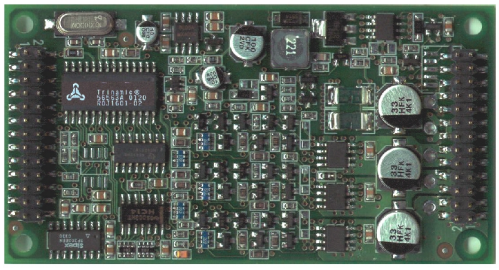
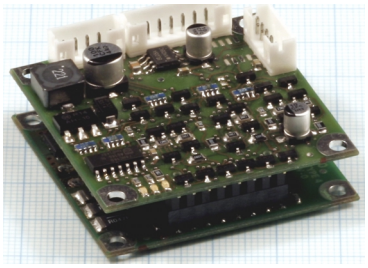
TMCM BLDC MODULE

Reference and Programming Manual

(modules: TMCM-160, TMCM-163)





Version 1.09
August 10th, 2007

TRINAMIC
MOTION CONTROL

Trinamic Motion Control GmbH & Co. KG
Sternstraße 67
D – 20357 Hamburg, Germany
http://www.trinamic.com
INFO@TRINAMIC.COM

Contents

Tables

1 Functional description

1.1 Introduction

The TMCM BLDC modules can either be remote controlled via the PC demonstration software or a user specific program, or they can be controlled by an analog voltage (stand alone mode). The function of the stand alone mode can be modified by the user by writing initialization values to the on-board EEPROM, e.g. a maximum rotation velocity, motor current limit and rotation direction. For a detailed hardware information refer to the specific hardware manuals (5 References).

1.2 Options for stand alone operation

Mode	Functionality	Software settings
PWM control	Motor PWM controlled by the analog input VIN. Motor direction controlled by DIR in pin.	Remote control flag = 0 Power on velocity = 0
PID enforced velocity	Maximum motor velocity v_{max} set via software. This velocity is scaled by VIN pin voltage and enforced by the PID velocity regulator.	Remote control flag = 0 Power on velocity = v_{max}
Constant velocity	Desired motor velocity v set via software	Remote control flag = 1 Power on velocity = v
Current control	With firmware version 1.07 or higher the maximum motor current can also be controlled by an analog signal on input AIN1. The velocity can either also be controlled by an analog signal on AIN0 or set by software.	Remote control flag: set bit 1 (i.e. remote control flag=2 or 3)

In all modes, the motor torque is limit by the maximum current setting. The polarity of the DIR pin can be inversed by the direction input reverse flag setting.

Refer to 3.2.1, Table 3.2, Type 128

1.3 Demonstration Application

You can use the demonstration application for the TMCM-BLDC modules to set the module into operation. Please remark that you should as a first step switch the module to remote controlled mode. You can use the TRINAMIC TMCL IDE to update the modules firmware and to test / set all the parameters of the module.

2 Interface Protocol

The TMCM modules use the TMCL 9 byte protocol. The interface parameters are 9600 baud, 8 bit, 1 stop bit (default values). Please refer to the TMCL manual for a more detailed description of the TMCL basics. However, the BLDC modules define a different function set, detailed in the following.

2.1 Assignment of the command bytes sent from the host to the BLDC-module:

Cmd[0]:	Module address (set to one for the RS232 module)
Cmd[1]:	Command byte / Instruction number
Cmd[2]:	"Type" byte (e.g. parameter number)
Cmd[3]:	"Axis" byte (always set to zero for a one axis module)
Cmd[4], Cmd[5], Cmd[6], Cmd[7]:	optional parameter (CMD[4] is the MSB, Cmd[7] is the LSB) Parameters can have one to four bytes.
Cmd[8]:	Check sum (1 byte sum of the complete datagram)

Example: Rotate right, motor #0, velocity = 350 (refer to 3.1.1 ROR - Rotate Right)

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$01	$00	$00	$00	$00	$01	$5e	$61

2.2 Assignment of the reply bytes sent from the BLDC-module:

Reply[0]:	Host Address
Reply[1]:	Module Address
Reply[2]:	Status
Reply[3]:	Last command byte which was received by the module
Reply[4],Reply[5],Reply[6],Reply[7]:	Value (Reply[4] is the MSB, Reply[7] is the LSB) All reply values are extended to 32 bit
Reply[8]:	Check sum

Example: get the actual position of motor #0 (refer to 3.2.2 GAP - TMCL Get Axis Parameters)

Sent:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor/ Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$06	$01	$00	$00	$00	$00	$00	$08

Reply:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Host-address	Target-address	Status	Instruction	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$02	$01	$64	$06	$00	$00	$02	$c7	$36

3 TMCL Command Set

To enable TMCL control set SAP Type 128 to "1" (refer to 3.2.1).

This chapter describes all TMCL commands. The commands are sorted by their command numbers. For every command the mnemonic with its syntax and the binary representation are given.
Ranges for the parameters are also given.
Last but not least some examples are given at the end of every command description. In the examples for the binary representation always RS232 communication (9 byte) format is used with module address 1 and host address 2. The motor number is always zero for a one axis module.

Please note: All RPM values (v_{rpm8}) are calculated for an 8 pole motor, i.e. 4 electrical field rotations per motor rotation, to make optimum use of the 16 bit velocity control parameters. To get the velocity v in unit [rpm] for a different motor, multiply the value v_{rpm8} by the number of actual poles (n_{poles}) and divide the result by 8. So, one gets

$$v[\mathrm{rpm}] = \frac{v_{rpm8} * n_{poles}}{8}$$

n_{poles}: number of poles of the actually used motor.

Position values (x_{pos}) are in units of 1/6 electrical periods per second

$$x[\mathrm{rotations}] = \frac{x_{pos2} - x_{pos1}}{6 * n_{poles} * 1/2}$$

The TMCM-160 module enfolds a TMCL command set as outlined in Table 1. The commands itself are described afterwards and within the tables Table 2 and Table 4.

TMCL Command	Instruction	Function
ROR	1	Rotate Right
ROL	2	Rotate Left
MST	3	Motor Stop
MVP	4	Move to Position
SAP	5	Set Axis Parameter
GAP	6	Get Axis Parameter
STAP	7	Store Axis Parameter
-	8	-
SGP	9	Set Global Parameter

Table 3.1: TMCL command overview

3.1 TMCL Motion Control Commands

3.1.1 ROR - Rotate Right

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

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

Mnemonic: ROR <motor number>, <velocity>

Value Range: -32767 ... + 32767

Related commands: ROL, MST, SAP, GAP

Binary representation:

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

Example:
Rotate right, motor #0, velocity = 350
Mnemonic: ROR 0, 350
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$01	$00	$00	$00	$00	$01	$5e	$61

3.1.2 ROL - Rotate Left

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

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

Mnemonic: ROL <motor number>, <velocity>

Value Range: -32767 ... + 32767

Related commands: ROR, MST, SAP, GAP

Binary representation:

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

Example:
Rotate left, motor #0, velocity = 1200
Mnemonic: ROL 0, 1200
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$02	$00	$00	$00	$00	$04	$b0	$b7

3.1.3 MST - Motor Stop

Description: This instruction stops the motor.

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

Related commands: ROL, ROR, SAP, GAP

Mnemonic: MST <motor number>

Binary representation:

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

Example: Stop motor #0
Mnemonic: MST 0
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$03	$00	$00	$00	$00	$00	$00	$04

3.1.4 MVP - Move to Position

Description: A movement towards the specified position is started, with automatic generation of acceleration- and deceleration ramps. The maximum velocity and acceleration are defined by axis parameters #4 and #5.
Two operation types are available:
Type 0: Moving to an absolute position in the range from - 8388608 to +8388608 (-2^{23} to$+2^{23}$).
Type 1: Starting a relative movement by means of an offset to the actual position. In this case, the resulting new position value must not exceed the above mentioned limits, too.

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

Related commands: SAP, GAP, MST

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

Binary representation:

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

Examples:
Move motor #0 to (absolute) position 90000
Mnemonic: MVP ABS, 0, 9000
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$04	$00	$00	$00	$01	$5f	$90	$f5

Move motor #0 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	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$04	$01	$00	$ff	$ff	$fc	$18	$18

Note: Before using the MVP command, one has to set MVP specific parameters (please refer command 3.2.1 SAP (set axis parameter), Type 4, 8-10).

3.2 TMCL – Axis Parameter Commands

3.2.1 SAP - TMCL Set Axis Parameter

Description: Most parameters of a TMCM module can be adjusted individually for each axis. Although these parameters vary widely in their formats (1 to 24 bits, signed or unsigned) and physical locations (controller RAM, controller EEPROM), they all can be set by this function. See STAP (section 3.2.3) for permanent storage of a modified value.

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

Related commands: GAP, STAP

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

Binary representation:

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

Example:
set the absolute maximum current of motor #0 to 200mA
Mnemonic: SAP 6, 0, 200
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$05	$06	$00	$00	$00	$00	$c8	$d4

3.2.1.1 List of Set Axis Parameter Commands

Type	Axis Parameter	Value Range	Description
1	Actual position	24 bit signed	Change the position and encoder counter without moving the motor
4	Max. positioning velocity	16 bit unsigned 1…32767	The maximum velocity used for MVP command when executing a ramp to a position.
5	PWM limit	8 bit unsigned 0…255	Set PWM limit, 0…255 ⇔ 0%…100%
6[1]	Max. current	8 bit unsigned 0…250, 255	Max. current setting, refer 3.2.1.2, Table 3.3 Set to 255 and store parameter, to restore factory default values for all stored parameter upon next power on.
7	Active brake speed	16 bit unsigned 0 … 32767	Velocity, below which the motor is braked actively. Below this velocity, the motor is reversed upon direction change command. It is short circuited upon motor stop command or after positioning, if the hard stop flag is set.
8[2]	Min. positioning speed	16 bit unsigned 0 … 32767	Minimum Positioning speed for MVP command: Positioning velocity, when motor position is within the slow-down distance of the target
9[2]	MVP Slow down dist.	16 bit unsigned	Slow-down distance for MVP command
10[2]	MVP Target reached distance	16 bit unsigned	Target reached distance for MVP command: Distance to the target, below which the motor is switched off / braked and the MVP command finishes.
128	Remote control flag	2 bits	Bit 0: Enable external velocity control Bit 1: Enable external current control (see sect. 3.2.1.3)
129	Direction input reverse flag	Boolean	Direction input reverse flag for direction control pin in stand alone mode
130[3]	P – Parameter	16 bit unsigned 1 … 32767	P – Parameter of PID regulator, Velocity deviation is divided by this value and the result is added to the actual PWM on each PID clock
131[3]	I – Parameter	16 bit unsigned 1 … 32767	I-Parameter: Integrated velocity deviation is divided by this value and added to PWM on each PID clock
132[3]	D – Parameter	16 bit unsigned 1 … 32767	D-Parameter: Change of velocity deviation between two calculations is divided by this value and added to PWM on each PID clock
133[3]	PID regulation loop delay	8 bit unsigned 0…255	PID calculation delay (0-255 * 10ms): Between each two PID calculation at least this delay is used
134	Current regulation loop delay	8 bit unsigned 0…254	Delay of current limitation algorithm
148	Disable stop switch	Boolean	Set (1) to disable stop switch function. This parameter can not be stored for security reasons.
149	Hard stop flag	Boolean	Hard stop flag: Set (1) to break motor hard, if velocity is below or equal active break velocity
157	Power on velocity	16 bit signed	Power on velocity: Velocity after power on, if RS232 is enabled. Maximum velocity in potentiometer mode if value is different from 0. Otherwise potentiometer mode uses direct PWM control.

Table 3.2: TMCL Commands, Set Axis Parameter (SAP)

[1] refer Table 3 and to the specific Hardware manuals (5 References)
[2] refer to the specific Hardware manuals (5 References)
[3] refer to the specific Hardware manuals (5 References)

3.2.1.2 Maximum Current Regulation

$$I_{Limit}[A] = \frac{A_{Limit}}{250} \cdot A_{max}[A]$$

	I_{Limit}	A_{Limit}	A_{max}	$A_{Resolution}$
TMCM-160	0...5A (9,0A)	0...138 (250)	9,25A	36mA
TMCM-163	0...10A (13,5A)	0...184 (250)	13,8A	54mA

Table 3.3: Max. Current Regulation with different modules

3.2.1.3 Remote Control Flag

The remote control flag is interpreted bitwise. Bit 0 controls the analog control mode for the velocity, and bit 1 controls the analog control mode for the maximum motor current.

Bit 0: Clearing this bit turns on the analog control mode for the velocity. The velocity can then be controlled by the potentiometer (on TMCM-160 evaluation kits) or the analog input pin AIN0. The speed range can be defined using SAP 157.
Setting this bit disables the analog control mode for the velocity. The velocity can then only be controlled by TMCL commands.

Bit 1 (Firmware V1.07 or higher): Setting this bit enables the analog control mode for the maximum motor current. The maximum motor current can then be controlled by the analog input pin AIN1. The motor current range can be defined using SAP 6. The motor current is near zero when there is 0V at AIN1, and it is at the value set by SAP 6 when AIN1 is at its maximum input voltage.
Clearing this bit disables this mode.

This means that there are four possible values for the remote control flag:

- 0: Velocity controlled by input AIN0, current controlled by TMCL commands only.
- 1: Velocity and current controlled by TMCL commands only.
- 2: Velociyt controlled by input AIN0, current controlled by input AIN1 (firmware V1.07 or higher).
- 3: Velocity controlled by TMCL commands only, current controlled by input AIN1 (firmware V1.07 or higher).

Changing this value also clears the stop switch flag and the position reached flag.

The following example shows how to change the remote control mode. The address is 1 and the Motor/Bank number is 0.

Address	Instruction	Type	Motor/Bank	Value	*Function*
1	5	128	0	0	sets analog velocity control mode
1	5	128	0	1	sets TMCL velocity control mode

Table 3.4: Example how to change the remote control mode

3.2.2 GAP - TMCL Get Axis Parameters

Description: Most parameters of a TMCM module can be adjusted individually for each axis. Although these parameters vary widely in their formats (1 to 24 bits, signed or unsigned) and physical locations (controller RAM, controller EEPROM), they all can be read by this function.
All parameters are extended to 32 bit value for the ease of use.

Related commands: SAP, STAP

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

Binary representation:

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

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

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$06	$01	$00	$00	$00	$00	$00	$08

Reply:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Host-address	Target-address	Status	Instruction	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$02	$01	$64	$06	$00	$00	$02	$C7	$36

3.2.2.1 List of Get Axis Parameter Commands

Type	Axis Parameter	Value Range	Description
0	Target position	24 bit signed	The target position of a currently executed ramp.
1	Actual position	24 bit signed	The actual position of the motor.
2	Target speed	16 bit signed	Programmed target velocity
3	Actual speed	16 bit signed	The actual velocity of the motor.
4	Max. positioning velocity	16 bit unsigned	See associated SAP command
5	Duty cycle limit	8 bit unsigned 0…255	Limit duty cycle to a defined value
6	Max. current	8 bit unsigned 0…250	See associated SAP command, refer 3.2.1.2
7	Active brake speed	16 bit unsigned 0…32767	Active brake velocity
8	Min. positioning speed	16 bit unsigned	See associated SAP command
9	MVP slow down distance	16 bit unsigned	See associated SAP command
10	MVP target reached distance	16 bit unsigned	See associated SAP command
128	Remote control flag	2 bits	See associated SAP command
129	Direction input reverse flag	Boolean	See associated SAP command
130	P-Parameter	16 bit unsigned	See associated SAP command
131	I-Parameter	16 bit unsigned	See associated SAP command
132	D-Parameter	16 bit unsigned	See associated SAP command
133	PID regulation loop delay	8 bit unsigned	See associated SAP command
134	Current regulation loop delay	8 bit unsigned	See associated SAP command
148	Disable stop switch	Boolean	See associated SAP command
149	Hard stop flag	Boolean	See associated SAP command
150	Actual motor current	8 bit unsigned	actual motor current (value limited to range of about 2 and 253), same calculation formula as for max. current regulation, refer 3.2.1.2
151	Actual voltage	16 bit unsigned	Actual Voltage (in 100mV steps)
152	Actual Temperature	Value	actual temperature, uses calculation formula for T[°C] = Round(0.590024… * Power(value, 0.997586…))
153	Actual PWM duty cycle	8 bit unsigned	Actual PWM duty cycle
154	ADC readout	8 bit unsigned	Potentiometer readout (analog input AIN0)
155	Stop flag	Boolean	Stop flag status: 1= Motor stopped due to stop switch
156	Error status	16 bit	Error status readout, refer 3.2.2.2, Table 3.6
157	Power on velocity	16 bit signed	See associated SAP command
158	Actual current limit	16 bit unsigned 0…256	Actual motor current limitation factor: 256: currently no limit
159	AIN1 readout	8 bit unsigned	Readout of external analog input AIN1

Table 3.5: TMCL Commands, Get Axis Parameter (GAP)

3.2.2.2 Error Status Readout

bit	Description	
0	UNDERVOLT	Set if supply voltage too low for motor operation
1	OVERTEMP	Set if overtemperature is limit is exceeded
2	TEMPWARN	Set if temperature warning level is reached
3	STOPSWITCH	Set if motor stopped due to stop switch or stop command
4	HALTED	Set if motor is actively braked
5	HALLERR	Set upon a hall sensor error (also upon power on of the module) Reset upon read of error status
6	POSMOD	This flag is set when the module is in positioning mode
7	POSEND	This flag becomes set if the motor has stopped at the end position

Table 3.6: Error Status

3.2.3 STAP – Store Axis Parameter

Description: Axis parameters are located in RAM memory, so modifications are lost at power down. This instruction enables permanent storing of the previously with SAP altered parameter. All parameters alterable with SAP – Set Axis Parameter can be stored, except Type 1 – Set Actual Position (refer to 3.2.1).

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

Related commands: SAP, GAP

Mnemonic: STAP <parameter number>

Binary representation:

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

Example: store the maximum speed of motor #0
Mnemonic: STAP 4, 0
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$07	$04	$00	$00	$00	$00	$00	$0C

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

3.3 Global Parameters

3.3.1 SGP - Set Global Parameter

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

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

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

Binary representation:

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

Example: Set the actual temperature to 25°C
Mnemonic: SGP 255, 0, 25
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$09	$ff	$00	$00	$00	$00	$19	$22

3.3.1.1 List of Global Parameter Commands

Type	Axis Parameter	Value Range	Description
65	RS232 and RS485 baud rate	Byte	Change of RS232 and RS485 Baud Rate, refer 3.3.1.2, Table 3.8
66	Serial address	8 bit unsigned, 0…255	Set address of module (default=1)
75	Telegram pause time	Byte	Telegram pause time for RS485 / RS232 interface (default=0). Sets delay after telegram receipt, until data becomes sent out (time in ms, 0=immediately). Important for RS485 interface, set to zero for RS232. An RS232 to RS485 converter used with Windows requires a setting of about 20.
76	Serial host address	Byte	Set host address to reply to host (default=2)
254	Hall sensor invert	Boolean	Hall sensor invert (1=invert) – Sets one of the motors with inverted hall scheme, e.g. Maxon
255	Actual Temperature	Byte	Actual temperature for temperature correction, value range 20°C…40°C *Calibrated at factory - It might be necessary to recalibrate after a firmware update*

Table 3.7: TMCL Commands, Set Global Parameters (SGP)

3.3.1.2 Change of RS232 and RS485 Baud Rate

New setting becomes active after next power on.

byte value	Baud Rate
0	9600 baud (default)
1	14400 baud
2	19200 baud
3	28800 baud
4	38400 baud
5	57600 baud
6	76800 baud
7	115200 baud
8	250000 baud

Table 3.8: Baud Rate

3.4 Get Version Number

Description: Provides number of the actual firmware version installed on the module.

Related commands: GAP, SAP

Mnemonic: GAP <parameter number>

Binary representation:

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

Example: request Version Number (i.e. 160V1.03)
Mnemonic: GAP 136
Binary:

Byte Index	0	1	2	3	4	5	6	7	8
Function	Target-address	Instruction Number	Type	Motor / Bank	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0	Checksum
Value (hex)	$01	$88	$00	$00	$00	$00	$00	$00	$89

Reply: "160V1.03", Note: this reply is not in the standard TMCL syntax

Byte Index	0	1	2	3	4	5	6	7	8
Function	Host-address	Version Number							
		Operand Byte7	Operand Byte6	Operand Byte5	Operand Byte4	Operand Byte3	Operand Byte2	Operand Byte1	Operand Byte0
Value (hex)	$02	$31	$36	$30	$56	$31	$2e	$30	$33

4 Documentation Revision

Version	Comment	Author	Description
1.07	New Release	HC	Revised initial version
1.08	Revised	HC	Some format changes
1.09	Updated	OK	New features of firmware version 1.07 added

Table 4.1: Documentation Revision

5 References

[TMCL] TMCL Manual, www.trinamic.com
TMCM 160 - Hardware Hardware Description of TMCM 160, www.trinamic.com
TMCM 163 - Hardware Hardware Description of TMCM 163, www.trinamic.com