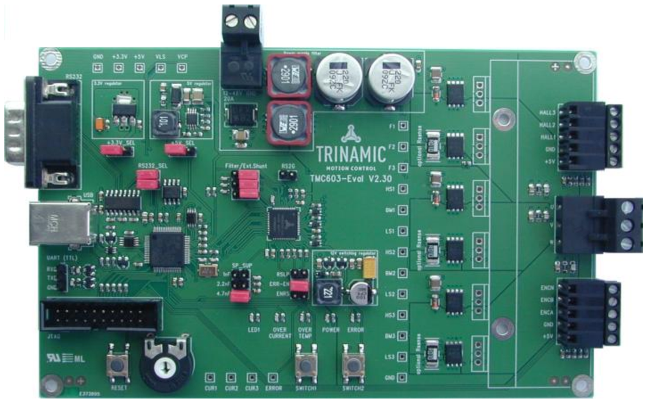
EVALUATION FOR ICs

EVALUATION BOARD

Firmware Version V2.05

EVAL BOARD MANUAL



TMC603-EVAL

Evaluation board for TMC603
Three Phase Motor Driver
18.8A RMS / 12… 48V DC
RS232 / UART, USB
Hall Sensor Interface
Encoder Interface
FOC Firmware

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

Table of Contents

1 Features

The TMC603 evaluation board makes it possible to evaluate the features of the TMC603 three phase BLDC motor driver with hall sensor interface. On the evaluation board the STM32F ARM Cortex-M3 microcontroller is used to control the TMC603. The FLASH memory of the microcontroller contains a program which configures the TMC603 and controls the communication with the PC via the USB interface and the RS232 interface. The PC software is based on Windows and allows tuning of all operation parameters for every three phase BLDC motor.

Application
- Evaluation of the features of the TMC603 three phase motor driver

Electrical data
- Motor current: up to 18.8A RMS nominal motor current
- Supply voltage: 12V… 48V operating voltage

Interfaces
- RS232 (UART)
- USB (type B)
- Inputs for encoder (ABN)
- Three digital hall sensors

Motor type
- Three phase BLDC motor
- Commutation with space vector PWM (SVPWM) based on hall sensor or encoder feedback
- Rotor position feedback (encoder or hall sensor)

Safety features
- Overcurrent / short to GND and undervoltage protection with diagnostics integrated

Software
- TMCL (TMCL-IDE and TMCM-BLDC)
- Firmware update via USB and RS232

Highlights
- Integrated current measurement using power MOS transistor RDSon
- Integrated break-before-make logic. No special microcontroller PWM hardware required
- EMV optimized current controlled gate drivers
- Internal Qgd (Gate-Drain Charge) protection supports latest generation of power MOSFETs
- Integrated supply concept with step down switching regulator
- Common rail charge pump allows for 100% PWM duty cycle

2 Order Codes

Order code	Description	Size of unit [mm^3]
TMC603-EVAL	Evaluation board for TMC603 three phase motor pre-driver	160 x 100 x 13.5

Table 2.1 Order codes

3 Hardware

3.1 Mechanical and Electrical Interfacing

3.1.1 Size of TMC603-EVAL and Mounting Holes

The board dimensions are 160mm x 100mm. Maximum component height (height above PCB level) without mating connectors is 13.5mm. There are six mounting holes (hole diameter: 3.2mm) suitable for M3 screws.

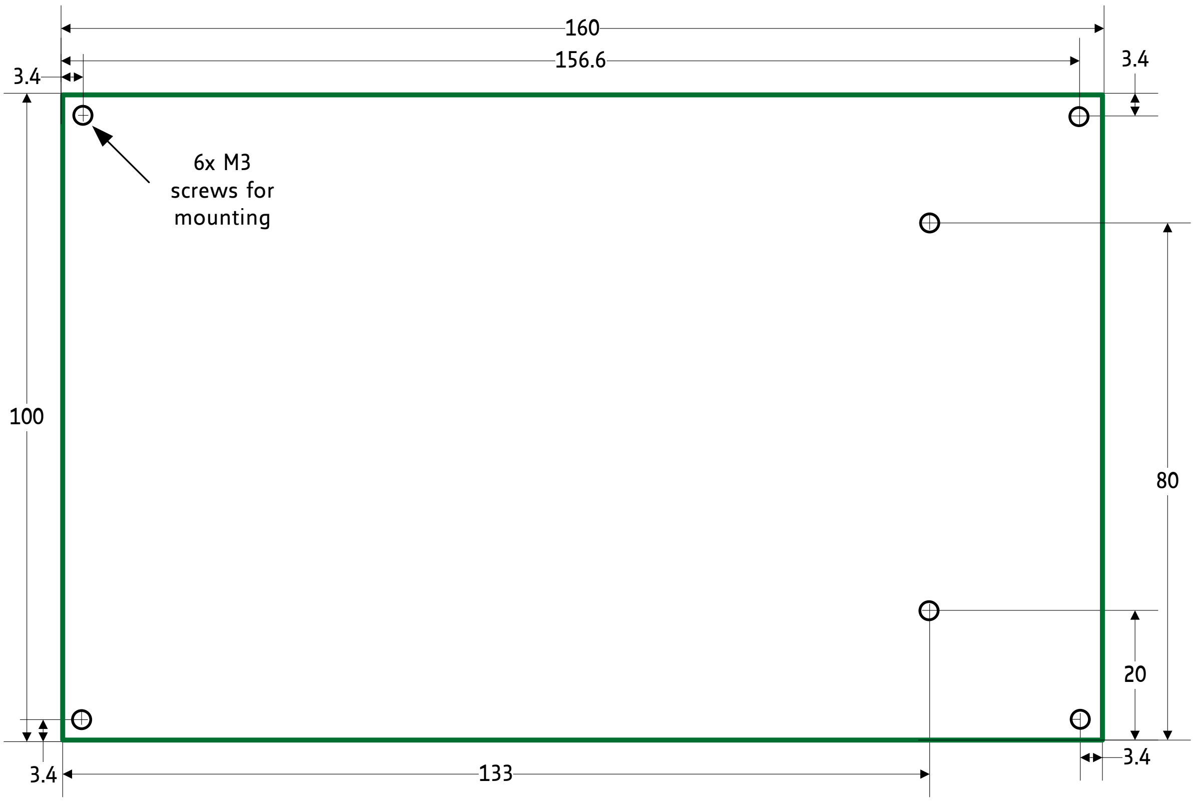


Figure 3.1 TMC603-EVAL dimensions

3.1.2 Connectors

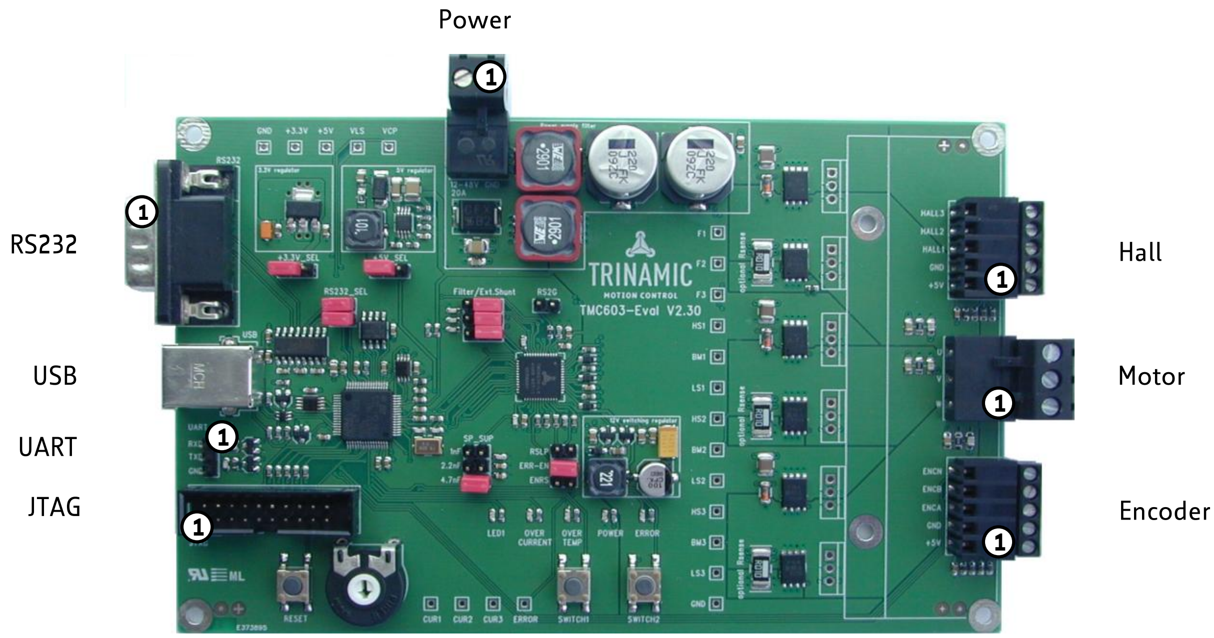


Figure 3.2 Connectors of TMC603-EVAL

CONNECTORS OF TMC603-EVAL

Label	Connector type	Mating connector type
Power	RIA 330-02, 2 pol., 5mm pitch, shrouded header	RIA 349-2, screw type terminal block, pluggable, centerline 5 mm / 0.197 inches, wire entry parallel to plug direction
Motor	RIA 330-03, 3 pol., 5mm pitch, shrouded header	RIA 349-3, screw type terminal block, pluggable, centerline 5 mm / 0.197 inches, wire entry parallel to plug direction
Encoder	RIA 182-05, 5 pol., 3.5mm pitch, header	RIA 169-05, screw type terminal block, pluggable, centerline 3.5 mm / 0.138 inches, wire entry parallel to plug direction
Hall sensor	RIA 182-05, 5 pol., 3.5mm pitch, header	RIA 169-05, screw type terminal block, pluggable, centerline 3.5 mm / 0.138 inches, wire entry parallel to plug direction
USB	USB, type B, 4 pol., female	USB, type B, 4 pol., male
UART	Multi-pin-connector, 3 pol., 2.54mm pitch	Female connector with 2.54mm pitch
RS232	DSUB, vertical, 9 pol., male, US-type	DSUB, 9 pol., female
JTAG	Low profile box header without locking bar, type 8380, 20 pol., DIN 41651, 2.54mm pitch	Low profile IDC socket connector, 20pol., DIN41651, 2.54mm pitch

Table 3.1 Connectors

3.1.2.1 Power Connector

Pin	Label	Direction	Description
1	GND	Power (GND)	Power supply and signal ground
2	+UB	Power (Supply input)	Supply voltage: +12… +48V DC

Table 3.2 Power connector

3.1.2.2 Motor Connector

Pin	Label	Direction	Description
1	W	Output	Motor coil connection W
2	V	Output	Motor coil connection V
3	U	Output	Motor coil connection U

Table 3.3 Motor connector

3.1.2.3 Hall Sensor Connector

Pin	Label	Direction	Description
1	+5V	Output	Power supply output for hall sensor, nom. +5V DC
2	GND	GND	Power supply and signal ground
3	HALL1	Input (5V TTL)	Hall sensor 1
4	HALL2	Input (5V TTL)	Hall sensor 2
5	HALL3	Input (5V TTL)	Hall sensor 3

Table 3.4 Hall sensor connector

3.1.2.4 Encoder Connector

Pin	Label	Direction	Description
1	+5V	Output	Power supply output for hall sensor, nom. +5V DC
2	GND	GND	Power supply and signal ground
3	ENC_A	Input	Encoder signal A
4	ENC_B	Input	Encoder signal B
5	ENC_N	Input	Encoder signal N

Table 3.5 Encoder connector

3.1.2.5 USB Connector

Pin	Label	Description
1	+5V	Board is self-powered – just use to detect availability of attached host system (e.g. PC)
2	USB-	Differential USB bus
3	USB+	Differential USB bus
4	GND	System and module ground

Table 3.6 USB connector

3.1.2.6 RS232 Connector

Pin	Label	Description
2	RXD	Received data line
3	TXD	Transmitted data line
5	GND	RS232 signal and system ground

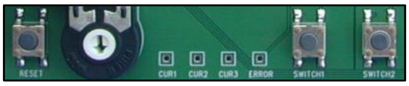
Table 3.7 RS232 connector

3.1.2.7 UART Connector

Pin	Label	Description
1	RXD	Received data line
2	TXD	Transmitted data line
3	GND	Signal and system ground

Table 3.8 UART connector

3.1.3 Switches and Potentiometer



Reset Potentiometer Switch1 Switch2

Figure 3.3 Switches and potentiometer

Label	Description
Reset	The reset switch is connected to the NRST pin of the µC. Press it to reset the module.
Potentiometer	The potentiometer is connected to pin PC5 of the µC (ADC_IN15/PC5). It can be used customer specific with GIO command.
Switch1	Switch1 is connected to pin PB1 of the µC (PB1/ADC_IN9/T3_CH4). The switch can be used customer specific with GIO command.
Switch2	Switch2 is connected to pin PB0 of the µC (PB0/ADC_IN8/T3_CH3). The switch can be used customer specific with the GIO command.

Table 3.9 Switches and potentiometer

3.1.4 Jumpers

Jumper	Label	Description
Select +3.3V supply	+3.3V_SEL	Jumper pins: Result 1, 2: +5V supply voltage of additional switching regulator (*default*). Precondition: +5V_SEL jumper on pin 1 and 2. 2, 3: +12V low side driver supply voltage of TMC603.
Select +5V supply	+5V_SEL	Jumper pins: Result 1, 2: +5V supply voltage of additional switching regulator (*default*) 2, 3: +5V Internal supply voltage of TMC603
Short to GND control resistor	RS2G	The short to GND control resister controls the delay time between switching on the high side MOSFET and the short to GND check. Jumper plugged: R_{S2G} = 470kΩ (t_{S2G} = 2µs) Jumper unplugged: R_{S2G} = 1.47MΩ (t_{S2G} = 6µs) *default*
Slope control resistor	RSLP	The slope control resistor sets output current for MOSFET drivers. Jumper plugged: R_{SLP} = 100kΩ (I_{gate} = 100mA) Jumper unplugged: R_{SLP} = 147kΩ (I_{gate} = 68mA) *default*
Error enable	ERR-EN	Jumper plugged: In case of error (e.g. short circuit on output) the TMC603 turns off itself by hardware. Jumper unplugged: Without jumper the shutdown has to be realized by the processor.

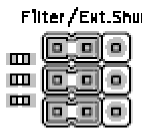
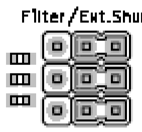
Jumper	Label	Description
Enable/disable shunt resistors for current measurement	ENRS	Set this jumper together with the three jumpers Filter/Ext_Shunt. There are two possibilities. - Set ENRS jumper together with three jumpers Filter/Ext_Shunt on pin position 2-3 (extern shunt). Now, the measurement of the motor current will be done with the shunt resistors on the board. - If the ENRS jumper is not set and the three jumpers Filter/Ext_Shunt are set on pin position 1-2 (filter position), there will be the filtered, measured motor voltages on the measuring points F1, F2, and F3.
Spike suppression time control capacitor	SP_SUP	An external capacitor on this pin controls the commutation spike suppression time for hall*FX™*. This pin charges the capacitor via an internal current source. All jumpers unplugged C_{SP_SUP} = 470pF (t_{SP_SUP} = 47µs) Jumper 1nF plugged C_{SP_SUP} = 1.47nF (t_{SP_SUP} = 147µs) Jumper 2.2nF plugged C_{SP_SUP} = 2.67nF (t_{SP_SUP} = 267µs) Jumper 1nF and 2.2nF C_{SP_SUP} = 3.67nF (t_{SP_SUP} = 367µs) Jumper 4.7nF plugged C_{SP_SUP} = 5.17nF (t_{SP_SUP} = 517µs) *Default* Jumper 1nF and 4.7nF C_{SP_SUP} = 6.17nF (t_{SP_SUP} = 617µs) Jumper 2.2nF and 4.7nF C_{SP_SUP} = 7,37nF (t_{SP_SUP} = 737µs) All jumpers plugged C_{SP_SUP} = 8,37nF (t_{SP_SUP} = 837µs)
Filter output / external sense resistor input	Filter/Ext.Shunt	Selects output signal of internal switched capacitor filters or input for external sense resistor (*default*). When the internal R_{DSon} of optional MOSFETs in TO-220 package are used to measure the actual motor current, set the jumper to pin 1 and 2. In that case, the filter outputs of TMC603 are  connected to the measuring points Fx (refer chapter 9). In addition, set the axis parameter 225 to zero (refer chapter 5), to select the current measurement by using the internal R_{DSon} of optional MOSFETs. When the external sense resistors are used to measure the actual motor current, set the jumpers to pin 2 and 3 (*default*). In that case, the external sense resistors are connected to the analog  inputs RSx of TMC603 (refer TMC603 datasheet). In addition, set the axis parameter 225 to one (refer chapter 5), to select the current measurement by using the external sense resistor.
RS232 select	RS232_SEL	Both jumpers plugged RS232 usable on D-sub connector Both jumper unplugged UART TTL usable on D-sub connector

Table 3.10 Jumpers

3.1.5 LEDs

LEDS OF THE TMC603-EVAL

Status	Label	Description
Error signal	ERROR	This red LED lights up upon an error is occurred by undervoltage of VLS or VCP as well as by short to ground of the power MOS half bridge.
Power on	POWER	This green LED lights up upon the power supply is working.
Temperature warning	OVER TEMP	This red LED flashes upon the power state has exceeded a critical temperature of 100°C (prewarning). This red LED lights up upon the power stage has exceeded a critical temperature of 120°C. The motor becomes switched off, until temperature falls below 110°C.
Current limit	OVER CURRENT	This red LED lights up upon the motor current has exceeded the current limit.
LED without defined function	LED1	This yellow LED is applicable and can be used customer specific. The SIO and GIO commands can be used for setting and read out.

Table 3.11 LEDs

3.1.6 Measuring Points

MEASURING POINTS OF THE TMC603-EVAL

Measuring point	Label	Description
Charge pump supply voltage	VCP	Charge pump supply voltage. Provides high side driver supply.
Low side driver supply voltage	VLS	Low side driver supply voltage for driving low side gates.
Power supply +5V DC	+5V	Power supply, nom. +5V DC
Power supply +3.3V DC	+3.3V	Power supply, nom. +3.3V DC
Ground	GND	Power supply and signal ground. There are two ground pins for the oscilloscope.
Error output	ERROR	Error output (open drain). The TMC603 has three different sources for signaling an error: - Undervoltage of the low side supply - Undervoltage of the charge pump - Short to GND detection Upon any of these events the error output is pulled low.
Current output	CUR1/2/3	Output of current measurement amplifier (for phase 1 to 3). The output signal is centered to 1.65V.
Filter output	F1/2/3	Output of internal switched capacitor filter (for phase 1 to 3). This signal is available only if filter outputs of TMC603 are selected (refer chapter 7).
High side output	HS1/2/3	High side MOSFET driver output (for phase 1 to 3)
Bridge output	BM1/2/3	Sensing input for bridge output. Used for MOSFET control and current measurement. (for phase 1 to 3)
Low side output	LS1/2/3	Low side MOSFET driver output (for phase 1 to 3)

Table 3.12 Measuring points

4 TMCL Overview

As with most TRINAMIC modules the software running on the STM32F ARM Cortex-M3 processor of the TMC603-EVAL 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 – normally – untouched throughout the whole lifetime, the firmware can be updated by the user. The firmware shipped with this evaluation board is related to the standard TMCL firmware shipped with most of TRINAMIC modules with regard to protocol and commands.

There are two software tools: the *TMCM-BLDC* and the *TMCL-IDE*. Whereas the TMCM-BLDC is used for testing different configurations in all modes of operation the TMCL-IDE is mainly designed for conceiving programs and firmware updates. New versions of the TMCM-BLDC and the TMCL-IDE can be downloaded free of charge from the TRINAMIC website (http://www.trinamic.com).

4.1 TMCM-BLDC

The TMCM-BLDC is a special small program for adjusting and testing settings of TMCM modules for BLDC motors. This software tool offers dialogues for all modes of operation. The TMCM-BLDC can be downloaded from www.trinamic.com and is compatible with the TMCL-IDE.

The TMCM-BLDC is a PC application running under WindowsXP / Vista / Windows 7 (Windows 3.x is not supported) that includes

- a connection dialogue for connecting the module,
- a dialogue for basic settings (motor settings, encoder settings, commutation mode, trace controller),
- three dialogues for operation modes, each for one mode of operation (torque mode, velocity mode, positioning mode),
- a dialogue for entering and executing TMCL commands in direct mode,
- a file menu for exporting/importing settings, storing or restoring them. Further, settings can be exported to TMCL for use with the TMCL-IDE.

The TMCM-BLDC is designed for finding initial settings, e.g. the values for P and I parameters of a specific mode of operation. Each value can be changed on the fly and the results are shown immediately on the diagrams. After optimum values are found they can be exported to the TMCL-IDE for developing programs that run standalone on the module later on. TRINIAMIC recommends using this TMCM-BLDC tool first and proceed afterwards with the TMCL-IDE.

4.1.1 Getting Started with TMCM-BLDC

The first step is to connect the module by clicking the *Connect* button. (Please refer to the specific hardware and firmware manuals of your module for further information about connecting cables etc. prior to this.) Proceed if the communication between module and PC is established.

THE PROGRAM SURFACE GIVES AN IMPRESSION HOW TO WORK WITH TMCM-BLDC:

The *settings* tab is needed for adjusting general settings of the module. The other three tabs are designed for trying out *torque mode*, *velocity mode*, and *positioning mode*. The last tab is used for controlling the module with *TMCL direct mode*. All TMCL commands can be entered as usual.

On the right side of the window are *status and error flags*. Below the tabs are diagrams for velocity, position, and current. These diagrams and the status/error information can be used for controlling settings visually in order to identify best results as well as deficient settings. It is possible to scale each X-axis and Y-axis to get a comfortable report. Status/error flags and diagrams are for diagnostic tasks only and remain visible, while the program is used. The input field on top can be chosen by selecting a specific tab.

Please note that the status/error information and the charts have to be activated by starting the trace controller on the settings tab, which polls the corresponding values from the board on a regular basis.

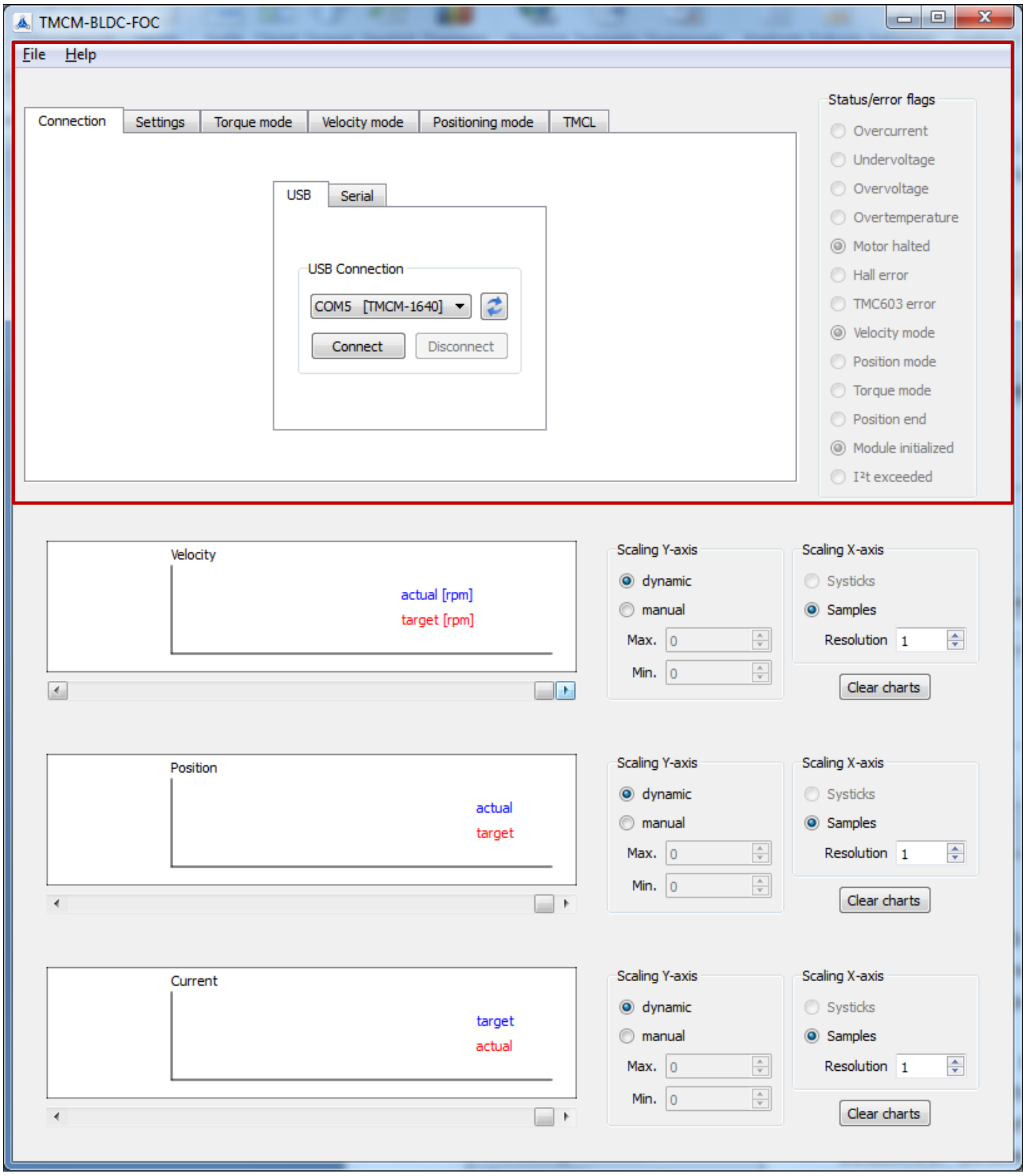


Figure 4.1 Connection tab of TMCM-BLDC

4.1.2 Dialogues of the TMCM-BLDC

4.1.2.1 Settings

After connecting the module with the *connect* button you can choose the *settings tab* and fill in basic values: *motor settings*, *encoder settings* and *commutation mode*. All settings correspond to specific axis parameters of your module.

The *trace controller* has to be started for displaying the curves on the diagrams below. Clicking the *start* button of the trace controller enables the status/error flags, too.

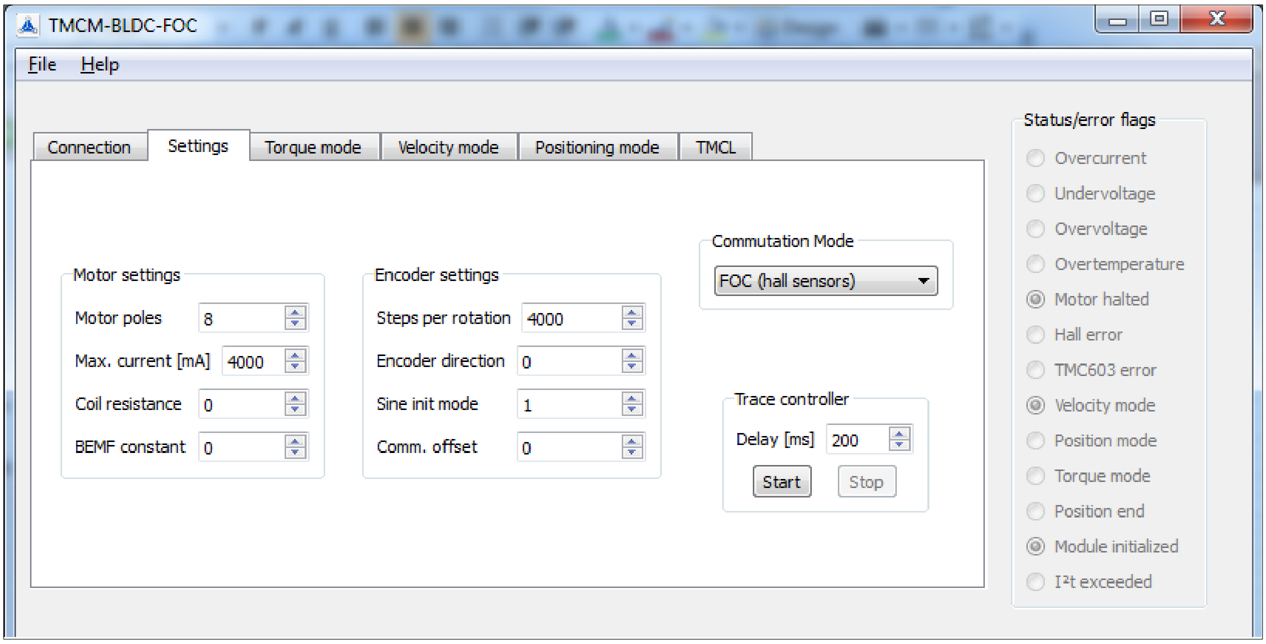


Figure 4.2 Settings tab of TMCM-BLDC

4.1.2.2 Torque Mode

The *torque mode tab* offers the possibility to test different current settings and to evaluate the *current control*. A target current can be chosen. Further, the *current PI control* can be adjusted by choosing values for the P and I parameters.

The drive can be started (in positive and negative direction) and stopped with the buttons in the *current control* field. The values can be calibrated on the fly while the drive is still active. The results will be shown immediately on the diagrams below.

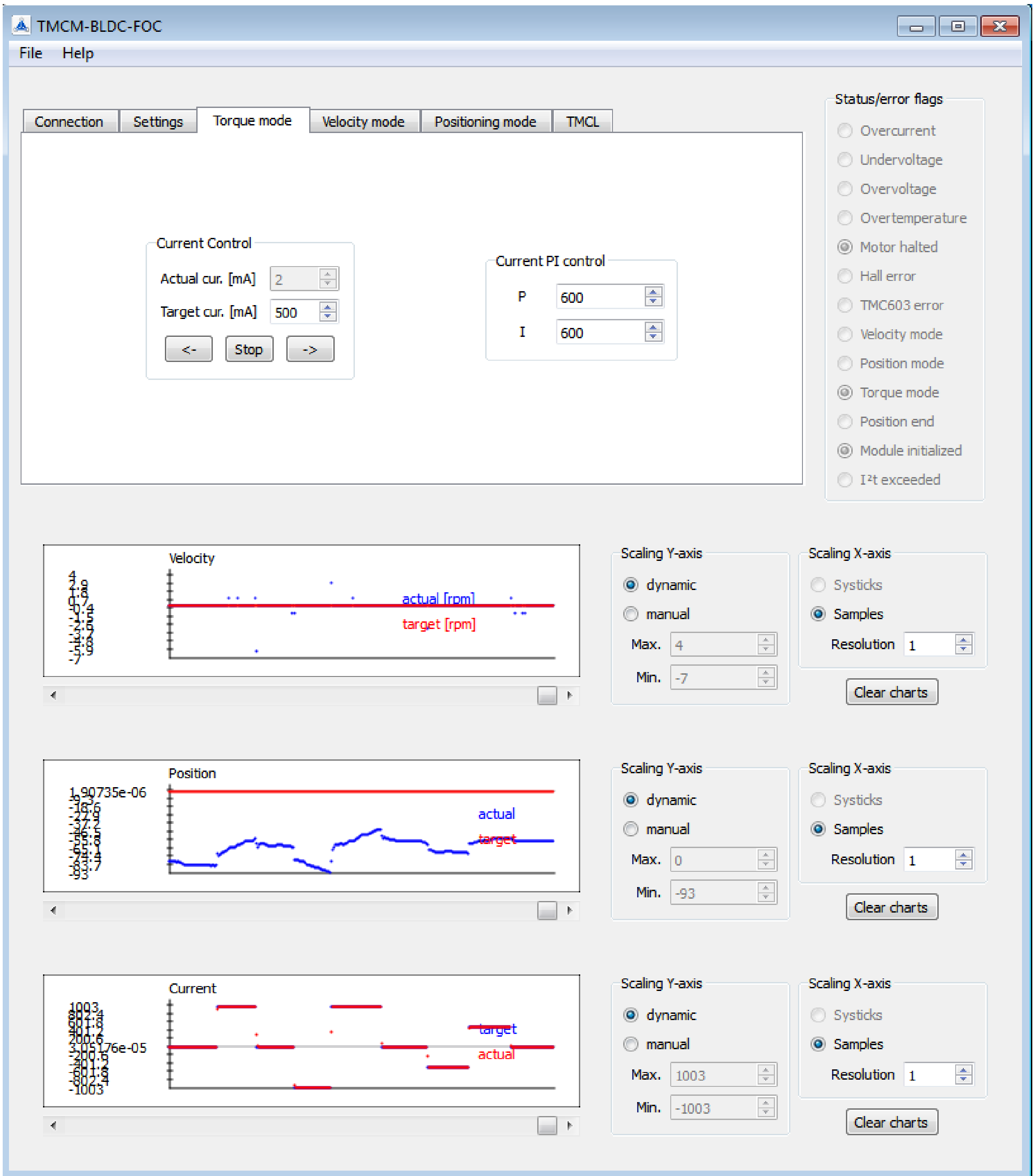


Figure 4.3 Torque mode tab of TMCM-BLDC

4.1.2.3 Velocity Mode

The input area of the *velocity mode tab* has three parts: the *velocity ramp control*, the *velocity control* and the *velocity PI control*. In the middle of the input area is the *velocity control*, which is used to start the drive (in positive and negative direction) in velocity mode with a chosen speed [rpm] or stop it. The *velocity ramp control* is needed for setting the maximum velocity [rpm] and the acceleration [rpm/s]. Further, the velocity ramp can be enabled by ticking the appropriate field. Disabling the velocity ramp leads to a hard stop.

On the right side is the *velocity PI control*. Here, the P and I parameter values can be set. The values can be calibrated on the fly while the drive is still active. The results will be shown immediately on the diagrams.

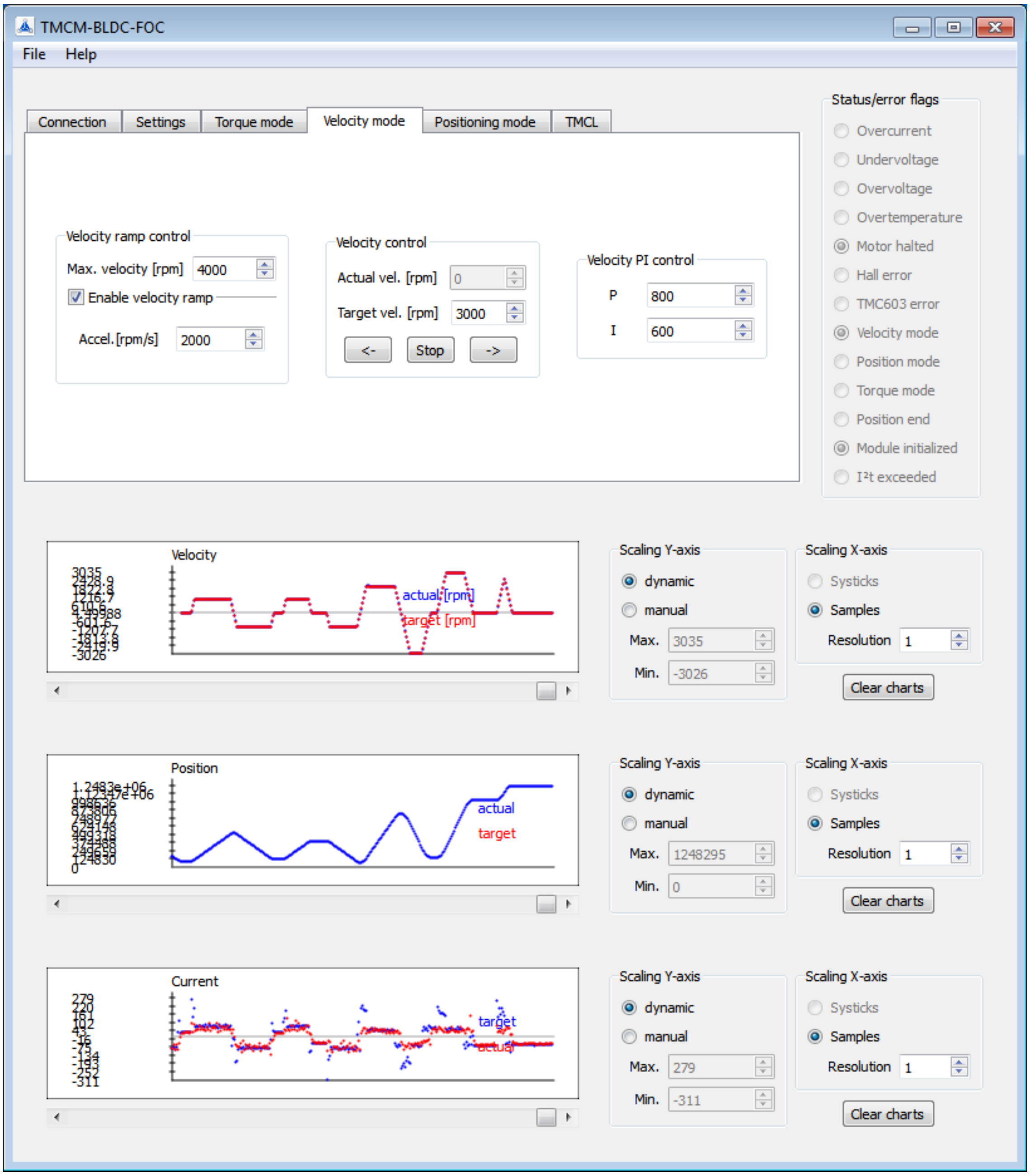


Figure 4.4 Velocity mode tab of TMCM-BLDC

4.1.2.4 Positioning Mode

The input area of the *positioning mode tab* has three parts: the *velocity ramp control*, the *positioning control*, and the *position p control*.

The *velocity ramp control* is the same as on the velocity mode tab. Maximum velocity and acceleration can be chosen and the velocity ramp can be enabled or disabled. In the middle of the positioning mode input area is the *positioning control* field. This is adequate designed to the TMCL command MVP (*move to position*). There are two possibilities to move in positive or negative direction: move absolutely or relatively to the actual position. Units for different commutation modes are as follows:

- The unit of the target position for positioning with encoder is *encoder steps per motor rotation*.
- The unit for positioning with hall sensors is $\frac{6 \times motorpoles}{2}$ *steps per motor rotation*.

The button *clear* sets the counter for positioning to zero. *Clear on NULL* is used with encoder. The actual position is set to zero when crossing the next N channel.

On the right side of the positioning mode tab input area is the *position P control*. Here, the value of the P parameter can be set. The value can be calibrated on the fly while the drive is still active. The results will be shown immediately on the diagrams.

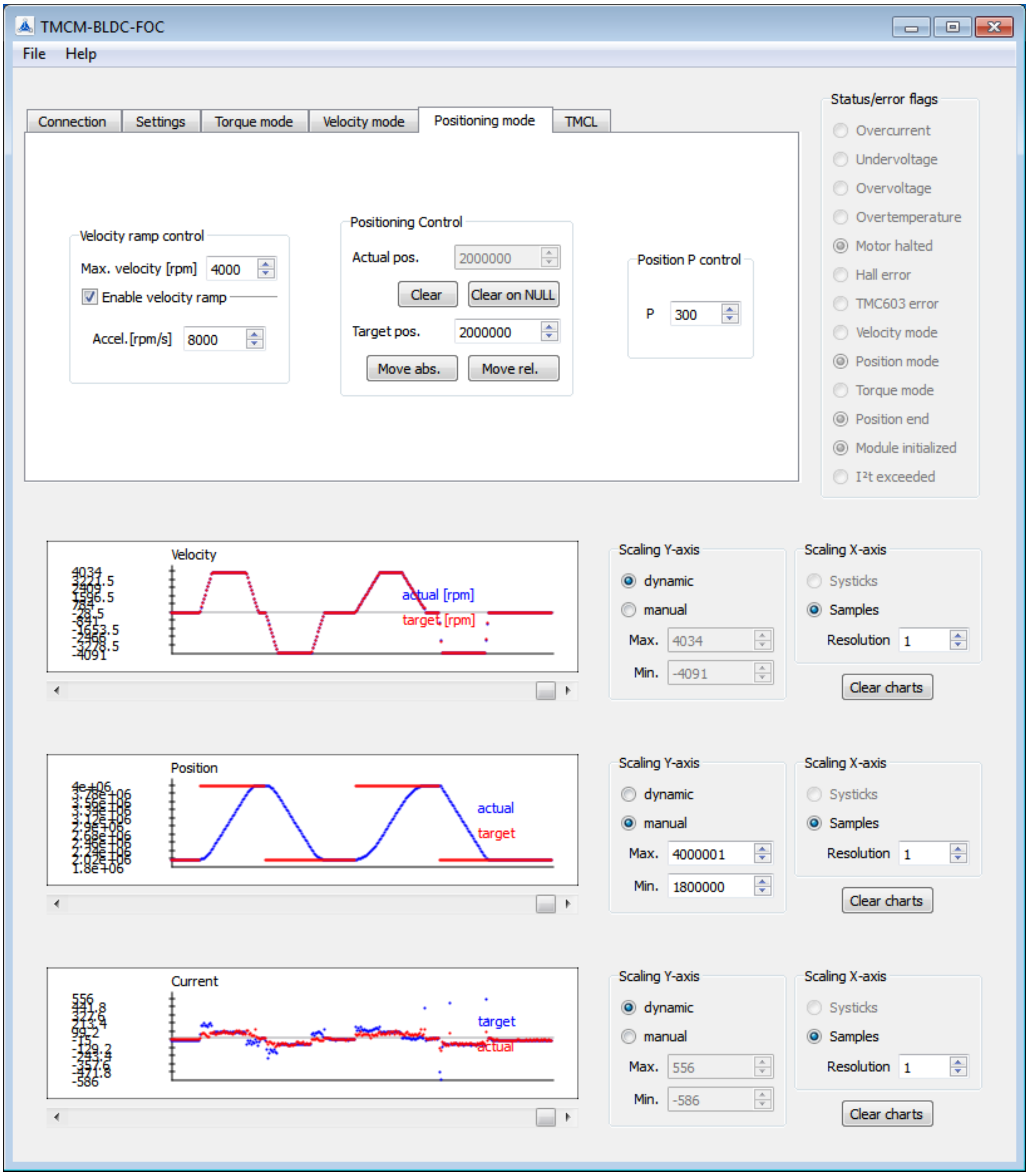


Figure 4.5 Positioning mode tab of TMCM-BLDC

4.1.2.5 TMCL

The input area of the TMCL tab has the same structure as the appropriate window for *TMCL direct mode* of the TMCL-IDE. *Command number*, *type*, *motor/bank* and a chosen *value* can be set. By clicking the *Send* button the request will be sent to the module. Immediately the reply of the module will be displayed in the *Reply* field.

Please refer to the complete lists of axis parameters and global parameters of your module in the appropriate firmware manual, too.

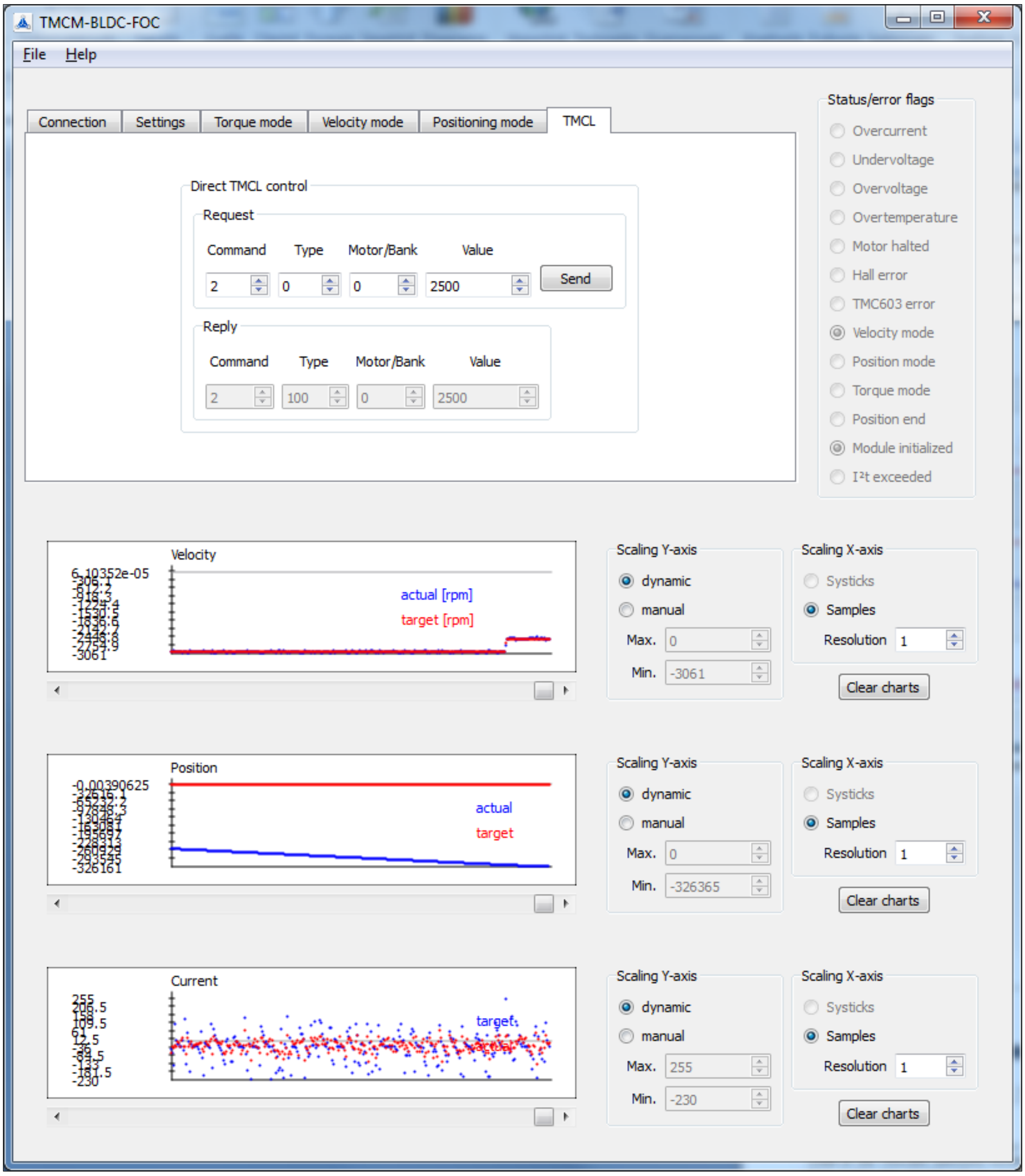


Figure 4.6 TMCL tab of TMCM-BLDC

4.1.3 File Menu of TMCM-BLDC

The file menu of the TMCM-BLDC offers the possibility to import and to export settings. This is useful for transferring settings from one module to another. Settings can be exported (*Export settings to *.ini*) and afterwards imported to another module with the command *Import settings from *.ini*.

Further, it is useful to export evaluated adjustments of the TMCM-BLDC program to a TMCL script used later in the TMCL-IDE. Therefore choose *Export settings to TMCL*.

Certainly actual values can be stored and restored on the module.

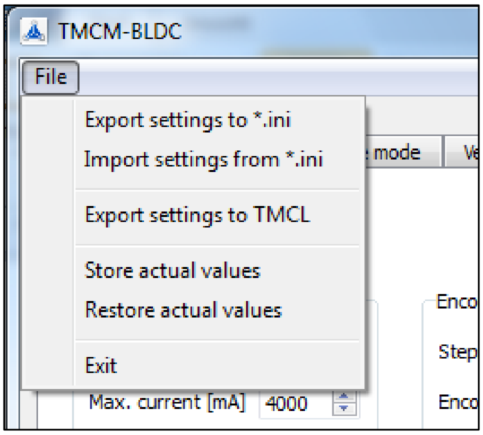


Figure 4.7 File menu of TMCM-BLDC

4.2 TMCL-IDE

The TMCL-IDE is an integrated development environment mainly for developing standalone TMCL applications, but it also includes a function for using the TMCL commands in direct mode. The TMCL-IDE is a PC application running under Windows 95/98/NT/2000/XP/Vista/Windows 7.

For the TMC603-EVAL in particular the dialogue for configuring BLDC modules is important. Most other functions can be used, too. Please refer to the TMCL-IDE User Manual (www.trinamic.com) for detailed information about this.

Please be sure to always use the latest version of the TMCL-IDE as its functionality is being extended and improved constantly.

4.2.1 BLDC Tool of the TMCL-IDE

The BLDC tool of the TMCL-IDE consists mainly of two parts: the left for adjusting axis parameters and testing them directly in praxis and the right for reporting generated values.

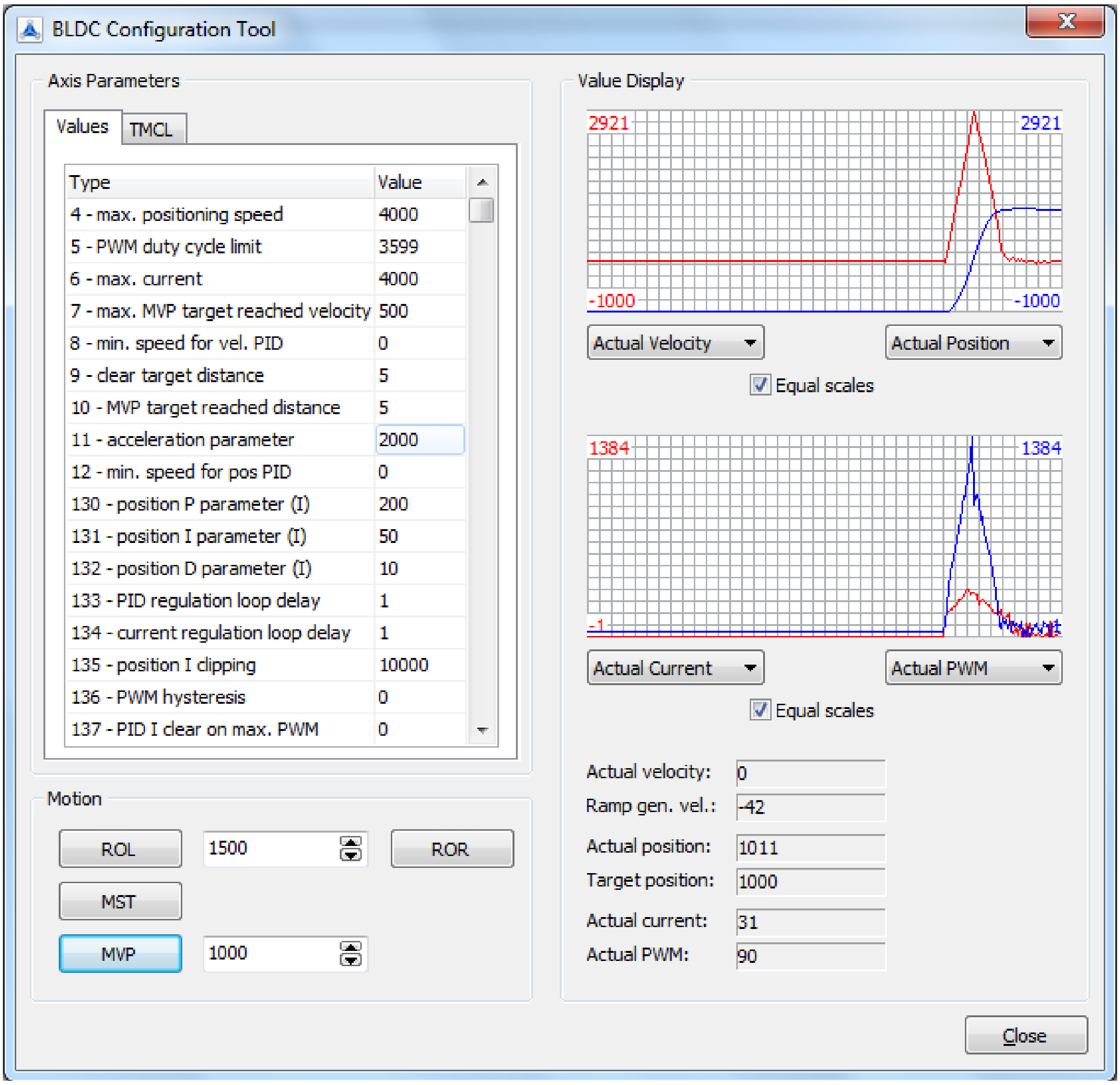


Figure 4.8 Configuration tool of the TMCL-IDE

4.2.1.1 Axis Parameters and Motion

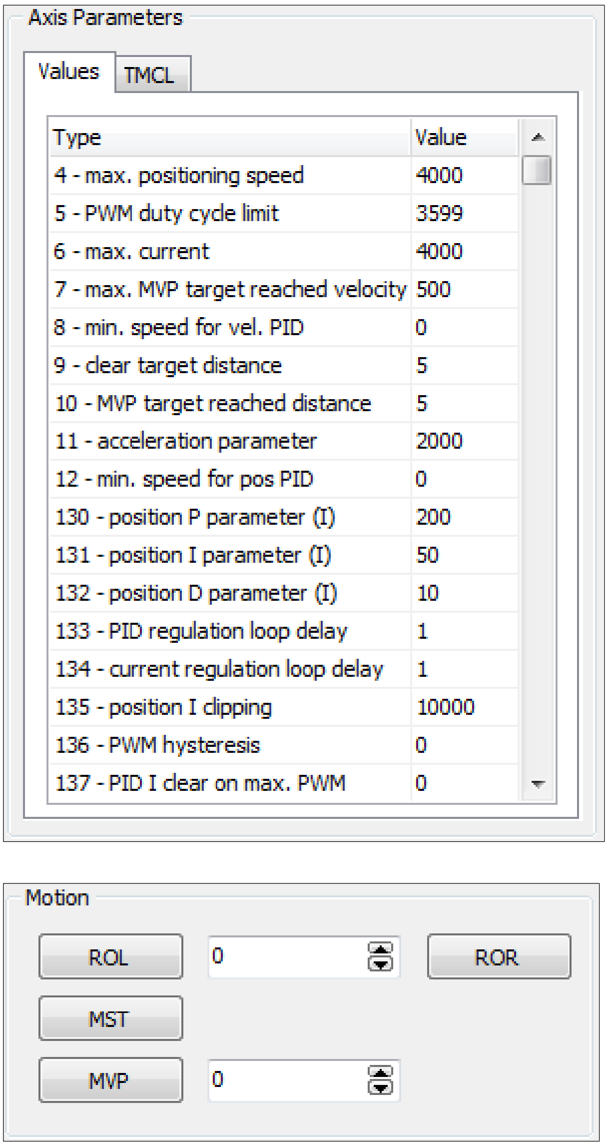


Figure 4.9 Values tab of the TMCL-IDE

On the left side of the BLDC tool are two tabs, which can be chosen:

- *Values*
- *TMCL*

The *values tab* offers a table with two columns: numbers of the TMCL axis parameters and appropriate descriptions on the left and given values on the right. The value of each axis parameter can be changed directly. If the overwritten value field is left the new value becomes valid. The TMCL command behind this action is *set axis parameter* (SAP). Immediately changes will be recognized and shown on the *TMCL tab*, where all SAP commands are displayed as TMCL mnemonics / text strings. The content of the TMCL™ tab can be marked and copied by clicking the *copy* button. Certainly it is possible to paste copied sections into the editor of the TMCL-IDE in order to create one's own application.

Below the two tabs of the *axis parameter* area is an input area for *motion* control. Using its functionality is important for testing the adjusted axis parameters directly in praxis. In the middle of this area are two value input fields: the above is for setting the speed for the commands *rotate left* (ROL) or *rotate right* (ROR). The value input field on the bottom is used for positioning. The TMCL™ command behind this action is termed *move to absolute position* (MVP ABS).

You will find further information about the TMCL commands in chapter 4.4.

4.2.1.2 Value Display

While the TMCL BLDC tool is active the *value display* offers a smart visualization of all chosen adjustments. Changes of axis parameters and other commands will be shown immediately in graphics and table.

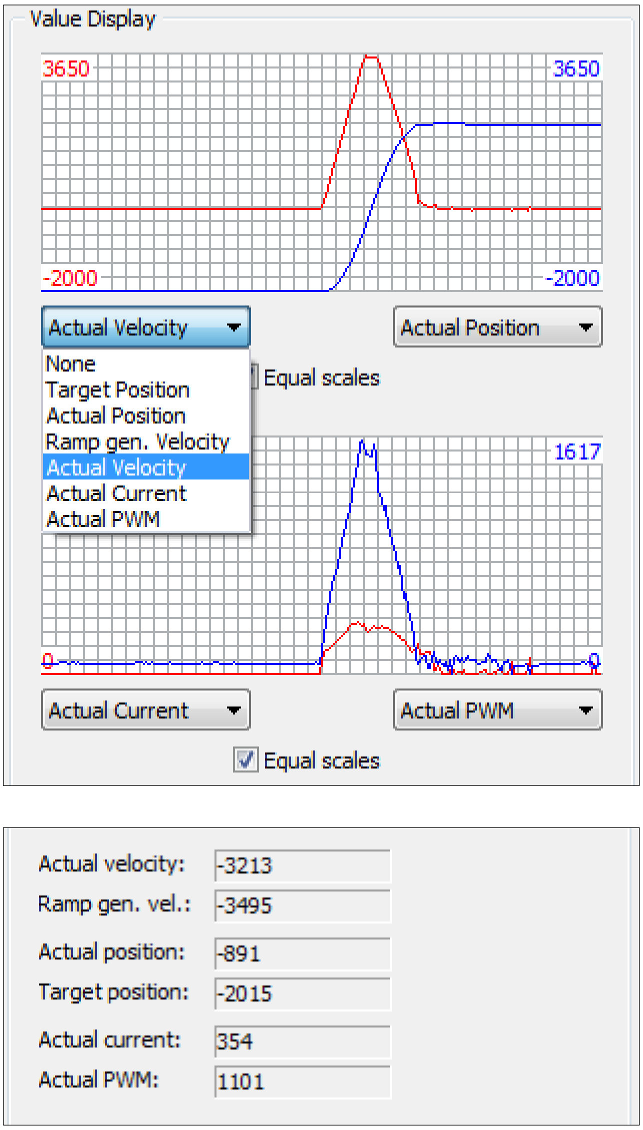


Figure 4.10 Value display of the TMCL-IDE

All parameter changes and commands which have been initiated with the help of the *values tab* and the *motion area* will be shown immediately on the screens and the accordant table of the *value display*. These tools offer read out values used for adjusting positioning operations, mainly. The TMCL™ command behind this is the *get axis parameter* command (GAP).

The *value display* area offers possibilities for changing its adjustments. Under each graphic are two buttons for choosing the aspects which have to be examined.

Select out of the following catalog:

- *None*
- *Target Position*
- *Actual Position*
- *Ramp. gen. Velocity*
- *Actual Velocity*
- *Actual Current*

At all, the curves of up to four aspects can be shown on the two screens at the same time.

Depending on the targets a customer is engaged with it might be helpful to equal the scales by ticking the appropriate fields (*default*).

4.3 TMCL Command Overview

In this section a short overview of the TMCL commands is given.

4.3.1 Motion Commands

These commands control the motion of the motor. They are the most important commands and can be used in direct mode or in stand-alone mode.

Mnemonic	Command number	Description
ROR	1	Rotate right
ROL	2	Rotate left
MST	3	Motor stop
MVP	4	Move to position

Figure 4.11 Motion commands

4.3.2 Parameter Commands

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

Mnemonic	Command number	Description
SAP	5	Set axis parameter
GAP	6	Get axis parameter
STAP	7	Store axis parameter into EEPROM
RSAP	8	Restore axis parameter from EEPROM
SGP	9	Set global parameter
GGP	10	Get global parameter
STGP	11	Store global parameter into EEPROM
RSGP	12	Restore global parameter from EEPROM

Figure 4.12 Parameter commands

4.3.3 I/O Port Commands

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

Mnemonic	Command number	Meaning
SIO	14	Set output
GIO	15	Get input

4.4 Commands

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

4.4.1 ROR (rotate right)

The motor will be instructed to rotate with a specified velocity in *right* direction (increasing the position counter).

Internal function: First, velocity mode is selected. Then, the velocity value is transferred to axis parameter #2 (*target velocity*).

Related commands: ROL, MST, SAP, GAP

Mnemonic: ROR 0, <velocity>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE <velocity>
1	don't care	0	-200000… +200000

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	1	don't care

Example:
Rotate right, velocity = 350
Mnemonic: ROR 0, 350

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$01	$00	$00	$00	$00	$01	$5e

4.4.2 ROL (rotate left)

The motor will be instructed to rotate with a specified velocity (opposite direction compared to ROR, decreasing the position counter).

Internal function: First, velocity mode is selected. Then, the velocity value is transferred to axis parameter #2 (*target velocity*).

Related commands: ROR, MST, SAP, GAP

Mnemonic: ROL 0, <velocity>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE <velocity>
2	don't care	0	-200000… +200000

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	2	don't care

Example:

Rotate left, velocity = 1200
Mnemonic: ROL 0, 1200

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$02	$00	$00	$00	$00	$04	$b0

4.4.3 MST (motor stop)

The motor will be instructed to stop.

Internal function: The axis parameter *target velocity* is set to zero.

Related commands: ROL, ROR, SAP, GAP

Mnemonic: MST 0

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
3	don't care	0	don't care

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	3	don't care

Example:

Stop motor
Mnemonic: MST 0

Binary:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$03	$00	$00	$00	$00	$00	$00

4.4.4 MVP (move to position)

The motor will be instructed to move to a specified relative or absolute position. It uses the acceleration/deceleration ramp and the positioning speed programmed into the unit. This command is non-blocking (like all commands). A reply will be sent immediately after command interpretation. Further commands may follow without waiting for the motor reaching its end position. The maximum velocity and acceleration are defined by axis parameters #4 and #11.

TWO OPERATION TYPES ARE AVAILABLE:

- Moving to an absolute position in the range from -2147483648… +2147483647.
- Starting a relative movement by means of an offset to the actual position. In this case, the new resulting position value must not exceed the above mentioned limits, too.

Internal function: A new position value is transferred to the axis parameter #0 *target position*.

Related commands: SAP, GAP, and MST

Mnemonic: MVP <ABS|REL>, 0, <position|offset value>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
4	0 ABS – absolute	0	<position> -2147483648… +2147483647
	1 REL – relative	0	<offset> -2147483648… +2147483647

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	4	don't care

Example MVP ABS:

Move motor to (absolute) position 9000
Mnemonic: MVP ABS, 0, 9000

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$04	$00	$00	$00	$00	$23	$28

Example MVP REL:

Move motor from current position 1000 steps backward (move relative -1000)
Mnemonic: MVP REL, 0, -1000

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$00	$04	$01	$00	$ff	$ff	$fc	$18

4.4.5 SAP (set axis parameter)

Most of the motion control parameters of the module can be specified by using the SAP command. The settings will be stored in SRAM and therefore are volatile. Thus, information will be lost after power off. ***Please use command STAP (store axis parameter) in order to store any setting permanently.***

Related commands: GAP, STAP, and RSAP

Mnemonic: SAP <parameter number>, 0, <value>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
5	<parameter number>	0	<value>

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	5	don't care

A list of all parameters which can be used for the SAP command is shown in section 5.

Example:

Set the absolute maximum current to 2000mA
Mnemonic: SAP 6, 0, 2000

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$05	$06	$00	$00	$00	$07	$D0

4.4.6 GAP (get axis parameter)

Most parameters of the TMC603-EVAL can be adjusted individually. They can be read out using the GAP command.

Related commands: SAP, STAP, and RSAP

Mnemonic: GAP <parameter number>, 0

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
6	<parameter number>	0	don't care

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	6	don't care

A list of all parameters which can be used for the GAP command is shown in section 5.

Example:

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

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$06	$01	$00	$00	$00	$00	$00

Reply:

Byte Index	0	1	2	3	4	5	6	7
Function	Host-address	Target-address	Status	Instructio n	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$00	$01	$64	$06	$00	$00	$02	$c7

4.4.7 STAP (store axis parameter)

The STAP command stores an axis parameter previously set with a *Set Axis Parameter command (SAP)* permanently. Most parameters are automatically restored after power up.

Internal function: An axis parameter value stored in SRAM will be transferred to EEPROM and loaded from EEPORM after next power up.

Related commands: SAP, RSAP, and GAP

Mnemonic: STAP <parameter number>, 0

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
7	<parameter number>	0	don't care*

*** The value operand of this function has no effect. Instead, the currently used value (e.g. selected by SAP) is saved.**

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	7	don't care

A list of all parameters which can be used for the STAP command is shown in section 5.

Example:

Store the maximum speed
Mnemonic: STAP 4, 0

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$07	$04	$00	$00	$00	$00	$00

Note: The STAP command will not have any effect when the configuration EEPROM is locked. The error code 5 (configuration EEPROM locked) will be returned in this case.

4.4.8 RSAP (restore axis parameter)

For all configuration related axis parameters non-volatile memory locations are provided. By default, most parameters are automatically restored after power up. A single parameter that has been changed before can be reset by this instruction also.

Internal function: The specified parameter is copied from the configuration EEPROM memory to its RAM location.

Related commands: SAP, STAP, and GAP

Mnemonic: RSAP <parameter number>, 0

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
8	<parameter number>	0	don't care

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	8	don't care

A list of all parameters which can be used for the RSAP command is shown in section 5.

Example:

Restore the maximum current
Mnemonic: RSAP 6, 0

Binary:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$08	$06	$00	$00	$00	$00	$00

4.4.9 SGP (set global parameter)

Global parameters are related to the host interface, peripherals or other application specific variables. The different groups of these parameters are organized in *banks* to allow a larger total number for future products. Currently, bank 0 is used for global parameters and bank 2 is intended for user variables.

Related commands: GGP, STGP, RSGP

Mnemonic: SGP <parameter number>, <bank number>, <value>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
9	<parameter number>	<bank number>	<value>

Reply in direct mode:

STATUS	VALUE
100 – OK	don't care

A list of all parameters which can be used for the SGP command is shown in section **Fehler! Verweisquelle konnte nicht gefunden werden.**.

Example:

Set variable 0 at bank 2 to 100
Mnemonic: SGP, 0, 2, 100

Binary:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$09	$00	$02	$00	$00	$00	$64

4.4.10 GGP (get global parameter)

All global parameters can be read with this function.

Related commands: SGP, STGP, RSGP

Mnemonic: GGP <parameter number>, <bank number>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
10	<parameter number>	<bank number>	don't care

Reply in direct mode:

STATUS	VALUE
100 – OK	<value>

A list of all parameters which can be used for the GGP command is shown in section 6.

Example:

Get variable 0 from bank 2
Mnemonic: GGP, 0, 2

Binary:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$0a	$00	$02	$00	$00	$00	$00

4.4.11 STGP (store global parameter)

Some global parameters are located in RAM memory, so modifications are lost at power down. This instruction copies a value from its RAM location to the configuration EEPROM and enables permanent storing. Most parameters are automatically restored after power up.

Related commands: SGP, GGP, RSGP

Mnemonic: STGP <parameter number>, <bank number>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
11	<parameter number>	<bank number>	don't care

Reply in direct mode:

STATUS	VALUE
100 – OK	don't care

A list of all parameters which can be used for the GGP command is shown in section 6.

Example:

Copy variable 0 at bank 2 to the configuration EEPROM
Mnemonic: STGP, 0, 2

Binary:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$0b	$00	$02	$00	$00	$00	$00

4.4.12 RSGP (restore global parameter)

This instruction copies a value from the configuration EEPROM to its RAM location and so recovers the permanently stored value of a RAM-located parameter. Most parameters are automatically restored after power up.

Related commands: SGP, GGP, STGP

Mnemonic: RSGP <parameter number>, <bank number>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
12	<parameter number>	<bank number>	don't care

Reply in direct mode:

STATUS	VALUE
100 – OK	don't care

A list of all parameters which can be used for the GGP command is shown in section 6.

Example:

Copy variable 0 at bank 2 from the configuration EEPROM to the RAM location
Mnemonic: RSGP, 0, 2

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$0c	$00	$02	$00	$00	$00	$00

4.4.13 SIO (set output) and GIO (get input / output)

The TMC603-EVAL provides two commands for dealing with inputs and outputs:

- ***SIO*** sets the status of the general digital output either to low (0) or to high (1).
- With ***GIO*** the status of the two available general purpose inputs of the module can be read out. The command reads out a digital or analogue input port. Digital lines will read 0 and 1, while the ADC channel delivers 12 bit in the range of 0… 4095.

CORRELATION BETWEEN I/OS AND BANKS

Inputs/ Outputs	Bank	Description
Digital inputs	Bank 0	Digital inputs are accessed in bank 0.
Analogue inputs	Bank 1	Analog inputs are accessed in bank 1.
Digital outputs	Bank 2	The states of the OUT lines (that have been set by SIO commands) can be read back using bank 2.

4.4.13.1 SIO (set output)

Bank 2 is used for setting the status of the general digital output either to low (0) or to high (1).

Internal function: the passed value is transferred to the specified output line.

Related commands: GIO, WAIT

Mnemonic: SIO <port number>, <bank number>, <value>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
14	<port number>	<bank number> 2	<value> 0/1

Reply structure:

STATUS	VALUE
100 – OK	don't care

Binary:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$0e	$07	$02	$00	$00	$00	$01

4.4.13.2 GIO (get input/output)

GIO can be used in direct mode or in standalone mode.

GIO IN STANDALONE MODE

In standalone mode the requested value is copied to the accumulator (accu) for further processing purposes such as conditioned jumps.

GIO IN DIRECT MODE

In direct mode the value is 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 specified line is read.

Related commands: SIO, WAIT

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

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
15	<port number>	<bank number>	don't care

Reply in direct mode:

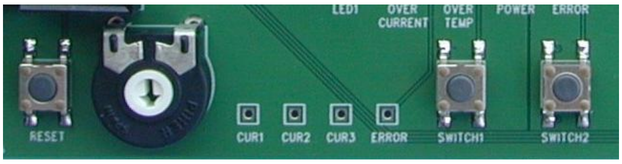
STATUS	VALUE
100 – OK	<status of the port>

Binary:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$0f	$00	$01	$00	$00	$00	$00

Reply:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Host-address	Target-address	Status	Instructio n	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$02	$01	$64	$0f	$00	$00	$01	$2e



Potentiometer Switch 1 Switch 2

Figure 4.13 Potentiometer and switches

PROVIDED SIO AND GIO COMMANDS

I/O	Digital	Analog	GIO <port>, <bank>	SIO <port>, <bank>, <value>	Value range
Potentiometer	-	x	GIO 0, 1	-	0… 4095
Switch 1	x	-	GIO 0, 0	-	0/1
Switch 2	x	-	GIO 1, 0	-	0/1
LED out	x	-	GIO 0, 2	SIO, 0, 2, <n>	0/1

4.4.14 Customer Specific TMCL Command Extension

The user definable functions UF0… UF7 are predefined functions without topic for user specific purposes. A user function UF command uses three parameters.

Internal function: Call user specific functions implemented in *C* by TRINAMIC.

Related commands: none

Mnemonic: UF0… UF7 <parameter number>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
64… 71	user defined	user defined	user defined

Reply in direct mode:

Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Target-address	Status	Instruction	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$02	$01	user defined	64… 71	user defined	user defined	user defined	user defined

4.4.15 TMCL Control Functions

There are several TMCL control functions, but for the user is only command 136 interesting. Other control functions can be used with axis parameters.

Command	Type	Parameter	Description	Access
136	0 – string 1 – binary	Firmware version	Get the module type and firmware revision as a string or in binary format. (*Motor/Bank* and *Value* are ignored.)	read

TYPE SET TO 0 - REPLY AS A STRING:

Byte index	Contents
1	Host Address
2… 9	Version string (8 characters, e.g. 603V2.02)

There is no checksum in this reply format!

TYPE SET TO 1 - VERSION NUMBER IN BINARY FORMAT:

The version number is output in the *value* field.

Byte index in value field	Contents
1	Version number, low byte
2	Version number, high byte
3	Type number, low byte
4	Type number, high byte

5 Axis Parameter Overview (SAP, GAP, STAP, RSAP)

The following section describes all axis parameters that can be used with the SAP, GAP, STAP and RSAP commands.

MEANING OF THE LETTERS IN COLUMN ACCESS:

Access type	Related command(s)	Description
R	GAP	Parameter readable
W	SAP	Parameter writable
E	STAP, RSAP	Parameter automatically restored from EEPROM after reset or power-on. These parameters can be stored permanently in EEPROM using STAP command and also explicitly restored (copied back from EEPROM into RAM) using RSAP.

Number	Axis Parameter	Description	Range [Unit]	Access
0	Target position	The target position of a currently executed ramp.	-2147483648… +2147483647	RW
1	Actual position	Set/get the position counter without moving the motor.	-2147483648… +2147483647	RW
2	Target speed	Set/get the desired target velocity.	-200000… +200000 [rpm]	RW
3	Actual speed	The actual velocity of the motor.	-2147483648… +2147483647 [rpm]	R
4	Max. absolute ramp velocity	The maximum velocity used for velocity ramp in velocity mode and positioning mode. Set this value to a realistic velocity which the motor can reach!	0… +200000 [rpm]	RWE
6	Max current	Set/get the max allowed motor current. *This value can be temporarily exceeded marginal due to the operation of the current regulator.	0… +20000 [mA]	RWE
7	MVP Target reached velocity	Maximum velocity at which end position flag can be set. Prevents issuing of end position when the target is passed at high velocity.	0… +200000 [rpm]	RWE
9	Motor halted velocity	If the actual speed is below this value the motor halted flag will be set.	0 +200000 [rpm]	RWE
10	MVP target reached distance	Maximum distance at which the position end flag is set.	0… +100000	RWE
11	Acceleration	Acceleration parameter for ROL, ROR, and the velocity ramp of MVP.	0… +100000 [RPM/s]	RWE
13	Ramp generator speed	The actual speed of the velocity ramp used for positioning and velocity mode.	-2147483648… +2147483647 [rpm]	R
25	Thermal winding time constant	Thermal winding time constant for the used motor. Used for I^2t monitoring.	0… +4294967295 [ms]	RWE
26	I^2t limit	An actual I^2t sum that exceeds this limit leads to increasing the I^2t exceed counter.	0… +4294967295	RWE
27	I^2t sum	Actual sum of the I^2t monitor.	0… +4294967295	R
28	I^2t exceed counter	Counts how often an I^2t sum was higher than the I^2t limit.	0… +4294967295	RWE
29	Clear I^2t exceeded flag	Clear the flag that indicates that the I^2t sum has exceeded the I^2t limit.	(ignored)	W
30	Minute counter	Counts the module operational time in minutes.	0… +4294967295 [min]	RWE

Number	Axis Parameter	Description	Range [Unit]	Access
31	BLDC re-initialization	1: restart the timer and initialize encoder.	(ignored)	W
133	PID regulation loop delay	Delay of the position and velocity regulator	0… +10 [ms]	RWE
134	Current regulation loop delay	Delay of the current regulator.	0… +10 [50µs]	RWE
146	Activate ramp	1: Activate velocity ramp generator for position and velocity mode. (Allows usage of acceleration and positioning velocity for MVP command.)	0/1	RWE
150	Actual motor current	Get actual motor current.	-2147483648… +2147483647 [mA]	R
151	Actual voltage	Actual supply voltage.	0… +4294967295	R
152	Actual driver temperature	Actual temperature of the motor driver.	0… +4294967295	R
155	Target current	Get desired target current or set target current to activate current regulation mode. (+= turn motor in right direction; -= turn motor in left direction)	-20000… +20000 [mA]	RW
156	Error/Status flags	Bit 0: Overcurrent flag. This flag is set if the max. current limit is exceeded. Bit 1: Undervoltage flag. This flag is set if supply voltage is too low for motor operation. Bit 2: Overvoltage flag. This flag is set if the motor becomes switched off due to overvoltage. Bit 3: Overtemperature flag. This flag is set if overtemperature limit is exceeded. Bit 4: Motor halted flag. This flag is set if the velocity does not reach the value set with GAP/SAP 9. Bit 5: Hall error flag. This flag is set upon a hall error. Bit 6: TMC603 error flag Bit 7: unused Bit 8: unused Bit 9: Velocity mode active flag Bit 10: Position mode active flag. Bit 11: Torque mode active flag. Bit 12: unused Bit 13: unused Bit 14: Position end flag. This flag is set if the motor has been stopped at the target position. Bit 15: unused Bit 16: unused Bit 17: I^2t exceeded flag. This flag is set if the I^2t sum exceeded the I^2t limit of the motor. (reset by SAP 29 after the time specified by the I^2t thermal winding time constant) *Flag 0 to 15 are automatically reset. Only flag 17 must be cleared manually.*	0…+4294967295	R
159	Commutation mode	6: FOC based on hall sensor 7: FOC based on encoder 8: FOC controlled (only velocity mode)	6, 7, 8	RWE
161	Encoder set NULL	1: set position counter to zero at next N channel event.	0/1	RWE
162	Switch set NULL	1: set position counter to zero at next switch event.	0/1	RWE

<table>
<tr><th>Number</th><th>Axis Parameter</th><th colspan="3">Description</th><th>Range [Unit]</th><th>Access</th></tr>
<tr><td>163</td><td>Encoder clear set NULL</td><td colspan="3">1: set position counter to zero only once
0: always at an N channel event</td><td>0/1</td><td>RWE</td></tr>
<tr><td rowspan="3">164</td><td rowspan="3">Activate stop switch</td><td>Bit 0</td><td>Left stop switch enable</td><td>When this bit is set the motor will be stopped if it is moving in negative direction and the left stop switch input becomes active</td><td rowspan="3">0… 3</td><td rowspan="3">RWE</td></tr>
<tr><td>Bit 1</td><td>Right stop switch enable</td><td>When this bit is set the motor will be stopped if it is moving in positive direction and the right stop switch input becomes active</td></tr>
<tr><td colspan="3">Please see parameter 166 for selecting the stop switch input polarity.</td></tr>
<tr><td>165</td><td>Actual encoder commutation offset</td><td colspan="3">This value represents the internal commutation offset.
(0 … max. encoder steps per rotation – 1)</td><td>0… 65535</td><td>RWE</td></tr>
<tr><td rowspan="2">166</td><td rowspan="2">Stop switch polarity</td><td>Bit 0</td><td>Left stop switch polarity</td><td>Bit set: Left stop switch input is high active
Bit clear: Left stop switch input is low active</td><td rowspan="2">0… 3</td><td rowspan="2">RWE</td></tr>
<tr><td>Bit 1</td><td>Right stop switch polarity</td><td>Bit set: Right stop switch input is high active
Bit clear: Right stop switch input is low active</td></tr>
<tr><td>172</td><td>P parameter for current PID</td><td colspan="3">P parameter of current PID regulator.</td><td>0… 65535</td><td>RWE</td></tr>
<tr><td>173</td><td>I parameter for current PID</td><td colspan="3">I parameter of current PID regulator.</td><td>0… 65535</td><td>RWE</td></tr>
<tr><td>177</td><td>Start current</td><td colspan="3">Motor current for controlled commutation.</td><td>0… +20000 [mA]</td><td>RWE</td></tr>
<tr><td>200</td><td>Current PID error</td><td colspan="3">Actual error of current PID regulator</td><td>-2147483648… +2147483647</td><td>R</td></tr>
<tr><td>201</td><td>Current PID error sum</td><td colspan="3">Error sum of current PID regulator</td><td>-2147483648… +2147483647</td><td>R</td></tr>
<tr><td>210</td><td>Actual hall angle</td><td colspan="3">Actual hall angle value</td><td>-32767… +32767</td><td>R</td></tr>
<tr><td>211</td><td>Actual encoder angle</td><td colspan="3">Actual encoder angle value</td><td>-32767… +32767</td><td>R</td></tr>
<tr><td>212</td><td>Actual controlled angle</td><td colspan="3">Actual controlled angle value</td><td>-32767… +32767</td><td>R</td></tr>
<tr><td>226</td><td>Position PID error</td><td colspan="3">Actual error of position PID regulator</td><td>-2147483648… +2147483647</td><td>R</td></tr>
<tr><td>228</td><td>Velocity PID error</td><td colspan="3">Actual error of velocity PID regulator</td><td>-2147483648… +2147483647</td><td>R</td></tr>
<tr><td>229</td><td>Velocity PID error sum</td><td colspan="3">Sum of errors of velocity PID regulator</td><td>-2147483648… +2147483647</td><td>R</td></tr>
<tr><td>230</td><td>P parameter for position PID</td><td colspan="3">P parameter of position PID regulator.</td><td>0… 65535</td><td>RWE</td></tr>
</table>

Number	Axis Parameter	Description	Range [Unit]	Access
234	P parameter for velocity PID	P parameter of velocity PID regulator.	0… 65535	RWE
235	I parameter for velocity PID	I parameter of velocity PID regulator.	0… 65535	RWE
238	Mass inertia constant	Mass inertia constant. Compensates the rotor inertia of the motor.	0… 65535	RWE
239	BEMF constant	BEMF constant of the motor. Used for current, position, and velocity regulation.	0… 65535 [rpm/(10V)]	RWE
240	Motor coil resistance	Resistance of motor coil. Used for current, position, and velocity regulation.	0… 65535 [mΩ]	RWE
241	Sine initialization speed	Velocity during initialization in init sine mode 2. Refer to axis parameter 249, too.	-200000… +200000 [rpm]	RWE
244	Init sine delay	Duration for sine initialization sequence. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	0… 10000 [ms]	RWE
245	Overvoltage protection	1: Enable overvoltage protection.	0/1	RWE
249	Init sine mode	0: Initialization in controlled sine commutation (determines the encoder offset) 1: Initialization in block commutation using hall sensors 2: Initialization in controlled sine commutation (use the previous set encoder offset)	0, 1, 2	RWE
250	Encoder steps	Encoder steps per rotation.	0… +65535	RWE
251	Encoder direction	Set the encoder direction in a way, that ROR increases position counter.	0/1	RWE
253	Number of motor poles	Number of motor poles.	+2… +254	RWE
254	Hall sensor invert	1: Invert the hall scheme	0/1	RWE

5.1 Axis Parameters Sorted by Functionality

The following section describes all axis parameters that can be used with the SAP, GAP, STAP, RSAP and AAP commands.

MEANING OF THE LETTERS IN COLUMN *ACCESS*:

Access type	Related command(s)	Description
R	GAP	Parameter readable
W	SAP, AAP	Parameter writable
E	STAP, RSAP	Parameter automatically restored from EEPROM after reset or power-on. These parameters can be stored permanently in EEPROM using STAP command and also explicitly restored (copied back from EEPROM into RAM) using RSAP.

MOTOR / MODULE SETTINGS

Number	Axis Parameter	Description	Range [Unit]	Access
253	Number of motor poles	Number of motor poles.	+2… +254	RWE
239	BEMF constant	BEMF constant of the motor. Used for current, position, and velocity regulation.	0… 65535 [rpm/(10V)]	RWE
240	Motor coil resistance	Resistance of motor coil. Used for current, position, and velocity regulation.	0… 65535 [mΩ]	RWE
238	Mass inertia constant	Mass inertia constant. Compensates the rotor inertia of the motor.	0… 65535	RWE
25	Thermal winding time constant	Thermal winding time constant for the used motor. Used for I²t monitoring.	0… +4294967295 [ms]	RWE
26	I²t limit	An actual I²t sum that exceeds this limit leads to increasing the I²t exceed counter.	0… +4294967295	RWE
27	I²t sum	Actual sum of the I²t monitor.	0… +4294967295	R
28	I²t exceed counter	Counts how often an I²t sum was higher than the I²t limit.	0… +4294967295	RWE
29	Clear I²t exceeded flag	Clear the flag that indicates that the I²t sum has exceeded the I²t limit.	(ignored)	W
30	Minute counter	Counts the module operational time in minutes.	0… +4294967295 [min]	RWE
245	Overvoltage protection	1: Enable overvoltage protection.	0/1	RWE

ENCODER / INITIALIZATION SETTINGS

Number	Axis Parameter	Description	Range [Unit]	Access
31	BLDC re-initialization	1: restart the timer and initialize encoder.	(Ignored)	W
159	Commutation mode	6: FOC based on hall sensor 7: FOC based on encoder 8: FOC controlled (only velocity mode)	6, 7, 8	RWE
165	Actual encoder commutation offset	This value represents the internal commutation offset. (0 … max. encoder steps per rotation)	0… 65535	RWE
177	Start current	Motor current for controlled commutation. This parameter is used in commutation mode.	0… +20000 [mA]	RWE
210	Actual hall angle	Actual hall angle value	-32767… +32767	R
211	Actual encoder angle	Actual encoder angle value	-32767… +32767	R

Number	Axis Parameter	Description	Range [Unit]	Access
212	Actual controlled angle	Actual controlled angle value	-32767… +32767	R
241	Sine initialization speed	Velocity during initialization in init sine mode 2. Refer to axis parameter 249, too.	-200000… +200000 [rpm]	RWE
244	Init sine delay	Duration for sine initialization sequence. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	0… 10000 [ms]	RWE
249	Init sine mode	0: Initialization in controlled sine commutation (determines the encoder offset) 1: Initialization in block commutation using hall sensors 2: Initialization in controlled sine commutation (use the previous set encoder offset)	0… 2	RWE
250	Encoder steps	Encoder steps per rotation.	0… +65535	RWE
251	Encoder direction	Set the encoder direction in a way, that ROR increases position counter.	0/1	RWE
254	Hall sensor invert	1: Hall sensor invert. Invert the hall scheme, e.g. used by some Maxon motors.	0/1	RWE

Torque Regulation Mode

Number	Axis Parameter	Description	Range [Unit]	Access
6	Max current	Set/get the max allowed motor current. This value can be temporarily exceeded marginal due to the operation of the current regulator.	0… +20000 [mA]	RWE
150	Actual motor current	Get actual motor current.	-2147483648… +2147483647 [mA]	R
155	Target current	Get desired target current or set target current to activate current regulation mode. (+= turn motor in right direction; -= turn motor in left direction)	-20000… +20000 [mA]	RW
134	Current regulation loop delay	Delay of the PID current regulator.	0… +10 [50µs]	RWE
172	P parameter for current PID	P parameter of current PID regulator.	0… 65535	RWE
173	I parameter for current PID	I parameter of current PID regulator.	0… 65535	RWE
200	Current PID error	Actual error of current PID regulator	-2147483648… +2147483647	R
201	Current PID error sum	Sum of errors of current PID regulator	-2147483648… +2147483647	R

Velocity Regulation Mode

Number	Axis Parameter	Description	Range [Unit]	Access
2	Target speed	Set/get the desired target velocity.	-2147483648… +2147483647 [rpm]	RW
3	Actual speed	The actual velocity of the motor.	-2147483648… +2147483647 [rpm]	R
9	Motor halted velocity	If the actual speed is below this value the motor halted flag will be set.	0 +200000 [rpm]	RWE

Number	Axis Parameter	Description	Range [Unit]	Access
133	PID regulation loop delay	Delay of the position and velocity	0… +10 [ms]	RWE
234	P parameter for velocity PID	P parameter of velocity PID regulator.	0… +10 [50µs]	RWE
228	Velocity PID error	Actual error of PID velocity regulator	-2147483648… +2147483647	R
229	Velocity PID error sum	Sum of errors of PID velocity regulator	-2147483648… +2147483647	R

Velocity Ramp Parameter

Number	Axis Parameter	Description	Range [Unit]	Access
4	Max. absolute ramp velocity	The maximum velocity used for velocity ramp in velocity mode and positioning mode. Set this value to a realistic velocity which the motor can reach!	0 +200000 [rpm]	RWE
11	Acceleration	Acceleration parameter for ROL, ROR, and the velocity ramp of MVP.	0… +100000 [RPM/s]	RWE
13	Ramp generator speed	The actual speed of the velocity ramp used for positioning and velocity mode.	-2147483648… +2147483647 [rpm]	R
146	Activate ramp	1: Activate velocity ramp generator for position PID control. (Allows usage of acceleration and positioning velocity for MVP command.)	0/1	RWE

Position Regulation Mode

Number	Axis Parameter	Description	Range [Unit]	Access
1	Actual position	Set/get the position counter without moving the motor.	-2147483648… +2147483647	RW
0	Target position	The target position of a currently executed ramp.	-2147483648… +2147483647	RW
7	MVP Target reached velocity	Maximum velocity at which end position flag can be set. Prevents issuing of end position when the target is passed at high velocity.	0 +200000 [rpm]	RWE
10	MVP target reached distance	Maximum distance at which the position end flag is set.	0… +100000	RWE
161	Encoder set NULL	1: set position counter to zero at next N channel event.	0/1	RWE
162	Switch set NULL	1: set position counter to zero at next switch event.	0/1	RWE
163	Encoder clear set NULL	1: set position counter to zero only once 0: always at an N channel event	0/1	RWEP

<table>
<tr><th>Number</th><th>Axis Parameter</th><th colspan="3">Description</th><th>Range [Unit]</th><th>Access</th></tr>
<tr><td rowspan="3">164</td><td rowspan="3">Activate stop switch</td><td>Bit 0</td><td>Left stop switch enable</td><td>When this bit is set the motor will be stopped if it is moving in negative direction and the left stop switch input becomes active</td><td rowspan="3">0… 3</td><td rowspan="3">RWE</td></tr>
<tr><td>Bit 1</td><td>Right stop switch enable</td><td>When this bit is set the motor will be stopped if it is moving in positive direction and the right stop switch input becomes active</td></tr>
<tr><td colspan="3">Please see parameter 166 for selecting the stop switch input polarity.</td></tr>
<tr><td rowspan="2">166</td><td rowspan="2">Stop switch polarity</td><td>Bit 0</td><td>Left stop switch polarity</td><td>Bit set: Left stop switch input is high active
Bit clear: Left stop switch input is low active</td><td rowspan="2">0… 3</td><td rowspan="2">RWE</td></tr>
<tr><td>Bit 1</td><td>Right stop switch polarity</td><td>Bit set: Right stop switch input is high active
Bit clear: Right stop switch input is low active</td></tr>
<tr><td>230</td><td>P parameter for position PID</td><td colspan="3">P parameter of position PID regulator. (</td><td>0… 65535</td><td>RWE</td></tr>
<tr><td>226</td><td>Position PID error</td><td colspan="3">Actual error of PID position regulator</td><td>-2147483648… +2147483647</td><td>R</td></tr>
</table>

Status Information

Number	Axis Parameter	Description	Range [Unit]	Access
151	Actual voltage	Actual supply voltage.	0… +4294967295	R
152	Actual driver temperature	Actual temperature of the motor driver.	0… +4294967295	R

Number	Axis Parameter	Description	Range [Unit]	Access
156	Error/Status flags	Bit 0: Overcurrent flag. This flag is set if max. current limit is exceeded. Bit 1: Undervoltage flag. This flag is set if supply voltage is too low for motor operation. Bit 2: Overvoltage flag. This flag is set if the motor becomes switched off due to overvoltage. Bit 3: Overtemperature flag. This flag is set if overtemperature limit is exceeded. Bit 4: Motor halted flag. This flag is set if the motor velocity does not reach the value set with SAP 9. Bit 5: Hall error flag. This flag is set upon a hall error. Bit 6: TMC603 error flag Bit 7: unused Bit 8: unused Bit 9: Velocity mode active flag Bit 10: Position mode active flag. Bit 11: Torque mode active flag. Bit 12: unused Bit 13: unused Bit 14: Position end flag. This flag is set if the motor has been stopped at the target position. Bit 15: unused Bit 16: unused Bit 17: I^2t exceeded flag. This flag is set if the I^2t sum exceeded the I^2t limit of the motor. (reset by SAP 29 or after the time specified by the I^2t thermal winding time constant) *Flag 0 to 15 are automatically reset. Only flag 17 must be cleared manually.*	0…+4294967295	R

6 Global Parameter Overview (SGP, GGP, STGP, RSGP)

The following section describes all global parameters that can be used with the SGP, GGP, STGP and RSGP commands.

TWO BANKS ARE USED FOR GLOBAL PARAMETERS:

- Bank 0 (global configuration of the module)
- Bank 2 (user TMCL variables)

6.1 Bank 0

PARAMETERS 64… 255

Parameters below 63 configure stuff like the serial address of the module RS232 baud rate or the telegram pause time. Change these parameters to meet your needs. The best and easiest way to do this is to use the appropriate functions of the TMCL-IDE. The parameters between 64 and 85 are stored in EEPROM only. A SGP command on such a parameter will always store it permanently and no extra STGP command is needed.

Take care when changing these parameters and use the appropriate functions of the TMCL-IDE to do it in an interactive way!

MEANING OF THE LETTERS IN COLUMN *ACCESS*:

Access type	Related command(s)	Description
R	GGP	Parameter readable
W	SGP, AGP	Parameter writable
E	STGP, RSGP	Parameter automatically restored from EEPROM after reset or power-on.

GLOBAL PARAMETERS OF BANK 0

Number	Global parameter	Description	Range	Access
64	EEPROM magic	Setting this parameter to a different value as $E4 will cause re-initialization of the axis and global parameters (to factory defaults) after the next power up. This is useful in case of miss-configuration.	0… 255	RWE
65	RS232 baud rate	0 – 9600 baud – *Default* 1 – 14400 baud 2 – 19200 baud 3 – 28800 baud 4 – 38400 baud 5 – 57600 baud 6 – 76800 baud – *Not supported by Windows!* 7 – 115200 baud	0… 7	RWE
66	Serial address	The module (target) address for RS232 and virtual COM port	0… 255	RWE
73	Configuration EEPROM lock flag	Write: 1234 to lock the EEPROM, 4321 to unlock it. Read: 1=EEPROM locked, 0=EEPROM unlocked.	0/1	RWE
75	Telegram pause time	Pause time before the reply via RS232 is sent. For RS232 set to 0.	0… 255	RWE
76	Serial host address	Host address used in the reply telegrams sent back via RS232.	0… 255	RWE
77	Auto start mode	0: Do not start TMCL application after power up (*default*). 1: Start TMCL application automatically after power up. *Note: the current initialization has to be finished first.*	0/1	RWE

Number	Global parameter	Description	Range	Access
81	TMCL code protection	Protect a TMCL program against disassembling or overwriting. 0 – no protection 1 – protection against disassembling 2 – protection against overwriting 3 – protection against disassembling and overwriting *If you switch off the protection against disassembling, the program will be erased first!* *Changing this value from 1 or 3 to 0 or 2, the TMCL program will be wiped off.*	0, 1, 2, 3	RWE
85	Do not restore user variables	0 – user variables are restored (*default*) 1 – user variables are not restored	0/1	RWE
128	TMCL application status	0 –stop 1 – run 2 – step 3 – reset	0… 3	R
129	Download mode	0 – normal mode 1 – download mode *Attention:* *Download mode can only be used if the motor has been stopped first. Otherwise the download mode setting will be disallowed.* *During download mode the motor driver will be deactivated and the actuator will be turned off.*	0/1	R
130	TMCL program counter	The index of the currently executed TMCL instruction.	0… 2047	R
132	Tick timer	A 32 bit counter that gets incremented by one every millisecond. It can also be reset to any start value.	0… +4294967295	RW
255	Suppress reply	0 – reply (*default*) 1 – no reply	0/1	RW

6.2 Bank 2

Bank 2 contains general purpose 32 bit variables for the use in TMCL applications. They are located in RAM and can be stored to EEPROM. After booting, their values are automatically restored to the RAM.

Up to 256 user variables are available.

Meaning of the Letters in Column *Access*:

Access type	Related command(s)	Description
R	GGP	Parameter readable
W	SGP, AGP	Parameter writable
E	STGP, RSGP	Parameter automatically restored from EEPROM after reset or power-on.

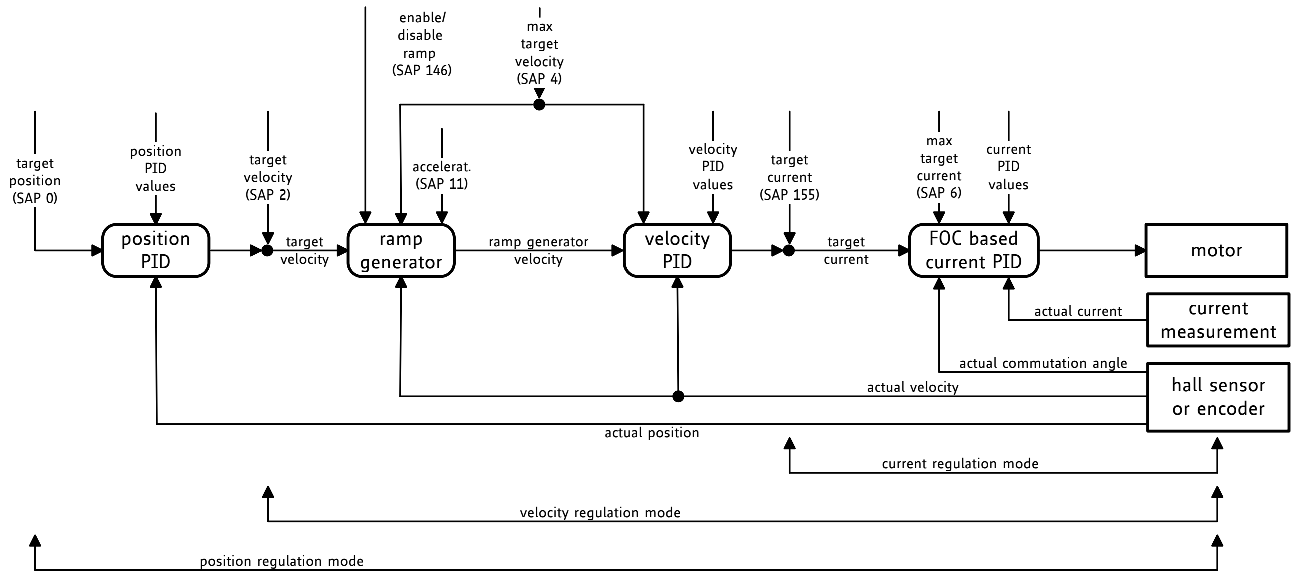
Global Parameters of Bank 2

Number	Global parameter	Description	Range	Access
0… 55	general purpose variable #0… 55	for use in TMCL applications	$-2^{31}…+2^{31}$ (int32)	RWE
56… 255	general purpose variables #56… #255	for use in TMCL applications	$-2^{31}…+2^{31}$ (int32)	RW

7 Motor Regulation

7.1 Structure of the Cascaded Motor Regulation Modes

The TMC603-EVAL supports a current, velocity, and position PID regulation mode for motor control in different application areas. These regulation modes are cascaded as shown in figure 7.1. The individual modes are explained in the following sections.



7.1 Cascaded regulation

7.2 Current Regulation

The current regulation mode uses a PID regulator to adjust a desired motor current. This target current can be set by axis parameter 155. The maximal target current is limited by axis parameter 6.

The PID regulation uses three basic parameters: The *P* and *I* value as well as the *timing control value*.

Timing Control Value

The timing control value (*current regulation loop multiplier*, axis parameter 134) determines how often the current regulation is invoked. It is given in multiple of 50µs:

$$t_{PIDDELAY} = x_{PIDRLD} \cdot 50\mu s$$

$t_{PIDDELAY}$ = resulting delay between two current regulation loops
x_{PIDRLD} = *current regulation loop multiplier* parameter

For most applications it is recommended to leave this parameter unchanged at its default of 2*50µs. Higher values may be necessary for very slow and less dynamic drives.

Structure of the Current Regulator

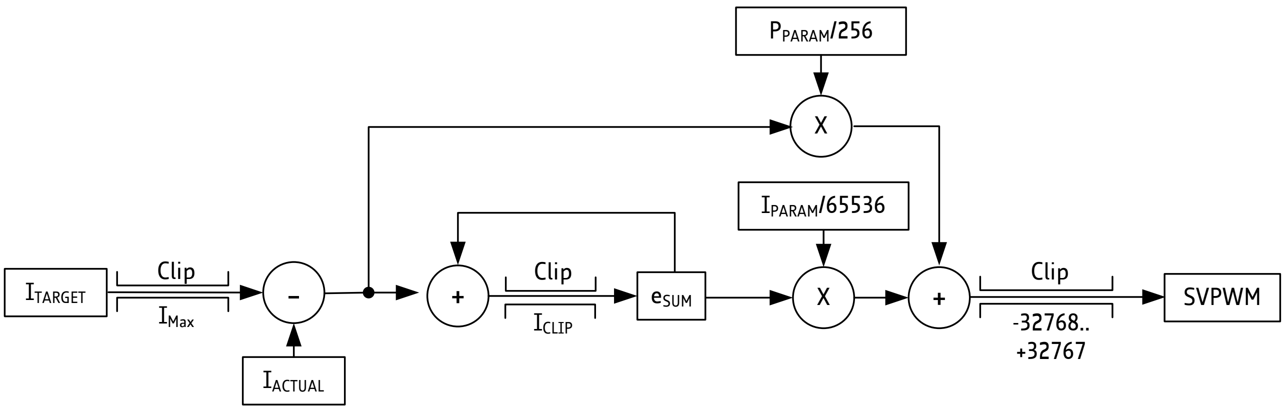


Figure 7.2 Current regulation

Parameter	Description
I_{ACTUAL}	Actual motor current (GAP 150)
I_{TARGET}	Target motor current (SAP 155)
I_{Max}	Max. motor current (SAP 6)
e_{SUM}	Error sum for integral calculation (GAP 201)
P_{PARAM}	Current P parameter (SAP 172)
I_{PARAM}	Current I parameter (SAP 173)

Parameterizing the Current Regulator Set

1. Set the *P* parameter and the *I* parameter to zero.
2. Start the motor by using a low target current (e.g. 1000 mA).
3. Modify the current *P* parameter. Start from a low value and go to a higher value, until the actual current nearly reaches 50% of the desired target current.
4. Do the same with the current *I* parameter.

For all tests set the motor current limitation to a realistic value, so that your power supply does not become overloaded during acceleration phases. If your power supply reaches current limitation, the unit may reset or undetermined regulation results may occur.

7.3 Velocity Regulation

Based on the current regulation the motor velocity can be controlled by the velocity PID regulator.

Timing Control Value

Also, the velocity PID regulator uses a timing control value (*PID regulation loop delay*, axis parameter 133) which determines how often the PID regulator is invoked. It is given in multiple of 1ms:

$$t_{PIDDELAY} = x_{PIDRLD} \cdot 1\text{ms}$$

$t_{PIDDELAY}$ = resulting delay between two PID calculations
x_{PIDRLD} = *PID regulation loop delay* parameter

For most applications it is recommended to leave this parameter unchanged at its default value of 1ms. Higher values may be necessary for very slow and less dynamic drives.

Structure of the Velocity Regulator

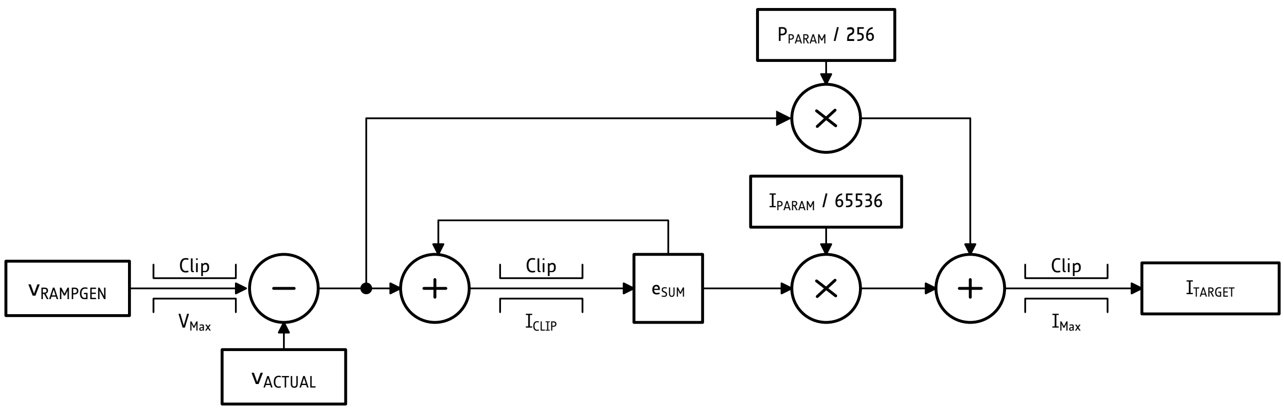


Figure 7.3 Velocity regulation

Parameter	Description
v_{ACTUAL}	Actual motor velocity (GAP 3)
$v_{RAMPGEN}$	Target velocity of ramp generator (SAP 2, GAP 13)
v_{Max}	Max. target velocity (SAP 4)
e_{SUM}	Error sum for integral calculation (GAP 229)
P_{PARAM}	Velocity P parameter (SAP 234)
I_{PARAM}	Velocity I parameter (SAP 235)
I_{Max}	Max. target current (SAP 6)
I_{Target}	Target current for current PID regulator (GAP 155)

Parameterizing the Velocity Regulator Set

1. Set the *velocity I parameter* to zero.
2. Start the motor by using a medium target velocity (e.g. 2000 rpm).

3. Modify the *velocity P parameter*. Start from a low value and go to a higher value, until the actual motor speed reaches 80 or 90% of the target velocity.
4. The lasting 10 or 20% speed difference can be reduced by slowly increasing the *velocity I parameter*.

7.4 Velocity Ramp Generator

For a controlled start up of the motor's velocity a velocity ramp generator can be activated/deactivated by axis parameter 146. The ramp generator uses the maximal allowed motor velocity (axis parameter 4), the acceleration (axis parameter 11) und the desired target velocity (axis parameter 2) to calculate a ramp generator velocity for the following velocity PID regulator.

7.5 Position Regulation

Based on current and velocity regulators the TMC603-EVAL supports a positioning mode based on encoder or hall sensor position. During positioning the velocity ramp generator can be activated to enable motor positioning with controlled acceleration or it can be disabled to support motor positioning with max allowed speed.

The PID regulation uses two basic parameters: the *P* regulation and a timing control value.

Timing Control Value

The timing control value (*PID regulation loop parameter* - axis parameter 133) determines how often the PID regulator is invoked. It is given in multiple of 1ms:

$$t_{PIDDELAY} = x_{PIDRLD} \cdot 1\text{ms}$$

$t_{PIDDELAY}$ = the resulting delay between two position regulation loops

x_{PIDRLD} = *PID regulation loop multiplier* parameter

For most applications it is recommended to leave the timing control value unchanged at its default of 1ms. Higher values may be necessary for very slow and less dynamic drives.

Structure of the Position Regulator

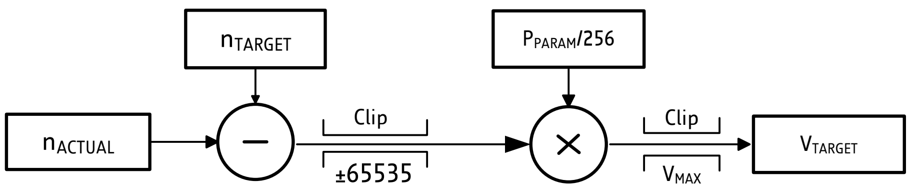


Figure 7.4 Positioning regulation

Parameter	Description
n_{ACTUAL}	Actual motor position (GAP 1)
n_{TARGET}	Target motor position (SAP 0)
P_{PARAM}	Position P parameter (SAP 230)
V_{MAX}	Max. allowed velocity (SAP 4)
V_{TARGET}	New target velocity for ramp generator (GAP 13)

Parameterizing the Position Regulation

Based on the velocity regulator only the position regulator P has to be parameterized.

1. Disable the velocity ramp generator and set position *P* parameter to zero.
2. Choose a target position and increase the position *P* parameter until the motor reaches the target position approximately.

3. Switch on the *velocity ramp generator*. Based on the *max. positioning velocity* (axis parameter 4) and the *acceleration value* (axis parameter 11) the ramp generator automatically calculates the *slow down point*, i.e. the point at which the velocity has to be reduced in order to stop at the desired target position.
4. Reaching the target position is signaled by setting the *position end flag*.

NOTE:
- In order to minimize the time until this flag becomes set, the positioning tolerance *MVP target reached distance* can be chosen with axis parameter 10.
- Since the motor typically is assumed not to signal target reached when the target was just passed in a short moment at a high velocity, additionally the maximum target reached velocity (*MVP target reached velocity)* can be defined by axis parameter 7.
- A value of zero for axis parameter 7 is the most universal, since it implies that the motor stands still at the target. But when a fast rising of the *position end flag* is desired, a higher value for the *MVP target reached velocity* parameter will save a lot of time. The best value should be tried out in the actual application.

CORRELATION OF AXIS PARAMETERS 10 AND 7, THE TARGET POSITION, AND THE POSITION END FLAG

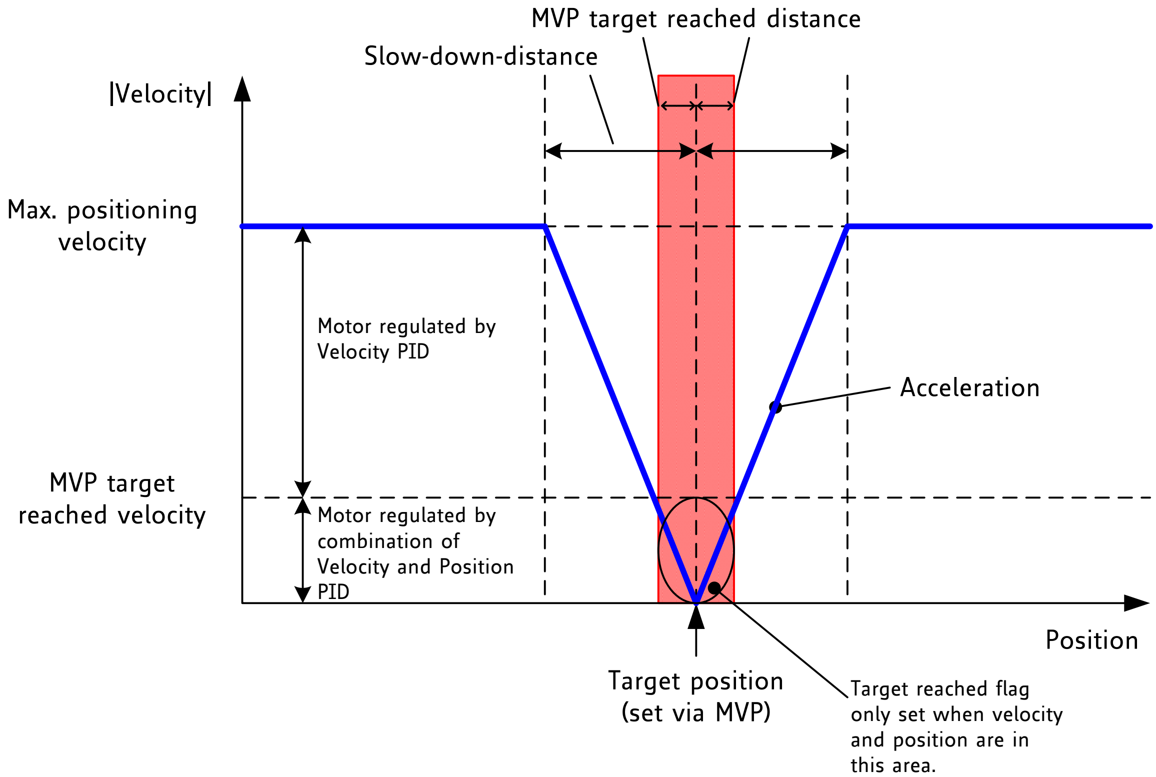


Figure 7.5 Positioning algorithm

Depending on motor and mechanics a low oscillation is normal. This can be reduced to at least +/-1 encoder steps. Without oscillation the regulation cannot keep the position!

8 Temperature Calculation

Axis parameter 152 delivers the actual ADC value of the motor driver. This ADC value can be converted to a temperature in °C as follows:

$$\mathrm{ADC} = actual\ value\ of\ GAP\ 152$$

$$\mathrm{B} = 3434\ (material\ constant)$$

$$\mathrm{R_{NTC}} = \frac{9011{,}2}{ADC} - 2.2$$

$$\mathrm{T} = \frac{B * 298{,}16}{B + (\ln\left(\frac{R_{NTC}}{10}\right) * 298.16} - 273.16\ ℃$$

Example 1:

ADC = 1000
R_{NTC} ≈ 6.81
T ≈ 35℃

Example 2:

ADC = 1200
R_{NTC} ≈ 5.31
T ≈ 42℃

9 I²t Monitoring

The I²t monitor determines the sum of the square of the motor current over a given time. The integrating time is motor specific. In the datasheet of the motor this time is described as *thermal winding time constant* and can be set for each module using axis parameter 25. The number of measurement values within this time depends on how often the current regulation and thus the I²t monitoring is invoked. The value of the actual I²t sum can be read by axis parameter 27. With axis parameter 26 the default value for the I²t limit can be changed (default: 211200). If the actual I²t sum exceeds the I²t limit of the motor, flag 17 (in axis parameter 156) is set and the motor pwm is set to zero as long as the I²t exceed flag is set. The actual regulation mode will not be changed. Furthermore, the I²t exceed counter is increased once every second as long as the actual I²t sum exceeds the I²t limit. The I²t exceed flag can be cleared manually using parameter 29 but only after the cool down time given by the *thermal winding time constant* has passed. The I²t exceed flag will not be reset automatically. The I²t limit can be determined as follows:

$$I^2t = \frac{I\ [mA]}{1000} * \frac{I\ [mA]}{1000} * t_{tw}\ [ms]$$

I is the desired average current
t_{tw} is the thermal winding time constant given by the motor datasheet

Example:

I²t limits for an average current of a) 1A, b) 2A, c) 3A and d) 4A over a thermal winding time of 13,2s.

a) $I^2t\ limit = \frac{1000\ [mA]}{1000} * \frac{1000\ [mA]}{1000} * 13200\ [ms] = 13200\ [mA^2 * ms]$

b) $I^2t\ limit = \frac{2000\ [mA]}{1000} * \frac{2000\ [mA]}{1000} * 13200\ [ms] = 52800\ [mA^2 * ms]$

c) $I^2t\ limit = \frac{3000\ [mA]}{1000} * \frac{3000\ [mA]}{1000} * 13200\ [ms] = 118800\ [mA^2 * ms]$

d) $I^2t\ limit = \frac{4000\ [mA]}{1000} * \frac{4000\ [mA]}{1000} * 13200\ [ms] = 211200\ [mA^2 * ms]$

10 Life Support Policy

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.

11 Revision History

11.1 Firmware Revision

Version	Date	Author	Description
2.00	2012-JUL-10	ED	New FOC firmware
2.02	2012-DEC-14	ED	- Axis parameter 209 deleted. - Axis parameter 241 (sine initialization speed) added. - Axis parameter 31 (BLDC re-initialization) added. - Global parameter 77 (auto start mode) updated. - Global parameter 129 (download mode) updated.
2.05	2013-APR-03	ED	- Axis parameter 159 updated: new FOC controlled mode. - Several axis parameter value ranges updated. - Axis parameter 212 new.

Table 11.1 Firmware revision

11.2 Document Revision

Version	Date	Author	Description
2.00	2012-AUG-06	SD	Manual for new *Field Orientated Control* (FOC) firmware - Commands SIO and GIO added - Axis parameters updated - Motor regulation updated
2.01	2013-JAN-05	SD	- Axis parameter 209 deleted. - Axis parameter 241 (sine initialization speed) added. - Axis parameter 31 (BLDC re-initialization) added. - Global parameter 77 (auto start mode) updated. - Global parameter 129 (download mode) updated.
2.02	2013-APR-03	SD	- Axis parameter 159 updated: new FOC controlled mode. - Several axis parameter value ranges updated. - Axis parameter 212 (actual controlled angle) new.

Table 11.2 Document revision

12 References

[TMC603] TMC603 Datasheet (please refer to our homepage http://www.trinamic.com)