Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

TMC5031 DATASHEET

Dual, cost-effective controller and driver for up to two 2-phase bipolar stepper motors. Integrated motion controller with SPI interface.

Applications

CCTV, Security
Antenna Positioning
Heliostat Controller
Battery powered applications
Office Automation
ATM, Cash recycler, POS
Lab Automation
Liquid Handling
Medical
Printer and Scanner
Pumps and Valves

Features and Benefits

2-phase stepper motors

Drive Capability up to 2 x 1.1A coil current

Motion Controller with **sixPoint™** ramp

Voltage Range 4.75… 16V DC

SPI Interface

2x Ref.-Switch input per axis

Highest Resolution 256 microsteps per full step

Full Protection & Diagnostics

stallGuard2™ high precision sensorless motor load detection

coolStep™ load dependent current saves up to 75% energy

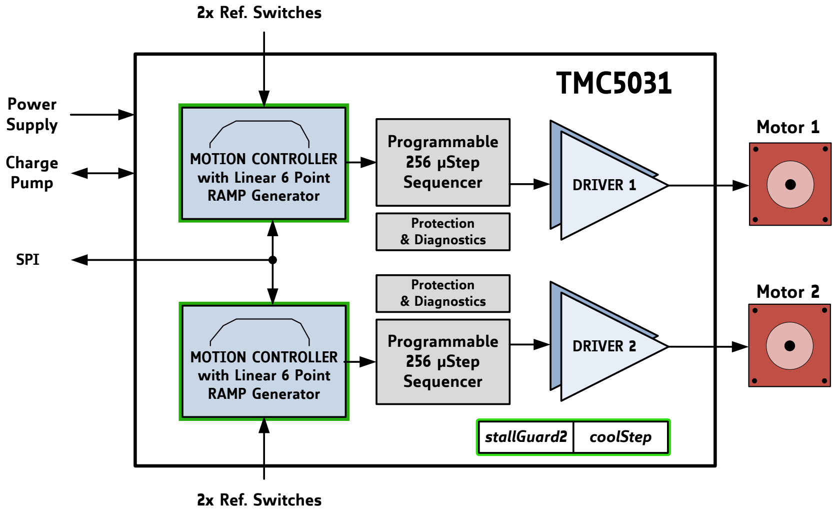
spreadCycle™ high-precision chopper for best current sine wave form and zero crossing with additional **chopSync2™**

Compact Size 7x7mm QFN48 package

Description

The TMC5031 is a low cost motion controller and driver IC for up to two stepper motors. It combines two flexible ramp motion controllers with energy efficient stepper motor drivers. The drivers support two-phase stepper motors and offer an industry-leading feature set, including high-resolution microstepping, sensorless mechanical load measurement, load-adaptive power optimization, and low-resonance chopper operation. All features are controlled by a standard SPI™ interface. Integrated protection and diagnostic features support robust and reliable operation. High integration, high energy efficiency and small form factor enable miniaturized designs with low external component count for cost-effective and highly competitive solutions.

Block Diagram

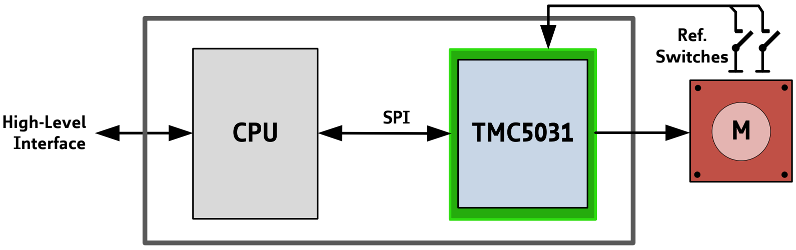

2x Ref. Switches
TMC5031
Power Supply
Charge Pump
SPI
MOTION CONTROLLER with Linear 6 Point RAMP Generator
Programmable 256 µStep Sequencer
Protection & Diagnostics
DRIVER 1
Motor 1
Protection & Diagnostics
MOTION CONTROLLER with Linear 6 Point RAMP Generator
Programmable 256 µStep Sequencer
DRIVER 2
Motor 2
stallGuard2
coolStep
2x Ref. Switches

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

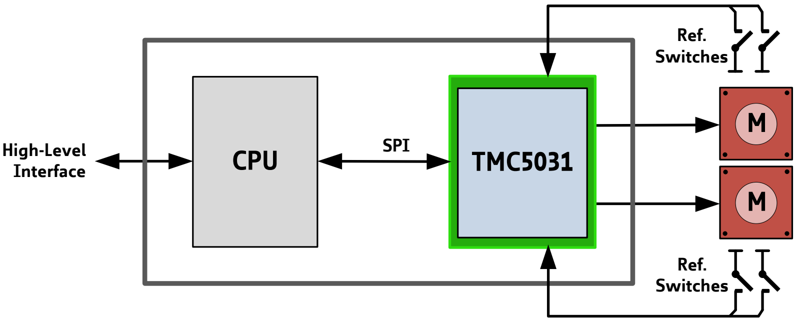
APPLICATION EXAMPLES: HIGH FLEXIBILITY – MULTIPURPOSE USE

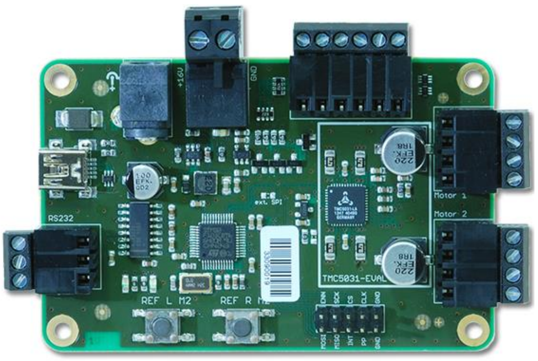
The TMC5031 scores with power density, complete motion controlling features and integrated power stages. It offers a versatility that covers a wide spectrum of applications from battery systems up to embedded applications with 1.1A current per motor. The small form factor keeps costs down and allows for miniaturized layouts. Extensive support at the chip, board, and software levels enables rapid design cycles and fast time-to-market with competitive products. High energy efficiency and reliability from TRINAMIC's coolStep technology deliver cost savings in related systems such as power supplies and cooling.

Miniaturized Design for up to two Stepper Motors



Two reference switch inputs can be used for each motor. A single CPU controls the whole system, which is highly economical and space saving.





TMC5031-EVAL Evaluation Board

Evaluation & Development Platform

The TMC5031-EVAL is a tiny evaluation board, combining the TMC5031 with its basic external components and a 32 bit microcontroller interfacing to a PC. The firmware source code is available from the TRINAMIC website to allow own modifications and to make design-in easy.

Order Codes

Order code	Description	Size [mm^2]
TMC5031-LA	Dual stallGuard2™ and coolStep™ controller/driver, QFN48	7 x 7
TMC5031-EVAL	Evaluation board for TMC5031	85 x 55

Table of Contents

1 Principles of Operation

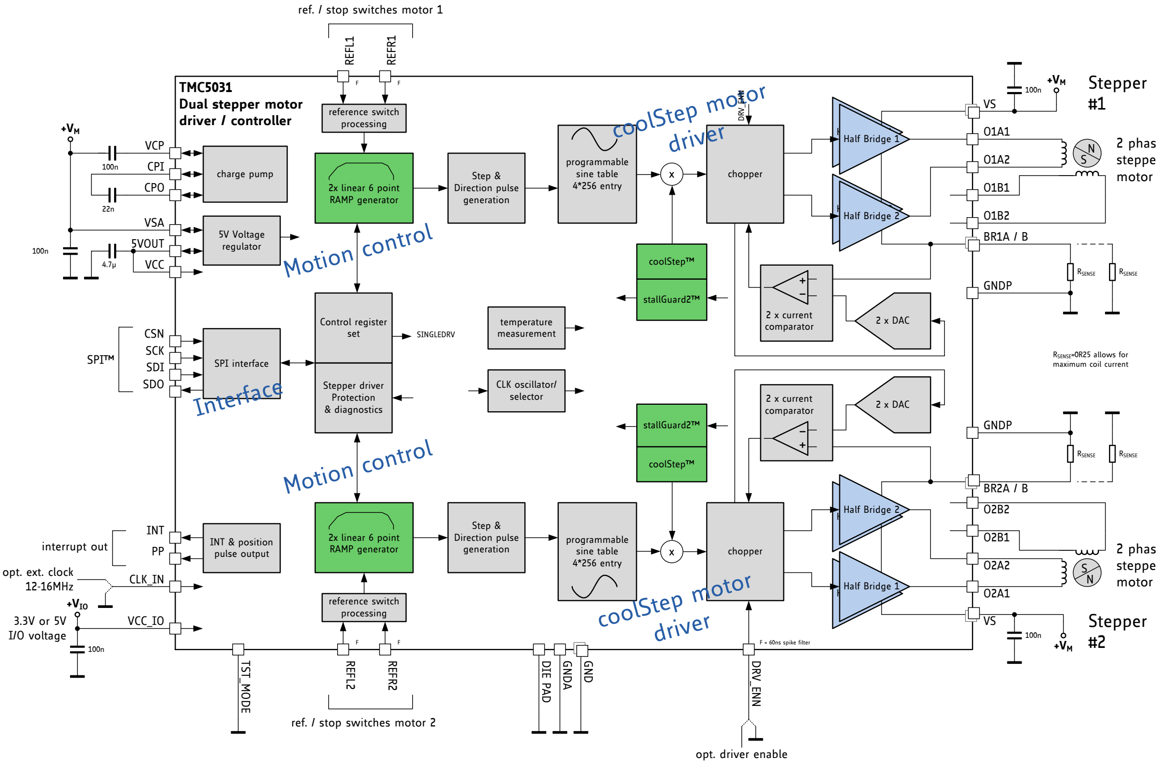


Figure 1.1 Basic application and block diagram

The TMC5031 motion controller and driver chip is an intelligent power component interfacing between the CPU and up to two stepper motors. All stepper motor logic is completely within the TMC5031. No software is required to control the motor – just provide target positions. The TMC5031 offers a number of unique enhancements which are enabled by the system-on-chip integration of driver and controller. The sixPoint ramp generator of the TMC5031 uses coolStep and stallGuard2 automatically to optimize every motor movement: TRINAMICs special features contribute toward lower system cost, greater precision, greater energy efficiency, smoother motion, and cooler operation in stepper motor applications. The clear concept and the comprehensive solution save design-in time.

1.1 Key Concepts

The TMC5031 implements several 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.

stallGuard2™ High-precision load measurement using the back EMF on the motor coils.

coolStep™ Load-adaptive current control which reduces energy consumption by as much as 75%.

spreadCycle™ High-precision chopper algorithm available as an alternative to the traditional constant off-time algorithm.

sixPoint™ Fast and precise positioning using a hardware ramp generator with a set of four acceleration / deceleration settings. Quickest response due to dedicated hardware.

In addition to these performance enhancements, TRINAMIC motor drivers also 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 SPI Control Interface

The SPI interface is a bit-serial interface synchronous to a bus clock. For every bit sent from the bus master to the bus slave, another bit is sent simultaneously from the slave to the master. Communication between an SPI master and the TMC5031 slave always consists of sending one 40-bit command word and receiving one 40-bit status word.

The SPI command rate typically is a few commands per complete motor motion.

1.3 Software

From a software point of view the TMC5031 is a peripheral with a number of control and status registers. Most of them can either be written only or 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.

1.4 Moving and Controlling the Motor

1.4.1 Integrated Motion Controller

The integrated 32 bit motion controller automatically drives the motors to target positions, or accelerates to target velocities. All motion parameters can be changed on the fly. The motion controller recalculates immediately. A minimum set of configuration data consists of acceleration and deceleration values and the maximum motion velocity. A start and stop velocity is supported as well as a second acceleration and deceleration setting. The integrated motion controller supports immediate reaction to mechanical reference switches and to the sensorless stall detection stallGuard2.

Benefits are:

- Flexible ramp programming
- Efficient use of motor torque for acceleration and deceleration allows higher machine throughput
- Immediate reaction to stop and stall conditions

1.5 Precision Driver with Programmable Microstepping Wave

Current into the motor coils is controlled using a cycle-by-cycle chopper mode. Two chopper modes are available: a traditional constant off-time mode and the new spreadCycle mode. Constant off-time mode provides higher torque at the highest velocity, while spreadCycle mode offers smoother operation and greater power efficiency over a wide range of speed and load. The spreadCycle chopper scheme automatically integrates a fast decay cycle and guarantees smooth zero crossing performance. Programmable microstep shapes allow optimizing the motor performance.

Benefits are:

- Significantly improved microstepping with low cost motors
- Motor runs smooth and quiet
- Reduced mechanical resonances yields improved torque

1.6 stallGuard2 – Mechanical Load Sensing

stallGuard2 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.

1.7 coolStep – Load Adaptive Current Control

coolStep drives the motor at the optimum current. It uses the stallGuard2 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, making the drive an efficient and precise solution.

Benefits are:

- *Energy efficiency*	power consumption decreased up to 75%
- *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

Figure 1.2 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.

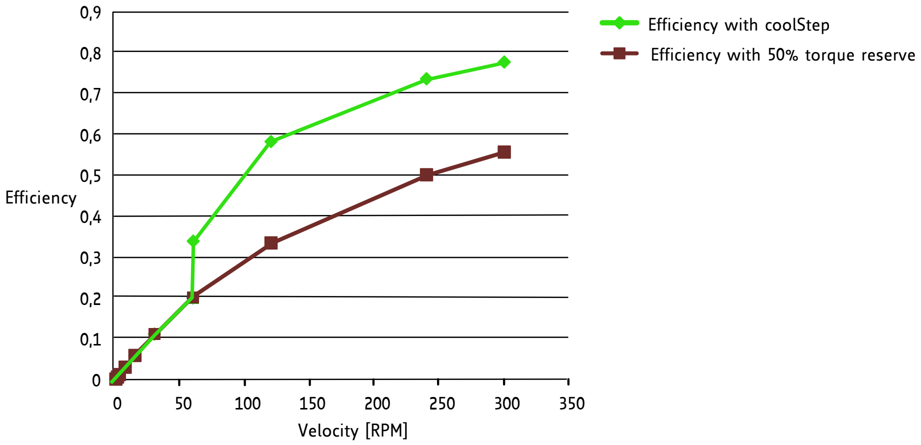


Figure 1.2 Energy efficiency with coolStep (example)

2 Pin Assignments

2.1 Package Outline

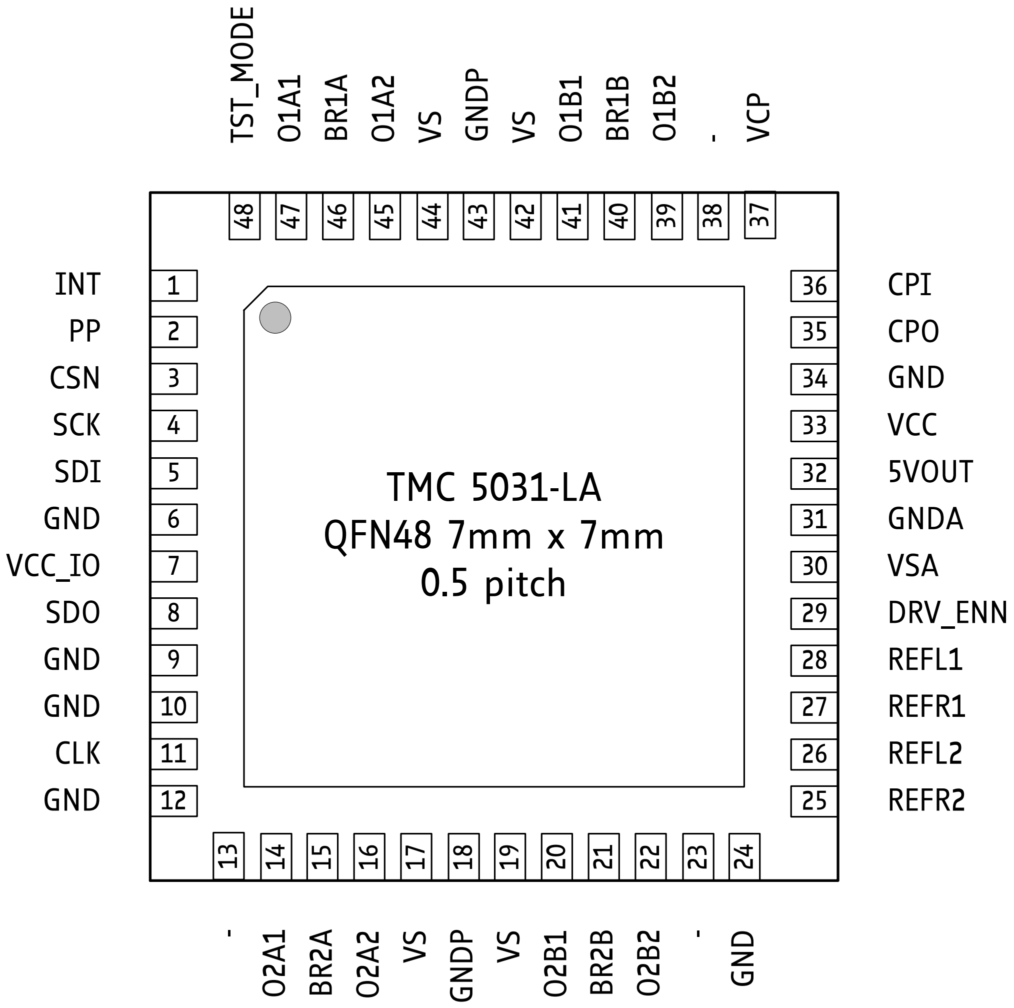


Figure 2.1 TMC5031 pin assignments.

2.2 Signal Descriptions

Pin	Number	Type	Function
GND	6, 9, 10, 12, 24, 34	GND	Digital ground pin for IO pins and digital circuitry.
VCC_IO	7		3.3V or 5V I/O supply voltage pin for all digital pins.
VSA	30		Analog supply voltage for 5V regulator – typically supplied with driver supply voltage. An additional 100nF capacitor to GND (GND plane) is recommended for best performance.
GNDA	31	GND	Analog GND
5VOUT	32		Output of internal 5V regulator. Attach 2.2µF or larger ceramic capacitor to GNDA near to pin for best performance. May be used to supply VCC of chip.

Pin	Number	Type	Function
VCC	33		5V supply input for digital circuitry within chip and charge pump. Attach 470nF capacitor to GND (GND plane). May be supplied by 5VOUT. A 2.2Ω resistor is recommended for decoupling noise from 5VOUT. When using an external supply, make sure, that VCC comes up before or in parallel to 5VOUT.
DIE_PAD	-	GND	Connect the exposed die pad to a GND plane. Provide as many as possible vias for heat transfer to GND plane.

Table 2.1 Low voltage digital and analog power supply pins

Pin	Number	Type	Function
CPO	35	O(VCC)	Charge pump driver output. Outputs 5V (GND to VCC) square wave with 1/16 of internal oscillator frequency.
CPI	36	I(VCP)	Charge pump capacitor input: Provide external 22 nF / 50 V capacitor to CPO.
VCP	37		Output of charge pump. Provide external 100 nF capacitor to VS.

Table 2.2 Charge pump pins

Pin	Number	Type	Function
INT	1	O (Z)	Tristate interrupt output. Can be programmed to provide interrupt output based on ramp generator flags *RAMP_STAT* bits 4, 5, 6 & 7 (*poscmp_enable*=1).
PP	2	O (Z)	Tristate position compare output for motor 1 (*poscmp_enable*=1).
CSN	3	I	Chip select input of SPI interface
SCK	4	I	Serial clock input of SPI interface
SDI	5	I	Data input of SPI interface
SDO	8	O (Z)	Tristate data output of SPI interface (enabled with CSN=0)
CLK	11	I	Clock input. Tie to GND using short wire for internal clock or supply external clock. The first high signal disables the internal oscillator until power down.
REFR2	25	I	Right reference switch input for motor 2
REFL2	26	I	Left reference switch input for motor 2
REFR1	27	I	Right reference switch input for motor 1
REFL1	28	I	Left reference switch input for motor 1
DRV_ENN	29	I	Enable input for motor drivers. The power stage becomes switched off (all motor outputs floating) when this pin becomes driven to a high level. Tie to GND for normal operation.
TST_MODE	48	I	Test mode input. Puts IC into test mode. Tie to GND for normal operation.
-	13, 23, 38	N.C.	Unused pins – no internal electrical connection. Leave open or tie to GND for compatibility with future devices.

Table 2.3 Digital I/O pins (all related to VCC_IO supply)

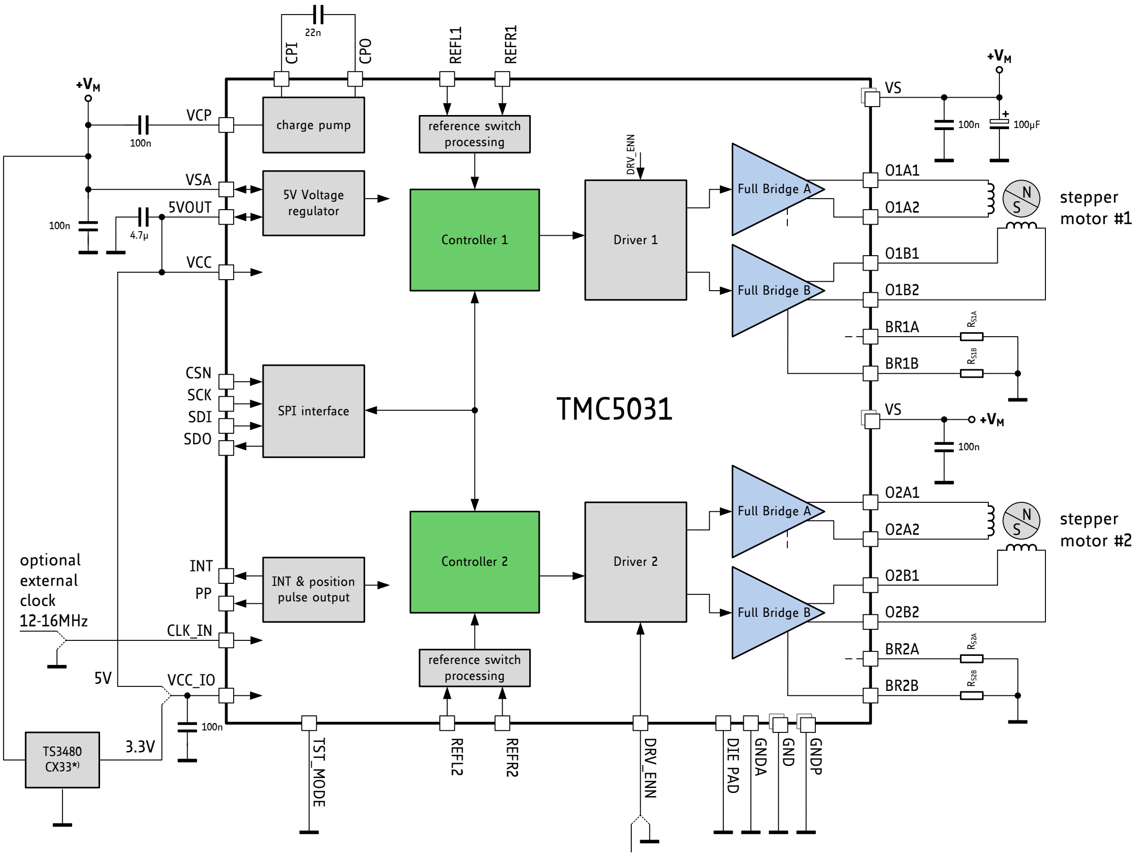
Pin	Number	Type	Function
O2A1	14	O (VS)	Motor 2 coil A output 1
BR2A	15		Sense resistor connection for motor 2 coil A. Place sense resistor to GND near pin.
O2A2	16	O (VS)	Motor 2 coil A output 2
VS	17, 19		Motor supply voltage. Provide filtering capacity near pin with shortest loop to nearest GNDP pin (respectively via GND plane).
GNDP	18	GND	Power GND. Connect to GND plane near pin.
O2B1	20	O (VS)	Motor 2 coil B output 1
BR2B	21		Sense resistor connection for motor 2 coil B. Place sense resistor to GND near pin.
O2B2	22	O (VS)	Motor 2 coil B output 2
O1B2	39	O (VS)	Motor 1 coil B output 2
BR1B	40		Sense resistor connection for motor 1 coil B. Place sense resistor to GND near pin.
O1B1	41	O (VS)	Motor 1 coil B output 1
VS	42, 44		Motor supply voltage. Provide filtering capacity near pin with shortest loop to nearest GNDP pin (respectively via GND plane).
GNDP	43	GND	Power GND. Connect to GND plane near pin.
O1A2	45	O (VS)	Motor 1 coil A output 2
BR1A	46		Sense resistor connection for motor 1 coil A. Place sense resistor to GND near pin.
O1A1	47	O (VS)	Motor 1 coil A output 1

Table 2.4 Power driver pins

3 Sample Circuits

The sample circuits show the connection of the external components in different operation and supply modes. The connection of the bus interface and further digital signals is left out for clarity.

3.1 Standard Application Circuit



) For a reliable start-up it is essential that VCC_IO comes up to a minimum of 1.5V before the TMC5031 leaves the reset condition. Therefore, TRINAMIC recommends using a fast-start-up voltage regulator (e.g. TS3480CX33) in a 3.3V environment.

Figure 3.1 Standard application circuit

The standard application circuit uses a minimum set of additional components in order to operate the motor. Use low ESR capacitors for filtering the power supply capable to cope with the current ripple. The current ripple often depends on the power supply and cable length. The VCC_IO voltage can be supplied from 5VOUT, or from a fast startup 3.3V regulator. In order to minimize linear voltage regulator power dissipation of the internal 5V voltage regulator in applications where VM is high, a different (lower) supply voltage can be used for VSA, if available. For best motor chopper performance, an optional R/C-filter de-couples 5VOUT from digital noise cause by power drawn from VCC.

Basic layout 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 5VOUT filtering capacitor directly to 5VOUT and GNDA pin. See layout hints for more details. Low ESR electrolytic capacitors are recommended for VS filtering.

Attention

In case VSA is supplied by a different voltage source, make sure that VSA does not exceed VS by more than one diode drop upon power up or power down.

3.1.1 VCC_IO Requirements

For a reliable start-up it is essential that VCC_IO comes up to a minimum of 1.5V before the TMC5031 leaves the reset condition. The reset condition ends earliest 50µs after the time when VSA exceeds its undervoltage threshold of typically 4.2V, or when 5VOUT exceeds its undervoltage threshold of typically 3.5V, whichever comes last.

THERE ARE THREE WAYS TO COME UP TO VCC_IO REQUIREMENTS

- 5VOUT can be used directly to supply VCC_IO. In this case there are no further requirements.

- An external low drop regulator can be used in a 3.3V environment as shown in Figure 3.1. Note, that most voltage regulators are not suitable for this application because they show a delayed boot up. The following external regulators are proved by TRINAMIC:

TS3480CX33	This regulator can be used within the full supply voltage range when tied to the motor supply voltage.
LD1117-3.3	This regulator can be used to supply VCC_IO from 5VOUT, or from a supply voltage of up to 15V.

- VCC_IO can be supplied externally as shown in Figure 3.2 . In this case it is mandatory to connect the Schottky diode to the logic supply of the external circuitry. Please note, that the 2K resistor is not to be used with 5V I/O voltage.

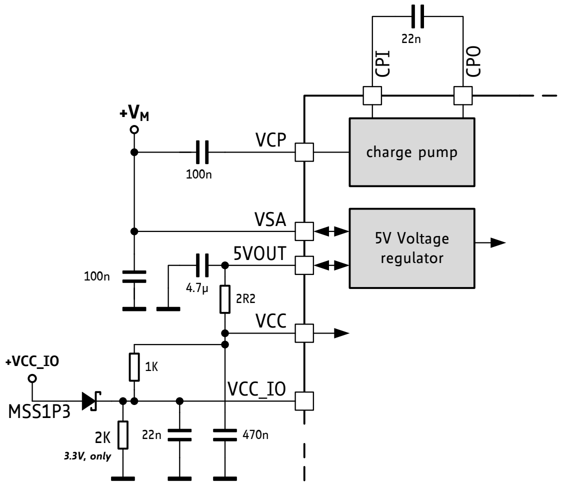


Figure 3.2 External supply of VCC_IO (showing optional filtering for VCC)

Refer to application note no. 028 *Supply Voltage Considerations: VCC_IO in TMC50xx Designs* (www.trinamic.com). Here you will find complete information about connecting VCC_IO.

3.2 5 V Only Supply

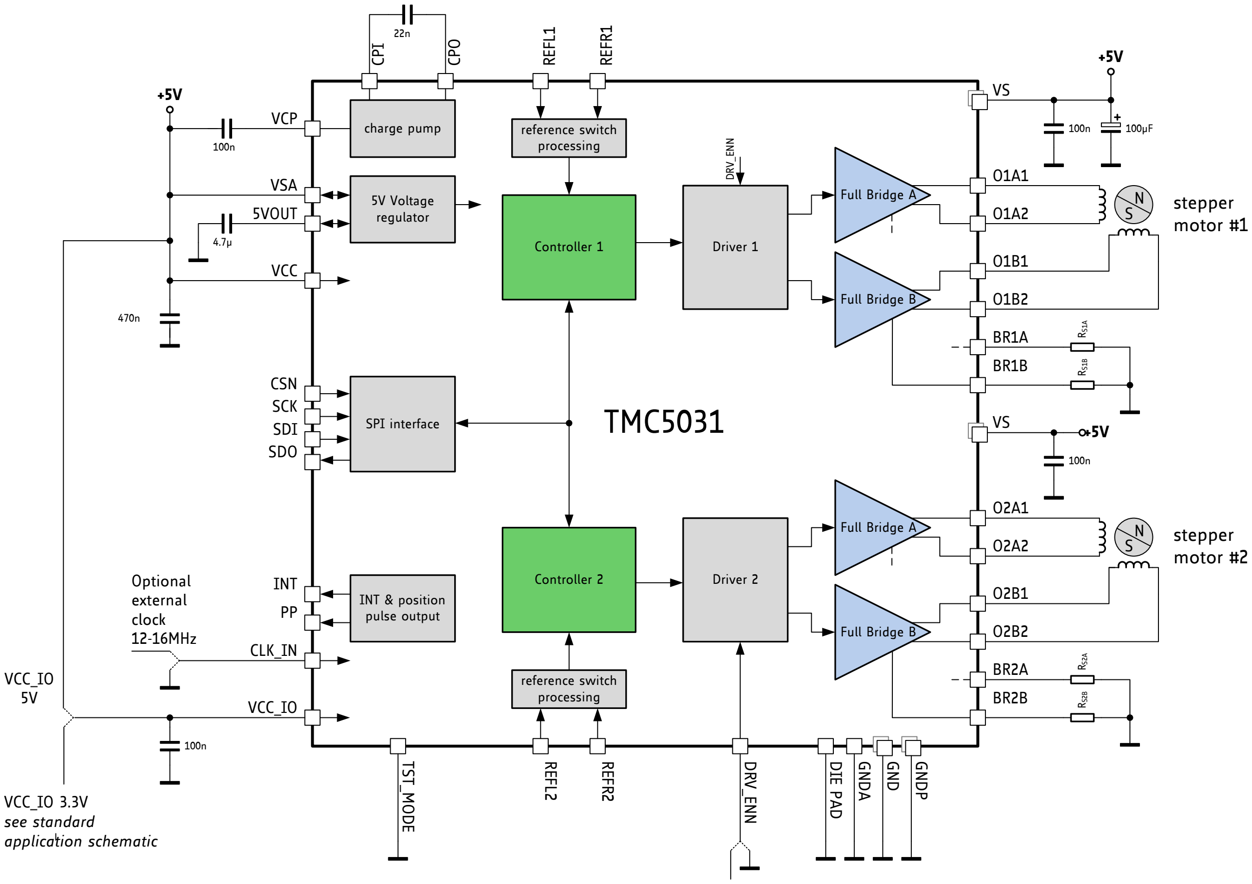


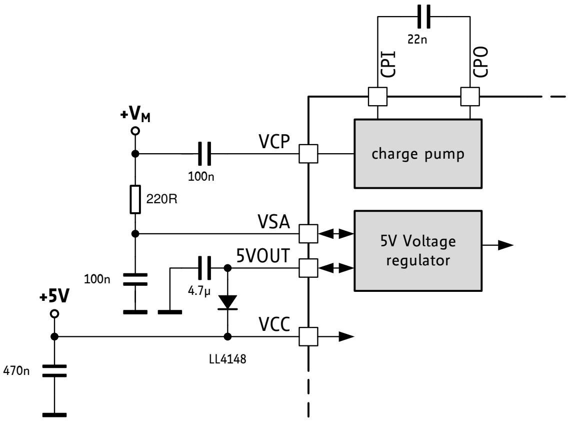
Figure 3.3 5V only operation

While the standard application circuit is limited to roughly 5.5 V lower supply voltage, a 5 V only application lets the IC run from a normal 5 V +/-5% supply. In this application, linear regulator drop must be minimized. Therefore, the major 5 V load is removed by supplying VCC directly from the external supply. In order to keep supply ripple away from the analog voltage reference, 5VOUT should have an own filtering capacity and the 5VOUT pin does not become bridged to the 5V supply.

3.3 External VCC Supply

Supplying VCC from an external supply is advised, when cooling of the chip is critical, e.g. at high environment temperatures in combination with high supply voltages (16 V), as the linear regulator is a major source of on-chip power dissipation. It must be made sure that the external VCC supply comes up before or synchronously with the 5VOUT supply, because otherwise the power-up reset event may be missed by the TMC5031. A diode from 5VOUT to VCC ensures this, in case the external voltage regulator is not a low drop type linear regulator. In order to prevent overload of the internal 5V regulator when using this diode, an additional series resistor has been added to VSA.

An alternative for reduced power dissipation is using a lower supply voltage for VSA, e.g. 6V to 12V. If power dissipation is critical, but no external supply is available, the clock frequency can be reduced as a first step by supplying external 12 MHz clock.



The diode is mandatory to satisfy power-up conditions!

Figure 3.4 Using an external 5V supply to reduce linear regulator power dissipation

3.3.1 Internal Regulator Bridged

In case a clean external 5V supply is available, it can be used for complete supply of analog and digital part (Figure 3.5). The circuit will benefit from a well-regulated supply, e.g. when using a +/-1% regulator. A precise supply guarantees increased motor current precision, because the voltage at 5VOUT directly is the reference voltage for all internal units of the driver, especially for motor current control. For best performance, the power supply should have low ripple to give a precise and stable supply at 5VOUT pin with remaining ripple well below 5mV. Some switching regulators have a higher remaining ripple, or different loads on the supply may cause lower frequency ripple. In this case, increase capacity attached to 5VOUT. In case the external supply voltage has poor stability or low frequency ripple, this would affect the precision of the motor current regulation as well as add chopper noise.

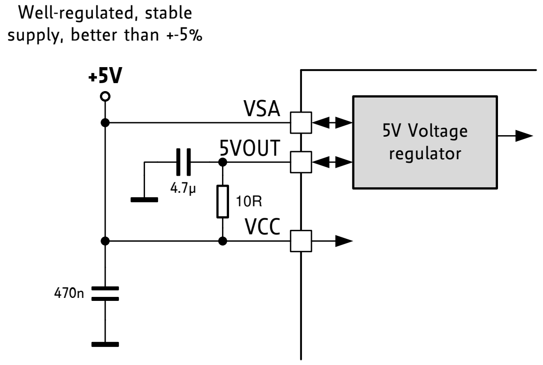


Figure 3.5 Using an external 5V supply to bypass internal regulator

3.4 Optimizing Analog Precision

The 5VOUT pin is used as an analog reference for operation of the TMC5031. Performance will degrade when there is voltage ripple on this pin. Most of the high frequency ripple in a TMC5031 design results from the operation of the internal digital logic. The digital logic switches with each edge of the clock signal. Further, ripple results from operation of the charge pump, which operates with roughly 1 MHz and draws current from the VCC pin. In order to keep this ripple as low as possible, an additional filtering capacitor can be put directly next to the VCC pin with vias to the GND plane giving a short connection to the digital GND pins (pin 6 and pin 34). Analog performance is best, when this ripple is kept away from the analog supply pin 5VOUT, using an additional series resistor of 2.2 Ω to 3.3 Ω. The voltage drop on this resistor will be roughly 100 mV (I_{VCC} * R).

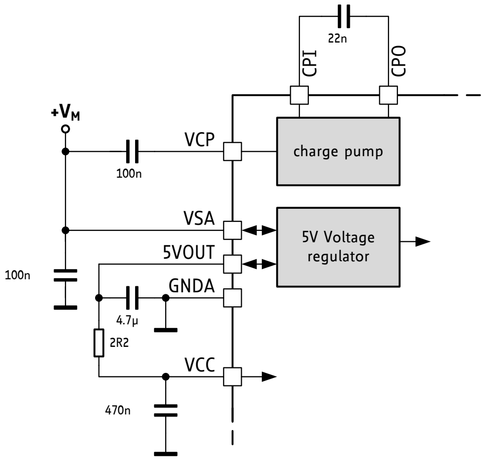
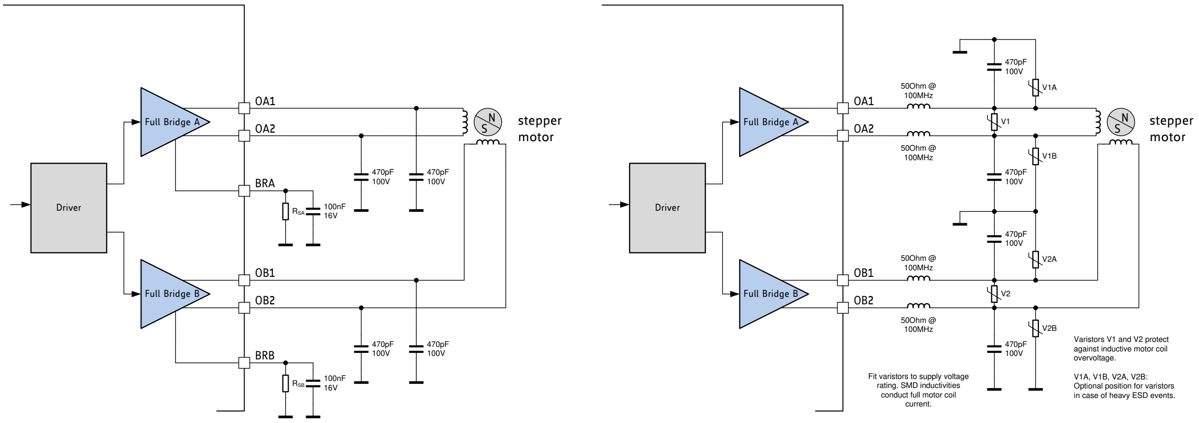


Figure 3.6 Adding an RC-Filter on VCC for reduced ripple

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. 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 might be varied between 100pF and 1nF. The capacitors also dampen high frequency noise injected from digital parts of the circuit 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 a varistor against ESD voltage.



Fit varistors to supply voltage rating. SMD inductivities conduct full motor coil current.

Varistors V1 and V2 protect against inductive motor coil overvoltage.

V1A, V1B, V2A, V2B: Optional position for varistors in case of heavy ESD events.

Figure 3.7 Simple ESD enhancement and more elaborate motor output protection

4 SPI Interface

4.1 SPI Datagram Structure

The TMC5031 uses 40 bit SPI™ (Serial Peripheral Interface, SPI is Trademark of Motorola) datagrams for communication with a microcontroller. Microcontrollers which are equipped with hardware SPI are typically able to communicate using integer multiples of 8 bit. The NCS line of the TMC5031 must be handled in a way, that it stays active (low) for the complete duration of the datagram transmission.

Each datagram sent to the TMC5031 is composed of an address byte followed by four data bytes. This allows direct 32 bit data word communication with the register set of the TMC5031. Each register is accessed via 32 data bits even if it uses less than 32 data bits.

For simplification, each register is specified by a one byte address:

- For a read access the most significant bit of the address byte is 0.
- For a write access the most significant bit of the address byte is 1.

Most registers are write only registers, some can be read additionally, and there are also some read only registers.

<table>
<tr><th colspan="40">TMC5031 SPI DATAGRAM STRUCTURE</th></tr>
<tr><td colspan="8">MSB (transmitted first)</td><td colspan="24">40 bit</td><td colspan="8">LSB (transmitted last)</td></tr>
<tr><td colspan="20">39 ...</td><td colspan="20">... 0</td></tr>
<tr><td colspan="8"><b>→ 8 bit address</b>
<b>← 8 bit SPI status</b></td><td colspan="32"><b>←→ 32 bit data</b></td></tr>
<tr><td colspan="8">39 ... 32</td><td colspan="32">31 ... 0</td></tr>
<tr><td colspan="8">→ to TMC5031:
RW + 7 bit address
← from TMC5031:
8 bit SPI status</td><td colspan="8">8 bit data</td><td colspan="8">8 bit data</td><td colspan="8">8 bit data</td><td colspan="8">8 bit data</td></tr>
<tr><td colspan="8">39 / 38 ... 32</td><td colspan="8">31 ... 24</td><td colspan="8">23 ... 16</td><td colspan="8">15 ... 8</td><td colspan="8">7 ... 0</td></tr>
<tr><td>w</td><td colspan="7">38...32</td><td colspan="4">31...28</td><td colspan="4">27...24</td><td colspan="4">23...20</td><td colspan="4">19...16</td><td colspan="4">15...12</td><td colspan="4">11...8</td><td colspan="4">7...4</td><td colspan="4">3...0</td></tr>
<tr><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

4.1.1 Selection of Write / Read (WRITE_notREAD)

The read and write selection is controlled by the MSB of the address byte (bit 39 of the SPI datagram). This bit is 0 for read access and 1 for write access. So, the bit named W is a WRITE_notREAD control bit. The active high write bit is the MSB of the address byte. So, 0x80 has to be added to the address for a write access. The SPI interface always delivers data back to the master, independent of the W bit. The data transferred back is the data read from the address which was transmitted with the *previous* datagram, if the previous access was a read access. If the previous access was a write access, then the data read back mirrors the previously received write data. So, the difference between a read and a write access is that the read access does not transfer data to the addressed register but it transfers the address only and its 32 data bits are dummies, and, further the following read or write access delivers back the data read from the address transmitted in the preceding read cycle.

A read access request datagram uses dummy write data. Read data is transferred back to the master with the subsequent read or write access. Hence, reading multiple registers can be done in a pipelined fashion.

Whenever data is read from or written to the TMC5031, the MSBs delivered back contain the SPI status, *SPI_STATUS*, a number of eight selected status bits.

Example:

For a read access to the register (*XACTUAL*) with the address 0x21, the address byte has to be set to 0x21 in the access preceding the read access. For a write access to the register (*VMAX*), the address byte has to be set to 0x80 + 0x27 = 0xA7. For read access, the data bit might have any value (-). So, one can set them to 0.

action	data sent to TMC5031	data received from TMC5031
read *XACTUAL*	→ 0x2100000000	← 0xSS & unused data
read *XACTUAL*	→ 0x2100000000	← 0xSS & XACTUAL
write *VMAX*:= 0x00ABCDEF	→ 0xA700ABCDEF	← 0xSS & XACTUAL
write *VMAX*:= 0x00123456	→ 0xA700123456	← 0xSS00ABCDEF

*)S: is a placeholder for the status bits *SPI_STATUS*

4.1.2 SPI Status Bits Transferred with Each Datagram Read Back

New status information becomes latched at the end of each access and is available with the next SPI transfer.

***SPI_STATUS* – status flags transmitted with each SPI access in bits 39 to 32**		
Bit	**Name**	**Comment**
7	-	reserved (0)
6	*status_stop_l(2)*	*RAMP_STAT2*[0] – 1: Signals motor 2 stop left switch status
5	*status_stop_l(1)*	*RAMP_STAT1*[0] – 1: Signals motor 1 stop left switch status
4	*velocity_reached(2)*	*RAMP_STAT2*[8] – 1: Signals motor 2 has reached its target velocity
3	*velocity_reached(1)*	*RAMP_STAT1*[8] – 1: Signals motor 1 has reached its target velocity
2	*driver_error(2)*	*GSTAT*[2] – 1: Signals driver 2 driver error (clear by reading *GSTAT*)
1	*driver_error(1)*	*GSTAT*[1] – 1: Signals driver 1 driver error (clear by reading *GSTAT*)
0	*reset_flag*	*GSTAT*[0] – 1: Signals, that a reset has occurred (clear by reading *GSTAT*)

4.1.3 Data Alignment

All data are right aligned. Some registers represent unsigned (positive) values, some represent integer values (signed) as two's complement numbers, single bits or groups of bits are represented as single bits respectively as integer groups.

4.2 SPI Signals

The SPI bus on the TMC5031 has four signals:

- SCK – bus clock input
- SDI – serial data input
- SDO – serial data output
- CSN – chip select input (active low)

The slave is enabled for an SPI transaction by a low on the chip select input CSN. Bit transfer is synchronous to the bus clock SCK, with the slave latching the data from SDI on the rising edge of SCK and driving data to SDO following the falling edge. The most significant bit is sent first. A minimum of 40 SCK clock cycles is required for a bus transaction with the TMC5031.

If more than 40 clocks are driven, the additional bits shifted into SDI are shifted out on SDO after a 40-clock delay through an internal shift register. This can be used for daisy chaining multiple chips.

CSN must be low during the whole bus transaction. When CSN goes high, the contents of the internal shift register are latched into the internal control register and recognized as a command from the master to the slave. If more than 40 bits are sent, only the last 40 bits received before the rising edge of CSN are recognized as the command.

4.3 Timing

The SPI interface is synchronized to the internal system clock, which limits the SPI bus clock SCK to half of the system clock frequency. If the system clock is based on the on-chip oscillator, an additional 10% safety margin must be used to ensure reliable data transmission. All SPI inputs as well as the ENN input are internally filtered to avoid triggering on pulses shorter than 20ns. Figure 4.1 shows the timing parameters of an SPI bus transaction, and the table below specifies their values.

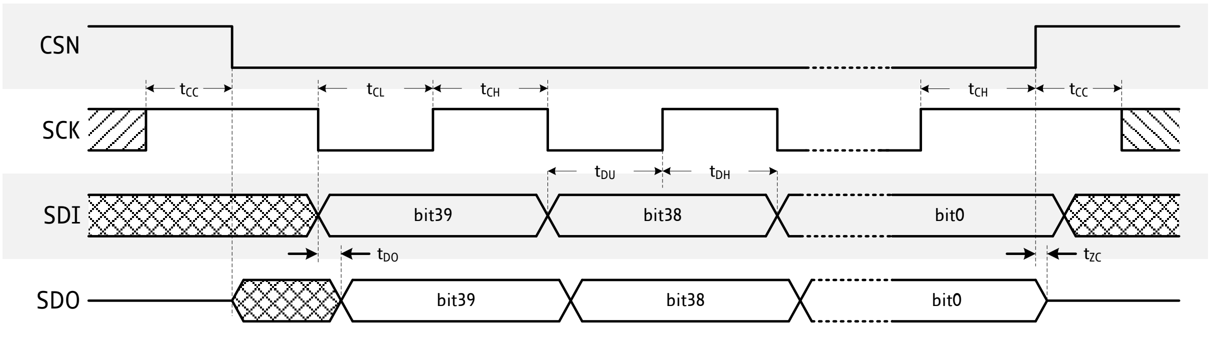


Figure 4.1 SPI timing

Hint
Usually this SPI timing is referred to as SPI MODE 3 (CPOL=1 and CPHA=1).

SPI interface timing	**AC-Characteristics** clock period: t_{CLK}					
Parameter	**Symbol**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
SCK valid before or after change of CSN	t_{CC}		10			ns
CSN high time	t_{CSH}	*) Min time is for synchronous CLK with SCK high one t_{CH} before CSN high only	t_{CLK}*)	>2t_{CLK}+10		ns
SCK low time	t_{CL}	*) Min time is for synchronous CLK only	t_{CLK}*)	>t_{CLK}+10		ns
SCK high time	t_{CH}	*) Min time is for synchronous CLK only	t_{CLK}*)	>t_{CLK}+10		ns
SCK frequency using internal clock	f_{SCK}	assumes minimum OSC frequency			4	MHz
SCK frequency using external 16MHz clock	f_{SCK}	assumes synchronous CLK			8	MHz
SDI setup time before rising edge of SCK	t_{DU}		10			ns
SDI hold time after rising edge of SCK	t_{DH}		10			ns
Data out valid time after falling SCK clock edge	t_{DO}	no capacitive load on SDO			t_{FILT}+5	ns
SDI, SCK and CSN filter delay time	t_{FILT}	rising and falling edge	12	20	30	ns

5 Register Mapping

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

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

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

NOTATION OF R/W FIELD	
R	Read only
W	Write only
R/W	Read- and writable register
R+C	Clear upon read (i.e. status bit becomes reset after readout)

OVERVIEW REGISTER MAPPING

REGISTER	**DESCRIPTION**
General Configuration Registers	These registers contain - global configuration - global status flags
Ramp Generator Motion Control Register Set	This register set offers registers for - choosing a ramp mode - choosing velocities - homing - acceleration and deceleration - target positioning
Ramp Generator Driver Feature Control Register Set	This register set offers registers for - driver current control - setting thresholds for coolStep operation - setting thresholds for different chopper modes - a reference switch and stallGuard2 event configuration - a ramp and reference switch status register
Motor Driver Register Set	This register set offers registers for - setting / reading out microstep table and counter - chopper and driver configuration - coolStep and stallGuard2 configuration - reading out stallGuard2 values and driver error flags

5.1 General Configuration Registers

GENERAL CONFIGURATION REGISTERS (0X00…0X1F)					
R/W	**Addr**	**n**	***Register***	**Description / *bit names***	
RW	0x00	11	*GCONF*	**Bit**	***GCONF* – Global configuration flags**
				0..2	Reserved, set to 0
				3	*poscmp_enable* 0: Outputs INT and PP are tristated. 1: Position compare pulse (PP) and interrupt output (INT) are available *Attention – do not leave the outputs floating in tristate condition, provide an external pull-up or set this bit 1.*
				4..6	Reserved, set to 0
				7	*test_mode* 0: Normal operation 1: Enable analog test output on pin REFR2 *TEST_SEL* selects the function of REFR2: 0…4: T120, DAC1, VDDH1, DAC2, VDDH2 *Attention: Not for user, set to 0 for normal operation!*
				8	*shaft1* 1: Inverse motor 1 direction
				9	*shaft2* 1: Inverse motor 2 direction
				10	*lock_gconf* 1: GCONF is locked against further write access.
R+C	0x01	4	*GSTAT*	**Bit**	***GSTAT* – Global status flags**
				0	*reset* 1: Indicates that the IC has been reset since the last read access to *GSTAT.*
				1	*drv_err1* 1: Indicates, that driver 1 has been shut down due to overtemperature or short circuit detection since the last read access. Read *DRV_STATUS1* for details. The flag can only be reset when all error conditions are cleared.
				2	*drv_err2* *1:* Indicates, that driver 2 has been shut down due to overtemperature or short circuit detection since the last read access. Read *DRV_STATUS2* for details. The flag can only be reset when all error conditions are cleared.
				3	*uv_cp* 1: Indicates an undervoltage on the charge pump. The driver is disabled in this case.
W	0x03	4	*TEST_SEL*	**Bit**	***SLAVECONF***
				3..0	*TEST_SEL*: selects the function of REFR2 in test mode: 0…4: T120, DAC1, VDDH1, DAC2, VDDH2 *Attention: Not for user, set to 0 for normal operation!*
R	0x04	8 + 8	*INPUT*	**Bit**	***INPUT***
				0..6	Unused, ignore these bits
				7	Reads the state of the DRV_ENN pin
				31.. 24	*VERSION*: 0x01=first version of the IC Identical numbers mean full digital compatibility.

GENERAL CONFIGURATION REGISTERS (0X00…0X1F)				
R/W	**Addr**	**n**	***Register***	**Description / *bit names***
W	0x05	32	*X_COMPARE*	Position comparison register for motor 1 position strobe. Activate *poscmp_enable* to get position pulse on output PP. *XACTUAL* = *X_COMPARE*: - Output PP becomes high. It returns to a low state, if the positions mismatch.

5.2 Ramp Generator Registers

*Addresses **Addr** are specified for motor 1 (upper value) and motor 2 (second address).*

5.2.1 Ramp Generator Motion Control Register Set

RAMP GENERATOR MOTION CONTROL REGISTER SET (MOTOR 1: 0X20…0X2D, MOTOR 2: 0X40…0X4D)					
R/W	**Addr**	**n**	***Register***	**Description / *bit names***	**Range [Unit]**
RW	0x20 0x40	2	*RAMPMODE*	*RAMPMODE:* 0: Positioning mode (using all A, D and V parameters) 1: Velocity mode to positive *VMAX* (using *AMAX* acceleration) 2: Velocity mode to negative *VMAX* (using *AMAX* acceleration) 3: Hold mode (velocity remains unchanged, unless stop event occurs)	0…3
RW	0x21 0x41	32	*XACTUAL*	Actual motor position (signed) *Hint:* This value normally should only be modified, when homing the drive. In positioning mode, modifying the register content will start a motion.	-2^31… +(2^31)-1
R	0x22 0x42	24	*VACTUAL*	Actual motor velocity from ramp generator (signed) The sign matches the motion direction. A negative sign means motion to lower XACTUAL.	+-(2^23)-1 [µsteps / t]
W	0x23 0x43	18	*VSTART*	Motor start velocity (unsigned) Set *VSTOP ≥ VSTART!*	0…(2^18)-1 [µsteps / t]
W	0x24 0x44	16	*A1*	First acceleration between *VSTART* and *V1* (unsigned)	0…(2^16)-1 [µsteps / ta²]
W	0x25 0x45	20	*V1*	First acceleration / deceleration phase target velocity (unsigned) 0: Disables *A1* and *D1* phase, use *AMAX*, *DMAX* only	0…(2^20)-1 [µsteps / t]
W	0x26 0x46	16	*AMAX*	Second acceleration between *V1* and *VMAX* (unsigned) This is the acceleration and deceleration value for velocity mode.	0…(2^16)-1 [µsteps / ta²]

RAMP GENERATOR MOTION CONTROL REGISTER SET (MOTOR 1: 0X20…0X2D, MOTOR 2: 0X40…0X4D)

R/W	Addr	n	*Register*	Description / *bit names*	Range [Unit]
W	0x27 0x47	23	*VMAX*	Motion ramp target velocity (for positioning ensure *VMAX* ≥ *VSTART*) (unsigned) This is the target velocity in velocity mode. It can be changed any time during a motion.	0…(2^23)-512 [µsteps / t]
W	0x28 0x48	16	*DMAX*	Deceleration between *VMAX* and *V1* (unsigned)	0…(2^16)-1 [µsteps / ta^2]
W	0x2A 0x4A	16	*D1*	Deceleration between *V1* and *VSTOP* (unsigned) *Attention: Do not set 0 in positioning mode, even if V1=0!*	1…(2^16)-1 [µsteps / ta^2]
W	0x2B 0x4B	18	*VSTOP*	Motor stop velocity (unsigned) *Attention: Set VSTOP ≥ VSTART!* *Attention: Do not set 0 in positioning mode!*	1…(2^18)-1 [µsteps / t]
W	0x2C 0x4C	16	*TZEROWAIT*	Waiting time after ramping down to zero velocity before next movement or direction inversion can start and before motor power down starts. Time range is about 0 to 2 seconds. This setting avoids excess acceleration e.g. from *VSTOP* to -*VSTART*.	0…(2^16)-1 * 512 t_{CLK}
RW	0x2D 0x4D	32	*XTARGET*	Target position for ramp mode (signed). Write a new target position to this register in order to activate the ramp generator positioning in *RAMPMODE*=0. Initialize all velocity, acceleration and deceleration parameters before. *Hint:* The position is allowed to wrap around, thus, *XTARGET* value optionally can be treated as an unsigned number. *Hint:* The maximum possible displacement is +/-((2^31)-1). *Hint:* When increasing *V1, D1 or DMAX during* a motion, rewrite *XTARGET* afterwards in order to trigger a second acceleration phase, if desired.	-2^31… +(2^31)-1

5.2.2 Ramp Generator Driver Feature Control Register Set

RAMP GENERATOR DRIVER FEATURE CONTROL REGISTER SET (MOTOR 1: 0X30…0X36, MOTOR 2: 0X50…0X56)

R/W	Addr	n	*Register*	Description / *bit names*	
W	0x30 0x50	5 + 5 + 4	*IHOLD_IRUN*	**Bit**	***IHOLD_IRUN* – Driver current control**
				4..0	*IHOLD* Standstill current (0=1/32…31=32/32)
				12..8	*IRUN* Motor run current (0=1/32…31=32/32) *Hint:* Choose sense resistors in a way, that normal *IRUN* is 16 to 31 for best microstep performance.
				19..16	*IHOLDDELAY* Controls the number of clock cycles for motor power down after a motion as soon as *TZEROWAIT* 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	0x31 0x51	23	*VCOOLTHRS*	This is the lower threshold velocity for switching on smart energy coolStep. (unsigned) Set this parameter to disable coolStep at low speeds, where it cannot work reliably. *VHIGH* ≥ |*VACT*| ≥ VCOOLTHRS: - coolStep is enabled, if configured (Only bits 22..8 are used for value and for comparison)	
W	0x32 0x52	23	*VHIGH*	This velocity setting allows velocity dependent switching into a different chopper mode and fullstepping to maximize torque. (unsigned) |*VACT*| ≥ *VHIGH*: - coolStep is disabled (motor runs with normal current scale) - If *vhighchm* is set, the chopper switches to *chm*=1 with *TFD*=0 (constant off time with slow decay, only). - chopSync2 is switched off (*SYNC*=0) - If *vhighfs* is set, the motor operates in fullstep mode. (Only bits 22..8 are used for value and for comparison)	
RW	0x34 0x54	11	***SW_MODE***	Switch mode configuration *See separate table!*	
R+C	0x35 0x55	14	***RAMP_STAT***	Ramp status and switch event status *See separate table!*	
R	0x36 0x56	32	*XLATCH*	Ramp generator latch position, latches *XACTUAL* upon a programmable switch event (see *SW_MODE*).	

time reference t for velocities: $t = 2^{24} / f_{CLK}$

time reference ta² for accelerations: $ta^2 = 2^{41} / (f_{CLK})^2$

6.2.2.1 *SW_MODE* – Reference Switch and stallGuard2 Event Configuration Register

0x34, 0x54: *SW_MODE* – REFERENCE SWITCH AND STALLGUARD2 EVENT CONFIGURATION REGISTER		
Bit	**Name**	**Comment**
11	*en_softstop*	0: Hard stop 1: Soft stop The soft stop mode always uses the deceleration ramp settings *DMAX*, V1, *D1*, *VSTOP* and *TZEROWAIT* for stopping the motor. A stop occurs when the velocity sign matches the reference switch position (REFL for negative velocities, REFR for positive velocities) and the respective switch stop function is enabled. A hard stop also uses *TZEROWAIT* before the motor becomes released. *Attention: Do not use soft stop in combination with stallGuard2.*
10	*sg_stop*	1: Enable stop by stallGuard2. Disable to release motor after stop event. *Attention: Do not enable during motor spin-up, wait until the motor velocity exceeds a certain value, where stallGuard2 delivers a stable result.*
9	-	Reserved, set to 0
8	*latch_r_inactive*	1: Activates latching of the position to *XLATCH* upon an inactive going edge on the right reference switch input REFR. The active level is defined by *pol_stop_r*.
7	*latch_r_active*	1: Activates latching of the position to *XLATCH* upon an active going edge on the right reference switch input REFR. *Hint:* Activate *latch_r_active* to detect any spurious stop event by reading *status_latch_r*.
6	*latch_l_inactive*	1: Activates latching of the position to *XLATCH* upon an inactive going edge on the left reference switch input REFL. The active level is defined by *pol_stop_l*.
5	*latch_l_active*	1: Activates latching of the position to *XLATCH* upon an active going edge on the left reference switch input REFL. *Hint:* Activate *latch_l_active* to detect any spurious stop event by reading *status_latch_l*.
4	*swap_lr*	1: Swap the left and the right reference switch input
3	*pol_stop_r*	Sets the active polarity of the right reference switch input 0=non-inverted, high active: a high level on REFR stops the motor 1=inverted, low active: a low level on REFR stops the motor
2	*pol_stop_l*	Sets the active polarity of the left reference switch input 0=non-inverted, high active: a high level on REFL stops the motor 1=inverted, low active: a low level on REFL stops the motor
1	*stop_r_enable*	1: Enables automatic motor stop during active right reference switch input *Hint:* The motor restarts in case the stop switch becomes released.
0	*stop_l_enable*	1: Enables automatic motor stop during active left reference switch input *Hint:* The motor restarts in case the stop switch becomes released.

6.2.2.2 *RAMP_STAT* – Ramp and Reference Switch Status Register

0x35, 0x55: *RAMP_STAT* – RAMP AND REFERENCE SWITCH STATUS REGISTER

R/W	Bit	Name	Comment
R	13	*status_sg*	1: Signals an active stallGuard2 input from the coolStep driver, if enabled. *Hint:* When polling this flag, stall events may be missed – activate *sg_stop* to be sure not to miss the stall event.
R+C	12	*second_move*	1: Signals that the automatic ramp requires moving back in the opposite direction, e.g. due to on-the-fly parameter change (Flag is cleared upon reading)
R	11	*t_zerowait_active*	1: Signals, that *TZEROWAIT* is active after a motor stop. During this time, the motor is in standstill.
R	10	*vzero*	1: Signals, that the actual velocity is 0.
R	9	*position_reached*	1: Signals, that the target position is reached. This flag becomes set while *XACTUAL* and *XTARGET* match.
R	8	*velocity_reached*	1: Signals, that the target velocity is reached. This flag becomes set while *VACTUAL* and *VMAX* match.
R+C	7	*event_pos_reached*	1: Signals, that the target position has been reached (*position_reached* becoming active). (Flag and interrupt condition are cleared upon reading) This bit is ORed to the *interrupt output* signal.
R+C	6	*event_stop_sg*	1: Signals an active StallGuard2 stop event. Reading the register will clear the stall condition and the motor may re-start motion, unless the motion controller has been stopped. (Flag and interrupt condition are cleared upon reading) This bit is ORed to the *interrupt output* signal.
R	5	*event_stop_r*	1: Signals an active stop right condition due to stop switch. The stop condition and the interrupt condition can be removed by setting *RAMP_MODE* to hold mode or by commanding a move to the opposite direction. In *soft_stop* mode, the condition will remain active until the motor has stopped motion into the direction of the stop switch. Disabling the stop switch or the stop function also clears the flag, but the motor will continue motion. This bit is ORed to the *interrupt output* signal.
	4	*event_stop_l*	1: Signals an active stop left condition due to stop switch. The stop condition and the interrupt condition can be removed by setting *RAMP_MODE* to hold mode or by commanding a move to the opposite direction. In *soft_stop* mode, the condition will remain active until the motor has stopped motion into the direction of the stop switch. Disabling the stop switch or the stop function also clears the flag, but the motor will continue motion. This bit is ORed to the *interrupt output* signal.
R+C	3	*status_latch_r*	1: Latch right ready (enable position latching using *SWITCH_MODE* settings *latch_r_active* or *latch_r_inactive*) (Flag is cleared upon reading)
	2	*status_latch_l*	1: Latch left ready (enable position latching using *SWITCH_MODE* settings *latch_l_active* or *latch_l_inactive*) (Flag is cleared upon reading)
R	1	*status_stop_r*	Reference switch right status (1=active)
	0	*status_stop_l*	Reference switch left status (1=active)