EVALBOARD USER MANUAL

Section: Firmware Manual

TMC6130-EVAL

Evaluation Board Firmware:

TMC6130 3-Phase Motor Driver

- **30A Peak / 6...30V DC**
- **USB / TTL-UART**
- **Hall Sensor Interface**
- **Encoder Interface**

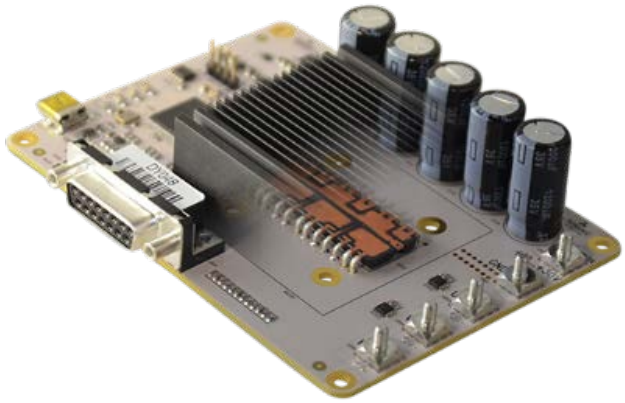

Image 1: TMC6130-EVAL

Document published online 2015-SEP-04. Available at www.trinamic.com

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

TRINAMIC
MOTION CONTROL

SUPPLEMENTAL DIRECTIVES

Read the entire documentation before you make use of this product:

- Ensure you meet the *Target User Requirements* on page 47.
- Read the *User Information and Directives* provided on page 47.
- Read the *ESD-Sensitivity Directives* provided on page 5.

Keep this manual and all other applicable and related documents complete, legible and accessible to the specified user at all times.

Failure to observe the *Supplemental Directives* could result in damage to product and things; to property or persons; or economic loss.

TRINAMIC is not liable for damages incurred as a result of improper use or disregard of the instructions provided in this *document.*

Please refer to our *User Information and Directives* for more details (page 47).

Contact *TRINAMIC support team*

In case you encounter difficulties or need additional advice, please contact our support team via: www.trinamic.com. Thank you.

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

TRINAMIC
MOTION CONTROL

TABLE OF CONTENTS

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

TRINAMIC
MOTION CONTROL

ESD-DEVICE INSTRUCTIONS

This product 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.

1 Features

The TMC6130-EVAL supports block commutation based on hall sensors for BLDC motors as well as 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-, and encoder-sensor settings and to find incorrectly connected hall sensor signals.

Application

Evaluation of the features of the TMC6130 three phase motor driver

Electrical Data

- Motor current up to 30A Peak
- Supply voltage from 6V up to 30V

Interfaces

- USB
- TTL-UART (with pin header)
- General purpose digital inputs
- General purpose analog inputs
- Digital hall sensor
- Incremental a/b/n encoder

Motor Type

- Three-phase BLDC or PMSM motor
- Sinusoidal commutation with space vector PWM (SVPWM) based on hall sensor or encoder feedback for PMSM motors
- Block commutation based on hall sensor for BLDC motors
- Rotor position feedback (encoder or hall sensor)

Software

- Available with TMCL™ firmware
- Stand-alone operation or remote controlled operation
- 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 USB and UART using the TMCL-IDE

2 TMCL Overview

The software that runs on the microprocessor of the TMC6130-EVAL 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 [TMC6130-EVAL].

The firmware is related to the standard TMCL firmware [TMCL] with regard to protocol and commands. The module is based on the Freescale K20 ARM Cortex-M4 microcontroller and the high performance pre-driver TMC6130. It supports standard TMCL with a special range of parameters and values. Firmware sample code and TMCL sample scripts are available on the product's web page.

2.1 Basic TMCL Formats and Commands

2.1.1 Request Format

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

Therefore, the binary representation of a command always has seven bytes. When a command is sent via UART or USB interface, it has to be enclosed by an address byte at the beginning and by a checksum byte at the 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 1: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
```

2.1.2 Reply Format

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

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 2: 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

2.1.3 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

2.1.4 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

2.1.5 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

2.2 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.

2.2.1 ROR (Rotate Right)

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

Internal function:

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

Related commands: ROL, MST, SAP, GAP

Mnemonic: ROR 0, <velocity>

Binary Representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	1	don't care

Example to rotate right with velocity = 350: Mnemonic: ROR 0, 350

Binary:

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

2.2.2 ROL (Rotate Left)

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

Internal function:

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

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

Binary Representation:

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

Reply in Direct Mode:

STATUS	COMMAND	VALUE
100 – OK	2	don't care

Example to rotate left with velocity = 1200: Mnemonic: ROL 0, 1200

Binary:

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

2.2.3 MST (Motor Stop)

The motor is instructed to stop.

Internal function: The axis parameter *target velocity* is set to zero.
Related commands: ROL, ROR, SAP, GAP
Mnemonic: MST 0

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	3	don't care

Example to stop the motor: Mnemonic: MST 0

Binary:

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

2.2.4 MVP (Move to Position)

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

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

Two operation types are available:

- 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>

Binary Representation:

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

Reply in Direct Mode:

STATUS	COMMAND	VALUE
100 – OK	4	don't care

Example to move motor to absolute position 9000: Mnemonic: MVP ABS, 0, 9000

Binary:

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

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

Binary:

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

2.2.5 SAP (Set Axis Parameter)

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>

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	5	don't care

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

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

Binary:

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

2.2.6 GAP (Get Axis Parameter)

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

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	6	don't care

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

Example to get the actual position of motor 0: Mnemonic: GAP 1, 0

Binary:

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

Reply:

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

2.2.7 STAP (Store Axis Parameter)

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

Binary representation:

COMMAND	TYPE	MOT/BANK	VALUE
7	<parameter number>	0	don't care[1]

Table Note ℹ **Please take note of the following notes concerning the table above:**

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	7	don't care

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

Example to store the maximum speed: Mnemonic: STAP 4, 0

Binary:

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

AREAS OF SPECIAL CONCERN ! **The STAP command will not have any effect when the configuration EEPROM is locked. The error code 5 (configuration EEPROM locked) will be returned in this case.**

2.2.8 RSAP (Restore Axis Parameter)

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

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

Internal function: The specified parameter is copied from the configuration EEPROM memory to its RAM location.
Related commands: SAP, STAP, and GAP
Mnemonic: RSAP <parameter number>, 0

Binary representation:

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

Reply in direct mode:

STATUS	COMMAND	VALUE
100 – OK	8	don't care

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

Example to restore the maximum current of motor 0: Mnemonic: RSAP 6, 0

Binary:

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

2.2.9 SGP (Set Global Parameter)

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

i The different groups of these parameters are organized in banks to allow a larger total number for future products. Currently, bank 0 is used for global parameters and bank 2 is intended for user variables.

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

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	don't care

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

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

Binary:

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

2.2.10 GGP (Get Global Parameter)

All global parameters can be read with this function.

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

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	<value>

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

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

Binary:

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

2.2.11 STGP (Store Global Parameter)

AREAS OF SPECIAL CONCERN

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>

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	don't care

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

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

Binary:

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

2.2.12 RSGP (Restore Global Parameter)

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>

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	don't care

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

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

Binary:

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

2.2.13 SIO (Set Output) and GIO (Get Input / Output)

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.
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

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 7: Correlation between I/Os and Banks

2.2.13.1 SIO (Set Output)

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

Internal function: The passed value is transferred to the specified output line.
Related commands: GIO, WAIT
Mnemonic: SIO <port number>, <bank number>, <value>

Binary representation:

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

Reply structure:

STATUS	VALUE
100 – OK	don't care

Binary:

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

2.2.13.2 GIO (Get Input / Output)

***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.

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

Internal function: The specified line is read.
Related commands: SIO, WAIT
Mnemonic: GIO <port number>, <bank number>

Binary representation:

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

Reply in direct mode:

STATUS	VALUE
100 – OK	<status of the port>

Binary:

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

Reply:

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

Provided *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
Digital input 2	X	-	GIO 2, 0	-	0/1
Analog input 0	-	X	GIO 0,1	-	0... 4095
Analog input 1	-	X	GIO 1,1	-	0... 4095
ADC single shunt	-	X	GIO 2,1	-	0... 4095
ADC phase A	-	X	GIO 3,1	-	0... 4095
ADC phase B	-	X	GIO 4,1	-	0... 4095
ADC VSupply	-	X	GIO 5,1	-	0... 4095
Digital output 0 (PTD1)	X	-	GIO 0, 2	SIO 0, 2	0/1
Digital output 1 PTD2	X	-	GIO 1, 2	SIO 1, 2	0/1
Digital output 2 PTD3	X	-	GIO 2, 2	SIO 2, 2	0/1
Digital output 3 PTD6	X	-	GIO 3, 2	SIO 3, 2	0/1

Table 8: Provided SIO and GIO Commands

2.2.14 TMCL Control Functions

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

Other control functions can be used with axis parameters.

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

Type set to 0 - reply as a string:

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

Table Note **Please take note of the following notes concerning the table above:**
There is no checksum in this reply format!

Type set to 1 - version number in binary format:

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 Note **Please take note of the following notes concerning the table above:**
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 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 9: Access Type Description

Axis Parameter 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

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
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.	0... +10 [50µs]	RWE
146	Activate ramp	1: Activate velocity ramp generator for position and velocity mode. (Allows usage of acceleration and positioning velocity for MVP command.)	0/1	RWE
150	Actual motor current	Get actual motor current.	-2147483648... +2147483647 [mA]	R
151	Actual voltage	Actual supply voltage.	0... +4294967295	R
152	Actual driver temperature	Actual temperature of the motor driver.	0... +4294967295	R
155	Target current	Get desired target current or set target current to activate current regulation mode. (+= turn motor in right direction; -= turn motor in left direction)	-30000... +30000 [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...+4294967295	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
177	Start current	Motor current for controlled commutation.	0... +30000 [mA]	RWE
200	Current PID error	Actual error of current PID regulator	-2147483648... +2147483647	R

Axis Parameter Description (Numbers 1-254)				
Number	**Axis Parameter**	**Description**	**Range [Unit]**	**Access**
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
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
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
241	Sine initialization speed	Velocity during initialization in init sine mode 2. Refer to axis parameter 249, too.	-200000... +200000 [rpm]	RWE
244	Init sine delay	Duration for sine initialization sequence. This parameter should be set in a way, that the motor has stopped mechanical oscillations after the specified time.	0... 10000 [ms]	RWE

Axis Parameter Description (Numbers 1-254)				
Number	**Axis Parameter**	**Description**	**Range [Unit]**	**Access**
249	Init sine mode	0: Initialization in controlled sine commutation (determines the encoder offset) 1: Initialization in block commutation using hall sensors 2: Initialization in controlled sine commutation (use the previous set encoder offset)	0, 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
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 10: Axis Parameter Description (Numbers 1-254)

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

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 11: Functional Access Type Descriptions

Axis Parameter Motor Settings

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

Table 12: Axis Parameter Motor Settings

Axis Parameter Encoder / Initialization 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... +30000 [mA]	RWE
210	Actual hall angle	Actual hall angle value	-32767... +32767	R
211	Actual encoder angle	Actual encoder angle value	-32767... +32767	R

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

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

Torque Regulation Mode				
Number	**Axis Parameter**	**Description**	**Range [Unit]**	**Access**
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

Table 14: 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 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 15: Velocity Regulation Mode

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 16: 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
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 17: 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	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 18: 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 19: Driver Information

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 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

Parameters from 64 upwards configure, for instance:

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

ℹ 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. 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

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 20: 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	UART baud rate	0 9600 baud 1 14400 baud 2 19200 baud 3 28800 baud 4 38400 baud 5 57600 baud 6 76800 baud 7 115200 baud Default	0... 7	RWE
66	Serial address	The module (target) address for RS232 and virtual COM port	0... 255	RWE
73	Configuration EEPROM lock flag	Write: 1234 to lock the EEPROM, 4321 to unlock it. Read: 1=EEPROM locked, 0=EEPROM unlocked.	0/1	RWE
75	Telegram pause time	Pause time before the reply via RS232 is sent. For RS232 set to 0.	0... 255	RWE
76	Serial host address	Host address used in the reply telegrams sent back via RS232.	0... 255	RWE
77	Auto start mode	0: Do not start TMCL application after power-up (default). 1: Start TMCL application automatically after power- up. Note: the current initialization has to be finished first.	0/1	RWE
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

Bank 0 Global Parameters				
Number	**Global Parameter**	**Description**	**Range**	**Access**
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 21: Bank 0 Global Parameters

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

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 22: Bank 2 Access Type Description

Bank 2 Global Parameters

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 23: Bank 2 Global Parameters

5 Motor Regulation

5.1 Structure of Cascaded Motor Regulation Modes

The TMC6130-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 5.1 below:

ℹ Individual modes are explained in the following sections.

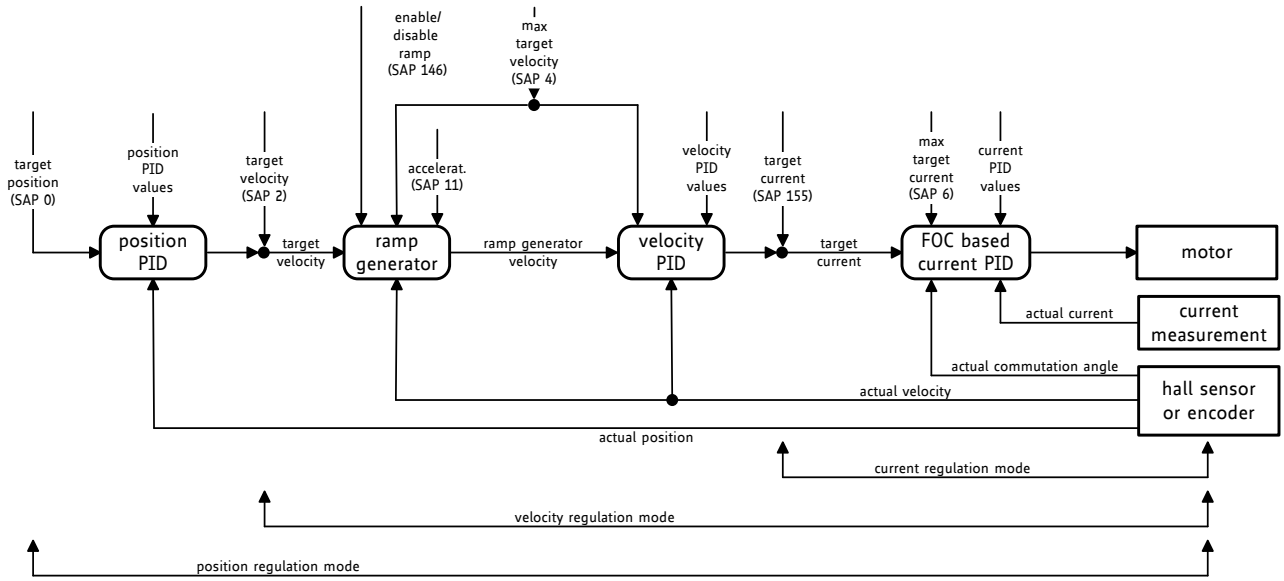

Figure 1: Cascaded Regulation

5.2 Current Regulation

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

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

The PID regulation uses three basic parameters:

- The *P* 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.

Structure of the Current Regulator

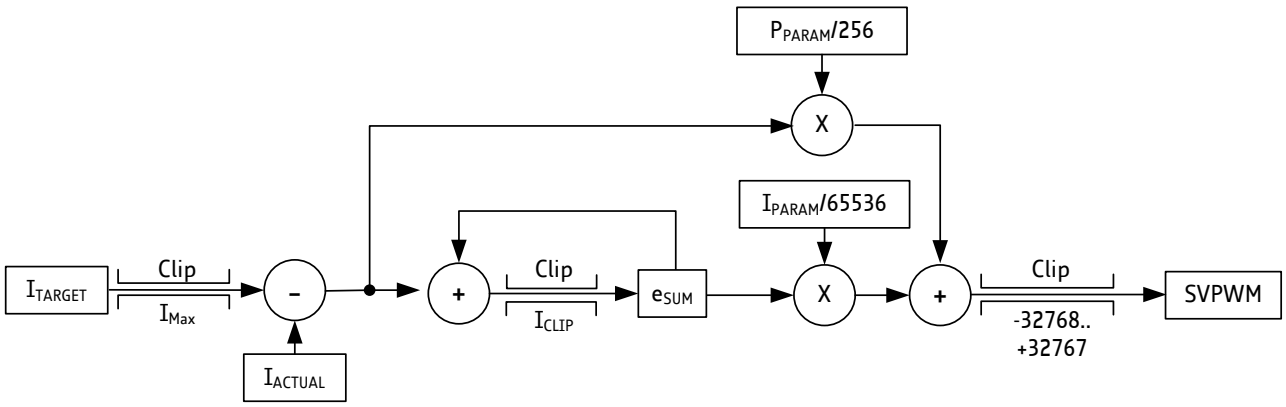

Figure 2: Current Regulation

Parameterizing the Current Regulator Set

- 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.
- 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.*

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 24: Current Regulation Set Parameters

5.3 Velocity Regulation

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

Timing Control Value

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

It is given in multiple of 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.

Structure of the Velocity Regulator

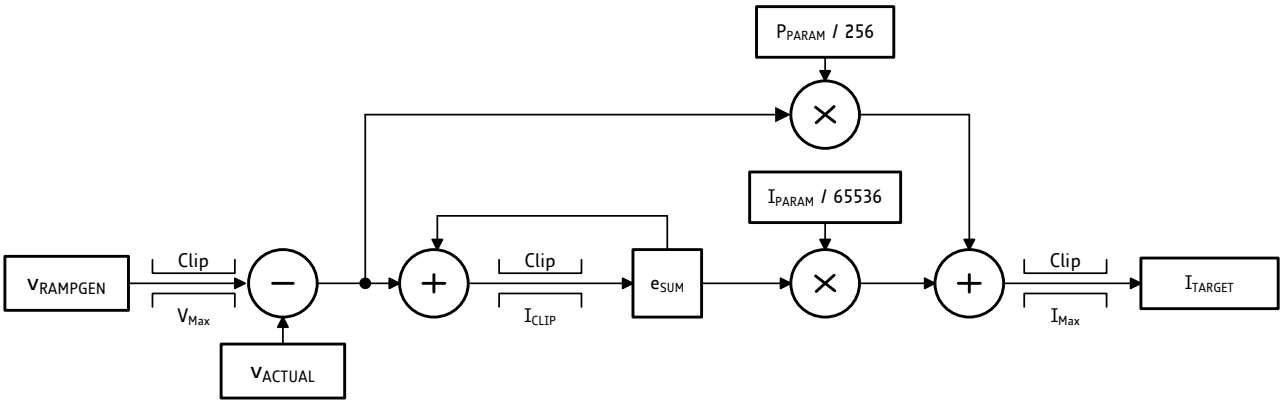

Figure 3: Velocity Regulation

Parameterizing the Velocity Regulator Set

- 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.

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 25: Parameter Description of Velocity Regulator Set

5.4 Velocity Ramp Generator

For a controlled startup of the motor's velocity a velocity ramp generator can be activated or 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.

5.5 Position Regulation

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

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

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.

Structure of the Position Regulator

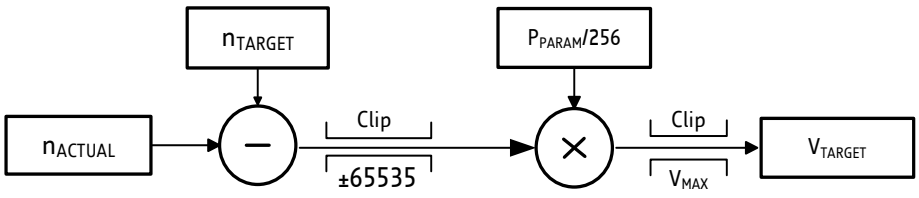

Figure 4: Positioning Regulation

Parameterizing the Position Regulation

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

- 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.
- 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 26: 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

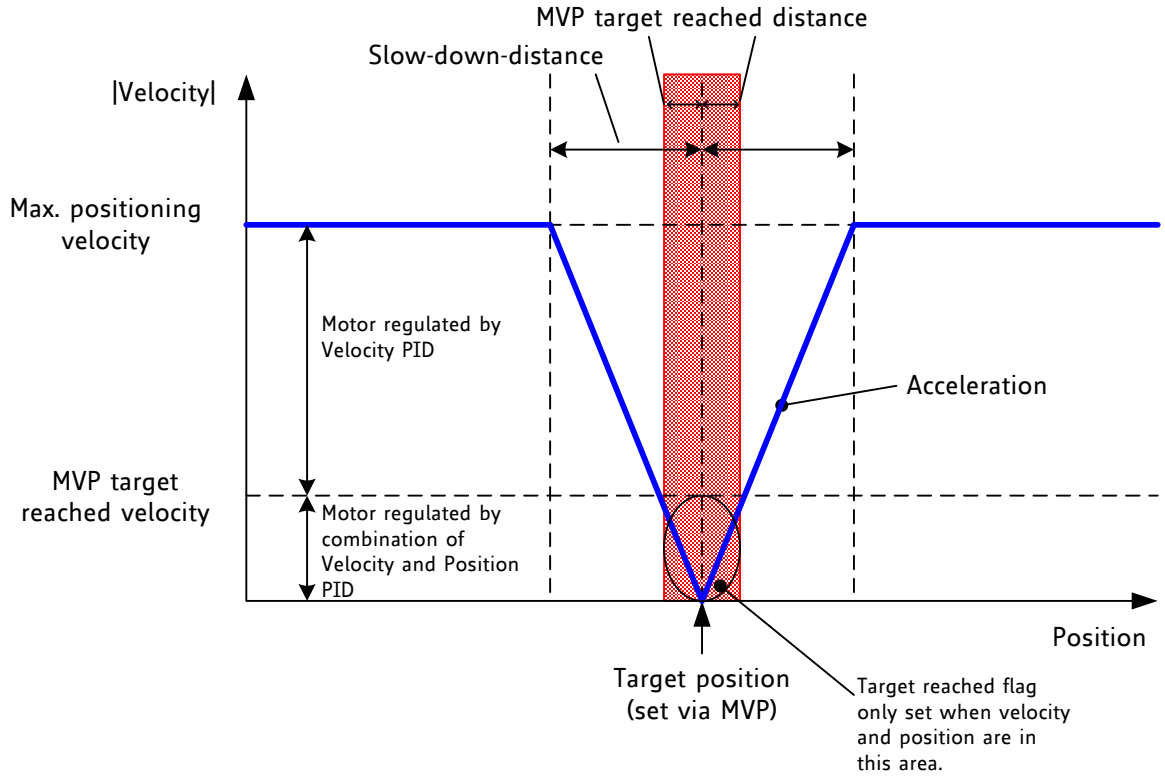

Figure 5: Positioning Algorithm

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!*

USER INFORMATION AND DIRECTIVES

Producer Information

The producer of the product TMC6130 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.

TMC6130 EVALBOARD: Combined Documents

The TMC6130-EVALBOARD User Manual contains two parts:

1. TMC6130-EVALBOARD User Manual: Section Hardware
2. TMC6130-EVALBOARD User Manual: Section Firmware

i This specific publication only contains the firmware documentation for the TMC6130 EVALBOARD.

This document "*EVALBOARD User Manual. Section: Firmware Manual*" is made available to you by TRINAMIC Motion Control GmbH & Co. KG in Hamburg, Germany.

To access other *Datasheets/User Manuals* pertaining to TMC6130, please visit the TMC6130 product page at www.trinamic.com .

Related Documents

Related documents are:

- TMC6130-LA DATASHEET User Manual

i They are made available to you for your convenience but are not part of the *TMC6130 EVALBOARD User Manual* documentation.

Collateral documents

This product documentation comes with additional tool kits, firmware and/or other items, as provided on the specified product page on our company website.

i They are made available to you for your convenience but are not part of the *TMC6130 EVALBOARD User Manual* documentation.

REVISION HISTORY

Firmware Revision History (FW V.)

Version	Date	Description
2.07	2015-MAY-22	Initial version.
2.09	2015-DEC-21	- ADC scaling updated. - Added setter and getter for digital outputs. - Added TMCL-Script reinitialization after power supply switched from off to on during USB-connection.

Table 27: Firmware Revision History

▷ Firmware Document Revision History (DC Rev.)

Version	Date	Author/s	Description
1.00	2015-SEP-04	ED/SV	Initial version.
1.01	2016-JAN-14	ED/SV	Added descriptions for digital outputs.

Table 28: Document Revision History

▷ Publication Date of Firmware Document Revision (PB)

Published online 2015-Sep-04. Available on product webpage at www.trinamic.com.

Hardware Revision History (HW V.)

Version	Date	Description
2.10	2015-DEC-01	Release Version.

Table 29: Hardware Revision History

▷ Hardware Document Revision History (DC Rev.)

NOTE:

→ *Please refer to documentation available on product webpage at www.trinamic.com.*

REFERENCES

Related Documents and Tools

[TMC6130-EVAL]	http://trinamic.com/products/integrated-circuits/evalboards/tmc6130-eval
[TMC6130]	http://trinamic.com/products/integrated-circuits/bldc-driver/tmc6130
[TMCL-IDE]	http://trinamic.com/software-tools/tmcl-ide
[TMCL]	http://www.trinamic.com/design-center/tmcl-code-samples

TABLE INDEX

FIGURE INDEX

IMAGE INDEX

CLOSING NOTE