller



Brushless Motor

1500

Idle Speed

MINTHROTTLE 1150

PWM Frequency

MAXTHROTTLE 1850

MINCOMMAND 800 - 1100

1900 – 2000

Output.cpp

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

Synerduino_Airplane_Diff_thust-GY91-1.8.16 - Output.cpp | Arduino 1.8.18

File Edit Sketch Tools Help

Synerduino_Airplane_Diff_thust-GY91-1.8.16 | 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

```
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  /*****************              BI Airplane Elevator Ailron             ******************/
1144    motor[0] = PIDMIX(0, 0, +1); //LEFT
1145    motor[1] = PIDMIX(0, 0, -1); //RIGHT
1146    servo[4] = (SERVODIR(4,2) * axisPID[PITCH]) + get_middle(4); //ELEVATOR
1147    servo[5] = (SERVODIR(5,2) * axisPID[ROLL]) + get_middle(5);//AILERON
1148    /*****************                BI Wing Elevon             ******************/
1149    //motor[0] = PIDMIX(0, 0, +1); //LEFT
1150    //motor[1] = PIDMIX(0, 0, -1); //RIGHT
1151    //servo[4] = (SERVODIR(4,2) * axisPID[ROLL]) + (SERVODIR(4,1) * axisPID[PITCH]) + get_middle(4); //LEFT
1152    //servo[5] = (SERVODIR(5,2) * axisPID[ROLL]) + (SERVODIR(5,1) * axisPID[PITCH]) + get_middle(5); //RIGHT
1153
1154  #elif defined( TRI )
1155    motor[0] = PIDMIX( 0,+4/3, 0); //REAR
1156    motor[1] = PIDMIX(-1,-2/3, 0); //RIGHT
1157    motor[2] = PIDMIX(+1,-2/3, 0); //LEFT
1158    servo[5] = (SERVODIR(5, 1) * axisPID[YAW]) + get_middle(5); //REAR
1159  #elif defined( QUADP )
1160    motor[0] = PIDMIX( 0,+1,-1); //REAR
1161    motor[1] = PIDMIX(-1, 0,+1); //RIGHT
1162    motor[2] = PIDMIX(+1, 0,+1); //LEFT
1163    motor[3] = PIDMIX( 0,-1,-1); //FRONT
1164  #elif defined( QUADX )
1165    motor[0] = PIDMIX(-1,+1,-1); //REAR_R
1166    motor[1] = PIDMIX(-1,-1,+1); //FRONT_R
1167    motor[2] = PIDMIX(+1,+1,+1); //REAR_L
1168    motor[3] = PIDMIX(+1,-1,-1); //FRONT_L
1169  #elif defined( Y4 )
1170    motor[0] = PIDMIX(+0,+1,-1);   //REAR_1 CW
1171    motor[1] = PIDMIX(-1,-1, 0); //FRONT_R CCW
1172    motor[2] = PIDMIX(+0,+1,+1);   //REAR_2 CCW
1173    motor[3] = PIDMIX(+1,-1, 0); //FRONT_L CW
1174  #elif defined( Y6 )
1175    motor[0] = PIDMIX(+0,+4/3,+1); //REAR
1176    motor[1] = PIDMIX(-1,-2/3,-1); //RIGHT
1177    motor[2] = PIDMIX(+1,-2/3,-1); //LEFT
1178    motor[3] = PIDMIX(+0,+4/3,-1); //UNDER_REAR
1179    motor[4] = PIDMIX(-1,-2/3,+1); //UNDER_RIGHT
```

1147 - 1144

Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM53

Links

ENG 9:23 AM 07/07/2023

INERTIAL MEASURING UNIT MEASURING UNIT

Pls 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

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

Magnetometer

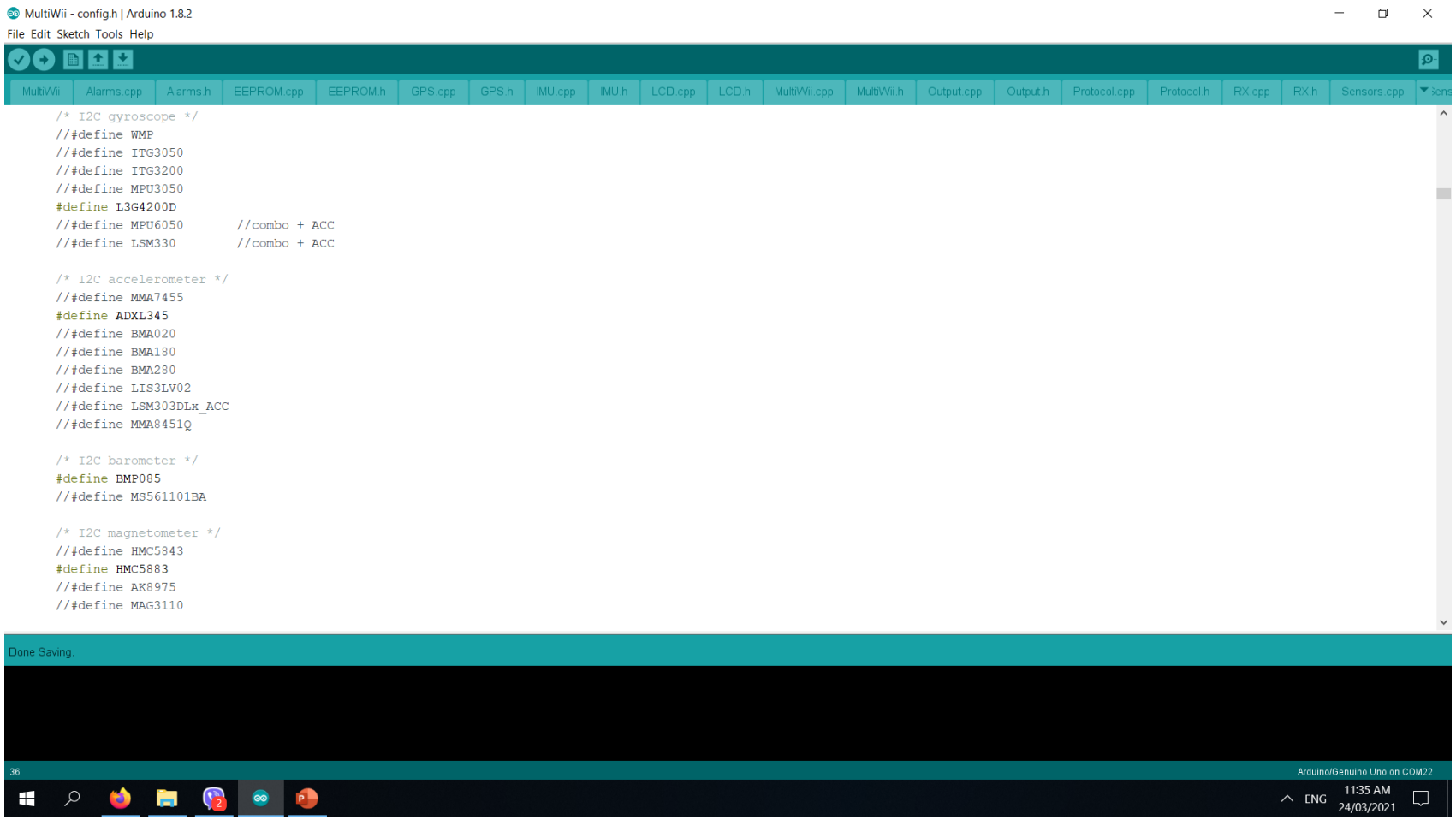
Barometer

Accelerometer

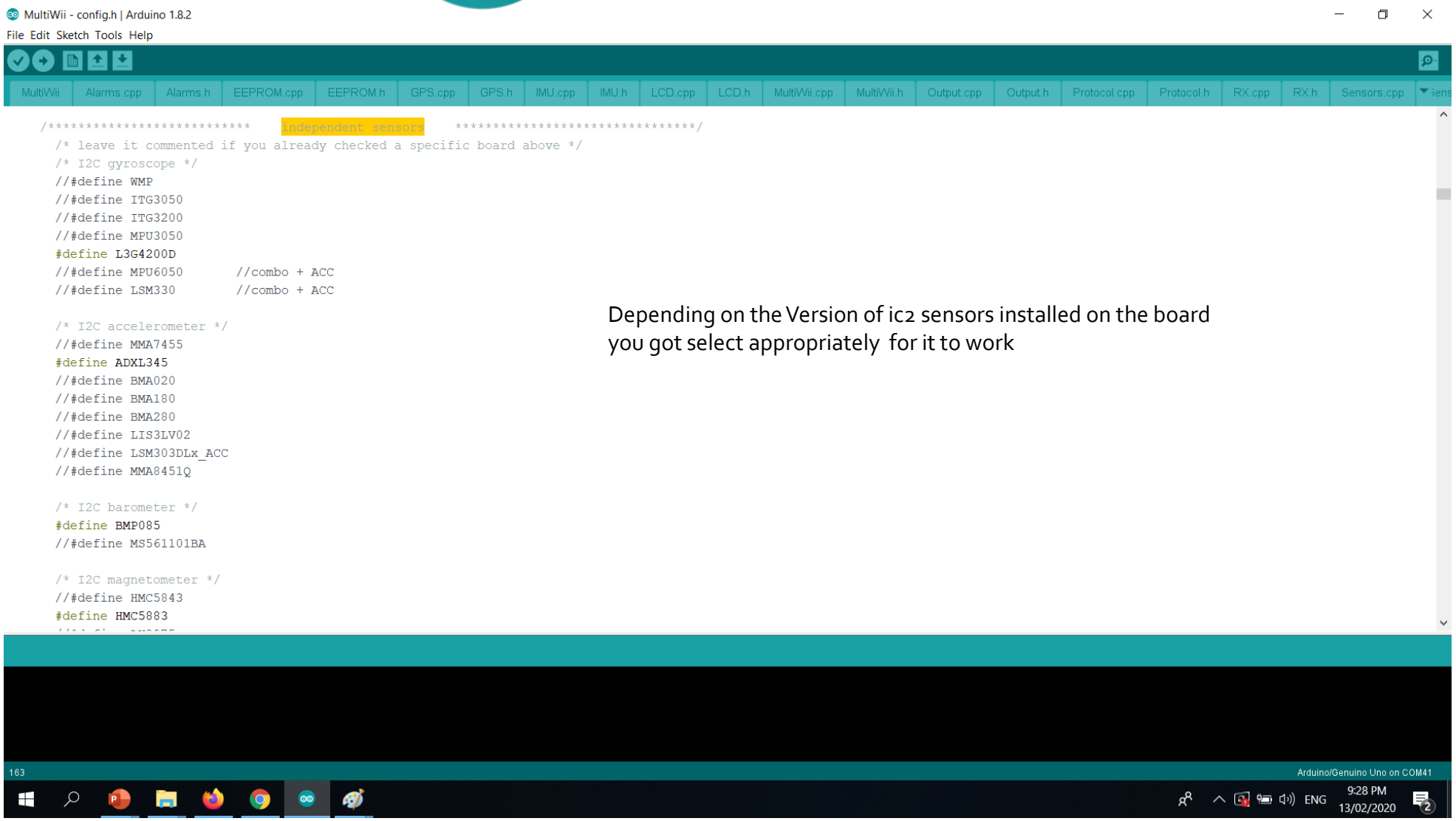
Gyroscope

Config.h

Sensors
#define L3G4200D
#define ADXL345
#define BMP085
#define MMC5883



CONFIG.H

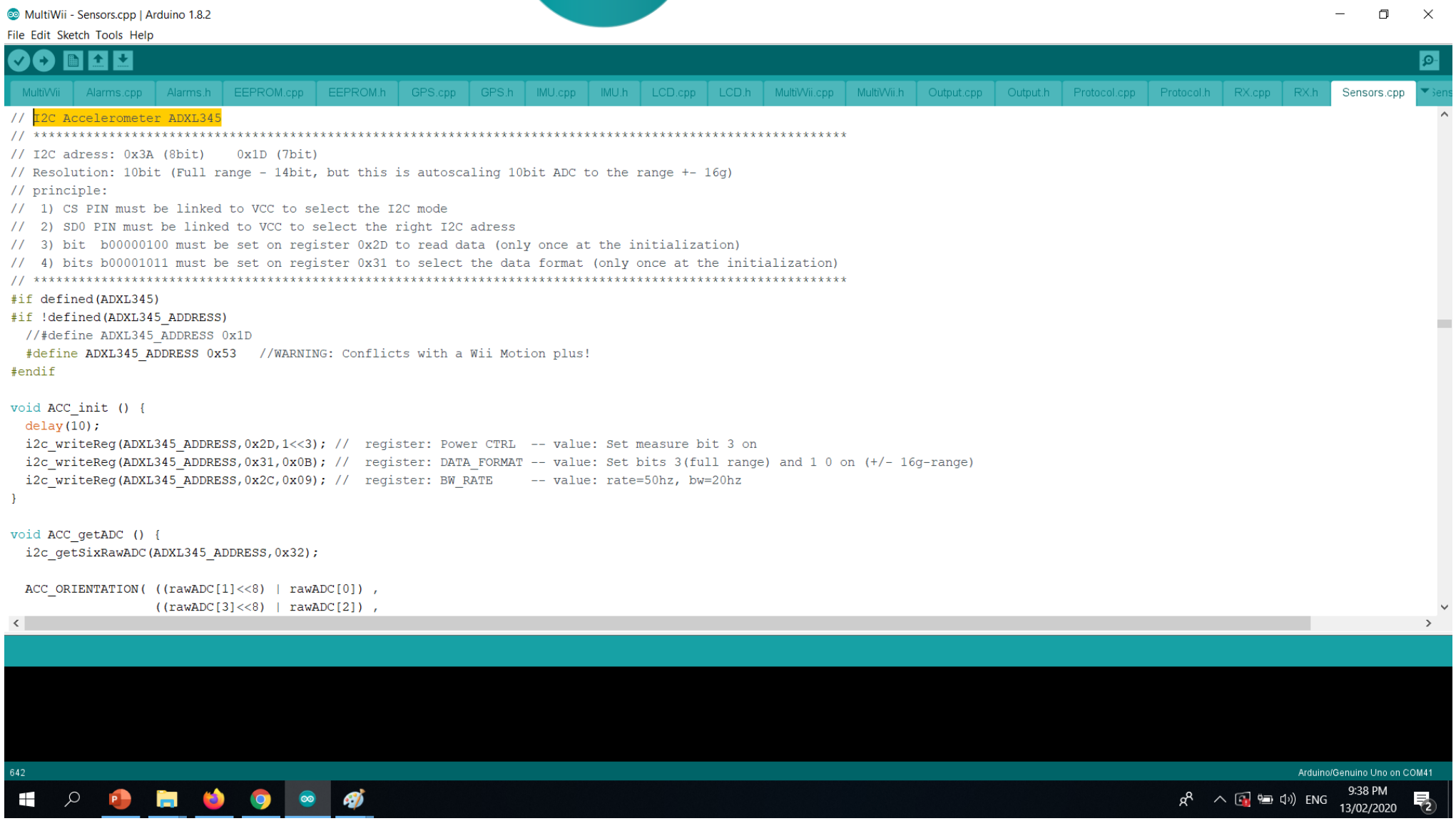


Depending on the Version of ic2 sensors installed on the board you got select appropriately for it to work

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
ENG
9:37 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
// 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) SD0 PIN must be linked to VCC to select the right I2C adress
// 3) bit b00000100 must be set on register 0x2D to read data (only once at the initialization)
// 4) bits b00001011 must be set on register 0x31 to select the data format (only once at the initialization)
// **
#if defined(ADXL345)
#if !defined(ADXL345_ADDRESS)
  //#define ADXL345_ADDRESS 0x1D
  #define ADXL345_ADDRESS 0x53   //WARNING: Conflicts with a Wii Motion plus!
#endif
void ACC_init () {
  delay(10);
  i2c_writeReg(ADXL345_ADDRESS,0x2D,1<<3); // register: Power CTRL -- value: Set measure bit 3 on
  i2c_writeReg(ADXL345_ADDRESS,0x31,0x0B); // register: DATA_FORMAT -- value: Set bits 3(full range) and 1 0 on (+/- 16g-range)
  i2c_writeReg(ADXL345_ADDRESS,0x2C,0x09); // register: BW_RATE   -- value: rate=50hz, bw=20hz
}
void ACC_getADC () {
  i2c_getSixRawADC(ADXL345_ADDRESS,0x32);
  ACC_ORIENTATION(((rawADC[1]<<8) | rawADC[0]) ,
                   ((rawADC[3]<<8) | rawADC[2]) ,
642
Arduino/Genuino Uno on COM41
ENG
9:38 PM
13/02/2020

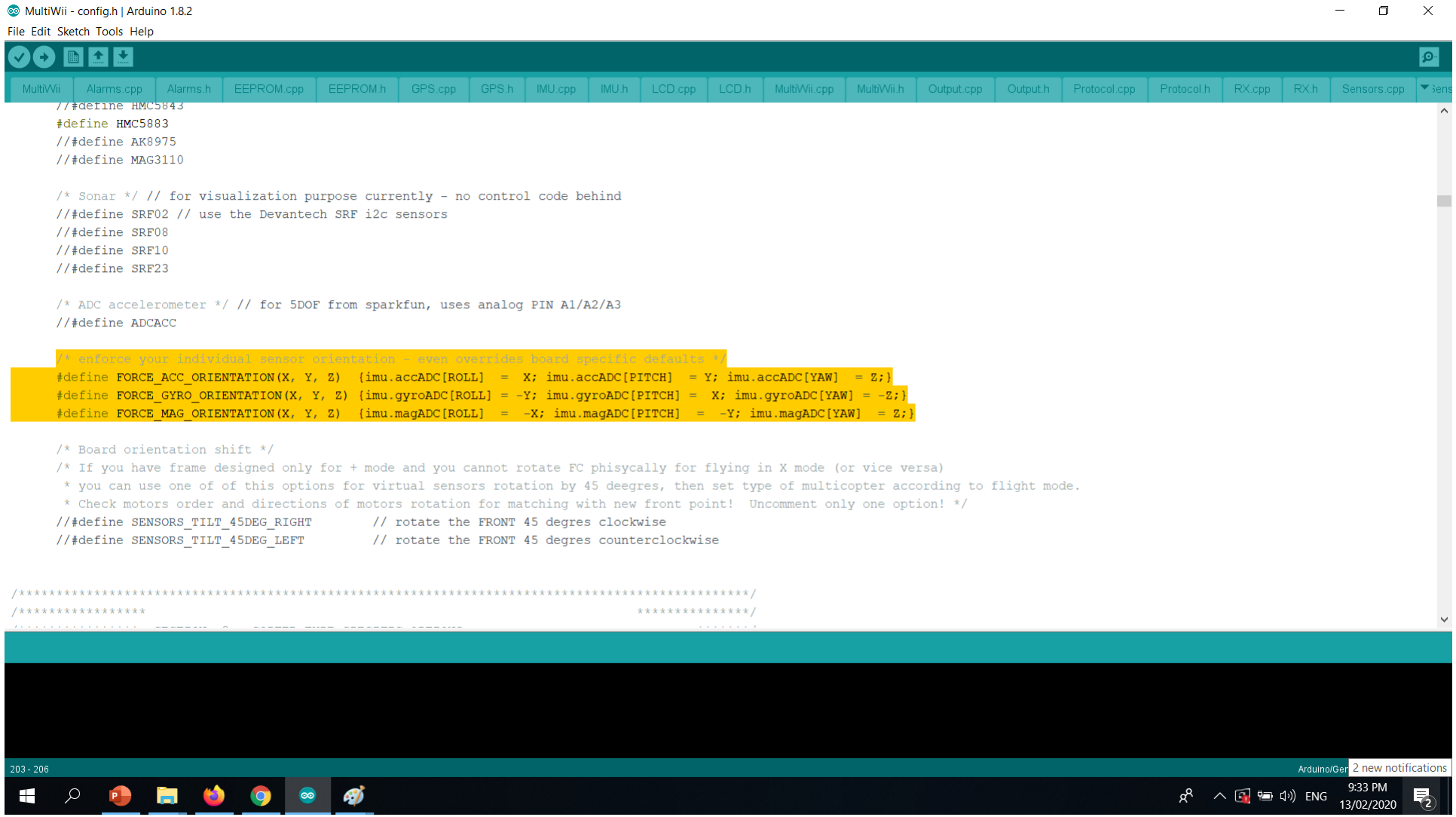
SENSORS.CCP


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

SENSORS.CPP


MultiWii - Sensors.cpp | Arduino 1.8.2
File Edit Sketch Tools Help
MultiWii
Alarms.cpp
Alarms.h
EEPROM.cpp
EEPROM.h
GPS.cpp
GPS.h
IMU.cpp
IMU.h
LCD.cpp
LCD.h
MultiWii.cpp
MultiWii.h
Output.cpp
Output.h
Protocol.cpp
Protocol.h
RX.cpp
RX.h
Sensors.cpp
// I2C Compass MMC5883
// I2C adress: 0x30 (8bit) 0x00 (7bit)
#if defined(MMC5883)
#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
1264 - 1265
Arduino/Genuino Uno on COM22
ENG
3:22 PM
31/03/2021

SENSOR ORIENTATION CONFIG.H



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

SERIAL COM

CONFIG.H

SERIAL BAUD RATE

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

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

SERIAL 1 , 3 IS RESERVE FOR TELEMETRY

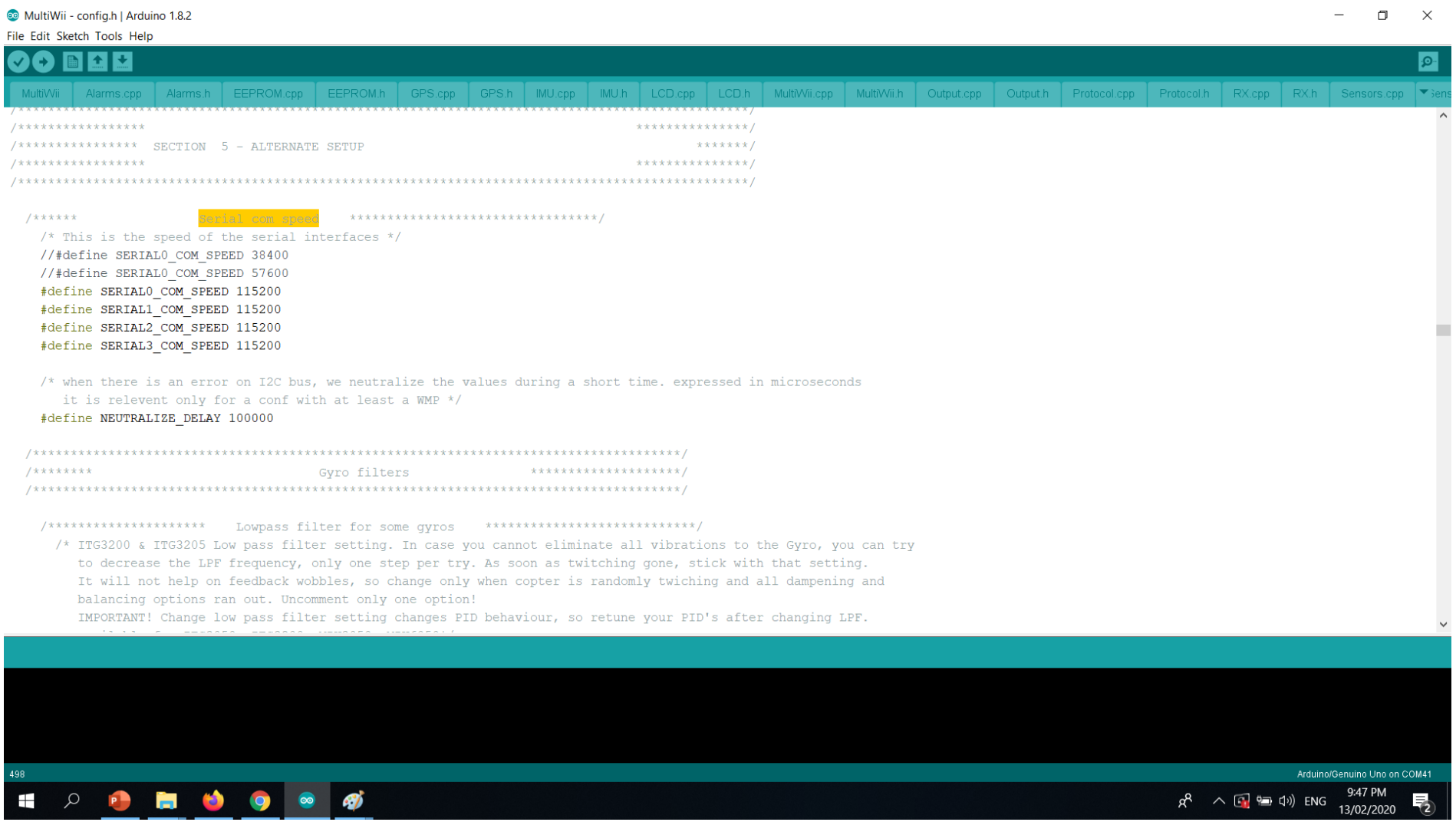
115200 FOR BLUETOOTH HC-05

38400 FOR XBEE RADIO

57600 for SIK RADIO

SERIAL 2 IS RESERVE FOR GPS

NMEA Baud 57600



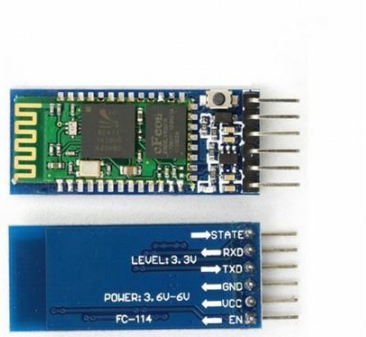
BLUETOOTH

Bluetooth setup with the USB TTL and Arduino IDE

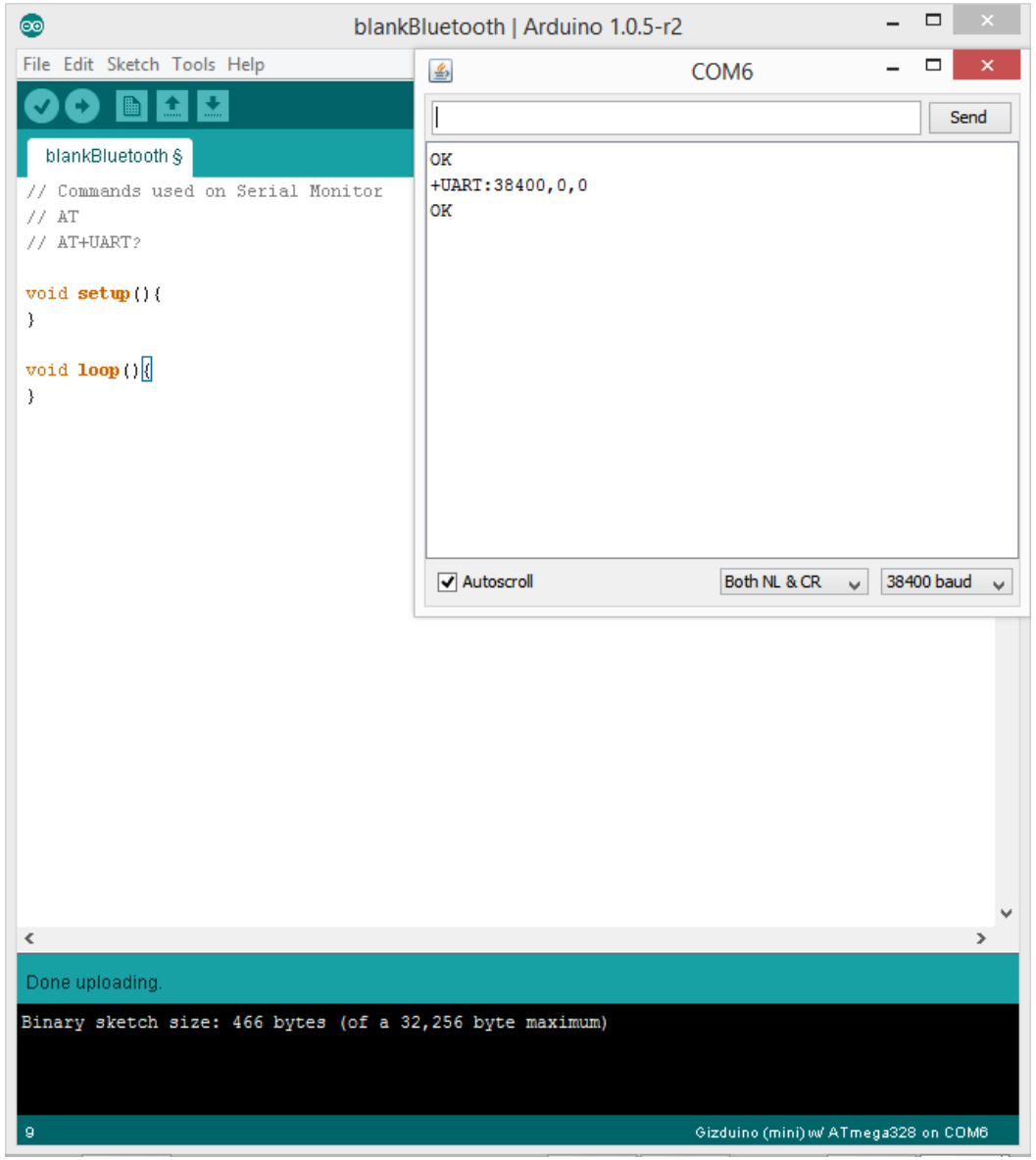
Arduino IDE>Tools>Serial Monitor (Push Button Before Connecting the USB) Set (Baud 38400) (Both NL & CR)

AT : check the connection
AT+VERSION : Check Version

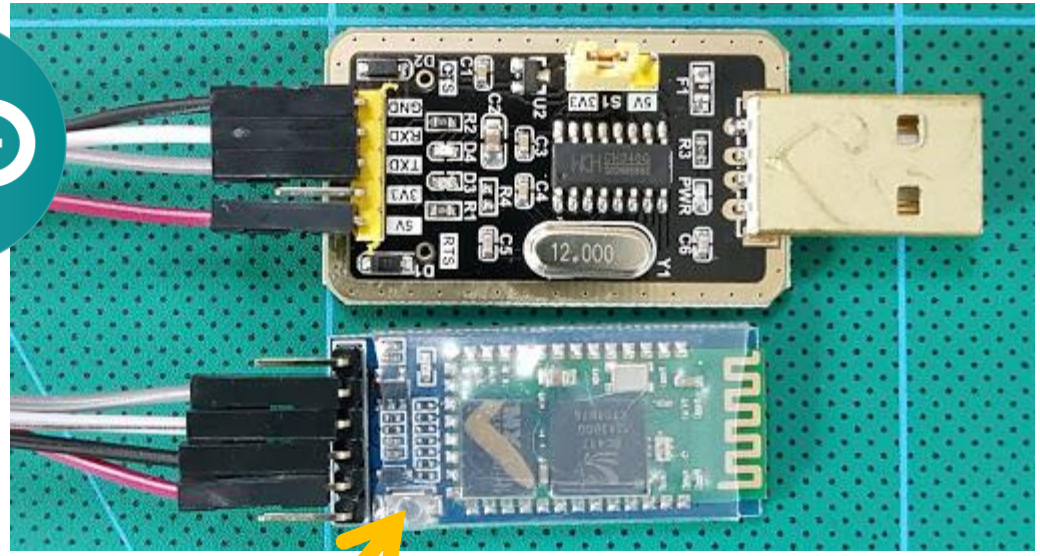
HC-05 (Recommended)
AT+NAME=Change name
AT+PSWD=1234 (Version 2)
AT+PSWD="1234" (Version 3)
AT+UART=115200,1,0

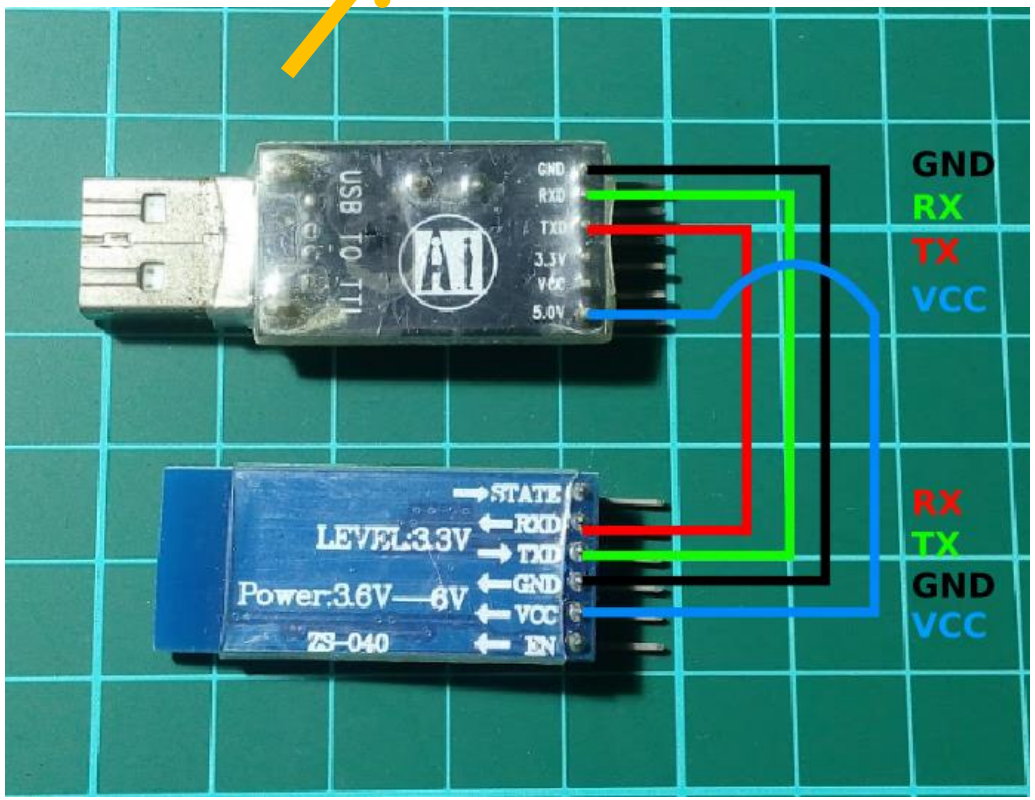


HC-05 (Recommended)



Bluetooth setup with the USB TTL and Arduino IDE





Arduino IDE>Tools>Serial Monitor (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

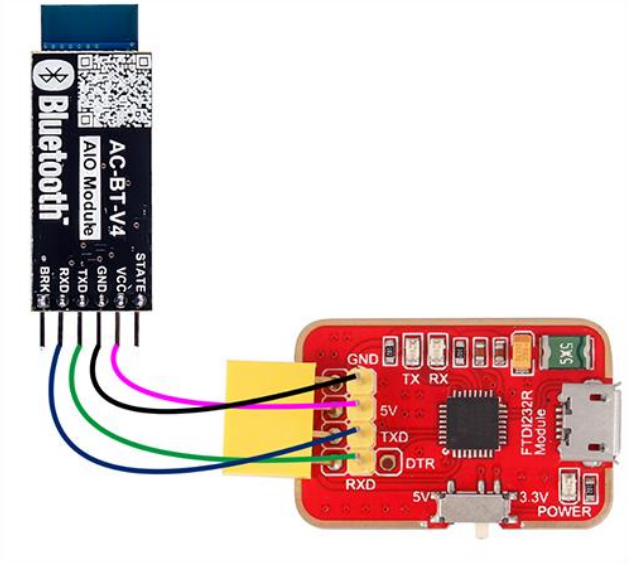
HC-05 (Recommended)

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

HM-10 (Original)

AT (Check if new configuration is working)
AT+NAME (Query name)
AT+ADDR (Query Mac address)
AT+BAUD (Query Baud)
AT+PASS (Query current Pincode)
AT+PIN **(Query current Pincode on some BL module)**
AT+TYPE (Query authentication mode)
AT+ROLE (Query Peripheral (Slave) or Central (Master) mode)

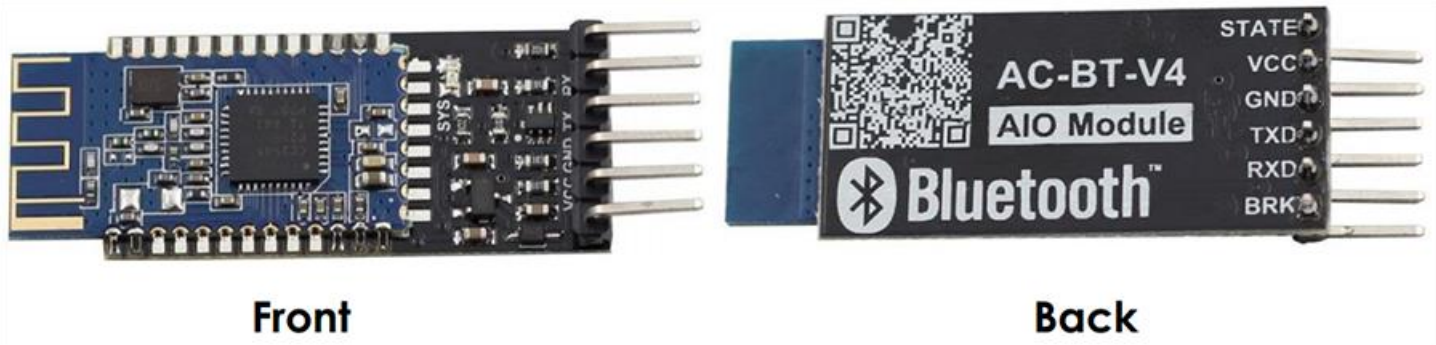
AT+TYPE
0:Not need PIN Code
1:Auth not need PIN
2:Auth with PIN
3:Auth and bond

AT+BAUD
0 – 9600:
1 – 19200
2 – 38400
3 – 57600 (Some BL its 4800)
4 – 115200
5 – 4800
6 – 2400
7 – 1200
8 – 230400 (Some BL its 115200)

AT+NAMEArduinoDrone
AT+BAUD4 set baud to 115200 (we want this for high speed)
AT+BAUD8 set baud to 115200 (on some BL module)
AT+PASS123456 Set password to 123456
AT+PIN123456 Set password to 123456 (on some BL module)

AT+ROLE
0 = Slave or Peripheral
1 = Master or Central.

AT+TYPE2
AT+TYPE1 (on Some BL modules)
AT+ROLE0



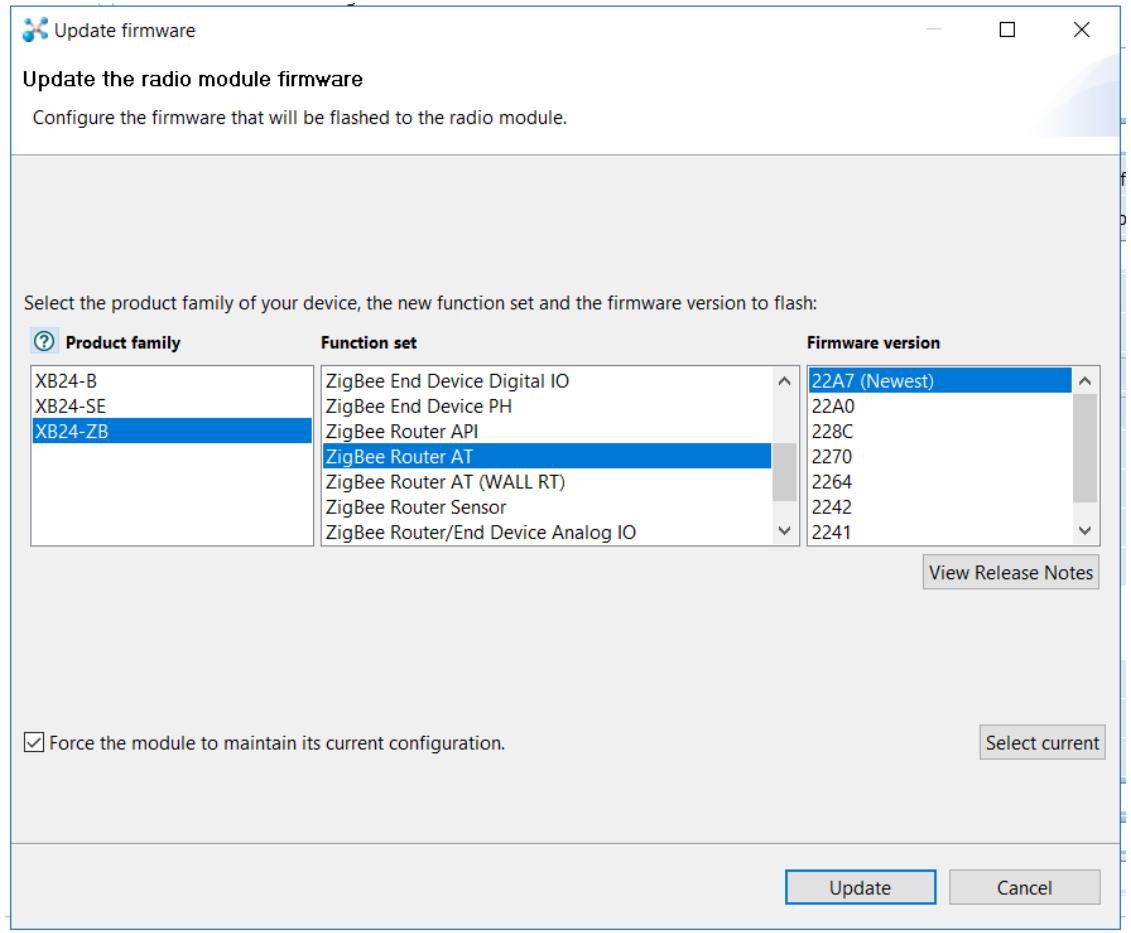
Front Back

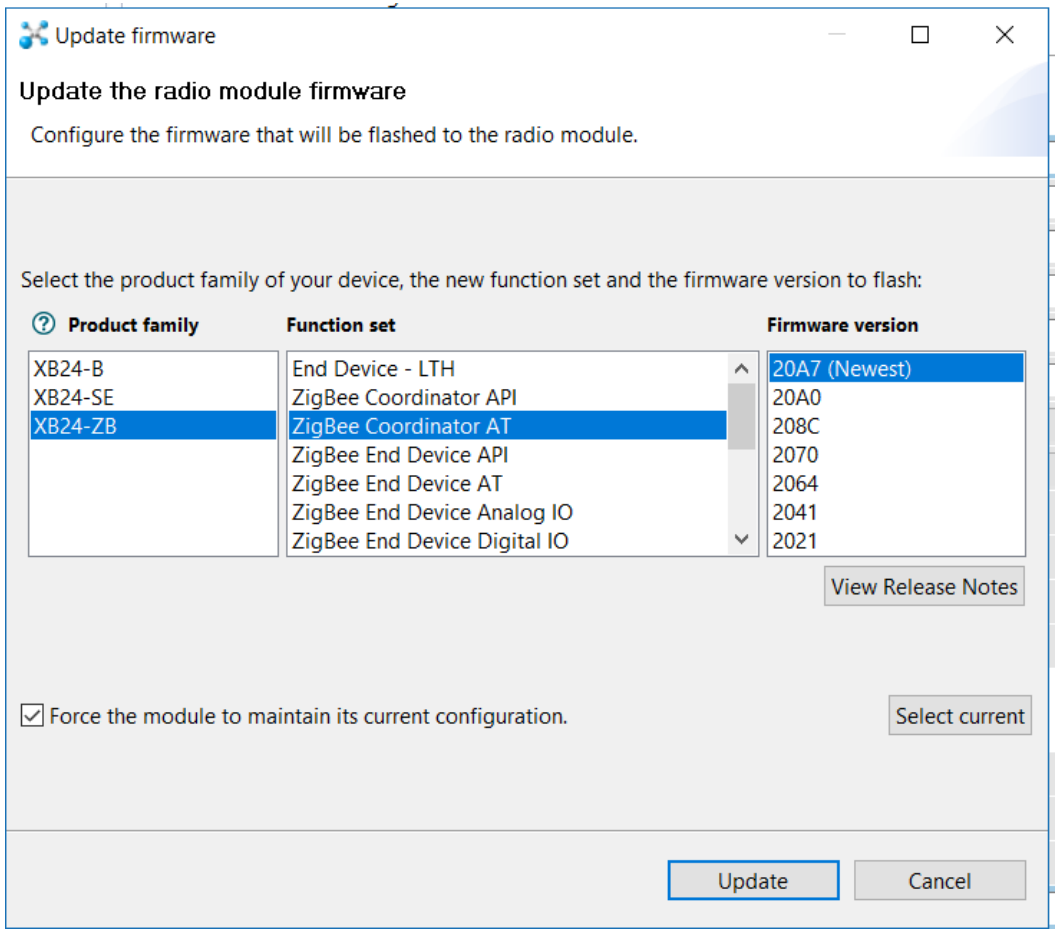
Note : there are several clones of this type in the market that can be very difficult to setup

XBEE RADIO

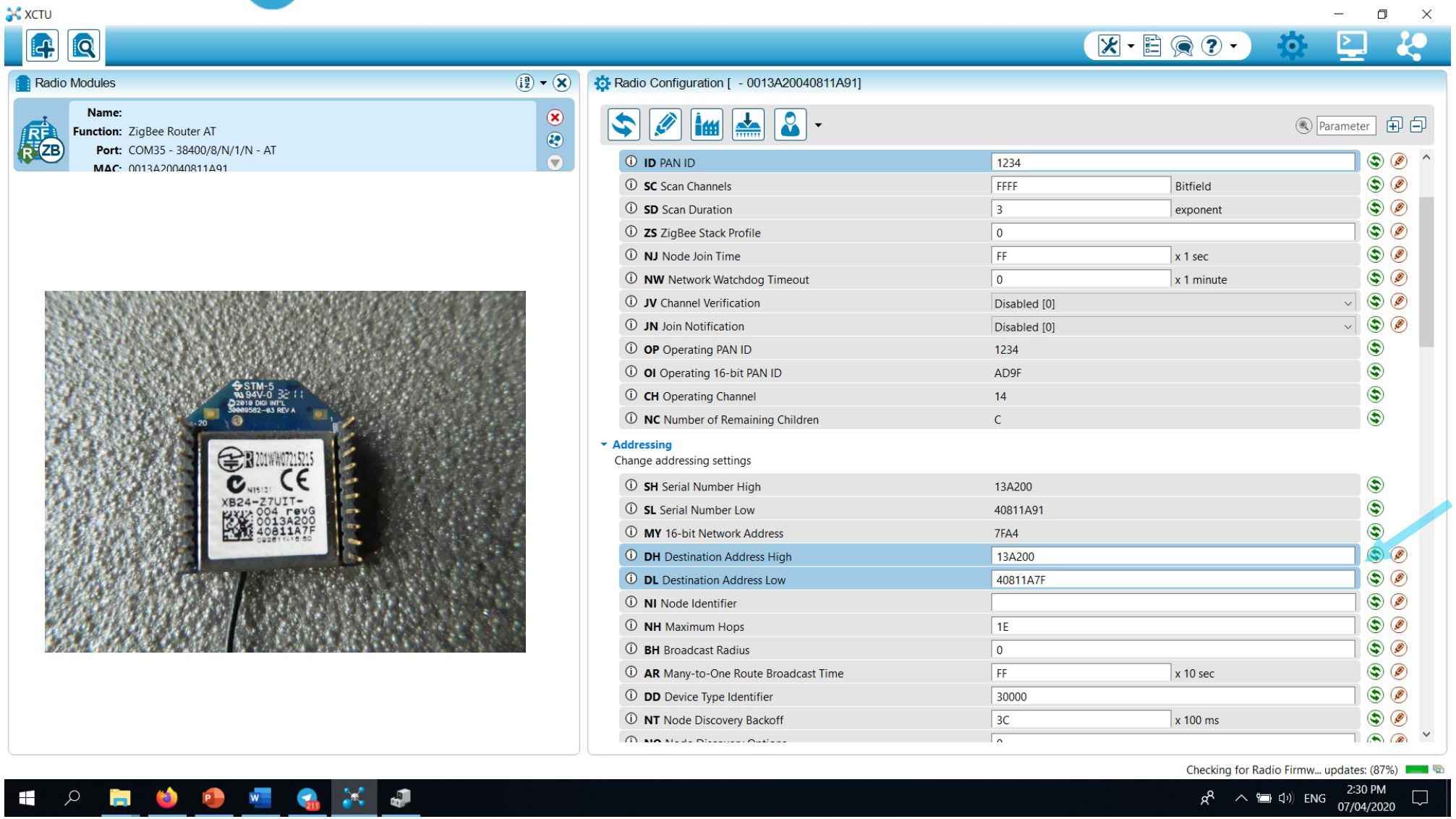
GROUND STATION ROUTER
38400 8/N/1/N - AT

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





GROUND STATION



GROUND STATION

XCTU

Radio Modules

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

Radio Configuration [- 0013A20040811A91]

Parameter

Security
Change security parameters

EE Encryption Enable	Disabled [0]	
EO Encryption Options	0	Bitfield
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

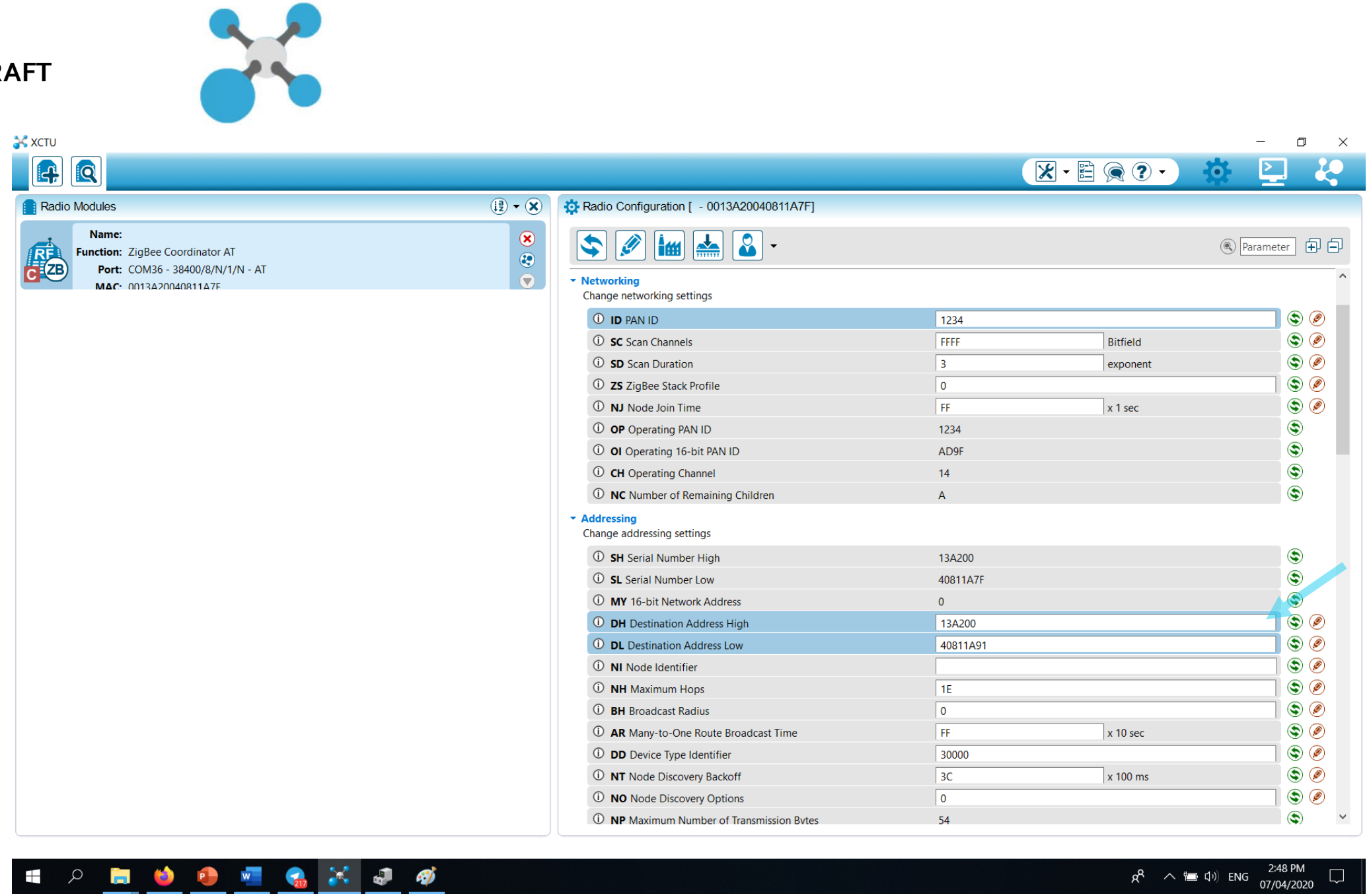
CT AT Command Mode Timeout	64	x 100ms
GT Guard Times	3E8	x 1ms
CC Command Sequence Character	2B	Recommended: 0x20-0x7F (ASCII)

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

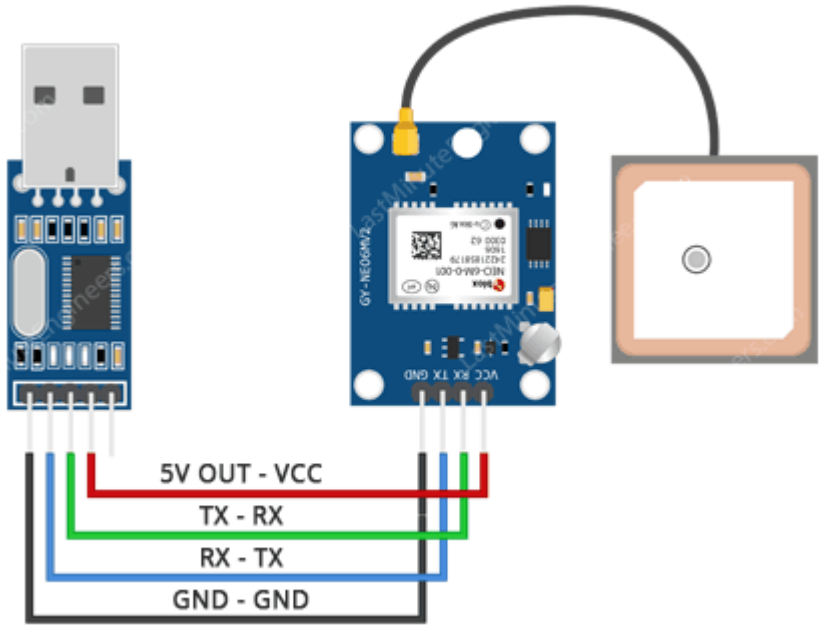
ENG 2:36 PM 07/04/2020

AIRCRAFT


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

GPS

GPS CONFIGURING



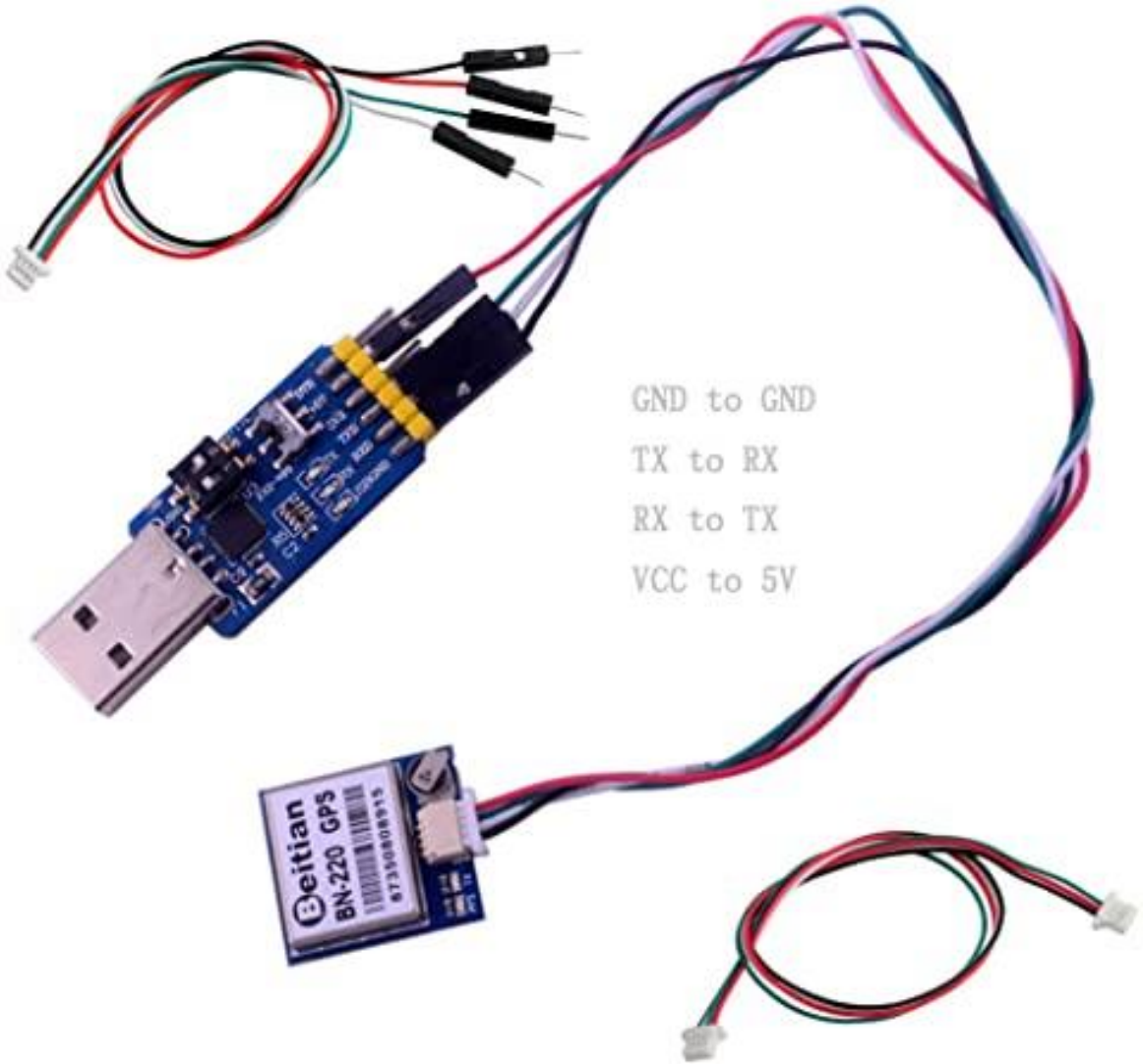
U BLOX NEO 6

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

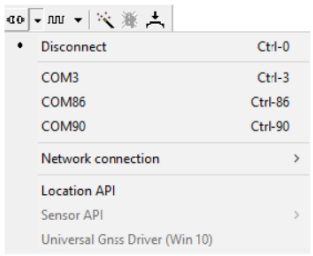
PLUG IN TO SERIAL TX 2 RX 2

USB TTL TO PROGRAM THE GPS

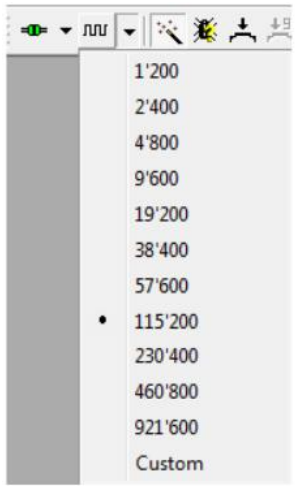
THIS GOES SAME ON THE DRONE SHIELD

U CENTER

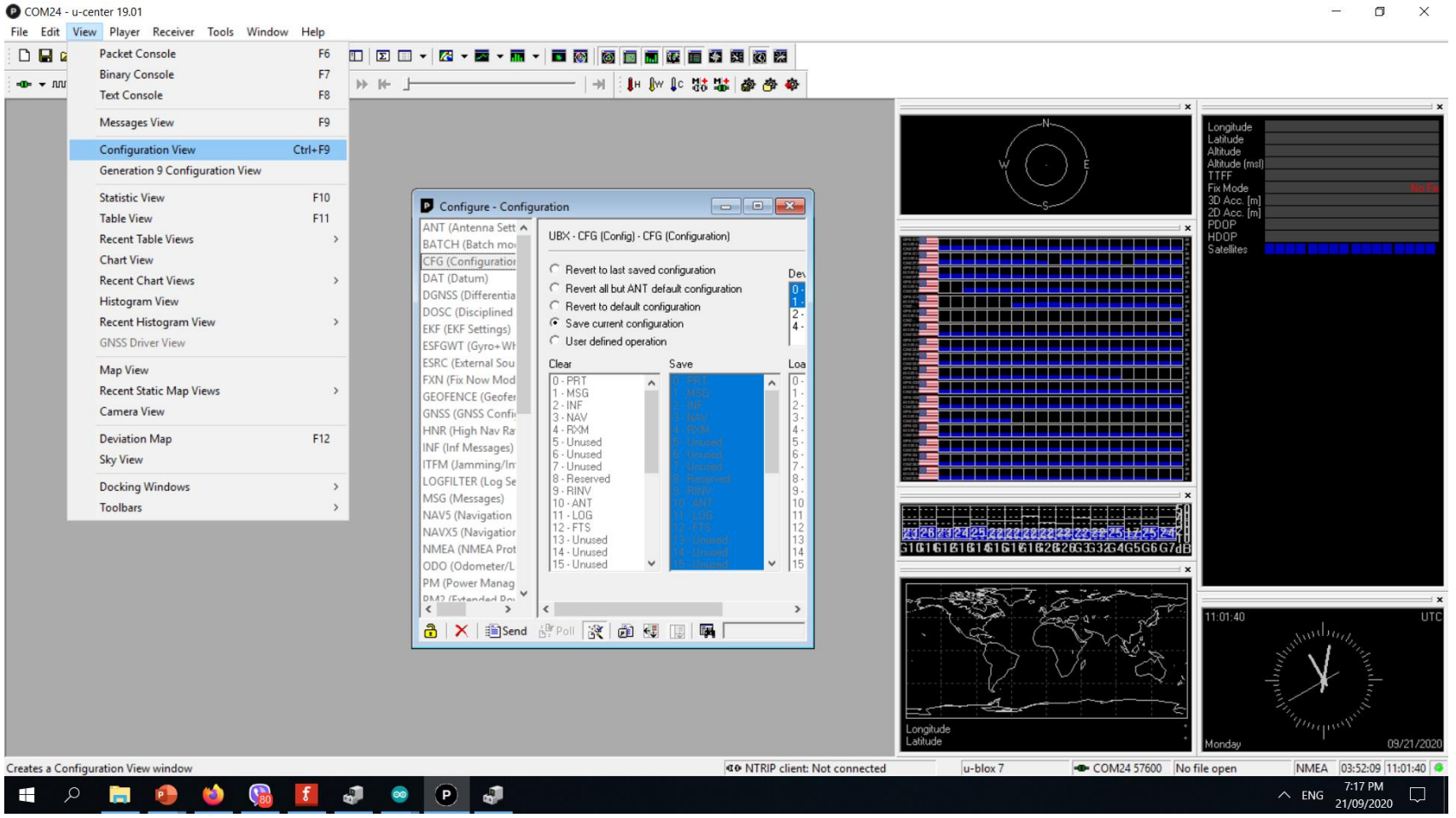
Connect to the device. Select the COM port your GPS and USBTTL is connected to



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

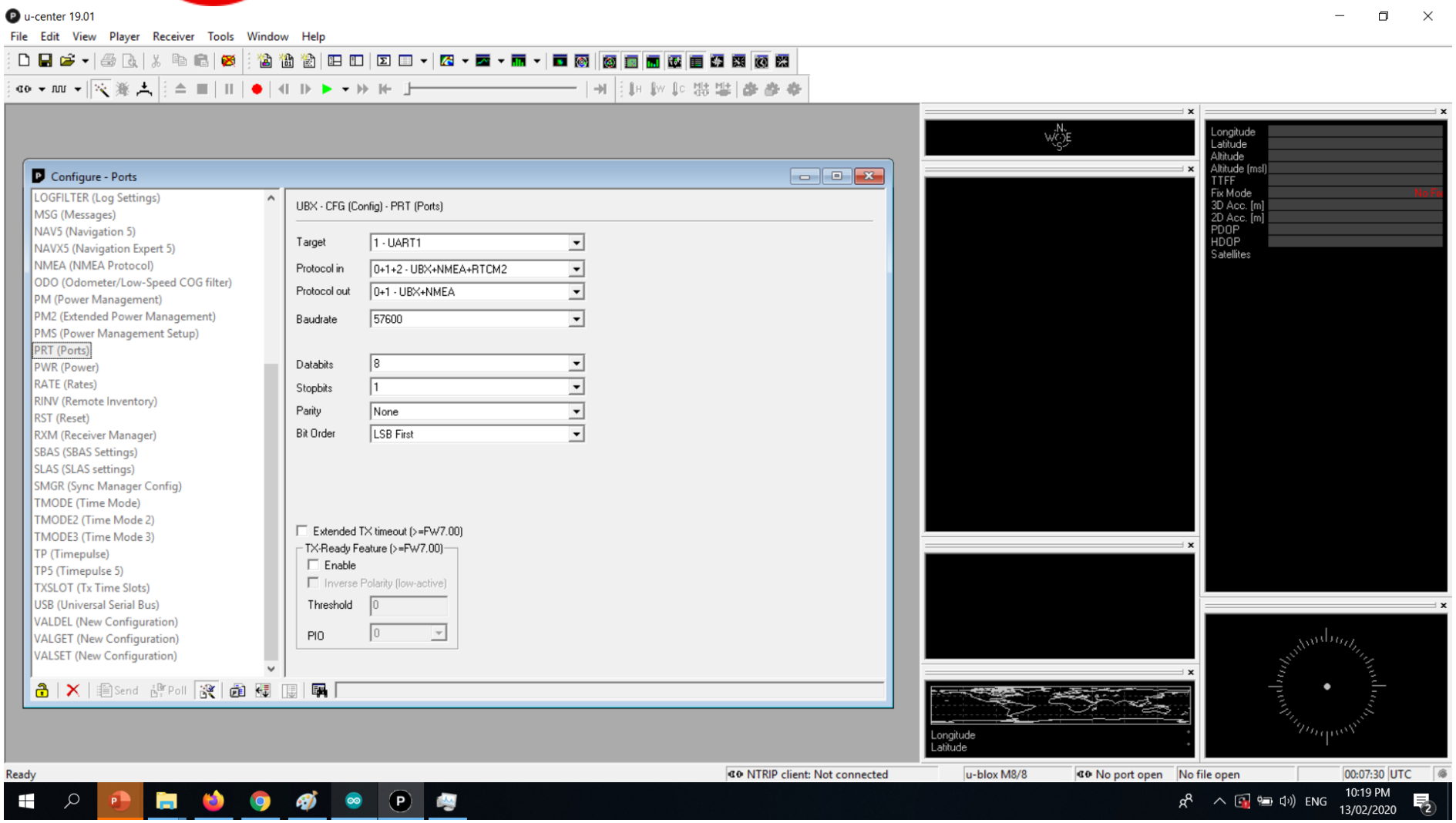


Go to View> Configuration View



U CENTER

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

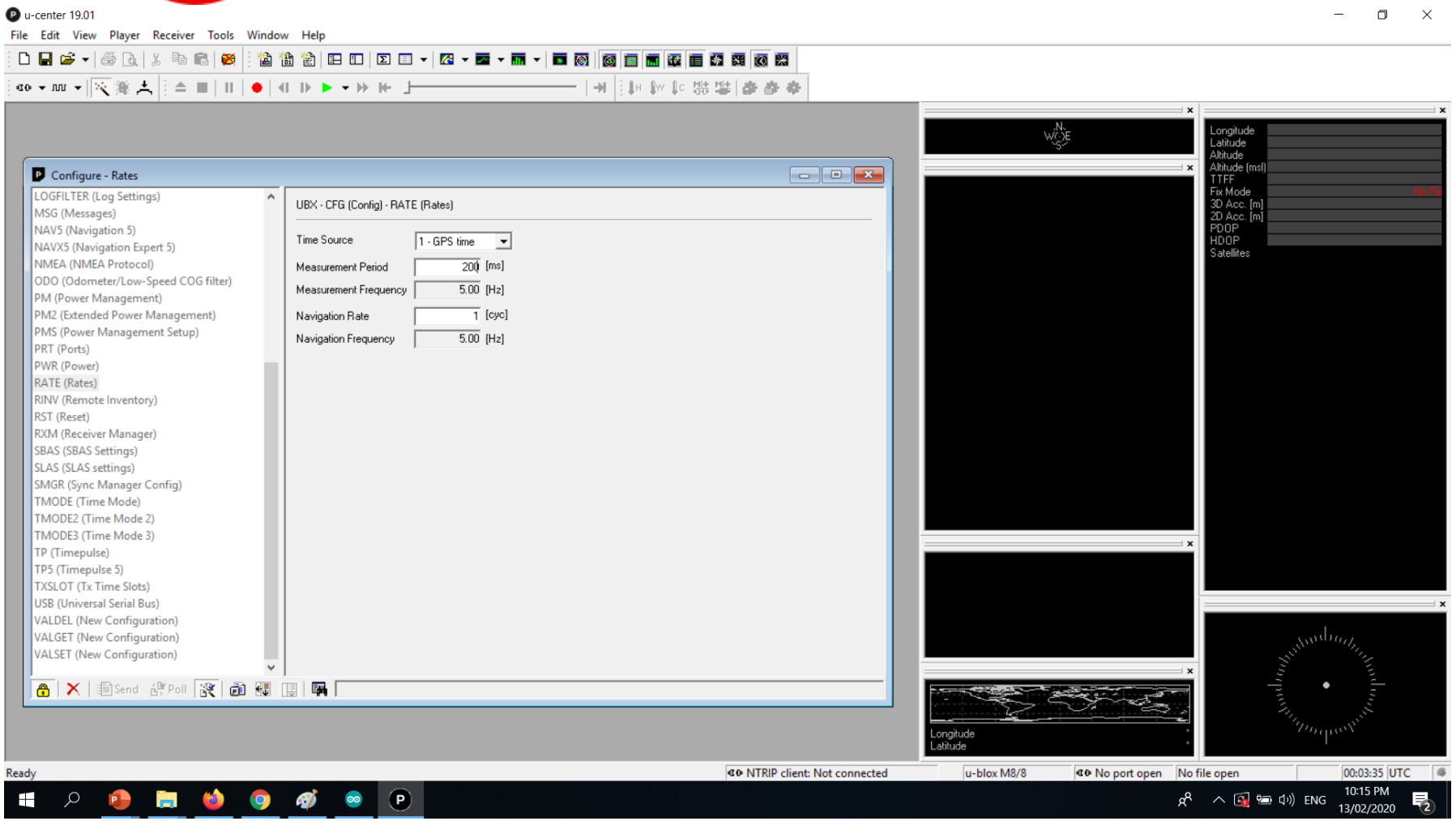


U CENTER

Rate 200ms
Frequency 5hz

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

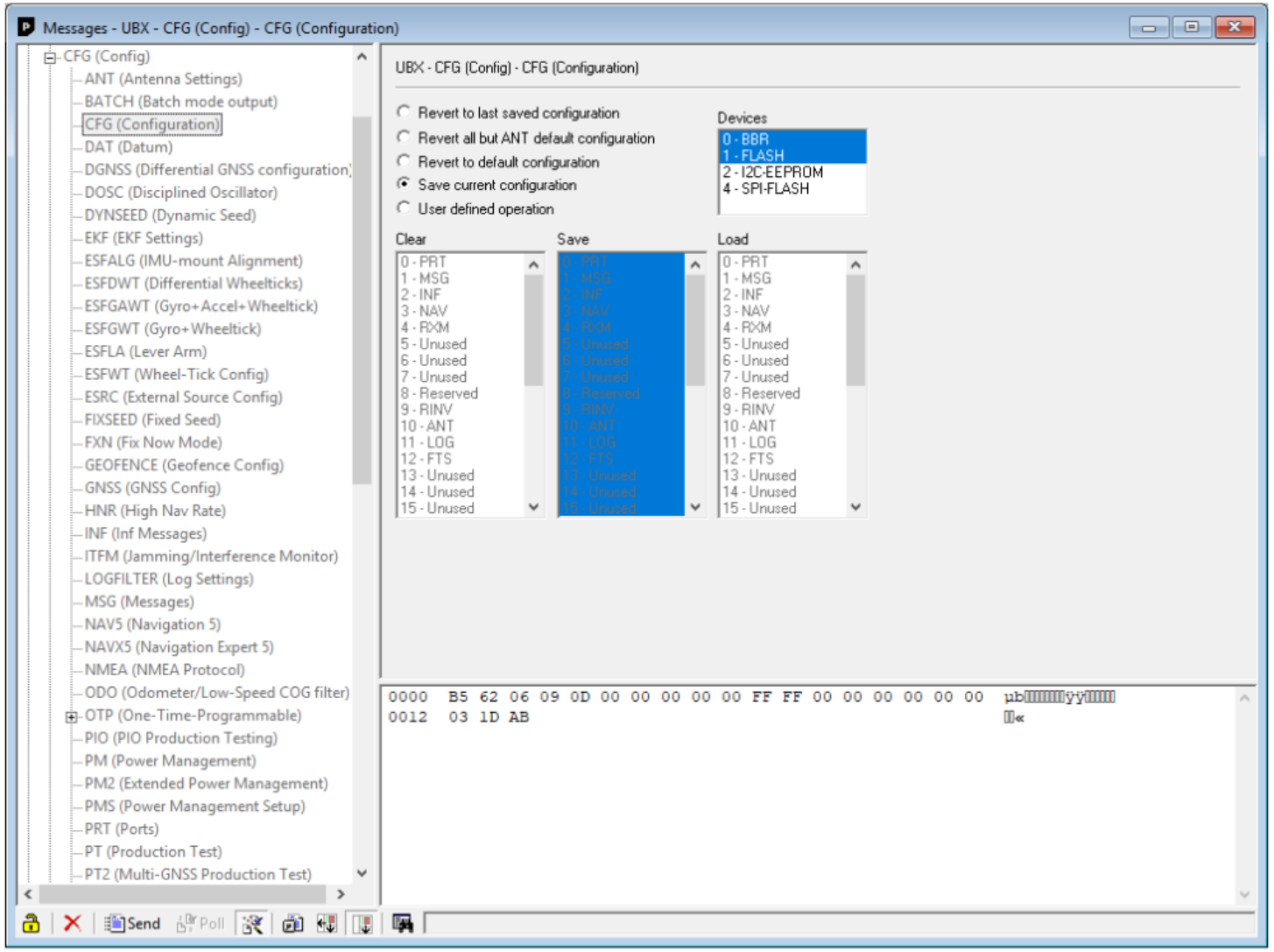
Exit the configuration window this will promp you to the save parameters .
Hit “Yes”



U CENTER

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

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



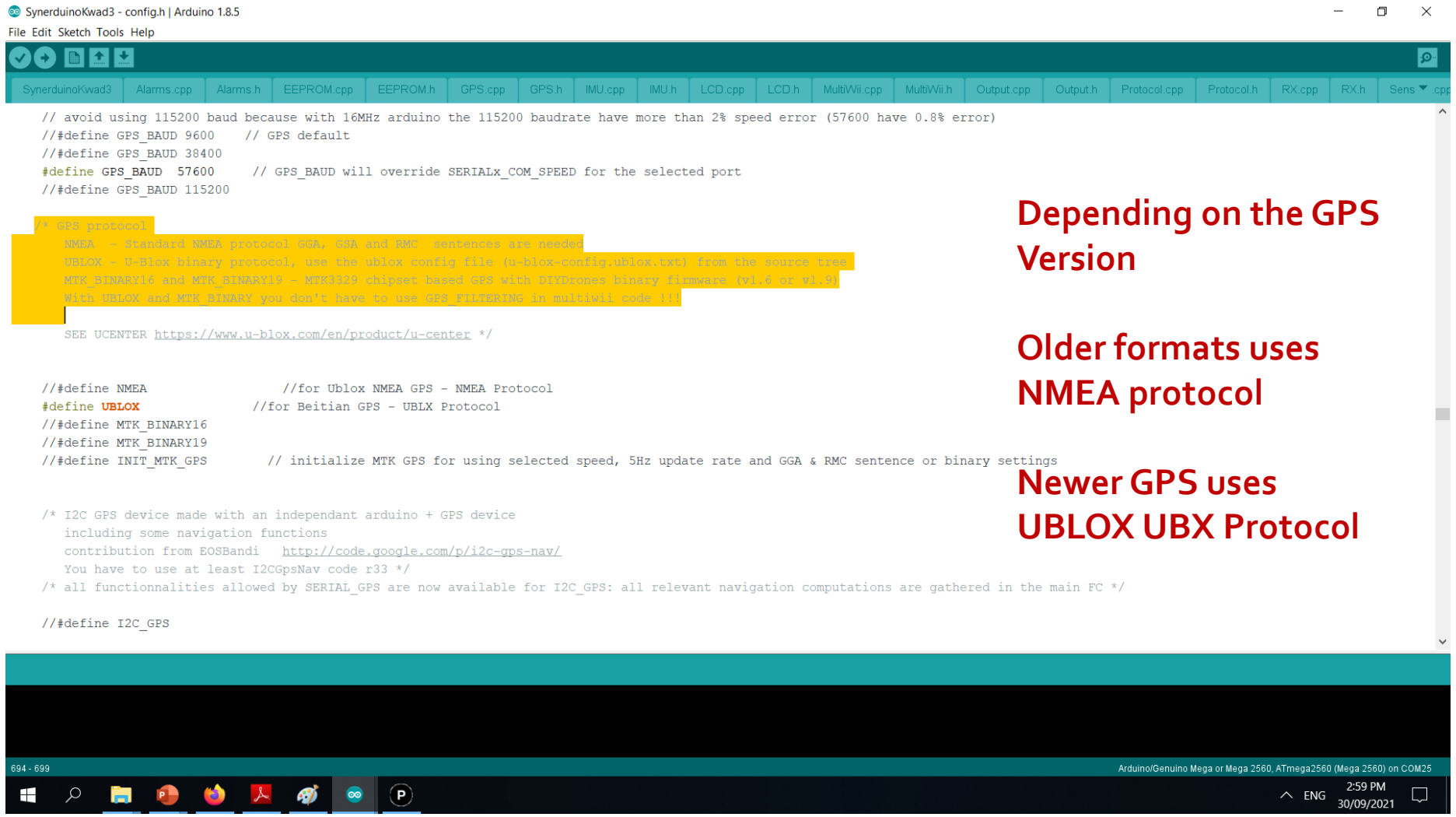
GPS

CONFIG.H

Set GPS BAUD to 57600 as configured to the GPS module

GPS

CONFIG.H

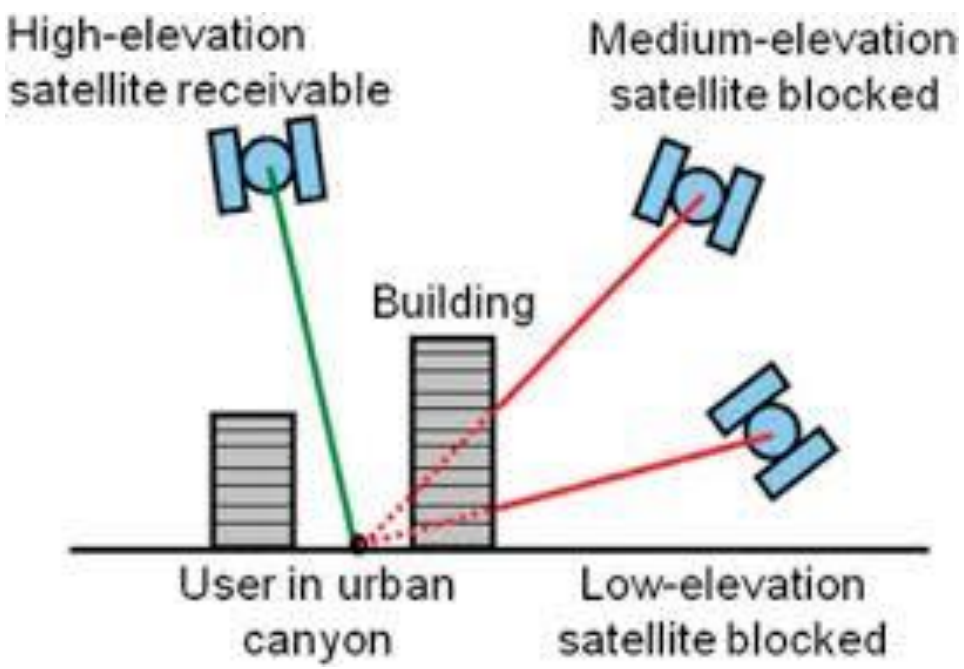


Depending on the GPS Version

Older formats uses NMEA protocol

Newer GPS uses UBLOX UBX Protocol

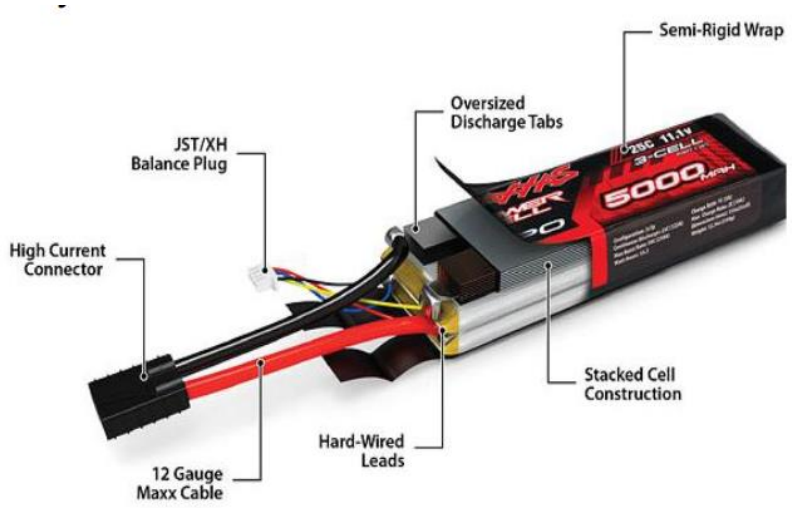
GPS

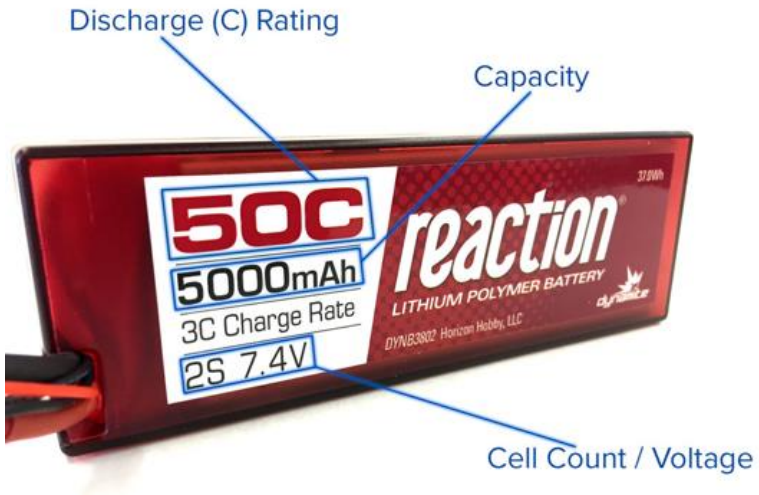


Note : GPS require a clear open area to get a proper fix and accuracy minimum 7 satellites but 10+ are Ideal

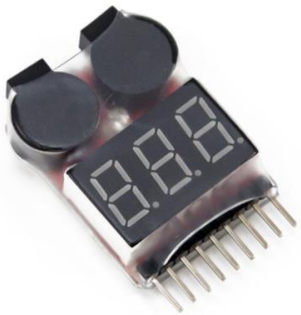
Flying next to a building can distort satellite signal deteriorating accuracy

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





BATTERY



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

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

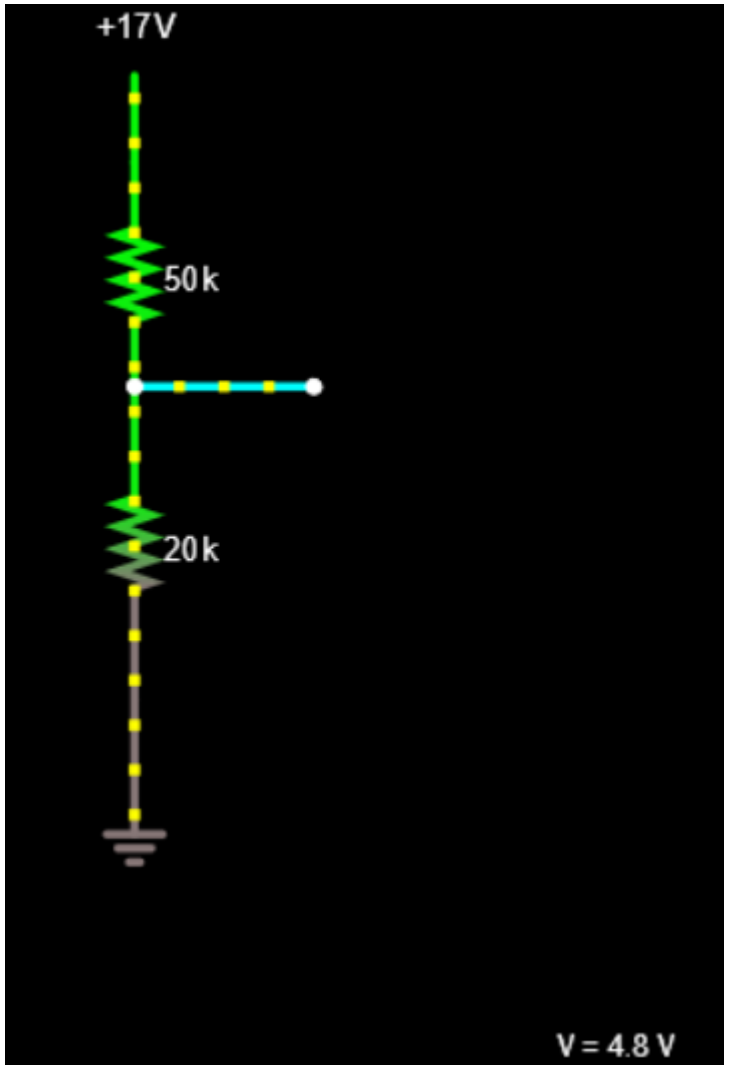
BATTERY STORAGE MODE IS 3.8V PER CELL

BATTERY DISCHARGE IS 3.6V PER CELL

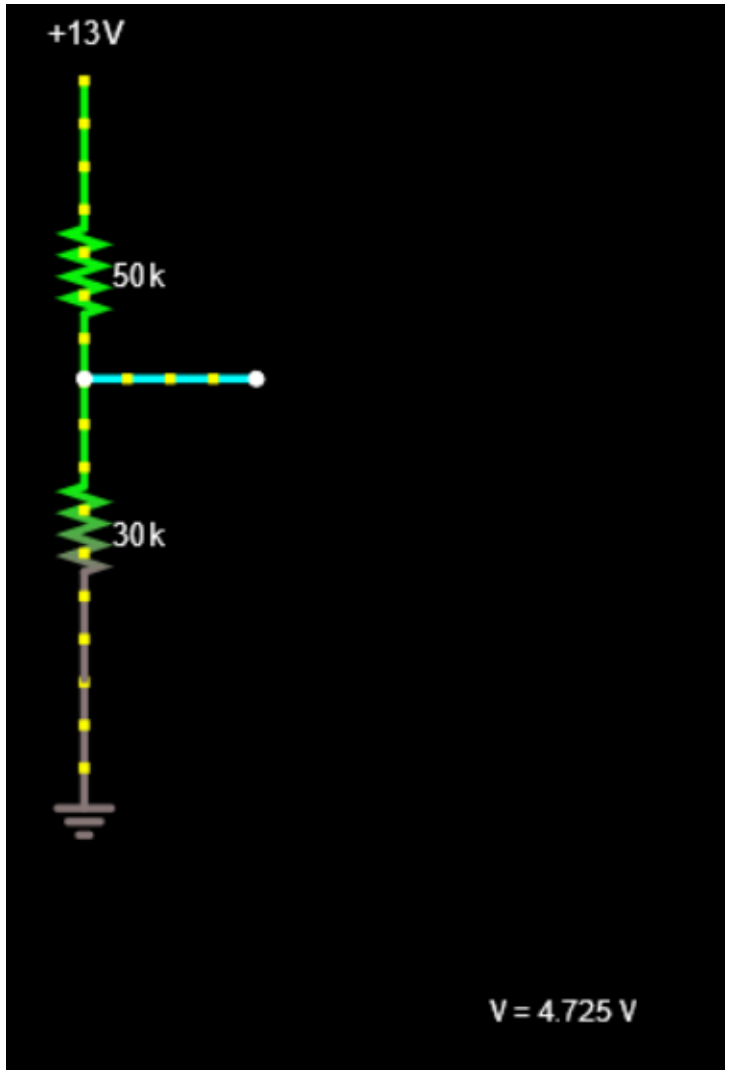
BATTERY FULL CHARGE IS 4.2V PER CELL

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

4S 16.8V



3S 12.6V



VOLTAGE READING

HOW MUCH POWER IN YOUR BATTERY

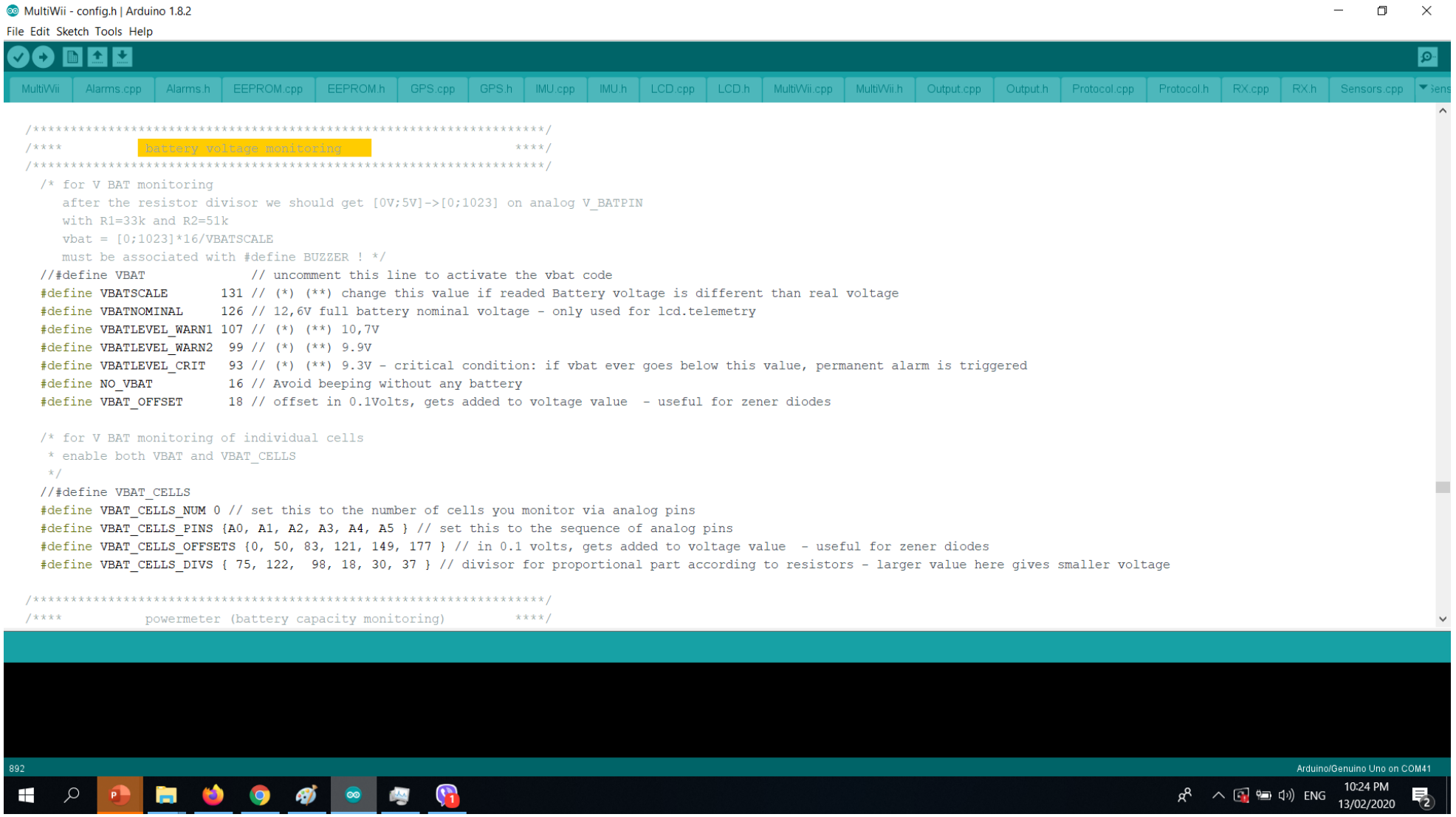
VOLTAGE DIVIDER

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

SWITCH THE VBAT JUMPER ACCORDINGLY TO THE BATTER CELLS YOUR USING

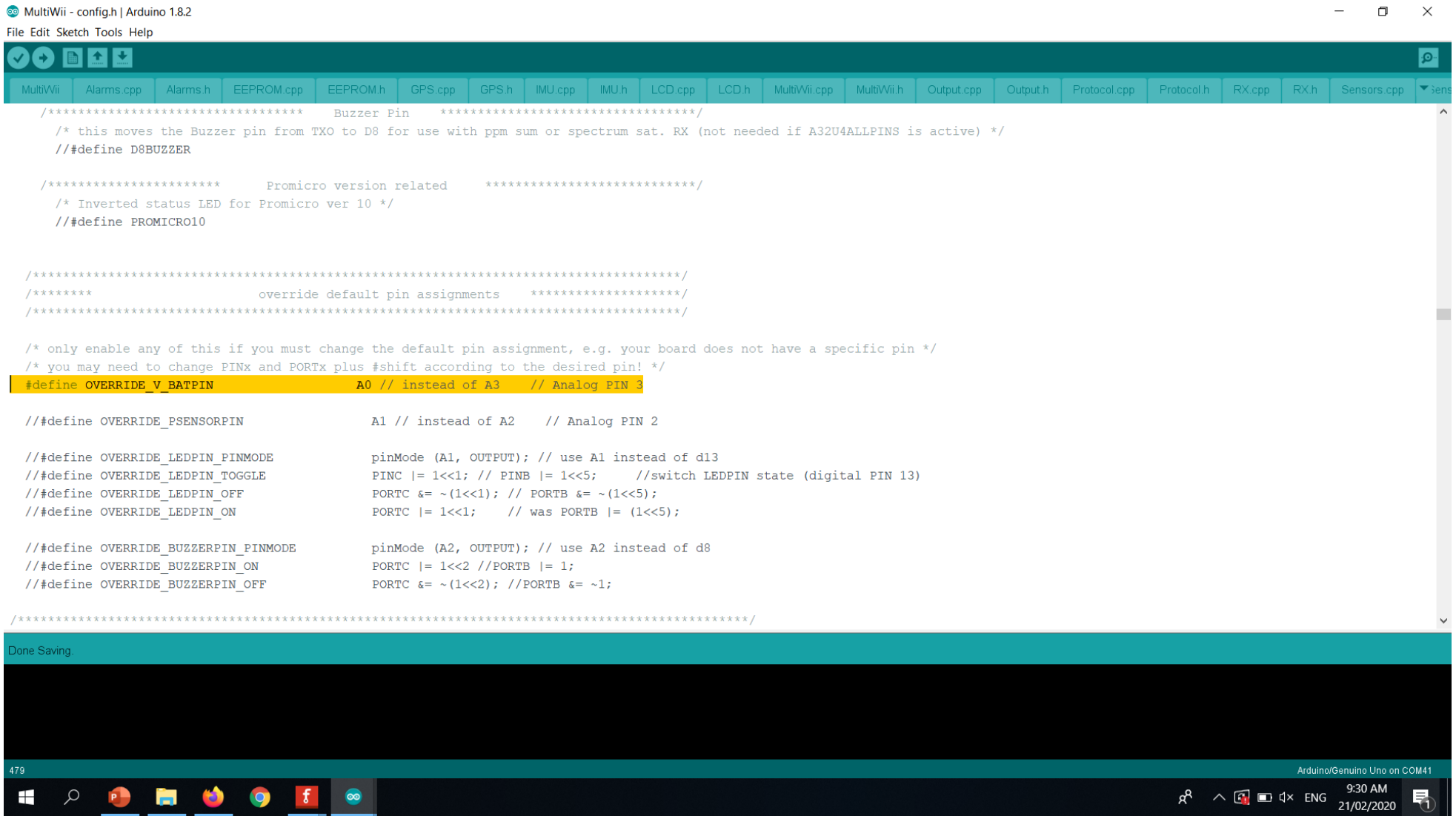
3S OR 4S

BATTERY MONITORING CONFIG.H



PIN ASSIGNMENTS

CONFIG.H



CONFIG.H

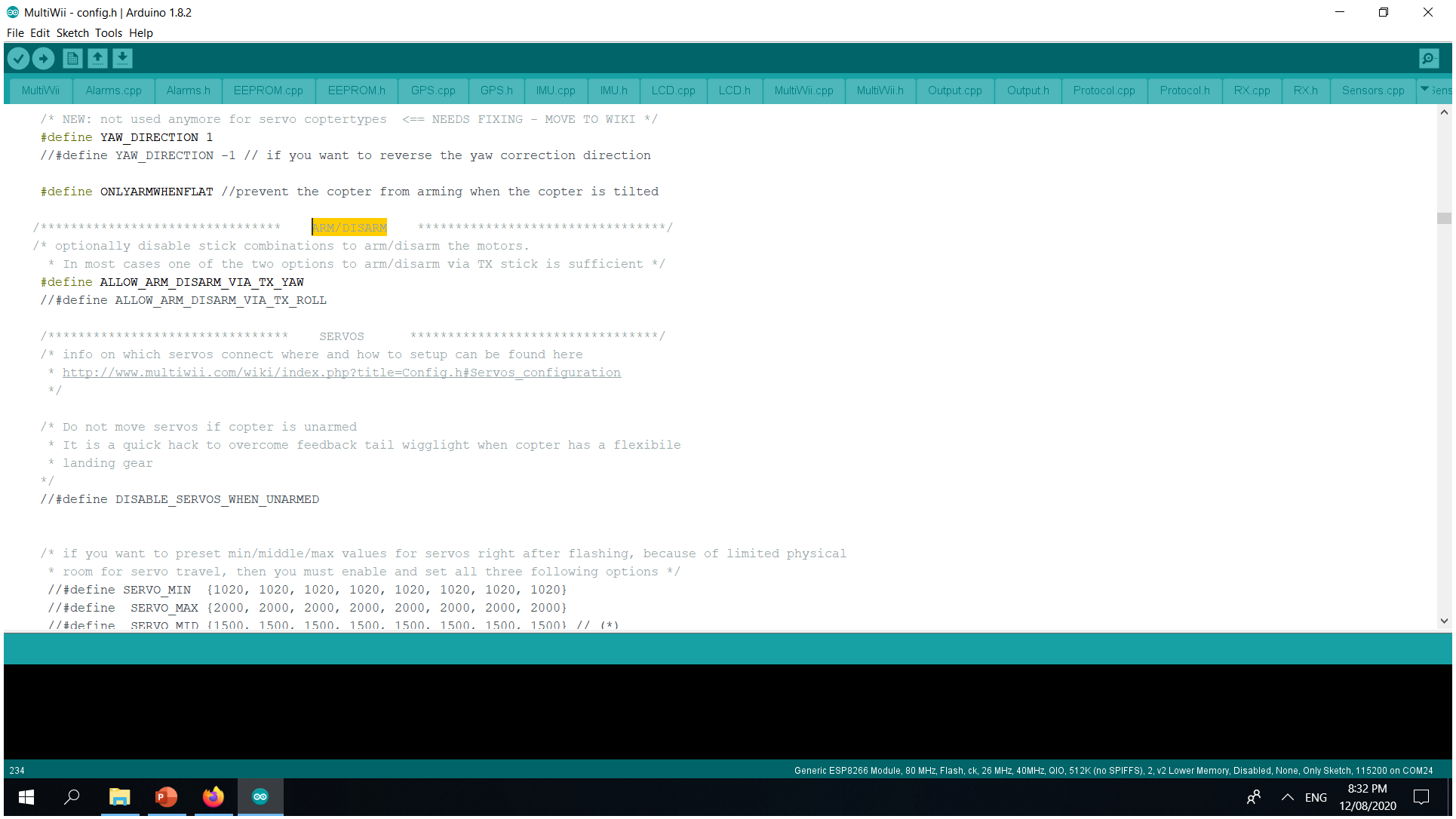
Arm/DisArm

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



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

CONFIG.H

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

FLYWIIGUI GROUND STATION

http://synerflight.com/flywiigui/

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

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

COMPASS (MAG AND GYRO)

PWM OUTPUT INDICATOR

PWM INPUT INDICATOR

SAVE CONFIG

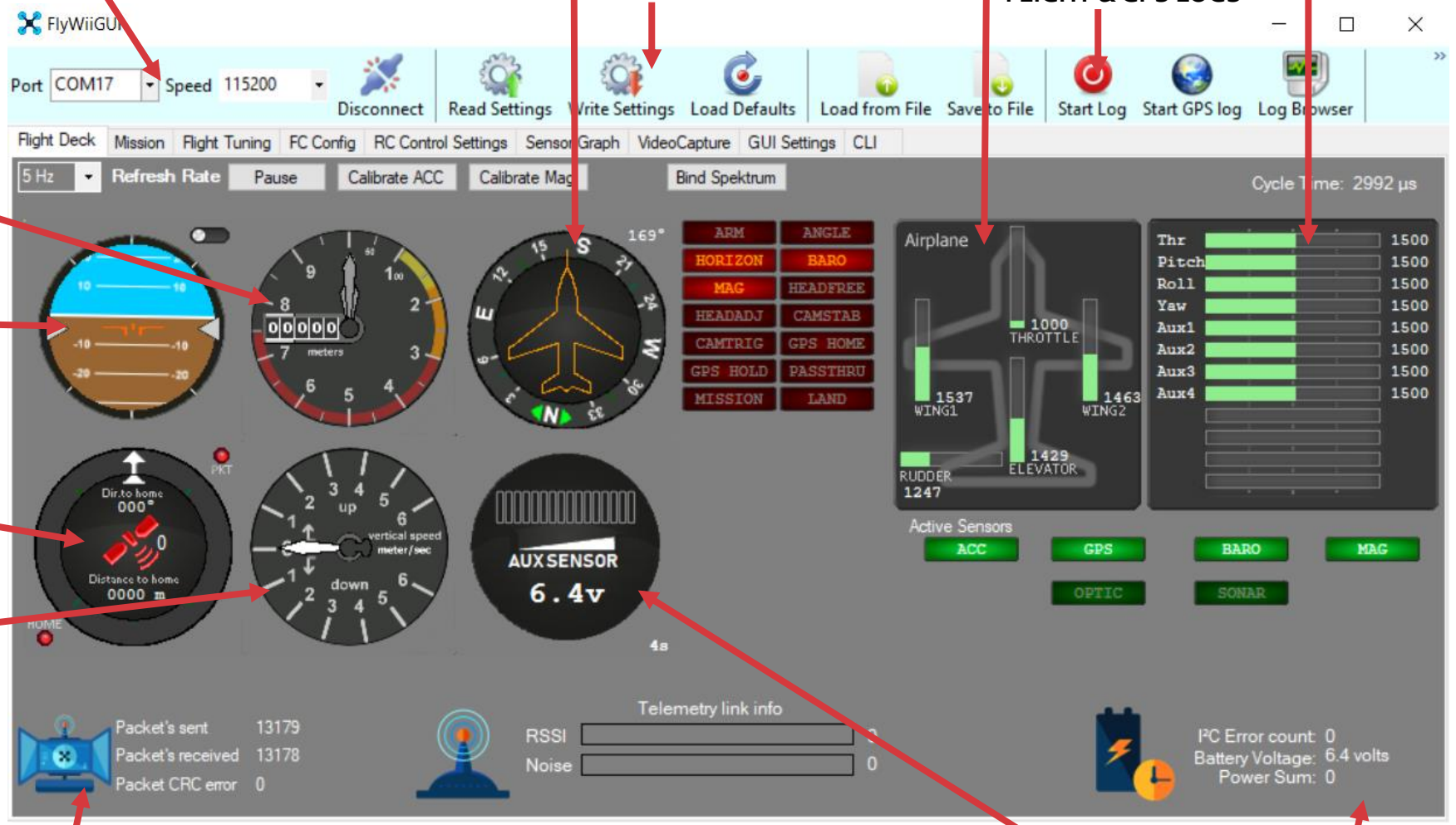
FLIGHT & GPS LOGS

ALTITUDE (BARO)

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

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

VERTICAL SPEED INDICATOR
(ACC Z AXIS)



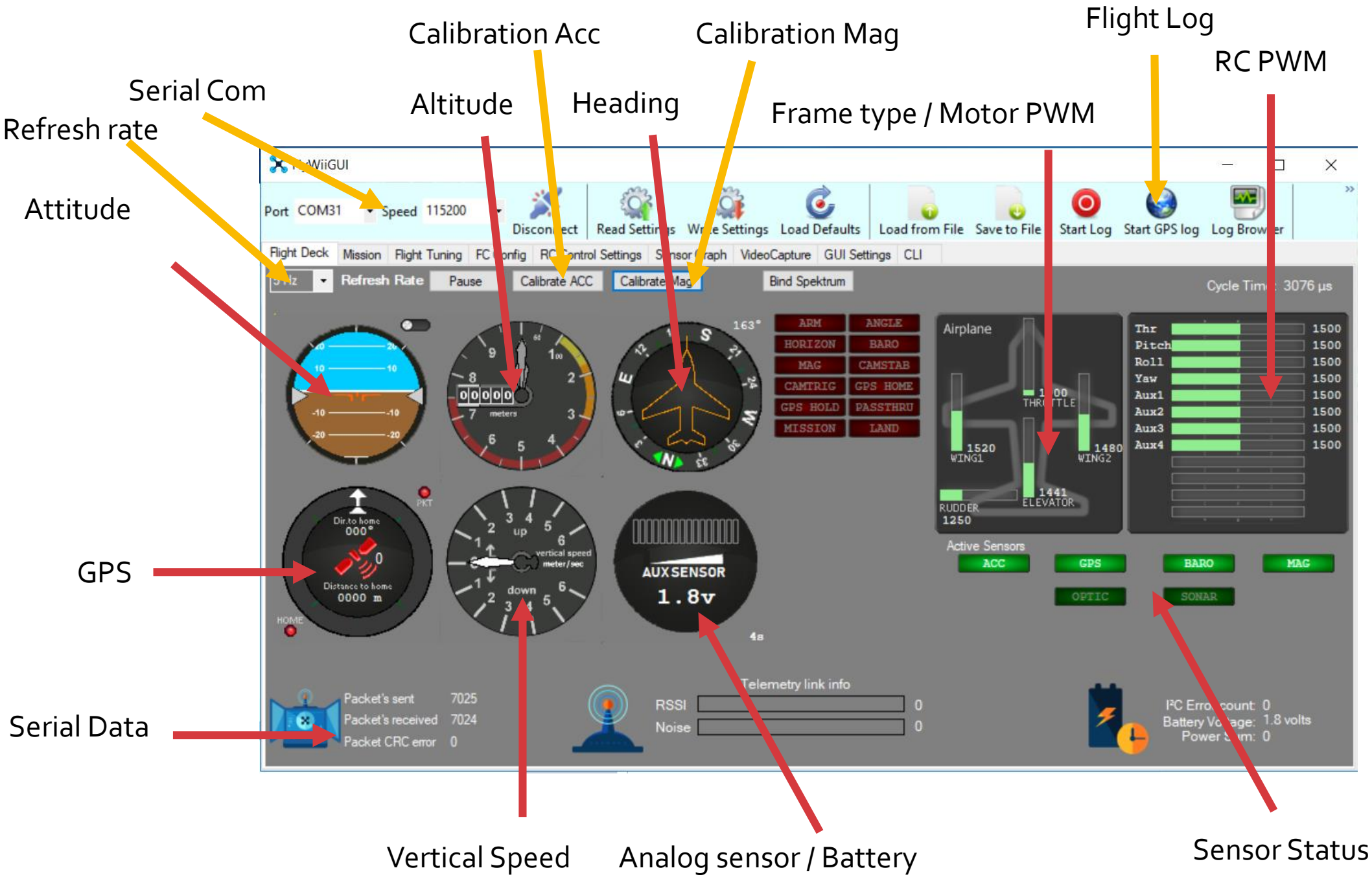
PACKETS STATUS

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

POWER STATUS / AUX SENSOR

ALSO KNOWN AS FUEL GAUGE

(VBAT)


Calibration Acc
Calibration Mag
Flight Log
RC PWM
Serial Com
Altitude
Heading
Frame type / Motor PWM
Refresh rate
Attitude
GPS
Serial Data
Vertical Speed
Analog sensor / Battery
Sensor Status
Port COM31
Speed 115200
Read Settings
Write Settings
Load Defaults
Load from File
Save to File
Start Log
Start GPS log
Flight Deck
Mission
Flight Tuning
VideoCapture
GUI Settings
CLI
Refresh Rate
Pause
Calibrate ACC
Calibrate Mag
Bind Spektrum
ARM
ANGLE
HORIZON
BARO
MAG
CAMSTAB
CAMTRIG
GPS HOME
GPS HOLD
PASSTHRU
MISSION
LAND
Airplane
Thr 1500
Pitch 1500
Roll 1500
Yaw 1500
Aux1 1500
Aux2 1500
Aux3 1500
Aux4 1500
1520
WING1
1480
WING2
1441
ELEVATOR
RUDDER
1250
Active Sensors
ACC
GPS
BARO
MAG
OPTIC
SONAR
AUX SENSOR
1.8v
Dir.to home
Distance to home
0000 m
vertical speed
meter/sec
Telemetry link info
RSSI
Noise
Packet's sent 7025
Packet's received 7024
Packet CRC error 0
Battery Voltage: 1.8 volts

Calibration Mag

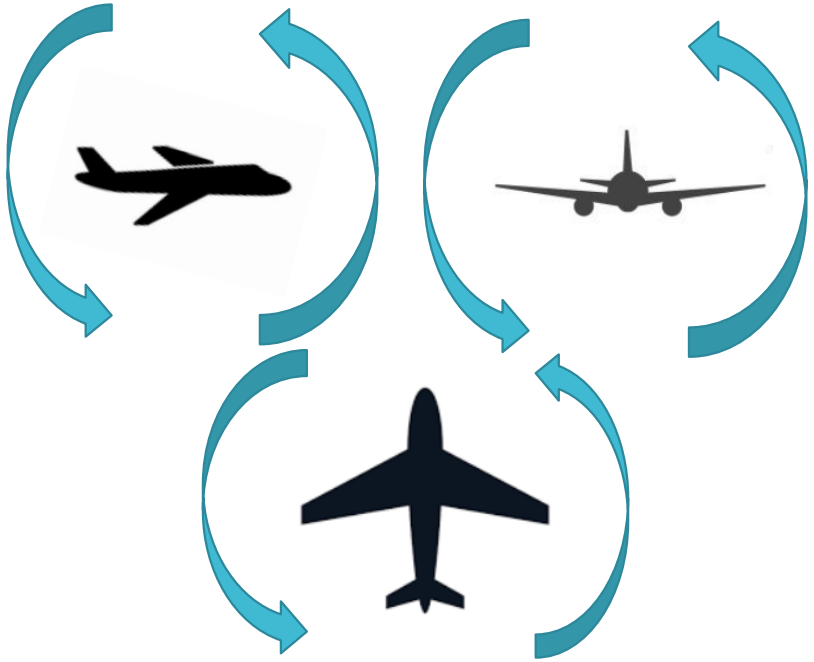
Refresh Rate . Telemetry update speed

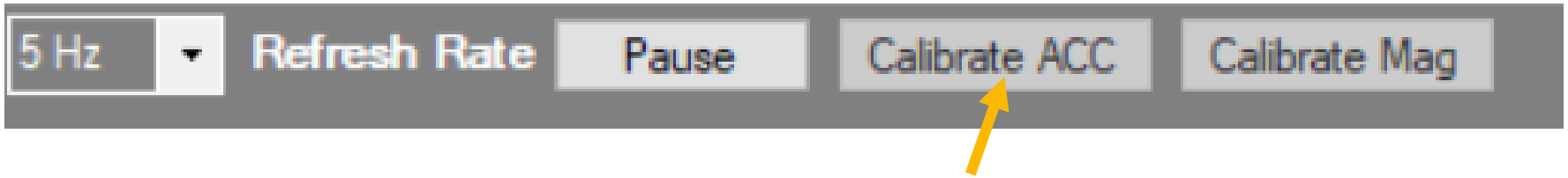
Acc Calibration . Set the drone down on a level surface . Away from any metal objects for 10 secs.

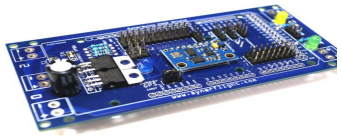
Mag Calibration . rotate the aircraft 360 degrees in all axis within 1 min. while the blue Led flashes

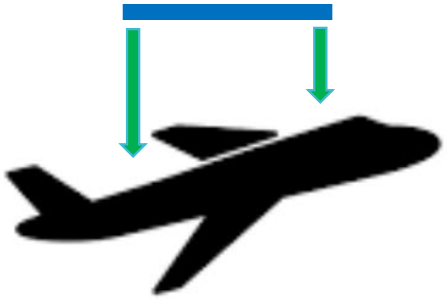
Mag Calibration must be perform when launching your drone in a new location for the first time. Pls verified the Compass if the drone heading matches your compass app in your phone.

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







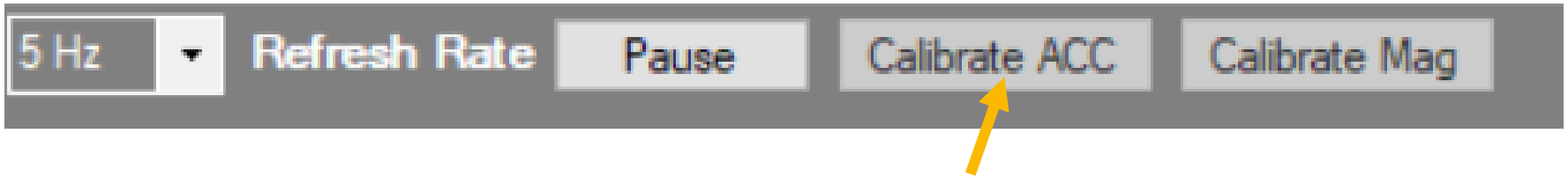


Flight controller Mounting in relation to Angle of Attack

In Neutral AOA – the Flight controller (Synerduino) can be set into a Neutral AOA to the airframe as being Level

This Hold true for any Fixwing that uses Flat plate airfoil wings to compensate for the lack of neutral lift unlike NACA or Cambered Airfoil

Calibration ACC



Due to the Nature of Flat wing Aircraft like the Synerduino Dart / FT Flyer an **Neutral Angle of Attack** must be identify and establish
The Synerduino Dart/FT flyer is set as 45 degrees AOA

Acc Calibration ,set the Aircraft on its neutral Angle of attack for 10 secs.

Max Speed AOA – this is when the aircraft is in its maximum Waypoint cruise speed as set by **Max Nav Banking** angle on Flight tuning tab Navigation settings

Neutral AOA – this is where the Aircraft AOA is settle on Horizon mode

Stall AOA – this is when the aircraft is slowing down also set by **Max Nav Banking** angle on **Flight tuning tab Navigation settings** . Happends when the mission is set to hold the aircraft tries to slow down or when on landing

This is for surface control Augmentation

FC Config

Gyro or Acc assisted Mode.

Check if Gyro move servos in right directions.

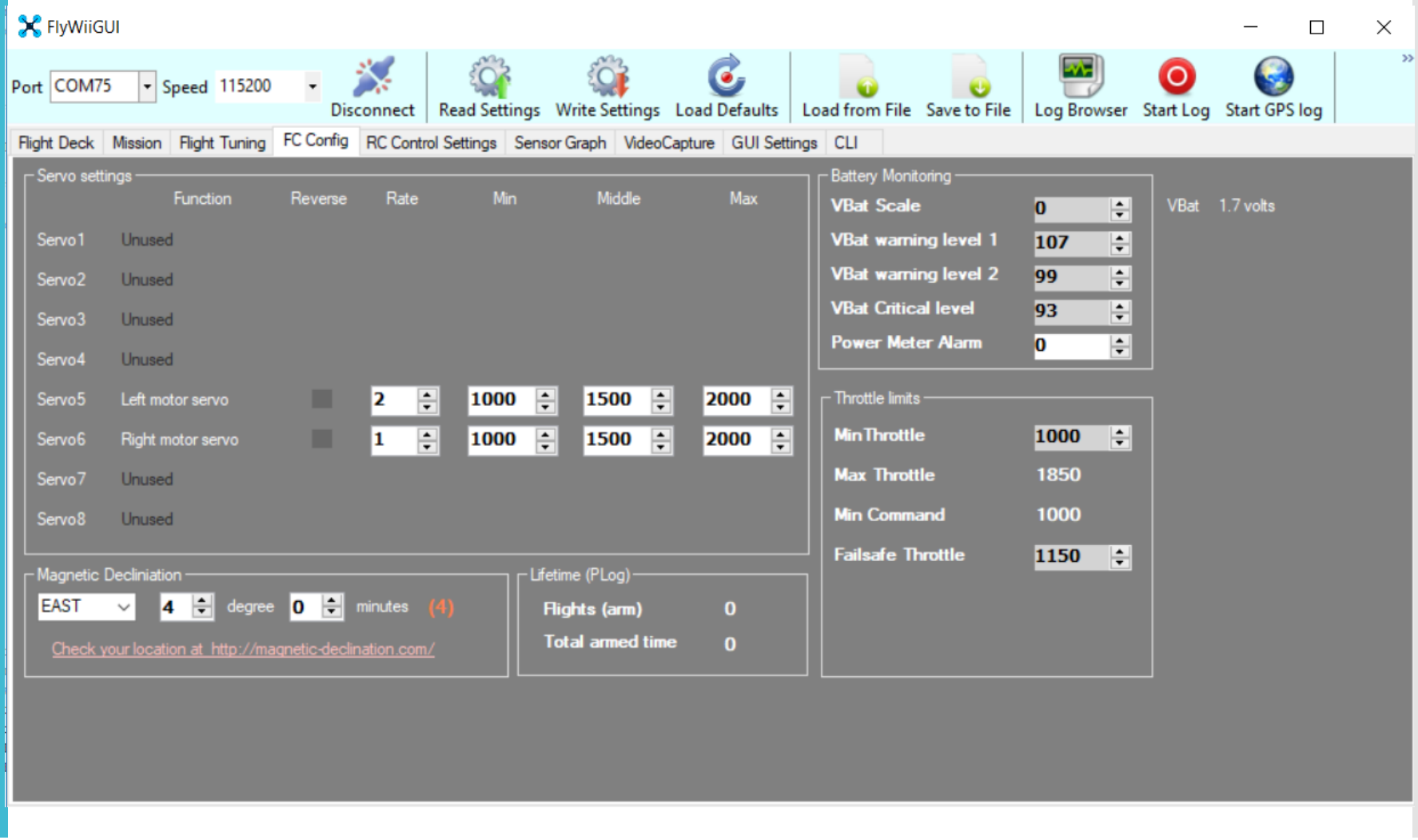
Lift a wingtip and Aileron goes up.

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.

Rate 1-norm or 2-rev



This is for surface control Augmentation

Servos Settings

Gyro or Acc assisted Mode.

Check if Gyro move servos in right directions.

Lift a Right wingtip and Right Aileron goes up.

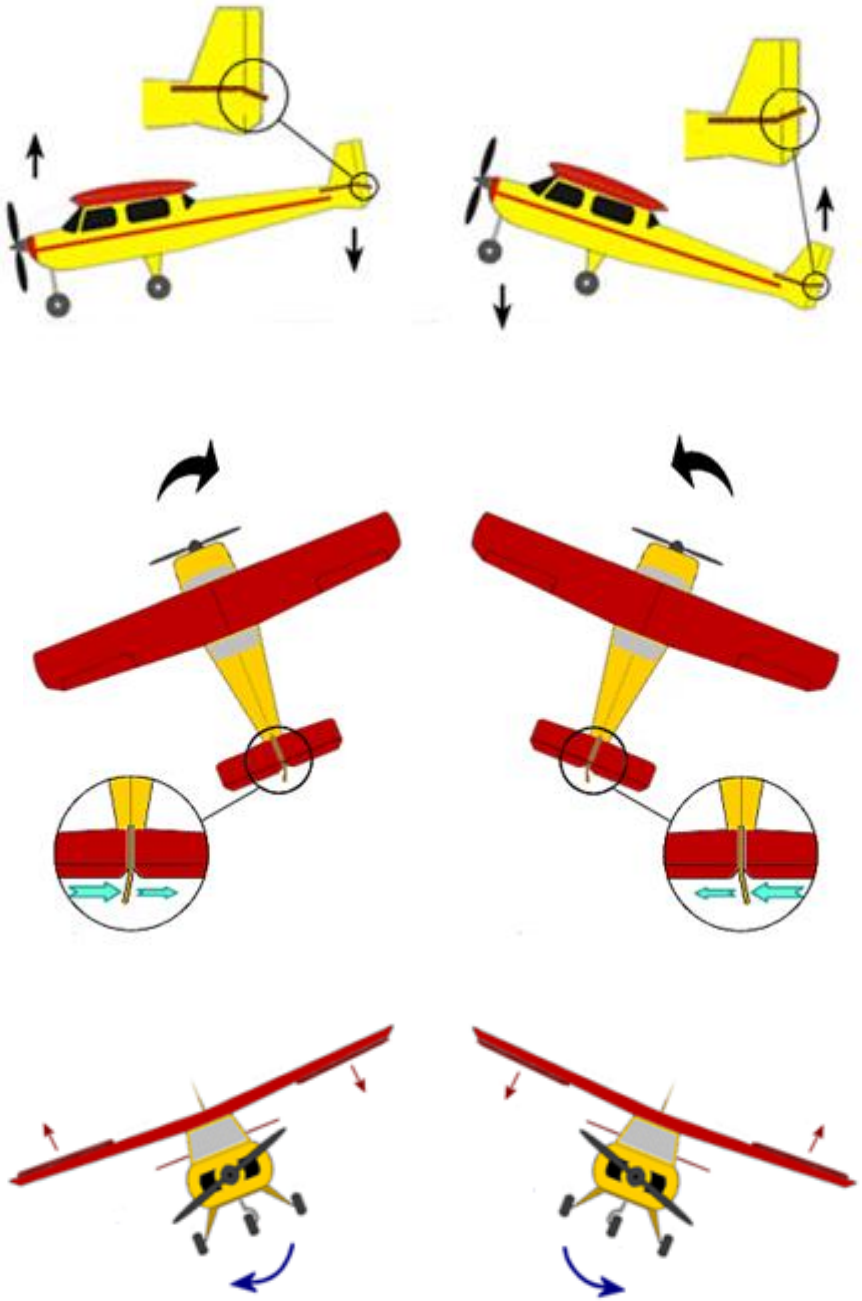
Lift the tail and Elevator goes up.

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

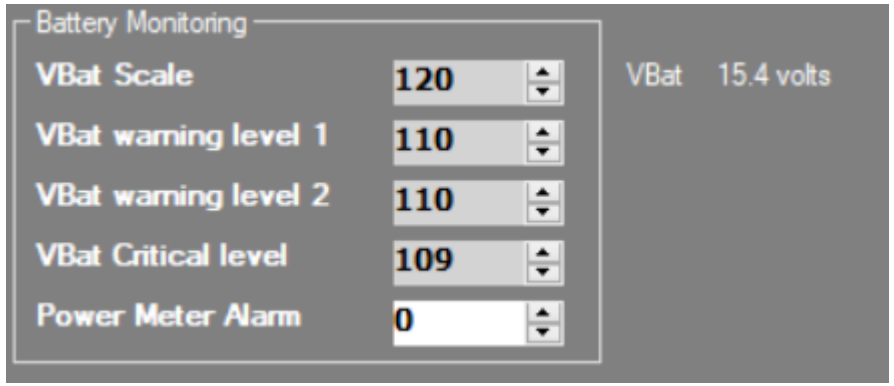
Control surfaces should oppose the disturbance applied to the aircraft

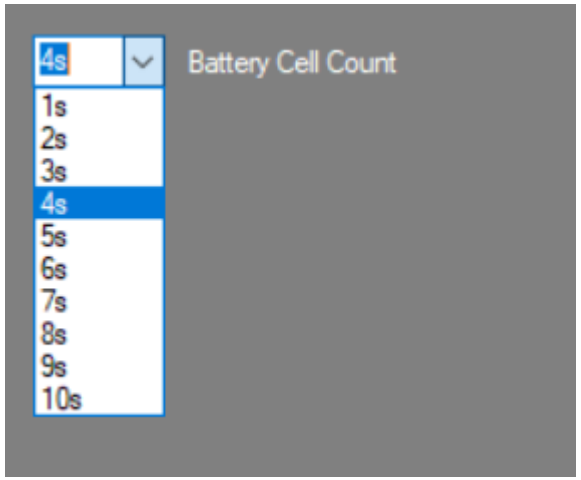


The arrow represent an external force acting on the aircraft opposite to it the control surface counter it with the opposite force

The Gyro and Accelerometer would always counter the force opposite to it .

FLYWII GUI





(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

The Diff thrust plane of Synerduino is rather unique

It behaves like a Bi-copter as also a plane

ARM — to activate the Aircraft

ANGLE - this is mostly use to keep the plane Level

HORIZON - this also may work for certain Leveling application

BARO — Altitude hold by throttle control

MAG — Heading hold

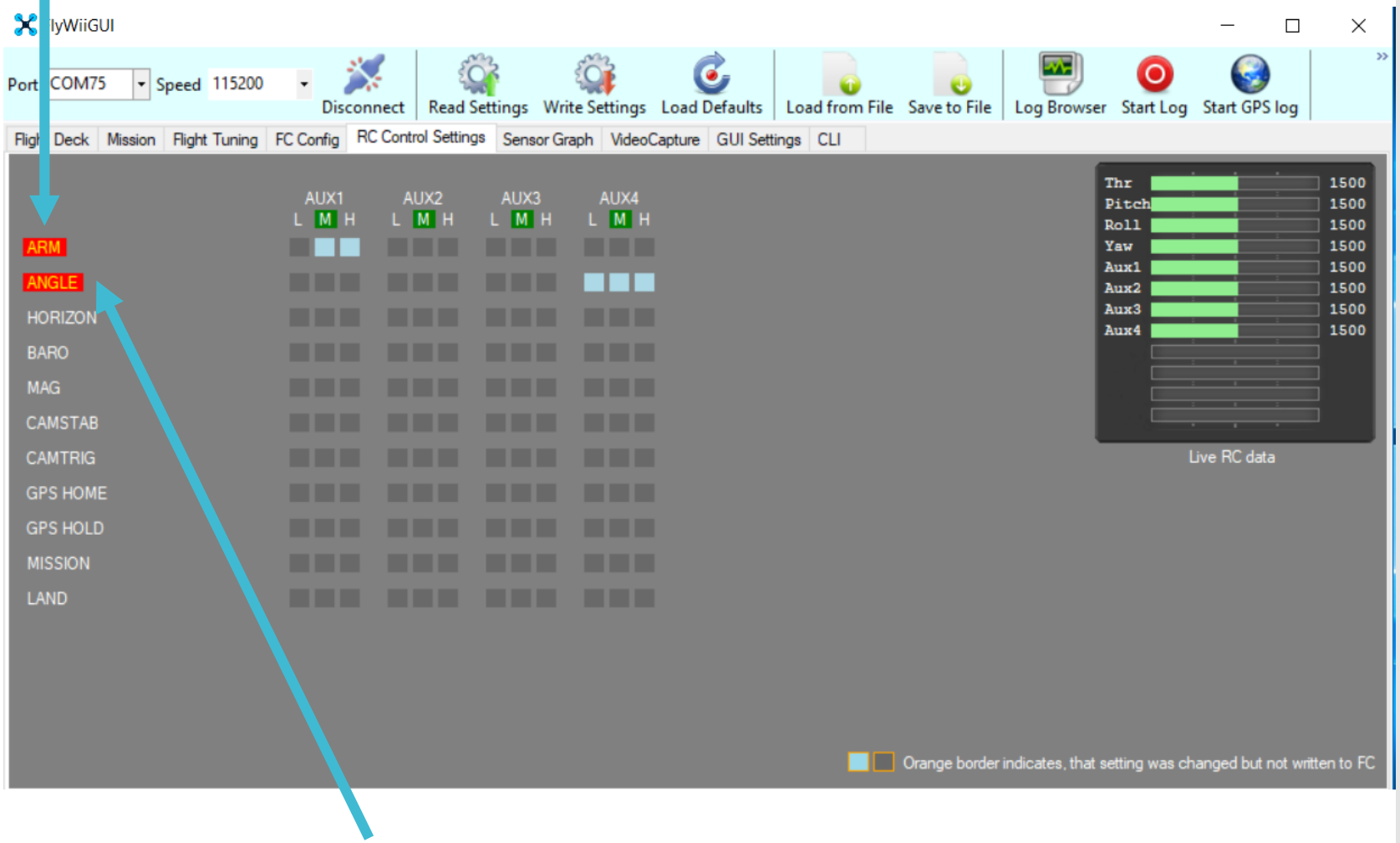
CAM TRIG — Payload Trigger

GPS HOME — RTH

MISSION — Perform Waypoint

LAND — this allows the Aircaft to Land after RTH , must use in conjunction to GPS Home

ARM – Arms the main motor of the aircraft



Use Angle mode for stabilization

- **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

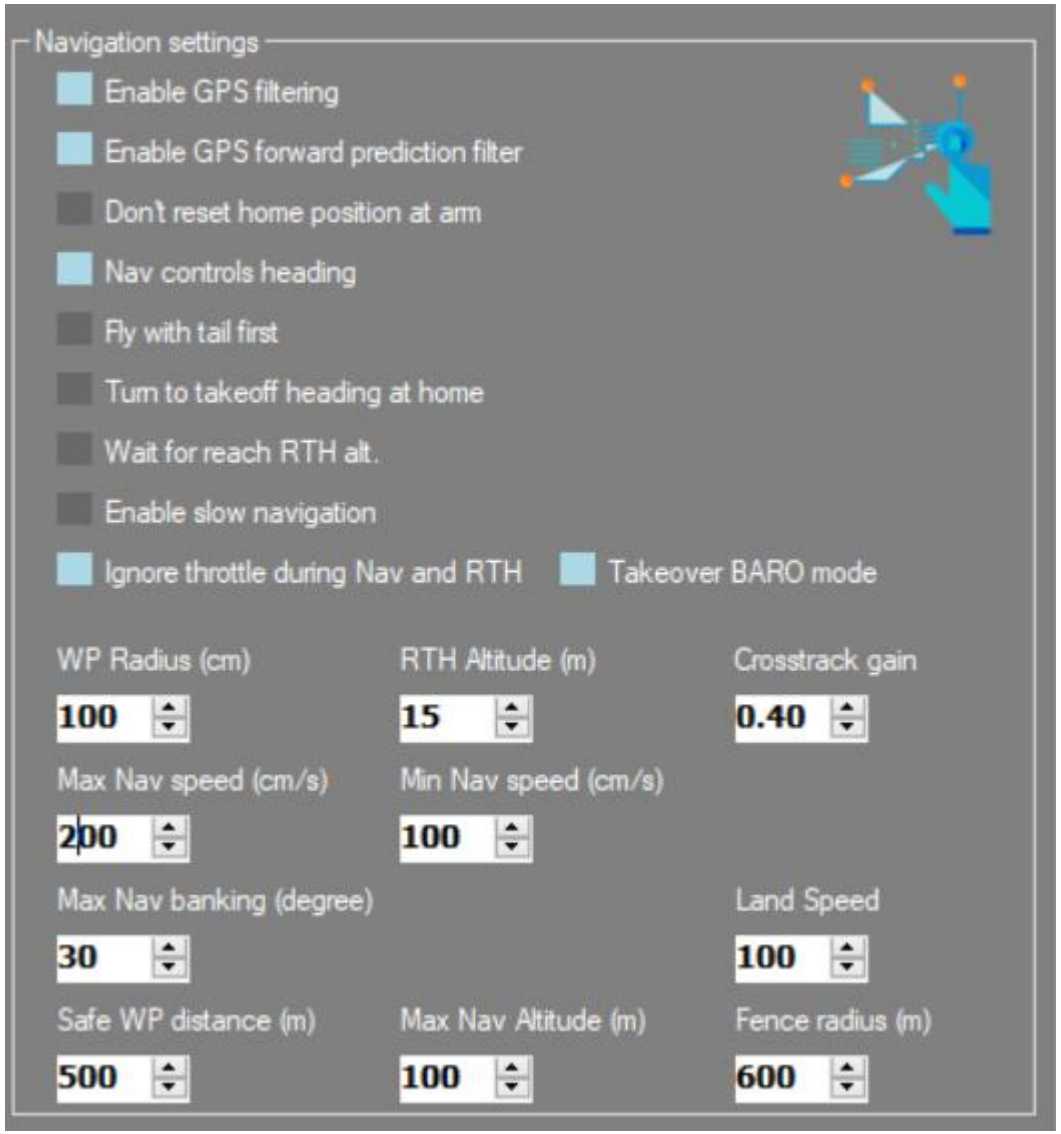
(o degrees = North)

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 with trigger it has reach the waypoint

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

Min Nav Speed – the speed the drone travel when with in the WP Radius 200cm/s for starters

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

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

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

Land Speed – speed of descending for Landing cm/s

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

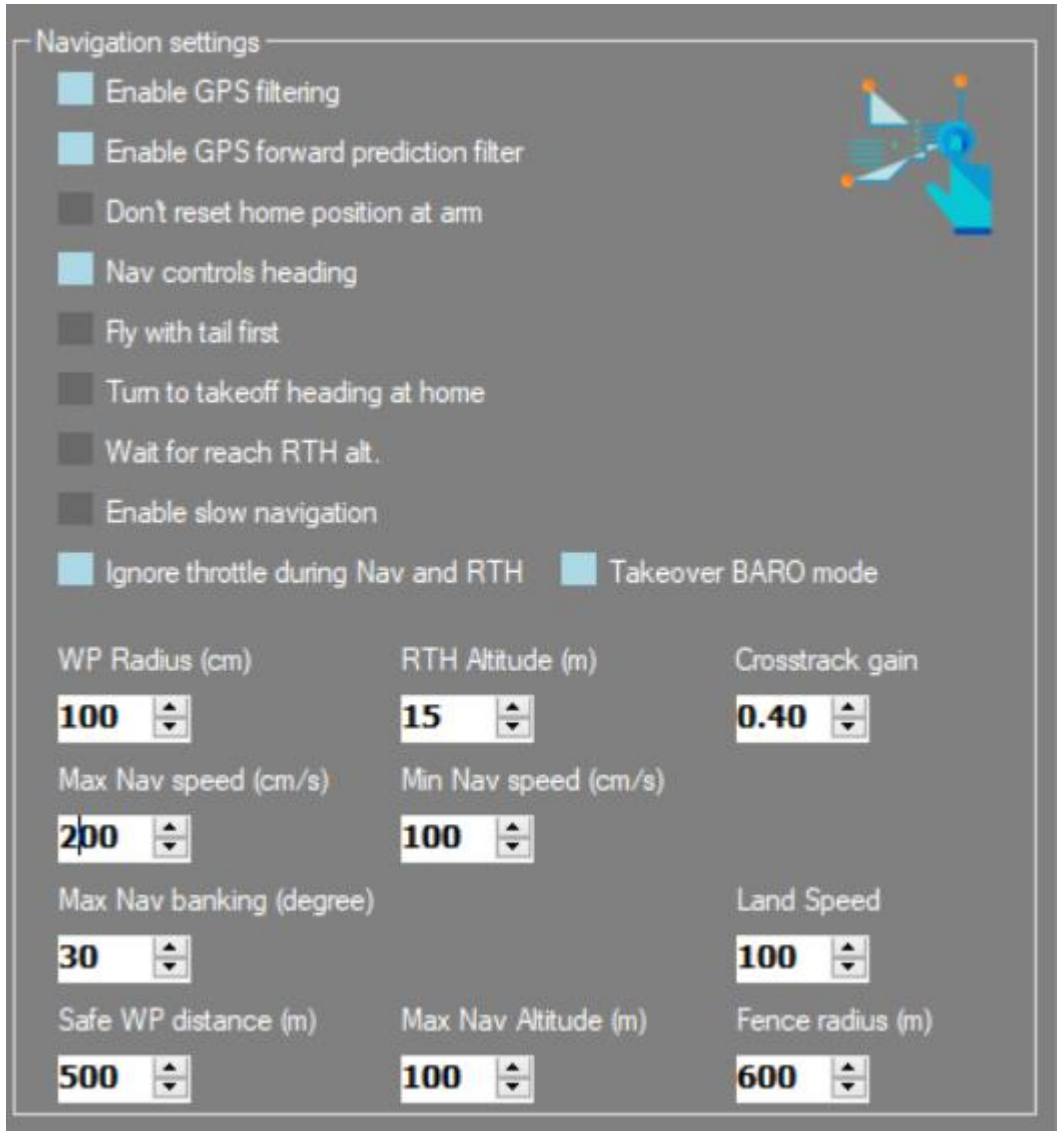
Fence Radius – Geo Fence to keep the drone with in the perimeter in relation to home position

CrossTrack gain - this tune the GPS and Nav sensitivity

GPS Filtering – use to enhance GPS accuracy

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

Other Navigation Functions



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 orientates to its heading right after arming

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

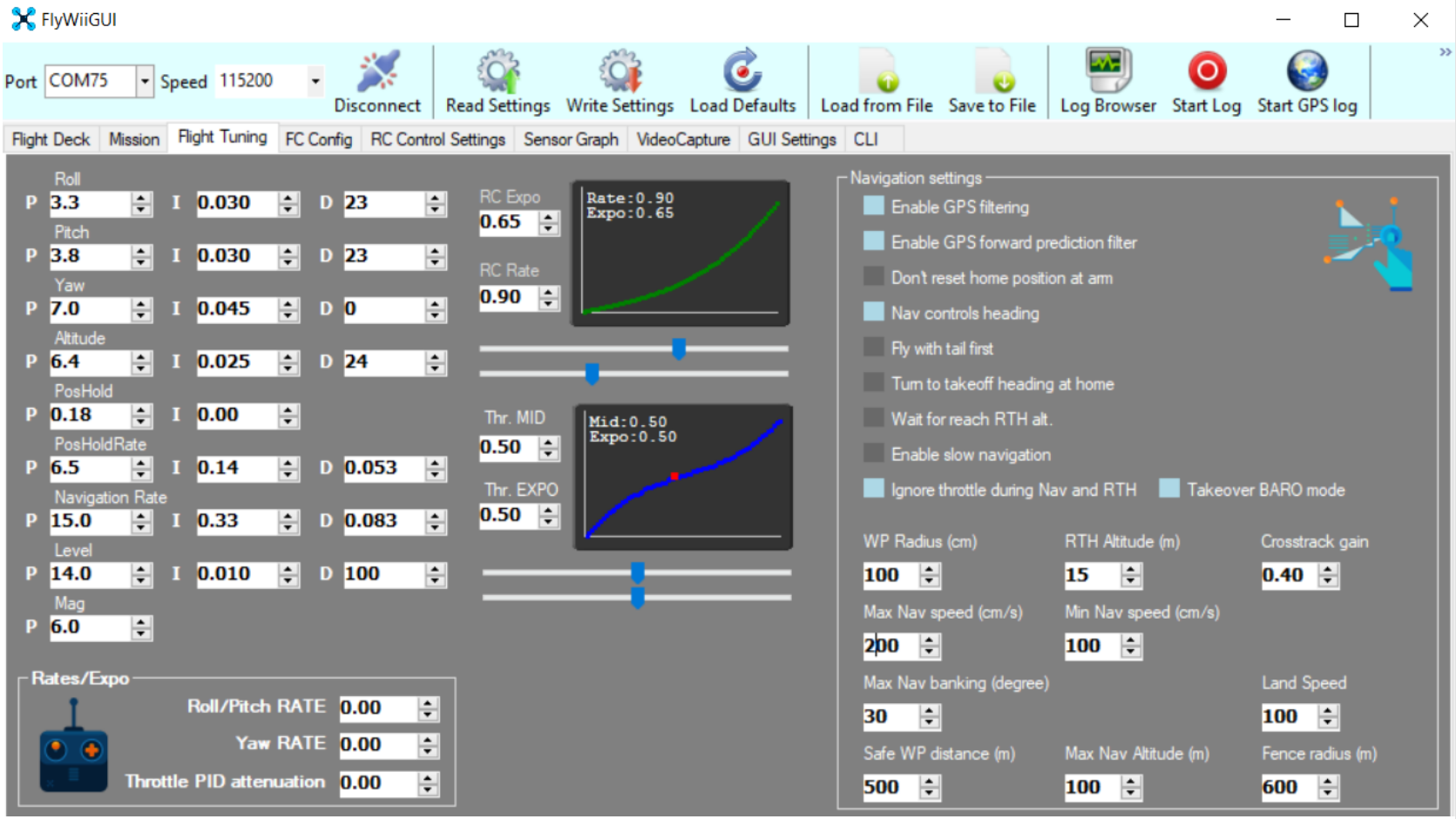
Enable slow navigation – this works with keeping the drone to its **Min Nav speed**

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

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



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 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

Aerobatic flight: Requires a slightly higher P

Gentle smooth flight: Requires a slightly lower P

I – Integral

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

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

A higher I will increase the heading hold capability

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

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

D- Divide / Derivative

This moderates the speed at which the Aircraft is returned to its original position.

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

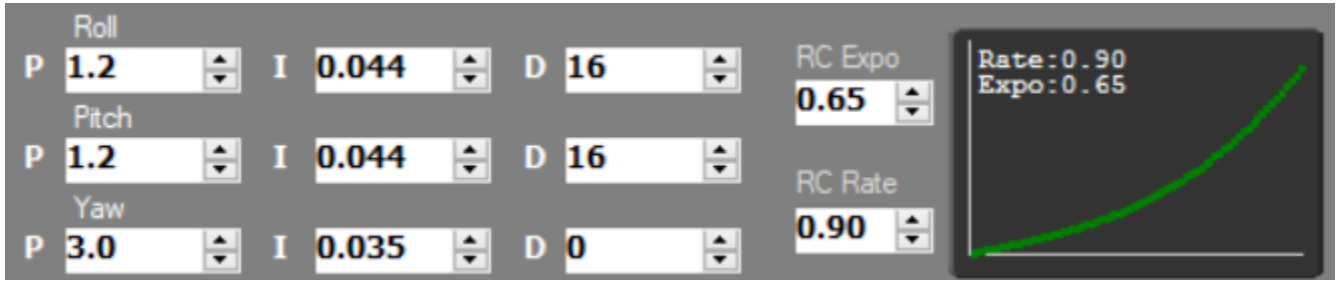
Increasing value for D: Dampens changes. Slower to react to fast changes

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



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

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.

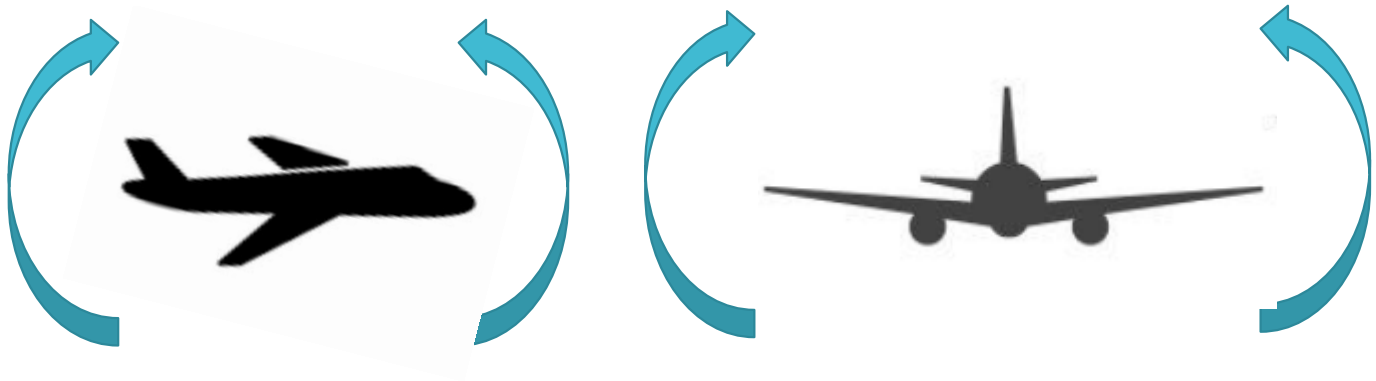
7 - Repeat for Yaw Axis. Your settings should now be suitable for flight tuning.

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 of slightly

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

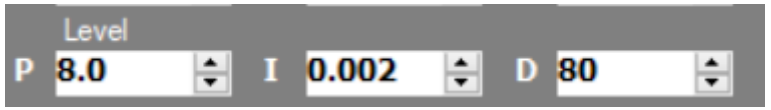
Decrease value for D until recovery from dramatic control changes results in unacceptable recovery oscillations, then increase D slightly Repeat above steps

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

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

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

PID : Level



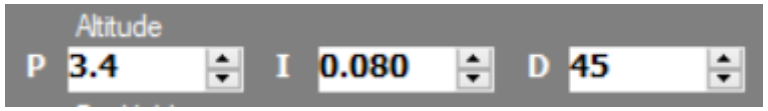
This will influence the flight characteristic with an accelerometer : this is Level Mode

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

Increase value for P will make the autolevel mode stronger

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

ALT Hold



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.

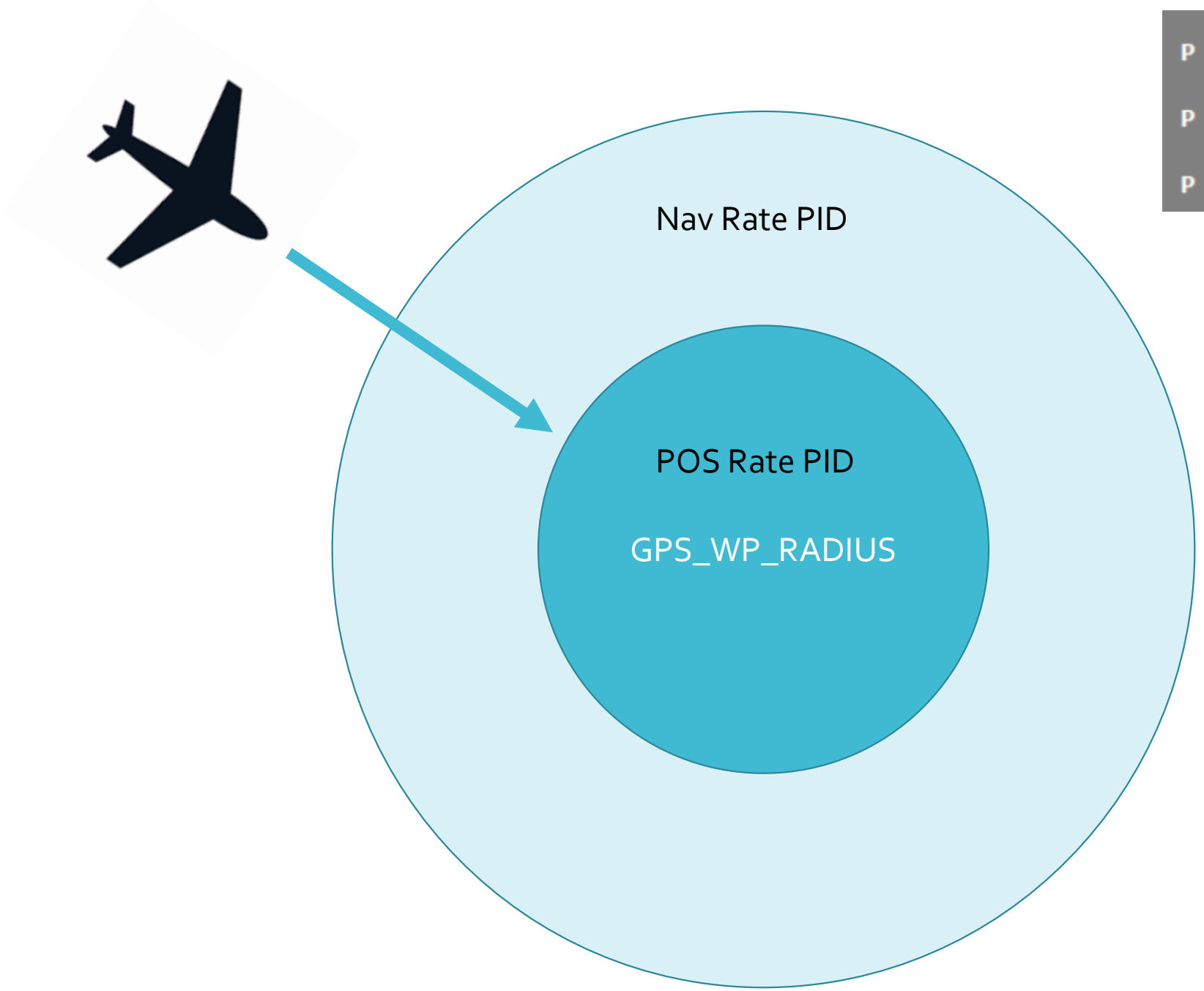
- To tune the Pos Rate PID, initially set P, I and D values to 0.
- Gradually increase P until the Aircraft begins to position hold with some drift.
- Gradually increase D until the Aircraft responds more quickly to undesired changes in attitude caused by the wind. If this value is set too high you will 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

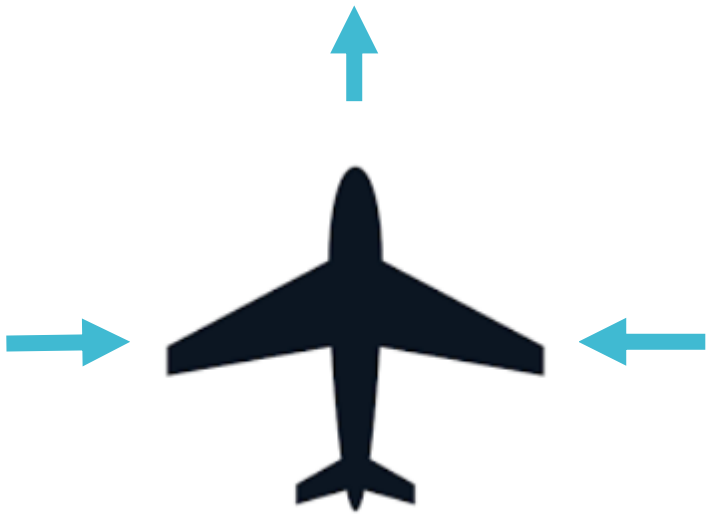
Enter loiter orbit mode

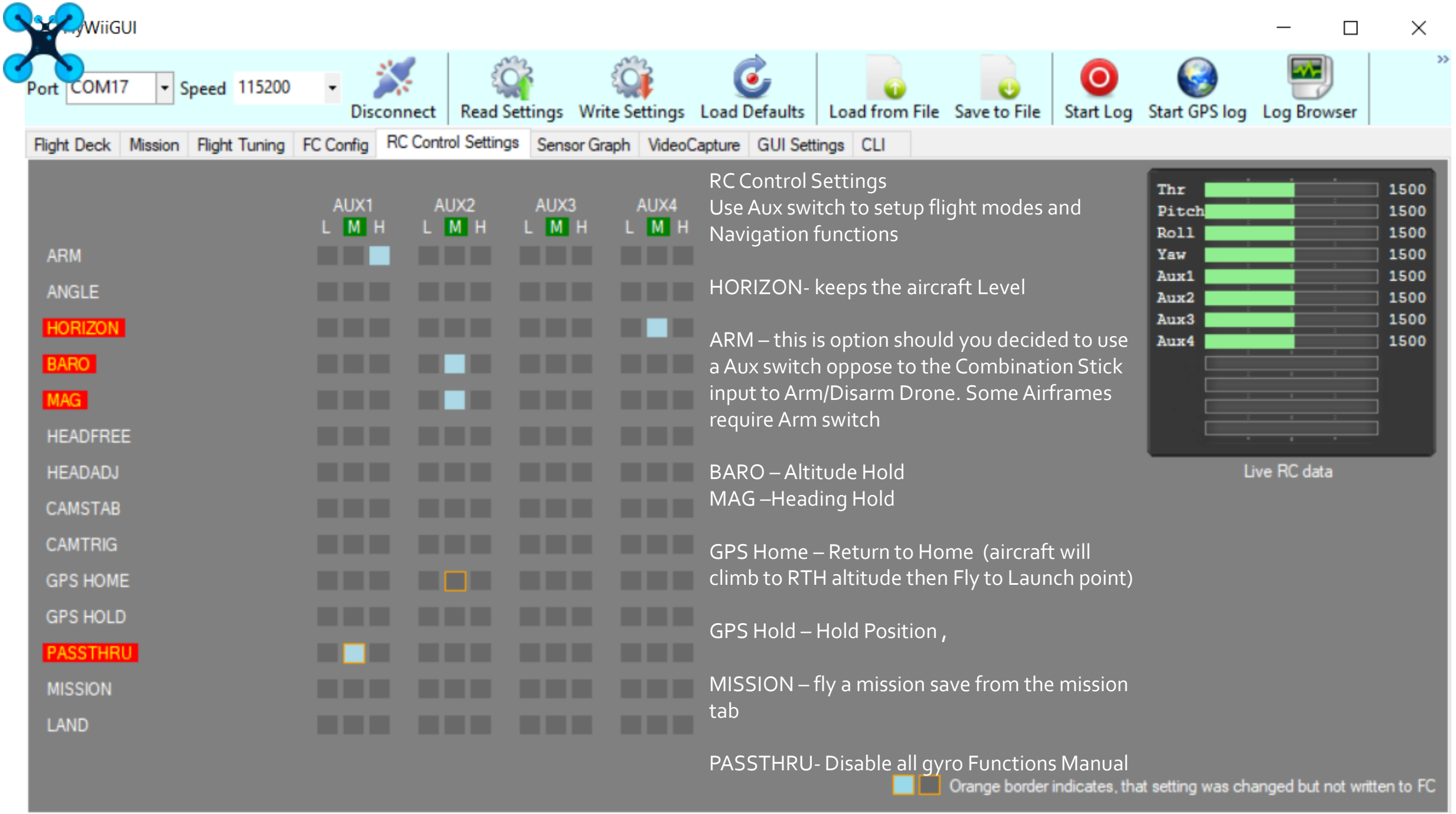
To tune the Pos Rate PID, initially set P, I and D values to 0.

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
Disconnect
Read Settings
Write Settings
Load Defaults
Load from File
Save to File
Start Log
Start GPS log
Log Browser
Flight Deck
Mission
Flight Tuning
FC Config
RC Control Settings
Sensor Graph
VideoCapture
GUI Settings
CLI
AUX1
AUX2
AUX3
AUX4
L M H
ARM
ANGLE
HORIZON
BARO
MAG
HEADFREE
HEADADJ
CAMSTAB
CAMTRIG
GPS HOME
GPS HOLD
PASSTHRU
MISSION
LAND
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
Thr 1500
Pitch 1500
Roll 1500
Yaw 1500
Aux1 1500
Aux2 1500
Aux3 1500
Aux4 1500
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

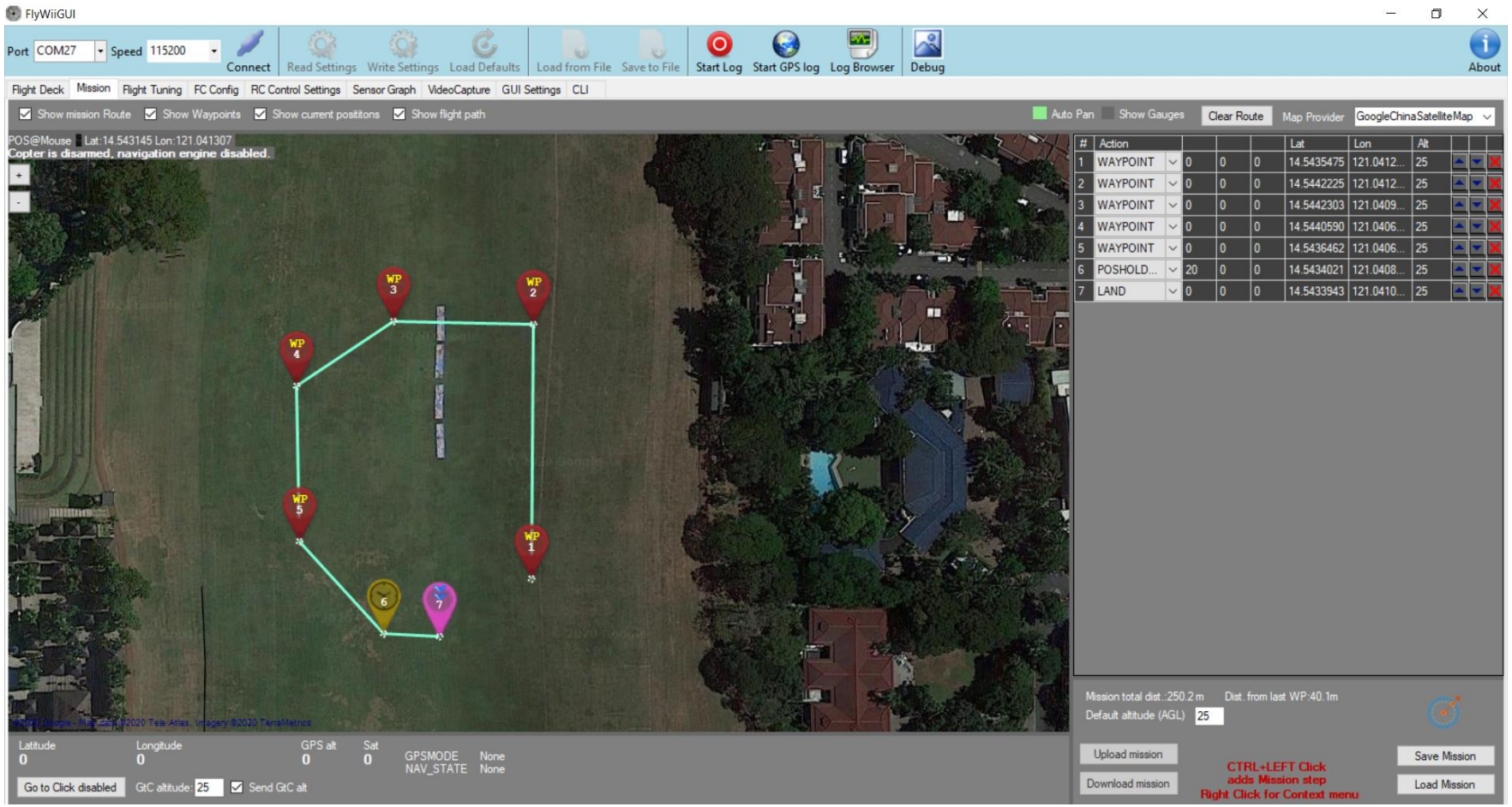
Time PosHold – Drone will wait X number of 00:00:00 then move to the next waypoint (limited for fixwing)

Unlimited PosHold – once the drone reach this point it will hover and wait till you switch out of Mission mode (not for fixwing)

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

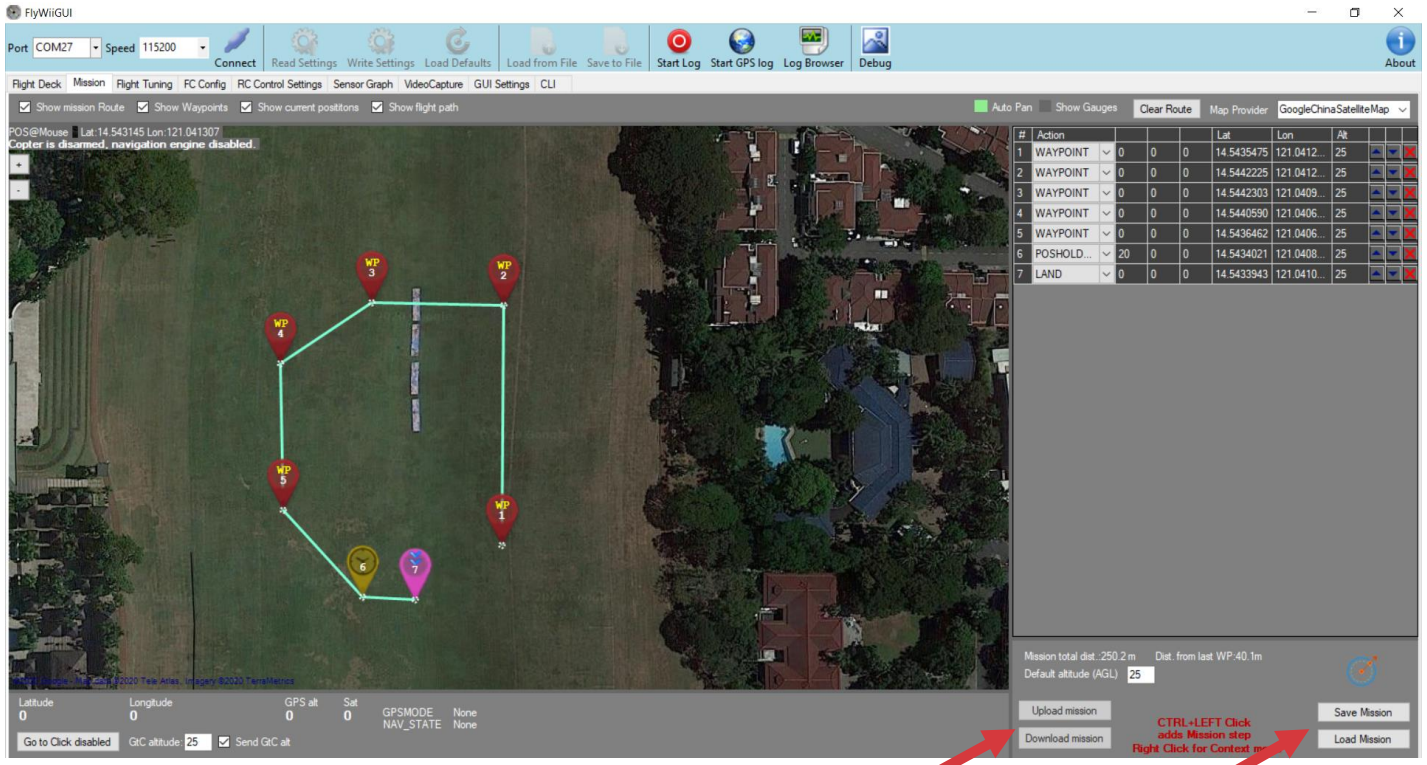
Missions

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 of 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.

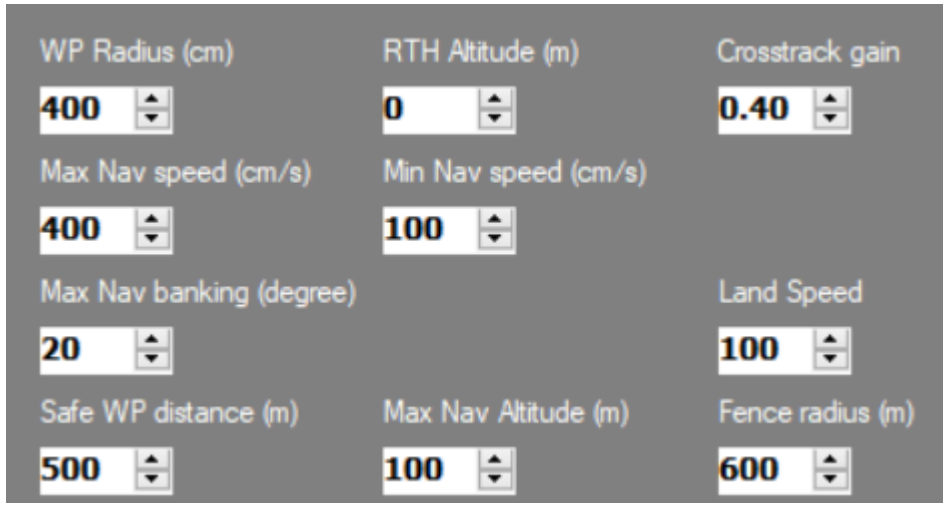
Only use Angle/Horizon + Gps Home/Hold together.
Do **NOT** Activate BARO Or MAG for navigation.
It will interfere with the navigation code.

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.

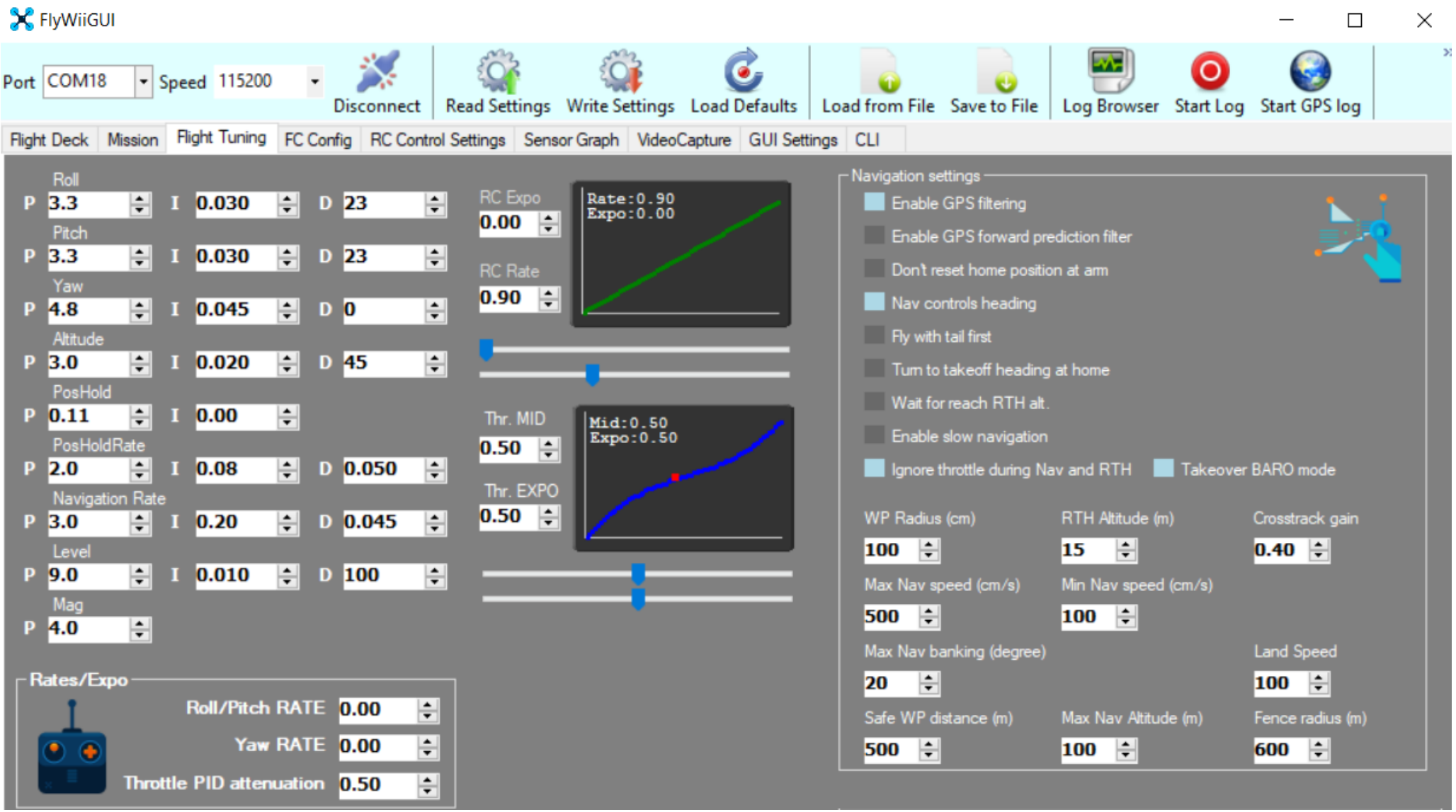
If Failsafe is active at return The plane will Disarm motor and descend to a "Landing"

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.

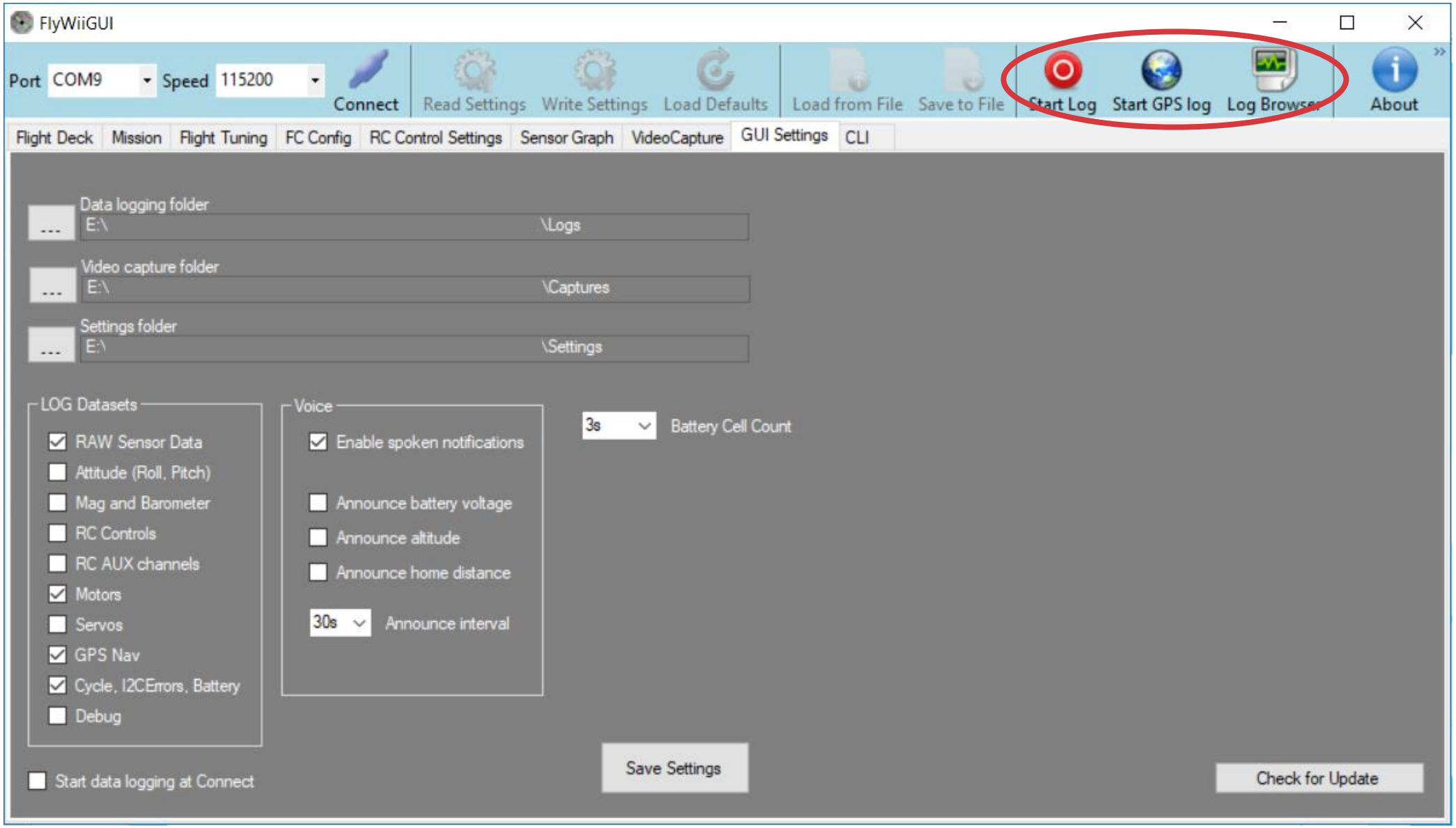


PID setup and config

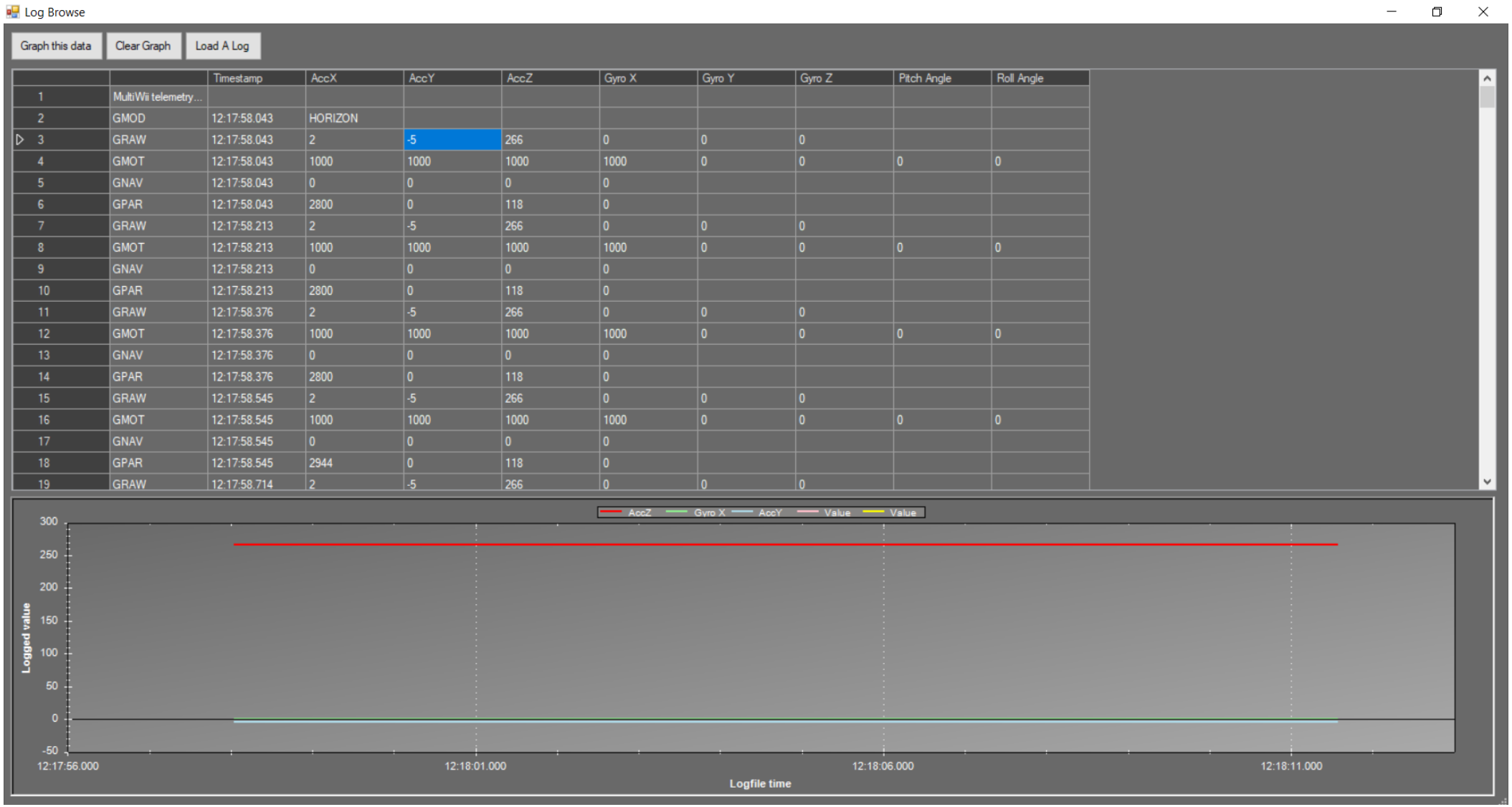


Graphs and Data Logging

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



Log Browser



		Timestamp	AccX	AccY	AccZ	Gyro X	Gyro Y	Gyro Z	Pitch Angle	Roll Angle
1	MultiWii telemetry...									
2	GMOD	12:17:58.043	HORIZON							
3	GRAW	12:17:58.043	2	-5	266	0	0	0		
4	GMOT	12:17:58.043	1000	1000	1000	1000	0	0	0	0
5	GNAV	12:17:58.043	0	0	0	0				
6	GPAR	12:17:58.043	2800	0	118	0				
7	GRAW	12:17:58.213	2	-5	266	0	0	0		
8	GMOT	12:17:58.213	1000	1000	1000	1000	0	0	0	0
9	GNAV	12:17:58.213	0	0	0	0				
10	GPAR	12:17:58.213	2800	0	118	0				
11	GRAW	12:17:58.376	2	-5	266	0	0	0		
12	GMOT	12:17:58.376	1000	1000	1000	1000	0	0	0	0
13	GNAV	12:17:58.376	0	0	0	0				
14	GPAR	12:17:58.376	2800	0	118	0				
15	GRAW	12:17:58.545	2	-5	266	0	0	0		
16	GMOT	12:17:58.545	1000	1000	1000	1000	0	0	0	0
17	GNAV	12:17:58.545	0	0	0	0				
18	GPAR	12:17:58.545	2944	0	118	0				
19	GRAW	12:17:58.714	2	-5	266	0	0	0		

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.

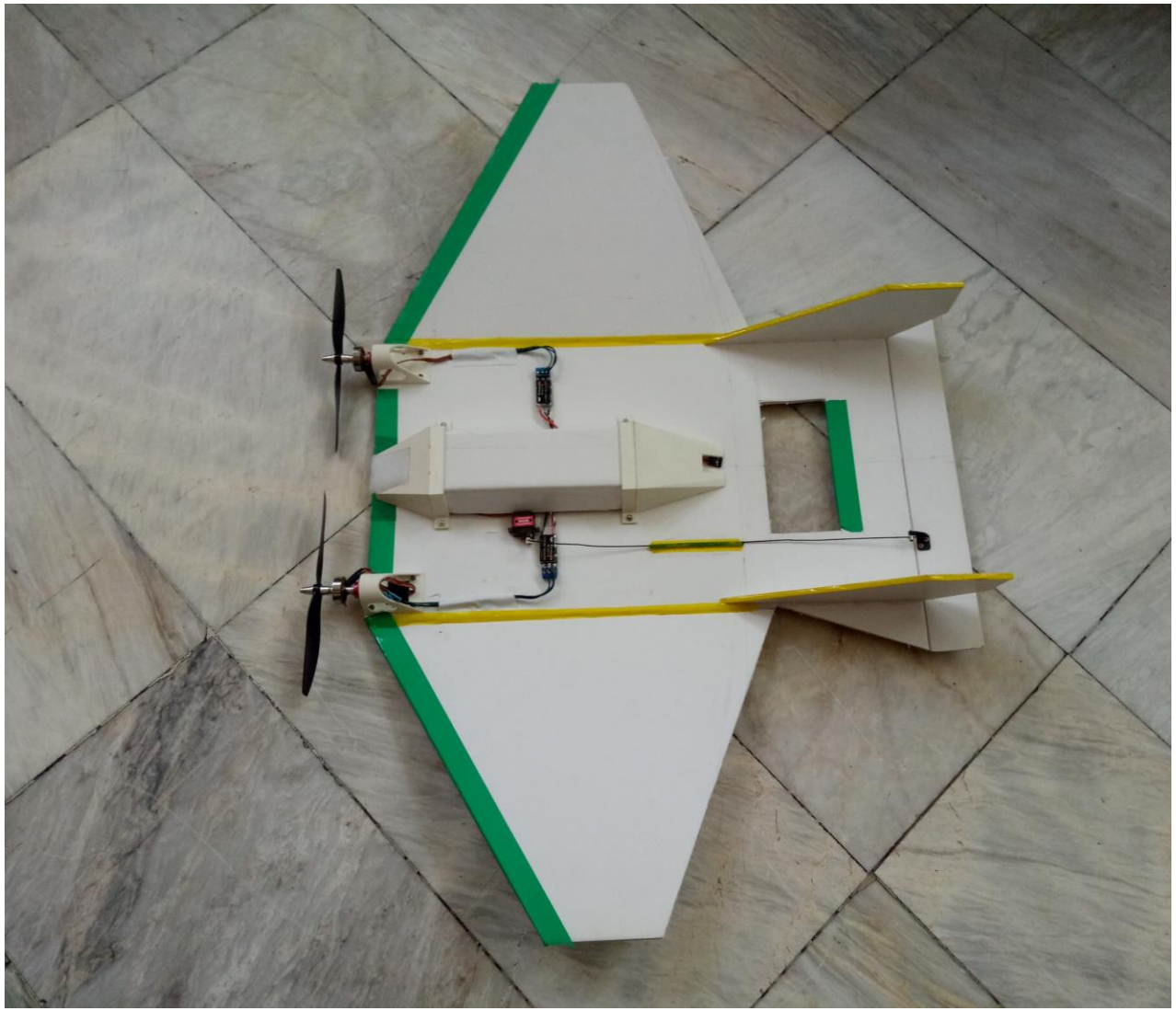
In RC control Setting set Aux 1 as Arm

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

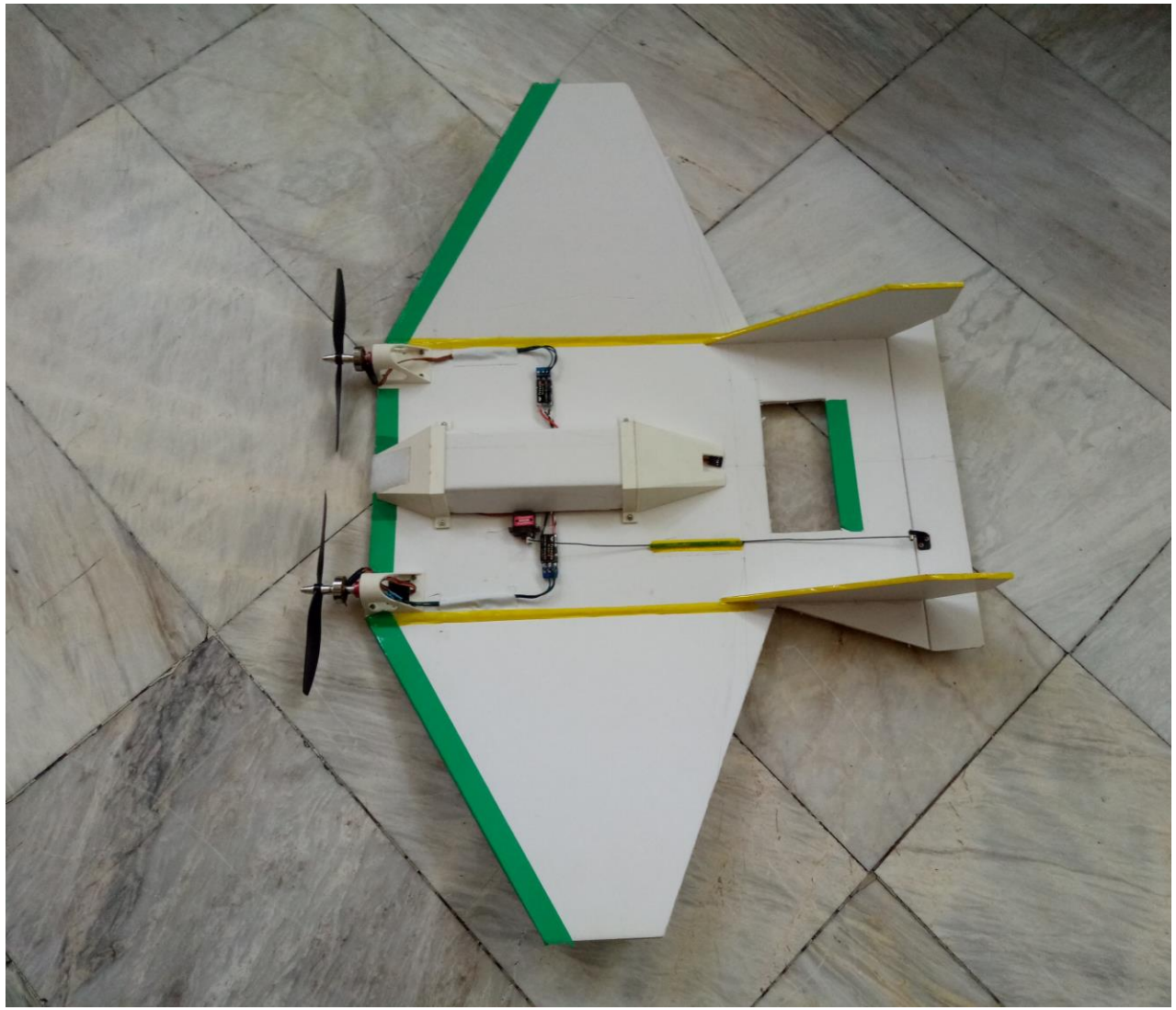
when on GPS Home , GPS Hold , Mission Flight modes & when USB is plugged in . (pls use Bluetooth telemetry)

Tests motor with Props off

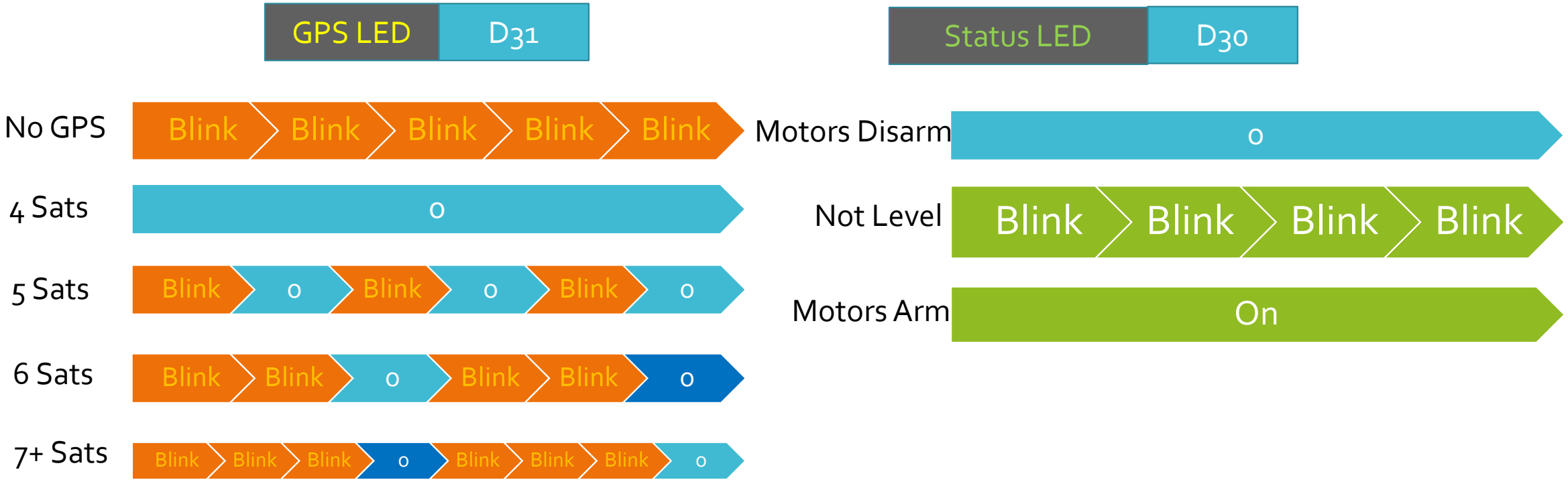
Baro and Mag preferably switch off when Arming

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 +