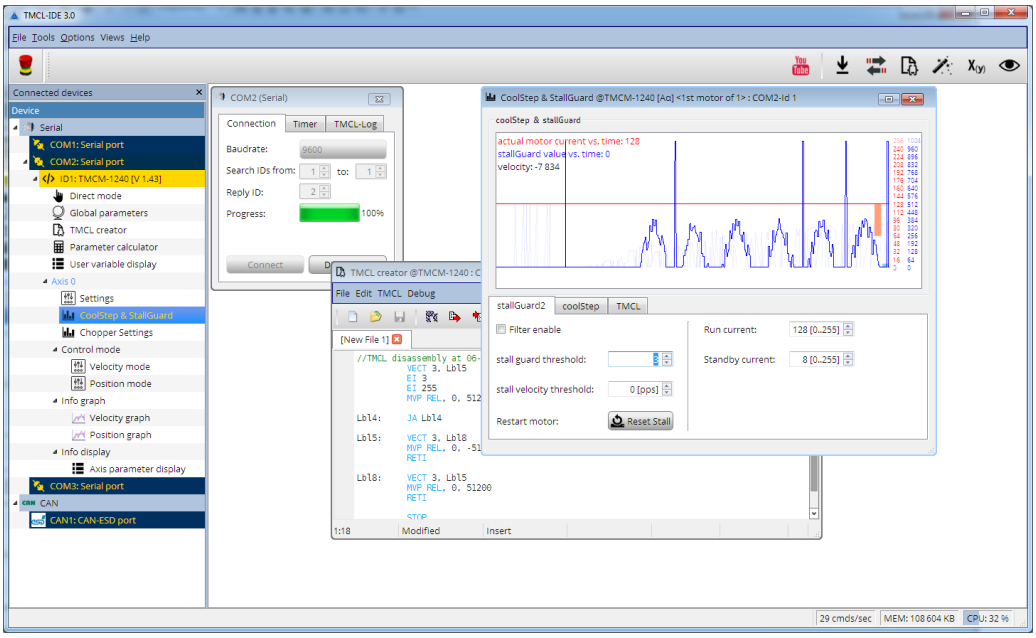
TMCL-IDE

Revision V3.3.0.0 | Document Revision V3.05 • 2021-MAR-04

The TMCL-IDE is an integrated development environment made for developing applications that use Trinamic modules and chips. It contains a set of tools for easily setting parameters, for visualizing measured data and for developing and debugging stand-alone applications with TMCL™, the Trinamic Motion Control Language.
The TMCL-IDE is available free of charge and runs on Windows 7, Windows 8.x or Windows 10. A version for Linux is also available free of charge.



Download newest version at: www.trinamic.com

Read entire documentation.

TRINAMIC
MOTION CONTROL
Now part of Maxim Integrated

Contents

1 Introduction

1.1 Getting the TMCL-IDE

The TMCL-IDE can be downloaded free of charge from software section of the TRINAMIC website: https://www.trinamic.com/support/software/tmcl-ide/#c414. The latest version can always be found there. Also older versions can be downloaded from there if needed.

1.2 Installing the TMCL-IDE

1.2.1 Windows

It is always possible to download a version with automated installation (filename: TMCL-IDE-3.x.x.x-Setup.exe). After downloading this file, just double click it to start the installation process. For ease of installation we recommend using this file.
There is also a non-install version. This is a ZIP file that contains all necessary files. After downloading this file, unpack it to one directory.

1.2.2 Linux

The Linux version can be found on GitHub. Please follow the link to GitHub from the Software section of the TRINAMIC website. Here you can also find detailed instructions for installing the TMCL-IDE on Linux.

1.3 Supported Interfaces

For connecting to a Trinamic module or to a Trinamic evaluation board, different interfaces can be used. These are USB, RS232, RS485 and CAN. Each module or evaluation board that is equipped with a USB interface can be connected directly via USB. It will then automatically be recognized by the TMCL-IDE.
For modules equipped with RS232 or RS485 interface, an appropriate interface will also be needed on the PC. Many standard off-the-shelf RS232 and RS485 interfaces can be used.
For connecting via CAN bus a CAN interface that is supported by the IDE will be needed. Table 1 contains a list of all currently supported CAN interfaces.

1.4 Launching the TMCL-IDE

On Windows, run the TMCL-IDE simply by choosing the TMCL-IDE entry from the start menu or by double clicking the TMCL-IDE desktop icon or (mainly if you are using the non-install version) by double clicking the TMCL-IDE.exe file.
On Linux, run the TMCL-IDE.sh script either from the command line or by clicking on it.

At first, a splash screen will appear that shows the progress of loading the program and all of its components. Then, the TMCL-IDE main window will appear.

2 The Main Window

After launching the TMCL-IDE the main window will appear on the screen. The main window contains the following parts:

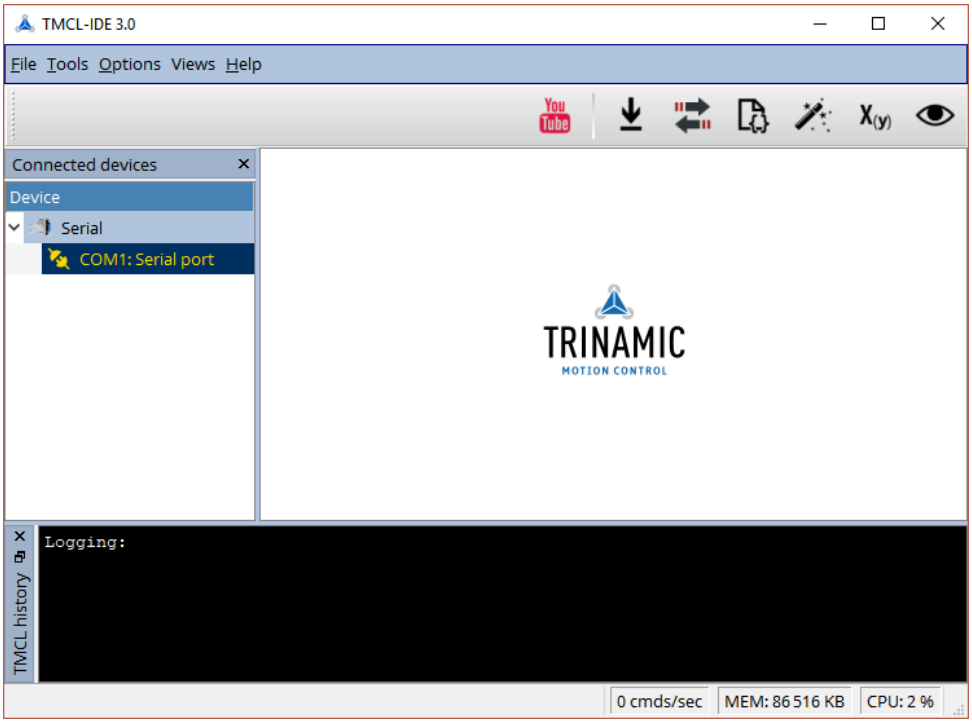


Figure 1: The TMCL-IDE Main Window

2.1 Menu Bar and Status Bar

The menu bar is placed at the top of the main window, the status bar is placed to the bottom. Both bars are not moveable.

Figure 2: The Menu and Status Bar

The status bar shows on the left side actual messages and on the right side the current TMCL command rate, which means the number of requests plus replies per second. Besides this, used memory and CPU load are displayed. The menu commands are sorted in five entries:

- File: Shortcut 'alt gr + p' allows shot of actual tool window as png file and to clipboard.
- Tools: Call container tools.
- Options: Properties of tool windows moving or behavior.
- Views: Hide or show the other windows around the central view.
- Help: Visit TRINAMIC YouTube channel, show some system info, open this document or looking for updates.

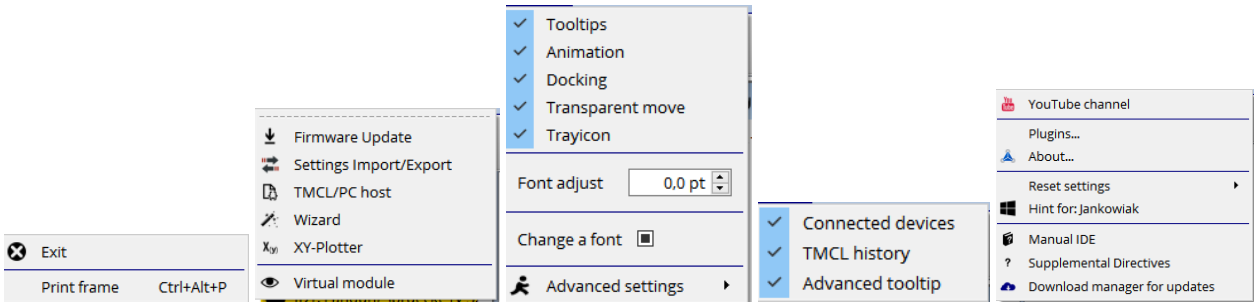


Figure 3: The Menu Bar's entries

The about box gives an overview of the paths where components are installed. An INI file is used to store all settings and is located in home path shown. The working directory is users temporary path plus TMCL-IDE. Some components are generating logging messages to the file debug.log. You can click on the link at the bottom to open this file with your system editor to view and save the content.

Figure 4: The TMCL-IDE About Box

2.2 Tool Bar

Here you can find most needed common tools like firmware update tool, TMCL-PC Host or the compilation of several wizards. Those are the same as menu bar's tools. In the right corner you can get by clicking the icon to open a list of all modules, You can choose any existing module to the related tools

Figure 5: The TMCL-IDE Tool Bar

Clicking on will call the Firmware Update Tool. Flash a given firmware file to the module.
The icon will open the The Settings Export/Import Tool. Choose a module and im- or export the parameter settings using files.
Clicking on will call the TMCL/PC Host. This tool enables writing TMCL instructions for controlling between various modules and their axes.
Call Wizards with . In the wizard tool you can pick a module to have a collection of available wizards.
plots up to four value pairs in a XY graph. Mix any values from any axes from any module.

2.3 Device with Tool Tree

The tree root entries represent the families of various serial physical interfaces: USB, serial communications port, CAN and also non-physical virtual modules. Each root entry contains the connected interfaces and each interface is the parent of one or more connected TMC Module. Each module is the parent of tools depending of its characteristics.

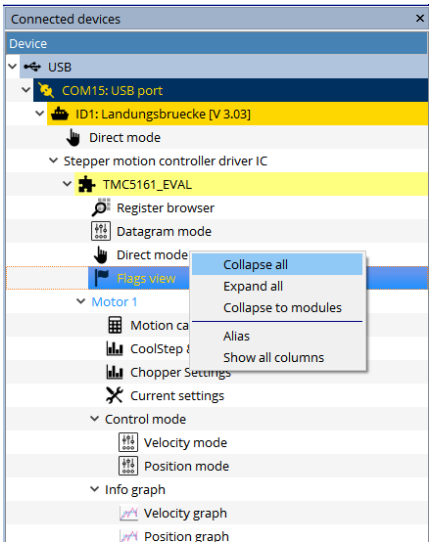


Figure 6: Tree with Connected Devices and their Tools

Mouse right click will open a popup menu. An useful item maybe *Alias* in case some identical modules are connected. *Alias* is a column with editible fields in the module rows so a unique name can be given.

If selected the TMCL history window and/or the advanced tooltip window will also be shown. These, the icon bar and device tree are freely moveable and can be arranged to an own layout.

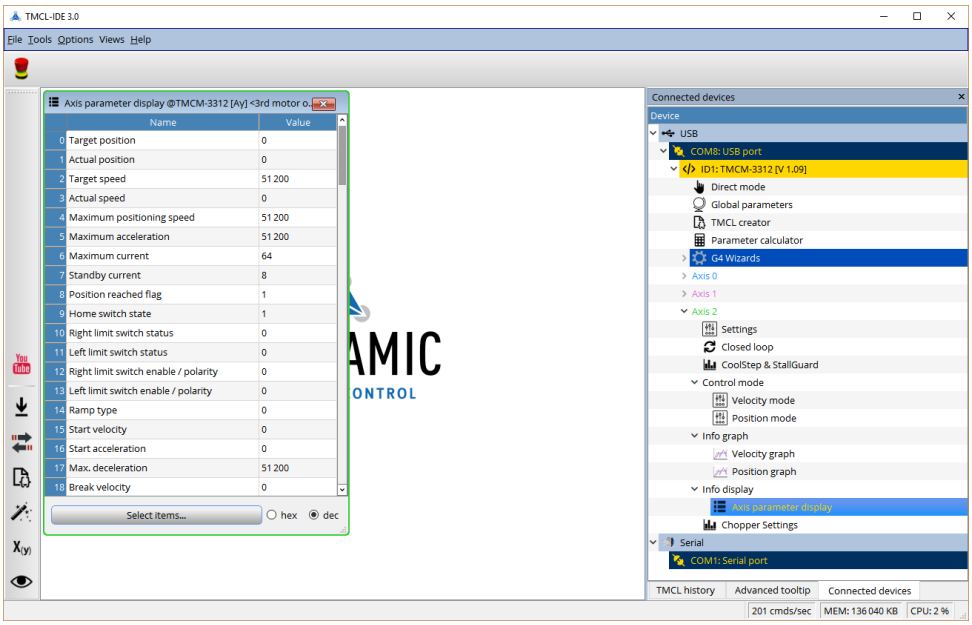


Figure 7: The TMCL-IDE Main Window with user layout

3 Connections

Depending on the host interfaces the module is equipped with there are different ways to connect the module to the PC. Many, but not all modules are equipped with a USB interface which is often the easiest way for a first connection to a PC. But also RS485, RS232 or CAN can be used to connect the module. All modules are equipped with at least one of these interfaces.

3.1 USB

For using a module with USB connection just plug in the USB cable into the module and the PC. Many TRINAMIC modules are also USB powered, but this will only work for configuring the module. USB power is not sufficient for powering motors, so it will always be necessary to connect the module also to a power supply in order to be able to run a motor using USB connection.

After plugging in the USB cable, the module will automatically appear in the module tree on the left hand side of the main window, and the tool tree that contains all tools that can be used with this module will be displayed underneath the module entry in the tree. Depending on the operating system of your PC it might be necessary to install the correct USB driver files for the module you are using. Mostly this will be done automatically by the TMCL-IDE. Sometimes it may also be necessary to install the driver manually. For this purpose, the driver files can be downloaded from the TRINAMIC website.

As all TRINAMIC modules that are equipped with a USB interface use the CDC class (communication device class) they will appear as virtual serial ports. Depending on the operating system they will either be shown as COMxx or /dev/ttyUSBxx, where xx stands for any number allocated by the operating system.

Clicking on the virtual COM port shown in the tree view will open the connection window for this port.

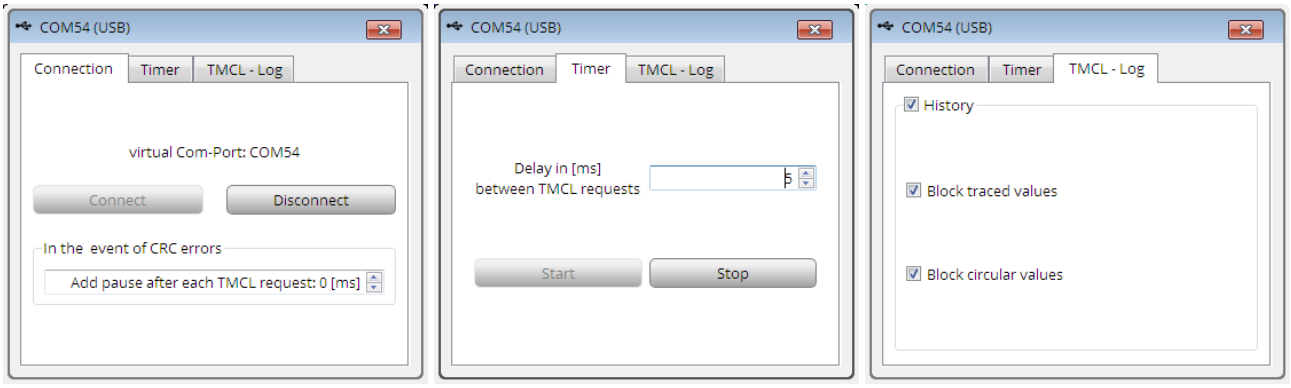


Figure 8: The USB Connection Window

3.1.1 Connection Settings

On the *Connection* tab of the USB connection window general connection settings can be made:

- Using the *Disconnect* button it is possible to temporarily close the USB connection to the module, so that other PC software can connect to the module without having to close the TMCL-IDE itself.
- Use the *Connect* button to re-connect to the module after the connection had been closed using the *Disconnect* button. Please be sure that no other program is accessing the module via the USB interface before re-connecting.

- Pause between TMCL commands: in some rare cases it seems to be necessary to insert pauses between commands as otherwise errors might occur. If this happens, set this value higher than zero. Normally this setting can be left at zero.

3.1.2 Timer Settings

Use the *Timer* tab of the USB connection window to control the timer which is used for regularly polling values from the module. This is needed for tools that regularly need to update values that they are displaying, like the Position Graph or the Velocity Graph for example. The following settings can be made here:

- *Delay between TMCL requests:* This is the polling interval. By default this is set to 5ms, but can be set lower or higher if needed.
- Use the *Stop* button to stop the timer. This will stop polling values from the module. Values being displayed in most tools will not be updated any more then.
- Use the *Start* button to start the timer. Values displayed in the tools will then be updated again.

3.1.3 TMCL Log Settings

Use the *TMCL Log* tab of the USB connection window to control which commands are being displayed in the TMCL Log window:

- The *History* checkbox generally switches on or off the history display for this module.
- *Block Traced Values:* This function prevents values that are regularly traced by the tools from being displayed in the TMCL Log window. Switching on this option considerably reduces the amount of data being displayed in the TMCL Log window.
- *Block Circular Values:* This function prevents values that are polled by the tools using the timer from being displayed in the TMCL Log window. Turning on this option also considerably reduces the amount of data being displayed in the TMCL Log window.

3.2 RS485 / RS232

Many TRINAMIC modules can also be connected via RS485, RS232 or a TTL level serial interface. The TMCL-IDE can also via these types of serial interfaces. For this purpose a serial port (RS485, RS232 or TTL level) connected to the PC (for example via USB) or built into the PC (for example as a PCI card) is necessary. Serial ports from most manufacurers can be used for this purpose. Take care that has been properly installed before trying to use it. Please see also the hardware manual of your module on how to properly connect the module to the serial port. Using RS485 it is also possible to connect more than one module to one port.

All serial ports (regardless of RS485, RS232 or TTL level) are shown in the tree view on the left hand side of the main window. Depending on the operating system their names are either COMxx or /dev/ttyxx where xx stands for any number allocated by the operating system. Click on the appropriate COM port (the one your module is connected to) to display the connection window for the specific port.

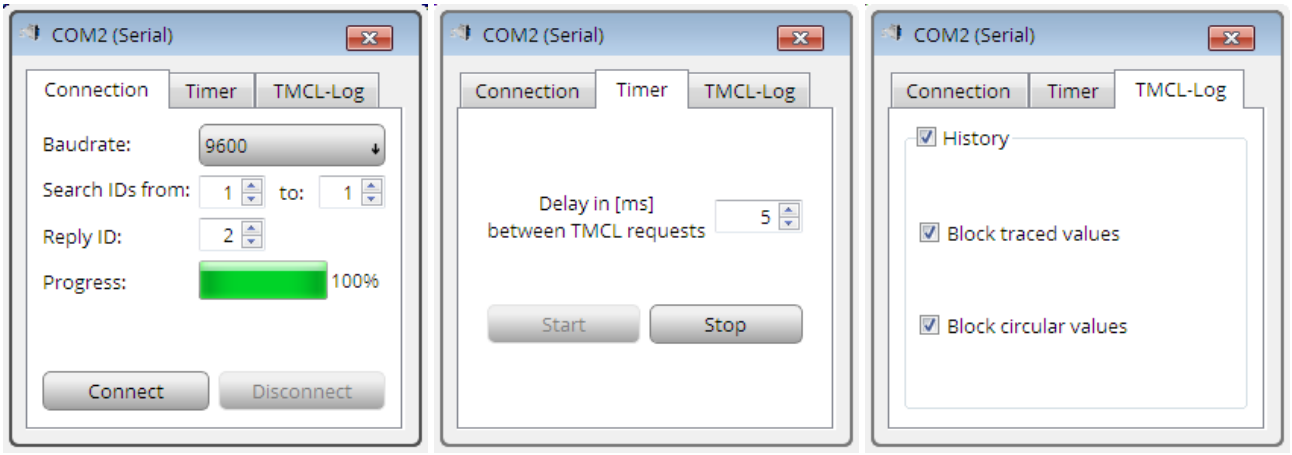


Figure 9: The Serial Port Connection Window

3.2.1 Connection Settings

Use the *Connection* tab to make general settings for the connection and to connect to your module. The following options are available:

- *Baudrate:* Choose the baud rate of the serial port here. The factory default value on all TRINAMIC modules is 9600bps, so this value is always good for a new module. Change this if you have set up your module to use a different baud rate.
- *Search IDs from/to:* It is possible to connect more than one module to an RS485 bus. For this reason, the TMCL-IDE can search for more than one module on the serial port. Enter the ID of the first module connected to the bus and the ID of the last module connected to the bus here. If only one module is connected you can normally leave both values at 1, as this is also the factory default setting on TRINAMIC modules. Or if the module is set to a different ID, set both values to that ID. If you are not sure about the ID setting of a module you can also enter from 1 to 255 so that the TMCL-IDE will automatically scan through all possible serial module IDs, but this will take some time.
- *Reply ID:* The reply ID of the connected modules. This should normally be the same on all modules. The factory default setting is 2.
- *Connect:* Click the *Connect* button to open the connection and to start searching for modules connected to the serial port. The search progress will be shown by the progress indicator. All modules that have been found will appear on the tree view on the left hand side of the main window.
- *Disconnect:* Click here to close the connection.

3.2.2 Timer Settings

Use the *Timer* tab of the serial port connection window to control the timer which is used for regularly polling values from the module. This is needed for tools that regularly need to update values that they are displaying, like the Position Graph or the Velocity Graph for example. The following settings can be made here:

- *Delay between TMCL requests:* This is the polling interval. By default this is set to 5ms, but can be set lower or higher if needed. The lowest possible value depends on the selected baud rate.
- Use the *Stop* button to stop the timer. This will stop polling values from the module. Values being displayed in most tools will not be updated any more then.
- Use the *Start* button to start the timer. Values displayed in the tools will then be updated again.

3.2.3 TMCL Log Settings

Use the *TMCL Log* tab of the serial port connection window to control which commands are being displayed in the TMCL Log window:

- The *History* checkbox generally switches on or off the history display for this port.
- *Block Traced Values:* This function prevents values that are regularly traced by the tools from being displayed in the TMCL Log window. Switching on this option considerably reduces the amount of data being displayed in the TMCL Log window.
- *Block Circular Values:* This function prevents values that are polled by the tools using the timer from being displayed in the TMCL Log window. Turning on this option also considerably reduces the amount of data being displayed in the TMCL Log window.

3.3 CAN

Modules that are equipped with a CAN interface can also be connected via the CAN bus. For this purpose, a CAN interface is needed. As CAN interfaces from different manufacturers also have different programming interfaces, not all of them are supported by the TMCL-IDE. Table 1 contains a list of all CAN interfaces supported by the TMCL-IDE. Only interfaces contained in this (steadily growing) list of CAN interfaces are supported by the TMCL-IDE. Maybe other CAN interfaces from the same manufacurers will also work with the TMCL-IDE, but have not been tested by TRINAMIC.

Please see the hardware manual of your module and also the manual of your CAN interface on how to properly connect the module to the CAN interface. Using CAN it is also possible to connect more than one module to the same bus.

List of supported CAN Interfaces	
Manufacturer	Type
ESD	CAN-USB/2 CAN-USB/Micro other CAN-USB interfaces
Kvaser	Leave Light V2
Peak	PCAN-USB (IPEH-002021 or IPEF-002022) PCAN-USB Pro
IXXAT	USB-to-CAN compact USB-to-CAN V2 compact

Table 1: List of supported CAN Interfaces

All supported CAN interfaces that are attached to the PC are shown in the tree view on the left hand side of the main window. Click on the appropriate interface in the tree view to open the connection window for this interface.

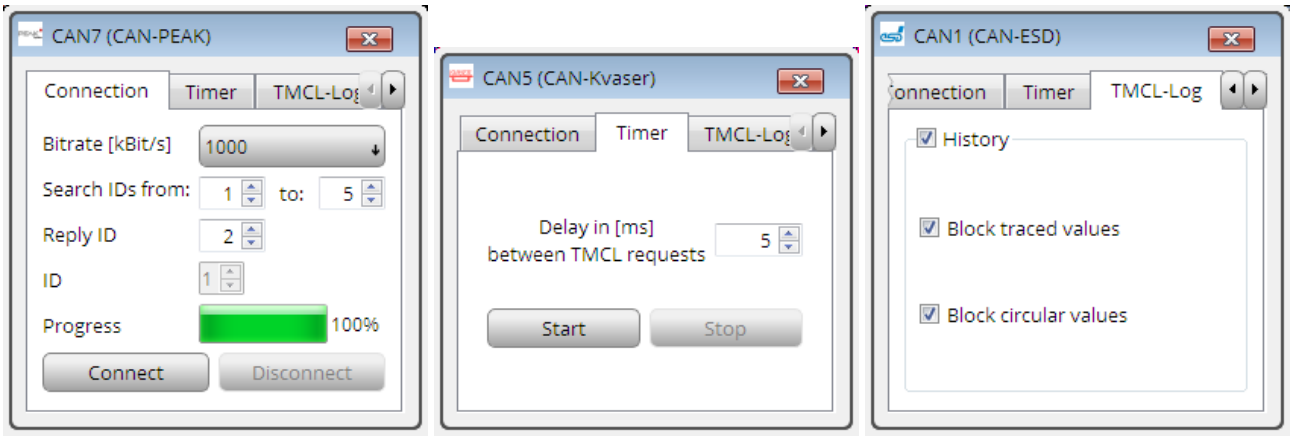


Figure 10: The CAN Bus Connection Window

3.3.1 Connection Settings

Use the *Connection* tab to make general settings for the connection and to connect to your module. The following options are available:

- *Bitrate:* Select the desired CAN bit rate here. The factory default setting on most TRINAMIC modules is 1000kBit/s. Change this if you have set up your module to use a different bit rate.
- *Search IDs from/to:* It is possible to connect more than one module to a CAN bus. For this reason, the TMCL-IDE can search for more than one module on the CAN bus. Enter the ID of the first module connected to the bus and the ID of the last module connected to the bus here. If only one module is connected you can normally leave both values at 1, as this is also the factory default setting on TRINAMIC modules. Or if the module is set to a different ID, set both values to that ID. If you are not sure about the ID setting of a module you can also enter from 1 to 2047 so that the TMCL-IDE will automatically scan through all possible CAN module IDs, but this will take some time.
- *Reply ID:* The reply ID of the connected modules. This should normally be the same on all modules. The factory default setting is 2.
- *ID* or *Base net:* Some CAN interfaces are equipped with more than one CAN port. The port to be used can be specified here.
- *Connect:* Click the *Connect* button to open the connection and to start searching for modules connected to the CAN bus. The search progress will be shown by the progress indicator. All modules that have been found will appear on the tree view on the left hand side of the main window.
- *Disconnect:* Click here to close the connection.

3.3.2 Timer Settings

Use the *Timer* tab of the CAN connection window to control the timer which is used for regularly polling values from the module. This is needed for tools that regularly need to update values that they are displaying, like the Position Graph or the Velocity Graph for example. The following settings can be made here:

- *Delay between TMCL requests:* This is the polling interval. By default this is set to 5ms, but can be set lower or higher if needed. The lowest possible value depends on the selected bit rate.

- Use the *Stop* button to stop the timer. This will stop polling values from the module. Values being displayed in most tools will not be updated any more then.
- Use the *Start* button to start the timer. Values displayed in the tools will then be updated again.

3.3.3 TMCL Log Settings

Use the *TMCL Log* tab of the CAN connection window to control which commands are being displayed in the TMCL Log window:

- The *History* checkbox generally switches on or off the history display for this port.
- *Block Traced Values:* This function prevents values that are regularly traced by the tools from being displayed in the TMCL Log window. Switching on this option considerably reduces the amount of data being displayed in the TMCL Log window.
- *Block Circular Values:* This function prevents values that are polled by the tools using the timer from being displayed in the TMCL Log window. Turning on this option also considerably reduces the amount of data being displayed in the TMCL Log window.

3.3.4 Hints for Peak CAN Interfaces

After installing a Peak CAN interface for the first time it might not work together with the TMCL-IDE. This is caused by the fact that none of Peak-Systems setups install the PCANBasic.dll, which is needed by the TMCL-IDE (this DLL contains the basic Peak CAN API used by the TMCL-IDE). You need to copy the PCANBasic.dll (from the Peak CD/DVD or from the product CD section of the Peak website) manually to your Windows System directory.

- On Windows 32-bit systems:
  - PCANBasic.dll 32 bit to Windows\System32.
- On Windows 64-bit systems:
  - PCANBasic.dll 32 bit to Windows\SysWOW64
  - PCANBasic.dll 64 bit to Windows\System32

4 Module specific Tools

Depending on the connected module or evaluation board, different tools appear on the tool tree. The sections in this chapter describe each of these tools.

4.1 The Main Module Tool

Click on the module name itself in the module tree to open the main module tool. This tool provides some information about the connected module itself.

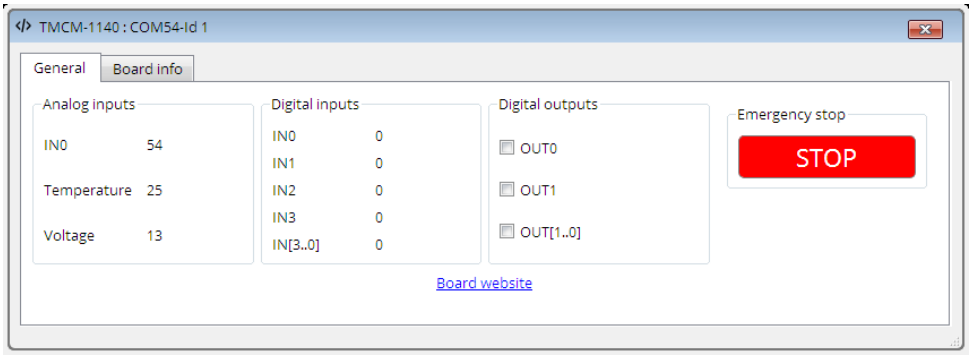


Figure 11: The Main Module Tool

On the *General* tab the states of all additional inputs and outputs of the module are shown. The states of additional outputs can also be changed (if the module is equipped with additional outputs). The *Emergency Stop* button stops all the TMCL program execution on the module and also stops all motors connected to the module.

The *Board info* tab provides some technical information about the module and its capabilities.

4.2 The Direct Mode

As every module that supports TMCL also supports the direct mode this tool is provided for every connected TMCL module. Direct mode means that each TMCL command that has been sent to the module directly gets executed.

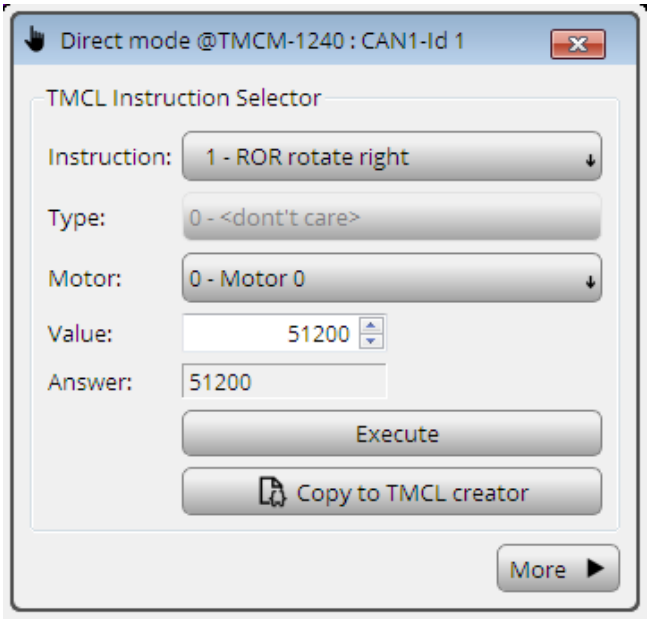


Figure 12: The Direct Mode Tool

Choose a command from the instruction selection menu. Depending on the command, you can also set the type and motor/bank parameter as well as the value parameter. Click the *Execute* button to send the command to the module. The reply value sent back from the module will be displayed in the *Answer* field.

The *Copy to TMCL Creator* button copies the command to the TMCL Creator.

In order to be able to see and also to modify the binary representation of a TMCL command, click the *More* button. The Direct Mode window will then be extended as can be seen in figure 13.

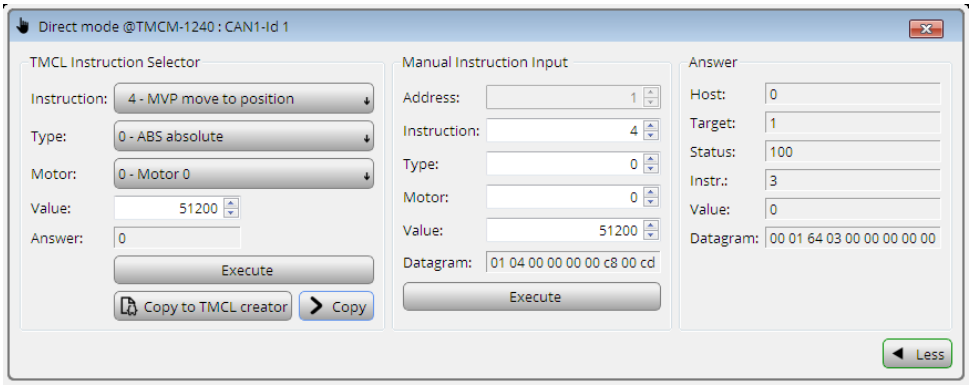


Figure 13: The Direct Mode Tool (extended)

Clicking *Copy* button that now appears in the *TMCL Instruction Selector* area copies a command choosen here to the *Manual Instruction Input* area. Here it can be changed further (which is not often needed by normal users). The resulting TMCL datagram that represents this command can also be seen here. Clicking the *Execute* button in the *Manual Instruction Input* area sends this command to the module. The reply datagram will be displayed in the *Answer* section.

4.3 The Global Parameters Tool

Using the Global Parameters tool, the global parameters of a module can be set. These are parameters like RS232 baud rate or CAN bit rate.

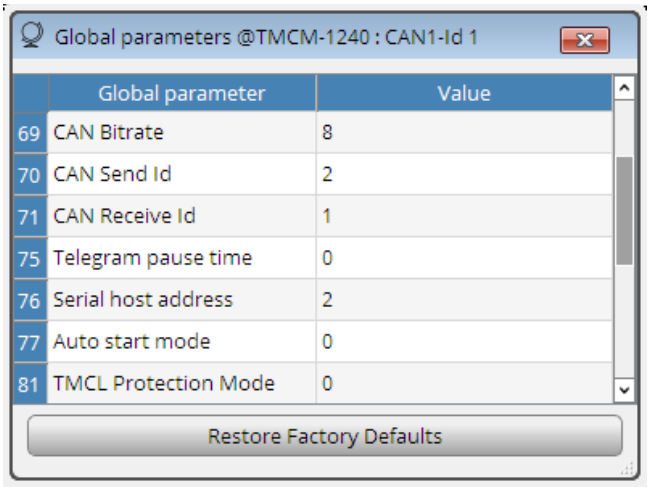


Figure 14: The Global Parameters Tool

All parameters that are supported by the connected module are shown in a list together with their current settings. To change a parameter, click on its value. The value can then be edited. The new value will be sent to the module after leaving the editor of this parameter. All these parameters are automatically stored permanently in the EEPROM of the module.

Clicking the *Restore Factory Defaults* button will completely reset the module. All global parameters will be restored to their factory default values, and also a TMCL program stored on the module will be erased.

4.4 The TMCL Creator

Using the TMCL Creator it is possible to develop stand-alone TMCL programs. These programs can be downloaded to a module and then run stand-alone on that module. Not all TRINAMIC modules support stand-alone TMCL programming. So the TMCL Creator is only provided for modules that support stand-alone TMCL programming.

The central part of the TMCL Creator is its main window with its own menu bar and the TMCL program editor. The TMCL program editor mainly provides the functionality of a standard text editor with built-in syntax highlighting for TMCL. Here, TMCL programs can be entered and modified. After a program has been entered it can be assembled, downloaded to the module and run on the module.

Chapter 8 explains the syntax of the TMCL™ programming language. Please see also the TMCL Firmware Manual of your module to learn more about using TMCL™ as a programming language.

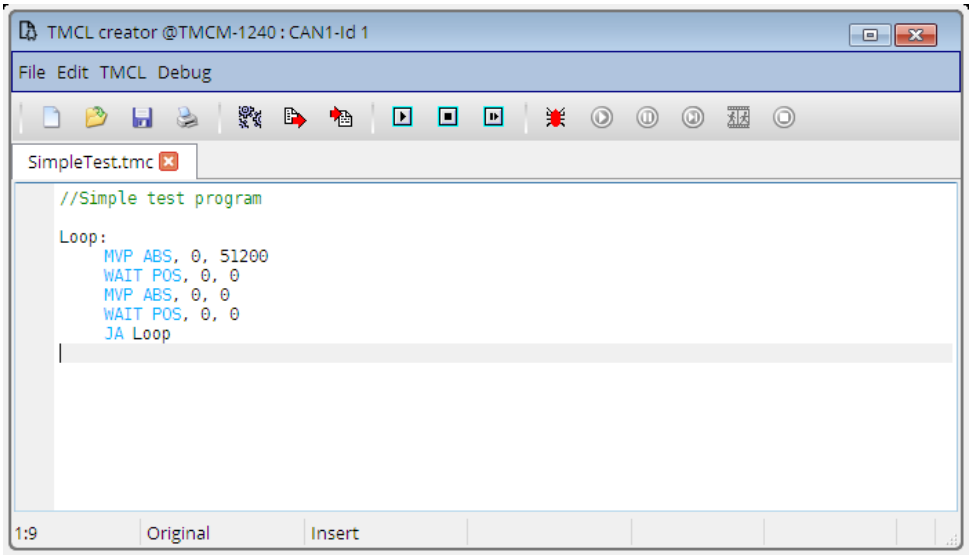


Figure 15: The TMCL Creator

4.4.1 The File Menu

The *File* menu of the TMCL Creator window provides the following functions:

- New: opens an empty editor tab for creating a new file.
- Open: loads a TMCL™ file into a new editor tab.
- Save: saves the content of the current editor tab to a file.
- Save as: saves the content of the current editor tab to a file using a new file name.
- Save all: saves all files that are currently loaded in the editor and that have been changed.
- Close: closes the currently open editor tab (same function as the close button of the editor tab).
- Print: print the content of the currently open editor tab.

4.4.2 The Edit Menu

The *Edit* menu provides standard text editor functions like *Undo*, *Redo*, *Cut*, *Copy*, *Paste* and a Find/Replace function. The *Settings* function opens a dialog where the text colors can be customized.

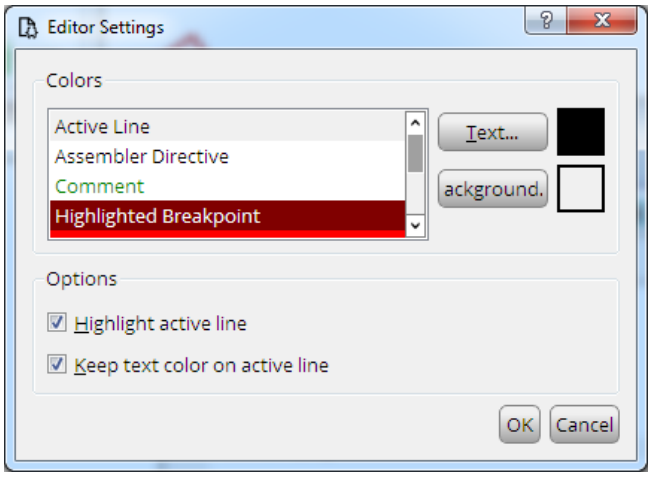


Figure 16: The Editor Settings Dialog

4.4.3 The TMCL Menu

The *TMCL* menu provides functions for assembling and downloading TMCL™ programs as well as for setting some options of the TMCL Creator. These are the following functions:

4.4.3.1 Assemble

Selecting the *Assemble* function in the TMCL™ menu assembles a file. If more than one file is open the currently selected file will be assembled. The progress of the assembly is displayed in a dialog (figure 17).

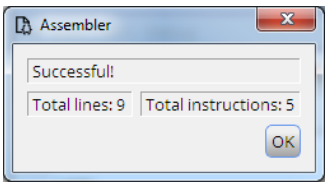


Figure 17: Progress of the Assembly

If an error occurs, the line containing the error will be highlighted and the assembly will be aborted. The error message will be displayed in the assembler progress dialog.

4.4.3.2 Download

After a TMCL™ program has been successfully assembled it can be downloaded into the module. To do this, select the *Download* function. The download progress is shown in a special dialog. Downloading can be aborted by clicking the Abort button.

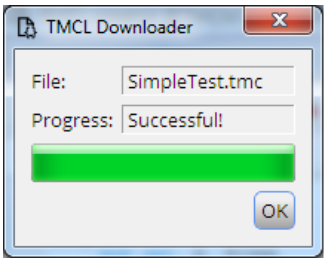


Figure 18: TMCL™ Program Download

4.4.3.3 Run

Select the *Run* function to run the TMCL™ program which is stored in the module. Execution then starts from the beginning of the program.

4.4.3.4 Stop

The *Stop* function stops a TMCL™ program that is running on the module. Please not that this function only stops the execution of the program. It does not stop the motors. The latter can be done by clicking the Emergency Stop button.

4.4.3.5 Continue

The *Continue* function continues the execution of a TMCL™ program that has been stopped using the *Stop* function from this menu.

4.4.3.6 Disassemble

The *Disassemble* function reads out the TMCL™ memory of a module and disassembles its contents. The result is written into a new editor page. This function allows checking the program which is currently stored in the TMCL™ program memory of a module.
The progress of downloading the program from the module and the disassembly is shown in a special dialog. It can be aborted by clicking the *Abort* button.

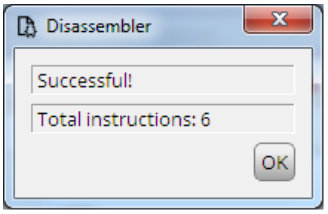


Figure 19: Program Disassembly

4.4.3.7 Options

Choosing the *Options* function displays the Asssembler/Debugger options dialog.

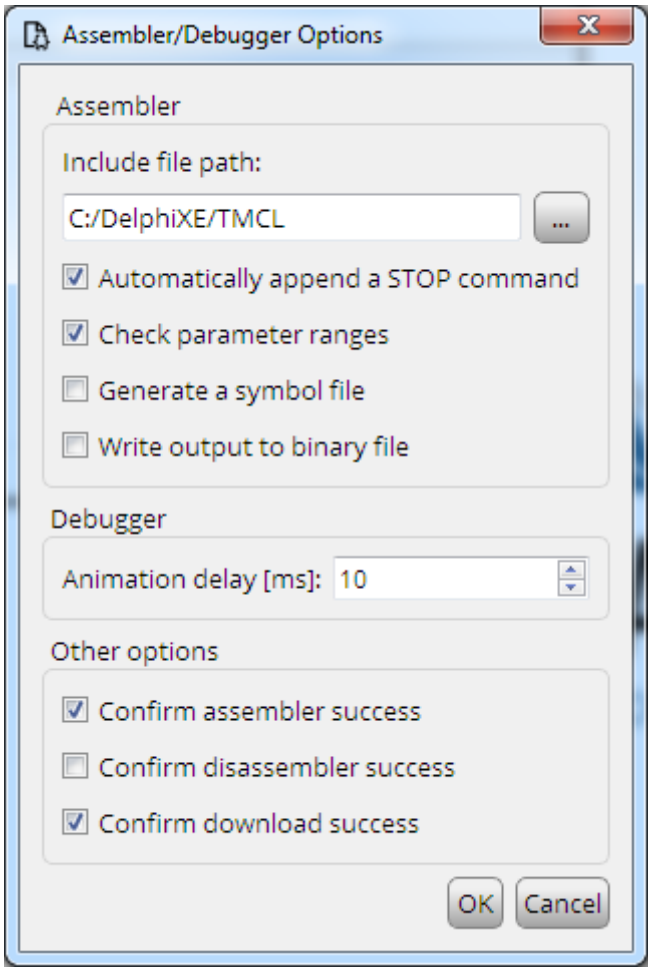


Figure 20: Assembler/Debugger Options

The following options can be set:

- *Include file path:* The path used to search for include files included by the #include directive.
- *Automatically append a STOP command:* If this option is selected the assembler automatically appends a Stop command at the end of every TMCL™ program (if the last command in the program is not already a STOP command).
- *Check parameter ranges:* All parameters used in TMCL™ commands will be checked against the parameter range entries in the module database of the TMCL-IDE. Violations of parameter ranges will lead to an error message. This option should normally be switched on.
- *Generate a symbol file:* The assembler generates a text file that contains the address of every label. This can be useful for starting the program from other addresses than 0. Please see also the sections about combining direct mode and standalone mode in the TMCL Firmware Manual of your module for more about this.
- *Write output to binary file:* The assembler writes its output also to a binary file. For every command eight bytes are written (the command with checksum, but without a device address).
- *Animation Delay:* Additional delay between two instructions when using the *Animate* function of the Debugger.

- *Confirm assembler success:* When set, the assembler dialog will stay open after successful TMCL program assembly. When not set, the assembler dialog will be closed automatically on success. Beginners should leave this option switched on (default setting), while experts might want to switch it off.
- *Confirm disassembler success:* When set, the disassembler dialog will stay open after successful TMCL program disassembly. When not set, the disassembler dialog will be closed automatically on success. Normally one can leave this option turned off (default setting).
- *Confirm download success:* When set, the download dialog will stay open after successful TMCL program download to a module. When not set, the download dialog will be closed automatically on success. Beginners should leave this option switched on (default setting), while experts might want to switch it off.

4.4.3.8 Auto Start Mode

Using this function the auto start mode of a module can be switched on or off. When auto start mode is switched on the TMCL™ program stored in the module will be executed automatically each time the supply power for the module is switched on.

4.4.4 The Debugger

To start the debugger either selecting *Debugger active* in the Debug menu or click the Debugger icon on the tool bar. The debugger then first checks if the program has been modified and in this case the program will automatically be assembled and downloaded again to be sure that the program in the editor is the same as the program stored on the module.

After the debugger has been successfully started, the debugger functions will be enabled and most other functions of the TMCL™ Creator will be disabled (in debug mode it is not possible to change the program in the editor). Other tools of the TMCL-IDE can still be used with care.
You can exit the debugger by clicking Debugger active in the Debug menu or the Debugger icon on the tool bar once again. Then, all debugger functions will be disabled and all other functions of the TMCL-IDE will be enabled again.

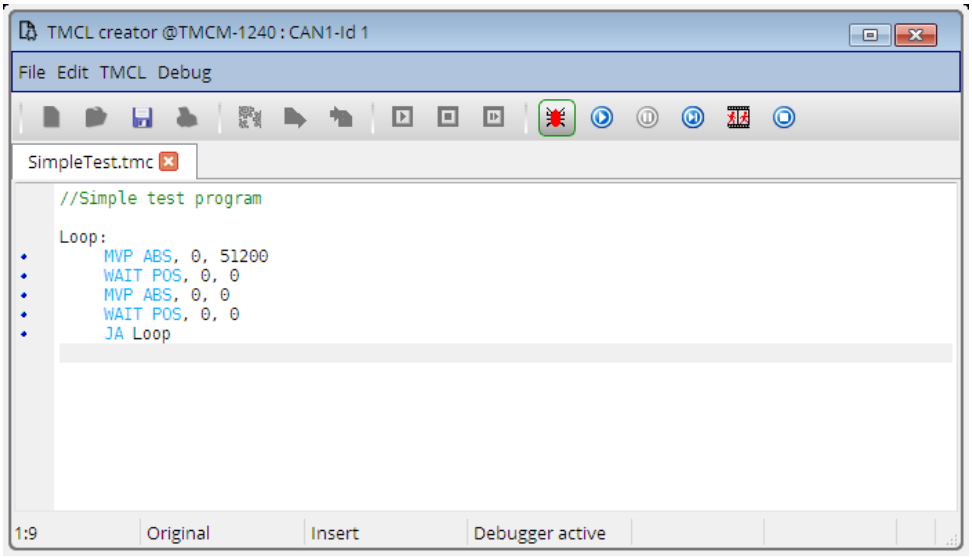


Figure 21: The TMCL Creator in Debug Mode

4.4.5 Breakpoints

Breakpoints can be set or removed by clicking the appropriate line on the left breakpoint bar of the editor. A blue bullet is displayed in every line where a breakpoint can be set. When a breakpoint is set a red bullet with a green tick will be displayed in that line.

After a program has been started either by the *Run* or by the *Animate* function of the debugger it will be stopped when a breakpoint is reached. It can be continued by clicking *Run*, *Animate* or *Step* again. It can also be reset by clicking *Stop & Reset* so that it can be restarted from the beginning again.

4.4.6 The *Run/Continue* Function

Choose the *Run/Continue* function from the Debug menu (or click the Run icon or press F9) to start the program or continue its execution. The program will be stopped either when its end is reached or when a breakpoint is reached or when the Pause function in the Debug menu or the Pause icon is selected. In the latter two cases the actual position in the program will be shown in the editor by a green arrow on the left side and the program execution can be continued by using the *Run/Continue* function, the *Step* function or the Animate function. The contents of the accumulator and the X register are also shown on the status bar while the program is paused.
If you wish to restart the program from the very beginning, select the *Stop/Reset* function before clicking *Run/Continue*.

4.4.7 The *Pause* Function

When a program is running (started by the *Run/Continue* or by the *Animate* function), the program execution can be interrupted at any time by selecting the Pause function from the Debug menu or by clicking the Pause icon on the tool bar or by pressing F2. Program execution can be continued by clicking *Run/-Continue*, *Animate* or *Step* again. While a program is paused the actual position in the program is shown by a green arrow in the editor. The contents of the accumulator and the X register are shown on the status bar of the TMCL Creator window.

4.4.8 The *Step* Function

Use this function for a step-by-step execution of the TMCL™ program. Every time the *Step* function is selected (by selecting *Step* in the Debug menu, clicking the *Step* icon on the tool bar or pressing F7) the next command in the TMCL™ program will be executed. The actual position is shown by a green arrow in the editor. The contents of the accumulator and the X register are also shown on the status bar.

4.4.9 The *Animate* Function

This function automatically executes the TMCL™ program step-by-step, so that the flow of the program can be seen. The actual position in the program is shown and updated after every command, and the contents of the accumulator and the X register are also shown on the status bar and updated after every command.

The program will be paused when running into a breakpoint or when the Pause function is selected. Program execution can be continued either by the *Run/Continue* function, the *Animate* function or the *Step* function.

4.4.10 The *Stop & Reset* Function

Selecting *Stop & Reset* from the Debug menu (or clicking the *Stop & Reset* button or pressing Ctrl+F2) stops the program execution immediately and resets the program. This means that starting the program again

using *Run/Continue*, *Animate* or *Step* will start the program from the beginning.

This function can also be used when the program execution is paused (by a breakpoint or by the *Pause* function) to reset it and to ensure that the program can be started again from the beginning.

4.5 The Parameter Calculator

The Parameter Calculator is a utility for converting between different units of motion parameters like the velocity and the acceleration. There are two different versions of the Parameter Calculator: one for modules that are based on the TMC429 motion controller (older modules like the TMCM-140, TMCM-343, TMCM-351, TMCM-1110, TMCM-1140, TMCM-1141, TMCM-1160, TMCM-1161, TMCM-1180, TMCM-1181, TMCM-3110 and TMCM-6110) where internal units are used for velocity and acceleration parameters and one for all other modules where velocity and acceleration values are given as pps (pulses per second) units.

The TMCL-IDE will always automatically show the version of the Parameter Calculator that fits for the connected module.

4.5.1 TMC429 based Modules

The Parameter Calculator for TMC429 based modules helps to convert between physical units for velocity and acceleration and the internal units used by those modules that are based on the TMC429 motion controller.

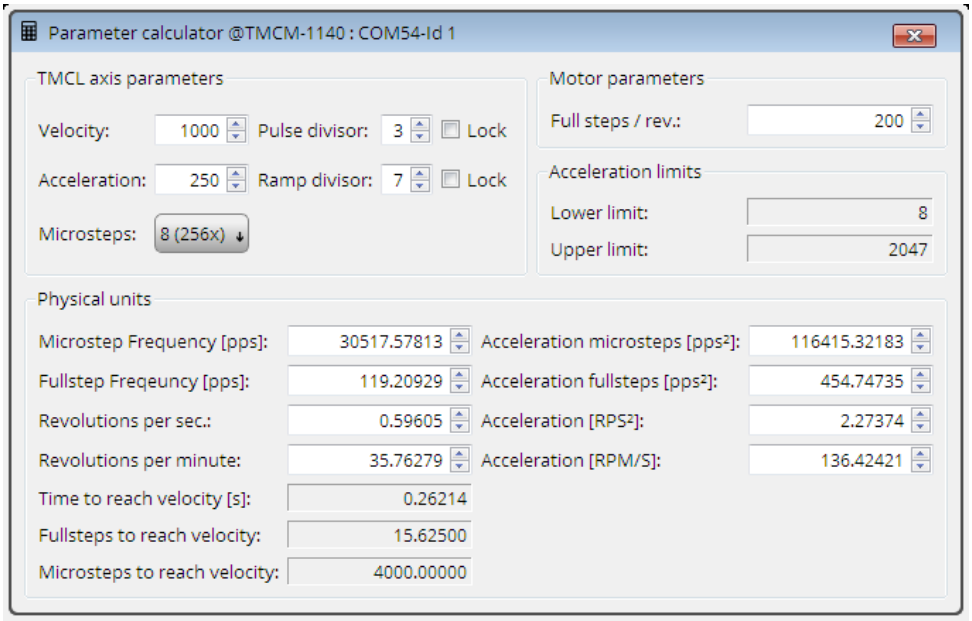


Figure 22: The Parameter Calculator for TMC429 based Modules

It is capable of converting from internal units to physical units as well as from physical units to internal units. Just change the value that you would like to convert and see how the other values get changed then. When changing a physical value the Parameter Calculator normally tries to find the best combination of pulse divisor and/or ramp divisor and velocity and/or acceleration to achieve this value. This results in the fact that pulse divisor and/or ramp divisor also get changed. If you do not wish the pulse divisor and/or the ramp divisor to get changed automatically check the *Lock* checkbox beside the pulse divisor and/or

the ramp divisor.

With some parameter combinations there are restrictions concerning the acceleration value. Such restrictions are also shown by the Parameter Calculator.

Please see the TMCL Firmware Manual of your module to learn more about the relations between the internal units of the module and the physical units for velocity and acceleration.

4.5.2 Modules that use pps Units

The Parameter Calculator for modules that use pps units just converts between pulses per second and units like rounds per minute or rounds per second.

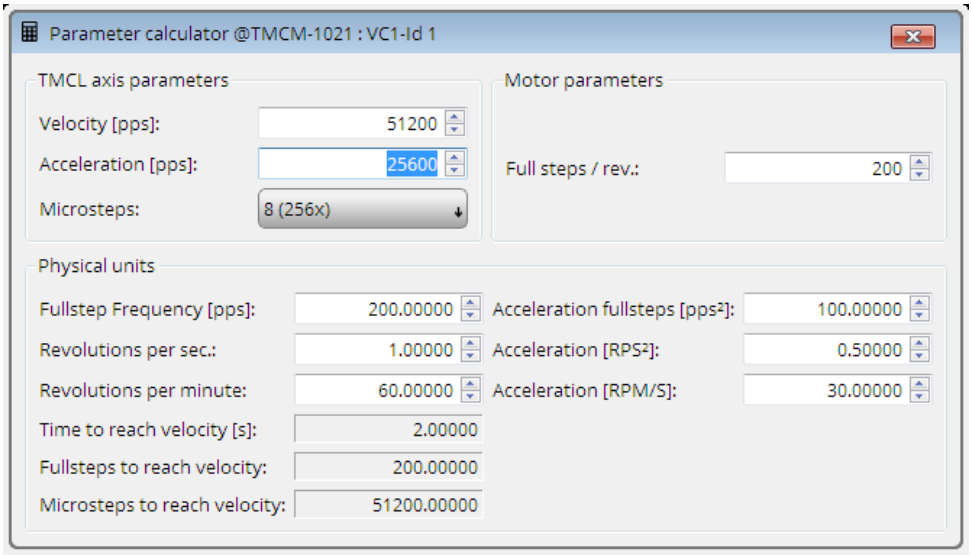


Figure 23: The Parameter Calculator

The Parameter Calculator can convert from pps units to the other units as well as from other units back to pps units. Just change the value that you would like to convert and see how all other parameters that depend on this value are automatically re-calculated then.

Please see also the TMCL Firmware Manual of your module to learn more about the relations between the different physical units.

4.6 The User Variable Display

The User Variable Display is a tool for displaying the contents of selected TMCL™ user variables with automatic periodic update. This allows permanent monitoring of TMCL™ user variables (which is also possible while a TMCL™ program is running on the module). It can be used for example to help debugging a TMCL™ program. This tool is provided for all modules that support stand-alone TMCL™ programming.

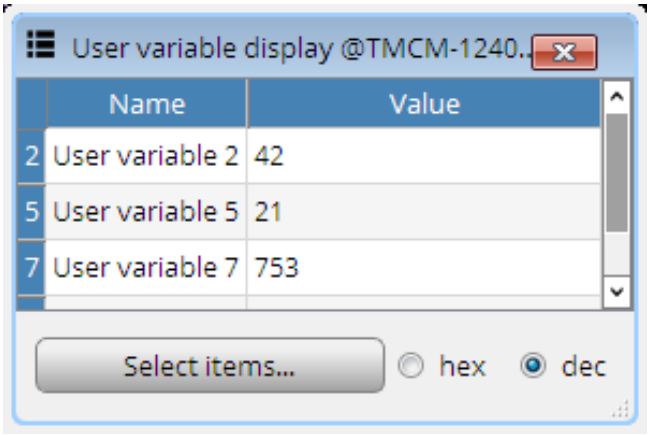


Figure 24: The User Variable Display

The contents of the displayed user variables can be presented in decimal or in hexadecimal form. This can be selected using the buttons at the bottom of the window. The displayed user variable contents are updated periodically, as fast as set up in the timer settings of the module connection. Clicking the *Select* button will show a dialog for selecting which variables are to be displayed.

4.7 The Settings Tool (Stepper Motor)

The Settings tool allows changing basic axis-specific settings. It is provided once for each axis of the connected module. Depending on the module type this tool looks different, as different modules might offer different axis settings.

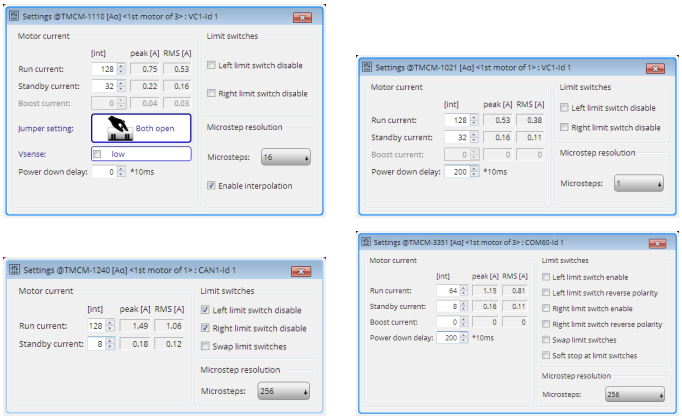


Figure 25: Different Appearances of the Settings Tool

The main settings that can be made here are the motor run current and standby current, the functionality of the limit switches and the microstep resolution. Most changes made here will become effective immediately. Each setting corresponds to one axis parameter in TMCL™. The correponding parameter numbers are shown when hovering over a setting with the mouse pointer. Please see the TMCL Firmware Manual of your module to learn more about each parameter.

4.8 The Closed Loop Tool

The closed-loop tool is provided for each axis of a closed-loop stepper motor module. With the help of the closed-loop tool it is possible to activate and to tune the closed-loop feature of such modules.

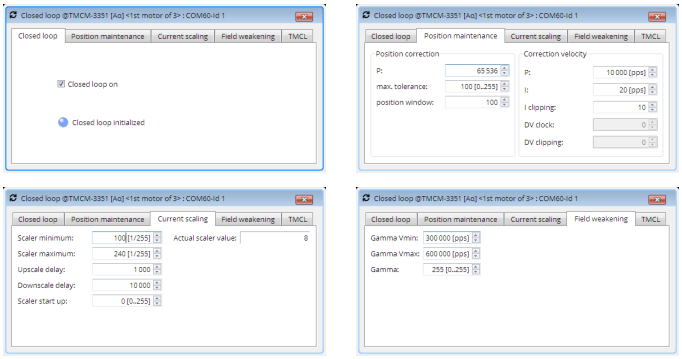


Figure 26: The Different Tabs of the Closed Loop Tool

The checkbox on the first tab switches the closed loop function of the module on and off. Hovering with the pointer of one of the closed-loop parameters shows the axis parameter number. Please see the TMCL Firmware Manual to learn more about the close loop parameters.

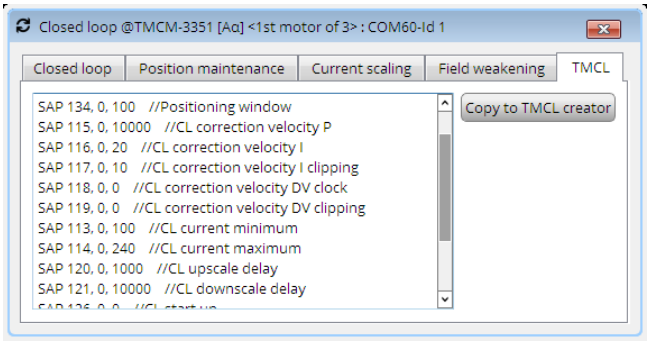


Figure 27: The TMCL Tab of the Closed Loop Tool

On the *TMCL* tab all settings are shown as TMCL commands. Click the *Copy to TMCL Creator* button to copy all these commands to the editor of the TMCL Creator.

4.9 The Chopper Settings

The Chopper Settings tool is provided for all modules that also support the StealthChop™ mode. All settings that are necessary to use the StealthChop™mode can be made in an interactive way using this tool. After having made all necessary settings, the TMCL™ commands that are necessary to set up StealthChop™ can directly be copied to the TMCL Creator.

4.9.1 The *Thresholds* Tab

On the *Thresholds* tab the threshold speeds for switching between different chopper schemes can be set.

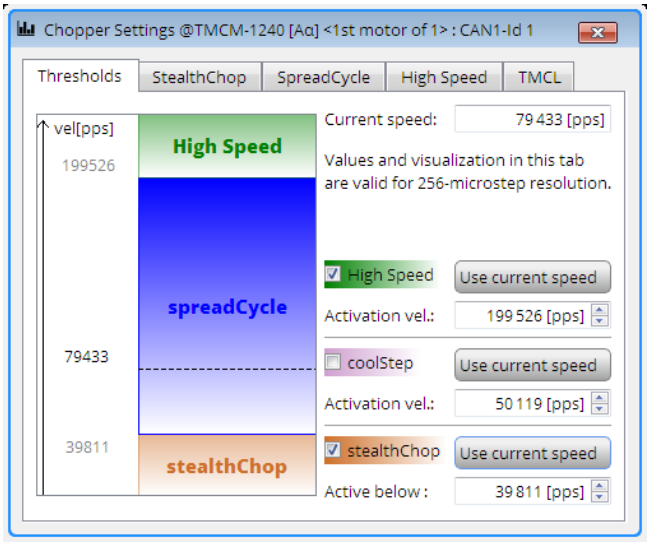


Figure 28: Chopper Settings — Thresholds

To use this tool, first run the motor at the desired velocity using the Velocity Mode tool. Now, set up the threshold speeds for switching between the different chopper modes. Click the *Use current speed* button to use the actual speed as a threshold speed. The following settings can be made:

- *High speed*: it is possible to switch to fullstep mode and/or to a different chopper scheme when the speed is above the activation velocity for this mode. Check the *High speed* check box to activate this feature. The features that are to be activated can be choosen on the *High Speed* tab.
- *CoolStep™*: above this speed, CoolStep™ can be activated. Use the CoolStep™ tool to further parametrize CoolStep™. Check the *coolStep* checkbox to activate this threshold speed.
- *StealthChop™*: if the speed drops below this speed the StealthChop™ scheme can be activated to make the motor run quitely. Check the *StealthChop™* checkbox to activate this feature.

Try running the motor at different speeds below and above each threshold to be sure that the motor runs stable with these settings.

4.9.2 The *StealthChop™* Tab

Use the settings on the StealthChop™ tab to further parametrize the StealthChop™ chopper scheme. This is needed to adapt StealthChop™ to different motors. In many cases the factory default values do not need to be changed much.

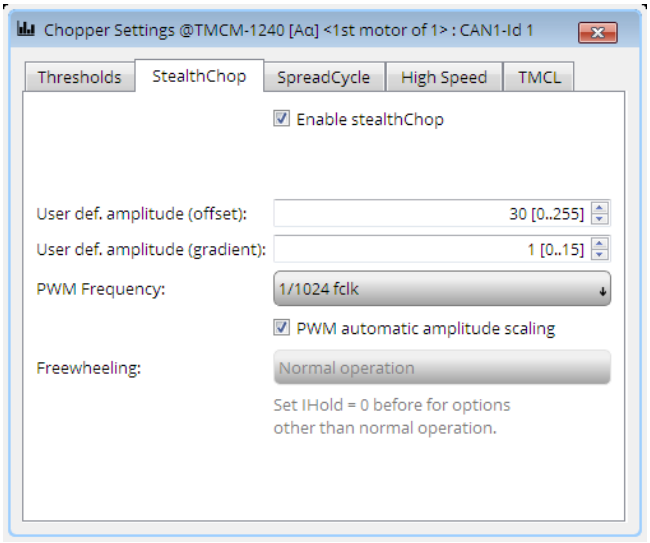


Figure 29: Chopper Settings — StealthChop™

The following settings can be made here:

- *Amplitude offset:* Maximum PWM amplitude when switching to StealthChop™ mode. Do not set too low. Values above 64 recommended with most modules.
- *Amplitude gradient:* Velocity dependent gradient for the PWM amplitude.
- *PWM frequency:* the frequency of the chopper PWM.
- *Freewheeling:* in StealthChop™mode, a freewheeling mode can be chosen which becomes active when the motor is stopped and the standby current (axis parameter #7) is set to 0.

4.9.3 The *SpreadCycle*™ Tab

On the SpreadCycle™ tab, all settings that are specific to the default chopper scheme can be made.

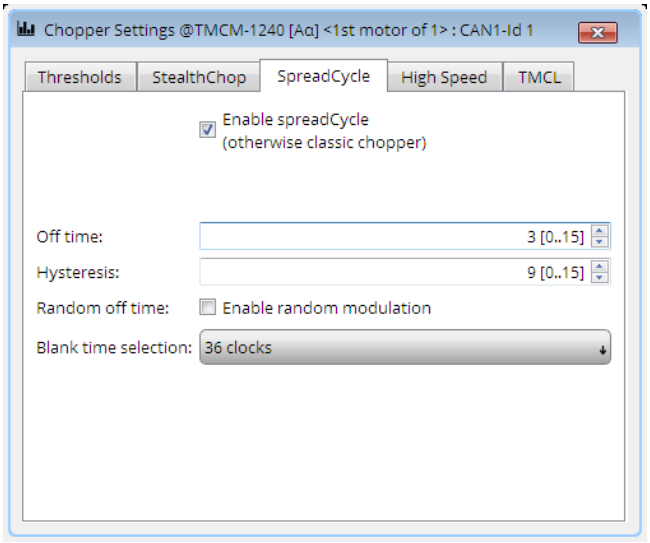


Figure 30: Chopper Settings — SpreadCycle™

For most cases, leave these values at their default settings. Please see the TMCL Firmware Manual of your module for more information about these settings.

4.9.4 The *High Speed* Tab

On the high speed tab it is possible to choose options that become active when the actual speed becomes greater than the high speed threshold value.

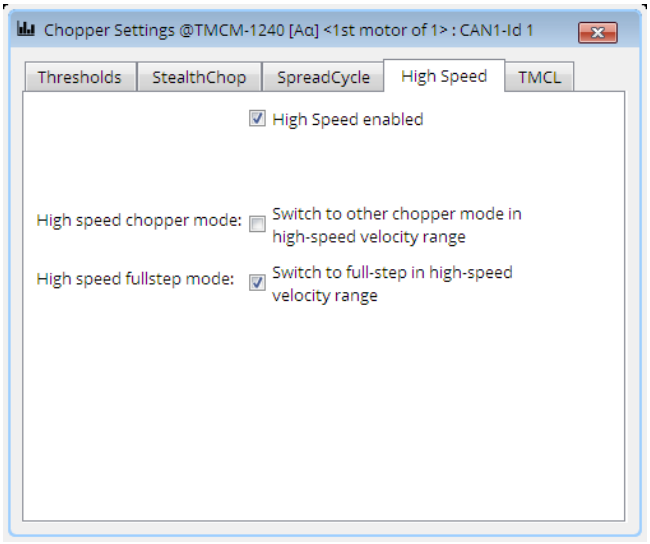


Figure 31: Chopper Settings — High Speed

The following options can be chosen here:

- *High speed chopper mode*: Switch to a different chopper mode when this option is checked and the speed is above the high speed threshold value.

- *High speed fullstep mode*: Switch to fullstep mode when this option is checked and the speed is above the high speed threshold value.

4.9.5 The *TMCL™* Tab

On the TMCL tab, all settings made using the Chopper Settings tool are shown as TMCL™ commands.

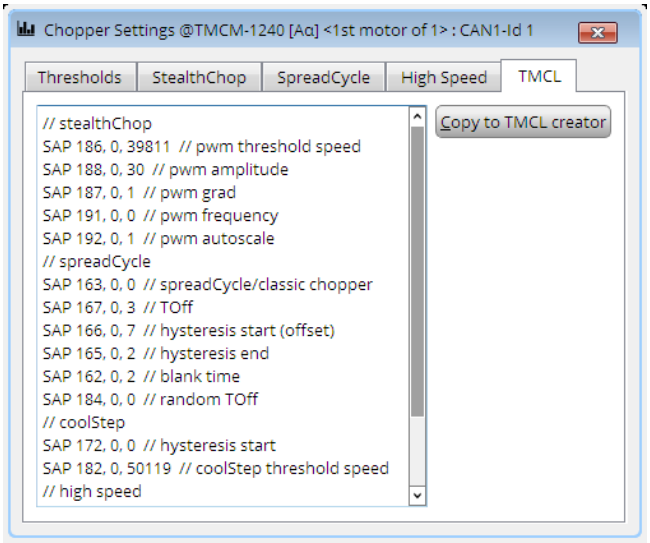


Figure 32: Chopper Settings — TMCL™

Click the *Copy to TMCL™ Creator* button to copy all the commands to the TMCL Creator. You can then easily use these settings in your TMCL™ program.

4.10 The Velocity Graph

The Velocity Graph is a tool for graphically displaying axis parameters, user variables and the states of input/output ports of the module. It is pre-configured for displaying the actual velocity and the target velocity, but it can be used for graphically monitoring also anything else. This can help for example when tuning ramp parameters or also when debugging a TMCL™ program that is running on the module. The data will be updated as fast as set up in the timer setting of the module connection.

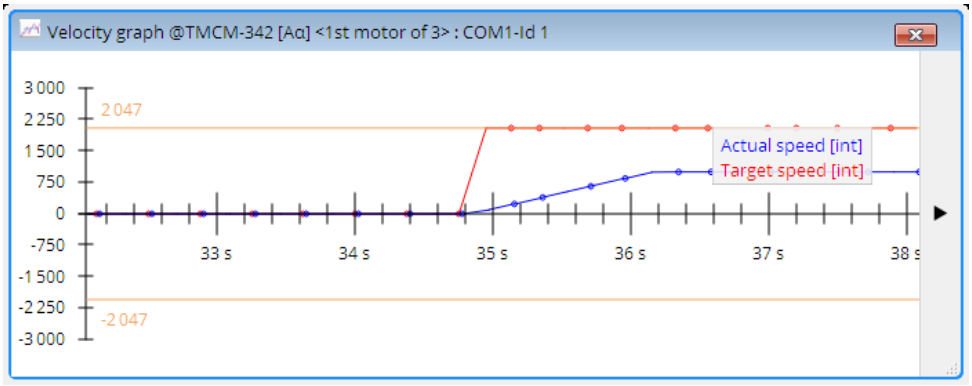


Figure 33: The Velocity Graph

Click the right mouse button over the graph window to display the context menu of the graph display window. The context menu offers some options for changing the appearance of the graph and for selecting the items that are to be displayed:

- *Bold points:* Selects if each data point of the graph shall be displayed as a bold point or just as a small point.
- *Points only:* Selects if the data points are to be linked by lines or just the data points are to be shown, not linked by lines.
- *Show min/max marker:* Selects if marker lines for the actual maximum and minimum values are to be shown.
- *Normalize data:* Selects if the data is to be normalized to the maximum and minimum scale values or if just the visible part of the graph is to be shown.
- *Select items:* Displays a dialog that allows selecting which parameters are to be displayed in the graph (please see section 4.10.1).
- *Change order:* Shows a dialog that allows changing the order of the parameters displayed in the graph.

4.10.1 Selecting Parameters to be displayed

The parameter selection dialog allows selecting the parameters that are to be displayed in the graph window.

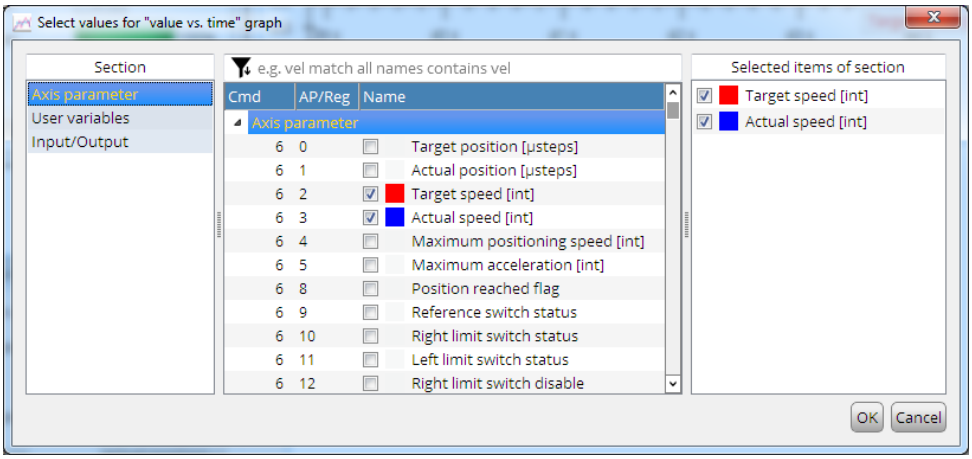


Figure 34: The Parameter Selection Dialog

It is grouped into three sections:

- *Section:* Here you can select between axis parameters, user variables and input/output ports.
- *Parameter list:* Pick parameters that are to be displayed from this list or deselect parameters that are no longer to be displayed here.
- *Selected items:* This shows which items are selected from each section. Right-click on a colored box to change the display color of an item.

4.10.2 Other Options

Click the arrow button on the right hand side of the graph window to select scaling options.

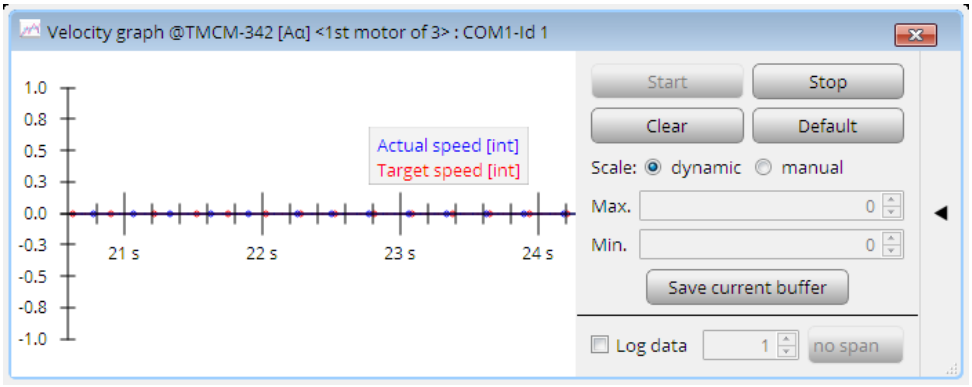


Figure 35: Graph Scaling Options

Here the following options can be selected:

- *Stop:* The *Stop* button stops the graph so that it can be viewed.
- *Start:* Clicking the *Start* button re-starts refreshing of the graph.
- *Clear:* This button clears the data buffer of the graph.
- *Default:* Sets all graph scaling parameters back to default values.

- *Max.* and *Min.:* Maximum and minimum values of the y scale of the graph.
- *Save current buffer:* saves the currently displayed data to a CSV file.
- *Log:* Logging can be enabled to continuously save the graph data to a CSV file. Either every data point can be logged (when *no span* is selected), or the data can be logged in configurable spans of seconds or minutes.

Click the arrow button on the right hand side of the window again to hide these options.

4.11 The Position Graph

The Position Graph is a tool for graphically displaying axis parameters. It works exactly the same way as the Velocity Graph (please see there), with the only exception that it is pre-configured for showing the actual position and the target position.

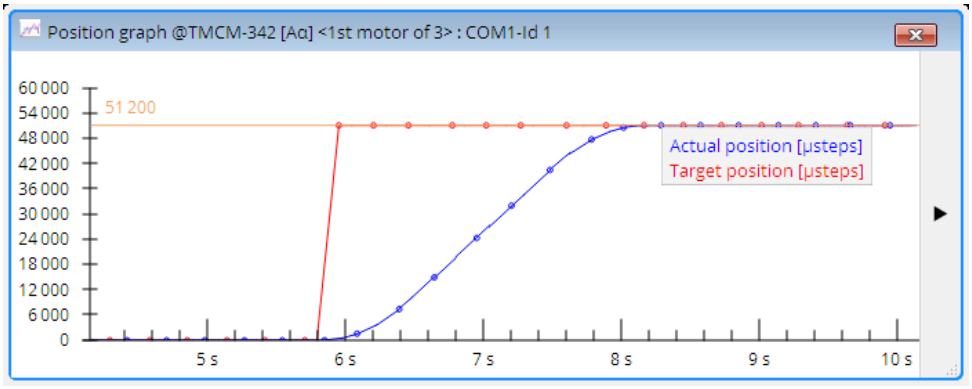


Figure 36: The Position Graph

4.12 The Torque Graph

The Torque Graph is a tool for graphically displaying axis parameters. It works exactly the same way as the Velocity Graph (please see there), with the only exception that it is pre-configured for showing the actual torque and the target torque. It is provided for BLDC motor modules only.

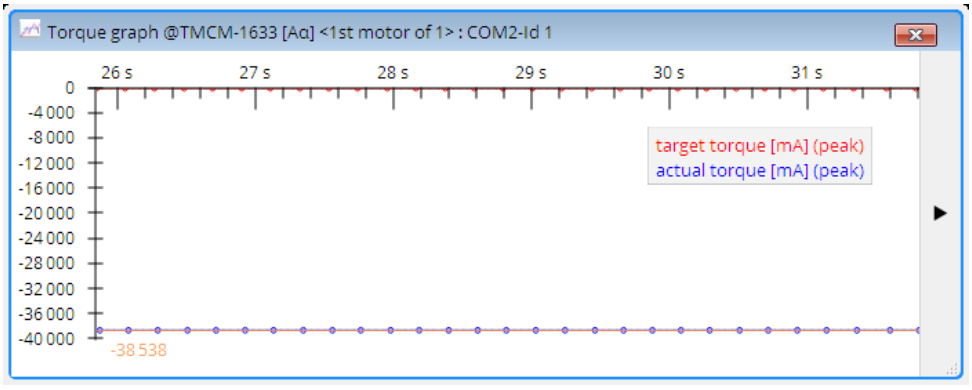


Figure 37: The Torque Graph

4.13 The Velocity Mode

The velocity mode is a tool that allows running a motor at a selectable velocioty. Depending on the capabilities of the module the velocity mode window looks different. Some modules are using different velocity units or have different options for the velocity mode. The different versions of the velocity mode all have the same basic functions.

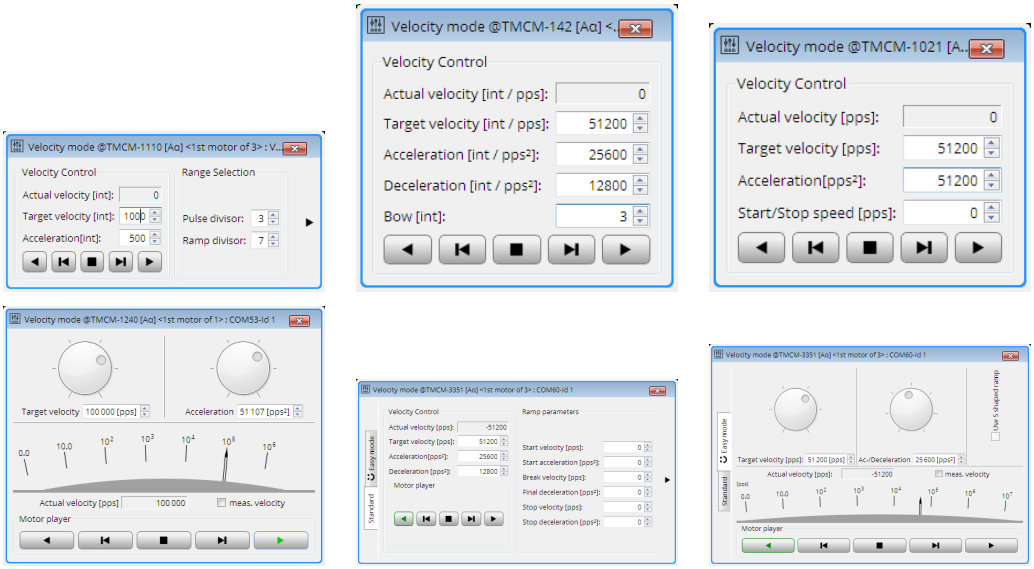


Figure 38: Different Appearances of the Velocity Mode Tool

The different versions of the velocity mode tool always provide the same buttons for moving the motor:

- ◄ The motor continous running in negative direction until clicking the stop button or a different button for running the motor.
- ⏮ The motor runs in negative direction as long as the button is being pressed. When releasing the button the motor will be stopped.
- ■ Stop: clicking this button will stop the motor.
- ⏭ The motor runs in positive direction as long as the button is being pressed. When releasing the button the motor will be stopped.
- ► The motor continous running in positive direction until clicking the stop button or a different button for running the motor.

Depending of the capabilities of the module the units are either internal units of the module ([int]) or microsteps per second (pulses per second, [pps]) or revolutions per minute ([prm]). Some versions of the velocity mode tool also provide an *Easy Mode*, where the most important parameters can be set using dials or sliders.

4.14 The Position Mode

The position mode is a tool that allows driving a motor to selectable positions. Depending on the capabilities of the module the position mode window looks different. Some modules are using different velocity units or have different options for the position mode. The different versions of the position mode all have

the same basic functions. The positions are always given as microsteps.

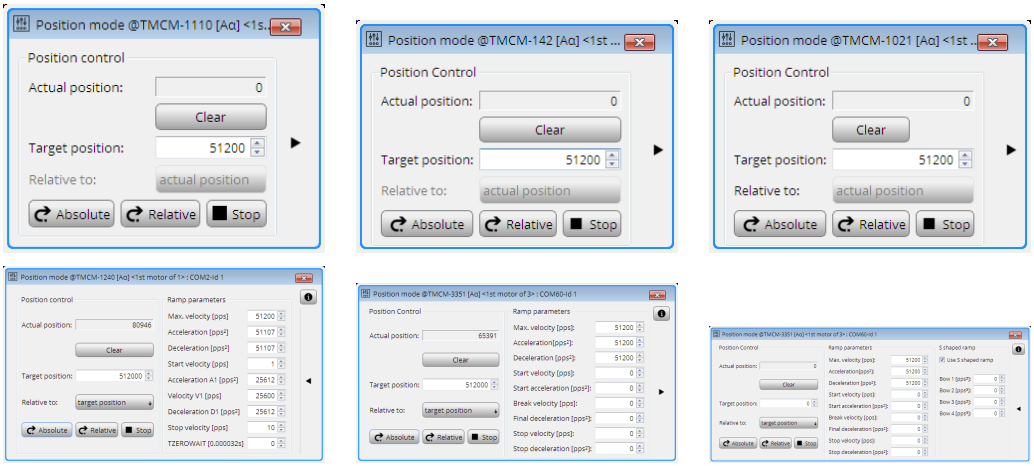


Figure 39: Different Appearances of the Position Mode Tool

The basic functions are the same in all versions of the position mode tool:

- *Actual Position:* Shows the actual position given in microsteps.
- *Target Position:* Enter the target position here, given in microsteps.
- *Absolute:* Click this button to move to the given target position, as absolute position. This generates an MVP ABS command.
- *Relative:* Click here to use the given target position as a relative offset to the actual position. This generates an MVP REL command.
- *Relative to:* Depending on the module and its firmware version it is possible to select the base used for a relative move. This can either be the actual position, the last target position or (on some closed loop modules) the encoder position.
- Click the arrow on the right edge of the window to show or hide the ramp parameters.

4.15 The Torque Mode (Stepper Motor)

With some closed-loop stepper motor modules it is also possible to drive the stepper motor using torque mode. This is only possible when closed-loop mode is active.

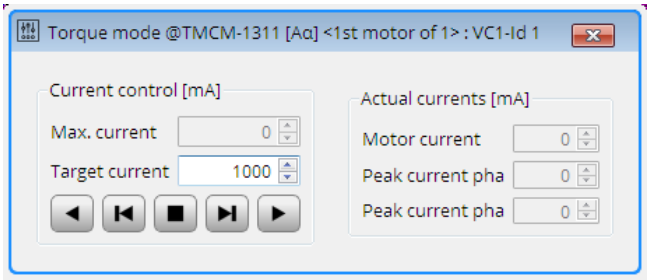


Figure 40: The Torque Mode for Stepper Motors

Please see the TMCL Firmware Manual of your module for more about this.

4.16 The Torque Mode (BLDC Motor)

With BLDC motor modules it is also possible to drive the motor using torque mode. This can be controlled using the Torque Mode tool.

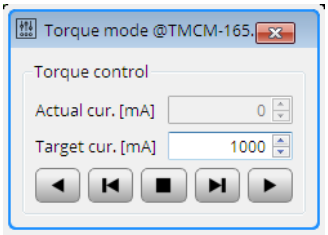


Figure 41: The Torque Mode for BLDC Motors

The desired motor current can be entered, and the actual motor current is also shown. The following functions are provided:

- ◄ The motor continous running in negative direction until clicking the stop button or a different button for running the motor.
- ⏮ The motor runs in negative direction as long as the button is being pressed. When releasing the button the motor will be stopped.
- ■ Stop: clicking this button will stop the motor.
- ⏭ The motor runs in positive direction as long as the button is being pressed. When releasing the button the motor will be stopped.
- ► The motor continous running in positive direction until clicking the stop button or a different button for running the motor.

4.17 The Axis Parameter Display

The Axis Parameter Display is a tool for displaying the values of selected axis parameters with automatic periodic update. This allows permanent monitoring of axis parameters (which is also possible while a TMCL™ program is running on the module). It can be used for example to help debugging a TMCL™ program. This tool is provided for all modules that support axis parameters (which most of the TRINAMIC modules do).

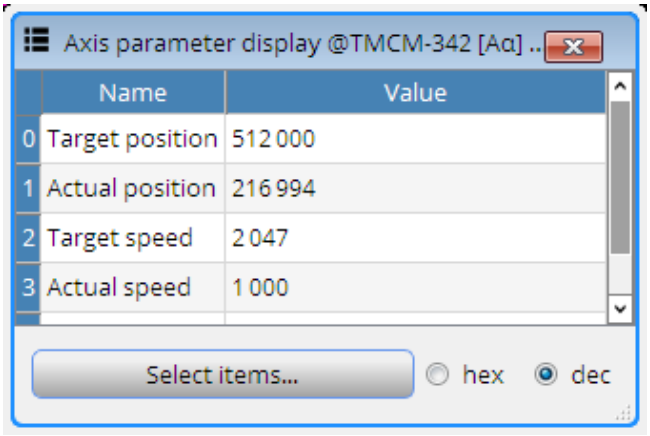


Figure 42: The Axis Parameter Display

The contents of the displayed axis parameters can be presented in decimal or in hexadecimal form. This can be selected using the buttons at the bottom of the window. The displayed axis parameter values are updated periodically, as fast as set up in the timer settings of the module connection. Clicking the *Select* button will show a dialog for selecting which axis parameters are to be displayed.

4.18 The StallGuard™ Adjuster

The StallGuard™ adjusting tool is provided for modules that are equipped with the StallGuard™ (not the StallGuard2™) feature. These are older modules like the TMCM-140, the TMCM-343 or the TMCM-351. This tool helps to find the necessary motor parameters when StallGuard™ is to be used. Figure 43 shows the StallGuard™ Adjuster tool.

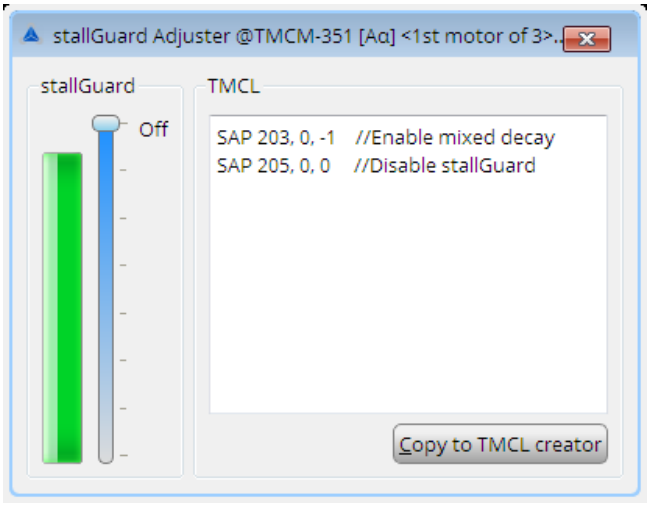


Figure 43: The StallGuard™ Adjuster

Use this tool as follows:

1. Use the Velocity Mode tool to run the motor at a specific velocity.

2. The green bar in the StallGuard™ Adjuster displays the actual load value. Use the slider to set the StallGuard™ threshold value. The motor will be stopped automatically if the load value reaches this value.

3. The necesseray settings are also displayed as TMCL commands in the TMCL area of the StallGuard™ Adjuster window. Click the *Copy to TMCL Creator* button to copy the commands to the TMCL Creator.

4.19 The StallGuard™ Profiler

The StallGuard™ profiler tool is provided for modules that are equipped with the StallGuard™ (not the StallGuard2™) feature. These are older modules like the TMCM-140, the TMCM-343 or the TMCM-351. This tool helps to find the best settings for using stall detection. It scans through given velocities and shows which velocities are the best ones to be used with StallGuard™.

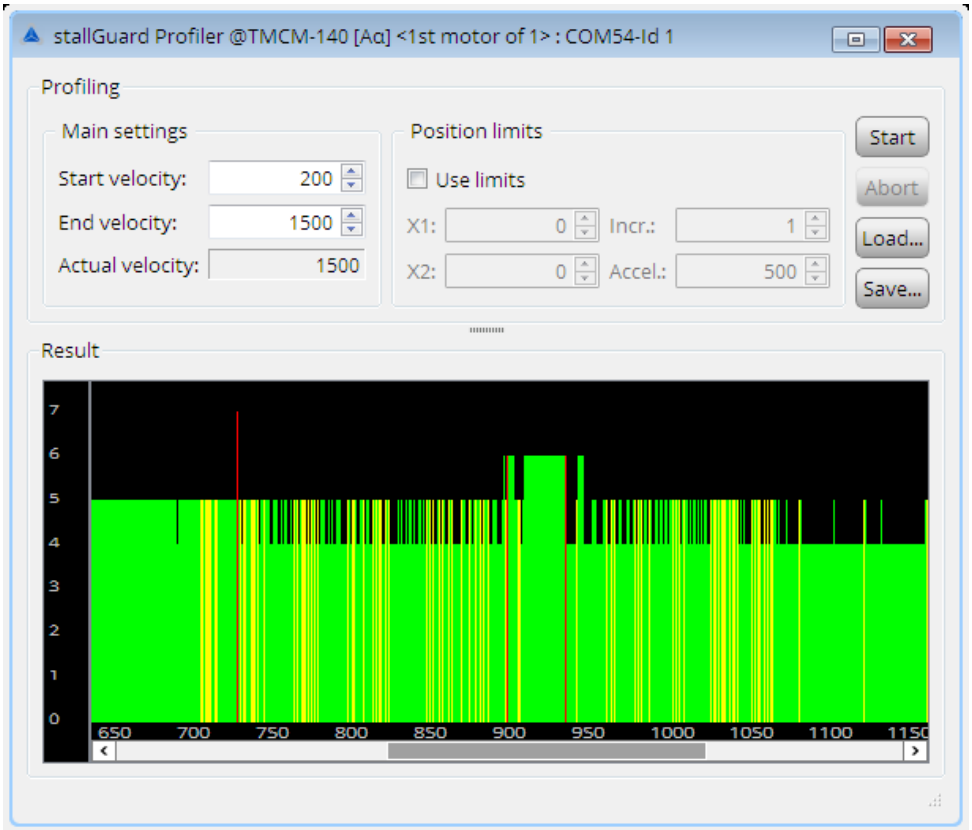


Figure 44: The StallGuard™ Profiler

Here is how to use the StallGuard™ profiler:

1. First, 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.

2. After these parameters have been entered, 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 will be measured ten times.

3. The actual velocity value shows the velocity that is currently being tested and so tells about the progress of the profile recording.

4. The recording of the profile can also be aborted by clicking the Abort button.
5. It is also possible to limit the motion between two positions. To use this feature, check the Use Limits checkbox and enter position limits. This is useful for example when recording the StallGuard™ profile of a linear drive.
6. A StallGuard™ profile that has been recorded can be saved using the Save button and loaded for viewing it again using the Load button.

4.19.1 The Result of the StallGuard™ Profiler

The result is shown graphically in the StallGuard™ profiler window. After the profile recording has finished you can scroll through the profile graph 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 color 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.

Three colors are used:

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

4.19.2 Interpreting the Result

In order to make effective use of the StallGuard™ feature one should choose a velocity where the load value is as low as possible and where the color 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 using yellow color can also be used, but with care as they might cause problems (maybe the motor will stop even when not being stalled).

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

It is unlikely that exactly the same result is produced when recording a profile with the same parameters again. Always two or more profiles should be recorded and compared against each other in order to get a stable result.

4.20 The CoolStep™ & StallGuard2™ Tool

The CoolStep™ & StallGuard2™ tool is provided for modules that are equipped with the CoolStep™ and the StallGuard2™ feature. These are most newer TRINAMIC stepper motor control modules. It helps to find the necessary motor parameter adjustments when StallGuard2™ and CoolStep™ are to be used.

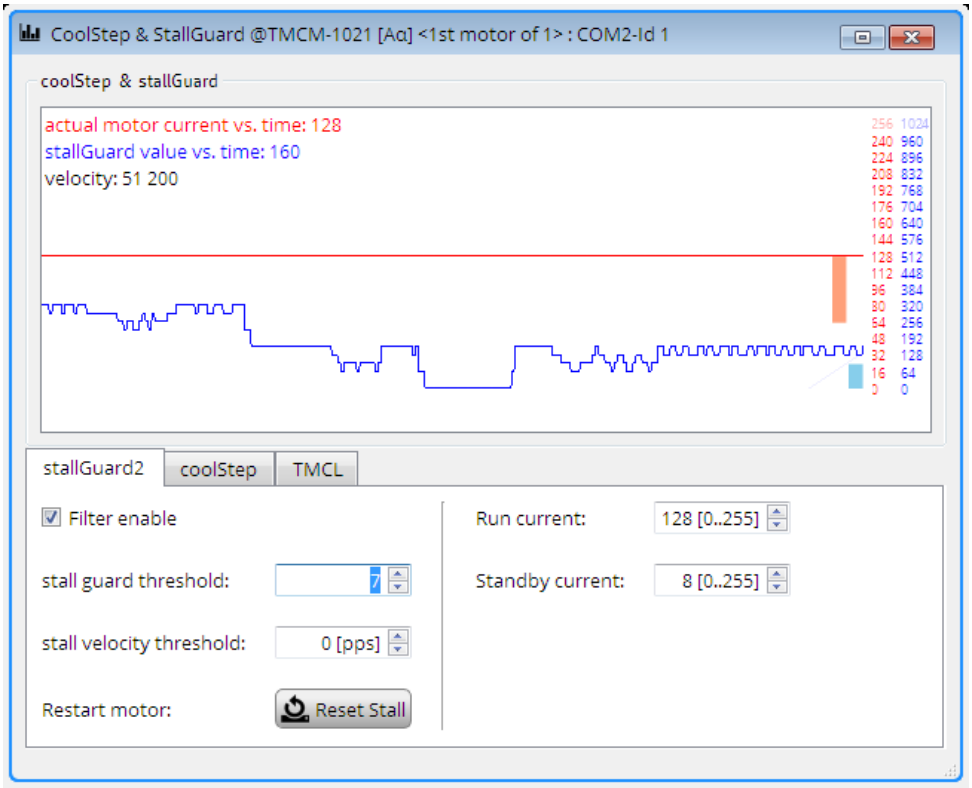


Figure 45: The CoolStep™ & StallGuard2™ Tool

This tool consists of a graph that shows the load value (blue line) and the motor current value (red line) and three tabs which concern different aspects of these features:

- StallGuard2™
- CoolStep™
- TMCL™

The CoolStep™ & StallGuard2™ tool provides an interactive way for adjusting the necessary settings. All settings that can be made with on the StallGuard2™ tab and on the CoolStep™ tab are immediately represented as TMCL™commands on the TMCL™ code tab.

4.20.1 The StallGuard2™ Tab

The StallGuard2™ tab contains the following parameters:

- Filter enable (SAP 173): this parameter enables the StallGuard2™ filter for more precision of the measurement. If enabled, the measurement frequency will be reduced to one measurement per four fullsteps. In most cases it is expedient to enable the filter before using CoolStep™.
- stallGuard threshold (SAP 174): this value controls the threshold level for stall detection and sets the optimum measurement range for the load value readout. A lower value will result in a higher sensitivity. Higher values will cause StallGuard2™ to be less sensitive.
- stall velocity threshold (SAP 181, also called stop-on-stall velocity threshold): the motor will be stopped automatically if a stall has been detected and the actual velocity is higher than this value. Setting this value to zero turns off this feature.

- Reset Stall: with some modules it is necessary to click this button to be able to re-start the motor after it has been stopped by the stop-on-stall feature.
- Run Current (SAP 6): the run current of the motor can also be set here.
- Standby Current (SAP 7): the standby current of the motor can also be set using this tool.

Please see the TMCL Firmware Manual of your module for further details about these parameters.

To set up the StallGuard2™ parameters, first run the motor at the desired velocity using the Velocity Mode tool. Now, set the stallGuard threshold value in such a way that the blue line reaches zero when the motor is being obstructed (stalled). The blue line should be as far away from zero as possible when the motor is running normally. Always try different velocities. The StallGuard2™feature does not work the same with all velocities.

4.20.2 The CoolStep™ Tab

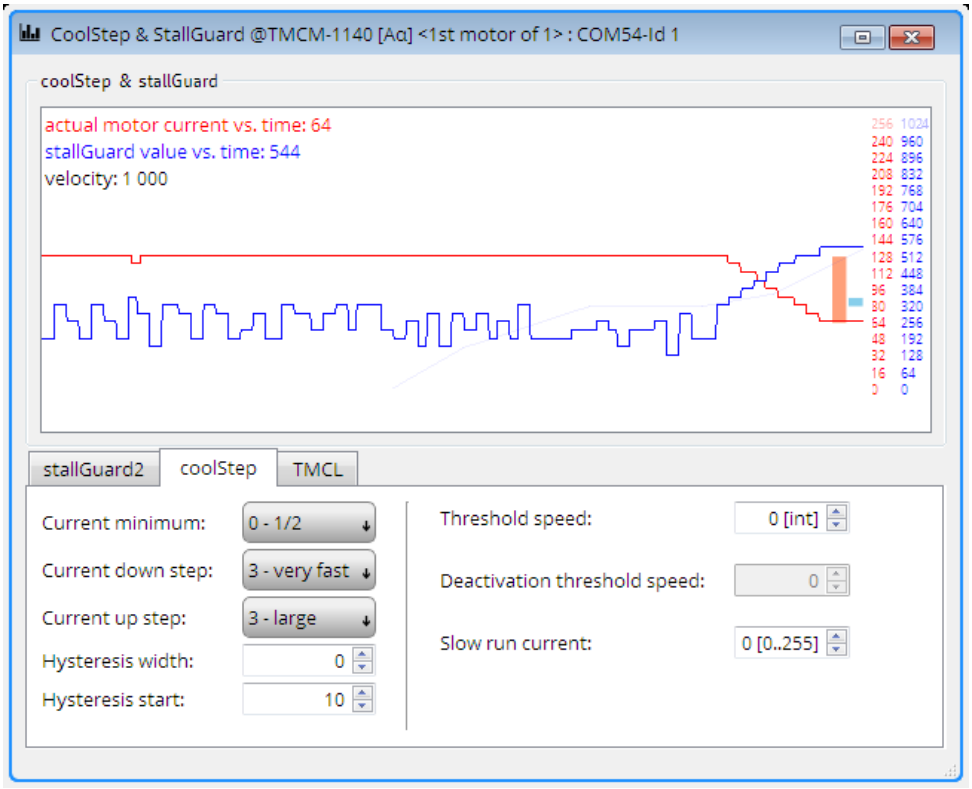


Figure 46: The CoolStep™ Tab

The CoolStep™ tab represents all settings that are necessary for using the CoolStep™ feature. These are mainly the following parameters:

- Current minimum (SAP 168): this can be 1/2 or 1/4 of the run current.
- Current down step (SAP 169): determines how fast the current is to be lowered when the load value is above uppper end of the CoolStep™ hysteresis.
- Current up step (SAP 171): determines how fast the current is to be raised when the load value is below the lower end of the CoolStep™ hysteresis.

- Hysteresis width (SAP 170): determines the width and hence the upper end of the CoolStep™ hysteresis.
- Hysteresis start (SAP 172): determines the lower end of the CoolStep™ hysteresis.
- Threshold speed (SAP 182): CoolStep™will become active when the actual velocity of the motor is above this value. Setting this value to zero makes CoolStep™ always become active.
- Slow run current (SAP 183): The motor will run using this current when the velocity is below the theshold speed, but greater than zero.

Please see the TMCL Firmware Manual of your module for further details about these parameters.

To set up the CoolStep™ parameters, first run the motor at the desired velocity using the Velocity Mode tool. Leave the threshold speed at zero first. Now, choose a CoolStep™ hystereis using the hysteresis start and width values. The hysteresis is also being shown as a blue bar on the right hand side of the graph. Then, also set up the current down step and current up step parameters as well as the current minimum parameter. The red line in the graph shows how the current is being lowered or raised. If desired you can now also set the threshold speed and slow run current.

4.20.3 The TMCL™ Tab

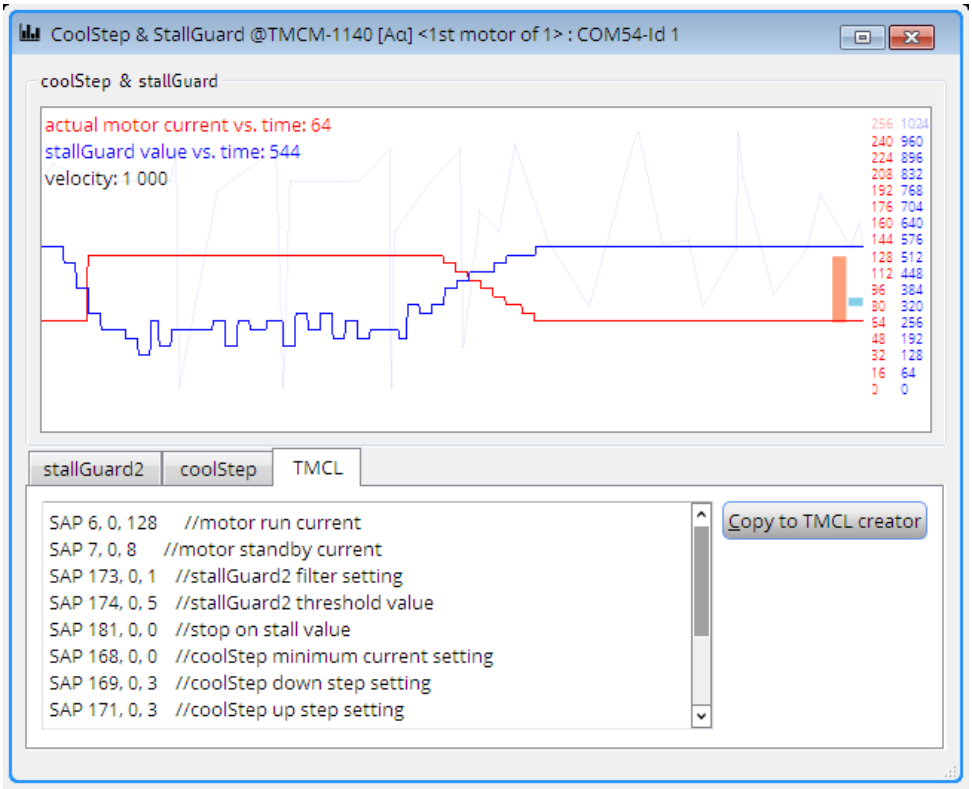


Figure 47: The TMCL™ Tab

The TMCL™ tab shows all adjustments which have been made on the other tabs of the CoolStep™ & StallGuard2™ tool as TMCL commands. Clicking the *Copy to TMCL Creator* button will copy all these commands to the TMCL Creator tool.

4.21 The Settings Tool (BLDC Motor)

The BLDC Settings tool provides functions for configuring a BLDC motor control module for use with a specific motor. It is provided once for each axis of the connected module.

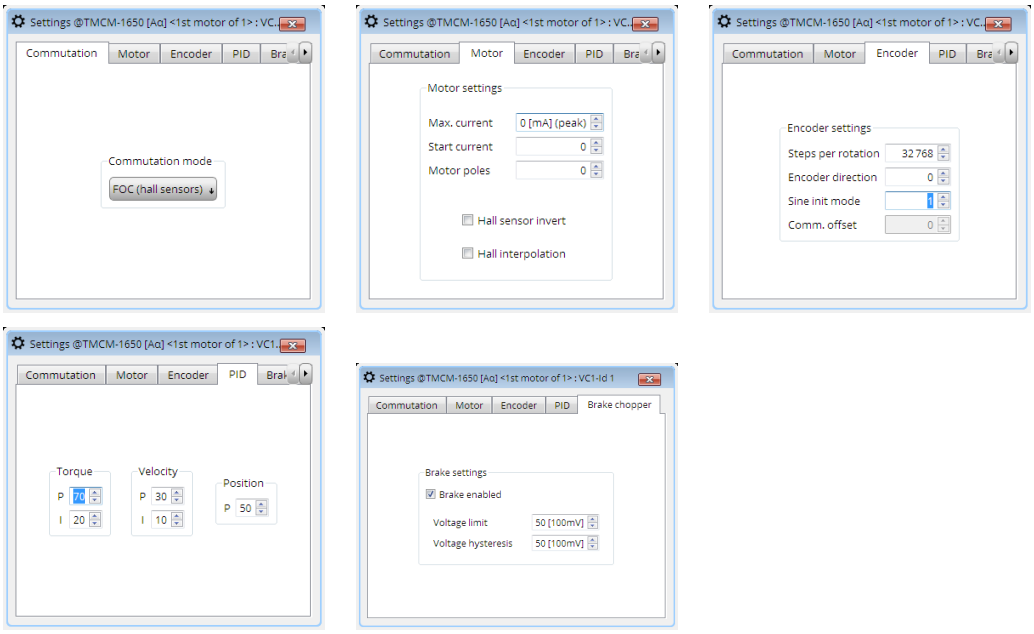


Figure 48: The BLDC Motor Settings Tool

Depending on the module itself, some settings might not be shown as not all modules support all these settings. The settings are grouped into the following tabs:

- *Communtation mode:* All commutation modes supported by the module are shown here. Pick the desired commutation mode from the list.
- *Motor:* All settings that are specific to the motor itself are shown here. These are mainly the maximum current, start current, number of poles and (if supported) if the hall signals have to be inverted or hall interpolation is to be used.
- *Encoder:* If supported, all settings specific to the encoder can be set here. These are the encoder steps per rotation (encoder resolution), encoder direction and the encoder initialization mode.
- *PID:* These are the parameters of the PID regulators.
- *Brake chopper:* If supported by the module, settings for an optional brake resistor can be made here.

After modifying a numerical value here, its background color will be changed to orange. This means that the modified value has not yet been sent to the module. Press the ENTER key on the keyboard to send the new value to the module. Its background color will then become white again. Each setting corresponds to one axis parameter in TMCL™. The correponding parameter numbers are shown when hovering over a setting with the mouse pointer. Please see the TMCL Firmware Manual of your module to learn more about each parameter.

4.22 The Flags Tool (BLDC Motor)

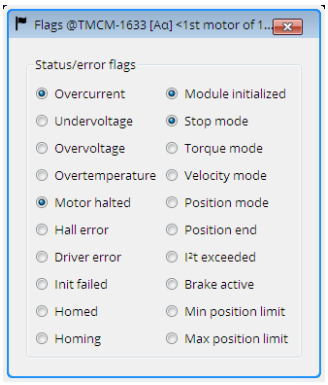


Figure 49: The BLDC Motor Flags Tool

The flags tool shows the states of all status flags and the states of all error flags that are provided by the connected module. Please see the TMCL Firmware Manual of your module to learn more about the status flags and the error flags provided by your module.

4.23 The Parameter Display

The Parameter Display is a tool for displaying the contents of selected TMCL™ axis parameters and user variables with automatic periodic update. This allows permanent monitoring of TMCL™ axis parameters and user variables (which is also possible while a TMCL™ program is running on the module). It can be used for example to help debugging a TMCL™ program. This tool is provided for BLDC motor modules.

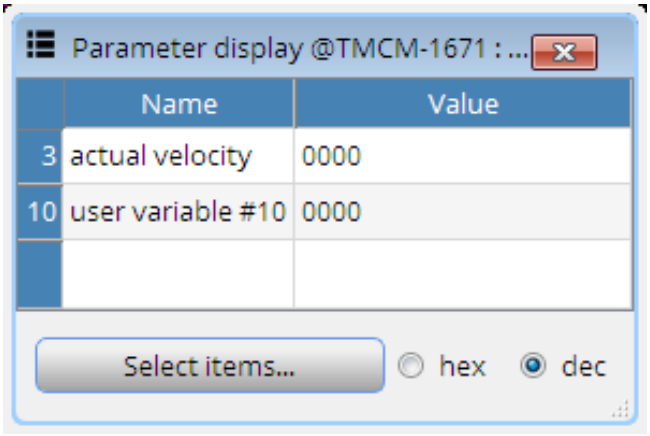


Figure 50: The Parameter Display

The contents of the displayed parameters can be presented in decimal or in hexadecimal form. This can be selected using the buttons at the bottom of the window. The displayed parameters are updated periodically, as fast as set up in the timer settings of the module connection. Clicking the *Select* button will show a dialog for selecting which parameters are to be displayed.

5 Evaluation Board specific Tools

When an evaluation board (Landungsbruecke or Startrampe in conjunction with one or two chip evaluation boards) is connected, some special evaluation board tools will appear in the tree view. These tools are described in the sections of this chapter.

Some of the module specific tools are also used in conjunction with evaluation boards. These are for example tools like Direct Mode, Position Graph, Velocity Graph, Position Mode, Velocity Mode and also some more tools described in chapter 4. Please see also there.

5.1 The Main Evaluation Board Tool

Click on the *Landungsbruecke* or *Startrampe* entry in the tree view to open the main evaluation board tool. This tool provides some information abot the connected evaluation board itself.

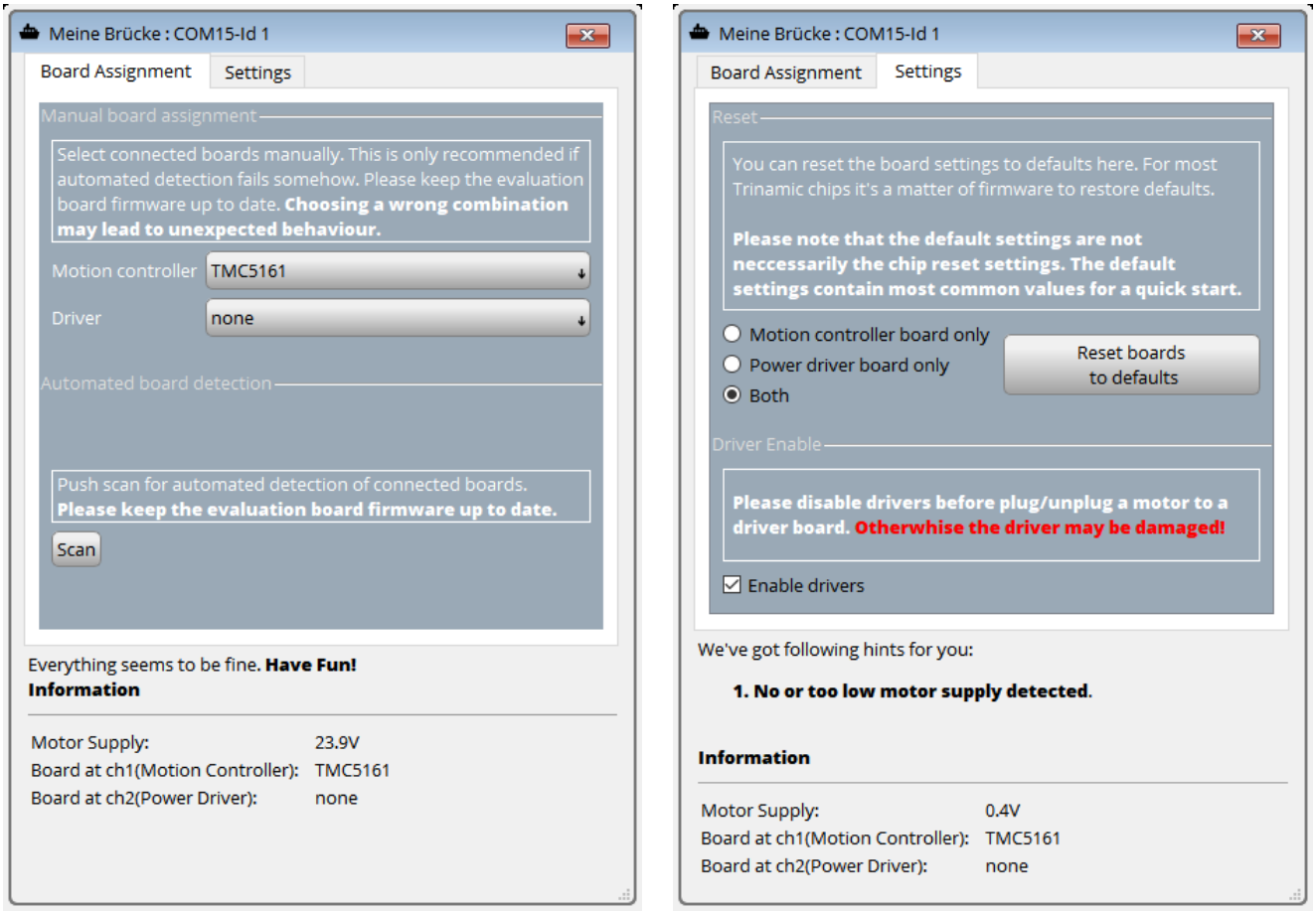


Figure 51: The Main Evaluation Board Tool

5.1.1 Board Assignment

On the *Board Assignment* tab, information about the evaluation boards connected to the Landungsbruecke or Startrampe board is shown. Under normal circumstances, the evaluation boards are detected automat-

ically. If desired the automatic board detection can be re-startet by clicking the *Scan* button.

If needed the connected evaluation boards can be set manually by picking them from the lists in the *Manual Board Assignment* section. Please note that setting the wrong boards may lead to unexpected behaviour.

5.1.2 Settings

On the *Settings* tab you can reset the connected evaluation boards to their reset state and also switch the motor drivers on or off.

5.1.3 Diagnostics

The *Diagnostics* section shows common diagnostic messages about the connected board.

5.2 The Register Browser

The register browser is the most important tool for evaluating chips. With the help of this tool it is possible to look into all registers of a chip an also to change the contents of the registers. So all the functionality of a chip can be explored in detail.

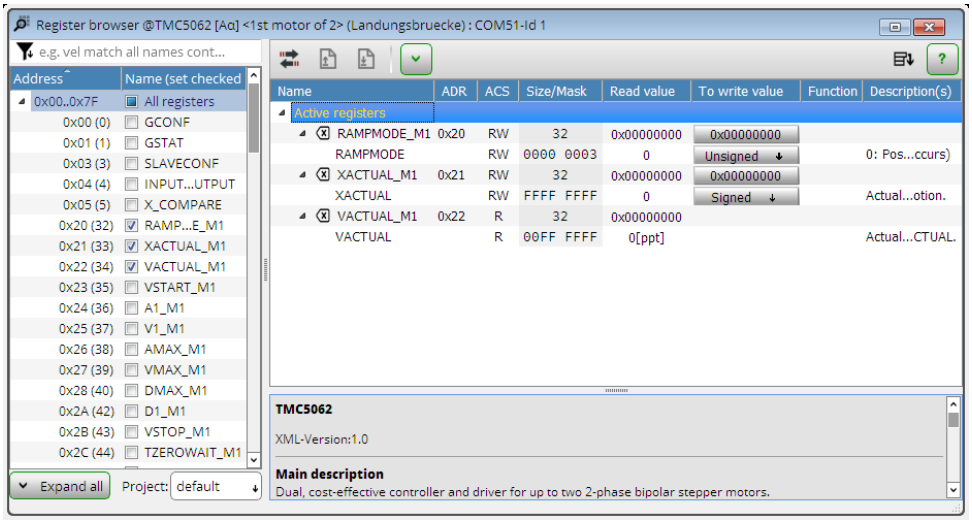


Figure 52: The Register Browser

5.2.1 Register List

On the left hand side of the register browser you can find a list of all registers. The registers are grouped together into functional groups. Click on a group to expand it and to show all registers that belong to that group. There is always also a group that contains all registers.

Check the checkbox of a register in the list to put it into the list of registers that are shown in the list of active registers.

5.2.2 Active Registers

In the center section of the register browser all register which are marked as active in the register list are shown. For every register there are the following entries:

- *Name:* The name of the register (same as in the chip datasheet). Click on the name to expand or to collapse the entry.
- *ADR:* The register address in the chip (same as in the chip datasheet).
- *ACS:* Shows how the register can be accessed. This can be R (read only), W (write only) or RW (read-/write).
- *Size/Mask:* Shows the width of the register (in bits) and which bits can be accessed.
- *Read value:* Shows the contents of the register if this register is readable. This value is cyclically read from the chip and updated here.
- *To Write value:* The contents of registers that are writable are shown here. Click on a value to modify it. Then the value modification dialog will be shown:

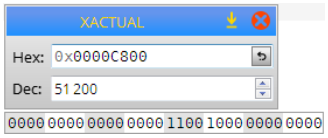


Figure 53: Changing a Register Value

  Here you can enter the new value either as a decimal value or as a hexadecimal value. Press the ENTER key to send the new value to the chip.

- *Function:* This column shows the functionality of single bits in the register, like in the chip datasheet.
- *Description:* Contains the overall describtion of the register (like in the chip datasheet).

5.2.3 Utilities

The register browser also contains some utilities:

- *Export/Import:* The export/import utility can save the values of all writable registers in different format. All writable registers are shown in a list which can either be sorted by the register addresses or chronological by the time of the last write to the registers. This list can then be saved in different ways:
  - INI file: an INI file can also be read back by the export/import utility and written to the registers again.
  - TPC file: a TMCL/PC source code file will be generated which can then be used by the TMCL/PC host.
  - Send to TMCL/PC host: instead of writing to a file the TMCL/PC source code can also be directly sent to the TMCL/PC host.
  - Other options: Here the register contents can be save to a C/C++ file with different options.

  Using the *Import* button an INI file can be read back and the values can be send to the registers. Click the *Apply to Device* button to send all registers contents to the chip.
- Generate a read command for the selected register and send this command to the TMCL/PC host.
- Generate a write command for the selected register and send this command to the TMCL/PC host.
- Read all registers at once.

5.3 The Flags View

Some chips are equipped with registers that provide flags. These are for example flags that represent the states of the end switches or the state of the ramp generator or error flags of a motor driver.
Such flags are shown in the flag view. Depending on the structure of the flags they are grouped into different tabs.

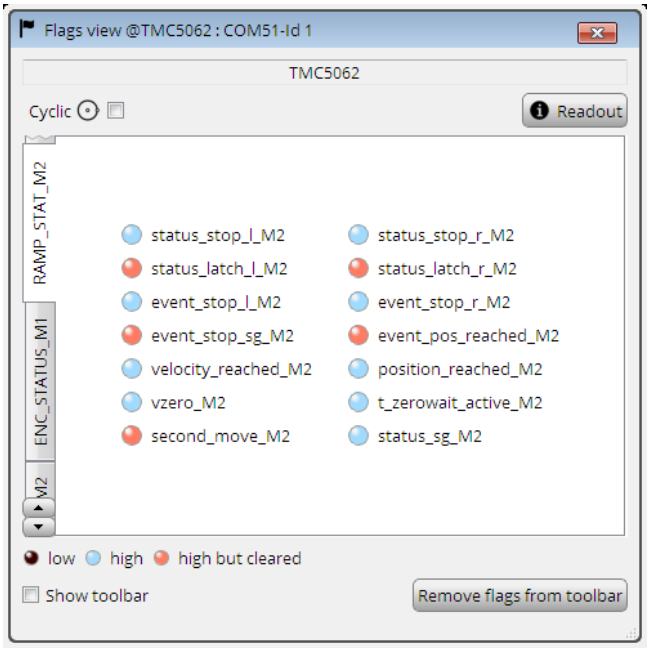


Figure 54: The Flags View

The flags view tool offers the following options:

- *Cyclic:* The flags view can either be updated automatically or manually. Check the *Cyclic* checkbox to switch on automatic update. If cyclic update is switched off then the flags view will not be updated automatically. In the this case click the *Readout* button to update the flags view.
- *Show Toolbar:* If this option is checked then flags can also be shown on a toolbar in the TMCL-IDE main window. You can then drag each flag to the toolbar. Clicking the *Remove flags from Toolbar* will remove all flags from the toolbar. Flags can also be dragged back from the toolbar to the flags view window.

5.4 The Chip-Click Tool

Some chips are equipped with input pins for configuring the functionality of the chip. These pins are normally directly connected to output pins of the Landungsbruecke or Startrampe board, so that their states can be set.

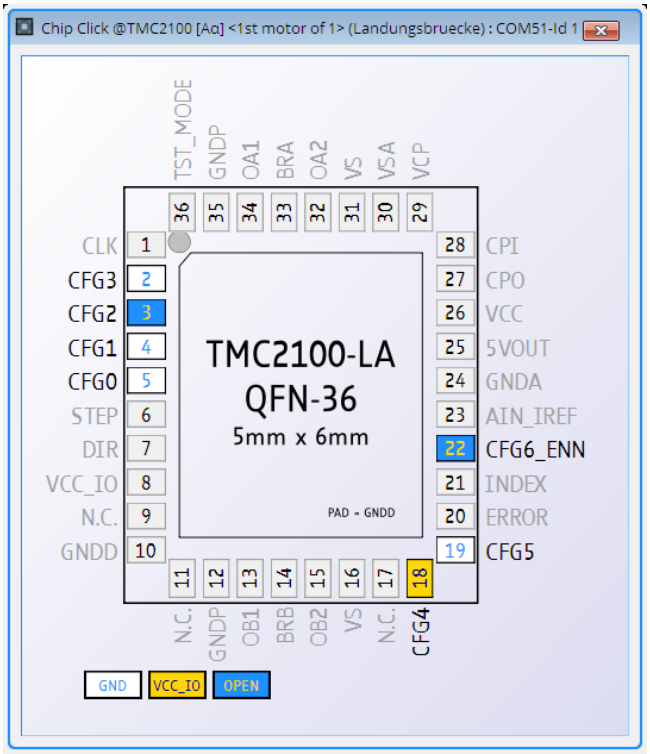


Figure 55: The Chip-Click Tool

The Chip-Click Tool is provided to set the states of those pins. The Chip-Click Tool shows all these pins on the footprint of the chip (as shown in the chip datasheet). Click on a pin to change between three possible states:

- GND : The pin is connected to ground and thus tied low.
- VCC_IO : The pin is connected to VCC_IO and thus pulled high.
- OPEN : The pin is unconnected an thus floating.

5.5 The Datagram Mode

The Datagram Mode is provided for all chips that are equipped with an SPI interface. It allows directly sending an SPI datagram to the connected chip.

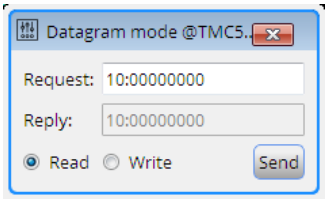


Figure 56: The Datagram Mode Tool

This tool provides the following functions:

- *Request:* Enter the datagram that you would like to send in hexadecimal form here. The first byte (seperated from the rest by a colon) typically is the register address. The rest (four bytes) represents the contents to be sent. After you have entered the data click the *Send* button to send it to the chip.
- *Reply:* Most chips send a reply datgram via SPI. This reply datagram will be displayed here.
- *Read / Write:* Here you can choose if the datagram shall be sent in read mode or in write mode, i.e. if the read/write bit shall be set or not set. Please see the datasheet of the chip for details on SPI datagrams.

6 Container Tools

Tools that are not specific to a module or axis are called container tools. These are tools that can work with all supported modules. Some can also work with more than one module at a time. All the container tools can be found on the right hand side of the icon bar at the top of the main window.

6.1 The TMCL/PC Host

The TMCL/PC Host is a tool that can control one or more modules using a TMCL program that is being excuted on the PC and not on the module. It also can be used to control evaluation boards (which do not support stand-alone TMCL programming) programmatically from the PC (which is sometimes needed for testing).

The language used by the TMCL/PC host is very similar to TMCL, but as the program is being executed on the PC it is called TMCL/PC. TMCL/PC is a dialect of TMCL that is executed on a PC as the host computer and is capable of controlling multiple Trinamic motion control modules and/or evaluation boards that are equipped with TMCL. Many commands are just put through to the modules, while other commands (for example branch commands) are executed directly on the host computer. Please see also the chapter about the syntax of TMCL/PC.

TMCL/PC programs can be developed and executed using the TMCL/PC host which is integrated into the TMCL-IDE as a container tool. The TMCL/PC host is not to be confused with the TMCL Creator.

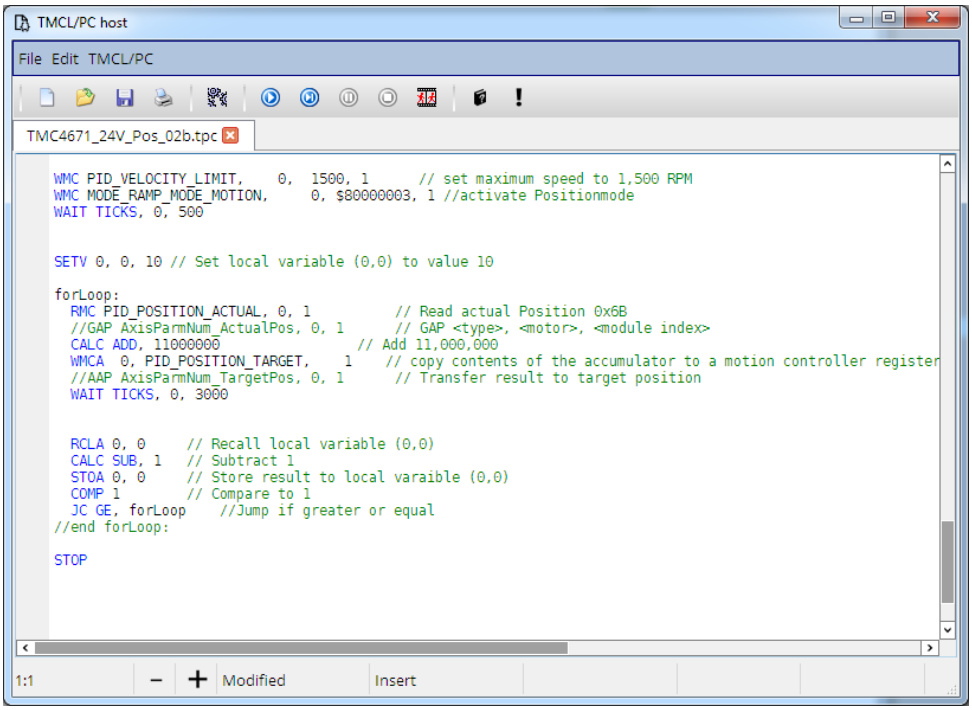


Figure 57: The TMCL/PC Host

6.1.1 The File Menu

The *File* menu of the TMCL Creator window provides the following functions:

- New: opens an empty editor tab for creating a new file.
- Open: loads a TMCL™ file into a new editor tab.
- Save: saves the content of the current editor tab to a file.
- Save as: saves the content of the current editor tab to a file using a new file name.
- Save all: saves all files that are currently loaded in the editor and that have been changed.
- Close: closes the currently open editor tab (same function as the close button of the editor tab).
- Print: print the content of the currently open editor tab.

6.1.2 The Edit Menu

The *Edit* menu provides standard text editor functions like *Undo*, *Redo*, *Cut*, *Copy*, *Paste* and a Find/Replace function. The *Settings* function opens a dialog where the text colors can be customized.

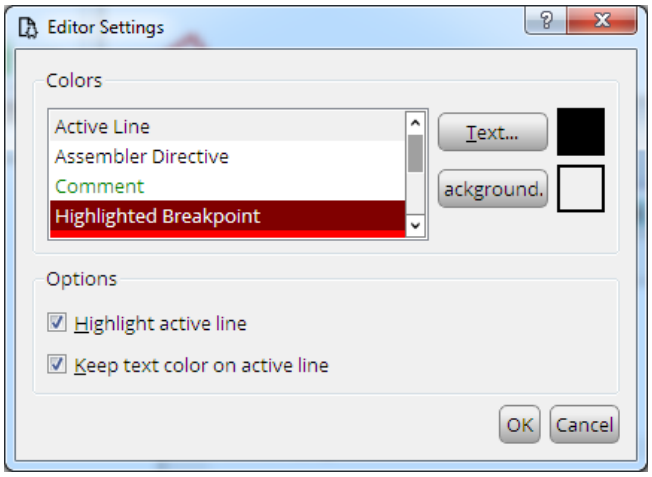


Figure 58: The Editor Settings Dialog

6.1.3 The TMCL/PC Menu

The *TMCL/PC* menu provides functions for assembling and executing TMCL™ programs as well as for setting some options of the TMCL/PC Host. These are the following functions:

- Assemble: Select *Assemble* from the *TMCL/PC* menu to assemble a file. If more than one file is open the currently selected file will be assembled. The process of the assembly is shown in a dialog (figure 59).

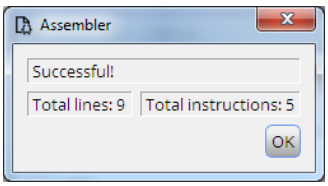


Figure 59: Progress of the TMCL/PC Assembly

- *Run/Continue:* After a program has been succesfully assembled it can be executed. Before executing a program please make sure that all motion control modules that are used by the program are connected and powerd. If a module does not respond to a command the program will be aborted with an error. This function is also used to continue the program after a breakpoint or after single stepping.
- *Animate:* This function automatically executes the TMCL/PC program step-by-step, so that the flow of the program can be seen. The actual position in the program is shown and updated after every command, and the contents of the accumulator and the X register are also shown on the status bar and updated after every command.
- *Single Step:* The *Single Step* function executes only the next command of a program. During single stepping, the next command to be executed is indicated by a green arrow on the left side of the editor. The contents of the local accumulator and the local X register are displayed on the status line at the bottom of the main window.
- *Pause:* The *Pause* function stops the execution of a running program, but the execution of the program can be continued by the *Run/Continue* function, the *Single Step* function or the *Animate* function.
- *Stop:* The *Stop & Reset* function stops the excution of a running program and resets the TMCL/PC interpreter. In this case the program cannot be continued, but it can be started again from the beginning using the *Run* function, the *Single Step* function or the *Animate* function.
- *Local Variables:* Choose *Local Variables* to open the Local Variables Window which can be used to inspect and to change the contents of TMCL/PC local variables.
- *Remove all Breakpoints:* This function removes all currently set breakpoints from the program.
- *Options:* The *Options* entry provides the following functions:
  - *Assembler/Debugger:* Opens the Assembler/Debugger Options dialog.
  - *Module Assignments:* Opens the Module Assigments dialog.

6.1.4 Breakpoints

For debugging purposes, breakpoints can be set. After a program has been successfully assembled, every line where a breakpoint can be set is indicated by a blue bullet on the left side of the editor. A breakpoint can be set by clicking on a blue bullet and is then indicated by a red bullet. A breakpoint can be removed by clicking on the red bullet again. The program execution will be stopped when the program reaches a breakpoint. It can then be continued by using the *Run/Continue* function, the *Animate* function or the *Single Step* function.

6.1.5 Assembler and Debugger Options

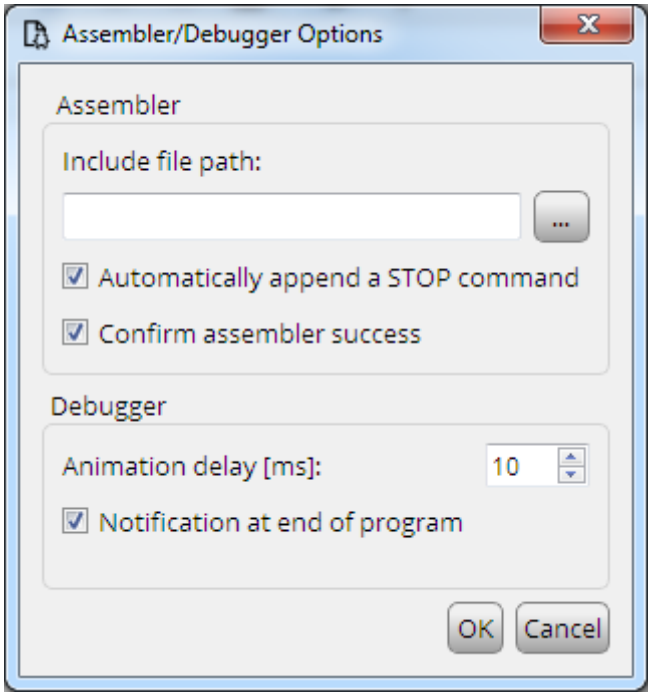


Figure 60: The TMCL/PC Options Dialog

In the Assembler/Debugger Options dialog the following options can be set:

- *Include file path:* The path used to search for include files included by the #include directive.
- *Automatically append a STOP command:* If this option is selected the assembler automatically appends a Stop command at the end of every TMCL/PC program (if the last command in the program is not already a STOP command).
- *Confirm assembler success:* If this option is selected then the assembler dialog will stay open after successful program assembly. With this option turned off the assembler dialog wil be closed automatically on success. Beginners should leave this option switched on (default setting), while experts might want to turn it off.
- *Animation delay:* Additional delay between two instructions when using the *Animate* function.
- *Notification at end of program:* If this option is selected the user will be notified at the end of the TMCL/PC program.

6.1.6 Module Assignments

At the beginning of a TMCL/PC program, all connected modules and/or evaluation boards that are to be controlled by the program have to be defined using module assignment commands. The Module Assigments dialog helps defining these assignments as it is capable of generating these commands.

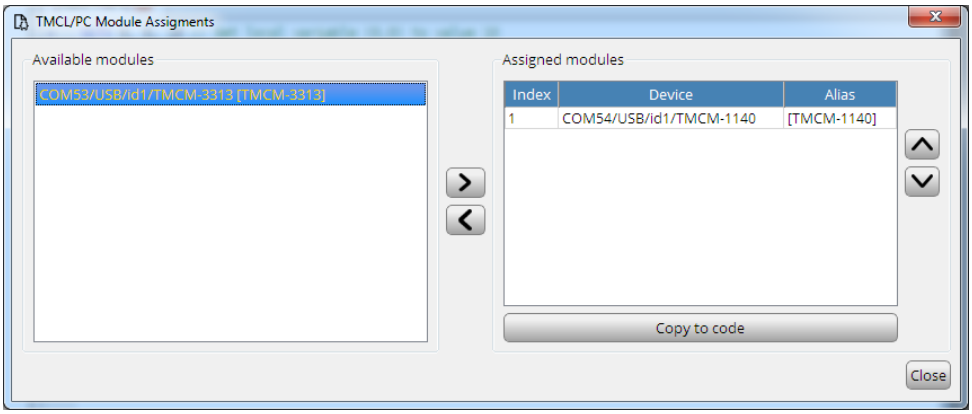


Figure 61: The Module Assigments Dialog

On the left side of the dialog, all available modules are shown. These are all modules and evaluation boards that are currently connected. Click on a module and then click the right arrow button or double click on a module to transfer it to the list of assigned modules.

On the right side of the dialog, all assigned modules are shown. Use the left arrow button to remove a module from the list of assigned modules (and to transfer it back to the list of available modules). Use the up and down arrow buttons to move an assigned module up and down on the list of assigned modules. Click on the index of a module to change its index, but take care not to assign the same index more than once. Mostly it is not necessary to change the module indices.

Finally, click the *Copy to code* button to generate the module assignment commands. These will be copied to the currently active editor page of the TMCL/PC host.

6.1.7 The Local Variables Window

The Local Variables Window provides functions for inspecting and changing the contents of the TMCL/PC local variables. It is a non-modal window that can be used together with the TMCL/PC main window.

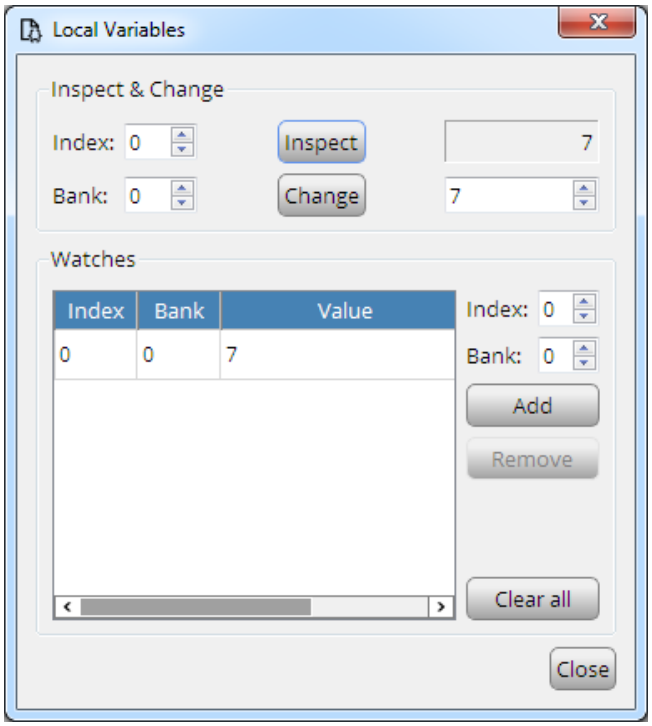


Figure 62: The Local Variables Window

Using the *Inspect & Change* section, the contents of local TMCL/PC variables can be inspected and also changed. First, enter the bank and the index of a local variable. Then, click the *Inspect* button to display its contents. The value will then be displayed on the right hand side beneath the *Inspect* button. To change its contents, enter a new value in the input field on the right hand side beneath the *Change* button. Then click the *Change* button, and the new value will be copied to the variable defined by the *Index* and *Bank* value.

Using the *Watches* section the contents of selectable local variable can also be displayed. The display will be updated whenever a TMCL/PC local variable gets changed during execution of a TMCL/PC program. This way it is possible to automatically watch the contents of local variables which can be very useful for debugging a TMCL/PC program. Enter the index and the bank number of a local variable and then click the *Add* button to add it to the list of watches. The *Remove* button can be used to remove a local variable from the watch list. The *Clear all* button clears the entire list of watches.

6.2 The Firmware Update Tool

The Firmware Update Tool provides the functionality for updating the firmware of TRINAMIC motor control modules and evaluation boards.

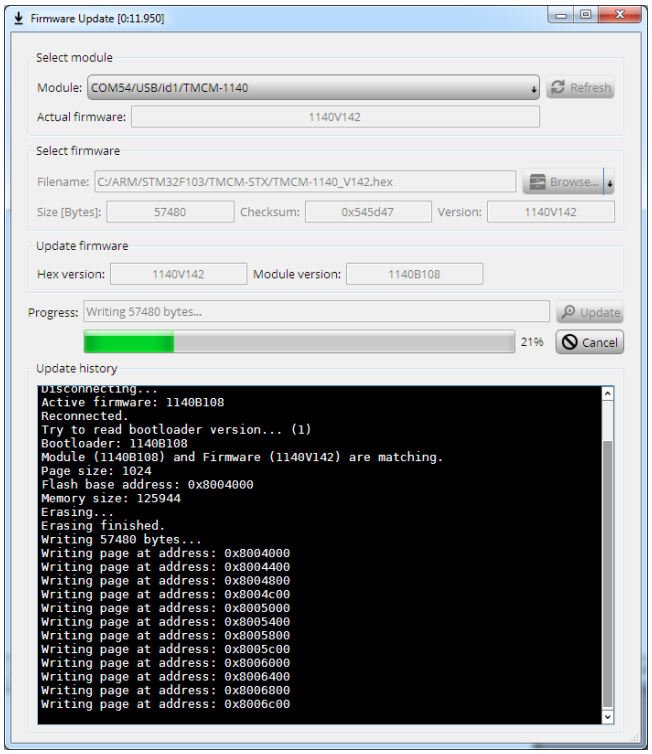


Figure 63: The Firmware Update Tool

6.2.1 Select Module

In the *Select Module* section all connected modules are shown in a list. If another module has been connected while the Firmware Update Tool is open press the *Refresh* button to refresh the list of available modules.

Select the module that is to be updated. After picking a module from the list, its actual firmware version will be displayed in the *Actual firmware* field.

6.2.2 Select Firmware

In the *Select Firmware* section you can select the firmware file to be downloaded to the module. Click the *Browse* button and select the desired firmware file. If the file is a valid firmware file then its size, check sum and version will be shown.

6.2.3 Update Firmware

Click the *Update* button to start the update process. The progress will be shown by the progress bar. The update history shows details of the upgrade process. This normally is only relevant if the update should go wrong, so that TRINAMIC can analyze the problem.

6.2.4 Hints for Firmware Updates

Here are some additional hints for firmware updates, especially for the case that updating the firmware should not work as expected.

6.2.4.1 Which Interface to choose?

With most modules, all interfaces the module is equipped with can also be used for firmware updates. So, firmware updates can take place via USB, RS232, RS485 or CAN. If the module is equipped with an USB interface then this is mostly the easiest way for updating the firmware.
When using RS485 it is often necessary that only the module that is to be updated is connected to the RS485 bus, so disconnect all other modules if problems should occur during firmware update.
For modules that are only equipped with a CAN interface and that are used with CANopen firmware, it is necessary to switch them into boot loader mode using the CANopen tab of the Connection window prior to using the Firmware Update Tool.

6.2.4.2 What if the Update Process has been aborted?

If anything should go wrong and the update process has been aborted, you can just try again. Power cycle the module (when using USB, also disconnect and then re-connect the USB cable) and also re-start the TMCL-IDE. Mostly the module is in bootloader mode then. It will be shown as module in bootloader mode in the tree view then.
When using RS232 or RS485 you will have to switch the baudrate in the connection window to 115200 before clicking the *Connect* button, as in bootloader mode the baud rate is always 115200, the module ID is always 1 and the reply ID is always 2.
When using CAN in bootloader mode, the bit rate is always 1000kBit/s, the module CAN ID is always 1 and the reply ID is always 2.

6.2.4.3 Old modules

Very old TRINAMIC modules (most of them (except for the TMCM-110) are not supported by the TMCL-IDE 3.x) use a different boot loader protocol. This is not supported by the Firmware Update Tool. In this case the firmware update will not be possible. For such old modules, use the TMCL-IDE 2.x to perform the firmware update. The TMCL-IDE 2.x can be downloaded as "Legacy IDE" from the TRINAMIC website.

6.3 The Settings Export/Import Tool

The Settings Export/Import Tool is a utility for reading out all important settings out of a module and generating TMCL source code from it. The generated code can then for example be taken over into the TMCL Creator.

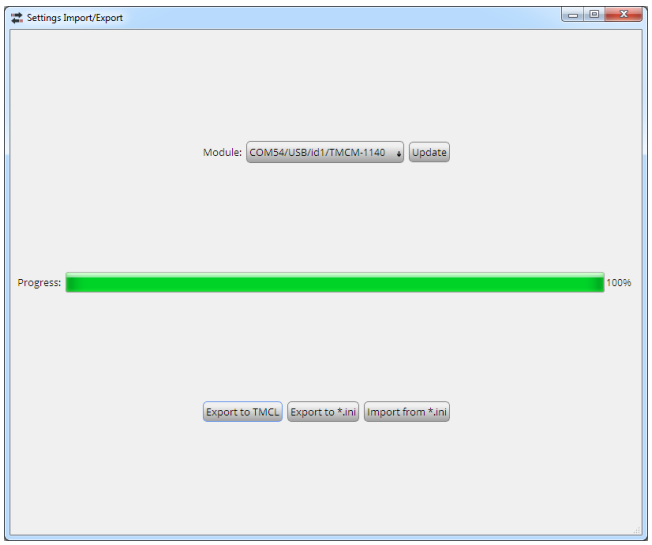


Figure 64: The Settings Export/Import Tool

The Settings Export/Import Tool provides the following functions:

- *Module:* A list of all connected modules is shown here. Choose one from the list. Click the *Update* button to refresh the list if when a new module has just been connected.
- *Export to TMCL:* This is the main function of this tool. It reads out all parameters and generates a TMCL program file that contains SAP commands for all these parameters. This file can then be used for example with the TMCL Creator.
- *Export to *.ini:* This function reads out all parameter settings and stores them in an INI file. This can be used to download the parameters later, using this tool. It is mainly to be used with evaluation boards.
- *Import from *.ini:* Use this function to download parameters that have previously been saved using the *Export to *.ini* function.

7 Wizards

The TMCL-IDE also provides tools that guide the user step by step through some more complex functions and setup processes of a module or a chip evaluation board. These kind of tools are called wizards. To use a wizard, first click the wizard icon on the tool bar to open the wizard selection window.

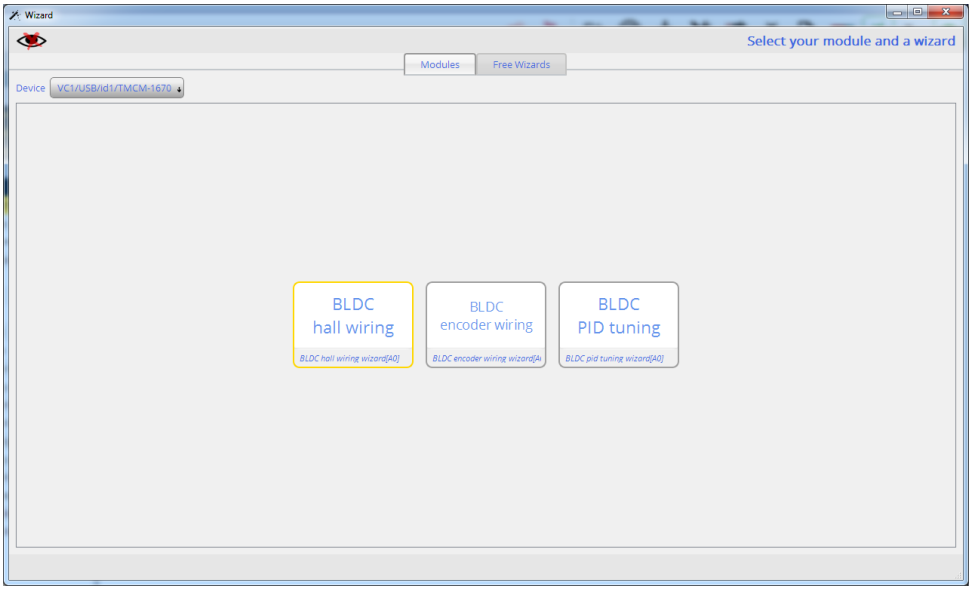


Figure 65: The Wizard Window

Here, one of the connected modules can be choosen. Depending on the module, all wizards supporting this module are shown. Now you can choose a wizard. Then you can go through it step-by-step. As all explanations are contained in the wizard itself the wizards are not described in detail here.

Here is a list of all wizards that are available in the current version of the TMCL-IDE:

- stallGuard2 Wizard
- coolStep Wizard
- Chopper Wizard
- dcStep Wizard
- Microstep Calibration Wizard
- BLDC Hall Wiring Wizard
- BLDC Encoder Tuning Wizard
- BLDC PID Tuning Wizard
- Weasel Configurator Wizard
- ESI Configurator Wizard

8 The Syntax of TMCL™

This section defines the syntax of the TMCL™ commands used in the TMCL™ Creator. Please see the TMCL™ Firmware Manual of your module for further explanations concerning the functionality of all TMCL™ commands that your module supports. The command mnemonics given there are used in the TMCL™ Creator. Please see also the sample program files that are available on the TRINAMIC website.

8.1 Assembler Directives

An assembler directive starts with the # sign, and the only directive is #include to include a file. The name of that file must be given after the #include directive. If this file has already been loaded into the editor then it will be taken from there. Otherwise it will be loaded from file, using the include file path that can be set in the Options dialogue of the TMCL™ Creator.

Example
#include test.tmc

8.2 Symbolic Constants

Symbolic constants are defined using the following syntax:

<Name> = <Value>

A Name must always start with a letter or the sign _ and may then contain any combination of letters, numbers and the sign _.

A Value must always be a decimal, hexadecimal or binary number or a constant expression. Hexadecimal numbers start with a $ sign, binary numbers start with a % sign.

Example

```
Speed=1000
Speed2=Speed/2
Mask=$FF
Binary Value=%1010101
```

8.3 Constant Expressions

Wherever a numerical value is needed, it can also be calculated during assembly. For this purpose constant expressions can be used. A constant expression is just a formula that evaluates to a constant value. The syntax is very similar to BASIC or other programming languages. Table 2 shows all functions and table 3 shows all operators that can be used in constant expressions.

The calculation takes place during compile time and not during runtime.

Internally, the assembler uses floating point arithmetic to evaluate a constant expression, but as TMCL™ commands only take integer values, the result of a constant expression will always be rounded to an integer value when being used as an argument to a TMCL™ command.

Functions in Constant Expressions	
Name	Function
SIN	Sinus
COS	Cosinus
TAN	Tangens
ASIN	Arcus Sinus
ACOS	Arcus Cosinus
ATAN	Arcus Tangens
LOG	Logarithm Base 10
LD	Logarithm Base 2
LN	Logarithm Base e
EXP	Power to Base e
SQRT	Square root
CBRT	Cubic root
ABS	Absolute value
INT	Integer (truncate)
ROUND	Integer (Round)
CEIL	Round upward
FLOOR	Round downward
SIGN	-1 if argument<1 0 if argument=0 1 if argument>0
DEG	Converts from radiant to degrees
RAD	Converts from degrees to radiant
SINH	Sinus hyperbolicus
COSH	Cosinus hyperbolicus
TANH	Tangens hyperbolicus
ASINH	Arcus sinus hyperbolicus
ACOSH	Arcus cosinus hyperbolicus
ATANH	Arcus tangens hyperbolicus

Table 2: Functions in Constant Expressions

Operators in Constant Expressions	
Operator	Function
()	Parenthesis

Operator	Function
^	Power
*	Multiply
/	Divide
\	Modulo
+	Add
-	Subtract

Table 3: Operators in Constant Expressions

Symbolic constants, floating point numbers, integer numbers, hexadecimal numbers and binary numbers can also be used in constant expressions.

Example

```
ROL 0, 7+9*8
Speed2=Speed*SIN(0.5)
MVP ABS, 0, 3*1000
Sin90=Sin(Rad(90))
```

8.4 Labels

The form used for labels is:

<Name>:

For names of labels the same rules as for names of symbolic constants apply.

Example

```
Loop:   MVP ABS, 0, 1000
        WAIT POS, 0, 0
        MVP ABS, 0, 0
        WAIT POS, 0, 0
        JA Loop
```

8.5 Comments

Comments always start with // (like in C++). The rest of the line will then be ignored.

8.6 TMCL™ Commands

Please see the TMCL™ Firmware Manual of your module for detailed explantions of all commands supported by the specific module.

9 The Syntax of TMCL/PC

The syntax of TMCL/PC is very similar to the syntax of TMCL™. Please see also chapter 8 for details. Most things defined there also apply for TMCL/PC. In this chapter we assume that the reader is already familiar with TMCL™, and only the differences between TMCL/PC and TMCL™ are pointed out.

9.1 Module Index Parameter

All commands that are directly put through to the modules have an additional module index parameter. This is to specify on which of the connected modules the command shall be executed. The module index parameter is the last parameter of the command. The module indices must be defined using the #module directive. Here is a list of all commands that use the additional module index parameter:

- ROL <motor>,<velocity>,<module index>
- ROR <motor>,<velocity>,<module index>
- MST <motor>, <module index>
- MVP <ABS|REL|COORD>, <motor>, <position>, <module index>
- SAP <type>, <motor>,<value>, <module index>
- GAP <type>, <motor>,<module index>
- STAP <type>, <motor>, <module index>
- RSAP <type>, <motor>, <module index>
- SGP <type>, <bank>, <value>, <module index>
- GGP <type>, <bank>, <module index>
- STGP <type>, <bank>, <module index>
- RSGP <type>, <bank>, <module index>
- RFS <START|STOP|STATUS>, <motor>, <module index>
- SIO <type>, <bank>, <value>, <module index>
- GIO <type>, <bank>, <module index>
- SCO <type>, <motor>, <value>, <module index>
- GCO <type>, <motor>, <module index>
- CCO <type>, <motor>, <module index>
- SAC <type>, <bank>, <value>, <module index>
- UF0 <type>, <motor>, <value>, <module index>
- UF1 <type>, <motor>, <value>, <module index>
- UF2 <type>, <motor>, <value>, <module index>
- UF3 <type>, <motor>, <value>, <module index>
- UF4 <type>, <motor>, <value>, <module index>
- UF5 <type>, <motor>, <value>, <module index>
- UF6 <type>, <motor>, <value>, <module index>
- UF7 <type>, <motor>, <value>, <module index>
- WAIT <POS|RFS|REFSW|LIMSW>, <motor>, <timeout>, <module index>

9.2 Module Index Definition

The module indices are defined using the #module directive. A module index can be any number between 1 and 255. The #module directive has two parameters: the module index that is to be defined and the key of the module that is to be assigned to this module index. Optionally the alias name of the module can also be supplied as a third parameter (in brackets). So the module definitions can look like this:

```
#module 3 COM2/Serial/id1/TMCM-1140 [TMCM_1140]
#module 1 COM54/USB/id1/TMCM-1161 [TMCM_1161]
```

Then, commands can use module index 1 and module index 3, like this:

```
MVP ABS, 0, 51200, 1
MVP REL, 0, 1000, 3
```

There is also a tool that can generate the module definitions: Just choose *Options/Module definitions* in the TMCL/PC menu to display the Module Assignment Dialog. There you can put together the module definitions and copy them into your TMCL/PC program.

9.3 Local Commands

All other TMCL commands are executed locally in the TMCL/PC interpreter. The TMCL/PC interpreter also has a local accumulator register, a local X register and local TMCL flags. The results of the GAP, GGP, GIO, GCO and SAC commands are copied into the local accumulator register. The following TMCL commands are executed locally and and work with the contents of the local accumulator register or local X register or local TMCL flags:

- COMP <value>
- CALC <type>, <value>
- CALCX <type>
- JC <condition>, <label>
- JA <label>
- CSUB <label>
- RSUB
- WAIT TICKS, 0, <time>
- STOP
- CLE <flags>
- AAP <type>, <axis>, <module index> (Copies the contents of the local accumulator to an axis parameter of a module.)
- AGP <type>, <axis>, <module index> (Copies the contents of the local accumulator to a global parameter of a module.)

Please note that the commands AAP and AGP are special cases: they are executed on the locally, but change a value in on module with the given module index. In fact, an SAP command respective an SGP command will be generated internally for the module.

9.4 Local Variables

The TMCL/PC interpreter provides an array of local variables for storing values locally. There are 256 banks of local variables, and each bank provides 256 local variables. For using these local variables, some special commands are provided:

- STOA <index>, <bank>: stores the contents of the local accumulator to the local variable specified by <index> and <bank>.
- STOX <index>, <bank>: stores the contents of the local X register to the local variable specified by <index> and <bank>.
- RCLA <index>, <bank>: copies the contents of the local variable specified by <index> and <bank> to the local accumulator.
- RCLX <index>, <bank>: copies the contents of the local variable specified by <index> and <bank> to the local X register.
- SETV <index>, <bank>, <value>: Initializes the local variable specified by <index> and <bank> with the value specified by <value>.

The range of the <index> and <bank> parameters is 0...255. The contents of local variables can also be displayed for debugging purposes using the Local Variables Window.

9.5 Evaluation Board Commands

There are also some special commands for using Trinamic evaluation boards (combinations of Startrampe or Landungsbruecke with other chip evaluation boards). These are the following commands:

- UFMC <type>, <motor>, <value>, <module index>: special UF command for the motor controller connected to the Startrampe/Landungsbruecke.
- UFDRV <type>, <motor>, <value>, <module index>: special UF command for the motor driver connected to the Startrampe/Landungsbruecke.
- WMC <register>, <upper>, <value>, <module index>: write to a motion controller register. Always set the parameter <upper> to 0 when writing to registers of up to 32 bits width. When writing to a 64 bit register, two WMC commands are needed. Always write to the lower 32 bits first with <upper> set to 0 and then write the upper 32 bits with <upper> set to 255.
- WDRV <register>, <upper>, <value>, <module index>: write to a motor driver register. Always set the parameter <upper> to 0 when writing to registers of up to 32 bits width. When writing to a 64 bit register, two WDRV commands are needed. Always write to the lower 32 bits first with <upper> set to 0 and then write the upper 32 bits with <upper> set to 255.
- RMC <register>, <upper>, <module index>: read from a motion controller register. Always set the parameter <upper> to 0 when reading from registers of up to 32 bits width. When reading from a 64 bit register two RMC commands are needed. First, read the lower 32 bits with the parameter <upper> set to 0 and then read the upper 32 bits with <upper> set to 255.
- RDRV <register>, <upper>, <module index>: read from a motor driver register. For Always set the parameter <upper> to 0 when reading from registers of up to 32 bits width. When reading from a 64 bit register two RDRV commands are needed. First, read the lower 32 bits with the parameter <upper> set to 0 and then read the upper 32 bits with <upper> set to 255.
- WMCA <type>, <register>, <module index>: copy contents of the accumulator to a motion controller register. This command can only access registers of up to 32 bit width.
- WDRVA <type>, <register>, <module index>: copy contents of the accumulator to a motor driver register. This command can only access registers of up to 32 bits width.

10 Figures Index

11 Tables Index

12 Supplemental Directives

12.1 Producer Information

12.2 Copyright

12.3 Trademark Designations and Symbols

Trademark designations and symbols used in this documentation indicate that a product or feature is owned and registered as trademark and/or patent either by TRINAMIC or by other manufacturers, whose products are used or referred to in combination with TRINAMIC's products and TRINAMIC's product documentation.

This PC Software is a non-commercial publication that seeks to provide concise scientific and technical user information to the target user. Thus, trademark designations and symbols are only entered in the Short Spec of this document that introduces the product at a quick glance. The trademark designation /symbol is also entered when the product or feature name occurs for the first time in the document. All trademarks and brand names used are property of their respective owners.

12.4 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.

12.5 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.

12.6 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.

12.7 Collateral Documents & Tools

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

13 Revision History

13.1 Software Revision

Version	Date	Author	Description
V3.0.21.0	2018-MAR-28	MJ	Release version.
V3.0.22.0	2018-SEP-09	MJ	Release version.
V3.0.23.0	2019-JAN-24	MJ	Release version.
V3.0.24.0	2019-APR-17	MJ	Release version.
V3.0.25.0	2019-OCT-22	MJ	Release version.
V3.0.26.0	2019-DEC-20	MJ	Release version.
V3.3.0.0	2021-MAR-24	MJ	Release version.

Table 4: Software Revision

13.2 Document Revision

Version	Date	Author	Description
V3.00	2018-JUN-28	OK	Completely revised version for the TMCL-IDE V3.x.
V3.01	2018-SEP-09	MJ	Covers TMCL-IDE V3.0.21.0.
V3.02	2019-APR-17	OK	Hints for Peak CAN interfaces added. Covers TMCL-IDE V3.0.24.0.
V3.03	2019-AUG-23	OK	Covers TMCL-IDE V3.0.25.0.
V3.04	2019-NOV-22	OK	List of supported CAN interfaces updated. Covers TMCL-IDE V3.0.26.0.
V3.05	2021-MAR-23	OK	Release for TMCL-IDE V3.3.0.0

Table 5: Document Revision