Guides Current Setup for Fixed Wing Aircraft

Setup for Fixed Wing Aircraft

If you're looking to fly a simple delta wing, scroll to bottom.

Some have asked if BetaFlight can control a Fixed wing aircraft.

Boris' Note:

I would not trust this as nobody is testing it, but it may work.

iNav fork puts a lot of effort in airplane modes. Its better to try them.

The page is to help those that wish to Experiment and Test Fixed Wing functionality with BetaFlight.

Do Remember that YOU are the TESTER.

Reference the Betaflight resource remapping. Note: Pin re-mapping requires BetaFlight 3.1 an higher.

AresFPV asked this in Boris' ßF Thread and then did research on how to do this. His findings are:

Fixed wing success: For the few of you that are trying to get fixed wing working in BetaFlight, I've been able to have some success on the bench. I'm using BetaFlight because iNAV doesn't have the piko blx target yet. This post helped a lot: https://www.rcgroups.com/forums/showpost.php?p=36886595&postcount=45058

Select WING as type.

Basically you need to map the right servos to the pins used by motors 3 and 4 (on a wing) using the CLI Resource. Use the servos tab to reverse them if needed. Set the PWM frequency as separate from PID loop and set it to 50hz to make the servos work (you will lose all of the new ESC protocols but that's not as important on a plane). The modes and mixes work fine for me as-is and I expect that everything else will be fine as well(like OSD). I just need to add FPV to it and get it up in the air!

A note from Maine_Guy 1-25-20

The notes below did not help me in my quest to get my 4 channel micro F4 board running (although helpful). Persistence, luck, and the latest build did.

The new version of Betaflight allows for timer remapping, which may alleviate issues noted below with the timer conflicts.

--mixer is

`mixer CUSTOMAIRPLANE`

--I kept motor 1 as originally assigned

`resource MOTOR 1 B00`

--I reassigned motor outputs 2-4 to servos

`resource SERVO 1 A02`

`resource SERVO 2 A03`

`resource SERVO 3 B01`

--motor mix

`mmix 0 1.000 0.000 0.000 0.000`

--servo mix (this one I found by trial and error - since the servo outputs don't seem to match - but it works!

`smix reset`

`smix 0 3 1 100 0 0 100 0` //

`smix 1 2 0 100 0 0 100 0` //

`smix 2 4 2 100 0 0 100 0` //

`smix 3 0 3 100 0 0 100 0` //is this needed? not sure

--lastly, reassign the timer on b01

`timer B01 AF1`

A Post by RCvehicleGuy:

Well I messed with trying to get betaflight to work with an airplane. It was clearly not intended to work. What I wanted to do was use the resource command to put servos 1-4 on motor outputs 1-4 on my naze board and do a custom smix to assign stabilized roll, stabilized elevator, rc throttle, and stabilized rudder to the outputs respectively. Like a futaba radio. It sounds simple but for the life of me couldn't get it to work last night. I think something is broken somewhere, these possibilities haven't been tested I would think, seeing as betaflight is designed for acro-quad flying it really only gets tested on acro-quads.

I think the key is the smix command, either it isn't working correctly or I don't understand it well enough to work. I'll try again when I have more time and see if I can't get this working. so to output stabilized roll to servo 1 I believe the smix command is, if you just reset the smix: smix 0 0 0 100 0 0 100 0, or something like that. rule 0, servo 0 (1), source 0 (PID roll), 100 (rate), 0 (null speed), 0 (min), 100 (max), 0 (box. don't know what this is). As I said I will try again. I already tried iNav but without the resource command I can't map servo 1 to motor output 1, or don't understand how that would work at least.

Part 2: OK! I'd like to update that I have the outputs on my Naze32 working for stabilized roll, pitch, and yaw, good for a typical 3 servo 4 channel plane. I am using a 1k cycle and pid loop time. I wanted to use motor output 4, resource b07, as rudder, but it seems broken on my Naze so I moved that to 3 and decided to just use the throttle output on my receiver instead. It's interesting to watch the PIDs wind-up with throttle applied. They wind up much harder for setpoint changes and hardly at all for board movements. This should make for some interesting flight characteristics. I'm excited to get this on the Crack Laser and in the air to see how it changes the way it flies. It will be funny to be doing a rolling loop and let go of the aileron and have iterm windup keep it rolling another 3 revs or so, lol.

I'm more interested in the applications of level mode. For people new to airplanes or for strangers who ask if they can fly, having a working level mode means hey why not. On small 3d foamies at least, what is usually a touchy aerobatic machine could be turned into what amounts to a 3 channel trainer with level mode working.

I'll update again after I get it installed in the plane, reversing figured out, and a testflight in. Here is a screenshot of my smix: https://www.rcgroups.com/forums/showthread.php?2464844-Betaflight-Flight-Controller-Firmware-Discussion-Thread/page3040#post36958637

smix 0 2 0 100 0 0 100 0 smix 1 3 1 100 0 0 100 0 smix 2 4 2 100 0 0 100 0 smix 3 5 7 100 0 0 100 0

Part 3: Well the thing about it continuing a roll for three revs after you finish a rolling maneuver is true, lol. I've set the I-terms to 0. My original understanding was that there is I term decay and I term gradually becomes less and less effective as it wound up and built up error would be forgiven after a while. Will try to tune some more. I did put the gyro on a perfectly flying plane, might be impossible to make it any better. Part 4: Well I've come to the conclusion that airplanes just don't need gyro stabilization, I'll keep playing with it, because it's interesting, the plane does fly. So far the only neat thing it has accomplished are hard stops for point rolls. I have D at 3 right now and it still has too much of a dampening effect, As you approach your desired rate it steps in and backs off the rate for you. Putting D at 0 and upping rates a little. After turning up the rates and getting rid of D it's starting to make sense. Going to lower my P on ailerons and introduce D back in there and see where it's at. Flies kind of weird, I guess we'll see if it works out. Part 4: It is interesting to fly. I think the application is better suited to something that is less impacted by the extra 7 grams or so a Naze32 adds. It really isn't awful but you can tell it is carrying more weight. The plane I have it on is a Twisted Hobbies Crack Laser. It's an EPP plane for 3D aerobatics that has an AUW of 160-170g or so. I've been flying 3d foamies for many years now and it's the only thing that I'm pretty good at. Auto level works, I need to tune it in better though. With an airplane, especially a flat wing foamie like this one, the elevator doesn't control pitch proportional to servo travel. So it needs some tuning. It works really well for the roll axis, but it doesn't have enough authority on the pitch axis. I have to hold the stick full back to get it to nose up enough to fly, with 45 degree angle limits in place. The strength parameter of angle mode might need to have pitch and roll decoupled for this to work really well for airplanes. The weather took a turn here so it may be a few days or more before I can really mess with the settings and see if I can get it working well. I need to experiment with the D-term and the D setpoint weight. The most interesting application of this is getting roll hesitations to be unnaturally crisp.

Realistically using a gyro on a plane like this is more pain than gain, but it's fun to experiment. I think the best application for these boards are quads and hopefully some experimenting will be done and helicopters will work too.

The smix I'm using looks like this: `# smix` `smix 0 2 0 100 0 0 100 0` *|pairs PID roll with motor out 1|* `smix 1 3 1 100 0 0 100 0` *|pairs PID pitch with motor out 2|* `smix 2 4 2 100 0 0 100 0` *|pairs PID yaw with motor out 3|* `smix 3 5 7 100 0 0 100 0` *|Attempt to pair RC Throttle to motor out 4|*

The smix is for single servo ailerons, elevator, and rudder. It's meant to run 3 servos off the first 4 outputs and an esc off the 4th. For my Naze, maybe it's the board or the code, but I can't get anything to output of motor out 4. I ended up hooking the throttle directly to the receiver and that's fine. But for dual aileron applications you'll want to use motor out 4 for another aileron. Or for when you need other things to work. To understand what all the parameters of smix do, you can google cleanflight smix and it is explained well. I think for flying wing smix there are video tutorials of people setting this up.

My resource setup looks like this: `# resource` `resource SERVO 1 A08` `resource SERVO 2 A11` `resource SERVO 3 B06` `resource SERVO 4 B07` These are the resources used by motors 1-4 typically. Interestingly, as identified in the Betaflight App under the servos tab and in the smix commands, as you may have noticed, all the servo labels in the resource command are -1 in value compared to the values used in smix and in the servos tab. So that is one bug that could be fixed. You must use an index between 1 and 8 when resourcing a servo. I tried to resource servo 0 and servo -1 to compensate for the strangeness and it doesn't let you. These resources are also only applicable to the Naze32 I'm pretty sure. Other boards are likely to have a different IO and to determine where to assign the servos can be done with the resource command. You will be able to see where motors 1-4 are resourced and just swap in the servos to these places. ISSUES: I've been having issues with the BF CLI by the way. Sometimes I get garbage characters in the top of the display and other weird things happen sometimes. I flashed cleanflight to my Naze and in the cleanflight CLI it's all pretty and straight. Not that betaflight doesn't work, but it looks a little weird, thought I'd say something. The smix only seems to work on mixer customairplane. When I use mixer custom I can't get the servos to work. As mentioned only the first three motor outs do anything, the fourth I couldn't get to work. The last two MAY work, I don't have pins soldered on the board right now.... I'm not aware of a servo update frequency parameter but having one could be good for airplanes. Those with digital servos can use 200-300hz update frequencies. For future helicopter applications I know that some tail servos have a smaller centerpulse, sort of half way between oneshot and PWM.

I think that's everything I learned so far regarding betaflight and planes.

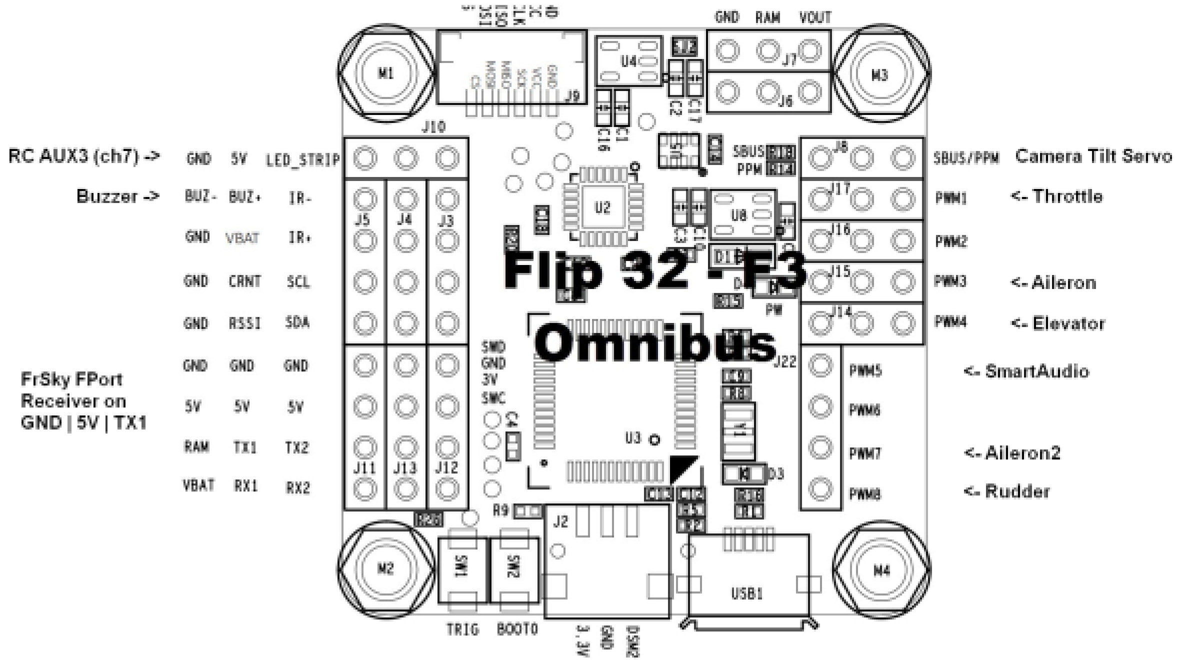
Post by: touchthebitum

Not a great video (first test in LOS flight) but it is only to illustrate that Betaflight can work pretty good with a fixed wing. I'm impressed of the landing speed in acc mode. It was very windy (about 20kmh) https://youtu.be/DpmqeoRGwgA

Betaflight-compatible FC on an EPP Foam Plane

by etheli

I wanted to add a Betaflight-compatible flight controller to a 28" EPP foam plane to get support for OSD, FrSky SBUS receiver, easy VTX frequency configuration, and stabilized flight modes. It took a bit of research and tweaking to come up with a good Betaflight configuration for airplane use. The configuration I came up with lays out on the F3 Omnibus board like this:



Using Betaflight v3.3.3, I started with a CUSTOMAIRPLANE model/mix type. The resources are allocated as follows:

```
resource MOTOR 1 B08 # motor PWM1 <– Throttle
resource SERVO 1 A02 # motor PWM4 <– Elevator
resource SERVO 2 A03 # motor PWM3 <– Aileron
resource SERVO 3 B07 # motor PWM7 <– Aileron2
resource SERVO 4 B06 # motor PWM8 <– Rudder
resource SERVO 5 B04 # SBUS/PPM <– Camera tilt [AUX2 (ch6)]
resource SERVO 6 A08 # ledstrip <– AUX3 (ch7)
```

```
# Rule Servo Source Rate Speed Min Max Box
smix 0 2 1 100 0 0 100 0 # Servo 1 Elevator <– Stabilized pitch
smix 1 3 0 100 0 0 100 0 # Servo 2 Aileron <– Stabilized roll
smix 2 4 0 100 0 0 100 0 # Servo 3 Aileron2 <– Stabilized roll
smix 3 5 2 100 0 0 100 0 # Servo 4 Rudder <– Stabilized yaw
smix 4 6 9 100 0 0 100 0 # Servo 5 <– RC AUX2 (ch6)
smix 5 7 10 100 0 0 100 0 # Servo 6 <– RC AUX3 (ch7)
```

Because of internal-processor-timer constraints the PWM2 output cannot run a servo (only a motor). Any needed servo reversing should be done in the flight controller (not in the transmitter). The servo-reversing commands look like this:

```
smix reverse 2 1 r # reverse Servo 1 Elevator
smix reverse 3 0 r # reverse Servo 2 Aileron
smix reverse 4 0 r # reverse Servo 3 Aileron2
smix reverse 5 2 r # reverse Servo 4 Rudder
smix reverse 6 9 r # reverse Servo 5 RC AUX2 (ch6)
smix reverse 7 10 r # reverse Servo 6 RC AUX3 (ch7)
```

In my setup I needed to reverse the Elevator and Rudder servos. See here for a file containing the basic configuration.

The plane flies great in gyro-stabilized (acro) mode. For tuning I needed to lower most of the values to tame oscillations, and I increased the servo 'rate' values to increase the responsiveness of the controls:

```
set p_pitch = 40
set i_pitch = 40
set d_pitch = 25
set p_roll = 25
set i_roll = 25
set d_roll = 15
set p_yaw = 50
set i_yaw = 30
set roll_srate = 100
set pitch_srate = 80
set yaw_srate = 80
```

For the self-leveling flight mode to work well, the plane should be in a somewhat pitch-up position when the accelerometer is calibrated in Betaflight. The calibration can also be trimmed up in that direction. I've posted a full build article for this plane here.

Betaflight for a Flying Delta Wing (using a Betaflight F3)

by DangerFlite

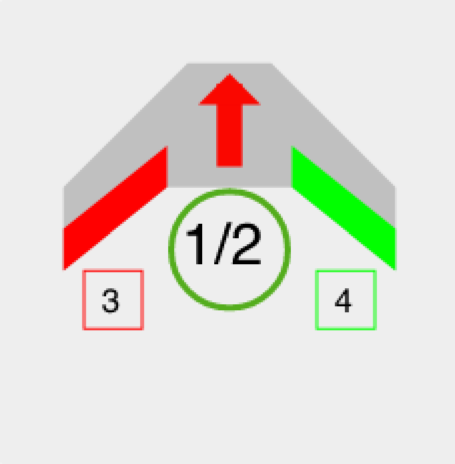
I've been reading the above text and following the thread on rcgroups trying to get a simple rate mode working on my delta wing using a Betaflight F3. I ran into problems because:

- I'm not using a Naze32, so the aforementioned resource assignments don't apply
- I'm only using two elevons, so the mixes applied above don't work for me
- Betaflight is confusing when it comes to servos (everywhere in betaflight it appears to be servo 3 / 4 for the elevons but when you assign resources, they are referred to as servo 1 / 2)
- It is a lot easier than it looks. The purpose of this guide is to help someone who doesn't already know

So far I have the flight controller working great in passthrough mode (manual flight), Acro/rate mode (stabilised), and Angle mode (self-level). Even in passthrough mode the BFF3 is still useful as it enabled the use of an SBUS receiver, telemetry, battery monitoring, and an excellent OSD.

Wiring configuration

- Connect the ESC to 'M1' pad on the Betaflight F3 (1/2 in below image)
- Connect left elevon servo to 'M3' pad on the Betaflight F3 (3 in below image)
- Connect right elevon servo to 'M4' pad on the Betaflight F3 (4 in below image)



Betaflight Settings

- Flash betaflight with full chip erase
- Configure your ports / receiver / other settings as normal
- Configuration tab > Select "Flying Wing" mixer type
- CLI tab > Run the following commands to get the servos working:
  > resource servo 1 b08

> resource servo 2 b09
> save

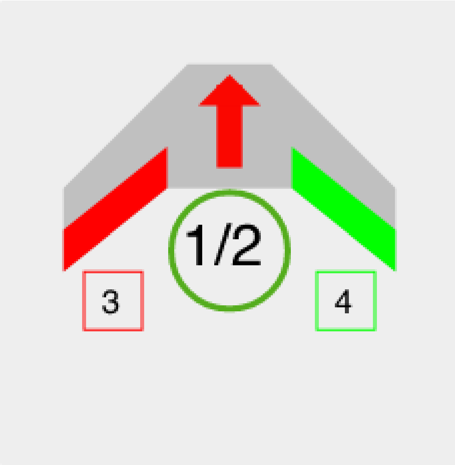
- Servo tab > Reverse either servo **3** or **4** (if needed) by setting the rate to -100 (assuming 100% of travel is needed)
- Modes tab > Set your arm switch
- Modes tab > Set up passthrough mode for manual flight (no stabilisation) and Angle mode for self levelling (useful once you get the trim right). I recommend having passthrough mode set in case of any strange tuning issues so you can take control.

Betaflight for a Flying Delta Wing (using a Matek f405 and Matek FCHUB-W PDB)

by druckgott

Wiring configuration

- Connect the ESC to 'M1' pad on the Matek FCHUB-W PDB
- Connect left elevon servo to 'SV3' pad on the Matek FCHUB-W PDB
- Connect right elevon servo to 'SV4' pad on the Matek FCHUB-W PDB



You have now to MAP the Servos to this Pins:

> resource servo 1 C07 resource servo 2 C08 save

Now it works

Some troubleshooting and extra notes:

If your plane reacts correctly to roll but wrongly to pitch (or vice versa), swap the resource assignments of the servos

My ESC refused to work at all even though it worked fine when the FC was set to the Quad X configuration. The below configuration fixed the issue. You might be fine to leave these at MULTISHOT125 or one of the DSHOT settings depending on what your ESC supports. It must be something to do with timers. The servos worked regardless of the ESC protocol selected.

- *Configuration tab > Set ESC/Motor protocol to "PWM"†*
- *Configuration tab > Enabled "Motor PWM Separated from PID Speed"†*
- *Configuration tab > Set "Motor PWM Frequency" to 480hz†*

All of the PID loop / gyro update frequency combinations I tried yielded the same results.

I had to raise pitch/roll P significantly and almost completely disable the I term to get the wing feeling responsive (and not overshoot the angles a lot). D remained basically where it was. Great having the OSD to configure this in the field!

Community

Discord

IntoFPV

Feeds

RSS

Atom

JSON

Built with Docusaurus

made with ❤️ by **VitroidFPV** and **un!t**