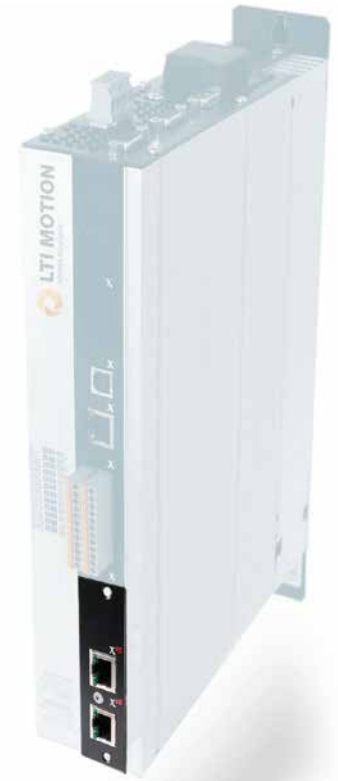# LTI MOTION
KÖRBER SOLUTIONS

# ServoOne
# User Manual

Field bus systems
CANopen and EtherCAT

ServoOne single-axis system
ServoOne multi-axis system
ServoOne junior

CANopen

EtherCAT
Technology Group

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

ServoOne single-axis system
ServoOne multi-axis system
ServoOne junior

Subject to technical change without notice.

The content of our user manual was compiled with the greatest care and attention based on the latest information available to us.

We should point out that this information product will not always be updated simultaneously with the ongoing technical development of our devices.

Information and specifications may change.
For information on the latest version please visit www.lti-motion.com.

# Table of contents

# 1 General information

The product DVD from LTI Motion 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 LTI Motion GmbH is a prerequisite for trouble-free operation and therefore for any claims for defects.

> **NOTE:**
> This user manual applies to the ServoOne single-axis and multi-axis system (referred to in the following as SO8) and the drive controller ServoOne junior (SOj). This document does not replace the operation manuals for the ServoOne and ServoOne junior.

## 1.3 Reference documents

| Document | Contents | ID no. Format |
|---|---|---|
| ServoOne Junior - Operation Manual | Safety, mechanical installation, electrical installation, commissioning, diagnostics, specifications, certification and applicable standards, technical data | 1300.20B.x PDF |
| ServoOne Single-Axis System - Operation Manual | Safety, mechanical installation, electrical installation, commissioning, diagnostics, specifications, certification and applicable standards, technical data | 1100.20B.x PDF |
| ServoOne Multi-Axis System - Operation Manual | Safety, mechanical installation, electrical installation, commissioning, diagnostics, STO, operation with servocontroller as supply, planning, application example, specifications, certification and applicable standards, technical data | 1101.20B.x PDF |
| ServoOne Multi-Axis System Supply Unit - Operation Manual | Safety, mechanical installation, electrical installation, commissioning, diagnostics, specifications, certification and applicable standards, technical data | 1101.21B.x PDF |
| ServoOne Sercos II - User Manual | Safety, commissioning, communication phases, parameter interface, error, warning and status messages, operation modes, weighting, referencing, touchprobe, parameter lists | 1100.29B.x PDF |
| ServoOne 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 | 1108.26B.x PDF |
| ServoOne - System Catalogue | Information, notes on ordering, specifications and technical data on: ServoOne junior, single and multi-axis system, supply units, safety systems, communication, technology, function packages, accessories and motors | 1100.24B.x PDF |
| ServoOne device help | Description of the software functionality of the ServoOne product ranges Firmware versions: <br> - SO junior from V1.30-xx <br> - SO single-axis system from V3.25-xx <br> - SO multi-axis system from V3.25-xx | 0842.26B.x PDF and HTML |
| DriveManager 5 PC user software- online programming help | DriveManager version 5.x. Graphic PC user software for initial commissioning and serial commissioning, operation, diagnostics and project management. | 0842.25B.x context-sensitive help |

*Table 1.1      Documents on the ServoOne 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 | |
|---|---|
| **i** | **NOTE:** <br> Useful information or reference to other documents. |
| **Digit** | **ACTION TO BE TAKEN:** <br> Processing step undertaken by the user or the system. |

*Table 1.2    Pictograms used for instructions and actions to be taken*

## 1.5 Disclaimer

**i** Following the documents on the devices from LTI MOTION GmbH is a prerequisite:
- For safe operation.
- To achieve stated performance features and product characteristics.

LTI MOTION GmbH 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 Helpline/support & GSS

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

Address:          LTI MOTION GmbH
                  Gewerbestrasse 5-9
                  35633 Lahnau

The Helpline is available by e-mail or telephone:

Service hours:    Mo.-Fr.: 8 a.m. - 5 p.m. (CET)
E-mail:           helpline@lti-motion.com
Telephone:        +49 6441 966-180

If you need service assistance, our specialists in Global Sales Support (GSS) will be pleased to be of assistance.

Internet:         www.lti-motion.com → Support & Service → Trouble Ticket

Service hours:    Mo.-Fr.: 8 a.m. - 5 p.m. (CET)
E-mail:           salessupport@lti-motion.com
Telephone:        +49 6441 966-0

**i** **NOTE:**
You will find information on our services on our web site www.lti-motion.com in "Support & Service".

# 1.9    Normative references

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

**IEC 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.

**EN 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

ℹ️ **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

functioning 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. **Electric 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 in which 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 (VBG 4 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. EN 60204 or 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 LTI MOTION:

| DANGER! | Risk of injury due to electrical power! |
|---|---|
| | · **Carelessness will result in serious injuries or death.** <br> 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.** <br> 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.** <br> 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.** <br> Electronic components may become hot during operation! <br> 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.** <br> Do not touch electronic components and contacts! <br> 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 ServoOne single-axis and multi-axis system and ServoOne junior to the situation in the installation.**

| CAUTION! | The user may not be aware of changes to the parameters of the field bus system! |
|---|---|
| | · This situation can result in uncontrolled behaviour of the drive system! <br> 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 ServoOne drive controller

The CANopen communication profile is defined in CiA DS-301. 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 CiA DS-402 (rev. 2.0) for electrical drives was produced based on the communication profile CiA DS-301 (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 CiA DS-402 gives greater independence from the device manufacturers.

In the following sections you will find an overview of the CANopen functionality integrated into the ServoOne drive controller and information on commissioning.

### 3.1.2 System requirements CANopen

| System requirements LTI Motion devices | |
|---|---|
| Devices: | ServoOne SOxx.xxx.xx4x.xxxx.x |
| Standardisation | ISO 11898 / IEC 61800-7 |
| EDS file: | Current "electronic data sheet" for CANopen (device description file for ServoOne drive controller as .eds file). Is available with the firmware in the download area at **www.lti-motion.com** |
| Communication profile | CiA DS-301 (rev. 4.01) |
| Device profile | CiA DS-402 (rev. 2.0) |
| Service tool: | PC user software DriveManager 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 a technology patented by Beckhoff Automation GmbH, Germany. The protocol published in IEC standard IEC 61158 is suitable for hard and soft real-timerequirements in automation technology.

## 3.2.2    EtherCAT functionality of the ServoOne drive controller

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 CiA DS-301 and CiA DS-402 can change to this new field bus technology with little effort.

## 3.2.3    System requirements EtherCAT

| System requirements LTI Motion devices | |
|---|---|
| Devices: | ServoOne SOxx.xxx.xx3x.xxxx.x |
| Standardisation | IEC 61158 / IEC 61784-2 / IEC 61800-7 |
| ESI file: | Current "EtherCAT slave information" (ESI device description file for ServoOne as .xml file). Is available with the firmware in the download area at **www.lti-motion.com** |
| Cable: | Patch cable from CAT 5e SFTP |
| Communication profile | IEC 61158 / ETG. 1000 |
| Device profile | (Rev. 2.0) CiA DS-402 (CoE) / ETG. 2200 |
| Service tool: | PC user software DriveManager 5 |
| Master: | EtherCAT master with support for the above-mentioned communication and device profiles |

*Table 3.2        System requirements EtherCAT*

## 3.3    Other documents

| LTI documentation | ID no. / file formats | Web site |
|---|---|---|
| ServoOne Junior - Operation Manual | 1300.20B.x / PDF | |
| ServoOne Single-Axis System - Operation Manual | 1100.20B.x / PDF | |
| ServoOne Multi-Axis System - Operation Manual | 1101.20B.x / PDF | www.lti-motion.com |
| ServoOne Multi-Axis System Supply Unit - Operation Manual | 1101.21B.x / PDF | |
| ServoOne - online device help | 0842.26B.x / PDF and HTML | |

*Table 3.3        LTI Motion documentation*

| Other field bus documentation | Web site |
|---|---|
| CiA®CiA DS-402 (V4.2.0): CANopen application layer and communication profile | www.can-cia.com |
| CiA® CiA DS-402: 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 IEC61158 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 ServoOne drive controller. The position in the ServoOne junior is identical.
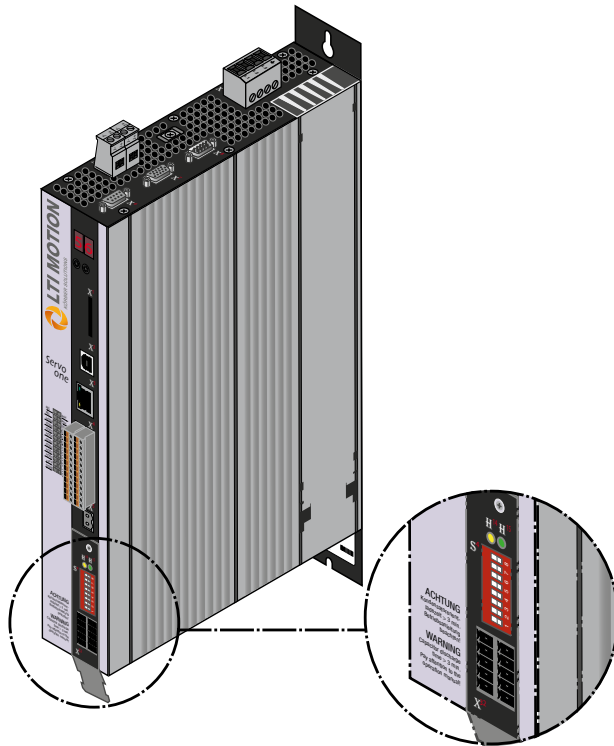


| | NOTE: |
|---|---|
| **i** | The CANopen field bus system is designed as a variant for the product ranges ServoOne junior, single-axis and multi-axis controllers! |

| 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.<br>· Short circuits can cause damage to the device!<br>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 ServoOne drive controllers.

| ServoOne single-axis and multi-axis system (SO8) | ServoOne junior drive controller (SO2) |
|---|---|



*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 **(ServoOne single-axis and multi-axis system)** |
| 4 | S4 | DIP switch for address assignment**(ServoOne junior)** |
| 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 servocontroller:

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 servocontroller).**

2.  **The DIP switch (S4, hardware address).**

3.  **A combination of software CAN address for the servocontroller 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 DriveManager (section "Field bus" --> CANopen/EtherCAT).

You will find more detailed information in the help for the PC user software DriveManager 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 ServoOne drive controller.

### DIP switch ServoOne drive controller (single-axis and multi-axis controller)
- Address setting: switch 1 to 7
- Activate/deactivate the 120 Ω bus terminating resistor: switch 8
- Number of possible addresses: 0 to 127
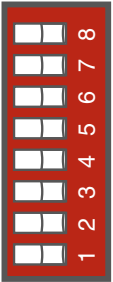
| Significance of the DIP switch on the ServoOne drive controller (single-axis and multi-axis controller) | | | |
|---|---|---|---|
| | DIP switch number | Significance | Address |
|  | 1 | $2^0$ | 1 |
| | 2 | $2^1$ | 2 |
| | 3 | $2^2$ | 4 |
| | 4 | $2^3$ | 8 |
| | 5 | $2^4$ | 16 |
| | 6 | $2^5$ | 32 |
| | 7 | $2^6$ | 64 |
| | 8 | Bus terminating resistor (120 Ω) | |

*Table 4.4        DIP switch ServoOne drive controller (single-axis and multi-axis system)*

### DIP switch ServoOne junior drive controller
- 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 ServoOne junior drive controller | | | |
|---|---|---|---|
| | 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 ServoOne junior drive controller*

### Example for the address setting on the DIP switch:

Setting the address "3":
- Prerequisite:
  Set bus address parameter **P 2005-COM_CAN_Adr** in DriveManager = 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 ServoOne and 7 on the ServoOne junior 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 servocontroller 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.
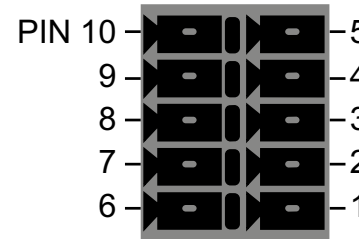
| Terminal block X32 pin assignment | | |
|---|---|---|
|  | | |
| **Functionally identical pins** | **Function** | **Description** |
| 10 and 5 | CAN_+24V | External 24 V supply |
| 9 and 4 | CAN_H | CAN high |
| 8 and 3 | CAN_SHLD | CAN shield (optional) |
| 7 and 2 | CAN_L | CAN low |
| 6 and 1 | 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
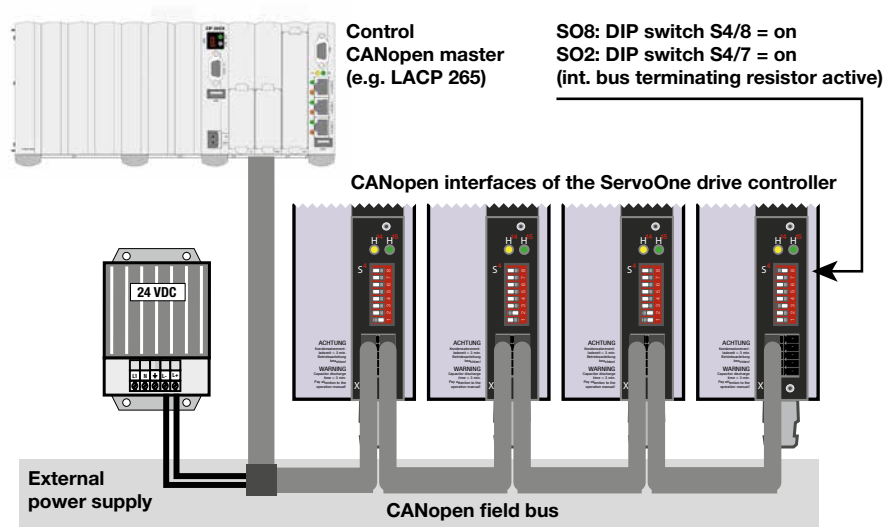


*Figure 4.2 Schematic illustration of the CANopen field bus system*

ℹ️ The CANopen field bus system:
- Is connected to the system via terminal X32.
- Is electrically isolated in relation the drive controller 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) <br> Activated bus terminating resistor: <br> • ServoOne - switch 8 on DIP switch S4 to "ON" <br> • ServoOne junior - 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 drive controller) |
| 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 | Factory setting |
| 500 kBaud | 100 m | |
| 250 kBaud[2] | 250 m | |
| 125 kBaud[2] | 500 m | |
| 50 kBaud[3] | 1000 m | |
| 20 kBaud[3] | 2500 m | |

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 ServoOne drive controller (X4, control connections) (connected to 24 V) (see chapter "4.6.5 Hardware enable" on page 20). | See operation manual "Specification of control connections": • ServoOne junior 1300.00B.x • ServoOne single-axis system 1100.20B.x • ServoOne multi-axis system 1101.20B.x |
| 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 drive controller. | |
| 5. | • Switch on the drive controller. | |

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 SO and SOj):

1. Connect control input for the hardware enable, ENPO on the SO and SOj, on the control connections (X4) to 24 V.
2. Connect control input STO (Safe Torque Off), ISDSH on the SO and SOj, 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 "ServoOne online device help".

> **NOTE:**
> If the control inputs ENPO and ISDSH on the control connections (X4) on the SO or SOj 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 drive controller

A 7-segment display integrated into the control unit for the drive controller (SO and SOj) indicates the device state.

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

| Device state of the drive controller | | | |
|---|---|---|---|
| **D1** | **D2** | **Description** | |
| *8.* | *8.* | Device in reset state | |
| | *0* | 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.<br>• 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 EN 61800-5-2.
2) The dot flashes when the power stage is active.

---

> **i** **NOTE:**
> You will find detailed information on the device state of the drive controller (SO and SOj) in the operation manuals and device help for the ServoOne 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" |

> **i** **NOTE:**
> For information on fault numbers and fault locations refer to the fault list in the "ServoOne – online device help" (available as PDF file and in HTML format in the download area at LTI MOTION).

# 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 ServoOne drive controller. The position of the EtherCAT field bus system in the ServoOne junior is identical.
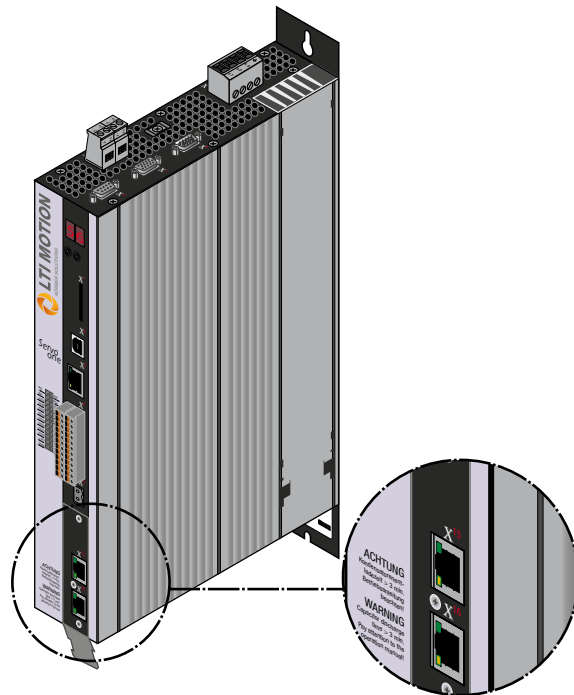


| ![i] | **NOTE:**<br>The EtherCAT field bus system is designed as a variant for the product ranges ServoOne junior, single-axis and multi-axis controllers! |
|---|---|

| 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.<br>• Short circuits can cause damage to the device!<br>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 ServoOne drive controllers.

| EtherCAT for the ServoOne drive controller (single-axis and multi-axis controller) | EtherCAT for the ServoOne junior drive controller |
|---|---|
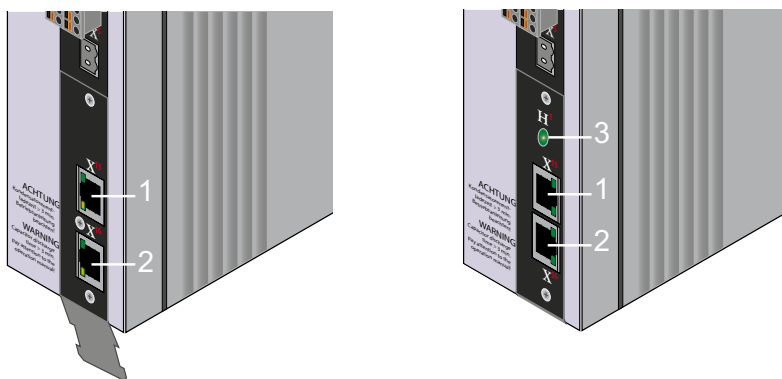


*Figure 5.1     Layout connections and controls EtherCAT*

| Connection | Designation | Function |
|---|---|---|
| 1 | X15 | RJ-45 socket - input |
| 2 | X16 | RJ-45 socket - output |
| 3 | H1 | Network status LED - RUN/ERROR (**only ServoOne junior**) |

*Table 5.1     Connections for the EtherCAT field bus system*
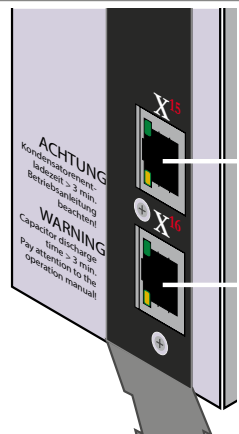
**Note:**
On the device series ServoOne junior the connectors X15/X16 are rotated by 180° compared to the device series single and multi-axis system.

## 5.3    Light emitting diodes - and field bus system flashing codes

### 5.3.1    ServoOne (single-axis and multi-axis controller)

Flashing codes on the LEDs on the RJ-45 sockets for the EtherCAT field bus system (ServoOne - single-axis and multi-axis controller, without additional LED):

| LED | Status EtherCAT - network status - NMT | | Figure |
|---|---|---|---|
| Top LED (green) | Link/ Activity | **Off = No link**<br>・ No connection to another bus user<br><br>**On = Link**<br>・ Connection to another bus user<br>・ No data exchange<br><br>**Flashing = Activity**<br>・ Data exchange active |  |
| Bottom LED (yellow)* | RUN | **Off = Initialisation**<br>・ Device is in the "Initialisation" state<br><br>**Flashing = Pre-Operational**<br>・ Device is in the "Pre-operational" state<br><br>**Single flash = Safe-Operational**<br>・ Device is in the "Safe-operational" state<br><br>**On = Operational**<br>・ Device is ready | |
| **\*** ... Is only active in the lower socket (output) if a further bus user is connected. | | | |

*Table 5.2     Flashing codes on the LEDs for the EtherCAT field bus system on the ServoOne (single-axis and multi-axis system)*

ServoOne junior

**The flashing codes on the LEDs on the RJ-45 sockets (ServoOne junior, with
additional LED - RUN/ERROR H1) indicate the status of the EtherCAT field bus system:**

| LED | | EtherCAT status - network status | Figure |
|---|---|---|---|
| Top LED (green) | Link/ Activity | **Off = No link**<br>・ No connection to another bus user | |
| | | **On = Link**<br>・ Connection to another bus user<br>・ No data exchange | |
| Bottom LED (green) | Link (PHY) | **Off = No link**<br>・ No connection to another bus user | |
| | | **On = Link**<br>・ Connection to another bus user<br>・ No data exchange | |
| Status LED **H1** (only SOj) | **RUN/ ERROR** | **Red = ERROR**<br><br>**Off = No ERROR**<br><br>**Flashing = Invalid configuration**<br><br>**Single flash = Local ERROR**<br><br>**Double flash = Watchdog timeout**<br><br>**Green = RUN**<br><br>**Off = Initialisation**<br>・ Device is in the "Initialisation" state<br>**Flashing = Pre-Operational**<br>・ Device is in the "Pre-operational" state<br>**Single flash = Safe-Operational**<br>・ Device is in the "Safe-operational" state<br>**On = Operational**<br>・ Device is ready | |

*Table 5.3      Flashing codes for the EtherCAT field bus system on the ServoOne junior*

## 5.4     Pin assignment of the RJ-45 socket

EIA/TIA-568A and EIA/TIA-568B are standards for 8-pin RJ-45 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.
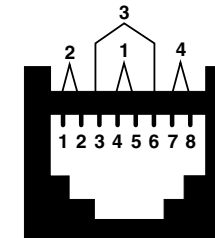


| Pin | EIA/TIA-568B Core pair number | Function | EIA/TIA-568B colour |
|---|---|---|---|
| 1 | 2 | Tx Data+ | White/orange dash |
| 2 | 2 | Tx Data- | Orange/white dash or orange |
| 3 | 3 | Rx Data+ | White/green dash |
| 4 | 1 | Not used | Blue/white dash or blue |
| 5 | 1 | Not used | White/blue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

*Table 5.4      Pin assignment for the RJ-45 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:



**EtherCAT Master (LACP 265)**
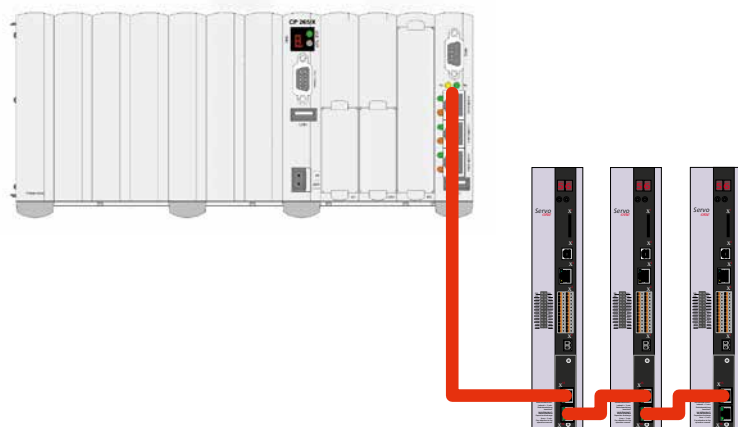
*Figure 5.2     EtherCAT topology*

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, RJ-45) and one output port (X16, RJ-45).
- 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.<br>For this reason:<br>· Have the EtherCAT field bus system installed by authorised personnel.<br>· Follow the procedure described and contact the Helpline at LTI MOTION GmbH if necessary. |

**ℹ️ 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:<br>· Have the EtherCAT field bus system installed by authorised personnel.<br>· Only use certified component(s) for an EtherCAT infrastructure.<br>· 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 SO and SOj):

1. Connect control input for the hardware enable, ENPO on the SO and SOj, on the control connections (X4) to 24 V.

2. Connect control input STO (Safe Torque Off), ISDSH on the SO and SOj, 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 "ServoOne online device help".

> **NOTE:**
> If the control inputs ENPO and ISDSH on the control connections (X4) on the SO or SOj 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 drive controller

A 7-segment display integrated into the control unit for the drive controller (SO and SOj) indicates the device state.

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

| Device state of the drive controller | | | |
|---|---|---|---|
| D1 | D2 | Description | |
| 8. | 8. | Device in reset state | |
| | 0 | 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) |

*Table 5.5    State indication, drive controller*

| Device state of the drive controller | | | |
|---|---|---|---|
| D1 | D2 | Description | |
| | 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.<br>· 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 EN 61800-5-2.
2) The dot flashes when the power stage is active.

*Table 5.5    State indication, drive controller*

> **NOTE:**
> You will find detailed information on the device state of the drive controller (SO and SOj) in the related operation manual.

## Example for a sequence of flashing

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

| | Fault: | Er = "Error" |
|---|---|---|
| | Fault number: | 02 = "Error in the parameter list" |
| | 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 "ServoOne – online device help" (available as PDF file and in HTML format in the download area at LTI MOTION).

# 6 Commissioning and configuration of the CANopen field bus system

## 6.1 PC user software DriveManager 5

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

| Technical data | DriveManager 5 |
|---|---|
| Support for the following functions | • Initial commissioning of your:<br>  • Drive system<br>  • CANopen device network |
| | • Quick serial commissioning using a configurable commissioning file (contains firmware, data set, iPLC program). |
| | • Operation and diagnostics, including using Cockpit and 6-channel oscilloscope. |
| | • Project management. |

*Table 6.1       Technical data DriveManager 5*

> **NOTE:**
> You will find more detailed information on installing and starting the PC user software  **DriveManager 5** in the "DM5 Installation Manual"
> (available as a PDF file in the download area at http://www.lti-motion.com).
>
> 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 ServoOne online device help for the related device (available as PDF files and html files in the download area at http://www.lti-motion.com).

## 6.2 Electronic device data sheet – EDS file

The EDS file is required to operate the drive controller in the CANopen device network.

The electronic data sheet (EDS):

- Is installed in the "CANopen master" (controller). Is used to integrate ServoOne drive controllers in the CANopen device network.
- Is available as a download (eds file, Firmware area) at http://www.lti-motion.com.

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

## 6.3    Commissioning and configuration

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

1. Operation manual for your drive controller.
2. ServoOne – online device help.
3. ServoOne – DriveManager 5 application help (implemented as context-sensitive help in DM5).
4. User Manual CANopen/EtherCAT.

### 6.3.1    Procedure for commissioning and configuration

| | |
|---|---|
| **1.** | Connect, wire and configure the drive controller and additional components as per the operation manual. |
| **2.** | Supply the drive controller with 24 V control voltage. |
| **3.** | Undertake the initial commissioning of the drive controller as per the online device help (power stage, motor, encoder, control, etc.). Note the following points:<br>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 drive controller. |
| **5.** | Save settings in the drive controller. |
| **6.** | Restart drive controller. |
| **7.** | Set-up CANopen network as per chapter "4.6.1 System connection of the CANopen field bus system" on page 19. Note the following points:<br>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:<br>Make sure the CANopen master transfers the mapping to the drive controller on boot-up! |
| **9.** | Switch on supply of power (mains). |
| **10.** | Select the operation mode for the drive controller via SDO access to object 0x6060 (Modes of operation). Supported operation modes are listed in chapter "10 CiA DS-402 functionality implemented in the drive controller" on page 53. |
| **11.** | Test quality of control and optimise if necessary. |

*Table 6.2        Commissioning the CANopen field bus system*

## 6.4    CANopen operation via the CiA DS-402 function implemented in the drive controller

The drive controller operation mode is selected via the CiA DS-402 function "Modes of operation" implemented. (See chapter "10 CiA DS-402 functionality implemented in the drive controller" on page 53 and chapter "11 Operation of the ServoOne drive controller via CiA DS-402" on page 63).

The individual configurations for the operation of the drive controller 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 drive controller are configured with motor data, control settings, I/O configurations and other functions.

The operation mode is selected via the **CiA DS-402 object 0x6060 (Modes of operation)**. You will find the operation modes supported by the drive controller in chapter "10 CiA DS-402 functionality implemented in the drive controller" on page 53 and chapter "11 Operation of the ServoOne drive controller via CiA DS-402" on page 63.

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

1. Position control.
2. Velocity control.
3. Torque control.
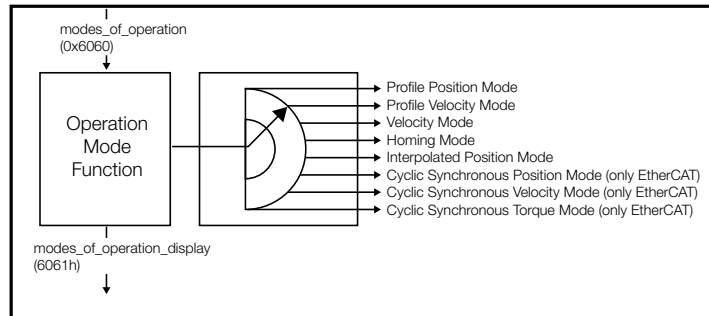
You can change between different operation modes. You will find the operation modes supported in object 6502h (supported Drive Modul).

## 6.4.2 Setting timing parameters

Three timing parameters are responsible for the communication between the drive controller 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 Set software address and baud rate via DriveManager.

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 chapter "4.4 Address assignment CANopen (node ID)" on page 16 |
| P 2006 COM_CAN_Adr | Baud rate CANopen | Permissible baud rates, see chapter "4.6.3 Transmission speeds of the CANopen field bus system (baud rates)" on page 20 |

*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 ServoOne drive controller: 1000 kbit. Changes to the baud rate in parameter **P 2006 COM_CAN_Baudrate** are only applied after restarting the drive controller.
3. Parameter **P 2059 COM_CAN_Baudrate_Act**. indicates the actual baud rate.

## 6.5.2    Commissioning instructions

**i**

**NOTE:**
Reasons why a drive controller does not respond to a telegram:

- The telegram frame (baud rate, data width) from the controller (CANopen master) is erroneous.
- A drive controller in CANopen device network has the wrong bus address or several drive controllers have the same bus address.
- Incorrectly established serial connection between controller (CANopen master) and drive controller.
- The 24 V supply on the CANopen connection is missing or the wiring is faulty.
- The drive controller is not in the NMT network state "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 6.5        CANopen network states*

## 6.5.3    System test on the controller (CANopen master)

Activating modified settings:

- Switch off and on drive controller.
- 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 drive controller configuration data (e.g. before "Resetting to factory setting", see next chapter):

1. Via DriveManager 5 (see PC user software DriveManager 5 - online application 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 DS-301.

**i**

**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 drive controller 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 drive controller's data.
   – The change is effective after a "Reset node" command or device restart.

2. Via DriveManager 5:
   – Select the corresponding drive controller in the tree structure in DM.
   – Save drive controller's data in a file (see "DriveManager application 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 drive controller 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 drive controller during the initialisation/restart! |
|---|---|
| ⚠️ | • There is a risk of loss of data and settings.<br>Wait until the drive controller has completed the initialisation/the restart (10 s)! |

# 7 Commissioning and configuration of the EtherCAT field bus system

## 7.1 DriveManager 5

**DriveManager 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.

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

*Table 7.1        Overview of DriveManager 5 functions*

> **i** **NOTE:**
> You will find more detailed information on installing and starting the PC user software
> **DriveManager 5** in the "DM5 Installation Manual"
> (available as a PDF file in the download area at http://www.lti-motion.com).
>
> 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 ServoOne online device help for the related device (available as PDF files and html files in the download area at http://www.lti-motion.com).

## 7.2 ESI (xml) – device description file

A prerequisite for the operation of the drive controller in the EtherCAT network is a 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 (ESI file (xml), Firmware area) at http://www.lti-motion.com.

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

## 7.3 Commissioning and configuration

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

1. Operation manual for your drive controller.
2. ServoOne – online device help.
3. ServoOne – DriveManager 5 application help (implemented as context-sensitive help in DM5).
4. User Manual CANopen/EtherCAT.

## Procedure for commissioning and configuration

**1.** Connect, wire and configure the drive controller and additional components as per the operation manual.

**2.** Supply the drive controller with 24 V control voltage.

**3.** Undertake the initial commissioning of the drive controller as per the online 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 drive controller (page 37).

**5.** Implement scaling and units in the drive controller similar to scaling in the master.
See chapter "10.3 Units and scaling, factor group" on page 58.

**6.** Save settings in the drive controller.

**7.** Restart drive controller.

**8.** Set-up EtherCAT network as per chapter "5.5.1 System connection of the EtherCAT field bus system" on page 26.

**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 drive controller on boot-up!

**10.** Switch on supply of power (mains).

**11.** Select the operation mode for the drive controller via SDO access to object 0x6060 (Modes of operation). Supported operation modes are listed in chapter "10 CiA DS-402 functionality implemented in the drive controller" on page 53.

**12.** Test quality of control and optimise if necessary.

*Table 7.2        Commissioning the CANopen field bus system*

## 7.4      EtherCAT operation via the CiA DS-402 function implemented in the drive controller

### 7.4.1      Selection of the operation mode (Modes of operation)

The operation mode is selected via the **CiA DS-402 object 0x6060 (Modes of operation)**. You will find the operation mode supported by the ServoOne drive controller in chapter "10 CiA DS-402 functionality implemented in the drive controller" on page 53 and chapter "11 Operation of the ServoOne drive controller via CiA DS-402" on page 63.

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

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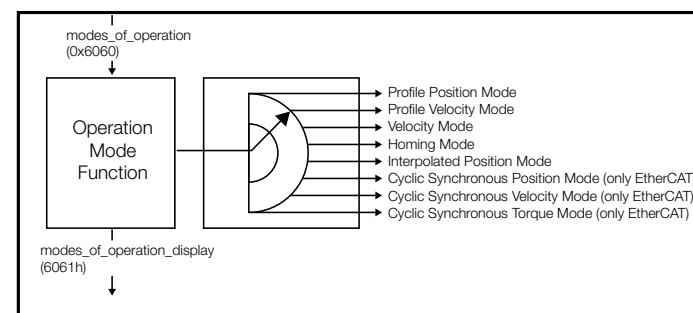


You can change between different operation modes. Please only select operation modes supported by your drive controller.

## 7.4.2    Setting timing parameters

Three timing parameters are responsible for the communication between the ServoOne drive controller and the 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 2266<br>Index 0<br>Index 1 | 0x60C2<br>Interpolation time base<br>Interpolation time exponent | s |

*Table 7.3    3 timing parameters*

**i**  **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

**i**  **NOTE:**
Reasons why a drive controller does not respond to a telegram:

The drive controller 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 drive controller.
- Wait for initialisation time of a few seconds to elapse.

### 7.5.3    Saving parameter settings

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

1.  Via DriveManager 5 (see PC user software DriveManager 5 - online application help).

2.  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.

3.  Another method is the **object 1010h Store parameters** that is specified in the DS-301.

> **i**  **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 drive controller 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 drive controller's data.
    – The change is effective after a "Reset node" command or device restart.

2.  Via DriveManager 5:
    – Select the corresponding drive controller in the tree structure in DM.
    – Save drive controller's data in a file (see "DriveManager application help")
    – On the menu bar under "Active device", select the command "Reset to factory settings".

> **i**  **NOTE:**
> After the "Reset to factory settings", in both cases it takes around 10 s for the drive controller 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 drive controller during the initialisation/restart! |
|---|---|
| ⚠️ | • There is a risk of loss of data and settings.<br>Wait until the drive controller has completed the initialisation/the restart (10 s)! |

# 8 Configuring the parameters for the devices - CANopen

## 8.1 CiA DS-301 functionality implemented

The CIA DS-301 profile offers a collection of CAN communication services, without exactly defining the application. The application-specific drive profile CiA DS-402 is based on these CAN communication services.

### 8.1.1 CAN communication services in the DS-301 profile

- Boot-up according to DS-301 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 CiA DS-402 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 DS-301.
- 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 CiA DS-301 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 (DS-301, CiA DS-402 and manufacturer-specific) for the ServoOne drive controller in the eds file ("Electronic data sheet", available as a download in the "Firmware" area at www.lti-motion.com).

The following table gives an overview with important DS-301 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 drive controller.
- Utilise the "multiplexed domain transfer protocol" in the CAL specification. The "multiplexed domain transfer protocol" transfers data of any length.

The drive controller 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":
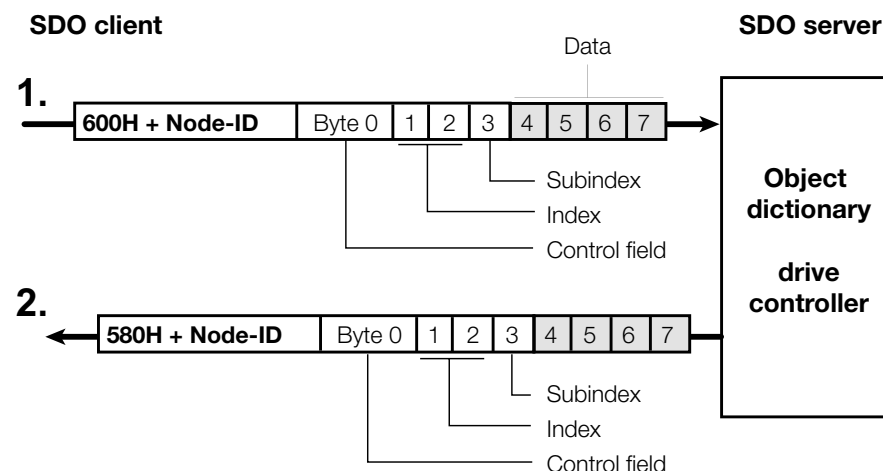


The CAL specification differentiates between 3 types of protocol services:

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 DS-301).

## 8.2.1    Parameter data types

**NOTE:**
– Some parameter settings are indicated as value substitution text by the software "DriveManager".
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 DriveManager 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*<br>USIGN16 / INT16*<br>USIGN32 / INT32 | Low word<br>Low byte | Low word<br>High byte | High word<br>Low byte | High word<br>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 DS-301 and CiA DS-402) the CANopen profile has an area defined for manufacturer-specific entries (device parameters). This area is between **2000 h and 5FFF h** (DS-301). 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:**
> DS-301 objects (1000h ... 1FFFh) and CiA DS-402 (6000h ... 9FFFh):
> - Can also be addressed by the master using the object numbers, even if they are displayed in DriveManager using device parameter numbers.
> - Should not be addressed using the manufacturer-specific objects (not recommended).

Example:

The object 1000h – Device Type exists in the DS-301 profile and as 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 DS-301).

### 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 ServoOne drive controller.
- The string to be transmitted "X-Axis".
- The hexadecimal entries for the numerical values.

| TIME | ID | Direction | DLC | byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Comment |
|------|----|-----------|-----|---|---|---|---|---|---|---|---|---------|
| 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 |

| TIME | ID | Direction | DLC | byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Comment |
|------|----|-----------|-----|---|---|---|---|---|---|---|---|---------|
| 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 ServoOne drive controller:

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

On each transmission of a parameter, the drive controller 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 drive controller 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 drive controller:

| Step | Action |
|------|--------|
| 1. | Notify download without logic check, for this purpose<br>• 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 drive controller.<br>Attention: Basic check mechanisms are still active, despite the disabled logic check.<br>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<br>• After the transmission of the parameter data set,<br>set parameter 11[4] PARA_SetCmd Register upload to the value 0.<br>• The logic check is activated.<br>• 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!<br>Download the parameter data set:<br>• With drive system stationary and switched off!<br>• 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 DS-301 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 drive controller) and
•   TxPDOs (process data sent by the drive controller).

The drive controller 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 drive controller 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 ServoOne drive controller can be defined as an event:
- The default settings in the ServoOne drive controller 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 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 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 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 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 drive controller can be mapped variably.
- The mapping is as per the definitions in the CANopen communication profile DS-301.
- 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 ServoOne drive controller.

## 8.3.3    Mapping notes

**NOTE:**
The ServoOne drive controller does not have any mapping selectors. The PDOs do not contain any mapping with the factory setting for the ServoOne drive controller (mapping = 0).This means that before communication with PDOs takes place:
- The mapping must be written to the drive controller via the controller,
- Or the data set must be transmitted.
  - The ServoOne drive controller 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 V4.15-02 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 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 drive controller!

## 8.4.2 Heartbeat

The ServoOne drive controller:

- Supports the heartbeat function according to DS-301. The heartbeat function operates as per the states and the state transitions of the network nodes (NMT state machine).
- During this process it is used as the **heartbeat producer**. This means it sends the heartbeat protocols to the controller via the **object 1017 h Producer Heartbeat Time**.

### Principle of operation of the heartbeat protocol

The **heartbeat producer** (drive controller) 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 drive controller 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**).



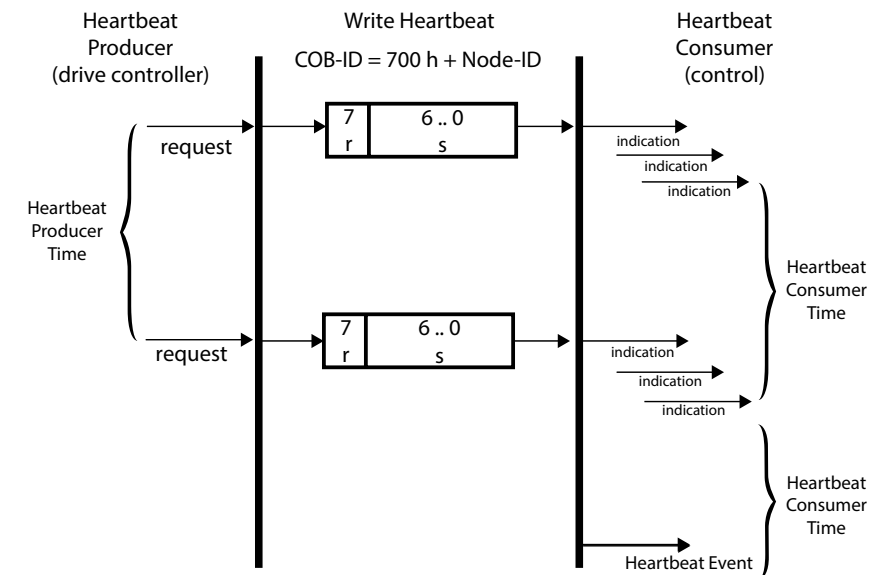*Figure 8.1    Heartbeat protocol*

| | | Bit 0 ... 6 | NMT - network state |
|---|---|---|---|
| r: | Reserved (bit 7, within 0) | 0 | Bootup |
| s: | Status of the heartbeat producer | 4 | Stopped/Safe OP |
| | | 5 | Operational |
| | | 127 | Pre-Operational |

## 8.5     Telegram failure monitoring

The drive controller 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:<br>· 0 = Sync<br>· 1 = RxPDO |
| P 2062 | COM_CAN_Timeout_Value | Enter value:<br>· Timeout [ms]<br>· 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

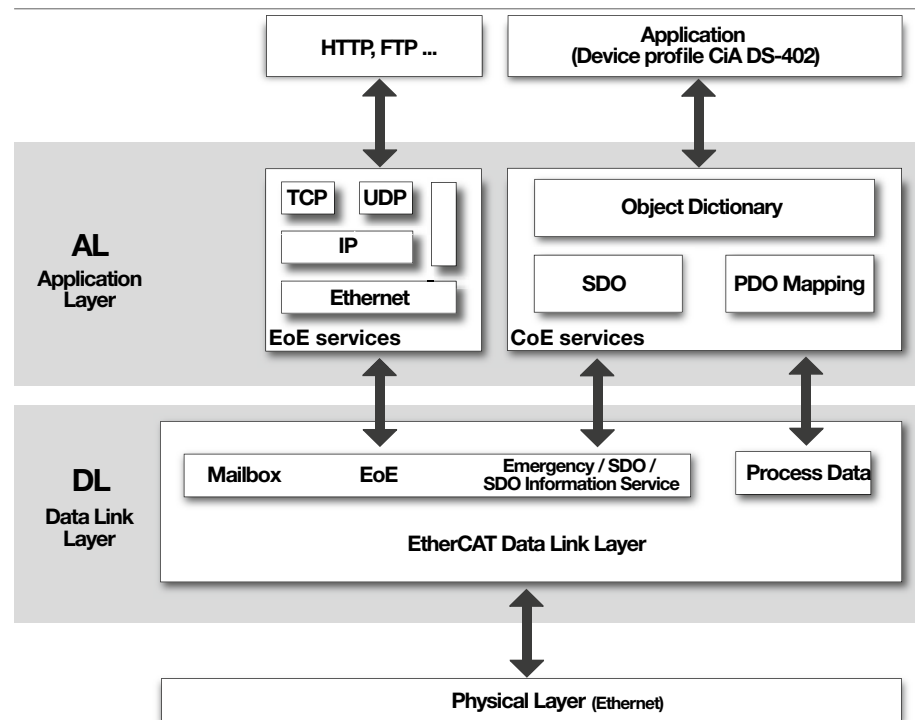The illustration below shows the structure of the EtherCAT technology based on the OSI 7-layer model:



*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 ServoOne drive controller (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 ServoOne drive controller supports, as per the CiA DS-402 device profile, "error codes". For content and structure see chapter "13 "Emergency messages"" on page 79

**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 ServoOne drive controller. The EtherCAT master receives information on objects supported by the EtherCAT slave such as:
  – Data types
  – Access options, etc.
- In the ServoOne drive controller 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 ServoOne drive controller:
  – 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 download area at LTI Motion GmbH.

**Process data (DS-301)**

- 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 DS-301 (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).

> **i** **Please note!** (From firmware V4.15-02 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 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 drive controller.
- The ServoOne drive controller supports:
  – EoE services (Ethernet over EtherCAT).
  – CoE services (CAN over EtherCAT).

## Application

- The ServoOne drive controller utilises the device profile CiA DS-402

## 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.
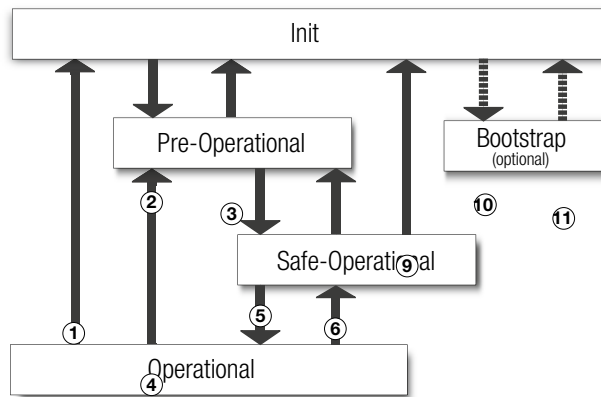


*Figure 9.2     EtherCAT NMT state diagram*

| 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.<br>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.<br>・ Update the slave firmware via the "Bootstrap" state.<br>・ 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"<br>・ Stop "Input Update"<br>・ Stop "Mailbox Communication" |
| 2 | (IP) | ・ Start "Mailbox Communication" |
| 3 | (PI) | ・ Stop "Mailbox Communication" |
| 4 | (OP) | ・ Stop "Output Update"<br>・ 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"<br>・ Stop "Mailbox Communication" |
| 10 | (IB) | ・ Start "Bootstrap Mode" |
| 11 | (BI) | ・ Restart Device |

*Table 9.2     Description of the EtherCAT NMT state transitions*

# 10 CiA DS-402 functionality implemented in the drive controller

The drive controller supports operation modes and their functions according to the CiA DS-402 device profile.

The operation modes according to CiA DS-402 implemented in the ServoOne drive controller:

| No. | Operation modes with profile generation in the drive controller | |
|---|---|---|
| 1 | Profile position mode | Page 67 |
| 2 | Velocity mode (U/F operation) | Page 70 |
| 3 | Profile velocity mode | Page 71 |
| 4 | Reserved | - |
| 5 | Reserved | - |
| 6 | Homing mode | Page 72 |

| No. | Operation modes with profile generation in the controller | |
|---|---|---|
| 7 | Interpolated position mode | Page 73 |
| 8 | Cyclic synchronous position mode (only EtherCAT) | Page 75 |
| 9 | Cyclic synchronous velocity mode (only EtherCAT) | Page 76 |
| 10 | Cyclic synchronous torque mode (only EtherCAT) | Page 77 |

## 10.1 Device control and state machine

The drive controller is controlled via the DRIVECOM state machine defined in CiA DS-402.

### 10.1.1 General information

The DEVICE CONTROL FUNCTION monitors all functions of the drive controller and is divided into:
- Operation of the state machine **(State machine)**
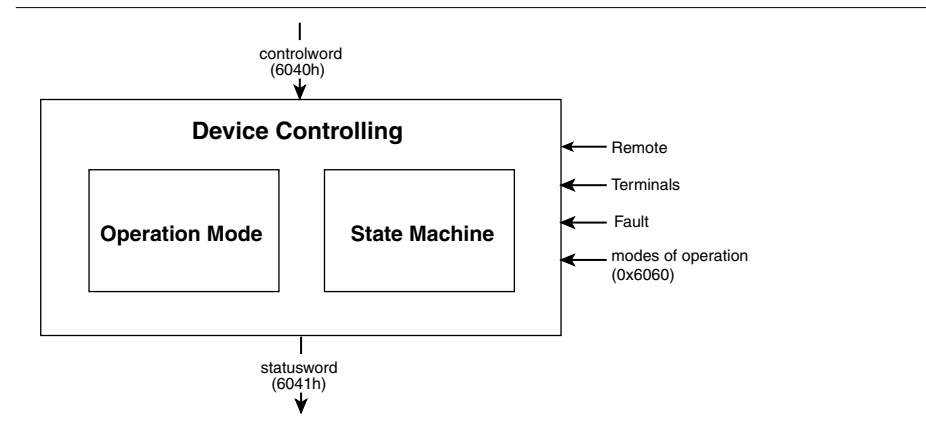- Operation mode-specific functions **(Operation mode)**



*Figure 10.1 Device controlling*

The state of the drive controller:
- 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 drive controller.

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 drive controller
- 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.
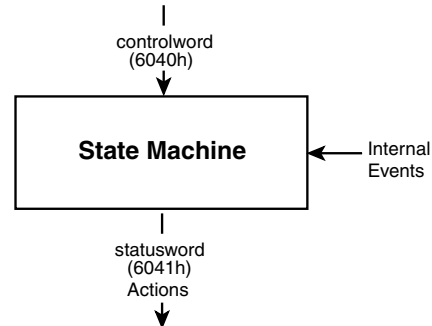
controlword
(6040h)

**State Machine** ← Internal Events

statusword
(6041h)
Actions

*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 drive controller (e.g. control word).
- The bus system used.

The current state is indicated in the status word. During operation the drive controller differentiates between:
- Standstill.
- Operation.
- Fault states.

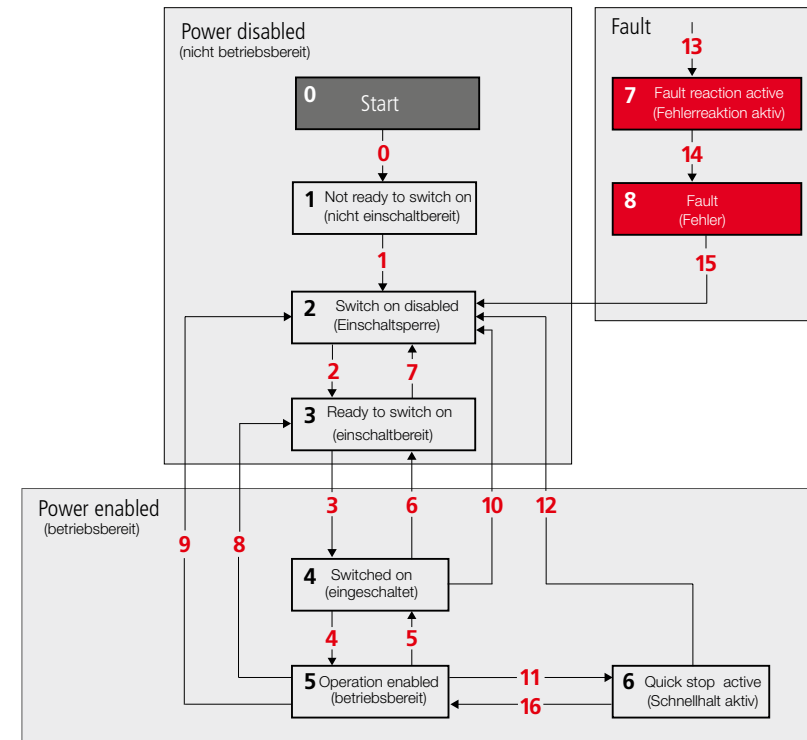## 10.1.3    DRIVE COM state machine schematic



*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 drive controller.<br>• The drive controller is initialised or self-test is in progress.<br>• If fitted, the brake is applied.<br>• Drive function inactive. |
| 2 | **Switch on disabled** | • Initialisation of the drive controller is completed.<br>• Drive controller parameters setup and can be changed.<br>• No electrical power is present in the DC link (230 V, e.g. for safety).<br>• Drive function is deactivated.<br>• STO (Safe Torque Off) and ENPO (Enable Power) inactive. |
| 3 | **Ready to switch on** | • DC link is electrically live.<br>• Drive controller parameters can be changed.<br>• Drive function is deactivated. |
| 4 | **Switched on** | • DC link is electrically live.<br>• Drive controller parameters can be changed.<br>• Power amplifier is ready.<br>• Drive function is deactivated. |
| 5 | **Operation enabled** | • The drive controller has not detected any errors.<br>• Drive function active and power supply present at motor.<br>• Drive controller parameters can be changed.<br>  (Relates to default application for the drive controller) |
| 6 | **Quick stop active** | • Drive controller parameters can be changed.<br>• Reaction is implemented as per quick stop option code<br>• Drive function active and power supply present at motor.<br>• 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.<br>• Fault has occurred in the drive controller.<br>• The fault reaction as per the fault reaction code is implemented<br>• Drive function active and power supply present at motor. |
| 8 | **Fault** | • Drive controller parameters can be changed.<br>• Fault has occurred in the drive controller.<br>• The removal of the supply of electrical power is dependent on the application.<br>• Drive function is deactivated. |

You will find more detailed information on the indication of the drive controller device state in chapter "4.7 Device state of the drive controller" on page 21 and in chapter "5.6 Device state of the drive controller" on page 27.

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

| No. | State transitions | Description |
|---|---|---|
| 9 | **Operation enabled -->**<br><br>**Switch on disabled** | Situation:<br>"Disable voltage" command received from the host. |
| | | Action:<br>The power section is shut down directly, the unbraked motor can rotate freely. |
| 10 | **Switched on -->**<br><br>**Switch on disabled** | Situation:<br>"Quick stop" or "Disable voltage" command received from the host. |
| | | Action:<br>The power section is shut down directly, the unbraked motor can rotate freely. |
| 11 | **Operation enabled -->**<br><br>**Quick stop active** | Situation:<br>"Quick stop" command received from the host. |
| | | Action:<br>"Quick stop" function is active. |
| 12 | **Quick stop active -->**<br><br>**Switch on disabled** | Situation:<br>"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:<br>The power section is switched off. |
| 13 | **All states -->**<br>**Fault reaction active** | Situation:<br>"Fault" has occurred in the drive controller. |
| | | Action:<br>Implement corresponding fault reaction. |
| 14 | **Fault reaction active -->**<br><br>**Fault** | Situation:<br>Fault reaction complete. |
| | | Action:<br>The drive controller is deactivated. The power section may be switched off. |
| 15 | **Fault -->**<br><br>**Switch on disabled** | Situation:<br>"Fault reset" command received from the host. |
| | | Action:<br>The drive controller runs through the "Fault reset" until the fault is rectified. When the "Fault" state is left, the host deletes the "Fault Reset" bit. |

| No. | State transitions | Description |
|---|---|---|
| 16 | **Quick stop active -->**<br>**Operation enabled** | Situation:<br>"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 page 57) |
| | | Action:<br>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*
*X ... 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 Figure "State machine schematic" on page 54 ) |
|---|---|---|---|---|---|---|
| | 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 | **Stop option code** (Stop 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 | Stop | 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 (P154) -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 |
| P154: 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 **DriveManager 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 **CiA DS-402 factor group**.

**Note:**
The drive controller 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 **DriveManager 5**.
2.    Via the **field bus system**.

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

**Note:**
You will find practical examples for scaling in the "ServoOne online device help".

### 10.3.1   Factor group objects as per CiA DS-402:

| 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 |

*Table 10.4      Factor group objects*

| Object | Object name | Object code | Type | Access |
|---|---|---|---|---|
| 0x6092 | **Feed constant** | ARRAY | Unsigned32 | rw |
| 0x6093 | **Position factor** | ARRAY | Unsigned32 | rw |
| 0x6094 | **Velocity encoder factor** | ARRAY | Unsigned32 | rw |
| 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 **DriveManager**. 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.

$$\text{Velocity encoder resolution} = \frac{\text{Encoder} \quad \dfrac{\text{Increments}}{\text{Second}}}{\text{Motor} \quad \dfrac{\text{Revolution}}{\text{Second}}}$$

### Object 6091h: Gear ratio

Defines the gear ratio of a gearbox on the motor.

### Object 6091h: Gear ratio

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

### 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).

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

### Object 6094h: Velocity encoder factor

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

$$\text{Velocity encoder factor} =$$

$$\frac{\text{Velocity encoder resolution} \cdot \text{Gear factor} \cdot \text{Position unit} \cdot F_{\text{Velocity (Notation index)}}}{\text{Feed constant} \cdot \text{Velocity unit} \cdot \text{Second} \cdot F_{\text{Positon (Notation index)}}}$$

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).

**Object 6097h: Acceleration factor**

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

**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 CiA DS-402 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 drive controller (control initialisation)!

## 10.4   I/O image

The following objects for the drive controller 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 CiA DS-402.
- 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 drive controller.
- 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 drive controller.
- 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   CiA DS-402 functionality implemented in the drive controller

### 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 122 - 124, 126):

| 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 (122 - 124, 126)

### 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 ServoOne drive controller via CiA DS-402

## 11.1 Parameter configuration

For operation via the CANopen and EtherCAT field bus systems (as per CiA DS-402 device profile), set the following parameters:

| Parameter no. | Name | Function | Setting |
|---|---|---|---|
| P 0159 | MPRO_CTRL_SEL | Control location selector | CiA DS-402 |
| P 0165 | MPRO_REF_SEL | Setpoint selector | CiA DS-402 |

*Table 11.1      Configuring ServoOne 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 drive controller expects setpoints from a higher lever controller in the parameter CON_IpRefTs.

Adjust the cycle time of the drive controller (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 ServoOne 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 | 0 | |
| 6 | | | |
| 7 | Fault reset | M | |
| 8 | Stop | 0 | |
| 9 | Reserved | 0 | |
| 10 | | | |
| 11 | Manufacturer specific | 0 | |
| 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 | Stop |

*Table 11.5     Operation mode-specific bits in the control word*

| Operation mode | Bit | | | |
| | 4 | 5 | 6 | 8 |
|---|---|---|---|---|
| Profile velocity mode | Reserved | Reserved | Reserved | Stop |
| Homing mode | Homing operation start | Reserved | Reserved | Stop |
| Interpolated position mode | Enable IP mode | Reserved | Reserved | Stop |
| 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 drive controller.

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 controller.

| 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 DriveManager 5 in the following parameters:

- P 34[0] = Actual status word device warning
- P 33[0-8] = Actual fault/actual warning = message in DM5

## 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 CiA DS-402 CANopen/EtherCAT (CiA DS-402(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 CiA DS-402").

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 CiA DS-402".

# 12 Operation modes

## 12.1 DS-402 compatible operation modes

The operation mode is switched over in a drive controller 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 CiA DS-402 device profile:

| no. | Operation modes with profile generation in the drive controller | |
|:---:|---|:---:|
| 1 | Profile position mode | Page 67 |
| 2 | Velocity mode (U/F operation) | Page 70 |
| 3 | Profile velocity mode | Page 71 |
| 4 | Reserved | - |
| 5 | Reserved | - |
| 6 | Homing mode | Page 72 |

*Table 12.1    Operation modes with profile generation in the drive controller*

| no. | Operation modes with profile generation in the controller | |
|:---:|---|:---:|
| 7 | Interpolated position mode | Page 73 |
| Manufacturer-specific operation modes (not defined in CiA DS-402): | | |
| 8 | Cyclic synchronous position mode (only EtherCAT) | Page 75 |
| 9 | Cyclic synchronous velocity mode (only EtherCAT) | Page 76 |
| 10 | Cyclic synchronous torque mode (only EtherCAT) | Page 77 |

*Table 12.2    Operation modes with profile generation in the controller*

## 12.2 Operation modes with profile generation in the drive controller

For the operation modes with profile generation in the drive controller, 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 drive controller 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
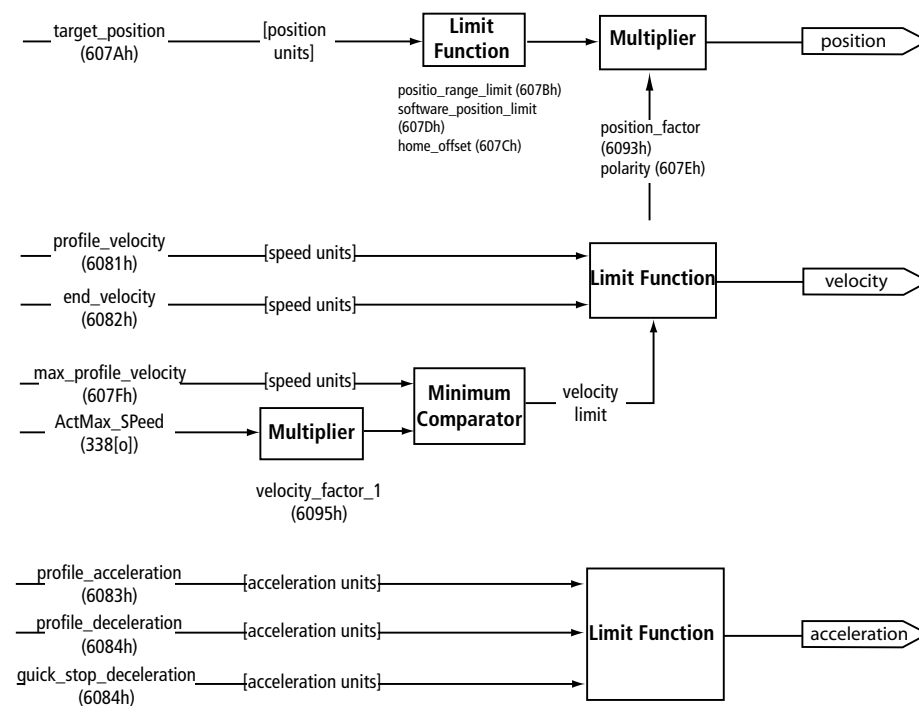


*Figure 12.1    Structure of profile position mode*

## 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 | Stop | 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 | Stop = 0: Target position not reached<br>Stop = 1: Axis is braking |
| | | 1 | Stop = 0: Target position reached<br>Stop = 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 745 MON_REFWINDOW |

*Table 12.5      PROFILE POSITION MODE bits in the status word*

## Setpoint transfer from a master (host computer)
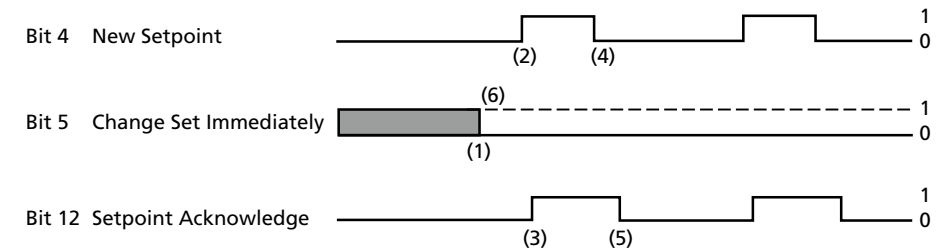


*Figure 12.2      Setpoint transfer from a master*

(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.
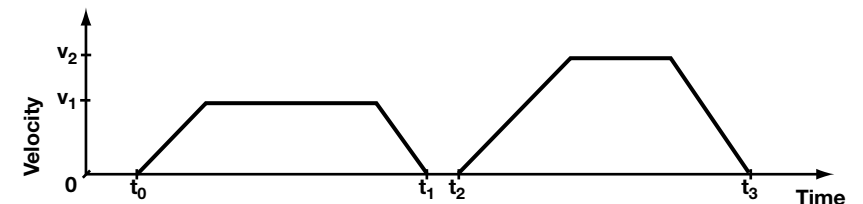


If the target position is reached at time $t_1$:
– The velocity is set to 0.
– The next target position is triggered at time $t_2$.

(6) If bit 5 "Change set immediately" is set to "1" (dotted line in Figure 6.4) the new target position is applied.
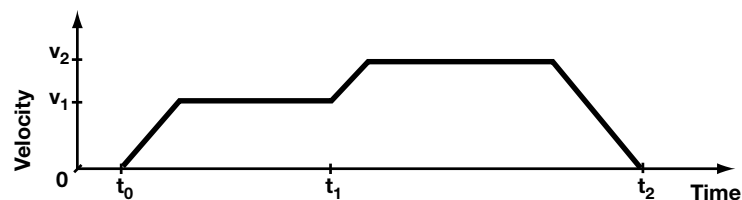
The drive receives:
– The 1st target position at time $t_0$,
– The 2nd target position at time $t_1$

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 58.

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

## 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 CiA DS-402 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 58").

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 |

*Table 12.8    Profile velocity mode*

**Note:**

The "Profile velocity mode" updates cyclically:

- The objects listed in the table.
- The object 0x6064 "Position actual value".

## Structure of the operation mode
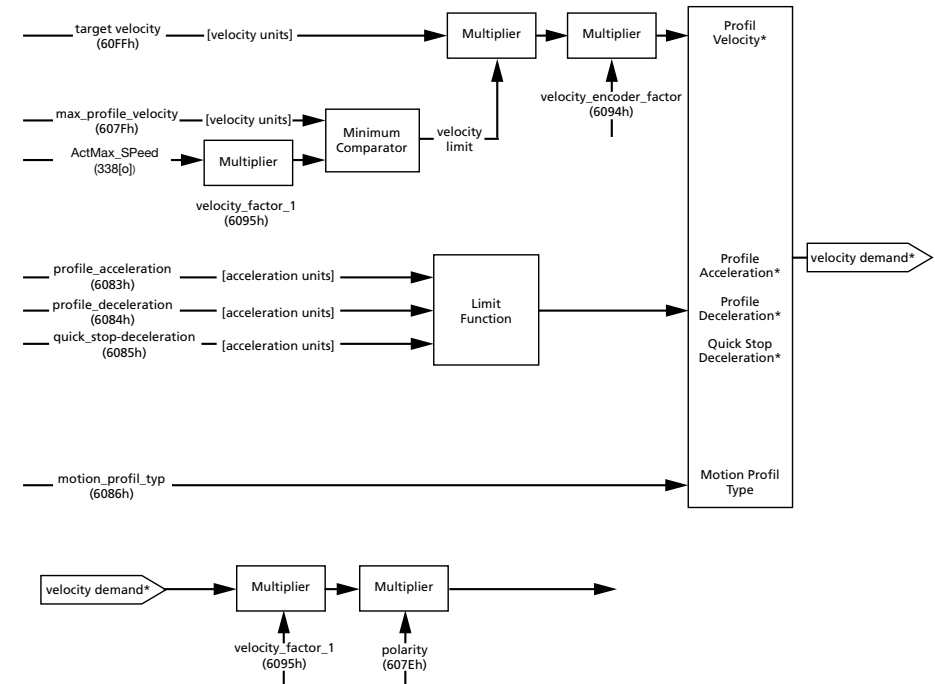


*Figure 12.3    Structure of profile velocity mode*

## 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:**
For information on controller-controlled homing of the drive using the touchprobe function, see chapter "14.1 Touch probe" on page 81.

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 100[0] to P 112[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<br>0x6099.02 - Slow jog speed | | | |

*Table 12.9      Objects for the "Homing mode" operation mode*
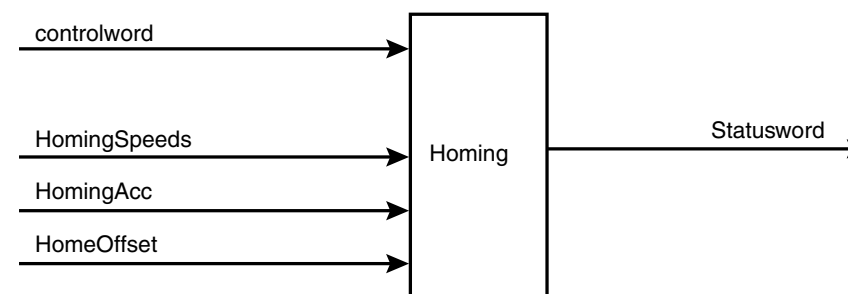


*Figure 12.4      Homing function*

The drive supports all 35 homing methods defined in CiA DS-402.

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

### 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 DriveManager 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 ServoOne online device help (available in the download area at www.lti-motion.com).

### 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 | Stop | 0 | Run instruction from bit 4 |
| | | 1 | Stops axis with deceleration |

*Table 12.10    HOMING MODE bits in the control word*

### Operation mode-specific bits in the status word

| Bit | Name | Value | Description |
|-----|------|-------|-------------|
| 10 | Target reached | 0 | Stop = 0: Target position not reached<br>Stop = 1: Axis is braking |
| | | 1 | Stop = 0: Target position reached<br>Stop = 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.<br>Homing failed.<br>Error code (see Online help) indicates the cause of the error. |
| 14 | ROT_0 | 1 | Axis stationary.<br>Velocity is less than parameter.<br>745 MON_REFWINDOW |

*Table 12.11    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 drive controller 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 12.12    Objects supported in the "Interpolated position mode"*
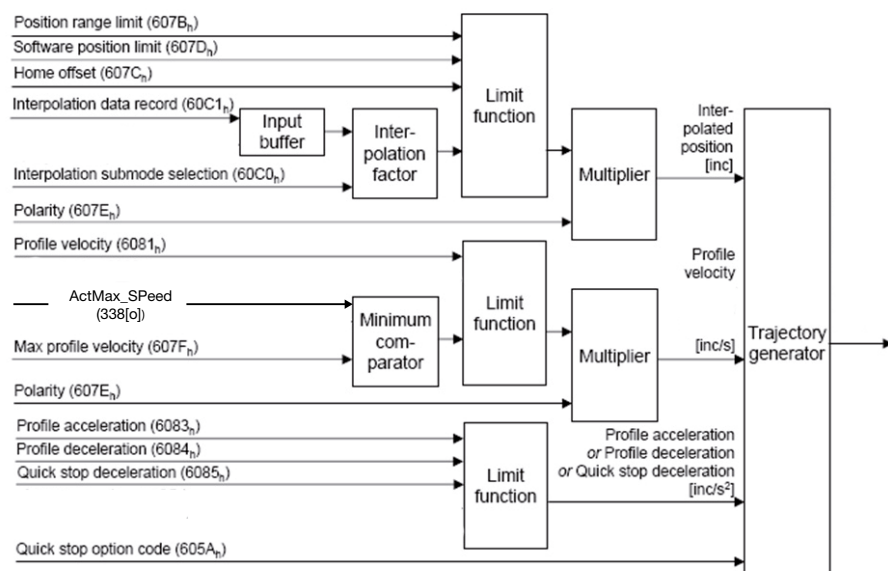
*Figure 12.5        Structure of the "Interpolated position mode"*

## 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 | Stop | 0 | Run instruction from bit 4 |
|   |      | 1 | Stop axis |

*Table 12.13     Operation mode-specific bits in the control word*

## Operation mode-specific bits in the status word

| Bit | Name | Value | Description |
|-----|------|-------|-------------|
| 10 | Target reached | 0 | Stop = 0: Position not (yet) reached<br>Stop = 1: Axis is braking |
|    |                | 1 | Stop = 0: Position reached<br>Stop = 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,<br>velocity is less than parameter<br>745 MON_REFWINDOW |
| 15 | Axis synchronized | 0 | Axis not synchronised |
|    |                   | 1 | Axis synchronised |

*Table 12.14     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 12.15     Objects supported in the "Cyclic synchronous position mode"*
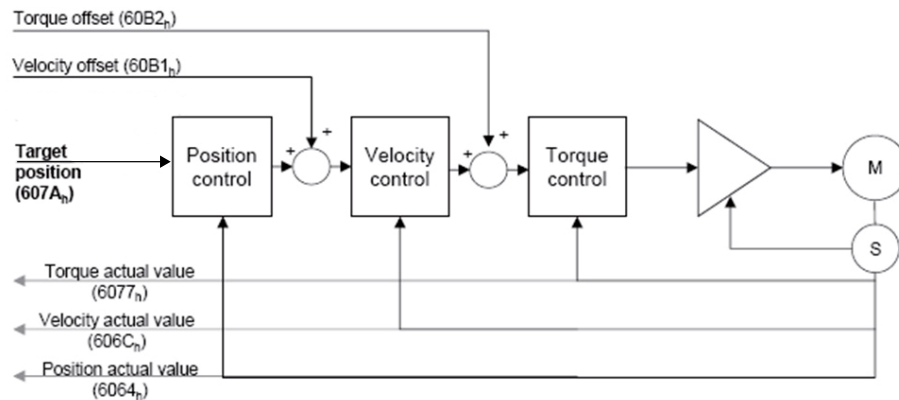


*Figure 12.6     Overview "Cyclic synchronous position mode"*

### 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 745 MON_REFWINDOW |
| 15 | Axis synchronized | 0 | Axis not synchronised |
| | | 1 | Axis synchronised |

*Table 12.16     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 12.17    Objects supported in the "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 745 MON_REFWINDOW |
| 15 | Axis synchronized | 0 | Axis not synchronised |
| | | 1 | Axis synchronised |

*Table 12.18    Operation mode-specific bits in the status word*
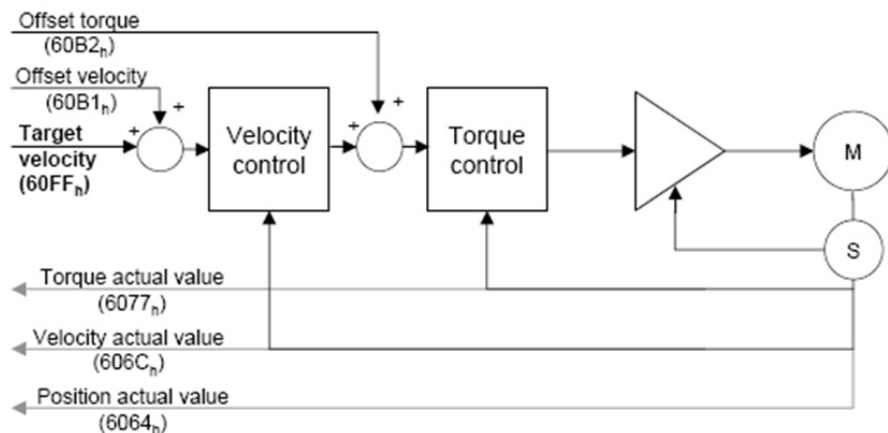


*Figure 12.7    Overview "Cyclic synchronous velocity mode"*

## 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 12.19    Objects supported in the "Cyclic synchronous torque mode"*
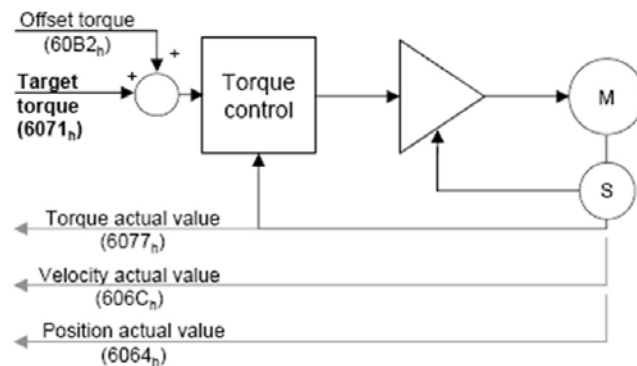


*Figure 12.8    Overview "Cyclic synchronous torque mode"*

### 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 745 MON_REFWINDOW |
| 15 | Axis synchronized | 0 | Axis not synchronised |
| | | 1 | Axis synchronised |

*Table 12.20    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 75) or "Cyclic synchronous velocity mode" (CSV, chapter "12.3.3 9 Cyclic synchronous velocity mode (only EtherCAT)" on page 76) 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 (CiA DS-402 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 12.21    CiA DS-402 objects for the external feed forward control*

| Parameter | Function | Value |
|---|---|---|
| **P 0375 – CON_IP_SFF – Scale** | Scaling speed feed forward control | 0 – 100 % referred to the feed forward control value |
| **P 0376 – CON_IP_TFF – Scale** | Scaling torque feed forward control | 0 – 100 % referred to the feed forward control value |
| **P 0379 – CON_IP_FF – Mode** | 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 |

*Table 12.22    ServoOne device parameters*

| Parameter | Function | Value |
|---|---|---|
| • Sub index 2 | Source torque feed forward control values | 0 = Internal feed forward control (default) 1 = External feed forward control |

*Table 12.22    ServoOne device parameters*

### Interpolation types:

Set the interpolation type via the parameter **P 0370 CON_IP**.
If you use external feed forward control, set this parameter to the value - SplineExtFF(2) = Interpolation with external feed forward.

### Checking the feed forward control variables in DriveManager 5

There are 2 ways of checking the feed forward variables sent externally in the ServoOne:

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 (drive controllers).

- 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 CiA DS-402** | | | **Drive controller** | | | | |
| Error | Emergency error code as per CiA DS-402 | 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 drive controller indicates errors in the field bus system, e.g. incorrect configurations, field bus malfunctions or others using the **error code 0xFF00 h.**

> **i** **Notes:**
> In the ServoOne online 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 CiA DS-402.

## 13.2 Error acknowledgement

Methods to acknowledge the device error:

1. The object 6040h, control word bit 7, edge controlled (for explanation see "In the ServoOne online device help you will find:").

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 drive controller.

5. The PC user software DriveManager.

6. Write the value "1" to the parameter 153 / **Object x 2099 MPRO_DRV-COM_FaultReset** (via DriveManager 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 chapter "11.3.1 Object 6040h control word" on page 63.

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 drive controller leaves the error state.

If the cause of the error has not been rectified:

- After the error acknowledgement by the "emergency message" the drive controller returns to the error state.

## 13.4   Error acknowledgement via the DriveManager

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

**i** **NOTE:**
You will find a list of "Error messages with rectification measures" in the "ServoOne online 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 2285 "Touch probe function selector"** to select between different implementations, e.g.

- CiA DS-402 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 CiA DS-402

1. To utilise the Touch probe function "Implementation according to CiA DS-402", set the **parameter 2285 "Touch probe function selector"** to 1 = "CiA DS-402(1)".
2. To acquire signals with both digital inputs ISD05 and ISD06, configure the **parameter 106 + 107 "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<br>0x60BA Touch probe pos1 pos value |
| | Alternatively:<br>• 0x60BB Touch probe pos1 neg value<br>• 0x60BC Touch probe pos2 pos value<br>• 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 drive controller 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 op. bit switches if the latch status changes) |
| 7 | Measuring point 1 saves negative edge value (only continuous op. 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 op. bit switches if the latch status changes) |
| 15 | Measuring point 2 saves negative edge value (only continuous op. 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)
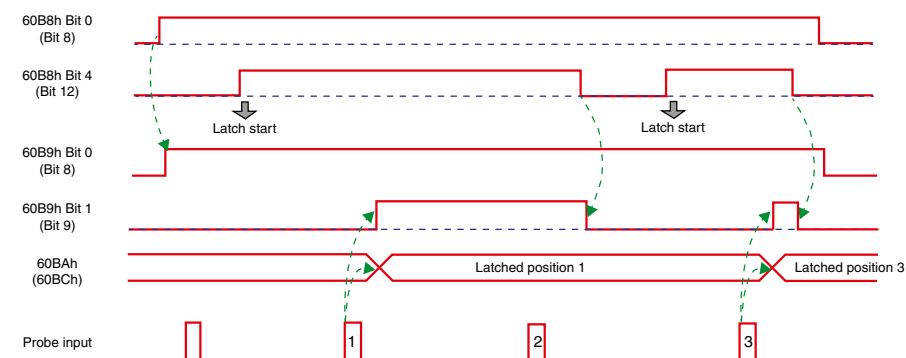


• Continuous Trigger Mode (60B8 h ,  Bit 1 = 1 or Bit 9 = 1)
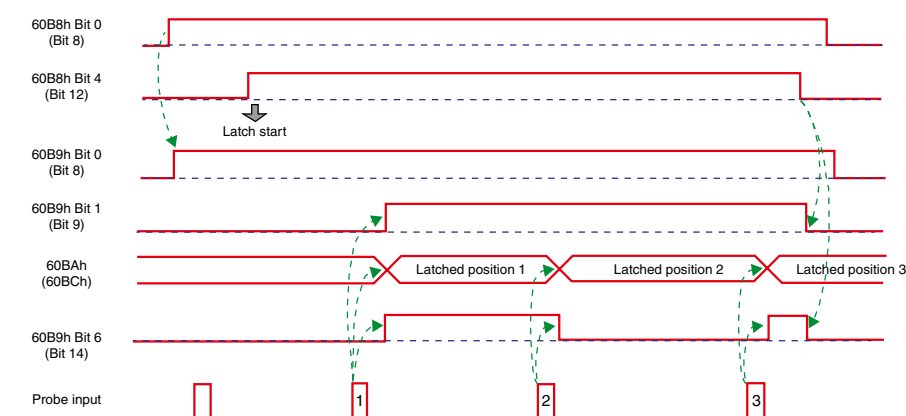


*Figure 14.1    Timing of a measurement in the "Single trigger mode" and "Continuous trigger mode"*

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 **P 106 + P 107 MPRO_Input_FS_ISD0x** as "PROBE(15)".

You will find descriptions of the parameters in the ServoOne device online 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<br>0x60BA Touch probe pos1 pos value |
| | Alternatively:<br>• 0x60BB Touch probe pos1 neg value<br>• 0x60BC Touch probe pos2 pos value<br>• 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 drive controller 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:
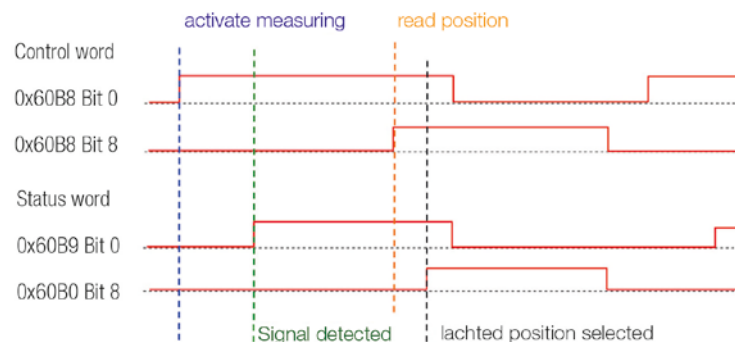


*Figure 14.2     Timing diagram for a Touch probe function measurement*

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 CiA DS-402.

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

| LTI documentation | ID no. / file formats | Web site |
|---|---|---|
| ServoOne Junior - Operation Manual | 1300.20B.x / PDF | www.lti-motion.com |
| ServoOne Single-Axis System - Operation Manual | 1100.20B.x / PDF | |
| ServoOne Multi-Axis System - Operation Manual | 1101.20B.x / PDF | |
| ServoOne Multi-Axis System Supply Unit - Operation Manual | 1101.21B.x / PDF | |
| ServoOne - online device help | 0842.26B.x / PDF and HTML | |

*Table 15.1      LTI documentation*

| Other field bus documentation | Web site |
|---|---|
| CiA® DS-301 (V 4.2.0): CANopen application layer and communication profile | www.can-cia.com |
| CiA® DS-402: 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 IEC61158 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

| | |
|---|---|
| DriveManager 5 | PC software from LTI for (initial) commissioning and process diagnostics on the devices in the ServoOne family |
| Electronic device data sheet – EDS file | Is installed in the "CANopen master" (controller) and is used to integrate ServoOne drive controllers into the CANopen device network (page 91) |
| SO | ServoOne drive controller:<br>**Single-axis system** - group of ServoOne devices (rated currents 4 - 450 A) that are connected to the AC mains and that operate an axis.<br>**Multi-axis system** - group of ServoOne devices (rated currents 4 - 170 A) that comprises DC-supplied axis modules and matched supply units. |
| SOj | ServoOne drive controller:<br>**Junior** - smallest member of the ServoOne 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