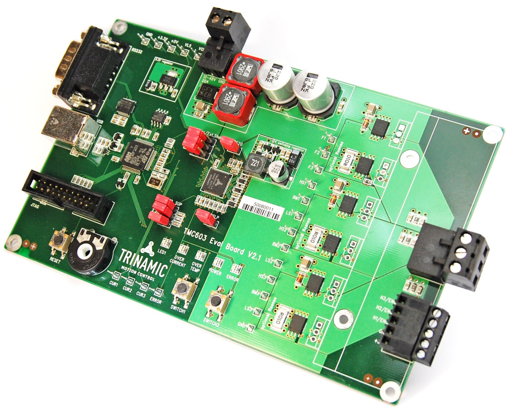
TMC603EVAL – MANUAL

Evaluation board for the TMC603 three phase motor driver with BLDC back EMF commutation hallFX™

TRINAMIC® Motion Control GmbH & Co. KG
Sternstraße 67
D – 20357 Hamburg
GERMANY
www.trinamic.com



1 Features

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

Motor type

- 3 phase BLDC motor
- Sine or block commutation
- Rotor position feedback: sensorless, encoder or hall sensor

Highlights

- Up to 18.8A (IRMS) nominal motor current
- 12V to 48V operating voltage
- Integrated current measurement using power MOS transistor RDSon
- hallFX™ sensorless back EMF commutation emulates hall sensors
- Integrated Break-before-Make logic: No special microcontroller PWM hardware required
- EMV optimized current controlled gate drivers
- Overcurrent / Short to GND and undervoltage protection and diagnostics integrated
- Internal QGD protection: Supports latest generation of Power MOSFETs
- Integrated supply concept: Step down switching regulator
- Common rail charge pump allows for 100% PWM duty cycle
- Communication to the PC via USB interface and RS232 interface
- Firmware update via USB interface and RS232 interface

Please check our website for the latest version of manual and firmware!

www.trinamic.com

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.

2 Contents

3 Functional Description

The TMC603 is a BLDC driver IC using external power MOS transistors. Its unique feature set allows equipping inexpensive and small drive systems with a maximum of intelligence, protection and diagnostic features. Depending on the desired commutation scheme and the bus interface requirements, the TMC603 forms a complete motor driver system in combination with an external microcontroller (STM32F ARM Cortex-M3). The complete analog amplification and filtering frontend as well as the power driver controller are realized in the TMC603.

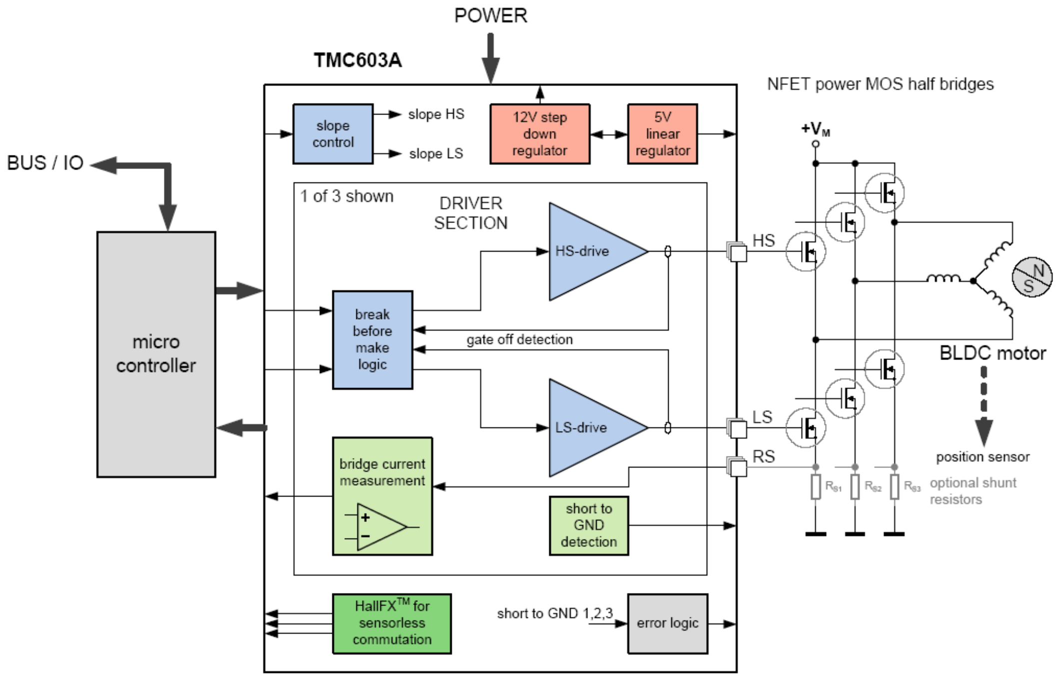


Figure 1: Structure of TMC603 evaluation board

The TMC603 evaluation board can either controlled via the PC software or the TMCL IDE or by a user specific program. This manual describes the BLDC specific parameters and BLDC specific commands available on the TMC603 evaluation board. Further, the evaluation board provides control instructions like branches, arithmetic functions, comparison commands and functions for reading / setting I/O lines. For these, please refer to the TMCL reference manual.

The PC software for the TMC603 evaluation board gives a simple way to set the evaluation board into operation. The TRINMAIC TMCL IDE allows to update the evaluation board's firmware and to set all of the evaluation board's parameters via RS232 or USB interface.

Attention: Please note that the BLDC motors may only be connected to or disconnected from the evaluation board while the board is disconnected from power supply, as otherwise the evaluation board could get damaged!

Attention: It is important to properly set motor and encoder settings before trying to operate the motor.

3.1 hallFX sensorless commutation

hallFX™ provides emulated hall sensor signals. The emulated hall sensor signals are available without a phase shift. Neither an error-prone PLL is necessary, like with many other systems, nor the knowledge of special motor parameters is required. Since it is based on the motors' back-EMF, a minimum motor velocity is required to get a valid signal. Therefore, the motor needs to be started without feedback, until the velocity is high enough to generate a reliable hallFX™ signal.

hallFX™ works perfectly with nearly every motor. It can be used a standard block commutation scheme, but the chopper must fulfill the following requirement: The coils must be open for some percentage of the chopper period, in order to allow the back-EMF of the motor to influence the coil voltages. The motor direction is determined by the start-up scheme, since the hallFX™ signals depend on the direction. Thus, the same commutation scheme is used for turn right and turn left. Only a single commutation table is required.

In a practical application, the motor can run with a duty cycle of 15% to 25% (minimum motor velocity at low load) up to 90% to 95% (maximum motor velocity). The exact values depend on the actual motor. With a lower duty cycle the motor would not start, or back EMF would be too small to yield a valid hallFX™ signal. With a higher duty cycle, the back EMF would not be visible at the coil voltages, because the coils would be connected to GND or VM nearly the whole time. The minimum resulting coil open time thus is 5% to 10%. This simple chopper scheme automatically gives a longer measuring time at low velocities, when back EMF is lower.

In order to start the motor running with hallFX™, it must reach a minimum velocity. The microcontroller needs to take care of this by starting the motor in a forward control mode, without feedback. The motor is turned up to a minimum velocity threshold, where you safely get correct hallFX™ signals. Since rotation of the motor cannot be measured during this phase, the motor needs to be current controlled. The current should be high enough to turn the mechanical load. For current control, a current PID regulator is available. For detailed description please refer to chapter 3.2.

Further, the acceleration and the minimum velocity threshold needs to be set matching the motor and application. The turning direction of the startup sequence automatically determines the direction of motor operation with hallFX™.

For more information about the three phase motor driver TMC603A with BLDC back EMF commutation hallFX™ and current sensing, please refer to TMC603A datasheet.

3.2 Parameterizing the current PID regulation and current limitation

The TMC603 evaluation board supports two commutation modes for forward control, like a stepper motor. These commutation modes are called controlled block commutation and controlled sine commutation. For these modes a current PID regulation is available. The target current can be set by the 'Start current' parameter which in every case is high enough to turn the mechanical load. The PID regulator has to be parameterized with respect to a given motor in a given application. Additionally, a current limitation is integrated. The maximum current limit is given by the 'Max. current' parameter.

The PID regulation uses five basic parameters. The P, I and D value as well as a I clipping value and a timing control value. The timing control value ('Current regulation loop delay' parameter) determines, how often the current regulation is evoked. It is given in multiple of 50µs:

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

$t_{PIDDELAY}$ is the resulting delay between two PID calculations.

x_{PIDRLD} is the 'Current regulation loop delay' parameter.

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

The PID regulation provides two parameter sets. These PID parameter sets are used as follows. Below a specified velocity threshold the motor current is regulated by a combination of parameter set 1 and parameter set 2 to get the best performance. So the switch over between both parameter sets is soft. Above the velocity threshold the motor current is only regulated by the parameter set 2. If the velocity threshold is set to zero, the parameter set 2 is used all the time. The velocity threshold is given by 'Velocity threshold for current PID' parameter.

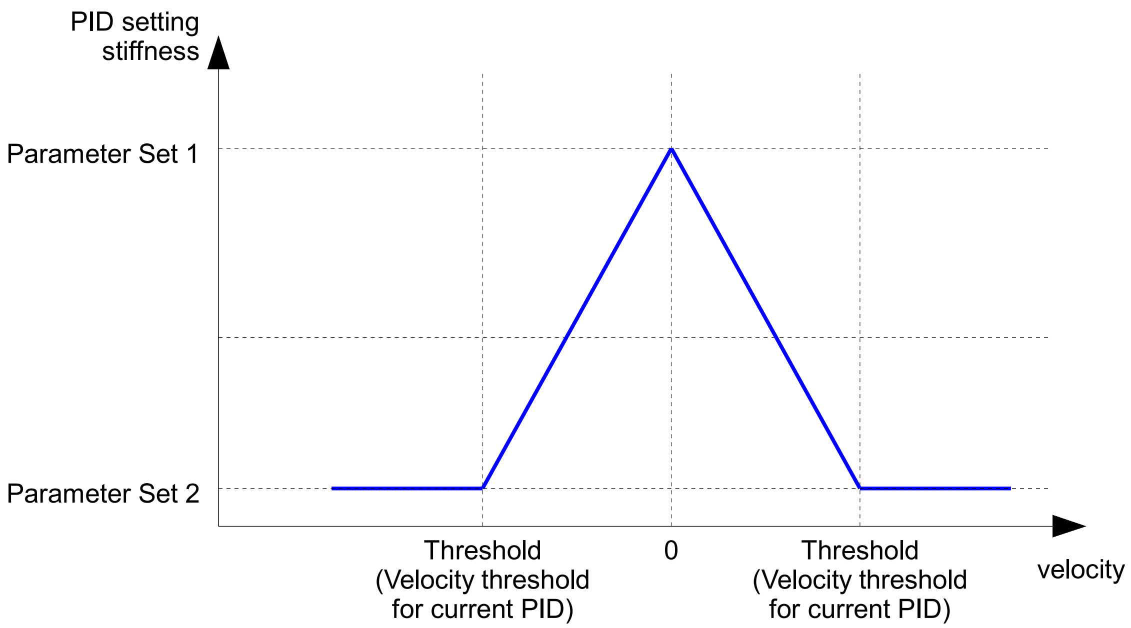


Figure 2: Current PID parameter sets

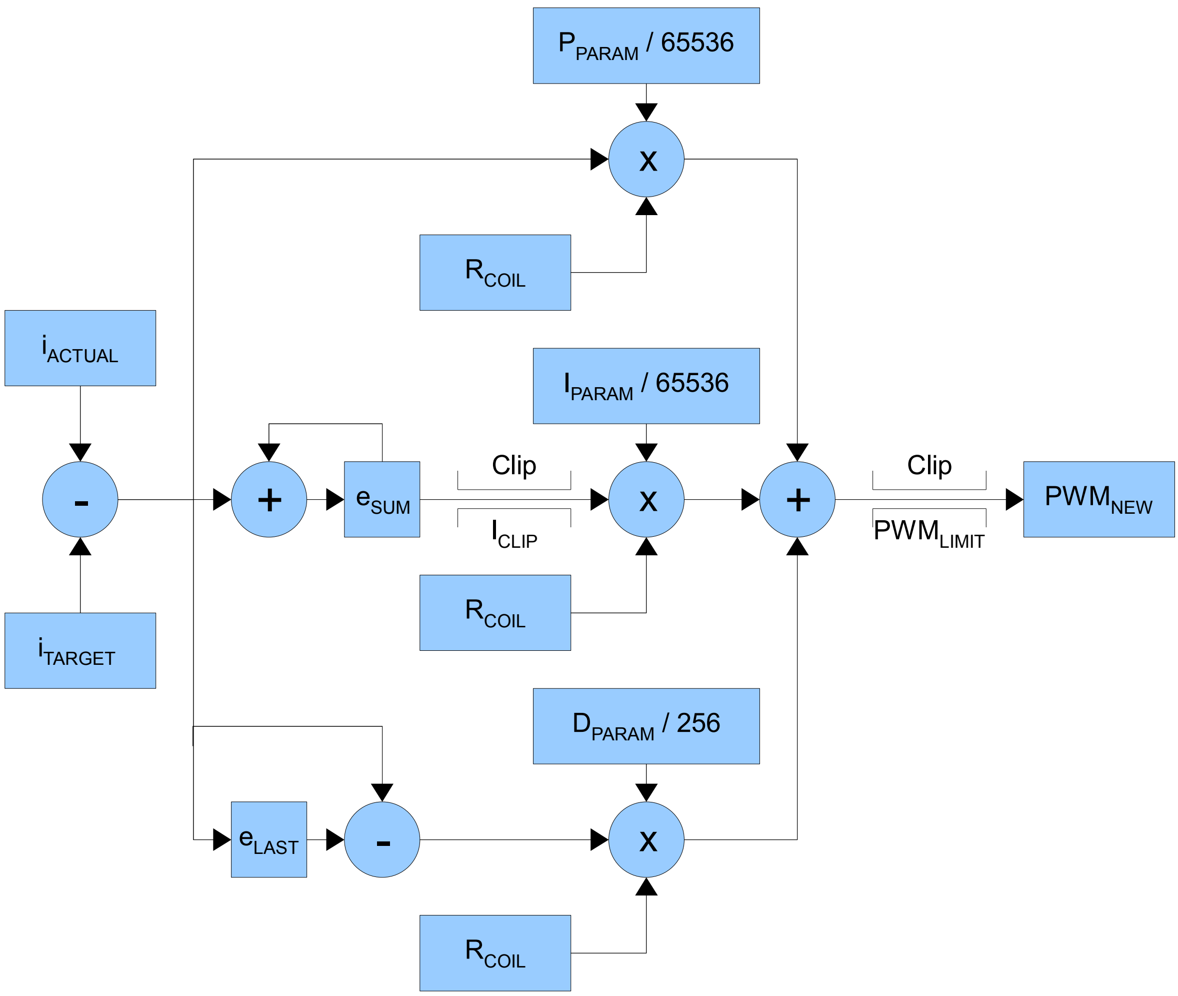


Figure 3: Current PID regulation

Parameter	Description
i_{ACTUAL}	Actual motor current
i_{TARGET}	Start current parameter
e_{LAST}	Error value of the last PID calculation
e_{SUM}	Error sum for integral calculation
R_{COIL}	Motor coil resistance parameter
P_{PARAM}	Current P-Parameter
I_{PARAM}	Current I-Parameter
D_{PARAM}	Current D-Parameter
I_{CLIP}	Current I-Clipping-Parameter
PWM_{LIMIT}	PWM Limit
PWM_{NEW}	New PWM value

Table 1: Current PID parameter description

To parameterize the current PID regulator for a given motor, first set the P, I and D parameter of both parameter sets to zero. Then switch to commutation mode 'controlled block commutation' or 'controlled sine commutation' respectively, and start the motor by using a low target velocity. Then modify the 'Current P-Parameter (II)'. This is the P parameter of parameter set 2. Start from a low value and go to a higher value, until the actual current nearly reaches the target current ('Start current' parameter).

After that, do the same for the 'Current I-Parameter (II)' with the 'Current D-Parameter (II)' still set to zero. For the 'Current I-Parameter (II)', there is also a clipping value. The 'Current I-Clipping Parameter (II)' should be set to a relatively low value to avoid overshooting, but high enough to reach the target current ('Start current' parameter). The 'Current D-Parameter (II)' can still be set to zero.

After having found suitable values for parameter set 2, the first parameter set (PID Parameter Set 1) should be set to much lower values, to minimize overshooting during zero-time of motor start. Then stop the motor and start again to test the current regulation settings. When the motor current is overshot during zero-time, set the PID parameter set 1 to once more lower values.

Attention: 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 goes to current limitation, the unit may reset or undetermined regulation results may occur.

3.3 Parameterizing the velocity PID regulation

The TMC603 evaluation board supports a velocity regulation. In this mode the motor speed is regulated by a velocity PID regulator.

The PID regulation uses five basic parameters. The P, I and D value as well as a I clipping value and a timing control value. The timing control value ('PID regulation loop delay' parameter) determines, how often the PID regulator is evoked. It is given in multiple of 1ms:

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

$t_{PIDDELAY}$ is the resulting delay between two PID calculations.

x_{PIDRLD} is the 'PID regulation loop delay' parameter.

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

Two different velocity PID regulations are available. The first PID algorithm is a standard velocity PID regulation. For this regulator a feed forward control is optionally available. The second PID algorithm is a integrating PID regulation. This means that the result of the PID calculation is integrated to the last PWM value. Both PID regulators are described in chapter 3.3.1 and 3.3.2. In particular the parameterization of PID regulation is specified.

The PID regulation provides two parameter sets. These PID parameter sets are used as follows. Below a specified velocity threshold the motor speed is regulated by a combination of parameter set 1 and parameter set 2 to get the best performance. So the switch over between both parameter sets is soft. Above the velocity threshold the motor speed is only regulated by the parameter set 2. If the velocity threshold is set to zero, the parameter set 2 is used all the time. The velocity threshold is given by 'Min. speed for position PID' parameter.

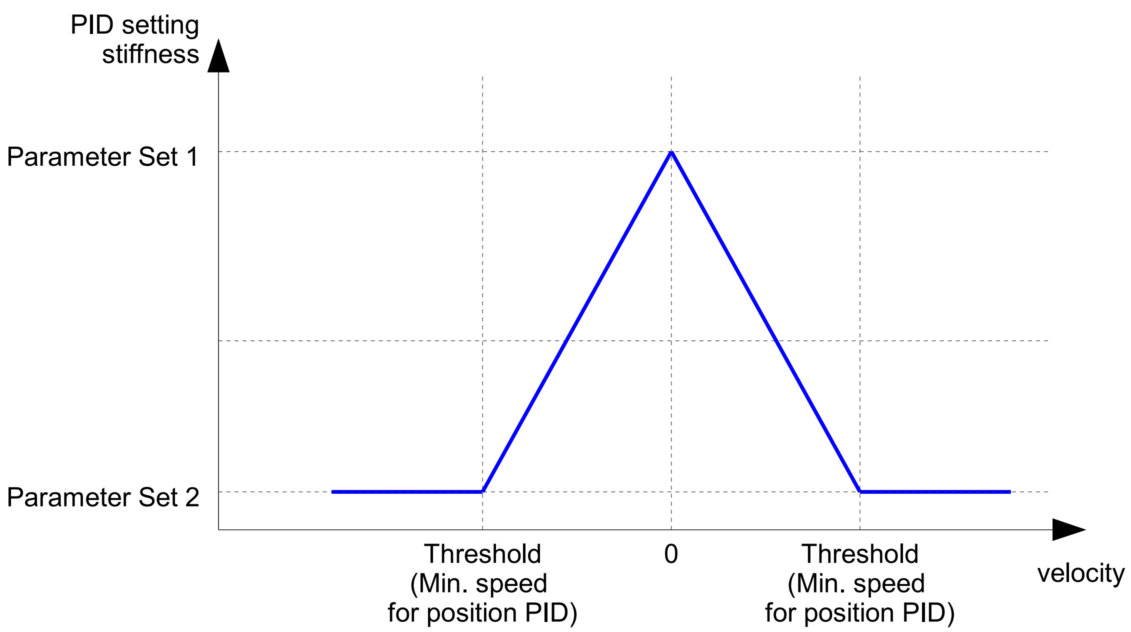


Figure 4: Velocity PID parameter sets

For the first tests, you can set both PID parameter sets equal. After having found suitable values, the second parameter set (PID Parameter Set 2) should be set to lower values, to minimize regulation oscillation during constant rotation.

3.3.1 Parameterizing the standard velocity PID regulation

The speed regulation uses a standard velocity PID regulator (feedback control). The user can either use the pure velocity PID parameter set for speed regulation (Figure 5) or an additional feed forward control (Figure 6). The feed forward control calculates a PWM value to overcome the actual BEMF voltage of the motor. This calculation is based on the BEMF constant, the coil resistance of the motor and the number of motor poles. It is recommended to use the velocity PID regulation in combination with the integrated feed forward control.

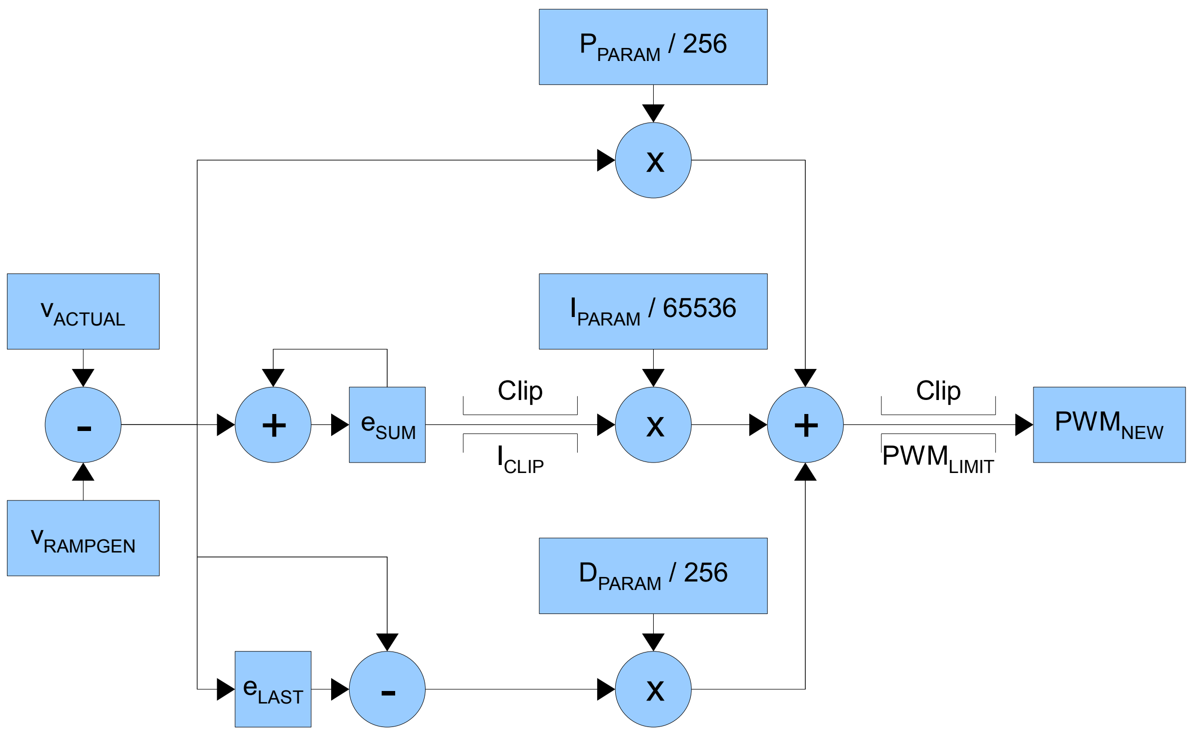


Figure 5: Standard velocity PID regulation

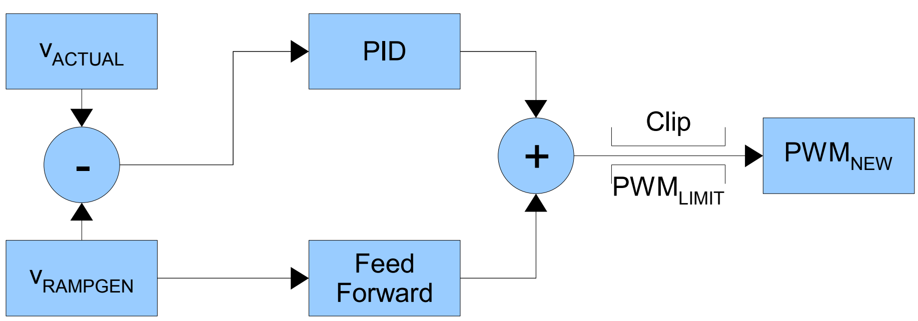


Figure 6: Velocity PID regulation with feed forward control

Parameter	Description
v_{ACTUAL}	Actual motor speed
$v_{RAMPGEN}$	Target speed of ramp generator
e_{LAST}	Error value of the last PID calculation
e_{SUM}	Error sum for integral calculation
P_{PARAM}	Velocity P-Parameter
I_{PARAM}	Velocity I-Parameter
D_{PARAM}	Velocity D-Parameter
I_{CLIP}	Velocity I-Clipping-Parameter
PWM_{LIMIT}	PWM Limit
PWM_{NEW}	New PWM value

Table 2: Velocity PID parameter description (1)

To parameterize the PID regulator for a given motor, first set the 'BEMF constant' in [rpm/(10V)], the 'Motor coil resistance' parameter in [$m\Omega$] and the 'Number of motor poles' to the correct value. The specific parameters can be find in the datasheet of the given motor. In most datasheets the BEMF constant can't be found. In that case the BEMF constant can be calculated by using the torque constant. Within SI units, the numeric value of the BEMF constant has the same numeric value as the numeric value of the torque constant. For example, a BLDC motor with a torque constant of 1Nm/A would have a BEMF constant of 1Vs/rad. The BEMF constant in [rpm/V] can be calculated as:

$$K_{BEMF} = \frac{60}{2 \cdot \pi \cdot K_{TORQUE}} \quad \text{in} \quad \frac{rpm}{V}$$

K_{BEMF} is the resulting BEMF constant of the given motor.

K_{TORQUE} is the torque constant from motor datasheet.

The torque constant is in unit [Nm/A] where 1Nm/A = 100cNm/A = 1000mNm/A. The resulting BEMF constant is in unit [rpm/V]. The additional factor follows from conversion of unit [Vs/rad] in unit [rpm/V]:

$$1 \frac{V}{rad/s} = \frac{1}{60} \frac{V}{rad/min} = \frac{2\pi}{60} \frac{V}{rpm} \quad \rightarrow \quad \frac{60}{2\pi} \frac{rpm}{V}$$

Example:

$$K_{TORQUE} = 70.5 \frac{mNm}{A} \quad \text{(torque constant from datasheet)}$$

$$K_{BEMF} = \frac{60}{2 \cdot \pi \cdot 0.0705 \frac{Nm}{A}} \approx 135 \frac{rpm}{V}$$

After setting the specific motor parameter, set the 'Velocity I-Parameter' and 'Velocity D-Parameter' to zero and start the motor by using a medium target velocity. Then modify the 'Velocity P-Parameter'. Start from a low value and go to a higher value, until fastest response without any oscillation is given (touch the motor / watch the actual velocity and motor current). To modify parameters, always use the double or half of the previous value, in order to really see how the parameter influences behavior.

After that, do the same for the 'Velocity I-Parameter' with the 'Velocity D-Parameter' still set to zero. For the 'Velocity I-Parameter', there is also a clipping value. The 'Velocity I-Clipping Parameter' should be set to a relatively low value to avoid overshooting, but high enough to reach the target velocity. Now, modify the 'Velocity D-Parameter' in the same way. It will dampen part of the oscillations caused by the other parameters, too.

For the first tests, you can set both PID parameter sets equal. After having found suitable values, the second parameter set (PID Parameter Set 2) should be set to lower values, to minimize regulation oscillation during constant rotation.

Attention: 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 goes to current limitation, the unit may reset or undetermined regulation results may occur.

3.3.2 Parameterizing the integrating velocity PID regulation

The speed control uses a integrating velocity PID regulation. This means that the result of the PID calculation is integrated to the last PWM value (Figure 7). This PID regulation is easier to use than the standard PID regulation, because no specific motor parameters are necessary to parameterize a given motor.

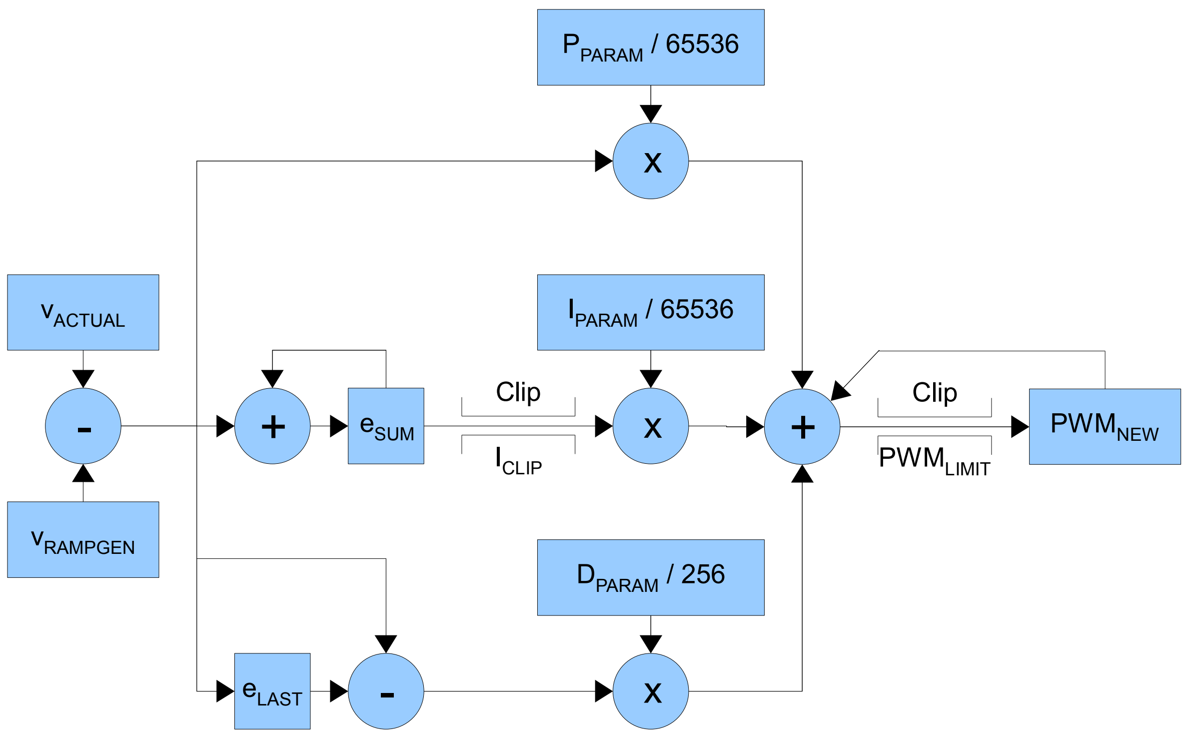


Figure 7: Integrating velocity PID regulation

Parameter	Description
v_{ACTUAL}	Actual motor speed
$v_{RAMPGEN}$	Target speed of ramp generator
e_{LAST}	Error value of the last PID calculation
e_{SUM}	Error sum for integral calculation
P_{PARAM}	Velocity P-Parameter
I_{PARAM}	Velocity I-Parameter
D_{PARAM}	Velocity D-Parameter
I_{CLIP}	Velocity I-Clipping-Parameter
PWM_{LIMIT}	PWM Limit
PWM_{NEW}	New PWM value

Table 3: Velocitiy PID parameter description (2)

To parameterize the PID regulator for a given motor, first set the 'Velocity I-Parameter' and 'Velocity D-Parameter' to zero and modify the 'Velocity P-Parameter'. Start from a low value and go to a higher value, until fastest response with minimum oscillation around the target velocity is given (touch the motor / watch the actual position and motor current). To modify parameters, always use the double or half of the previous value, in order to really see how the parameter influences behavior.

After that, do the same for the 'Velocity I-Parameter' with the 'Velocity D-Parameter' still set to zero. For the 'Velocity I-Parameter', there is also a clipping value. The 'Velocity I-Clipping Parameter' should be set to a relatively low value to avoid overshooting, but high enough to reach the target velocity. Now, modify the 'Velocity D-Parameter' in the same way. It will dampen part of the oscillations caused by the other parameters, too.

For the first tests, you can set both PID parameter sets equal. After having found suitable values, the second parameter set (PID Parameter Set 2) should be set to lower values, to minimize regulation oscillation during constant rotation.

Attention: 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 goes to current limitation, the unit may reset or undetermined regulation results may occur.

3.4 Parameterizing the positioning algorithm

The TMC603 evaluation board supports a positioning mode based on the encoder position or hall sensor position. This mode uses a PID regulator for position control. The PID regulator has to be parameterized with respect to a given motor in a given application.

The PID regulation uses five basic parameters. The P, I and D value as well as a I clipping value and a timing control value. The timing control value ('PID regulation loop delay' parameter) determines, how often the PID regulator is evoked. It is given in multiple of 1ms:

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

$t_{PIDDELAY}$ is the resulting delay between two PID calculations.

x_{PIDRLD} is the 'PID regulation loop delay' parameter.

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

To parameterize the PID regulator for a given motor, first set the 'Position I-Parameter' and 'Position D-Parameter' to zero and modify the 'Position P-Parameter'. Start from a low value and go to a higher value, until fastest response with minimum oscillation around the target position is given (touch the motor / watch the actual position and motor current). Try with low velocities and stand still if you want to tune positioning response. To modify parameters, always use the double or half of the previous value, in order to really see how the parameter influences behavior.

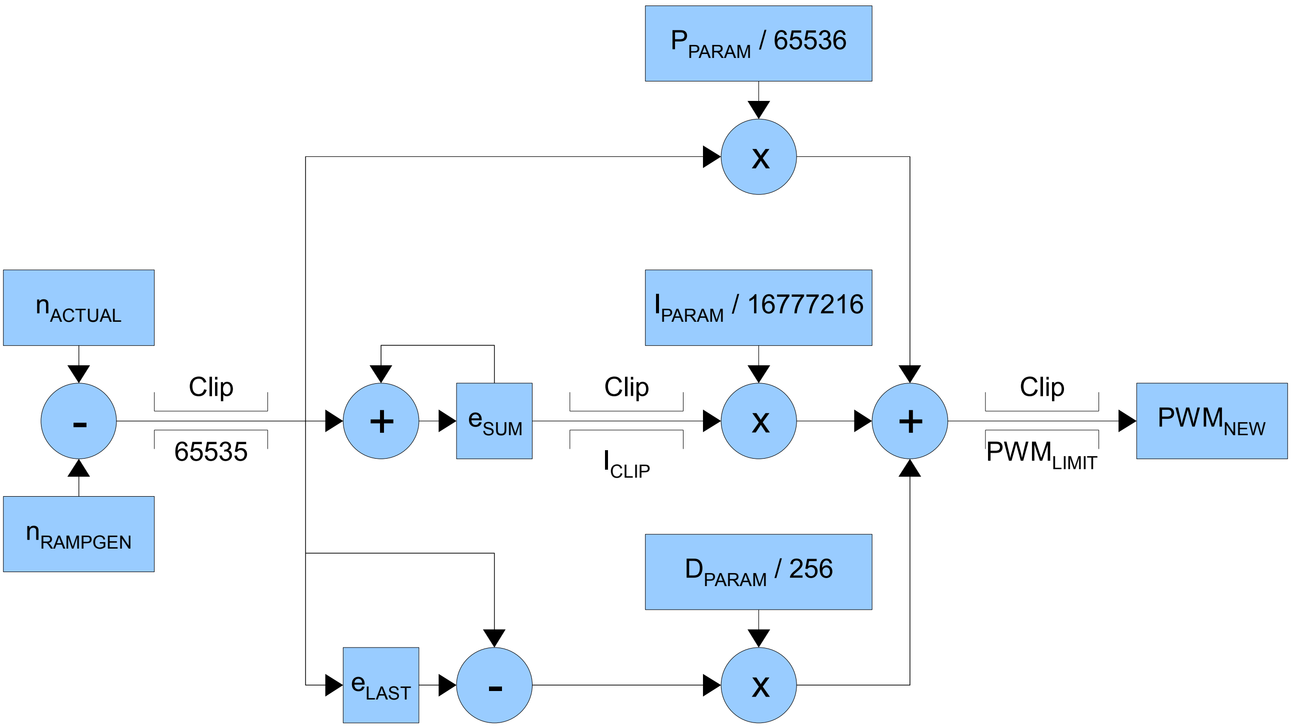


Figure 8: Positioning PID regulation

Parameter	Description
n_{ACTUAL}	Actual motor position
$n_{RAMPGEN}$	Target position of ramp generator
e_{LAST}	Error value of the last PID calculation
e_{SUM}	Error sum for integral calculation
P_{PARAM}	Position P-Parameter
I_{PARAM}	Position I-Parameter
D_{PARAM}	Position D-Parameter
I_{CLIP}	Position I-Clipping-Parameter
PWM_{LIMIT}	PWM Limit
PWM_{NEW}	New PWM value

Table 4: Position PID parameter description

After that, do the same for the 'Position I-Parameter' with the 'Position D-Parameter' still set to zero. For the 'Position I-Parameter', there is also a clipping value. The 'Position I-Clipping Parameter' should be set to a relatively low value to avoid overshooting, but high enough to allow exact position maintenance at stand still, when maximum required stand still torque is applied to the motor axis.

Now, modify the 'Position D-Parameter' in the same way. It will dampen part of the oscillations caused by the other parameters, too. An additional parameter is the 'PWM Hysteresis'. This value can be adapted to give lowest oscillation of the motor at stand still. It helps to compensate the motor's detent torque by driving the PID regulator near the motor reaction threshold even for lowest PID differences.

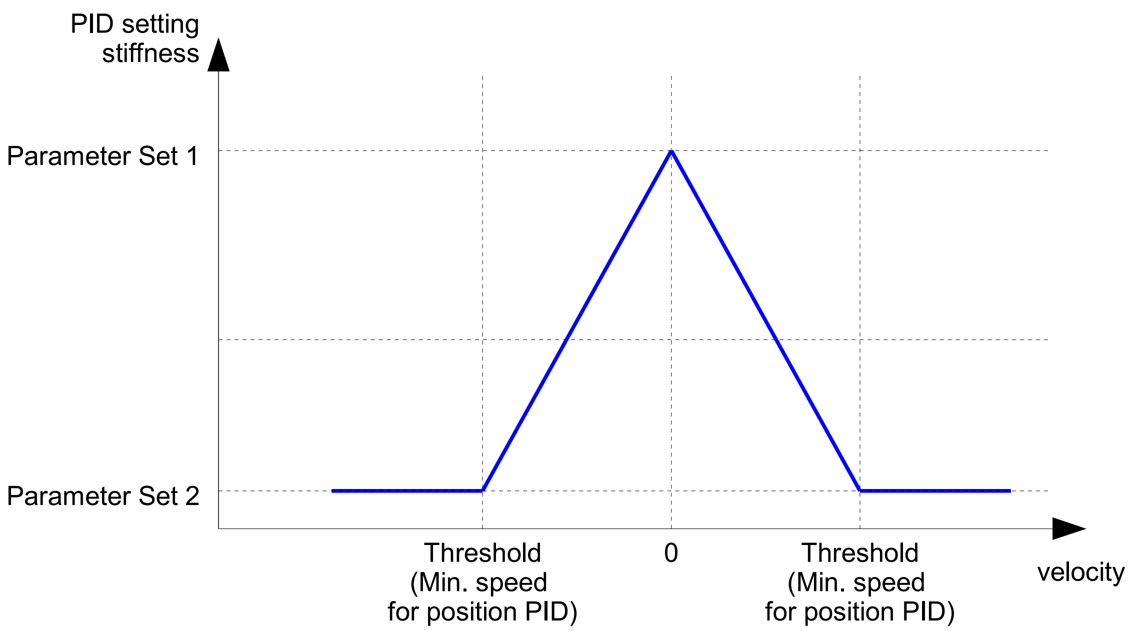


Figure 9: Positioning PID parameter sets

The PID regulation provides two parameter sets. These PID parameter sets are used as follows. Below a specified velocity threshold the motor position is regulated by a combination of parameter set 1 and parameter set 2 to get the best performance. So the switch over between both parameter sets is soft. Above the velocity threshold the motor position is only regulated by the parameter set 2. If the velocity threshold is set to zero, the parameter set 2 is used all the time. The velocity threshold is given by 'Min. speed for position PID' parameter.

For the first tests, you can set both PID parameter sets equal. After having found suitable values, the second parameter set (PID Parameter Set 2) should be set to lower values, to minimize regulation oscillation during constant rotation.

For positioning regulation, an integrated ramp generator with linear ramp control is used. Parameters for the ramp generator are the 'Max. positioning velocity' as well as the 'Acceleration' value. The ramp generator automatically calculates the slow down point, i.e. the point at which velocity is to be reduced in order to stop at the programmed target position. This calculation is based on the 'Max. positioning velocity' and the 'Acceleration' setting.

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

Depending upon the motor and mechanics respectively, a bit of oscillation is normal, in the best case it can be reduced to be at least +/-1 encoder step, because otherwise the regulation can not keep the position.

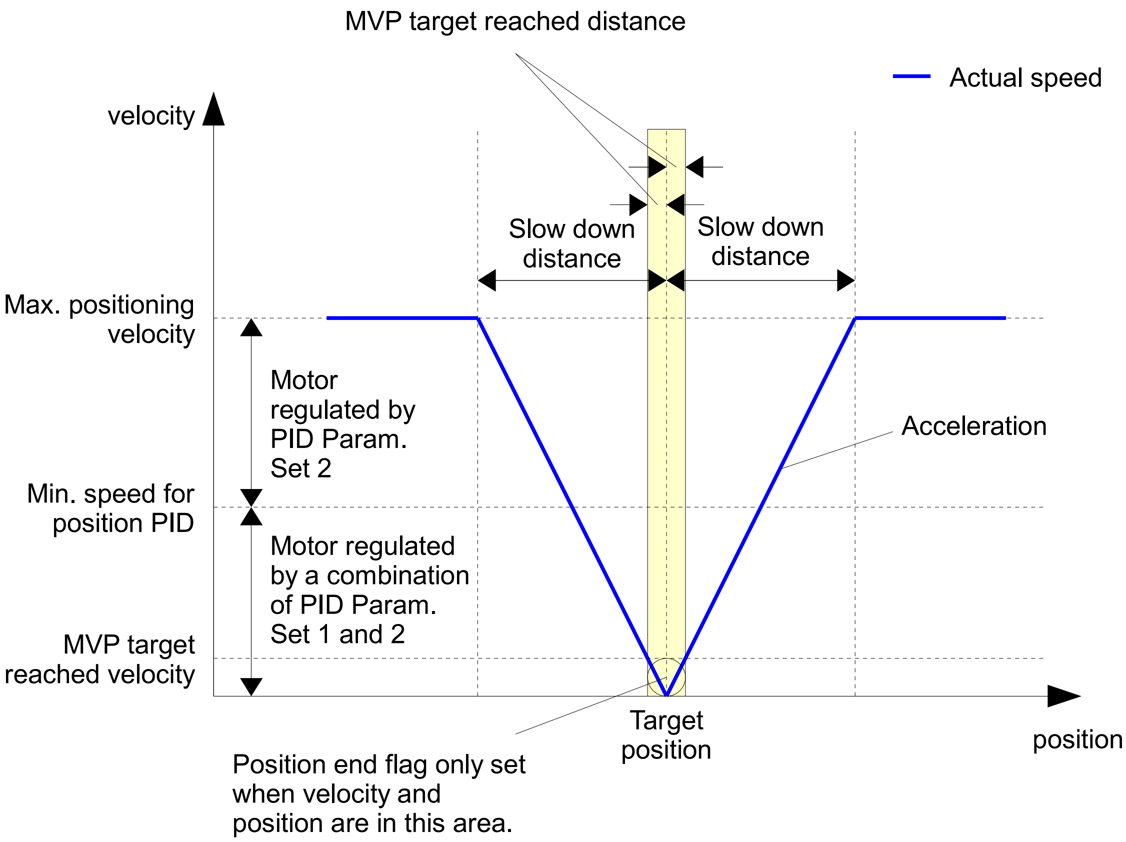


Figure 10: Positioning algorithm

In order to dampen oscillations as much as possible, also try using the 'PWM hysteresis' parameter. Especially with a friction in the mechanics, increasing this parameter might help to overcome it and to speed up the regulation.

Attention: 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 goes to current limitation, the unit may reset or undetermined regulation results may occur.

4 Quickstart

4.1 Connecting the motor

For a first quick start, plug the jumpers on the TMC603 evaluation board to their default settings (Table 6). Connect a three phase BLDC motor to the motor coil output (X502). The pins are marked by 'U', 'V' and 'W' (Figure 11). Connect the hall sensors to the connector X505, to run the BLDC motor by using hall sensors. The pins are marked by '+5V', 'GND', 'H1', 'H2' and 'H3'.

Attention: Please note that the BLDC motors may only be connected to or disconnected from the evaluation board while the board is disconnected from power supply, as otherwise the evaluation board could get damaged!

4.2 Connecting the RS232 interface and power supply

Connect the RS232 interface (X301) to the PC by using a null modem cable. The baud rate is automatically set to 9600 baud by the PC software. Alternatively, the TMC603 evaluation board allows a connection via USB interface (X300).

Now, connect the power supply between 12V and 48V and at least 1A to the connector X105 (power supply). The positive terminal is marked '12V-48V'. The green "POWER" LED will light up after attaching the power supply.

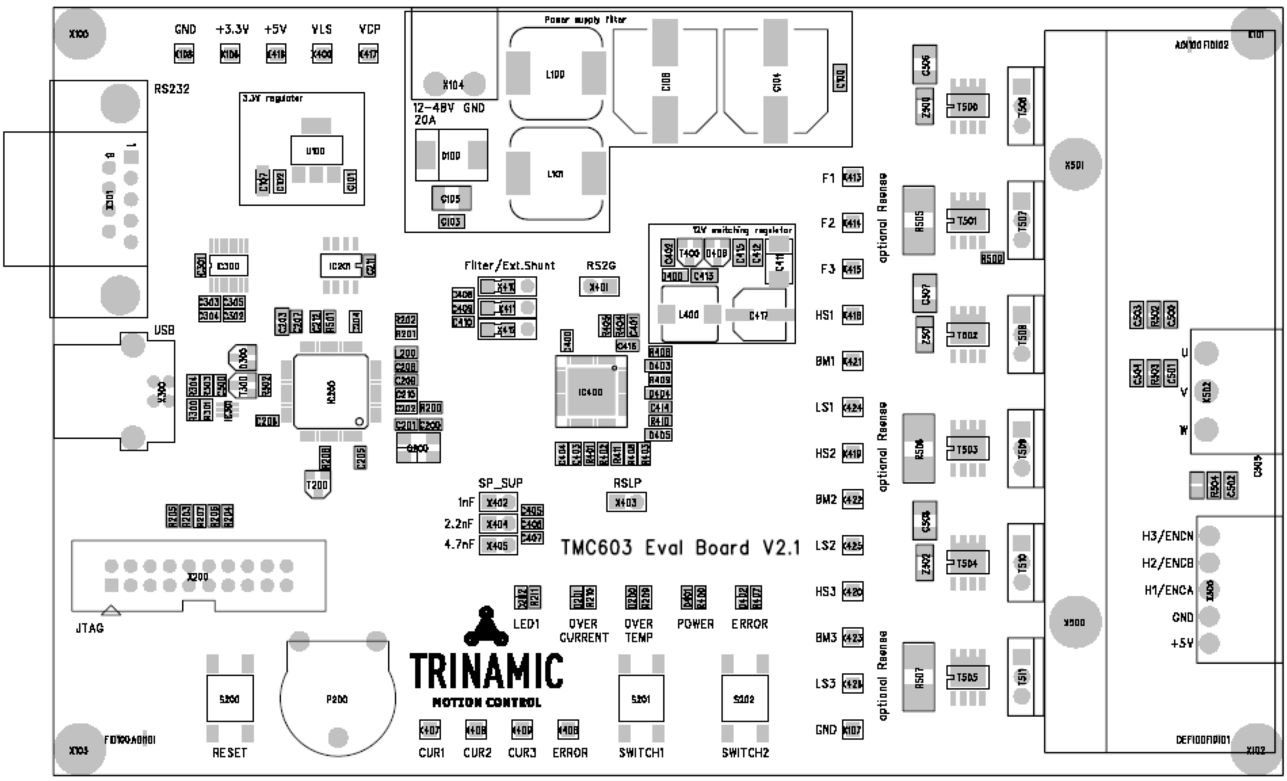


Figure 11: Connectors of TMC603 evaluation board

4.3 Establishing the connection

Start the TMC603 evaluation PC software on your PC and select the associated COM port that is to be used. If you press the 'Connect' button the connection will established. The connection establishment was successful if the default settings are read automatically.

4.4 Selecting the commutation mode

On tab 'Basic Operation' select the commutation mode. Several commutation modes are available. In commutation mode 'Block Comm. (Hall Sensors)' the motor will run by using hall sensors. In commutation mode 'Block Comm. (sensorless)' the motor will run by using the BLDC back EMF commutation hallFX™. Please keep in mind to press the 'Apply' button. You can only change the communication mode when the motor is stopped.

4.5 Configuring the motor parameters and hall sensor settings

All motor parameters and hall sensor settings can be changed on tab 'Settings'. Please set the number of motor poles and the resistance of motor coils. Optionally you can change the hall sensor polarity for BLDC motors with inverted hall scheme. Please keep in mind to press the 'Apply' button.

4.6 Configuring the current and velocity regulation settings

The current regulation settings can be changed on tab 'Current Regulation'. There are two different PID parameter sets. The first parameter set (PID Parameter Set 1) is used at lower velocity and the second parameter set (PID Parameter Set 2) is used at higher velocity. The parameter 'Threshold' adjusts the velocity limit in [rpm] to switch between first and second PID parameter set. If the threshold is set to zero, the second PID parameter set is only used. The parameter 'Max. Current' sets the maximum motor current (RMS) in [mA]. Please keep in mind to press the 'Apply' button.

The velocity regulation settings can be changed on tab 'Velocity Regulation'. There are also two different PID parameter sets. The first parameter set (PID Parameter Set 1) is used at lower velocity and the second parameter set (PID Parameter Set 2) is used at higher velocity. The parameter 'Threshold' adjusts the velocity limit in [rpm] to switch between first and second PID parameter set. If the threshold is set to zero, the second PID parameter set is only used. Please keep in mind to press the 'Apply' button.

4.7 Starting the motor

Go back to tab 'Basic Operation'. If commutation mode 'Block Comm. (Hall Sensors)' is selected, set the parameter 'Velocity' in [rpm] and start the motor by pressing the 'Left' or 'Right' button. If the 'Stop' button is pressed the motor will stop. In order to change the motor velocity, enter a new value and press again the 'Left' and 'Right' button respectively. The parameter 'Acceleration' sets the acceleration of motor in [rpm/s].

If commutation mode 'Block Comm. (sensorless)' is selected, at first set the parameter 'Start Current' in [mA]. This parameter defines the maximum current for startup (controlled block commutation). The parameter 'Threshold' adjusts the velocity limit in [rpm] to switch from controlled block commutation to sensorless commutation hallFX™. Thereafter, set the parameter 'Velocity' in [rpm] and start the motor by pressing the 'Left' or 'Right' button. The motor will ramp up the velocity as soon as the adjusted start current is reached. If the velocity threshold is reached, the firmware switches from controlled block commutation to sensorless commutation hallFX™ automatically. If the ramp up is failed, please check you current and velocity regulation settings.

Please take into account that in commutation mode 'Block Comm. (sensorless)' the maximum PWM duty cycle is limited to 90%. This is necessary because the BEMF voltage is incapable of measurement clearly above the PWM duty cycle of 90%.

5 Hardware

5.1 Board size 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 four mounting holes (hole diameter: 3.2mm) suitable for M3 screws.

5.2 Connectors

The TMC603 evaluation board is equipped with the following connectors:

Connector	Pin	Label	Description
Power supply	1	12-48V	Power supply input, +12V to +48V DC
	2	GND	Power supply and signal ground
Motor coil output	1	W	Motor coil connection W
	2	V	Motor coil connection V
	3	U	Motor coil connection U
Hall sensor / encoder	1	+5V	Power supply input for hall sensor or encoder, nom. +5V DC
	2	GND	Power supply and signal ground
	3	H1/ENCA	Hall sensor 1 or encoder signal A
	4	H2/ENCB	Hall sensor 2 or encoder signal B
	5	H3/ENCN	Hall sensor 3 or encoder signal N
USB			Standard USB type B
RS232	2		Received data line
	3		Transmitted data line
	5		RS23 signal and system ground

Table 5: Connectors of TMC603 evaluation board

Attention: Please note that the BLDC motors may only be connected to or disconnected from the evaluation board while the board is disconnected from power supply, as otherwise the evaluation board could get damaged!

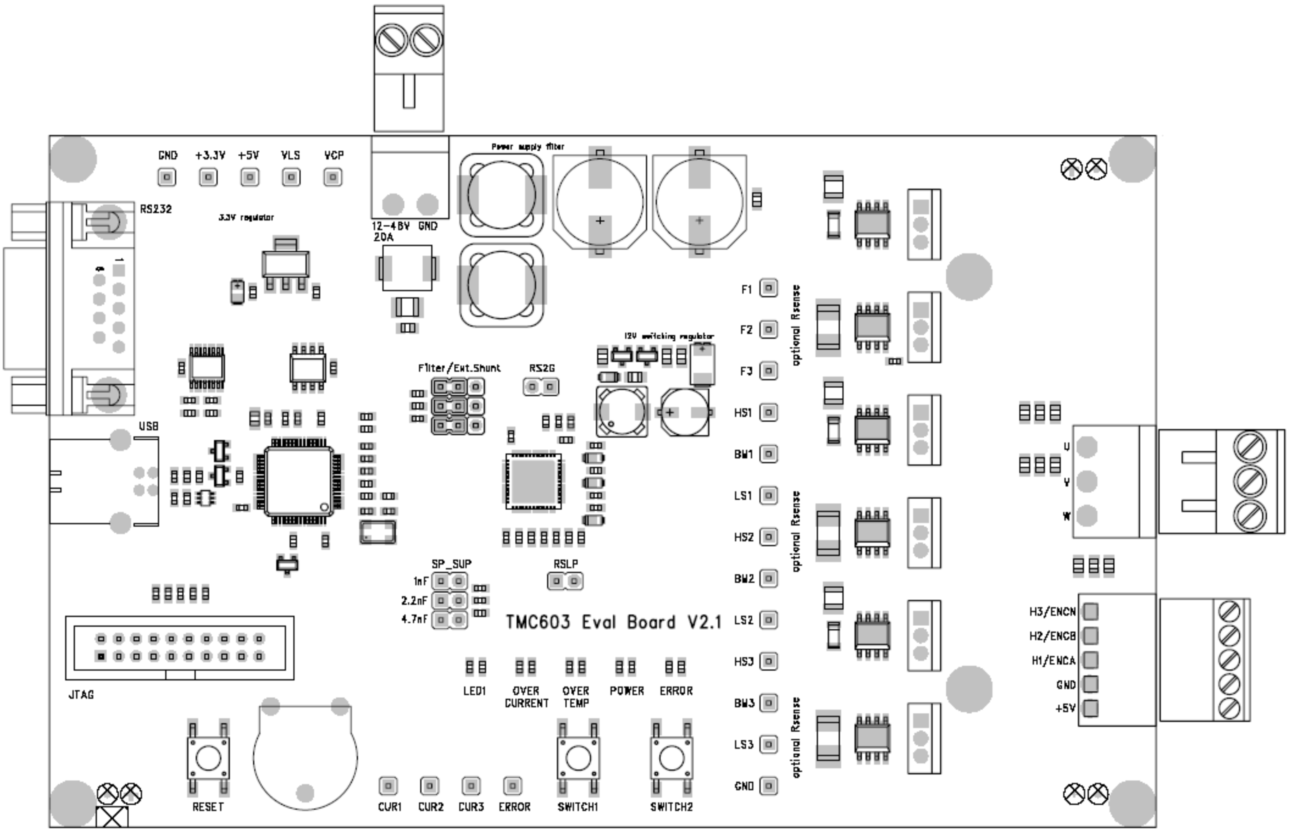


Figure 12: Connectors of TMC603 evaluation board

5.3 Jumpers

The TMC603 evaluation board is equipped with the following jumpers:

Jumper	Label	Description
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 unplugged: R_{S2G} = 470kΩ (t_{S2G} = 2µs) (default) Jumper plugged: R_{S2G} = 1.47MΩ (t_{S2G} = 6µs)
Slope control resistor	RSLP	The slope control resistor sets output current for MOSFET drivers. Jumper unplugged: R_{SLP} = 100kΩ (I_{gate} = 100mA) (default) Jumper plugged: R_{SLP} = 147kΩ (I_{gate} = 68mA)
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 filter or input for external sense resistor (default).

Table 6: Jumpers of TMC603 evaluation board

5.4 Status Signals

The TMC603 evaluation board provides five status LEDs.

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 lights up upon the power stage has exceeded a critical temperature of 125°C. The motor becomes switched off.
Current limit	OVER CURRENT	This red LED lights up upon the motor current has exceeded the current limit.

Table 7: Status signals of TMC603 evaluation board

5.5 Measuring points

The TMC603 evaluation board is equipped with the following measuring points:

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
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	CURx	Output of current measurement amplifier (for phase 1 to 3). The output signal is centered to 1.65V.
Filter output	Fx	Output of internal switched capacitor filter (for phase 1 to 3). This signal is available only if filter output (Filter/Ext.Shunt) is selected.
High side output	HSx	High side MOSFET driver output (for phase 1 to 3)
Bridge output	BMx	Sensing input for bridge output. Used for MOSFET control and current measurement. (for phase 1 to 3)
Low side output	LSx	Low side MOSFET driver output (for phase 1 to 3)

Table 8: Measuring points of TMC603 evaluation board

6 PC-Software

PC software to operate the TMC603 evaluation board is supplied on the TMC TechLibCD or can be downloaded at http://www.trinamic.com. The program can be used with Windows 98, Windows 2000 and Windows XP. To install it, just copy the file "TMC603EVAL.EXE" to the hard disk of your PC. To run the program, double click the file.

6.1 Overview

After starting the software, the main window will be shown (Figure 13). The PC software is divided into several parts.

- Inferface
- Module Type
- Info
- Measured Values
- Status Flags
- Error Flags
- Basic Operation
- Settings
- Current Regulation
- Velocity Regulation
- Positioning Regulation

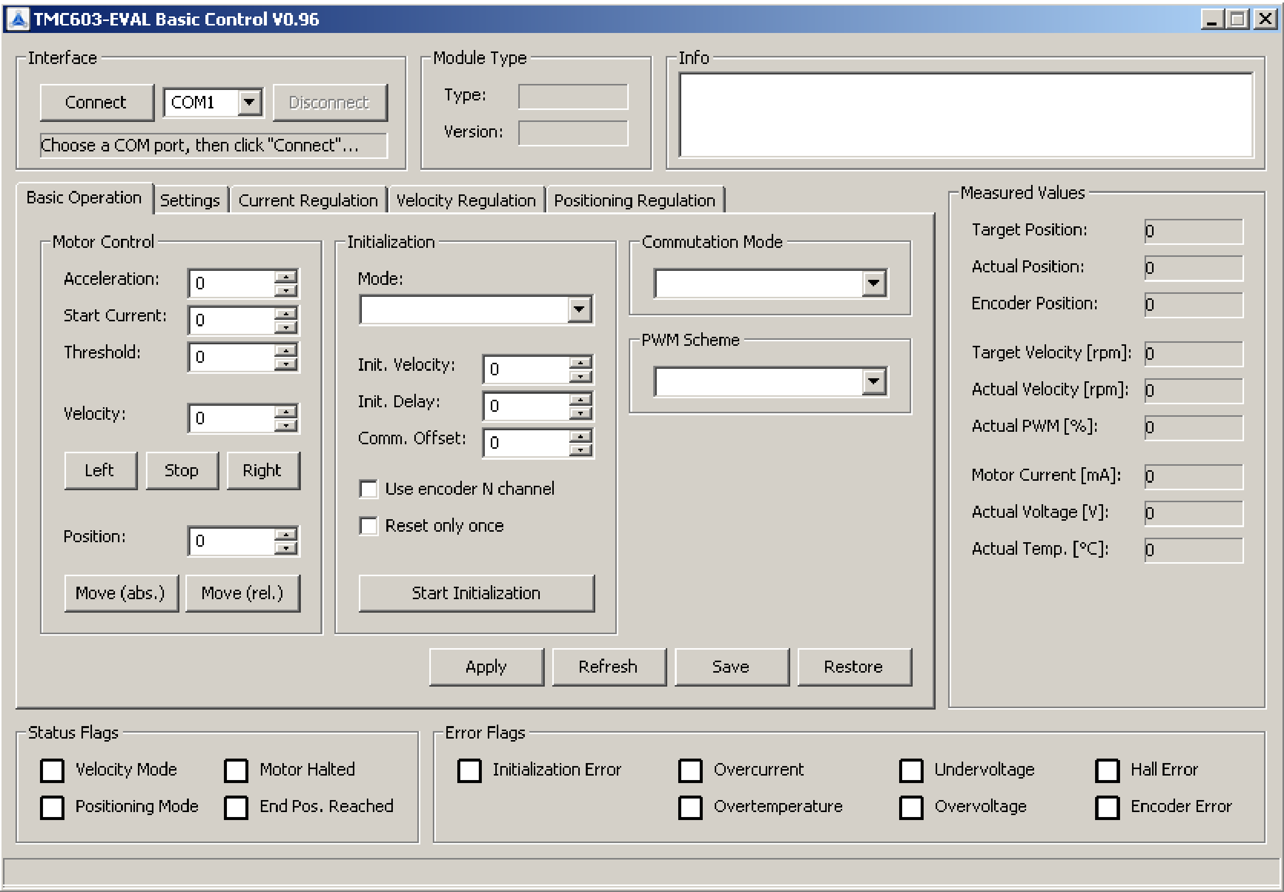


Figure 13: TMC603 evaluation PC software

6.2 Interface

The TMC603 evaluation board allows a communication to the PC via RS232 interface and via USB interface. If you use the RS232 interface, you need a null modem cable to connect the TMC603 evaluation board to your PC. After connecting the evaluation board to your PC, make sure that the evaluation board is supplied with power. Then, start the PC software by double clicking the file 'TMC603EVAL.EXE' and select the associated COM port that is to be used. The baud rate is automatically set to 9600 baud by the PC software. Press the 'Connect' button to establish the connection. All default parameters may be read automatically. Press the 'Disconnect' button to close the connection.

6.3 Module Type

After establishing the connection, the module type and the firmware version are read. You can find this information in the 'Module Type' group.

Parameter	Description
Type	Module type name 'TMC603-EVAL'
Version	Firmware version

Table 9: Module Type

6.4 Info

For all parameters you can find a short description in the info box. Put your cursor on the desired input box and the info box shows a short description of this parameter.

6.5 Measured Values

The 'Measured Values' group shows some information (Table 10) about the actual motor position, actual velocity, actual motor current, supply voltage and actual motor temperature. In the following table the column 'Type' shows the type number of the corresponding axis parameter.

Parameter	Description	Type
Target Position	The target position of a currently executed ramp.	0
Actual Position	The actual position of the BLDC motor.	1
Encoder Position	Actual encoder position / counter value.	209
Target Velocity [rpm]	Programmed target velocity in [rpm].	2
Actual Velocity [rpm]	Actual velocity of the motor in [rpm].	3
Actual PWM [%]	Actual PWM duty cycle (0% … 100%)	153
Motor Current [mA]	Actual motor current in [mA].	150
Actual Voltage [V]	Actual supply voltage in [V].	151
Actual Temp. [°C]	Actual motor temperature in [°C].	152

Table 10: Measured Values

6.6 Status Flags

The status flags give some information about the actual state of TMC603 evaluation board. The following status flags are available:

Parameter	Description
Velocity Mode	This flag is set if the module is in velocity mode.
Positioning Mode	This flag is set if the module is in positioning mode.
Motor Halted	This flag is set if motor has been switched off.
End Pos. Reached	This flag becomes set if the motor has been stopped at the end position

Table 11: Status Flags

The 'End Pos. Reached' flag (end position reached) informs the user, that the motor has reached the specified target position. This flag becomes set, whenever the following conditions are fulfilled at the same time:

- the distance is within range as defined by parameter 'Targ. Rea. Dis.' (target reached distance)
- the sum of target velocity and actual velocity is within range as defined by parameter 'Targ. Rea. Vel.' (target reached velocity)

The 'End Pos. Reached' flag becomes reset by the following user actions:

- 'Left' or 'Right' button is pressed
- 'Stop' button is pressed
- new positioning value is entered
- setting of the actual position

The 'Motor Halted' flag becomes set, when the motor is stopped by switching off the drivers. Switching off the drivers makes the motor powerless an it is easy to turn by hand. The motor additionally stops to regulate to the actual target position. This flag becomes set upon the following conditions:

- Undervoltage
- Overvoltage (if overvoltage detection is enabled)
- Overtemperature
- 'Stop' button is pressed

As long as any one or multiple of these conditions become true, the target velocity becomes set to zero and the motor is being switched off. The 'Motor Halted' flag becomes reset, as soon as all conditions triggering a halt have been cleared.

6.7 Error Flags

The error flags give some information about the error state of TMC603 evaluation board. The following error flags are available:

Parameter	Description
Initialization Error	This flag is set if initialization is failed.
Overcurrent	This flag is set if overcurrent limit is exceeded.
Overtemperature	This flag is set if overtemperature limit is exceeded.
Undervoltage	This flag is set if supply voltage to low for motor operation
Overvoltage	This flag is set if the motor becomes switched off due to overvoltage.
Hall Error	This flag is set upon a hall error.
Encoder Error	This flag is set upon an encoder error.

Table 12: Error Flags

The hall error and encoder error flags become automatically reset upon readout.

6.8 Basic Operation

The tab 'Basic Operation' provides all basic parameters to control a BLDC motor. The parameters are divided into several groups. The group 'Motor Control' allows to control the BLDC motor by using the velocity mode or the positioning mode. In velocity mode, the motor speed is regulated by using a PID regulation (refer chapter 3.3). In positioning mode, the motor position is regulated by using also a PID regulation (refer chapter 3.4). All these parameters are described in table 13. The column 'Type' shows the type number of the corresponding axis parameter.

Parameter		Description	Type
Motor Control			
	Acceleration	Acceleration value in [RPM/s]. This parameter is used for velocity and positioning mode	11
	Start Current	Motor current for controlled commutation in [mA]. This parameter is used in commutation mode 01, 04, 05 and in initialization of sine.	177
	Threshold	The velocity limit in [rpm] to switch from controlled block commutation to sensorless commutation hallFX™.	4
	Velocity	Target velocity in [rpm]. When 'Disable PID' is selected, the PWM duty cycle in [%] can be directly entered.	-
	Left / Right	This button starts the rotation in 'Left'/'Right' direction.	-
	Stop	This button stops the motor.	-
	Position	Target position of motor.	-
	Move (abs.)	This button starts the movement towards the specified absolute target position, with automatic generation of acceleration and deceleration ramps.	-
	Move (rel.)	This button starts the movement towards the specified relative target position by means of an offset to the last ramp generator position. In those case, the resulting new position value must not exceed the above mentioned limits, too.	-

Table 13: Motor Control

The group 'Initialization' contains all parameters for initialization of sine commutation. The parameters are described in the following table (Table 14). The column 'Type' shows the type number of the corresponding axis parameter.

Parameter		Description	Type
Initialization			
	Mode	00 Motor controlled (First Init.): The motor rotates in controlled sine commutation. At this time the commutation offset is calculated and stored in the EEPROM. 01 Motor controlled (Reinit.): The motor rotates in controlled sine commutation. At this time the N channel event is detected. The commutation offset is not calculated. Rather the commutation offset is restored from EEPROM.	249
	Init. Velocity	Velocity in [rpm] for sine initialization.	241
	Init. Delay	Duration for sine initialization sequence in [ms]. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	244
	Comm. Offset	This value represents the internal commutation offset. (0 … max. encoder steps per rotation)	165
	Use encoder N channel	If this parameter is set, the encoder counter will be set to zero at next N channel event.	161
	Reset only once	If this parameter is set, the encoder counter will be set to zero only once. Otherwise the encoder counter will be always set to zero at a N channel event.	163
	Start Initialization	This button starts the sine initialization.	160

Table 14: Sine Initialization

The TMC603 evaluation board supports currently six different commutation modes. In this way the BLDC motor can be rotated either in block commutation or in sine commutation by using hall sensors or encoder. Additionally, a controlled block commutation and sine commutation is available without any feedback about the rotor position. The following table (Table 15) shows a list of commutation modes. The column 'Type' shows the type number of the corresponding axis parameter.

Parameter	Description	Type
Commutation Mode	00 Block Comm. (Hall Sensors): In this mode the BLDC motor rotates in block commutation by using hall sensors. 01 Block Comm. (sensorless): In this mode the motor starts in controlled block commutation. As soon as the desired velocity threshold is reached, the firmware switches automatically to sensorless block commutation. Than, the motor rotates by using the back EMF commutation hallFX™. 02 Sine Comm. (Hall Sensors): In this mode the BLDC motor rotates in sine commutation by using hall sensors. 03 Sine Comm. (Encoder): In this mode the BLDC motor rotates in sine commutation by using encoder. 04 Block Comm. (controlled): In this mode the BLDC motor rotates in controlled block commutation. 05 Sine Comm. (controlled): In this mode the BLDC motor rotates in controlled sine commutation.	159

Table 15: Commutation Mode

When the commutation mode 'Block Comm. (Hall Sensors)' is used, an additional option is available. This parameter is called 'PWM Scheme'. By using this parameter the commutation chopper mode can be selected. Three different chopper modes are available. These modes are shown in the following table (Table 16). The column 'Type' shows the type number of the corresponding axis parameter.

Parameter	Description	Type
PWM Scheme	00 High Side Chopper: In this chopper mode the high side drivers are chopped only. The low side drivers are in on-state. 01 Low Side Chopper: In this chopper mode the low side drivers are chopped only. The high side drivers are in on-state. 02 Dual Side Chopper: Both, the high side drivers and low side drivers are chopped with the same signal. The dual side chopper mode is required for sensorless commutation hallFX™.	167

Table 16: PWM Scheme

6.9 Settings

The tab 'Settings' provides all general parameters of the associated motor, hall sensors and encoder. The parameters are divided into several groups. The group 'Motor Parameters' allows to change all motor-specific settings like number of motor poles, resistance of motor coils and so on (Table 17).

The groups 'Hallsensor Parameters' and 'Encoder Parameters' provide several settings for hall sensor and encoder that is to be used. These parameters are described in table 18 and 19. Table 20 shows additional options for overvoltage protection and feed forward control. The column 'Type' shows the type number of the corresponding axis parameter.

Parameter		Description	Type
Motor Parameters			
	Motor Poles	Number of motor poles. Used for velocity measurement and regulation.	253
	BEMF Const.	BEMF constant of motor in [rpm/(10V)]. Used for feedforward control in current regulation, positioning regulation and velocity regulation.	239
	Coil Resistance	Resistance of motor coil in [$m\Omega$]. Used for current regulation, positioning regulation and velocity regulation.	240
	Gear Ratio	Transmission ratio of gear, e.g. gear ratio of 6 for transmission ratio 6:1. Used for positioning regulation.	211

Table 17: Motor Parameters

Parameter		Description	Type
Hallsensor Parameters			
	Commutation Offset CW	This parameter helps to tune hall sensor based initialization in a way, that the motor has the same velocity for left and right turn. It compensates for tolerance and hysteresis of the hall sensors. It is added to the commutation angle upon CW turn. This parameter is used in commutation mode 02.	242
	Commutation Offset CCW	This parameter helps to tune hall sensor based initialization in a way, that the motor has the same velocity for left and right turn. It compensates for tolerance and hysteresis of the hall sensors. It is added to the commutation angle upon CCW turn. This parameter is used in commutation mode 02.	243
	Reverse Hall Sensor Polarity	Set this parameter, if the hall sensor polarity is reversed. Sets one of the motors with inverted hall scheme, e.g. some Maxon motors.	254

Table 18: Hallsensor Parameters

Parameter		Description	Type
Encoder Parameters			
	Actual Position	Change the position counter without moving the motor.	1
	Encoder Steps	Encoder steps per rotation.	250
	Reverse Encoder Direction	Set this parameter in a way, that turn right increase position counter.	251

Table 19: Encoder Parameters

Parameter		Description	Type
Other Options			
	Overvoltage Protection	Enables overvoltage protection.	245
	Feed Forward Compensation	Enables feedforward control. If feedforward control is selected, use the right BEMF constant of BLDC motor.	-

Table 20: Other Options

6.10 Current Regulation

The tab 'Current Regulation' provides all parameters for current regulation and current limitation respectively. The group 'General Options' shows all general settings for current regulation. These parameters are described in table 21. The column 'Type' shows the type number of the corresponding axis parameter.

The current PID regulation provides two different parameter sets. These PID parameter sets are used as follows. Below a specified velocity threshold the motor current is regulated by a combination of parameter set 1 and parameter set 2 to get the best performance. Above the velocity threshold the motor current is only regulated by the parameter set 2. If the velocity threshold is set to zero, the parameter set 2 is used all the time. The two parameter sets can be find in group 'PID Parameter Set 1'. and 'PID Parameter Set 2'. The parameters are described in table 22 and 23. The column 'Type' shows the type number of the corresponding axis parameter. A detailed description about the PID current regulation can be find in chapter 3.2.

Parameter		Description	Type
General Options			
	Calc. Delay	Delay of current limitation / PID current regulation in [50µs].	134
	Threshold	Adjust the velocity limit in [rpm] to switch between first current PID parameter set and second current PID parameter set.	176
	PWM Change	Maximum PWM change per PID interval.	246
	Max. Current	Maximum motor current in [mA].	6

Table 21: General Current Regulation Options

Parameter		Description	Type
PID Parameter Set 1		This PID parameter set is used at lower velocity. The velocity limit is adjustable by the 'Threshold' parameter.	
	P Parameter	P-Parameter of PID current regulation.	168
	I Parameter	I-Parameter of PID current regulation.	169
	D Parameter	D-Parameter of PID current regulation.	170
	I Clipping:	I-Clipping-Parameter of PID current regulation	171

Table 22: Current PID Parameter Set 1

Parameter		Description	Type
PID Parameter Set 2		This PID parameter set is used at higher velocity. The velocity limit is adjustable by the 'Threshold' parameter.	
	P Parameter	P-Parameter of PID current regulation.	172
	I Parameter	I-Parameter of PID current regulation.	173
	D Parameter	D-Parameter of PID current regulation.	174
	I Clipping:	I-Clipping-Parameter of PID current regulation	175

Table 23: Current PID Parameter Set2

6.11 Velocity Regulation

The tab 'Velocity Regulation' provides all parameters for velocity control. The group 'General Options' shows all general settings such as calculation delay and PWM limit.

The option 'Disable PID' disables the velocity regulation. In this case the motor PWM duty cycle can be directly entered (refer 7.4). When the velocity regulation is enabled, two different control algorithms can be selected. The first control algorithm is a standard PID regulation. The second control algorithm is an integrating PID regulation. This means that the result of the PID calculation is integrated to the PWM value. All these parameters are described in table 24. The column 'Type' shows the type number of the corresponding axis parameter.

The velocity PID regulation provides two parameter sets. These PID parameter sets are used as follows. Below a specified velocity threshold the motor speed is regulated by a combination of parameter set 1 and parameter set 2 to get the best performance. Above the velocity threshold the motor speed is only regulated by the parameter set 2. If the velocity threshold is set to zero, the parameter set 2 is used all the time. The two parameter sets can be find in group 'PID Parameter Set 1'. and 'PID Parameter Set 2'. The parameters are described in table 25 and 26. The column 'Type' shows the type number of the corresponding axis parameter.

The group 'PID Controller' shows the actual velocity error and the sums of errors of the velocity PID regulator (Table 27). A detailed description about the PID velocity regulation can be find in chapter 3.3.

Parameter		Description	Type
General Options			
	Calc. Delay	PID calculation delay in [ms].	133
	Threshold	Adjust the velocity limit in [rpm] to switch between first velocity PID parameter set and second velocity PID parameter set.	8
	PWM Change	Maximum PWM change per PID interval.	246
	Hysteresis	Compensates dead time of PWM and motor friction.	136
	PWM Limit	Set PWM limitation (0% … 100%)	5
	Disable PID	Disable the PID calculation. The motor PWM is then directly derived from the target velocity.	147
	Standard PID	Enable the standard PID calculation.	147
	PID integrated to PWM	Enable an integrating PID algorithm. In that case the result of the PID calculation is integrated to the PWM value. This PID regulation is easier to use than the standard PID regulation.	147

Table 24: General Velocity Regulation Options

Parameter		Description	Type
PID Parameter Set 1		This PID parameter set is used at lower velocity. The velocity limit is adjustable by the 'Threshold' parameter.	
	P Parameter	P-Parameter of PID velocity regulation.	140
	I Parameter	I-Parameter of PID velocity regulation.	141
	D Parameter	D-Parameter of PID velocity regulation.	142
	I Clipping	I-Clipping-Parameter of PID velocity regulation	143

Table 25: Velocity PID Parameter Set 1

Parameter		Description	Type
PID Parameter Set 2		This PID parameter set is used at higher velocity. The velocity limit is adjustable by the 'Threshold' parameter.	
	P Parameter	P-Parameter of PID velocity regulation.	234
	I Parameter	I-Parameter of PID velocity regulation.	235
	D Parameter	D-Parameter of PID velocity regulation.	236
	I Clipping	I-Clipping-Parameter of PID velocity regulation	237

Table 26: Velocity PID Parameter Set 2

Parameter		Description	Type
PID Controller			
	PID Vel. Error	Actual error of PID velocity regulation.	228
	PID Vel. Error Sum	Sums of errors of PID velocity regulation.	229

Table 27: Velocity PID Controller Outputs

6.12 Positioning Regulation

The tab 'Positioning Regulation' provides all parameters for positioning control. The group 'General Options' shows all general settings such as maximum positioning velocity. All these parameters and options are described in table 28. The column 'Type' shows the type number of the corresponding axis parameter.

The positioning PID regulation provides two parameter sets. These PID parameter sets are used as follows. Below a specified velocity threshold the motor speed is regulated by a combination of parameter set 1 and parameter set 2 to get the best performance. Above the velocity threshold the motor speed is only regulated by the parameter set 2. If the velocity threshold is set to zero, the parameter set 2 is used all the time. The two parameter sets can be find in group 'PID Parameter Set 1'. and 'PID Parameter Set 2'. The parameters are described in table 29 and 30. The column 'Type' shows the type number of the corresponding axis parameter.

The group 'PID Controller' shows the actual positioning error and the sums of errors of the positioning PID regulator (Table 31). A detailed description about the PID position regulation can be find in chapter 3.4.

Parameter		Description	Type
General Options			
	Calc. Delay	PID calculation delay in [ms].	133
	Threshold	Adjust the velocity limit in [rpm] to switch between first positioning PID parameter set and second positioning PID parameter set.	12
	PWM Change	Maximum PWM change per PID interval.	246
	Hysteresis	Compensates dead time of PWM and motor friction.	136
	Max. Pos. Vel.	The maximum positioning velocity in [rpm] used for positioning mode when executing a ramp to a position.	4
	Clr. Targ. Dis.	This parameter sets the clear target distance. The velocity is set to zero if actual position differs from motor position for more than this value, until the motor catches up. Prevents velocity overshoot if the motor can't follow the velocity ramp.	9
	Targ. Rea. Vel.	Target reached velocity is the maximum velocity in [rpm] at which the 'End Position Reached' flag can be set. Prevents issuing of end position when the target is passed at high velocity.	7
	Targ. Rea. Dis.	Target reached distance is the maximum distance at which the 'End Position Reached' flag can be set.	10
	Mass Inertia Const.	Mass inertial constant for positioning regulation. Compensates mass moment of inertia of rotor.	238
	PID opt. Clear I-Max.	Clears I-Sum if PWM reaches maximum value of 100%	137
	PID opt. Clear I-Sign.	Clears I-Sum if the position overshoots the target value.	138

Table 28: General Positioning Regulation Options

Parameter		Description	Type
PID Parameter Set 1		This PID parameter set is used at lower velocity. The velocity limit is adjustable by the 'Threshold' parameter.	
	P Parameter	P-Parameter of PID positioning regulation.	130
	I Parameter	I-Parameter of PID positioning regulation.	131
	D Parameter	D-Parameter of PID positioning regulation.	132
	I Clipping:	I-Clipping-Parameter of PID positioning regulation	135

Table 29: Positioning PID Parameter Set 1

Parameter		Description	Type
PID Parameter Set 2		This PID parameter set is used at higher velocity. The velocity limit is adjustable by the 'Threshold' parameter.	
	P Parameter	P-Parameter of PID positioning regulation.	230
	I Parameter	I-Parameter of PID positioning regulation.	231
	D Parameter	D-Parameter of PID positioning regulation.	232
	I Clipping:	I-Clipping-Parameter of PID positioning regulation	233

Table 30: Positioning PID Parameter Set 2

Parameter		Description	Type
PID Controller			
	PID Pos. Error	Actual error of PID positioning regulation.	226
	PID Pos. Error Sum	Sums of errors of PID positioning regulation.	227

Table 31: Positioning PID Controller Outputs

7 TMCL Communication Protocol

The TMC603 evaluation board supports TMCL direct mode (binary commands only, no ASCII interface) over RS232 or USB. The default baudrate is 9600 baud for RS232 communication.

In direct mode the TMCL communication in most cases follows a strict master / slave relationship. That is, the host computer acting as master will send a command to the TMC603 evaluation board. The TMCL interpreter on the evaluation board will then interpret this command. As soon as this step has been done, the evaluation board will send back a reply over RS232 to the master. Only then should the master transfer the next command.

The architecture of TMCL communication protocol are explained in the following chapters.

7.1 TMCL Basics

7.1.1 Binary Command Format

Every command has a mnemonic and a binary representation. When commands are sent from a host to the TMC603 evaluation board, the binary format has to be used. Every command consists of an one-byte address field, a one-byte command field, a one-byte type field, a one-byte motor/bank field, a four-byte value field and a one-byte check sum field.

So the binary command format when using RS232 is as follows:

Byte index	Number of bytes	Description
0	1	Module address
1	1	Command number
2	1	Type number
3	1	Motor or bank number
4...7	4	Value (MSB first)
8	1	Check sum

The check sum is calculated by adding up all other bytes using an 8-bit addition.

7.1.2 Binary Reply Format

Every time a command has been sent to a module, the module sends a reply. When using RS232 the format of the reply is as follows:

Byte index	Number of bytes	Description
0	1	Reply address
1	1	Module address
2	1	Status (e.g. 100 means 'no error')
3	1	Command number
4...7	4	Value (MSB first)
8	1	Check sum

The check sum is also calculated be adding up all the other bytes using an 8-bit addition. Attention: Do not send the next command before you have received the reply.

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

Byte index	Description
0	Reply address
1	Module address
2	Status (e.g. 100 means 'no error')
3	Command number
4...7	Value (MSB first)
8	Check sum

7.2 TMCL Command Overview

In this chapter an overview of the TMCL commands is given. All commands are described in chapter 7.3.

7.2.1 Motion commands

These commands controls the motion of the BLDC motor. They are the most important commands and can be used in direct mode.

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

7.2.2 Parameter commands

These commands are used to set, read and store axis parameters or global parameters. These commands can also be used in direct mode.

TMCL command	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 form EEPROM

7.3 TMCL Command Dictionary

This chapter describes all TMCL commands. The commands are sorted by their command numbers. For every command the mnemonic with its syntax and the binary representation are given. Ranges for the parameters are also given.

Some examples are given at the end of every command description. In the examples for the binary representation always RS232/USB communication (9 byte) format is used with module address 1 and host address 2. The motor number is always zero for a one axis module.

7.3.1 ROR Command

Description: This instruction starts rotation in “right” direction, i.e. increasing the position counter.

Internal function: First, velocity mode is selected. Then, the velocity value is transferred to axis parameter #0 ("target velocity").

Related commands: ROL, MST, SAP, GAP

Mnemonic: ROR <motor number>, <velocity>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
1	(don't care)	<motor number>	<velocity>

Reply in direct mode:

STATUS	VALUE
100 - OK	<velocity>

Value range: 0...100000, actual usable value depends on motor

Example:

Rotate right, motor #0, velocity = 350

Mnemonic: ROR 0, 350

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x01	0x00	0x00	0x00	0x00	0x01	0x5E	0x61

7.3.2 ROL Command

Description: This instruction starts rotation in “left” direction, i.e. decreasing the position counter.

Internal function: First, velocity mode is selected. Then, the velocity value is transferred to axis parameter #0 ("target velocity").

Related commands: ROR, MST, SAP, GAP

Mnemonic: ROL <motor number>, <velocity>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
2	(don't care)	<motor number>	<velocity>

Reply in direct mode:

STATUS	VALUE
100 - OK	<velocity>

Value range: 0...100000, actual usable value depends on motor

Example:

Rotate left, motor #0, velocity = 1200

Mnemonic: ROL 0, 1200

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x02	0x00	0x00	0x00	0x00	0x04	0xB0	0xB7

7.3.3 MST Command

Description: This instruction stops the motor by ramping down the velocity.

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

Related commands: ROR, ROL, SAP, GAP

Mnemonic: MST <motor number>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
3	(don't care)	<motor number>	(don't care)

Reply in direct mode:

STATUS	VALUE
100 - OK	<velocity>

Example:

Stop motor #0

Mnemonic: MST 0

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x03	0x00	0x00	0x00	0x00	0x00	0x00	0x04

7.3.4 MVP Command

Description: A movement towards the specified position is started, with automatic generation of acceleration- and deceleration ramps. The maximum velocity and acceleration are defined by axis parameter #4 and #11.

Two operation types are available:
Type 0: Moving to an absolute position in the range form -2^{31} to $+2^{31}$.
Type 1: Starting a relative movement by means of an offset to the last ramp generator position. In this case, the resulting new position value must not exceed the above mentioned limits, too.

Internal function: A new position value is transferred to the axis parameter #2 “target position”.

Related commands: MST, SAP, GAP

Mnemonic: MVP <ABS | REL>, <motor number>, <position | offset>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
4	0 ABS - absolute	<motor number>	<position>
	1 REL - relative	<motor number>	<offset>

Reply in direct mode:

STATUS	VALUE
100 - OK	(don't care)

Example:

Move motor #0 to (absolute) position 90000

Mnemonic: MVP ABS, 0, 90000

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x04	0x00	0x00	0x00	0x01	0x5F	0x90	0xF5

Move motor #0 from current position 1000 steps backward (move relative -1000)

Mnemonic: MVP ABS, 0, -1000

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x04	0x01	0x00	0xFF	0xFF	0xFC	0x18	0x18

Note: Before using the MVP command, one has to set MVP specific parameters (please refer command SAP (set axis parameter), type 4, 8-12).

7.3.5 SAP Command

Description: Most parameter of the TMC603 evaluation board can be adjusted individually. See chapter 7.4 for a complete list of all axis parameters.

Internal function: The parameter format is converted ignoring leading zeros (or ones for negative values). The parameter is transferred to the correct position in the appropriate device.

Related commands: GAP, STAP, RSAP, AAP

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

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
5	<parameter number>	<motor number>	<value>

Reply in direct mode:

STATUS	VALUE
100 - OK	(don't care)

Example:

Set the maximum current of motor #0 to 200mA

Mnemonic: SAP 6, 1, 200

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x05	0x06	0x00	0x00	0x00	0x00	0xC8	0xD4

7.3.6 GAP Command

Description: Most parameter of the TMC603 evaluation board can be adjusted individually. They all can be read by this function. See chapter 7.4 for a complete list of all axis parameters.

Internal function: The parameter is read out of the correct position in the appropriate device. The parameter format is converted adding leading zeros (or ones for negative values).

Related commands: SAP, STAP, RSAP, AAP

Mnemonic: GAP <parameter number>, <motor number>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
6	<parameter number>	<motor number>	(don't care)

Reply in direct mode:

STATUS	VALUE
100 - OK	<value>

Example:

Get the actual position of motor #0

Mnemonic: GAP 2, 1

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x06	0x01	0x00	0x00	0x00	0x00	0x00	0x08

Reply:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Host address	Target. address	Status	Instruct. number	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x02	0x01	0x64	0x06	0x00	0x00	0x02	0xC7	0x36

STATUS	VALUE
100 – OK	711

7.3.7 STAP Command

Description: Axis parameters are located in RAM memory, so modifications are lost at power down. This instruction enables permanent storing. Most parameters are automatically restored after power up (see axis parameter list in chapter 7.4).

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

Related commands: SAP, GAP, RSAP, AAP

Mnemonic: STAP<parameter number>, <motor number>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
7	<parameter number>	<motor number>	(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	VALUE
100 - OK	(don't care)

Example:

Store the maximum current of motor #0

Mnemonic: STAP 6, 1

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x07	0x06	0x00	0x00	0x00	0x00	0x00	0x0E

Note: The STAP command will not have any effect when the configuration EEPROM is locked.

7.3.8 RSAP Command

Description: For all configuration related axis parameters, non-volatile memory locations are provided. By default, mast parameters are automatically restored after power up (see axis parameter list in chapter 7.4). A single parameter that has been changed before can be reset by this instruction.

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

Related commands: SAP, GAP, STAP, AAP

Mnemonic: RSAP<parameter number>, <motor number>

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
8	<parameter number>	<motor number>	(don't care)

Reply in direct mode:

STATUS	VALUE
100 - OK	(don't care)

Example:

Restore the maximum current of motor #0

Mnemonic: RTAP 6, 1

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x08	0x06	0x00	0x00	0x00	0x00	0x00	0x0F

7.3.9 SGP Command

Description: Global parameters are related to the host interface, peripherals or application specific variables. The different groups of these parameters are organized in 'banks' to allow a larger total number for future products. Currently, only bank 0 and 1 are used for global parameters. See chapter 7.5 for a complete parameter list.

Internal function: The parameter format is converted ignoring leading zeros (or ones for negative values). The parameter is transferred to the correct position in the appropriate device.

Related commands: GGP, STGP, RSGP, AGP

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

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
9	<parameter number>	<bank number>	<value>

Reply in direct mode:

STATUS	VALUE
100 - OK	(don't care)

Example:

Set the serial address of the target device to 3

Mnemonic: SGP 66, 0, 3

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x09	0x42	0x00	0x00	0x00	0x00	0x03	0x4F

7.3.10 GGP Command

Description: All global parameters can be read with this function. See chapter 7.5 for a complete parameter list.

Internal function: The parameter is read out of the correct position in the appropriate device. The parameter format is converted adding leading zeros (or ones for negative values).

Related commands: SGP, STGP, RSGP, AGP

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

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
10	<parameter number>	<bank number>	(don't care)

Reply in direct mode:

STATUS	VALUE
100 - OK	<value>

Example:

Get the serial address of the target device

Mnemonic: GGP 66, 0

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x0A	0x42	0x00	0x00	0x00	0x00	0x00	0x4D

Reply:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Host address	Target. address	Status	Instruct. number	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x02	0x01	0x64	0x0A	0x00	0x00	0x00	0x01	0x72

STATUS	VALUE
100 – OK	1

7.3.11 STGP Command

Description: Some global parameters are located in RAM memory, so modifications are lost at power down. This instruction enables permanent storing. Most parameters are automatically restored after power up (see the list of global parameters in chapter 7.5).

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

Related commands: SGP, GGP, RSGP, AGP

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

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
11	<parameter number>	<bank number>	(don't care)

Reply in direct mode:

STATUS	VALUE
100 - OK	(don't care)

Example:

Store the serial address of the target device

Mnemonic: STGP 66, 0

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x0B	0x42	0x00	0x00	0x00	0x00	0x00	0x4E

Note: The STGP command will not have any effect when the configuration EEPROM is locked.

7.3.12 RSGP Command

Description: This instruction recovers the (permanently) stores the value of a RAM located parameter. Please see chapter 7.5 for a list of available parameters.

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

Related commands: SGP, GGP, STGP, AGP

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

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
12	<parameter number>	<bank number>	(don't care)

Reply in direct mode:

STATUS	VALUE
100 - OK	(don't care)

Example:

Restore the serial address of the target device

Mnemonic: RSGP 66, 0

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x0C	0x42	0x00	0x00	0x00	0x00	0x00	0x4F

7.3.13 Get Version Number

Description: This instruction returns the module type and firmware revision either as a string or in binary format.

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
136	0 - string	(don't care)	(don't care)
	1 - binary		

Reply in direct mode:

STATUS	VALUE
100 - OK	(don't care)

Example:

Request module type and firmware revision (e.g. 603V1.26)

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x88	0x00	0x00	0x00	0x00	0x00	0x00	0x89

Reply:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Host address	Module type and firmware revision							
		Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Value (hex)	0x02	0x36	0x30	0x33	0x56	0x00	0x32	0x2E	0x32

Note: This reply is not in the standard TMCL syntax.

7.3.14 Restore Factory Settings

Description: This instruction resets all settings stored in the EEPROM to their factory defaults. This command does not sen back a reply.

Binary representation:

INSTRUCTION NO.	TYPE	MOT/BANK	VALUE
137	(don't care)	(don't care)	(don't care)

Example:

Restore factory default settings

Binary:

BYTE INDEX	0	1	2	3	4	5	6	7	8
Function	Target address	Instruct. number	Type	Motor/ Bank	Operand Byte 3	Operand Byte 2	Operand Byte 1	Operand Byte 0	Check sum
Value (hex)	0x01	0x89	0x00	0x00	0x00	0x00	0x00	0x00	0x8A

7.4 Axis Parameters

The following section describes all axis parameters that can be used with the SAP, GAP, STAP and RSAP commands. The letters under 'access' mean: R = readable (GAP), W = writable (SAP), E = automatically restored from EEPROM after reset or power on.

Type	Axis Parameter	Value Range	Description	Access
0	Target position	32 bit signed	The target position of a currently executed ramp.	R
1	Actual position	32 bit signed	Change the position counter without moving the motor.	RW
2	Target speed	32 bit signed	Programmed target velocity in [rpm].	R
3	Actual speed	32 bit signed	The actual velocity of the motor in [rpm].	R
4	Max. positioning velocity/ velocity threshold for hallFX™	32 bit signed	The maximum velocity in [rpm] used for MVP command when executing a ramp to a position. In sensorless commutation mode the velocity threshold in [rpm] for hallFX™.	RWE
5	PWM limit	32 bit signed 0...3599	Set PWM limit (0%...100%).	RWE
6	Max current	32 bit unsigned	Max. current setting in [mA].	RWE
7	MVP Target reached velocity	32 bit signed	Maximum velocity in [rpm] at which end position can be set. Prevents issuing of end position when the target is passed at high velocity.	RWE
8	Min. speed for velocity PID	32 bit signed	Adjusts the limit in [rpm] to switch between first velocity PID parameter set and second velocity PID parameter set.	RWE
9	Clear target distance	32 bit signed	Velocity is set to 0 if actual position differs from motor position for more than value, until the motor catches up. Prevents velocity overshoot if the motor can't follow the velocity ramp.	RWE
10	MVP target reached distance	32 bit signed	Maximum distance at which the position end flag is set.	RWE
11	Acceleration	32 bit signed	Acceleration parameter for MVP, ROL and ROR. Value in [RPM/s].	RWE
12	Min. speed for position PID	32 bit signed	Adjusts the limit in [rpm] to switch between first position PID parameter set and second position PID parameter set.	RWE
130	Position P-Parameter (I)	32 bit signed	P–Parameter of PID position regulator (first position parameter set)	RWE
131	Position I-Parameter (I)	32 bit signed	I–Parameter of PID position regulator (first position parameter set)	RWE
132	Position D-Parameter (I)	32 bit signed	D–Parameter of PID position regulator (first position parameter set)	RWE
133	PID regulation loop delay	32 bit unsigned	PID calculation delay in [ms]: Set operational frequency PID	RWE
134	Current regulation loop delay	32 bit unsigned	Delay of current limitation algorithm / PID current regulator in [50µs].	RWE

Type	Axis Parameter	Value Range	Description	Access
135	Position I-Clipping-Parameter (I)	32 bit signed	Adjust in stand still to lowest possible value at which the motor keeps its position. A too high value causes overshooting at positioning mode. (first position parameter set)	RWE
136	PWM-Hysteresis	32 bit unsigned	Compensates dead time of PWM and motor friction.	RWE
137	PID opt. Clear I-Max	Boolean	1: Clears I–Sum if PWM reaches maximum value of 100%.	RWE
138	PID opt. Clear I-Sign	Boolean	1: Clears I–Sum if the position overshoots the target value.	RWE
140	Velocity P-Parameter (I)	32 bit signed	P–Parameter of PID velocity regulator. This PID parameter set is used at lower velocity. (first velocity parameter set)	RWE
141	Velocity I-Parameter (I)	32 bit signed	I–Parameter of PID velocity regulator. This PID parameter set is used at lower velocity. (first velocity parameter set)	RWE
142	Velocity D-Parameter (I)	32 bit signed	D–Parameter of PID velocity regulator. This PID parameter set is used at lower velocity. (first velocity parameter set)	RWE
143	Velocity I-Clipping-Parameter (I)	32 bit signed	This PID parameter set is used at lower velocity. (first velocity parameter set)	RWE
146	Velocity PID-Control	Boolean	1: Activate velocity ramp generator for MVP. Allows usage of acceleration and positioning velocity for MVP command.	RWE
147	Disable PID	8 bit unsigned	0: Enable the standard PID calculation. 1: Disable the PID calculation. The Motor PWM is then directly derived from the target velocity. 2: Enable an integrating PID algorithm. In this case the result of the PID calculation is integrated to the PWM value. This PID regulation is easier to use than the standard PID regulation.	RWE
150	Actual motor current	32 bit unsigned	Actual motor current in [mA].	R
151	Actual voltage	32 bit unsigned	Actual supply voltage.	R
152	Actual temperature	32 bit unsigned	Actual temperature.	R
153	Actual PWM duty cycle	32 bit unsigned 0...3599	Actual PWM duty cycle (0%...100%).	R

Type	Axis Parameter	Value Range	Description	Access
156	Error/Status flags	16 bit unsigned	Bit 0: Overcurrent flag. This flag is set if overcurrent limit is exceeded. Bit 1: Undervoltage flag. This flag is set if supply voltage to 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 motor has been switched off. Bit 5: Hall error flag. This flag is set upon a hall error. Bit 6: Encoder error flag. This flag is set upon an encoder error. Bit 9: Initialization error of sine commutation. This flag is set if initialization is failed. Bit 10: Position mode flag. This flag is set when the module is in positioning mode. Bit 11: Position end flag. This flag becomes set if the motor has been stopped at the end position.	R
159	Commutation mode	8 bit unsigned	0: Block commutation with hall sensors 1: Sensorless block commutation (hallFX™) 2: Sine commutation with hall sensors 3: Sine commutation with encoder 4: Controlled block commutation 5: Controlled sine commutation	RWE
160	Re-Initialization of Sine	Boolean	1: Sine commutation is re-initialized. Attention: Depending on initialization mode, stop motor before issuing this command!	RW
161	Encoder set NULL	Boolean	1: Set Encoder counter to zero at next N channel event.	RWE
163	Encoder clear set NULL	Boolean	1: Set encoder counter zero only once 0: always at a N channel event, respectively switch event.	RWE
165	Actual commutation offset	32 bit signed	This value represents the internal commutation offset. (0 ... max. Encoder steps per rotation)	RWE
167	Block PWM scheme	8 bit unsigned	0: PWM chopper on high side, HI on low side 1: PWM chopper on low side, HI on high 2: PWM chopper on low side and high side	RWE
168	Current P-Parameter (I)	32 bit signed	P–Parameter of PID current regulator. This PID parameter set is used at lower velocity. (first current parameter set)	RWE
169	Current I-Parameter (I)	32 bit signed	I–Parameter of PID current regulator. This PID parameter set is used at lower velocity. (first current parameter set)	RWE
170	Current D-Parameter (I)	32 bit signed	D–Parameter of PID current regulator. This PID parameter set is used at lower velocity. (first current parameter set)	RWE
171	Current I-Clipping-Parameter (I)	32 bit signed	I-Clipping–Parameter of PID current regulator. This PID parameter set is used at lower velocity. (first current parameter set)	RWE

Type	Axis Parameter	Value Range	Description	Access
172	Current P-Parameter (II)	32 bit signed	P–Parameter of PID current regulator. This PID parameter set is used at higher velocity. (second current parameter set)	RWE
173	Current I-Parameter (II)	32 bit signed	I–Parameter of PID current regulator. This PID parameter set is used at higher velocity. (second current parameter set)	RWE
174	Current D-Parameter (II)	32 bit signed	D–Parameter of PID current regulator. This PID parameter set is used at higher velocity. (second current parameter set)	RWE
175	Current I-Clipping-Parameter (II)	32 bit signed	I-Clipping–Parameter of PID current regulator. This PID parameter set is used at higher velocity. (second current parameter set)	RWE
176	Velocity threshold for current PID	32 bit signed	Adjusts the limit in [rpm] to switch between first current PID controller and second current PID controller.	RWE
177	Start current	32 bit unsigned	Motor current for controlled commutaion in [mA]. This parameter is used in commutation mode 1, 4, 5 and in initialisation of sine.	RWE
209	Actual encoder position	32 bit signed	Actual encoder position / counter value	R
211	Gear ratio	32 bit signed	Transmission ratio of gear, e.g. gear ratio of 6 for transmission ratio 6:1	RW
226	Position Error	32 bit signed	Actual error of PID position regualtor	R
227	Position Error Sum	32 bit signed	Sums of errors of PID position regualtor	R
228	Velocity Error	32 bit signed	Actual error of PID velocity regualtor	R
229	Velocity Error Sum	32 bit signed	Sums of Errors of PID velocity regualtor	R
230	Position P-Parameter (II)	32 bit signed	P–Parameter of PID position regulator (second position parameter set)	RWE
231	Position I-Parameter (II)	32 bit signed	I–Parameter of PID position regulator (second position parameter set)	RWE
232	Position D-Parameter (II)	32 bit signed	D–Parameter of PID position regulator (second position parameter set)	RWE
233	Position I-Clipping-Parameter (II)	32 bit signed	Adjust in stand still to lowest possible value at which the motor keeps its position. A too high value causes overshooting at positioning mode. (second position parameter set)	RWE
234	Velocity P-Parameter (II)	32 bit signed	P–Parameter of PID velocity regulator (second position parameter set)	RWE
235	Velocity I-Parameter (II)	32 bit signed	I–Parameter of PID velocity regulator (second position parameter set)	RWE
236	Velocity D-Parameter (II)	32 bit signed	D–Parameter of PID velocity regulator (second position parameter set)	RWE
237	Velocity I-Clipping-Parameter (II)	32 bit signed	I-Clipping–Parameter of PID current regulator. This PID parameter set is used at lower velocity. (second position parameter set)	RWE

Type	Axis Parameter	Value Range	Description	Access
238	Mass inertial constant	32 bit signed 0...65535	Mass inertial constant for position regulation. Compensates mass moment of inertia of rotor.	RWE
239	BEMF constant	32 bit signed	BEMF constant of motor in [rpm/(10V)]. Used for current regulation, position regulation and velocity regulation. Feedforward control for current regulation, position regulation and velocity regulation is disabled if BEMF constant is set to zero.	RWE
240	Motor coil resistance	32 bit signed	Resistance of motor coil in [m□]. Used for current regulation, position regulation and velocity regulation.	RWE
241	Init sine speed	16 bit signed	Velocity in [rpm] for sine initialization.	RWE
242	Init sine block offset CW	16 bit signed	This parameter helps to tune hall sensor based initialization in a way, that the motor has the same velocity for left and right turn. It compensates for tolerance and hysteresis of the hall sensors. It is added to the Commutation offset upon CW turn initialization.	RWE
243	Init sine block offset CCW	16 bit signed	This parameter helps to tune hall sensor based initialization in a way, that the motor has the same velocity for left and right turn. It compensates for tolerance and hysteresis of the hall sensors. It is added to the Commutation offset upon CCW turn initialization.	RWE
244	Init sine delay	16 bit signed	Duration for sine initialization sequence in [ms]. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	RWE
245	Overvoltage protection	Boolean	1: Enable overvoltage protection.	RWE
246	Maximum PWM change per PID interval	16 bit unsigned 0...3599	Maximum PWM change per PID interval.	RWE
247	Sine Compensation Factor	32 bit signed 0...255	Compensates the propagation delay of the MCU	RWE
249	Init sine mode	8 bit unsigned	0: Initialization in controlled sine commutation	RWE
250	Encoder steps	32 bit unsigned	Encoder Steps per Rotation.	RWE
251	Encoder direction	Boolean	Encoder direction [0/1]. Set this flag in a way, that turn right increases position counter.	RWE
252	Encoder null polarity	Boolean	Encoder null polarity for nulling of position counter.	RWE
253	Number of motor poles	8 bit unsigned 2...254	Number of motor poles.	RWE
254	Hall sensor invert	Boolean	1: Hall sensor invert. Sets one of the motors with inverted hall scheme, e.g. some Maxon motors	RWE

Position end flag:

The position end flag informs the user, that the motor has reached the specified target position. This flag becomes set, whenever the following conditions are fulfilled at the same time:

- the distance is within range as defined by parameter 'target reached distance'
- the sum of target velocity and actual velocity is within range as defined by parameter 'target reached velocity'

The position end flag becomes reset by the following user actions:

- ROL or ROR instruction
- MST instruction
- new MVP command
- setting of the actual position

Motor halted flag:

The motor halted flag becomes set, when the motor is stopped by switching off the drivers. Switching off the drivers makes the motor powerless an it is easy to turn by hand. The motor additionally stops to regulate to the actual target position. This flag becomes set upon the following conditions:

- Undervoltage
- Overvoltage (if overvoltage detection is enabled)
- Overtemperature
- MST instruction

As long as any one or multiple of these conditions become true, the target velocity becomes set to zero and the motor is being switched off. The motor halted flag becomes reset, as soon as all conditions triggering a halt have been cleared.

7.5 Global Parameters

The following section describes all global parameters that can be used with the SGP, GGP, STGP and RSGP commands The letters under 'access' mean: R = readable (GGP), W = writable (SGP), E = automatically restored from EEPROM after reset or power on.

Type	Axis Parameter	Value Range	Description	Access
65	RS232 baud rate	8 bit unsigned (0...8)	Change of RS232 baud rate. Reset afterwards. 0: 9600 baud (default) 1: 14400 baud 2: 19200 baud 3: 28800 baud 4: 38400 baud 5: 57600 baud 6: 76800 baud 7: 115200 baud 8: 230400 baud	RWE
66	Serial address	8 bit unsigned (0...255)	The target address for RS-232 (default=1).	RWE
73	Configuration EEPROM lock flag	Boolean	Lock and unlock the EEPROM 0: EEPROM unlocked 1: EEPROM locked	RWE
75	Telegram pause time	8 bit unsigned	Pause time in [ms] before reply via RS232 will be sent (default=0).	RWE
76	Serial host address	8 bit unsigned	Host address used in the reply telegrams sent back via RS232 (default=2).	RWE

7.6 TMCL Code Examples

7.6.1 Block commutation by using hall sensors

The following TMCL demo shows how to configurate the TMC603 evaluation board to run the BLDC motor in block commutation by using hallsensor feedback. Please refer chapter 7.4 for detailed information about the axis parameters.

TMCL demo application:

```
  MST 0               //Stop motor

  //*******************************************************************
  // Configuration
  //*******************************************************************

  //General settings
  SAP 6, 0, 6000      //Set maximum motor current: 6000mA
  SAP 11, 0, 200      //Set acceleration: 200rpm/s

  SAP 253, 0, 4       //Set number of motor poles: 4
  SAP 240, 0, 350     //Set resistance of motor coils: 350mOhm
  SAP 254, 0, 0       //Disable reverse hall sensor polarity

  SAP 147, 0, 2       //Select the integrating PID algorithm
  SAP 159, 0, 0       //Select block commutation by using hall sensors
  SAP 167, 0, 0       //Select block PWM scheme with chopper on high side

  //Velocity PID parameter settings
  SAP 8, 0, 1000      //Set the velocity threshold: 1000rpm

  SAP 140, 0, 1000    //Set P-Parameter of the 1st parameter set: 1000
  SAP 141, 0, 167     //Set I-Parameter of the 1st parameter set: 167
  SAP 142, 0, 30      //Set D-Parameter of the 1st parameter set: 30
  SAP 143, 0, 160     //Set I-Clipping-Parameter of the 1st parameter set: 160

  SAP 234, 0, 2500    //Set P-Parameter of the 1st parameter set: 2500
  SAP 235, 0, 42      //Set I-Parameter of the 1st parameter set: 42
  SAP 236, 0, 8       //Set D-Parameter of the 1st parameter set: 8
  SAP 237, 0, 640     //Set I-Clipping-Parameter of the 1st parameter set: 640

  //*******************************************************************
  // Program sequence
  //*******************************************************************
Repeat:
  ROR 0, 2000         //Rotate right, target velocity 2000rpm
  WAIT TICKS, 0, 2500

  ROL 0, 2000         //Rotate left, target velocity 2000rpm
  WAIT TICKS, 0, 2500

  MST 0               //Stop motor
  WAIT TICKS, 0, 2500

  JA Repeat           //Infinite loop

  STOP
```

7.6.2 Sensorless block commutation by using hallFX

The following TMCL demo shows how to configurate the TMC603 evaluation board to run the BLDC motor in sensorless block commutation by using the back EMF commutation hallFX™. The motor starts in controlled block commutation. After startup the motor switchs to sensorless block commutation (hallFX). Please refer chapter 7.4 for detailed information about the axis parameters.

TMCL demo application:

```
MST 0                //Stop motor

//*******************************************************************
// Configuration
//*******************************************************************

//General settings
SAP 4, 0, 2000       //Set velocity threshold: 2000rpm
SAP 6, 0, 6000       //Set maximum motor current: 6000mA
SAP 11, 0, 500       //Set acceleration: 500rpm/s
SAP 177, 0, 3000     //Set start current: 3000mA

SAP 253, 0, 4        //Set number of motor poles: 4
SAP 240, 0, 350      //Set resistance of motor coils: 350mOhm

SAP 147, 0, 2        //Select the integrating PID algorithm
SAP 159, 0, 1        //Select sensorless block commutation by using hallFX
SAP 167, 0, 2        //Select block PWM scheme with dual side chopper

//Current PID parameter settings
SAP 176, 0, 1000     //Set the velocity threshold: 1000rpm

SAP 168, 0, 50       //Set P-Parameter of the 1st parameter set: 50
SAP 169, 0, 10       //Set I-Parameter of the 1st parameter set: 10
SAP 170, 0, 0        //Set D-Parameter of the 1st parameter set: 0
SAP 171, 0, 6600     //Set I-Clipping-Parameter of the 1st parameter set: 6600

SAP 172, 0, 800      //Set P-Parameter of the 1st parameter set: 800
SAP 173, 0, 10       //Set I-Parameter of the 1st parameter set: 10
SAP 174, 0, 0        //Set D-Parameter of the 1st parameter set: 0
SAP 175, 0, 8600     //Set I-Clipping-Parameter of the 1st parameter set: 8600

//Velocity PID parameter settings
SAP 8, 0, 1000       //Set the velocity threshold: 1000rpm

SAP 140, 0, 1000     //Set P-Parameter of the 1st parameter set: 1000
SAP 141, 0, 167      //Set I-Parameter of the 1st parameter set: 167
SAP 142, 0, 30       //Set D-Parameter of the 1st parameter set: 30
SAP 143, 0, 160      //Set I-Clipping-Parameter of the 1st parameter set: 160

SAP 234, 0, 2500     //Set P-Parameter of the 1st parameter set: 2500
SAP 235, 0, 42       //Set I-Parameter of the 1st parameter set: 42
SAP 236, 0, 8        //Set D-Parameter of the 1st parameter set: 8
SAP 237, 0, 640      //Set I-Clipping-Parameter of the 1st parameter set: 640


//*******************************************************************
// Program sequence
//*******************************************************************
ROR 0, 2200          //Rotate right, target velocity 2200rpm
WAIT TICKS, 0, 2500

STOP
```

7.6.3 Sinus commutation by using encoder feedback

The following TMCL demo shows how to configurate the TMC603 evaluation board to run the BLDC motor in sine commutation by using encoder feedback. Before starting the motor, an initialization sequence for sine commutation is executed. At this time, the encoder commutation offset is calculated and is automatically stored to EEPROM. There after, the motor moves to homing position. Then, the motor continuously moves to absolute position 10000 and -10000. Please refer chapter 7.4 for detailed information about the axis parameters.

TMCL demo application 1:

```
MST 0                //Stop motor

//*****************************************************************
// Configuration
//*****************************************************************

//General settings
SAP 6, 0, 6000       //Set maximum motor current: 6000mA
SAP 11, 0, 1000      //Set acceleration: 1000rpm/s
SAP 177, 0, 2000     //Set start current: 2000mA

SAP 253, 0, 4        //Set number of motor poles: 4
SAP 240, 0, 350      //Set resistance of motor coils: 350mOhm
SAP 250, 0, 2000     //Set encoder steps per rotation: 2000
SAP 251, 0, 1        //Set encoder direction

SAP 147, 0, 2        //Select the integrating PID algorithm
SAP 159, 0, 3        //Select sine commutation by using encoder feedback

SAP 4, 0, 2000       //Set maximum velocity: 2000rpm
SAP 7, 0, 10         //Set target reached velocity: 10rpm
SAP 10, 0, 20        //Set target reached distance: 20

SAP 241, 0, 60       //Velocity in [rpm] for sine initialization
SAP 244, 0, 2000     //Duration for sine initialization sequence in [ms]. This parameter
                     //should be set in a way, that the motor has stopped mechanical
                     //oscillations after the specified time.

SAP 163, 0, 1        //Set encoder counter to zero only once

//Current PID parameter settings
SAP 176, 0, 1000     //Set the velocity threshold: 1000rpm

SAP 168, 0, 50       //Set P-Parameter of the 1st parameter set: 50
SAP 169, 0, 10       //Set I-Parameter of the 1st parameter set: 10
SAP 170, 0, 0        //Set D-Parameter of the 1st parameter set: 0
SAP 171, 0, 6600     //Set I-Clipping-Parameter of the 1st parameter set: 6600

SAP 172, 0, 800      //Set P-Parameter of the 1st parameter set: 800
SAP 173, 0, 10       //Set I-Parameter of the 1st parameter set: 10
SAP 174, 0, 0        //Set D-Parameter of the 1st parameter set: 0
SAP 175, 0, 8600     //Set I-Clipping-Parameter of the 1st parameter set: 8600

//Position PID parameter settings
SAP 12, 0, 450       //Set the velocity threshold: 450rpm

SAP 130, 0, 187500   //Set P-Parameter of the 1st parameter set: 187500
SAP 131, 0, 20200    //Set I-Parameter of the 1st parameter set: 20200
SAP 132, 0, 1000     //Set D-Parameter of the 1st parameter set: 1000
SAP 135, 0, 500000   //Set I-Clipping-Parameter of the 1st parameter set: 500000

SAP 230, 0, 207500   //Set P-Parameter of the 1st parameter set: 207500
SAP 231, 0, 40400    //Set I-Parameter of the 1st parameter set: 40400
SAP 232, 0, 10       //Set D-Parameter of the 1st parameter set: 10
SAP 233, 0, 500000   //Set I-Clipping-Parameter of the 1st parameter set: 500000
```

```
  //*******************************************************************
  // Program sequence
  //*******************************************************************

  //Initialization of sine commutation
  SAP 161, 0, 1        //Set encoder counter to zero at next N channel event
  SAP 160, 0, 0        //Start initialization of sine commutation
Loop:
  GAP 160, 0           //Wait until initialization of sine commutation is ready
  JC ZE, Loop

  //Homing procedure
  MST 0                //Stop motor
  MVP ABS, 0, 0        //Move to absolute position 0 (Home)
  WAIT POS, 0, 0       //Wait until position is reached
  WAIT TICKS, 0, 500

  //Main loop
Repeat:
  MVP ABS, 0, 10000    //Move to absolute position 10000
  WAIT POS, 0, 0       //Wait until position is reached
  WAIT TICKS, 0, 500

  MVP ABS, 0, -10000   //Move to absolute position -10000
  WAIT POS, 0, 0       //Wait until position is reached
  WAIT TICKS, 0, 500

  JA Repeat            //Infinite loop

  STOP
```

After restarting the evaluation board, no initialization sequence is required, when the initialization sequence is already executed and the encoder commutation offset is already stored to EEPROM before. Only the encoder N channel event has to be found. In that case the motor is rotated in controlled sine commutation, until the encoder N channel event is occurred. In the following a code example can be found. In order to ensure that this example works properly, restart the evaluation board before execute this TMCL program.

TMCL demo application 2:

```
  MST 0                //Stop motor

  //*******************************************************************
  // Configuration
  //*******************************************************************

  //General settings
  SAP 6, 0, 6000       //Set maximum motor current: 6000mA
  SAP 11, 0, 1000      //Set acceleration: 1000rpm/s
  SAP 177, 0, 2000     //Set start current: 2000mA

  SAP 253, 0, 4        //Set number of motor poles: 4
  SAP 240, 0, 350      //Set resistance of motor coils: 350mOhm
  SAP 250, 0, 2000     //Set encoder steps per rotation: 2000
  SAP 251, 0, 1        //Set encoder direction

  SAP 147, 0, 2        //Select the integrating PID algorithm
  SAP 159, 0, 3        //Select sine commutation by using encoder feedback

  SAP 4, 0, 2000       //Set maximum velocity: 2000rpm
  SAP 7, 0, 10         //Set target reached velocity: 10rpm
  SAP 10, 0, 20        //Set target reached distance: 20

  SAP 163, 0, 1        //Set encoder counter to zero only once
```

```
  //Current PID parameter settings
  SAP 176, 0, 1000     //Set the velocity threshold: 1000rpm

  SAP 168, 0, 50       //Set P-Parameter of the 1st parameter set: 50
  SAP 169, 0, 10       //Set I-Parameter of the 1st parameter set: 10
  SAP 170, 0, 0        //Set D-Parameter of the 1st parameter set: 0
  SAP 171, 0, 6600     //Set I-Clipping-Parameter of the 1st parameter set: 6600

  SAP 172, 0, 800      //Set P-Parameter of the 1st parameter set: 800
  SAP 173, 0, 10       //Set I-Parameter of the 1st parameter set: 10
  SAP 174, 0, 0        //Set D-Parameter of the 1st parameter set: 0
  SAP 175, 0, 8600     //Set I-Clipping-Parameter of the 1st parameter set: 8600

  //Position PID parameter settings
  SAP 12, 0, 450       //Set the velocity threshold: 450rpm

  SAP 130, 0, 187500   //Set P-Parameter of the 1st parameter set: 187500
  SAP 131, 0, 20200    //Set I-Parameter of the 1st parameter set: 20200
  SAP 132, 0, 1000     //Set D-Parameter of the 1st parameter set: 1000
  SAP 135, 0, 500000   //Set I-Clipping-Parameter of the 1st parameter set: 500000

  SAP 230, 0, 207500   //Set P-Parameter of the 1st parameter set: 207500
  SAP 231, 0, 40400    //Set I-Parameter of the 1st parameter set: 40400
  SAP 232, 0, 10       //Set D-Parameter of the 1st parameter set: 10
  SAP 233, 0, 500000   //Set I-Clipping-Parameter of the 1st parameter set: 500000

  //*******************************************************************
  // Program sequence
  //*******************************************************************
  //Re-Initialization of sine commutation
  SAP 159, 0, 5        //Select controlled sine commutation without any feedback
  SAP 161, 0, 1        //Set encoder counter to zero at next N channel event
  ROR 0, 10            //Rotate right, target velocity 10rpm
Loop:
  GAP 160, 0           //Wait until initialization of sine commutation is ready
  JC ZE, Loop

  MST 0                //Stop motor
  SAP 159, 0, 3        //Select sine commutation by using encoder feedback

  //Homing procedure
  MST 0                //Stop motor
  MVP ABS, 0, 0        //Move to absolute position 0 (Home)
  WAIT POS, 0, 0       //Wait until position is reached
  WAIT TICKS, 0, 500

  //Main loop
Repeat:
  MVP ABS, 0, 10000    //Move to absolute position 10000
  WAIT POS, 0, 0       //Wait until position is reached
  WAIT TICKS, 0, 500

  MVP ABS, 0, -10000   //Move to absolute position -10000
  WAIT POS, 0, 0       //Wait until position is reached
  WAIT TICKS, 0, 500

  JA Repeat            //Infinite loop

  STOP
```

8 Firmware

PC software to operate the TMC603 evaluation board is supplied on the TMC TechLibCD or can be downloaded at http://www.trinamic.com. The program can be used with Windows 98, Windows 2000 and Windows XP. To install it, just copy the file 'TMC603EVAL.EXE' to the hard disk of your PC. To run the program, double click the file.

8.1 Updating the firmware

The TMC603 evaluation software enables to modify the basic operations of motor control. However to update the firmware of the TMC603 evaluation board, the TMCL IDE is required. The TMCL IDE is supplied on the TMC TechLibCD or can be downloaded at http://www.trinamic.com. To install it, just copy the file 'TMCL.EXE' to the hard disk of your PC. To run the program, double click the file.

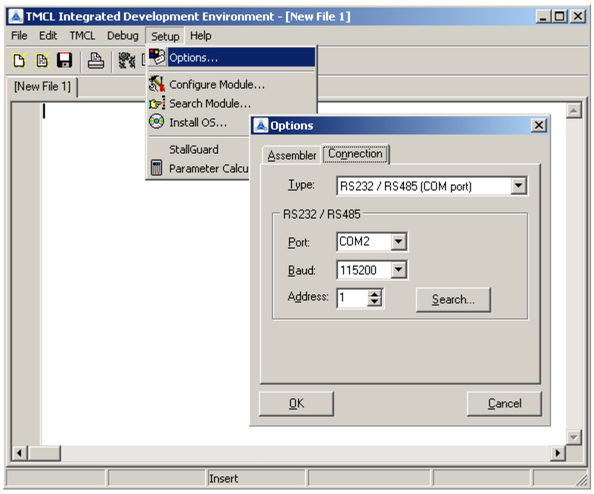


Figure 14: Connection options

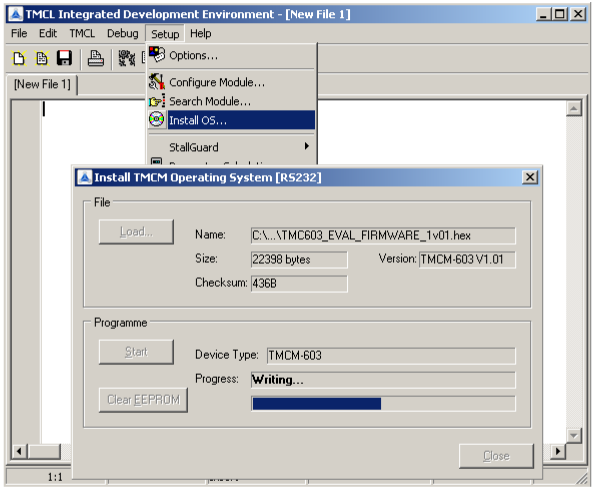


Figure 15: Install operating system

At first, open the connection options. Then select 'RS232/RS485 (COM port)' and the COM port that is to be used. Set the baud rate to 9600 baud. The module address is 1.

Now, open the install OS window. At first, load the new firmware file by pressing the 'Load' button. The file is then checked if it is a TMCL firmware file. Its device type and version number will be displayed. Then, press the 'Start' button to program the new firmware into the module. Please make sure that there will be no power cut or cut of the serial connection during the programming process. The program checks if the device type in the firmware file and the device type of the evaluation board are identical. An error message will be displayed if this is should not be the case. If everything is okay, the new firmware will be programmed into the evaluation board and verified afterwards. The programming progress is shown by the status bar.

After the programming process, you can close the TMCL IDE. Then, start the TMC603 evaluation software to run the motor.

9 List of figures

10 List of tables

11 References

TMC603A datasheet	(see http://www.trinamic.com)
TMC603-EVAL schematics	(see http://www.trinamic.com)
TMCL Reference and Programming Manual	(see http://www.trinamic.com)

12 Revision History

12.1 Documentation Revision

Version	Date	Author	Description
0.90	13.07.10	MW	Initial Release.
0.91	28.07.10	MW	Chapter 3, 6 and 7 added and changed.
0.92	03.08.10	MW	Some changes in chapter 3.2, 3.3 and 3.4. In chapter 4.1 added remark to default jumper settings. In chapter 6 added remark to corresponding axis parameters.
0.93	06.08.10	MW	In chapter 3 changed figure 1. Chapter 7.6 TMCL code examples added. Some cosmetic changes.

12.2 Firmware Revision

Version	Date	Author	Description
1.24	13.07.10	MW	Initial Release.
1.26	16.07.10	MW	USB communication added.

12.3 PC Software Revision

Version	Date	Author	Description
0.97	13.07.10	MW	Initial Release

12.4 Hardware Revision

Version	Date	Author	Description
2.10	13.07.10	MW	Initial Release