

## Firmware configuration

The “silverware” quadcopter software has a number of settings that can be changed by the user. The settings are placed in file **config.h**. After changing the settings, the project should be compiled again (build) and flashed to the quadcopter. Note that compiling the code also saves the changes to files automatically.

Other files can contain some settings too, notably the **pid.c** - for acro mode pids , **angle\_pid.c** , for level mode pids, and **hardware.h** - depending on the particular code, hardware specific information.

The settings are broken down into a few categories:

- Settings that affect flight control, such as rate , expo, and switch assignment
- Settings that control features
- settings that control quadcopter hardware, such as motor type and frequency

This info is for the H8 mini code, some settings may differ slightly in some builds.

### Flight control settings

This settings comprise of rate control, high and low, expo, if needed to be performed by quad itself ( for stock tx) and switch config, mainly for level mode switching.

#### Rate

The rate is changed by the following lines in control.h file. The unit is degrees / second, and is for low rates. So on low rates (stock tx) the quad will perform a 180 degree turn in 1 second at maximum control deflection.

```
// rate in deg/sec
// for low rates ( acro mode)
#define MAX_RATE 180.0f
#define MAX_RATEYAW 180.0f
```

High rates are controlled by a multiplier, which is applied to the low rates above.

```
// multiplier for high rates
// devo/module uses high rates only
#define HIRATEMULTI 2.0f
#define HIRATEMULTIYAW 2.0f
```

### Switches (auxiliary channels)

In this firmware, the switches/buttons control quadcopter functions such as level/acro mode , headless, dual rates and flips.

This functions can be assigned to any channel, if using a devo or a module. The number of channels

depends on radio protocol specifications.

For the stock tx, only some buttons can be used. The H8 tx supports only the rate button (called expert here) and using the trims as switches. The flip button can only be used for flips.

```
// level / acro mode switch
// CH_AUX1 = gestures
// 0 - flip
// 1 - expert
// 2 - headfree
// 3 - headingreturn
// 4 - AUX1 ( gestures <<v and>>v)
// 5 - AUX2+ ( none )
// 6 - Pitch trims
// 7 - Roll trims
// 8 - Throttle trims
// 9 - Yaw trims
// 10 - on always
// 11 - off always
// CH_ON , CH_OFF , CH_FLIP , CH_EXPERT
// CH_HEADFREE , CH_RTH , CH_AUX1 , CH_AUX2 , CH_AUX3 , CH_AUX4
// CH_PIT_TRIM, CH_RLL_TRIM
#define LEVELMODE CH_AUX1
```

### Stock tx

The trim buttons do not do anything in the stock tx. To use the trims for controlling a function, set the function channel to the trim (such as CH\_PIT\_TRIM - pitch trims) and enable “#define USE\_STOCK\_TX”

*Do not enable “#define USE\_STOCK\_TX” if you want to use a devo or tx module and the trims are set as a function!*

### Devo tx

In the above code the level function is set to “CH\_AUX1” which is the “gestures channel”. This is the default, since it works with any tx. For a devo, you could set it to “CH\_RTH” instead, for example (return home channel)

The devo channels are assigned as follows in the devo itself:

```
CHANNEL1      // Aileron
CHANNEL2,     // Elevator
CHANNEL3,     // Throttle
CHANNEL4,     // Rudder
CHANNEL5,     // Leds / Inverted
CHANNEL6,     // Flip
CHANNEL7,     // Still camera
CHANNEL8,     // Video camera
CHANNEL9,     // Headless
CHANNEL10,    // Return To Home
CHANNEL11,    // Calibrate
```

~~The quad firmware supports “Flip” (channel6, called “0” in config.h), “headless” (channel9, called “2” in config.h) and “return home” (channel10, called “3” in config.h)~~

**New Versions of H8mini and H101 code support DEVO\_CHAN\_5 - DEVO\_CHAN\_11. The above text is still valid for the Acro only versions of H8mini and H101!**

## Software features

The configuration file also contains options to enable a number of features, which may be needed on a personal basis.

Feature list:

1. auto throttle
2. lower throttle (mix)
3. motor filter
4. clipff
5. throttle transient compensation
6. gyro lpf frequency (software filter)
7. Gyro lpf filter (hardware)

### Auto throttle feature:

The auto throttle will keep a constant upwards thrust as the quad leans in one direction. Used in level mode, usually.

Example: Suppose the quadcopter is at a 45 degrees angle. Because of the angle, only half the thrust will contribute to maintaining thrust. The other half will provide lateral acceleration. With “auto throttle” on, the software will increase throttle to compensate. As a result, the quad will maintain altitude better while maneuvering.

```
#define AUTO_THROTTLE
```

### Lower throttle feature

With this option enabled, the quadcopter will lower the throttle in order to maintain better control near full throttle. Without it (default) the quadcopter will usually lean slightly in one direction at full throttle, as there may not be enough control ability. (the motors can't increase over maximum)

```
#define MIX_LOWER_THROTTLE
```

### Motor filter feature

The motor filter is a 3 tap hanning fir filter that is applied to the motor output. Enabled by default, as it uses little resources.

```
#define MOTOR_FILTER
```

### Clipff feature

The clip feedforward feature is an experimental feature that is supposed to help improve control near

full throttle, and also help with vibration issues. It may also improve maximum throttle by a small amount in presence of vibration.

```
#define CLIP_FF
```

### Throttle transient compensation feature

The TTCF feature improves throttle response of the quadcopter. The curve has been measured for 6mm motors, and it may not work the same with other motors. With this feature on, the quadcopter will react faster to throttle changes.

This feature only works on the throttle stick, it does not affect controls in any other way.

```
#define THROTTLE_TRANSIENT_COMPENSATION
```

### Gyro lpf filter (software)

The software lpf filter is a custom gyro filter, additional to the hardware filter which is present in the gyro device. The hardware filter is set around 43Hz by default, but for additional vibration removal, a software filter can be added. Recommended: 2nd order 88, 4th order 88 .

Enabling the filter may require slight lowering of D term, if set near maximum. A lowering of D term may need lowering of P term. As such , it's probably a good idea to have the filter enabled before you tune a quad.

```
// set only one below
//#define SOFT_LPF_1ST_023HZ
//#define SOFT_LPF_1ST_043HZ
//#define SOFT_LPF_1ST_100HZ
//#define SOFT_LPF_2ND_043HZ
#define SOFT_LPF_2ND_088HZ
//#define SOFT_LPF_4TH_088HZ
//#define SOFT_LPF_4TH_160HZ
//#define SOFT_LPF_4TH_250HZ
//#define SOFT_LPF_NONE
```

### Gyro lpf filter (hardware)

The hardware gyro lpf filter is part of the gyro integrated circuit. It's setting changes the frequency of the internal filter. The internal filter also has a latency which may affect pid D term. A lower latency may be better, however vibration will limit the usefulness of higher gyro frequencies.

```
// gyro filter 0 = 250hz delay 0.97mS
// gyro filter 1 = 184hz delay 2.9mS
// gyro filter 2 = 92hz delay 3.9mS
// gyro filter 3 = 41hz delay 5.9mS (Default)
// gyro filter 4 = 20hz
// gyro filter 5 = 10hz
// gyro filter 6 = 5hz
// gyro filter 7 = 3600hz delay 0.17mS
```

```
#define GYRO_LOW_PASS_FILTER 3
```

From:

<http://sirdomsen.diskstation.me/dokuwiki/> - **Silverware Wiki**

Permanent link:

<http://sirdomsen.diskstation.me/dokuwiki/doku.php?id=config.h&rev=1478942923>

Last update: **2016/11/12 10:28**

