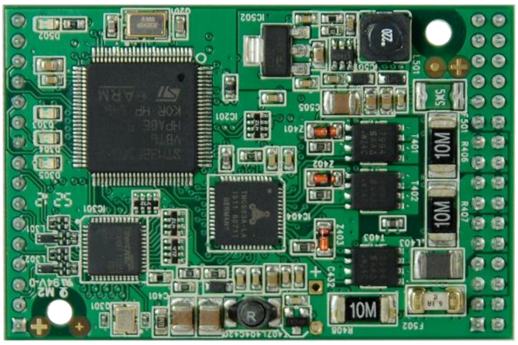
V 2.8

ETHERCAT™ MANUAL



TMCM-1632

1-axis BLDC
controller / driver module
3A RMS / 24V DC nominal
EBUS (EtherCAT™) interface
hall*FX*™

TMCM-1610-KR

1-axis BLDC
controller / driver module
3A RMS / 24V DC nominal
EBUS (EtherCAT™) interface
hall*FX*™

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

TRINAMIC
MOTION CONTROL

Table of contents

1 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 Features

2.1 TMCM-1632

The TMCM-1632 is a highly integrated single axis BLDC servo controller module with EtherCAT™ interface (E-Bus). The module has been designed in order to be plugged onto a baseboard. It offers two E-Bus ports for daisy-chaining several modules and connects them to an EtherCAT™ master (e.g. embedded PC) for multi-axes solutions. The module offers hall sensor and incremental encoder (a/b/n) inputs.

Applications
- Demanding single and multi-axis BLDC motor solutions with high update rate / close synchronization between master (e.g. industrial PC) and motion controller board(s) (slave)

Electrical data
- Supply voltage: +24V DC nominal (+12V… +28.5V DC max.)
- Motor current: 3A RMS continuous and up to 5A RMS (programmable) peak

Interfaces
- 2x EBUS (EtherCAT™) interface
- 1x RS232 (TTL_3V3 level)
- 4 general purpose inputs (+24V and +5V compatible) and 2 general purpose outputs (open drain). Outputs are shared with inputs (same connector pins).

Integrated motion controller
- High performance ARM Cortex™-M3 microcontroller for system control and communication protocol handling

Integrated motor driver
- High performance integrated pre-driver (TMC603A)
- Support for sensorless back EMF commutation (hall*FX*™)
- High-efficient operation, low power dissipation (MOSFETs with low $R_{DS(ON)}$)
- Dynamic current control
- Integrated protection

Software
- Via the EtherCAT™ mailbox motor parameters are written/read using the TMCL™ protocol

Please refer to the TMCM-1632 Hardware Manual for further information.

2.2 TMCM-1610-KR

The TMCM-1610-KR is a highly compact controller/driver module for brushless DC motors with E-Bus (EtherCAT) interface, optional encoder and / or hall sensor feedback. For communication with the gripper module TMCM-KR-842 the modules offers a serial interface and supply voltage connector.

Applications

- KUKA youBot

Electrical data

- Supply voltage: +24VDC nom. (+12V .. +28.5V DC)
- Motor current: 3A RMS continuous and up to 5A RMS (programmable) peak

Integrated motion controller

- High performance ARM Cortex™-M3 microcontroller for system control and communication protocol handling

Integrated motor driver

- High performance integrated pre-driver (TMC603A)
- Support for sensorless back EMF commutation (hall*FX*™)
- High-efficient operation, low power dissipation (MOSFETs with low $R_{DS(ON)}$)
- Dynamic current control
- Integrated protection

Interfaces

- E-Bus (EtherCAT™)
- Hall sensor interface (+5V TTL or open-collector signals)
- Encoder interface (+5V TTL or open-collector signals)
- Gripper serial interface and supply voltage connector

Software

- Via the EtherCAT™ mailbox motor-parameters are written/read using the TMCL™ protocol

Please refer to the TMCM-1610-KR Hardware Manual for further information.

3 Overview

As with most TRINAMIC modules the software running on the ARM Cortex™-M3 processor of the TMCM-1632/ TMCM-1610-KR consists of two parts, a boot loader and the firmware itself. Whereas the boot loader is installed during production and testing at TRINAMIC and remains – normally – untouched throughout the whole lifetime, the firmware can be updated by the user.

The TMCM-1632/TMCM-1610-KR is an EtherCAT™ slave device. The whole communication with an EtherCAT™ master follows a strict master-slave-relationship. Via the TMCL™ mailbox motor parameters are written and/or read using TRINAMICs TMCL™ protocol.

All parameters and commands which can be used are described on the following pages.

Please refer to paragraph 6 for more information about updating the firmware. New versions can be downloaded free of charge from the TRINAMIC website (http://www.trinamic.com).

4 SyncManager

The SyncManager enables consistent and secure data exchange between the EtherCAT™ master and the local application, and it generates interrupts to inform both sides of changes. The SyncManager is configured by the EtherCAT master. The communication direction is configurable, as well as the communication mode (buffered mode and mailbox mode). The SyncManager uses a buffer located in the memory area for exchanging data. Access to this buffer is controlled by the hardware of the SyncManager.

A buffer has to be accessed beginning with the start address, otherwise the access is denied. After accessing the start address, the whole buffer can be accessed, even the start address again, either as a whole or in several strokes. A buffer access finishes by accessing the end address, the buffer state changes afterwards. The end address cannot be accessed twice inside a frame. Two communication modes are supported by SyncManagers, the *buffered mode* and the *mailbox mode*.

4.1 Buffered mode

The buffered mode allows both sides, EtherCAT™ master and local application, to access the communication buffer at any time. The consumer gets always the latest consistent buffer which was written by the producer, and the producer can always update the content of the buffer. The buffered mode is used for cyclic process data.

Data transfer between EtherCAT™ master (PC etc.) und slave (TMCM-1632/TMCM-1610-KR) is done using the dual port memory of the ET1200 EtherCAT-IC on the slave. The buffered mode allows writing and reading data simultaneously without interference. If the buffer is written faster than it is read out, old data will be dropped. The buffered mode is also known as 3-buffermode. One buffer of the three buffers is allocated to the producer (for writing), one buffer to the consumer (for reading), and the third buffer keeps the last consistently written data of the producer.

0x1000 - 0x10FF	Buffer 0 (visible)
0x1100 - 0x11FF	Buffer 1 (invisible, shall not be used)
0x1200 - 0x12FF	Buffer 2 (invisible, shall not be used)
0x1300	Next usable RAM space

All buffers are controlled by the SyncManager. Only buffer 0 is configured by the SyncManager and addressed by ECAT and µController.

Figure 4.1: SyncManager buffer allocation

As an example, figure 4.1 demonstrates a configuration with start address 0x1000 and Length 0x100. The other buffers shall not be read or written. Access to the buffer is always directed to addresses in the range of buffer 0.

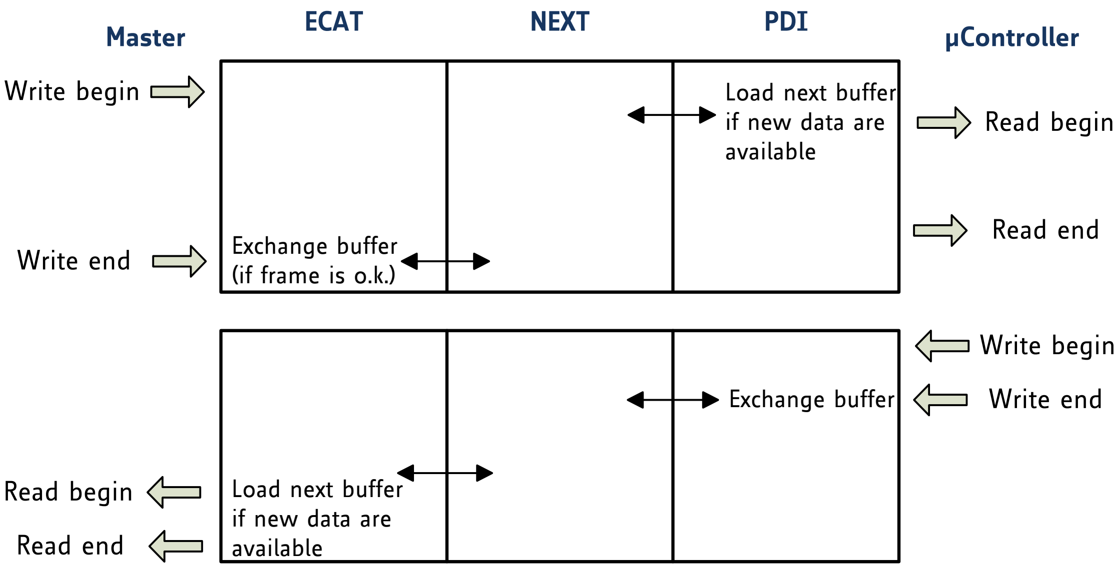


Figure 4.2: SyncManager buffered mode interaction

4.2 Mailbox mode

The mailbox mode implements a handshake mechanism for data exchange, so that no data will be lost. Each side, EtherCAT™ master or local application will get access to the buffer only after the other side has finished its access. The mailbox mode only allows alternating reading and writing. This assures all data from the producer reaches the consumer. The mailbox mode uses just one buffer of the configured size. At first, after initialization/activation, the buffer (mailbox, MBX) is writeable. Once it is written completely, write access is blocked, and the buffer can be read out by the other side. After it was completely read out, it can be written again. The time it takes to read or write the mailbox does not matter. The mailbox mode is used for the application layer protocol. Via the mailbox the motor-parameters of the TMCM-1632/TMCM-1610-KR can be written/read using the TMCL™ protocol.

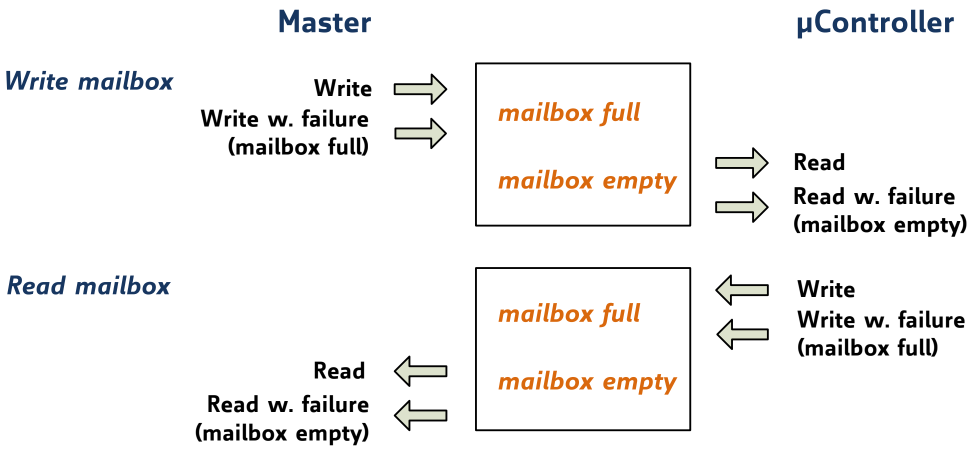


Figure 4.3: SyncManager mailbox interaction

5 EtherCAT™ slave state machine

The EtherCAT™ slave state machine has four states, which are shown in figure 5.1. After power ON the slave state machine is in the *Init state*. In this situation mailbox and process data communication is impossible. The EtherCAT™ master initializes the SyncManager channels 0 and 1 for the communication via mailbox.

While changeover from *Init state* to *Pre-Operational state* the EtherCAT™ slave checks the correct initialization of the mailbox. Afterwards mailbox communication is possible. Now, in the *Pre-Operational state* the master initializes the SyncManager channels for the process data and the FMMU channels. Furthermore adjustments are sent, which differ from the default values.

While changeover from *Pre-Operational state* to *Safe-Operational state* the EtherCAT™ slave checks the correct initialization of the SyncManager channels for the process data as well as the adjustments for the Distributed Clocks. Before accepting the change of state, the EtherCAT™ slave copies actual input data into the accordant DP-RAM array of the EtherCAT™ slave controller. In the *Safe-Operational state* mailbox and process data communication are possible, but the slave holds its outputs in a safe situation and actualizes the input data periodically.

Before the EtherCAT™ slave changes the state to *Operational* it has to transfer valid output data. In the *Operational state* the EtherCAT™ slave copies the output data from the EtherCAT™ master to its outputs. Process data communication and mailbox communication are possible now.

The *Bootstrap state* is only used for updating the firmware. This state is reachable form the *Init state*. During *Bootstrap state* mailbox communication is available over *File-Access over EtherCAT.* Beyond this mailbox communication or process data communication is not possible.

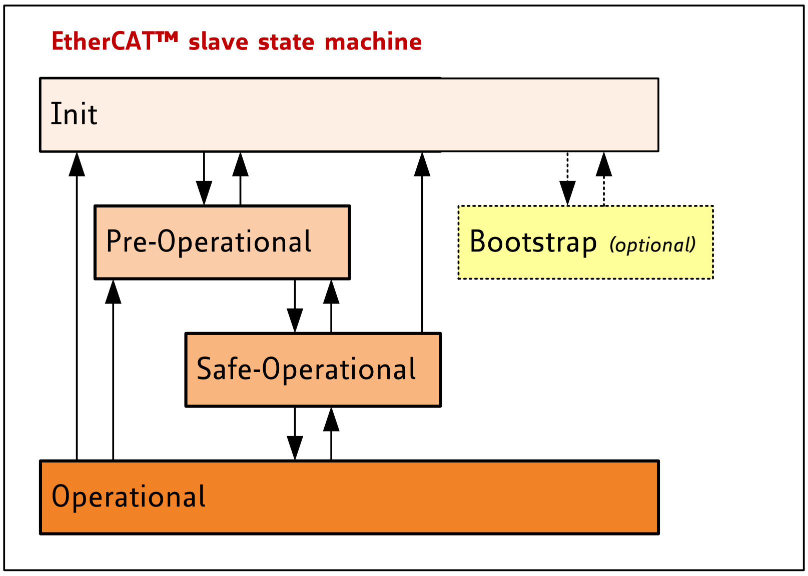


Figure 5.1: EtherCAT™ slave state machine

State / state change	Services
Init	- No communication on application layer - Master has access to the DL-information registers
Init to Pre-Operational	- Master configures registers, at least: ▪ DL address register ▪ SyncManager channels for mailbox communication - Master initializes Distributed Clock synchronization - Master requests *Pre-Operational* state ▪ Master sets AL control register - Wait for AL status register confirmation
Pre-Operational	- Mailbox communication on the application layer - No process data communication
Pre-Operational to Safe-Operational	- Master configures parameters using the mailbox: ▪ e.g., process data mapping - Master configures DL Register: ▪ SyncManager channels for process data communication ▪ FMMU channels - Master requests *Safe-Operational state* - Wait for AL Status register confirmation
Safe-Operational	- Mailbox communication on the application layer - Process data communication, but only inputs are evaluated. Outputs remain in safe state
Safe-Operational to Operational	- Master sends valid outputs - Master requests *Operational state* (AL Control/Status) - Wait for AL Status register confirmation
Operational	- Inputs and outputs are valid
Bootstrap	Recommended if firmware updates are necessary - State changes only from and to *Init* - No Process Data communication - Mailbox communication on application layer, only FoE protocol available (possibly limited file range)

<u>Three LEDs display the actual activity:</u>

LED	Description	
Link Out active	OFF	No link.
	blinking	Link and activity.
	single flash	Link without activity.
Link In active	OFF	No link.
	blinking	Link and activity.
	single flash	Link without activity.
RUN state	OFF	The device is in state *Init*.
	blinking	The device is in state *Pre-Operational*.
	single flash	The device is in state *Safe-Operational*.
	ON	The device is in state *Operational*.
	flickering (fast)	The device is in state *Bootstrap*.

6 EtherCAT™ Firmware update

For firmware updates the EtherCAT™ state machine of the slave has to be switched to *Bootstrap state*. The *file access over EtherCAT™* protocol (FoE) is used.

The two mailboxes for data transfers have the following parameters:

- Data output buffer: Start-address: 4096, length: 268 byte
- Data input buffer: Start-address: 4364, length: 40 byte

7 Process data

In standard configuration for data transfer the following buffers are used (slave view):

Data output buffer / EtherCAT™ master -> slave data transfer

Data output buffer: Start-address: 4096(0x1000), length: first 5 bytes

Start address	End address	Data type	Data value / contents
0x1000	0x1003	SIGNED32	Reference (Position, Velocity, or Current)
0x1004	0x1005	UNSIGNED8	Controller Mode 0: Motor stop 1: Positioning-Mode 2: Velocity-Mode 3: no more action 4: set position to reference 6: Current-Mode

Data input buffer / EtherCAT slave -> master data transfer

Data input buffer: Start-address: 4216(0x1078), length: first 32 bytes

Start address	End address	Data type	Data value / contents
0x1078	0x107B	SIGNED32	Actual position (32-Bit up-down counter)
0x107C	0x107F	SIGNED32	Actual current [mA]
0x1080	0x1083	SIGNED32	Actual velocity [rpm] (motor axis)
0x1084	0x1087	UNSIGNED32	Error flags Bit – Description 0 – Overcurrent 1 – Undervoltage 2 – Overvoltage 3 – Overtemperature 4 – Motor halted 5 – Hall error flag 6-8 – --- 9 – Velocity mode active 10 – Position mode active 11 – Torque mode active 12-13 – --- (not used for these modules) 14 – Position end flag 15 – Module initialized 16 – EtherCAT timeout flag (reset by SAP 158) 17 – I²t exceeded flag (SAP 29) *Flag 0 to 15 are automatically reset.*
0x1088	0x108B	SIGNED32	Target position
0x108C	0x108F	SIGNED32	Target current
0x1090	0x1093	SIGNED32	Target velocity
0x1094	0x1097	SIGNED32	Ramp generator velocity

All numbers are stored in little endian format. (least significant byte is stored at the lowest address)

8 TMCL™ mailbox

The TMCM-1632 and TMCM-1610-KR EtherCAT™ slave modules support the TMCL™ protocol in direct mode. The communication follows a strict master-slave-relationship. Via the TMCL™ mailbox motor-parameters can be read and/or written.

8.1 Binary command format

Every command has a mnemonic and a binary representation. When commands are sent from a host to a module, the binary format has to be used. Every command consists of a one-byte command field, a one-byte type field, a one-byte motor/bank field and a four-byte value field. So the binary representation of a command always has seven bytes.

Transmit an 8-byte command:

Bytes	Meaning
1	Module address: 0: drive 1: gripper
1	Command number
1	Type number
1	Motor or Bank number
4	Value (*MSB first!*)

Every time a command has been sent to a module, the module sends a reply.

Receive an 8-byte reply:

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

8.2 Status codes

The reply contains a status code.

The status code can have one of the following values:

Code	Meaning
100	Successfully executed, no error
2	Invalid command
3	Wrong type
4	Invalid value
5	Configuration EEPROM locked
6	Command not available
8	Parameter is password protected

8.3 Behavior after interrupted EtherCat™ communication

In operational mode as well as in mailbox mode every controller module checks the available communication with the EtherCAT™ master. Therefore, the time since the last master command is measured. If the last command was ago more than the timeout defined by global parameter 90, the module detects that either the operational mode is left or the EtherCAT™ cable has been unplugged. Thereby the following three situations can occur:

1. The communication is interrupted before the module is initialized. Thereby the module does nothing and stays in uninitialized state.

2. The communication is interrupted after module initialization. Then the positioning mode is activated and the actual position is used as absolute target position and will be hold.

3. The communication is interrupted during module initialization. Then the module initialization is performed and afterwards the module switches to positioning mode and holds the actual position.

Note: When the velocity ramp for the positioning mode is disabled, the position will be hold as fast as possible. When the velocity ramp is activated, the motor will slow down and drive back to the position softly.

9 Putting the TMCM-1632 into operation

Here you can find basic information for putting your TMCM-1632 module into operation. Please connect the TMCM-1610-KR module adequate and refer to its hardware manual for further information about the specific connectors.

The things you need:

- TMCM-1632
- EtherCAT™ interface suitable to your TMCM-1632 with cables
- Supply voltage +24V DC (+12… +28.5V DC) for the driver
- Supply voltage +24V DC (+12… +28.5V DC) for the logic*)
- Encoder
- BLDC motor
- EtherCAT™ master and corresponding software

*) V_{driver} and V_{logic} both accept input voltages between 12V and 28.5V DC. V_{logic} is used to supply the on-board digital logic via an on-board switching regulator whereas V_{driver} is connected to the motor driver stage. If necessary, V_{driver} can be switched off in order to disable motor driver safely while V_{logic} is still supplied in order to preserve communication and system state (e.g. encoder position). In case this option is not required, V_{driver} and V_{logic} can be tied together.

Precautions:

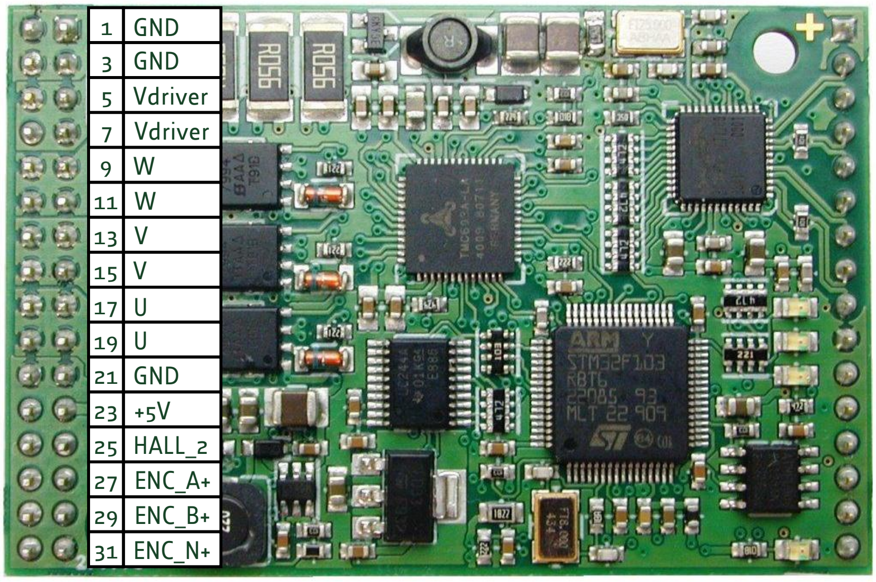
- Do not mix up connections or short-circuit pins.
- Avoid bounding I/O wires with motor power wires as this may cause noise picked up from the motor supply.
- Do not exceed the maximum power supply of 28.5V DC.
- ***Do not connect or disconnect the motor while powered!***
- ***Start with power supply OFF!***

9.1 Starting up

The following figure will show you which pins have to be used.

Motor and power connector J1

2	GND	1	GND
4	GND	3	GND
6	Vdriver	5	Vdriver
8	Vlogic	7	Vdriver
10	W	9	W
12	W	11	W
14	V	13	V
16	V	15	V
18	U	17	U
20	U	19	U
22	GND	21	GND
24	HALL_1	23	+5V
26	HALL_3	25	HALL_2
28	ENC_A-	27	ENC_A+
30	ENC_B-	29	ENC_B+
32	ENC_N-	31	ENC_N+



Communication and GPIO connector J2

1	GND
2	LINK_OUT_TX+
3	LINK_OUT_TX-
4	LINK_IN_RX+
5	LINK_IN_RX-
6	LINK_IN_TX+
7	LINK_IN_TX-
8	LINK_OUT_RX+
9	LINK_OUT_RX-
10	GND
11	TXD_TTL_3V3
12	RXD_TTL_3V3
13	IN_0
14	IN_1
15	OUT_2
16	OUT_3

Figure 9.1: Connectors of TMCM-1632

1. **Connect the communication and GPIO connector J2 as follows.**
   A single row 16 pin header with 2.54mm pitch is used for connecting all communication (EBUS and RS232) and GPIO (4 inputs and 2 outputs) signals

Pin	Label	Description
1	GND	Module ground (power supply and signal ground)
2	LINK_OUT_TX+	EBUS port 1, low voltage differential (LVDS) transmit signal (non-inverting)
3	LINK_OUT_TX-	EBUS port 1, low voltage differential (LVDS) transmit signal (inverting)
4	LINK_IN_RX+	EBUS port 0, low voltage differential (LVDS) receive signal (non-inverting)
5	LINK_IN_RX-	EBUS port 0, low voltage differential (LVDS) receive signal (inverting)
6	LINK_IN_TX+	EBUS port 0, low voltage differential (LVDS) transmit signal (non-inverting)
7	LINK_IN_TX-	EBUS port 0, low voltage differential (LVDS) transmit signal (inverting)
8	LINK_OUT_RX+	EBUS port 1, low voltage differential (LVDS) receive signal (non-inverting)
9	LINK_OUT_RX-	EBUS port 1, low voltage differential (LVDS) receive signal (inverting)
10	GND	Module ground (power supply and signal ground)
11	TXD_TTL_3V3	RS232 transmit data signal (3V3 TTL level)
12	RXD_TTL_3V3	RS232 receive data signal (3V3 TTL level, +5V signal level tolerant)
13	IN_0	General purpose input 0
14	IN_1	General purpose input 1
15	OUT_2	General purpose output 2
16	OUT_3	General purpose output 3

Table 9.1: Connector J2

2. **Connect the motor and power connector J1 as follows.**
   A double row 32 pin header with 2.54mm pitch is used for connecting all motor related signals and module power supply.

Pin	Label	Description	Pin	Label	Description
1	GND	Module ground (power supply and signal ground)	2	GND	Module ground (power supply and signal ground)
3	GND	Module ground (power supply and signal ground)	4	GND	Module ground (power supply and signal ground)
5	V_{driver}*)	Module driver supply voltage	6	V_{driver}*)	Module driver supply voltage
7	V_{logic}*)	Module logic supply voltage	8	V_{driver}*)	Module driver supply voltage
9	W	Motor coil W	10	W	Motor coil W
11	W	Motor coil W	12	W	Motor coil W
13	V	Motor coil V	14	V	Motor coil V
15	V	Motor coil V	16	V	Motor coil V
17	U	Motor coil U	18	U	Motor coil U
19	U	Motor coil U	20	U	Motor coil U
21	GND	Module ground (power supply and signal ground)	22	GND	Module ground (power supply and signal ground)
23	+5V	+5V output (100mA max.) for encoder and / or hall sensor supply	24	HALL_1	Hall sensor 1 signal input
25	HALL_2	Hall sensor 2 signal input	26	HALL_3	Hall sensor 3 signal input
27	ENC_A+	Encoder a channel input (non-inverting)	28	ENC_A-	Encoder a channel input (inverting)
29	ENC_B+	Encoder b channel input (non-inverting)	30	ENC_B-	Encoder b channel input (inverting)
31	ENC_N+	Encoder n channel input (non-inverting)	32	ENC_N-	Encoder n channel input (inverting)

Table 9.2: Connector J1

3. **Switch ON the power supply.**
   The power LED is ON now.

Figure 9.2: LEDs

If this does not occur, switch power OFF and check your connections as well as the power supply.

4. **Inform your EtherCAT™ master about his new slave module.**
   You will find all necessary data in the EEPROM of your TMCM-1632 / TMCM-1610-KR (trinamic TMCMxxx.xml). How to proceed depends on your system managing program.

5. **The EtherCAT™ master starts the initialization.**
   The paragraph *EtherCAT™ slave state machine* (5) informs you about the data flow during the initialization. At the end of the initialization the EtherCAT™ slave enters the *Operational state*. It is possible to use the mailbox from the *Pre-Operational state* on.

   The EtherCAT™ LEDs of the module show the states of the state machine and the communication activity. Please refer to paragraph 5 for further information about denotations of LED flash signals.

6. **Configure your module as follows:**
   It is necessary to configure every module, because after updating to a new firmware version, all axis parameter will be set during the first start of the module to the factory default values contained in the update file. Previously set parameter will be overwritten automatically if the version numbers of the existing and the new version are different.

Command	Description
SAP4	Maximum target velocity.
SAP6	Maximum target motor current.
SAP8	Threshold velocity for velocity regulation to switch between first and second velocity PID parameter set.
SAP11	Acceleration
SAP12	Threshold velocity for position regulation to switch between first and second position PID parameter set.
SAP130	P parameter of position PID regulator (first parameter set)
SAP131	I parameter of position PID regulator (first parameter set)
SAP132	D parameter of position PID regulator (first parameter set)
SAP133	PID regulation loop multiplier.
SAP135	I-Clipping parameter of position PID regulator (first parameter set)

Command	Description
SAP140	P parameter of velocity PID regulator (first parameter set, used at lower velocity)
SAP141	I parameter of velocity PID regulator (first parameter set, used at lower velocity)
SAP142	D parameter of velocity PID regulator (first parameter set, used at lower velocity)
SAP143	I-Clipping parameter of velocity PID regulator. (first parameter set, used at lower velocity)
SAP146	Activate velocity ramp for velocity and position mode. (Allows usage of defined acceleration for MVP and ROR/ROL command)
SAP230	P parameter of position PID regulator (second parameter set)
SAP231	I parameter of position PID regulator (second parameter set)
SAP232	D parameter of position PID regulator (second parameter set)
SAP233	I-Clipping parameter of position PID regulator (second parameter set) (A too high value causes overshooting at positioning mode.)
SAP234	P parameter of velocity PID regulator (second parameter set)
SAP235	I parameter of velocity PID regulator (second parameter set)
SAP236	D parameter of velocity PID regulator (second parameter set)
SAP237	I-Clipping Parameter for velocity PID regulator (second parameter set)

CAUTION: The default values of the following parameters must not be modified:

Command	Description
SAP159	Commutation mode: 7 (FOC encoder based)
SAP249	Encoder init mode: 1 (initialization based on hall sensor signals)
SAP250	4000 encoder steps per rotation
SAP251	Reversion of the encoder direction active: 1
SAP253	Number of motor poles: 16
SAP254	Hall sensor inverted: 1

Note, that the table above can help you configuring your module. Please refer to paragraph 11.5 for further information about using the command set axis parameter (SAP) *and specific values for each parameter.*

7. **Parameterize the TMCM-1632/TMCM-1610-KR controller for initial commissioning:**
To protect parameters like motor poles and max target current against overwriting by default values included in a new firmware version, some protected parameters are automatically overwritten by global parameters, which are not influenced by further firmware updates. So for example axis parameter 253 (motor poles) will be overwritten by global parameter 0 of bank 2 and axis parameter 6 (max target current) will be overwritten by global parameter 1 of bank 2 on power on.

Therefore, at initial commissioning of a module, these global parameters have to be set once as follows:

Set default motor poles: (global parameter 0 of bank 2)

GGP:	10, 0, 2, -	(read actual value)
SGP:	9, 0, 2, 16	(set default value (8 for axis 5))
STGP:	11, 0, 2, -	(save default value)

Set default max target current: (global parameter 1 of bank 2)

GGP:	10, 1, 2, -	(read actual value)
SGP:	9, 1, 2, 4000	(set default value (1000 for axis 5,4))
STGP:	11, 1, 2, -	(save default value)

Note: Global parameter 0 to 15 are password protected. To change these parameters it is necessary to approve them with the right password!

The minute counter and the I^2t counter are also stored in an EEPROM area which is not affected by further firmware updates. Therefore these counters have to be set to zero at initial module commissioning as follows:

Set minute counter: (axis parameter 30)

GAP:	6, 30, 0, -	(read actual value)
SAP:	5, 30, 0, value	(set initial value)
STAP:	7, 30, 0, -	(save value)

Set I^2t counter: (axis parameter 28)

GAP:	6, 28, 0, -	(read actual value)
SAP:	5, 28, 0, value	(set initial value)
STAP:	7, 28, 0, -	(save value)

Note: Axis parameter 28 and 30 are password protected. To change these parameters it is necessary to approve them with the right password!

10 TMCL™ command overview

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

10.1 Motion commands

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

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

10.2 Parameter commands

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

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

10.3 List of commands

The following commands are currently supported:

Mnemonic	Number	Parameter	Description
ROR	1	<motor number>, <velocity>	Rotate right with specified velocity
ROL	2	<motor number>, <velocity>	Rotate left with specified velocity
MST	3	<motor number>	Stop motor movement
MVP	4	ABS|REL, <motor number>, <position|offset>	Move to position (absolute or relative)
SAP	5	<parameter>, <motor number>, <value>	Set axis parameter (motion control specific settings)
GAP	6	<parameter>, <motor number>	Get axis parameter (read out motion control specific settings)
STAP	7	<parameter>, <motor number>	Store axis parameter permanently (non volatile)
RSAP	8	<parameter>; <motor number>	Restore axis parameter

11 Commands

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

11.1 ROR (rotate right)

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

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

The module is based on the ARM Cortex-M3 microcontroller and high performance pre-driver TMC603A. This makes possible choosing a velocity between 0 and 2047.

Related commands: ROL, MST, SAP, GAP

Mnemonic: ROR 0, <velocity>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
1	(don't care)	0	<velocity> 0… 2047

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	1	(don't care)

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

Binary:

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

11.2 ROL (rotate left)

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

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

The module is based on the ARM Cortex-M3 microcontroller and high performance pre-driver TMC603A. This makes possible choosing a velocity between 0 and 2047.

Related commands: ROR, MST, SAP, GAP

Mnemonic: ROL 0, <velocity>

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
2	(don't care)	0	<velocity> 0… 2047

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	2	(don't care)

Example:

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

Binary:

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

11.3 MST (motor stop)

With this command the motor will be instructed to stop with deceleration ramp (soft stop).

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

Related commands: ROL, ROR, SAP, GAP

Mnemonic: MST 0

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	3	(don't care)

Example:

Stop motor
Mnemonic: MST 0

Binary:

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

11.4 MVP (move to position)

With this command the motor will be instructed to move to a specified relative or absolute position. It will use the acceleration/deceleration ramp and the max target velocity programmed into the unit. This command is non-blocking (like all commands) – that is, a reply will be sent immediately after command interpretation and initialization of the motion controller. Further commands may follow without waiting for the motor reaching its end position. The maximum target velocity and acceleration are defined by axis parameters #4 and #11.

Two operation types are available:

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

Please note, that the distance between the actual position and the new one should not be more than 2147483647 microsteps. Otherwise the motor will run in the wrong direction for taking a shorter way. If the value is exactly 2147483648 the motor maybe stops.

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

Related commands: SAP, GAP, and MST

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

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
4	0 ABS – absolute	0	<position>
	1 REL – relative	0	<offset>

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	4	(don't care)

Example MVP ABS:

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

Binary:

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

Example MVP REL:

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

Binary:

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

11.5 SAP (set axis parameter)

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

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, and RSAP

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

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	5	(don't care)

Please note, that for the binary representation <parameter number> has to be filled with the **number** ***and the <value> has to be filled with a value from*** **range*****.***

A list of all parameters which can be used for the SAP command is shown in section 11.9.

Example:

Set the maximum target current to 200mA
Mnemonic: SAP 6, 0, 200

Binary:

Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$00	$05	$06	$00	$00	$00	$00	$c8

11.6 GAP (get axis parameter)

Most parameters of the TMCM-1632/TMCM-1610-KR can be adjusted individually. With these parameters they can be read out (the value read is only output).

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, and RSAP

Mnemonic: GAP <parameter number>, 0

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	6	(don't care)

A list of all parameters which can be used for the GAP command is shown in section 11.9.

Example:

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

Binary:

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

Reply:

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

⇨ **status=no error, position=711**

11.7 STAP (store axis parameter)

An axis parameter previously set with a *Set Axis Parameter command (SAP)* will be stored permanent. Most parameters are automatically restored after power up (refer to axis parameter list in chapter 11.9).

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

Related commands: SAP, RSAP, and GAP

Mnemonic: STAP <parameter number>, 0

Binary representation:

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

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	7	(don't care)

A list of all parameters which can be used for the STAP command is shown in section 11.9.

Example:

Store the maximum target velocity
Mnemonic: STAP 4, 0

Binary:

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

Note: The STAP command will not have any effect when the configuration EEPROM is locked. The error code 5 (configuration EEPROM locked, see also section 8.2) will be returned in this case.

11.8 RSAP (restore axis parameter)

For all configuration related axis parameters non-volatile memory locations are provided. By default, most parameters are automatically restored after power up (refer to axis parameter list in chapter 11.9). A single parameter that has been changed before can be reset by this instruction also.

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

Relate commands: SAP, STAP, and GAP

Mnemonic: RSAP <parameter number>, 0

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	8	(don't care)

A list of all parameters which can be used for the RSAP command is shown in section 11.9.

Example:

Restore the maximum target current
Mnemonic: RSAP 6, 0

Binary:

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

11.9 Axis parameter overview (SAP, GAP, STAP, RSAP)

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

Meaning of the letters in column Access:

R = readable (GAP)
W = writable (SAP)
E = stored and automatically restored from EEPROM after reset or power-on
P = protected

Number	Axis Parameter	Description	Range [Unit]	Access
0	Target position	The target position of a currently executed ramp.	-2147483648… +2147483647	RW
1	Actual position	Set/get the position counter without moving the motor.	-2147483648… +2147483647	RW
2	Target velocity	Set/get the desired target velocity.	-200000…+200000 [rpm]	RW
3	Actual velocity	The actual velocity of the motor.	-2147483648… +2147483647 [rpm]	R
4	Max target velocity	The maximum velocity used for velocity ramp in velocity mode and positioning mode. Set this value to a realistic velocity which the motor can reach!	0…200000 [rpm]	RWE
6	Max target current	Set/get the max target motor current. *This parameter is only limiting the target value (set point) of the current, not its actual value. Depending on the PID control parameters the actual current can exceed this current limit. See chapter 12.3 for details.	0…20000 [mA]	RWEP
7	MVP Target reached velocity	Maximum velocity at which end position can be set. Prevents issuing of end position when the target is passed at high velocity.	0…200000 [rpm]	RWE
8	Threshold velocity for velocity PID	Threshold velocity for velocity regulation to switch between first and second velocity PID parameter set.	0…200000 [rpm]	RWE
9	Motor halted velocity	If the actual velocity is below this value the motor halted flag will be set.	0…200000 [rpm]	RWEP
10	MVP target reached distance	Maximum distance at which the position end flag is set.	0…100000	RWE
11	Acceleration	Acceleration parameter for ROL, ROR, and the velocity ramp of MVP.	0…100000 [rpm/s]	RWE
12	Threshold velocity for position PID	Threshold velocity for position regulation to switch between first and second position PID parameter set.	0…200000 [rpm]	RWE
13	Ramp generator velocity	The actual velocity of the velocity ramp used for positioning and velocity mode.	-2147483648… +2147483647 [rpm]	R
14	velocity threshold for hallFX™	Velocity to switch from controlled to hallFX™ mode. Set this value to a realistic velocity which the motor can reach in controlled mode!	0…200000 [rpm]	RWE
25	Thermal winding time constant	Thermal winding time constant for the used motor. Used for I^2t monitoring.	0…4294967295 [ms]	RWEP
26	I^2t limit	An actual I^2t sum that exceeds this limit leads to increasing the I^2t exceed counter once a second.	0…4294967295	RWEP
27	I^2t sum	Actual sum of the I^2t monitor.	0…4294967295	R

Number	Axis Parameter	Description	Range [Unit]	Access
28	I²t exceed counter	Counts how often an I²t sum was higher than the I²t limit.	0…4294967295	RWEP
29	Clear I²t exceeded flag	Clear the flag that indicates that the I²t sum has exceeded the I²t limit.	(ignored)	W
30	Minute counter	Counts the module operational time in minutes.	0…4294967295 [min]	RWEP
130	P parameter for position PID (I)	P parameter of position PID regulator (first parameter set)	0…65535	RWE
131	I parameter for position PID (I)	I parameter of position PID regulator (first parameter set)	0…65535	RWE
132	D parameter for position PID (I)	D parameter of position PID regulator (first parameter set)	0…65535	RWE
133	PID regulation loop multiplier	Delay multiplier for velocity/positioning PID calculation algorithm.	0…10	RWEP
134	Current regulation loop multiplier	Delay multiplier for current limitation algorithm / PID current regulator.	0…10	RWEP
135	I-Clipping parameter for position PID (I)	I-Clipping parameter of position PID regulator (first parameter set) (A too high value causes overshooting at positioning mode.)	0…1000	RWE
140	P parameter for velocity PID (I)	P parameter of velocity PID regulator (first parameter set, used at lower velocity)	0…65535	RWE
141	I parameter for velocity PID (I)	I parameter of velocity PID regulator (first parameter set, used at lower velocity)	0…65535	RWE
142	D parameter for velocity PID (I)	D parameter of velocity PID regulator (first parameter set, used at lower velocity)	0…65535	RWE
143	I-Clipping parameter for velocity PID (I)	I-Clipping parameter of velocity PID (first parameter set, used at lower velocity)	0…1000	RWE
146	Activate ramp	1: Activate velocity ramp generator. Allows usage of acceleration and max velocity for ROL, ROR, and MVP command.	0/1	RWE
150	Actual motor current	Get actual motor current.	-2147483648… +2147483647 [mA]	R
151	Actual voltage	Actual supply voltage.	0…4294967295	R
152	Actual driver temperature	Actual temperature of the motor driver.	0…4294967295	R
155	Target current	Get desired target current or set target current to activate current regulation mode. (+= turn motor in right direction; -= turn motor in left direction)	-20000…+20000 [mA]	RW

Number	Axis Parameter	Description	Range [Unit]	Access
156	Error/Status flags	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: unused Bit 7: unused Bit 8: unused Bit 9: Velocity mode active flag Bit 10: Position mode active flag. Bit 11: Torque mode active flag. Bit 12: unused Bit 13: unused Bit 14: Position end flag. This flag is set if the motor has been stopped at the target position. Bit 15: Module initialized flag. This flag is set if the module is initialized properly. Bit 16: EtherCAT timeout flag. This flag is set if the communication between EtherCAT master and axis controller has been interrupted. (reset by SAP 158) Bit 17: I^2t exceeded flag. This flag is set if the I^2t sum exceeded the I^2t limit of the motor. (reset by SAP 29 after the time specified by the I^2t thermal winding time constant) *Flag 0 to 15 are automatically reset. Only flag 16 and 17 must be cleared manually.*	0…4294967295	R
157	Module supply current	Get actual supply current of the module. (only TMCM-1632 since hardware version v2.1)	0…4294967295 [mA]	R
158	Clear EtherCAT timeout flag	Clear the flag that indicates a communication timeout between the EtherCAT master and the controller.	(ignored)	W
159	Commutation mode	6: FOC with hall sensors 7: FOC with encoder	6/7	RWEP
161	Encoder set NULL	1: set position counter to zero at next N channel event.	0/1	RWEP
162	Switch set NULL	1: set position counter to zero at next switch event.	0/1	RWEP
163	Encoder clear set NULL	1: set position counter to zero only once 0: always at an N channel event, respectively switch event.	0/1	RWEP

<table>
<tr><th>Number</th><th>Axis Parameter</th><th colspan="3">Description</th><th>Range [Unit]</th><th>Access</th></tr>
<tr><td rowspan="3">164</td><td rowspan="3">Activate stop switch</td><td>Bit 0</td><td>Left stop switch enable</td><td>When this bit is set the motor will be stopped if it is moving in negative direction and the left stop switch input becomes active</td><td rowspan="3">0… 3</td><td rowspan="3">RWEP</td></tr>
<tr><td>Bit 1</td><td>Right stop switch enable</td><td>When this bit is set the motor will be stopped if it is moving in positive direction and the right stop switch input becomes active</td></tr>
<tr><td colspan="3">Please see parameter 166 for selecting the stop switch input polarity.</td></tr>
<tr><td>165</td><td>Actual encoder commutation offset</td><td colspan="3">This value represents the internal commutation offset.
(0 … max. encoder steps per rotation)</td><td>0…65535</td><td>RWEP</td></tr>
<tr><td rowspan="2">166</td><td rowspan="2">Stop switch polarity</td><td>Bit 0</td><td>Left stop switch polarity</td><td>Bit set: Left stop switch input is high active
Bit clear: Left stop switch input is low active</td><td rowspan="2">0…3</td><td rowspan="2">RWEP</td></tr>
<tr><td>Bit 1</td><td>Right stop switch polarity</td><td>Bit set: Right stop switch input is high active
Bit clear: Right stop switch input is low active</td></tr>
<tr><td>172</td><td>P parameter for current PID</td><td colspan="3">P parameter of current PID regulator.</td><td>0…65535</td><td>RWE</td></tr>
<tr><td>173</td><td>I parameter for current PID</td><td colspan="3">I parameter of current PID regulator.</td><td>0…65535</td><td>RWE</td></tr>
<tr><td>174</td><td>D parameter for current PID</td><td colspan="3">D parameter of current PID regulator.</td><td>0…65535</td><td>RWE</td></tr>
<tr><td>175</td><td>I-Clipping parameter for current PID</td><td colspan="3">I-Clipping parameter of current PID regulator.</td><td>0…1000</td><td>RWE</td></tr>
<tr><td>177</td><td>Start current</td><td colspan="3">Motor current for controlled commutation. This parameter is used in commutation mode 7 only when init sine mode = 0 or 2.</td><td>0…20000 [mA]</td><td>RWEP</td></tr>
<tr><td>200</td><td>Current PID error</td><td colspan="3">Actual error of current PID regulator</td><td>-2147483648…
+2147483647</td><td>R</td></tr>
<tr><td>201</td><td>Current PID error sum</td><td colspan="3">Sum of errors of current PID regulator</td><td>-2147483648…
+2147483647</td><td>R</td></tr>
<tr><td>209</td><td>Actual encoder position</td><td colspan="3">Actual encoder position / counter value</td><td>-2147483648…
+2147483647</td><td>R</td></tr>
<tr><td>226</td><td>Position PID error</td><td colspan="3">Actual error of position PID regulator</td><td>-2147483648…
+2147483647</td><td>R</td></tr>
<tr><td>227</td><td>Position PID error sum</td><td colspan="3">Sum of errors of position PID regulator</td><td>-2147483648…
+2147483647</td><td>R</td></tr>
<tr><td>228</td><td>Velocity PID error</td><td colspan="3">Actual error of velocity PID regulator</td><td>-2147483648…
+2147483647</td><td>R</td></tr>
<tr><td>229</td><td>Velocity PID error sum</td><td colspan="3">Sum of errors of velocity PID regulator</td><td>-2147483648…
+2147483647</td><td>R</td></tr>
</table>

Number	Axis Parameter	Description	Range [Unit]	Access
230	P parameter for position PID (II)	P parameter of position PID regulator. (second parameter set)	0…65535	RWE
231	I parameter for position PID (II)	I parameter of position PID regulator. (second parameter set)	0…65535	RWE
232	D parameter for position PID (II)	D parameter of position PID regulator. (second parameter set)	0…65535	RWE
233	I-Clipping parameter for position PID (II)	I-Clipping parameter of position PID regulator. (second parameter set) (A too high value causes overshooting at positioning mode.)	0…1000	RWE
234	P parameter for velocity PID (II)	P parameter of velocity PID regulator. (second parameter set)	0…65535	RWE
235	I parameter for velocity PID (II)	I parameter of velocity PID regulator. (second parameter set)	0…65535	RWE
236	D parameter for velocity PID (II)	D parameter of velocity PID regulator. (second parameter set)	0…65535	RWE
237	I-Clipping parameter for velocity PID (II)	I-Clipping parameter of velocity PID regulator. (second parameter set, used at higher velocity)	0…1000	RWE
238	Mass inertia constant	Mass inertia constant. Compensates the rotor inertia of the motor.	0…65535	RWEP
239	BEMF constant	BEMF constant of the motor. Feed forward control for current, position, and velocity regulation is disabled if BEMF constant is set to zero.	0…65535 [rpm/(10V)]	RWEP
240	Motor coil resistance	Resistance of motor coil.	0…65535 [mΩ]	RWEP
241	Init sine velocity	Velocity for sine initialization. A positive sign initializes in right direction, a negative sign in left motor direction.	-32767… +32767 [rpm]	RWEP
244	Init sine delay	Duration for sine initialization sequence. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	0…10000 [ms]	RWEP
245	Overvoltage protection	1: Enable overvoltage protection.	0/1	RWEP
249	Init sine mode	0: Initialization in controlled sine commutation (determines the encoder offset) 1: Initialization in block commutation using hall sensors 2: Initialization in controlled sine commutation (use the previous set encoder offset)	0…2	RWEP
250	Encoder steps	Encoder steps per rotation.	0…65535	RWEP
251	Encoder direction	Set the encoder direction in a way, that ROR increases position counter.	0/1	RWEP
253	Number of motor poles	Number of motor poles.	2…254	RWEP
254	Hall sensor invert	1: Hall sensor invert. Invert the hall scheme, e.g. used by some Maxon motors.	0/1	RWEP

11.9.1 Axis parameter sorted by functionality

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

Meaning of the letters in column Access:

R = readable (GAP)
W = writable (SAP)
E = stored and automatically restored from EEPROM after reset or power-on
P = protected

Number	Axis Parameter	Description	Range [Unit]	Access
		Motor/module settings		
253	Number of motor poles	Number of motor poles.	2…254	RWEP
239	BEMF constant	BEMF constant of the motor. Feed forward control for current, position, and velocity regulation is disabled if BEMF constant is set to zero.	0…65535 [rpm/(10V)]	RWEP
240	Motor coil resistance	Resistance of motor coil.	0…65535 [mΩ]	RWEP
238	Mass inertia constant	Mass inertia constant. Compensates the rotor inertia of the motor.	0…65535	RWEP
25	Thermal winding time constant	Thermal winding time constant for the used motor. Used for I^2t monitoring.	0…4294967295 [ms]	RWEP
26	I^2t limit	An actual I^2t sum that exceeds this limit leads to increasing the I^2t exceed counter once a second.	0…4294967295	RWEP
27	I^2t sum	Actual sum of the I^2t monitor.	0…4294967295	R
28	I^2t exceed counter	Counts how often an I^2t sum was higher than the I^2t limit.	0…4294967295	RWEP
29	Clear I^2t exceeded flag	Clear the flag that indicates that the I^2t sum has exceeded the I^2t limit.	(ignored)	W
30	Minute counter	Counts the module operational time in minutes.	0…4294967295 [min]	RWEP
158	Clear EtherCAT timeout flag	Clear the flag that indicates a communication timeout between the EtherCAT master and the controller.	(ignored)	W
245	Overvoltage protection	1: Enable overvoltage protection.	0/1	RWEP
		Encoder/initialization settings		
254	Hall sensor invert	1: Hall sensor invert. Invert the hall scheme, e.g. used by some Maxon motors.	0/1	RWEP
250	Encoder steps	Encoder steps per rotation.	0…65535	RWEP
209	Actual encoder position	Actual encoder position / counter value	-2147483648… +2147483647	R
251	Encoder direction	Set the encoder direction in a way, that ROR increases position counter.	0/1	RWEP
165	Actual encoder commutation offset	This value represents the internal commutation offset. (0 … max. encoder steps per rotation)	0…65535	RWEP
177	Start current	Motor current for controlled commutation. This parameter is used in commutation mode 7 only when init sine mode = 0 or 2.	0…20000 [mA]	RWEP

Number	Axis Parameter	Description	Range [Unit]	Access
249	Init sine mode	0: Initialization in controlled sine commutation (determines the encoder offset) 1: Initialization in block commutation using hall sensors 2: Initialization in controlled sine commutation (use the previous set encoder offset)	0…2	RWEP
241	Init sine velocity	Velocity for sine initialization. A positive sign initializes in right direction, a negative sign in left motor direction.	-32767… +32767 [rpm]	RWEP
244	Init sine delay	Duration for sine initialization sequence. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	0…10000 [ms]	RWEP
14	velocity threshold for hallFX™	Velocity to switch from controlled to hallFX™ mode. Set this value to a realistic velocity which the motor can reach in controlled mode!	0…200000 [rpm]	RWE
159	Commutation mode	6: FOC with hall sensors 7: FOC with encoder	6/7	RWEP
		Torque regulation mode		
6	Max target current	Set/get the max target motor current. *This parameter is only limiting the target value (set point) of the current, not its actual value. Depending on the PID control parameters the actual current can exceed this current limit. See chapter 12.3 for details.	0…20000 [mA]	RWEP
150	Actual motor current	Get actual motor current.	-2147483648… +2147483647 [mA]	R
155	Target current	Get desired target current or set target current to activate current regulation mode. (+= turn motor in right direction; -= turn motor in left direction)	-20000…+20000 [mA]	RW
134	Current regulation loop multiplier	Delay multiplier of current limitation algorithm / PID current regulator.	0…10	RWEP
172	P parameter for current PID	P parameter of current PID regulator.	0…65535	RWE
173	I parameter for current PID	I parameter of current PID regulator.	0…65535	RWE
174	D parameter for current PID	D parameter of current PID regulator.	0…65535	RWE
175	I-Clipping parameter for current PID	I-Clipping parameter of current PID regulator.	0…1000	RWE
200	Current PID error	Actual error of current PID regulator	-2147483648… +2147483647	R
201	Current PID error sum	Sum of errors of current PID regulator	-2147483648… +2147483647	R
		Velocity regulation mode		
3	Actual velocity	The actual velocity of the motor.	-2147483648… +2147483647 rpm]	R
2	Target velocity	Set/get the desired target velocity.	-200000…+200000 [rpm]	RW
9	Motor halted velocity	If the actual velocity is below this value the motor halted flag will be set.	0…200000 [rpm]	RWEP

Number	Axis Parameter	Description	Range [Unit]	Access
133	PID regulation loop multiplier	Delay multiplier for velocity/positioning PID calculation algorithm.	0…10	RWEP
8	Threshold velocity for velocity PID	Threshold velocity for velocity regulation to switch between first and second velocity PID parameter set.	0…200000 [rpm]	RWE
140	P parameter for velocity PID (I)	P parameter of velocity PID regulator (first parameter set, used at lower velocity)	0…65535	RWE
141	I parameter for velocity PID (I)	I parameter of velocity PID regulator (first parameter set, used at lower velocity)	0…65535	RWE
142	D parameter for velocity PID (I)	D parameter of velocity PID regulator (first parameter set, used at lower velocity)	0…65535	RWE
143	I-Clipping parameter for velocity PID (I)	I-Clipping parameter of velocity PID (first parameter set, used at lower velocity)	0…1000	RWE
234	P parameter for velocity PID (II)	P parameter of velocity PID regulator. (second parameter set)	0…65535	RWE
235	I parameter for velocity PID (II)	I parameter of velocity PID regulator. (second parameter set)	0…65535	RWE
236	D parameter for velocity PID (II)	D parameter of velocity PID regulator. (second parameter set)	0…65535	RWE
237	I-Clipping parameter for velocity PID (II)	I-Clipping parameter of velocity PID regulator. (second parameter set, used at higher velocity)	0…1000	RWE
228	Velocity PID error	Actual error of PID velocity regulator	-2147483648… +2147483647	R
229	Velocity PID error sum	Sum of errors of PID velocity regulator	-2147483648… +2147483647	R
		Velocity ramp parameter		
4	Max target velocity	The maximum velocity used for velocity ramp in velocity mode and positioning mode. Set this value to a realistic velocity which the motor can reach!	0…200000 [rpm]	RWE
11	Acceleration	Acceleration parameter for ROL, ROR, and the velocity ramp of MVP.	0…100000 [rpm/s]	RWE
13	Ramp generator velocity	The actual velocity of the velocity ramp used for positioning and velocity mode.	-2147483648… +2147483647 [rpm]	R
146	Activate ramp	1: Activate velocity ramp generator for position PID control. Allows usage of acceleration and positioning velocity for MVP command.	0/1	RWE
		Position regulation mode		
1	Actual position	Set/get the position counter without moving the motor.	-2147483648… +2147483647	RW
0	Target position	The target position of a currently executed ramp.	-2147483648… +2147483647	RW
7	MVP Target reached velocity	Maximum velocity at which end position can be set. Prevents issuing of end position when the target is passed at high velocity.	0…200000 [rpm]	RWE
10	MVP target reached distance	Maximum distance at which the position end flag is set.	0…100000	RWE
161	Encoder set NULL	1: set position counter to zero at next N channel event.	0/1	RWEP

Number	Axis Parameter	Description	Range [Unit]	Access
162	Switch set NULL	1: set position counter to zero at next switch event.	0/1	RWEP
163	Encoder clear set NULL	1: set position counter to zero only once 0: always at an N channel event, respectively switch event.	0/1	RWEP
12	Threshold velocity for position PID	Threshold velocity for position regulation to switch between first and second position PID parameter set.	0…200000 [rpm]	RWE
130	P parameter for position PID (I)	P parameter of position PID regulator (first parameter set)	0…65535	RWE
131	I parameter for position PID (I)	I parameter of position PID regulator (first parameter set)	0…65535	RWE
132	D parameter for position PID (I)	D parameter of position PID regulator (first parameter set)	0…65535	RWE
135	I-Clipping parameter for position PID (I)	I-Clipping parameter of position PID regulator (first parameter set) (A too high value causes overshooting at positioning mode.)	0…1000	RWE
230	P parameter for position PID (II)	P parameter of position PID regulator. (second parameter set)	0…65535	RWE
231	I parameter for position PID (II)	I parameter of position PID regulator. (second parameter set)	0…65535	RWE
232	D parameter for position PID (II)	D parameter of position PID regulator. (second parameter set)	0…65535	RWE
233	I-Clipping parameter for position PID (II)	I-Clipping parameter of position PID regulator. (second parameter set) (A too high value causes overshooting at positioning mode.)	0…1000	RWE
226	Position PID error	Actual error of PID position regulator	-2147483648… +2147483647	R
227	Position PID error sum	Sum of errors of PID position regulator	-2147483648… +2147483647	R
		Status information		
151	Actual voltage	Actual supply voltage.	0…4294967295	R
152	Actual driver temperature	Actual temperature of the motor driver.	0…4294967295	R
156	Error/Status flags	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: unused Bit 7: unused Bit 8: unused Bit 9: Velocity mode active flag Bit 10: Position mode active flag. Bit 11: Torque mode active flag.	0…4294967295	R

Number	Axis Parameter	Description	Range [Unit]	Access
		Bit 12: unused Bit 13: unused Bit 14: Position end flag. This flag is set if the motor has been stopped at the target position. Bit 15: Module initialized flag. This flag is set if the module is initialized properly. Bit 16: EtherCAT timeout flag. This flag is set if the communication between EtherCAT master and axis controller has been interrupted. (reset by SAP 158) Bit 17: I²t exceeded flag. This flag is set if the I²t sum exceeded the I²t limit of the motor. (reset by SAP 29 after the time specified by the I²t thermal winding time constant) *Flag 0 to 15 are automatically reset. Only flag 16 and 17 must be cleared manually.*		
157	Module supply current	Get actual supply current of the module. (only TMCM-1632 since hardware version v2.1)	0…4294967295 [mA]	R
Stop switch parameters				
164	Activate stop switch	Bit 0 – Left stop switch enable – When this bit is set the motor will be stopped if it is moving in negative direction and the left stop switch input becomes active Bit 1 – Right stop switch enable – When this bit is set the motor will be stopped if it is moving in positive direction and the right stop switch input becomes active Please see parameter 166 for selecting the stop switch input polarity.	0…3	RWEP
166	Stop switch polarity	Bit 0 – Left stop switch polarity – Bit set: Left stop switch input is high active; Bit clear: Left stop switch input is low active Bit 1 – Right stop switch polarity – Bit set: Right stop switch input is high active; Bit clear: Right stop switch input is low active	0…3	RWEP

11.9.2 Protected axis parameters

Some axis parameters are write-protected to prevent the normal user to set values whose changes could damage the mechanics of the robot. These parameters are marked with a "P" in the access-column. To change these axis parameters, the manufacturer can use parameter 248 with the right password to approve all axis parameters. This activation can be reversed by using parameter 248 with a wrong password or reboot the axis controller. If a user tries to set a protected parameter the reply status of the TMCL command is "REPLY_WRITE_PROTECTED" (8).

11.10 SGP (set global parameter)

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, and only bank 2 is intended to use for user variables. Parameter 0 to 15 of bank 2 are password protected. Parameter 16 to 55 can be used for user variables without password protection.

Related commands: GGP, STGP, RSGP

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

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	(don't care)

Example: set variable 0 at bank 2 to 100
Mnemonic: SGP, 0, 2, 100

Binary:

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

11.11 GGP (get global parameter)

All global parameters can be read with this function.

Related commands: SGP, STGP, RSGP

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

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	<value>

Example: get variable 0 from bank 2
Mnemonic: GGP, 0, 2

Binary:

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

11.12 STGP (store global parameter)

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

Related commands: SGP, GGP, RSGP

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

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	(don't care)

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

Binary:

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

11.13 RSGP (restore global parameter)

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

Related commands: SGP, GGP, STGP

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

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	(don't care)

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

Binary:

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

11.14 Example usage of SGP, GGP, STGP, RSGP

Note: Parameter bank 0 and 1 are reserved for global module parameters. Only bank 2 is intended to use for user variables. (Parameter 0 to 15 of bank 2 are password protected. Parameter 16 to 55 can be used for user variables without password protection.)

Scenario:

a) Store e.g. a homing position (10.000) in variable 16 of bank 2
SGP (9), 16, 2, 10.000
b) Copy the value from RAM location to EEPROM
STGP (11), 16, 2, 0
c) Reboot module
d) Read variable 16 of bank 2
GGP (10), 16, 2 → returns the value (10.000)
e) Change the variable 16 of bank 2
SGP (9), 16, 2, 5.000
f) Overwrite the value in the RAM with the EEPROM value
RSGP (12), 16, 2
g) Read variable 16 of bank 2
GGP (10), 16, 2 → returns the value (10.000)

11.15 Global parameter overview (SGP, GGP, STGP, RSGP)

The following section describes all global parameters that can be used with the SGP, GGP, STGP and RSGP commands on bank 0.

Meaning of the letters in column Access:

R = readable (GGP)
W = writable (SGP)
E = automatically restored from EEPROM after reset or power-on
P = protected

Number	Axis Parameter	Description	Range [Unit]	Access
90	EtherCAT timeout	SGP/GGP: Set/Get Timeout to determine an interrupted communication with the EtherCAT master. (automatically stored in EEProm)	0… +4294967295 [ms]	RWEP

11.16 TMCL control functions

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

The parameter 136 has a special reply format for the string representation. There is no checksum in the reply format. The reply is structured as follows:

Byte index	Contents
1	Host address
2..9	Version string (8 characters, e.g. "1632V200")

The binary format of parameter 136 uses the normal TMCL reply format with the value field assigned as follows:

Byte index in value field	Contents
1	Version number, low byte
2	Version number, high byte
3	Type number, low byte (currently not used)
4	Type number, high byte (currently not used)

12 PID regulation

12.1 Structure of the cascaded motor regulation modes

The TMCM-1632/TMCM-1610-KR support a current, velocity, and position PID regulation mode for motor control in different application areas. These regulation modes are cascaded as shown in figure 12.1. The individual modes are explained in the following sections.

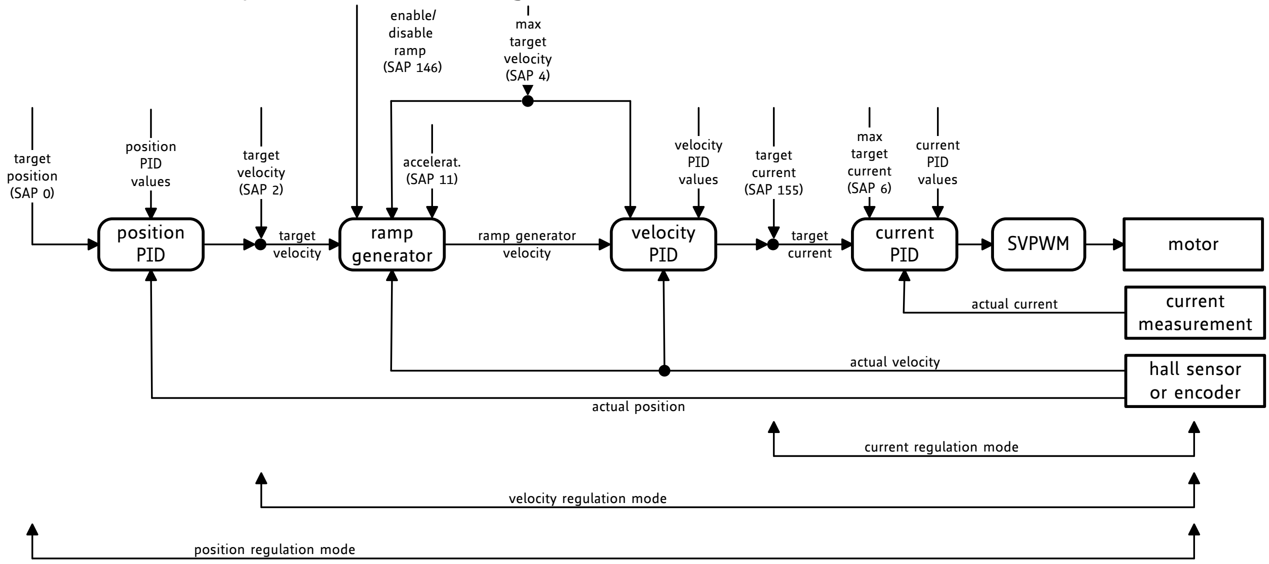


Figure 12.1: Cascaded PID regulation

12.2 Current PID regulation

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

The PID regulation uses five basic parameters: The *P*, *I*, *D* and *I-Clipping value* as well as the *timing control value*. The timing control value (*current regulation loop multiplier*, axis parameter 134) determines how often the current regulation is invoked. It is given in multiple of 25µs:

= resulting delay between two PID calculations
= *current regulation loop multiplier* parameter

For most applications it is recommended to leave this parameter unchanged at its default of 2*25µs. Higher values may be necessary for very slow and less dynamic drives. The structure of the current PID regulator is shown in figure 12.2. It has to be parameterized with respect to a given motor.

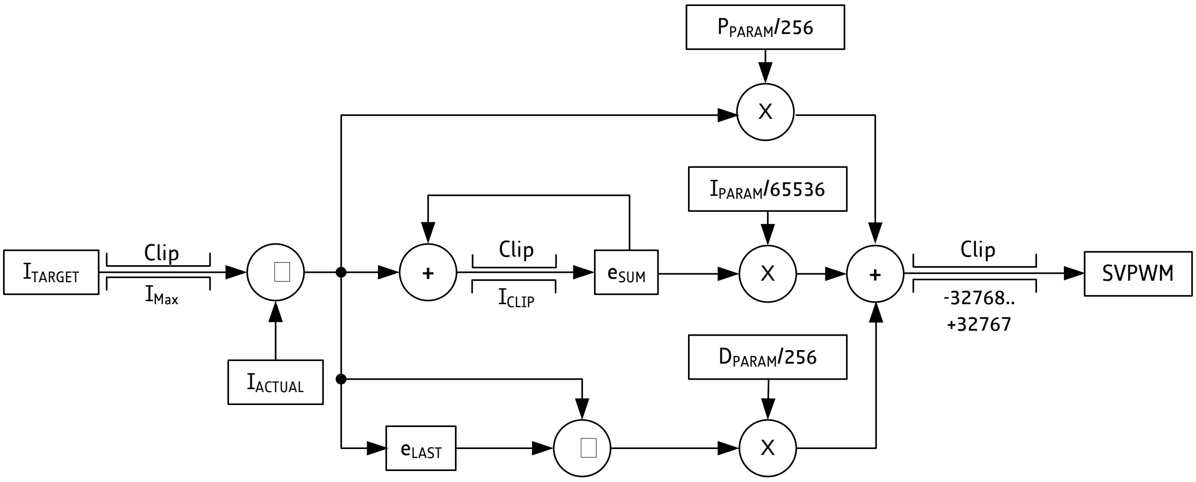


Figure 12.2: Current PID regulation

Parameter	Description
I_{ACTUAL}	Actual motor current (GAP 150)
I_{TARGET}	Target motor current (SAP 155)
I_{Max}	Max. target motor current (SAP 6)
e_{LAST}	Error value of the last PID calculation (GAP 200)
e_{SUM}	Error sum for integral calculation (GAP 201)
P_{PARAM}	Current P parameter (SAP 172)
I_{PARAM}	Current I parameter (SAP 173)
D_{PARAM}	Current D parameter (SAP 174)
I_{CLIP}	Current I-Clipping parameter (SAP 175) (a value of 1000 allows the I-part to use 100% of the SVPWM)

Table 12.1: Current PID parameter description

To parameterize the current PID regulator for a given motor, first set the P, I and D parameter to zero. Then start the motor by using a low target current (e.g. 1000mA). Then modify the *current P parameter*. Start from a low value and go to a higher value, until the actual current nearly reaches the desired target current.

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

<u>*Attention:*</u> 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.

12.3 Velocity PID regulation

Based on the current regulation the motor velocity can be controlled by the velocity PID regulator. Also, the velocity PID regulator uses a timing control value (*PID regulation loop multiplier*, axis parameter 133) which determines how often the PID regulator is invoked. It is given in multiple of 1ms:

= resulting delay between two PID calculations
= *PID regulation loop multiplier* 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. The structure of the velocity PID regulator is shown in figure 12.3.

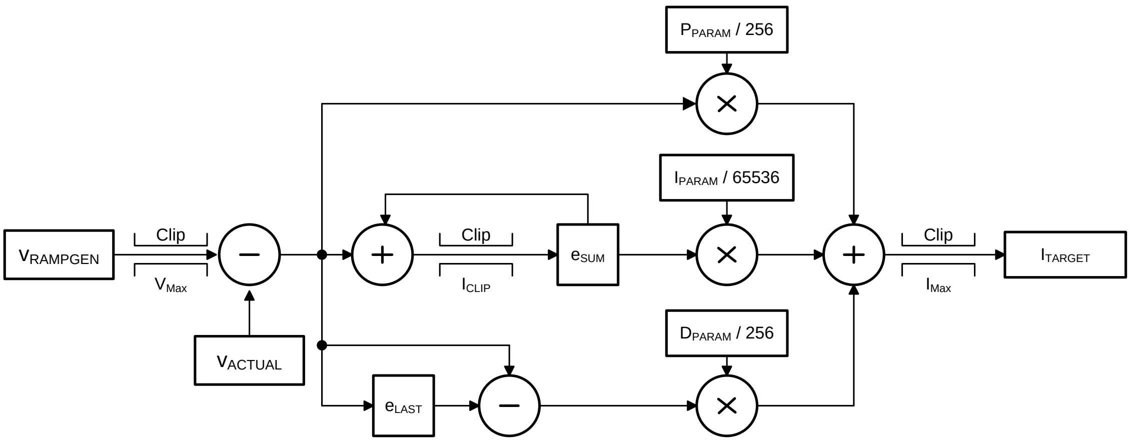


Figure 12.3: Velocity PID regulation

Parameter	Description
v_{ACTUAL}	Actual motor velocity (GAP 3)
$v_{RAMPGEN}$	Target velocity of ramp generator (SAP 2, GAP 13)
v_{Max}	Max. target velocity (SAP 4)
e_{LAST}	Error value of the last PID calculation (GAP 228)
e_{SUM}	Error sum for integral calculation (GAP 229)
P_{PARAM}	Velocity P parameter (SAP 140, SAP 234)
I_{PARAM}	Velocity I parameter (SAP 141, SAP 235)
D_{PARAM}	Velocity D parameter (SAP 142, SAP 236)
I_{CLIP}	Velocity I-Clipping parameter (SAP 143, SAP 237) [1/10%] of I_{Max} (a value of 1000 allows the I-part to reach I_{Max})
I_{Max}	Max. target current (SAP 6)
I_{Target}	Target current for current PID regulator (GAP 155)

Table 12.2: Parameter description for velocity PID regulation

To parameterize the PID regulator, set the *velocity I parameter* and *velocity D parameter* to zero and start the motor by using a medium target velocity (e.g. 2000 rpm). Then modify the *velocity P parameter* only. Start from a low value and go to a higher value, until the actual motor velocity reaches 80 or 90 % of the desired motor velocity. The rest of the velocity difference can be reduced by using a high I-Clipping value (e.g. 1000) and a slow increase of the *velocity I parameter* with the *velocity D parameter* still set to zero. For the first tests, both PID parameter sets can be set equal.

12.4 Velocity ramp generator

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

12.5 Position PID regulation

Based on the current and velocity PID regulators the TMCM-1632/TMCM-1610-KR support a positioning mode based on encoder or hall sensor position. During positioning the velocity ramp generator can be activated to enable motor positioning with controlled acceleration or disabled to support motor positioning with max. target velocity. The structure of the position PID regulator is shown in figure 12.4.

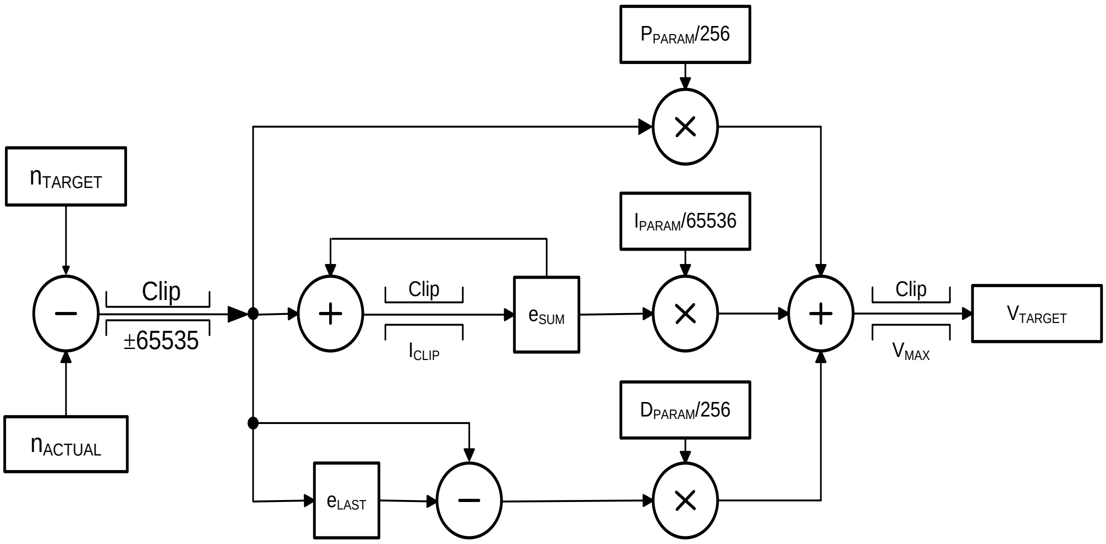


Figure 12.4: Positioning PID regulation

Parameter	Description
n_{ACTUAL}	Actual motor position (GAP 1)
n_{TARGET}	Target motor position (SAP 0)
e_{LAST}	Error value of the last PID calculation (GAP 226)
e_{SUM}	Error sum for integral calculation (GAP 227)
P_{PARAM}	Position P parameter (SAP 130, SAP 230)
I_{PARAM}	Position I parameter (SAP 131, SAP 231)
D_{PARAM}	Position D parameter (SAP 132, SAP 232)
I_{CLIP}	Position I-Clipping parameter (SAP 135, SAP 233) [1/10%] of V_{MAX} (a value of 1000 allows the I-part to reach V_{MAX})
V_{MAX}	Max. allowed velocity (SAP 4)
V_{TARGET}	New target velocity for ramp generator (GAP 13)

Table 12.3: Position PID parameter description

The PID regulation uses five basic parameters. The P, I, D, and I-Clipping value as well as a timing control value. The timing control value (*PID regulation loop parameter* - axis parameter 133) determines how often the PID regulator is invoked. It is given in multiple of 1ms:

= the resulting delay between two PID calculations

= *PID regulation loop multiplier* 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.

Based on the velocity PID regulator the position PID regulator can be parameterized as P regulator in the simplest case. Therefore, disable the velocity ramp generator and set *position P, I, and D parameter* to zero. Now set a target position and increase the *position P parameter* until the motor reaches the target position approximately.

After finding a good *position P parameter* the velocity ramp generator can be switched on again. Based on the *Max. positioning velocity* (axis parameter 4) as well as the *acceleration* (axis parameter 11) 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 desired target position. 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*) can be chosen by axis parameter 10. Since the motor typically is assumed not to signal target reached when the target was just passed in a short moment at a high velocity, additionally a maximum target reached velocity (*MVP target reached velocity*) can be defined by axis parameter 7. A value of zero is the most universal, since it implies that the motor stands still at the target. But when a fast rising of the *Position end flag* is desired, a higher value for *MVP target reached velocity* parameter will save a lot of time. The best value should be tried out in the actual application. The correlation of axis parameter 10, 7, the target position and the position end flag are summarized in figure 12.5.

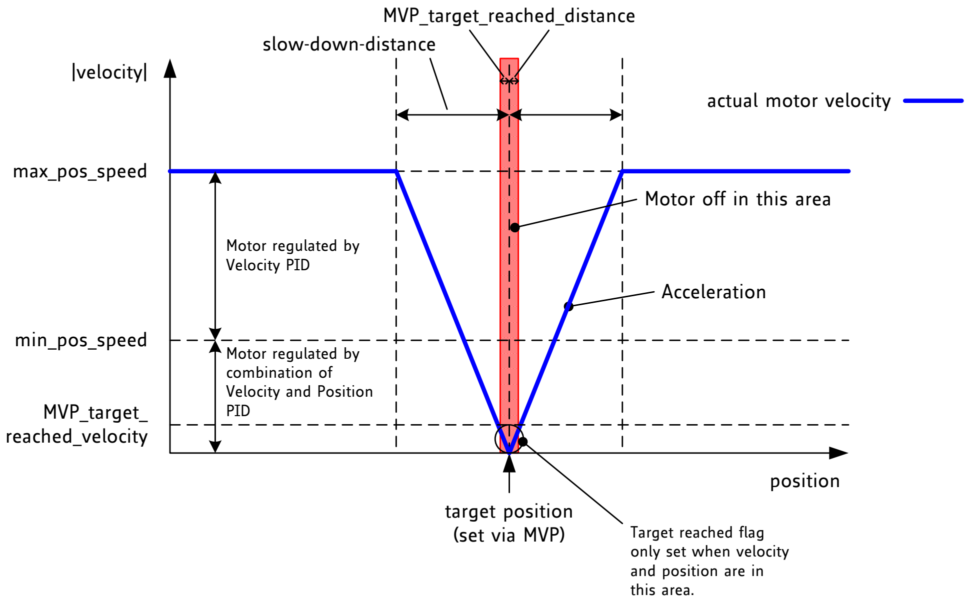


Figure 12.5: Positioning algorithm

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 cannot keep the position.

12.6 Parameter sets for PID regulation

The velocity and position PID regulation provides two parameter sets, which are used as follows:

- Below a specified velocity threshold the PID regulator uses a combination of parameter set 1 and parameter set 2
- Above the velocity threshold the PID regulator uses only parameter set 2. If the velocity threshold is set to zero, parameter set 2 is used all the time. (The switch over between both parameter sets is soft.)

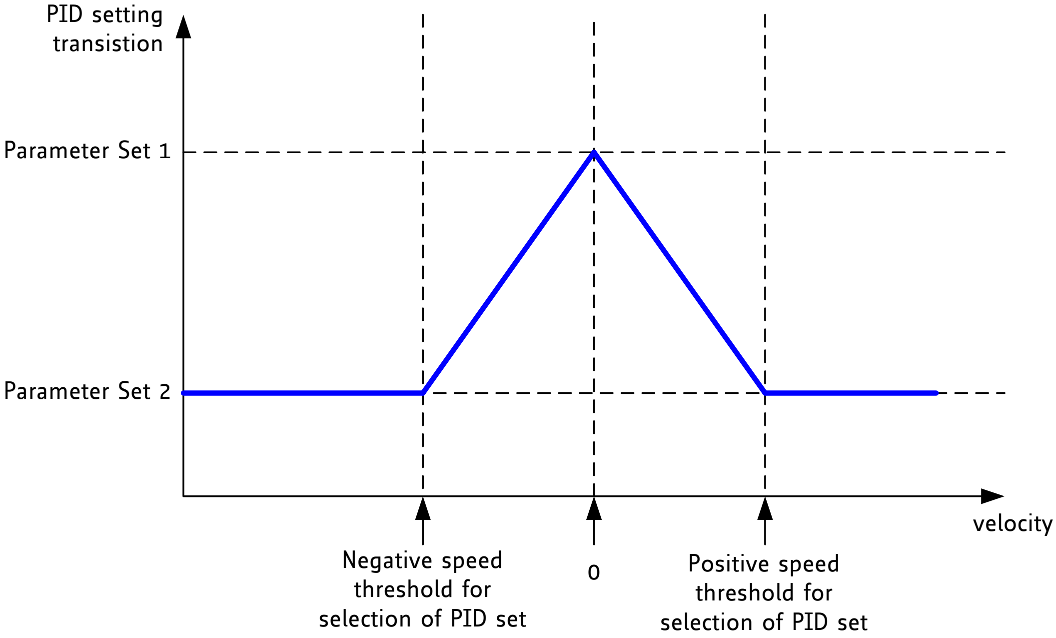


Figure 12.5: Transition between PID parameter sets

The velocity thresholds for velocity, and position PID regulation can be set as follows:

- axis parameter 8: velocity threshold for velocity PID regulator
- axis parameter 12: velocity threshold for position PID regulator

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.

13 Temperature calculation

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

Example 1:

ADC = 1000

$R_{NTC} \approx 6{,}81$

$T \approx 35$

Example 2:

ADC = 1200

$R_{NTC} \approx 5{,}31$

$T \approx 42$

14 I²t monitoring

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

is the desired average current

is the thermal winding time constant given by the motor datasheet

Example:

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

a)

b)

c)

d)

15 Revision history

15.1 Firmware revision

Version	Date	Author	Description
1.32	2010-OCT-18	OK	First version supporting all features
1.37	2011-MAR-03	ED	Velocity and positioning regulation updated
1.38	2011-MAR-15	ED	Added PWM mode Updated current, velocity, and positioning mode
1.39	2011-MAR-18	ED	Velocity regulation updated
1.40	2011-MAR-24	ED	Updated velocity ramp for positioning mode, updated current limitation, updated standard PID parameters for positioning mode
1.41	2011-APR-05	ED	Added PWM, Current, and Initialize mode in process data output buffer Added explicit usage of the Initialize mode before any commanded motor movement Updates/Fixes in current measurement, current regulation, transmission-corrections for *MaxPositioningSpeed* and *acceleration* Added SAP/GAP 15 for initialization in mailbox mode More debugging and testing
1.42	2011-APR-6	MJ, ED	Protection against flashing a module with a firmware which is not intended for the module. Added support for password protected parameter.
1.43	2011-APR-20	ED	Password for secured parameters changed. Added usage of axis parameter 4 as maximal velocity for velocity ramp in velocity mode. Init_sine_speed default value set to a negative value for TMCM-KR841controller. Changed default values for Max_positioning_speed and acceleration for assumed gear ratio of 1. Added global parameter 90 for default EtherCAT timeout. Axis parameter 156 changed from 16 to 32 Bit. Added axis parameter 158 to clear the EtherCAT timeout flag. Added axis parameter 25, 26, 27, 28 for I²t monitoring. Added axis parameter 30 for monitoring operational time in minutes. Added error flag IIT_EXCEEDED in axis parameter 156. Added axis parameter 29 to reset the I²t exceeded flag.
1.44	2011-Jun-15	ED	Added actual pwm to process data output buffer. Updated sine initialization based on block hall.
1.45	2011-Aug-23	ED	Regulators cascaded. Removed parameter 147. Removed parameter 211. (Gear ratio is ever 1) Added GAP 157 (module supply current) (TMCM-

			1632). Added axis parameter 9 (motor halted velocity). Use of global parameters for firmware update save values. (max target current and motor poles have to be set for initial commissioning as described in 9.1.7) Removed password protection for axis parameter 7 and 10. Updated communication with gripper module (TMCM-841). Updated I²t-Exceeded-behavior. Removed axis parameter 137, 252.
1.46	2011-Nov-22	ED	Changed SGP 90 to password protected. Added actual motor direction info to GAP 153. Added sign for GAP 150. Current measurement and I²t monitoring updated. PID-parameterization and regulation updated. Removed PWM change limitation and axis parameter 246.
1.47	2011-Dec-12	ED	Timing of velocity measurement updated. Current measurement in Block/Hall mode updated.
1.48	2011-Dec-19	ED	Wrong set hall error flag at sine initialization corrected. Communication with EtherCAT-chip updated. Standard current regulation delay multiplier and I-Parameter updated.
2.00	2013-Sept-23	ED	Replaced current regulator with a FOC-current/flux regulator (new PID parameter needed). Removed PWM regulation mode. Replaced encoder initialization. Commutation changed to FOC-encoder based mode (AP 159: 7). Removed AP 5, 15, 136, 153, 154, 160, and 167. Removed first current PID parameter set (AP 168, 169, 170, 171, and 176). Removed AP 242, 243, and 247. Added target motor position, target motor current, target velocity and ramp generator velocity to PDO. Added clipping of many input parameters. Updated PDO-input-output-buffer processing for faster handling of POSITION_END_FLAG. Increased PWM frequency from 20kHz to 40kHz. Increased current regulation frequency from 10kHz to 20kHz.

Table 15.1: Firmware revision

15.2 Document revision

Version	Date	Author	Description
1.0	2010-DEC-09	SD	Initial version
1.1	2011-MAR-03	ED	SAP (11.2.3) and GAP (11.2.4) parameter list updated Parameter list for module configuration updated (9.1.6) Chapter 12: PID regulation updated
1.2	2011-MAR-04	SD	New front page, minor changes
1.3	2011-MAR-04	ED	SAP, GAP, STAP and RSAP axis parameter summarized in section 11.9
1.4	2011-MAR-15	ED	Added axis parameter 154/155 (PWM /current mode) Updated axis parameter 156 (Error/Status flags)
1.5	2011-MAR-23	ED	Updated position PID regulation picture
1.6	2011-MAR-28	SD	Figures in chapter 12 updated, captions corrected, minor changes
	2011-APR-01	ED	PWM, Target, and Initialize mode added in process data output buffer (chapter 7) Updated behavior after aborted EtherCat communication
1.7	2011-APR-05	ED	Added SAP/GAP 15 for initialization in mailbox mode Added Bit 15 in GAP 156: for "module initialized" status bit Minor changes
1.8	2011-APR-08	ED	Added description for SGP, GGP, STGP, RSGP, and an application scenario to store a user variable with these commands. Added SAP/GAP 248 to approve SAP for password protected parameters. Added description for usage of password protected parameters (11.9.1).
1.9	2011-APR-20	ED	Added GAP 13: get ramp generator speed Updated description for axis parameter 4. Added global parameter 90 for EtherCAT timeout. Added flag for EtherCAT timeout in axis parameter 156. Added axis parameter 158 to clear the EtherCAT timeout flag. Added axis parameter 25, 26, 27, 28 for I^2t monitoring. Added axis parameter 30 for monitoring operational time in minutes. Added error flag IIT_EXCEEDED in axis parameter 156. Added axis parameter 29 to reset the I^2t exceeded flag. Added short description of I^2t monitoring and examples to find a meaningful I^2t limit value.
2.0	2011-Sep-01	ED	Updated *data input buffer* entries. Updated paragraph *12.3 Current PID regulation*. Added *TMCL control functions*. Updated error flags in data input buffer. Removed TMCL parameter 147. Removed TMCL parameter 211. (Gear ratio is ever 1) Added chapter 13 temperature measurement. Added GAP 157 (module supply current). Added axis parameter 9 (motor halted velocity). Updated 9.1.7 parameterization for initial commissioning. Removed password protection for axis parameter 7 and 10. Updated I^2t-Exceeded-behavior. Added chapter 11.9.1 Axis parameter sorted by

			functionality. Updated chapter 12. current, velocity, and position regulation.
2.1	2011-Sep-07	ED	Updated description for parameter 156 (error/status flags). Removed axis parameter 137, 252.
2.2	2011-Nov-22	ED	Added axis parameter 200 and 201. Changed SGP 90 to password protected. Added actual motor direction info to GAP 153. Added sign for GAP 150. Added string/binary format description for command 136. I^2t section updated. Removed axis parameter 246 (PWM change limitation). Some minor changes.
2.3	2011-Dec-12	ED	Updated description for GAP 157. Updated naming and description for axis parameter 133 and 134.
2.4	2011-Dec-19	ED	Updated standard current regulation delay multiplier in 12.3.
2.5	2012-Feb-23	ED	Updated process data output/input buffer in chapter 7. Updated axis parameter overview. Updated PID regulation chapter. Some minor changes. (e.g. TMCM-KR-841 -> TMCM-1610-KR)
2.6	2012-Mar-23	ED	Updated process data input buffer in chapter 7.
2.7	2013-July-23	ED	Updated both axis parameter lists due parameter clipping.
2.8	2013-Sept-09	ED	Updated current regulation delay from 50µs to 25µs.

Table 15.2: Document revision

16 References

[TMCM-1632] TMCM-1632 Hardware Manual (please refer to our homepage http://www.trinamic.com)
[TMCM-1610-KR] TMCM-1610-KR Hardware Manual
[TMC603] TMC603 Data Sheet (please refer to our homepage http://www.trinamic.com)