TMCM-1613 FIRMWARE MANUAL

TMCM-1613 Firmware Version 1.00 • 2016-MAR-28 | Document Revision 1.00 • 2016-MAR-28

The TMCM-1613 firmware performs hall sensor-based block commutation using single-shunt current measurement on the TMCM-1613 module. The firmware supports TMCL(TM) commands for standalone operation as well as remote control through an UART interface. Additionally, it implements an analog mode of operation with voltage-controlled speed, which can be selected at startup time. The pre-installed boot loader provides support for firmware updates through the UART interface.

Figure 1:TMCM-1613 Product Photo

Features

- Block commutation for BLDC motors
- Motor current up to 30A Peak
- Supply voltage 6...24V DC
- Configured with TMCL™ software
- TMCL-IDE updates via UART
- Cascaded motor regulation modes

Order Code

Order code	Description	Size
TMCM-1613	1-Axis BLDC controller/driver 500W/24V with PWM and Step/Dir control, hall sensor based	70 x 76 x 25 mm³

Table 1: TMCM-1613 Order Codes

Read entire documentation; especially the Supplemental Directives in Chapter 6 (page 53)

TABLE OF CONTENTS

TMCM-1613 FIRMWARE MANUAL ... 1
SHORT SPEC ... 1
Features ... 1
Order Code ... 1
TABLE OF CONTENTS ... 2
FIRMWARE MANUAL ... 4
1. Basic TMCL Formats and Commands ... 4
1.1. Request Format ... 5
1.2. Reply Format ... 6
1.3. Parameter Commands ... 7
1.4. I/O Port Commands ... 7
2. Detailed TMCL Commands Description ... 8
2.1. ROR (Rotate Right) ... 8
2.2. ROL (Rotate Left) ... 9
2.3. MST (Motor Stop) ... 10
2.4. MVP (Move to Position) ... 11
2.5. SAP (Set Axis Parameter) ... 13
2.6. GAP (Get Axis Parameter) ... 14
2.7. STAP (Store Axis Parameter) ... 15
2.8. RSAP (Restore Axis Parameter) ... 16
2.9. SGP (Set Global Parameter) ... 17
2.10. GGP (Get Global Parameter) ... 18
2.11. STGP (Store Global Parameter) ... 19
2.12. RSGP (Restore Global Parameter) ... 20
2.13. SIO (Set Output) and GIO (Get Input / Output) ... 21
2.14. SIO (Set Output) ... 22
2.15. GIO (Get Input / Output) ... 23
2.16. TMCL Control Functions ... 25
3. Axis Parameter Overview (SAP, GAP, STAP, RSAP) ... 26
3.1. Axis Parameters Sorted by Functionality ... 30
4. Global Parameter Overview (SGP, GGP, STGP, RSGP) ... 36
4.1. Bank 0 ... 36
4.2. Bank 2 ... 39
5. Motor Regulation ... 40
5.1. Structure of Cascaded Motor Regulation Modes ... 40
5.2. Current Regulation ... 41
5.3. Structure of the Current Regulator ... 42
5.4. Velocity Regulation ... 44

Read entire documentation; especially the "Supplemental Directives" (page 53).

5.4.1. Structure of the Velocity Regulator 44
5.5. Velocity Ramp Generator 47
5.6. Position Regulation 47

6. Module Specific Behaviors 50

6.1. Default Mode 52
6.2. DC Mode 52
6.3. Analog Mode 52
6.4. Mixed Mode 52

APPENDICES 53

7. Supplemental Directives 53

7.1. Figures Index 55
7.2. Tables Index 56
7.3. Revision History 58

Read entire documentation; especially the "Supplemental Directives" (page 53).

FIRMWARE MANUAL

1. Basic TMCL Formats and Commands

TMCL is a motion control-oriented command set that provides pre-configures, easily adabtable totation (ROR, ROL) and positioning commands (MVP). TMCL is designed to quickly connect motors to a TMCM module. The TMCM-1613 firmware quickly connects to a three-phase motor.

ℹ TMCL is available for TRINAMIC board level solutions and also for PANdrive.

TMCL-1613 has extensive Command Set

In order to configure the module to your design specification an extensive command set for all necessary motor control parameters is made available to you. Configuration options are explained in detail in this manual.

TMCM-1613 firmware runs on a microprocessor and consists of two parts:

- **Boot loader:**
  The boot loader is installed by TRINAMIC during production. It remains untouched throughout its entire product lifetime.
- **Firmware**:
  The firmware can be updated by the user. New versions can be downloaded free of charge from the product's web page [TMCM-1613].

Functional Scope of TMCM-1613

In this manual the focus is entirely on how to use the TMCM-1613 firmware for the TMCM-1613 module in order to control a 3-phase motor according to your design specification. The TMCM-1613 firmware supports standard TMCL with an additional specified range of parameters and values.

Firmware sample code and TMCL sample scripts are available on the product's web page.

The TMCM module is based on Freescale KE ARM Cortex-M0+ microcontroller and the high performance pre-driver TMCM-1613.

NOTE:

→ *The firmware is related to the standard TMCL firmware [TMCL] with regard to protocol and commands.*

→ *The TMCL firmware is available for USB and field buses like RS232, RS485 and CAN but for configuration of the TMCM-1613 only an UART interface is made available.*

→ *TMCL can be used as script language only. For more information, please refer to the TMCL manual at www.trinamic.com .*

Read entire documentation; especially the "Supplemental Directives" (page 53).

1.1. Request Format

When commands are sent from a host to a module, the request format has to be used.

Process Description of Request Format

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.

AREAS OF SPECIAL CONCERN !

When a command is sent via UART interface, it must be enclosed by an address byte at the beginning and by a checksum byte at te end.

In this case it consists of nine bytes. The binary command format for UART and USB is structured as follows:

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

Table 2:TMCL Request Format

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: TMCL Request Format

Read entire documentation; especially the "Supplemental Directives" (page 53).

1.2. Reply Format

Whenever a command is sent to a module, the module sends a reply.

TMCL Reply Format Structure

The reply format for UART and USB is structured as follows:

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 3: TMCL Reply Format

TMCL Reply Status Code

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 4: TMCL Reply Status Codes

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 5: TMCL Motion Commands

Read entire documentation; especially the "Supplemental Directives" (page 53).

1.3. 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 6: TMCL Parameter Commands

1.4. I/O Port Commands

Direct Mode and Standalone Mode

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 7: TMCL I/O Port Commands

Read entire documentation; especially the "Supplemental Directives" (page 53).

2. Detailed TMCL Commands Description

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

2.1. ROR (Rotate Right)

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

Process Description:

Internal function:

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

Related commands: ROL, MST, SAP, GAP

Mnemonic: ROR 0, <velocity>

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

Table 8: ROR Request in Direct Mode

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

Table 9: ROR 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**
Function	Target Address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$01	$00	$00	$00	$00	$01	$5e

Table 10: ROR Example: Rotate right with Velocity=350

Read entire documentation; especially the "Supplemental Directives" (page 53).

2.2. ROL (Rotate Left)

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

Process Description:

Internal function:

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

Related commands: ROR, MST, SAP, GAP
Mnemonic: ROL 0, <velocity>

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

Table 11: ROL Request in Direct Mode

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

Table 12: ROL 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**
Function	Target Address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$02	$00	$00	$00	$00	$04	$b0

Table 13: ROL Example: Rotate left with Velocity=1200

Read entire documentation; especially the "Supplemental Directives" (page 53).

2.3. MST (Motor Stop)

The motor is instructed to stop.

Process Description:

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

Related commands: ROL, ROR, SAP, GAP
Mnemonic: MST 0

ℹ An example for MST is provided in Table 16.

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

Table 14: MST (Motor Stop) Request in Direct Mode

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

Table 15: 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**
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$03	$00	$00	$00	$00	$00	$00

Table 16: MST Example: Stop Motor at Mnemonic: MST 0

Read entire documentation; especially the “Supplemental Directives” (page 53).

2.4. MVP (Move to Position)

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

Two available Operation Types

The motor is instructed to move to a specified relative or absolute position. It uses the predefined acceleration/deceleration ramp and the positing 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:**
  Moving to an absolute position in the range from:
  -2147483648… +2147483647.
- **REL:**
  Starting a relative movement by means of an offset to the actual position.

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

Related commands: SAP, GAP, and MST
Mnemonic: MVP <ABS|REL>, 0, <position|offset value>

MVP 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 17: MVP ABS/ REL Request in Direct Mode

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

Table 18: 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 19: 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 20: (MVP REL Example: Move Motor 1000 Steps to Relative Position

2.5. SAP (Set Axis Parameter)

Most of the motion control parameters of the module can be specified by using the SAP command.

SRAM Settings Process Description

The settings are stored in SRAM and therefore are volatile. Thus, information is lost after power-off.

Related commands: GAP, STAP, and RSAP
Mnemonic: SAP <parameter number>, 0, <value>

NOTE:

→ *You must use command STAP in order to store your specified setting permanently.*
→ *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 21: SAP (Set Axis Parameter) Request in Direct Mode

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

Table 22: SAP (Set Axis Parameter) Reply in Direct Mode

ℹ A list of all parameters that 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	Host-address	Target-Address	Status	Instruction	OperandByte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$05	$06	$00	$00	$00	$07	$D0

Table 23: SAP Example: Absolute Max. Current 2000MA:

2.6. GAP (Get Axis Parameter)

SRAM Settings Process Description

Most parameters of the TMC6130-EVAL can be adjusted individually. They can be read out using the GAP command.

Related commands: SAP, STAP, and RSAP
Mnemonic: GAP <parameter number>, 0

NOTE:

→ *A GAP request example is provided in Table 26, 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 24: GAP (Get Axis Parameter) Request in Direct Mode

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

Table 25: 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 26: 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 27: GAP Reply Example

2.7. STAP (Store Axis Parameter)

STAP Settings stored in SRAM

The STAP command stores an axis parameter previously set with a Set Axis Parameter command (SAP) permanently.

ℹ Most parameters are automatically restored after power up.

Internal function:
An axis parameter 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

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

Table 28: STAP (Store Axis Parameter) Request in Direct Mode

[1] *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 29: 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 30: 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.

2.8. RSAP (Restore Axis Parameter)

For all configuration related axis parameters non-volatile memory locations are provided.

Resetting a single Parameter

By default, most parameters are automatically restored after power up. A single parameter that has been changed before can be reset by this instruction also.

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

Related commands: SAP, STAP, and GAP
Mnemonic: RSAP <parameter number>, 0

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 31: RSAP Request in Direct Mode

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

Table 32: 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 Maxium 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 33: RSAP Example: Restore Maximum Motor Current 0

2.9. SGP (Set Global Parameter)

Global parameters are related to the host interface, peripherals or other application specific variables.

Organization of Parameters in Banks

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>

An example for SGP is provided below:

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

Table 34: SGP (Set Global Parameter) Request in Direct Mode

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

Table 35: 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 36: SBP Example: Mnemonic: SGP, 0, 2,100

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>

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

Table 37: GGP (Get Global Parameter) Request in Direct Mode

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

Table 38: 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.

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

Table 39:GGP Example: Get Variable 0 from Bank 2

Read entire documentation; especially the "Supplemental Directives" (page 53).

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>

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

Table 40: STGP Request (Store Global Parameter) in Direct Mode

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

Table 41: STGP Reply (Store Global Parameter) in Direct Mode

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

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

Table 42: STGP Example: Restore Variable 0 to Bank 2 to EEPROM Configuration

2.12. RSGP (Restore Global Parameter)

RSGP Configuration

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.

Related commands: SGP, GGP, STGP
Mnemonic: RSGP <parameter number>, <bank number>

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

Table 43: RSGP Request (Store Global Parameter) in Direct Mode

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

Table 44: RSGP Reply (Store Global Parameter) in Direct Mode

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

RSGP Example: Restore Variable 0 to Bank 2 to EEPROM Configuration: Mnemonic: RSGP, 0, 2								
Byte Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function	Target Address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$01	$0c	$00	$02	$00	$00	$00	$00

Table 45: RSGP Example: Restore Variable 0 to Bank 2 to EEPROM Configuration

Read entire documentation; especially the "Supplemental Directives" (page 53).

2.13. SIO (Set Output) and GIO (Get Input / Output)

Command Settings

The TMC6130-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 46: Correlation between I/Os (SIO and GIO) and Banks

2.14. 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).

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

Related commands: GIO, WAIT
Mnemonic: SIO <port number>, <bank number>, <value>

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

Table 47: SIO Request in Direct Mode

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

Table 48: SIO Reply in Direct Mode

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

Table 49: SIO Example

2.15. GIO (Get Input / Output)

Two Options: Direct Mode or Standalone Mode

***GIO* can be used in direct mode or in standalone mode.**
In standalone mode, the requested value is copied to the accumulator (accu) for further processing purposes; such as conditioned jumps.

OPTION 1: IN STANDALONE MODE

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

OPTION 2: IN DIRECT MODE

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

Internal function: The specified line is read.

Related commands: SIO, WAIT
Mnemonic: GIO <port number>, <bank number>

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

Table 50: GIO Request in Direct Mode

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

Table 51: GIO Reply in Direct Mode

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

Table 52: GIO Request Example

GIO Reply Example								
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)	$02	$01	$64	$0f	$00	$00	$01	$2e

Table 53: GIO Reply Example

Read entire documentation; especially the "Supplemental Directives" (page 53).

Available *SIO* and *GIO* Commands					
I/O	**Digital**	**Analog**	**GIO <port>, <bank>**	**SIO <port>, <bank>, <value>**	**Value Range**
Digital input 0	X	-	GIO 0, 0	-	0/1
Digital input 1	X	-	GIO 1, 0	-	0/1
VIN+ Detect	X	-	GIO 2, 0	-	0/1
ADC Input 0	-	X	GIO 0, 1	-	0... 4095
ADC single shunt	-	X	GIO 1, 1	-	0... 4095
ADC VSupply	-	X	GIO 2, 1	-	0... 4095
ADC Motor Temp	-	X	GIO 3, 1	-	0... 4095
UART1-Tx State	X	-	GIO 0, 2	-	0/1
UART1-Tx	X	-	-	SIO 0, 2	0/1

Table 54: Available SIO and GIO Commands

2.16. TMCL Control Functions

TMCL Control Command 136

There are several TMCL control functions. The most important one for users is command 136.

- ℹ Other control functions can be used with axis parameters.
- ℹ Two possible reply examples are provided below.

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 55: TMCL Command 136: Request in Direct Mode

Command 136: Request in Direct Mode Type set to 0. Reply as a String:[1]	
Byte index	**Contents**
1	Host Address
2… 9	Version string (8 characters, e.g. 1613V101)

Table 56: TMCL Command 136: Request in Direct Mode

[1] There is no checksum in this reply format!

TMCL Control Functions: Type Set to 1. Version Number in Binary Format:[1]	
Byte Index in Value Field	**Contents**
1	Version number, low byte.
2	Version number, high byte.
3	Type number, low byte.
4	Type number, high byte.

Table 57: TMCL Control Funtions: Type Set to 1.

[1] The version number is output in the value field.

3. 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 58: 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… +30000 [mA]	RWE
7	MVP Target reached velocity	Maximum velocity at which end position flag can be set. Prevents issuing of end position when the target is passed at high velocity.	0… +200000 [rpm]	RWE
9	Motor halted velocity	If the actual speed is below this value the motor halted flag will be set.	0 +200000 [rpm]	RWE
10	MVP target reached distance	Maximum distance at which the position end flag is set.	0… +100000	RWE
11	Acceleration	Acceleration parameter for ROL, ROR, and the velocity ramp of MVP.	0… +100000 [RPM/s]	RWE
13	Ramp generator speed	The actual speed of the velocity ramp used for positioning and velocity mode.	-2147483648… +2147483647 [rpm]	R

Axis Parameter Description (Numbers 1-254)				
Number	**Axis Parameter**	**Description**	**Range [Unit]**	**Access**
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 [µs]	RWE
134	Current regulation loop delay	Delay of the current regulator.	2… +10 [50µs]	RWE
146	Activate ramp	1: Activate velocity ramp generator for position and velocity mode. (Allows usage of acceleration and positioning velocity for MVP command.)	0/1	RWE
150	Actual motor current	Get actual motor current.	-2147483648… +2147483647 [mA]	R
151	Actual voltage	Actual supply voltage.	0… +4294967295	R
152	Actual driver temperature	Actual temperature of the motor driver.	0… +4294967295	R
155	Target current	Get desired target current or set target current to activate current regulation mode. (+= turn motor in right direction; -= turn motor in left direction)	-30000… +30000 [mA]	RW

•→*Continued on next page!*

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…+4294967295	R
159	Commutation mode	0: Block based on hall sensor	0	R
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	Start single-shunt offset measurement	Triggers a measurement of the single-shunt offset and reads if it's ready.	0… 1	RW
175	Single-shunt offset	Offset of the single-shunt current measurement.	0… 4096	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
214	Driver diagnostic	Driver diagnostic value	0..1000 [0,1%]	R

Axis Parameter Description (Numbers 1-254)				
Number	**Axis Parameter**	**Description**	**Range [Unit]**	**Access**
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
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
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 59: Axis Parameter Description (Numbers 1-254)

Read entire documentation; especially the "Supplemental Directives" (page 53).

3.1. 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	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 60: 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 61: Axis Parameter Motor Settings

Axis Parameter 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	0	R
254	Hall sensor invert	1: Invert the hall scheme	0/1	RWE

Table 62: Axis Parameter Encoder / Initialization Settings

•→*Turn page for torque regulation mode.*

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… +30000 [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)	-30000… +30000 [mA]	RW
134	Current regulation loop delay	Delay of the current regulator.	2… +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

Table 63: Torque Regulation Mode

•→Turn page for 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 will be 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 64: Velocity Regulation Mode

Read entire documentation; especially the "Supplemental Directives" (page 53).

Velocity Ramp Parameter

Velocity Regulation 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 65: Velocity Regulation 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
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 66: Position Regulation Mode

Read entire documentation; especially the "Supplemental Directives" (page 53).

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	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...+4294967295	R

Table 67: Axis Parameter Status Information

Read entire documentation; especially the "Supplemental Directives" (page 53).

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 68: Driver Information

Read entire documentation; especially the "Supplemental Directives" (page 53).

4. 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 Bank Settings are available

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)

4.1. Bank 0

Parameters 64… 255

Configuration Details

Parameters from 64 upwards configure, can be configured by setting the serial address of the module, the UART baud rate, and the telegram pause time.

!

Change these parameters to meet your needs.
The best and easiest way to do this is to use the appropriate functions of the TMCL-IDE:

- The parameters between 64 and 85 are stored in EEPROM automatically.
- A SGP command on such a parameter will always store it permanently and no extra STGP command is needed.

NOTE:

→ *Take special care when changing these parameters.*
→ *Use the appropriate functions of the TMCL-IDE to do this 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 69: Bank 0 Access Types

Read entire documentation; especially the "Supplemental Directives" (page 53).

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	UART baud rate	0 | 9600 baud | Default *Higher baudrates are not available in this module.*	0	R
66	Serial address	The module (target) address for RS232 and virtual COM port	0… 255	RWE
73	Configuration EEPROM lock flag	Write: 1234 to lock the EEPROM, 4321 to unlock it. Read: 1=EEPROM locked, 0=EEPROM unlocked.	0/1	RWE
75	Telegram pause time	Pause time before the reply via RS232 is sent. For RS232 set to 0.	0… 255	RWE
76	Serial host address	Host address used in the reply telegrams sent back via RS232.	0… 255	RWE
77	Auto start mode	0: Do not start TMCL application after power up (default). 1: Start TMCL application automatically after power up. Note: the current initialization has to be finished first.	0/1	RWE
78	Module specific behavior	0: Default mode: module is controlled by UART. ADC Input 0, Digital Input 0 and UART-Tx Digital Output pins are not available. 1: DC mode: velocity is controlled by supply voltage (target velocity) 0...24V correspond to 0...Max Ramp Velocity. Direction is inverted if voltage is negative. Digital Input 1 is used as Stop pin. ADC Input 0, Digital Input 0 and UART-Tx Digital Output pins are not available. 2: Analog Mode: velocity is controlled by Analog Input 0 (target velocity). 0...15V correspond to 0...Max Ramp Velocity. Direction is controlled by Digital Input 0. Digital Input 1 is used as Stop pin. UART communication is not available. UART-Tx Digital Output pin is not available. 3: Mixed mode: Digital Input 1 is used as stop pin. The module is able to receive UART commands, but not to answer them. UART-Tx Digital Output is in this mode available. Analog Input 0 and Digital Input 0 are not available. See an extended description in Section 5: Module Specific Behaviors. *Note: this parameter must be stored explicit by STGP! Analog Mode cannot be set as initial mode by STGP.*	0/1/2/3	RWE

Read entire documentation; especially the "Supplemental Directives" (page 53).

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 Attention: Download mode can only be used if the motor has been stopped first. Otherwise the download mode setting will be disallowed. During download mode the motor driver will be deactivated and the actuator will be turned off.	0/1	R
130	TMCL program counter	The index of the currently executed TMCL instruction.	0… 2047	R
132	Tick timer	A 32 bit counter that gets incremented by one every millisecond. It can also be reset to any start value.	0… +4294967295	RW
255	Suppress reply	0 – reply (default) 1 – no reply	0/1	RW

Table 70: Bank 0 Global Parameters

Read entire documentation; especially the "Supplemental Directives" (page 53).

4.2. Bank 2

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

ℹ 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 71: 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	$-2^{31}...+2^{31}$ (int32)	RWE

Table 72: Bank 2 Global Parameters

5. Motor Regulation

5.1. Structure of Cascaded Motor Regulation Modes

The TMCM-1613 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 5.1 below:

ℹ Individual modes are explained in the following sections.

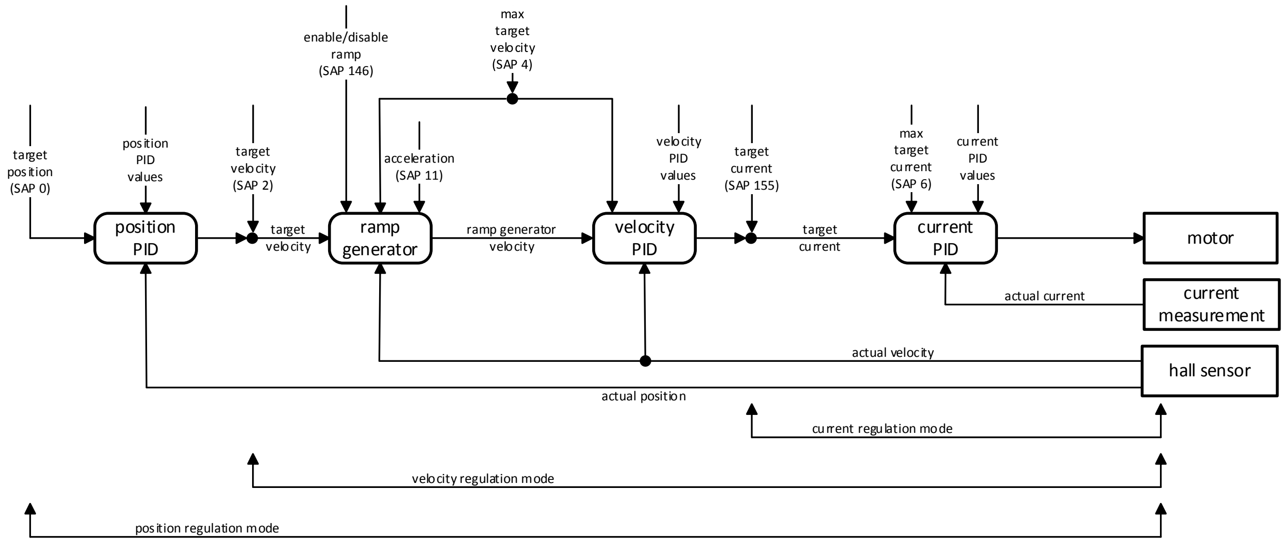

Figure 2: Cascaded Regulation

Read entire documentation; especially the "Supplemental Directives" (page 53).

5.2. 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*.

Timing Control Value

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

It is given in multiple of 50µs:

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

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

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.

5.3. Structure of the Current Regulator

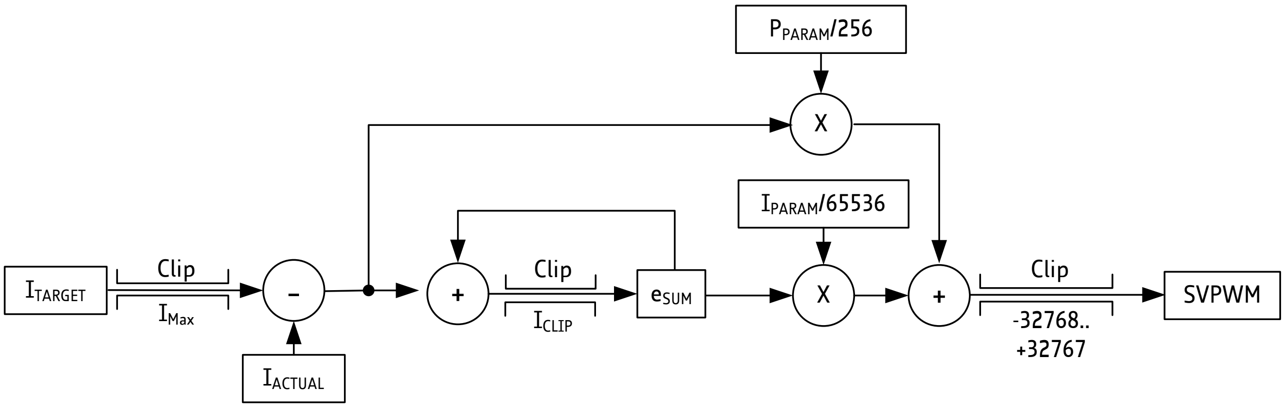

Figure 3: Current Regulation

Parameterizing the Current Regulator Set

In order to parameterize the current regulator set, do as follows:

Action:

- Set the P parameter and the 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:
Proper setting of parameters.

See descriptions in the Table below:

NOTE:

→ *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.*

Read entire documentation; especially the "Supplemental Directives" (page 53).

Description of Current Regulation Set Parameters

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

Table 73: Current Regulation Set Parameters

5.4. Velocity Regulation

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

Timing Control Value

The velocity PID regulator uses a timing control value (*PID regulation loop delay*, axis parameter 133) which determines how often the PID regulator is invoked.

It is given in multiple of 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.

5.4.1. Structure of the Velocity Regulator

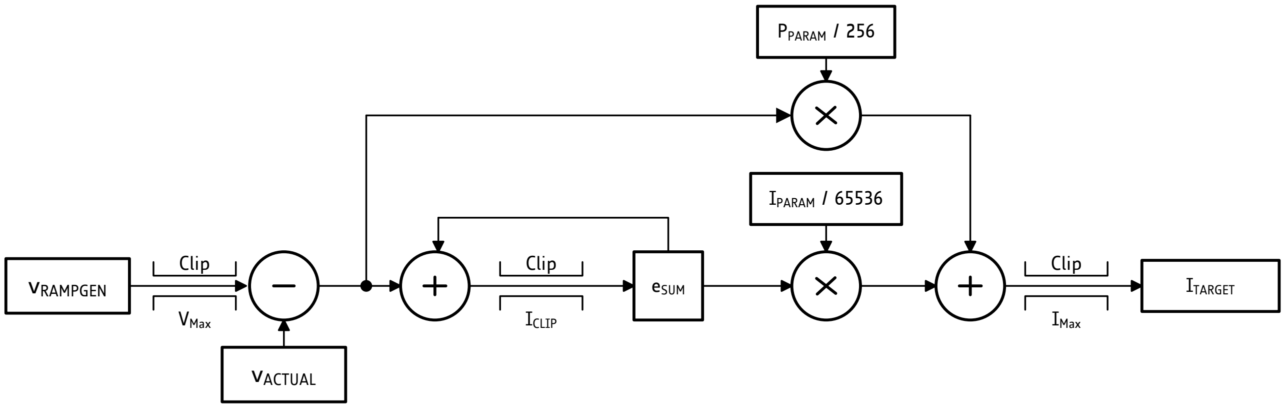

Figure 4: Velocity Regulation

Parameterizing the Velocity Regulator Set

In order to parameterize the velocity regulator set, 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.
  - i Start from a low value and go to a higher value, until the actual motor speed reaches 80 or 90% of the target velocity.
  - i The lasting 10 or 20% speed difference can be reduced by slowly increasing the velocity I parameter.

Result:
The velocity regulator set is now set.

•→*Turn page for parameter description table of velocity regulator set.*

Read entire documentation; especially the "Supplemental Directives" (page 53).

Read entire documentation; especially the "Supplemental Directives" (page 53).

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

Table 74: Parameter Description of Velocity Regulator Set

Read entire documentation; especially the "Supplemental Directives" (page 53).

5.5. Velocity Ramp Generator

For a controlled startup 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.

5.6. Position Regulation

Based on current and velocity regulators, the TMC6130-EVAL supports a positioning mode configured with encoder or hall sensor position.

Positioning 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.

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.

Structure of the Position Regulator

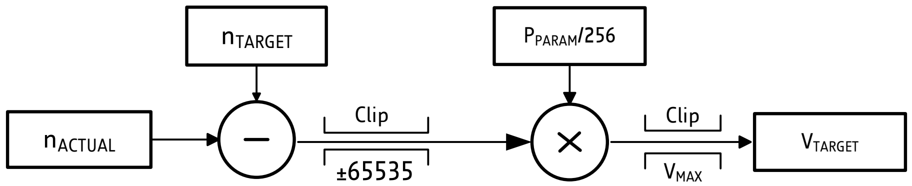

Figure 5: Positioning Regulation

Read entire documentation; especially the "Supplemental Directives" (page 53).

Parameterizing the Position Regulation

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

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 motor reaches the target position approximately.
- 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.

Result:
The position regular is set.

NOTE:

→ *Reaching the target position is signaled by setting the position end flag.*

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 75: 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

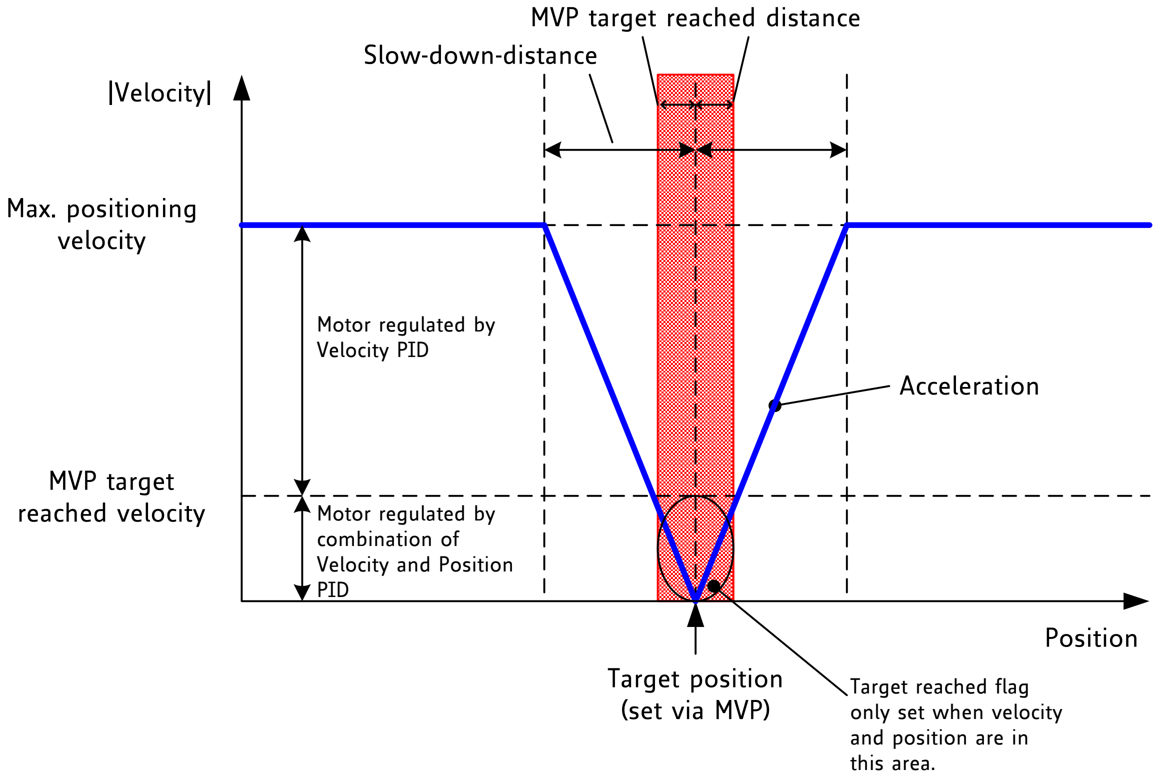

Figure 6: Positioning Algorithm

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!*

6. Module Specific Behaviors

The TMCM-1613 has four different modes available, which powers the module with specific behaviors and pin configurations. A

Handle Module-Specific Behavior

At start-time, it will be decided in which mode the module starts, based on the global parameter *Handle Module-Specific Behavior* (GP 78) and the voltage at the Analog Input 0 pin.

The following diagram reflects the start-up procedure:

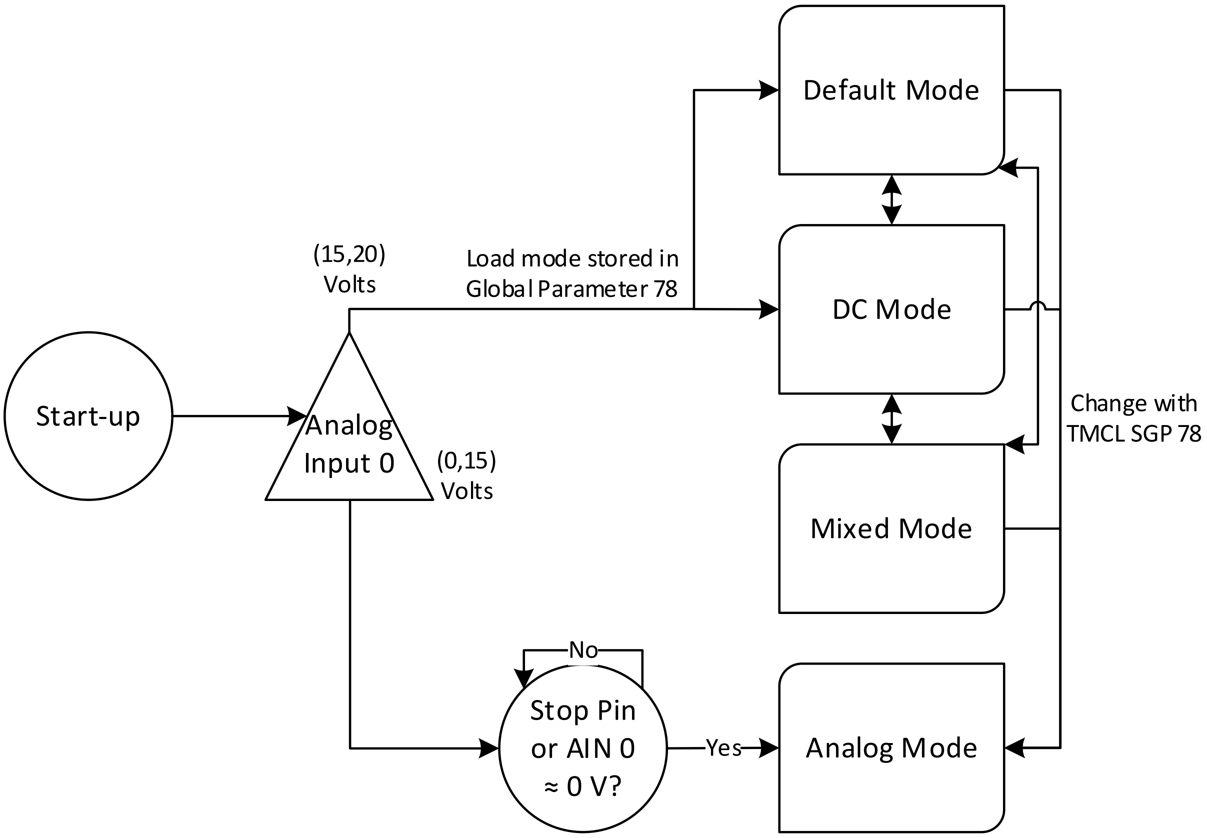

Figure 7: Start-up Procedure of Module Specific Behavior

Process Description

When the module starts, it first reads the voltage at the Analog Input 0 and decides if it is under 15 Volts or beyond. Between 15 and 20 Volts, the mode stored in Global Parameter 78 will be the first one loaded.

AREAS OF SPECIAL CONCERN **!**

It is important to connect the Dongle (TMCM-1613-USB) to the USB before start-up, otherwise the voltage at the analog input will not be ready when the module tries to read it.

After that, with a SGP 78 command, any other mode can be reached. When a STGP is sent, the current mode will be stored in the EEPROM memory to be loaded the next time at start-up. During Default or DC Mode, a STGP can be sent, so that the current mode will be stored in the EEPROM memory and be loaded the next time at start-up.

•→*Turn page for more information pertaining to this issue.*

AREAS OF SPECIAL CONCERN

Continued

!

On the other hand, if the voltage at Analog Input 0 is under 15 V during the start-up, the first mode loaded will be Analog Mode.
This mode will not actually start until either the Stop Pin is set or the voltage is 0, so that the motor doesn't immediately start to rotate with a high speed. If at the beginning one of this conditions is fulfilled, the mode will be activated at that time. From Analog Mode, no other mode can be reached, since the UART communication is not available. Neither can be this mode stored in Global Parameter 78, because the STGP TMCL command will not be received by the module, and because in that case, the other three modes could never be used.

The pins and functionalities that are different in the different modes are presented in the next table:

Available pins and functions in each Specific Behavior Mode				
Functionality	Mode Availability			
	Default	DC	Analog	Mixed
UART Communication	Available	Available	Not available	Module receives but does not answer
Analog Input 0	Not available	Not available	Available	Not available
Digital Input 0	Not available	Not available	Available	Not available
Digital Input 1	Available	Available (Stop pin)	Available (Stop pin)	Available (Stop pin)
UART-Tx Digital Output	Not available	Not available	Not available	Available

Table 76: Available pins and functions in each Specific Behavior Mode

Read entire documentation; especially the "Supplemental Directives" (page 53).

6.1. Default Mode

TMCL Command through UART interface

In the Default Mode, the TMCM-1613 operates the usual way like in other Trinamic modules. It reacts to TMCL commands through the UART interface. The motor can be controlled through rotate commands or move to position (MVP) commands. The configuration can be set or read back by axis or global parameters. In this mode, there is a digital input available for the user (Digital Input 1).

6.2. DC Mode

Configuration Description

In this mode, the module is in Velocity Mode and the target velocity is set by the supply voltage instead of by the rotate motor commands. The velocity range is between 0 and the value written in Max Ramp Velocity (Axis Parameter 4), and corresponds to voltages between 0 and 24 V. The same velocities can be achieved with negative voltages from 0 to -24 V, where -24 V corresponds to Max Ramp Velocity in negative direction. This means that low speeds that correspond to voltages between 0 and the minimum voltage (positive or negative) required to power the TMCM-1613 are not available in this mode.

The Digital Input 1 is used as Stop pin. When it is set to 1, the motor will stay still. Value 0 indicates right direction and value 1 indicates left direction. The negative voltage detection is performed by the additional circuit board TMCM-1613-REC, which transforms the voltage sign into the Digital Input VIn+ Detect.

ℹ UART communication is fully available in this mode.

6.3. Analog Mode

Configuration Description

The Analog Input 0 controls the target velocity when the current mode is Analog Mode. The velocity range is between 0 and the value written in Max Ramp Velocity (Axis Parameter 4), and corresponds to voltages between 0 and 10 V. Higher voltages (up to 20 V) will write a target speed equal to Max Ramp Velocity.

The Digital Input 1 is used as Stop pin. When it is set to 1, the motor will stay still. The direction is controlled by the Digital Input 0. Value 0 indicates right direction and value 1 indicates left direction.

ℹ UART communication is fully available in this mode.

6.4. Mixed Mode

Configuration Description

In the Mixed Mode, the TMCM-1613 reacts to TMCL commands through the UART interface but does not answer to them. The motor can be controlled through rotate commands or move to position (MVP) commands. The configuration can be set by axis or global parameters.

ℹ In this mode, the Digital Input 1 is used as Stop pin and there is a digital output available for the user (Digital Input UART-Tx).

Read entire documentation; especially the "Supplemental Directives" (page 53).

APPENDICES

7. Supplemental Directives

Producer Information

The producer of the product TMCM-1613 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.

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.

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.

Product Documentation Details

This Firmware Manual contains the ***User Information*** for the ***Target User***.

Collateral Documents & Tools

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

Read entire documentation; especially the "Supplemental Directives" (page 53).

7.1. Figures Index

Figure 1:TMCM-1613 Product Photo 1
Figure 1: Cascaded Regulation 40
Figure 2: Current Regulation... 42
Figure 3: Velocity Regulation... 44
Figure 4: Positioning Regulation.. 47
Figure 5: Positioning Algorithm 49
Figure 6: Start-up Procedure of Module Specific Behavior 50

Read entire documentation; especially the "Supplemental Directives" (page 53).

7.2. Tables Index

Table 1: TMCM-1613 Order Codes 1
Table 2:TMCL Request Format 5
Table 3: TMCL Reply Format 6
Table 4: TMCL Reply Status Codes 6
Table 5: TMCL Motion Commands 6
Table 6: TMCL Parameter Commands 7
Table 7: TMCL I/O Port Commands 7
Table 8: ROR Request in Direct Mode 8
Table 9: ROR Reply in Direct Mode 8
Table 10: ROR Example: Rotate right with Velocity=350 8
Table 11: ROL Request in Direct Mode 9
Table 12: ROL Reply in Direct Mode 9
Table 13: ROL Example: Rotate left with Velocity=1200 9
Table 14: MST (Motor Stop) Request in Direct Mode 10
Table 15: MST (Motor Stop) Reply in Direct Mode 10
Table 16: MST Example: Stop Motor at Mnemonic: MST 0 10
Table 17: MVP ABS/ REL Request in Direct Mode 11
Table 18: MVP ABS / REL Reply in Direct Mode 11
Table 19: MVP ABS Example: Move Motor to Absolute Position 9000 12
Table 20: (MVP REL Example: Move Motor 1000 Steps to Relative Position 12
Table 21: SAP (Set Axis Parameter) Request in Direct Mode 13
Table 22: SAP (Set Axis Parameter) Reply in Direct Mode 13
Table 23: SAP Example: Absolute Max. Current 2000MA: 13
Table 24: GAP (Get Axis Parameter) Request in Direct Mode 14
Table 25: GAP (Get Axis Parameter) Reply in Direct Mode 14
Table 26: GAP Request Example: Get actual Position of Motor 0 14
Table 27: GAP Reply Example 14
Table 28: STAP (Store Axis Parameter) Request in Direct Mode 15
Table 29: STAP (Store Axis Parameter) Reply in Direct Mode 15
Table 30: STAP Example: Store Maximum Speed: STAP 4,0 15
Table 31: RSAP Request in Direct Mode 16
Table 32: RSAP Reply in Direct Mode 16
Table 33: RSAP Example: Restore Maximum Motor Current 0 16
Table 34: SGP (Set Global Parameter) Request in Direct Mode 17
Table 35: SGP (Set Global Parameter) Reply in Direct Mode 17
Table 36: SBP Example: Mnemonic: SGP, 0, 2,100 17
Table 37: GGP (Get Global Parameter) Request in Direct Mode 18
Table 38: GGP (Get Global Parameter) Reply in Direct Mode 18
Table 39:GGP Example: Get Variable 0 from Bank 2 18
Table 40: STGP Request (Store Global Parameter) in Direct Mode 19
Table 41: STGP Reply (Store Global Parameter) in Direct Mode 19
Table 42: STGP Example: Restore Variable 0 to Bank 2 to EEPROM Configuration 19
Table 43: RSGP Request (Store Global Parameter) in Direct Mode 20
Table 44: RSGP Reply (Store Global Parameter) in Direct Mode 20
Table 45: RSGP Example: Restore Variable 0 to Bank 2 to EEPROM Configuration 20
Table 46: Correlation between I/Os (SIO and GIO) and Banks 21
Table 47: SIO Request in Direct Mode 22
Table 48: SIO Reply in Direct Mode 22
Table 49: SIO Example 22

Read entire documentation; especially the "Supplemental Directives" (page 53).

Table 50: GIO Request in Direct Mode 23
Table 51: GIO Reply in Direct Mode 23
Table 52: GIO Request Example 23
Table 53: GIO Reply Example 23
Table 54: Available SIO and GIO Commands 24
Table 55: TMCL Command 136: Request in Direct Mode 25
Table 56: TMCL Command 136: Request in Direct Mode 25
Table 57: TMCL Control Funtions: Type Set to 1. 25
Table 58: Access Type Description 26
Table 59: Axis Parameter Description (Numbers 1-254) 29
Table 60: Functional Access Type Descriptions 30
Table 61: Axis Parameter Motor Settings 30
Table 62: Axis Parameter Encoder / Initialization Settings 30
Table 63: Torque Regulation Mode 31
Table 64: Velocity Regulation Mode 32
Table 65: Velocity Regulation Mode 33
Table 66: Position Regulation Mode 33
Table 67: Axis Parameter Status Information 34
Table 68: Driver Information 35
Table 69: Bank 0 Access Types 36
Table 70: Bank 0 Global Parameters 38
Table 71: Bank 2 Access Type Description 39
Table 72: Bank 2 Global Parameters 39
Table 73: Current Regulation Set Parameters 43
Table 74: Parameter Description of Velocity Regulator Set 46
Table 75: Position Regulation Parameters 48
Table 76: Available pins and functions in each Specific Behavior Mode 51
Table 77: Firmware Revision History 58
Table 78: Document Revision History 58

Read entire documentation; especially the “Supplemental Directives” (page 53).

7.3. Revision History

Firmware Revision History		
Version	**Date**	**Description**
1.00	2016-MAR-28	Initial firmware version.

Table 77: Firmware Revision History

Document Revision History			
Version	**Date**	**Author/s**	**Description**
1.00	2016-MAR-28	BS/SV	Initial documentation

Table 78: Document Revision History

Read entire documentation; especially the "Supplemental Directives" (page 53).