newkamikaze
June 27, 2016 at 5:38 pm

DIY FPV Racing Drone (Part 2) - Setup

9 min 33K

Multicopters, DIY or Do It Yourself

Tutorial

In the first part , I told you how to assemble a quadcopter for FPV flights. Now it's time to set it up. If interested, welcome to cat.

Let me make a reservation right away that I do not pretend to be an expert at all; this is only the third quadcopter I have assembled. In addition, setup is a very subjective thing. Nevertheless, I hope that the article will be useful to someone and help save time.
Let's move directly to the setup. First, simpler things, and then - His Majesty the flight controller.

Firmware and configuration of MinimOSD

The most popular (but not the only) firmware for MinimOSD is MWOSD . First you need to flash the

board for it, and then configure it. If you can set up the OSD via a PC (more about this in the section on setting up a PC), then for the firmware you need an FTDI adapter or an Arduino. How to do this through an FTDI programmer, and most importantly where to get old drivers for it, is shown in this video . The main thing is not to forget to uncomment the following lines in Config.h before flashing the firmware: Of the parameters, I began to display only battery voltage, flight time and the selected flight mode.

```
 #define MINIMOSD
#define CLEANFLIGHT
```

Firmware and configuration of regulators

I also have a special USB adapter for flashing the regulators, but you can do without it by connecting via a PC. I thought for a long time whether to install Multishot or the already proven Oneshot125? On the one hand, on the outdated F330 chip there is not much difference in speed; on the other hand, there is a melody from “Star Wars” when turned on and, as they write on the forums, “a cleaner signal.” We have resolved all the occasional complaints that the motors spontaneously begin to rotate at maximum speed when connected to the CLI. As a result, I installed the latest version of

BLHeli (14.5 at the time of assembly), turned on Damped Light and set Motor Timing to “Medium”. Later I calibrated the regulators according to these instructions .

Flight controller firmware

At this stage I was stuck the longest, as there were problems with the firmware. It turned out that the first time you sew, you must close the boot contacts (as in this video). By the way, sometimes it happens that the PC is write-protected and it is impossible to flash new firmware. Here are instructions on how to fix it.

Perhaps the most popular firmware today, and deservedly so, is Cleanflight . After installing it, you just need to configure the receiver protocol and the quadcopter can fly quite well. For configuration, a convenient graphical shell is used Cleanflight Configurator .
Thanks to the open source firmware, it has several branches (forks). The most interesting of them is Betaflight from a person under the nickname Boris B. The firmware is developing very dynamically and some of its “features” are then transferred to the “parent” Cleanflight (for example, the Airmode flight mode). The downside of Betaflight is that releases come out quite often, and their stability is not always high. By the way, this reason delayed the writing of this article for several weeks. At the time the quadcopter assembly was completed, version 2.8.0 had just been released, which had a couple of errors and was distinguished by unfriendly default settings. The corrected version 2.8.1 RC1 appeared very quickly, but my experience as a programmer told me that it was better to wait for the release. I was right, because simultaneously with the release of version 2.8.1, Betaflight Configurator also appeared . We can say that this is a new stage in the history of this firmware. The fact is that as Betaflight developed, it moved more and more away from Cleanflight and the latter’s configurator became more and more useless, since the bulk of the settings were still done through the CLI console. By the way, at the time of writing this article, Betaflight version number 2.9.0 had already been released, but due to negative reviews I did not update to it.
Below I will describe in detail how I configured my quadcopter using Betaflight Configurator.

PC setup via Betaflight Configurator

Setup tab

I calibrated the accelerometer.

Ports Tab

- In order for OSD to work, I enabled MSP for the UART2 port.
- Do not under any circumstances disable MSP for UART1 port. It is paralleled with USB and by disabling data transfer, you no longer connect to the PC via the USB connector.

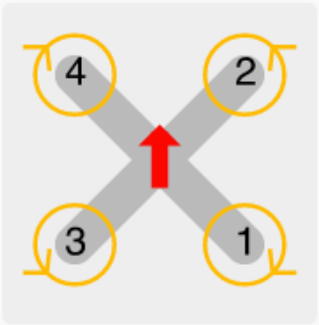
Identifier	Data	Logging	Telemetry	RX	GPS
UART1	MSP 115200	Blackbox 115200	Disabled AUTO	Serial RX	57600
UART2	MSP 115200	Blackbox 115200	Disabled AUTO	Serial RX	57600
UART3	MSP 115200	Blackbox 115200	Disabled AUTO	Serial RX	57600

Configuration Tab

Actually, almost all the settings are made here.

- In the “Board and Sensor Alignment” section I indicated that my PC is rotated 90 degrees along the yaw axis. The correctness of this parameter can then be checked in the Setup tab.
- In “Receiver Mode” I chose RX_PPM.
- I chose ONESHOT125 as the regulator protocol (why not MULTISHOT, I wrote above).
- I disabled “Unsynced PWM output”, again because I don’t use MULTISHOT.
- To ensure that the motors always rotate in a reinforced state, I disabled MOTOR_STOP.
- I enabled the “Disarm motors regardless of throttle value” option, since I will be doing armature motors on a separate channel.
- I reduced the “Minimum Throttle” value to 1030. This is the idle speed, I selected it subjectively.
- Enabled the “VBAT” option to activate the battery charge indicator. I did not change the remaining parameters in the “Battery Voltage” section, since the charge readings correspond to reality. If there is an error in them, it can be removed by setting the “Voltage Scale”.
- In “Other Features” I activated BLACKBOX, SUPEREXPO_RATES, as well as LED_STRIP and turned off AIRMODE. I'll tell you about it a little lower.

Mixer



Quad X

ESC/Motor Features

ONESHOT125 — ESC/Motor protocol

- Unsynced PWM output
- MOTOR_STOP — Don't spin the motors when armed
- Disarm motors regardless of throttle value (When arming via AUX channel)

1030	Minimum Throttle
1500	Middle Throttle [RC inputs center value]
1850	Maximum Throttle
1000	Minimum Command

Board and Sensor Alignment

0	Roll Degrees	GYRO Alignment	Default
0	Pitch Degrees	ACCEL Alignment	Default
90	Yaw Degrees	MAG Alignment	Default

Accelerometer Trim

0	Accelerometer Roll Trim
0	Accelerometer Pitch Trim

Receiver Mode

RX_PPM	PPM RX input
RX_SERIAL	Serial-based receiver (SPEKSAT, SBUS, SUMD)
RX_PARALLEL_PWM	PWM RX input (one wire per channel)
RX_MSP	MSP RX input (control via MSP port)

Battery Voltage

- VBAT — Battery voltage monitoring

3,3	Minimum Cell Voltage
4,3	Maximum Cell Voltage
3,5	Warning Cell Voltage
110	Voltage Scale
1.4	Battery Voltage

Serial Receiver Provider

Note: Remember to configure a Serial Port (via Ports tab) and choose a Serial Receiver Provider when using RX_SERIAL feature.

SPEKTRUM1024
SPEKTRUM2048
SBUS
SUMD
SUMH
XBUS_MODE_B
XBUS_MODE_B_RJ01
IBUS

Current Sensor

- CURRENT_METER — Battery current monitoring

400	Scale the output voltage to milliamps [1/10th mV/A]
0	Offset in millivolt steps
0.00	Battery Current

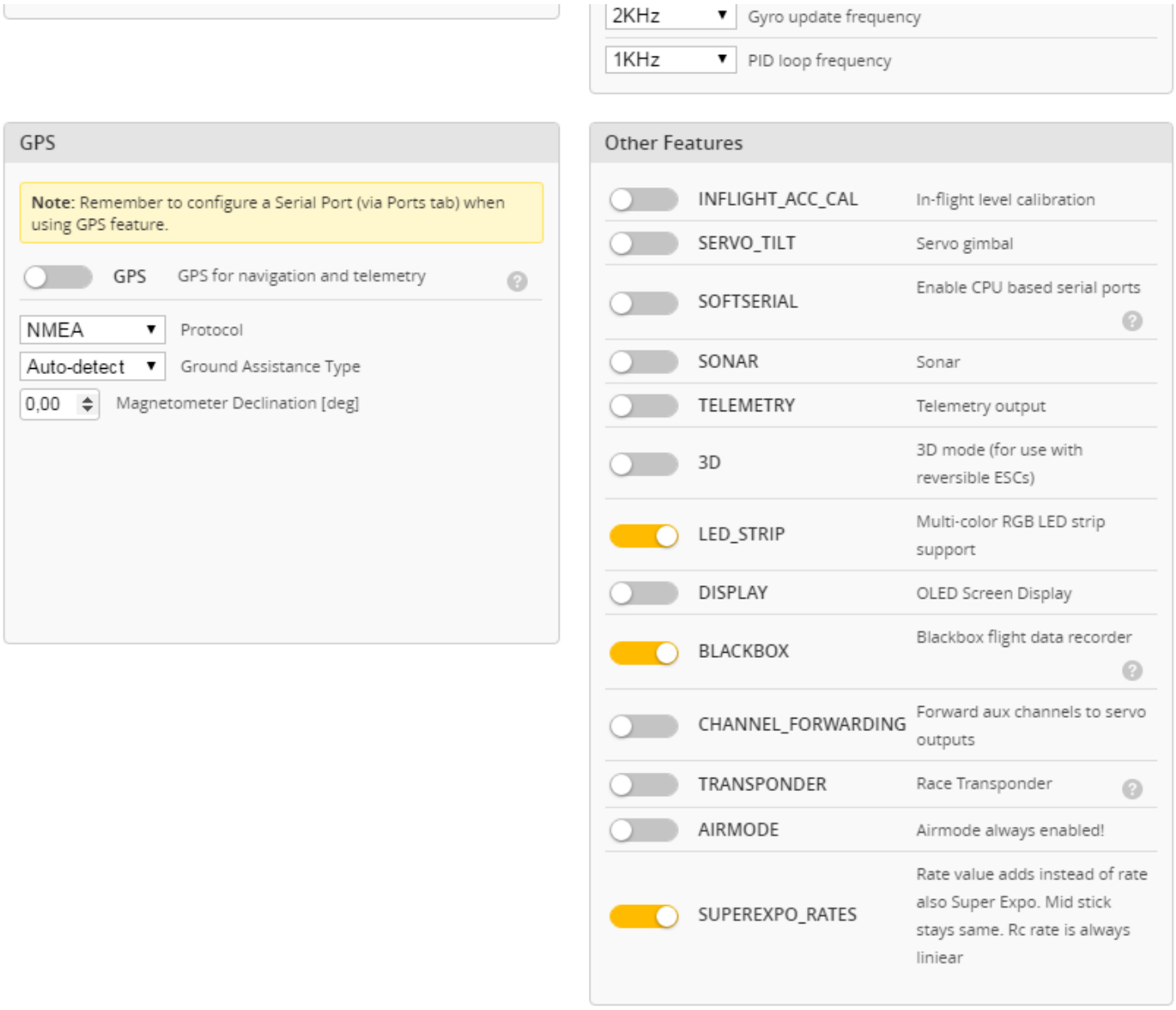
- Enable support for legacy Multiwii MSP current output

RSSI (Signal Strength)

- RSSI_ADC — Analog RSSI input

System configuration

Note: Make sure your FC is capable to operate on these speeds! Check CPU and cycletime stability. Changing this may require PID re-tuning. TIP: Disable Accelerometer and other sensors to gain more performance. (CLI commands: set acc_hardware = NONE, set baro_hardware = NONE, set mag_hardware = NONE)



Failsafe tab

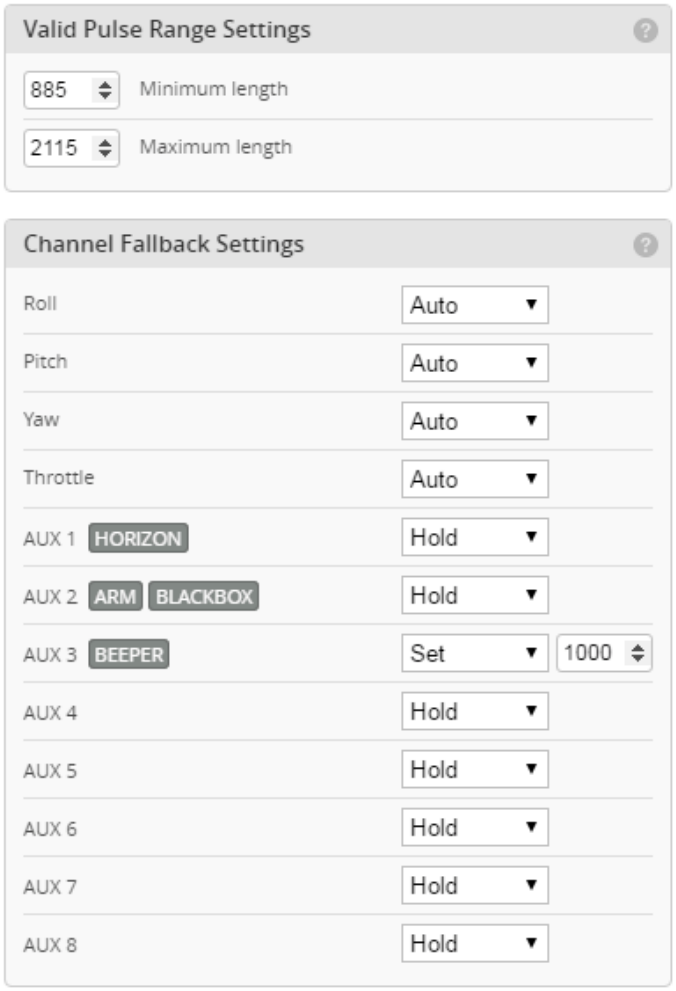
With failsafe, everything turned out to be somewhat more complicated than I had previously imagined. On specialized forums, sometimes there are holivars on the topic “where is it better to configure failsafe: on the receiver or on the PC?” In fact, the correct question is: “Where is it better to configure failsafe: only on the receiver or on the receiver and on the PC?”
It is necessary to configure failsafe on the receiver in any case. Here it is necessary to make an

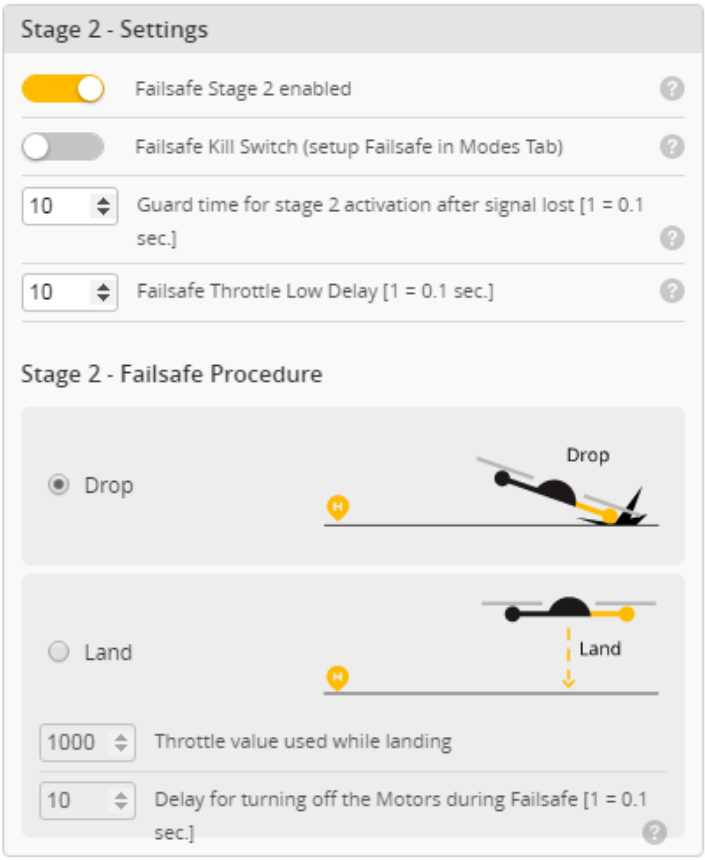
important clarification that we are talking about the Frsky D4R-II receiver, working using the PPM protocol. Receivers with S.Bus failsafe are configured differently.
The Frsky D4R-II has three options for behavior when the signal from the transmitter is lost:

- transmit a signal to the PC that simulates preset positions of sticks and switches (Pre-set Positions mode, this is what is described in the manual)
- continue to transmit the latest data received from the transmitter to the PC (Hold Last Position mode)
- stop transmitting the signal to the PC (No Pulse mode)

By default, the Frsky D4R-II is set to the Hold Last Position mode, which helps the device fly far away. So if you use failsafe only on the receiver, you need to configure the Pre-set Positions mode. Another thing is that failsafe is activated even with a short-term loss of signal. It will be very unpleasant if the signal is restored in a split second, but the quadcopter has already disarmed the motors and is falling down. The situation can be improved by setting failsafe on a PC, since there this mode has a response delay, which serves as a filter against short-term signal losses. In addition, there is a customizable scenario according to which the quadcopter will behave if failsafe is activated. For example, you can turn on the stabilization mode and try to sit down more or less softly or even activate RTH, if there is one. But in practice, such things are in demand on large devices with GPS, and on small and fast ones, experienced pilots advise not to split hairs and turn off the engines after losing the signal. Due to the high flight speed, it will be safer for others.
The nuance is that if the Pre-set Positions or Hold Last Position mode is set on the receiver, then the PC will not even know that a signal loss has occurred. Thus, failsafe on a PC can only be used if the receiver is set to No Pulse mode. On the Frsky D4R-II it is installed by briefly (less than 1 second) pressing the failsafe button **with the transmitter turned off** (video instruction).
On the Failsafe tab of Betaflight Configurator I did the following:

- In the “Channel Fallback Settings” section, I set the values of the transmitter switches that will need to be simulated. In my case, I just turned on the beeper on AUX3.
- Failsafe Stage 2 activated.
- Set the failsafe mode activation delay to 1 second (value 10 in the “Guard time for stage 2 activation...” item)
- I set the operating time of the motors after activation of failsafe to 1 second (value 10 in the Failsafe Throttle Low delay item).
- In Failsafe Procedure, I chose the scenario with a fall (Drop), rather than with an attempt to land smoothly (Land).





PID Tuning Tab

Setting up PID is a process that you absolutely shouldn't get into without knowing the ford. First of all, a theoretical understanding of these three components is necessary. Here are some articles that can help with this:

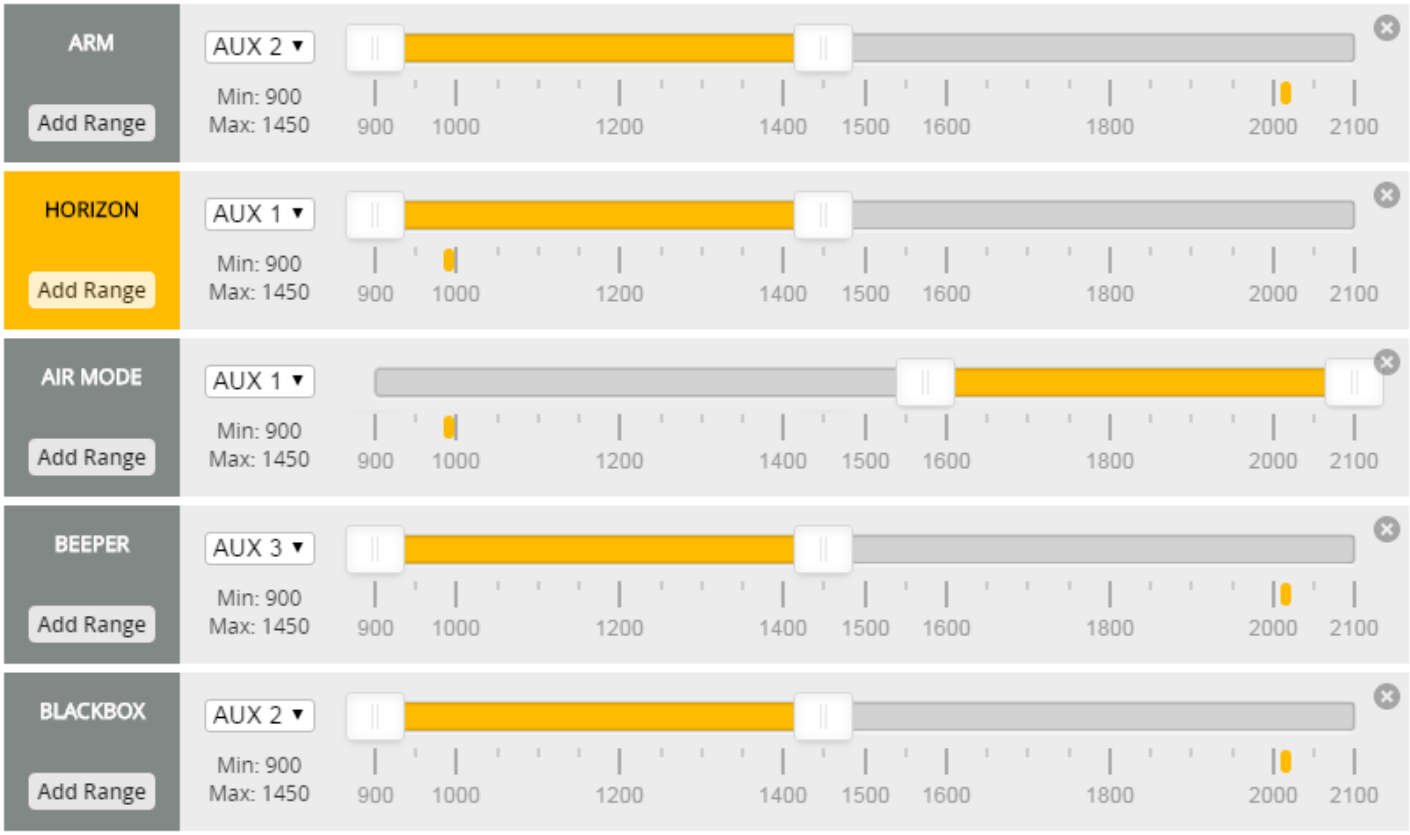
- PID for quadcopters (translation)
- Say a word about PID controllers
- And more about PID controllers
- Basics of setting up a PID controller using Blackbox logs (translation)
- Video Just about setting up PIDs

I don't feel ready for such a setting, so I left the default values, since Betaflight allows you to fly well with them. But I still made two changes:

- Selected Float in the “PID Math” section. This is a receiver of the LuxFloat mode from Cleanflight and is recommended to be installed only on PCs with F3 or F4 processors.
- I chose MEASUREMENT in the “Derivative method” section. In short, MEASUREMENT is preferable for freestyle, and ERROR is preferable for racing.

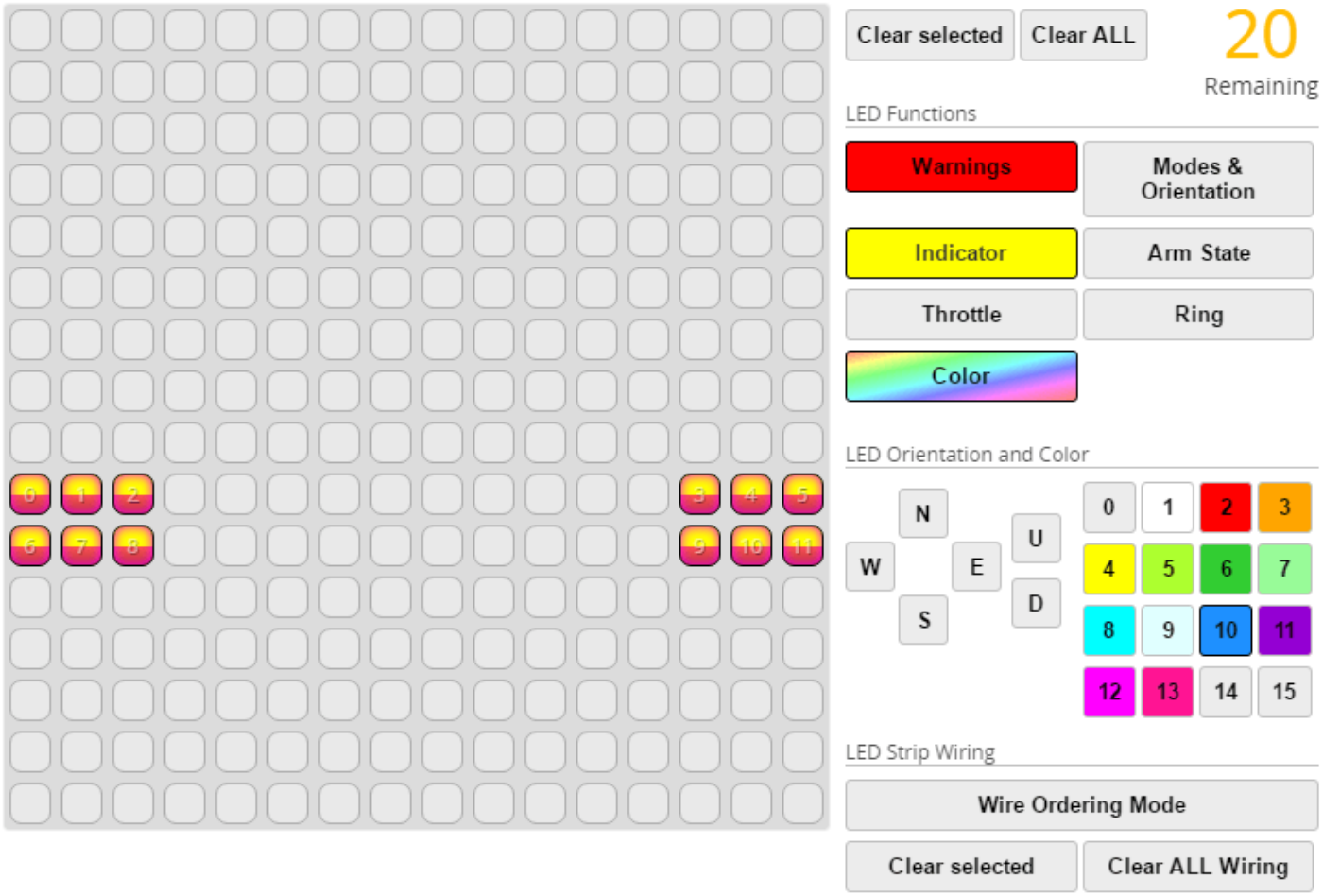
Modes Tab

I put arming of the motors and Blackbox (AUX1), activation of the Horizon flight mode (AUX2) and activation of the beeper (AUX3) on the switches.
I would also like to say something about AIRMODE. Initially, it was a unique “feature” of Betaflight, but at some point it became so popular that Boris B shared it with the authors of Cleanflight and now this mode is available there too.
Despite the fact that AIRMODE is displayed as a separate flight mode, it is more of an additional option rather than a full-fledged mode. It allows the quadcopter to maintain a given angle even with minimal throttle. This is why it is not recommended to use AIRMODE together with modes with stabilization. In addition, landing with AIRMODE is also not an easy process: the quadcopter starts jumping like a frog. Experienced pilots prefer to simply “drop” the quadcopter, turning off the motors a couple of tens of centimeters above the ground. By the way, if you have enabled stopping the motors at zero throttle (MOTOR_STOP option in the Configuration tab) and at the same time AIRMODE is running, then the motors will not stop, since AIRMODE has a higher priority.
In Betaflight version 2.8.1, a new feature has appeared: you can enable AIRMODE in the background (something like a passive perk in games) and then it is always active and will not be displayed in the Modes tab, or, as before, set its activation to some -or channel. This is done in the “Other Features” tab of the Configuration tab.
At my place, I did not turn on AIRMODE in the background, since I also use the mode with HORIZON stabilization. Thus, I have two flight modes on AUX1: HORIZON (for stabilized flights and landing) and ACRO + AIRMODE.



LED Strip Tab

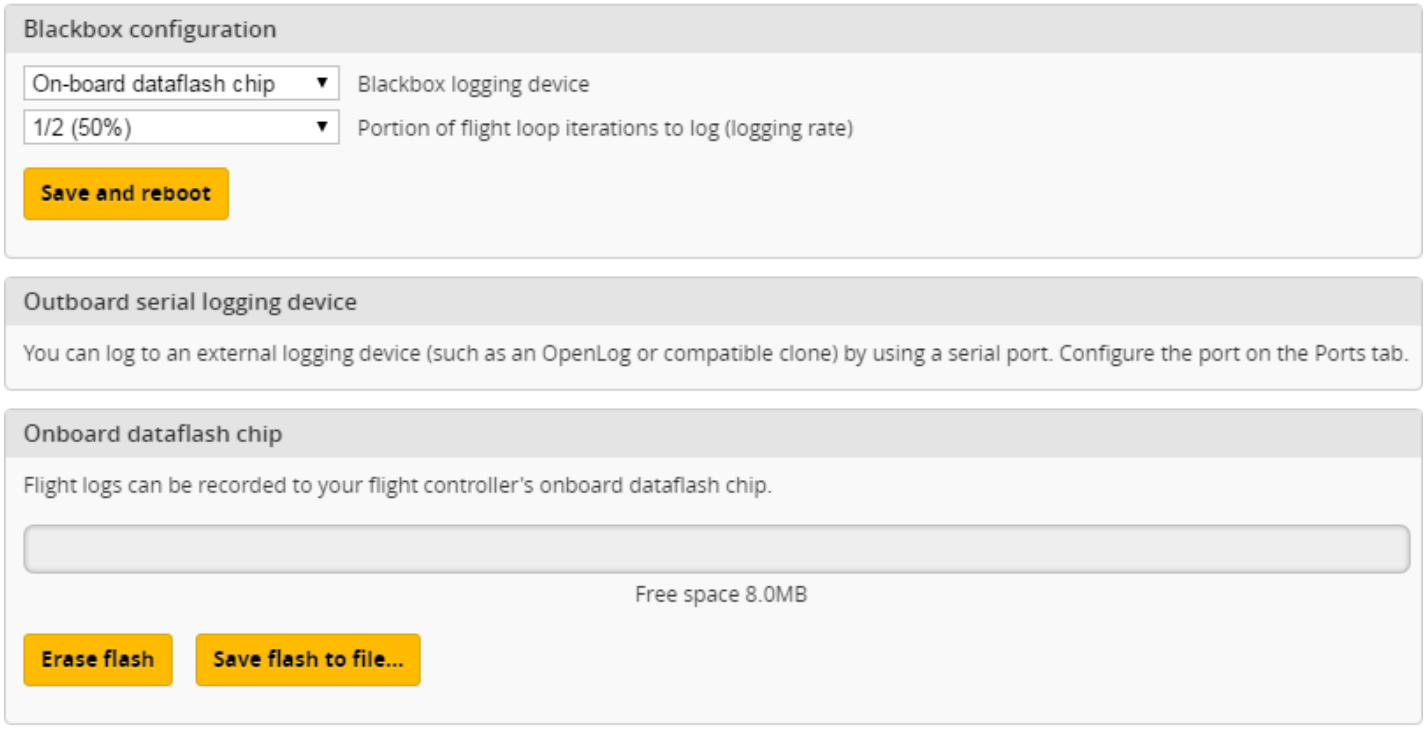
I configured my backlight to show warnings, turn/brake indicators, and glow blue when none of these are present.



Blackbox tab

Blackbox is the “black box” of the quadcopter. Needed for diagnostics, more precise PID tuning, and also to be able to overlay infographics on a flight video (like here). Data is written to an external logger or, if memory capacity allows, to the internal memory of the PC. For example, Naze32 Acro does not have enough of it, while Naze32 Deluxe and SPRacingF3 Acro have enough, although not by much. With default settings, 2MB of memory can record flight data lasting 3-4 minutes, which is very little. The situation is aggravated by the fact that it is impossible to implement “round-trip” recording, as in car recorders, due to the low memory read/write speed. The only option is to reduce the recording speed several times. For diagnostics such data will be of little use, but for video it will be just fine. In this tab I did the following:

- Selected recording to internal memory (option “On-board dataflash chip”) in the “Blackbox logging device” section.
- In the “Portion of flight loop iterations to log” section, I chose 50%, which essentially reduces the recording speed by half.



As I wrote above, logging is started by the same toggle switch on the equipment as arming, so that recording begins when the motors start. You can read more about the operation and settings of Blackbox in the manual . Also some useful links on the topic:

- Article Setup Blackbox in Cleanflight
- Translation of the article “Overlaying Blackbox data on video”
- Translation of the article “Basics of setting up a PID controller using Blackbox logs”
- Video on setting up and using Blackbox
- Blackbox Explorer application for viewing logs
- Blackbox tools application for working with data. For example, you can render in png

CLI tab

The CLI is a console through which you can change more advanced settings, as well as make a backup copy of all settings. I did the following: Betaflight also has a very useful option to connect to the OSD via PC. Now you can tuck this scarf away without worrying that you might need to connect to it. To connect to the OSD board via a PC, you need to connect the battery to the quadcopter, then enter the *serialpassthrough* command in the CLI with the necessary parameters, then disconnect (Disconnect) Betaflight from the PC and run MWOSD. I got it right the first time. Regarding the

parameters, for Micro MinimOSD connected to UART2 (my case) they are as follows: This completes the configuration of the flight controller.

```
 set small_angle = 180 # Включить возможность армить моторы даже в перевёрнутом состоянии
set vbat_pid_compensation = ON # Включить компенсацию PID при разрядке батареи
save # Сохранить настройки
```

```
 serialpassthrough 1 115200
```

Transmitter setup

Each pilot configures the transmitter (aka “remote control”) individually: timers, mixes, voice notifications, etc. The only thing that is highly advisable to do is check the minimum, average and maximum values of the control sticks in the configurator. This is done in the Reciever tab. Ideal values are 1000 - 1500 - 2000. In my case they were 996 - 1508 - 2020, which is not very good. Firstly, “falling out” outside the range (values less than 1000 and more than 2000) are bad in themselves. Secondly, a central position other than 1500 will be perceived by the PC as a slight steering, which it will work out and the quadcopter will constantly drift in one direction. In a word, it makes sense to tinker and configure it “as it should.”
How to set these values on the Taranis transmitter is shown here . I have a Turnigy 9XR PRO, where this is done in the Limits section. You can also configure it through the eePskye program (Limits tab), but this is inconvenient, since the result is not immediately visible in Betaflight Configurator. This must be done for each of the four control channels.
After adjustment, the central values were as close as possible to 1500, but for me they began to “jump” by about 5 units in one direction or the other. I don’t know what this is connected with, probably the remote control values are borderline for a PC and after rounding them this effect is obtained. To solve this problem, Betaflight (and CleanFlight too) has a special command that sets up filtering for such things. The value can be from 0 to 32 and as it increases, the control sensitivity decreases. Control becomes softer. You must understand that after a certain threshold, softness will turn into softness, so for myself I chose the minimum value at which the twitching disappeared.

That's all, have a good flight!

```
set deadband = 6
```

Tags: quadcopter

Hubs: Multicopters, DIY or Do It Yourself

14 Karma 0 Rating

@newkamikaze

User

Comments

There are no comments here yet, you can be the first!

Only full-fledged users can post comments. Come in , please.

Publications

BEST OF THE DAY SIMILAR

Zangasta
20 hours ago

Ode to the Futility of Argument

Simple 6 min 8.7K

Opinion

+111 77 162

Mike-M
21 hours ago

M.Video: We don’t care

Simple 34 min 24K

Case

+51 40 104

Quled
19 hours ago

AI porn photo generators: ethics, trends and legislation

Simple 6 min 16K

Analytics

+35 52 119

DmitryZlobec
18 hours ago

In RISC-V Rust

Average 16 min 4.2K

Case

+30 29 3

Bright_Translate
yesterday at 05:00

The unexpected interaction of branch prediction and memory subsystems

10 min 3K

Translation

+30 thirty 1

roman_deev
22 hours ago

News from the world of OpenStreetMap for December

9 min 2.3K

Digest

+20 12 0

IgorSL
yesterday at 05:23

How I tried to write a flight simulator

Average 9 min 7.6K

Retrospective

+19 37 9

alizar
1 hour ago

When old software is better than new

Simple 7 min 3.5K

Opinion

+16 15 18

dvgureev
16 hours ago

Large test of GPT4, GPT3.5, YandexGPT, GigaChat, Saiga in a RAG task. Part 2/2

Average 5 minutes 10K

Review

+13 43 eleven

Machinelearning_Ai
5 hours ago

NumPy: Practice Your Data Science Skills

Average 21 min 1.7K

Translation

+12 32 6

We are developing a marketplace for supply: the experience of EVRAZ

Turbo

show more

COURSES

Middle Python developer
January 11, 2024 · 110,000 ₽ · Yandex Workshop

Middle frontend developer
January 11, 2024 · 89,000 ₽ · Yandex Workshop

Profession: Fullstack developer
January 11, 2024 · RUB 213,840 · Hexlet

More courses on Habr Career

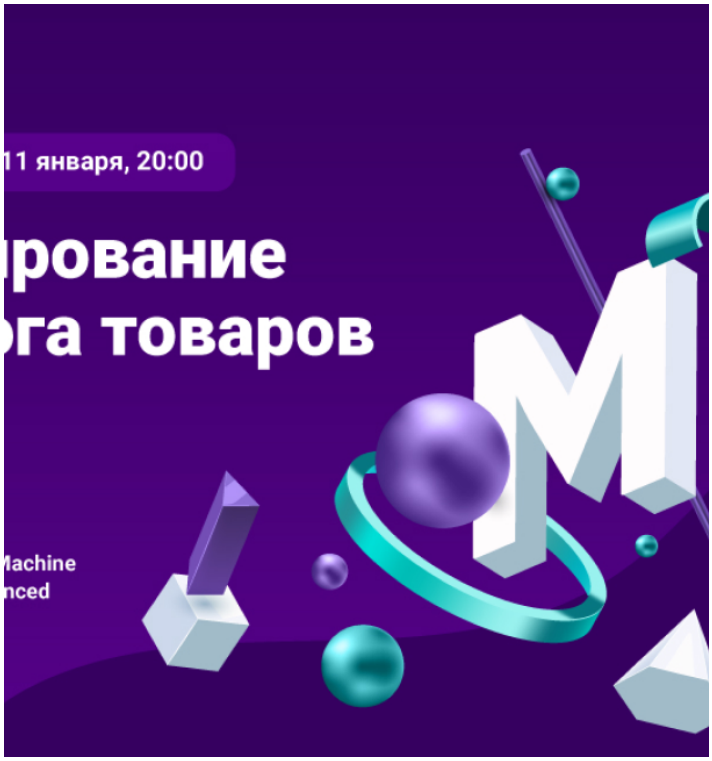
JUST A MOMENT

Делу – время, потехе – час, IT-ивентам – наш Календарь

Помоги хабраслизню найти самые праздничные видеоигры

Карты, деньги, два способа оплаты: как устроена СБП

UPCOMING EVENTS

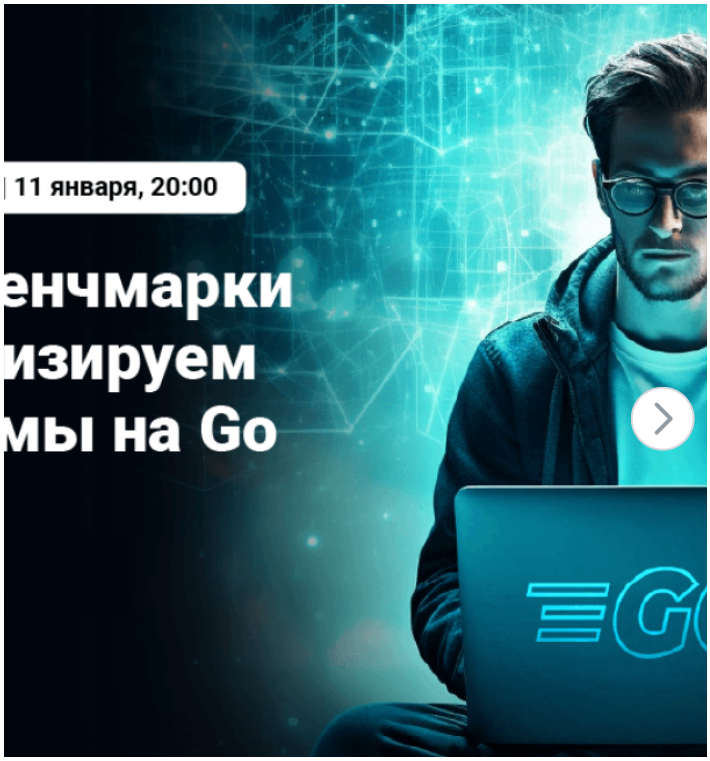


Open lesson “Product catalog ranking”

January 10 20:00

Online

More details in the calendar



Open webinar “Writing benchmarks and optimizing Go programs”

January 10 20:00

Online

More details in the calendar

Language setting

Technical support