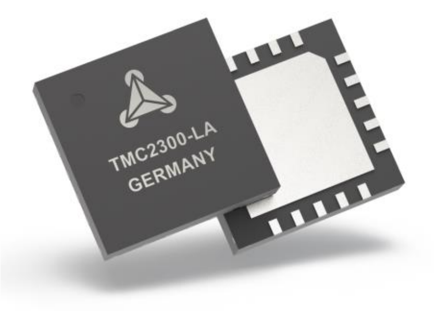
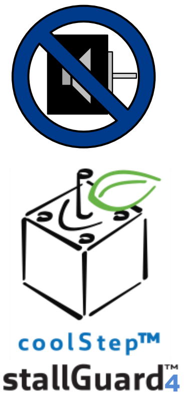
TMC2300 Datasheet

Low Voltage Driver for Two-Phase Stepper Motors up to 1.2A RMS - StealthChop™ for Quiet Movement - UART Interface Option. With StallGuard Sensorless Homing and CoolStep Energy Saving.





APPLICATIONS

- IOT & Handheld devices
- Battery operated equipment
- Printers, POS
- Miniature 3D Printers
- Toys
- Office and home automation
- CCTV, Security
- HVAC
- Mobile medical devices

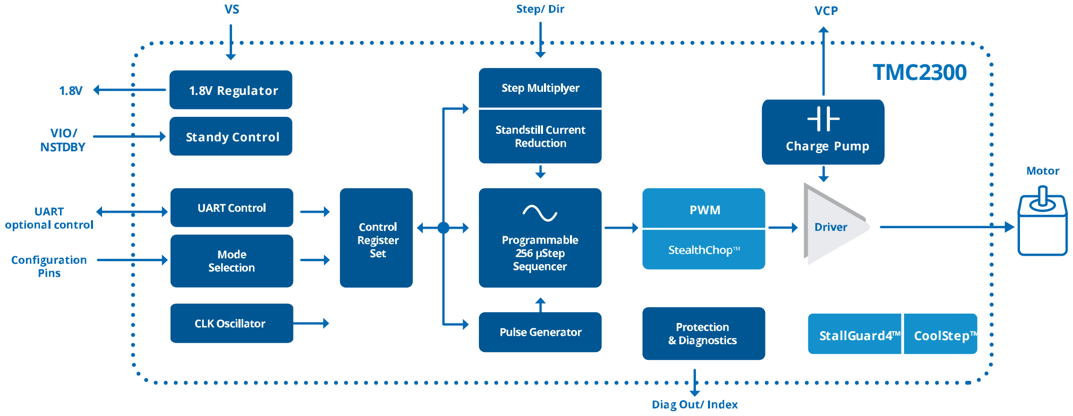
FEATURES AND BENEFITS

Voltage Range 2V (1.8V) … 11V DC
Battery Operation min. 2 AA / NiMh cells, or 1-2 Li-Ion cells
2-phase Stepper Motors up to 1.2A RMS, 2A peak
Standby <50nA typ. current draw
STEP/DIR Interface up to 256 microsteps for stepper motor
Smooth Running 256 microsteps by **microPlyer™** interpolation
StealthChop2™ silent motor operation
Low RDSon LS 170mΩ & HS 170mΩ (typ.)
Automatic Standby current reduction
Stall Detection StallGuard4 in StealthChop mode
CoolStep load dependent energy saving up to 90%
Single Wire UART for advanced configuration options
Integrated Pulse Generator for standalone motion
Full Protection & Diagnostics
Tiny of QFN 3*3 with 20 pins

DESCRIPTION

Working from a single Li-Ion cell or dual AA batteries the TMC2300 is optimally suited for battery operated equipment. TRINAMICs sophisticated StealthChop2 chopper ensures noiseless stepper operation, maximum efficiency and best torque. Its fast current regulation allows for highly dynamic motion. Integrated power-MOSFETs with internal charge-pump for best-in-class RDSon even at low supply voltage, handle motor current up to 1.2A RMS. Together with a tiny standby current, this guarantees a long battery life. Protection and diagnostic features support robust and reliable operation. A simple to use UART interface allows configuration. This advanced driver ensures noiseless and most precise operation for cost-effective and highly competitive solutions.

BLOCK DIAGRAM

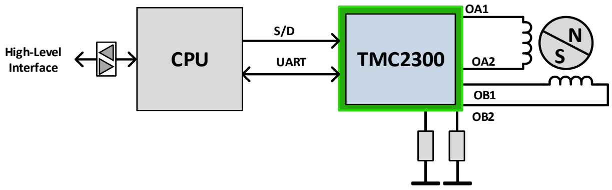


TRINAMIC
MOTION CONTROL

APPLICATION EXAMPLES: SIMPLE SOLUTIONS – HIGHLY EFFECTIVE

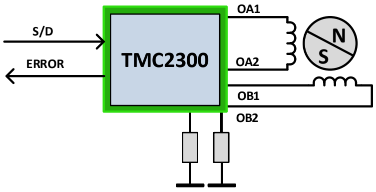
The TMC2300 scores with power density, integrated power MOSFETs, smooth and quiet operation, and a congenial simplicity. The TMC2300 covers a wide spectrum of applications from battery systems to embedded applications with up to 1.2A motor current per coil. TRINAMICs unique chopper mode StealthChop2 optimizes drive performance. StealthChop reduces motor noise to the point of silence at low velocities. Standby current reduction, as well as CoolStep keeps costs for battery consumption and cooling down. Extensive support enables rapid design cycles and fast time-to-market with highly competitive products.

UART Interface for Full Diagnostics and Control

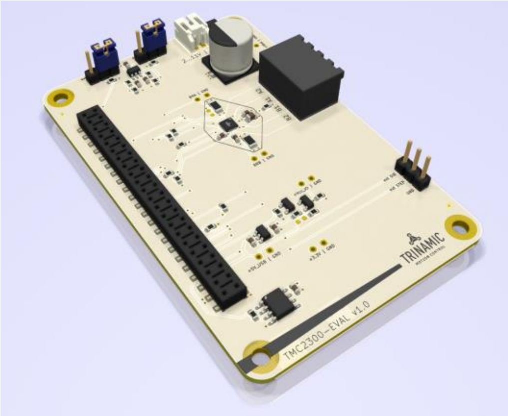


A CPU operates the driver via step and direction signals. It accesses diagnostic information and configures the TMC2300 via the UART interface. The CPU manages motion control and the TMC2300 drives the motor and smoothens and optimizes drive performance.

Standalone Step/Dir Stepper Driver



No control interface is required. The driver operates the motor based on a step & direction signal and pin-selection of the microstep resolution. In this mode, advanced stall detection and diagnostic features are not available.



The TMC2300-EVAL is part of TRINAMICs universal evaluation board system which provides a convenient handling of the hardware as well as a user-friendly software tool for evaluation. The TMC2300 evaluation board system consists of three parts: STARTRAMPE (base board), ESELSBRÜCKE (connector board with test points), and TMC2300-EVAL.

Order Codes

Order code	PN	Description	Size [mm^2]
TMC2300-LA	00-0182	Low voltage stepper motor driver with CoolStep	3 x 3
TMC2300-LA-T	00-0182-T	-T denotes tape on reel packing of devices	
TMC2300-BOB	40-0181	Break out board with TMC2300	25 x 25
TMC2300-EVAL	40-0175	Evaluation board for TMC2300 stepper motor driver	85 x 55
ESELSBRÜCKE	40-0098	Connector board fitting to Landungsbrücke	61 x 38
LANDUNGSBRÜCKE	40-0167	Baseboard for TMC2300-EVAL and further evaluation boards	85 x 55

Table of Contents

1 Principles of Operation

The TMC2300 low voltage motor driver is intended for battery-operated, space- and standby-power-critical driver applications. Its silent drive technology StealthChop enables non-bugging motion control for portable, home and office applications. A highly efficient power stage, boosted by an internal charge pump for best in-class RDSon resistance, provides high motor current from a tiny package even at low supply voltages. With this, dual AA batteries can be drained down to typically 2.0V (voltage must not drop below 1.8V, provide sufficient supply buffer capacitors).

The TMC2300 requires just a few control pins on its tiny package. It allows selection of the most important setting: the desired microstep resolution. A choice of 8, 16, 32 or 64 microsteps, or from fullstep up to 1/256 step adapts the driver to the desired motion precision.

Even at low microstepping rate, the TMC2300 offers a number of unique enhancements over comparable products: TRINAMICs sophisticated StealthChop2 chopper plus the microstep enhancement MicroPlyer ensure noiseless operation, maximum efficiency and best motor torque. Its precise current regulation and optional combination with CoolStep allow additional energy savings and reduce cooling infrastructure requirements. Protection and diagnostic features support robust and reliable operation. A simple-to-use 8 bit UART interface opens up more tuning and control options. Industries' most advanced low voltage step & direction stepper motor driver family upgrades designs to noiseless and most precise operation for cost-effective and highly competitive solutions.

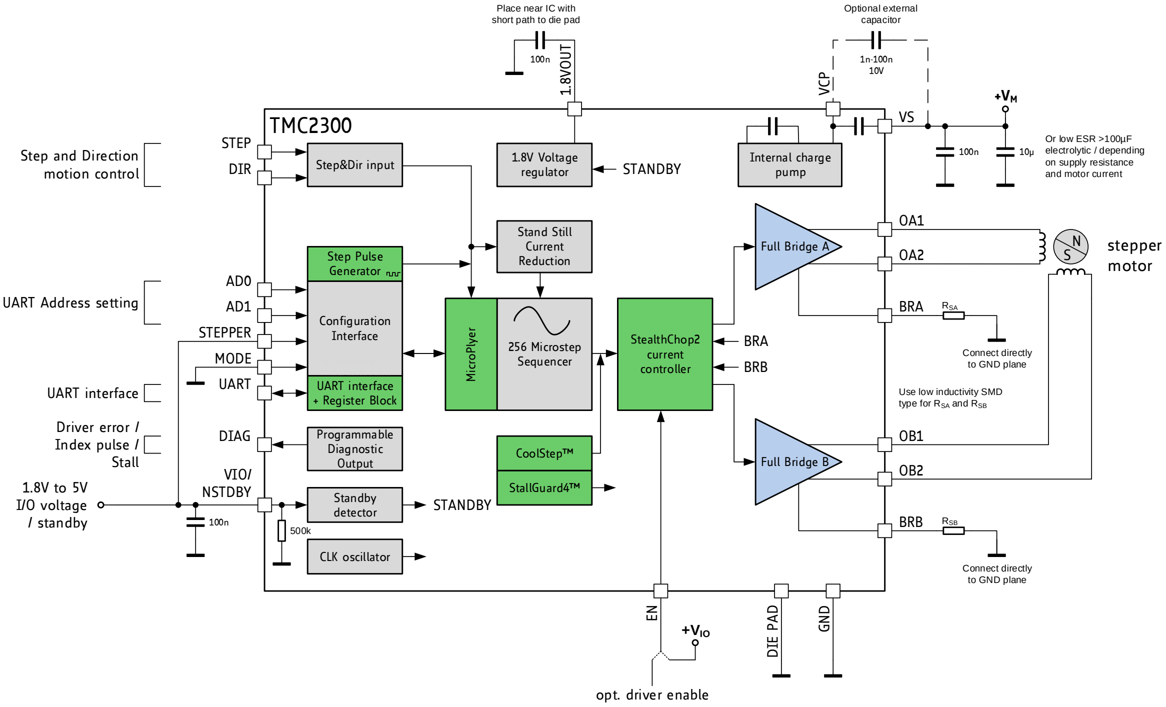


Figure 1.1 TMC2300 basic application block diagram for stepper motors

MODES OF OPERATION:

OPTION 1: Standalone stepper STEP/DIR Driver (Legacy Mode)

A CPU (µC) generates step & direction signals. The TMC2300 operates the motor as commanded by the configuration pins and STEP/DIR signals. Motor run current is fixed by sense resistor setting. The pin PDN_UART selects automatic standstill current reduction. Feedback from the driver to the CPU is granted by the DIAG output signal. Enable or disable the motor using the EN pin and VIO/NSTANDBY pin.

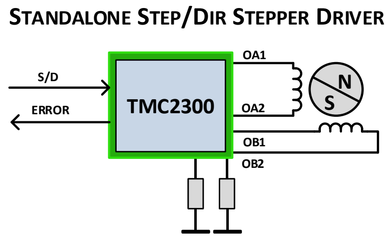

STANDALONE STEP/DIR STEPPER DRIVER

Figure 1.2 Stand-alone driver

OPTION 2: STEP/DIR Driver with Full Diagnostics and Control

UART

Similar to Option 1, but pin PDN_UART is connected to the CPU UART interface.

Additional options (label UART):

- \+ CoolStep energy saving
- \+ StallGuard sensorless homing
- \+ Detailed diagnostics and thermal management
- \+ Passive braking and freewheeling for flexible, lowest power stop modes
- \+ More options for microstep resolution setting (fullstep to 256 microstep)
- \+ Software controlled motor current setting and more chopper options
- \+ Option for motion using internal pulse generator (no STEP/DIR signals required)

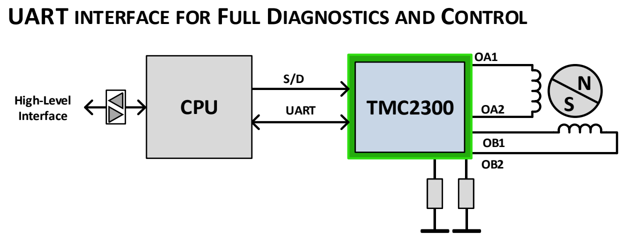

UART INTERFACE FOR FULL DIAGNOSTICS AND CONTROL

Figure 1.3 STEP/DIR Driver with UART

This mode allows replacing all control lines like ENN, DIAG, MS1, MS2 by a single interface line. This way, only three signals are required for full control: STEP, DIR and PDN_UART. Even motion without external STEP pulses is provided by an internal programmable step pulse generator: Just set the desired motor velocity. However, no ramping is provided by the TMC2300. Access to multiple driver ICs is possible using 4 different address settings or via an analog multiplexer IC.

1.1 Key Concepts

The TMC2300 implements advanced features which are exclusive to TRINAMIC products. These features contribute toward greater precision, greater energy efficiency, higher reliability, smoother motion, and cooler operation in many stepper motor applications.

StealthChop2™	No-noise, high-precision chopper algorithm for inaudible motion and inaudible standstill of the motor. Allows faster motor acceleration and deceleration than StealthChop™ and extends StealthChop to low stand still motor currents.
StallGuard™	Sensorless motor load measurement. It allows sensorless homing of a drive by sensing mechanical obstacles. Further, mechanics can be validated by monitoring motor load.
MicroPlyer™	Microstep interpolator for obtaining full 256 microstep smoothness with lower resolution step inputs starting from fullstep
CoolStep™	Uses StallGuard measurement in order to adapt the motor current for best efficiency and lowest heat-up of motor and driver

In addition to these performance enhancements, TRINAMIC motor drivers offer safeguards to detect and protect against shorted outputs, output open-circuit, overtemperature, and undervoltage conditions for enhancing safety and recovery from equipment malfunctions.

1.2 Control Interfaces

The TMC2300 supports both, discrete control lines for basic mode selection and a UART based single wire interface with CRC checking.

1.2.1 UART Interface

The single wire interface allows unidirectional operation (for parameter setting only), or bi-directional operation for full control and diagnostics. It can be driven by any standard microcontroller UART or even by bit banging in software. Baud rates from 9600 Baud to 500k Baud may be used. No baud rate configuration is required, as the TMC2300 automatically adapts to the masters' baud rate. The frame format is identical to the intelligent TRINAMIC controller & driver ICs TMC51XX and TMC22XX. A CRC checksum allows data transmission over longer distance. For fixed initialization sequences, store the data including CRC into the µC, thus consuming only a few 100 bytes of code for a full initialization. CRC may be ignored during read access, if not desired. This makes CRC use an optional feature! The IC has a fixed address selected by 2 pins. Multiple drivers can be programmed in parallel by tying together all interface pins, in case no read access is required. An optional addressing can be provided by analog multiplexers, like 74HC4066.

From a software point of view the TMC2300 is a peripheral with a number of control and status registers. Most of them can either be written only or are read only. Some of the registers allow both, read and write access. In case read-modify-write access is desired for a write only register, a shadow register can be realized in master software.

1.3 Moving and Controlling the Motor

1.3.1 STEP/DIR Interface

The motor is controlled by a step and direction input. Active edges on the STEP input can be rising edges or both rising and falling edges as controlled by a special mode bit (DEDGE). Using both edges cuts the toggle rate of the STEP signal in half, which is useful for communication over slow interfaces such as optically isolated interfaces. The state sampled from the DIR input upon an active STEP edge determines whether to step forward or back. Each step can be a fullstep or a microstep, in which there are 2, 4, 8, 16, 32, 64, 128, or 256 microsteps per fullstep. A step impulse with a low state on DIR increases the microstep counter and a high state decreases the counter by an amount controlled by the microstep resolution. An internal table translates the counter value into the sine and cosine values which control the motor current for microstepping.

UART

1.3.2 Internal Step Pulse Generator

Some applications do not require a precisely co-ordinated motion – the motor just is required to move until a certain event occurs, or a certain distance is passed. The TMC2300 comes with an internal pulse generator for these applications: Just provide the velocity via UART interface to move the motor. The velocity sign automatically controls the direction of the motion. This way, the motor will move without processor pulse generation. The processor just needs to terminate motion, when the target is reached, e.g. by polling the microstep counter. However, the pulse generator does not integrate a ramping function. Motion at higher velocities will require ramping up and ramping down the velocity value via software.

STEP/DIR mode and internal pulse generator mode can be mixed in an application!

1.4 StealthChop2 Driver

StealthChop is a voltage chopper-based principle. It is optimum especially for low voltage operation, because microstepping sine waves are generated even at 100% duty cycle. It especially guarantees that the motor is absolutely quiet in standstill and in slow motion, except for noise generated by bearings. Unlike other voltage mode choppers, StealthChop2 does not require any configuration. It automatically learns the best settings during the first motion after power up and further optimizes the settings in subsequent motions. An initial homing sequence is sufficient for learning. StealthChop2 allows high motor dynamics, by reacting at once to a change of motor velocity.

Benefits of using StealthChop2:

- Significantly improved microstepping with low cost motors
- Motor runs smooth and quiet
- Absolutely no standby noise
- Reduced mechanical resonance yields improved torque

UART

1.5 StallGuard4 – Load Sensing

StallGuard4 provides an accurate measurement of the load on the motor. It can be used for stall detection as well as other uses at loads below those which stall the motor, such as CoolStep load-adaptive current reduction. This gives more information on the drive, allowing functions like sensorless homing and diagnostics of the drive mechanics.

UART

1.6 CoolStep – Load Adaptive Current

CoolStep drives the motor at the optimum current. It uses the StallGuard4 load measurement information to adjust the motor current to the minimum amount required in the actual load situation. This saves energy and keeps the components cool.

Benefits are:

- *Energy efficiency*	power consumption decreased up to 90% (w. no load on motor)
- *Motor generates less heat*	improved mechanical precision
- *Less or no cooling*	improved reliability
- *Use of smaller motor*	less torque reserve required → cheaper motor does the job
- *Less motor noise*	Due to less energy exciting motor resonances

Figure 1.4 shows the efficiency gain of a 42mm stepper motor when using CoolStep compared to standard operation with 50% of torque reserve. coolStep is enabled above 60RPM in the example.

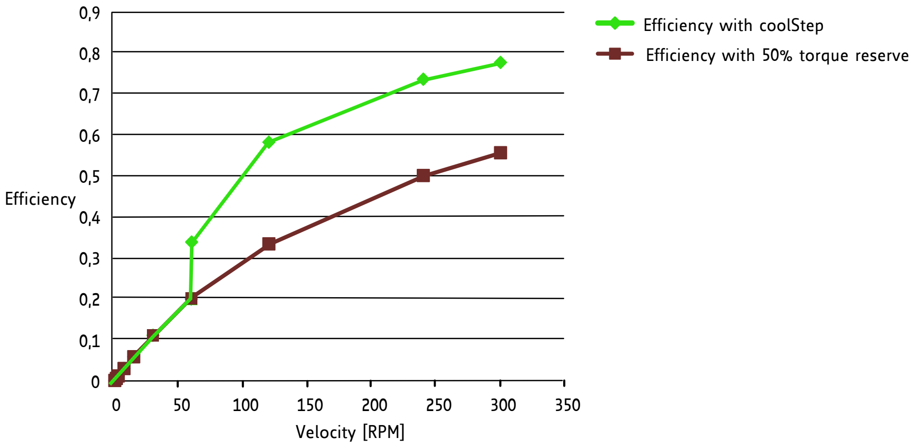


Figure 1.4 Energy efficiency with coolStep (example)

1.7 Automatic Standstill Power Down

An automatic current reduction drastically reduces application power dissipation and cooling requirements. Per default, the stand still current reduction is enabled by pulling PDN_UART input to GND. It reduces standstill power dissipation to less than 33% by going to slightly more than half of the run current.

Modify stand still current, delay time and decay via UART. Automatic freewheeling and passive motor braking are provided as an option for stand still. Passive braking reduces motor standstill power consumption to zero, while still providing effective dampening and braking!

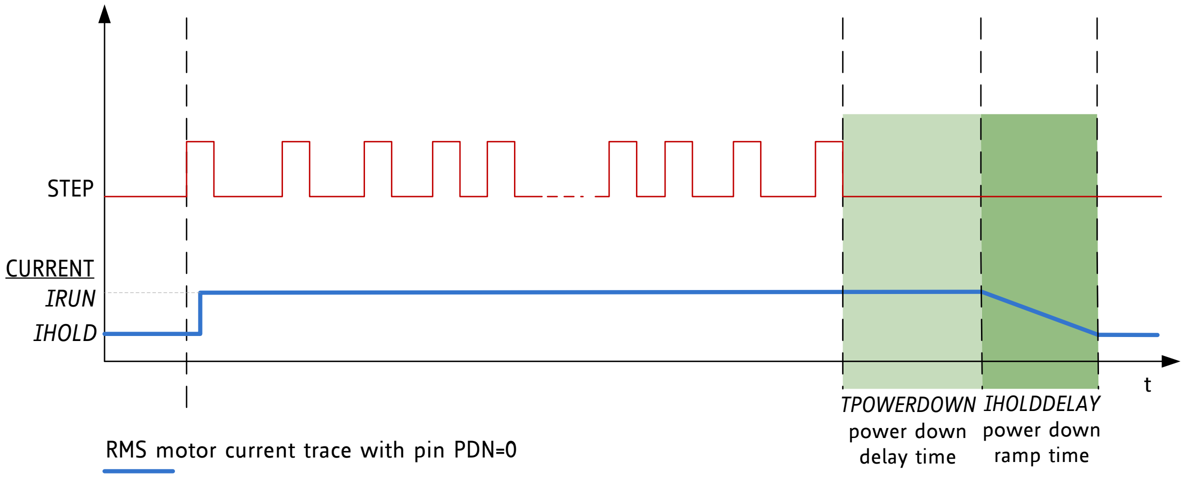


Figure 1.5 Automatic Motor Current Power Down

1.8 Index Pulse

The index output gives one pulse per electrical rotation, i.e. one pulse per each four fullsteps. It shows the internal sequencer microstep 0 position (*MSTEP* near 0). This is the power on position. In combination with a mechanical home switch, a more precise homing is enabled.

2 Pin Assignments

The TMC2300 comes in a tiny package in order to fit miniaturized devices. For the ease of use, pinning is shown separately for both function-modes.

2.1 Package Outline TMC2300

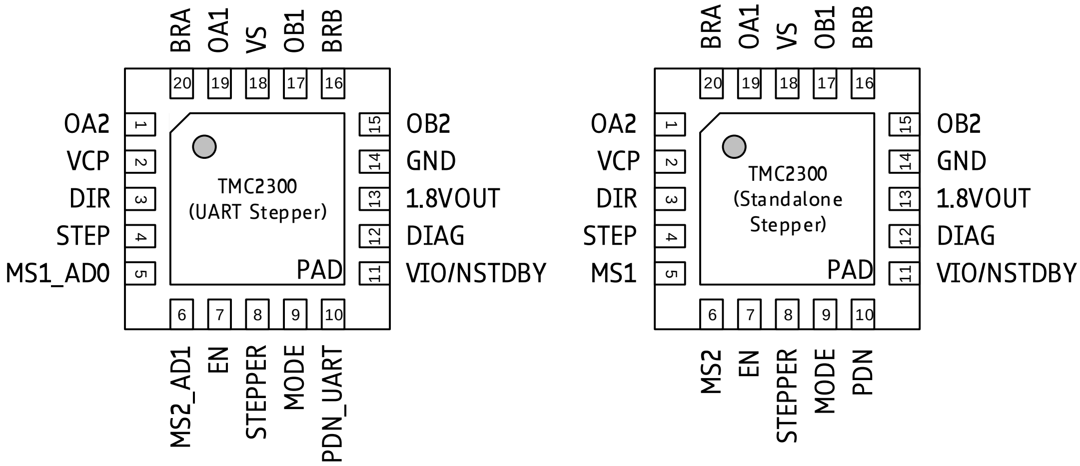


Figure 2.1 TMC2300 Pinning Top View Stepper Driver – QFN20, 3x3mm², 0.4mm pitch

2.2 Signal Descriptions / Stepper modes

Pin	Number	Type	Function
OA2	1		Motor coil A output 2
VCP	2		Charge pump voltage. Optionally tie to VS using 1nF to 100nF capacitor. May be left unconnected.
DIR	3	DI	DIR input (Analog test output in factory test mode)
STEP	4	DI	STEP input
MS1_AD0	5	DI	Microstep resolution configuration MS2, MS1: 00: 1/8, 01: 1/32, 10: 1/64, 11: 1/16 For UART based configuration selection of UART Address 0…3 (AD0=LSB, AD1=MSB)
MS2_AD1	6	DI	
EN	7	DI	Enable input. The power stage becomes switched off (all motor outputs floating) when this pin becomes driven to a low level. Also used to clear error flags.
STEPPER	8	DI	Mode selection input. STEPPER, MODE: 00: do not use 01: do not use 10: Stepper 11: Standalone Stepper (CLK, TST input in factory test mode)
MODE	9	DI	
PDN_UART	10	DIO	UART Input/Output. In standalone mode: Inverted power-down control input (low = automatic standstill current reduction).
VIO/NSTDBY	11		1.8V to 5V IO supply voltage for all digital pins. IC goes to standby mode and resets, when this pin is pulled to GND.
DIAG	12	DO	Diagnostic output. High level upon driver error or stall. Reset by EN=low.

Pin	Number	Type	Function
1.8VOUT	13		Output of internal 1.8V regulator. Attach 100nF ceramic capacitor to GND near to pin for best performance. Provide the shortest possible loop to the GND pad.
GND	14		GND. Connect to GND plane near pin.
OB2	15		Motor coil B output 2
BRB	16		Sense resistor connection for coil B. Place sense resistor to GND near pin.
OB1	17		Motor coil B output 1
VS	18		Motor supply voltage. Provide filtering capacity >10µF near pin with shortest possible loop to GND pad.
OA1	19		Motor coil A output 1
BRA	20		Sense resistor connection for coil A. Place sense resistor to GND near pin.
Exposed die pad	-		Connect the exposed die pad to a GND plane. Provide as many as possible vias for heat transfer to GND plane.

3 Sample Circuits

The sample circuits show the connection of external components in different operation and supply modes. The connection of the bus interface and further digital signals is left out for clarity. The TMC2300 is configured for different application modes by two pins, as well as by settings available via the UART interface.

STEPPER/MODE: CONFIGURATION OF OPERATION MODE

STEPPER	MODE	Operation Mode
VCC_IO	GND	UART controlled stepper driver, MS1 and MS2 select the UART address.
VCC_IO	VCC_IO	Standalone Stepper. Set power down mode using PDN, and microstep resolution using MS1 and MS2.

3.1 Standard Application Circuit

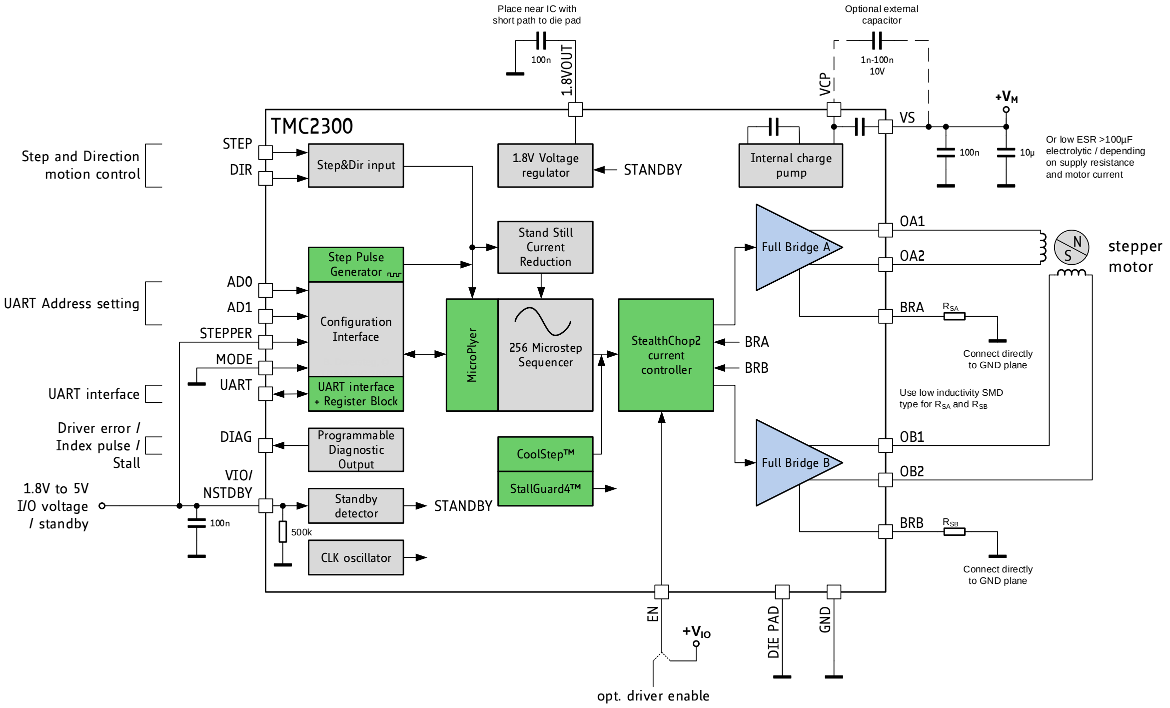


Figure 3.1 Standard Application Circuit for 2V to 11V Supply

The standard application circuit uses a minimum set of additional components to operate a stepper motor. Two sense resistors set the motor coil current. See chapter 8 to choose the right sense resistors. Take care to keep power supply ripple due to chopper operation at a few 100mV, max., especially when low voltage operation is desired. Use a ceramic, or low ESR capacitors for filtering the power supply. The capacitors need to cope with the current ripple caused by chopper operation. A minimum capacity of 100µF electrolytic, or 10µF ceramic capacitor near the driver is recommended to keep ripple low. Actual demand will depend on the internal power supply resistance and the desired motor current. VCC_IO can be supplied from a separate supply, e.g. a 3.3V regulator, or be driven by a microcontroller port pin. AD0 and AD1 set the UART address. A charge pump capacitor can be added for slightly reduced power dissipation when operating near the lower voltage limit of roughly 2V.

Basic layout and component hints

Place sense resistors and all filter capacitors as close as possible to the related IC pins. Use a solid common GND for all GND connections, also for sense resistor GND. Connect 1.8VOUT filtering capacitor directly to 1.8VOUT and the GND pin. See layout hints for more details. Low ESR electrolytic capacitors are recommended for VS filtering unless supply resistance is very low.

3.2 Standalone Stepper

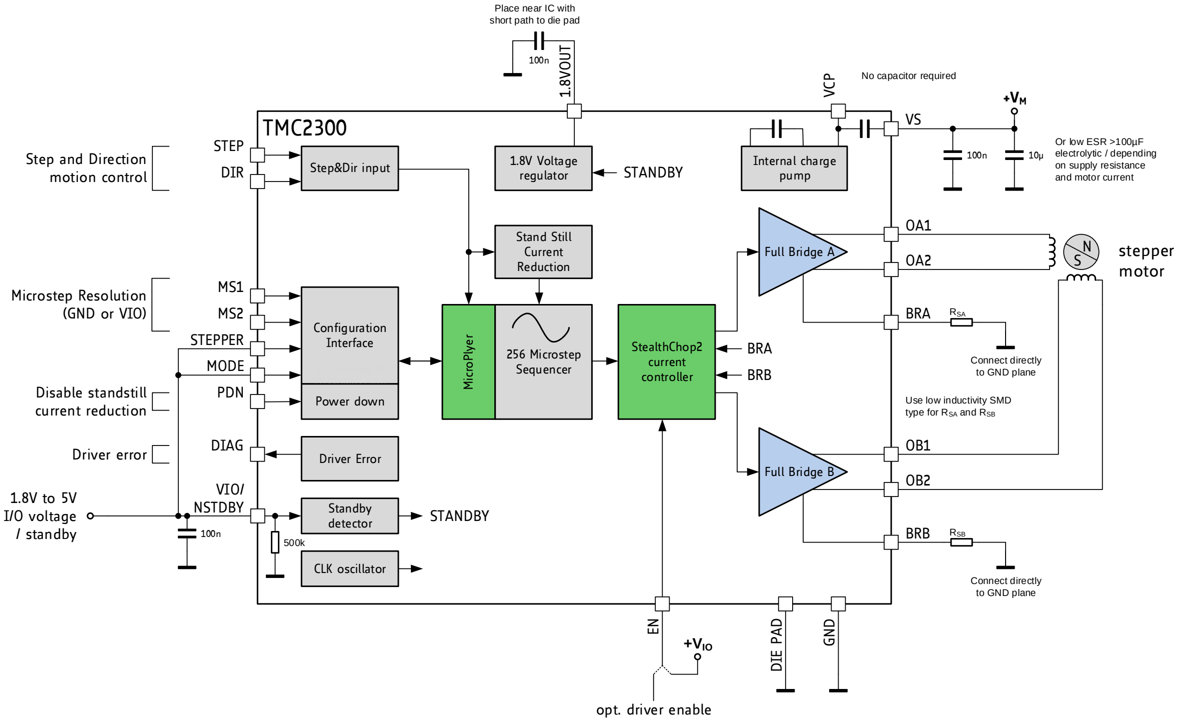


Figure 3.2 Standalone Stepper Operation

The standalone stepper application uses just three interface lines, to operate the stepper: STEP, DIR and DIAG for feedback. Microstep resolution and standstill power down become pre-configured by tying MS1, MS2 and PDN to VIO or GND. The motor current is determined by the choice of sense resistors.

3.2.1 Configuration Pins

The TMC2300 provides three configuration pins. These pins allow quick configuration for standalone operation.

PDN_UART: CONFIGURATION OF STANDSTILL POWER DOWN (STANDALONE STEPPER MODE)	
PDN_UART	**Current Setting**
GND	Enable automatic power down in standstill periods
VIO	Disable automatic power down
UART interface	When using the UART interface, the power-down configuration pin is *automatically* disabled. Program *IHOLD* as desired for standstill periods.

MS1/MS2: CONFIGURATION OF MICROSTEP RESOLUTION FOR STEP INPUT (STANDALONE STEPPER MODE)		
MS2	**MS1**	**Microstep Setting**
GND	GND	8 microsteps
GND	VIO	32 microsteps
VIO	GND	64 microsteps
VIO	VIO	16 microsteps

3.3 Highly Efficient Driver

The TMC2300 integrates a highly efficient power stage, offering low RDSon even at low supply voltages, due to its internal charge pump. This enables high motor current drive capability and low power dissipation for battery powered applications.

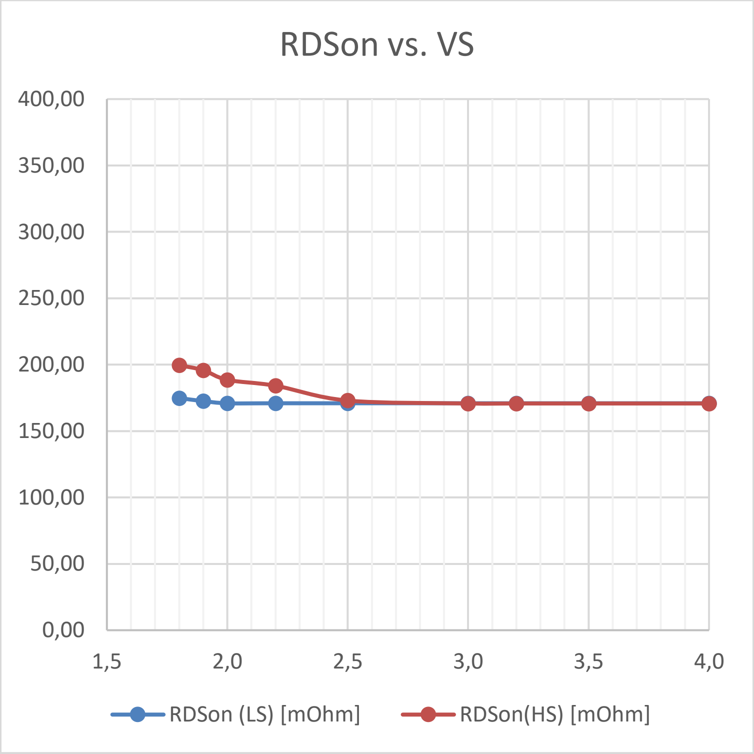


Figure 3.3 RDSon Variation over Supply Voltage

When operating at a high motor current, the driver power dissipation due to MOSFET switch on-resistance significantly heats up the driver. This power dissipation will significantly heat up the PCB cooling infrastructure, if operated at an increased duty cycle. This in turn leads to a further increase of driver temperature. An increase of temperature by about 100°C increases MOSFET resistance by roughly 50%. This is a typical behavior of MOSFET switches. Therefore, under high duty cycle, high load conditions, thermal characteristics have to be carefully taken into account, especially when increased environment temperatures are to be supported. Refer the thermal characteristics and the layout hints for more information. As a thumb rule, thermal properties of the PCB design become critical for the tiny QFN 3mm x 3mm package at or above 0.8A RMS motor current for increased periods of time. For currents above 0.8A, a 4-layer PCB layout with 5 via contact of the die attach pad to the GND plane is required. Keep in mind that resistive power dissipation raises with the square of the motor current. On the other hand, this means that a small reduction of motor current significantly saves heat dissipation and energy.

Pay special attention to good thermal properties of your PCB layout, when going for 0.8A RMS current or more.

3.4 Low Power Standby

Battery powered applications, as well as mains powered applications conforming to EU energy saving regulations, often require a standby mode, where the power-supply remains on. Current consumption in this mode must be minimized. Control near zero power TMC2300 standby operation by switching off the I/O supply voltage on VIO_NSTDBY pin. At the same time make sure, that no digital input pin is at a high level. An input level above VIO_NSTDBY would hinder pulling down VIO_NSTDBY, due to the ESD protection diodes in each digital I/O pin. These diodes clamp each input to a level between GND and the IO supply voltage VIO_NSTDBY. Therefore, stop the motor first, and allow sufficient time for the motor to come to a standstill, pull the enable input EN low, and also all other input pins, to switch off the motor completely before switching off VIO voltage. All driver registers are reset to their power-up defaults after leaving standby mode. See Figure 3.4.

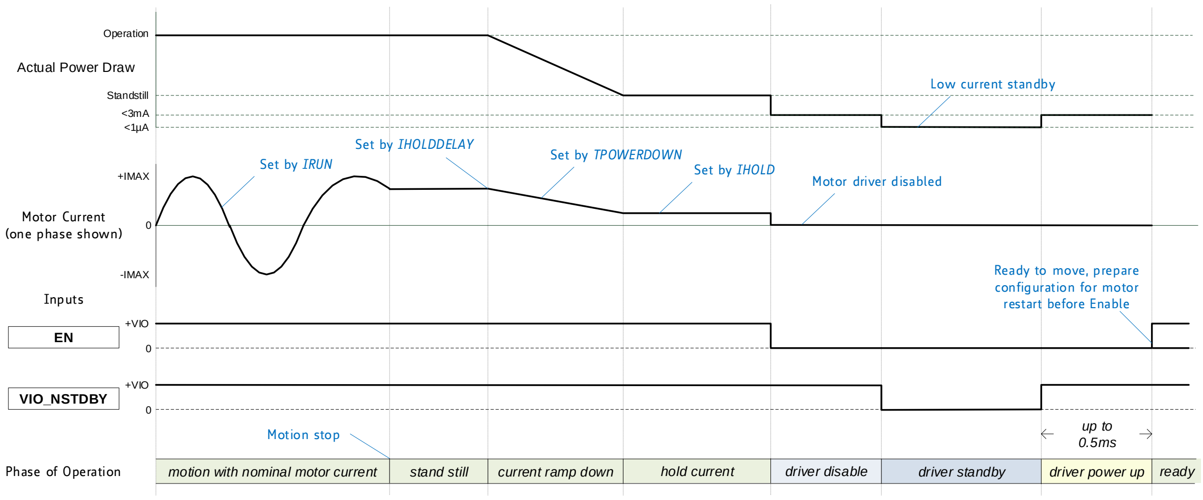


Figure 3.4 Switching to Standby and Back On

3.4.1 Restart the Stepper Motor Without Position Loss

A self-locking drive allows switching off the motor completely without loss of position. Locking can result from mechanical friction and from the stepper motor cogging torque. Most stepper motors have a cogging torque in the range of a few percent of their nominal torque, which also will contribute to the motor locking in a certain position. Due to their construction, most motors lock at a fullstep position. A full step position is characterized by the position yielded with both coils at identical absolute current. With n-times microstepping, fullstep positions are reached each n steps. The fist fullstep position is reached when exactly n/2 steps are done following a driver power-up. The internal microstep counter shows 128, 384, 640 or 896 when a fullstep position is reached.

The motor will pull into the same step after power up, as long as the rotor position and electrical position differ by up to +-2 fullsteps, given that no external force pulls the motor into a certain direction. An offset of maximum one fullstep is safest.

When powering up the driver, all registers become reset to zero. This also affects the internal position counter. Thus, the position counter will restart from 0 after power up. With the enable pin fixed at "1", the motor current will pull the motor to this (halfstep) position. With this, several options to keep track of the motor position result:

METHODS FOR POSITION RECOVERY

Interface	Enable Pin EN	Actions prior to power down	Actions at power up
Stand Alone or UART	Fixed=VIO	Keep track of the motor position by counting steps following initial power up. Prior to power down, move to a position which can be divided by 4*microstep resolution. At these positions, *MSCNT* is 0. Store the position.	*MSCNT* is cleared to 0 automatically. Start moving the motor as desired.
Stand Alone or UART	Controlled by CPU	Keep track of the motor position by counting steps following initial power up. For best results with low friction drives, move to a fullstep position prior to power down. Store the position. *Example*: at 32 microstep resolution, fullstep positions are 16+n*32, i.e. -48, -16, 16, 48,…	Apply a number of steps to restore *MSCNT* to the stored value prior to enabling the motor driver. number of step pulses= (position modulo 4*microstep resolution) *Example*: at 32 microstep setting, each step pulse increments *MSCNT* by 256/32. Calculate position modulo 128 to yield the required number of steps. Apply 10 steps with DIR=0 to increment MSCNT to a value of 80.
UART (option)		Read out *MSCNT* and store it together with the absolute motor position.	

3.5 Driver Protection and EME Circuitry

Some applications have to cope with ESD events caused by motor operation or external influence. Despite ESD circuitry within the driver chips, ESD events occurring during operation can cause a reset or even a destruction of the motor driver, depending on their energy. Especially plastic housings and belt drive systems tend to cause ESD events of several kV. It is best practice to avoid ESD events by attaching all conductive parts, especially the motors themselves to PCB ground, or to apply electrically conductive plastic parts. In addition, the driver can be protected up to a certain degree against ESD events or live plugging / pulling the motor, which also causes high voltages and high currents into the motor connector terminals. A simple scheme uses capacitors at the driver outputs to reduce the dV/dt caused by ESD events. Larger capacitors will bring more benefit concerning ESD suppression, but cause additional current flow in each chopper cycle, and thus increase driver power dissipation, especially at high supply voltages. The values shown are example values – they may be varied between 100pF and 470pF. The capacitors also dampen high frequency noise injected from digital parts of the application PCB circuitry and thus reduce electromagnetic emission. A more elaborate scheme uses LC filters to de-couple the driver outputs from the motor connector. Varistors in between of the coil terminals eliminate coil overvoltage caused by live plugging. Optionally protect all outputs by suppressor diodes to GND, or by a diode network as shown.

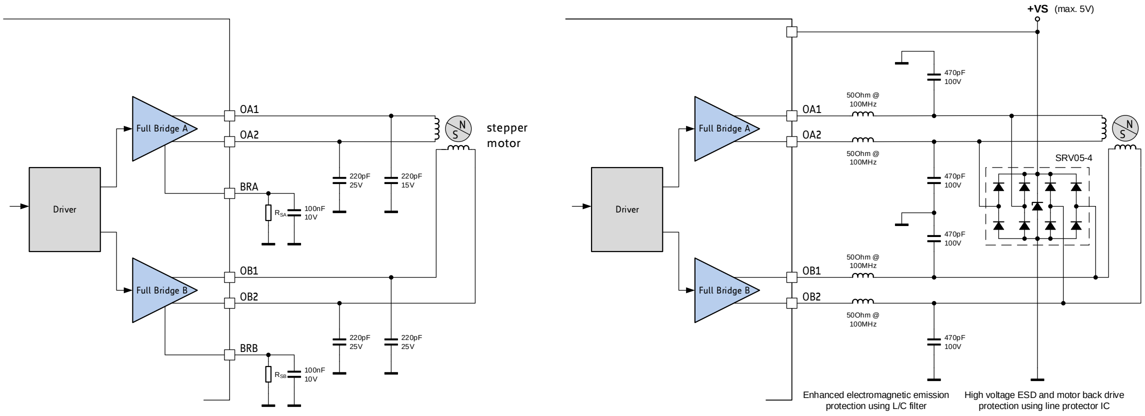


Figure 3.5 Simple ESD & EMI enhancement and more elaborate motor output protection

4 UART Single Wire Interface

The UART single wire interface allows control of the TMC2300 with any microcontroller UART. It shares transmit and receive line like an RS485 based interface. Data transmission is secured using a cyclic redundancy check, so that increased interface distances (e.g. over cables between two PCBs) can be bridged without danger of wrong or missed commands even in the event of electro-magnetic disturbance. The automatic baud rate detection makes this interface easy to use.

4.1 Datagram Structure

4.1.1 Write Access

UART WRITE ACCESS DATAGRAM STRUCTURE																			
each byte is LSB…MSB, highest byte transmitted first																			
0 … 63																			
sync + reserved								**8 bit slave address**			**RW + 7 bit register addr.**			**32 bit data**			**CRC**		
0…7								8…15			16…23			24…55			56…63		
1	0	1	0	Reserved (don't cares but included in CRC)				*SLAVEADDR*=(MS2, MS1)			register address		1	data bytes 3, 2, 1, 0 (high to low byte)			CRC		
0	1	2	3	4	5	6	7	8	…	15	16	…	23	24	…	55	56	…	63

A sync nibble precedes each transmission to and from the TMC2300 and is embedded into the first transmitted byte, followed by an addressing byte (0 to 3, selected by pins MS1 (LSB) and MS2 for TMC2300). Each transmission allows a synchronization of the internal baud rate divider to the master clock. The actual baud rate is adapted and variations of the internal clock frequency are compensated. Thus, the baud rate can be freely chosen within the valid range. Each transmitted byte starts with a start bit (logic 0, low level on UART pin) and ends with a stop bit (logic 1, high level on UART pin). The bit time is calculated by measuring the time from the beginning of start bit (1 to 0 transition) to the end of the sync frame (1 to 0 transition from bit 2 to bit 3). All data is transmitted bytewise. The 32 bit data words are transmitted with the highest byte first.

A minimum baud rate of 9000 baud is permissible, assuming maximum clock frequency (worst case for low baud rate). Maximum baud rate is $f_{CLK}/16$ due to the required stability of the baud clock.

The slave address *SLAVEADDR* is selected by MS1 (bit 0) and MS2 (bit 1) in the range 0 to 3.

The communication becomes reset if a pause time of longer than 63 bit times between the start bits of two successive bytes occurs. This timing is based on the last correctly received datagram. In this case, the transmission needs to be restarted after a failure recovery time of minimum 12 bit times of bus idle time. This scheme allows the master to reset communication in case of transmission errors. Any pulse on an idle data line below 16 clock cycles will be treated as a glitch and leads to a timeout of 12 bit times, for which the data line must be idle. Other errors like wrong CRC are also treated the same way. This allows a safe re-synchronization of the transmission after any error conditions. Remark, that due to this mechanism an abrupt reduction of the baud rate to less than 15 percent of the previous value is not possible.

Each accepted write datagram becomes acknowledged by the receiver by incrementing an internal cyclic datagram counter (8 bit). Reading out the datagram counter allows the master to check the success of an initialization sequence or single write accesses. Read accesses do not modify the counter.

The UART line must be logic high during idle state.

4.1.2 Read Access

UART READ ACCESS REQUEST DATAGRAM STRUCTURE																
each byte is LSB…MSB, highest byte transmitted first																
sync + reserved								**8 bit slave address**			**RW + 7 bit register address**			**CRC**		
0…7								8…15			16…23			24…31		
1	0	1	0	Reserved (don't cares but included in CRC)				*SLAVEADDR*=(MS2,MS1)			register address		0	CRC		
0	1	2	3	4	5	6	7	8	…	15	16	…	23	24	…	31

The read access request datagram structure is identical to the write access datagram structure, but uses a lower number of user bits. Its function is the addressing of the slave and the transmission of the desired register address for the read access. The TMC2300 responds with the same baud rate as the master uses for the read request.

In order to ensure a clean bus transition from the master to the slave, the TMC2300 does not immediately send the reply to a read access, but it uses a programmable delay time after which the first reply byte becomes sent following a read request. This delay time can be set in multiples of eight bit times using *SENDDELAY* time setting (default=8 bit times) according to the needs of the master.

UART READ ACCESS REPLY DATAGRAM STRUCTURE																			
each byte is LSB…MSB, highest byte transmitted first																			
0 …… 63																			
sync + reserved								**8 bit master address**			**RW + 7 bit register addr.**			**32 bit data**			**CRC**		
0…7								8…15			16…23			24…55			56…63		
1	0	1	0	reserved (0)				0xFF			register address		0	data bytes 3, 2, 1, 0 (high to low byte)			CRC		
0	1	2	3	4	5	6	7	8	…	15	16	…	23	24	…	55	56	…	63

The read response is sent to the master using address code %11111111. The transmitter becomes switched inactive four bit times after the last bit is sent.

Address %11111111 is reserved for read access replies going to the master.

4.2 CRC Calculation

An 8 bit CRC polynomial is used for checking both read and write access. It allows detection of up to eight single bit errors. The CRC8-ATM polynomial with an initial value of zero is applied LSB to MSB, including the sync- and addressing byte. The sync nibble is assumed to always be correct. The TMC2300 responds only to correctly transmitted datagrams containing its own slave address. It increases its datagram counter for each correctly received write access datagram.

$$CRC = x^8 + x^2 + x^1 + x^0$$

Serial calculation example

```
CRC = (CRC << 1) OR (CRC.7 XOR CRC.1 XOR CRC.0 XOR [new incoming bit])
```

C-Code Example for CRC calculation

```
void swuart_calcCRC(UCHAR* datagram, UCHAR datagramLength)
{
  int i,j;
  UCHAR* crc = datagram + (datagramLength-1); // CRC located in last byte of message
  UCHAR currentByte;

  *crc = 0;
  for (i=0; i<(datagramLength-1); i++) {      // Execute for all bytes of a message
    currentByte = datagram[i];                // Retrieve a byte to be sent from Array
    for (j=0; j<8; j++) {
      if ((*crc >> 7) ^ (currentByte&0x01))   // update CRC based result of XOR operation
      {
        *crc = (*crc << 1) ^ 0x07;
      }
      else
      {
        *crc = (*crc << 1);
      }
      currentByte = currentByte >> 1;
    } // for CRC bit
  } // for message byte
}
```

4.3 UART Signals

The UART interface on the TMC2300 uses a single bi-direction pin:

UART Interface Signal	
PDN_UART	Non-inverted data input and output. I/O with Schmitt Trigger and VCC_IO level.
MS1_AD0	IC UART address bit 0 (LSB)
MS2_AD1	IC UART address bit 1

The IC checks PDN_UART for correctly received datagrams with its own address continuously. It adapts to the baud rate based on the sync nibble, as described before. In case of a read access, it switches on its output drivers and sends its response using the same baud rate. The output becomes switched off four bit times after transfer of the last stop bit.

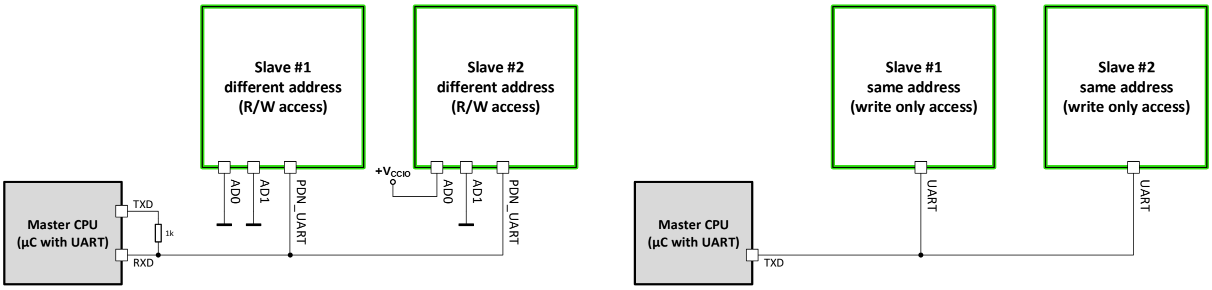


Figure 4.1 Attaching the TMC2300 to a microcontroller UART

4.4 Addressing Multiple Slaves

WRITE ONLY ACCESS

If read access is not used, and all slaves are to be programmed with the same initialization values, no addressing is required. All slaves can be programmed in parallel like a single device (Figure 4.1.).

ADDRESSING MULTIPLE SLAVES

As the TMC2300 uses has a limited number of UART addresses, in principle only up to four ICs can be accessed per UART interface channel. Adding analog switches allows separated access to individual ICs. This scheme is similar to an SPI bus with individual slave select lines (Figure 4.2).

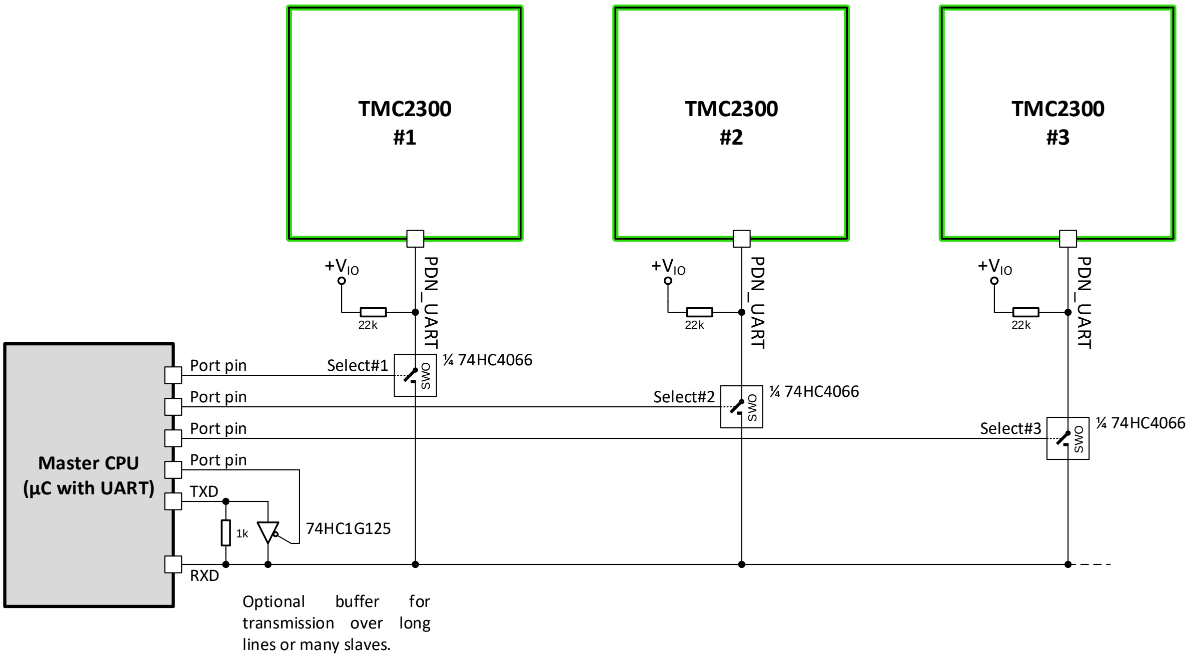


Figure 4.2 Addressing multiple TMC2300 via single wire interface using analog switches

PROCEED AS FOLLOWS TO CONTROL MULTIPLE SLAVES:

- Set the UART to 8 bits, no parity. Select a baud rate safely within the valid range. At 250kBaud, a write access transmission requires 320µs (=8 Bytes * (8+2) bits * 4µs).
- Before starting an access, activate the select pin going to the analog switch by setting it high. All other slaves select lines shall be off, unless a broadcast is desired.
- When using the optional buffer, set TMC2300 transmission send delay to an appropriate value allowing the µC to switch off the buffer before receiving reply data.
- To start a transmission, activate the TXD line buffer by setting the control pin low.
- When sending a read access request, switch off the buffer after transmission of the last stop bit is finished.
- Take into account, that all transmitted data also is received by the RXD input.

5 Register Map

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
- *Reset default*: 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
Velocity Dependent Driver Feature Control Register Set	This register set offers registers for - driver current control, stand still reduction - setting thresholds for different chopper modes - internal pulse generator control
Chopper Register Set	This register set offers registers for - optimization of StealthChop2 and read out of internal values - passive braking and freewheeling options - driver diagnostics - driver enable / disable
CoolStep and StallGuard Control Registers	These registers allow for - Sensorless stall detection for homing - Adaptive motor current control for best efficiency

5.1 General Registers

GENERAL CONFIGURATION REGISTERS (0X00…0X0F)

R/W	Addr	n	Register	Description / bit names	
RW	0x00	10	*GCONF*	**Bit**	***GCONF* – Global configuration flags**
				0	*set to 0*
				1	*extcap* (*Reset default*=0) 0: Operation without external capacitor on VCP. 1: External capacitor available. No switching delays.
				2	*set to 0*
				3	*shaft* 1: Inverse stepper motor direction
				4	*diag_index* 0: DIAG output normal 1: DIAG pin outputs index pulse flag instead
				5	*diag_step* 0: DIAG output normal 1: DIAG output shows step pulses from internal pulse generator (toggle upon each step)
				6	*multistep_filt* (*Reset default*=1) 0: No filtering of STEP pulses 1: Software pulse generator optimization enabled when fullstep frequency > 750Hz (roughly). TSTEP shows filtered step time values when active.
				7	*test_mode* 0: Normal operation 1: Enable analog test output on pin DIR *IHOLD*[1..0] selects the function of DIR: 0…1: T120, DAC *Attention: 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 since the last read access to *GSTAT*. 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 since the last read access. Read DRV_STATUS for details. The flag can only be cleared when all error conditions are cleared.
				2	*u3v5* 1: Actual state of the supply voltage comparator. A high value means that the voltage sinks below 3.5V. This flag is not latched and thus does not need to be cleared.
R	0x02	8	*IFCNT*		Interface transmission counter. This register becomes incremented with each successful UART interface write access. Read out to check the serial transmission for lost data. Read accesses do not change the content. The counter wraps around from 255 to 0.
W	0x03	4	*SLAVECONF*	**Bit**	***SLAVECONF***
				11..8	SENDDELAY for read access (time until reply is sent): 0, 1: 8 bit times

General configuration registers (0x00…0x0F)

R/W	Addr	n	Register	Description / bit names	
					2, 3: 3*8 bit times 4, 5: 5*8 bit times 6, 7: 7*8 bit times 8, 9: 9*8 bit times 10, 11: 11*8 bit times 12, 13: 13*8 bit times 14, 15: 15*8 bit times
R	0x06	10 + 8	*IOIN*	**Bit**	***INPUT*** (Reads the state of all input pins available)
				0	EN (1=enable driver)
				1	NSTDBY (0=standby, 1=enable)
				2	AD0
				3	AD1
				4	DIAG
				5	STEPPER 1: UART interface on
				6	PDN_UART input
				7	MODE input 0: UART controlled operation
				8	STEP
				9	DIR
				10	COMP_A1A2 1: during LS passive braking: A1 voltage > A2 voltage
				11	COMP_B1B2 1: during LS passive braking: B1 voltage > B2 voltage
				31.. 24	*VERSION*: 0x40=first version of the IC Identical numbers mean full digital compatibility.

5.2 Velocity Dependent Control

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* (*Reset default=8*) Standstill current (0=1/32 … 31=32/32) Setting *IHOLD*=0 allows to choose freewheeling or coil short circuit (passive braking) for motor stand still.
				12..8	*IRUN* (*Reset default=31*) Motor run current (8=9/32 … 31=32/32) Working with values below 8 is not recommended. *Hint:* Choose sense resistors in a way, that normal *IRUN* is 16 to 31 for best performance.
				19..16	*IHOLDDELAY* (*Reset default=1*) Controls the number of clock cycles for motor power down after 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* (*Reset default=20*) Sets the delay time from stand still (*stst*) detection to motor current power down. Time range is about 0 to 5.6 seconds. *0…((2^8)-1)* * 2^18 t_{CLK} *Attention: A minimum setting of 2 is required to allow automatic tuning of StealthChop PWM_OFFS_AUTO.*
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. The *TSTEP* related threshold uses a hysteresis of 1/16 of the compare value to compensate for jitter in the clock or the step frequency: (*Txxx**15/16)-1 is the lower compare value for each *TSTEP* based comparison. This means, that the lower switching velocity equals the calculated setting, but the upper switching velocity is higher as defined by the hysteresis setting.
W	0x22	24	*VACTUAL*		*VACTUAL* allows moving the motor by UART control. It gives the motor velocity in +-(2^23)-1 [µsteps / t] 0: Normal operation. Driver reacts to STEP input. /=0: Motor moves with the velocity given by *VACTUAL.* Step pulses can be monitored via DIAG output, using *diag_step* or *diag_index* setting. The motor direction is controlled by the sign of *VACTUAL*. Optionally, poll *MSCNT* at least once each 511 steps to keep track of the actual position by accumulating the difference of the actual value to the previous value.

5.3 StallGuard Control

COOLSTEP AND STALLGUARD CONTROL REGISTER SET **(0X14, 0X40…0X42)**				
R/W	**Addr**	**n**	***Register***	**Description / *bit names***
W	0x14	10	*TCOOLTHRS*	*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 stall output signal becomes enabled when exceeding this velocity. *TCOOLTHRS* ≥ *TSTEP* - CoolStep is enabled - Stall output signal on pin DIAG is enabled
W	0x40	8	*SGTHRS*	*SGTHRS* Detection threshold for stall. The StallGuard value *SG_VALUE* becomes compared to this threshold. A stall is signaled with *SG_VALUE* ≤ *SGTHRS**2
R	0x41	10	*SG_VALUE*	StallGuard result. *SG_RESULT* becomes updated with each fullstep, independent of *TCOOLTHRS* and *SGTHRS*. A higher value signals a lower motor load and more torque headroom. Intended for StealthChop mode, only. Bits 9 and 0 will always show 0. Scaling to 10 bit is for compatibility to StallGuard2.
W	0x42	16	*COOLCONF*	CoolStep configuration *See separate table!*

5.3.1 *COOLCONF* – Smart Energy Control CoolStep

0X42: *COOLCONF* – SMART ENERGY CONTROL COOLSTEP AND STALLGUARD

Bit	Name	Function	Comment
…	-	unused	*reserved, set to 0*
15	*seimin*	minimum current for smart current control	0: 1/2 of current setting (*IRUN*) (requires *IRUN* ≥ 10) 1: 1/4 of current setting (*IRUN*) (requires *IRUN* ≥ 20)
14	*sedn1*	current down step speed	%00: For each 32 StallGuard values decrease by one %01: For each 8 StallGuard values decrease by one %10: For each 2 StallGuard values decrease by one %11: For each StallGuard value decrease by one
13	*sedn0*		
12	-	reserved	set to 0
11	*semax3*	StallGuard hysteresis value for smart current control	If the StallGuard 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 StallGuard value %00 … %11: 1, 2, 4, 8
5	*seup0*		
4	-	reserved	set to 0
3	*semin3*	minimum StallGuard value for smart current control and smart current enable	If the StallGuard 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*		

5.4 Sequencer Registers

The sequencer registers have a pure informative character and are read-only. They help for special cases like storing the last motor position before power off in battery powered applications.

MICROSTEPPING CONTROL REGISTER SET (0X60…0X6B)

R/W	**Addr**	**n**	***Register***	**Description / *bit names***	**Range [Unit]**
R	0x6A	10	*MSCNT*	Microstep counter. Indicates actual position in the microstep table for *CUR_A*. *CUR_B* uses an offset of 256 into the table. Reading out *MSCNT* allows determination of the motor position within the electrical wave.	0…1023

5.5 Chopper Control Registers

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=* *0x13008001*
R	0x6F	32	***DRV_ STATUS***	Driver status flags and current level read back *See separate table!*		
RW	0x70	22	***PWMCONF***	StealthChop PWM chopper configuration *See separate table!*		*Reset default=* *0xC40D1024*
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
				bit 23… 16	*PWM_GRAD_AUTO*: Automatically determined gradient value	0…255

5.5.1 *CHOPCONF* – Chopper Configuration

0X6C: *CHOPCONF* – CHOPPER CONFIGURATION

Bit	Name	Function	Comment
31	*diss2vs*	Low side short protection disable	0: Short protection low side is on 1: Short protection low side 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. This mode is not compatible with the step filtering function (*multistep_filt*)
28	*intpol*	interpolation to 256 microsteps	1: The actual microstep resolution (*MRES*) becomes extrapolated to 256 microsteps for smoothest motor operation. *(Default: 1)*
27	*mres3*	*MRES* micro step resolution	%0000: Native 256 microstep setting.
26	*mres2*		
25	*mres1*		%0001 … %1000: 128, 64, 32, 16, 8, 4, 2, FULLSTEP Reduced microstep resolution (default 32 in UART mode). The resolution gives the number of microstep entries per sine quarter wave. When choosing a lower microstep resolution, the driver automatically uses microstep positions which result in a symmetrical wave. Number of microsteps per step pulse = 2^*MRES*
24	*mres0*		
23	-	reserved	set to 0
22			
21			
20			
19			
18			
17			
16	*tbl1*	*TBL* blank time select	%00 … %11: Set comparator blank time to 16, 24, 32 or 40 clocks *Hint*: %00 or %01 is recommended for most applications *(Default: %01)*
15	*tbl0*		
14	-	reserved	set to 0
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2			
1			
0	*enabledrv*	driver enable	1: Enable driver *(Default: 1, enable)*

5.5.2 *PWMCONF* – Voltage PWM Mode StealthChop

0X70: *PWMCONF* – VOLTAGE MODE PWM STEALTHCHOP

Bit	Name	Function	Comment
31	*PWM_LIM*	PWM automatic scale amplitude limit when switching on	Limit for *PWM_SCALE_AUTO* when switching on StealthChop following a disable condition. This value defines the upper limit for bits 7 to 4 of the automatic current control when switching back on. It can be set to reduce the current peak following a disable condition. It does not limit *PWM_GRAD* or *PWM_GRAD_AUTO* offset. (Default = 12)
30			
29			
28			
27	*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 *(Default)* … 8: 4 increments … 15: 7.5 increments (fastest regulation)
26			
25			
24			
23	-	reserved	set to 0
22	-	reserved	set to 0
21	*freewheel1*	Allows different standstill modes	Stand still option when motor current setting is zero (*I_HOLD*=0). %00: Normal operation (always selected with IHOLD>0) %01: Freewheeling %10: Coil shorted using LS drivers (passive braking) %11: Coil shorted using HS drivers (passive braking)
20	*freewheel0*		
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) *PWM_GRAD_AUTO* is initialized with *PWM_GRAD* 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.
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*

0x70: *PWMCONF* – Voltage mode PWM StealthChop

Bit	Name	Function	Comment
			1: Enable automatic current control *(Reset default)* Set to zero (for a short time) in order to force a new initialization of *PWM_OFS_AUTO=PWM_OFS* and *PWM_GRAD_AUTO=PWM_GRAD.*
17 16	*pwm_freq1* *pwm_freq0*	PWM frequency selection	%00: $f_{PWM}=2/1024\ f_{CLK}$ %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_AMPL* to compensate for the velocity-dependent motor back-EMF. (*Reset default=16*) With automatic scaling (*pwm_autoscale*=1) the value is used for first initialization, only. Set *PWM_GRAD* to the application specific value (it can be read out from *PWM_GRAD_AUTO*) to speed up the automatic tuning process.
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=36*) When using automatic scaling (*pwm_autoscale*=1) the value is used for initialization, only. The autoscale function starts with *PWM_SCALE_AUTO=PWM_OFS* and finds the required offset to yield the target current automatically. *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*).

5.5.3 DRV_STATUS – Driver Status Flags

0X6F: DRV_STATUS – DRIVER STATUS FLAGS AND CURRENT LEVEL READ BACK

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	-	reserved	Ignore these bits
29			
28			
27			
26			
25			
24			
23	-	reserved	Ignore these bits
22			
21			
20	*CS_ ACTUAL*	actual motor current / smart energy current	Actual current control scaling, for monitoring the function of the automatic current scaling.
19			
18			
17			
16			
15	-	reserved	Ignore these bits
14			
13			
12			
11	-	0	Ignore these bits
10			
9	*t150*	150°C comparator	1: Temperature threshold is exceeded, driver is off
8	*t120*	120°C comparator	1: Temperature prewarning threshold is exceeded
7	*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.
6	*ola*	open load indicator phase A	
5	*s2vsb*	low side short indicator phase B	1: Short on low-side MOSFET detected on phase A or B. The driver becomes disabled. The flags stay active, until the driver is disabled by software (*enabledrv*=0) or by the ENN input. Flags are separate for both chopper modes.
4	*s2vsa*	low side short indicator phase A	
3	*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 (*enabledrv*=0) or by the ENN input.
2	*s2ga*	short to ground indicator phase A	
1	*ot*	overtemperature flag	1: The 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.
0	*otpw*	overtemperature pre-warning flag	1: The overtemperature pre-warning threshold is exceeded. The overtemperature pre-warning flag is common for both bridges.

6 StealthChop™

StealthChop is an extremely quiet mode of operation for stepper motors. It is based on a voltage mode PWM. In case of standstill and at low velocities, the motor is absolutely noiseless. Thus, StealthChop operated stepper motor applications are very suitable for indoor or home use. The motor operates absolutely free of vibration at low velocities. With StealthChop, the motor current is applied by driving a certain effective voltage into the coil, using a voltage mode PWM. With the enhanced StealthChop2, the driver automatically adapts to the application for best performance. No more configurations are required. Optional configuration allows for tuning the setting in special cases, or for storing initial values for the automatic adaptation algorithm.

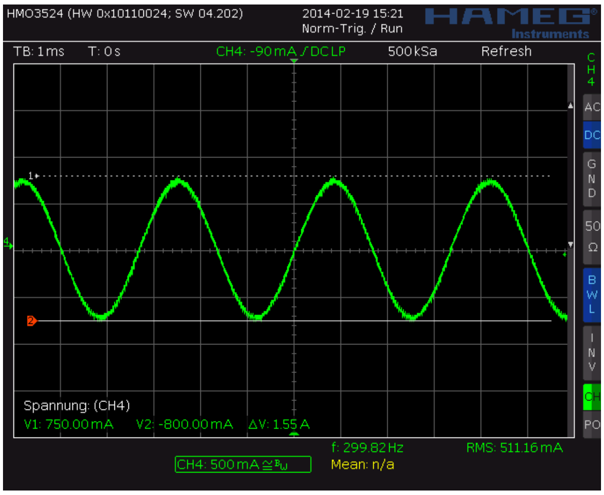


Figure 6.1 Motor coil sine wave current with StealthChop (measured with current probe)

6.1 Automatic Tuning

StealthChop2 integrates an automatic tuning procedure (AT), which adapts the most important operating parameters to the motor automatically. This way, StealthChop2 allows high motor dynamics and supports powering down the motor to very low currents. Just two steps have to be respected by the motion controller for best results: Start with the motor in standstill, but powered with nominal run current (AT#1). Move the motor at a medium velocity, e.g. as part of a homing procedure (AT#2). Figure 6.2 shows the tuning procedure.

Border conditions in for AT#1 and AT#2 are shown in the following table:

AUTOMATIC TUNING TIMING AND BORDER CONDITIONS			
Step	**Parameter**	**Conditions**	**Duration**
AT#1	*PWM_ OFS_AUTO*	- Motor in standstill and actual current scale (*CS*) is identical to run current (*IRUN*). - If standstill reduction is enabled (pin PDN_UART=0), an initial step pulse switches the drive back to run current. - Pin VS at operating level.	≤ 2^20+2*2^18 t_{CLK}, ≤ 130ms (with internal clock)
AT#2	*PWM_ GRAD_AUTO*	- Motor must move at a velocity, where a significant amount of back EMF is generated and where the full run current can be reached. Conditions: - 1.5 * *PWM_OFS_AUTO* < *PWM_SCALE_SUM* < 4 * *PWM_OFS_AUTO* - *PWM_SCALE_SUM* < 255. *Hint: A typical range is 60-300 RPM. Determine best conditions with the evaluation board and monitor PWM_SCALE_AUTO going down to zero during tuning.*	8 fullsteps are required for a change of +/-1. For a typical motor with *PWM_GRAD_AUTO* optimum at 64 or less, up to 400 fullsteps are required.

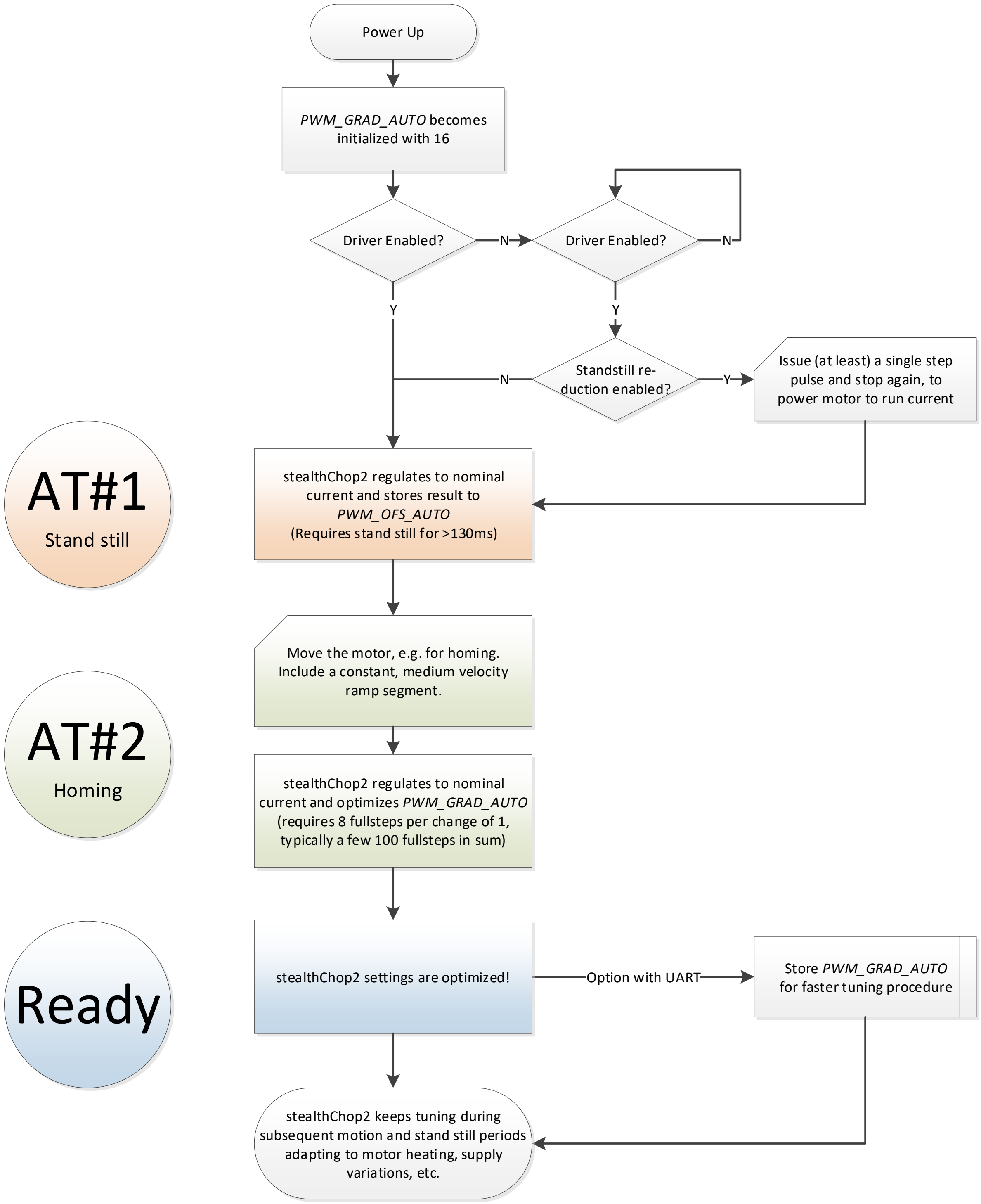


Figure 6.2 StealthChop2 automatic tuning procedure

Attention with varying supply voltage:
Modifying the supply voltage VS invalidates the result of the automatic tuning process. However, automatic tuning adapts to changed conditions whenever AT#1 and AT#2 conditions are fulfilled.
This is no problem with sinking supply voltage, i.e. due to the battery running low, as the regulator corrects by increasing the PWM value. However, with an increasing supply voltage, motor current rises, as the lower regulator limit is given by the result of the las AT#1 phase. Take this into account, when experimenting with a lab supply and modifying supply voltage.

6.2 StealthChop Options

UART

In order to match the motor current to a certain level, the effective PWM voltage becomes scaled depending on the actual motor velocity. Several additional factors influence the required voltage level to drive the motor at the target current: The motor resistance, its back EMF (i.e. directly proportional to its velocity) as well as the actual level of the supply voltage. Two modes of PWM regulation are provided: The automatic tuning mode (AT) using current feedback (*pwm_autoscale* = 1, *pwm_autograd* = 1) and a feed forward velocity-controlled mode (*pwm_autoscale* = 0). The feed forward velocity-controlled mode will not react to a change of the supply voltage or to events like a motor stall, but it provides very stable amplitude. It does not use nor require any means of current measurement. This is perfect when motor type and supply voltage are well known. Therefore, we recommend the automatic mode, unless current regulation is not satisfying in the given operating conditions.

It is recommended to operate in automatic tuning mode.

Non-automatic mode (*pwm_autoscale=0*) should be taken into account only with well-known motor and operating conditions. In this case, programming via the UART interface is required. The operating parameters *PWM_GRAD* and *PWM_OFS* can be determined in automatic tuning mode initially.

The StealthChop PWM frequency can be chosen in four steps in order to adapt the chopper frequency to the motor inductance. It balances low current ripple and good higher velocity performance vs. increased dynamic power dissipation at higher frequency.

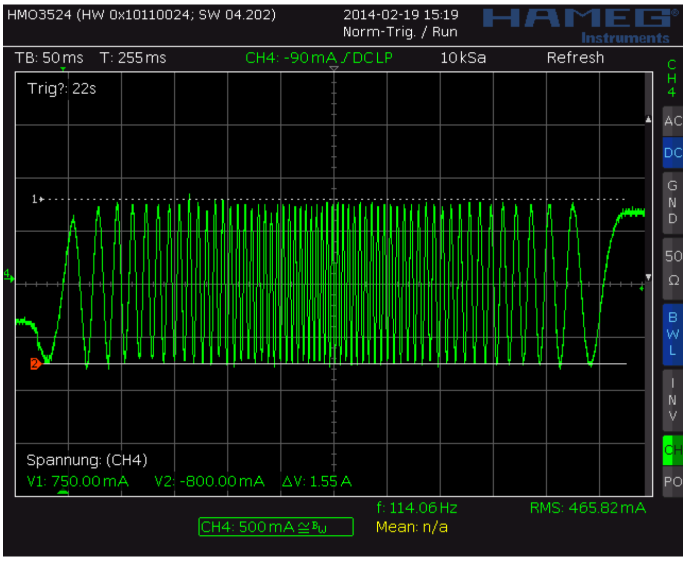
CHOICE OF PWM FREQUENCY

Clock frequency f_{CLK}	**PWM_FREQ=%00** $f_{PWM}=2/1024\ f_{CLK}$	**PWM_FREQ=%01** $f_{PWM}=2/683\ f_{CLK}$ (default)	**PWM_FREQ=%10** $f_{PWM}=2/512\ f_{CLK}$	**PWM_FREQ=%11** $f_{PWM}=2/410\ f_{CLK}$
12MHz (typ. value)	23.4kHz	35.1kHz	46.9kHz	58.5kHz

Table 6.1 Choice of PWM frequency – green / light green: recommended

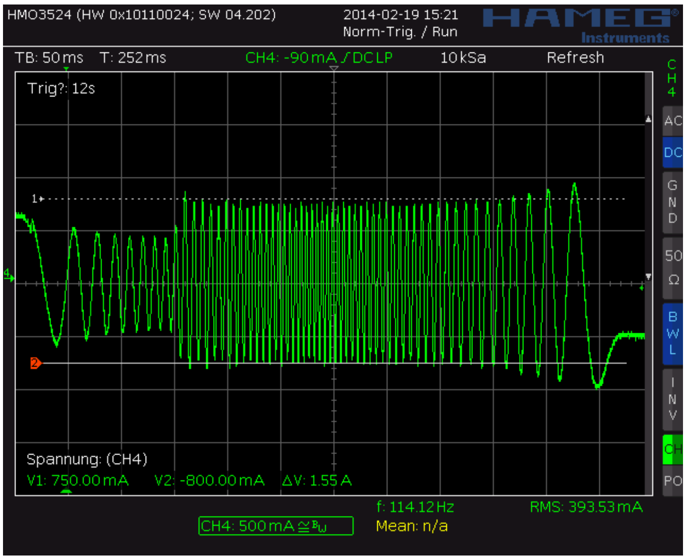
6.3 StealthChop Current Regulator

In StealthChop voltage PWM mode, the autoscaling function (*pwm_autoscale* = 1, *pwm_auto_grad* = 1) regulates the motor current to the desired current setting. Automatic scaling is used as part of the automatic tuning process (AT), and for subsequent tracking of changes within the motor parameters. The driver measures the motor current during the chopper on time and uses a proportional regulator to regulate *PWM_SCALE_AUTO* in order match the motor current to the target current. *PWM_REG* is the proportionality coefficient for this regulator. Basically, the proportionality coefficient should be as small as possible in order to get a stable and soft regulation behavior, but it must be large enough to allow the driver to quickly react to changes caused by variation of the motor target current (e.g. change of VREF). During initial tuning step AT#2, *PWM_REG* also compensates for the change of motor velocity. Therefore, a high acceleration during AT#2 will require a higher setting of *PWM_REG*. With careful selection of homing velocity and acceleration, a minimum setting of the regulation gradient often is sufficient (*PWM_REG*=1). *PWM_REG* setting should be optimized for the fastest required acceleration and deceleration ramp (compare Figure 6.3 and Figure 6.4). The quality of the setting *PWM_REG* in phase AT#2 and the finished automatic tuning procedure (or non-automatic settings for *PWM_OFS* and *PWM_GRAD*) can be examined when monitoring motor current during an acceleration phase Figure 6.5.



HMO3524 (HW 0x10110024; SW 04.202)
2014-02-19 15:19
Norm-Trig. / Run
TB: 50 ms T: 255 ms CH4: -90 mA ⁄ DC LP 10 kSa Refresh
Trig?: 22s
Spannung: (CH4)
V1: 750.00 mA V2: -800.00 mA ΔV: 1.55 A
f: 114.06 Hz RMS: 465.82 mA
CH4: 500 mA Mean: n/a

Figure 6.3 Scope shot: good setting for PWM_REG



HMO3524 (HW 0x10110024; SW 04.202)
2014-02-19 15:21
Norm-Trig. / Run
TB: 50 ms T: 252 ms CH4: -90 mA ⁄ DC LP 10 kSa Refresh
Trig?: 12s
Spannung: (CH4)
V1: 750.00 mA V2: -800.00 mA ΔV: 1.55 A
f: 114.12 Hz RMS: 393.53 mA
CH4: 500 mA Mean: n/a

Figure 6.4 Scope shot: too small setting for PWM_REG during AT#2

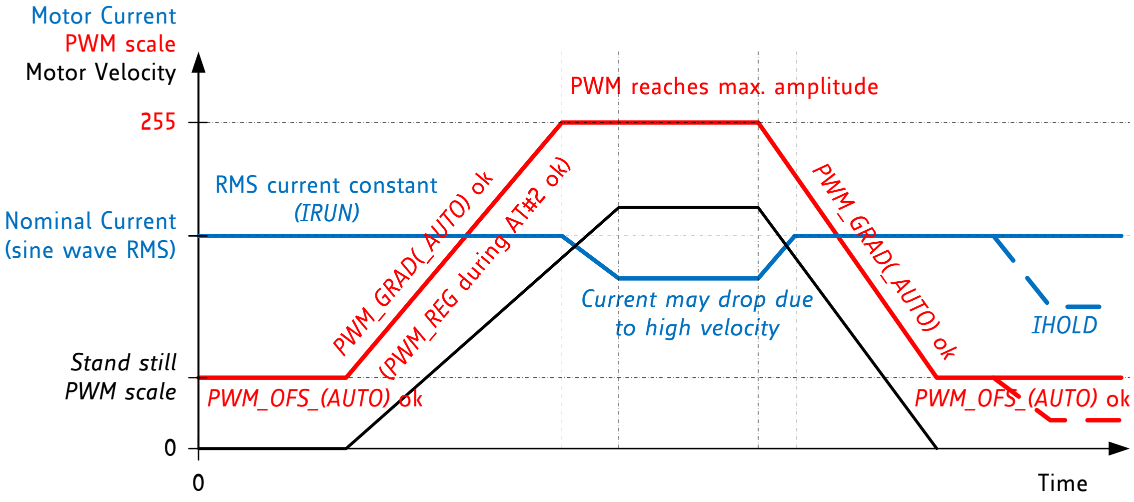


Figure 6.5 Successfully determined PWM_GRAD(_AUTO) and PWM_OFS(_AUTO)

Quick Start
For a quick start, see the Quick Configuration Guide in chapter 15.

6.3.1 Lower Current Limit

The StealthChop current regulator imposes a lower limit for motor current regulation. As the coil current can be measured in the shunt resistor during chopper on phase only, a minimum chopper duty cycle allowing coil current regulation is given by the blank time as set by *TBL* and by the chopper frequency setting. Therefore, the motor specific minimum coil current in StealthChop autoscaling mode rises with the supply voltage and with the chopper frequency. A lower blanking time allows a lower current limit. It is important for the correct determination of *PWM_OFS_AUTO*, that in AT#1 the run current set by the sense resistor and *IRUN* is well within the regulation range. Lower currents (e.g. for standstill power down) are automatically realized based on *PWM_OFS_AUTO* and *PWM_GRAD_AUTO* respectively based on *PWM_OFS* and *PWM_GRAD* with non-automatic current scaling. The freewheeling option allows going to zero motor current.

Lower motor coil current limit for StealthChop2 automatic tuning:

$$I_{Lower\ Limit} = t_{BLANK} * f_{PWM} * \frac{V_M}{R_{COIL}}$$

With V_M the motor supply voltage and R_{COIL} the motor coil resistance.
$I_{Lower\ Limit}$ can be treated as a thumb value for the minimum nominal *IRUN* motor current setting.

EXAMPLE:

A motor has a coil resistance of 5Ω, the supply voltage is 8.4V. With *TBL*=%01 and *PWM_FREQ*=%00, t_{BLANK} is 24 clock cycles, f_{PWM} is 2/(1024 clock cycles):

$$I_{Lower\ Limit} = 24\ t_{CLK} * \frac{2}{1024\ t_{CLK}} * \frac{8.4V}{5\Omega} = \frac{24}{512} * \frac{8.4V}{5\Omega} = 78mA$$

This means, the motor target current for automatic tuning must be 78mA or more, taking into account all relevant settings.

Attention
For automatic tuning, a lower coil current limit applies. The motor current in automatic tuning phase AT#1 must exceed this lower limit. $I_{LOWER\ LIMIT}$ can be calculated or measured using a current probe. Setting the motor run-current or hold-current below the lower current limit during operation by modifying *IRUN* and *IHOLD* is possible after successful automatic tuning.

The lower current limit also limits the capability of the driver to respond to changes of VREF.

6.4 Velocity Based Scaling

UART

Velocity based scaling scales the StealthChop amplitude based on the time between each two steps, i.e. based on *TSTEP*, measured in clock cycles. This concept basically does not require a current measurement, because no regulation loop is necessary. A pure velocity-based scaling is available via UART programming, only, when setting *pwm_autoscale* = 0. The basic idea is to have a linear approximation of the voltage required to drive the target current into the motor. The stepper motor has a certain coil resistance and thus needs a certain voltage amplitude to yield a target current based on the basic formula I=U/R. With R being the coil resistance, U the supply voltage scaled by the PWM value, the current I results. The initial value for PWM_AMPL can be calculated:

$$PWM\_AMPL = \frac{374 * R_{COIL} * I_{COIL}}{V_M}$$

With V_M the motor supply voltage and I_{COIL} the target RMS current

The effective PWM voltage U_{PWM} (1/SQRT(2) x peak value) results considering the 8 bit resolution and 248 sine wave peak for the actual PWM amplitude shown as *PWM_SCALE*:

$$U_{PWM} = V_M * \frac{PWM\_SCALE}{256} * \frac{248}{256} * \frac{1}{\sqrt{2}} = V_M * \frac{PWM\_SCALE}{374}$$

With rising motor velocity, the motor generates an increasing back EMF voltage. The back EMF voltage is proportional to the motor velocity. It reduces the PWM voltage effective at the coil resistance and thus current decreases. The TMC2300 provides a second velocity dependent factor (*PWM_GRAD*) to compensate for this. The overall effective PWM amplitude (*PWM_SCALE_SUM*) in this mode automatically is calculated in dependence of the microstep frequency as:

$$PWM\_SCALE\_SUM = PWM\_OFS + PWM\_GRAD * 256 * \frac{f_{STEP}}{f_{CLK}}$$

With f_{STEP} being the microstep frequency for 256 microstep resolution equivalent
and f_{CLK} the clock frequency supplied to the driver or the actual internal frequency

As a first approximation, the back EMF subtracts from the supply voltage and thus the effective current amplitude decreases. This way, a first approximation for *PWM_GRAD* setting can be calculated:

$$PWM\_GRAD = C_{BEMF}\left[\frac{V}{\frac{rad}{s}}\right] * 2\pi * \frac{f_{CLK} * 1.46}{V_M * MSPR}$$

C_{BEMF} is the back EMF constant of the motor in Volts per radian/second (See chapter 7).

MSPR is the number of microsteps per rotation, e.g. 51200 = 256µsteps multiplied by 200 fullsteps for a 1.8° motor.

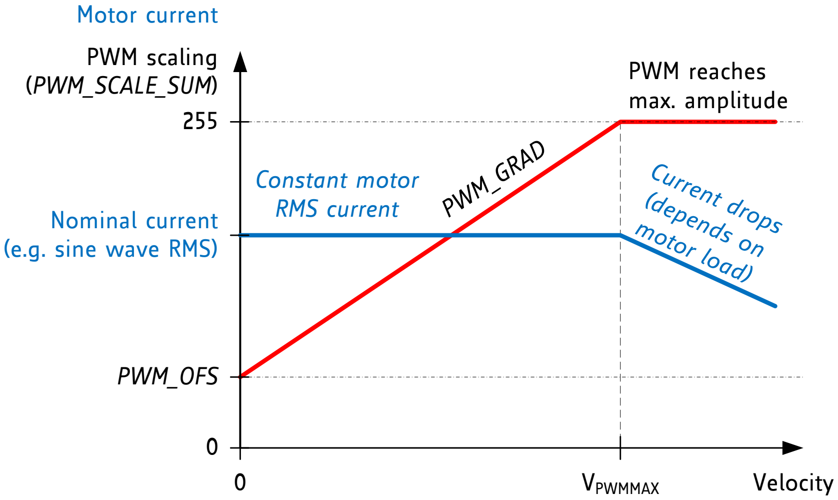


Figure 6.6 Velocity based PWM scaling (pwm_autoscale=0)

Hint
The values for *PWM_OFS* and *PWM_GRAD* can easily be optimized by tracing the motor current with a current probe on the oscilloscope. Alternatively, automatic tuning determines these values and they can be read out from *PWM_OFS_AUTO* and *PWM_GRAD_AUTO*.

Hint
Start the motor from standstill when switching on StealthChop the first time and keep it stopped for at least 128 chopper periods to allow StealthChop to do initial standstill current control.

6.5 Flags in StealthChop

UART

As StealthChop uses voltage mode driving, status flags based on current measurement respond slower, respectively the driver reacts delayed to sudden changes of back EMF, like on a motor stall.

Attention
A motor stall, or abrupt stop of the motion during operation in StealthChop can trigger an overcurrent condition. Depending on the previous motor velocity, and on the coil resistance of the motor, it significantly increases motor current for a time of several 10ms. With low velocities, where the back EMF is just a fraction of the supply voltage, there is no danger of triggering the short detection. When homing using StallGuard4 to stop the motor upon stall, this is basically avoided.

6.5.1 Open Load Flags

In StealthChop mode, OLA and OLB show if the current regulation sees that the nominal current can be reached on both coils.

- A flickering OLA or OLB can result from asymmetries in the sense resistors or in the motor coils.
- An interrupted motor coil leads to a continuously active open load flag for the coil.
- One or both flags are active, if the current regulation did not succeed in scaling up to the full target current within the last few fullsteps (because no motor is attached or a high velocity exceeds the PWM limit).

With StealthChop, *PWM_SCALE_SUM* can be checked to detect the correct coil resistance.

6.5.2 PWM_SCALE_SUM Informs about the Motor State

Information about the motor state is available with automatic scaling by reading out *PWM_SCALE_SUM*. As this parameter reflects the actual voltage required to drive the target current into the motor, it depends on several factors: motor load, coil resistance, supply voltage, and current setting. Therefore, an evaluation of the *PWM_SCALE_SUM* value allows checking the motor operation point. When reaching the limit (255), the current regulator cannot sustain the full motor current, e.g. due to a drop in supply voltage.

6.6 Freewheeling and Passive Braking

UART

StealthChop provides different options for motor standstill. These options can be enabled by setting the standstill current *IHOLD* to zero and choosing the desired option using the *FREEWHEEL* setting. The desired option becomes enabled after a time period specified by *TPOWERDOWN* and *IHOLD_DELAY*. Current regulation becomes frozen once the motor target current is at zero current in order to ensure a quick startup. With the freewheeling options, both freewheeling and passive braking can be realized. Passive braking is an effective eddy current motor braking, which consumes a minimum of energy, because no active current is driven into the coils. However, passive braking will allow slow turning of the motor when a continuous torque is applied.

Hint
Operate the motor within your application when exploring StealthChop. Motor performance often is better with a mechanical load, because it prevents the motor from stalling due mechanical oscillations which can occur without load.

Parameters Related to StealthChop

Parameter	Description	Setting	Comment
PWM_LIM	Limiting value for limiting the current jerk when switching on StealthChop following a disable condition. Reduce the value to yield a lower current peak.	0 … 15	Upper four bits of 8 bit amplitude limit (Default=12)
pwm_autoscale	Enable automatic current scaling using current measurement or use forward controlled velocity based mode.	0	Forward controlled mode
		1	Automatic scaling with current regulator
pwm_autograd	Enable automatic tuning of *PWM_GRAD_AUTO*	0	disable, use *PWM_GRAD* from register instead
		1	enable
PWM_FREQ	PWM frequency selection. Use the lowest setting giving good results. The frequency measured at each of the chopper outputs is half of the effective chopper frequency f_{PWM}.	0	$f_{PWM}=2/1024\ f_{CLK}$
		1	$f_{PWM}=2/683\ f_{CLK}$
		2	$f_{PWM}=2/512\ f_{CLK}$
		3	$f_{PWM}=2/410\ f_{CLK}$
PWM_REG	User defined PWM amplitude (gradient) for velocity-based scaling or regulation loop gradient when *pwm_autoscale*=1.	1 … 15	Results in 0.5 to 7.5 steps for *PWM_SCALE_AUTO* regulator per fullstep
PWM_OFS	User defined PWM amplitude (offset) for velocity-based scaling and initialization value for automatic tuning of *PWM_OFFS_AUTO*.	0 … 255	*PWM_OFS*=0 disables linear current scaling based on current setting
PWM_GRAD	User defined PWM amplitude (gradient) for velocity-based scaling and initialization value for automatic tuning of *PWM_GRAD_AUTO*.	0 … 255	
FREEWHEEL	Stand still option when motor current setting is zero (*I_HOLD*=0). The freewheeling option makes the motor easy movable, while both coil short options realize a passive brake.	0	Normal operation
		1	Freewheeling
		2	Coil short via LS drivers
		3	Coil short cia HS drivers
PWM_SCALE_AUTO	Read back of the actual StealthChop voltage PWM scaling correction as determined by the current regulator.	-255 … 255	(read only) Scaling value becomes frozen when driver is disabled
PWM_SCALE_AUTO *PWM_OFS_AUTO*	Allow monitoring of the automatic tuning and determination of initial values for *PWM_OFS* and *PWM_GRAD*.	0 … 255	(read only)
enabledrv	General enable for the motor driver	0	Driver off, all outputs hi-Z
		1	Driver enabled
TBL	Comparator *blank time*. This time needs to safely cover the switching event and the duration of the ringing on the sense resistor. Choose a setting of 1 or 2 for typical applications. For higher capacitive loads, 3 may be required. Lower settings allow StealthChop to regulate down to lower coil current values.	0	16 t_{CLK}
		1	24 t_{CLK}
		2	32 t_{CLK}
		3	40 t_{CLK}

7 Fitting the Motor

Especially for low voltage operation, the motor should be carefully selected to give a good fit to the application's mechanics, as well as available supply voltage and current. Therefore, it is important to understand the supply voltage requirement for a given motor. Both, the generation of a certain torque, and the ability to provide this torque at a desired velocity, require a motor specific voltage. These two components add up.

Main relevant parameters for a stepper motor:

Nominal (RMS) coil current	$I_{COILNOM}$ [A]	
Nominal coil resistance	R_{COIL} [Ω]	
Rated coil voltage	$U_N = R_{COIL} * I_{COILNOM}$ [V]	(sometimes specified instead of $I_{COILNOM}$)
Holding torque at $I_{COILNOM}$	HoldingTorque [Nm]	

The specified motor torque is reached with the RMS I_{COIL} current in both motor coils, in order to build up the required magnetic field strength. A lower current will basically proportionally generate a lower torque, e.g. 70% of torque at 70% current. Even a reduction to 70% saves a lot of energy, because power dissipation goes with the square of the current. Thus, a motor with more reserves can offer better efficiency!

With this, calculate the required supply voltage U_{BAT} for motor stand still and slow motion, taking into account the driver's power stage resistance plus 0.3V loss in the sense resistor:

$$U_{BAT} \;=\; (R_{COIL} + 2 * 0.17\Omega) * I_{COIL} * \sqrt{2} + 0.3V$$

I_{COIL} is the RMS motor current which gives the desired torque.

For higher velocity operation (more than a few electrical rotations per second), the motor specific back EMF constant C_{BEMF} should be additionally taken into account (see below explanation). With this, the lowest feasible supply voltage for a given motor and a maximum velocity [RPM] calculates to:

$$U_{BAT} \;=\; 0.3V + ((R_{COIL} + 2 * 0.17\Omega) * I_{COIL} + \frac{HoldingTorque[Nm]}{2 * I_{COILNOM}} * \frac{2\pi * Velocity[RPM]}{60}) * \sqrt{2}$$

> *Adapt your motor to battery operation*
>
> With most motor suppliers you have the chance to adapt the coil winding. This allows to trade in a lower motor voltage for battery operation versus higher motor current. E.g. a motor with a short, thick coil wire can work at a lower voltage, than the same motor with a long, thin coil wire, but it needs a higher current for the same torque.

UNDERSTANDING THE BACK EMF CONSTANT OF A MOTOR

The back EMF constant is the voltage a motor generates when turned with a certain velocity. Often motor datasheets do not specify this value, as it can be deducted from motor torque and coil current rating. Within SI units, the back EMF constant C_{BEMF} has the same numeric value as the torque constant. For example, a motor with a torque constant of 1 Nm/A would have a C_{BEMF} of 1V/rad/s. Turning such a motor with 1 rps (1 rps = 1 revolution per second = 6.28 rad/s) generates a back EMF voltage of 6.28V. Thus, the back EMF constant can be calculated as:

$$C_{BEMF}\left[\frac{V}{rad/s}\right] = \frac{HoldingTorque[Nm]}{2 * I_{COILNOM}[A]}$$

$I_{COILNOM}$ is the motor's rated phase current for the specified holding torque

HoldingTorque is the motor specific holding torque, i.e. the torque reached at $I_{COILNOM}$ on both coils. The torque unit is [Nm] where 1Nm = 100Ncm = 1000mNm.

The BEMF voltage is valid as RMS voltage per coil, thus the nominal current has a factor of 2 in this formula.

8 Selecting Sense Resistors

Set the desired maximum motor current by selecting an appropriate value for the sense resistor. The following table shows the RMS current values which can be reached using standard resistors and motor types fitting without additional motor current scaling. Additional 15mΩ PCB resistance are included in the calculation.

CHOICE OF R_{SENSE} AND RESULTING MAX. MOTOR CURRENT

R_{SENSE} [Ω]	**RMS current [A]** *IRUN*=31	**Fitting motor type at max. current setting** (examples)
1.50	0.15	150mA motor
1.20	0.18	200mA motor
1.00	0.22	300mA motor
0.82	0.27	
0.75	0.29	
0.68	0.32	400mA motor
0.50	0.42	500mA motor
470m	0.45	
390m	0.53	600mA motor
330m	0.61	700mA motor
270m	0.73	800mA motor
220m	0.87	1A motor *)
180m	1.02	
150m	1.18	1.2A motor *)

*) At high currents, duty cycle restriction for motion might apply, due to heat up of IC and board.

Sense resistors should be carefully selected. The full motor current flows through the sense resistors. Due to chopper operation the sense resistors see pulsed current from the MOSFET bridges. Therefore, a low-inductance type such as film or composition resistors is required to prevent voltage spikes causing ringing on the sense voltage inputs leading to unstable measurement results. Also, a low-inductance, low-resistance PCB layout is essential. Any common GND path for the two sense resistors must be avoided, because this would lead to coupling between the two current sense signals. A massive ground plane is best. Please also refer to layout considerations in chapter 20.

The sense resistor needs to be able to conduct the peak motor coil current in motor standstill conditions, unless standby power is reduced. Under normal conditions, the sense resistor conducts less than the coil RMS current, because no current flows through the sense resistor during the slow decay phases. A 0.25W type is sufficient for most applications up to 800mA RMS.

Attention

Be sure to use a symmetrical sense resistor layout and short and straight sense resistor traces of identical length. Well matching sense resistors ensure best performance.

A compact layout with massive ground plane is best to avoid parasitic resistance effects.

9 Motor Current Control

The basic motor current is set by the value of the sense resistors. Several possibilities allow scaling down motor current, e.g. to adapt for different motors, or to reduce motor current in standstill or low load situations.

METHODS FOR SCALING MOTOR CURRENT			
Method	**Parameters**	**Range**	**Primary Use**
Pin EN	Disable / enable driver stage	1: Motor enable 0: Motor disable	- Disable motor to allow freewheeling or power saving
Pin PDN_UART	Disable / enable standstill current reduction to *IHOLD*	0: Standstill current reduction enabled. 1: Disable	- Enable current reduction to reduce heat up in stand still
UART interface	*IHOLD_IRUN* *TPOWERDOWN*	*IRUN*, *IHOLD*: 1/32 to 32/32 of full scale current.	- Fine programming of run and hold (stand still) current - Change *IRUN* for situation specific motor current (range 8 to 31)

Select the sense resistor to deliver enough current for the motor at full current scale. This is the default current scaling (*IRUN* = 31).

STANDALONE MODE RMS RUN CURRENT CALCULATION:

$$I_{RMS} = \frac{325mV}{R_{SENSE} + 30m\Omega} * \frac{1}{\sqrt{2}}$$

IRUN and *IHOLD* allow for scaling of the actual current scale (*CS*) from 9/32 (*IRUN*, minimum value), resp. 1/32 to 32/32 when using UART interface, or via automatic standstill current reduction:

RMS CURRENT CALCULATION WITH UART CONTROL OPTIONS OR HOLD CURRENT SETTING:

$$I_{RMS} = \frac{CS + 1}{32} * \frac{V_{FS}}{R_{SENSE} + 30m\Omega} * \frac{1}{\sqrt{2}}$$

CS is the current scale setting as set by the *IHOLD* and *IRUN*.
V_{FS} is the full-scale voltage (please refer to electrical characteristics, V_{SRT}). Default is 325mV.

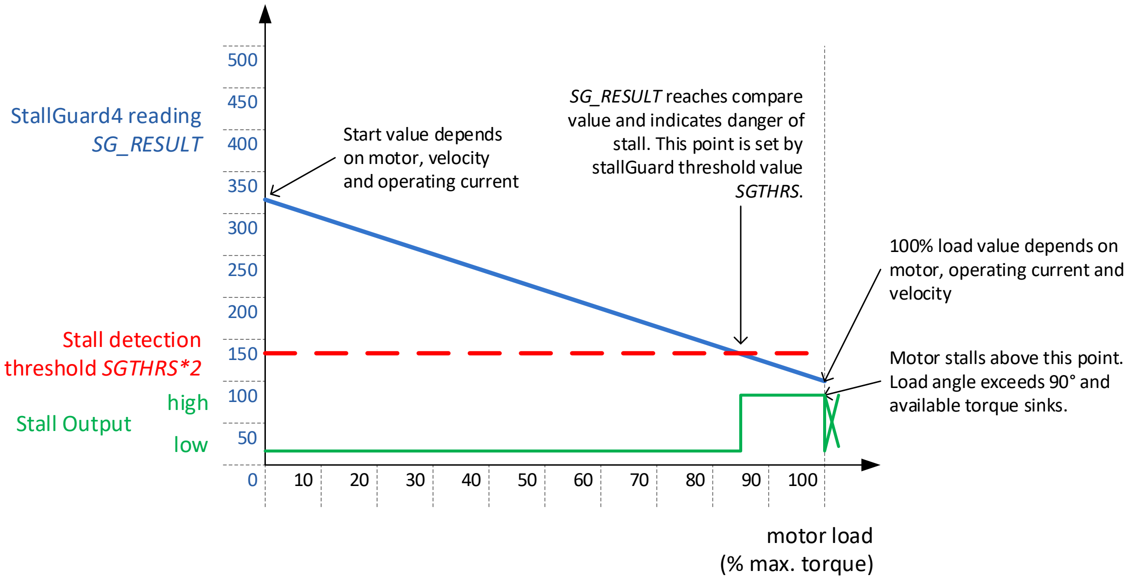
Hint
For best precision of current setting, measure and fine tune the current in the application.

PARAMETERS FOR MOTOR CURRENT CONTROL

Parameter	**Description**	**Setting**	**Comment**
IRUN	Current scale when motor is running. Scales coil current values as taken from the internal sine wave table. For proper operation, do not set values lower than 8. Optimum range is 16 to 32.	8 … 31	scaling factor 0: 1/32, …, 8: 9/32 … 31: 32/32 *IRUN* is full scale (setting 31) in standalone mode.
IHOLD	Identical to *IRUN*, but for motor in stand still.	0 … 31	
IHOLD DELAY	Allows smooth current reduction from run current to hold current. *IHOLDDELAY* controls the number of clock cycles for motor power down after *TPOWERDOWN* in increments of 2^18 clocks: 0=instant power down, 1..15: Current reduction delay per current step in multiple of 2^18 clocks. *Example:* When using *IRUN*=31 and *IHOLD*=16, 15 current steps are required for hold current reduction. A *IHOLDDELAY* setting of 4 thus results in a power down time of 4*15*2^18 clock cycles, i.e. roughly one second at 16MHz clock frequency.	0	instant *IHOLD*
		1 … 15	$1*2^{18}$ … $15*2^{18}$ clocks per current decrement
TPOWER DOWN	Sets the delay time from stand still (*stst*) detection to motor current power down. Time range is about 0 to 5.6 seconds.	0 … 255	0…((2^8)-1) * 2^18 t_{CLK} A minimum setting of 2 is required to allow automatic tuning of *PWM_OFFS_AUTO*

10 StallGuard4 Load Measurement

StallGuard4 provides an accurate measurement of the load on the motor. It is developed for operation in conjunction with StealthChop. StallGuard can be used for stall detection as well as other uses at loads below those which stall the motor, such as CoolStep load-adaptive current reduction. The StallGuard4 measurement value changes linearly over a wide range of load, velocity, and current settings, as shown in Figure 10.1. When approaching maximum motor load, the value goes down to a motor-specific lower value. This corresponds to a load angle of 90° between the magnetic field of the coils and magnets in the rotor. This also is the most energy-efficient point of operation for the motor.



Start value depends on motor, velocity and operating current

SG_RESULT reaches compare value and indicates danger of stall. This point is set by stallGuard threshold value *SGTHRS*.

100% load value depends on motor, operating current and velocity

Motor stalls above this point. Load angle exceeds 90° and available torque sinks.

Figure 10.1 Function principle of StallGuard4

Parameter	Description	Setting	Comment
SGTHRS	This value controls the StallGuard4 threshold level for stall detection. It compensates for motor specific characteristics and controls sensitivity. A higher value gives a higher sensitivity. A higher value makes StallGuard4 more sensitive and requires less torque to indicate a stall.	0… 255	The double of this value is compared to *SG_RESULT.* The stall output becomes active if *SG_RESULT* fall below this value.
Status word	**Description**	**Range**	**Comment**
SG_RESULT	This is the *StallGuard4 result.* A higher reading indicates less mechanical load. A lower reading indicates a higher load and thus a higher load angle.	0… 510	Low value: highest load High value: high load

In order to use StallGuard4, check the sensitivity of the motor at border conditions.

10.1 Tuning StallGuard4

The StallGuard4 value *SG_RESULT* is affected by motor-specific characteristics and application-specific demands on load, coil current, and velocity. Therefore, the easiest way to tune the StallGuard4 threshold *SGTHRS* for a specific motor type and operating conditions is interactive tuning in the actual application.

INITIAL PROCEDURE FOR TUNING STALLGUARD *SGTHRS*

1. Operate the motor at the normal operation velocity for your application and monitor *SG_RESULT*.
2. Apply slowly increasing mechanical load to the motor. Check the lowest value of *SG_RESULT* before the motor stalls. Use this value as starting value for *SGTHRS* (apply half of the value).
3. Now monitor the StallGuard output signal via DIAG output (configure properly, also set *TCOOLTHRS*) and stop the motor when a pulse is seen on the respective output. Make sure, that the motor is safely stopped whenever it is stalled. Increase *SGTHRS* if the motor becomes stopped before a stall occurs.
4. The optimum setting is reached when a stall is safely detected and leads to a pulse at DIAG in the moment where the stall occurs. *SGTHRS* in most cases can be tuned for a certain motion velocity or a velocity range. Make sure, that the setting works reliable in a certain range (e.g. 75% to 150% of desired velocity) and also under extreme motor conditions (lowest and highest applicable temperature).

DIAG is pulsed by StallGuard, when *SG_RESULT* falls below *SGTHRS*. It is only enabled in StealthChop mode, and when *TCOOLTHRS* ≥ *TSTEP*
The external motion controller should react to a single pulse by stopping the motor if desired. Set *TCOOLTHRS* to match the lower velocity threshold where StallGuard delivers a good result.

SG_RESULT measurement has a high resolution, and there are a few ways to enhance its accuracy, as described in the following sections.

10.2 StallGuard4 Update Rate

The StallGuard4 measurement value *SG_RESULT* is updated with each full step of the motor. This is enough to safely detect a stall, because a stall always means the loss of four full steps.

10.3 Detecting a Motor Stall

To safely detect a motor stall, the stall threshold must be determined using a specific *SGTHRS* setting and a specific motor velocity or velocity range. Further, the motor current setting has a certain influence and should not be modified, once optimum values are determined. Therefore, the maximum load needs to be determined that the motor can drive without stalling. At the same time, monitor *SG_RESULT* at this load. The stall threshold should be a value safely within the operating limits, to allow for parameter stray. More refined evaluation may also react to a change of *SG_RESULT* rather than comparing to a fixed threshold. This will rule out certain effects which influence the absolute value.

10.4 Limits of StallGuard4 Operation

StallGuard4 does not operate reliably at extreme motor velocities: Very low motor velocities (for many motors, less than one revolution per second) generate a low back EMF and make the measurement unstable and dependent on environment conditions (temperature, etc.). Other conditions will also lead to a poor response of the measurement value *SG_RESULT* to the motor load. Very high motor velocities, in which the full sinusoidal current is not driven into the motor coils also leads to poor response. These velocities are typically characterized by the motor back EMF exceeding the supply voltage.

11 CoolStep Operation

CoolStep is an automatic smart energy optimization for stepper motors based on the motor mechanical load, making them "green".

11.1 User Benefits

Energy efficiency – consumption decreased up to 90%
Motor generates less heat – improved mechanical precision
Less cooling infrastructure – for motor and driver
Cheaper motor – does the job!

CoolStep allows substantial energy savings, especially for motors which see varying loads or operate at a high duty cycle. Because a stepper motor application needs to work with a torque reserve of 30% to 50%, even a constant-load application allows significant energy savings because CoolStep automatically enables torque reserve when required. Reducing power consumption keeps the system cooler, increases motor life, and allows reducing cost in the power supply and cooling components.

Reducing motor current by half results in reducing power by a factor of four.

11.2 Setting up for CoolStep

CoolStep is controlled by several parameters, but two are critical for understanding how it works:

Parameter	Description	Range	Comment
SEMIN	4-bit unsigned integer that sets a *lower threshold*. If *SG_RESULT* goes below this threshold, CoolStep increases the current to both coils. The 4-bit *SEMIN* value is scaled by 32 to cover the lower half of the range of the 10-bit *SG* value. (The name of this parameter is derived from SmartEnergy, which is an earlier name for CoolStep.)	0	disable CoolStep
		1…15	threshold is *SEMIN**32 Once *SGTHRS* has been determined, use 1/16**SGTHRS*+1 as a starting point for *SEMIN*.
SEMAX	4-bit unsigned integer that controls an *upper threshold*. If SG is sampled equal to or above this threshold enough times, CoolStep decreases the current to both coils. The upper threshold is (*SEMIN* + *SEMAX* + 1)*32.	0…15	threshold is (*SEMIN*+*SEMAX*+1)*32 0 to 2 recommended

Figure 11.1 shows the operating regions of CoolStep:

- The black line represents the *SG_RESULT* measurement value.
- The blue line represents the mechanical load applied to the motor.
- The red line represents the current into the motor coils.

When the load increases, *SG_RESULT* falls below *SEMIN*, and CoolStep increases the current. When the load decreases, *SG_RESULT* rises above (*SEMIN* + *SEMAX* + 1) * 32, and the current is reduced.

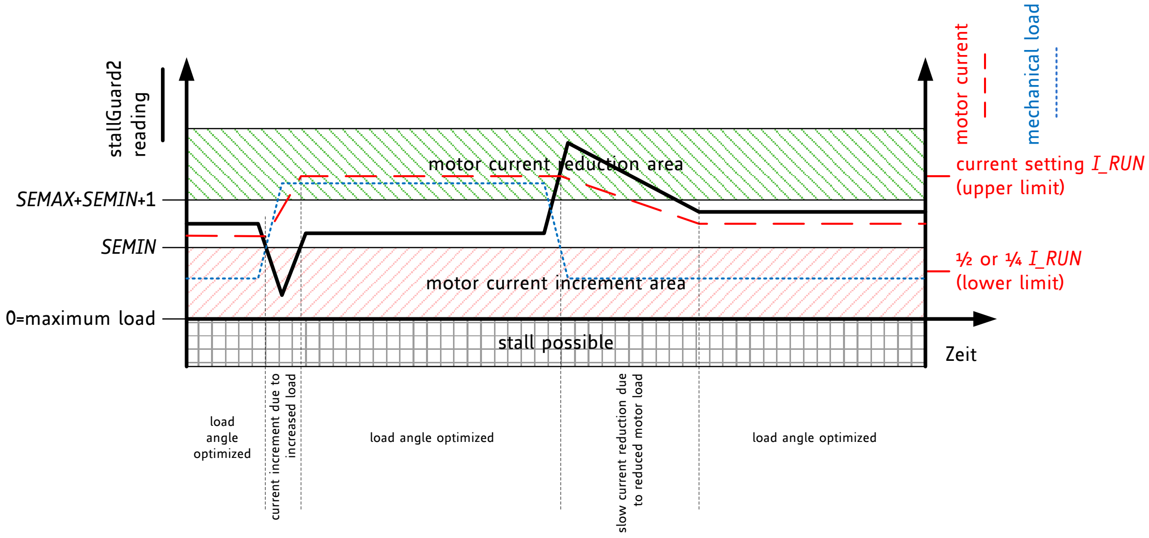


Figure 11.1 CoolStep adapts motor current to the load

Five more parameters control CoolStep and one status value is returned:

Parameter	**Description**	**Range**	**Comment**
SEUP	Sets the *current increment step*. The current becomes incremented for each measured StallGuard2 value below the lower threshold.	0…3	step width is 1, 2, 4, 8
SEDN	Sets the number of StallGuard2 readings above the upper threshold necessary for each *current decrement* of the motor current.	0…3	number of StallGuard2 measurements per decrement: 32, 8, 2, 1
SEIMIN	Sets the *lower motor current limit* for CoolStep operation by scaling the *IRUN* current setting. Operate well above the minimum motor current as determined for StealthChop current regulation.	0 1	0: 1/2 of *IRUN* 1: 1/4 of *IRUN* *IRUN* ≥ 16 recommended
TCOOLTHRS	Lower velocity threshold for switching on CoolStep and stall output. Below this velocity CoolStep becomes disabled (not used in STEP/DIR mode). Adapt to the lower limit of the velocity range where StallGuard2 gives a stable result.	1… 2^20-1	Specifies lower CoolStep velocity by comparing the threshold value to *TSTEP*
Status word	**Description**	**Range**	**Comment**
CSACTUAL	This status value provides the *actual motor current scale* as controlled by CoolStep. The value goes up to the *IRUN* value and down to the portion of *IRUN* as specified by *SEIMIN*.	0…31	1/32, 2/32, … 32/32

11.3 Tuning CoolStep

CoolStep uses *SG_RESULT* to operate the motor near the optimum load angle of +90°. The basic setting to be tuned is *SEMIN*. Set *SEMIN* to a value which safely activates CoolStep current increment before the motor stalls. In case *SGTHRS* has been tuned before, a lower starting value is

SEMIN = 1+*SGTHRS*/16.

The current increment speed is specified in *SEUP*, and the current decrement speed is specified in *SEDN*. They can be tuned separately because they are triggered by different events that may need different responses. The encodings for these parameters allow the coil currents to be increased much more quickly than decreased, because crossing the lower threshold is a more serious event that may require a faster response. If the response is too slow, the motor may stall. In contrast, a slow response to crossing the upper threshold does not risk anything more serious than missing an opportunity to save power.

CoolStep operates between limits controlled by the current scale parameter *IRUN* and the *seimin* bit.

Attention
When CoolStep increases motor current, spurious detection of motor stall may occur. For best results, disable CoolStep during StallGuard based homing.
In case StallGuard is desired in combination with CoolStep, try increasing coolStep lower threshold *SEMIN* as required.

11.3.1 Response Time

For fast response to increasing motor load, use a high current increment step *SEUP*. If the motor load changes slowly, a lower current increment step can be used to avoid motor oscillations.

Hint
The most common and most beneficial use is to adapt CoolStep for operation at the typical system target operation velocity and to set the velocity thresholds according. As acceleration and decelerations normally shall be quick, they will require the full motor current, while they have only a small contribution to overall power consumption due to their short duration.

11.3.2 Low Velocity and Standby Operation

Because CoolStep is not able to measure the motor load in standstill and at very low RPM, a lower velocity threshold is provided for enabling CoolStep. It should be set to an application specific default value. Below this threshold the normal current setting via *IRUN* respectively *IHOLD* is valid.

12 STEP/DIR Interface

The STEP and DIR inputs provide a simple, standard interface compatible with many existing motion controllers. The microPlyer step pulse interpolator brings the smooth motor operation of high-resolution microstepping to applications originally designed for coarser stepping.

12.1 Timing

Figure 12.1 shows the timing parameters for the STEP and DIR signals, and the table below gives their specifications. Only rising edges are active. STEP and DIR are sampled and synchronized to the system clock. If the signal source is far from the chip, and especially if the signals are carried on cables, the signals should be filtered or differentially transmitted.

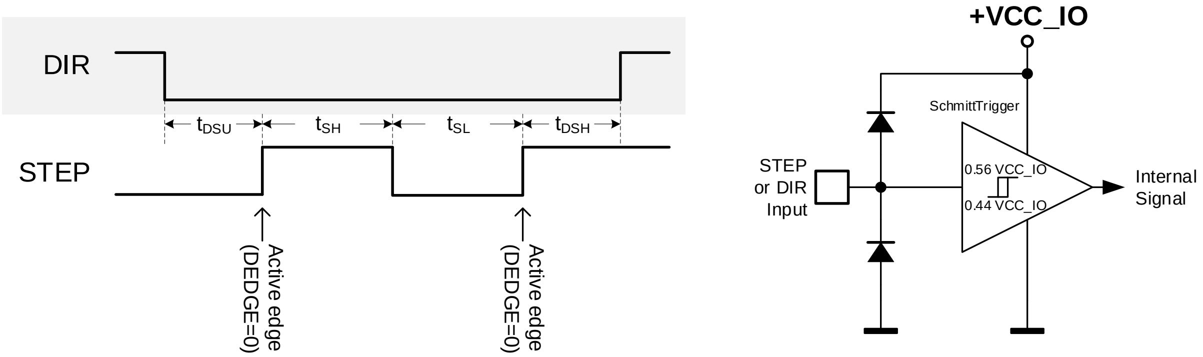


Figure 12.1 STEP and DIR timing, Input pin filter

STEP and DIR interface timing	**AC-Characteristics** (taking into account possible lowest internal clock generator frequency)					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
step frequency (at maximum microstep resolution)	f_{STEP}				4	MHz
fullstep frequency	f_{FS}				8	kHz
STEP input minimum low time	t_{SL}		120			ns
STEP input minimum high time	t_{SH}		120			ns
DIR to STEP setup time	t_{DSU}		20			ns
DIR after STEP hold time	t_{DSH}		20			ns

12.2 Changing Resolution

The TMC2300 includes an internal microstep table with 1024 sine wave entries to generate sinusoidal motor coil currents. These 1024 entries correspond to one electrical revolution or four fullsteps. The microstep resolution setting determines the step width taken within the table. Depending on the DIR input, the microstep counter is increased (DIR=0) or decreased (DIR=1) with each STEP pulse by the step width. The microstep resolution determines the increment respectively the decrement. At maximum resolution, the sequencer advances one step for each step pulse. At half resolution, it advances two steps. Increment is up to 256 steps for fullstepping. The sequencer has special provision to allow seamless switching between different microstep rates at any time. When switching to a lower microstep resolution, it calculates the nearest step within the target resolution and reads the current vector at that position. This behavior especially is important for low resolutions like fullstep and halfstep, because any failure in the step sequence would lead to asymmetrical run when comparing a motor running clockwise and counterclockwise.

EXAMPLES:

Fullstep:	Cycles through table positions: 128, 384, 640 and 896 (45°, 135°, 225° and 315° electrical position, both coils on at identical current). The coil current in each position corresponds to the RMS-Value (0.71 * amplitude). Step size is 256 (90° electrical)
Half step:	The first table position is 64 (22.5° electrical), Step size is 128 (45° steps)
Quarter step:	The first table position is 32 (90°/8=11.25° electrical), Step size is 64 (22.5° steps)

This way equidistant steps result and they are identical in both rotation directions. Some older drivers also use zero current (table entry 0, 0°) as well as full current (90°) within the step tables. This kind of stepping is avoided because it provides less torque and has a worse power dissipation in driver and motor.

Step position	table position	current coil A	current coil B
Half step 0	64	38.3%	92.4%
Full step 0	128	70.7%	70.7%
Half step 1	192	92.4%	38.3%
Half step 2	320	92.4%	-38.3%
Full step 1	384	70.7%	-70.7%
Half step 3	448	38.3%	-92.4%
Half step 4	576	-38.3%	-92.4%
Full step 2	640	-70.7%	-70.7%
Half step 5	704	-92.4%	-38.3%
Half step 6	832	-92.4%	38.3%
Full step 3	896	-70.7%	70.7%
Half step 7	960	-38.3%	92.4%

See chapter 3.2.1 for resolution settings available in stand-alone mode.

12.3 MicroPlyer Step Interpolator and Stand Still Detection

For each active edge on STEP, MicroPlyer produces microsteps at 256x resolution, as shown in Figure 12.2. It interpolates the time in between of two step impulses at the step input based on the last step interval. This way, from 2 microsteps (128 microstep to 256 microstep interpolation) up to 256 microsteps (full step input to 256 microsteps) are driven for a single step pulse.

The step rate for the interpolated 2 to 256 microsteps is determined by measuring the time interval of the previous step period and dividing it into up to 256 equal parts. The maximum time between two microsteps corresponds to 2^{20} (roughly one million system clock cycles), for an even distribution of 256 microsteps. At 12 MHz system clock frequency, this results in a minimum step input frequency of roughly 12 Hz for MicroPlyer operation. A lower step rate causes a standstill event to be detected. At that frequency, microsteps occur at a rate of (system clock frequency)/2^{16} ~ 256 Hz. When a stand still is detected, the driver automatically begins standby current reduction if selected by pin PDN.

Attention
MicroPlyer only works perfectly with a jitter-free STEP frequency.

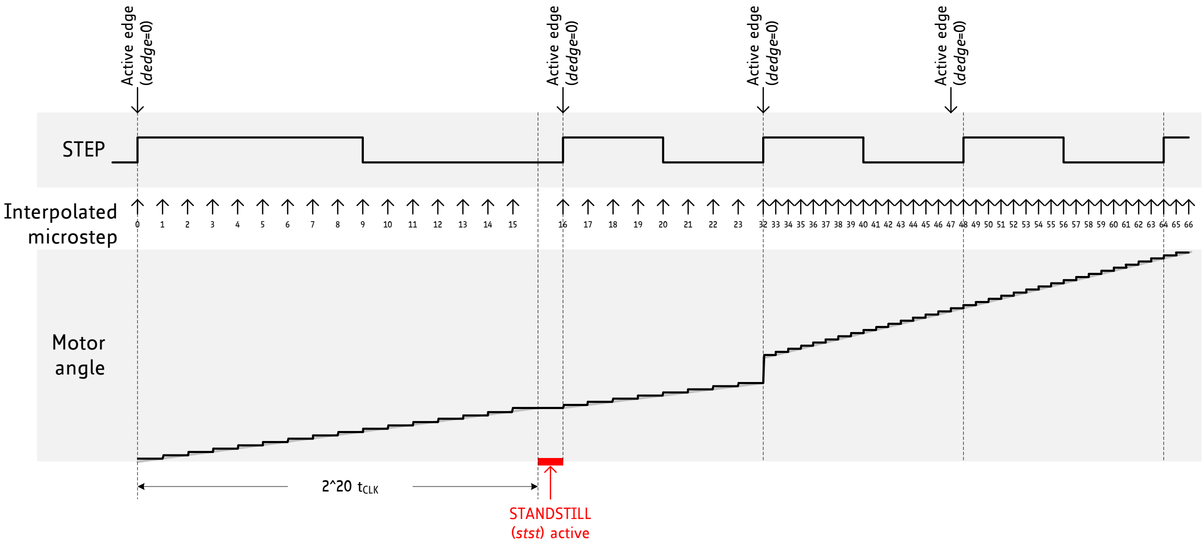


Figure 12.2 microPlyer microstep interpolation with rising STEP frequency (Example: 16 to 256)

In Figure 12.2, the first STEP cycle is long enough to set the *stst* bit standstill. Detection of standstill will enable the standby current reduction. This bit is cleared on the next STEP active edge. Then, the external STEP frequency increases. After one cycle at the higher rate microPlyer adapts the interpolated microstep rate to the higher frequency. During the last cycle at the slower rate, microPlyer did not generate all 16 microsteps, so there is a small jump in motor angle between the first and second cycles at the higher rate.

12.4 Index Signal

An active index output (enable *diag_index*) signals that the sine curve of motor coil A is at its positive zero transition. This correlates to the zero point of the microstep sequence. Usually, the cosine curve of coil B is at its maximum at the same time. Thus, the index signal is active once within each electrical period, and corresponds to a defined position of the motor within a sequence of four fullsteps. The index output this way allows the detection of a certain microstep pattern, and thus helps to detect a position with more precision than a stop switch can do.

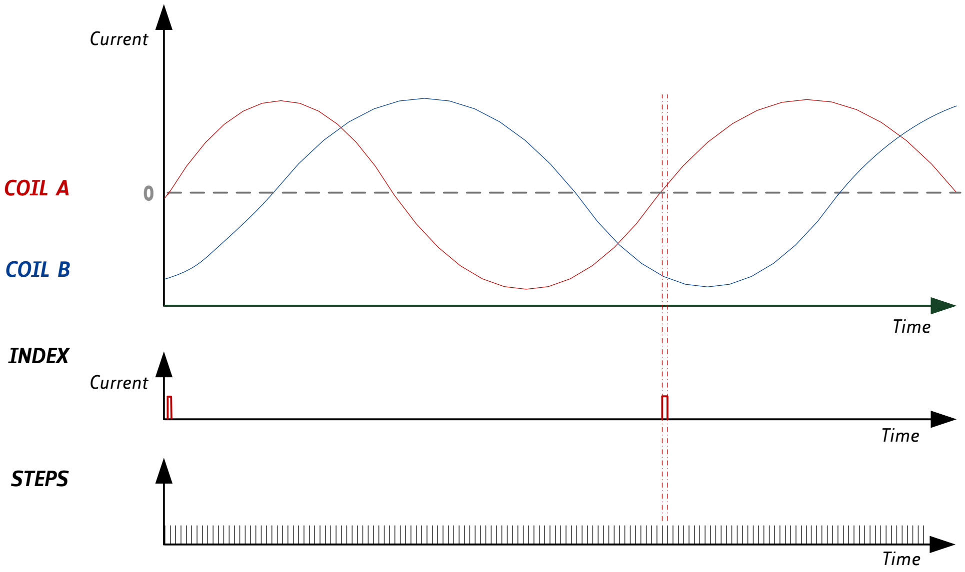


Figure 12.3 Index signal at positive zero transition of the coil A sine curve

Hint
The index output allows precise detection of the microstep position within one electrical wave, i.e. within a range of four fullsteps. With this, homing accuracy and reproducibility can be enhanced to microstep accuracy, even when using an inexpensive home switch.

13 Internal Step Pulse Generator

UART

The TMC2300 integrates a high-resolution step pulse generator, allowing motor motion via the UART interface. However, no velocity ramping is provided. Ramping is not required, if the target motion velocity is smaller than the start & stop frequency of the motor. For higher velocities, ramp up the frequency in small steps to accelerate the motor, and ramp down again to decelerate the motor. Figure 13.1 shows an example motion profile ramping up the motion velocity in discrete steps. Choose the ramp velocity steps considerably smaller than the maximum start velocity of the motor, because motor torque drops at higher velocity, and motor load at higher velocity typically increases.

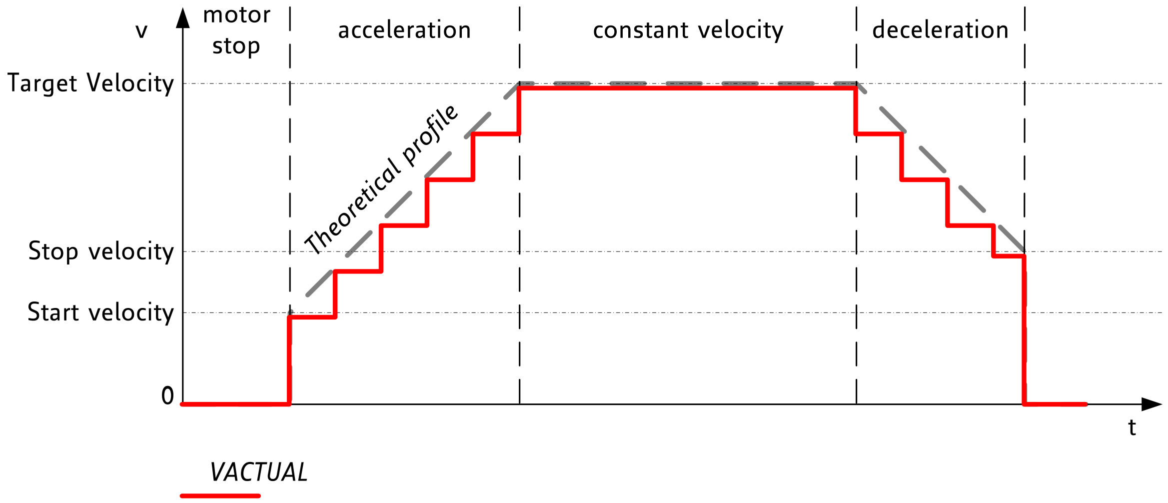


Figure 13.1 Software generated motion profile

Parameter vs. Units		
Parameter / Symbol	**Unit**	**calculation / description / comment**
f_{CLK}[Hz]	[Hz]	clock frequency of the TMC2300 in [Hz]
µstep velocity v[Hz]	µsteps / s	v[Hz] = *VACTUAL*[2300] * (f_{CLK}[Hz] / 2^24) With nominal oscillator: v[Hz] = *VACTUAL*[2300] * 0.715Hz
USC microstep count	counts	microstep resolution in number of microsteps (i.e. the number of microsteps between two fullsteps – normally 256)
rotations per second v[rps]	rotations / s	v[rps] = v[Hz] / USC / FSC FSC: motor fullsteps per rotation, e.g. 200
TSTEP, TCOOLTHRS	-	*TSTEP* = f_{CLK} / f_{STEP} The time reference for velocity threshold is referred to the actual microstep frequency of the step input respectively velocity v[Hz].
VACTUAL	Two's complement signed internal velocity	*VACTUAL*[2300] = (f_{CLK}[Hz]/2 / 2^23) / v[Hz] With nominal oscillator: *VACTUAL*[2300] = 0.715Hz / v[Hz]

Hint
To monitor internal step pulse execution, program the DIAG output to provide step pulses (*GCONF.diag_step*). It toggles upon each step and thus shows half the microstep frequency. Use a timer input on your CPU to count pulses. Alternatively, regularly poll *MSCNT* to grasp steps done in the previous polling interval. It wraps around from 1023 to 0.

14 Driver Diagnostic Flags

The TMC2300 drivers supply a complete set of diagnostic and protection capabilities, like short to GND protection, short to VS protection and undervoltage detection. A detection of an open load condition allows testing if a motor coil connection is interrupted. See the *DRV_STATUS* table for details.

14.1 Temperature Measurement

The driver integrates a two-level temperature sensor (pre-warning and thermal shutdown) for diagnostics and for protection of the IC against excess heat. Heat is mainly generated by the motor driver stages. Most critical situations, where the driver MOSFETs could be overheated, are avoided by the short to GND protection. For many applications, the overtemperature pre-warning will indicate an abnormal operation situation and can be used to initiate user warning or power reduction measures like motor current reduction. The thermal shutdown is just an emergency measure and temperature rising to the shutdown level should be prevented by design.

TEMPERATURE THRESHOLDS

Temperature Level	**Comment**
150°C	This value is relatively safe to switch off the driver stage before the IC can be destroyed by overheating. On a large PCB, the power MOSFETs reach roughly 150°C peak temperature when the temperature detector is triggered with this setting.
120°C	Temperature level for pre-warning. In most applications, reaching this level is a sign for abnormal heat accumulation. The overtemperature pre-warning threshold of 120°C gives lots of headroom to react to high driver temperature, e.g. by reducing motor current, or increasing waiting-time in between of two motions.

Attention
Overtemperature protection cannot in all cases avoid thermal destruction of the IC. In case the rated motor current is exceed, e.g. by operating a motor in StealthChop with wrong parameters, or with automatic tuning parameters not fitting the operating conditions, excess heat generation can quickly heat up the driver before the overtemperature sensor can react. This is due to a delay in heat conduction over the IC die.

After triggering the overtemperature sensor (*ot* flag), the driver remains switched off until the system temperature falls below the pre-warning level (*otpw*) to avoid continuous heating to the shutdown level.

14.2 Short Protection

The TMC2300 power stages are protected against a short circuit condition by an additional measurement of the current flowing through each of the power stage MOSFETs. This is important, as most short circuit conditions result from a motor cable insulation defect, e.g. when touching the conducting parts connected to the system ground. The short detection is protected against spurious triggering, e.g. by ESD discharges, by retrying three times before switching off the motor.

Once a short condition is safely detected, the both driver bridges become switched off, and the *s2ga* or *s2gb* flag, respectively *s2vsa* or *s2vsb* becomes set. In order to restart the motor, disable and re-enable the driver. Note, that short protection cannot protect the system and the power stages for all possible short events, as a short event is rather undefined and a complex network of external components may be involved. Therefore, short circuits should basically be avoided.

14.3 Open Load Diagnostics

Interrupted cables are a common cause for systems failing, e.g. when connectors are not firmly plugged. The TMC2300 detects open load conditions by checking, if it can reach the desired motor coil current. This way, also undervoltage conditions, high motor velocity settings or short and overtemperature conditions may cause triggering of the open load flag, and inform the user, that motor torque may suffer. In motor stand still, open load cannot always be measured, as the coils might eventually have zero current.

Open load detection is provided for system debugging.
In order to safely detect an interrupted coil connection, read out the open load flags at low or nominal motor velocity operation, only. A flicker may occur with asymmetric sense resistors and does not harm. However, the *ola* and *olb* flags have just informative character and do not cause any action of the driver.

14.4 Diagnostic Output

The diagnostic output DIAG provides important status information. An active DIAG output shows that the driver cannot work normally. The index output signals the microstep counter zero position, to allow referencing (homing) a drive to a certain current pattern. The function set of the DIAG output can be modified by UART. Figure 14.1 shows the available signals and control bits.

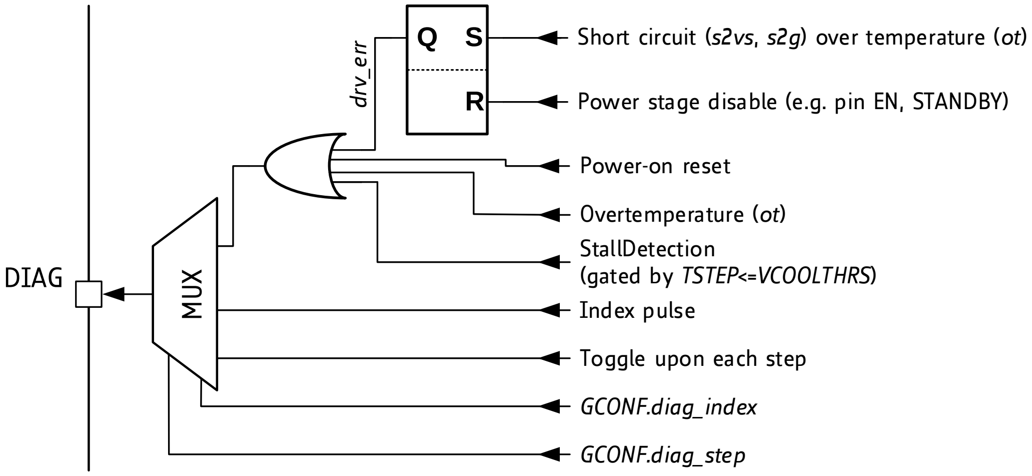


Figure 14.1 DIAG output

15 Quick Configuration Guide

This guide is meant as a practical tool to come to a first configuration. Do a minimum set of measurements and decisions for tuning the driver to determine UART-settings. The flow-charts concentrate on the basic function set to make a motor run smoothly. Once the motor runs, you may decide to explore additional features, e.g. freewheeling in more detail. A current probe on one motor coil is a good aid to find the best settings, but it is not a must.

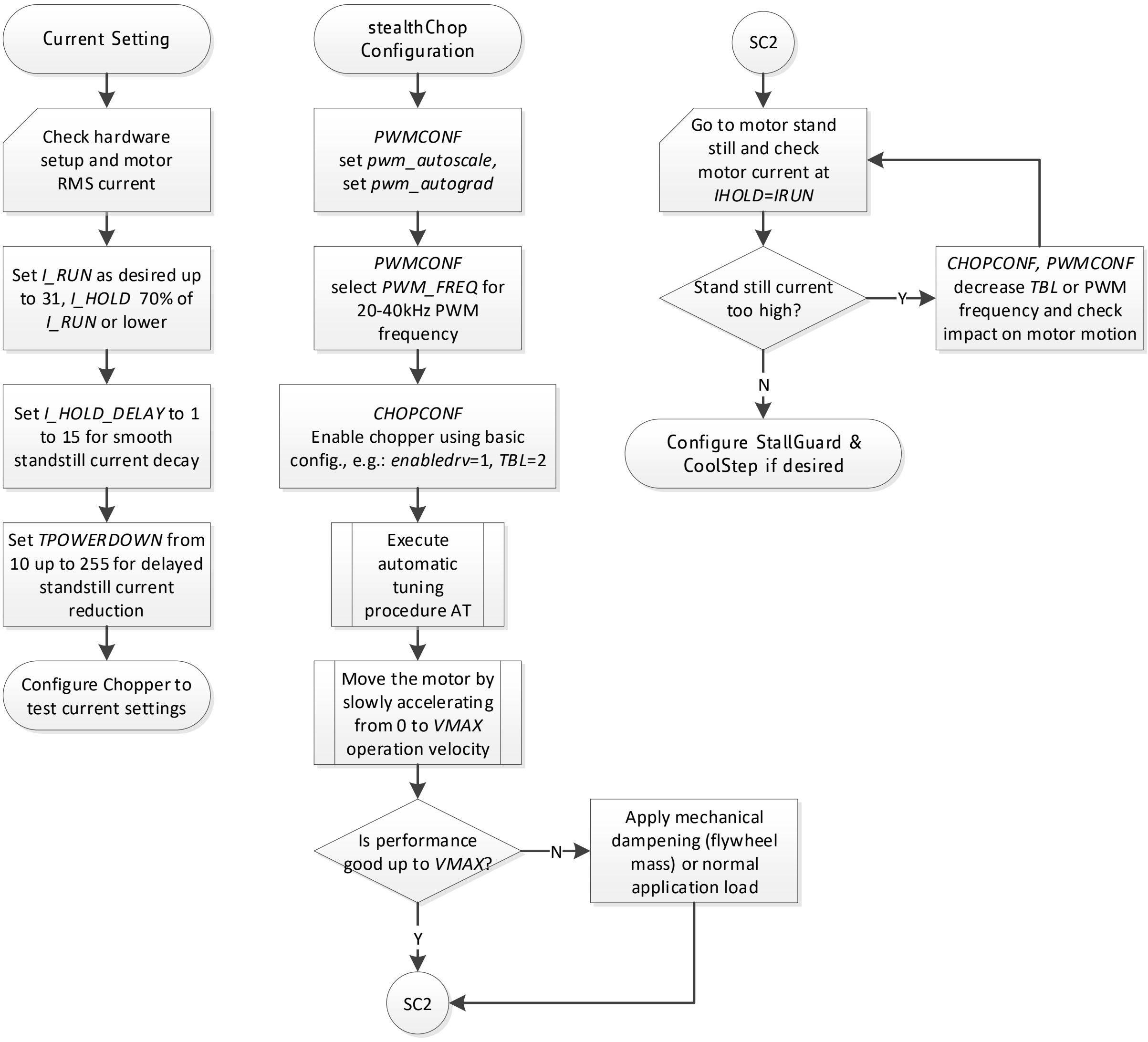


Figure 15.1 Current Setting and Chopper Configuration

Hint
Use the evaluation board to explore settings and to generate the required configuration datagrams.

Figure 15.2 Configuration for CoolStep in StealthChop mode

16 External Reset

The chip is loaded with default values during power on via its internal power-on reset. In order to reset the chip to power on defaults, any of the supply voltages monitored by internal reset circuitry (VS or VCC_IO) must be cycled. It is easiest and safest to cycle VCC_IO in order to completely reset the chip. Also, current consumed from VCC_IO is low and therefore it has simple driving requirements. Due to the input protection diodes not allowing the digital inputs to rise above VCC_IO level, all inputs must be driven low during this reset operation. When this is not possible, an input protection resistor may be used to limit current flowing into the related inputs.

17 Clock Oscillator

The clock is the timing reference for all functions: the chopper frequency, the blank time, the standstill power down timing, and the internal step pulse generator etc. The on-chip clock oscillator is not calibrated, but relatively temperature-stable. The internal clock frequency is roughly 12MHz. When the internal pulse generator is used, and increased precision is desired, measure the internal frequency by doing a test-motion (with motor disabled) and adapt the pulse frequency to the actual value of the frequency. Store the calibration value into the microcontroller's EEPROM for the application.

18 Absolute Maximum Ratings

The maximum ratings may not be exceeded under any circumstances. Operating the circuit at or near more than one maximum rating at a time for extended periods shall be avoided by application design.

Parameter	Symbol	Min	Max	Unit
Supply voltage operating with inductive load *)	V_{VS}	-0.5	11.2	V
Supply and bridge voltage max. *)	V_{VMAX}		13.2	V
I/O supply voltage	V_{VIO}	-0.5	5.5	V
digital supply voltage	V_{1V8OUT}	-0.5	1.95	V
Logic input voltage	V_I	-0.5	V_{VIO}+0.5	V
MODE input voltage (Do not exceed both, VCC_IO and 5VOUT by more than 10%, as this enables a test mode)	V_{VREF}	-0.5	6	V
Maximum current to / from digital pins and analog low voltage I/Os	I_{IO}		+/-10	mA
1.8V regulator output current (internal plus external load)	I_{5VOUT}		20	mA
Power bridge repetitive output current	I_{Ox}		2.0	A
Maximum VS current (both bridges operating)	I_{VS}		2.8	A
Maximum BRx current	I_{Ox}		2.0	A
Junction temperature	T_J	-50	150	°C
Storage temperature	T_{STG}	-55	150	°C
ESD-Protection for handling (Human body model, HBM)	V_{ESD}		1.5	kV

*) Stray inductivity of GND and VS connections will lead to ringing of the supply voltage when driving an inductive load. This ringing results from the fast switching slopes of the driver outputs in combination with reverse recovery of the body diodes of the output driver MOSFETs. Even small trace inductivities as well as stray inductivity of sense resistors can easily generate a few volts of ringing leading to temporary voltage overshoot. This should be considered when working near the maximum voltage.

19 Electrical Characteristics

19.1 Operational Range

Parameter	Symbol	Conditions	Min	Max	Unit
Junction temperature	T_J		-40	125	°C
Supply voltage	V_{VS}		2	11	V
Supply & IO voltage battery empty limit	V_{VS}		1.8		V
I/O supply voltage	V_{VIO}		2	5.25	V
RMS current per full bridge output for continuous operation (value for design guideline)	I_{RMS}	$V_{VS}<2.1V$	0.1	0.6	A
		$V_{VS}\geq 2.1V$		0.8	
		$V_{VS}\geq 2.2V$		1.0	
RMS motor coil current per fullbridge, duty cycle limited operation	I_{RMS}	$V_{VS}\geq 2.5V$		1.2	A
Peak output current per fullbridge output (sine wave)	I_{Ox}			1.7	A
Sum of output current (VS supply pin current)	I_{VS}			2.4	A

19.2 DC and Timing Characteristics

DC characteristics contain the spread of values guaranteed within the specified supply voltage range unless otherwise specified. Typical values represent the average value of all parts measured at +25°C. Temperature variation also causes stray to some values. A device with typical values will not leave Min/Max range within the full temperature range.

Power supply current	**DC-Characteristics** V_{VS} = 8.0V, V_{VIO}=3.3V					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Total supply current, driver disabled	I_{VS}	disable via UART		4	8	mA
		disable via EN=0		1.5	3	mA
Total supply current, operating	I_{VS}	default chopper, no load		7	12	mA
IO supply current operating	I_{VIO}	no load on outputs, inputs at V_{IO} or GND		60	200	µA
Total supply current, low-power standby, I_{VS}	I_{VS}	V_{VIO} < 0.2V		0.03	1	µA

Motor driver section	**DC- and Timing-Characteristics** V_{VS} = 8.0V, V_{VIO}=3.3V					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
RDS_{ON} lowside MOSFET	R_{ONL}	measure at 100mA, 25°C, V_{VS} ≥ 3.2V		0.17	0.25	Ω
RDS_{ON} highside MOSFET	R_{ONH}	measure at 100mA, 25°C, V_{VS} ≥ 3.2V		0.17	0.25	Ω
slope, rising	$t_{SLPRISE}$	value for reference		20		ns
slope, falling	$t_{SLPFALL}$	value for reference		7		ns
Current sourcing, driver off	I_{OIDLE}	O_{XX} pulled to GND	6	13	30	µA
Recommended / max. VS power-up slope to >5V	VS_{RAMP}	Hint: Normally satisfied due to ext. capacitor on VS.		<0.33 rcd.	1	V/µS

Charge pump	**DC-Characteristics**					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Charge pump output voltage (mean)	V_{VCP}-V_{VS}	V_{VS} ≥ 3.5V	4.2	5.1	5.7	V

Linear regulator	**DC-Characteristics** V_{VS} = 8.0V, V_{VIO}=3.3V					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Output voltage	V_{5VOUT}	I_{1V8OUT} = 0mA T_J = 25°C	1.65	1.8	1.95	V

Clock oscillator and input	**Timing-Characteristics**					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Clock oscillator frequency	f_{CLKOSC}	t_J=-50°C		11.7		MHz
	f_{CLKOSC}	t_J= 25°C	9	12.0	15	MHz
	f_{CLKOSC}	t_J=150°C		12.1		MHz

Detector levels	**DC-Characteristics** V_{VS} = 8.0V, V_{VIO}=3.3V					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
V_{VS} undervoltage threshold for RESET	V_{UV_VS}	V_{VS} rising		1.8	2.1	V
		V_{VS} falling	1.5	1.7	1.85	V
V_{VIO} undervoltage threshold for RESET	V_{UV_VIO}	V_{VIO} rising		1.8	1.9	V
		V_{VIO} falling	1.5	1.7		
V_{VIO} low power standby input voltage	$V_{UV\_VIOHYST}$				0.4	V
Worst case power-up delay time		V_{VS} = 2.0V, V_{VIO}=2.0V			500	µs
Short to GND detector threshold (V_{VS} - V_{Ox})	V_{OS2G}	V_{VS} ≥ 3.5V required for operation	0.5	0.8		V
Short to VS detector threshold (V_{Ox})	V_{OS2G}		1.0	1.2	1.6	V
Short detector delay (high side / low side switch on to short detected)	t_{S2G}			1		µs
Overtemperature prewarning 120°C	t_{OTPW}	Temperature rising	100	120	140	°C
Overtemperature shutdown 150°C	t_{OT150}	Temperature rising	135	150	170	°C
3.5V Detector Threshold *U3V5*	V_{VS3V5}	VS falling	3.15	3.5		V
3.5V Detector Threshold *U3V5*	V_{VS3V5}	VS rising		3.7	4.1	V
COMPA1A2, COMPB1B2 Offset Voltage	V_{CABOFS}	Lowside MOSFETs ON			+-10	mV

Sense resistor voltage levels	**DC-Characteristics** **f_{CLK}=16MHz**					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Sense input peak threshold voltage (low sensitivity)	V_{SRT}	*csactual*=31 *CUR_A/B*=248 Hyst.=0; I_{BRxy}=0		325		mV
Sense input tolerance / motor current full scale tolerance	I_{COIL}		-5		+5	%
Internal resistance from pin BRxy to internal sense comparator (additional to sense resistor)	R_{BRxy}			30		mΩ

Digital pins	**DC-Characteristics**					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Input voltage low level	V_{INLO}		-0.3		0.3 V_{VIO}	V
Input voltage high level	V_{INHI}		0.7 V_{VIO}		V_{VIO}+0.3	V
Input Schmitt trigger hysteresis	V_{INHYST}	V_{VIO}=3.3V		0.12 V_{VIO}		V
Output voltage low level	V_{OUTLO}	I_{OUTLO} = 2mA, V_{VIO}=3.3V			0.2	V
Output voltage high level	V_{OUTHI}	I_{OUTHI} = -2mA, V_{VIO}=3.3V	V_{VIO}-0.2			V
Input leakage current	I_{ILEAK}		-1		1	µA
Digital pin capacitance	C			3.5		pF

19.3 Thermal Characteristics

The following table shall give an idea on the thermal resistance of the package. The thermal resistance for a four-layer board will provide a good idea on a typical application. Actual thermal characteristics will depend on the PCB layout, PCB type and PCB size. The thermal resistance will benefit from thicker CU (inner) layers for spreading heat horizontally within the PCB. Also, air flow will reduce thermal resistance.

A thermal resistance of 40K/W for a typical board means, that the package is capable of continuously dissipating 1W at an ambient temperature of 85°C with the die temperature staying below/at 125°C. Note, that a thermally optimized layout is required.

Parameter	**Symbol**	**Conditions**	**Typ**	**Unit**
Typical power dissipation	P_D	StealthChop, 1A RMS in two phase motor, sinewave, 35kHz chopper, 11V, 60°C peak surface of package (motor QSH4218-035-10-027)	1	W
Typical power dissipation	P_D	StealthChop, 0.7A RMS in two phase motor, sinewave, 35kHz chopper, 11V, 45°C peak surface of package (motor QSH4218-035-10-027)	0.5	W
Thermal resistance junction to ambient on a multilayer board	R_{TMJA}	Dual signal and two internal power plane board (2s2p) as defined in JEDEC EIA JESD51-5 and JESD51-7 (FR4, 35µm CU, 70mm x 133mm, d=1.5mm)	40	K/W
Thermal resistance junction to case	R_{TJC}	Junction to heat slug of package	7	K/W

Table 19.1 Thermal characteristics

Note
A spread-sheet for calculating power dissipation is available on www.trinamic.com.

20 Layout Considerations

20.1 Exposed Die Pad

The TMC2300 uses its die attach pad to dissipate heat from the drivers and the linear regulator to the board. For best electrical and thermal performance, use a reasonable amount of solid, thermally conducting vias between the die attach pad and the ground plane. The printed circuit board should have a solid ground plane spreading heat into the board and providing for a stable GND reference.

20.2 Wiring GND

All signals of the TMC2300 are referenced to their respective GND. Directly connect all GND pins under the device to a common ground area (GND and die attach pad). The GND plane right below the die attach pad should be treated as a virtual star point. For thermal reasons, the PCB top layer shall be connected to a large PCB GND plane spreading heat within the PCB.

Attention

Especially the sense resistors are susceptible to GND differences and GND ripple voltage, as the microstep current steps make up for voltages down to 0.5 mV. No current other than the sense resistor current should flow on their connections to GND and to the TMC2300. Optimally place them close to the IC, with one or more vias to the GND plane for each sense resistor. The two sense resistors for one coil should not share a common ground connection trace or vias, as also PCB traces have a certain resistance.

20.3 Supply Filtering

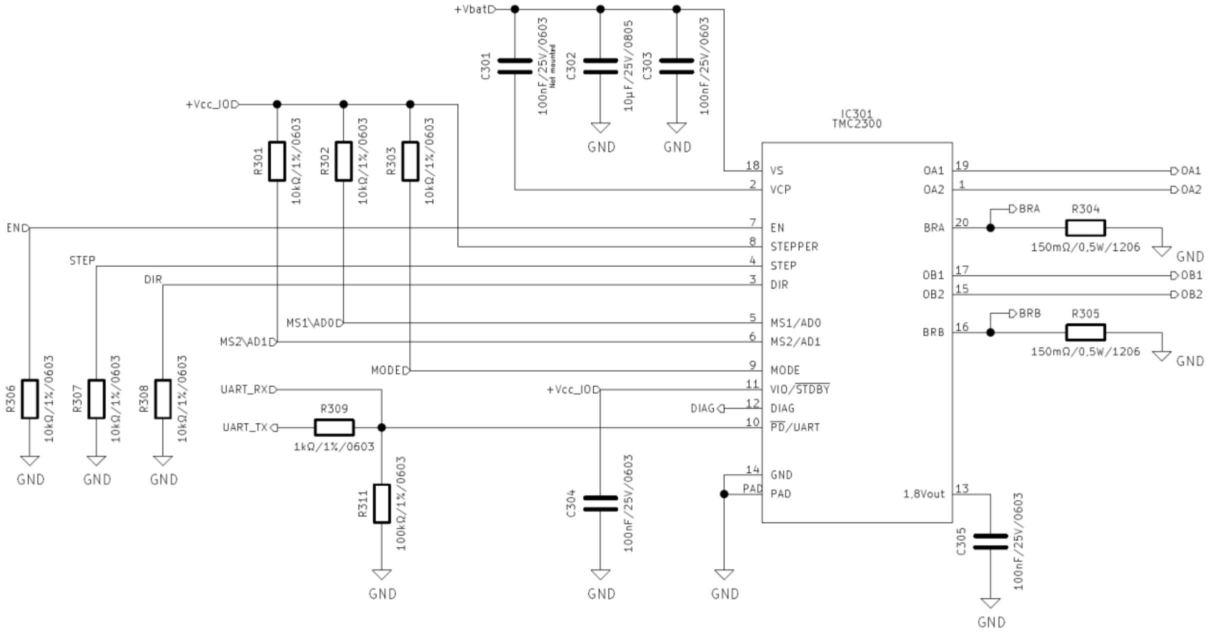
The 1.8VOUT output voltage ceramic filtering capacitor (100 nF recommended) should be placed as close as possible to the 1.8VOUT pin, with its GND return going directly to the die pad or the nearest GND pin. This ground connection shall not be shared with other loads or additional vias to the GND plane. Use as short and as thick connections as possible.

The motor supply pins VS should be decoupled with a ceramic, or a ceramic plus an electrolytic capacitor (47 µF or larger is recommended, depending on the motor coil current). Place the capacitors close to the device.

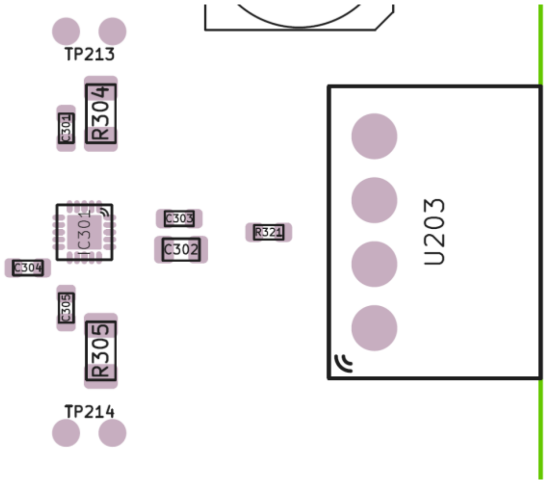
Take into account that the switching motor coil outputs have a high dV/dt. Thus, capacitive stray into high resistive signals can occur, if the motor traces are near other traces over longer distances.

20.4 Layout Example

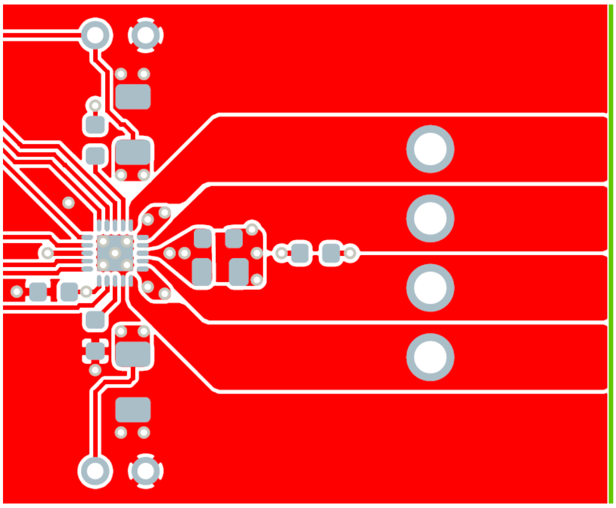
Schematic



Placement (Excerpt)



Top Layout (Excerpt, showing die pad vias)



The complete schematics and layout data for all evaluation boards are available on the TRINAMIC website.

21 Package Mechanical Data

21.1 Dimensional Drawings QFN20

Attention: Drawings not to scale.

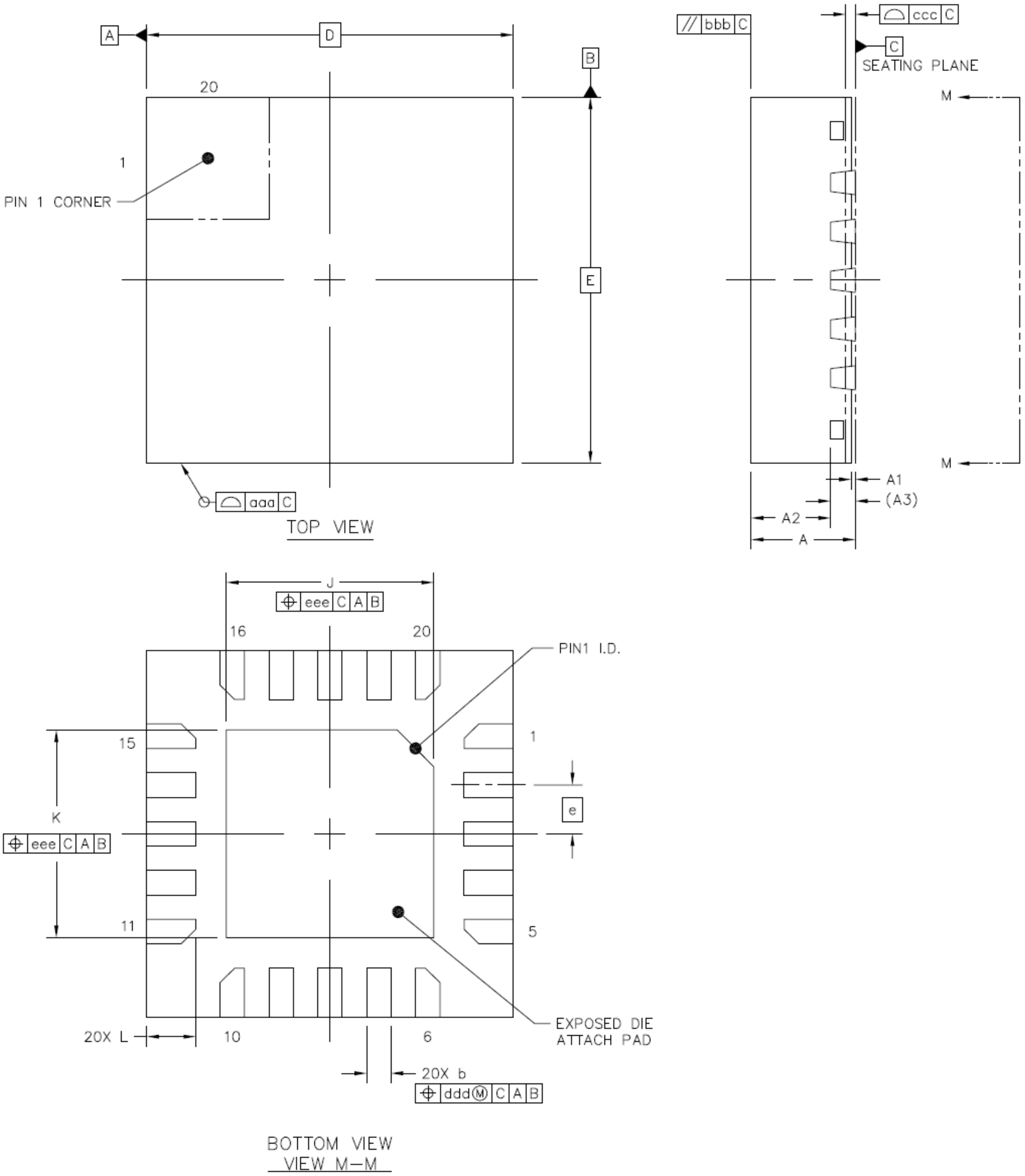


Figure 21.1 Dimensional drawings QFN20

Parameter [mm]	Ref	Min	Nom	Max
total thickness	A	0.8	0.85	0.9
stand off	A1	0	0.035	0.05
mold thickness	A2		0.65	0.67
lead frame thickness	A3		0.203	
Lead width	b	0.15	0.2	0.25
body size X	D		3.0	
body size Y	E		3.0	
lead pitch	e		0.4	
exposed die pad size X	J	1.6	1.7	1.8
exposed die pad size Y	K	1.6	1.7	1.8
lead length	L	0.35	0.4	0.45
package edge tolerance	aaa			0.1
mold flatness	bbb			0.1
coplanarity	ccc			0.08
lead offset	ddd			0.1
exposed pad offset	eee			0.1

21.2 Package Codes

Type	Package	Temperature range	Code & marking
TMC2300-LA	QFN20 (RoHS)	-40°C ... +125°C	(TMC logo) 2300

22 Design Philosophy

The TMC2300 is our entry into battery powered devices for IOT and mobile devices. While the TMC2300 inherits its super silent and energy efficient chopper from the TMC2208 family, well known in 3D printer community, it is only the second device featuring the brand-new sensorless stall detection StallGuard4. The challenge was to realize a wide voltage range, highly efficient power driver and control logic for advanced algorithms in a tiny 3mm x 3mm package. We are proud of the resulting power stage design with an internal charge pump: It showed overwhelming results, even for our developers! Additionally, we squeezed in a full set of protection functions and a considerable logic part for the advanced algorithms, only limited by the maximum die size possible in this package. The result is a perfect 100% use of the available space.

Integration at street level cost was possible by squeezing know-how into a few mm² of layout, using one of the most modern smart power processes. The IC comprises all the knowledge gained from designing motion controller and driver chips, drivers, and complex motion control systems for more than 20 years. Our deep involvement, practical testing and the stable team ensure the highest level of confidence and product reliability.

Bernhard Dwersteg, CTO and co-founder

23 Table of Figures

24 Revision History

Version	**Date**	**Author** **BD= Bernhard Dwersteg**	**Description**
V0.7	2019-May-07	BD	First version based on TMC2209-Datasheet / Adapted but not finished
V0.8	2019-May-29	BD	Edited electrical data based on prototype measurements
V0.9	2019-Jun-28	BD	Electrical data from IC-Test
V0.92	2019-Jul-18	BD	Added RDSon vs VS measurements
V0.93	2019-Jul-25	BD	Minor changes
V0.94	2019-Jul-29	BD	Added chapter selecting a motor
V1.00	2019-Aug-02	BD	Adapted ESD rating, removed hint preliminary
V1.01	2019-Aug-16	BD	Minor wording

Table 24.1 Document Revisions

25 References

[TMC2300-EVAL] TMC2300 Evaluation board: Manuals, software and PCB data available on www.trinamic.com

Calculation sheet TMC2300_Calculations.xlsx www.trinamic.com