MCU Motor Studio
User Manual

Description

User manual for MCU Motor Studio PC Tool.

Table of Contents

List of Figures

List of Tables

Conventions used in this document

Numerical Values

Hexadecimal number: 0xABC or 0h12F
Decimal number: 123 or 0d123 (explicitly indicating the decimal numbers)
Binary number: 0b111

Signals

Active low signals are indicated by a _N at the end of the signal name. Example: RESET_N.

Assertion of a signal shall mean its activation (transition to the active state). De-assertion of a signal shall mean its deactivation (transition to the inactive state).

Bus signals are indicated by [x:y] at the end of signal name. Example: DATA[3:0] indicates a four bit bus with the individual bus signals DATA[3], DATA[2], DATA[1] and DATA[0].

Registers

Register names are indicated by square brackets […]. Example: [ABCD].

Two or more of the same kind of registers, fields, and bit names are collectively referred to by using a numerical suffix n. Example: [XYZ1], [XYZ2] and [XYZ3] are collectively referred to as [XYZn].

The bit width of a register is expressed as [x:y] where x is the number of the most significant bit and y is the number of the least significant bit. Example: [XYZ][3:0] indicates a four bit-wide register named XYZ. The configuration value of a register is expressed by either a hexadecimal number or a binary number. Example: [ABCD].EFG = 0x01 (hexadecimal), [XYZn].XY = 0b1 (binary).

The following definitions apply for Bytes and Words:

Byte	8 bits
Half Word	16 bits
Word	32 bits
Double Word	64 bits

Unless specified otherwise, registers support only word access.

Register which are indicated to be reserved must not be rewritten. The read value from reserved registers must not be used.

Properties of each bit in a register are expressed as follows:

R	Read only
W	Write only
W1C	Write 1 Clear; the corresponding bit is cleared (=0) when "1" is written to this bit.
W1S	Write 1 Set; the corresponding bit is set (=1) when "1" is written to this bit.
R/W	Read and Write are possible.
R/W0C	Read/Write 0 Clear
R/W1C	Read/Write 1 Clear
R/W1S	Read/Write 1 Set
RS/WC	Read Set/Write Clear; set after read operation, cleared after write operation.

Reading from register bits having a default value of "—" will result in an unknown value.

In case of write accesses to registers containing both read/write (R/W) and read-only (R) bits, the read-only bits shall be written with their default value. If this default is "—", follow the instructions of each register.

Reserved bits of Write-only (W) register should be written with their default value. If this default is "—", follow the instructions of each register.

1. Introduction

The TOSHIBA 3-Phase Motor's Vector Control Solution has two main components:

- A highly scalable and fully configurable Motor Control Firmware designed for the TMPM4K series MCUs, featuring Field Oriented Control (FOC) of up to three motors.
- The "MCU Motor Studio" PC Tool for Microsoft Windows, is utilized for parameter configuration, drive control and real-time logging of various motor parameters in a high-speed Digital Storage Oscilloscope fashion.

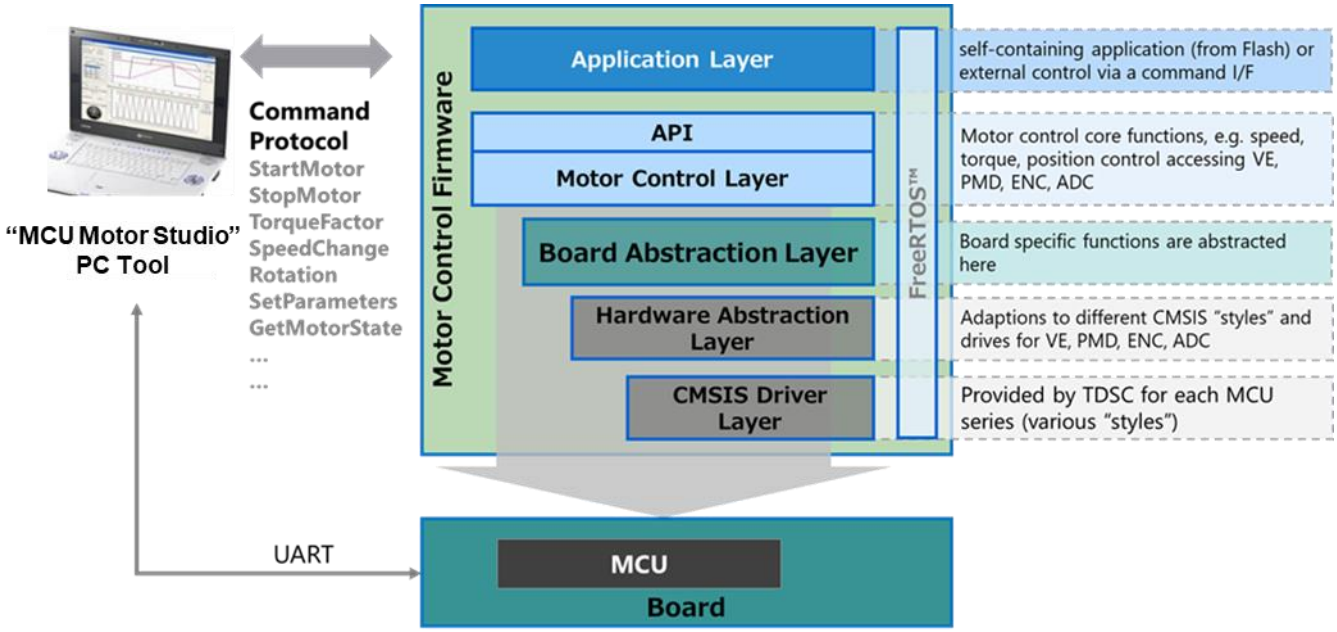


Figure 1-1 TOSHIBA 3-Phase Motor's Vector Control Solution

The main scope of this document is to describe the "MCU Motor Studio" PC Tool for Microsoft Windows, its features, components, configuration and usage.

All statements made in this guide are based on version 3.10 of the Motor Control Firmware and version, 4.1.0 of the "MCU Motor Studio" PC Tool. Minor differences in the configuration and the PC tool's look & feel are possible depending on the versions used. However, all operation principles remain unchanged.

The terms "MCU Motor Studio" and "PC Tool" as used throughout this document are referring to one and the same software entity and are fully interchangeable.

The next chapters will detail the functionality, the layout, configuration and usage of the "MCU Motor Studio".

2. Main Features

"MCU Motor Studio" was designed as a Tool for configuration and external control of the M4Kx series of MCUs. The table below summarizes its main characteristics and all supported features. Some of these are only available for particular firmware versions/configurations. Such limitations are strictly enforced by the concrete hardware platform, the used MCU and the particular configuration of the firmware that is executed on each platform.

Every single feature is fully implemented and tested for at least one platform/configuration and is therefore applicable to all others if supported by the underlying hardware:

Table 2.1 "MCU Motor Studio" Main Features List

"TOSHIBA MCU Motor Studio" Feature List	
Motor Drive Control	Speed / Torque / Position (per channel)
Sequencer	Sequential Command List Execution / Wait Stage Support PC tool Layout Control Sequence Save / Recall / Clear
Board Information	Base & Power Board Configuration / Versioning Information / Supported Features
Parameter Configuration	Run-time Motor / Encoder / Board / PI Regulator / System (limitations apply)
Statistics	Up to 4 fixed parameters (Speed, Torque, Current) Up to 3 channels individually with optional CSV logging of the selected parameters Scalable Timeline / Manual Zoom Status Information / DC Voltage / Temperature / Errors
Layout Configuration	Save Current View / Restore Default View
Motor Parameter Configuration	Configuration Save / Recall from XML Export to C Header file for Firmware inclusion. NVM (EEPROM/Flash) Save/ Recall (in future releases)
Sensor-less Position Control	Advanced Software Positioning / Advanced Turn Control
Sensor Position Control	Linear Motion Control
Application Control	Demo Control
DSO / HS-DSO	Logging of up to 8 parameters with scalable spread factor
Built-in Calculators	Current Sensitivity / Voltage Sensitivity / Pole Pairs / Single Phase Impedance
Supported MCU Devices	M4Kx
Supported OS	Microsoft Windows 10.x
Supported Languages	English / German / Japanese

Not all of the features can be configured and used simultaneously on any of the family members. It is all dependent on the available memory and in some cases may require optimization of the task footprint.

3. Set-up

3.1. Requirements

3.1.1. Supported platforms

The application is built to run exclusively on Microsoft Windows based platforms (**Microsoft Windows 10 32/64-bit OS).**

3.1.2. Disk space

5MB of free disk space is needed for the typical installation package.

12MB of free disk space is needed for the application. Please allow some more disk space for the log an configuration files that can be generated during the normal operation.

3.1.3. Memory usage

The minimal amount of RAM required to ensure normal operation in all possible cases is **128MB**.

3.1.4. Connectivity

At least one free Serial or USB port for the protocol communication (via the assigned COM port) is required. A second one might be needed given the optional HS-DSO communication will be used.

3.1.5. Additional Software Components

“MCU Motor Studio” is a .NET framework-dependent published application. It does not include the .NET runtime and libraries, just the executable and all third-party components it is utilizing.
If the needed .NET resources are not pre-installed, a dedicated set-up program will be automatically started. It will attempt to download and install the missing prerequisite components. In such cases the user shall ensure there is active Internet connection and the access to the component vendor’s websites is permitted.
Alternatively, an offline installation of the required .NET 3.5 SP1, .NET 4.0 Client Profile and Microsoft Windows® Installer 3.1 packages can be performed prior to the “MCU Motor Studio” installation.

3.2. Package Content

The installer package contains “MCU Motor Studio.msi” - the Microsoft Windows® installer.

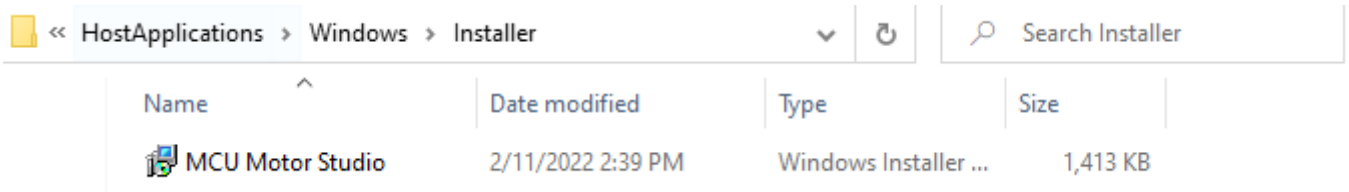


Figure 3-1 Installation package content

3.3. Detailed Guide

3.3.1. Precautions during installation

Please note below precautions during installation,

1. Do not attempt to install from a network drive or removable media. The installation package has to be copied to a local drive.
2. Please ensure that the installer has valid Code Signing Certificate.
3. It is recommended to use small path names during installation, please avoid the use of special characters as directory names.

4. Multiple instances or various versions are not supported.
5. Please un-install any previous versions of “MCU Motor Studio”, if already installed.
6. When executing the installer, make sure that no suspicious files exist in the same directory.
7. When executing the installer, copy the files to a newly created temporary directory before executing it.
8. It is recommended that you do not leave files downloaded from external sites in the download directory. For example, if you download a crafted DLL file without knowing it, and then download the installer and run it, you will run the installer with the crafted DLL file in the same directory, which is dangerous.

In case user attempts to install “MCU Motor Studio” when software is previously installed the following error message will appear:

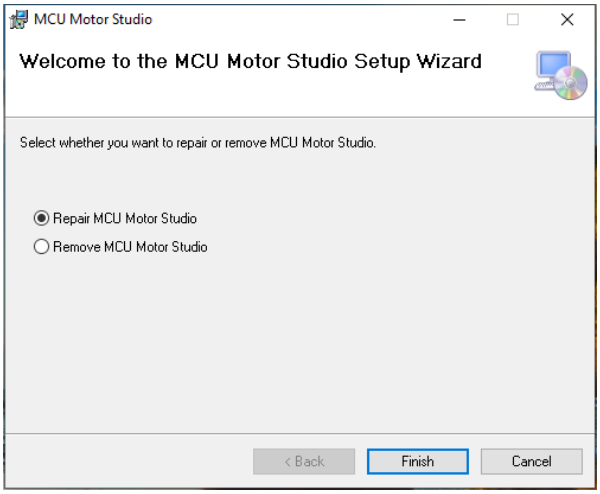


Figure 3-2 Application installation failure window

3.3.2. Installation steps

Execute “MCU Motor Studio.msi” - the Microsoft Windows® installer – please ensure user has admin privileges prior to the installation.

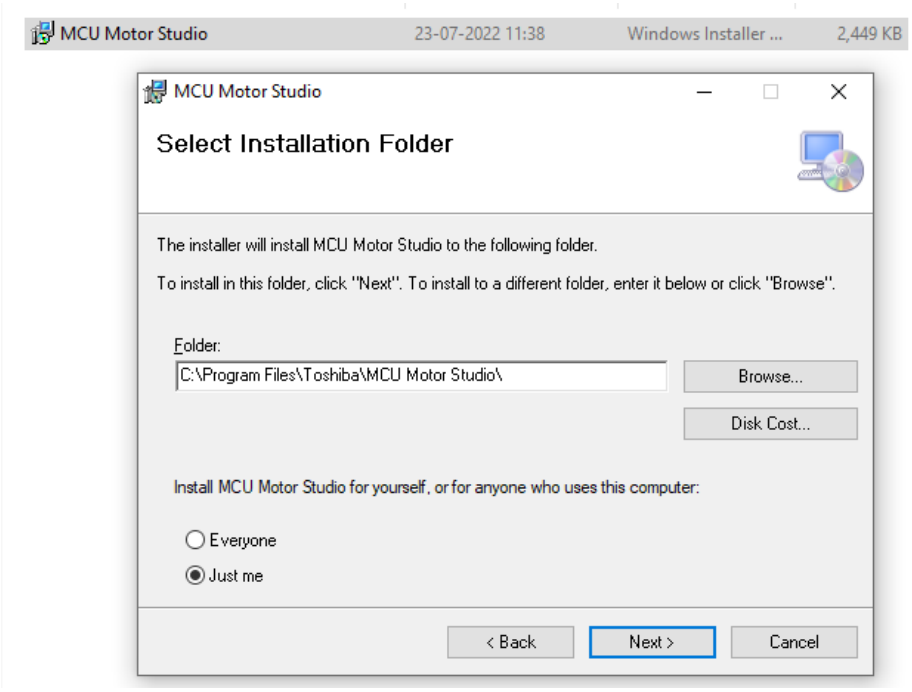


Figure 3-3 Installation window

Please ensure that the installer file is digitally signed using a valid Toshiba certificate by clicking on ‘Show more details’.

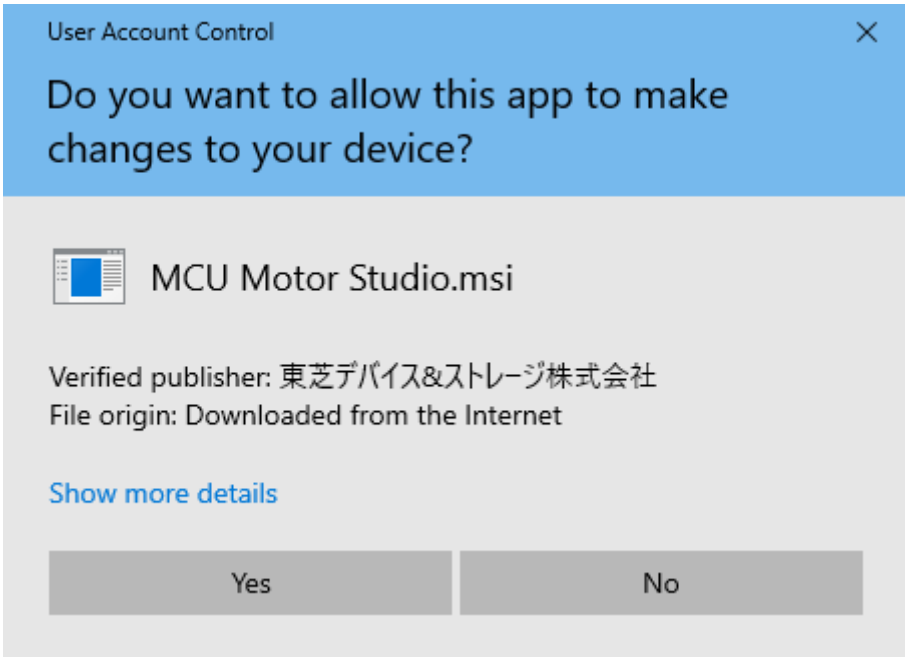


Figure 3-4 Digitally signed installer message window

3.4. Install MSI from command line with Administrator

The MCU Motor Studio installation using command prompt with Administrator.
In case user faces any issues during normal installation process, please try command prompt based msi file installation, below is the procedure.

- Right click on Windows Start, choose Command Prompt (Admin)
- In the command prompt, input “msiexec /i “path\MCU Motor Studio_64bit.msi“”
- Press Enter to start the install process.

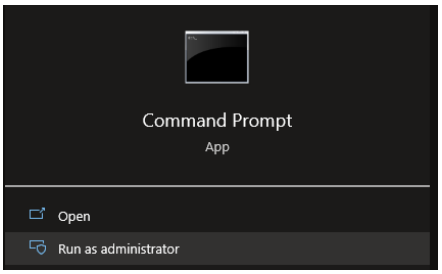


Figure 3-5 Install from Command Prompt

3.4.1. Initial configurations

After successful installation “MCU Motor Studio” will be automatically started and an initial configuration will be prompted. You need to have a supported platform already powered and connected to one of the COM ports. Please select the correct COM port and confirm:

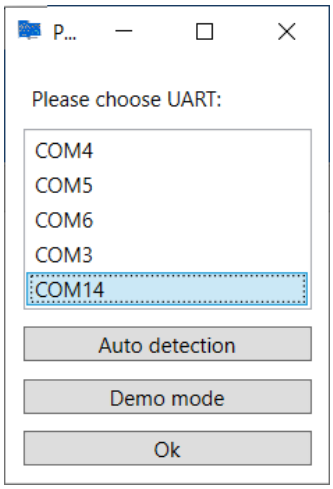


Figure 3-6 Initial COM port selection

The Tool features **Auto Detection mode**. If selected all available COM ports will be scanned in ascending order, attempting to establish communication with the attached board. The process may be rather slow, depending on the number of ports in the system. It is strongly recommended to have the platform powered and connected to the PC with any required USB to Serial bridge driver properly installed prior to any connection attempts.

The last used COM port will be memorized and used the next time the Tool is started. The COM port section window will only be prompted again, if the attempted connection cannot be established.

Next you shall select the motor channel you would like to start with. The prompt will be presented even if only one channel is configured and usable on the platform that is currently connected, given the MCU is capable of driving more channels.

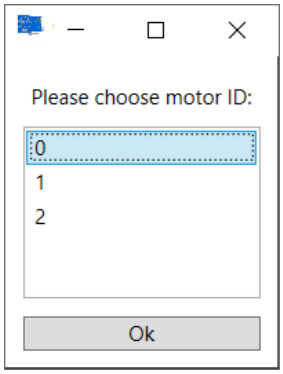


Figure 3-7 Initial motor channel selection

The last used channel will be memorized and used at the next start. The channel selection will be prompted again, only if the channel is not available, typically as different platform with different capabilities is connected.
You are ready to start using the powerful PC tool and dive into the intriguing world of Toshiba's Motor Control Solution. **Welcome to "MCU Motor Studio"**!

3.5. De-installation

There are several cases where the application has to be completely removed:
New "MCU Motor Studio" version is available and shall be installed – in that case the user needs to remove the previously installed one.
The current version is to be re-installed – very unlikely, but in some cases this might be one of the first trials to resolve issues with the application.
The application if is not needed anymore – although it is very tiny, sometimes a clean-up is needed.

The removal is very easy and follows the Microsoft Windows standard approach. From Control Panel -> Programs select “MCU Motor Studio” in the list (as illustrated below), hit Uninstall and confirm the removal with OK:

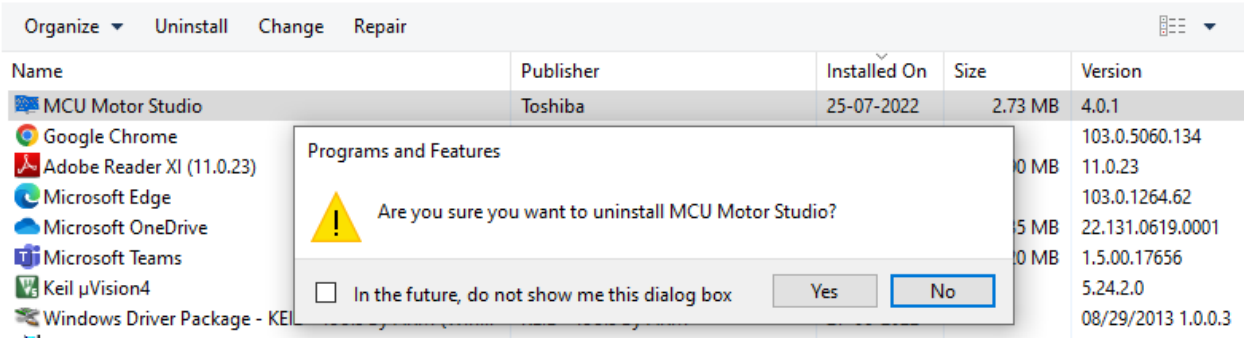


Figure 3-8 “MCU Motor Studio” de-installations

After restart all “MCU Motor Studio” related files and folder shall be completely removed. Typically, there is no need to perform any manual deletion of any empty folders and/or files, unless you have specified different location for storing auto generated files, such as logs, parameter configuration files, sequences, etc. It is assumed in this case that the user may want to keep these and therefore these are not automatically removed.

4. Detailed PC tool Description

4.1. Main window

4.1.1. Overview

The tool comes with a default layout that orders most common controls assuming standard application with selectable speed/torque control of three motor channels, system and board information panels, DSO monitor and no advanced features like position or demo control:

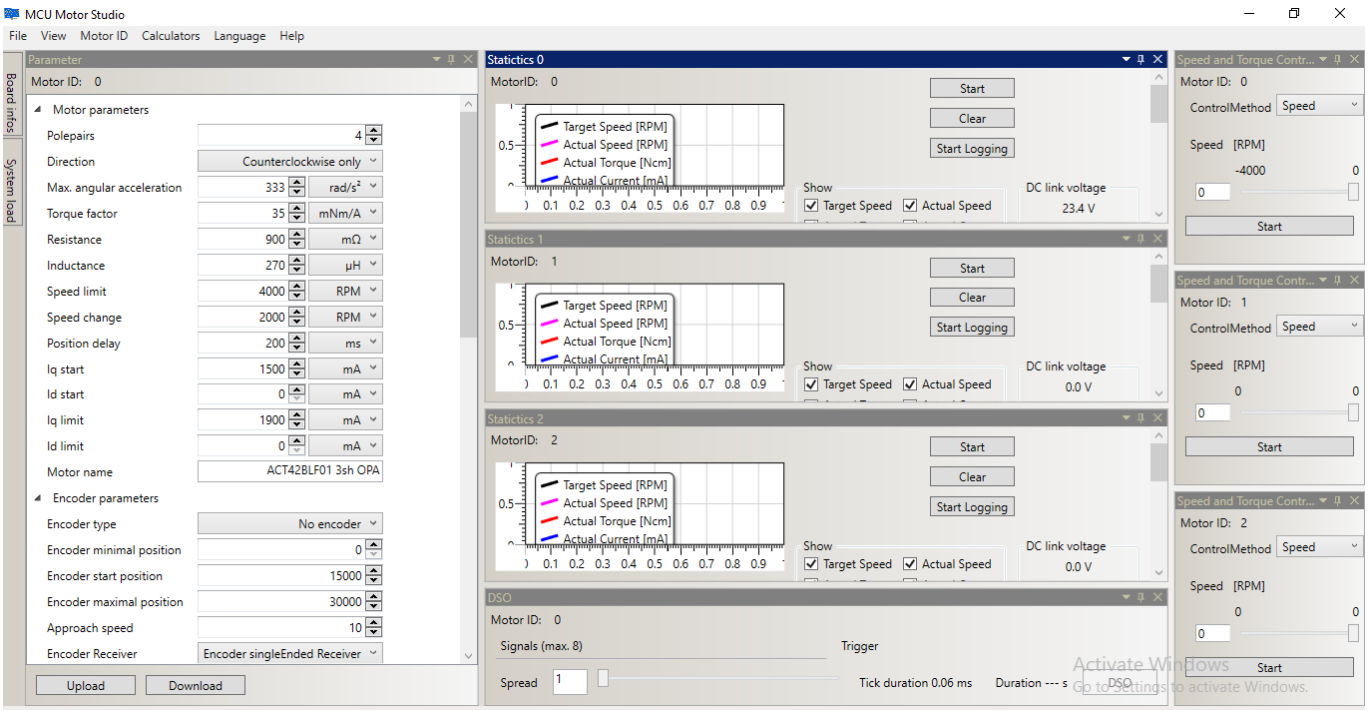


Figure 4-1 Default window layout

The current (working) layout can be changed and adopted to the actual needs or even extended with

more control windows to the full convenience of the user. Any changes in the layout are temporary and valid for the current working session only. Those can be made permanent, memorizing them via File->Layout->Save. In this case the last stored layout will be used during the next start-up sessions, until new one is memorized or the default one is recalled.

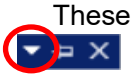
The default layout cannot be overwritten or altered in this or any other fashion. It can only be recalled at any time via File->Layout->Reset.

4.1.2. Layout Control

Every window has three distinctive pictograms located in the top right corner, as illustrated with the example below:

Figure 4-2 Default window layout

These have the following function:


drop-down arrow – select further visualization options Float, Show, Hide, Auto Hide or Dock


pin – to automatically dock the window on the left top side


x – to close/hide the window

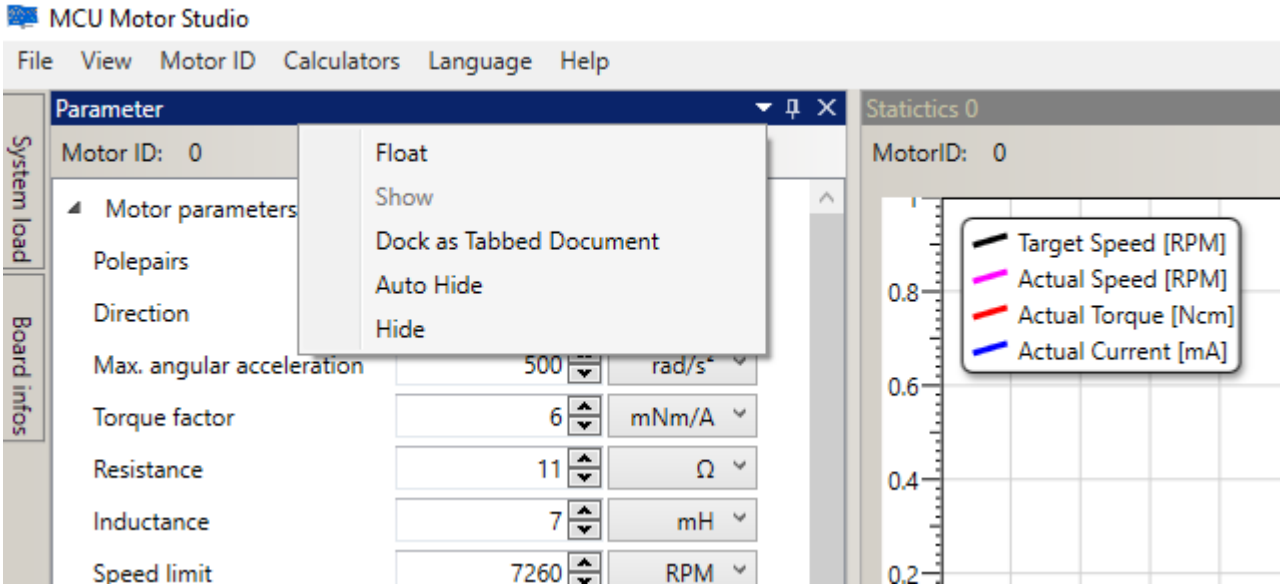


Figure 4-3 Layout control

Hidden Windows can be visualized again only via the **View-> Show** menu bar option.

4.1.2.1. Docking

The “Dock as Tabbed Document” feature is not usable at present. Please do not select this option.

4.1.2.2. Hide/Show

All available window/controls can be made invisible for the current session or view. These can be recovered at any point of time, restoring their default state and not the last used one. The Hide/Show option is available per Window.
The auto hide option allows the windows to be tabbed on the left side and only made visible when the user hovers the mouse pointer over the corresponding tab.

4.1.2.3. Float

Selecting the Float visualization option will place the window on top of all others brining it in the middle on focus. All windows below are still accessible and can be used.
This option is typically used for re-arrangements or for ease of a particular control as illustrated with the sequencer window below:

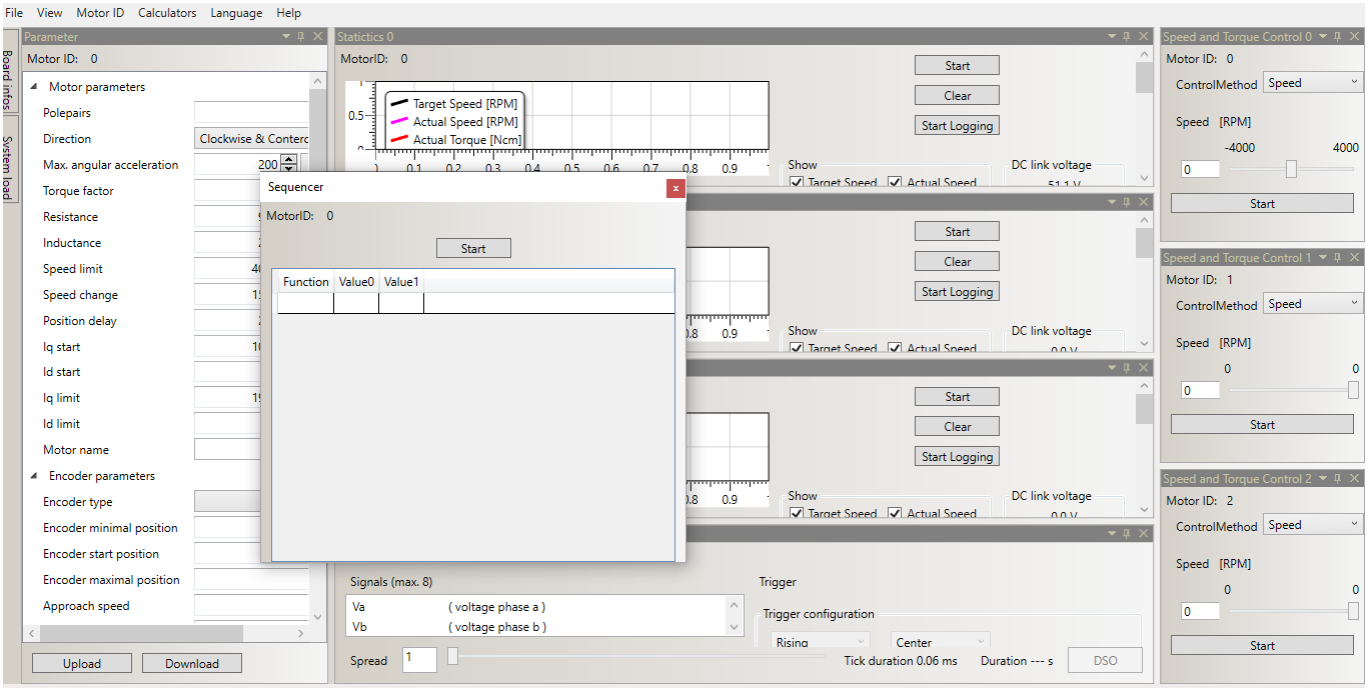


Figure 4-4 Sequencer window in float visualization mode

4.1.2.4. Re-arrange

Once a control window is put in float mode it can be docked at the left, top, right and bottom side of the main window as indicated by the single arrows:

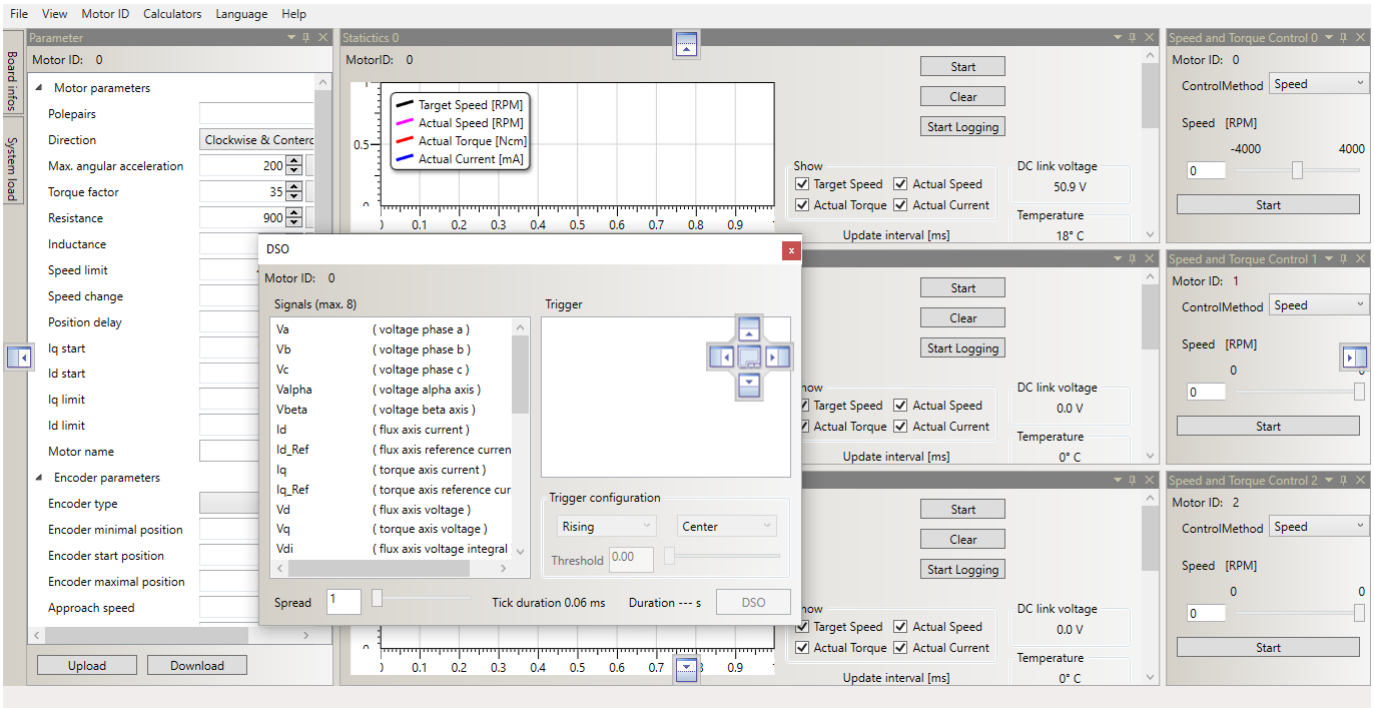


Figure 4-5 Re-arrange control window in the main window

Each arrow indicates the new position the window will be docked to.

The square in the middle permits tabbing of the floating window to the one on the back, Statistics 1 in the case illustrated above.

The floating control window can be docked at the left, top, right and bottom side of any other control window as well:

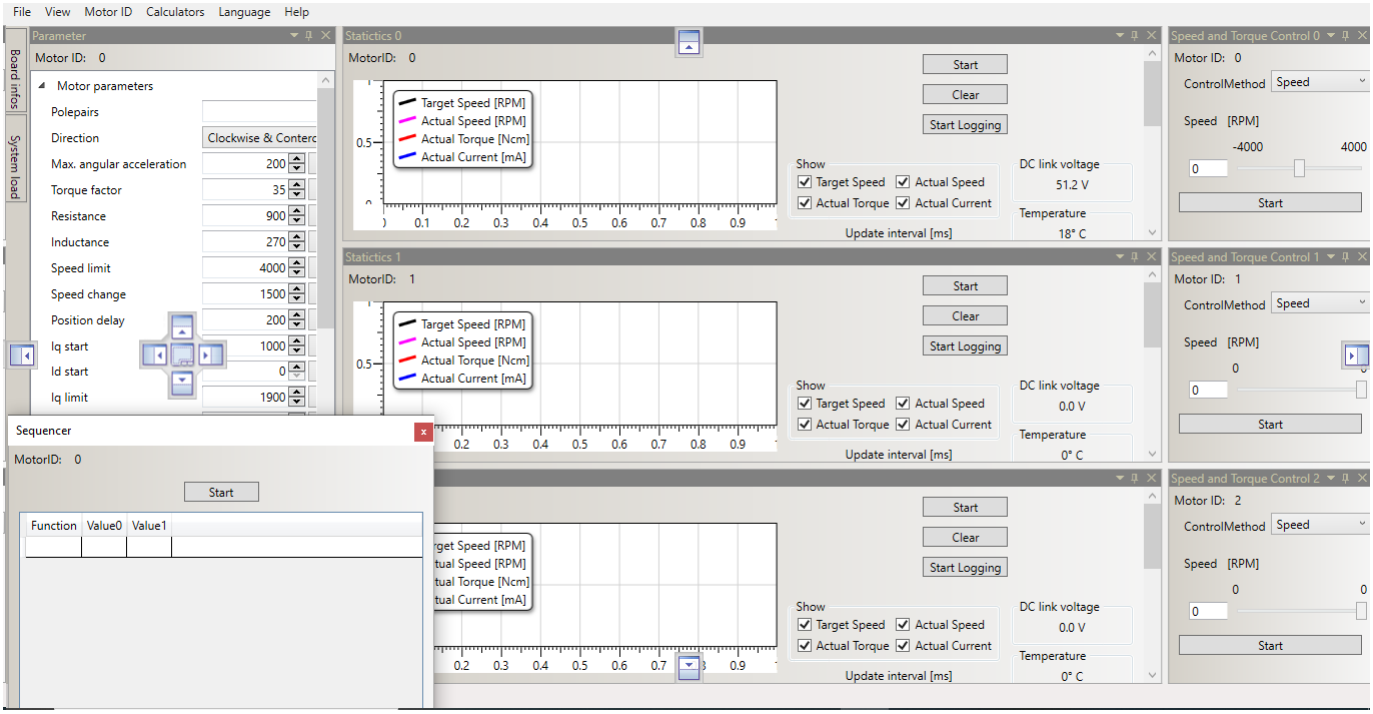


Figure 4-6 Re-arrange control window in another control window

A typical arraignment for single channel control with Statistics and DSO enabled would look like:

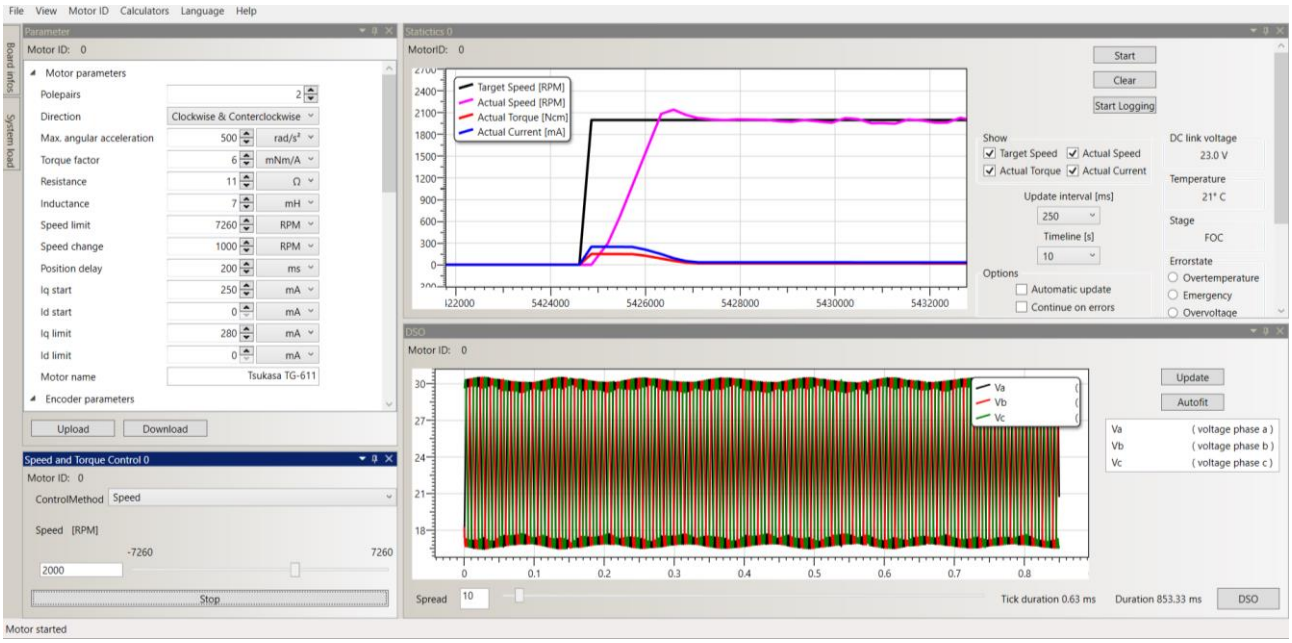


Figure 4-7 User specific re-arranged layout

4.1.2.5. Tab

Any floating control Window can be tabbed to another one using the middle square.

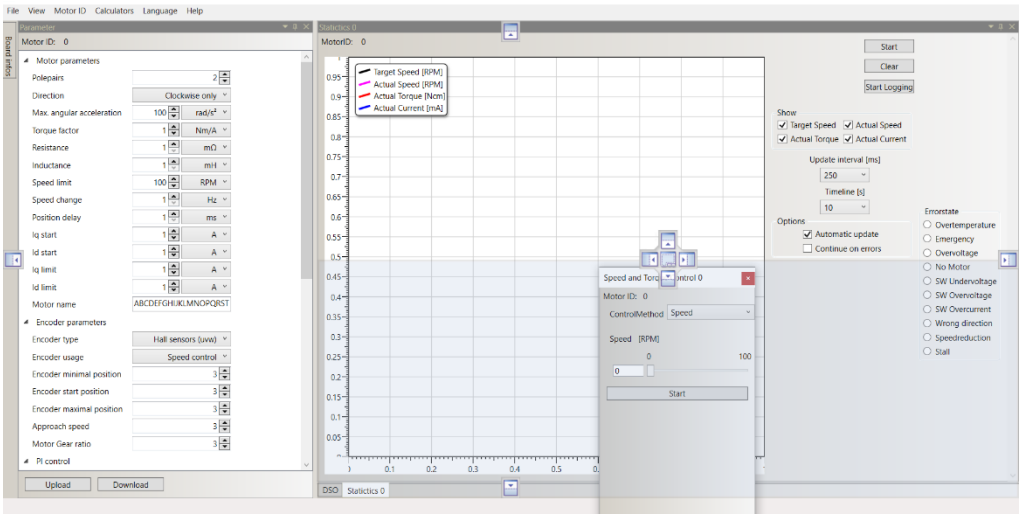


Figure 4-8 Tab a control window

4.1.3. Menu Bar

Most of the configuration and utility features of the Tool are available via the Menu Bar. Details of all options are listed below.

4.1.3.1. File

The **File** menu gives the user a possibility to save, load, clear or export various configurations including layout, set of parameters, execution sequence and is some cases recall the default ones.
If implemented for the particular platform, the current set of parameters may be stored to or read from the internal non-volatile memory.

4.1.3.1.1. Layout

Two options are possible:

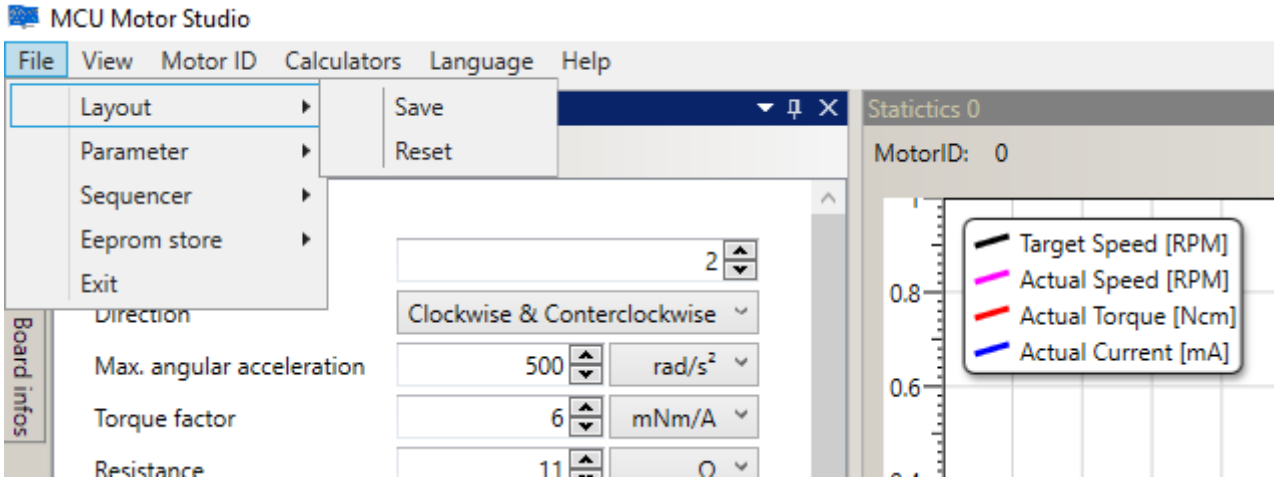


Figure 4-9 File->Layout menu

Save – the current working layout will be memorized and used until it is changed again. All modifications that are not saved will be lost upon next start-up of “MCU Motor Studio”

Reset – recall the default layout

4.1.3.1.2. Parameter

There are three options available:

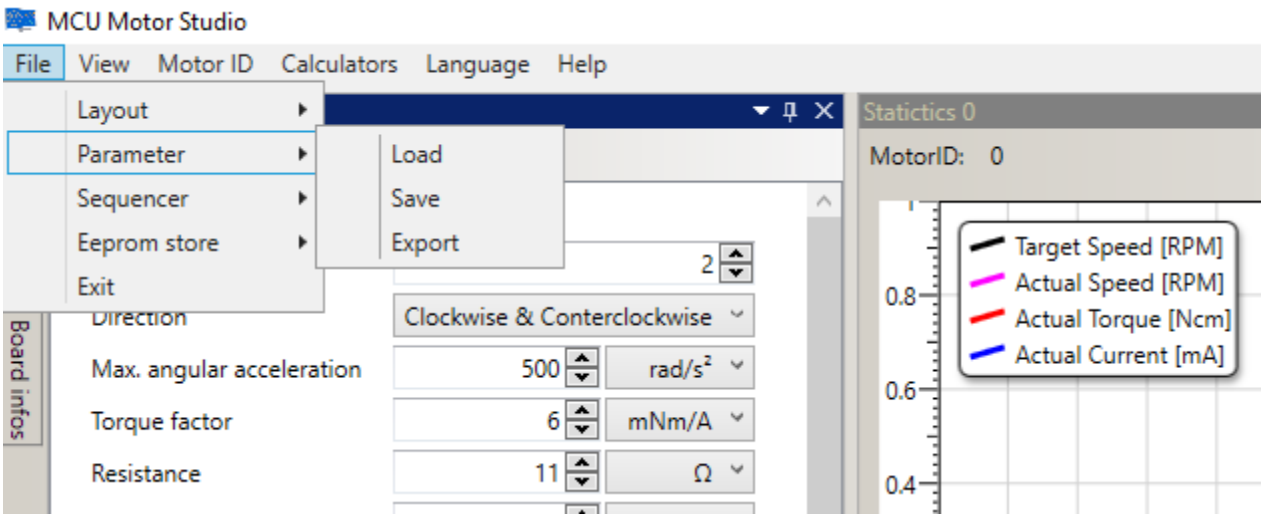


Figure 4-10 File->Parameter menu

Load – opens a channel parameter set that was previously generated using the **Parameter->Save** option. The file format is XML and the **Upload** button needs to be hit so that the newly loaded configuration, visible in the **Parameter** window is uploaded to the target MCU. The default file location is C:\Users\userName\AppData\Roaming\Toshiba Electronic Devices & Storage Corporation\MCUMotorStudio, where UserName is the windows® account used for the installation. Please ensure sufficient access rights. The location can be freely selected. “MCU Motor Studio” does not remember the last used path though:

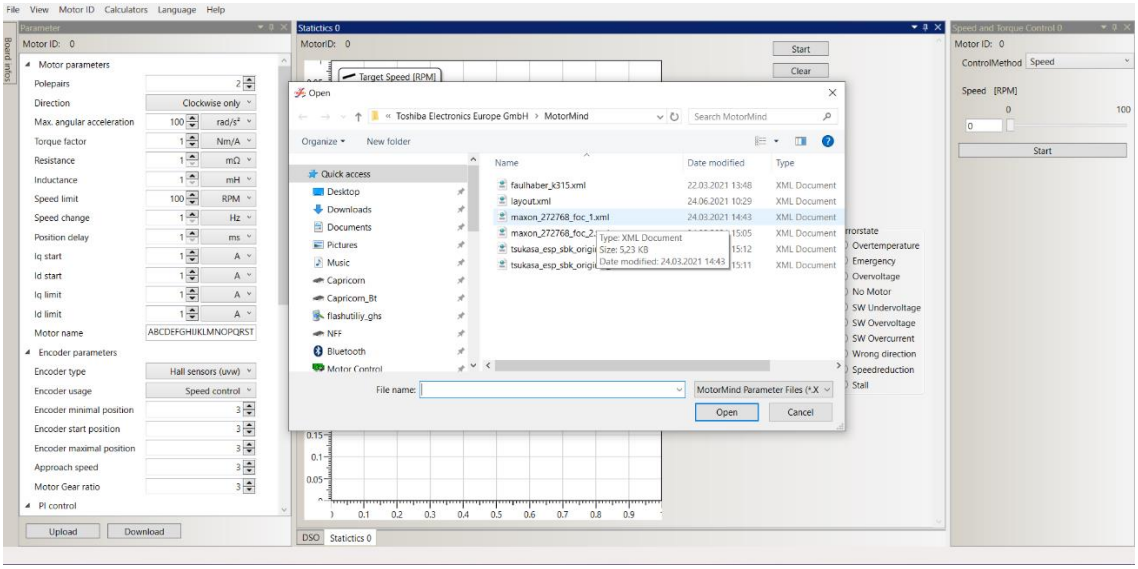


Figure 4-11 Open parameter configuration file

Save – the current set of parameters will be stored for later usage. The default location is C:\Users\ userName\AppData\Roaming\Toshiba Electronic Devices & Storage Corporation\MCUMotorStudio where UserName is the windows® account used for the installation. Please ensure sufficient access rights. The location can be freely selected. Please note that “MCU Motor Studio” does not remember the last used path:

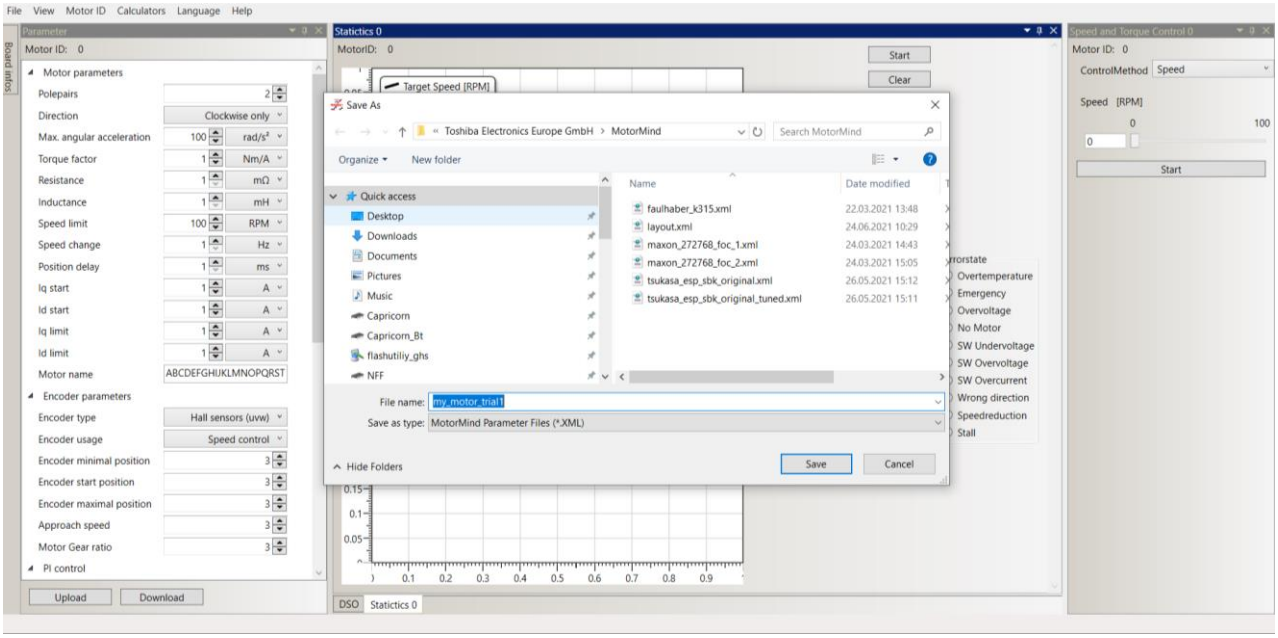


Figure 4-12 Save parameter configuration file

Export – (in future releases) the current set of parameters will be exported to a C header file that can be directly included into the Motor Control Firmware and used as compiled-in parameters for the selected channel. Please note that the motor name may not exceed 20 characters and is by default made part of the header file name. You are free to change the name according to the needs. The default export location is C:\Users\userName\AppData\Roaming\Toshiba Electronic Devices & Storage Corporation\MCUMotorStudio where UserName is the windows® account used for the installation. Please ensure sufficient access rights. The export location can be freely selected. “MCU Motor Studio” does not remember the last used path:

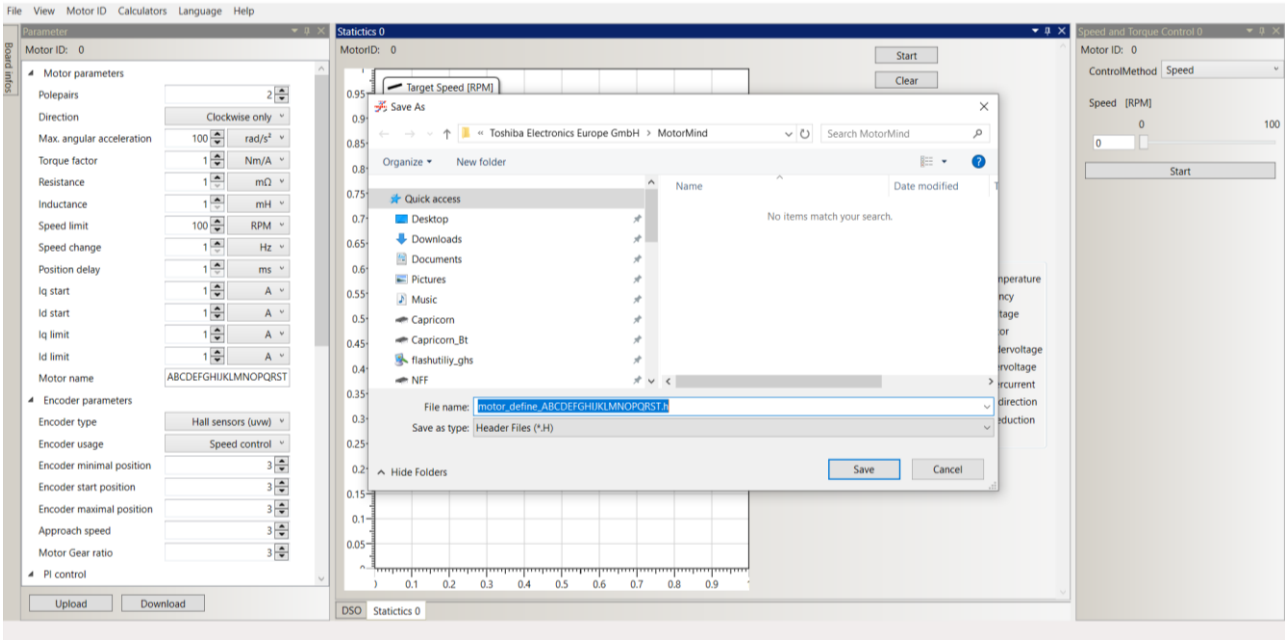


Figure 4-13 Export parameter configuration file

Please note that the exported file format, dependent on the “MCU Motor Studio” version, will not be compliant with all Motor Control Firmware versions, as the set of parameters has evolved over time. In such case minimal manual correction will be required.
Please consult the Toshiba Motor Control Firmware User’s Manual for details on the parameter header files, their format and inclusion options.

4.1.3.1.3. Sequencer

There are three options available:

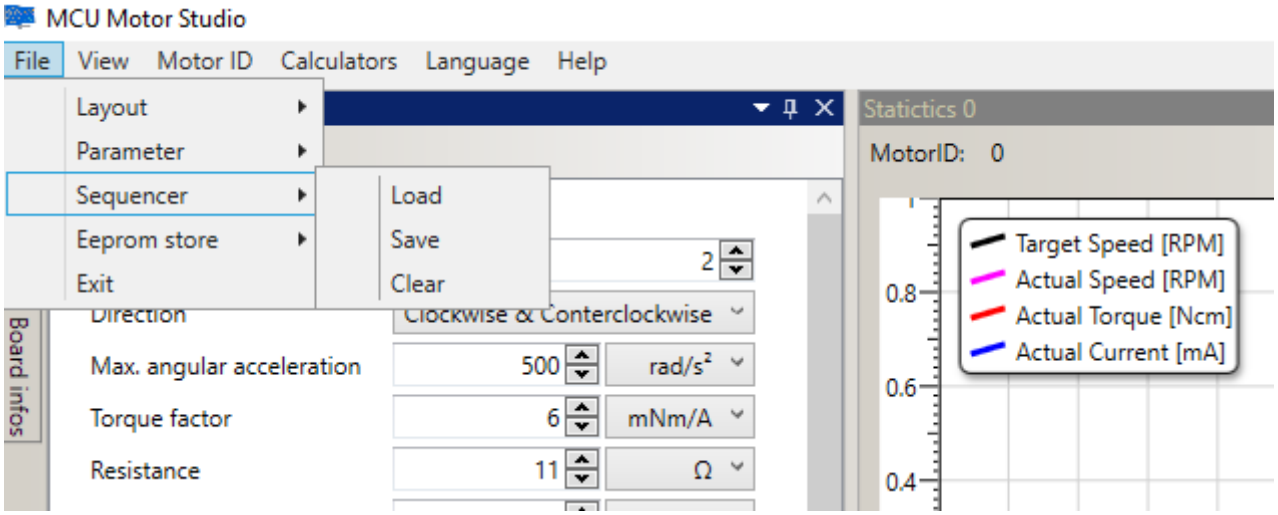


Figure 4-14 File->Sequencer menu

Load – load a previously generated and saved sequence from a XML file. The default location is C:\Users\UserName\AppData\Roaming\Toshiba Electronics Europe GmbH\"MCU Motor Studio"\, where UserName is the windows® account used for the installation. Any other path can be chosen in-stead.

Save – store the currently generated/used sequence to an XML file for later usage. The default saving location is C:\Users\UserName\AppData\Roaming\Toshiba Electronics Europe GmbH\"MCU Motor Studio" \, where UserName is the windows® account used for the installation. Please ensure sufficient access rights. The saving location can be freely selected. "MCU Motor Studio" does not remember the last used path:

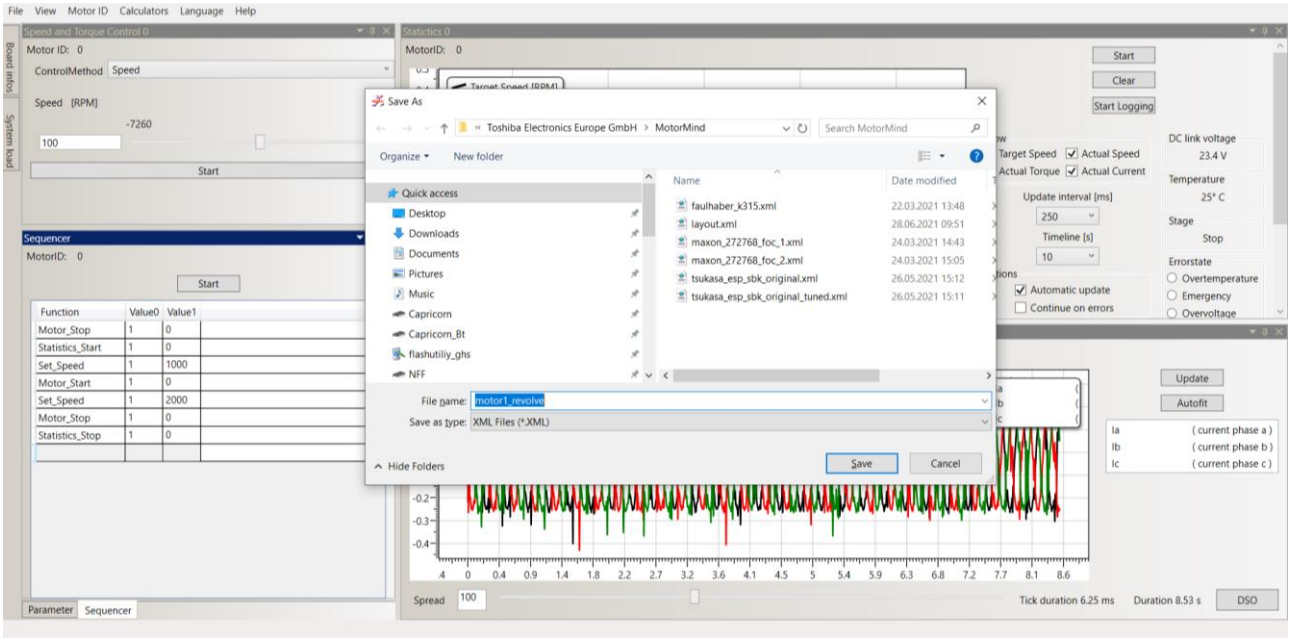


Figure 4-15 Sequencer Load/Save operation

Clear – empty the current sequence list. All unsaved changes will be lost:

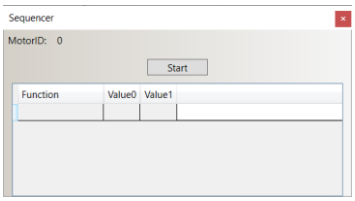


Figure 4-16 Empty sequencer list

4.1.3.1.4. EEPROM Store (in future releases)

There are two possible commands:

Store – write the current set of parameters to the non-volatile memory of the MCU, only if EEPROM emulation is implemented for that particular family/member.

Clear – erase the current set of parameters in the non-volatile memory. At next start-up the Firmware will recognize that the storage is empty and will use the compiled-in parameters, if such are available:

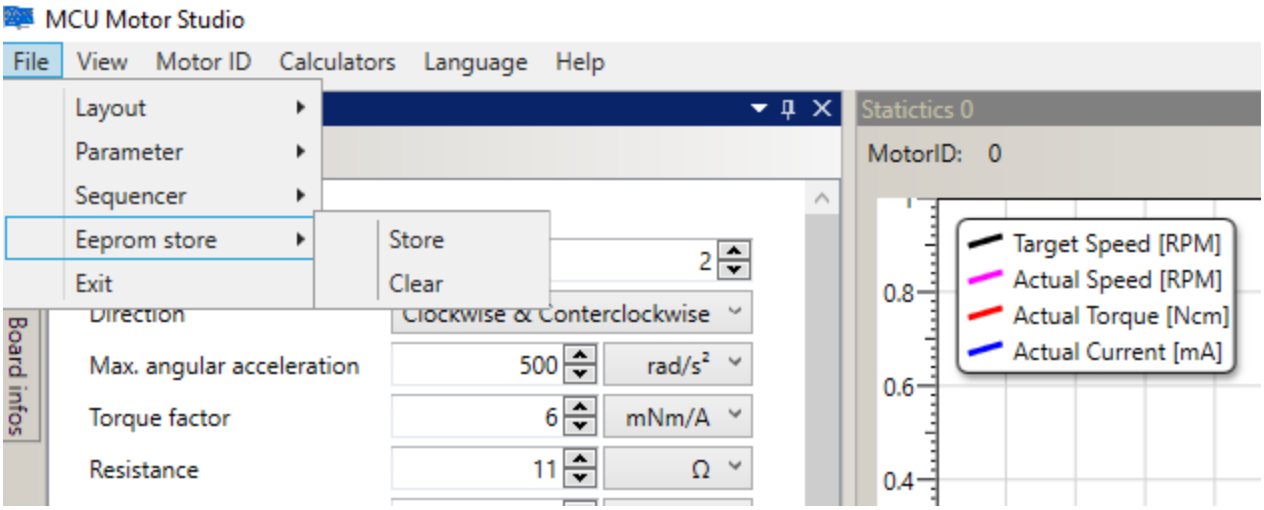


Figure 4-17 File->EEPROM store menu

4.1.3.2. View

The View menu gives the user a possibility to inspect the list of available control windows that are not currently visible and visualize the needed ones, selecting them one by one from the list. The number and type of control windows available in “MCU Motor Studio” and this menu respectively is only dependent on the features compiled in and available on the platform currently connected to the Tool:

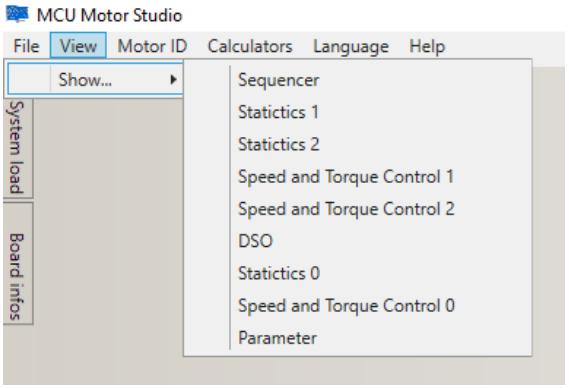


Figure 4-18 View menu

4.1.3.3. Motor ID

The Motor ID menu gives the user a possibility to select the active motor channel. Parameter configuration, DSO/HS-DSO, Sequencer and all file operations are bond to and can only be performed to the currently selected (active) motor channel.

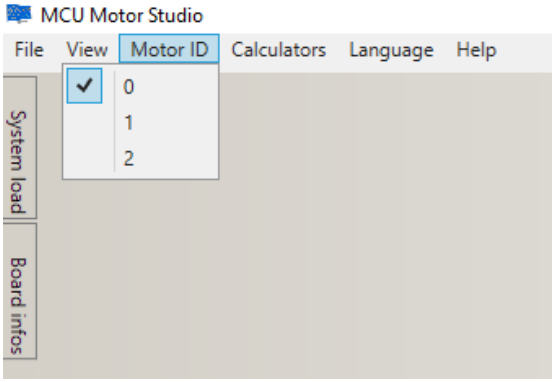


Figure 4-19 Motor ID menu

Speed/Torque Control and Statistics is available simultaneously for all the supported channels. The number of motor channels available in “MCU Motor Studio” and this menu respectively is only dependent on the number of such available on the platform currently connected to the Tool:
Please select only Motor ID “0”.

4.1.3.4. Calculators

The **Calculators** offers a few handy tools to support the proper configuration of some parameter for the used motors and power boards.

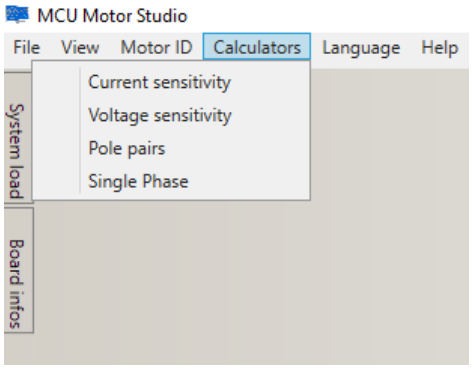


Figure 4-20 Calculators menu

4.1.3.4.1. Current sensitivity

Regardless of the used current measurement technique the sensitivity of the used circuitry, a typical example presented in the calculator itself, it is a very important parameter that will allow the Motor Control Firmware to properly interpret the results of the analog to digital conversion of the phase currents.
Simply input the values of the shunt and the resistors used with the operational amplifier and use the **Calculate** button to get the results displayed in the bottom of the window.

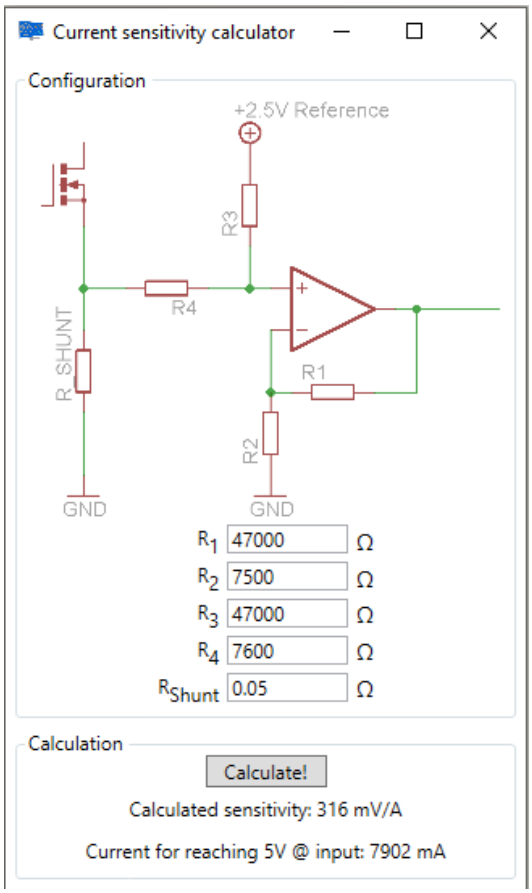


Figure 4-21 Current measurement sensitivity calculator

4.1.3.4.2. Voltage sensitivity

Similarly, to the current measurement, the sensitivity of the DC link voltage measurement circuitry, again a typical example depicted in the calculator itself, it is a very important parameter that will allow the Motor Control Firmware to properly interpret the results of the analog to digital conversion.

Simply input the values of the resistors building the voltage divider as used with the operational amplifier and hit the **Calculate** button to get the results displayed in the bottom of the window.

Besides the sensitivity, the maximum measurable voltage will be automatically calculated, even if this might not be achievable on the particular power board.

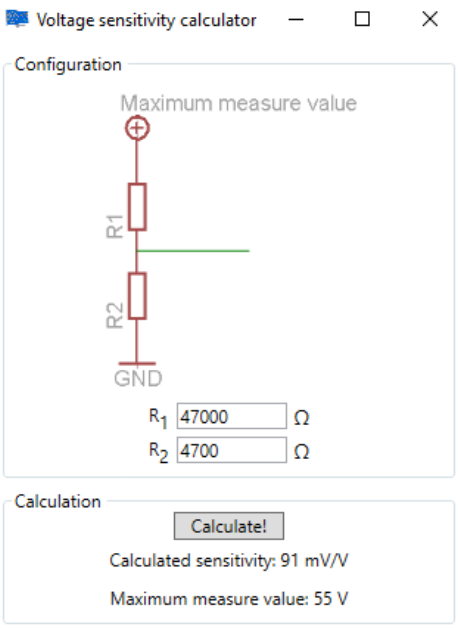


Figure 4-22 Voltage measurement sensitivity calculator

4.1.3.4.3. Pole pairs

A very handy tool if the number of poles and pole pairs respectively is not known/specified in the datasheet. Having the possibility to measure the actual speed with an angular acceleration measurement device, such as tachometer, together with the calculator will help determine the number of pole-pairs.

Select the active channel, start the motor with pre-defined speed, 1000 RPM for example. Measure the actual speed and input in the data.

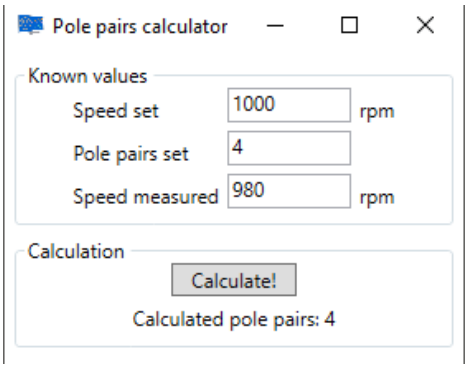


Figure 4-23 Pole-pairs calculator

4.1.3.4.4. Single Phase

Given only the line to line inductance and resistance are known, the calculator will evaluate the single phase parameters, considering the used winding connection (delta or star):

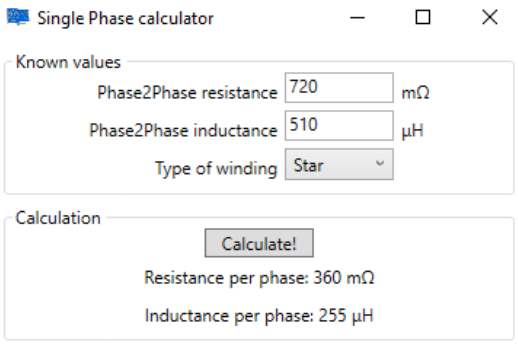


Figure 4-24 Single phase calculator

4.1.3.5. Language

The Language menu gives the user a possibility to select the tool language among a few supported ones. At present these are limited to English (default), Japanese (in future releases) and German. When Automatic option is selected the Tool will switch to currently configured Windows locale if supported or English in all other cases:

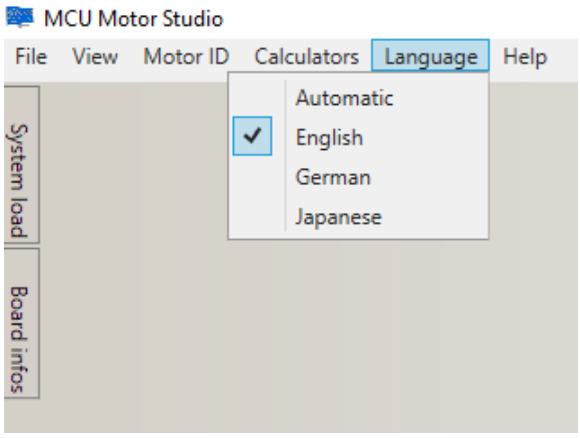


Figure 4-25 Language menu

There might be fields that are only available in English. It is mostly as the translation of internationally used engineering terms/units does not make sense. In some few cases the localization of the Tool was not yet updated.

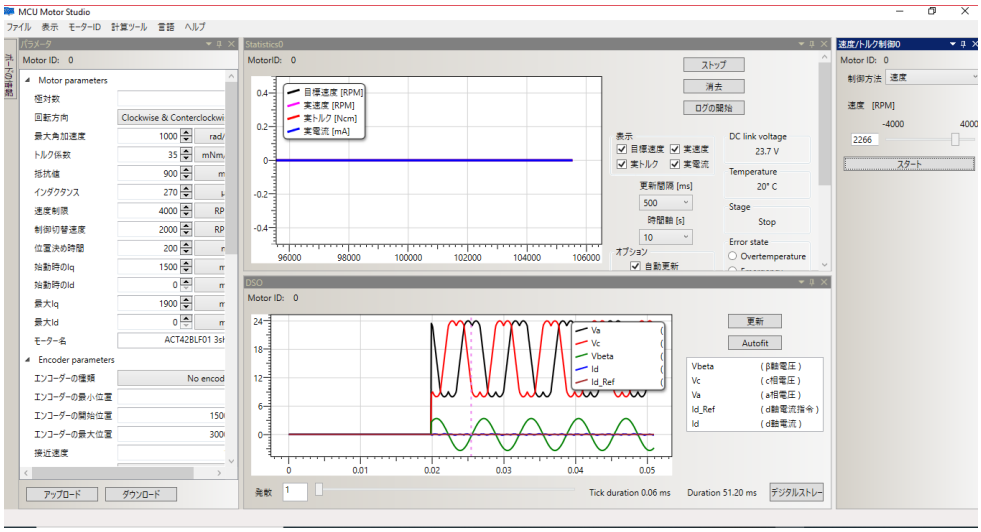


Figure 4-26 “MCU Motor Studio” with Japanese locale

4.1.3.6. Help

The help menu provides direct access to a support forum (currently inactive) and the Toshiba Semi-conductor and Storage website.

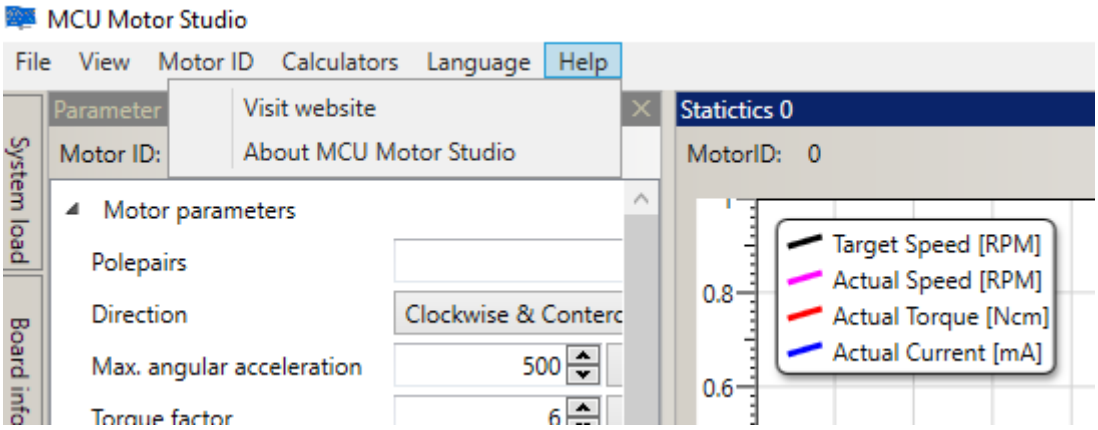


Figure 4-27 Help menu

The tool version and information of all used third-party components with their versions, license, web-sites and usage information is given in the standard About window.
Please kindly note that as the used components information is provided by third-parties, it might be out of date. Toshiba has no control over the correctness and cannot influence the regular updates:

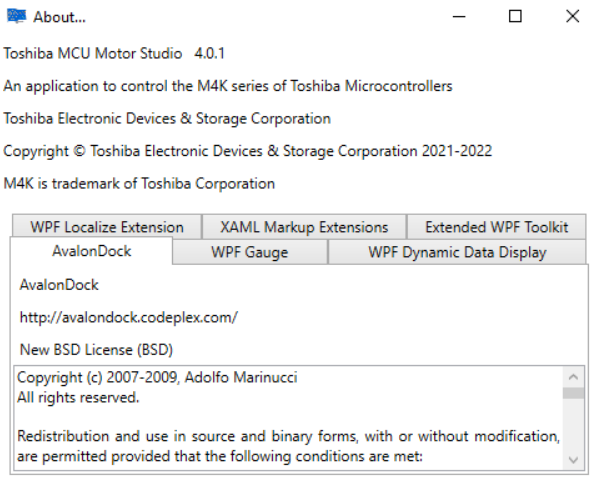


Figure 4-28 "MCU Motor Studio" About window

4.2. Parameter Window

The parameter window gives the full list of parameters as configured and used for the active motor channel and reported by the connected platform/executed firmware configuration:

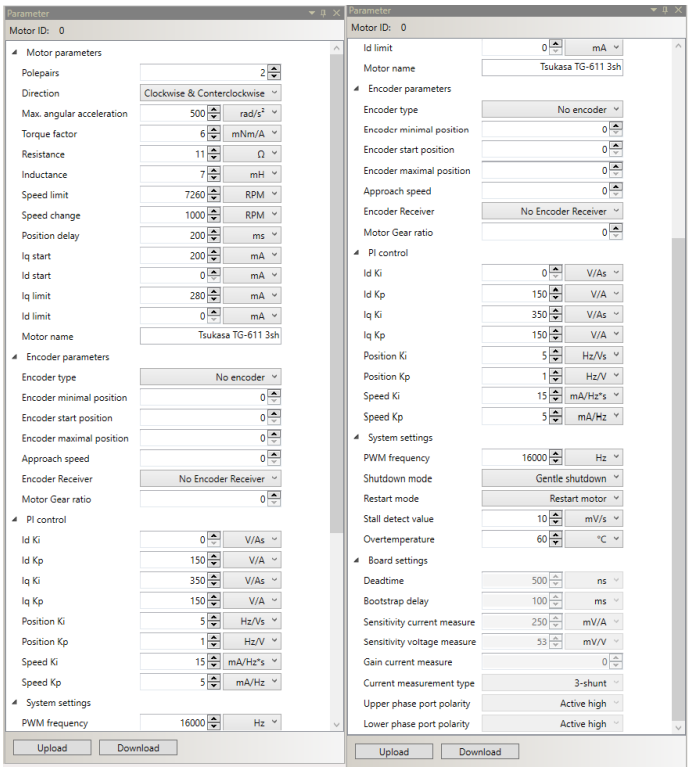


Figure 4-29 Parameter windows scroll up/scroll down

All parameters are grouped by functionality and are detailed bellow.
Most of them can be changed while the motor is revolving. The new values will be internally applied at strictly specified points of time/execution. Some like the Speed change and Position delay will only take effect after a stop and restart of the motor.

Motor parameters – a number of BLDC/PMSM motor characteristics need to be specified for each channel that is to be used. Typically these can be directly found in the datasheet, some like the inductance may be measured, others like the number of pole pairs can be calculated. For each parameter the value and the unit may be specified. A complete list is provided in the table below:

Parameter	Description	Supported Units
Polepairs	Number of poles divided by 2	-
Direction	Supported direction of rotation – counter-clockwise, clock-wise or both	-
Maximal angular acceleration	Specifies both the maximal allowed acceleration and de-acceleration of the motor	rad/s2 Hz/s RPM/s RPM/min
Torque factor	The torque constant of the motor, representing the actual slope of its torque / current curve	mNm/A Nm/A mNm/mA
Resistance	Phase resistance	Ω mΩ μΩ
Inductance	Phase inductance	mH μH nH
Speed limit	Rated (maximal) angular speed	Hz (electrical) RPM (mechanical)

Speed change	Minimal angular speed at which the firmware shall change from Force into FOC mode. Keep above the rated one to always stay in Force mode. The firmware uses a 5% hysteresis for the change FOC to Force.	Hz (electrical) RPM (mechanical)
Position delay	The minimal wait time needed for the initial rotor positioning, as long as no position detection is utilized	Ms
Iqstart	Torque-axis current to be used in Force mode	A mA
Idstart	Direct-axis current to be used in Force mode. Please keep 0 for BLDC motors	A mA
Iqlimit	Maximal allowed torque-axis current	A mA
Idlimit	Maximal allowed direct-axis current. Please keep 0 for BLDC motors	A mA
Motor Name	Up to 20 characters used as identifier of the motor	-

Table 4.1 Motor Parameters List

Encoder parameters – specify the type of used external senor, the intended usage, its resolution and the reduction factor (if equipped with a gear). A relative position, limits and desired approach speed shall be specified regardless whether linear motion control will be used or not. A complete list is pro-vided in the table below:

Parameter	Description	Supported Values
Encoder Type	The type of used external sensor	No sensor Hall sensor (uvw) Hall sensor (uv) Incremental encoder (abz) Incremental encoder (ab) Single pulse AMS – AS5145A
Encoder minimal position	The minimal relative position that can be driven to in Linear Motion Control	0 ~ 1000000
Encoder start position	The relative start position for Linear Motion Control	0 ~ 1000000
Encoder maximal position	The minimal relative position that can be driven to in Linear Motion Control	0 ~ 1000000
Approach speed	The maximal speed to be used for the final approach when positioning in Linear Motion Control	0 ~ 100000
Encoder Receiver		No Encoder Receiver Encoder Resolver Receiver Encoder Differential Receiver Encoder singleEnded Receiver
Motor Gear ratio	The reduction factor of the gear attached to the motor. Please note that increase factor is not supported at present and might be added later	0 ~ 10000

Table 4.2 Encoder Parameters List

PI control parameters – specify the configuration parameters of the proportional-integral regulation. A complete list is provided in the table below:

Parameter	Description	Supported Units
Id Ki	Integral coefficient for the direct-axis current regulation	V/As
Id Kp	Proportional coefficient for the direct-axis current regulation	V/A
Iq Ki	Integral coefficient for the torque-axis current regulation	V/As
Iq Kp	Proportional coefficient for the torque-axis current regulation	V/A
Position Ki	Integral coefficient for the position regulation	Hz/Vs
Position Kp	Proportional coefficient for the position regulation	Hz/V
Speed Ki	Integral coefficient for the speed regulation	mA/Hzs
Speed Kp	Proportional coefficient for the speed regulation	mA/Hz

Table 4.3 PI Control Parameters List

System settings – specify various system characteristics, depending on the connected platform. The PWM frequency and the stop mode are always available:

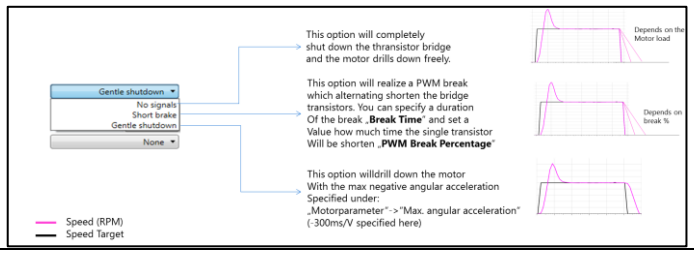
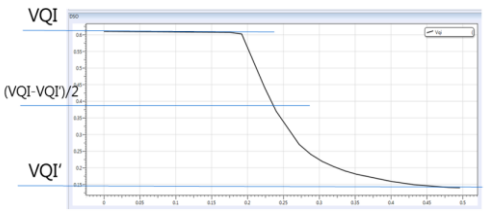
Parameter	Description	Supported Units
PWM Frequency	The rate of current control, angle lag compensation and PWM output change in kHz. Recommended start value is 16kHz. Lower update rate, already below 12kHz may lead to unwanted noise building. A higher update rate may result switching loses.	1 ~ 50 kHz
Shut down mode	Specify the desired motor control stop mode: Gentle break - fully controlled reduction of the speed with maximum angular de-acceleration. Short breaking mode - the Firmware alternates short the upper/lower phases of the motor. Self-break mode - no control is applied and the motor is stopping on its own. 	-

Table 4.4 System Settings List

The following system parameters are visible only if the specified options are configured in the firmware on the connected platform or certain mode is selected:

Parameter	Description	Supported Units
Restart Mode	Available only when **Stall Detect** is enabled. Specifies the system behavior upon field stall detection. Selectable are attempt to restart the motor or the more conservative option to stop it.	Switch off motor Restart motor
Stall detect value	Available only when **Stall Detect** is enabled. Specifies the minimal drop of the $\mathbf{V_{qi}}$ in **mV/s** that shall be treated as field stall. A good starting value is illustrated below, marked as ($\mathbf{V_{qi}}$-$\mathbf{V_{qi}'}$)/2., typically it shall be set to around a 90% drop from the normal value: 	0 ~ 100000
Brake time	Only if **Short brake mode** is selected. Specifies the maximum time to stop the motor in **ms**.	0 ~ 10000
PWM brake percentage	Only if **Short brake mode** is selected. Specifies the percentage of the time to	0 ~ 100%

	short brake the upper or lower side.	
Over temperature (in future release)	Only when **Temperature control** is enabled. Specifies the temperature in degrees centigrade at which the motor shall be stopped to avoid damages to the MOSFETs.	-40 ~ 150°C
Software overvoltage	Available only when **Software over- and undervoltage detection** is enabled. The DC link voltage at which the protection shall be enforced.	V
Software undervoltage	Available only when **Software over- and undervoltage detection** is enabled. The DC link voltage at which the protection shall be enforced.	V
Software overcurrent	Available only when **Software overcurrent detection** is enabled. The phase current at which the protection shall be enforced.	mA/A
Speed reduction percentage	Only when **Load dependent speed reduction** is enabled. The desired speed reduction in percentage of the set one.	%
External Speed Control	Only when **External Speed Control** is supported (at present only on **Sigma Board**). Specifies the inputs for the basic direction and speed commands and their parameters if any.	None External voltage External PWM signal External servo pulse
CAN ID	Only if **CAN** is enabled. The unique CAN 2.0A / CAN 2.0B device identifier for the bus communication	11- or 29-bit device identifier

Table 4.5 Optional System Settings List

Most of the optional settings are available for the firmware build presented below. Typically, not all system related features/settings will be enabled/visible at once:

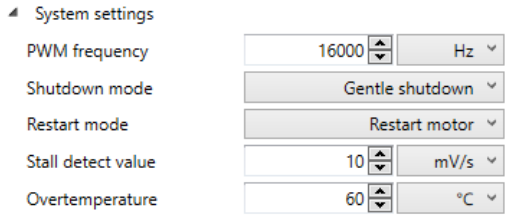


Figure 4-30 Optional System Settings

Board settings – specify various electrical parameters of the MOSFETs and the measurement circuitry on the power board. This configuration is typically static, marked as read-only and therefore no changes are permitted via the Tool. All read-only fields are visualized by a gray background. In some special cases the platform may allow changes via “MCU Motor Studio”, in that case the firmware is complied with the USE_RW_BOARD_SETTINGS. It is highly advisable to disable it again, once the proper settings are found and fixed. A complete list is provided in the table below:

Parameter	Description	Supported Units
Deadtime	The delay/time to ensure the FET is turned off	ns
Bootstrap delay	The time needed to charge the bootstrap capacitors	ms
Sensitivity current measurement	The sensitivity of the DC link	mV/A

	voltage measurement circuitry	µV/A
Sensitivity voltage measurement	The sensitivity of the phase current measurement circuitry, specified for all phase at once. Phase differences are not supported.	mV/V
Gain current measurement	An additional gain specified in values that a MCU specific and map to pre-defined and fixed amplification factor	0 - 10
Current measurement type	The type of phase current sensing	1-shunt 2-sensor (in future releases) 3-shunt
Upper phase port polarity	The active (turn on) polarity of the used transistors in the upper side of the H-Bridge	Active low Active high
Lower phase port polarity	The active (turn on) polarity of the used transistors in the lower side of the H-Bridge	Active low Active high

Table 4.6 Board Settings List

Any changes of a parameter value and/or unit will be indicated by coloring the parameter name in red as illustrated below:

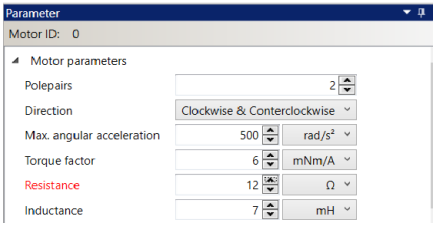


Figure 4-31 Changed parameter indication

There are two buttons to apply changes:
Upload – will send (upload) all changed parameters to the connected target, making them the used one for the current session. These will be lost if the platform is powered off.

Download – read out the current platform configuration and update the values/units in the parameter window of the Tool. Please note that unlike the one altered with “MCU Motor Studio”, any changes that were taken over will not be indicated using different color.

4.3. Board Information Window

The board information window gives a comprehensive information about the currently connected platform grouped in Device Information, Firmware Features and Highspeed-DSO:

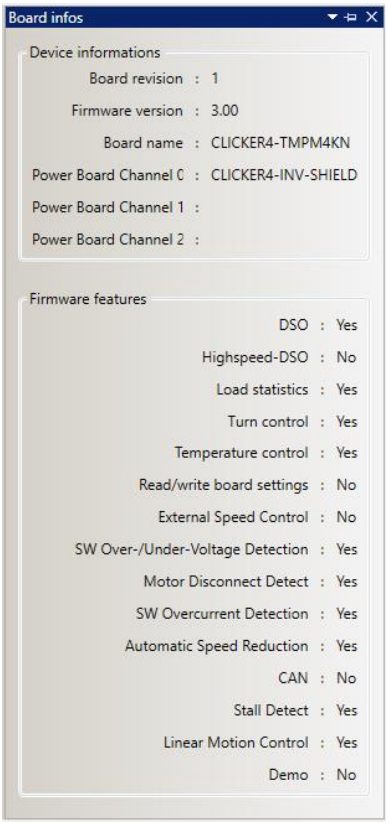


Figure 4-32 Board information window

Device Information – includes the actual firmware version and the hardware platform configuration in terms of main and power boards connected.

Firmware Features – a full list of all features that a firmware may support with an indication of which are currently configured in/enabled. Not all platforms and MCUs can support all of the features in the list.

Highspeed-DSO – support for a real-time, high-speed logging of parameters via dedicated hardware adapter

4.4. Speed & Torque Control Channel 0/1/2 Window

As the most recent Motor Control Firmware versions feature a simultaneous control of up to three channels, the Tool was extended (from version 3.0.0.x onwards) to provide individual Speed/Torque Control & Statistics windows for every single channel. It eases the usage by eliminating the need of selecting/switching the active channel and allows fast stop/start and speed/torque change as all controls are immediately available.
These windows can be individually shown, hidden, re-arranged, re-sized and used. Only one can be at focus at a time though. The channel it is controlling is indicated by the number in the title bar, counting starting at 0 and going up to 2:

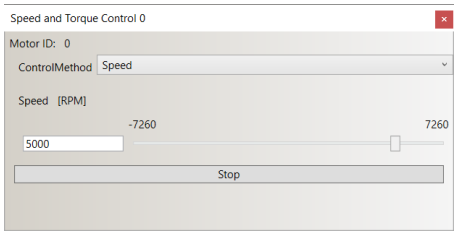


Figure 4-33 Speed control window channel 0

The ControlMethod drop-down is used to select the currently active control method. Two options are available – by Speed and by Torque. There is a slight difference in the look & feel as illustrated on Figure 4-33 and Figure 4-35. The Torque control method may be disabled in some firmware versions, thus unavailable for selection. Speed Control is always available.

The **Speed slidebar** indicates the speed limits in both directions, specified in RPM. It may be slid to select the desired speed and rotational direction. As soon as it is left in position, the new values will be read and transmitted to the target. The Speed textbox will be automatically updated giving the exact reading.

The **Speed textbox** can be used to directly input the desired speed and direction. As soon as enter is pressed or the focus is changed to any other element of the control window, the speed and direction will be read and transmitted to the target. The Speed slidebar will be automatically updated. A negative speed is treated as CCW direction with the absolute value of the speed for both input methods.

The **Start/Stop button** is used to start and stop the motor respectively. The value is automatically adjusted to the possible action, depending only on the current state of the motor. When a connection to the target platform is lost, the button may get out of synchronization. In case of Stop this may result the following error, which can be safely ignored (attempt to stop an already stopped motor shall naturally fail):

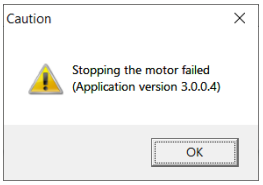


Figure 4-34 Motor Stop failure at lost synchronization

The Torque sidebar indicates the torque limit, specified in mNm. It can be slid to select the desired value. Naturally only positive numbers are accepted. As soon as it is left in position, the new torque will be read and transmitted to the target. The Torque textbox will be automatically updated giving the exact reading.

The MaxSpeed slidebar and MaxSpeed textbox have the same functionally as their Speed counterparts. The only difference is that the set speed in that case is not a strict request, but rather a desired value. The firmware will attempt to achieve and will reduce it, given the set torque cannot be maintained.

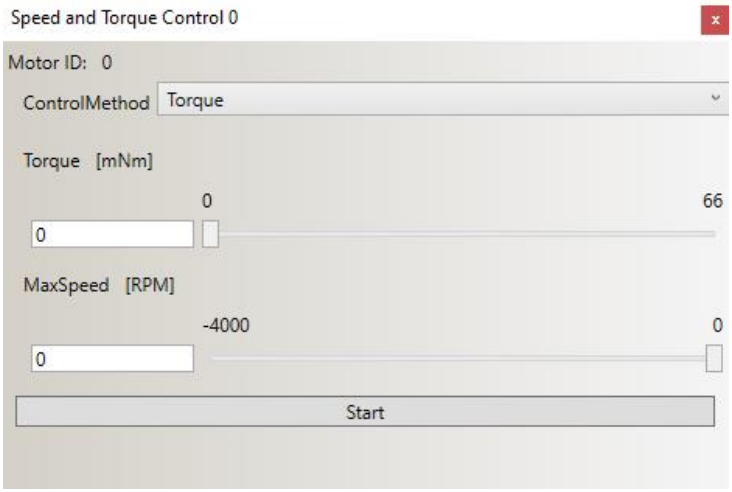


Figure 4-35 Torque control window channel 0

The Tool main window with Speed/Torque and Statistics windows for all three channels example is depicted below:

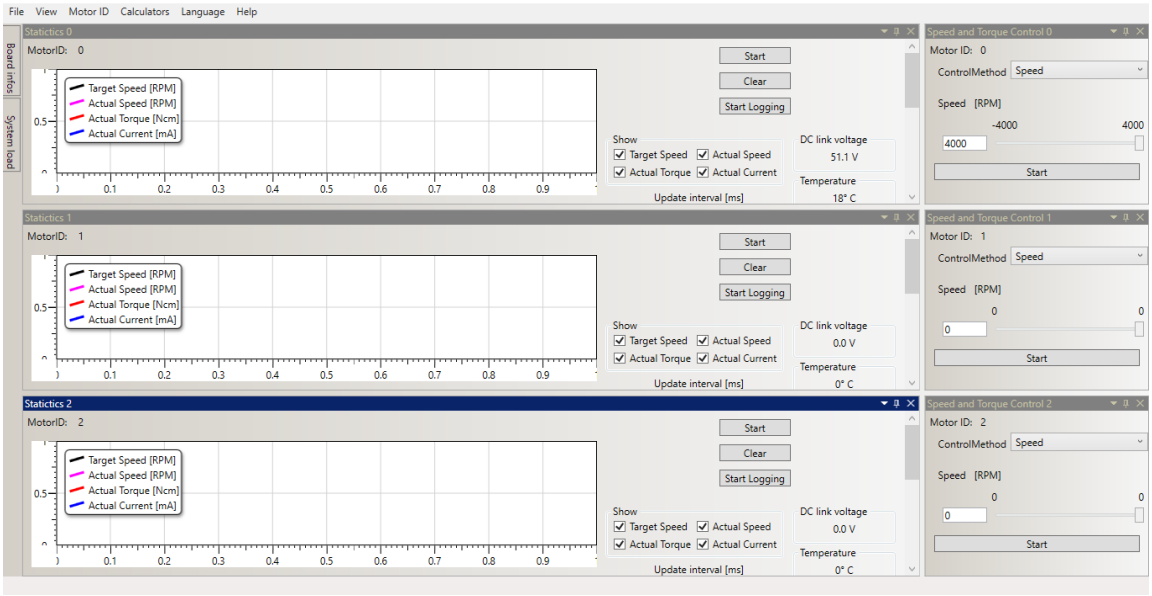


Figure 4-36 Three channel control layout example

The screenshot above was taken with 15.4” TFT monitor. It is strongly recommended to use at least 24” for better operation and visualization, especially in the cases all three channels have to be operated.

4.5. Statistics Channel 0/1/2 Window

For each channel there is a separate **Statistics** control window. It allows several actions and gives run-time information about the motor channel it is assigned to:

Parameter Chart – a fully scalable (automatic or manually) two-dimensional timing chart visualizing the run-time change of up to five control parameters. The number and exact parameter to be visualized/hidden is configurable via the Show tick-boxes with several limitations:

- **Timestamp** is always present and can’t be selected/de-selected.
- **Actual Current** is unconditionally available for selection/de-selection.
- The used **Control Method** determines the remaining three parameters that might be displayed.
- The **Color code** used to distinguish the individual lines of the chart and on the legend placed in the top left corner is automatically assigned.

Control By Speed	Control By Torque
Target Speed - set desired rotational speed in RPM	**Target Torque** - set desired torque in Ncm
Actual Speed – the actual rotational speed at the time of sampling	**Actual Torque** - the actual torque at the time of sampling
Actual Torque – the actual torque in Ncm	**Actual Spe**ed – the actual speed in RPM
Actual Current – the used q-axis current in mA	
Timestamp - a relative timestamp measure in number of system timer ticks	

Table 4.7 Statistics parameter

All four parameters may be hidden even it is not usable in that way. In some special diagnostics/trial versions of the firmware the meaning of these parameters may be overridden for test/tuning purposes.

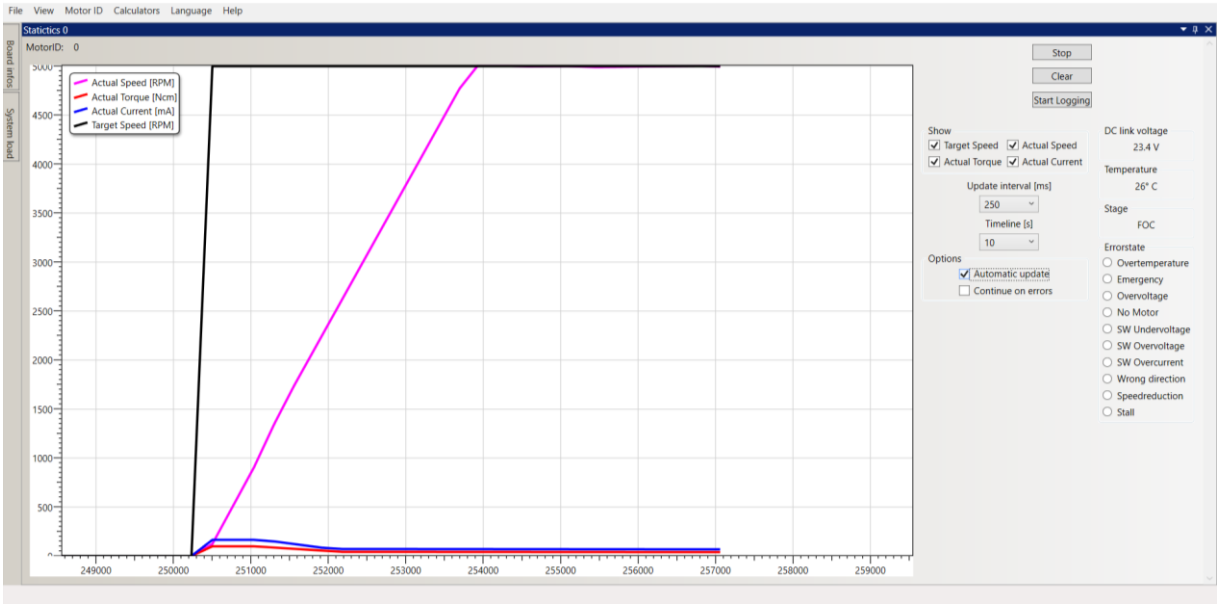


Figure 4-37 Statistics control window

The **Start/Stop** button is used to start and stop the collection of statistics information. The action value and the button label is automatically adjusted to the next possible action, depending only on the current state of the statistics collection, e.g. **Stop** is shown whenever the collection is started and ongoing.

The Clear button will wipe off all the drown charts. Please note that if done while still collecting, the drawing will immediately resume, so that some screen artefacts or undesired effects might appear.

The Start/Stop Logging button is used to start and respectively stop the optional logging of the selected parameters to a CSV file. The default saving location is C:\Users\userName\AppData\Roaming\Toshiba Electronic Devices & Storage Corporation\MCUMotorStudio, where UserName is the windows account used for the installation. Please ensure sufficient access rights. The saving location can be freely selected. “MCU Motor Studio” does not remember the last used path. The default log file name is statisctis_MotorName.csv, where MotorName is identifier of the motor for the associated channel. The example below is taken from the Demo mode, where the motor identifier is set to the illustrative string ABCDEFGHIJKLMNOPQRST:

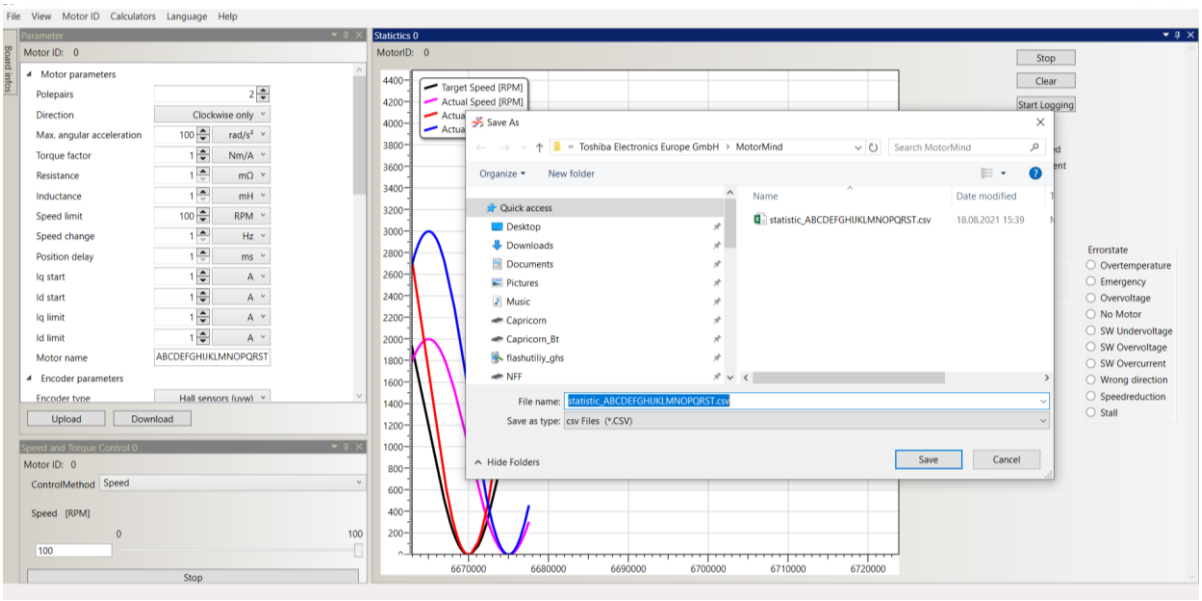


Figure 4-38 Logging dialog window

The **Show** group of tick boxes allows individual selection/de-selection of the control parameters to be shown/hidden on the chart and logged respectively. Untick to hide/stop logging.

The **Update interval [ms]** drop-down box is used to select the desired sampling interval among the supported 1, 50, 100, 250, 500 and 1000ms. Due to the limitation of the Window system timers, the 1ms/10ms sampling may not be precise and shall be avoided. A typical and recommended value is 250ms.

The **Timeline** drop-down box allows scaling of the Parameter Chart's x-axis. Selectable are 5, 10, 20, 30 and 60 seconds with 10 being the default and recommended value.

The **Option** group of tick boxes gives further control over the Parameter Chart update:

- **Automatic update** – the automatic chart scaling and shift may be overridden/set on hold. It allows the user to zoom or move the chart for detailed inspection of selected curve fragments, etc. In that case the box will be automatically unticked, indicating manual operation and all sampled values will not be shown. To resume the automatic update and bring the currently drawn values to the middle of the chart, simply select the option again.

The automatic update is disabled whenever the chart is set on focus and it is being:

- Zoomed in/out using the scroll wheel of the mouse.
- Moved left/right/up/down by pressing and holding the left mouse button (right for a left-handed configuration) and moving the mouse in the desired direction.
- Zoomed & moved.

The collection and the optional logging of control parameters is not affected by this option, as it only controls the visualization on the chart.

- **Continue on errors tick** allows further collection of control parameters even if an error (stall, overtemperature, over voltage, etc.) is detected. It might be useful for troubleshooting or in all cases where automatic restart is attempted. This option is disabled by default resulting automatic stop of further parameter collection and chart update.

The **DC link voltage** provides read-only information about the recent (last transmitted) measured power board DC voltage. On some versions of the firmware, mostly experimental or whenever the system is still tuned, this value may be clamped to a fixed voltage and will not be changing over time. On others it may not be available at all, as the underlying hardware may not feature means to measure it, including Demo mode. In the latest the label will not be shown at all, as illustrated on Figure 4-39. "n.a." will be displayed given the voltage measurement was not successful or "MCU Motor Studio" failed to obtain a value from the attached platform.

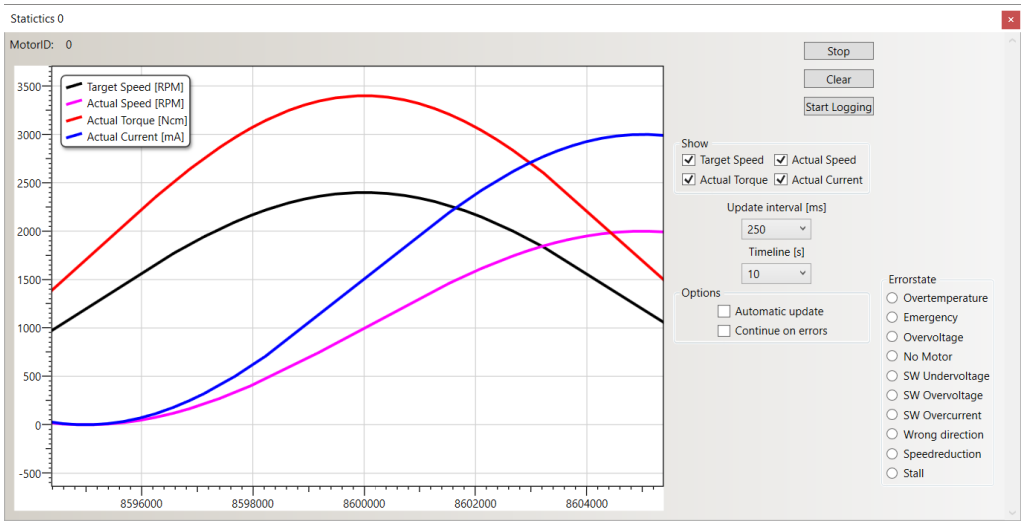


Figure 4-39 Statistics without DC Link voltage and Temperature indication (Demo mode)

The Temperature gives read-only information about the recent (last transmitted) measured temperature on the board, typically sensed closed to the MOSFETs. On some versions of the firmware it may not be available at all, as the underlying hardware does not feature means to measure. The value is also not available whenever “MCU Motor Studio” is executed in Demo mode. In both cases the label will not be shown at all. “n.a.” will be displayed given the temperature measurement was not successful or “MCU Motor Studio” failed to obtain a value from the attached platform.

The **Stage** indicates, as the name suggest, the current state of the motor as executed by the drive control. It may be one of **Stop, Emergency, Break, Bootstrap, Initposition, Force and FOC**. Please refer to **Toshiba Motor Control Firmware User’s Manual** for further details on the individual stages. The field is not available in Demo mode or when the connection to the target is not established or lost.

4.6. DSO Window

The Digital Storage Oscilloscope (DSO) is an optional feature that allows logging of various system, motor and control parameter during normal operation. The DSO control window will therefore only be available if this service is enabled (the default configuration) in the version of the of the Motor Control Firmware running on the platform currently connected to “MCU Motor Studio”.

The DSO control window has two major layouts – configuration and data:

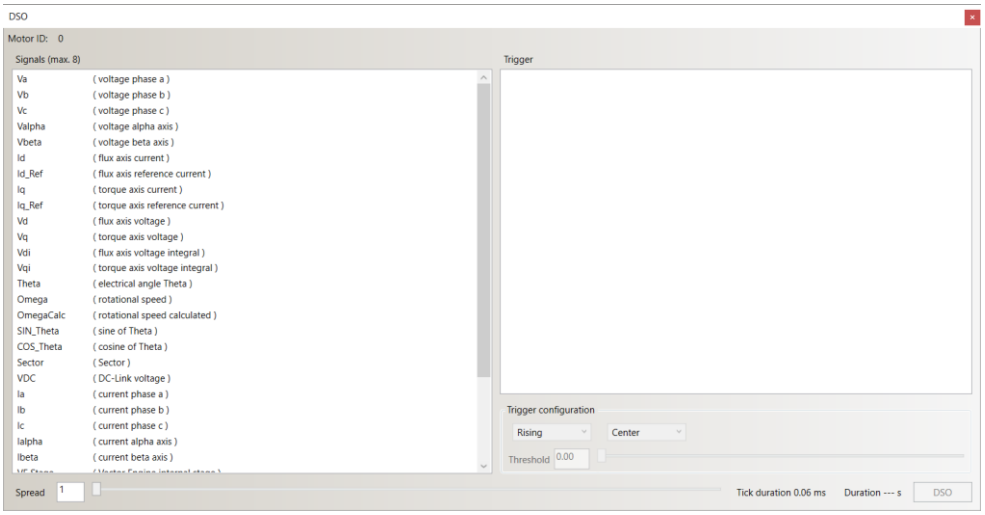


Figure 4-40 DSO Control Window – Initial (Configuration) layout

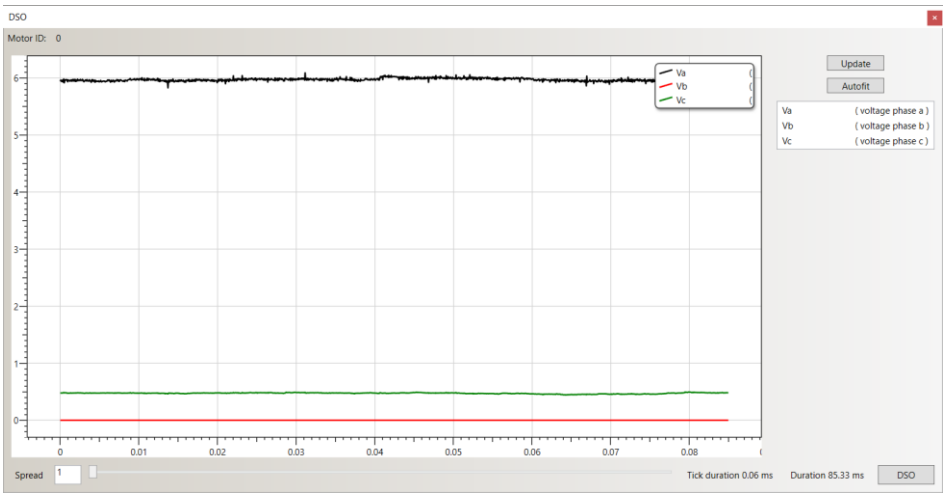


Figure 4-41 DSO Control window - Data layout (example capture with idle motor)

The configuration layout is used to select the parameters that will be collected, the sampling rate at which it will be done and the starting point (configurable trigger or immediately). The data layout is used to draw the graphical representation of the collected results and provide means for detailed inspection of these. The **DSO** button is used to switch between these two. At least one parameter shall be selected before the button becomes active. When pressed while in Control layout it will not only result a layout change, but will initiate a data collection (the first sample taken at matching trigger condition or immediately), whose progress will be indicated by a green progress bar in the middle of the data chart:

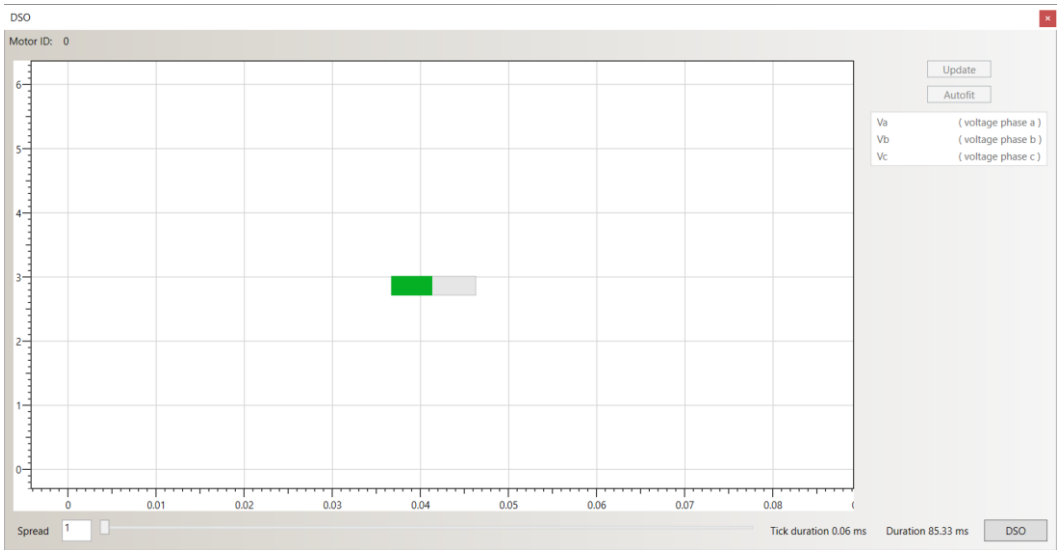


Figure 4-42 DSO Data Collection

4.6.1. DSO Configuration Layout

All signals that may be collected are listed on the lefts side of the configuration layout under **Signals**. The user may select up to **8 parameters** by simply clicking on every desired one. All selected signals will be listed on the right side in the **Trigger** section, following the order these were selected and will be collected/drawn:

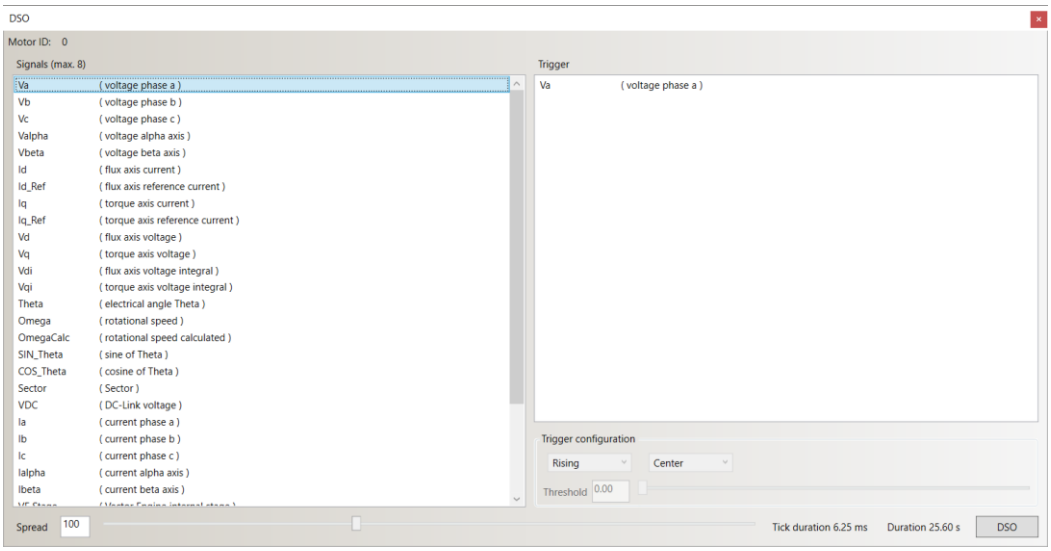


Figure 4-43 DSO Parameter selection

A signal can be deselected by clicking it once again.
The Spread slide bar is used to adjust the sampling rate in number of ticks. Where a tick is the time representation of the used PWM rate. Valid values are 1 to 256. Spread factor of 100 would mean that the parameters are sampled once every one hundred ticks. The single tick duration and total time needed for the capture are indicated by **Tick duration** and **Duration** respectively. The example on figure 4-42 is done with the typically PWM frequency of 16kHz, resulting sampling rate of 62.5µs and a total duration of 25,60s as a total of 4096 samples need to be taken, one every 6.25ms (100 ticks).

The default **Trigger condition** is none, thus data collection will start immediately after pressing the DSO button. A trigger is always related to a particular signal and is set for the current data collection round. It will be reverted back to none at every switch from data to control layout or the signal is deselected/another one is selected. There might be only one valid trigger at a time. Only edge changes are currently supported. The **Threshold** defines the level at which the set condition is considered as matching.

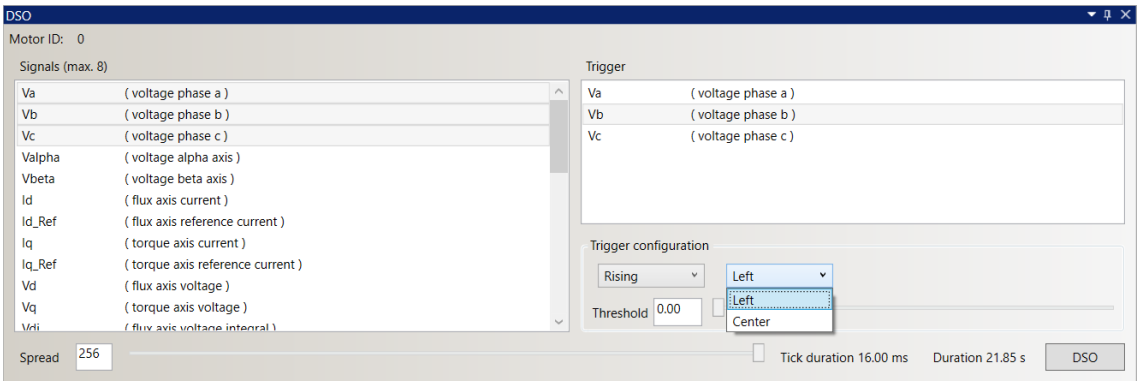


Figure 4-44 DSO trigger configuration

The third option defines the display alignment, namely whether to display the data set collected at the trigger condition data in the middle or on the left side of the data chart. A vertical dash line will indicate the starting point as illustrated on Figure 4-45:

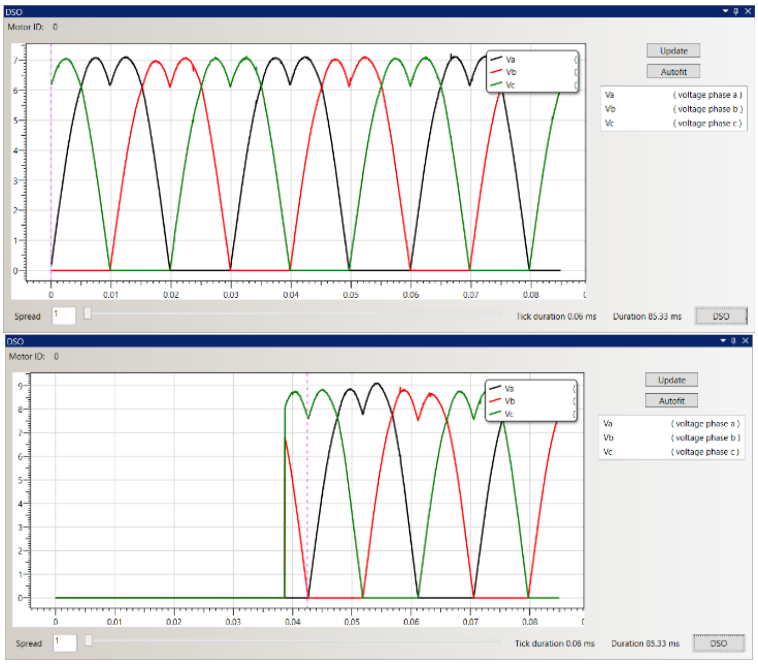


Figure 4-45 DSO trigger left & center capture alignments

To set/change new trigger, a signal in the Trigger list has to be selected. The trigger will only become effective if the DSO button is pressed while the trigger signal is still selected/on focus.
A trigger can be reset/deleted by clicking the signal once again or selecting another signal.

4.6.2. DSO Data Layout

The layout consists largely of a scalable (automatic or manually) two-dimensional timing Data Chart. It visualizes the run-time change of the selected parameters. Data logging is not supported. For all cases it is needed, the HS-DSO feature shall be used instead. Please refer to the next sub-chapter for details.

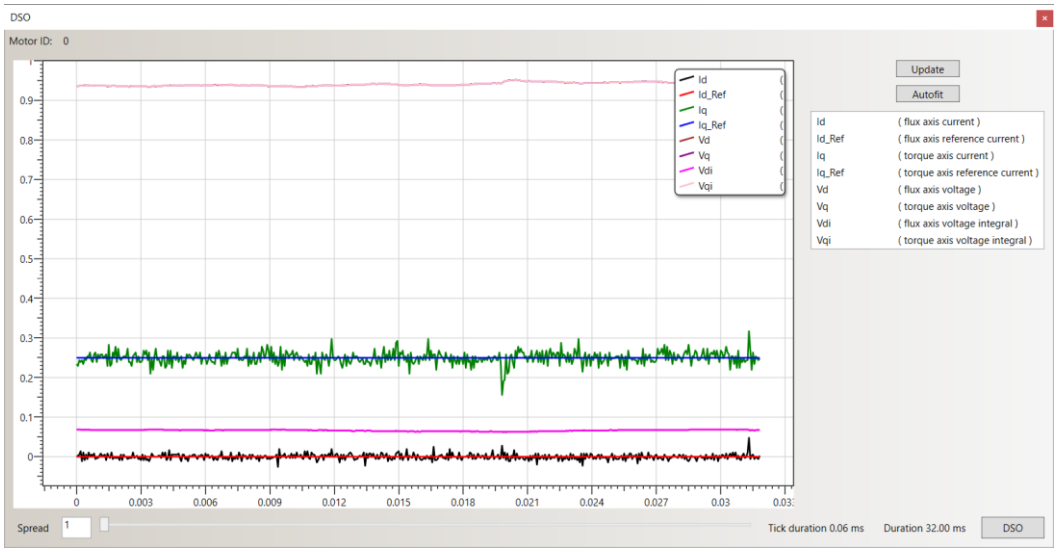


Figure 4-46 DSO Maximal number of displayed parameters (idle motor)

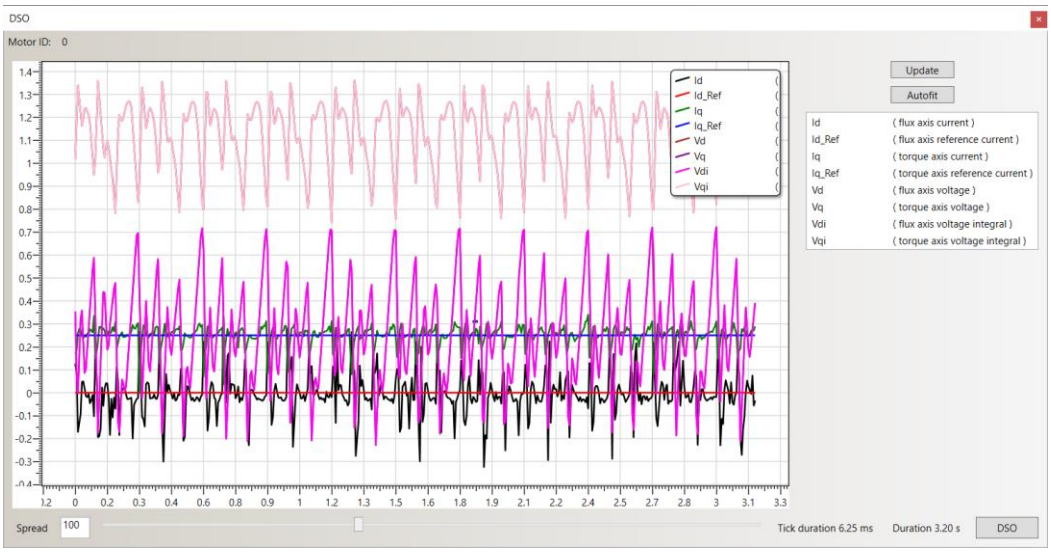


Figure 4-47 DSO Maximal number of displayed parameters (BLDC Motor @ 100RPM, single shunt)

The navigate the chart it has to be on focus. The following operations are supported:

- Zoom in/out using the scroll wheel of the mouse.
- Move left/right/up/down by pressing and holding the left mouse button (right for a left-handed configuration) and moving the mouse in the desired direction.
- Restore the default zoom pressing the Autofit button

Another capture with the same trigger conditions can be initiated by pressing the Update button. Any changes in the sampling rate, done via the **Spread** slide bar, will be applied.

Caution shall be taken when collecting various parameters as due to the different levels/ranges the scaling may be non-proportional and result precision lost for all the signals with much lower levels:

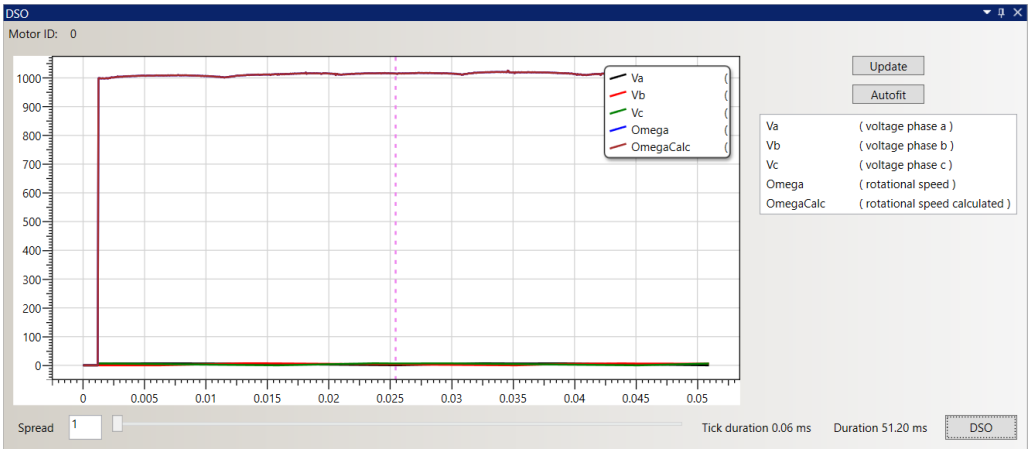


Figure 4-48 DSO Non-proportional scaling & precision loss

4.7. HS-DSO Window (in future releases)

The High-Speed Digital Storage Oscilloscope (HS-DSO) is a variant of the DSO, where the data is continuously collected at “real-time” using separate UART channel and much higher baud rates. It requires special high-speed capable or cable adapter like the FTDI C232HD-EDHSP-0 and properly installed host (Windows®) system driver for the flawless operation. The feature is disabled by default.
The HS-DSO control window will only be available if all of the below conditions are fulfilled:

- The HS-DSO service is enabled (not a default configuration) in the version of the of the Motor Control Firmware running on the platform currently connected to “MCU Motor Studio”.
- A high-speed FTDI adapter was found.

Even if available, it will only be usable if the high-speed serial communication with the target can be established and maintained. Otherwise all its user interface controls (buttons, signal selection, etc.) will be grayed out and inaccessible:

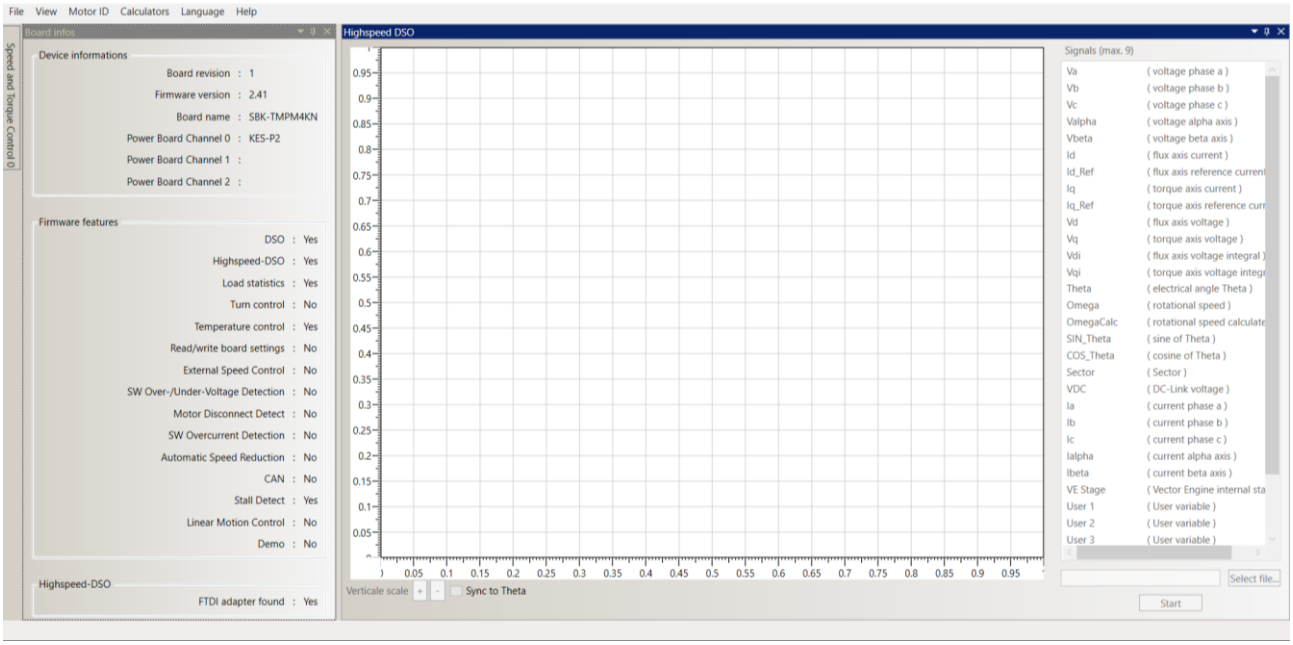
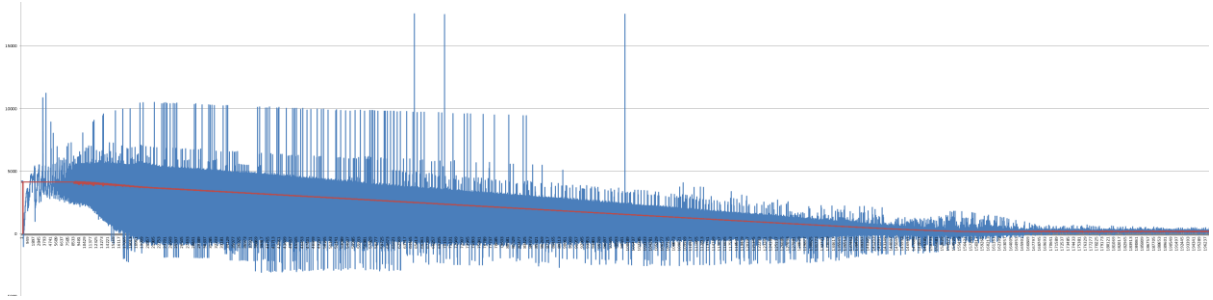


Figure 4-49 HS-DSO Control window

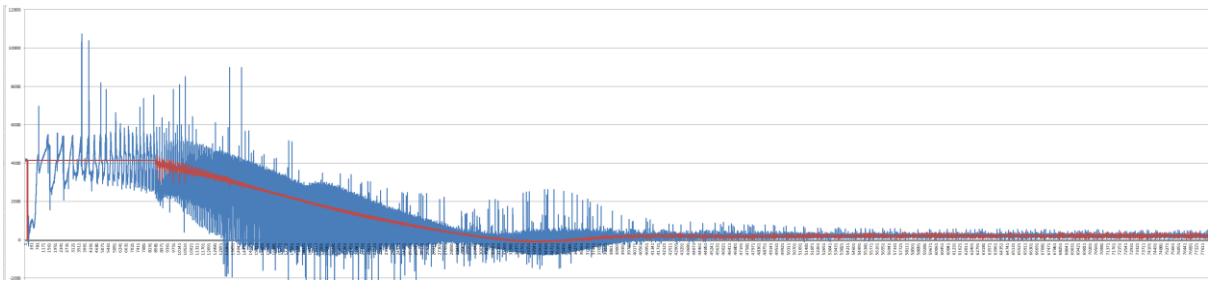
Although looking similar to the DSO, the HS-DSO is in a way superior and offers some few useful extensions:

- Continuous and real-time sampling of the selected parameters (DSO is limited to 4096 samples per run). Start/Stop button is used to control the data recording
- Logging to CSV file
- Synchronization with Theta to avoid signal jitter and improve the precision
- Up to 9 parameters can be sampled simultaneously (DSO is limited to 8)

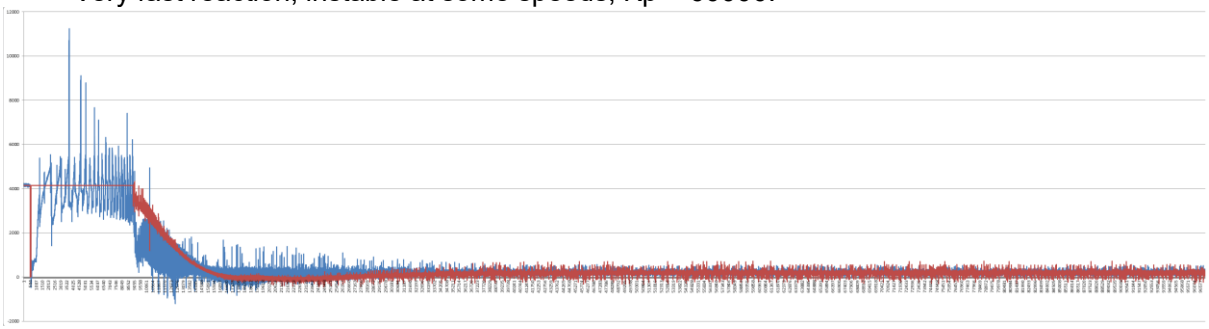
The only disadvantage is that unlike the DSO, a configurable trigger is not supported. There is no difference in all other aspect of the operation and the usage of the data chart.
The figure below contains series of captures taken while tuning the position Kp parameter of PI regulator with a Nanotec DB42S03 motor. The blue line represents Iq, the red the reference Iqref:
Slow regulation and reaction, Kp = 4000:



Stable operation and fast reaction at all speeds, Kp = 16000:



Very fast reaction, instable at some speeds, Kp = 60000:



Very fast reaction, overregulation resulting filed stalls and instable operation, Kp = 90000:

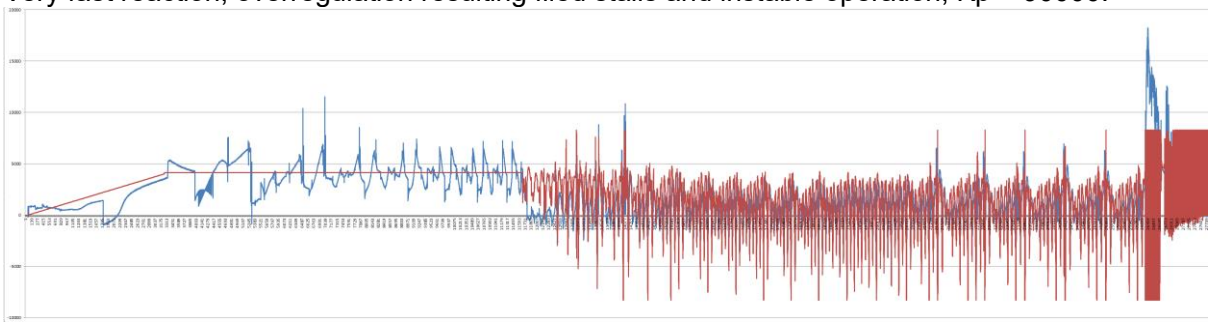


Figure 4-50 HS-DSO Sample captures

4.8. System Load Window

The Motor Control Firmware is capable of doing a dynamic self-assessment of its performance, given the Load Statistics feature is enabled (default configuration).

Figure 4-51 System load window

The reported and drawn values are pure mathematical correlation of the idle and current loads represented in percentage. “MCU Motor Studio” is retrieving these on regular basis and generates

dynamic gauge indication on its System load control window as illustrated on Figure 4-48.

Due to the low priority on the related task in the Motor Control Firmware, the load presentation may be delayed in case of heavy load, even with some rapid changes to follow the actual value. It shall be used with caution, but still gives good indication of the resources usage and available free CPU load.

The position of the control window is illustrative. By default, it will be docked to the left side of the main window. The feature is not available in demo mode.

4.9. Demo Control Window

The Demo Control window provides a basic control for the demo application, offering several buttons for state change and a simple state/transition status indication. However, the set speed is still controlled via the **Slider and the direction via button B5**.
Pressing the Run button will start the motor and the active state will become “Running …”.

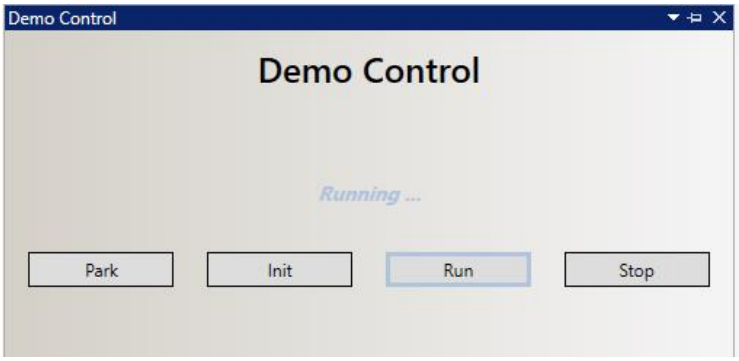


Figure 4-52 Demo control window during ”Run”

Pressing the Stop button will stop the motor and the active state will become “*Stopping …*”. As soon as the motor stands still, it will be changed to “*Idle*”:

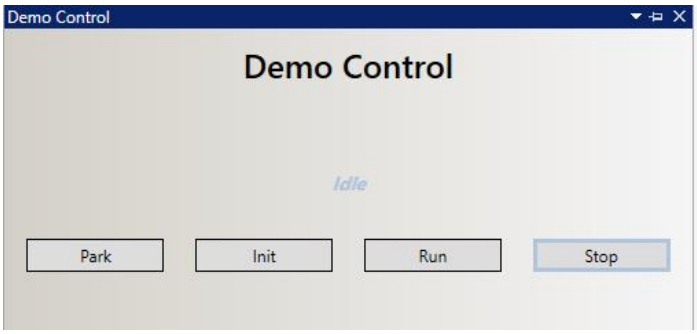


Figure 4-53 Demo control window during ”idle”

As both **Park** and **Init** states are not supported, pressing the corresponding buttons will result no action.
LEDs L2, L3, L5 and L6 will be updated accordingly. Mixed usage of Run/Stop and B6 buttons is supported.

4.10. Turn Control Window

The (Advanced) Turn Control is an open-loop sensor-less core feature of the firmware allowing precise rotor positioning by simply utilizing the computational power of the CPU and the dedicated hardware accelerators.
The Turn Control window will only be visible and accessible if the feature is enabled in the version of the Motor Control Firmware on the platform currently connected to “MCU Motor Studio”.
The user can define number of full mechanical turns, an additional (optional and only available in case of Advanced Turn Control) angle in tenth of degrees and a desired speed. The motor will be rotated the requested number of turns and finally stopped quite precisely at the desired additional angle if such was specified. The drive will be performed at the desired speed if possible. The acceleration, de-acceleration and the maximal possible drive speed will be automatically calculated to match the request as close as

possible:

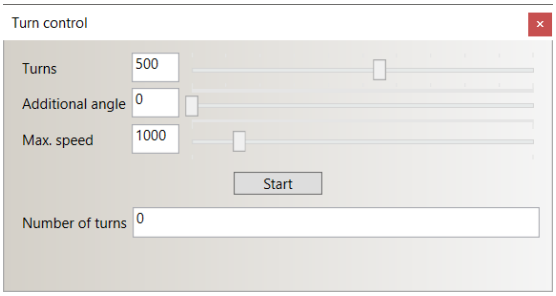


Figure 4-54 Turn Control window

Turns – specify the number of mechanical turns the motor shall be rotated by. Negative numbers have the meaning of turns in counter-clockwise direction (CCW). The number of electrical turns will be handled internally - calculated based on the pole-pairs configuration parameter. The value can be changed with either the slide bar or the input field, resulting automatic update of the other one.

Additional angle – only available for Advanced Turn Control, introduced with the M4KN (TXZ/TXZ+ series). The parameter specifies an additional fraction of a mechanical turn to be done in units of tenth of degrees. Supported values are therefore only 0 to 3600. Changes done with the slidebar will be automatically reflected in the input field and vice versa.

Max speed – the maximal rotational speed the motor may revolve specified in RPM. The value is motor dependent and may not exceed its rated speed. The speed may not be achieved if the requested number of turns and the acceleration/de-acceleration rate are not sufficient to do so.

Start/Stop – Start/Abort the positioning operation. Once Stop is pressed the request will be reset, thus pressing Start again will result a new request with the parameters specified by the sidebars. Resume is not supported.

Number of turns – accumulative counter for the number of full mechanical turns done since the power up of the platform. The value is read-only and can't be reset. As the motor can be driven in both CW and CCW direction the value may not always be the absolute number of turns done, e.g. turns done in CCW direction will result counter decrement, whereas all turns in CW direction will result increment.
An example request for 100 turns and final angle of 30 with 1000 RPM that is ongoing is illustrated below:

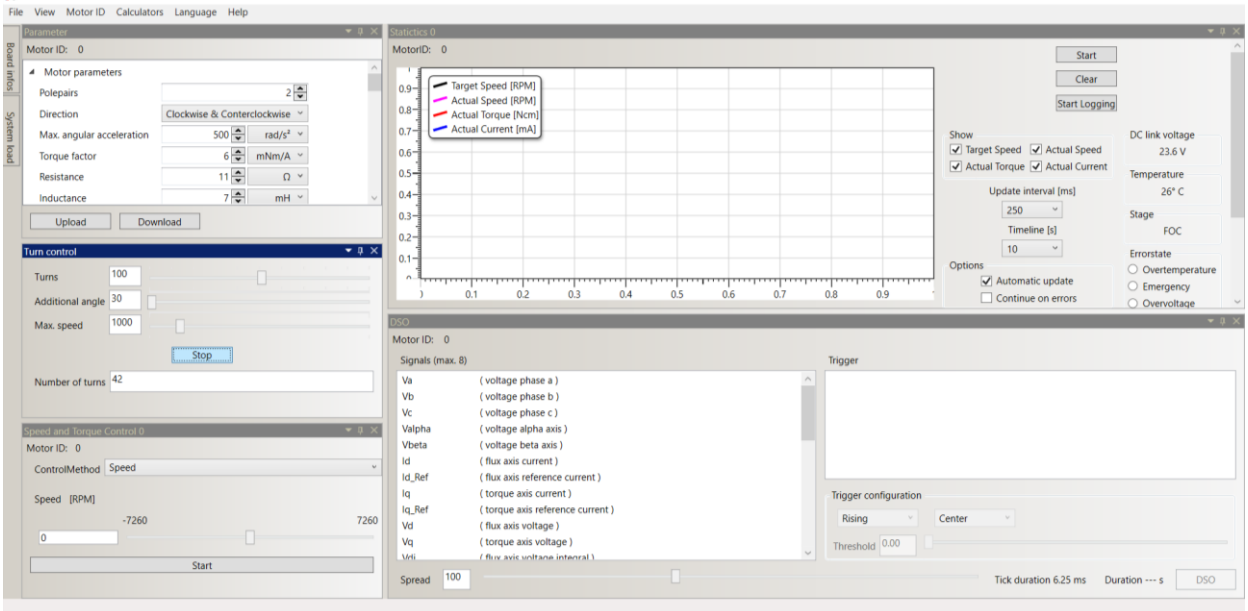


Figure 4-55 Turn Control operation

In some versions of the firmware the motor will not be stopped automatically, thus when the rotor becomes idle the user shall stop it manually via the button. In that case the Start/Stop label on the button may not be updated accurately.

The position of the control window is illustrative. By default, it will be docked to the left side of the main window. The feature is not available in demo mode.

4.11. Linear Motion Window

The Linear Motion Control is an optional precise positioning service, which unlike the Advanced Turn Control, requires external sensor for its operation. This is a closed-loop positioning solution that ensures fixed precision at higher system cost.

The feature is disable by default. The Linear Motion Control window will only be visible and accessible if it is enabled in the particular version of the Motor Control Firmware running on the platform currently connected to “MCU Motor Studio”.

The positioning relies on relative values. The calculation is based on a start pulse count and minimal/maximal allowed values as defined by the encoder configuration of the particular motor. While the rotor revolves the sensor generates pulses, which are counted up or down for clockwise (CW) and counter- clockwise (CCW) direction respectively. The start pulse count is always mapped to the zero counter of the advanced encoder circuit which refers to the initial value of the external sensor. The precision is dependent on the number of pulses generated per electrical turn and therefore varies from system to system.

The performance and precision of the feature is dependent on the used motor, its configuration and the service configuration in the particular firmware build.

Pulses – specify the new pulse count to be reached before the rotor shall be stopped. Negative numbers are not supported. The value can be changed with either the slidebar or the input field, resulting automatic update of the other one.

Max linear speed – the maximal rotational speed the motor may revolve specified in RPM. The value is motor dependent and may not exceed its rated speed. The speed may not be achieved if the requested number of pulses and the acceleration/de-acceleration rate are not sufficient to do so.

Start/Stop – Start/Stop/Resume the positioning operation. When Stop is pressed the request will be paused if the desired pulse count is not yet reached, otherwise it will be stopped. Pressing Start again will resume the operation.

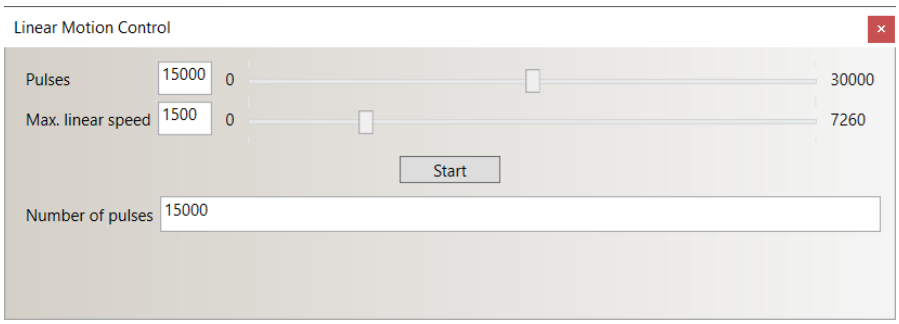


Figure 4-56 Linear Motion window

Number of pulses – current pulse count, starting value is the one specified in the encoder settings. The value is read-only and can’t be reset. The value may be autonomously changing when using the demo applications, as most of these rely on the linear motion for their operation.

A typical request for 120 pulses (20 full turns given a Hall sensor is used) at 1500 RPM which is not yet started is illustrated below:

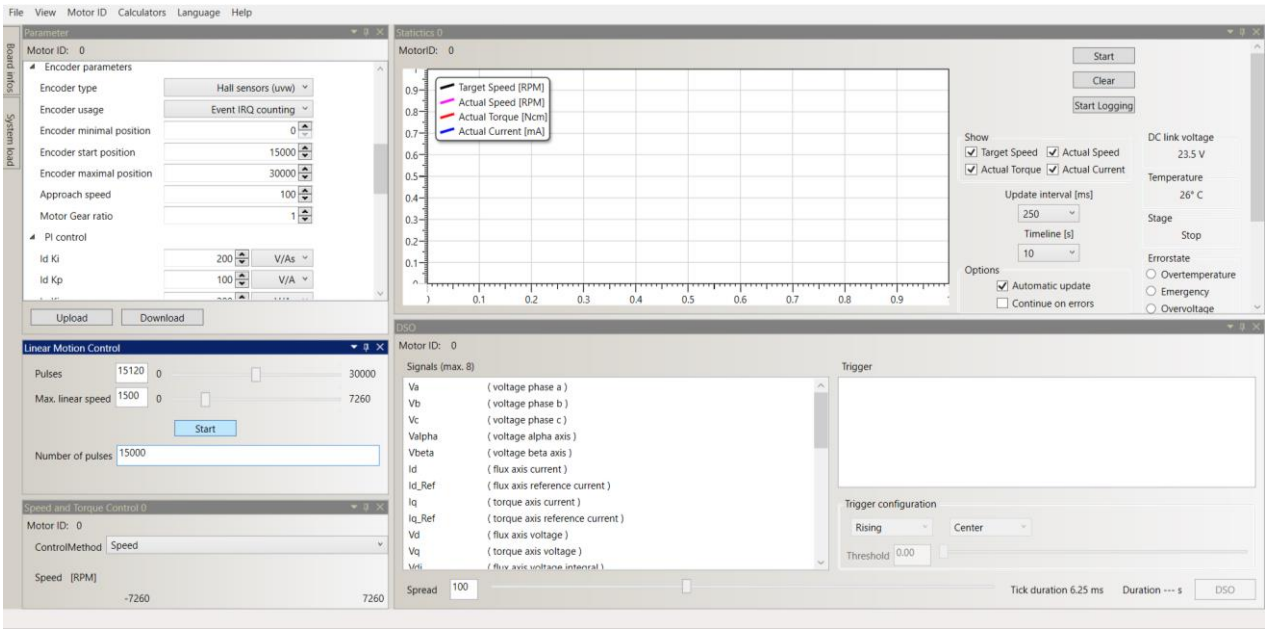


Figure 4-57 Linear Motion operation

The position of the control window is illustrative. By default, it will be docked to the left side of the main window. The feature is not available in demo mode.

4.12. Command Sequencer Window

“MCU Motor Studio” features a simple command sequencer that supports wait stages and the most used core control commands as defined in the Communication Protocol Specification. Minor differences among the firmware & “MCU Motor Studio” versions are likely, as the protocol is updated with new features and/or the list of supported parameters is extended. Best effort is done to constantly increase the number of supported commands, aiming full protocol coverage. As of now the following groups of commands are implemented:

- Execution Control – these are script execution control commands allowing sequence termination, restart and delays between the individual command (wait stages).
- Motor Control – commands to drive a BLDC motor, select the used control method, the active motor, start and stop the motor.
- Parameter Control – set desired speed or torque.
- Statistics Control – basic control of the statistics window, limited to start, stop and clear operation.
- DSO Control – initiate a single DSO capture.

The Control window is illustrated on the figure below. It provides information about the currently selected active motor via the MotorID field, start button to initiate the sequence execution and a sequence grid. Clicking on a row in the grid allows to edit or create a new command. The drop-down style Function filed contains list of supported ones to select from. Value 0 and Value 1 allow entering of parameters if such are to be specified.

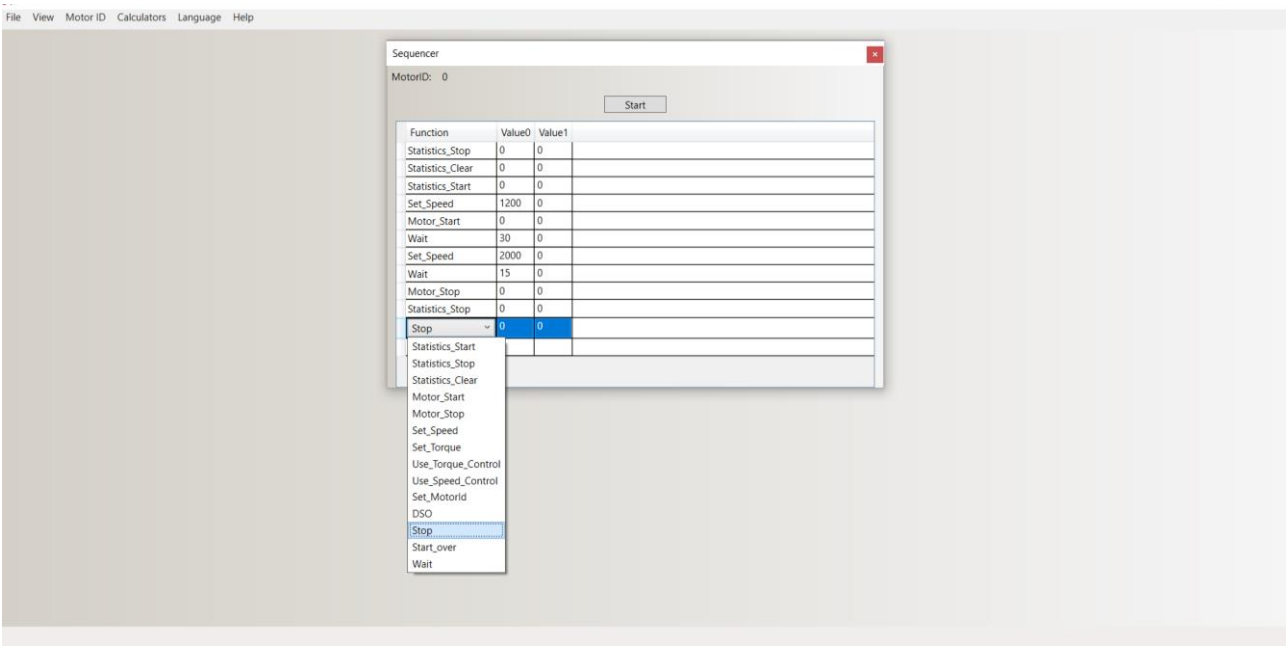


Figure 4-58 Command Sequencer window

A command can be removed by selecting the particular grid row and pressing the Delete key. Multiple sequential commands can be deleted at once. The selection can be done in two ways:

- Shift key is pressed and hold while the commands are selected with the arrows.
- The left mouse button (right for left-handed configuration) is pressed and hold, while the mouse is dragged to mark the desired block of commands.

The selected grid rows will be marked with a blue background and can be removed from the sequence by pressing the Delete key.

Deletion snot recoverable.
An example sequence is presented below. The sequence:

- Clears the statistics window and starts it.
- 1200 RPM will be set as desired speed for the currently selected active motor.
- The motor will be started and will rotate for 30 seconds.
- The speed will be increased to 2000 RPM.
- After another 15 seconds the motor will be stopped, so the statistics
- Finally, the script execution will be terminated

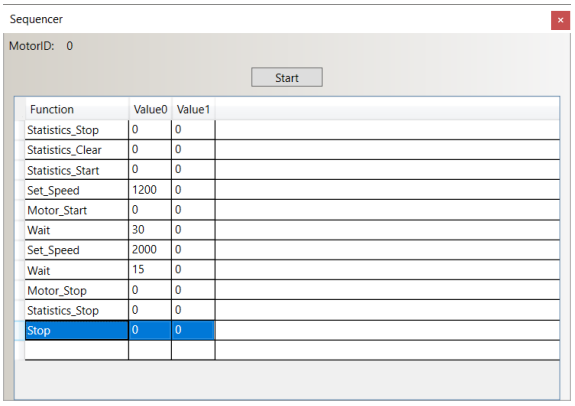


Figure 4-59 Example command sequence

The sequence save, load and clear operations are accessible via the File menu of the menu bar and were already described in a previous chapter.

5. Using “MCU Motor Studio”

There are different approaches for the initial system set-up. “MCU Motor Studio” eases the run-time configuration without the need to rebuilt the firmware. It also gives the possibility to fine tune the system and calculate various parameters.

Regardless of the usage, an important rule is to be followed: **External power supplies with current and voltage limiters shall be used during the initial set-up and configuration, in order to avoid short currents, hardware damages or person injuries!**

5.1. Configuration

Most of the configuration steps during the initial set-up are already described in the Motor Control Firmware User’s Manual. Please refer to the document and related chapter for further details.

The subsequent chapters will detail functionality that is rather “MCU Motor Studio” specific and complements the Firmware manual.

5.1.1. Pole-pair calculation

In some rare cases the information about the number of poles is not available, either due to missing or incomplete specification or due to the fact that the exact motor type is not known. Even a visual inspection may not be sufficient as two or more coils might be connected together. “MCU Motor Studio” features a tool and simple procedure to properly calculate this important parameter.

In the description below the term “pole-pairs” refer to the exact number of electrical poles Xpole, which is needed for proper angular speed control. The mathematical formula to calculate the electrical frequency, given the speed in Hz is as follows:

$$Felmax\ (Hz) = \frac{RPMmax}{60} \times Xpole$$

Equation 5.1: Electrical speed of a motor

It has to be noted that only the electrical speed, given in Hz is dependent on the number of electrical poles. The mechanical speed given in RPM represents the true physical turns and is not affected by the number of poles.

A brief description of the calculation procedure was given in the Pole pairs subchapter. The information given here is to further explain/detail it.

Step 1 – The estimated number of pole-pairs shall be set in the Motor parameters, Polepairs field. If no estimation can be made, 1 shall be kept.

Step 2 – The motor shall be started with decent speed, one in the middle of the supported range for example.

Step 3 – The actual speed is measured using a dedicated equipment/device, like tachometer.

Step 4 – The set speed, the used polepair parameter and the measurement result are input to the calculator and Calculate button is hit. The proper value is given under Calculated pole pairs.

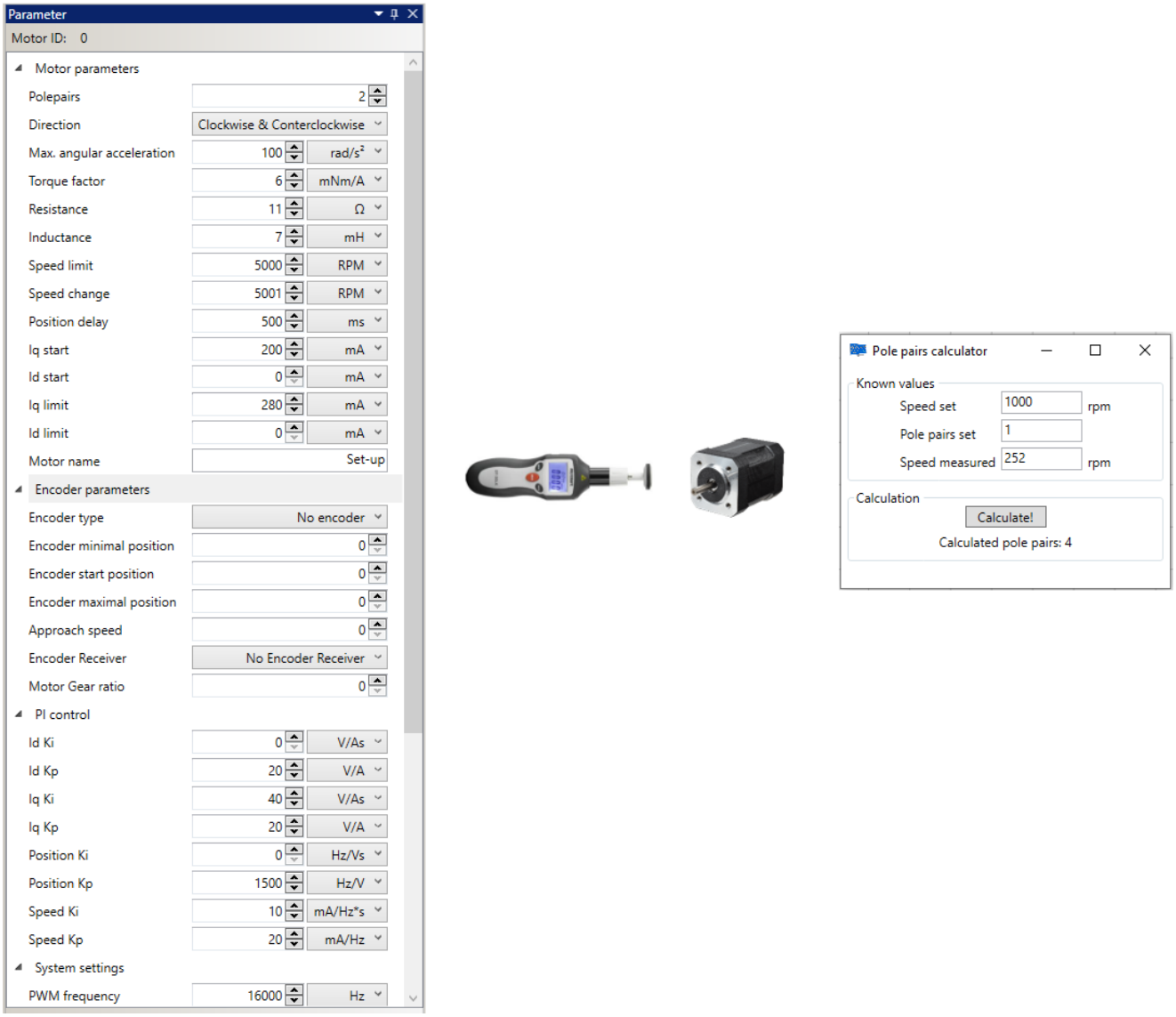


Figure 5-1 Polepair calculator usage

Caution shall be taken selecting the test speed. The general rule is “the higher the test speed, the better the precision of the calculation”, still considering the capabilities of the measurement equipment.

5.1.2. Calculation of Phase Resistance / Inductance

The motor coil’s resistance and inductance are very important for the proper computation of the phase currents, the transformations and the PWM control signals. For the cases these are not available, a measurement has to be done.

The line to line resistance and inductance between every two phases shall be measured with a multimeter on a fully disconnected motor. The procedure cannot be applied to motors with asymmetric phases, as the measurement results will not be sufficient for a proper calculation.

The average of the three line to line measurement result shall be used to produce a mean value, following the equation below:

$$R = (Ruv + Rvw + Rwu)/3$$
$$L = (Luv + Lvw + Lwu)/3$$

Equation 5.2 Average line to line resistance and inductance

The next step is to determine the coli configuration – delta or star. Without this information the calculator can’t be utilized or the results might be improper.

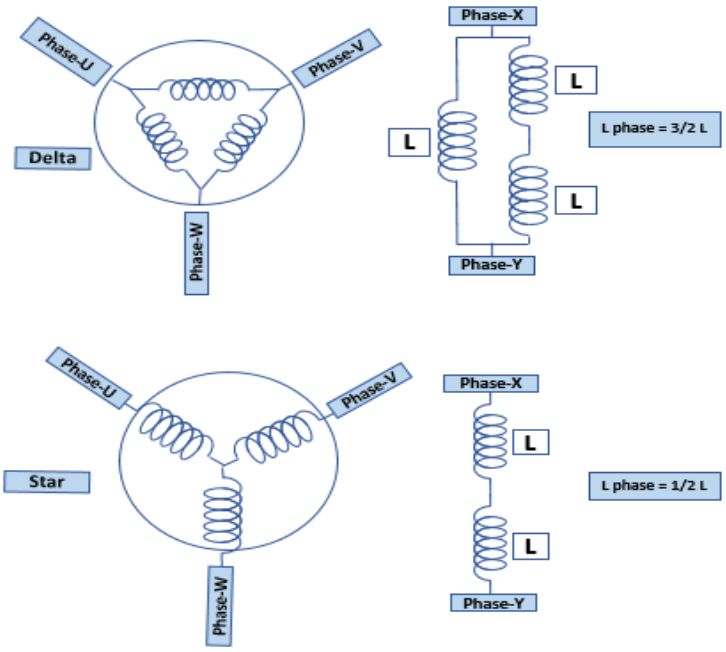


Figure 5-2 Motor coils induction relation & calculation

Once the mean line to line resistance, mean line to line inductance and the coil configuration are input, the phase counterparts will be calculated:

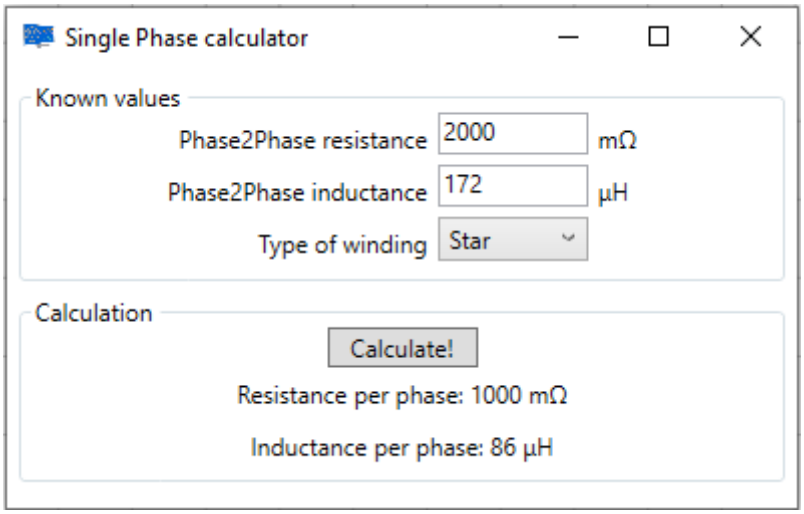


Figure 5-3 Phase resistance and inductance calculation

In some cases, this procedure does make sense even if the parameters are specified.

6. External Software components

Sl.No	Software component	Remarks
1	.NET Framework 3.5	-
2	.NET Framework 3.5	-
3	.NET Framework 4.0	-
4	Windows Presentation Foundation (WPF)	.NET framework based graphical subsystem
5	AvalonDock	WPF control for adding a docking layout system to your application.
6	WPF Gauge	WPF control for adding gauges. Copyright (c) 2011 by Terry Phillips under BSD license
7	WPF Dynamic Data Display	For adding interactive visualization of dynamic data
8	Extend WPF Toolkit	Extended.Wpf.Toolkit 4.2.0
9	XAML Markup Extension	For obtaining a value of specific XAML type.
10	WPF Localize Extension	To localize any type of DependencyProperties or native Properties on DependencyObjects
11	Microsoft Visual Studio Installer Projects 1.0.0	For Preparing installer package.

Table 6.1 External Software components

7. References

[1] TMPM4K Group(2) Datasheet, Revision 1.0, October 2018, Toshiba Electronic Devices & Storage Corporation
[2] Toshiba Motor Studio Firmware User's Manual, Revision 1.0.0, April 2022, Toshiba Electronic Devices & Storage Corporation.
[3] Reference Manual Advanced Vector Engine Plus (A-VE+-B), Revision 3.0, May 2018, Toshiba Electronic Devices & Storage Corporation
[4] Reference Manual Advanced Programmable Motor Control Circuit (A-PMD-A), Revision 2.1, July 2020, Toshiba Electronic Devices & Storage Corporation
[5] Reference Manual Advanced Encoder Input Circuit(32-bit) (A-ENC32-A), Revision 1.1, October 2018, Toshiba Electronic Devices & Storage Corporation
[6] Reference Manual 12-bit Analog to Digital Converter (ADC-I), Revision 0.1, July 2020, Toshiba Electronic Devices & Storage Corporation

8. Revision history

Revision	Date	Changes
1.0.0	2022/05/05	Baselined Version
1.1.0	2022/12/01	Internal review comments are incorporated.

Table 8.1 Revision history

Trademarks

- FreeRTOS™ is a trademark of Amazon Web Services, inc in the US and/or elsewhere. All rights reserved.
- Microsoft® and Windows® are either registered trademarks Microsoft Corporation in the United States and/or elsewhere. All rights reserved.
- Arm® , Cortex® ,Cortex®-M3, Cortex®-M4,Keil® and µVision® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
- Click boards™ is a trademark of MIKROELEKTRONIKA. All rights reserved.
- FTDI may be registered trademarks of “Future Technology Devices International Limited”. All rights reserved.
- IAR Systems® and IAR Embedded Workbench® are registered trademarks are owned by IAR Systems. All rights reserved
- SEGGER and J-Link are trademarks or registered trademarks of SEGGER Microcontroller GmbH & Co. KG. All rights reserved.

Other Company names, product names and service names mentioned herein may be trademarks of their respective companies.

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

Toshiba Electronic Devices & Storage Corporation

https://toshiba.semicon-storage.com/