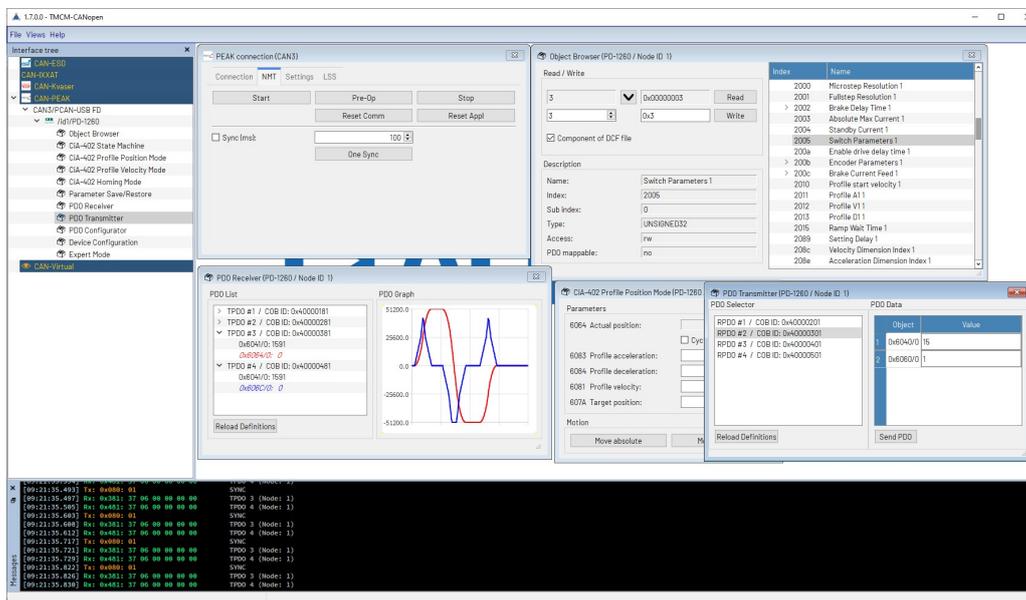


# TMCM-CANopen

Revision V1.7.0.0 | Document Revision V1.05 • 2022-APR-26

The TMCM-CANopen software is a tool for exploring Trinamic modules that use the CANopen® protocol. Offering similar ease of use for CANopen® modules as the TMCL-IDE does for TMCL modules the TMCM-CANopen tool shortens design cycles with all Trinamic CANopen® modules. It provides easy access to all CANopen® objects and simplifies the use of standard CiA-402 modes like profile positioning mode, profile velocity mode, and homing mode. More sophisticated functions like DCF file generation and PDO communication make the setup and usage of Trinamic CANopen® modules easy.

The TMCL-CANopen tool is available free of charge and runs on Windows 7, Windows 8.x, Windows 10 and Windows 11.



# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Getting TMCM-CANopen	4
1.2	Installing TMCM-CANopen	4
1.3	Supported Interfaces	4
1.4	Launching TMCM-CANopen	4
1.5	General conventions	4
1.5.1	Number representation	4
1.5.2	Entering values	4
<b>2</b>	<b>The Main Window</b>	<b>5</b>
2.1	The Menu Bar	5
2.2	Interface and Tool Tree	5
2.3	Log Window	6
<b>3</b>	<b>CAN Bus Connection</b>	<b>7</b>
3.1	Connection	8
3.2	NMT	8
3.3	Settings	8
3.4	LSS	9
3.5	Hints for Peak CAN Interfaces	10
<b>4</b>	<b>EDS Files</b>	<b>11</b>
4.1	About EDS Files	11
4.2	EDS Files for Trinamic Modules	11
4.3	Integrating EDS Files into TMCM-CANopen	11
<b>5</b>	<b>Device specific Tools</b>	<b>13</b>
5.1	The Main Module Tool	13
5.1.1	General	13
5.1.2	File Info	13
5.1.3	EMCY	13
5.2	The Object Browser	14
5.3	CiA-402 State Machine	15
5.4	CiA-402 Profile Position Mode	15
5.5	CiA-402 Profile Velocity Mode	16
5.6	CiA-402 Homing Mode	17
5.7	Parameter Save/Restore	18
5.8	PDO Receiver	18
5.9	PDO Transmitter	20
5.10	PDO Configurator	21
5.11	Device Configuration	23
5.11.1	Generating DCF Files	23
5.11.2	Using DCF Files	23
5.12	Expert Mode	24
<b>6</b>	<b>Figures Index</b>	<b>26</b>
<b>7</b>	<b>Tables Index</b>	<b>27</b>
<b>8</b>	<b>Supplemental Directives</b>	<b>28</b>
8.1	Producer Information	28
8.2	Copyright	28
8.3	Trademark Designations and Symbols	28

8.4	Target User	28
8.5	Disclaimer: Life Support Systems	28
8.6	Disclaimer: Intended Use	28
8.7	Collateral Documents & Tools	29
<b>9</b>	<b>Revision History</b>	<b>30</b>
9.1	Software Revision	30
9.2	Document Revision	30

# 1 Introduction

## 1.1 Getting TMCM-CANopen

The TMCM-CANopen tool can be downloaded free of charge from the software section of the Trinamic website: <https://www.trinamic.com/support/software/#c3224>. The latest version can always be found there. Also older versions can be downloaded from there if needed.

## 1.2 Installing TMCM-CANopen

The TMCM-CANopen tool is provided as a non-install version. This is a ZIP archive that contains all necessary files. After downloading this file, unpack it to one directory on your PC.

## 1.3 Supported Interfaces

For connecting to the CAN bus a CAN interface that is supported by TMCM-CANopen will be needed. Table 1 contains a list of all currently supported CAN interfaces.

## 1.4 Launching TMCM-CANopen

Run TMCM-CANopen simply by double clicking the TMCM-CANopen.exe file from the directory where TMCM-CANopen has been copied to. You can also create a desktop shortcut or attach it to your Start menu.

## 1.5 General conventions

Here are some general conventions that apply throughout the entire TMCM-CANopen tool.

### 1.5.1 Number representation

TMCM-CANopen uses hexadecimal and decimal numbers. Hexadecimal numbers are always shown preceded by "0x", like in C language.

### 1.5.2 Entering values

In many cases, numbers can be entered as decimal numbers or as hexadecimal numbers. Enter hexadecimal numbers preceded by "0x", like in C language. Negative decimal numbers are entered with a preceding "-" sign.

Some input fields will turn yellow when a new value is being entered. This means that the value entered will be written to the corresponding object by pressing the Return key or the Enter key. It is also possible to write out the same value again just by pressing the Return key or the Enter key once again.

## 2 The Main Window

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

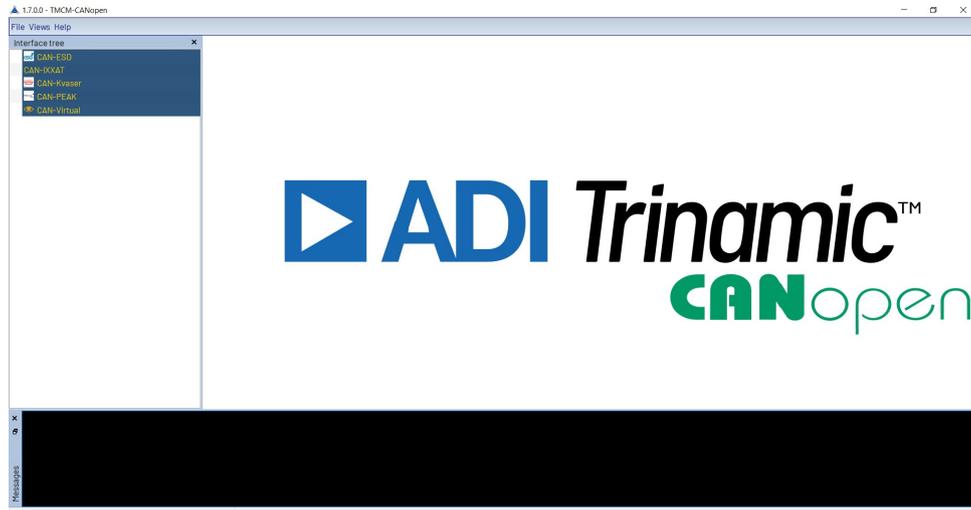


Figure 1: The TMCM-CANopen Main Window

### 2.1 The Menu Bar

The menu bar is placed at the top of the main window. It is not moveable.

Figure 2: The Menu Bar

The menu bar contains three entries:

- File: Contains the Exit entry for terminating the program.
- Views: Contains entries for hiding or showing the tree view and the log window.
- Help: Contains entries that take the user to the TMCM-CANopen website, show the TMCM-CANopen manual and show information about the program version.

### 2.2 Interface and Tool Tree

The tree root entries represent all physically connected CAN interfaces and also non-physical virtual modules. Each root entry contains the connected interfaces and each interface is the parent of one or more connected CANopen devices. Each device is the parent of tools depending of its characteristics.

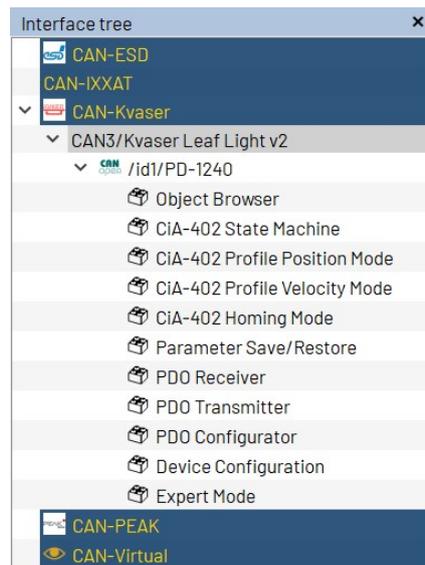


Figure 3: Tree with Connected Devices and their Tools

### 2.3 Log Window

The log window will also be shown when turned on in the View menu (default setting). It shows all CAN bus traffic. CAN messages sent by TMCM-CANopen are shown in orange color. CAN messages received by TMCM-CANopen (sent by devices connected to the CAN bus) are shown in light green color. In front of each CAN message a time stamp is shown in light blue color. The meaning of each CAN message is shown in silver color. Please note that only the meanings of CANopen messages using COB-IDs of the standard CANopen COB-ID set can be decoded by TMCM-CANopen. Other messages from TMCM-CANopen are shown in white color. Using its context menu it is possible to copy or to clear the content of the log window.

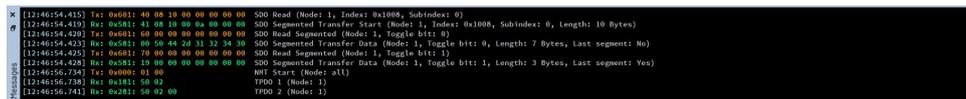


Figure 4: Log Window showing CANopen Messages

The log window and the tree view are freely moveable and can be arranged to an own layout.

### 3 CAN Bus Connection

For connecting to the CAN bus, a CAN interface is needed. As CAN interfaces from different manufacturers also have different programming interfaces, not all of them are supported by TMCM-CANopen. Table 1 contains a list of all CAN interfaces supported by TMCM-CANopen. Only interfaces contained in this (steadily growing) list of CAN interfaces are supported by TMCM-CANopen (these are the same interfaces as supported by the TMCL-IDE). Maybe other CAN interfaces from the same manufacturers will also work with TMCM-CANopen, but have not been tested by Trinamic.

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

List of supported CAN Interfaces	
Manufacturer	Type
ESD	CAN-USB/2 CAN-USB/Micro other CAN-USB interfaces
Kvaser	Leave Light V2
Peak	PCAN-USB (IPEH-002021 or IPEH-002022) PCAN-USB FD (IPEH-004022) PCAN-USB Pro PCAN-USB Pro FD (not tested)
IXXAT	USB-to-CAN compact USB-to-CAN V2 compact

Table 1: List of supported CAN Interfaces

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



Figure 5: The CAN Bus Connection Window

### 3.1 Connection

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

- *Bitrate*: Select the desired CAN bit rate here. The factory default setting on most Trinamic modules is 1000kBit/s. Change this if you have set up your module to use a different bit rate.
- *Search Node IDs from/to*: It is possible to connect more than one module to a CAN bus. For this reason, TMCM-CANopen can search for more than one module on the CAN bus. Enter the node ID of the first module connected to the bus and the node ID of the last module connected to the bus here. If only one module is connected you can normally leave both values at 1, as this is also the factory default setting on Trinamic modules. Or if the module is set to a different ID, set both values to that ID. If you are not sure about the ID setting of a module you can also enter from 1 to 127 so that TMCM-CANopen will automatically scan through all possible CANopen node IDs, but this will take some time.
- *Ch. (Channel)*: Some CAN interfaces are equipped with more than one CAN port. In such a case the *Channel* entry will be shown. The port to be used can be specified there. For CAN interfaces that are equipped with only one port (which is the case for most CAN interfaces) this setting will not be shown.
- *Connect*: Click the *Connect* button to open the connection and to start searching for modules connected to the CAN bus. The search progress will be shown by the progress indicator. All modules that have been found will appear on the tree view on the left hand side of the main window.
- *Disconnect*: Click here to close the connection.

### 3.2 NMT

Use the *NMT* tab to send NMT commands to all devices connected to the CANopen network. The following NMT commands are available:

- *Start*: Issues an NMT Start command for all modules connected to the bus.
- *Pre-Op*: Issues an NMT Pre-Op command for all modules connected to the bus.
- *Stop*: Issues an NMT Stop command for all modules connected to the bus.
- *Reset Comm*: Issues an NMT Reset Communication command for all modules connected to the bus.
- *Reset Appl*: Issues an NMT Reset Application command for all modules connected to the bus.
- *Sync (ms)*: When the *Sync (ms)* checkbox is switched on a SYNC command will be sent periodically. The time between two SYNC commands can also be set here.
- *One Sync*: Sends just one SYNC command on the CAN bus.

### 3.3 Settings

Use the *Settings* tab of the CAN connection window to control some general settings:

- *Log CAN messages*: This checkbox switches on or off CAN message logging for this port. When switched on (default setting), all CAN messages sent and received on this CAN port will be shown in the Log window.
- *Block non-CANopen COB-IDs*: When checked, all CAN messages with a COB-ID that does not belong to the CANopen standard COB-ID set will not be shown in the log window and will also not be further processed by TMCM-CANopen. This way, other non-CANopen traffic on the same bus can be masked out from TMCM-CANopen.

- *Polling enabled*: Some tools in TMC-CANopen use polling of specific objects by SDO on a regular base. Using this checkbox, this kind of polling can generally be switched off or on. By default, this is switched on.
- *Polling interval*: This is the interval for tools that use polling of objects by SDO. The default value is 100ms.
- *SDO timeout*: This is the timeout value for SDO communication. When accessing an object via SDO communication the module has to answer within this time. Otherwise an error message will be shown. Mostly the default value of 500ms is sufficient, but there may be cases where this timeout value has to be set higher.
- *LSS fastscan timeout*: This is the timeout value needed for the LSS fastscan algorithm. The default value is 500ms. When many slave devices are connected or with slave devices that respond slower, a higher value will be needed. When only a few devices are connected that have a short response time also lower values can be used.
- *LSS activate bitrate delay*: Sets the delay which is used with the LSS activate bitrate command (for single devices as well as for the entire network).

### 3.4 LSS

The LSS tab provides support for the CANopen Layer Setting Service (LSS). With slave devices that support LSS the node ID and bit rate can be set using these functions. In the *Network* section, network wide LSS functions are provided.

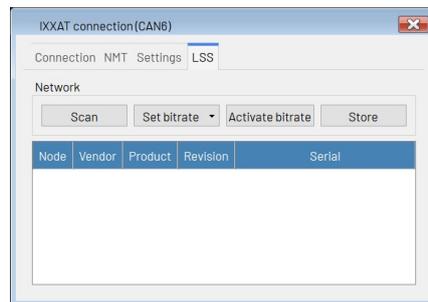


Figure 6: The LSS tab (with empty LSS slave device list)

- *Scan*: Clicking this button erases the list of LSS devices and starts an LSS fastscan. All connected devices that support fastscan and that are in unconfigured (LSS Waiting) state will be found and shown in the list of LSS devices. Depending on the number of unconfigured devices and the LSS fastscan timeout setting the LSS fastscan process can take some time (at least  $33 \cdot LSS_{fastscan\_timeout}$ ). Setting the LSS fastscan timeout to a lower value accelerates this process. But be careful, as with a too low timeout setting some unconfigured LSS devices might not be found.
- *Set bitrate*: This button opens a menu for choosing the CAN bit rate setting of all slave devices. The new setting will not be activated, nor will the bit rate used by the TMC-CANopen software be changed.
- *Activate bitrate*: This will send an LSS activate bitrate command to all slave devices. The delay entered in the LSS activate bitrate delay setting will be used. After the end of this delay time all slave devices will be using the new bitrate setting. Again, this will not change the bitrate used by TMC-CANopen.
- *Store*: All LSS slave devices will store their new node ID and bitrate settings permanently.

Clicking the right mouse button over the list of LSS modules brings up a context menu containing functions for individual devices. It also offers the possibility to add devices to the list that are already configured, so that their node ID and bitrate settings can also be changed.

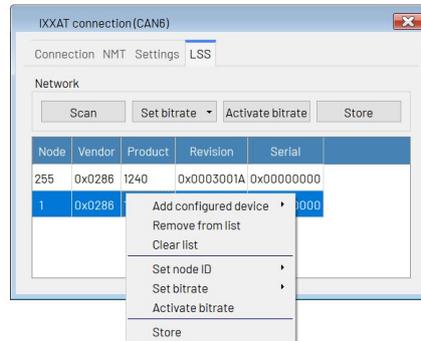


Figure 7: The LSS device list containing two LSS devices and showing its context menu

- **Add configured device:** Use this function to add a configured device to the list of LSS modules. In the submenu, just enter its node ID and press the return key. It will then appear in the list.
- **Remove device:** Removes the currently selected device from the list.
- **Clear list:** Clears the entire LSS device list.
- **Set node ID:** This function changes the node ID of the currently selected device. In the submenu, enter the new node ID and press the return key. The node ID of that device will then be changed (but the setting will not be stored permanently in the device).
- **Set bitrate:** Use this to change the bitrate of the currently selected device. Choose the new bitrate from the submenu. The new bitrate will be set but will not be activated on the device.
- **Activate bitrate:** Activates the new bitrate setting on the currently selected device only. The LSS bitrate activation delay will be used. Use with care as this can lead to the fact that one device uses a different bitrate than the other devices.
- **Store:** Any changes made to the node ID and/or bitrate settings of the selected device will be stored permanently in its non-volatile memory.

### 3.5 Hints for Peak CAN Interfaces

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

- On Windows 32-bit systems:
  - [PCANBasic.dll 32 bit](#) to Windows\System32.
- On Windows 64-bit systems:
  - [PCANBasic.dll 32 bit](#) to Windows\SysWOW64
  - [PCANBasic.dll 64 bit](#) to Windows\System32

## 4 EDS Files

### 4.1 About EDS Files

In order to be able to work properly with a CANopen slave device, TMCM-CANopen needs to have some knowledge of its CANopen object dictionary and its other capabilities. For this purpose, EDS (electronic data sheet) files are used. Every CANopen slave device vendor provides an EDS file for each of his slave devices. Thus, also for each Trinamic CANopen® module an EDS file is provided on Trinamic’s website.

### 4.2 EDS Files for Trinamic Modules

The EDS files for all Trinamic CANopen® modules are already integrated into TMCM-CANopen, so one does not have to care about this. Only if for example a newer firmware version for a specific module is provided by Trinamic it might be necessary to integrate the newer EDS file into TMCM-CANopen. The same will also be necessary when slave devices of other vendors are to be used with TMCM-CANopen.

### 4.3 Integrating EDS Files into TMCM-CANopen

Other EDS files (that can either be newer EDS files from Trinamic or EDS files from other manufacturers) can be integrated into TMCM-CANopen as follows:

1. Open the directory that contains the TMCM-CANopen.exe file
2. Create a directory called “UserEds” there.
3. Copy the new EDS files into the *UserEds* directory.

Name	Date modified	Type	Size
objects	08.12.2020 11:07	file folder	
platforms	11.12.2020 12:04	File folder	
Plugins	08.12.2020 10:59	File folder	
printsupport	11.12.2020 12:04	File folder	
Resources	08.12.2020 11:06	File folder	
styles	11.12.2020 12:04	File folder	
Temp	11.12.2020 12:11	File folder	
UserEds	09.12.2020 14:31	File folder	
IDEComm5.12.9.dll	08.12.2020 10:17	Application extens...	1.030 KB
IDECommon5.12.9.dll	08.12.2020 10:22	Application extens...	295 KB
IDECore5.12.9.dll	08.12.2020 10:15	Application extens...	1.406 KB
IDEDBG5.12.9.dll	08.12.2020 10:20	Application extens...	371 KB
IDEGui5.12.9.dll	08.12.2020 10:20	Application extens...	2.153 KB
IDEPlugin5.12.9.dll	08.12.2020 10:22	Application extens...	1.542 KB
libgcc_s_dw2-1.dll	19.03.2018 14:12	Application extens...	112 KB
libstdc++-6.dll	19.03.2018 14:12	Application extens...	1.507 KB
libwinpthread-1.dll	19.03.2018 14:12	Application extens...	46 KB
Qt5Charts.dll	12.06.2020 06:53	Application extens...	1.593 KB
Qt5Core.dll	26.10.2020 11:07	Application extens...	6.477 KB
Qt5Gui.dll	11.06.2020 14:41	Application extens...	6.785 KB
Qt5Network.dll	11.06.2020 14:42	Application extens...	1.849 KB
Qt5PrintSupport.dll	11.06.2020 14:42	Application extens...	349 KB
Qt5SerialBus.dll	11.06.2020 14:52	Application extens...	278 KB
Qt5SerialPort.dll	11.06.2020 14:48	Application extens...	90 KB
Qt5Svg.dll	11.06.2020 14:49	Application extens...	363 KB
Qt5Widgets.dll	11.06.2020 14:42	Application extens...	6.193 KB
Qt5Xml.dll	11.06.2020 14:42	Application extens...	226 KB
TMCM-CANopen.exe	11.12.2020 12:04	Application	347 KB

Figure 8: The UserEds directory

When an EDS file from the *UserEds* directory contains the same vendor ID and product ID as an EDS file that is already built into TMCM-CANopen then the EDS file from the *UserEds* directory will be used.

If no EDS file can be found for a device then only the [Expert Mode](#) tool can be used. All other tools of TMCM-CANopen will not be usable in such a case.

## 5 Device specific Tools

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

### 5.1 The Main Module Tool

Click on the module name itself in the tree view to open the main module tool. This provides some general information about the connected module as well as the possibility to send NMT commands to this module only.

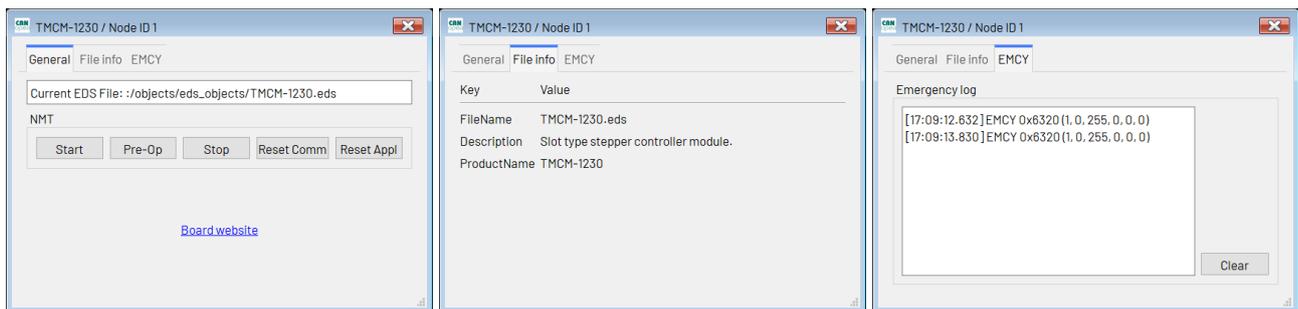


Figure 9: The Main Module Window

#### 5.1.1 General

The *General* tab provides the name of the EDS file and NMT functions for this module:

- *Start*: Issues an NMT Start command for this module.
- *Pre-Op*: Issues an NMT Pre-Op command for this module.
- *Stop*: Issues an NMT Stop command for this module.
- *Reset Comm*: Issues an NMT Reset Communication command for this module.
- *Reset Appl*: Issues an NMT Reset Application command for this module.

The link to the board website provided here can only work correctly when the EDS file also contains the correct board website URL (which often is not the case).

#### 5.1.2 File Info

The *File Info* tab contains some information from the EDS file. These are the EDS file name as contained by the EDS file, the product description and the product name.

#### 5.1.3 EMCY

The *EMCY* tab contains a log of the CANopen emergency messages (EMCYs) produced by this device. For each EMCY, a time stamp, the error code and (in parentheses) the five additional bytes are shown. Please note that the time stamp shown here may differ from the time stamp shown in the [main log window](#) by a few milliseconds. Use the *Clear* button to clear the emergency log. Explanations of the EMCY messages used by the individual device can be found in its CANopen firmware manual.

## 5.2 The Object Browser

The object browser is the main tool for reading and writing CANopen objects via SDO.

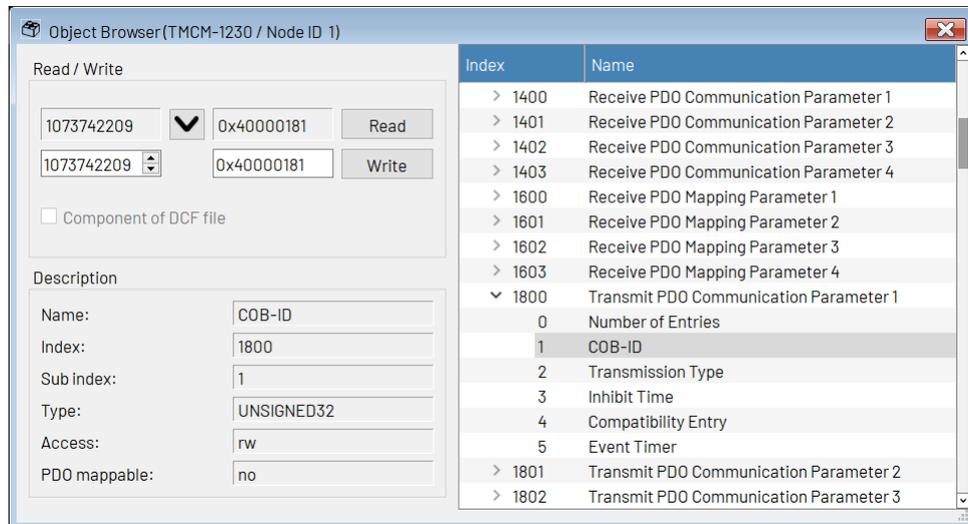


Figure 10: The Object Browser

On the right hand side of the object browser window, all objects contained by the EDS file are shown. Click on an object to select it. If an object contains sub indices, an arrow will be shown beneath the object entry. Click the arrow to open the object sub indices. Click on a sub index to select it.

Whenever an object or a sub index is selected, its content will be read and displayed on the upper left side of the window. Click the *Read* button to read it again. Numerical values will be shown as decimal numbers in the left display field and as hexadecimal values in the right display field. Clicking the  button copies the contents of the display fields into the input fields. The left input field can only be used to enter decimal numbers, whereas the right input field serves for entering hexadecimal numbers (preceded by "0x" like in C language) or decimal numbers. Whenever one of these input fields gets changed, the other one will also be changed to the same value. Clicking the *Write* button writes the contents of the input field to the object. Objects that contain visible strings or octet strings are only shown in the right display field, and a new value can be entered also in the right input field only. Visible strings can be entered as text, and octet strings can be entered as hexadecimal bytes (two digits without preceding "0x") separated by spaces.

Whenever an object has been written, the *Component of DCF File* checkbox will be checked. This means that when generating a DCF file using the [Device Configuration](#) tool the last entered value of this object will also be written to the DCF file. Uncheck this checkbox if the object is not to be written to the DCF file. You do not have to care about this checkbox when not using DCF file generation.

The object browser also shows some information about the selected object:

- *Name*: The name of the object.
- *Index*: The index of the object as a hexadecimal value.
- *Sub index*: The sub index of the object as a decimal value.
- *Type*: The data type of the object.
- *Access*: The access type of the object.
- *PDO mappable*: Shows if the object can be mapped to PDOs.

### 5.3 CiA-402 State Machine

With motion control modules that use the CiA-402 profile, the CiA-402 state machine tool can be used.

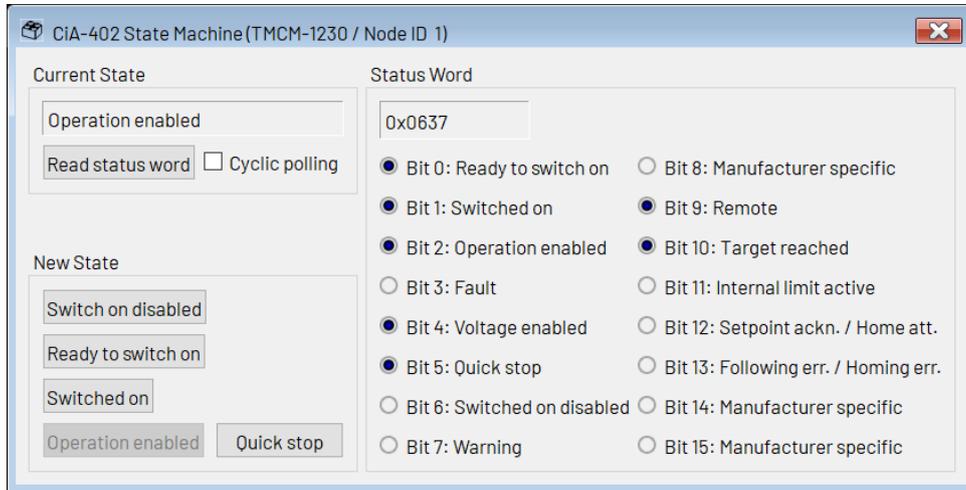


Figure 11: The CiA-402 State Machine Tool

Its window is divided into three sections:

- **Current State:** In this section the current state of the state machine is shown. Click the *Read status word* button to update the current state display. When the *Cyclic polling* checkbox is checked the current state display will be updated on a regular base, by polling the status word via SDO.
- **Status Word:** Here, the status word is shown as a hexadecimal value. Also the single bits of the status word are shown with their meanings.
- **New State:** Use the buttons provided here to switch to a different state. The current state and status word display will be updated automatically afterwards.

When using a multi-axis module, an axis selection menu will be shown at the top of the window so that the tool can be used for each axis of a multi-axis module.

### 5.4 CiA-402 Profile Position Mode

With motion control modules that support the CiA-402 profile position (pp) mode, the CiA-402 profile position mode tool can be used.

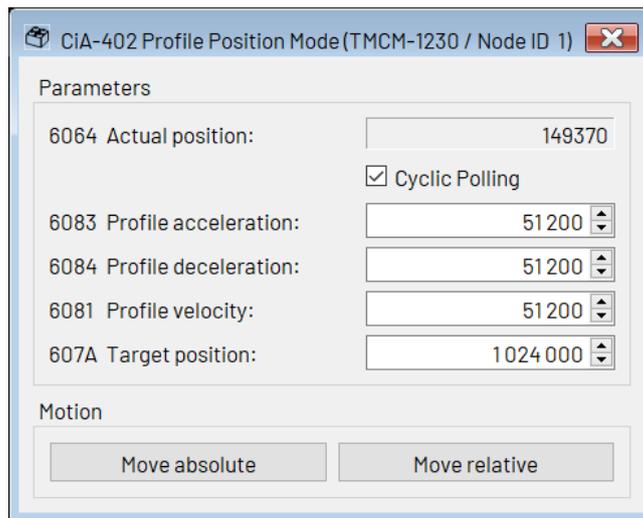


Figure 12: The CiA-402 Profile Position Mode Tool

In the *Parameters* section the acceleration, deceleration, velocity and target position can be set. When the *Cyclic Polling* checkbox is checked the actual position display will be updated on a regular base, by polling the actual position object via SDO.

Click the *Move absolute* or the *Move relative* button to start the motion.

When using a multi-axis module, an axis selection menu will be shown at the top of the window so that the tool can be used for each axis of a multi-axis module.

## 5.5 CiA-402 Profile Velocity Mode

With motion control modules that support the CiA-402 profile velocity (pv) mode, the CiA-402 profile velocity mode tool can be used.

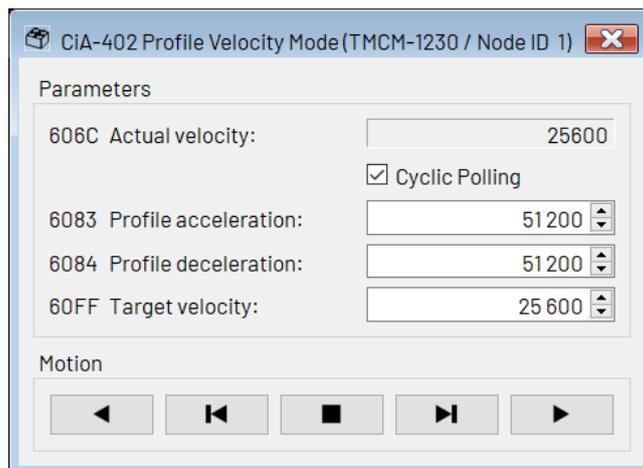


Figure 13: The CiA-402 Profile Velocity Mode Tool

In the *Parameters* section the acceleration, deceleration, and target velocity can be set. When the *Cyclic Polling* checkbox is checked the actual velocity display will be updated on a regular base, by polling the actual velocity object via SDO.

Use the arrow buttons in the *Motion* section to control the motor:

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

When using a multi-axis module, an axis selection menu will be shown at the top of the window so that the tool can be used for each axis of a multi-axis module.

## 5.6 CiA-402 Homing Mode

With motion control modules that support the CiA-402 homing (hm) mode, the CiA-402 homing mode tool can be used.

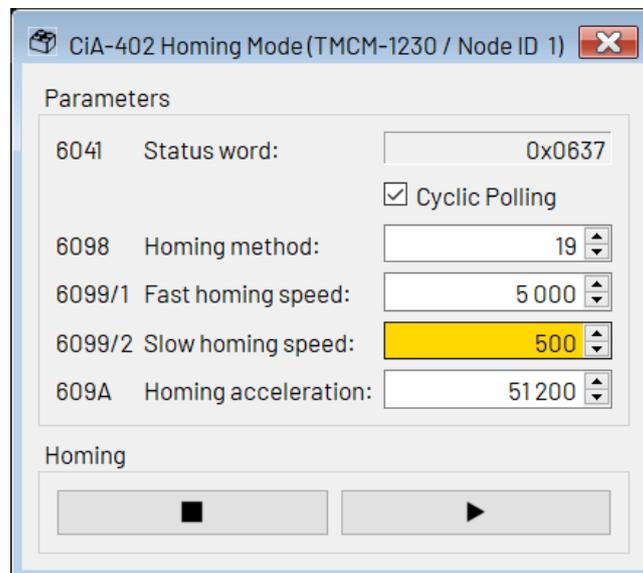


Figure 14: The CiA-402 Homing Mode Tool

In the *Parameters* section the homing method, homing speeds and homing acceleration can be set. When the *Cyclic Polling* checkbox is checked the status word display will be updated on a regular base, by polling the status word object via SDO.

Use the buttons in the *Homing* section to start or to interrupt the homing.

When using a multi-axis module, an axis selection menu will be shown at the top of the window so that the tool can be used for each axis of a multi-axis module.

## 5.7 Parameter Save/Restore

This tool can be used with all devices that support saving parameters to non-volatile memory.

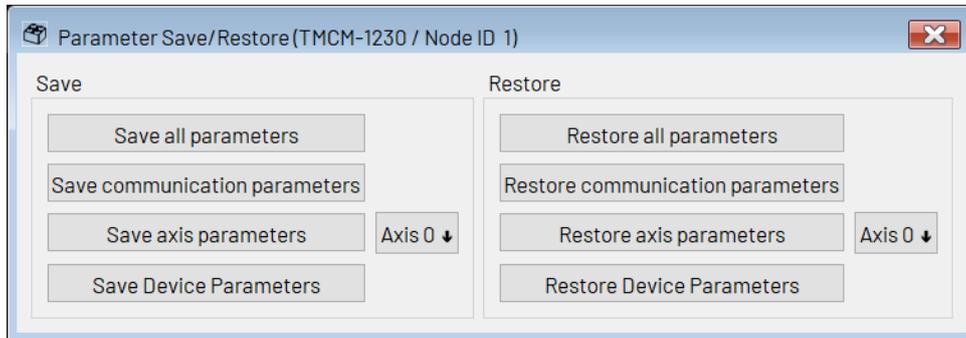


Figure 15: The Parameter Save/Restore Tool

Using this tool, the parameter save/restore functions of a device can be used easily, without having to remember the CANopen save and load signatures. The tool provides the following functions:

- *Save all parameters*: Writes the save signature to object 1010<sub>h</sub> sub-index 1 in order to save all parameters.
- *Save communication parameters*: Writes the save signature to object 1010<sub>h</sub> sub-index 2 in order to save just the communication parameters.
- *Save axis parameters*: Writes the save signature to object 1010<sub>h</sub> sub-index 4 (or on multi-axis modules:  $4 + axis\_number$ ) in order to save just the parameters of the selected axis.
- *Save device parameters*: Writes the save signature to object 1010<sub>h</sub> sub-index 5 (or on multi-axis modules:  $4 + number\_of\_motors$ ) in order to save just the device related parameters (configuration of general purpose I/O ports for example).
- *Restore all parameters*: Writes the load signature to object 1011<sub>h</sub> sub-index 1 in order to restore all parameters to their default settings.
- *Restore communication parameters*: Writes the load signature to object 1011<sub>h</sub> sub-index 2 in order to restore just the communication parameters to their default settings.
- *Restore axis parameters*: Writes the load signature to object 1011<sub>h</sub> sub-index 4 (or on multi-axis modules:  $4 + axis\_number$ ) in order to restore just the parameters of the selected axis to their factory default settings.
- *Restore device parameters*: Writes the load signature to object 1011<sub>h</sub> sub-index 5 (or on multi-axis modules:  $4 + number\_of\_motors$ ) in order to restore just the device related parameters.

## 5.8 PDO Receiver

The PDO Receiver tool displays the contents of PDOs sent out by the device (TPDOs). When opened for the first time, the PDO Receiver tool reads out all TPDO definitions from the device in order to be able to display the PDO contents correctly. Click on the arrow beneath a TPDO entry to view its contents. The display will be updated each time such a PDO has been sent out by the device and reaches the PDO

Receiver tool. Inactive TPDOs are displayed in gray colour. The PDO contents are shown with the object index (hexadecimal), the object sub-index (decimal) and the contents of that object. It is also possible to select PDO entries to be displayed in a graphical way. If this is selected the entry will be displayed in *italic letters*. The colour of each PDO entry can also be individually selected.

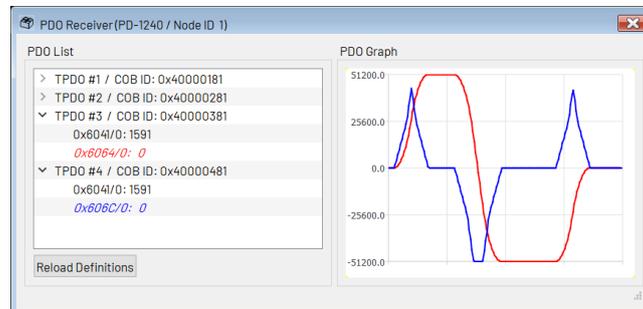


Figure 16: The PDO Receiver Tool

Double click on the displayed contents in order to change between decimal and hexadecimal value representation of the content. Clicking the right mouse button over a PDO entry brings up a context menu. In this menu, hexadecimal or decimal value representation can also be selected and the PDO entry can be added to the graphical view.

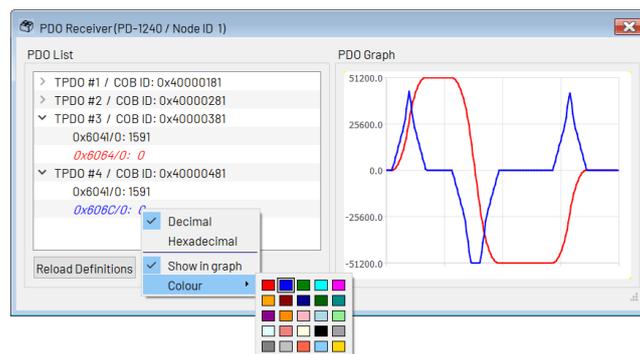


Figure 17: The PDO Entry Context Menu

The PDO entry context menu offers the following functions:

- *Decimal*: Selects decimal value representation (same function as double clicking the value).
- *Hexadecimal*: Selects hexadecimal value representation (same function as double clicking the value).
- *Show in graph*: Selects (or deselects) the PDO entry to be displayed in the graphical view.
- *Colour*: Selects the colour used for the graphical view. The numerical value will also take that colour.

Clicking the right mouse button over the graphical view brings up the PDO graph context menu.

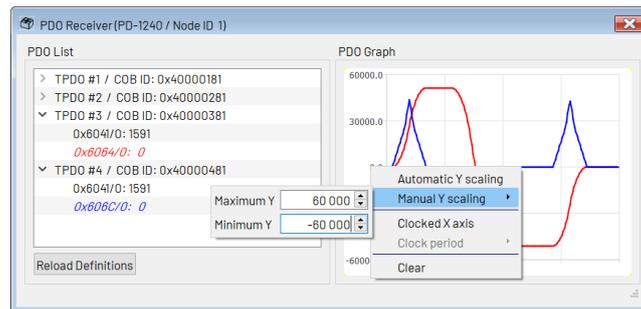


Figure 18: The PDO Graph Context Menu

This context menu offers the following functions:

- **Automatic Y scaling:** Selects automatic Y scaling (default setting). The range of the Y scale is automatically set so that all displayed curves will fit into the graph. Turning off this option will select manual Y scaling.
- **Manual Y scaling:** When automatic Y scaling is turned off, the Y range can be entered here.
- **Clocked X axis:** Selects if the X axis shall be continued using an internal timer (turned off by default). This can be useful when event triggered PDOs are to be displayed. For time triggered PDOs or synchronized PDOs it is mostly better to turn off this feature.
- **Clock period:** When clocked X axis is turned on the clock period can be entered here.
- **Clear:** Clears the entire graph.

If the PDO definitions have been changed on the device, click the *Reload Definitions* button. The tool will then read all TPDO definitions from the device once again. Please note that this will also reset all settings for the graphical view.

## 5.9 PDO Transmitter

Using the PDO Transmitter tool, PDOs can be sent to a slave device in an easy and convenient way. When opened for the first time, the PDO Transmitter tool reads out all RPDO definitions from the device. All RPDOs are shown in the *PDO Selector* area on the left hand side of the window. Click on an RPDO to select that one. Inactive RPDOs are shown in gray colour and cannot be selected. After an RPDO has been selected, all the objects mapped to that PDO are shown in the *PDO Data* area on the right hand side of the window. For each object the desired value can be entered, either as a decimal number or as a hexadecimal number (starting with "0x" like in C language). After you have entered a value for each of the mapped objects, click the *Send PDO* button to send out the PDO.

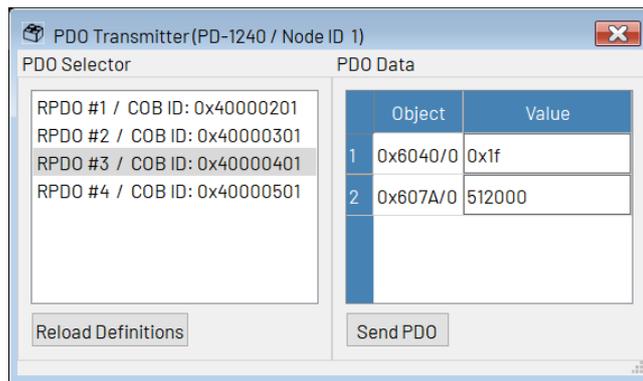


Figure 19: The PDO Transmitter Tool

If the PDO definitions have been changed on the device, click the *Reload Definitions* button. The tool will then read all RPDO definitions from the device once again.

### 5.10 PDO Configurator

The PDO Configurator is a tool for easily changing all PDO definitions of a device. When opened for the first time, the PDO Configurator tool reads all RPDO definitions and all TPDO definitions from the device. The window is divided into an RPDO tab and a TPDO tab.

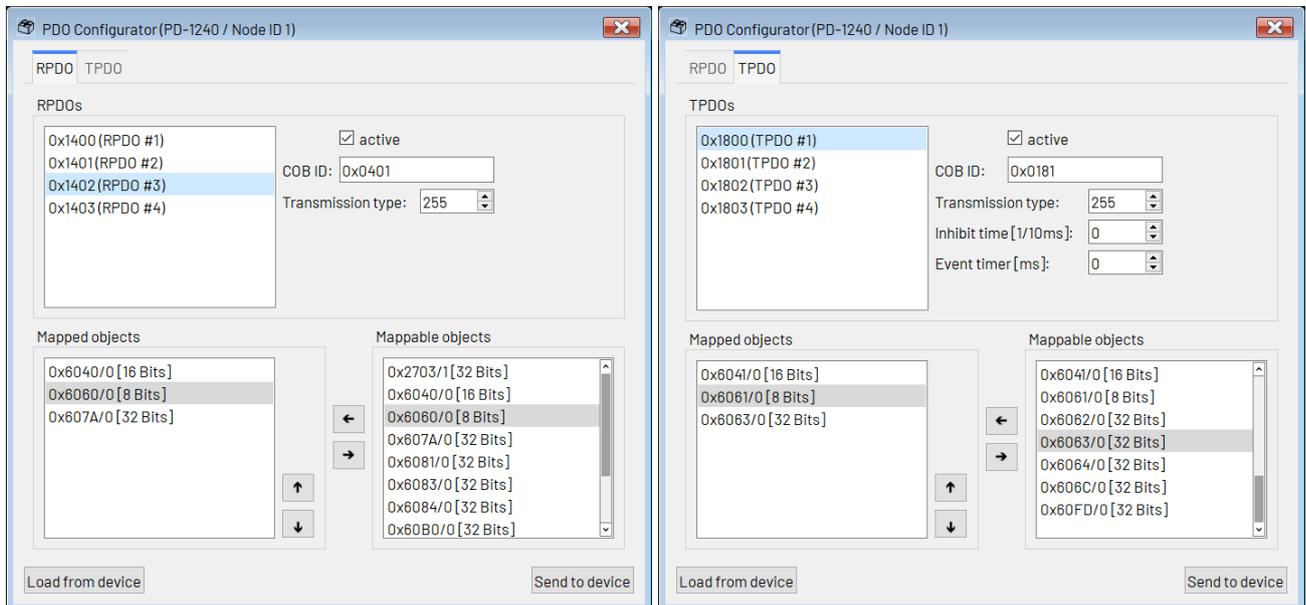


Figure 20: The PDO Configurator Tool

Use the RPDO tab to view and to edit the RPDO definitions. In the upper area of the RPDO tab, all RPDOs supported by the device are shown. Select an RPDO from the list to view and to edit it. Beside the RPDO list, all communication settings of the selected RPDO are shown and can be changed:

- **Active:** use this checkbox to activate or deactivate the selected RPDO (this will internally change bit 31 of its COB-ID).

- *COB-ID*: this is the COB-ID of the selected RPDO. Only the lower 11 bits are shown and can be set here.
- *Transmission type*: the transmission type of the selected RPDO.

The lower area of the RPDO tab presents two lists: one on the left side that shows all objects that are currently mapped to the selected RPDO (*Mapped objects*) and one that shows all objects that can be mapped to RPDOs (*Mappable objects*). To map an object to the selected RPDO, select the desired object from the list of mappable objects. Then click the left arrow button or double click the object. The object will then appear also on the list of mapped objects. Please note that the object will not be removed from the list of mappable objects, as each object can also be mapped more than once. To remove an object from the selected RPDO, select it in the list of mapped objects and click the right arrow button or double click the object. It will then be removed from the list of mapped objects. Use the up and down arrow buttons for changing the order of the mapped objects in the list. The currently selected object in the list of mapped objects can be moved up or down using these two buttons.

Use the TPDO tab to view and to edit the TPDO definitions. In the upper area of the TPDO tab, all TPDOs supported by the device are shown. Select a TPDO from the list to view and to edit it. Beside the TPDO list, all communication settings of the selected TPDO are shown and can be changed:

- *Active*: use this checkbox to activate or to deactivate the selected TPDO (this will internally change bit 31 of its COB-ID).
- *COB-ID*: this is the COB-ID of the selected TPDO. Only the lower 11 bits are shown and can be set here.
- *Transmission type*: the transmission type of the selected TPDO.
- *Inhibit time*: the inhibit time of the selected TPDO (given in 1/10ms).
- *Event timer*: the event timer of the selected TPDO (given in ms). Please bear in mind that the transmission type must be set to 254 or to 255 in order to be able to use an event timer value greater than zero.

The lower area of the TPDO tab presents two lists: one on the left side that shows all objects that are currently mapped to the selected TPDO (*Mapped objects*) and one that shows all objects that can be mapped to TPDOs (*Mappable objects*). To map an object to the selected TPDO, select the desired object from the list of mappable objects. Then click the left arrow button or double click the object. The object will then appear also on the list of mapped objects. Please note that the object will not be removed from the list of mappable objects, as each object can also be mapped more than once. To remove an object mapping from the selected TPDO, select it in the list of mapped objects and click the right arrow button or double click the object. It will then be removed from the list of mapped objects. Use the up and down arrow buttons for changing the order of the mapped objects in the list. The currently selected object in the list of mapped objects can be moved up or down using these two buttons.

After having made all necessary changes, click the *Send to device* button to send all changed PDO communication parameters and mappings to the device. Please note that only those parameters that have been changed will be sent to the device. During this step, all values sent are also written into the internal DCF file buffer, so that after this step the new values can also be saved to a DCF file (using the [Device Configuration](#) tool).

The *Load from device* button reloads all PDO definitions from the device. Use this function with care, as all changes made in the PDO Configurator which have not yet been sent to the device will be lost by reloading the PDO definitions from the device.

## 5.11 Device Configuration

Using the Device Configuration tool, DCF files (according to CiA-305 standard) can be generated, and DCF files can be used to configure devices.

### 5.11.1 Generating DCF Files

The *DCF Generation* tab page is used to generate DCF files. All necessary information according to the CiA-305 standard can be entered in the *File Description* and *Device Commissioning* section of this tab page. This information will be written to the DCF file, without being further processed by the TMCM-CANopen tool. When writing values to CANopen objects using the [Object Browser](#) or the [PDO Configurator](#), the last written value will always be written to an internal buffer. When generating a DCF file, all these last written values will also be written to the DCF file. If an object value is not to be written to the DCF file, uncheck its *Component of DCF file* checkbox in the [Object Browser](#) before generating the DCF file. Clicking the *Generate DCF File* button will start DCF file generation.

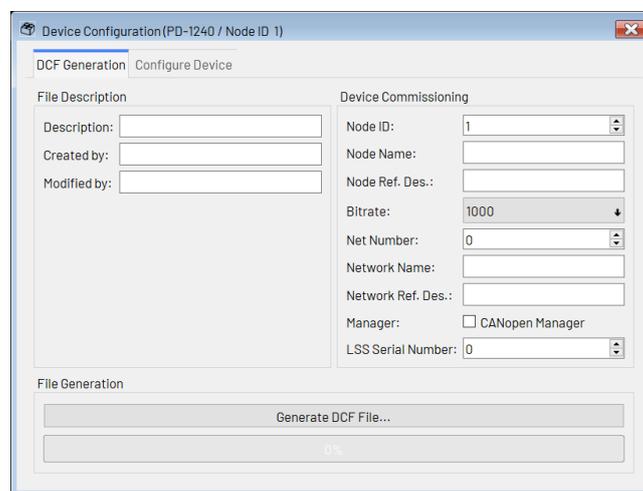


Figure 21: DCF File Generation

So the following procedure can be used to generate a DCF file:

1. Use the [Object Browser](#) to write the desired value to each object that is to be saved in the DCF file. Should you have written a wrong value, just write a new value to that object. If you wish not to save that object value in the DCF file at all, uncheck the *Component of DCF file* checkbox. Objects changed using the [PDO Configurator](#) will also be marked to be saved in the DCF file.
2. Open the Device Configuration tool and enter all necessary file information and device commissioning information.
3. Click the *Generate DCF File* button in the Device Configuration tool to generate the DCF file.

### 5.11.2 Using DCF Files

For using a DCF file to configure a module, select the *Configure Module* tab page of the Device Configuration tool.

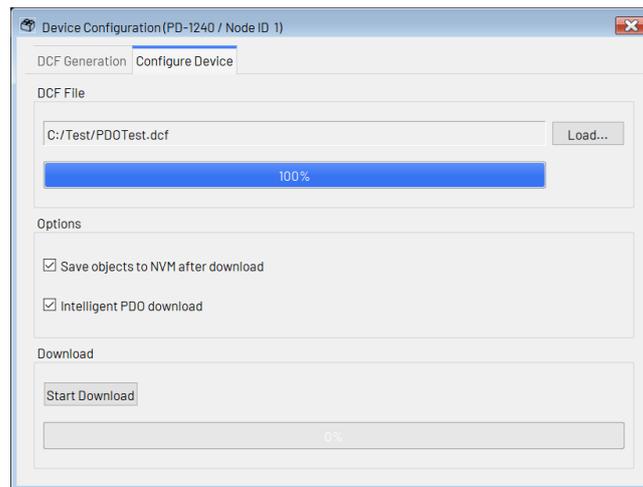


Figure 22: DCF File Usage

Click the *Load* button to load a DCF file. Set the options as necessary:

- *Save objects to NVM after download*: All objects will be saved to non volatile memory after download (if the device supports this). This is achieved by writing the save signature to object 1010<sub>n</sub> sub-index 1 after the download of all objects contained by the DCF file has completed.
- *Intelligent PDO download*: When this option is selected, PDO definition objects contained by the DCF file will be downloaded in such a way that PDO definitions will work correctly. If this option is not selected, all objects contained by the DCF file will be downloaded in numerical order. In most cases, this option should be switched on, so normally leave this checked.

Click the *Start Download* button to start the download of the values contained by the DCF file.

## 5.12 Expert Mode

The Expert Mode tool can be used to directly access CANopen objects on a device, without needing the information contained in the EDS file. This can be necessary either if there is no EDS file for the connected device or if objects are to be accessed that are not contained in the EDS file (for example when developing a new device).

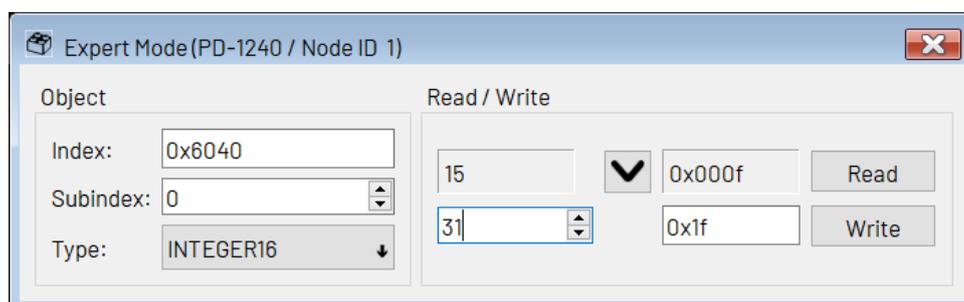


Figure 23: The Expert Mode

In the *Object* section on the left hand side of the window, the object index and sub-index can be entered. Also, the object type can be selected. When reading from an object the tool will try to automatically recognize the object type (even if there is an EDS file this tool will not use the information from it). Click

the *Read* button in the *Read / Write* area of the window to read the data from the object. The data will be shown as a decimal value in the left display field and as a hexadecimal value in the right display field. Clicking the  button will copy the contents of the display fields into the input fields. In the left input field, data can be entered as a decimal value only, whereas in the right input field either hexadecimal or decimal numbers can be entered. Whenever one of these input fields gets changed, the other one will also be changed to the same value. When the data type *VISIBLE STRING* or *OCTET STRING* is selected, only the right display field and the right input field will be used. In such cases, visible strings can be entered as text, and octet strings can be entered as hexadecimal bytes (two digits without preceding "0x") separated by spaces.

Clicking the *Write* button will write the data to the object.

## 6 Figures Index

1	The TMCM-CANopen Main Window . . . . .	5	11	The CiA-402 State Machine Tool . . . . .	15
2	The Menu Bar . . . . .	5	12	The CiA-402 Profile Position Mode Tool . . . . .	16
3	Tree with Connected Devices and their Tools . . . . .	6	13	The CiA-402 Profile Velocity Mode Tool . . . . .	16
4	Log Window showing CANopen Messages . . . . .	6	14	The CiA-402 Homing Mode Tool . . . . .	17
5	The CAN Bus Connection Window . . . . .	7	15	The Parameter Save/Restore Tool . . . . .	18
6	The LSS tab (with empty LSS slave device list) . . . . .	9	16	The PDO Receiver Tool . . . . .	19
7	The LSS device list containing two LSS devices and showing its context menu . . . . .	10	17	The PDO Entry Context Menu . . . . .	19
8	The UserEds directory . . . . .	11	18	The PDO Graph Context Menu . . . . .	20
9	The Main Module Window . . . . .	13	19	The PDO Transmitter Tool . . . . .	21
10	The Object Browser . . . . .	14	20	The PDO Configurator Tool . . . . .	21
			21	DCF File Generation . . . . .	23
			22	DCF File Usage . . . . .	24
			23	The Expert Mode . . . . .	24

## 7 Tables Index

1	List of supported CAN Interfaces . . .	7	3	Document Revision . . . . .	30
2	Software Revision . . . . .	30			

## 8 Supplemental Directives

### 8.1 Producer Information

### 8.2 Copyright

TRINAMIC owns the content of this user manual in its entirety, including but not limited to pictures, logos, trademarks, and resources. © Copyright 2022 TRINAMIC. All rights reserved. Electronically published by TRINAMIC, Germany.

Redistribution of sources or derived formats (for example, Portable Document Format or Hypertext Markup Language) must retain the above copyright notice, and the complete data sheet, user manual, and documentation of this product including associated application notes; and a reference to other available product-related documentation.

### 8.3 Trademark Designations and Symbols

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

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

### 8.4 Target User

The documentation provided here, is for programmers and engineers only, who are equipped with the necessary skills and have been trained to work with this type of product.

The Target User knows how to responsibly make use of this product without causing harm to himself or others, and without causing damage to systems or devices, in which the user incorporates the product.

### 8.5 Disclaimer: Life Support Systems

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

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

Information given in this document is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. Specifications are subject to change without notice.

### 8.6 Disclaimer: Intended Use

The data specified in this user manual is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose

or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given.

In particular, this also applies to the stated possible applications or areas of applications of the product. TRINAMIC products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (safety-Critical Applications) without TRINAMIC's specific written consent.

TRINAMIC products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by TRINAMIC. TRINAMIC conveys no patent, copyright, mask work right or other trade mark right to this product. TRINAMIC assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

## 8.7 Collateral Documents & Tools

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

## 9 Revision History

### 9.1 Software Revision

Version	Date	Author	Description
V1.3.0.0	2020-DEC-09	OK	Release version.
V1.3.0.1	2021-FEB-03	OK	DCF parameter download improved.
V1.4.0.0	2021-MAR-23	OK	LSS functionality added.
V1.5.0.0	2021-JUN-11	OK	PDO Configurator added. DCF parameter download: changed PDO communication parameters will also be downloaded when mappings of that PDO have not been changed. Send abort message on the CAN bus when an SDO timeout has occurred.
V1.6.0.0	2021-DEC-03	OK	Fixed: Sub-indices greater than 9 were not correctly read from the EDS file. Fixed: Homing mode tool was not correctly usable with multi-axis modules. PDO Configurator: Mapping/unmapping also possible using double click. PDO Receiver: Graphical view added. Parameter Save/Restore: Save/Restore device parameter buttons added.
V1.7.0.0	2022-APR-26	OK	Multiple connections possible with Kvaser CAN interface. Peak CAN interface can be used together with Peak CAN monitor. Ixxat CAN interface can be used together with Ixxat CAN monitor. Filtering of non CANopen messages possible. Format of CAN message logging renewed. Decoding of logged CAN messages. Expert mode: also possible to enter visible strings and octet strings.

Table 2: Software Revision

### 9.2 Document Revision

Version	Date	Author	Description
V1.00	2020-DEC-18	OK	First release of the manual for TMCM-CANopen V1.3.0.0.
V1.01	2021-MAR-03	OK	Release of the manual for TMCM-CANopen V1.3.0.1.
V1.02	2021-MAR-23	OK	Release of the manual for TMCM-CANopen V1.4.0.0.
V1.03	2021-JUN-11	OK	Release of the manual for TMCM-CANopen V1.5.0.0.
V1.04	2021-DEC-03	OK	Release of the manual for TMCM-CANopen V1.6.0.0.
V1.05	2022-APR-26	OK	Release of the manual for TMCM-CANopen V1.7.0.0.

Table 3: Document Revision