IDX

Application Notes

idx.maxongroup.com

IDX Industrial Drives
Application Notes
CCMC | Edition 2019-11 | DocID rel9103

TABLE OF CONTENTS

READ THIS FIRST

THESE INSTRUCTIONS ARE INTENDED FOR QUALIFIED TECHNICAL PERSONNEL. PRIOR COMMENCING WITH ANY ACTIVITIES…

- you must carefully read and understand this manual and
- you must follow the instructions given therein.

IDX DRIVES ARE CONSIDERED AS PARTLY COMPLETED MACHINERY ACCORDING TO EU DIRECTIVE 2006/42/EC, ARTICLE 2, CLAUSE (G) AND ARE INTENDED TO BE INCORPORATED INTO OR ASSEMBLED WITH OTHER MACHINERY OR OTHER PARTLY COMPLETED MACHINERY OR EQUIPMENT.
THEREFORE, YOU MUST NOT PUT THE DEVICE INTO SERVICE,…

- unless you have made completely sure that the other machinery fully complies with the EU directive’s requirements!
- unless the other machinery fulfills all relevant health and safety aspects!
- unless all respective interfaces have been established and fulfill the herein stated requirements!

••page intentionally left blank••

1 ABOUT

1.1 About this Document

1.1.1 Intended Purpose

The purpose of the present document is to provide you specific information to cover particular cases or scenarios that might come in handy during commissioning of your drive system.
Use for other and/or additional purposes is not permitted. maxon, the manufacturer of the equipment described, does not assume any liability for loss or damage that may arise from any other and/or additional use than the intended purpose.

The present document is part of a documentation set. The below overview shows the documentation hierarchy and the interrelationship of its individual parts:

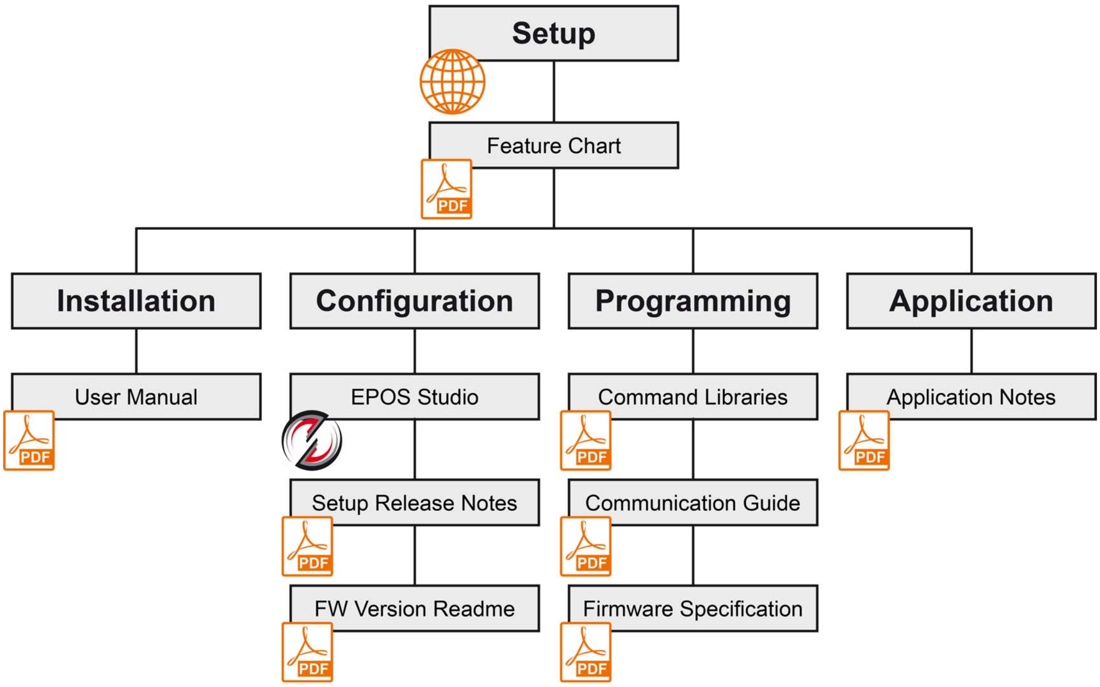


Figure 1-1 Documentation structure

1.1.2 Target Audience

This document is meant for trained and skilled personnel working with the equipment described. It conveys information on how to understand and fulfill the respective work and duties.

This document is a reference book. It does require particular knowledge and expertise specific to the equipment described.

1.1.3 How to use

Take note of the following notations and codes which will be used throughout the document.

Notation	Explanation
«Abcd»	indicating a title or a name (such as of document, product, mode, etc.)
¤Abcd¤	indicating an action to be performed using a software control element (such as folder, menu, drop-down menu, button, check box, etc.) or a hardware element (such as switch, DIP switch, etc.)
(n)	referring to an item (such as order number, list item, etc.)
*	referring to an internal value
***	referring to a not yet implemented item
➔	denotes “see”, “see also”, “take note of”, or “go to”

Table 1-1 Notations used

In the later course of the present document, the following abbreviations and acronyms will be used:

Short form	Meaning
EPOS4	hardware and/or firmware functionalities based on maxon EPOS4 platform
IDX	any type of IDX drive
IDX 56	any type of IDX 56 drive
IDX 56 CANopen	IDX drive with positioning controller and CANopen interface
IDX 56 EtherCAT	IDX drive with positioning controller and EtherCAT interface
IDX 56 I/O	IDX drive with speed controller and I/O interface

Table 1-2 Abbreviations and acronyms used

1.1.4 Symbols and Signs

Requirement / Note / Remark
Indicates an action you must perform prior continuing or refers to information on a particular item.

Best Practice
Gives advice on the easiest and best way to proceed.

Material Damage
Points out information particular to potential damage of equipment.

1.1.5 Trademarks and Brand Names

For easier legibility, registered brand names are listed below and will not be further tagged with their respective trademark. It must be understood that the brands (the below list is not necessarily concluding) are protected by copyright and/or other intellectual property rights even if their legal trademarks are omitted in the later course of this document.

Brand name	Trademark owner
Adobe® Reader®	© Adobe Systems Incorporated, USA-San Jose, CA
APOSS®	© zub machine control AG, CH-Rothenburg
CANopen® CiA®	© CiA CAN in Automation e.V, DE-Nuremberg
EtherCAT®	© EtherCAT Technology Group, DE-Nuremberg, licensed by Beckhoff Automation GmbH, DE-Verl
TwinCAT®	© Beckhoff Automation GmbH, DE-Verl
Windows®	© Microsoft Corporation, USA-Redmond, WA

Table 1-3 Brand names and trademark owners

1.1.6 Sources for additional Information

For further details and additional information, please refer to below listed sources:

#	Reference
[1]	IEC/EN 60204-1: Safety of machinery – Electrical equipment of machines
[2]	IEC/EN 61800-5-2: Adjustable speed electrical power drive systems

Table 1-4 Sources for additional information

1.1.7 Copyright

CCMC | IDX Application Notes | Edition 2019-11 | DocID rel9103

maxon motor ag
Brünigstrasse 220 +41 41 666 15 00
CH-6072 Sachseln www.maxongroup.com

1.2 About the Devices

maxon's «IDX» are compact, high-performance, IP 65-protected, brushless DC drives with either integrated positioning controller or speed controller particularly suitable for the use in harsh environmental conditions. They deliver up to 0.7 Nm of continuous torque and come in a wide range of configurable options that allow full adaption to suit specific needs.

The IDX firmware is based on maxon's EPOS4 platform. Therefore, many of the contents covered in this application note are identical or similar to EPOS4.

1.3 About the Safety Precautions

IMPORTANT NOTICE: PREREQUISITES FOR PERMISSION TO COMMENCE INSTALLATION

IDX drives are considered as partly completed machinery according to EU Directive 2006/42/EC, Article 2, Clause (g) and **are intended to be incorporated into or assembled with other machinery or other partly completed machinery or equipment**.

WARNING

Risk of Injury
Operating the device without the full compliance of the surrounding system with the EU directive 2006/42/EC may cause serious injuries!

- *Do not operate the device, unless you have made sure that the other machinery fulfills the requirements stated in EU directive!*
- *Do not operate the device, unless the surrounding system fulfills all relevant health and safety aspects!*
- *Do not operate the device, unless all respective interfaces have been established and fulfill the stated requirements!*

SAFETY FIRST!

Keep in mind: Safety first! Always!

- Do not engage with any work unless you possess the stated skills (➔chapter “1.1.2 Target Audience” on page 1-5)!
- Refer to ➔chapter “1.1.4 Symbols and Signs” on page 1-6 to understand the subsequently used indicators!
- You must observe any regulation applicable in the country and/or at the site of implementation with regard to health and safety/accident prevention and/or environmental protection!

DANGER

High voltage and/or electrical shock
Touching live wires causes death or serious injuries!

- *Consider any power cable as connected to live power, unless having proven the opposite!*
- *Make sure that neither end of cable is connected to live power!*
- *Make sure that power source cannot be engaged while work is in process!*
- *Obey lock-out/tag-out procedures!*
- *Make sure to securely lock any power engaging equipment against unintentional engagement and tag it with your name!*

Requirements

- *Make sure that all associated devices and components are installed according to local regulations.*
- *Be aware that, by principle, an electronic apparatus cannot be considered fail-safe. Therefore, you must make sure that any machine/apparatus has been fitted with independent monitoring and safety equipment. If the machine/apparatus should break down, if it is operated incorrectly, if the control unit breaks down or if the cables break or get disconnected, etc., the complete drive system must return – and be kept – in a safe operating mode.*
- *Be aware that you are not entitled to perform any repair on components supplied by maxon.*

Electrostatic sensitive device (ESD)

- *Wear working cloth and use equipment in compliance with ESD protective measures.*
- *Handle devices with extra care.*

••page intentionally left blank••

2 CONTROLLER ARCHITECTURE

CONTENT

2.1 In Brief

A wide variety of operating modes permit flexible configuration of drive and automation systems by using positioning, speed and current regulation. The built-in interface allows online commanding by CAN or EtherCAT bus master units as well as networking to multiple axes drives.

Good quality velocity PI control is made possible by the use of algorithms for estimating the motor rotation velocity from the measured rotor position that are based either on a low pass filter or on a velocity observer.

OBJECTIVE

The present application note explains the EPOS4/IDX controller architecture.

In addition to PID position regulation, the functionalities of the built-in acceleration and velocity feedforward are described.

The functionality of the velocity PI controller, the low pass filter, and the observer used for estimating the velocity are described. The benefits of each velocity estimation method are highlighted and illustrated by using practical examples.

SCOPE

Hardware	Order #	Firmware version	Reference
EPOS4/IDX	–	0160h	IDX Firmware Specification
IDX 56	various	0160h or higher	

Table 2-5 Controller architecture | Covered hardware and required documents

TOOLS

Tools	Description
Software	«EPOS Studio» Version 3.6 or higher

Table 2-6 Controller architecture | Recommended tools

2.2 Overview

The EPOS4/IDX controller architecture contains three built-in control loops.

- Current regulation is used in all modes.
- Position or velocity regulation is only used in position-based or velocity-based modes, respectively.

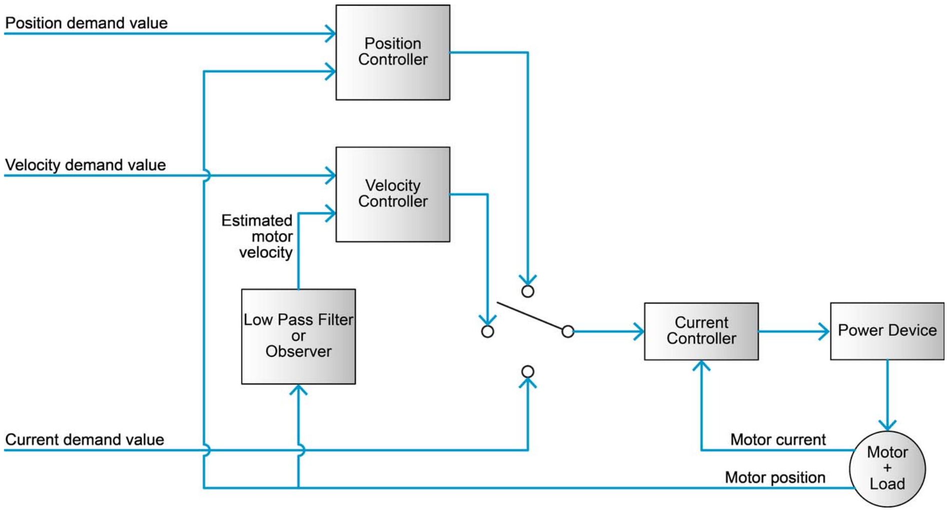


Figure 2-2 Controller architecture | Overview

2.3 Regulation Methods

2.3.1 Current Regulation

During a movement within a drive system, forces and/or torques must be controlled. Therefore, as a principal regulation structure, EPOS4/IDX offers current-based control.

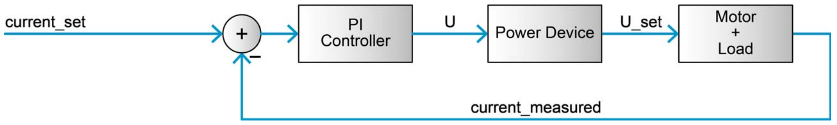


Figure 2-3 Controller architecture | Current regulator

CONSTANTS

Sampling period: $T_s = 0.04ms$

OBJECT DICTIONARY ENTRIES

Symbol	Unit	Name	Index	Subindex
K_{P_EPOS4}	$\frac{mV}{A}$	Current controller P gain	0x30A0	0x01
K_{I_EPOS4}	$\frac{mV}{A \cdot ms}$	Current controller I gain	0x30A0	0x02

Table 2-7 Controller architecture | Current regulation – Object dictionary

CONVERSION OF PI CONTROLLER PARAMETERS (EPOS4/IDX TO SI UNITS)

$$K_{P\_SI} = 0.001 \cdot K_{P\_EPOS4}$$

$$K_{I\_SI} = 0.001 \cdot K_{I\_EPOS4}$$

Current controller parameters in SI units can be used in analytical or numerical simulations via the following transfer function:

$$C_{current}(s) = K_{P\_SI} + \frac{K_{I\_SI}}{s}$$

ANTI-WINDUP

In order to prevent degradation of the control performance when the control input stays at the limit value for long time, an anti-windup algorithm is implemented preventing the integral part of the PI controller to take values larger than the ones bound on the control input.

TRANSPORT DELAY OF THE CONTROL LOOP

Total transport delay of the current regulation loop is always smaller than 0.06 ms.

2.3.2 Velocity Regulation (with Feedforward)

EPOS4/IDX offers velocity regulation based on the subordinated current control.

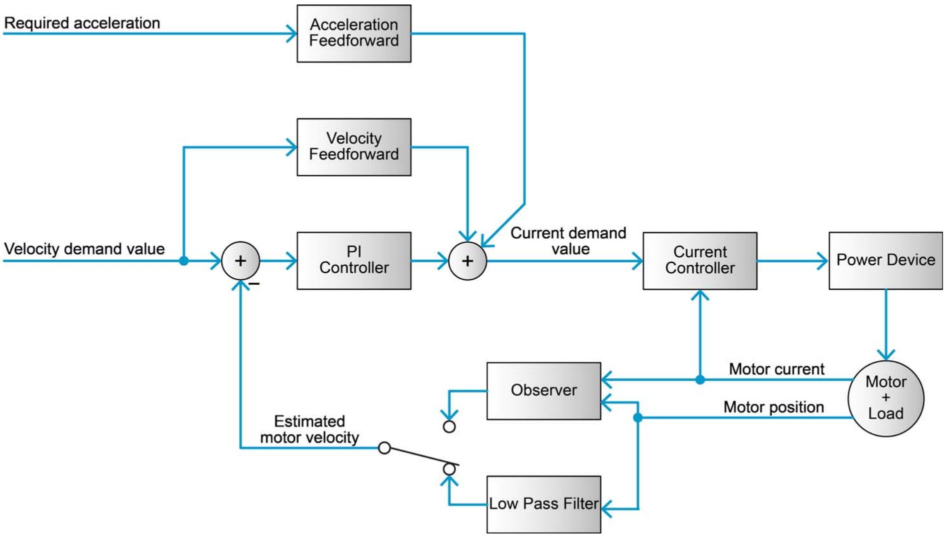


Figure 2-4 Controller architecture | Velocity regulator with feedforward

CONSTANTS

Sampling period: $T_s = 0.4ms$

OBJECT DICTIONARY ENTRIES FOR CONTROLLER

Symbol	Unit	Name	Index	Subindex
$K_{P\omega\_EPOS4}$	$\frac{mA \cdot s}{rad}$	Velocity controller P gain	0x30A2	0x01
$K_{I\omega\_EPOS4}$	$\frac{mA}{rad}$	Velocity controller I gain	0x30A2	0x02
FF_{ω_EPOS4}	$\frac{mA \cdot s}{rad}$	Velocity controller FF velocity gain	0x30A2	0x03
FF_{α_EPOS4}	$\frac{mA \cdot s^2}{rad}$	Velocity controller FF acceleration gain	0x30A2	0x04

Table 2-8 Controller architecture | Velocity regulation – Object dictionary

CONVERSION OF PI CONTROLLER PARAMETERS (EPOS4/IDX TO SI UNITS)

$$K_{P\omega\_SI} = 0.001 \cdot K_{P\omega\_EPOS4}$$

$$K_{I\omega\_SI} = 0.001 \cdot K_{I\omega\_EPOS4}$$

$$FF_{\omega\_SI} = 0.001 \cdot FF_{\omega\_EPOS4}$$

$$FF_{\alpha\_SI} = 0.001 \cdot FF_{\alpha\_EPOS4}$$

Velocity controller parameters in SI units can be used in analytical or numerical simulations via transfer function for the PI controller:

$$C_{velocity}(s) = K_{P\omega\_SI} + \frac{K_{I\omega\_SI}}{s}$$

ANTI-WINDUP

An anti-windup algorithm is implemented to prevent integration wind-up in PI controller, when the actuators are saturated.

LOW PASS FILTER

The estimation of the motor velocity can be done by using the measured time between consecutive sensor edges, which is low pass filtered in order to eliminate the effects of measurement noise. The transfer function of the low pass filtered estimation functionality that can be used in simulations has the following form:

$$C_{FilterEstimator}(s) = \frac{1}{1 + \frac{K_{P\omega\_SI}}{48 \cdot K_{I\omega\_SI}} \cdot s}$$

OBSERVER

An alternative to the low pass filter is the use of an observer. Thereby, the observed velocity is calculated in two steps. First; prediction of the velocity, position, and external torque, based on the parameters that define the mechanical transfer function of the system. Second; correction of the predicted values based on the newly measured rotor position.

OBJECT DICTIONARY ENTRIES FOR OBSERVER

Symbol	Unit	Name	Index	Subindex
k_{m_EPOS4}	$\frac{mNm}{A}$	Torque constant	0x3001	0x05
l_{θ_EPOS4}	1	Velocity observer position correction gain	0x30A3	0x01
l_{ω_EPOS4}	Hz	Velocity observer velocity correction gain	0x30A3	0x02
l_{T_EPOS4}	$\frac{mNm}{rad}$	Velocity observer load correction gain	0x30A3	0x03
r_{EPOS4}	$\frac{\mu Nm}{rpm}$	Velocity observer friction	0x30A3	0x04
J_{EPOS4}	$g \cdot cm^2$	Velocity observer inertia	0x30A3	0x05

Table 2-9 Controller architecture | Velocity observer – Object dictionary

All parameters relevant for the observer operation can be entered either manually or can be obtained from the EPOS4/IDX auto tuning procedure. The auto tuning automatically executes the identification experiments, identifies the relevant parameters that characterize the drive train, and calculates the values of the observer correction gains.

CONVERSION OF OBSERVER PARAMETERS (EPOS4/IDX TO SI UNITS)

$$k_{m\_SI} = 0.001 \cdot k_{m\_EPOS4}$$

$$J_{SI} = 0.0000001 \cdot J_{EPOS4}$$

$$r_{SI} = \frac{0.00003}{\pi} \cdot r_{EPOS4}$$

$$l_{\theta\_SI} = l_{\theta\_EPOS4}$$

$$l_{\omega\_SI} = l_{\omega\_EPOS4}$$

$$l_{T\_SI} = 0.001 \cdot l_{T\_EPOS4}$$

The transfer functions characterizing the two steps in the observer calculations and that can be used in numerical simulation of the velocity controller with observer are the following:

PREDICTION STEP

$$\theta_{Observed} = \frac{\omega_{Observed}}{s}$$

$$\omega_{Observed} = \frac{k_{M\_SI} \cdot i_{Measured} - T_{Observed}}{J_{SI} \cdot s + r_{SI}}$$

CORRECTION STEP

$$\theta_{Observed} = \theta_{Observed} + l_{\theta\_SI} \cdot (\theta_{Measured} - \theta_{Observed})$$

$$\omega_{Observed} = \omega_{Observed} + l_{\omega\_SI} \cdot (\theta_{Measured} - \theta_{Observed})$$

$$T_{Observed} = T_{Observed} + l_{T\_SI} \cdot (\theta_{Measured} - \theta_{Observed})$$

WHEN SHOULD THE LOW PASS FILTER BE USED TO ESTIMATE THE VELOCITY?

The estimation of the motor velocity based on measuring the time between consecutive sensor edges and low pass filtering does not rely on any additional information on the mechanical system to which the motor is attached. Therefore, it is suitable in cases when there is no information on the mechanical properties of the system available or when the characteristics of the system change significantly over time.

Typical examples are cases in which the moment of inertia or viscous friction that the motor encounters change significantly during operation.

The solution with the filter gives good results in cases when a high-resolution position sensor is used and when the motor is operated at relatively high velocities (more than 20% of nominal motor speed). However, in cases when the resolution of the position sensor is low and/or the motor operates at low speed, the estimation with the observer results in a better control performance.

WHEN SHOULD THE OBSERVER BE USED TO ESTIMATE THE VELOCITY?

In order to use the observer for estimating the rotational velocity of the motor, parameters, such as inertia and viscous friction coefficient of the drive system, need to be known and must be stable over time and should not change a lot during operation. In EPOS4/IDX, there is an option to identify all the required parameters by using the «Auto Tuning Wizard».

The use of the observer brings most advantages when the position feedback sensor has a low resolution. Typical example is the use of Hall sensors for feedback instead of an incremental encoder, or the use of incremental encoders with up to 500 counts per turn. In general, the use of the observer provides a less noisy estimation of the rotor velocity resulting in better regulation and less audible noise especially at low operational velocities.

In addition, the velocity observer can be set stiffer (compared to the case when the filter is used) due to better quality of the estimated feedback signal resulting in a very good dynamical response.

However, when encoders with high resolution (above 500 counts per turn) are used, the performance of the system with observer is similar to its performance in the case when the low pass filter is used.

TRANSPORT DELAY OF THE CONTROL LOOP

Total transport delay of the velocity regulation loop is always smaller than 0.4 ms.

2.3.3 Position Regulation (with Feedforward)

EPOS4/IDX is able to close a positioning control loop based on the subordinated current control.

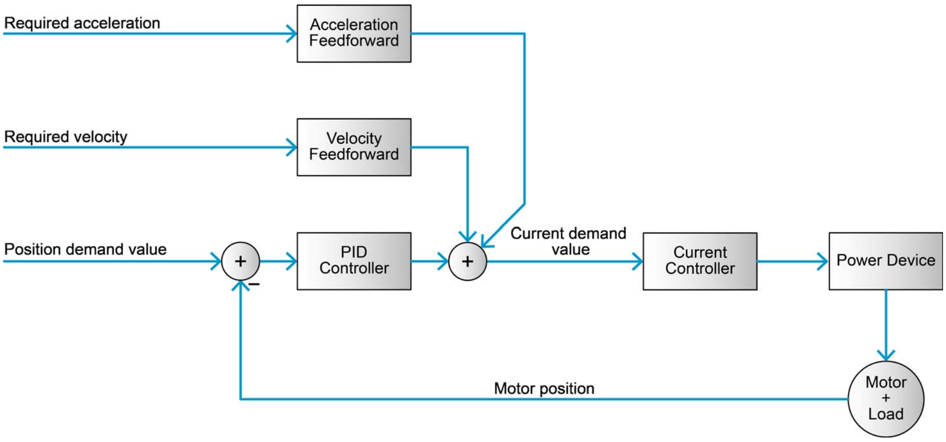


Figure 2-5 Controller architecture | Position regulator with feedforward

CONSTANTS

Sampling period: $T_s = 0.4\mathrm{msec}$

OBJECT DICTIONARY ENTRIES

Symbol	Unit	Name	Index	Subindex
K_{PP_EPOS4}	$\frac{mA}{rad}$	Position controller P gain	0x30A1	0x01
K_{IP_EPOS4}	$\frac{mA}{rad \cdot s}$	Position controller I gain	0x30A1	0x02
K_{DP_EPOS4}	$\frac{mA \cdot s}{rad}$	Position controller D gain	0x30A1	0x03
FF_{ω_EPOS4}	$\frac{mA \cdot s}{rad}$	Position controller FF velocity gain	0x30A1	0x04
FF_{α_EPOS4}	$\frac{mA \cdot s^2}{rad}$	Position controller FF acceleration gain	0x30A1	0x05

Table 2-10 Controller architecture | Position regulation – Object dictionary

The position controller is implemented as PID controller. To improve the motion system's setpoint fol- lowing, positioning regulation is supplemented by feedforward control. Thereby, velocity feedforward serves for compensation of speed-proportional friction, whereas acceleration feedforward considers known inertia. In addition, the differential part of the PID Controller signal is low pass filtered before it is added to the proportional and integral part. Low pass filtering is done to prevent negative influence on the control performance by the differentiation of noisy measured motor position.

CONVERSION OF PI CONTROLLER PARAMETERS (EPOS4/IDX TO SI UNITS)

$$K_{PP\_SI} = 0.001 \cdot K_{P\_EPOS4}$$

$$K_{IP\_SI} = 0.001 \cdot K_{I\_EPOS4}$$

$$K_{DP\_SI} = 0.001 \cdot K_{D\_EPOS4}$$

$$FF_{\omega\_SI} = 0.001 \cdot FF_{\omega\_EPOS4}$$

$$FF_{\alpha\_SI} = 0.001 \cdot FF_{\alpha\_EPOS4}$$

Position controller parameters in SI units can be used in analytical or numerical simulations via transfer function:

$$C_{position}(s) = K_{PP\_SI} + \frac{K_{IP\_SI}}{s} + \frac{K_{DP\_SI} \cdot s}{1 + \frac{K_{DP\_SI}}{10 \cdot K_{PP\_SI}} \cdot s}$$

ANTI-WINDUP

The anti-windup method is used to prevent integration wind-up in PID controller when the actuators are saturated.

2.3.4 Operation Modes with Feedforward

Acceleration and velocity feedforward are effective in «Profile Position Mode» (PPM), «Profile Velocity Mode» (PVM), and «Homing Mode» (HMM). All other operating modes are not affected.

PURPOSE OF VELOCITY FEEDFORWARD

Velocity feedforward provides additional current in cases, where the load increases with speed, such as speed-dependent friction. The load is assumed to proportionally increase with speed. The optimal velocity feedforward parameter in SI units is:

$$FF_{\omega\_SI} = \frac{r_{SI}}{k_{m\_SI}}$$

Meaning: With given total friction proportional factor in SI units r_{SI} relative to the motor shaft, and the motor's torque constant also in SI units k_{m_SI}, you ought to adjust the velocity feedforward parameter to:

$$FF_{\omega\_EPOS4} = 1000 \cdot FF_{\omega\_SI} = 1000 \cdot \frac{r_{SI}}{k_{m\_SI}}$$

PURPOSE OF ACCELERATION FEEDFORWARD

Acceleration feedforward provides additional current in cases of high acceleration and/or high load inertias. The optimal acceleration feedforward parameter in SI units is:

$$FF_{a\_SI} = \frac{J_{SI}}{k_{m\_SI}}$$

Meaning: With given total inertia in SI units J_{SI} relative to the motor shaft, and the motor's torque constant in SI units k_{m_SI}, you ought to adjust the acceleration feedforward parameter to:

$$FF_{\alpha\_EPOS4} = 1000 \cdot FF_{\alpha\_SI} = 1000 \cdot \frac{J_{SI}}{k_{m\_SI}}$$

TRANSPORT DELAY OF THE CONTROL LOOP

Total transport delay of the position regulation loop is always smaller than 0.4 ms.

2.4 Regulation Tuning

maxon's «EPOS Studio» features regulation tuning as a powerful wizard allowing to automatically tune all controller, estimator, and feedforward parameters described above for most drive systems within a few minutes.

••page intentionally left blank••

3 FIRMWARE UPDATE WITHOUT USE OF «EPOS STUDIO»

CONTENTS

3.1 In Brief

OBJECTIVE

The present application note explains how to carry out a firmware update of an EPOS4/IDX drive without the use of the «EPOS Studio» directly via the existing bus systems. The compatibility of the various versions as well as the necessary implementation sequences for the different communication interfaces are described.

SCOPE

Hardware	Order #	Firmware version	Reference
EPOS4/IDX	–	0160h	IDX Firmware Specification
IDX 56	various	0160h or higher	

Table 3-11 EtherCAT integration | Covered hardware and required documents

TOOLS

Tools	Description
Software	«EPOS Studio» Version 3.6 or higher (required for initial export of Program Data File, only)

Table 3-12 EtherCAT integration | Recommended tools

3.2 Preconditions

SUPPORTED INTERFACES

The IDX drive supports the following communication interfaces:

- USB
- CANopen
- EtherCAT

The firmware version 0x0160 or higher is required.

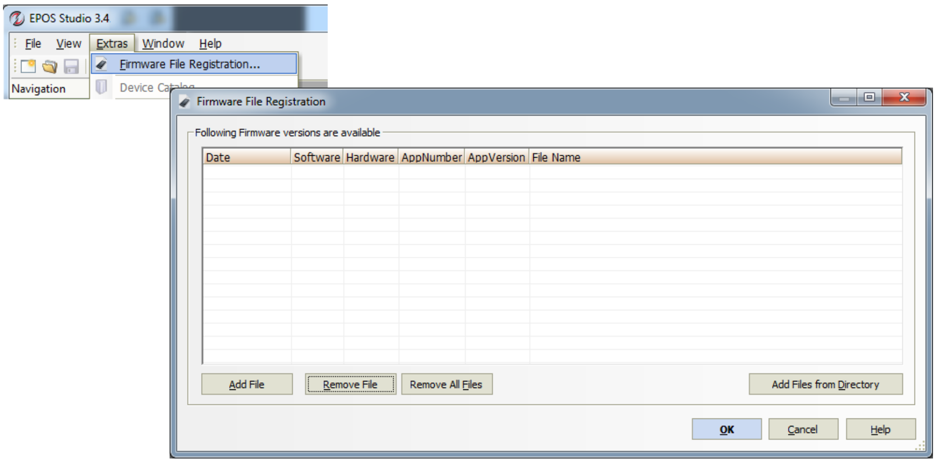
The firmware update sequence requires a «Program Data File» containing the desired firmware version. These files are exported using the «EPOS Studio».

STARTING «EPOS STUDIO»

1) Make sure you installed «EPOS Studio» version 3.6 (or higher) on your PC.
   If not the case, download the latest version here: ➔http://epos.maxongroup.com
2) Start «EPOS Studio» without creating a new project. Thereby, an online connection to an IDX drive is not necessary.

EXPORTING «PROGRAM DATA FILE»

1) In the main menu ¤Extras¤, open the dialog ¤Firmware File Registration¤.
2) Add a firmware file and click the ¤Add File¤ button.

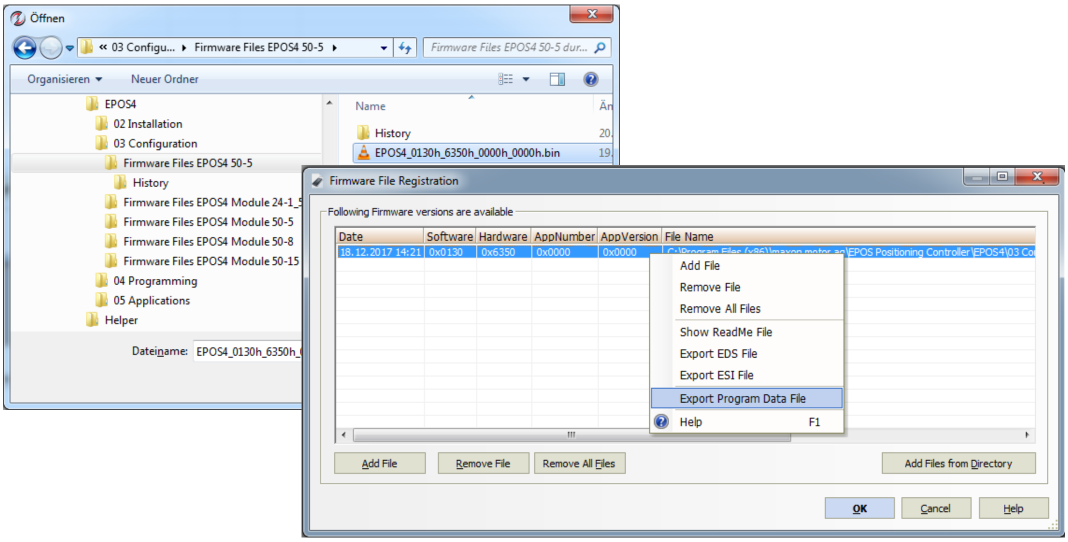


Following Firmware versions are available

Date	Software	Hardware	AppNumber	AppVersion	File Name

Figure 3-6 Firmware update without «EPOS Studio» | Open firmware file registration dialog

3) Select the firmware and click right to open the context menu, then click ¤Export Program Data File¤.


Öffnen
« 03 Configu... ▸ Firmware Files EPOS4 50-5 ▸
Firmware Files EPOS4 50-5 dur...
Organisieren ▾
Neuer Ordner
EPOS4
02 Installation
03 Configuration
Firmware Files EPOS4 50-5
History
Firmware Files EPOS4 Module 24-1_5
Firmware Files EPOS4 Module 50-5
Firmware Files EPOS4 Module 50-8
Firmware Files EPOS4 Module 50-15
04 Programming
05 Applications
Helper
Name
History
EPOS4_0130h_6350h_0000h_0000h.bin
Dateiname: EPOS4_0130h_6350h_(
Firmware File Registration
Following Firmware versions are available
Add File
Remove File
Remove All Files
Show ReadMe File
Export EDS File
Export ESI File
Export Program Data File
Help F1
Add File
Remove File
Remove All Files
Add Files from Directory
OK
Cancel
Help

Date	Software	Hardware	AppNumber	AppVersion	File Name
18.12.2017 14:21	0x0130	0x6350	0x0000	0x0000	C:\Program Files (x86)\maxon motor ag\EPOS Positioning Controller\EPOS4\03 Co

Figure 3-7 Firmware update without «EPOS Studio» | Export program data file

4) Select the directory to export file and click ¤OK¤.

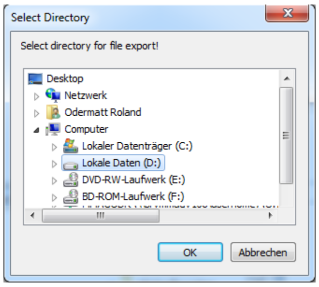
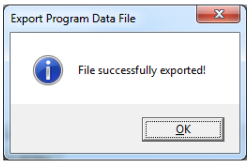


Figure 3-8 Firmware update without «EPOS Studio» | Select export directory

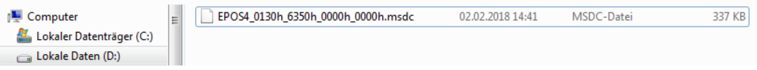
5) Click ¤OK¤ to confirm.



File successfully exported!

Figure 3-9 Firmware update without «EPOS Studio» | Confirm export directory

6) Check the exported firmware file (*.msdc).



EPOS4_0130h_6350h_0000h_0000h.msdc 02.02.2018 14:41 MSDC-Datei 337 KB

Figure 3-10 Firmware update without «EPOS Studio» | Check firmware file

3.3 Firmware Update via USB

SUPPORTED UPDATE PATHS

The following table shows the compatibility of a given firmware version for direct update via the USB interface.

OLD firmware version	NEW firmware version	
	0x0160	**higher**
0x0160		✓
higher	✓	

Table 3-13 Firmware update without «EPOS Studio» | USB – Old vs. new firmware version

SEQUENCE

Steps

a) Prepare controller (➔“Prepare Controller” on page 3-25)

b) Download «program data file» (CiA 302-3) “IDX_wwwwh_xxxxh_yyyyh_zzzzh.msdc”
(➔“Download «Program Data File» (CiA 302-3)” on page 3-26)

c) Check identity (➔“Check Identity” on page 3-27)

3.4 Firmware Update via CANopen

SUPPORTED UPDATE PATHS

The following table shows the compatibility of a given firmware version for direct update via the CANopen interface.

OLD firmware version	NEW firmware version	
	0x0160	**higher**
0x0160		✓
higher	✓	

Table 3-14 Firmware update without «EPOS Studio» | CANopen – Old vs. new firmware version

SEQUENCE

Steps

a) Prepare controller (➔“Prepare Controller” on page 3-25)

b) Download «program data file» (CiA 302-3) “IDX_wwwwh_xxxxh_yyyyh_zzzzh.msdc”
(➔“Download «Program Data File» (CiA 302-3)” on page 3-26)

c) Check identity (➔“Check Identity” on page 3-27)

3.5 Firmware Update via EtherCAT

SUPPORTED UPDATE PATHS

The following table shows the compatibility of a given firmware version for direct update via the EtherCAT interface.

OLD firmware version	NEW firmware version	
	0x0160	higher
0x0160		✓
higher	✓	

Table 3-15 Firmware update without «EPOS Studio» | EtherCAT – Old vs. new firmware version

SEQUENCE

Steps

a) Prepare controller (➔“Prepare Controller” on page 3-25)

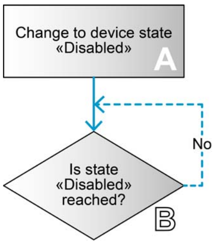
b) Download «program data file» (FoE) “IDX_wwwwh_xxxxh_yyyyh_zzzzh.msdc” (➔“Download «Program Data File» (FoE)” on page 3-27)

c) Check identity (➔“Check Identity” on page 3-27)

3.6 Steps: How to…

The following section describes the implementation of the required steps during the different firmware update sequences.

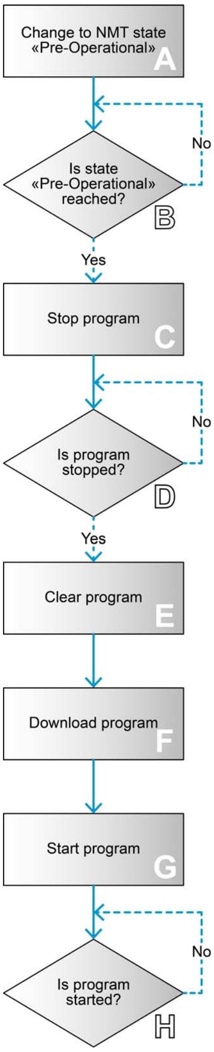
3.6.1 Prepare Controller


Change to device state «Disabled»
A
No
Is state «Disabled» reached?
B

#	Step	Description
A	Change to device control state «Disabled»	➔separate document «IDX Firmware Specification»; chapter “Device Control”
B	Check state	

Table 3-16 Firmware update without «EPOS Studio» | How to prepare the controller

3.6.2 Download «Program Data File» (CiA 302-3)



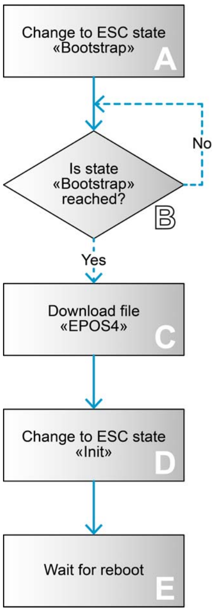
#	Step	Description	
A	Change to device control state «Pre-Operational» (1)	➔separate document «IDX Communication Guide»; chapter “CAN Communication”	
B	Check NMT state (1)		
C	Stop program (2)	Write «Stop» to object «Program Control»	
		Object Value Timeout	0x1F51-01 0x00 (Stop) 10 ms
D	Wait until program is stopped	Read value from object «Program Control»	
		Object Expected value Wait timeout	0x1F51-01 0x00 (Stopped) 10’000 ms
E	Clear program	Write «Clear» to object «Program Control»	
		Object Value Timeout	0x1F51-01 0x03 (Clear) 20’000 ms
F	Download program	Write file content to object «Program Data»	
		Object File Timeout	0x1F50-00 IDX_wwwwh_xxxxh_yyyyh_zzzzh.msdc 10’000 ms
G	Start program (2)	Write «Start» to object «Program Control»	
		Object Value Timeout	0x1F51-01 0x01 (start) 10 ms
H	Wait until program is started	Read value from object «Program Control»	
		Object Expected value Wait timeout	0x1F51-01 0x01 (Started) 10’000 ms

(1) only for CANopen interface

(2) During starting or stopping the program, the communication protocol is aborted. The controller does not respond to the received command. Reduce timeout and do not check communication result.

Table 3-17 Firmware update without «EPOS Studio» | How to download the program data file (CiA 302-3)

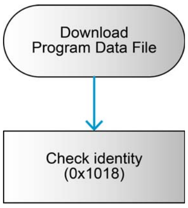
3.6.3 Download «Program Data File» (FoE)


Change to ESC state «Bootstrap»
A
No
Is state «Bootstrap» reached?
B
Yes
Download file «EPOS4»
C
Change to ESC state «Init»
D
Wait for reboot
E

#	Step	Description	
A	Change to ESC state «Bootstrap»		
B	Check ESC state		
C	Download file «IDX»	Write file content to file «IDX» using FoE protocol	
		Password	0
		Source file	IDX_wwwwh_xxxxh_yyyyh_zzzzh.msdc
		Target file	IDX
D	Change to ESC state «Init»		
E	Wait for reboot	Read ESC state	

Table 3-18 Firmware update without «EPOS Studio» | How to download the program data file (FoE)

3.6.4 Check Identity



Step	Description	
Check «Product Code»	Read value from object «Identity – Product code»	
	Object	0x1018-02
	Expected value	Hardware version and application number
Check «Revision number»	Read value from object «Identity – Revision number»	
	Object	0x1018-03
	Expected value	Software version and application version

Table 3-19 Firmware update without «EPOS Studio» | How to check identity

3.7 Object Dictionary

OBJECTS IN «STOPPED» STATE

While the program is stopped, only a few objects are accessible.

Index	Name
0x1000-00	Device type
0x1008-00	Manufacturer device name
0x1018-01	Identity – Vendor-ID
0x1018-02	Identity – Product code
0x1018-03	Identity – Revision number
0x1018-04	Identity – Serial number
0x1F50-00	Program data
0x1F51-00	Program control
0x1F56-00	Program software identification

Table 3-20 Firmware update without «EPOS Studio» \| Objects in «Stopped» state

OBJECTS VALUES IN «STOPPED» STATE

While the program is stopped, the displayed values of the following objects differ.

Index	Name	Program started Application active	Program stopped Bootloader active
0x1000-0x00	Device type	0x00020192	0x0000012E
0x1018-0x02	Product code	• High word: Hardware version • Low word: Application number	• High word: Hardware version • Low word: 0x0000
0x1018-0x03	Revision number	• High word: Software version • Low word: Application version	• High word: 0x0000 • Low word: 0x0000

Table 3-21 Firmware update without «EPOS Studio» \| Objects values in «Stopped» state

4 ETHERCAT INTEGRATION

CONTENTS

4.1 In Brief

OBJECTIVE

The present application note explains how to integrate the IDX into an EtherCAT Master Environment.

SCOPE

Hardware	Order #	Firmware version	Reference
EPOS4/IDX	–	0160h	IDX Firmware Specification
IDX 56	various	0160h or higher	

Table 4-22 EtherCAT integration | Covered hardware and required documents

TOOLS

Tools	Description
Software	«EPOS Studio» Version 3.6 or higher
	for zub’s MACS Multi-Axis EtherCAT Masters ➔“Required Tools” on page 4-45

Table 4-23 EtherCAT integration | Recommended tools

4.2 Beckhoff TwinCAT

4.2.1 Integrating ESI Files

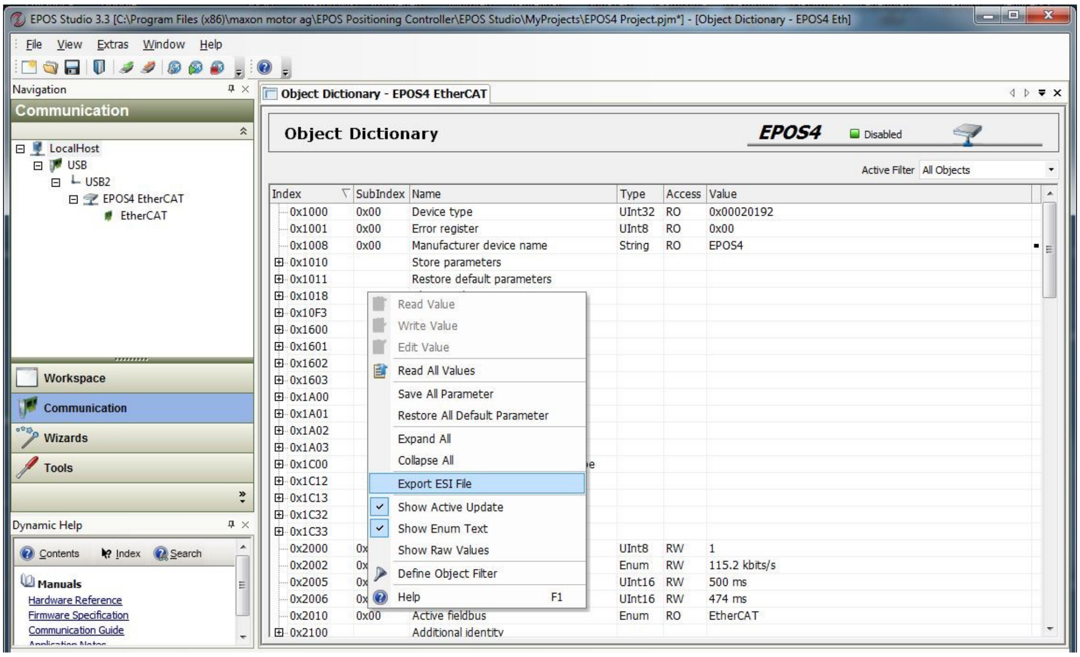
To integrate an IDX EtherCAT axis into the Beckhoff Master System, copy the ESI (EtherCAT Slave Information) XML file to the following folder. Note that the actual folder designation (***) depends on the TwinCAT version you are using:

- For **TwinCAT XAE** use path “C:\TwinCAT\***3.1\Config\Io\EtherCAT\”.
- For **TwinCAT2** use path “C:\TwinCAT\Io\EtherCAT\”.

4.2.2 How to export the ESI File

You can export the ESI file using the «EPOS Studio»:

1) Make sure that the EPOS4/IDX is connected and Online.
2) In the Object Dictionary, click right to open the context menu, then select ¤Export ESI File¤.



Index	SubIndex	Name	Type	Access	Value
0x1000	0x00	Device type	UInt32	RO	0x00020192
0x1001	0x00	Error register	UInt8	RO	0x00
0x1008	0x00	Manufacturer device name	String	RO	EPOS4
0x1010		Store parameters			
0x1011		Restore default parameters			
0x1018					
0x10F3					
0x1600					
0x1601					
0x1602					
0x1603					
0x1A00					
0x1A01					
0x1A02					
0x1A03					
0x1C00					
0x1C12					
0x1C13					
0x1C32					
0x1C33					
0x2000	0x		UInt8	RW	1
0x2002	0x		Enum	RW	115.2 kbits/s
0x2005	0x		UInt16	RW	500 ms
0x2006	0x		UInt16	RW	474 ms
0x2010	0x00	Active fieldbus	Enum	RO	EtherCAT
0x2100		Additional identity			

Figure 4-11 EtherCAT integration – Beckhoff TwinCAT | Export ESI file

4.2.3 Scanning the EtherCAT Slave Device

1) Connect the IDX to the EtherCAT Master and turn on power.
2) Open the Beckhoff System Manager and create a new project using menu ¤File¤, then ¤New¤.
3) Open menu ¤Options¤, then select ¤Show Real Time Ethernet Compatible Devices¤.

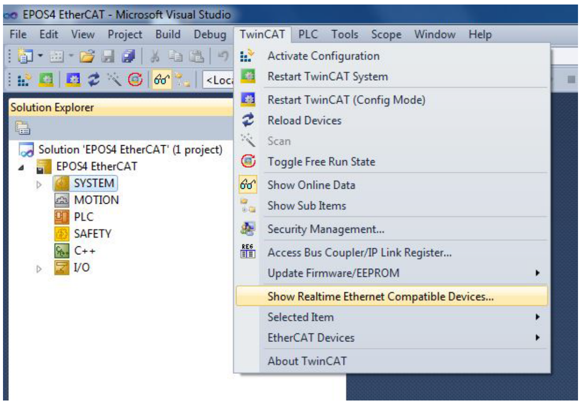

EPOS4 EtherCAT - Microsoft Visual Studio
File Edit View Project Build Debug TwinCAT PLC Tools Scope Window Help
Solution Explorer
Solution 'EPOS4 EtherCAT' (1 project)
EPOS4 EtherCAT
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Activate Configuration
Restart TwinCAT System
Restart TwinCAT (Config Mode)
Reload Devices
Scan
Toggle Free Run State
Show Online Data
Show Sub Items
Security Management...
Access Bus Coupler/IP Link Register...
Update Firmware/EEPROM
Show Realtime Ethernet Compatible Devices...
Selected Item
EtherCAT Devices
About TwinCAT

Figure 4-12 EtherCAT integration – Beckhoff TwinCAT | Create new project

4) If “Installed and ready to use devices” does not list a network card, you will need to install the EtherCAT driver for one of the present network cards.

a) Click one of the listed network cards.

b) Click ¤Install¤.

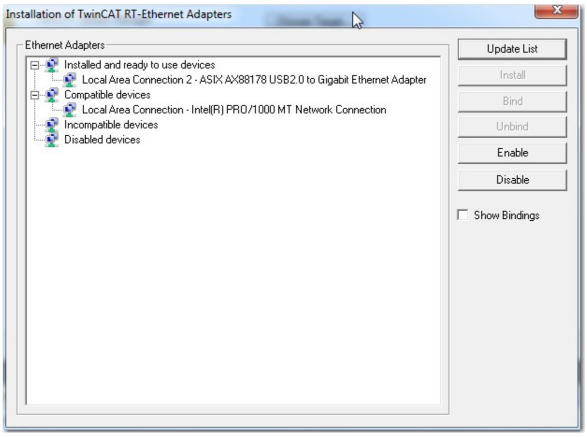

Installation of TwinCAT RT-Ethernet Adapters
Ethernet Adapters
Installed and ready to use devices
Local Area Connection 2 - ASIX AX88178 USB2.0 to Gigabit Ethernet Adapter
Compatible devices
Local Area Connection - Intel(R) PRO/1000 MT Network Connection
Incompatible devices
Disabled devices
Update List
Install
Bind
Unbind
Enable
Disable
□ Show Bindings

Figure 4-13 EtherCAT integration – Beckhoff TwinCAT | Install Ethernet adapters

5) In the TwinCAT System Manager navigation tree, click right on ¤I/O Devices¤, then select ¤Scan¤.

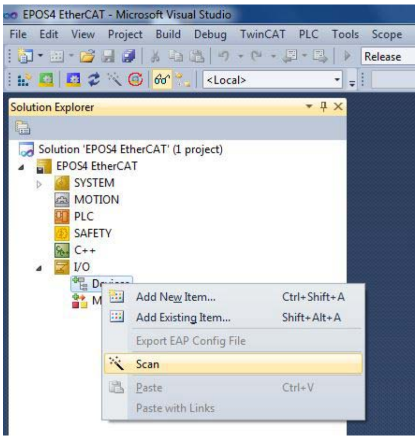
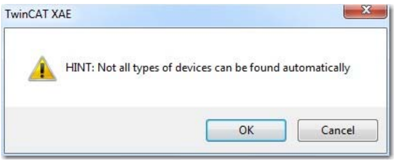

EPOS4 EtherCAT - Microsoft Visual Studio
File Edit View Project Build Debug TwinCAT PLC Tools Scope
Release
<Local>
Solution Explorer
Solution 'EPOS4 EtherCAT' (1 project)
EPOS4 EtherCAT
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Add New Item... Ctrl+Shift+A
Add Existing Item... Shift+Alt+A
Export EAP Config File
Scan
Paste Ctrl+V
Paste with Links

Figure 4-14 EtherCAT integration – Beckhoff TwinCAT | Scan devices

6) Click ¤OK¤ to confirm.



TwinCAT XAE

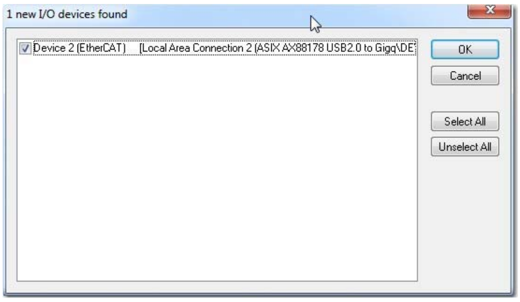
HINT: Not all types of devices can be found automatically

Figure 4-15 EtherCAT integration – Beckhoff TwinCAT | Confirmation

7) All detected E/A devices (network cards) will be listed.

   a) Tick to select the network card to which the EtherCAT devices are connected to and untick all others.

   b) Click ¤OK¤.

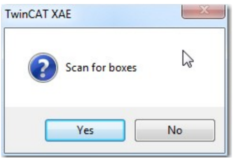


1 new I/O devices found

☑ Device 2 (EtherCAT) [Local Area Connection 2 (ASIX AX88178 USB2.0 to Gigq\DE

Figure 4-16 EtherCAT integration – Beckhoff TwinCAT | New I/O devices found

8) Click ¤YES¤ to confirm.

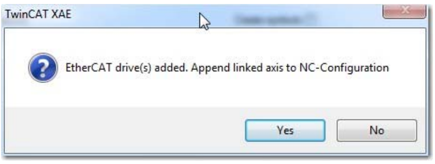


TwinCAT XAE

Scan for boxes

Figure 4-17 EtherCAT integration – Beckhoff TwinCAT | Scan for boxes confirmation

9) The TwinCAT System Manager now searches for connected devices. If one or more controller were found, the following message will appear.
Click ¤Yes¤.



TwinCAT XAE

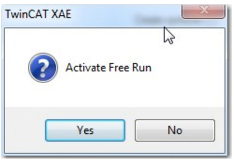
EtherCAT drive(s) added. Append linked axis to NC-Configuration

Figure 4-18 EtherCAT integration – Beckhoff TwinCAT | Add drives message

10) Make your selection depending on the intended use:

- Click ¤Yes¤ if you plan to use the drive as a NC-Configuration
- Click ¤No¤ if you do not plan to use the drive a NC-Configuration

11) Click ¤Yes¤ to confirm.

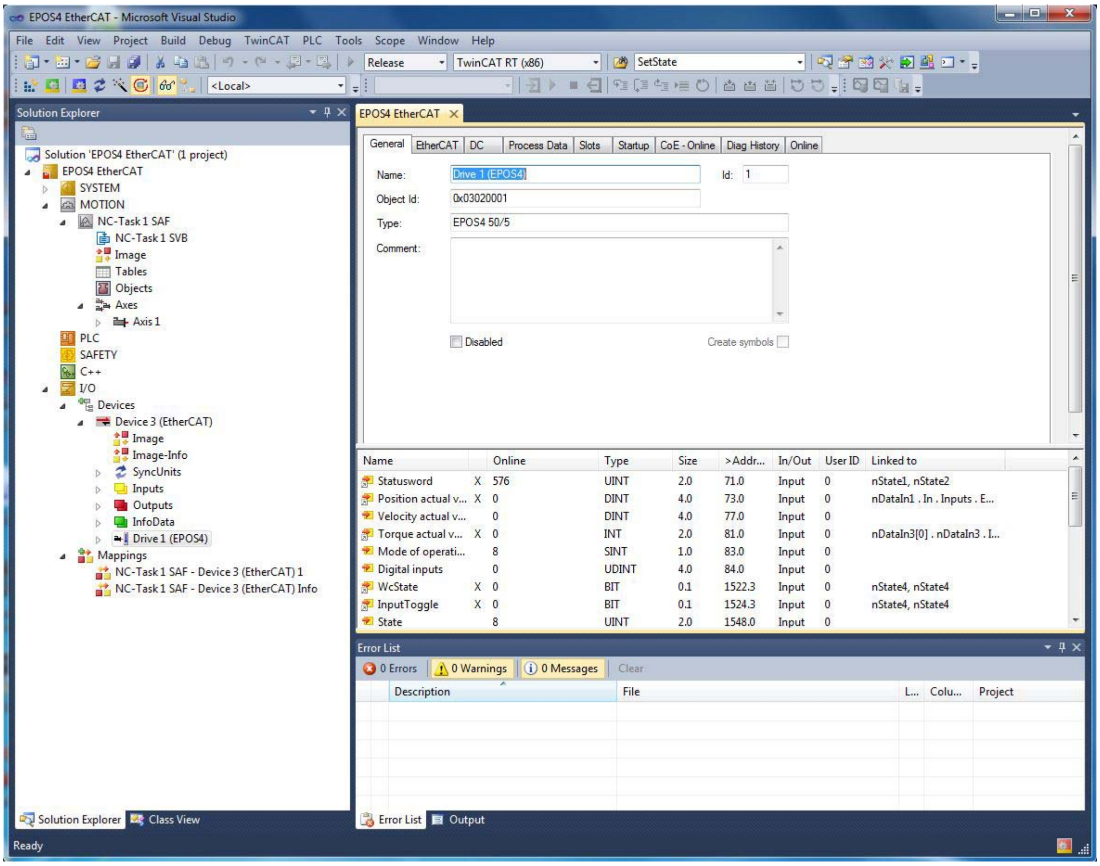


TwinCAT XAE

Activate Free Run

Figure 4-19 EtherCAT integration – Beckhoff TwinCAT | Activate free run message

12) Save the project.



Name: Drive 1 (EPOS4) Id: 1

Object Id: 0x03020001

Type: EPOS4 50/5

Comment:

□ Disabled Create symbols □

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Statusword	X	576	UINT	2.0	71.0	Input	0	nState1, nState2
Position actual v...	X	0	DINT	4.0	73.0	Input	0	nDataIn1 . In . Inputs . E...
Velocity actual v...		0	DINT	4.0	77.0	Input	0	
Torque actual v...	X	0	INT	2.0	81.0	Input	0	nDataIn3[0] . nDataIn3 . I...
Mode of operati...		8	SINT	1.0	83.0	Input	0	
Digital inputs		0	UDINT	4.0	84.0	Input	0	
WcState	X	0	BIT	0.1	1522.3	Input	0	nState4, nState4
InputToggle	X	0	BIT	0.1	1524.3	Input	0	nState4, nState4
State		8	UINT	2.0	1548.0	Input	0	

Error List

0 Errors | 0 Warnings | 0 Messages | Clear

Description	File	L...	Colu...	Project

Figure 4-20 EtherCAT integration – Beckhoff TwinCAT | Save project

4.2.4 Configuration for commanding in a Cyclic Synchronous Mode

Via the EtherCAT interface, usually the following operating modes will be used:

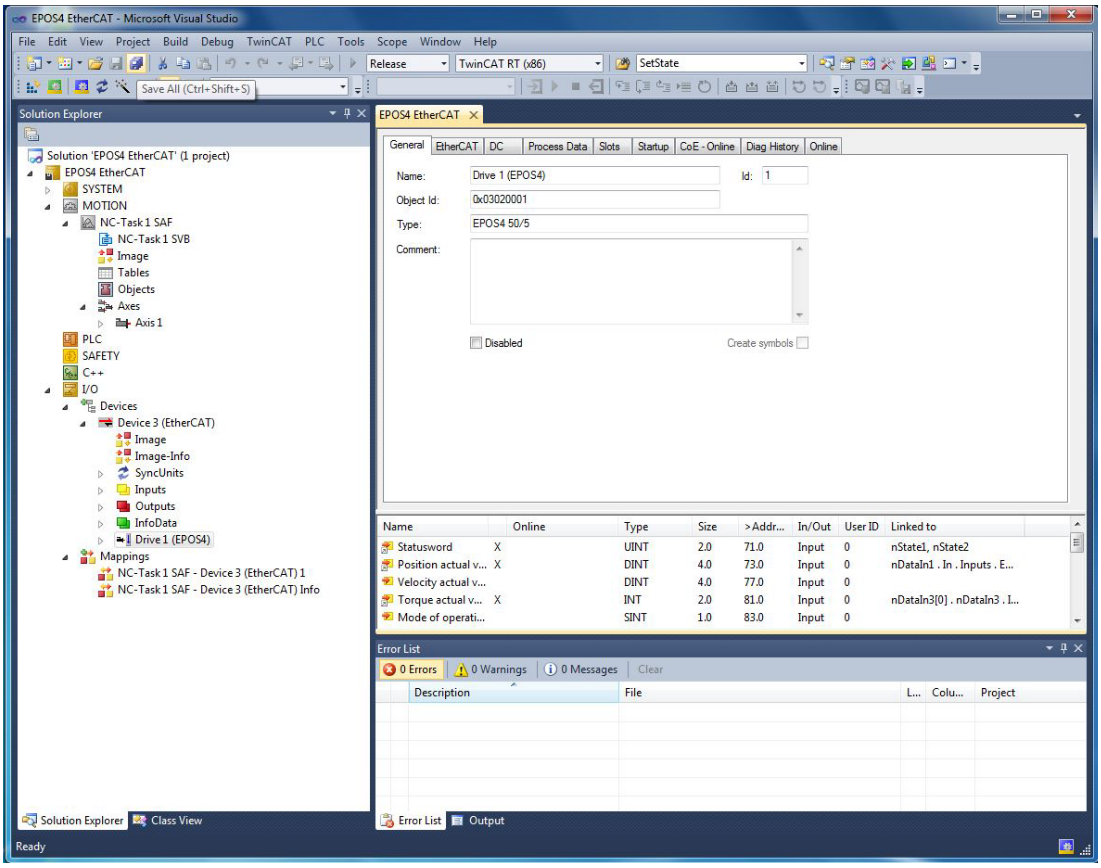
- Cyclic Synchronous Position (CSP)
- Cyclic Synchronous Velocity (CSV)
- Cyclic Synchronous Torque (CST)

If you intend to operate the IDX in Cycle Synchronous Mode, you will need to configure PDO Mapping accordingly by defining “Slots”.

Additionally, the following “regular” EPOS4/IDX operating modes may be used:

- Profile Position Mode (PPM)
- Profile Velocity Mode (PVM)

1) Upon recognition of the involved axes, the structure tree will be displayed (example).



Name: Drive 1 (EPOS4)
Id: 1
Object Id: 0x03020001
Type: EPOS4 50/5
Comment:
□ Disabled
Create symbols □

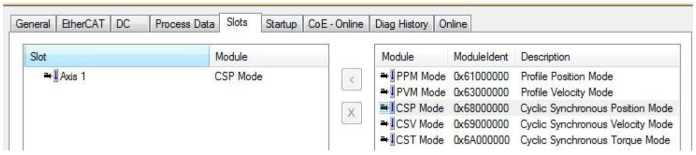
Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Statusword	X		UINT	2.0	71.0	Input	0	nState1, nState2
Position actual v...	X		DINT	4.0	73.0	Input	0	nDataIn1 . In . Inputs . E...
Velocity actual v...			DINT	4.0	77.0	Input	0	
Torque actual v...	X		INT	2.0	81.0	Input	0	nDataIn3[0] . nDataIn3 . I...
Mode of operati...			SINT	1.0	83.0	Input	0	

Figure 4-21 EtherCAT integration – Beckhoff TwinCAT | Structure tree

2) Use the tab ¤Slots¤ to allocate the operating mode to be used:

a) Select a ¤Slot¤ from the left pane.

b) Select the desired operating mode from the right pane ¤Module¤.



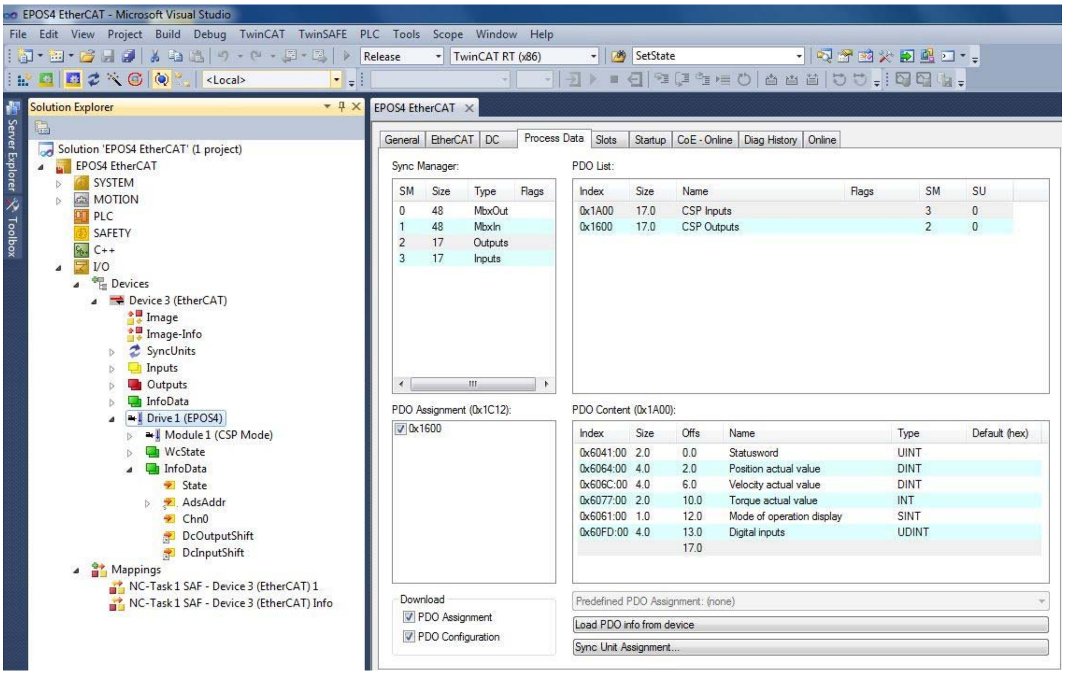
Slot	Module
Axis 1	CSP Mode

Module	ModuleIdent	Description
PPM Mode	0x61000000	Profile Position Mode
PVM Mode	0x63000000	Profile Velocity Mode
CSP Mode	0x68000000	Cyclic Synchronous Position Mode
CSV Mode	0x69000000	Cyclic Synchronous Velocity Mode
CST Mode	0x6A000000	Cyclic Synchronous Torque Mode

Figure 4-22 EtherCAT integration – Beckhoff TwinCAT | Configure slots

4.2.5 Changing PDO Mapping using Beckhoff TwinCAT

1) Select the device in the Solution Explorer, then click the PDO you wish to edit.


EPOS4 EtherCAT - Microsoft Visual Studio
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Tools Scope Window Help
Release
TwinCAT RT (x86)
SetState
<Local>
Solution Explorer
Solution 'EPOS4 EtherCAT' (1 project)
EPOS4 EtherCAT
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Devices
Device 3 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Outputs
InfoData
Drive 1 (EPOS4)
Module 1 (CSP Mode)
WcState
InfoData
State
AdsAddr
Chn0
DcOutputShift
DcInputShift
Mappings
NC-Task 1 SAF - Device 3 (EtherCAT) 1
NC-Task 1 SAF - Device 3 (EtherCAT) Info
EPOS4 EtherCAT
General EtherCAT DC Process Data Slots Startup CoE - Online Diag History Online

Sync Manager:

SM	Size	Type	Flags
0	48	MbxOut	
1	48	MbxIn	
2	17	Outputs	
3	17	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	17.0	CSP Inputs		3	0
0x1600	17.0	CSP Outputs		2	0

PDO Assignment (0x1C12):

☑ 0x1600

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x6041:00	2.0	0.0	Statusword	UINT	
0x6064:00	4.0	2.0	Position actual value	DINT	
0x606C:00	4.0	6.0	Velocity actual value	DINT	
0x6077:00	2.0	10.0	Torque actual value	INT	
0x6061:00	1.0	12.0	Mode of operation display	SINT	
0x60FD:00	4.0	13.0	Digital inputs	UDINT	
		17.0			

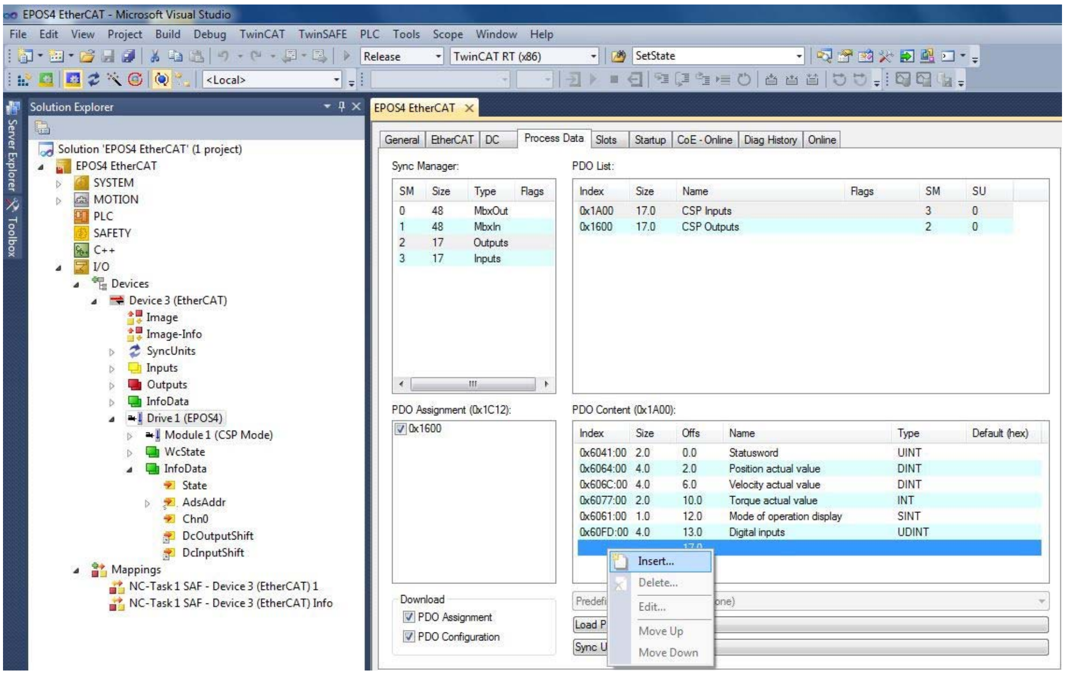
Download
☑ PDO Assignment
☑ PDO Configuration

Predefined PDO Assignment: (none)

Figure 4-23 EtherCAT integration – Beckhoff TwinCAT | Display process data

2) Click the desired preconfigured PDO mapping from the list. Click right to open the context menu.

3) Choose either ¤Delete¤ to remove an existing variable or ¤Insert¤ to add a new variable.


EPOS4 EtherCAT - Microsoft Visual Studio
File Edit View Project Build Debug TwinCAT TwinSAFE PLC Tools Scope Window Help
Release
TwinCAT RT (x86)
SetState
<Local>
Solution Explorer
Solution 'EPOS4 EtherCAT' (1 project)
EPOS4 EtherCAT
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Devices
Device 3 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Outputs
InfoData
Drive 1 (EPOS4)
Module 1 (CSP Mode)
WcState
InfoData
State
AdsAddr
Chn0
DcOutputShift
DcInputShift
Mappings
NC-Task 1 SAF - Device 3 (EtherCAT) 1
NC-Task 1 SAF - Device 3 (EtherCAT) Info
EPOS4 EtherCAT
General EtherCAT DC Process Data Slots Startup CoE - Online Diag History Online

Sync Manager:

SM	Size	Type	Flags
0	48	MbxOut	
1	48	MbxIn	
2	17	Outputs	
3	17	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	17.0	CSP Inputs		3	0
0x1600	17.0	CSP Outputs		2	0

PDO Assignment (0x1C12):

☑ 0x1600

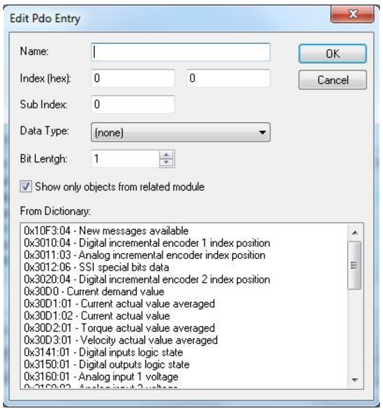
PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x6041:00	2.0	0.0	Statusword	UINT	
0x6064:00	4.0	2.0	Position actual value	DINT	
0x606C:00	4.0	6.0	Velocity actual value	DINT	
0x6077:00	2.0	10.0	Torque actual value	INT	
0x6061:00	1.0	12.0	Mode of operation display	SINT	
0x60FD:00	4.0	13.0	Digital inputs	UDINT	
		17.0			

Download
☑ PDO Assignment
☑ PDO Configuration

Figure 4-24 EtherCAT integration – Beckhoff TwinCAT | Select PDO

4) Select the object you wish to map and click ¤OK¤.



Edit Pdo Entry

Name:

Index (hex): 0 0

Sub Index: 0

Data Type: (none)

Bit Lentgh: 1

☑ Show only objects from related module

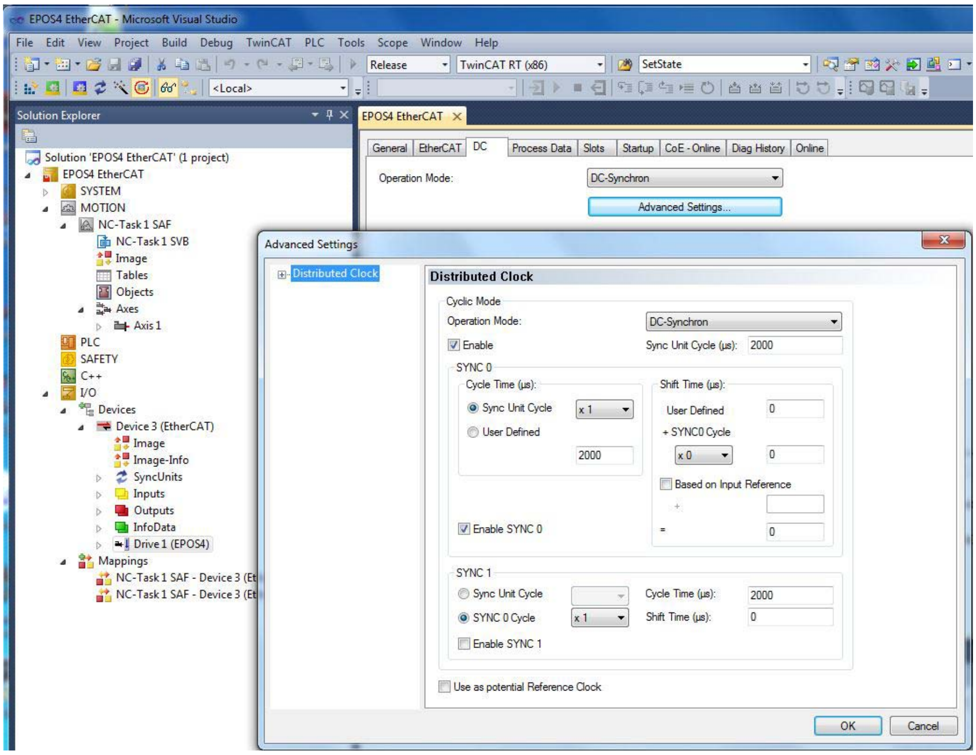
From Dictionary:

0x10F3:04 - New messages available
0x3010:04 - Digital incremental encoder 1 index position
0x3011:03 - Analog incremental encoder index position
0x3012:06 - SSI special bits data
0x3020:04 - Digital incremental encoder 2 index position
0x30D0 - Current demand value
0x30D1:01 - Current actual value averaged
0x30D1:02 - Current actual value
0x30D2:01 - Torque actual value averaged
0x30D3:01 - Velocity actual value averaged
0x3141:01 - Digital inputs logic state
0x3150:01 - Digital outputs logic state
0x3160:01 - Analog input 1 voltage

Figure 4-25 EtherCAT integration – Beckhoff TwinCAT | Edit PDO values

VERIFY CSP SETTINGS

5) Enable the Distributed Clock from the EPOS4/IDX.



Operation Mode: DC-Synchron

Advanced Settings...

Distributed Clock

Cyclic Mode

Operation Mode: DC-Synchron

☑ Enable Sync Unit Cycle (µs): 2000

SYNC 0

Cycle Time (µs):

● Sync Unit Cycle x 1

○ User Defined

2000

Shift Time (µs):

User Defined 0

+ SYNC0 Cycle

x 0 0

□ Based on Input Reference

+

= 0

☑ Enable SYNC 0

SYNC 1

○ Sync Unit Cycle Cycle Time (µs): 2000

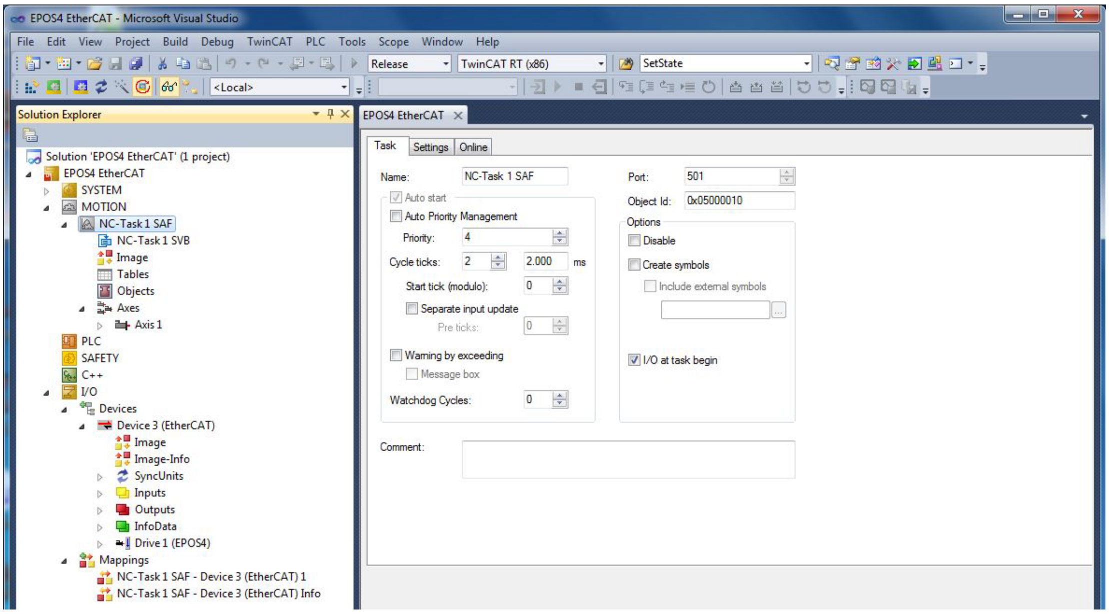
● SYNC 0 Cycle x 1 Shift Time (µs): 0

□ Enable SYNC 1

□ Use as potential Reference Clock

Figure 4-26 EtherCAT integration – Beckhoff TwinCAT | Set distributed clock

6) In the Solution Explorer, click on tree item ¤NC-Task 1 SAF¤, then tab ¤Task¤.
Set cycle time to 2 ms.



Name: NC-Task 1 SAF

☑ Auto start

☐ Auto Priority Management

Priority: 4

Cycle ticks: 2 2.000 ms

Start tick (modulo): 0

☐ Separate input update

Pre ticks: 0

☐ Warning by exceeding

☐ Message box

Watchdog Cycles: 0

Port: 501

Object Id: 0x05000010

Options

☐ Disable

☐ Create symbols

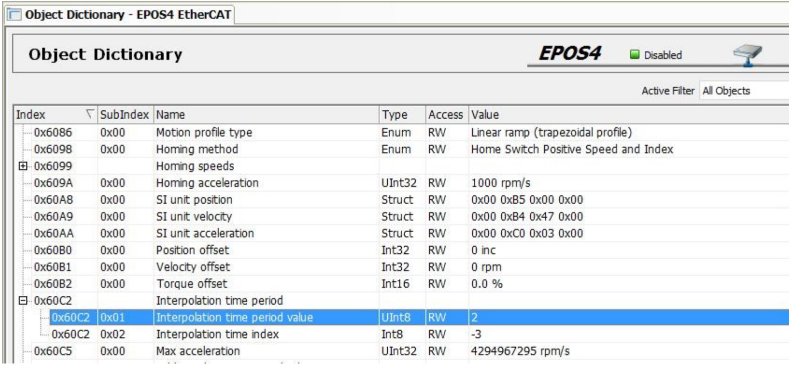
☐ Include external symbols

☑ I/O at task begin

Comment:

Figure 4-27 EtherCAT integration – Beckhoff TwinCAT | Set cycle ticks 1

7) For **CSP** and **CSV** mode, set object 0x60C2-01 to the same value as the “Cyclic ticks” in ¤NC-Task 1 SAF¤ (➔step 6). For **CST** mode, set it to zero.



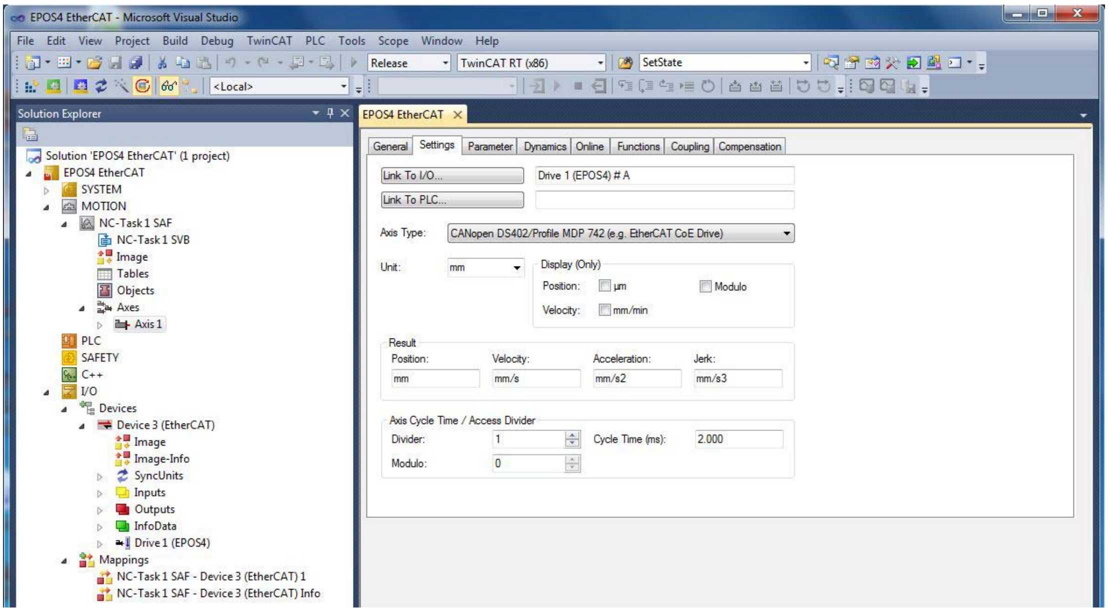
Active Filter All Objects

Index	SubIndex	Name	Type	Access	Value
0x6086	0x00	Motion profile type	Enum	RW	Linear ramp (trapezoidal profile)
0x6098	0x00	Homing method	Enum	RW	Home Switch Positive Speed and Index
⊞ 0x6099		Homing speeds			
0x609A	0x00	Homing acceleration	UInt32	RW	1000 rpm/s
0x60A8	0x00	SI unit position	Struct	RW	0x00 0xB5 0x00 0x00
0x60A9	0x00	SI unit velocity	Struct	RW	0x00 0xB4 0x47 0x00
0x60AA	0x00	SI unit acceleration	Struct	RW	0x00 0xC0 0x03 0x00
0x60B0	0x00	Position offset	Int32	RW	0 inc
0x60B1	0x00	Velocity offset	Int32	RW	0 rpm
0x60B2	0x00	Torque offset	Int16	RW	0.0 %
⊟ 0x60C2		Interpolation time period			
0x60C2	0x01	Interpolation time period value	UInt8	RW	2
0x60C2	0x02	Interpolation time index	Int8	RW	-3
0x60C5	0x00	Max acceleration	UInt32	RW	4294967295 rpm/s

Figure 4-28 EtherCAT integration – Beckhoff TwinCAT | Set cycle ticks 2

4.2.6 Configuration of the Axis

1) In the tab ¤Settings¤, verify that ¤Link To I/O…¤ is assigned to the EPOS4/IDX axis (naming is by your choice).



Link To I/O...: Drive 1 (EPOS4) # A

Link To PLC...:

Axis Type: CANopen DS402/Profile MDP 742 (e.g. EtherCAT CoE Drive)

Unit: mm

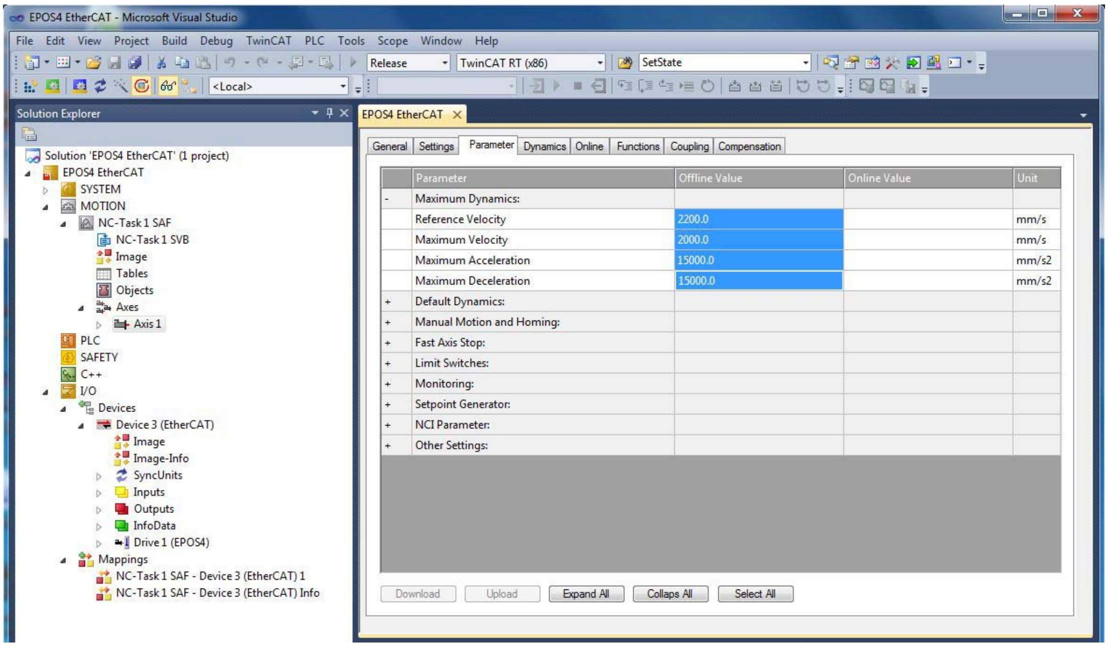
Display (Only)
Position: □ µm □ Modulo
Velocity: □ mm/min

Result
Position: mm
Velocity: mm/s
Acceleration: mm/s2
Jerk: mm/s3

Axis Cycle Time / Access Divider
Divider: 1
Cycle Time (ms): 2.000
Modulo: 0

Figure 4-29 EtherCAT integration – Beckhoff TwinCAT | Link axis

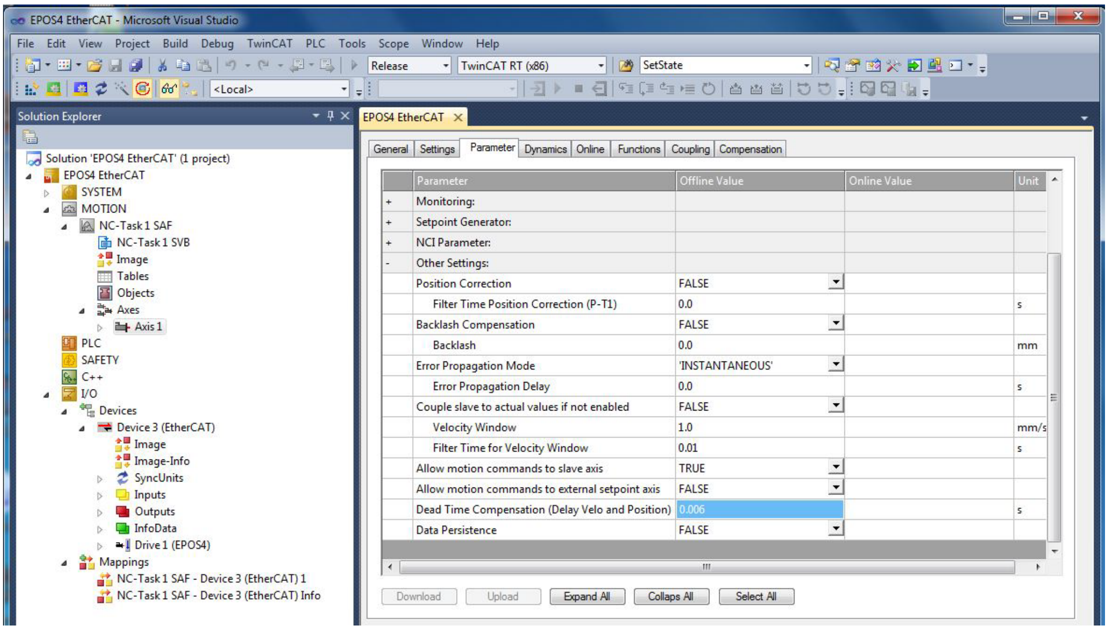
2) In the tab ¤Parameter¤, adjust the motor speed settings as to the motor’s capability and to the supply voltage.



	Parameter	Offline Value	Online Value	Unit
-	Maximum Dynamics:			
	Reference Velocity	2200.0		mm/s
	Maximum Velocity	2000.0		mm/s
	Maximum Acceleration	15000.0		mm/s2
	Maximum Deceleration	15000.0		mm/s2
+	Default Dynamics:			
+	Manual Motion and Homing:			
+	Fast Axis Stop:			
+	Limit Switches:			
+	Monitoring:			
+	Setpoint Generator:			
+	NCI Parameter:			
+	Other Settings:			

Figure 4-30 EtherCAT integration – Beckhoff TwinCAT | Set speed settings

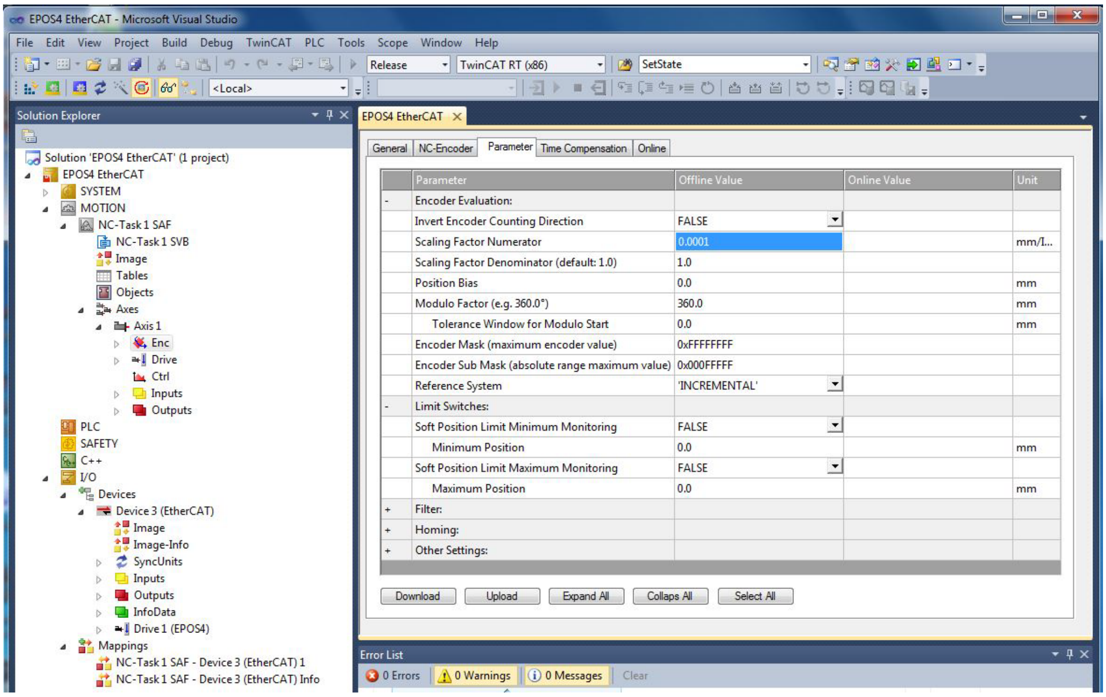
3) Set Dead Time Compensation to approximately three to four times the set NC-Task SAF Cycle ticks (➔“Verify CSP Settings” on page 4-36; step 5)


EPOS4 EtherCAT - Microsoft Visual Studio
File Edit View Project Build Debug TwinCAT PLC Tools Scope Window Help
Release
TwinCAT RT (x86)
SetState
<Local>
Solution Explorer
Solution 'EPOS4 EtherCAT' (1 project)
EPOS4 EtherCAT
SYSTEM
MOTION
NC-Task 1 SAF
NC-Task 1 SVB
Image
Tables
Objects
Axes
Axis 1
PLC
SAFETY
C++
I/O
Devices
Device 3 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Outputs
InfoData
Drive 1 (EPOS4)
Mappings
NC-Task 1 SAF - Device 3 (EtherCAT) 1
NC-Task 1 SAF - Device 3 (EtherCAT) Info
EPOS4 EtherCAT
General Settings Parameter Dynamics Online Functions Coupling Compensation
Download Upload Expand All Collaps All Select All

	Parameter	Offline Value	Online Value	Unit
+	Monitoring:			
+	Setpoint Generator:			
+	NCI Parameter:			
-	Other Settings:			
	Position Correction	FALSE		
	Filter Time Position Correction (P-T1)	0.0		s
	Backlash Compensation	FALSE		
	Backlash	0.0		mm
	Error Propagation Mode	'INSTANTANEOUS'		
	Error Propagation Delay	0.0		s
	Couple slave to actual values if not enabled	FALSE		
	Velocity Window	1.0		mm/s
	Filter Time for Velocity Window	0.01		s
	Allow motion commands to slave axis	TRUE		
	Allow motion commands to external setpoint axis	FALSE		
	Dead Time Compensation (Delay Velo and Position)	0.006		s
	Data Persistence	FALSE		

Figure 4-31 EtherCAT integration – Beckhoff TwinCAT | Set dead time compensation

4) Make sure to set the correct encoder resolution.

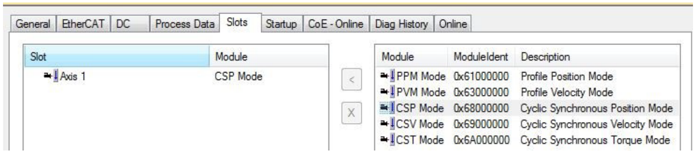

EPOS4 EtherCAT - Microsoft Visual Studio
File Edit View Project Build Debug TwinCAT PLC Tools Scope Window Help
Release
TwinCAT RT (x86)
SetState
<Local>
Solution Explorer
Solution 'EPOS4 EtherCAT' (1 project)
EPOS4 EtherCAT
SYSTEM
MOTION
NC-Task 1 SAF
NC-Task 1 SVB
Image
Tables
Objects
Axes
Axis 1
Enc
Drive
Ctrl
Inputs
Outputs
PLC
SAFETY
C++
I/O
Devices
Device 3 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Outputs
InfoData
Drive 1 (EPOS4)
Mappings
NC-Task 1 SAF - Device 3 (EtherCAT) 1
NC-Task 1 SAF - Device 3 (EtherCAT) Info
EPOS4 EtherCAT
General NC-Encoder Parameter Time Compensation Online
Download Upload Expand All Collaps All Select All
Error List
0 Errors 0 Warnings 0 Messages Clear

	Parameter	Offline Value	Online Value	Unit
-	Encoder Evaluation:			
	Invert Encoder Counting Direction	FALSE		
	Scaling Factor Numerator	0.0001		mm/I...
	Scaling Factor Denominator (default: 1.0)	1.0		
	Position Bias	0.0		mm
	Modulo Factor (e.g. 360.0°)	360.0		mm
	Tolerance Window for Modulo Start	0.0		mm
	Encoder Mask (maximum encoder value)	0xFFFFFFFF		
	Encoder Sub Mask (absolute range maximum value)	0x000FFFFF		
	Reference System	'INCREMENTAL'		
-	Limit Switches:			
	Soft Position Limit Minimum Monitoring	FALSE		
	Minimum Position	0.0		mm
	Soft Position Limit Maximum Monitoring	FALSE		
	Maximum Position	0.0		mm
+	Filter:			
+	Homing:			
+	Other Settings:			

Figure 4-32 EtherCAT integration – Beckhoff TwinCAT | Set encoder settings

5) Configure the modes as follows:

SETTINGS FOR CSP MODE



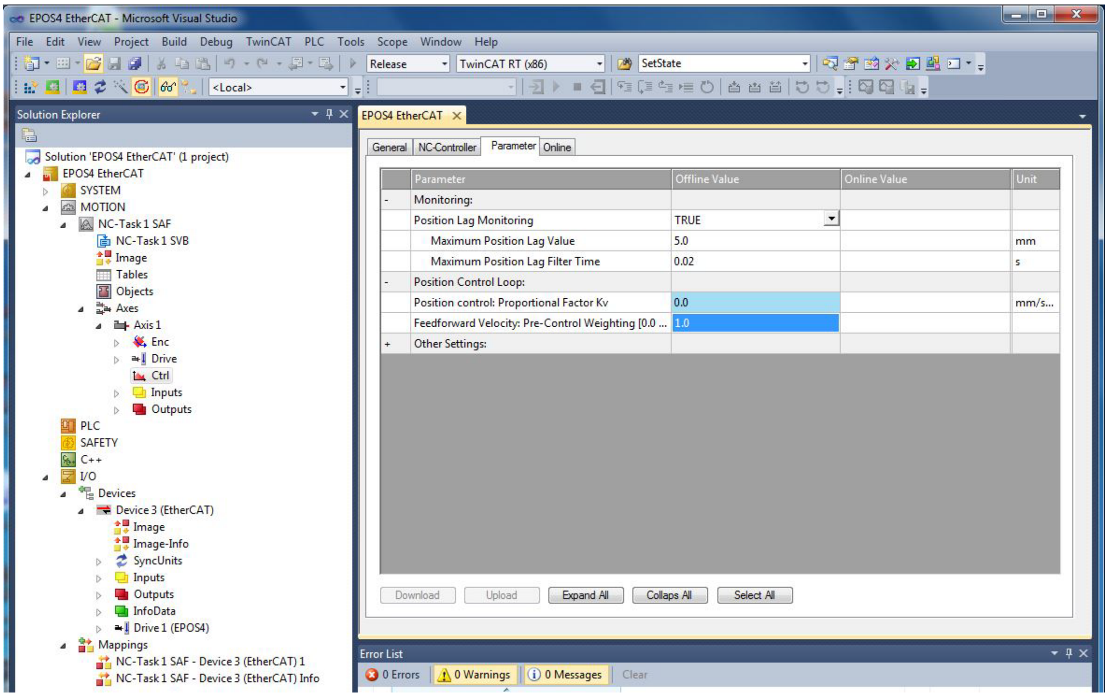
Slot	Module
Axis 1	CSP Mode

Module	ModuleIdent	Description
PPM Mode	0x61000000	Profile Position Mode
PVM Mode	0x63000000	Profile Velocity Mode
CSP Mode	0x68000000	Cyclic Synchronous Position Mode
CSV Mode	0x69000000	Cyclic Synchronous Velocity Mode
CST Mode	0x6A000000	Cyclic Synchronous Torque Mode

Figure 4-33 EtherCAT integration – Beckhoff TwinCAT | Set CSP settings

Configure the position control loop as follows:

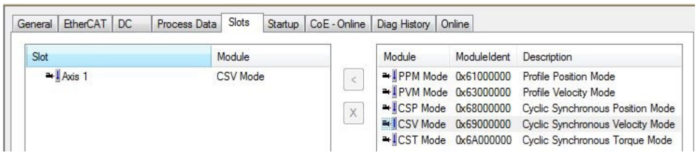
- Position control: Proportional Factor Kv → select a Kv factor suitable for your drive system
- Feedforward Velocity: Pre-Control Weighting [0.0…1.0] → “1.0”



	Parameter	Offline Value	Online Value	Unit
-	Monitoring:			
	Position Lag Monitoring	TRUE		
	Maximum Position Lag Value	5.0		mm
	Maximum Position Lag Filter Time	0.02		s
-	Position Control Loop:			
	Position control: Proportional Factor Kv	0.0		mm/s...
	Feedforward Velocity: Pre-Control Weighting [0.0 ...	1.0		
+	Other Settings:			

Figure 4-34 EtherCAT integration – Beckhoff TwinCAT | Set position control loop settings

SETTINGS FOR CSV MODE



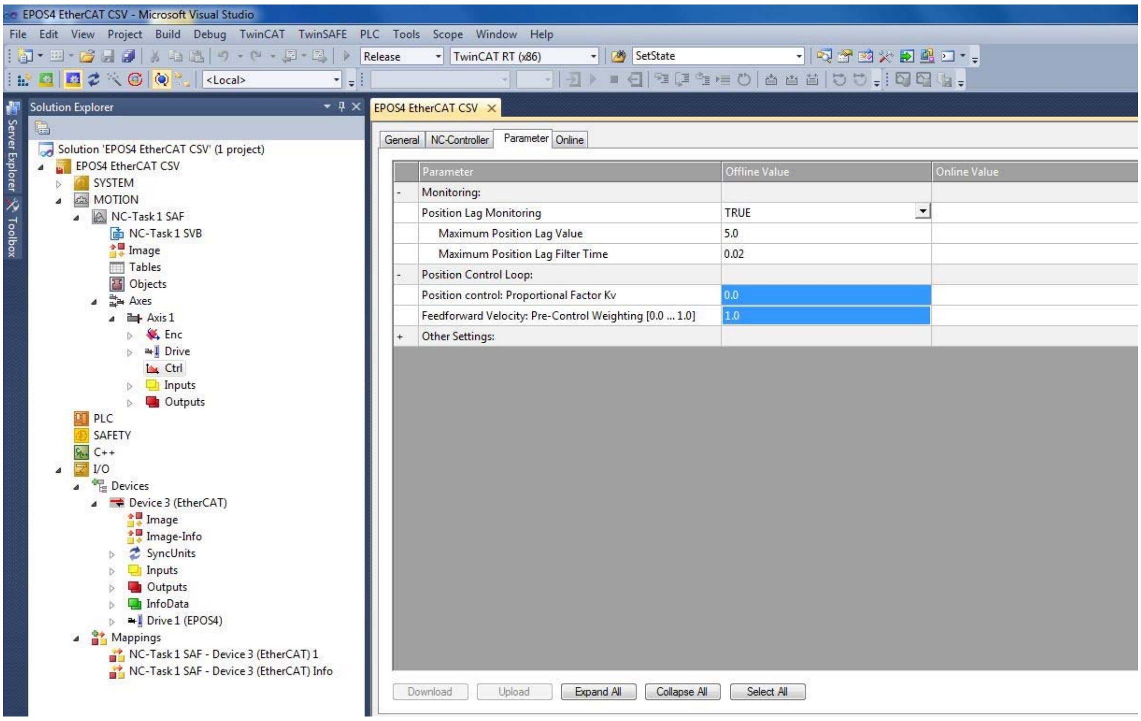
Slot	Module
Axis 1	CSV Mode

Module	ModuleIdent	Description
PPM Mode	0x61000000	Profile Position Mode
PVM Mode	0x63000000	Profile Velocity Mode
CSP Mode	0x68000000	Cyclic Synchronous Position Mode
CSV Mode	0x69000000	Cyclic Synchronous Velocity Mode
CST Mode	0x6A000000	Cyclic Synchronous Torque Mode

Figure 4-35 EtherCAT integration – Beckhoff TwinCAT | Set CSV settings

Configure the position control loop as follows:

- Position control: Proportional Factor Kv → “0.0”
- Feedforward Velocity: Pre-Control Weighting [0.0…1.0] → “1.0”



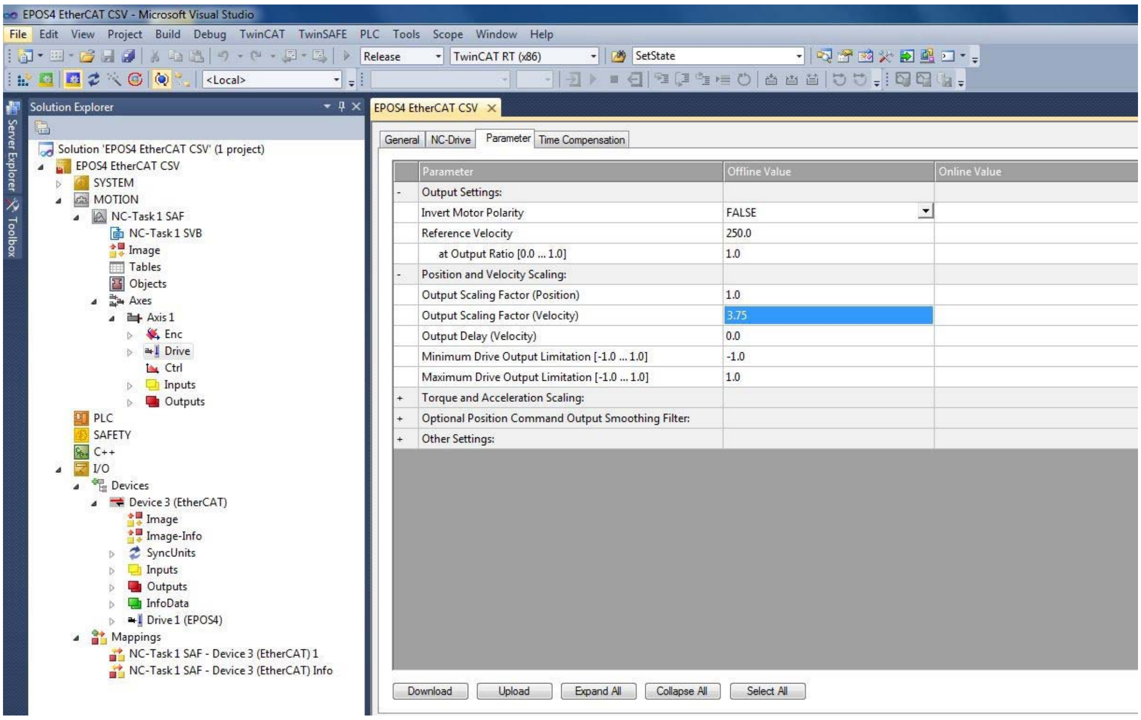
	Parameter	Offline Value	Online Value
-	Monitoring:		
	Position Lag Monitoring	TRUE	
	Maximum Position Lag Value	5.0	
	Maximum Position Lag Filter Time	0.02	
-	Position Control Loop:		
	Position control: Proportional Factor Kv	0.0	
	Feedforward Velocity: Pre-Control Weighting [0.0 ... 1.0]	1.0	
+	Other Settings:		

Figure 4-36 EtherCAT integration – Beckhoff TwinCAT | Set position control loop settings

In the tab ¤Parameter¤, set the correct “Output Scaling Factor (Velocity)”. Scaling may be calculated as follows:

Scaling = 7500 / (Encoder count number * 4)

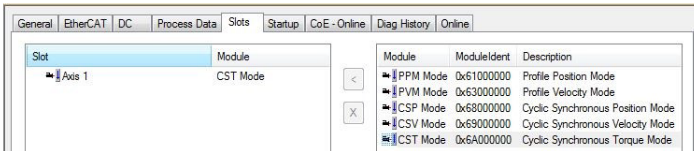
e.g. Encoder with 500 counts per turn: Scaling = 7500 / (500 * 4) = 3.75



	Parameter	Offline Value	Online Value
-	Output Settings:		
	Invert Motor Polarity	FALSE	
	Reference Velocity	250.0	
	at Output Ratio [0.0 ... 1.0]	1.0	
-	Position and Velocity Scaling:		
	Output Scaling Factor (Position)	1.0	
	Output Scaling Factor (Velocity)	3.75	
	Output Delay (Velocity)	0.0	
	Minimum Drive Output Limitation [-1.0 ... 1.0]	-1.0	
	Maximum Drive Output Limitation [-1.0 ... 1.0]	1.0	
+	Torque and Acceleration Scaling:		
+	Optional Position Command Output Smoothing Filter:		
+	Other Settings:		

Figure 4-37 EtherCAT integration – Beckhoff TwinCAT | Set output scaling factor

SETTINGS FOR CST MODE



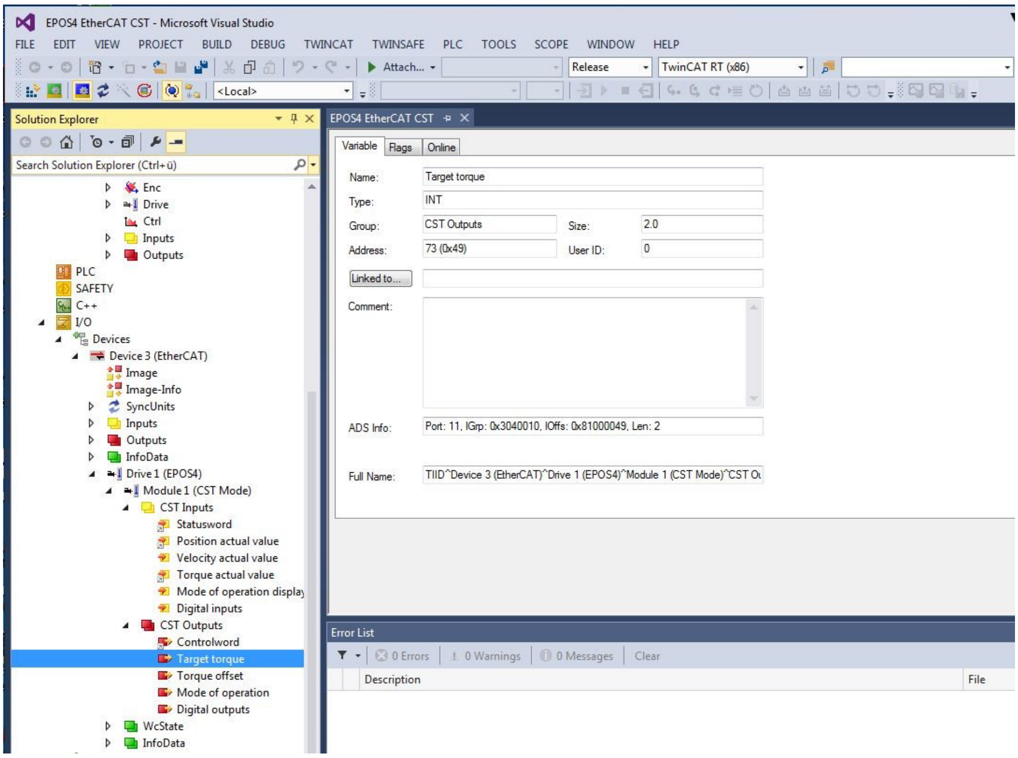
General | EtherCAT | DC | Process Data | Slots | Startup | CoE - Online | Diag History | Online

Slot	Module
Axis 1	CST Mode

Module	ModuleIdent	Description
PPM Mode	0x61000000	Profile Position Mode
PVM Mode	0x63000000	Profile Velocity Mode
CSP Mode	0x68000000	Cyclic Synchronous Position Mode
CSV Mode	0x69000000	Cyclic Synchronous Velocity Mode
CST Mode	0x6A000000	Cyclic Synchronous Torque Mode

Figure 4-38 EtherCAT integration – Beckhoff TwinCAT | Set CST settings

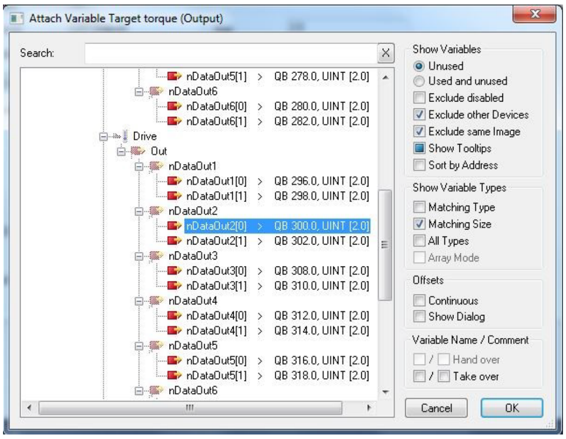
In the Solution Explorer, select ¤CST Outputs¤ and set the link for the “Target Torque” variable.



Name: Target torque
Type: INT
Group: CST Outputs Size: 2.0
Address: 73 (0x49) User ID: 0
Comment:
ADS Info: Port: 11, IGrp: 0x3040010, IOffs: 0x81000049, Len: 2
Full Name: TIID^Device 3 (EtherCAT)^Drive 1 (EPOS4)^Module 1 (CST Mode)^CST Ou

Figure 4-39 EtherCAT integration – Beckhoff TwinCAT | Set target torque

In folder ¤Drive¤ \ ¤Out¤, select “nDataOut2(0)” of Axis 1 as link variable.



```
nDataOut5[1] > QB 278.0, UINT [2.0]
nDataOut6
  nDataOut6[0] > QB 280.0, UINT [2.0]
  nDataOut6[1] > QB 282.0, UINT [2.0]
Drive
  Out
    nDataOut1
      nDataOut1[0] > QB 296.0, UINT [2.0]
      nDataOut1[1] > QB 298.0, UINT [2.0]
    nDataOut2
      nDataOut2[0] > QB 300.0, UINT [2.0]
      nDataOut2[1] > QB 302.0, UINT [2.0]
    nDataOut3
      nDataOut3[0] > QB 308.0, UINT [2.0]
      nDataOut3[1] > QB 310.0, UINT [2.0]
    nDataOut4
      nDataOut4[0] > QB 312.0, UINT [2.0]
      nDataOut4[1] > QB 314.0, UINT [2.0]
    nDataOut5
      nDataOut5[0] > QB 316.0, UINT [2.0]
      nDataOut5[1] > QB 318.0, UINT [2.0]
    nDataOut6
```

Show Variables
- ● Unused
- ○ Used and unused
- □ Exclude disabled
- ☑ Exclude other Devices
- ☑ Exclude same Image
- ■ Show Tooltips
- □ Sort by Address

Show Variable Types
- □ Matching Type
- ☑ Matching Size
- □ All Types
- □ Array Mode

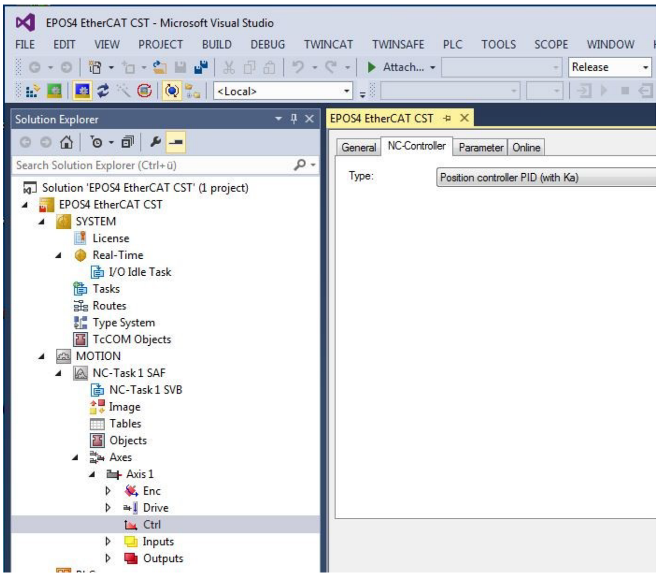
Offsets
- □ Continuous
- □ Show Dialog

Variable Name / Comment
- □ / □ Hand over
- □ / □ Take over

Figure 4-40 EtherCAT integration – Beckhoff TwinCAT | Configure position control loop

Configure the position control loop as follows:

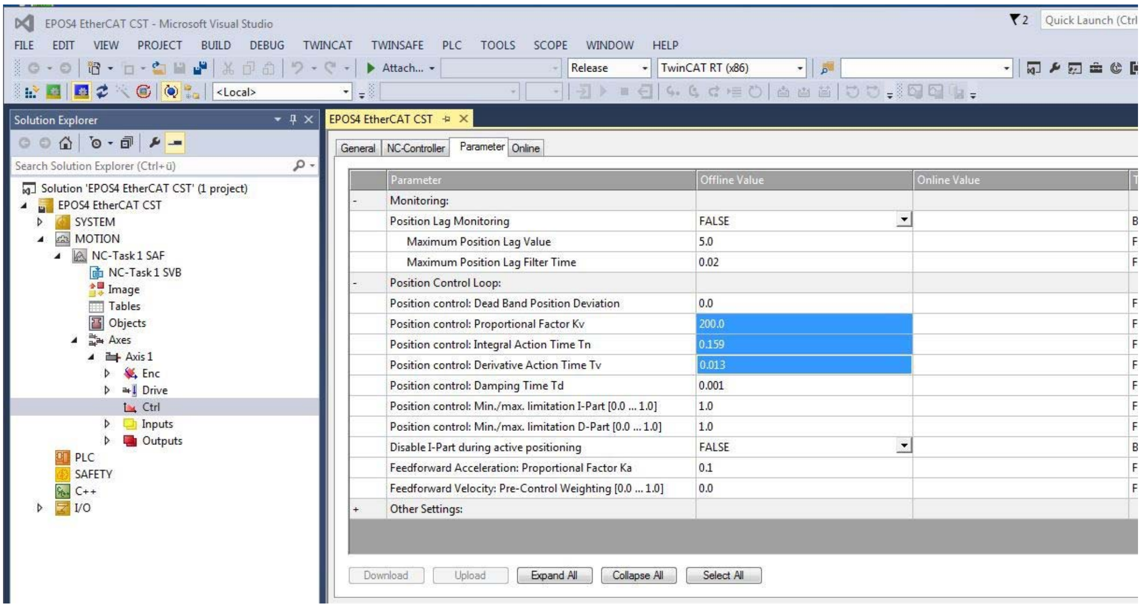
- NC-Controller Type: Position controller PID (with Ka)



Type: Position controller PID (with Ka)

Figure 4-41 EtherCAT integration – Beckhoff TwinCAT | Configure position control type

- Tn = Kp / Ki (EPOS4/IDX object 0x30A1-01 and object 0x30A1-02)
- Tv = Kd / Kp (EPOS4/IDX object 0x30A1-03 and object 0x30A1-01)
- Kv must be termined empirically



	Parameter	Offline Value	Online Value	T
-	Monitoring:			
	Position Lag Monitoring	FALSE		B
	Maximum Position Lag Value	5.0		F
	Maximum Position Lag Filter Time	0.02		F
-	Position Control Loop:			
	Position control: Dead Band Position Deviation	0.0		F
	Position control: Proportional Factor Kv	200.0		F
	Position control: Integral Action Time Tn	0.159		F
	Position control: Derivative Action Time Tv	0.013		F
	Position control: Damping Time Td	0.001		F
	Position control: Min./max. limitation I-Part [0.0 ... 1.0]	1.0		F
	Position control: Min./max. limitation D-Part [0.0 ... 1.0]	1.0		F
	Disable I-Part during active positioning	FALSE		B
	Feedforward Acceleration: Proportional Factor Ka	0.1		F
	Feedforward Velocity: Pre-Control Weighting [0.0 ... 1.0]	0.0		F
+	Other Settings:			

Figure 4-42 EtherCAT integration – Beckhoff TwinCAT | Configure position control parameters

4.3 zub’s MACS Multi-Axis EtherCAT Masters

OBJECTIVE

This chapter explains the required configuration of so-called «MasterMACS» or «MACS5» multi-axis motion controllers to command an EPOS4/IDX by EtherCAT.

BASICS

The MACS master controller product series of the Swiss company zub AG (➔www.zub.ch/en) are freely programmable just as a PLC but are mainly designed for sophisticated coordination and synchronization of multi-axis drive systems. These masters can process the motion of one or multiple axis and command the EPOS4/IDX via CAN or EtherCAT.
As member of the maxon group, the company «zub machine control AG» provides you with industry-proven, high-end solutions that are supported by both maxon motor ag and zub AG.

Available are different types of masters: ➔www.zub.ch/en/products/product-gallery.html

The information given in this chapter refers to the following two product types. They are commanding the EPOS4/IDX in Cyclic Synchronous Position (CSP) mode via EtherCAT:

- MasterMACS
- MACS5

With other MACS product types, with other EPOS4/IDX operating modes (e.g. CSV), or for commanding via CAN (instead of EtherCAT), an adapted configuration will be required.

PRECONDITIONS

The information given in this chapter presumes that there is some level of experience present concerning the functionality and usage of zub’s development environment («APOSS») as well as to programming language.

The software and all manuals are free for download from zub’s website:
➔www.zub.ch/en/downloads.html

FUNDAMENTALS

Typical PLCs use the *:esi file and a System Manager tool to configure master and slave. The MACS software development environment does not offer such a System Manager tool. The configuration of the communication and data exchange is defined as part of the application’s source code. This code section will be typically handled by an include file (*.mi) which can easily be copied into application programs. Thus, configuration can be quite simple but remains still very flexible.

The present chapter provides the code of a typical configuration of the MACS master and the EPOS4/IDX commanded via EtherCAT based on the CSP mode. You may copy/paste the required code from the document into the source code editor of the APOSS development environment in use by MACS controllers.

Take note of the remarks and explanations in the code. They will help you to get a better understanding on the different configuration tasks and the possible additional motor configuration which must be checked or adapted.

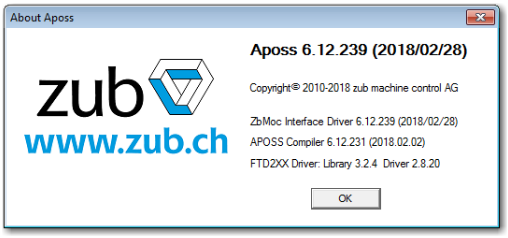
REQUIRED TOOLS

Software Development Environment

APOSS

V 6.12.239 (or higher)

Menu item Help
- About Program



Aposs 6.12.239 (2018/02/28)

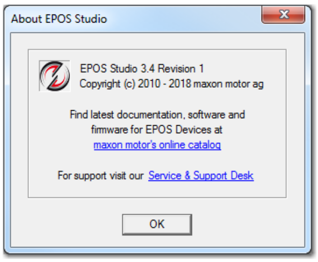
Copyright© 2010-2018 zub machine control AG

ZbMoc Interface Driver 6.12.239 (2018/02/28)
APOSS Compiler 6.12.231 (2018.02.02)
FTD2XX Driver: Library 3.2.4 Driver 2.8.20

EPOS Studio

V 3.6 (or higher)

Menu item Help
- About EPOS Studio



EPOS Studio 3.4 Revision 1
Copyright (c) 2010 - 2018 maxon motor ag

Find latest documentation, software and firmware for EPOS Devices at maxon motor's online catalog

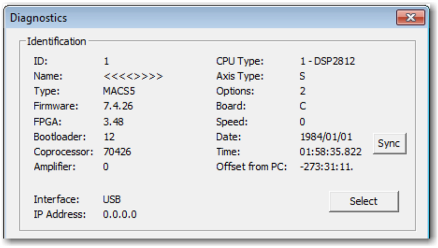
For support visit our Service & Support Desk

Firmware Versions

MACS5 / MasterMACS

7.4.26 (or higher)

Menu item Controller
- Diagnostics

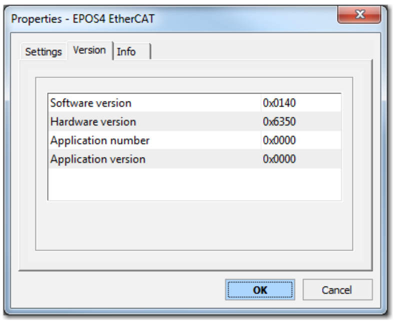


ID:	1	CPU Type:	1 - DSP2812
Name:	<<<<>>>>	Axis Type:	S
Type:	MACS5	Options:	2
Firmware:	7.4.26	Board:	C
FPGA:	3.48	Speed:	0
Bootloader:	12	Date:	1984/01/01
Coprocessor:	70426	Time:	01:58:35.822
Amplifier:	0	Offset from PC:	-273:31:11.
Interface:	USB		
IP Address:	0.0.0.0		

EPOS4/IDX

0x160 (or higher)

- Icon
- Properties
- Version

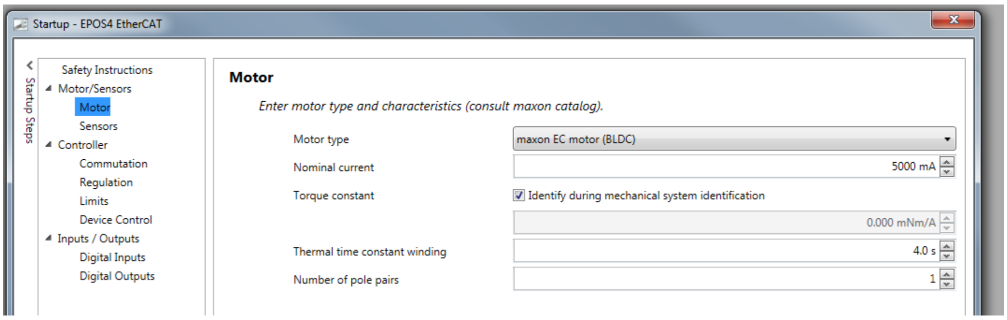


Software version	0x0140
Hardware version	0x6350
Application number	0x0000
Application version	0x0000

4.3.1 EPOS4/IDX: Configuration Tasks

EPOS STUDIO’S “STARTUP WIZARD”

1) Configure motor, sensor, and system data.



Motor

Enter motor type and characteristics (consult maxon catalog).

Motor type: maxon EC motor (BLDC)

Nominal current: 5000 mA

Torque constant: ☑ Identify during mechanical system identification

0.000 mNm/A

Thermal time constant winding: 4.0 s

Number of pole pairs: 1

Figure 4-43 EtherCAT integration – zub’s MACS Multi-Axis EtherCAT Masters | EPOS Studio “Startup Wizard”

2) Conclude by pressing the ¤Finish¤ button to save all configured data in the EPOS4/IDX.

EPOS STUDIO’S “REGULATION TUNING“

1) Tune the current control (with or without load).
Current control parameters do not depend on the load. Therefore, current control tuning can be processed without a load.

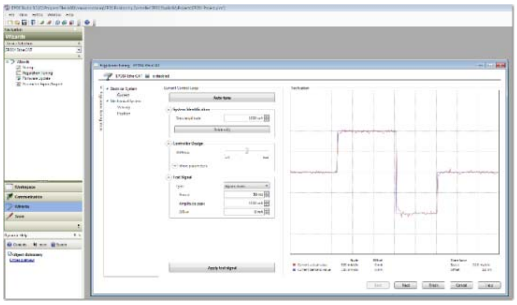


Figure 4-44 EtherCAT integration – zub’s MACS Multi-Axis EtherCAT Masters | EPOS Studio “Regulation Tuning“ 1

2) Tune the position control loop with load.
Position control parameters depend on the load, particularly if no gear box is present. Therefore, position control tuning should be processed with the attached load.

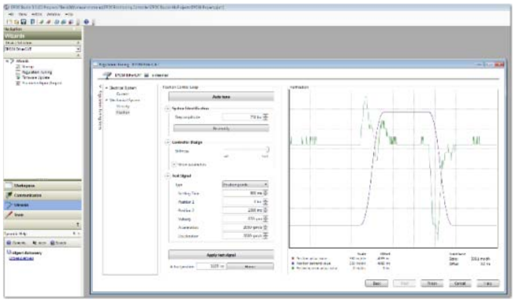


Figure 4-45 EtherCAT integration – zub’s MACS Multi-Axis EtherCAT Masters | EPOS Studio “Regulation Tuning” 2

3) Velocity control does not apply in CSP mode. Thus, tuning of the velocity control loop is not mandatory but recommended anyway.
4) Find detailed information on control tuning in the application note ➔chapter "2 Controller Architecture" on page 2-11.
5) Conclude by pressing the ¤Finish¤ buttonto save all configured data in the EPOS4/IDX controller.

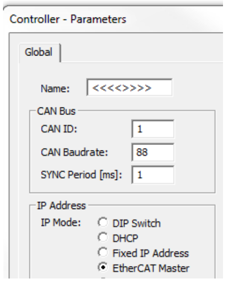
4.3.2 MasterMACS / MACS5: Setup Tasks

MACS5 IP MODE CONFIGURATION

With a MACS5 in use, configuration of the IP mode as "EtherCAT Master" is necessary.

- Menu item: Controller
- Parameters
- Global / Axis
- Mark the radio button "EtherCAT Master"

Note
By default, the MasterMACS is configured as "EtherCAT Master".



Controller - Parameters

Global

Name: <<<<>>>>

CAN Bus
CAN ID: 1
CAN Baudrate: 88
SYNC Period [ms]: 1

IP Address
IP Mode: ○ DIP Switch
○ DHCP
○ Fixed IP Address
● EtherCAT Master

Figure 4-46 EtherCAT integration – zub's MACS Multi-Axis EtherCAT Masters | MACS5 IP Mode Configuration

MACS & EPOS4/IDX: CONFIGURATION OF CONSISTENT PARAMETERS

1) Configure the encoder resolution of the MACS and EPOS4/IDX with matching values.
2) Configure the "Following error window" of the MACS parameter "POSERR" with a higher value than the EPOS4/IDX object 0x6065-00.
3) Ensure that configuration of the MACS' parameter "EtherCAT SYNC period" (default: 1000 µs) and the EPOS4/IDX's "Interpolation time period value" object 0x60C2-01 correspond to each other.
   - We recommend to keep the MACS' default setting of 1000 µs and to configure the **EPOS4/IDX's object 0x60C2-01 to a value of 1 ms**.
   - If the settings of this MACS and EPOS4/IDX parameters do not match, electrical noise and/or malfunction in position control may occur.

DEACTIVATION OF MACS POSITION CONTROL

The MACS position control is not required in case of EPOS4/IDX's "Homing" or "CSP" mode. Therefore, position control can be deactivated by setting the MACS position control parameters (P, I, D) to "0" (zero), compare code extract "SetupAxisParam".

LOCATION OF INCLUDE FILES (*.MI)

If there are include files (*.mi) in use by a program they must be located in the same directory as the application program (*.m). Otherwise, the include file will not be found by the APOSS compiler.

HOMING PROCESSED BY EPOS4/IDX

It is recommended to use the “Homing” mode of the EPOS4/IDX before activating CSP and the MACS controller to process path planning and generation of cyclic demand position updates.

- Using EPOS4/IDX's “Homing” mode offers the possibility to use all homing methods of the EPOS4/IDX.
- Refer to separate document ➔«IDX Firmware Specification»; chapter “Homing Mode (HMM)” to learn more about the different homing methods.

4.3.3 MACS: Configuration of EtherCAT Communication & Start-up Procedure

The initial configuration of communication and start-up procedure of the MACS can be split into different consecutive steps (functions) which can be integrated in an include file (*.mi) for usage by multiple application programs.

The typical flow of function calls will look as follows:

1) Initial setup tasks
   a) Setup some base axis parameters of the MACS
      ➔source code of function “**SetupAxisParam()**”
   b) Setup CSP: Configure PDOs, SYNC period, and activate CSP mode
      ➔source code of function “**SetupDriveCommandingCSP()**”
2) Start EtherCAT communication
   a) Configure and intitiate EtherCAT communication
      ➔source code of function “**EtherCATMasterStart()**”
3) MACS system setup tasks
   a) Setup the bus module
      ➔source code of function “**SetupBusModule()**”
   b) Setup virtual amplifier
      ➔source code of function “**SetupVirtAmp()**”
   c) Setup virtual counter inputs
      ➔source code of function “**SetupVirtCntInp()**”

The later description provides code samples of all above mentioned functions as well as a simple application program. The source code can be copied from the document and pasted into an include file (such as “MACS-EPOS4/IDX-Config.mi”) which then initially can be called up by the application program (such as “MACS-EPOS4/IDX-Test.m”).

INITIAL SETUP TASKS: SETUPAXISPARAM(), SETUPPDOMAPPING()

```
#include "sysdef.mi"  // Standardized include file by zub
#pragma NOIMPLICIT

// Setup MACS axis parameters
long SetupAxisParam(long axis, long EncCpt, long MaxRpm, long MaxAcc)
{
// Ensure that this corresponds to the encoder and EPOS4/IDX configuration!
set posencqc x(Axis) (EncCpt*4)  // Set enc. resolution per turn [inc]
set posencrev x(Axis) 1          // Default: 1
set feeddist x(Axis) (EncCpt*4)  // Set feed resolution (= output) per turn [inc]
set feedrev x(Axis) 1            // Default: 1

set velmax  x(Axis) MaxRpm       // Set max. velocity [rpm]
set rampmin x(Axis) MaxAcc       // Set max. acceleration [ms] (0- > MaxRpm)

set kprop x(Axis) 0        // Disable P-Gain of PID-Controller
set kder  x(Axis) 0        // Disable D-Gain of PID-Controller
set kint  x(Axis) 0        // Disable I-Gain of PID-Controller

set poserr x(Axis) 200000  // Set max following error [inc]
// EPOS4/IDX processes position control,
// => Configuration of EPOS4/IDX’s “Following Error Window” (0x6065-00) [inc]
}

// Setup CSP: Configure PDOs, SYNC period, activate OP mode
long SetupDriveCommandingCSP(long DriveId)
// DriveId is 1000000 plus the EtherCAT slave position in the bus
{
sdowriten DriveId 0x1C12 0x00 1 0x00     // Disable entry 0x1C12
sdowriten DriveId 0x1C13 0x00 1 0x00     // Disable entry 0x1C13

sdowriten DriveId 0x1A00 0 1 0           // Clear PDO 0x1A00 entries
sdowriten DriveId 0x1A00 1 4 0x60410010  // PDO 0x1A00 entry: Status
sdowriten DriveId 0x1A00 2 4 0x60640020  // PDO 0x1B00 entry: Actual position
sdowriten DriveId 0x1A00 0 1 2           // PDO 0x1A00 entry: Number of entries

sdowriten DriveId 0x1A01 0 1 0           // Clear PDO 0x1A01 entries
sdowriten DriveId 0x1A02 0 1 0           // clear PDO 0x1A02 entries
sdowriten DriveId 0x1A03 0 1 0           // clear PDO 0x1A03 entries

sdowriten DriveId 0x1600 0 1 0           // Clear PDO 0x1600 entries
sdowriten DriveId 0x1600 1 4 0x60400010  // PDO 0x1600 entry: Cmd
sdowriten DriveId 0x1600 2 4 0x607A0020  // PDO 0x1600 entry: Position set point
sdowriten DriveId 0x1600 0 1 2           // PDO 0x1600 entry: Number of entries

sdowriten DriveId 0x1601 0 1 0           // Clear PDO 0x1601 entries
sdowriten DriveId 0x1602 0 1 0           // Clear PDO 0x1602 entries
sdowriten DriveId 0x1603 0 1 0           // Clear PDO 0x1603 entries

sdowriten DriveId 0x1C12 1 2 0x1600      // PDO 0x1C12:01 index
sdowriten DriveId 0x1C12 0 1 1           // PDO 0x1C12 count

sdowriten DriveId 0x1C13 1 2 0x1A00      // PDO 0x1C13:01 index
sdowriten DriveId 0x1C13 0 1 1           // PDO 0x1C13 count

// Ensure that the EtherCAT SYNC and EPOS4/IDX interpolation time period correspond
// It is recommended to use 1 ms (which is also the default setting value of the MACS)
ecatmasterconfig 0x1000 0 1000    // MACS:  EtherCAT master SYNC: 1000 us
sdowriten DriveId 0x60C2 1 0 1    // EPOS4/IDX: Interpolation time period value: 1ms

// Set EPOS4/IDX operating mode: CSP
sdowriten DriveId 0x6060 0 1 8   // CSP = mode 8
}
```

START ETHERCAT COMMUNICATION: ETHERCATMASTERSTART()

```
// Starting EtherCAT
long EtherCATMasterStart()
{
ecatmastercommand 0x1000 2    //  Map Input and Output buffers (go to safeop)
ecatmastercommand 0x1000 3    //  Request & wait OP state for all slaves
}
```

MACS SYSTEM SETUP TASKS: SETUPBUSMODULE(), SETUPVIRTAMP(), SETUPVIRTCNTINP()

```
// Setup bus modules
long SetupBusModule(long Axis, long PdoNumber)
{
long busmod
busmod = Axis-1

BUSMOD_PARAM(busmod, BUSMOD_MODE) = 0      // delete existing bus module
BUSMOD_PARAM(busmod, BUSMOD_BUSTYPE) = 2   // EtherCAT Master

BUSMOD_PARAM(busmod, BUSMOD_PISRC_INPUT1) =
VIRTAMP_PROCESS_SRCINDEX(busmod,PO_VIRTAMP_CMDWORD)   // CMD Word
BUSMOD_PARAM(busmod, BUSMOD_PISRC_INPUT2) =
VIRTAMP_PROCESS_SRCINDEX(busmod,PO_VIRTAMP_REFPOS)    // Position setpoint

BUSMOD_PARAM(busmod, BUSMOD_TXMAP_INPUT1) = PdoNumber*0x01000000 + 2*0x00010000 + 0
// pdo; length in bytes; bytes offset of control word
BUSMOD_PARAM(busmod, BUSMOD_TXMAP_INPUT2) = PdoNumber*0x01000000 + 4*0x00010000 + 2
// pdo; length in bytes; bytes offset of target position

BUSMOD_PARAM(busmod, BUSMOD_RXMAP_POVALUE1) = PdoNumber*0x01000000 + 2*0x00010000 + 0
// pdo ; length in bytes; bytes offset of status word
BUSMOD_PARAM(busmod, BUSMOD_RXMAP_POVALUE2) = PdoNumber*0x01000000 + 4*0x00010000 + 2
// pdo ; length in bytes; bytes offset of position actual value

BUSMOD_PARAM(busmod, BUSMOD_MODE) = 2

// Start bus module
ecatmasterconfig (Axis) busmod 0
}

// Setup Virtual Amplifier
long SetupVirtAmp(long Axis)
{
long modno
modno = Axis-1
// virtual amplifiers have a fixed connection to axes number, axe 1 use amp 0
VIRTAMP_PARAM(modno,VIRTAMP_PISRC_CMDWORD)= AXE_PROCESS_SRCINDEX(modno,REG_CNTRLWORD)
VIRTAMP_PARAM(modno,VIRTAMP_PISRC_REFPOS) = AXE_PROCESS_SRCINDEX(modno,REG_COMPOS)
VIRTAMP_PARAM(modno,VIRTAMP_PISRC_REFVEL) = AXE_PROCESS_SRCINDEX(modno,REG_REFERENCE)
VIRTAMP_PARAM(modno,VIRTAMP_PISRC_REFACC) = AXE_PROCESS_SRCINDEX(modno,PID_FFACCPART)
VIRTAMP_PARAM(modno,VIRTAMP_PISRC_STATUS) =
BUSMOD_PROCESS_SRCINDEX(modno,PO_BUSMOD_VALUE1)
VIRTAMP_PARAM(modno,VIRTAMP_CNTRLW_PWROFF)    = 0x06
VIRTAMP_PARAM(modno,VIRTAMP_CNTRLW_PWRONDIS)  = 0x06
VIRTAMP_PARAM(modno,VIRTAMP_CNTRLW_PWRONENP)  = 0x0F
VIRTAMP_PARAM(modno,VIRTAMP_CNTRLW_PWRONENN)  = 0x0F
VIRTAMP_PARAM(modno,VIRTAMP_CNTRLW_QUICKSTOP) = 0x02
VIRTAMP_PARAM(modno,VIRTAMP_CNTRLW_RESET)     = 0x80
VIRTAMP_PARAM(modno,VIRTAMP_STOPDELAY)        = 0x0
VIRTAMP_PARAM(modno,VIRTAMP_ERROR_BITMASK)    = 0x0008
VIRTAMP_PARAM(modno,VIRTAMP_ERROR_POLARITY)   = 1
VIRTAMP_PARAM(modno,VIRTAMP_MODE) = 1
// has to be the last one because it activates all
}

// Setup Virtual Counter inputs
long SetupVirtCntInp()
{
VIRTCOUNTIN_PARAM(0,VIRTCNTIN_PISRC_COUNTER)=
BUSMOD_PROCESS_SRCINDEX(0,PO_BUSMOD_VALUE2)
VIRTCOUNTIN_PARAM(0,VIRTCNTIN_MODE) = 3    // source is absolute and is taken as it is
}
```

4.3.4 Simple Application Program

The include file “MACS-EPOS-Config.mi“ holds the functions and source code listed on the last pages. This include file must be part of the application program.

```
// Main-Program (.m File)

// Include file providing the required setup functions
#include "MACS-EPOS4/IDX-Config.mi"
// Remark: Include files have to be located in the same directory like the *.m file

long Axis, PdoNumber, DriveId, EncCpt, MaxRpm, MaxAcc

Axis = 1
PdoNumber = 1
DriveId = 1000001  // = 1000000 plus the EtherCAT slave position in the bus
EncCpt = 500       // Encoder resolution [cpt]
MaxRpm = 3000      // Max. speed [rpm]
MaxAcc = 100       // Max. acceleration [ms]: 0 -> MaxRpm

errclr  // Clear all error states, if there are any present

ecatmastercommand 0x1000 0    // Disable master
ecatmastercommand 0x1000 1    // Start master

// Initial setup tasks
SetupAxisParam(Axis, EncCpt, MaxRpm, MaxAcc)
SetupDriveCommandingCSP(DriveId)

// Start EtherCAT communication
EtherCATMasterStart()

// MACS system setup tasks
SetupBusModule(Axis,PdoNumber)
SetupVirtAmp(Axis)
SetupVirtCntInp()

// Clear error states, if there are any present
Errclr              // Clear MACS error states
Amperrclr x(Axis)   // Clear EPOS4/IDX error states

delay(20)           // Wait 20 ms
motor on x(Axis)    // Enable the power stage

// Start simple cyclic movement of the motor
while(1) do
   vel x(Axis) 50     // Use 50% of MACS max. velocity
   acc x(Axis) 30     // Use 30% of MACS max. acceleration
   dec x(Axis) 30     // Use 30% of MACS max. deceleration

   // Move absolute 10 turns
   posa x(Axis) (get posencqc)*10
   delay 500        // Wait 500 ms
   posa x(Axis) 0   // Move absolute to position 0
   delay 200        // Wait 200 ms
endwhile
```

5 DEVICE PROGRAMMING

CONTENT

5.1 In Brief

A wide variety of operating modes permit flexible configuration of the drive system by using positioning, velocity, and current regulation. The built-in CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

Some parameters of the IDX are defined during the course of factory configuration. Therefore, the corresponding objects can no longer be changed during operation.

OBJECTIVE

The present application note explains typical commanding sequences for different operating modes based on writing/reading commands to access the Object Dictionary. For detailed information…

- on the objects itself ➔separate document «IDX Firmware Specification» (subsequently referred to as “FwSpec”),
- on the command structure ➔separate document «IDX Communication Guide» and ➔«EPOS Studio»; tool “Command Analyzer”.

SCOPE

Hardware	Order #	Firmware version	Reference
EPOS4/IDX	–	0160h	IDX Firmware Specification
IDX 56	various	0160h or higher	

Table 5-24 Device programming | Covered hardware and required documents

TOOLS

Tools	Description
Software	«EPOS Studio» Version 3.6 or higher

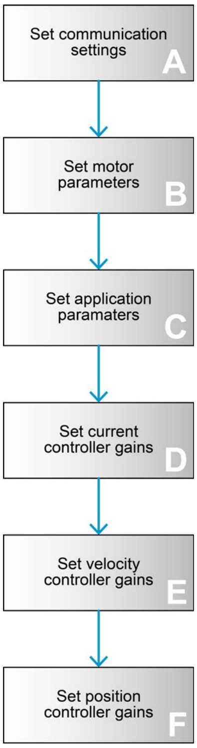
Table 5-25 Device programming | Recommended tools

5.2 First Step

Before the motor will be activated, motor parameters, position sensor parameters, and controller gains must be set. For detailed description ➔FwSpec.

Notes

- *For detailed information on the command structure ➔«EPOS Studio» (command analyzer).*
- *In the later course of the present chapter, the stated units ([inc], [rpm], [rpm/s]) can vary depending on the system units configured by the objects 0x60A8, 0x60A9, and 0x60AA. All stated units correspond to their respective default configuration.*


Set communication settings
A
Set motor parameters
B
Set application paramaters
C
Set current controller gains
D
Set velocity controller gains
E
Set position controller gains
F

#	Object name	Object	User value [default value]
A	Node-ID	0x2000-00	User-specific [1]; typically configured by DIP switches
	CAN bit rate	0x2001-00	User-specific [0] (= 1 Mbit/s)
B	Output current limit	0x3001-02	User-specific [mA]
	Max motor speed	0x6080-00	User-specific: Motor or mechanical limits [rpm]
	Max gear input speed	0x3003-03	User-specific: Gear or mechanical limits [rpm]
C	Software position limit	0x607D-xx	User-specific [0 inc]
D	Current control parameter set: • Current controller PI gains	0x30A0-xx	Motor-specific and load-specific: Determine optimal parameter using “Regulation Tuning” in «EPOS Studio».
E	Velocity control parameter set: • Velocity controller PI gains • Velocity controller FF gains	0x30A2-xx	
	Velocity observer parameter set	0x30A3-xx	
F	Position control parameter set: • Position controller PID gains • Position controller FF gains	0x30A1-xx	

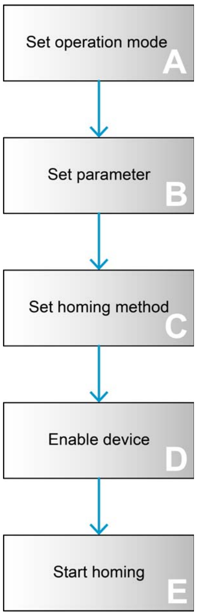
Table 5-26 Device programming | First step

5.3 Homing Mode (HMM)

Note
For details on Controlword bits (0x6040) ➔FwSpec.

START HOMING

The axis references to an absolute position using the selected homing method.



Set operation mode A

Set parameter B

Set homing method C

Enable device D

Start homing E

#	Object name	Object	User value [default value]
A	Modes of operation	0x6060-00	0x06 (Homing Mode)
B	Following error window	0x6065-00	User-specific [2'000 inc]
	Home offset move distance	0x30B1-00	User-specific [0 inc]
	Quick stop deceleration	0x6085-00	User-specific [10'000 rpm/s]
	Speed for switch search	0x6099-01	User-specific [100 rpm]
	Speed for zero search	0x6099-02	User-specific [10 rpm]
	Homing acceleration	0x609A-00	User-specific [1'000 rpm/s]
	Home position	0x30B0-00	User-specific [0 inc]
C	Homing method	0x6098-00	Select homing method (for details ➔FwSpec)
D	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
E	Controlword (Start homing)	0x6040-00	0x001F

Table 5-27 Device programming | Homing Mode (Start)

READ STATUS

Read statusword

Object name	Object	User value [default value]
Statusword (Target reached / Homing attained)	0x6041-00	Homing procedure is completed successfully if bit 12 (="Homing attained") is set to "1".

Table 5-28 Device programming | Homing Mode (Read)

STOP HOMING

Stop homing

Object name	Object	User value [default value]
Controlword (Switch on & Enable)	0x6040-00	0x000F
or		
Controlword (Halt homing)	0x6040-00	0x011F
or		
Controlword (Quick stop)	0x6040-00	0x000B

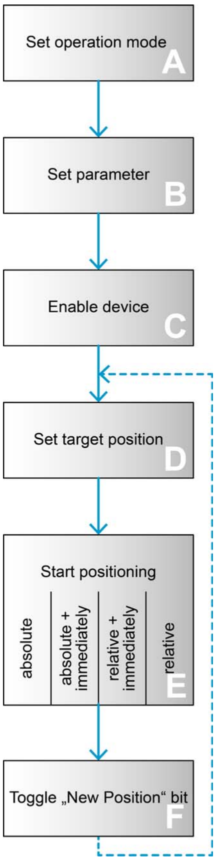
Table 5-29 Device programming | Homing Mode (Stop)

5.4 Profile Position Mode (PPM)

Note
For details on Controlword bits (0x6040) and Statusword bits (0x6041) ➔FwSpec.

SET POSITION

The axis moves to an absolute or relative position using a motion profile.



#	Object name	Object	User value [default value]
A	Modes of operation	0x6060-00	0x01 (Profile Position Mode)
B	Following error window	0x6065-00	User-specific [2'000 inc]
	Profile velocity	0x6081-00	Desired velocity [1'000 rpm]
	Profile acceleration	0x6083-00	User-specific [10'000 rpm/s]
	Profile deceleration	0x6084-00	User-specific [10'000 rpm/s]
	Quick stop deceleration	0x6085-00	User-specific [10'000 rpm/s]
	Motion profile type	0x6086-00	User-specific [0]
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Target position	0x607A-00	Desired position [inc]
E	Controlword (absolute position)	0x6040-00	0x001F
		or	
	Controlword (absolute position, start immediately)	0x6040-00	0x003F
		or	
	Controlword (relative position, start immediately)	0x6040-00	0x007F
		or	
	Controlword (relative position)	0x6040-00	0x005F
F	Controlword (New Position)	0x6040-00	0x000F (toggle “New Position”)

Table 5-30 Device programming | Profile Position Mode (Set)

READ STATUS

Object name	Object	User value [default value]
Statusword (Target reached)	0x6041-00	The axis has reached the target position if bit 10 (=”Target reached”) is set to “1” and bit 8 (=”Halt”) of the Controlword (0x6040) was not activated (for details ➔FwSpec).

Table 5-31 Device programming | Profile Position Mode (Read)

STOP POSITIONING

Stop positioning

Object name	Object	User value [default value]
Controlword (Halt profile position mode)	0x6040-00	0x010F
or		
Controlword (Quick stop)	0x6040-00	0x000B

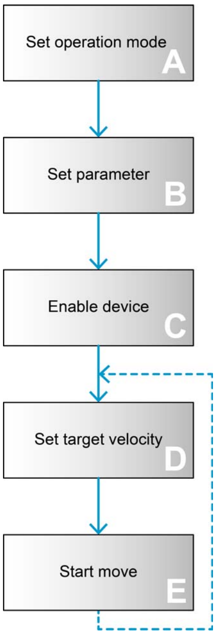
Table 5-32 Device programming | Profile Position Mode (Stop)

5.5 Profile Velocity Mode (PVM)

Note
For details on Controlword bits (0x6040) and Statusword bits (0x6041) ➔FwSpec.

START VELOCITY

Motor shaft rotates with a certain speed with velocity profile.



#	Object name	Object	User value [default value]
A	Modes of operation	0x6060-00	0x03 (Profile Velocity Mode)
B	Profile acceleration	0x6083-00	User-specific [10'000 rpm/s]
	Profile deceleration	0x6084-00	User-specific [10'000 rpm/s]
	Quick stop deceleration	0x6085-00	User-specific [10'000 rpm/s]
	Motion profile type	0x6086-00	User-specific [0]
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Target velocity	0x60FF-00	Desired velocity [rpm]
E	Controlword	0x6040-00	0x000F

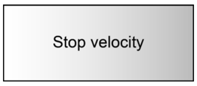
Table 5-33 Device programming | Profile Velocity Mode (Start)

READ STATUS

Object name	Object	User value [default value]
Statusword (Target velocity reached)	0x6041-00	Target velocity is reached if bit 10 is set (for details ➔FwSpec).

Table 5-34 Device programming | Profile Velocity Mode (Read)

STOP VELOCITY



Object name	Object	User value [default value]
Controlword (Halt profile velocity mode)	0x6040-00	0x010F
or		
Controlword (Quick stop)	0x6040-00	0x000B

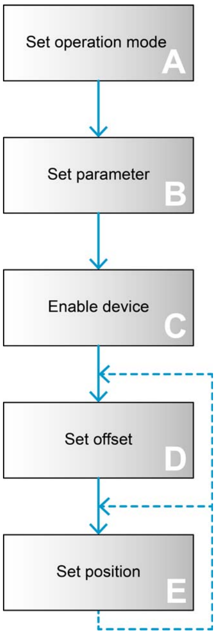
Table 5-35 Device programming | Profile Velocity Mode (Stop)

5.6 Cyclic Synchronous Position Mode (CSP)

Note
For details on Controlword bits (0x6040) ➔FwSpec.

SET POSITION

The axis moves to an absolute or relative position.



#	Object name	Object	User value [default value]
A	Modes of operation	0x6060-00	0x08 (Cyclic Synchronous Position Mode)
B	Max motor speed	0x6080-00	User-specific; motor or mechanical limits [rpm]
	Max gear input speed	0x3003-03	User-specific: Gear or mechanical limits [rpm]
	Profile deceleration (1)	0x6084-00	User-specific [10'000 rpm/s]
	Quick-stop deceleration (1)	0x6085-00	User-specific [10'000 rpm/s]
	Interpolation time period	0x60C2-xx	Master's SYNC period (Cycle ticks) [1 ms]
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Torque offset	0x60B2-00	Torque offset [per thousand of "Motor rated torque"; 0x6076]
	Position offset	0x60B0-00	Position offset [0 inc]
E	Target position	0x607A-00	Desired position [inc]

(1) deceleration values are used for stopping only, they are not used for normal operation

Table 5-36 Device programming | Cyclic Synchronous Position Mode (Set)

STOP MOTION

Object name	Object	User value [default value]
Controlword (Quick stop)	0x6040-00	0x000B

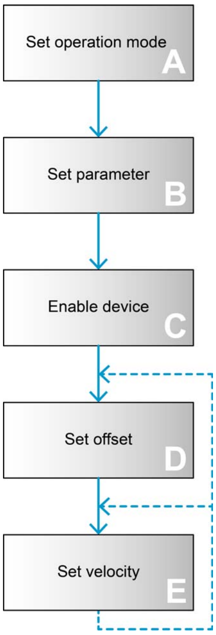
Table 5-37 Device programming | Cyclic Synchronous Position Mode (Stop)

5.7 Cyclic Synchronous Velocity Mode (CSV)

Note
For details on Controlword bits (0x6040) ➔FwSpec.

SET VELOCITY

The axis moves with the commanded velocity.

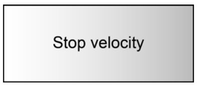


#	Object name	Object	User value [default value]
A	Modes of operation	0x6060-00	0x09 (Cyclic Synchronous Velocity Mode)
B	Max motor speed	0x6080-00	User-specific; motor or mechanical limits [rpm]
	Max gear input speed	0x3003-03	User-specific: Gear or mechanical limits [rpm]
	Profile deceleration (1)	0x6084-00	User-specific [10'000 rpm/s]
	Quick-stop deceleration (1)	0x6085-00	User-specific [10'000 rpm/s]
	Interpolation time period	0x60C2-xx	Master’s SYNC period (Cycle ticks) [1 ms]
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Velocity offset	0x60B1-00	Velocity offset [rpm]
E	Target velocity	0x60FF-00	Desired velocity [rpm]

(1) deceleration values are used for stopping only, they are not used for normal operation

Table 5-38 Device programming | Cyclic Synchronous Velocity Mode (Set)

STOP MOTION



Object name	Object	User value [default value]
Target velocity	0x60FF-00	0x0000
	or	
Controlword (Quick stop)	0x6040-00	0x000B

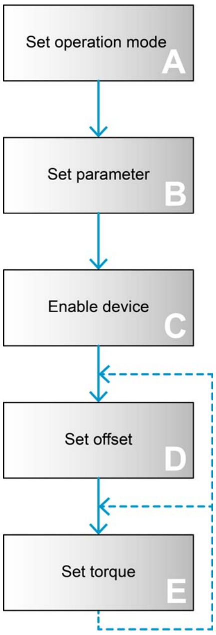
Table 5-39 Device programming | Cyclic Synchronous Velocity Mode (Stop)

5.8 Cyclic Synchronous Torque Mode (CST)

Note
For details on Controlword bits (0x6040) ➔FwSpec.

SET TORQUE

Applies a certain torque (that is: torque = current x torque constant) to the motor winding.



#	Object name	Object	User value [default value]
A	Modes of operation	0x6060-00	0x0A (Cyclic Synchronous Torque Mode)
B	Max motor speed	0x6080-00	User-specific; motor or mechanical limits [rpm]
	Max gear input speed	0x3003-03	User-specific: Gear or mechanical limits [rpm]
	Profile deceleration (1)	0x6084-00	User-specific [10'000 rpm/s]
	Quick-stop deceleration (1)	0x6085-00	User-specific [10'000 rpm/s]
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Torque offset	0x60B2-00	Torque offset [per thousand of “Motor rated torque”; 0x6076]
E	Target torque	0x6071-00	Desired torque [per thousand of “Motor rated torque”; 0x6076]

(1) deceleration values are used for stopping only, they are not used for normal operation

Table 5-40 Device programming | Cyclic Synchronous Torque Mode (Set)

STOP MOTION

Object name	Object	User value [default value]
Target torque	0x6071-00	0x0000
or		
Controlword (Quick stop)	0x6040-00	0x000B

Table 5-41 Device programming | Cyclic Synchronous Torque Mode (Stop)

5.9 State Machine

CLEAR FAULT

Resetting “Fault” condition sends the Controlword with value 0x0080.

Object name	Object	User value [default value]
Controlword (Fault reset)	0x6040-00	0x0000;0x0080

Table 5-42 Device programming | State machine (clear fault)

SEND NMT SERVICE

Object name	Object	User value [default value]
Node ID (Unique Node ID or “0” (zero) for all nodes)		
Command specifier:	0x01 0x02 0x80 0x81 0x82	Start remote node Stop remote node Enter pre-operational Reset node Reset communication

Table 5-43 Device programming | State machine (send NMT service)

5.10 Motion Info

GET MOVEMENT STATE

Object name	Object	User value [default value]
Read statusword	0x6041-00	The bits are partly depending on the operating mode (for details ➔FwSpec)

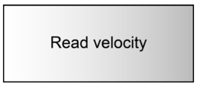
Table 5-44 Device programming | Motion info (Get movement state)

READ POSITION

Object name	Object	User value [default value]
Read position	0x6064-00	Position actual value [inc]

Table 5-45 Device programming | Motion info (Read position)

READ VELOCITY



Object name	Object	User value [default value]
Read velocity actual value averaged	0x30D3-01	Velocity actual value averaged [rpm]

Table 5-46 Device programming | Motion info (Read velocity)

READ TORQUE

Object name	Object	User value [default value]
Read torque actual value	0x6077-00	Torque actual value [per thousand of “Motor rated torque”; 0x6076]

Table 5-47 Device programming | Motion info (Read torque)

5.11 Utilities

STORE ALL PARAMETERS

Saves all parameters.

Object name	Object	User value [default value]
Save all parameters	0x1010-01	0x65766173 “save”

Table 5-48 Device programming | Utilities (Store all parameters)

RESTORE ALL DEFAULT PARAMETERS

Restores all parameters to factory settings.

Object name	Object	User value [default value]
Restore all default parameters	0x1011-01	0x64616F6C “load”

Table 5-49 Device programming | Utilities (Restore all default parameters)

LIST OF FIGURES

LIST OF TABLES

INDEX

maxon

EtherCAT

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

CCMC | IDX Application Notes | Edition 2019-11 | DocID rel9103

maxon motor ag
Brünigstrasse 220 +41 41 666 15 00
CH-6072 Sachseln www.maxongroup.com