

Battery Monitoring

Betaflight has a battery monitoring feature. The voltage of the main battery can be measured by the system and used to trigger a low-battery warning [buzzer \(Buzzer.md\)](#), on-board status LED flashing and LED strip patterns.

Low battery warnings can:

- Help ensure you have time to safely land the aircraft
- Help maintain the life and safety of your LiPo/LiFe batteries, which should not be discharged below manufacturer recommendations

Minimum and maximum cell voltages can be set, and these voltages are used to auto-detect the number of cells in the battery when it is first connected.

Per-cell monitoring is not supported, as we only use one ADC to read the battery voltage.

Supported targets

All targets support battery voltage monitoring unless status.

Connections

When dealing with batteries **ALWAYS CHECK POLARITY!**

Measure expected voltages **first** and then connect to the flight controller. Powering the flight controller with incorrect voltage or reversed polarity will likely fry your flight controller. Ensure your flight controller has a voltage divider capable of measuring your particular battery voltage.

Naze32

The Naze32 has an on-board battery divider circuit; just connect your main battery to the VBAT connector.

CAUTION: When installing the connection from main battery to the VBAT connector, be sure to first disconnect the main battery from the frame/power distribution board. Check the wiring very carefully before connecting battery again. Incorrect connections can immediately and completely destroy the flight controller and connected peripherals (ESC, GPS, Receiver etc.).

CC3D

The CC3D has no battery divider. To use voltage monitoring, you must create a divider that gives a 3.3v MAXIMUM output when the main battery is fully charged. Connect the divider output to S5_IN/PA0/RC5.

Notes:

- S5_IN/PA0/RC5 is Pin 7 on the 8 pin connector, second to last pin, on the opposite end from the GND/+5/PPM signal input.
- When battery monitoring is enabled on the CC3D, RC5 can no-longer be used for PWM input.

Sparky

See the [Sparky board chapter \(boards/Board%20-%20Sparky.md\)](#).

Configuration

Enable the VBAT feature.

Configure min/max cell voltages using the following CLI setting:

vbat_scale - Adjust this to match actual measured battery voltage to reported value (which may be displayed via the status command)

vbat_max_cell_voltage - Maximum voltage per cell, used for auto-detecting battery voltage in 0.1V units, i.e. 43 = 4.3V

vbat_min_cell_voltage - Minimum voltage per cell; this triggers battery-critical alarms, in 0.1V units, i.e. 33 = 3.3V

vbat_warning_cell_voltage - Warning voltage per cell; this triggers battery-warning alarms, in 0.1V units, i.e. 34 = 3.4V

vbat_hysteresis - Sets the hysteresis value for low-battery alarms, in 0.1V units, i.e. 1 = 0.1V

e.g.

```
set vbat_scale = 110
set vbat_max_cell_voltage = 43
set vbat_min_cell_voltage = 33
set vbat_warning_cell_voltage = 34
set vbat_hysteresis = 1
```

Current Monitoring

Current monitoring (amperage) is supported by connecting a current meter to the appropriate current meter ADC input (see the documentation for your particular board).

When enabled, the following values calculated and used by the telemetry and OLED display subsystems:

- Amps
- mAh used
- Capacity remaining

Configuration

Enable current monitoring using the CLI command:

```
feature CURRENT_METER
```

Configure the current meter type using the `amperage_meter_type` settings here:

Value	Sensor Type
NONE	None
ADC	ADC/hardware sensor
VIRTUAL	Virtual sensor

Configure capacity using the `battery_capacity` setting, in mAh units.

If you're using an OSD that expects the multiwii current meter output value, then set `multiwii_amperage_meter_output` to ON (this multiplies amperage sent to MSP by 10 and truncates negative values)).

ADC Sensor

The current meter needs to be configured so the value read at the Analog to Digital Converter (ADC) input matches actual current draw. Just like you need a voltmeter to correctly calibrate your voltage reading you also need an ammeter to calibrate the current sensor.

Unlike voltage sensing which is usually quite good from the factory, current sensing varies significantly from board to board and should be calibrated.

It is recommended to set `multiwii_amperage_meter_output` to OFF when calibrating ADC current sensor.

To measure the current a linear response device is used that converts the current traveling through it to a voltage to be read by the ADC on your flight controller.

The maximum voltage that the flight controller can read is 3.3V (3300 mV), this is usually the limiting factor on the maximum measurable current.

Most sensors use a shunt resistor to measure current however there are a few that use hall effect sensors.

The flight controller uses the following equation to convert the measured ADC voltage to a current.

```
Current(Amps) = ADC (mV) / amperage_meter_scale * 10 + amperage_meter_offset / 1000
```

Where the calibrations are:

Setting	Description
<code>amperage_meter_scale</code>	The scaling factor in mV/10A
<code>amperage_meter_offset</code>	The offset in mA

This is in the mathematical form of $y = x/m + b$ and with a few measurements along this line you can calibrate any combination of sensor and flight controller to a high accuracy.

Calibrate using an ammeter

!!Important: Always take off your props before doing any testing!!

To calibrate your flight controller with a current meter follow these steps.

1. Make a copy of [this google sheet \(https://docs.google.com/spreadsheets/d/1kL-X_FT9x2oqrwQEctDsEUhgdY19upNGc78M6FfjXY/\)](https://docs.google.com/spreadsheets/d/1kL-X_FT9x2oqrwQEctDsEUhgdY19upNGc78M6FfjXY/). It will do all of the maths for you.
2. Hook your ammeter up in series with your drone and a charged battery. I suggest an XT60 extender with one lead cut. Now your ammeter will be displaying the true current draw of your system.
3. Connect to your flight controller through the configurator and check your current calibrations. Change them in the google sheet if needed.
4. Use the motor tab to increase the throttle and change the current draw of the drone to around 1 A on the ammeter (it does not matter if it is not exact).
5. Switch back to the power and battery tab and record current from the ammeter in the measured current column and the current reported by Betaflight in the flight controller current column (both in amps, to 2 decimal places).
6. Repeat this measurement (steps 4 and 5) 3 or more times at various currents from 0 to 5 Amps (make sure not to go over your ammeter rated current).
7. Once this is done make sure the results are linear on the graph and that the regression value is green. You can now update to the new calibration values and enjoy accurate battery usage information.

The same method can be applied to both hall effect sensors and shunt resistors. Shunt resistors will usually have an offset of less than +/- 1000 mA however hall effect sensors offsets will be much higher.

Note that while your calibration may be correct there is still a maximum current measure that you may exceed at maximum throttle, this will cause the current recorded by the flight controller to ceiling at this value even if the actual current is higher.

As a result the reported mAh used may be less than is actually used, always keep an eye on the battery voltage as well.

If you do not want to use google sheets then simply use some other tool that preforms a linear regression on the dataset. Multiply `amperage_meter_scale` used in testing by the slope and subtract the intercept in mA from `amperage_meter_offset` to get the corrected calibration values.

Virtual Sensor

The virtual sensor uses the throttle position to calculate an estimated current value. This is useful when a real sensor is not available. The following settings adjust the virtual sensor calibration:

Setting	Description
<code>amperage_meter_scale</code>	The throttle scaling factor [centiamps, i.e. 1/100th A]
<code>amperage_meter_offset</code>	The current at zero throttle (while disarmed) [centiamps, i.e. 1/100th A]

There are two simple methods to tune these parameters: one uses a battery charger and another depends on actual current measurements.

Tuning Using Actual Current Measurements

If you know your craft's current draw (in Amperes) while disarmed (I_{min}) and at maximum throttle while armed (I_{max}), calculate the scaling factors as follows:

```
amperage_meter_scale = (Imax - Imin) * 100000 / (Tmax + (Tmax * Tmax / 50))
amperage_meter_offset = Imin * 100
```

Note: Tmax is maximum throttle offset (i.e. for max_throttle = 1850, Tmax = 1850 - 1000 = 850)

For example, assuming a maximum current of 34.2A, a minimum current of 2.8A, and a Tmax max_throttle = 1850:

```
amperage_meter_scale = (Imax - Imin) * 100000 / (Tmax + (Tmax * Tmax / 50))
                     = (34.2 - 2.8) * 100000 / (850 + (850 * 850 / 50))
                     = 205
amperage_meter_offset = Imin * 100 = 280
```

Tuning Using Battery Charger Measurement

If you cannot measure current draw directly, you can approximate it indirectly using your battery charger.

However, note it may be difficult to adjust amperage_meter_offset using this method unless you can measure the actual current draw with the craft disarmed.

Note:

- This method depends on the accuracy of your battery charger; results may vary.
- If you add or replace equipment that changes the in-flight current draw (e.g. video transmitter, camera, gimbal, motors, prop pitch/sizes, ESCs, etc.), you should recalibrate.

The general method is:

1. Fully charge your flight battery
2. Fly your craft, using >50% of your battery pack capacity (estimated)
3. Note Cleanflight's reported mAh drawn
4. Fully charge your flight battery again, noting the amount of mAh recharged
5. Adjust amperage_meter_scale to according to the formula given below
6. Repeat and test

Given (a) the mAh recharged and (b) the cleanflight reported mAh drawn, calculate a new amperage_meter_scale value as follows:

```
amperage_meter_scale = old_amperage_meter_scale * (mAh_recharged /
cleanflight_reported_mAh_drawn)
```

For example, assuming:

- An amount recharged of 1500 mAh
- A Cleanflight reported current drawn of 2000 mAh
- An existing amperage_meter_scale value of 400 (the default)

Then the updated amperage_meter_scale is:

```
amperage_meter_scale = old_amperage_meter_scale * (mAh_recharged /  
cleanflight_reported_mAh_drawn)  
= 400 * (1500 / 2000)  
= 300
```