

DROFLY PRO V3

User Guide

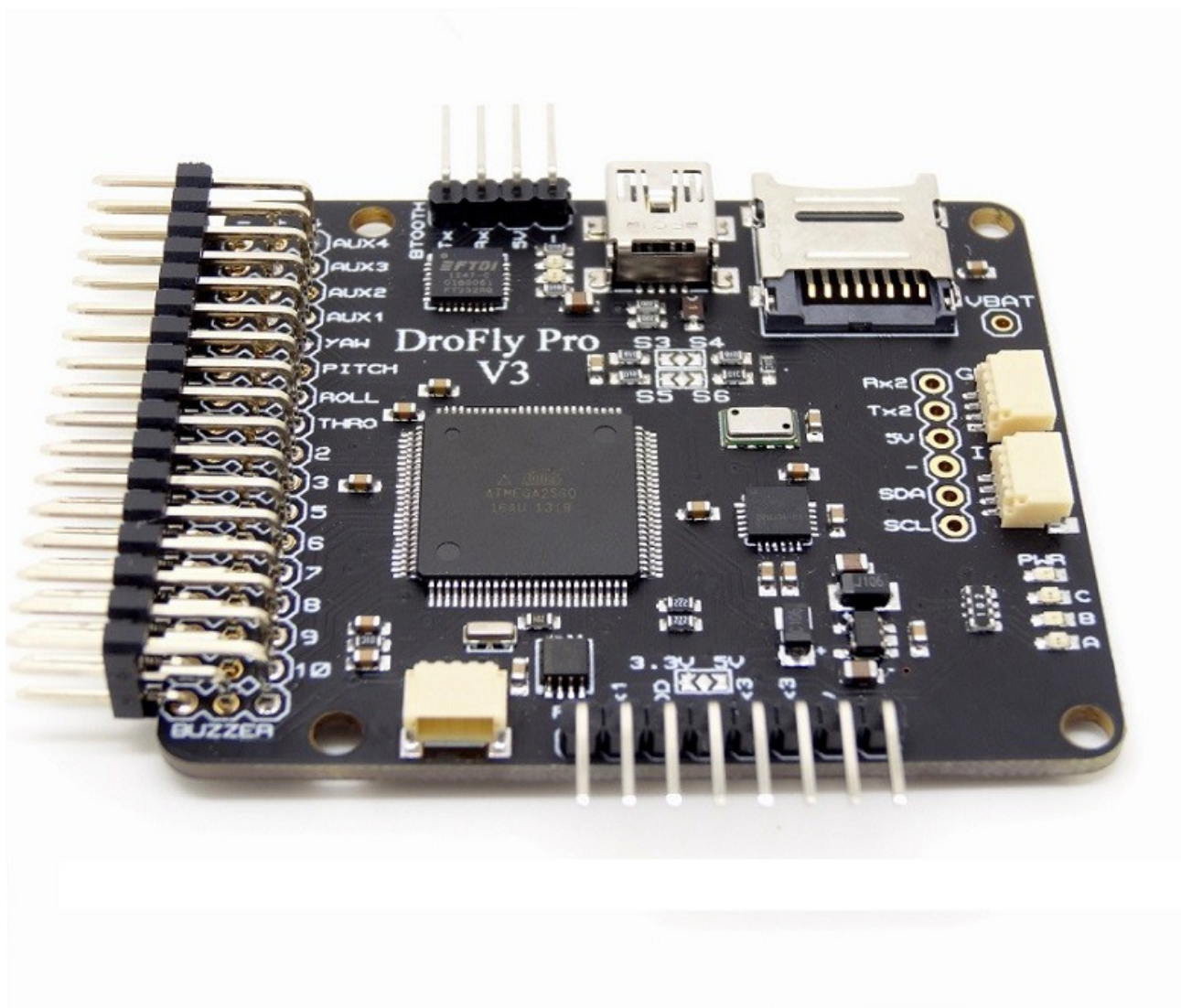


Table Of Contents

1/ INTRODUCTION

1.1/ Presentation

1.2/ Changes

2/ HARDWARE SETUP

2.1/ Setup on your frame

2.2/Connecting Flight Controller

a. To BEC

b. To Receiver

c. To Motors

2.3/ Enable LiPo

3/ SOFTWARE SETUP

3.1/ Getting started with Arduino Software

a. Download

b. Install

3.2/ Getting started with the Multiwii Code

a. Download

b. Install

3.3/ Multiwii code setup: config.h file

a. Number of motors

b. Flight Controller Selection

3.4/ Uploading Multiwii code into Flight Controller

3.5/ Using Multiwii GUI

3.6/ SD card

4/ ADDITIONAL MODULES

4.1/ GPS+HMC

4.2/ Bluetooth

INTRODUCTION

1.1/ Presentation

MultiWii is an open source program's original author is a French program Alex, after several years of joint efforts of many enthusiasts, and now this board has become increasingly mature, has become very popular abroad, cost-effective flight control one.

That multi-axis aircraft MWC fans can easily access the high cost of this product, through the MWC program to control the hardware platform, can achieve a variety of multi-spindle, high-quality flight results, flexibility can be adjusted into a very stable or very flexible to fit aerial, FPV or stunt flying and many different needs.

Compared to KK or ACM, MWC's more stable flight performance, flight skills required for novice lower debugging process more intuitive.

Dedicated MultiWii GUI program can easily observe the PID parameters, flight control signal input / output status, sensor / switch-state, as well as aircraft attitude and heading sensor signal state of each group, sufficient for the initial configuration of the aircraft easier.

MultiWii now supports multiple sensors and IMUs.

More information available here: <http://www.multiwii.com/>

Even if using MultiWii is simple, even for a beginner, it requires a few hardware and software initial tasks.

This user manual will guide you through the basic steps to get your model flying. This section will explain how to edit, setup and upload the MultiWii code into your flight controller.

1.2/ Changes

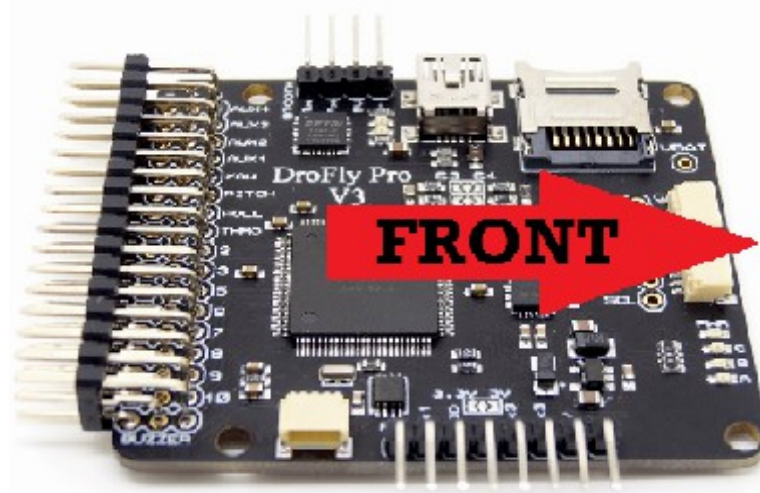
Here are the changes since DroFly Pro V2

DroFly Pro V2	DroFly Pro V3
HMC5883	SD port
Size : 50mm*50mm	Size : 58,5mm*50mm
	Battery level control with solder bridge
	Buzzer Output

HARDWARE

2.1/ Setup on your frame

First, you need to identify the front of your flight controller.
By default the front of the FC is like below:

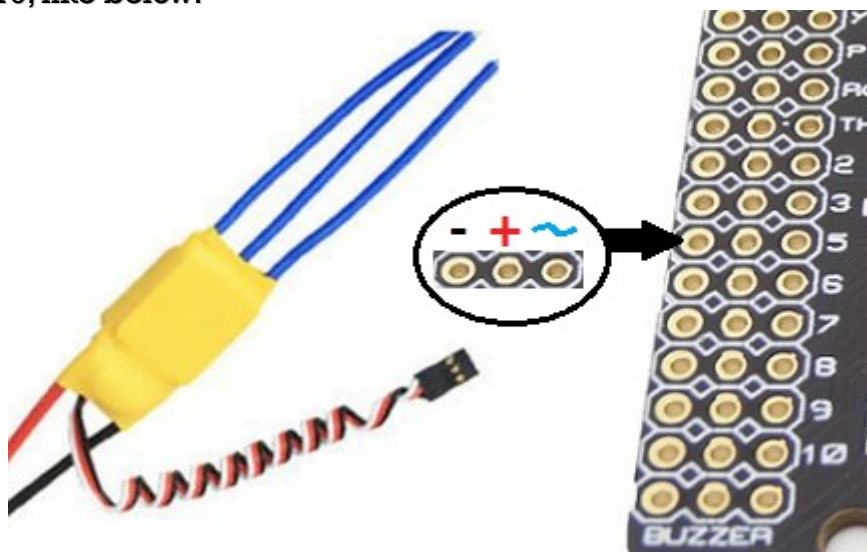


Anyway, you can easily identify flight controller orientation using GUI.

2.2/Connecting Flight Controller

a. To BEC

Flight controller must be powered by ESC BEC (5V) or by UBEC (5V) on connector n°2, 3, 5, 6, 7, 8, 9, 10, like below:



b. To Receiver

Here are 3 ways to connect your receiver to the flight controller depending of your hardware:

-Classic Receiver:

Each port of the receiver is connected to the flight controller. You can connect up to 8 channels.

-PPM Receiver:

You need to enable PPM receiver functionality in config.h file. Uncomment this lines:

```
#define SERIAL_SUM_PPM
#define PPM_ON_THROTTLE
```

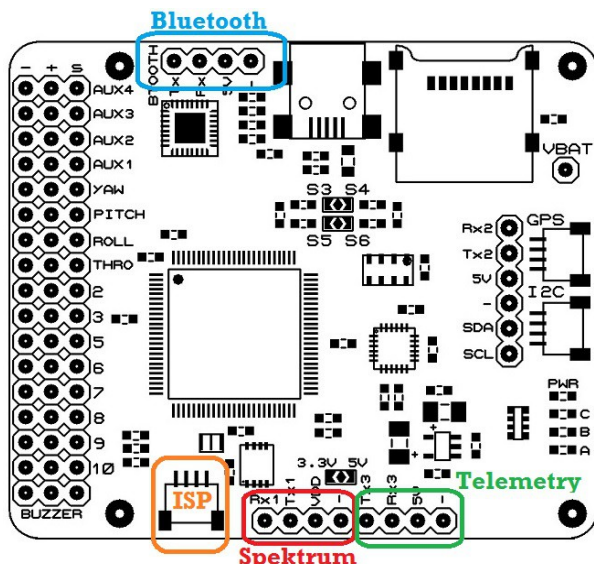
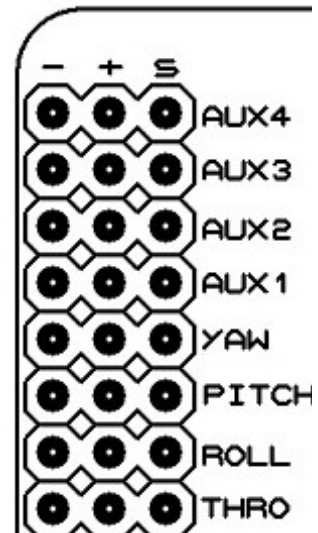
Your Throttle is now your signal.

-Spektrum Satellite:

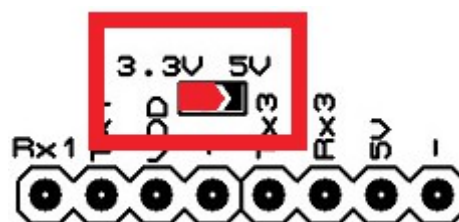
For a Spektrum Satellite, you must uncomment the line:

```
#define SERIAL_SUM_PPM [The configuration you have in fonction of your transmitter]
uncomment line #define SPEKTRUM 1024 if you're using DSM2,
or #define SPEKTRUM 2048 if you're using DSMX.
```

Then connect the Spektrum satellite to the dedicated port of the flight controller like below:



Spektrum satellite requires 3.3V on VCC. The solder bridge need to be activated as below:

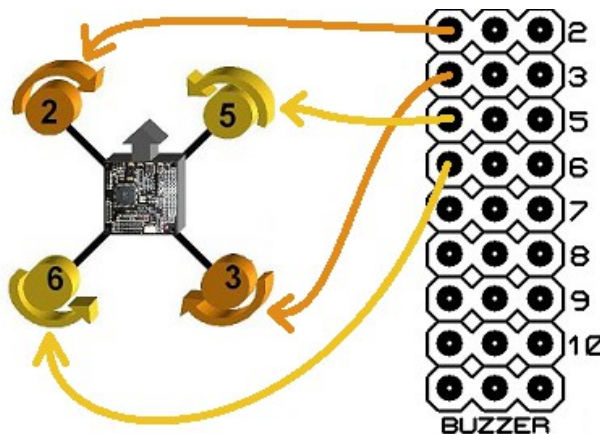


c. To ESC

MultiWii code can handle various types of flying models with different number of motors and servos.

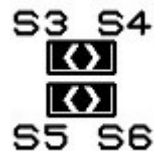
Please refer to section **3.3.b** to identify which type of model you're flying then connect ESCs accordingly to the flight controller.

DEFAULT CONFIGURATION

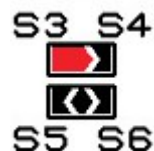


2.3/ Enable LiPo

You can select your own level battery on the board with these solder bridges. You will connect the center and the side you want. The objective is control the battery level and warn you when it is too low.



For select S3, solder as below :



You need to modify these lines, you can find them in *config.h*.

```
#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
```


SOFTWARE SETUP

ARDUINO



3.1/ Getting started with Arduino software

This section will explain how to setup Arduino software on your computer in order to edit and upload the Multiwii Code as you want.

a.Download

Latest Arduino software version is available here:

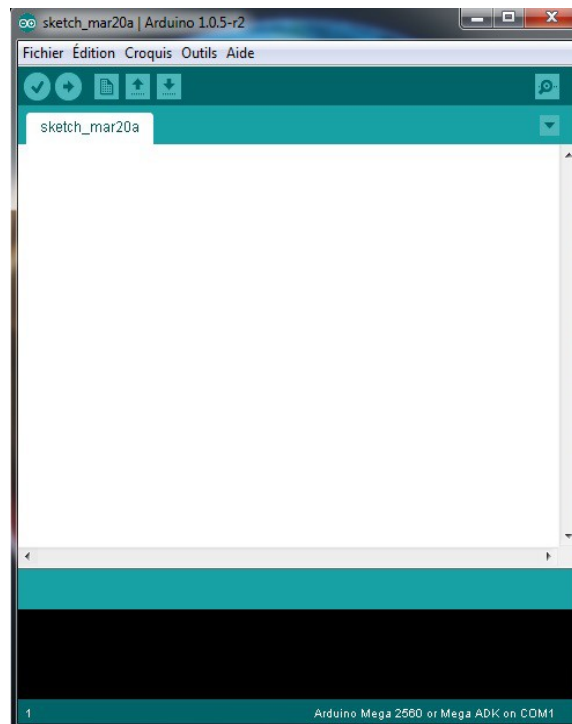
<http://arduino.cc/en/Main/Software>

Select and download the latest version available for your computer/operating system then extract the zipfile.

b.Install

Open the extracted folder and Open Arduino program (arduino.exe for Windows users).

This window will be displayed.



Arduino software is now installed and ready to use on your computer.

3.2/ Getting started with the Multiwii Code

This section will explain how to tune MultiWii code to fit your flying model.

a. Download

Please download the latest version of the MultiWii code available on this page:

<http://fernitronix.fr/component/attachments/download/24>

The downloaded file should be named as follow:

"MultiWii2.3_DroflyProV3.zip"

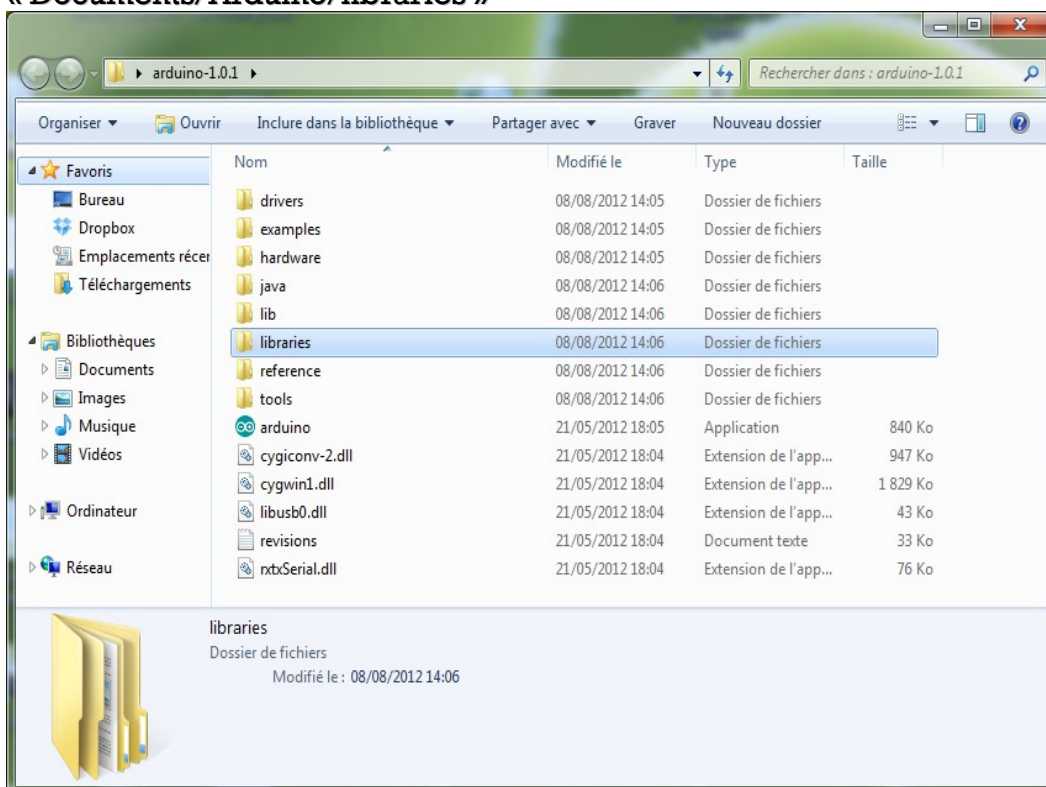
b. Install

Extract it in a dedicated folder; 2 subfolders will be created:

- MultiWii contains the code to be uploaded onto the flight controller
- MultiWiiConf contains the user interface to be used once MultiWii code uploaded onto the flight controller.

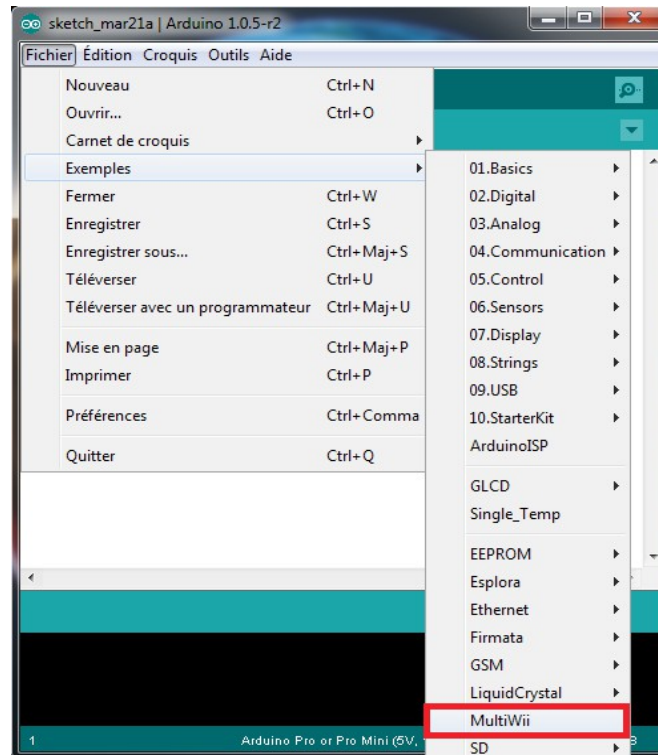
Then move « Multiwii » in :

« Documents/Arduino/libraries »



Open Arduino program (*arduino.exe* for Windows users).

Open the *File* menu, in the *Examples* section. You must see the *MultiWii* in the list.



Click on MultiWii it opens the full project code.

WARNING

BEFORE ANY HANDLING, MAKE SURE THE PROPELLERS ARE REMOVED.

3.3/ Multiwii code setup: config.h file

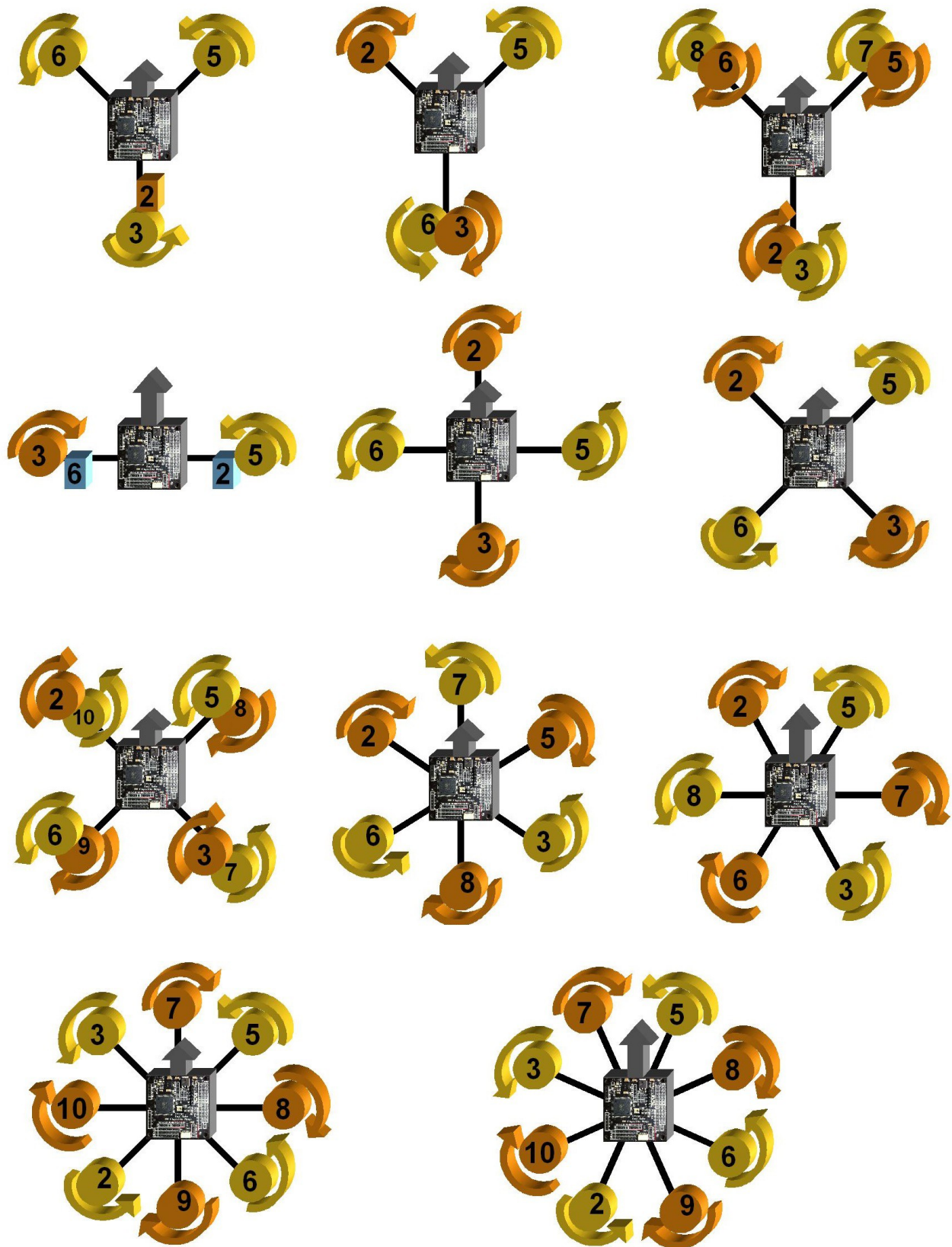
The MultiWii code is splitted in several files, displayed in different tabs in Arduino. Each one of them is dedicated to a specific functionality (sensors, GPS, LCD screen, etc...). *config.h* is the parameters file to be edited in order to fit your flying model.

WARNING: Do NOT modify other files unless you're an advanced user.

Click on the *config.h* tab to enter the code.

a. Number of motors

Multicopter models can take numerous different designs and number of motors. MultiWii code can handle all the listed types of models:



Identify the type of model you need then uncomment the relevant line in the code (just remove “//” at the beginning of the line).

Example, for a quadricopter X (2 front motors, 2 rear motors):

```

MultiWii_dev_20120606 | Arduino 1.0
File Edit Sketch Tools Help
MultiWii_dev_20120606 Buzzer EEPROM GPS IMU LCD LED Output R
/*****
/*****
/***** SECTION 1 - BASIC SETUP
/*****
/*****

/***** The type of multicopter *****/
// #define GIMBAL
// #define BI
// #define TRI
// #define QUADP
#define QUADX
// #define Y4
// #define Y6
// #define HEX6
// #define HEX6X
// #define OCTOX8
// #define OCTOFLATP
// #define OCTOFLATX
// #define FLYING_WING
// #define VTAIL4
// #define AIRPLANE // Howto setup =>>> http://fotoflygarn.blogspot.com/2012/04/multiwii-helicopter.html
/*****
// Heli is beta test .....!
// Howto setup =>>> http://fotoflygarn.blogspot.se/2012/04/multiwii-helicopter.html
// #define HELI_120_CCPM // PatrikE Experimental
// #define HELI_120_CCPM // PatrikE Experimental

```

b. Flight Controller Selection

Now, you need to select the type of flight controller you are using.

Uncomment the **#define DROTEK_DROFLY_V3** line.

```

// #define DROTEK_10DOF // Drotek 10DOF with ITG3200, BMA180, HMC5883, BMP085, w or w/o LLC
// #define DROTEK_10DOF_MS // Drotek 10DOF with ITG3200, BMA180, HMC5883, MS5611, LLC
// #define DROTEK_6DOFv2 // Drotek 6DOF v2
// #define DROTEK_6DOF_MPU // Drotek 6DOF with MPU6050
// #define DROTEK_10DOF_MPU//
// #define DROTEK_DROFLY_V2//
#define DROTEK_DROFLY_V3// Drotek DroFlyPro "V3" with MPU6050 and MS5611BA
// #define DROTEK_DROFLY_V3_GPS// Drotek DroFlyPro "V3" with MPU6050 and MS5611BBA + UBL0X GPS and HMC5883L integrated

```


The MultiWii code is now ready to be uploaded into your flight controller.

Please note this user manual describes only the basics of MutliWii code. Additional parameters (rx type, gimbal parameters, etc...) can be fine-tuned in the config.h file.

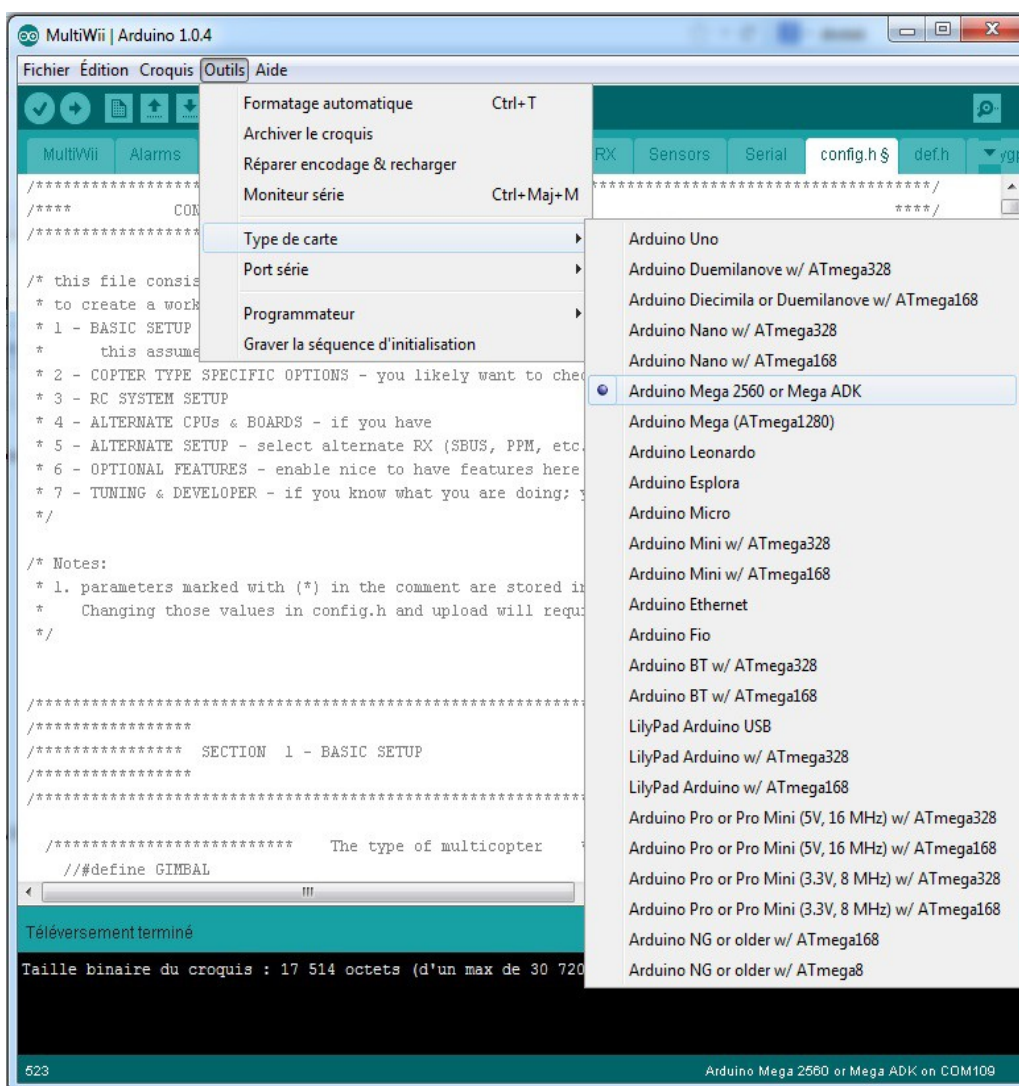
Please refer to the MultiWii FAQ for more details: <http://www.multiwii.com/faq>

3.4/ Uploading Multiwii code into Flight Controller

In order to compile and upload the code into the flight controller, you need to select :

-The type of board:

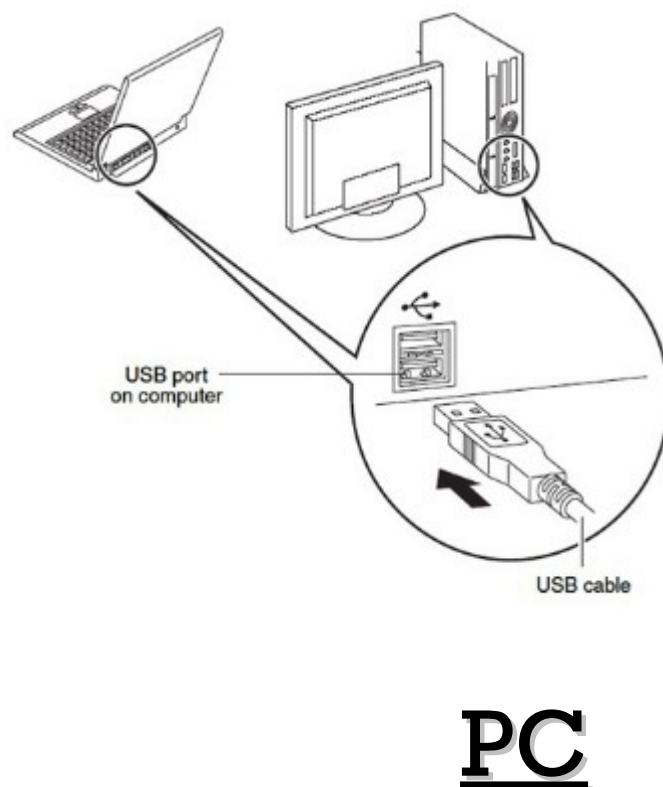
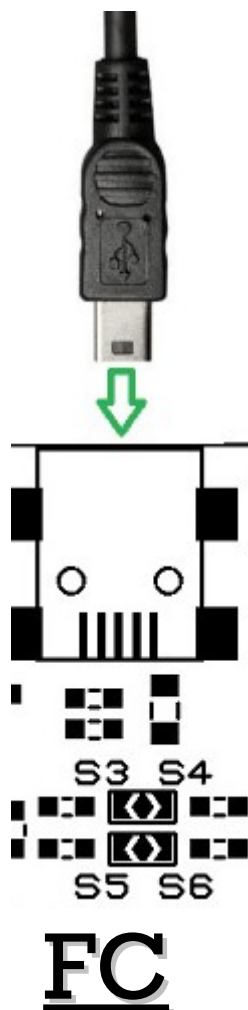
In Arduino program: Tools>>Board>>Arduino Mega 2560



-The serial port

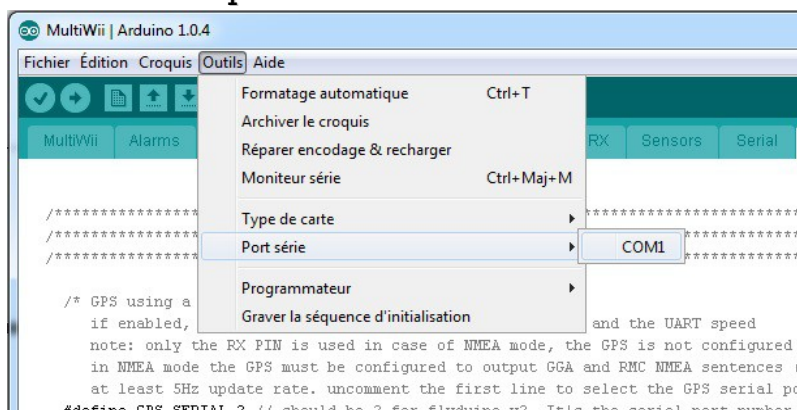
Now, connect the flight controller to your PC using a miniUSB-USB cable.

For Windows users, on first plug, a new device should be automatically detected and installed.



Once the message “*device successfully installed and ready to use*” is displayed, you can

select the relevant COM port in Arduino Tools menu:



Now click on the « upload » button:



This will compile and upload the code into the flight controller.

You can see compilation progress bar at the bottom-right of the Arduino window.



Once compilation complete, the code will be automatically uploaded into the flight controller.

The red and green LEDs on the flight controller will blink rapidly until “Done uploading” message appears at the bottom-left of the window. Your board is updated.

3.5/ Using Multiwii GUI

MultiWii code comes with a GUI (Graphical User Interface) which helps the user to setup:

-sensors recognition, accelerometer/compass calibration , PIDs, rx channels



Download and informations:

<https://code.google.com/p/multiwii/downloads/list>

3.6/ SD card

DroflyProV3 used a SD card as datalogger.

This is an option and you can fly without the card or without the fonction activated.

During the flight, the card memorize :

- Differents warnings (level battery, i2c, failsafe..)
- Flight data (time in flight, time of operating, number of starts...)
- GPS coordinates of the UAV at regular intervals.

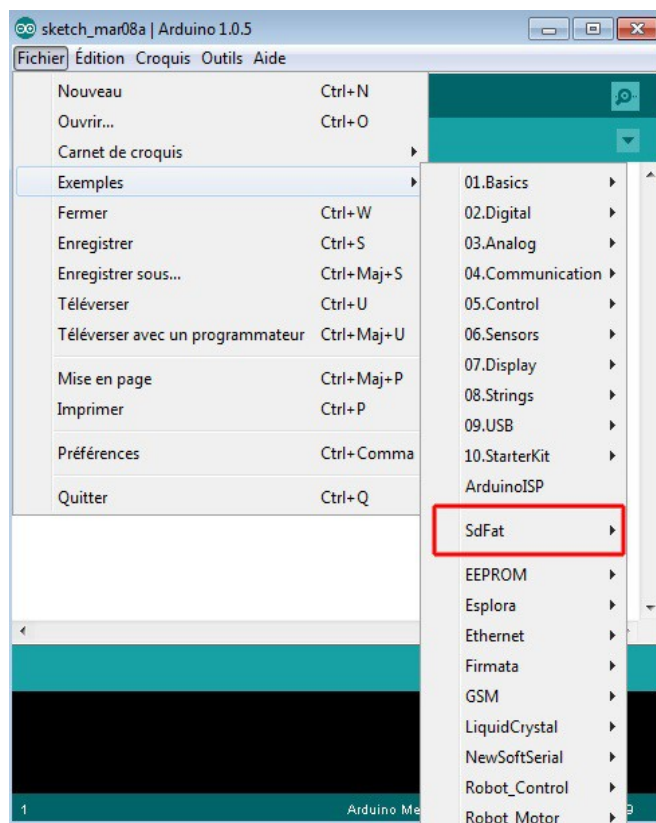
This allows after a flying session to analyze flight data and trace the path of such drone on Google Earth.

After all, you need to download the follow library:

<http://fernitronix.fr/documents/DroflyProv3/sdfatlib20131225.zip>

Move it in the Arduino library folder.

You should be have it now :



In *config.h*, uncomment:


```
#define LOG_PERMANENT 4096
#define MWI_SDCARD // activation of sdcard fonctionnality, needed for other defines underneath
#define SDCARD_LOGGER // Logging of permlog and gps if available
#define LOG_PERMANENT_SD_ONLY // Disable permanent logging on eeprom
#define LOG_GPS_POSITION 2 // Write GPS position to log. Parameter is the number of seconds between two logs
```

```
#define LOG_PERMANENT 4096
// #define LOG_PERMANENT_SHOW_AT_STARTUP // enable to display log at startup
// #define LOG_PERMANENT_SHOW_AT_L // enable to display log when receiving 'L'
// #define LOG_PERMANENT_SHOW_AFTER_CONFIG // enable to display log after exiting LCD config menu
// #define LOG_PERMANENT_SERVICE_LIFETIME 36000 // in seconds; service alert at startup after 10 hours of armed time

/* Logging to SDCARD module
*/
#define MWI_SDCARD // activation of sdcard fonctionnality, needed for other defines underneath
#define SDCARD_LOGGER // Logging of permlog and gps if available
#define LOG_PERMANENT_SD_ONLY // Disable permanent logging on eeprom
#define LOG_GPS_POSITION 2 // Write GPS position to log. Parameter is the number of seconds between two logs
// #define CSPIIN 45 // By default : 53 on mega boards, 10 on others. refer to your board specs
```

Upload the code.

Then you need a soft to process files between SD card and Google Earth.
The Soft is available here : <http://rcnet.com/manuels/RCNet/RCNetDT.V091.zip>

It should be named as “ KmlConverter” (Requires Java)

**The Datalogger is activated ONLY when the motors are armed.
SD card needs to be formatted in FAT before use.**

4/ ADDICTIONAL MODULES

4.1/ GPS+HMC

You can connect a serial GPS module to the serial port 2 of the Drofly Pro V3.
With our Ublox GPS Module with his HMC5883(magnetometer), you have to uncommment the lines below.

```
// #define DROTEK_10DOF // Drotek 10DOF with ITG3200, BMA180, HMC5883, BMP085, w or w/o LLC
// #define DROTEK_10DOF_MS // Drotek 10DOF with ITG3200, BMA180, HMC5883, MS5611, LLC
// #define DROTEK_6DOFv2 // Drotek 6DOF v2
// #define DROTEK_6DOF_MPU // Drotek 6DOF with MPU6050
// #define DROTEK_10DOF_MPU//
// #define DROTEK_DROFLY_V2//
// #define DROTEK_DROFLY_V3// Drotek DroFlyPro "V3" with MPU6050 and MS5611BA
#define DROTEK_DROFLY_V3_GPS// Drotek DroFlyPro "V3" with MPU6050 and MS5611BBA + UBL0X GPS and HMC5883L integrated
```

The code is simplified to allow to be understood by everyone.

Make sure you've uncommment these lines in *config.h*.


```
#define GPS_SERIAL 2          // should be 2 for flyduino v2. It's the serial port number on arduino MEGA
//#define GPS_PROMINI_SERIAL  // Will Autosense if GPS is connected when ardu boots.

// 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 57600

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

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

#define GPS_SERIAL 2

Defines the serial port number of the GPS.

#define GPS_BAUD 57600

Defines the speed of GPS.

#define UBLOX

Defines your GPS. (Ublox with the kit "Drofly Pro V3 + GPS")

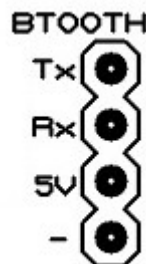
The GPS baud must be equal to the Drofly config. Otherwise it dosen't work.

4.2/ Bluetooth

Drofly Pro V3 can receive a Bluetooth module as previous version.



Bluetooth connection is used for get a feedback of data during the flight, mainly.
You can plug a Bluetooth module on this port:



You can easily config your PID with the Android application available here:

<https://play.google.com/store/apps/details?id=com.ezio.multiwii&hl=fr>