

Overview

To use all AutoQuad ESC32 features calibrations are required. The calibration measures and calculates drive characteristics and must therefore be performed for every different motor, propeller and voltage (LiPo cell count) combination.

There are two different calibrations, the current limiter calibration and the feed forward calibration. The current limiter calibration is required for the virtual current limiter which performs better than the standard PI controller. The feed forward calibration is required for the closed loop control modes (RPM and thrust).

You can find more info on [ESC32 Current limiter calibration here](#)

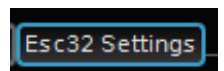
Prerequisites

- Attach your motor with prop on a non movable surface and make sure the motor is attached so it can't move.
- Use fresh lipo's
- use the prop you intend to use otherwise the data will not be valid.
- **Be careful!!!** both calibrations will ramp up the motor output to max so use a wide open space without audience presence.

Calibration with QGroundControl

Connect the ESC32 using your FTDI interface to the computer and start QGCS. Make sure the AutoQuad widget is visible in the engineering tab. Click on Esc32 Settings.

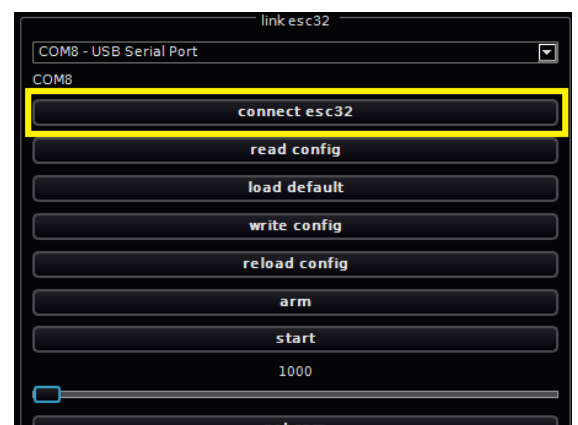
Proceed with connecting the esc32 by clicking on the 'connect esc32' button. That one!!, make sure you did NOT use any other connect button only that one.



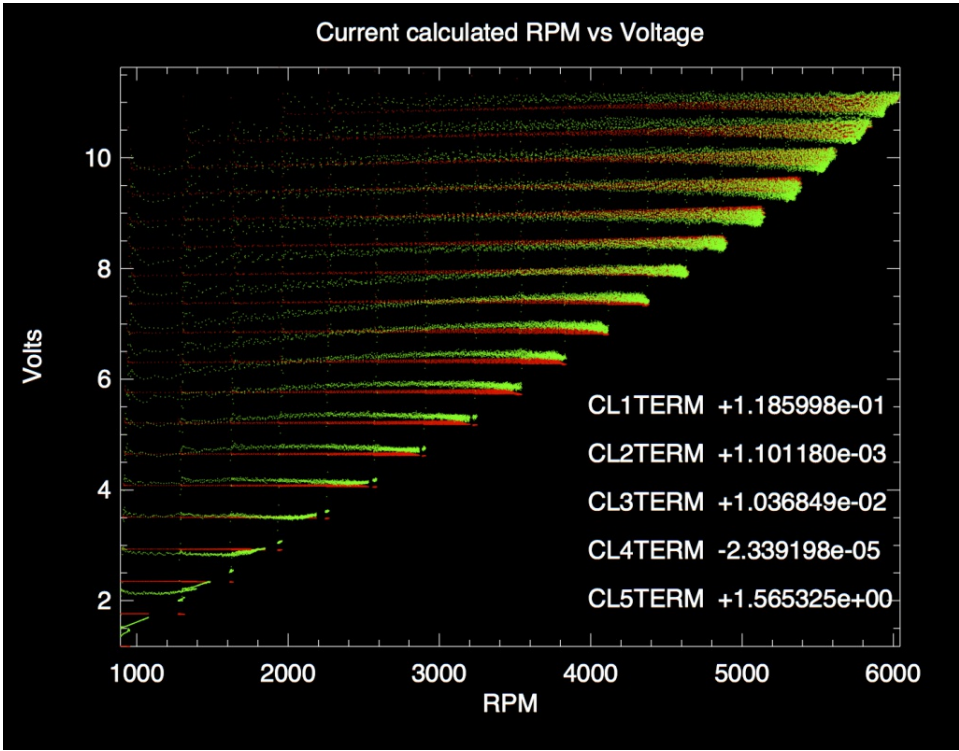
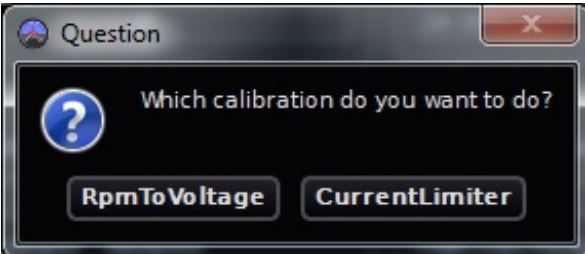
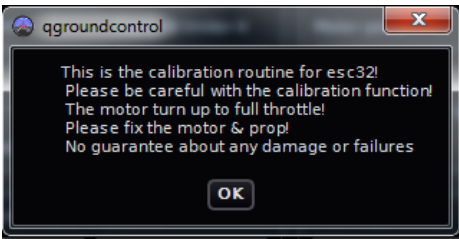
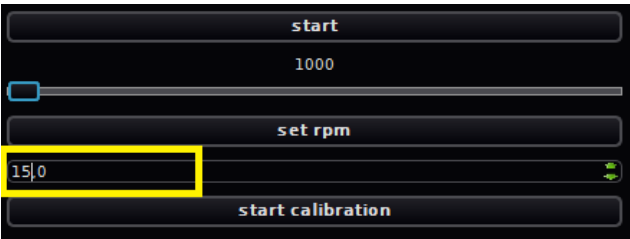
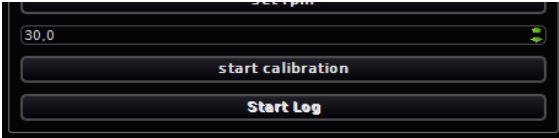
To set the maximum current to use in the calibrations you can adjust the default (30A) to any number. It will probably be good practice to start low and watch the first calibration. If you table, motor and room is still intact you can set a higher number.

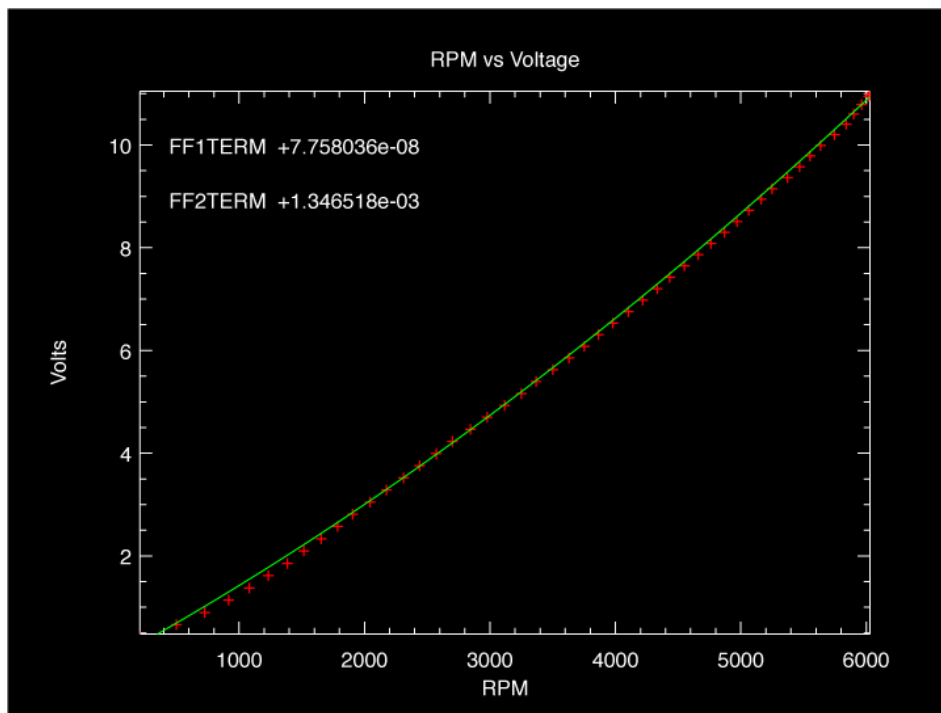
Final check and click on 'start calibration'. The following warning will be displayed.

Next you can select the two calibrations. Select the one you need (or do both).



The calibration will now start and take some time. The calibration sequence will ramp up the duty cycle in several steps and measure rpm, current, duty cycle and voltage. It can produce output like this if you graph the data (next versions or if you use the linux version with pplot). In the current QGCS a comma separated file is generated in the QGCS directory that can be imported into a spreadsheet program.





After the calibration the current limiter (1-5) or FF (1-2) data is displayed and entered in the Esc32 PID screen.

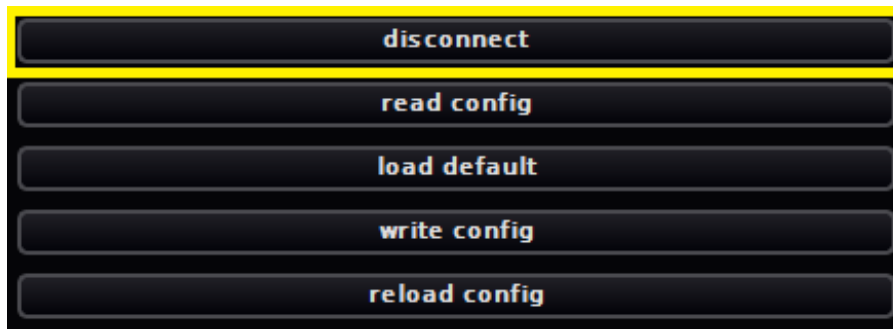
Now click on 'write config' to save them to the ESC32 memory.

Esc32 PID	
Esc32 P	Current limiter 1
0.750	1.083585e-01
Esc32 I	Current limiter 2
0.001	1.118480e-03
FF1 Term	Current limiter 3
6.602521e-08	2.000165e-02
FF2 Term	Current limiter 4
1.307364e-03	-1.675907e-05

Esc32	
Current limiter 5	Blanking Microsec.
1.547213e+00	30.000

disconnect
read config
load default
write config
reload config

Final check. Click on disconnect esc32 and again on connect esc32 for a definitive check if the values have been stored to flash and can be retrieved by using connect esc32



You have to manually enter the calibration data in your other esc32's.

Calibration via command line (Ubuntu)

When using the QGCS, the executables for calibration are in the software, but when using the commandline binaries on linux or OSX you need to compile them.

Prerequisites

1. ESC32 source on the [google code page](#)
2. eigen3 development library for your distro
3. plplot development library for your distro

Install the required packages

```
sudo apt-get install subversion build-essential libplplot-dev libeigen3-dev
```

Add the current user to the dialout group to run esc32Cal as normal user

```
sudo usermod -a -G dialout $USER
```

After the next login esc32Cal can be run as normal user.

Checkout the latest ESC32 code

```
svn checkout http://esc32.googlecode.com/svn/trunk/ esc32
```

Edit Makefile (esc32/ground) to adapt Eigen3 library path

```
esc32Cal.o: esc32Cal.cc esc32.h
```

```
$(CC) -c $(ALL_CFLAGS) esc32Cal.cc -I/opt/local/include -I/usr/include/eigen3
```

Compile esc32Cal

```
make esc32Cal
```

Note: The warnings can be ignored!

Available Parameters

-cl	Current Limiter calibration
-r2v	Feed Forward calibration (RPM to Voltage)
-a amps	Limits current during calibration
-p device	USB to TTL converter port
-b baud	Baud rate
-t file	Telemetry output file

The device port (-p switch) must be set and depends on the used converter (“dmesg” can help to identify the port). It’s **important to limit the maximum current** (-a switch) to match the drive combination (motor/propeller/voltage). A value too high can burn the motor or ESC! Ctrl+C can be used to terminate esc32Cal.

Current Limiter calibration

```
./esc32Cal --cl -p /dev/ttyUSB0 -a 15
```

At the end of the calibration the CLxTERM values are displayed:

```
#define DEFAULT_CL1TERM    +3.358974e-01
#define DEFAULT_CL2TERM    +1.124178e-03
#define DEFAULT_CL3TERM    +3.390607e-02
#define DEFAULT_CL4TERM    -3.829096e-05
#define DEFAULT_CL5TERM    +1.367444e+00
```

Feed Forward calibration

```
./esc32Cal --r2v -p /dev/ttyUSB0 -a 15
```

At the end of the calibration the FFxTERM values are displayed:

```
#define DEFAULT_FF1TERM    +8.008604e-08
#define DEFAULT_FF2TERM    +1.249541e-03
```