

## Steuerung

Der canBLC wertet zwei CAN Identifier aus. Den Identifier 0x001 enthält die Stellwerte und der Identifier 0x100 wird zur Konfiguration verwendet. Alle anderen Identifier werden von der Peripherie ignoriert. Das Tastverhältnis der PWM des Reglers lässt sich von 0-255 linear einstellen. Dies entspricht einem Tastverhältnis von 0 - 95%. Die verwendete Methode zur Erkennung des Nulldurchgangs der backEMF lässt prinzipbedingt keine 100% Tastverhältnis zu. Bis zu 99% wären aber erreichbar.

## Konfiguration

Über einen Frame mit dem Identifier 0x100 (*Config*) lassen sich verschiedene Parameter des Reglers einstellen. Um mehrere Regler an einem CAN-Bus zu betreiben, muss zunächst jedem Regler eine eindeutige Adresse zugewiesen werden. Für diese Einstellung darf nur jeweils ein Regler am Bus angeschlossen sein. Alle anderen Parameter können gemeinsam für alle Regler am Bus eingestellt werden.

ID	Name	0	1	2	3	4	5	6	7
0x001	setPWM	BLDC1	BLDC2	BLDC3	BLDC4	BLDC5	BLDC6	BLDC7	BLDC8
0x100	Config	bldc ADR	timing	minpwm	maxpwm	misc	n.a.	n.a.	n.a.
0x10#	feedback							n.a.	n.a.

Tabelle 1: CAN-Frame Übersicht

Der Aufbau des *Config*-Frames ist in Tabelle 1 zu sehen. Über das Timing (*timing*) lässt sich der Kommutierungszeitpunkt einstellen. Das *timing* Byte entspricht einem ganzzahligen Teiler der standardmäßig eingestellten 30°. Der Winkel  $\alpha$  zwischen dem Nulldurchgang der backEMF und dem Schalten der Kommutierung ist:  $\alpha = \frac{60^\circ}{\text{timing}+1}$ . Das minimale Tastverhältnis (*minpwm*) der PWM bestimmt die Schwelle, ab der der Motor gestartet bzw. gestoppt wird. Mit dem maximalen Tastverhältnis (*maxpwm*) lässt sich dieses, und damit der resultierende Motorstrom, nach oben hin begrenzen. Die

Skalierung für *minpwm* und *maxpwm* entspricht der der *setpwm*-Werte. Jeweils über ein Bit des *misc*-Byte (Tabelle 2) lässt sich eine von zwei Startmethoden (*startup*) auswählen und das Rückmelden von Statusinformationen (*feedback*) aktivieren oder deaktivieren.

Alle Einstellungen sind zunächst nicht permanent. Sobald die Spannungsversorgung unterbrochen wird, geht jede Änderung verloren. Um sie dauerhaft zu speichern, können sie durch Setzen des *save\_config*-Bits des *misc*-Bytes ins Flash des Mikrocontrollers gespeichert werden (Tabelle 2).

misc								
Bit	7	6	5	4	3	2	1	0
Name	save_config	set_addr	set_timing	set_timeout	n.a.	n.a.	startup	feedback
0	nicht schreiben	keine Änderung	keine Änderung	keine Änderung			schnell	disable
1	im Flash speichern	Übernehmen	Übernehmen	Übernehmen			Rampe	enable

Tabelle 2: CAN-Frame Config(0x100): misc(7. Byte)

Um ein versehentliches Ändern der Konfiguration eines Reglers zu verhindern, muss zur Änderung von Parametern zusätzlich ein Bit im *misc*-Byte gesetzt werden. Solange der Motor nicht läuft, lassen sich sämtliche Parameter verändern. Im laufenden Betrieb kann nur das Timing angepasst werden und das Rücksenden von Statusinformationen ein bzw. ausgeschaltet werden.

Der Regler kann wieder in seinen ursprünglichen Zustand zurückgesetzt werden, indem ihm im *Config*-Frame das Bitmuster *0xF007FA17ED*, wie in Tabelle 3 zu sehen, gesendet wird.

ID	Name	Funktion	0	1	2	3	4	5	6	7
0x100	Config	Reset	0xF0	0x07	0xFA	0x17	0xED	0x00	0x00	0x00

Tabelle 3: CAN-Frame Config(0x100): Reset

Der Regler besitzt nach diesem Reset wieder die Adresse 1. Das minimale Tastverhältnis *minpwm* liegt bei 1% und *maxpwm* hat den maximalen Wert von 95%, das Tastverhältnis der PWM wird damit nach oben hin nicht begrenzt. Die Rückmeldung des Reglers ist aktiv.

### Statusmeldung über CAN

Ist im *misc*-Byte das *feedback*-Bit gesetzt, sendet der Regler Statusinformationen zurück. Er verwendet dafür, abhängig von der ihm zugeordneten Adresse, unterschiedliche Identifier. Die höchstwertigen 7 Bit sind dabei immer gleich, während die restlichen 4 seiner Adresse entsprechen.