

# TMCC160 TMCL™ FIRMWARE MANUAL

TMCC160 TMCL Firmware Version 2.09 • 2015-NOV-25 | Document Revision 1.0 • 2015-DEC-04

**The TMCL™ Firmware is used in combination with the integrated TMCC160 motionCookie™ microsystem with 3-Phase BLDC/PMSM gate driver for up to 24V and 1A gate current.**



## Features

- The TMCL Firmware can operate in stand-alone or in remote control mode.
- Nonvolatile program memory for scripts with up to 2048 TMCL commands for stand-alone operation.
- PC-based application development software TMCL-IDE Firmware update via RS232, RS485, and CAN using the TMCL-IDE Firmware update.
- The TMCL Firmware supports block commutation, which is based on hall sensors for BLDC motors. It also supports sinusoidal commutation based on encoder or hall sensor feedback for PMSM motors.
- A controlled mode for sine commutation is also available and can be used to set the correct hall sensor and encoder settings and to find incorrectly connected hall sensor signals.

# TABLE OF CONTENTS

<b>TMCC160 TMCL™ Firmware Manual .....</b>	<b>2</b>
<b>Features .....</b>	<b>2</b>
<b>TABLE OF CONTENTS.....</b>	<b>2</b>
<b>1 TMCL Overview.....</b>	<b>4</b>
<b>1.1 Basic TMCL Formats and Commands.....</b>	<b>4</b>
1.1.1 Request Format .....	4
1.1.2 Reply Format .....	6
1.1.3 Motion Commands.....	7
1.1.4 Parameter Commands.....	7
1.1.5 I/O Port Commands.....	7
<b>1.2 Detailed TMCL Commands.....</b>	<b>8</b>
1.2.1 ROR (Rotate Right) .....	8
1.2.2 ROL (Rotate Left) .....	9
1.2.3 MST (Motor Stop).....	10
1.2.4 MVP (Move to Position).....	11
1.2.5 SAP (Set Axis Parameter) .....	12
1.2.6 GAP (Get Axis Parameter).....	13
1.2.7 STAP (Store Axis Parameter) .....	14
1.2.8 RSAP (Restore Axis Parameter).....	15
1.2.9 SGP (Set Global Parameter).....	16
1.2.10 GGP (Get Global Parameter) .....	17
1.2.11 STGP (Store Global Parameter).....	18
1.2.12 RSGP (Restore Global Parameter) .....	19
1.2.13 SIO (Set Output) and GIO (Get Input / Output).....	20
1.2.14 TMCL Control Functions.....	23
<b>2 Axis Parameter Overview (SAP, GAP, STAP, RSAP) .....</b>	<b>25</b>
<b>2.1 Axis Parameters Sorted by Functionality.....</b>	<b>31</b>
<b>3 Global Parameter Overview (SGP, GGP, STGP, RSGP).....</b>	<b>38</b>
<b>3.1 Bank 0 .....</b>	<b>38</b>
<b>3.2 Bank 2 .....</b>	<b>41</b>
<b>4 Motor Regulation .....</b>	<b>42</b>
<b>4.1 Structure of Cascaded Motor Regulation Modes.....</b>	<b>42</b>
<b>4.2 Current Regulation .....</b>	<b>42</b>
<b>4.3 Velocity Regulation.....</b>	<b>45</b>



- 4.4 **Velocity Ramp Generator**.....47
- 4.5 **Position Regulation** .....47
- 5 I<sup>2</sup>T-Monitoring**..... 50
- 6 Supplemental Directives** ..... 52
- ESD DEVICE INSTRUCTIONS .....52
- Producer Information.....52
- Copyright.....52
- Target User .....52
- Disclaimer: Life Support Systems.....53
- Disclaimer: Intended Use .....53
- 7 Safety Alerts**..... 54
- 8 References** ..... 54
- 8.1 **Revision History** .....54
- 8.2 **Related Documents and Tools**.....54
- 8.3 **Table Index** .....55
- 8.4 **Figure Index**.....57
- 8.5 **Keyword Index** .....58



# 1 TMCL Overview

The software that runs on the microprocessor of the TMCC160 motionCookie™ consists of two parts:

**1. Boot loader:**

The boot loader is installed by TRINAMIC during production. It remains untouched throughout its entire product lifetime.

**2. Firmware:**

The firmware can be updated by the user. The firmware is related to the standard TMCL firmware [TMCL] with regard to protocol and commands. The module is based on an ARM Cortex-M4 microcontroller and the high performance pre-driver TMC6130. It supports standard TMCL with a special range of parameters and values. TMCL sample scripts are available on the product's web page.

## 1.1 Basic TMCL Formats and Commands

### 1.1.1 Request Format

**Description of Request Fields**

**When commands are sent from a host to a module, the request format must be used. Every request command consists of:**

- A one-byte command field
- A one-byte type field
- A one-byte motor/bank field
- A four-byte value field.

Consequently, the binary representation of a command always has seven bytes.

**However, when a command is sent via RS232, or RS485UART interface, it has to be enclosed by an address byte at the beginning and a checksum byte at the end. In this case it consists of nine bytes.**

RS232 and RS485 TMCL Request Format	
Bytes	Meaning
1	Module address
1	Command number
1	Type number
1	Motor or Bank number
4	Value (MSB first!)
1	Checksum

*Table 1: RS232 and RS485 TMCL Request Format*

•→ Please turn to next page for checksum calculation.



## Checksum Calculation

The checksum is calculated by adding up all bytes (including the module address byte) using 8-bit addition. Here is a C-example for the calculation:

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

Checksum = Command[0];
for(i=1; i<8; i++) { Checksum+=Command[i]; }

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

### *Calculation Examples 1: C-Example for Checksum Calculation*

- i** In case of CAN communication, the checksum is omitted. Consequently, the request format then only consists of 8 bytes.



## 1.1.2 Reply Format

### RS232 and RS485 Reply Format

Whenever a command is sent to a module, the module sends a reply. The reply format for RS232 and RS485 is structured as follows:

RS253 and RS485 TMCL Reply Format	
Bytes	Description
1	Reply address
1	Module address
1	Status (e.g. 100 means no error)
1	Command number
4	Value (MSB first!)
1	Checksum

Table 2: RS253 and RS485 TMCL Reply Format

The checksum is calculated similar to the checksum of the request format. The status code can have one of the following values:

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

Table 3: TMCL Reply Status Codes

#### **NOTE:**

→ In case of CAN communication, the checksum is omitted. Consequently, the request format then only consists of 8 bytes.



### 1.1.3 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 standalone mode.

TMCL Motion Commands		
Mnemonic	Command Number	Description
ROR	1	Rotate right.
ROL	2	Rotate left.
MST	3	Motor stop.
MVP	4	Move to position.

Table 4: TMCL Motion Commands

### 1.1.4 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 standalone mode.

TMCL Parameter Commands		
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.

Table 5: TMCL Parameter Commands

### 1.1.5 I/O Port Commands

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

TMCL I/O Port Commands		
Mnemonic	Command Number	Meaning
SIO	14	Set output
GIO	15	Get input

Table 6: TMCL I/O Port Commands



## 1.2 Detailed TMCL Commands

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

### 1.2.1 ROR (Rotate Right)

**Process**

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

**Description**

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

**Related commands:** ROR, MST, SAP, GAP

**Mnemonic:** ROR 0, <velocity>

**NOTE:**

→ An example for ROR is provided in [Table 8](#).

ROR Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE <velocity>
1	don't care	0	-200000... +200000

*Table: ROR (Rotate Right) Request in Direct Mode*

ROR Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 – OK	1	don't care

*Table 7: ROR (Rotate Right) Reply in Direct Mode*

ROR Example: Rotate right with velocity = 350: Mnemonic: ROR 0, 350								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$01	\$01	\$00	\$00	\$00	\$00	\$01	\$5e

*Table 8: ROR Example: Rotate right with Velocity=350*





### 1.2.2 ROL (Rotate Left)

**Process**

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

**Description**

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

**Related commands:** ROR, MST, SAP, GAP

**Mnemonic:** ROL 0, <velocity>

**NOTE:**

→ An example for ROL is provided in [Table 11](#).

ROL Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE <velocity>
2	don't care	0	-200000... +200000

Table 9: ROL (Rotate Left) Request in Direct Mode

ROL Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 - OK	2	don't care

Table 10: ROL (Rotate Left) Reply in Direct Mode

ROL Example:								
Request: Rotate left with Velocity = 1200: Mnemonic: ROL 0, 1200								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$01	\$02	\$00	\$00	\$00	\$00	\$04	\$b0

Table 11: ROL Example: Rotate left with Velocity=1200



### 1.2.3 MST (Motor Stop)

#### Process

The motor is instructed to stop with the MST function. The axis parameter target velocity is set to zero.

#### Description

**Related commands:** ROL, ROR, SAP, GAP

**Mnemonic:** MST 0

#### NOTE:

→ An example for MST is provided [Table 14](#).

MST Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE
3	don't care	0	don't care

Table 12: MST (Motor Stop) Request in Direct Mode

MST Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 – OK	3	don't care

Table 13: MST (Motor Stop) Reply in Direct Mode

MST Example: Stop Motor at Mnemonic: MST 0								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$01	\$03	\$00	\$00	\$00	\$00	\$00	\$00

Table 14: MST Example: Stop Motor at Mnemonic: MST 0



### 1.2.4 MVP (Move to Position)

#### Two available Operation Types

The motor is instructed to move to a specified relative or absolute position.

The motor uses the predefined acceleration/deceleration ramp and the positioning speed. This setting can be changed by the user. The command is non-blocking (like all commands). A reply is sent immediately after command interpretation. Further commands can follow even if the motor has not yet reached its target position. The maximum velocity and acceleration are defined by axis parameters #4 and #11.

**Two operation types are available:**

- **ABS:** Moves the motor to an absolute position in the range from - 2147483648... +2147483647.
- **REL:** Moves the motor relative to the actual position.

**NOTE:**

→ [Table 17](#) (page 12) provides an ABS operation example, [Table 18](#) (page 12) and REL example.

#### Process Description

A new position value is transferred to the axis parameter #0 target position.

**Related commands:** SAP, GAP, and MST

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

MVP (ABS / REL) Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE
4	0 ABS – absolute	0	<position> -2147483648... +2147483647
	1 REL – relative	0	<offset> -2147483648... +2147483647

Table 15: MVP ABS/ REL Request in Direct Mode

MVP Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 – OK	4	don't care

Table 16: MVP ABS / REL Reply in Direct Mode

- Please turn page for ABS and REL examples.



MVP ABS Example:								
Move Motor to Absolute Position 9000: Mnemonic: MVP ABS, 0, 9000								
Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	\$01	\$04	\$00	\$00	\$00	\$00	\$23	\$28

Table 17: MVP ABS Example: Move Motor to Absolute Position 9000

MVP REL Example:								
Move Motor 1000 steps to Relative Position (move relative -1000): Mnemonic: MVP REL, 0, -1000								
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

Table 18: (MVP REL Example: Move Motor 1000 Steps to Relative Position

### 1.2.5 SAP (Set Axis Parameter)

**SRAM Settings**  
**Process**  
**Description**

Most of the motion control parameters of the module can be specified by using the SAP command. The settings are stored in SRAM and therefore are volatile. Thus, information is lost after power-off.

- i You must use command STAP (store axis parameter) in order to store your specified setting permanently.

**Related commands:** GAP, STAP, and RSAP

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

**NOTE:**

→ An Example for setting the axis parameter is provided in [Table 21](#) (page 13).

SAP Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE
5	<parameter number>	0	<value>

Table 19: SAP (Set Axis Parameter) Request in Direct Mode

SAP Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 - OK	5	don't care

Table 20: SAP (Set Axis Parameter) Reply in Direct Mode

- i A list of all parameters, which can be used for the SAP command, is shown in section 3.



SAP Example: Absolute Maximum Current 2000mA: Mnemonic: SAP6, 0, 2000								
Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	\$01	\$05	\$06	\$00	\$00	\$00	\$07	\$D0

Table 21: SAP Example: Absolute Max. Current 2000MA:

### 1.2.6 GAP (Get Axis Parameter)

#### Separate Adjustment of TMCC160-EVAL Parameters

Most parameters of the TMCC160-EVAL can be adjusted separately. They can be read out using the GAP command.

**Related commands:** SAP, STAP, and RSAP

**Mnemonic:** GAP <parameter number>, 0

**NOTE:**

→ A GAP request example is provided in [Table 24](#), and for a GAP reply example in [Table 25](#) (page 14).

GAP Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE
6	<parameter number>	0	don't care

Table 22: GAP (Get Axis Parameter) Request in Direct Mode

GAP Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 - OK	6	don't care

Table 23: GAP (Get Axis Parameter) Reply in Direct Mode

- i** A list of all parameters which can be used for the GAP command is shown in section 3.

GAP Request Example: Get the actual position of motor 0: Mnemonic: GAP 1, 0								
Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	\$01	\$06	\$01	\$00	\$00	\$00	\$00	\$00

Table 24: GAP Request Example: Get actual Position of Motor 0



GAP Reply Example								
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	\$01	\$64	\$06	\$00	\$00	\$02	\$c7

Table 25: GAP Reply Example

### 1.2.7 STAP (Store Axis Parameter)

#### STAP Settings stored in SRAM

The STAP command stores an axis parameter on a permanent basis; which was previously set with a Set Axis Parameter command (SAP).

An axis parameter stored in SRAM will be transferred to EEPROM and loaded from EEPROM after next power-up.

**Related commands:** SAP, RSAP, and GAP

**Mnemonic:** STAP <parameter number>, 0

**NOTE:**

→ An example for STAP is provided below.

STAP Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE
7	<parameter number>	0	don't care <sup>1</sup>

Table 26: STAP (Store Axis Parameter) Request in Direct Mode

<sup>1</sup> The value operand of this function has no effect. Instead, the currently used value (e.g. selected by SAP) is saved.

STAP Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 – OK	7	don't care

Table 27: STAP (Store Axis Parameter) Reply in Direct Mode

- i A list of all parameters which can be used for the STAP command is shown in section 3.

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

Table 28: STAP Example: Store Maximum Speed: STAP 4,0

- i The STAP command has no effect when the configuration EEPROM is locked. In this case, the error code 5 (configuration EEPROM locked) is returned.



### 1.2.8 RSAP (Restore Axis Parameter)

**Non-Volatile Memory Locations**

For all configuration-related axis parameters non-volatile memory locations are provided. By default, most parameters are automatically restored after power-up.

A single parameter - that was changed beforehand - can now be reset as well, as explained below:

**Process Description**

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

**Related commands:** SAP, STAP, and GAP

**Mnemonic:** RSAP <parameter number>, 0

**NOTE:**

→ An example for RSAP is provided below.

RSAP Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE
8	<parameter number>	0	don't care

Table 29: RSAP Request in Direct Mode

RSAP Reply in Direct Mode		
STATUS	COMMAND	VALUE
100 – OK	8	don't care

Table 30: RSAP Reply in Direct Mode

- i A list of all parameters - which can be used for the RSAP command - is shown in section 3.

RSAP Example:								
Restore Maximum Motor Current 0: Mnemonic: RSAP 6, 0								
Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	\$01	\$08	\$06	\$00	\$00	\$00	\$00	\$00

Table 31: RSAP Example: Restore Maximum Motor Current 0



### 1.2.9 SGP (Set Global Parameter)

#### Organization of Parameters in Banks

Global parameters are related to the host interface, peripherals or other application specific variables. The different groups of these parameters are organized in banks to allow a larger total number for future products.

Currently, bank 0 is used for global parameters and bank 2 is intended for user variables.

**Related commands:** GGP, STGP, RSGP

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

**NOTE:**

→ An example for SGP is provided below.

SGP Request in Direct Mode			
COMMAND	TYPE	MOT/BANK	VALUE
9	<parameter number>	<bank number>	<value>

Table 32: SGP (Set Global Parameter) Request in Direct Mode

SGP Reply in Direct Mode	
STATUS	VALUE
100 – OK	don't care

Table 33: SGP (Set Global Parameter) Reply in Direct Mode

- i A list of all parameters which can be used for the SGP command is shown in section 4.

SGP Example: Set Variable 0 at Bank 2 to 100: Mnemonic: SGP, 0, 2, 100								
Byte Index	0	1	2	3	4	5	6	7
Function	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	\$01	\$09	\$00	\$02	\$00	\$00	\$00	\$64

Table 34: SGP Example: Mnemonic: SGP, 0, 2, 100





### 1.2.10 GGP (Get Global Parameter)

**Read out all Global Parameters**

All global parameters can be read with this function.

**Related commands:** SGP, STGP, RSGP

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

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

Table 35: GGP (Get Global Parameter) Request Direct Mode

Reply in Direct Mode	
STATUS	VALUE
100 – OK	<value>

Table 36: GGP (Get Global Parameter) Reply in Direct Mode

- i** A list of all parameters which can be used for the GGP command is shown in section 4.

Example: Get Variable 0 from Bank 2: Mnemonic: GGP, 0, 2								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$01	\$0a	\$00	\$02	\$00	\$00	\$00	\$00

Table 37: GAP Example: Get Variable 0 from Bank 2



### 1.2.11 STGP (Store Global Parameter)

#### STGP Configuration

Some global parameters are located in RAM memory. Consequently, modifications are lost at power-down.

The 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>

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

Table 38: STGP Request in Direct Mode

Reply in Direct Mode	
STATUS	VALUE
100 – OK	don't care

Table 39: STGP Reply in Direct Mode

- i A list of all parameters which can be used for the STGP command is shown in section 4.

Example: Restore Variable 0 of bank 2 to EEPROM Configuration: Mnemonic: STGP, 0, 2								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Target-address	Instruction Number	Type	Motor/Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$01	\$0b	\$00	\$02	\$00	\$00	\$00	\$00

Table 40: STGP Example: Restore Variable 0 of bank 2 to EEPROM Configuration



### 1.2.12 RSGP (Restore Global Parameter)

**Process**

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

**Description**

**Related commands:** SGP, GGP, STGP

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

**NOTE:**

→ An example for RSGP is provided below.

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

Table 41: RSGP (Restore Global Parameter) Request in Direct Mode

Reply in Direct Mode	
STATUS	VALUE
100 – OK	don't care

Table 42: RSGP (Restore Global Parameter) Reply in Direct Mode

- i** A list of all parameters which can be used for the RSGP command is shown in section 4.

RSGP Example:								
Copy Variable 0 at Bank 2 from EEPROM Configuration to RAM Location: Mnemonic: RSGP, 0, 2								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$01	\$0c	\$00	\$02	\$00	\$00	\$00	\$00

Table 43: RSGP: Copy Variable 0 at Bank 2 from EEPROM Configuration to RAM Location



### 1.2.13 SIO (Set Output) and GIO (Get Input / Output)

#### SIO und GIO Command Settings

The TMCC160-EVAL provides two commands for dealing with inputs and outputs:

- **SIO:**  
Sets the status of the general digital output either to low (0) or to high (1).
- **GIO:**  
Reads out the status of the two available general purpose inputs of the module.

#### **NOTE:**

→ The command reads out a digital or analogue input port.

→ Digital lines read 0 and 1. ADC channel that delivers 12 bit (value of 0... 4095).

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

Table 44: Correlation between I/Os and Banks



### 1.2.13.1 SIO (Set Output)

#### Setup of General Output Status

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

The passed value is transferred to the specified output line.

**Related commands:** GIO, WAIT

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

**NOTE:**

→ An SIO reply example is provided below.

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

Table 45: SIO Request in Direct Mode

Reply in Direct Mode	
STATUS	VALUE
100 - OK	don't care

Table 46: SIO Reply in Direct Mode

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

Table 47: SIO Reply Example



### 1.2.13.2 GIO (Get Input / Output)

**Process**

GIO can be used in direct mode or in standalone mode. The specified line is read.

**Description:**  
**Direct Mode or Standalone Mode**

**Related commands:** SIO

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

**OPTION 1: IN STANDALONE MODE**

The requested value is copied to the accumulator (accu) for further processing purposes; such as conditioned jumps.

**OPTION 2: IN DIRECT MODE**

The value is output in the value field of the reply without affecting the accumulator. The actual status of a digital output line can also be read.

**NOTE:**

- For each operation mode an example is provided further down.
- We also provide a table with all available SIO and GIO commands on page 23.

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

Table 48: GIO (Get Input / Output) Request in Direct Mode

Reply in Direct Mode	
STATUS	VALUE
100 – OK	<status of the port>

Table 49: GIO (Get Input / Output) Reply in Direct Mode

Example: GIO Request								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$01	\$0f	\$00	\$01	\$00	\$00	\$00	\$00

Table 50: GIO Request Example

Example: GIO Reply								
Byte Index	0	1	2	3	4	5	6	7
<b>Function</b>	Host-address	Target-address	Status	Instruction	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
<b>Value (hex)</b>	\$02	\$01	\$64	\$0f	\$00	\$00	\$01	\$2e

Table 51: GIO Reply Example



Available SIO and GIO Commands					
I/O	Digital	Analog	GIO <port>, <bank>	SIO <port>, <bank>, <value>	Value Range
Digital input 0 (REF_R)	X	-	GIO 0, 0	-	0/1
Digital input 1 (REF_L)	X	-	GIO 1, 0	-	0/1
Analog input 0	-	X	GIO 0,1	-	0... 4095
ADC single shunt	-	X	GIO 1,1	-	0... 4095
ADC phase A	-	X	GIO 2,1	-	0... 4095
ADC phase B	-	X	GIO 3,1	-	0... 4095
ADC VSupply	-	X	GIO 4,1	-	0... 4095
ADC Temp	-	X	GIO 5,1	-	0... 4095

Table 52: Available SIO and GIO Commands

### 1.2.14 TMCL Control Functions

#### TMCL Control Command 136

There are several TMCL control functions, the most important for user is command 136.

**NOTE:**

- Other control functions can be used with axis parameter
- An example for the two possible replies are provided in [Table 54](#) and [Table 55](#) further down.

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

Table 53: TMCL Command 136 Request in Direct Mode

Command 136 – Reply in Direct Mode	
Type set to 0. Reply as a String:	
Byte Index	Contents
1	Host Address
2... 9	Version string (8 characters, e.g. C160V2.08)

Table 54: TMCL Command 136 Reply in Direct Mode, Type Set to 0

**i** There is no checksum in this reply format!

•→ Please refer to next page for Reply in Direct Mode, type set to 1.



<b>Command 136 – Reply in Direct Mode</b>	
Type set to 1. Version Number in Binary Format:	
<b>Byte Index in Value Field</b>	<b>Contents</b>
1	Version number, low byte
2	Version number, high byte
3	Type number, low byte
4	Type number, high byte

*Table 55: TMCL Command 136, Reply in Direct Mode, Type Set to 1*

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





## 2 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.

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

Table 56: Access Type Description

Axis Parameter Description (Numbers 1- ...254)				
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. absolute ramp velocity	The maximum velocity used for velocity ramp in velocity mode and positioning mode. Set this value to a realistic velocity which the motor can reach!	0... +200000 [rpm]	RWE
6	Max current	Set/get the max allowed motor current. *This value can be temporarily exceeded marginal due to the operation of the current regulator.	0... +14000 [mA]	RWE
7	MVP Target reached velocity	Maximum velocity at which end position flag can be set. Prevents issuing of end position when the target is passed at high velocity.	0... +200000 [rpm]	RWE
9	Motor halted velocity	If the actual speed is below this value the motor halted flag is set.	0.. +200000 [rpm]	RWE
10	MVP target reached distance	Maximum distance at which the position end flag is set.	0... +100000	RWE
11	Acceleration	Acceleration parameter for ROL, ROR, and the velocity ramp of MVP.	0... +100000 [RPM/s]	RWE



Axis Parameter Description (Numbers 1- ...254)				
Number	Axis Parameter	Description	Range [Unit]	Access
13	Ramp generator speed	The actual speed of the velocity ramp used for positioning and velocity mode.	-2147483648... +2147483647 [rpm]	R
25	Thermal winding time	Thermal winding time constant for the used motor. Used for I <sup>2</sup> t monitoring.	0..300000 [ms]	RWE
26	I <sup>2</sup> T limit	An actual I <sup>2</sup> t sum that exceeds this limit leads to increasing the I <sup>2</sup> t exceed counter.	0..270000000	RWE
27	I <sup>2</sup> T sum	Actual sum of the I <sup>2</sup> t monitor.	0..2147483647	R
28	I <sup>2</sup> T exceed counter	Counts how often an I <sup>2</sup> t sum was higher than the I <sup>2</sup> t limit.	0..4294967295	RWE
29	Clear I <sup>2</sup> T exceeded flag	Clear the flag that indicates that the I <sup>2</sup> t sum has exceeded the I <sup>2</sup> t limit.	(ignored)	W
31	BLDC re-initialization	Restart timer and BLDC regulation.	(ignored)	W
133	PID regulation loop delay	Delay of the position and velocity regulator	0... +20 [50µs]	RWE
134	Current regulation loop delay	Delay of the current regulator.	0... +10 [50µs]	RWE
146	Activate ramp	1: Activate velocity ramp generator for position and velocity mode. (Allows usage of acceleration and positioning velocity for MVP command.)	0/1	RWE
150	Actual motor current	Get actual motor current.	-2147483648... +2147483647 [mA]	R
151	Actual voltage	Actual supply voltage.	0...+480 [100mV]	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)	-14000... +14000 [mA]	RW



Axis Parameter Description (Numbers 1- ...254)				
Number	Axis Parameter	Description	Range [Unit]	Access
156	Error/Status flags	Bit 0: Overcurrent flag. This flag is set if the max. current limit is exceeded. Bit 1: Undervoltage flag. This flag is set if supply voltage is too low for motor operation. Bit 2: Overvoltage flag. This flag is set if the motor becomes switched off due to overvoltage. Bit 3: Overtemperature flag. This flag is set if overtemperature limit is exceeded. Bit 4: Motor halted flag. This flag is set if the velocity does not reach the value set with GAP/SAP 9. Bit 5: Hall error flag. This flag is set upon a hall error. Bit 6: Driver error flag Bit 7: Init error flag Bit 8: Stop mode active flag 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  Flag 0 to 15 are automatically reset.	0...+42949672 95	R
159	Commutation mode	0: Block based on hall sensor 6: FOC based on hall sensor 7: FOC based on encoder 8: FOC controlled	0, 6, 7, 8	RWE
161	Encoder set NULL	1: set position counter to zero at next N channel event.	0/1	RWE
163	Encoder clear set NULL	1: set position counter to zero only once 0: always at an N channel event	0/1	RWE
165	Actual encoder commutation offset	This value represents the internal commutation offset. (0 ... max. encoder steps per rotation - 1)	0... 65535	RWE
172	P parameter for current PID	P parameter of current PID regulator.	0... 65535	RWE
173	I parameter for current PID	I parameter of current PID regulator.	0... 65535	RWE
175	Single-shunt offset	The actual ADC offset of the single-shunt current measurement.	0..4095	RWE



Axis Parameter Description (Numbers 1- ...254)				
Number	Axis Parameter	Description	Range [Unit]	Access
176	single-shunt vref	Manually set/get the single-shunt reference voltage.		RWEX
177	Start current	Motor current for controlled commutation.	0... +14000 [mA]	RWE
190	Debug value 0	Freely usable debugging value.		RW
191	Debug value 1	Freely usable debugging value.		RW
192	Debug value 2	Freely usable debugging value.		RW
193	Debug value 3	Freely usable debugging value.		RW
194	Debug value 4	Freely usable debugging value.		RW
195	Debug value 5	Freely usable debugging value.		RW
196	Debug value 6	Freely usable debugging value.		RW
197	Debug value 7	Freely usable debugging value.		RW
198	Debug value 8	Freely usable debugging value.		RW
199	Debug value 9	Freely usable debugging value.		RW
200	Current PID error	Actual error of current PID regulator	-2147483648... +2147483647	R
201	Current PID error sum	Error sum of current PID regulator	-2147483648... +2147483647	R
202	Flux PID error	Actual error of flux PID regulator	-2147483648... +2147483647	R
203	Flux PID error sum	Error sum of flux PID regulator	-2147483648... +2147483647	R
205	Enable single-shunt measurement	Use single-shunt measurement for block commutation.	0..1	RWE
208	Dual shunt factor	Manually adjust the dual shunt current measurement factor.		RWEX
209	Single shunt factor	Manually adjust the single shunt current measurement factor.		RWEX
210	Actual hall angle	Actual hall angle value	-32767... +32767	R
211	Actual encoder angle	Actual encoder angle value	-32767... +32767	R
212	Actual controlled angle	Actual controlled angle value	-32767... +32767	R
214	Driver diagnostic	Driver diagnostic value	0..1000 [0,1%]	R
215	Driver acknowledge	Acknowledge driver status.	(ignored)	W
216	Enable driver SPI	Disable the driver and initialize the driver SPI access.	(ignored)	W
217	Driver status register 2	Read/Write driver status register 2	-2147483648... +2147483647	RW



Axis Parameter Description (Numbers 1- ...254)				
Number	Axis Parameter	Description	Range [Unit]	Access
218	Driver status register 3	Read/Write driver status register 3	-2147483648... +2147483647	RW
219	Driver status register 4	Read/Write driver status register 4	-2147483648... +2147483647	RW
226	Position PID error	Actual error of position PID regulator	-2147483648... +2147483647	R
228	Velocity PID error	Actual error of velocity PID regulator	-2147483648... +2147483647	R
229	Velocity PID error sum	Sum of errors of velocity PID regulator	-2147483648... +2147483647	R
230	P parameter for position PID	P parameter of position PID regulator.	0... 65535	RWE
234	P parameter for velocity PID	P parameter of velocity PID regulator.	0... 65535	RWE
235	I parameter for velocity PID	I parameter of velocity PID regulator.	0... 65535	RWE
237	Enable brake chopper	Enable brake chopper functionality.	0 / 1	RWE
238	Brake chopper limit voltage	If the brake chopper is enabled and supply voltage exceeds this value, the brake chopper output will be activated.	50..480 [100mV]	RWE
239	Brake chopper hysteresis	An activated brake chopper will be disabled if the actual supply voltage is lower than (limit voltage-hysteresis).	0..50 [100mV]	RWE
241	Sine initialization speed	Velocity during initialization in encoder init mode 2. Refer to axis parameter 249, too.	-200000... +200000 [rpm]	RWE
244	Init sine delay	Duration for sine initialization sequence. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	0... 10000 [ms]	RWE
249	Encoder Init mode	0: Initialization in controlled sine commutation (determines the encoder offset) 1: Initialization in block commutation using hall sensors 2: Initialization in controlled sine commutation (use the previous set encoder offset)	0, 1, 2	RWE
250	Encoder steps	Encoder steps per rotation.	0... +65535	RWE
251	Encoder direction	Set the encoder direction in a way, that ROR increases position counter.	0/1	RWE
252	Hall interpolation	Select hall interpolation to interpolate the 16-bit FOC commutation angle between hall states.	0/1	RWE



Axis Parameter Description (Numbers 1- ...254)				
Number	Axis Parameter	Description	Range [Unit]	Access
253	Number of motor poles	Number of motor poles.	+2... +254	RWE
254	Hall sensor invert	1: Invert the hall scheme	0/1	RWE

*Table 57: Axis Parameter Description (Numbers 1 ...254)*



## 2.1 Axis Parameters Sorted by Functionality

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

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

Table 58: Functional Access Type Descriptions

Axis Parameter Motor Settings				
Number	Axis Parameter	Description	Range [Unit]	Access
253	Number of motor poles	Number of motor poles.	+2... +254	RWE

Table 59: Axis Parameter Motor Settings

Axis Parameter Encoder / Initialization Settings				
Number	Axis Parameter	Description	Range [Unit]	Access
31	BLDC re-initialization	Restart timer and bldc regulation.	(ignored)	W
159	Commutation mode	0: Block based on hall sensor 6: FOC based on hall sensor 7: FOC based on encoder 8: FOC controlled	0, 6, 7, 8	RWE
165	Actual encoder commutation offset	This value represents the internal commutation offset. (0 ... max. encoder steps per rotation)	0... 65535	RWE
177	Start current	Motor current for controlled commutation.	0... +14000 [mA]	RWE
210	Actual hall angle	Actual hall angle value	-32767... +32767	R
211	Actual encoder angle	Actual encoder angle value	-32767... +32767	R
212	Actual controlled angle	Actual controlled angle value	-32767... +32767	R
241	Sine initialization speed	Velocity during initialization in encoder init mode 2. Refer to axis parameter 249, too.	-200000... +200000 [rpm]	RWE



Axis Parameter Encoder / Initialization Settings				
Number	Axis Parameter	Description	Range [Unit]	Access
244	Init sine delay	Duration for sine initialization sequence. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	0... 10000 [ms]	RWE
249	Encoder init mode	0: Initialization in controlled sine commutation (determines the encoder offset) 1: Initialization in block commutation using hall sensors 2: Initialization in controlled sine commutation (use the previous set encoder offset)	0... 2	RWE
250	Encoder steps	Encoder steps per rotation.	0... +65535	RWE
251	Encoder direction	Set the encoder direction in a way, that ROR increases position counter.	0/1	RWE
252	Hall interpolation	Select hall interpolation to interpolate the 16-bit FOC commutation angle between hall states.	0/1	RWE
254	Hall sensor invert	1: Invert the hall scheme	0/1	RWE

Table 60: Axis Parameter Encoder / Initialization Settings

Torque Regulation Mode				
Number	Axis Parameter	Description	Range [Unit]	Access
6	Max current	Set/get the max allowed motor current. *This value can be temporarily exceeded marginal due to the operation of the current regulator.	0... +14000 [mA]	RWE
150	Actual motor current	Get actual motor current.	-2147483648... +2147483647 [mA]	R
155	Target current	Get desired target current or set target current to activate current regulation mode. (+ = turn motor in right direction; - = turn motor in left direction)	-14000... +14000 [mA]	RW
134	Current regulation loop delay	Delay of the current regulator.	0... +10 [50µs]	RWE
172	P parameter for current PID	P parameter of current PID regulator.	0... 65535	RWE
173	I parameter for current PID	I parameter of current PID regulator.	0... 65535	RWE
200	Current PID error	Actual error of current PID regulator	-2147483648... +2147483647	R
201	Current PID error sum	Error sum of current PID regulator	-2147483648... +2147483647	R





Torque Regulation Mode				
Number	Axis Parameter	Description	Range [Unit]	Access
208	Dual shunt factor	Manually adjust the dual shunt current measurement factor.		RWEX

Table 61: Torque Regulation Mode

### Velocity Regulation Mode

Velocity Regulation Mode				
Number	Axis Parameter	Description	Range [Unit]	Access
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
9	Motor halted velocity	If the actual speed is below this value the motor halted flag is set.	0 +200000 [rpm]	RWE
133	PID regulation loop delay	Delay of the position and velocity regulator	0... +20 [μs]	RWE
234	P parameter for velocity PID	P parameter of velocity PID regulator.	0... 65535	RWE
235	I parameter for velocity PID	I parameter of velocity PID regulator.	0... 65535	RWE
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

Table 62: Velocity Regulation Mode



## Velocity Ramp Parameter

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

Table 63: Velocity Ramp Mode

## Position Regulation Mode

Position Regulation Mode				
Number	Axis Parameter	Description	Range [Unit]	Access
1	Actual position	Set/get the position counter without moving the motor.	-2147483648... +2147483647	RW
0	Target position	The target position of a currently executed ramp.	-2147483648... +2147483647	RW
7	MVP Target reached velocity	Maximum velocity at which end position flag can be set. Prevents issuing of end position when the target is passed at high velocity.	0 +200000 [rpm]	RWE
10	MVP target reached distance	Maximum distance at which the position end flag is set.	0... +100000	RWE
161	Encoder set NULL	1: set position counter to zero at next N channel event.	0/1	RWE
163	Encoder clear set NULL	1: set position counter to zero only once 0: always at an N channel event	0/1	RWE
230	P parameter for position PID	P parameter of position PID regulator.	0... 65535	RWE
226	Position PID error	Actual error of PID position regulator	-2147483648... +2147483647	R

Table 64: Position Regulation Mode



## Axis Parameter Status Information

Axis Parameter Status Information				
Number	Axis Parameter	Description	Range [Unit]	Access
151	Actual voltage	Actual supply voltage.	0... +4294967295	R
152	Actual driver temperature	Actual temperature of the motor driver.	0... +4294967295	R
156	Error/Status flags	<p>Bit 0: Overcurrent flag. This flag is set if the max. current limit is exceeded.</p> <p>Bit 1: Undervoltage flag. This flag is set if supply voltage is too low for motor operation.</p> <p>Bit 2: Overvoltage flag. This flag is set if the motor becomes switched off due to overvoltage.</p> <p>Bit 3: Overtemperature flag. This flag is set if overtemperature limit is exceeded.</p> <p>Bit 4: Motor halted flag. This flag is set if the velocity does not reach the value set with GAP/SAP 9.</p> <p>Bit 5: Hall error flag. This flag is set upon a hall error.</p> <p>Bit 6: Driver error flag</p> <p>Bit 7: Init error flag</p> <p>Bit 8: Stop mode active flag</p> <p>Bit 9: Velocity mode active flag</p> <p>Bit 10: Position mode active flag.</p> <p>Bit 11: Torque mode active flag.</p> <p>Bit 12: unused</p> <p>Bit 13: unused</p> <p>Bit 14: Position end flag. This flag is set if the motor has been stopped at the target position.</p> <p>Bit 15: Module initialized flag</p> <p>Flag 0 to 15 are automatically reset.</p>	0...+4294967295	R

Table 65: Axis Parameter Status Information



## Driver Information

Driver Information				
Number	Axis Parameter	Description	Range [Unit]	Access
214	Driver diagnostic	Driver diagnostic value	0..1000 [0,1%]	R
215	Driver acknowledge	Acknowledge driver status.	(ignored)	W
216	Enable driver SPI	Disable the driver and initialize the driver SPI access.	(ignored)	W
217	Driver status register 2	Read/Write driver status register 2	-2147483648... +2147483647	RW
218	Driver status register 3	Read/Write driver status register 3	-2147483648... +2147483647	RW
219	Driver status register 4	Read/Write driver status register 4	-2147483648... +2147483647	RW

Table 66: Driver Information

## Brake Chopper

Brake Chopper				
Number	Axis Parameter	Description	Range [Unit]	Access
237	Enable brake chopper	Enable brake chopper functionality.	0 / 1	RWE
238	Brake chopper limit voltage	If the brake chopper is enabled and supply voltage exceeds this value, the brake chopper output will be activated.	50..480 [100mV]	RWE
239	Brake chopper hysteresis	An activated brake chopper will be disabled if the actual supply voltage is lower than (limit voltage-hysteresis).	0..50 [100mV]	RWE

Table 67: Brake Chopper

## I<sup>2</sup>T Monitoring

I <sup>2</sup> T Monitoring				
Number	Axis Parameter	Description	Range [Unit]	Access
25	Thermal winding time	Thermal winding time constant for the used motor. Used for I <sup>2</sup> t monitoring.	0..300000 [ms]	RWE
26	I <sup>2</sup> T limit	An actual I <sup>2</sup> t sum that exceeds this limit leads to increasing the I <sup>2</sup> t exceed counter.	0..270000000	RWE
27	I <sup>2</sup> T sum	Actual sum of the I <sup>2</sup> t monitor.	0..2147483647	R
28	I <sup>2</sup> T exceed counter	Counts how often an I <sup>2</sup> t sum was higher than the I <sup>2</sup> t limit.	0..4294967295	RWE
29	Clear I <sup>2</sup> T exceeded flag	Clear the flag that indicates that the I <sup>2</sup> t sum has exceeded the I <sup>2</sup> t limit.	(ignored)	W

Table 68: I<sup>2</sup>T Monitoring



## Single-Shunt Current Measurement

Single-Shunt Current Measurement				
Number	Axis Parameter	Description	Range [Unit]	Access
209	Single shunt factor	Manually adjust the single shunt current measurement factor.		RWEX
175	Single-shunt offset	The actual ADC offset of the single-shunt current measurement.	0..4095	RWE
176??				
205	Enable single-shunt measurement	Use single-shunt measurement for block commutation.	0..1	RWE

*Table 69: Single-Shunt Current Measurement*



### 3 Global Parameter Overview (SGP, GGP, STGP, RSGP)

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

#### Two banks are used for global parameters

- Bank 0 for global configuration of the module (chapter 4.1).
- Bank 2 for user TMCL variables (chapter 4.2)

#### 3.1 Bank 0

**Parameters 64... 255** Parameters from 64 upwards configure, the following:

- The serial address of the module.
- The UART baud rate.
- The telegram pause time.

#### Changing Parameters

The best and easiest way to change the parameters in order to meet your specification is to use the appropriate functions of the TMCL-IDE.

The parameters between 64 and 85 are stored in EEPROM automatically. A SGP command on such a parameter will always store it permanently and no extra STGP command is needed. Take care when changing these parameters and use the appropriate functions of the TMCL-IDE to do it in an interactive way!

Bank 0 Access Types		
Access Type	Related Commands	Description
R	GGP	Parameter readable
W	SGP, AGP	Parameter writable
E	STGP, RSGP	Parameter automatically restored from EEPROM after reset or power-on.

*Table 70: Bank 0 Access Types*



## Bank 0 Global Parameters

Bank 0 Global Parameters					
Number	Global Parameter	Description	Range	Access	
64	EEPROM magic	Setting this parameter to a different value as \$D0 will cause re-initialization of the axis and global parameters (to factory defaults) after the next power -up. This is useful in case of miss-configuration.	0... 255	RWE	
65	RS232/RS485 baud rate	0	9600 baud	0... 7	RWE
		1	14400 baud		
		2	19200 baud		
		3	28800 baud		
		4	38400 baud		
		5	57600 baud		
		6	76800 baud		
		7	115200 baud		
66	Serial address	The module (target) address for RS232 and RS485.	0... 255	RWE	
69	CAN bit rat	2	20 kBit/s	2....8	RWE
		3	50 kBit/s		
		4	100 kBit/s		
		5	125 kBit/s		
		6	250 kBit/s		
		7	500 kBit/s		
		8	1000 kBit/s		
70	CAN send ID	The actual send ID of the CAN interface.	0..65535	RWE	
71	CAN receive ID	The actual receive ID of the CAN interface.	0..65535	RWE	
73	Configuration EEPROM lock flag	Write: 1234 to lock the EEPROM, 4321 to unlock it. Read: 1=EEPROM locked, 0=EEPROM unlocked.	0/1	RWE	
75	Telegram pause time	Pause time before the reply via RS232 or RS485 is sent.	0... 255	RWE	
76	Serial host address	Host address used in the reply telegrams sent back via RS232 and RS485.	0... 255	RWE	
77	Auto start mode	0: Do not start TMCL application after power up (default). 1: Start TMCL application automatically after power up. Note: the current initialization has to be finished first.	0/1	RWE	



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

Table 71: Bank 0 Global Parameters





### 3.2 Bank 2

**General Purpose** Bank 2 contains general purpose 32 bit variables for use in TMCL applications.  
**Bit Variables in** They are located in RAM and can be stored to EEPROM. After booting, their values  
**Bank 2** are automatically restored to RAM. Up to 56 user variables are available.

Bank 2 Access Type Description		
Access Type	Related Commands	Description
R	GGP	Parameter readable
W	SGP, AGP	Parameter writable
E	STGP, RSGP	Parameter automatically restored from EEPROM after reset or power-on.

Table 72: Bank 2 Access Type Description

Bank 2 Global Parameters				
Number	Global Parameter	Description	Range	Access
0... 55	general purpose variable #0... 55	for use in TMCL applications	$-2^{31} \dots +2^{31}$ (int32)	RWE

Table 73: Bank 2 Global Parameters



## 4 Motor Regulation

### 4.1 Structure of Cascaded Motor Regulation Modes

The TMCC160-EVAL supports a current, velocity, and position PID regulation mode for motor control in different application areas. These regulation modes are cascaded as shown in [Figure 1](#) below. Individual modes are explained in the following sections.

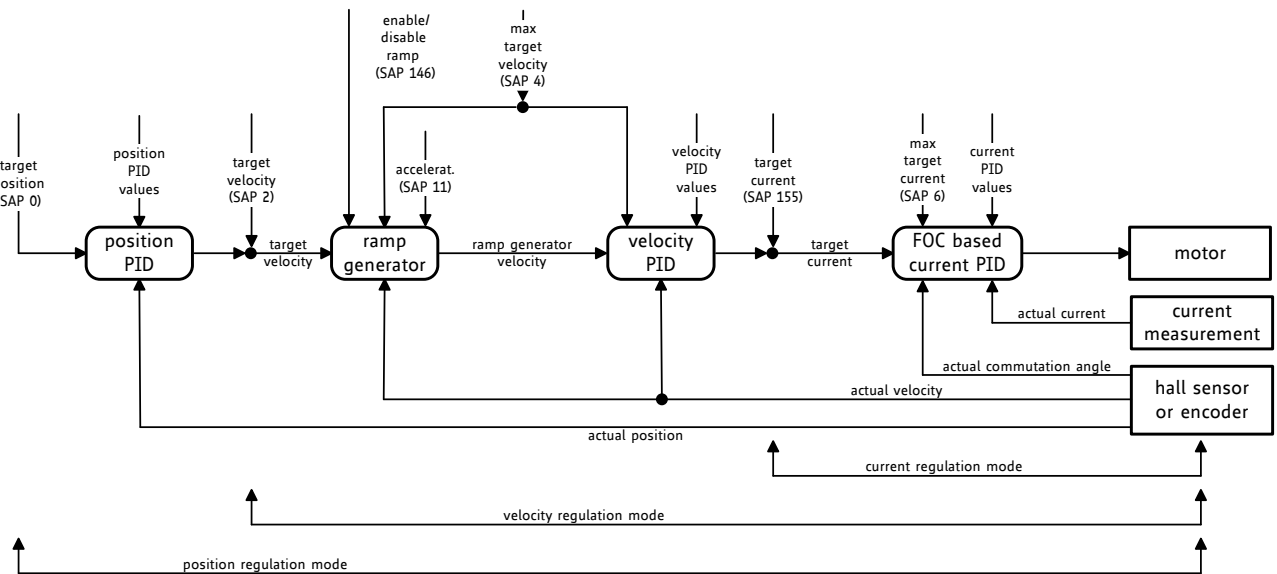


Figure 1: Cascaded Regulation

### 4.2 Current Regulation

The current regulation mode uses a PID regulator to adjust a desired motor current.

#### Axis Parameter 155: Target Current Settings

This target current can be set by axis parameter 155. The maximal target current is limited by axis parameter 6.

The PID regulation uses three basic parameters:

- The P value.
- The I value.
- The timing control value.

•→ Please turn page for more information on the timing control value.



**Timing Control Value** The timing control value (*current regulation loop multiplier*, axis parameter 134) determines how often the current regulation is invoked.

**Calculation Example**

It is given in multiple of 50µs:

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

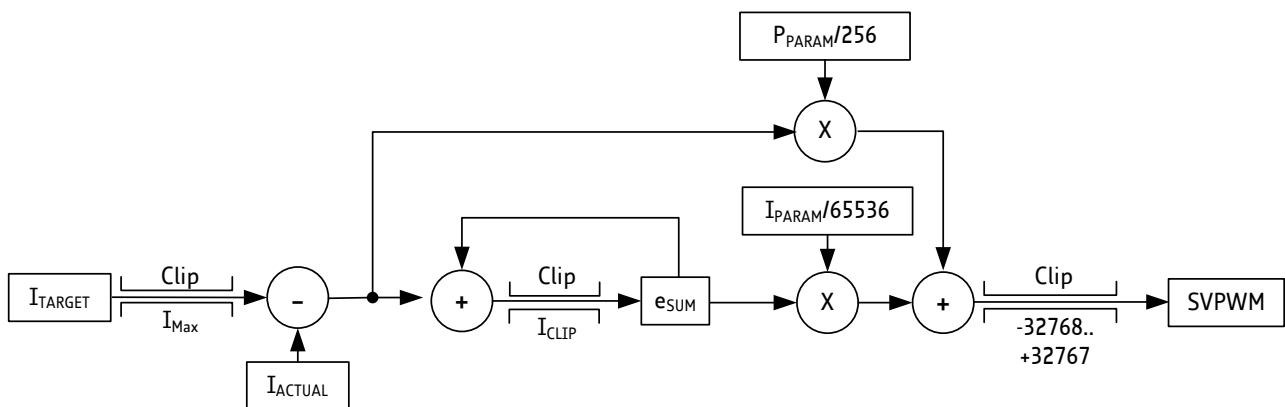
$t_{PIDDELAY}$  = resulting delay between two current regulation loops

$x_{PIDRLD}$  = current regulation loop multiplier parameter

- i For most applications it is recommended to leave this parameter unchanged – at its default of 1\*50µs. Higher values may be necessary for very slow and less dynamic drives.

*Calculation Examples 2: Current Regulator Timing Control Value*

**Structure of the Current Regulator**



*Figure 2: Current Regulation*

**Parameterizing the Current Regulator Set**

**In order to configure the current regulator, do as follows:**

**Action:**

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

**Result:**

Then, the actual motor current is close to the target current.

**NOTE:**

- See descriptions in the Table below.
- For all tests set the motor current limitation to a realistic value, so that your power supply does not become overloaded during acceleration phases. If your power supply reaches current limitation, the unit may reset or undetermined regulation results may occur.
- Turn page for list of Current Regulation Parameters ([Table 74](#))



## Current Regulation Parameters

Current Regulation Parameters	
Parameter	Description
I <sub>ACTUAL</sub>	Actual motor current (GAP 150)
I <sub>TARGET</sub>	Target motor current (SAP 155)
I <sub>Max</sub>	Max. motor current (SAP 6)
e <sub>SUM</sub>	Error sum for integral calculation (GAP 201)
P <sub>PARAM</sub>	Current P parameter (SAP 172)
I <sub>PARAM</sub>	Current I parameter (SAP 173)

Table 74: Current Regulation Set Parameters



### 4.3 Velocity Regulation

#### Timing Control Value

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 delay*, axis parameter 133) that determines how often the PID regulator is invoked. It is given in multiple of 50µs:

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

$t_{PIDDELAY}$  = resulting delay between two PID calculations

$x_{PIDRLD}$  = PID regulation loop delay parameter

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

#### Calculation Examples 3: Velocity Regulator Timing Control Value

#### Structure of the Velocity Regulator

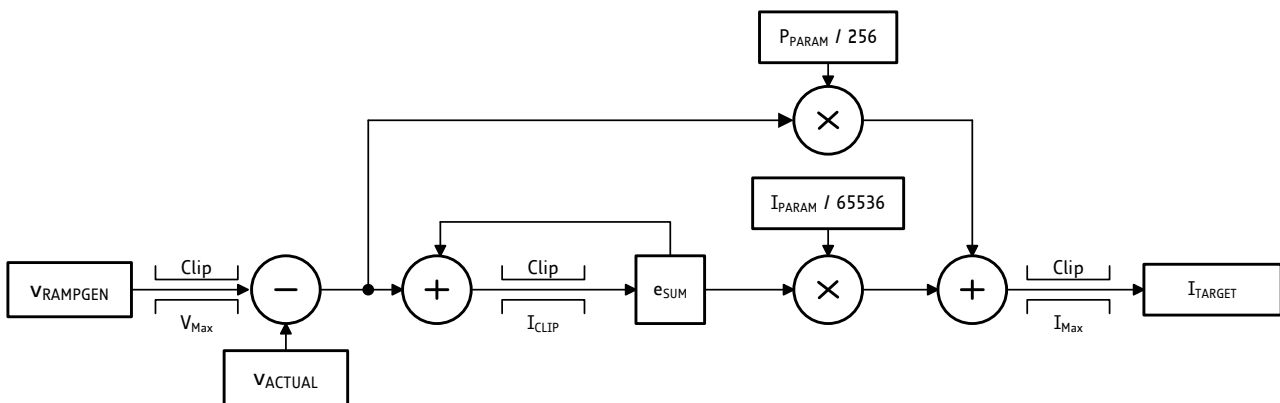


Figure 3: Velocity Regulation

#### Parameterizing the Velocity Regulator Set

**In order to configure the velocity regulator, do as follows:**

**Action:**

- Set the velocity I parameter to zero.
- Start the motor by using a medium target velocity (e.g. 2000 rpm).
- Modify the velocity P parameter. Start from a low value and go to a higher value, until the actual motor speed reaches 80 or 90% of the target velocity.
- The lasting 10 or 20% speed difference can be reduced by slowly increasing the velocity I parameter.

**Result:**

Then, the actual velocity is close to the target velocity.



## Parameter Description of Velocity Regulator Set

Parameter	Description
V <sub>ACTUAL</sub>	Actual motor velocity (GAP 3)
V <sub>RAMPGEN</sub>	Target velocity of ramp generator (SAP 2, GAP 13)
V <sub>Max</sub>	Max. target velocity (SAP 4)
e <sub>SUM</sub>	Error sum for integral calculation (GAP 229)
P <sub>PARAM</sub>	Velocity P parameter (SAP 234)
I <sub>PARAM</sub>	Velocity I parameter (SAP 235)
I <sub>Max</sub>	Max. target current (SAP 6)
I <sub>Target</sub>	Target current for current PID regulator (GAP 155)

*Table 75: Parameter Description of Velocity Regulator Set*



## 4.4 Velocity Ramp Generator

### Axis

For a controlled startup of the motor's velocity a velocity ramp generator can be activated or deactivated by axis parameter 146.

### 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.

## 4.5 Position Regulation

### Positioning

Based on current and velocity regulators, the TMCC160 supports a positioning mode configured with encoder or hall sensor position.

### Mode

### Configuration

During positioning the velocity ramp generator can be activated to enable motor positioning with controlled acceleration or it can be disabled to support motor positioning with max allowed speed.

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

### Timing Control

### Value

The timing control value (*PID regulation loop parameter* - axis parameter 133) determines how often the PID regulator is invoked. It is given in multiple of 50µs:

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

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

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

- i** For most applications it is recommended to leave the timing control value unchanged at its default of 50µs. Higher values may be necessary for very slow and less dynamic drives.

Calculation Examples 4: Position Regulator Timing Control Value

## Structure of the Position Regulator

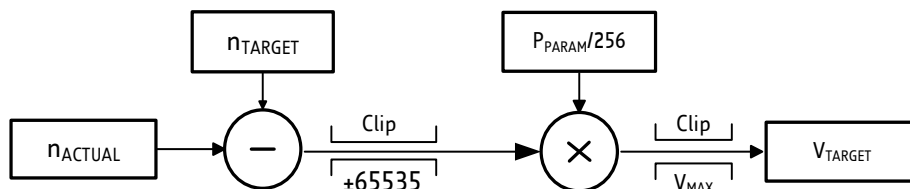


Figure 4: Positioning Regulation



**Parameterizing the Position Regulation**

Only the P parameter of the position regulator must to be parameterized because the regulator is based on the velocity regulator.

**In order to configure the position regular, do as follows:**

**Action:**

- Disable the velocity ramp generator and set position P parameter to zero.
  - Choose a target position and increase the position P parameter until the actual motor position is close to the target position.
  - Switch on the velocity ramp generator. Based on the max. Positioning velocity (axis parameter 4) and the acceleration value (axis parameter 11) the ramp generator automatically calculates the slow down point, i.e. the point at which the velocity has to be reduced in order to stop at the desired target position.
- i Reaching the target position is signaled by setting the position end flag.

**Result:**

Then, the actual position is close to the target position.

**Position Regulation Parameters**

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

Table 76: Position Regulation Parameters

**NOTE:**

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





### Correlation of Axis Parameters 10 and 7, the Target Position, and the Position End Flag

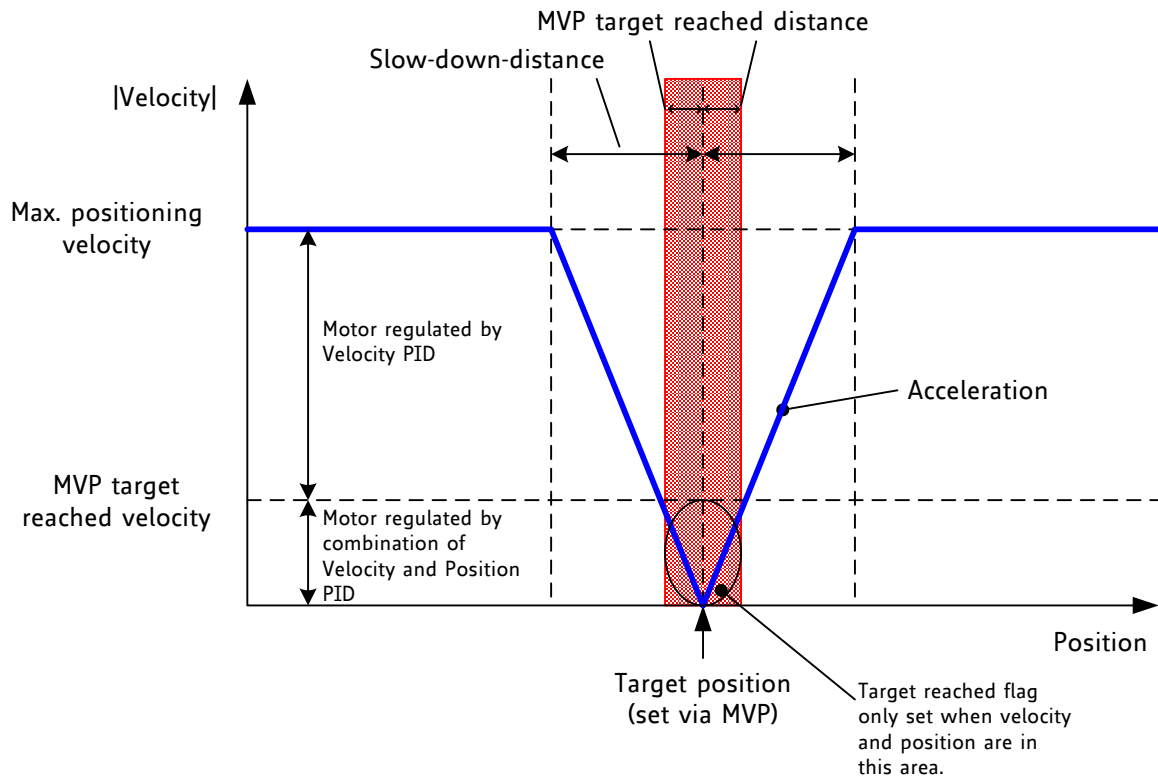


Figure 5: Positioning Algorithm

**NOTE:**

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



## 5 I<sup>2</sup>T-Monitoring

The I<sup>2</sup>t monitor determines the sum of the square of the motor current over a given time.

### Thermal Winding Time Constant of the Motor

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 a specific time span - depends on how often the current regulation and thus the I<sup>2</sup>t monitoring is invoked.

### Configuration of I<sup>2</sup>T Monitor

**In order to configure the I<sup>2</sup>T monitor, the following parameters need to be known and set accordingly:**

- The value of the actual I<sup>2</sup>t sum can be read by axis parameter 27.
- With axis parameter 26 the default value for the I<sup>2</sup>t limit can be changed.
- If the actual I<sup>2</sup>t sum exceeds the I<sup>2</sup>t limit of the motor, flag 17 (in axis parameter 156) is set and the motor PWM is set to zero as long as the I<sup>2</sup>t exceed flag is set.

#### **NOTE:**

- Also note that the I<sup>2</sup>t exceed counter is increased once every second as long as the actual I<sup>2</sup>t sum exceeds the I<sup>2</sup>t limit.
- The I<sup>2</sup>t 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 I<sup>2</sup>t exceed flag will not be reset automatically. The I<sup>2</sup>t limit can be determined as follows:

The I<sup>2</sup>t limit can be determined as follows:

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

$I$  is the desired average current

$t_{tw}$  is the thermal winding time constant given by the motor datasheet

*Calculation Examples 5: Determine I<sup>2</sup>t limit*



**Example:**  
**Configuration of**  
**the I<sup>2</sup>t Limits**

I<sup>2</sup>t limits for an average current of a) 1A, b) 2A, c) 3A and d) 4A over a thermal winding time of 10,0s.

$$\text{a) } I^2t \text{ limit} = \frac{1000 \text{ [mA]}}{1000} * \frac{1000 \text{ [mA]}}{1000} * 10000 \text{ [ms]} = 10000 \text{ [mA}^2 * \text{ms]}$$

$$\text{b) } I^2t \text{ limit} = \frac{2000 \text{ [mA]}}{1000} * \frac{2000 \text{ [mA]}}{1000} * 10000 \text{ [ms]} = 40000 \text{ [mA}^2 * \text{ms]}$$

$$\text{c) } I^2t \text{ limit} = \frac{3000 \text{ [mA]}}{1000} * \frac{3000 \text{ [mA]}}{1000} * 10000 \text{ [ms]} = 90000 \text{ [mA}^2 * \text{ms]}$$

$$\text{d) } I^2t \text{ limit} = \frac{4000 \text{ [mA]}}{1000} * \frac{4000 \text{ [mA]}}{1000} * 10000 \text{ [ms]} = 160000 \text{ [mA}^2 * \text{ms]}$$

*Calculation Examples 6: I<sup>2</sup>t Configuration Limits*



## APPENDICES

### 6 Supplemental Directives

#### ESD DEVICE INSTRUCTIONS



**TMCC160 is an ESD-sensitive CMOS device. It is sensitive to electrostatic discharge.**

- Provide effective grounding to protect personnel and machines.
- Ensure work is performed in a non-static environment.
- Use personal ESD control footwear and ESD wrist straps, if necessary.

**Failure to do so can result in defects, damages and decreased reliability.**

#### Producer Information

The producer of the product TMCC160 is TRINAMIC GmbH & Co. KG in Hamburg, Germany; hereafter referred to as TRINAMIC. TRINAMIC is the supplier; and in this function provides the product and the production documentation to its customers

#### Copyright

TRINAMIC owns the content of this user manual in its entirety, including but not limited to pictures, logos, trademarks, and resources.

© Copyright 2015 TRINAMIC®. All rights reserved. Electronically published by TRINAMIC®, Germany. All trademarks used are property of their respective owners.

Redistributions of source or derived format (for example, Portable Document Format or Hypertext Markup Language) must retain the above copyright notice, and the complete Datasheet User Manual documentation of this product including associated Application Notes; and a reference to other available product-related documentation.

#### Target User

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

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

•→Please turn page for more information.



**Disclaimer: Life Support Systems**

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

Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

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

**Disclaimer: Intended Use**

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

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

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



## 7 Safety Alerts

In this chapter we provide an overview of all *Safety Alerts* that are provided in this documentation. Please refer to the appropriate chapters and/or sections for detailed information about the nature of the *Safety Alerts*. Thank you.

List of Safety Alerts in this Documentation	
Type	Page
ESD Device Instructions	<a href="#">52</a>

Table 77: List of Safety Alerts in this Documentation

## 8 References

### 8.1 Revision History

#### Firmware Revision History (FW V.)

Version	Date	Description
2.08	2015-AUG-21	Initial version.
2.09	2015-NOV-25	Parameters for ADC scaler added.

Table 78: Firmware Revision History

#### ▷ Firmware Document Revision History (DC Rev.)

Version	Date	Author/s	Description
1.00	2015-DEC-04	SV/ED	Initial version.

Table 79: Document Revision History

#### Hardware Revision History (HW V.)

Version	Date	Description
1.00	2014-OCT-10	Initial version.

Table 80: Hardware Revision History

#### **NOTE:**

→ Please refer to documentation available on product webpage at [www.trinamic.com](http://www.trinamic.com).

### 8.2 Related Documents and Tools

[TMCC160-EVAL]

[TMCL-IDE]

[TMCL]

Go to product page on: [www.trinamic.com](http://www.trinamic.com)



### 8.3 Table Index

Table 1: RS232 and RS485 TMCL Request Format.....	4
Table 2: RS253 and RS485 TMCL Reply Format.....	6
Table 3: TMCL Reply Status Codes .....	6
Table 4: TMCL Motion Commands.....	7
Table 5: TMCL Parameter Commands.....	7
Table 6: TMCL I/O Port Commands .....	7
Table 7: ROR (Rotate Right) Reply in Direct Mode .....	8
Table 8: ROR Example: Rotate right with Velocity=350 .....	8
Table 9: ROL (Rotate Left) Request in Direct Mode .....	9
Table 10: ROL (Rotate Left) Reply in Direct Mode.....	9
Table 11: ROL Example: Rotate left with Velocity=1200.....	9
Table 12: MST (Motor Stop) Request in Direct Mode .....	10
Table 13: MST (Motor Stop) Reply in Direct Mode.....	10
Table 14: MST Example: Stop Motor at Mnemonic: MST 0.....	10
Table 15: MVP ABS/ REL Request in Direct Mode .....	11
Table 16: MVP ABS / REL Reply in Direct Mode .....	11
Table 17: MVP ABS Example: Move Motor to Absolute Position 9000.....	12
Table 18: (MVP REL Example: Move Motor 1000 Steps to Relative Position .....	12
Table 19: SAP (Set Axis Parameter) Request in Direct Mode.....	12
Table 20: SAP (Set Axis Parameter) Reply in Direct Mode .....	12
Table 21: SAP Example: Absolute Max. Current 2000MA: .....	13
Table 22: GAP (Get Axis Parameter) Request in Direct Mode .....	13
Table 23: GAP (Get Axis Parameter) Reply in Direct Mode.....	13
Table 24: GAP Request Example: Get actual Position of Motor 0.....	13
Table 25: GAP Reply Example .....	14
Table 26: STAP (Store Axis Parameter) Request in Direct Mode.....	14
Table 27: STAP (Store Axis Parameter) Reply in Direct Mode .....	14
Table 28: STAP Example: Store Maximum Speed: STAP 4,0.....	14
Table 29: RSAP Request in Direct Mode.....	15
Table 30: RSAP Reply in Direct Mode .....	15
Table 31: RSAP Example: Restore Maximum Motor Current 0 .....	15
Table 32: SGP (Set Global Parameter) Request in Direct Mode .....	16
Table 33: SGP (Set Global Parameter) Reply in Direct Mode.....	16
Table 34: SGP Example: Mnemonic: SGP, 0, 2, 100 .....	16
Table 35: GGP (Get Global Parameter) Request Direct Mode.....	17
Table 36: GGP (Get Global Parameter) Reply in Direct Mode .....	17
Table 37: GAP Example: Get Variable 0 from Bank 2 .....	17
Table 38: STGP Request in Direct Mode.....	18
Table 39: STGP Reply in Direct Mode .....	18
Table 40: STGP Example: Restore Variable 0 of bank 2 to EEPROM Configuration .....	18
Table 41: RSGP (Restore Global Parameter) Request in Direct Mode.....	19
Table 42: RSGP (Restore Global Parameter) Reply in Direct Mode .....	19
Table 43: RSGP: Copy Variable 0 at Bank 2 from EEPROM Configuration to RAM Location .....	19
Table 44: Correlation between I/Os and Banks .....	20
Table 45: SIO Request in Direct Mode.....	21
Table 46: SIO Reply in Direct Mode.....	21



Table 47: SIO Reply Example .....	21
Table 48: GIO (Get Input / Output) Request in Direct Mode .....	22
Table 49: GIO (Get Input / Output) Reply in Direct Mode .....	22
Table 50: GIO Request Example .....	22
Table 51: GIO Reply Example.....	22
Table 52: Available SIO and GIO Commands.....	23
Table 53: TMCL Command 136 Request in Direct Mode .....	23
Table 54: TMCL Command 136 Reply in Direct Mode, Type Set to 0 .....	23
Table 55: TMCL Command 136, Reply in Direct Mode, Type Set to 1 .....	24
Table 56: Access Type Description.....	25
Table 57: Axis Parameter Description (Numbers 1 ...254).....	30
Table 58: Functional Access Type Descriptions.....	31
Table 59: Axis Parameter Motor Settings .....	31
Table 60: Axis Parameter Encoder / Initialization Settings .....	32
Table 61: Torque Regulation Mode .....	33
Table 62: Velocity Regulation Mode.....	33
Table 63: Velocity Ramp Mode .....	34
Table 64: Position Regulation Mode .....	34
Table 65: Axis Parameter Status Information .....	35
Table 66: Driver Information .....	36
Table 67: Brake Chopper.....	36
Table 68: I <sup>2</sup> T Monitoring .....	36
Table 69: Single-Shunt Current Measurement.....	37
Table 70: Bank 0 Access Types .....	38
Table 71: Bank 0 Global Parameters .....	40
Table 72: Bank 2 Access Type Description.....	41
Table 73: Bank 2 Global Parameters .....	41
Table 74: Current Regulation Set Parameters.....	44
Table 75: Parameter Description of Velocity Regulator Set .....	46
Table 76: Position Regulation Parameters.....	48
Table 77: List of Safety Alerts in this Documentation .....	54
Table 78: Firmware Revision History .....	54
Table 79: Document Revision History .....	54
Table 80: Hardware Revision History .....	54





## 8.4 Figure Index

Figure 1: Cascaded Regulation .....	42
Figure 2: Current Regulation.....	43
Figure 3: Velocity Regulation.....	45
Figure 4: Positioning Regulation.....	47
Figure 5: Positioning Algorithm.....	49



## 8.5 Keyword Index

5.5 Position Regulation .....	47	Overview	
Axis Parameter Overview (SAP, GAP, STAP, RSAP) ....	25	TMCL .....	4
Checksum Calculation .....	5	Position Regulation Parameters.....	48
Copyright .....	52	Producer Information .....	52
Current Regulation .....	42	References	
Disclaimer: Intended Use.....	53	Related Documents and Tools .....	54
Disclaimer: Life Support Systems.....	53	Revision History .....	54
ESD-DEVICE INSTRUCTIONS .....	52	Short Spec	
Example: Rotate right with velocity = 350: Mnemonic:		Features .....	2
ROR 0, 350 .....	8	Structure of Cascaded Motor Regulation Modes .....	42
Global Parameter Overview (SGP, GGP, STGP, RSGP)	38	Supplemental Directives .....	52
I <sup>2</sup> T-Monitoring .....	50	TABLE OF CONTENTS.....	2
Motor Regulation.....	42	Target User.....	52
		Timing Control Value .....	47
		Velocity Ramp Generator.....	47
		Velocity Regulation .....	45

