



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



## TMC4331A DATASHEET

TMC4331A Document Revision 1.01 • 2016-NOV-25

The S-ramp and sixPoint™ ramp motion controller for stepper motors is optimized for high velocities, allowing on-the-fly changes. TMC4331A offers SPI and Step/Dir interfaces.



Figure 1: Sample Image TMC4331A

\*Marking details are explained on page 172.

### Features

- SPI Interfaces for  $\mu$ C with easy-to-use protocol.
- SPI Interfaces for SPI motor stepper drivers.
- Integrated ChopSync™ and dcStep™ support.
- Internal ramp generator generating S-shaped ramps or sixPoint™ ramps supporting on-the-fly changes.
- Controlled PWM output.
- Reference switch handling.
- Hardware and virtual stop switches.
- Extensive Support of TMC stepper motor drivers.
- Electronic gearing support.

### Applications

- Textile, sewing machines
- Office automation
- Pumps and valves
- CCTV, security
- POS
- Heliostat controllers
- Printers, scanners
- Factory automation
- CNC machines
- ATM, cash recycler
- Lab automation
- Robotics

### Block Diagram: TMC4331A Interfaces & Features

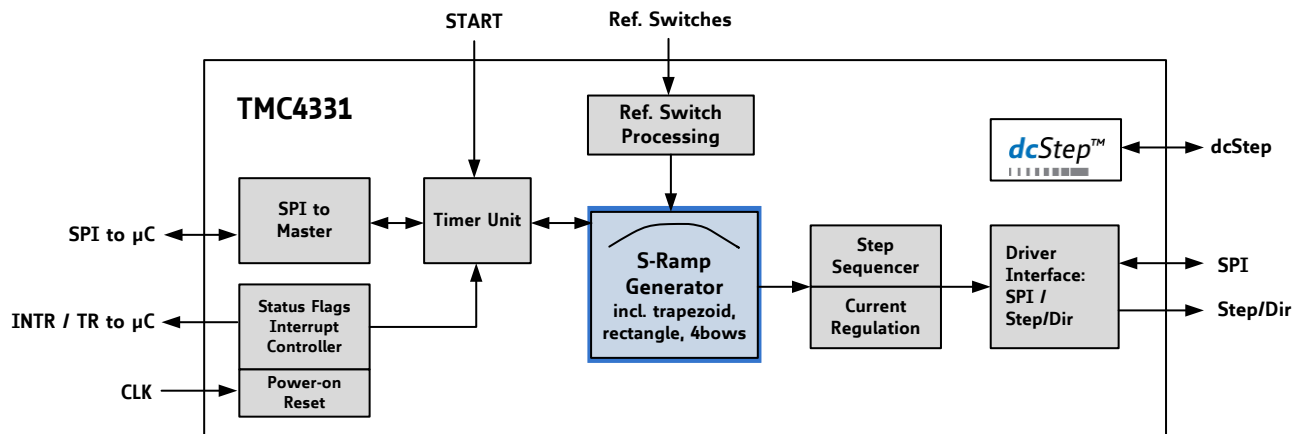


Figure 2: Block Diagram

© 2015 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany — Terms of delivery and rights to technical change reserved. Download newest version at: [www.trinamic.com](http://www.trinamic.com).



Read entire documentation; especially the Supplemental Directives in chapter 18 (page 173).

## Functional Scope of TMC4331A

**TMC4331A is a miniaturized high-performance motion controller for stepper motor drivers, particularly designed for fast and jerk-limited motion profile applications with a wide range of ramp profiles. The S-shaped or sixPoint™ velocity profile, and open-loop features offer many configuration options to suit the user’s specifications, as presented below:**

### S-Shaped Velocity Profile

S-shaped ramp profiles are jerk-free. Seven ramp segments form the S-shaped ramp that can be optimally adapted to suit the user’s requirements. High torque with high velocities can be reached by calibrating the bows of the ramp, as explained in this user manual.

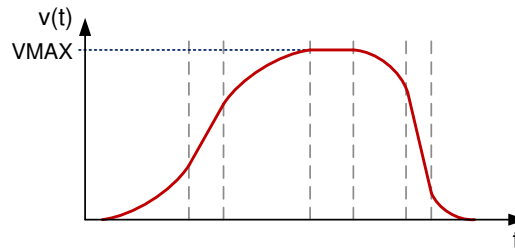


Figure 3: S-shaped Velocity Profile

- i More information on ramp configurations and other velocity profiles, e.g. sixPoint™ ramps, are provided in chapter 6 (Page 24).

### Reference Switch Support

A typical hardware setup for open-loop operation with enhanced modifications, by use of external stop switches with the TMC26x motor stepper driver is shown below. Home switches with different configurations are also supported.

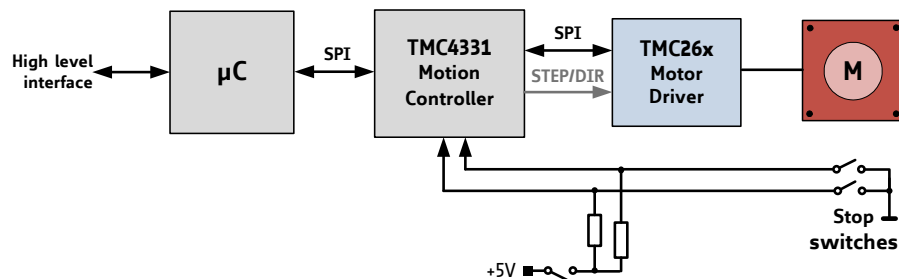


Figure 4: Open-Loop Hardware Set-up with TMC26x supporting External Stop Switches

### Open-loop Operation with dcStep™ Feature

A typical hardware setup for dcStep operation with a TMC2130 stepper motor driver is shown in the diagram below. This feature is also available for TMC26x stepper motor drivers.

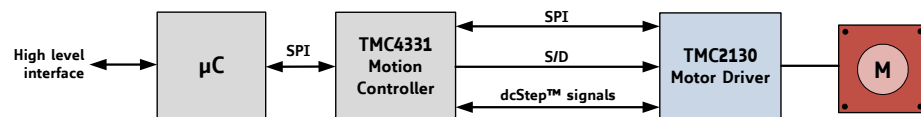


Figure 5: Hardware Set-up for Open-loop Operation with TMC2130

## Order Codes

Order code	Description	Size
TMC4331A-LA	Motion controller with dcStep features, QFN32	4 x 4 mm <sup>2</sup>

Table 1: TMC4331A Order Codes



# TABLE OF CONTENTS

<b>TMC4331A DATASHEET</b> .....	<b>1</b>
<b>SHORT SPEC</b> .....	<b>1</b>
<b>Features</b> .....	<b>1</b>
<b>Applications</b> .....	<b>1</b>
<b>Block Diagram: TMC4331A Interfaces &amp; Features</b> .....	<b>1</b>
<b>Functional Scope of TMC4331A</b> .....	<b>2</b>
<b>Order Codes</b> .....	<b>2</b>
<b>TABLE OF CONTENTS</b> .....	<b>3</b>
<b>MAIN MANUAL</b> .....	<b>8</b>
<b>1. Pinning and Design-In Process Information</b> .....	<b>8</b>
1.1. Pin Assignment: Top View .....	8
1.2. Pin Description .....	9
1.3. System Overview .....	10
<b>2. Application Circuits</b> .....	<b>11</b>
2.1. TMC4331A Standard Connection: VCC=3.3V .....	11
2.2. TMC4331A with TMC26x Stepper Connection.....	11
2.3. TMC4331A with TMC248 Stepper Driver .....	12
2.4. TMC4331A with TMC2130 Stepper Driver .....	12
<b>3. SPI Interfacing</b> .....	<b>13</b>
3.1. SPI Datagram Structure .....	13
3.1.1. SPI Timing Description .....	16
<b>4. Input Filtering</b> .....	<b>17</b>
4.1. Input Filtering Examples.....	19
<b>5. Status Flags and Events</b> .....	<b>20</b>
5.1. Status Event Description .....	21
5.2. SPI Status Bit Transfer .....	22
5.3. Generation of Interrupts.....	22
5.4. Connection of Multiple INTR Pins .....	23
<b>6. Ramp Configurations for different Motion Profiles</b> .....	<b>24</b>
6.1. Step/Dir Output Configuration .....	25
6.1.1. Step/Dir Output Configuration Steps .....	25
6.1.2. STPOUT: Changing Polarity .....	25
6.2. Altering the Internal Motion Direction.....	26
6.3. Configuration Details for Operation Modes and Motion Profiles .....	27
6.3.1. Starting Point: Choose Operation Mode .....	28
6.3.2. Stop during Motion .....	28
6.3.3. Motion Profile Configuration .....	29
6.3.4. No Ramp Motion Profile.....	30
6.3.5. Trapezoidal 4-Point Ramp without Break Point.....	31
6.3.6. Trapezoidal Ramp with Break Point.....	31
6.3.7. Position Mode combined with Trapezoidal Ramps .....	32



6.3.8.	Configuration of S-Shaped Ramps.....	33
6.3.9.	Changing Ramp Parameters during S-shaped Motion or Switching to Positioning Mode.....	34
6.3.10.	Configuration of S-shaped Ramp with <i>ASTART</i> and <i>DFINAL</i> .....	34
6.3.11.	S-shaped Mode and Positioning: Fast Motion.....	35
6.4.	Start Velocity <i>VSTART</i> and Stop Velocity <i>VSTOP</i> .....	36
6.4.1.	S-shaped Ramps with Start and Stop Velocity.....	40
6.4.2.	Combined Use of <i>VSTART</i> and <i>ASTART</i> for <i>S-shaped Ramps</i> .....	41
6.5.	sixPoint Ramps.....	42
6.6.	U-Turn Behavior.....	43
6.6.1.	Continuous Velocity Motion Profile for S-shaped Ramps.....	44
6.7.	Internal Ramp Generator Units.....	45
6.7.1.	Clock Frequency.....	45
6.7.2.	Velocity Value Units.....	45
6.7.3.	Acceleration Value Units.....	45
6.7.4.	Bow Value Units.....	46
6.7.5.	Overview of Minimum and Maximum Values:.....	46
<b>7.</b>	<b>External Step Control and Electronic Gearing.....</b>	<b>47</b>
7.1.	Description of Electronic Gearing.....	48
7.2.	Indirect External Control.....	48
7.3.	Switching from External to Internal Control.....	49
<b>8.</b>	<b>Reference Switches.....</b>	<b>50</b>
8.1.	Hardware Switch Support.....	51
8.1.1.	Stop Slope Configuration for Hard or Linear Stop Slopes.....	51
8.1.2.	How Active Stops are indicated and reset to Free Motion.....	52
8.1.3.	How to latch Internal Position on Switch Events.....	52
8.2.	Virtual Stop Switches.....	53
8.2.1.	Enabling Virtual Stop Switches.....	53
8.2.2.	Virtual Stop Slope Configuration.....	53
8.2.3.	How Active Virtual Stops are indicated and reset to Free Motion.....	54
8.3.	Home Reference Configuration.....	55
8.3.1.	Home Event Selection.....	55
8.3.2.	HOME_REF Monitoring.....	56
8.3.3.	Homing with STOPL or STOPR.....	56
8.4.	Target Reached / Position Comparison.....	57
8.4.1.	Connecting several Target-reached Pins.....	57
8.4.2.	Use of TARGET_REACHED Output.....	58
8.4.3.	Position Comparison of Internal Values.....	58
8.5.	Repetitive and Circular Motion.....	59
8.5.1.	Repetitive Motion to XTARGET.....	59
8.5.2.	Activating Circular Motion.....	59
8.5.3.	Uneven or Noninteger Microsteps per Revolution.....	60
8.5.4.	Release of the Revolution Counter.....	61
8.6.	Blocking Zones.....	61
8.6.1.	Activating Blocking Zones during Circular Motion.....	61
8.6.2.	Circular Motion with and without Blocking Zone.....	62
<b>9.</b>	<b>Ramp Timing and Synchronization.....</b>	<b>63</b>
9.1.	Basic Synchronization Settings.....	64
9.1.1.	Start Signal Trigger Selection.....	64



9.1.2.	User-specified Impact Configuration of Timing Procedure.....	64
9.1.3.	Delay Definition between Trigger and internally generated Start Signal .....	65
9.1.4.	Active START Pin Output Configuration .....	65
9.1.5.	Ramp Timing Examples.....	66
9.2.	Shadow Register Settings.....	69
9.2.1.	Shadow Register Configuration Options.....	70
9.2.2.	Delayed Shadow Transfer.....	74
9.3.	Pipelining Internal Parameters .....	75
9.3.1.	Configuration and Activation of Target Pipeline.....	75
9.3.2.	Using the Pipeline for different internal Registers.....	76
9.3.3.	Pipeline Mapping Overview .....	77
9.3.4.	Cyclic Pipelining .....	78
9.3.5.	Pipeline Examples.....	78
9.4.	Masterless Synchronization of Several Motion Controllers via START Pin.....	80
<b>10.</b>	<b>Serial Data Output .....</b>	<b>81</b>
10.1.	Getting Started with TMC Motor Drivers .....	82
10.2.	Sine Wave Lookup Tables.....	83
10.2.1.	Actual Current Values Output .....	84
10.2.2.	How to Program the Internal MSLUT.....	84
10.2.3.	Setup of MSLUT Segments .....	85
10.2.4.	Current Waves Start Values.....	86
10.2.5.	Default MSLUT .....	86
10.2.6.	Explanatory Notes for Base Wave Inclinations .....	87
10.3.	SPI Output Interface Configuration Parameters .....	89
10.3.1.	Pins dedicated to SPI Output Communication .....	89
10.3.2.	Setup of SPI Output Timing Configuration .....	89
10.3.3.	Current Diagrams .....	90
10.3.4.	Change of Microstep Resolution.....	90
10.3.5.	Cover Datagrams Communication between $\mu$ C and Driver .....	90
10.3.6.	Sending Cover Datagrams.....	91
10.3.7.	Configuring Automatic Generation of Cover Datagrams .....	92
10.4.	Overview: TMC Motor Driver Connections.....	93
10.4.1.	TMC Stepper Motor Driver Settings .....	93
10.4.2.	TMC Motor Driver Response Datagram and Status Bits.....	94
10.4.3.	Events and Interrupts based on Motor Driver Status Bits.....	94
10.4.4.	Stall Detection and Stop-on-Stall.....	95
10.5.	TMC23x, TMC24x Stepper Motor Driver.....	96
10.5.1.	TMC23x Setup .....	96
10.5.2.	TMC24x Setup .....	96
10.5.3.	TMC23x/24x Status Bits .....	97
10.5.4.	Automatic Fullstep Switchover for TMC23x/24x.....	97
10.5.5.	Mixed Decay Configuration for TMC23x/24x .....	98
10.5.6.	ChopSync Configuration for TMC23x/24x Stepper Drivers.....	98
10.5.7.	Doubling ChopSync Frequency during Standstill.....	98
10.5.8.	Using TMC24x stallGuard Characteristics .....	99
10.6.	TMC26x Stepper Motor Driver.....	100
10.6.1.	TMC26x Setup (SPI mode) .....	100
10.6.2.	TMC26x Setup (S/D mode).....	100
10.6.3.	Sending Cover Datagrams to TMC26x .....	101



10.6.4.	Automatic Continuous Streaming of Cover Datagrams for TMC26x.....	101
10.6.5.	TMC26x SPI Mode: Automatic Fullstep Switchover .....	102
10.6.6.	TMC26x S/D Mode: Automatic Fullstep Switchover.....	102
10.6.7.	TMC 26x S/D Mode: Change of Current Scaling Parameter .....	103
10.6.8.	TMC26x Status Bits.....	103
10.6.9.	TMC26x Status Response .....	103
10.7.	TMC389 Stepper Motor Driver .....	104
10.8.	TMC2130 Stepper Motor Driver.....	105
10.8.1.	Set-up TMC2130 Support (SPI Mode).....	105
10.8.2.	Set-up TMC2130 Support (S/D Mode) .....	105
10.8.3.	Sending Cover Datagrams to TMC2130 .....	106
10.8.4.	Automatic Continuous Streaming of Cover Datagrams for TMC2130.....	106
10.8.5.	TMC2130 SPI Mode: Automatic Fullstep Switchover .....	107
10.8.6.	TMC2130 S/D Mode: Automatic Fullstep Switchover.....	107
10.8.7.	TMC 2130 S/D Mode: Changing current Scaling Parameter.....	107
10.8.8.	TMC2130 Status Response .....	108
10.9.	Connecting Non-TMC Stepper Motor Driver or SPI-DAC at SPI output interface .....	109
10.9.1.	Connecting a SPI-DAC.....	110
10.9.2.	DAC Data Transfer.....	110
10.9.3.	Changing SPI Output Protocol for SPI-DAC.....	110
10.9.4.	DAC Address Values.....	111
10.9.5.	DAC Data Values .....	111
<b>11.</b>	<b>Current Scaling .....</b>	<b>113</b>
11.1.	Hold Current Scaling .....	114
11.2.	Freewheeling.....	114
11.3.	Current Scaling during Motion .....	115
11.3.1.	Drive Scaling .....	115
11.3.2.	Alternative Drive Scaling .....	115
11.3.3.	Boost Current.....	116
11.4.	Scale Mode Transition Process Control .....	117
11.5.	Current Scaling Examples.....	119
<b>12.</b>	<b>Controlled PWM Output .....</b>	<b>121</b>
12.1.	PWM Output Generation and Scaling Possibilities .....	122
12.1.1.	PWM Scale Example.....	123
12.2.	PWM Output Generation for TMC23x/24x .....	124
12.3.	Switching between SPI and Voltage PWM Modes .....	125
<b>13.</b>	<b>dcStep Support for TMC26x or TMC2130 .....</b>	<b>126</b>
13.1.	Enabling dcStep for TMC26x Stepper Motor Drivers .....	128
13.2.	Setup: Minimum dcStep Velocity.....	129
13.3.	Enabling dcStep for TMC2130 Stepper Motor Drivers .....	131
<b>14.</b>	<b>Reset and Clock Gating .....</b>	<b>132</b>
14.1.	Power-On-Reset .....	132
14.2.	Manual Software Reset .....	132
14.3.	Reset Indication .....	132
14.4.	Activating Clock Gating manually .....	133
14.5.	Clock Gating Wake-up.....	133
14.6.	Automatic Clock Gating Procedure .....	134



<b>TECHNICAL SPECIFICATIONS</b> .....	<b>135</b>
<b>15. Complete Register and Switches List</b> .....	<b>135</b>
15.1. General Configuration Register GENERAL_CONF 0x00 .....	135
15.2. Reference Switch Configuration Register REFERENCE_CONF 0x01 .....	138
15.3. Start Switch Configuration Register START_CONF 0x02.....	141
15.4. Input Filter Configuration Register INPUT_FILT_CONF 0x03 .....	143
15.5. SPI Output Configuration Register SPI_OUT_CONF 0x04.....	144
15.6. Current Scaling Configuration Register CURRENT_CONF 0x05 .....	147
15.7. Current Scale Values Register SCALE_VALUES 0x06 .....	148
15.8. Various Scaling Configuration Registers.....	148
15.9. Motor Driver Settings Register STEP_CONF 0x0A .....	149
15.10. Event Selection Registers 0x0B..0X0D .....	150
15.11. Status Event Register (0x0E).....	151
15.12. Status Flag Register (0x0F) .....	152
15.13. Various Configuration Registers .....	153
15.14. PWM Configuration Registers.....	154
15.15. Ramp Generator Registers.....	155
15.16. External Clock Frequency Register .....	159
15.17. Target and Compare Registers .....	159
15.18. Pipeline Registers .....	160
15.19. Shadow Register.....	160
15.20. Reset and Clock Gating Register .....	161
15.21. dcStep Registers.....	161
15.22. Transfer Registers .....	162
15.23. SinLUT Registers .....	163
15.24. SPI-DAC Configuration Registers.....	164
15.25. TMC Version Register .....	164
<b>16. Absolute Maximum Ratings</b> .....	<b>165</b>
<b>17. Electrical Characteristics</b> .....	<b>166</b>
17.1. Power Dissipation .....	166
17.2. General IO Timing Parameters.....	167
17.3. Layout Examples .....	168
17.3.1. Internal Circuit Diagram for Layout Example.....	168
17.3.2. Top Layer: Assembly Side .....	169
17.3.3. Inner Layer (GND).....	169
17.3.4. Inner Layer (Supply VS) .....	170
17.4. Package Dimensions .....	171
17.5. Package Material Information .....	172
17.6. Marking Details provided on Single Chip.....	172
<b>APPENDICES</b> .....	<b>173</b>
<b>18. Supplemental Directives</b> .....	<b>173</b>
ESD-DEVICE INSTRUCTIONS.....	173
<b>19. Tables Index</b> .....	<b>175</b>
<b>20. Figures Index</b> .....	<b>177</b>
<b>21. Revision History</b> .....	<b>179</b>





# MAIN MANUAL

## 1. Pinning and Design-In Process Information

In this chapter you are provided with a list of all pin names and a functional description of each.

### 1.1. Pin Assignment: Top View

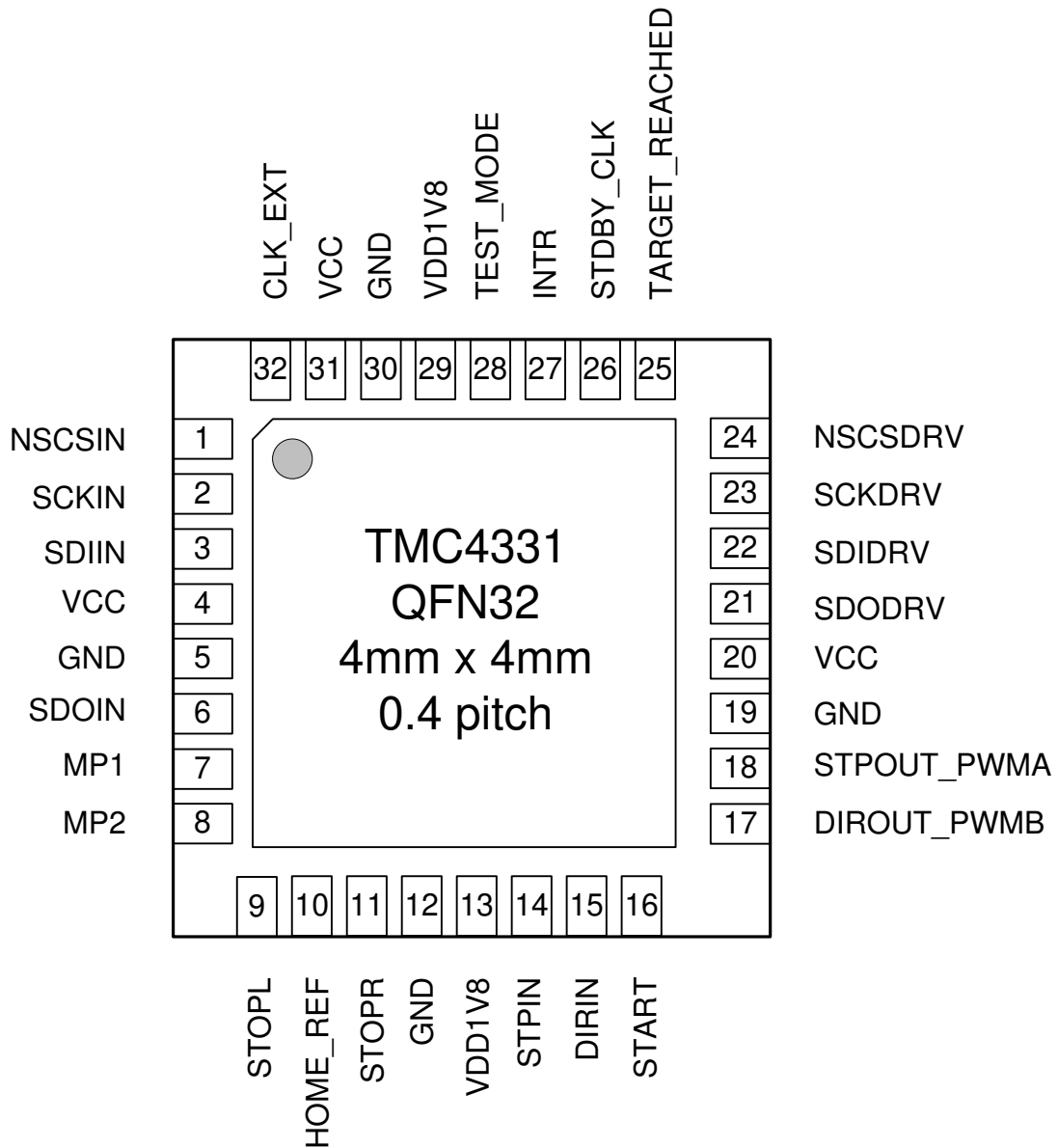


Figure 6: Package Outline: Pin Assignments Top View



## 1.2. Pin Description

Pin Names and Descriptions			
Pin	Number	Type	Function
<i>Supply Pins</i>			
GND	5, 12, 19, 30	GND	Digital ground pin for IOs and digital circuitry.
VCC	4, 20, 31	VCC	Digital power supply for IOs and digital circuitry (3.3V... 5V).
VDD1V8	13, 29	VDD	Connection of internal generated core voltage of 1.8V.
CLK_EXT	32	I	Clock input to provide a clock with the frequency fCLK for all internal operations.
TEST_MODE	28	I	Test mode input. Tie to low for normal operation.
<i>Interface Pins for <math>\mu</math>C</i>			
NSCSIN	1	I	Low active chip selects input of SPI interface to $\mu$ C.
SCKIN	2	I	Serial clock for SPI interface to $\mu$ C.
SDIIN	3	I	Serial data input of SPI interface to $\mu$ C.
SDOIN	6	O	Serial data output of SPI interface to $\mu$ C (Z if NSCSIN=1).
INTR	27	O	Interrupt output, programmable PD/PU for wired-and/or.
TARGET_REACHED	25	O	Target reached output, programmable PD/PU for wired-and/or.
<i>Reference Pins</i>			
STOPL	9	I (PD)	Left stop switch. External signal to stop a ramp. If not connected, internal pull-down resistor is active.
HOME_REF	10	I (PD)	Home reference signal input. External signal for reference search. If not connected, internal pull-down resistor is active.
STOPR	11	I (PD)	Right stop switch. External signal to stop a ramp. If not connected, internal pull-down resistor is active.
STPIN	14	I (PD)	Step input for external step control. If not connected, internal pull-down resistor is active.
DIRIN	15	I (PD)	Direction input for external step control. If not connected, internal pull-down resistor is active.
START	16	IO	Start signal input/output.
<i>S/D Output Pins</i>			
STPOUT PWMA DACA	18	O	Step output. First PWM signal (Sine). First DAC output signal (Sine).
DIROUT PWMB DACB	17	O	Direction output. Second PWM signal (Cosine). Second DAC output signal (Cosine).
•→ <i>Continued on next page!</i>			



Pin Names and Descriptions			
Pin	Number	Type	Function
<i>Interface Pins for Stepper Motor Drivers</i>			
NSCSDRV PWMB	24	O	Low active chip selects output of SPI interface to motor driver. Second PWM signal (Cosine) to connect with PHB (TMC23x/24x).
SCKDRV MDBN	23	O	Serial clock output of SPI interface to motor driver. MDBN output signal for MDBN pin of TMC23x/24x.
SDODRV PWMA	21	O	Serial data output of SPI interface to motor driver. First PWM signal (Sine) to connect with PHA (TMC23x/24x).
SDIDRV ERR	22	I (PD)	Serial data input of SPI interface to motor driver. Error input signal to ERR pin of TMC23x/24x. If not connected, internal pull-down resistor is active.
MP1	7	I (PD)	DC_IN as external dcStep input control signal. If not connected, internal pull-down resistor is active.
MP2	8	IO	DCSTEP_ENABLE as dcStep output control signal. SPE_OUT as output signal, connect to SPE pin of TMC23x/24x.
STDBY_CLK	26	O	StandBy signal or internal CLK output or ChopSync output.

Table 2: Pin Names and Descriptions

### 1.3. System Overview

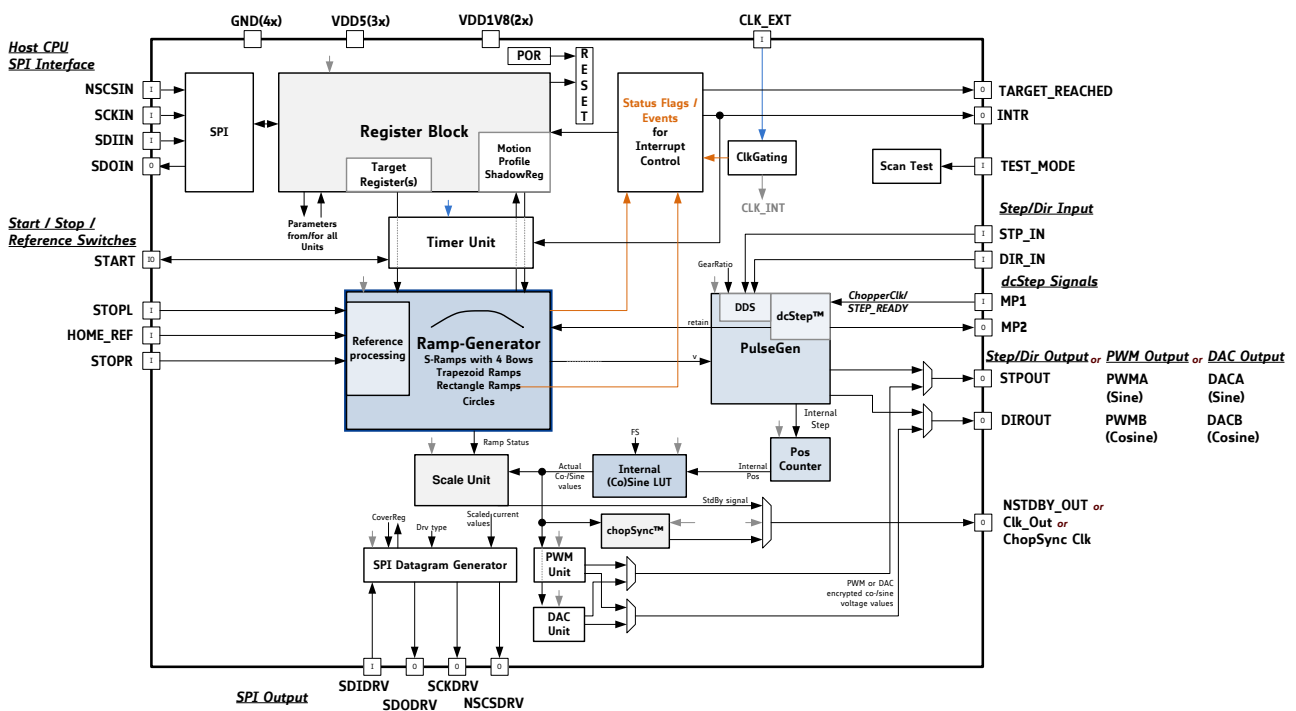


Figure 7: System Overview



## 2. Application Circuits

In this chapter application circuit examples are provided that show how external components can be connected.

### 2.1. TMC4331A Standard Connection: VCC=3.3V

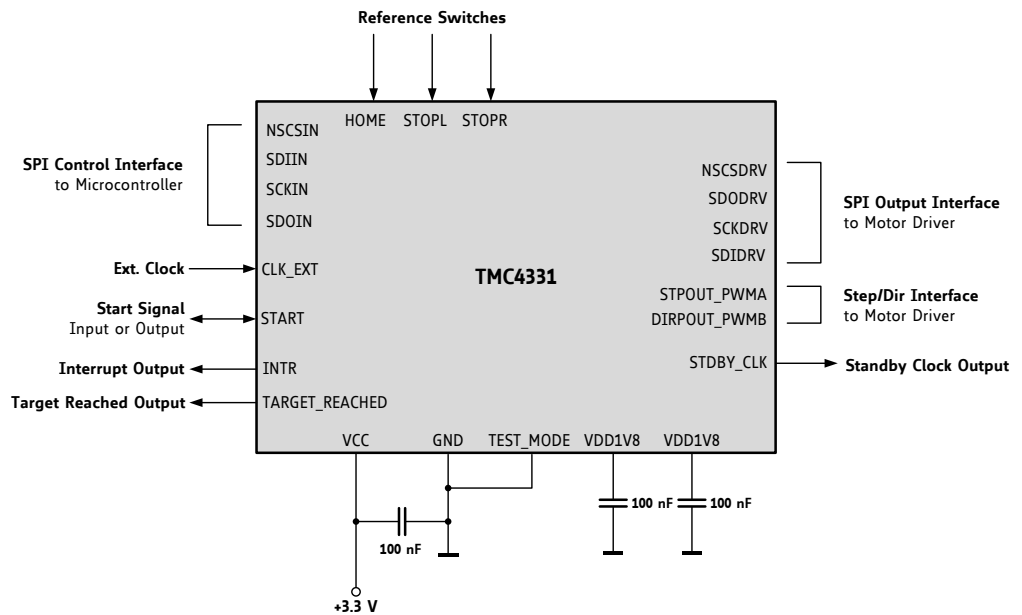


Figure 8: TMC4331A Connection: VCC=3.3V

### 2.2. TMC4331A with TMC26x Stepper Connection

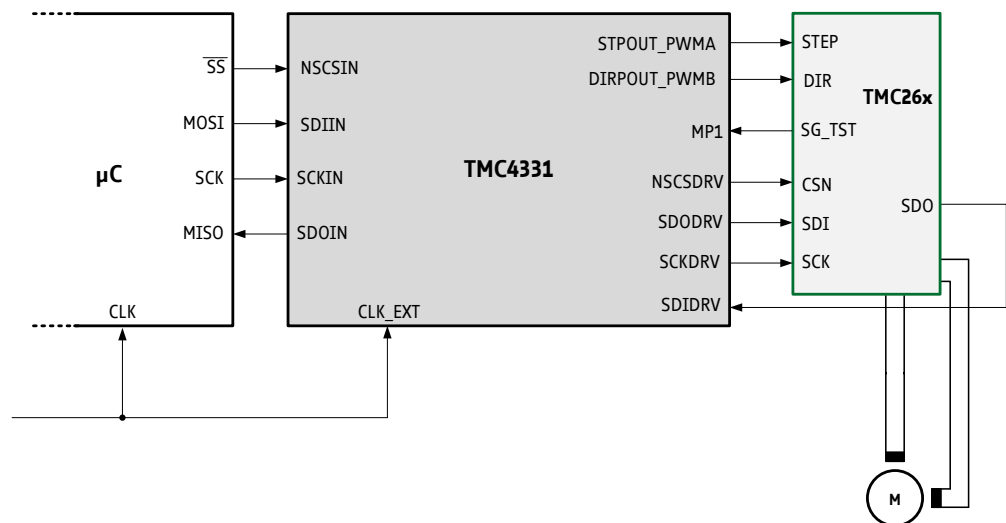


Figure 9: TMC4331A with TMC26x Stepper Driver in SPI Mode or S/D Mode



**2.3.  
TMC4331A with  
TMC248 Stepper  
Driver**

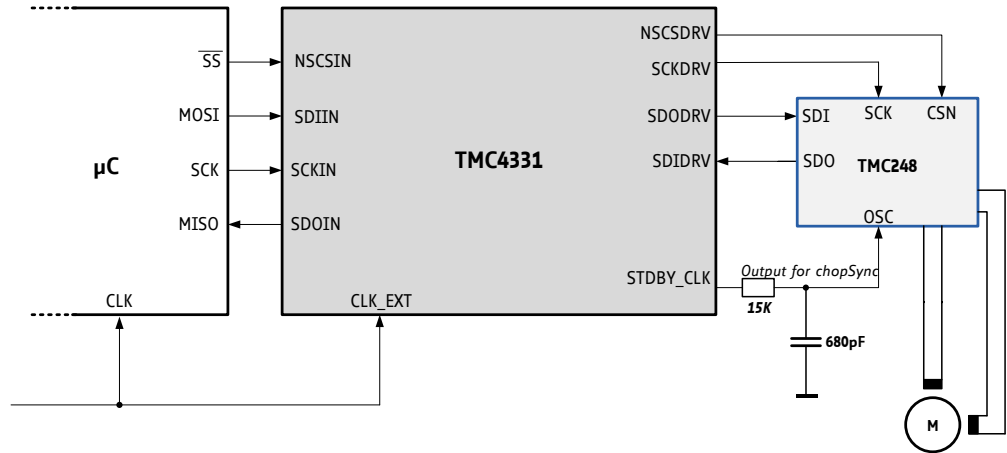


Figure 10: TMC4331A with TMC248 Stepper Driver in SPI Mode

**2.4.  
TMC4331A with  
TMC2130  
Stepper Driver**

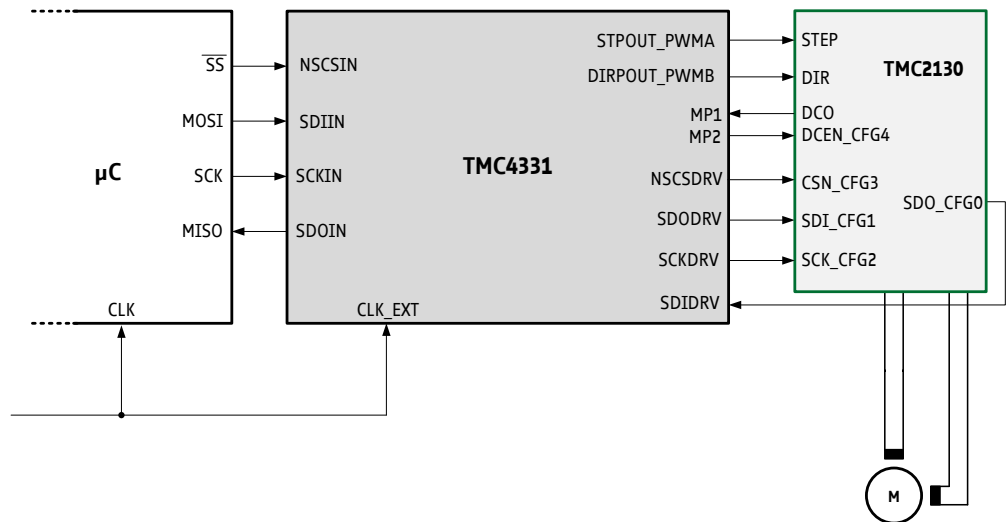


Figure 11: TMC4331A with TMC2130 Stepper Driver in SPI Mode or S/D Mode



### 3. SPI Interfacing

**TMC4331A uses 40-bit SPI datagrams for communication with a microcontroller. The bit-serial interface is 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. In the following chapter information is provided about the SPI control interface, SPI datagram structure and SPI transaction process.**

SPI Input Control Interface Pins		
Pin Name	Type	Remarks
NSCSIN	Input	Chip Select of SPI-μC interface (low active)
SCKIN	Input	Serial clock of SPI-μC interface
SDIIN	Input	Serial data input of SPI-μC interface
SDOIN	Output	Serial data output of SPI-μC interface

Table 3: SPI Input Control Interface Pins

#### 3.1. SPI Datagram Structure

- Microcontrollers that are equipped with hardware SPI are typically able to communicate using integer multiples of 8 bit.
- The NSCSIN line of the TMC4331A has to stay active (low) for the complete duration of the datagram transmission.
- Each datagram that is sent to TMC4331A is composed of an address byte followed by four data bytes. This allows direct 32-bit data word communication with the register set of TMC4331A. Each register is accessed via 32 data bits; even if it uses less than 32 data bits.
  - i Each register is specified by a one-byte address:
    - For read access the most significant bit of the address byte is 0.
    - For write access the most significant bit of the address byte is 1.

**NOTE:**

→ Some registers are write only registers. Most registers can be read also; and there are also some read only registers.

TMC4331A SPI Datagram Structure																																							
MSB (transmitted first)			40 bits				LSB (transmitted last)																																
<b>39</b>			...				<b>0</b>																																
→ 8-bit address ← 8-bit SPI status			← → 32-bit data																																				
39 ... 32			31 ... 0																																				
→ to TMC4331: RW + 7-bit address ← from TMC4331: 8-bit SPI status			8-bit data		8-bit data		8-bit data		8-bit data																														
39 / 38 ... 32			31 ... 24		23 ... 16		15 ... 8		7 ... 0																														
W	38...32		31...28		27...24		23...20		19...16		15...12		11...8		7...4		3...0																						
39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 12: TMC4331A SPI Datagram Structure



**Read/Write Selection Principles and Process**

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. Consequently, the bit named W is a WRITE\_notREAD control bit.

The active high write bit is the MSB of the address byte. Consequently, 0x80 must be added to the address for a write access.

The SPI interface always delivers data back to the master, independent of the Write bit W.

Difference between Read and Write Access	
If ...	Then ...
The previous access was a read access.	The data transferred back is the data read from the address which was transmitted with the previous datagram.
The previous access was a write access	The data read back mirrors the previously received write data.

Figure 13: Difference between Read and Write Access

**Conclusion:**

Consequently, 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.

**NOTE:**

→ Please note that the following read delivers back data read from the address transmitted in the preceding read cycle. The data is latched immediately after the read request.

**AREAS OF SPECIAL CONCERN**



**A read access request datagram uses dummy write data.**

Read data is transferred back to the master with the subsequent read or write access.

**Use of Dummy Write Data**

- i Reading multiple registers can be done in a pipelined fashion. Data that is delivered is latched immediately after the initiated data transfer.

**Read and Write Access Examples**

For read access to register *XACTUAL* with the address 0x21, the address byte must be set to 0x21 in the access preceding the read access. For write access to register *VACTUAL*, the address byte must be set to 0x80 + 0x22 = 0xA2. For read access, the data bit can have any value, e.g., 0.

Read and Write Access Examples		
Action	Data sent to TMC	Data received from TMC
read <i>XACTUAL</i>	→ 0x2100000000	← 0xSS <sup>1)</sup> & unused data
read <i>XACTUAL</i>	→ 0x2100000000	← 0xSS & <i>XACTUAL</i>
write <i>VACTUAL</i> := 0x00ABCDEF	→ 0xA200ABCDEF	← 0xSS & <i>XACTUAL</i>
write <i>VACTUAL</i> := 0x00123456	→ 0xA200123456	← 0xSS00ABCDEF

Table 4: Read and Write Access Examples

<sup>1)</sup> SS is a placeholder for the status bits SPI\_STATUS.



## Data Alignment

All data is right-aligned. Some registers represent unsigned (positive) values; others represent integer values (signed) as two's complement numbers. Some registers consist of switches that are represented as bits or bit vectors.

## SPI Transaction Process

The SPI transaction process is as follows:

- The slave is enabled for SPI transaction by a transition to low level on the chip select input NSCSIN.
  - Bit transfer is synchronous to the bus clock SCKIN, with the slave latching the data from SDIIN on the rising edge of SCKIN and driving data to SDOIN following the falling edge.
  - The most significant bit is sent first.
- i. A minimum of 40 SCKIN clock cycles is required for a bus transaction with TMC4331A.

## AREAS OF SPECIAL CONCERN

### System Behavior Specifics

### Take the following aspects into consideration:

- **Whenever data is read from or written to the TMC4331A**, the first eight bits that are delivered back contain the SPI status *SPI\_STATUS* that consists of eight user-selected event bits. The selection of these bits are explained in chapter 5.2. (Page 22).
- **If less than 40 clock cycles are transmitted**, the transfer is not valid; even for read access. However, sending only eight clock cycles can be useful to obtain the SPI status because it sends the status information back first.
- **If more than 40 clocks cycles are transmitted**, the additional bits shifted into SDIIN are shifted out on SDOIN after a 40-clock delay through an internal shift register. This can be used for daisy chaining multiple chips.
- **NSCSIN must be low during the whole bus transaction.** When NSCSIN 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 NSCSIN* - are recognized as the command.

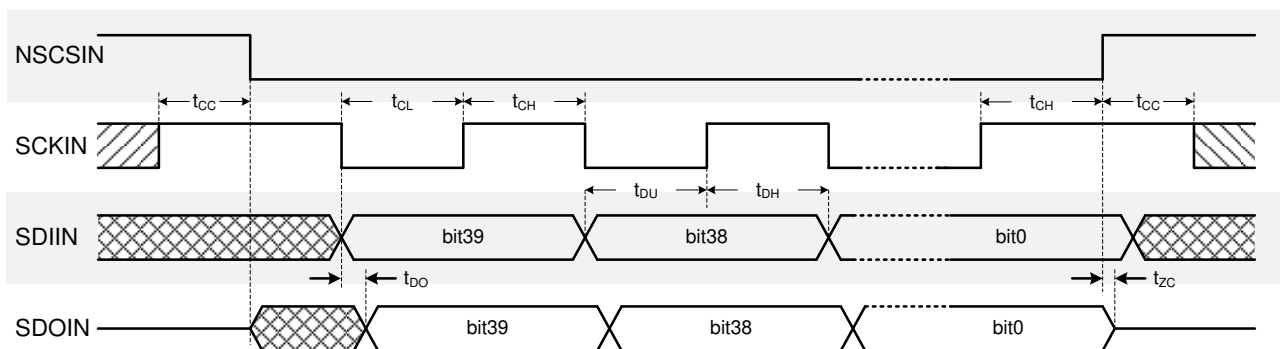


Figure 14: SPI Timing Datagram





### 3.1.1. SPI Timing Description

The SPI interface is synchronized to the internal system clock, which limits SPI bus clock SCKIN to a quarter of the system clock frequency. The signal processing of SPI inputs is supported with internal Schmitt Trigger, but not with RC elements.

#### **NOTE:**

→ In order to avoid glitches at the inputs of the SPI interface between  $\mu C$  and TMC4331A, external RC elements have to be provided.

Figure 14 shows the timing parameters of an SPI bus transaction, and the table below specifies the parameter values.

SPI Interface Timing						
SPI Interface Timing	AC Characteristics:		External clock period: $t_{CLK}$			
Parameter	Symbol	Conditions	Min	Type	Max	Unit
SCKIN valid before or after change of NSCSIN	$t_{CC}$		10			ns
NSCSIN high time	$t_{CSH}$	Min. time is for synchronous CLK with SCKIN high one $t_{CH}$ before SCSIN high only.	$t_{CLK}$	$>2 \cdot t_{CLK} + 10$		ns
SCKIN low time	$t_{CL}$	Min. time is for synchronous CLK only.	$t_{CLK}$	$>t_{CLK} + 10$		ns
SCKIN high time	$t_{CH}$	Min. time is for synchronous CLK only.	$t_{CLK}$	$>t_{CLK} + 10$		ns
SCKIN frequency using external clock (Example: $f_{CLK} = 16$ MHz)	$f_{SCK}$	Assumes synchronous CLK.			$f_{CLK} / 4$ (4)	MHz
SDIIN setup time before rising edge of SCKIN	$t_{DU}$		10			ns
SDIIN hold time after rising edge of SCKIN	$t_{DH}$		10			ns
Data out valid time after falling SCKIN clock edge	$t_{DO}$	No capacitive load on SDOIN.			$t_{FILT} + 5$	ns

Table 5: SPI Interface Timing

$$i \quad t_{CLK} = 1 / f_{CLK}$$



## 4. Input Filtering

Input signals can be noisy due to long cables and circuit paths. To prevent jamming, every input pin provides a Schmitt trigger. Additionally, several signals are passed through a digital filter. Particular input pins are separated into three filtering groups. Each group can be programmed individually according to its filter characteristics. In this chapter informed on the digital filtering feature of TMC4331A is provided; and how to separately set up the digital filter for input pins.

Input Filtering Groups		
Pin Names	Type	Remarks
STPIN DIRIN	Inputs	Step/Dir interface inputs.
STOPL HOME_REF STOPR	Inputs	Reference input pins.
START	Input	START input pin.

*Table 6: Input Filtering Groups (Assigned Pins)*

Register Names			
Register Names	Register Address		Remarks
<i>INPUT_FILT_CONF</i>	0x03	RW	Filter configuration for all four input groups.

*Table 7: Input Filtering (Assigned Register)*

### Input Filter Assignment

Every filtering group can be configured separately with regard to input sample rate and digital filter length.

The following groups exist:

- Step/Dir input pins.
- Reference input pins.
- Start input pin.



## Input Sample Rate (SR)

Input sample rate =  $f_{CLK} / 2^{SR}$  where:

$SR$  (extended with a particular name extension) is in [0... 7].

- i This means that the next input value is considered after  $2^{SR}$  clock cycles.

## Sample Rate Configuration

Sample Rate Configuration	
SR Value	Sample Rate
0	$f_{CLK}$
1	$f_{CLK}/2$
2	$f_{CLK}/4$
3	$f_{CLK}/8$
4	$f_{CLK}/16$
5	$f_{CLK}/32$
6	$f_{CLK}/64$
7	$f_{CLK}/128$

Table 8: Sample Rate Configuration

## Digital Filter Length ( $FILT\_L$ )

- i The filter length  $FILT\_L$  can be set within the range [0... 7].
- i The filter length  $FILT\_L$  specifies the number of sampled bits that must have the same voltage level to set a new input bit voltage level.

## Digital Filter Length Configuration Table

Configuration of Digital Filter Length	
$FILT\_L$ value	Filter Length
0	No filtering.
1	2 equal bits.
2	3 equal bits.
3	4 equal bits.
4	5 equal bits.
5	6 equal bits.
6	7 equal bits.
7	8 equal bits.

Table 9: Configuration of Digital Filter Length



## 4.1. Input Filtering Examples

The following three examples depict input pin filtering of three different input filtering groups.

- i After passing Schmitt trigger, voltage levels are compared to internal signals, which are processed by the motion controller.
- i The sample points are depicted as green dashed lines.

### Example 1: Reference Input Pins

In this example every second clock cycle is sampled. Two sampled input bits must be equal to receive a valid input voltage.

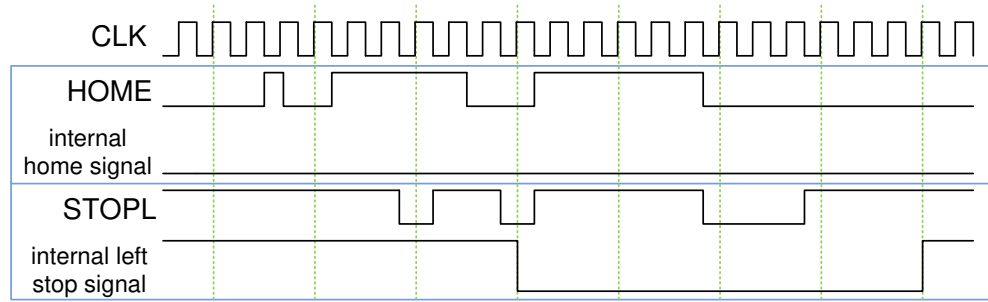


Figure 15: Reference Input Pins:  $SR\_REF = 1$ ,  $FILT\_L\_REF = 1$

### Example 2: START Input Pin

This example shows the START input pattern at every fourth clock cycle:

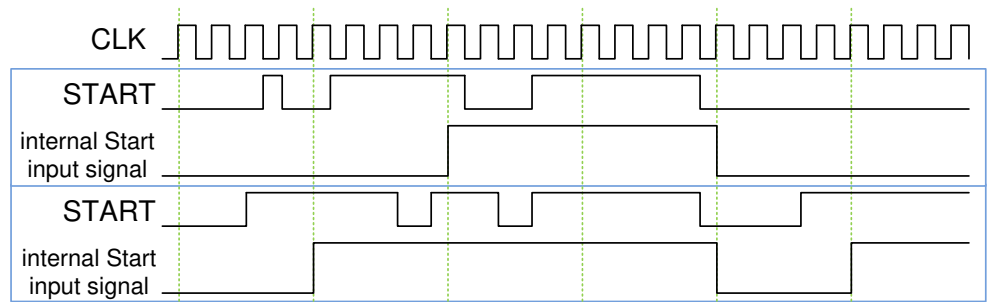


Figure 16: START Input Pin:  $SR\_S = 2$ ,  $FILT\_L\_S = 0$

### Example 3: S/D Input Pins

This example shows every clock cycle bit. Eight sampled input bits must be equal to receive a valid input voltage.

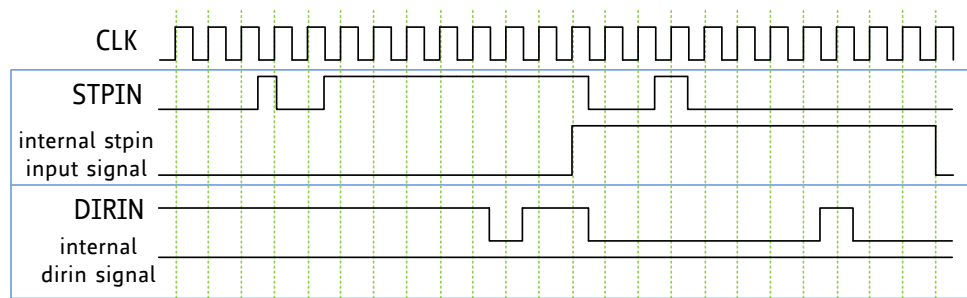


Figure 17: S/D Input Pins:  $SR\_SD\_IN = 0$ ,  $FILT\_L\_SD\_IN = 7$



## 5. Status Flags and Events

TMC4331A provides a range of over 20 status flags and status events in order to obtain short information on the internal status or motor driver status. These flags and events can be read out from dedicated registers. In the following chapter, you are informed about the generation of interrupts based on status events. Status events can also be assigned to the first eight SPI status bits, which are sent within each SPI datagram.

Pin Names: Status Events		
Pin Names	Type	Remarks
INTR	Output	Interrupt output to indicate status events.

Table 10: Pins Names: Status Events

Register Names: Status Flags and Events			
Register Name	Register Address		Remarks
<i>GENERAL_CONF</i>	0X00	RW	Bits: 15, 29, 30.
<i>STATUS_FLAGS</i>	0X0F	R	Status flags of TMC4331A and the connected TMC motor driver chip.
<i>EVENTS</i>	0X0E	R+C W	Events triggered by altered TMC4331A status bits.
<i>SPI_STATUS_SELECTION</i>	0X0B	RW	Selection of 8 out of 32 events for SPI status bits.
<i>EVENT_CLEAR_CONF</i>	0X0C	RW	Exceptions for cleared event bits.
<i>INTR_CONF</i>	0X0D	RW	Selection of events for INTR output.

Table 11: Register Names: Status Flags and Events



## 5.1. Status Event Description

**Status events are based on status bits. If the status bits change, related events are triggered from inactive to active level. Resetting events back to inactive must be done manually.**

### Association of Status Bits

Status bits and status events are associated in different ways:

- Status flags reflect the as-is-condition, whereas status events indicate that the dedicated information has changed since the last read request of the *EVENTS* register. Several status events are associated with one status bit.
- Some status events show the status transition of one or more status bits out of a status bit group. The motor driver flags, e.g., trigger only one motor driver event *MOTOR\_EV* in case one of the selected motor driver status flags becomes active.
- In case a flag consists of more than one bit, the number of associated events that can be triggered corresponds to the valid combinations. The *VEL\_STATE* flag, e.g., has two bit but three associated velocity state events (b'00/b'01/b'10). Such an event is triggered if the associated combination switches from inactive to active.

#### **NOTE:**

→ *Some events have no equivalence in the STATUS\_FLAGS register 0x0F (e.g., COVER\_DONE which indicates new data from the motor driver chip).*

### Automatic Clearance of EVENTS

The *EVENTS* register 0x0E is automatically cleared after reading the register; subsequent to an SPI datagram request. Events are important for interrupt generation and SPI status monitoring.

#### **NOTE:**

→ *It is recommended to clear EVENTS register 0x0E by read request before regular operation.*

### AREAS OF SPECIAL CONCERN



**Recognition of a status event can fail; in case it is triggered right before or during *EVENTS* register 0x0E becomes cleared.**

In order to prevent events from being cleared, assign *EVENT\_CLEAR\_CONF* register 0x0C according to the particular event in the *EVENTS* register:

#### **Action:**

- Set related *EVENT\_CLEAR\_CONF* register bit position to 1.

#### **Result:**

The related event is not cleared when *EVENTS* register is read out.

**In order to clear these events, do the following, if necessary:**

#### **Action:**

- Set related *EVENTS* register 0x0E bit position to 1.

#### **Result:**

The related event is cleared by writing to the *EVENTS* register.

### How to Avoid Lack of Information



## 5.2. SPI Status Bit Transfer

Up to eight events can be selected for permanent SPI status report. Consequently, these events are always transferred at the most significant transfer bits within each TMC4331A SPI response.

### Assign an Event to a Status Bit

In order to select an event for the SPI status bits, assign the *SPI\_STATUS\_SELECTION* register 0x0B according to the particular event in the *EVENTS* register:

#### Action:

- Set the related *SPI\_STATUS\_SELECTION* register bit position to 1.

#### Result:

The related event is transferred with every SPI datagram response as *SPI\_STATUS*.

#### NOTE:

- The bit positions are sorted according to the event bit positions in the *EVENTS* register 0x0E. In case more than eight events are selected, the first eight bits (starting from index 0 = LSB) are forwarded as *SPI\_STATUS*.

## 5.3. Generation of Interrupts

Similar to *EVENT\_CLEAR\_CONF* register and *SPI\_STATUS\_SELECTION* register, events can be selected for forwarding via INTR output. The selected events are ORed to one signal which means that INTR output switches active as soon as one of the selected events triggers.

### Generate Interrupts

In order to select an event for the INTR output pin, assign the *INTR\_CONF* register 0x0D according to the particular event in the *EVENTS* register:

#### Action:

- Set the related *INTR\_CONF* register bit position to 1.

#### Result:

The related event is forwarded at the INTR output. If more than one event is requested, INTR becomes active as soon as one of the selected events is active.

### INTR Output Polarity

Per default, the INTR output is low active.

**In order to change the INTR polarity to high active, do the following:**

#### Action:

- Set *intr\_pol* = 1 (*GENERAL\_CONF* register 0x00).

#### Result:

INTR is high active.



## 5.4. Connection of Multiple INTR Pins

**INTR pin can be configured for a shared interrupt signal line of several TMC4331A interrupt signals to the microcontroller.**

### Connecting several Interrupt Pins

**In order to make use of a Wired-Or or Wired-And behavior, the below described actions must be taken:**

**Action:**

- **Step 1:** Set *intr\_tr\_pu\_pd\_en* = 1 (*GENERAL\_CONF* register 0x00).

**OPTION 1: WIRED-OR**

**Action:**

- **Step 2:** Set *intr\_as\_wired\_and* = 0 (*GENERAL\_CONF* register 0x00).

**Result:**

The INTR pin works efficiently as Wired-Or (default configuration).

- i In case INTR pin is inactive, the pin drive has a weak inactive polarity output. If one of the connected pins is activated, the whole line is set to active polarity.

**OPTION 2: WIRED-AND**

**Action:**

- **Step 2:** Set *intr\_as\_wired\_and* = 1 of the *GENERAL\_CONF* register 0x00.

**Result:**

In case no interrupt is active, the INTR pin has a strong inactive polarity output. During the active state, the pin drive has a weak active polarity output. Consequently, the whole signal line is activated in case all pins are forwarding the active polarity.





## 6. Ramp Configurations for different Motion Profiles

Step generation is one of the main tasks of a stepper motor motion controller. The internal ramp generator of TMC4331A provides several step generation configurations with different motion profiles. They can be configured in combination with the velocity or positioning mode.

Pin Names: Ramp Generator		
Pin Names	Type	Remarks
STPOUT_PWMA	Output	Step output signal.
DIROUT_PWMB	Output	Direction output signal.

Table 12: Pin Names: Ramp Generator

Register Names: Ramp Generator			
Register Name	Register Address	Remarks	
<i>GENERAL_CONF</i>	0x00	RW	Ramp generator affecting bits 5:0.
<i>STP_LENGTH_ADD</i>	0x10	RW	Additional step length in clock cycles; 16 bits.
<i>DIR_SETUP_TIME</i>			Additional time in clock cycles when no steps will occur after a direction change; 16 bits.
<i>RAMPMODE</i>	0x20	RW	Requested motion profile and operation mode; 3 bits.
<i>XACTUAL</i>	0x21	RW	Current internal microstep position; signed; 32 bits.
<i>VACTUAL</i>	0x22	R	Current step velocity; 24 bits; signed; no decimals.
<i>AACTUAL</i>	0x23	R	Current step acceleration; 24 bits; signed; no decimals.
<i>VMAX</i>	0x24	RW	Maximum permitted or target velocity; signed; 32 bits= 24+8 (24 bits integer part, 8 bits decimal places).
<i>VSTART</i>	0x25	RW	Velocity at ramp start; unsigned; 31 bits=23+8.
<i>VSTOP</i>	0x26	RW	Velocity at ramp end; unsigned; 31 bits=23+8.
<i>VBREAK</i>	0x27	RW	At this velocity value, the acceleration/deceleration will change during trapezoidal ramps; unsigned; 31 bits=23+8.
<i>AMAX</i>	0x28	RW	Maximum permitted or target acceleration; unsigned; 24 bits=22+2 (22 bits integer part, 2 bits decimal places).
<i>DMAX</i>	0x29	RW	Maximum permitted or target deceleration; unsigned; 24 bits=22+2.
<i>ASTART</i>	0x2A	RW	Acceleration at ramp start or below VBREAK; unsigned; 24 bits=22+2.
<i>DFINAL</i>	0x2B	RW	Deceleration at ramp end or below VBREAK; unsigned; 24 bits=22+2.
<i>BOW1</i>	0x2D	RW	First bow value of a complete velocity ramp; unsigned; 24 bits=24+0 (24 bits integer part, no decimal places).
<i>BOW2</i>	0x2E	RW	Second bow value of a complete velocity ramp; unsigned; 24bits=24+0.
<i>BOW3</i>	0x2F	RW	Third bow value of a complete velocity ramp; unsigned; 24 bits=24+0.
<i>BOW4</i>	0x30	RW	Fourth bow value of a complete velocity ramp; unsigned; 24 bits=24+0.
<i>CLK_FREQ</i>	0x31	RW	External clock frequency $f_{CLK}$ ; unsigned; 25 bits.
<i>XTARGET</i>	0x37	RW	Target position; signed; 32 bits.

Table 13: Register Names: Ramp Generator



## 6.1. Step/Dir Output Configuration

This section focuses on the description of the Step/Dir output configuration.

### 6.1.1. Step/Dir Output Configuration Steps

Step/Dir output signals can be configured for the driver circuit.

**If step signals must be longer than one clock cycle, do as follows:**

**Action:**

- Set proper *STP\_LENGTH\_ADD* register 0x10 (bit 15:0).

**Result:**

The resulting step length is equal to *STP\_LENGTH\_ADD*+1 clock cycles. This is how the step length is assigned within a range of up to 1-up-to-2<sup>16</sup> clock cycles.

**Action:**

- Set proper *DIR\_SETUP\_TIME* register 0x10 (bit 31:16).

**Result:**

The delay period between DIROUT and STPOUT voltage level transitions last *DIR\_SETUP\_TIME* clock cycles. No steps are sent via STPOUT for *DIR\_SETUP\_TIME* clock cycles after a level change at DIROUT.

#### **PRINCIPLE:**

DIROUT does not change the level:

- During active step pulse signal
- For (*STP\_LENGTH\_ADD*+1) clock cycles after the step signal returns to inactive level

### 6.1.2. STPOUT: Changing Polarity

**STPOUT characteristics can be set differently, as follows:**

Per default, the step output is high active because a rising edge at STPOUT indicates a step.

**In order to change the polarity, do as follows:**

**Action:**

- Set *step\_inactive\_pol* = 1 (bit3 of *GENERAL\_CONF* register 0x00).

**Result:**

Each falling edge indicates a step.

### How to prompt Level Change with every Step

**In order to prompt a step at every level change, do as follows:**

**Action:**

- Set *toggle\_step* = 1 (bit4 of *GENERAL\_CONF* register 0x00).

**Result:**

Every level change indicates a step.

### DIROUT: Changing the Polarity

Per default, voltage level 1 at DIROUT indicates a negative step direction. DIROUT characteristics can be set differently, as shown below.

**In order to change polarity, do as follows:**

**Action:**

- Set *pol\_dir\_out* = 0 (bit5 of *GENERAL\_CONF* register 0x00).

**Result:**

A high voltage level at DIROUT indicates a positive step direction.

#### **NOTE:**

→ *DIROUT* is based on the internal  $\mu$ Step position *MSCNT* and is therefore based on the internal *SinLUT*, see [10.2](#), page [83](#).

