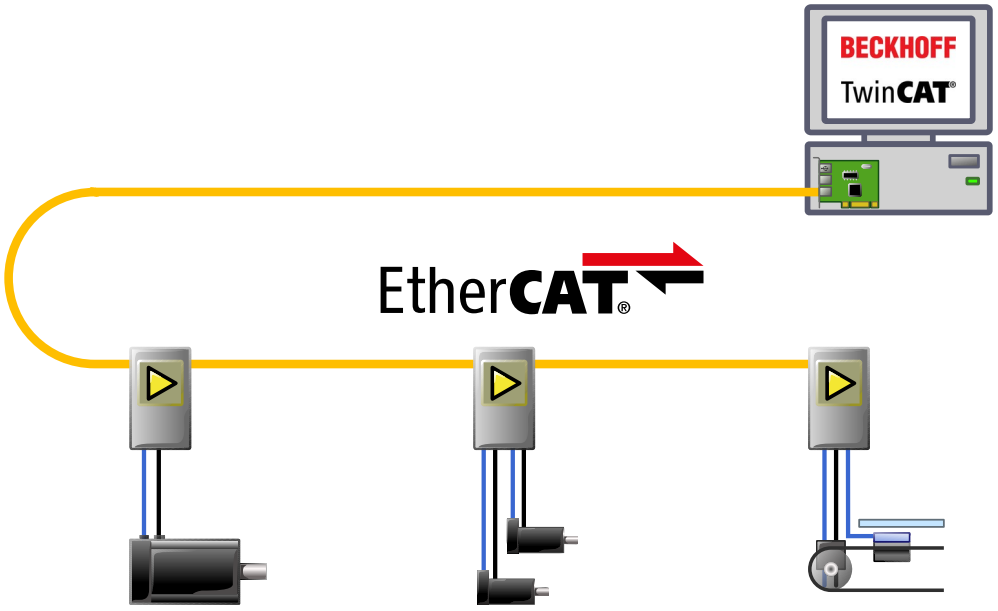
TwinCAT Setup Guide

EtherCAT Fieldbus

This user guide explains how to use *Beckhoff TwinCAT NCI* or *CNC* in conjunction with *Triamec EtherCAT* drives.



Document	SWTC_TwinCAT-UserGuideEcat_EP
Version	018, 2024-06-17
Source	Q:\Doc\Software\SWTC\SWTC_TwinCAT-UserGuide\
Destination	T:\doc\SW\
Owner	mvx

www.triamec.com

Table of Contents

1 Configuring the Drives

Before a drive can be used with *TwinCAT*, the drive has to be configured with the TAM System Explorer. Refer to the setup guide [1] on how to setup the drive parameters, install new firmware and how to save the parameters persistently on the drive.

Remarks:

- For configuration the drive has to be connected with the PC through a USB or Ethernet connection and the TAM System Explorer has to be installed on the connected PC. It is recommended to set **File > Preferences > Startup > Acquired Adapters** to **Triamec devices over USB** or **Triamec Devices over Ethernet** to accelerate booting. The *EtherCAT* control system PC does not require an installation of the TAM System Explorer if it is not intended to be used for configuration or debugging of the drive.
- When using the TAM System Explorer to configure the drive, the drive must not follow the *EtherCAT* commands. Therefore set Axes[].Commands.General.OverwriteControlSystem to 1 or, when using the *Axis Module*, use the button **Attach** to set the *OverwriteControlSystem* flag. Don't forget to reset the command when the configuration is done.
- Set Axes[].Parameters.PositionController.PositionUnit for each axis for proper position scaling as discussed in chapter 5.1.
- To not loose the settings on power loss, save the configuration persistent on the drive. Later changes of the configuration need to be persisted again to not be lost after a power-down. Also save the configuration as a `*.TAMcfg` file on the PC.

Note It's recommend to setup the axes on the drive in the same units as used on the *TwinCAT* side (i.e. mm or degrees).

2 Preparation of TwinCAT and DC

Triamec provides a *Release Package* for *TwinCAT* integration over *EtherCAT* on the website [6].

Find the *EtherCAT Slave Information* (*ESI*) file `TriamecMinFw4.22.xml` (or newer) inside the release package and copy it to `C:\TwinCAT\3.1\Config\Io\EtherCAT`. If *TwinCAT* was already open, choose **TwinCAT > EtherCAT Devices > Reload Device Descriptions**.

Open a new *TwinCAT* Project and add an *EtherCAT* Adapter.

1. Right-click on the **IO > Devices** node and choose **Add new item...**
2. Add a **EtherCAT > EtherCAT Master** type device.

If there is no *EtherCAT* driver installed yet, use the menu **TwinCAT > Show Realtime Ethernet Compatible Devices** in the *TwinCAT* environment.

Setup the Distributed Clock settings for the *EtherCAT Master* at **EtherCAT Master > EtherCAT > Advanced Settings > Distributed Clocks**. Set the checkbox to **DC in use** and the radio button to **Independent DC Time** (see Figure 1).

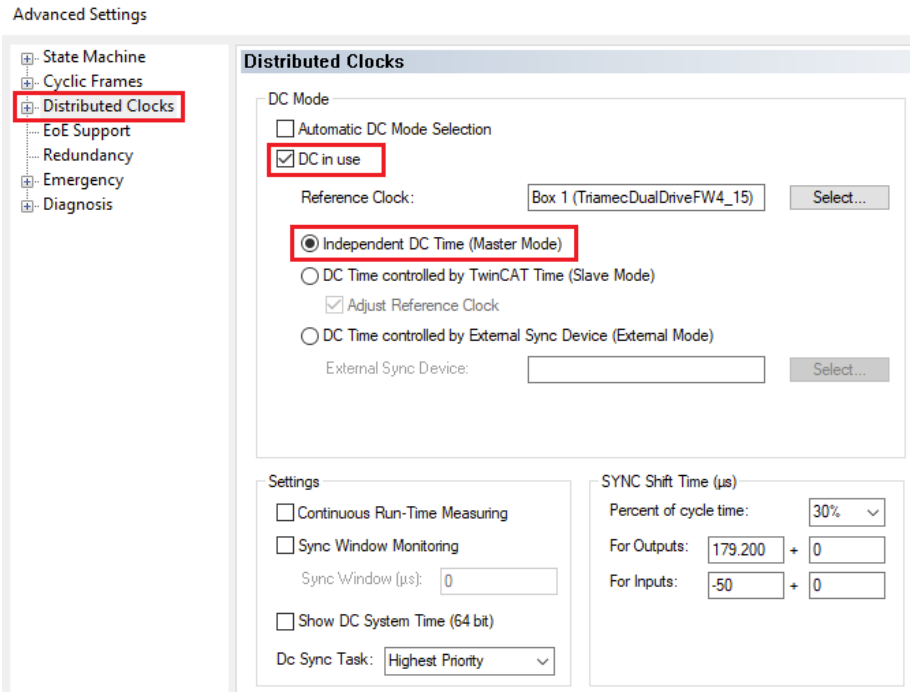


Figure 1: DC Mode settings on EtherCAT Master

Make sure the cycle time of Distributed Clock (DC) Sync is a multiple of 100µs. Choose for example

- **System > RealTime > Settings > BaseTime** 500µs
- For *NCI*: Set **MOTION > NC-Task1 SAF > Task > Cycle ticks** = 1 500µs (must be n* 100µs)
- For *CNC*: Set **MOTION > CNC Task GEO > Task > Cycle ticks** = 1 500µs (must be n* 100µs)

Consider the sync shift settings at **EtherCAT Master > EtherCAT > Advanced Settings > Distributed Clocks > SyncShiftTime** if the jitter on the host PC is large.

3 Drive Setup in TwinCAT

On the *EtherCAT* master, select **Add New Item...** from the context menu and find the corresponding Type at **Triamec Motion AG > Drives**. Then choose to which configuration the new Axes should be linked to (Figure 2).

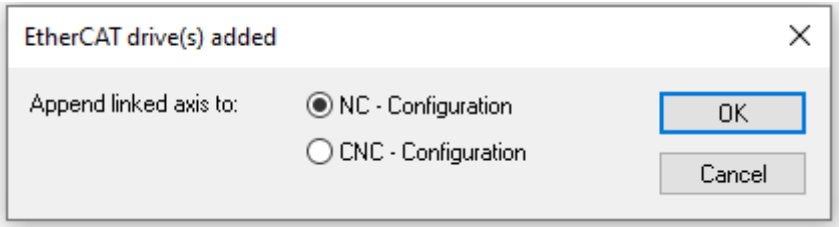


Figure 2: Axis link configuration dialog

- Select **NC - Configuration**, if you intend to use the *NCI* module.
- Select **CNC - Configuration**, if you intend to use the *CNC* module.
- Click **Cancel**, if you add a drive for already present *NC* axes.
  Then link the axes using **MOTION > Axis > Configuration > LinkTo...**

Finally check the setting **Use as potential Reference Clock** at **Drive > EtherCAT > AdvancedSettings > DistributedClock** (Figure 3).

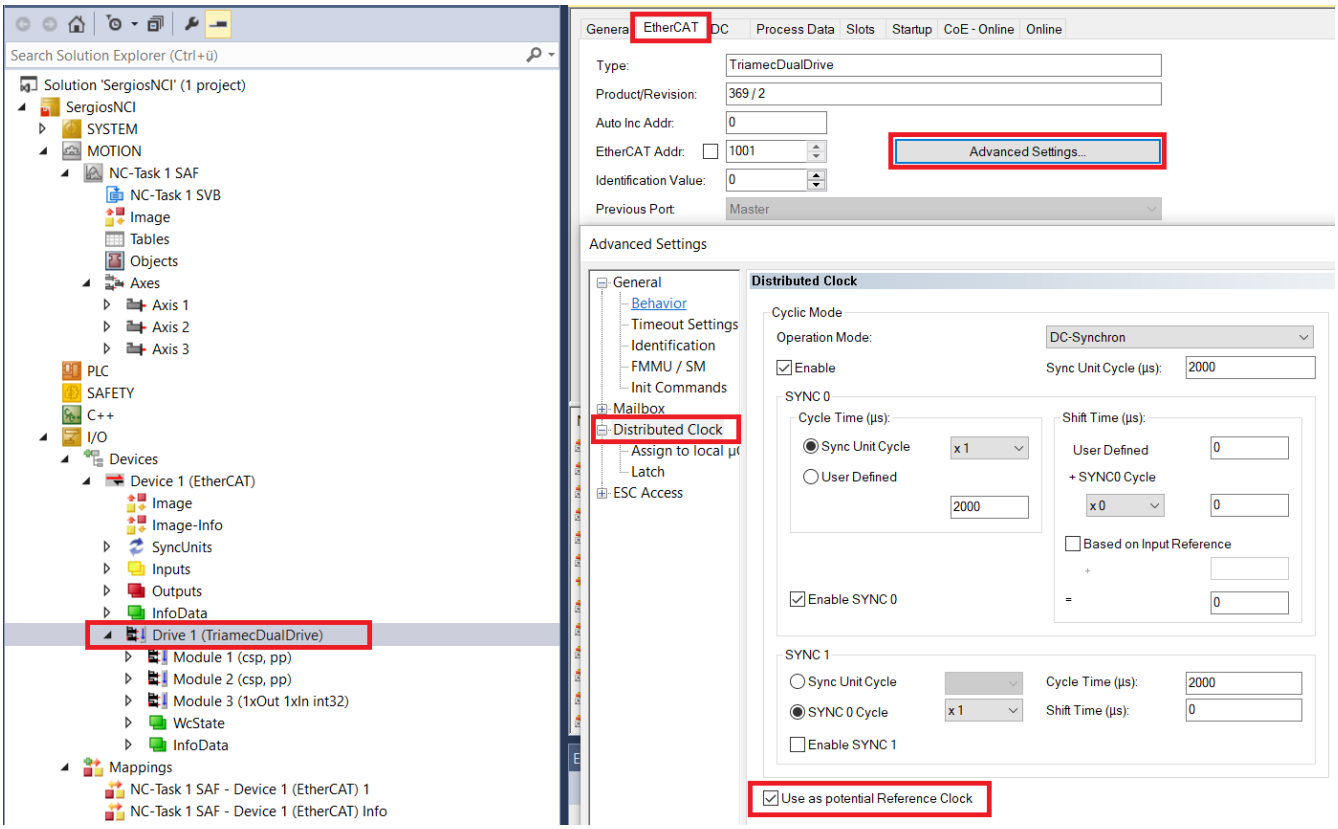


Figure 3: Setting Drive as potential Reference Clock

3.1 Using the NCI

If you chose to use an *NC* configuration in chapter 3, a new *NC* axis was added for every axis in the drive and connections were made between the drive and the *NC*. Before using the axis, configure at least the following Parameters at **MOTION > NC-Task SAF > Axes > Axis N:**

At **MOTION > NC-Task SAF > Axes > Axis N:**

- **Settings > Unit**
- **Parameter > Maximum Dynamics**
- **Parameter > Default Dynamics**
- **Parameter > Other Settings > Couple slave to actual values if not enabled** = **TRUE**

Then choose **TwinCAT > Activate Configuration** to run *TwinCAT*.

The **NC > Axis > Online** interface should then look like in Figure 4.

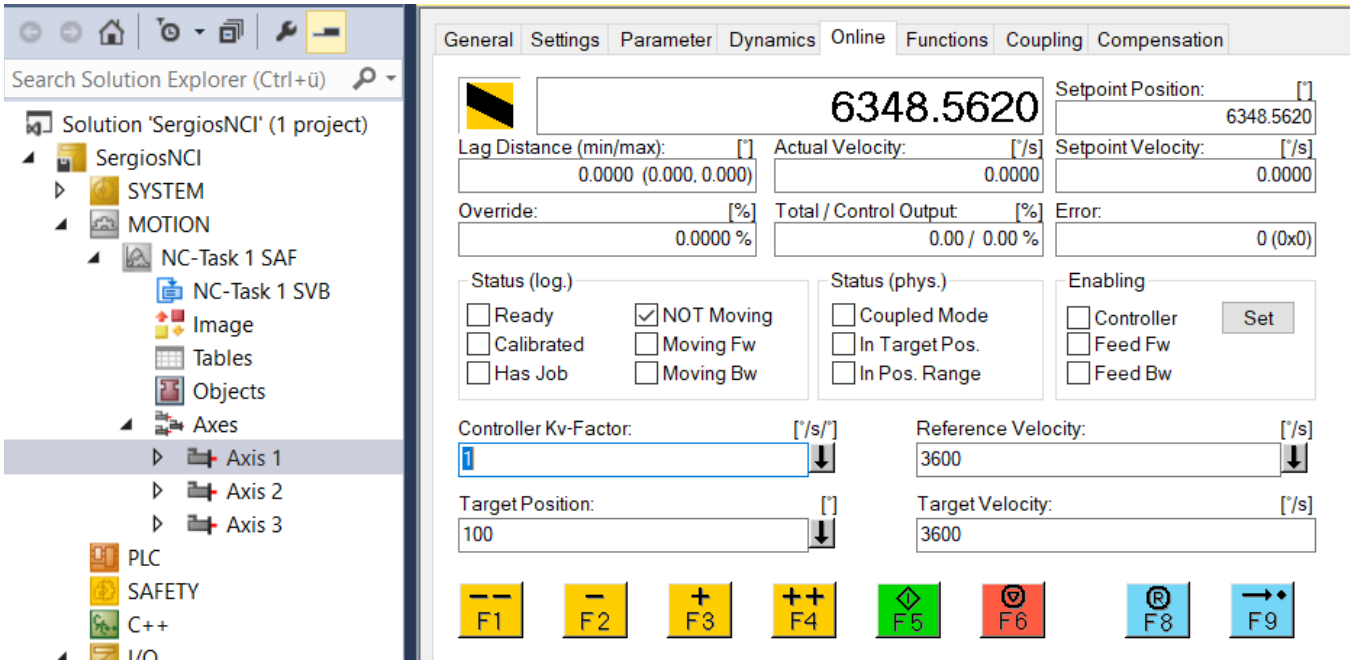


Figure 4: Online interface of NC axis

First, all checkboxes in **Enabling** are off. You can move the axis by hand and see the actual position reflect its position. If the actual position is grayed out, the axis state is not valid. Make sure the drive is in the *EtherCAT* operational state (see **I/O > Drive > Online**) and the drive is not in an error state, see also cyclic signal **MessageId** (objects 0x603F for axis 0 and 0x683F for axis 1).

Now enable the axis by clicking the **Set** button in the **Enabling** frame. Then press **All** (Figure 5).

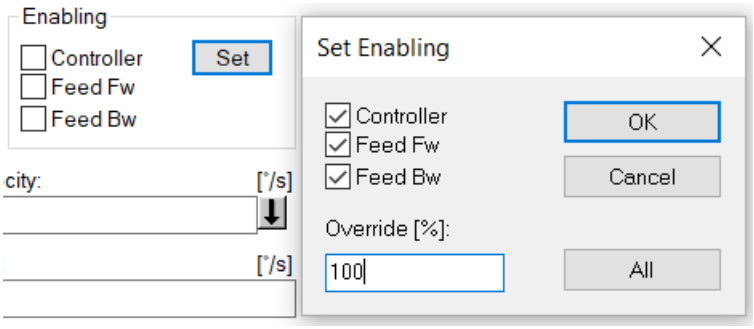


Figure 5: Set Enabling dialog

Press **+ (F2)** and **- (F3)** to move the axis and check the physical axis and the change of the actual position display.

3.2 Using the CNC

If you chose to use a *CNC* configuration in chapter 3, a new *CNC* axis was added for every axis in the drive and connections were made automatically. Otherwise do the following steps.

- Attach the *CNC* using **Add New Item...** from the context menu of the **MOTION** node, if not already present.
- Attach an axis using the context menu of **MOTION > CNC > Axes** and select **Add new item...**
- Select the drive axis using the **Link To...** dialog in the **Configuration** tab of the Axis.

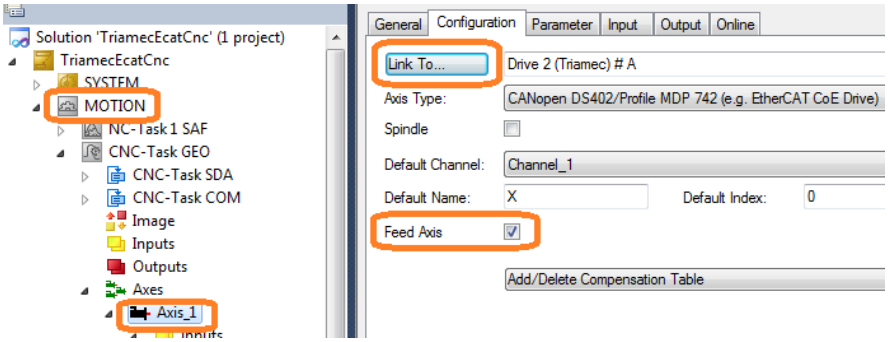


Figure 6: Link Axis to CNC

For both, manually added and automatically added axes, do the following steps.

- First attach a **Channel** using the context menu of **MOTION > CNC** and select **Add new item...**
- Check the checkbox for each axis at **MOTION > CNC > Axis > Configuration > Feed Axis**, and
- configure the **Default Channel**.
- In the **Parameter** tab of each axis, add the following entry using the **Notepad…** button. This ensures proper functionality of the *HMI* with *Triamec* devices.

```
customer.val[0]                    0x100            ( P-AXIS-00510 : Free user defined values )
```

Choose **TwinCAT > Activate Configuration** to run *TwinCAT*.
Each axis can now be moved by hand and the actual position should change correctly in the **MOTION > CNC > Channel > Online** tab.

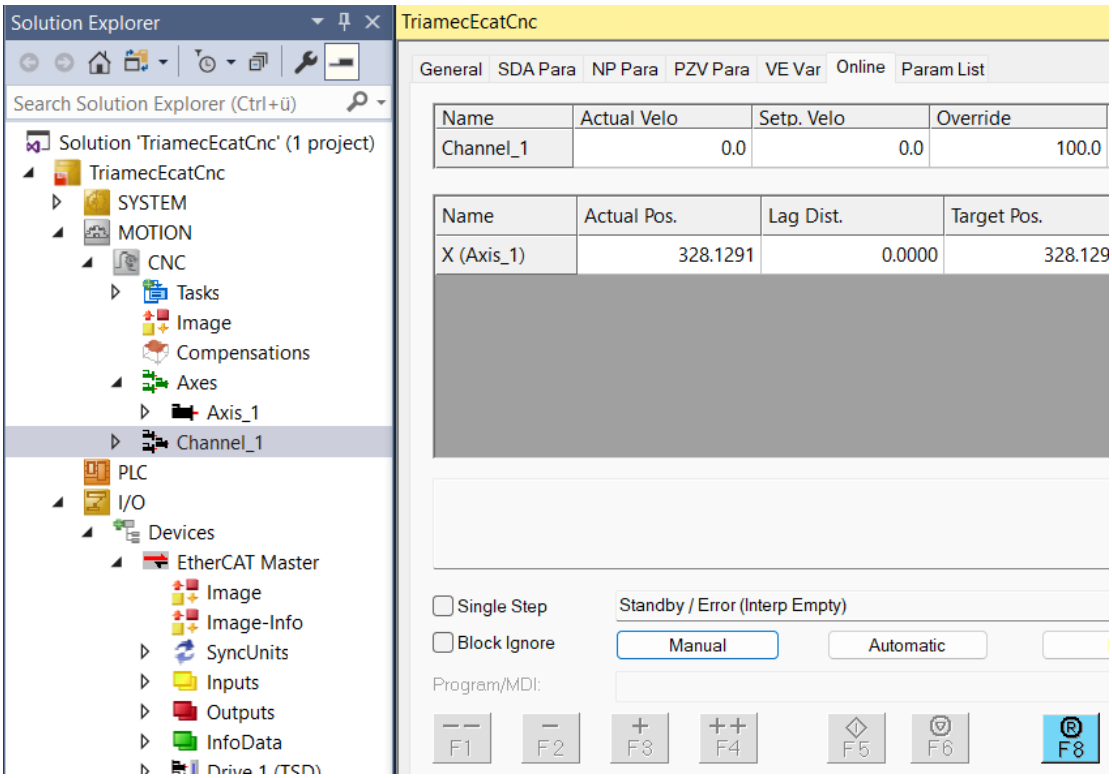


Figure 7: Channel Online View

Enabling the axes requires an *HLI* command to the *CNC*. Open the *CNC* channel control window: **MOTION > GEO > Channel > Online**.

Switch to operation mode **Manual**. Select the first axis. Press **+ (F2)** and **- (F3)** to move the axis and check the physical axis and the change of the actual position display.

Note If the drive enters an error state, the *CNC* will also enter its error state but it will remain at `ModeOfOperation = 8` and `ControlWord = 0x1F`.
Always clear errors using the *CNC Reset*.

If a drive error is cleared using direct USB or Ethernet access, the drive will clear its error and instantly go back to coupled state, following the commanded positions from the *CNC*. The *CNC* on the other side is still in error state and will set the commanded position to the actual position of the drive. This chasing condition can lead to errors or noise.

4 Triamec Library

An open source library is available from the *Release Package* [6]. It contains optional, but handy helper functions to access drive registers and control axes from the PLC program.

Example usage of selected library functions is shown in the *Triamec* example solutions, available on *GitHub* [7].

Extract the library file `TriamecEcat<version>.library` from the *Release Package* Zip and install it in *TwinCAT*: Go to **PLC > Library Repository…**, click **Install**, and browse to the file mentioned above.

5 Positions

5.1 Position Scaling

EtherCAT uses increments (integer32) for positions, whereas *Triamec* drives use the double format representing SI units (***meter*** or ***degree*** or similar). Therefore the drives need to convert double to integer.

The standard scale factor of *EtherCAT* is 10'000 inc/mm or 10'000 inc/degree. Since the conversion factor depends on the axis type (linear or rotational), the drives need to know their axis unit. Specify the axis unit using Parameters.PositionController.PositionUnits in the drive axis configuration.

To change the default scaling, use the startup parameter **0x6092:01** for the first axis and **0x6892.01** for the second axis. Its units are inc/mm or inc/degree. It is part of the startup parameter list as shown below. The standard is 10'000. To get for example 1nm resolution, choose the value 1'000'000. Be aware that this reduces the maximum possible position to 2^{31}*1nm = 2.1m.

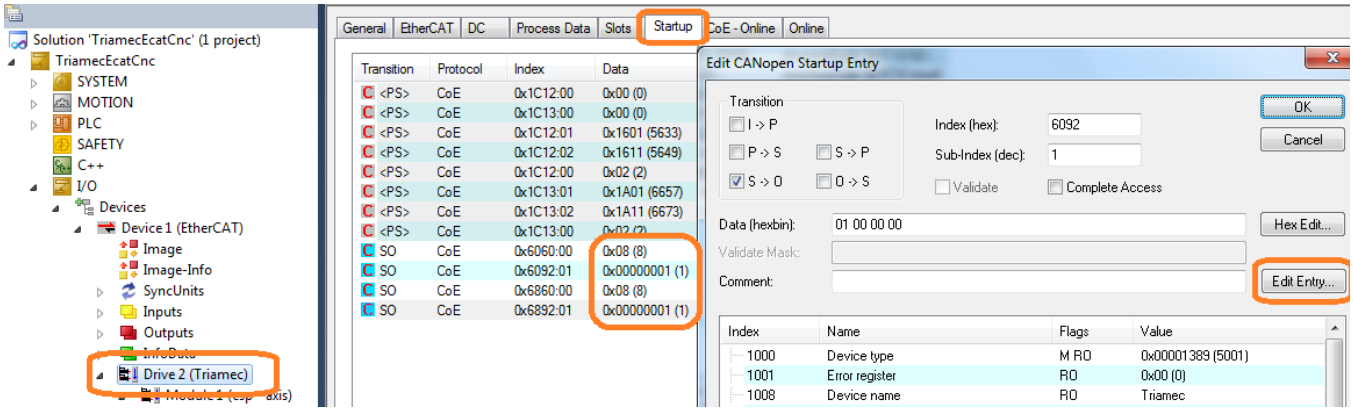


Figure 8: Change Scaling at EtherCAT Interface

This changes the scale at the interface. The *TwinCAT* path planner modules must also know this scaling factor.

Note It is recommended to select the highest possible resolution (e.g. 1nm ≙ 1'000'000). If your working range exceeds the 2.1m mentioned above, select an appropriately adapted resolution that covers your area.

CNC

Adapt the *CNC* axis parameter (Figure 9)

```
getriebe[0].wegaufz      100      ( P-AXIS-00234 : Path resolution of the measuring system (num)
```

The unit of this parameter is 10'000 inc/mm or inc/degree. Its default is 1. Use this equation for the values of the drive and CNC parameters

0x6092.01 = wegaufz * 10'000

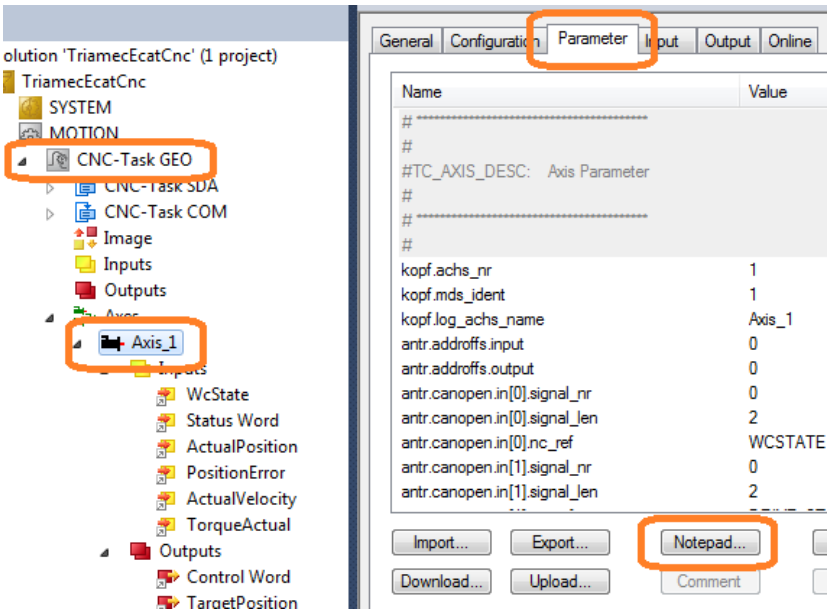


Figure 9: Change Resolution of Scale in CNC

NCI

For a resolution of 1nm:

- Set the *Feed* parameter to 1'000'000 as described above.
- Set **MOTION > NC > Axes > Axis N > Enc > Parameter >**
  Scaling Factor Numerator = 0.000001 (1/*Feed* parameter)
  Scaling Factor Denominator = 1 (default)
- **Save** and **Activate** the configuration to apply the changes.

5.2 CSP Mode

The standard mode of operation is CSP (Cyclic synchronous position), where commanded positions are expected cyclically. There are three interpolation modes, specified with the following drive register:

Axes[].Parameters.PathPlanner.StreamInterpolatorMode

- Polynom3A A 3rd order recursive polynom into the last three positions
- Polynom4A A 4th order polynom exactly fitting into the last five positions (legacy)
- BSpline3A A cubic B-Spline of the last three positions

Choose Polynom3A in standard situations with a smooth target path. Choose BSpline3A if the path trajectory is not smooth.

5.3 Modulo and Spindle

If the Triamec parameters Axes[].Parameters.PathPlanner.ModuloPositionMaximum and ModuloPositionMinimum are unequal to 0.0, the axis runs in modulo mode. In CSP mode, it expects commanded positions within this range. The actual positions from the encoders will wrap as soon as

the respective commanded positions wrap.

The commanded velocity is not considered directly for path generation in *CSP* mode. However, If the velocity is provided, the drive can evaluate the correct modulo wrap. See *CSV* mode for the required velocity scaling.

I.e., if The NC-rate is 1ms and the target speed is larger than 30’000 turns/min. The drive will face more than half a modulo per position update and it cannot resolve correct positions, if velocities are not provided.

Drive

Set the modulo ranges, i.e. for an axis with unit degrees.

- Axis.Parameters.PathPlanner.ModuloPositionMaximum = 360
- Axis.Parameters.PathPlanner.ModuloPositionMinimum = 0.0

CNC

For details on CNC axis, spindle, and channel settings see the Beckhoff or CNC documentation. If the axis is rotational with the CNC in units of 0.0001° and should get a 360° Modulo, set:

```
kenngr.achs_typ                 ROTATOR      (P-AXIS-00018 Type, 2 (ROTATOR) or 4 (SPINDLE))
kenngr.achs_mode                0x4          (P-AXIS-00015 Modulo)
getriebe[i].moduloo             3600000      (P-AXIS-00126 : [10-4degree] Upper modulo limit)
getriebe[i].modulou             0            (P-AXIS-00127 : [10-4degree] Lower modulo limit)
```

For CNC spindle mode consider Axis-Parameters (over-write some settings above). The 720° modulo allows faster speed at slow update rates of the position:

```
kenngr.achs_typ                 SPINDLE      (P-AXIS-00018 Type, 2 (ROTATOR) or 4 (SPINDLE))
kenngr.vb_prozent               100          (P-AXIS-00217 [0.1 %]Factor for speed reached)
getriebe[0].vb_min_null         1000         (P-AXIS-00216 [10-3Grad/s] Drehzahl null (Spindel))
getriebe[0].beschl_kennlinie.typ 0           (P-AXIS-00202 Type of the acceleration curve )
getriebe[i].moduloo             7200000      (P-AXIS-00126 : [10-4degree] Upper modulo limit)
antr.mode_act_pos               2            (P-AXIS-00122)
antr.mode_cmd_pos               2            (P-AXIS-00123)
antr.drive_encoder_range        3600000      (1)
```

and channel parameters:

```
spdl_anzahl                     1            (P-CHAN-00082 Spindelanzahl)
main_spindle_ax_nr              1            (P-CHAN-00051 Logical axis number main spindle)
main_spindle_name               S            (P-CHAN-00053 Bezeichnung der Hauptspindel)
spindel[0].log_achs_nr          1            (P-CHAN-00036 Logical axis number of the Spindel)
spindel[0].bezeichnung          S1           (P-CHAN-00007 Bezeichnung der Spindel)
cax_face_id                     2            (Fräsmaschinen)
main_spindle_gear_change        0
```

1 This value has to be scaled with getriebe[..].wegaufz and getriebe[..].wegaufn.

NCI

Set the following values at **NCI > Axis > Enc > Parameters**

- **Scaling Factor Numerator** = 0.000001 (same as 1/*Feed* parameter, chapter 5.1)
- **Scaling Factor Denominator** = 1
- **Modulo Factor** = 360 (same as *Drive* setting)
- **Encoder Mask** = 0x36EE80 (*modulo * Feed* = 3600000)
- **Encoder Sub Mask** = 0xFFFFFF

Check also the *Beckhoff* documentation on using Modulo.

and in **NC > Axis > Drive > Parameter**:

- Set **Output Scaling Factor (Velocity)** = 2000
- Set **Axis > Settings > Unit** = **Degree**
- Activate the checkbox **Axis > Settings > Modulo**.

5.4 CSV Mode

The standard mode of operation is CSP Mode (8), where cyclic positions are commanded at the interface. With FW Release 4.5.0 CSV mode (9) is supported too, where cyclic velocities are commanded.

The resolution of the velocities are scaled by the same factor as for CSP mode: If for a rotational axis

- the feed object 0x6092.01 is 10000 inc/degree,
- a position of 10000 inc will correspond to ***1 degree*** and
- a velocity of 10000 inc will correspond to ***1 degree/s***

The drive side interpolation uses cubic B-splines in the target velocity and integrates these to a 4th order polynom in the position. To facilitate switching back from CSV mode to CSP mode, the signal General/Signals/EtherCAT/targetPositionError is provided, which is the deviation between expected position and commanded position.

CNC

Configuring velocity mode for a spindle, the axis parameter *P-AXIS-00320 "OPEN_POSITION_LOOP_MODE"* can be set. Find more information in [8].

```
antr.operation_mode                OPEN_POSITION_LOOP_MODE
```

NCI

An Beckhoff function block `MC_WriteDriveOperationMode` in PLC Library `Tc2_MC2` can be used to initiate the change of the operation mode. For further information see Beckhoff documentation.

5.5 CST Mode

The standard mode of operation is CSP Mode (8), where cyclic positions are commanded at the interface. With FW Release 4.22.0 CST mode (10) is supported too, where cyclic torque is commanded.

Switchover from and to CST mode (10) is only performed when the axis is disabled. The active mode is indicated by the object `0x6061 Mode of Operation Display`. The change from/to CST mode can be commanded with the object `0x6060 Mode of Operation`.

The object `0x6071 Target torque` must be written cyclically with the torque set-point. If the axis is enabled, the target torque will become active immediately and the movement will start. The value is specified in relation to the maximum motor current, whereby a value of 32767 (0x7FFF) corresponds to the value entered in the Axes[].Parameters.Motor.PeakCurrent register.

Torque offset option in CSP/CSV Mode

To use the `0x6071 Target torque` object as a torque offset in CSP/CSV operating mode, the following option can be activated:

Axes[].Parameters.General.Options = 0x00004000

5.6 Referencing (Homing)

Absolute encoders return the correct position without a reference move. When using *TwinCAT CNC*, make sure to set the following parameters to avoid offsets between *CNC* and drive positions.

kenngr.set_refpos_mode	OFFSET
kenngr.set_refpos_offset	*0*

Drive controlled homing is described in [3]. The *TwinCAT Example Project* [7] contains the usage of this mode.

Reference moves controlled by *TwinCAT* are used to determine the position difference between drive position and *TwinCAT* position. This is then taken into account by adding an offset between the interface data and the display data. The drive positions are not referenced in this case.

6 Control and Status Word

6.1 Control Word (0x6040)

Some of the *Control Word* bits depend on the mode of operation.

bit	***Function***	***CSP** (8) / **CSV** (9)*	***CST (10)***	***PP** (1) / **PV** (3)*	***HM** (6)*
0	Switch on				
1	Enable voltage				
2	Quick stop				
3		Enable operation			
4				New set-point	Start homing
5		Output cycle counter		Change set immediately	
6				Relative target value	
7	Fault reset				
8				Halt	Halt
9				Change on set-point	
10					
11				Limit velocity acceleration (*)	

Bit 2 `Quick stop` shall be set (=1) by the control device to deactivate an optional quick stop. *Triamec* does not support the `Quick stop` yet.

(*) In PP mode, bit 11 limits velocity and acceleration, not only by the *Triamec* path planner parameters, but also by the cyclic velocity and acceleration: The velocity limit is taken from the cyclic `0x60FF Target velocity` word in inc/s. I.e. if the feed constant is 10000, a value of 20000 corresponds to 2 mm/s for linear axes or 2°/s for rotational axes. The acceleration value must be commanded to the drive by an additional cyclic telegram. Set up a *Subscriber*, as described in 7.4, to the *URI Address* below, according to the drive axis. The value must be sent in drive units.

- Axes[0].Commands.PathPlanner.Anew = 0x00227D00
- Axes[1].Commands.PathPlanner.Anew = 0x002A7D00

6.2 Status Word (0x6041)

The *Status Word* bits also depend on the mode of operation.

bit	***Function***	***CSP** (8) / **CSV** (9)*	***CST** (10)*	***PP** (1) / **PV** (3)*	***HM** (6)*
0 1 2	Ready to switch on Switched on Operation enabled				
3	Fault				
5	Quick stop (1 = is not active/not supported)				
6	Switch on disabled				
7	Warning				
8 (*)	Follow me (Triamec)				
9	Remote (local=0)				
10		Toggle status (not implemented)		Target reached	
11	Internal limit active				
12		Drive follows the command value (Kap 6.2 of ETG6010_V1i1i0)		Set-point acknowledge	Homing attained
13		Extended toggle (not implemented)		Following error	Homing error
15 (*)	Reference done (Triamec)				

(*) Description of the *Triamec* specific bits:

`Bit8` indicates that the drive actual position does not follow the target position. Instead, the control system is expected to track the actual position by setting target=actual.

`Bit15` indicates that referencing has been processed and is valid.

Statusword for profile position mode (PP)

Bit	***Value***	***Definition***
10	0	Halt (Bit 8 in controlword) = 0: Target position is not reached Halt (Bit 8 in controlword) = 1: Axis decelerates
	1	Halt (Bit 8 in controlword) = 0: Target position reached Halt (Bit 8 in controlword) = 1: Velocity of axis is 0
12	0	Previous set-point already processed, waiting for new set-point
	1	Previous set-point still in process, set-point overwriting shall be accepted
13	0	No following error
	1	Following error

Statusword for profile velocity mode (PV)

Bit	*Value*	*Definition*
10	0	Halt (Bit 8 in controlword) = 0: Target not reached Halt (Bit 8 in controlword) = 1: Axis decelerates
	1	Halt (Bit 8 in controlword) = 0: Target reached Halt (Bit 8 in controlword) = 1: Velocity of axis is 0
12	0	Speed is not equal 0
	1	Speed is equal 0
13 (*)	0	No following error
	1	Following error

(*) Even in PV mode, the internal position controller of the Triamec drive is active and therefore monitoring the position error.

Statusword for homing mode (HM)

Bit 13	*Bit12*	*Bit10*	*Definition*
0	0	0	Homing procedure is in progress
0	0	1	Homing procedure is interrupted or not started
0	1	0	Homing is attained, but target is not reached
0	1	1	Homing procedure is completed successfully
1	0	0	Homing error occurred, velocity is not 0
1	0	1	Homing error occurred, velocity is 0
1	1	X	reserved

6.3 Start Sequence

A typical start sequence for the mode of operation *CSP (8)* or CSV (9) is:

Status Word	***Control Word***	
0x121	0x0 0x2	ReadyToSwitchOn EnableVoltage
0x121	0x3	ReadyToSwitchOn EnableVoltage+SwitchOn
0x127	0xB	Enabled EnableOperation
0x1027		CoupledMotion

Please note the following behavior of target position versus actual position in this sequence: As long as bit8=followMe is set, the drive actual position does not follow the target position. Instead, the control system is expected to track the actual position by setting target=actual.

During the enabling phase, the actual position may be changing due to the axis phasing procedure until the drive sets *bit1* and *bit2* (*Status Word* = 0x127) to indicate, it is enabled. Now the control system sets the command bit 0x8 as it wants to control the target position itself. At the same moment, it should stop tracking the actual position. Now the drive sets bit 0x1000 and resets bit 0x100 to indicate, it is ready to follow the target positions. If the control system continues to track the actual position at this stage, this will cause a chasing behavior.

7 Register Access

Triamec drive registers are accessible through *EtherCAT* in two ways. Either cyclically, using *CoE Process Data Objects (PDO)*, or acyclic using *EtherCAT SDO Mailbox* transfer.

Finding register addresses is described in chapter 7.1.

How to use acyclic data exchange is explained in chapter 7.3.

How to setup cyclic data exchange is explained in chapter 7.4.

7.1 Find the CoE Address

First find the CoE address of the register that should be exchanged with *TwinCAT*. The *CoE* address is called the *URI Address* in the *Triamec* domain.

Open the *TAM System Explorer* and select the register in the tree view. Use the tab **General** and find the entry in **Identification > URI Address** (Figure 10).

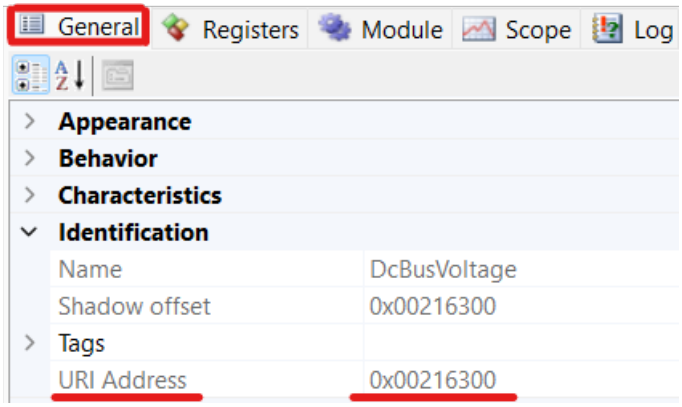


Figure 10: URI Address of registers

Obtain the URI-Address of the second axis of a drive from the first axis by adding 0x80000.

Arrays

The concept for arrays is identical. Each array field has its absolute *URI-Address*. The address of each field increments together with its index in the array +1.

- Application.Variables.Floats[0] URI-Address 0x00238601
- Application.Variables.Floats[1] URI-Address 0x00238602
- Application.Variables.Doubles[0] URI-Address 0x00238501
- Application.Variables.Doubles[1] URI-Address 0x00238503

Note Most drive *Registers* are 32Bit in size. For arrays with 64Bit values, the index increases by 2 (see examples above) and the encoding is LiddleEndian.

7.2 Committing Parameters

Changing *Registers* in a Commands tree takes immediate action. Whereas Registers in the Parameters tree require a commit to apply new values. This allows changing a set of *Parameters* and apply them all together.

There are different *Commit Groups*, where all have its dedicated commit command.

Commit Group	Command Register
General	General.Commands.CommitParameter
PathPlanner	Axes[].Commands.PathPlanner.CommitParameter
PositionController	Axes[].Commands.PositionController.CommitParameter
PathPlanner	Axes[].Commands.CurrentController.CommitParameter

To find the matching commit group for a Register, use the *TAM System Explorer*:

1. Find the target *Parameter* by navigating the *Register* tree and select it.
2. In the tab view, change to the **General** tab.
3. Find the value under **Identification > Tags > commitGroup**

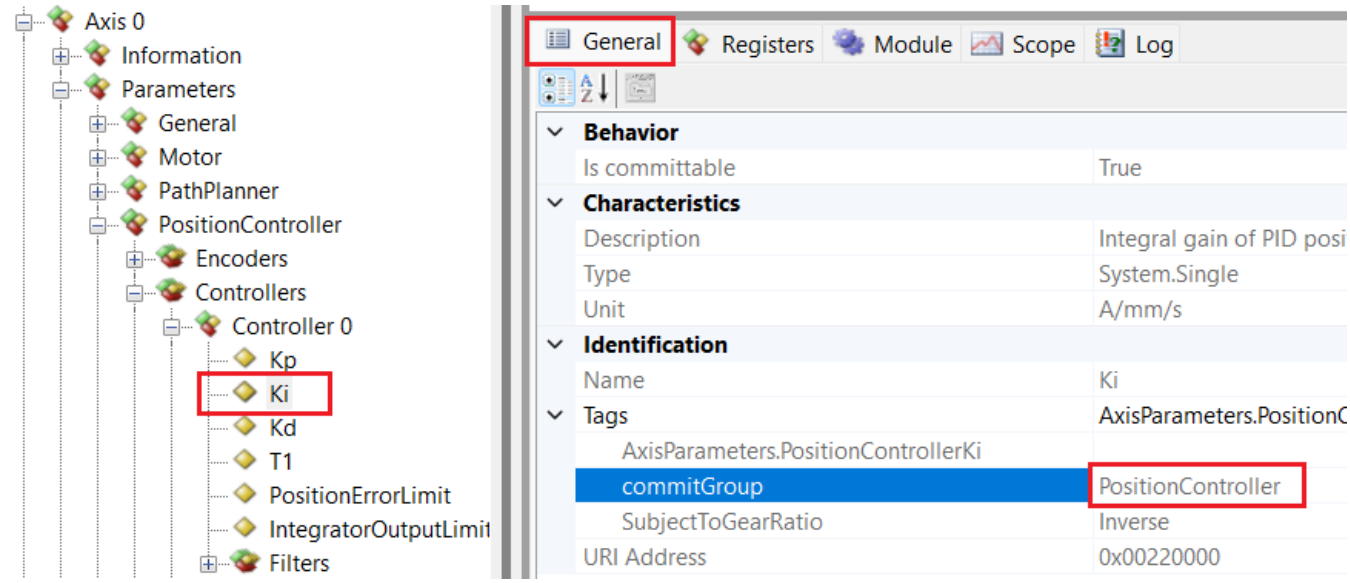


Figure 11: Find the Commit Group of a Parameter

All commit commands are Boolean, and reverted to False automatically, when the commit has finished.

Use the *TE_RegisterWriteInt32 FB* from the Library with the value 1, to set a commit command to true.

Warning	Do not change parameters before the last commit finished.
Warning	Write access of commitable parameters is not allowed for cyclic access.

7.3 Acyclic with SDO Mailbox

The Triamec library is a wrapper of the CoE access functions from the library *Tc2_EtherCAT*. It is recommended to use the Triamec Function Blocks for inherent data type consistency.

Declaration:

```
axis                    : use TE_AxisDirect or TE_AxisNci (or TE_AxisCnc of the sample code);
readReg                 : TE_RegisterReadInt32;
writeReg                : TE_RegisterWriteInt32;
```

Configuration:

```
axis.config.driveId     := 1004;                // see adapter netId
axis.config.netId       := '192.168.10.99.2.1'; // see drive EtherCAT Addr
```

Code snippet to read an integer register:

```
State0:
   readReg(axis:=axis, execute := FALSE);
State1:
   readReg.address   := 16#238701;
   readReg(axis:=axis, execute := TRUE);
```

```
   IF readReg.done THEN
      // result is available in readReg.value
      readReg(axis:=axis, execute := FALSE);
   END_IF
```

Code snippet to write an integer register:

```
state0:
   write(axis:=axis, execute := FALSE);
State1:
   write.value      := 137;
   write.address := 16#238701;
   write(axis:=axis, execute := TRUE);
   IF write.done THEN
      write(axis:=axis, execute := FALSE);
   END_IF
```

7.4 Cyclic with PDO Extensions

The axis *PDO* mapping corresponds to the standard drive profile *CiA402*. This only contains standardized axis data. With the slot mechanism, *EtherCAT* offers the option of flexible PDO mapping. This means that modules can be added or omitted depending on the application. The only exception is the *"Main Control"* module, which must be present.

This chapter describes how to set up additional cyclic data via *PDO Extensions*. Four *Extensions* are available for exchanging Drive registers configured by the customer.

The following example shows how to use Extension 0.

- Open **I/O > EtherCAT Master > Drive > Slots** (see Figure 12).
- Select In/Out of the Extension 0 on the left side.
- Select one choice from the right side. Note the Modules are available in different data formats.
- Use the button < to add the choice to an *Extension Slot*.

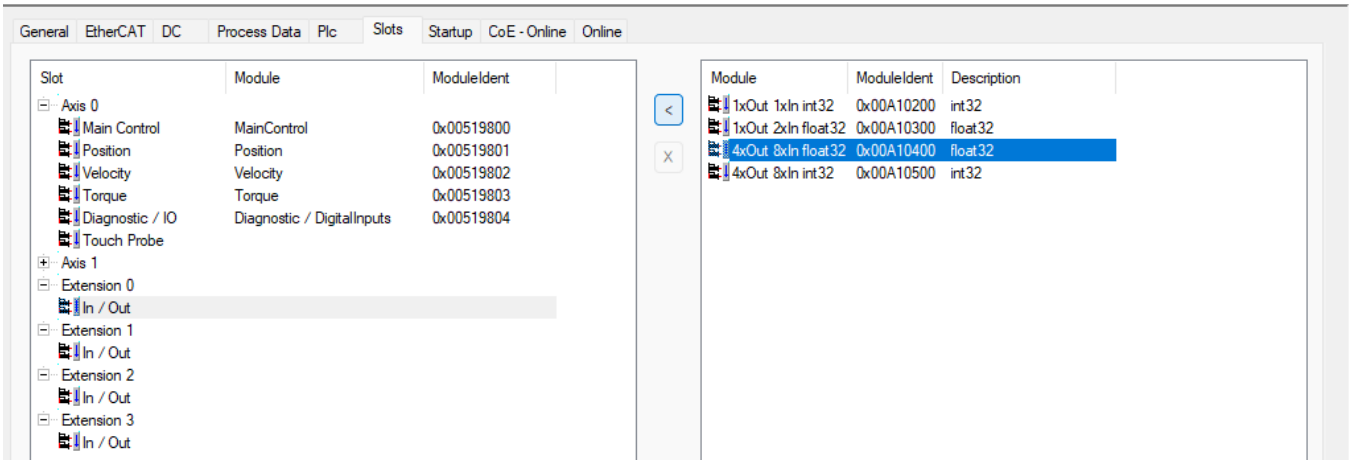


Figure 12: Flexible slot configuration in TwinCAT

- New variables will appear as an additional *Module* of the cyclic interface of the *Drive*.

Next, the drive configuration for the publishers and subscribers must be completed as described in the following chapters.

Publish Registers

Values sent from Drive to *TwinCAT* are configured as *Publishers* on the Drive side as follows.

4. Find the URI Address of the Drive Register to be published (see chapter 7.1).
5. Insert the URI Address into General.Parameters.EtherCAT.Publishers[*N*].Uri*X*,
   Where *UriX* is one of the available Registers Uri1 to Uri8,
   and *N* is the index of the selected *Publisher*.

The *Publisher* index is linked to the configuration made on the *TwinCAT* side. For convenience, choose the smallest available index first. Published registers will be written to the **Module > In** variables. This corresponds to *EtherCAT TxPDOs*.

Subscribe Registers

Values sent from *TwinCAT* to Drive are configured as *Subscribers* on the Drive side.

6. Find the URI Address of the Drive Register to where the incoming value should be passed to (see chapter 7.1).
7. Insert the URI Address into General.Parameters.EtherCAT.Subscribers[*N*].Uri*X*,
   Where *UriX* is one of the available Registers Uri1 to Uri4,
   and *N* is the index of the selected *Subscriber*.

The *Subscriber* index is linked to the configuration made on the *TwinCAT* side. For convenience, choose

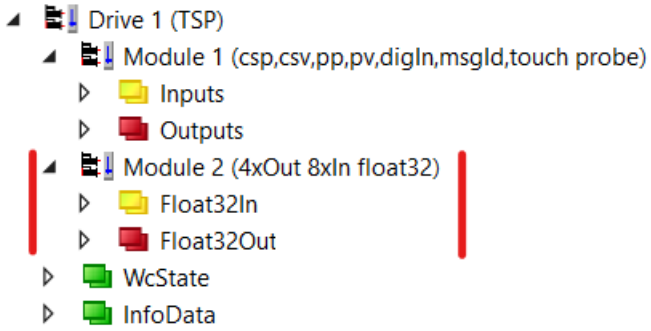


Figure 13: Extension Module added in Solution Explorer

the smallest available index first. Subscribed registers will be sourced from the **Module > Out** variables. This corresponds to *EtherCAT RxPDOs*.

After activating the TwinCAT project, the cyclic data will be exchanged according to the configuration.

8 Explicit Identification (Optional)

Triamec drives support the *EtherCAT* "Explicit Device Identification" with firmware ≥ 4.4.0. The feature requires the *ESI* file `Triamec1.4.xml` or newer and is only recommended if swap prevention is required by design.

First set an address in the drive register General.Parameters.LinkAddress and make the parameter persistent (see also chapter 1).

Setup the address as **Identification Value** in the **EtherCAT** section of the drive (Figure 14). If this value is not changeable, the *ESI* file version might be smaller than 1.4. Then open the **Advanced Settings** dialog and activate **General > Behavior > Check Identification**.

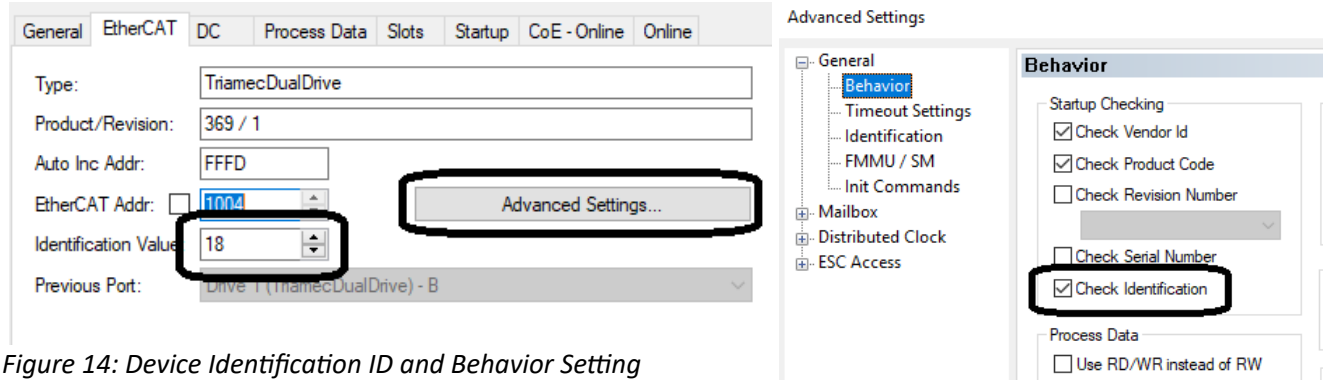


Figure 14: Device Identification ID and Behavior Setting

Then select **Explicit Device Identification** in the same dialog under **General > Identification** (Figure 15).

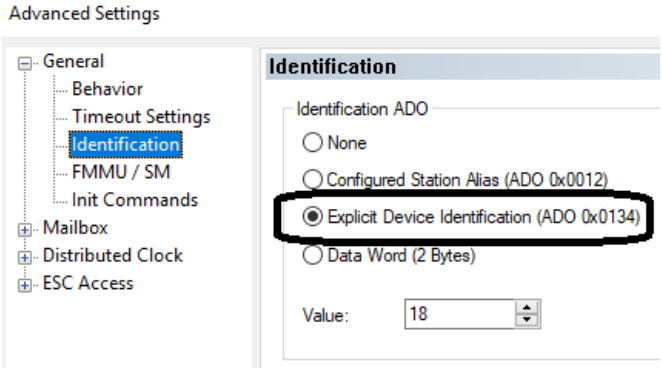


Figure 15: Explicit Device Identification Setting

9 Diagnostics

9.1 Cyclic MessageId

Two registers can be used to analyse error situations.

- 0x22A7 MessageId Axes[].Signals.General.MessageId (16 bit)
- 0x22A6 Message Axes[].Signals.General.Message (string)

The register *MessageId* is only available, if the firmware is newer than 4.16 and the EtherCAT cyclic interface "TSD_minFW4.16/TSP_minFW4.16" has been selected. Document [5] contains all error codes with a description and possible solutions. Values smaller than 6144 denote warnings.

The string representation "Message" shows the message in English including any parameters that might be interesting. This is not available cyclically.

The Triamec library that is handed out with the sample code contains functions to load the detailed message string whenever the *MessageId* changes. The mode is selected using ***axis.config.eventMode***:

- Disabled Do not load the string
- InternalOnly Load the string but do not throw any TwinCAT events
- InternationalWithoutDetails Load the *message* string and
  Throw events not based on this string, but based on language XML files.
  These are visible with the HMI or the TwinCAT event bar at ([2]).
- EnglishWithDetails Load the *Message* string and throw events showing this string (English only).
  This is visible with the HMI or with the window "TwinCAT logged events"

Please note that *messageId* is 16 bits where the older item *errorCombined* was 8 bits. When upgrading the library from 1.1.0 to 1.2.x without upgrading the *EtherCAT* interface, the connection in the IO manager must be reestablished ignoring the size. See Figure 16.

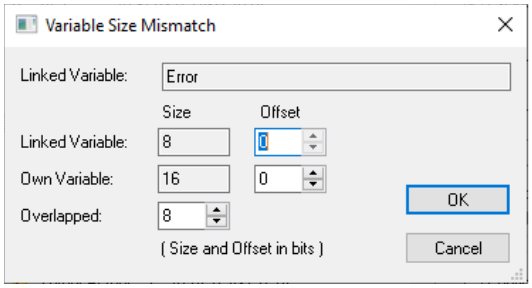


Figure 16: Assign variable with size mismatch

9.2 Non-Standard Registers

The hardware revision register 0x1018:03 reserves 8 bits for the minor revision. The CAN standard specifies 16 bits (EN 50325).

The register Torque (0x6077) is in absolute Units of mA. CAN does not specify the Unit, but specifies that the value should be relative to NominalTorque.

The drives support the automatic homing modes with modeOfOperation 6 (HM). This mode is activated according to the standard. However, in contrast to the standard, the homing method is specified using the register Axes[].Parameters.Homing.Type and all parameters in this folder. The CoE registers 0x6098

2 C:\TwinCAT\3.1\Components\TcEventLogger\TcEventBar.exe.

and 0x60E3 are not supported. See [3] for a description of the homing method.

9.3 Trouble Shooter

On the *TwinCAT* side, check:

- The correct *ESI* file is present at: `C:\TwinCAT\3.1\Config\Io\EtherCAT\`.
- The *ESI* file is loaded with: **TwinCAT > EtherCAT Devices > Reload Device Descriptions**.
- Make sure the LLDP Protocol is disabled in the *EtherCAT* network adapter settings.
- Under **I/O > Devices > EtherCAT master > Drive** check the configuration of **Slots**, **Startup**, **NC-A**, etc.
- Check that **I/O > Devices > EtherCAT master > Drive > DC > Operation Mode** is set to **DC-Synchron**.
- **All drives are assigned to the correct sync unit, check Adapter/EtherCAT/SyncUnitAssignment**
  - *NCI*: All assigned to **NC-Task SAF.**
  - *CNC*: All assigned to **CNC-Task GEO.**

On the *Drive* side, check:

- General.Signals.EtherCAT.LinkPll.State should be DcMaster or DcSlave
- Axes[].Parameters.PositionController.PositionUnit should be specified
- Axes[].Commands.PathPlanner.StreamRate
- Axes[].Signals.General.EtherCAT

If *EtherCAT* does not control an axis as expected, check the following

- The *EtherCAT* state of the drive must be OP (Operational).
- The Triamec command register Axes[].Commands.General.OverwriteControlSystem must be zero, which is its default value. A value 1 is used to control the drive using the TAM System Explorer and disables *EtherCAT* command requests.
- The drive parameter General.Parameters.Standalone must be False. The value True allows using the drive without a Link and disables *EtherCAT* command requests and *EtherCAT* synchronization. Also, it suppresses the error *LinkNotReady*.
- Make sure, Axes[0].Parameters.PositionController.MasterPositionSource points to the correct encoder and Axes[0].Parameters.PositionController.PositionUnit is correctly set.
- Check the following drive signals visible at the *EtherCAT* interface and in the *TAM System Explorer* register:
  - Axes[].Signals.General.EtherCAT.ErrorCombined
  - Axes[].Signals.General.StateMerged
  - Axes[].Signals.General.EtherCAT.ControlWord

10 References

[1] “Servo Drive Setup Guide, TSD and TSP Series”, ServoDrive-SetupGuide_EP019.pdf, Triamec Motion AG, 2023.

[2] “Cyclic Telegrams with EtherCAT”, AN104_EtherCAT-CyclicTelegrams_EP003.pdf, Triamec Motion AG, 2023.

[3] “Homing Procedures and Setup”, AN141_HomingProceduresAndSetup_EP003.pdf, Triamec Motion AG, 2023.

[4] “Twincat Library: Accessing Drive Registers”, AN109_TwinCAT-AccessingDriveRegisters_EP006.pdf, Triamec Motion AG, 2023.

[5] “Drive Messages”, AN102_DriveMessages_EP004.pdf, Triamec Motion AG, 2023.

[6] “TwinCAT Integration for Drives with EtherCAT Interface”, https://www.triamec.com/en/ethercat.html, Triamec Motion AG, 2023.

[7] “TwinCAT Solution Examples”on *GitHub* https://github.com/Triamec

[8] “Functional description: TF5200 TwinCAT 3 CNC Spindle”, https://download.beckhoff.com/download/document/automation/twincat3/TF5200_spindle_en.pdf

Revision History

Version	Date	Editor	Comment
001	2016-11-16	mvx	First Release
002	2017-06-16	mvx	Release with Homing
003	2017-11-20	mvx	Reference to AN109, add status and control word content
004	2018-11-06	dg	Chapter Modulo enhanced.
005	2019-02-12	mvx	Explicit Device Identification
006	2019-04-18	mvx	Velocity modes of operation
007	2019-09-23	dg	Synchronized with TSD User Guide EP001
008	2020-05-14	mvx	Added CNC Spindel-Settings (not published yet) and Triamec-Parameter in CNC
009	2020-11-03	dg	Error codes updated
010	2021-09-01	mvx	New error and message concept (FW > 4.13.6)
011	2022-08-17	mvx, sm	Update template, update to FW 4.16 instructions, general update
012	2022-09-12	sm	Update modulo and encoder mask description, fix doc version index
013	2023-01-27	sm	General update of content.
014	2023-04-19	sm, rb	Merge AN104 and AN109 Register Access into this guide, minor fixes
015	2023-11-02	rb	Velocity mode additions
016	2024-04-09	rb	Cyclic synchronous torque mode (CST), EtherCAT interface
017	2024-05-27	rb	Description of feed rate and encoder scaling improved
018	2024-06-10	rb	ESI, flexible PDO mapping (FW >= 4.22), GitHub examples

Triamec Motion AG
Lindenstrasse 16
6340 Baar / Switzerland

Phone +41 41 747 4040
Email info@triamec.com
Web www.triamec.com

Disclaimer