



# CAN\_Drv module configuration reference manual

© Basecamelectronics

Version: 1.05

Edit: 13. Aug. 2020

## Introduction

CAN\_Drv is a DC brushless motor driver with the CAN-bus interface, that is intended to be used with the CAN-enabled SimpleBGC32 controller in a camera stabilization system, replacing the on-board drivers.

CAN driver allows to build 3-axis stabilizer based on a modular scheme (example:

[https://www.basecamelectronics.com/files/SBGC\\_CAN\\_Modular\\_System\\_Connection\\_Diagram.pdf](https://www.basecamelectronics.com/files/SBGC_CAN_Modular_System_Connection_Diagram.pdf)). System supports up to four additional motors for the other tasks. Such modular scheme benefits by the optimized wiring between modules that is perfectly immune to EMI noise, and better motor control algorithm compared to conventional SimpleBGC32 controllers with the integrated drivers.

The module specifications and pin-out could be found on the product page:

[https://www.basecamelectronics.com/can\\_driver/](https://www.basecamelectronics.com/can_driver/) This manual describes the configuration and calibration of CAN-driver.

## Assigning drivers to the stabilization axes

Before configuring each driver, you must assign which physical driver will match each of the stabilization axes. Each driver has a unique identifier and must have a unique address on the CAN bus. In the SimpleBGC32 system, the addresses are denoted as CAN\_Drv # 1..7. The address can be set by the hardware-specific jumpers on the driver card. The address is from 1 to 7. If the jumpers are not soldered, then the software address assignment is allowed. To do this, start scanning all connected modules (it can last long enough, wait until the end). Then assign the addresses to the modules and write these settings to the EEPROM. To determine which module exactly matches the ones found in the list, you can sequentially connect the modules to the bus one at a time and assign them addresses.

When modules received addresses from the list CAN\_Drv # 1..7, you can use them wherever the outputs to the motors are assigned. For the three main stabilization axes, it can be done in the section "Hardware" → "Motor configuration" → "Motor outputs".

Table 1: Address selection by jumpers

SW3	SW2	SW1	Address
0	0	0	software-assigned
0	0	1	drv#1
0	1	0	drv#2
0	1	1	drv#3
1	0	0	drv#4
1	0	1	drv#5
1	1	0	drv#6
1	1	1	drv#7

## Settings

In the "CAN Modules" tab there are two types of settings: the "CAN Driver software configuration" - parameters for a particular motor, and the "CAN Driver hardware configuration" - parameters for a particular board schematics version. **WARNING!** It is not recommended to make changes to the "CAN Driver hardware configuration" in case the driver works correctly, because entering incorrect values can destroy the board.

If the CAN driver is used to stabilize one of the main axes (YAW, PITCH, ROLL), then some parameters are taken from other GUI tabs. These fields can be left blank in the "CAN Modules" tab; the system doesn't use them.

Table2: Parameters of CAN driver overridden in case of use for the main stabilization axes.

CAN_Drv parameter	The preemptive parameter from the other tabs
Number of poles	Hardware → NUM.POLES
Encoder type	Encoders → Encoder type
Encoder/motor gear ratio	Encoders → Motor/encoder gearing ratio
Encoder El. field offset	Encoders → El. field offset
Encoder Zero angle offset	Encoders → Offset
Heat model - heating factor	Encoders → Heating factor
Heat model - cooling factor	Encoders → Cooling factor

Max. effective power	Hardware → Motor Power Settings → Power
Magnetic link of rotor to stator	Hardware → MAGNETIC LINK.

Table3: Motor-related parameters (software configuration)

Parameter	Min	Max	Description
Module to edit			Select a module for editing parameters, #1..7
Motor resistance (single phase), Ohm	0	655,35	The resistance of motor single phase in Ohms (resistance between two motor leads divided by two).
Motor inductance Ld (single phase), mH	0	-	Ld (d-axis inductance) of the motor phase in mH. (the smallest inductance of the motor between the two motor leads divided by two). It is unlimited when editing, usually less than 100 mH.
Motor inductance Lq/Ld ratio	1	100	Inductance Lq - is the biggest inductance of the motor. Lq / Ld is the ratio of the inductances.
Motor temperature at calibration	-128	127	Temperature of motor measured by the external temp. sensor in the motor when RL calibration is done. If sensor is not connected, environment temperature is used, but you can write a correct value manually. Driver uses this value for the resistance-over-temperature compensation.
Current loop bandwidth, rad/s	0	15000	It determines the bandwidth of the closed-loop system. The higher the value, the greater is the PI current control gain, the more accurate and faster reaction rate, but the greater is the acoustic noise from the current loop. In the stabilization task it may add additional gain to the main loop.
Current loop I-term gain	0	255	The P/I ratio for the PI current regulator. The range 0..255 corresponds to the ratios from 0.1 to 10 with the exponential scale, where 127 = 1.0.

Over-current protection (software avg), A	0	655	RMS value of the software current limit at which a protection will trigger (emergency stop).
Over-speed protection, RPM	0	65535	Threshold for the motor speed at which a protection will trigger (emergency stop)
Current limiter (clipping), A	0	655	If RMS current exceeds this threshold, it will be limited (clipped). Motor still tracks the given setpoints for the torque but starting from some point it's limited. Error is not thrown. To use this function, a current PI-loop should be enabled and configured properly ("current loop bandwidth" != 0).
PWM frequency, kHz	18	64	PWM frequency of the power stage FETs. The higher is the value, the higher are losses on the keys and their heating, but the lower is resolution. Also it defines the requirements for the capacitance of the filtering capacitors. For low-resistance motors it may be useful to set the frequency higher.
<b>Temperature settings for motor and driver</b>			
Compensate R over temperature		x	Driver will compensate for changes in motor's resistance varying over the temperature. It uses an external temp. sensor or a virtual temperature model. Such compensation is required in synchronous mode or when the current PI-loop is not used.
Allow to use virtual temperature model for R compensation			If external temp. sensor is not connected, this option allows to use virtual temperature model of the motor. It's required to properly setup this model. The algorithm is described below in this document.
Virtual model - heating factor	0	255	This parameter determines how quickly the motor warms up, depending on the current applied to it.
Virtual model - cooling factor	0	255	This parameter determines how quickly the motor cools down due to difference with the environment temperature.
Over-temperature protection (int. sensor), C	0	124	The threshold for over-temperature protection, where temperature is measured on the driver's PCB.
Over-temperature protection (ext. sensor), C	0	124	The threshold for over-temperature protection, where temperature is measured (or estimated by the model) on the motor. Ext. temp. sensor connection is described in the section "Protection systems".

Over-heat action (temp. sensor/ model):			Action required when the over-temperature protection is triggered: <ul style="list-style-type: none"> <li>• No action – protection is disabled</li> <li>• Gradually limit power – do not throw error but clip the max. current</li> <li>• Generate emergency stop error – interrupt normal system operation and stop all motors</li> </ul>
<b>Settings for the tasks different from the main stabilization task*</b>			
Number of poles*	1	255	Number of motor poles (generally, it's a number of magnets)
Max. effective power*	0	255	Maximum power applied to the motor. The commanded torque setpoint of 100% corresponds to this value and uses POWER/255 of available voltage. For example, if you increase the voltage of power supply but want to keep the same torque response, you have to reduce the POWER parameter correspondingly.
Max. speed limit, rpm	0	65535	The motor speed limit for a servo-mode.
Encoder type*	-	-	For the list of the supported encoders and more information on connecting and configuring encoders please refer to the manual <a href="https://www.basecamelectronics.com/files/SimpleBGC_32bit_Encoders.pdf">https://www.basecamelectronics.com/files/SimpleBGC_32bit_Encoders.pdf</a>
Encoder/motor gear ratio*	-	-	The ratio of the encoder's angle of rotation to the angle of the motor, and the inversion (positive value - encoder is non-inverted, negative - it is inverted).
Encoder El. field offset*	-32767	32767	Electrical angle offset. It is calibrated by the system. <b>Set zero value if encoder is not configured!</b>
Encoder Zero angle offset*	-32767	32767	Used for a servo-mode only (position control loop) and defines the position where angle is zero. <b>Set zero value if encoder is not configured!</b>
Magnetic link of rotor to stator*	0	655.35	A running motor generates a back-EMF. To correct it, specify this constant.  Standard dimension: $Ke = V \cdot s / rad$ (Volt * second / radian)  Conversion into our units: $Ke\_basecam = Ke \cdot 18,70328124 \cdot NumPoles$

Torque constant (N*m/A)	0	655	Motor torque-to-current characteristic
Moment of inertia (g*m <sup>2</sup> )	0	13107	Full moment of inertia of the system motor + payload
PID controller gains. Over speed/ Over Angle: P I LPF frequency, Hz	0 0 0	655.35 655.35 500	The parameters of the PI regulator over speed and position, proportional and integral parts. They are used for a servo-mode only. If speed control mode is used, there is no need to fill in the "Over Angle" parameters. In the position control mode, the parameters of both PI-loops are required. Additionally, a low-pass filter may be used to cut off high-frequency noise.
Notch filters Center frequency, Hz Width, Hz Gain, Db	0 0 -127	500 255 127	Filters can greatly improve the quality of PID controller operation in some cases. These filters can reject narrow bandwidth resonances in case when the system has a pronounced mechanical resonance. It allows to increase the feedback gain and get more accurate and stable response of the PI regulator. But this filter will be useless if oscillations appear in the broad frequency range. In this case, consider using a low-pass filter. With the parameter <b>Gain</b> , you can control the effect of the notch filter. Positive value to reject, negative value to amplify. <i>Hint:</i> you can detect the resonance frequency by the oscilloscope or by the audio spectroscope on smartphone, if vibration is audible.

\* Parameters in this group have no effect if using CAN Drv for the main stabilization axes. See Table 2 for replacing parameters.

Table 4: Board-related parameters (hardware configuration)

**WARNING!** In most cases hardware-related parameters should not be changed, unless you are a developer of the CAN\_Drv board! The hardware settings are protected against accidental writing by the password: **197342682**

FET drive output type	Driver output Type: 6-channel complementary PWM (high and low side with dead-time - separate channels for controlling low-side and high-side gate of the transistor with dead-time) 3-channel PWM with three driver turn-on outputs
-----------------------	---

PWM output polarity	The polarity of the timer output for the low-side and high-side gate is separate. It can take two values: "high" and "low". "High" - the transistor is open at a high logic level at the microcontroller pin, "low" - at a low level.
Dead-time, ns	A dead-time control circuit eliminates shoot-through currents through the main power FETs during switching transitions and provides high efficiency for the buck regulator. It is a delay between the opening/closing of a low-side gate and the closing/opening of the high-side.
Op-amp type	Type of operational amplifier: Internal - built-in in the MCU External - external circuit None (gain = 1) - current sensor is used
Op-amp gain	The gain of an operational amplifier.
Shunt resistance, mOhm ICS sensitivity, mV/A	In the case of using the op-amp, the value of this field indicates the resistance of the shunt. In the case of using a current sensor - it indicates its sensitivity.
Current sensor R offset ratio	The ratio of resistive divider that adds a voltage offset for a current sensing circuit. <i>Divider shifts the output of the op-amp to make it equal about half of the MCU supply voltage, when no current flows through the shunt resistor.</i>
ADC rise pulse delay, ns	The time delay adds after switching transitions before current digitizing.
ADC sampling time factor	The factor of the duration of current digitizing.
Battery voltage sensor R divider	The voltage divider ratio.
Max PWM duty cycle, %	The maximum permissible PWM modulation factor is specified for the 18kHz frequency. The higher is the PWM frequency, the lower is the ratio. Inside the driver, a calculation is made using the formula (MMI - max modulation index): $MMI = 1 - (PWM\ Freq\ in\ kHz)/18 * (1 - (Max\ PWM\ duty\ cycle\ in\ \%/100))$
Over-current protection threshold, A	A threshold for a current at which the fast hardware short-circuit protection will trigger.

# Calibrations

## LR calibration

**WARNING:** *It is recommended to remove the camera from the gimbal during this calibration, because generated vibrations can adversely affect the camera.*

Firstly, the resistance is calibrated, when the motor is supplied with a constant voltage (equals to the power supply voltage). The stator is locked to the rotor by some force. Approximate duration is about a second. Then the  $L_d$ ,  $L_q$  inductances are measured, when the rotor is moved into two static positions and calibrated there by sending a series of short current pulses. If rotor starts to oscillate, it may be required to damp it by adding an extra weight or softly touching by hands (but preventing shift in position).

The measurement lasts about 10 seconds.

**NOTE:** because the full voltage of the power supply source is applied to the motor, it may draw high current (depending on the phase terminal resistance or the motor's winding). If the power supply can not provide enough current, calibration results may become inconsistent. The solution would be reducing voltage or taking power source with higher amperage capabilities. Current limit is not applied for RL calibration!

## B-EMF constant calibration.

Before starting the calibration, it is necessary to check the correctness of the stator resistance parameter and the number of motor poles - otherwise the constant will be calculated incorrectly. It is necessary to rotate the motor by hands to perform the calibration. Do fast enough rotations in both directions. If possible, use a drill/screwdriver to increase the value of the generated EMF and thereby increase the signal-to-noise ratio. It's more important for the motor with small inductance of the windings. When calibrating, peak values are displayed in GUI. When a particular speed of rotation is reached, the constant ceases to grow - this is the required value.

## Temperature Model Setup

It is necessary to connect a temperature sensor installed in the motor close to the windings for reliable temperature protection and compensation of the temperature resistance of the windings. But in its absence, you can use the heating-and-cooling model. Two parameters are defined, which determine the *heating rate* depending on the current consumed, and the *cooling rate* depending on the temperature of the environment.

Preparation for setup:

1. Temporarily set the "Current loop bandwidth" = 0, "Current loop I-term" = 0
2. POWER must be set no more than 150 so that there is a margin for compensation
3. R and L values are calibrated



4. Compensation for motor resistance over the temperature in a couple with the temperature model is enabled

User actions:

1. The motor must be turned on, engaged for its task and loaded, that is, it must consume a constant current close to that specified by the POWER parameter.
2. Now we can observe in the "Monitoring" tab the error value D\_C\_PID\_ERR, and its change in time as the motor warms up. We select the heating coefficient of the temperature model in such a way to minimize the error. The cooling coefficient is set to 0 temporarily. If necessary, repeat the heating sequence by rebooting the system (to reset the model) and cooling down the motor in advance.
3. To adjust the cooling coefficient, turn off the engine and wait for the motor to cool. Then briefly turn on the motor and check the error value. With the correct coefficient, the calculated value of the winding resistance will match with the real one, and the error will be close to zero.
4. Repeat these steps until you get as low error value as possible. The high precision is not important there because the model is very simplified, and a 20-30% tolerance is acceptable for our application.

## Initialization and modes of work

If the battery isn't connected, but the main board has 5V power supply - e.g., a USB cable is connected, and the driver's power is off, - the board will be in the search mode. When the drivers get the power, they will be initialized by the main board. If there is a power failure, an error will occur.

There are three main modes of the driver:

- The asynchronous operation, setting the specified electrical angle. The system works in this mode if the parameter "Encoder El. field offset" (electrical angle offset) is not specified. **ATTENTION!** In this mode, the motor is continuously supplied with a full current, which can cause overheating of the motor and/or driver. Watch the temperature and adjust the max. power parameter for safe operation. This mode is intended only for doing the encoder calibration.
- With the encoder enabled and calibrated, but current loop disabled (parameters "Motor inductance Ld" and "Current loop bandwidth", "Current loop I-term gain" are set to 0). This mode is similar to the mode of a conventional SimpleBGC32 controller with encoders.
- A full-featured FOC with current feedback when all motor parameters are set.

## Voltage drop compensation

- The rated voltage of a fully charged battery, defined in the tab "Service -> Battery monitoring -> Compensate voltage drop", is taken into account. The actual measured voltage is compared with it, and the difference is compensated.
- With a supply voltage in the range of 75% - 100%, the motor torque remains unchanged - with the corresponding current consumption by the driver, but only if there is a margin for

increasing POWER. Suppose if POWER = 204 is specified, with a voltage drop we will increase this parameter, and at a voltage level of 75% of the nominal we will have the final compensated POWER = 255, which with the lower voltage provides the same torque. As a result, the response of the system remains constant.

- When the voltage drops below 75% of the specified level, the output torque drops in proportion to the voltage.
- If the real voltage is greater than the specified rated voltage, compensation is not performed in the opposite direction.

**ATTENTION!** For the correct operation of voltage drop compensation, it is essential to specify the rated supply voltage and motor resistance correctly!

## CAN\_DRV usage scenarios

The main controller can use CAN-drivers to stabilize a camera (one driver per axis) with the feedback from the main IMU sensor, or as external servo-controllers with the feedback from the encoder, in closed loop over torque, speed or position. A support for additional motors in servo mode will be added to the firmware in the future. Some examples when additional motors may be used together with the gimbal controller::

- The motor driver of the 4th axis, which intended to maintain the three main axes in the "normal" (i.e., orthogonal) state, extending the range of working angles of the system
- The focus, zoom and iris drives based on brushless motors (silent, fast and precise), linked to the SimpleBGC32 controller and controlled remotely or by scripts.
- Motion controller - pre-programmed camera rotations and simultaneous movement in 3D space built completely with the brushless motors..
- Additional motors with reaction masses to increase the precision of stabilization for long-range cameras.
- .. and so on. Your ideas are welcome!

## Protection systems

CAN\_Drv module has several types of protection:

- Fast hardware under-voltage protection is implemented using an integrated comparator. When the voltage level is less than 6V, the protection is triggered, and the PWM is switched off. The approximate hysteresis value is 100 mV, i.e., after reaching ~ 6.1V, PWM will be turned on.
- Slow software under-voltage protection - the threshold specified by the parameter "Service → Battery Monitoring → Low voltage threshold,V (stop motors)", reduced by 5%. When the voltage drops below threshold, motors are turned OFF; when the level is restored back (with some hysteresis applied), the error will be reset and the driver resumes operation.
- Fast hardware over-current protection is implemented using an integrated comparator. It protects the board from short circuit. The level is specially overestimated since even a short

surge current will cause triggering of protection. It is determined by the max. impulse current characteristic of the transistor from the datasheet. To resume normal operation, system should be restarted.

- Slow software over-current protection. It triggers when the average current in the motor exceeds the specified threshold. To resume normal operation, system should be restarted.
- Protection by the signal on the “Emergency stop” port. The input is pulled up internally. The switch is normally open, and to trigger a protection it connects to the ground. To resume normal operation, system should be restarted.
- Protection against loss of communication. If the board does not receive a synchronization message within ~20ms, the motor torque is set to zero.
- Protection against overheating of the driver board. The temperature is read by the built-in temperature sensor. If it exceeds the specified limit, system stops working. To resume normal operation, system should be restarted.
- Protection against motor overheating. It is possible to use a temperature sensor DS18B20 (1-GND, 2-DQ, 3-Vdd) to monitor the temperature of the motor. The common sensor lead (DQ) must be connected to the corresponding terminal of the board. The Vdd is connected to + 5V, or connected to pin 2 (DQ) (parasitic power - only two wires are used). If the motor temperature sensor is not connected, a simple model may be used. This model estimates the temperature of a motor taking into account consumed power and dissipated power and must be configured to provide adequate results.
- FOC duration. Occurs when the period of the FOC cycle is abnormal.
- Speed of rotation exceeds threshold. System will be interrupted with the “Emergency stop” error. To resume normal operation, system should be restarted.

## Modes of LED operation.

1. **Fast blinking orange** - bootloader mode is active, starts when power is applied. During auto-search, it can briefly light up green.
2. **Continuously red** when the power is turned on - it is an internal error of the bootloader firmware. You need to update the firmware with an external adapter. Contact manufacturer of the module for details.
3. **Slowly blinking green** - the board is in the operating mode, but not initialized.
4. **Constantly green** - the board is in the normal operating mode and initialized.
5. **Blinking red** - an error has occurred. Depending on the type of blinking, you can determine the reason of error (also it can be found in the GUI). Frequent blinking means that there are several errors. Denoting "•" for a short blink, "-" for a long blink:

Overcurrent protection:	A:	• -
Undervoltage protection:	B:	- • • •
Emergency pin:	C:	- • - •
FOC duration:	D:	- • •
MCU overheat:	E:	•

DRV overheat:	F:	••-•
Overcurrent software	G:	--•
CAN Bus off:	H:	••••
Encoder not found:	I:	••