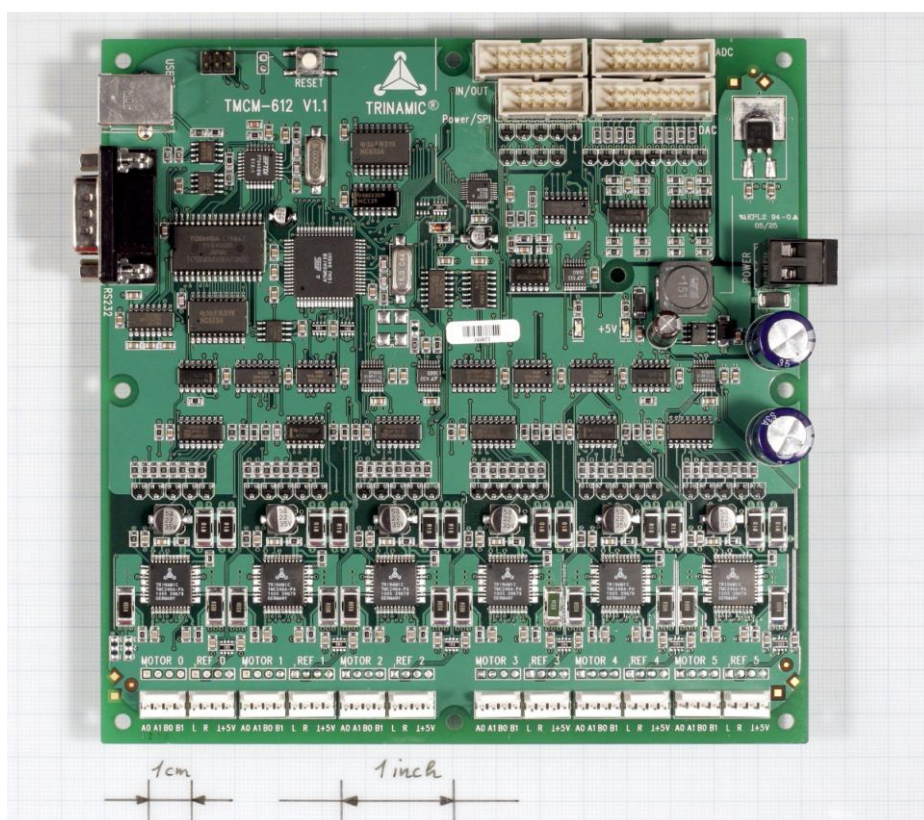


TMCM-612

6-Axis Controller / High-Resolution Driver Board 1.1A /34 V + Data Acquisition



Manual

Version: 1.13
March 29th, 2012



TRINAMIC

MOTION CONTROL

Trinamic Motion Control GmbH & Co KG

Sternstraße 67

D - 20357 Hamburg, Germany

Phone +49-40-51 48 06 - 0

FAX: +49-40-51 48 06 - 60

<http://www.trinamic.com>

Contents

1	Introduction	4
2	Life support policy	5
3	Electrical and Mechanical Interfacing	6
3.1	Dimensions	6
3.2	Connecting the TMC612 module	6
3.2.1	Connectors used on the TMC612 module	7
3.2.2	Power supply	7
3.2.3	LED indicators	7
3.2.4	Motor connectors	7
3.2.5	Stop switches / Reference switches	7
3.2.6	RS232 interface	8
3.2.7	USB interface	8
3.2.8	General purpose I/O	8
3.2.9	Reset button	8
3.2.10	ISP connector – restore to factory default	9
3.2.11	ADC connector	9
3.2.12	DAC connector	10
3.2.13	Expansion Connector	10
4	Operational Ratings	11
4.1	Main technical data	11
5	Functional Description	12
5.1	System Architecture	12
5.1.1	Microcontroller	12
5.1.2	TMCL EEPROM	12
5.1.3	TMC428 Motion Controller	12
5.1.4	Stepper Motor Drivers	13
5.1.5	ADC / DAC converter	13
5.2	StallGuard™ - Sensorless Motor Stall Detection	13
5.2.1	StallGuard adjusting tool	14
5.2.2	StallGuard profiler	14
5.3	Reference switches	15
5.3.1	Left and right limit switches	15
5.3.2	Triple Switch Configuration	15
5.3.3	One Limit Switch for circular systems	16
5.4	USB	16
6	Putting the TMC612 into Operation	17
7	TMC612 Operational Description	18
7.1	Calculation: Velocity and Acceleration vs. Microstep- and Fullstep-Frequency	18
8	TMCL	19
8.1	Differences in TMCL commands	19
8.1.1	MVP COORD	19
8.1.2	WAIT RFS	20
8.2	Additional commands	20
8.2.1	Read the ADC: Uf0	20
8.2.2	Write to the DAC: Uf1	20
8.2.3	Set the polarity of the stop switches: Uf2	20
8.2.4	Read from the additional data RAM: Uf3	20
8.2.5	Write to the additional data RAM: Uf4	21
9	Revision History	22
9.1	Documentation Revision	22
9.2	Firmware Revision	22

List of Figures

Figure 3.1: Dimensions	6
Figure 3.2: Overview of the connectors	7
Figure 3.3: Motor and reference switch connection	7
Figure 3.4: Restoring all parameters to factory default	9
Figure 5.1: Main parts of the TMC612	12
Figure 5.2: StallGuard adjusting tool.....	14
Figure 5.3: The StallGuard Profiler	14
Figure 5.4: Left and right limit switches	15
Figure 5.5: Limit switch and reference switch	16
Figure 5.6: One reference switch	16

List of Tables

Table 1.1: Order codes	4
Table 3.1: General purpose I/Os	8
Table 3.2: ADC connector	9
Table 3.3: DAC connector	10
Table 3.4: Expansion connector.....	10
Table 4.1: Operational Ratings.....	11
Table 5.1: StallGuard parameter SAP 205	13
Table 5.2: Pinout reference switches.....	15
Table 7.1: TMC428 Velocity parameters.....	18
Table 9.1: Documentation Revisions.....	22
Table 9.2: Firmware Revisions.....	22

1 Introduction

The TMC612 is a six axis 2-phase stepper motor controller and driver module with a high performance data acquisition part. The integrated 8 channel 16 bit ADC converter can be programmed to do a step-synchronous input voltage scan and store values at a high data rate. The module provides a high microstep resolution in order to do very exact positioning and measurement tasks. The measurement results can be transferred to a PC using the high-speed USB interface. A number of analog output channels and digital I/Os can be used to control further instrumentation.

This feature set makes the module pre-destined for analytical instruments.

The TMC612 comes with the PC-based software development environment TMCL-IDE for the Trinamic Motion Control Language (TMCL). User specific data acquisition extensions are available upon request. The TMC612 can be controlled via the high-speed USB interface or via its RS-232 interface.

Applications

- Controller / driver board for control of up to 6 Axis with very high precision
- Versatile possibilities of applications in stand alone or pc controlled mode

Motor type

- Coil current from 300mA to 1.1A RMS (1.5A peak)
- 12V to 34V nominal supply voltage

Interface

- RS232 or USB host interface
- Inputs for reference and stop switches
- General purpose analog and digital I/Os
- Eight 16 bit ADC inputs (0 – 10V)
- Eight 10 bit DAC outputs (0 – 10V)

Highlights

- Up to 64 times microstepping
- 500kHz, 16 bit AD converter
- 128kbyte RAM for data acquisition
- Automatic ramp generation in hardware
- StallGuard™ option for sensorless motor stall detection
- Full step frequencies up to 20kHz
- On the fly alteration of motion parameters (e.g. position, velocity, acceleration)
- Local reference move using sensorless StallGuard™ feature or reference switch
- Dynamic current control
- TRINAMIC driver technology: No heatsink required
- Many adjustment possibilities make this module the solution for a great field of demands

Software

- Stand-alone operation using TMCL or remote controlled operation
- TMCL program storage: 16 KByte EEPROM (2048 TMCL commands)
- PC-based application development software TMCL-IDE included

Other

- Pluggable connectors for motor and reference switches
- RoHS compliant latest from 1 July 2006
- Size: 160x160mm²

Order code	Description
TMC612/SG	6.axis controller/driver and data acquisition module, StallGuard

Table 1.1: Order codes

2 Life support policy

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

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

© TRINAMIC Motion Control GmbH & Co. KG 2008

Information given in this data sheet 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.

3 Electrical and Mechanical Interfacing

3.1 Dimensions

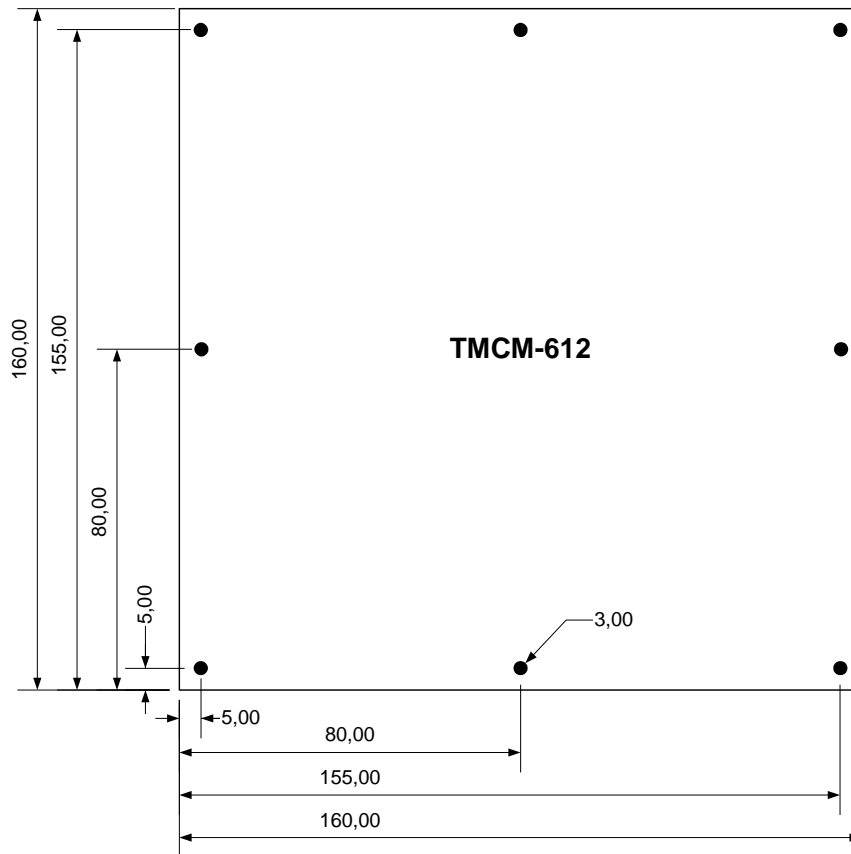


Figure 3.1: Dimensions

3.2 Connecting the TMCM-612 module

Figure 3.2 gives an overview of all the connectors. The following sections describe all the connectors in detail.

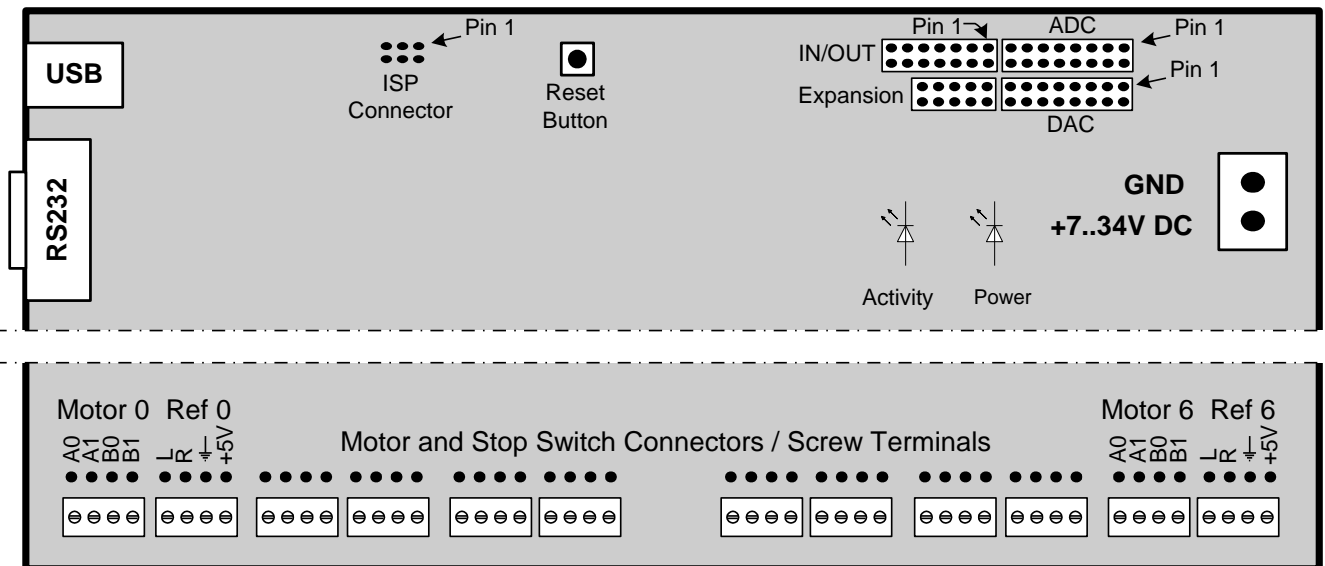


Figure 3.2: Overview of the connectors

3.2.1 Connectors used on the TCM-612 module

All connectors used on the TCM-612 module are industry standard connectors except for motor and stop switches. So mating connectors can be obtained from many different manufacturers.

- Motor and stop switches: 1x4 pin, 2.54mm pitch, AMP 640456-4 connector
- ADC and DAC connectors: industry standard header, 2x8 pins, 2.54mm pitch.
- I/O: industry standard header, 2x7pins, 2.54mm pitch.
- Expansion (Power/SPI): industry standard header, 2x5 pins, 2.54mm pitch.

3.2.2 Power supply

Connect a power supply of max. 34V DC here (the minimum operating voltage is 12V). The device is protected against wrong polarity by a diode that shorts the power supply when the polarity is wrong.

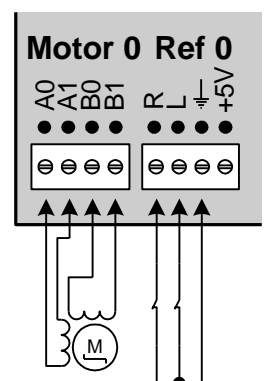
3.2.3 LED indicators

There are two LEDs on the board. The right LED ("Power", marked +5V) lights up when the unit is powered. The other LED ("Activity") flashes when the unit is running normally.

3.2.4 Motor connectors

The stepper motors can be connected with 4 pin 2.54mm pitch connectors. The soldering points behind the connectors are electrically identical. The pin assignments of the connectors are printed on the board. Connect one coil of the motor to the terminals marked "Ao" and "A1" and the other coil to the connectors marked "Bo" and "B1". See Figure 3.2. **Warning: Never connect or disconnect a motor while the unit is powered! This can damage the motor drivers and maybe also other parts of the unit!**

Figure 3.3: Motor and reference switch connection



3.2.5 Stop switches / Reference switches

The stop switches can be connected to the terminals marked "L" and "R" and to the GND terminal. The switches are "normally closed".

The reference switch connectors also have a "+5V" terminal. This is a 5V output that can be used to supply photo couplers or digital hall sensors.

The left stop switch is also used as the reference switch.

3.2.6 RS232 interface

The RS232 interface (default 9600 bps, max. 115200 bps) is one way to connect the unit to a PC or a microcontroller with RS232 interface. All TMCL commands can be sent to the unit via this interface. A null modem cable has to be used to connect the TMC612 to a PC, so the following connections have to be made:

TMC612 pin	PC pin
2	3
3	2
5	5

The pin assignments of the RS232 socket of the TMC612 are as follows:

Pin number	Signal name
2	RxD
3	TxD
5	GND

All the other pins of this connector are not connected.

3.2.7 USB interface

The USB interface is also a way to connect the unit to a PC, when higher communication speed is needed. The interface supports the USB 2.0 standard. Please see chapter 5.4 on how to install the device driver that is needed to communicate with the TMC612 via USB.

The USB interface and the RS232 interface should not be used simultaneously.

3.2.8 General purpose I/O

The general purpose I/O connector provides eight digital input/output lines. Each of these lines can be programmed to be used either as a digital output or as a digital input or as an analogue input with 10 bits accuracy and a maximum input voltage of +5V. All digital inputs and outputs operate at TTL level, so the maximum voltage is 5V. The maximum current when used as a digital output is 20mA. The pin assignments of the connector are as follows:

Pin	Signal	Pin	Signal
1	Alarm Input	2	GND
3	I/O 0	4	I/O 1
5	I/O 2	6	I/O 3
7	I/O 4	8	I/O 5
9	I/O 6	10	I/O 7
11	+5V	12	GND
13	+5V	14	GND

Table 3.1: General purpose I/Os

The alarm input also is a digital input with TTL level and an internal pull-up resistor. The functionality of this input can be configured to stop all motors when it is high or to stop all motors when it is low or to no function at all (please see the software section for details).

Pin 1 of the connector is shown in Figure 3.2 and is also marked with an arrow on the board. The pins with odd numbers are those near to the edge of the board.

3.2.9 Reset button

Pressing the reset button resets the microcontroller. All motors are then stopped immediately and everything is re-initialized.

3.2.10 ISP connector – restore to factory default

This connector is used for two purposes:

- Programming the CPU via an in-circuit programmer: **This is to be done by Trinamic only and not by the user!** (The user can upgrade the firmware via the RS232 or USB interface using the "Install OS" function in the TMCL IDE.)
- **Restoring all parameters to their factory default values:** Nearly all parameters can be stored in the EEPROM of the CPU. If some parameters have been set wrongly this can lead to a case of miss-configuration where the module can not be reached by a PC any more. In such circumstances, all parameters can be reset to their factory default values by doing the following:
 1. Turn off the power.
 2. Link the pins 1 and 3 of the ISP connector with a jumper (as shown in Figure 3.4).
 3. Turn on the power and wait until the "Activity" LED flashes fast (much faster than normal).
 4. Turn off the power.
 5. Remove the link between the pins 1 and 3 of the ISP connector.
 6. Turn on the power and wait until the LED flashes normally (this can take some seconds).
 Now, all parameters are restored to their factory default values, and the unit should work normally again.

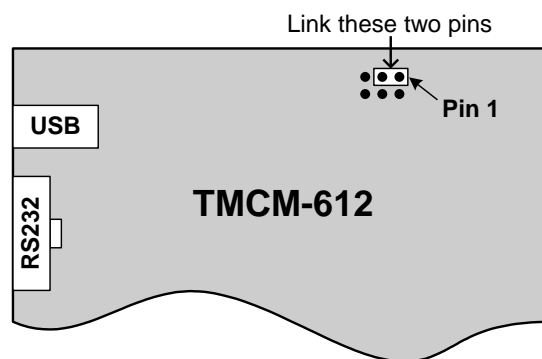


Figure 3.4: Restoring all parameters to factory default

3.2.11 ADC connector

The ADC connector is marked with "ADC" on the board and provides eight analogue inputs with 16 bit accuracy and an input voltage range of 0..+10V. The pin assignments of this connector are as follows:

Pin	Signal	Pin	Signal
1	ADC input 0	2	GND
3	ADC input 1	4	GND
5	ADC input 2	6	GND
7	ADC input 3	8	GND
9	ADC input 4	10	GND
11	ADC input 5	12	GND
13	ADC input 6	14	GND
15	ADC input 7	16	GND

Table 3.2: ADC connector

Pin 1 is marked with an arrow on the board and is also shown in Figure 3.2. All pins with odd numbers are those near to the edge of the board.

3.2.12 DAC connector

The DAC connector is marked with "DAC" on the board and provides eight analogue outputs with 10 bit accuracy and an output voltage range of 0..+10V. The pin assignments of the DAC connector are as follows:

Pin	Signal	Pin	Signal
1	DAC output 0	2	GND
3	DAC output 1	4	GND
5	DAC output 2	6	GND
7	DAC output 3	8	GND
9	DAC output 4	10	GND
11	DAC output 5	12	GND
13	DAC output 6	14	GND
15	DAC output 7	16	GND

Table 3.3: DAC connector

Pin 1 is marked with an arrow on the board and is also shown in Figure 3.2. All pins with odd numbers are those near to the edge of the board.

3.2.13 Expansion Connector

The expansion connector is marked with "Power/SPI" on the board. Here, additional peripheral device can be attached to the CPU via SPI or UART interface. Also, the analogue voltages (+5V and +15V) are provided here. The pin assignments of this connector are as follows:

Pin	Signal	Pin	Signal
1	+15V (analogue)	2	DAC ref. 3.1V
3	+5V (analogue)	4	+5V (digital)
5	UART RxD (TTL level)	6	UART TxD (TTL level)
7	SPI_CS	8	SPI_MISO
9	SPI_SCK	10	SPI_MOSI

Table 3.4: Expansion connector

Pin 1 is marked with an arrow on the board and is also shown in Figure 3.2. All pins with odd numbers are those near to the edge of the board.

4 Operational Ratings

Symbol	Parameter	Min	Typ	Max	Unit
V_S	DC Power supply voltage for operation	12	15 ... 28	34	V
I_{COIL}	Motor coil current for sine wave peak (chopper regulated, adjustable via software)	0	0.3 ... 1.5	1.5	A
f_{CHOP}	Motor chopper frequency		36.8		kHz
I_S	Power supply current (per motor)		$\ll I_{COIL}$	$1.4 * I_{COIL}$	A
V_{INPROT}	Input voltage for StopL, StopR, GPIO (internal protection diodes)	-0.5	0 ... 5	$V_{+5V} + 0.5$	V
V_{ANA}	INx analog measurement range of I/Os		0 ... 5		V
V_{ADC}	Analog measurement range		0 ... 10		V
V_{DAC}	Analog output range		0 ... 10		V
V_{INLO}	INx, StopL, StopR low level input		0	0.9	V
V_{INHI}	INx, StopL, StopR high level input (integrated 10k pullup to +5V for Stop)	2	5		V
I_{OUTI}	OUTx max +/- output current (CMOS output) (sum for all outputs max. 50mA)			+/-20	mA
T_{ENV}	Environment temperature at rated current (no cooling)	-40		+70	°C

Table 4.1: Operational Ratings

4.1 Main technical data

- Supply voltage: DC, 12..34V
- Motor type: bipolar, two-phase stepper motor
- Maximum peak coil current: 1.5A (adjustable by software in 255 steps)
- Interfaces:
 - RS232 (default 9600 bps, max. 115200 bps)
 - USB 2.0
- eight general purpose inputs/outputs (as output: 5V, max. 20mA, or as input: TTL level digital or analog max. 5V, 10 bit)
- eight analogue inputs with 16 bit accuracy and input voltage range 0..+10V
- eight analogue outputs with 10 bit accuracy and output voltage range 0..+10V
- one alarm input (TTL level)
- two stop switch inputs for every motor (TTL level), polarity selectable for every motor
- CPU: ATmega128
- Clock frequency: 16MHz
- Stepper motor controller: two TMC428
- Stepper motor driver: six TMC246 (with StallGuard) or six TMC236 (without StallGuard), extended for 64 micro-steps
- EEPROM for TMCL program storage: 16kBytes (suitable for up to 2048 TMCL commands)
- Additional 128kB RAM for data acquisition
- Firmware upgrades possible through RS232 or USB interface
- Operating temperature range: -40..70°C

5 Functional Description

In Figure 5.1 the main parts of the TCM-612 module are shown. The module mainly consists of two TMC428 motion controller, six TMC246 stepper motor driver, the TMCL program memory (EEPROM) and the host interfaces (RS-232 and USB). Special are the ADC and DAC converters and the extra data RAM of 128kbyte.

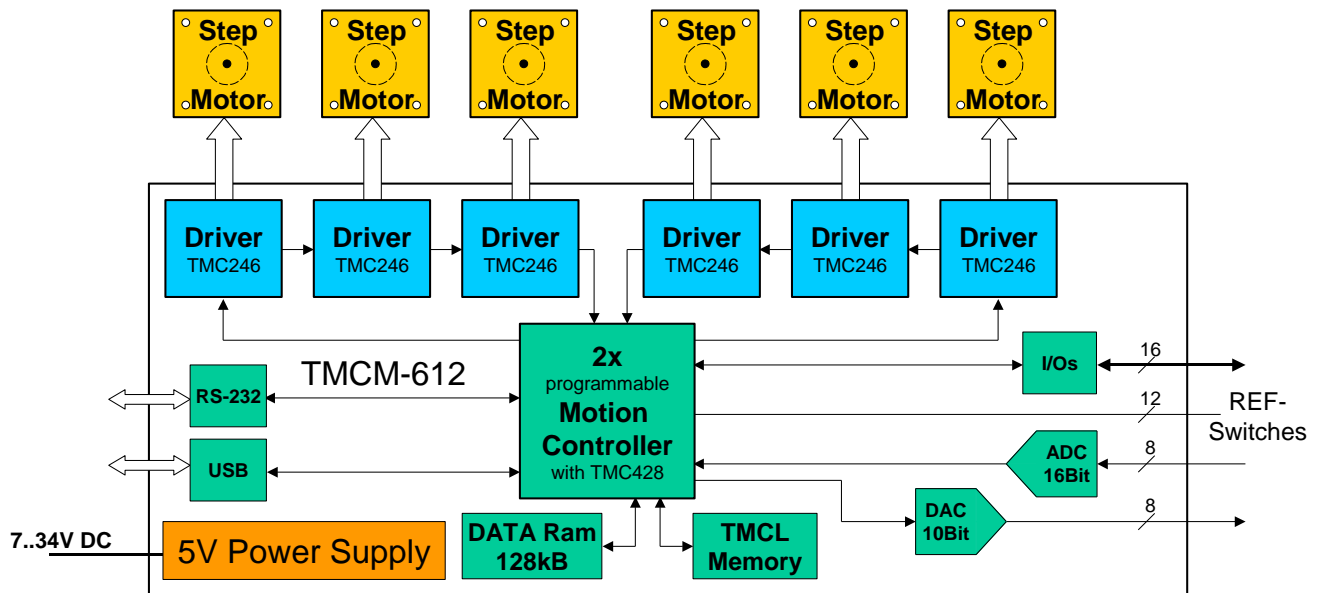


Figure 5.1: Main parts of the TCM-612

5.1 System Architecture

The TCM-612 integrates a microcontroller with the TMCL (Trinamic Motion Control Language) operating system. The motion control real-time tasks are realized by the TMC428.

5.1.1 Microcontroller

On this module, the Atmel Atmega128 is used to run the TMCL operating system and to control the TMC428. The CPU has 128Kbyte flash memory and a 2Kbyte EEPROM. The microcontroller runs the TMCL (Trinamic Motion Control Language) operating system which makes it possible to execute TMCL commands that are sent to the module from the host via the RS232 and USB interface. The microcontroller interprets the TMCL commands and controls the TMC428 who execute the motion commands.

The flash ROM of the microcontroller holds the TMCL operating system and the EEPROM memory of the microcontroller is used to permanently store configuration data.

The TMCL operating system can be updated via the RS232 interface. Use the TMCL IDE to do this.

5.1.2 TMCL EEPROM

To store TMCL programs for stand alone operation the TCM-612 module is equipped with a 16kByte EEPROM attached to the microcontroller. The EEPROM can store TMCL programs consisting of up to 2048 TMCL commands.

5.1.3 TMC428 Motion Controller

The TMC428 is a high-performance stepper motor control IC and can control up to three 2-phase-stepper-motors. Motion parameters like speed or acceleration are sent to the TMC428 via SPI by the microcontroller. Calculation of

ramps and speed profiles are done internally by hardware based on the target motion parameters. The TMC612 has two TMC428 for 6 axis.

5.1.4 Stepper Motor Drivers

On TMC612 modules the TMC246 driver chips are used. These chips are fully compatible with the TMC236 chips, but have the additional StallGuard feature. These drivers are very easy to use. They can control the currents for the two phases of the stepper motors. 16x microstepping and maximum output current of 1500mA are supported by these driver ICs.

As the power dissipation of the TMC236 and TMC246 chips is very low no heat sink or cooling fan is needed. The temperature of the chips does not get high. The coils will be switched off automatically when the temperature or the current exceeds the limits and automatically switched on again when the values are within the limits again.

5.1.5 ADC / DAC converter

The ADC converter can be programmed to do a step synchronous input voltage scan and store the values at a high data rate. This data can be stored in the extra 128 kbytes data RAM.

5.2 StallGuard™ - Sensorless Motor Stall Detection

The TMC612/SG modules are equipped with the StallGuard option. The StallGuard option makes it possible to detect if the mechanical load on a stepper motor is too high or if the traveler has been obstructed. The load value can be read using a TMCL command or the module can be programmed so that the motor will be stopped automatically when it has been obstructed or the load has been too high.

StallGuard can also be used for finding the reference position without the need for a reference switch: Just activate StallGuard and then let the traveler run against a mechanical obstacle that is placed at the end of the way. When the motor has stopped it is definitely at the end of its way, and this point can be used as the reference position.

To use StallGuard in an actual application, some manual tests should be done first, because the StallGuard level depends upon the motor velocities and on the occurrence of resonances. When switching on StallGuard, the motor operation mode is changed and microstep resolution may be worse. Thus, StallGuard should be switched off when not in use.

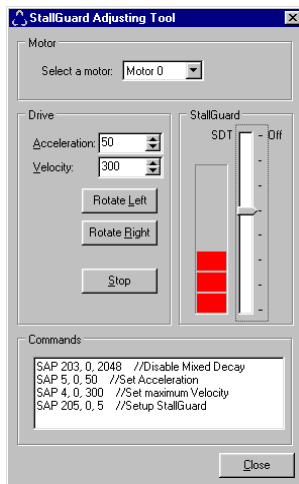
Mixed decay should be switched off when StallGuard operational in order to get usable results.

Value	Description
-7..-1	Motor stops when StallGuard value is reached and position is set zero (useful for reference run).
0	StallGuard function is deactivated (default)
1..7	Motor stops when StallGuard value is reached and position is not set zero.

Table 5.1: StallGuard parameter SAP 205

To activate the StallGuard feature use the TMCL-command SAP 205 and set the StallGuard threshold value according to Table 5.1. The actual load value is given by GAP 206. The TMCL IDE has some tools which let you try out and adjust the StallGuard function in an easy way. They can be found at "StallGuard" in the "Setup"-menu and are described in the following chapters.

5.2.1 StallGuard adjusting tool



The StallGuard adjusting tool helps to find the necessary motor parameters when StallGuard is to be used. This function can only be used when a module is connected that features StallGuard. This is checked when the StallGuard adjusting tool is selected in the "Setup" menu. After this has been successfully checked the StallGuard adjusting tool is displayed.

First, select the axis that is to be used in the "Motor" area.

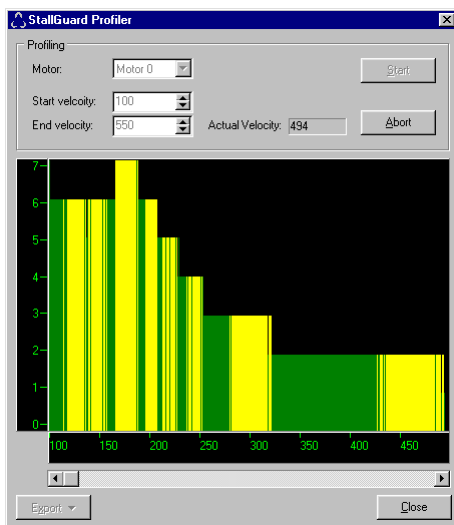
Now you can enter a velocity and an acceleration value in the "Drive" area and then click "Rotate Left" or "Rotate Right". Clicking one of these buttons will send the necessary commands to the module so that the motor starts running. The red bar in the "StallGuard" area on the right side of the windows displays the actual load value. Use the slider to set the StallGuard threshold value. If the load value reaches this value the motor stops. Clicking the "Stop" button also stops the motor.

All commands necessary to set the values entered in this dialogue are displayed in the "Commands" area at the bottom of the window. There, they can be selected, copied and pasted into the TMCL editor.

Figure 5.2: StallGuard adjusting tool

5.2.2 StallGuard profiler

The StallGuard profiler is a utility that helps you find the best parameters for using stall detection. It scans through given velocities and shows which velocities are the best ones. Similar to the StallGuard adjusting tool it can only be used together with a module that supports StallGuard. This is checked right after the StallGuard profiler has been selected in the "Setup" menu. After this has been successfully checked the StallGuard profiler window will be shown.



First, select the axis that is to be used. Then, enter the "Start velocity" and the "End velocity". The start velocity is used at the beginning of the profile recording. The recording ends when the end velocity has been reached. Start velocity and end velocity must not be equal. After you have entered these parameters, click the "Start" button to start the StallGuard profile recording. Depending on the range between start and end velocity this can take several minutes, as the load value for every velocity value is measured ten times. The "Actual velocity" value shows the velocity that is currently being tested and so tells you the progress of the profile recording. You can also abort a profile recording by clicking the "Abort" button. The result can also be exported to Excel or to a text file by using the "Export" button.

Figure 5.3: The StallGuard Profiler

5.2.2.1 The result of the StallGuard profiler

The result is shown as a graphic in the StallGuard profiler window. After the profile recording has finished you can scroll through the profile graphic using the scroll bar below it. The scale on the vertical axis shows the load value: a higher value means a higher load. The scale on the horizontal axis is the velocity scale. The colour of each line shows the standard deviation of the ten load values that have been measured for the velocity at that point. This is an indicator for the vibration of the motor at the given velocity. There are three colours used:

- Green: The standard deviation is very low or zero. This means that there is effectively no vibration at this velocity.
- Yellow: This colour means that there might be some low vibration at this velocity.

- Red: The red colour means that there is high vibration at that velocity.

5.2.2.2 Interpreting the result

In order to make effective use of the StallGuard feature you should choose a velocity where the load value is as low as possible and where the colour is green. The very best velocity values are those where the load value is zero (areas that do not show any green, yellow or red line). Velocities shown in yellow can also be used, but with care as they might cause problems (maybe the motor stops even if it is not stalled).

Velocities shown in red should not be chosen. Because of vibration the load value is often unpredictable and so not usable to produce good results when using stall detection.

As it is very seldom that exactly the same result is produced when recording a profile with the same parameters a second time, always two or more profiles should be recorded and compared against each other.

5.3 Reference switches

With reference switches, an interval for the movement of the motor or the zero point can be defined. Also a step loss of the system can be detected, e.g. due to overloading or manual interaction, by using a travel-switch. The TMCM-612 has one left and right reference switch input for each motor.

Motor X	Direction	Name	Limits	Description
0, 1, 2, 3, 4, 5	In	R	TTL	Right reference switch input for Motor #X
0, 1, 2, 3, 4, 5	In	L	TTL	Left reference switch input for Motor #X

Table 5.2: Pinout reference switches

Note: 10k pullup resistors for reference switches are included on the module.

5.3.1 Left and right limit switches

The TMCM-612 can be configured so that a motor has a left and a right limit switch (Figure 5.4). *The motor then stops when the traveler has reached one of the limit switches.*

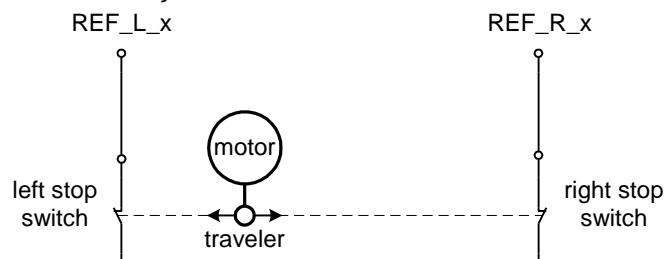


Figure 5.4: Left and right limit switches

5.3.2 Triple Switch Configuration

It is possible to program a tolerance range around the reference switch position. This is useful for a triple switch configuration, as outlined in Figure 5.5. In that configuration two switches are used as automatic stop switches, and one additional switch is used as the reference switch between the left stop switch and the right stop switch. The left stop switch and the reference switch are wired together. The center switch (travel switch) allows for a monitoring of the axis in order to detect a step loss.

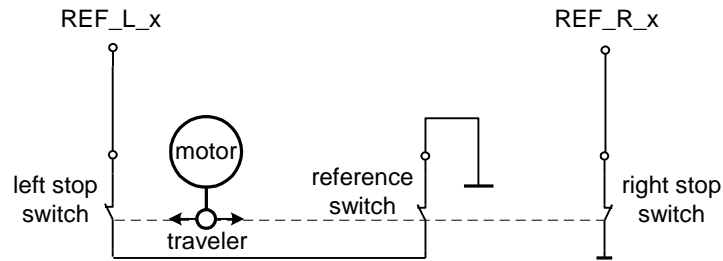


Figure 5.5: Limit switch and reference switch

5.3.3 One Limit Switch for circular systems

If a circular system is used (Figure 5.6), only one reference switch is necessary, because there are no end-points in such a system.

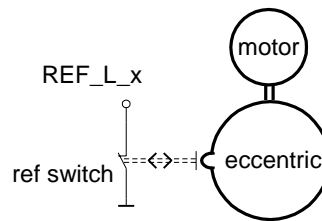


Figure 5.6: One reference switch

5.4 USB

To make use of the USB interface, a device driver has to be installed first. There is a device driver shipped on the CD that can be used with Windows 98, Windows ME, Windows 2000 and Windows XP. The device driver can not be used with Windows NT4 and Windows 95 as these operating systems do not support USB at all. In most Linux distributions the driver for the USB chip used on the TMCM-612 device (FT245BM) is already included in the kernel. When the TMCM-612 module is connected to the USB interface of a PC for the first time, you will be prompted for a driver by the operating system. Now, insert the CD and select the "tmcm-612.inf" file there. The driver will then be installed and is now ready to be used.

Please note that the TMCM-612 always needs its own power supply and is not powered by the USB bus. So the module will not be recognized if it is not powered.

To use the USB connection with the TMCL IDE, at least version 1.31 of the IDE is needed. In the "Connection" screen of the "Options" dialog, select "USB (TMCM-612)" and then select the module in the "Device" list box. Now all communication between the TMCL IDE and the module uses the USB interface.

To control the TMCM-612 module from your own PC applications the USB version of the "TMCL Wrapper DLL" is needed.

6 Putting the TMC612 into Operation

On the basis of a small example it is shown step by step how the TMC612 is set into operation. Experienced users could skip this chapter and proceed to chapter 7:

Example: The following application is to implement with the TMCL-IDE Software development environment in the TMC612 module. For data transfer between the host PC and the module the RS-232 interface is employed.

A formula how "speed" is converted into a physical unit like rotations per seconds can be found in 7.1 Calculation: Velocity and Acceleration vs. Microstep- and Fullstep-Frequency

- Turn Motor 0 left with speed 500
- Turn Motor 1 right with speed 500
- Turn Motor 2 with speed 500, acceleration 5 and move between position +10000 and -10000.

Step 1: Connect the RS-232 Interface as specified in 3.2.6.

Step 2: Connect the motors as specified in 3.2.4.

Step 3: Connect the power supply.

Step 4: Switch on the power supply. An on-board LED should starting to flash. This indicates the correct configuration of the microcontroller.

Step 5: Start the TMCL-IDE Software development environment. Type in the following TMCL programme:
A description for the TMCL commands can be found in Appendix A.

```
//A simple example for using TMCL and TMCL-IDE

    ROL 0, 500                //Rotate motor 0 with speed 500
    WAIT TICKS, 0, 500
    MST 0
    ROR 1, 250                //Rotate motor 1 with 250
    WAIT TICKS, 0, 500
    MST 1

    SAP 4, 2, 500             //Set max. Velocity
    SAP 5, 2, 50              //Set max. Acceleration
Loop: MVP ABS, 2, 10000       //Move to Position 10000
    WAIT POS, 2, 0            //Wait until position reached
    MVP ABS, 2, -10000        //Move to Position -10000
    WAIT POS, 2, 0            //Wait until position reached
    JA Loop                   //Infinite Loop
```

Step 6: Click on Icon "Assemble" to convert the TMCL into machine code.
Then download the program to the TMC612 module via the Icon "Download".

Step 7: Press Icon "Run". The desired program will be executed.

The program is stored to the EEPROM of the microcontroller. If the TMCL autostart option in "Configure Module" tab "Other" is activated the program will be executed at each power on.

A documentation about the TMCL operations can be found in the TMCL reference manual. The next chapter discusses additional operations to turn the TMC612 into a high performance motion control system.

7 TMC428 Operational Description

7.1 Calculation: Velocity and Acceleration vs. Microstep- and Fullstep-Frequency

The values of the parameters, sent to the TMC428 do not have typical motor values, like rotations per second as velocity. But these values can be calculated from the TMC428-parameters, as shown in this document. The parameters for the TMC428 are:

Signal	Description	Range
f_{CLK}	clock-frequency	0..16 MHz
velocity	-	0..2047
a_max	maximum acceleration	0..2047
pulse_div	divider for the velocity. The higher the value is, the less is the maximum velocity default value = 0	0..13
ramp_div	divider for the acceleration. The higher the value is, the less is the maximum acceleration default value = 0	0..13
Usrs	microstep-resolution (microsteps per fullstep = 2^{Usrs})	0..7 (a value of 7 is internally mapped to 6 by the TMC428)

Table 7.1: TMC428 Velocity parameters

The **microstep-frequency** of the stepper motor is calculated with

$$usf[Hz] = \frac{f_{CLK}[Hz] \cdot velocity}{2^{pulse_div} \cdot 2048 \cdot 32} \quad \text{with usf: microstep-frequency}$$

To calculate the **fullstep-frequency** from the microstep-frequency, the microstep-frequency must be divided by the number of microsteps per fullstep.

$$fsf[Hz] = \frac{usf[Hz]}{2^{Usrs}} \quad \text{with fsf: fullstep-frequency}$$

The change in the pulse rate per time unit (pulse frequency change per second – the **acceleration a**) is given by

$$a = \frac{f_{CLK}^2 \cdot a_{max}}{2^{pulse_div + ramp_div + 29}}$$

This results in an acceleration in fullsteps of:

$$af = \frac{a}{2^{Usrs}} \quad \text{with af: acceleration in fullsteps}$$

Example:

$f_{CLK} = 16 \text{ MHz}$
 velocity = 1000
 $a_{max} = 1000$
 pulse_div = 1
 ramp_div = 1
 usrs = 6

$$\text{msf} = \frac{16\text{MHz} \cdot 1000}{2^1 \cdot 2048 \cdot 32} = \underline{\underline{122070.31\text{Hz}}}$$

$$\text{fsf}[\text{Hz}] = \frac{122070.31}{2^6} = \underline{\underline{1907.34\text{Hz}}}$$

$$a = \frac{(16\text{MHz})^2 \cdot 1000}{2^{1+1+29}} = \underline{\underline{119.21 \frac{\text{MHz}}{\text{s}}}}$$

$$af = \frac{119.21 \frac{\text{MHz}}{\text{s}}}{2^6} = \underline{\underline{1.863 \frac{\text{MHz}}{\text{s}}}}$$

If the stepper motor has e.g. 72 fullsteps per rotation, the number of rotations of the motor is:

$$\text{RPS} = \frac{\text{fsf}}{\text{fullsteps per rotation}} = \frac{1907.34}{72} = 26.49$$

$$\text{RPM} = \frac{\text{fsf} \cdot 60}{\text{fullsteps per rotation}} = \frac{1907.34 \cdot 60}{72} = 1589.46$$

8 TMCL

Like most of the other Trinamic motion control modules, the TMC612 is also equipped with TMCL, the Trinamic Motion Control Language. The TMCL language in this unit has been extended so that six motors can be controlled with the normal TMCL commands. With a few exceptions, all commands work as described in the "TMCL Reference and Programming Manual". The main difference is that the range of the "Motor" parameter has been extended to six motors: its range is now 0..5 so that all commands that need a motor number can address all six motors. All axis parameters can be set independently for each motor.

TMCL, the TRINAMIC Motion Control Language, is described in a separate documentation, the TMCL Reference and Programming Manual. This manual is provided on the TMC TechLib CD and on the web site of TRINAMIC: www.trinamic.com. Please refer to these sources for updated data sheets and application notes.

The TMC TechLib CD-ROM including data sheets, application notes, schematics of evaluation boards, software of evaluation boards, source code examples, parameter calculation spreadsheets, tools, and more is available from TRINAMIC by request and comes with each module.

8.1 Differences in TMCL commands

There are only two commands that are slightly different on the TMC612 module. They are as follows:

8.1.1 MVP COORD

The MVP ABS and the MVP REL commands are the same as with the other modules, but the MVP COORD command has some more options. For this reason the "motor" parameter with the MVP COORD command is interpreted as follows on the TMC610 module:

- Moving only one motor: set the "Motor" parameter to the motor number (0..5).
- Moving multiple motors without interpolation: Set bit 7 of the "Motor" parameter. Now the bits 0..5 of the "Motor" parameter define which motors are to be started. Each of these bits stands for one motor.
- Moving multiple motors using interpolation: Set bit 6 of the "Motor" parameter. . Now the bits 0..5 of the "Motor" parameter define which motors are to be moved using interpolation. Each of these bits stands for one motor. It is not possible to start a group of more than three motors using

interpolation. However, it is possible to start one group of three motors right after starting a group of other three motors.

Examples:

- MVP COORD, \$47, 2 moves motors 0, 1 and 2 to coordinate 2 using interpolation.
- MVP COORD, \$87, 5 moves motors 0, 1 and 2 to coordinate 5 without using interpolation.

Warning: the interpolation feature is not available in firmware versions prior to 6.31. If necessary, get the latest firmware from the Trinamic website and upgrade your module.

8.1.2 WAIT RFS

Waiting for the reference search of multiple motors with the WAIT RFS command is not supported. The range of the "motor" parameter is 0..5 (for the six motors). To wait for multiple reference searches, just use one WAIT RFS command for each motor.

8.2 Additional commands

Some of the user defined commands are used to access the additional features of the TMC612 like ADC, DAC, reference switch polarity and the additional data acquisition RAM.

8.2.1 Read the ADC: Uf0

The Uf0 command is used to read the additional 16-bit ADC. The command selects the channel, starts the conversion and then returns the result. The "motor/bank" parameter is used to select the channel (0..7). In TMCL direct mode use the manual input. The result is in the range of 0..65535, where 65535 means +10V. The other parameters of this command are not used and should be set to zero.

Example: To read channel 3 of the ADC, use Uf0 0, 3, 0.

8.2.2 Write to the DAC: Uf1

The Uf1 command is used to set the value of the additional 10-bit DACs. So, the value can be set between 0 and 1023. A value of 1023 equals to an output voltage of +10V. The "motor/bank" parameter is used to specify the channel (0..7), and the "value" parameter is used to specify the output value.

The "type" parameter specifies if a constant value or the accumulator or the x register is to be output on the DAC (type=0 outputs a constant value, type=1 outputs the accumulator, type=2 outputs the x register).

Example:

- To set DAC channel 5 to 517, use Uf1 0, 5, 517.
- To set DAC channel 5 to the value of the accumulator, use Uf1 1, 5, 0.
- To set DAC channel 5 to the value of the x register, use Uf1 2, 5, 0.

8.2.3 Set the polarity of the stop switches: Uf2

The Uf2 command is used to set the stop switch polarity for every motor. The "value" parameter of the command is used as a bit mask, where bit 0 stands for motor 0, bit 1 for motor 1 and so on. When the corresponding bit is set the polarity of the stop switches of that motor will be inverted.

The "type" and the "motor/bank" parameter of this command are not used and should be set to zero.

8.2.4 Read from the additional data RAM: Uf3

With firmware revision 6.35 or higher, the commands Uf3 and Uf4 can be used to access the additional RAM. The Uf3 command is used to read data from the additional data acquisition RAM. Depending on the "type" parameter the Uf3 command has six different functions:

- Uf3 0, 0, <n>: Set RAM read pointer to the value <n>.
- Uf3 1, 0, 0: Set RAM read pointer to a value stored in the accumulator.
- Uf3 2, 0, 0: Get RAM read pointer (copy its value to the accumulator).
- Uf3 3, 0, 0: Read value from RAM at the address given by the RAM read pointer.
- Uf3 4, 0, 0: Read value from RAM at the address given by the RAM read pointer, then increment the RAM read pointer by one so that it points to the next memory location.
- Uf3 5, 0, <n>: Read value from RAM at a fixed address given by the value <n>.

With these commands it is possible to read data stored in the additional RAM into the accumulator register so that it can be processed further. Of course these commands can also be used in direct mode so that e.g. a host can read data that has previously been stored in the RAM e.g. by a TMCL program.

The RAM read pointer makes it possible to access the RAM at an address that has previously been set. It can also be incremented automatically. So the accumulator register does not have to be used for such purposes.

The UF3 and UF4 commands address the RAM as an array of 32 bit words so up to 32767 values can be stored in the RAM using these commands (the RAM read pointer should not be set to values exceeding 32767).

8.2.5 Write to the additional data RAM: UF4

The UF4 command is used to write data to the additional data acquisition RAM. Depending on the "type" parameter the UF4 command has six different functions:

- UF4 0, 0, <n>: Set RAM write pointer to the value <n>.
- UF4 1, 0, 0: Set RAM write pointer to a value stored in the accumulator.
- UF4 2, 0, 0: Get RAM write pointer (copy its value to the accumulator).
- UF4 3, 0, 0: Write contents of accumulator to the RAM at the address given by the RAM write pointer.
- UF4 4, 0, 0: Write contents of accumulator to the RAM at the address given by the RAM write pointer and then increment the RAM write pointer so that it points to the next memory location.
- UF4 5, 0, <n>: Write contents of accumulator to the RAM at a fixed address given by the value <n>.
- UF4 6, 0, <n>: Write a fixed value <n> to the RAM at the address given by the RAM write pointer.
- UF4 7, 0, <n>: Write a fixed value <n> to the RAM at the address given by the RAM write pointer and then increment the RAM write pointer so that it points to the next memory location.

With these commands it is possible to write data to the additional RAM so that it can be stored for further processing (e.g. taking samples from the ADC for later processing). Of course these commands can also be used in direct mode so that a host can write values to the RAM to be processed by the TMCM-612 then.

The RAM write pointer makes it possible to access the RAM at an address that has previously been set. The RAM write pointer can also be incremented automatically after each write access so that the accumulator does not have to be used for this purpose. This command is available in firmware revision 6.35 or higher.

In the following example, ADC values are measured and stored to the RAM every second. The examples makes use of the automatic increment feature.

```

UF4 0, 0, 0 //Set RAM write pointer to 0
Loop:
GIO 0, 1 //Read ADC 0
UF4 4, 0, 0 //Store the value to the RAM with auto increment
WAIT TICKS, 0, 10
UF4 2, 0, 0 //Check if RAM already full
COMP 32767
JC LE, Loop

```

9 Revision History

9.1 Documentation Revision

Version	Date	Author	Description
1.00	11-Nov-04	OK	Initial version
1.01	07-Nov-05	OK	AD and DAC voltages corrected
1.10	15-Sep-06	HC	Major Revision
1.11	16-May-08	OK	Interpolation feature added
1.12	1-Apr-09	OK	Commands UF3 and UF4 added
1.13	29-Mar-12	OK	Command UF1 extended (firmware V6.37)

Table 9.1: Documentation Revisions

9.2 Firmware Revision

Version	Comment	Description
6.00	Initial Release	Please refer to TMCL documentation
6.31		Also provides the interpolation feature
6.35		Additional RAM can be addressed using commands UF3 and UF4
6.37		UF1 command extended so that also the accumulator or x register can be output on the DAC.

Table 9.2: Firmware Revisions