MSD Servo Drive

User Manual

Field bus systems
CANopen and EtherCAT

Single-Axis System
Multi-Axis System
Compact

CANopen

EtherCAT

In this documentation the functionality of the following devices is described:

MSD Servo Drive Single-Axis system
MSD Servo Drive Multi-Axis system
MSD Servo Drive Compact

MSD Servo Drive User Manual
Field bus systems CANopen and EtherCAT

ID no.: CA65647-001, Rev 2.0

Date: 03/2020

The German version is the original of this user manual.

Table of contents

1 General information

The product CD from Moog contains the complete documentation for the related product series. The documentation for a product series includes the operating manual (hardware description), device help (software description) as well as further user manuals (e.g. field bus description) and specifications. They are available in the formats PDF, HTML or chm.

1.1 Target group

Dear user,

The user manual forms part of the product and contains important information on operation and service. The user manual is aimed at all persons who undertake mounting, installation, commissioning and servicing work on the product.

1.2 Prerequisites

Prerequisites for the usage of the CANopen and EtherCAT field bus systems:

- The documents related to the devices are legible and accessible.
- Read and understand the Operation Manual for your drive system first.
- You are familiar with the CANopen and EtherCAT field bus systems through training courses.

Following the documentation on the devices from Moog GmbH is a prerequisite for trouble-free operation and therefore for any claims for defects.

NOTE:
This User Manual applies to the MSD Servo Drive Single-Axis System and Multi-Axis System (referred to in the following as Servo Drive or AC-AC or DC-AC for short) and the MSD Servo Drive Compact. This document does not replace the Operation Manuals for the MSD Servo Drive and MSD Servo Drive Compact.

1.3 Reference documents

Document	Contents	ID no. Format
MSD Single-Axis Servo Drive Compact- Operation Manual	Safety, mechanical installation, electrical installation, commissioning, diagnostics, specifications, certification and applicable standards, technical data	CA97555-001 PDF
MSD Servo Drive AC-AC Servo Drive Single-Axis System - Operation Manual	Safety, mechanical installation, electrical installation, commissioning, diagnostics, specifications, certification and applicable standards, technical data	CA65642-001 PDF
MSD Servo Drive DC-AC Servo Drive Multi-Axis System- Operation Manual	Safety, mechanical installation, electrical installation, commissioning, diagnostics, STO, operation with AC-AC Servo Drive as supply, planning, application example, specifications, certification and applicable standards, technical data	CA97554-001 PDF
MSD Power Supply Unit Multi-Axis System- Operation Manual	Safety, mechanical installation, electrical installation, commissioning, diagnostics, specification, certification and applicable standards, technical data	CA97556-001 PDF
MSD Servo Drive Sercos II - User Manual	Safety, commissioning, communication phases, parameter interface, error, warning and status messages, operation modes, weighting, referencing, touchprobe, parameter lists	CA65648-001
MSD Servo Drive Sercos III - User Manual	Safety, installation and connection, commissioning and configuration, setting parameters, data transmission, scaling and weighting, functionality, error message and diagnostics, parameter lists	CA97557-001 PDF
MSD Servo Drive Field bus systems Profibus/Profinet - User Manual	Description and configuration of the parameters for the MSD Servo Drive on the PROFIBUS/PROFINET field bus system	CA65645-001
Modular Multi-Axis Servo Drive System - MSD - Ordering Catalog	Information, notes on ordering, specifications and technical data on: MSD Single-Axis Servo Drive Compact, MSD Single-Axis System, MSD Multi-Axis System, safety technology, communication, technology, function packages, accessories and motors	CDL 29950-en
MSD Servo Drive - Device Help	Description of the software functionality MSD Servo Drive, firmware versions: - MSD Single-Axis Servo Drive Compact from V1.30-xx - MSD Single-Axis System from V124-xx - MSD Multi-Axis System from V124-xx	CB40859-001 PDF and HTML
Moog DriveAdminsitrator 5 PC user software - Program help	Context-sensitive help for Moog DriveAdministrator version 5.x graphic PC user software for initial commissioning and serial commissioning, operation, diagnostics and project management	CB19692-001

Table 1.1 Documents on the MSD Servo Drive system

1.4 Pictograms for useful information

The pictograms used in this document for useful information and actions to be taken signify for the user the following:

Instructions and actions to be taken	
	**NOTE:** Useful information or reference to other documents.
Digit	**ACTION TO BE TAKEN:** Action undertaken by the user or the system.

Table 1.2 Pictograms used for instructions and actions to be taken

1.5 Disclaimer

Following the documents on the devices from Moog is a prerequisite:
- For safe operation.
- To achieve stated performance features and product characteristics.

Moog does not accept any liability for injuries, damage or financial losses that result from the failure to follow the documents.

1.6 Transport, storage

Follow the instructions on the transport, storage and correct usage of the devices stated in the Operation Manual in "Technical data".

1.7 Disposal

Follow the applicable national regulations! If necessary, dispose of individual parts, depending on their characteristics and existing national regulations,
e.g. as
- Electrical waste
- Plastic
- Metal

or engage a certified disposal organisation with scrapping.

1.8 Support

Our Helpline will help you with fast, specific assistance if you have any technical queries relating to project planning or commissioning your device.

Address: Moog GmbH
Hanns-Klemm-Straße 28
D-71034 Böblingen

Phone: +49 7031 622-0
Fax: +49 7031 622-100
E-Mail: drives-support@moog.com

If you need service assistance, the Moog specialists will be pleased to be of assistance.

Service - Please contact us

Phone: +49 7031 622-0
E-Mail: info.germany@moog.com

1.9 Normative references

The following standards apply to the CANopen and EtherCAT field bus system:

IEC/EN 61158 -
Industrial communication networks. Fieldbus specifications.

IEC 61508-4 (1998-12) -
Functional safety of electrical/electronic/ programmable electronic safety related systems – Part 4: Definitions and abbreviations.

IEC 61784-1 (2004-7) -
Digital data communications for measurement and control – Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems.

IEC 50325-4 2002 Part 4: CANopen -
European standard that describes the CANopen standard in the context of the CAN user organisation.

2 Safety

2.1 Overview

Our devices are state-of-the-art and comply with recognised safety regulations, nevertheless hazards can arise. In this chapter:

- We provide information on residual risks and hazards that can emanate from our devices on usage as intended.
- We warn about the foreseeable misuse of our devices.
- We refer to the necessary care and measures to be taken to prevent risks.

2.2 Measures for safety

Note:
Only install and place in operation your device taking into account the documents for the related device family!

Our devices are quick and safe to operate. For your own safety and for the safe function of your device, please be sure to observe the following points:

1. **Follow safety instructions for the devices!**
   - Follow safety instructions and warnings in the information products related to the devices!
2. **Electrical drives are dangerous due to:**
   - Electrical voltages > 230 V/460 V! Dangerous voltages may be present 10 min. after the power is switched off. So check that electrical power is not present!
   - Rotating parts.
   - Automatically starting drives.
   - Hot components and surfaces.
   - Electrostatic discharge.
3. **Protection against magnetic and electromagnetic fields during installation and operation**
   Persons fitted with heart pacemakers, metallic implants and hearing aids must not be allowed access to the following areas:
   - Areas in the immediate vicinity of electrical equipment!
   - Areas where electronics components and drive controllers are installed, repaired and operated!
   - Areas where motors are installed, repaired and operated!
     Particular hazards emanate from motors with permanent magnets.
4. **Your qualifications**
   To prevent injury or damage, personnel may only work on the device if they have
   electrical engineering qualifications and
   knowledge of:
   - National health and safety regulations (DGUV V3 in Germany).
   - Erection, installation, commissioning and operation of the device.

   All work in other areas, such as transport, storage and disposal is only allowed to be undertaken by trained personnel.
   The warranty will be rendered void on failure to follow these instructions!
5. **During installation observe the following:**
   - Comply with connection conditions and technical data as per the information product and the rating plate!
   - Comply with standards and directives on electrical installation, such as cable cross-section, shielding, etc.!
   - Do not touch electronic components and contacts!
     Electrostatic discharge can harm people and destroy components!
   - Take protection measures and use protective devices as per the applicable regulations (e.g. IEC/EN 60204 or IEC/EN 61800-5-1)!
   - Take "device earthing" protection measure!

2.3 General safety instructions and warnings

Hazards may emanate from our devices. For this reason pay attention to:

- The safety instructions and warnings in this document!
- Safety instructions and warnings in other documents, in particular the operation manuals related to the device product ranges!

Generally applicable safety instructions and warnings for the users of devices from Moog:

DANGER!	Risk of injury due to electrical power!
	· **Carelessness will result in serious injuries or death.** Follow safety instructions and warnings in this document and on the device.

WARNING!	Risk of injury due to electrical power!
	· **Carelessness may result in serious injuries or death.** Follow safety instructions and warnings in this document and on the device.

CAUTION!	Risk of injury or damage to the device due to incorrect operation!
	· **Carelessness may result in minor injuries or damage.** Follow safety instructions and warnings in this document and on the device.

WARNING!	Risk of injury due to hot surfaces and components!
	· **Carelessness may result in serious burns.** Electronic components may become hot during operation! Follow safety instructions and warnings in this document and on the device!

WARNING!	Risk of injury or damage due to electrostatic discharge!
	· **Electrostatic discharge can destroy components and, in the worst case, cause injury or death.** Do not touch electronic components and contacts! Follow safety instructions and warnings in this document and on the device!

Pay attention to **special safety instructions and warnings** that are given here in the document before a specific action and that warn the user about a **specific hazard**!

2.3.1 Safety instructions related to this documentation

The CANopen and EtherCAT field bus systems are communication systems that are to be adapted at the boundaries of the drive systems MSD Servo Drive Single-Axis System and Multi-Axis System and MSD Servo Drive Compact to the situation in the installation.

CAUTION!	The user may not be aware of changes to the parameters in the field bus system!
	· This situation can result in uncontrolled behaviour of the drive system! Prior to system start, check parameters!

2.4 Important information

In the Operation Manuals for the drive product ranges you will find detailed information on the following areas:

- Intended use.
- Important instructions on the installation of your device.
- Responsibilities of installers and organisations operating complete machines or installations.
- Relevant laws, standards and directives applied.

3 Device description

3.1 CANopen field bus system

The CANopen field bus system describes the exchange of data and is based on the CAN networking concept (CAN = Controller Area Network). The CANopen standard defines fundamental communication mechanisms (communication profile) and the functionality of the communicating devices (device profile).

3.1.1 CANopen functionality of the MSD Servo Drive

The CANopen communication profile is defined in CiA301. This profile:

- Describes services and protocols used in CANopen.
- Differentiates between process data objects (PDOs) and service data objects (SDOs).
- Defines simple network management.

The device profile CiA402 (rev. 2.0) for electrical drives was produced based on the communication profile CiA301 (rev. 4.01). This device profile:

- Describes operation modes and objects.
- Defines the functionality and structure of the object dictionary for the devices.

The usage of CANopen devices that comply with the device profile CiA402 gives greater independence from the device manufacturers.

In the following sections you will find an overview of the CANopen functionality integrated into the MSD Servo Drive and information on commissioning.

3.1.2 System requirements CANopen

System requirements Moog devices	
Devices:	MSD Servo Drive
Standardisation	ISO 11898 / IEC/EN 61800-7
EDS file:	Current "electronic data sheet" for CANopen (field bus device description file for MSD Servo Drive as .eds file). Is available with the firmware in the download area at **www.moogsoftwaredownload.com/msd.html**
Communication profile	CiA301 (rev. 4.01)
Device profile	CiA402 (rev. 2.0)
Service tool:	PC user software Moog DriveAdministrator 5
Master:	CANopen master with support for the above-mentioned communication and device profiles

Table 3.1 System requirements CANopen

3.2 EtherCAT field bus system

If the issue is real-time Ethernet systems, these days EtherCAT has become well-established in the automation sector.

The structure of an EtherCAT network can be straightforwardly adapted to a machine infrastructure and can have a

- Star, ring or line structure.
- With patch cable or crossover cable.

The goal of the EtherCAT field bus system is to create one computer network from the control level in the office area to the field devices in industrial production plants (vertical integration of the automation technology).

3.2.1 The EtherCAT protocol

EtherCAT® is a registered trade mark and patented technology licensed by Beckhoff Automation GmbH, Germany.

The protocol published in IEC standard IEC/EN 61158 is suitable for hard and soft real-time requirements in automation technology.

3.2.2 EtherCAT functionality of the MSD Servo Drive

To minimise the familiarisation effort, usage was made of familiar communication and device profiles from the application layer. Users who are familiar with the communication and device profiles CiA301 and CiA402 can change to this new field bus technology with little effort.

3.2.3 System requirements EtherCAT

System requirements Moog devices	
Devices:	MSD Servo Drive
Standardisation	IEC/EN 61158 / IEC/EN 61784-2 / IEC/EN 61800-7
ESI file:	Current "EtherCAT slave information" (ESI field bus device description file for MSD Servo Drive as .xml file). Is available with the firmware in the download area at **www.moogsoftwaredownload.com/msd.html**
Cable:	Patch cable from CAT 5e SFTP
Communication profile	IEC/EN 61158 / ETG. 1000
Device profile	(Rev. 2.0) CiA402 (CoE) / ETG. 2200
Service tool:	PC user software Moog DriveAdministrator 5
Master:	EtherCAT master with support for the above-mentioned communication and device profiles

Table 3.2 System requirements EtherCAT

3.3 Other documents

Moog documentation	ID no. / file formats	Web site
MSD Servo Drive Compact - Operation Manual	CA97555-001 / PDF	drives-support@moog.com
MSD Servo Drive Single-Axis System - Operation Manual	CA65642-001 / PDF	
MSD Servo Drive Multi-Axis System - Operation Manual	CA97554-001 / PDF	
MSD Servo Drive Multi-Axis System Power Supply Unit - Operation Manual	CA97556-001 / PDF	
MSD Servo Drive - Device Help	CB40859-001 / PDF and HTML	

Table 3.3 Moog documentation

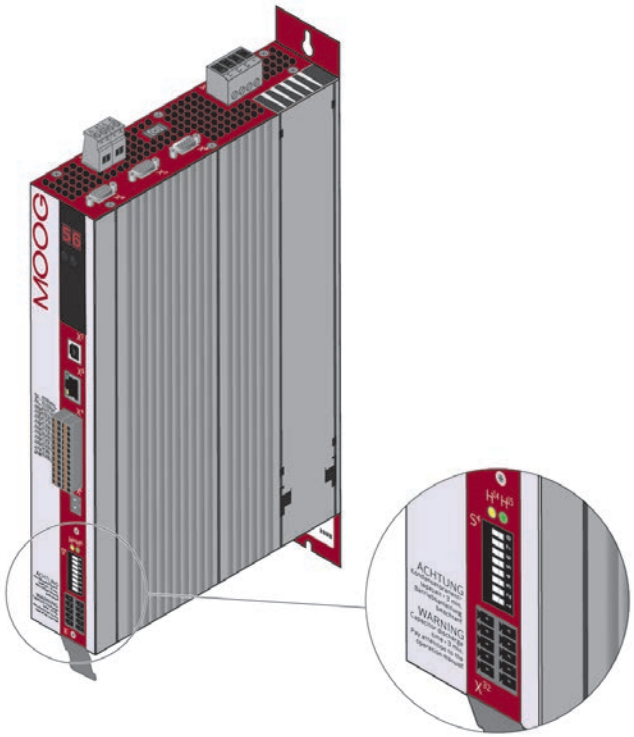
Other field bus documentation	Web site
CiA® CiA402 (V4.2.0): CANopen application layer and communication profile	www.can-cia.com
CiA® CiA402: CANopen device profile for drives and motion control	
EtherCAT Communication Specification Version 1.0 2004	www.ethercat.org
EtherCAT Indicator Specification Proposal V0.91 2005	www.ethercat.org
Series IEC/EN 61158 standards (-1 to -6-x): Industrial communication networks – Fieldbus specifications	See: www.beuth.de
Other information on the CANopen field bus system	www.can-cia.com
Other information on the EtherCAT field bus system	www.ethercat.org

Table 3.4 Other field bus documentation

4 Installation and connection of the CANopen field bus system

4.1 Position

The illustration (below) shows the position of the CANopen field bus system in the MSD Servo Drive. The position in the MSD Servo Drive Compact is identical.



NOTE:
The CANopen field bus system is designed as a variant for the product ranges MSD Servo Drive Compact, Single-Axis and Multi-Axis Servo Drives!

CAUTION!	Risk of injury and/or damage due to electrical power!
	• Touching bare, electrically live wires or wires from which the insulation has been stripped can cause an electric shock and burns. • Short circuits can cause damage to the device! On mounting electrical components, e.g. wires and cables, make sure electrical power is not present! If necessary, replace damaged cables!

4.2 Connections and controls

The figure below shows the position of the connections, controls and diagnostic LEDs for the CANopen field bus system on the MSD Servo Drive.

MSD Servo Drive Single-Axis and Multi-Axis System (G392/G395/G393/G397)	MSD Servo Drive Compact (G394)

Figure 4.1 Layout, connections and controls, CANopen

Connection	Designation	Function
1	H14	LED (yellow) CANopen network status
2	H15	LED (green) status indication - supply voltage
3	S4	DIP switch for address assignment **(MSD Servo Drive Single-Axis and Multi-Axis System)**
4	S4	DIP switch for address assignment **(MSD Servo Drive Compact)**
5	X32	System connection

Table 4.1 Connections, controls and LEDs for the CANopen field bus system

4.3 Light emitting diodes - flashing codes for the field bus system

LED	CANopen status - network status - NMT	
H14 (yellow LED)	NMT INIT / BOOT-UP	LED flashes with 100 ms cycle
	NMT STOPPED	LED flashes with 800 ms cycle
	NMT PRE-OPERATIONAL	LED flashes with 1600 ms cycle
	NMT OPERATIONAL	LED illuminates continuously

Table 4.2 LED H14 - flashing codes/network status

LED	CANopen status - supply voltage	
H15 (green LED)	+24 V supply voltage is present at the CANopen field bus system	LED illuminates continuously

Table 4.3 LED H15 - flashing codes/supply voltage

4.4 Address assignment CANopen (node ID)

Ask your planner which node ID is to be used for your Servo Drive:

There are 3 ways of assigning the address for the CANopen field bus system, via:

1. **The software CAN address (P 2005-COM_CAN_Adr in the Servo Drive).**
2. **The DIP switch (S4, hardware address).**
3. **A combination of software CAN address for the Servo Drive and the DIP switch (S4, hardware address).**
   Node ID = Hardware address (S4) + software address **P 2005-COM_CAN_Adr**.
   This type of addressing appropriate if
   - The same parameter set is used for up to 15 drives
   - And the lowest address is 30.

4.4.1 Address assignment via the software CAN address

Type the address for your field bus system in the parameter **P 2005-COM_CAN_Adr** in the PC user software Moog DriveAdministrator (section "Field bus" --> CANopen/ EtherCAT).

You will find more detailed information in the help for the PC user software Moog DriveAdministrator 5.

4.4.2 Address assignment via DIP switch (S4)

Set the hardware address of your CANopen field bus system on the DIP switch (S4) on the MSD Servo Drive.

DIP switch MSD Servo Drive (Single-Axis and Multi-Axis Servo Drive)

- Address setting: switch 1 to 7
- Activate/deactivate the 120 Ω bus terminating resistor: switch 8
- Number of possible addresses: 0 to 127

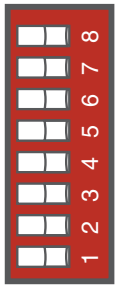
Significance of the DIP switch on the MSD Servo Drive (Single-Axis and Multi-Axis Servo Drive)			
	DIP switch number	**Significance**	**Address**
	1	2^0	1
	2	2^1	2
	3	2^2	4
	4	2^3	8
	5	2^4	16
	6	2^5	32
	7	2^6	64
	8	Bus terminating resistor (120 Ω)	

Table 4.4 DIP switch MSD Servo Drive (Single-Axis and Multi-Axis System)

DIP switch MSD Servo Drive Compact

- Address setting: switch 1 to 6.
- Activate/deactivate the 120 Ω bus terminating resistor: switch 7.
- Number of possible addresses: 0 to 63.

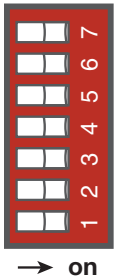
Significance of the DIP switch on the MSD Servo Drive Compact			
	DIP switch number	**Significance**	**Address**
	1	2^0	1
	2	2^1	2
	3	2^2	4
	4	2^3	8
	5	2^4	16
	6	2^5	32
	7	Bus terminating resistor (120 Ω)	

Table 4.5 DIP switch MSD Servo Drive Compact

Example for the address setting on the DIP switch:

Setting the address "3":

- Prerequisite:
  Set bus address parameter **P 2005-COM_CAN_Adr** in Moog DriveAdministrator = 0.
- Set switch 1 and 2 to "ON".
- Add together the significance of the switches $2^0 + 2^1$.
- Resulting device address = 3.

Additional notes on the address setting

NOTE:
Switch 8 on the MSD Servo Drive and 7 on the MSD Servo Drive Compact are used for the bus termination.

You device accepts changes to the address for the CANopen field bus system on a

- Reset node command.
- Device start.
- Device restart.

You will find the active bus address in the parameter **P 2058 COM_CAN_Adr_Act**.

4.4.3 Address assignment via a combination of software CAN address for the Servo Drive and the DIP switch (S4, hardware address)

Example: Setting address "33":

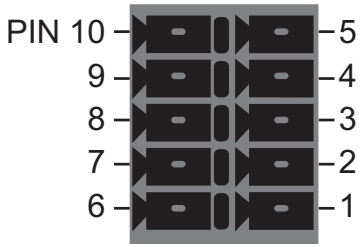
1. Set bus address parameter **P 2005-COM_CAN_Adr** (software address) to "30".
2. Set switch 1 and 2 to S4 (hardware address) to "ON"

The resulting node ID (software + hardware) is 30 + 3 = "33".

4.5 Pin assignment for terminal X32

NOTE:
1. Functionally identical pins are connected in the CANopen field bus system.
2. The CANopen field bus system requires an **external** +24 V supply!
3. The terminal connection X32 is designed for spring terminals.

Terminal block X32 pin assignment



Functionally identical pins	Function	Description
10 and 5	CAN_+24V	External +24 V supply
9 and 4	CAN_H	CAN high
8 and 3	CAN_SHLD	CAN shield (optional)
7 and 2	CAN_L	CAN low
6 and 1	CAN_GND	CAN ground (0 V)

Table 4.6 Pin assignment, terminal X32

4.6 Installation of the CANopen field bus system

4.6.1 System connection of the CANopen field bus system

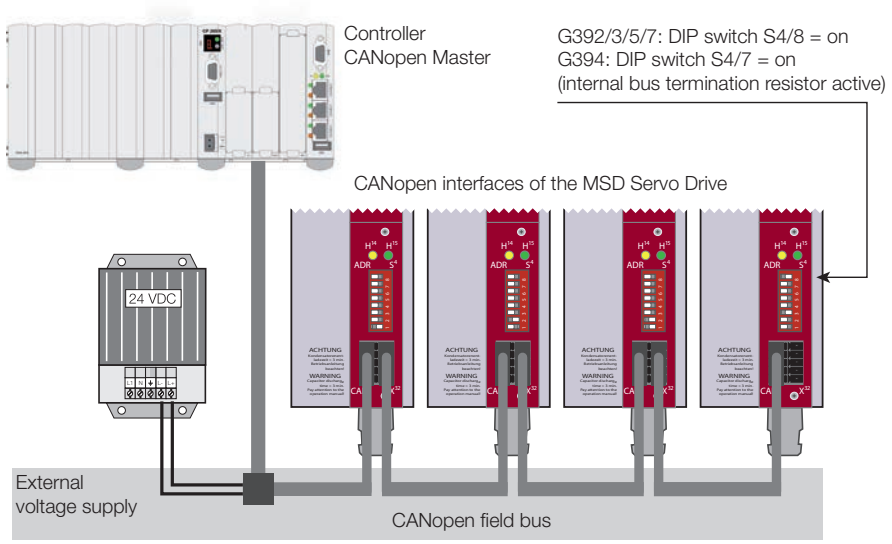


Figure 4.2 Schematic illustration of the CANopen field bus system

The CANopen field bus system:
- Is connected to the system via terminal X32.
- Is electrically isolated in relation the Servo Drive electronics.
- Receives an external 24 V power supply via the terminal X32.

4.6.2 Specification of the CANopen field bus system

Connection	Terminal X32 (spring terminal)
Wave terminating resistor - bus termination -	120 Ω (internal) Activating bus terminating resistor: • MSD Servo Drive - switch 8 on DIP switch S4 to "ON" • MSD Servo Drive Compact - switch 7 on DIP switch S4 to "ON"
Maximum input frequency	1 MHz
External power supply	+24 V +25%, 50 mA (floating in relation to the Servo Drive)
Voltage ripple	Maximum 3 Vpp
Current consumption	Maximum 50 mA per bus user
Cable type	4-core shielded

Table 4.7 Specification, CANopen field bus system

4.6.3 Transmission speeds of the CANopen field bus system (baud rates)

Baud rates for the CANopen field bus system:

Transmission speed (baud rate)	Maximum cable length over the entire network [1]	Comment
1000 kBaud	25 m (82.021 ft)	Factory setting
500 kBaud	100 m (328.084 ft)	
250 kBaud[2]	250 m (820.21 ft)	
125 kBaud[2]	500 m (1640.4199 ft)	
50 kBaud[3]	1000 m (3280.8399 ft)	
20 kBaud[3]	2500 m (8202.0997 ft)	

1) The estimated cable length for a transmission delay of 5 ns/m and total input, output delay for a device is:
   - *1 M-800 kbit/s: 210 ns*
   - *500 - 250 kbit/s: 300 ns (this figure includes 2 * 40 ns for the optocoupler)*
   - *125 kbit/s: 450 ns (this figure includes 2 * 100 ns for the optocoupler)*
   - *50 -10 kbit/s: effective delay = delay from (recessive to dominant + dominant to recessive) divided by 2.*
2) For a cable length > 200 m the usage of optocouplers is recommended. If the optocouplers are connected between the CAN controller and receiver, the optocoupler transmission delay affects the maximum possible cable length. The transmission delay of the optocoupler used reduces the maximum cable length by 4 m/10 ns.
3) For a cable length greater than 1 km, a bridge or a repeater may be required.

Table 4.8 Transmission speeds of the CANopen field bus system (baud rates)

Note:
Please ensure the permissible cable lengths for the transmission speeds are not exceeded!

4.6.4 Installation of the CANopen field bus system

1.	• Make sure the hardware enable (ENPO and ISDSH) is wired on the MSD Servo Drive (X4, control connections) (connected to 24 V) (see chapter “4.6.5 Hardware enable” on page 20).	See Operation Manual "Specification of control connections": • MSD Servo Drive Compact CA97555-001 • MSD Servo Drive Single-Axis System CA65642-001 • MSD Servo Drive Multi-Axis System CA97554-001
2.	• Connect the signal wires for the CANopen field bus system (terminal X32).	See: • “4.5 Pin assignment for terminal X32” on page 18 • “4.6.1 System connection of the CANopen field bus system” on page 19
3.	• Connect the external power supply +24 V.	
4.	• Activate the internal bus terminating resistor on the last Servo Drive.	
5.	• Switch on the Servo Drive.	

You will find further information in the chapter "Commissioning and configuration of the CANopen field bus system"

4.6.5 Hardware enable

Prerequisite for the operation of the power stage (see the Operation Manuals MSD Servo Drive and MSD Servo Drive Compact):

1. Connect control input for the hardware enable, ENPO on the MSD Servo Drive and MSD Servo Drive Compact, on the control connections (X4) to 24 V.
2. Connect control input STO (Safe Torque Off), ISDSH on the MSD Servo Drive and MSD Servo Drive Compact, on the control connections (X4) to 24 V.
3. The higher-level controller must include the logic for the function STO (category 3, PL d, EN ISO 13849-1) as per the "MSD Servo Drive - Device Help".

NOTE:
If the control inputs ENPO and ISDSH on the control connections (X4) on the MSD Servo Drive or MSD Servo Drive Compact are not connected, the device remains in the:

- State 1 = "Not ready to switch on" or
- State 2 = "Switch on disabled".

4.7 Device state of the Servo Drive

A 7-segment display integrated into the control unit for the Servo Drive (MSD Servo Drive and MSD Servo Drive Compact) indicates the device state.

You will find the most important information on the device state of the Servo Drive in the following table. "**D1**" is the **1st digit** and "**D2**" the **2nd digit** on the 7-segment display:

Device state of the Servo Drive			
D1	**D2**	**Description**	
8.	8.	Device in reset state	
	0th	Initialisation on device start	(Start)
S. *)	1.	1) Not ready to switch on (no DC link voltage)	(NotReadyToSwitchOn)
S. *)	2.	1) Switch on disabled (DC link in order, power stage not ready)	(SwitchOnDisabled)
	3.	Ready to switch on (power stage ready)	(ReadyToSwitchOn)
	4.	Switched on (device is electrically live) 2)	(SwitchedOn)
	5.	Operation enable (power applied to drive and drive ready for setpoint input) 2)	(OperationEnable)
	6.	Quick stop 2)	(QuickStopActive)
	7.	Fault reaction active 2)	(FaultReactionActive)
E	r	Fault (see below)	(Fault)
Indication if there is a fault			
E	r.	• Fault without "point" - fault that can be acknowledged as per your programming. • Fault indication with "point" - fault that can only be acknowledged after rectifying the cause of the fault.	
X	X	Fault number (decimal)	
Y	Y	Fault location (decimal)	

1) S. flashes if the function STO (Safe Torque Off) is active, indication extinguishes if function is inactive.
*) This is not a "safe indication" in the context of IEC/EN 61800-5-2.
2) The dot flashes when the power stage is active.

NOTE:
You will find detailed information on the device state of the Servo Drive (MSD Servo Drive and MSD Servo Drive Compact) in the Operation Manuals and Device Help for the MSD Servo Drive product ranges.

Example for a sequence of flashing

➢ [Er > 02 > 05] * [Er > 02 > 05] * [Er > 02 > 05] ...

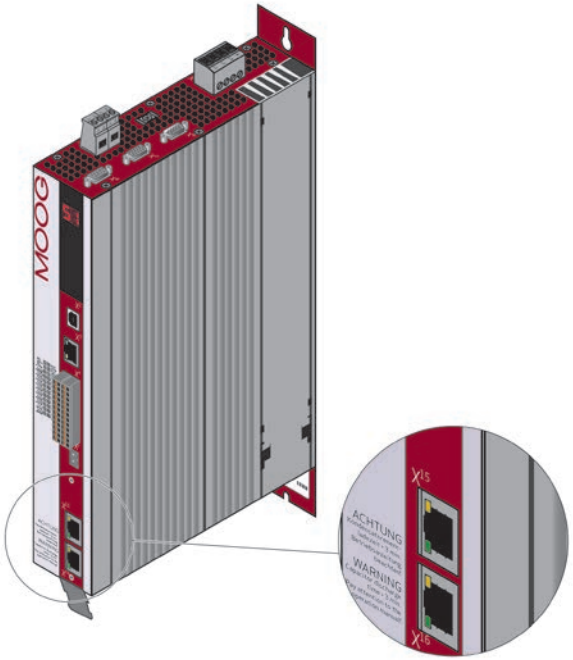
Er	Fault:	Er = "Error"
02	Fault number:	02 = "Error in the parameter list"
05	Fault location:	05 = "Check the actual values in the parameter list"

NOTE:
For information on fault numbers and fault locations refer to the fault list in the "MSD Servo Drive – Device Help" (available as PDF file and in HTML format in the download area at Moog).

5 Installation and connection of the EtherCAT field bus system

5.1 Position

The illustration (below) shows the position of the EtherCAT field bus system in the MSD Servo Drive. The position of the EtherCAT field bus system in the MSD Servo Drive Compact is identical.



NOTE:
The EtherCAT field bus system is designed as a variant for the product ranges MSD Servo Drive Compact, Single-Axis and Multi-Axis Servo Drive!

CAUTION!	Risk of injury and/or damage due to electrical power!
	• Touching bare, electrically live wires or wires from which the insulation has been stripped can cause an electric shock and burns. • Short circuits can cause damage to the device! On mounting electrical components, e.g. wires and cables, make sure electrical power is not present! If necessary, replace damaged cables!

5.2 Connections and controls

The figure below shows the position of the connections, controls and diagnostic LEDs for the EtherCAT field bus system on the MSD Servo Drives.

EtherCAT for the MSD Servo Drive (Single-Axis and Multi-Axis Servo Drive G392/G395/G393/G397)	EtherCAT for the MSD Servo Drive Compact (G394)

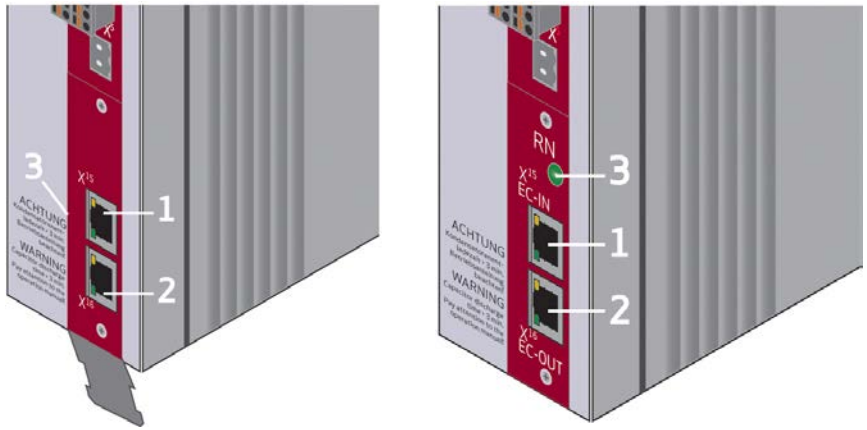


Figure 5.1 Layout, connections and controls, EtherCAT

Connection	Designation	Function
1	X15	RJ45 socket - ETHERCAT IN
2	X16	RJ45 socket - ETHERCAT OUT
3	RN	LED - ETHERCAT RUN

Table 5.1 Connections for the EtherCAT field bus system

Note:
On the device series MSD Servo Drive Compact the connectors X15/X16 are rotated by 180° compared to the device series Single-Axis and Multi-Axis System.

5.3 Light emitting diodes - and field bus system flashing codes

5.3.1 MSD Servo Drive (Single-Axis and Multi-Axis Servo Drive)

Flashing codes on the LEDs on the RJ45 sockets for the EtherCAT field bus system (MSD Servo Drive - Single-Axis and Multi-Axis Servo Drive, without additional LED):

Connector X15			
LED	**EtherCAT - network state**		**Figure**
L/A (green)	**Link/ Activity**	**Off = No link** • No connection to another bus user	
		On = Link • Connection to another bus user • No data exchange	
		Flashing = Activity • Data exchange active	
RN (green)	**RUN**	**Off = Initialisation** • Device is in the "Initialisation" state	
		Flashing = Pre-Operational • Device is in the "Pre-operational" state	
		Single flash = Safe-Operational • Device is in the "Safe-operational" state	
		On = Operational • Device is ready	

Table 5.2 Flashing codes on the LEDs for the EtherCAT field bus system on the MSD Servo Drive (Single-Axis and Multi-Axis System)

Connector X16			
LED	**EtherCAT - network state**		**Figure**
L/A (green)	**Link/ Activity**	**Off = No link** · No connection to another bus user	
		On = Link · Connection to another bus user · No data exchange	
		Flashing = Activity · Data exchange active	
Bottom LED (green)	-	· No function	

Table 5.3 Flashing codes on the LEDs for the EtherCAT field bus system on the MSD Servo Drive (Single-Axis and Multi-Axis System)

5.3.2 MSD Servo Drive Compact

The flashing codes on the LEDs on the RJ45 sockets
(**MSD Servo Drive Compact**, with additional EtherCAT RUN (RN) LED):

LED	EtherCAT - network state		Figure
Top LED (green)	**Link/ Activity**	**Off = No link** · No connection to another bus user	
		On = Link · Connection to another bus user · No data exchange	
		Flashing = Activity · Data exchange active	
Bottom LED (green)	**Link (PHY)**	**Off = No link** · No connection to another bus user	
		On = Link · Connection to another bus user · No data exchange	
LED **RN**	**RUN**	**Green = RUN**	
		Off = Initialisation · Device is in the "Initialisation" state	
		Flashing = Pre-Operational · Device is in the "Pre-operational" state	
		Single flash = Safe-Operational · Device is in the "Safe-operational" state	
		On = Operational · Device is ready	

Table 5.4 Flashing codes for the EtherCAT field bus system on the MSD Servo Drive Compact

5.3.3 ERROR indicator MSD Servo Drive and MSD Servo Drive Compact

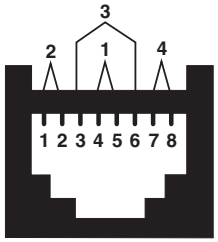
The seven segment display indicates the following EtherCAT errors:

Er 14 01	EtherCAT Watchdog Timeout
Er 14 02	Wrong EEP data for EtherCAT communication controller or not supported controller
Er 14 03	Internal ram error
Er 14 04	Invalid configuration
Er 14 05	Missing EtherCAT receive process data on Sync Manager 2
Er 14 06	Local error

On this topic, see also chapter 5.6 on page 28.

5.4 Pin assignment of the RJ45 socket

EIA/TIA-568A and EIA/TIA-568B are standards for 8-pin RJ45 connectors.
The difference between the standards is pairs 2 and 3 are swapped. The pin assignment given below shows the colour code for the standard EIA/TIA-568B.



Pin	EIA/TIA-568B Core pair number	Function	EIA/TIA-568B colour
1	2	Tx Data+	White/orange stripe
2	2	Tx Data-	Orange/white stripe or orange
3	3	Rx Data+	White/green stripe
4	1	Not used	Blue/white stripe or blue
5	1	Not used	White/blue stripe
6	3	Rx Data-	Green/white stripe or green
7	4	Not used	White/brown stripe
8	4	Not used	Brown/white stripe or brown

Table 5.5 Pin assignment for the RJ45 socket with the standard EIA/TIA-568B

5.5 Installation of the EtherCAT field bus system

5.5.1 System connection of the EtherCAT field bus system

Schematic illustration of the EtherCAT field bus system:

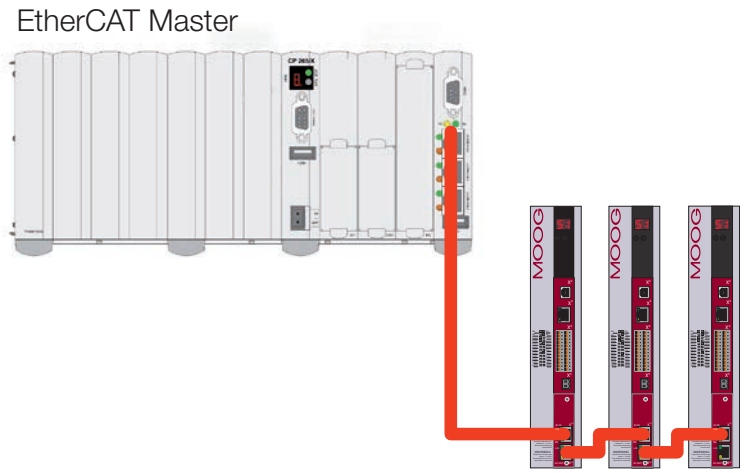


Figure 5.2 EtherCAT topology

Connect system in the form of a line topology:

- The EtherCAT master is at the start of the line.
- Each EtherCAT slave has one input port (X15, RJ45) and one output port (X16, RJ45).
- Connect the EtherCAT slaves in a line (see figure above).
  For this purpose connect the cable coming from the:
  - EtherCAT master to the input port on the 1st EtherCAT slave.
  - Connect the output port on the 1st EtherCAT slave to the input on the 2nd EtherCAT slave and so on.
- On the last EtherCAT slave in the line the output port remains open.

An open output on the last EtherCAT slave in the line results in a logical short circuit on the transmit (Tx) and receive (Rx) lines in the EtherCAT network. For this reason every EtherCAT network is to be considered a logical ring.

CAUTION!	Damage due to incorrect wiring
	Faults due to incorrect wiring, mixing up the input and output can result in erroneous addressing. In the worst case your drive system will be damaged. For this reason: • Have the EtherCAT field bus system installed by authorised personnel. • Follow the procedure described and contact the Helpline at Moog if necessary.

NOTE:
For the best performance and quality of the data transmission use certified, tested, ready-made cables of category CAT 5e.

CAUTION!	Damage due to incorrect installation
	Never use EtherCAT and standard Ethernet field bus systems in one physical network! This situation will degrade the drive system and can result in the failure of the communication. In the worst case your drive system will be damaged. For this reason: • Have the EtherCAT field bus system installed by authorised personnel. • Only use certified component(s) for an EtherCAT infrastructure. • To differentiate between field bus systems, use a different cable colour for the EtherCAT field bus system!

5.5.2 Hardware enable

Prerequisite for the operation of the power stage (see the Operation Manuals MSD Servo Drive and MSD Servo Drive Compact):

1. Connect control input for the hardware enable, ENPO on the MSD Servo Drive and MSD Servo Drive Compact, on the control connections (X4) to +24 V.
2. Connect control input STO (Safe Torque Off), ISDSH on the MSD Servo Drive and MSD Servo Drive Compact, on the control connections (X4) to +24 V.
3. The higher-level controller must include the logic for the function STO (category 3, PL d, EN ISO 13849-1) as per the "MSD Servo Drive - Device Help".

NOTE:
If the control inputs ENPO and ISDSH on the control connections (X4) on the MSD Servo Drive or MSD Servo Drive Compact are not connected, the device remains in the:

- State 1 = "Not ready to switch on" or
- State 2 = "Switch on disabled".

5.6 Device state of the Servo Drive

A 7-segment display integrated into the control unit for the Servo Drive (MSD Servo Drive and MSD Servo Drive Compact) indicates the device state.

You will find the most important information on the device state of the Servo Drive in the following table. "**D1**" is the **1st digit** and "**D2**" the **2nd digit** on the 7-segment display:

Device state of the Servo Drive			
D1	**D2**	**Description**	
8.	8.	Device in reset state	
	Oth	Initialisation on device start	(Start)
S. *)	1.	1) Not ready to switch on (no DC link voltage)	(NotReadyToSwitchOn)
S. *)	2.	1) Switch on disabled (DC link in order, power stage not ready)	(SwitchOnDisabled)
	3.	Ready to switch on (power stage ready)	(ReadyToSwitchOn)
	4.	Switched on (device is electrically live) 2)	(SwitchedOn)
	5.	Operation enable (power applied to drive and drive ready for setpoint input) 2)	(OperationEnable)
	6.	Quick stop 2)	(QuickStopActive)
	7.	Fault reaction active 2)	(FaultReactionActive)
E	R	Fault (see below)	(Fault)
Indication if there is a fault			
Device state of the Servo Drive			
D1	**D2**	**Description**	
E	r.	• Fault without "point" - fault that can be acknowledged as per your programming. • Fault indication with "point" - fault that can only be acknowledged after rectifying the cause of the fault.	
X	X	Fault number (decimal)	
Y	Y	Fault location (decimal)	

1) S. flashes if the function STO (Safe Torque Off) is active, indication extinguishes if function is inactive.
*) This is not a "safe indication" in the context of IEC/EN 61800-5-2.
2) The dot flashes when the power stage is active.

Table 5.6 State indication, Servo Drive

NOTE:
You will find detailed information on the device state of the Servo Drive (MSD Servo Drive and MSD Servo Drive Compact) in the related Operation Manual.

Example for a sequence of flashing

➢ [Er > 02 > 05] * [Er > 02 > 05] * [Er > 02 > 05] ...

Er	Fault:	Er = "Error"
02	Fault number:	02 = "Error in the parameter list"
05	Fault location:	05 = "Check the actual values in the parameter list"

NOTE:
For information on fault numbers and fault locations refer to the fault list in the "MSD Servo Drive – Device Help" (available as PDF file in the download area at Moog).

6 Commissioning and configuration of the CANopen field bus system

6.1 PC user software Moog DriveAdministrator 5

The PC user software **"Moog DriveAdministrator 5"** is used to commission the drive system and the CANopen communication.

Technical data	Moog DriveAdministrator 5

Support for the following functions	• Initial commissioning of your:
	  • Drive system
	  • CANopen device network

	• Quick serial commissioning using a configurable commissioning file (contains firmware, data set, Moog PLC program).
	• Operation and diagnostics, including using Cockpit and 6-channel oscilloscope.
	• Project management.

Table 6.1 Technical data Moog DriveAdministrator 5

NOTE:
You will find more detailed information on installing and starting the PC user software **Moog DriveAdministrator 5** in the "MDA5 Installation Manual" (available as a PDF file in the download area at https://www.moogsoftwaredownload.com/msd.html).

You will find more detailed information on the initial commissioning and optimisation of the software functions and control circuits in the Operation Manual and the MSD Servo Drive - Device Help for the related device (available as PDF file in the download area at https://www.moogsoftwaredownload.com/msd.html).

6.2 Electronic device data sheet – EDS file

The EDS file is required to operate the Servo Drive in the CANopen device network.
The electronic data sheet (EDS):

- Is installed in the "CANopen master" (controller). Is used to integrate MSD Servo Drives in the CANopen device network.
- Is available as a download from https://www.moogsoftwaredownload.com/msd.html under “FW”.

NOTE:
The version of the eds file can change with new firmware.

6.3 Commissioning and configuration

To commission and configure your Servo Drive and your CANopen device network, you will require the following documents:

1. Operation Manual for your Servo Drive.
2. MSD Servo Drive – Device Help.
3. MSD Servo Drive – Moog DriveAdministrator 5 - Program Help (implemented as context-sensitive help in MDA5).
4. User Manual CANopen/EtherCAT.

6.3.1 Procedure for commissioning and configuration

1.	Connect, wire and configure the Servo Drive and additional components as per the Operation Manual.
2.	Supply the Servo Drive with +24 V control voltage.
3.	Undertake the initial commissioning of the Servo Drive as per the Device Help (power stage, motor, encoder, control, etc.). Note the following points: The hardware enable ENPO and ISDSH on terminal X4 is not connected on standard devices.
4.	Configure timing parameters (see page 31), baud rate (see page 31) and node ID (see 16) in the Servo Drive.
5.	Save settings in the Servo Drive.
6.	Restart Servo Drive.
7.	Set-up CANopen network as per chapter 4.6.1. Note the following points: The CANopen field bus system requires an external 24 V power supply via terminal X32!
8.	Integrate the EDS file in the CANopen master and make configuration settings for the master (cycle time, baud rate, node ID, mapping). Note the following points: Make sure the CANopen master transfers the mapping to the Servo Drive on boot-up!
9.	Switch on supply of power (mains).
10.	Select the operation mode for the Servo Drive via SDO access to object 0x6060 (Modes of operation). Supported operation modes are listed in chapter 10.
11.	Test quality of control and optimise if necessary.

Table 6.2 Commissioning the CANopen field bus system

6.4 CANopen operation via the CiA402 function implemented in the Servo Drive

The Servo Drive operation mode is selected via the CiA402 function "Modes of operation" implemented. (See chapter 10 and chapter 12).

The individual configurations for the operation of the Servo Drive on the CANopen field bus system are described in the following.

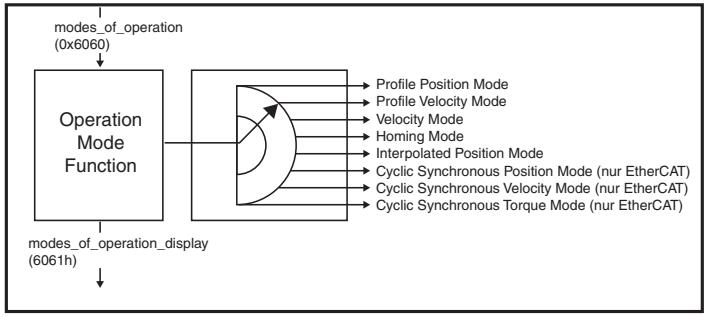
6.4.1 Selection of the operation mode (Modes of operation)

During initial commissioning, the parameters in the Servo Drive are configured with motor data, control settings, I/O configurations and other functions.

The operation mode is selected via the **CiA402 object 0x6060 (Modes of operation)**. You will find the operation modes supported by the Servo Drive in chapter 10 and chapter 12.

The control mode is coupled to the operation mode. There are 3 control modes:

1. Position control.
2. Velocity control.
3. Torque control.



You can change between different operation modes. You will find the operation modes supported in object 6502h (supported Drive Modul).

6.4.2 Setting timing parameters

Three timing parameters are responsible for the communication between the Servo Drive and controller. Note the following points:

- Set the 3 timing parameters to the same value (bus cycle time).
- Ensure the 3 timing parameters have different units.

Parameter	Object name	Unit
P 0306	Internal interpolator – cycle time	ms
P 2015	Communication cycle period / 0x1006 (only CANopen)	µs
P 2266 Index 0 Index 1	0x60C2 Interpolation time base Interpolation time exponent	s

Table 6.3 3 timing parameters

6.5 CANopen configuration

6.5.1 Setting software address and baud rate via Moog DriveAdministrator

Set the software address and the baud rate of the CANopen field bus system using the following parameters:

Parameter	Object name	Description
P 2005 COM_CAN_Adr	Software address CANopen	Address definition via parameter, see chap. 4.4.
P 2006 COM_CAN_Adr	Baud rate CANopen	Permissible baud rates, see chap. 4.6.3.

Table 6.4 Parameters for software address and baud rate

NOTE:

1. You will find the active bus address in the boot-up message and in parameter **P 2058 COM_CAN_Adr_Act** (Addition of software and hardware address).
2. "Baud rate" factory setting for the MSD Servo Drive: 1000 kbit. Changes to the baud rate in parameter **P 2006 COM_CAN_Baudrate** are only applied after restarting the Servo Drive.
3. Parameter **P 2059 COM_CAN_Baudrate_Act**. indicates the actual baud rate.

6.5.2 Commissioning instructions

NOTE:
Reasons why a Servo Drive does not respond to a telegram:

- The telegram frame (baud rate, data width) from the controller (CANopen master) is erroneous.
- A Servo Drive in CANopen device network has the wrong bus address or several Servo Drives have the same bus address.
- Incorrectly established serial connection between controller (CANopen master) and Servo Drive.
- The +24 V supply on the CANopen connection is missing or the wiring is faulty.
- The Servo Drive is not in the NMT network state"Operational" or "Pre-Operational".
  Checking the actual network state:
  Parameter P 2060 COM_CAN_NMT_State

Parameter P 2060	NMT - network state
0	Bootup
1	Init
4	Stopped/Safe OP
5	Operational
127	Pre-Operational

Table 6.5 CANopen network states

6.5.3 System test on the controller (CANopen master)

Activating modified settings:

- Switch off and on Servo Drive.
- Wait for initialisation time of a few seconds to elapse (depending on the number of drive controllers in the CANopen device network)
- Drive controller sends boot-up message once (ID 700h + Node ID = 701h for device address 1). If this is the case, the communication is working.

6.5.4 Saving parameter settings

Save Servo Drive configuration data (e.g. before "Resetting to factory setting", see next chapter):

1. Via Moog DriveAdministrator 5 (see PC user software Moog DriveAdministrator 5 - Program Help).
2. Write the value 1 to the field bus system in sub index 1 of **object 200Bh PARA_SetCmd** .
   Once the data have been saved the field bus automatically sets sub index 1 of **object 200Bh PARA_SetCmd** to the value 0. This change indicates the completion of the procedure and is used to monitor the timing.
3. Another method is the **object 1010h Store parameters** that is specified in the CiA301.

Please note:
Some objects are "**objects that cannot be saved"**; these objects are automatically processed and initialised by the controller (CANopen master) (e.g. **object 0x6060 Modes of operation**).

6.5.5 Resetting parameter settings to factory setting

There are 2 ways to set the parameter settings for the Servo Drive to the factory setting:

1. Via field bus system:
   - Write the value 1 to sub index 3 of the **object 200Bh PARA_SetCmd**. **Please note:** Affects the "Baud rate/device address" setting!
   - Write the value 1 to sub index 1 of the **object 200Bh PARA_SetCmd**. This saves the Servo Drive's data.
   - The change is effective after a "Reset node" command or device restart.
2. Via Moog DriveAdministrator 5:
   - Select the corresponding Servo Drive in the tree structure in MDA.
   - Save Servo Drive's data in a file (see "Moog DriveAdministrator - Program Help")
   - On the menu bar under "Active device", select the command "Reset to factory settings".

NOTE:
After the "Reset to factory settings", in both cases it takes around 10 s for the Servo Drive to signal it is "ready". During this time there is a self-test and the "Reset to factory setting" takes place.

Caution	Do not switch off Servo Drive during the initialisation/restart!
	• There is a risk of loss of data and settings. Wait until the Servo Drive has completed the initialisation/the restart (10 s)!

7 Commissioning and configuration of the EtherCAT field bus system

7.1 Moog DriveAdministrator 5

Moog DriveAdministrator 5 is a PC user software application with graphical user interface. It is used for initial commissioning and serial commissioning, and for the operation of the drive system. It also facilitates implementation of the EtherCAT communication.

Moog DriveAdministrator 5	Tasks/ functions
Supports you during the following tasks	• Initial commissioning of your drive system • Configuration of your EtherCAT device network
	• Quick serial commissioning using a configurable commissioning file (contains firmware, parameters,Moog PLC program).
	• Operation and diagnostics, including using Cockpit and 6-channel oscilloscope.
	• Project management.

Table 7.1 Overview of Moog DriveAdministrator 5 functions

NOTE:
You will find more detailed information on installing and starting the PC user software **Moog DriveAdministrator 5** in the "MDA5 Installation Manual" (available as a PDF file in the download area at https://www.moogsoftwaredownload.com/msd.html).

You will find more detailed information on the initial commissioning and optimisation of the software functions and control circuits in the Operation Manual and the MSD Servo Drive - Device Help for the related device (available as PDF files and html files in the download area at https://www.moogsoftwaredownload.com/msd.html).

7.2 ESI (xml) – field bus device description file

A prerequisite for the operation of the Servo Drive in the EtherCAT network is a field bus device description file = ESI file (xml):

- Is installed in the "EtherCAT master" (controller).
- Is used to integrate drive controllers into the EtherCAT device network.
- Transfers data, properties and functionality of the field bus system to the "EtherCAT master".
- Provides the manufacturer of the EtherCAT bus user.
- Is available as a download from https://www.moogsoftwaredownload.com/msd.html under "FW".

NOTE:
The version of the ESI file (xml) can change with new firmware.

7.3 Commissioning and configuration

To commission and configure your Servo Drive and your EtherCAT device network, you will require the following documents:

1. Operation Manual for your Servo Drive.
2. MSD Servo Drive – Device Help.
3. MSD Servo Drive – Moog DriveAdministrator 5 - Program Help (implemented as context-sensitive help in MDA5).
4. User Manual CANopen/EtherCAT.

Procedure for commissioning and configuration

1.	Connect, wire and configure the Servo Drive and additional components as per the Operation Manual.
2.	Supply the Servo Drive with 24 V control voltage.
3.	Undertake the initial commissioning of the Servo Drive as per the Device Help (power stage, motor, encoder, control, etc.). Note the following points: The hardware enable ENPO and ISDSH on terminal X4 is not connected on standard devices.
4.	Configure timing parameters in the Servo Drive (page chapter 7.4.2).
5.	Implement scaling and units in the Servo Drive similar to scaling in the master. See chapter 10.3.
6.	Save settings in the Servo Drive.
7.	Restart Servo Drive.
8.	Set-up EtherCAT network as per chapter 5.5.1.
9.	Integrate the ESI (xml) file in the EtherCAT master and make configuration settings for the master (cycle time, mapping (from ESI file or manually)). Note the following points: Make sure the EtherCAT master transfers the mapping to the Servo Drive on boot-up!
10.	Switch on supply of power (mains).
11.	Select the operation mode for the Servo Drive via SDO access to object 0x6060 (Modes of operation). Supported operation modes are listed in chapter 10.
12.	Test quality of control and optimise if necessary.

Table 7.2 Commissioning the CANopen field bus system

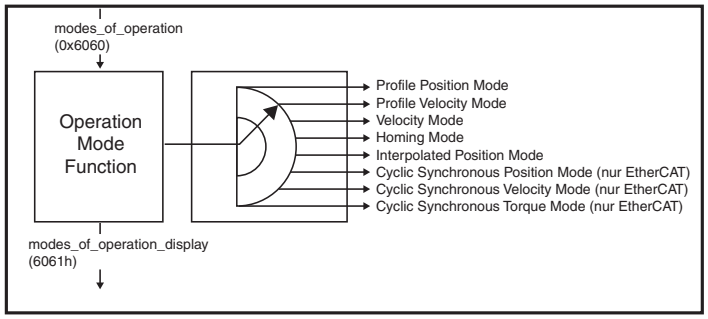
7.4 EtherCAT operation via the CiA402 function implemented in the Servo Drive

7.4.1 Selection of the operation mode (Modes of operation)

The operation mode is selected via the **CiA402 object 0x6060 (Modes of operation)**. You will find the operation mode supported by the MSD Servo Drive in chapter 10 and chapter 12.

A control mode is coupled to the operation mode. There are 3 control modes:

1. Position control.
2. Velocity control.
3. Torque control.



You can change between different operation modes. Please only select operation modes supported by your Servo Drive.

7.4.2 Setting timing parameters

Two timing parameters are responsible for the communication between the MSD Servo Drive and the controller. Note the following points:

- Set the 2 timing parameters to the same value (bus cycle time).
- The 2 timing parameters have different units.

Parameter	Object name	Unit
P 0306	Internal interpolator – cycle time	ms
P 2266 Index 0 Index 1	0x60C2 Interpolation time base Interpolation time exponent	s

Table 7.3 2 timing parameters

NOTE:
For the EtherCAT field bus system set the parameter **P 2266 MPRO_402_IntTimePeriod (object 0x60C2)** to the cycle time for the setpoints (or the telegrams).

7.5 EtherCAT configuration

7.5.1 Commissioning instructions

NOTE:
Reasons why a Servo Drive does not respond to a telegram:

The Servo Drive is not in the NMT network states "Operational" or "Pre-Operational"!

- Checking the actual network state:
  Parameter P 2060 COM_CAN_NMT_State

Parameter 2060	NMT - network state
0	Bootup
1	Init
4	Stopped/Safe OP
5	Operational
127	Pre-Operational

Table 7.4 EtherCAT network states

7.5.2 System test on the controller (EtherCAT master)

Activating modified settings:

- Switch off and on Servo Drive.
- Wait for initialisation time of a few seconds to elapse.

7.5.3 Saving parameter settings

Save Servo Drive configuration data (e.g. before "Resetting to factory setting", see next chapter):

4. Via Moog DriveAdministrator 5 (see PC user software Moog DriveAdministrator 5 - Program Help).
5. In the field bus system, write the value 1 to sub index 1 of **object 200Bh PARA_SetCmd**.
   Once the data have been saved the field bus automatically sets sub index 1 of **object 200Bh PARA_SetCmd** to the value 0. This change indicates the completion of the procedure and is used to monitor the timing.
6. Another method is the **object 1010h Store parameters** that is specified in the CiA301.

Please note:
Some objects are "**objects that cannot be saved"**; these objects are automatically processed and initialised by the controller (CANopen master) (e.g. **object 0x6060 Modes of operation**).

7.5.4 Resetting parameter settings to factory setting

There are 2 ways to set the parameter settings for the Servo Drive to the factory setting:

1. Via field bus system:
   – Write the value 1 to sub index 3 of the **object 200Bh PARA_SetCmd**.
   Please note: Affects the "Baud rate/device address" setting!
   – Write the value 1 to sub index 1 of the **object 200Bh PARA_SetCmd**.
   This saves the Servo Drive's data.
   – The change is effective after a "Reset node" command or device restart.
2. Via Moog DriveAdministrator 5:
   – Select the corresponding Servo Drive in the tree structure in MDA.
   – Save Servo Drive's data in a file (see "Moog DriveAdministrator - Program Help")
   – On the menu bar under "Active device", select the command "Reset to factory settings".

NOTE:
After the "Reset to factory settings", in both cases it takes around 10 s for the Servo Drive to signal it is "ready". During this time there is a self-test and the "Reset to factory setting" takes place.

Caution	Do not switch off Servo Drive during the initialisation/restart!
	• There is a risk of loss of data and settings. Wait until the Servo Drive has completed the initialisation/the restart (10 s)!

7.6 Ethernet over EtherCAT (EoE)

To minimise the wiring effort, in particular in series production machine manufacture, it is recommended to implement the communication between the PC user software application Moog DriveAdministrator 5 and the drive controllers using Ethernet over EtherCAT (EoE). Here the Ethernet TCP/IP communication is added to the EtherCAT data traffic in a tunnel.

A separate network connection to the service and diagnostics interface X3 (MSD Servo Drive, MSD Servo Drive Functional Safety) and X9 (MSD Servo Drive Compact) is then not required.

NOTE:
Due to the amount of data, it is recommended to undertake the initial commissioning of the axis controller using Ethernet TCP/IP via the service and diagnostics interface X3 (MSD Servo Drive, MSD Servo Drive Functional Safety) and X9 (MSD Servo Drive Compact)!

The service and diagnostics interface has the following IP configuration as delivered:

IP address:	192.168.39.5
Subnet mask:	255.255.255.0

On the assignment of the virtual IP address (EoE IP address for the EtherCAT port (X15/X16) by the EtherCAT master, it is to be noted that this is not in the same network (192.168.39.0)!

During the configuration of the network for the usage of EoE, a series of boundary conditions must be met to make possible communication between Moog DriveAdministrator 5 and the Servo Drives.

Here a differentiation must be made between 2 application cases:

1. Moog DriveAdministrator 5 installed on service PC
2. Moog DriveAdministrator 5 installed on EtherCAT master PC

7.6.1 Application case I: Moog DriveAdministrator 5 installed on service PC

In this application case, Moog DriveAdministrator 5 is installed on a separate service PC. The network structure is as follows:

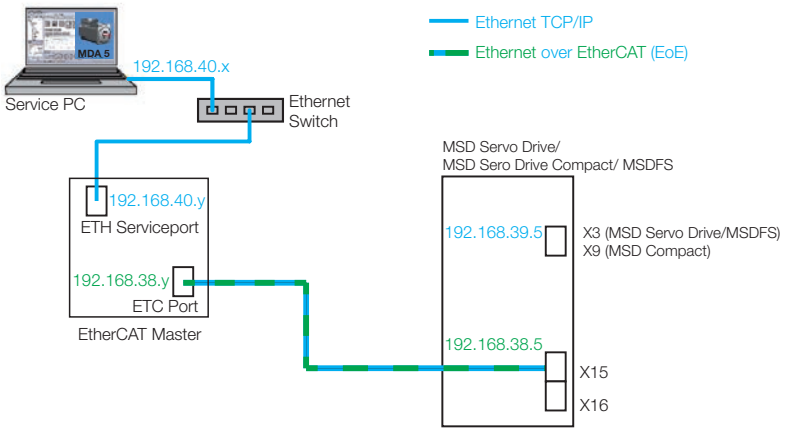


Figure 7.1 Application case I: Moog DriveAdministrator 5 installed on service PC

Network configuration service PC

IP address:	192.168.40.x
Subnet mask:	255.255.255.0

Network configuration EtherCAT master (Ethernet service interface)

IP address:	192.168.40.y
Subnet mask:	255.255.255.0

Network configuration EtherCAT master (EtherCAT interface)	
IP address:	192.168.38.y
Subnet mask:	255.255.255.0

Network configuration Servo Drive (EtherCAT interface, EoE)	
IP address:	192.168.38.5
Subnet mask:	255.255.255.0

Network configuration Servo Drive (service and diagnostics interface)	
IP address:	192.168.39.5
Subnet mask:	255.255.255.0

On the service PC, you must use the command window as an administrator to configure a static route between the Ethernet TCP/IP network and the EoE network. The EtherCAT master PC must support IP routing for this purpose and this feature must also be activated.

The command for adding the static route is:

route add destination_network mask subnet_mask gateway_IP

In the example shown:

route add 192.168.38.0 mask 255.255.255.0 192.168.40.y

It is important for this application case that the device addresses

1. Servo Drive service and diagnostics interface
2. Servo Drive EtherCAT interface and EtherCAT master EtherCAT interface
3. Service PC and EtherCAT master Ethernet service interface

are in three different networks!

7.6.2 Application case II: Moog DriveAdministrator 5 installed on EtherCAT master PC

In this application case, Moog DriveAdministrator 5 is installed on the EtherCAT master PC.
The network structure is installed as follows:

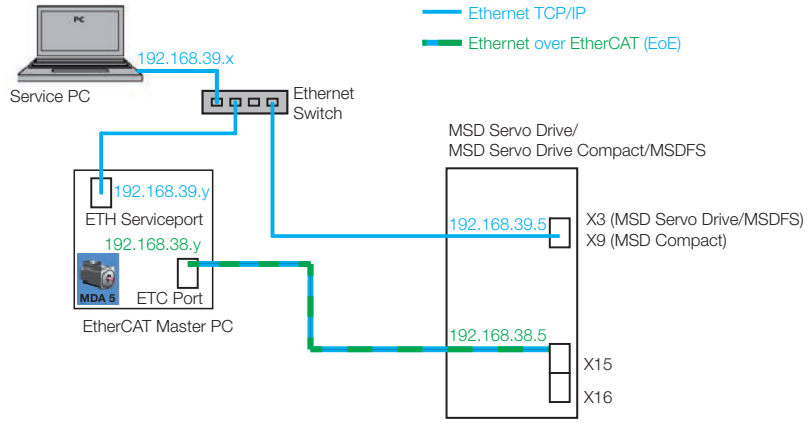


Figure 7.2 Application case II: Moog DriveAdministrator 5 installed on EtherCAT master PC

Network configuration service PC	
IP address:	192.168.39.x
Subnet mask:	255.255.255.0

Network configuration EtherCAT master (Ethernet service interface)	
IP address:	192.168.39.y
Subnet mask:	255.255.255.0

Network configuration EtherCAT master (EtherCAT interface)

IP address:	192.168.38.y
Subnet mask:	255.255.255.0

Network configuration Servo Drive (EtherCAT interface, EoE)

IP address:	192.168.38.5
Subnet mask:	255.255.255.0

Network configuration Servo Drive (service and diagnostics interface)

IP address:	192.168.39.5
Subnet mask:	255.255.255.0

For this application case, only two different networks are required:

1. Service PC and EtherCAT master Ethernet service interface, as well as Servo Drive service and diagnostics interface
2. Servo Drive EtherCAT interface and EtherCAT master EtherCAT interface

8 Configuring the parameters for the devices - CANopen

8.1 CiA301 functionality implemented

The CiA301 profile offers a collection of CAN communication services, without exactly defining the application. The application-specific drive profile CiA402 is based on these CAN communication services.

8.1.1 CAN communication services in the CiA301 profile

- Boot-up according to CiA301 V4.01 (guarding boot via identifier 700h).
- 4 variable, mappable TxPDOs (transmission type 1 to 240, 254 and 255 dec possible).
- 4 variable, mappable RxPDOs (transmission type 1 to 240, 254 and 255 dec possible).
- An emergency object error code according to CiA402 plus manufacturer-specific error locationand number, operating hours of the device.
- A sync object. The NMT master (controller) synchronises the slaves in the network using sync objects.
- NMT state machine according to CiA301.
- Node guarding and heartbeat.
- Operating cycle:
  Minimum cycle time for PDO protocols (1 ms).
  Protocols arriving faster overwrite previous protocols.
- SDO protocols and NMT services are processed acyclically.
  Typical processing times are between 1 and 5 ms.
- Initialisation values for the COB IDs (communication object IDs) according to predefined connection set (pre-defined setting for a master/slave connection).
- Access to device parameters 2000h - 5FFFh (expedited/non-expedited)

8.1.2 Communication objects and object dictionary for the CiA301 profile

The central instance of all CANopen nodes is the so-called object dictionary, like all other field bus protocols. Every CANopen device requires the so-called object dictionary. Along with standardised entries, this dictionary contains the "communication objects" relevant for the device.

You will find a complete overview of the CAN objects supported (CiA301, CiA402 and manufacturer-specific) for the MSD Servo Drive in the eds file (under https://www.moogsoftwaredownload.com/msd.html).

The following table gives an overview with important CiA301 objects:

Object	Object name	Object code	Type	Attr.
0x1000	Device_type	VAR	Unsigned32	ro
0x1001	Error_register	VAR	Unsigned8	ro
0x1003	Pre-Defined_error_field one subentry	ARRAY	Unsigned32	ro
0x1005	COB-ID_SYNC	VAR	Unsigned32	rw
0x1006	Communication_cycle_period	VAR	Unsigned32	rw
0x1007	Synchronous_window_length	VAR	Unsigned32	rw
0x1008	Manufacturer device name	String		
0x1009	Manufacturer hardware version	String		
0x100A	Manufacturer software version	String		
0x100C	Guard_time	VAR	Unsigned16	
0x100D	Life_time_factor	VAR	Unsigned8	
0x1010	Store parameters	ARRAY	Unsigned32	rw
0x1011	Restore default parameters	ARRAY	Unsigned32	rw
0x1014	COD-ID_EMCY	VAR	Unsigned32	

Table 8.1 Extract from object dictionary

Object	Object name	Object code	Type	Attr.
0x1017	Producer_heartbeat_time	VAR	Unsigned16	rw
0x1018	Identity_Object all 4 entries (serial number, ...) support	RECORD	Identity (23h)	ro
0x1400	1st_receive_PDO_parameter	RECORD	PDO CommPar	rw
0x1401	2nd_receive_PDO_parameter	RECORD	PDO CommPar	rw
0x1402	3rd_receive_PDO_parameter	RECORD	PDO CommPar	rw
0x1403	4rd_receive_PDO_parameter	RECORD	PDO CommPar	rw
0x1600	1st_receive_PDO_mapping max. 8 objects	RECORD	PDO Mapping (21h)	rw
0x1601	2nd_receive_PDO_mapping max. 8 objects	RECORD	PDO Mapping	rw
0x1602	3rd_receive_PDO_mapping max. 8 objects	RECORD	PDO Mapping	rw
0x1603	4th_receive_PDO_mapping max. 8 objects	RECORD	PDO Mapping	rw
0x1800	1st_transmit_PDO_parameter	RECORD	PDO CommPar (20h)	rw
0x1801	2nd_transmit_PDO_parameter	RECORD	PDO CommPar (20h)	rw
0x1802	3rd_transmit_PDO_parameter	RECORD	PDO CommPar	rw
0x1803	4th_transmit_PDO_parameter	RECORD	PDO CommPar	rw
0x1A00	1st_transmit_PDO_mapping max. 8 objects	RECORD	PDO Mapping	rw
0x1A01	2nd_transmit_PDO_mapping max. 8 objects	RECORD	PDO Mapping	rw
0x1A02	3rd_transmit_PDO_mapping max. 8 objects	RECORD	PDO Mapping	rw
0x1A03	3rd_transmit_PDO_mapping max. 8 objects	RECORD	PDO Mapping	rw

Table 8.1 Extract from object dictionary

These objects make CANopen communication flexible. You will find a description of the individual functions in the following.

8.2 Parameter channel (service data objects)

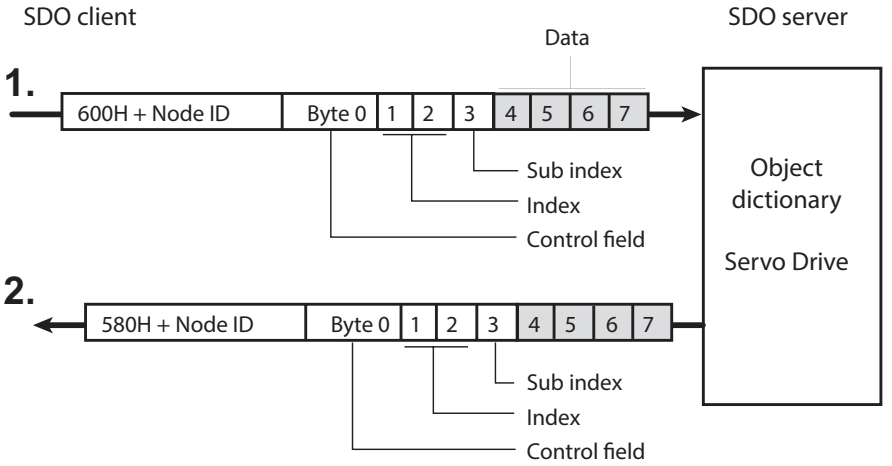
Service data objects (SDOs):

- Make possible read and write access to the object dictionary for the Servo Drive.
- Utilise the "multiplexed domain transfer protocol" in the CAL specification. The "multiplexed domain transfer protocol" transfers data of any length.

The Servo Drive uses an integrated SDO server for the SDO transfer and communications via 2 reserved "identifiers".

1. Receive SDO: **600H**
2. Transmit SDO: **580H**

Example SDO data transfer in the "expedited mode":



The CAL specification differentiates between 3 types of protocol services:

1. Download protocol (write)
2. Upload protocol (read)
3. Abort protocol (error)

The download and upload protocols differentiate between the:

- "Expedited multiplexed domain protocol" (access to objects with a data length of up to 4 bytes) and the
- "Multiplexed domain protocol" (access to objects with any data length).

Entries in the "control field" (see figure above) are generated by the CANopen driver. The data transferred generate the entries (you will find more information on the "control field" in the CIA profile CiA301).

8.2.1 Parameter data types

NOTE:

– Some parameter settings are indicated as value substitution text by the software "Moog DriveAdministrator".
Example: Parameter **P 0045 0-MOT_Type = PSM**

– For the communication via the field bus system, use the corresponding numerical values for the value substitution text. The software Moog DriveAdministrator displays the numerical values in brackets ( ).
Example: Parameter **P 0045 0-MOT_Type = PSM (1)**

Parameter data types supported by the drive controllers:

Parameter data type	Value range	Description
USIGN8	0 ... 255	
USIGN16	0 ... 65 535	
USIGN32	0 ... 4 294 967 295	
INT8	-128 ... 127	
INT16	-32 768 ... 32 767	
INT32	-2 147 483 648 ... 2 147 483 647	
FLOAT32	-3.4 E^{38} ... 3.4 E^{38}	See IEEE
STRING		

Table 8.2 Parameter data types

8.2.2 Data types in the control protocol

The SDO protocol shows the data types:

- With the correct sign.
- As 32-bit values.
- In the Intel format.

Data byte in the SDO protocol	3	4	5	6
USIGN8 / INT8* USIGN16 / INT16* USIGN32 / INT32	Low word Low byte	Low word High byte	High word Low byte	High word High byte
FLOAT32	IEEE format			
STRING				
* Filled with correct sign (00H and FFH)				

Table 8.3 Depiction of the data types in the SDO protocol

8.2.3 Device parameters

Use parameter numbers to address device parameters.

Along with the standardised objects (according to CiA301 and CiA402) the CANopen profile has an area defined for manufacturer-specific entries (device parameters).
This area is between **2000 h and 5FFF h** (CiA301). For instance to read or write the parameter **P 0455 MOT_FNOM** (rated frequency of the motor):

- Form object index from **2000 h + parameter number (hex).**
  2000 h + 1C7 h (0455 hex) = 21C7 h

NOTE:

CiA301 objects (**1000h ... 1FFFh**) and CiA402 (**6000h ... 9FFFh**):

- Can also be addressed by the master using the object numbers, even if they are displayed in Moog DriveAdministrator using device parameter numbers.
- Should not be addressed using the manufacturer-specific objects (not recommended).

Example:

The object **1000h** – Device Type exists in the CiA301 profile and as a device parameter with the parameter number **2011**. Via CANopen (or EtherCAT) it can be accessed in 2 ways simultaneously. To make the read/write access unambiguous, address this object using the profile-specific object number **1000h** (as per CiA301).

8.2.4 Example:
Read access to the device parameter of data type "string", parameter 3 DV_DeviceAliasName

The following table shows:

- The **parameter 3 DV_DeviceAliasName** for the MSD Servo Drive.
- The string to be transmitted "X-Axis".
- The hexadecimal entries for the numerical values.

TIME	ID	Direction	DLC	byte								Comment
				0	1	2	3	4	5	6	7	
18.992445	Tx	601	8	40	03	20	00	00	00	00	00	Read object 2003h (= parameter 3)
18.992972	Rx	581	8	41	03	20	00	64	00	00	00	Response: transmit 64h --> 100 bytes
35.514341	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 1
35.514594	Rx	581	8	**00**	**58**	**2d**	**41**	**78**	**69**	**73**	00	Response segment 1 - contains "X-Axis"
36.269620	Tx	601	8	70	00	00	00	00	00	00	00	Request segment 2
36.270175	Rx	581	8	10	00	00	00	00	00	00	00	Response segment 2
36.982385	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 3
36.982664	Rx	581	8	00	00	00	00	00	00	00	00	Response segment 3
37.686447	Tx	601	8	70	00	00	00	00	00	00	00	Request segment 4
37.686706	Rx	581	8	10	00	00	00	00	00	00	00	Response segment 4
38.421344	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 5
38.421604	Rx	581	8	00	00	00	00	00	00	00	00	Response segment 5
39.053526	Tx	601	8	70	00	00	00	00	00	00	00	Request segment 6
39.053787	Rx	581	8	10	00	00	00	00	00	00	00	Response segment 6
39.749081	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 7
39.749347	Rx	581	8	00	00	00	00	00	00	00	00	Response segment 7
40.428981	Tx	601	8	70	00	00	00	00	00	00	00	Request segment 8
40.429249	Rx	581	8	10	00	00	00	00	00	00	00	Response segment 8
41.085839	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 9
41.086198	Rx	581	8	00	00	00	00	00	00	00	00	Response segment 9
41.740755	Tx	601	8	70	00	00	00	00	00	00	00	Request segment 10
41.741148	Rx	581	8	10	00	00	00	00	00	00	00	Response segment 10
42.514034	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 11
42.514294	Rx	581	8	00	00	00	00	00	00	00	00	Response segment 11
43.172512	Tx	601	8	70	00	00	00	00	00	00	00	Request segment 12
43.172787	Rx	581	8	10	00	00	00	00	00	00	00	Response segment 12
43.908571	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 13
43.908831	Rx	581	8	00	00	00	00	00	00	00	00	Response segment 13
44.668466	Tx	601	8	70	00	00	00	00	00	00	00	Request segment 14
44.668740	Rx	581	8	10	00	00	00	00	00	00	00	Response segment 14
53.884044	Tx	601	8	60	00	00	00	00	00	00	00	Request segment 15
53.884414	Rx	581	8	0b	00	00	00	00	00	00	00	Response segment 15 - No further segments

Explanation of the ASCII values transmitted:

- The 1st segment of the parameter transmits the 6-byte string "X-Axis".
- The subsequent segments of the parameter contain "zeroes".

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Bytes transmitted (Hex)	**58**	**2d**	**41**	**78**	**69**	**73**
Interpretation (ASCII)	**X**	**–**	**A**	**x**	**i**	**s**

8.2.5 Parameter set download using CANopen

There are 2 ways of transferring a parameter from the CANopen master to the MSD Servo Drive:

1. As an SDO transfer (object-wise).
2. As a data set download (block without logic check).

On each transmission of a parameter, the Servo Drive checks whether the parameter matches its existing data set. The check on the new parameter values also covers existing parameter values (plausibility check or logic check) for parameters that are functionally related.

If the Servo Drive rejects a parameter from a valid data set on object-wise SDO transfer, a data set download without logic check may rectify the problem.

Transmitting parameter data set via CANopen without logic check to the Servo Drive:

Step	Action
1.	Notify download without logic check, for this purpose • Write the value 1 to parameter **11[4] PARA_SetCmd** Register upload (logic check is deactivated).
2.	• Parameters for the data set are transmitted sequentially to the Servo Drive. Attention: Basic check mechanisms are still active, despite the disabled logic check. Example: If the value range limits are not respected by the download of a parameter, the SDO protocol is rejected (abort message).
3.	End download and activate logic check, for this purpose • After the transmission of the parameter data set, set parameter **11[4] PARA_SetCmd** Register upload to the value 0. • The logic check is activated. • In the event of an error, the user receives an "emergency message".
4.	Save the parameter data set

Warning	Incorrect operation during the parameter set download:
	• Can result in uncontrolled behaviour or irreparable damage to the drive system, in the worst case injury or death! Download the parameter data set: • With drive system stationary and switched off! • Secure against switching on for the duration of the download!

8.3 Process data (PDO)

Process data objects are used to transfer defined data via the mapping.
The "transmission types" define the type of transmission.

8.3.1 Process data (PDO) - transmission types

The CANopen profile CiA301 defines various "transmission types" for the PDO transmission. The "transmission types" and the "event control" can be set separately for all supported:

- RxPDOs (process data received from the Servo Drive) and
- TxPDOs (process data sent by the Servo Drive).

The Servo Drive supports **3** transmission types:

1. **Acyclic synchronous type no. 0 h**

   PDOs with this transmission type are not cyclic, but instead are event-controlled with reference to the "SYNC object".

   For the reception the following applies:
   – Regular RxPDO data are not expected.
   – If a RxPDO is received, it is applied after the arrival of the next "SYNC object".

   For sending (TxPDO data) the following applies:
   – If the event for sending is present, the next "SYNC object" is waited for and then the data sent (from firmware version 2.15-00).

2. **Cyclic synchronous types no. 1 h to F0 h**

   PDOs with this transmission type are cyclic, with reference to the "SYNC object" for the timing.

   For the reception the following applies:
   – A numerical value (**1 h to F0 h**) defines the number of SYNC objects transferred for which an RxPDO is expected and applied.

   For sending (TxPDO data) the following applies:
   – After applying the RxPDO, a TxPDO with a specific number (**1 h to F0 h**) is sent.

3. **Asynchronous types no. FE h to FF h**

   PDOs with this transmission type are asynchronous and event-controlled.
   The Servo Drive applies each RxPDO immediately after reception and then sends a TxPDO.

   Special aspects:
   – The SYNC object is irrelevant for this transmission type.
   – The event of type FF h is defined in the related device profile.

NOTE:
The transmission types are set via the CANopen objects:
- **1400 h for RxPDO**
- **1800 h for TxPDO**

NOTE:
Activate event control:
- Set transmission types to "asynchronous types" (FE h or FF h).

Function of the event control

Sending process data (PDO) under "event control" means the data are sent when they arise. An event to send a TxPDO is, for example, a change in a bit within the TxPDO. This means the contents of the mapped data for this TxPDO are changed.

NOTE:
PDO mapping is the representation of the application objects (real-time data) from the object dictionary in the process data objects.

NOTE:
Changes to TxPDO contents have no effect on sending other TxPDO data under event control.

Example:

- TxPDO1 contains the mapped status word 6041 h.
- TxPDO2 contains the actual current position.

Changes to the mapped status word 6041 h for TxPDO1 cannot be used as an event to send TxPDO2 data. If this action is necessary:

- Map the status word 6041 h to TxPDO2.

Selection of the events:

Each change in the bits in the TxPDO data for the MSD Servo Drive can be defined as an event:

- The default settings in the MSD Servo Drive will result in the monitoring of all bits (maximum 64 bits = 8 bytes) for changes and therefore the evaluation as an event.
- To mask the event control for individual bits, the bits have event masks.

The **parameter P 2007 (COM_301_EvMask)** contains event masks that inhibit individual bits of the TxPDO data for events. TxPDO data have sub indices. Each sub index covers 32 bits of the TxPDO data.

Division of the sub indices in the field parameter P 2007 - COM_301_ EvMask "Event mask for asynchronous transmit pdos"

Sub index	Name	Value	Description	Type
0	EvMsk_TxPdo1L	FFFFFFFFh	Event mask for txpdo 1 byte 0-3	uint32
1	EvMsk_TxPdo1H	FFFFFFFFh	Event mask for txpdo 1 byte 4-8	uint32
2	EvMsk_TxPdo2L	FFFFFFFFh	Event mask for txpdo 2 byte 0-3	uint32
3	EvMsk_TxPdo2H	FFFFFFFFh	Event mask for txpdo 2 byte 4-8	uint32
4	EvMsk_TxPdo3L	FFFFFFFFh	Event mask for txpdo 3 byte 0-3	uint32
5	EvMsk_TxPdo3H	FFFFFFFFh	Event mask for txpdo 3 byte 4-8	uint32
6	EvMsk_TxPdo4L	FFFFFFFFh	Event mask for txpdo 4 byte 0-3	uint32
7	EvMsk_TxPdo4H	FFFFFFFFh	Event mask for txpdo 4 byte 4-8	uint32

Table 8.4 Field parameter **P 2007**

Example: usage of the masks

To permit the usage of the lower 16 bits of TxPDO1 as an event, write to the sub indices for field parameter **P 2007** as follows:

- Sub index 0 (event mask TxPDO1L bytes 0 – 3) = 0000 FFFF h
- Sub index 1 (event mask TxPDO2H bytes 4 – 7) = 0000 0000 h

Minimum time for successive TxPDOs of the same type

Along with possibility of inhibiting specific content of an TxPDO as an event, the minimum time between 2 TxPDOs of the same type can be entered. This "inhibit time" can be entered per TxPDO via the objects 0x1800 (TxPDO1), 0x1801 (TxPDO2), 0x1802 (TxPDO3) and 0x1803 (TxPDO4) sub index 03 (inhibit time).

NOTE:
The unit for the inhibit time is [100 µs], where only integer multiples of 1 ms are permissible for the firmware. The smallest steps during the configuration of the inhibit time are 10, 20, 30, ... , that is in 1 ms (10*100 µs) steps.

NOTE:
To send TxPDO data cyclically, in **sub index 5 (event timer)** for the **objects**

- **0x1800 (TxPDO1)**
- **0x1801 (TxPDO2)**
- **0x1802 (TxPDO3)**
- **0x1803 (TxPDO4)**

set a cycle time in ms.

8.3.2 PDO mapping

Explanations on process data mapping:

- Process for combining process data from different data sources and data structures into a uniform data source and data structure.
- The parameters for all 4 RxPDOs and 4 TxPDOs in the Servo Drive can be mapped variably.
- The mapping is as per the definitions in the CANopen communication profile CiA301.
- The manufacturer-specific parameter area (2001 h to 5FFF h) contains a large portion of device-specific parameters. The device-specific parameters (objects):
  - Can also be mapped to the PDOs.
  - Are in the electronic data sheet – EDS file for the MSD Servo Drive.

8.3.3 Mapping notes

NOTE:
The MSD Servo Drive does not have any mapping selectors.
The PDOs do not contain any mapping with the factory setting for the MSD Servo Drive (mapping = 0). This means that before communication with PDOs takes place:

- The mapping must be written to the Servo Drive via the controller,
- Or the data set must be transmitted.
  - The MSD Servo Drive saves communication settings, e.g. mapping or transmission types in data sets. These data sets can be transferred complete and do not need to be written again.

Objects of relevance for the mapping:

RxPDOs	TxPDOs
1600 h RxPD01 mapping	1A00 h TxPD01 mapping
1601 h RxPD02 mapping	1A01 h TxPD02 mapping
1602 h RxPD03 mapping	1A02 h TxPD03 mapping
1603 h RxPD04 mapping	1A03 h TxPD04 mapping

NOTE:
Each PDO can map as a maximum:

- 8 objects.
- 8 bytes.

Table 8.5 Mapping-relevant objects

Please note! (From firmware V124.15-02 no longer necessary)

- Fill PDO with an even number of bytes!
- If there is an odd number of bytes, fill the PDO using a "dummy byte" in parameter **P 2055 "COM_301_U8" (object 0x2807)**!

8.4 Heartbeat function

8.4.1 Configuration of the error checking mechanisms

There are 2 error checking protocols for the CANopen field bus system:

1. The "heartbeat function"
2. The "node guarding function"

NOTE!
For new designs the CiA (CAN in Automation - user organisation) recommends the heartbeat function. The advantage of the heartbeat protocol is that it is not necessary to send any request telegrams (remote frames). This aspect reduces the bus load on the CAN network.

Do not use the "heartbeat" and "node guarding" functions at the same time in the Servo Drive!

8.4.2 Heartbeat

The MSD Servo Drive:

- Supports the heartbeat function according to CiA301. The heartbeat function operates as per the states and the state transitions of the network nodes (NMT state machine).
- Can be used as both **heartbeat producer** and **heartbeat consumer**. The MSD Servo Drive can be used as the **heartbeat producer** and **heartbeat consumer** at the same time.
- **Heartbeat producer:** as the heartbeat producer, the Servo Drive sends heartbeat messages to the controller at the interval defined in **object 1017 Producer Heartbeat Time**.
- **Heartbeat consumer**: as the heartbeat consumer, the Servo Drive monitors heartbeat telegrams from the controller: the node ID of the controller and the monitoring time must be entered in **object 1016 sub index 1 Consumer Heartbeat Time**. The monitoring of **one** heartbeat producer is supported by the MSD Servo Drive.

Structure of object 1016h Consumer Heartbeat Time

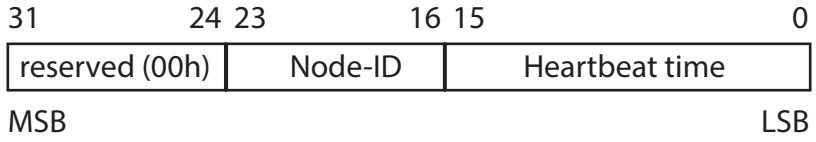


Figure 8.1 Object 1016 h Consumer Heartbeat Time

- **Object 1016 h Consumer Heartbeat Time** indicates the expected heartbeat cycle time.
- **Object 1016 h** supports the sub indices 0 and 1. I.e. only one **heartbeat producer** can be monitored.
- The **Consumer Heartbeat Time** should be greater than the corresponding **Heartbeat Producer Time**.
- The monitoring of the **heartbeat producer** starts with the reception of the first **heartbeat**.
- Before the reception of the first **heartbeat**, the status of the heartbeat producer is unknown.

Principle of operation of the heartbeat protocol

- The **heartbeat producer** (Servo Drive) sends telegrams at regular intervals (**heartbeat protocols**), for this purpose:
  - Enter the value for the intervals (in ms) in the **object 1017 h Producer Heartbeat Time**.
  - If the value for the intervals in the **object 1017 h Producer Heartbeat TIME = 0,** as in the factory setting, the heartbeat function is inactive.
- The **heartbeat consumer** expects the **heartbeat protocols** from the **heartbeat producer** at the intervals set.
- The **heartbeat protocols** start immediately after the entry of the **heartbeat producer time.**
- If the Servo Drive is switched on with the **heartbeat producer time ≠ 0**, the heartbeat protocol starts with the **NMT state transition Initialisation -> Pre-Operational** and the **boot-up message** is seen as the first **heartbeat protocol**.
- If **heartbeat protocols** do not arrive at the **heartbeat consumer** in the time set, the **heartbeat consumer** triggers an event (**heartbeat event**).

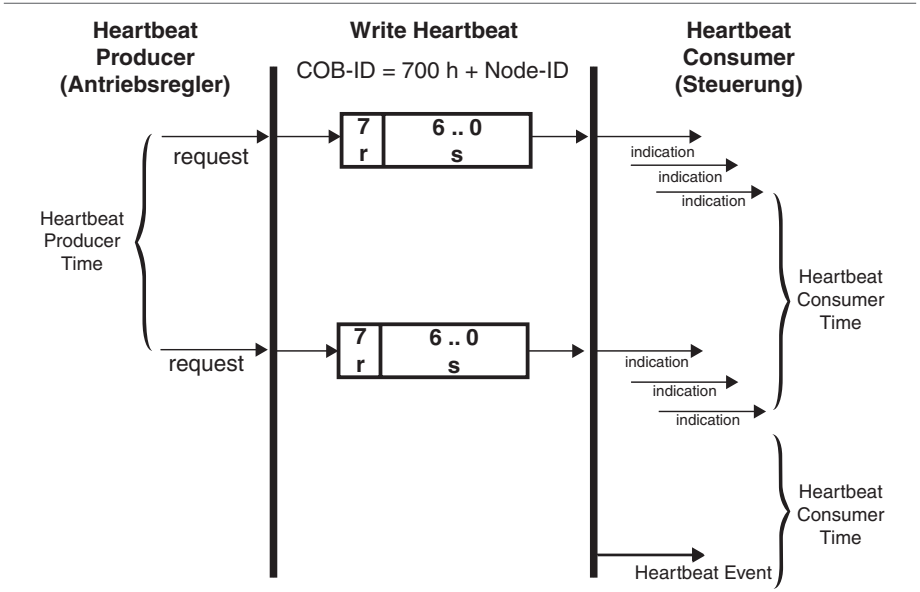


Figure 8.2 Heartbeat protocol

r:	Reserved (bit 7, within 0)
s:	Status of the heartbeat producer

Bit 0 ... 6	NMT - network state
0	Bootup
4	Stopped/Safe OP
5	Operational
127	Pre-Operational

8.5 Telegram failure monitoring

The Servo Drive has a function:

- That monitors sync telegrams or RxPDOs arriving.
- That triggers an error message if the sync telegrams or the sync telegrams fail for a period previously configured.

Parameters for the configuration of the telegram failure monitoring:

Parameter ID	Name	Description
P 2061	COM_CAN_Timeout_Type	Selection of the signal to be monitored: • 0 = Sync • 1 = RxPDO
P 2062	COM_CAN_Timeout_Value	Enter value: • Timeout [ms] • 0 = Monitoring inactive

Table 8.6 Parameters for "telegram failure monitoring"

The parameter **P 2061 COM_CAN_Timeout_Type** is used to select the signal to be monitored:

- The sync signal arriving or
- The RxPDOs.

In the parameter **P 2062 COM_CAN_Timeout_Value**:

- Enter time (in ms) from the last configured signal to the detection of a telegram failure.
- Enter the value "0" for "monitoring inactive".

9 Configuring the parameters for the devices - EtherCAT

9.1 EtherCAT functionality implemented

The illustration below shows the structure of the EtherCAT technology based on the OSI 7-layer model:

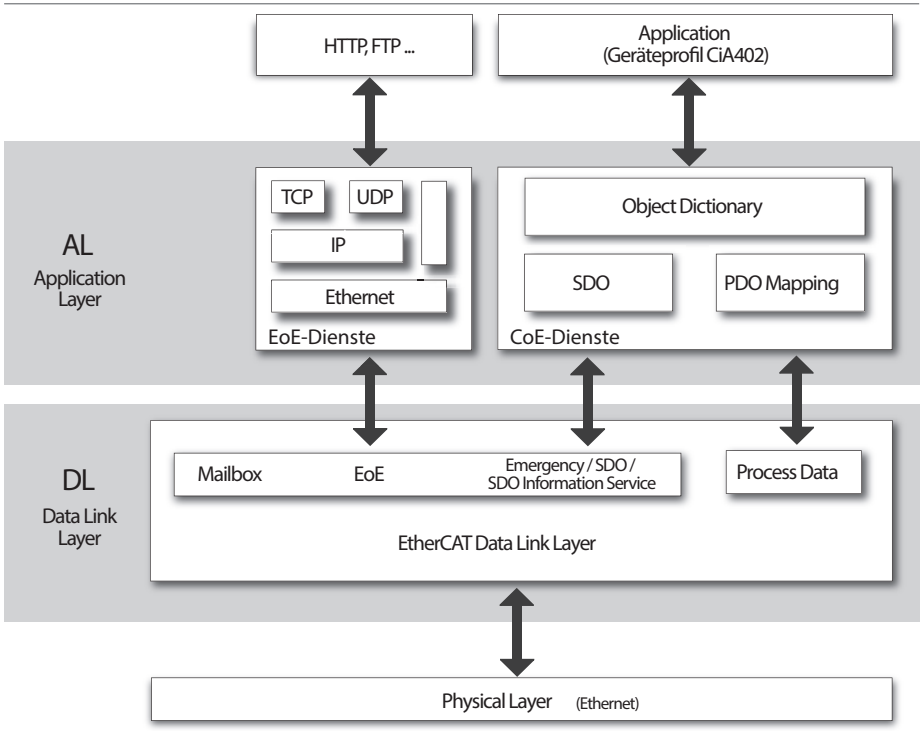


Figure 9.1 Structure of the EtherCAT technology

Physical layer

- Based on the physical aspects of IEEE802.3/100 BaseTX Ethernet.
- Defines electrical properties of the EtherCAT field bus system, such as:
  - Connectors,
  - Cable length and properties,
  - Bit coding and bit timing.

Data link layer

- Is split into "Mailbox" and "Process Data".
- Connects network users.
- Assigns priorities to data packets.
- Undertakes error monitoring and error corrections.

Mailbox

- Contains all services that do not have content that is time-critical for the process data and that do not affect process data when executed.
- Utilises the SDO channel to deliver service data for the drive parameters.
- Is the basis for EoE services and error handling (emergency telegrams).
- Implemented in the MSD Servo Drive (CoE):
  - SDO/Abort
  - Initiate SDO Download
  - Download SDO Segment
  - Initiate SDO Upload
  - Upload SDO Segment
  - Abort SDO Transfer
  - Select device parameters via object ID 2000 h + parameter number.

Emergency

- The emergency service transmits error messages.
- Unlike the CANopen field bus system (slave sends error messages independently) the EtherCAT master collects error messages.
- The MSD Servo Drive supports, as per the CiA402 device profile, "error codes". For content and structure see chapter 13.

SDO information service

- The EtherCAT master reads the object dictionary for the EtherCAT slave via the SDO information service.
- Has access to the list of objects in the MSD Servo Drive.
  The EtherCAT master receives information on objects supported by the EtherCAT slave such as:
  – Data types
  – Access options, etc.
- In the MSD Servo Drive is an alternative to the EDS file (electronic data sheet) for the CANopen field bus system.

EoE - Ethernet over EtherCAT

- Uses, e.g. the function "tunnelling of standard Ethernet frames in EtherCAT frames". Using this function EtherCAT transmits protocols such as TCP/IP.
- EoE functionality in the MSD Servo Drive:
  – Initiate EoE request.
  – Initiate EoE response.
  – EoE fragment request.
  – EoE fragment response.

Distributed clocks

- Each EtherCAT slave has a clock (distributed clock). A pulse synchronises the distributed clocks in the EtherCAT network.
- The clock in one EtherCAT slave in the EtherCAT network is used as the "reference clock".
- You can configure the "distributed clocks" via the controller. Set a multiple of the timebase (125 µs) as the cycle time!

ESI file - EtherCAT slave information

- XML file
- Is installed in the "EtherCAT master" (controller).
- Is used to connect the EtherCAT slave to the EtherCAT master (controller) in the EtherCAT network.
- Contains the configuration (mapping, etc.) for the related operation modes and is available in the "Firmware" section in the Moog download area.

Process data (CiA301)

- Cyclic transmission of mapped process data objects (PDO), such as:
  – Position values.
  – Velocity values.
  – Torque setpoints and torque actual values.
- Process data profile:
  – 4 RxPDOs.
  – 4 TxPDOs.
- Transmission length maximum 8 bytes per PDO.
- Variable mapping as per CiA301 (cf. CANopen).
- Cycle times:
  – Transmission of cyclic position setpoints with maximum 8 kHz (125 µs)
  – Transmission of cyclic velocity setpoints with maximum 8 kHz (125 µs).
  – Transmission of cyclic torque setpoints with maximum 8 kHz (125 µs).

Please note! (From firmware V124.15-02 no longer necessary)
- Fill the PDO with an even number of bytes!
- If there is an odd number of bytes, fill the PDO using a "dummy byte" in parameter **P 2055 "COM_301_U8"** (object 0x2807)!

Application layer

- Utilises communication objects (COB) to exchange data between the network users. Communication objects are an elementary component for creating CANopen applications.
- Also includes the object dictionary. The object dictionary:
  – Is based on the CANopen profile and makes it possible for the user to access objects also via EtherCAT systems.
  – Is the basis for SDO and PDO access on the Servo Drive.
- The MSD Servo Drive supports:
  – EoE services (Ethernet over EtherCAT).
  – CoE services (CAN over EtherCAT).

Application

- The MSD Servo Drive utilises the device profile CiA402

9.2 EtherCAT network management (NMT)

- Functions of the EtherCAT NMT:
  - Initialisation, error monitoring, status monitoring in the network and monitoring of the individual network users.
- Compliant with the CANopen field bus system. In the EtherCAT field bus system the "Stopped" state is replaced with "Safe-Operational".
- Depending on the functionality of the controller software, individual state transitions are undertaken automatically or via the PLC.

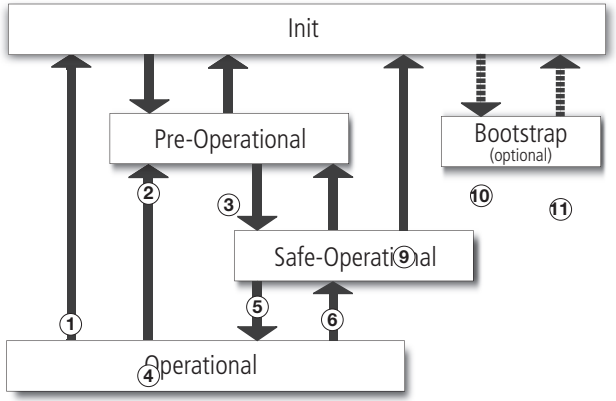


Figure 9.2 EtherCAT NMT state diagram

State	Description
Init	Initialisation, device start.
Pre-Operational	Device ready for setting the parameters. Mailbox communication functional.
Safe-Operational	PDO input data (TxPDO device) are read. PDO output data (RxPDO device) are ignored.
Operational	Cyclic I/O communication. PDO output data (RxPDO device) are processed.
Bootstrap (optional)	· Can be accessed via the "Init" state. · Update the slave firmware via the "Bootstrap" state. · The mailbox communication is active but limited to the FoE protocol (File-Access over EtherCAT).

Table 9.1 EtherCAT NMT state description

No.	State transition	Action
1	(OI)	· Stop "Output Update" · Stop "Input Update" · Stop "Mailbox Communication"
2	(IP)	· Start "Mailbox Communication"
3	(PI)	· Stop "Mailbox Communication"
4	(OP)	· Stop "Output Update" · Stop "Input Update"
5	(PS)	· Start "Input Update"
6	(SP)	· Stop "Input Update"
7	(SO)	· Start "Output Update"
8	(OS)	· Stop "Output Update"
9	(SI)	· Stop "Input Update" · Stop "Mailbox Communication"
10	(IB)	· Start "Bootstrap Mode"
11	(BI)	· Restart Device

Table 9.2 Description of the EtherCAT NMT state transitions

10 CiA402 functionality implemented in the Servo Drive

The Servo Drive supports operation modes and their functions according to the CiA402 device profile.

The operation modes according to CiA402 implemented in the MSD Servo Drive:

No.	Operation modes with profile generation in the Servo Drive	
1	**Profile position mode**	**Page 73**
2	**Velocity mode (U/F operation)**	**Page 76**
3	**Profile velocity mode**	**Page 77**
4	**Reserved**	**-**
5	**Reserved**	**-**
6	**Homing mode**	**Page 78**

No.	Operation modes with profile generation in the controller	
7	**Interpolated position mode**	**Page 80**
8	**Cyclic synchronous position mode (only EtherCAT)**	**Page 81**
9	**Cyclic synchronous velocity mode (only EtherCAT)**	**Page 82**
10	**Cyclic synchronous torque mode (only EtherCAT)**	**Page 83**

10.1 Device control and state machine

The Servo Drive is controlled via the DRIVECOM state machine defined in CiA402.

10.1.1 General information

The DEVICE CONTROL FUNCTION monitors all functions of the Servo Drive and is divided into:

- Operation of the state machine **(State machine)**
- Operation mode-specific functions **(Operation mode)**

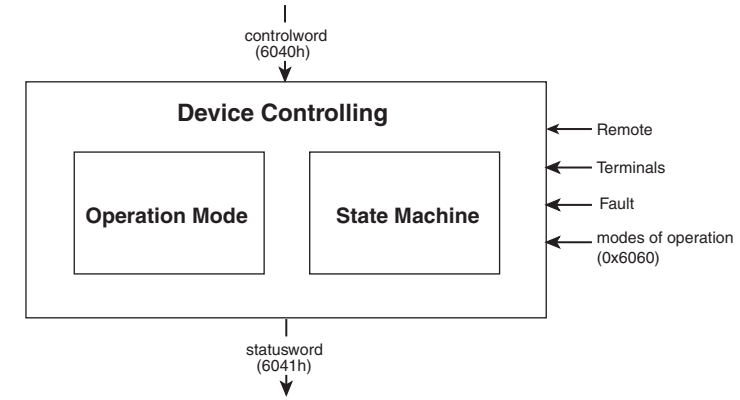


Figure 10.1 Device controlling

The state of the Servo Drive:

- Is controlled by the control word.
- Is indicated by the status word.

Process data objects (PDO) and service data objects (SDO) in the network control the Servo Drive.

The state machine

- Is controlled by the control word and
- Influenced by internal events, e.g. errors.

10.1.2 State machine

The state machine describes:

- The drive status.
- The control options for the master (user commands).
- Internal error states.

The specific status of the Servo Drive

- Indicates specific internal or external behaviour and
- At the same time limits the control commands.

Example:
Point-to-point positioning is only possible in the **"Operation enabled"** state.

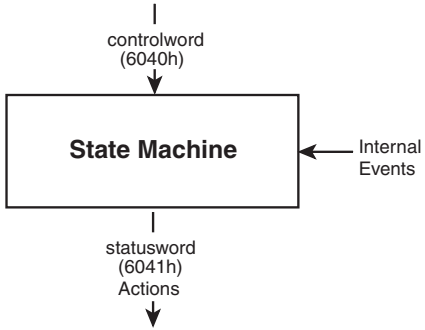


Figure 10.2 State machine

The state machine passes through transitions and states (see "State machine schematic"). The transitions and states are dependent on:

- The device profile setting in the Servo Drive (e.g. control word).
- The bus system used.

The current state is indicated in the status word. During operation the Servo Drive differentiates between:

- Standstill.
- Operation.
- Fault states.

10.1.3 DRIVE COM state machine schematic

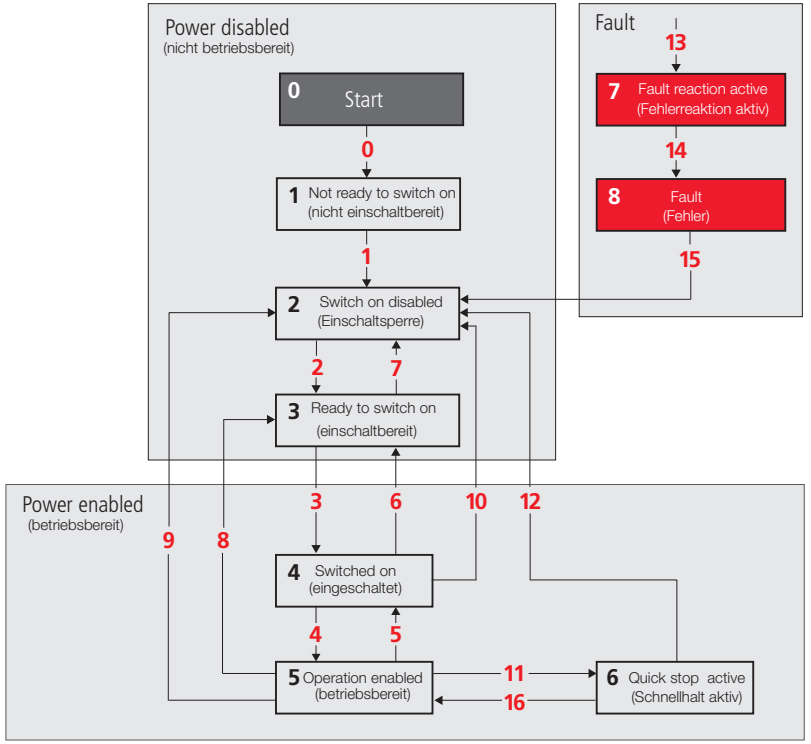


Figure 10.3 State machine schematic

0 to 8	State identifiers
1 to 16	State transitions

No.	State identifiers	Device state
0	**Start**	• Initialisation on device startup
1	**Not ready to switch on**	• Control voltage (24 V) is present at the Servo Drive. • The Servo Drive is initialised or self-test is in progress. • If fitted, the brake is applied. • Drive function inactive.
2	**Switch on disabled**	• Initialisation of the Servo Drive is completed. • Drive controller parameters setup and can be changed. • No electrical power is present in the DC link (230 V, e.g. for safety). • Drive function is deactivated. • STO (Safe Torque Off) and ENPO (Enable Power) inactive.
3	**Ready to switch on**	• DC link is electrically live. • Drive controller parameters can be changed. • Drive function is deactivated.
4	**Switched on**	• DC link is electrically live. • Drive controller parameters can be changed. • Power amplifier is ready. • Drive function is deactivated.
5	**Operation enabled**	• The Servo Drive has not detected any errors. • Drive function active and power supply present at motor. • Drive controller parameters can be changed. (Relates to default application for the Servo Drive)
6	**Quick stop active**	• Drive controller parameters can be changed. • Reaction is implemented as per quick stop option code • Drive function active and power supply present at motor. • If the "Quick stop option code" is set to "5" in the "Quick stop active" state, use the command "Enable operation" to change to the "Operation enabled" state.
7	**Fault reaction active**	• Drive controller parameters can be changed. • Fault has occurred in the Servo Drive. • The fault reaction as per the fault reaction code is implemented • Drive function active and power supply present at motor.
8	**Fault**	• Drive controller parameters can be changed. • Fault has occurred in the Servo Drive. • The removal of the supply of electrical power is dependent on the application. • Drive function is deactivated.

You will find more detailed information on the indication of the Servo Drive device state in chap. 4.7 and in chap. 5.6.

No.	State transitions	Description
0	**Start --> Not ready to switch on**	Situation: Reset.
		Action: Initialisation on device startup and/or self-test.
1	**Not ready to switch on --> Switch on disabled**	Situation: The Servo Drive has tested itself and/or initialised.
		Action: Activate communication.
2	**Switch on disabled --> Ready to switch on**	Situation: "Shutdown" command received from the host.
		Action: None.
3	**Ready to switch on --> Switched on**	Situation: "Switch on" command received from the host.
		Action: The power section is switched on.
4	**Switched on --> Operation enabled**	Situation: "Enable operation" command received from the host.
		Action: Drive function is active.
5	**Operation enabled --> Switched on**	Situation: "Disable operation" command received from the host.
		Action: Drive operation is deactivated.
6	**Switched on --> Ready to switch on**	Situation: "Shutdown" command received from the host.
		Action: The power section is switched off.
7	**Ready to switch --> Switch on disabled**	Situation: "Quick stop" and "Disable voltage" command received from the host.
		Action: None.
8	**Operation enabled --> Ready to switch on**	Situation: "Shutdown" command received from the host.
		Action: The power section is shut down directly, the unbraked motor can rotate freely.

No.	State transitions	Description
9	**Operation enabled --> Switch on disabled**	Situation: "Disable voltage" command received from the host.
		Action: The power section is shut down directly, the unbraked motor can rotate freely.
10	**Switched on --> Switch on disabled**	Situation: "Quick stop" or "Disable voltage" command received from the host.
		Action: The power section is shut down directly, the unbraked motor can rotate freely.
11	**Operation enabled --> Quick stop active**	Situation: "Quick stop" command received from the host.
		Action: "Quick stop" function is active.
12	**Quick stop active --> Switch on disabled**	Situation: "Quick stop" command is complete or "Disable voltage" received from the host. This state transition occurs (standstill in the "Quick stop active" state) if the "Quick stop option code" is set to "5".
		Action: The power section is switched off.
13	**All states --> Fault reaction active**	Situation: "Fault" has occurred in the Servo Drive.
		Action: Implement corresponding fault reaction.
14	**Fault reaction active --> Fault**	Situation: Fault reaction complete.
		Action: The Servo Drive is deactivated. The power section may be switched off.
15	**Fault --> Switch on disabled**	Situation: "Fault reset" command received from the host.
		Action: The Servo Drive runs through the "Fault reset" until the fault is rectified. When the "Fault" state is left, the host deletes the "Fault Reset" bit.
16	**Quick stop active --> Operation enabled**	Situation: "Enable operation" command received from the host. This state transition takes place if the "Quick stop option code" is set to 5, 6, 7 or 8 (see chapter 10.2).
		Action: The drive function is active.

10.1.4 Bit combinations for the state machine in the status word 6041h

No.	Device state	Status word 6041h						
		6	5	4	3	2	1	0
1	**Not ready to switch on**	0	4	X	0	0	0	0
2	**Switch on disabled**	1	4	X	0	0	0	0
3	**Ready to switch on**	0	1	X	0	0	0	1
4	**Switched on**	0	1	X	0	0	1	1
5	**Operation enabled**	0	1	X	0	1	1	1
6	**Quick stop active**	0	0	X	0	1	1	1
7	**Fault reaction active**	0	4	X	1	1	1	1
8	**Fault**	0	4	X	1	0	0	0

Table 10.1 Bit combinations in the device states 4 ... bit is irrelevant for this state

10.1.5 Bit combinations for the control commands for the state transitions - control word 6040h

The following bit combinations (bits of the control word 0-3 and 7) form the device control commands for the state transitions:

Device control command for the state transition	Bits in the control word 6040h					State transitions (see Fig. "State machine schematic" on page 60)
	7 Fault reset	3 Enable operation	2 Quick Stop	1 Enable voltage	0 Switch on	
Stop	0	X	1	1	0	2, 6, 8
Switch on	0	X	1	1	1	3
Inhibit power	0	X	X	0	1	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Reset fault	↑	X	X	X	X	15

Table 10.2 Bit combinations for state transitions X ... bit is irrelevant for this state

10.2 Option codes

"Option codes" are available to configure the behaviour for the following events:

Object	Object name / Settings supported	Event	Object code	Type	Access
605Ah	**Quick stop option code** (Quick stop function - trigger quick stop) 0: Disable drive function 1: Slow down on slow down ramp 2: Slow down on quick stop ramp 3: Slow down on the current limit 4: Slow down on the voltage limit 5: Slow down on slow down ramp and stay in quick stop 6: Slow down on quick stop ramp and stay in quick stop 7: Slow down on the current limit and stay in quick stop 8: Slow down on the voltage limit and stay in quick stop	Quick stop	VAR	Int16	rw
0x605B	**Shutdown option code** (Shutdown function - controlled run-down) -1: Reaction as per quick stop option code 0: Disable drive function 1: Slow down with slow down ramp; disable the drive	Shutdown	VAR	Int16	rw
605Ch	**Disable operation option code** (Disable control function - uncontrolled coasting) 0: Disable drive function 1: Slow down with slow down ramp and then disabling of the drive function	Disable operation	VAR	Int16	rw
605Dh	**Halt option code** (Halt function - stop the movement in progress) 0: Disable drive, motor is free to rotate 1: Slow down on slow down ramp 2: Slow down on quick stop ramp 3: Slow down on the current limit 4: Slow down on the voltage limit	Halt	VAR	Int16	rw

Table 10.3 Option codes

Object	Object name / Settings supported	Event	Object code	Type	Access
605Eh	**Fault reaction option code** (Fault reaction function - stop in the event of a fault) -2: EXT_TO: External reaction, disable drive at time out (**P0154**) -1: EXT_TO_ROTO: External reaction, disable drive at time out or motor standstill 0: Disable drive, motor is free to rotate 1: Slow down on slow down ramp 2: Slow down on quick stop ramp 3: Slow down on the current limit 4: Slow down on the voltage limit	Fault	VAR	Int16	rw
P0154: MPRO_DRVCOM_ROTO_Time					

Table 10.3 Option codes

Note:
The quick stop ramp is implemented with the smoothing set for the movement profile ramps. The fault stop ramp is always implemented without smoothing, even if smoothing is programmed.

10.3 Units and scaling, factor group

The PC user software **Moog DriveAdministrator 5** has a scaling wizard.
Using the scaling wizard it is possible:

- To configure user units.
- To enter them with their mechanical and electrical relationship.
- To display them in the form of the parameters of the **CiA402 factor group**.

Note:
The Servo Drive calculates the following parameters internally:

- Position factor.
- Velocity encoder factor.
- Acceleration factor.

Parameters (e.g. feed constant, gear ratio) are saved in the formulae for the basis for the calculation.

There are 2 way of changing application variables:

1. In **Moog DriveAdministrator 5**.
2. Via the **field bus system**.

Note:
Modified scaling parameters are applied on the next control initialisation (restarting the Servo Drive, start of control) and the scaling recalculated.

Note:
You will find practical examples for scaling in the "MSD Servo Drive - Device Help".

10.3.1 Factor group objects as per CiA402:

Object	Object name	Object code	Type	Access
0x607E	**Polarity**	VAR	Unsigned8	rw
0x6089	**Position notation index**	VAR	Integer8	rw
0x608A	**Position dimension index** (indication for scaling block)	VAR	Unsigned8	rw
0x608B	**Velocity notation index**	VAR	Integer8	rw
0x608C	**Velocity dimension index** (indication for scaling block)	VAR	Unsigned8	rw
0x608D	**Acceleration notation index**	VAR	Integer8	rw
0x608E	**Acceleration dimension index** (indication for scaling block)	VAR	Unsigned8	rw
0x608F	**Position encoder resolution**	VAR	Unsigned8	rw
0x6090	**Velocity encoder resolution**	ARRAY	Unsigned32	rw
0x6091	**Gear ratio**	ARRAY	Unsigned32	rw
0x6092	**Feed constant**	ARRAY	Unsigned32	rw
0x6093	**Position factor**	ARRAY	Unsigned32	rw
0x6094	**Velocity encoder factor**	ARRAY	Unsigned32	rw

Table 10.4 Factor group objects

Object	Object name	Object code	Type	Access
0x6097	**Acceleration factor**	ARRAY	Unsigned32	rw

Table 10.4 Factor group objects

The user can calculate and enter the factor group objects independent of the scaling wizard in **Moog DriveAdministrator**. For this purpose make the related encoder settings.

10.3.2 Calculation of the most important factor group parameters

Object 608Fh: Position encoder resolution

Defines the ratio of rotary encoder increments to the motor revolutions.

$$\text{Position encoder resolution} = \frac{\text{Encoder increments}}{\text{Motor revolutions}}$$

Object 6090h: Velocity encoder resolution

Defines the ratio of rotary encoder increments per second to the motor revolutions per second.

$$\text{Velocity encoder resolution} = \frac{\text{Encoder } \frac{\text{Increments}}{\text{Second}}}{\text{Motor } \frac{\text{Revolutions}}{\text{Second}}}$$

Object 6091h: Gear ratio

Defines the gear ratio of a gearbox on the motor.

$$\text{Gear ratio} = \frac{\text{Revolutions of the motor shaft}}{\text{Revolutions of the drive shaft}}$$

Object 6092h: Feed constant

Defines the ratio of the feed in position units to the revolutions of the drive shaft (take into account gearbox, if fitted).

$$\text{Feed constant} = \frac{\text{Feed}}{\text{Revolutions of the drive shaft}}$$

Object 6093h: Position factor

Converts the required position (in position units) into the internal format (in increments).

$$\text{Position factor} = \frac{\text{Position encoder resolution} \bullet \text{Gear factor}}{\text{Feed constant}}$$

Object 6094h: Velocity encoder factor

Converts the required velocity (in velocity units) into the internal format (in increments).

$$\text{Velocity encoder factor} = \frac{\text{Velocity encoder resolution} \bullet \text{Gear factor} \bullet \text{Position unit} \bullet F_{\text{velocity (Notation index)}}}{\text{Feed constant} \bullet \text{Velocity unit} \bullet \text{Sekond} \bullet F_{\text{Positon (Notation index)}}}$$

An example for $F_{\text{velocity (Notation index)}}$ or $F_{\text{Positon (Notation index)}}$ would be 10^2 or 10^{-6}

Object 6097h: Acceleration factor

Converts the acceleration (in acceleration units) into the internal format (in increments/s).

$$\text{Acceleration factor} = \frac{\text{Velocity unit} \bullet \text{Velocity encoder factor}}{\text{Acceleration unit} \bullet \text{Second}}$$

Object 607Eh: Polarity

Depending on the value of the Polarity flag, multiply by 1 or -1:
- Position actual value and setpoint.
- Velocity actual value and setpoint.

Pay attention to usage of the Polarity parameter as per CiA402 V2.0!

Bits 0 to 5 = Reserved (don't use)
Bit 6 = Velocity polarity
Bit 7 = Position polarity

NOTE:
Changes to the parameters for the factor group:
- Make changes with power stage deactivated!
- Activate changes by restarting the Servo Drive (control initialisation)!

10.4 I/O image

The following objects for the Servo Drive indicate the status of the inputs and outputs and are implemented.

10.4.1 Object 60FDh – digital inputs

The **object 60FDh**:
- Is implemented so it is compliant with the device profile CiA402.
- Makes it possible to evaluate functions for the digital inputs defined in the device profile.
- Is a **function-related input image** and not an image of the physical inputs present. As such it is irrelevant, for example, to which input a limit switch is connected. The bit that describes the state of the limit switch is defined within the object.

Bit	Assignment
0	Negative limit switch
1	Positive limit switch
2	Homing switch
18	Status request – safe stop
19	ENPO (enable power)

Table 10.5 Object 60FDh – digital inputs

10.4.2 Object 2079h – MPRO_INPUT_STATE

The **object 2079h**:

- Is manufacturer-specific.
- Provides an image of the digital inputs on the Servo Drive.
- Supports process data

Bit	Assignment
0	State input ENPO (enable power)
1	State input ISD00
2	State input ISD01
3	State input ISD02
4	State input ISD03
5	State input ISD04
6	State input ISD05
7	State input ISDSH
8 to 15	Not used
16	State input ISD06
17	Not used
18	State input ISA00
19	State input ISA01
30 to 31	Not used

Table 10.6 Object 2079h – MPRO_INPUT_STATE

10.4.3 Object 208Fh – MPRO_OUTPUT_STATE

The **object 208Fh**:

- Is manufacturer-specific.
- Provides an image of the digital outputs on the Servo Drive.
- Supports process data

Bit	Assignment
0	State output OSD00
1	State output OSD01
2	State output OSD02
3 to 5	Not used
6	State output motor brake
7	State relay output
8 to 14	Not used
15	State relay output "STO (Safe Torque Off)"

Table 10.7 Object 208Fh – MPRO_OUTPUT_STATE

10.4.4 Setting digital outputs via field bus

Prepare the 3 digital outputs (OSD00, OSD01, OSD02 plus RELOUT1) for access via field bus:

- Configure output selectors **"MPRO_Output_FS_xxx"** (parameters **P 0122** to **P 0124**, **P 0126**):

Setting	Description
(39) Output set via communication option in 1 ms cycle	Set the output via communication option, update in 1 ms cycle
(40) Output set via communication option in NC cycle	Set the output via communication option, update in control cycle (62.5 µs)

Table 10.8 Parameter setting for the object - **MPRO_Output_FS_OSDxx** (**P 0122** *to* **P 0124, P 0126**)

10.4.5 Object 60FE - digital outputs

Set the outputs via the manufacturer-specific **object 60FE** (function selector for digital outputs = **COM_NC (39) or COM_NC (40)**):

Bit	Assignment
16	OSD00
17	OSD01
18	OSD02
23	RELOUT1
22	Motor brake (OSD03)

Table 10.9 Object 60FE - digital outputs

11 Operation of the MSD Servo Drive via CiA402

11.1 Parameter configuration

For operation via the CANopen and EtherCAT field bus systems (as per CiA402 device profile), set the following parameters:

Parameter no.	Name	Function	Setting
P 0159	MPRO_CTRL_SEL	Control location selector	CiA402
P 0165	MPRO_REF_SEL	Setpoint selector	CiA402

Table 11.1 Configuring MSD Servo Drive parameters

You will find these parameters in "Basic settings" in the "Motion profile" dialog box.

11.2 Interpolation of the setpoints

During the cycle time the Servo Drive expects setpoints from a higher lever controller in the parameter **CON_IpRefTs**.

Adjust the cycle time of the Servo Drive (interpolate), if in the operation mode used:

- The internal profile generator is inactive and
- Setpoints are to be transmitted cyclically.

This is the case, e.g., in the **Cyclic synchronous position mode**.

Adjust cycle time in device parameter **P 0306[0]**:

Parameter no.	Name	Function
P 0306	CON_IpRefTs	Cycle time for the setpoints in the IP mode (interpolation mode)

Table 11.2 Configuring MSD Servo Drive parameters

11.3 Control word

11.3.1 Object 6040h control word

The object 6040h control word is represented in the parameter **P 2208[0] MPRO_402_Controlword**. The control word contains bits for:

- State control.
- Controlling the operation modes.
- Manufacturer-specific options.

The bits in the control word are defined as follows:

Bit	Description	M/O	Bit significance
0	Switch on	M	LSB
1	Enable voltage	M	
2	Quick stop	M	
3	Enable operation	M	
4	Operation mode specific	O	
6			
7	Fault reset	M	
8	Halt	O	
9	Reserved	O	
10			
11	Manufacturer specific	O	
15			MSB

Table 11.3 Bits in the control word, O ... Optional, M ... Mandatory

Bits 0 - 3 and 7:

Trigger DEVICE CONTROL COMMANDS using the following schematic in the control word:

Command	Bits in the control word					Transitions
	Bit 7 Fault reset	Bit 3 Enable operation	Bit 2 Quick Stop	Bit 1 Enable voltage	Bit 0 Switch on	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	↑	X	X	X	X	15
NOTE: Set the bits in the control word one after the other! Bits marked with X are irrelevant * ... In the "SWITCH ON" state the drive executes the function for the state. ** ... In the "SWITCH ON" state the drive does not execute the function for the state.						

Table 11.4 Device control commands

Bits 4 - 6 and 8

Bits 4 - 6 and 8 have different functions depending on the active operation mode ("Modes of operation display"):

Operation mode	Bit			
	4	5	6	8
Profile position mode	New setpoint	Change set immediately	Absolute/ relative	Halt

Table 11.5 Operation mode-specific bits in the control word

Operation mode	Bit			
	4	5	6	8
Profile velocity mode	Reserved	Reserved	Reserved	Halt
Homing mode	Homing operation start	Reserved	Reserved	Halt
Interpolated position mode	Enable IP mode	Reserved	Reserved	Halt
Cyclic synchronous position mode	Reserved	Reserved	Reserved	Reserved
Cyclic synchronous velocity mode	Reserved	Reserved	Reserved	Reserved
Cyclic synchronous torque mode	Reserved	Reserved	Reserved	Reserved

Table 11.5 Operation mode-specific bits in the control word

Bits 7 and 11 - 15

Bit	Name	Value	Description
7	Clear fault	0 ⇨ 1	Clear fault
11			No function
. . .	No function		No function
15			No function

Table 11.6 Operation mode-specific bits in the control word

11.3.2 Status word

Object 6041h status word

The object 6041h status word is represented in the parameter **P 2209[0] MPRO_402_ Statusword**. The status word indicates the actual status of the Servo Drive.
It contains the bits for:

- The actual device state.
- The state of the operation mode.
- The states of the manufacturer-specific functions.

Bits in the status word

Bit	Description	M/O
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	Fault	M
4	Voltage enabled	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacturer specific	O
9	Remote	M
10	Target reached	M
11	Internal limit active	M
12 - 13	Operation mode specific	O
14 - 15	Manufacturer specific	O

Table 11.7 Bits in the status word, O ... Optional, M ... Mandatory

Bits 0 - 3, 5 and 6:

The bits 0 - 3, 5 and 6 indicate the STATUS of the Servo Drive.

Value (binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1x1 0000	Switch on disabled
xxxx xxxx x011 0001	Ready to switch on
xxxx xxxx x011 0011	Switched on
xxxx xxxx x011 0111	Operation enabled
xxxx xxxx x001 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Table 11.8 Bits for the device state "status word"

Bit 4: Voltage enabled

Power supply is present.

Bit 5: Quick Stop

In the LOW state, bit 5 indicates that the drive is undertaking a "quick stop". If the drive is ready to operate, bits 0, 1 and 2 of the "status word" are set to 1. The other bits indicate the states of the drive, e.g. undertake "quick stop".

In the event of a fault the FAULT bit is set (1).

Bit 7: Warning

Bit 7 indicates warnings, e.g. temperature limits. The device state does not change if there
are warnings. You will find more detailed information on the existing warning in the error code. The definition of the warnings/warning levels can be set in Moog DriveAdministrator 5 in the following parameters:

- **P 34[0]** = Actual status word device warning
- **P 33[0-8]** = Actual fault/actual warning = message in MDA5

Bit 8: Manufacturer-specific

Not currently used.

Bit 9: Remote

Bit 9 indicates that the control location selector (parameter **P 0159 MPRO_CTRL_SEL**) is configured for CiA402 CANopen/EtherCAT (CiA402(5)) and the control word (object 6040h) is processed.

Bit 10: Target reached

Bit 10 is set automatically:

- If a SETPOINT is reached. The setpoint is dependent on the operation mode. Changes to the setpoint by the master change bit 10.
- On "quick stop" (OPTION CODE 5, 6, 7 or 8) after termination of the "quick stop".
- On STOP request at standstill.

Bit 11: Internal limit active

Bit 11 is:

- Set if internal limits are reached.
- Dependent on the operation mode.

Bit 12 and 13:

Bits 12 and 13 are dependent on the operation mode (see next chapter "Operation mode CiA402").
Overview:

Operation mode	Bit	
	12	13
Profile position mode	Setpoint acknowledge	Following error
Profile velocity mode	Speed	Max slippage error
Homing mode	Homing attained	Homing error
Interpolated position mode	IP mode active	Reserved
Cyclic synchronous position mode	Target position ignored	Following error
Cyclic synchronous velocity mode	Target velocity ignored	Reserved
Cyclic synchronous torque mode	Target torque ignored	Reserved

Table 11.9 Operation mode-specific bits in the control word

Bit 14 and 15:

Bits 14 and 15 are manufacturer-specific. You will find explanations in the related operation mode in the chapter "Operation modes CiA402".

12 Operation modes

12.1 CiA402 compatible operation modes

The operation mode is switched over in a Servo Drive using the object 0x6060 Modes of operation and can be undertaken in the state "Operation enabled" (power applied to the motor).
The object 6061 h Modes of operation display indicates the actual operation mode.

The following operation modes are saved in the CiA402 device profile:

No.	Operation modes with profile generation in the Servo Drive	
1	**Profile position mode**	**Page 73**
2	**Velocity mode (U/F operation)**	**Page 76**
3	**Profile velocity mode**	**Page 77**
4	**Reserved**	**-**
5	**Reserved**	**-**
6	**Homing mode**	**Page 78**

Table 12.1 Operation modes with profile generation in the Servo Drive

No.	Operation modes with profile generation in the Servo Drive	
7	**Interpolated position mode**	**Page 80**
Manufacturer-specific operation modes (not defined in CiA402):		
8	**Cyclic synchronous position mode (only EtherCAT)**	**Page 81**
9	**Cyclic synchronous velocity mode (only EtherCAT)**	**Page 82**
10	**Cyclic synchronous torque mode (only EtherCAT)**	**Page 83**

Table 12.2 Operation modes with profile generation in the controller

12.2 Operation modes with profile generation in the Servo Drive

For the operation modes with profile generation in the Servo Drive, the trajectory is calculated internally based on the target position/target velocity specified. The external controller has an effect on the control word and the target values.

The profile parameters in the drive describe:

- How the Servo Drive reaches this position / velocity.
- The shape of the movement profile (e.g. trapezoid, triangular / gradient of the ramps, etc.).

The drive generates the related profile.

12.2.1 1 Profile position mode

Function description

The "Profile position mode" operation mode supports two ways of specifying the target position:

- Set of setpoints
- Single setpoint

Set of setpoints:

Once the position setpoint is reached, the drive moves immediately to the next target position transferred.

Single setpoint:

Once the drive reaches the target position:

- This situation is signalled to the master
- The drive receives a new setpoint
- The drive stops at each target position before it moves to the next target position.

The timing of the following bits controls the specification of the target position:

- New setpoint in the control word
- Change set immediately in the control word
- Setpoint acknowledge in the status word

These bits implement subsequent positioning while a positioning task is in progress.

In the "Profile position mode" operation mode (Modes of operation = 1)

- The axis completes relative or absolute positioning movements.
- The drive is in position control.

Object no.	Object name	Object code	Type	Attr.
0x607A	Target position	VAR	Integer32	rw
0x607d	Software position limit	ARRAY	Integer32	rw
0x6081	Profile velocity	VAR	Unsigned32	rw
0x6083	Profile acceleration	VAR	Unsigned32	rw
0x6084	Profile deceleration	VAR	Unsigned32	rw
0x6085	Quick stop deceleration	VAR	Unsigned32	rw
0x6064	Position actual value	VAR	Integer32	r
0x607E	Polarity	VAR	Unsigned8	rw

Table 12.3 Supported objects in the "Profile position mode"

Set the scaling via:

- The scaling wizard or
- The factor group objects.

Structure of the operation mode

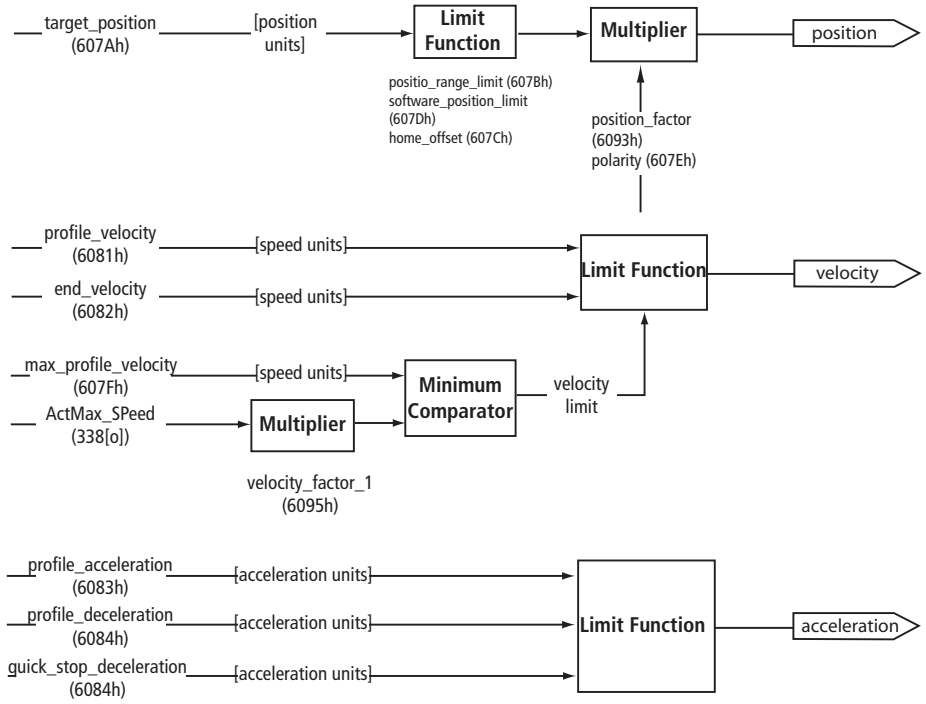


Figure 12.1 Structure of profile position mode

Operation mode-specific bits in the control word

Bit	Name	Value	Description
4	New setpoint	0	Target position not applied
		1	Target position applied
5	Change set immediately	0	Terminate actual positioning and start next positioning
		1	Interrupt actual positioning and start next positioning
6	Absolute / relative	0	Target position is an absolute value
		1	Target position is a relative value
8	Halt	0	Implement positioning
		1	Stop axis with the deceleration profile (if acceleration profile not active)

Table 12.4 PROFILE POSITION MODE bits in the control word

Operation mode-specific bits in the status word

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: Target position not reached Halt = 1: Axis is braking
		1	Halt = 0: Target position reached Halt = 1: Velocity of the axis is 0
12	Setpoint acknowledge	0	Trajectory generation has not applied the position values
		1	Trajectory generation has applied the position values
13	Following error	0	No following error
		1	Following error
14	ROT_0	1	Axis is at standstill, velocity is less than parameter **P 0745 MON_REFWINDOW**

Table 12.5 PROFILE POSITION MODE bits in the status word

Setpoint transfer from a master (host computer)

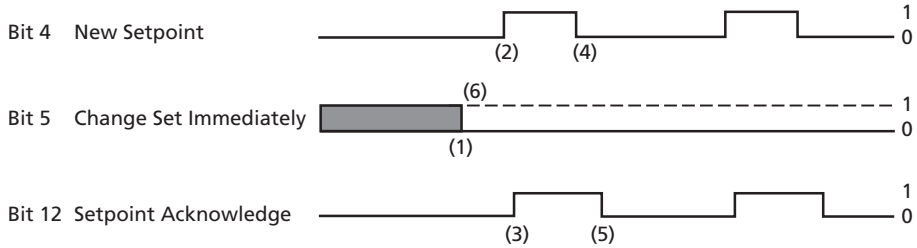


Figure 12.2 Setpoint transfer from a master

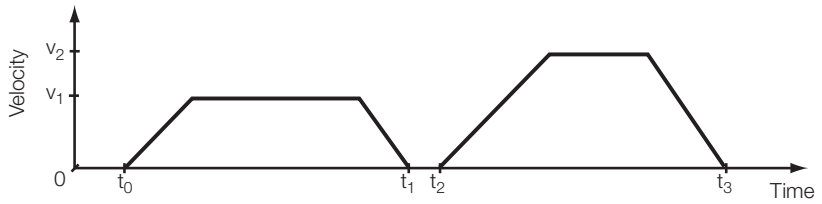
(1) If bit 5 "Change set immediately" is equal to "0" (continuous line in the figure above), the drive expects a "single setpoint".

(2) Once the setpoint is transferred to the drive, the master activates the positioning by setting bit 4 "New setpoint" in the control word.

(3) After the new data have been detected and saved, the drive sets bit 12 "Setpoint acknowledge" in the status word.

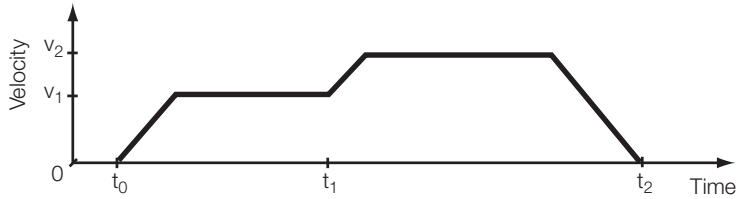
(4) The master clears bit 4 "New setpoint".

(5) By clearing bit 12 "Setpoint acknowledge" the drive applies a new setpoint.



If the target position is reached at time t_1:
- The velocity is set to 0.
- The next target position is triggered at time t_2.

(6) If bit 5 "Change set immediately" is set to "1" (dotted line in Figure 12.2) the new target position is applied.



The drive receives:
- The 1st Target position at time t_0,
- The 2nd target position at time t_1

And continues the movements.

12.2.2 2 Velocity mode (U/F operation)

The velocity mode operation mode (Modes of operation = 2) controls the drive in frequency-controlled operation (U/f operation).

The units, the setpoint and the ramp sizes are defined by the settings in the factor group. On this topic, also see chapter “10.3 Units and scaling, factor group” on page 64.

The device supports the following objects in the "Velocity mode" operation mode:

Object no.	Object name	Object code	Type
0x6042	vl Target velocity	VAR	Integer16
0x6046	vl Min Max amount	ARRAY	Unsigned32
0x6048	vl Velocity acceleration	ARRAY	Unsigned32
0x6049	vl Velocity deceleration	ARRAY	Unsigned32

Table 12.6 Velocity mode

Object no.	Object name	Description
0x6046	vl Min Max amount	Index: 0 = Min. velocity in user units; 1 = Max. velocity in user units
0x6048	vl Velocity acceleration	Index: 0 = Velocity change in user units; 1 = Per time unit
0x6049	vl Velocity deceleration	Index: 0 = Velocity change in user units; 1 = Per time unit

Table 12.7 Limits in U/f operation

Operation mode-specific bits in the control word

Bit	Name	Value	Description
5	Unlock ramp	0	Fix ramp function generator output value at actual value
		1	Ramp function generator follows the ramp function generator input value
6	Reference ramp	0	Set ramp function generator input value to ZERO
		1	Ramp function generator input value corresponds to setpoint
8	Halt	0	No command
		1	Stop axis

Table 12.8 PROFILE VELOCITY MODE (U/F OPERATION) bits in the control word

12.2.3 3 Profile velocity mode

In the "Profile velocity mode" operation mode (Modes of operation = 3):

- The device is operated using a velocity setpoint as per the CiA402 device profile.
- The drive is in velocity control.

The units, the setpoint and the ramp sizes are defined by the settings in the factor group (see chapter “10.3 Units and scaling, factor group” on page 64).

Relevant object for this "Profile velocity mode" operation mode:

Object no.	Object name	Object code	Type
0x606C	Velocity actual value	VAR	Int32
0x60FF	Target velocity	VAR	Int32
0x6094	Velocity encoder factor	ARRAY	Int32
0x6083	Profile acceleration	VAR	Int32
0x6084	Profile deceleration	VAR	Int32
0x6085	Quick stop deceleration	VAR	UInt32
0x607E	Polarity	VAR	UInt8
0x607F	Max. profile velocity	VAR	UInt32

Table 12.9 Profile velocity mode

Note:
The "Profile velocity mode" updates cyclically:
- The objects listed in the table.
- The object 0x6064 "Position actual value".

Operation mode-specific bits in the control word

Bit	Name	Value	Description
8	Halt	0	Execute movement
		1	Stop axis

Table 12.10 PROFILE VELOCITY MODE bits in the control word

Operation mode-specific bits in the status word

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: Target velocity not (yet) reached Halt = 1: Axis slows down
		1	Halt = 0: Target velocity reached Halt = 1: Axis has velocity 0
12	Speed	0	Speed is not 0
		1	Speed is 0
13	Maximum slippage error	0	Maximum deceleration not achieved
		1	Maximum deceleration achieved

Table 12.11 PROFILE VELOCITY MODE bits in the status word

Structure of the operation mode

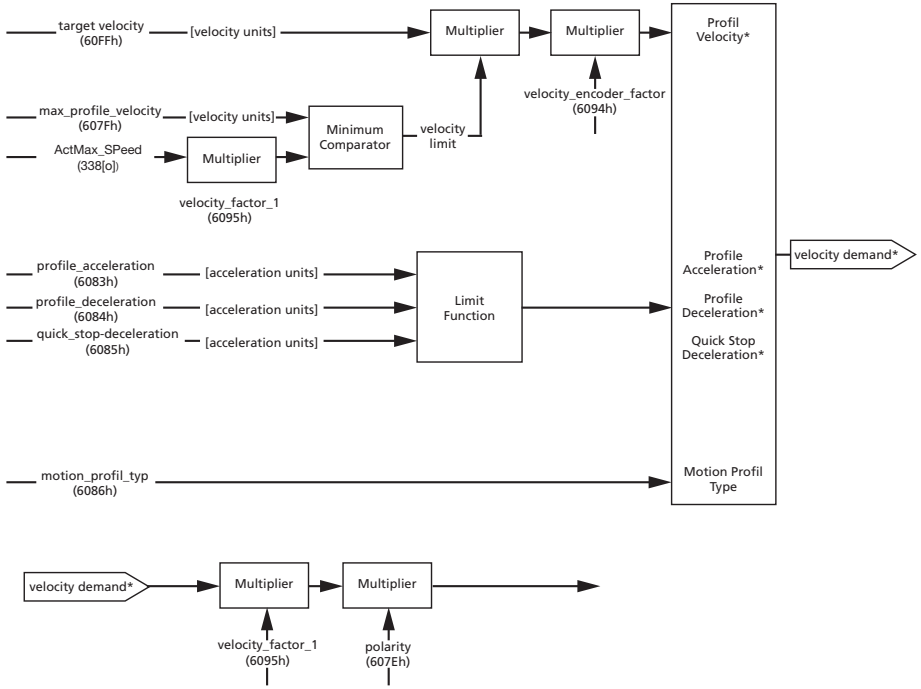


Figure 12.3 Structure of profile velocity mode

12.2.4 Homing mode

The "Homing mode" operation mode (Modes of operation = 6) homes a position-controlled axis. The homing method programmed (homing method object 0x6098) defines the movement of the drive.

Note:
Controller-controlled homing of the drive using the touchprobe function. See chapter “14.1 Touch probe” on page 87.

The homing methods differ in the usage of hardware switches, reference marks and zero pulses in the encoder system.

Note:
For the following limit switch and reference mark functionality, configure parameters for appropriate digital inputs:

- Limit switch function
- LCW - clockwise HW limit switch
- LCCW - counter-clockwise HW limit switch
- HOMSW - reference mark

P 0100[0] to **P 0112[0]** = function assignment for the digital inputs

Relevant objects for the "Homing mode" operation mode:

Object no.	Object name	Object code	Type	Attr.
0x607C	HomeOffset	VAR	Integer32	rw
0x6098	HomingMethod	VAR	Integer8	rw
0x6099	HomingSpeeds *	ARRAY	Unsigned32	rw
0x609A	HomingAcc	VAR	Unsigned32	rw

* 0x6099.01 - Fast jog speed
0x6099.02 - Slow jog speed

Table 12.12 Objects for the "Homing mode" operation mode

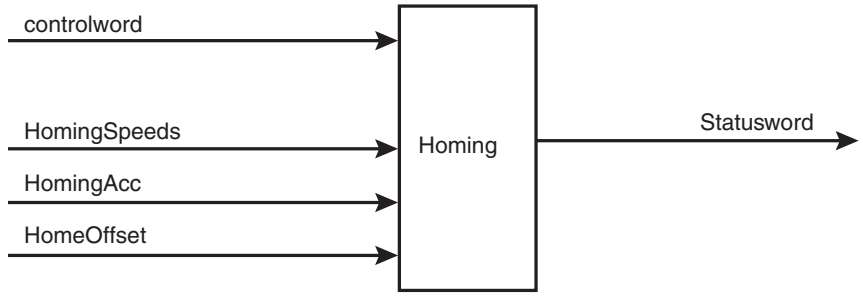


Figure 12.4 Homing function

The drive supports all 35 homing methods defined in CiA402.

You will find information on the functions and movement sequences for the homing methods in the online help for the Servo Drive.

Home Offset

The object "Home Offset" is the difference between the 0 position of the application and the "Home Position" found during the homing. The difference is indicated in position units.

After homing the sum of "Home Offset" and "Home Position" is the new zero position.

All the following absolute positioning tasks relate to this new zero position.

The homing method and its properties can be changed in 2 ways:

1. Via Moog DriveAdministrator 5.
2. Via the CANopen field bus system.

The parameters configured via CANopen control the objects for the "Homing mode". Example: Changes to object 0x6098 change the homing method.

NOTE:
For more detailed information, please refer to the MSD Servo Drive - Device Help (available in the download area at www.moogsoftwaredownload.com/msd.html)

Operation mode-specific bits in the control word

Bit	Name	Value	Description
4	Homing operation start	0	Homing inactive
		0 ⇨ 1	Start homing
		1 ⇨ 0	Interrupt homing
8	Halt	0	Run instruction from bit 4
		1	Stops axis with deceleration

Table 12.13 HOMING MODE bits in the control word

Operation mode-specific bits in the status word

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: Target position not reached Halt = 1: Axis is braking
		1	Halt = 0: Target position reached Halt = 1: Axis at velocity 0
12	Homing attained	0	Homing not undertaken
		1	Homing undertaken successfully
13	Homing error	0	No homing error
		1	Homing error. Homing failed. Error code (see Online help) indicates the cause of the error.
14	ROT_0	1	Axis stationary. Velocity is less than parameter **P0745 MON_REFWINDOW**

Table 12.14 HOMING MODE bits in the status word

12.3 Cyclic operation modes, profile generation in the controller

In the cyclic operation modes:

- The controller generates the profile.
- The internal profile generator in the drive is not active.
- The drive interpolates the setpoints transferred cyclically from the controller depending on the operation mode (position, velocity, torque).

12.3.1 7 Interpolated position mode

The "Interpolated position mode" operation mode (Modes of operation = 7):

- Is used to position axes via CANopen.
- Moves one or more axes, co-ordinated by a controller.

The controller:

- In the "Interpolated position mode" prepares the movement profile without the profile generator.
- Cyclically transfers position values.
- The Servo Drive undertakes the "fine interpolation" (e.g. linear interpolation).

The changes to the position setpoints per unit time are defined by the profile the axis is following.

The device supports the following objects in the "Interpolated position mode" operation mode:

Object no.	Object name	Object code	Type
0x60C0	Interpolation sub mode select	VAR	Integer16
0x60C1	Interpolation data record	ARRAY	Integer32
0x60C2	Interpolation time period	RECORD	Index0: Unsigned8 Index1: Integer8

Table 12.15 Objects supported in the "Interpolated position mode"

The object 0x60C0 "Interpolation sub mode select" supports the following settings:

Object 0x60C0	Description	P 370[0] - CON_IP
-1	NoIp = No interpolation	0
0	Lin = Linear interpolation	1
-2	SplineExtFF = Interpolation with external feed forward control	2
-3	SplineII = Cubic spline interpolation	3
-4	NonIPSpline = Cubic spline approximation	4
-5	Cos = Cosine interpolation	5

Table 12.16 Settings, object 0x60C0

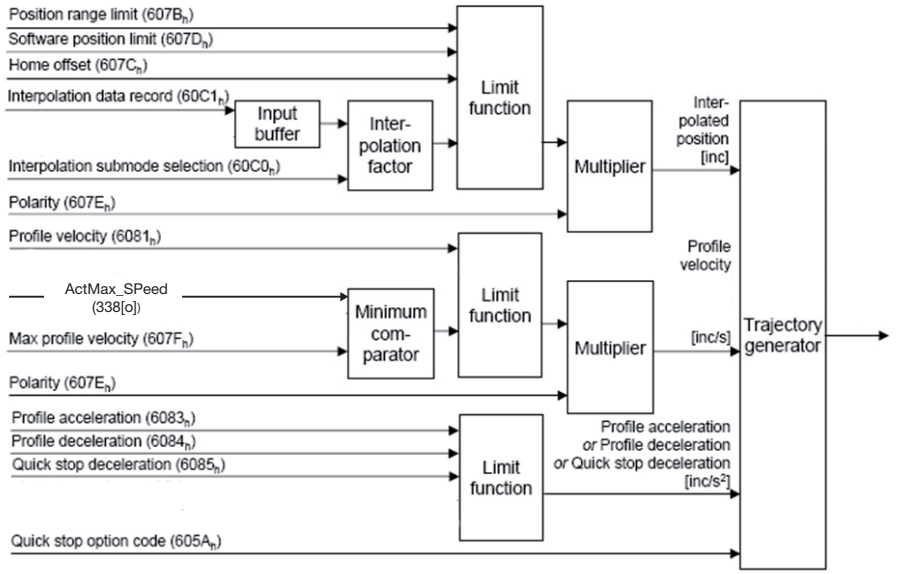


Figure 12.5 Structure of the "Interpolated position mode"

Operation mode-specific bits in the control word

Bit	Name	Value	Description
4	Enable IP Mode	0	Interpolating operation not active
		1	Interpolating operation active
8	Halt	0	Run instruction from bit 4
		1	Stop axis

Table 12.17 Operation mode-specific bits in the control word

Operation mode-specific bits in the status word

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: Position not (yet) reached Halt = 1: Axis is braking
		1	Halt = 0: Position reached Halt = 1: Axis at velocity 0
12	IP mode active	0	Interpolating operation not active
		1	Interpolating operation active
14	ROT_0	0	Axis in motion
		1	Axis is at standstill, velocity is less than parameter **P0745 MON_REFWINDOW**
15	Axis synchronized	0	Axis not synchronised
		1	Axis synchronised

Table 12.18 Operation mode-specific bits in the control word

12.3.2 8 Cyclic synchronous position mode (only EtherCAT)

In the "Cyclic synchronous position mode" operation mode (Modes of operation = 8):

- The drive cyclically receives position setpoints from the controller.
- The drive implements position control.
- Optionally, the controller transfers an additive velocity and torque setpoint as feed forward control values.

The device supports the following objects in the "Cyclic synchronous position mode" operation mode:

Object no.	Object name	Object code	Type
0x607A	Target position	VAR	Integer32
0x60B1	Velocity offset	VAR	Integer32
0x60B2	Torque offset	VAR	Integer16

Table 12.19 Objects supported in the "Cyclic synchronous position mode"

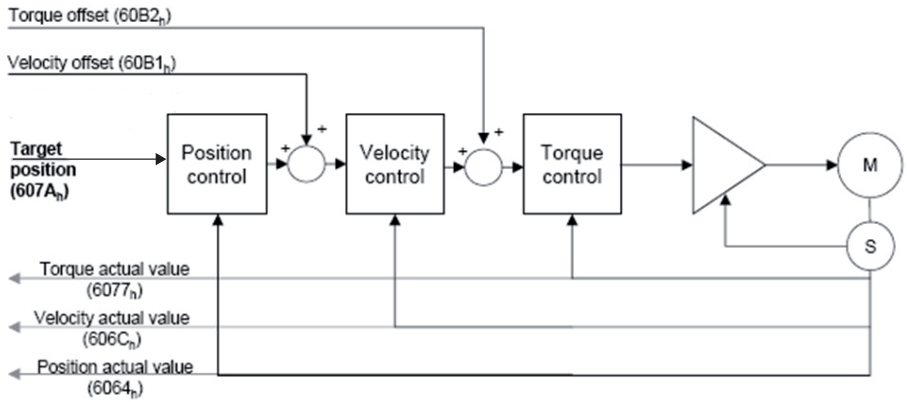


Figure 12.6 Overview "Cyclic synchronous position mode"

Operation mode-specific bits in the status word

Bit	Name	Value	Description
12	Target position	0	Target position ignored
		1	Use target position as input
13	Following error	0	No following error
		1	Following error
14	ROT_0	0	Axis in motion
		1	Axis is at standstill, velocity is less than parameter **P0745 MON_REFWINDOW**
15	Axis synchronized	0	Axis not synchronised
		1	Axis synchronised

Table 12.20 Operation mode-specific bits in the status word

12.3.3 9 Cyclic synchronous velocity mode (only EtherCAT)

In the "Cyclic synchronous velocity mode" operation mode (Modes of operation = 9):

- The controller cyclically transfers velocity setpoints to the drive.
- The drive implements velocity control.
- Optionally, the controller transfers an additive velocity and torque setpoint as a feed forward control values.

The device supports the following objects in the "Cyclic synchronous velocity mode" operation mode:

Object no.	Object name	Object code	Type
0x60FF	Target velocity	VAR	Integer32
0x60B1	Velocity offset	VAR	Integer32
0x60B2	Torque offset	VAR	Integer16

Table 12.21 Objects supported in the "Cyclic synchronous velocity mode"

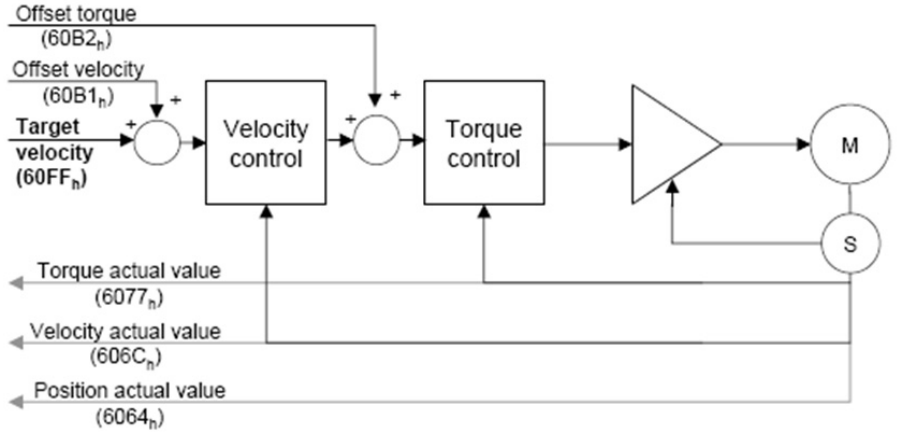


Figure 12.7 Overview "Cyclic synchronous velocity mode"

Operation mode-specific bits in the status word

Bit	Name	Value	Description
12	Target velocity	0	Target velocity ignored
		1	Use target velocity as input
14	ROT_0	0	Axis in motion
		1	Axis is at standstill, velocity is less than parameter **P0745 MON_REFWINDOW**
15	Axis synchronized	0	Axis not synchronised
		1	Axis synchronised

Table 12.22 Operation mode-specific bits in the status word

12.3.4 10 Cyclic synchronous torque mode (only EtherCAT)

In the "Cyclic synchronous torque mode" operation mode (Modes of operation = 10):

- The controller cyclically transfers torque setpoints to the drive.
- The drive implements current control.
- Optionally, the controller transfers an additive torque setpoint as a feed forward control value.

The device supports the following objects in the "Cyclic synchronous torque mode" operation mode:

Object no.	Object name	Object code	Type
0x6071	Target torque	VAR	Integer16
0x60B2	Torque offset	VAR	Integer16

Table 12.23 Objects supported in the "Cyclic synchronous torque mode"

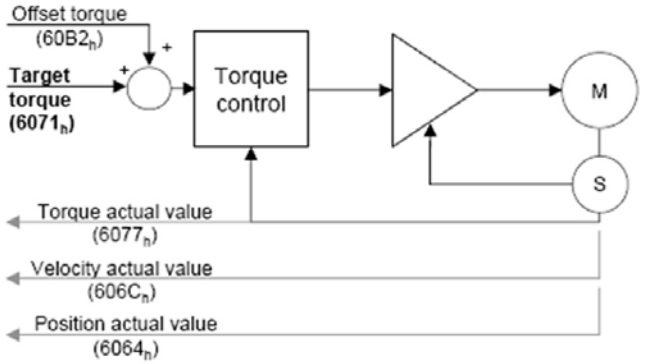


Figure 12.8 Overview "Cyclic synchronous torque mode"

Operation mode-specific bits in the status word

Bit	Name	Value	Description
12	Target torque	0	Target torque ignored
		1	Use target torque as input
14	ROT_0	0	Axis in motion
		1	Axis is at standstill, velocity is less than parameter **P0745 MON_REFWINDOW**
15	Axis synchronized	0	Axis not synchronised
		1	Axis synchronised

Table 12.24 Operation mode-specific bits in the status word

12.4 External velocity/torque feed forward control

In the "Cyclic synchronous position mode" (CSP, chapter“12.3.2 8 Cyclic synchronous position mode (only EtherCAT)” on page 81) or "Cyclic synchronous velocity mode" (CSV, chapter “12.3.3 9 Cyclic synchronous velocity mode (only EtherCAT)” on page 82) operation modes:

- Optionally the controller transfers external feed forward control values for the velocity and the torque.
- For this purpose switch over the internal feed forward control in the drive.

The following tables show relevant settings:

Object no.	Object name	Data type	Standardisation
0x60B1	Velocity offset	Integer32	As per the scaling for speeds (CiA402 factor group)
0x60B2	Torque offset	Integer16	In [‰] referred to motor rated torque in object 0x6076, i.e. a value of 1000 corresponds to the motor rated torque.

Table 12.25 CiA402 objects for the external feed forward control

Parameter	Function	Value
P 0375 – CON_IP_SFFScale	Scaling, speed feed forward control	0 – 100 % referred to the feed forward control value
P 0376 – CON_IP_TFFScale	Scaling, torque feed forward control	0 – 100 % referred to the feed forward control value
P 0379 – CON_IP_FFMode	Switching over the feed forward control sources and special setpoint formats	See individual sub indices
• Sub index 0	Position high resolution	0 = 32-bit position setpoint (default)
• Sub index 1	Source, speed feed forward control values	0 = Internal feed forward control (default) 1 = External feed forward control
• Sub index 2	Source, torque feed forward control values	0 = Internal feed forward control (default) 1 = External feed forward control

Table 12.26 MSD Servo Drive device parameters

Interpolation types

Set the interpolation type via the parameter **P 0370 CON_IP**.

Checking the feed forward control variables in Moog DriveAdministrator 5

There are 2 ways of checking the feed forward variables sent externally in the MSD Servo Drive:

1. You will find the objects for feed forward control as device parameters in the area CANopen/EtherCAT
2. Using the internal oscilloscope you can acquire the variables nref_Ext (external velocity feed forward control) and mref_Ext (external torque feed forward control).

13 "Emergency messages"

13.1 General

"Emergency messages":

- Provide information on errors and malfunctions in a network or in the related bus users (Servo Drives).
- Have a high priority.

Bit assignment, "emergency message":

Byte	0	1	2	3	4	5	6	7
Bit:	0 ... 7	8 ... 15	16 ... 23	24 ... 39		40 ... 47	48 ... 63	
Profile	**Device profile CiA402**			**Servo Drive**				
Error	Emergency error code as per CiA402		Error register (object 1001 h)	Error number	Error location	Operating hours counter (in complete hours)		

Table 13.1 Emergency message

Crucial bytes in the "emergency message" for error diagnostics:

- Byte 3 - error number (indicates an error event)
- Byte 4 - error location (indicates the cause of the error)

The Servo Drive indicates errors in the field bus system, e.g. incorrect configurations, field bus malfunctions or others using the **error code 0xFF00 h.**

Notes:
In the MSD Servo Drive - Device Help you will find:

- A list of the "error codes" with related "emergency messages" and rectification measures.

You will find a detailed and general explanation of the "emergency messages" in CiA402.

13.2 Error acknowledgement

Methods to acknowledge the device error:

1. The object 6040 h, control word bit 7, edge controlled
   ("You will find a list of "Error messages with rectification measures" in the "MSD Servo Drive Online Device Help".").
2. The control input programmed with reset functionality.
3. A rising edge on the control terminal (hardware enable ENPO).
4. The two buttons under the 7-segment display on the Servo Drive.
5. The PC user software Moog DriveAdministrator.
6. Write the value "1" to the parameter **P0153** /
   Object x 2099 MPRO_DRV-COM_FaultReset
   (via Moog DriveAdministrator or the field bus system).

13.3 Error acknowledgement via the field bus system

The commonest method of error acknowledgement is "error acknowledgement via the field bus system". For this purpose:

- Trigger a rising edge in parameter **P 2208[0] MPRO_402_Controlword** in **object 6040 h**, bit 7. You will find more detailed information in chap. "11.3.1 Object 6040h control word" on page 69.

The following "emergency message" signals the error acknowledgement:

ID	Data bytes	Description
Emergency	00 00 00 00 00 00 00 00	"Emergency message" error acknowledgement

Table 13.2 Error acknowledgement

If the cause of the error has been rectified:

- The Servo Drive leaves the error state.

If the cause of the error has not been rectified:

- After the error acknowledgement, the Servo Drive returns to the error state due to the "emergency message".

13.4 Error acknowledgement via the Moog DriveAdministrator

- Locate cause of error using the **Moog DriveAdministrator 5** error message (see Error message **Moog DriveAdministrator** or Error history).
- Rectify cause of error.
- Reset error via Device status dialog box.

NOTE:
You will find a list of "Error messages with rectification measures" in the "MSD Servo Drive - Device Help".

14 Technology functions

14.1 Touch probe

The Touch probe function acquires drive positions as a function of the following input signals:

- Digital input ISD05.
- Digital input ISD06.
- Zero pulse.

Use the parameter **P2285 "Touch probe function selector"** to select between different implementations, e.g.:

- CiA402 implementation.
- Manufacturer-specific implementation (BECK) with representation of all latch values in **object 60BA h.**
- Manufacturer-specific implementation (SPECIFIC) with representation of the values in the related **objects 60BA h, 60BB h, 60BC h and 60BD h.**

14.1.1 Touch probe implementation according to CiA402

1. To utilise the Touch probe function "Implementation according to CiA402", set the parameter **P2285 "Touch probe function selector"** to 1 = "CiA402(1)".
2. To acquire signals with both digital inputs ISD05 and ISD06, configure the parameter **P0106 + P0107 "MPRO_Input_FS_ISD0x"** as PROBE(15). You will find descriptions of the parameters in the device online help, in "Function selector digital inputs".

Then map following objects to the process data:

Name	Value
RxPDO	0x60B8 Touch probe function
TxPDO	0x60B9 Touch probe status 0x60BA Touch probe pos1 pos value
	Alternatively: • 0x60BB Touch probe pos1 neg value • 0x60BC Touch probe pos2 pos value • 0x60BD Touch probe pos2 neg value

Table 14.1 Object RxPDO and TxPDO: Touch Probe function

Via **object 0x60B8 h "Touch probe function"** set the signal to be triggered. Settings available:

- Trigger with touch probe 1 (dig. input ISD05) on rising/falling edge.
- Trigger with touch probe 2 (dig. Input ISD06) on rising/falling edge.
- Trigger on encoder zero pulse, rising edge.

Bit	Function
0	Activate measuring point 1
1	Measuring point 1 continuous operation
2	Measuring point 1 zero pulse (measurement starts only with zero pulse)
3	-
4	Measuring point 1 enable latch on positive edge (also use for encoder zero signal)
5	Measuring point 1 enable latch on negative edge
6	-
7	-
8	Activate measuring point 2
9	Measuring point 2 continuous operation
10	Measuring point 2 zero pulse (measurement starts only with zero pulse)
11	-
12	Measuring point 2 enable latch on positive edge (also use for encoder zero signal)
13	Measuring point 2 enable latch on negative edge
14	-
15	-

Table 14.2 Touch probe latch control object: 60B8 h

The **object 0x60B9 h** provides the status of the Touch probe function.

If the Servo Drive registers an activated signal from **object 0x60B8 h**, the touch probe sets the corresponding bit in the status word.

Bit	Function
0	Activate measuring point 1
1	Measuring point 1 saves positive edge value
2	Measuring point 1 saves negative edge value
3	-
4	-
5	-
6	Measuring point 1 saves positive edge value (only continuous operation. Bit switches if the latch status changes)
7	Measuring point 1 saves negative edge value (only continuous operation. Bit switches if the latch status changes)
8	Activate measuring point 2
9	Measuring point 2 saves positive edge value
10	Measuring point 2 saves negative edge value
11	-
12	-
13	-
14	Measuring point 2 saves positive edge value (only continuous operation. Bit switches if the latch status changes)
15	Measuring point 2 saves negative edge value (only continuous operation. Bit switches if the latch status changes)

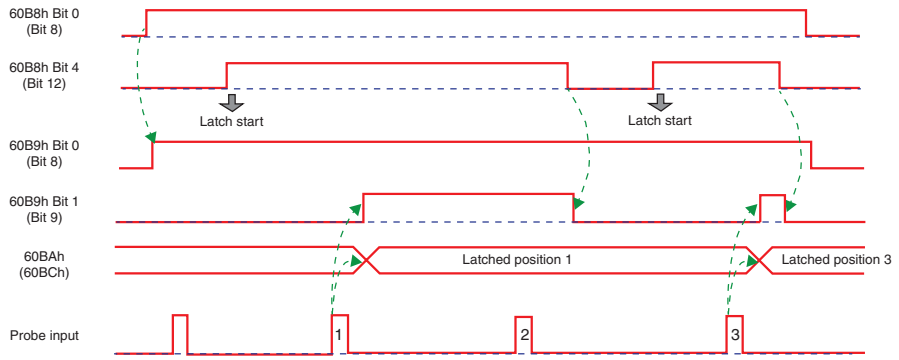
Table 14.3 Touch probe latch status object: 60B9 h

Timing diagram

The following timing diagram shows a measurement based on an example of the touch probe ISD05 and the
related bits in the touch probe control word and status word

- In the single trigger mode and
- In the continuous trigger mode.

• Single Trigger Mode (60B8 h , Bit 1 = 0 or Bit 9 = 0)



• Continuous Trigger Mode (60B8 h , Bit 1 = 1 or Bit 9 = 1)

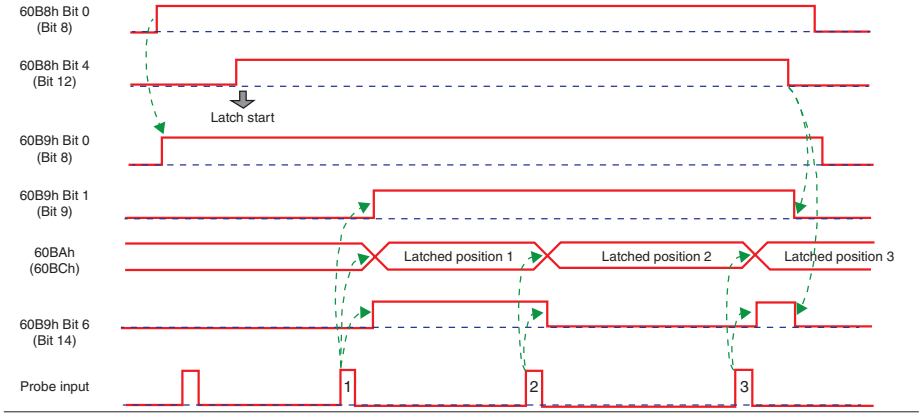


Figure 14.1 Timing of a measurement in the "Single trigger mode" and "Continuous trigger mode"

The timing of the other signals that can be configured is corresponding.

14.1.2 Manufacturer-specific touch probe implementation

1. To utilise the function "Manufacturer-specific implementation", set the parameter **P 2285 Touch probe function selector** to
   – **2** = "BECK2" or
   – **3** = "SPECIFIC(3)".
2. To acquire signals with both digital inputs ISD05 and ISD06, configure the parameter **P0106 + P0107 MPRO_Input_FS_ISD0x** as "PROBE(15)".

You will find descriptions of the parameters in the MSD Servo Drive - Device Help, in "Function selector digital inputs".

Then map the following objects or use SDO to read or write:

Name	Value
RxPDO	0x60B8 Touch probe function
TxPDO	0x60B9 Touch probe status 0x60BA Touch probe pos1 pos value
	Alternatively: • 0x60BB Touch probe pos1 neg value • 0x60BC Touch probe pos2 pos value • 0x60BD Touch probe pos2 neg value

Table 14.4 Object RxPDO and TxPDO: Touch Probe function

Using the **object 0x60B8 h "Touch probe function"** define the edge on which the signal is triggered:

1. Falling edge
2. Rising edge
3. Both edges.

Bits 0 - 4 in the object "Touch probe function" (edge controlled) activate the related functions.

Bits 8 - 11 in the object "Touch probe status" acquire and control the functions saved.

Note:
Start new measurement after the arrival of the signal configured!
Starting the measurement - reset related bit and set again.

Bit	Value (bin)	Value (hex)	Description
0	0000 0000 0000 0001	xx01	Enable external latch 1 (positive rise) - digital input ISD05
1	0000 0000 0000 0010	xx02	Enable external latch 1 (negative rise) - digital input ISD05
2	0000 0000 0000 0100	xx04	Enable external latch 2 (positive rise) - digital input ISD06
3	0000 0000 0000 1000	xx08	Enable external latch 2 (negative rise) - digital input ISD06
4	0000 0000 0001 0000	xx10	Enable internal latch C (positive rise) - Encoder zero pulse
5-7	-	-	Reserved
8-12	0000 0001 0000 0000	01xx	Read external latch 1 (positive rise) - digital input ISD05
	0000 0010 0000 0000	02xx	Read external latch 1 (negative rise) - digital input ISD05
	0000 0011 0000 0000	03xx	Read external latch 2 (positive rise) - digital input ISD06
	0000 0100 0000 0000	04xx	Read external latch 2 (negative rise) - digital input ISD06
	0000 0101 0000 0000	05xx	Read internal latch C (positive rise) - Encoder zero pulse
13-15	-	-	Reserved

Table 14.5 Object 0x60B8 h: Touch probe function

The **object 0x60B9 h** provides the status of the Touch probe function.

If the Servo Drive registers an activated signal from **object 0x60B8 h**, the touch probe sets the corresponding bit (0 - 4) in the status word.

Bit	Value (bin)	Value (hex)	Description
0	0000 0000 0000 0001	xx01	External latch 1 valid (positive rise)
1	0000 0000 0000 0010	xx02	External latch 1 valid (negative rise)
2	0000 0000 0000 0100	xx04	External latch 2 valid
3	0000 0000 0000 1000	xx08	External latch 2 valid

Table 14.6 Object 0x60B9 h: Touch probe status

Bit	Value (bin)	Value (hex)	Description
4	0000 0000 0001 0000	xx10	Internal latch C valid (positive rise)
5-7	-	-	Reserved
8-11	0000 0001 0000 0000	01xx	Acknowledge value external latch 1 (positive rise)
	0000 0010 0000 0000	02xx	Acknowledge value external latch 1 (negative rise)
	0000 0011 0000 0000	03xx	Acknowledge value external latch 2 (positive rise)
	0000 0100 0000 0000	04xx	Acknowledge value external latch 2 (negative rise)
	0000 0101 0000 0000	05xx	Acknowledge value internal latch C (positive rise)
12-15	0001 0000 0000 0000	1xxx	Reserved
	0010 0000 0000 0000	2xxx	Reserved
	0100 0000 0000 0000	4xxx	Reserved
	1000 0000 0000 0000	8xxx	Reserved

Table 14.6 Object 0x60B9 h: Touch probe status

After setting the bits 8 – 11 in the status word, depending on **P 2285** Touch probe function selector, the position saved is written to the **object 0x60BA h, 0x60BB h, 0x60BC h or 0x60BD h**.

Timing diagram

The timing diagram for a measurement based on an example of the touch probe ISD05 and the related bits in the touch probe control word and status word:

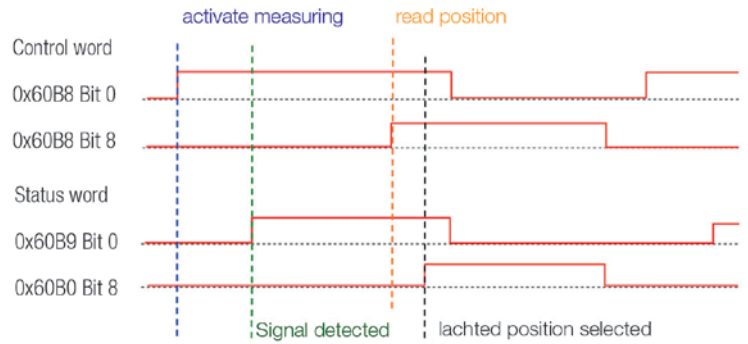


Figure 14.2 Timing diagram for a Touch probe function measurement

The timing of the other signals that can be configured is corresponding.

14.2 Round table function

Use the following objects to configure the Round table function:

Object	Object name	Object code	Type
0x607B h	Position range limit	ARRAY	Integer32
0x60F2 h	Positioning option code	VAR	Unsigned16

Table 14.7 Objects for round table function

For the Round table function use the **object 0x60F2 h "Positioning option code"** contrary to the standardisation according to CiA402.

Pay attention to bits 6 and 7 (see table):

Value (hex)	Meaning
0x00	As for linear
0x40	Counter-clockwise direction of rotation
0x80	Clockwise direction of rotation
0xC0	Path optimised

Table 14.8 Bit assignment, object 0x60F2

15 Other documents

Moog documentation	ID no. / file formats	Web site
MSD Servo Drive Compact - Operation Manual	CA97555-001 / PDF	https://www.moogsoftwaredownload.com/msd.html
MSD Servo Drive AC-AC Single-Axis System - Operation Manual	CA65642-001 / PDF	
MSD Servo Drive DC-AC Multi-Axis System - Operation Manual	CA97554-001 / PDF	
MSD Servo Drive Multi-Axis System Supply Unit - Operation Manual	CA97556-001 / PDF	
MSD Servo Drive - Device Help	CB40859-001 / PDF and HTML	

Table 15.1 Moog documentation

Other field bus documentation	Web site
CiA® CiA301 (V 4.2.0): CANopen application layer and communication profile	www.can-cia.com
CiA® CiA402: CANopen device profile for drives and motion control (Rev. V 2.0)	
EtherCAT Communication Specification Version 1.0 2004	www.ethercat.org
EtherCAT Indicator Specification Proposal V0.91 2005	www.ethercat.org
Series IEC/EN 61158 standards (-1 to -6-x): Industrial communication networks – Fieldbus specifications	See: www.beuth.de
Other information on the CANopen field bus system	www.can-cia.com
Other information on the EtherCAT field bus system	www.ethercat.org

Table 15.2 Other field bus documentation

16 Glossary

Moog DriveAdministrator 5	PC software from Moog for (initial) commissioning and process diagnostics on the devices in the MSD Servo Drive family
Electronic device data sheet – EDS file	Is installed in the "CANopen master" (controller) and is used to integrate MSD Servo Drives into the CANopen device network (page 91)
MSD	MSD Servo Drive: **Single-Axis System** - group of MSD Servo Drive devices (rated currents 4 - 450 A) that are connected to the AC mains and that operate an axis. **Multi-axis System** - group of MSD Servo Drive devices (rated currents 4 - 170 A) that comprises DC-supplied axis modules and matched supply units.
MSD Compact	MSD Servo Drive: **Compact** - smallest member of the MSD Servo Drive family (rated currents 2 - 16 A). The devices are connected to the AC mains and operate an axis.
ESI (xml) device description file	Is used to integrate drive controllers into the EtherCAT device network. Transfers data, properties and functionality of the field bus system to the "EtherCAT master" (page 91).
State machine	This describes the transitions between the various system states. A state transition is triggered by a defined event, e.g. a control sequence or setting an input.

17 Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

TAKE A CLOSE LOOK.

Moog solutions are only a click away. Visit our worldwide Web site for more information and the Moog facility nearest you.

Australia
+61 3 9561 6044
Service +61 3 8545 2140
info.australia@moog.com
service.australia@moog.com

Brazil
+55 11 3572 0400
info.brazil@moog.com
service.brazil@moog.com

Canada
+1 716 652 2000
info.canada@moog.com

China
+86 21 2893 1600
Service +86 21 2893 1626
info.china@moog.com
service.china@moog.com

France
+33 1 4560 7000
Service +33 1 4560 7015
info.france@moog.com
service.france@moog.com

Germany
+49 7031 622 0
Service +49 7031 622 197
info.germany@moog.com
service.germany@moog.com

Hong Kong
+852 2 635 3200
info.hongkong@moog.com

India
+91 80 4057 6666
Service +91 80 4057 6664
info.india@moog.com
service.india@moog.com

Ireland
+353 21 451 9000
info.ireland@moog.com

Italy
+39 0332 421 111
Service 800 815 692
info.italy@moog.com
service.italy@moog.com

Japan
+81 46 355 3767
info.japan@moog.com
service.japan@moog.com

Korea
+82 31 764 6711
info.korea@moog.com
service.korea@moog.com

Luxembourg
+35 2 40 46 401
info.luxembourg@moog.com

The Netherlands
+31 252 462 000
info.thenetherlands@moog.com
service.netherlands@moog.com

Russia
+7 831 713 1811
Service +7 831 764 5540
info.russia@moog.com
service.russia@moog.com

Singapore
+65 677 36238
Service +65 651 37889
info.singapore@moog.com
service.singapore@moog.com

South Africa
+27 12 653 6768
info.southafrica@moog.com

Spain
+34 902 133 240
info.spain@moog.com

Sweden
+46 31 680 060
info.sweden@moog.com

Turkey
+90 216 663 6020
info.turkey@moog.com

United Kingdom
+44 (0) 1684 858000
Service +44 (0) 1684 278369
info.uk@moog.com
service.uk@moog.com

USA
+1 716 652 2000
info.usa@moog.com
service.usa@moog.com

MOOG
Moog GmbH
Hanns-Klemm-Straße 28
D-71034 Böblingen
Phone +49 7031 622 0
Telefax +49 7031 622 100

www.moog.com/industrial
drives-support@moog.com

Technical alterations reserved.

The contents of our documentation have been compiled with greatest care and in compliance with our present status of information.

Nevertheless we would like to point that this document cannot always be updated parallel to the technical further development of our products.

Information and specifications may be changed at any time.
For information on the latest version please refer to drives-support@moog.com.

ID no.: CA65647-001, Rev. 2.0

Date: Date: 03/2020

Applicable as from firmware version: 124.25