



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



TMC4361A DATASHEET

TMC4361A Document Revision 1.10 • 2016-JUL-20

The S-ramp and sixPoint™ ramp motion controller for stepper motors is optimized for high velocities, allowing on-the-fly changes. TMC4361A offers SPI and Step/Dir interfaces, as well as an encoder interface for closed-loop operation.

NOTE:

→ TMC4361A is a product upgrade of TMC4361.



Figure 1: Sample Image
TMC4361A Closed-Loop Drive

*Marking details are explained on page 227.

Features

- SPI Interfaces for μ C with easy-to-use protocol.
- SPI Interfaces for SPI motor stepper drivers.
- Encoder interface for incremental or serial encoders.
- Closed-loop operation for Step and SPI 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.

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: TMC4361A Interfaces & Features

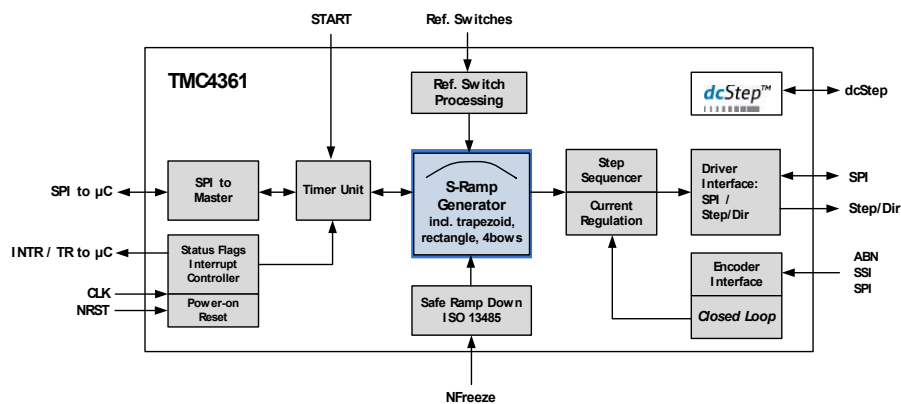


Figure 2: Block Diagram

© 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.



Read entire documentation; especially the Supplemental Directives in chapter 20 (page 228).

• SHORT SPEC •

Functional Scope of TMC4361A

TMC4361A 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, closed-loop 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.

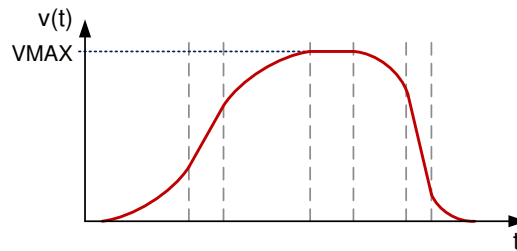


Figure 3: S-shaped Velocity Profile

- i More information on ramp configurations and other velocity profiles, e.g. sixPoint™ ramps, are provided in chapter 6 (Page 28).

Closed-loop Operation Feature

A typical hardware setup for closed-loop operation with a TMC262 stepper motor gate driver is shown in the diagram below. In case internal MOSFETs are desired, combine the TMC4361A with the TMC2620, the TMC261 or the TMC2660.

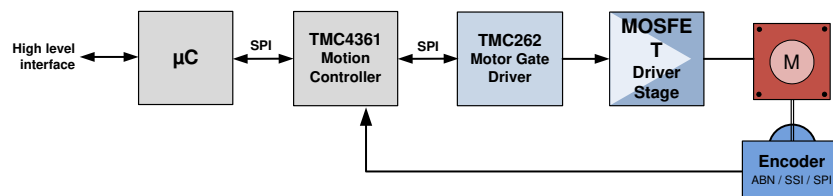


Figure 4: Hardware Set-up for Closed-loop Operation with TMC262

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.

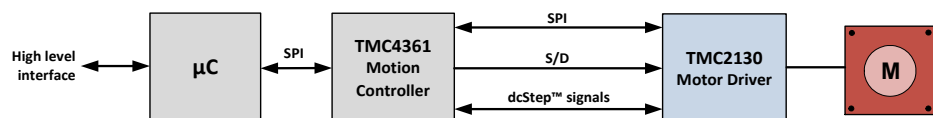


Figure 5: Hardware Set-up for Open-loop Operation with TMC2130

Order Codes

Order code	Description	Size
TMC4361A-LA	Motion controller with closed-loop and dcStep features, QFN40	6 x 6 mm ²

Table 1: TMC4361A Order Codes



TABLE OF CONTENTS

TMC4361A DATASHEET	1
SHORT SPEC	1
Features	1
Applications	1
Block Diagram: TMC4361A Interfaces & Features	1
Functional Scope of TMC4361A	2
Order Codes	2
TABLE OF CONTENTS	3
MAIN MANUAL	10
1. Pinning and Design-In Process Information	10
1.1. Pin Assignment: Top View	10
1.2. Pin Description	11
1.3. System Overview	13
2. Application Circuits	14
2.1. TMC4361A Standard Connection: VCC=3.3V	14
2.2. TMC4361A with TMC26x Stepper Connection	14
2.3. TMC4361A with TMC248 Stepper Driver	15
2.4. TMC4361A with TMC2130 Stepper Driver	15
3. SPI Interfacing	16
3.1. SPI Datagram Structure	16
3.1.1. SPI Timing Description	19
4. Input Filtering	20
4.1. Input Filtering Examples	22
4.2. Configuration of Step/Dir Input Filter	23
5. Status Flags and Events	24
5.1. Status Event Description	25
5.2. SPI Status Bit Transfer	26
5.3. Generation of Interrupts	26
5.4. Connection of Multiple INTR Pins	27
6. Ramp Configurations for different Motion Profiles	28
6.1. Step/Dir Output Configuration	29
6.1.1. Step/Dir Output Configuration Steps	29
6.1.2. STPOUT: Changing Polarity	29
6.2. Altering the Internal Motion Direction	30
6.3. Configuration Details for Operation Modes and Motion Profiles	31
6.3.1. Starting Point: Choose Operation Mode	32



6.3.2.	Stop during Motion	32
6.3.3.	Motion Profile Configuration.....	33
6.3.4.	No Ramp Motion Profile.....	34
6.3.5.	Trapezoidal 4-Point Ramp without Break Point.....	35
6.3.6.	Trapezoidal Ramp with Break Point	35
6.3.7.	Position Mode combined with Trapezoidal Ramps	36
6.3.8.	Configuration of S-Shaped Ramps.....	37
6.3.9.	Changing Ramp Parameters during S-shaped Motion or Switching from Velocity to Positioning Mode.....	38
6.3.10.	Configuration of S-shaped Ramp with <i>ASTART</i> and <i>DFINAL</i>	38
6.3.11.	S-shaped Mode and Positioning: Fast Motion	39
6.4.	Start Velocity <i>VSTART</i> and Stop Velocity <i>VSTOP</i>	40
6.4.1.	S-shaped Ramps with Start and Stop Velocity	44
6.4.2.	Combined Use of <i>VSTART</i> and <i>ASTART</i> for <i>S-shaped Ramps</i>	45
6.5.	sixPoint Ramps	46
6.6.	U-Turn Behavior	47
6.6.1.	Continuous Velocity Motion Profile for S-shaped Ramps.....	48
6.7.	Internal Ramp Generator Units.....	49
6.7.1.	Clock Frequency	49
6.7.2.	Velocity Value Units	49
6.7.3.	Acceleration Value Units	49
6.7.4.	Bow Value Units	50
6.7.5.	Overview of Minimum and Maximum Values:.....	50
7.	External Step Control and Electronic Gearing	51
7.1.	Description of Electronic Gearing	52
7.2.	Indirect External Control.....	52
7.3.	Switching from External to Internal Control.....	53
8.	Reference Switches	55
8.1.	Hardware Switch Support.....	56
8.1.1.	Stop Slope Configuration for Hard or Linear Stop Slopes.....	56
8.1.2.	How Active Stops are indicated and reset to Free Motion	57
8.1.3.	How to latch Internal Position on Switch Events	57
8.2.	Virtual Stop Switches	58
8.2.1.	Enabling Virtual Stop Switches.....	58
8.2.2.	Virtual Stop Slope Configuration.....	58
8.2.3.	How Active Virtual Stops are indicated and reset to Free Motion.....	59
8.3.	Home Reference Configuration	60
8.3.1.	Home Event Selection.....	61
8.3.2.	HOME_REF Monitoring	61
8.3.3.	Homing with STOPL or STOPR.....	62
8.4.	Target Reached / Position Comparison	63
8.4.1.	Connecting several Target-reached Pins	63
8.4.2.	Use of TARGET_REACHED Output.....	64



8.4.3.	Position Comparison of Internal Values	65
8.5.	Repetitive and Circular Motion.....	66
8.5.1.	Repetitive Motion to XTARGET	66
8.5.2.	Activating Circular Motion.....	66
8.5.3.	Uneven or Noninteger Microsteps per Revolution	67
8.5.4.	Release of the Revolution Counter	68
8.6.	Blocking Zones	68
8.6.1.	Activating Blocking Zones during Circular Motion.....	68
8.6.2.	Circular Motion with and without Blocking Zone	69
9.	Ramp Timing and Synchronization	70
9.1.	Basic Synchronization Settings.....	71
9.1.1.	Start Signal Trigger Selection.....	71
9.1.2.	User-specified Impact Configuration of Timing Procedure.....	71
9.1.3.	Delay Definition between Trigger and internally generated Start Signal	72
9.1.4.	Active START Pin Output Configuration	72
9.1.5.	Ramp Timing Examples	73
9.2.	Shadow Register Settings.....	76
9.2.1.	Shadow Register Configuration Options	77
9.2.2.	Delayed Shadow Transfer	81
9.3.	Pipelining Internal Parameters	83
9.3.1.	Configuration and Activation of Target Pipeline	83
9.3.2.	Using the Pipeline for different internal Registers	84
9.3.3.	Pipeline Mapping Overview	85
9.3.4.	Cyclic Pipelining	86
9.3.5.	Pipeline Examples	86
9.4.	Masterless Synchronization of Several Motion Controllers via START Pin.....	89
10.	Serial Data Output	90
10.1.	Getting Started with TMC Motor Drivers	91
10.2.	Sine Wave Lookup Tables	92
10.2.1.	Actual Current Values Output	93
10.2.2.	How to Program the Internal MSLUT	94
10.2.3.	Setup of MSLUT Segments.....	95
10.2.4.	Current Waves Start Values	96
10.2.5.	Default MSLUT	96
10.2.6.	Explanatory Notes for Base Wave Inclinations	97
10.3.	SPI Output Interface Configuration Parameters.....	99
10.3.1.	How to enable SPI Output Communication	99
10.3.2.	Setup of SPI Output Timing Configuration	100
10.3.3.	Current Diagrams.....	101
10.3.4.	Change of Microstep Resolution	101
10.3.5.	Cover Datagrams Communication between μ C and Driver	101
10.3.6.	Sending Cover Datagrams	102
10.3.7.	Configuring Automatic Generation of Cover Datagrams	103



10.4.	Overview: TMC Motor Driver Connections	104
10.4.1.	TMC Stepper Motor Driver Settings.....	104
10.4.2.	TMC Motor Driver Response Datagram and Status Bits.....	105
10.4.3.	Events and Interrupts based on Motor Driver Status Bits.....	105
10.4.4.	Stall Detection and Stop-on-Stall	106
10.5.	TMC23x, TMC24x Stepper Motor Driver	107
10.5.1.	TMC23x Setup.....	107
10.5.2.	TMC24x Setup.....	107
10.5.3.	TMC23x/24x Status Bits.....	108
10.5.4.	Automatic Fullstep Switchover for TMC23x/24x	108
10.5.5.	Mixed Decay Configuration for TMC23x/24x	109
10.5.6.	ChopSync Configuration for TMC23x/24x Stepper Drivers	109
10.5.7.	Doubling ChopSync Frequency during Standstill	109
10.5.8.	Using TMC24x stallGuard Characteristics.....	110
10.6.	TMC26x Stepper Motor Driver	112
10.6.1.	TMC26x Setup (SPI mode).....	112
10.6.2.	TMC26x Setup (S/D mode).....	112
10.6.3.	Sending Cover Datagrams to TMC26x	113
10.6.4.	Automatic Continuous Streaming of Cover Datagrams for TMC26x	113
10.6.5.	TMC26x SPI Mode: Automatic Fullstep Switchover	114
10.6.6.	TMC26x S/D Mode: Automatic Fullstep Switchover	114
10.6.7.	TMC 26x S/D Mode: Change of Current Scaling Parameter	115
10.6.8.	TMC26x Status Bits	116
10.6.9.	TMC26x Status Response.....	116
10.7.	TMC389 Stepper Motor Driver	116
10.8.	TMC2130 Stepper Motor Driver	117
10.8.1.	Set-up TMC2130 Support (SPI Mode)	117
10.8.2.	Set-up TMC2130 Support (S/D Mode)	117
10.8.3.	Sending Cover Datagrams to TMC2130	118
10.8.4.	Automatic Continuous Streaming of Cover Datagrams for TMC2130.....	118
10.8.5.	TMC2130 SPI Mode: Automatic Fullstep Switchover	119
10.8.6.	TMC2130 S/D Mode: Automatic Fullstep Switchover	119
10.8.7.	TMC 2130 S/D Mode: Changing current Scaling Parameter.....	119
10.8.8.	TMC2130 Status Response	120
10.9.	Connecting Non-TMC Stepper Motor Driver or SPI-DAC at SPI output interface.....	121
10.9.1.	Connecting a SPI-DAC.....	122
10.9.2.	DAC Data Transfer	122
10.9.3.	Changing SPI Output Protocol for SPI-DAC	122
10.9.4.	DAC Address Values.....	123
10.9.5.	DAC Data Values.....	123
11.	Current Scaling	125
11.1.	Hold Current Scaling.....	126
11.2.	Freewheeling	126



11.3.	Current Scaling during Motion	127
11.3.1.	Drive Scaling	127
11.3.2.	Alternative Drive Scaling	127
11.3.3.	Boost Current	128
11.4.	Scale Mode Transition Process Control	129
11.5.	Current Scaling Examples	131
12.	NFREEZE and Emergency Stop.....	133
12.1.1.	Configuration of FREEZE Function	133
12.1.2.	Configuration of <i>DFREEZE</i> for automatic Ramp Stop	134
13.	Controlled PWM Output	135
13.1.	PWM Output Generation and Scaling Possibilities.....	136
13.1.1.	PWM Scale Example.....	137
13.2.	PWM Output Generation for TMC23x/24x	138
13.3.	Switching between SPI and Voltage PWM Modes.....	139
14.	dcStep Support for TMC26x or TMC2130.....	140
14.1.	Enabling dcStep for TMC26x Stepper Motor Drivers	142
14.2.	Setup: Minimum dcStep Velocity	143
14.3.	Enabling dcStep for TMC2130 Stepper Motor Drivers	145
15.	Decoder Unit: Connecting ABN, SSI, or SPI Encoders correctly	146
15.1.1.	Selecting the correct Encoder.....	147
15.1.2.	Disabling digital differential Encoder Signals	148
15.1.3.	Inverting of Encoder Direction	148
15.1.4.	Encoder Misalignment Compensation.....	149
15.2.	Incremental ABN Encoder Settings.....	150
15.2.1.	Automatic Constant Configuration of Incremental ABN Encoder	150
15.2.2.	Manual Constant Configuration of Incremental ABN Encoder	150
15.3.	Incremental Encoders: Index Signal: N resp. Z	151
15.3.1.	Setup of Active Polarity for Index Channel	151
15.3.2.	Configuration of N Event.....	151
15.3.3.	External Position Counter <i>ENC_POS</i> Clearing	152
15.3.4.	Latching External Position	153
15.3.5.	Latching Internal Position	153
15.4.	Absolute Encoder Settings	154
15.4.1.	Singleturn or Multiturn Data	154
15.4.2.	Automatic Constant Configuration of Absolute Encoder	155
15.4.3.	Manual Constant Configuration of incremental ABN Encoder	155
15.4.4.	Absolute Encoder Data Setup	156
15.4.5.	Emitting Encoder Data Variation.....	157
15.4.6.	SSI Clock Generation.....	158
15.4.7.	Enabling Multicycle SSI request.....	159
15.4.8.	Gray-encoded SSI Data Streams	159
15.4.9.	SPI Encoder Data Evaluation	160



15.4.10.	SPI Encoder Mode Selection	161
15.4.11.	SPI Encoder Configuration via TMC4361A	162
16.	Possible Regulation Options with Encoder Feedback.....	163
16.1.	Feedback Monitoring.....	163
16.1.1.	Target-Reached during Regulation	163
16.2.	PID-based Control of <i>XACTUAL</i>	164
16.2.1.	PID Readout Parameters.....	164
16.2.2.	PID Control Parameters and Clipping Values.....	165
16.2.3.	Enabling PID Regulation	165
16.3.	Closed-Loop Operation	166
16.3.1.	Basic Closed-Loop Parameters	166
16.3.2.	Enabling and calibrating Closed-Loop Operation.....	167
16.3.3.	Limiting Closed-Loop Catch-Up Velocity	168
16.3.4.	Enabling the Limitation of the Catch-Up Velocity	168
16.3.5.	Enabling Closed-Loop Velocity Mode	169
16.3.6.	Closed-loop Scaling.....	170
16.3.7.	Closed-Loop Scaling Transition Process Control	171
16.3.8.	Back-EMF Compensation during Closed-loop Operation.....	172
16.3.9.	Encoder Velocity Readout Parameters.....	173
16.3.10.	Encoder Velocity Filter Configuration.....	173
16.3.11.	Encoder Velocity equals 0 Event	173
17.	Reset and Clock Gating.....	174
17.1.	Manual Hardware Reset.....	174
17.2.	Manual Software Reset	174
17.3.	Reset Indication.....	174
17.4.	Activating Clock Gating manually.....	175
17.5.	Clock Gating Wake-up	175
17.6.	Automatic Clock Gating Procedure.....	176
18.	Serial Encoder Output.....	177
18.1.1.	Configuration and Enabling of SSI Output Interface.....	178
18.1.2.	Disabling differential Encoder Output Signals	179
18.1.3.	Gray-encoded SSI Output Data	179
TECHNICAL SPECIFICATIONS		180
19.	Complete Register and Switches List.....	180
19.1.	General Configuration Register GENERAL_CONF 0x00.....	180
19.2.	Reference Switch Configuration Register REFERENCE_CONF 0x01	183
19.3.	Start Switch Configuration Register START_CONF 0x02	186
19.4.	Input Filter Configuration Register INPUT_FILT_CONF 0x03	188
19.5.	SPI Output Configuration Register SPI_OUT_CONF 0x04	189
19.6.	Current Scaling Configuration Register CURRENT_CONF 0x05	192
19.7.	Current Scale Values Register SCALE_VALUES 0x06.....	193
19.8.	Encoder Signal Configuration (0x07)	194



19.9.	Serial Encoder Data Input Configuration (0x08)	197
19.10.	Serial Encoder Data Output Configuration (0x09)	198
19.11.	Motor Driver Settings Register STEP_CONF 0x0A	199
19.12.	Event Selection Registers 0x0B..0X0D	200
19.13.	Status Event Register (0x0E)	201
19.14.	Status Flag Register (0x0F)	202
19.15.	Configuration Registers: Closed-Loop, Switches, etc.	203
19.16.	Ramp Generator Registers	205
19.17.	External Clock Frequency Register	209
19.18.	Target and Compare Registers.....	209
19.19.	Pipeline Registers	210
19.20.	Shadow Register	210
19.21.	Freeze Register	211
19.22.	Reset and Clock Gating Register	211
19.23.	Encoder Registers	212
19.24.	PID & Closed-Loop Registers	214
19.25.	Miscellaneous Registers	215
19.26.	Transfer Registers	217
19.27.	SinLUT Registers.....	218
19.28.	TMC Version Register.....	219
20.	Absolute Maximum Ratings	220
21.	Electrical Characteristics.....	221
21.1.	Power Dissipation	221
21.2.	General IO Timing Parameters.....	222
21.3.	Layout Examples	223
21.4.	Internal Circuit Diagram for Layout Example.....	223
21.5.	Components Assembly for Application with Encoder.....	224
21.6.	Top Layer: Assembly Side	224
21.7.	Inner Layer (GND)	225
21.8.	Inner Layer (Supply VS).....	225
21.9.	Package Dimensions.....	226
21.10.	Package Material Information	227
21.11.	Marking Details provided on single Chip.....	227
APPENDICES.....	228
22.	Supplemental Directives	228
	ESD-DEVICE INSTRUCTIONS.....	228
23.	Tables Index.....	230
24.	Figures Index.....	232
25.	Revision History	234



1.2. Pin Description

Pin Names and Descriptions			
Pin	Number	Type	Function
<i>Supply Pins</i>			
GND	6, 15, 25, 36	GND	Digital ground pin for IOs and digital circuitry.
VCC	5, 26, 37	VCC	Digital power supply for IOs and digital circuitry (3.3V... 5V).
VDD1V8	16, 35	VDD	Connection of internal generated core voltage of 1.8V.
CLK_EXT	38	I	Clock input to provide a clock with the frequency fCLK for all internal operations.
NRST	39	I (PU)	Low active reset. If not connected, Power-on-Reset and internal pull-up resistor is active.
TEST_MODE	34	I	Test mode input. VCC = 3.3V: Tie to low for normal operation. VCC = 5.0V: Tie to VDD1V8 for normal operation.
NFREEZE	19	I (PU)	Low active safety pin to immediately freeze output operations. If not connected, internal pull-up resistor is active.
<i>Interface Pins for μC</i>			
NSCSIN	2	I	Low active chip selects input of SPI interface to μ C.
SCKIN	3	I	Serial clock for SPI interface to μ C.
SDIIN	4	I	Serial data input of SPI interface to μ C.
SDOIN	7	O	Serial data output of SPI interface to μ C (Z if NSCSIN=1).
INTR	33	O	Interrupt output, programmable PD/PU for wired-and/or.
TARGET_REACHED	31	O	Target reached output, programmable PD/PU for wired-and/or.
STDBY_CLK	32	O	StandBy signal or internal CLK output or ChopSync output.
<i>Reference Pins</i>			
STOPL	12	I (PD)	Left stop switch. External signal to stop a ramp. If not connected, internal pull-down resistor is active.
HOME_REF	13	I (PD)	Home reference signal input. External signal for reference search. If not connected, internal pull-down resistor is active.
STOPR	14	I (PD)	Right stop switch. External signal to stop a ramp. If not connected, internal pull-down resistor is active.
STPIN	17	I (PD)	Step input for external step control. If not connected, internal pull-down resistor is active.
DIRIN	18	I (PD)	Direction input for external step control. If not connected, internal pull-down resistor is active.
START	20	IO	Start signal input/output.
•→ <i>Continued on next page!</i>			



Pin Names and Descriptions			
Pin	Number	Type	Function
<i>S/D Output Pins</i>			
STPOUT PWMA DACA	24	O	Step output. First PWM signal (Sine). First DAC output signal (Sine).
DIROUT PWMB DACB	23	O	Direction output. Second PWM signal (Cosine). Second DAC output signal (Cosine).
<i>Interface Pins for Stepper Motor Drivers</i>			
NSCSDRV PWMB SDO	30	O	Low active chip selects output of SPI interface to motor driver. Second PWM signal (Cosine) to connect with PHB (TMC23x/24x). Serial data output of serial encoder output interface.
SCKDRV MDBN NSDO	29	O	Serial clock output of SPI interface to motor driver. MDBN output signal for MDBN pin of TMC23x/24x. Negated serial data output of serial encoder output interface.
SDODRV PWMA SCLK	27	IO	Serial data output of SPI interface to motor driver. First PWM signal (Sine) to connect with PHA (TMC23x/24x). Clock input of serial encoder output interface.
SDIDRV ERR NSCLK	28	I (PD)	Serial data input of SPI interface to motor driver. Error input signal to ERR pin of TMC23x/24x. Negated clock input of serial encoder output interface. If not connected, internal pull-down resistor is active.
MP1	8	I (PD)	DC_IN as external dcStep input control signal. If not connected, internal pull-down resistor is active.
MP2	9	IO	DCSTEP_ENABLE as dcStep output control signal. SPE_OUT as output signal, connect to SPE pin of TMC23x/24x.
<i>Encoder Interface Pins</i>			
N	21	I (PD)	N signal input of incremental encoder input interface. If not connected, internal pull-down resistor will be active.
NNEG	22	I (PD)	Negated N signal input of incremental encoder input interface. If not connected, internal pull-down resistor will be active.
B SDI	10	I (PD)	B signal input of incremental encoder input interface. Serial data input signal of serial encoder interface (SSI/SPI). If not connected, internal pull-down resistor is active.
BNEG NSDI SDO_ENC	11	IO	Negated B signal input of incremental encoder input interface. Negated serial data input signal of SSI encoder input interface. Serial data output of SPI encoder input interface.
A SCLK	40	IO	A signal input of incremental encoder interface. Serial clock output signal of serial encoder interface (SSI/SPI).
ANEG NSCLK NSCS_ENC	1	IO	Negated A signal input of incremental encoder interface. Negated serial clock output signal of serial encoder interface. Low active chip select output of SPI encoder input interface.

Table 2: Pin Names and Descriptions



1.3. System Overview

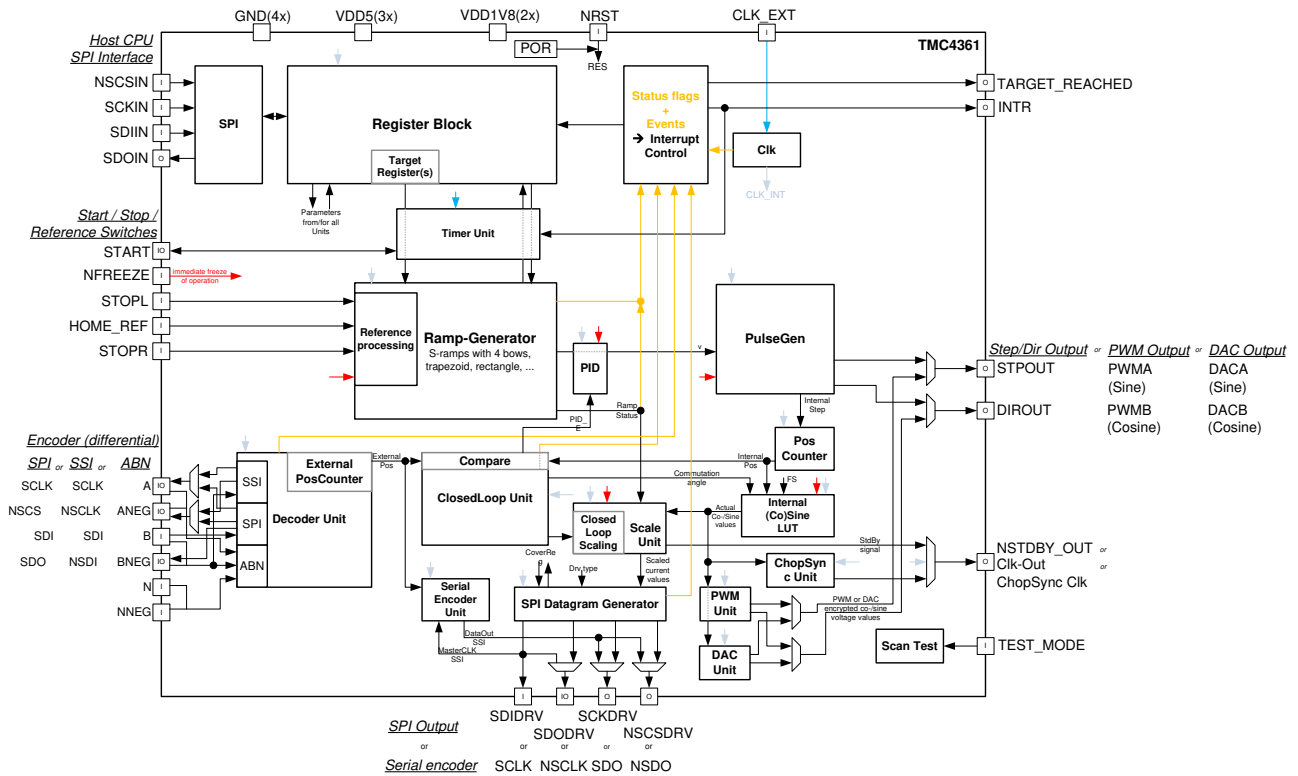


Figure 7: System Overview



2. Application Circuits

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

2.1.

TMC4361A Standard Connection: VCC=3.3V

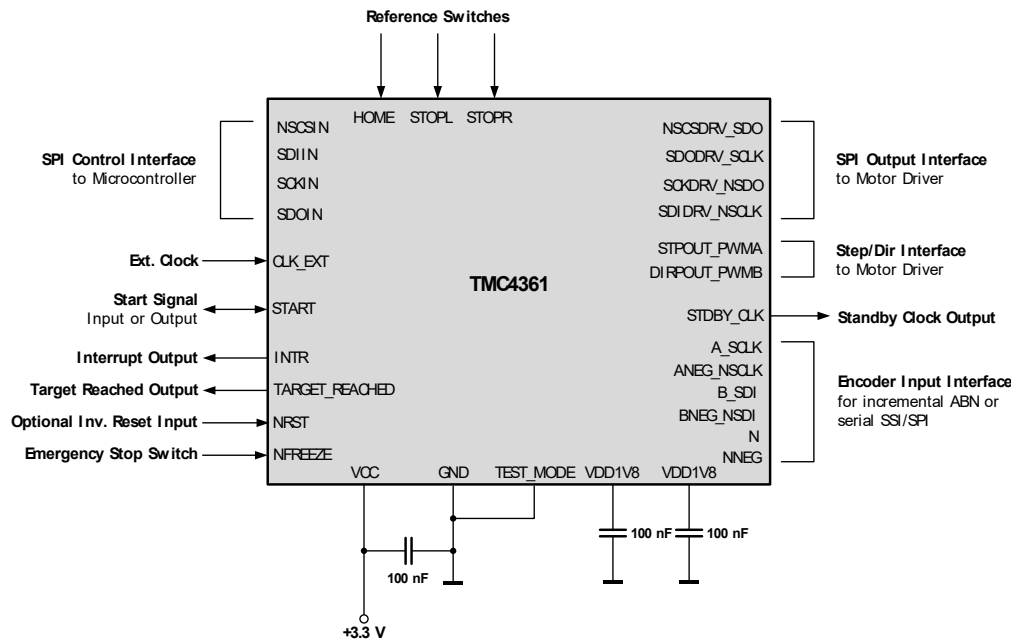


Figure 8: TMC4361A Connection: VCC=3.3V

2.2. TMC4361A with TMC26x Stepper Connection

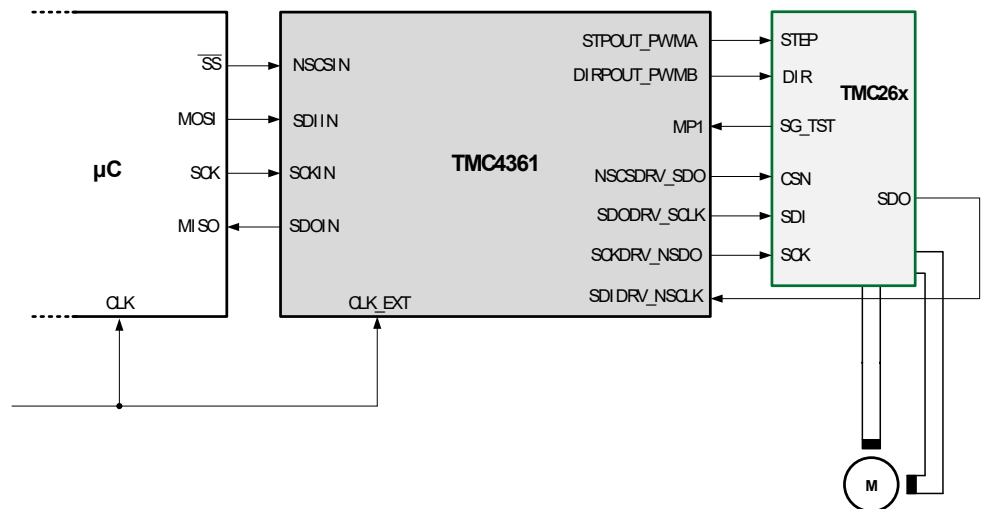


Figure 9: TMC4361A with TMC26x Stepper Driver in SPI Mode or S/D Mode



**2.3.
TMC4361A with
TMC248 Stepper
Driver**

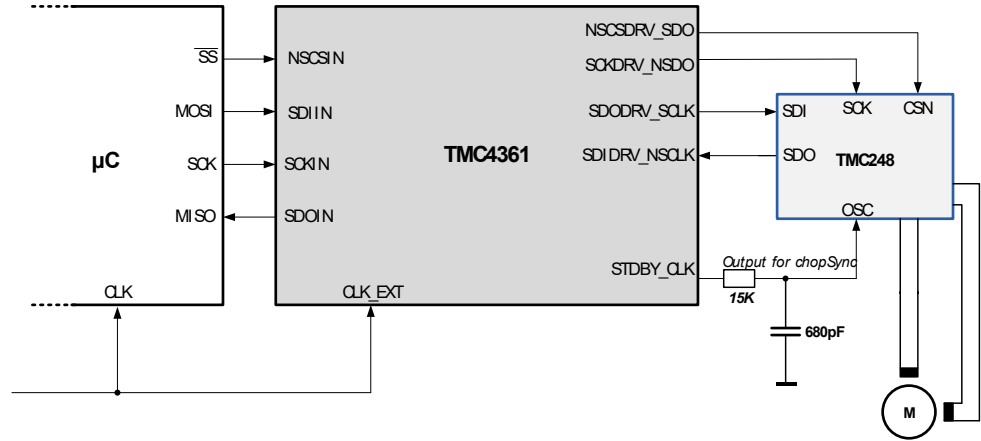


Figure 10: TMC4361A with TMC248 Stepper Driver in SPI Mode

**2.4.
TMC4361A with
TMC2130
Stepper Driver**

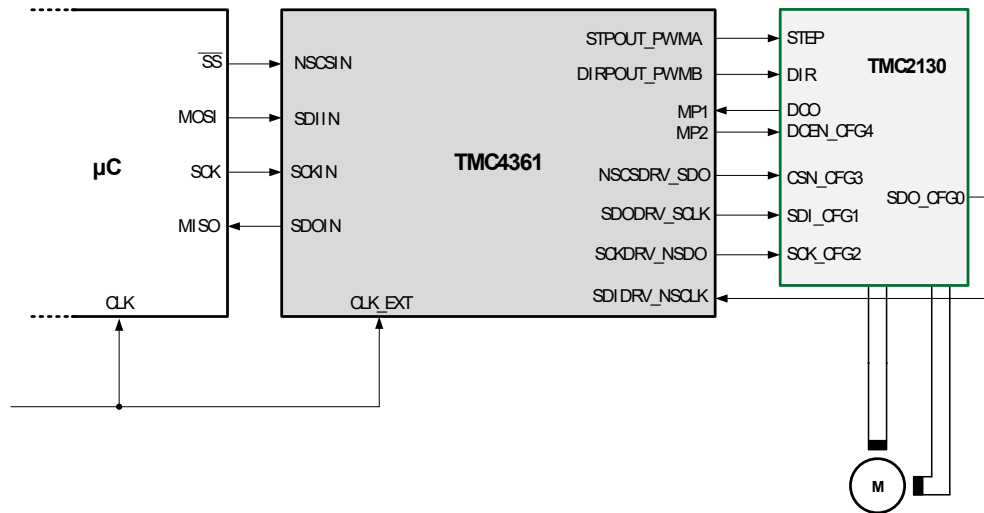


Figure 11: TMC4361A with TMC2130 Stepper Driver in SPI Mode or S/D Mode



3. SPI Interfacing

TMC4361A 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- μ C interface (low active)
SCKIN	Input	Serial clock of SPI- μ C interface
SDIIN	Input	Serial data input of SPI- μ C interface
SDOIN	Output	Serial data output of SPI- μ C 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 TMC4361A has to stay active (low) for the complete duration of the datagram transmission.
- Each datagram that is sent to TMC4361A is composed of an address byte followed by four data bytes. This allows direct 32-bit data word communication with the register set of TMC4361A. Each register is accessed via 32 data bits; even if it uses less than 32 data bits.
 - 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.

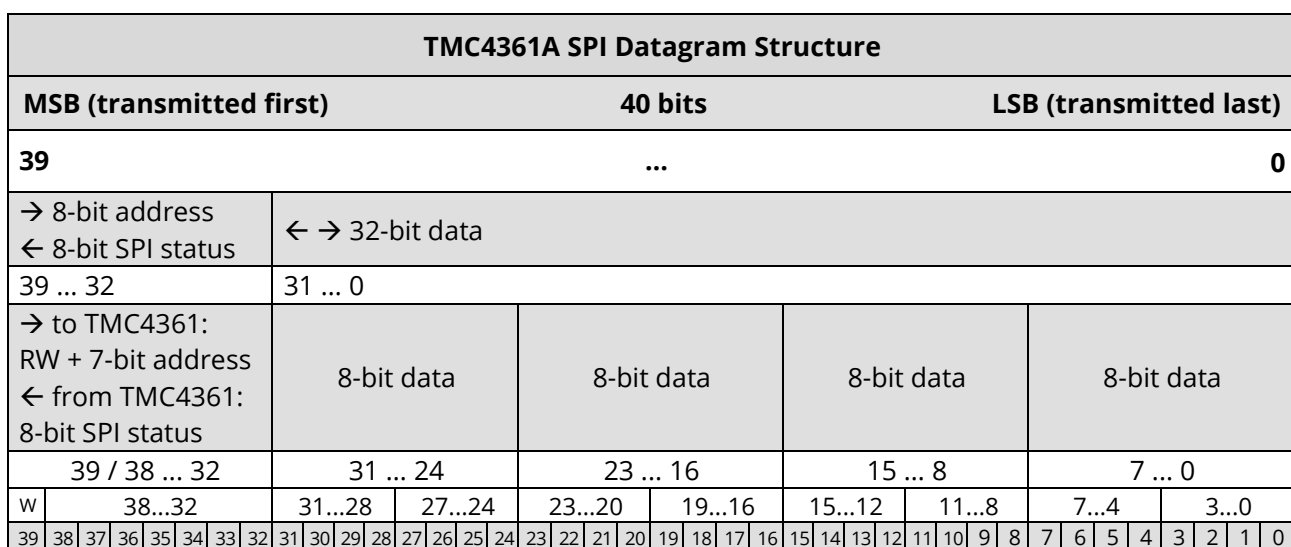


Figure 12: TMC4361A SPI Datagram Structure



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 ¹⁾ & 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

¹⁾ SS is a placeholder for the status bits *SPL_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 TMC4361A.

AREAS OF SPECIAL CONCERN



Take the following aspects into consideration:

System Behavior Specifics

- **Whenever data is read from or written to the TMC4361A**, 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 26).
- **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.

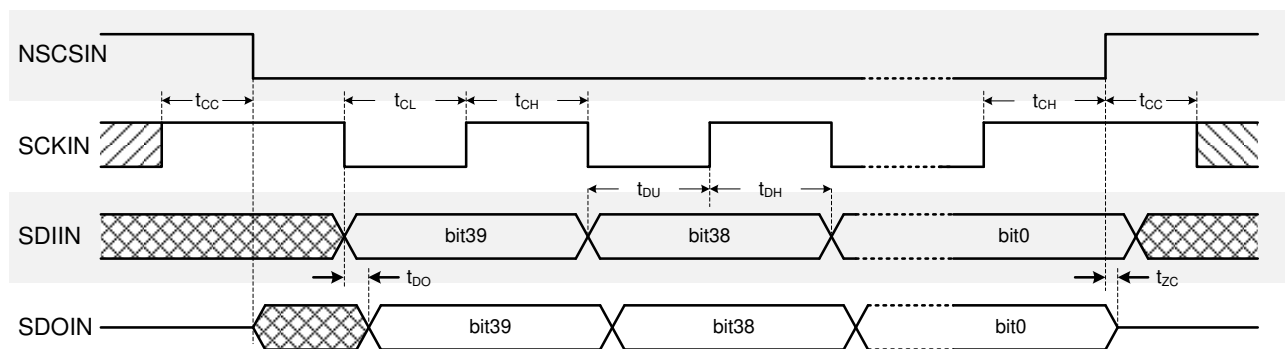


Figure 14: SPI Timing Datagram



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 μC and TMC4361A, 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}	$>2 \cdot t_{\text{CLK}} + 10$		ns
SCKIN low time	t_{CL}	Min. time is for synchronous CLK only.	t_{CLK}	$>t_{\text{CLK}} + 10$		ns
SCKIN high time	t_{CH}	Min. time is for synchronous CLK only.	t_{CLK}	$>t_{\text{CLK}} + 10$		ns
SCKIN frequency using external clock (Example: $f_{\text{CLK}} = 16 \text{ MHz}$)	f_{SCK}	Assumes synchronous CLK.			$f_{\text{CLK}} / 4$ (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.			$t_{\text{FILT}} + 5$	ns

Table 5: SPI Interface Timing

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



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 four filtering groups. Each group can be programmed individually according to its filter characteristics. In this chapter informed on the digital filtering feature of TMC4361A is provided; and how to separately set up the digital filter for input pins.

Input Filtering Groups		
Pin Names	Type	Remarks
A_SCLK B_SDI N ANEG_NSCLK BNEG_NSDI NNEG	Inputs	Encoder interface input pins.
STOPL HOME_REF STOPR	Inputs	Reference input pins.
START	Input	START input pin.
SDODRV_SCLK SDIDRV_NSCLK	Inputs	Master clock input interface pins for serial encoder.
STPIN DIRIN	Inputs	Step/Dir interface inputs.

Table 6: Input Filtering Groups (Assigned Pins)

Register Names			
Register Names	Register Address		Remarks
INPUT_FILT_CONF	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:

- Encoder interface input pins.
- Reference input pins.
- Start input pin.
- Master clock input pins of encoder output interface.
- Step/Dir input pins.

NOTE:

→ Differentiated handling for Step/Dir input pins is necessary, as explained on the following pages.



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.

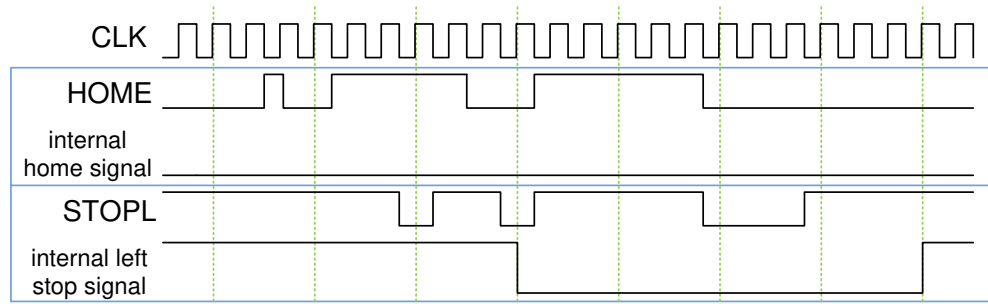


Figure 15: Reference Input Pins: $SR_REF = 1$, $FILT_L_REF = 1$

Example 2:

START Input Pin

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

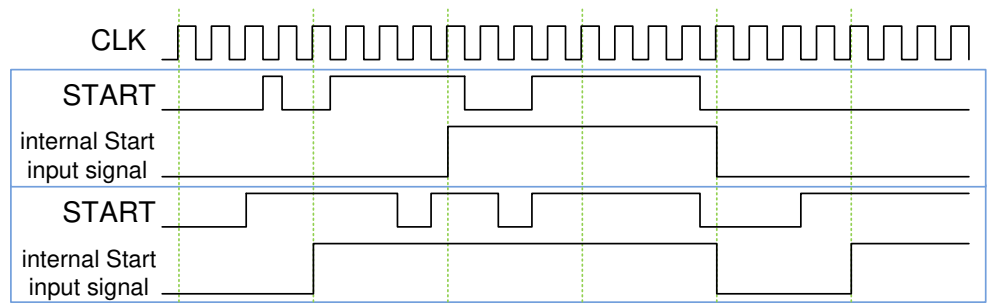


Figure 16: START Input Pin: $SR_S = 2$, $FILT_L_S = 0$

Example 3:

Encoder Interface Input Pins

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

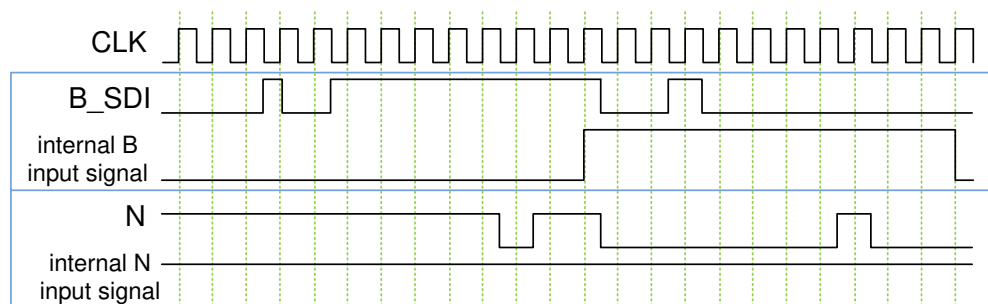


Figure 17: Encoder Interface Input Pins: $SR_ENC_IN = 0$, $FILT_L_ENC_IN = 7$



4.2. Configuration of Step/Dir Input Filter

Step/Dir input filtering setup differs slightly from the other groups, because the other four groups already complete the whole *INPUT_FILT_CONF* register 0x03.

This is why it is possible to assign the Step/Dir input group to one of the existing groups by setting the appropriate bit in front of the setup parameters.

- i If no group is selected, Step/Dir input filtering is automatically assigned to the encoder input interface filter group.

Step/Dir Pin Filter Assignment

The following example shows the filter settings for Step/Dir interface input pins, which are taken from the reference input pin group.

Step/Dir input pin filter settings are derived from the Reference input filter group:

SR_SDIN = 6, FILT_L_SDIN = 3

NOTE:

→ Other input filter groups are:

- SR_ENC_IN = 5, FILT_L_ENC_IN = 6
- SR_REF = 6, FILT_L_REF = 3
- SR_S = 2, FILT_L_S = 4
- SR_ENC_OUT = 0
- FILT_L_ENC_OUT = 0)

Step/Dir Input Filter Parameter

Bits of register 0x03:	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	9	8	7	6	5	4	3	2	1	0
Input filter group:	Serial clock inputs				START input				Reference inputs				Encoder inputs																	
Filter parameter:	FILT_L_ENC_OUT		SR_ENC_OUT		FILT_L_S		SR_S		FILT_L_REF		SR_ENC_REF		FILT_L_ENC_IN		SR_ENC_IN															
Example:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	0	0	1	1	0	0	1	0	1

 = possible selection bits to assign Step/Dir input filter parameter

Figure 18: Step/Dir Input Filter Parameter



5. Status Flags and Events

TMC4361A provides 32 status flags and 32 status events 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
GENERAL_CONF	0X00	RW	Bits: 15, 29, 30.
STATUS_FLAGS	0X0F	R	32 status flags of TMC4361A and the connected TMC motor driver chip.
EVENTS	0X0E	R+C W	32 events triggered by altered TMC4361A status bits.
SPI_STATUS_SELECTION	0X0B	RW	Selection of 8 out of 32 events for SPI status bits.
EVENT_CLEAR_CONF	0X0C	RW	Exceptions for cleared event bits.
INTR_CONF	0X0D	RW	Selection of 32 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 carried out 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

