PD57/60-1076 TMCL™ Firmware Manual

Firmware Version V1.02 | Document Revision V1.02 • 2019-APR-23

The PD57/60-1076 PANdrive is a full mechantronic solution, made up of a TMCM-1076 module and a NEMA 23 or NEMA 24 (57mm or 60mm flange size) stepper motor. The drive is controlled via a step and direction interface. One configuration pin selects the current control mode between StealthChop™ for absolute silent motor control and SpreadCycle™ for high speed. A TTL UART interface allows for more advanced configuration and permanent parameter storage via TMCL-IDE.

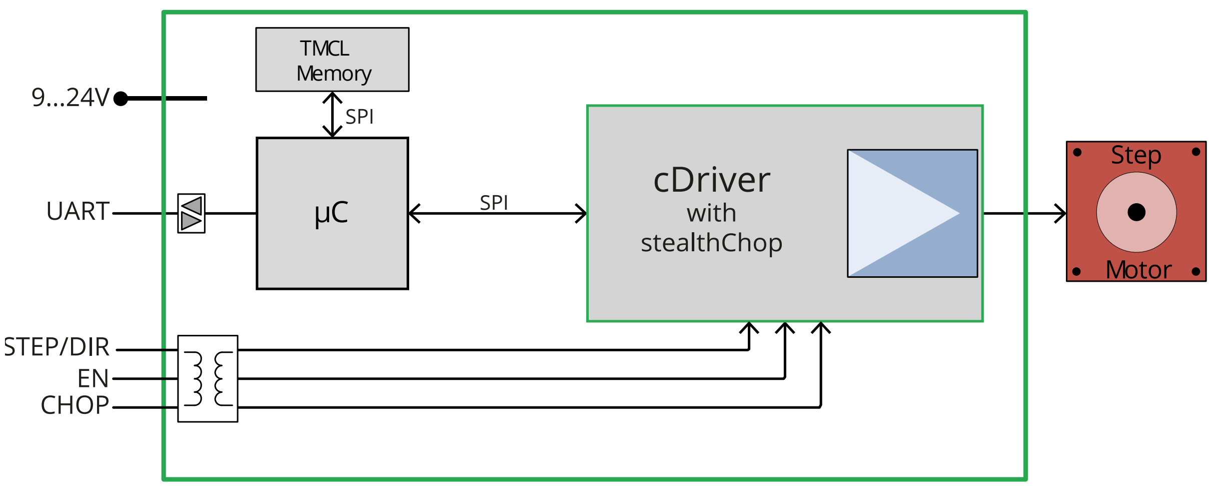
Features

- Supply Voltage +10. . . +30V DC
- Up to 3A RMS motor current
- Step and direction interface
- MicroPlyer™ to 256 microsteps
- StealthChop™ silent PWM mode
- SpreadCycle™ smart mixed decay
- StallGuard2™ load detection
- CoolStep™ autom. current scaling
- UART configuration interface

Applications

- Laboratory Automation
- Manufacturing
- Semiconductor Handling
- Robotics
- Factory Automation
- CNC
- Life Science
- Biotechnology
- Liquid Handling

Simplified Block Diagram



Read entire documentation.

Contents

1 Features

The PD57/60-1076 is a full mechatronic solution. It is made up of a TMCM-1076 module and a NEMA 23 or NEMA 24 (57mm or 60mm flange size) motor. It is highly integrated, offers a convenient handling and can be used in many decentralized applications. This PANdrive has been designed for coil currents up to 3A RMS and 24V DC supply voltage. Two digital inputs are present for enabling the module and choosing the desired chopper mode. With its high energy efficiency from TRINAMIC's CoolStep™ technology cost for power consumption is kept down.

Main characteristics

- Motion controller & stepper motor driver:
  - Hardware motion profile calculation in real-time.
  - On the fly alteration of motion parameters (e.g. position, velocity, acceleration).
  - High performance microcontroller for overall system control and communication protocol handling.
  - Up to 256 microsteps per full step.
  - High-efficient operation, low power dissipation.
  - Dynamic current control.
  - Integrated protection.
  - StallGuard2™ feature for stall detection.
  - CoolStep™ feature for reduced power consumption and heat dissipation.
  - StealthChop™ feature for quiet operation and smooth motion.
  - DcStep™ feature for load dependent speed control.
- Interfaces
  - TTL-UART (for configuration).
  - Step/Direction.
  - Enable Input.
  - Chopper Mode Input.

Software

TMCL: remote controlled operation alone or during step/direction mode. PC-based application development software TMCL-IDE available for free.

Electrical data

- Supply voltage: +24V nominal (10...30V DC supply range).
- Motor current: up to 3A RMS / 4.2A peak (programmable).

Please see also the separate Hardware Manual.

1.1 StallGuard2

StallGuard2 is a high-precision sensorless load measurement using the back EMF of the coils. It can be used for stall detection as well as other uses at loads below those which stall the motor. The StallGuard2 measurement value changes linearly over a wide range of load, velocity, and current settings. At maximum motor load, the value reaches zero or is near zero. This is the most energy-efficient point of operation for the motor.

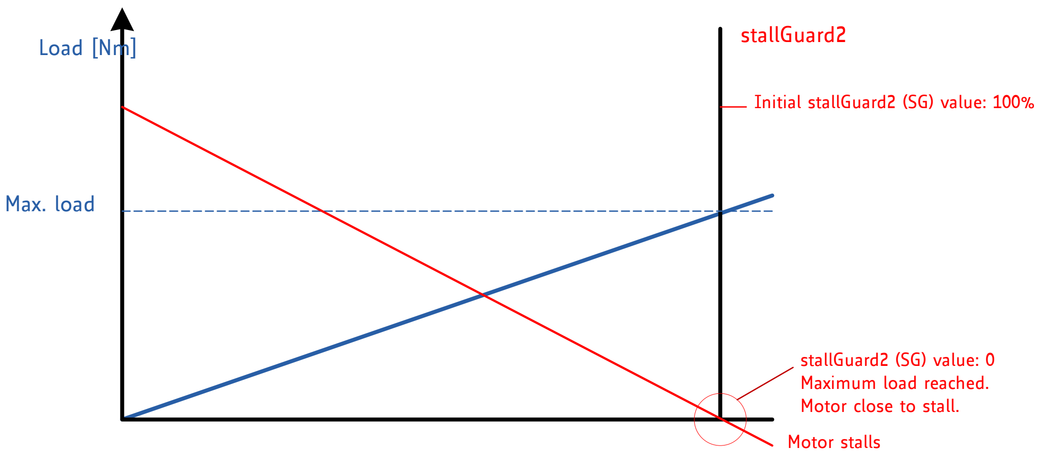


Figure 1: StallGuard2 Load Measurement as a Function of Load

1.2 CoolStep

CoolStep is a load-adaptive automatic current scaling based on the load measurement via StallGuard2 adapting the required current to the load. Energy consumption can be reduced by as much as 75%. CoolStep allows substantial energy savings, especially for motors which see varying loads or operate at a high duty cycle. Because a stepper motor application needs to work with a torque reserve of 30% to 50%, even a constant-load application allows significant energy savings because CoolStep automatically enables torque reserve when required. Reducing power consumption keeps the ystem cooler, increases motor life, and allows cost reduction.

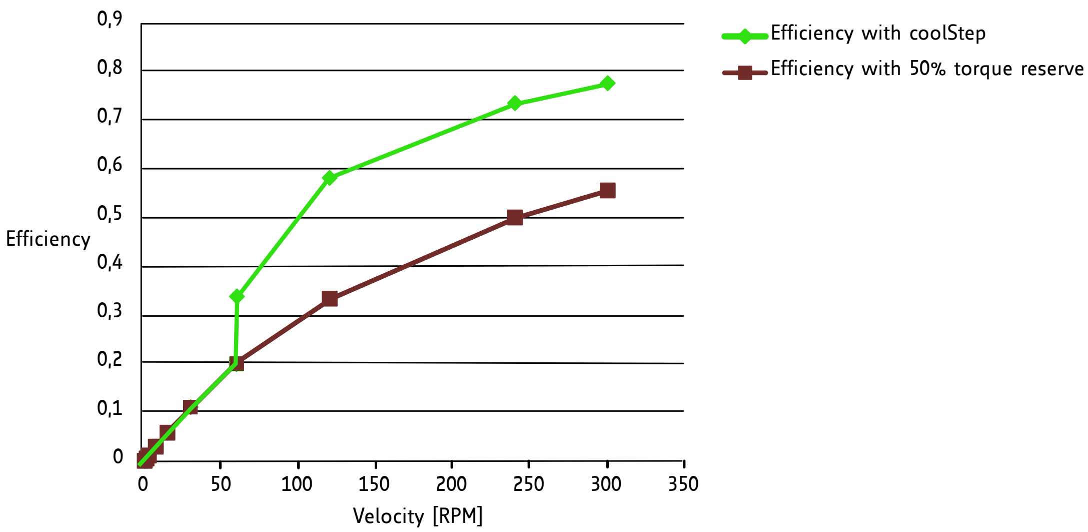


Figure 2: Energy Efficiency Example with CoolStep

2 First Steps with TMCL

In this chapter you can find some hints for your first steps with the PD57/60-1076 and TMCL. You may skip this chapter if you are already familiar with TMCL and the TMCL-IDE.

Things that you will need

- Your PD57/60-1076 PANdrive™.
- A 5V TTL-UART cable.
- A power supply (24V DC) for your PD57/60-1076 module.
- The TMCL-IDE 3.x already installed on your PC

2.1 Basic Setup

First of all, you will need a PC with Windows (at least Windows 7) and the TMCL-IDE 3.x installed on it. If you do not have the TMCL-IDE installed on your PC then please download it from the TMCL-IDE product page of Trinamic's website (http://www.trinamic.com) and install it on your PC.

Please also ensure that your PD57/60-1076 is properly connected to your power supply and that the stepper motor is properly connected to the module. Please see the PD57/60-1076 hardware manual for instructions on how to do this. **Do not connect or disconnect a stepper motor to or from the module while the module is powered!**
Then, please start up the TMCL-IDE. After that you can connect your PD57/60-1076 via UART and switch on the power supply for the module (while the TMCL-IDE is running on your PC). The module will be recognized by the TMCL-IDE.

2.2 Using the TMCL Direct Mode

At first try to use some TMCL commands in direct mode. In the TMCL-IDE a tree view showing the PD57/60-1076 and all tools available for it is displayed. Click on the Direct Mode entry of the tool tree. Now, the Direct Mode tool will pop up.

In the Direct Mode tool you can choose a TMCL command, enter the necessary parameters and execute the command.

2.3 Changing Axis Parameters

Next you can try changing some settings (also called axis parameters) using the SAP command in direct mode. Choose the SAP command. Then choose the parameter type and the motor number. Last, enter the desired value and click execute to execute the command which then changes the desired parameter. The following table points out the most important axis parameters. Please see chapter 4 for a complete list of all axis parameters.

Most important axis parameters					
Number	Axis Parameter	Description	Range [Units]	Default	Access
6	Maximum current	Motor current used when motor is running. The maximum value is 31 which means 100% of the maximum current of the module.	$0 \ldots 31$	24	RWE

Number	Axis Parameter	Description	Range [Units]	Default	Access
7	Standby current	The current used when the motor is not running. The maximum value is 31 which means 100% of the maximum current of the module. This value should be as low as possible so that the motor can cool down when it is not moving.	$0 \ldots 31$	3	RWE
140	Microstep Resolution	Microstep resolutions per full step: 0 - fullstep 1 - halfstep 2 - 4 microsteps 3 - 8 microsteps 4 - 16 microsteps 5 - 32 microsteps 6 - 64 microsteps 7 - 128 microsteps 8 - 256 microsteps	$0 \ldots 8$	4	RWE
141	Microstep Interpolation	Interpolation of the current microstep resolution to 256 microsteps: 0 - No interpolation 1 - Interpolation to 256 microsteps	$0 \ldots 1$	1	RWE

Table 1: Most important Axis Parameters

3 TMCL and the TMCL-IDE — An Introduction

As with most TRINAMIC modules the software running on the microprocessor of the PD57/60-1076 consists of two parts, a boot loader and the firmware itself. Whereas the boot loader is installed during production and testing at TRINAMIC and remains untouched throughout the whole lifetime, the firmware can be updated by the user. New versions can be downloaded free of charge from the TRINAMIC website (http://www.trinamic.com).
The PD57/60-1076 supports TMCL direct mode (binary commands).
In direct mode the TMCL communication over RS-232, RS-485, CAN, and USB follows a strict master/slave relationship. That is, a host computer (e.g. PC/PLC) acting as the interface bus master will send a command to the PD57/60-1076. The TMCL interpreter on the module will then interpret this command, do the initialization of the motion controller, read inputs and write outputs or whatever is necessary according to the specified command. As soon as this step has been done, the module will send a reply back over the interface to the bus master. Only then should the master transfer the next command.

Normally, the module will just switch to transmission and occupy the bus for a reply, otherwise it will stay in receive mode. It will not send any data over the interface without receiving a command first. This way, any collision on the bus will be avoided when there are more than two nodes connected to a single bus.
The Trinamic Motion Control Language [TMCL] provides a set of structured motion control commands. Every command has a binary representation and a mnemonic. The binary format is used to send commands from the host to a module in direct mode, whereas the mnemonic format is used for easy usage of the commands when developing standalone TMCL applications using the TMCL-IDE (IDE means Integrated Development Environment).

There is also a set of configuration variables for the axis and for global parameters which allow individual configuration of nearly every function of a module. This manual gives a detailed description of all TMCL commands and their usage.

3.1 Binary Command Format

Every command has a mnemonic and a binary representation. When commands are sent from a host to a module, the binary format has to be used. Every command consists of a one-byte command field, a one-byte type field, a one-byte motor/bank field and a four-byte value field. So the binary representation of a command always has seven bytes. When a command is to be sent via RS-232, RS-485, RS-422 or USB interface, it has to be enclosed by an address byte at the beginning and a checksum byte at the end. In these cases it consists of nine bytes.

The binary command format with RS-232, RS-485, RS-422 and USB is as follows:

TMCL Command Format	
Bytes	Meaning
1	Module address
1	Command number
1	Type number
1	Motor or Bank number
4	Value (MSB first!)
1	Checksum

Table 2: TMCL Command Format

Info	The checksum is calculated by accumulating all the other bytes using an 8-bit addition.

Note	When using the CAN interface, leave out the address byte and the checksum byte. With CAN, the CAN-ID is used as the module address and the checksum is not needed because CAN bus uses hardware CRC checking.

3.1.1 Checksum Calculation

As mentioned above, the checksum is calculated by adding up all bytes (including the module address byte) using 8-bit addition. Here are two examples which show how to do this:

Checksum calculation in C:

```
unsigned char i, Checksum;
unsigned char Command[9];

//Set the Command array to the desired command
Checksum = Command[0];
for(i=1; i<8; i++)
      Checksum+=Command[i];

   Command[8]=Checksum; //insert checksum as last byte of the command
//Now, send it to the module
```

Checksum calculation in Delphi:

```
var
  i, Checksum: byte;
  Command: array[0..8] of byte;

  //Set the Command array to the desired command

  //Calculate the Checksum:
  Checksum:=Command[0];
  for i:=1 to 7 do Checksum:=Checksum+Command[i];
  Command[8]:=Checksum;
  //Now, send the Command array (9 bytes) to the module
```

3.2 Reply Format

Every time a command has been sent to a module, the module sends a reply. The reply format with RS-232, RS-485, RS-422 and USB is as follows:

TMCL Reply Format	
Bytes	Meaning
1	Reply address
1	Module address
1	Status (e.g. 100 means no error)
1	Command number
4	Value (MSB first!)
1	Checksum

Table 3: TMCL Reply Format

Info The checksum is also calculated by adding up all the other bytes using an 8-bit addition. Do not send the next command before having received the reply!

Note When using CAN interface, the reply does not contain an address byte and a checksum byte. With CAN, the CAN-ID is used as the reply address and the checksum is not needed because the CAN bus uses hardware CRC checking.

3.2.1 Status Codes

The reply contains a status code. The status code can have one of the following values:

TMCL Status Codes	
Code	Meaning
100	Successfully executed, no error
101	Command loaded into TMCL program EEPROM
1	Wrong checksum
2	Invalid command
3	Wrong type
4	Invalid value
5	Configuration EEPROM locked
6	Command not available

Table 4: TMCL Status Codes

3.3 TMCL Command Overview

This sections gives a short overview of all TMCL commands.

Overview of all TMCL Commands			
Command	Number	Parameter	Description
SAP	5	<parameter>, <motor number>, <value>	Set axis parameter (motion control specific settings)
GAP	6	<parameter>, <motor number>	Get axis parameter (read out motion control specific settings)
STAP	7	<parameter>, <motor number>, <value>	Store axis parameter (store motion control specific settings)
RSAP	8	<parameter>, <motor number>	Restore axis parameter (restore motion control specific settings)
GIO	15	<port number>, <bank number>	Get value of analog/digital input

Table 5: Overview of all TMCL Commands

3.4 TMCL Commands by Subject

3.4.1 Parameter Commands

These commands are used to set, read and store axis parameters or global parameters. Axis parameters can be set independently for each axis, whereas global parameters control the behavior of the module itself. These commands can also be used in direct mode and in standalone mode.

Parameter Commands		
Mnemonic	Command number	Meaning
SAP	5	Set axis parameter
GAP	6	Get axis parameter
STAP	7	Store axis parameter
RSAP	8	Restore axis parameter

Table 6: Parameter Commands

3.4.2 I/O Port Commands

These commands control the external I/O ports and can be used in direct mode as well as in standalone mode.

I/O Port Commands		
Mnemonic	Command number	Meaning
GIO	15	Get input

Table 7: I/O Port Commands

3.5 Detailed TMCL Command Descriptions

The module specific commands are explained in more detail on the following pages. They are listed according to their command number.

3.5.1 SAP (Set Axis Parameter)

With this command most of the motion control parameters of the module can be specified. The settings will be stored in SRAM and therefore are volatile. That is, information will be lost after power off.

Info For a table with parameters and values which can be used together with this command please refer to section 4.

Internal function: The specified value is written to the axis parameter specified by the parameter number.
Related commands: GAP, AAP.

Mnemonic: SAP <parameter number>, <axis>, <value>

Binary representation

Binary Representation			
Instruction	Type	Motor/Bank	Value
5	see chapter 4	0	<value>

Reply in Direct Mode	
Status	Value
100 - OK	don't care

Example Set the maximum positioning speed for motor 0 to 51200 pps.
Mnemonic: SAP 4, 0, 51200.

Binary Form of SAP 4, 0, 51200	
Field	Value
Target address	01_h
Instruction number	05_h
Type	04_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	$C8_h$
Value (Byte 0)	00_h
Checksum	$D2_h$

3.5.2 GAP (Get Axis Parameter)

Most motion / driver related parameters of the PD57/60-1076 can be adjusted using e.g. the SAP command. With the GAP parameter they can be read out. In standalone mode the requested value is also transferred to the accumulator register for further processing purposes (such as conditional jumps). In direct mode the value read is only output in the value field of the reply, without affecting the accumulator.

Info For a table with parameters and values that can be used together with this command please refer to section 4.

Internal function: The specified value gets copied to the accumulator.
Related commands: SAP, AAP.

Mnemonic: GAP <parameter number>, <axis>

Binary Representation			
Instruction	Type	Motor/Bank	Value
6	see chapter 4	0	<value>

Reply in Direct Mode	
Status	Value
100 - OK	value read by this command

Example
Get the actual position of motor 0.
Mnemonic: GAP 1, 0.

Binary Form of GAP 1, 0	
Field	Value
Target address	01_h
Instruction number	06_h
Type	01_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	00_h
Checksum	08_h

3.5.3 STAP (Store Axis Parameter)

This command is used to store TMCL axis parameters permanently in the EEPROM of the module. This command is mainly needed to store the default configuration of the module. The contents of the user variables can either be automatically or manually restored at power on.

Info For a table with parameters and values which can be used together with this command please refer to dection 4.

Internal function: The axis parameter specified by the type and bank number will be stored in the EEPROM.

Related commands: SAP, AAP, GAP, RSAP.

Mnemonic: STAP <parameter number>, <bank>

Binary Representation			
Instruction	Type	Motor/Bank	Value
7	see chapter 4	0	0 (don't care)

Reply in Direct Mode	
Status	Value
100 - OK	0 (don't care)

Example
Store axis parameter #6.
Mnemonic: STAP 7, 6.

Binary Form of STAP 6, 12	
Field	Value
Target address	01_h
Instruction number	07_h
Type	06_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	00_h
Checksum	$0E_h$

3.5.4 RSAP (Restore Axis Parameter)

With this command the contents of an axis parameter can be restored from the EEPROM. By default, all axis parameters are automatically restored after power up. An axis parameter that has been changed before can be reset to the stored value by this instruction.

Info For a table with parameters and values which can be used together with this command please refer to section 4.

Internal function: The axis parameter specified by the type and bank number will be restored from the EEPROM.

Related commands: SAP, AAP, GAP, RSAP.

Mnemonic: RSAP <parameter number>, <bank>

Binary Representation			
Instruction	Type	Motor/Bank	Value
8	see chapter 4	0	0 (don't care)

Reply in Direct Mode	
Status	Value
100 - OK	0 (don't care)

Example
Restore axis parameter #6.
Mnemonic: RSAP 8, 6.

Binary Form of RSAP 8, 6	
Field	Value
Target address	01_h
Instruction number	08_h
Type	06_h
Motor/Bank	00_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	00_h
Checksum	$0A_h$

3.5.5 GIO (Get Input)

With this command the status of the available general purpose outputs of the module can be read. The function reads a digital or an analog input port. Digital input ports will read as 0 or 1. In standalone mode the requested value is copied to the accumulator register for further processing purposes such as conditional jumps. In direct mode the value is only output in the value field of the reply, without affecting the accumulator. The actual status of a digital output line can also be read.

Internal function: The state of the i/o line specified by the type parameter and the bank parameter is read.

Related commands: SIO.

Mnemonic: GIO <port number>, <bank number>

Binary Representation			
Instruction	Type	Motor/Bank	Value
15	<port number>	<bank number> (0/1/2)	0 (don't care)

Reply in Direct Mode	
Status	Value
100 - OK	status of the port

Example
Get the value of ADC channel 0.
Mnemonic: GIO 0, 1.

Binary Form of GIO 0, 1	
Field	Value
Target address	01_h
Instruction number	$0F_h$
Type	00_h
Motor/Bank	01_h
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	00_h
Value (Byte 0)	00_h
Checksum	11_h

Reply (Status=no error, Value=302)	
Field	Value
Host address	02_h
Target address	01_h
Status	64_h
Instruction	$0F_h$
Value (Byte 3)	00_h
Value (Byte 2)	00_h
Value (Byte 1)	01_h
Value (Byte 0)	$2E_h$
Checksum	$A5_h$

Bank 0 – Digital Inputs
The digital input states can be accessed in bank 0.

Digital Inputs in Bank 0			
Port	Description	Command	Range
0 - CHOP	Chopper selection	GIO 0, 0	0/1
1 - Enable	Enable module	GIO 1, 0	0/1

3.5.6 Customer specific Command Extensions (UF0. . . UF7 – User Functions)

These commands are used for customer specific extensions of TMCL. They will be implemented in C by Trinamic. Please contact the sales department of Trinamic Motion Control GmbH & Co KG if you need a customized TMCL firmware.

Related commands: none.

Mnemonic: UF0. . . UF7

Binary Representation			
Instruction	Type	Motor/Bank	Value
64. . . 71	<user defined>	0 <user defined>	0 <user defined>

Reply in Direct Mode	
Status	Value
100 - OK	user defined

4 Axis Parameters

Most motor controller features of the PD57/60-1076 module are controlled by axis parameters. Axis parameters can be modified or read using SAP, GAP and AAP commands. Some axis parameters can also be stored to or restored from the EEPROM using STAP and RSAP commands. This chapter describes all axis parameters that can be used on the PD57/60-1076 module.

Axis 0 Parameters of the PD57/60-1076 Module					
Number	Axis Parameter	Description	Range [Units]	Default	Access
6	Maximum current	Motor current used when motor is running. The maximum value is 31 which means 100% of the maximum current of the module.	$0 \ldots 31$	24	RWE
7	Standby current	The current used when the motor is not running. The maximum value is 31 which means 100% of the maximum current of the module. This value should be as low as possible so that the motor can cool down when it is not moving.	$0 \ldots 31$	3	RWE
9	Standby current delay	Controls the number of clock cycles for motor power down after a motion as soon as standstill is detected and TZEROWAIT has expired. 0 means instant power down and 1-15 is the delay pero current reduction step in multiple of 2^{18} clocks.	$0 \ldots 15$	0	RWE
22	Speed threshold for CoolStep / fullstep	TSTEP corresponding to the speed threshold for de-activating CoolStep™or switching to fullstep mode.	$0 \ldots 1048575$	0	RWE
135	TStep	Actual measured time between two 1/256 microsteps. This value during a certain speed can be written to TPWMTHRS.	$0 \ldots 1048575$	0	RWE
136	StealthChop VLimit	Direct access to the TPWMTHRS register (0x13 of the TMC5130) for the StealthChop Velocity Limit Configuration. Use STGP and RSGP to access the stored value in the EEPROM.	$0 \ldots 1048575$	0	RWE
137	PWMConf	Direct access to the PWMConf register (0x70 of the TMC5130) for the StealthChop Configuration. Use STGP and RSGP to access the stored value in the EEPROM.	$0 \ldots 4294967295$	328136	RWE

Number	Axis Parameter	Description	Range [Units]	Default	Access
138	CoolConf	Access to the COOLCONF register (0x6D of the TMC5130) for the CoolStep Configuration. Use STGP and RSGP to access the stored value in the EEPROM.	$0 \ldots 4294967295$	33011	RWE
139	CHOPConf	Access to the CHOPCONF register (0x6C of the TMC5130) for the Chopper Configuration (excluded Microstep Configuration). Use STGP and RSGP to access the stored value in the EEPROM.	$0 \ldots 4294967295$	33011	RWE
140	Microstep Resolution	Microstep resolutions per full step: 0 - fullstep 1 - halfstep 2 - 4 microsteps 3 - 8 microsteps 4 - 16 microsteps 5 - 32 microsteps 6 - 64 microsteps 7 - 128 microsteps 8 - 256 microsteps	$0 \ldots 8$	4	RWE
141	Microstep Interpolation	Interpolation of the current microstep resolution to 256 microsteps: 0 - No interpolation 1 - Interpolation to 256 microsteps	$0 \ldots 1$	1	RWE
142	Double Edge Steps	Step impulse at each step edge to reduce step frequency requirement. 0 - Single edge 1 - Double edge	$0 \ldots 1$	0	RWE
168	SmartEnergy current minimum (SEIMIN)	Sets the lower motor current limit for CoolStep operation by scaling the maximum current (see axis parameter 6) value. 0 - 1/2 of CS 1 - 1/4 of CS	$0 \ldots 1$	0	RWE
169	SmartEnergy current down step	Sets the number of StallGuard2 readings above the upper threshold necessary for each current decrement of the motor current. Number of StallGuard2 measurements per decrement: Scaling: 0. . . 3: 32, 8, 2, 1. 0: slow decrement, 3: fast decrement	$0 \ldots 3$	0	RW

Number	Axis Parameter	Description	Range [Units]	Default	Access
170	SmartEnergy hysteresis	Sets the distance between the lower and the upper threshold for StallGuard2 reading. Above the upper threshold the motor current becomes decreased. Hysteresis: ([AP172] + 1) * 32. Upper StallGuard threshold: ([AP172] + [AP170] + 1) * 32	$0 \ldots 15$	0	RW
171	SmartEnergy current up step	Sets the current increment step. The current becomes incremented for each measured StallGuard2 value below the lower threshold see SmartEnergy hysteresis start). Current increment step size: Scaling: 0. . . 3: 1, 2, 4, 8. 0: slow increment, 3: fast increment / fast reaction to rising load	$0 \ldots 3$	0	RW
172	SmartEnergy hysteresis start	The lower threshold for the StallGuard2 value (see SmartEnergy current up step).	$0 \ldots 15$	0	RW
173	StallGuard2 filter enable	Enables the StallGuard2 filter for more precision of the movement. If set, reduces the measurement frequency to one measurement per four fullsteps. In most cases it is expedient to set the filtered mode before using CoolStep. Use the standard mode for step loss detection. 0 - standard mode 1 - filtered mode	$0 \ldots 1$	0	RW
174	StallGuard2 threshold	This signed value controls StallGuard2 threshold level for stall output and sets the optimum measurement range for readout. A lower value gives a higher sensitivity. Zero is the starting value. A higher value makes StallGuard2 less sensitive and requires more torque to indicate a stall.	$-64 \ldots 63$	0	RW
180	SmartEnergy actual current	This status value provides the actual motor current setting as controlled by CoolStep. The value goes up to the CS value and down to the portion of CS as specified by SEIMIN. Actual motor current scaling factor: 0. . . 31: 1/32, 2/32, . . . 32/32	$0 \ldots 31$	0	R
182	SmartEnergy threshold speed	Above the speed which corresponds to this TSTEP value, CoolStep becomes enabled.	$0 \ldots 1048575$	0	RWE

Number	Axis Parameter	Description	Range [Units]	Default	Access
206	Load value	Actual current control scaling for monitoring smart energy current scaling or automatic current scaling.	$0 \ldots 1023$	0	R

Table 8: All PD57/60-1076 Axis 0 Parameters

5 Module Specific Configuration

This section explains how to use the two digital inputs of the PD57/60-1076. It also describes the default configuration for microstep resolution and interpolation as well as for current setting.

5.1 Digital Inputs

The PD57/60-1076 includes two digital inputs: ENABLE and CHOP. The ENABLE input is a low active input which powers the motor when the ENABLE is not set.

The CHOP input allows the user to choose between different chopper modes. When CHOP is cleared, the chopper mode is SpreadCycle. When CHOP is set, the active chopper is StealthChop. The change from SpreadCycle to StealthChop is only allowed when the motor is powered and in still stand. The chopper mode will remain SpreadCycle until both conditions are fulfilled.

NOTICE	Note that the voltage at the digital inputs can be reversed depending on the reference voltage of the optocoupler. See the Hardware Manual of PD57/60-1076 for more details.

NOTICE	In the TMC5160 Manual, available at www.trinamic.com, you can find more information about how to configure StealthChop and SpreadCycle.

5.2 Default Configuration

By default, the PD57/60-1076 uses the configuration shown in Section 4 for each of the parameters. In the following table we show several of the variables which can be also stored in the EEPROM permanent storage.

Default Configuration of PD57/60-1076 Parameters			
Name	Command	Default Value	Meaning
Microstep Resolution	SAP/GAP 140, 0	4	Resolution: 16 microsteps per full step.
Microstep Interpolation	SAP/GAP 141, 0	1	Interpolation to 256 microsteps.
Maximum Current	SAP/GAP 6, 0	24	Current when motor is running is 24/31.
Standby Current	SAP/GAP 7, 0	3	Current when motor is not running is 3/31.

Table 9: Default Configuration of PD57/60-1076 Parameters

In summary, with the default configuration:

- The PD57/60-1076 works with 16 microsteps per full step interpolated to 256 microsteps.
- The current when motor is running is 2.3A RMS and 0.290A RMS when not running.

5.3 Reset to Factory Defaults

To reset all settings to their factory defaults, open the Global Parameters tool in the TMCL-IDE and click the Restore Factory Defaults button.

5.4 Configuration with the TMCL-IDE Plugin

When using a PC and the TMCL-IDE, the PD57/60-1076 can be easily configured with the Module Settings plugin. This tool provides a direct graphic interface to the most important parameters of the module.

Figure 3 shows the first tab of the tool, which is called Current. In this tab, the tool shows the current value of the axis parameters 6, 7 and 8, related to the motor current, and allows the user to modify them. The button on the bottom left corner sends a STAP for each of the parameters related to the current tab. In this case, parameters 6, 7 and 8 would be stored in the EEPROM of the module.

The next tab, Microstep, works in a similar way and affects parameters 140, 141 and 142. When this tab is active, these three parameters would be stored in EEPROM after clicking on Store Current Tab.

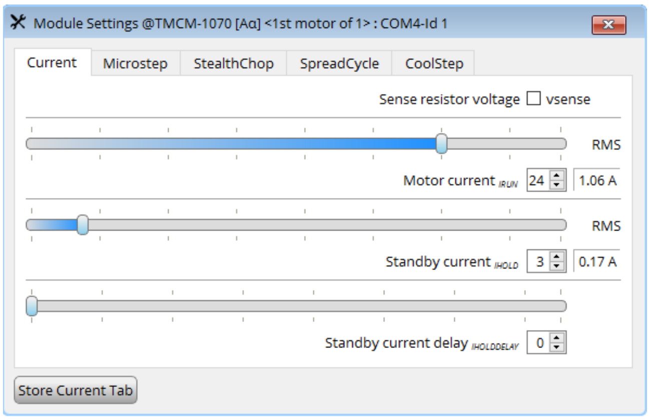


Figure 3: TMCL-IDE Plugin: Tab Current

The StealthChop tab is shown on Figure 4. Parameter 137 "Chopconf" contains the value to be written in the register with the same name. The graphical interface allows the user to change individual fields of the register without having to look up the field bits in the reference manual of TMC5160.

The last field "Upper Velocity Limit" relates to parameter 136. When rotating at the desired speed limit, click on "Read from TSTEP" to fill it automatically with the value from register 135 converted to step per second. The velocity value is approximated because of the possible inaccuracy of the TMC5160 internal clock.

Store Current Tab button stores in this case parameters 136 and 137 in the EEPROM.

SpreadCycle tab works like StealthChop tab for axis parameter 139 and the fields of its related register.

The last tab "CoolStep" is shown on Figure 5. To configure CoolStep™, we offer a more advanced plugin (click on "CoolStep" on the plugin list of the module). The exceptions are parameters 22 and 182, Threshold Speed and Deactivation Threshold Speed respectively. Rotate the motor at the desired speed for each parameter and click on Read from TSTEP to assign the correct value. Parameters 22 and 182 together with 138 (which contains configuration from the CoolStep plugin) will be stored in the EEPROM after a click on Store Current Tab.

NOTICE	In the TMC5160 Manual, available at www.trinamic.com, you can find more information about how to configure StealthChop, SpreadCycle and CoolStep.

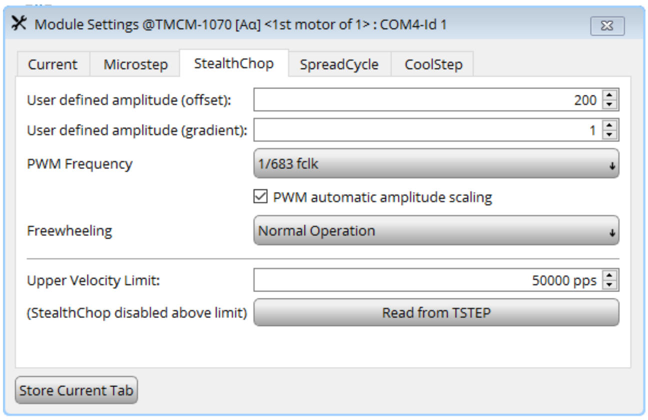


Figure 4: TMCL-IDE Plugin: Tab StealthChop

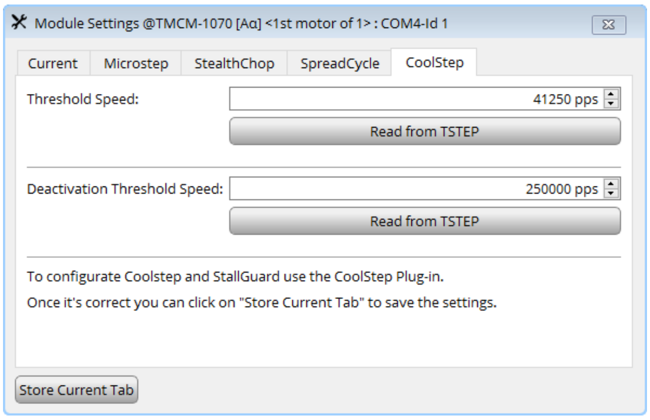


Figure 5: TMCL-IDE Plugin: Tab CoolStep

6 Hints and Tips

This chapter gives some hints and tips on using the functionality of TMCL, for example how to use and parameterize the built-in reference search algorithm. You will also find basic information about StallGuard2™ and CoolStep™ in this chapter.

6.1 StallGuard2

The module is equipped with motor driver chips that feature load measurement. This load measurement can be used for stall detection. StallGuard2 delivers a sensorless load measurement of the motor as well as a stall detection signal. The measured value changes linear with the load on the motor in a wide range of load, velocity and current settings. At maximum motor load the StallGuard value goes to zero. This corresponds to a load angle of 90° between the magnetic field of the stator and magnets in the rotor. This also is the most energy efficient point of operation for the motor.

Stall detection means that the motor will be stopped automatically when the load gets too high. This function is configured mainly using axis parameters #174 and #181.

Stall detection can for example be used for finding the reference point without the need for reference switches. A short routine written in TMCL is needed to use StallGuard for reference searching.

6.2 CoolStep

This section gives an overview of the CoolStep related parameters. Please bear in mind that the figure only shows one example for a drive. There are parameters which concern the configuration of the current. Other parameters are there for velocity regulation and for time adjustment.

Figure 6 shows all the adjustment points for CoolStep. It is necessary to identify and configure the thresholds for current (I6, I7 and I183) and velocity (V182). Furthermore the StallGuard2 feature has to be adjusted (SG170). It can also be enabled if needed (SG181).

The reduction or increasing of the current in the CoolStep area (depending on the load) has to be configured using parameters I169 and I171.

In this chapter only basic axis parameters are mentioned which concern CoolStep and StallGuard2. The complete list of axis parameters in chapter 4 contains further parameters which offer more configuration options.

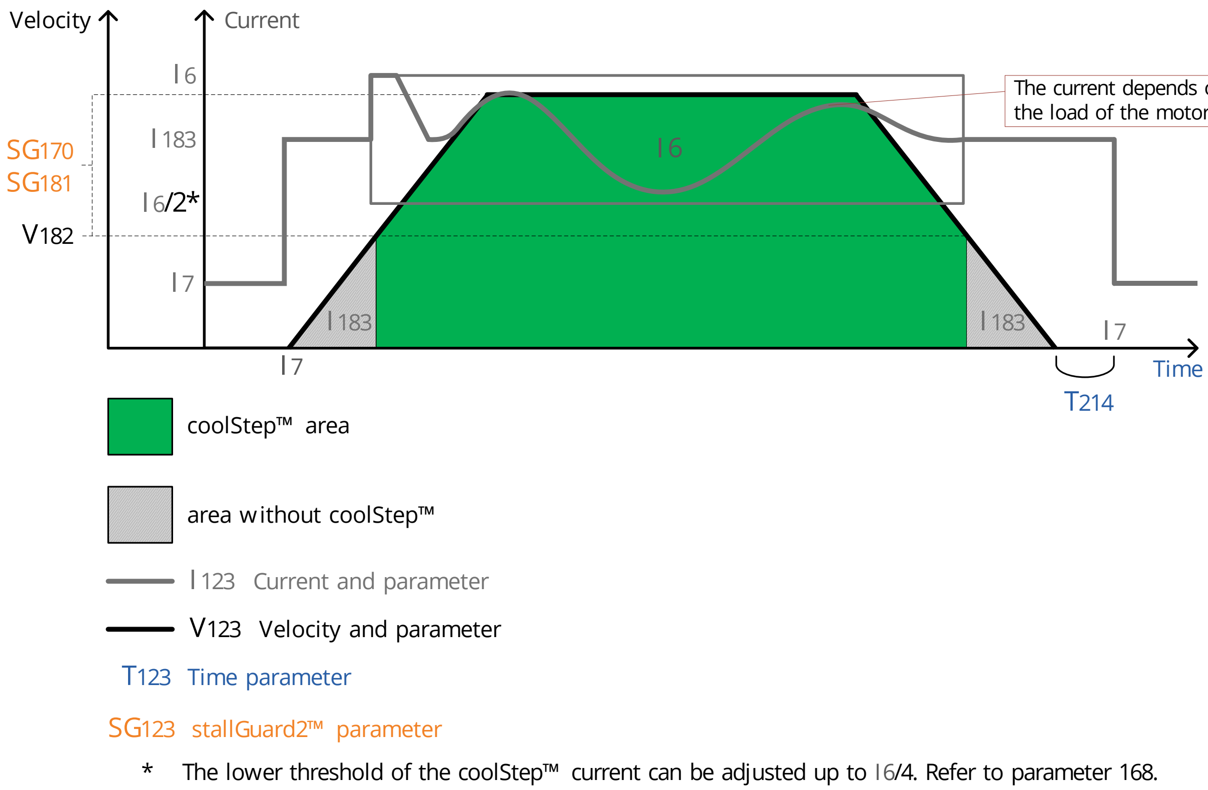


Figure 6: CoolStep Adjustment Points and Thresholds

CoolStep Adjustment Points and Thresholds		
Number	Axis Parameter	Description
I6	Absolute maximum current	The maximum value is 255. This value means 100% of the maximum current of the module. The current adjustment is within the range 0...255 and can be adjusted in 32 steps (0...255 divided by eight; e.g. step 0 = 0...7, step 1 = 8...15 and so on). Too high values may cause motor damage!
I7	Standby current	The current limit two seconds after the motor has stopped.
I168	smartEnergy current minimum	Sets the lower motor current limit for CoolStep operation by scaling the CS (Current Scale, see axis parameter 6) value. Minimum motor current: 0 - 1/2 of CS 1 - 1/4 of CS
I169	smartEnergy current down step	Sets the number of StallGuard2 readings above the upper threshold necessary for each current decrement of the motor current. Number of StallGuard2 measurements per decrement: Scaling: 0...3: 32, 8, 2, 1 0: slow decrement 3: fast decrement
I171	smartEnergy current up step	Sets the current increment step. The current becomes incremented for each measured StallGuard2 value below the lower threshold (see smartEnergy hysteresis start). current increment step size: Scaling: 0...3: 1, 2, 4, 8 0: slow increment 3: fast increment
SG170	smartEnergy hysteresis	Sets the distance between the lower and the upper threshold for StallGuard2 reading. Above the upper threshold the motor current becomes decreased.
SG181	Stop on stall	Below this speed motor will not be stopped. Above this speed motor will stop in case StallGuard2 load value reaches zero.
V182	smartEnergy threshold speed	Above this speed CoolStep becomes enabled.
T214	Power down delay	Standstill period before the current is changed down to standby current. The standard value is 200 (which means 2000msec).

Table 10: CoolStep Adjustment Points and Thresholds

7 Figures Index

8 Tables Index

9 Supplemental Directives

9.1 Producer Information

9.2 Copyright

9.3 Trademark Designations and Symbols

Trademark designations and symbols used in this documentation indicate that a product or feature is owned and registered as trademark and/or patent either by TRINAMIC or by other manufacturers, whose products are used or referred to in combination with TRINAMIC's products and TRINAMIC's product documentation.

This TMCL™ Firmware Manual is a non-commercial publication that seeks to provide concise scientific and technical user information to the target user. Thus, trademark designations and symbols are only entered in the Short Spec of this document that introduces the product at a quick glance. The trademark designation /symbol is also entered when the product or feature name occurs for the first time in the document. All trademarks and brand names used are property of their respective owners.

9.4 Target User

The documentation provided here, is for programmers and engineers only, who are equipped with the necessary skills and have been trained to work with this type of product.

The Target User knows how to responsibly make use of this product without causing harm to himself or others, and without causing damage to systems or devices, in which the user incorporates the product.

9.5 Disclaimer: Life Support Systems

TRINAMIC Motion Control GmbH & Co. KG does not authorize or warrant any of its products for use in life support systems, without the specific written consent of TRINAMIC Motion Control GmbH & Co. KG.

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.

Information given in this document is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. Specifications are subject to change without notice.

9.6 Disclaimer: Intended Use

The data specified in this user manual is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given.

In particular, this also applies to the stated possible applications or areas of applications of the product. TRINAMIC products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (safety-Critical Applications) without TRINAMIC's specific written consent.

TRINAMIC products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by TRINAMIC. TRINAMIC conveys no patent, copyright, mask work right or other trade mark right to this product. TRINAMIC assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

9.7 Collateral Documents & Tools

This product documentation is related and/or associated with additional tool kits, firmware and other items, as provided on the product page at: www.trinamic.com.

10 Revision History

10.1 Firmware Revision

Version	Date	Author	Description
V1.02	2018-MAR-26	OK	First release.

Table 11: Firmware Revision

10.2 Document Revision

Version	Date	Author	Description
V1.00	2018-MAR-26	OK	First release.
V1.01	2019-JAN-25	OK	Reset to factory default description changed. Superfluous global parameters removed.
V1.02	2019-APR-23	OK	Superfluous sentence in first chapter removed.

Table 12: Document Revision