5 Register Mapping

This chapter gives an overview of the complete register set. Some of the registers bundling a number of single bits are detailed in extra tables. The functional practical application of the settings is detailed in dedicated chapters.

Note
- All registers become reset to 0 upon power up, unless otherwise noted.
- Add 0x80 to the address **Addr** for write accesses!

NOTATION OF HEXADECIMAL AND BINARY NUMBERS	
0x	precedes a hexadecimal number, e.g. 0x04
%	precedes a multi-bit binary number, e.g. %100

NOTATION OF R/W FIELD	
R	Read only
W	Write only
R/W	Read- and writable register
R+C	Clear upon read

OVERVIEW REGISTER MAPPING

REGISTER	**DESCRIPTION**
General Configuration Registers	These registers contain - global configuration - global status flags - interface configuration - and I/O signal configuration
Velocity Dependent Driver Feature Control Register Set	This register set offers registers for - driver current control - setting thresholds for coolStep operation - setting thresholds for different chopper modes - setting thresholds for dcStep operation
Motor Driver Register Set	This register set offers registers for - setting / reading out microstep table and counter - chopper and driver configuration - coolStep and stallGuard2 configuration - dcStep configuration - reading out stallGuard2 values and driver error flags

5.1 General Configuration Registers

GENERAL CONFIGURATION REGISTERS (0X00…0X0F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	
RW	0x00	17	*GCONF*	**Bit**	***GCONF* – Global configuration flags**
				0	*recalibrate* 1: Zero crossing recalibration during driver disable (via ENN or via TOFF setting)
				1	*faststandstill* Timeout for step execution until standstill detection: 1: Short time: 2^18 clocks 0: Normal time: 2^20 clocks
				2	*en_pwm_mode* 1: stealthChop voltage PWM mode enabled (depending on velocity thresholds). Switch from off to on state while in stand-still and at IHOLD= nominal IRUN current, only.
				3	*multistep_filt* 1: Enable step input filtering for stealthChop optimization with external step source (default=1)
				4	*shaft* 1: Inverse motor direction
				5	*diag0_error* 1: Enable DIAG0 active on driver errors: Over temperature (*ot*), short to GND (*s2g*), undervoltage chargepump (*uv_cp*) DIAG0 always shows the reset-status, i.e. is active low during reset condition.
				6	*diag0_otpw* 1: Enable DIAG0 active on driver over temperature prewarning (*otpw*)
				7	*diag0_stall* 1: Enable DIAG0 active on motor stall (set *TCOOLTHRS* before using this feature)
				8	*diag1_stall* 1: Enable DIAG1 active on motor stall (set *TCOOLTHRS* before using this feature)
				9	*diag1_index* 1: Enable DIAG1 active on index position (microstep look up table position 0)
				10	*diag1_onstate* 1: Enable DIAG1 active when chopper is on (for the coil which is in the second half of the fullstep)
				11	*diag1_steps_skipped* 1: Enable output toggle when steps are skipped in dcStep mode (increment of *LOST_STEPS*). Do not enable in conjunction with other DIAG1 options.
				12	*diag0_int_pushpull* 0: DIAG0 is open collector output (active low) 1: Enable DIAG0 push pull output (active high)
				13	*diag1_pushpull* 0: DIAG1 is open collector output (active low) 1: Enable DIAG1 push pull output (active high)

GENERAL CONFIGURATION REGISTERS (0X00…0X0F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	
				14	*small_hysteresis* 0: Hysteresis for step frequency comparison is 1/16 1: Hysteresis for step frequency comparison is 1/32
				15	*stop_enable* 0: Normal operation 1: Emergency stop: ENCA_DCIN stops the sequencer when tied high (no steps become executed by the sequencer, motor goes to standstill state).
				16	*direct_mode* 0: Normal operation 1: Motor coil currents and polarity directly programmed via serial interface: Register *XDIRECT* (0x2D) specifies signed coil A current (bits 8..0) and coil B current (bits 24..16). In this mode, the current is scaled by *IHOLD* setting. Velocity based current regulation of stealthChop is not available in this mode. The automatic stealthChop current regulation will work only for low stepper motor velocities.
				17	*test_mode* 0: Normal operation 1: Enable analog test output on pin DCO. *IHOLD*[1..0] selects the function of DCO: 0…2: T120, DAC, VDDH *Hint:* Not for user, set to 0 for normal operation!
R+ WC	0x01	3	*GSTAT*	**Bit**	***GSTAT* – Global status flags** (Re-Write with '1' bit to clear respective flags)
				0	*reset* 1: Indicates that the IC has been reset. All registers have been cleared to reset values.
				1	*drv_err* 1: Indicates, that the driver has been shut down due to overtemperature or short circuit detection. Read DRV_STATUS for details. The flag can only be cleared when the temperature is below the limit again.
				2	*uv_cp* 1: Indicates an undervoltage on the charge pump. The driver is disabled during undervoltage. This flag is latched for information.
R	0x04	8 + 8	*IOIN*	**Bit**	***INPUT***
					Reads the state of all input pins available
				0	STEP
				1	DIR
				2	DCEN_CFG4
				3	DCIN_CFG5
				4	DRV_ENN
				5	DCO_CFG6
				6	1
				7	unused
				31.. 24	*VERSION*: 0x30=first version of the IC Identical numbers mean full digital compatibility.
W	0x06		*OTP_PROG*	**Bit**	***OTP_PROGRAM* – OTP programming**

GENERAL CONFIGURATION REGISTERS (0x00…0x0F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	
					Write access programs OTP memory (one bit at a time), Read access refreshes read data from OTP after a write
				2..0	*OTPBIT* Selection of OTP bit to be programmed to the selected byte location (n=0..7: programs bit n to a logic 1)
				5..4	*OTPBYTE* Set to 00
				15..8	*OTPMAGIC* Set to *0xbd* to enable programming. A programming time of minimum 10ms per bit is recommended (check by reading *OTP_READ*).
R	0x07		*OTP_READ*	**Bit**	***OTP_READ*** (Access to OTP memory result and update) *See separate table!*
				7..0	*OTP0* byte 0 read data
RW	0x08	5	*FACTORY_ CONF*	4..0	*FCLKTRIM* (*Reset default: OTP*) 0…31: Lowest to highest clock frequency. Check at charge pump output. The frequency span is not guaranteed, but it is tested, that tuning to 12MHz internal clock is possible. The devices come preset to 12MHz clock frequency by OTP programming. *(Reset Default: OTP)*
W	0x09	19	*SHORT_ CONF*	**Bit**	***SHORT_CONF***
				3..0	*S2VS_LEVEL*: Short to VS detector level for lowside FETs. Checks for voltage drop in LS MOSFET and sense resistor. 4 (highest sensitivity) … 15 (lowest sensitivity) *Hint:* Settings from 1 to 3 will trigger during normal operation due to voltage drop on sense resistor. *(Reset Default: OTP 6 or 12)*
				11..8	*S2G_LEVEL*: Short to GND detector level for highside FETs. Checks for voltage drop on high side MOSFET 2 (highest sensitivity) … 15 (lowest sensitivity) *Attention: Settings below 6 not recommended at >52V operation – false detection might result* *(Reset Default: OTP 6 or 12)*
				17..16	*SHORTFILTER*: Spike filtering bandwidth for short detection 0 (lowest, 100ns), 1 (1µs), 2 (2µs) 3 (3µs) *Hint:* A good PCB layout will allow using setting 0. Increase value, if erroneous short detection occurs. *(Reset Default = %01)*
				18	*shortdelay*: Short detection delay 0=750ns: normal, 1=1500ns: high The short detection delay shall cover the bridge switching time. 0 will work for most applications. *(Reset Default = 0)*
W	0x0A	22	*DRV_CONF*	**Bit**	**DRV_CONF**

GENERAL CONFIGURATION REGISTERS (0x00…0x0F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	
				4..0	*BBMTIME:* Break-Before make delay 0=shortest (100ns) … 16 (200ns) … 24=longest (375ns) >24 not recommended, use *BBMCLKS* instead *Hint:* Choose the lowest setting safely covering the switching event in order to avoid bridge cross-conduction. Add roughly 30% of reserve. *(Reset Default = 0)*
				11..8	*BBMCLKS*: 0..15: Digital BBM time in clock cycles (typ. 83ns). The longer setting rules (*BBMTIME* vs. *BBMCLKS*). *(Reset Default: OTP 4 or 2)*
				17..16	*OTSELECT*: Selection of over temperature level for bridge disable, switch on after cool down to 120°C / OTPW level. 00: 150°C 01: 143°C 10: 136°C (not recommended when VSA > 24V) 11: 120°C (not recommended, no hysteresis) *Hint:* Adapt overtemperature threshold as required to protect the MOSFETs or other components on the PCB. *(Reset Default = %00)*
				19..18	*DRVSTRENGTH*: Selection of gate driver current. Adapts the gate driver current to the gate charge of the external MOSFETs. 00: weak 01: weak+TC (medium above OTPW level) 10: medium 11: strong *Hint:* Choose the lowest setting giving slopes <100ns. *(Reset Default = %10)*
				21..20	*FILT_ISENSE*: Filter time constant of sense amplifier to suppress ringing and coupling from second coil operation 00: low – 100ns 01: – 200ns 10: – 300ns 11: high– 400ns *Hint:* Increase setting if motor chopper noise occurs due to cross-coupling of both coils. *(Reset Default = %00)*

GENERAL CONFIGURATION REGISTERS (0X00…0X0F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	
W	0x0B	8	*GLOBAL SCALER*	7..0	Global scaling of Motor current. This value is multiplied to the current scaling in order to adapt a drive to a certain motor type. This value should be chosen before tuning other settings, because it also influences chopper hysteresis. 0: Full Scale (or write 256) 1 … 31: Not allowed for operation 32 … 255: 32/256 … 255/256 of maximum current. *Hint:* Values >128 recommended for best results *(Reset Default = 0)*
R	0x0C	16	*OFFSET_ READ*	15..8	Offset calibration result phase A (signed)
				7..0	Offset calibration result phase B (signed)

5.1.1 *OTP_READ* – OTP configuration memory

The OTP memory holds power up defaults for certain registers. All OTP memory bits are cleared to 0 by default. Programming only can set bits, clearing bits is not possible. Factory tuning of the clock frequency affects *otp0.0* to *otp0.4*. The state of these bits therefore may differ between individual ICs.

0x07: *OTP_READ* – OTP MEMORY MAP			
Bit	**Name**	**Function**	**Comment**
7	*otp0.7*	*otp_TBL*	Reset default for *TBL*: 0: *TBL*=%10 (-3µs) 1: *TBL*=%01 (-2µs)
6	*otp0.6*	*otp_BBM*	Reset default for *DRVCONF.BBMCLKS* 0: *BBMCLKS*=4 1: *BBMCLKS*=2
5	*otp0.5*	*otp_S2_LEVEL*	Reset default for *Short detection Levels*: 0: *S2G_LEVEL = S2VS_LEVEL = 6* 1: *S2G_LEVEL = S2VS_LEVEL = 12*
4	*otp0.4*	*OTP_FCLKTRIM*	Reset default for *FCLKTRIM* 0: lowest frequency setting 31: highest frequency setting *Attention: This value is pre-programmed by factory clock trimming to the default clock frequency of 12MHz and differs between individual ICs! It should not be altered.*
3	*otp0.3*		
2	*otp0.2*		
1	*otp0.1*		
0	*otp0.0*		

5.2 Velocity Dependent Driver Feature Control Register Set

VELOCITY DEPENDENT DRIVER FEATURE CONTROL REGISTER SET (0X10...0X1F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	
W	0x10	5 + 5 + 4	*IHOLD_IRUN*	**Bit**	***IHOLD_IRUN* – Driver current control**
				4..0	*IHOLD* Standstill current (0=1/32...31=32/32) In combination with stealthChop mode, setting *IHOLD*=0 allows to choose freewheeling or coil short circuit for motor stand still.
				12..8	*IRUN* Motor run current (0=1/32...31=32/32) *Hint:* Choose sense resistors in a way, that normal IRUN is 16 to 31 for best microstep performance.
				19..16	*IHOLDDELAY* Controls the number of clock cycles for motor power down after a motion as soon as standstill is detected (*stst*=1) and *TPOWERDOWN* has expired. The smooth transition avoids a motor jerk upon power down. 0: instant power down 1..15: Delay per current reduction step in multiple of 2^18 clocks
W	0x11	8	*TPOWER DOWN*	*TPOWERDOWN* sets the delay time after stand still (*stst*) of the motor to motor current power down. Time range is about 0 to 4 seconds. *Attention: A minimum setting of 2 is required to allow automatic tuning of stealthChop PWM_OFFS_AUTO.* *Reset Default = 10* *0...((2^8)-1)* * 2^18 t_{CLK}	
R	0x12	20	*TSTEP*	Actual measured time between two 1/256 microsteps derived from the step input frequency in units of 1/fCLK. Measured value is (2^20)-1 in case of overflow or stand still. All TSTEP related thresholds use a hysteresis of 1/16 of the compare value to compensate for jitter in the clock or the step frequency. The flag *small_hysteresis* modifies the hysteresis to a smaller value of 1/32. (*Txxx**15/16)-1 or (*Txxx**31/32)-1 is used as a second compare value for each comparison value. This means, that the lower switching velocity equals the calculated setting, but the upper switching velocity is higher as defined by the hysteresis setting. In dcStep mode *TSTEP* will not show the mean velocity of the motor, but the velocities for each microstep, which may not be stable and thus does not represent the real motor velocity in case it runs slower than the target velocity.	
W	0x13	20	*TPWMTHRS*	This is the upper velocity for stealthChop voltage PWM mode. *TSTEP* ≥ *TPWMTHRS* - stealthChop PWM mode is enabled, if configured - dcStep is disabled	

VELOCITY DEPENDENT DRIVER FEATURE CONTROL REGISTER SET (0X10...0X1F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	
W	0x14	20	*TCOOLTHRS*	This is the lower threshold velocity for switching on smart energy coolStep and stallGuard feature. (unsigned) Set this parameter to disable coolStep at low speeds, where it cannot work reliably. The stop on stall function (enable with *sg_stop* when using internal motion controller) and the stall output signal become enabled when exceeding this velocity. In non-dcStep mode, it becomes disabled again once the velocity falls below this threshold. *TCOOLTHRS* ≥ *TSTEP* ≥ *THIGH*: - coolStep is enabled, if configured - stealthChop voltage PWM mode is disabled *TCOOLTHRS* ≥ *TSTEP* - Stop on stall is enabled, if configured - Stall output signal (DIAG0/1) is enabled, if configured	
W	0x15	20	*THIGH*	This velocity setting allows velocity dependent switching into a different chopper mode and fullstepping to maximize torque. (unsigned) The stall detection feature becomes switched off for 2-3 electrical periods whenever passing *THIGH* threshold to compensate for the effect of switching modes. *TSTEP* ≤ *THIGH*: - coolStep is disabled (motor runs with normal current scale) - stealthChop voltage PWM mode is disabled - If *vhighchm* is set, the chopper switches to *chm*=1 with *TFD*=0 (constant off time with slow decay, only). - If *vhighfs* is set, the motor operates in fullstep mode and the stall detection becomes switched over to dcStep stall detection.	
RW	0x2D	9+9	*XDIRECT*	This register is used in direct coil current mode, only (*direct_mode* = 1). It bypasses the internal sequencer. Specifies signed coil A current (bits 8..0) and coil B current (bits 24..16). In this mode, the current is scaled by *IHOLD* setting. Velocity based current regulation of stealthChop is not available in this mode. The automatic stealthChop current regulation will work only for low stepper motor velocities.	2x -255...+255

Microstep velocity time reference t for velocities: TSTEP = f_{CLK} / f_{STEP}

5.2.1 dcStep Miniumum Velocity Register

DCSTEP MINIMUM VELOCITY REGISTER (0X33)				
R/W	**Addr**	**n**	***Register***	**Description */ bit names***
W	0x33	23	*VDCMIN*	Automatic commutation dcStep minimum velocity. Enable dcStep by DCEN pin. In this mode, the actual position is determined by the sensorless motor commutation and becomes fed back to the external motion controller. In case the motor becomes heavily loaded, *VDCMIN* is used as the minimum step velocity. *Hint:* Also set *DCCTRL* parameters in order to operate dcStep. (Only bits 22… 8 are used for value and for comparison)

Time reference t for *VDCMIN*: t = 2^24 / f_{CLK}

5.3 Motor Driver Registers

MICROSTEPPING CONTROL REGISTER SET (0X60…0X6B)					
R/W	**Addr**	**n**	***Register***	**Description *I bit names***	**Range [Unit]**
W	0x60	32	*MSLUT[0]* microstep table entries 0…31	Each bit gives the difference between entry x and entry x+1 when combined with the corresponding *MSLUTSEL W* bits: 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3	32x 0 or 1 *reset default= sine wave table*
W	0x61 … 0x67	7 x 32	*MSLUT[1…7]* microstep table entries 32…255	This is the differential coding for the first quarter of a wave. Start values for *CUR_A* and *CUR_B* are stored for *MSCNT* position 0 in *START_SIN* and *START_SIN90*. *ofs31*, *ofs30*, …, *ofs01*, *ofs00* … *ofs255*, *ofs254*, …, *ofs225*, *ofs224*	7x 32x 0 or 1 *reset default= sine wave table*
W	0x68	32	***MSLUTSEL***	This register defines four segments within each quarter *MSLUT* wave. Four 2 bit entries determine the meaning of a 0 and a 1 bit in the corresponding segment of *MSLUT*. *See separate table!*	*0<X1<X2<X3* *reset default= sine wave table*
W	0x69	8 + 8	*MSLUTSTART*	bit 7… 0: *START_SIN* bit 23… 16: *START_SIN90* *START_SIN* gives the absolute current at microstep table entry 0. *START_SIN90* gives the absolute current for microstep table entry at positions 256. Start values are transferred to the microstep registers *CUR_A* and *CUR_B*, whenever the reference position *MSCNT*=0 is passed.	*START_SIN* *reset default =0* *START_SIN90* *reset default =247*
R	0x6A	10	*MSCNT*	Microstep counter. Indicates actual position in the microstep table for *CUR_A*. *CUR_B* uses an offset of 256 (2 phase motor). *Hint:* Move to a position where *MSCNT* is zero before re-initializing *MSLUTSTART* or *MSLUT* and *MSLUTSEL*.	0…1023
R	0x6B	9 + 9	*MSCURACT*	bit 8… 0: *CUR_A* (signed): Actual microstep current for motor phase A as read from MSLUT (not scaled by current) bit 24… 16: *CUR_B* (signed): Actual microstep current for motor phase B as read from *MSLUT* (not scaled by current)	+/-0…255

DRIVER REGISTER SET (0X6C…0X7F)					
R/W	**Addr**	**n**	***Register***	**Description */ bit names***	**Range [Unit]**
RW	0x6C	32	***CHOPCONF***	chopper and driver configuration *See separate table!*	*reset default= 0x10410150*
W	0x6D	25	***COOLCONF***	coolStep smart current control register and stallGuard2 configuration *See separate table!*	
W	0x6E	24	*DCCTRL*	dcStep (*DC*) automatic commutation configuration register (enable via pin DCEN or via *VDCMIN*): bit 9… 0: *DC_TIME*: Upper PWM on time limit for commutation (*DC_TIME* * 1/f_{CLK}). Set slightly above effective blank time *TBL*. bit 23… 16: *DC_SG*: Max. PWM on time for step loss detection using dcStep stallGuard2 in dcStep mode. (*DC_SG* * 16/f_{CLK}) Set slightly higher than *DC_TIME*/16 0=disable *Hint:* Using a higher microstep resolution or interpolated operation, dcStep delivers a better stallGuard signal. DC_SG is also available above VHIGH if vhighfs is activated. For best result also set vhighchm.	
R	0x6F	32	***DRV_ STATUS***	stallGuard2 value and driver error flags *See separate table!*	
W	0x70	22	***PWMCONF***	Voltage PWM mode chopper configuration *See separate table!*	*reset default= 0xC40C001E*
R	0x71	9+8	*PWM_SCALE*	Results of stealthChop amplitude regulator. These values can be used to monitor automatic PWM amplitude scaling (255=max. voltage).	
				bit 7… 0: *PWM_SCALE_SUM*: Actual PWM duty cycle. This value is used for scaling the values *CUR_A* and *CUR_B* read from the sine wave table.	0…255
				bit 24… 16: *PWM_SCALE_AUTO*: 9 Bit signed offset added to the calculated PWM duty cycle. This is the result of the automatic amplitude regulation based on current measurement.	signed -255…+255
R	0x72	8+8	*PWM_AUTO*	These automatically generated values can be read out in order to determine a default */* power up setting for *PWM_GRAD* and *PWM_OFS*.	
				bit 7… 0: *PWM_OFS_AUTO*: Automatically determined offset value	0…255

DRIVER REGISTER SET (0X6C…0X7F)						
R/W	**Addr**	**n**	***Register***	**Description */ bit names***		**Range [Unit]**
				bit 23… 16	*PWM_GRAD_AUTO*: Automatically determined gradient value	0…255
R	0x73	20	*LOST_STEPS*	Number of input steps skipped due to higher load in dcStep operation, if step input does not stop when DC_OUT is low. This counter wraps around after 2^20 steps. Counts up or down depending on direction. Only with SDMODE=1.		

MICROSTEP TABLE CALCULATION FOR A SINE WAVE EQUIVALENT TO THE POWER ON DEFAULT

$$round\left(248 * sin\left(2 * PI * \frac{i}{1024} + \frac{PI}{1024}\right)\right) - 1$$

- *i*:[0… 255] is the table index
- The amplitude of the wave is 248. The resulting maximum positive value is 247 and the maximum negative value is -248.
- The round function rounds values from 0.5 to 1.4999 to 1

5.3.1 *MSLUTSEL* – Look up Table Segmentation Definition

0x68: *MSLUTSEL* – LOOK UP TABLE SEGMENTATION DEFINITION			
Bit	**Name**	**Function**	**Comment**
31	*X3*	LUT segment 3 start	The sine wave look up table can be divided into up to four segments using an individual step width control entry *Wx*. The segment borders are selected by *X1*, *X2* and *X3*. Segment 0 goes from 0 to *X1*-1. Segment 1 goes from *X1* to *X2*-1. Segment 2 goes from *X2* to *X3*-1. Segment 3 goes from *X3* to 255. For defined response the values shall satisfy: 0<*X1*<*X2*<*X3*
30			
29			
28			
27			
26			
25			
24			
23	*X2*	LUT segment 2 start	
22			
21			
20			
19			
18			
17			
16			
15	*X1*	LUT segment 1 start	
14			
13			
12			
11			
10			
9			
8			
7	*W3*	LUT width select from *ofs(X3)* to *ofs255*	Width control bit coding *W0*…*W3*: %00: MSLUT entry 0, 1 select: -1, +0 %01: MSLUT entry 0, 1 select: +0, +1 %10: MSLUT entry 0, 1 select: +1, +2 %11: MSLUT entry 0, 1 select: +2, +3
6			
5	*W2*	LUT width select from *ofs(X2)* to *ofs(X3-1)*	
4			
3	*W1*	LUT width select from *ofs(X1)* to *ofs(X2-1)*	
2			
1	*W0*	LUT width select from *ofs00* to *ofs(X1-1)*	
0			

5.3.2 *CHOPCONF* – Chopper Configuration

0x6C: *CHOPCONF* – CHOPPER CONFIGURATION			
Bit	**Name**	**Function**	**Comment**
31	*diss2vs*	short to supply protection disable	0: Short to VS protection is on 1: Short to VS protection is disabled
30	*diss2g*	short to GND protection disable	0: Short to GND protection is on 1: Short to GND protection is disabled
29	*dedge*	enable double edge step pulses	1: Enable step impulse at each step edge to reduce step frequency requirement.
28	*intpol*	interpolation to 256 microsteps	1: The actual microstep resolution (*MRES*) becomes extrapolated to 256 microsteps for smoothest motor operation (useful for STEP/DIR operation, only)
27	*mres3*	*MRES* micro step resolution	%0000: Native 256 microstep setting. Normally use this setting with the internal motion controller.
26	*mres2*		
25	*mres1*		
24	*mres0*		%0001 … %1000: 128, 64, 32, 16, 8, 4, 2, FULLSTEP Reduced microstep resolution esp. for STEP/DIR operation. The resolution gives the number of microstep entries per sine quarter wave. The driver automatically uses microstep positions which result in a symmetrical wave, when choosing a lower microstep resolution. step width=2^*MRES* [microsteps]
23	*tpfd3*	*TPFD* passive fast decay time	*TPFD* allows dampening of motor mid-range resonances. Passive fast decay time setting controls duration of the fast decay phase inserted after bridge polarity change N_{CLK}= 128**TPFD* %0000: Disable %0001 … %1111: 1 … 15
22	*tpfd2*		
21	*tpdf1*		
20	*tpfd0*		
19	*vhighchm*	high velocity chopper mode	This bit enables switching to *chm*=1 and *fd*=0, when *VHIGH* is exceeded. This way, a higher velocity can be achieved. Can be combined with *vhighfs*=1. If set, the *TOFF* setting automatically becomes doubled during high velocity operation in order to avoid doubling of the chopper frequency.
18	*vhighfs*	high velocity fullstep selection	This bit enables switching to fullstep, when *VHIGH* is exceeded. Switching takes place only at 45° position. The fullstep target current uses the current value from the microstep table at the 45° position.
17	-	reserved	reserved, set to 0
16	*tbl1*	*TBL* blank time select	%00 … %11: Set comparator blank time to 16, 24, 36 or 54 clocks *Hint*: %01 or %10 is recommended for most applications
15	*tbl0*		
14	*chm*	chopper mode	0: Standard mode (spreadCycle) 1: Constant off time with fast decay time. Fast decay time is also terminated when the negative nominal current is reached. Fast decay is after on time.
13	-	reserved	Reserved, set to 0
12	*disfdcc*	fast decay mode	*chm*=1: *disfdcc*=1 disables current comparator usage for termination of the fast decay cycle
11	*fd3*	*TFD [3]*	*chm*=1: MSB of fast decay time setting *TFD*

0x6C: *CHOPCONF* – CHOPPER CONFIGURATION				
Bit	**Name**	**Function**	**Comment**	
10	*hend3*	*HEND* hysteresis low value *OFFSET* sine wave offset	*chm*=0	%0000 … %1111: Hysteresis is -3, -2, -1, 0, 1, …, 12 (1/512 of this setting adds to current setting) This is the hysteresis value which becomes used for the hysteresis chopper.
9	*hend2*			
8	*hend1*			
7	*hend0*		*chm*=1	%0000 … %1111: Offset is -3, -2, -1, 0, 1, …, 12 This is the sine wave offset and 1/512 of the value becomes added to the absolute value of each sine wave entry.
6	*hstrt2*	*HSTRT* hysteresis start value added to *HEND*	*chm*=0	%000 … %111: Add 1, 2, …, 8 to hysteresis low value *HEND* (1/512 of this setting adds to current setting) *Attention: Effective HEND+HSTRT ≤ 16.* *Hint:* Hysteresis decrement is done each 16 clocks
5	*hstrt1*			
4	*hstrt0*			
		TFD [2..0] fast decay time setting	*chm*=1	Fast decay time setting (MSB: *fd3*): %0000 … %1111: Fast decay time setting *TFD* with N_{CLK}= 32**TFD* (%0000: slow decay only)
3	*toff3*	*TOFF* off time and driver enable	Off time setting controls duration of slow decay phase N_{CLK}= 12 + 32**TOFF* %0000: Driver disable, all bridges off %0001: 1 – use only with *TBL* ≥ 2 %0010 … %1111: 2 … 15	
2	*toff2*			
1	*toff1*			
0	*toff0*			

5.3.3 *COOLCONF* – Smart Energy Control coolStep and stallGuard2

0x6D: *COOLCONF* – SMART ENERGY CONTROL COOLSTEP AND STALLGUARD2			
Bit	**Name**	**Function**	**Comment**
…	-	reserved	set to 0
24	*sfilt*	stallGuard2 filter enable	0 Standard mode, high time resolution for stallGuard2 1 Filtered mode, stallGuard2 signal updated for each four fullsteps (resp. six fullsteps for 3 phase motor) only to compensate for motor pole tolerances
23	-	reserved	set to 0
22	*sgt6*	stallGuard2 threshold value	This signed value controls stallGuard2 level for stall output and sets the optimum measurement range for readout. A lower value gives a higher sensitivity. Zero is the starting value working with most motors. -64 to +63: A higher value makes stallGuard2 less sensitive and requires more torque to indicate a stall.
21	*sgt5*		
20	*sgt4*		
19	*sgt3*		
18	*sgt2*		
17	*sgt1*		
16	*sgt0*		
15	*seimin*	minimum current for smart current control	0: 1/2 of current setting (*IRUN*) 1: 1/4 of current setting (*IRUN*)
14	*sedn1*	current down step speed	%00: For each 32 stallGuard2 values decrease by one %01: For each 8 stallGuard2 values decrease by one %10: For each 2 stallGuard2 values decrease by one %11: For each stallGuard2 value decrease by one
13	*sedn0*		
12	-	reserved	set to 0
11	*semax3*	stallGuard2 hysteresis value for smart current control	If the stallGuard2 result is equal to or above (*SEMIN*+*SEMAX*+1)*32, the motor current becomes decreased to save energy. %0000 … %1111: 0 … 15
10	*semax2*		
9	*semax1*		
8	*semax0*		
7	-	reserved	set to 0
6	*seup1*	current up step width	Current increment steps per measured stallGuard2 value %00 … %11: 1, 2, 4, 8
5	*seup0*		
4	-	reserved	set to 0
3	*semin3*	minimum stallGuard2 value for smart current control and smart current enable	If the stallGuard2 result falls below *SEMIN**32, the motor current becomes increased to reduce motor load angle. %0000: smart current control coolStep off %0001 … %1111: 1 … 15
2	*semin2*		
1	*semin1*		
0	*semin0*		

5.3.4 *PWMCONF* – Voltage PWM Mode stealthChop

0x70: *PWMCONF* – VOLTAGE MODE PWM STEALTHCHOP				
Bit	**Name**	**Function**		**Comment**
31 30 29 28	*PWM_LIM*	PWM automatic scale amplitude limit when switching on		Limit for *PWM_SCALE_AUTO* when switching back from spreadCycle to stealthChop. This value defines the upper limit for bits 7 to 4 of the automatic current control when switching back. It can be set to reduce the current jerk during mode change back to stealthChop. It does not limit *PWM_GRAD* or *PWM_GRAD_AUTO* offset. (Default = 12)
27 26 25 24	*PWM_REG*	Regulation loop gradient		User defined maximum PWM amplitude change per half wave when using *pwm_autoscale*=1. (1…15): 1: 0.5 increments (slowest regulation) 2: 1 increment 3: 1.5 increments 4: 2 increments *(Reset default)*) … 8: 4 increments … 15: 7.5 increments (fastest regulation)
23	-	reserved		set to 0
22	-	reserved		set to 0
21 20	*freewheel1* *freewheel0*	Allows different standstill modes		Stand still option when motor current setting is zero (*I_HOLD*=0). %00: Normal operation %01: Freewheeling %10: Coil shorted using LS drivers %11: Coil shorted using HS drivers
19	*pwm_ autograd*	PWM automatic gradient adaptation	0	Fixed value for *PWM_GRAD* (*PWM_GRAD_AUTO* = *PWM_GRAD*)
			1	Automatic tuning (only with *pwm_autoscale*=1) (*Reset default*) *PWM_GRAD_AUTO* is initialized with *PWM_GRAD* while *pwm_autograd*=0 and becomes optimized automatically during motion. Preconditions 1. *PWM_OFS_AUTO* has been automatically initialized. This requires standstill at IRUN for >130ms in order to a) detect standstill b) wait > 128 chopper cycles at *IRUN* and c) regulate *PWM_OFS_AUTO* so that -1 < *PWM_SCALE_AUTO* < 1 2. Motor running and 1.5 * *PWM_OFS_AUTO* < *PWM_SCALE_SUM* < 4* *PWM_OFS_AUTO* and *PWM_SCALE_SUM* < 255. Time required for tuning *PWM_GRAD_AUTO* About 8 fullsteps per change of +/-1. Also enables use of reduced chopper frequency for tuning *PWM_OFS_AUTO*.
18	*pwm_ autoscale*	PWM automatic amplitude scaling	0	User defined feed forward PWM amplitude. The current settings *IRUN* and *IHOLD* have no influence! The resulting PWM amplitude (limited to 0…255) is: *PWM_OFS * ((CS_ACTUAL+1) / 32)* *+ PWM_GRAD * 256 / TSTEP*
			1	Enable automatic current control *(Reset default)*

0x70: *PWMCONF* – VOLTAGE MODE PWM STEALTHCHOP			
Bit	**Name**	**Function**	**Comment**
17 16	*pwm_freq1* *pwm_freq0*	PWM frequency selection	%00: f_{PWM}=2/1024 f_{CLK} *(Reset default)* %01: f_{PWM}=2/683 f_{CLK} %10: f_{PWM}=2/512 f_{CLK} %11: f_{PWM}=2/410 f_{CLK}
15 14 13 12 11 10 9 8	*PWM_ GRAD*	User defined amplitude gradient	Velocity dependent gradient for PWM amplitude: *PWM_GRAD* * 256 / *TSTEP* This value is added to *PWM_OFS* to compensate for the velocity-dependent motor back-EMF. Use *PWM_GRAD* as initial value for automatic scaling to speed up the automatic tuning process. To do this, set *PWM_GRAD* to the determined, application specific value, with *pwm_autoscale*=0. Only afterwards, set *pwm_autoscale*=1. Enable stealthChop when finished. *Hint:* After initial tuning, the required initial value can be read out from *PWM_GRAD_AUTO*.
7 6 5 4 3 2 1 0	*PWM_ OFS*	User defined amplitude (offset)	User defined PWM amplitude offset (0-255) related to full motor current (*CS_ACTUAL*=31) in stand still. (*Reset default=30*) Use *PWM_OFS* as initial value for automatic scaling to speed up the automatic tuning process. To do this, set *PWM_OFS* to the determined, application specific value, with *pwm_autoscale*=0. Only afterwards, set *pwm_autoscale*=1. Enable stealthChop when finished. *PWM_OFS* = 0 will disable scaling down motor current below a motor specific lower measurement threshold. This setting should only be used under certain conditions, i.e. when the power supply voltage can vary up and down by a factor of two or more. It prevents the motor going out of regulation, but it also prevents power down below the regulation limit. *PWM_OFS* > 0 allows automatic scaling to low PWM duty cycles even below the lower regulation threshold. This allows low (standstill) current settings based on the actual (hold) current scale (register *IHOLD_IRUN*).

5.3.5 *DRV_STATUS* – stallGuard2 Value and Driver Error Flags

0x6F: DRV_STATUS – STALLGUARD2 VALUE AND DRIVER ERROR FLAGS			
Bit	**Name**	**Function**	**Comment**
31	*stst*	standstill indicator	This flag indicates motor stand still in each operation mode. This occurs 2^20 clocks after the last step pulse.
30	*olb*	open load indicator phase B	1: Open load detected on phase A or B. *Hint:* This is just an informative flag. The driver takes no action upon it. False detection may occur in fast motion and standstill. Check during slow motion, only.
29	*ola*	open load indicator phase A	
28	*s2gb*	short to ground indicator phase B	1: Short to GND detected on phase A or B. The driver becomes disabled. The flags stay active, until the driver is disabled by software (*TOFF*=0) or by the ENN input.
27	*s2ga*	short to ground indicator phase A	
26	*otpw*	overtemperature pre-warning flag	1: Overtemperature pre-warning threshold is exceeded. The overtemperature pre-warning flag is common for both bridges.
25	*ot*	overtemperature flag	1: Overtemperature limit has been reached. Drivers become disabled until *otpw* is also cleared due to cooling down of the IC. The overtemperature flag is common for both bridges.
24	*stallGuard*	stallGuard2 status	1: Motor stall detected (*SG_RESULT*=0) or dcStep stall in dcStep mode.
23	-	reserved	Ignore these bits
22			
21			
20	*CS ACTUAL*	actual motor current / smart energy current	Actual current control scaling, for monitoring smart energy current scaling controlled via settings in register *COOLCONF*, or for monitoring the function of the automatic current scaling.
19			
18			
17			
16			
15	*fsactive*	full step active indicator	1: Indicates that the driver has switched to fullstep as defined by chopper mode settings and velocity thresholds.
14	*stealth*	stealthChop indicator	1: Driver operates in stealthChop mode
13	*s2vsb*	short to supply indicator phase B	1: Short to supply detected on phase A or B. The driver becomes disabled. The flags stay active, until the driver is disabled by software (*TOFF*=0) or by the ENN input. Sense resistor voltage drop is included in the measurement!
12	*s2vsa*	short to supply indicator phase A	
11	-	reserved	Ignore this bit
10	-	reserved	Ignore this bit
9	*SG_ RESULT*	stallGuard2 result respectively PWM on time for coil A in stand still for motor temperature detection	Mechanical load measurement: The stallGuard2 result gives a means to measure mechanical motor load. A higher value means lower mechanical load. A value of 0 signals highest load. With optimum *SGT* setting, this is an indicator for a motor stall. The stall detection compares *SG_RESULT* to 0 in order to detect a stall. *SG_RESULT* is used as a base for coolStep operation, by comparing it to a programmable upper and a lower limit. It is not applicable in stealthChop mode. stallGuard2 works best with microstep operation or dcStep. Temperature measurement: In standstill, no stallGuard2 result can be obtained. *SG_RESULT* shows the chopper on-time for motor coil A instead. Move the motor to a determined microstep position at a certain current setting to get a rough estimation of motor temperature by a reading the chopper on-time. As the motor heats up, its coil resistance rises and the chopper on-time increases.
8			
7			
6			
5			
4			
3			
2			
1			
0			