EPOS2

Positioning Controllers

Application Notes

epos.maxonmotor.com

Document ID: rel7167

PLEASE READ THIS FIRST

The present document represents a compilation of (hopefully) helpful “Good-to-Knows” that might come in handy in your daily work with EPOS2 Positioning Controllers.

The individual chapters cover particular cases or scenarios and are intended to give you a hand for efficient setup and parameterization of your system.

We strongly stress the following facts:

- *The present document does not replace any other documentation covering the basic installation and/ or parameterization described therein!*
- *Also, any aspect in regard to health and safety, as well as to secure and safe operation are not covered in the present document – it is intended and must be understood as complimenting addition to those documents!*

TABLE OF CONTENTS

••page intentionally left blank••

1 About this Document

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 motor, 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. Please find below an overview on the documentation hierarchy and the interrelationship of its individual parts:

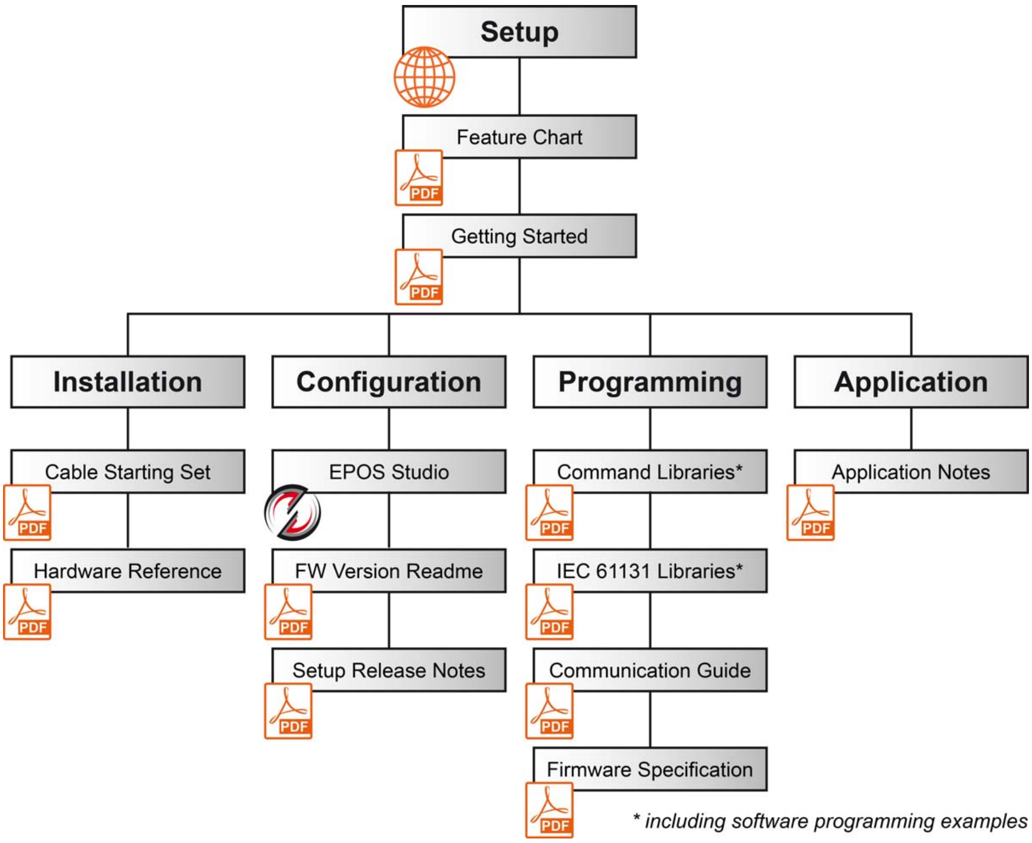


Figure 1-1 Documentation Structure

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.

1.3 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.)
➔	denotes “see”, “see also”, “take note of” or “go to”

Table 1-1 Notations used in this Document

1.4 Symbols and Signs

1.4.1 Safety Alerts

Take note of when and why the alerts will be used and what the consequences are if you should fail to observe them!

Safety alerts are composed of…

- a signal word,
- a description of type and/or source of the danger,
- the consequence if the alert is being ignored, and
- explanations on how to avoid the hazard.

Following types will be used:

1) **DANGER**
Indicates an **imminently hazardous situation**. If not avoided, the situation will result in death or serious injury.
2) **WARNING**
Indicates a **potentially hazardous situation**. If not avoided, the situation **can** result in death or serious injury.
3) **CAUTION**
Indicates a **probable hazardous situation** and is also used to alert against unsafe practices. If not avoided, the situation **may** result in minor or moderate injury.

Example:

DANGER

High Voltage and/or Electrical Shock
Touching live wires causes death or serious injuries!
- *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 with your name!*

1.4.2 Prohibited Actions and Mandatory Actions

The signs define prohibitive actions. So, you **must not**!

Examples:

Do not touch!

Do not operate!

The signs point out actions to avoid a hazard. So, you **must**!

Examples:

Unplug!

Tag before work!

1.4.3 Informatory 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.

Reference
Refers to particular information provided by other parties.

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
CANopen® CiA®	© CiA CAN in Automation e.V, DE-Nuremberg
Excel	© Microsoft Corporation, USA-Redmond, WA
Micro-Fit™ Mini-Fit Jr.™	© Molex, USA-Lisle, IL
Pentium®	© Intel Corporation, USA-Santa Clara, CA
Windows®	© Microsoft Corporation, USA-Redmond, WA

Table 1-2 Brand Names and Trademark Owners

1.6 Sources for additional Information

Find the latest edition of additional documentation and software also on the Internet:
➔www.maxonmotor.com

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

#	Reference
[1]	CiA 301 Communication Profile for Industrial Systems www.can-cia.org
[2]	CiA 402 Device Profile for Drives and Motion Control www.can-cia.org
[3]	CiA 305 Layer Setting Services (LSS) and Protocols www.can-cia.org
[4]	CiA 306 Electronic Data Sheet Specification www.can-cia.org
[5]	Konrad Etschberger: Controller Area Network ISBN 3-446-21776-2
[6]	maxon motor: EPOS2 Communication Guide EPOS Positioning Controller DVD or www.maxonmotor.com
[7]	Dr. Urs Kafader: The selection of high-precision microdrives ISBN 978-3-9520143-6-3 Also availably from the “maxon academy” www.maxonmotor.com

Table 1-3 Sources for additional Information

1.7 System Units

Unit Dimension	Definition
Position units	steps (quadcounts = 4 x Encoder Counts / Revolution)
Velocity units	rpm (Revolutions per Minute)
Acceleration units	rpm/s (Velocity Unit / Second)

Table 1-4 Default Unit Dimensions

1.8 Copyright

maxon motor ag
Brünigstrasse 220
P.O.Box 263
CH-6072 Sachseln
Switzerland

Phone +41 41 666 15 00
Fax +41 41 666 16 50

www.maxonmotor.com

••page intentionally left blank••

2 Digital Inputs and Outputs

2.1 In Brief

Drive systems typically require inputs and outputs – “Home Switch”, Positive/Negative Limit Switches” and “Brake Output” with sufficient current, just to mention a few.

The inputs and outputs can easily be configured using the «Configuration Wizard» and may be changed online via CANopen or serial bus.

2.1.1 Objective

The present Application Note explains the functionality of digital inputs and outputs and features “in practice examples” suitable for daily use.

Contents

2.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	Cable Starting Set Hardware Reference
EPOS2 50/5	347717	2110h or higher	Cable Starting Set Hardware Reference
EPOS2 Module 36/2	360665	2110h or higher	Hardware Reference
EPOS2 24/5	367676	2110h or higher	Cable Starting Set Hardware Reference
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	Cable Starting Set Hardware Reference

Table 2-5 Digital Inputs & Outputs – covered Hardware and required Documents

2.1.3 Tools

Tools	Description
Crimper	Molex hand crimper (63819-0000)
	Molex hand crimper (63819-0900)
Software	«EPOS Studio» Version 2.00 or higher

Table 2-6 Digital Inputs & Outputs – recommended Tools

2.2 Functionality

2.2.1 Digital Inputs

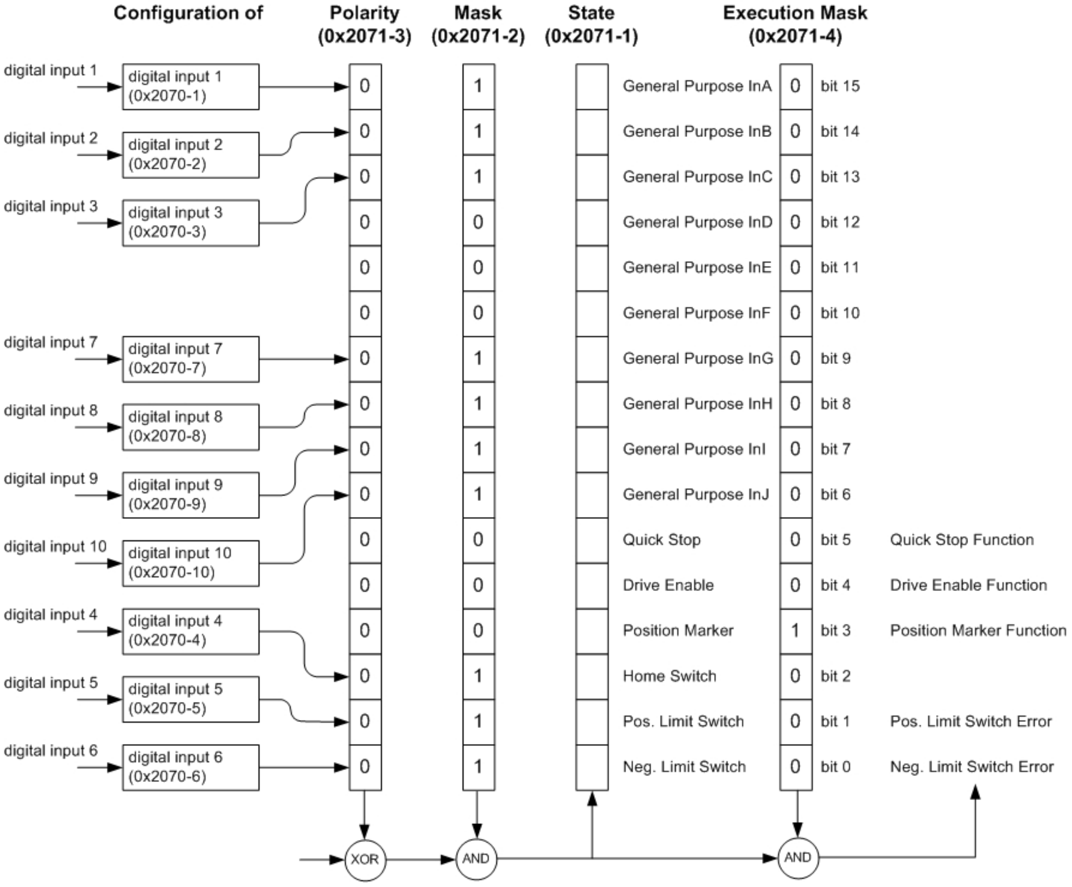


Figure 2-2 Digital Input Functionality – EPOS2 50/5 Overview (default Configuration)

Configuration Parameter

Name	Index	Sub-index	Description
Configuration of Digital Input 1 (➔Table 2-9)	0x2070	0x01	Defines functionality assigned to DigIN1.
Configuration of Digital Input 2 (➔Table 2-9)	0x2070	0x02	Defines functionality assigned to DigIN2.
Configuration of Digital Input 3 (➔Table 2-9)	0x2070	0x03	Defines functionality assigned to DigIN3.
Configuration of Digital Input 4 (➔Table 2-9)	0x2070	0x04	Defines functionality assigned to DigIN4.
Configuration of Digital Input 5 (➔Table 2-9)	0x2070	0x05	Defines functionality assigned to DigIN5. **Not available with EPOS2 Module 36/2!**
Configuration of Digital Input 6 (➔Table 2-9)	0x2070	0x06	Defines functionality assigned to DigIN6. **Not available with EPOS2 Module 36/2!**
Configuration of Digital Input 7 (➔Table 2-9)	0x2070	0x07	Defines functionality assigned to DigIN7. **Not available with EPOS2 24/5 and EPOS2 24/2!**
Configuration of Digital Input 8 (➔Table 2-9)	0x2070	0x08	Defines functionality assigned to DigIN8. **Not available with EPOS2 24/5 and EPOS2 24/2!**
Configuration of Digital Input 9 (➔Table 2-9)	0x2070	0x09	Defines functionality assigned to DigIN9. **Not available with EPOS2 Module 36/2, EPOS2 24/5 and EPOS2 24/2!**
Configuration of Digital Input 10 (➔Table 2-9)	0x2070	0x0A	Defines functionality assigned to DigIN10. **Only available with EPOS2 50/5!**
Digital Input Functionalities Mask (➔Table 2-10)	0x2071	0x02	Displayed state of Digital Input Functionalities may be filtered.
Digital Input Functionalities Polarity (➔Table 2-11)	0x2071	0x03	Polarity of Digital Input Functionalities.
Digital Input Functionalities Execution Mask (➔Table 2-10)	0x2071	0x04	Execution of Digital Input Functionalities can be inhibited.

Table 2-7 Digital Input – Configuration Parameter

Input Parameter

Name	Index	Sub-index	Description
Digital Input Functionalities State (➔Table 2-10)	0x2071	0x01	Display state of Digital Input Functionalities.

Table 2-8 Digital Input – Input Parameter

Input Configuration Values

Parameter “Configuration of Digital Input” defines bit position in “Digital Input Functionalities State”.

Value	Functionality	Description
15	General Purpose A	State can be read.
14	General Purpose B	State can be read.
13	General Purpose C	State can be read.
12	General Purpose D	State can be read.
11	General Purpose E	State can be read.
10	General Purpose F	State can be read.
9	General Purpose G	State can be read.
8	General Purpose H	State can be read.
7	General Purpose I	State can be read.
6	General Purpose J	State can be read.
5	Quick Stop	Set Quick Stop profile.
4	Device Enable	Enables/disables device.
3	Position Marker	Samples current position.
2	Home Switch	Used in some homing modes.
1	Positive Limit Switch	Generates limit error / used in some homing modes.
0	Negative Limit Switch	Generates limit error / used in some homing modes.

Table 2-9 Digital Input – Input Configuration Values

Parameter Description

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
General Purpose A	General Purpose B	General Purpose C	General Purpose D	General Purpose E	General Purpose F	General Purpose G	General Purpose H

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
General Purpose I	General Purpose J	Quick Stop	Device Enable	Position Marker	Home Switch	Pos. Limit Switch	Neg. Limit Switch

Table 2-10 Digital Input – Execution Mask Parameter

Polarity Values

Bit	0	1
associated pin	high active	low active

Table 2-11 Digital Input – Polarity Values

Note

- *“Digital Input Functionalities State” will only be displayed, if “Digital Input Functionalities Mask” is set to Enable.*
- *“Digital Input Functionalities State” enables/disables the specific function.*

2.2.2 Digital Outputs

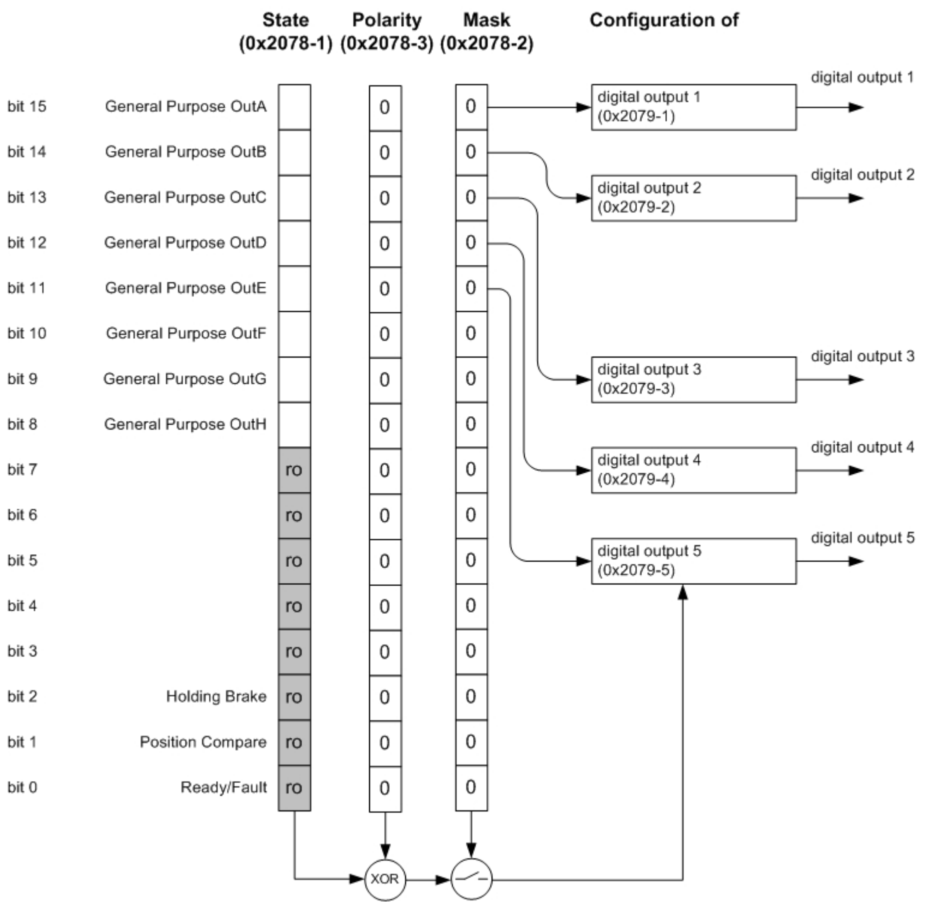


Figure 2-3 Digital Output Functionality – EPOS2 Overview (default Configuration)

Configuration Parameter

Name	Index	Sub-index	Description
Configuration of Digital Output 1 (➔Table 2-13)	0x2079	0x01	Defines functionality assigned to DigOUT1. **Not available with EPOS2 24/2!**
Configuration of Digital Output 2 (➔Table 2-13)	0x2079	0x02	Defines functionality assigned to DigOUT2. **Not available with EPOS2 24/2!**
Configuration of Digital Output 3 (➔Table 2-13)	0x2079	0x03	Defines functionality assigned to DigOUT3. **Not available with EPOS2 Module 36/2!**
Configuration of Digital Output 4 (➔Table 2-13)	0x2079	0x04	Defines functionality assigned to DigOUT4. **Not available with EPOS2 Module 36/2!**
Configuration of Digital Output 5 (➔Table 2-13)	0x2079	0x05	Defines functionality assigned to DigOUT5. **Not available with EPOS2 24/5 and EPOS2 24/2!**
Digital Output Functionalities State (➔Table 2-14)	0x2078	0x01	State of digital outputs may be set.
Digital Output Functionalities Mask (➔Table 2-14)	0x2078	0x02	Digital outputs may be filtered.
Digital Input Functionalities Polarity (➔Table 2-15)	0x2078	0x03	Change of polarity of digital output.

Table 2-12 Digital Output – Configuration Parameter

Output Configuration Values

Parameter “Configuration of Digital Output” defines bit position in “Digital Output Functionalities State”.

Value	Functionality	Description
15	General Purpose A	State can be read.
14	General Purpose B	State can be read.
13	General Purpose C	State can be read.
12	General Purpose D	State can be read.
11	General Purpose E	State can be read.
10…8	not used	–
7…3	reserved	–
2	Holding Brake	Active output = activated brake Inactive output = deactivated brake
1	Position compare	Trigger output of Position Compare.
0	Ready / Fault	Active on Device Ready / Inactive on Fault

Table 2-13 Digital Output – Output Configuration Values

Parameter Description

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10…3	Bit 2	Bit 2	Bit 0
General Purpose A	General Purpose B	General Purpose C	General Purpose D	General Purpose E	not used / reserved	Holding Brake	Position Compare	Ready / Fault

Table 2-14 Digital Output – Execution Mask Parameter

Polarity Values

Bit	0	1
associated pin	not inverted 1 → high 0 → low	inverted 0 → high 1 → low

Table 2-15 Digital Output – Polarity Values

Note
A change in “Digital Output Functionalities State” is only of effect, if “Digital Output Functionalities Mask” is set to Enable.

2.3 Connection

2.3.1 EPOS2 70/10

Signal Cable 16core (275932) – Connector J5

Head A **Head B**

Figure 2-4 Signal Cable 16core

Technical Data	
Cable cross-section	16 x 0.14 mm²
Length	3 m
Head A	Molex Micro-Fit 3.0 16 poles (430-25-1600) Molex Micro-Fit 3.0 female crimp terminals (43030-xxxx)
Head B	Cable end sleeves 0.14 mm²

Table 2-16 Signal Cable 16core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		–	IN_COM2	Common signal 2 for DigIN4…6
brown	2		–	IN_COM1	Common signal 1 for DigIN1…3
green	3		–	DigIN6	Digital input 6 “Negative Limit Switch”
yellow	4		–	DigIN5	Digital input 5 “Positive Limit Switch”
grey	5		–	DigIN4	Digital input 4 “Home Switch”
pink	6		–	DigIN3	Digital input 3 “General Purpose”
blue	7		–	DigIN2	Digital input 2 “General Purpose”
red	8		–	DigIN1	Digital input 1 “General Purpose”
black	9		–	+V Opto IN	External supply input voltage for Digital Outputs (+12…24 VDC)
violet	10		–	DigOUT4	Digital output 4 “Brake”
grey/ pink	11		–	DigOUT3	Digital output 3 “General Purpose”
red/blue	12		–	DigOUT2	Digital output 2 “General Purpose”
white/ green	13		–	DigOUT1	Digital output 1 “General Purpose”
brown/ green	14		–	DigOUT_Gnd	Digital OUT ground reference to “+V Opto IN”
white/ yellow	15		–	DigIN11	Digital input 11 “Power Stage Enable”
yellow/ brown	16		–	IN_COM3	Common signal 3 for DigIN11

Table 2-17 Signal Cable 16core – Pin Assignment EPOS2 70/10

Signal Cable 6x2core (300586) – Connector J5A

Head A **Head B**

Figure 2-5 Signal Cable 6x2core

Technical Data	
Cable cross-section	6 x 2 x 0.14 mm^2
Length	3.00 m
Head A	Molex Micro-Fit 3.0 12 poles (430-25-1200) Molex Micro-Fit 3.0 female crimp terminals (43030-xxxx)
Head B	Cable end sleeves 0.14 mm^2

Table 2-18 Signal Cable 6x2core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	*1*		*1*	*+5VOUT*	*Reference output voltage +5 V*
brown	*2*			*A_Gnd*	*Analog signal ground*
green	*3*		*2*	*AnIN2−*	*Negative analog signal input 2*
yellow	*4*			*AnIN2+*	*Positive analog signal input 2*
grey	*5*		*3*	*AnIN1−*	*Negative analog signal input 1*
pink	*6*			*AnIN1+*	*Positive analog signal input 1*
blue	7		4	D_GND	Digital signal ground
red	8			D_GND	Digital signal ground
black	9		5	DigIN8/	Digital input 8 “High Speed Command” complement or cos− input
violet	10			DigIN8	Digital input 8 “High Speed Command” or cos+ input
grey/ pink	11		6	DigIN7/	Digital input 7 “High Speed Command” complement or sin− input
red/blue	12			DigIN7	Digital input 7 “High Speed Command” or sin+ input

Table 2-19 Signal Cable 6x2core – Pin Assignment EPOS2 70/10

Signal Cable 3x2core (378173) – Connector J5B

Head A **Head B**

Figure 2-6 Signal Cable 3x2core

Technical Data	
Cable cross-section	3 x 2 x 0.14 mm², twisted pair
Length	3.00 m
Head A	Molex Micro-Fit 3.0 6 poles (430-25-0600) Molex Micro-Fit 3.0 female crimp terminals (43030-xxxx)
Head B	Cable end sleeves 0.14 mm²

Table 2-20 Signal Cable 3x2core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		1	DigIN9/	Digital input 9 “High Speed Command” complement
red	2			DigIN9	Digital input 9 “High Speed Command”
brown	3		2	DigOUT5/	Digital output 5 “High Speed Output” complement
green	4		3	$+V_{AUX}$	Auxiliary output voltage +5 VDC / 150 mA
yellow	5			D_GND	Digital signal ground
grey	6		2	DigOUT5	Digital output 5 “High Speed Output”

Table 2-21 Signal Cable 3x2core – Pin Assignment EPOS2 70/10

2.3.2 EPOS2 50/5

Signal Cable 16core (275932) – Connector J6

Head A **Head B**

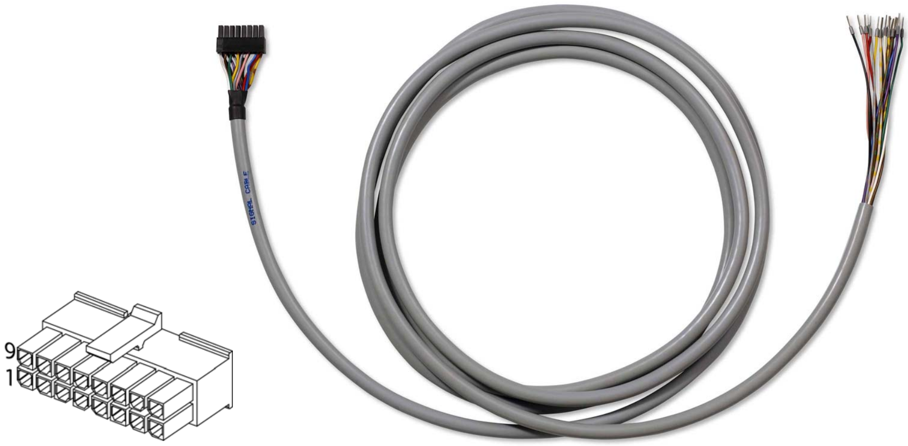


Figure 2-7 Signal Cable 16core

Technical Data	
Cable cross-section	16 x 0.14 mm²
Length	3 m
Head A	Molex Micro-Fit 3.0 16 poles (430-25-1600) Molex Micro-Fit 3.0 female crimp terminals (430-30-0010)
Head B	Cable end sleeves 0.14 mm²

Table 2-22 Signal Cable 16core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		–	IN_COM2	Common signal 2 for DigIN4…6
brown	2		–	IN_COM1	Common signal 1 for DigIN1…3
green	3		–	DigIN6	Digital Input 6 “Negative Limit Switch”
yellow	4		–	DigIN5	Digital Input 5 “Positive Limit Switch”
grey	5		–	DigIN4	Digital Input 4 “Home Switch”
pink	6		–	DigIN3	Digital Input 3 “General Purpose”
blue	7		–	DigIN2	Digital Input 2 “General Purpose”
red	8		–	DigIN1	Digital Input 1 “General Purpose”
black	9		–	+V Opto IN	External supply input voltage for Digital Outputs (+12…24 VDC)
violet	10		–	DigOUT4	Digital Output 4 “Brake / General Purpose”
grey/ pink	11		–	DigOUT3	Digital Output 3 “Brake / General Purpose”
red/blue	12		–	DigOUT2	Digital Output 2 “General Purpose”
white/ green	13		–	DigOUT1	Digital Output 1 “General Purpose”
brown/ green	14		–	DigOUT_Gnd	Digital OUT ground reference to “+V Opto IN”
white/ yellow	15		–	DigIN11	Digital Input 11 “Power Stage Enable”
yellow/ brown	16		–	IN_COM3	Common signal 3 for DigIN11

Table 2-23 Signal Cable 16core – Pin Assignment EPOS2 50/5

Signal Cable 6x2core (300586) – Connector J5

Head A **Head B**

Figure 2-8 Signal Cable 6x2core

Technical Data	
Cable cross-section	6 x 2 x 0.14 mm^2
Length	3.00 m
Head A	Molex Micro-Fit 3.0 12 poles (430-25-1200) Molex Micro-Fit 3.0 female crimp terminals (430-30-0010)
Head B	Cable end sleeves 0.14 mm^2

Table 2-24 Signal Cable 6x2core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		1	DigIN10/	Digital Input 10 “High Speed Command” complement
brown	2			DigIN10	Digital Input 10 “High Speed Command”
green	3		2	DigIN9/	Digital Input 9 “High Speed Command” complement
yellow	4			DigIN9	Digital Input 9 “High Speed Command”
grey	5		3	DigIN7/	Digital Input 7 “High Speed Command” complement
pink	6			DigIN7	Digital Input 7 “High Speed Command”
blue	7		4	DigIN8/	Digital Input 8 “High Speed Command” complement
red	8			DigIN8	Digital Input 8 “High Speed Command”
black	9		5	$+V_{AUX}$	Auxiliary output voltage +5 VDC / 150 mA
violet	10			D_GND	Digital signal ground
grey/ pink	11		6	DigOUT5/	Digital Output 5 “High Speed Command” complement
red/blue	12			DigOUT5	Digital Output 5 “High Speed Command”

Table 2-25 Signal Cable 6x2core – Pin Assignment EPOS2 50/5

2.3.3 EPOS2 Module 36/2

Connector Array

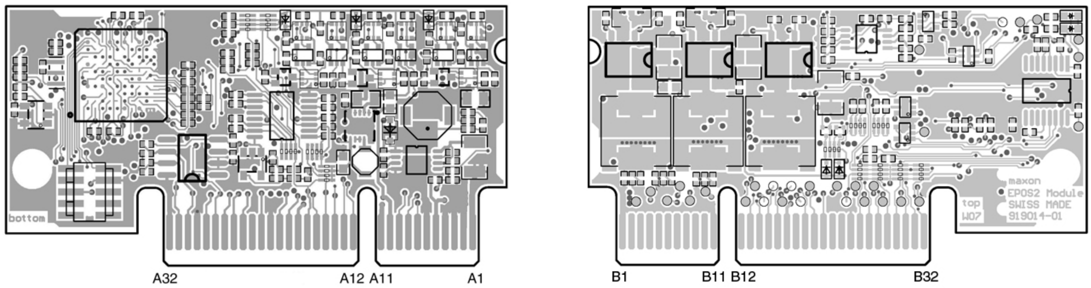


Figure 2-9 EPOS2 Module 36/2 – PCB with Connector Array

PCB Connectors	
PCB	On-board card edge connector
Suitable plugs	PCI Express (PCIe), 2 x 32 pins (vertical or horizontal), pitch 1 mm Vertical: Tyco (2-1775801-1) or FCI (10018783-11111TLF) Horizontal: Tyco (1761465-2) or Meritec (983172-064-2MMF)
Suitable retainer	FCI PCI Express Retainer, blue (10042618-002LF)

Table 2-26 EPOS2 Module 36/2 – PCB Connectors

Pin	Signal	Description
A6	Power_GND	Ground of supply voltage
A10	$+V_{aux}$	Auxiliary voltage output +5 VDC
	$+V_{DDin}$	Auxiliary supply voltage input +5 VDC (optional)
A21	GND	Ground of digital output
A22	DigOUT5	Digital Output 5
B12	GND	Ground of digital input
B13	DigIN1	Digital Input 1
B14	DigIN2	Digital Input 2
B15	DigIN3	Digital Input 3
B16	DigIN4	Digital Input 4
B17	GND	Ground of digital input
B18	DigIN7	Digital Input 7 “High Speed Command”
B19	DigIN7\	Digital Input 7 “High Speed Command” complement
B20	DigIN8	Digital Input 8 “High Speed Command”
B21	DigIN8\	Digital Input 8 “High Speed Command” complement
B22	DigOUT1	Digital Output 1
B23	DigOUT2	Digital Output 2

Table 2-27 EPOS2 Module 36/2 – Pin Assignment

2.3.4 EPOS2 24/5

Signal Cable 16core (275932) – Connector J6

Head A **Head B**

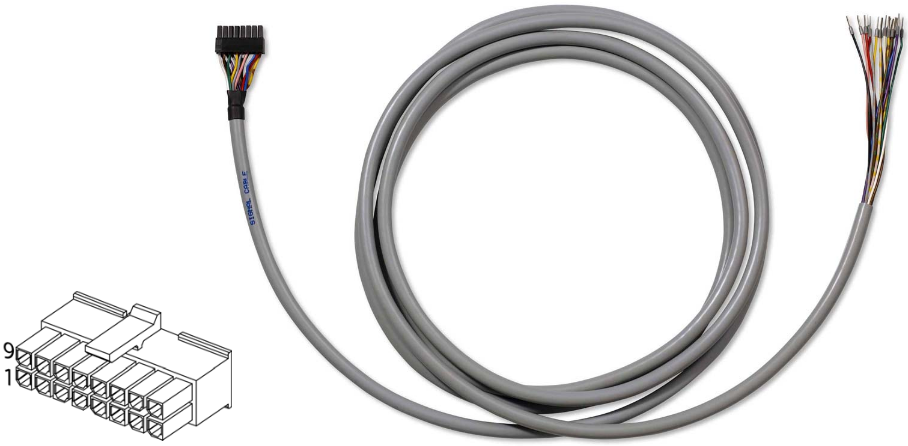


Figure 2-10 Signal Cable 16core

Technical Data	
Cable cross-section	16 x 0.14 mm²
Length	3 m
Head A	Molex Micro-Fit 3.0 16 poles (430-25-1600) Molex Micro-Fit 3.0 female crimp terminals (430-30-0010)
Head B	Cable end sleeves 0.14 mm²

Table 2-28 Signal Cable 16core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		–	D_Gnd	Digital signal ground
brown	2		–	D_Gnd	Digital signal ground
green	3		–	DigIN6	Digital Input 6 “Negative Limit Switch”
yellow	4		–	DigIN5	Digital Input 5 “Positive Limit Switch”
grey	5		–	DigIN4	Digital Input 4 “Home switch”
pink	6		–	DigIN3	Digital Input 3 “General Purpose”
blue	7		–	DigIN2	Digital Input 2 “General Purpose”
red	8		–	DigIN1	Digital Input 1 “General Purpose”
black	9 *1)		–	+Vout	Auxiliary supply voltage output (+11…+24 VDC)
	9 *2)			+VC	Logic supply voltage output (+11…+24 VDC)
violet	10		–	DigOUT4	Digital Output 4 “Brake”
grey/ pink	11		–	DigOUT3	Digital Output 3 “General Purpose”
red/blue	12		–	DigOUT2	Digital Output 2 “General Purpose”
white/ green	13		–	DigOUT1	Digital Output 1 “General Purpose”
brown/ green	*14*		–	*A_Gnd*	*Analog signal ground*
white/ yellow	*15*		–	*AnIN2*	*Analog Input 2*
yellow/ brown	*16*		–	*AnIN1*	*Analog Input 1*

Remarks:
*1) jumper JP4 is set (initial setting)
*2) if jumper JP4 is open, a separate logic supply voltage may be applied

Table 2-29 Signal Cable 16core – Pin Assignment EPOS2 24/5

2.3.5 EPOS2 24/2

Connector J1

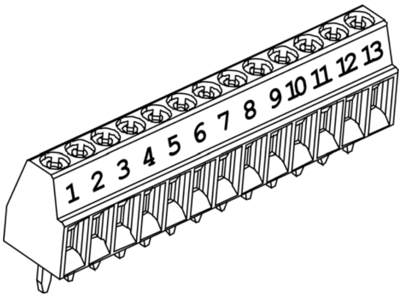


Figure 2-11 Connector J1

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
–	1		–	DigIN1	Digital Input 1 “General Purpose”
–	2		–	DigIN2	Digital Input 2 “General Purpose”
–	3		–	DigIN3	Digital Input 3 “General Purpose”
–	4		–	DigIN4	Digital Input 4 “Home Switch”
–	5		–	DigIN5	Digital Input 5 “Positive Limit Switch”
–	6		–	DigIN6	Digital Input 6 “Negative Limit Switch”
–	7		–	D_Gnd	Digital signal ground
–	8		–	$+V_{OUT}$	Auxiliary supply voltage Output (+5 VDC / 10 mA)
–	9		–	DigOUT3	Digital Output 3 “General Purpose”
–	10		–	DigOUT4	Digital Output 4 “General Purpose”
–	11		–	D_Gnd	Digital signal ground
–	12		–	Power_Gnd	Power ground
–	13		–	$+V_{CC}$	Power supply voltage (+9…24 VDC)

Table 2-30 Connector J1 – Pin Assignment EPOS2 24/2

2.4 Configuration

Configuration is handled by a dynamic wizard assisting you in selecting desired functions and assigning them to inputs and outputs of you choice.

Note
The following explanations show you how to initiate the Configuration Wizard. Its further coarse will then depend on the functions and options you will actually chose. The stated figures are thereby meant as examples.

2.4.1 Step A: Open I/O Configuration Wizard

1) Complete standard system configuration (Startup Wizard) in «EPOS Studio».
2) Doubleclick ¤I/O Configuration Wizard¤ to commence configuration.

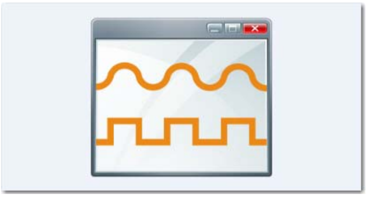


Figure 2-12 Open I/O Configuration Wizard

3) A screen will appear showing the number of I/Os available for configuration.
4) Click ¤Next¤ to continue.

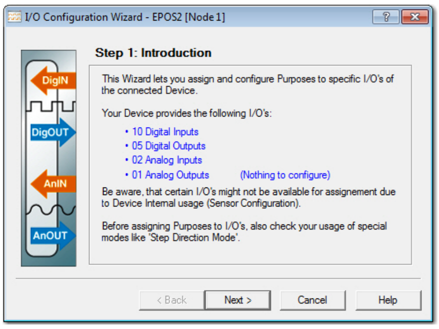


Figure 2-13 Configuration Wizard – Introduction

2.4.2 Step B: Configure Digital Inputs

1) Select predefined functions you wish to use by ticking respective check boxes. An available digital input will automatically be assigned to your selection.
2) If you wish to assign a particular digital input to a given function, select desired input from the ¤Dropdown menu¤ in column “Input”.

3) Click ¤Next¤ to continue.

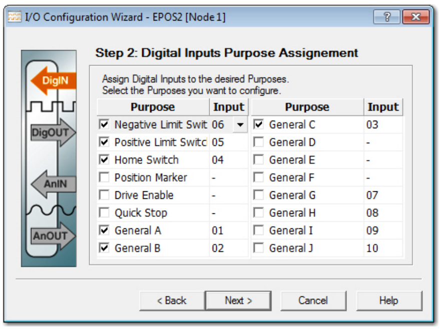


Figure 2-14 Configuration Wizard – Configure Digital Inputs

4) Define mask, type of switch (NPN or PNP) and switch output state.
5) Set limit switch error.
6) Click ¤Next¤ to continue.
7) Repeat for every earlier selected digital input.

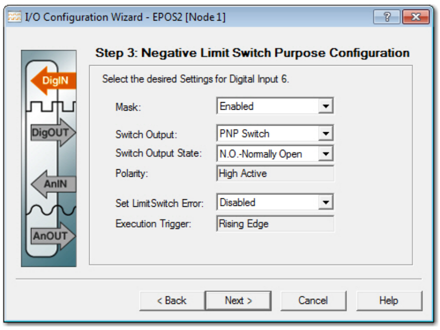


Figure 2-15 Configuration Wizard – Configure Digital Input Functionality

2.4.3 Step C: Configure Digital Outputs

1) Select predefined functions you wish to use by ticking respective check boxes. An available digital output will automatically be assigned to your selection.
2) If you wish to assign a particular digital output to a given function, select desired input from the ¤Dropdown menu¤ in column “Output”.
3) Click ¤Next¤ to continue.

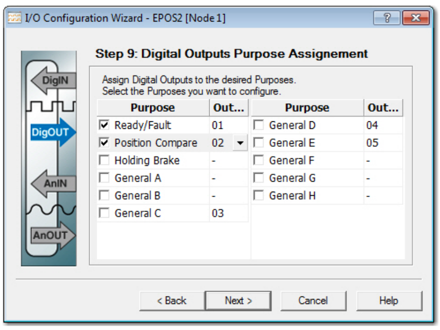


Figure 2-16 Configuration Wizard – Configure Digital Outputs

2.4.4 Step D: Save Configuration

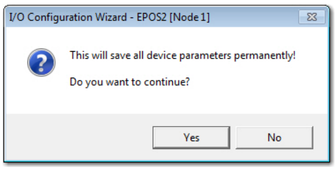


Figure 2-17 Safe Configuration

Note
You may check the status and alter the configuration at any time using the «I/O Monitor».

2.5 Wiring Examples

2.5.1 EPOS2 70/10

2.5.1.1 Proximity Switches

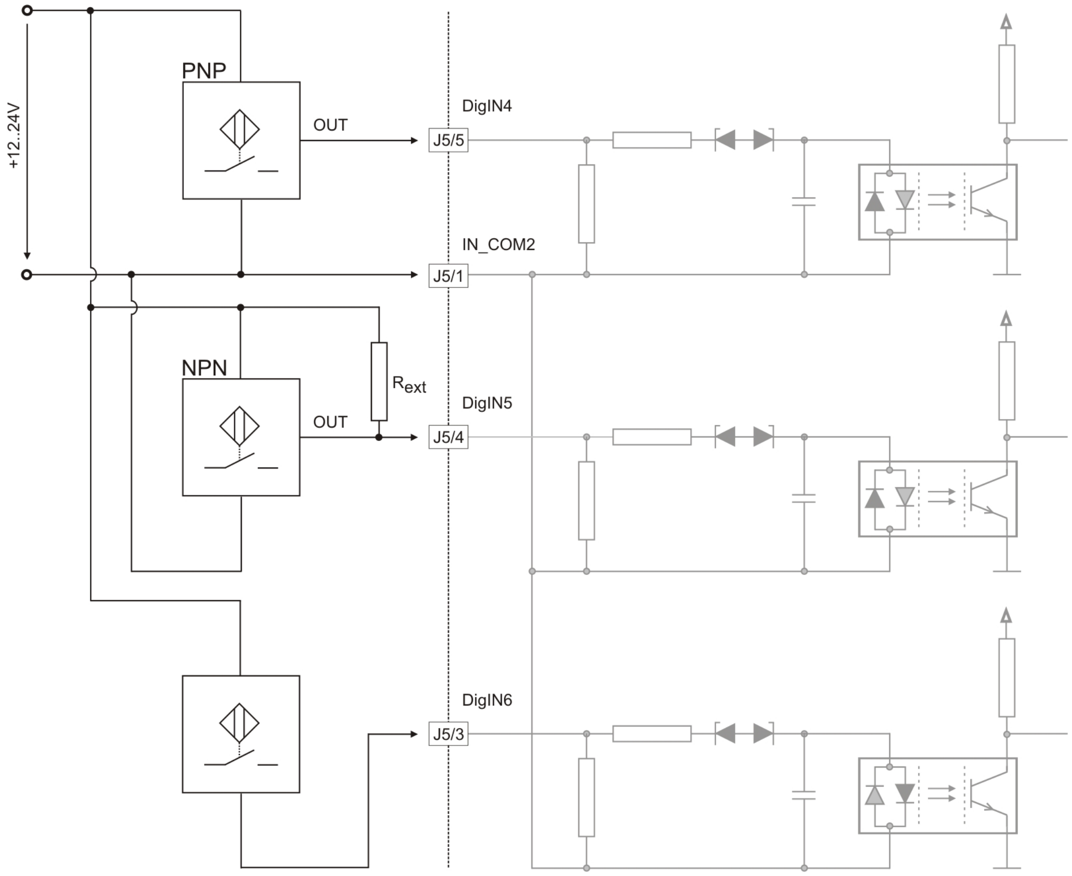


Figure 2-18 EPOS2 70/10 – DigIN4…6 / Proximity Switches

Best Practice
- *Preferably, use 3-wire PNP proximity switches.*
- *Using 3-wire NPN proximity switches requires an additional pull-up resistor.*
  R_{ext} *(12 V) = 560 Ω (300 mW)*
  R_{ext} *(24 V) = 3 kΩ (200 mW)*
- *By principle, using 2-wire proximity switches is possible.*

2.5.1.2 Permanent Magnet Brake

EPOS2 70/10 output 4 permits direct activation of loads with very high current demand (such as motor brakes and warning lights, etc.).

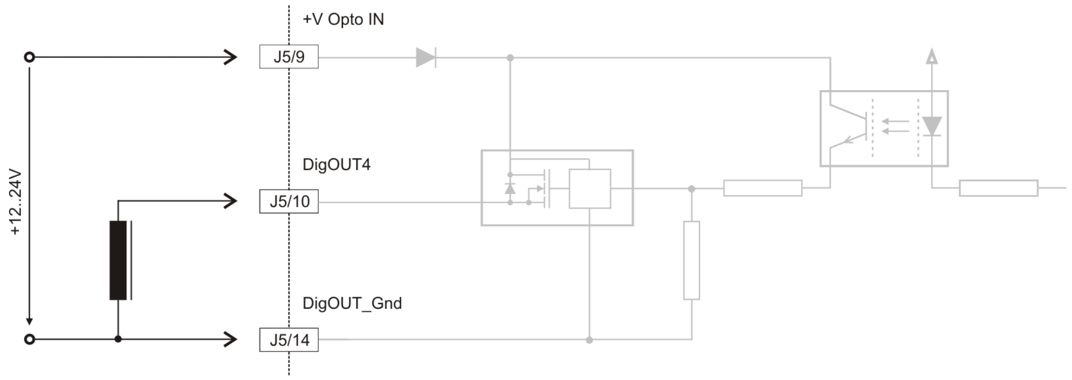


Figure 2-19 EPOS2 70/10 – DigOUT4 / permanent Magnet Brake

2.5.2 EPOS2 50/5

2.5.2.1 Proximity Switches

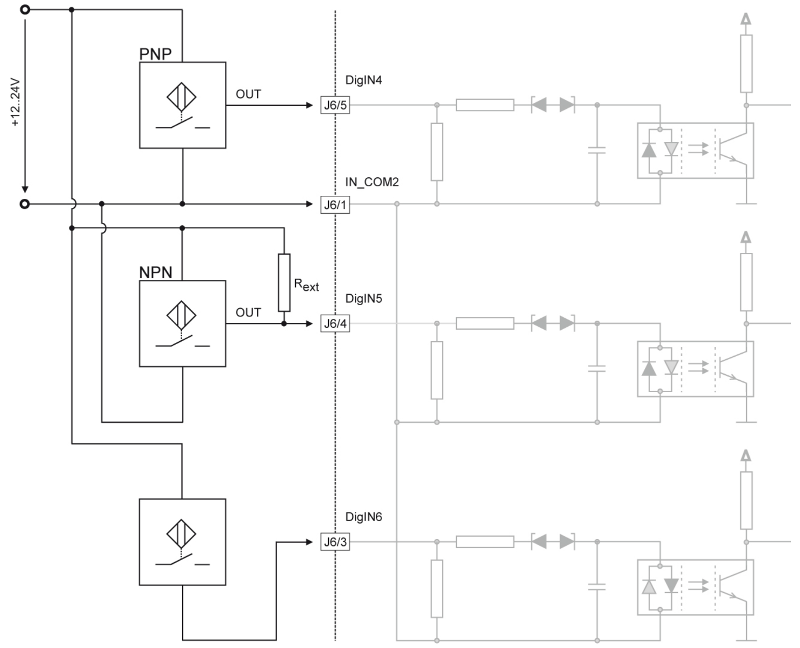


Figure 2-20 EPOS2 50/5 – DigIN4…6 / PNP/NPN Proximity Switches

Best Practice

- *We recommend the use of 3-wire PNP proximity switches.*
- *The use of 3-wire NPN proximity switches requires an additional external pull-up resistor:*
  – R_{ext} *(12 V) = 560 Ω (300 mW)*
  – R_{ext} *(24 V) = 3 kΩ (200 mW)*
- *The use of 2-wire proximity switches is possible.*

2.5.2.2 Permanent Magnet Brake

EPOS2 50/5 output 4 permits direct activation of loads with very high current demand (such as motor brakes and warning lights, etc.).

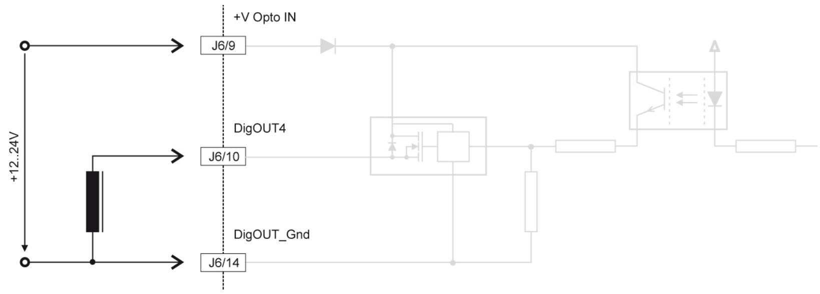


Figure 2-21 EPOS2 50/5 – DigOUT4 / permanent Magnet Brake

2.5.3 EPOS2 Module 36/2

2.5.3.1 Digital Inputs

PNP 3-Wire Model

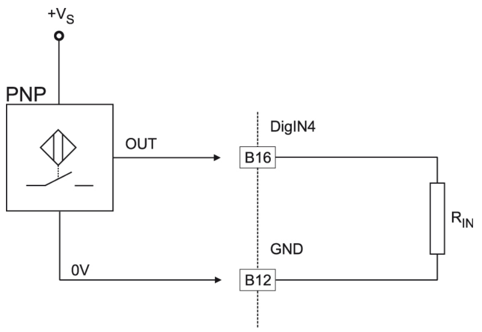


Figure 2-22 EPOS2 Module 36/2 – DigIN4 / PNP Proximity Switch (applies also for DigIN2/3)

Photoelectric Sensor

$$R_{ext} = \frac{R_{IN} \cdot (V_S - V_{IN})}{V_{IN}}$$

Note:
Logic level threshold V_{IN} assumed 5 V.

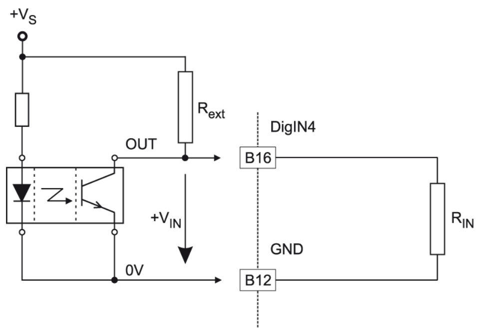


Figure 2-23 EPOS2 Module 36/2 – DigIN4 / Photoelectric Sensor (applies also for DigIN2/3)

2.5.3.2 Digital Outputs

Digital Output 1 “sink”

Max. input voltage	+36 VDC
Max. load current	50 mA
Max. voltage drop	<1.0 V @ 50 mA

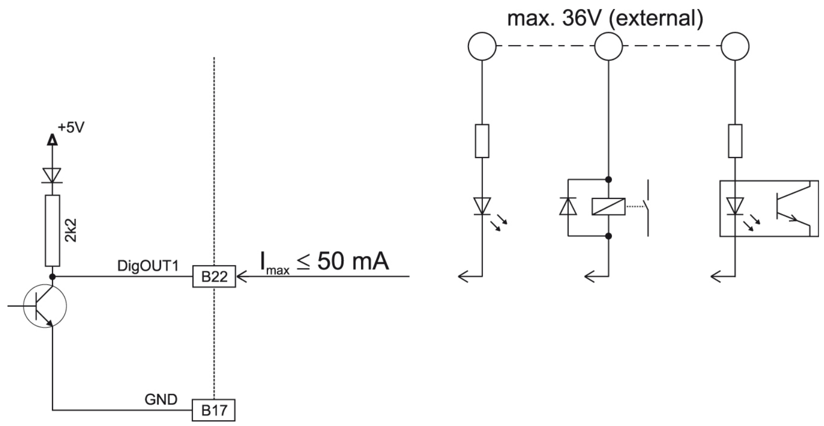


Figure 2-24 EPOS2 Module 36/2 – DigOUT1 “sink” (applies also for DigIN2)

Digital Output 1 “source”

Output voltage	$U_{out} \approx 5\ V - 0.75\ V - (I_{load} \times 2200\ \Omega)$
Max. load current	$I_{load} \leq 2$ mA

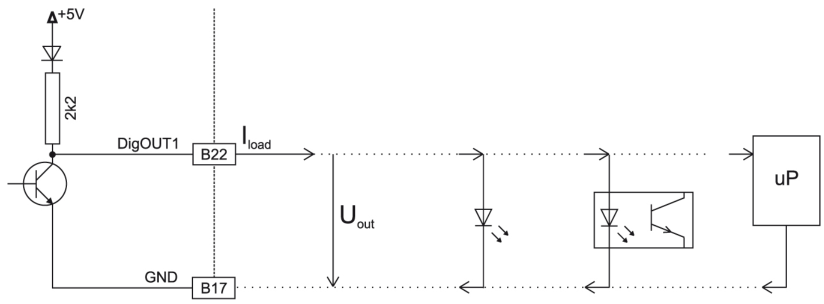


Figure 2-25 EPOS2 Module 36/2 – DigOUT1 “source” (applies also for DigIN2)

2.5.4 EPOS2 24/5

2.5.4.1 Proximity Switches

PNP 3-Wire Model

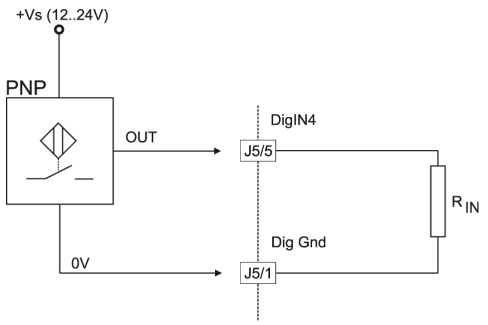


Figure 2-26 EPOS2 24/5 – DigIN4 / PNP Proximity Switch (applies also for DigIN5/6)

NPN 3-Wire Model

R_{ext} (12 V) = 510 Ω (300 mW)
R_{ext} (24 V) = 4.3 kΩ (150 mW)
R_{IN} = 4 kΩ

NPN 2-Wire Model

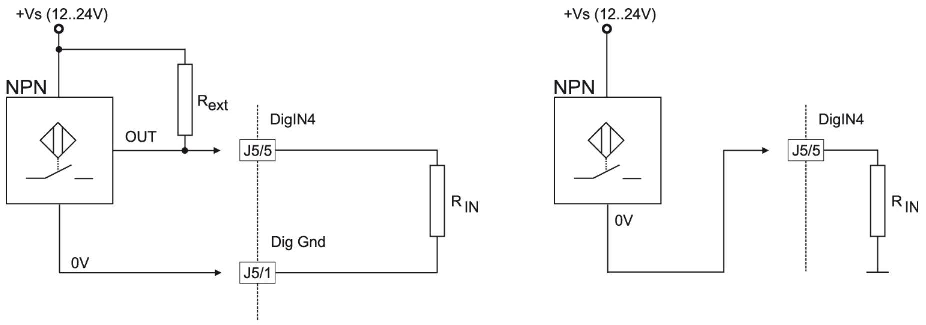


Figure 2-27 EPOS2 24/5 – DigIN4 / NPN Proximity Switch (applies also for DigIN5/6)

2.5.4.2 Digital Outputs

Digital Output “sink”

Max. input voltage	+30 VDC
Max. load current	100 mA
Max. voltage drop	0.5 V @ 100 mA

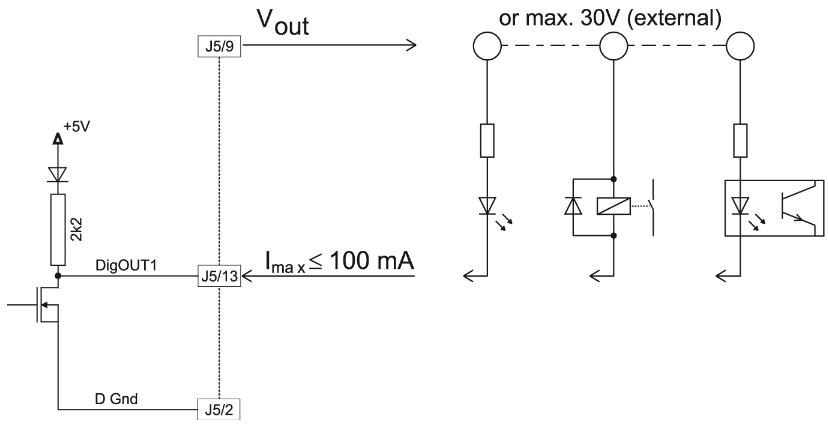


Figure 2-28 EPOS2 24/5 – DigOUT1 “sink”

Digital Output “source”

Output voltage	$U_{out} \approx 5\ V - 0.75\ V - (I_{load} \times 2200\ \Omega)$
Max. load current	$I_{load} \leq 2\ mA$

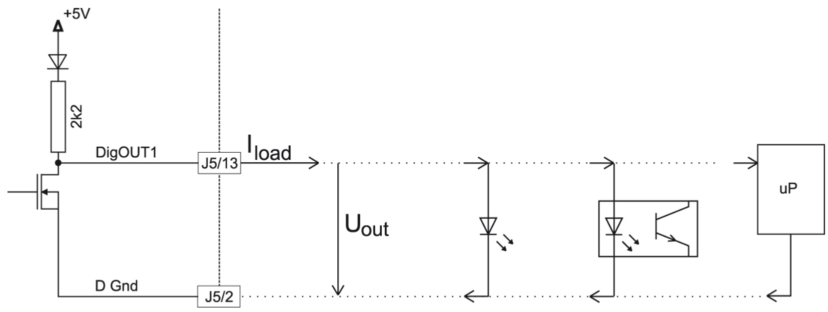


Figure 2-29 EPOS2 24/5 – DigOUT1 “source”

2.5.5 EPOS2 24/2

2.5.5.1 Proximity Switches

PNP 3-Wire Model
R_{IN} = 11 kΩ

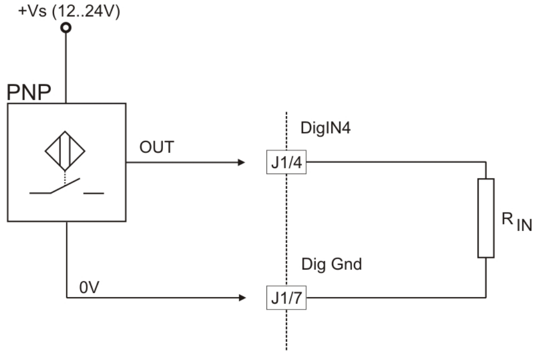


Figure 2-30 EPOS2 24/2 – DigIN4 / PNP Proximity Switch (applies also for DigIN5/6)

2.5.5.2 Photoelectric Sensor

3-Wire Model
R_{ext} = (12 V) = 20 kΩ (300 mW)
R_{ext} = (24 V) = 51 kΩ (150 mW)
R_{IN} = 11 kΩ

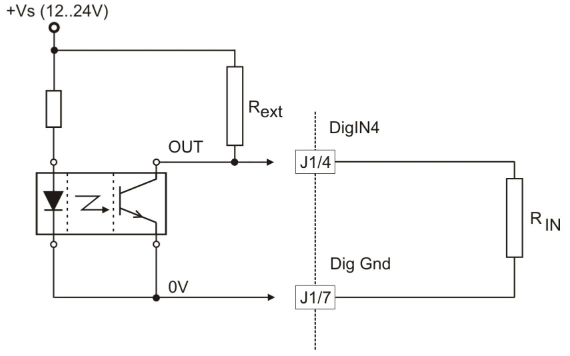


Figure 2-31 EPOS2 24/2 – DigIN4 / Photoelectric Sensor (analogously valid also for DigIN5/6)

••page intentionally left blank••

3 Analog Inputs and Outputs

3.1 In Brief

Drive systems typically require inputs and outputs.

The analog inputs may be used for general purpose process values (such as temperature, pressure, torque from an external sensor, etc.). Also featured are predefined functions for analog inputs (such as respective setpoints for Current Mode, Velocity Mode and Position Mode).

EPOS2 50/5 additionally supports an analog output for general purposes.

The inputs and outputs can easily be configured using the «Configuration Wizard» and may be changed online via CANopen or serial bus.

3.1.1 Objective

The present Application Note explains the functionality of analog inputs and outputs and features “in practice examples” suitable for daily use.

Contents

3.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	Cable Starting Set Hardware Reference
EPOS2 50/5	347717	2110h or higher	Cable Starting Set Hardware Reference
EPOS2 Module 36/2	360665	2110h or higher	Hardware Reference
EPOS2 24/5	367676	2110h or higher	Cable Starting Set Hardware Reference
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	Cable Starting Set Hardware Reference

Table 3-31 Analog Inputs and Outputs – covered Hardware and required Documents

3.1.3 Tools

Tools	Description
Crimper	Molex hand crimper (63819-0000)
	Molex hand crimper (63819-0900)
Software	«EPOS Studio» Version 2.00 or higher

Table 3-32 Analog Inputs and Outputs – recommended Tools

3.2 Functionality

3.2.1 Analog Inputs

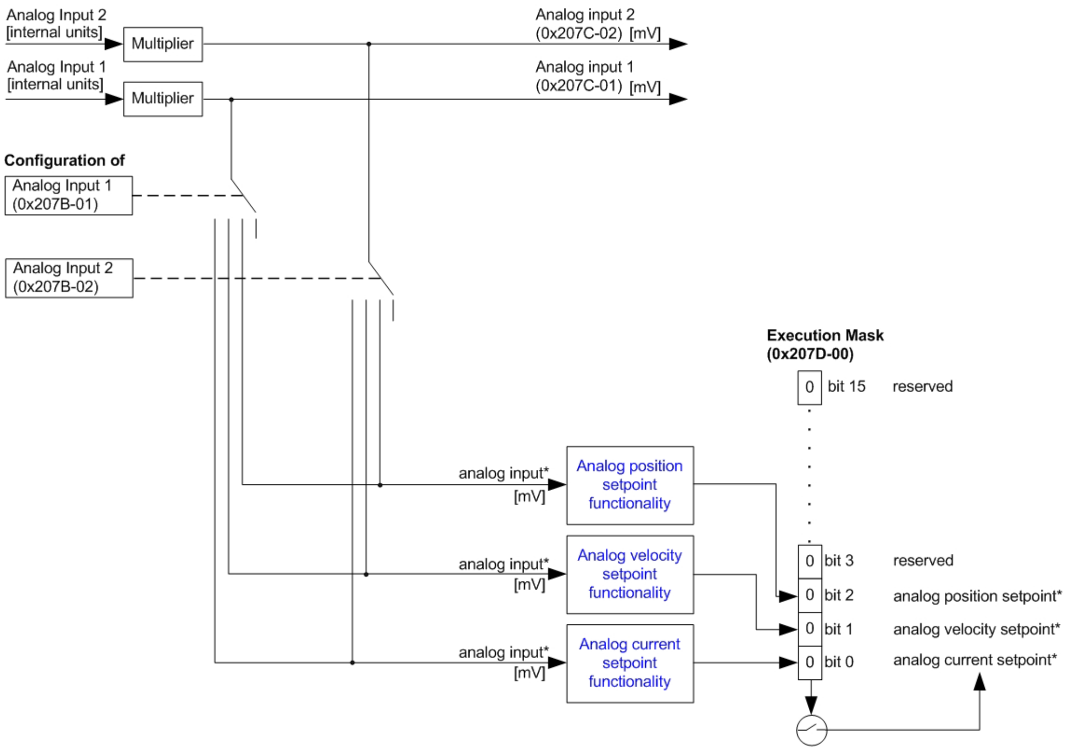


Figure 3-32 Analog Input Functionality – EPOS2 Overview (default Configuration)

Configuration Parameter

Name	Index	Sub-index	Description
Configuration of Analog Input 1 (➔Table 3-35)	0x207B	0x01	Defines functionality assigned to AnIN1.
Configuration of Analog Input 2 (➔Table 3-35)	0x207B	0x02	Defines functionality assigned to AnIN2.
Analog Input Functionalities Execution Mask (➔Table 3-36)	0x207D	0x00	Execution of analog input functionality can be inhibited.

Table 3-33 Analog Input – Configuration Parameter

Input Parameter

Name	Index	Sub-index	Description
Analog Input 1	0x207C	0x01	Display measured voltage at AnIN1.
Analog Input 2	0x207C	0x02	Display measured voltage at AnIN2.

Table 3-34 Analog Input – Input Parameter

Input Configuration Values

Parameter “Configuration of Analog Input” defines bit position in “Analog Input Functionalities State”.

Value	Functionality	Description
15	General Purpose A	State can be read.
14	General Purpose B	State can be read.
13	General Purpose C	State can be read.
12	General Purpose D	State can be read.
11	General Purpose E	State can be read.
10	General Purpose F	State can be read.
9	General Purpose G	State can be read.
8	General Purpose H	State can be read.
7…3	reserved	–
2	Position Setpoint	Analog input is used to command control function in Position Mode.
1	Velocity Setpoint	Analog input is used to command control function in Velocity Mode.
0	Current Setpoint	Analog input is used to command control function in Current Mode.

Table 3-35 Analog Input – Input Configuration Values

Parameter Description

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved	reserved	reserved	reserved	reserved	Position Setpoint	Velocity Setpoint	Current Setpoint

Table 3-36 Analog Input – Execution Mask Parameter

Note
With the execution mask, execution of analog input functionality can be inhibited.

3.2.2 Analog Output (EPOS2 50/5 only)

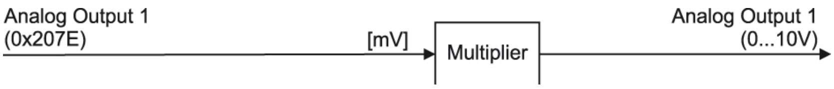


Figure 3-33 Analog Output Functionality – EPOS2 Overview (default Configuration)

Output Parameter

Name	Index	Sub-index	Description
Analog Output 1	0x207E	0x00	Defines voltage level set at AnOUT1.

Table 3-37 Analog Output – Output Parameter

Note
This object is used to set the voltage level [mV] of the Analog Output 1. Immediately after write to this object, the value is transferred to the Analog Output 1.

3.3 Connection

3.3.1 EPOS2 70/10

Signal Cable 6x2core (300586) – Connector J5A

Head A **Head B**

Figure 3-34 Signal Cable 6x2core

Technical Data	
Cable cross-section	6 x 2 x 0.14 mm²
Length	3.00 m
Head A	Molex Micro-Fit 3.0 12 poles (430-25-1200) Molex Micro-Fit 3.0 female crimp terminals (43030-xxxx)
Head B	Cable end sleeves 0.14 mm²

Table 3-38 Signal Cable 6x2core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		1	+5VOUT	Reference output voltage +5 V
brown	2			A_Gnd	Analog signal ground
green	3		2	AnIN2−	Negative analog signal input 2
yellow	4			AnIN2+	Positive analog signal input 2
grey	5		3	AnIN1−	Negative analog signal input 1
pink	6			AnIN1+	Positive analog signal input 1
blue	*7*		*4*	*D_GND*	*Digital signal ground*
red	*8*			*D_GND*	*Digital signal ground*
black	*9*		*5*	*DigIN8/*	*Digital input 8 "High Speed Command" complement or cos− input*
violet	*10*			*DigIN8*	*Digital input 8 "High Speed Command" or cos+ input*
grey/ pink	*11*		*6*	*DigIN7/*	*Digital input 7 "High Speed Command" complement or sin− input*
red/blue	*12*			*DigIN7*	*Digital input 7 "High Speed Command" or sin+ input*

Table 3-39 Signal Cable 6x2core – Pin Assignment EPOS2 70/10

3.3.2 EPOS2 50/5

Signal Cable 4x2core (350390) – Connector J7

Head A **Head B**

Figure 3-35 Signal Cable 4x2core

Technical Data	
Cable cross-section	4 x 2 x 0.14 mm^2
Length	3.00 m
Head A	Molex Micro-Fit 3.0 8 poles (430-25-0800) Molex Micro-Fit 3.0 female crimp terminals (430-30-0010)
Head B	Cable end sleeves 0.14 mm^2

Table 3-40 Signal Cable 4x2core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		1	AnOUT1	Analog signal output 1”General Purpose”
red	2		4	not connected	–
brown	3		1	A_Gnd	Analog signal ground
green	4		2	AnIN2−	Negative analog signal input 2 “General Purpose”
yellow	5		2	AnIN2+	Positive analog signal input 2 “General Purpose”
grey	6		3	AnIN1−	Negative analog signal input 1 “General Purpose”
pink	7		3	AnIN1+	Positive analog signal input 1 “General Purpose”
blue	8		4	A_Gnd	Analog signal ground

Table 3-41 Signal Cable 4x2core – Pin Assignment EPOS2 50/5

3.3.3 EPOS2 Module 36/2

Connector Array

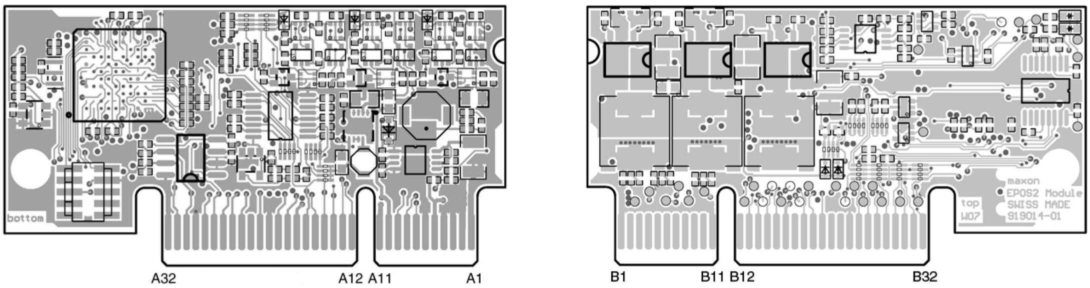


Figure 3-36 EPOS2 Module 36/2 – PCB with Connector Array

PCB Connectors	
PCB	On-board card edge connector
Suitable plugs	PCI Express (PCIe), 2 x 32 pins (vertical or horizontal), pitch 1 mm Vertical: Tyco (2-1775801-1) or FCI (10018783-11111TLF) Horizontal: Tyco (1761465-2) or Meritec (983172-064-2MMF)
Suitable retainer	FCI PCI Express Retainer, blue (10042618-002LF)

Table 3-42 EPOS2 Module 36/2 – PCB Connectors

Pin	Signal	Description
A18	GND	Analog input ground
A19	AnIN1	Analog Input 1
A20	AnIN2	Analog Input 2
others	➔separate document «EPOS2 Module 36/2 Hardware Reference»	

Table 3-43 EPOS2 Module 36/2 – Pin Assignment

3.3.4 EPOS2 24/5

Signal Cable 16core (275932) – Connector J5

Head A **Head B**

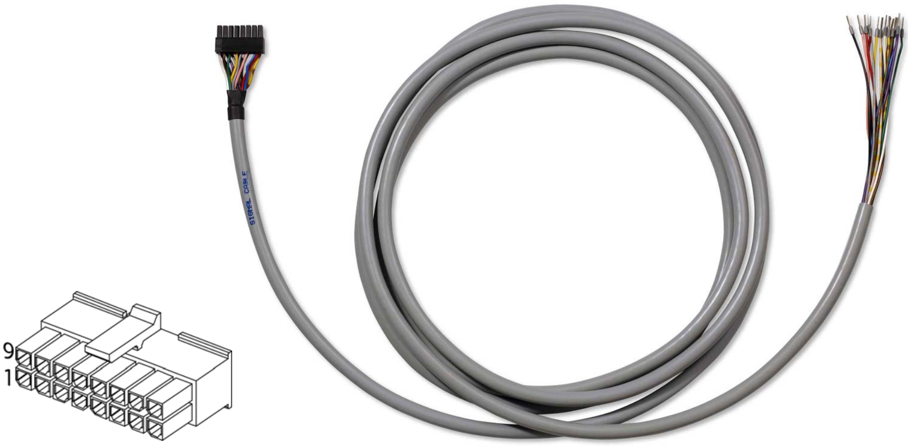


Figure 3-37 Signal Cable 16core

Technical Data	
Cable cross-section	16 x 0.14 mm²
Length	3 m
Head A	Molex Micro-Fit 3.0 16 poles (430-25-1600) Molex Micro-Fit 3.0 female crimp terminals (430-30-0010)
Head B	Cable end sleeves 0.14 mm²

Table 3-44 Signal Cable 16core – Technical Data

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
white	1		–	D_Gnd	Digital signal ground
brown	2		–	D_Gnd	Digital signal ground
green	3		–	DigIN6	Digital Input 6 “Negative Limit Switch”
yellow	4		–	DigIN5	Digital Input 5 “Positive Limit Switch”
grey	5		–	DigIN4	Digital Input 4 “Home switch”
pink	6		–	DigIN3	Digital Input 3 “General Purpose”
blue	7		–	DigIN2	Digital Input 2 “General Purpose”
red	8		–	DigIN1	Digital Input 1 “General Purpose”
black	9 *1)		–	$+V_{out}$	Auxiliary supply voltage output (+11…+24 VDC)
	9 *2)			$+V_C$	Logic supply voltage output (+11…+24 VDC)
violet	10		–	DigOUT4	Digital Output 4 “Brake”
grey/pink	11		–	DigOUT3	Digital Output 3 “General Purpose”
red/blue	12		–	DigOUT2	Digital Output 2 “General Purpose”
white/green	13		–	DigOUT1	Digital Output 1 “General Purpose”
brown/green	14		–	A_Gnd	Analog signal ground
white/yellow	15		–	AnIN2	Analog Input 2
yellow/brown	16		–	AnIN1	Analog Input 1

Remarks:
*1) jumper JP4 is set (initial setting)
*2) if jumper JP4 is open, a separate logic supply voltage may be applied

Table 3-45 Signal Cable 16core – Pin Assignment EPOS2 24/5

3.3.5 EPOS2 24/2

Connector J2

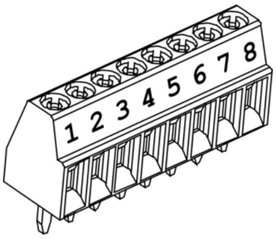


Table 3-46 Connector J2

Wire	Head A Pin	Head B Pin	Twisted Pair	Signal	Description
–	*1*		*1*	*CAN high*	*CAN high bus line*
–	*2*		*4*	*CAN low*	*CAN low bus line*
–	*3*		*1*	*RS232 RxD*	*RS232 receive*
–	*4*		*2*	*RS232 TxD*	*RS232 transmit*
–	*5*		*2*	*GND*	*Ground*
–	6		3	AnIN1	Analog Input 1
–	7		3	AnIN2	Analog Input 2
–	8		4	A_Gnd	Analog signal ground

Table 3-47 Connector J2 – Pin Assignment EPOS2 24/2

3.4 Configuration

Configuration is handled by a dynamic wizard assisting you in selecting desired functions and assigning them to inputs and outputs of you choice.

Note
The following explanations show you how to initiate the Configuration Wizard. Its further coarse will then depend on the functions and options you will actually chose. The stated figures are thereby meant as examples.

3.4.1 Step A: Open I/O Configuration Wizard

1) Complete standard system configuration (Startup Wizard) in «EPOS Studio».
2) Doubleclick ¤I/O Configuration Wizard¤ to commence configuration.

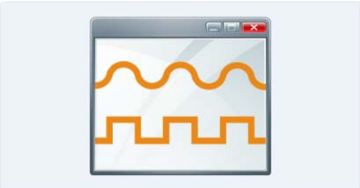


Figure 3-38 Open I/O Configuration Wizard

3) A screen will appear showing the number of I/Os available for configuration.
4) Click ¤Next¤ to continue.

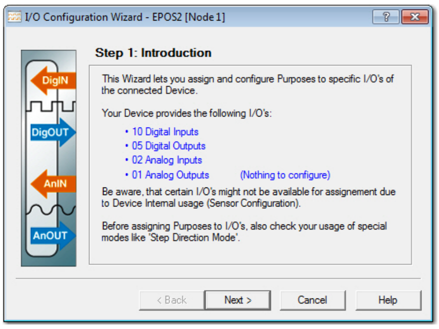


Figure 3-39 Configuration Wizard – Introduction

5) Click ¤Next¤ several times to skip configuration of digital I/Os.

3.4.2 Step B: Configure Analog Inputs

1) Select predefined functions you wish to use by ticking respective check boxes. An available analog input will automatically be assigned to your selection.
2) If you wish to assign a particular analog input to a given function, select desired input from the ¤Dropdown menu¤ in column “Input”.

3) Click ¤Next¤ to continue.

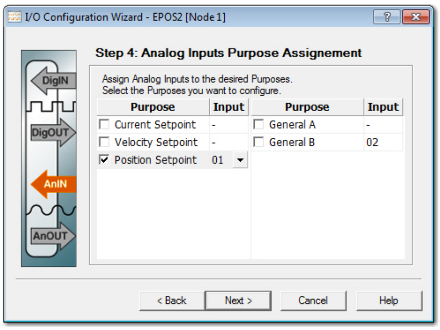


Figure 3-40 Configuration Wizard – Configure Analog Inputs

4) Define execution mask, setpoint scaling and setpoint offset.
5) Click ¤Next¤ to continue.
6) Repeat for every earlier selected analog input.

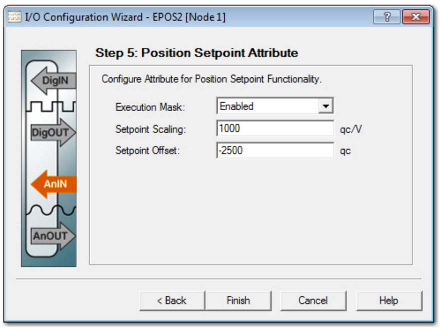


Figure 3-41 Configuration Wizard – Configure Analog Input Functionality

3.4.3 Step C: Save Configuration

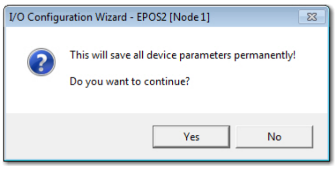


Figure 3-42 Safe Configuration

Note
You may check the status and alter the configuration at any time using the «I/O Monitor».

••page intentionally left blank••

4 Master Encoder Mode

4.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

Alternatively, EPOS2 can also be commanded by digital position values. Used are either an incremental encoder (Master Encoder Mode) for setting the values of the device, or PLC-generating step pulses (Step/Direction Mode) can be used to command the device. Inputs and outputs can easily be configured using the «Configuration Wizard» and may be changed online via CANopen or serial bus.

4.1.1 Objective

In «Master Encoder Mode», the motor follows a reference input produced by an external encoder. A gearing factor may also be defined using software parameters. Two motors can be very easily synchronised using this method.

The present Application Note explains structure, functionality and use of the operation mode «Master Encoder Mode» and features “in practice examples” suitable for daily use.

Contents

4.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	
EPOS2 50/5	347717	2110h or higher	
EPOS2 Module 36/2	360665	2110h or higher	
EPOS2 24/5	367676	2110h or higher	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	

Table 4-48 Master Encoder Mode – covered Hardware and required Documents

4.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 4-49 Master Encoder Mode – recommended Tools

4.2 System Structure

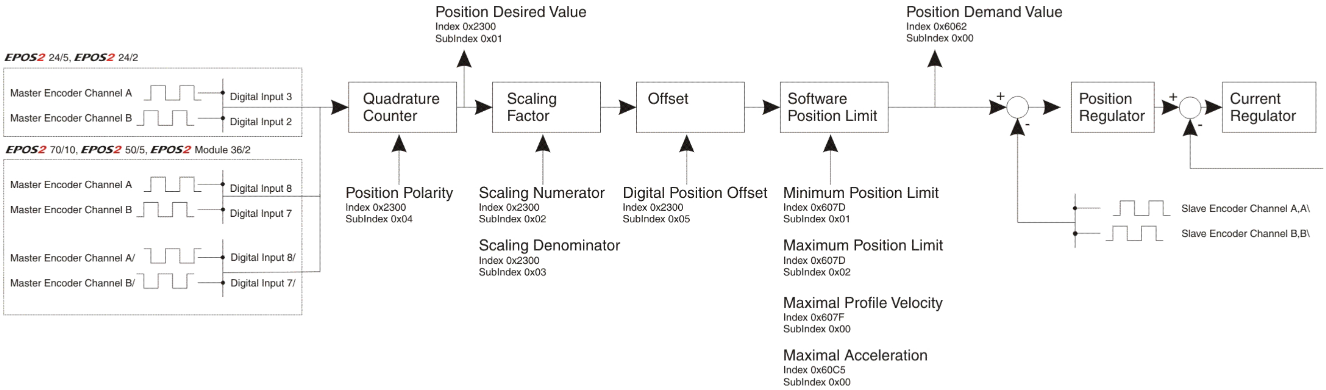


Figure 4-43 Master Encoder Mode – System Structure

Quadrature Counter

EPOS2 70/10, EPOS2 50/5 & EPOS2 Module 36/2

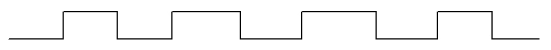
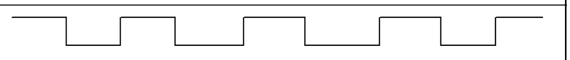
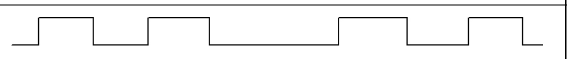
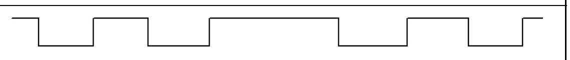
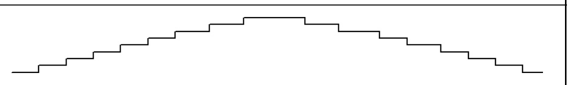
Channel A	Digital Input 8	
Channel A\	Digital Input 8\	
Channel B	Digital Input 7	
Channel B\	Digital Input 7\	
Digital Position Desired Value (Polarity = 0)		

Table 4-50 Quadrature Counter – EPOS2 70/10, EPOS2 50/5 & EPOS2 Module 36/2

EPOS2 24/5 & EPOS2 24/2

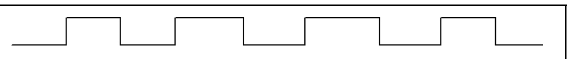
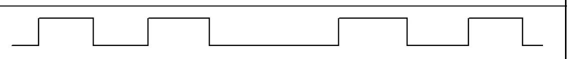
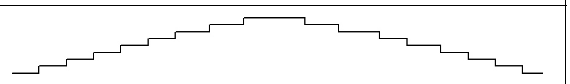
Channel A	Digital Input 3	
Channel B	Digital Input 2	
Digital Position Desired Value (Polarity = 0)		

Table 4-51 Quadrature Counter – EPOS2 24/5 & EPOS2 24/2

Value	EPOS2 70/10 EPOS2 50/5	EPOS2 Module 36/2	EPOS2 24/5	EPOS2 24/2
Input Voltage	0…5 VDC	0…5 VDC	0…24 VDC	0…24 VDC
Max. Input Voltage	−12…+12 VDC	−24…+24 VDC	−30…+30 VDC	−30…+30 VDC
Logic 0	typical <1.0 V	typical <0.8 V	typical <1.5 V	typical <0.7 V
Logic 1	typical >2.4 V	typical >2.0 V	typical >3.0 V	typical >2.4 V
Max. Input Frequency	5 MHz (differential) 2.5 MHz (single-ended)	5 MHz (differential) 2.5 MHz (single-ended)	100 kHz (3.3…5.0 V)	300 kHz (3.3…5.0 V)

Table 4-52 Master Encoder Mode – Hardware Description (Digital Inputs)

Input Parameter

Name	Index	Sub-index	Description
Digital Position Scaling Numerator	0x2300	0x02	Numerator of the scaling factor. Can be used for electronic gearing or to reduce to input frequency.
Digital Position Scaling Denominator	0x2300	0x03	Denominator of the scaling factor. Can be used for electronic gearing or to reduce to input frequency.
Digital Position Polarity	0x2300	0x04	Polarity of the direction input. The direction can be changed (0 = positive, 1 = negative).
Digital Position Offset	0x2300	0x05	Gives a dynamic displacement in reference to the encoder's desired position.
Minimum Position Limit	0x607D	0x01	Defines the negative position limit for the position demand value.
Maximum Position Limit	0x607D	0x02	Defines the positive position limit for the position demand value.
Maximum ProfileVelocity	0x607F	0x00	This value is used as velocity limit in a position (or velocity) profile mode.
Maximum Acceleration	0x60C5	0x00	Allows to limit the acceleration to prevent mechanical damages. Represents the limit of the other acceleration/deceleration objects.

Table 4-53 Master Encoder Mode – Input Parameter

Output Parameter

Name	Index	Sub-index	Description
Digital Position Desired Value	0x2300	0x01	Counter value of the up/down counter. Serves as base for the scaling and limiting functions.
Position Demand Value	0x6062	0x00	The Master Encoder Mode's output after scaling and limiting. It is the setting value for the position regulator.

Table 4-54 Master Encoder Mode – Output Parameter

Best Practice

- *Use a scaling factor ≤1 for better behavior. Due to the fact that no interpolation is implemented, movements with factors >1 will result in bigger position jumps, thus producing current peaks.*
- *Switch off software position limitation and set maximum /minimum position limits to INT32_MAX, respectively to INT32_MIN!*

4.3 Configuration

4.3.1 Step 1: System Configuration

Complete standard system configuration (Startup Wizard) in «EPOS Studio» (➔separate document «Getting Started» of respective hardware. Thereby observe following topics:

- Minimum External Wiring
- Communication Setting
- Motor Type
- Motor Pole Pair
- Motor Data
- Position Sensor Type
- Position Regulation

Figure 4-44 Startup Wizard

4.3.2 Step 2: Regulation Tuning

In Master Encoder Mode, current regulator and position regulator must be tuned. Speed regulator will not be used (➔separate document «Getting Started» of respective hardware).

Best Practice
- *Use Profile Position Mode to test regulator behavior!*
- *Use Position Mode for small steps, only!*

- Current Regulator (Current Step)
- Position Regulator (Profile Position Step)

Figure 4-45 Regulation Tuning

4.3.3 Step 3: I/O Configuration and Wiring

1) Perform wiring:

Hardware	From	To
EPOS2 70/10 EPOS2 50/5 EPOS2 Module 36/2	Master Encoder Channel A	Digital Input 8, 8\
	Master Encoder Channel B	Digital Input 7, 7\
EPOS2 24/5 EPOS2 24/2	Master Encoder Channel A	Digital Input 3
	Master Encoder Channel B	Digital Input 2

Table 4-55 Master Encoder Mode – Wiring

2) Start I/O Configuration Wizard to configure I/Os.

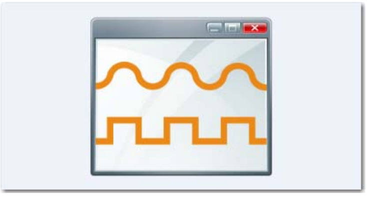


Figure 4-46 Configuration Wizard

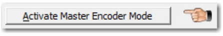
3) Configure inputs:

Hardware	Configure…	…as…
EPOS2 70/10 EPOS2 50/5 EPOS2 Module 36/2	Digital Input 7	General Purpose A
	Digital Input 8	General Purpose B
	any available Digital Input	Enable *1)
	any available Digital Output	Ready *2)
EPOS2 24/5 EPOS2 24/2	Digital Input 2	General Purpose A
	Digital Input 3	General Purpose B
	any available Digital Input	Enable *1)
	any available Digital Output	Ready *2)
Remarks: *1) In order to clear a fault condition, the device must be reset. Set input "Enable" to active. *2) Output "Ready" can be used to report a fault condition.		

Table 4-56 Configuration of Inputs

4.3.4 Step 4: Master Encoder Mode

Activate and configure Master Encoder Mode using «EPOS Studio».



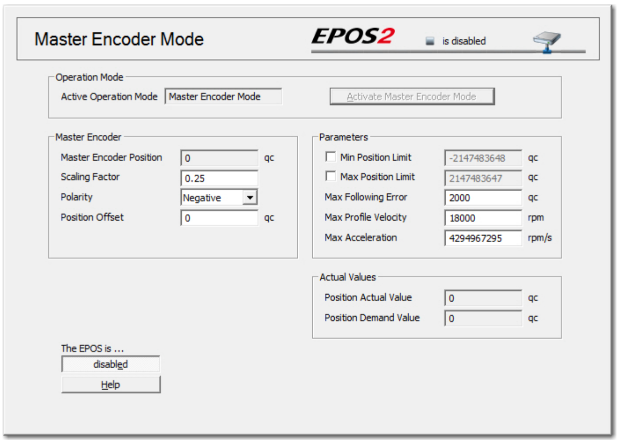


Figure 4-47 Master Encoder Mode – Configuration

4.3.5 Step 5: Save all Parameters

1) Click right on used node (Navigation Window -> Workspace or Communication).
2) Click menu item ¤Save All Parameter¤.

4.4 Application Examples

A typical application for the Master Encoder Mode is a dual axes system.

- The master axis is configured, enabled and commanded via the serial interface (RS232, USB or CAN bus) and is working in "ProfilePosition Mode" or "Profile Velocity Mode".
- The slave axis is working in "Master Encoder Mode".
- The CAN bus interface is only used for configuration, monitoring and enabling.
- The set values for the slave axis are calculated using the encoder signals of the master axis.

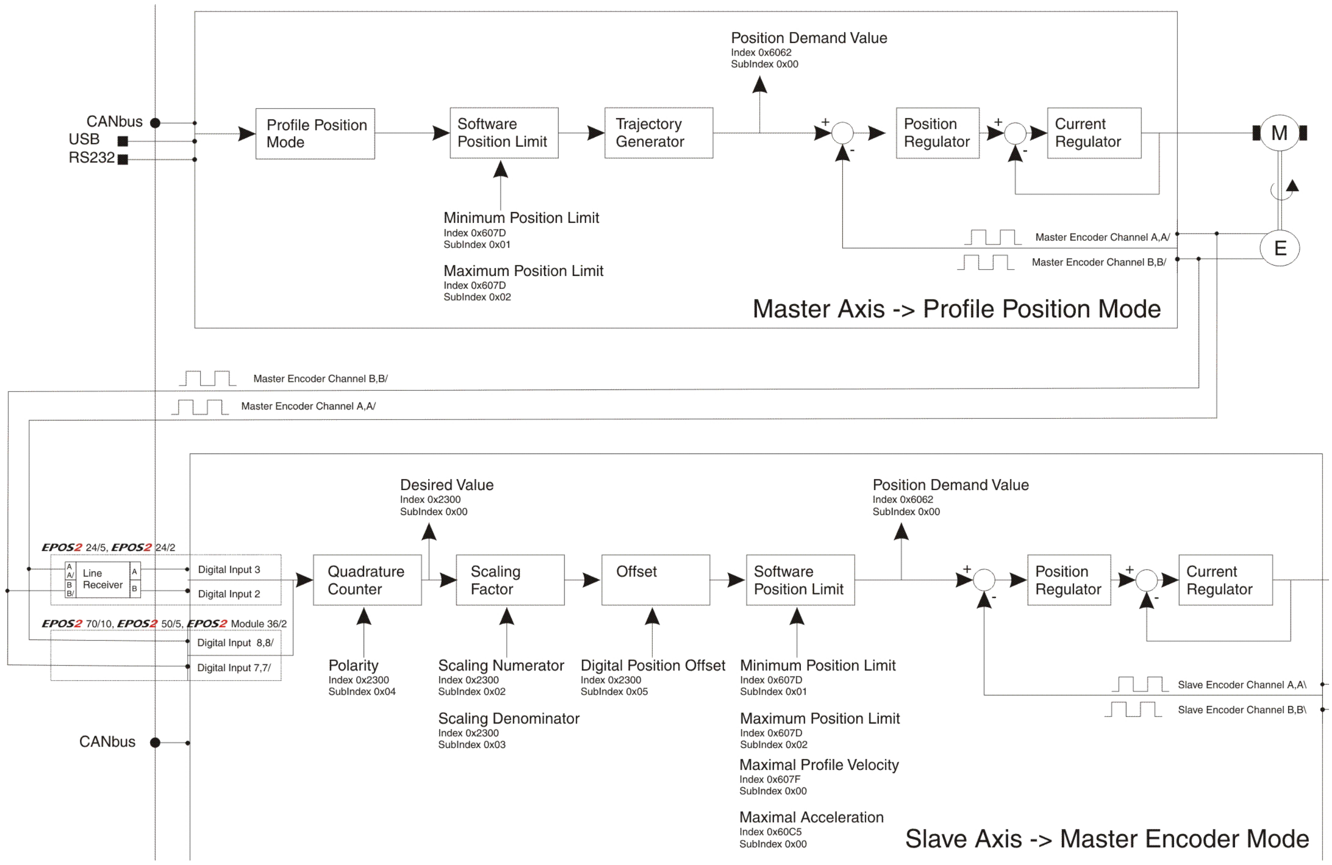


Figure 4-48 Master Encoder Mode – Application Example: Dual Axes System

Calculation of Velocity of Slave Axis

The velocity of the slave axis is not only defined by the scaling factor, but also by the ratio of the encoder resolution of the master and slave axes.

$$Velocity_{SlaveAxis} = Velocity_{MasterAxis} \cdot \frac{EncRes_{MasterAxis}}{EncRes_{SlaveAxis}} \cdot Polarity[1,-1] \cdot \frac{ScalingNumerator_{SlaveAxis}}{ScalingDenominator_{SlaveAxis}}$$

$EncRes$ [pulses per turn]

$Velocity$ [rpm]

Limiting Factors

Maximal permitted Motor Speed
Below figures represent theoretical achievable speeds. For applicable maximum permissible speed of the employed motor ➔catalog motor data!

Main limiting factor is the input frequency of the encoder signals.

Master Axis Encoder [pulse/turn]	Slave Axis Max. Input Frequency			Master Axis Max. Velocity [rpm] (Scaling Factor 1)
500	EPOS2 70/10	differential	5 MHz	600 000
500	EPOS2 70/10	single-ended	2.5 MHz	300 000
500	EPOS2 50/5	differential	5 MHz	600 000
500	EPOS2 50/5	single-ended	2.5 MHz	300 000
500	EPOS2 Module 36/2	differential	5 MHz	600 000
500	EPOS2 Module 36/2	single-ended	2.5 MHz	300 000
500	EPOS2 24/5		100 kHz	12 000
500	EPOS2 24/2		500 kHz	60 000
1000	EPOS2 70/10	differential	5 MHz	300 000
1000	EPOS2 70/10	single-ended	2.5 MHz	150 000
1000	EPOS2 50/5	differential	5 MHz	300 000
1000	EPOS2 50/5	single-ended	2.5 MHz	150 000
1000	EPOS2 Module 36/2	differential	5 MHz	300 000
1000	EPOS2 Module 36/2	single-ended	2.5 MHz	150 000
1000	EPOS2 24/5	differential	100 kHz	6 000
1000	EPOS2 24/2	differential	500 kHz	30 000
5000	EPOS2 70/10	differential	5 MHz	60 000
5000	EPOS2 70/10	single-ended	2.5 MHz	30 000
5000	EPOS2 50/5	differential	5 MHz	60 000
5000	EPOS2 50/5	single-ended	2.5 MHz	30 000
5000	EPOS2 Module 36/2	differential	5 MHz	60 000
5000	EPOS2 Module 36/2	single-ended	2.5 MHz	30 000
5000	EPOS2 24/5		100 kHz	1 200
5000	EPOS2 24/2		500 kHz	3 000
Limitations: – EC motor, sinusoidal commutation: max. 25 000 rpm – EC motor, block commutation: max. 100 000 rpm				

Table 4-57 Master Encoder Mode – Limiting Factors

Note
Higher velocities can be reached by increasing the scaling factor >1. Thereby consider applicable restrictions (➔“Best Practice” on page 4-61).

5 Step/Direction Mode

5.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

Alternatively, EPOS2 can also be commanded by digital position values. Used are either an incremental encoder (Master Encoder Mode) for setting the values of the device, or PLC-generating step pulses (Step/Direction Mode) can be used to command the device. Inputs and outputs can easily be configured using the «Configuration Wizard» and may be changed online via CANopen or serial bus.

5.1.1 Objective

In «Step/Direction Mode», the motor axis follows a digital signal step-by-step. This mode can replace stepper motors. It can also be used to control the EPOS2 by a PLC without CAN interface.

The present Application Note explains structure, functionality and use of the operation mode «Step/Direction Mode» and features “in practice examples” suitable for daily use.

Contents

5.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	
EPOS2 50/5	347717	2110h or higher	
EPOS2 Module 36/2	360665	2110h or higher	
EPOS2 24/5	367676	2110h or higher	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	

Table 5-58 Step/Direction Mode – covered Hardware and required Documents

5.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 5-59 Step/Direction Mode – recommended Tools

5.2 System Structure

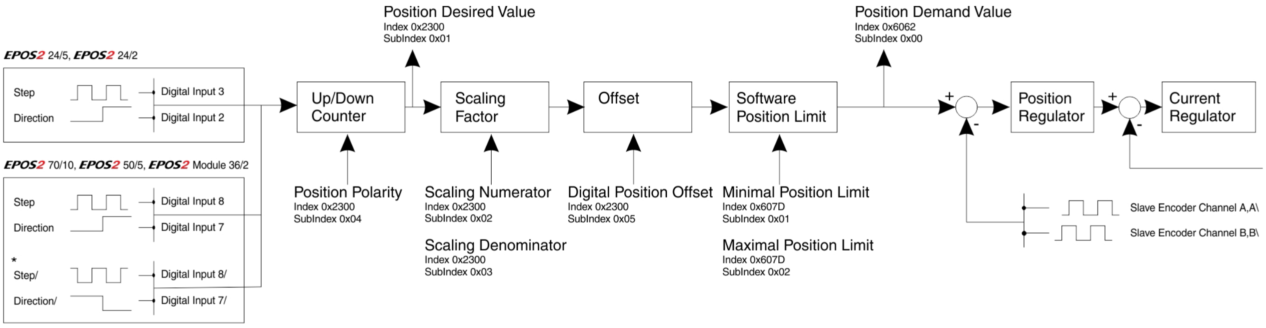


Figure 5-49 Step/Direction Mode – System Structure

Up/Down Counter

EPOS2 70/10, EPOS2 50/5 & EPOS2 Module 36/2

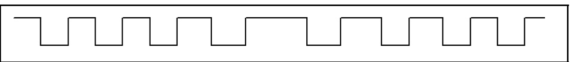
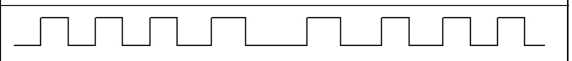
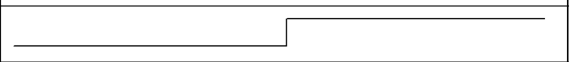
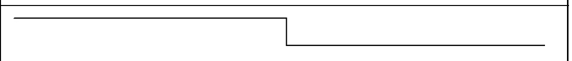
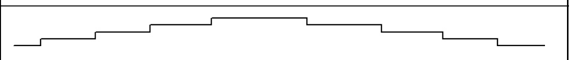
Step	Digital Input 8	
Step\	Digital Input 8\	
Direction	Digital Input 7	
Direction\	Digital Input 7\	
Digital Position Desired Value (Polarity = 0)		

Table 5-60 Up/Down Counter – EPOS2 70/10, EPOS2 50/5 & EPOS2 Module 36/2

EPOS2 24/5 & EPOS2 24/2

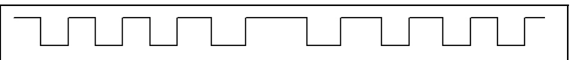
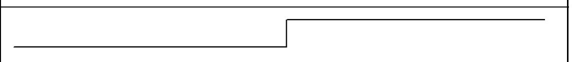
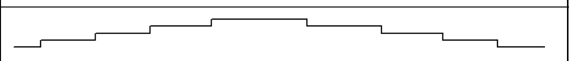
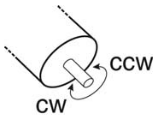
Step	Digital Input 3	
Direction	Digital Input 2	
Digital Position Desired Value (Polarity = 0)		

Table 5-61 Up/Down Counter – EPOS2 24/5 & EPOS2 24/2

Value	EPOS2 70/10 EPOS2 50/5	EPOS2 Module 36/2	EPOS2 24/5	EPOS2 24/2
Input Voltage	0…5 VDC	0…5 VDC	0…24 VDC	0…24 VDC
Max. Input Voltage	−12…+12 VDC	−24…+24 VDC	−30…+30 VDC	−30…+30 VDC
Logic 0	typically <1.0 V	typically <0.8 V	typically <1.5 V	typical <0.7 V
Logic 1	typically >2.4 V	typically >2.0 V	typically >3.0 V	typical >2.4 V
Max. Input Frequency	5 MHz (differential) 2.5 MHz (single-ended)	5 MHz (differential) 2.5 MHz (single-ended)	100 kHz (3.3…5.0 V)	300 kHz (3.3…5.0 V)

Table 5-62 Step/Direction Mode – Hardware Description (Digital Inputs)

Definition of Direction of Rotation



As seen towards motor output flange, definition is as follows:
Direction Input Low: CCW
Direction Input High: CW

Input Parameter

Name	Index	Sub-index	Description
Digital Position Scaling Numerator	0x2300	0x02	Numerator of the scaling factor. Can be used for electronic gearing or to reduce to input frequency.
Digital Position Scaling Denominator	0x2300	0x03	Denominator of the scaling factor. Can be used for electronic gearing or to reduce to input frequency.
Digital Position Polarity	0x2300	0x04	Polarity of the direction input. The direction can be changed (0 = positive, 1 = negative).
Digital Position Offset	0x2300	0x05	Gives a dynamic displacement in reference to the encoder’s desired position.
Minimum Position Limit	0x607D	0x01	Defines the negative position limit for the position demand value.
Maximum Position Limit	0x607D	0x02	Defines the positive position limit for the position demand value.
Maximum ProfileVelocity	0x607F	0x00	This value is used as velocity limit in a position (or velocity) profile mode.
Maximum Acceleration	0x60C5	0x00	Allows to limit the acceleration to prevent mechanical damages. Represents the limit of the other acceleration/deceleration objects.

Table 5-63 Step/Direction Mode – Input Parameter

Output Parameter

Name	Index	Sub-index	Description
Digital Position Desired Value	0x2300	0x01	Counter value of the up/down counter. Serves as base for the scaling and limiting functions.
Position Demand Value	0x6062	0x00	The Step/Direction Mode’s output after scaling and limiting. It is the setting value for the position regulator.

Table 5-64 Step/Direction Mode – Output Parameter

Best Practice

- *Use a scaling factor ≤1 for better behavior. Due to the fact that no interpolation is implemented, movements with factors >1 will result in bigger position jumps, thus producing current peaks.*
- *Switch off software position limitation and set maximum /minimum position limits to INT32_MAX, respectively to INT32_MIN!*

5.3 Configuration

5.3.1 Step 1: System Configuration

Complete standard system configuration (Startup Wizard) in «EPOS Studio» (➔separate document «Getting Started» of respective hardware. Thereby observe following topics:

- Minimum External Wiring
- Communication Setting
- Motor Type
- Motor Pole Pair
- Motor Data
- Position Sensor Type
- Position Regulation

Figure 5-50 Startup Wizard

5.3.2 Step 2: Regulation Tuning

In Master Encoder Mode, current regulator and position regulator must be tuned. Speed regulator will not be used (➔separate document «Getting Started» of respective hardware).

Best Practice
- *Use Profile Position Mode to test regulator behavior!*
- *Use Position Mode for small steps, only!*

- Current Regulator (Current Step)
- Position Regulator (Profile Position Step)

Figure 5-51 Regulation Tuning

5.3.3 Step 3: I/O Configuration and Wiring

1) Perform wiring:

Hardware	From	To
EPOS2 70/10 EPOS2 50/5 EPOS2 Module 36/2	Step	Digital Input 8, 8\
	Direction	Digital Input 7, 7\
EPOS2 24/5 EPOS2 24/2	Step	Digital Input 3
	Direction	Digital Input 2

Table 5-65 Step/Direction Mode – Wiring

2) Start I/O Configuration Wizard to configure I/Os.

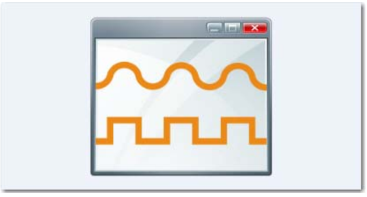


Figure 5-52 Configuration Wizard

3) Configure inputs:

Hardware	Configure…	…as…
EPOS2 70/10 EPOS2 50/5 EPOS2 Module 36/2	Digital Input 7	General Purpose A
	Digital Input 8	General Purpose B
	any available Digital Input	Enable *1)
	any available Digital Output	Ready *2)
EPOS2 24/5 EPOS2 24/2	Digital Input 2	General Purpose A
	Digital Input 3	General Purpose B
	any available Digital Input	Enable *1)
	any available Digital Output	Ready *2)
Remarks: *1) In order to clear a fault condition, the device must be reset. Set input “Enable” to active. *2) Output “Ready” can be used to report a fault condition.		

Table 5-66 Configuration of Inputs

5.3.4 Step 4: Step/Direction Mode

Activate and configure Step/Direction Mode using «EPOS Studio».

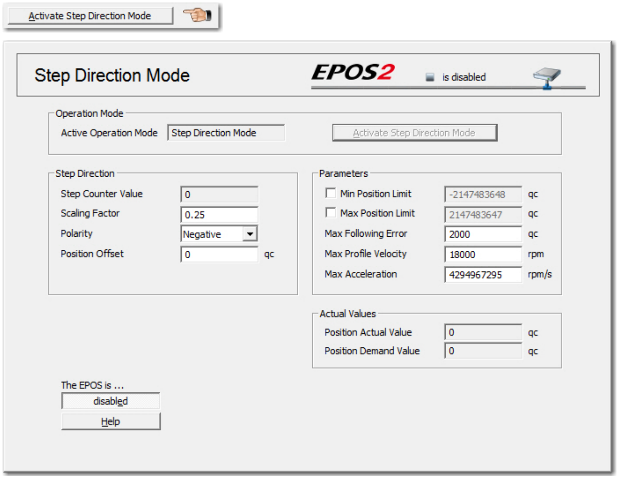


Figure 5-53 Step/Direction Mode – Configuration

5.3.5 Step 5: Save all Parameters

1) Click right on used node (Navigation Window -> Workspace or Communication).
2) Click menu item ¤Save All Parameter¤.

5.4 Application Examples

Typical applications for the Step/Direction Mode are single or multiple axes systems commanded and controlled by digital I/Os, such as stepper motors.

- During the process, no serial interface will be necessary. The device can entirely be controlled by digital inputs and outputs.
- An interface (RS232, USB or CAN bus) is only necessary for configuration.
- The device is enabled by a digital input, a digital output indicates whether the device is ready (no error) or not.
- Velocity or position are commanded by the digital inputs “Step” and “Direction”.

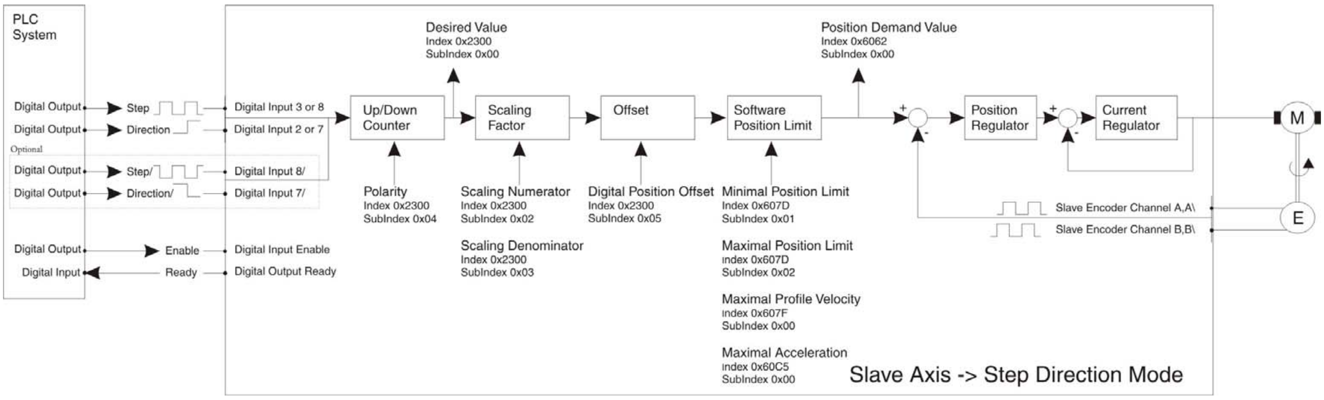


Figure 5-54 Step/Direction Mode – Application Example: Slave Axis System

Calculation of Input Frequency / Velocity of Slave Axis

The velocity of the slave axis is defined by the input frequency of the step input and the scaling factor.

$$StepInputFrequency = Velocity \cdot \frac{4 \cdot EncRes}{60} \cdot \frac{ScalingDenominator}{ScalingNumerator}$$

$$Velocity = StepInputFrequency \cdot \frac{60}{4 \cdot EncRes} \cdot Polarity[1,-1] \cdot \frac{ScalingNumerator}{ScalingDenominator}$$

$EncRes$ [pulses per turn]

$StepInputFrequency$ [Hz]

$Velocity$ [rpm]

Limiting Factors

Maximal permitted Motor Speed
Below figures represent theoretical achievable speeds. For applicable maximum permissible speed of the employed motor ➔catalog motor data!

The primary limiting factor is the step signal's input frequency. Below table shows the maximum velocity of the slave axis assuming a scaling factor of 1. To command higher velocities, the scaling factor can be used to reduce the step input's input frequency.

Encoder [pulse/turn]	Max. Step Input Frequency			Max. Velocity [rpm] (Scaling Factor 1)
500	EPOS2 70/10	differential	5 MHz	150 000
		single-ended	2.5 MHz	75 000
	EPOS2 50/5	differential	5 MHz	150 000
		single-ended	2.5 MHz	75 000
	EPOS2 Module 36/2	differential	5 MHz	150 000
		single-ended	2.5 MHz	75 000
	EPOS2 24/5		100 kHz	3 000
	EPOS2 24/2		500 kHz	15 000
1000	EPOS2 70/10	differential	5 MHz	75 000
		single-ended	2.5 MHz	37 500
	EPOS2 50/5	differential	5 MHz	75 000
		single-ended	2.5 MHz	37 500
	EPOS2 Module 36/2	differential	5 MHz	75 000
		single-ended	2.5 MHz	37 500
	EPOS2 24/5	differential	100 kHz	1 500
	EPOS2 24/2	differential	500 kHz	7 500
5000	EPOS2 70/10	differential	5 MHz	15 000
		single-ended	2.5 MHz	7 500
	EPOS2 50/5	differential	5 MHz	15 000
		single-ended	2.5 MHz	7 500
	EPOS2 Module 36/2	differential	5 MHz	15 000
		single-ended	2.5 MHz	7 500
	EPOS2 24/5		100 kHz	300
	EPOS2 24/2		500 kHz	1 500
Limitations: – EC motor, sinusoidal commutation: max. 25 000 rpm – EC motor, block commutation: max. 100 000 rpm				

Table 5-67 Step/Direction Mode – Limiting Factors

Note
Higher velocities can be reached by increasing the scaling factor >1. Thereby consider applicable restrictions (➔"Best Practice" on page 5-69).

6 Interpolated Position Mode

6.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

For fast communication with several EPOS devices, use the CANopen protocol. The individual devices of a network are commanded by a CANopen master.

6.1.1 Objective

«Interpolated Position Mode» is used to control multiply coordinated axes or a single axis with the need for time interpolation of setpoint data. The trajectory is calculated by the CANopen master and passed on to the controller's interpolated position buffer as a set of points. The controller then reads the points from the buffer and performs linear or cubic interpolation between them.

The present Application Note explains structure, functionality and use of the operation mode «Interpolated Position Mode» and features "in practice examples" suitable for daily use.

Contents

6.1.2 Scope

Hardware	Order #	Firmware Version	Reference (➔page 1-12)
EPOS2		2101h	Firmware Specification Communication Guide (➔[6])
EPOS2 70/10	375711	2120h or higher	
EPOS2 50/5	347717	2110h or higher	
EPOS2 Module 36/2	360665	2110h or higher	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	
CANopen Network			CiA 301 V4.2 (➔[1]) CiA 402 V3.0 (➔[2])

Table 6-68 Interpolated Position Mode – covered Hardware and required Documents

6.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 6-69 Interpolated Position Mode – recommended Tools

6.2 In Detail

6.2.1 Introductory Analogy

Let us assume: In a company, a department manager must convert the department goals into clear tasks for his coworkers. It must be considered that the individual tasks oftentimes stand to each other in close interdependency. Thus, the department manager will gladly count on capable coworkers, being able to solve their tasks even on basis on just substantial data. For the solution's quality, it is in particular important that it…

a) is factually correct; i.e. it will not require further checks,

b) will be finished in time and

c) was reached efficiently.

The functionality «Interpolated Position Mode» values up the positioning controller EPOS2 to such a "capable coworker" in a superordinate drive system. Following, the thesis' description:

In a drive system, normally several axes must be moved according to the guidelines of a central controller. This can take place in the way that each local axis controller receives the next target position in real time – in time and at the same time to each sampling instance. This strategy has the advantage that the local controllers need only little intelligence. However, the central controller must compute target positions for every sampling interval and must communicate the data to every local controller in real time.

As to above analogy…

- it would be favorable if only few, but substantial points of the driving profiles would be considered,
- it would be desirable if the corresponding data could be transmitted to the local controller not necessarily at the same time, but rather in time.

Both goals can be reached by interpolation and data buffering.

First, the central controller decides which points of the local trajectories are substantial for a synchronized total movement. Then, each relevant point of the local trajectories is supplemented with the corresponding velocity and time – i.e. triplicates of the kind (position, velocity, time = PVT) are formed. These triplicates are then transferred to the associated axis controllers, in time. Each local controller possesses a buffer to receive these data. EPOS2's buffer covers 64 locations for triplicates. The data transfer to the EPOS2 is in time as long as the buffer contains 1 to 64 new triplicates.

In EPOS2, local position regulation is sampled with a rate of 1 kHz. Thus, requiring 1000 target positions per second in real time. These target positions are computed in EPOS2 by means of interpolation. Each triplicate forms a base point with the abscissa time and the two ordinates position and velocity. Therefore, two triplicates deliver two abscissas and four corresponding ordinates, permitting an interpolation polynomial of third order unambiguously computed between the two base points. The computation, as well as the evaluation of the polynomial in the local sampling clock, take place on basis of simple arithmetic and are efficiently executed by the EPOS2.

The endpoint of the polynomial [n] forms the starting point of the polynomial [n+1]. Therefore, it is sufficient to indicate only the relative time in a data triplicate (i.e. the length of the time interval). In fact, with the EPOS2, the time distance of two base points can be selected between 1 ms and 255 ms. This interval length can be adapted by the central controller to realize the desired total movement.

With the goal of all controllers within the drive system referring to the same time base, the central controller initiates periodically a time check. This time synchronization takes place with the EPOS2 via the "SYNC Time Stamp Mechanism".

Finally, Interpolated Position Mode can be qualified as follows: The resulting smooth driving profiles, as well as the close temporal synchronization allow to superpose several high-dynamic single movements to a precise total movement in a drive system.

6.2.2 General Description

The Interpolated Position Mode described in the CANopen specification CiA 402 V3.0 is a general case. The objects are well-specified or a linear interpolation (PT). The interpolation type can also be extended by manufacturer-specific algorithms (selectable by «Interpolation Submode Selection», Object 0x60C0).

6.2.3 Spline Interpolation

For the Interpolated Position Mode, the interpolation type is a cubic spline interpolation. The higher-level trajectory planner sends a set of interpolation points by PVT reference point. Each PVT reference point contains information on position, velocity and time of a profile segment end point. The trajectory generator of the drive performs a third order interpolation between the actual and the next reference point.

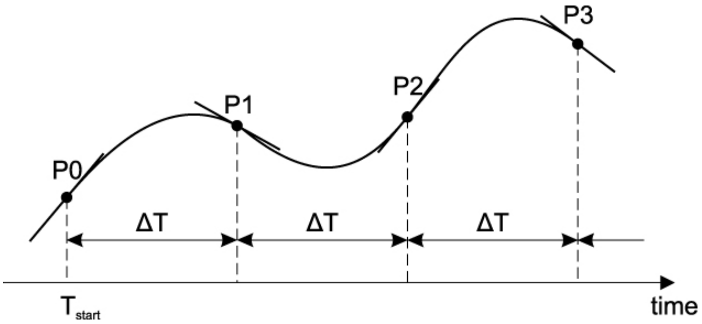


Figure 6-55 Interpolated Position Mode – PVT Principle

From two successive PVT reference points, the interpolation parameters a, b, c and d can be calculated:

```
d = P[t0] = P[n]
c = V[t0] = V[n]
b = T-2[n] * (3 * (P[n] - P[n-1]) + T[n] * (V[n] + 2 * V[n-1]))
a = T-3[n] * (2 * (P[n] - P[n-1]) + T[n] * (V[n] + V[n-1]))
```

The interpolated values for position, velocity and (possibly also) acceleration will be calculated as follows:

```
P(t) = a * (t - t0)3 + b * (t - t0)2 + c * (t - t0) + d
V(t) = 3a * (t - t0)2 + 2b * (t - t0) + c
A(t) = 6a * (t - t0) + 2b
```

t_0: time of interpolation segment end (→ in this calculation t_0 is greater then t!)

It is not mandatory that the time intervals are identical.

6.2.4 SYNC Time Stamp Mechanism

Can be used to synchronize the motion clock of the drive with a master clock in the network.

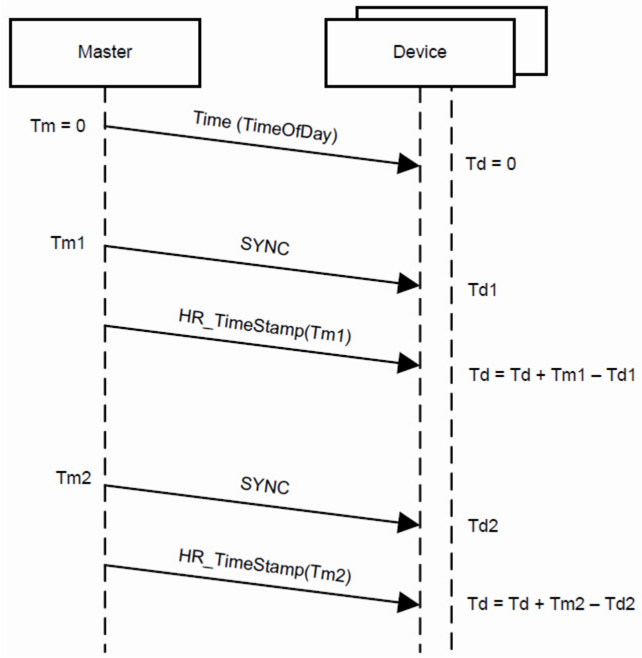


Figure 6-56 Interpolated Position Mode – Clock Synchronization

The synchronisation method is similar to IEEE 1588 and uses the CANopen CiA 301 SYNC Service (COB-Id 0x80) and ➔“High Resolution Time Stamp” on page 6-83.

The SYNC Frame will be transmitted periodically by the SYNC master. The exact transmitting time (Tm1) may be stored by latching an internal 1 µs timer. The reception time (Td1) of the SYNC message will be stored by latching the device-internal motion clock timer. As a follow-up, the measured transmitting time (Tm1) will be sent to the drive using the High Resolution Time Stamp. The device then adjusts its internal motion clock time in relation to the time latched in the last SYNC.

By sending a CANopen CiA 301 TIME Service (by default COB-Id 0x100, or defined as to ➔“COB-ID Time Stamp Object” on page 6-83), the device-internal motion clock timer can be reset to “0”.

6.3 IPM Implementation by maxon

The Interpolated Position Mode is implemented in the EPOS2 as an additional operational mode (operating mode 7 as specified in CiA 402 V3.0).

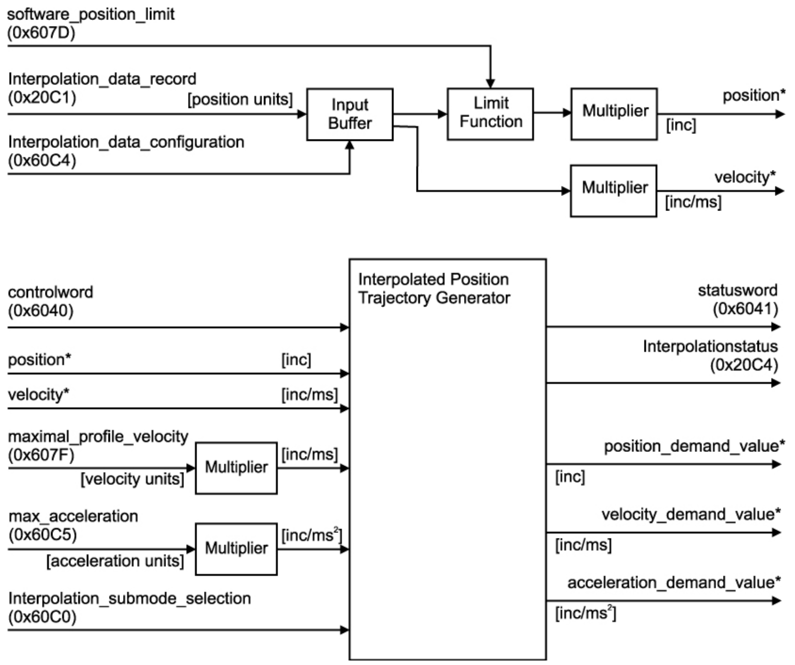


Figure 6-57 Interpolated Position Mode – Interpolation Controller

6.3.1 Interpolated Position Data Buffer

PVT reference points will be sent in a manufacturer-specific 64 bit data record of a complex data structure to a FIFO object.

6.3.1.1 Definition of complex Data Structure 0x0040

MSB		LSB
Time (unsigned8)	Velocity (signed24)	Position (signed32)

Table 6-70 Interpolated Position Mode – IPM Data Buffer Structure

6.3.1.2 Structure of the FIFO

The FIFO will be implemented by a circular buffer with the length of 64 entries

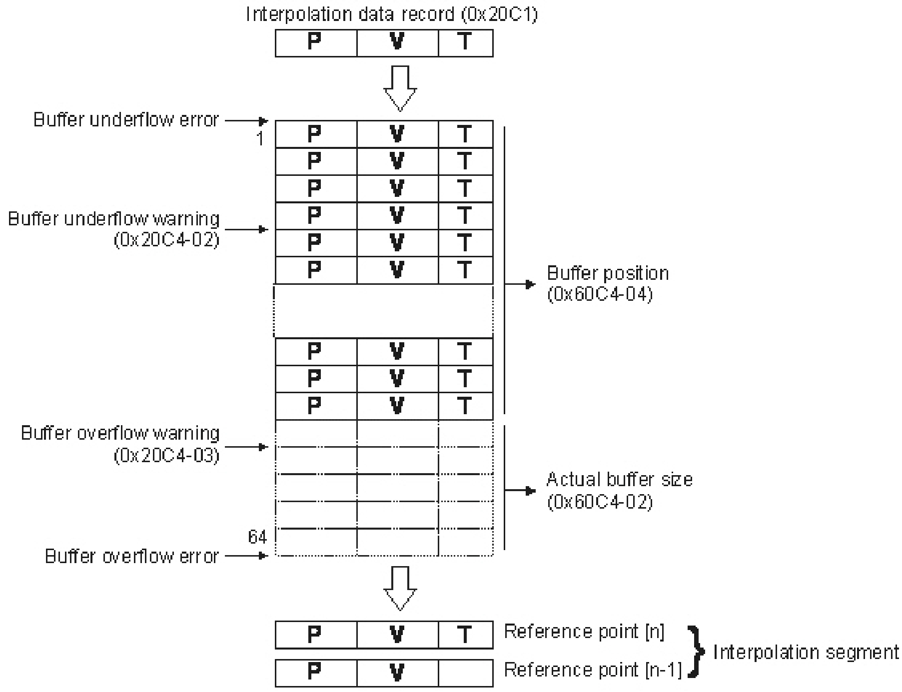


Figure 6-58 Interpolated Position Mode – FIFO Organization

6.3.2 Interpolated Position Mode FSA

The interpolated position finite state automaton is a sub FSA of the Operation enable state.

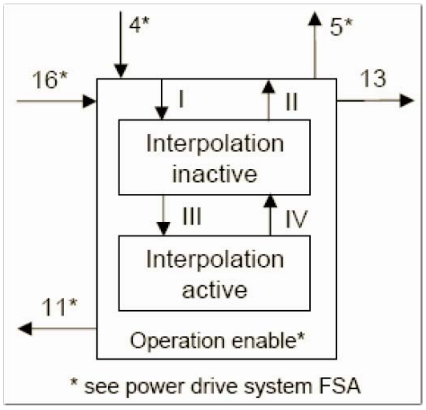


Figure 6-59 Interpolated Position Mode – FSA

FSA State	Function
Interpolation inactive	The drive device accepts input data and buffers it for interpolation calculations, but does not move the axis.
Interpolation active	The drive device accepts input data and moves the axis.

Table 6-71 Interpolated Position Mode – FSA States and supported Functions

Transition	Event		Action
I	ip mode selected (➔object 6060h, page 6-90)		clear data buffer
II	ip mode not selected (➔object 6060h, page 6-90)		none
III	enable ip mode:	set Controlword bit 4 to 1	none
IV	disable ip mode:	set Controlword bit 4 to 0 or ip data record with time = 0	none

Table 6-72 Interpolated Position Mode – Transition Events and Actions

6.3.3 Configuration Parameters

Parameter	Index	Description
Interpolation Sub Mode Selection	0x60C0	Indicates the actually chosen interpolation mode.
Interpolation Time Period	0x60C2	Indicates the configured interpolation cycle time.
Interpolation Data Configuration	0x60C4	Provides information on configuration and state of the buffer. It can also be used to clear the buffer.
Software Position Limit	0x607D	Contains the sub-parameters «Minimal Position Limit» and «Maximal Position Limit» that define the absolute position limits or the position demand value. A new target position will be checked against these limits
Position Window	0x6067	Permits definition of a position range around a target position to be regarded as valid. If the drive is within this area for a specified time, the related Statusword control bit 10 «Target reached» is set.
Position Window Time	0x6068	Defines the time or the position window.
Profile Velocity	0x6081	If calculated velocity of the interpolation exceeds this value, a warning bit in Interpolation Buffer Status Word will be set.
Profile Acceleration	0x6083	If calculated acceleration of the interpolation exceeds this value, a warning bit in Interpolation Buffer Status Word will be set.
Maximal Profile Velocity	0x607F	If calculated velocity of the interpolation exceeds this value, an error bit in Interpolation Buffer Status Word will be set and the device will switch to Fault reaction state.
Maximal Acceleration	0x60C5	If calculated acceleration of the interpolation exceeds this value, an error bit in Interpolation Buffer Status Word will be set and the device will switch to Fault reaction state.
Interpolation Status	0x20C4	The Interpolation buffer underflow/overflow warning level is configured in subindex 2 and 3.

Table 6-73 Interpolated Position Mode – Configuration Parameters

6.3.4 Commanding Parameters

Parameter	Index	Description
Controlword	0x6040	The mode will be controlled by a write access to the Controlword's mode-dependent bits.
Interpolation Data Record	0x20C1	Contains a FIFO to feed PVT reference points to the data buffer.

Table 6-74 Interpolated Position Mode – Commanding Parameters

Controlword (Interpolated Position Mode-specific Bits)

Bit 15…9	Bit 8	Bit 7	Bit 6, 5	Bit 4	Bit 3…0
➔FwSpec	Halt	➔FwSpec	reserved (0)	Enable ip mode	➔FwSpec

Table 6-75 Interpolated Position Mode – Controlword

Name	Value	Description
Enable ip mode	0	Interpolated position mode inactive
	1	Interpolated position mode active
Halt	0	Execute instruction of bit 4
	1	Stop axis with profile deceleration

Table 6-76 Interpolated Position Mode – Controlword Bits

6.3.5 Output Parameters

Parameter	Index	Description
Interpolation status	0x20C4	The mode's statusword is placed in subindex 1 of this object.
Statusword	0x6041	Mode state can be observed by Statusword bits.
Position Demand Value	0x6062	The output of the trajectory generator – it is used as input for the position control function.

Table 6-77 Interpolated Position Mode – Output Parameters

Statusword (Interpolated Position Mode-specific Bits)

Bit 15, 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9…0
➔FwSpec	reserved	ip mode active	➔FwSpec	Target reached	➔FwSpec

Table 6-78 Interpolated Position Mode – Statusword

Name	Value	Description
Target reached	0	Halt = 0: Target Position not (yet) reached Halt = 1: Axle decelerates
	1	Halt = 0: Target Position reached Halt = 1: Velocity of axle is 0
ip mode active	0	ip mode inactive
	1	ip mode active

Table 6-79 Interpolated Position Mode – Statusword Bits

6.3.6 Object Description in Detail

6.3.6.1 COB-ID Time Stamp Object

Description

Defines the COB-ID of the Time Stamp Object (TIME). In EPOS2, this value is immutable.

Name	COB-ID Time Stamp Object	
Index	0x1012	
Subindex	0x00	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00000100	
Value Range	0x00000100	0x00000100
PDO Mapping	no	

6.3.6.2 High Resolution Time Stamp

Description

Contains the timestamp of the last received SYNC Object [1us]. The resolution of the device internal motion clock timer depend on the selected CAN bitrate (bit time) e.g. 1 µs at 1 Mbit/s. After a write access to this object, the EPOS2 calculates the difference between the received timestamp and the internal latched timestamp of the SYNC Object. This time difference is used as correction for the IPM time calculations.

Name	High Resolution Time Stamp	
Index	0x1013	
Subindex	0x00	
Type	UNSIGNED32	
Access	RW	
Default Value	–	
Value Range	–	–
PDO Mapping	yes	

6.3.6.3 Interpolation Data Record

Description

Sets PVT reference points in the interpolated position mode in the cubic spline interpolation sub-mode. The position is given absolute in [Position units], typically [qc]), the velocity is given in [Velocity units], typically [rpm]), and the time is given in [ms]. The object structure is defined in ➔“Interpolated Position Data Buffer” on page 6-79.

Remarks

Normally used to feed PVT reference points to the drive while a PVT motion is executing. Therefore the object may be mapped to a RxPDO with transmission type of 255 (asynchronous).

In the Interpolation active state at least two data records have to be in the FIFO. Otherwise a Queue underflow Emergency will be launched and the drive changes to Fault reaction state.

A data record with time = 0 changes the state to Interpolation inactive without any error.

Name	Interpolation Data Record	
Index	0x20C1	
Subindex	0x00	
Type	complex data structure 0x0040	
Access	WO	
Default Value	–	
Value Range	–	–
PDO Mapping	yes	

6.3.6.4 Interpolation Status

Description

Provides access to status information on the IP input data buffer.

Name	Interpolation Status
Index	0x20C4
Number of entries	0x03

Name	Interpolation Buffer Status	
Index	0x20C4	
Subindex	0x01	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–
PDO Mapping	yes	

Bit 15	Bit 14	Bit 13…12	Bit 11…8	Bit 7…4	Bit 3…0
IP Mode active	Buffer enabled	reserved (0)	IPM buffer errors	reserved (0)	IPM buffer warnings

Table 6-80 Interpolation Buffer Status Word

Name	Bit	Value	Description
Underflow Warning	0	0	No buffer underflow warning
		1	Buffer underflow warning level (0x20C4-2) is reached
Overflow Warning	1	0	No buffer overflow warning
		1	Buffer overflow warning level (0x20C4-3) is reached
Velocity Warning	2	0	No profile velocity violation detected
		1	IPM velocity greater than profile velocity (0x6081) detected
Acceleration Warning	3	0	No profile acceleration violation detected
		1	IPM acceleration greater than profile acceleration (0x6083) detected
Underflow Error	8	0	No buffer underflow error
		1	Buffer underflow error (trajectory abort)
Overflow Error	9	0	No buffer overflow error
		1	Buffer overflow error (trajectory abort)
Velocity Error	10	0	No maximal profile velocity error
		1	IPM velocity greater than maximal profile velocity (0x607F) detected
Acceleration Error	11	0	No maximal profile acceleration error
		1	IPM acceleration greater than maximal profile acceleration (0x60C5) detected
Buffer enabled	14	0	Disabled access to the input buffer
		1	Access to the input buffer enabled
IP Mode active	15	0	IP mode inactive (same as bit 12 in statusword)
		1	IP mode active

Table 6-81 Interpolation Buffer Status Bits

Description

Gives the lower signalization level of the data input FIFO. If the filling level is below this border the warning flag (bit 0) in the Interpolation buffer status will be set.

Name	Interpolation Buffer Underflow Warning	
Index	0x20C4	
Subindex	0x02	
Type	UNSIGNED16	
Access	RW	
Default Value	4	
Value Range	0	63
PDO Mapping	no	

Description

Gives the higher signalization level of the data input FIFO. If the filling level is above this border the warning flag (bit 1) in the Interpolation buffer status will be set.

Name	Interpolation Buffer Overflow Warning	
Index	0x20C4	
Subindex	0x03	
Type	UNSIGNED16	
Access	RW	
Default Value	60	
Value Range	1	64
PDO Mapping	no	

6.3.6.5 Interpolation Sub Mode Selection

Description

Indicates the actually chosen interpolation mode.

Name	Interpolation Sub Mode Selection	
Index	0x60C0	
Subindex	0x00	
Type	INTEGER16	
Access	RW	
Default Value	−1	
Value Range	−1	−1
PDO Mapping	no	

Value	Description
−32 768… −2	Manufacturer-specific (reserved)
−1	cubic spline interpolation (PVT)
0	Linear interpolation (not yet implemented)
1…32 767	reserved

Table 6-82 Interpolation Sub Mode Selection – Definition

6.3.6.6 Interpolation Time Period

Description

Indicates the configured interpolation cycle time. The interpolation time period (subindex 0x01) value is given in $10^{\text{interpolation time index}}$ per second. The interpolation time index (subindex 0x02) is dimensionless.

Name	Interpolation Time Period
Index	0x60C2
Number of entries	0x02

Name	Interpolation Time Period Value	
Index	0x60C2	
Subindex	0x01	
Type	UNSIGNED8	
Access	RW	
Default Value	1	
Value Range	1	1
PDO Mapping	no	

Name	Interpolation Time Index	
Index	0x60C2	
Subindex	0x01	
Type	INTEGER8	
Access	RW	
Default Value	−3	
Value Range	−3	−3
PDO Mapping	no	

6.3.6.7 Interpolation Data Configuration

Description

Provides the maximal buffer size and is given in interpolation data records.

Name	Interpolation Data Configuration
Index	0x60C4
Number of entries	0x06

Name	Maximum Buffer Size	
Index	0x60C4	
Subindex	0x01	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	64	64
PDO Mapping	no	

Description

Provides the actual free buffer size and is given in interpolation data records.

Name	Actual Buffer Size	
Index	0x60C4	
Subindex	0x02	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	0	64
PDO Mapping	yes	

Description

The value 0 indicates a FIFO buffer organization.

Name	Buffer Organization	
Index	0x60C4	
Subindex	0x03	
Type	UNSIGNED8	
Access	RW	
Default Value	–	
Value Range	–	–
PDO Mapping	no	

Value	Description
0	FIFO buffer
1	Ring buffer (not supported)
2…255	reserved

Table 6-83 Buffer Organization – Definition

Description

Provides used buffer space and is given in interpolation data records. Writing to this object has no effect.

Name	Buffer Position	
Index	0x60C4	
Subindex	0x04	
Type	UNSIGNED16	
Access	RW	
Default Value	0	
Value Range	0	64
PDO Mapping	no	

Description

Interpolation data record size is 8 bytes.

Name	Size of Data Record	
Index	0x60C4	
Subindex	0x05	
Type	UNSIGNED8	
Access	WO	
Default Value	–	
Value Range	8	8
PDO Mapping	no	

Description

If 0 is written, the data buffer is cleared and the access to it is denied. If 1 is written, the access to the data buffer is enabled.

Related Objects

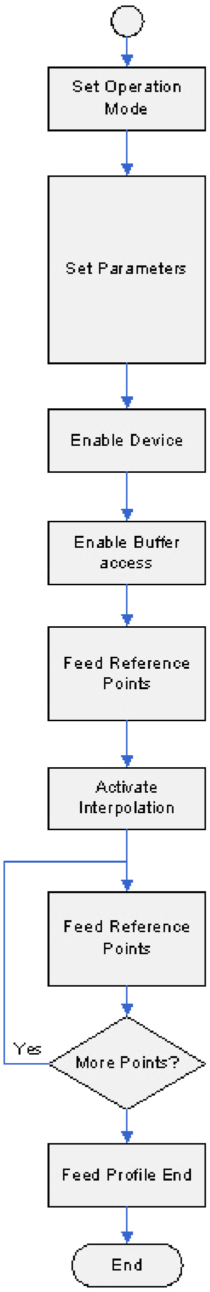
➔“Interpolation Status” on page 6-84

Name	Buffer Clear	
Index	0x60C4	
Subindex	0x06	
Type	UNSIGNED8	
Access	WO	
Default Value	0	
Value Range	0	1
PDO Mapping	no	

Value	Description
0	Clear input buffer (and all data records) access disabled
1	Enable access to the input buffer for the drive functions
2…255	reserved

Table 6-84 Buffer Clear – Definition

6.3.7 Typical IPM Commanding Sequence



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0x07 (Interpolated Position Mode)
Max. Following Error Min. Position Limit Max. Position Limit Max. Profile Velocity Max. Acceleration Profile Velocity Profile Acceleration Quick Stop Deceleration	0x6065-00 0x607D-01 0x607D-02 0x607F-00 0x60C5-00 0x6081-00 0x6083-00 0x6084-00	Application specific [2000 qc] Application specific [−2147483648 qc] Application specific [2147483647 qc] Motor specific [25000 rpm] Application specific [4294967295 rpm/s] Application specific [1000 rpm] Application specific [10000 rpm/s] Application specific [10000 rpm/s]
Controlword (Shutdown) Controlword (SwitchOn)	0x6040-00 0x6040-00	0x0006 0x000F
Buffer Clear	0x60C4-06 0x60C4-06	0x00 (disable and clear buffer) 0x01 (enable buffer)
Interpolation Data Record	0x20C1-00	Reference points (PVT), minimum 2 points!
Controlword (enable ip mode)	0x6040-00	0x001F
if (Interpolation Buffer Status) **do** Interpolation Data Record **until** (Interpolation Buffer Status)	0x20C4-01 0x20C1-00 0x20C4-01	Bit 0 == 1 (Underflow Warning) Reference point (PVT) Bit 1 == 1 (Overflow Warning)
Interpolation Data Record	0x20C1-00	Reference point (PVT) with time = 0

Table 6-85 Interpolated Position Mode – typical Command Sequence

As long as the interpolation is active, feeding of new reference points is the main task. To minimize the communication overhead, it might make sense to map the “Interpolation Data Record” in a (asynchronous) receive PDO. If the “Interpolation Buffer Status” is mapped to an event trigger transmit PDO (possibly along with the Statusword), processing of reference point feeding can easier be implemented.

6.4 Configuration

1) Complete standard system configuration (Startup Wizard) in «EPOS Studio» (➔separate document «Getting Started» of respective hardware.
2) Start CANopen Wizard.
3) Select “Restore Default COB-IDs”.
4) Enter settings for “Receive PDO1”:
   a) Tick “PDO is valid”.
   b) Set Transmission Type to “Asynchronous”.
   c) Click ¤Next¤.

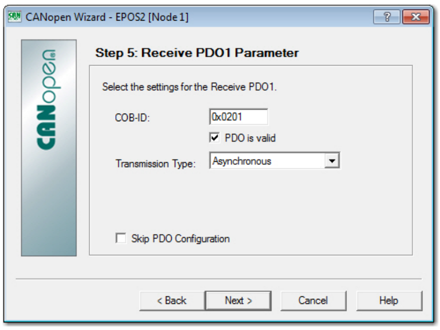


Figure 6-60 CANopen Wizard #5

5) Change Mapping:
   a) Delete all mapped objects.
   b) Select “Interpolation Data Record” from Mappable Objects and add to Mapped Objects No 1 using ¤>>¤.
   c) Click ¤OK¤.

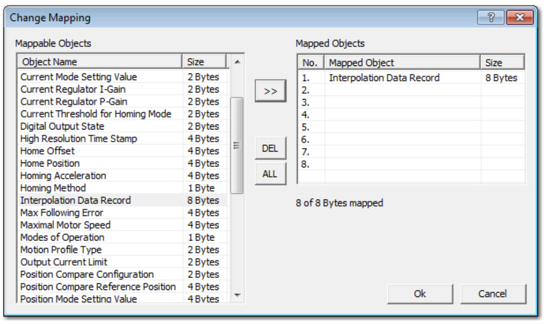


Figure 6-61 Change Mapping Receive PDO1

6) Enter settings for “Transmit PDO1”:
   a) Tick “PDO is valid”.
   b) Set Transmission Type to “Asynchronous”.
   c) Set Inhibit Time (e.g. 5.0 ms).
   d) Click ¤Next¤.

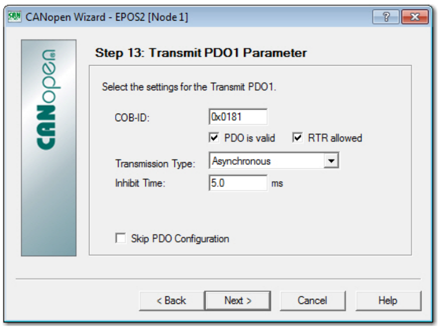


Figure 6-62 CANopen Wizard #13

7) Change Mapping:
   a) Delete all mapped objects.
   b) Select “Interpolation Buffer Status” from Mappable Objects and add to Mapped Objects No 1 using ¤>>¤.
   c) Select “StatusWord” from Mappable Objects and add to Mapped Objects No 2 using ¤>>¤.
   d) Click ¤OK¤.

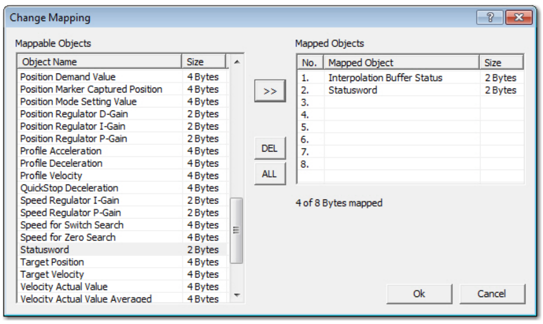


Figure 6-63 Change Mapping Transmit PDO1

8) Complete CANopen Wizard.

6.4.1 Motion Synchronization

Interpolated Position Mode enables the synchronized motion of multiple axes. The movement of a number of slave axes can be synchronized if they all run in IPM, and if they all possess the same time.

To start a number of slave axes synchronously, map the controlword to a synchronous RPDO, then use the mapped controlword to enable interpolation for all axes. There will be no reaction until next SYNC. Then, all drives will enable interpolated motion at once, setting the SYNC arrival time as the path specification's "zero" time.

If the axes have been synchronized by the SYNC Time Stamp Mechanism, the moving axes will run synchronous within an accuracy of microseconds.

If the CAN (SYNC) master is not able to produce the high resolution time stamp, an EPOS2 might be used as clock master. Do so by mapping "High Resolution Time Stamp" object (0x1013) to a synchronous transmit PDO in the "clock master EPOS2". Other EPOS2s in the system must be configured as clock slaves with the "High Resolution Time Stamp" object mapped to an asynchronous receive PDO with identical COB-ID as the clock master's transmit PDO.

Note
The resolution of the EPOS2 internal microsecond timer depends on the CAN bitrate since a CAN controller-internal hardware counter is used as timing reference. This hardware counter will be incremented by the bit time.

6.4.2 Interruption in Case of Error

If a currently running interpolation (index 0x20C4, subindex 0x03 "Interpolation Status" bit 15 "ip mode active" set) will be interrupted by an occurring error, the EPOS2 will react accordingly (i.e. disabling the controller and changing to state switch on disabled).

The interpolation can only be restarted by re-synchronization due to the fact that state "Operation enable" must be entered again, whereby the bit "ip mode active" will be cleared.

••page intentionally left blank••

7 Regulation Tuning

7.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

«Regulation Tuning» is an important attribute of EPOS2. It is a procedure for automatic start-up of all relevant regulation modes, such as current, velocity and/or positioning control. This intelligent tool is easy to handle and substantially increases the use of the positioning control unit.

7.1.1 Objective

The present Application Note explains use of «Regulation Tuning» and features "in practice examples" suitable for daily use.

Contents

7.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	Cable Starting Set Hardware Reference
EPOS2 50/5	347717	2110h or higher	Cable Starting Set Hardware Reference
EPOS2 Module 36/2	360665	2110h or higher	Hardware Reference
EPOS2 24/5	367676	2110h or higher	Cable Starting Set Hardware Reference
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	Cable Starting Set Hardware Reference

Table 7-86 Regulation Tuning – covered Hardware and required Documents

7.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 7-87 Regulation Tuning – recommended Tools

7.2 Regulation Structures

EPOS2 can be interconnected within three essential regulation structures.

7.2.1 Current Control

To provide accurate motion control, given forces and/or torques within the drive system need to be compensated. Hence, EPOS2 offers a current control loop. The current controller is implemented as a PI controller.

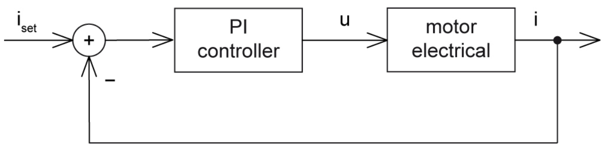


Figure 7-64 Regulation Tuning – Current Control

Current control can be operated either directly as the main regulator, or it serves as subordinated regulator in one of the two following cascade regulation structures.

7.2.2 Velocity Control (with Velocity and Feedforward Acceleration)

Based on the subordinated current control, a velocity control loop can be established. The velocity controller is implemented as a PI controller.

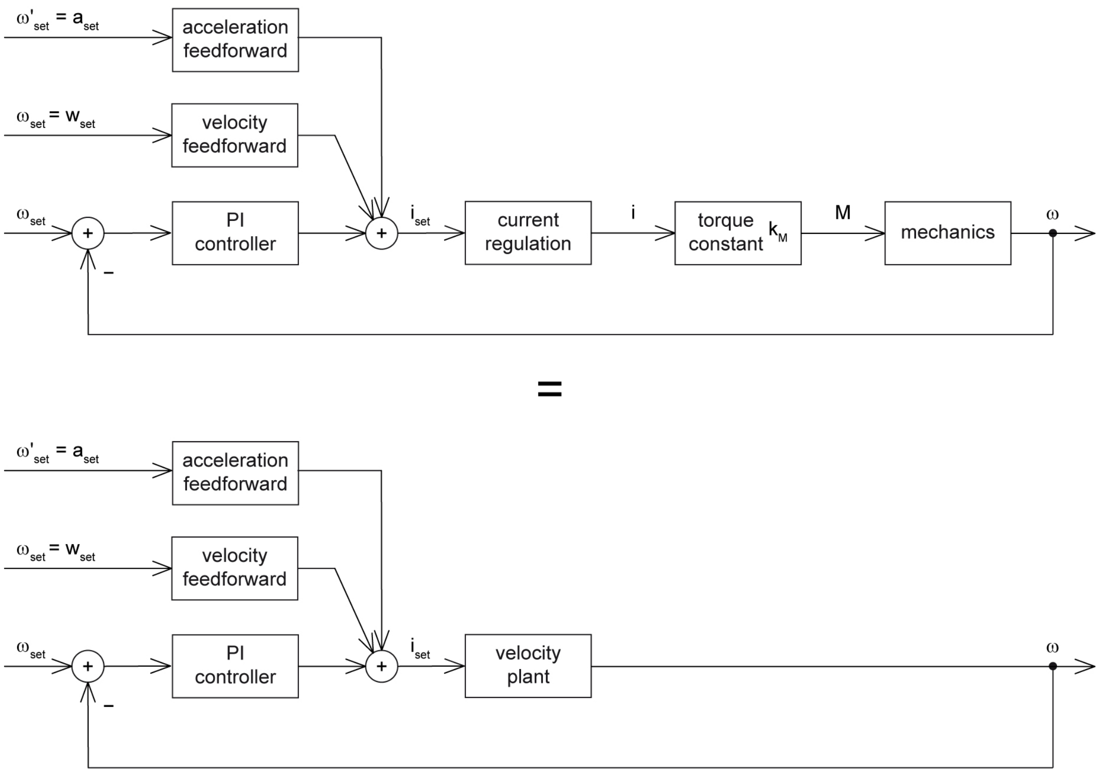


Figure 7-65 Regulation Tuning – Velocity Control

7.2.3 Position Control (with Velocity and Feedforward Acceleration)

Based on the subordinated current control, a position control loop can be established. The position controller is implemented as a PID controller.

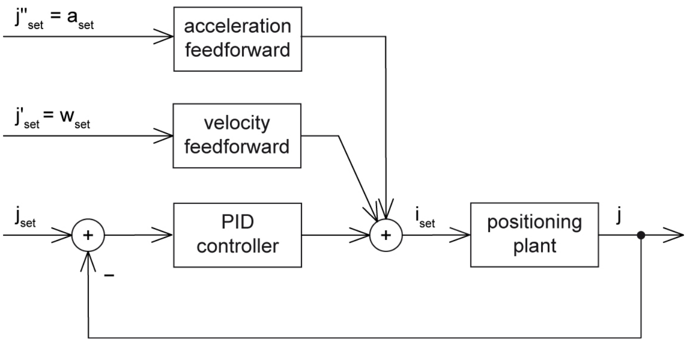


Figure 7-66 Regulation Tuning – Position Control

To improve the reference action of the motion system, position control is supplemented by feedforward control. Velocity feedforward compensates for speed-proportional friction, whereas known inertia can be taken into account by acceleration feedforward.

7.3 Working Principle

«Regulation Tuning» is based on three features:

1) **Identification and modeling** of the plant.
2) **Mapping** model parameters of the plant to derivate controller parameters (PI, PID, feedforward).
3) **Verification** of the resulting regulation structure.

7.3.1 Identification and Modeling

For identification, the plant is activated by a two-point element – positive and negative current of varying amplitudes, which are based on motor parameters – until a stable oscillation of a fixed amplitude is achieved. This experiment is repeated at a different frequency. The characteristics of the oscillations represent substantial properties of the plant.

Hence, the modeling parameters of a simple mathematical model of the plant can be calculated.

7.3.2 Mapping

Now, the model parameters serve for calculation of controller parameters (PI or PID, respectively) and of feedforward velocity and acceleration parameters.

The validity range of the regulation parameters is characterized, among other aspects, by the regulation bandwidth which is determined as well.

7.3.3 Verification

To achieve proper operation with the gained motion control parameters, the system reaction is verified with a motion profile corresponding to the calculated bandwidth.

7.4 Regulation Tuning Wizard

«Regulation Tuning» is a procedure for automated parameterization of the three above mentioned motion controller types (current, velocity and positioning regulation) including position control's feedforward parameters.

For successful Regulation Tuning, correct setup of system parameters in Startup Wizard is essential. Particularly important are…

- Motor data,
- Encoder data, and
- Communication with the PC.

Initiating the "Regulation Tuning Wizard"

1) Complete standard system configuration (Startup Wizard) in «EPOS Studio».
2) Select ¤Wizards¤ and select ¤Regulation Tuning¤.

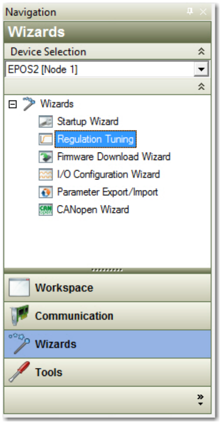


Figure 7-67 Regulation Tuning Wizard

3) Select one of the two modes (for details ➔"Tuning Modes" on page 7-99):

   – ¤Auto Tuning¤

   – ¤Expert Tuning¤

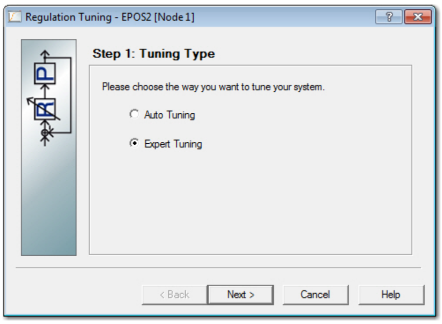


Figure 7-68 Regulation Tuning Mode Selection

7.5 Tuning Modes

7.5.1 Auto Tuning

Auto Tuning is the Regulation Tuning's "very-easy-to-use option". The only thing needed to accomplish automated tuning is to push the start button. A message will inform you that the system will move during the subsequent procedure. Upon confirming the message, Auto Tuning will commence. All required settings are already implemented, so Auto Tuning can parameterize the motion system for most common load cases without further help.

Under certain conditions (strong motor cogging torque, unbalanced friction, low position sensor resolution, etc.) however, or to cover particular requirements (wear, noise or energy optimized operation), Expert Tuning may be used.

7.5.2 Expert Tuning

Expert Tuning offers additional self-describing options for optimum regulation behavior. The following example illustrates tuning using Position Control. Handling of Current Control or Velocity Control however are similar.

Expert Tuning's user interface is divided in four sections:

a) Cascade
b) Identification
c) Parameterization
d) Verification:

Cascade

Provides information on the selected cascade structure.

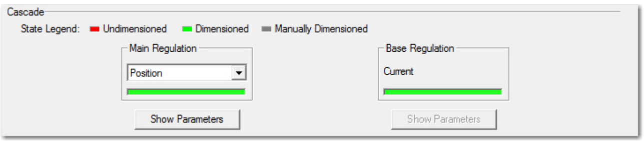


Figure 7-69 Expert Tuning – Cascade

The view is split into two panes; "Main Regulation" and "Base Regulation" (or subordinated regulation). Their respective status is displayed in colored bars:

- Red: Undimensioned – the controller is not yet parameterized.
- Green: Dimensioned – the controller is already parameterized.
- Grey: Manually Dimensioned – the control parameters are being set manually (➔"Manual Tuning" on page 7-101).

Click ¤Show Parameters¤ to view/alter the currently set values.

Velocity control can be viewed and adjusted (in "Main Regulation" window), even if the position was originally defined to be the main controlled variable. However, in order to avoid inconsistencies with the position main regulations, current control cannot be changed. If velocity control's current regulation needs to be optimized, velocity must be defined as Main Regulation variable.

Now, Regulation Tuning is being executed in three steps:

Identification

Figure 7-70 Expert Tuning – Identification

Tick ¤Identify¤ if identification of a new plant is necessary (e.g. if the plant properties have changed). In this case, the status of the corresponding controller, as well as all controllers of higher regulation hierarchy, will change to “Undimensioned” (red).

By adjusting the identification amplitude, nonlinear properties (e.g. Coulomb Friction) can be simulated appropriately and can be considered in the plant model by means of harmonic linearization. However, presetting already offers a good basis for plant identification for most applications.

Parameterization

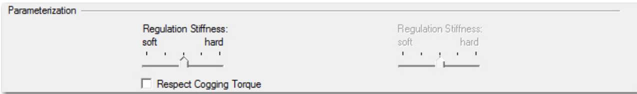


Figure 7-71 Expert Tuning – Parameterization

The calculated controller parameters can be modified to match given requirements by means of sliders:

- “Soft” means: slow regulation behavior, but well dampened.
- “Hard” means: quick regulation behavior, but less dampened.

Tick ¤Respect Cogging Torque¤ to achieve a hard, nevertheless well dampened motion regulation, which brings particular advantages for motors with high cogging torque. In case of unbalanced friction, the regulation behavior can be improved with this adjustment as well.

Verification

The verification of the resulting control system – including feedforward – permits examination of the overall performance. The verification can either take place with a movement profile (which takes bandwidth of the position regulation into account), or a step response. As interesting feature; in addition to the position, the corresponding current is recorded, too.

To zoom the recorded diagrams, crop the “area of interest” and click right.

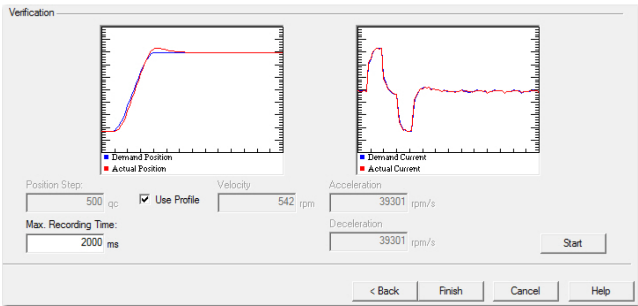


Figure 7-72 Expert Tuning – Verification

The parameters “Position Step”, “Velocity”, “Acceleration” and “Deceleration” are computed automatically. They can be adjusted only if the positioning controller is in state “Manually Dimensioned” (grey).

The parameter “Max. Recording Time” limits the time interval for data acquisition. This can be useful, if details concerning the beginning of the movement profile are of interest.

¤Start¤ launches Expert Tuning. ¤Finish¤ will save the obtained feedback and feedforward parameters in the EPOS2 and make them valid for all operation modes. ¤Cancel¤ will reject the results and returns to the starting situation.

7.5.3 Manual Tuning

In certain conditions, you might wish to change control parameters manually to see how the system reacts without performing automated system identification and modeling.

Also, the manual mode can be used…

- for fine tuning and optimization in very demanding applications, or
- if the outcome of Auto Tuning/Expert Tuning is not satisfactory.

Initiate Manual Tuning by selecting ¤Manually Dimensioned¤ in ¤Show Parameter¤ dialog (➔“Cascade” on page 7-99). As a result, the status will switch to “Manually Dimensioned” (grey), thus neither automated identification nor parameterization will be carried out. In addition, you can define the motion profile (➔“Verification” on page 7-100).

After ticking ¤Identify¤, or if you make any changes (➔“Parameterization” on page 7-100), Manual Tuning is terminated showing status “Undimensioned” (red).

••page intentionally left blank••

8 Device Programming

8.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

8.1.1 Objective

The present Application Note explains typical commanding sequences for different operating modes. The explanations are based on writing/reading commands to access the Object Dictionary. For detailed information on the objects itself ➔separate document «EPOS2 Firmware Specification» (subsequently referred to as “FwSpec”). For detailed information on the command structure ➔«EPOS Studio» (command analyzer).

Contents

8.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	
EPOS2 50/5	347717	2110h or higher	
EPOS2 Module 36/2	360665	2110h or higher	
EPOS2 24/5	367676	2110h or higher	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	

Table 8-88 Device Programming – covered Hardware and required Documents

8.1.3 Tools

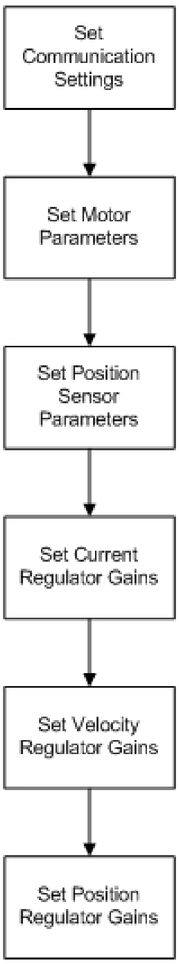
Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 8-89 Device Programming – recommended Tools

8.2 First Step

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

Note
For detailed information on the command structure ➔«EPOS Studio» (command analyzer).



Object Name	Object	User Value [Default Value]
CAN Bitrate RS232 Baudrate	0x2001-00 0x2002-00	User-specific [0] User-specific [3]
Motor Type Continuous Current Limit Pole Pair Number Thermal Time Constant Winding	0x6402-00 0x6410-01 0x6410-03 0x6410-05	Motor-specific [10] Motor-specific [5000] Motor-specific [1] Motor-specific [40]
Encoder Pulse Number Position Sensor Type	0x2210-01 0x2210-02	Sensor-specific [500] Sensor-specific [1]
Current Regulator P-Gain Current Regulator I-Gain	0x60F6-01 0x60F6-02	Motor-specific. Determine optimal parameter using “Regulation Tuning” in «EPOS Studio».
Speed Regulator P-Gain Speed Regulator I-Gain	0x60F9-01 0x60F9-02	Motor-specific. Determine optimal parameter using “Regulation Tuning” in «EPOS Studio».
Position Regulator P-Gain Position Regulator I-Gain Position Regulator D-Gain	0x60FB-01 0x60FB-02 0x60FB-03	Motor-specific. Determine optimal parameter using “Regulation Tuning” in «EPOS Studio».

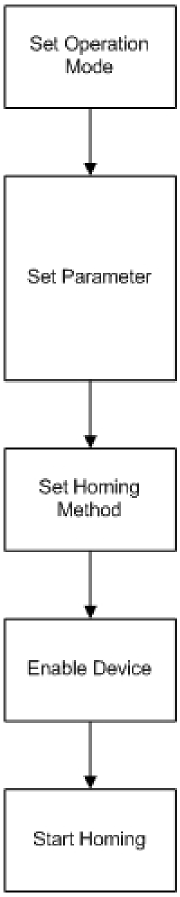
Table 8-90 Device Programming – First Step

8.3 Homing Mode

8.3.1 Start Homing

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

Note
For details on bits to be set for the "Controlword function" (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0x06 (Homing Mode)
Max. Following Error Home Offset Max. Profile Velocity Quick Stop Deceleration Speed for Switch Search Speed for Zero Search Homing Acceleration Current Threshold Homing Mode Home Position	0x6065-00 0x607C-00 0x607F-00 0x6085-00 0x6099-01 0x6099-02 0x609A-00 0x2080-00 0x2081-00	User-specific [2000 qc] User-specific [0 qc] Motor-specific [25000 rpm] User-specific [10000 rpm/s] User-specific [100 rpm] User-specific [10 rpm] User-specific [1000 rpm/s] User-specific [500 mA] User-specific [0 qc]
Homing Method	0x6098-00	Select Homing Method (➔FwSpec)
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Controlword (Start homing mode)	0x6040-00	0x001F

Table 8-91 Device Programming – Homing Mode (Start)

8.3.2 Read Status

Object Name	Object	User Value [Default Value]
Statusword (Target reached / Homing attained)	0x6041-00	Home position is reached if bit 10 / bit 12 is set to 1.

Table 8-92 Device Programming – Homing Mode (Read)

8.3.3 Stop Positioning

Note
For details on bits to be set for the "Controlword function" (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.

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

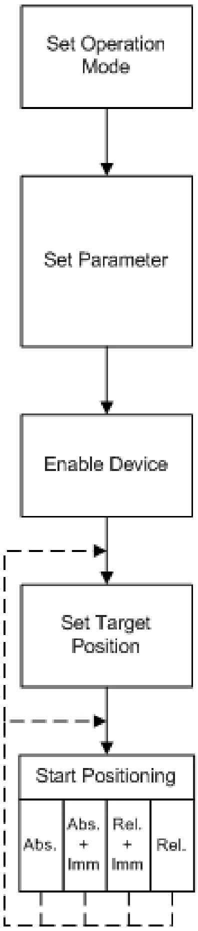
Table 8-93 Device Programming – Homing Mode (Stop)

8.4 Profile Position Mode

8.4.1 Set Position

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

Note
For details on bits to be set for the “Controlword function” (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0x01 (Profile Position Mode)
Max. Following Error Min. Position Limit Max. Position Limit Max. Profile Velocity Profile Velocity Profile Acceleration Profile Deceleration Quick Stop Deceleration Motion Profile Type	0x6065-00 0x607D-01 0x607D-02 0x607F-00 0x6081-00 0x6083-00 0x6084-00 0x6085-00 0x6086-00	User-specific [2000 qc] User-specific [−2147483648 qc] User-specific [2147483647 qc] Motor-specific [25000 rpm] Desired Velocity [1000 rpm] User-specific [10000 rpm/s] User-specific [10000 rpm/s] User-specific [10000 rpm/s] User-specific [0]
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Target Position	0x607A-00	Desired Position [qc]
Controlword (absolute pos.) or Controlword (absolute pos., start immediately) or Controlword (relative pos., start immediately) or Controlword (relative positioning)	0x6040-00 0x6040-00 0x6040-00 0x6040-00	0x001F 0x003F 0x007F 0x005F

Table 8-94 Device Programming – Profile Position Mode (Set)

8.4.2 Read Status

Object Name	Object	User Value [Default Value]
Statusword (Target reached)	0x6041-00	The axis is at target position if bit 10 is set.

Table 8-95 Device Programming – Profile Position Mode (Read)

8.4.3 Stop Positioning

Object Name	Object	User Value [Default Value]
Controlword (Stop positioning) or Controlword (Quick stop)	0x6040-00 0x6040-00	0x010F 0x000B

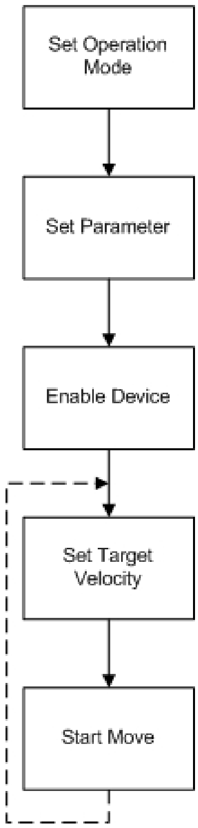
Table 8-96 Device Programming – Profile Position Mode (Stop)

8.5 Profile Velocity Mode

8.5.1 Start Velocity

Motor shaft rotates with a certain speed with velocity profile.

Note
For details on bits to be set for the “Controlword function” (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0x03 (Profile Velocity Mode)
Max. Profile Velocity Profile Acceleration Profile Deceleration Quick Stop Deceleration Motion Profile Type	0x607F-00 0x6083-00 0x6084-00 0x6085-00 0x6086-00	Motor-specific [25000 rpm] User-specific [10000 rpm/s] User-specific [10000 rpm/s] User-specific [10000 rpm/s] User-specific [0]
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Target Velocity	0x60FF-00	Velocity for movement [rpm]
Controlword	0x6040-00	0x000F

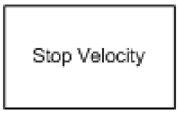
Table 8-97 Device Programming – Profile Velocity Mode (Start)

8.5.2 Read Status

Object Name	Object	User Value [Default Value]
Statusword (Target velocity reached)	0x6041-00	Target velocity is reached if bit 10 is set.

Table 8-98 Device Programming – Profile Velocity Mode (Read)

8.5.3 Stop Velocity



Object Name	Object	User Value [Default Value]
Controlword (Halt Profile Velocity Mode) or Controlword (Quick stop)	0x6040-00 0x6040-00	0x010F 0x000B

Table 8-99 Device Programming – Profile Velocity Mode (Stop)

8.6 Interpolated Position Mode (PVT)

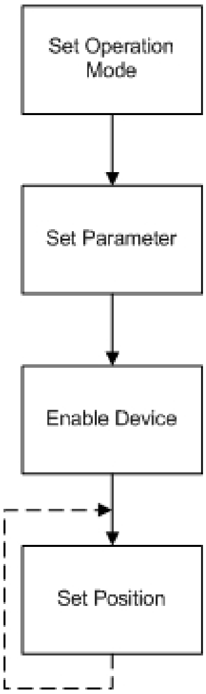
For detailed information ➔chapter “6 Interpolated Position Mode” on page 6-75.

8.7 Position Mode

8.7.1 Set Position

The axis moves to the new absolute position with maximum acceleration and maximum velocity without particular trajectory. If the difference between actual and new position is greater than “Max Following Error”, an emergency procedure will be launched.

Note
For details on bits to be set for the “Controlword function” (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0xFF (Position Mode)
Max. Following Error Min. Position Limit Max. Position Limit Max. Profile Velocity Max. Acceleration	0x6065-00 0x607D-01 0x607D-02 0x607F-00 0x60C5-00	User-specific [2000 qc] User-specific [−2147483648 qc] User-specific [2147483647 qc] Motor-specific User-specific [4294967295]
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Position Mode Setting Value	0x2062-00	New Position [qc]

Table 8-100 Device Programming – Position Mode (Set)

8.7.2 Stop Positioning

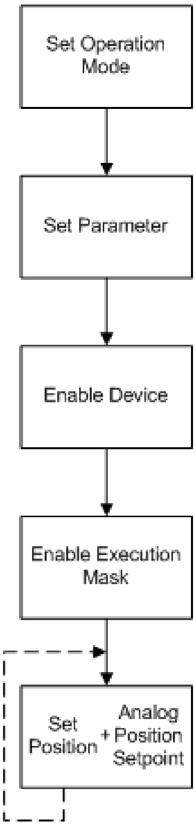
Object Name	Object	User Value [Default Value]
Controlword (Quick stop)	0x6040-00	0x000B

Table 8-101 Device Programming – Position Mode (Stop)

8.7.3 Set Position with analog Setpoint

For details ➔FwSpec, chapter “Position Mode”.

Note
For details on bits to be set for the “Controlword function” (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0xFF (Position Mode)
Max. Following Error Min. Position Limit Max. Position Limit Max. Profile Velocity Max. Acceleration Configuration of Analog Input x Analog Position Setpoint Scaling Analog Position Setpoint Offset	0x6065-00 0x607D-01 0x607D-02 0x607F-00 0x60C5-00 0x207B-0x 0x2303-01 0x2303-02	User-specific [2000 qc] User-specific [−2147483648 qc] User-specific [2147483647 qc] Motor-specific User-specific [4294967295] 0x02 User-specific [0] User-specific [0]
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Analog Input Functionalities Execution Mask	0x207D-00	0x02
Position Mode Setting Value Analog Position Setpoint	0x2062-00 0x2303-04	New Position [qc] Calculated value

Table 8-102 Device Programming – Position Mode (Set, analog)

8.7.4 Stop Positioning from analog Setpoint

Object Name	Object	User Value [Default Value]
Controlword (Quick stop)	0x6040-00	0x000B

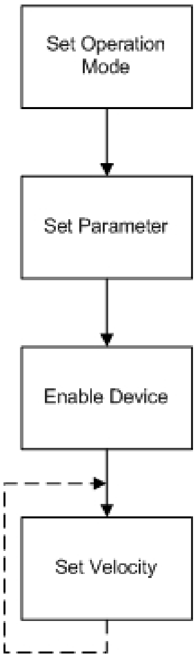
Table 8-103 Device Programming – Position Mode (Stop, analog)

8.8 Velocity Mode

8.8.1 Set Velocity

Motor shaft runs with a certain speed with maximum acceleration.

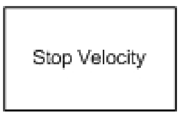
Note
For details on bits to be set for the "Controlword function" (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0xFE (Velocity Mode)
Max. Profile Velocity Max. Acceleration	0x607F-00 0x60C5-00	Motor-specific User-specific [4294967295]
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Velocity Mode Setting Value	0x206B-00	Velocity for movement [rpm]

Table 8-104 Device Programming – Velocity Mode (Set)

8.8.2 Stop Velocity



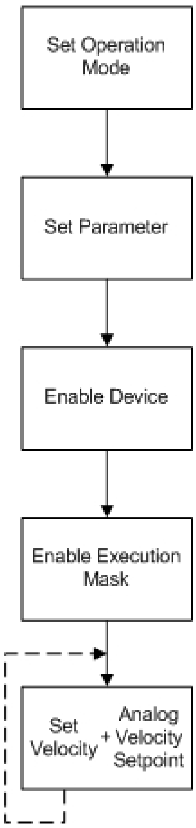
Object Name	Object	User Value [Default Value]
Velocity Mode Setting Value or Controlword (Quick stop)	0x206B-00 0x6040-00	0x00000000 0x000B

Table 8-105 Device Programming – Velocity Mode (Stop)

8.8.3 Set Velocity with analog Setpoint

For details ➔FwSpec, chapter “Velocity Mode”.

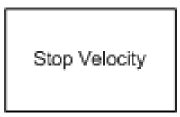
Note
For details on bits to be set for the “Controlword function” (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0xFE (Velocity Mode)
Max. Profile Velocity Max. Acceleration Configuration of Analog Input x Analog Velocity Setpoint Scaling Analog Velocity Setpoint Offset Analog Velocity Setpoint Notation Index	0x607F-00 0x60C5-00 0x207B-0x 0x2302-01 0x2302-02 0x2302-03	Motor-specific User-specific [4294967295] 0x01 User-specific [0] User-specific [0] User-specific [0]
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Analog Input Functionalities Execution Mask	0x207D-00	0x01
Velocity Mode Setting Value Analog Position Setpoint	0x206B-00 0x2303-04	Velocity for movement [rpm] (scaling depending on 0x608B) Calculated value

Table 8-106 Device Programming – Velocity Mode (Set, analog)

8.8.4 Stop Velocity from analog Setpoint



Object Name	Object	User Value [Default Value]
Controlword (Quick stop)	0x6040-00	0x000B

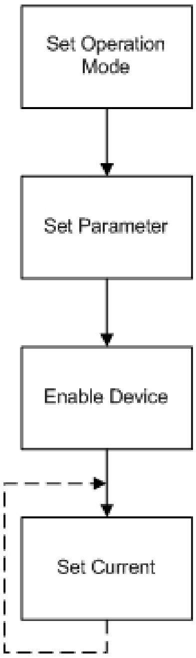
Table 8-107 Device Programming – Velocity Mode (Stop, analog)

8.9 Current Mode

8.9.1 Set Current

This command applies a certain current at the motor winding.

Note
For details on bits to be set for the “Controlword function” (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0xFD (Current Mode)
Continuous Current Limit Max. Speed in Current Mode Thermal Time Constant Winding	0x6410-01 0x6410-04 0x6410-05	Motor-specific (➔catalog for motor data)
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Current Mode Setting Value	0x2030-00	User-specific current [mA]

Table 8-108 Device Programming – Current Mode (Set)

8.9.2 Stop Motion

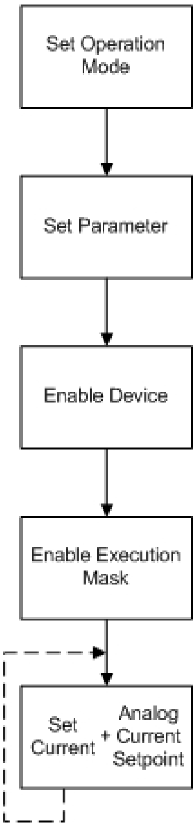
Object Name	Object	User Value [Default Value]
Current Mode Setting Value or Controlword (Quick stop)	0x2030-00 0x6040-00	0x0000 0x0002

Table 8-109 Device Programming – Current Mode (Stop)

8.9.3 Set Current with analog Setpoint

For details ➔FwSpec, chapter “Current Mode”.

Note
For details on bits to be set for the “Controlword function” (0x6040-00) ➔separate document «EPOS2 Firmware Specification», chapter 8.2.85 Controlword.



Object Name	Object	User Value [Default Value]
Modes of Operation	0x6060-00	0xFD (Current Mode)
Continuous Current Limit Max. Speed in Current Mode Thermal Time Constant Winding Configuration of Analog Input x Analog Current Setpoint Scaling Analog Current Setpoint Offset Analog Current Setpoint Notation Index	0x6410-01 0x6410-04 0x6410-05 0x207B-0x 0x2301-01 0x2301-02 0x2301-03	Motor-specific for all parameters (➔catalog for motor data) 0x00 User-specific [0] User-specific [0] User-specific [0]
Controlword (Shutdown) Controlword (Switch on & Enable)	0x6040-00 0x6040-00	0x0006 0x000F
Analog Input Functionalities Execution Mask	0x207D-00	0x00
Current Mode Setting Value Analog Current Setpoint	0x2030-00 0x2301-04	Demanded motor current [mA] Calculated value

Table 8-110 Device Programming – Current Mode (Set, analog)

8.9.4 Stop Motion from analog Setpoint

Object Name	Object	User Value [Default Value]
Controlword (Quick stop)	0x6040-00	0x0002

Table 8-111 Device Programming – Current Mode (Stop, analog)

8.10 State Machine

8.10.1 Clear Fault

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

Clear Fault

Object Name	Object	User Value [Default Value]
Controlword (Fault Reset)	0x6040-00	0x0080

Table 8-112 Device Programming – State Machine (Clear Fault)

8.10.2 Send NMT Service

NMT Service

Object Name	Object	User Value [Default Value]
Node ID (Unique Node ID or 0 for all nodes)		
Command specifier:	0x01 0x02 0x80 0x81 0x82	Start Remote Node Stop Remote Node Enter Pre-Operational Reset Node Reset Communication

Table 8-113 Device Programming – State Machine (Send NMT Service)

8.11 Motion Info

8.11.1 Get Movement State

Object Name	Object	User Value [Default Value]
Read Statusword	0x6041-00	Bit 10 tells states that target is reached. For details ➔FwSpec.

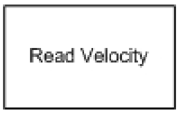
Table 8-114 Device Programming – Motion Info (Get Movement State)

8.11.2 Read Position

Object Name	Object	User Value [Default Value]
Read Position	0x6064-00	Position [qc]

Table 8-115 Device Programming – Motion Info (Read Position)

8.11.3 Read Velocity



Object Name	Object	User Value [Default Value]
Read Velocity	0x2028-00	Velocity [rpm]

Table 8-116 Device Programming – Motion Info (Read Velocity)

8.11.4 Read Current

Object Name	Object	User Value [Default Value]
Read Current	0x6078-00	Current [mA]

Table 8-117 Device Programming – Motion Info (Read Current)

8.12 Utilities

8.12.1 Store all Parameters

Saves all parameters.

Object Name	Object	User Value [Default Value]
Save All Parameters	0x10101-01	0x65766173 “save”

Table 8-118 Device Programming – Utilities (Store all Parameters)

8.12.2 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 8-119 Device Programming – Utilities (Restore all default Parameters)

8.12.3 Restore default PDO COB-ID

Sets all COB-IDs of PDOs to default (Node ID based) value.

Object Name	Object	User Value [Default Value]
Restore Default COB-IDs	0x1011-05	0x64616F6C “load”

Table 8-120 Device Programming – Utilities (Restore default PDO COB-ID)

••page intentionally left blank••

9 Controller Architecture

9.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

In addition to the standard EPOS2 PID position control, also feedforward compensation is available. The feedforward compensation provides faster setpoint following in applications with higher load inertia and accelerations and/or in applications with considerable speed-dependent load (as with friction-afflicted drives). With some EPOS2 Positioning Controllers, dual loop regulation is available.

9.1.1 Objective

The present Application Note explains the EPOS2 controller architecture. Furthermore explained will be mapping of internal controller parameters to controller parameters in SI units, and vice versa.

In addition to PID position regulation, the functionalities of built-in acceleration and velocity feedforward are described. Their advantages, compared to simple PID control are shown using two “in practice examples”.

Contents

9.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2121h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	
EPOS2 50/5	347717	2110h or higher	
EPOS2 Module 36/2	360665	2110h or higher	
EPOS2 24/5	367676	2110h or higher	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	

Table 9-121 Controller Architecture – covered Hardware and required Documents

9.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 9-122 Controller Architecture – recommended Tools

9.2 Overview

The EPOS2 controller architecture contains three built-in control loops.

- Current regulation is used in all modes.
- Position and velocity controllers are only used in position-based, respectively velocity-based modes.
- Current control loop receives as input the position, respectively velocity controller’s output.

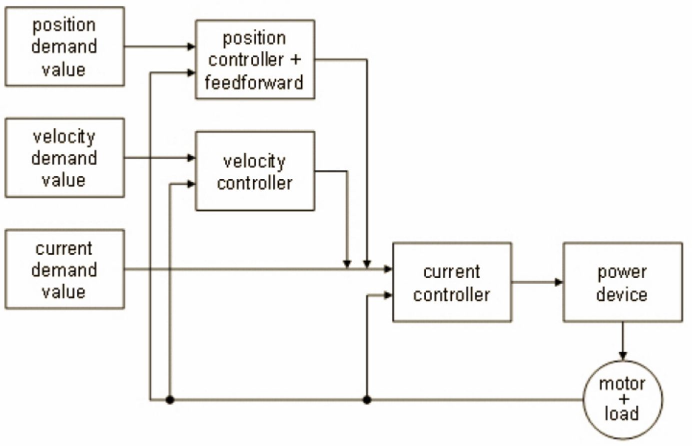


Figure 9-73 Controller Architecture

9.3 Regulation Methods

9.3.1 Current Regulation

During a movement within a drive system, forces and/or torques must be controlled. Therefore, as a principal regulation structure, EPOS2 offers current-based control.

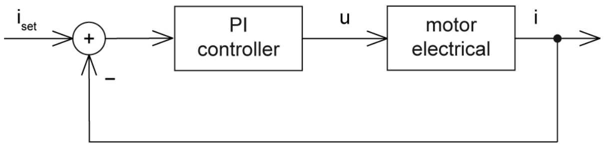


Figure 9-74 Controller Architecture – Current Regulator

Constants

Sampling period: T_s = 100 µs

Object Dictionary Entries

Symbol	Name	Index	Subindex
K_{P_EPOS2}	Current Regulator P-Gain	0x60F6	0x01
K_{I_EPOS2}	Current Regulator I-Gain	0x60F6	0x02

Table 9-123 Current Regulation – Object Dictionary

Conversion of PI Controller Parameters (EPOS2 to SI Units)

$$K_{P...SI} = \frac{1\Omega}{2^8} \cdot K_{P...EPOS2} = 3.91 m\Omega \cdot K_{P...EPOS2}$$

$$K_{I...SI} = \frac{1\Omega}{2^8 T_s} \cdot K_{I...EPOS2} = 39.1\frac{\Omega}{s} \cdot K_{I...EPOS2}$$

Current controller parameters in SI units can be used in analytical calculations, respectively numerical simulations via transfer function:

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

9.3.2 Velocity Regulation (with Feedforward)

Based on the subordinated current control, EPOS2 also offers velocity regulation.

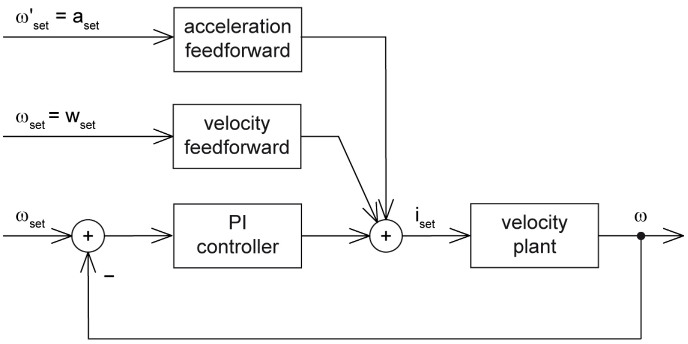


Figure 9-75 Controller Architecture – Velocity Regulator

Constants

Sampling period: T_s = 1 ms

Object Dictionary Entries

Symbol	Name	Index	Subindex
K_{P_EPOS2}	Speed Regulator P-Gain	0x60F9	0x01
K_{I_EPOS2}	Speed Regulator I-Gain	0x60F9	0x02
K_{ω_EPOS2}	Velocity Feedforward Factor in Speed Regulator	0x60F9	0x04
K_{α_EPOS2}	Acceleration Feedforward Factor in Speed Regulator	0x60F9	0x05

Table 9-124 Velocity Regulation – Object Dictionary

Conversion of PI Controller Parameters (EPOS2 to SI Units)

$$K_{P...SI} = 20\frac{\mu A}{(rad)/s} \cdot K_{P...EPOS2}$$

$$K_{I...SI} = 5\frac{(mA)/s}{(rad)/s} \cdot K_{I...EPOS2}$$

Velocity controller parameters in SI units can be used in analytical calculations, respectively numerical simulations via transfer function:

$$C_{velocity}(s) = K_{P...SI} + \frac{K_{I...SI}}{s}$$

Conversion of Feedforward Parameters (EPOS2 to SI Units)

Velocity feedforward: $K_{\omega...SI} = 1\frac{\mu A}{(rad)/s} \cdot K_{\omega...EPOS2}$

Acceleration feedforward: $K_{\alpha...SI} = 1\frac{\mu A}{(rad)/s^2} \cdot K_{\alpha...EPOS2}$

9.3.3 Position Regulation (with Feedforward)

Based on the subordinated current control, EPOS2 is able to close a positioning control loop.

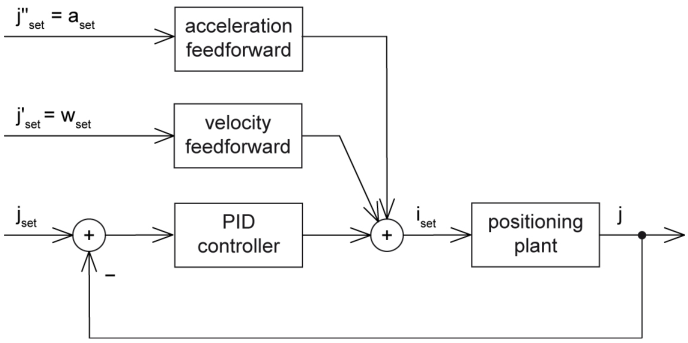


Figure 9-76 Controller Architecture – Position Regulator with Feedforward

Constants

Sampling period: T_s = 1 ms

Object Dictionary Entries

Symbol	Name	Index	Subindex
K_{P_EPOS2}	Position Regulator P-Gain	0x60FB	0x01
K_{I_EPOS2}	Position Regulator I-Gain	0x60FB	0x02
K_{D_EPOS2}	Position Regulator D-Gain	0x60FB	0x03
K_{ω_EPOS2}	Velocity Feedforward Factor in Position Regulator	0x60FB	0x04
K_{α_EPOS2}	Acceleration Feedforward Factor in Position Regulator	0x60FB	0x05

Table 9-125 Position Regulation with Feedforward – Object Dictionary

The position controller is implemented as PID controller. To improve the motion system's setpoint following, positioning regulation is supplemented by feedforward control. Thereby, velocity feedforward serves for compensation of speed-proportional friction, whereas acceleration feedforward considers known inertia.

Conversion of PI Controller Parameters (EPOS2 to SI Units)

$$K_{P...SI} = 10\frac{mA}{rad} \cdot K_{P...EPOS2}$$

$$K_{I...SI} = 78\frac{(mA)/s}{rad} \cdot K_{I...EPOS2}$$

$$K_{D...SI} = 80\frac{\mu As}{rad} \cdot K_{D...EPOS2}$$

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

$$C_{position}(s) = K_{P...SI} + \frac{K_{I...SI}}{s} + \frac{K_{D...SI}S}{1 + \frac{K_{D...SI}}{16K_{P...SI}}s}$$

Document ID: rel7167
Edition: November 2017

Conversion of Feedforward Parameters (EPOS2 to SI Units)

Velocity feedforward: $$K_{\omega \ldots SI} = 1\frac{\mu A}{(rad)/s} \cdot K_{\omega \ldots EPOS2}$$

Acceleration feedforward: $$K_{\alpha \ldots SI} = 1\frac{\mu A}{(rad)/s^2} \cdot K_{\alpha \ldots EPOS2}$$

9.3.4 Operation Modes with Feedforward

Acceleration and velocity feedforward have an effect in «Profile Position Mode», «Profile Velocity Mode» and «Homing Mode». All other operating modes are not influenced.

9.3.4.1 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 increase proportional with speed. The optimal velocity feedforward parameter in SI units is…

$$K_{\omega \ldots SI} = \frac{r}{k_M}$$

Meaning: With given total friction proportional factor “r” relative to the motor shaft, and the motor’s torque constant “k_M”, you ought to adjust the velocity feedforward parameter to…

$$K_{\omega \ldots EPOS2} = \frac{r}{k_M} \cdot \frac{(rad)/s}{1\mu A} = \frac{r}{k_M} \cdot \frac{10^6(rad)/s}{A}$$

9.3.4.2 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…

$$K_{\alpha \ldots SI} = \frac{J}{k_M}$$

Meaning: With given total inertia “J” relative to the motor shaft, and the motor’s torque constant “k_M”, you ought to adjust the acceleration feedforward parameter to…

$$K_{\alpha \ldots EPOS2} = \frac{J}{k_M} \cdot \frac{(rad)/s^2}{1\mu A} = \frac{J}{k_M} \cdot \frac{10^6(rad)/s^2}{A}$$

9.4 Regulation Tuning

maxon motor’s «EPOS Studio» features «Regulation Tuning» as powerful wizard allowing to automatically tune all controller and feedforward parameters described above for most drive systems within a few minutes. For details ➔chapter “7 Regulation Tuning” on page 7-95.

9.5 Dual Loop Regulation

Available with EPOS2 70/10, EPOS2 50/5 and EPOS2 Module 36/2 only!

In many applications it is common to use gears to increase motor torque, or screw spindles to transform motor rotation into linear movement. The gear itself is made of a lot of different parts, such as, belts, pinions, pulleys, spindles, etc.

The associated elasticity and backlash of these parts create an effect of compliance and as well as a delay in the drive chain. Often, the mechanical transmission between motor and load has some backlash, too, resulting in a certain “delay” being introduced to the plant. This delay influences the regulation stability and may have such big impact that one may be forced to reduce the dynamic behavior or the precision of the drive.

To overcome these limitations and to combine a motor/gear system with a precise and high dynamic regulation, it will be necessary to control the motor movement as well as the load movement. This results in a new control structure called “dual loop”, featuring two individual encoders – one directly mounted to the motor, the another mounted at the gear or linear slide or directly on/near to the load.

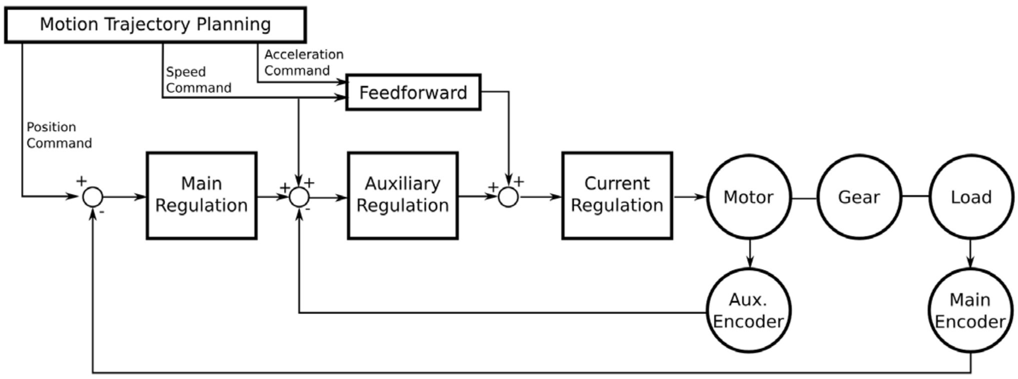


Figure 9-77 Dual Loop Architecture

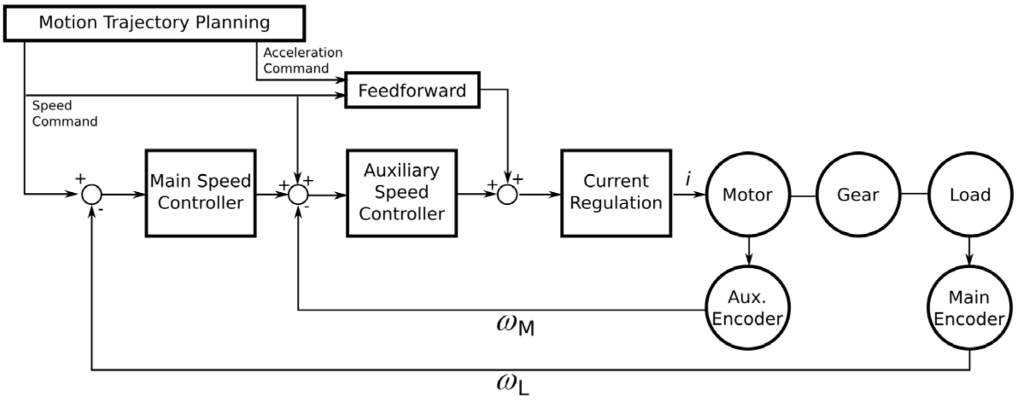
The auxiliary regulation is designed to provide damping and dynamic system behavior while the main regulation generates the desired position precision.

9.5.1 Current Regulation

The dual loop current controller is implemented similar to the current controller in a single loop system. For details ➔chapter “9.3.1 Current Regulation” on page 9-124.

9.5.2 Velocity Regulation (with Feedforward)

The design is based on current regulation.



ω_M motor speed
ω_L load speed

Figure 9-78 Dual Loop Velocity Regulation

In velocity mode, the auxiliary controller appropriately stabilizes the loop; however, the main controller provides the correct speed feedback.

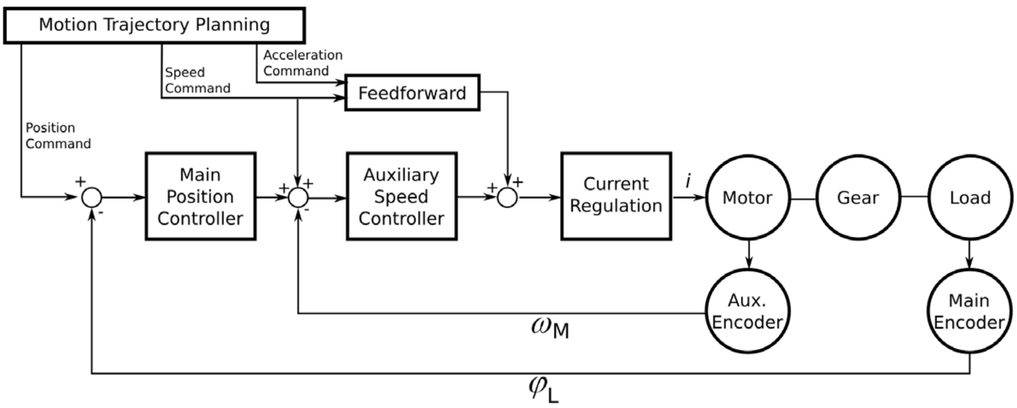
The dual loop velocity controller (that is main controller and auxiliary controller together) is implemented as PI controller.

Conversion parameters

Conversion of PI controller and feedforward parameters in dual loop (EPOS2 to SI units) are identical to that in single loop (➔chapter “9.3.2 Velocity Regulation (with Feedforward)” on page 9-125).

9.5.3 Position Regulation (with Feedforward)

The design is based on current regulation.



ω_M motor speed
φ_L load position

Figure 9-79 Dual Loop Position Regulation

In position mode, the auxiliary controller is designed to stabilize the loop, whereas the main controller provides the correct position feedback.

The dual loop position controller (that is main controller and auxiliary controller together) is realized as PID controller and features the same sampling period as the dual loop velocity controller.

Conversion parameters

Conversion of PI controller and feedforward parameters in dual loop (EPOS2 to SI units) are identical to that in single loop (➔chapter “9.3.3 Position Regulation (with Feedforward)” on page 9-126).

9.5.4 Conclusion

The dual loop topology is adequate if the ratio of motor inertia and load inertia is not too large. The drive elements (motor, gear, encoders, load) must be dimensioned correctly.

General Selection Practice

To achieve reliability of the system, follow the scheme below to determine the individual components:

- **Motor**
  Chose a motor capable to fulfill the load’s requirements for maximum torque, continuous torque, and speed. For detailed information ➔chapter “1.6 Sources for additional Information” on page 1-12, item [7].
- **Gear**
  Chose a gear capable to fulfill the load’s torque and speed range. Boundary conditions are maximum motor load, maximum gear load, and the associated speed limits.
  Another influence that might need consideration is the minimum motor heat dissipation capability. Use the following formula to determine the optimum gear ratio:

  $$I = \sqrt{\frac{Jl}{Jm}}$$

  Jl load inertia
  Jm motor inertia
- **Motor Encoder**
  Chose a motor encoder capable to provide sufficient stiffness in the inner loop. A few hundred increments per revolution as the motor encoder’s minimum resolution are recommended.
- **Load Encoder**
  Chose a load encoder capable to at least deliver the required resolution and accuracy on the load side.

General Rule
With Dual Loop Regulation, the following general restriction applies:

$$AuxEncoderResolution \cdot GearRatio \leq MainEncoderResolution$$

9.5.5 Auto Tuning

The dual loop start up is similar to the start up of the single loop regulation and can be described with the following major steps:

1) Identification and modeling of the plant.
2) Calculation of all controller parameters (current, auxiliary, main, feedforward).
3) Mapping; the calculated controller parameters (main, auxiliary) are mathematically transformed to PI controller parameters (for velocity regulation) or to PID controller parameters (for position regulation).
4) Verification; the system’s dynamic response is measured and displayed using the scope function in «EPOS2 Studio». This allows verification, whether the system behavior is as expected.

9.6 Application Examples

Please find below two “in practice examples” suitable for daily use.

For comparability and validity reasons, the measured simulation results are converted to the units “mA”, “rpm” and “qc”!

9.6.1 Example 1: System with high Inertia and low Friction

System Components

Item	Description	Setting
Controller EPOS2 50/5 (347717)		
Motor maxon EC 40 (118896)	No load speed (line 2)	n_0 = 10’400 rpm
	No load current (line 3)	I_0 = 258 mA
	Nominal current (line 6)	I_n = 3.4 A
	Resistance phase to phase (line 10)	R = 1.25 Ω
	Inductance phase to phase (line 11)	L = 0.319 mH
	Torque constant (line 12)	k_M = 38.2 mNm/A
	Rotor inertia (line 16)	J_{motor} = 85 gcm²
Encoder HEDL 5540 (110516)	Encoder pulse number	500
Mechanical load Fly wheel	Inertia	J_{load} = 5000 gcm²

Table 9-126 Controller Architecture – Example 1: Components

Model of the Plant

The following parameters can be deduced:

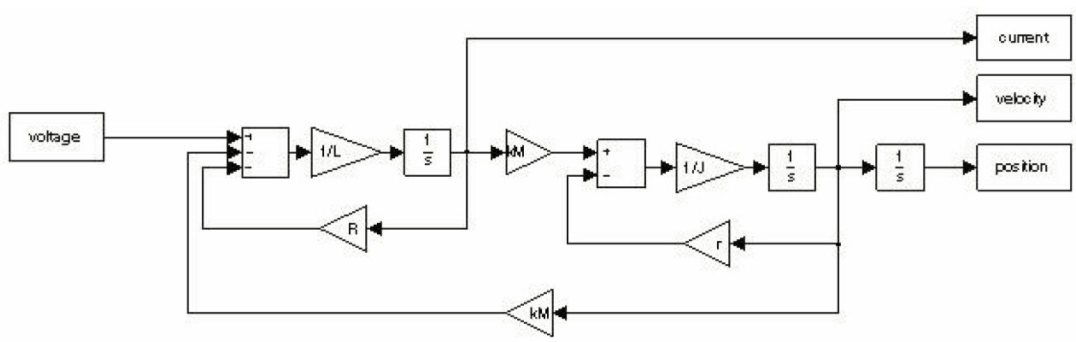


Figure 9-80 Example1 – Block Diagram

Electrical Part

R = 1.25 Ω

L = 0.319 mH

Interface between electrical and mechanical Parts

$$k_M = 38.2\frac{mNm}{A}$$

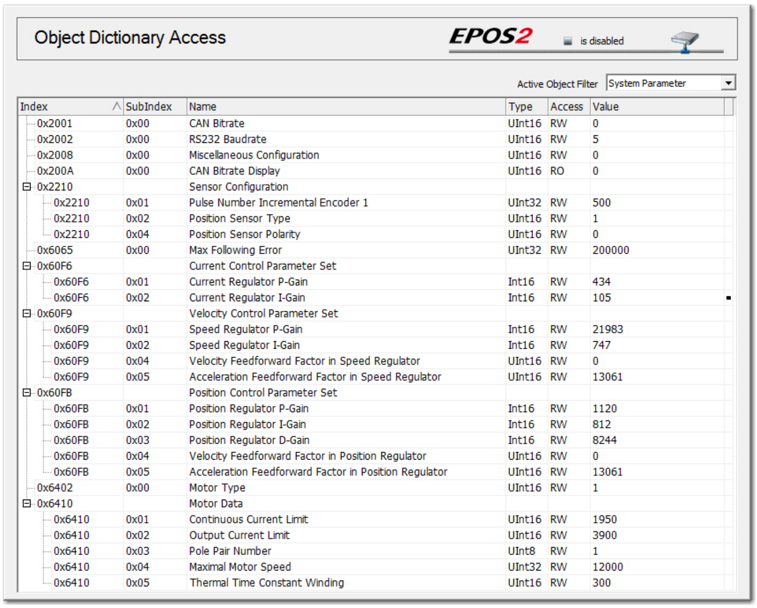
Mechanical Part

$$J = J_{motor} + J_{load} = 5085gcm^2$$

$$r = \frac{k_M I_0}{n_0 \frac{2\pi rad}{1} \cdot \frac{1min}{60s}} = \frac{9.86mNm}{1089rad^2} = 9.05\frac{\mu Nm}{(rad)/s}$$

- Input is the voltage at the motor winding.
- Outputs are current, velocity or position.

Regulation Tuning as to the described conditions results in the following controller and feedforward parameters:


Object Dictionary Access — EPOS2 — is disabled

Active Object Filter: System Parameter

Index	SubIndex	Name	Type	Access	Value
0x2001	0x00	CAN Bitrate	UInt16	RW	0
0x2002	0x00	RS232 Baudrate	UInt16	RW	5
0x2008	0x00	Miscellaneous Configuration	UInt16	RW	0
0x200A	0x00	CAN Bitrate Display	UInt16	RO	0
0x2210		Sensor Configuration			
0x2210	0x01	Pulse Number Incremental Encoder 1	UInt32	RW	500
0x2210	0x02	Position Sensor Type	UInt16	RW	1
0x2210	0x04	Position Sensor Polarity	UInt16	RW	0
0x6065	0x00	Max Following Error	UInt32	RW	200000
0x60F6		Current Control Parameter Set			
0x60F6	0x01	Current Regulator P-Gain	Int16	RW	434
0x60F6	0x02	Current Regulator I-Gain	Int16	RW	105
0x60F9		Velocity Control Parameter Set			
0x60F9	0x01	Speed Regulator P-Gain	Int16	RW	21983
0x60F9	0x02	Speed Regulator I-Gain	Int16	RW	747
0x60F9	0x04	Velocity Feedforward Factor in Speed Regulator	UInt16	RW	0
0x60F9	0x05	Acceleration Feedforward Factor in Speed Regulator	UInt16	RW	13061
0x60FB		Position Control Parameter Set			
0x60FB	0x01	Position Regulator P-Gain	Int16	RW	1120
0x60FB	0x02	Position Regulator I-Gain	Int16	RW	812
0x60FB	0x03	Position Regulator D-Gain	Int16	RW	8244
0x60FB	0x04	Velocity Feedforward Factor in Position Regulator	UInt16	RW	0
0x60FB	0x05	Acceleration Feedforward Factor in Position Regulator	UInt16	RW	13061
0x6402	0x00	Motor Type	UInt16	RW	1
0x6410		Motor Data			
0x6410	0x01	Continuous Current Limit	UInt16	RW	1950
0x6410	0x02	Output Current Limit	UInt16	RW	3900
0x6410	0x03	Pole Pair Number	UInt8	RW	1
0x6410	0x04	Maximal Motor Speed	UInt32	RW	12000
0x6410	0x05	Thermal Time Constant Winding	UInt16	RW	300

Figure 9-81 Example1 – System Parameters, real

For numerical simulation, the conversion results from EPOS2 to SI units are as follows:

Current Controller

$$K_{P...EPOS2} = 434 \quad \rightrightarrows \quad K_{P...SI} = 1.70\Omega$$

$$K_{I...EPOS2} = 105 \quad \rightrightarrows \quad K_{I...SI} = 4.11\frac{k\Omega}{s}$$

Velocity Controller

$$K_{P...EPOS2} = 21983 \quad \rightrightarrows \quad K_{P...SI} = 0.440\frac{A}{(rad)/s}$$

$$K_{I...EPOS2} = 747 \quad \rightrightarrows \quad K_{I...SI} = 3.74\frac{A/s}{(rad)/s}$$

Position Controller

$$K_{P...EPOS2} = 1120 \quad \rightrightarrows \quad K_{P...SI} = 11.2\frac{A}{rad}$$

$$K_{I...EPOS2} = 812 \quad \rightrightarrows \quad K_{I...SI} = 63.2\frac{A/s}{rad}$$

$$K_{D...EPOS2} = 8244 \quad \rightrightarrows \quad K_{D...SI} = 0.660\frac{As}{rad}$$

Positioning and Velocity Feedforward

$$K_{\omega...EPOS2} = 0 \quad \rightrightarrows \quad K_{\varpi...SI} = 0\frac{A}{(rad)/s}$$

$$K_{\alpha...EPOS2} = 13061 \quad \rightrightarrows \quad K_{\alpha...SI} = 13.06\frac{mA}{(rad)/s^2}$$

Plausibility Check

$$K_{\omega...SI} = \frac{r}{k_M} = 237\frac{\mu A}{(rad)/s} \quad (\rightrightarrows) \quad K_{\varpi...SI} = 237\frac{\mu A}{(rad)/s} \sim 0\frac{A}{(rad)/s}$$

$$K_{a...SI} = \frac{J}{k_M} = \frac{5085 \cdot 10^{-7}\frac{Nm}{(rad)/s}}{38.2 \cdot 10^{-3}\frac{Nm}{A}} = 13.3\frac{mA}{(rad)/s^2}$$

Verification of Current Control

The plant is connected to the PI current controller. The controller is parameterized as described above.

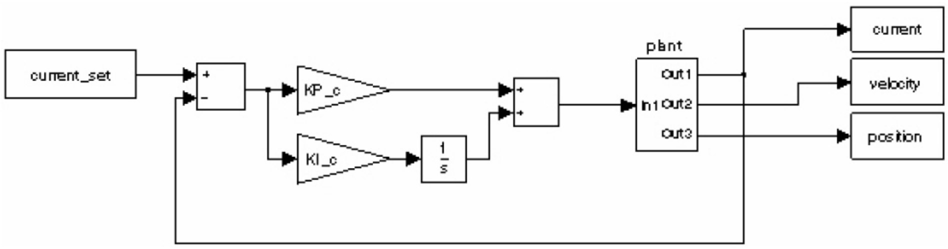


Figure 9-82 Example1 – Current Regulation, Block Model

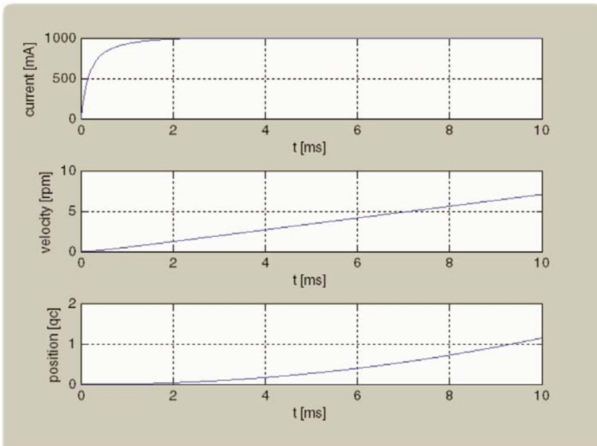


Figure 9-83 Example1 – Current Regulation, simulated

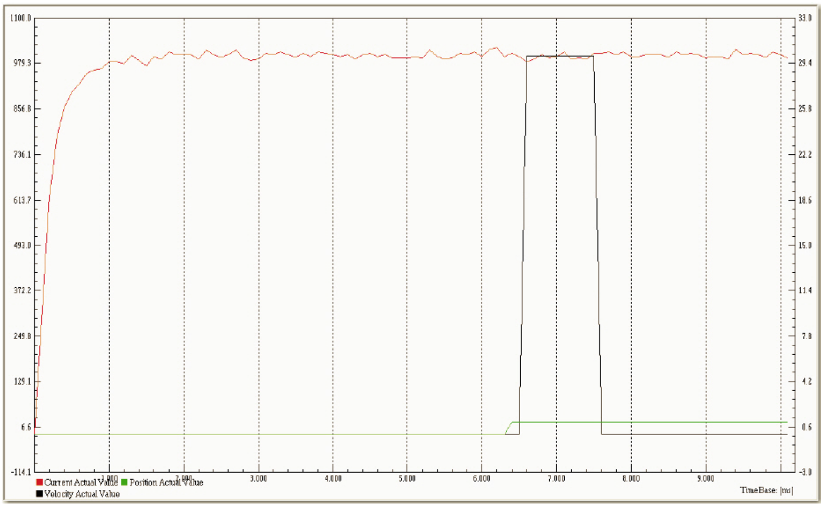


Figure 9-84 Example1 – Current Regulation, measured

Verification of Velocity Control

The PI velocity controller is connected to current regulation.

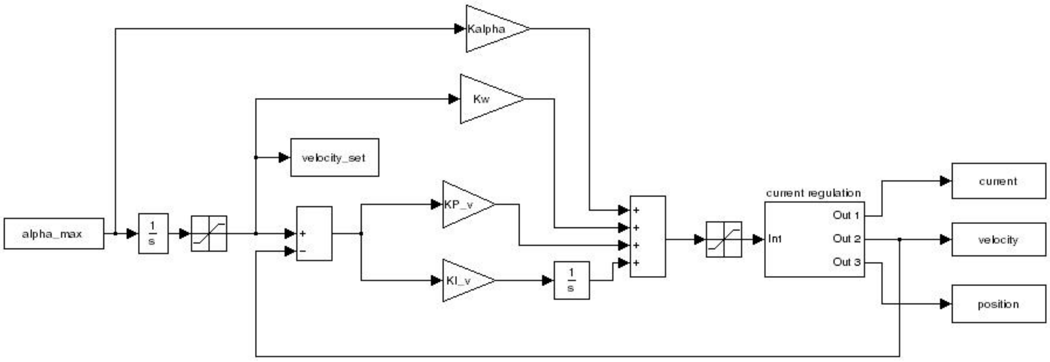


Figure 9-85 Example1 – Velocity Regulation, Block Model

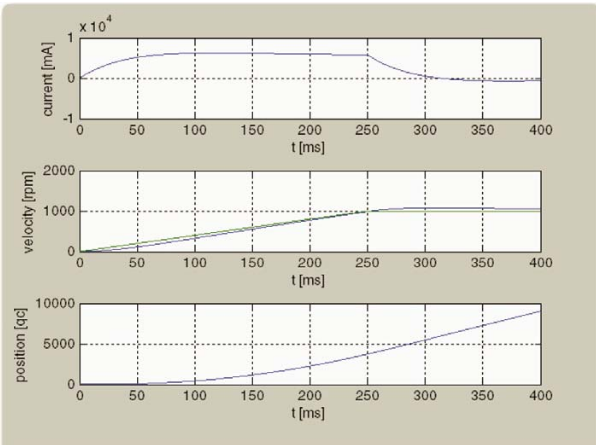


Figure 9-86 Example1 – Velocity Regulation, simulated

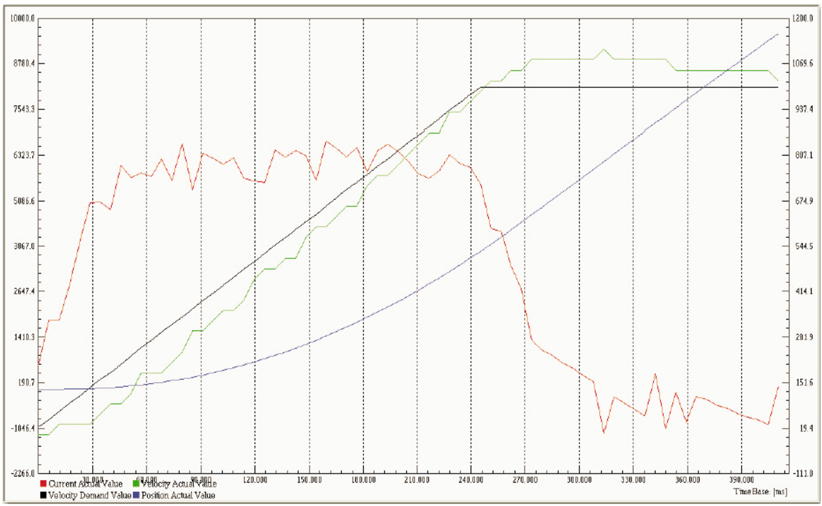


Figure 9-87 Example1 – Velocity Regulation, measured

Verification of Position Control with Feedforward

The PID position controller is connected to current regulation.

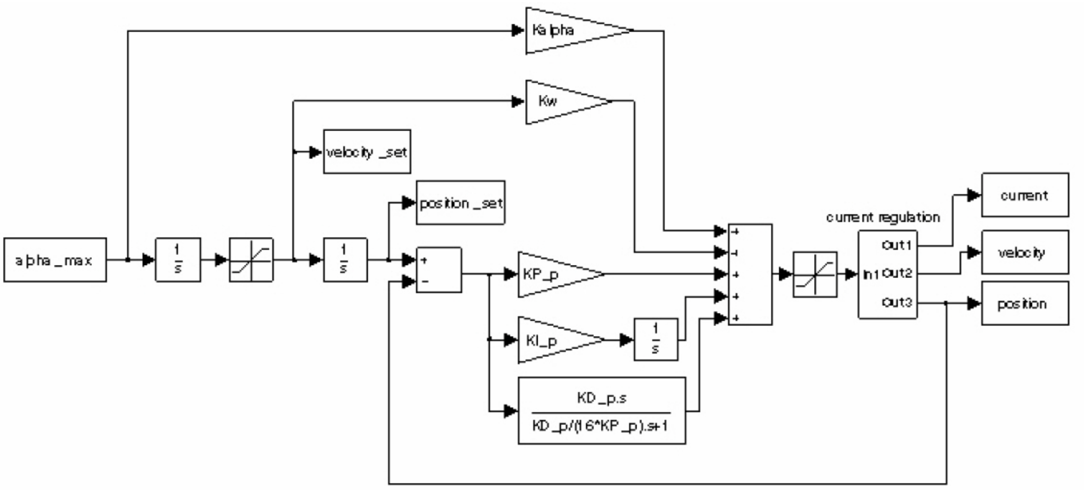


Figure 9-88 Example1 – Position Control with Feedforward, Block Model

With correct Feedforward

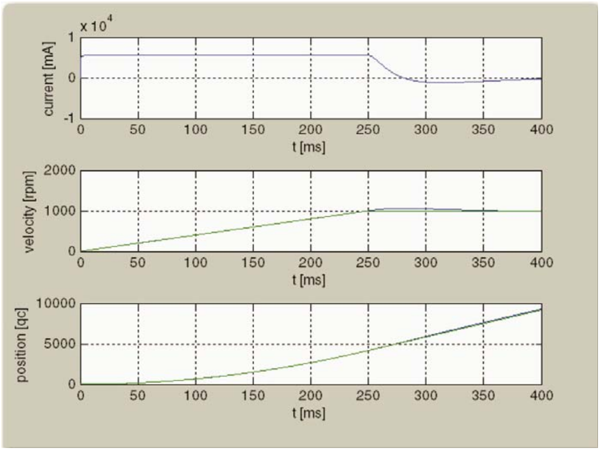


Figure 9-89 Example1 – Position Control with Feedforward, simulated

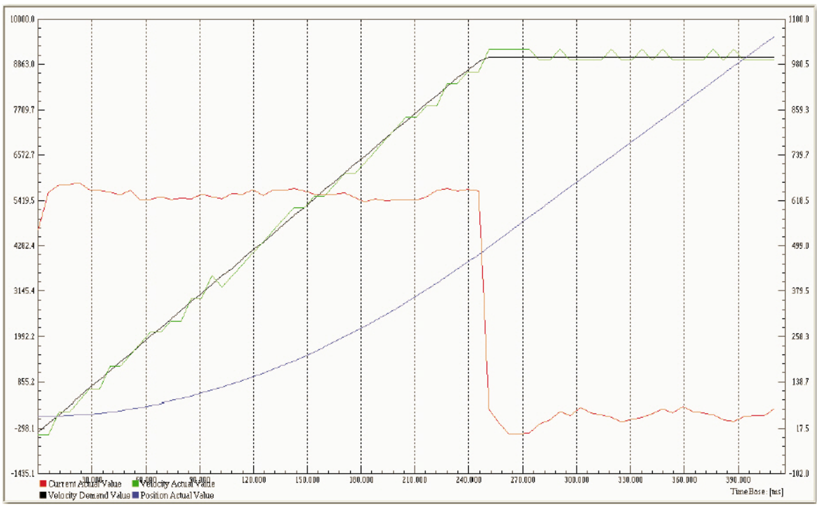


Figure 9-90 Example1 – Position Control with Feedforward, measured

Without Feedforward

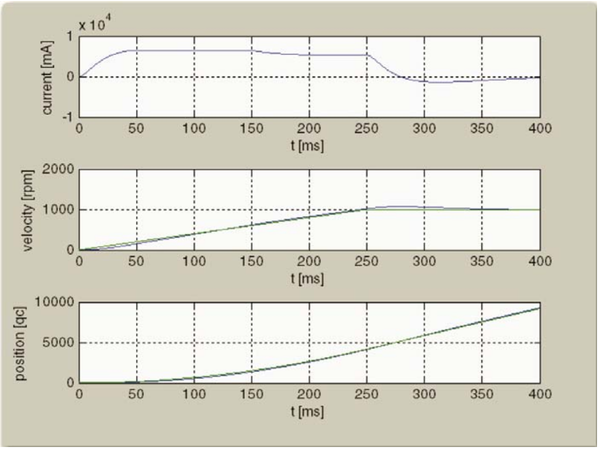


Figure 9-91 Example1 – Position Control without Feedforward, simulated

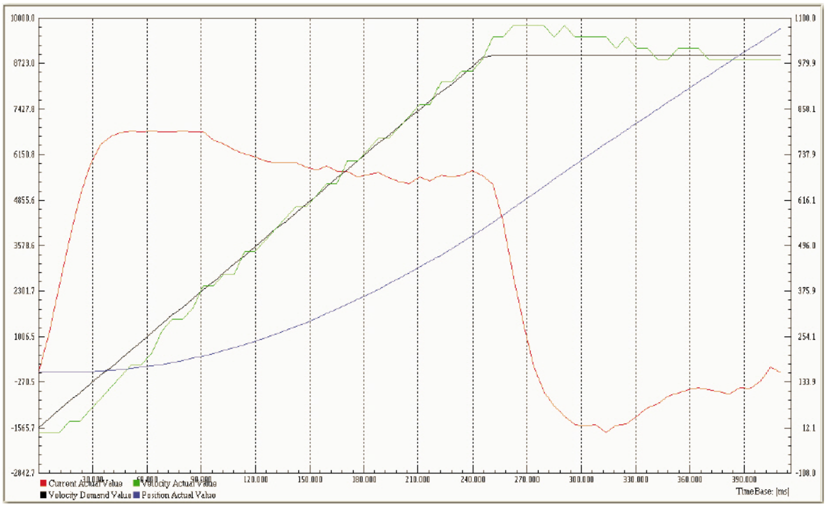


Figure 9-92 Example1 – Position Control without Feedforward, measured

With incorrect Feedforward (acceleration Feedforward parameter doubled)

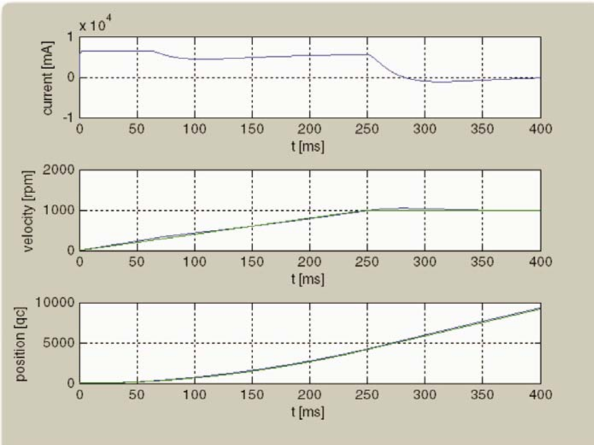


Figure 9-93 Example1 – Position Control with incorrect Feedforward, simulated

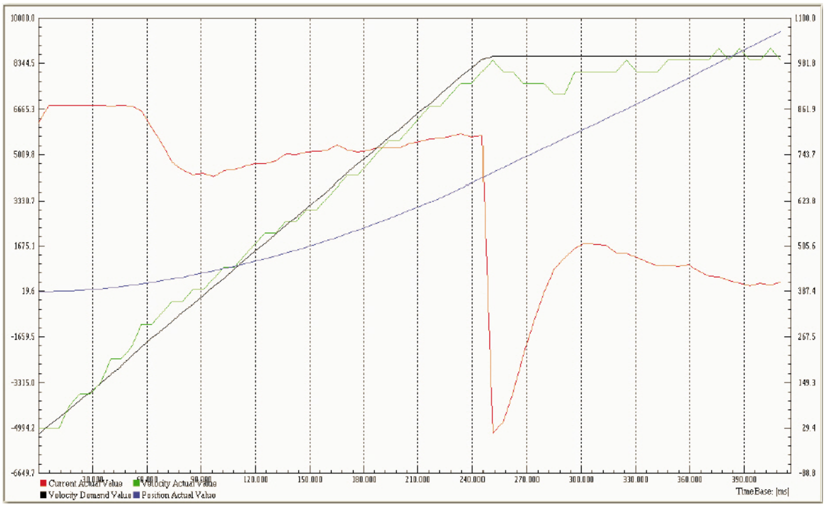


Figure 9-94 Example1 – Position Control with incorrect Feedforward, measured

9.6.2 Example 2: System with low Inertia, but high Friction

Figure 9-95 Controller Architecture – Example 2: System with low Inertia/high Friction

System Components

Item	Description	Setting
Controller EPOS2 50/5 (347717)		
Motor maxon RE 35 (273754)	No load speed (line 2)	n_0 = 7530 rpm
	No load current (line 3)	I_0 = 92.7 mA
	Nominal current (line 6)	I_n = 1.95 A
	Resistance phase to phase (line 10)	R = 2.07 Ω
	Inductance phase to phase (line 11)	L = 0.620 mH
	Torque constant (line 12)	k_M = 52.5 mNm/A
	Rotor inertia (line 16)	J_{motor} = 72 gcm^2
Encoder HEDL 5540 (110514)	Encoder pulse number	500
Mechanical load Linear Drive	Inertia	J_{load} = 100 gcm^2
	Friction, velocity-dependent $M_r = 211\frac{\mu Nm}{(rad)/s}\omega + 8.65mNm \cdot sign(\omega)$	

Table 9-127 Controller Architecture – Example 2: Components

Model of the Plant

The following parameters can be deduced:

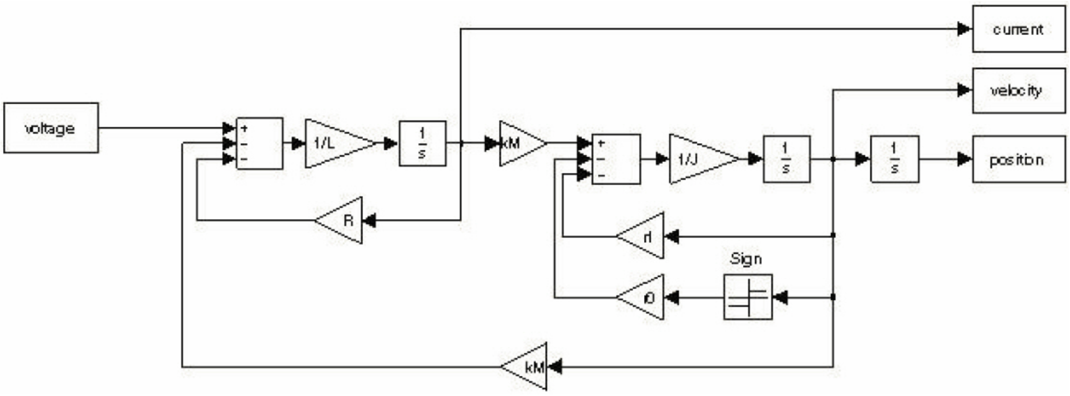


Figure 9-96 Example 2 – Block Diagram

Electrical Part

R = 2.07 Ω

L = 0.620 mH

Interface between electrical and mechanical Parts

$$k_M = 52.5\frac{mNm}{A}$$

Mechanical Part

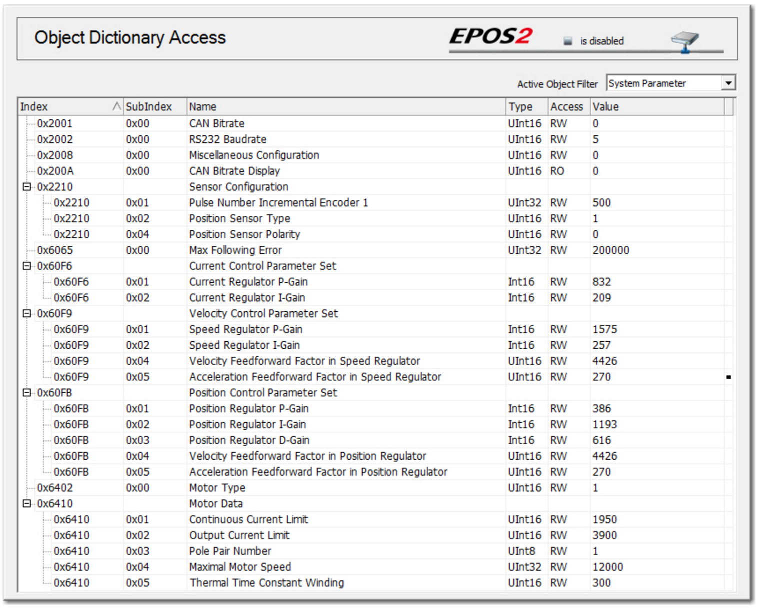
$$J = J_{motor} + J_{load} = 172gcm^2$$

$$r_0 = 8.65mNm$$

$$r_1 = \underbrace{211\frac{\mu Nm}{(rad)/s}}_{load} + \underbrace{\frac{k_M I_0}{n_0\frac{2\pi rad}{1}\cdot\frac{1min}{60s}}}_{motor} = (211+6)\frac{\pi Nm}{(rad)/s} = 217\frac{\pi Nm}{(rad)/s}$$

- Input is the voltage at the motor winding.
- Outputs are current, velocity or position.

Regulation Tuning according to the described conditions results in the following controller and feedforward parameters:



Index	SubIndex	Name	Type	Access	Value
0x2001	0x00	CAN Bitrate	UInt16	RW	0
0x2002	0x00	RS232 Baudrate	UInt16	RW	5
0x2008	0x00	Miscellaneous Configuration	UInt16	RW	0
0x200A	0x00	CAN Bitrate Display	UInt16	RO	0
0x2210		Sensor Configuration			
0x2210	0x01	Pulse Number Incremental Encoder 1	UInt32	RW	500
0x2210	0x02	Position Sensor Type	UInt16	RW	1
0x2210	0x04	Position Sensor Polarity	UInt16	RW	0
0x6065	0x00	Max Following Error	UInt32	RW	200000
0x60F6		Current Control Parameter Set			
0x60F6	0x01	Current Regulator P-Gain	Int16	RW	832
0x60F6	0x02	Current Regulator I-Gain	Int16	RW	209
0x60F9		Velocity Control Parameter Set			
0x60F9	0x01	Speed Regulator P-Gain	Int16	RW	1575
0x60F9	0x02	Speed Regulator I-Gain	Int16	RW	257
0x60F9	0x04	Velocity Feedforward Factor in Speed Regulator	UInt16	RW	4426
0x60F9	0x05	Acceleration Feedforward Factor in Speed Regulator	UInt16	RW	270
0x60FB		Position Control Parameter Set			
0x60FB	0x01	Position Regulator P-Gain	Int16	RW	386
0x60FB	0x02	Position Regulator I-Gain	Int16	RW	1193
0x60FB	0x03	Position Regulator D-Gain	Int16	RW	616
0x60FB	0x04	Velocity Feedforward Factor in Position Regulator	UInt16	RW	4426
0x60FB	0x05	Acceleration Feedforward Factor in Position Regulator	UInt16	RW	270
0x6402	0x00	Motor Type	UInt16	RW	1
0x6410		Motor Data			
0x6410	0x01	Continuous Current Limit	UInt16	RW	1950
0x6410	0x02	Output Current Limit	UInt16	RW	3900
0x6410	0x03	Pole Pair Number	UInt8	RW	1
0x6410	0x04	Maximal Motor Speed	UInt32	RW	12000
0x6410	0x05	Thermal Time Constant Winding	UInt16	RW	300

Figure 9-97 Example 2 – System Parameters, real

For numerical simulation, the conversion results from EPOS2 to SI units are as follows:

Current Controller

$$K_{P...EPOS2} = 832 \quad \rightrightarrows \quad K_{P...SI} = 3.25\Omega$$

$$K_{I...EPOS2} = 209 \quad \rightrightarrows \quad K_{I...SI} = 8.17\frac{k\Omega}{s}$$

Velocity Controller

$$K_{P...EPOS2} = 1575 \quad \rightrightarrows \quad K_{P...SI} = 31.5\frac{mA}{(rad)/s}$$

$$K_{I...EPOS2} = 257 \quad \rightrightarrows \quad K_{I...SI} = 1.29\frac{A/s}{(rad)/s}$$

Position Controller

$$K_{P...EPOS2} = 386 \quad \rightrightarrows \quad K_{P...SI} = 3.86\frac{A}{rad}$$

$$K_{I...EPOS2} = 1193 \quad \rightrightarrows \quad K_{I...SI} = 93.1\frac{A/s}{rad}$$

$$K_{D...EPOS2} = 616 \quad \rightrightarrows \quad K_{D...SI} = 49.3\frac{mAs}{rad}$$

Positioning and Velocity Feedforward

$$K_{\omega...EPOS2} = 4426 \quad \rightrightarrows \quad K_{\varpi...SI} = 4.42\frac{mA}{(rad)/s}$$

$$K_{\alpha...EPOS2} = 270 \quad \rightrightarrows \quad K_{\alpha...SI} = 270\frac{\mu A}{(rad)/s^2}$$

Plausibility Check

$$K_{\omega...SI} = \frac{r_1}{k_M} = 4.13\frac{mA}{(rad)/s}$$

$$K_{a...SI} = \frac{J}{k_M} = \frac{172 \cdot 10^{-7}\frac{Nm}{(rad)/s}}{52.5 \cdot 10^{-3}\frac{Nm}{A}} = 327\frac{\mu A}{(rad)/s^2}$$

Verification of Current Control

The plant is connected to the PI current controller. The controller is parameterized as described above.

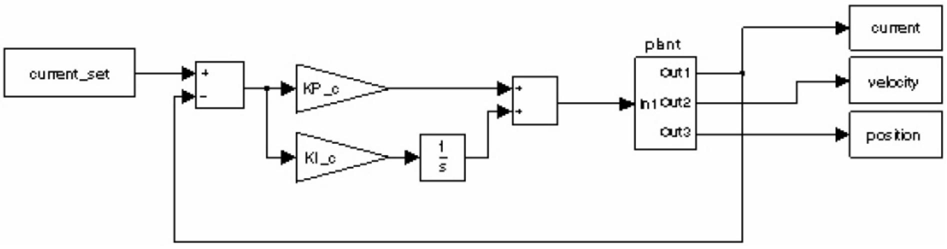


Figure 9-98 Example 2 – Current Regulation, Block Model

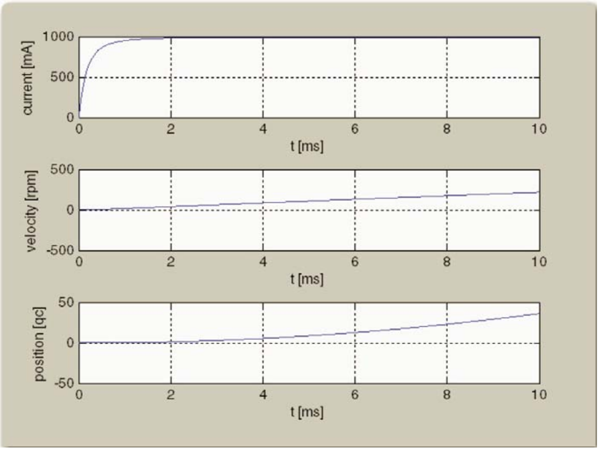


Figure 9-99 Example 2 – Current Regulation, simulated

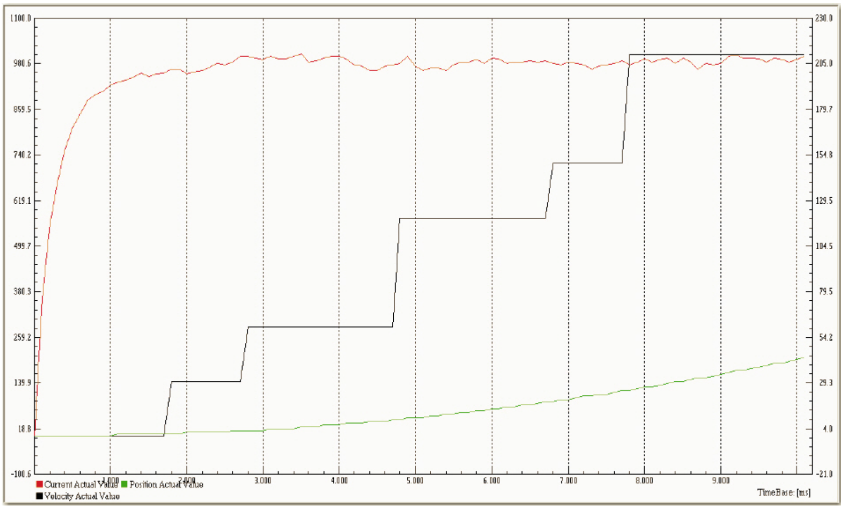


Figure 9-100 Example 2 – Current Regulation, measured

Verification of Velocity Control

The PI velocity controller is connected to current regulation.

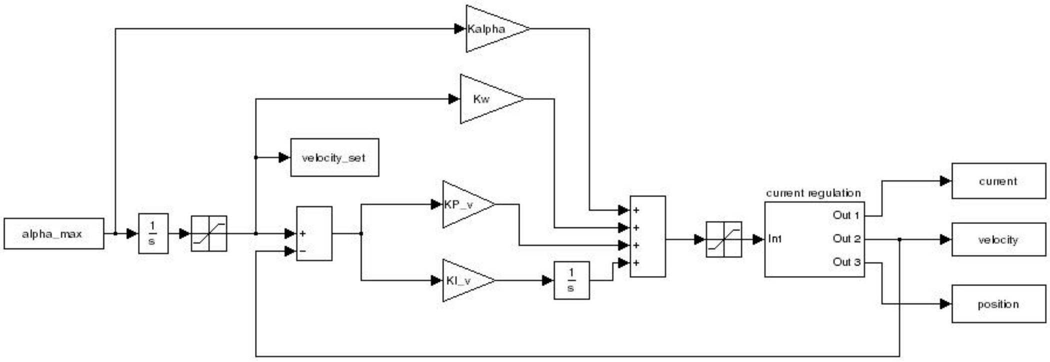


Figure 9-101 Example 2 – Velocity Regulation, Block Model

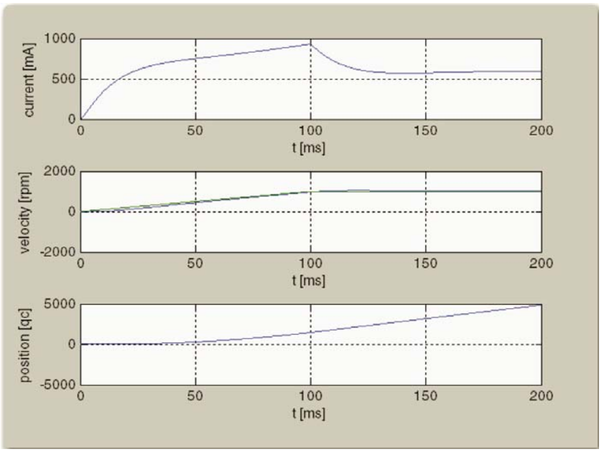


Figure 9-102 Example 2 – Velocity Regulation, simulated

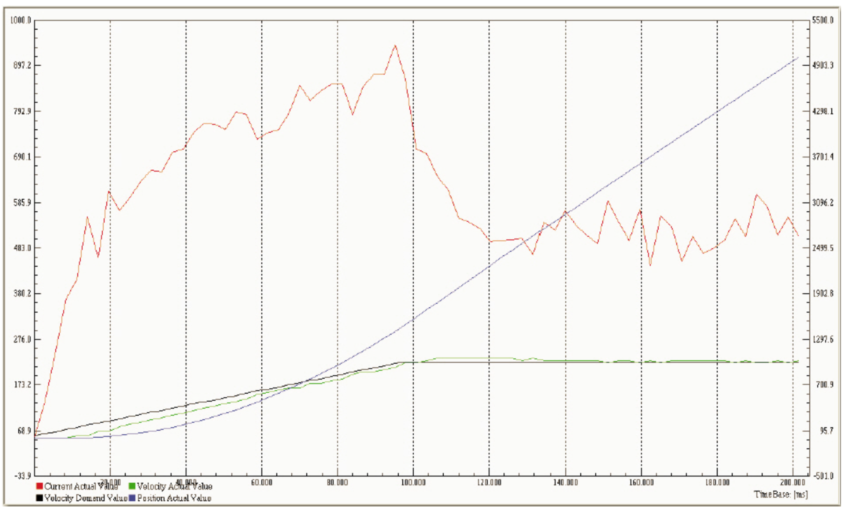


Figure 9-103 Example 2 – Velocity Regulation, measured

Verification of Position Control with Feedforward

The PID position controller is connected to current regulation.

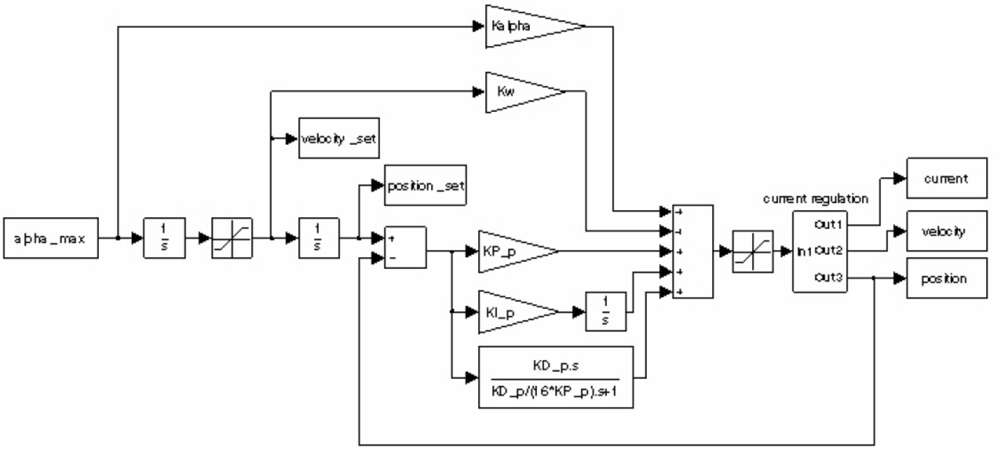


Figure 9-104 Example 2 – Position Control with Feedforward, Block Model

With correct Feedforward

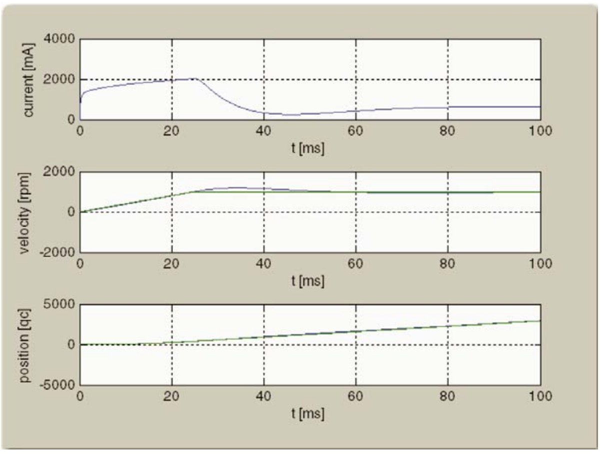


Figure 9-105 Example 2 – Position Control with Feedforward, simulated

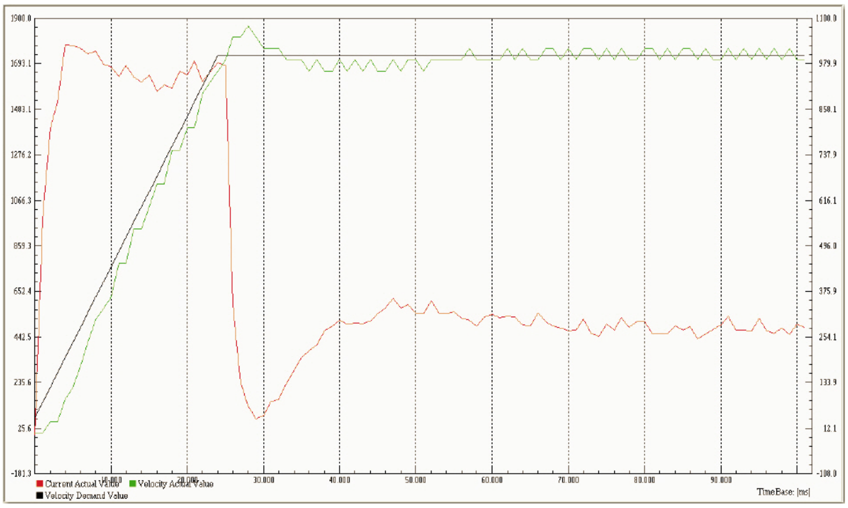


Figure 9-106 Example 2 – Position Control with Feedforward, measured

Without Feedforward

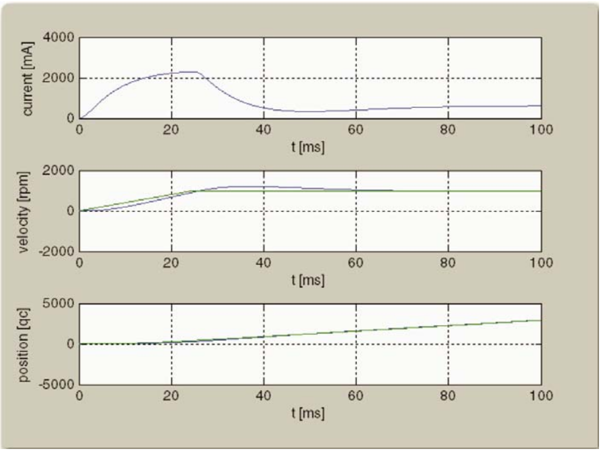


Figure 9-107 Example 2 – Position Control without Feedforward, simulated

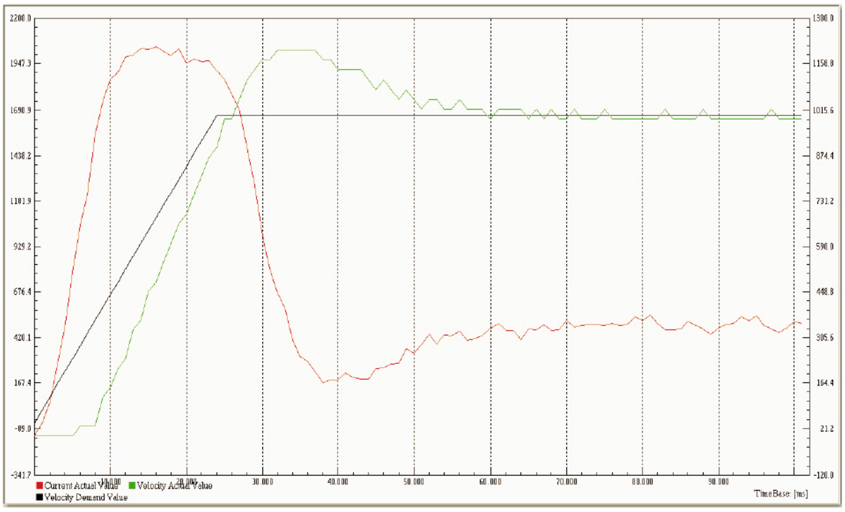


Figure 9-108 Example 2 – Position Control without Feedforward, measured

9.7 Conclusion

Scaling of the internal controller parameters is a specific EPOS2 feature. To understand these parameters and to use them in analytical calculations, respectively numerical simulations, understanding on how to map EPOS2's internal controller parameters to SI units controller parameters, and vice versa, is essential.

In practice, direct drive systems are often used because of their lower overall costs and the requirement for a backlash-free behavior. As a result, the ratio between motor inertia and load inertia often are 1:10, or higher.

Therefore, EPOS2's PID position control with feedforward compensation is of great advantage. Compared to simple PID control, the feedforward compensation provides significant faster and more accurate setpoint following.

••page intentionally left blank••

10 CANopen Basic Information

10.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

For fast communication with several EPOS2 devices, we suggest to use the CANopen protocol. The individual devices within the network are commanded by a CANopen master.

10.1.1 Objective

The present Application Note explains the functionality of the CANopen structure and protocol. It also describes the configuration process in a step-by-step procedure.

Contents

10.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification Communication Guide
EPOS2 70/10	375711	2120h or higher	
EPOS2 50/5	347717	2110h or higher	
EPOS2 Module 36/2	360665	2110h or higher	
EPOS2 24/5	367676	2110h or higher	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	
CANopen Network			CiA 301 V4.2 (➔[1]) CiA 402 V3.0 (➔[2])

Table 10-128 CANopen Basic Information – covered Hardware and required Documents

10.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 10-129 CANopen Basic Information – recommended Tools

10.2 Network Structure

maxon EPOS2 drives' CAN interface follows the CiA CANopen specification CiA 301 V4.2 "Communication Profile for Industrial Systems" and CiA 402 V3.0 "Device Profile for Drives and Motion Control".

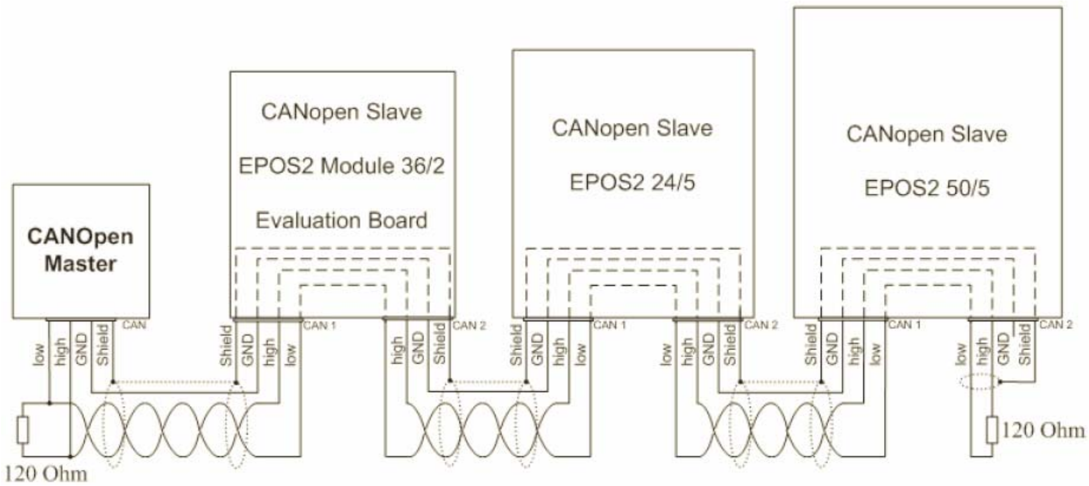


Figure 10-109 CANopen Network Structure (Example)

The CAN bus line must be terminated at both ends using a termination resistor of typically 120 Ω.

Use the internal bus termination as far as available on the EPOS2 Positioning Controller. The bus termination can be switched on by DIP switch.

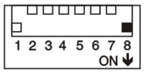
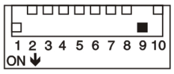
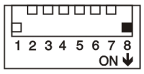
Device	Bus terminated with 120 Ω	DIP Switch Setting
EPOS2 70/10	DIP switch 8 "ON"	
EPOS2 50/5	DIP switch 9 "ON"	
EPOS2 24/5	DIP switch 8 "ON"	
EPOS2 24/2	DIP switch 6 "ON"	

Table 10-130 DIP Switch Settings for CAN Bus Termination

10.3 Configuration

Follow below step-by-step instructions for correct CAN communication setup.

10.3.1 Step 1: CANopen Master

Use one of the PC/CAN interface cards or PLCs listed below. For all of them, motion control libraries, examples and documentation are available on the Internet (for URLs ➔page 1-12).

Recommended Component	Manufacturer / Contact	Supported Product	maxon Motion Control Library
PC/CAN Interface Card*1)	**IXXAT** www.ixxat.de	All offered CANopen cards	Windows 32-Bit DLL
	Vector www.vector-informatik.de	All offered CANopen cards	Windows 32-Bit DLL
	National Instruments www.ni.com/can	All offered CANopen cards	Windows 32-Bit DLL
PLCs*2)	**Beckhoff** www.beckhoff.de	All offered CANopen cards	IEC 61131-3 Beckhoff Library
	Siemens www.siemens.com **Helmholz** www.helmholz.de	S7-300 with Helmholz CAN300 Master	Delivered and supported by Helmholz
	VIPA www.vipa.de	VIPA 214-2CM02 CAN-Master	IEC 61131-3 VIPA Library

Remarks:

*1) Interface driver of CANopen card must be installed!

*2) All CAN products of other manufacturers may also be used. However, no motion control library is available.

Table 10-131 CANopen Basic Information – recommended Components

10.3.2 Step 2: CAN Bus Wiring

The two-wire bus line must be terminated at both ends using a termination resistor of 120 Ω. Twisting is recommended, shielding is suggested (depending on EMC requirements).

EPOS2 Positioning Controller

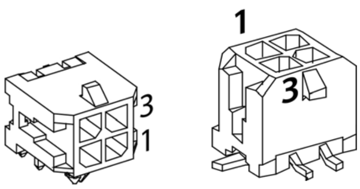
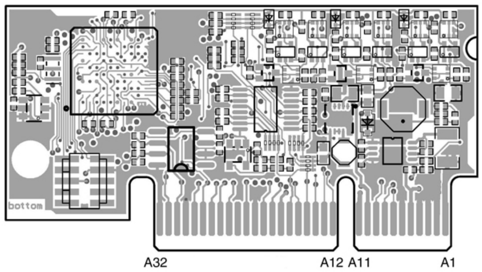
EPOS2 70/10 (375711) **EPOS2 50/5 (347717)** **EPOS2 24/5 (367676)** **EPOS2 24/2 (390438)(380264)(390003)(530239)**	**EPOS2 Module 36/2 (360665)**
Pin 1 “CAN high”	A31 “CAN high”
Pin 2 “CAN low”	A30 “CAN low”
Pin 3 “CAN GND”	A32 “CAN GND”
Pin 4 “CAN shield”	–
 CAN Connector Types	 Connector Array

Table 10-132 CAN Bus Wiring – Controller

CAN Bus Line

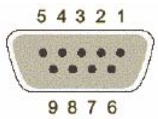
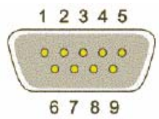
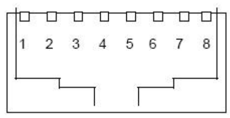
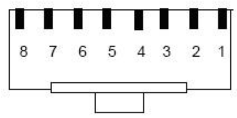
CAN 9 Pin D-Sub (DIN41652) on PLC **or** **PC/CAN Interface**		**CAN RJ45 on PLC** **or** **PC/CAN Interface**	
Pin 7 “CAN_H” high bus line		Pin 1 “CAN_H” high bus line	
Pin 2 “CAN_L” low bus line		Pin 2 “CAN_L” low bus line	
Pin 3 “CAN_GND” Ground		Pin 3 “CAN_GND” Ground Pin 7 “CAN_GND” Ground	
Pin 5 “CAN_Shield” Cable Shield		Pin 6 “CAN_Shield” Cable Shield	
Female 	Male 	Female 	Male 
D-Sub Connector		RJ45 Connector	

Table 10-133 CAN Bus Wiring – CAN Bus Line

10.3.3 Step 3: CAN Node ID

Generally applicable Rules

- *An unique Node ID (CAN ID) must be defined for all devices within the CAN network.*
- *The CAN ID results in the summed values of the stated DIP switches set to “1” (ON) or the connected input lines, respectively. The address can be coded using binary code.*
- *By setting all stated DIP switches to “0” (OFF) – or by letting the input lines open, respectively – the CAN IDs may be configured by software (changing the object “Node ID”). In this case, the number of addressable nodes is 127.*

10.3.3.1 EPOS2 70/10, EPOS2 50/5 & EPOS2 24/5 (DIP Switch 1…7, Addresses 1…127)

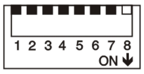
Switch	Binary Code	Valence	DIP Switch
1	2^0	1	 EPOS2 70/10 & EPOS2 24/5  EPOS2 50/5
2	2^1	2	
3	2^2	4	
4	2^3	8	
5	2^4	16	
6	2^5	32	
7	2^6	64	

Table 10-134 EPOS2 70/10, EPOS2 50/5 & EPOS2 24/5 – CAN ID

Examples

Use following table as a (non-concluding) guide:

CAN ID/Switch	1	2	3	4	5	6	7	
Valence	1	2	4	8	16	32	64	
CAN ID								**Calculation**
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	2
32	0	0	0	0	0	1	0	32
35	1	1	0	0	0	1	0	1 + 2 + 32
127	1	1	1	1	1	1	1	1 + 2 + 4 + 8 + 16 + 32 + 64

Table 10-135 DIP Switch 1…7 Settings (Example)

10.3.3.2 EPOS2 Module 36/2 (Input Line 1…7, Addresses 1…127)

Note

- *The set CAN ID can be observed by adding the valences of all inputs connected externally to GND.*
- *The CAN ID may also be configured by software if all input lines are open or externally connected to +3.3 VDC.*

Pin	Binary Code	Valence	Signal	Description
B24	–	–	GND	Ground for CAN ID settings
B25	2^0	1	CANID1	CAN ID 1
B26	2^1	2	CANID2	CAN ID 2
B27	2^2	4	CANID3	CAN ID 3
B28	2^3	8	CANID4	CAN ID 4
B29	2^4	16	CANID5	CAN ID 5
B30	2^5	32	CANID6	CAN ID 6
B31	2^6	64	CANID7	CAN ID 7

Table 10-136 EPOS2 Module 36/2 – CAN ID

For examples on DIP switch settings ➔Table 10-135.

10.3.3.3 EPOS2 24/2 (DIP Switch 1…4, Addresses 1…15)

Switch	Binary Code	Valence	DIP Switch
1	2^0	1	
2	2^1	2	
3	2^2	4	
4	2^3	8	

Table 10-137 EPOS2 24/2 – CAN ID

Examples:

Use following table as a (non-concluding) guide:

	CAN ID/Switch	1	2	3	4	
	Valence	1	2	4	8	
CAN ID	**DIP Setting**					**Calculation**
1		1	0	0	0	1
2		0	1	0	0	2
8		0	0	0	1	8
11		1	1	0	1	1 + 2 + 8
15		1	1	1	1	1 + 2 + 4 + 8

Table 10-138 Switch 1…4 Settings (Example)

10.3.4 Step 4: CAN Communication

For EPOS2, following CAN bit rates are available:

Object "CAN Bitrate" (Index 0x2001, Subindex 0x00)	Bit rate	Max. Line Length according to CiA 102
0	1 MBit/s	25 m
1	800 kBit/s	50 m
2	500 kBit/s	100 m
3	250 kBit/s	250 m
4	125 kBit/s	500 m
(5)	reserved	–
6	50 kBit/s	1000 m
7	20 kBit/s	2500 m
(8)	not supported (10 kBit/s)	–
9	automatic bit rate detection	–

Table 10-139 CAN Communication – Bit Rates and Line Lengths

Note

- *All devices within the CAN bus must use the same bit rate!*
- *The CANopen bus' maximum bit rate depends on the line length. Use «EPOS Studio» to configure bit rate by writing object "CAN Bit rate" (Index 0x2001, Subindex 0x00).*

10.3.5 Step 5: Activate Changes

Activate changes by saving and resetting the EPOS2 using «EPOS Studio».

1) Execute menu item ¤Save All Parameters¤.
2) Select context menu item ¤Reset Node¤ of the selected node.

10.3.6 Step 6: Communication Test

Use a CAN monitor program (supported by PC's or PLC CAN interface's manufacturer) to check wiring and configuration:

1) Reset all EPOS2 devices in the bus.
2) Upon power on, the EPOS2 will send a boot up message.
3) Make sure that all connected devices send a boot up message. If not, EPOS will produce a "CAN in Error Passive Mode".
4) Boot up message:
   COB-ID = 0x700 + Node ID
   Data [0] = 0x00

As an example, the figure below shows the incoming message on CAN bus (EPOS2 Node ID = 1) displayed by a CAN monitor supplied by IXXAT.

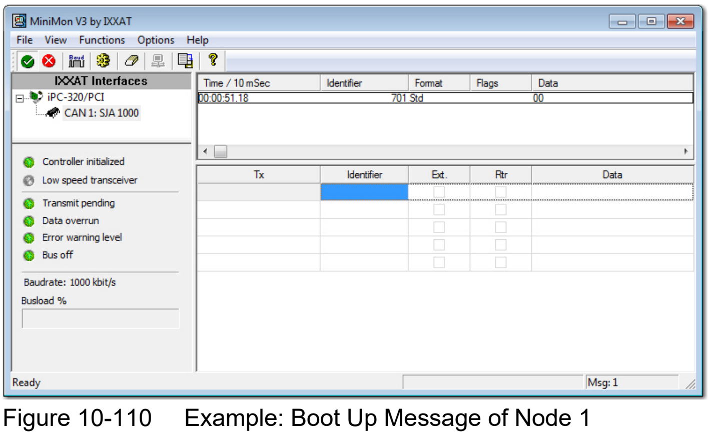


Figure 10-110 Example: Boot Up Message of Node 1

10.4 SDO Communication

A **Service Data Object (SDO)** reads from/writes to entries of the Object Dictionary. The SDO transport protocol allows transmission of objects of any size. SDO communication can be used to configure the EPOS2's object.

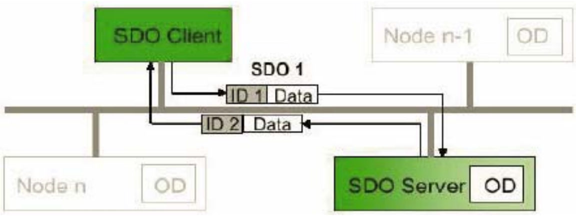


Figure 10-111 SDO Communication

Two different transfer types are supported:

- Normal transfer: A segmented SDO protocol used to read/write objects larger 4 bytes. This means that the transfer is split into different SDO segments (CAN frames).
- Expedited transfer: A non-segmented SDO protocol, used for objects smaller 4 bytes.

Almost all EPOS2 Object Dictionary entries can be read/written using the non-segmented SDO protocol (expedited transfer). Only the data recorder buffer must be read using the segmented SDO protocol (normal transfer). Thus, only non-segmented SDO protocol will be further explained. For details on the segmented protocol (normal transfer) ➔CANopen specification (CiA 301).

10.4.1 Expedited SDO Protocol

Reading Object

Client => Server

COB-ID	Data [Byte 0]	Data [Byte 1]	Data [Byte 2]	Data [Byte 3]	Data [Byte 4]	Data [Byte 5]	Data [Byte 6]	Data [Byte 7]
0x600 + Node-ID		Index LowByte	Index HighByte	Sub-Index	Reserved			

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	X	X	X	X	X

Server => Client

COB-ID	Data [Byte 0]	Data [Byte 1]	Data [Byte 2]	Data [Byte 3]	Data [Byte 4]	Data [Byte 5]	Data [Byte 6]	Data [Byte 7]
0x580 + Node-ID		Index LowByte	Index HighByte	Sub-Index	Object Byte 0	Object Byte 1	Object Byte 2	Object Byte 3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	X	n		e	s

Figure 10-112 SDO Upload Protocol (Expedited Transfer) – Read

Writing Object

Client => Server

COB-ID	Data [Byte 0]	Data [Byte 1]	Data [Byte 2]	Data [Byte 3]	Data [Byte 4]	Data [Byte 5]	Data [Byte 6]	Data [Byte 7]
0x600 + Node-ID		Index LowByte	Index HighByte	Sub-Index	Object Byte 0	Object Byte 1	Object Byte 2	Object Byte 3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1	X	n		e	s

Server => Client

COB-ID	Data [Byte 0]	Data [Byte 1]	Data [Byte 2]	Data [Byte 3]	Data [Byte 4]	Data [Byte 5]	Data [Byte 6]	Data [Byte 7]
0x580 + Node-ID		Index LowByte	Index HighByte	Sub-Index	Reserved			

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	X	X	X	X	X

Figure 10-113 SDO Upload Protocol (Expedited Transfer) – Write

Abort SDO Protocol (in Case of Error)

Server => Client

COB-ID	Data [Byte 0]	Data [Byte 1]	Data [Byte 2]	Data [Byte 3]	Data [Byte 4]	Data [Byte 5]	Data [Byte 6]	Data [Byte 7]
0x580 + Node-ID		Index LowByte	Index HighByte	Sub-Index	Abort Code			

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	X	X	X	X	X

Figure 10-114 SDO Upload Protocol (Expedited Transfer) – Abort

Note
For detailed descriptions of “Abort Codes” ➔FwSpec.

Legend	
ccs	client command specifier (Bit 7…5)
scs	server command specifier (Bit 7…5)
X	not used (always “0”)
n	Only valid if e = 1 and s = 1, otherwise 0. If valid, it indicates the number of bytes in Data [Byte 4…7] that do not contain data. Bytes [8 − n, 7] do not contain segment data.
e	Transfer type (0: normal transfer; 1: expedited transfer)
s	Size indicator (0: data set size is not indicated; 1: data set size is indicated)

Table 10-140 SDO Transfer Protocol – Legend

Overview on important Command Specifier ([Byte 0] → Bit 7…5)

Type	Length	Sending Data [Byte 0]	Receiving Data [Byte 0]
Reading Object	1 Byte	40	4F
	2 Byte	40	4B
	3 Byte	40	43
Writing Object	1 Byte	2F (or 22)	60
	2 Byte	2B (or 22)	60
	4 Byte	23 (or 22)	60
	not defined	22	60

Table 10-141 Command Specifier (Overview)

10.4.2 SDO Communication Examples

Read “Current Regulator P-Gain” (Index 0x60F6, Subindex 0x01) from node 1:

CANopen Sending SDO Frame			CANopen Receiving SDO Frame		
COD-ID	0x601	0x600 + Node ID	COD-ID	0x581	0x580 + Node ID
Data [0]	0x40	ccs = 2	Data [0]	0x4B	scs = 2, n = 2, e = 1, s = 1
Data [1]	0xF6	Index LowByte	Data [1]	0xF6	Index LowByte
Data [2]	0x60	Index HighByte	Data [2]	0x60	Index HighByte
Data [3]	0x01	Subindex	Data [3]	0x01	Subindex
Data [4]	0x00	reserved	Data [4]	0x90	P-Gain LowByte
Data [5]	0x00	reserved	Data [5]	0x01	P-Gain HighByte
Data [6]	0x00	reserved	Data [6]	0x00	reserved
Data [7]	0x00	reserved	Data [7]	0x00	reserved
			Current Regulator P-Gain: 0x00000190 = 400		

Table 10-142 Example “Read”

Write “Current Regulator P-Gain” (Index 0x60F6, Subindex 0x01) to node 1:

CANopen Sending SDO Frame			CANopen Receiving SDO Frame		
COD-ID	0x601	0x600 + Node ID	COD-ID	0x581	0x580 + Node ID
Data [0]	0x2B	ccs = 1, n = 2, e = 1, s = 1	Data [0]	0x60	scs = 3
Data [1]	0xF6	Index LowByte	Data [1]	0xF6	Index LowByte
Data [2]	0x60	Index HighByte	Data [2]	0x60	Index HighByte
Data [3]	0x01	Subindex	Data [3]	0x01	Subindex
Data [4]	0x12	P-Gain LowByte	Data [4]	0x00	reserved
Data [5]	0x34	P-Gain HighByte	Data [5]	0x00	reserved
Data [6]	0x00	reserved	Data [6]	0x00	reserved
Data [7]	0x00	reserved	Data [7]	0x00	reserved
			Current Regulator P-Gain: new value		

Table 10-143 Example “Write”

Read “Unknown Object” (Index 0x2000, Subindex 0x08) from node 1:

CANopen Sending SDO Frame			CANopen Receiving SDO Frame		
COD-ID	0x601	0x600 + Node ID	COD-ID	0x581	0x580 + Node ID
Data [0]	0x40	ccs =2	Data [0]	0x80	scs = 3
Data [1]	0x00	Index LowByte	Data [1]	0x00	Index LowByte
Data [2]	0x20	Index HighByte	Data [2]	0x20	Index HighByte
Data [3]	0x08	Subindex	Data [3]	0x08	Subindex
Data [4]	0x00	reserved	Data [4]	0x11	Abort Code [Byte 0]
Data [5]	0x00	reserved	Data [5]	0x00	Abort Code [Byte 1]
Data [6]	0x00	reserved	Data [6]	0x09	Abort Code [Byte 2]
Data [7]	0x00	reserved	Data [7]	0x06	Abort Code [Byte 3]
			Abort code: 0x06090011 → the last read or write command had a wrong object subindex.		

Table 10-144 Example “Read”

10.5 PDO Communication

Process Data Objects (PDOs) – unconfirmed services containing no protocol overhead – are used for fast data transmission (real-time data) with a high priority. Consequently, they represent an extremely fast and flexible method to transmit data from one node to any number of other nodes. PDOs may contain up to 8 data bytes that can be specifically compiled and confirmed to suit own requirements. Each PDO has a unique identifier and is transmitted by only one node, but it can be received by more than one (producer/consumer communication).

The CANopen network management is node-oriented and follows a master/slave structure. It requires one device in the network, which serves as **NMT (Network Management)** Master. The other nodes are NMT Slaves.

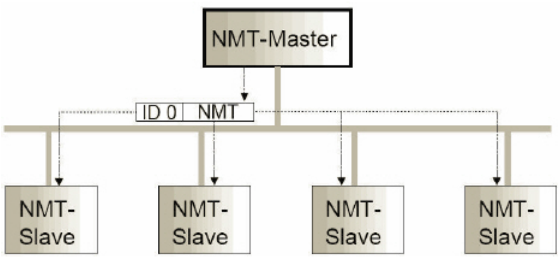


Figure 10-115 Network Management (NMT)

The CANopen NMT Slave devices implement a state machine that automatically brings every device to “Pre-Operational” state, once powered and initialized. In this state, the node may be configured and parameterized via SDO (e.g. using a configuration tool), PDO communication is not permitted. Thus, to switch from “Pre-Operational” to “Operational”, you will need to send the “Start Remote Node Protocol”. For detailed information on NMT Services ➔separate document «EPOS2 Communication Guide».

Function	COB-ID	CS (Byte 0)	Node ID (Byte 1)	Functionality
Start Remote Node Protocol	0	0x01	0 (all)	All EPOS2 (all CANopen nodes) will enter NMT State “Operational”.
	0	0x01	n	The EPOS2 (or CANopen node) with Node ID n will enter NMT State “Operational”.
Enter Pre-Operational Protocol	0	0x80	0 (all)	All EPOS2 (all CANopen nodes) will enter NMT State “Pre-Operational”.
	0	0x80	n	The EPOS2 (or CANopen node) with Node ID n will enter NMT State “Pre-Operational”.

Table 10-145 NMT Functionality

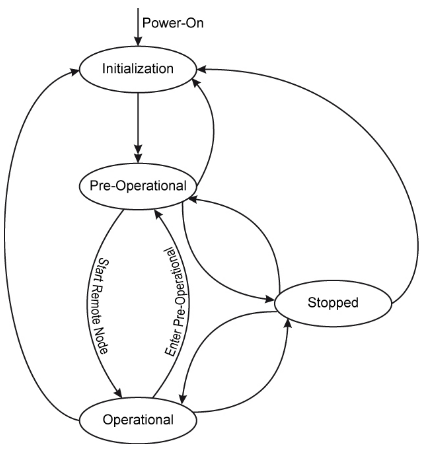


Figure 10-116 NMT Slave State Diagram

10.5.1 PDO Transmissions

PDO transmissions may be driven by remote requests, event triggered and actuated by Sync message received:

- Remotely requested:
  Another device may initiate the transmission of an asynchronous PDO by sending a remote transmission request (remote frame).
- Event triggered (only Transmit PDOs):
  An event of a mapped object (e.g. velocity changed) will cause the transmission of the TxPDO. Subindex 3h of object "Transmit PDO X Parameter" contains the inhibit time, which represents the minimum interval for PDO transmission. The value is defined as a multiple of 100 us.
- Synchronous transmission:
  In order to initiate simultaneous sampling of input values of all nodes, a periodically transmitted Sync message is required. Synchronous PDO transmission takes place in cyclic and acyclic transmission mode. Cyclic transmission means that the node waits for the Sync message after which it sends its measured values. Its PDO transmission type number (1…240) indicates the Sync rate it listens to (the number of Sync messages the node waits before next transmission of its values). The EPOS supports only Sync rates of 1.

10.5.2 PDO Mapping

Default application objects' mapping as well as the supported transmission mode is described in the Object Dictionary for each PDO. PDO identifiers may have high priority to guarantee short response time. PDO transmission is not confirmed. PDO mapping defines the application objects to be transmitted within a PDO. It describes sequence and length of the mapped application objects. A device supporting variable mapping of PDOs must support this during the Pre-Operational state. If dynamic mapping during Operational state is supported, the SDO Client is responsible for data consistency.

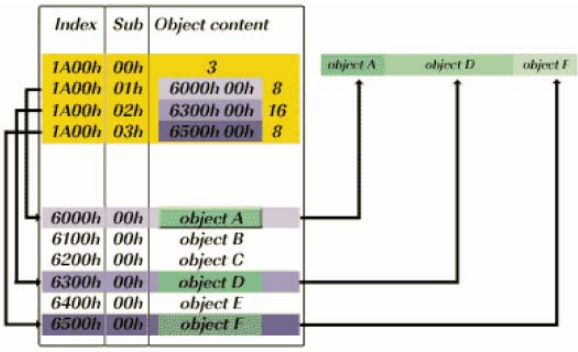


Figure 10-117 PDO Mapping

10.5.3 PDO Configuration

For PDO Configuration, the device must be in Pre-Operational state!

The following section will explain how to configuration must be implemented step-by-step. Use «EPOS Studio» for all changes in the Object Dictionary described below. For each step, an example quotes “Receive PDO 1” and “Node 1”.

10.5.3.1 Step 1: Configure COB-ID

The default value of the COB-ID depends on the Node ID (Default COB-ID = PDO-Offset + Node ID). Otherwise, the COB-ID can be set in a defined range. Below table shows all default COB-IDs and their ranges:

Object	Index	Subindex	Default COB-ID Node 1
TxPDO 1	0x1800	0x01	0x181
TxPDO 2	0x1801	0x01	0x281
TxPDO 3	0x1802	0x01	0x381
TxPDO 4	0x1803	0x01	0x481
RxPDO 1	0x1400	0x01	0x201
RxPDO 2	0x1401	0x01	0x301
RxPDO 3	0x1402	0x01	0x401
RxPDO 4	0x1403	0x01	0x501

Table 10-146 COB-IDs – Default Values and Value Range

Changed COB-IDs can be reset by “Restore Default PDO COB-IDs” using context menu of ¤Object Dictionary¤ view in «EPOS Studio».

Example: Object → “COB-ID used by RxPDO 1” (Index 0x1400, Subindex 0x01):

Default COB ID RxPDO 1 = 0x200 + Node ID = 0x201
In Range COB ID RxPDO 1 = 0x233

10.5.3.2 Step 2: Set Transmission Type

Type 0x01	TxPDOs	Data is sampled and transmitted after the occurrence of the SYNC.
	RxPDOs	Data is passed on to the EPOS2 and transmitted after the occurrence of the SYNC.
Type 0xFD	TxPDOs	Data is sampled and transmitted after the occurrence of a remote transmission request (RTR).
Type 0xFF	TxPDOs	Data is sampled and transmitted after the occurrence of a remote transmission request or an internal event (value changed).
	RxPDOs	Data is directly passed on to the EPOS2 application.

Example: Object → “Transmission Type” (Index 0x1400, Subindex 0x02)
Value = 0xFF

10.5.3.3 Step 3: Number of Mapped Application Objects

Disable the PDO by writing zero to object "Number of Mapped Application Objects in…".

Example: Object → "Number of Mapped Application Objects in RxPDO 1" (Index 0x1600, Subindex 0x00)
Value = 0x00

10.5.3.4 Step 4: Mapping Objects

Set value from an object.

Example: Object1 → "1st Mapped Object in RxPDO 1" (Index 0x1600, Subindex 0x01)
Object2 → "2nd Mapped Object in RxPDO 1" (Index 0x1600, Subindex 0x02)
Object3 → "3rd Mapped Object in RxPDO 1" (Index 0x1600, Subindex 0x03)

RxPDO 1	#	Mapped Object	
	1	Object_1 = 0x60400010	→ Controlword (16-bit)
	2	Object_2 = 0x607A0020	→ Target Position (32-bit)
	3	Object_3 = 0x60FB0210	→ Position Regulator I-Gain (16-bit)

Note
For details on all mappable objects ➔FwSpec, chapters "Receive PDO… Parameter" and "Transmit PDO… Parameter".

10.5.3.5 Step 5: Number of mapped Application Objects

Enable PDO by writing the value of the number of objects in object "Number of Mapped Application Objects in…".

Example: Object → "Number of Mapped Application Objects in RxPDO 1" (Index 0x1600, Subindex 0x00)

10.5.3.6 Step 6: Activate Changes

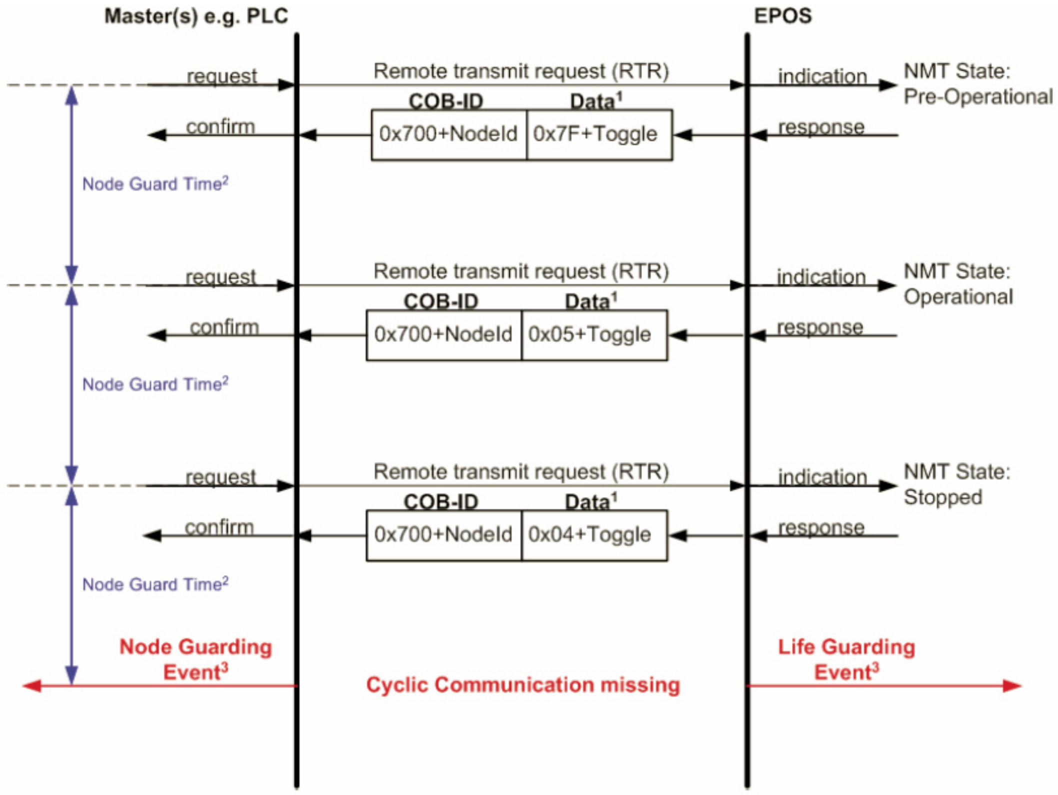
Changes will directly be activated.

Execute menu item ¤Save All Parameters¤ in the context menu of the used node («EPOS Studio» \ Navigation Window \ Workspace or Communication) or in the context menu in the view "Object Dictionary".

10.6 Node Guarding Protocol

Used to detect absent devices that do not transmit PDOs regularly (e.g. because of bus-off). The NMT Master can manage a database where, among other information, expected states of all connected devices are recorded, which is known as Node Guarding. With cyclic Node Guarding, the NMT Master regularly polls its NMT Slaves. To detect the absence of the NMT Master, the slaves test internally, whether Node Guarding is taking place in the defined time interval (Life Guarding).

Node Guarding is initiated by the NMT Master in Pre-Operational state of the slave by transmitting a Remote Frame. Node Guarding is also activated if Stopped State is active.



Legend: 1) Data Field / 2) Node Guard Time / 3) Node/Life Guarding Event

Figure 10-118 Node Guarding Protocol – Timing Diagram

Data Field

Holds the NMT State. Upon receipt of a node guard answer, bit 8 toggles between 0x00 and 0x80. Thus, the data field supports the following values:

Value	Toggle	EPOS2 NMT State
0x04	not set	Stopped
0x84	set	Stopped
0x05	not set	Operational
0x85	set	Operational
0x7F	not set	Pre-Operational
0xFF	set	Pre-Operational

Table 10-147 Node Guarding Protocol – Data Field

Node Guard Time

Is calculated as follows: $NodeGuardTime = GuardTime \cdot LifeTimeFactor$

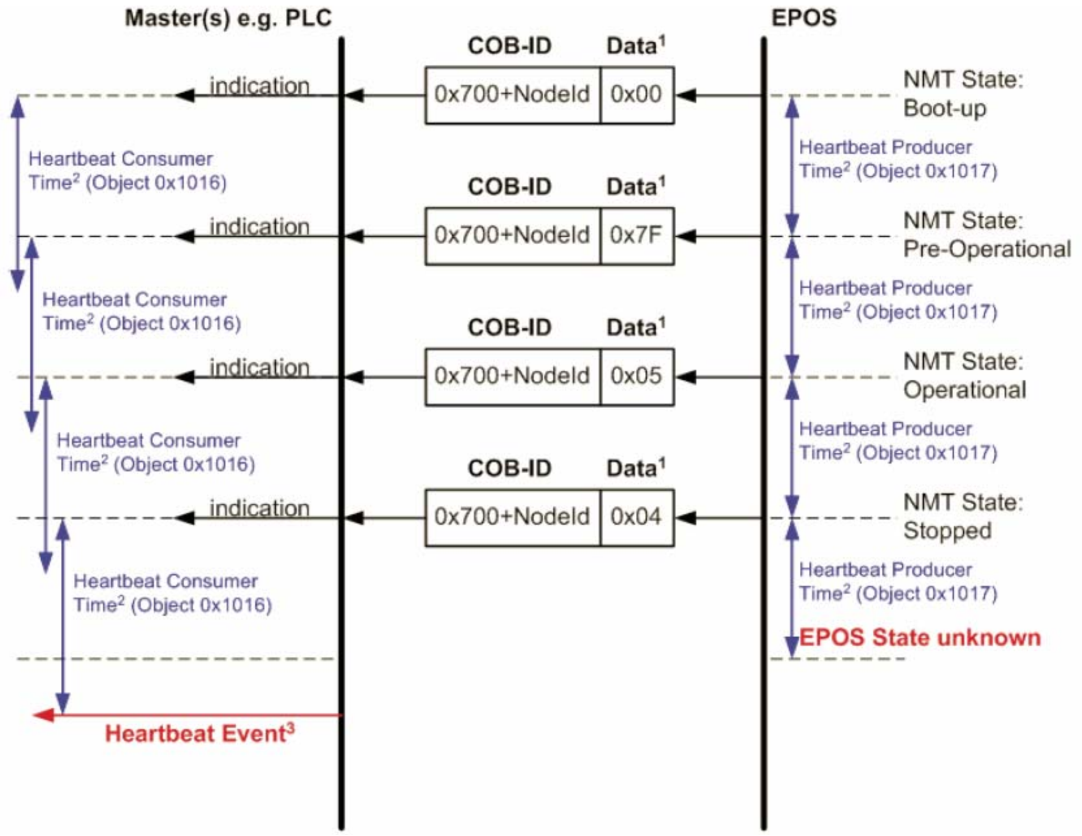
Node / Life Guarding Event

In case EPOS2 misses the Remote Transmit Request (RTR), it will change it’s device state to error (Node Guarding Error).

In case the answer is missed by the Master System, it may react with the Node Guarding Event.

10.7 Heartbeat Protocol

The Heartbeat Protocol has a higher priority than the Node Guarding Protocol, if both are enabled, only the Heartbeat Protocol is supported. The EPOS2 transmits a cyclic heartbeat message if the Heartbeat Protocol is enabled (Heartbeat Producer Time 0 = Disabled / greater than 0 = enabled). The Heartbeat Consumer guards receipt of the Heartbeat within the Heartbeat Consumer Time. If the Heartbeat Producer Time is configured in EPOS2, it will start immediately with the Heartbeat Protocol.



Legend: 1) Data Field / 2) Heartbeat Producer and Heartbeat Consumer Time / 3) Hearbeat Event

Figure 10-119 Heartbeat Protocol – Timing Diagram

Data Field

Holds the NMT State. Each time the value of toggle between 0x00 and 0x80. Therefore the following values for the data field are possible:

Value	EPOS2 NMT State
0x00	Bootup
0x04	Stopped
0x05	Operational
0xFF	Pre-Operational

Table 10-148 Heartbeat Protocol – Data Field

Heartbeat Producer Time and Heartbeat Consumer Time

The Heartbeat Consumer Time must be longer than the Heartbeat Producer Time because of generation, sending and indication time ($HeartbeatConsumerTime \geq HeartbeatProducerTime + 5ms$). Each indication of the Master resets the Heartbeat Consumer Time.

Heartbeat Event

If EPOS2 is in an unknown state (e.g. supply voltage failure), the Heartbeat Protocol cannot be sent to the Master. The Master will recognize this event upon elapsed Heartbeat Consumer Time and will generate a Heartbeat Event.

••page intentionally left blank••

11 USB or RS232 to CAN Gateway

11.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

For simple point-to-point communication, EPOS2 also supports an USB or RS232 interface. In order to access a network using USB or RS232 protocols, EPOS2 includes an USB-to-CANopen, respectively a RS232-to-CANopen gateway functionality.

11.1.1 Objective

The present Application Note explains the functionality of the built-in communication gateway USB to CANopen or RS232 to CANopen. Advantages and disadvantages of this communication structures are discussed.

Contents

11.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2110h	Firmware Specification Communication Guide
EPOS2 70/10	375711	2120h or higher	
EPOS2 50/5	347717	2110h or higher	
EPOS2 Module 36/2	360665	2110h or higher	
EPOS2 24/5	367676	2110h or higher	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	

Table 11-149 USB or RS232 to CAN Gateway – covered Hardware and required Documents

11.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 11-150 USB or RS232 to CAN Gateway – recommended Tools

11.2 Communication Structure

Using the gateway functionality, the master can access all other EPOS2 devices connected to the CAN Bus via USB port or RS232 interface of the gateway device. Even other CANopen devices (I/O modules) supporting the CANopen CiA 301 may be accessed.

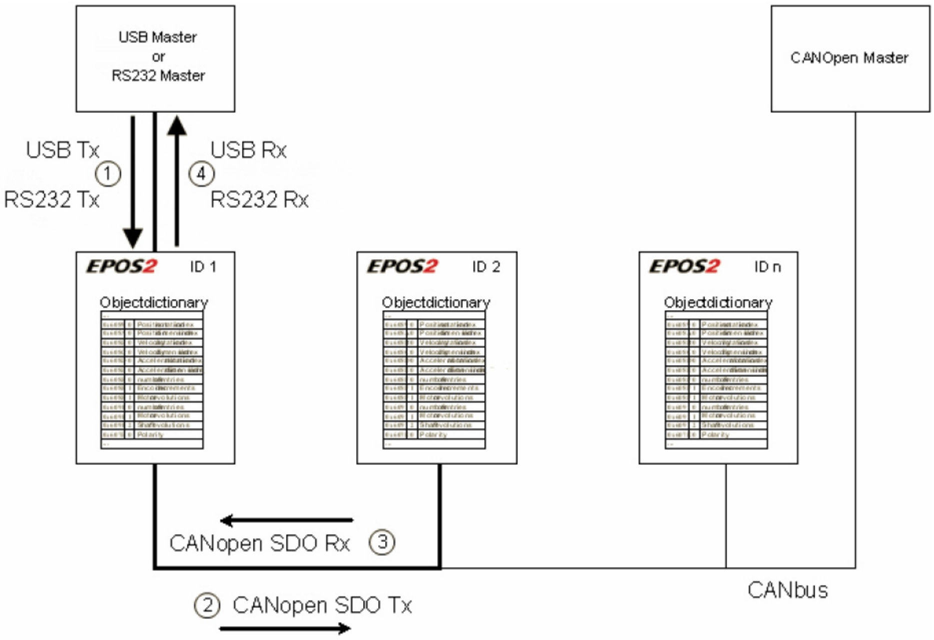


Figure 11-120 Gateway Communication Structure

Communication data are exchanged between USB/RS232 master and the gateway using a maxon-specific USB/RS232 protocol. The data between the gateway and the addressed device are exchanged using the CANopen SDO protocol according to the CiA 301.

For details on CAN bus wiring ➔chapter “10 CANopen Basic Information” on page 10-147.

Step	Protocol	Sender → Receiver	Description
1	USB [maxon-specific] or RS232 [maxon-specific]	USB or RS232 Master ↓ EPOS2 ID 1, Gateway	Command including the node ID is sent to the device working as a gateway. The gateway decides whether to execute the command or to translate and forward it to the CAN bus. **Criteria:** Node ID = 0 (Gateway) → Execute Node ID = DIP switch → Execute else → Forward to CAN
2	CANopen [SDO]	EPOS2 ID 1, Gateway ↓ EPOS2 ID 2	The gateway is forwarding the command to the CAN network. The USB/RS232 command is translated to a CANopen SDO service.
3	CANopen [SDO]	EPOS2 ID 2 ↓ EPOS2 ID 1, Gateway	The EPOS2 ID 2 is executing the command and sending the corresponding CAN frame back to the gateway.
4	USB [maxon-specific] or RS232 [maxon-specific]	EPOS2 ID 1, Gateway ↓ USB or RS232 Master	The gateway is receiving the CAN frame corresponding to the SDO service. This CAN frame is translated back to the USB/RS232 frame and sent back to the USB/RS232 master.

Table 11-151 Communication Data Exchange

11.3 Communication Examples

The examples employ following abbreviations:

Legend	
ccs	client command specifier (Bit 7…5)
scs	server command specifier (Bit 7…5)
X	not used (always “0”)
n	Only valid if e = 1 and s = 1, otherwise 0. If valid, it indicates the number of bytes in Data [Byte 4…7] that do not contain data. Bytes [8 − n, 7] do not contain segment data (Bit 3 and 2).
e	Transfer type (0: normal transfer; 1: expedited transfer) (Bit 1)
s	Size indicator (0: data set size is not indicated; 1: data set size is indicated) (Bit 0)

Table 11-152 SDO Transfer Protocol – Legend

11.3.1 USB

Object:	DeviceType, Index 0x1000, Subindex 0x00
Node:	2
USB Command:	ReadObject
CANopen Service:	SDO Upload (Expedited Transfer)

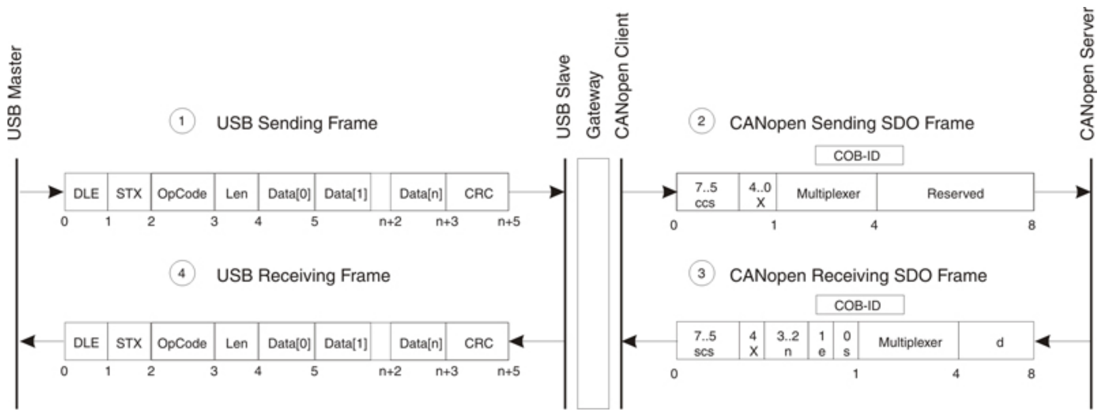


Figure 11-121 Communication via USB (Example)

Step 1: USB Sending Frames		
DLE	0x90	Data Link Escape
STX	0x02	Start of Text
OpCode	0x10	ReadObject command
Len	0x02	2 Data Words
Data[0]	0x00	Index Byte 0
Data[1]	0x10	Index Byte 1
Data[2]	0x00	Sub Index
Data[3]	0x02	Node Id
CRC[0]	0xDF	Checksum Byte 0
CRC[1]	0xF2	Checksum Byte 1

Step 2: CANopen Sending SDO Frame		
COB-ID	0x602	0x600 + Node ID
Data[0]	0x40	ccs = 2
Data[1]	0x00	Index Byte 0
Data[2]	0x10	Index Byte 1
Data[3]	0x00	Sub Index
Data[4]	0x00	reserved
Data[5]	0x00	reserved
Data[6]	0x00	reserved
Data[7]	0x00	reserved

Table 11-153 Communication via USB (Example) – Steps 1/2

Step 4: USB Receiving Frame		
DLE	0x90	Data Link Escape
STX	0x02	Start of Text
OpCode	0x00	Answer to ReadObject
Len	0x04	4 Data Words
Data[0]	0x00	ErrorCode Byte 0
Data[1]	0x00	ErrorCode Byte 1
Data[2]	0x00	ErrorCode Byte 2
Data[3]	0x00	ErrorCode Byte 3
Data[4]	0x92	DeviceType Byte 0
Data[5]	0x01	DeviceType Byte 1
Data[6]	0x02	DeviceType Byte 2
Data[7]	0x00	DeviceType Byte 3
CRC[0]	0x9A	Checksum Byte 0
CRC[1]	0xED	Checksum Byte 1

Step 3: CANopen Receiving SDO Frame		
COB-ID	0x582	0x580 + Node ID
Data[0]	0x43	scs = 2, n = 0, e = 1, s = 1
Data[1]	0x00	Index LowByte
Data[2]	0x10	Index HighByte
Data[3]	0x00	Sub Index
Data[4]	0x92	DeviceType Byte 1
Data[5]	0x01	DeviceType Byte 2
Data[6]	0x02	DeviceType Byte 3
Data[7]	0x00	DeviceType Byte 4

Table 11-154 Communication via USB (Example) – Steps 3/4

11.3.2 RS232

Object:	DeviceType, Index 0x1000, Subindex 0x00
Node:	2
USB Command:	ReadObject
CANopen Service:	SDO Upload (Expedited Transfer)

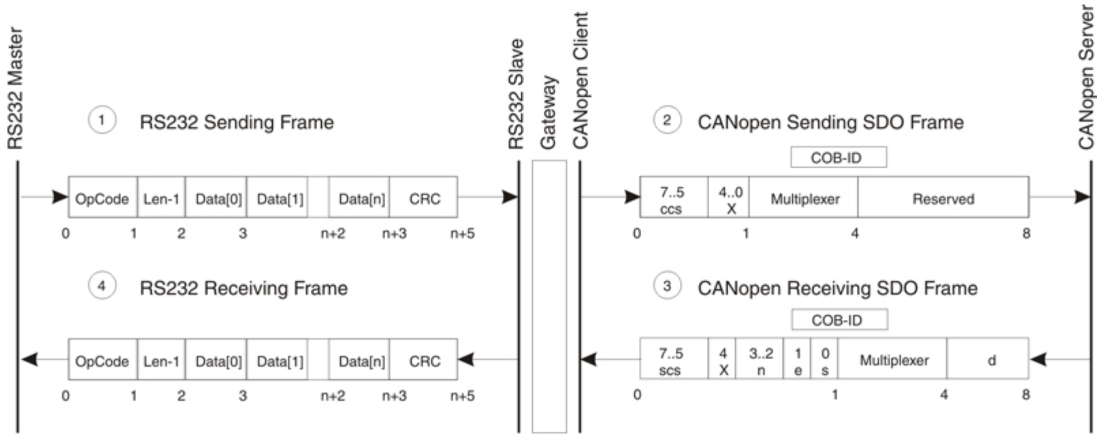


Figure 11-122 Communication via RS232 (Example)

Step 1: RS232 Sending Frames		
OpCode	0x10	ReadObject command
Len-1	0x01	2 Data Words
Data[0]	0x00	Index Byte 0
Data[1]	0x10	Index Byte 1
Data[2]	0x00	Sub Index
Data[3]	0x02	Node Id
CRC[0]	0x10	Checksum Byte 0
CRC[1]	0xCD	Checksum Byte 1

Step 2: CANopen Sending SDO Frame		
COB-ID	0x602	0x600 + Node ID
Data[0]	0x40	ccs = 2
Data[1]	0x00	Index Byte 0
Data[2]	0x10	Index Byte 1
Data[3]	0x00	Sub Index
Data[4]	0x00	reserved
Data[5]	0x00	reserved
Data[6]	0x00	reserved
Data[7]	0x00	reserved

Table 11-155 Communication via RS232 (Example) – Steps 1/2

Step 4: RS232 Receiving Frame		
OpCode	0x00	Answer to ReadObject
Len-1	0x03	4 Data Words
Data[0]	0x00	ErrorCode Byte 0
Data[1]	0x00	ErrorCode Byte 1
Data[2]	0x00	ErrorCode Byte 2
Data[3]	0x00	ErrorCode Byte 3
Data[4]	0x92	DeviceType Byte 0
Data[5]	0x01	DeviceType Byte 1
Data[6]	0x02	DeviceType Byte 2
Data[7]	0x00	DeviceType Byte 3
CRC[0]	0xEB	Checksum Byte 0
CRC[1]	0x6D	Checksum Byte 1

Step 3: CANopen Receiving SDO Frame		
COB-ID	0x582	0x580 + Node ID
Data[0]	0x43	scs = 2, n = 0, e = 1, s = 1
Data[1]	0x00	Index LowByte
Data[2]	0x10	Index HighByte
Data[3]	0x00	Sub Index
Data[4]	0x92	DeviceType Byte 0
Data[5]	0x01	DeviceType Byte 1
Data[6]	0x02	DeviceType Byte 2
Data[7]	0x00	DeviceType Byte 3

Table 11-156 Communication via RS232 (Example) – Steps 3/4

11.4 Command Translation

The USB/RS232 command set is designed approximate to CANopen services. All USB/RS232 commands have a directly corresponding service in the CAN network, thus simplifying the gateway functionality. Between two subsequent USB/RS232 commands, no data must be stored or buffered, thus minimizing Gateway's memory use. All received data are directly forwarded to the CAN bus.

USB/RS232 Command		CANopen Service
ReadObject	→	Initiate SDO Upload / Expedited Transfer
InitiateSegmentedRead	→	Initiate SDO Upload / Normal Transfer
SegmentRead	→	Upload SDO Segment
WriteObject	→	Initiate SDO Download / Expedited Transfer
InitiateSegmentedWrite	→	Initiate SDO Download / Normal Transfer
SegmentWrite	→	Download SDO Segment
SendNMTService	→	NMT Service
ReadLSSFrame	→	LSS Service
SendLSSFrame	→	LSS Service

Table 11-157 Command Translation – USB/RS232 to CANopen Service

11.5 Limiting Factors

The number of segments has a big influence on the data exchange performance. Exchanging data directly with a device connected to RS232 (no gateway), a data segment can transfer up to 63 Bytes per command, thus for 1kB of data, 17 commands must be sent. Compared to sending data to a device addressed via gateway, 147 commands must be sent. CANopen services (normal transfer) allow only 7 bytes to be transferred in a segment. Therefore, the CANopen segment limits also the RS232 segment. Please keep in mind; the gateway is not capable of buffering data nor to split data into several CANopen services.

Considering the segment size, CANopen is the limiting factor for the communication performance. Considering the bit rate of the two field buses, the RS232 interface is the limiting factor. Communication via gateway cannot take advantage of the CAN bus' high bit rate, it is limited by the RS232's slow bit rate and the small CANopen segment size.

Description	USB Protocol	RS232 Protocol	CANopen	USB to CANopen Gateway	RS232 to CANopen Gateway
Max. bit rate	1 MBit/s	115.2 kBit/s	1 MBit/s	1 MBit/s	115.2 kBit/s
Max. segment size	63 Bytes	63 Bytes	7 Bytes	7 Bytes	7 Bytes
Conclusion					
Transfer Rate	Fast	Slow	Fast	Fast	Slow
Segment Size	Big	Big	Small	Small	Small

Table 11-158 USB or RS232 to CAN Gateway – Limiting Factors

However, these limiting factors must be put into perspective, because most of the elements in the Object Dictionary are 32-bit parameters, or even smaller. Thus, segmented transfer is used very rarely. Segmented transfer will only be used to read the data recorder's data buffer or for firmware download.

11.6 Timing

11.6.1 RS232

The primary bottleneck in communication via RS232 to CANopen gateway is the RS232 bit rate. The maximum RS232 bit rate (115.2 kBit/s) is ten times smaller than the maximum CAN bit rate (1 MBit/s). The duration of the communication depends more or less on the RS232 bit rate used. The following timing example shows communication delaying for addressing a device via the gateway.

Example

Test Platform	Pentium 4, 2.66 GHz, Windows XP, EPOS_UserInterface
Command	ReadObject, 32-Bit Object
RS232 Bit rate	38400 Bit/s (Default)
CAN Bit rate	1 MBit/s (Default)
Time via Gateway	10.125 ms (measured)
Time without Gateway	9.995 ms (measured)
Delay	130 µs

Table 11-159 RS232 to CAN Gateway – Timing

11.6.2 Timing Values

Measured values are based on PC using IXXAT card with driver VCI3.

CAN	Read/Write 8-bit / 16-bit / 32-bit object (2 CAN frames @ 8/8 bytes)		Read/Write (200 CAN frames @ 8/8 bytes)	
Bit Rate	**Calculated**	**Measured**	**Calculated**	**Measured**
1 MBit/s	220 µs	794 µs	44 ms	159 ms
800 kBit/s	275 µs	850 µs	55 ms	170 ms
500 kBit/s	440 µs	1.0 ms	88 ms	204 ms
250 kBit/s	880 µs	1.5 ms	196 ms	307 ms
125 kBit/s	1.8 ms	2.4 ms	360 ms	488 ms
50 kBit/s	4.4 ms	5.3 ms	880 ms	1052 ms
20 kBit/s	11 ms	12.4 ms	2.2 s	2.5 s

Table 11-160 Timing – CAN Bus (CANopen SDO Services)

USB	Read/Write 8-bit / 16-bit / 32-bit object (2 USB frames @ 10/14 bytes)		Read/Write (200 2 USB frames @ 10/14 bytes)	
Bit Rate	**Calculated**	**Measured**	**Calculated**	**Measured**
1 MBit/s	2 ms	2.3 ms	400 ms	474 ms

Table 11-161 Timing – USB

RS232	Read/Write 8-bit / 16-bit / 32-bit object (2 RS232 frames @ 10/14 bytes)		Read/Write (200 RS232 frames @ 10/14 bytes)	
Bit Rate	Calculated	Measured	Calculated	Measured
115200 Bit/s	2.083 ms	3.9 ms	0.42 s	0.8 s
57600 Bit/s	4.16 ms	7.2 ms	0.83 s	1.4 s
38400 Bit/s	6.25 ms	10.4 ms	1.25 s	2.1 s
19200 Bit/s	12.5 ms	20.5 ms	2.5 s	4.1 s
14400 Bit/s	16.6 ms	27.2 ms	3.33 s	5.5 s
9600 Bit/s	25.0 ms	40.7 ms	5.0 s	8.2 s

Table 11-162 Timing – RS232 (maxon-specific protocol)

11.7 Conclusion

The gateway functionality enables easy connection to the CAN network without the need of a separate CAN interface card to monitor a CAN network. Also, wiring of the CAN network does not require alteration. By simply plugging the USB or RS232 cable into one of the EPOS2 Positioning Controllers, all other EPOS2 devices in the network can be controlled and monitored.

The delay in CAN communication can be neglected when considering the time needed with RS232 baud rate. Thus, the gateway does not slow down the RS232 communication. Thereby, it does not really make any difference (except in segmented transfers) whether the master is addressing a device in the CAN network directly via RS232 or via the gateway.

••page intentionally left blank••

12 Data Recording

12.1 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 CANopen interface allows networking to multiple axes drives as well as online commanding by CAN bus master units.

EPOS and EPOS2 both feature a built-in data recorder to debug errors and to monitor motion control parameters and actual values.

12.1.1 Objective

The present Application Note explains the functionality of the built-in data recorder. Features and configuration options are explained.

Contents

12.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		all	Firmware Specification
EPOS2 70/10	375711	all	
EPOS2 50/5	347717	all	
EPOS2 Module 36/2	360665	all	
EPOS2 24/5	367676	all	
EPOS2 24/2	380264 390003 390438 530239	2121h or higher	
EPOS		all	Firmware Specification
EPOS 70/10	300583	all	
EPOS 24/1	280937 302267 302287 317270	all	
EPOS P 24/5	323232	all	
MCD EPOS 60 W	326343	all	
MCD EPOS P 60 W	315665	all	

Table 12-163 Data Recording – covered Hardware and required Documents

12.1.3 Tools

Tools	Description
Software	«EPOS Studio» Version 2.00 or higher

Table 12-164 Data Recording – recommended Tools

12.2 Overview

12.2.1 Launching the Data Recorder

1) Start «EPOS Studio».
2) Start Data Recorder – either click right ¤Selected Node¤ or click ¤Tools¤ in Navigation Window.
3) Following screen will be displayed:

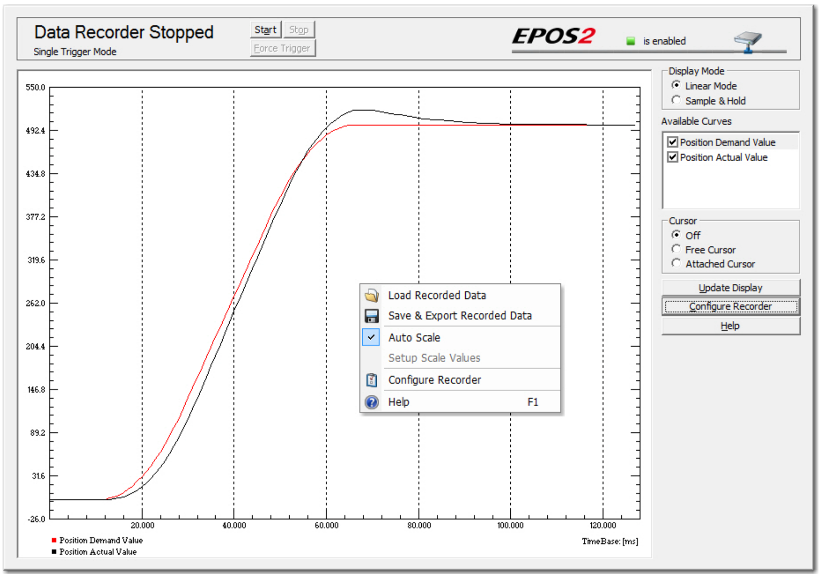


Figure 12-123 Data Recorder Overview

12.2.2 Control Elements and their Function

Title Bar

Control Element	Description / Function	
Status	Displays data recorder's status. The following states are possible:	
	Data Recorder Running Continuous Acquisition Mode	Data are continuously recorded and displayed.
	Data Recorder Waiting Single Trigger Mode	On standby, waiting to receive a trigger to start a single data record (for trigger options ➔ page 12-183).
	Data Recorder Triggered Single Trigger Mode	Sampling in process until data buffer is full.
	Data Recorder Stopped Single Trigger Mode or Continuous Acquisition Mode	Recording completed and stopped, results are being displayed.
Start	Commences sampling. In "Single Trigger Mode", the data recorder is waiting for a trigger. In "Continuous Acquisition Mode", the data recorder is continuously recording and displaying data.	
Stop	Stops sampling. Latest recorded data are being displayed.	
Force trigger	A trigger has been activated.	

Table 12-165 Data Recording – Title Bar

Options Bar

Control Element	Description / Function	
Display Mode	Linear Mode	To display data, linear interpolation will be used.
	Sample & Hold	Between samples, no interpolation will be used.
Available Curves	Available curves will be listed. Tick check box to show/untick to hide a curve in the display.	
Cursor	Off	No curser will be displayed.
	Free Cursor	Curser will appear, as soon as the mouse is moved.
	Attached Cursor	Moving the mouse will attach the cursor to the selected curve. Use "Available Curves" to select another curve.
Update Display	Last sampled data will be loaded and displayed.	
Configure Recorder	To select sampled data and to configure the data recorder (➔"Data Recorder Configuration" on page 12-181).	

Table 12-166 Data Recording – Option Bar

Display

Control Element	Description / Function
Zoom	**Zoom in:** Click left and draw a rectangle over desired area – status indication (upper left corner) will change to “Zoomed”. **Zoom out:** Click right – status indication will disappear.
Cursor	If activated, the cursor will appear as small circle. Cursor’s actual coordinates are displayed in the upper right corner.
Left / Right Scale	Each data set may be displayed in either left or right pane (➔Data Recorder Configuration).
Time Scale	At bottom border with corresponding time base at lower right corner.
Legend	Currently displayed curves’ legend appears in lower left corner.

Table 12-167 Data Recording – Display

Context Menu

Control Element	Description / Function	
Load Recorded Data	Load recorded data from file (*.rda).	
Save & Export Recorded Data	Save recorded data to file in following file formats:	
	*.rda	Binary Format (for use with «EPOS Studio»)
	*.txt	ASCII Text Format (for import in Microsoft Excel)
	*.csv	Comma Separated Values (for import in Microsoft Excel)
	*.bmp	Bitmap Format
Auto Scale	Select this option to automatically calculate optimal scale values.	
Setup Scale Values	If “Auto Scale” is deselected, left/right pane and time scale can be defined manually.	
Manual	Open connected device’s online help manual.	
Configure Recorder	To select sampled data and to configure data recorder (➔Data Recorder Configuration).	

Table 12-168 Data Recording – Context Menu

12.3 Data Recorder Configuration

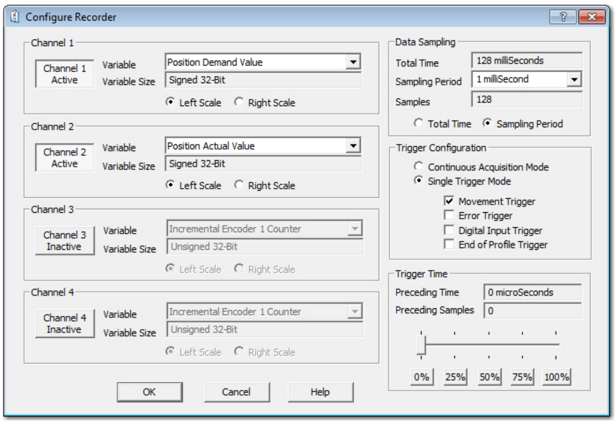


Figure 12-124 Data Recording – “Configure Recorder” Dialog

Channel 1…4

Control Element	Description / Function
Channel Active/ Inactive	Activate/deactivate up to four recorder channels.
Variable	Select desired variables to be recorded.
Variable Size	Displays size of selected variable.
Left / Right Scale	Select pane to display the recorded data.

Table 12-169 “Configure Recorder” – Channel

Data Sampling

Control Element	Description / Function
Total Time	Displays total duration.
Sampling Period	Select sampling period.
Samples	Displays number of samples per channel.
Total Time or Sampling Period	Select whether to determine the total time or the sampling period.

Table 12-170 “Configure Recorder” – Data Sampling

Trigger Configuration

Control Element	Description / Function	
Continuous Acquisition Mode	Data will continuously be recorded.	
Single Trigger Mode	Movement Trigger	A trigger is activated upon every start of a movement.
	Error Trigger	A trigger is activated upon an occurring error.
	Digital Input Trigger	A trigger is activated at an edge of a digital input. **Note:** In “Homing Mode”, also the current threshold will be interpreted as a trigger.
	End of Profile Trigger	A trigger is activated at the end of a movement profile.

Table 12-171 “Configure Recorder” – Trigger Configuration

Trigger Time

Control Element	Description / Function
Preceding Time	The lead time to be displayed prior activation of a trigger. “100%” permits display of data prior the trigger. **Best Practice:** Use the trigger time in combination with the error trigger to debug errors.
Preceding Samples	Displays the number of samples before the trigger.

Table 12-172 “Configure Recorder” – Trigger Time

12.4 Example: Data Recording in “Profile Position Mode”

12.4.1 Step 1: Configure Data Recorder

1) Click ¤Configure Recorder¤ in the options bar or select ¤Configure Recorder¤ from the context menu.

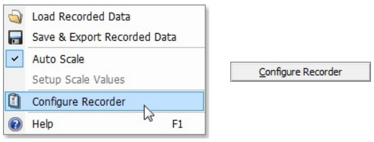


Figure 12-125 Configure Data Recorder

2) Select the following variables:
   - Position Demand Value
   - Position Actual Value
   - Velocity Actual Value
   - Current Actual Value
3) Select a sampling period of 1 ms.
4) Select ¤Single Trigger Mode¤ and tick ¤Movement Trigger¤.
5) Select a preceding time of 0 microseconds.

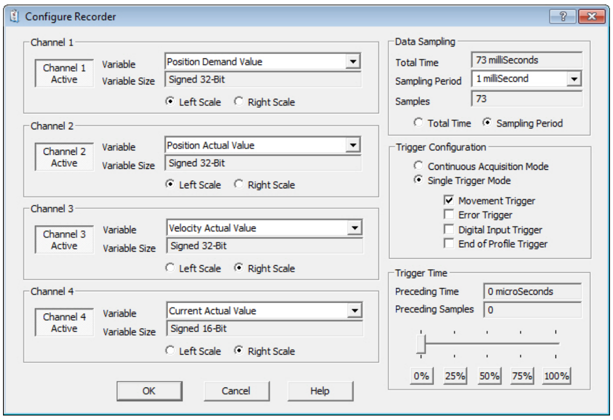


Figure 12-126 Select Configuration Options

6) Click ¤OK¤ to save settings.

12.4.2 Step 2: Execute Movement

1) Change the active view to "Profile Position Mode".
2) Activate "Profile Position Mode".
3) Enable the EPOS and start a relative movement.

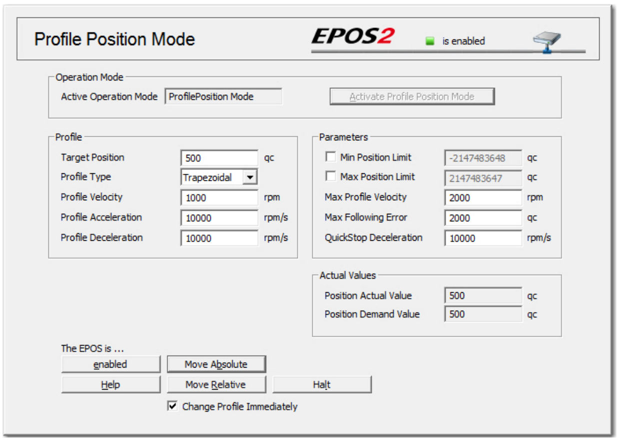


Figure 12-127 Execute Movement

12.4.3 Step 3: Update Display

Change back to the view "Data Recording". If the display does not automatically refresh, press ¤Update Display¤ button.

12.4.4 Step 4: Save recorded Data

1) Click right ¤Save & Export Recorded Data¤ to open context menu.

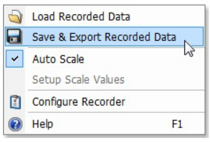


Figure 12-128 Save recorded Data

2) Select desired path.
3) Enter a file name.
4) Press ¤Save¤.

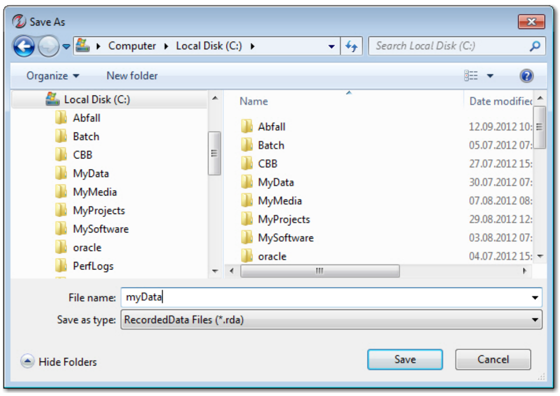


Figure 12-129 Save recorded Data

Best Practice
Save recorded data as ASCII text file or as bitmap!

12.4.5 Step 5: Analyze recorded Data

1) Select cursor mode ¤Attached Cursor¤.
2) Tick ¤Position Actual Value¤ in “Available Curves”.
3) Move cursor over the display and read the attached curve's values.

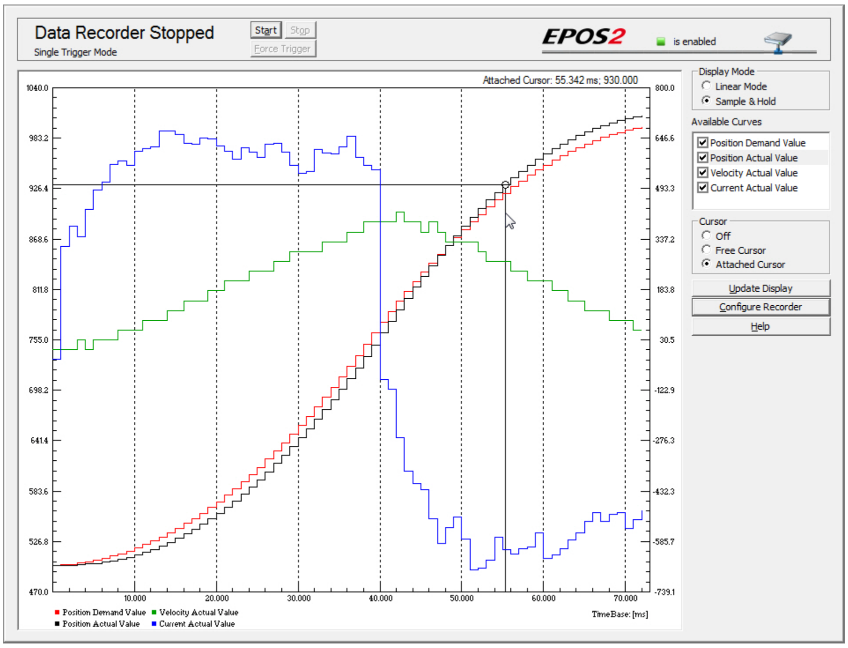


Figure 12-130 Analyze recorded Data

12.4.6 Step 6: Restart Data Recorder

Click ¤Start¤ to restart and prepare the data recorder for the next recording.

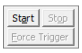


Figure 12-131 Restart Data Recorder

12.5 Data Recorder Specifications

12.5.1 Functionalities

Recorder

- Executed in current regulator (max 10 kHz sampling rate)
- Configurable sampling rate
- Total buffer size: 512 words

While the data recorder is running, data are sampled to a ring buffer until a trigger is set. After a trigger, data will be recorded until the buffer is full.

Variables

- Max. four variables of the Object Dictionary
- 16-bit and 32-bit variables are allowed (one word)
- 8-bit variables need 16-bits in the data recorder memory

Trigger

Following automatic trigger modes are supported:

- Manuel Trigger – set by communication
- Movement Trigger – set at movement start
- Error Trigger – set by error
- Digital Input Trigger – set by digital input
- End of Profile Trigger – set at movement stop

12.5.2 Object Description

12.5.2.1 Data Recorder Control

Description

The data recorder is controlled by write access.

Name	Data Recorder Control	
Index	0x2010	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	0	
Value Range	0	3

Bit	Description
15…2	reserved
1	0 = no trigger 1 = force trigger
0	0 = stop recorder 1 = start recorder

Table 12-173 Data Recorder Control – Bits

12.5.2.2 Data Recorder Configuration

Description

Configures the auto trigger functions.

Name	Data Recorder Configuration
Index	0x2011
Subindex	0x00
Type	UNSIGNED16
Access	RW
Default Value	0
Value Range	➔Table 12-174

Bit	Description
15…4	reserved
3	1 = trigger at end of profile
2	1 = trigger upon digital input
1	1 = trigger by error state
0	1 = trigger at movement start

Table 12-174 Data Recorder Configuration – Bits

12.5.2.3 Data Recorder Sampling Period

Description

Sampling period as a multiple of the current regulator cycle (n multiplied by 0.1ms).

Name	Data Recorder Sampling Period	
Index	0x2012	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	10	
Value Range	0	65535

12.5.2.4 Data Recorder Number of Preceding Samples

Description

Number of preceding samples defines the trigger position in the data recorder buffer.

Name	Data Recorder Number of Preceding Samples	
Index	0x2013	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	0	
Value Range	0	65535

12.5.2.5 Data Recorder Number of Sampling Variables

Description

Number of variables (max. 4) to be recorded.

Name	Data Recorder Number of Sampling Variables	
Index	0x2014	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	0	
Value Range	0	4

12.5.2.6 Data Recorder Index of Variables

Description

Index of Object Dictionary.

Related Objects

➔Data Recorder Subindex of Variables

Name	Data Recorder Index of Variables
Index	0x2015
Number of entries	0x05

Names	Data Recorder Index of Variable 1 Data Recorder Index of Variable 2	Data Recorder Index of Variable 3 Data Recorder Index of Variable 4
Index	0x2015	
Subindex	0x01…0x04	
Type	UNSIGNED16	
Access	RW	
Default Value	0	
Value Range	➔Object Dictionary	

12.5.2.7 Data Recorder Subindex of Variables

Description

Subindex of Object Dictionary.

Related Objects

➔Data Recorder Index of Variables

Name	Data Recorder Subindex of Variables
Index	0x2016
Number of entries	0x05

Names	Data Rec… Subindex of Variable 1 Data Rec… Subindex of Variable 2	Data Rec… Subindex of Variable 3 Data Rec… Subindex of Variable 4
Index	0x2016	
Subindex	0x01…0x04	
Type	UNSIGNED16	
Access	RW	
Default Value	0	
Value Range	➔Object Dictionary	

12.5.2.8 Data Recorder Status

Description

Data recorder’s status.

Name	Data Recorder Status
Index	0x2017
Subindex	0x00
Type	UNSIGNED16
Access	RO
Default Value	0
Value Range	➔Table 12-175

Bit	Description
15…2	reserved
1	0 = not triggered 1 = triggered
0	0 = stopped 1 = running

Table 12-175 Data Recorder Status – Bits

12.5.2.9 Data Recorder Max. Number of Samples

Description

Defines the maximal number of samples per variable. The parameter is dynamically calculated by the data recorder.

The maximal number of samples is the memory size (512 words) divided by the sum of the variable size (in words) of all configured variables.

Name	Data Recorder max. Number of Samples	
Index	0x2018	
Subindex	0x00	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

Example:

Sum of Variable Size [word]	Example	Number of Samples
1	1 x 16-bit variable or 1 x 8-bit variable	512
2	1 x 32-bit variable	256
3	1 x 16-bit and 1 x 32-bit variable	170
…	…	…
8	4 x 32-bit variables	64

Table 12-176 Data Recorder Max. Number of Samples – Example

12.5.2.10 Data Recorder Number of recorded Samples

Description

Represents the number of already recorded data samples.

Name	Data Recorder Number of recorded Samples	
Index	0x2019	
Subindex	0x00	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

12.5.2.11 Data Recorder Vector Start Offset

Description

Offset to the start of the recorded data vector within the ring buffer.

Name	Data Recorder Vector Start Offset	
Index	0x201A	
Subindex	0x00	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

12.5.2.12 Data Recorder Data Buffer

Description

Memory for the different data recorder's ring buffers. Memory allocation is dynamically calculated when the recorder is started.

Name	Data Recorder Data Buffer	
Index	0x201B	
Subindex	0x00	
Type	Domain	
Access	RO	
Default Value	–	
Value Range	–	–

Data Buffer Segmentation (Example: 2 x 16-bit variables, 1 x 32-bit variable)

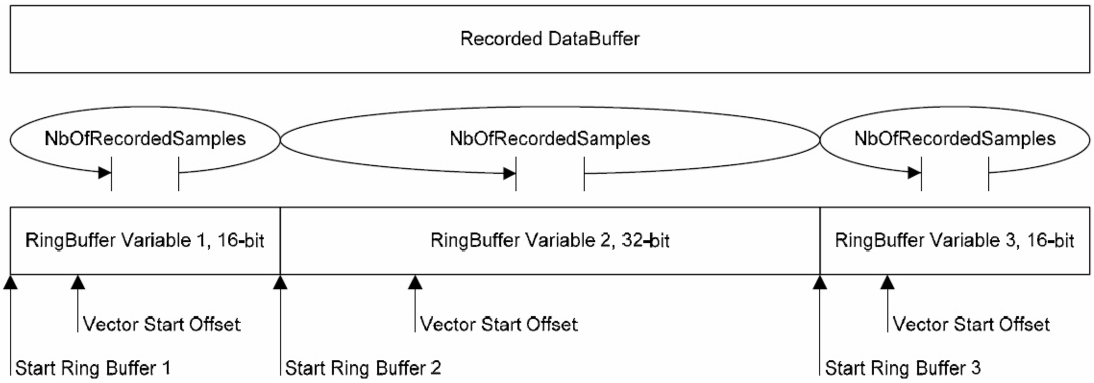


Figure 12-132 Data Recorder Data Buffer – Segmentation

StartRingBuffer1 = 0

StartRingBuffer2 = MaxNbOfSamples * nbOfWords(Variable1)

StartRingBuffer3 = MaxNbOfSamples * (nbOfWords(Variable1) + nbOfWords(Variable2))

••page intentionally left blank••

13 Extended Encoders Configuration

13.1 In Brief

In addition to standard incremental digital encoders to detect the actual position, a number of other sensor types may be used:

- SSI absolute encoder (single or multi turn, 6 to 32 bit resolution, Gray or binary code, RS422)
- Analog incremental encoder (2-channel, max. 10 bit interpolation, Sinus-Cosinus 1 Vss)
- Digital incremental encoder (2-channel or 3-channel, up to 2 500 000 impulse, RS422)

13.1.1 Objective

The present Application Note explains the configuration of extended encoders and features “in practice examples” suitable for daily use.

Contents

13.1.2 Scope

Hardware	Order #	Firmware Version	Reference
EPOS2		2120h	Firmware Specification
EPOS2 70/10	375711	2120h or higher	Cable Starting Set Hardware Reference
EPOS2 50/5	347717	2120h or higher	Cable Starting Set Hardware Reference
EPOS2 Module 36/2	360665	2120h or higher	Hardware Reference

Table 13-177 Extended Encoders Configuration – covered Hardware and required Documents

13.1.3 Tools

Tools	Description
Crimper	Molex hand crimper (63819-0000)
Software	«EPOS Studio» Version 2.00 or higher

Table 13-178 Extended Encoders Configuration – recommended Tools

13.2 Hardware Signals

The extended position sensors can be connected to the EPOS2 Positioning Controllers's digital inputs and outputs.

13.2.1 EPOS2 70/10

Signal 2 Connector (J5A)

Contains differential “High Speed Command” digital inputs.

Additionally available are differential analog inputs.

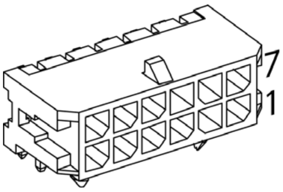


Figure 13-133 EPOS2 70/10 – Signal 2 Connector (J5A)

Pin	Signal	Description
1	+5VOUT	Reference output voltage +5 V
2	A_Gnd	Analog signal ground
3	AnIN2−	Analog Input 2, negative signal
4	AnIN2+	Analog Input 2, positive signal
5	AnIN1−	Analog Input 1, negative signal
6	AnIN1+	Analog Input 1, positive signal
7	D_Gnd	Digital signal ground
8	D_Gnd	Digital signal ground
9	DigIN8/	Digital Input 8 “High Speed Command” complement
10	DigIN8	Digital Input 8 “High Speed Command”
11	DigIN7/	Digital Input 7 “High Speed Command” complement
12	DigIN7	Digital Input 7 “High Speed Command”

Table 13-179 EPOS2 70/10 – Signal 2 Connector (J5A)

Signal 3 Connector (J5B)

Contains differential “High Speed Command” digital I/Os.

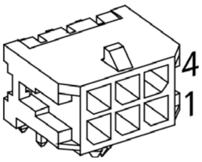


Figure 13-134 EPOS2 70/10 – Signal 3 Connector (J5B)

Pin	Signal	Description
1	DigIN9/	Digital Input 9 “High Speed Command” complement
2	DigIN9	Digital Input 9 “High Speed Command”
3	DigOUT5/	Digital Output 5 “High Speed Output” complement
4	$+V_{AUX}$	Auxiliary output voltage +5DC / 150 mA
5	D_Gnd	Digital signal ground
6	DigOUT5	Digital Output 5 “High Speed Output”

Table 13-180 EPOS2 70/10 – Signal 3 Connector (J5B)

13.2.2 EPOS2 50/5

Signal 1 Connector (J5)

Contains differential “High Speed” digital inputs and outputs.

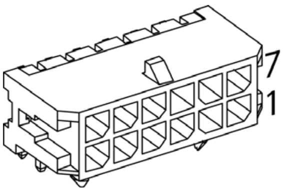


Figure 13-135 EPOS2 50/5 – Signal 1 Connector (J5)

Pin	Signal	Description
1	DigIN10/	Digital Input 10 “High Speed Command” complement
2	DigIN10	Digital Input 10 “High Speed Command”
3	DigIN9/	Digital Input 9 “High Speed Command” complement
4	DigIN9	Digital Input 9 “High Speed Command”
5	DigIN7/	Digital Input 7 “High Speed Command” complement
6	DigIN7	Digital Input 7 “High Speed Command”
7	DigIN8/	Digital Input 8 “High Speed Command” complement
8	DigIN8	Digital Input 8 “High Speed Command”
9	$+V_{AUX}$	Auxiliary output voltage (+5 VDC / 150 mA)
10	D_Gnd	Digital signal ground
11	DigOUT5/	Digital Output 5 “High Speed Output” complement
12	DigOUT5	Digital Output 5 “High Speed Output”

Table 13-181 EPOS2 50/5 – Signal 1 Connector (J5)

13.2.3 EPOS2 Module 36/2

Arrays A and B (excerpt)

Pin	Signal	Description
A10	$+V_{aux}$	Auxiliary voltage output +5 VDC
	$+V_{DDin}$	Auxiliary supply voltage input +5 VDC (optional)
B12	GND	Ground of digital input
B13	DigIN1	Digital Input 1
B14	DigIN2	Digital Input 2
B15	DigIN3	Digital Input 3
B16	DigIN4	Digital Input 4
B17	GND	Ground of digital input
B18	DigIN7	Digital Input 7 “High Speed Command”
B19	DigIN7\	Digital Input 7 “High Speed Command” complement
B20	DigIN8	Digital Input 8 “High Speed Command”
B21	DigIN8\	Digital Input 8 “High Speed Command” complement

Table 13-182 EPOS2 Module 36/2 – Pin Assignment Arrays A & B

13.3 Sensor Types

13.3.1 SSI Absolute Encoder

13.3.1.1 General Description

The Synchronous Serial Interface (SSI) is an interface to connect an absolute position sensor to a controller, such as EPOS2 70/10 or EPOS2 50/5. This interface uses a clock signal from the controller to the sensor and a data signal from the sensor back to the controller. The serial data stream from the sensor begins with the most significant bit.

The number of data bits (for multi turn and single turn resolution) and the clock rate can be configured.

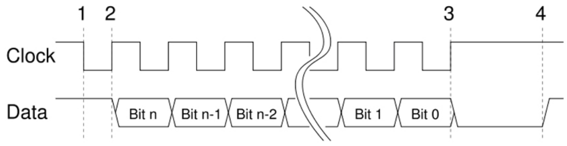


Figure 13-136 SSI Principle

13.3.1.2 EPOS2 Implementation

The EPOS2's SSI interface uses DigOUT5 and DigOUT5/ as differential clock output and DigIN 9 and DigIN 9/ as differential data input.

If the supply voltage of the SSI sensor is 5 V and the current is less than 150 mA, it can be directly supplied from the $+V_{AUX}$ signal (J5-9, respectively J5B-4). Otherwise, an external power supply must be connected to power the sensor.

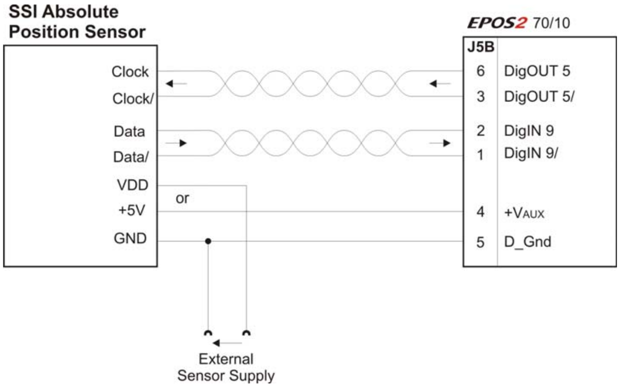


Figure 13-137 EPOS2 70/10 – SSI Encoder Connection

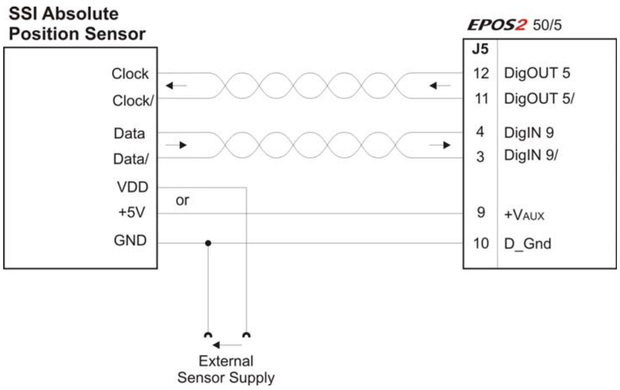


Figure 13-138 EPOS2 50/5 – SSI Encoder Connection

Differential	
DigIN9 “High Speed Command”	Connector [J5B] Pins [1] / [2]
Min. differential input voltage	±200 mV
Line receiver (internal)	EIA RS422 Standard
Max. input frequency	5 MHz

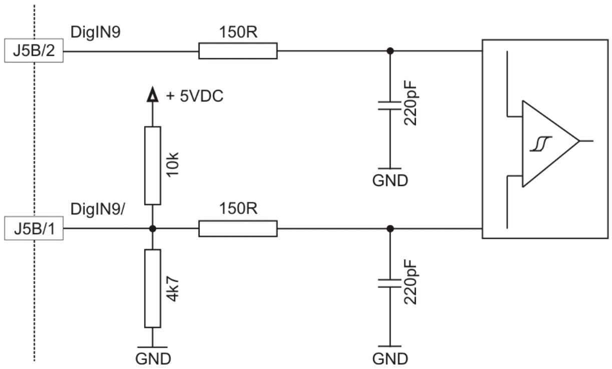


Figure 13-139 EPOS2 70/10 & EPOS2 50/5 – DigIN9 “Differential” Circuit

Differential	
DigOUT5 “High Speed Output”	Connector [J5B] Pins [3] / [6]
Differential output voltage	min 1.5 V @ R_L = 54 Ω
Output current	max. 60 mA
Line transceiver (internal)	EIA RS422 Standard
Max. output frequency	5 MHz

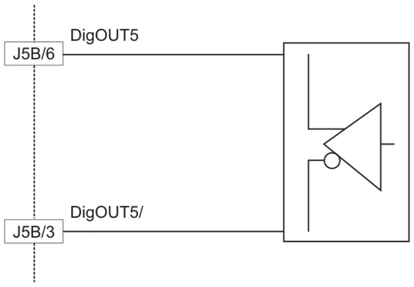


Figure 13-140 EPOS2 70/10 & EPOS2 50/5 – DigOUT5 “Differential” Circuit

13.3.1.3 Choice of Manufacturers for SSI Absolute Encoders

Manufacturer	Contact
Baumer	Baumer Electric AG, CH-8501 Frauenfeld www.baumerelectric.com
Heidenhain	DR. JOHANNES HEIDENHAIN GmbH, DE-83292 Traunreut www.heidenhain.de
Hengstler	HENGSTLER GmbH, DE-78554 Aldingen www.hengstler.com
Posital Fraba	POSITAL GmbH, DE-51063 Cologne www.posital.de
and others	

Table 13-183 SSI Absolute Encoder – Manufacturers (not concluding)

13.3.2 Incremental Encoder 2

13.3.2.1 General description

The incremental signals are transmitted as square-wave pulse trains A and B, phase-shifted by 90° electrical. The signals A and B and their inverted signals typically have TTL levels.

13.3.2.2 EPOS2 Implementation

A second incremental encoder can be connected to the EPOS2's digital inputs DigIN7 to DigIN9, the same inputs which are used for «Master Encoder Mode» and «Step/Direction Mode». Therefore, this two modes cannot be used in conjunction with the Incremental Encoder 2.

If the supply voltage of the incremental encoder is 5 V and the current is less than 150 mA, it can be directly supplied from the $+V_{AUX}$ signal (J5-9, respectively J5B-4). Otherwise, an external power supply must be connected to power the sensor.

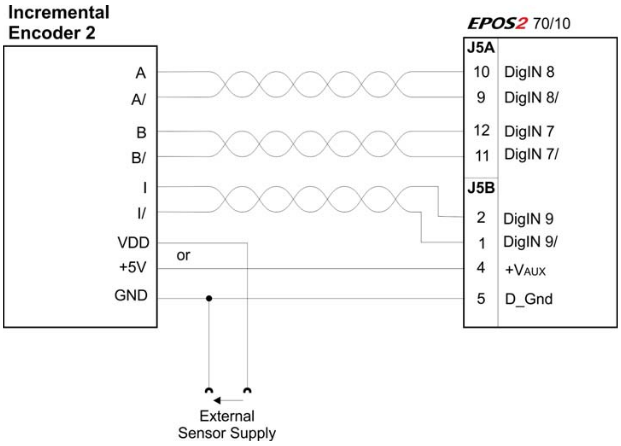


Figure 13-141 EPOS2 70/10 – Incremental Encoder 2 Connection

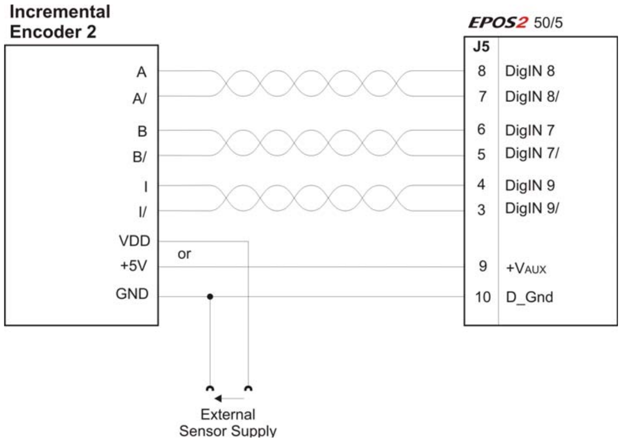


Figure 13-142 EPOS2 50/5 – Incremental Encoder 2 Connection

Differential	
DigIN7 “High Speed Command” DigIN8 “High Speed Command”	Connector [J5A] Pins [9] / [10] Connector [J5A] Pins [11] / [12]
Min. differential input voltage	±200 mV
Line receiver (internal)	EIA RS422 Standard
Max. input frequency	5 MHz

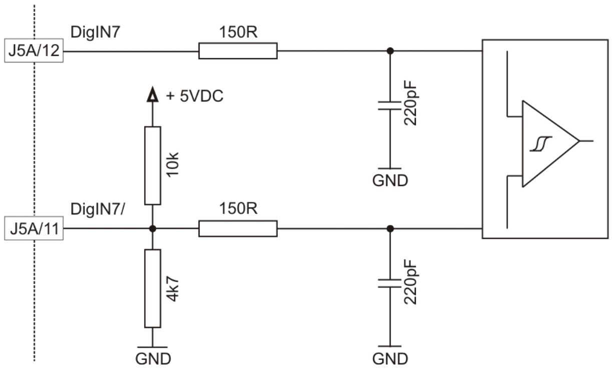


Figure 13-143 EPOS2 70/10 & EPOS2 50/5 – DigIN7 “Differential” Circuit (analogously valid also for DigIN8)

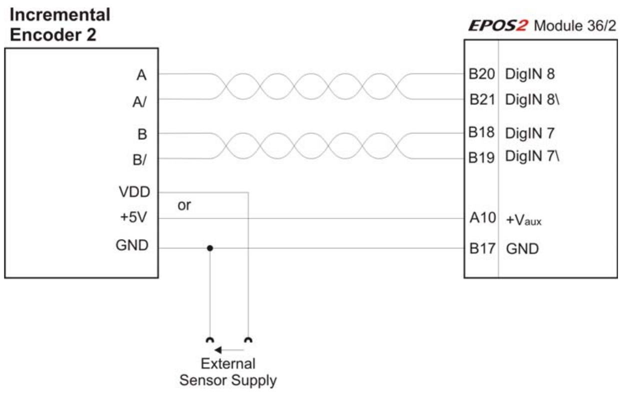


Figure 13-144 EPOS2 Module 36/2 – Incremental Encoder 2 Connection

Differential	
DigIN7 “High Speed Command” DigIN8 “High Speed Command”	Pins [B18] / [B19] Pins [B20] / [B21]
Min. differential input voltage	±200 mV
Line receiver (internal)	EIA RS422 Standard
Max. input frequency	5 MHz

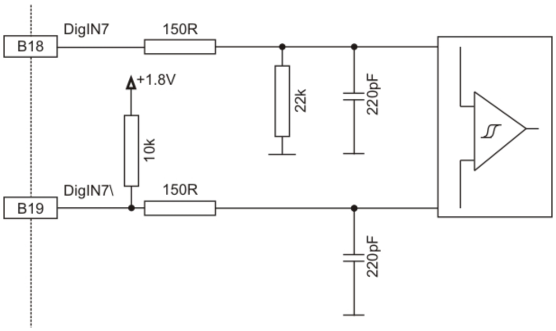


Figure 13-145 EPOS2 Module 36/2 – DigIN7 “Differential” Circuit (analogously valid also for DigIN8)

13.3.2.3 Choice of Manufacturers for Incremental Encoder 2

Manufacturer	Contact
maxon	maxon motor ag, CH-6072 Sachseln www.maxonmotor.com
Baumer	Baumer Electric AG, CH-8501 Frauenfeld www.baumerelectric.com
Heidenhain	DR. JOHANNES HEIDENHAIN GmbH, DE-83292 Traunreut www.heidenhain.de
Hengstler	HENGSTLER GmbH, DE-78554 Aldingen www.hengstler.com
Scancon	SCANCON A/S, DK-3450 Alleroed www.scancon.dk
and others	

Table 13-184 Incremental Encoder 2 – Manufacturers (not concluding)

13.3.3 Sinus Incremental Encoder 2

13.3.3.1 General Description

The sinusoidal incremental signals A and B are phase-shifted by 90° electrical. The differential signal has an amplitude of typically 1 Vpp. The number of periods per turn can be configured.

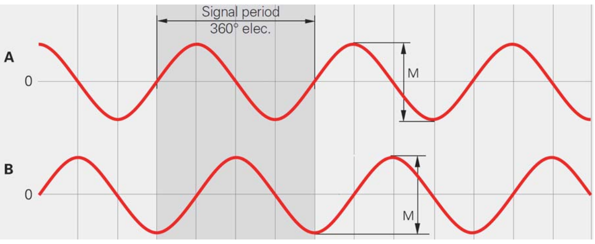


Figure 13-146 Sinus Incremental Encoder Principle

13.3.3.2 EPOS2 Implementation

A sinus incremental encoder can be connected to the EPOS2's digital inputs DigIN7 and DigIN8, the same inputs which are used for «Master Encoder Mode» and «Step/Direction Mode». Therefore, this two modes cannot be used in conjunction with the Sinus Incremental Encoder 2.

If the supply voltage of the SSI sensor is 5 V and the current is less than 150 mA, it can be directly supplied from the $+V_{AUX}$ signal (J5-9, respectively J5B-4). Otherwise, an external power supply must be connected to power the sensor.

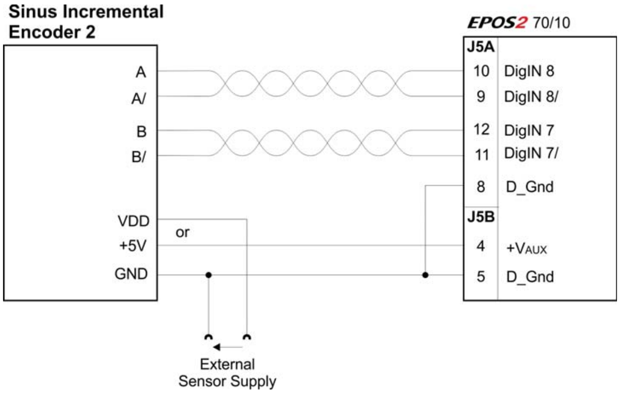


Figure 13-147 EPOS2 70/10 – Sinus Incremental Encoder Connection

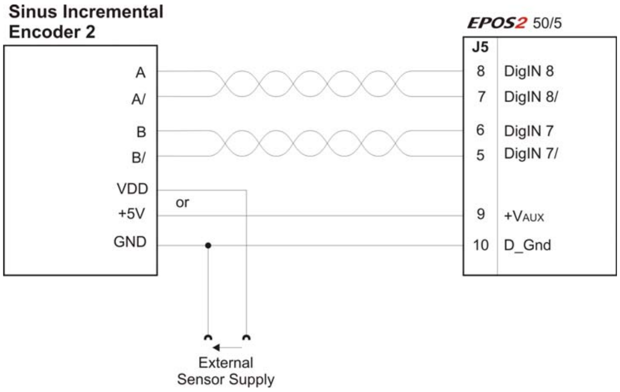


Figure 13-148 EPOS2 50/5 – Sinus Incremental Encoder Connection

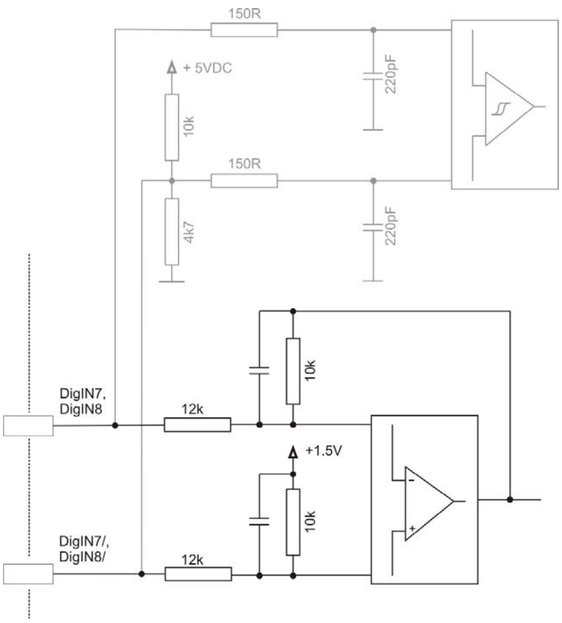


Figure 13-149 EPOS2 70/10 & EPOS2 50/5 – DigIN7/DigIN8 “Differential” Input Circuit of Sinus Incremental Encoder 2

13.3.3.3 Choice of Manufacturers for Sinus Incremental Encoder 2

Manufacturer	Contact
Baumer	Baumer Electric AG, CH-8501 Frauenfeld www.baumerelectric.com
Heidenhain	DR. JOHANNES HEIDENHAIN GmbH, DE-83292 Traunreut www.heidenhain.de
Hengstler	HENGSTLER GmbH, DE-78554 Aldingen www.hengstler.com
and others	

Table 13-185 Sinus Incremental Encoder 2 – Manufacturers (not concluding)

13.4 Configuration Objects

Note
The subsequent information is an extract of the ➔separately available document «EPOS2 Firmware Specification» showing the configuration objects for the extended encoders.

- *Some combinations of sensors can only be configured if the controller structure is set to 1 (velocity auxiliary controller).*
- *With a single loop structure, the main sensor will be used regardless if it is mounted to the motor or to the load.*

13.4.1 Controller Structure

Description

Used to define the dual loop controller structure. Without auxiliary controller, the structure is single loop.

Remarks

If a controller structure will be set to a value that is in conflict with the actual position sensor type, the sensor type will be set to “0” (Unknown sensor).

Can only be changed in “Disable” state.

Name	Controller Structure
Index	0x2220
Subindex	0x00
Type	UNSIGNED16
Access	RW
Default Value	–
Value Range	➔Table 13-186

Value	Description
0	no auxiliary controller
1	velocity auxiliary controller (available with EPOS2 70/10, EPOS2 50/5 and EPOS2 Module 36/2 only)

Table 13-186 Controller Structure

13.4.2 Sensor Configuration

Name	Sensor Configuration
Index	0x2210
Number of entries	4

Description

Used to define the main and the auxiliary controller's sensor type.

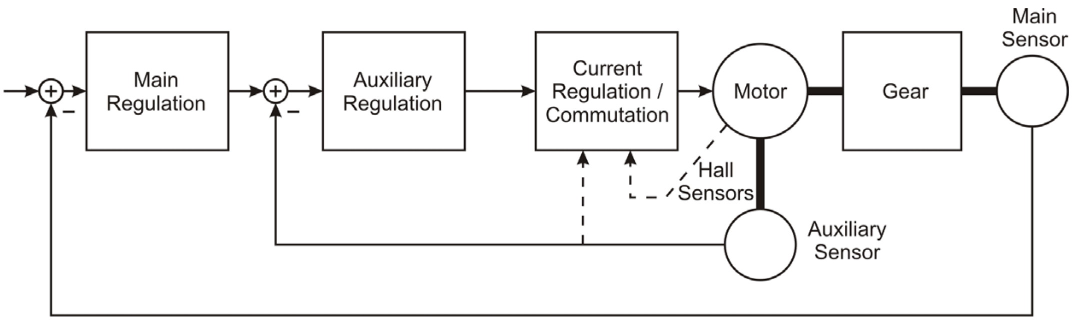


Figure 13-150 Regulation, Sensor and Gear Overview

Name	Position Sensor Type
Index	0x2210
Subindex	0x02
Type	UNSIGNED16
Access	RW
Default Value	0x01
Value Range	➔Table 13-187 and Table 13-188

Bit	Description
15…12	reserved (0)
11…8	Sensor type of auxiliary controller
7…4	reserved (0)
3…0	Sensor type of main controller

Table 13-187 Position Sensor Type – Bits

Value	Description	Abbreviation
0	Unknown sensor (undefined)	–
1	Incremental Encoder 1 with index (3-channel)	Inc Enc1
2	Incremental Encoder 1 without index (2-channel)	
3	Hall Sensors (Remark: consider worse resolution)	Hall
4	Absolute encoder SSI *1)	SSI
5	reserved	–

Value	Description	Abbreviation
6	Incremental Encoder 2 with index (3-channel) *1)	Inc Enc2
7	Incremental Encoder 2 without index (2-channel) *2)	
8	Sinus Incremental Encoder 2 *1)	Sin Inc Enc2
Remarks: *1) only available with EPOS2 70/10 and EPOS2 50/5 *2) only available with EPOS2 70/10, EPOS2 50/5 and EPOS2 Module		

Table 13-188 Supported Sensor Types

Description

Used to change the position sensor polarity.

Remarks

Can only be changed in “Disable” state.

The absolute position may be corrupted after changing this parameter.

Name	Position Sensor Polarity
Index	0x2210
Subindex	0x04
Type	UNSIGNED16
Access	RW
Default Value	0x00
Value Range	➔Table 13-189

Bit	Value	Name	Description
0	0	Incremental Encoder 1	normal Enc1 polarity (CCW counts positive)
	1		inverted Enc1 polarity (or encoder mounted on motor shaft)
1	0	Hall sensors	normal Hall sensor polarity (maxon standard)
	1		inverted Hall sensor polarity
2	0	SSI Encoder	normal SSI polarity (CCW counts positive)
	1		inverted SSI polarity
3	0	Incremental Encoder 2 *2)	normal Enc2 polarity (CCW counts positive)
	1		inverted Enc2 polarity (or encoder mounted on motor shaft)
4	0	Sinus Incremental Encoder *1)	normal Enc2Sin Encoder polarity (CCW counts positive)
	1		inverted Enc2Sin Encoder polarity
5…15	(0)	reserved	–
Remarks: *1) only available with EPOS2 70/10 and EPOS2 50/5 *2) only available with EPOS2 70/10, EPOS2 50/5 and EPOS2 Module 36/2			

Table 13-189 Position Sensor Polarity

13.4.3 SSI Encoder Configuration

Description

Used to configure the interpretation of the SSI Encoder.

Remark

Changes are only supported in “Disable” state.

Name	SSI Encoder Configuration
Index	0x2211
Number of entries	4

Description

SSI data rate (SSI clock frequency) in [kbit/s].

Remark

The maximal data rate depends on the actual line length and the employed SSI encoders’ specifications. Typically are 400 kbit/s for cable lengths <50 m.

Name	SSI Encoder Datarate	
Index	0x2211	
Subindex	0x01	
Type	UNSIGNED16	
Access	RW	
Default Value	500	
Value Range	400	2 000

Description

Defines the number of multi-turn and single-turn bits. The maximal number of bits for both values combined is 32. The resolution is $2^{\text{number of bits single-turn}}$.

Name	SSI Encoder Number of Data Bits
Index	0x2211
Subindex	0x02
Type	UNSIGNED16
Access	RW
Default Value	3085 (0x0C0D)
Value Range	➔Table 13-190

Bit	Name	Value		
		Minimal	Maximal	Default
15…8	number of bits multi-turn	0	26	12
7…0	number of bits single-turn	6	23	13

Table 13-190 SSI Encoder Number of Data Bits

Description

Position received from encoder [Position units] (➔page 1-13).

Name	SSI Encoder Actual Position	
Index	0x2211	
Subindex	0x03	
Type	INTEGER32	
Access	RO	
Default Value	–	
Value Range	–	–

Description

Defines the SSI's encoding type.

Name	SSI Encoding Type
Index	0x2211
Subindex	0x04
Type	UNSIGNED16
Access	RW
Default Value	0
Value Range	➔Table 13-191

Value	Description
0	SSI Encoder binary type
1	SSI Encoder Gray coded

Table 13-191 SSI Encoding Type

13.4.4 Incremental Encoder 2 Configuration

Description

Used to configure the interpretation of the Incremental Encoder 2.

Remarks

Can only be changed in “Disable” state.

The absolute position may be corrupted after changing this parameter.

Name	Incremental Encoder 2 Configuration
Index	0x2212
Number of entries	3

Description

The encoder’s pulse number must be set to number of pulses per turn of the connected Incremental Encoder.

Name	Incremental Encoder 2 Pulse Number	
Index	0x2212	
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	500	
Value Range	16	2 500 000

Description

Holds the internal counter register of the Incremental Encoder 2. It shows the actual encoder position in quad counts [qc].

Name	Incremental Encoder 2 Counter	
Index	0x2212	
Subindex	0x02	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	–	–

Description

Holds the Incremental Encoder 2 counter reached upon last detected encoder index pulse. It shows the actual encoder index position in quad counts [qc].

Name	Incremental Encoder 2 Counter at Index Pulse	
Index	0x2212	
Subindex	0x03	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	–	–

13.4.5 Sinus Incremental Encoder 2 Configuration

Description

Used to configure the Sinus Incremental Encoder 2 Configuration’s interpretation.

Remarks

Can only be changed in “Disable” state.

The absolute position may be corrupted after changing this parameter.

Name	Sinus Incremental Encoder 2 Configuration
Index	0x2213
Number of entries	2

Description

Defines the resolution of “Sinus Incremental Encoder 2”. The parameter pulses per turn must be set to the number of pulses per revolution of the connected Sinus Incremental Encoder.

This value multiplied by $2^{\text{number of interpolation bits}}$ is the total resolution of the Sinus Incremental Encoder.

The values are further limited as follows:

Max. resolution: $2^{\text{number of interpolation bits}}$ * pulses per turn ≤ 10 000 000

Min. resolution: $2^{\text{number of interpolation bits}}$ * pulses per turn ≥ 64

Name	Sinus Incremental Encoder 2 Resolution
Index	0x2213
Subindex	0x01
Type	UNSIGNED32
Access	RW
Default Value	0x00800006
Value Range	➔Table 13-192

Bit	Name	Value		
		Minimal	Maximal	Default
15…8	pulses per turn	1	2 500 000	2048
7…0	number of interpolation bits	2	10	4

Table 13-192 Encoder 2 Resolution

Description

Position received from Sinus Incremental Encoder [Position units] (➔page 1-13).

Name	Sinus Incremental Encoder 2 Actual Position	
Index	0x2213	
Subindex	0x02	
Type	INTEGER32	
Access	RO	
Default Value	–	
Value Range	–	–

13.5 Application Examples

Best Practice
The system should work correct if you employ components as listed and configure them as described.
If not the case, check the objects' configuration after executing the described wizards and adjust/tune them according to the actual components employed.

13.5.1 Example 1: Single Loop DC Motor / Gear / SSI Absolute Encoder

Equipment	Type / Specifications
Controller	maxon motor controller EPOS2 70/10 (375711)
Motor	maxon DC motor (any)
Gear	maxon gear (any) reduction 23:1 (absolute 576:25), recommended input speed <6000 rpm
Absolute SSI Encoder	Baumer BMMH (42S105C 12/13 B25) Coding: Gray Interface Data Rate: 500 kbit/s Singleturn Data Bits: 12 Multiturn Data Bits: 13

Table 13-193 Example 1 – Setup

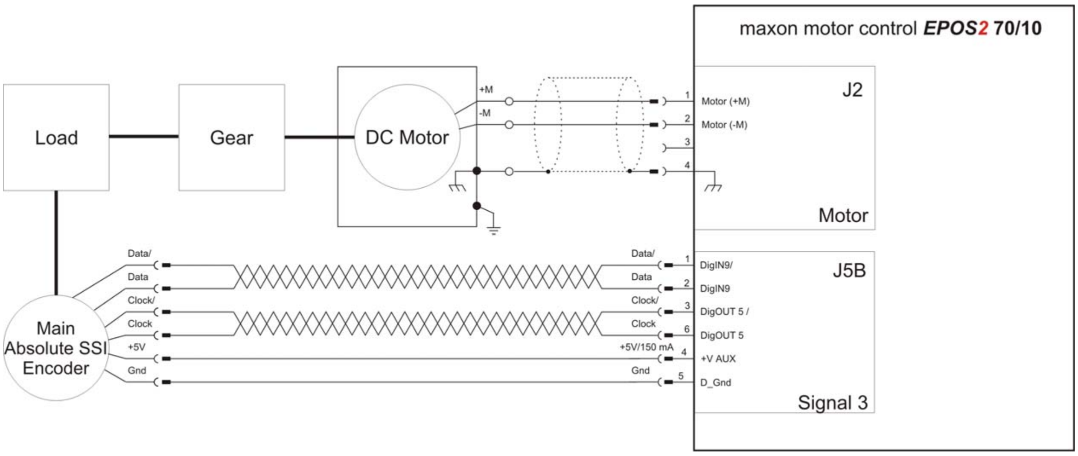


Figure 13-151 Example 1 – Wiring Diagram

1) Wire the system according to the wiring diagram (➔Figure 13-151).
2) Follow the configuration steps in the "Startup Wizard" of «EPOS Studio».
3) Upon successful configuration, start the "Regulation Tuning Wizard".

4) Now your system is ready to use.
   For verification purposes: The related objects should have been set as follows:

Index	SubIndex	Name	Type	Access	Value
0x2210		Sensor Configuration			
0x2210	0x02	Position Sensor Type	UInt16	RW	260
0x2210	0x04	Position Sensor Polarity	UInt16	RW	0
0x2211		SSI Encoder Configuration			
0x2211	0x01	SSI Encoder Data Rate	UInt16	RW	500
0x2211	0x02	SSI Encoder Number of Data Bits	UInt16	RW	3340
0x2211	0x04	SSI Encoder Encoding Type	UInt16	RW	1
0x2230		Gear Configuration			
0x2230	0x01	Gear Ratio Numerator	UInt32	RW	576
0x2230	0x02	Gear Ratio Denominator	UInt16	RW	25
0x2230	0x03	Gear Maximal Speed	UInt32	RW	600
0x6402	0x00	Motor Type	UInt16	RW	10

Figure 13-152 Example 1 – Object Configuration

13.5.2 Example 2: Dual Loop Incremental Encoder (2 Ch) / EC Motor / Gear / Incremental Encoder (3 Ch)

Equipment	Type / Specifications
Controller	maxon motor controller EPOS2 50/5 (347717)
Motor	maxon EC motor (any)
Auxiliary Encoder	maxon Encoder MR Counts per turn: 1000 inc. Number of Channels: 2 (or 3)
Gear	maxon gear (any) reduction 5.8:1 (absolute 23:4), recommended input speed <8000 rpm
Main Encoder	Baumer BHF (16.05A 7200-E2-5) Counts per turn: 7200 inc. Number of Channels: 3

Table 13-194 Example 2 – Setup

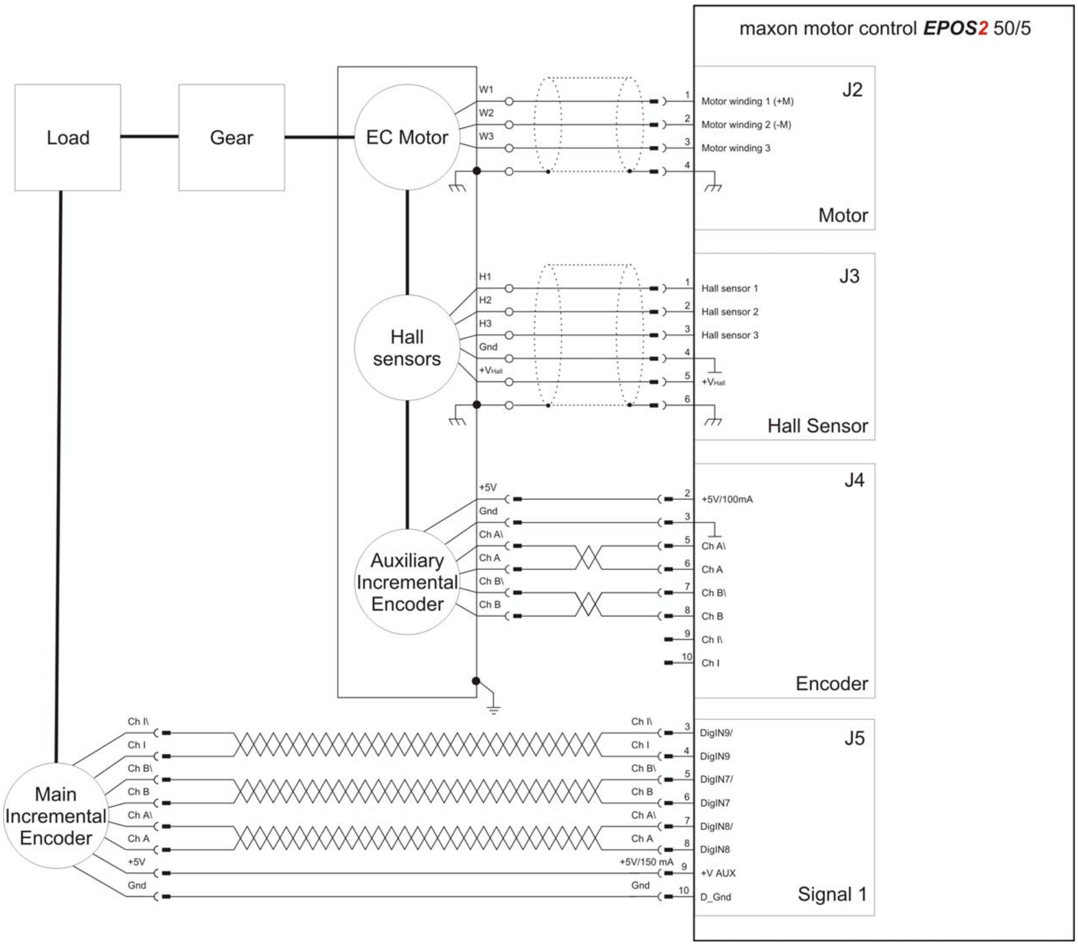


Figure 13-153 Example 2 – Wiring Diagram

1) Wire the system according to the wiring diagram (➔Figure 13-153).
2) Follow the configuration steps in the “Startup Wizard” of «EPOS Studio».
3) Upon successful configuration, start the “Regulation Tuning Wizard”.
4) Now your system is ready to use.
   For verification purposes: The related objects should have been set as follows:

Index	SubIndex	Name	Type	Access	Value
0x2210		Sensor Configuration			
0x2210	0x01	Pulse Number Incremental Encoder 1	UInt32	RW	500
0x2210	0x02	Position Sensor Type	UInt16	RW	260
0x2210	0x04	Position Sensor Polarity	UInt16	RW	0
0x2212		Incremental Encoder 2 Configuration			
0x2212	0x01	Incremental Encoder 2 Pulse Number	UInt32	RW	500
0x2220	0x00	Controller Structure	UInt16	RW	1
0x2230		Gear Configuration			
0x2230	0x01	Gear Ratio Numerator	UInt32	RW	576
0x2230	0x02	Gear Ratio Denominator	UInt16	RW	25
0x2230	0x03	Gear Maximal Speed	UInt32	RW	600
0x6402	0x00	Motor Type	UInt16	RW	10

Figure 13-154 Example 2 – Object Configuration

LIST OF FIGURES

LIST OF TABLES

••page intentionally left blank••

INDEX

F

H

I

L

M

N

O

maxon motor ag
Brünigstrasse 220
P.O.Box 263
CH-6072 Sachseln
Switzerland

Phone +41 41 666 15 00
Fax +41 41 666 16 50

www.maxonmotor.com