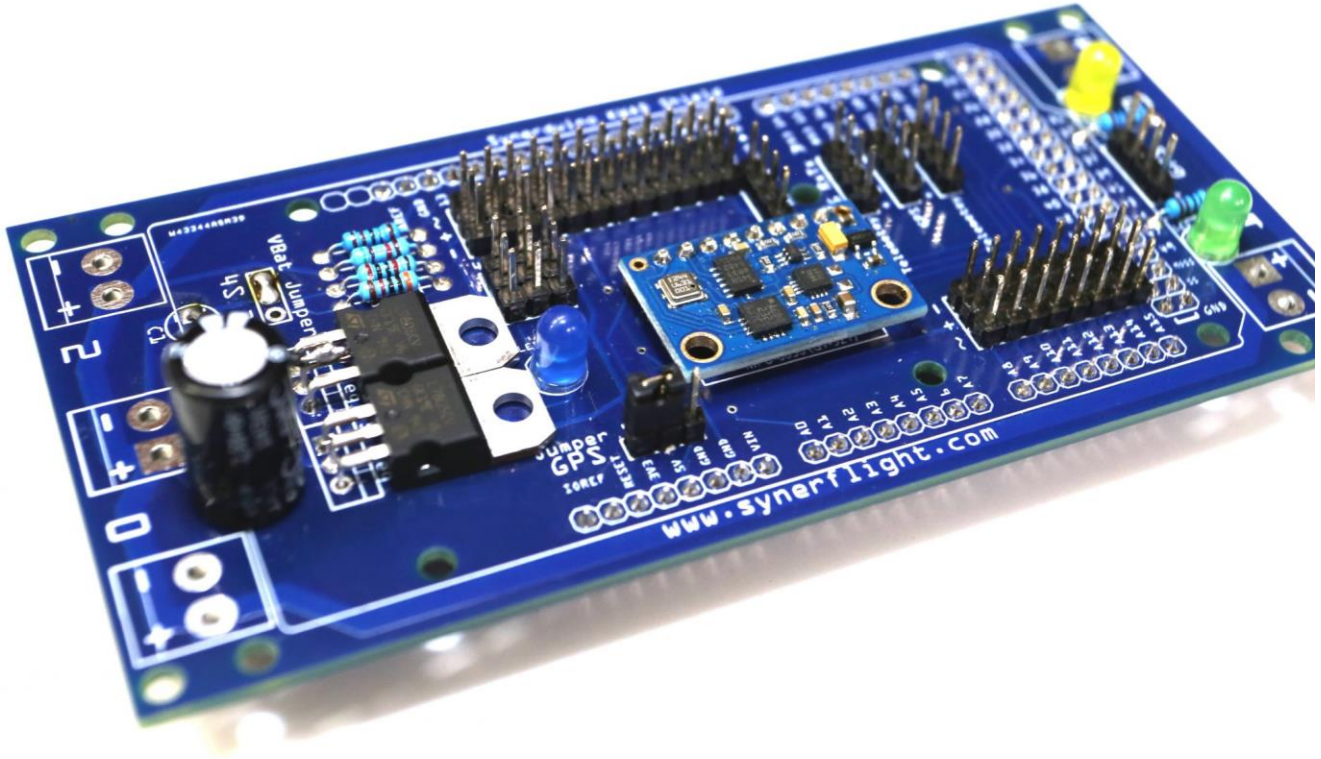
Synerduino Shield Differential propulsion for Tanks and Boats



Hardware

- Synerduino shield
- Arduino2560 MEGA
- Tank or Boat Chassis Kit
- Brushed Motors 6V – 12V
- Gearbox (if your motor set came with it)
- Brush ESC for Car/Boat with Reverse function 6V-12V
- Ublox6 GPS
- Bluetooth

Application Needed

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

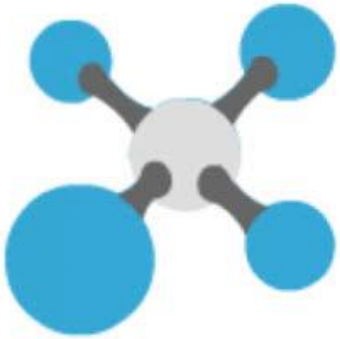
ARDUINO IDE

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

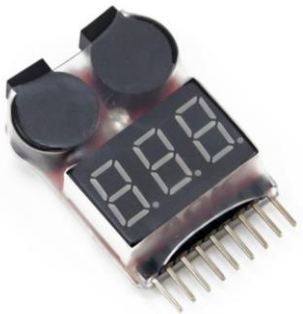
XCTU Configuration Platform

Tools needed

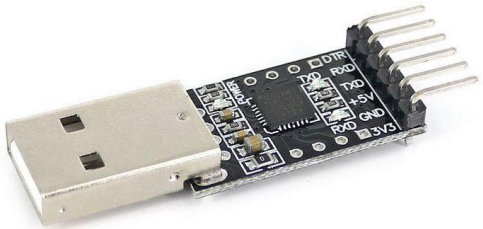
Screwdriver

Cutter knife

Soldering Set



Voltage alarm



USB TTL FTDI

Tools needed

Double sided tape

Transparent tape

5A Balance Charger and Battery (3s 1100mah – 1300mah)

Wheel and Geared motor

Car/Tank 2pcs

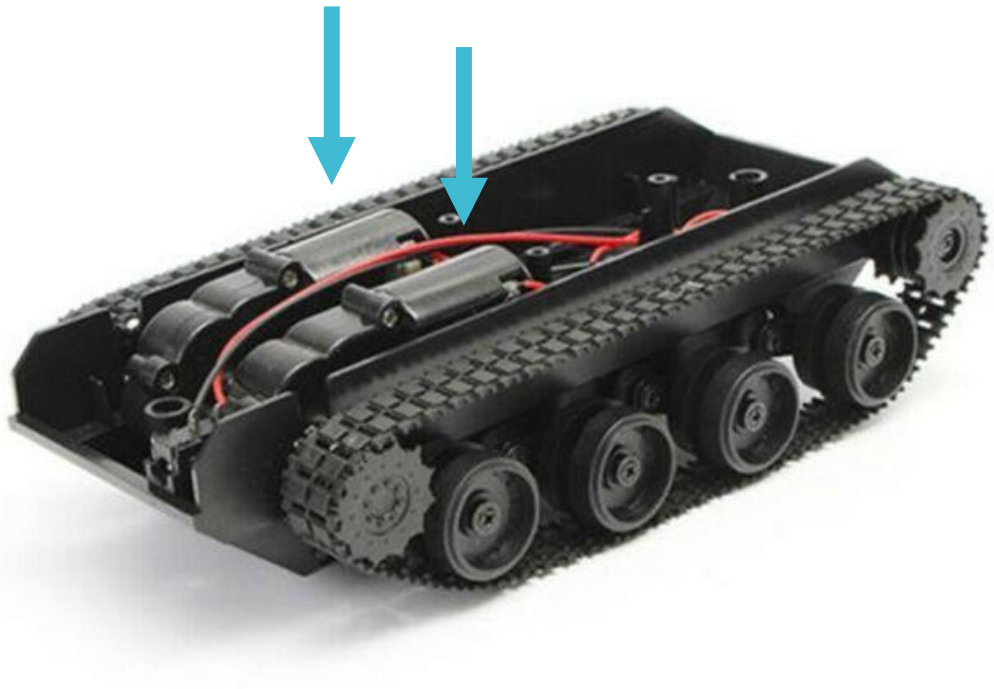
Brushed ESCs with reversing function

2 Pcs of each

Boat Propeller with Waterproof motor Pods

Boat 2pcs

2pcs Motor Differential drive



2pcs Motor Differential drive

Synerduino Kwad Shield

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

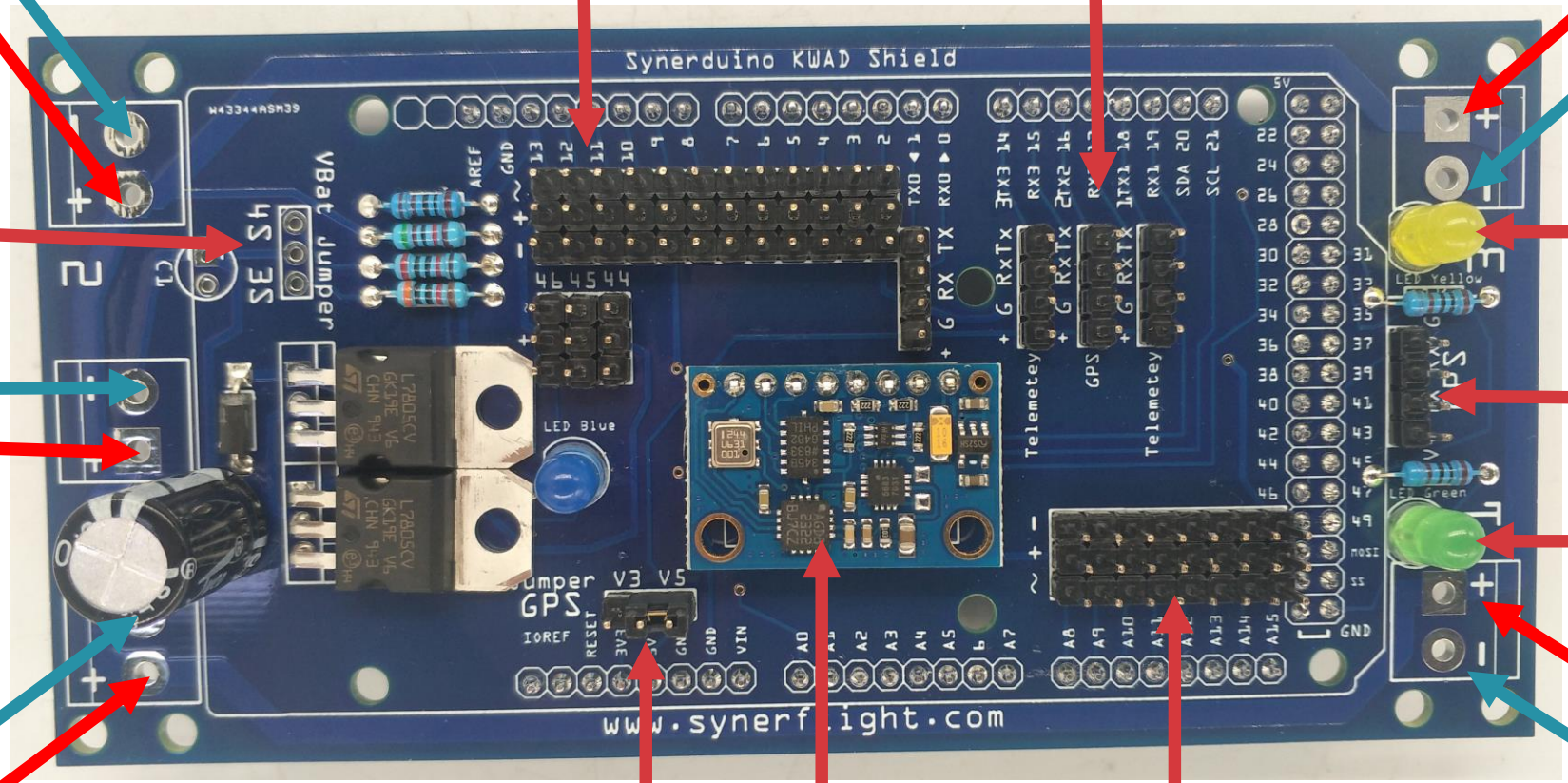
Set jumper to the Battery Cell count (Solded)

GPS LED

Power input 3s 11.1V

GPS Serial Pins

Status LED



ESC is Solder on Top side only

GPS Voltage Jumper

RC PWM in

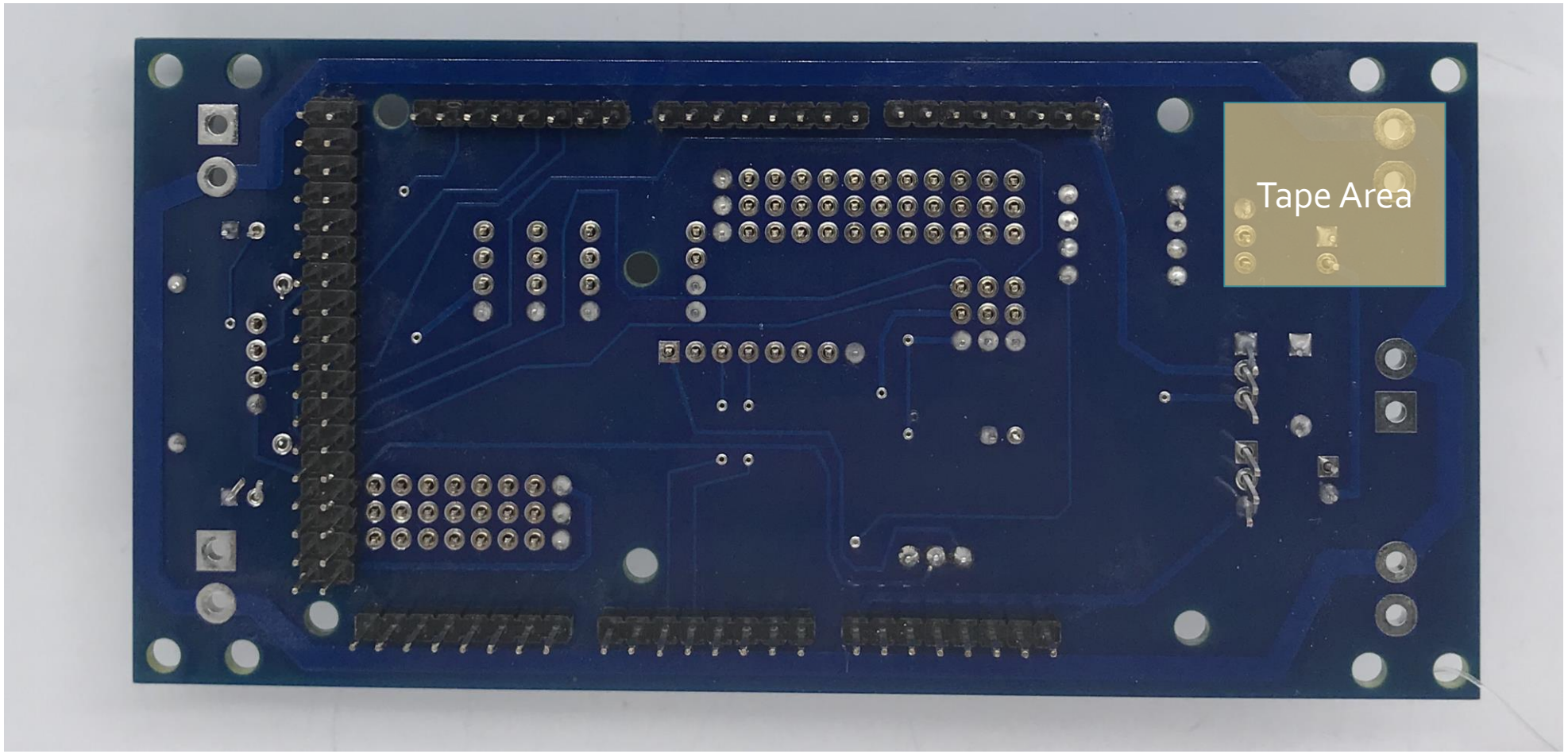
ESC is Solder on Top side only

For improve performance IMU must be protected from the Environment

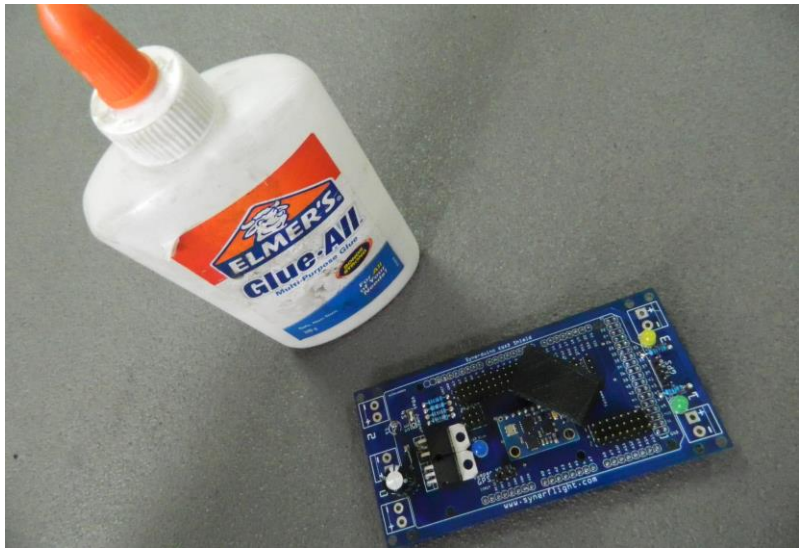
IMU : L3G4200D Gyro / ADXL345 Accelerometer / BMP180 – 85 Baro / MMC5883 Mag

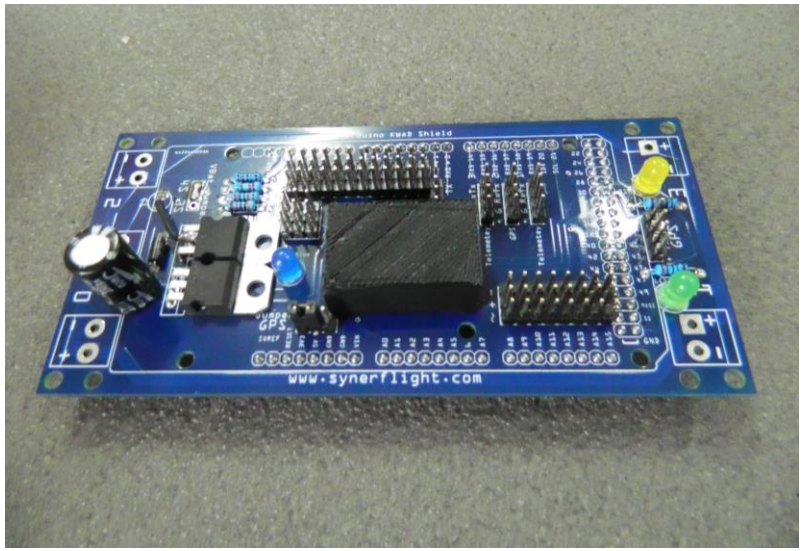
Synerduino Kwad Shield Preparation

Ensure insulation from the Arduino board add tape on these areas

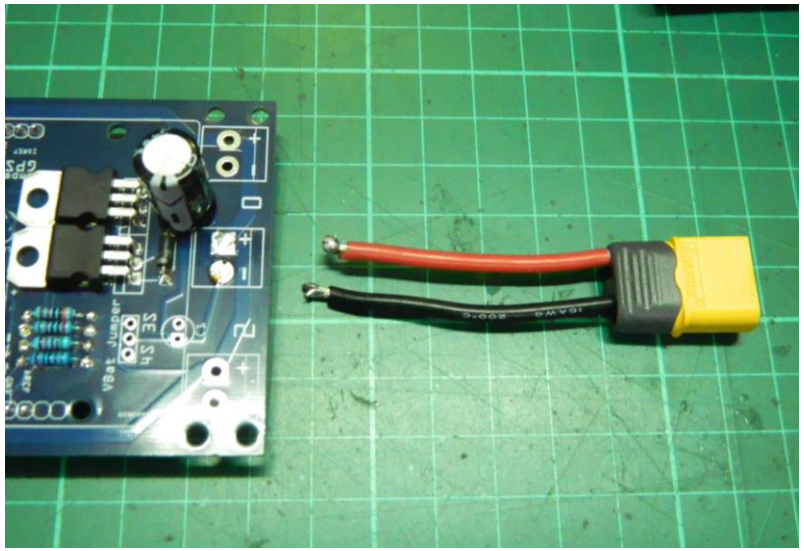


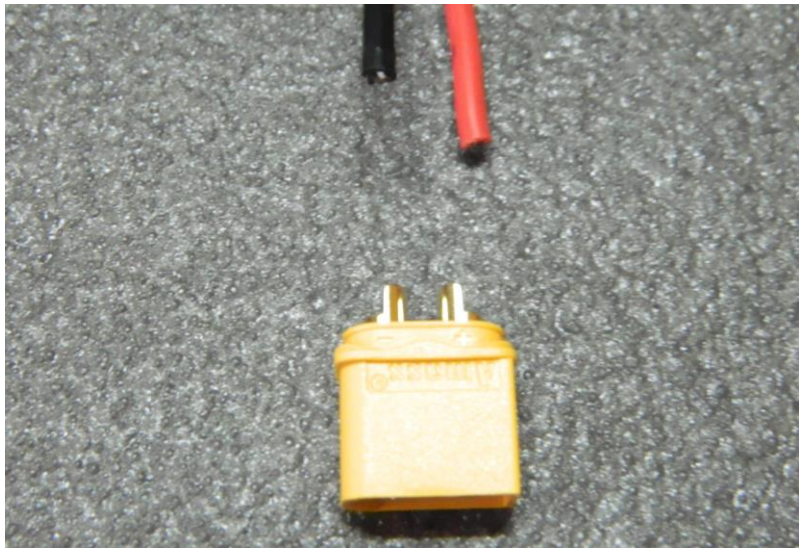
Use PVA to glue the cover to the IMU sensors seal the rim of the cover all the way





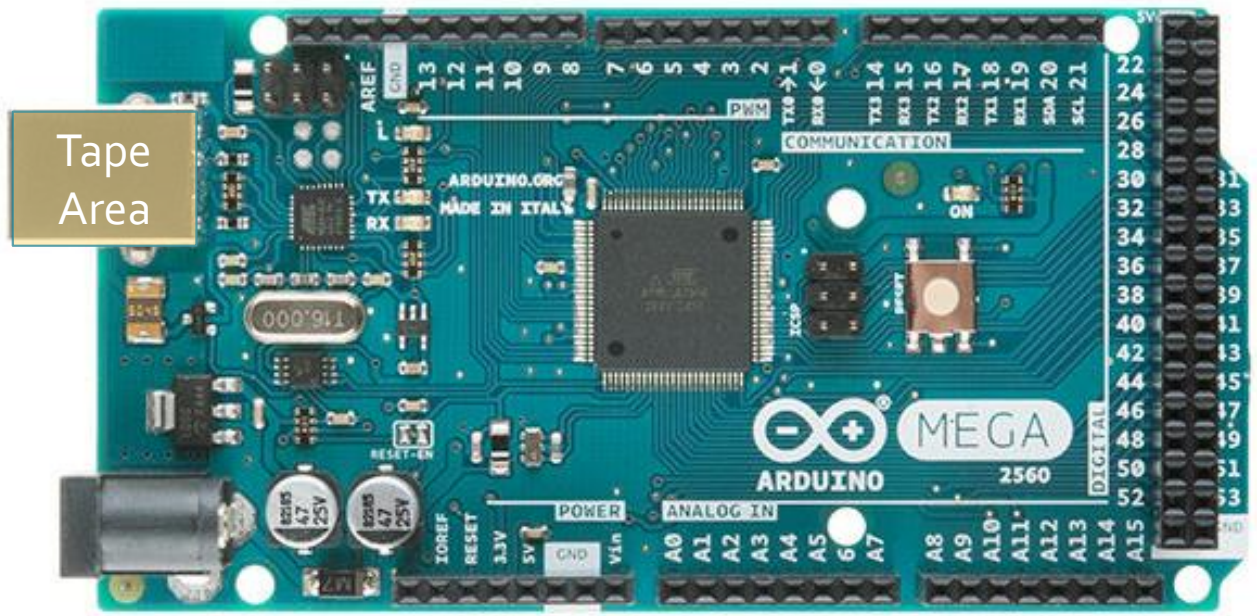
Pls check the polarity of the XT60Plugs to match up with the board’s polarity



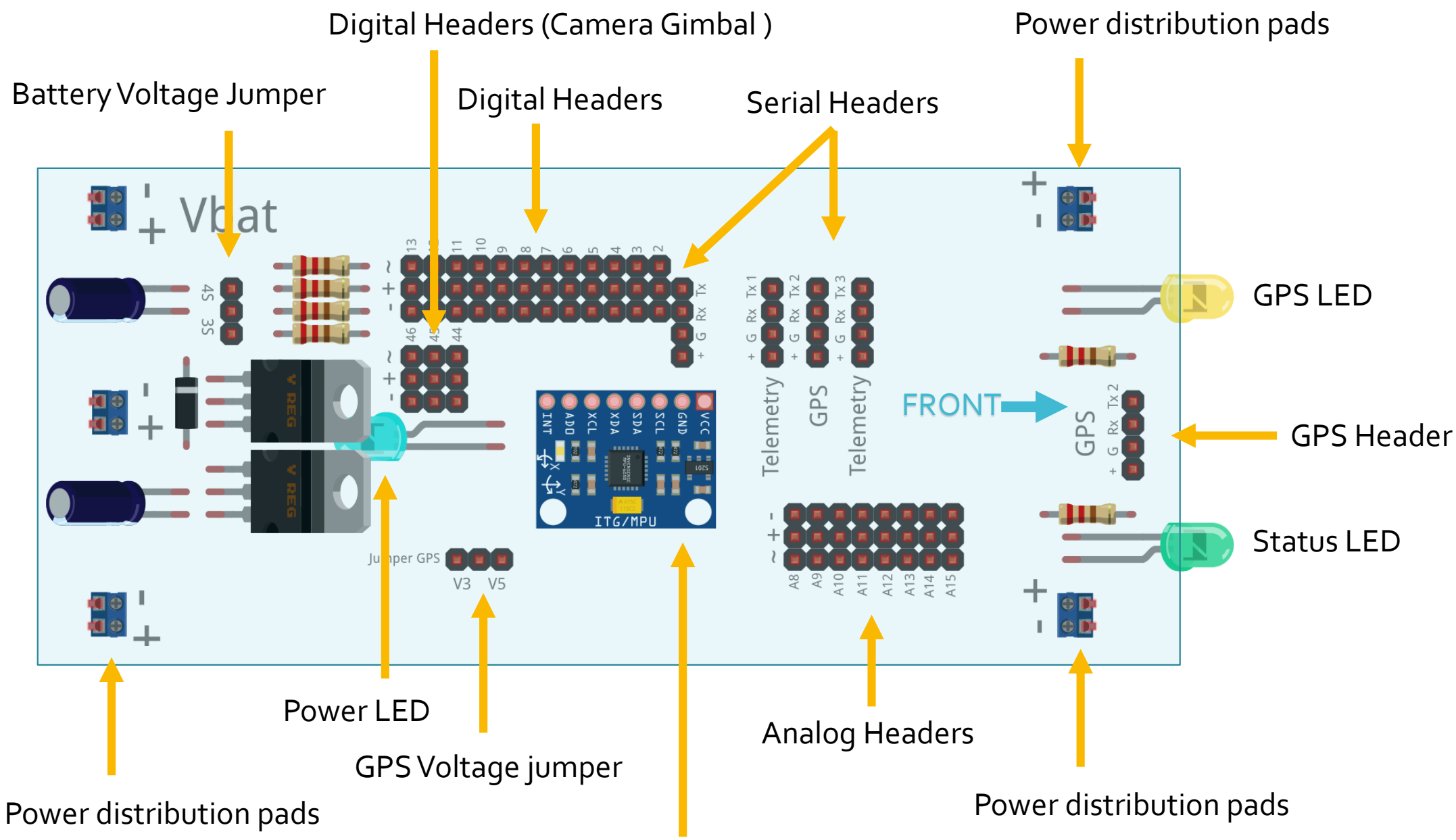


Arduino Board Preparation

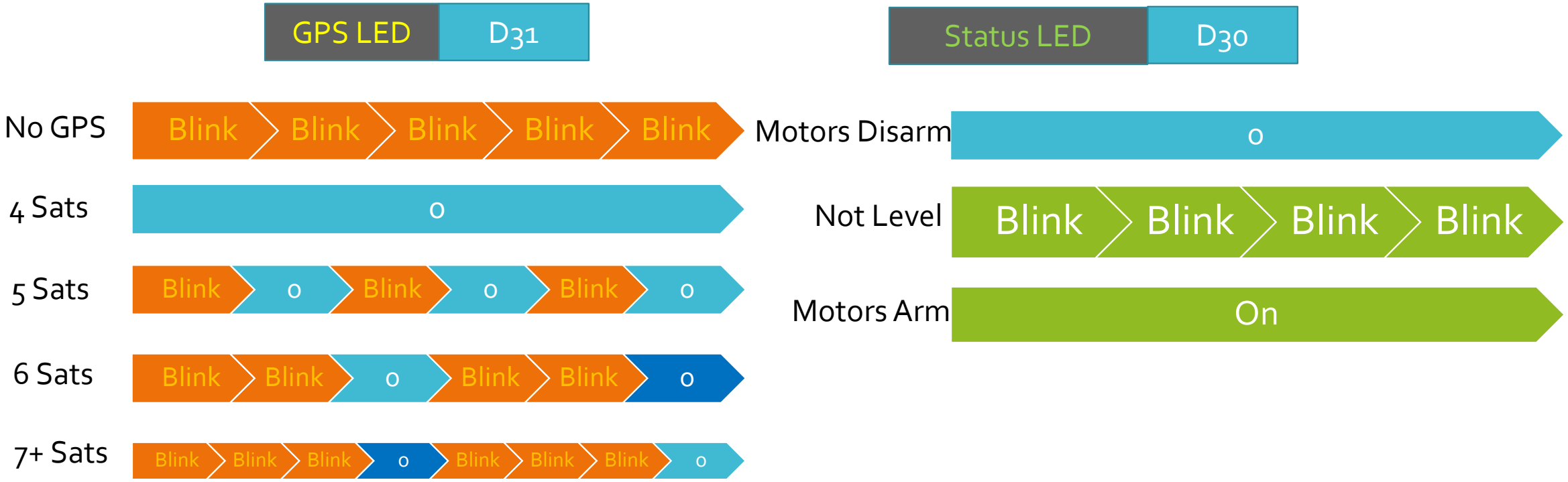
Ensure insulation from the Arduino board add tape on these areas



2560 MEGA

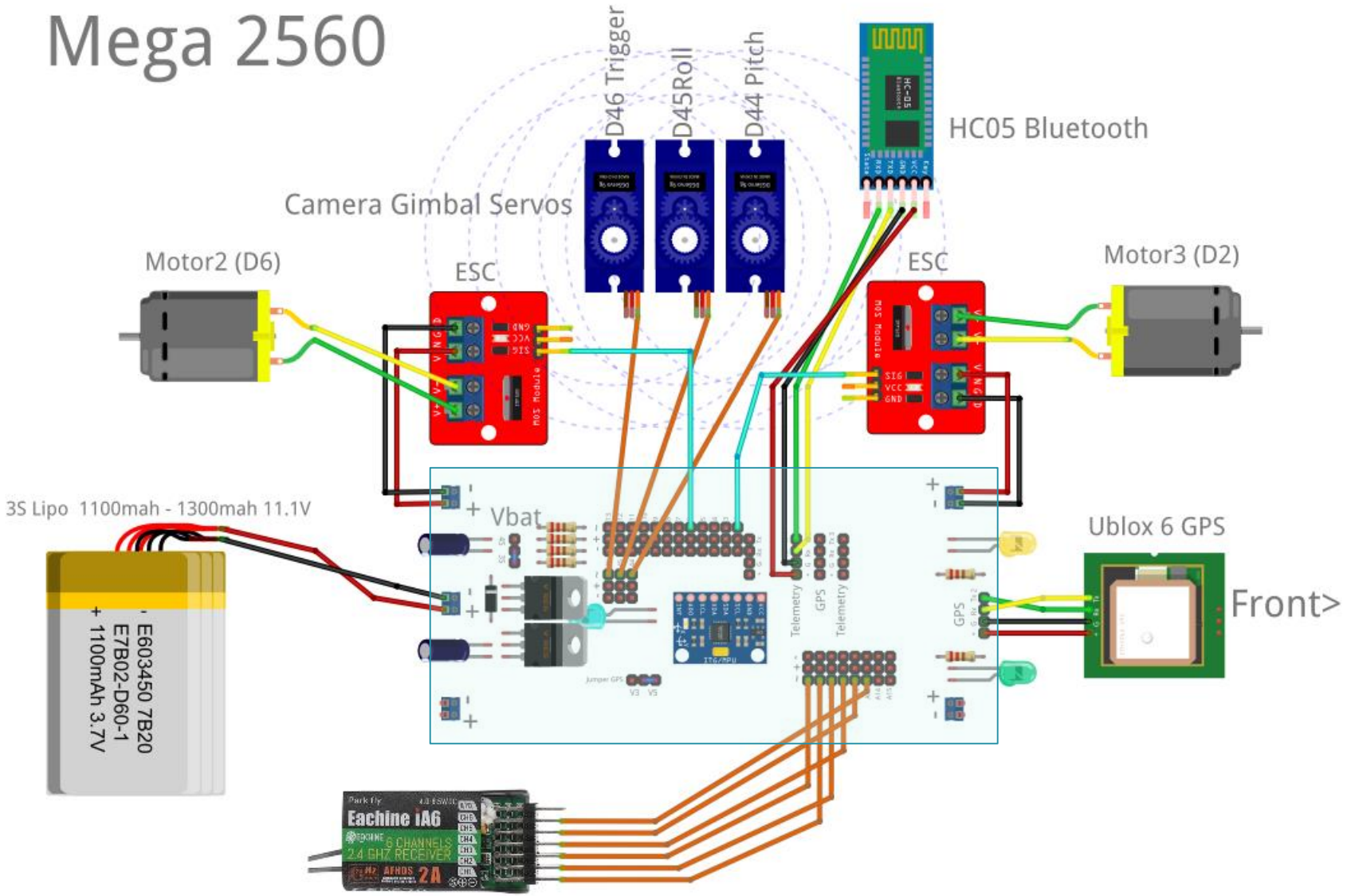

Digital Headers (Camera Gimbal)
Power distribution pads
Battery Voltage Jumper
Digital Headers
Serial Headers
Vbat
GPS LED
FRONT
GPS Header
Status LED
Power LED
GPS Voltage jumper
Analog Headers
Power distribution pads
Power distribution pads
IMU : L3G4200D Gyro / ADXL345 Acc / BMP180 – 85 Baro / MMC5883 Mag

LED Indicator

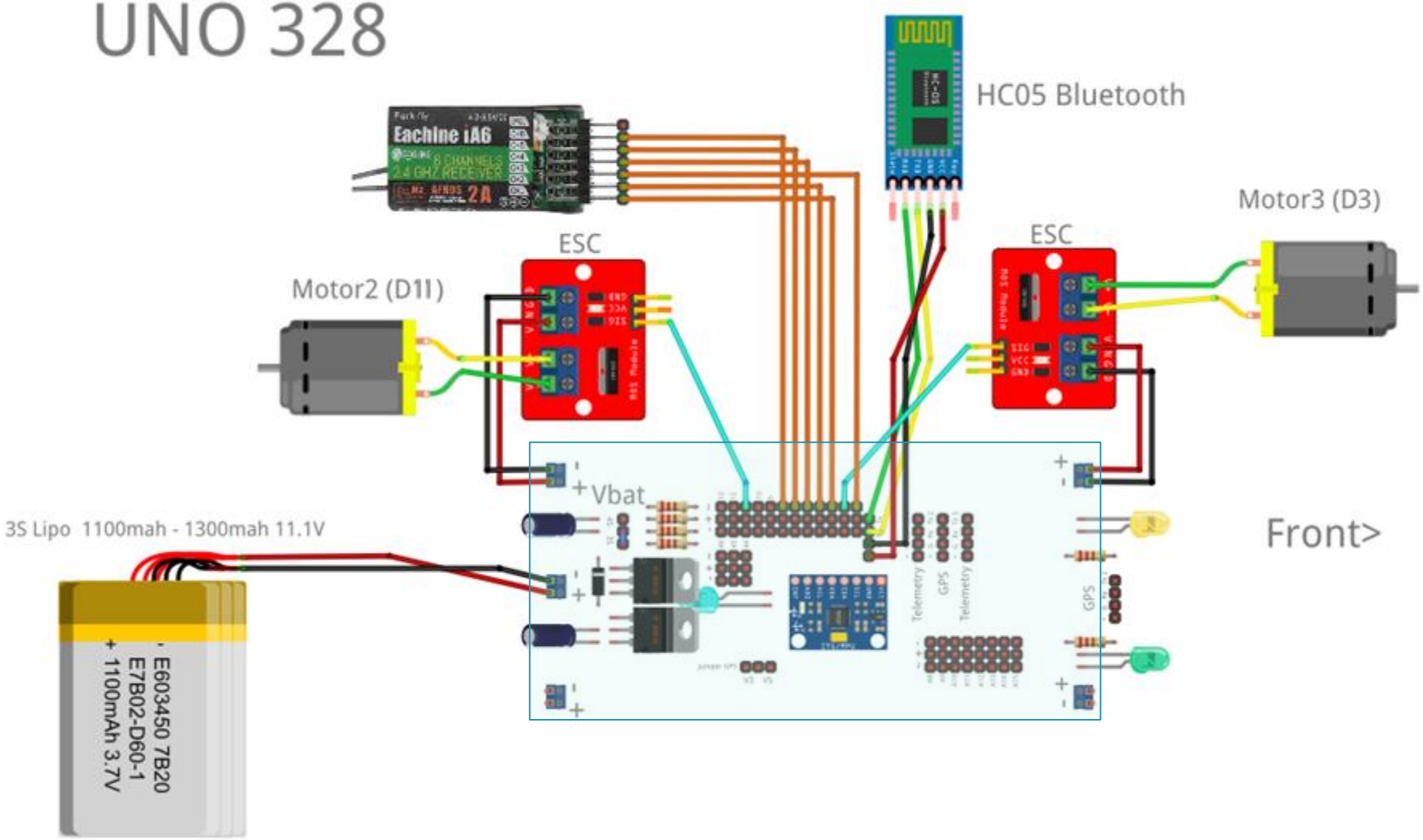


indicate a valid GPS fix by flashing the LED
- led work as sat number indicator
- No GPS FIX -> LED blinks constant speed
- Fix and sat no. below 5 -> LED off
- Fix and sat no. >= 5 -> LED blinks, one blink for 5 sat, two blinks for 6 sat, three for 7 +

Mega 2560

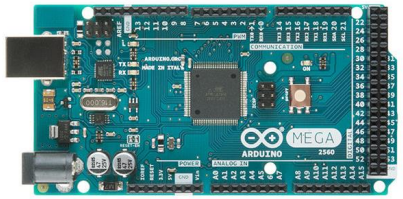

D46 Trigger
D45Roll
D44 Pitch
HC05 Bluetooth
Camera Gimbal Servos
Motor2 (D6)
ESC
ESC
Motor3 (D2)
3S Lipo 1100mah - 1300mah 11.1V
Vbat
Ublox 6 GPS
Front>
E603450 7B20
E7B02-D60-1
1100mAh 3.7V
Eachine iA6
6 CHANNELS
2.4 GHZ RECEIVER
AFHDS 2A

UNO 328


HC05 Bluetooth
Eachine iA6
8 CHANNELS
2.4 GHZ RECEIVER
2A
ESC
Motor2 (D11)
ESC
Motor3 (D3)
Vbat
3S Lipo 1100mah - 1300mah 11.1V
E603450 7B20
E7B02-D60-1
1100mAh 3.7V
Front>

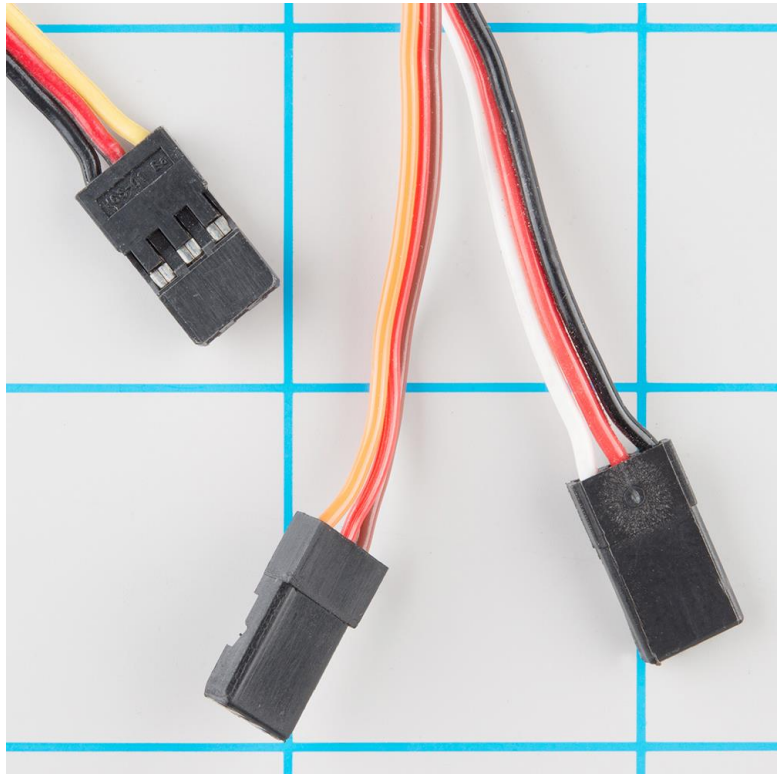
PWM INPUT Assignment

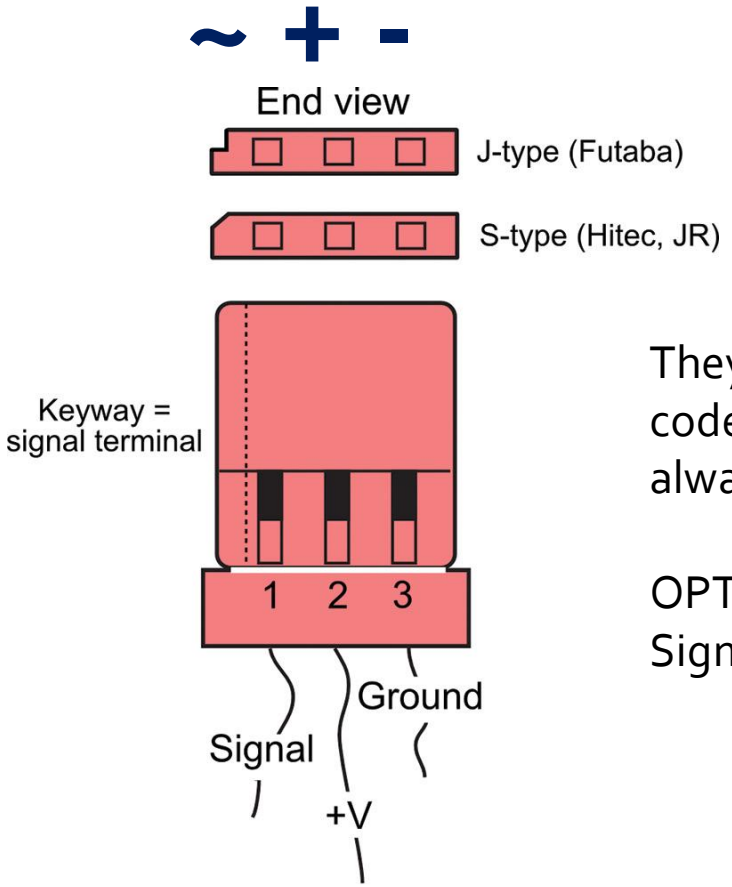
Pls Check the output pin from your Radio Rx manual



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

SERVO HEADER





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

OPTO Wires may only have Signal and Negative Wires o

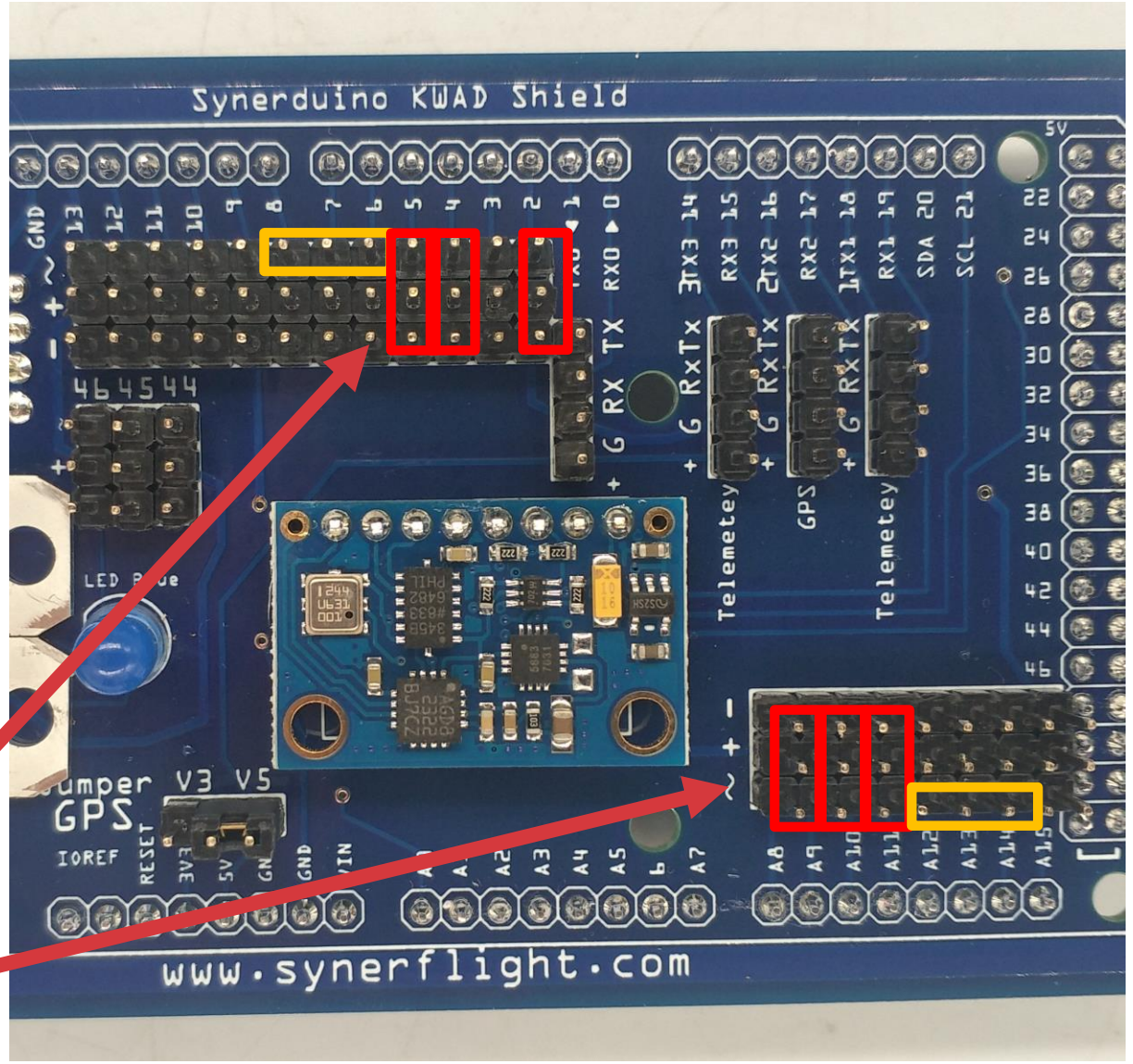
PWM RECEIVER

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

MOSTLY RUDDER AUX1 AND AUX2

UNO PWM IN

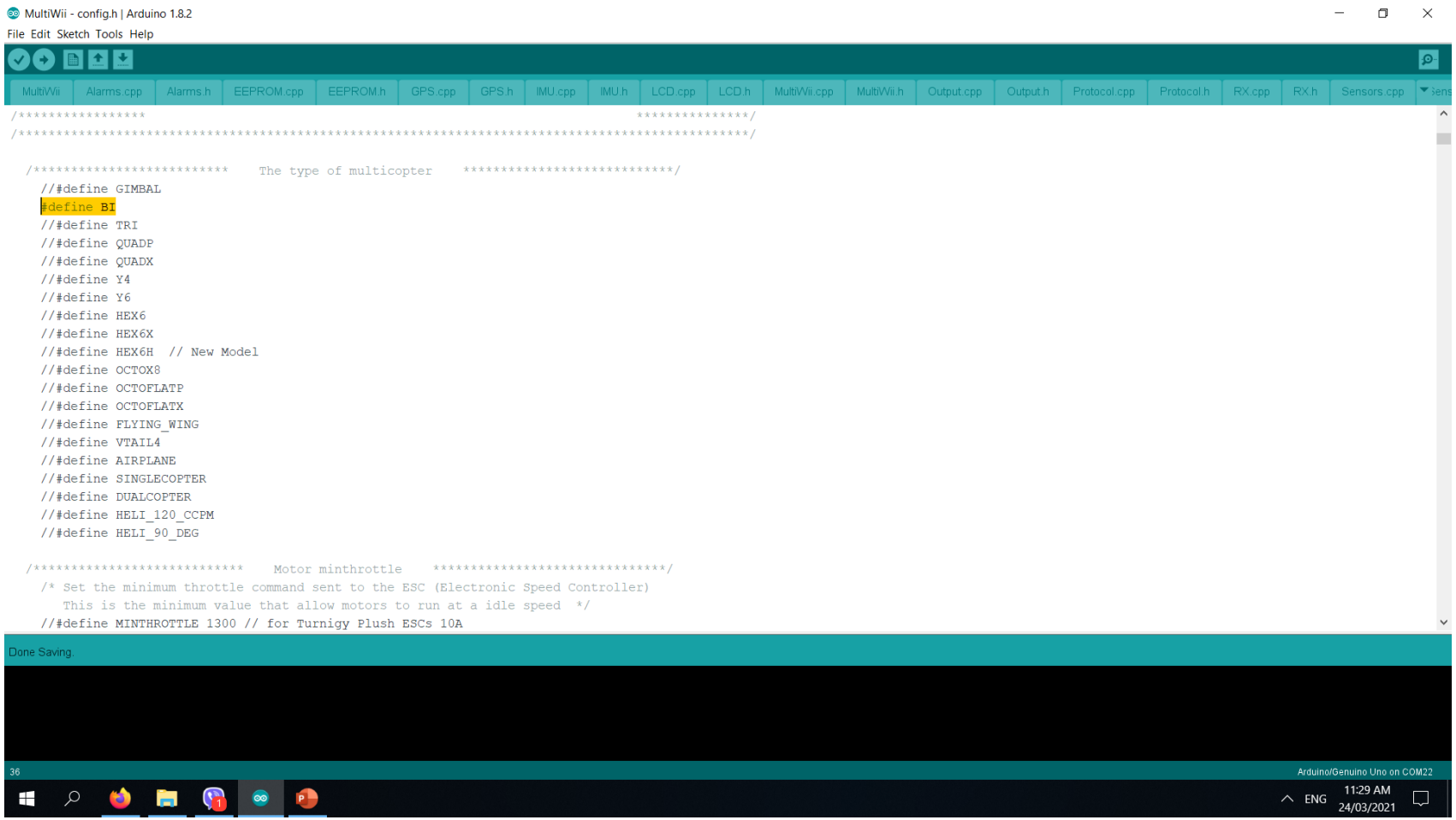
MEGA PWM IN



Config.h

Uncomment #Define Bi

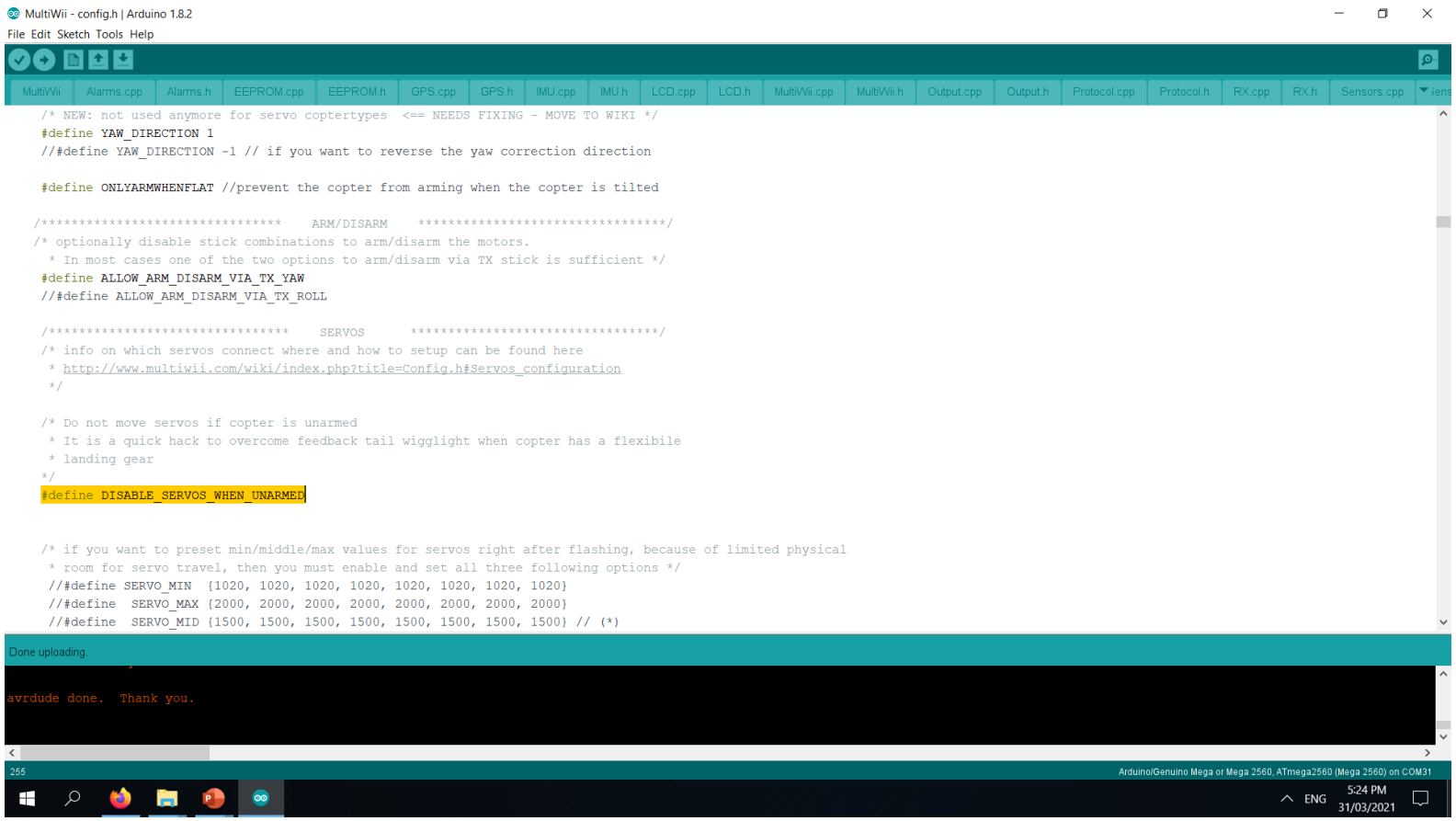
This puts the vehicle into Bicopter or surface vehicle mode



Config.h

Uncomment Disable servos when unarmed

Stop the motor from accidentally running when the vehicle is disarm



Airframe/Chassis

For Tank and Boat chassis Motors use Pins

Select (#Define Bi) Frame on the CONFIG.H

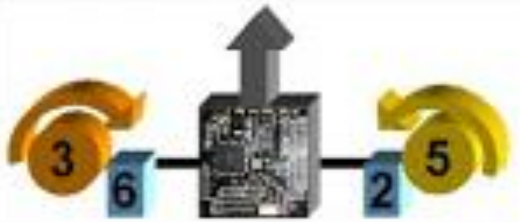
D11 & D3 UNO 328

D6 & D2 MEGA 2560

BICOPTER avatar style

for Arduino 328p:

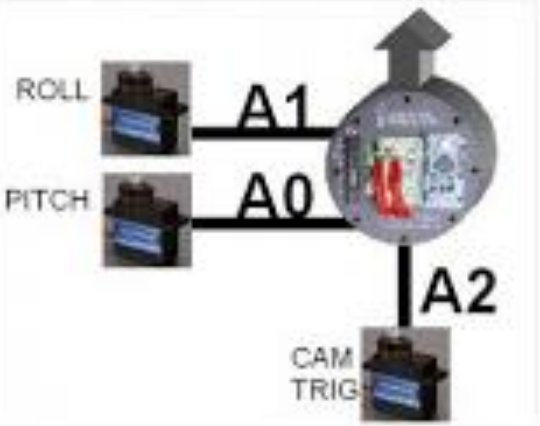
for arduino mega:



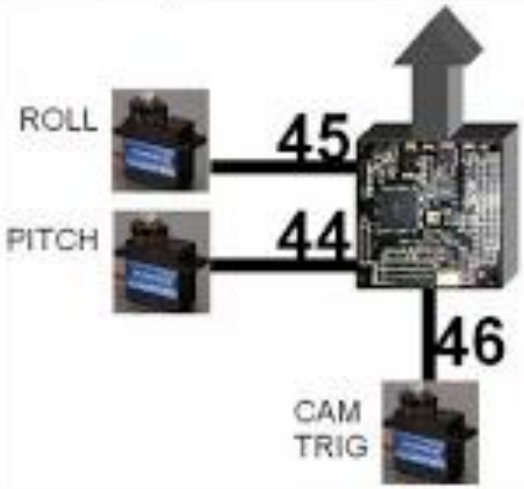
Pure stabilized gimbal system

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 328p:



for arduino mega:

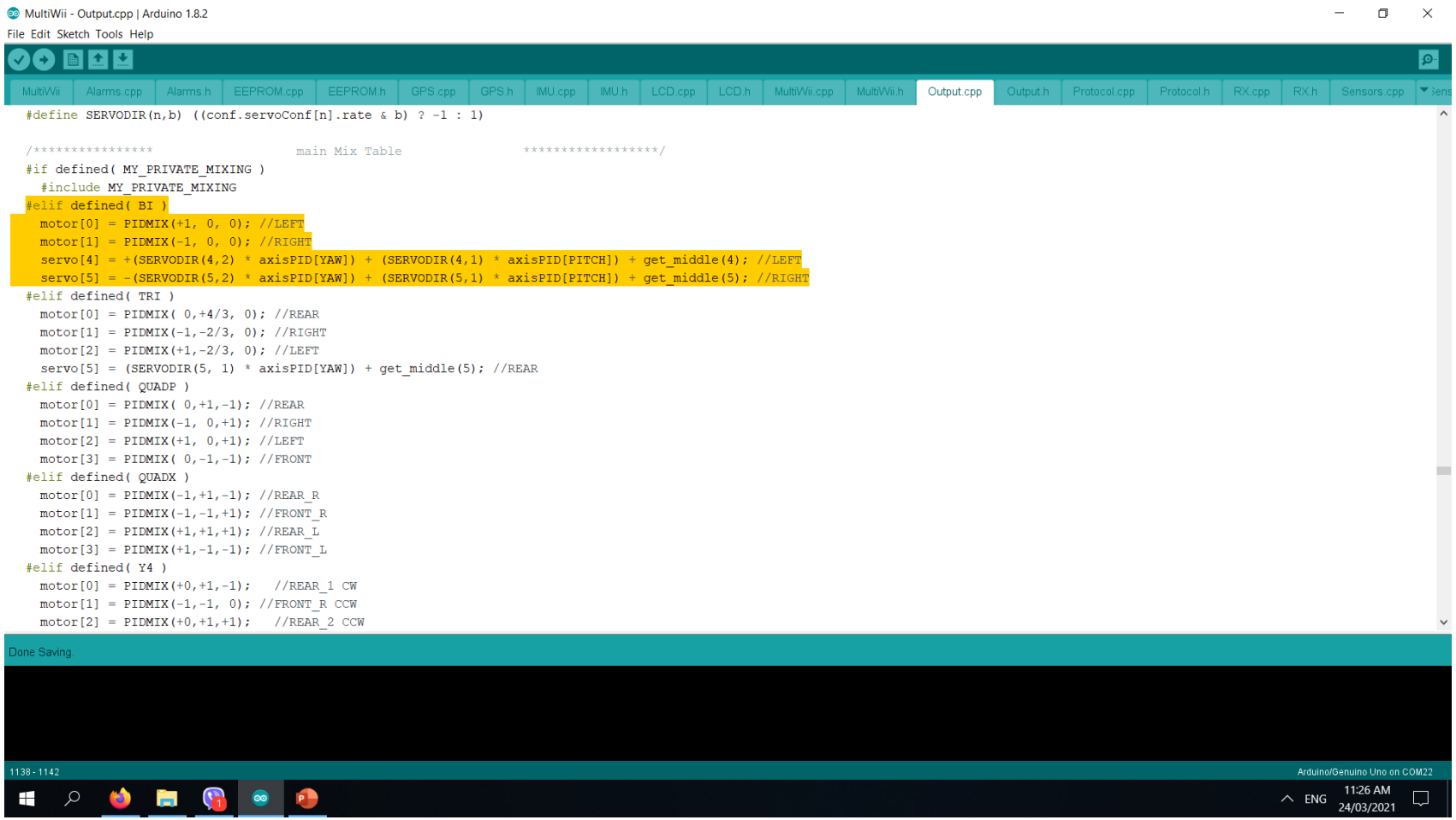


Output.cpp

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

Under ServoDir 4 & 5 (servo number , servo direction)

1=forward 2=reverse



INERTIAL MEASURING UNIT MEASURING UNIT

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

Magnetometer

Barometer

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

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

Accelerometer

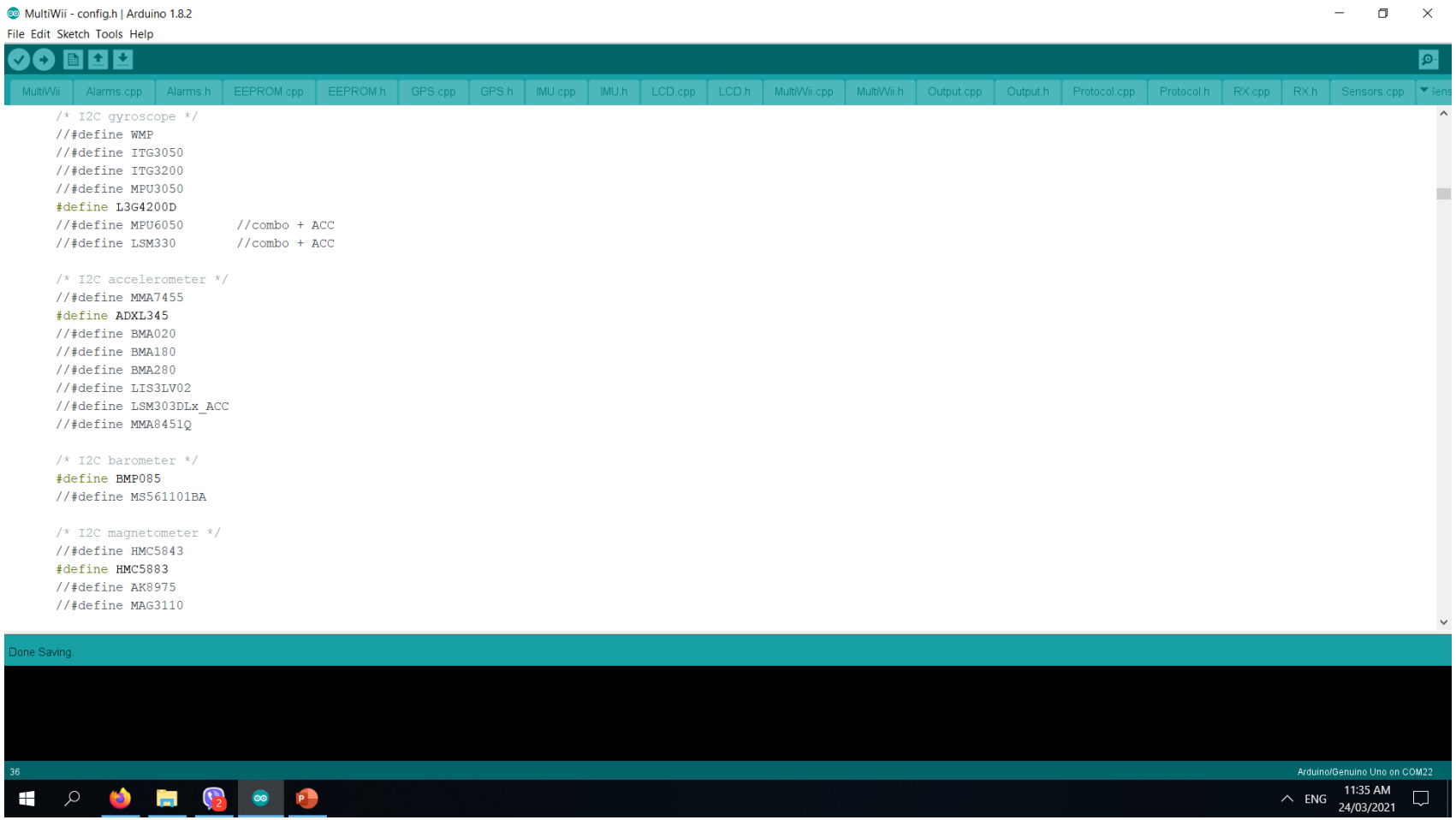
Gyroscope

Each sensor has a corresponding address registry set by manufacturer

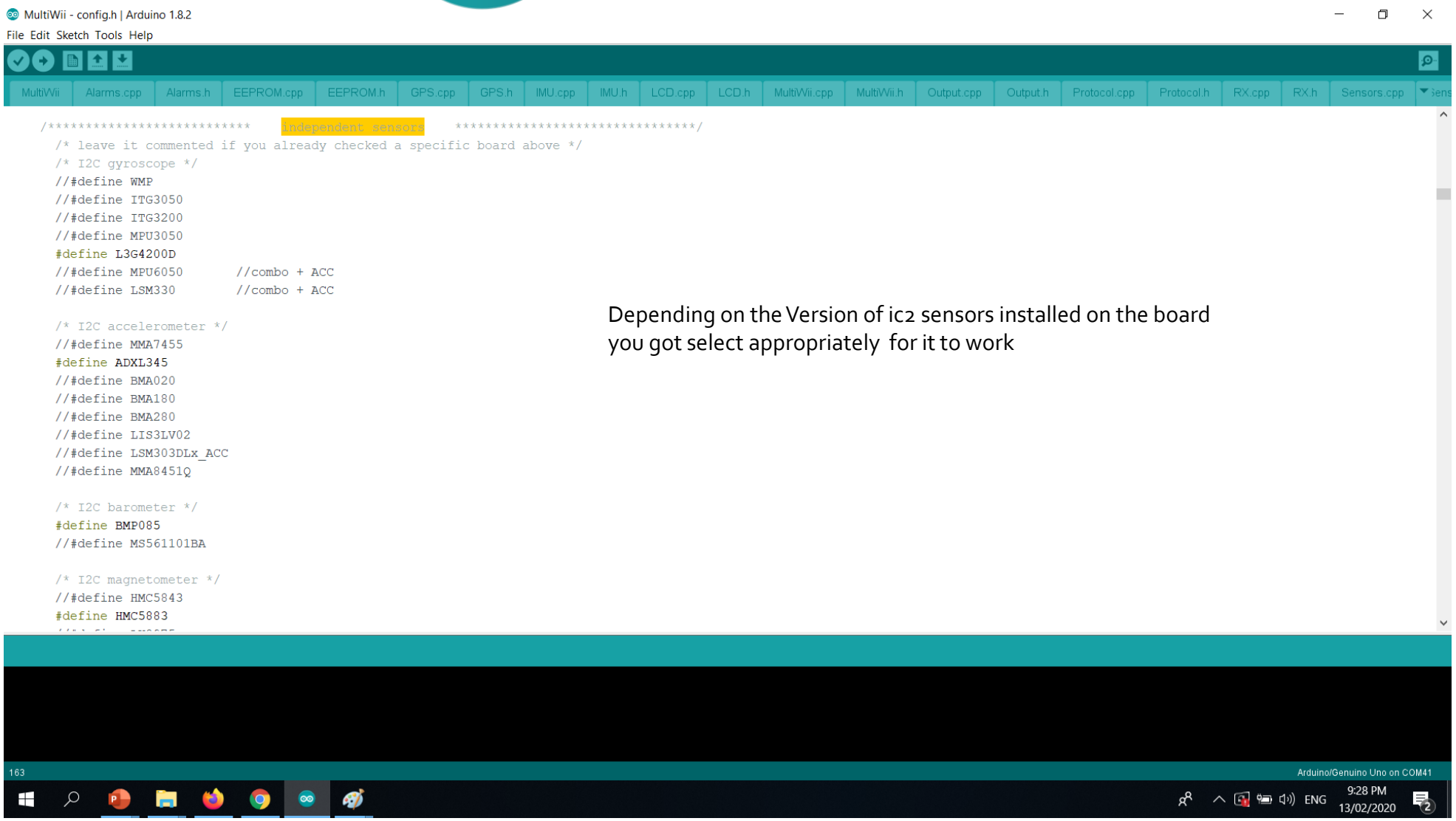
You can find it on sensors.ccp tab

Config.h

Sensors
#define L3G4200D
#define ADXL345
#define BMP085
#define HMC5883



CONFIG.H

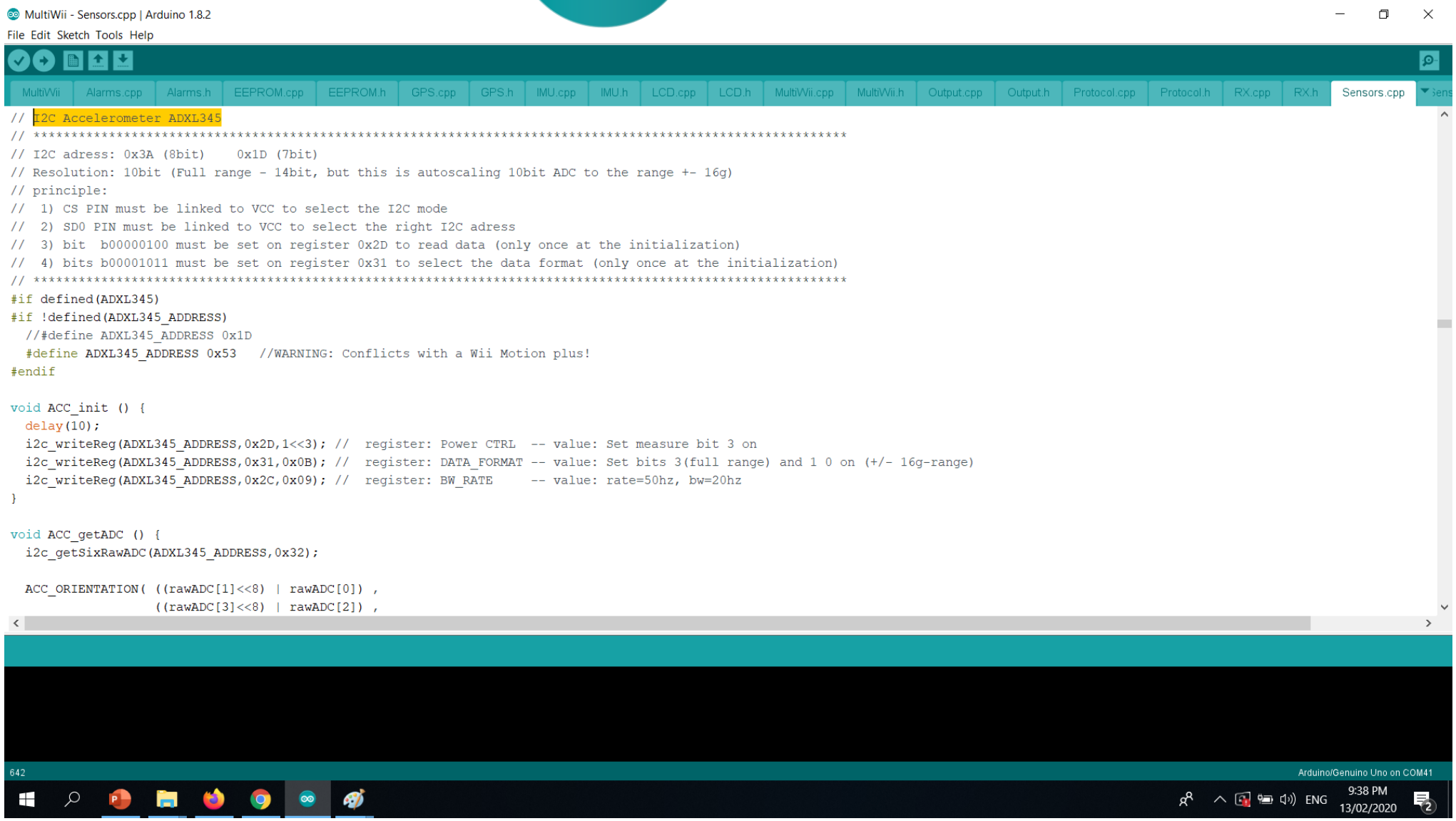


Depending on the Version of ic2 sensors installed on the board you got select appropriately for it to work

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

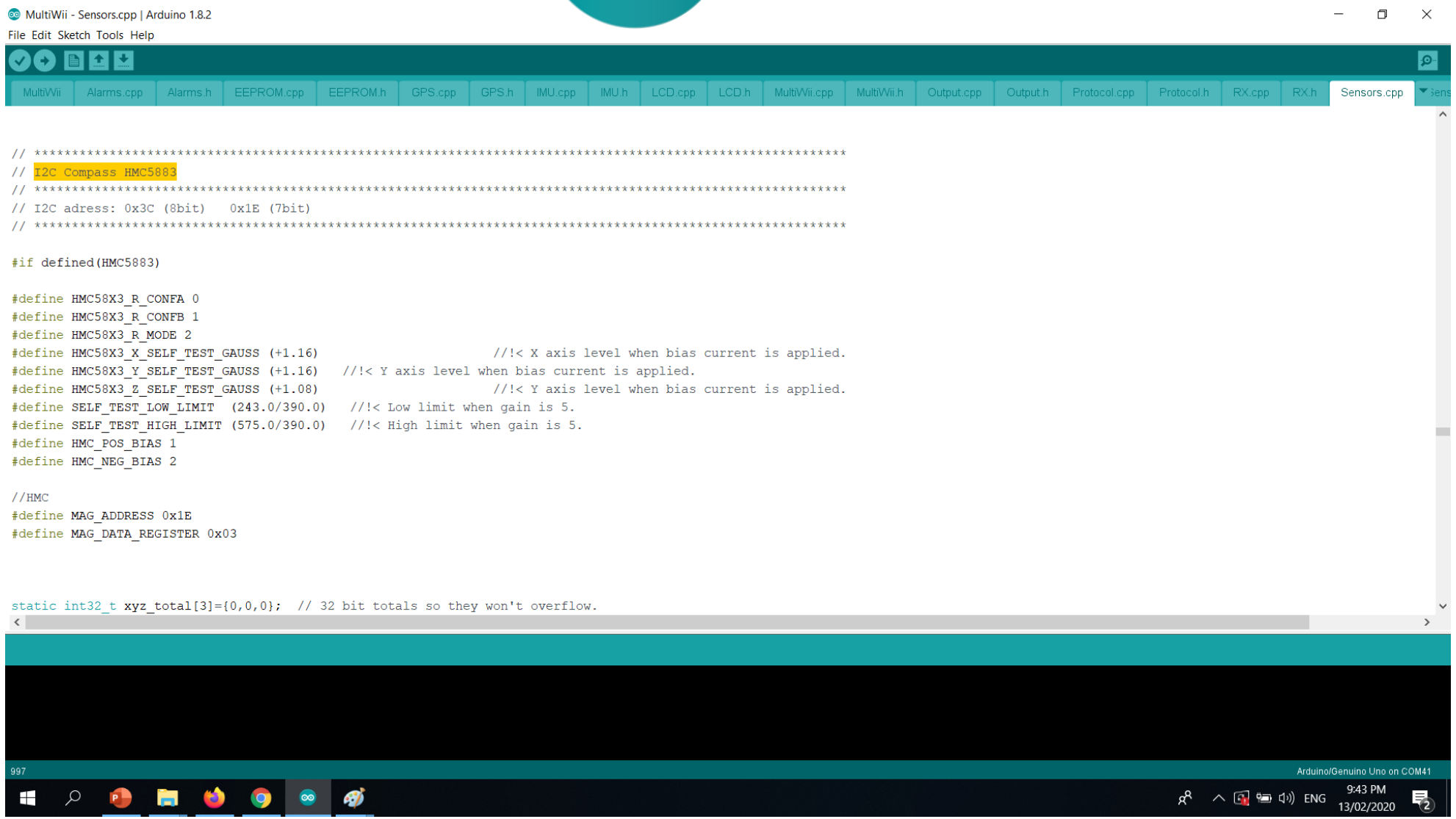
SENSORS.CCP


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

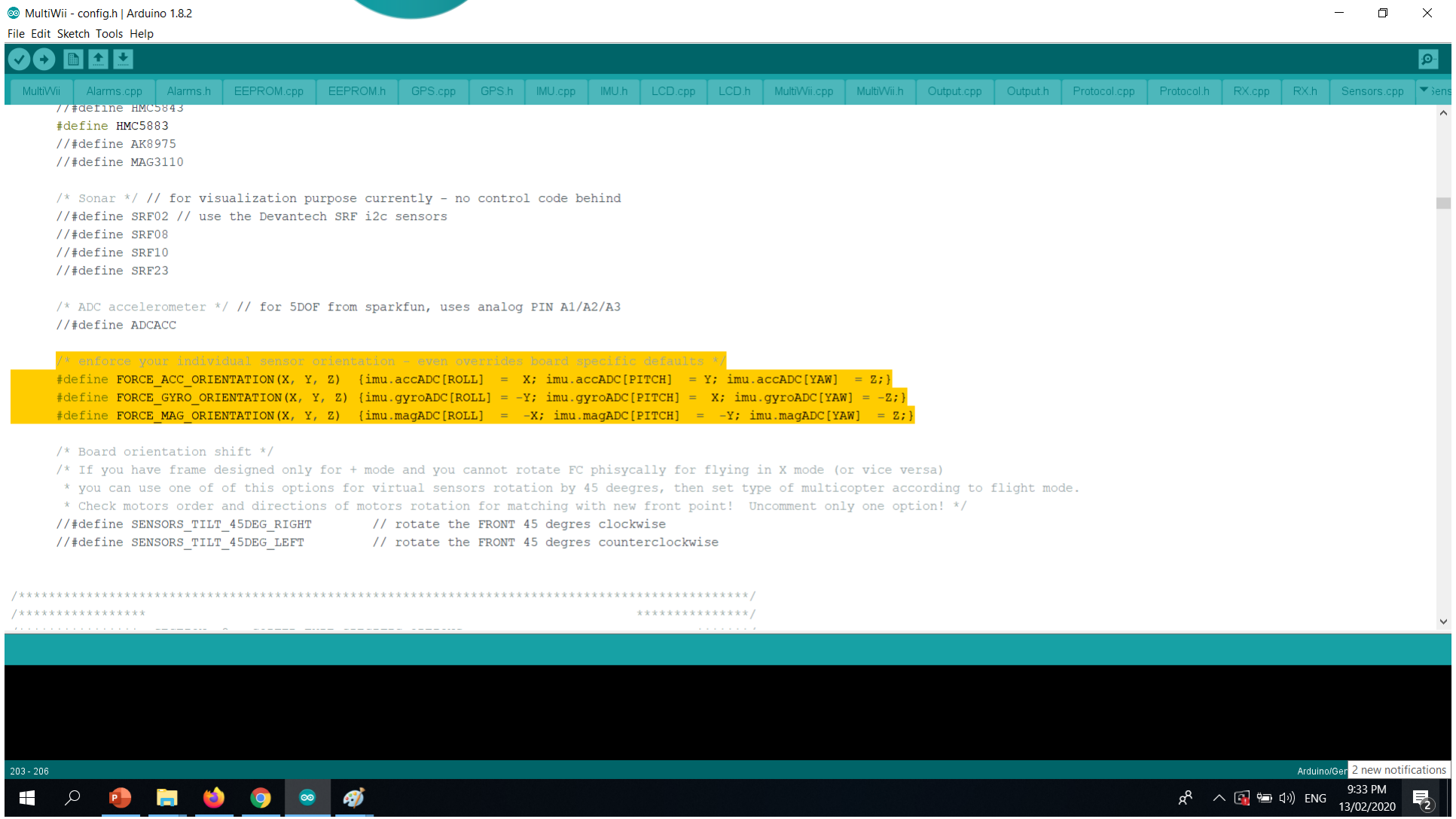
SENSORS.CCP


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

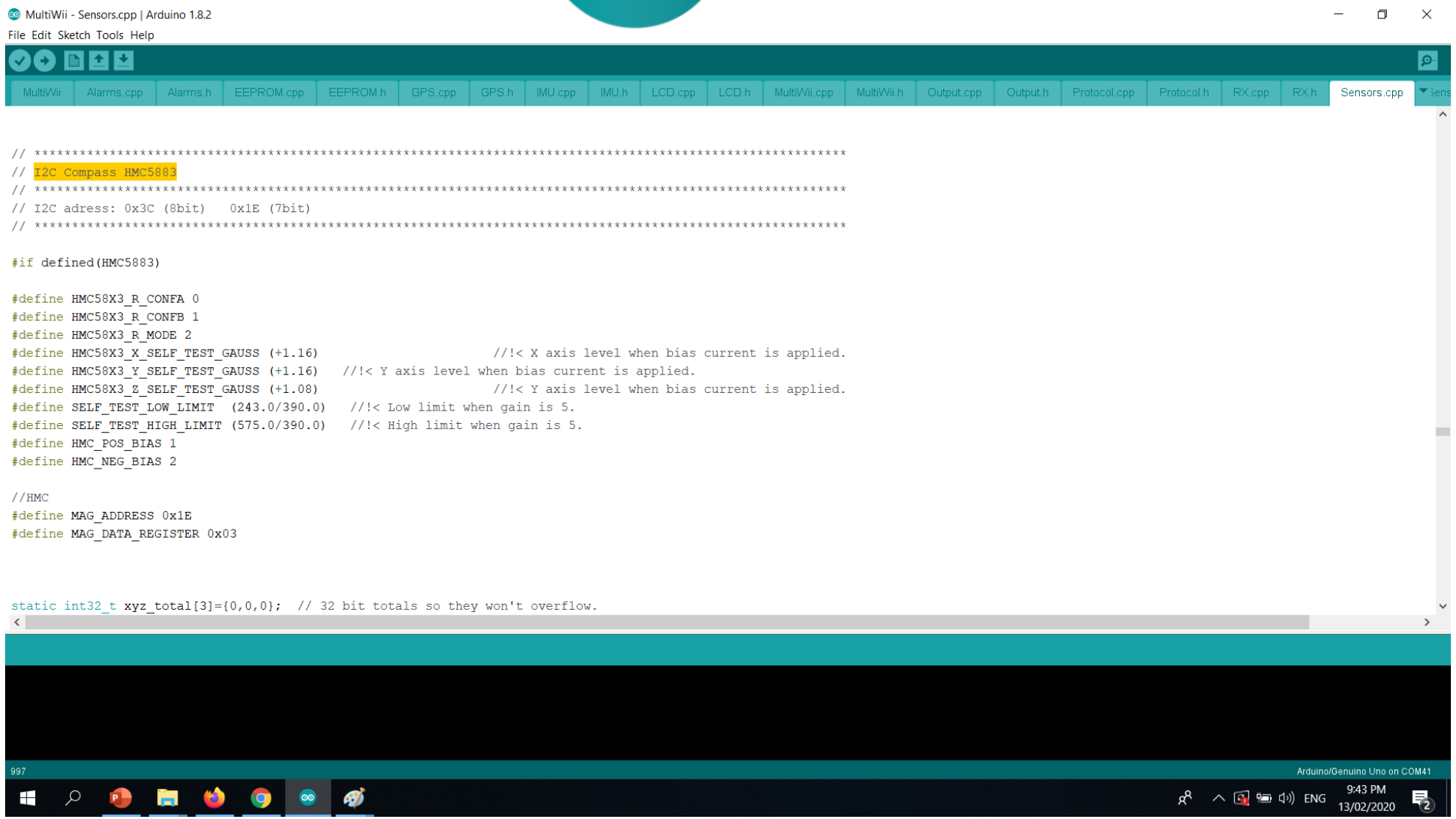

MultiWii - Sensors.cpp | Arduino 1.8.2
File Edit Sketch Tools Help
MultiWii
Alarms.cpp
Alarms.h
EEPROM.cpp
EEPROM.h
GPS.cpp
GPS.h
IMU.cpp
IMU.h
LCD.cpp
LCD.h
MultiWii.cpp
MultiWii.h
Output.cpp
Output.h
Protocol.cpp
Protocol.h
RX.cpp
RX.h
Sensors.cpp
// I2C Compass HMC5883
// I2C adress: 0x3C (8bit) 0x1E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

CONFIG.H



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

SENSORS.CPP


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

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


TELEMETRY
GROUND
STATION
Port COM4
Speed 115200
Connect
Read Settings
Write Settings
Load Defaults
Load from File
Save to File
Start Log
Start GPS log
Log Browse
Flight Deck
Mission
Flight Tuning
FC Config
RC Control Settings
Sensor Graph
VideoCapture
GUI Settings
CLI
Bind Spektrum
QuadCopter +
1500
FRONT
1500
LEFT
1500
RIGHT
1500
REAR
Thr
Pitch
Roll
Yaw
Aux1
Aux2
Aux3
Aux4
Aux5
Aux6
Aux7
Aux8
Active Sensors
ACC
GPS
BARO
0.0v
Telemetry link info
RSSI
Noise
no data
no data
Packet's sent 0
Packet's received 0
Packet CRC error 0
I²C 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

COMPASS (MAG AND GYRO)

SAVE CONFIG

PWM OUTPUT INDICATOR

PWM INPUT INDICATOR

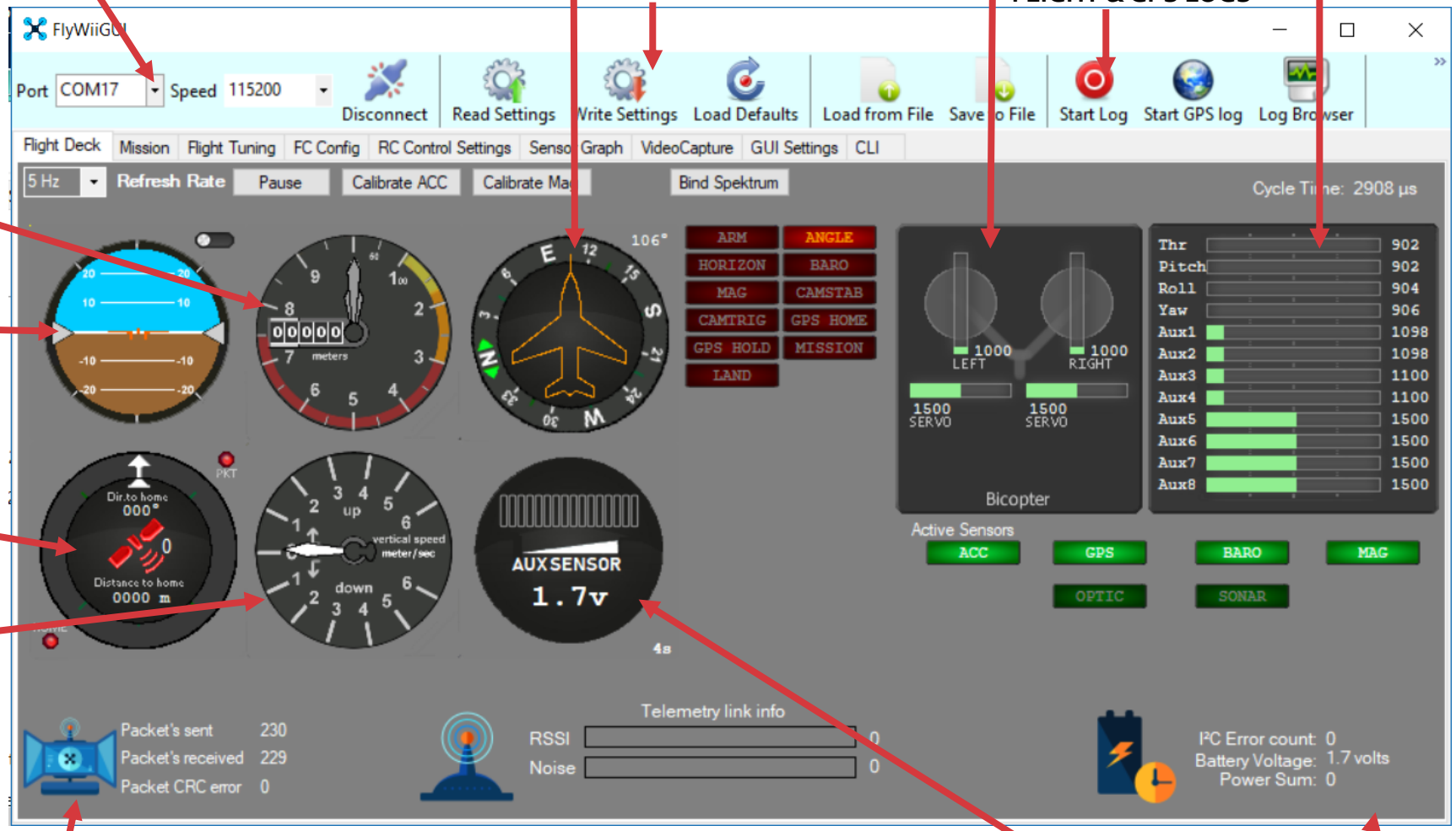
FLIGHT & GPS LOGS

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

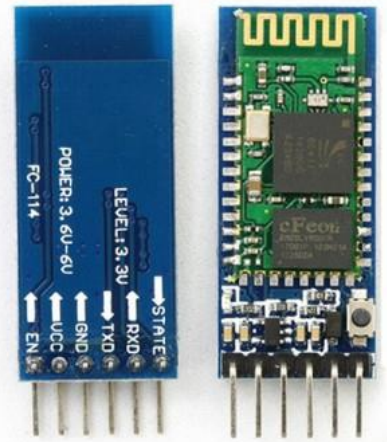
(IF THE ERROR NUMBERS ARE HIGH PLS CHECK YOUR TELEMETRY CONFIG)

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

TELEMETRY

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

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

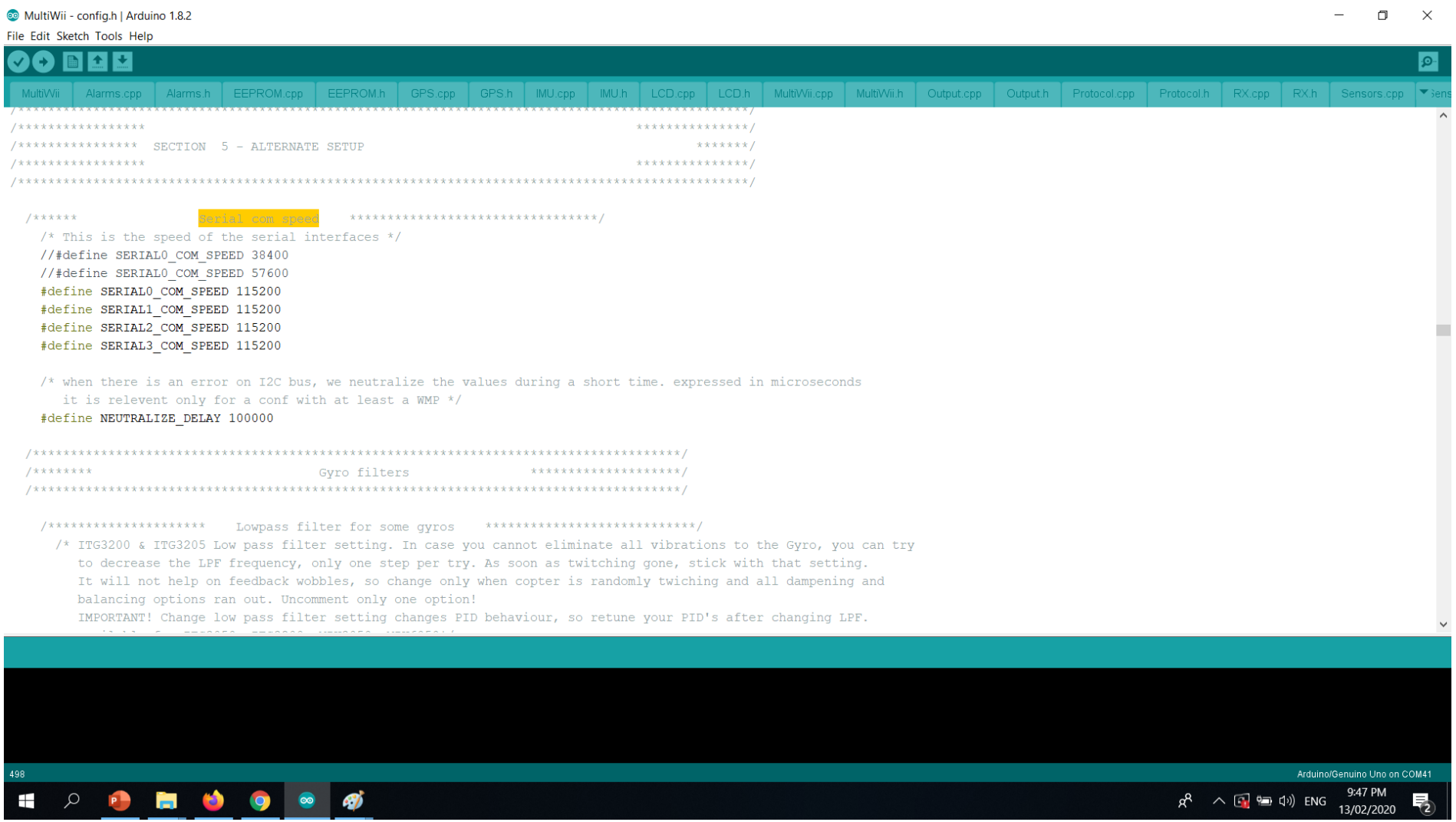
115200 FOR BLUETOOTH HC-05

38400 FOR XBEE RADIO

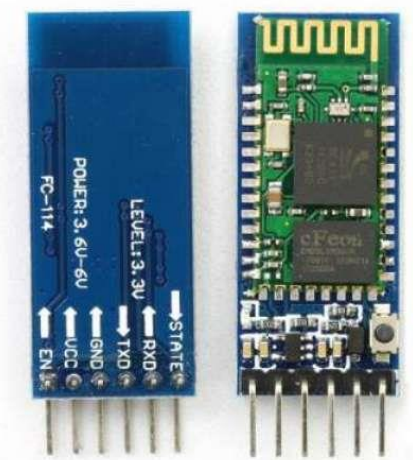
57600 for SIK RADIO

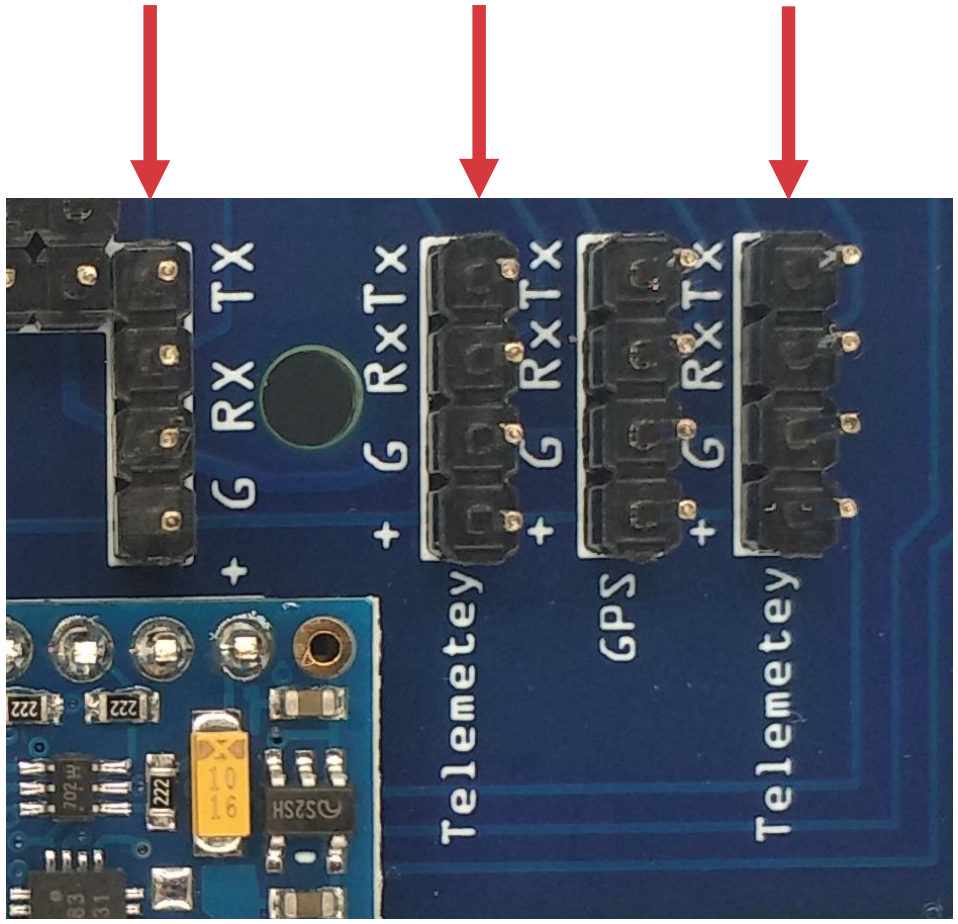
SERIAL 2 IS RESERVE FOR GPS

NMEA Baud 57600

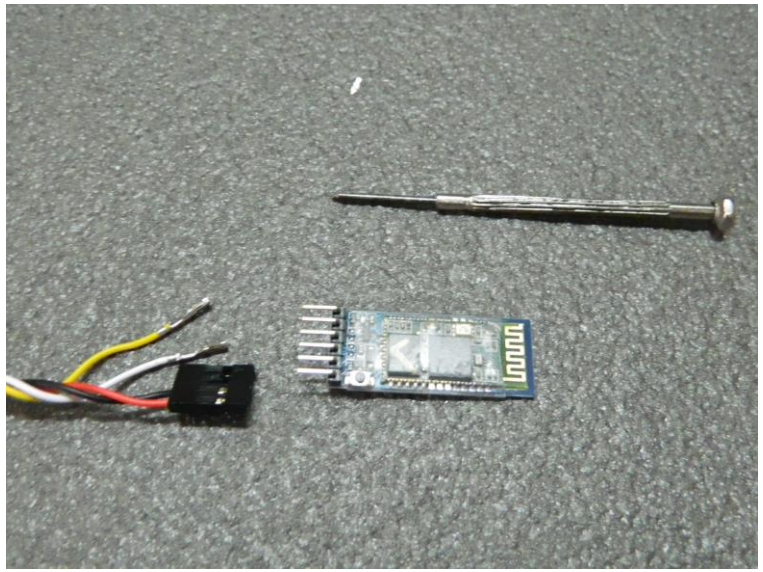


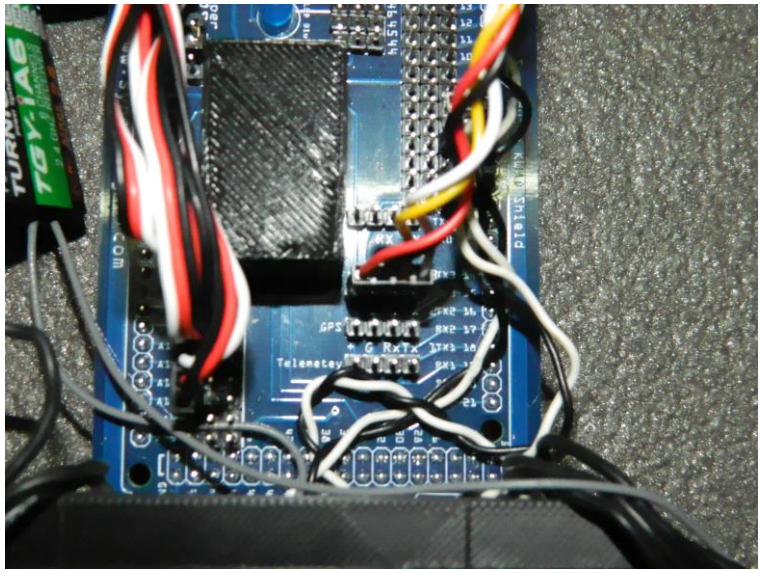
Telemetry Pins





Bluetooth





BLUETOOTH PLUG INTO SERIAL 1 OR SERIAL 3

115200 FOR BLUETOOTH HC-05

ATTENTION:

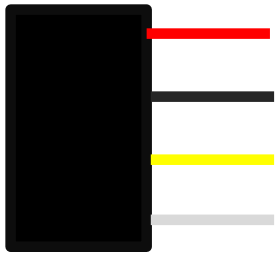
YOU MAY NEED TO REARRANGE THE HEADERS TO CONNECT THE BLUETOOTH MODULE TO THE SHIELD BOARD ACCORDINGLY

VCC >> +

GND >> G

TX >> RX

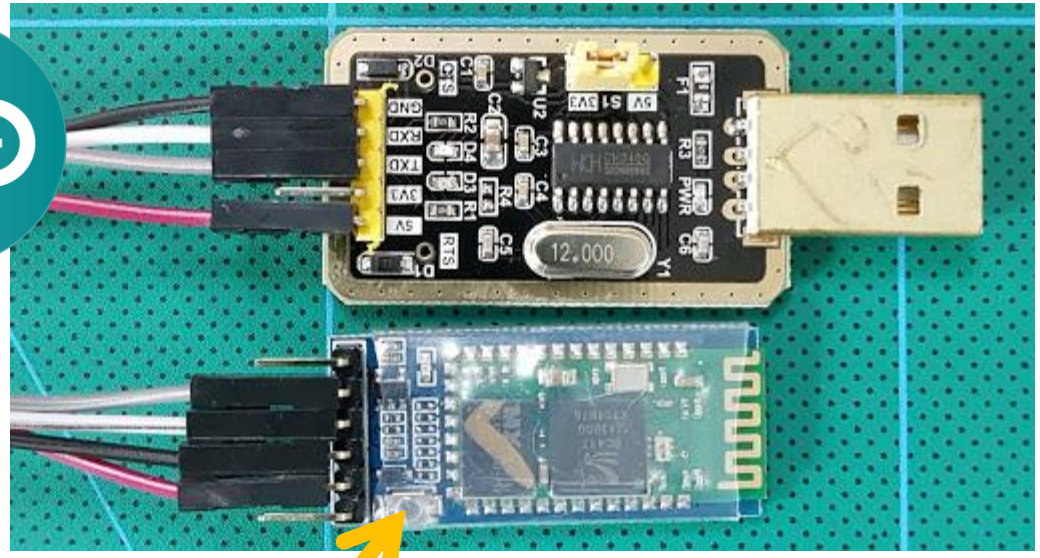
RX >>TX

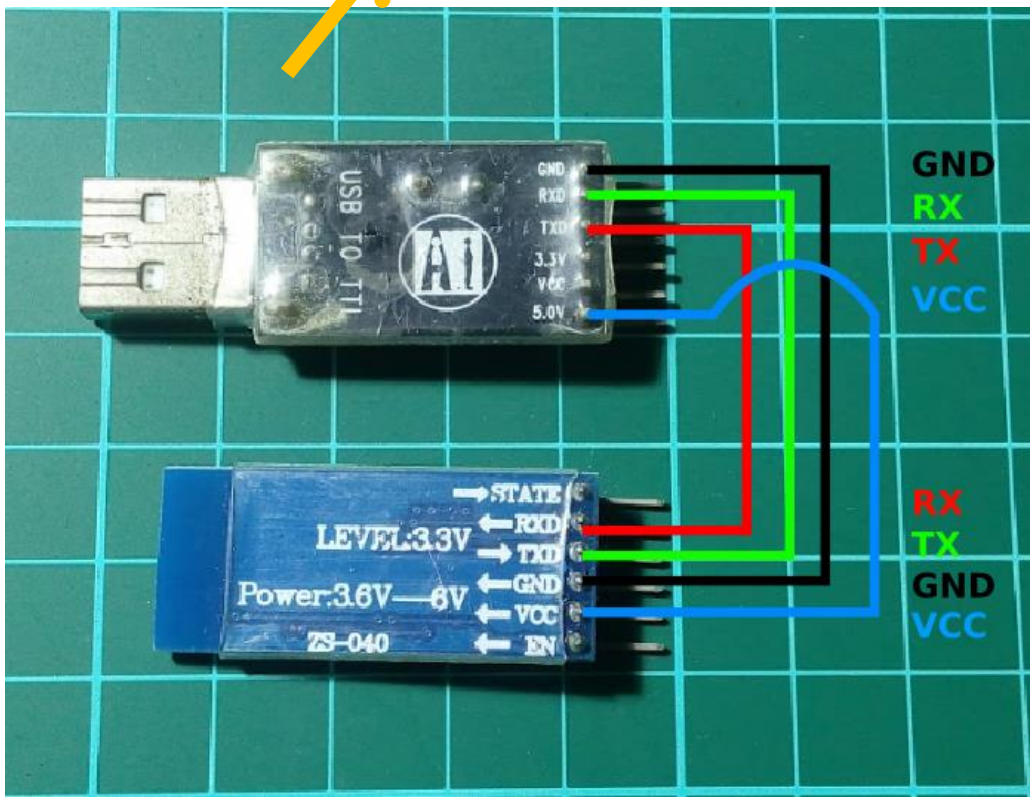


SEE TO IT THE WIRES COLOR CODE MATCHES THE MARKINGS

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





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 **(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

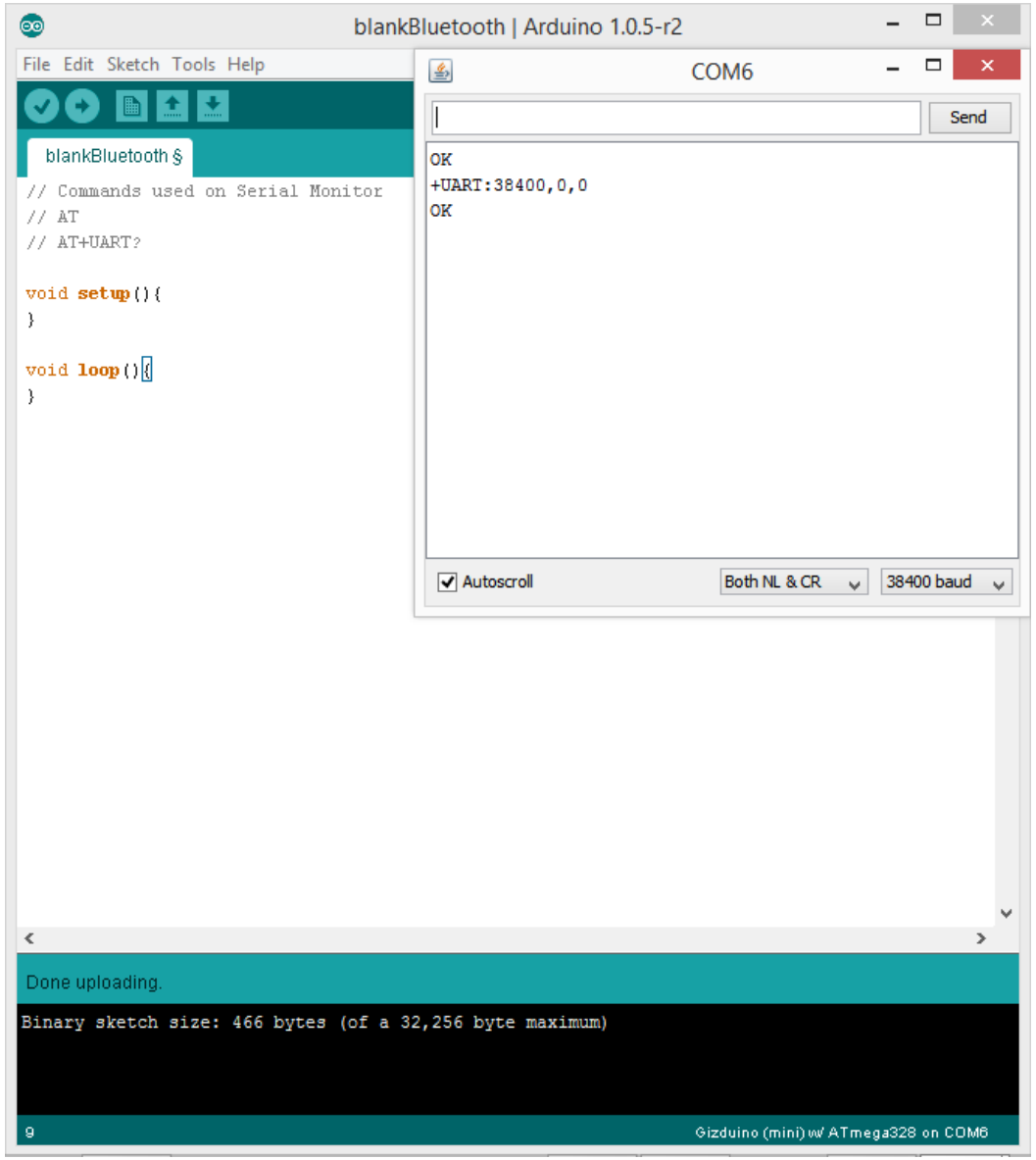
1 set to 1200bps
2 set to 2400bps
3 set to 4800bps
4 set to 9600bps (Default)
5 set to 19200bps
6 set to 38400bps
7 set to 57600bps
8 set to 115200bps

Bluetooth setup with the USB TTL and Arduino IDE

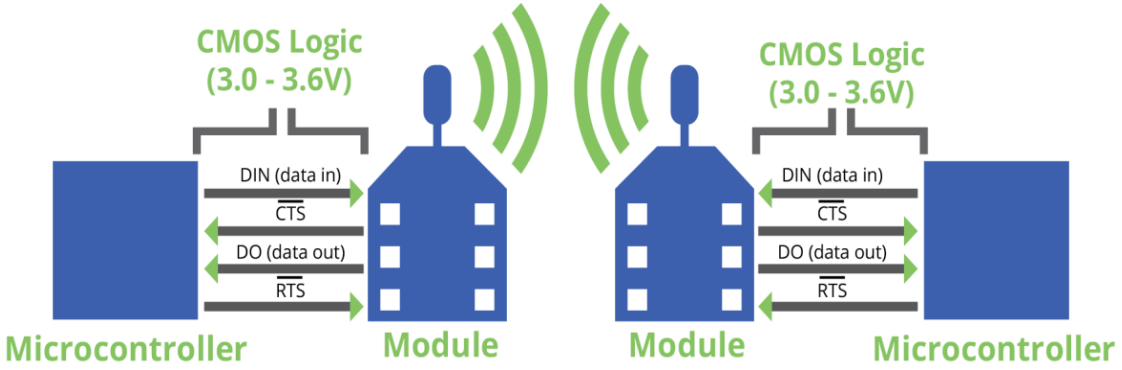
Arduino IDE>Tools>Serial Monitor (Push Button Before Connecting the USB) Set (Baud 38400) (Both NL & CR)

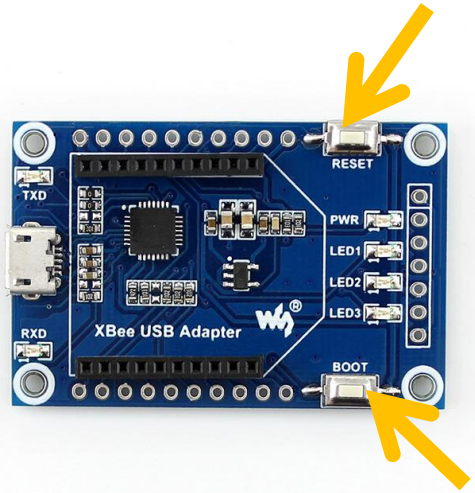
AT : check the connection
AT+VERSION : Check Version

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

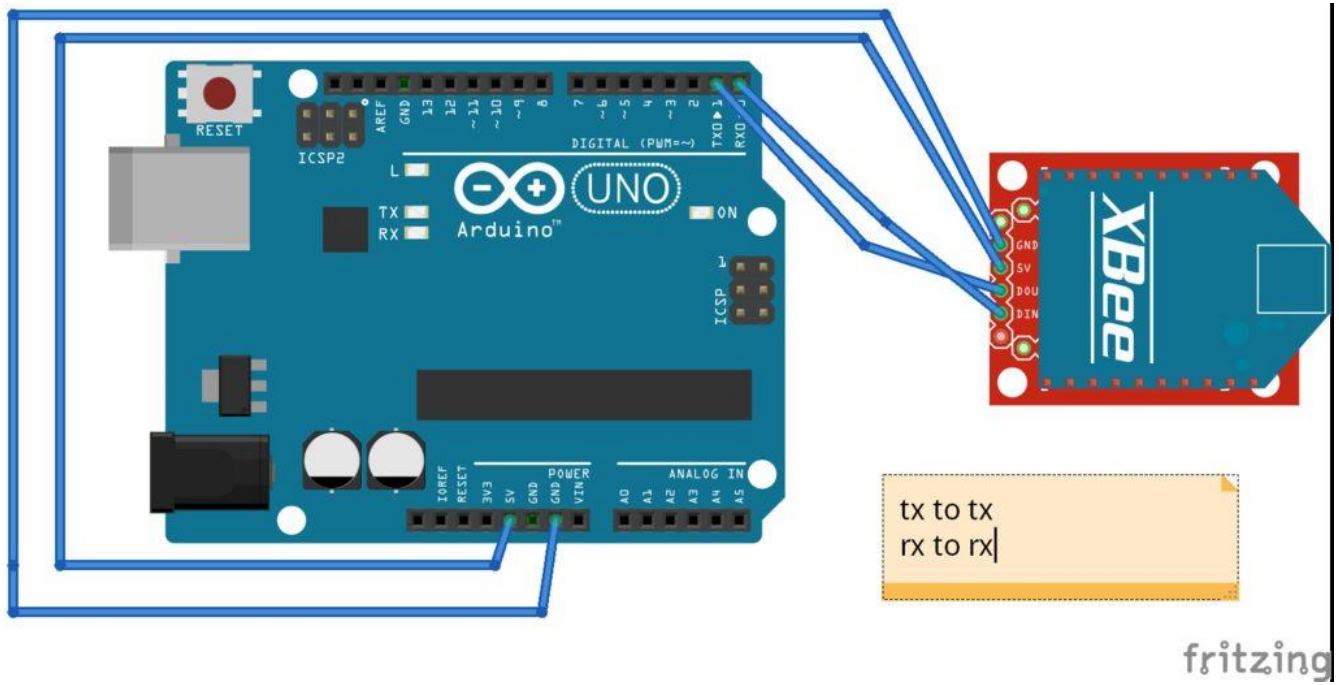


38400 FOR XBEE RADIO



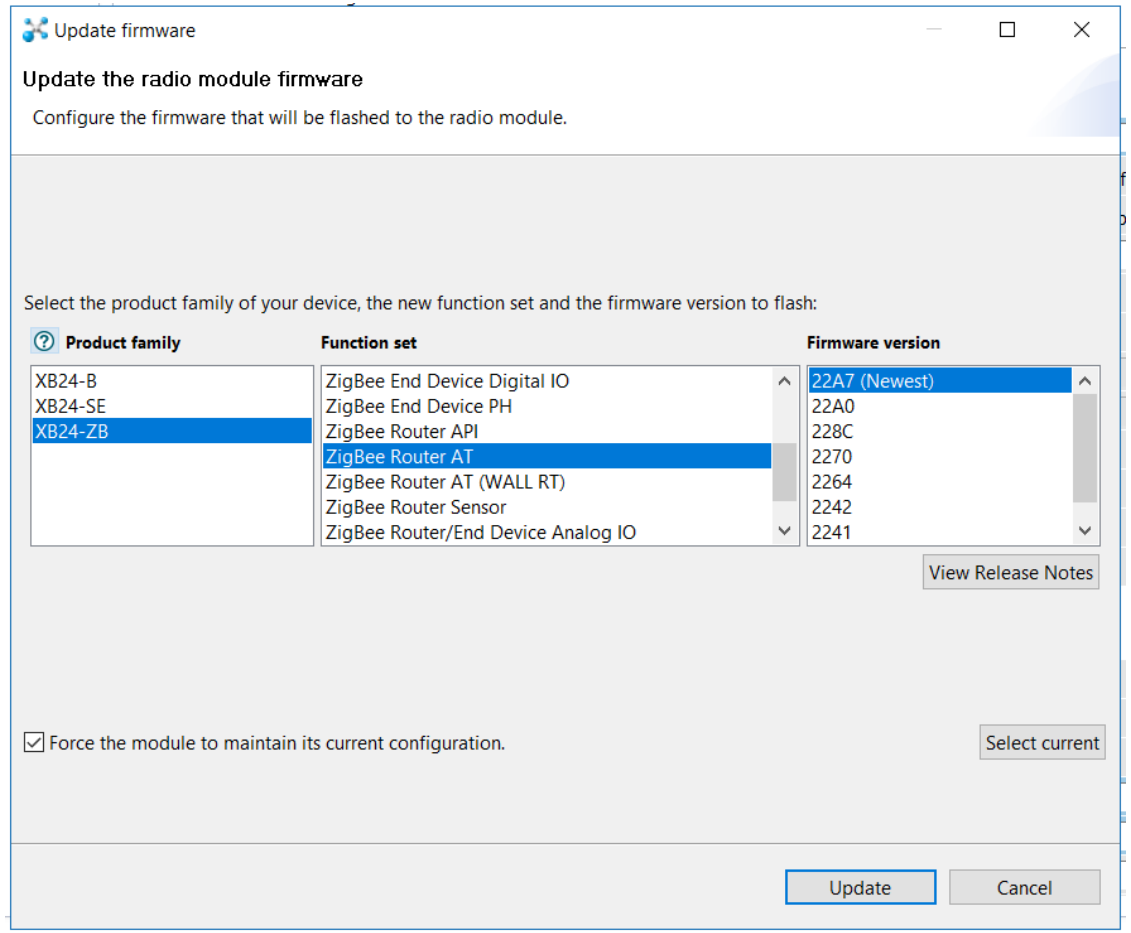


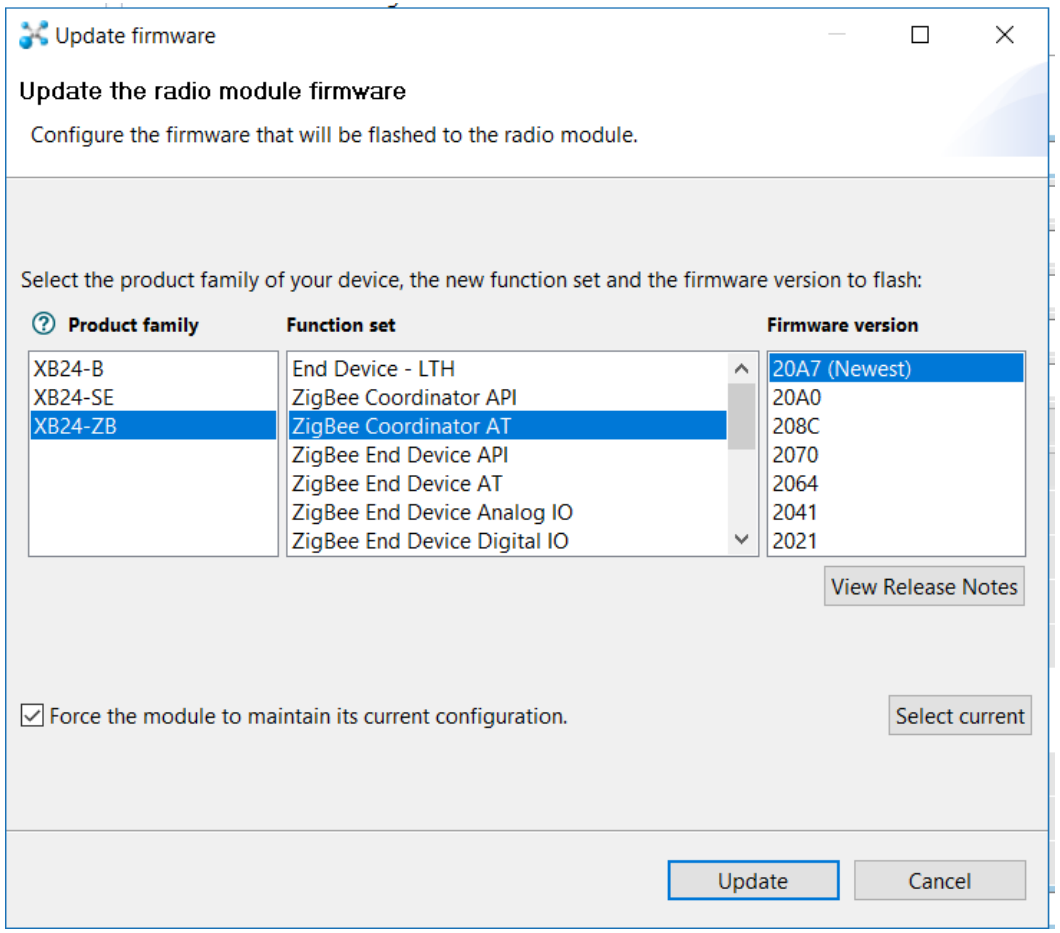
GET THE USB MODULE WITH BOOT AND RESET BUTTON AS YOU MAY NEED TO RESET THE XBEE WHEN UPDATING FIRMWARE



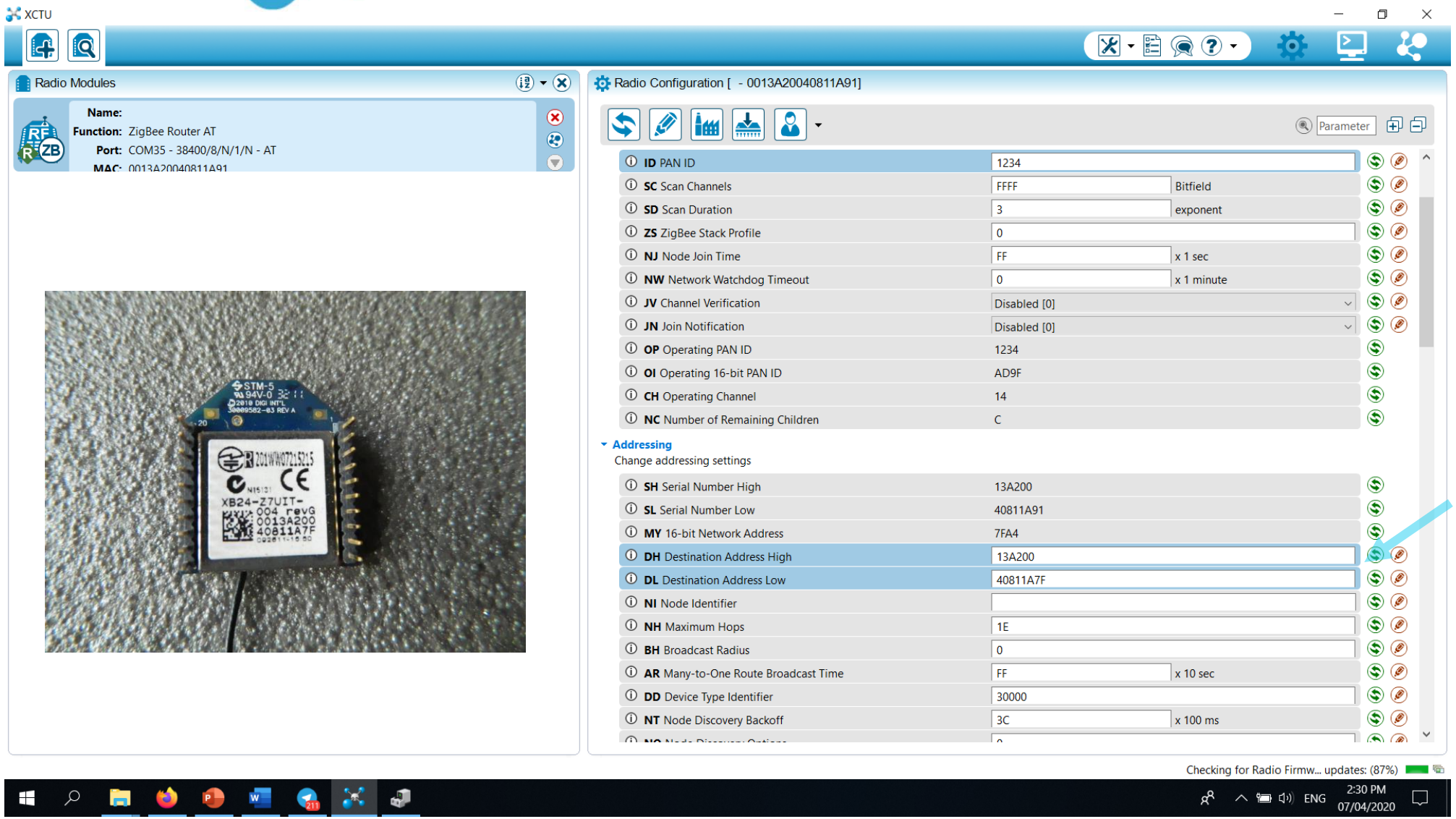
GROUND STATION ROUTER
38400 8/N/1/N - AT

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

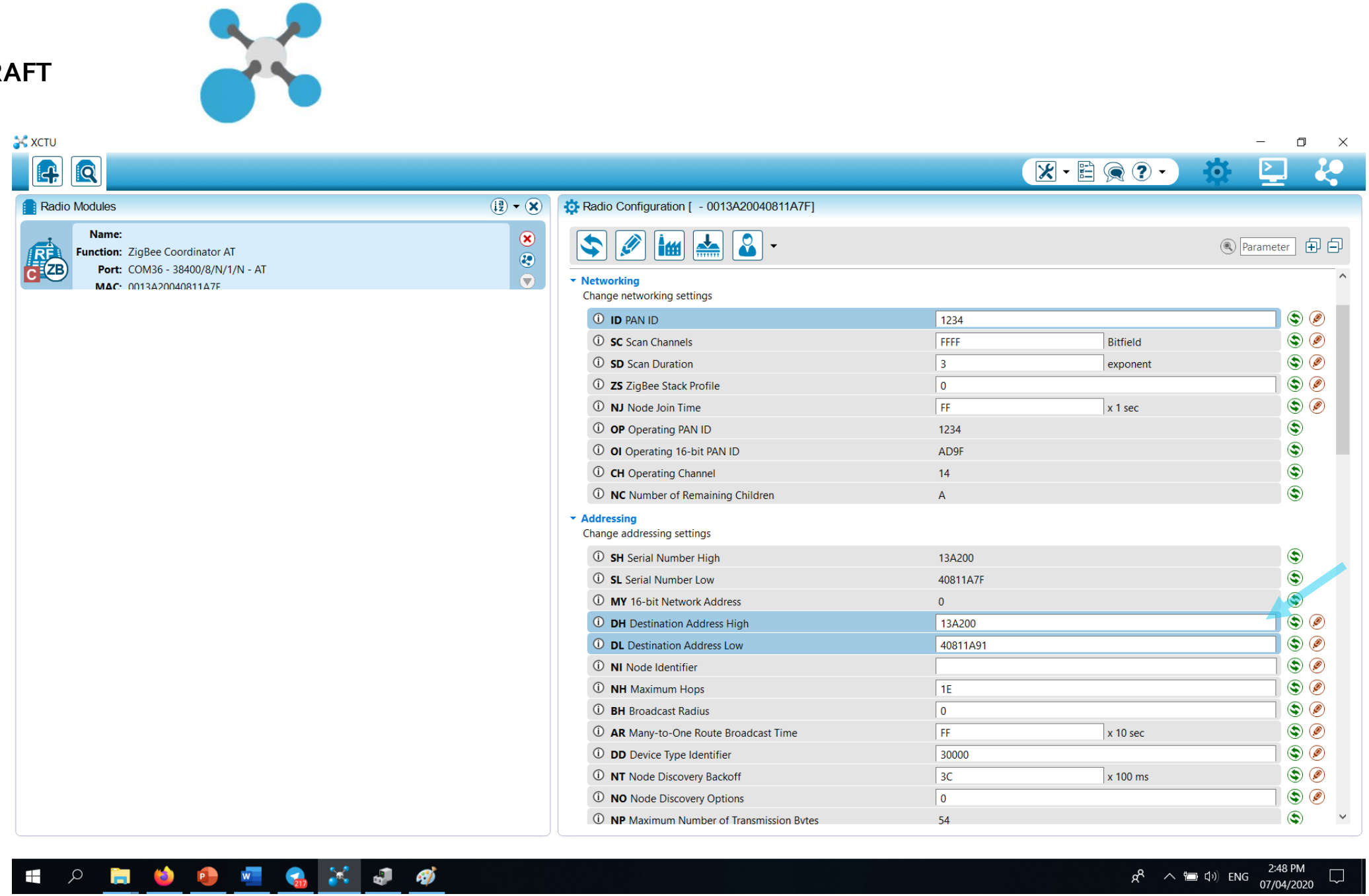
CT AT Command Mode Timeout	64	x 100ms
GT Guard Times	3E8	x 1ms
CC Command Sequence Character	2B	Recommended: 0x20-0x7F (ASCII)

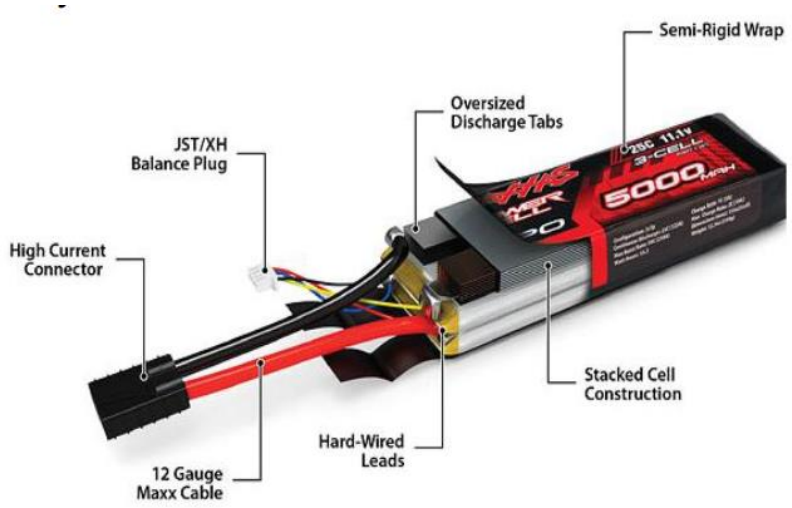
Sleep Modes
Configure low power options to support end device children

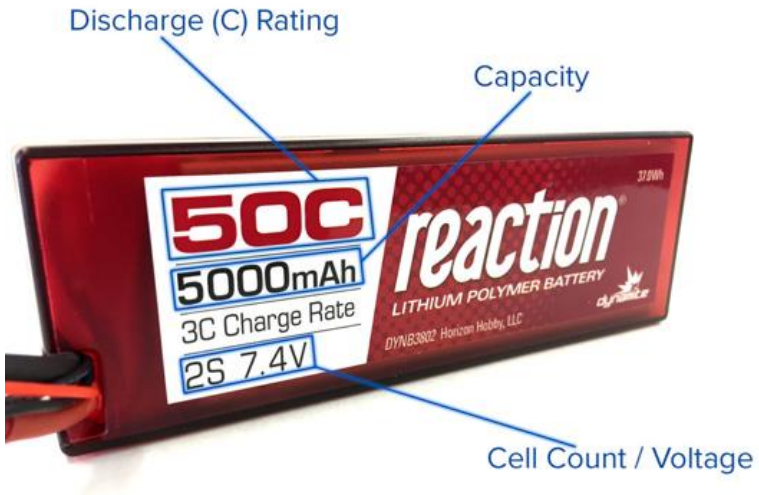
SM Sleep Mode	No Sleep (Router) [0]	
SN Number of Cyclic Sleep Periods	1	
SO Sleep Options	0	
SP Cyclic Sleep Period	20	x 10 ms
ST Time before Sleep	1388	x 1 ms

ENG 2:36 PM 07/04/2020

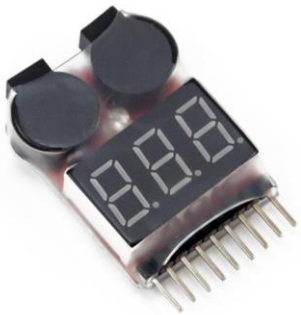
AIRCRAFT


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





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

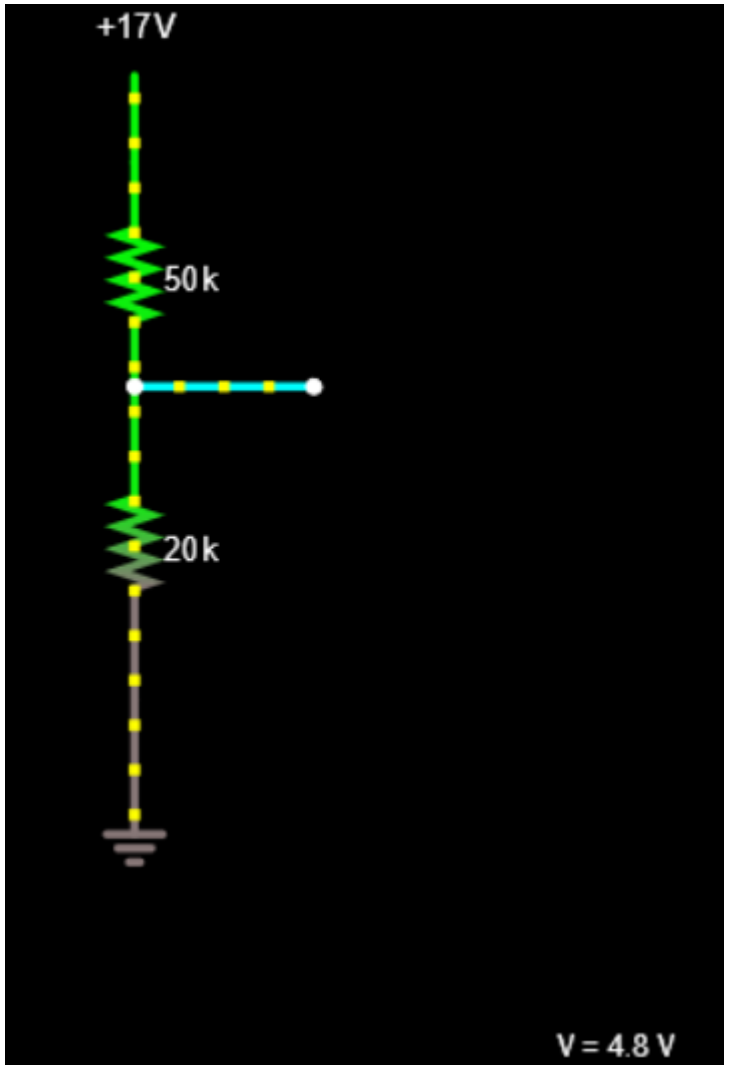
BATTERY STORAGE MODE IS 3.8V PER CELL

BATTERY DISCHARGE IS 3.6V PER CELL

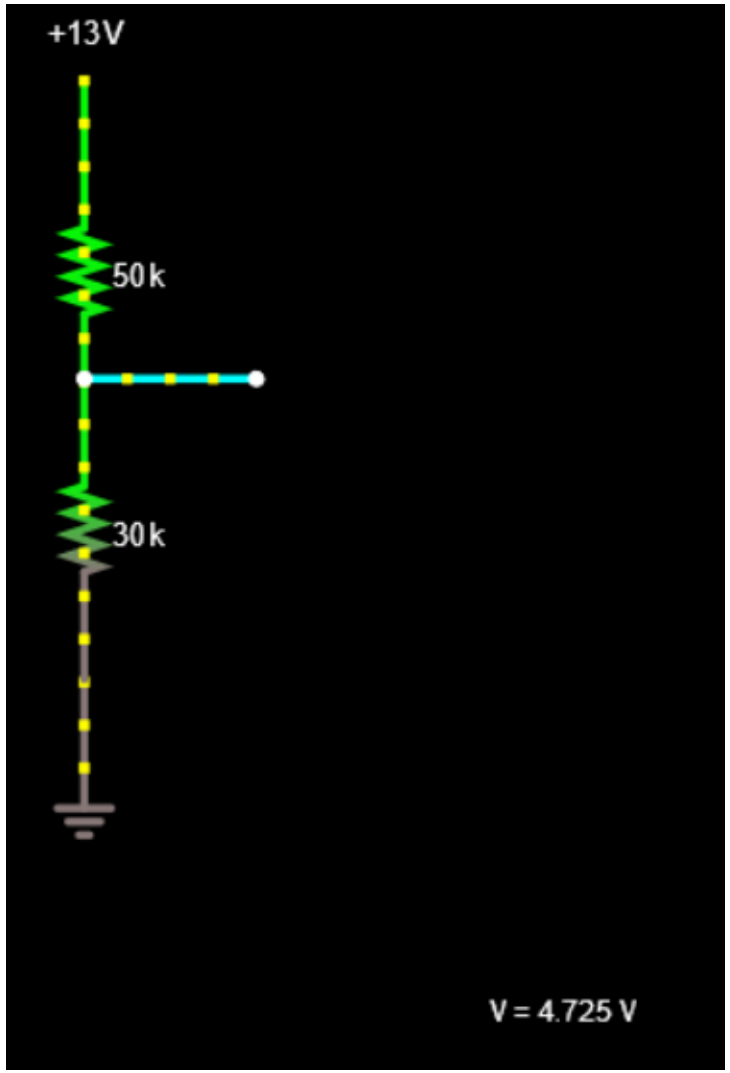
BATTERY FULL CHARGE IS 4.2V PER CELL

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

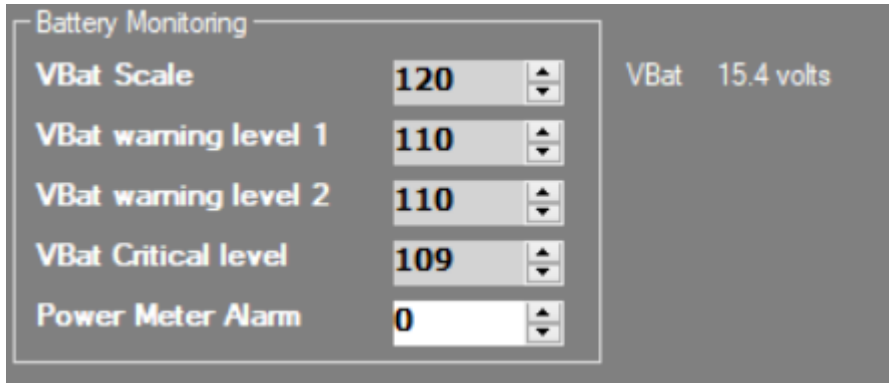
SWITCH THE VBAT JUMPER ACCORDINGLY TO THE BATTER CELLS YOUR USING

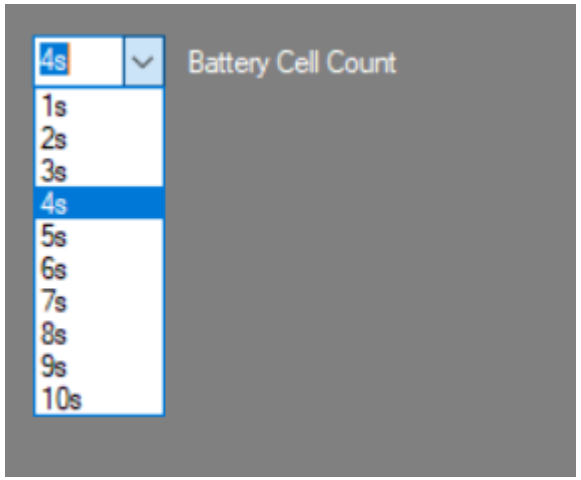
3S OR 4S

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
/**** 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 ! */
//#define 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
*/
//#define 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 COM41
ENG
10:24 PM
13/02/2020

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

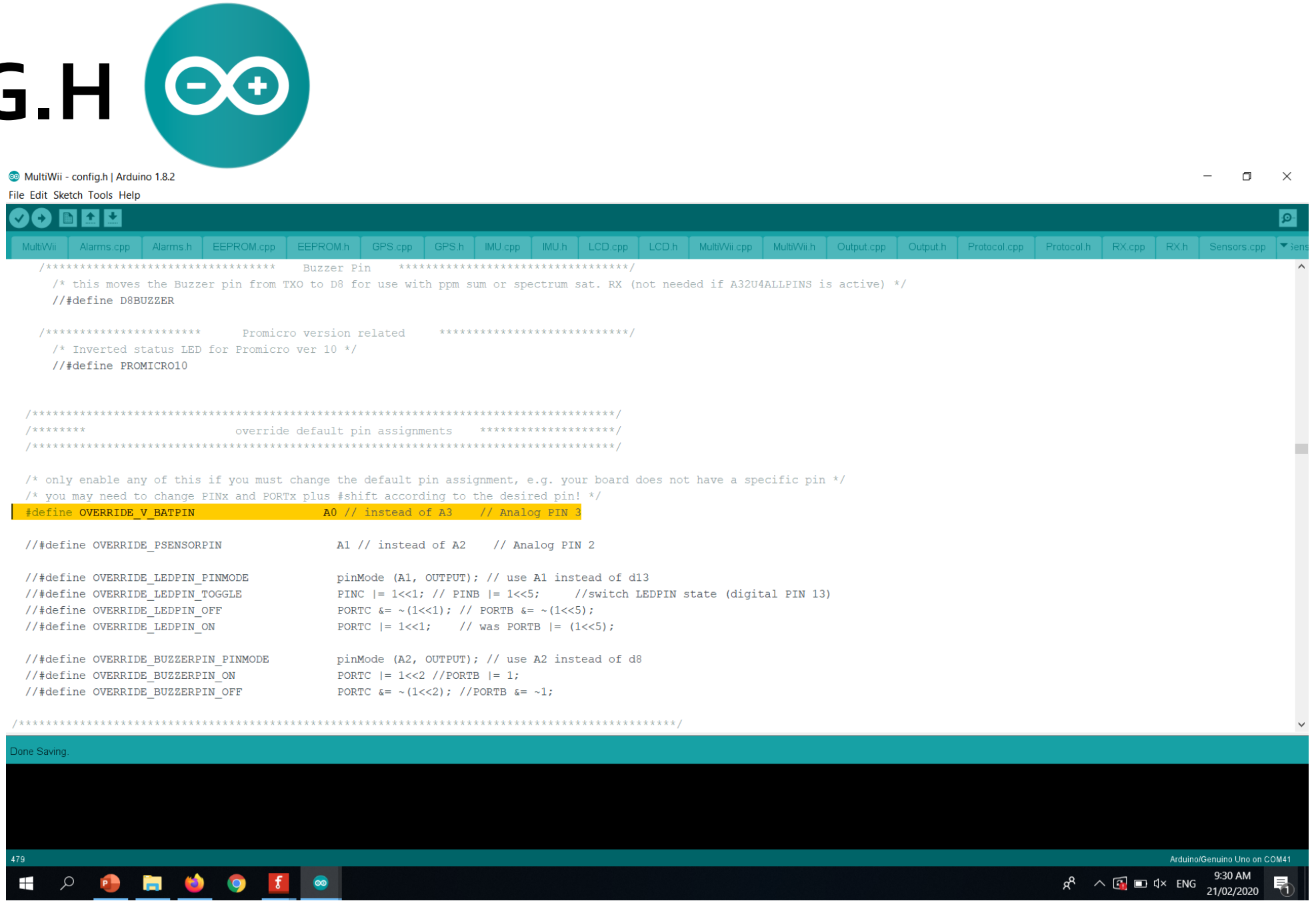
(GUI SETTINGS TAB)

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

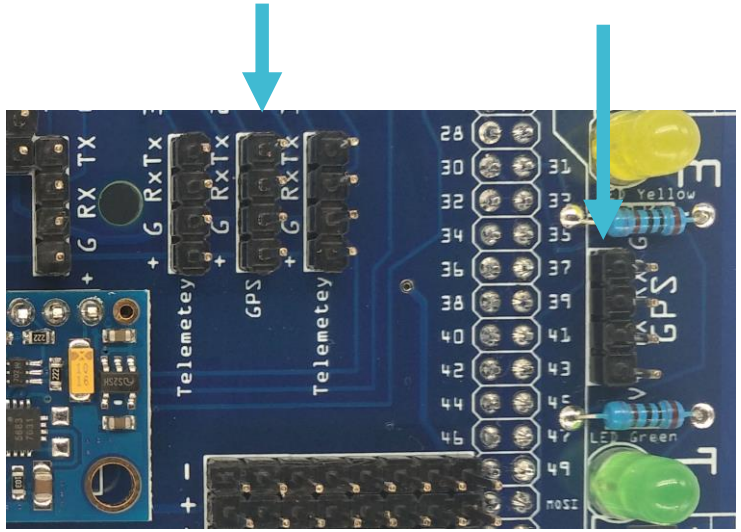
THIS BOARD SUPPORTS 2S-4S BATTERY

BATTERY

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
/********** Buzzer Pin **********/
/* 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
/********** Promicro version related **********/
/* 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
ENG
9:30 AM
21/02/2020

GPS



U BLOX NEO 6

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

ATTENTION:

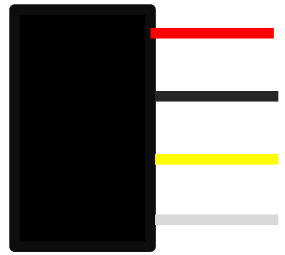
YOU MAY NEED TO REARRANGE THE HEADERS TO CONNECT THE GPS MODULE TO THE SHIELD BOARD ACCORDINGLY

VCC >> +

GND >> G

TX >> RX

RX >>TX



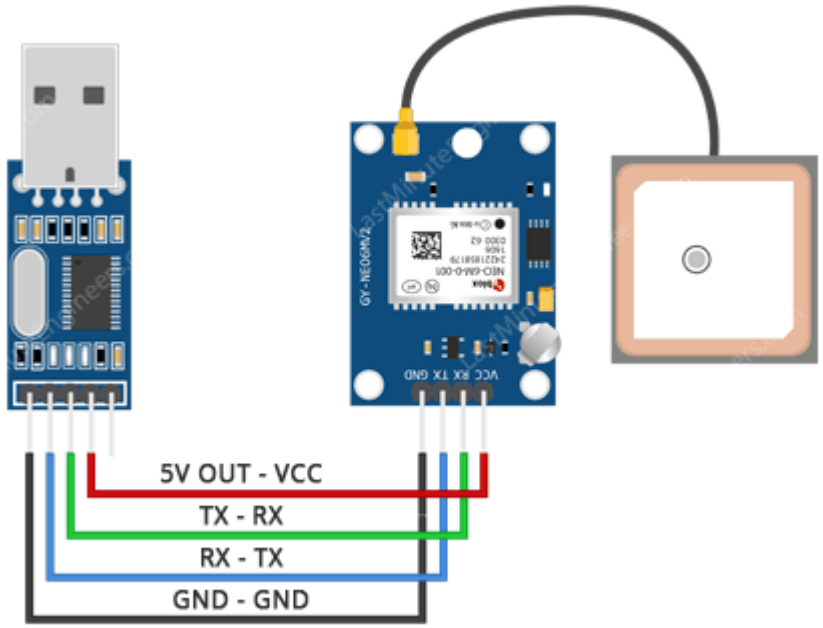
SEE TO IT THE WIRES COLOR CODE MATCHES THE MARKINGS

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 CONFIGURING



U BLOX NEO 6

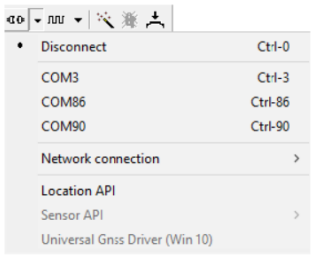
PLUG IN TO SERIAL TX 2 RX 2

USB TTL TO PROGRAM THE GPS

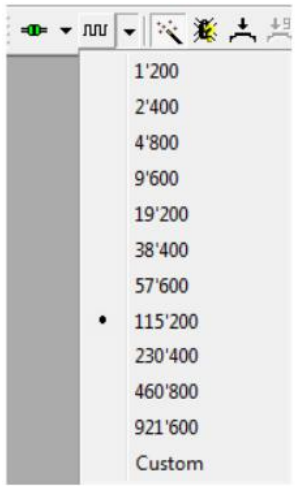
THIS GOES SAME ON THE DRONE SHIELD

U CENTER

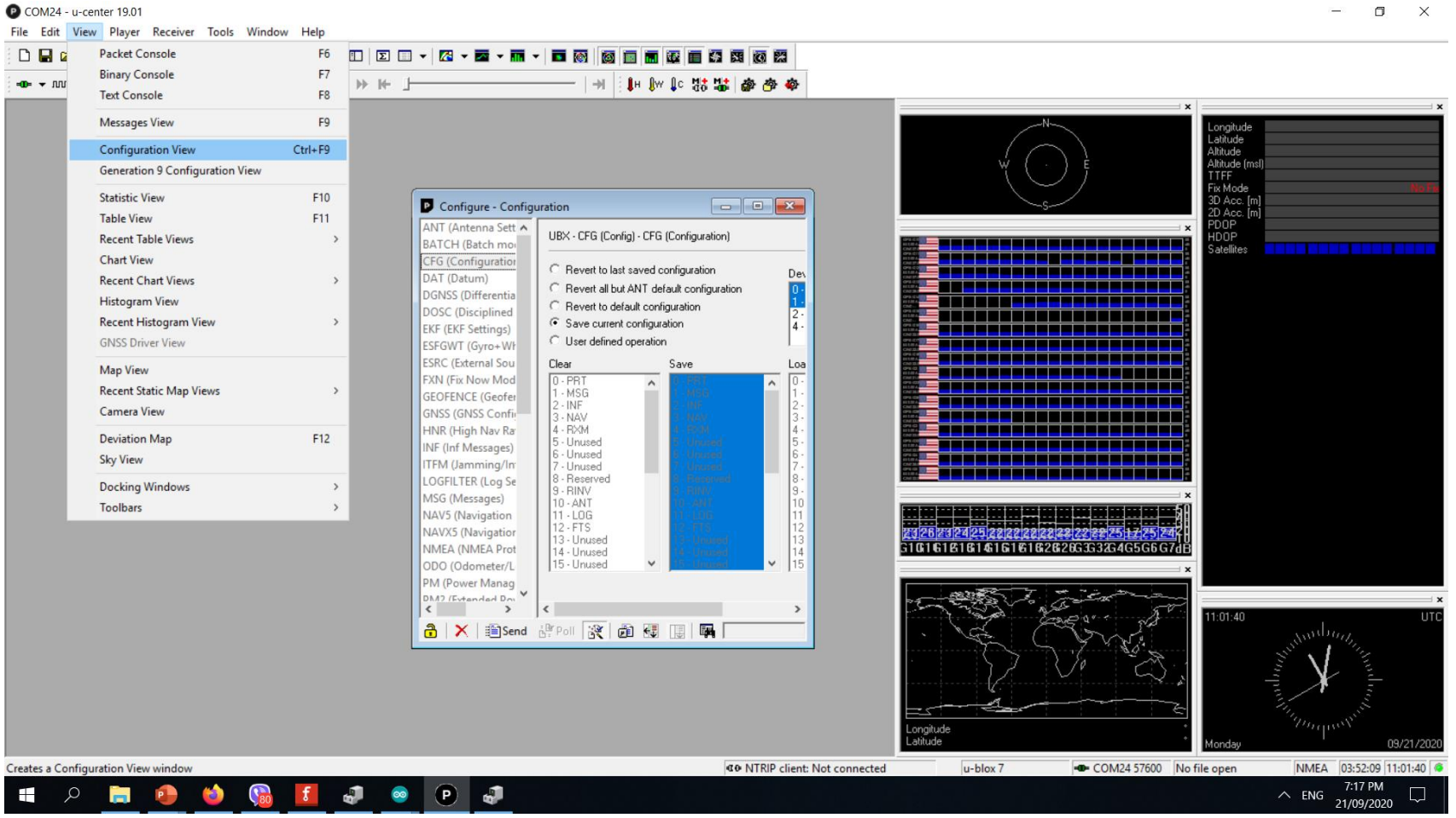
Connect to the device. Select the COM port your GPS and USBTTL is connected to



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

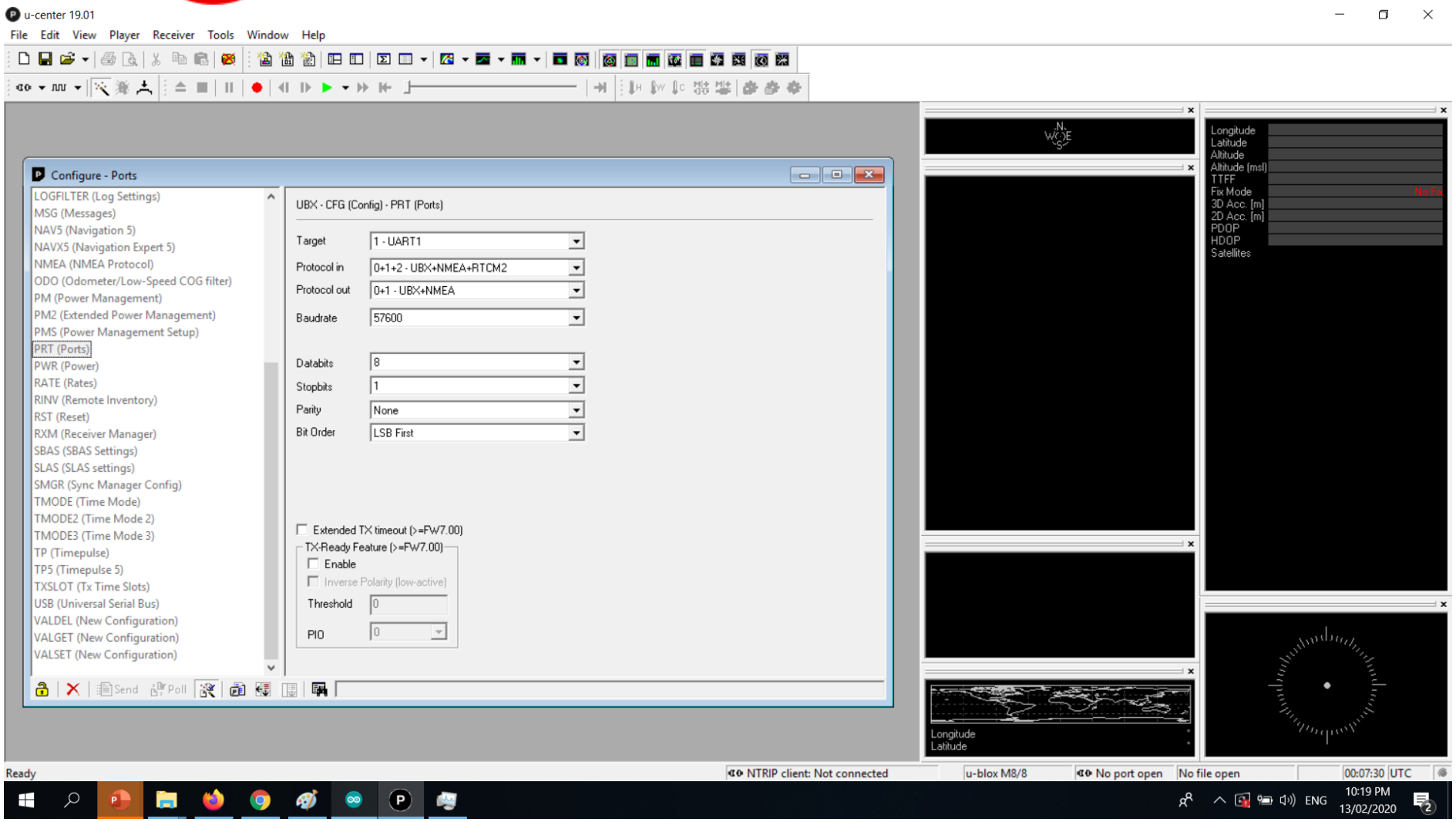


Go to View> Configuration View



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. Send the message to the receiver

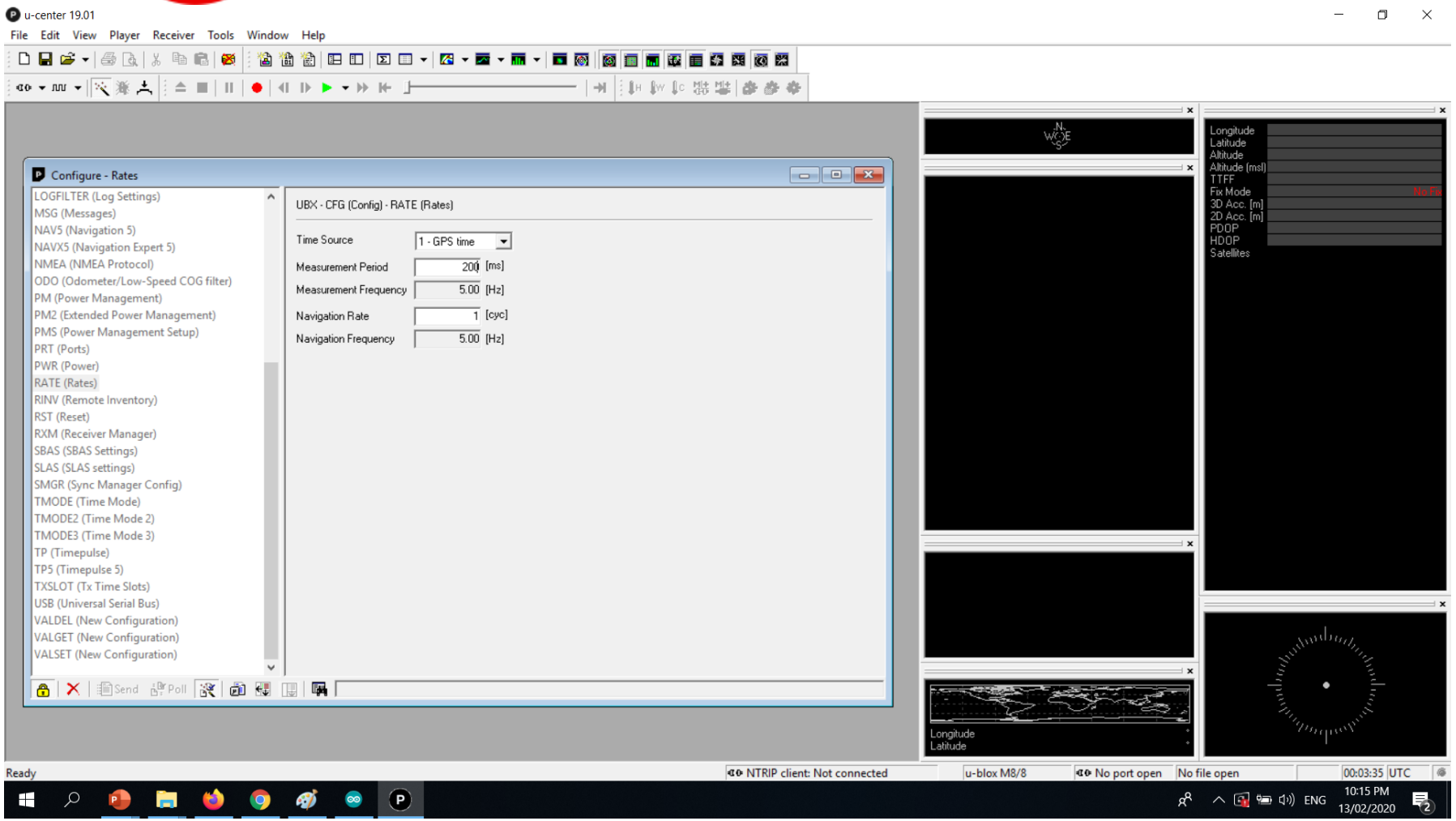


U CENTER

Rate 200ms
Frequency 5hz

Note after updating the settings and hitting the send button below

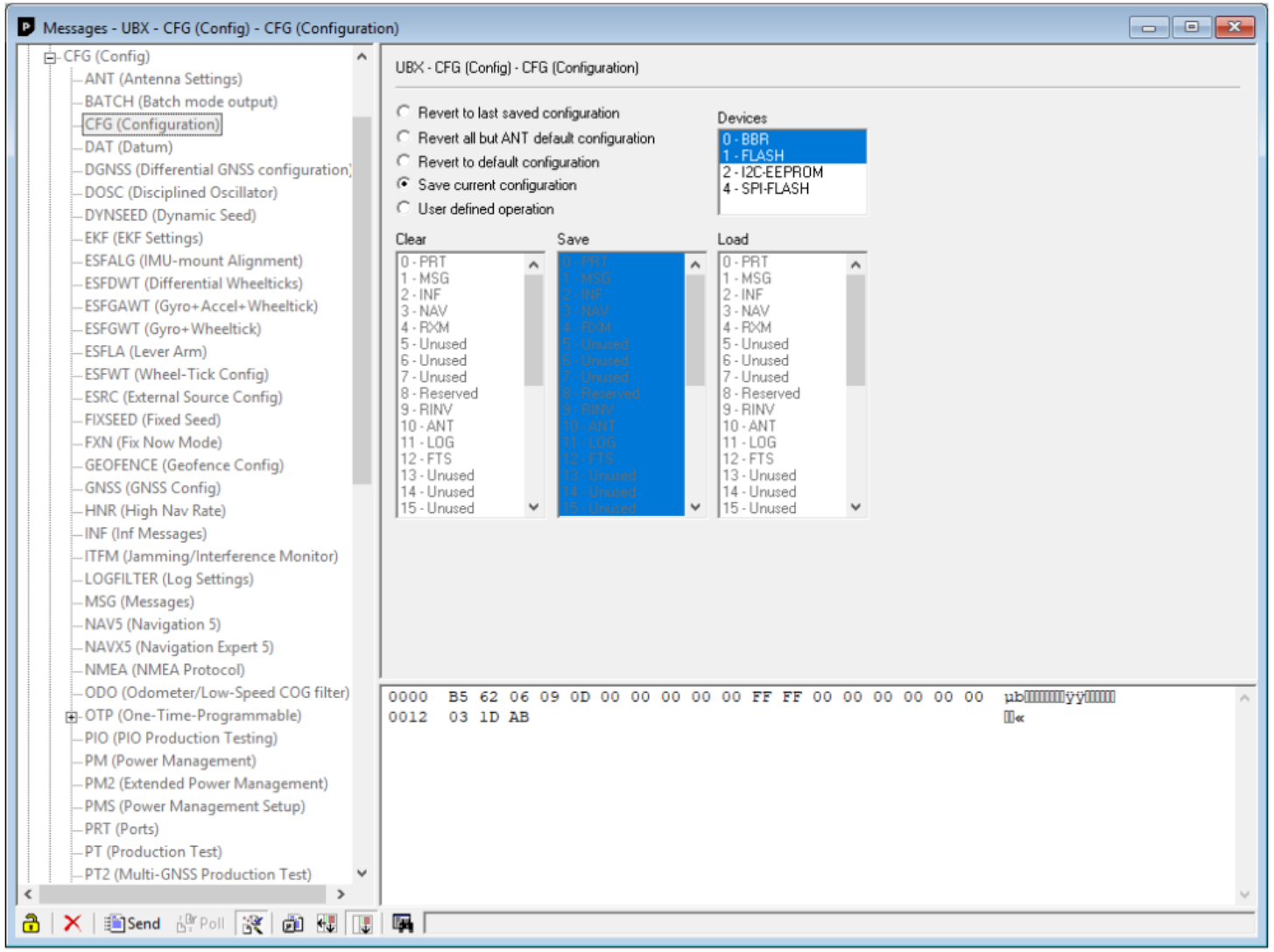
Exit the configuration window this will promp you to the save parameters . Hit “Yes”



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



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
/* 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
GPS
/* 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
ENG
4:05 PM
20/02/2020

CAMERA GIMBAL

CONFIG.H

Photo for illustration purpose only.

MultiWii - config.h | Arduino 1.8.2

File Edit Sketch Tools Help

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

```
  */
  //#define DISABLE_SERVOS_WHEN_UNARMED


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

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

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

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

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

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

259

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

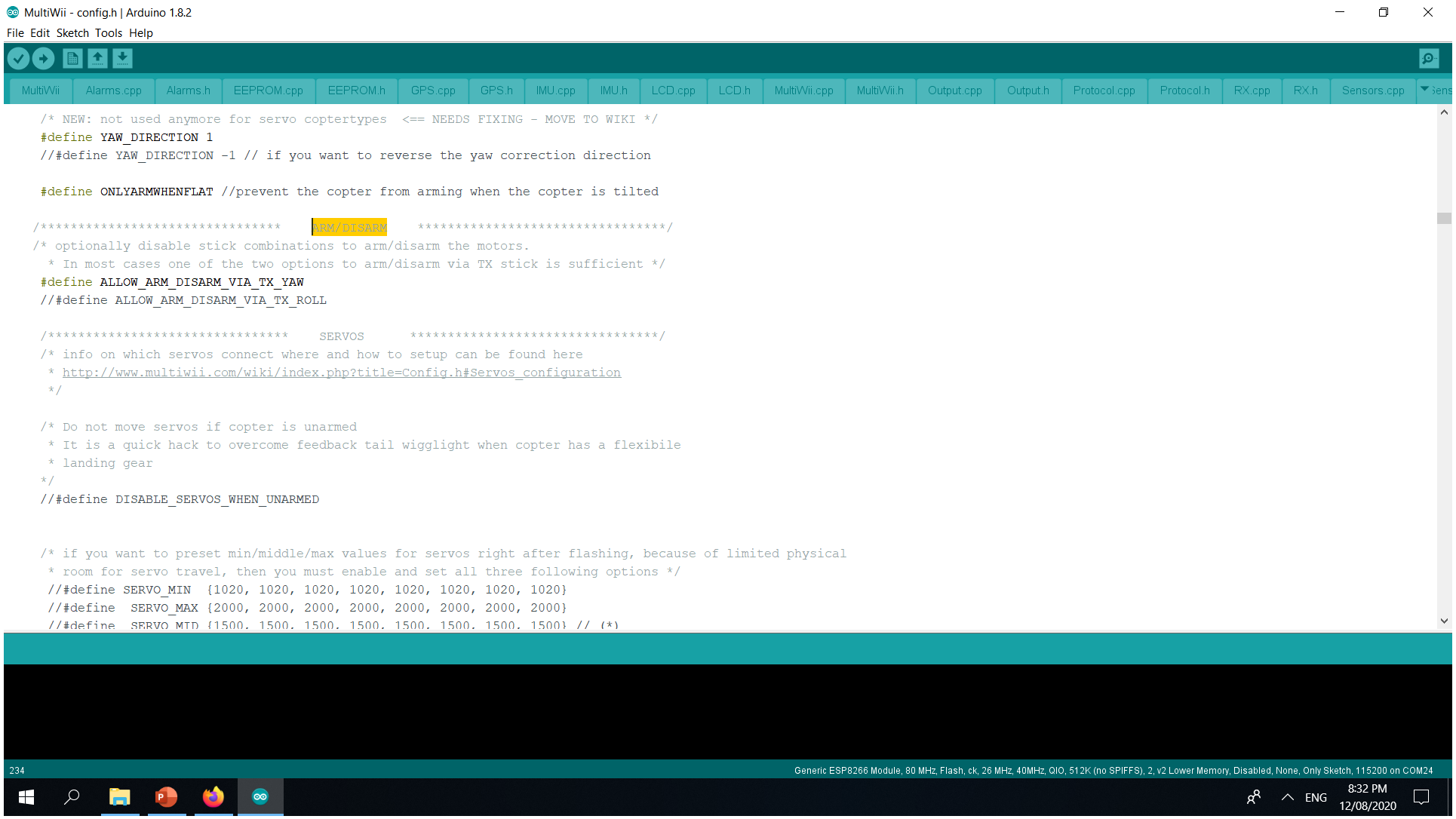
ENG 6:16 PM 11/08/2020

CONFIG.H

Arm/DisArm

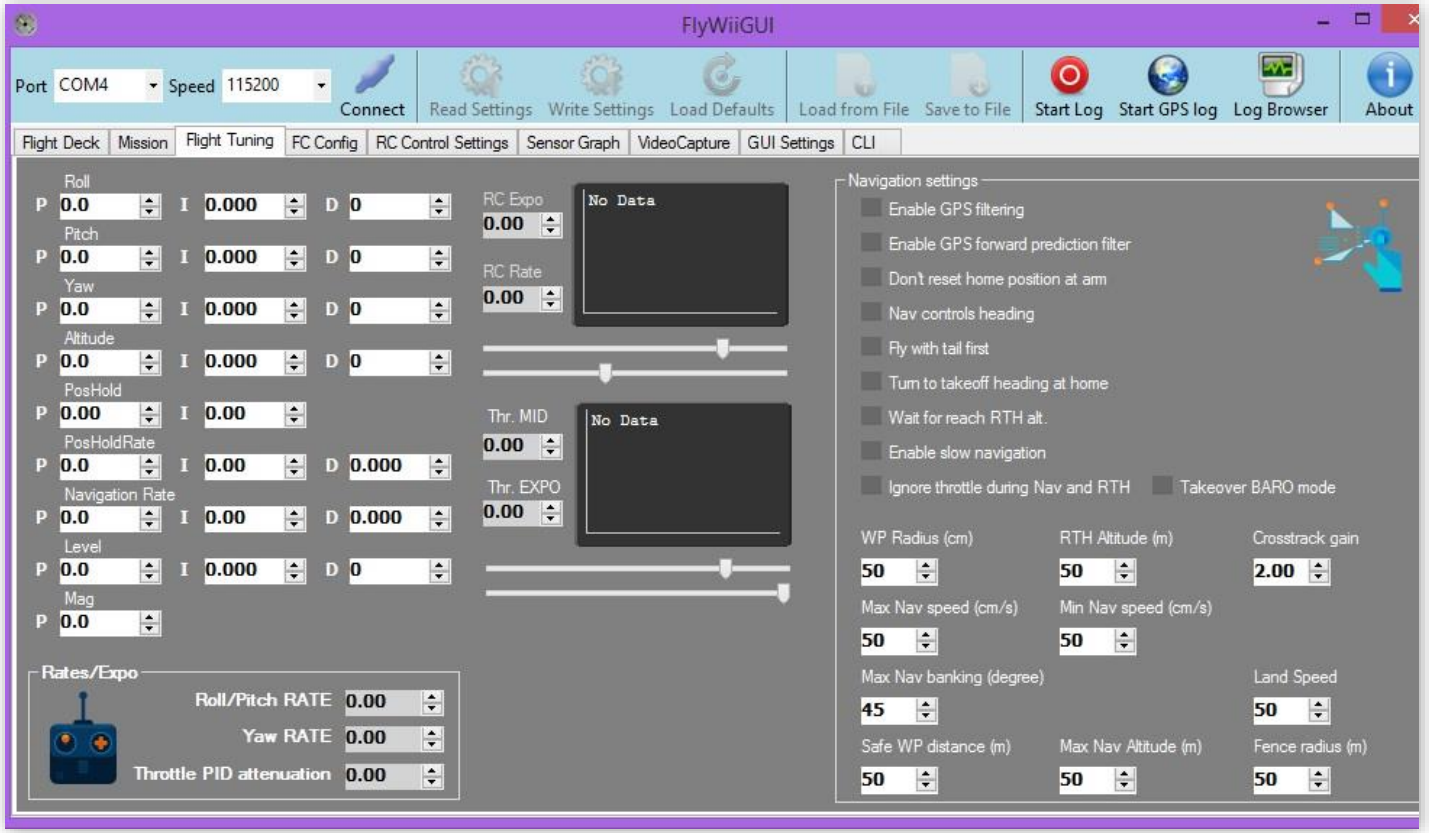
Option for combination stick command to start and stop the drone

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



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

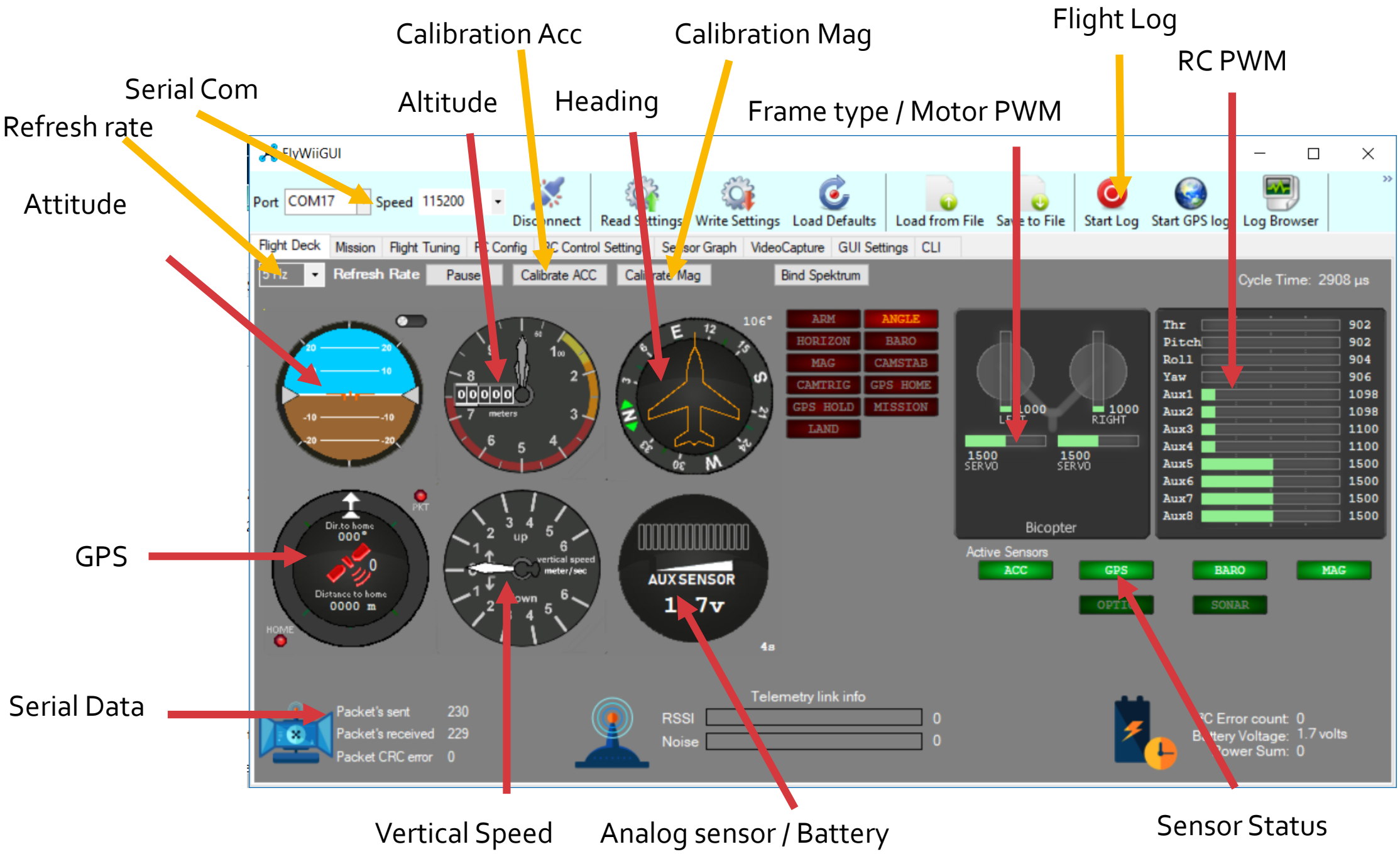
FLYWII GUI



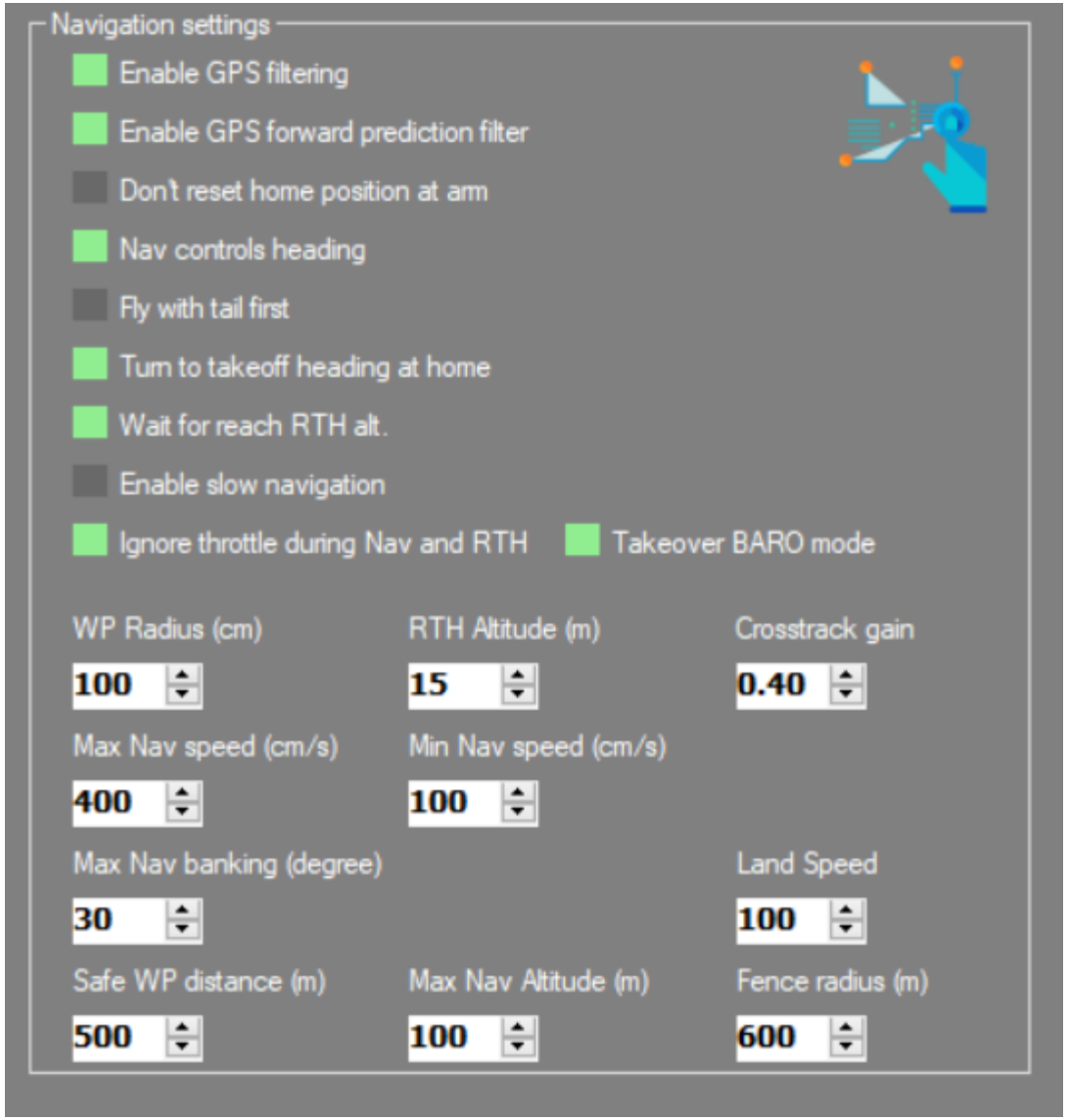
DEDICATED USER INTERFACE AS CONFIGURATOR AND GROUND STATION FOR THE ARDUINO DRONE

For our Arduino Drone we provide the FlywiiGUI just for this

http://synerflight.com/flywiigui/


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
FlyWiiGUI
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
PID Config
RC Control Setting
Sensor Graph
VideoCapture
GUI Settings
CLI
Refresh Rate
Pause
Calibrate ACC
Calibrate Mag
Bind Spektrum
Cycle Time: 2908 µs
ARM
ANGLE
HORIZON
BARO
MAG
CAMSTAB
CAMTRIG
GPS HOME
GPS HOLD
MISSION
LAND
meters
Dir.to home
Distance to home
vertical speed
meter/sec
AUX SENSOR
Thr 902
Pitch 902
Roll 904
Yaw 906
Aux1 1098
Aux2 1098
Aux3 1100
Aux4 1100
Aux5 1500
Aux6 1500
Aux7 1500
Aux8 1500
1000 RIGHT
1500 SERVO
Bicopter
Active Sensors
ACC
GPS
BARO
MAG
SONAR
Packet's sent 230
Packet's received 229
Packet CRC error 0
Telemetry link info
RSSI 0
Noise 0

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

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

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

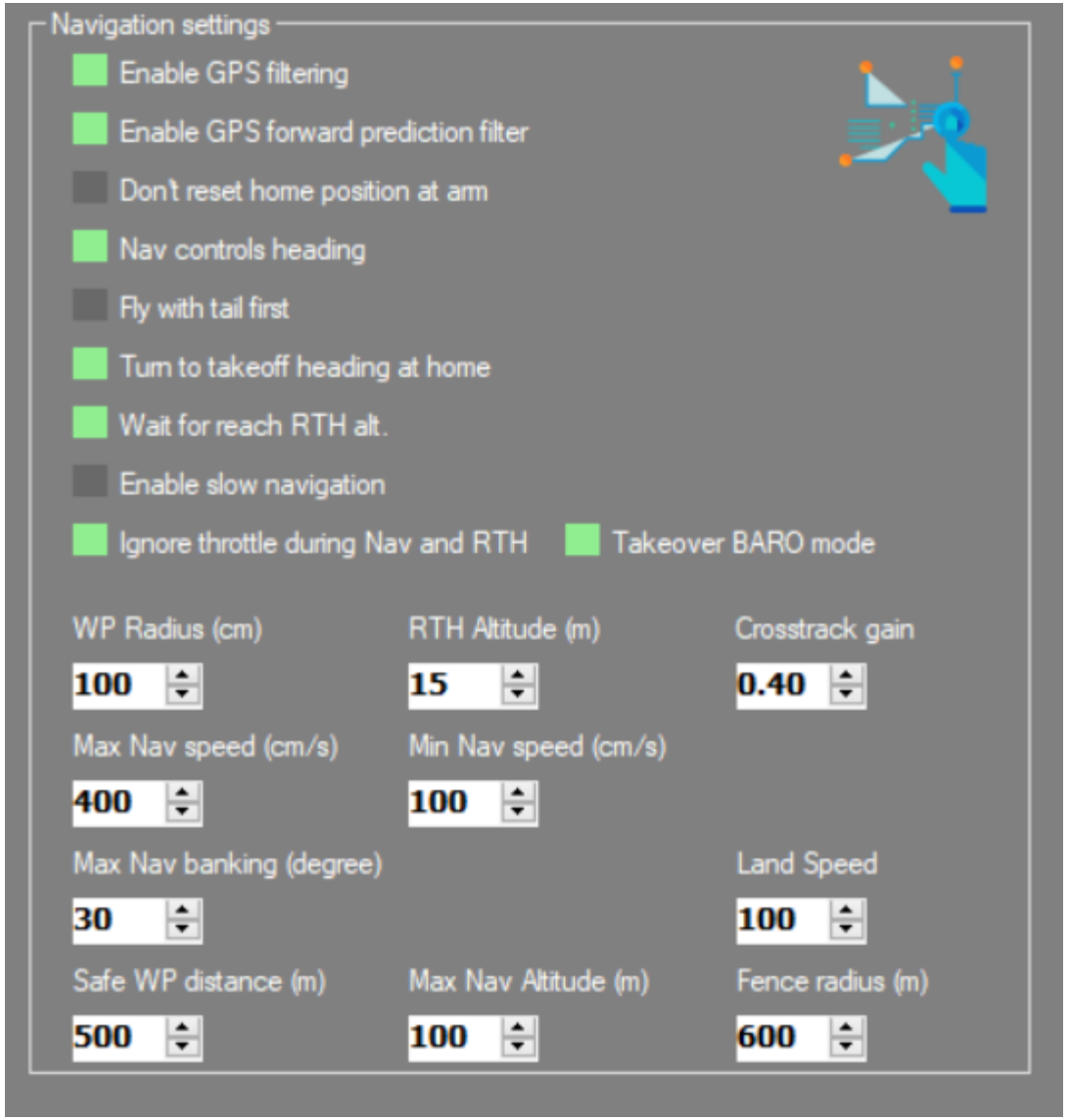
Safe WP Distance – max distance between waypoint before its null out

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 Vehicle's location and to compensate for lag . (optional) – not necessary for most application



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

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

Fly tail first – makes the Vehicle drive reverse

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

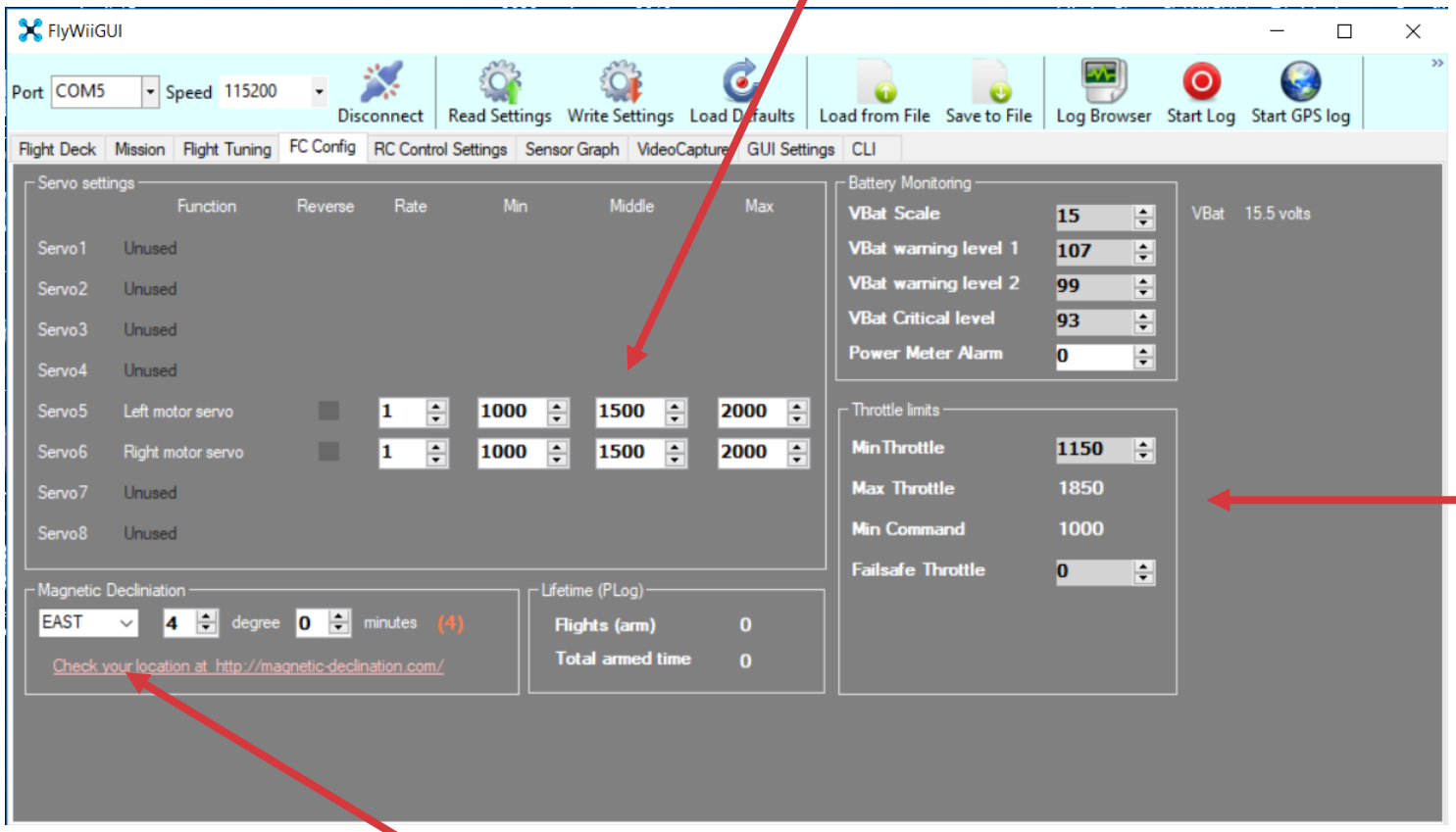
Enable slow navigation – this works with keeping the drone to its **Min Nav speed**

Ignore throttle and Take over Baro – Not applicable

Other Navigation Functions

FC Config

SERVO REVERSE OPTION – IF THE SERVO / REVERSABLE ESCS IS OPERATING IN THE WRONG DIRECTION FROM CONTROLS



MOTOR THROTTLE RANGE PWM TO THE MOTOR

THIS ALSO CONTROLS THE MOTOR IDLE SPEED ON ARM

IMPORTANT TO KNOW THE MAGNETIC DECLINATION OF YOUR REGION

THIS AID ANY AUTONOMOUS FUNCTION THAT REQUIRES COMPASS

- HEADING HOLD
- GPS HOLD
- RTH
- MISSION

CALIBRATE COMPASS AT THE FLIGHT DECK TAB AFTER SETTING THIS UP

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 Vehicle is controlled by Pos Rate.

- When the Vehicle 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 multirotor is controlled by Pos Rate.

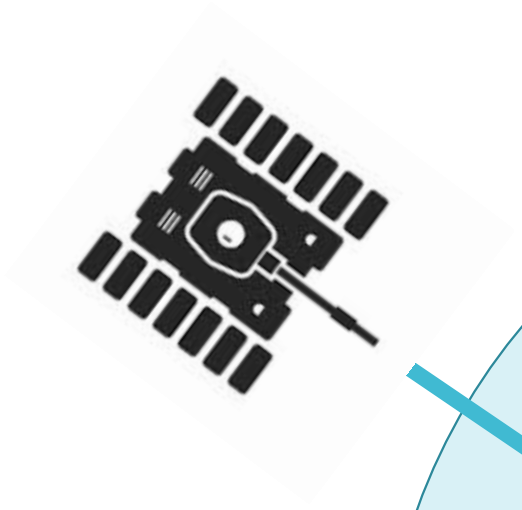
- When the Vehicle is within the defined distance of the hold location or waypoint (set by GPS_WP_RADIUS in config.h) the Pos Rate PID is used, when further away from the location the Nav Rate PID is used to return to within the defined waypoint radius.

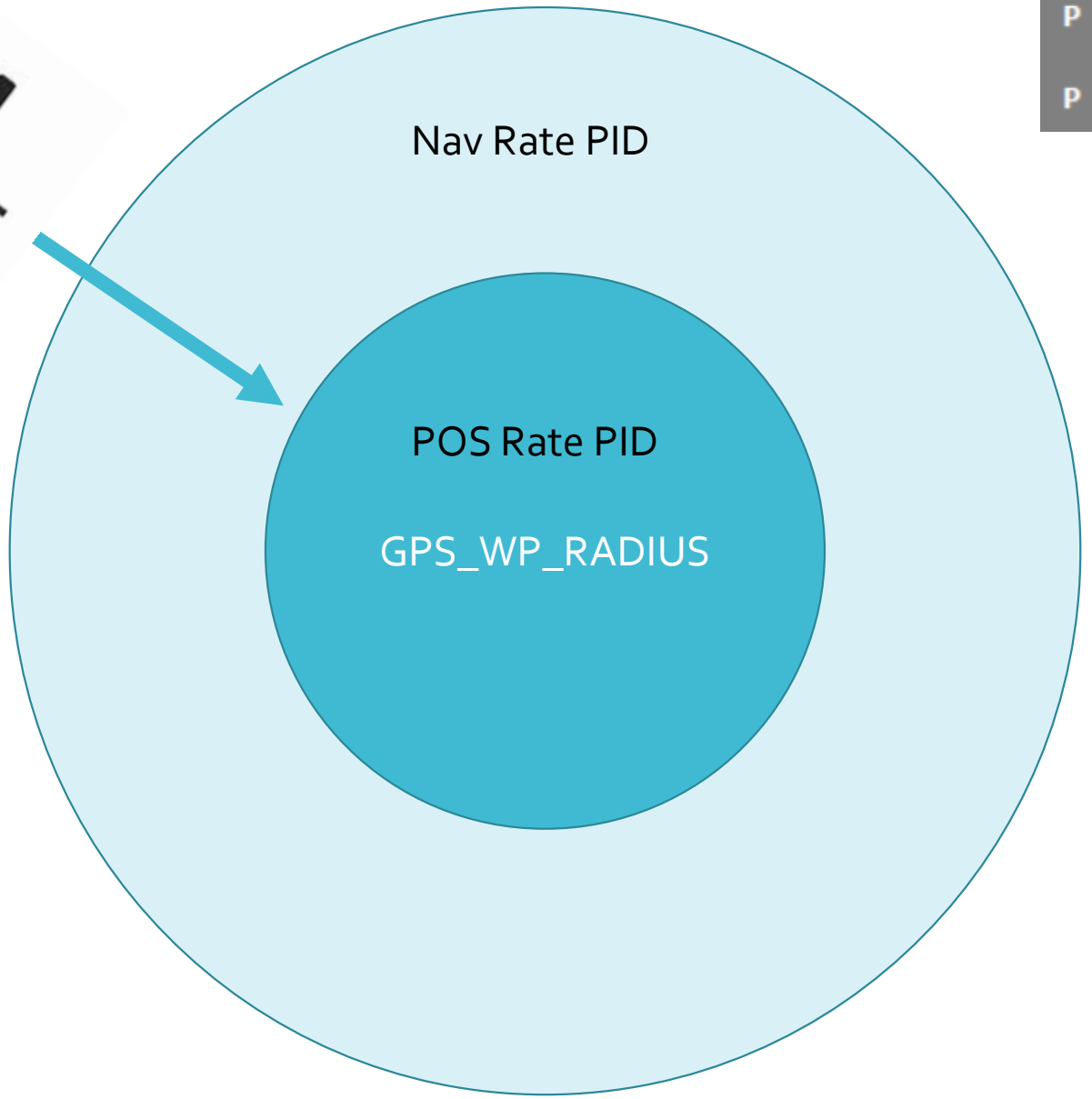
- To tune the Pos Rate PID, initially set P, I and D values to 0.

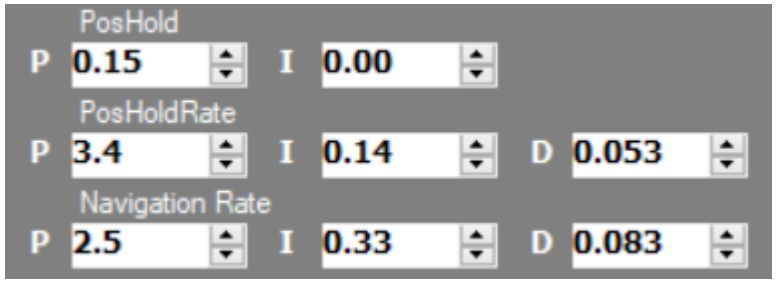
- Gradually increase P until the Vehicle begins to position hold with some drift.

- Gradually increase D until the Vehicle 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.

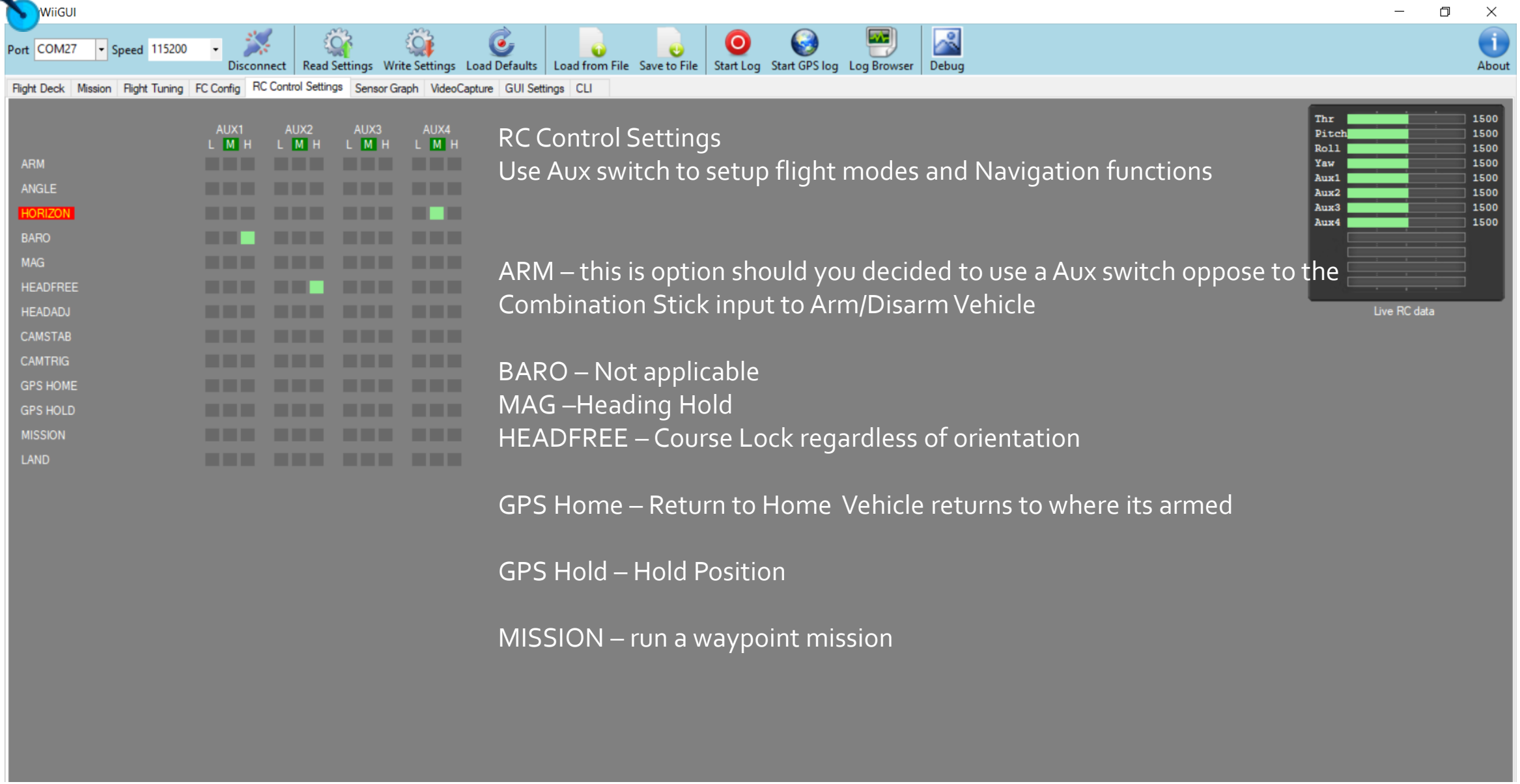
- If needed, gradually increase I value to allow the PID controller to compensate for long lasting error, ie if it is being blown by the wind.







The Navigation responds is the vehicle will orientate to the direction of the waypoint
It will proceed to drive straight until it reach the WP_Radius and stop
With multiple waypoint it will repeat the process till it completes the mission



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

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

BARO – Not applicable
MAG –Heading Hold
HEADFREE – Course Lock regardless of orientation

GPS Home – Return to Home Vehicle returns to where its armed

GPS Hold – Hold Position

MISSION – run a waypoint mission

- 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

Missions

Note: Only functional for Mega 2560 Boards with GPS

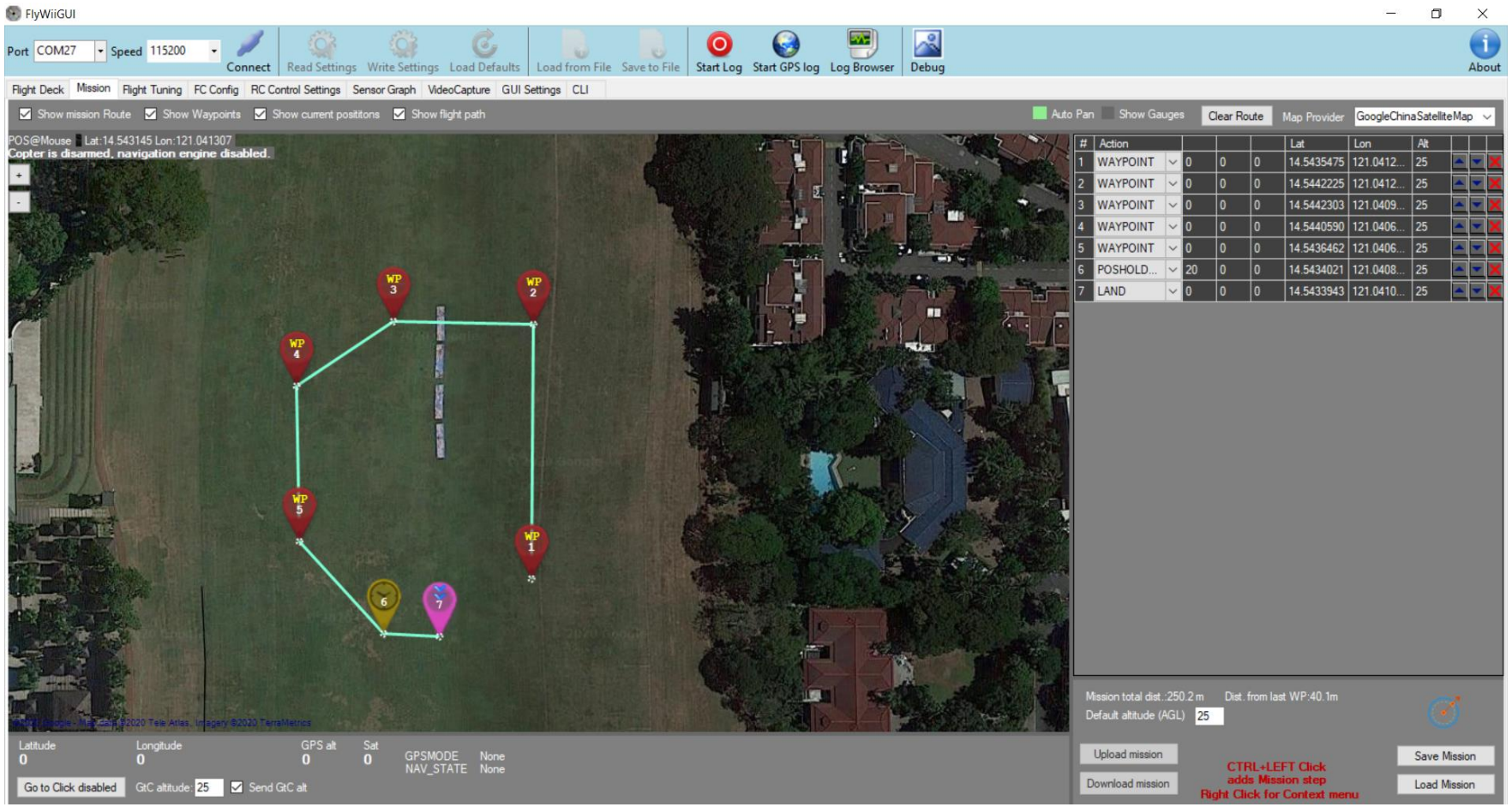
Waypoint – the Vehicle with travel between those points

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

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

Land – the Vehicle Stop when has reach this point (Must be place at the end of the mission)

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



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

Note: Only functional for Mega 2560 Boards with GPS

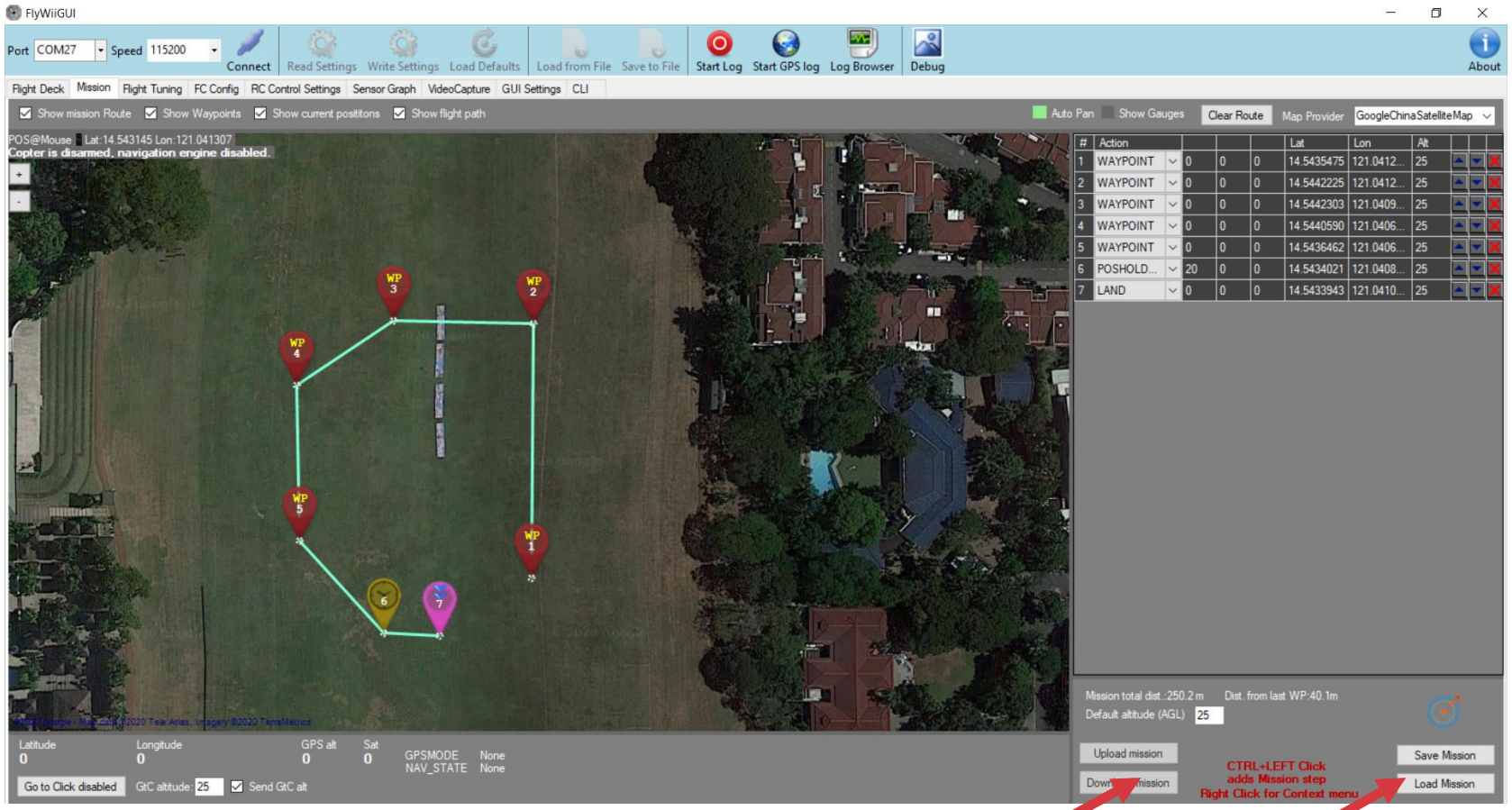
Waypoint – the Vehicle with travel between those points

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

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

Land – the Vehicle Stop when has reach this point (Must be place at the end of the mission)

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



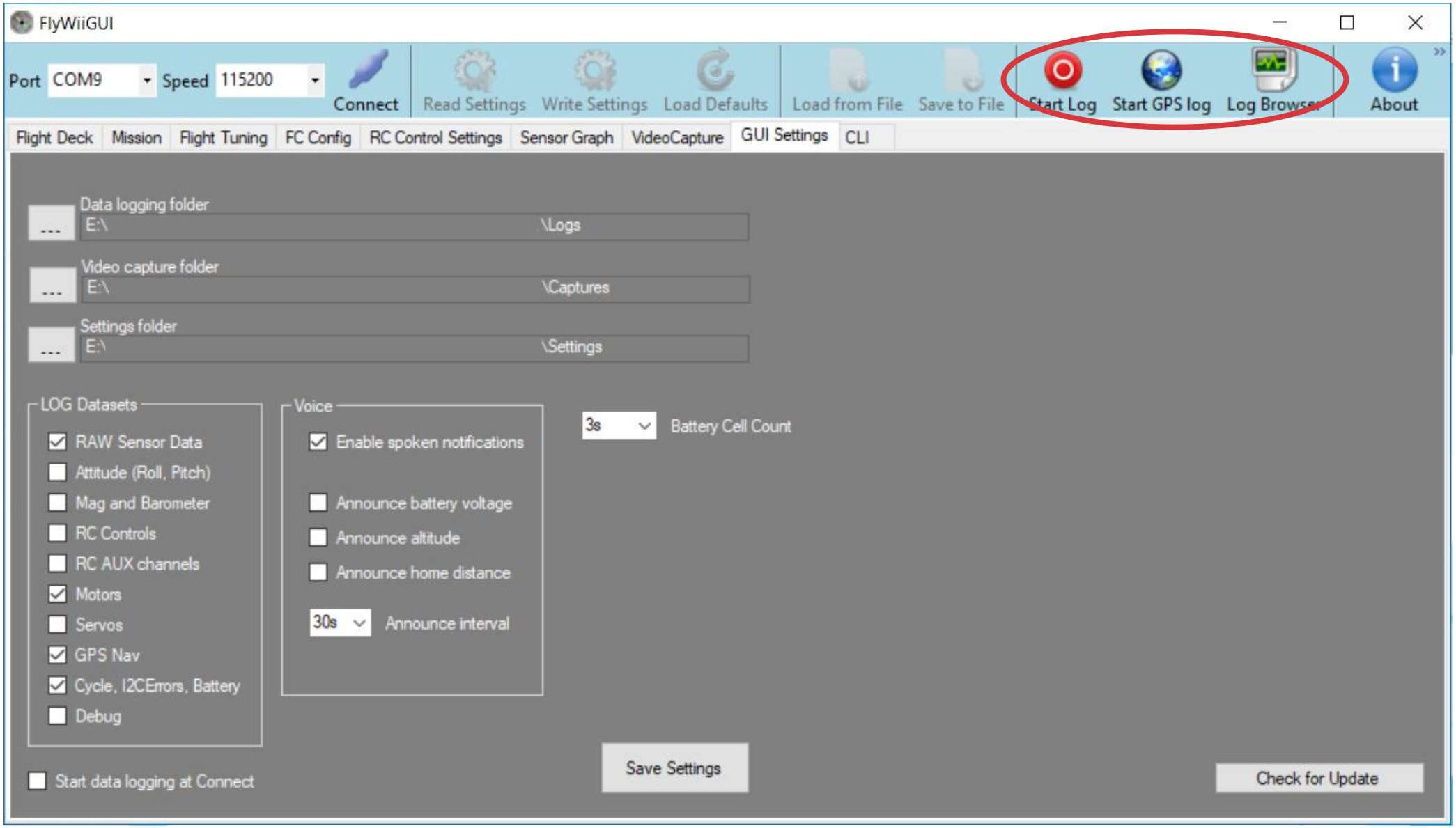
Mission upload to /download from Vehicle

Mission Save to /Open from File

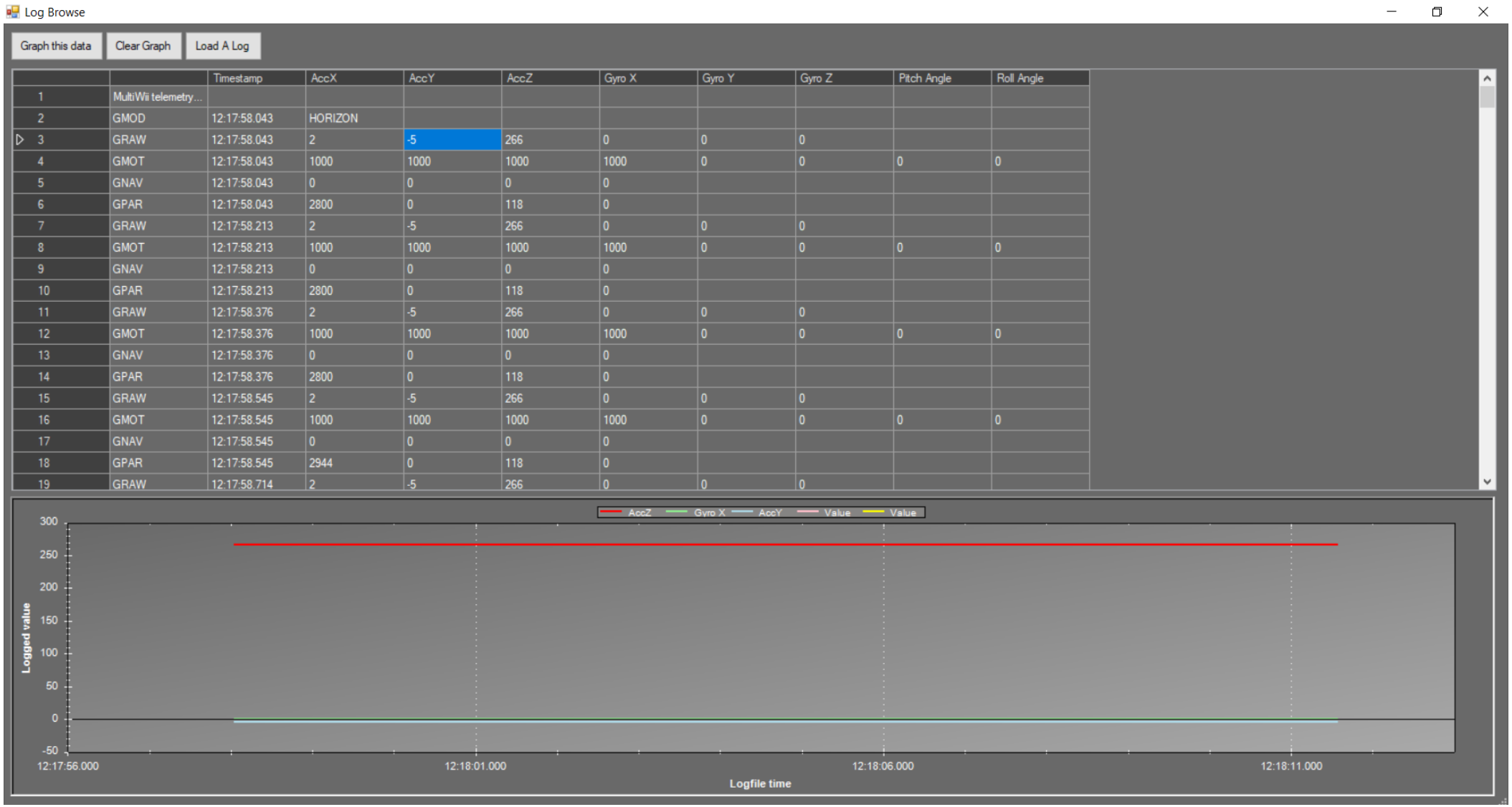
Graphs and Data Logging

Telemetry Log

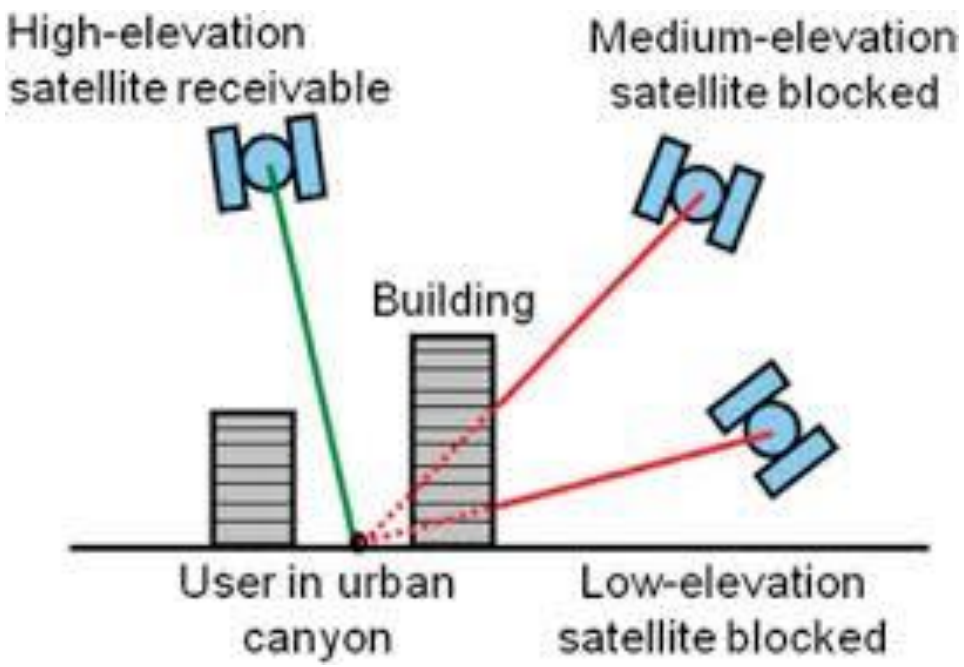
GUI Settings (where you save your PID ,Flight Logs and Video Logs)



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		



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

operating next to a building can distort satellite signal deteriorating accuracy

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

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)

You can Test with the Vehicle's wheels off the ground first

Pls calibrate ACC and Mag in the FlyWii GUI Dashboard

Ensure the compass is facing the correct orientation