DATASHEET

TMC406

TRINAMIC MOTION CONTROL CHIP

Features

The TMC406 is the inexpensive solution for processor based high performance multi-axis stepper motor control systems. It supplies all time critical front-end functions which are required to drive up to 6 stepper motors. CPU overhead is drastically reduced, while the CPU has the complete control over velocity ramps, which is essential in multi-dimensional robotic applications. Using the TMC406, multi-axis controllers can be built with a single CPU, avoiding complex and expensive distributed software systems. The required additional peripheral components are minimized. A complete six-axis controller can be built by just adding an inexpensive microcontroller, the desired power drivers and an external quad DAC for each two motors, if microstepping is desired.

- Controls up to six 2-phase stepper motors
- Microstepping with 16 steps/fullstep
- Programmable micro step table
- Supports driver current control for power saving
- Internal position counters
- Position interrupt for detection of position counter overflow
- 18 general purpose I/Os
- Simple to use fast 16 bit wide parallel interface
- Directly interfaces to industry standard DSPs and CPUs
- High current 12 and 24 mA CMOS output drivers
- Small 144 pin TQFP package (-TI)
- Low power consumption, industry standard 5V CMOS

TRINAMIC Microchips GmbH
Deelbögenkamp 4C
22297 Hamburg
GERMANY

T +49 - (0) 40 - 51 48 06 - 0
F +49 - (0) 40 - 51 48 06 - 60
WWW.TRINAMIC.COM
INFO@TRINAMIC.COM

V1.1a / 28-Feb-2000

Life support policy

TRINAMIC Microchips GmbH does not authorize or warrant any of its products for use in life support systems, without the specific written consent of TRINAMIC Microchips GmbH.
Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

Electrical data

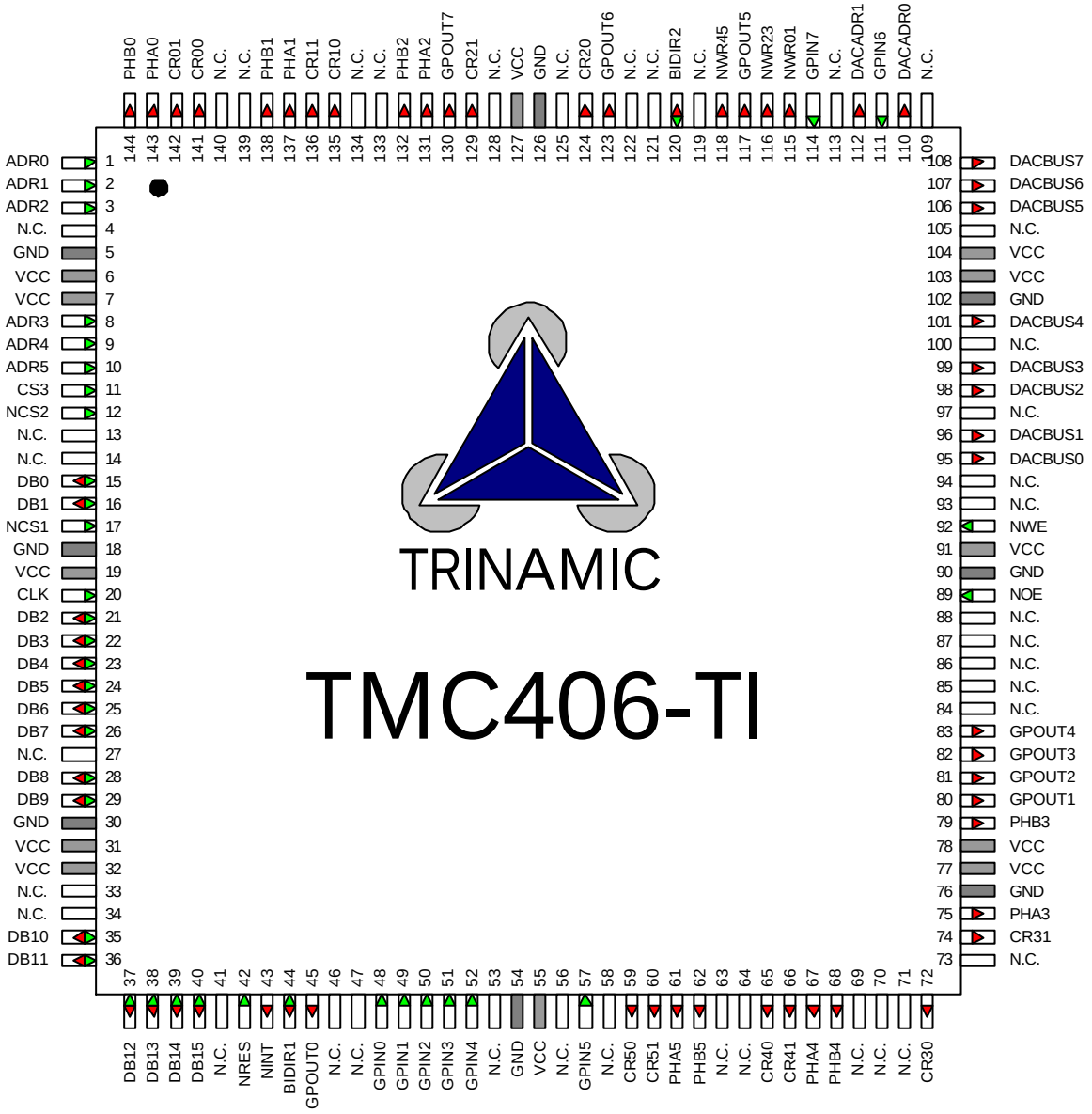


figure 1: TMC406 pinout

Pinout

Pin	TQFP 144	In/Out	Description
CLK	20	I	Clock Input
NRESET	42	I	Reset Input
ADR 0..5	1..3, 8..10	I	Address Bus
DB 0..15	15, 16, 21..26, 28, 29, 35..40	I/O	Bidirectional Data Bus
NCS1, NCS2, CS3	17, 12, 11	I	Chip Select The Chip is selected with NCS1=0, NCS2=0 and CS3=1
NOE	89	I	Read Enable and Output Enable for Data Bus
NWE	92	I	Write Enable Input
NINT	43	O	Interrupt Output
BIDIR1, BIDIR2	44, 120	I/O	General Purpose Bidirectional I/O, 24 mA driver current
GPOUT 0..7	45, 80..83, 117, 123, 130	O	General Purpose Output, 24 mA driver current
GPIN 0..7	48..52, 57, 111, 114	I (S)	General Purpose Input, e.g. for Reference Switches
PHA0, PHB0	143, 144	O	Phase A and Phase B output for Motor Driver 0
PHA1, PHB1	137, 138	O	Phase A and Phase B output for Motor Driver 1
PHA2, PHB2	131, 132	O	Phase A and Phase B output for Motor Driver 2
PHA3, PHB3	75, 79	O	Phase A and Phase B output for Motor Driver 3
PHA4, PHB4	67, 68	O	Phase A and Phase B output for Motor Driver 4
PHA5, PHB5	61, 62	O	Phase A and Phase B output for Motor Driver 5
CR00,CR01	141, 142	O	Digital Current Control for Motor Driver 0
CR10,CR11	135, 136	O	Digital Current Control for Motor Driver 1
CR20,CR21	124, 129	O	Digital Current Control for Motor Driver 2
CR30,CR31	72, 74	O	Digital Current Control for Motor Driver 3
CR40,CR41	65, 66	O	Digital Current Control for Motor Driver 4
CR50,CR51	59, 60	O	Digital Current Control for Motor Driver 5
DACADR0, DACADR1	110, 112	O	Address Bus for external Quad 8 Bit DACs Mapping: 00=Motor i, Ph. A; 01= Motor i, Ph. B; 10=Motor i+1, Ph. A; 11=Motor i+1, Ph. B
DACBUS 0..7	95, 96, 98, 99, 101, 106..108	O	Data Bus for external 8 Bit DACs
NWR01	115	O	Write Enable for DACs for Motor 0 and Motor 1
NWR23	116	O	Write Enable for DACs for Motor 2 and Motor 3
NWR45	118	O	Write Enable for DACs for Motor 4 and Motor 5
VCC	6, 7, 19, 31, 32, 55, 77, 78, 91, 103, 104, 127		+5V Supply
GND	5, 18, 30, 54, 76, 90, 102, 126		Power Ground

(S) are Schmitt-Trigger inputs

Absolute Maximum Ratings

DC Supply Voltage	-0.3V ≤ VDD ≤ 7V
DC Voltage on any Pin	VSS -0.3V≤ Vin ≤VDD+0.3V
Input Current	≤ 10mA
Output Current	≤ 50mA
Power dissipation @ 85°C TC	≤ 1.2W
ESD Voltage on any Pin	1000V
Max. Junction Temperature	≤ 150°C
Storage Temperature	-55°C .. 150°C

Recommended Operating Conditions / Typical Characteristics

(VDD 5V ± 10 %)

	Min	Typ	Max	Units
Digital Supply Voltage VCC	3.0	5.0	5.5	V
Tamb	-40	+25	+85	°C
Operating Frequency	0	20	45	MHz
Supply Current f=0 MHz		0		mA
Supply Current f=20MHz		10	40	mA

TTL DC Characteristics (VDD 5V ± 10 %; -25°C ≤TA ≤ 85°C)

	Min	Typ	Max	Units
Input Low Level	0		0.8	V
Input High Level	2.0		5.5	V
Switching Threshold		1.5		V
Schmitt Trigger Neg. Threshold VDD=5.0V	0.8	1.0		V
Schmitt Trigger Pos. Threshold VDD=5.0V		1.5	2.0	V
Schmitt-Trigger Hysteresis		0.5		V
Output Low Level, I=12mA (resp. I=24 mA)		0.2	0.4	V
Output High Level, I=-12mA (resp. I=-24 mA)	2.4	VCC-0.3		V
Input Current		±1	±10	µA
High Impedance Current (GND to VDD)			±10	µA
Input Capacitance		5		pF
Bidirectional Capacitance		7		pF

Note: 3.3V Operation is also possible. Timing values are less than 40% slower.

Bus interface

The signals NOE, NWE, CS are sampled with the falling clock slope. All other signals are sampled with the rising clock slope. The timings given are calculated for the worst case, i.e. that the input signals are recognized with a delay of one clock due to input change near the falling clock slope. The Tristate-Buffers are asynchronously controlled by CS, NOE and NWR.

Read access:

t_{RDDAV}:	max. 1.5 t_{CLK}+15ns	(CS+NOE+ADR valid to DATA OUT valid)
t_{HIZ}:	max. 10ns	(Read end to data lines tristated)

A read occurs during the overlap of an active CS, an active OE (NOE low) and an inactive WE (NWE high). Read data is latched internally when the read access is first recognized. The read cycle can be finished either by taking OE or CS inactive.

Write access:

t_{WR}:	min. 2 t_{CLK}	(CS+NWR active)
t_{WRDIN}:	max. 1.5 t_{CLK}-15ns	(CS+NWR active to ADR and DATA valid)
t_{WRSDH}:	min. 2.5 t_{CLK}+10ns	(ADR and DATA hold time from CS+NWR active)

A write occurs during the overlap of an active CS and an active WE (NWE low). The internal write occurs one and a half clock after the write condition is recognized.

The timing diagram shows a read access followed by a write access. The unused chip select lines (ncs1, ncs2) are assumed active (low). Since the illustrated timing is synchronous, the initial data setup time can be shortened here.

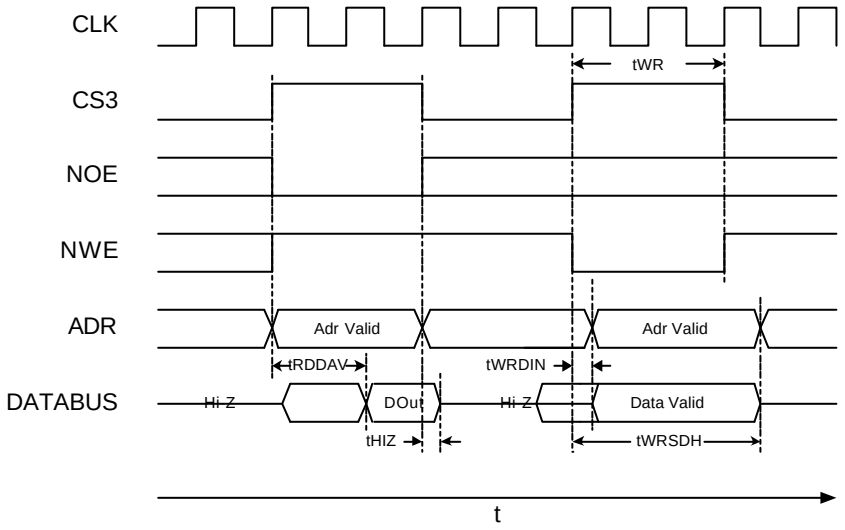


figure 2: read and write cycle

DAC interface

This interface is designed to directly control quad 8 bit DACs like the Texas Instruments TLC7226. The duration of a write access on this 8 bit bus is 3 clocks and thus every DAC is updated every 12 system clocks. The DAC data should be clocked in with the rising nWRxx signal (write goes inactive).

DAC Bus Timing:

t_{NWRxx}:	1 t_{CLK}	(Duration of NWRxx)
$t_{DACADROFS}$:	min. 1 t_{CLK}-10ns	(Address setup time for DAC write)
$t_{DACADRHOLD}$:	min. 1 t_{CLK}	(Address hold time for DAC write)
$t_{DACDATAOFS}$:	min. 1 t_{CLK}-20ns	(Data setup time for DAC write)
$t_{DACDATAHOLD}$:	min. 1 t_{CLK}	(Data hold time for DAC write)

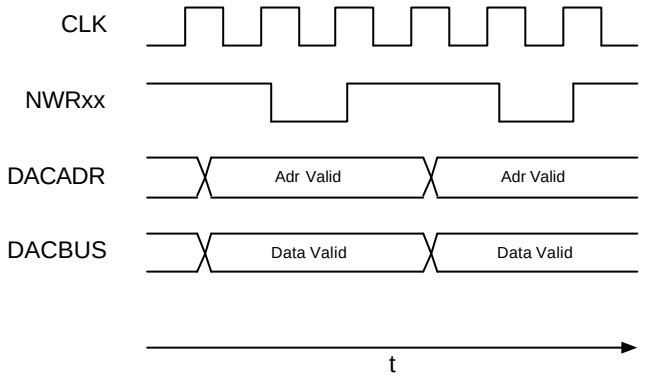


figure 3: DAC timing

Register set

Name	Address	R/W	Description
V_0	0	W	10 Bit signed velocity value (-512..+511)
POS_0	1	RW	10 Bit unsigned Position value (counts step pulses)
CUR_0	2	W	Bit 0,1: Current Control Outputs CURREDUCT00, 01 Bit 2,3: Shift right of DAC output (0..2 bit): 0=the RAM table is output on DAC Bits 7..2 1=the RAM table is output on DAC Bits 6..1 2=the RAM table is output on DAC Bits 5..0 3=the same as 2, but bit 5 is set to zero
DIV_0	3	W	Bit 0,1: Clock divider for pulse generator (divides by 1,2,4 or 8) Bit 2: Position compare bit: When different from bit 9 of the the position register, the interrupt output is activated
$V_{1..5}$	4,8,..,20		Motors 1..5: Same functions as registers 0..3 for Motor 0
$POS_{1..5}$	5,9,..,21		
$CUR_{1..5}$	6,10,..,22		
$DIV_{1..5}$	7,11,..,23		
CLKDIV	32	W	Bit 4..0: Common clock divider for all pulse generators (divides by 1..32)
GPIO	33	W	Bit 0,1: BIDIR output value Bit 2,3: BIDIR 1,2 output enable (inverted, 1 = Input) Bit 8..15: GPOUT 0..7
GPIO	33	R	Bit 0,1: BIDIR 1,2 input value Bit 2..9: GPIN 0..7
RAMTABLE	48..63	W	Bits 7..2: 6 Bit data word The 16 RAM locations are output depending on the position value of each motor: When Bit 4 of POS_i is '0', bits 3..0 select RAM location 0..15 (in ascending order). When Bit 4 of POS_i is '1', bits 3..0 select RAM location 15..0 (in descending order).

Reset values

All internal registers are cleared to 0 upon reset of the chip unless the following registers:

CUR_i: Is reset to 1111 binary, switching the motor drivers to minimum current. For operation set the shift right value to 00, 01 or 10.

When the BIDIR-Pins are used as inputs with active high input signals, a resistor should be provided to limit current flow in Reset State

Step frequency (microsteps)

$$f_{step} = \frac{f_{clk}}{clkdiv + 1} \cdot v_i / 2^{14+div_i}$$

full step frequency =1/16 micro step frequency

Position compare function

The NINT output is activated (low), when the comparison bit for at least one position counter does not match the MSB (bit9) of the corresponding position counter. This function can be used to generate an interrupt whenever a position counter overflows or underflows. An interrupt is also issued, when the position counter passes the middle of the counting range i.e. each 512 microsteps.

Programming the TMC406

Setting up the chip for operation

To drive a motor in microstep operation, it is necessary to set up the RAM-table in the TMC406. Usually it needs to be programmed with a quarter 16-entry sine wave, or a similar waveform, which can be adapted to meet the motor's special microstep-characteristics. This table is output to the external DACs controlling the absolute motor coil current, resulting in a half sine wave. The phase shift between the current control outputs for the two phases of one motor is 90 degrees, resulting in a sine and a cosine wave current.

Example:

```
word quartersintab[16]={13,37,62,86,109,131,152,171,189,205,219,231,240,247,252,255};
                        /* 255 * SIN([0.5..15.5]/32*PI) */
```

Driving the motor

You can both drive the motor by changing the position counter value, or automatically by setting the velocity value for the desired motor speed.

Velocity ramp generation

It is recommended to set up an interrupt procedure, which polls the motor position continuously and increases or decreases the velocity, to speed up or slow down the motor. To stop the motor at a desired position, it is necessary to find out when the motor has to be slowed down again. One of the simplest ways to find this point, is to store the number of steps which were needed in the acceleration phase. The same number of steps will be needed in the slow down phase (with negative acceleration). Since this is not always exactly true, add some offset to slow down a bit earlier.

Extending the position registers

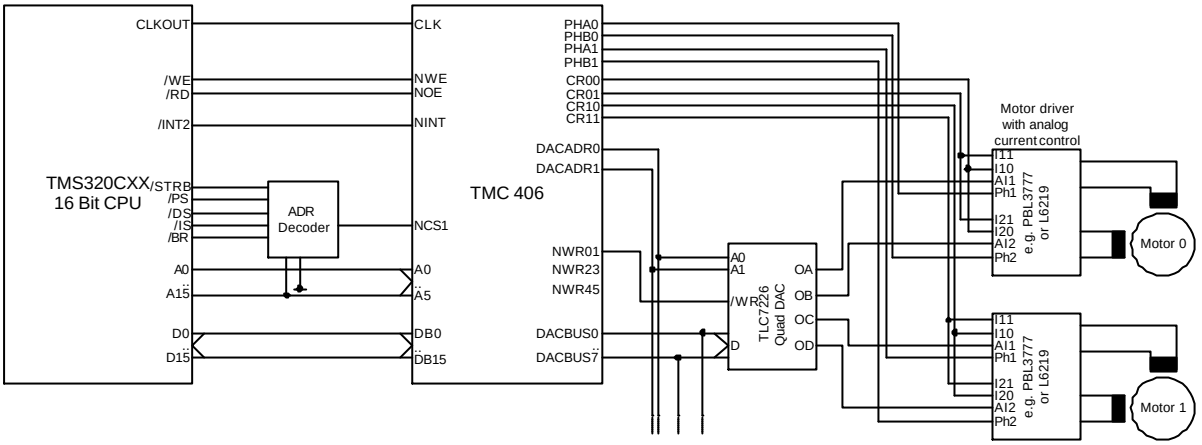
The internal position registers have to be extended to 16, 24 or more bits for most applications, when the exact motor position has to be known. This is usually done using a polling algorithm, i.e. the host processor reads each position counter in an interrupt procedure with a fixed frequency. Whenever the processor sees an overflow or underflow of the TMC406 position counter, by comparing the new value to the previous contents, it increases or decreases its internal position counter variable, depending on the sign of the velocity.

Finding the reference position

To exactly find the reference position switch, the following procedure is recommended: In the first step do a coarse search for the reference switch with a high driving speed, then slow down to a speed, where the host processor can monitor each motor step. When the switch is detected, do not change the TMC406 position register, because this would influence the motor position. Instead, store the actual position register value as an offset value, which has to be subtracted from the read out position afterwards.

Application circuit

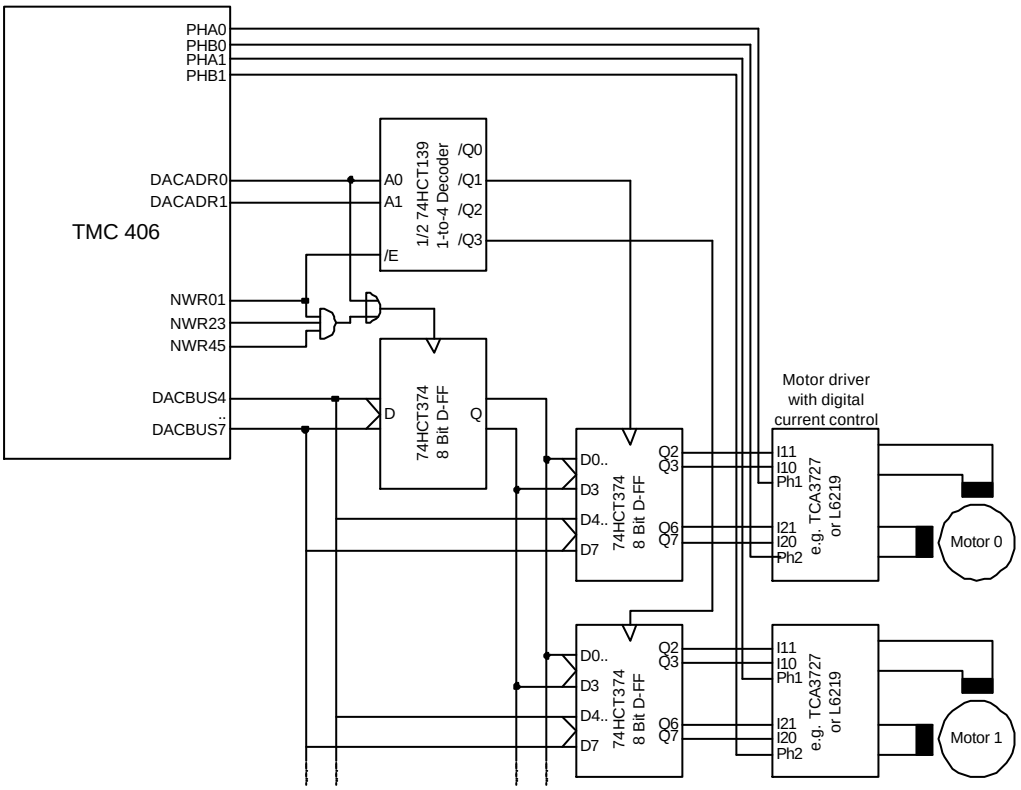
Standard application with high-performance DSP and fine microstepping



schematic 1: Sample application

Implementing microstepping without DACs

Microsteps can also be accomplished using stepper motor drivers with internal DACs, e.g. several standard drivers support up to 16 different currents selectable via up to 4 digital inputs. To interface to these drivers, the TMC406 DAC output has to be de-multiplexed. An example using standard ministep-drivers is shown in schematic 2. The circuit can even be simplified, when less motors or less current levels are desired.



schematic 2: Microstepping without DACs