CANopen Programmer's Manual

CANopen®

EtherCAT®

copley
controls

Contents

1 ABOUT THIS MANUAL

1.1 Overview and Scope

This manual describes the CANopen implementation developed by Copley Controls for the Accelnet, Xenus, R-Series, Stepnet, and Argus drives. It contains useful information for anyone who participates in the evaluation or design of a distributed motion control system. The reader should have prior knowledge of motion control, networks, and CANopen.

Comments

Copley Controls welcomes your comments on this manual.
See http://www.copleycontrols.com for contact information.

Copyrights

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Copley Controls. Accelnet, Stepnet, Xenus, and CME 2 are registered trademarks of Copley Controls.

Document Validity

We reserve the right to modify our products. The information in this document is subject to change without notice and does not represent a commitment by Copley Controls. Copley Controls assumes no responsibility for any errors that may appear in this document.

1.2 Product Warnings

Use caution in designing and programming machines that affect the safety of operators.

WARNING

The programmer is responsible for creating program code that operates safely for the drives and motors in any given machine.

Failure to heed this warning can cause equipment damage, injury, or death.

Revision History

Revision	Date	Comments
00	June 13, 2017	Initial Release
01	November 22, 2017	Added to 0x2183, corrected graphic for PDO mapping. Corrected 0x2400 & 0x2401 data tables.
02	March 13, 2019	Added new parameters
03	April 26, 2019	Fixed minor issues
04	August 10, 2021	Updated object definitions, parameters, format

1.3   References

Readers of this book should also read information on CAN and CANopen at the "CAN in Automation" website at http://www.can-cia.org.

EtherCAT standards can be found on the EtherCAT Technolody Group (ETG) website:

   https://www.ethercat.org/default.htm

EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation Gmbh, Germany.

Those interested in

Running CAM Tables from RAM should also see the *Copley Camming User Guide* found at www.copleycontrols.com.

Information on Copley Controls Software can be found at www.copleycontrols.com.

Defining Documents

CiA 301: CANopen Application Layer and Communication Profile

Includes the physical and data link layers of the ISO 7-layer reference model. Physical connections, electrical characteristics of network signals, and bit-level communications. Grouping of data into frames, error detection, and confirmation of data received by slaves. Master/slave protocols, data types, and communication objects.
PDO, SDO, Time Stamps, Emergency object, network management.
Object definitions in Communication Profile: 0x1000 to 0x1FDFF.

CiA 402 Part 1: Device Profile for Drives and Motion Control, General Definitions

Specifies the mapping of the drive and motion control profile onto the generic power drive system (PDS) interface as defined in IEC 61800-7-1. Introduces objects in Standardized Device Profile Area, 0x6000 to 0x9FFF.

CiA 402 Part 2: Device Profile for Drives and Motion Control, Operation Modes and Application Data

Detailed object definitions for control of the power drive system. Factor Groups, Profile Position Mode, Homing mode, Position control function, Interpolated Position Mode, Profile Velocity Mode, Profile Torque Mode, Velocity Mode, Cyclic Synchronous Position Mode, Cyclic Synchronous Torque Mode, Inputs and Outputs. Shows object dictionary addressing for multi-axis drives.

CiA 402 Part 3: Device Profile for Drives and Motion Control, PDO Mapping

Specifies the PDO sets for servo drives and stepper drives. Mandatory RPDO and TPDO for control of drive.

IEC 61800-7-1: Adjustable Speed Power Drive Systems

Generic interface and use of profiles for power drive systems.

IEC 61800-7: ETG Implementation Guideline for the CiA402 Drive Profile

Specifications for using the IEC 61800-7 withing EtherCAT based servo drives.
Function groups for Position, Velocity, Torque, Torque Limiting, Homing, and Touch Probe.
Factor Groups, and PDOs.

1.4   Object Description Conventions

The samples below feature an **Object**, **Sub-Object**, and **tables of the data** in the objects.

Object

> **Object descriptions** are set off by **bold type** and a **heavy separator line**.
> As seen in this example:

SHUTDOWN OPTION CODE   INDEX 0x605B{ XE \B "0x605B" }

Sub-Object

> **Sub-index object descriptions** have a **regular** typeface and a **thinner line**.
> As seen in this example:

GEAR RATIO – SHAFT REVOLUTIONS INDEX 0x6091{ XE "0x6091" }, SUB-INDEX 2

> **Sub-index object 0** contains the number of elements contained by the record.

> **Color code Key:**
>
> - Objects used only in **CANopen** will be identified with this green color.
> - Objects used only in **EtherCAT** will be identified with this red color.
> - Objects used in both **CANopen** and **EtherCAT** will be identified with this dark blue color.

TRANSMIT PDO COMMUNICATION PARAMETERS   INDEX 0X1800 – 0X1807

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	NO	-

These objects allow configuration of communication parameters of each transmit PDO object.
Sub-index 0 contains the number of sub-elements of this record.

PDO COB-ID   INDEX 0X1800 – 0X1807, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See Default Values, below.	NO	R

This object holds the CAN object ID used by the PDO. The ID is formatted as follows:

Bit	Description
0-10	11-bit identifier for standard (CAN 2.0A) identifiers, or the lower 11 bits for extended (CAN 2.0B) identifiers.
11-28	Upper 18 bits of extended identifiers. For standard identifiers, these bits should be written as zeros.

Bit	Description
29	Identifier format. This bit is clear for standard (11-bit) identifiers, and set for extended (29-bit) identifiers.
30	If set, remote transmit requests (RTR) are not allowed on this PDO. If clear, the PDO is transmitted in response to a remote request.
31	Identifies the PDO as valid if clear. If set, the PDO is disabled and its mapping may be changed.

Default Values

The default values for this object are specified in the DS-301 CANopen specification.
These values are:

Index	Default ID
0x1800	0x00000180 + drive CAN node ID.
0x1801	0x00000280 + drive CAN node ID.
0x1802	0x00000380 + drive CAN node ID.
0x1803	0x00000480 + drive CAN node ID.
0x1804	0x80000000
0x1805	0x80000000
0x1806	0x80000000
0x1807	0x80000000

CANopen Data Types

These are the standard CANopen data types that are in Copley EDS (for CANopen) files.
They are found in 0x0001~0x001F of the OD (Object Dictionary) for Static Data Types.

Type	OD Index	Remarks
Non-Numeric Groups of Bits		
BOOLEAN	0x0001	
Integer Numbers		
INTEGER8	0x0002	
INTEGER16	0x0003	
INTEGER24	0x0010	
INTEGER32	0x0004	
INTEGER40	0x0012	
INTEGER48	0x0013	
INTEGER56	0x0014	
INTEGER64	0x0015	
UNSIGNED8	0x0005	
UNSIGNED16	0x0006	
UNSIGNED24	0x0016	
UNSIGNED32	0x0007	
Integer Numbers		
UNSIGNED40	0x0018	
UNSIGNED48	0x0019	

Type	OD Index	Remarks
UNSIGNED56	0x001A	
UNSIGNED64	0x001B	
Floating Point (Real) Numbers		
REAL32	0x0008	32-bit single precision number
REAL64	0x0011	64-bit single precision number
Groups of Numbers		
OCTET_STRING	0x000A	Array of 8-bit unsigned integers
UNICODE_STRING	0x000B	Array of 16-bit unsigned integers
Groups of Characters		
VISIBLE_STRING	0x0009	A string of ASCII characters
Other Numbers		
TIME_OF_DAY	0x000C	A 48-bit integer of number of days since January 1, 1984 and milliseconds since midnight of the current day.
TIME_DIFFERENCE	0x000D	A 48-bit integer of the number of days and milliseconds since midnight
DOMAIN	0x000F	A block of data

CANopen Object Definition Format and Key

OBJECT NAME		INDEX 0X			
Type	Access	Units	Range	Map PDO	Memory

Object name	Defined by DS-402 or Manufacture Specific
Index	Defined by DS-402 or Manufacture Specific may contain sub-indexes
Type	Data type defined in section 2.3.2 CANopen Data Types, or 2.3.4 Ether Data Types
Access	RO= Read Only; R/W= Read/Write
Units	Defined by dimensional properties of the object. E.g.: bits, seconds, counts, etc.
Range	Limit by type or defined by object
Map PDO	NO= Not mappable, T= Transmit only, R= Receive Only, TR= Transmit or Receive, EVENT= transmit on event
Memory	R= RAM only, F= Flash only, RF= RAM and FLASH

EtherCAT Data Types

These data types are the same in both IEC 61131-3 (PLC Open) and in Copley EDS (for CANopen) and ESI (for EtherCAT) files.

Type	Size	Remarks
Non-Numeric Groups of Bits		
BYTE	8 bits (1 byte)	Also 1 OCTET
WORD	16 bits (2 byte)	Also 2 OCTET
DWORD	32 bits (4 byte)	Also 4 OCTET
LWORD	64 bits, (8 byte)	Also 8 OCTET
Integer Numbers		
SINT	8 bits (1 byte)	Signed short integer
INT	16 bits (2 byte)	Signed integer
DINT	8 bits (1 byte)	Signed double integer
LINT	64 bits, (8 byte)	Signed 64-bit integer
USINT	32 bits (4 byte)	Unsigned short integer
UINT	16 bits (2 byte)	Unsigned integer
UDINT	32 bits (4 byte)	Unsigned double integer
ULINT	64 bits, (8 byte)	Unsigned 64-bit integer
Floating Point (Real) Numbers		
REAL	32 bit (4 byte)	A single real number
LREAL	64 bit (8 byte)	A double real number
Groups of Numbers		
ARRAY [0..N] OF UINT	[1..N+1] * 16 Bits	A grouping of UINT that can hold up to 16384 members.
Characters (ASCII)		
CHAR	8 bits (1 byte)	ASCII coding for letters, numbers, symbols, and non-printing characters
Groups of Characters		
STRING(40)	320 bits (40 bytes)	Contains 40 8-bit ASCII characters
STRING (N)	N * 8 bits (N bytes)	Contains N * 8, 8-bit ASCII characters

2   CAN & CANOPEN

2.1   Copley Controls Drives in CANopen Networks

Copley's CANopen Drives

Several lines of Copley Controls drives, including Accelnet, Stepnet, Xenus, Argus, and the ruggedized R-Series, offer distributed motion control through support of the *Controller Area Network* (CAN) and the CANopen network profiles. Using CANopen, the drives can take instruction from a master application to perform homing operations, point-to-point motion, profile velocity motion, profile torque, and interpolated motion. (These drives also support serial communication.)

CAN & CANopen

CAN specifies the data link and physical connection layers of a fast, reliable network. The CANopen profiles specify how various types of devices, including motion control devices, can use the CAN network in a highly efficient manner.

Architecture

As illustrated below, in a CANopen motion control system, control loops are closed on the individual drives, not across the network. A master application coordinates multiple devices, using the network to transmit commands and receive status information. Each device can transmit to the master or any other device on the network. CANopen provides the protocol for mapping device and master internal commands to messages that can be shared across the network.
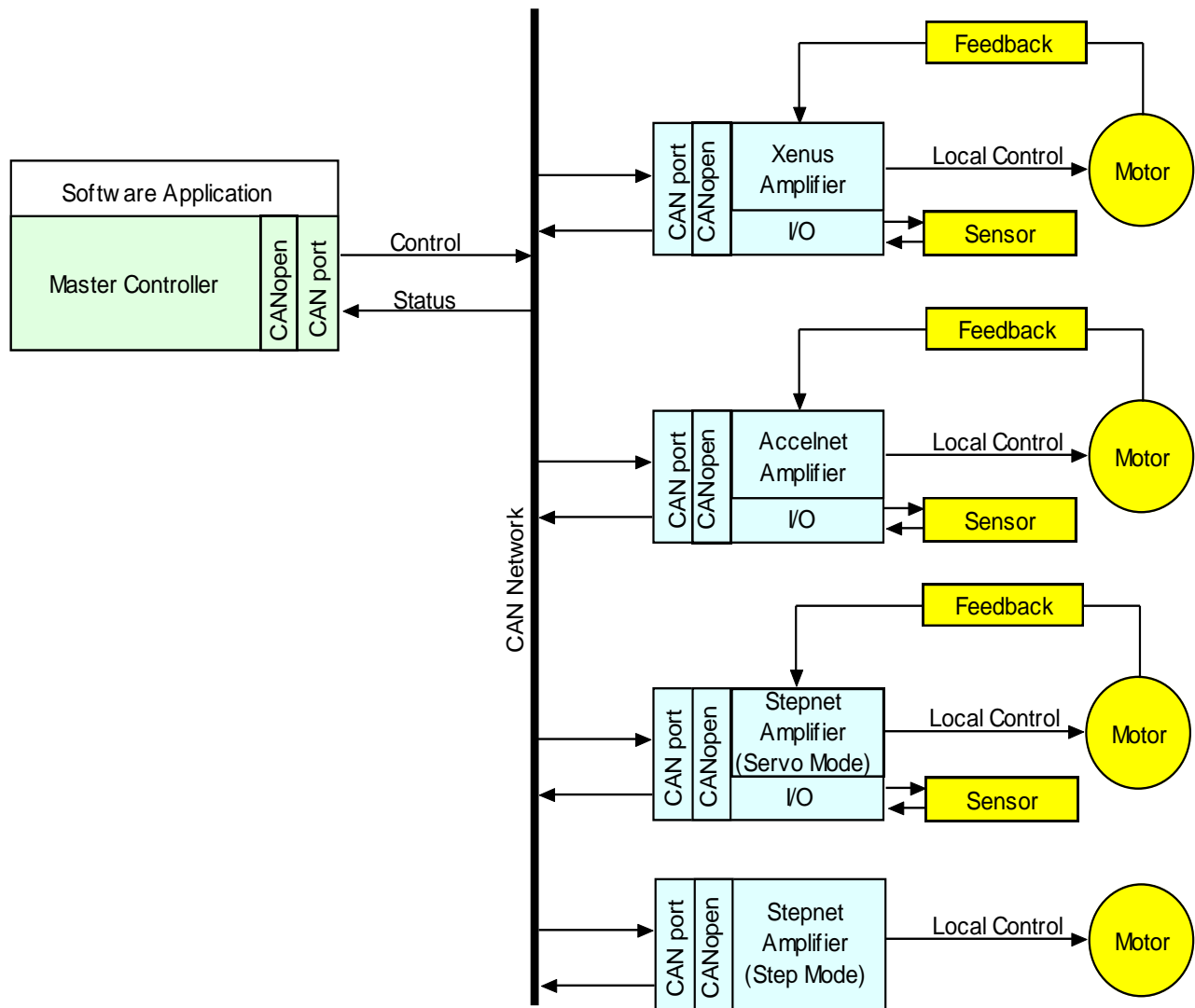
Figure 1—CANopen motion control system

A CANopen network can support up to 127 nodes. Each node has a seven-bit node ID in the range of 1-127. Node ID 0 is reserved for the master and slave addresses can be 1~127.

Addressing Drives on a CANopen Network

The CANopen Master always has an address = 0. Slave address are always greater than 0. In Copley multi-axis drives, Axis 1 will have a node address equal to the address switch setting.

The following axes in the drive appear as independent nodes on the network, each with an incrementing address greater than the switch address.

Using the Control Word as an example, it will be the same in all of the drives and axes. When setting the address switches for multi-axis drives in a network, keep in mind that the effective addresses of a multi-axis drive will be greater than what is shown on the switches.
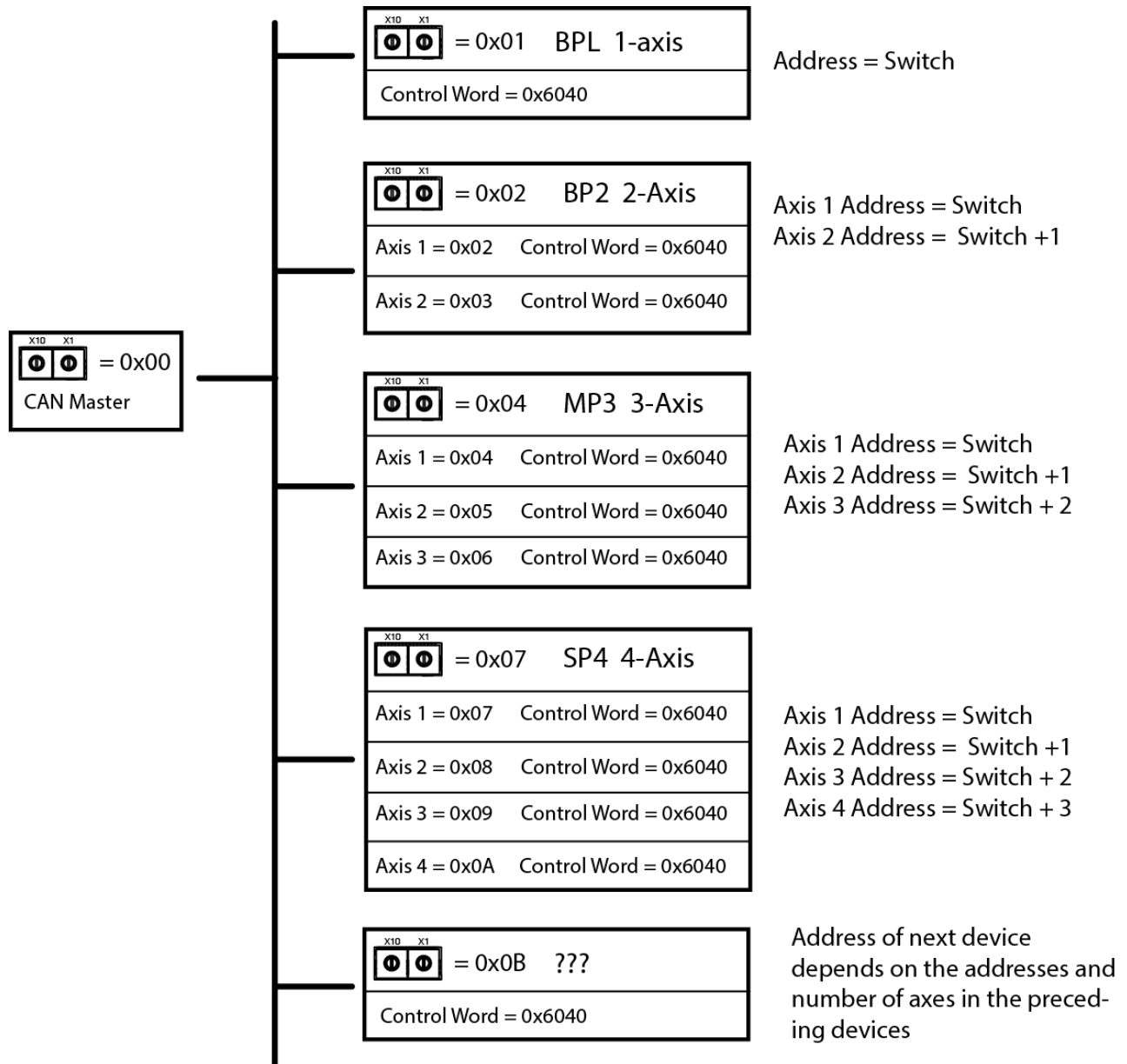
Figure 2—Addressing Drives on a CANopen Network

Example of a CANopen Move Sequence

→ CANopen master transmits a control word to initialize all devices.

→ Devices transmit messages indicating their status (in this example, all are operational).

→ CANopen master transmits a message instructing devices to perform homing operations.

→ Devices indicate that homing is complete.

→ CANopen master transmits messages instructing devices to enter position profile mode (point-to-point motion mode) and issues first set of point-to-point move coordinates.

→ Devices execute their moves, using local position, velocity, and current loops, and then transmit actual position information back to the network.

→ CANopen master issues next set of position coordinates.

Overview of the CAN Protocol

A Network for Distributed Control

The backbone of CANopen is CAN, a serial bus network originally designed by Robert Bosch GmbH to coordinate multiple control systems in automobiles.

The CAN model lends itself to distributed control. Any device can broadcast messages on the network. Each device receives all messages and uses filters to accept only the appropriate messages. Thus, a single message can reach multiple nodes, reducing the number of messages that need to be sent. This also greatly reduces bandwidth required for addressing, allowing distributed control at real-time speeds across the entire system.

CAN Benefits

Other benefits of CAN include:

- Wide use of CAN in automobiles and many other industries assures availability of inexpensive hardware and continued support. Ready availability of standard components also reduces system design effort.

- CAN's relative simplicity reduces training requirements.

- By distributing control to devices, CAN eliminates the need for multiple wire connections between devices and a central controller. Fewer connections enable increased reliability in harsh operating conditions.

- Device-based error checking and handling methods make CAN networks even more reliable.

Physical Layer

The physical layer of CAN is a differentially driven, two-wire bus, terminated by 120-Ohm resistors at each end. The maximum bit rate supported by CAN is 1,000,000 bits/second for up to 25 meters. Lower bit rates are required for longer network lengths.

The CAN Message

Overview

CANopen messages are transmitted within CAN messages (a CAN message is also known as a communication object or COB).

CAN Message Format

CAN messages are communicated over the bus in the form of network packets. Each packet consists of an identifier (CAN message ID), control bits, and zero to eight bytes of data.

CRC Error Checking

Each packet is sent with CRC (cyclic redundancy check) information to allow controllers to identify and re-send incorrectly formatted packets.

CAN Message ID

Every CAN message has a CAN message ID (also known as COB-ID). The message ID plays two important roles:

It provides the criteria by which the message is accepted or rejected by a node, and it determines the message's priority, as described below.

CAN Message Priority

The priority of a CAN message is encoded in the message ID. The lower the value of the message ID, the higher the priority of the message. When two or more devices attempt to transmit packets at the same time, the packet with the highest priority succeeds. The other devices back off and retry.

This method of collision handling allows for a high bandwidth utilization compared to other network technologies. For instance, Ethernet handles collisions by requiring both devices to abort transmission and retry.

For More Information

For more information on the CAN protocol, see CAN Specification 2.0, Robert Bosch GmbH, and ISO 11898, Road Vehicles, Interchange of Digital Information, Controller Area Network (CAN) for high-speed communication.

Overview of the CANopen Profiles

Communication and Device Profiles

CANopen is a set of profiles built on a subset of the CAN application layer protocol. The CANopen profiles achieve two basic objectives:

- Specifies methods for packaging multiple CAN messages to send large blocks of data as a single entity.

- Standardizes and simplifies communication between devices within several application types, including motion control.

Developed by the *CAN In Automation* (CiA) group, CANopen includes the underlying *CANopen Application Layer and Communication Profile (DS 301)* and several device profiles, including *CANopen Profile for Drives and Motion Control (DSP 402)*.

Communication Profile

The *Application Layer and Communication Profile* describes the communication techniques used by devices on the network. All CANopen applications must implement this profile.

Profile for Drives and Motion Control

Each of the CANopen device profiles describes a standard device for a certain application. Copley Controls CANopen drives comply with the *Profile for Drives and Motion Control.* This profile specifies a state machine and a position control function. It also supports several motion control modes, including:

- Homing
- Profile position
- Profile velocity
- Profile torque
- Interpolated position
- CSP: Cyclic synchronous position
- CSV: Cyclic synchronous velocity
- CST: Cyclic synchronous torque
- CSTCA: Cyclic synchronous torque with commutation angle

The drive's operating mode is set using the Mode Of Operation object (index 0x6060). The *Profile for Drives and Motion Control* also contains modes currently not supported by Copley.

EtherCAT CoE

EtherCAT is a communication and control network based on the EtherNet hardware layer.
It supports the following communication profiles:

- CAN application profile over EtherCAT (CoE)
- File Accesss over EtherCAT (FoE)
- Ethernet over EtherCAT (EoE)
- Servo drive profile, according to IEC 61800-7-201 (SoE)

Copley's EtherCAT drives support CoE and FoE. Most CANopen objects are in common with CoE objects.

Those objects that are not in common are color-identified with the object name in red.
CANopen objects that are not supported by CoE are identified in green.

As discussed in section *1.4 Object Description Conventions* the following describes the use of object-coloring key:

- Objects used only in **CANopen** will be identified with this green color.
- Objects used only in **EtherCAT** will be identified with this red color.
- Objects used in both **CANopen** and **EtherCAT** will be identified with this dark blue color.

2.2 Defining and Accessing CANopen Devices

Defining a Device: CANopen Objects and Object Dictionaries

Objects and Dictionaries

The primary means of controlling a device on a CANopen network is by writing to device parameters and reading device status information. For this purpose, each device defines a group of parameters that can be written, and status values that can be read. These parameters and status values are collectively referred to as the device's objects.

These objects define and control every aspect of a device's identity and operation. For instance, some objects define basic information such as device type, model, and serial number. Others are used to check device status and deliver motion commands.

The entire set of objects defined by a device is called the device's object dictionary. Every device on a CANopen network must define an object dictionary, and nearly every CANopen network message involves reading values from or writing values to the object dictionaries of devices on the network.

Object Dictionary as Interface

The object dictionary is an interface between a device and other entities on the network.
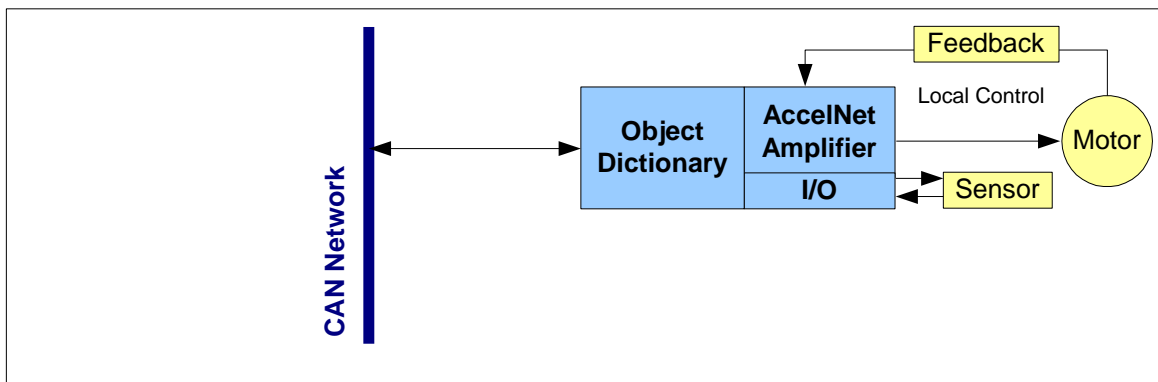
Figure 3—CANopen motion control system

CANopen Profiles and the Object Dictionary

The CANopen profiles specify the mandatory and optional objects that comprise most of an object dictionary. The Communication Profile specifies how all devices must communicate with the CAN network. For instance, the Communication Profile specifies dictionary objects that set up a device's ability to send and receive messages. The device profiles specify how to access particular functions of a device. For instance, the *CANopen Profile for Drives and Motion Control (DSP 402)* specifies objects used to control device homing and position control.

In addition to the objects specified in the *Application Layer and Communication Profile* and device profiles, CANopen allows manufacturers to add device-specific objects to a dictionary.

Addressing Drives on an EtherCAT Network: Device ID

There are two forms of identifying devices on a network:

1) The first is done by the master, scanning the network and assigning addresses that begin with -1 and decrementing thereafter. In this mode of identification the EtherCAT master assigns the Device ID based on the position of the node of the network.

2) The second, used for explicit device identification, employs switches on each drive. These produce the *Device Identification Value* (Device ID).

The Device ID is also saved in the SII (Slave Information Interface) eeprom as the *Configured Station Alias* parameter.

Address 0 is not allowed. Each drive must have address that is unique, but the difference between addresses will not be affected by the number of axes in a drive.

Instead, the object addresses for each axis will increment by 0x800, and the PDO addresses for each axis will increment by 0x40.

Figure 3 below shows the switch settings and access to the objects and PDOs.
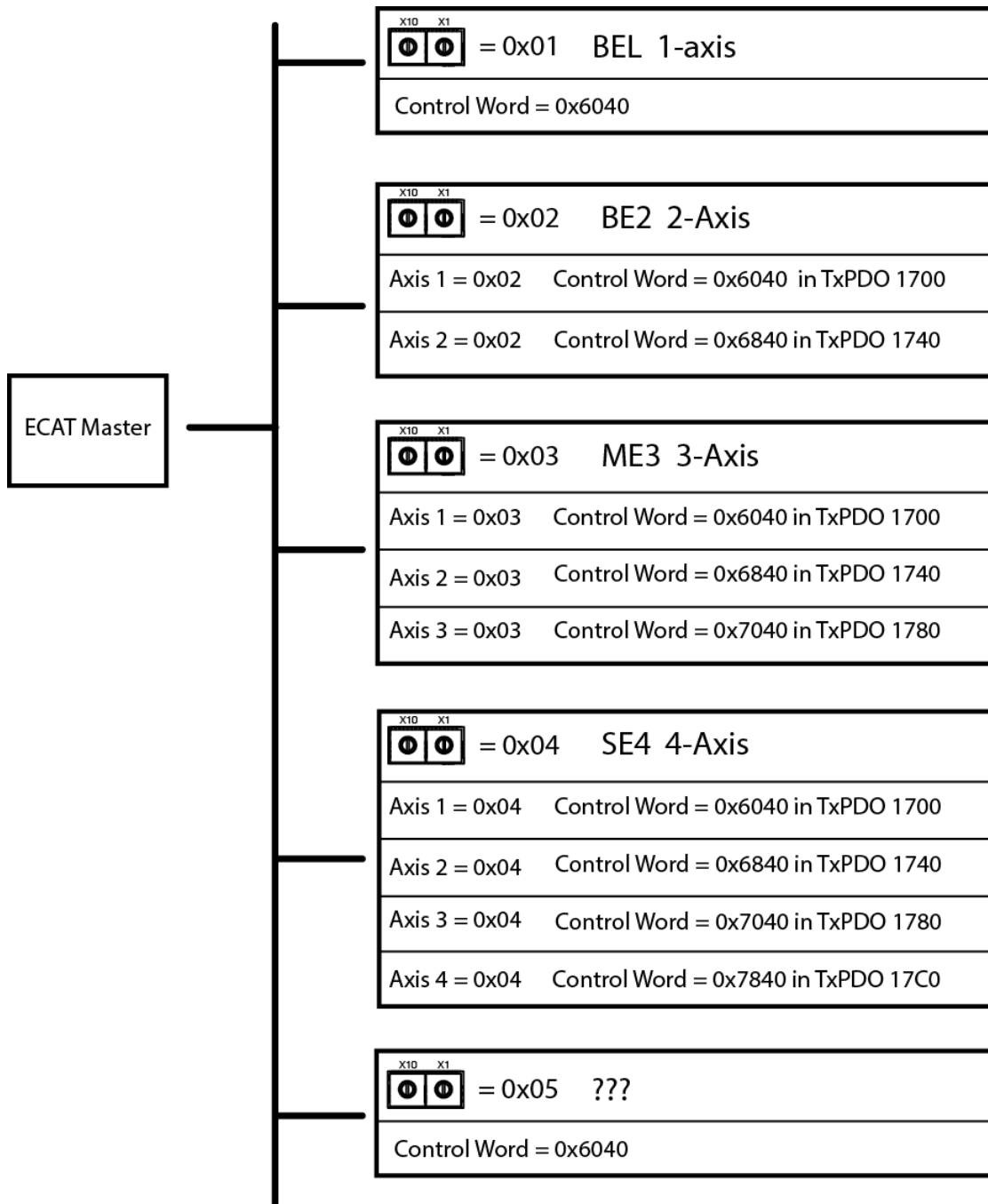
Figure 4—Switch settings and access to objects and PDOs

2.3   Object Dictionary Structure

An object dictionary is a lookup table. Each object is identified by a 16-bit index with an eight-bit sub-index. Most objects represent simple data types, such as 16-bit integers, 32-bit integers, and strings. These can be accessed directly by the 16-bit index.

Other objects use the sub-index to represent groups of related parameters. For instance, the Motor Data object (index 0x2383) has 24 sub-index objects defining basic motor characteristics such as motor type, motor wiring configuration, and Hall sensor type. (The sub-index provides up to 255 subentries for each index.)

The organization of the dictionary is specified in the profiles, as shown below.

Index Range	Objects
0000	Not used
0001-001F	Static Data Types
0020-003F	Complex Data Types
0040-005F	Manufacturer Specific Complex Data Types
0060-007F	Device Profile Specific Static Data Types (including those specific to motion control)
0080-009F	Device Profile Specific Complex Data Types (including those specific to motion control)
00A0-0FFF	Reserved
1000-1FFF	Communication Profile Area (DS 301)
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardized Device Profile Area (including Profile for Motion Control)
A000-FFFF	Reserved

Objects in the range of 0x2000~0x27FF and 0x6000~0x67FF use an offset of 0x800 to add to the base object address for multi-axis drives. These are the ranges of object addresses for multi-axis drives. The items with "n/a" are not defined in DS-402:

Address	DS-402	Copley
0x2000 to 0x27FF	n/a	Axis A
0x2800 to 0x2FFF	n/a	Axis B
0x3000 to 0x37FF	n/a	Axis C
0x3800 to 0x3FFF	n/a	Axis D
0x6000 to 0x67FF	Axis 0	Axis A
0x6800 to 0x6FFF	Axis 1	Axis B
0x7000 to 0x77FF	Axis 2	Axis C
0x7800 to 0x7FFF	Axis 3	Axis D

Accessing the Object Dictionary

Two Basic Channels

CANopen provides two ways to access a device's object dictionary:

- The Service Data Object (SDO)
- The Process Data Object (PDO)

Each can be described as a channel for access to an object dictionary.

SDOs and PDOs

SDO	PDO
The SDO protocol allows any object in the object dictionary to be accessed, regardless of the object's size. This comes at the cost of significant protocol overhead.	One PDO message can transfer up to eight bytes of data in a CAN message. There is no additional protocol overhead for PDO messages.
Transfer is always confirmed.	PDO transfers are unconfirmed.
Has direct, unlimited access to the object dictionary.	Requires prior setup, wherein the CANopen master application uses SDOs to map each byte of the PDO message to one or more objects. Thus, the message itself does not need to identify the objects, leaving more bytes available for data.
Employs a client/server communication model, where the CANopen master is the sole client of the device object dictionary being accessed.	Employs a peer-to-peer communication model. Any network node can initiate a PDO communication, and multiple nodes can receive it.
An SDO has two CAN message identifiers: a transmit identifier for messages from the device to the CANopen master, and a receive identifier for messages from the CANopen master.	Transmit PDOs are used to send data from the device and receive PDOs are used to receive data.
SDOs can be used to access the object dictionary directly.	A PDO can be used only after it has been configured using SDO transfers.
Best suited for device configuration, PDO mapping, and other infrequent, low priority communication between the CANopen master and individual devices. Such transfers tend to involve the setting up of basic node services; thus, the term **service data object**.	Best suited for high-priority transfer of small amounts of data, such as delivery of set points from the CANopen master or broadcast of a device's status. Such transfers tend to relate directly to the application process; thus, the term **process data object.**
For more information about SDOs, see SDOs: Description and Examples	For more information about PDOs, see PDOs: Description and Examples
For help deciding whether to use an SDO or a PDO see SDO vs. PDO: Design Considerations	

Copley SDOs and PDOs

The Communication Profile requires the support of at least one SDO per device. (Without an SDO, there would be no way to access the object dictionary.) It also specifies default parameters for four PDOs. Copley Controls CANopen drives each support 1 SDO and 16 PDOs (eight transmit-PDOs and eight receive-PDOs).
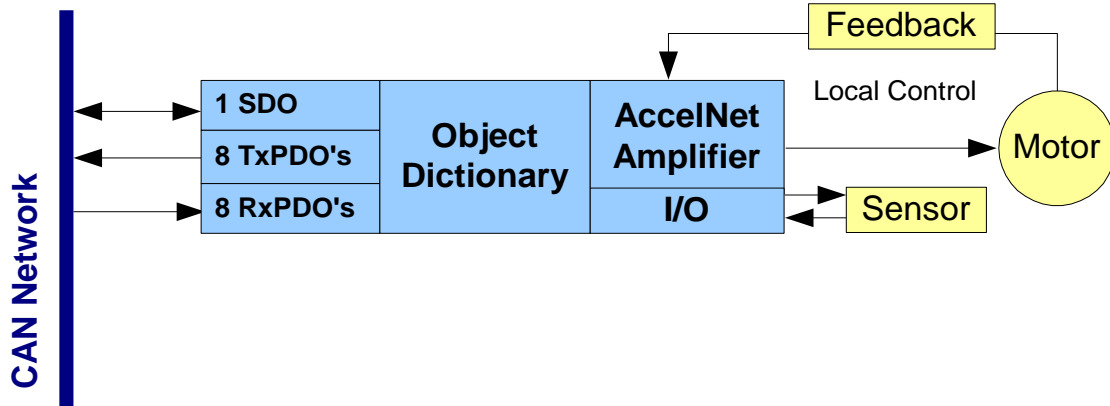
Figure 5—Copley SDOs and PDOs

SDOs: Description and Examples

Overview

Each drive provides one SDO. The CANopen master can use this SDO to configure, monitor, and control the device by reading from and writing to its object dictionary.

SDO CAN Message IDs

The SDO protocol uses two CAN message identifiers. One ID is used for messages sent from the CANopen master (SDO client) to the drive (SDO server). The other ID is used for messages sent from the SDO server to the SDO client.

The CAN message ID numbers for these two messages are fixed by the CANopen protocol. They are based on the device's node ID (which ranges from 1 to 127). The ID used for messages from the SDO client to the SDO server (i.e. from the CANopen master to the drive) is the hex value 0x600 + the node ID. The message from the SDO server to the SDO client is 0x580 + the node ID. For example, an drive with node ID 7 uses CAN message IDs 0x587 and 0x607 for its SDO protocol.

Client/Server Communication

The SDO employs a client/server communication model. The CANopen master is the sole client. The device is the server. The CANopen master application should provide a client SDO for each device under its control.

The CAN message ID of an SDO message sent from the CANopen master to a device should match the devices receive SDO message identifier. In response, the CANopen master should expect an SDO message whose CAN message ID matches the devices transmit SDO message identifier.

SDO Message Format

The SDO uses a series of CAN messages to send the segments that make up a block of data. The full details of the SDO protocol are described in the *CANopen Application Layer and Communication Profile*.

Confirmation

Because an SDO transfer is always confirmed, each SDO transfer requires at least two CAN messages (one from the master and one from the slave).

Confirmation Example

Updating an object that holds an eight-byte long value requires six CAN messages, for example:

1. The master sends a message to the device indicating its intentions to update an object in the device's dictionary. The message includes the object's index and sub-index values as well as the size (in bytes) of the data to be transferred.

2. The device responds to the CANopen master indicating that it is ready to receive the data.

3. The CANopen master sends one byte of message header information and the first 7 bytes of data. (Because SDO transfers use one byte of the CAN message data for header information, the largest amount of data that can be passed in any single message is 7 bytes.)

4. The device responds indicating that it received the data and is ready for more.

5. The CANopen master sends the remaining byte of data along with the byte of header information.

6. The device responds indicating success.

Segmented, Expedited and Block Transfers

As in the example above, most SDO transfers consist of an initiate transfer request from the client, followed by series of confirmed eight-byte messages. Each message contains one byte of header information and a segment (up to seven bytes long) of the data being transferred.

For the transfer of short blocks of data (four bytes or less), the Communication Profile specifies an expedited SDO method. The entire data block is included in the initiate SDO message (for downloads) or in the response (for uploads). Thus, the entire transfer is completed in two messages.

The Communication Profile also describes a method called block SDO transfers, where many segments can be transferred with a single acknowledgement at the end of the transfer. Copley Controls CANopen drives do not require use of the block transfer protocol.

PDOs: Description and Examples for CANopen

Overview

Each drive provides eight transmit PDOs and eight receive PDOs. A transmit PDO is used to transmit information from the device to the network. A receive PDO is used to update the device.

Default PDO Message Identifiers

The Communication Profile reserves four CAN message identifiers for transmit PDOs and four identifiers for receive PDOs. These addresses are described later in this chapter (see Receive PDO Communication Parameters).

The first four transmit PDOs and receive PDOs provided in Copley Controls CANopen drives use these default addresses. The addresses of the remaining four transmit PDOs and receive PDOs are null by default. The designer can reconfigure any PDO message identifier.

PDO Peer- to-Peer Communication

Peer-to-peer relationships are established by matching the transmit PDO identifier of the sending node to a receive PDO identifier of one or more other nodes on the network.

Any device can broadcast a PDO message using one of its eight transmit PDOs. The CAN identifier of the outgoing message matches the ID of the sending PDO. Any node with a matching receive PDO identifier will accept the message.

PDO Peer-to- Peer Example

For instance, Node 1, transmit PDO 1, has a CAN message ID of 0x0189. Node 2, receive PDO 1 has a matching ID, as does Node 3. They both accept the message. Other nodes do not have a matching receive PDO, so no other nodes accept the message.

PDO Mapping

PDO mapping allows optimal use of the CAN message's eight-byte data area.

Mapping uses the SDO to configure dictionary objects in both the sending and the receiving node to know, for each byte in the PDO message:

- The index and sub-index which objects are to be accessed
- The type of data
- The length of the data

Thus, the PDO message itself carries no transfer control information, leaving all eight bytes available for data. (Contrast this with the SDO, which uses one byte of the CAN message data area to describe the objects being written or read, and the length of the data.)

Mappable Objects

Not all objects in a device's object dictionary can be mapped to a PDO. If an object can be mapped to a PDO, the MAP PDO field in the object's description in this manual contains the word EVENT or the word YES.

Dynamic PDO Mapping

Copley supports the CANopen option of dynamic PDO mapping, which allows the CANopen master to change the mapping of a PDO during operation. For instance, a PDO might use one mapping in Homing Mode, and another mapping in Profile Position Mode.

PDO Transmission Modes

PDOs can be sent in one of two transmission modes:

1. **Synchronous.** Messages are sent only after receipt of a specified number of synchronization (SYNC) objects, sent at regular intervals by a designated synchronization device.

   (For more information on the SYNC object, see SYNC and High-resolution Time Stamp Messages)

2. **Asynchronous.** The receipt of SYNC messages does not govern message transmission. Synchronous transmission can be cyclic, where the message is sent after a predefined number of SYNC messages, or acyclic, where the message is triggered by some internal event but does not get sent until the receipt of a SYNC message.

PDO Triggering Modes

The transmission of a transmit PDO message from a node can be triggered in one of three ways:

	Trigger	**Description**
1	Event	Message transmission is triggered by the occurrence of an object specific event. For synchronous PDOs this is the expiration of the specified transmission period, synchronized by the reception of the SYNC object. For acyclically transmitted synchronous PDOs and asynchronous PDOs the triggering of a message transmission is a device-specific event specified in the device profile.
2	SYNC message	For synchronous PDOs, the message is transmitted after a specified number of SYNC cycles have occurred.
3	Remote Request	The transmission of an asynchronous PDO is initiated on receipt of a remote request initiated by any other device.

Default PDO Mappings

Copley Controls CANopen drives are shipped with the default PDO mappings specified in the *Profile for Drives and Motion Control.* These mappings are:

RECEIVE PDOs		**TRANSMIT** PDOs	
PDO	**Default mapping**	**PDO**	**Default mapping**
1	0x6040 (Control Word)	1	0x6041(Status Word)
2	0x6040, 0x6060 (Mode Of Operation)	2	0x 6041, 0x 6061
3	0x6040, 0x607A (Target Position)	3	0x 6041, 0x6064 (Position Actual Value)
4	0x6040, 0x60FF (Target Velocity)	4	0x 6041, 0x606C (Actual Velocity)
5	0x6040, 0x6071 (Target Torque)	5	0x 6041, 0x6077 (Torque Actual Value)
6	0x6040	6	0x 6041
7	0x6040	7	0x 6041, 0x60FD (Digital Inputs)
8	0x6040, 0x6060	8	no default mapping

For more information see the *CANopen Profile for Drives and Motion Control (DSP 402).*

PDO Examples

The designer has broad discretion in the use of PDOs. For example:

→ On the device designated as the SYNC message and time stamp producer, map a transmit PDO to transmit the high-resolution time stamp message on a periodic basis. Map receive PDOs on other devices to receive this object.

→ On each drive, map a transmit PDO to transmit PVT buffer status updates in interpolated position mode. Map a receive PDO to receive PVT segments.

→ Another transmit PDO could transmit general drive status updates.

The Copley Controls CANopen Motion Libraries product (CML) uses these default mappings:

RECEIVE PDOs		TRANSMIT PDOs	
PDO	Default mapping	PDO	Default mapping
1	IP move segment command (index 0x2010) Used to receive the PVT segments.	4	Trajectory Buffer Status object (index 0x2012). This is also used with transmission type 255. The PDO will be transmitted each time a segment is read from the buffer, or on an error condition.
5	High-resolution Time Stamp (index 0x1013) on the drive designated as the time-stamp transmitter. CML programs this object with transmit type 10 (transmit every 10 sync cycles). The sync cycle is 10 milliseconds. Thus, the timestamp is transmitted every 100 milliseconds.	5	High-resolution Time Stamp (index 0x1013) on all but the time-stamp transmitter.
		2	Various status information: Status Word (index 0x6041), Manufacturer Status Register object (index 0x1002), and Input Pin States (index 0x2190). CML programs this PDO to transmit on an event (transmission type 255). This causes the PDO to be transmitted any time an input pin changes, or a status bit changes. Note that Copley input pins have a programmable debounce time, so if one of the inputs is connected to something that might change rapidly, then the debounce time can be used to keep it from overloading the CANopen network.

PDO REQUEST OBJECT   0x2002

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	-	1~7	NO	R

Writing a PDO number 1~7 to this object will cause that transmit PDO to be sent.
Find the PDO numbering and details here: Default PDO Mappings.

PDOs: Timing Considerations for EtherCAT

Under EtherCAT control, PDOs are typically exchanged at a synchronous rate determined by the EtherCAT master software. Copley drives for EtherCAT have two internal operating frequencies.

- Current loop   16 kHz   "PWM"

- Velocity / Position loops   4 kHz   "Servo"

Fixed PDOs

▪ Guaranteed to operate at the Servo frequency and are mandated when process data must be updated with constant timing.

▪ Not accessible for user customization.

Non-fixed PDO (Mappable)

▪ Not guaranteed to pass data consistently at the Servo frequency

▪ Lower priority tasks than Fixed PDOs

▪ Configurable by users.

Sync Managers

▪ SM 0: Master sends SDO to slave mailbox

▪ SM 1: Master receives SDO from slave mailbox

▪ SM 2: Master sends PDOs to slave

▪ SM 3: Master receives PDOs from slave

Default PDO Mappings

These mappings are:

RECEIVE PDO, Sync Manager 2			TRANSMIT PDO, Sync Manager 3		
Index	RxPDO	Default mapping	Index	TxPDO	Default mapping
0x1600	1	Non-Fixed PDOs User Mappable	0x1A00	1	Non-Fixed PDOs User Mappable
0x1601	2		0x1A01	2	
0x1602	3		0x1A02	3	
0x1603	4		0x1A03	4	
0x1700	5	0x6 040, 0x607A (Target Position), 0x60B1 (Velocity Offset), 0x60B2 (Torque Offset) For CSP mode	0x1B00	5	0x 6041 (Status Word), 0x6064 (Actual Motor Position), 0x60F4 (Position Following Error), 0x606C (Actual Motor Velocity), 0x6077 (Torque Actual Value)
0x1701	6	0x6040, 0x60FF, 0x60B2 For CSV mode	0x1B01	6	0x6041 (Status Word), 0x2242 (Load Encoder Position), 0x2240 (Motor Encoder Position), 0x2200 (Analog/ Digital Reference Input Value), 0x219A (Input Pin State)
0x1702	7	0x6040, 0x6071 (Target Torque),			

		For CST mode
0x1703	8	0x2327 (U Input), 0x2328 (V Input) For UV Current mode
0x1704	9	0x6040, 0x6071, 0x60EA (Commutation Angle) For CSTCA mode

PDOs: Description and Examples for EtherCAT

Transmit and Receive labels in the ESI (XML) file apply to the EtherCAT slave (Copley drive)
RxPDO = Master -> Slave
TXPDO = Master <- Slave

The ESI files are set up to configure drives in the CSP (Cyclic Synchronous Position) mode by default. Items in Bold are the RxPdo, TXPDO, and SDO that are used in this configuration.

Non-fixed, empty RxPdo, Mappable

Type	Object	Name	Fixed	SM
RxPdo	0x1600	Receive PDO 1	no	n/a
RxPdo	0x1601	Receive PDO 2	no	n/a
RxPdo	0x1602	Receive PDO 3	no	n/a
RxPdo	0x1603	Receive PDO 4	no	n/a

Fixed RxPdo with contents assigned, Not-Mappable

Type	Object	Name	Fixed	SM	Used for
RxPdo	0x1700	Receive PDO 5	yes	2	CSP Cyclic Synchronous Position
	0x6040	Control word			
	0x607A	Profile target position			
	0x60B1	Velocity offset (feedforward)			
	0x60B2	Torque offset (acceleration feedforward)			
RxPdo	0x1701	Receive PDO 6	yes	n/a	CSV Cyclic Synchronous Velocity
	0x6040	Control word			
	0x60FF	Target velocity			
RxPdo	0x1702	Receive PDO 7	yes	n/a	CST Cyclic Synchronous Torque
	0x6040	Control word			
	0x6071	Target torque			
RxPdo	0x1703	Receive PDO 8	yes	n/a	UV Current Control
	0x2327	UV mode U input			
	0x2328	UV mode V input			
RxPdo	0x1704	Receive PDO 9	yes	n/a	CSTCA Cyclic Sync Torque + Commutation Angle
	0x6040	Control word			

Type	Object	Name	Fixed	SM	Used for
	0x6071	Target torque			
	0x60EA	Commutation Angle			

Non-fixed, empty TxPdo, Mappable

Type	Object	Name	Fixed	SM
TxPdo	0x1A00	Transmit PDO 1	no	n/a
TxPdo	0x1A01	Transmit PDO 2	no	n/a
TxPdo	0x1A02	Transmit PDO 3	no	n/a
TxPdo	0x1A03	Transmit PDO 4	no	n/a

Fixed TxPdo with contents assigned, Not-Mappable

Type	Object	Name	Fixed	SM	Used for
TXPDO	0x1B00	Transmit PDO 5	yes	3	CSP, CSV, CST, UV, CSTCA
	0x6041	Status word			
	0x6064	Actual motor position			
	0x60F4	Position loop error (following error)			
	0x606C	Actual motor velocity			
	0x6077	Torque actual value			
TXPDO	0x1B01	Transmit PDO 6	yes	3	CSP, CSV, CST, UV, CSTCA
	0x6041	Status word			
	0x6064	Load encoder position			
	0x60F4	Motor encoder position			
	0x606C	Analog/ Digital Reference			
	0x6077	Input pin state			

Mailbox / CoE / InitCmd SDO

0x6060   Mode of operation = 8 (CSP mode)
Configured by SDO and is set on Pre-Op to Safe-Op transition (PS)

SDO vs. PDO: Design Considerations

Differences Between SDO and PDO

As stated earlier, SDOs and PDOs can both be described as channels through which CAN messages are sent, and both provide access to a device's object dictionary, but each has characteristics that make it more appropriate for certain types of data transfers.

Here is a review of the differences between SDOs and PDOs, and some design considerations indicated by those differences:

SDO	PDO	Design Considerations
The accessed device always confirms SDO messages. This makes SDOs slower.	PDO messages are unconfirmed. This makes PDOs faster.	To transfer 8 bytes or less at real-time speed, use a PDO. For instance, to receive control instructions and transmit status updates.
One SDO transfer can send long blocks of data, using as many CAN messages as required.	A PDO transfer can only send small amounts of data (up to eight bytes) in a single CAN message. Mapping allows very efficient use of those eight bytes.	To transfer large amounts of low priority data, use the SDO. Also, if confirmation is absolutely required, use an SDO.

SDO	PDO	Design Considerations
Asynchronous.	Synchronous or asynchronous. Cyclic or acyclic.	Use PDO when synchronous or broadcast communications are required.
The SDO employs a client-server communication model. The CANopen master is the client. It reads from and writes to the object dictionaries of devices. The device being accessed is the server.	The PDO employs a peer-to-peer communication model. Any device can send a PDO message, and a PDO message can be received and processed by multiple devices.	For instance, to communicate set points from the master to multiple devices for a multi-axis move, or to have a device broadcast its status.
All communications can be performed through the SDO without using any PDOs.	The CANopen master application uses SDO messages to map the content of the PDO, at a cost of increased CPU cycles on the CANopen master and increased bus traffic.	If the application does not benefit from the use of a PDO for a certain transfer, consider using SDO to avoid the extra overhead. For instance, if an object's value is updated only once (as with many configuration objects), the SDO is more efficient. If the object's value is updated repeatedly, a PDO is more efficient.

3 HOW TO MAP (OR RE-MAP) A PDO

3.1 Process Overview

Two objects in the device's object dictionary define a PDO:

- A PDO's **communication object** defines the PDO's CAN message ID and its communication type (synchronous or asynchronous) and triggering type (event-drive or cyclic).

- A PDOs **mapping object** maps every data byte in the PDO message to an object in the device's object dictionary.

Mapping a PDO is the process of configuring the PDO's communication and mapping objects.

Note: Drives that use Firmware Feature Set C, cannot handle remote request PDOs.

3.2 To Map a Receive PDO

The general procedure for mapping a receive PDO follows. (The procedure for mapping a transmit PDO is similar).

Stage	Step	Sub-steps/Comments
1	Disable the PDO	In the PDO's mapping object (Receive PDO Mapping Parameters, index 0x1601), set the sub-index 0 (NUMBER OF MAPPED OBJECTS) to zero. This disables the PDO.
2	Set the communication parameters	If necessary, set the PDO's CAN message ID (PDO COB-ID) using sub-index 1 of the PDO's RECEIVE PDO Communication Parameters (index 0x1401). Choose the PDO's transmission type (PDO TYPE) in sub-index 2 of object 0x1401. A value in the range [0-240] = synchronous; [254-255] = asynchronous.
3	Map the data	Using the PDO's mapping parameters (sub-indexes 1-4 of Receive PDO Mapping Parameters, index 0x1601), you can map up to 4 objects (whose contents must total to no more than 8 bytes), as follows: In bits 0-7 of the mapping value, enter the size (in bits) of the object to be mapped, as specified in the object dictionary. In bits 8-15, enter the sub-index of the object to be mapped. Clear bits 8-15 if the object is a simple variable. In bits 16-31, enter the index of the object to be mapped.
4	Set the number of mapped objects and enable the PDO	In the PDO's Receive PDO Mapping Parameters (index 0x1601), set sub-index 0 (NUMBER OF MAPPED OBJECTS) to the actual number of objects mapped. This properly configures the PDO. Also, the presence of a non-zero value in the NUMBER OF MAPPED OBJECTS object enables the PDO.

Example: Mapping a Receive PDO

This example illustrates the general procedure for mapping a receive PDO. In the example, the device state change commands and to the Mode Of Operation object (index 0x6060) to receive mode change commands.

Stage	Step	Sub-steps/Comments
1	Disable the PDO.	In the PDO's mapping object (Receive PDO Mapping Parameters, index 0x1601), set the sub-index 0 (NUMBER OF MAPPED OBJECTS) to zero. This disables the PDO.
2	Set the communication parameters.	In this case, it is not necessary to set the CAN message ID of the PDO, because the default value is acceptable. In the PDO TYPE object (sub-index 2 of RECEIVE PDO Communication Parameters, index 0x1401) choose a value in the range [254-255] so that the PDO transmits immediately upon request (without waiting for a synchronization message).
3	Map the data.	In the device's Receive PDO Mapping Parameters object (index 0x1601): 1: To map the Control Word to the PDO, set object 1601, sub-index 1 to: 0x 6040 00 10 Bits 16-31 contain the index of the object to be mapped Bits 8-15 clear; the mapped object has no subindex Bits 0-7 show the size of the Control Word (16 bits) in hex 2: To map the Mode Of Operation object to the PDO, set sub-index 2 to: 0x 6060 00 08 Bits 16-31 contain the index of the object to be mapped Bits 8-15 clear; the mapped object has no subindex Bits 0-7 show the size of the Change of Mode object (8 bits) in hex
4	Set the number of mapped objects and enable the PDO.	In the PDO's Receive PDO Mapping Parameters object (index 0x1601), set sub-index 0 (NUMBER OF MAPPED OBJECTS) to 2, the actual number of objects mapped. This properly configures the PDO. Also, the presence of a non-zero value in the NUMBER OF MAPPED OBJECTS object enables the PDO.

3.3   Objects that Define SDOs and PDOs

SERVER SDO PARAMETERS   INDEX 0X1200

Type	Access	Units	Range	Map PDO	Memory
Record	RO	-	-	NO	R

Holds the COB-ID (communication object ID, also known as CAN message ID) values used to access the drive's SDO. Sub-index 0 contains the number of sub-elements of this record.

SDO RECEIVE COB-ID   INDEX 0X1200, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	0x600-0x67F	NO	R

CAN object ID used by the drive to receive SDO packets. The value is 0x600 + the drive's CAN node ID.

SDO TRANSMIT COB-ID   INDEX 0X1200, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	0x580-0x5FF	NO	R

This value gives the CAN object ID used by the drive to transmit SDO packets. The value is 0x580 + the drive's CAN node ID.

RECEIVE PDO COMMUNICATION PARAMETERS   INDEX 0x1400 – 0x1407

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	NO	R

These objects allow configuration of the communication parameters of each of receive PDO. Sub-index 0 contains the number of sub-elements of this record.

PDO COB-ID   INDEX 0X1400 – 0X1407, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See Default Values, below.	NO	R

CAN message ID used by the PDO. The ID is formatted as follows:

Bit	Description
0-10	Give the 11-bit identifier for standard (CAN 2.0A) identifiers, or the lower 11 bits for extended (CAN 2.0B) identifiers.
11-28	Give the upper 18 bits of extended identifiers. For standard identifiers, these bits should be written as zeros.
29	Defines the identifier format. This bit is clear for standard (11-bit) identifiers and set for extended (29-bit) identifiers.
30	Reserved
31	Identifies the PDO as valid if clear. If set, the PDO is disabled and its mapping may be changed.

Default Values

The default values for this object are specified in the DS-301 CANopen specification. These values are:

Index	Default ID
0x1400	0x00000200 + drive CAN node ID.
0x1401	0x00000300 + drive CAN node ID.
0x1402	0x00000400 + drive CAN node ID.
0x1403	0x00000500 + drive CAN node ID.
0x1404	0x80000000
0x1405	0x80000000
0x1406	0x80000000
0x1407	0x80000000

PDO TYPE   INDEX 0X1400 – 0X1407, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	-	See *Description*, below	NO	R

This object controls the behavior of the PDO when new data is received. The following codes are defined for receive PDOs:

Code	Behavior
0-240	The received data is held until the next SYNC message. When the SYNC message is received the data is applied.
241-253	Reserved.
254-255	The received data is applied to its mapped objects immediately upon reception.

RECEIVE PDO MAPPING PARAMETERS   INDEX 0x1600 – 0x1607

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	NO	R

EtherCAT drives support 0x1600-0x1603, CAN drives support all. These objects allow the mapping of each of the receive PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0x1600 – 0x1607, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	-	0-4	NO	R

This value gives the total number of objects mapped to this PDO. It can be set to 0 to disable the PDO operation and must be set to 0 before changing the PDO mapping.

Once the PDO mapping has been established by configuring the objects in sub-indexes 1 – 4, this value should be updated to indicate the actual number of objects mapped to the PDO.

PDO MAPPING   INDEX 0x1600 – 0x1607, SUB-INDEX 1 – 8

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See *Description*, below	NO	R

When a PDO message is received, the data passed with the PDO message (up to 8 bytes) is used to update the objects mapped to the PDO. The values in the PDO mapping objects identify which object(s) the PDO data maps to. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. Must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

RECEIVE PDO MAPPING PARAMETERS (FIXED) INDEX 0x1700

Type	Access	Units	Range	Map PDO	Memory
Record	R	-	-	NO	R

These objects allow the mapping of each of the receive PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0x1700, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	R	-	0-4	NO	R

This value gives the total number of objects mapped to this PDO.

PDO MAPPING   INDEX 0x1700, SUB-INDEX 1 – 4

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	R	-	See *Description*, below	NO	R

When a PDO message is received, the passed data (up to 8 bytes) is used to update the objects mapped to the PDO. The values in the PDO mapping objects identify which object(s) are mapped to the PDO data. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. Must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

Because Index 0x1700 is read only, the available mapped objects are fixed. They include the following:

Sub-index	Value	Description
0	4	Number of mapped objects.
1	0x60400010	Control word
2	0x607A0020	Target position
3	0x60B10020	Offset added to the velocity command in CSP or CSV mode.
4	0x60B20010	Offset added to the torque command in CSP, CSV or CST modes.

RECEIVE PDO MAPPING PARAMETERS (FIXED)   INDEX 0X1701

Type	Access	Units	Range	Map PDO	Memory
Record	R	-	-	NO	R

These objects allow the mapping of each of the receive PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0X1701, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	R	-	0-4	NO	R

This value gives the total number of objects mapped to this PDO.

PDO MAPPING   INDEX 0X1701, SUB-INDEX 1 – 3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	R	-	See *Description*, below	NO	R

When a PDO message is received, the data passed with the PDO message (up to 8 bytes) is used to update the objects mapped to the PDO. The values in the PDO mapping objects identify which object(s) are mapped to the PDO data. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. Must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

Because Index 0x1701 is read only, the available mapped objects are fixed. They include the following:

Sub-index	Value	Description
0	3	Number of mapped objects.
1	0x60400010	Control word
2	0x60FF0020	Target velocity
3	0x60B20010	Offset added to the torque command in CSP, CSV or CST modes.

RECEIVE PDO MAPPING PARAMETERS (FIXED)   INDEX 0X1702

Type	Access	Units	Range	Map PDO	Memory
Record	R	-	-	NO	R

These objects allow the mapping of each of the receive PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0X1702, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	R	-	0-4	NO	R

This value gives the total number of objects mapped to this PDO.

PDO Mapping Index 0x1702, Sub-Index 1 – 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	R	-	See *Description*, below	NO	R

When a PDO message is received, the data passed with the PDO message (up to 8 bytes) is used to update the objects mapped to the PDO. The values in the PDO mapping objects identify which object(s) are mapped to the PDO data. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc. Each is structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. Must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

Index 0x1702 is read only, the available mapped objects are fixed. They include the following:

Sub-index	Value	Description
0	2	Number of mapped objects.
1	0x60400010	Control word
2	0x60710010	Target torque

RECEIVE PDO MAPPING PARAMETERS (FIXED) INDEX 0X1703

Type	Access	Units	Range	Map PDO	Memory
Record	R	-	-	NO	R

These objects allow the mapping of each of the receive PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0X1703, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	R	-	0-4	NO	R

This value gives the total number of objects mapped to this PDO.

PDO MAPPING INDEX 0X1703, SUB-INDEX 1 – 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	R	-	See *Description*, below	NO	R

When a PDO message is received, the data passed with the PDO message (up to 8 bytes) is used to update the objects mapped to the PDO. The values in the PDO mapping objects identify which object(s) are mapped to the PDO data. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. Must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

Index 0x1703 is read only, the available mapped objects are fixed. They include the following:

Sub-index	Value	Description
0	2	Number of mapped objects.
1	0x23270010	V command
2	0x23280010	U command

RECEIVE PDO MAPPING PARAMETERS INDEX 0X1704

Type	Access	Units	Range	Map PDO	Memory
Record	R	-	-	NO	R

These objects allow the mapping of each of the receive PDO objects to be configured.

NUMBER OF MAPPED OBJECTS INDEX 0X1704, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	R	-	0-4	NO	R

This value gives the total number of objects mapped to this PDO.

PDO MAPPING INDEX 0X1704, SUB-INDEX 1 – 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	R	-	See *Description*, below	NO	R

When a PDO message is received, the data passed with the PDO message (up to 8 bytes) is used to update the objects mapped to the PDO. The values in the PDO mapping objects identify which object(s) the PDO data maps to. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. Must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

Because Index 0x1704 is read only, the available mapped objects are fixed. They include the following:

Sub-index	Value	Description
0	2	Number of mapped objects.
1	0x60400010	Control word
2	0x60710010	Target Torque
3	0x60EA0010	Phase angle

TRANSMIT PDO COMMUNICATION PARAMETERS   INDEX 0x1800 – 0x1807

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	NO	R

These objects allow configuration of communication parameters of each transmit PDO object. Sub-index 0 contains the number of sub-elements of this record.

PDO COB-ID   INDEX 0X1800 – 0X1807, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See Default Values, below.	NO	R

This object holds the CAN object ID used by the PDO. The ID is formatted as follows:

Bit	Description
0-10	11-bit identifier for standard (CAN 2.0A) identifiers, or the lower 11 bits for extended (CAN 2.0B) identifiers.
11-28	Upper 18 bits of extended identifiers. For standard identifiers, these bits should be written as zeros.
29	Identifier format. This bit is clear for standard (11-bit) identifiers and set for extended (29-bit) identifiers.
30	If set, remote transmit requests (RTR) are not allowed on this PDO. If clear, the PDO is transmitted in response to a remote request.
31	Identifies the PDO as valid if clear. If set, the PDO is disabled and its mapping may be changed.

Default Values

The default values for this object are specified in the DS-301 CANopen specification. These values are:

Index	Default ID
0x1800	0x00000180 + drive CAN node ID.
0x1801	0x00000280 + drive CAN node ID.
0x1802	0x00000380 + drive CAN node ID.
0x1803	0x00000480 + drive CAN node ID.
0x1804	0x80000000
0x1805	0x80000000
0x1806	0x80000000
0x1807	0x80000000

PDO TYPE   INDEX 0x1800 – 0x1807, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	-	See *Description*, below	EVENT	R

This object identifies which events trigger a PDO transmission:

Code	Behavior
0	The PDO is transmitted on the next SYNC message following a PDO event. See PDO Events, below, for a description of a PDO event.
1-240	The PDO is transmitted every N SYNC messages, where N is the PDO type code. For example, a PDO with type code 7 would be transmitted on every 7th SYNC message.
241-251	Reserved.
252	The PDO is transmitted on the SYNC message following a remote request.
253	The PDO is transmitted immediately in response to a remote request.
254-255	The PDO is transmitted immediately in response to an internal PDO event.

PDO Events

Some objects in the object dictionary have special PDO events associated with them. If such an object is mapped to a transmit PDO, then the PDO may be configured with a code that relies on this event to trigger its transmission. The codes that use PDO events are 0 and 255.

An example of an object that has a PDO event associated with it is the Device Status object (index 0x6041). This object triggers an event to any mapped transmit PDO each time its value changes. A transmit PDO which included this object in its mapping would have its event signaled each time the status register changed.

Most objects in the object dictionary do not have PDO events associated with them. Those that do are identified by the word **EVENT** in the *PDO Mapping* fields of their descriptions.

TRANSMIT PDO MAPPING PARAMETERS   INDEX 0x1A00 – 0x1A07

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	NO	R

EtherCAT drives support 0x1A00-0x1A03, CAN drives support all. These objects allow the mapping of each of the transmit PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0x1A00 – 0x1A07, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	-	0-4	NO	R

Total number of objects mapped to this PDO. It can be set to 0 to disable the PDO operation and must be set to 0 before changing the PDO mapping.

Once the PDO mapping has been established by configuring the objects in sub-indexes 1 – 4, this value should be updated to indicate the actual number of objects mapped to the PDO.

PDO MAPPING   INDEX 0x1A00 – 0x1A07, SUB-INDEX 1 – 8

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See *Description*, below	NO	R

When a PDO message is transmitted, the data passed with the PDO message (up to 8 bytes) is gathered from the objects mapped to the PDO. The values in the PDO Mapping objects identify which object(s) the PDO data maps to. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. This value must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

TRANSMIT PDO MAPPING PARAMETERS   INDEX 0x1B00

Type	Access	Units	Range	Map PDO	Memory
Record	RO	-	-	NO	R

These objects allow the mapping of each of the transmit PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0x1B00, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RO	-	0-4	NO	R

Total number of objects mapped to this PDO. Once the PDO mapping has been established by configuring the objects in sub-indexes 1 – 5, this value should be updated to indicate the actual number of objects mapped to the PDO.

PDO MAPPING   INDEX 0x1B00, SUB-INDEX 1 – 5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below	NO	R

When a PDO message is transmitted, the data passed with the PDO message (up to 8 bytes) is gathered from the objects mapped to the PDO. The values in the PDO Mapping objects identify which object(s) the PDO data maps to. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. This value must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

Because Index 0x1B00 is read only, the objects that can be mapped are fixed. The available mapped objects are as follows:

Sub-index	Value	Description
0	5	Number of mapped objects.
1	0x60410010	Status word
2	0x60640020	Actual position
3	0x60F40020	Position error
4	0x606C0020	Actual velocity
5	0x60770010	Actual torque

TRANSMIT PDO MAPPING PARAMETERS   INDEX 0x1B01

Type	Access	Units	Range	Map PDO	Memory
RECORD	RO	-	-	NO	R

These objects allow the mapping of each of the transmit PDO objects to be configured.

NUMBER OF MAPPED OBJECTS   INDEX 0x1B01, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RO	-	0-4	NO	R

Total number of objects mapped to this PDO. Once the PDO mapping has been established by configuring the objects in sub-indexes 1 – 5, this value should be updated to indicate the actual number of objects mapped to the PDO.

PDO MAPPING   INDEX 0x1B01 SUB-INDEX 1 – 5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below	NO	R

When a PDO message is transmitted, the data passed with the PDO message (up to 8 bytes) is gathered from the objects mapped to the PDO. The values in the PDO Mapping objects identify which object(s) are mapped to the PDO data. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. This value must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

Because Index 0x1B01 is read only, the objects that can be mapped are fixed. The available mapped objects are as follows:

Sub-index	Value	Description
0	5	Number of mapped objects.
1	0x60410010	Status word
2	0x22420020	Load encoder position
3	0x22400020	Motor encoder position
4	0x22000010	Analog/Digital reference input value
5	0x219A0020	Input pin state

SYNC MANAGER TYPE   INDEX 0X1C00

Type	Access	Units	Range	Map PDO	Memory
UINT8	RO	-	-	NO	R

This object gives the number and type of the communication channels.

Sub-index	Value
0	4: number of another available sub-index
1	SM 0: Mailbox Write
2	SM 1: Mailbox Read
3	SM 2: Process Data Write
4	SM 3: Process Data Read

SYNC MANAGER 2 PDO ASSIGNMENT OBJECT   INDEX 0X1C12-0X1C13

Type	Access	Units	Range	Map PDO	Memory
Record	R	-	-	NO	R

This object is used to assign a sync manager to PDOs.

NUMBER OF MAPPED OBJECTS   INDEX 0X1C12   0X1C13, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	-	0-4	NO	R

This value gives the total number of objects mapped to this PDO. It can be set to 0 to disable the PDO operation and must be set to 0 before changing the PDO mapping.

Once the PDO mapping has been established by configuring the objects in sub-indexes 1 – 4, this value should be updated to indicate the actual number of objects mapped to the PDO.

PDO MAPPING   INDEX 0X1C12   0X1C13, SUB-INDEX 1 – 8

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See *Description*, below	NO	R

When a PDO message is received, the data passed with the PDO message (up to 8 bytes) is used to update the objects mapped to the PDO. The values in the PDO mapping objects identify which object(s) are mapped to the PDO data. The first object is specified by the value in sub-index 1; the second object is identified by sub-index 2, etc.

Each of the PDO mapping values consist of a 32-bit value structured as follows:

Bit	Description
0-7	Size (in bits) of the object being mapped. Must match the actual object size as defined in the object dictionary.
8-15	Sub-index of the object to be mapped.
16-31	Index of the object to be mapped.

4 NETWORK MANAGEMENT

4.1   Network Management Overview: CANopen

Overview

Network Management Services and Objects

Network management services on the CANopen network include device state control, device monitoring, synchronization, and emergency handling. Special communication objects, as summarized below, provide these services.

Object	Description
Network Management (NMT)	This object provides services to control the state of the device, including the initialization, starting, monitoring, resetting, and stopping of nodes. It also provides device-monitoring services (node-guarding and heartbeat).
Synchronization (SYNC)	Broadcast periodically by a specified device or the CANopen master to allow synchronized activity among multiple devices. The CAN message ID of the SYNC message is 80.
Time Stamp	Broadcast periodically by a specified device or the CANopen master to allow devices to synchronize their clocks.
Emergency	Transmitted by a device when an internal error occurs.

Network Manager Node

Normally, a single node (such as a PC) is designated as the network manager. The network manager runs the software that issues all NMT messages. The network manager node can be the same node that runs the CANopen master application.

General Device State Control

State Machine

Every CANopen device implements a simple state machine. The machine defines three states (described below). The network manager application uses NMT messages to interact with the state machine and control state changes.

Device States

The following states are defined for Copley Controls CANopen drives:

State	Description
Pre-operational	Every node enters this state after power-up or reset. In this state, the device is not functional, but will communicate over the CANopen network. PDO transfers are not allowed in pre-operational state, but SDO transfers may be used.
Operational	This is the normal operating state for all devices. SDO and PDO transfers are both allowed.
Stopped	No communication is allowed in this state except for network management messages. Neither SDO nor PDO transfers may be used.

State Control Messages

One use of NMT messages is to control state changes on network devices. The following NMT messages are sent by the network manager to control these state changes. Each of these messages can be either sent to a single node (by node ID), or broadcast to all nodes.

Message	Effect
Reset	Causes each receiving node to perform a soft reset and come up in pre-operational state.
Reset communications	Causes each receiving node to reset its CANopen network interface to power-on state and enter pre-operational state. This is not a full device reset, just a reset of the CANopen interface.
Pre-operational	Causes the receiving node(s) to enter pre-operational state. No reset is performed.
Start	Causes the node(s) to enter operational state.
Stop	Causes the node(s) to enter stopped state.

Device Monitoring

CANopen

Monitoring Protocols

In addition to controlling state machines, NMT messages provide services for monitoring devices on the network. Monitoring services use one of two protocols: heartbeat and node guarding.

Heartbeat Protocol

The heartbeat protocol allows the network manager application to detect problems with a device or its network connection. The CANopen master configures the device to periodically transmit a heartbeat message indicating the device's current state (pre-operational, operational, or stopped). The network manager monitors the heartbeat messages. Failure to receive a node's heartbeat messages indicates a problem with the device or its connection to the network. See 0x1017.

Node-guarding Protocol

The node-guarding protocol is similar to the heartbeat, but it allows both the device and the network manager to monitor the connection between them. The network manager configures the device (node) to expect node-guarding messages at some interval. The network manager then sends a message to the configured device at that frequency, and the device responds with a node-guarding message. This allows both the network manager and the device to identify a network failure if the guarding messages stop. See 0x100C, and 0x100D.

EtherCAT

The EtherCAT master controls a Watchdog function that is sent with each Sync Manager. Via the master, settings can be made for the time-out, and to enable/disable the WD. These are below the level of SDO and PDO and are not covered in this document.

SYNC and High-resolution Time Stamp Messages

The SYNC message is a standard CANopen message used to synchronize multiple devices and to trigger the synchronous transmission of PDOs.

In addition, to allow more accurate synchronization of device clocks, Copley Controls CANopen drives use the optional high-resolution time stamp message specified in the Communication Profile.

Normally, a single device produces both the SYNC message and the high-resolution time stamp message. Copley drives can produce the SYNC and high-resolution time stamp messages.

We recommend using an drive as the master sync generator. This assures greater timing accuracy and allows the drive PVT segment buffer to be filled with the minimum number of PVT segments at all times during operation.

Time Stamp PDOs

The device designated as the time stamp producer should have a transmit PDO mapped for the high-resolution time stamp message. This PDO should be configured for synchronous transmission, based on the SYNC message. We recommend sending this message approximately every 100 milliseconds.

Every other device (all time stamp consumers) should have a receive PDO mapped for the high-resolution time stamp message. The message ID of each receive PDO used to receive a time stamp should match the ID of the transmit PDO used to send the time stamp.

Configuring the devices in this fashion causes the time stamp producer to generate a transmit PDO for every N sync messages. This PDO is received by each of the time stamp consumers on the network and causes them to update their internal system times based on the message content. The result is that all devices on the network act as though they share the same clock input and remain tightly synchronized.

Emergency Messages

A device sends an 8-byte emergency message (EMCY) when an error occurs in the device. It contains information about the error type, and Copley-specific information. A device need only send one EMCY message per event. Any device can be configured to accept EMCY messages.

EMCY Message Structure

The EMCY message is structured as follows:

Bytes	Description
0, 1	Standard CANopen emergency error code for errors active on the drive. See EMCY Message Error Codes
2	Error register object value See Error Register
3	Reserved for future use (0 for now).
4, 5	Bit mask representing the Copley Controls codes for active error conditions on the drive (see EMCY Message Copley-Specific Error Conditions).
6, 7	Reserved for future use (0 for now).

EMCY Message Error Codes

Bytes 0 and 1 of the EMCY message describe the standard CANopen error codes used by Copley Drives:

Error Code (hex)	Description
2280	Encoder Feedback Error
2310	Current Limited
2320	Short Circuit
3110	Mains Over Voltage
3120	Mains Under Voltage
3310	Output Voltage Limited
4210	Drive Over Temperature
4300	Motor Temperature Sensor
5080	Drive error
7122	Phasing Error
7380	Positive Limit Switch
7381	Negative Limit Switch
7390	Tracking Error
73A0	Position Wrapped Around +/- 2^{31} Counts
8130	Node Guarding Error or Heartbeat Error

EMCY Message Copley-Specific Error Conditions

The bit mask in bytes 4 and 5 of the EMCY message maps 1 bit for each error condition active on the drive. The mapped bits have the following meanings:

Bit	Description
0	Output short circuit
1	Drive over temperature
2	Drive over voltage
3	Drive under voltage
4	Motor over temperature input active
5	Encoder power error (indicates the 5V encoder supply over current)
6	Motor phasing error
7	Output current limited
8	Output voltage limited
9	Positive limit switch
10	Negative limit switch
11	Tracking error
12	Position input wrapped around +/- 2^{31} bits
13	Latched Fault
14	Node guarding error

4.2 Network Management Objects: CANopen

COB ID SYNC MESSAGE   INDEX 0X1005

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See SYNC ID Format, below.	NO	R

This object defines the CAN object ID (COB-ID) associated with the SYNC message. The SYNC message is a standard CANopen message type used to synchronize multiple devices on a CANopen network.

SYNC ID Format

The SYNC message ID is formatted as follows:

Bits	Description
0-10	Give the 11-bit identifier for standard (CAN 2.0A) identifiers, or the lower 11 bits for extended (CAN 2.0B) identifiers.
11-28	Give the upper 18 bits of extended identifiers. For standard identifiers, these bits should be written as zeros.
29	Identifier format. This bit is clear for standard (11-bit) identifiers and set for extended (29-bit) identifiers.
30	If set, the drive is configured as the SYNC message producer. This bit should be set in at most one drive on a network.
31	Reserved

COMMUNICATION CYCLE PERIOD   INDEX 0X1006

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	microseconds	-	NO	R

This object defines the interval between SYNC messages in units of microseconds.

An drive configured as a SYNC message producer will not produce SYNC messages unless this object contains a non-zero value. A value of zero in this object disables SYNC message production.

Drives not configured to produce SYNC messages ignore the value of this object.

GUARD TIME   INDEX 0X100C

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	-	NO	R

This object gives the time between node-guarding requests that are sent from the network master to this drive. The drive will respond to each request with a node-guarding message indicating the internal state of the drive.

If the drive has not received a node-guarding request within the time period defined by the product of the guard time and the Life Time Factor (index 0x100D, p. 49), the drive will treat this lack of communication as a fault condition.

LIFE TIME FACTOR   INDEX 0x100D

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	-	-	NO	R

This object gives a multiple of the GUARD Time (index 0x100C, p. 48). The drive expects to receive a node-guarding request within the time period defined by the product of the guard time and the lifetime factor. If the drive has not received a node-guarding request within this time period, it treats this condition as a fault.

HIGH-RESOLUTION TIME STAMP   INDEX 0x1013

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	microseconds	0 - 294,967,295	YES	R

This object holds a time stamp indicating the drive's internal time (in microseconds) when the last SYNC message was received (or transmitted for the SYNC producer). Writing to this object will cause the drive to adjust its internal clocks to reconcile the difference between the value passed and the internal value of the time stamp.

The purpose of this object is to allow multiple drives to synchronize their clocks across the CANopen network. To enable this feature, one drive should be selected as a high-resolution time stamp producer. This drive should have a transmit PDO configured to transmit this object to the rest of the network at a rate of approximately 10 Hertz (once every 100 milliseconds).

Every other drive should have a receive PDO configured (using the same COB-ID as the producer's transmit PDO) to update its time stamp using the value passed by the producer.

EMERGENCY OBJECT ID   INDEX 0x1014

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	-	NO	R

CAN message ID used with the emergency object. See Emergency Messages, p. 46 and the *CANopen Application Layer and Communication Profile (DS 301).*

EMERGENCY OBJECT ID INHIBIT TIME   INDEX 0x1015

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	-	NO	R

Inhibit time for the emergency object. See Emergency Messages, p. 46 and the *CANopen Application Layer and Communication Profile (DS 301).*

PRODUCER HEARTBEAT TIME   INDEX 0x1017

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	-	NO	R

This object gives the frequency at which the drive will produce heartbeat messages. This object may be set to zero to disable heartbeat production. Note that only one of the two node-guarding methods may be used at once. If this object is non-zero, then the heartbeat protocol is used regardless of the settings of the node-guarding time and lifetime factor.

4.3   Network Management Overview: EtherCAT/CANopen

NETWORK OPTIONS   INDEX 0X21B3

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Bits	Bit Mapped	NO	RF

	Network options. Configures the drive's network.
EtherCAT	

Bits	Meaning
0	If set, disable some extra checks of the SYNC0 configuration which were added for improved network conformance.
1	If set, the drive will follow the EtherCAT state machine even when running in a non-EtherCAT mode of operation.
2	If set, object 0x1002 Event Status is the bit-wise OR of all axes Event Status for multi-axis drives. If clear, 0x1002 is for axis 1 only of a multi-axis drive.
3	If set, the drive station alias is stored in flash. If clear the station alias is set from switches.
4~7	Reserved
8	If set, PDO mapping will be saved to flash in addition to the parameters that are saved using 0x1010 Save Parameters object.
9	Set for UDP command mode. Clear for EtherCAT mode.
10~15	Reserved

CANopen	

Bits	Meaning
0	Must be clear to select CANopen networking.
1~2	Reserved
3	If set, use an alternative method of assigning node IDs to each axis. See parameter 0x21B1 Input Mapping for Network Node ID for a description of this method.
4~7	Reserved
8	If set, PDO mapping will be saved to flash in addition to the parameters that are saved using 0x1010 Save Parameters object.
9	If set, PDO communication settings will be saved to flash in addition to the parameters that are saved using 0x1010 Save Parameters object.
10~15	Reserved

Note: 0x1002 is not in the range of objects that can use the 0x800 offset for axes in multi-axis drives.

For a multi-axis drive, setting bit 2 of the Network Options combines the individual axis events into a single register, 0x1002, by ORing the events of all axes.

NETWORK STATUS WORD   INDEX 0X21B4

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	-	-	T	R

Network status word. Bit mapped as follows:

CANopen

Bits	Meaning		
0-1	CANopen node status. This field will take one of the following values:		
	Value	Status	
	0	The CANopen interface is disabled.	
	1	Stopped mode.	
	2	Preoperational mode.	
	3	Operational mode.	
4	Set if the CANopen SYNC message is missing.		
5	Set on a CANopen guard error.		
8	Set if the CAN port is in 'bus off' state.		
9	Set if the CAN port is in 'transmit error passive' state.		
10	Set if the CAN port is in 'receive error passive' state.		
11	Set if the CAN port is in 'transmit warning' state.		
12	Set if the CAN port is in 'receive warning' state.		

DeviceNet

Bit	Meaning
0	Set if duplicate MAC ID check failed.
1	Set if device is online.
2	Set if at least one communication object timed out.
3	Set if at least one communication object has been established.
4-7	Reserved.
8-14	Same bit mapping as for CANopen.
15	Always set for DeviceNet.

MACRO

Bit	Meaning
0	Set if the MACRO network is detected,
1	Set if the drive is being disabled by the MACRO master.
2	Set if the MACRO network has been broken (i.e. once detected but now gone).
3	Set on heartbeat error.
4-15	Reserved

EtherCAT

Bit	Meaning
0	Set if distributed clock is enabled (SYNC0 enabled and period set to a legal value).
1	Set if distributed clock is locked.
2	If the distributed clock is locked, this bit identifies whether it is locked to the current loop period (0), or position loop period (1).
3-15	Reserved

TIMESTAMP INDEX 0x10F8

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED64	RO	ns	$0-2^{64}-1$	R	R

Local timestamp of the device in nanoseconds.

DIAGNOSTIC HISTORY   INDEX 0x10F3

Type	Access	Units	Range	Map PDO	Memory
ARRAY	RW	-	-	NO	R

History of diagnosis messages up to 100.

MAXIMUM MESSAGES   INDEX 0x10F3, SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RO	8	0 to 2^8-1	NO	R

Read: Maximum number of diagnosis messages which can be stored in the diagnosis history (SubIndex 6 onwards)

Write: For devices with dynamic memory the number of stored diagnosis messages may be adapted by the EtherCAT master.

NEWEST MESSAGE   INDEX 0x10F3, SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RO	8	0 to 2^8-1	NO	R

SubIndex of the newest diagnosis message (6-105)

Default value = 0

NEWEST ACKNOWLEDGEMENT   INDEX 0x10F3, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	8	0 to 2^8-1	NO	R

Default value = 0

Overwrite Mode (SI5, Bit4 = 0)	Acknowledge Mode (SI5, Bit4 =1)
Read = 0: When the message queue will be overwritten, the slave shall set SI3 to 0	Read = 0: No messages have been acknowledged so far Read <> 0: SubIndex of latest acknowledged diagnosis message (6-255)
Writing = 0: (support optional) the slave will clear all messages, i.e. resetting SI2, SI3, SI4 and Si5 Bit 5	Writing = 0: (support optional) All acknowledged messages will be deleted.
Writing = 1…5: The slave shall return SDO-Abort with codes 0x06090030 (value range of parameter exceeded) or 0x06090032 (value of parameter written too low)	Writing = 1…5: The slave shall return SDO-Abort with codes 0x06090030 (value range of parameter exceeded) or 0x06090032 (value of parameter written too low)
Writing = 6…[SI0] 1: SI3 = Written value without checking	Writing = 6…[SI0]: Messages are acknowledged 0
Writing > [SI0]…255: SDO-Abort with codes 0x06090030 or 0x06090031(value of parameter written too high)	Writing > [SI0]…255: SDO-Abort with codes 0x06090030 or 0x06090031(value of parameter written too high)

New Messages Available   Index 0x10F3, Sub-Index 4

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RO	8	0, 1	NO	R

Overwrite Mode	Acknowledge Mode
0: newest message was read	0: no unacknowledged message
1: newest message was not read	1: diagnosis messages are available which can be acknowledged (SI2 <> SI3)

Flags   Index 0x10F3, Sub-Index 5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	16	0 to $2^{16}-1$	NO	R

Flags to control the sending and storing of diagnosis messages

Which functionality the slave supports is described by the ESI / SII.

When writing SI5, the read-only bits should match the current values. Bit5 shall be "don't care".

The slave should send an abort with 0x6090030 Value exceeded in case the read-only bits differ from current values

Bit	Description
0	Enable Emergency sending (according to ETG.1000-6) 0: default if device does not support Emergency sending 1: new diagnosis messages shall be sent as Emergency message
1	Disable info messages 0: Info messages are stored in the diagnosis message queue (default) 1: Info messages will not be stored in the diagnosis message queue
2	disable warning messages 0: Warning messages are stored in the diagnosis message queue (default) 1: Warning messages will not be stored in the diagnosis message queue
3	disable error messages 0: Error messages are stored in the message queue (default) 1: Error messages will not be stored in the diagnosis message queue
4	Mode selection for diagnosis history handling 0: Overwrite Mode old messages are overwritten by new ones when buffer is full 1: Acknowledge Mode New messages do only overwrite messages which were acknowledged before
5	Overwrite/Discard Information In Overwrite Mode: 1: unacknowledged messages have been overwritten (=buffer overrun) (SI3 is set to 0, too) In Acknowledge Mode: 1: message buffer is full with unacknowledged messages and a new massage is discarded.
6-15	Reserved

DIAGNOSIS MESSAGE   INDEX 0X10F3, SUB-INDEX 6-105

Type	Access	Units	Range	Map PDO	Memory
OCTET STRING	RO	-	-	NO	R

Diagnosis message buffer.

Depending on SI1 the EtherCAT slave can store up to 250 diagnosis messages; the first message is stored in SubIndex 6, the second in SubIndex 7 and so on. When the queue is full, the EtherCAT slave shall overwrite SubIndex 6 and so on, that always the latest maximum messages (SI1) shall be accessible by the EtherCAT master.

For more information on Diagnosis Message, refer to *ETG1020 Protocol Enhancements.*

4.4　Sending Serial Commands over CANopen and EtherCAT

Overview

CANopen object 0x2000 (sub-index 0) is used to send serial commands and retrieve the response from the drive. Each serial command consists of two parts, a command message sent to the drive, and a response message retrieved from it.

Sending a command to the drive is done by writing to CANopen object 0x2000. The first byte sent is the command code of the serial command to be executed. This is followed by any data bytes that are required for the command. Then, the response from the drive is retrieved by reading from object 0x2000. The first byte received will be an error code (same error codes as used in the serial interface). This is followed by zero or more bytes of response data.

For example, to read actual position, the following bytes would be written to object 0x2000 using an SDO transfer:

> 0x0C 0x17 0x00

The first byte (0x0C) is the command code for a GET command. The second and third bytes (0x17 0x00) make up the one word of data passed to a GET command. This data word (0x0017) is the variable ID that is to be read (in this case, variable 0x17, which is the actual position). The response is read from an SDO reading back the value of object 0x2000.

For example, if the following data bytes were read from 0x2000:

> 0x00 0x34 0x12 0x78 0x56

The first byte gives an error code. A zero here indicates no error. The next four bytes are the position read back from the drive. In this case, the position read back is 0x12345678.

Byte order

The byte order of data sent to or from the drive requires some further explanation.

The drive (serial port interface) works internally with 16-bit words of data. All serial commands take zero or more words of data and return zero or more words. When 32-bit values are passed to or from the drive, they are always sent most significant word first. When this array of 16-bit words of data is sent over the CANopen interface, each word of data is split into two bytes. CANopen always sends data least significant byte first. Therefore, when a 32-bit value is sent over the CANopen interface, it's first split into two 16-bit words (most significant word followed by least significant word). Then, each word is split into two bytes using the CANopen standard of least significant byte followed by most significant.

For example, the 32-bit value 0x12345678 would first be split into the words 0x1234 0x5678. These two words would then be split into the bytes 0x34 0x12 0x78 0x56.

Any serial command that is processed by the main drive firmware (as opposed to the boot loader) can be sent over the CANopen interface using this method. Any command that needs to be sent to the boot loader (such as a firmware upload) cannot be sent using this method.

SERIAL PORT COMMAND SEND　　INDEX 0x2000

Type	Access	Bits	Range	Map PDO	Memory
ARRAY [0..16383] of UNSIGNED16	RW	262144	0 to $2^{16}-1$ for each entry	No	RF

Used to send serial port commands over a CANopen/EtherCAT bus.

5 DEVICE CONTROL, CONFIGURATION, & STATUS

5.1 Device Control and Status Overview

Control Word, Status Word, and Device Control Function

Device Control Function Block

The *CANopen Profile for Drives and Motion Control (DSP 402)* describes control of the drive in terms of a control function block with two major sub-elements: the operation modes and the state machine.

Control and Status Words

As illustrated below, the Control Word object (index 0x6040, p. 61) manages device mode and state changes. The Status Word object (index 0x6041, p. 61) identifies the current state of the drive. The Mode Of Operation object (index 0x6060, p. 67) sets the drive's operating mode.
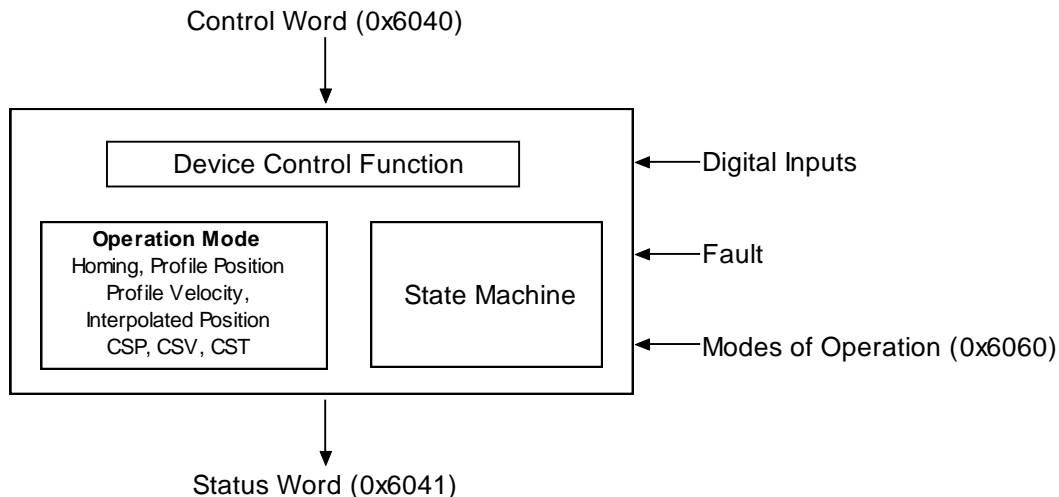
Other factors affecting control functions include digital input signals, fault conditions, and settings in various dictionary objects.

Operation Modes

As controlled by the Mode Of Operation object (index 0x6060, p. 67), Copley Controls CANopen drives support homing, profile position, profile velocity, profile torque, and interpolated position modes.

State Machine Nesting

Note that the Communication Profile also specifies a state machine, with three states: pre-operational, operational, and stopped. The entire device control function block described in this chapter, including the device state machine, operates in the operational state of the Communication Profile state machine.

State Machine and States

The state machine describes the status and possible control sequences of the drive. The state also determines which commands are accepted.

States are described below:

State	Description
Not Ready to Switch On	Low-level power (e.g. _ 15V, 5V) has been applied to the drive. The drive is being initialized or is running self-test. A brake, if present, is applied in this state. The drive function is disabled.
Switch On Disabled	Drive initialization is complete. The drive parameters have been set up. Drive parameters may be changed. The drive function is disabled.
Ready to Switch On	The drive parameters may be changed. The drive function is disabled.
Switched On	High voltage has been applied to the drive. The power drive is ready. The drive parameters may be changed. The drive function is disabled.
Operation Enable	No faults have been detected. The drive function is enabled, and power is applied to the motor. The drive parameters may be changed. (This corresponds to normal operation of the drive.)
Quick Stop Active	The drive parameters may be changed. The quick stop function is being executed. The drive function is enabled, and power is applied to the motor. If the 'Quick-Stop-Option-Code' is switched to 5 (Stay in Quick-Stop), the drive cannot exit the Quick-Stop-State but can be transmitted to 'Operation Enable' with the command 'Enable Operation."
Fault Reaction Active	The drive parameters may be changed. A non-fatal fault has occurred in the drive. The quick stop function is being executed. The drive function is enabled, and power is applied to the motor.
Fault	The drive parameters may be changed. A fault has occurred in the drive. The drive function is disabled.

State Changes Diagram

Diagram

The following diagram from the *CANopen Profile for Drives and Motion Control (DSP 402)* shows the possible state change sequences of an drive. Each transition is numbered and described in the legend below.
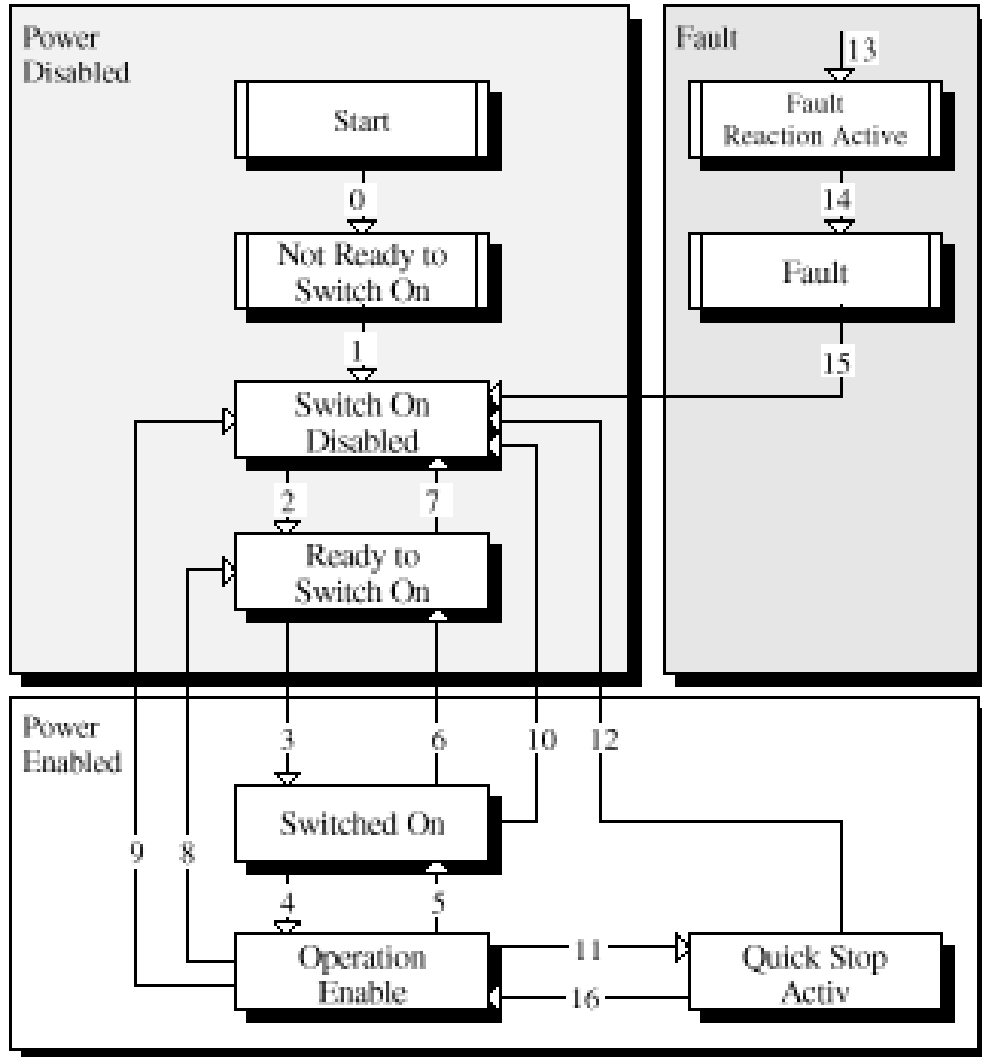
Figure 6—CANopen motion control system

State Changes Diagram Legend

	From State	To State	Event/Action
0	Startup	Not Ready to Switch On	Event: Reset. Action: The drive self-tests and/or self-initializes.
1	Not Ready to Switch On	Switch On Disabled	Event: The drive has self-tested and/or initialized successfully. Action: Activate communication and process data monitoring
2	Switch On Disabled	Ready to Switch On	Event: 'Shutdown' command received from host. Action: None
3	Ready to Switch On	Switched On	Event: 'Switch On' command received from host. Action: The power section is switched on if it is not already switched on.
4	Switched On	Operation Enable	Event: 'Enable Operation' command received from host. Action: The drive function is enabled.
5	Operation Enable	Switched On	Event: 'Disable Operation' command received from host. Action: The drive operation is disabled.
6	Switched On	Ready to Switch On	Event: 'Shutdown' command received from host. Action: The power section is switched off.
7	Ready to Switch On	Switch On Disabled	Event: 'Quick stop' command received from host. Action: None
8	Operation Enable	Ready to Switch On	Event: 'Shutdown' command received from host. Action: The power section is switched off immediately, and the motor is free to rotate if unbraked
9	Operation Enable	Switch On Disabled	Event: 'Disable Voltage' command received from host. Action: The power section is switched off immediately, and the motor is free to rotate if unbraked
10	Switched On	Switch On Disabled	Event: 'Disable Voltage' or 'Quick Stop' command received from host. Action: The power section is switched off immediately, and the motor is free to rotate if unbraked
11	Operation Enable	Quick Stop Active	Event: 'Quick Stop' command received from host. Action: The Quick Stop function is executed.
12	Quick Stop Active	Switch On Disabled	Event: 'Quick Stop' is completed or 'Disable Voltage' command received from host. This transition is possible if the Quick-Stop-Option-Code is not 5 (Stay in Quick-Stop) Action: The power section is switched off.
13	FAULT	Fault Reaction Active	A fatal fault has occurred in the drive. Action: Execute appropriate fault reaction.
14	Fault Reaction Active	Fault	Event: The fault reaction is completed. Action: The drive function is disabled. The power section may be switched off.
15	Fault	Switch On Disabled	Event: 'Fault Reset' command received from host. Action: A reset of the fault condition is carried out if no fault exists currently on the drive.

	From State	To State	Event/Action
			After leaving the 'Fault' state the Bit 'Fault Reset' of the Control Word has to be cleared by the host.
16	Quick Stop Active	Operation Enable	Event: 'Enable Operation' command received from host. This transition is possible if the Quick-Stop-Option-Code is 5, 6, 7, or 8 (see the Quick Stop Option Code object, index 0x6085, p. 65). Action: The drive function is enabled.

5.2 Device Control and Status Objects

Control & Status Objects

CONTROL WORD   INDEX: 0x6040

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below.	EVENT	R

This object is used to controls the state of the drive. It can be used to enable / disable the drive output, start, and abort moves in all operating modes, and clear fault conditions.

Control Word Bit Mapping

The value programmed into this object is bit-mapped as follows:

Bits	Description
0	Switch On. This bit must be set to enable the drive.
1	Enable Voltage. This bit must be set to enable the drive.
2	Quick Stop. If this bit is clear, then the drive is commanded to perform a quick stop.
3	Enable Operation. This bit must be set to enable the drive.
4-6	Operation mode specific. Descriptions appear in the sections that describe the various operating modes. Also see Mode Of Operation (index 0x6060, p. 67).
7	Reset Fault. A low-to-high transition of this bit makes the drive attempt to clear any latched fault condition.
8	Halt. If the bit is set, the drive will perform a halt.
9-15	Reserved

STATUS WORD   INDEX 0x6041

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	-	See *Description*, below.	Event	R

This object identifies the current state of the drive and is bit-mapped as follows:

Bits	Description
0	Ready to switch on.
1	Switched on.
2	Operation Enabled. Set when the drive is enabled.
3	Fault. If set, a latched fault condition is present in the drive.
4	Voltage enabled. Set if the drive bus voltage is above the minimum necessary for normal operation.
5	Quick Stop. When clear, the drive is performing a quick stop.
6	Switch on disabled.
7	Warning. Set if a warning condition is present on the drive. Read the Manufacturer Status Register object (index 0x1002, p. 62) for details of what warning is bit indicates.
8	Set if the last trajectory was aborted rather than finishing normally.

Bits	Description
9	Remote. Set when the drive is being controlled by the CANopen interface. When clear, the drive may be monitored through this interface, but some other input source is controlling it. Other input sources include the serial port, drive CVM program, analog reference input, digital command signals (i.e. PWM input or master controller), and internal function generator. The input source is controlled by the 'drive desired state' value, which is normally programmed by the CME-2 software. This setting can be manipulated through the CANopen interface through the Desired State object (index 0x2300, p. 68).
10	Target Reached. This bit is set when the drive is finished running a trajectory, and the Position Error (index 0x60F4, p. 144) has been within the Position Tracking Window (index 0x6067, p. 143) for the programmed time. The bit is not cleared until a new trajectory is started.
11	Internal Limit Active. This bit is set when one of the drive limits (current, voltage, velocity or position) is active. The specific bits from the Manufacturer Status Register (index 0x1002, p. 62) that cause this bit to be set can be customized by using the mask defined in the Limit Status Mask object (index 0x2184, p. 64).
12-13	The meanings of these bits are operation mode specific:

Bit	Profile Position Mode	Profile Velocity Mode	Profile Torque Mode	Homing Mode	Interpolated Position Mode
12	Set point acknowledge	Speed = 0	Reserved	Homing attained	Interpolated pos. mode active
13	Following error	Maximum slippage error	Reserved	Homing error	Reserved

For information on operation modes, see Mode Of Operation (index 0x6060, p. 67).

Bits	Description
14	Set when the drive is performing a move and cleared when the trajectory finishes. This bit is cleared immediately at the end of the move, not after the motor has settled into position.
15	Set if the home position has been captured, cleared if it hasn't.

MANUFACTURER STATUS REGISTER   INDEX 0X1002

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below.	T	R

This object only is used with single-axis drives. For multi-axis drives, use 0x2185
This 32-bit object is a bit-mapped status register with the following fields:

Bit	Description
0	Short circuit detected
1	Drive over temperature
2	Over voltage
3	Under voltage
4	Motor temperature sensor active
5	Feedback error
6	Motor phasing error
7	Current output limited
8	Voltage output limited

Bit	Description
9	Positive limit switch active
10	Negative limit switch active
11	Enable input not active
12	Amp is disabled by software
13	Trying to stop motor
14	Motor brake activated
15	PWM outputs disabled
16	Positive software limit condition
17	Negative software limit condition
18	Tracking error
19	Tracking warning
20	Drive is currently in a reset condition
21	Position has wrapped. The Position variable cannot increase indefinitely. After reaching a certain value the variable rolls back. This type of counting is called position wrapping or modulo count.
22	Drive fault. See the fault latch for more info.
23	Velocity limit has been reached.
24	Acceleration limit has been reached.
25	Position Error (index 0x60F4, p. 144) is outside Position Tracking Window (index 0x6067, p. 143).
26	Home switch is active.
27	In motion. This bit is set if the trajectory generator is running a profile OR the Position Error (index 0x60F4, p. 144) is outside the Position Tracking Window (index 0x6067, p. 143). This is clear when the drive is settled in position.
28	Velocity window. Set if the absolute velocity error exceeds the velocity window value.
29	Phase not yet initialized. If the drive is phasing with no Halls, this bit is set until the drive has initialized its phase.
30	Command fault. PWM or another command signal not present. If **Allow 100% Output** option is enabled, by a setting Bit 3 of Digital Input Command Configuration (Object 0x2320, p. 119), this fault will not detect a missing PWM command.

Status Registers for Multi-Axis Drives

Object 0x1002 shows the Event Status for single axis drives. Optionally, setting bit 2 of 0x21B3 will configure 0x1002 to be a logical OR of the Event Status objects of all of the axes of EtherCAT drives. To show the Event Status of individual axes in multi-axis EtherCAT drives, use the objects shown below. Multi-axis CANopen drives use 0x1001 for each axis which look like independent nodes on the network.

DRIVE EVENT WORD   INDEX 0X2185

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	Bit mapped	T	R

Holds the same status register as 0x1002, but is in the object range that allows it to be used for multi-axis products.

Example:
Axis A   0x2185
Axis B   0x2985
Axis C   0x3185
Axis D   0x3985

'STICKY' EVENT STATUS REGISTER   INDEX 0X2180

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RO	32	Bit mapped	T	R

Sticky Drive Event Status Register. This read-only parameter is bit-mapped in exactly the same way as the Manufacturer Status Register (index 0x1002, p. 62), but instead of giving the present status of the drive, the sticky version indicates any bits in the Manufacturer Status Register that have been set since the last reading of the sticky register.

The sticky register is similar to the Latched Event Status Register (index 0x2181, p. 64), but the latched register must be cleared explicitly, whereas the sticky register is cleared automatically each time it is read.

LATCHED EVENT STATUS REGISTER   INDEX 0X2181

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	Bit mapped	TR	R

This is a latched version of the Manufacturer Status Register object (index 0x1002, p. 62). Bits are set by the drive when events occur. Bits are cleared only by a set command.

When writing to the Latched Event Status Register, any bit set in the written value will cause the corresponding bit in the register to be cleared. For example, writing the value 0x0010020C would clear bits 2, 3, 9, and 20. To clear the short circuit detected bit, write a 1 to the register. To clear all bits, write 0xFFFFFFFF to the register.

LIMIT STATUS MASK   INDEX 0X2184

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	-	TR	RF

This parameter defines which bits in the Manufacturer Status Register object (index 0x1002, p. 62) can set the limit bit (bit 11) of the Status Word object (index 0x6041, p. 61). If a Manufacturer Status Register bit and its corresponding Limit Mask bit are both set, then the CANopen Status Word limit bit is set. If all selected a Manufacturer Status Register bits are clear, then the limit bit is clear.

Error Codes

ABORT OPTION CODE   INDEX 0x6007

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Values	0 to 3	TR	RF

Abort option code for CANopen / EtherCAT drives.

Value	Description
0	No action
1	Fault signal
2	Disable voltage command
3	Quick Stop command
4 to 2^{15}-1	Reserved

ERROR CODE   INDEX 0x603F

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RO	16	-		R

Provides the error code of the last error that occurred in the drive.
These are the supported error types for Copley drives:

0x2320	Short circuit		0x7380	Positive position limit
0x4210	Amp over temp		0x7381	Negative position limit
0x3110	Amp over voltage		0x7390	Tracking error
0x3120	Amp under voltage		0x73A0	Position wrap
0x4300	Motor over temp		0x5080	Used for faults with no other emergency
0x2280	Feedback error		0x8130	Node guarding error
0x7122	Phasing error		0x61FF	Command error
0x2310	Current limit		0x5440	STO fault
0x3310	Voltage limit			

QUICK STOP OPTION CODE   INDEX 0x605A

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Bits	See *Description*, below.	TR	RF

This object defines the behavior of the drive when a quick stop command is issued. The values are defined as follows:

Value	Description
0	Disable the drive's outputs
1	Slow down using the normal slow down ramp programmed in Profile Deceleration (index 0x6084, p. 157). When the move has been successfully aborted the drive's state will transition to the 'switch on disabled' state.

Value	Description
2	Slow down using the quick stop ramp programmed in Quick Stop Deceleration (index 0x6085, p. 158) then transition to 'switch on disabled'.
3	Stop the move abruptly and transition to 'switch on disabled'.
5	Slow down using the slow down ramp. The drive state will remain in the 'quick stop' state after the move has been finished.
6	Slow down using the quick stop ramp and stay in 'quick stop' state.
7	Stop the move abruptly and stay in 'quick stop' state.

All other values will produce unspecified results and should not be used.

SHUTDOWN OPTION CODE INDEX 0x605B

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	See *Description*, below.	TR	RF

This object defines the behavior of the drive when the drive's state is changed from "operation enabled" to "ready to switch on." The following values are defined:

Value	Description
0	Disable the drive's outputs.
1	Slow down using the slow down ramp (i.e. the normal move deceleration value), then disable outputs.

All other values will produce unspecified results and should not be used.

DISABLE OPERATION OPTION CODE   INDEX 0x605C

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	TR	RF

This object defines the behavior of the drive when the drive's state is changed from "operation enabled" to "switched on." The following values are defined.

Value	Description
0	Disable the drive's outputs.
1	Slow down using the slow down ramp (i.e. the normal move deceleration value), then disable outputs.

All other values will produce unspecified results and should not be used.

HALT OPTION CODE   INDEX 0x605D

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	TR	RF

This object defines the behavior of the drive when a halt command is issued. The values are defined as follows:

Value	Description
0	Disable the drive's outputs.
1	Slow down using the slow down ramp (i.e. the normal move deceleration value), then stay in Operation Enabled.
2	Slow down using the quick stop ramp, then stay in Operation Enabled.
3	Slow down on current limit, then stay in Operation Enabled.
4	Slow down on voltage limit, then stay in Operation Enabled.

All other values will produce unspecified results and should not be used.

FAULT REACTION OPTION CODE   INDEX 0x605E

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0	See *Description*, below.	TR	R

This object defines the behavior when a fault occurs, currently we only support the Disable Drive Function (value = 0).

MODE OF OPERATION   INDEX 0x6060

Type	Access	Units	Range	Map PDO	Memory
INTEGER8	RW	Integers	See *Description*, below.	TR	R

This object selects the drive's mode of operation. The modes of operation presently supported by this device are:

Mode	Description
1	Profile Position mode.
3	Profile Velocity mode.
4	Profile Torque mode.
6	Homing mode.
7	Interpolated Position mode.
8	CSP: Cyclic Synchronous Position mode.
9	CSV: Cyclic Synchronous Velocity mode.
10	CST: Cyclic Synchronous Torque mode.
11	CSTCA: Cyclic Synchronous Torque with Commutation Angle

The drive will not accept other values.

Note that there may be some delay between setting the mode of operation and the drive assuming that mode. To read the active mode of operation, use object 0x6061.

MODE OF OPERATION DISPLAY   INDEX 0x6061

Type	Access	Units	Range	Map PDO	Memory
INTEGER8	RO	Integers	See *Description*, below.	T	R

This object displays the current mode of operation.

DESIRED STATE   INDEX 0x2300

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 to 42	TR	R

This object defines what input source controls the drive, and what general mode the drive runs in. It is encoded as follows:

Code	Description
0	Disabled.
1	The current loop is driven by the programmed current value.
2	The current loop is driven by the analog command input.
3	The current loop is driven by the PWM & direction input pins.
4	The current loop is driven by the internal function generator.
5	The current loop is driven by UV commands via PWM inputs.
6	UV Current mode
7	Current command slaved to other axis
11	The velocity loop is driven by the programmed velocity value.
12	The velocity loop is driven by the analog command input.
13	The velocity loop is driven by the PWM & direction input pins.
14	The velocity loop is driven by the internal function generator.
17	Velocity command slaved to other axis
21	In servo mode, the position loop is driven by the trajectory generator.
22	In servo mode, the position loop is driven by the analog command input.
23	In servo mode, the position loop is driven by the digital inputs (pulse & direction, master encoder, etc.).
24	In servo mode, the position loop is driven by the internal function generator.
25	In servo mode, the position loop is driven by the camming function.
26	Analog reference commands velocity to position loop.
30	In servo mode, the position loop is driven by the CANopen interface.
31	In microstepping mode, the position loop is driven by the trajectory generator.
33	In microstepping mode, the position loop is driven by the digital inputs (pulse & direction, master encoder, etc.).
34	In microstepping mode, the position loop is driven by the internal function generator.
35	In microstepping mode, the position loop is driven by the camming function.
36	In microstepping mode, analog command references drive velocity.
37	In microstepping mode, position slaved to another axis.
40	In microstepping mode, the drive is driven by the CANopen interface.
42	Microstepping diagnostic mode. The current loop is driven by the programmed current value, and the phase angle is micro-stepped.
	Unlisted codes are reserved.

Note that this object should normally be programmed to 30 (or 40 for stepper motors) for use under the CANopen interface.

SUPPORTED DRIVE MODES   INDEX 0X6502

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below.	T	R

This bit-mapped value gives the modes of operation supported by the drive.

The standard device profile (DSP402) defines several modes of operation. Each mode is assigned one bit in this variable. A drive indicates its support for the mode of operation by setting the corresponding bit. The modes of operation supported by this device, and their corresponding bits in this object, are as follows:

Bit	Description
0	Position profile mode (pp).
2	Profile velocity mode (pv).
3	Profile torque mode (tq).
5	Homing mode (hm).
6	Interpolated position mode (ip).
7	Cyclic sync position mode (csp).
8	Cyclic sync velocity mode (csv).
9	Cyclic sync torque mode(cst).
10	Cyclic sync torque with commutation angle (CSTCA)

The current version of drive firmware supports only these five modes of operation and the corresponding bits are the only ones set in the object. Therefore the expected value of this object is 0x000003ED. Future versions of Copley Controls CANopen drive firmware might support additional operating modes. If so, those versions will return additional values.

5.3   Error Management Objects

ERROR REGISTER   INDEX 0x1001

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RO	-	See *Description*, below.	T	R

This object is a bit-mapped list of error conditions present in the drive. The bits used in this register are mapped as follows:

Bits	Description
0	Generic error. This bit is set any time there is an error condition in the drive.
1	Current error. Indicates either a short circuit on the motor outputs, or excessive current draw by the encoder.
2	Voltage error. The DC bus voltage supplied to the drive is either over or under the drive's limits.
3	Temperature error. Either the drive or motor is over temperature. Note that the drive will only detect a motor over temperature condition if an drive input has been configured to detect this condition.
4	Communication error. The drive does not presently use this bit.
5-6	Reserved.
7	The following errors cause this bit to be set; Motor phasing error, tracking error, limit switch active.

PRE-DEFINED ERROR OBJECT   INDEX 0x1003

Type	Access	Units	Range	Map PDO	Memory
Array	RW	-	-	NO	R

This object provides an error history. Each sub-index object holds an error that has occurred on the device and has been signaled via the Emergency Object. See Emergency Messages (p. 46). The entry at sub-index 0 contains the number of errors that are recorded in the array starting at sub-index 1. Each new error is stored at sub-index 1. Older errors move down the list.

NUMBER OF ERRORS   INDEX 0x1003, SUB-INDEX 0

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	-	0-8	NO	R

Number of errors in the error history (number of sub-index objects 1-8). Writing a 0 deletes the error history (empties the array). Writing a value higher than 0 results in an error.

STANDARD ERROR FIELD   INDEX 0x1003, SUB-INDEX 1-8

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	-	NO	R

One sub-index object for each error found, up to 8 errors. Each is composed of a 16-bit error code and a 16-bit additional error information field. The error code is contained in the lower 2 bytes (LSB) and the additional information is included in the upper 2 bytes (MSB).

TRACKING ERROR WINDOW   INDEX 0x2120

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Counts	0 to $2^{32}-1$	TR	RF

Also known as Position Tracking Error Limit. Specifies the maximum absolute Position Error (index 0x60F4, p. 144) allowed before a tracking error event is triggered. If the Position Error exceeds this value, then the tracking warning bit (bit 18) is set in the Manufacturer Status Register (index 0x1002, p. 62).Using the Fault Mask object (index 0x2182, p. 71), the tracking error event can be configured to either disable the drive immediately, or abort the present move and continue holding position.

FAULT MASK   INDEX 0x2182

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See *Description*, below.	TR	RF

This variable is used to configure which drive events cause latching faults. Setting a fault mask bit to 1 causes the associated drive event to cause a latching fault when it occurs. Setting a fault mask bit to 0 disables fault latching on the associated event. Latched faults may cleared using the

Latching Fault Status Register Object (index 0x2183, p. 72).

The fault mask is bit-mapped as follows:

Bits	Contents
0	Data flash CRC failure. This bit is read only and cannot be cleared. It indicates that the drive detected corrupted flash data values on startup. The drive will remain disabled and indicate a fault condition.
1	Drive internal error. This bit is read only and cannot be cleared. It indicates that the drive failed its power-on self-test. The drive will remain disabled and indicate a fault condition.
2	Short circuit. If set, then the drive will latch a fault condition when a short circuit is detected on the motor outputs. If clear, the drive will disable its outputs for 100 milliseconds and then re-enable.
3	Drive over temperature. If set, this bit will cause an drive over temperature condition to act as a latching fault. If clear, the drive will re-enable as soon as it cools sufficiently.
4	Motor over temperature. If set, an active input on a motor temperature sensor will cause the drive to latch a fault condition. If clear, the drive will re-enable as soon as the over temperature input becomes inactive.
5	Over voltage. Determines whether excessive bus voltage will cause a latching fault.
6	Under voltage. Determines whether inadequate bus voltage will cause a latching fault.
7	Feedback fault. Allows encoder power errors to cause latching faults. Feedback faults occur if: a digital encoder draws too much current from the 5-volt source on the drive; a resolver or analog encoder is disconnected; a resolver or analog encoder has levels out of tolerance. This is not available for all amps.
8	Phasing error. If set, phasing errors are latched. If clear, the drive is re-enabled when the phasing error is removed.
9	Tracking error. If set, a tracking error will cause the drive to latch in the disabled state. If clear, a tracking error will cause the present move to be aborted, but the drive will remain enabled.
10	Output current limited by I^2T algorithm.
11	FPGA failure. This bit is read only and cannot be cleared. It indicates that the drive detected an FPGA failure. The drive will remain disabled and indicate a fault condition.
12	Command input lost fault. If set: programs the drive to latch in the disabled state when the command input is lost. This fault is currently only available on special drives.

Bits	Contents
13-31	Reserved

LATCHING FAULT STATUS REGISTER   INDEX 0x2183

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Bits	Bit mapped	TR	R

Shows which latching faults have occurred in the drive. When a latching fault has occurred, the fault bit (bit 22) of the Manufacturer Status Register object (index 0x1002, p. 62) is set. The cause of the fault can be read from this register.

To clear a fault condition, write a 0xffffffff to the associated bit in this register.

The events that cause the drive to latch a fault are programmable. See Fault Mask object (index 0x2182, p. 71) for details.

Latched Faults

Bit	Fault Description
0	Short circuit detected.
1	Drive over temperature.
2	Over voltage.
3	Under voltage.
4	Motor temperature sensor active.
5	Encoder power error.
6	Motor phasing error.
7	Current output limited.
8	Voltage output limited.
9	Positive limit switch active.
10	Negavtive switch limit active.
11	Enable input not active
12	Drive disabled by software.
13	Trying to stop motor.
14	Motor brake activated.
15	PWM outputs disabled.
16	Positive software limit condition.
17	Negative software limit condition.
18	Following error fault.
19	Following error warning.
20	Drive is currently resetting.
21	Position has wrapped.
22	Drive fault.
23	Velocity limit has been reached.
24	Acceleration limit has been reached.
25	Position tracking.
26	Home switch is active.
27	In motion.
28	Velocity window.

Latched Faults

29	Phase not yet initialized.
30	Command fault.
31	Reserved.

STATUS OF SAFETY CIRCUIT (STO)   INDEX 0x219D

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Integers	0 to 8		R

This parameter allows the status of the safety circuit in Plus family drives to be queried.
For drives without a safety circuit, this parameter is reserved.

Bit	Description
0	Set when safety input 0 is preventing the drive from enabling.
1	Set when safety input 1 is preventing the drive from enabling.
8	This read/write bit can be used to force the 'drive is unsafe' output of the safety circuit to go active for testing purposes. Write 1 to force.

5.4   Basic Drive Configuration Objects

DEVICE TYPE   INDEX 0x1000

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below.	NO	R

Describes the type of device and its functionality.

This 32-bit value is composed of two 16-bit components. The lower two bytes identify the device profile supported by the device. This drive supports the DSP402 device profile, indicated by the value 0x0192.

The upper two bytes give detailed information about the type of motors the drive can control. The bit mapping of this value is defined by the *CANopen Profile for Drives and Motion Control (DSP 402)*. For Copley Controls CANopen drives, this value is 0x0006, indicating that Copley Controls supports servo and stepper devices.

DEVICE NAME   INDEX 0x1008

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RO	320	-	NO	R

An ASCII string which gives the drive's model number.

HARDWARE VERSION STRING   INDEX 0x1009

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RO	320	-	NO	R

Describes drive hardware version.

SOFTWARE VERSION NUMBER   INDEX 0x100A

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RO	320	-	NO	R

Contains an ASCII string listing the software version number of the drive.

BOOT LOADER SOFTWARE VERSION   INDEX 0x100B

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RO	320	-	NO	R

Contains an ASCII string listing the boot loader software version number of the drive.

SAVE PARAMETERS   INDEX 0x1010

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	144	-	NO	RF

Allows the current values programmed into the drive's objects to be saved to flash memory. The various sub-index values of this object allow either all objects, or specific groups of objects to be saved. Sub-index 0 contains the number of sub-elements of this record.

SAVE ALL OBJECTS   INDEX 0x1010, SUB-INDEX 1 OR STRING

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	-	NO	R

When read, this object will return the value 1 indicating that the device is able to save objects in this category. When the ASCII string "save" (or, the corresponding 32-bit value 0x65766173) is written to this object, all objects in the object dictionary that can be saved to flash are written. Objects written to flash will resume the stored value after an drive reset.

Note that not every object in the object dictionary may be written to flash. Presently, the objects that define the drive's CANopen communication interface are not stored to flash and will resume default values on startup. Most other objects may be stored to flash.

SAVE COMMUNICATION PARAMETERS   INDEX 0x1010, SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	-	NO	R

When read, this object returns the value 1, indicating that the device can save objects in this category. When the ASCII string "save" (or, the corresponding 32-bit value 0x65766173) is written to this object, all objects in the object dictionary that can be saved to flash are written. Objects written to flash resume the stored value after an drive reset.

Objects in the category are the objects with indexes in the range 0x1000 – 0x1FFF.

SAVE DEVICE PROFILE PARAMETERS INDEX 0x1010, SUB-INDEX 3

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	-	NO	R

When read, this object returns the value 1, indicating that the device can save objects in this category. When the ASCII string "save" (or, the corresponding 32-bit value 0x65766173) is written to this object, all objects in the object dictionary that can be saved to flash are written. Objects written to flash resume the stored value after an drive reset.

Objects in the category are the objects with indexes in the range 0x6000 – 0x9FFF.

SAVE MANUFACTURER SPECIFIC PARAMETERS   INDEX 0x1010, SUB-INDEX 4

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	-	NO	R

When read, this object returns the value 1 indicating that the device is able to save objects in this category. When the ASCII string "save" (or, the corresponding 32-bit value 0x65766173) is written to this object, all objects in the object dictionary that can be saved to flash are written. Objects written to flash resume the stored value after an drive reset. Objects in the category are the objects with indexes in the range 0x2000 – 0x5FFF.

IDENTITY OBJECT   INDEX 0X1018

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RO	144	-	NO	R

This object can uniquely identify an drive by unique manufacturer ID, serial number, and product revision information. Sub-index 0 contains the number of sub-elements of this record.

VENDOR ID   INDEX 0X1018, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	0x000000AB	NO	R

A unique identifier assigned to Copley Controls.

The value of this identifier is fixed at: 0x000000AB

PRODUCT CODE   INDEX 0X1018, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below.	NO	R

Identifies the specific drive model. Also known as Drive Hardware Type. (Index 0x2384, Sub-Index 13

The currently defined values for this object are:

Value	Product
0x0000	Accelus Card
0x0001	Accelus Panel without pull-up/down on inputs
0x0002	Accelus Panel with pull-up/down on inputs
0x0100	Junus Panel
0x0200	Accelnet Module
0x0201	Xenus Panel
0x0300	Accelnet Panel, custom dual axis
0x0310	XSJ Beta version
0x0320	XTL Resolver Prototype
0x0330	XTL Prototype
0x0340	XSJ Resolver
0x0350	STX
0x0360	ACJ-R
0x0370	ACK-R
0x0380	AEP
0x03A0	Accelnet Panel.
0x03B0	Stepnet, 3-axis.
0x03C0	Custom drive (800-1638)
0x1000	XEL Prototype.
0x1001	XEL (Xenus+ EtherCAT)
0x1008	XEL-R (Xenus+ EtherCAT Resolver)
0x1010	XML (Xenus+ MACRO)
0x1020	XPL (Xenus+ CAN)
0x1030	AEM (EtherCAT Module)
0x1040	APM (CAN Module)

Value	Product
0x1050	AE2 (2-axis EtherCAT Module)
0x1060	AP2 (2-axis CAN Module)
0x1070	SEM (Stepper EtherCAT Module)
0x1080	SPM (Stepper CAN Module)
0x1090	SE2 (2-axis Stepper EtherCAT Module)
0x10A0	SP2 (2-axis Stepper CAN Module)
0x10B0	XE2 (2-axis Xenus+ EtherCAT)
0x10C0	BE2 (2-axis Accelnet+ EtherCAT)
0x10D0	XP2 (2-axis Xenus+ CAN)
0x10E0	BP2 (2-axis Accelnet+ CAN)
0x10F0	TE2 (2-axis Stepnet+ EtherCAT)
0x1100	TP2 (2-axis Stepnet+ CAN)
0x1110	BEL (Accelnet+ EtherCAT)
0x1120	BPL (Accelnet+ CAN)
0x1130	TEL (Stepnet+ EtherCAT)
0x1140	TPL (Stepnet+ CAN)
0x1150	SP4 (4-axis Stepnet+ CAN)
0x1170	XM2 (2-axis Xenus+ MACRO)
0x1180	BML (Accelnet+ MACRO)
0x1190	SE4 (4-axis Stepnet+ EtherCAT)
0x11B0	XEC (Xenus+ Compact, EtherCAT)
0x11C0	XPC (Xenus+ Compact, CAN)

REVISION NUMBER   INDEX 0x1018, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	-	NO	R

Identifies the revision of the CANopen interface.

SERIAL NUMBER   INDEX 0x1018, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	-	NO	R

The drive's serial number. Holds the same value as Drive Serial Number (Index 0x2384, Sub-Index 1

DRIVE SCALING CONFIGURATION   INDEX 0x2080

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	Bits	Bit mapped	NO	F

This read-only parameter defines the units used for current and voltage readings from the drive:

Bits	Description	
0 -1	Identify units for current readings:	
	0	0.01 A
	1	0.001 A
	2	0.0001 A
	3	0.00001 A
2 - 7	Reserved	
8 - 9	Identify units for voltage readings:	
	0	0.1 V
	1	0.01 V
	2	0.001 V
	3	0.0001 V
10 - 31	Reserved	

DRIVE NAME   INDEX 0x21A0

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RW	320		NO	F

This object may be used to assign a name to an drive. The data written here is stored to flash memory and is not used by the drive. Although this object is documented as holding a string (i.e. ASCII data), any values may be written here. Up to 40 bytes are stored.

INTER-DRIVE COMMUNICATION CONFIGURATION   INDEX 0x21A4

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to 2^{32} -1	NO	RF

Inter-drive communication configuration. Bit-mapped as follows:

Bit	Description
0	Set for IDC Master. Clear for IDC slave devices
1	Set to disable serial command forwarding via IDC.
8-10	Address of partner axis for cross-coupling

INTER-DRIVE COMMUNICATION STATUS   INDEX 0x21A5

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to 2^{32} -1	NO	R

Inter-drive communication status. Object not available on 8367 based products. Bitmapped as follows:

Bit	Description
0	Synchronized to IDC bus if set.
1	Address assignment complete if set
2	IDC running normally if set.
8	Set if IDC is reset
16-18	Assigned IDC address

IP ADDRESS FOR UDP CONTROL   INDEX 0x21B5

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to 2^{32} -1	NO	RF

IP address. This is used for Ethernet enabled drives running in UDP/IP or TCP/IP modes. When the drive is configured to obtain it's own IP address using DHCP this parameter will return 0 until an IP address has been assigned at which point this parameter will return that address. The address assigned by the server is stored to flash and the drive will request the same address from the DHCP server the next time it powers up. IP addresses are normally displayed as a series of 4 byte with decimal points between them. An IP address of 1.2.3.4 would be stored to this parameter as an integer with the value 0x04030201.
Object not available on 8367-based products.

MISC. DRIVE OPTIONS REGISTER   INDEX 0x2420

Type	Access	Bits	Range	Map PDO	Memory
INTEGER32	RW	32	Bit mapped	TR	RF

Miscellaneous Drive Options Register. Bit-mapped as follows:

Bit	Description
0	If set, input pins 1, 2, and 3 are pulled high on the drive. If clear the pins are not pulled up. This option is only available on the Junus drive.
1	Reserved.
2	If set, limit switch inputs will only abort a trajectory in progress but will not affect current output. If clear, limit switches limit current.
3	If set, save PDO configuration to a file in the CVM file system when a "Save to Flash" command is received over the CANopen network. If clear, a PDO is not saved.
4	If set, a limit switch activation will be treated as a fault in the CANopen Status Word (CANopen index 0x6041).
5-6	When encoder wrap is enabled, these bits control the direction of motion for absolute moves in trapezoidal and S-curve profile modes. 0 – move in the shortest direction 1 – Always move in the positive direction 2 – Always move in the negative direction 3 – reserved

Bit	Description
7	If set, analog command values will use digital data written to an SPI serial interface connected to drive input pins & multimode port. This is available on some plus drives for use in digitally interfacing with a Delta Tau controller
8-31	Reserved

FLASH PROGRAM UPDATER 0X2001

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	-	NO	R

For updating firmware on products that are not Plus family products. CME 2 is the preferred method for this. Contact Copley for more information if your application requires this.

NETWORK NODE ID CONFIGURATION   INDEX 0X21B0

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	NO	RF

This object is used to configure the CANopen network bit rate and node ID for the drive.

The bit rate is read only at power-up or reset.

Likewise, the ID is calculated at power-up or reset (and only then) using a combination of general-purpose input pins and a programmed offset value. On certain models, an address switch is also used. The resulting value is clipped to a 7-bit ID in the range 0 to 127.

The configuration parameter is bit-mapped as follows. Values written here are stored to flash memory. The new network configuration will not take effect until the drive is reset.

Bit	Description
0-6	Give the node ID offset value.
7	Used only on DeviceNet firmware. If this bit is set, then the drive will be software disabled on startup and will remain disabled until it is enabled by a DeviceNet I/O message with the enable bit set.
8-10	Number of input pins (0-7) to read on startup for the node ID value. If input pins are used (i.e., the value in bits 8-10 is not zero), the inputs can be mapped to node ID bits through the object Input Mapping for Network Node ID (index 0x21B1, p. 81).
11	This bit is ignored on drives that do not have an address switch. On drives with an address switch, setting this bit programs the drive to use the address selector switch as part of the address calculation. In this case, the node ID value is equal to the sum of: The value read from the designated input pins, shifted up 4 bits. The address switch value. The programmed offset value. Note that since the node ID is always clipped to the lowest 7 bits, no more than 3 input pins will ever have an effect on the node address when the address switch is used.
12-15	Network bit rate setting.

The network bit rate is encoded as one of the following values:

Code	Bit Rate (bits / second)
0	1,000,000
1	800,000
2	500,000
3	250,000

Code	Bit Rate (bits / second)
4	125,000
5	50,000
6	20,000
7-15	Reserved for future use

INPUT MAPPING FOR NETWORK NODE ID   INDEX 0X21B1

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	See *Description*, below.	NO	F

When the Network Node ID Configuration object (index 0x21B0, p. 80) indicates that 1 or more input pins will be used to select the CAN node ID, this mapping register is used to select which input pins will be mapped to which ID bit. Fields include:

Bit	Description
0-3	Identify the general-purpose input pin associated with ID bit 0.
4-7	Identify the general-purpose input pin associated with ID bit 1.
8-11	Identify the general-purpose input pin associated with ID bit 2.
12-15	Identify the general-purpose input pin associated with ID bit 3.
16-19	Identify the general-purpose input pin associated with ID bit 4.
20-23	Identify the general-purpose input pin associated with ID bit 5.
24-27	Identify the general-purpose input pin associated with ID bit 6.
28-30	Reserved for future use.
31	Set to enable this register. Clear to use default mapping.

If bit 31 is zero, then a default bit mapping is used and the rest of this register is ignored. The default bit mapping uses the top N input pins and maps them such that the high-numbered pins are used for higher-numbered bits in the ID. For example, the Accelnet panel drive has 12 general purpose input pins (0 to 11). If 3 of these pins are used for ID configuration and the default mapping is used, then the highest 3 pins (9, 10 and 11) will be used for the ID. In this case, pin 9 is bit 0, pin 10 is bit 1 and pin 11 is bit 2. If bit 31 is set, then the rest of this register is used to define which input pin is assigned to which bit of the ID. The input pins are numbered from 0 to 15 and each nibble of the register gives the input pin number associated with one bit of the ID.

For example, if three input pins are configured for address selection and the mapping register is set to 0x80000012, then input pin 2 is used for ID bit 0, input pin 1 is used for ID bit 1, and input pin 0 is used for ID bit 2.

Note that the CAN node ID is calculated at startup only. The input pins assigned to the node ID are sampled once during power up and used to calculate the ID. These pins may be assigned other uses after power up if necessary.

CURRENT STATE OF THE CAN ID SELECTION SWITCH INDEX 0x2197

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	-	0 - 15	YES	R

This object gives the current value of the CAN address switch.
For drives that do not have a switch, the value returned is undefined.

BUFFERED ENCODER OUTPUT CONFIGURATION (MULTI-PORT)   INDEX 0x2241

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-	TR	RF

Multi-mode Port Configuration. The available settings are:

Value	Description
0	Output buffered primary encoder (hardware buffering).
1	Configure pins as inputs.
2	Output simulated encoder outputs tracking motor encoder.
3	Output simulated encoder outputs tracking position encoder.

DRIVE MODEL NUMBER   INDEX 0x6503

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RO	320	-	NO	R

This ASCII string gives the drive model number.

DRIVE MANUFACTURER   INDEX 0x6504

Type	Access	Units	Range	Map PDO	Memory
VISIBLE_STRING	RO	-	-	NO	R

This ASCII string identifies the drive's manufacturer as "Copley Controls ".

MANUFACTURER'S WEB ADDRESS   INDEX 0x6505

Type	Access	Units	Range	Map PDO	Memory
VISIBLE_STRING	RO	-	-	NO	R

This ASCII string gives the web address of Copley Controls.

SERVO LOOP CONFIG   INDEX 0x2301

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Bits	Bit mapped	TR	RF

This parameter allows various parts of the drive servo loops to be enabled/disabled. It's mapped as follows:

Bit	Description
0	If set, this disables the velocity loop gains. The velocity loop command feed forward gain (parameter 0x157) is still active as are the velocity loop output filters.
1	If set, this enables the position loop I and D gains (parameters 0x155 and 0x156). If clear, these parameters are treated as zeros.
2	If set, the velocity error window is calculated using a filtered version of the motor velocity. When cleared, the unflitered velocity is used.
3	If set, the velocity loop is used to stop the motor when the drive is disabed. If cleared the position loop is used in position mode and velocity loop is used in velocity mode.
4	If set, the analog reference input is used to add a current offset. Parameter 0x19 is used to scale the current in the same way it is used when running in mode 2.

WATCHDOG TIMEOUT FOR COMMAND INPUT   INDEX 0x2302

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Bits	0 to 2^{16} -1	NO	RF

Watchdog timeout (in ms). If non-zero, then an error will occur if a serial port command hasn't been received within this much time. When such an error occurs, the drive will be disabled. This parameter is supported on plus drives starting with version 4.18 firmware. Object not available on 8367 based products.

DRIVE DATA INDEX 0x2384

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	848	-	NO	F

This record lists various drive parameters. Sub-index 0 contains the number of sub-elements of this record.

DRIVE SERIAL NUMBER   INDEX 0x2384, SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RO	32	0 to 2^{32}-1	NO	F

Gives the drive serial number.

DRIVE MANUFACTURING INFO   INDEX 0x2384, SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RO	320		NO	F

Date of manufacture of the drive.

DRIVE PEAK CURRENT LIMIT INDEX 0x2384, SUB-INDEX 3

Type	Access	Units	Range (Adc)	Map PDO	Memory
INTEGER16	RO	0.01 amps	(0 to 2^{16}-1) / 100	NO	F

The drive's peak current rating.

DRIVE CONTINUOUS CURRENT   INDEX 0x2384, SUB-INDEX 4

Type	Access	Units	Range (Adc)	Map PDO	Memory
INTEGER16	RO	0.01 amps	(0 to 2^{16}-1) / 100	NO	F

The drive's continuous current rating.

DRIVE PEAK CURRENT TIME   INDEX 0x2384, SUB-INDEX 5

Type	Access	Units	Range (ms)	Map PDO	Memory
INTEGER16	RO	milliseconds	0 to 2^{16}-1	NO	F

The maximum time for which the drive is rated to output peak current.

DRIVE MAXIMUM VOLTAGE   INDEX 0x2384, SUB-INDEX 6

Type	Access	Units	Range (Vdc)	Map PDO	Memory
INTEGER16	RO	0.1 volts	(0 to 2^{16}-1) / 10	NO	F

Maximum bus voltage rating for drive.

DRIVE MINIMUM VOLTAGE   INDEX 0x2384, SUB-INDEX 7

Type	Access	Units	Range (Vdc)	Map PDO	Memory
INTEGER16	RO	0.1 volts	(0 to 2^{16}-1) / 10	NO	F

Minimum bus voltage rating for drive.

DRIVE VOLTAGE HYSTERESIS   INDEX 0x2384, SUB-INDEX 8

Type	Access	Units	Range (Vdc)	Map PDO	Memory
INTEGER16	RO	0.1 volts	(0 to 2^{16}-1) / 10	NO	F

Hysteresis for maximum bus voltage cut-out.

DRIVE MAXIMUM TEMPERATURE   INDEX 0x2384, SUB-INDEX 9

Type	Access	Units	Range (deg C)	Map PDO	Memory
INTEGER16	RO	degrees centigrade	0 to 2^{16}-1	NO	F

Temperature limit for drive.

DRIVE TEMPERATURE HYSTERESIS   INDEX 0x2384, SUB-INDEX 10

Type	Access	Units	Range (deg C)	Map PDO	Memory
INTEGER16	RO	degrees centigrade	0 to 2^{16}-1	NO	F

Hysteresis value for drive over temperature cut-out.

DRIVE CURRENT LOOP PERIOD   INDEX 0x2384, SUB-INDEX 11

Type	Access	Units	Range (ns)	Map PDO	Memory
UNSIGNED16	RW	10 nanoseconds	(0 to 2^{16}-1) x 10	NO	F

Current loop update period in 10-nanosecond units.

DRIVE SERVO LOOP PERIOD   INDEX 0x2384, SUB-INDEX 12

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integer	0 to 2^{16}-1	NO	F

Servo loop update period as a multiple of the current loop period. Default value is 4.

DRIVE TYPE CODE   INDEX 0x2384, SUB-INDEX 13

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	-	See *Description*, below.	NO	F

Identifies the specific drive model. Also known as Drive Hardware Type. Identical to Product Code (index 0x1018, Sub-index 2, p. 76). Go to page 76 (index 0x1018, Sub-index 2) for a table of current defined values.

CURRENT CORRESPONDING TO MAX A/D READING   INDEX 0x2384, SUB-INDEX 14

Type	Access	Units	Range (Adc)	Map PDO	Memory
INTEGER16	RO	0.01 amps	(0 to 2^{16}-1) / 100	NO	F

Drive current corresponding to maximum A/D reading.

VOLTAGE CORRESPONDING TO MAX A/D READING   INDEX 0x2384, SUB-INDEX 15

Type	Access	Units	Range (Vdc)	Map PDO	Memory
INTEGER16	RO	0.1 volts	(0 to 2^{16}-1) / 10	NO	F

Drive voltage corresponding to maximum A/D reading.

ANALOG INPUT SCALING FACTOR   INDEX 0X2384, SUB-INDEX 16

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	-	-	NO	F

Drive analog input scaling factor.

DRIVE MINIMUM PWM OFF TIME   INDEX 0X2384, SUB-INDEX 17

Type	Access	Units	Range (ns)	Map PDO	Memory
UNSIGNED16	RO	10 ns	(0 to 2^{16}-1) x 10	NO	F

This fixed drive parameter gives the minimum amount of time for which all PWM outputs must be disabled for each current loop cycle.

PWM DEAD TIME AT CONTINUOUS CURRENT LIMIT INDEX 0X2384, SUB-INDEX 18

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	CPU cycles	0 to 2^{16}-1	NO	F

This fixed drive parameter gives the PWM dead time used at or above the continuous current limit. The dead time below the continuous current limit is a linear function of this parameter and PWM Dead Time At Zero Current (Index 0x2384, Sub-Index 19, p. 85).

PWM DEAD TIME AT ZERO CURRENT   INDEX 0X2384, SUB-INDEX 19

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	CPU cycles	0 to 2^{16}-1	NO	F

This fixed drive parameter gives the PWM dead time used at or above the continuous current limit. The dead time below the continuous current limit is a linear function of this parameter and PWM Dead Time At Continuous Current Limit (Index 0x2384, Sub-Index 18 p. 85).

PEAK CURRENT INTERNAL REGEN RESISTOR   INDEX 0X2384, SUB-INDEX 20

Type	Access	Units	Range (Adc)	Map PDO	Memory
INTEGER16	RO	0.01 amps	(0 to 2^{16}-1) / 100	NO	F

The drive's peak current rating for its internal regen resistor.

CONTINUOUS CURRENT INTERNAL REGEN RESISTOR   INDEX 0X2384, SUB-INDEX 21

Type	Access	Units	Range (Adc)	Map PDO	Memory
INTEGER16	RO	0.01 amps	(0 to 2^{16}-1) / 100	NO	F

The drive's continuous current rating for its internal regen resistor.

TIME AT PEAK CURRENT INTERNAL REGEN RESISTOR   INDEX 0X2384, SUB-INDEX 22

Type	Access	Units	Range (ms)	Map PDO	Memory
UNSIGNED16	RO	ms	0 to 2^{16}-1	NO	F

The drive's maximum time at peak current rating for its internal regen resistor.

ANALOG ENCODER SCALING FACTOR   INDEX 0X2384, SUB-INDEX 23

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	-	-	NO	F

This parameter selects the resolution of an analog encoder input. The parameter is not used for other encoder types.

FIRMWARE VERSION NUMBER   INDEX 0X2384, SUB-INDEX 24

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	Bytes		NO	R

The version number consists of a major version number and a minor version number. The minor number is passed in bits 0-7; the major number is in bits 8-15. For example, the version 1.12 would be encoded 0x010C.

AXIS COUNT   INDEX 0X2384, SUB-INDEX 25

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	Integers	1 to 4	NO	R

Returns the number of axis implemented by this drive.

INTERNAL REGEN CURRENT LIMIT   INDEX 0X2384, SUB-INDEX 26

Type	Access	Units	Range (Adc)	Map PDO	Memory
INTEGER16	RO	mA	$(0$ to 2^{15}-1$) / 1000$	NO	F

Drive internal maximum regen current.

FPGA IMAGE VERSION NUMBER   INDEX 0X2384, SUB-INDEX 27

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	Integer	0 to 2^{15}-1	NO	R

FPGA firmware version number (available on certain drive models).

NIOS PROCESSOR FIRMWARE VERSION   INDEX 0X2384, SUB-INDEX 28

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	Integer	-	NO	R

Firmware version of second processor for drives equipped with two processors.

MISC. HARDWARE OPTIONS   INDEX 0X2384, SUB-INDEX 29

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	Integer	0 to 2^{32}-1	NO	F

Firmware version of second processor for drives equipped with two processors.

CURRENT LEVEL FOR MINIMUM PWM DEADTIME   INDEX 0X2384, SUB-INDEX 30

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	-	NO	F

Firmware version of second processor for drives equipped with two processors.

DRIVE DATA INDEX 0X6510

Type	Access	Units	Range	Map PDO	Memory
RECORD	RO	-	-	NO	R

The contents of this object are the same as those of 0x2384 (Index 0x2384).

This object is provided for compatibility with CANopen applications. Refer to 0x2384 for details.

FIRMWARE VERSION NUMBER (EXTENDED)   INDEX 0X2422

Type	Access	Units	Range	Map PDO	Memory
UNSIGNIED32	RO	Words	Upper, lower words	NO	R

Firmware Version Number (extended). The upper 16 bits give the same major/minor version number as Firmware Version Number (Index 0x2384, Sub-Index 24

The lower 16 bits hold a release number (upper byte) and a reserved byte (lower).

DEVICE TYPE   INDEX 0X67FF

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	-	NO	R

Holds the same data as object 0x1000. Repeated as required by the CANopen specification.

PWM MODE   INDEX 0X2140

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Bits	See *Description*, below.	TR	RF

PWM mode and status. This bit-mapped register allows some details of the PWM output to be controlled and monitored. Fields are described below:

Bit	Description
0	Force bus clamping if set, disable bus clamping if clear. Note that if bit 1 is set, then this bit is ignored.
1	Automatic bus clamping mode if set. Setting this bit causes bus clamping mode to be automatically selected based on the output voltage. Bit 0 is ignored if this bit is set.
3	Factory reserved. If set, DBrk mode is enabled.
4	Use hex voltage limiting if set, circular limiting if clear. This setting is only used with brushless motors.
6	Double PWM frequency if set.
8	Status bit, set when bus clamping is active.

RUNNING SUM OF USER CURRENT LIMIT   INDEX 0X2116

Type	Access	Units	Range (Percent)	Map PDO	Memory
INTEGER16	RO	0.01%	$(0 \text{ to } 2^{15}\text{-}1) / 100$	T	R

Running sum of user current limit.

RUNNING SUM OF AMP CURRENT LIMIT INDEX 0X2117

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01%	$(0 \text{ to } 2^{15}\text{-}1) / 100$	T	R

Running sum of the amp current limit.

D/A CONVERTER CONFIGURATION   INDEX 0x21E0

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Bits	Bit mapped	NO	RF

This parameter sets the mode for the D/A converter on drives so equipped.
The bits are mapped as follows:

Bit	Description
0-3	Define the mode of the D/A converter
16-17	Identify the axis associated with the D/A converter
Mode	**Description**
0	Manual configuration (set using parameter 0x135)
1	Actual current of configured axis

D/A CONVERTER OUTPUT VALUE   INDEX 0x21E1

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	mV	-2^{15} to $+2^{15}-1$	TR	R

For drives that support an auxiliary D/A converter, this parameter sets the output value in mV units when the D/A is in manual mode. In other modes, the current value being output on the D/A can be read here.

FAN TURN ON TEMPERATURE   INDEX 0x21E2

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	degrees centigrade	-2^{15} to $+2^{15}-1$	NO	RF

For products with an internal fan which can be software controlled, this parameter gives the temperature where the fan will first turn on.

FAN MAX SPEED TEMPERATURE   INDEX 0x21E3

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	degrees centigrade	-2^{15} to $+2^{15}-1$	NO	RF

Fan max speed temperature. For products with a software controlled internal fan, this gives the temperature at which the fan will run at top speed. Must be >= the fan turn on temperature or it will be ignored. Object not available on 8367 based products.

ANALOG ENCODER COSINE PHASE ADJUSTMENT INDEX 0x21E4

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.1 degrees	-2^{15} to $+2^{15}-1$	NO	RF

Encoder cosine angular offset. This parameter gives the angular error (in 0.1 deg units) of the encoder cosine signal. It's used to compensate for imperfections in the encoder signals. This adjustment is only used if the encoder cosine scaling value (object 0x220D) is non-zero. Object not available on 8367 based products.

5.5   Basic Motor Configuration Objects

MOTOR TYPE   **INDEX 0x6402**

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW		See *Description*, below.	NO	R

Defines the type of motor connected to the drive:

Value	Description
0x8000	Not specified.
0x8010	Brushed servo motor.
0x8020	Microstepper
0x8030	Brushless servo motor.

MOTOR MODEL NUMBER   **INDEX 0x6403**

Type	Access	Units	Range	Map PDO	Memory
STRING[40]	RW	ASCII	See *Description*, below.	NO	F

The motor's model number.

MOTOR MANUFACTURER   **INDEX 0x6404**

Type	Access	Units	Range	Map PDO	Memory
STRING[40]	RW	ASCII	See *Description*, below.	NO	F

The motor's manufacturer name.

MOTOR DATA   **INDEX 0x2383**

Type	Access	Units	Range	Map PDO	Memory
RECORD	RW	-	-	NO	F

This record holds a variety of motor parameters. Note that all motor parameters are stored to non-volatile memory on the drive. The programmed values are preserved across power cycles. Sub-index 0 contains the number of sub-elements of this record.

MOTOR TYPE   INDEX 0x2383, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below.	NO	F

Defines the type of motor connected to the drive:

Type	Description
0	Rotary motor.
1	Linear motor.

MOTOR POLE PAIRS   INDEX 0X2383, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	0 to $2^{15}-1$	NO	F

Number of motor pole pairs (electrical cycles) per rotation. For example, a 1.8 deg/step motor would require setting motor poll pairs to 50. This parameter is only used for rotary motors. For linear motors its value is ignored.

MOTOR WIRING CONFIGURATION   INDEX 0X2383, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	See *Description*, below.	NO	F

Defines the direction of the motor wiring:

Type	Description
0	Standard wiring.
1	Motor's U and V wires are swapped.

MOTOR HALL TYPE   INDEX 0X2383, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	See *Description*, below.	NO	F

Defines the type of Hall Effect sensors attached to the motor:

Type	Description
0	No Hall sensors available.
1	Digital Hall sensors.
2	Analog Hall sensors.

MOTOR HALL WIRING   INDEX 0X2383, SUB-INDEX 5

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	See *Description*, below.	NO	F

Defines the wiring of the Hall sensors. Bit-mapped as follows (when analog Halls are used, only bit 8 is relevant):

Bits	Description
0-2	The Hall wiring code (see below).
3	Reserved.
4	Invert W Hall input if set. Inversion occurs after Halls wiring has been modified by bits 0-2.
5	Invert V Hall input if set. Inversion occurs after Halls wiring has been modified by bits 0-2.
6	Invert U Hall input if set. Inversion occurs after Halls wiring has been modified by bits 0-2.
7	Reserved.
8	Swap analog Halls if set.
9-15	Reserved.

The Hall wiring codes define the order of the Hall connections:

Code	Hall ordering
0	U V W
1	U W V
2	V U W
3	V W U
4	W V U
5	W U V
6	Reserved
7	Reserved

MOTOR HALL OFFSET INDEX 0X2383, SUB-INDEX 6

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	degrees	-360 to +360	NO	RF

Offset angle to be applied to the Hall sensors.

MOTOR RESISTANCE   INDEX 0X2383, SUB-INDEX 7

Type	Access	Units	Range (ohm)	Map PDO	Memory
INTEGER16	RW	0.01 Ohm	0 to 327.67	NO	F

Motor winding resistance, in 0.01-Ohm units.

MOTOR INDUCTANCE   INDEX 0X2383, SUB-INDEX 8

Type	Access	Units	Range (mH)	Map PDO	Memory
INTEGER16	RW	0.01 millihenry	0 – 327.67	NO	F

Motor winding inductance, in 0.01 milliHenry units.

MOTOR INERTIA   INDEX 0X2383, SUB-INDEX 9

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Rotary: 0.000001 Kg / cm^2 Linear: 0.0001 Kg.	Rotary: 0 to 4295 Kg / cm^2 Linear: 0 to 429 x 10^3 kg	NO	F

Motor inertia.

MOTOR BACK EMF CONSTANT   INDEX 0X2383, SUB-INDEX 10

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Rotary: 0.01 V/KRPM Linear: 0.01 V/m/s	Rotary: 0 to 21.47 x 10^6 V/KRPM Linear: 0 to 21.47 x 10^6 V/m/s	NO	F

Motor back-EMF constant.

MOTOR MAXIMUM VELOCITY INDEX 0X2383, SUB-INDEX 11

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts / second	0 to 215 x 10^6 count/sec	NO	F

Maximum motor velocity.

MOTOR TORQUE CONSTANT   INDEX 0X2383, SUB-INDEX 12

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Rotary: 0.001 Nm / Amp Linear: 0.00001 N.	Rotary: 0 to 2.15×10^6 Nm/Amp Linear: 0 to 21.5×10^3 V/m/s	NO	F

Motor Torque (Force) constant.

MOTOR PEAK TORQUE   INDEX 0X2383, SUB-INDEX 13

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Rotary: 0.001 Nm Linear: 0.00001 N	Rotary: 0 to 2.15×10^6 Nm Linear: 0 to 21.5×10^3 N	NO	F

Motor Peak Torque (Force).

MOTOR CONTINUOUS TORQUE   INDEX 0X2383, SUB-INDEX 14

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Rotary: 0.001 Nm/Amp Linear: 0.00001 N/Amp	Rotary: 0 to 2.15×10^6 Nm/A Linear: 0 to 21.5×10^3 Nm/A	NO	F

Motor Continuous Torque (Force).

MOTOR HAS TEMPERATURE SENSOR INDEX 0X2383, SUB-INDEX 15

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integer	See *Description*, below.	NO	F

Value	Description
0	No temperature sensor available.
1	Temperature sensor is available.

MOTOR HAS BRAKE   INDEX 0X2383, SUB-INDEX 16

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integer	See *Description*, below.	NO	F

Value	Description
0	The motor has a brake.
1	The motor does not have a brake.

DELAY FROM ERROR TO BRAKE ACTIVE   INDEX 0X2383, SUB-INDEX 17

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 - 10,000	NO	F

Also known as Brake/Stop Delay Time. When the drive is disabled, it will actively decelerate the motor for this amount of time (in milliseconds) before activating the brake output.

This delay may be cut short if the motor velocity falls below the value programmed in Motor Brake Velocity (Index 0x2383, Sub-Index 19, p. 93).

MOTOR BRAKE DELAY   INDEX 0X2383, SUB-INDEX 18

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 - 10,000	NO	F

After the brake output is activated, the drive will stay enabled for this amount of time to allow the brake to engage.

MOTOR BRAKE VELOCITY   INDEX 0X2383, SUB-INDEX 19

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts / second	0 to 429 x 10^6 count/sec	NO	F

During the **Delay from Error to Brake Active** (Index 0x2383, Sub-Index 17, p. 92), if the motor's actual velocity falls below this value the brake output is activated immediately.

MOTOR ENCODER TYPE   INDEX 0X2383, SUB-INDEX 20

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below.	NO	F

Also known as Motor Encoder Type. Identifies the type of encoder attached to the motor:

Value	Description
0	Primary (differential) quad encoder
1	No encoder (use motor back EMF for velocity estimation)
2	Analog encoder sine cosine
3	Secondary quad encoder from input lines (multimode port)
4	Low frequency analog encoder
5	Resolver
6	Digital Halls for position and velocity estimates
7	Analog encoder, updated at current loop rate.
8	Custom Y Encoder
9	Panasonic
10	SPI Command
11	EnDat
12	SSI
13	BiSS
14	Serial Encoders from Sanyo Denki, Tamagawa, Panasonic, and HD Systems
15	Custom encoders from HD Systems
16	Simple analog potentiometer feedback
17	Gurley virtual absolute encoder
18	Custom K Encoder
19	S2 Custom Encoder
20	Hiperface
21	Wire saving incremental encoder which outputs hall signals on encoder lines at power-up
22	Sankyo absolute encoder
23	Custom M encoder HG absolute
24	Digital inputs used as tertiary encoder inputs. Inputs configured as single ended or differential by using Digital Input Command Configuration. Not used in Desired State modes 3, 13, and 23 (PWM or Digital Input Command Modes)

25	Tachometer input
26	Tamagawa TS5643 absolute encoder
27	Hiperface DSL (using external adapter board)

ENCODER UNITS   INDEX 0X2383, SUB-INDEX 21

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	NO	F

This value defines the units used to describe linear motor encoders. It is not used with rotary motors.

Value	Description
0	micrometers
1	nanometers
2	millimeters

MOTOR ENCODER DIRECTION   INDEX 0X2383, SUB-INDEX 22

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below.	NO	F

Motor encoder direction. Value 0 for standard, value 1 to reverse direction.

MOTOR ENCODER COUNTS/REV   INDEX 0X2383, SUB-INDEX 23

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	counts / rev	0 to $2^{32}-1$	NO	F

For rotary motors gives the number of counts/motor revolution. When a resolver is used as the motor feedback device, this parameter sets the resolution of the interpolated position.
This parameter is not used for linear motors.

MOTOR ENCODER RESOLUTION   INDEX 0X2383, SUB-INDEX 24

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	encoder units / count	0 to $2^{15}-1$	NO	F

Number of Encoder Units (sub-index 21) / count. Only used with linear motors.

MOTOR ELECTRICAL DISTANCE   INDEX 0X2383, SUB-INDEX 25

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	encoder units / cycle	0 to $2^{31}-1$	NO	F

Number of Encoder Units (sub-index 21) / motor electrical cycle. Only used with linear motors.

ENCODER INDEX PULSE DISTANCE   INDEX 0X2383, SUB-INDEX 26

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	-	-	0 to $2^{31}-1$	NO	F

Reserved.

MOTOR UNITS   INDEX 0x2383, SUB-INDEX 27

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	-	-	-	NO	F

Reserved: Only used by CME 2 for display. 0 = metric, 1 = English.

ANALOG ENCODER SHIFT   INDEX 0x2383, SUB-INDEX 28

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 - 8	NO	F

This value gives the number of bits of interpolation to be applied to an analog encoder.
The fundamental encoder resolution will be increased by a multiplier of 2^n where n is the value programmed in this parameter.

The range of this value is 0 to 8 giving possible multipliers of 1 to 256.

MICROSTEPS/REV   INDEX 0x2383, SUB-INDEX 29

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	microsteps	0 to $2^{32}-1$	NO	F

Microsteps per revolution for microstepping motors.

LOAD ENCODER TYPE INDEX 0x2383, SUB-INDEX 30

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below.	NO	F

Also known as Position Encoder Type. This bit-mapped value defines the type of encoder attached to the load:

Bits	Description
0-2	Encoder Type (see below).
3	Reserved.
4	Linear encoder if set, rotary encoder if clear.
5	Passive load encoder if set.

The encoder type codes define the type of encoder.

Code	Encoder Type
0	No load encoder present.
1	Primary incremental quadrature encoder.
2	Analog encoder.
3	Multi-mode port incremental quadrature encoder.
4	Low frequency analog encoder
5	Resolver.

LOAD ENCODER DIRECTION   INDEX 0x2383, SUB-INDEX 31

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	NO	F

Also known as Position Encoder Direction. Load encoder direction. Value 0 for standard, value 1 to reverse direction.

LOAD ENCODER RESOLUTION   INDEX 0x2383, SUB-INDEX 32

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	encoder units / count	0 to $2^{32}-1$	NO	F

Only used with linear motors. Also known as Position Encoder Resolution. Number of *Encoder Units* / encoder count. For information, see Encoder Units (Index 0x2383, Sub-Index 21, p. 94).

MOTOR GEAR RATIO   INDEX 0x2383, SUB-INDEX 33

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	F

Reserved.

NUMBER OF RESOLVER CYCLES/MOTOR REV   INDEX 0x2383, SUB-INDEX 34

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 to $2^{16}-1$	NO	F

Number of Resolver Cycles/Motor Rev. This parameter is only used with resolver feedback devices.

MOTOR DATA   INDEX 0x6410

Type	Access	Units	Range	Map PDO	Memory
RECORD	RW	-	-	NO	F

The contents of this object are the same as those of 0x2383 (p.89). This object is provided for compatibility with CANopen applications. Refer to 0x2383 for details.

MOTOR BRAKE ENABLE DELAY TIME   INDEX 0x2199

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	milliseconds	0 to 2^{15}-1	NO	F

This parameter gives a delay between enabling the drive PWM outputs and releasing the brake. Positive values mean the PWM is enabled first and then the brake is released N milliseconds later. Negative values cause the brake to be released before PWM outputs are enabled.

MOTOR ENCODER WRAP   INDEX 0x2220

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Counts	-2^{15} to $+2^{15}$-1	NO	RF

Actual motor position will wrap back to zero when this value is reached. Setting this value to zero disables this feature.

LOAD ENCODER WRAP   INDEX 0x2221

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Counts	-2^{15} to $+2^{15}$-1	NO	RF

Actual load position will wrap back to zero when this value is reached. Setting this value to zero disables this feature.

MOTOR ENCODER OPTIONS   INDEX 0x2222

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	-	NO	F

Specifies various configuration options for the motor encoder. The mapping of option bits to function depends on the encoder type.

Quadrature Encoder	
Bits	**Description**
0	If set, ignore differential signal errors (if detected in hardware)
1	If set, select single ended encoder inputs (if available in hardware)
2	If set, ignore differential signal errors on index input only (if supported by hardware)
3	If set, don't use index input at all. Useful when index input is being used by a different encoder interface
4	Reserved
5	If set, allows initialization of encoder type or options without resetting the position to 0. Normally the position would be set to 0.
Resolver (encoder type 5):	
Bit	**Description**
16	Set for NSK custom incremental resolver
17	Set for NSK custom absolute resolvers

18	Set for NSK custom resolvers on normal brushless motors. Clear for normal resolvers, or NSK resolvers on custom NSK

EnDat Encoder (Type 11)

Bits	Description
0-5	Number of bits of single turn data available from encoder
8-12	Number of bits of multi-turn data available from encoder
16	If set, analog inputs are supplied by encoder
17	If set, use multi-mode port
18	If set, read position using EnDat 2.2 style commands rather than default 2.1 style
19	If set, read encoder at current loop update rate. Otherwise, encoder is read at servo loop period.
20-23	Number of least significant bits of encoder reading to discard

SSI Encoder (Type 12)

Bits	Description
0-5	Number of bits of position data available
8-11	Number of extra bits sent with position data
12	*If set, ignore first bit of data sent by encoder
13	If set, encoder outputs position data using Gray code
14	*If set, pull clock low briefly after data (custom for Codechamp encoder)
15	If set, data is sent least signification bit first.
16-21	Encoder Bit Rate. If set, use 100 kHz units. If zero, use default 1 MHz units.
22	*If set, use setting of Motor Encoder Counts/Rev (0x62) to determine how many data bits to use
23	If set, extra status bits are before position data. If clear, extra status bits are after position data. Default is clear.
24	If set, first bit sent is 'data valid' bit
25	If set, use multi-mode port for SSI interface
26	If set, extra bits after position data are treated as fault bits and generate an encoder fault if any are set.

* NOTE – these three bits are depreciated and will be removed in future firmware versions

BiSS (Type 13)

Bits	Description
0-5	Number of bits of single turn data
8-12	Number of bits of multiturn data
15	If set, assume encoder position data wraps after number of encoder counts programmed in Motor Encoder Counts/Rev (0x62)
16	Set for modeC encoder format
17	Set to sample at servo loop rate (default at current loop rate)
19	Set to treat the encoder error bit as a warning (no fault)
20	If set, encoder error and warning bits are active low

21	Set if encoder status bits are sent before position data, clear if status bits are sent after position data
22	If set, encoder error bit is transmitted before warning bit. If clear, warning bit sent first.
23	If set, error bits are sent after alignment bits. If clear, encoder error bits are sent between alignment bits and position data
24-27	Number of alignment bits (reserved bits sent before position info)
28	If set, use multi-mode encoder. If clear, use primary encoder.
29	If set, use multi-mode encoder. If clear, use a primary encoder
30	If set, use 2.5 MHz baud rate. If clear, use 4 MHz baud rate.

BiSS encoders are not always consistent with order in which data is sent. We treat data as consisting of three fields, position data <P>, 2 status bits <S> and optional alignment bits <A> which we ignore. Formatting bits identify order of these three fields.
Chart below shows order of fields based on format code. Note that data is always sent most significant bit first, so leftmost field is first transmitted.

Format Order of fields

0   <P>   <S>   <A>

1   <S>   <P>   <A>

2   <P>   <A>   <S>

3   <S>   <A>   <P>

Absolute A format. Tamagawa, Panasonic, Harmonic Drives, Sanyo Denki, N-A format. (Type 14)	
Bits	**Description**
0-5	Number of bits of single turn data
8-12	Number of bits of multi-turn data
16-19	Number of LSB to discard from reading
20-22	Number of consecutive CRC errors to ignore before generating an error
24-27	Encoder sub-type (0=Tamagawa, 1=Panasonic absolute, 2=HD systems, 3=Panasonic Incremental, 4=Sanyo Denki, 5=Tamagawa Single Turn)
28	Bit rate (set for 4 Mbit, clear for 2.5 Mbit)
29	If set, use multi-mode encoder. If clear, use a primary encoder.
30	If set, treat encoder battery errors as warnings.
31	Read the encoder's internal temperature sensor. Currently for Sanyo Denki and Panasonic encoders. Temperature value read from encoder can be read as encoder register 0.
Incremental Type E (Type 15)	
Bits	**Description**
0	If set, incremental encoder. If clear, absolute encoder.

	8	If set, disable interpolation of position

Gurley Virtual Absolute (Type 17)

Bits	Description
0	If set, invert sine/cosine signals
1	If set, invert virtual absolute signal
2	If set, use custom interface board (customer specific)
3	If set, use encoder digital index input for VABS. If clear, use encoder analog index (if available).
8	If set, switch from algorithmic phase initialization to encoder-based phasing as soon as absolute position is found
9	If set, treat any VABS warnings as encoder fault. If clear, these warnings set status bits but aren't treated as encoder errors.

Custom Absolute Encoder (Type 18)

Bits	Description
28	If set, use multimode input. If clear, use primary encoder input.

S2 Custom Encoder (Type 19)

Bits	Description
0-4	Number of bits of single turn position data / rev.
8	Set for incremental encoders, clear for absolute
9	Use multimode port if set. If clear use primary encoder interface
10	If set, treat encoder battery errors as warnings

Wire saving incremental (type 21):

Bits	Description
0-15	These bits are the same as a normal incremental encoder (type 0)
16	If set, reverse direction of simulated hall signals after powerup
17	If set, sample halls 10ms after they stabilize on power-up. If clear, sample them after 100ms.
18	If set, force the simulated hall signals to transition coincident with the index signal

Sankyo Absolute Encoder (Type 22):

Bits	Description
0	Ignore battery errors if set
1-31	Reserved

Custom Absolute Encoder M (Type 23)	
Bits	**Description**
0	Use the multimode port if set
1	Ignore battery errors

Tachometer input (type 25):

Bit	**Description**
0	If set, read tach from analog encoder sine input. If clear, read from analog reference input.

Tamagawa TS5643 (type 26):

Bits	**Description**
0	If set, use the multi-mode port
1	Don't generate faults on error bits reported by encoder

LOAD ENCODER OPTIONS   INDEX 0X2223

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	-	NO	F

Specifies various configuration options for the motor encoder. The mapping of option bits to function depends on the encoder type.

Quadrature Encoder	
Bit	**Description**
0	If set, ignore differential signal errors (if detected in hardware).
1	If set, select single ended encoder inputs (if available in hardware).
EnDat Encoder	
Bit	**Description**
0-4	Number of bits of single turn data available from encoder.
8-12	Number of bits of multiturn data available from encoder.
16	Set if analog inputs are supplied by encoder.

MOTOR ENCODER STATUS INDEX 0X2224

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	-	YES	R

Motor encoder status. This parameter gives additional status information for the encoder. Bits set in the status word are latched and cleared when the status value is read. The format of this status word is dependent on the encoder type. Many error bits are taken directly from encoder data stream. For a full description of what these error bits mean, please consult the encoder manufacturer.

Qaudrature	
Bits	**Description**
0	Only used for custom incremental encoders. Set on startup if encoder did not transmit hall information successfully.
1	Set on bad differential signal levels on any of encoder inputs.

BiSS	
Bits	**Description**
0	CRC error on data received from encoder
1	Encoder failed to transmit data to amp
2	Error bit on encoder stream is active
3	Warning bit on encoder stream is active
4	Encoder transmission delay is too long

EnDAT	
Bits	**Description**
0	CRC error on data received from encoder
1	Failed to detect encoder connected to drive
2	Error bit on encoder stream is active
3	Encoder failed to respond to request for position

Tamagawa & Panasonic	
Bits	**Description**
0	Over speed error reported by encoder
1	Absolute position error reported by encoder
2	Counting error reported by encoder
3	Counter overflow reported by encoder
5	Multi-turn error reported by encoder
6	Battery error reported by encoder
7	Battery warning reported by encoder
8	Error bit 0 reported by encoder
9	Error bit 1 reported by encoder
10	Comm error 0
11	Comm error 1
15	CRC error on data received from encoder

Sanyo Denki & Harmonic Drives (encoder type 14)	
Bits	**Description**
0	Battery warning reported by encoder
1	Battery error reported by encoder

3	Over speed reported by encoder
4	Memory error reported by encoder
5	STERR reported by encoder
6	PSERR reported by encoder
7	Busy error reported by encoder

Harmonic Drives (type 15)

Bits	Description
0	System error reported by encoder
1	Overflow error reported by encoder
2	Mode error reported by encoder
3	Battery error reported by encoder
4	CRC error on data received from encoder
5	No data received from encoder on read

Gurley Virtual Absolute (type 17)

Bits	Description
0	Amplitude of Sine/ Cosine signals is out of range
1	Encoder power current limited
2	Encoder moving too fast during initialization
3	Missing trigger signal (occurs when using custom interface hardware)
4	Virtual absolute signal changed state at incorrect time
5	Invalid virtual absolute data received
6	Encoder has not finished initializing position

Custom Absolute Encoder K (type 18)

Bits	Description
0	Busy bit from encoder set
1	ABSALM bit from encoder set
2	INPALM bit from encoder set
8	CRC error on data received from encoder

S2 Custom Encoder (type 19)

Bits	Description
0	Battery error alarm bit from encoder
1	Encoder error alarm from encoder
2	Battery warning alarm bit from encoder
3	Absolute error alarm bit from encoder
4	Over speed error alarm bit from encoder
5	Overheat error alarm bit from encoder
8	CRC error on data received from encoder
9	Encoder not responding to queries from drive

Hiperface encoder (type 20)

Bits	Description
0	Analog sine/ cosine signals missing or too low
1	Error bit received from encoder
2	No respone received from encoder

3	Checksum error on encoder response
4	Digital data from encoder doesn't agree with analog angle

Sankyo Absolute Encoder (type 22)

Bits	Description
0	Set if encoder is not responding to commands
1	Set if error bit is returned by encoder
2	Set if encoder returns incorrectly formatted data
3-7	Reserved
8	MR Sensor amplitude error reported by encoder
9	Multi rotation data error reported by encoder
10	Battery error reported from encoder
11	Reserved error returned by encoder
12	MR Sensor error reported by encoder
13	Over speed error reported by encoder
14	Temperature error returned by encoder
15	Reserved error returned by encoder

Custom Absolute Encoder M (type 23)

Bits	Description
0	CPU Error reproted by encoder
1	Reserved alarm reported by encoder
2	Data alarm reported by encoder
3	Thermal alarm reported by encoder
4	Thermal warning reported by encoder
5	Multi revolution alarm reported by encoder
6	Absolute position lost warning reported by encoder
7	Battery disconnected reported by encoder
8-12	Reserved
13	Incorrect data type returned from encoder
14	Encoder not responding to reads
15	Encoder CRC data error

Tamagawa TS6543 (type 26)

Bits	Description
0	Battery error reported by encoder
1	Overflow error reported by encoder
2	Over speed error reported by encoder
3	Battery alarm reported by encoder
4	Encoder preload status bit
5	Counter error reported by encounder
8	CRC error reading data from encoder
9	Encoder is not sending data

LOAD ENCODER STATUS   INDEX 0X2225

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	-	YES	R

Load encoder status. Same as parameter 0x12E, but for the load encoder.

PHASING MODE   INDEX 0X21C0

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below.	NO	F

Controls the mechanism used by the drive to compute the motor phasing angle. Determines what inputs the drive uses to initialize and maintain the phase angle. This variable is normally set using CME and stored to flash, but it can also be accessed via object 0x21C0.

The values that can be programmed into this object are as follows:

Code	Description
0	Standard mode. Use digital Hall inputs to initialize phase, then switch to an encoder to maintain it. The encoder is the primary sensing device with the Hall effect sensors used to monitor and adjust the phase angle as necessary during operation. This mode gives smooth operation and should be selected for most applications.
1	Trapezoidal (hall based) phasing. The Hall sensors are used for phasing all the time. This mode can be used if no encoder is available.
2	Like mode 0 except that the phase angle is not adjusted based on the Hall inputs. Hall sensors are still required to initialize the phase angle at startup.
3	Analog Halls (90°). Only available on drives with the necessary analog inputs.
4	DC Brush.
5	Algorithmic phase init mode (wake & wiggle).
6	Encoder based phasing. Use with resolver or Servo Tube motor.
7	Trapezoidal commutation with phase angle interpolation.

Algorithmic Phase Init Mode Details

When mode 5 is selected the drive enters a state machine used to initialize its phase. While the drive is performing this operation, bit 29 of the Manufacturer Status Register (0x1002) is set.

At the start of the phase init algorithm the drive will wait to be enabled. Once enabled, the main algorithm will start. If the drive is disabled during the phase initialization, it will wait to be enabled again and start over.

When the phase init algorithm ends successfully, bit 29 the Manufacturer Status Register (0x1002) is cleared and the drive will start using the encoder input to maintain its phasing info. If the algorithm fails for any reason, bit 29 remains set and bit 6 (phase error) is also set in the status word. The drive is then disabled.

To restart the phase init algorithm, object 0x21C0 can be written with the value 5. Bit 29 of the status register will immediately be set and the phase init algorithm will restart as soon as the drive is enabled.

Note that no profiles can be started until the phase init algorithm is completed.

MAX CURRENT TO USE WITH ALGORITHMIC PHASE INITIALIZATION   INDEX 0x21C2

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	TR	RF

See Algorithmic Phase Init Mode Details (p. 105).

ALGORITHMIC PHASE INITIALIZATION TIMEOUT   INDEX 0x21C3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $+2^{16}-1$	TR	RF

See Algorithmic Phase Init Mode Details (p. 105).

ALGORITHMIC PHASE INITIALIZATION CONFIG   INDEX 0x21C4

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	milliseconds	-2^{15} to $+2^{15}-1$	TR	RF

See Algorithmic Phase Init Mode Details (p. 105). Bit-mapped:

Bits	Description
0	If clear, use algorithmic phase initialization. If set force the phase angle to zero degrees.
1	If set, increment the initial phase angle by 90 degrees after each failed attempt.
2	If set, use the Motor Hall Offset (Index 0x2383, Sub-Index 6, p. 91), as the initial angle for the first attempt.
3-15	Reserved.

SECONDARY ANALOG REFERENCE OFFSET   INDEX 0x2314

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	millivolts	-2^{15} to $+2^{15}-1$	TR	F

Offset for secondary analog reference input.

SECONDARY ANALOG REF. CALIBRATION   INDEX 0x2315

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	millivolts	-2^{15} to $+2^{15}-1$	NO	F

Calibration offset for second analog reference input.

ANALOG ENCODER SINE OFFSET   INDEX 0x220B

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	A/D Units	-2^{15} to $+2^{15}-1$	NO	F

Encoder sine offset. This is set in A/D units and only used with resolvers and servo-tube motors. It gives an offset which is added to the encoder sine signal before calculating position.

ANALOG ENCODER COSINE OFFSET   INDEX 0x220C

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	A/D Units	-2^{15} to $2^{15}-1$	NO	F

Encoder cosine offset. This is set in A/D units and only used with resolvers and servo-tube motors. It gives an offset which is added to the encoder sine signal before calculating position.

ANALOG ENCODER COSINE SCALING FACTOR   INDEX 0x220D

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	BOOL	0 to $+2^{16}-1$	NO	F

Used by the resolver & servo tube encoder calculations. This scaler is used to adjust the cosine signal amplitude so that it's the same as the sine signal amplitude. If set to zero, both the scaling and offsets (0x18F,0x190) will be ignored. If non-zero the cosine is scaled by N/32768 where N is the value of this parameter.

ANALOG ENCODER SIGNAL MAGNITUDE   INDEX 0x220E

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.1 mV^2	-2^{15} to $+2^{15}-1$	T	.R

Returns the magnitude squared of the analog encoder signals (sin*sin + cos*cos).
This scaler is used to adjust the sin/cos signal amplitude.

MOTOR ENCODER CALIBRATION SETTINGS   INDEX 0x2226

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-	NO	RF

The meaning of this object is dependent on the encoder type.

LOAD ENCODER CALIBRATION SETTINGS   INDEX 0x2227

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-	NO	RF

The meaning of this object is dependent on the encoder type.

OPEN MOTOR WIRING CURRENT CHECK   INDEX 0x2142

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.01 Adc	-2^{15} to $+2^{15}-1$	NO	RF

If object 0x2199 is greater then zero, then during that time period on enable this current will be applied to the motor wiring to check to ensure that the motor is connected. If the programmed current can not be applied to the motor, then a motor disconnected fault will be flagged.

MOTOR TEMP THERMISTOR CONSTANTS   INDEX 0x220F

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[0..4] OF UINT	RW	80	-	R	RF

Steinhart constants for motor analog motor temperature sensor. This parameter is only used on drives that include a motor temperature sensor analog input (XEL, XPL, XML). For such drives, this parameter can be used to define the type of NTC thermistor connected to the analog input. If nonzero, the motor temperature (in deg C) will be read from parameter 0x2209 rather then the analog voltage. The parameter uses the same format as an output pin configuration, a 16-bit Integer followed by two 32-bit integers. The three integer values contain the A, B and C Steinhart coefficients for the motor thermistor. The three coefficients are scaled by the following constants: A: 1.0e6, B : 1.0e7, C: 1.0e10

For example, A thermistor with coefficients 1.4626e3, 2.4024e4 and 8.0353e8 would be configured with the three integer values: 1463, 2402 and 804.

MOTOR ENCODER SHIFT   INDEX 0x2228

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RW	16	0 to $2^{16}-1$	NO	RF

Motor encoder down-shift. This parameter is useful when using very high resolution encoders that would otherwise have limited speed and travel distance due to the range of position and velocity parameters. Setting the down-shift causes the position read from the encoder to be right-shifted before being used.

For example, setting this parameter to a value of 2 effectively cuts the encoder resolution by a factor of 4.

When this parameter is set, the servo loops use fractional encoder counts, therefore the encoder resolution is not completely lost.

LOAD ENCODER SHIFT   INDEX 0x2229

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RW	16	0 to $2^{16}-1$	NO	RF

Load encoder down-shift. Same as parameter 0x2228, but for the load encoder.

CONFIGURATION FOR ENCODER ADJUSTMENT TABLE   INDEX 0x222A

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RW	32	0,1	NO	RF

Bits	Description
0	Set to enable encoder adjustment table
1	Set for resolver angle adjustment tables, clear for normal encoder adjustment tables.

ENCODER FILTER CONFIG   INDEX 0x222B

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to $2^{32}-1$	NO	RF

Encoder error filter configuration. The encoder error filter can be used to detect and ignore bad position data from an encoder or temporary encoder errors. Bad encoder readings are detected by comparing an expected position (based on extrapolation of previous readings) and the actual reading from the encoder. Object not available on 8367 based products.

Bits	Description
0-3	Maximum number of consecutive bad samples to ignore. If zero then the filter is disabled.
4-15	Reserved
16-27	Maximum error between extrapolated reading and actual reading to consider the reading bad.
28-31	Reserved

ENCODER FILTER STATUS   INDEX 0x222C

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to $2^{32}-1$	NO	R

Encoder error filter status. This can be cleared by writing zero to it.

Bits	Description
0-3	Count of consecutive bad readings
4-7	Reserved
8	Set if an encoder fault was generated by the filter.
9-15	Reserved
16-31	Total number of times the extrapolated position has been used due to a detected error.

MOTOR TEMP THERMISTOR CONSTANTS   INDEX 0x220F

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[0..4] OF UINT	RW	80	-	R	RF

Steinhart constants for motor analog motor temperature sensor. This parameter is only used on drives that include a motor temperature sensor analog input (XEL, XPL, XML). For such drives, this parameter can be used to define the type of NTC thermistor connected to the analog input. If nonzero, the motor temperature (in deg C) will be read from parameter 0x2209 rather then the analog voltage. The parameter uses the same format as an output pin configuration, a 16-bit Integer followed by two 32-bit integers. The three integer values contain the A, B and C Steinhart coefficients for the motor thermistor. The three coefficients are scaled by the following constants: A: 1.0e6, B : 1.0e7, C: 1.0e10

For example, A thermistor with coefficients 1.4626e3, 2.4024e4 and 8.0353e8 would be configured with the three integer values: 1463, 2402 and 804.

5.6   Real-time Drive and Motor Status Objects

ANALOG/DIGITAL REFERENCE INPUT VALUE   INDEX 0x2200

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	millivolts	-2^{15} to $+2^{15}-1$	T	R

Most recent value read from the reference A/D input (millivolts). Available on certain drives.

HIGH VOLTAGE REFERENCE   INDEX 0x2201

Type	Access	Units	Range (Vdc)	Map PDO	Memory
INTEGER16	RO	0.1 volts	-2^{15} to $+2^{15}-1$	T	R

The voltage present on the high-voltage bus. Also known as Bus Voltage.

DRIVE TEMPERATURE   INDEX 0x2202

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	degrees centigrade	-2^{15} to $+2^{15}-1$	T	R

The drive temperature.

SYSTEM TIME   INDEX 0x2141

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	milliseconds	0 to $2^{32}-1$	T	R

Time since startup.

WINDING A CURRENT   INDEX 0x2203

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

The current present on one of the motor windings (0.01-amp units).

WINDING B CURRENT   INDEX 0x2204

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

The current present on one of the motor windings (0.01-amp units).

SINE FEEDBACK VOLTAGE   INDEX 0x2205

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	millivolts	-2^{15} to $+2^{15}-1$	T	R

The voltage present on the analog feedback, sine input (millivolts). Not available on all drives.

Also known as analog Sine Input Voltage.

COSINE FEEDBACK VOLTAGE   INDEX 0x2206

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	millivolts	-2^{15} to $+2^{15}-1$	T	R

Voltage present on the analog feedback, cosine input (millivolts). Available on certain drives.

A/D OFFSET VALUE   INDEX 0X2207

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	millivolts	-2^{15} to $+2^{15}-1$	YES	R

Primarily of diagnostic interest, this object gives the offset value applied to the internal A/D unit. It is part of a continuous calibration routine that the drive performs on itself while running.

CURRENT OFFSET A   INDEX 0X2210

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	YES	R

A calibration offset value, calculated at startup, and applied to the winding A current reading.

CURRENT OFFSET B   INDEX 0X2211

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	YES	R

A calibration offset value, calculated at startup, and applied to the winding B current reading

X AXIS OF CALCULATED STATOR CURRENT VECTOR   INDEX 0X2212

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

X axis of calculated stator current vector. Units: 0.01 A.

Y AXIS OF CALCULATED STATOR CURRENT VECTOR   INDEX 0X2213

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Y axis of calculated stator current vector. Units: 0.01 A.

STATOR VOLTAGE- X AXIS   INDEX 0X221A

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.1 volt	-2^{15} to $+2^{15}-1$	YES	R

X axis of stator output voltage vector.

STATOR VOLTAGE- Y AXIS   INDEX 0X221B

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.1 volt	-2^{15} to $+2^{15}-1$	YES	R

Y axis of stator output voltage vector.

RMS CURRENT CALCULATION PERIOD   INDEX 0X2114

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	milliseconds	-2^{15} to $+2^{15}-1$	TR	F

This sets the period over which the RMS current is calculated. If this value is set to zero, then the RMS current will be updated each time it is read for the period since the last read. In this case, the RMS current must be read at least once every 65536 current loop periods (about every 4 seconds) for the returned RMS values to be accurate.

RMS CURRENT OVER SET PERIOD   INDEX 0x2115

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

RMS current over the period set in parameter 0x130.

MOTOR PHASE ANGLE   INDEX 0x2260

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	degrees	0 - 360	T	R

Motor phase angle, derived from motor commutation.

Hall State   Index 0x2261

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	Bits 0~2	0 - 15	T	R

The lower three bits of the returned value give the present state of the Hall input pins.

The Hall state is the value of the Hall lines AFTER the ordering and inversions specified in the Hall wiring configuration have been applied.

MOTOR PHASE ANGLE   INDEX 0x2262

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	degrees	0 - 360	TR	R

Same as 0x2260 but writeable. Writes are only useful when running in diagnostic micro-stepping mode.

ENCODER PHASE ANGLE   INDEX 0x2263

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	degrees	0 - 360	T	R

For feedback types, such as resolver, that can also calculate phase angle information.
This parameter allows the phase information to be read directly.

PHASE ADVANCE   INDEX 0x2264

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	degrees	-180 to 180	R	RF

Phase advance scaled so 32k is 180 degrees.

Note: Preliminary experimental parameter subject to change. Object not available on 8367-based products.

DIGITAL COMMAND INPUT SCALING FACTOR   INDEX 0x2321

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-	TR	RF

Digital Command Input Scaling Factor. This value gives the amount of current to command at 100% PWM input. The scaling depends on what the PWM input is driving:

Current mode: 0.01 A Velocity (Accelus): 0.1 counts/s

In position mode the scaling factor is a ratio of two 16-bit values. The first word passed gives the numerator and the second word gives the denominator. This ratio determines the number of encoder units moved for each pulse (or encoder count) input. For example, a ratio of 1/3 would cause the motor to move 1 encoder unit for every three input steps.

5.7   Digital I/O Configuration Objects

INPUT PIN STATES   INDEX 0x2190

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	-	See *Description*, below	T	R

The 16-bit value returned by this command gives the current state (high/low) of the drive's input pins after debouncing. The inputs are returned one per bit as shown below.

Bits	Description
0	Input 1
1	Input 2
2	Input 3
3	Input 4
4	Input 5
5	Input 6
6	Input 7
7	Input 8
8	Input 9
9	Input 10
10	Input 11
11	Input 12
12	Input 13
13	Input 14
14	Input 15
15	Input 16

There is a PDO event associated with the input states object that can transmit a PDO any time an input pin changes state.

INPUT PIN STATE   INDEX 0x219A

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	R0	Bits	Bit mapped	T	R

32-bit version of parameter 0x2190. Useful on drives with more than 16 input pins.

Some drives have one or more pull-up resistors associated with their general-purpose input pins. On these drives, the state of the pull-ups can be controlled by writing to this register.

This register has one bit for each pull-up resistor available on the drive. Setting the bit causes the resistor to pull any inputs connected to it up to the high state when they are not connected. Bits 0 – 7 of this register are used to control pull-up resistor states. Each bit represents an input number. Bit 0 = IN1, bit 1 = IN2, etc.

On drives that allow groups of inputs to be configured as either single ended or differential, bit 8 controls this feature. Set bit 8 to 0 for single ended, 1 for differential.

INPUT PIN CONFIG REGISTER (16 BIT)   INDEX 0x2191

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below	TR	RF

Some drives have one or more pull-up resistors associated with their general-purpose input pins. On these drives, the state of the pull-ups can be controlled by writing to this register.

This register has one bit for each pull-up resistor available on the drive. Setting the bit causes the resistor to pull any inputs connected to it up to the high state when they are not connected. Bits 0 – 7 of this register are used to control pull-up resistor states. Each bit represents an input number. Bit 0 = IN1, bit 1 = IN2, etc.

On drives that allow groups of inputs to be configured as either single ended or differential, bit 8 controls this feature. Set bit 8 to 0 for single ended, 1 for differential.

INPUT PIN CONFIG REGISTER (32 BIT)   INDEX 0x219C

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	See *Description*, below	TR	RF

Some drives have one or more pull-up resistors associated with their general-purpose input pins. On these drives, the state of the pull-ups can be controlled by writing to this register.

This register has one bit for each pull-up resistor available on the drive. Setting the bit causes the resistor to pull any inputs connected to it up to the high state when they are not connected. Bits 0 – 7 of this register are used to control pull-up resistor states. Each bit represents an input number. Bit 0 = IN1, bit 1 = IN2, etc.

On drives that allow groups of inputs to be configured as either single ended or differential, bit 8 controls this feature. Set bit 8 to 0 for single ended, 1 for differential.

INPUT PIN CONFIGURATION   INDEX 0x2192

Type	Access	Bits	Range	Map PDO	Memory
Array	RW	256	-	NO	RF

This object consists of N identical sub-elements, where N is the number of input pins available on the drive. Sub-index 0 contains the number of sub-elements of this array.

INPUT PIN CONFIGURATION   INDEX 0x2192, SUB-INDEX 1-N

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below	TR	RF

These values allow functions to be assigned to each of the input pins. The available functions are:

Code	Description
0	No function
1	Reserved for future use (no function)
2	Reset the drive on the rising edge of the input.
3	Reset the drive on the falling edge of the input.
4	Positive side limit switch. Active high. See Misc. Drive Options Register (index 0x2420, p. 79).
5	Positive side limit switch. Active low. See Misc. Drive Options Register (index 0x2420, p. 79).
6	Negative side limit switch. Active high. See Misc. Drive Options Register (index 0x2420, p. 79).

Code	Description
7	Negative side limit switch. Active low. See Misc. Drive Options Register (index 0x2420, p. 79).
8	Motor temperature sensor. Active high.
9	Motor temperature sensor. Active low.
10	Disable drive when high. Clear latched faults on low to high transition.
11	Disable drive when low. Clear latched faults on high to low transition.
12	Reset on rising edge. Disable drive when high.
13	Reset on falling edge. Disable drive when low.
14	Home switch. Active high.
15	Home switch. Active low.
16	Disable drive when high.
17	Disable drive when low.
19	PWM synchronization. Only for high speed inputs; see drive data sheet.
20	Halt motor and prevent a new trajectory when high.
21	Halt motor and prevent a new trajectory when low.
22	High resolution analog divide when high.
23	High resolution analog divide when low.
24	High speed position capture on rising edge. Only for high speed inputs.
25	High speed position capture on falling edge. Only for high speed inputs.
26	Counter input, rising edge. Note: Upper byte of this parameter designates which Indexer register to store the count in.
27	Counter input, falling edge. Note: Upper byte of this parameter designates which Indexer register to store the count in.

INPUT PIN DEBOUNCE VALUES   INDEX 0X2195

Type	Access	Units	Range	Map PDO	Memory
ARRAY	RW	-	-	TR	RF

This object consists of N identical sub-index objects, where N is the number of input pins available on the drive. (Sub-index object 0 contains the number of elements of this record.) These values allow debounce times to be assigned to each of the input pins. Each sub-index object can be described as shown below:

INPUT PIN DEBOUNCE VALUES   INDEX 0X2195, SUB-INDEX 1-N

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 - 10,000	TR	RF

The debounce time for the input identified by the sub-index in milliseconds. This time specifies how long an input must remain stable in a new state before the drive recognizes the state.

RAW INPUT PIN VALUE (16 BIT)   INDEX 0X2196

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	-	See *Description*, below.	T	R

This object shows the current state of the input pins before debouncing.

The inputs are returned one per bit. The value of IN1 is returned in bit 0 (1 if high, 0 if low), IN2 in bit 1, etc. For input states with debouncing, see Input Pin States (index 0x2190, p. 113).

RAW INPUT PIN VALUE (32 BIT)   INDEX 0X219B

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below.	T	R

The 32-bit value returned by this command gives the current state (high/low) of the drive's input pins. Unlike input pin states, no debounce is applied when reading the inputs using this variable. The inputs are returned one per bit. The value of IN1 is returned in bit 0 (1 if high, 0 if low), IN2 in bit 1, etc. For input states with debouncing, see Input Pin States (index 0x2190, p. 113).

OUTPUT PIN CONFIGURATION   INDEX 0X2193

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	496	-	NO	RF

This array consists of N identical sub-elements, where N is the number of outputs. Sub-index 0 contains the number of sub-elements of this array.

Type	Access	Units	Range	Map PDO	Memory
Variable	RW	-	See *Description*, below.	NO	RF

The values programmed into these objects allow the drive's digital outputs to be driven by internal drive events, or externally driven.

Each output configuration consists of a 16-bit configuration word (bits 0-15), followed by a variable number of words (2-4), depending on the configuration code chosen. The configuration word is defined as follows:

Bits	Configuration
0-2	Define which internal register drives the output. The acceptable values for these bits are as follows:

	Value	Description
	0	Word 2 (bits 16-32) is used as a mask of the drive's Manufacturer Status Register object (index 0x1002, p. 62). When any bit set in the mask is also set in the Manufacturer Status Register object, the output goes active.
	1	Word 2 (bits 16-32) is used as a mask of the drive's Latched Event Status Register (index 0x2181, p. 64). When any bit set in the mask is also set in the Latched Event Status Register, the output goes active and remains active until the necessary bit in the Latched Event Status Register is cleared.
	2	Puts the output in manual mode. Additional words are not used in this mode, and the output's state follows the value programmed in the manual output control register.
	3	Word 2 (bits 16-32) is used as a mask of the drive's Trajectory Generator Status object (index 0x2252, p. 211). When any bit set in the mask is also set in the Trajectory Generator Status object the output goes active.
	4	Output goes active if the actual axis position is between the low position specified in words 2 and 3 (bits 16-47) and the high position specified in words 4 and 5 (bits 48-80).
	5	Output goes active if the actual axis position crosses, with a low to high transition; the position specified in words 2 and 3 (bits 16-47). The output will stay active for number of milliseconds specified in words 4 and 5 (bits 48-80).
	6	Same as 5 but for a high to low crossing.
	7	Same as 5 but for any crossing.

Bits	Configuration
3-7	Reserved for future use.
8	If set, the output is active low. If clear, the output is active high.
9-15	Reserved for future use.

OUTPUT STATES AND PROGRAM CONTROL   INDEX 0X2194

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	See *Description*, below.	TR	R

When read, this parameter gives the active/inactive state of the drive's general-purpose digital outputs. Each bit represents an input number. Bit 0 = digital output 1 (OUT1), bit 1 = OUT2, etc., up to OUT*n*, the number of digital outputs on the drive. Additional bits are ignored.

Outputs that have been configured for program control can be set by writing to this parameter (see the Output pin configuration object, index 0x2193, p. 116 for pin configuration details). Set a bit to activate the output. It will be activated high or low according to how it was programmed. Clear a bit to make the output inactive. If an output was not configured for program control it will not be affected.

OUTPUT COMPARE CONFIGURATION   INDEX 0X2160

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RW	32	Bit mapped	R	R

Bits	Description
0	Set to enable compare module
1	Set to invert the active state of the output
2	If set, toggle output on compare match.
	If clear, pulse output for a programmable time
3~4	Define mode of compare module (see 0x2161 below)
5~31	Reserved for future use. Should be set to zero.

OUTPUT COMPARE STATUS INDEX 0X2161

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RW	32	Bit mapped	T	R

Bits	Description
0	Current value of compare out (read only)
1	Set when position matches compare register 0. Write 1 to clear
2	Set when position matches compare register 1. Write 1 to clear
3~31	Reserved

OUTPUT COMPARE VALUE 0   INDEX 0X2162

Type	Access	Bits	Range	Map PDO	Memory
DINT	RW	32	-2^{31} to $+2^{31}-1$	R	R

Compare value 0

OUTPUT COMPARE VALUE 1   INDEX 0X2163

Type	Access	Bits	Range	Map PDO	Memory
DINT	RW	32	-2^{31} to $+2^{31}-1$	R	R

Compare value 1

OUTPUT COMPARE INCREMENT   INDEX 0X2164

Type	Access	Units	Range	Map PDO	Memory
DINT	RW	Encoder counts	-2^{31} to $+2^{31}-1$	R	R

Signed 32-bit value used to update compare values in some modes.

OUTPUT COMPARE PULSE WIDTH   INDEX 0X2165

Type	Access	Units	Range	Map PDO	Memory
DINT	RW	10 ns	-2^{31} to $+2^{31}-1$	R	R

Compare pulse period The lower 20-bits of this parameter give the period of the compare output pulse in 10 ns units.

DIGITAL CONTROL INPUT CONFIGURATION   INDEX 0X2320

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	TR	RF

Defines the configuration of the digital control inputs when the drive is running in a mode that uses them as a control source.

The lower 8 bits control the PWM input configuration for controlling current and velocity modes. The upper 8 bits configure the digital inputs when running in position mode.

Bits	Description		
0	If set, use PWM in signed/magnitude mode. If clear, use PWM in 50% duty cycle offset mode.		
1	Invert the PWM input if set.		
2	Invert the sign bit if set.		
3	Allow 100% duty cycle if set. If clear, treat 100% duty cycle as a zero command, providing a measure of safety in case of controller failure or cable break.		
4-7	Reserved for future use.		
8-9	Input pin interpretation for position mode (see below).		
	Value	Description	
	0	Step & Direction mode.	
	1	Separate up & down counters.	
	2	Quadrature encoder input.	
10-11	Reserved for future use.		
12	Count falling edges if set, rising edges if clear.		
13	Invert command signal.		
14-15	Selects source of digital position input command.		
	Value	Description	
	0	Single ended high-speed inputs.	
	1	Multi-mode encoder port.	
	2	Differential high-speed inputs.	
	3	Motor encoder port.	

DIGITAL CONTROL INPUT SCALING   INDEX 0x2321

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	See *Description*, below.	See *Description*, below.	TR	RF

When the drive is running in a mode that takes input from the digital control input pins (as determined by the setting of object 0x2300, Desired State), this object gives the amount of current to command at 100% PWM input. The scaling depends on what the PWM input is driving:

> Current mode: 0.01 A
> Velocity: 0.1 counts/s

In position mode the scaling factor is a ratio of two 16-bit values. The first word passed gives the numerator and the second word gives the denominator. This ratio determines the number of encoder units moved for each pulse (or encoder count) input.

For example, a ratio of 1/3 would cause the motor to move 1 encoder unit for every three input steps.

DIGITAL INPUTS   INDEX 0x60FD

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below.	T	R

This object gives the present value of the digital inputs of the drive. The lower 16 bits are defined by the device profile and show the value of input based on the function associated with them. The upper 16 bits give the raw values of the inputs connected to the drive in the same ordering as Input Pin States (index 0x2190, p. 113).

Bits	Description
0	Negative limit switch is active when set.
1	Positive limit switch is active when set.
2	Home switch is active when set.
3	Drive enable input is active when set.
4-15	Reserved.
16-31	Raw input mapping. These bits contain the same data as Input Pin States (index 0x2190, p. 113).

INPUT SHAPING FILTER   INDEX 0X2254

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[0..23] of UNSIGNED16	RW	384	RECORD	NO	F

This filter is used to modify the trajectory before its input to the position loop. This can be used to compensate for low frequency resonances in the load. The parameter is an array of 32-bit values. The first four values are used to store information about the input shaping filter (filter type, frequency, etc.) and are mostly unused by the firmware. The only exception is that the MSB of the first word should not be set to ensure compatibility with future firmware versions. The remaining 32-bit values are pairs of IEEE floating point values. Each pair defines a time (first value) and an impulse amplitude (second value).

Up to eight pairs may be passed for up to 8 impulses in the input shaping filter.
The time values are specified in seconds and must be >= 0.0. The impulse values are unit-less and must have an absolute magnitude of < 16.0.

TRAJECTORY GENERATION OPTION   INDEX 0X2255

Type	Access	Bits	Range	Map PDO	Memory
DINT	RW	32	Bit mapped	NO	RF

Trajectory options. This parameter is used to modify the behavior of some trajectory modes. Its interpretation depends on the trajectory mode being used.
The following trajectory modes currently make use of this parameter:

EtherCAT CSP Mode	
Bits	Description
0-7	Number of extra loop cycles to extrapolate trajectories if input data from the master is not received.
8-15	Reserved
16	If set, jump to quick stop mode if master data is not received within the number of cycles set in bits 0-7.
17	If set, and the interpolation time object (0x60C2) is non-zero, then the calculated velocity will be filtered, and a trajectory acceleration will also be calculated. If not set, velocity is unfiltered, and acceleration is not calculated (zero).
18-31	Reserved

TRAJECTORY SEQUENCE STATUS   INDEX 0X2256

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RO	32	0 to 2^{32} -1	T	R

Trajectory sequence buffer status. The trajectory sequence buffer is used in CANopen profile position mode and stores trajectory segments added using the 'set of setpoints' method described in the CANopen specification.This parameter allows the buffer status to be queried. Object not available on 8367 based products. It's bitmapped as follows:

EtherCAT CSP Mode	
Bits	Description
0-7	Number of free locations in the buffer
8-15	Number of full locations in the buffer
16-31	Reserved

REGISTRATION OFFSET FOR STEP & DIRECTION MODE   INDEX 0X2325

Type	Access	Bits	Range	Map PDO	Memory
DINT	RW	32	-2^{31} to $+2^{31}-1$	NO	R

Registration Offset For Pulse & Direction Mode. When running in pulse & direction mode (Desired State (0x24) = 23), this parameter may be used to inject an offset into the master position. The offset will immediately be cleared once it has been applied to the master position, so this parameter will normally be read back as zero when running in pulse and direction mode 23.

UV MODE CONFIGURATION   INDEX 0X2326

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	Bit mapped	NO	RF

UV configuration. This parameter is used to configure the drive when running in UV mode (desired state 5). Bit-mapped as follows (undocumented bits are reserved for future use):

Bits	Meaning	
0-1	Define the source of the UV command inputs	
	Value	Description
	0	PWM inputs.
	1	Analog reference inputs (for drives with two analog reference pins).
	2	Analog encoder inputs.
	3	Directly set over the serial/network interface.
2-7	Reserved	
8-9	Define the format of the UV inputs:	
	Value	Description
	0	120 degree current commands.
	1	90 degree current commands.
	2	Angle/Magnitude form. U input gives magnitude, V gives angle.
1015	Define the format of the UV inputs	
16	If set, the value of the Motor Hall Offset (0x4F) is added to the UV angle .	
17	If set, the drive will use field oriented control. Normally FOC is disabled in UV mode due to the ambiguity of the phase angle with zero inputs. This is best used when running in angle/magnitude format.	

UV MODE U INPUT   INDEX 0x2327

Type	Access	Bits	Range	Map PDO	Memory
INTEGER16	RW	16	-2^{15} to $+2^{15}-1$	TR	R

U input when running in UV mode. This parameter can be used to read the calculated U value, or to set a U value when the UV inputs are being directly set over the serial/network interface.

UV MODE V INPUT   INDEX 0x2328

Type	Access	Bits	Range	Map PDO	Memory
INTEGER16	RW	16	-2^{15} to $+2^{15}-1$	TR	R

V input when running in UV mode. This parameter can be used to read the calculated V value, or to set a V value when the UV inputs are being directly set over the serial/network interface.

PULSE & DIRECTION COUNTER   INDEX 0x2329

Type	Access	Bits	Range	Map PDO	Memory
INTEGER16	RW	16	-2^{15} to $+2^{15}-1$	NO	R

Raw counter value from pulse & direction input hardware. This can be read when running in any mode, not just pulse & direction modes. This parameter can be written also, but should not be written when the amp is being driven by the pulse & direction inputs. Writing in that mode will cause the amp to treat the change in the counter as real pulse inputs resulting in possible unexpected motion

PWM INPUT DUTY CYCLE   INDEX 0x232A

Type	Access	Bits	Range	Map PDO	Memory
INTEGER16	RO	16	-2^{15} to $+2^{15}-1$	T	R

PWM input duty cycle. This parameter can be used to read the duty cycle of the PWM input. The returned 16-bit value gives the duty cycle in the range +/-32k. Parameter 0x2320 is used to configure the PWM input.

PWM INPUT DEADBAND   INDEX 0x232B

Type	Access	Bits	Range	Map PDO	Memory
INTEGER16	RW	16	0 to $2^{15}-1$	NO	R

PWM input deadband. Range of 0 to 32k equals deadband of 0 to 100%. This parameter was added to plus products starting with version 2.75.

CROSS COUPLING POSITION LOOP KP   INDEX 0x2378

Type	Access	Bits	Range	Map PDO	Memory
INTEGER16	RW	16	-2^{15} to $+2^{15}-1$	NO	RF

Cross coupling KP gain. On dual axis drives this gain is applied to the difference in position error of the two axes.

POSITION OFFSET   INDEX 0x60B0

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-2^{31} to $+2^{31}-1$	TR	R

This object provides the offset for a target position in user-defined units.

VELOCITY OFFSET   INDEX 0x60B1

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-2^{31} to $+2^{31}-1$	TR	R

This object provides the offset for the target velocity in user-defined units. In CSP mode of operation this is the velocity feedforward value. In CSV mode, it is added to the commanded velocity from the drive.

TORQUE OFFSET   INDEX 0x60B2

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Rated Torque / 1000	-2^{15} to $+2^{15}-1$	TR	R

This object provides the offset for a target torque in user-defined units. Used in CSV and CSP modes of operation. In CSP & CSV modes, this provides torque feedforward. In CST mode, it holds the commanded additive torque value from the drive plus the target torque value.

COMMUTATION ANGLE   INDEX 0x60EA

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	$360° / 2^{16}$	0 to $2^{16}-1$	TR	R

Space vector modulation Q-axis phase angle.

CONFIGURE I/O OPTIONS   INDEX 0x2198

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-	NO	RF

This parameter is used to configure optional features of the general purpose I/O for Plus module drives. Supported drives are AEM, APM, AE2, AP2, GEM, GPM, ME3, MP3, ME4, MP4, SEM, SPM, SE2, SP2, SE4, SP4.

Bit-mapped as follows:

Bit	Description
0	Set to enable the I/O extension function. Clear to use the I/O as general purpose I/O.
1	When in the I/O extension mode, this bit determines whether the CANopen/EtherCAT status LED is treated as a single bi-color status LED, or whether it's two different LEDs (one red error LED, one green state LED).
24-31	Reserved.

I/O EXTENSION OPTIONS   INDEX 0x21A1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Bit Mapped	-	TR	RF

This parameter is used to configure the features of I/O extension when it is enabled.   Bit-mapped:

Bit	Description
0-7	Number of bits to transfer less 1 (E.g. set to 19 to transfer 20 bits)
8	Reserved
9	Automatically restart transmission if set
10	Leave the CS line low after transfer if set
11	Status bit indicating new receive data is available. Auto cleared when data is read via parameter 0x18E
12	Clock polarity setting
13	Data phase setting
14-15	Reserved
16-23	Clock period (100 ns units)
24-27	Reserved
28	Set to enable SPI I/O extension feature. Clear for LED & switches interface
29-31	Reserved

I/O EXTENSION TRANSMIT DATA   INDEX 0x21A2

Type	Access	Bits	Range	Map PDO	Memory
Array [0..15] of Unsigned16	RW	256	-	R	R

This parameter contains data to be sent to the SPI network.

I/O EXTENSION RECEIVE DATA   INDEX 0x21A3

Type	Access	Units	Range	Map PDO	Memory
Array [0..15] of Unsigned16	RO	-	-	T	R

This parameter contains data received from the SPI network. Additional information on I/O extension is available on the Copley Controls web-site.

Application note AN102: http://www.copleycontrols.com/Motion/pdf/IO-Extension.pdf

6 CONTROL LOOP CONFIGURATION

6.1 Overview: Nested Position, Velocity, & Current Loops

Nesting of Control Loops and Modes

Copley Controls drives use up to three nested control loops - current, velocity, and position - to control a motor in three associated operating modes.

In position mode, the drive uses all three loops. As shown in the typical system illustrated below, the position loop drives the nested velocity loop, which drives the nested current loop.
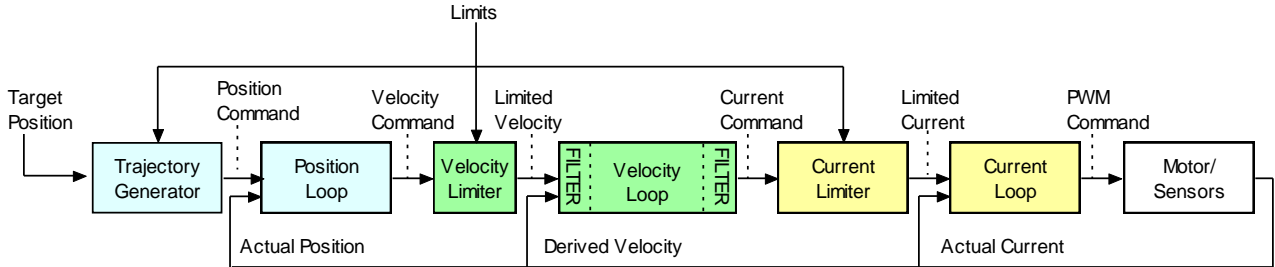
Figure 7—Nested control loops

In velocity mode, the velocity loop drives the current loop. In current mode, the current loop is driven directly by external or internal current commands.

Basic Attributes of All Control Loops

These loops (and servo control loops in general) share several common attributes:

Loop Attribute	Description
Command input	Every loop is given a value to which it will attempt to control. For example, the velocity loop receives a velocity command that is the desired motor speed.
Limits	Limits are set on each loop to protect the motor and/or mechanical system.
Feedback	The nature of servo control loops is that they receive feedback from the device they are controlling. For example, the position loop uses the actual motor position as feedback.
Gains	These are constant values that are used in the mathematical equation of the servo loop. The values of these gains can be adjusted during drive setup to improve the loop performance. Adjusting these values is often referred to as *tuning* the loop.
Output	The loop generates a control signal. This signal can be used as the command signal to another control loop or the input to a power drive.

The Position Loop

Position Loop Diagram

The CANopen master provides a target position to the drive's internal trajectory generator. In turn the generator provides the position loop a position command and velocity and acceleration limit values. The position loop applies corrective gains in response to feedback to forward a velocity command to the velocity loop. The inputs to the position loop vary with different operating modes. The following diagram summarizes the position loop in position profile mode.
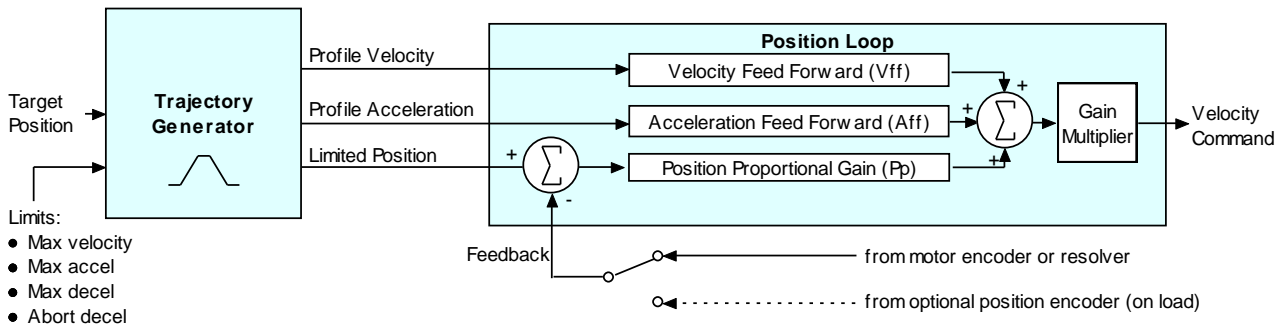
Figure 8—Position loop diagram

Trajectory Generator Inputs and Limits

The inputs to the trajectory generator include profile position, velocity, and acceleration values. They are accessed through different sets of mode-specific objects as summarized below.

Mode	Input Object Name/ID	Description	Page #
Homing	Homing Method / 0x6098	Defines the method to find the motor home position	188
	Homing Speeds / 0x6099	The sub-index objects of 0x6099 hold the two velocities (fast and slow) used when homing.	189
	Homing Acceleration / 0x609A	Defines the acceleration used for all homing moves.	189
	Home Offset / 0x607C	Used in homing mode as an offset between the home sensor position and the zero position.	190
Profile Position	Motion Profile Type / 0x6086	Selects the type of trajectory profile to use. Choices are trapezoidal, S-curve, and velocity.	158
	Target Position / 0x607A	Destination position of the move.	207
	Profile Velocity / 0x6081	The velocity that the trajectory generator attempts to achieve when running in position profile mode.	157
	Profile Acceleration / 0x6083	Acceleration that the trajectory generator attempts to achieve when running in position profile mode	157

Mode	Input Object Name/ID	Description	Page #
	Profile Deceleration / 0x6084	Deceleration that the trajectory generator attempts to achieve at the end of a trapezoidal profile when running in position profile mode.	157
	Trajectory Jerk Limit / 0x2121	Defines the maximum jerk (rate of change of acceleration) for use with S-curve profile moves.	211
Interpolated Position	IP move segment command / 0x2010	Used to send PVT segment data and buffer commands when running in interpolated position mode.	220

Position Loop Inputs

Inputs from the trajectory generator to the position loop are described below.

Input Object Name/ID	Description	Page #
Instantaneous Commanded Velocity / 0x2250	Velocity to which the position loop's velocity feed forward gain is applied.	142
Instantaneous Commanded Acceleration / 0x2251	Acceleration to which the position loop's acceleration feed forward gain is applied.	142
Position Demand Value / 0x6062	Motor position (in units of counts) to which the drive is currently trying to move the axis.	142

Position Loop Feedback

The feedback to the loop is the actual motor position, obtained from a position sensor attached to the motor (most often a quadrature encoder). This is provided by Position Actual Value object (index 0x6063, p. 142).

Position Loop Gains

The following gains are used by the position loop to calculate the output value:

Gain	Description
Pp - Position loop proportional	The loop calculates its Position Error (index 0x60F4, p. 144) as the difference between the Position Actual Value and the Position Demand Value. This error in turn is multiplied by the proportional gain value. The primary effect of this gain is to reduce the following error.
Vff - Velocity feed forward	The value of the Instantaneous Commanded Velocity object is multiplied by this value. The primary effect of this gain is to decrease following error during constant velocity.
Aff - Acceleration feed forward	The value of the Instantaneous Commanded Acceleration object is multiplied by this value. The primary effect of this gain is to decrease following error during acceleration and deceleration.

These gains are accessed through the sub-index objects of the Position Loop Gains object (index 0x2382, sub-index 1-6, p. 144).

Position Loop Output

The output of the position loop is a velocity value that is fed to the velocity loop as a command input. This output is associated with two objects, as described below.

Output Object Name/ID	Description	Page #
Velocity Command Value / Index 0x606B	Velocity that the velocity loop is currently trying to attain. In normal operation, this value is provided by the position loop and is identical to the Position loop control effort. Optionally, the velocity loop can be controlled by one of several alternate control sources. In this case, the Velocity command value comes from the analog reference input, the digital PWM inputs, or the internal function generator.	154
Position Loop Control Effort / Index 0x60FA	Normally, this value is provided by the position loop. When the velocity loop is driven by an alternate control source, the Position loop control effort object does not hold a meaningful value.	144

Modulo Count (Position Wrap)

The position variable cannot increase indefinitely. After reaching a certain value the variable rolls back. This type of counting is called modulo count. See bit 21 of the Manufacturer Status Register object (index 0x1002, p. 62).

The Velocity Loop

Overview of the Velocity Loop

As shown below, the velocity loop limiting stage accepts a velocity command, applies limits, and passes a limited velocity command to the input filter. The filter then passes a velocity command to the summing junction. The summing junction subtracts the actual velocity, represented by the feedback signal, and produces an error signal. (The velocity loop feedback signal is always from the motor feedback device even when an additional encoder is attached to the load.) The error signal is then processed using the integral and proportional gains to produce a current command. Programmable digital filters are provided on both the input and output command signals.
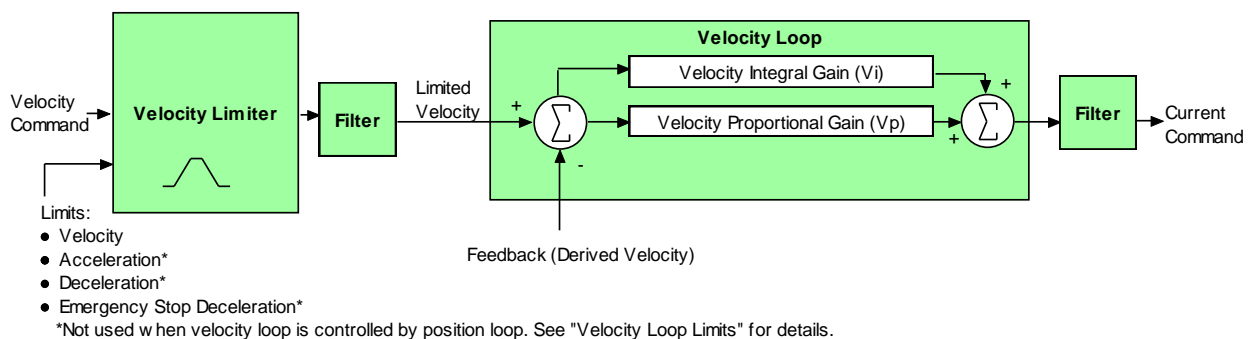
Figure 9—Velocity Loop diagram

Velocity Loop Limits

The velocity loop starts with a command limiter. This is useful because the position loop may produce large spikes in its output velocity command value that are beyond the safe operating range of the motor. During normal operation, with the velocity loop driven by the position loop, the limiter requires and accepts only a maximum velocity value.

Optionally, the velocity loop can be driven by an alternate source of control (such as such as the device's serial port, digital I/O channels, analog reference, or internal generator), without input from the position loop. (See Overview) In these cases, the velocity loop limiter also requires and accepts maximum acceleration and deceleration values. Velocity limiter parameters are accessed through the following objects:

Limiter Object Name/ID	Page #
Velocity Loop – Maximum Velocity / 0x2103 (used in all control modes)	151
*Velocity Loop Maximum Acceleration / Index 0x2100 (used only without position loop)	151
*Velocity Loop Maximum Deceleration/ Index 0x2101 or 0x60C6 (used only without position loop)	151
Velocity Loop Emergency Stop Deceleration / 0x2102 (used only without position loop)	151
*Not used when velocity loop is controlled by position loop.	

Velocity Loop Input

The output of the velocity loop limiter is the input of the velocity loop. It is accessed through the object Limited Velocity (index 0x2230, p. 152).

Velocity Loop Gains

The velocity loop uses the velocity gains. See Velocity Loop Configuration Objects.

VELOCITY LOOP MAXIMUM ACCELERATION   INDEX 0X2100

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	1000 enc counts / sec2	0 to $2^{32}-1$	TR	RF

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is positive (i.e. the speed is increasing in either direction).

VELOCITY LOOP MAXIMUM DECELERATION   INDEX 0X2101 OR 0X60C6

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	1000 enc counts / sec2	0 to $2^{32}-1$	TR	RF

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is negative (i.e. the speed is decreasing in either direction). With 0x60C6, user-defined units are possible using the factor group variables.

VELOCITY LOOP EMERGENCY STOP DECELERATION   INDEX 0X2102

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	1000 enc counts / sec2	0 to $2^{32}-1$	TR	RF

The deceleration rate used during the time that the drive is trying to actively stop a motor before applying the brake output.

Also known as the Velocity Loop Fast Stop Ramp.

Note that this feature is not used when the position loop is driving the velocity loop. In that case, the trajectory generator's abort acceleration is used.

VELOCITY LOOP – MAXIMUM VELOCITY   INDEX 0X2103

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	0 to $2^{32}-1$	TR	RF

This velocity value is a limit on the commanded velocity used by the velocity loop.
Also known as the Velocity Loop Velocity Limit. The velocity loop's commanded velocity can be generated by several sources, including the output of the position loop. Velocity Loop-Maximum Velocity allows that velocity to be limited to a specified amount.

VELOCITY ERROR WINDOW – PROFILE POSITION INDEX 0X2104

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	0 to 2^{32}-1	TR	RF

Also known as the Velocity Tracking Window, this object defines the velocity loop error window. If the absolute velocity error exceeds this value, then the velocity window bit of the

Manufacturer Status *Register* object (index 0x1002, p. 63) is set. The Velocity Window bit will only be cleared when the velocity error has been within the Velocity Error Window for the timeout period defined in the Velocity Error Window Time object (index 0x2120, p. 70).

VELOCITY ERROR WINDOW TIME INDEX 0X2105

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to 2^{16}-1	TR	RF

Also known as Velocity Tracking Time. When the absolute velocity error remains below the limit set in the Velocity Error Window – Profile Position object (index 0x2104, p. 146) the Velocity Window bit (bit 28) in the

Manufacturer Status Register object (index 0x1002, p. 63) is cleared.

VELOCITY LOOP OUTPUT FILTER COEFFICIENTS INDEX 0X2106

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[0..13] of UINT	RW	224	-	NO	RF

Programs the filter coefficients of a bi-quad filter structure that acts on the velocity loop output. Contact Copley Controls for more information.

HALL VELOCITY MODE SHIFT VALUE INDEX 0X2107

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	RF

This parameter is only used in Hall velocity mode. It specifies a left shift value for the position and velocity information calculated in that mode.

VELOCITY LOOP COMMAND FILTER COEEFICIENTS INDEX 0X2108

Type	Access	Units	Range	Map PDO	Memory
ARRAY[0..13] of UINT	RW	224	-	NO	RF

Programs the filter coefficients of a bi-quad filter structure that acts on the velocity loop input. Contact Copley Controls for more information.

ANALOG INPUT FILTER COEFFICIENTS INDEX 0X2109

Type	Access	Units	Range	Map PDO	Memory
ARRAY[0..13] of UINT	RW	224	-	NO	F

Programs the filter coefficients of a bi-quad filter structure that acts on the analog reference input at servo loop update rate (3 kHz). Contact Copley Controls for more information.

LMITED VELOCITY   INDEX 0X2230

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts/sec	-2^{31} to $+2^{31}-1$	T	R

This is the commanded velocity after it passes through the velocity loop limiter and the velocity command filter. It is the velocity value that the velocity loop will attempt to achieve.

LOAD ENCODER VELOCITY   INDEX 0X2231

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts / sec	-2^{31} to $+2^{31}-1$	T	R

Also known as Position Encoder Velocity. Copley Controls supports the use of two encoders on a system, where the motor encoder is on the motor and the load or position encoder is on *the* load (the device being controlled). In such a system, the actual velocity objects read the motor encoder velocity, and the velocity loop acts on the motor encoder input. Object 0x2231 reads the load encoder velocity.

UNFILTERED MOTOR ENCODER VELOCITY   INDEX 0X2232

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}-1$	T	R

Unfiltered motor velocity.

VELOCITY LOOP ERROR   INDEX 0X2233

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}-1$	T	R

Velocity loop error.

LOAD ENCODER VELOCITY   INDEX 0X2234

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}-1$	T	R

Load encoder velocity.

PROGRAMMED VELOCITY COMMAND   INDEX 0X2341

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	-2^{31} to $+2^{31}-1$	TR	RF

Gives the commanded velocity value when running in programmed velocity mode (see mode 11, Desired State object, p. 68, and Overview, p. 228).

VELOCITY LOOP GAINS   INDEX 0X2381

Type	Access	Bits	Range	Map PDO	Memory
Record	RW	112	-	NO	RF

This object contains the various gain values used to optimize the velocity control loop.

VELOCITY LOOP PROPORTIANAL GAIN   INDEX 0X2381, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

This gain value is multiplied by the velocity loop error. The velocity loop error is the difference between the desired and actual motor velocity.

VELOCITY LOOP INTEGRAL GAIN   INDEX 0X2381, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

This gain value is multiplied by the integral of the velocity loop error.

VELOCITY LOOP ACCELERATION FEED FORWARD   INDEX 0X2381, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

This gain value is multiplied by the Instantaneous Commanded Acceleration (index 0x2251, p. 137) from the trajectory generator. The result is added to the output of the *velocity* loop.

VELOCITY LOOP GAIN SCALER   INDEX 0X2381, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-1, 0, +8	TR	RF

Velocity loop output is shifted this many times to arrive at the commanded current value. Positive values result in a right shift while negative values result in a left shift. The shift allows *the* velocity loop gains to have reasonable values for very high- or low-resolution encoders.

Recommended values for this parameter are 8, 0 or -1.

VELOCITY LOOP VI DRAIN (INTEGRAL BLEED)   INDEX 0X2381, SUB-INDEX 5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

Modifies the effect of velocity loop integral gain. The higher the Vi Drain value, the faster the integral sum is lowered.

VELOCITY LOOP COMMAND FEED FORWARD   INDEX 0X2381, SUB-INDEX 6

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $+2^{15}-1$	TR	RF

The input command (after limiting) to the velocity loop is scaled by this value and added in to the output of the velocity loop.

ACTUAL MOTOR VELOCITY INDEX   INDEX 0X6069

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}-1$	T	R

Actual motor velocity.

VELOCITY SENSOR SELECTION   INDEX 0X606A

Type	Access	Units	Range	Map PDO	Memory
See description	RW	-	0	TR	R

This object specifies how actual velocity is measured. Currently, Copley Controls drives support only the use of position encoders for calculation of actual velocity. This should be set to zero.

Any value *other* than zero will return an error.

VELOCITY COMMAND VALUE   INDEX 0x606B

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts/sec	-2^{31} to $+2^{31}-1$	T	R

Also known as commanded velocity. The velocity that the velocity loop is currently trying to attain.

When the drive is running in homing, profile position, or interpolated position mode, the velocity *command* value is the output of the position loop, and the input to the velocity loop.

Copley Controls CANopen drives support some modes in which the velocity command is produced from a source other than the position loop. In these modes, the command velocity comes from the analog reference input, the digital PWM inputs, or the internal function generator.

User defined units are achievable using the factor group objects.

ACTUAL VELOCITY   INDEX 0x606C

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts/sec	-2^{31} to $+2^{31}-1$	T	R

This object contains exactly the same information as object 0x6069. User defined units are achievable using the factor group objects.

VELOCITY ERROR WINDOW – PROFILE VELOCITY   INDEX 0x606D

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	0.1 counts/sec	0 to $2^{16}-1$	TR	R

Object 0x606D holds the same value as index 0x2104. It is included because the CANopen Profile for Drives and Motion Control (DSP 402) mandates it for support of *profile* velocity mode operation. In the Copley Controls implementation, 0x2104 and 0x606D differ only in the data type. Object 0x606D is UNSIGNED16 and 0x2104 is INTEGER32. Changes made to either object affect both objects.

VELOCITY ERROR WINDOW TIME   INDEX 0x606E

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $2^{16}-1$	TR	R

Object 606E holds the same value as 0x2105. It is included because the CANopen Profile for Drives and Motion Control (DSP 402) mandates it for support of profile *velocity* mode operation. Changes made to either 0x606E or 0x2105 affect both objects.

VELOCITY THRESHOLD   INDEX 0x606F

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	0.1 enc counts / sec	0 to $2^{16}-1$	R	R

This object determines the threshold to use when considering the state of the speed=0 bit of the status word. User defined units are achievable using the factor group objects.

VELOCITY THRESHOLD TIME   INDEX 0x6070

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	ms	0 to $2^{16}-1$	R	R

This object shall indicate the configured velocity threshold time.

POSITION RANGE LIMIT   INDEX 0x607B

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	80	-	R	R

This object indicates the maximum and minimum position range limits applied to the Position Demand values. On reaching either limit the drive will wrap to the other. Wrap-around can be prevented by setting *the* Software Position Limits inside the Position Range Limits.
Sub-index 0 contains the number of sub-elements in this record.

MINIMUM POSITION RANGE LIMIT   INDEX 0x607B, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	R	R

User defined units are achievable using the factor group objects.

MAXIMUM POSITION RANGE LIMIT   INDEX 0x607B, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	R	R

User defined units are achievable using the factor group objects.

SOFTWARE POSITION LIMITS   INDEX 0x607D

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	80	-	R	RF

This object holds the maximum and minimum absolute position limits for the Position Demand value and Position Actual value. They are only in effect after the drive has been referenced (Homing is successful). User defined units are achievable using the factor group objects.

MINIMUM SOFTWARE POSITION LIMITS   INDEX 0x607D, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	TR	RF

Software limits are only in effect after the drive has been referenced (i.e. homing has been successfully completed). Set to less than negative software limit to disable.

MAXIMUM SOFTWARE POSITION LIMITS   INDEX 0x607D, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	TR	RF

Software limits are only in effect after the drive has been referenced (i.e. homing has been successfully completed). Set to greater than positive software limit to disable.

MAXIMUM PROFILE VELOCITY   INDEX 0x607F

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to $2^{32}-1$	R	R

The maximum allowed velocity in either direction during a profile velocity move.

The units are user-defined via Factor Group settings.

VELOCITY LOOP GAINS   INDEX 0X60F9

Type	Access	Units	Range	Map PDO	Memory
RECORD	RW	-	-	YES	R

This object is no longer recommended. Use object 0x2381 (p.149). This object contains the various gain values used to optimize the velocity control loop.

TARGET VELOCITY   INDEX 0X60FF

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	-2^{31} to $+2^{31}-1$	TR	R

In profile velocity mode, this object is an input to the drive's internal trajectory generator. Any change to the target velocity triggers an immediate update to the trajectory generator.

Note that this is different from the way the profile position works. In that mode, changing the trajectory input parameters doesn't affect the trajectory generator until bit 4 of the Control Word object (index 0x6040, p. 61) has been changed from 0 to 1.

User defined units are achievable using the factor group objects.

MAXIMUM MOTOR SPEED   INDEX 0X6080

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	-2^{31} to $+2^{31}-1$	TR	R

In profile velocity mode, this object is an input to the drive's internal trajectory generator. Any change to the target velocity triggers an immediate update to the trajectory generator.

Note that this is different from the way the profile position works. In that mode, changing the trajectory input parameters doesn't affect the trajectory generator until bit 4 of the Control Word object (index 0x6040, p. 61) has been changed from 0 to 1.

User defined units are achievable using the factor group objects.

PROFILE VELOCITY   INDEX 0X6081

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts/sec	0 to $2^{32}-1$	TR	R

In profile position mode, this value is the velocity that the trajectory generator will attempt to achieve. Note that the value programmed here is not passed to the internal trajectory generator until the *move* has been started or updated using the Control Word. See Operation, p. 193, for more information.

END VELOCITY   INDEX 0X6082

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts/sec	0 to $2^{32}-1$	NO	R

This object indicates the targeted velocity that the drive shall have upon reaching the target position. Normally, the drive would use an end velocity of 0.

User defiend units are achievable using the *factor* group objects.

Note: This object is only supported with the trapazoidal trajectory generator.

PROFILE ACCELERATION   INDEX 0x6083

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts/sec^2	0 to $2^{32}-1$	TR	RF

In profile position mode, this value is the acceleration that the trajectory generator attempts to achieve. For S-curve moves, this value is also used to decelerate at the end of the move.

Note that the value programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 193, for more information.

User defined units are achievable using the factor group objects.

PROFILE DECELERATION   INDEX 0x6084

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts/sec^2	0 to $2^{32}-1$	TR	RF

Deceleration that the trajectory generator uses at the end of a trapezoidal profile when running in position profile mode.

Note that this value is only used when running trapezoidal or profile position special velocity mode profiles. The S-curve profile generator uses the Profile Acceleration object (index 0x6083, p. 153) as the acceleration target for both the start and end of moves.

Note that the value programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 193, for more information.

User defined units are achievable using the factor group objects.

QUICK STOP DECELERATION   INDEX 0x6085

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts/sec^2	0 to $2^{32}-1$	TR	RF

Also known as Trajectory Abort Deceleration. This object gives the deceleration value used when a trajectory needs to be stopped as the result of a quick stop command.

When a quick stop command is issued, the command velocity is decreased by this value until it reaches zero. This occurs in all position modes (homing, profile position, and interpolated position modes), and for all trajectory generators (trapezoidal and S-curve).

Note that unlike most trajectory configuration values, this value is NOT buffered. Therefore, if the value of this object is updated during an abort, the new value is used immediately.

Also note that setting this object to zero causes the abort to run with unlimited deceleration. The command velocity is immediately set to zero.

User defined units are achievable using the factor group objects.

MOTION PROFILE TYPE   INDEX 0x6086

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	TR	R

This object selects the type of trajectory profile to use when running in profile position mode. The supported values for this object are:

Mode	Description
0	Trapezoidal profile mode.
3	S-curve profile mode (Jerk limited).
-1	Velocity mode.

The drive will not accept other values. See Operation, p. 193, for more information.

Note that the value programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 193, for more information.

PROFILE JERK   INDEX 0x60A4

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	48	-	NO	R

Indicates the configured set of jerk parameters that can be used during profile moves.
Sub-index 0 holds the number of elements in this object.

PROFILE JERK 1   INDEX 0x60A4, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to $2^{32}-1$	NO	R

Operates the same as 0x2121 but uses Factor Group units which are user-defined.

VELOCITY LOOP MAXIMUM ACCELERATION   INDEX 0x60C5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to $2^{32}-1$	R	R

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is positive (i.e. the speed is increasing in either direction).

VELOCITY LOOP MAXIMUM DECELERATION   INDEX 0x60C6

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to $2^{32}-1$	R	R

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is negative (i.e. the speed is decreasing in either direction).

Gain	Description
Vp - Velocity loop proportional	The velocity error (the difference between the actual and the limited commanded velocity) is multiplied by this gain. The primary effect of this gain is to increase bandwidth (or decrease the step-response time) as the gain is increased.
Vi - Velocity loop integral	The integral of the velocity error is multiplied by this value. Integral gain reduces the velocity error to zero over time. It controls the DC accuracy of the loop, or the flatness of the top of a square wave signal. The error integral is the accumulated sum of the velocity error value over time.

Velocity Loop Filters

The velocity loop contains two programmable digital filters. The input filter should be used to reduce the effects of a noisy velocity command signal. The output filter can be used to reduce the excitation of any resonance in the motion system.

Two filter classes can be programmed: the Low-Pass and the Custom Bi-Quadratic. The Low-Pass filter class includes the Single-Pole and the Two-Pole Butterworth filter types. The Custom Bi-Quadratic filter allows advanced users to define their own filters incorporating two poles and two zeros.

Program the filters using Velocity Loop Output Filter Coefficients (index 0x2106, p. 152) and Velocity Loop Command Filter Coefficients (index 0x2108, p. 152).

Velocity Loop Output

The output of the velocity loop is accessed in the Commanded Current object (index 0x221D, p. 161).

The Current Loop

Overview of the Current Loop

As shown below, the current limiter accepts a current command from the velocity loop, applies limits, and passes a limited current value to the summing junction. The summing junction takes the commanded current, subtracts the actual current (represented by the feedback signal), and produces an error signal. This error signal is then processed using the integral and proportional gains to produce a command. This command is then applied to the drive's power stage.
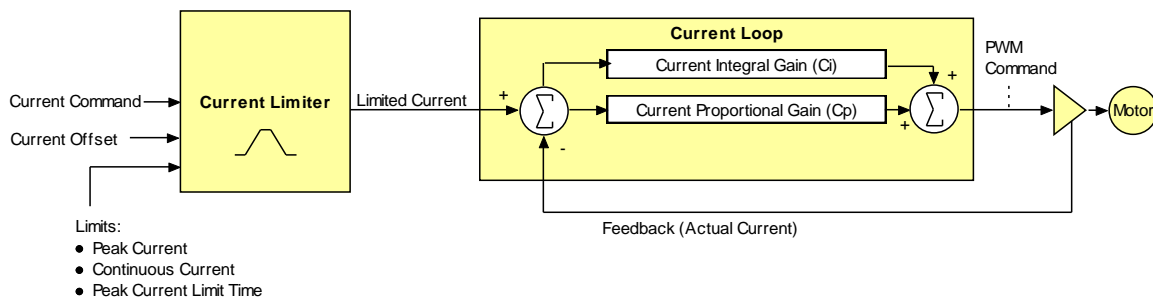
Figure 30—Current Loop diagram

Current Loop Limits

The commanded current value is first reduced based on a set of current limit parameters designed to protect the motor. These current limits are accessed through the following objects:

Output Object Name/ID	Description	Page #
User Peak Current Limit / 0x2110	Maximum current that can be generated by the drive for a short duration of time. This value cannot exceed the peak current rating of the drive.	160
User Continuous Current Limit / 0x2111	Maximum current that can be constantly generated by the drive.	160
User Peak Current Limit Time / 0x2112	Maximum amount of time that the peak current can be applied to the motor before it must be reduced to the continuous limit.	160

Current Loop Input

The output of the current limiting block is the input to the current loop. It is accessed through the object Limited Current object (index 0x221E, p. 161).

Current Loop Gains

The current loop uses these gains:

Gain	Description
Cp - Current loop proportional	The current error (the difference between the actual and the limited commanded current) is multiplied by this value. The primary effect of this gain is to increase bandwidth (or decrease the step-response time) as the gain is increased.
Ci - Current loop integral	The integral of the current error is multiplied by this value. Integral gain reduces the current error to zero over time. It controls the DC accuracy of the loop, or the flatness of the top of a square wave signal. The error integral is the accumulated sum of the current error value over time.

These gains are represented by Current Loop Gains (index 0x2380, p.161) and its sub-index objects.

Current Loop Output

The output of the current loop is a command that sets the duty cycle of the PWM output stage of the drive.

6.2 Position Loop Configuration Objects

INSTANTANEOUS COMMANDED VELOCITY   INDEX 0x2250

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts / sec	0 to $2^{32}-1$	T	R

This is the velocity output from the trajectory generator. It is the velocity by which the position loop's Position Loop Velocity Feed Forward gain (Index 0x2382, Sub-Index 2, p. 145) is multiplied.

INSTANTANEOUS COMMANDED ACCELERATION   INDEX 0x2251

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	10 counts / sec^2	0 to $(2^{32}-1)/10$ counts/sec^2	T	R

This is the acceleration output from the trajectory generator. It is the acceleration by which the position loop's gain (Index 0x2382, Sub-Index 3, p. 145) is multiplied.

POSITION DEMAND VALUE   INDEX 0x6062

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This is the motor position (in units of counts) to which the drive is currently trying to move the axis. This value is updated every servo cycle based on the drive's internal trajectory generator. Identical to Position Demand Value (index 0x60FC. p. 145).

POSITION ACTUAL VALUE   INDEX 0x6063

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	T	R

This is the actual motor position as calculated by the drive every servo cycle based on the state of the encoder input lines, and as used by the position loop. For single encoder systems, this is the same as the Motor Encoder Position object (index 0x2240). For dual encoder systems, it is the same as Load Encoder Position (index 0x2242, p. 148).

POSITION ACTUAL VALUE   INDEX 0x6064

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	T	R

This object holds the same value as Position Actual Value object (index 0x6063, p. 142). User defined units are achievable using the factor group objects.

TRACKING WARNING WINDOW   INDEX 0x6065

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	counts	0 to $+2^{32}-1$	TR	RF

This object holds the maximum position error that the drive will tolerate before indicating a tracking warning. If the absolute position error (defined as the difference between the actual motor position and the position command value) exceeds this window, then the warning bit (bit 19) of the Manufacturer Status Register (index 0x1002, p. 62) is set.

Note that this following error window generates a warning, not an drive fault. A separate tracking error window may be programmed which will cause an drive fault condition if exceeded. See the Tracking Error Window object (index 0x2120, p. 71) for details.

User defined units are achievable using the factor group objects.

FOLLOWING ERROR TIMEOUT   INDEX 0x6066

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	ms	0 to $2^{16}-1$	TR	RF

This object shall indicate the configured time for a following error condition.

POSITION TRACKING WINDOW   INDEX 0x6067

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	counts	0 to $+2^{32}-1$	TR	RF

Size of the drive's tracking window. When the absolute position error of the motor is less than or equal to the position tracking window value, the motor is considered to be tracking the desired position correctly. This is true both when moving and when resting in position.

The target reached bit (bit 10) is set in the Status Word (index 0x6041, p. 61) when the drive has finished running a trajectory, and the position error has been within the position tracking window for the programmed time.

The Manufacturer Status Register (index 0x1002, p. 62) has two bits that are affected by the tracking window. Bit 25 is set any time the motor position has fallen outside the position tracking window (whether in motion or not), and bit 27 is set when the motor position is outside the position tracking window, or the drive is in motion.

User defined units are achievable using the factor group objects.

POSITION TRACKING WINDOW TIME   INDEX 0x6068

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $+2^{16}-1$	TR	RF

Accesses the time component of the position tracking window. The motor will only be treated as tracking properly when the position error has been within the Position Tracking Window (index 0x6067, p. 143) for at least this long. The tracking window bit (bit 25) in the Manufacturer Status Register (index 0x1002, p. 62) will not be cleared until the position has been within the position tracking window for at least this long.

MAXIMUM SLIPPAGE-PROFILE VELOCITY MODE   INDEX 0x60F8

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	counts	0 to $2^{32}-1$	TR	R

Object 60F8 is included because the *CANopen Profile for Drives and Motion Control (DSP 402)* mandates it for support of profile velocity mode operation. This object is identical to Tracking Warning Window (index 0x6065, p. 143). A change to either object is reflected in the other.

User defined units are achievable using the factor group objects.

POSITION ERROR (FOLLOWING ERROR ACTUAL VALUE)   INDEX 0x60F4

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

Also known as following error. This object gives the difference, in units of counts, between the Position Actual Value object (index 0x6063, p. 142) and the Position Demand Internal Value object (index 0x60FC, p. 147).

This value is calculated as part of the position control loop. It is also the value that the various tracking windows are compared to. See Tracking Warning Window object (index 0x60FC, p. 147), Position Tracking Window object (index 0x6067, p. 143), and Tracking Error Window object (index 0x2120, p. 71).

User defined units are achievable using the factor group objects.

POSITION LOOP CONTROL EFFORT   INDEX 0x60FA

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	-	-2^{31} to $+2^{31}-1$	T	R

The position loop produces a command effort as the output of the position control loop. This object gives access to that value. Most common, this value represents the input to the velocity loop. When the velocity loop is enabled the default units are 0.1 counts/sec.

POSITION LOOP GAINS   INDEX 0x2382

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	128	-	YES	RF

This object contains the various gain values used to optimize the position control loop. Sub-index 0 contains the number of sub-elements of this record.

POSITION LOOP PROPORTIONAL GAIN   INDEX 0X2382, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 to $2^{16}-1$	TR	RF

This gain value is multiplied by the position loop error. The position loop error is the difference between the Position Demand Value (index 0x60FC, p. 147) and the Position Actual Value (index 0x6064, p. 142).

POSITION LOOP VELOCITY FEED FORWARD   INDEX 0x2382, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 to 2^{16}-1	TR	RF

This value is multiplied by the Instantaneous Commanded Velocity (index 0x2250, p. 142) generated by the trajectory generator. The product is added to the output of the position loop.

This gain is scaled by 1/16384. Therefore, setting this gain to 0x4000 (16384) would cause the input velocity to be multiplied by 1.0, and the result added to the output of the position loop.

POSITION LOOP ACCELERATION FEED FORWARD   INDEX 0x2382, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	RF

This value is multiplied by the Instantaneous Commanded Acceleration (index 0x2251, p. 142) generated by the trajectory generator. The product is added to the output of the position loop.

POSITION LOOP OUTPUT GAIN MULTIPLIER   INDEX 0x2382, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 to 2^{16}-1	TR	RF

The output of the position loop is multiplied by this value before being passed to the velocity loop. This scaling factor is calculated such that a value of 100 is a 1.0 scaling factor.

This parameter is most useful in dual loop systems.

POSITION LOOP INTEGRAL GAIN (KI)   INDEX 0x2382, SUB-INDEX 5

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	0 to 2^{15}-1	TR	RF

This gain value is multiplied by the integral of the position loop error.

POSITION LOOP DERIVATIVE GAIN (KD)   INDEX 0x2382, SUB-INDEX 6

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	0 to 2^{15}-1	TR	RF

This gain value is multiplied by the derivative of the position loop error.

POSITION LOOP PI DRAIN (INTEGRAL BLEED)   INDEX 0x2382, SUB-INDEX 7

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	0 to 2^{15}-1	TR	RF

Modifies the effect of position loop integral gain. The higher the Pi Drain value, the faster the integral sum is lowered.

POSITION LOOP GAINS   INDEX 0x60FB

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	YES	R

This object is no longer recommended. Use object 0x2382 (p.144). This object contains the various gain values used to optimize the position control loop. Sub-index 0 contains the number of sub-elements of this record.

POSITION LOOP PROPORTIONAL GAIN   INDEX 0X60FB, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	0 to 2^{15}-1	TR	R

This gain value is multiplied by the position loop error. The position loop error is the difference between the Position Demand Value (index 0x60FC, p. 147) and the Position Actual Value (index 0x6064, p. 142).

POSITION LOOP VELOCITY FEED FORWARD   INDEX 0X60FB, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	0 to 2^{15}-1	TR	R

This value is multiplied by the Instantaneous Commanded Velocity (index 0x2250, p. 142) generated by the trajectory generator. The product is added to the output of the position loop.

This gain is scaled by 1/16384. Therefore, setting this gain to 0x4000 (16384) would cause the input velocity to be multiplied by 1.0, and the result added to the output of the position loop.

POSITION LOOP ACCELERATION FEED FORWARD   INDEX 0X60FB, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Integers	0 to 2^{15}-1	TR	R

This value is multiplied by the Instantaneous Commanded Acceleration (index 0x2251, p. 142) generated by the trajectory generator. The product is added to the output of the position loop.

POSITION LOOP OUTPUT GAIN MULTIPLIER   INDEX 0X60FB, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 to 2^{16}-1	TR	R

The output of the position loop is multiplied by this value before being passed to the velocity loop. This scaling factor is calculated such that a value of 100 is a 1.0 scaling factor.

This parameter is most useful in dual loop systems.

POSITION LOOP INTEGRAL GAIN (KI)   INDEX 0X60FB , SUB-INDEX 5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Integers	0 to 2^{16}-1	TR	R

This gain value is multiplied by the integral of the position loop error.

POSITION LOOP DERIVATIVE GAIN (KD)   INDEX 0X60FB , SUB-INDEX 6

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	R

This gain value is multiplied by the derivative of the position loop error.

POSITION LOOP PI DRAIN (INTEGRAL BLEED)   INDEX 0X60FB , SUB-INDEX 7

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	R

Modifies the effect of position loop integral gain. The higher the Pi Drain value, the faster the integral sum is lowered.

CROSS COUPLING PROPORTIONAL (KP) GAIN   INDEX 0X2378

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $2^{15}-1$	NO	RF

Only supported on dual axis drives. This gain is applied to the difference in position error of the two axes.

CROSS COUPLING INTEGRAL (KI) GAIN   INDEX 0X2379

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $2^{15}-1$	NO	RF

Only supported on dual axis drives. This gain is applied to the difference in position error of the two axes.

CROSS COUPLING DRAIN (KD) GAIN   INDEX 0X237A

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $2^{15}-1$	NO	RF

Only supported on dual axis drives. This gain is applied to the difference in position error of the two axes.

MOTOR COGGING COMPENSATION   INDEX 0X237B

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Bits	-2^{15} to $+2^{15}-1$	NO	RF

Motor cogging compensation. This was added to plus drives starting with version 3.18 firmware. This scales the current command to the motor based on the sine of the phase angle plus a programmable offset. The parameter is mapped as follows. Object not available on 8367 based products.

Bit	Description
0-7	Gives an angular offset in units of 360/256 deg.
8-15	Reserved for future use
16-31	Gives scaling value. Scale = 1.0 + X/16384 where X is the unsigned value programmed in these bits. Resulting scale ranges from 0 <= scale < 5.0.

POSITION DEMAND INTERNAL VALUE   INDEX 0X60FC

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This value is the output of the trajectory generator and represents the commanded position input to the position control loop. Each servo cycle the trajectory generator will update this value, and the position loop will attempt to drive the motor to this position. Identical to Position Demand Value (index 0x6062, p. 142).

SOFTWARE LIMIT DECELERATION   INDEX 0X2253

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts / sec^2	0 to $2^{32}-1$	TR	RF

The deceleration rate used when approaching a software limit.

MOTOR ENCODER POSITION   INDEX 0X2240

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	TR	R

For single-encoder systems, this is the same as the Position Actual Value object (index 0x6063, p. 142). For dual-encoder systems this gives the motor position rather than the load encoder position. For more information, see Load Encoder Velocity (index 0x2231, p. 152).

LOAD ENCODER POSITION INDEX 0X2242

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	TR	R

For dual encoder systems, this object gives the load (position) encoder position and is the same as the Position Actual Value object (index 0x6063, p. 142). For single encoder systems, this object is not used.

RESOLVER ANGLE   INDEX 0X2243

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	degrees	-2^{15} to $+2^{15}-1$	T	R

Resolver angle scaled so 180 deg is 32768. Only valid when using a resolver as the motor encoder feedback. Reserved for other encoder types. Object not available on 8367 based products.

MINIMUM PWM PULSE WIDTH   INDEX 0X2323

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	microseconds	-2^{15} to $2^{15}-1$	NO	RF

Minimum PWM pulse width in microseconds. Used when running in PWM position mode. In this mode the PWM input pulse width is captured by the drive and used to calculate an absolute position using the following formula:

pos = ((PW-MIN) / (MAX-MIN)) * SCALE + OFFSET

Where this parameter is the minimum pulse width (MIN), parameter 0x13D is the maximum pulse width (MAX), parameter 0xA9 is the scaling factor (SCALE) and parameter 0x10F is the offset (OFFSET).

MAXIMUM PWM PULSE WIDTH   INDEX 0X2324

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	microseconds	-2^{15} to $2^{15}-1$	NO	RF

Maximum PWM pulse width used when running in PWM position mode.

6.3 Xenus Regen Resistor Objects

XENUS REGEN RESISTOR RESISTANCE   INDEX 0x2150

Type	Access	Units	Range (ohm)	Map PDO	Memory
UNSIGNED16	RW	0.01 Ω	0 to 655 ohm	TR	RF

Regen resistor resistance.

XENUS REGEN RESISTOR CONTINUOUS POWER   INDEX 0x2151

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	watts	0 to 2^{16}-1	TR	RF

Regen resistor, continuous power.

XENUS REGEN RESISTOR PEAK POWER   INDEX 0x2152

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	watts	0 to 2^{16}-1	TR	RF

Regen resistor, peak power.

XENUS REGEN RESISTOR PEAK TIME   INDEX 0x2153

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to 2^{16}-1	TR	RF

This parameter contains data to be sent to the SPI network.   Bit-mapped:

Regen resistor, peak time at peak power.

XENUS REGEN RESISTOR TURN-ON VOLTAGE   INDEX 0x2154

Type	Access	Units	Range (Vdc)	Map PDO	Memory
UNSIGNED16	RW	0.1 Vdc	0 to 6,554	TR	RF

Regen resistor, turn-on voltage.

XENUS REGEN RESISTOR TURN-OFF VOLTAGE   INDEX 0x2155

Type	Access	Units	Range (Vdc)	Map PDO	Memory
UNSIGNED16	RW	0.1 Vdc	0 to 6,554	TR	RF

Regen resistor, turn-off voltage.

XENUS REGEN RESISTOR MODEL STRING   INDEX 0x2156

Type	Access	Bits	Range	Map PDO	Memory
STRING[40]	RW	320	-	NO	F

Regen resistor model number string.

XENUS REGEN RESISTOR STATUS   INDEX 0X2157

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	-	See *Description*, below.	T	R

Describes regen system status. Bit-mapped as follows:

Bit	Description
0	Set if the regen circuit is currently closed.
1	Set if regen is required based on bus voltage.
2	Set if the regen circuit is open due to an overload condition. The overload may be caused by either the resistor settings or the internal drive protections.
3-15	Reserved for future use.

6.4 Velocity Loop Configuration Objects

VELOCITY LOOP MAXIMUM ACCELERATION   INDEX 0x2100

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	1000 enc counts / sec^2	0 to 2^{32}-1	TR	RF

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is positive (i.e. the speed is increasing in either direction).

VELOCITY LOOP MAXIMUM DECELERATION   INDEX 0x2101 OR 0x60C6

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	1000 enc counts / sec^2	0 to 2^{32}-1	TR	RF

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is negative (i.e. the speed is decreasing in either direction). With 0x60C6, user-defined units are possible using the factor group variables.

VELOCITY LOOP EMERGENCY STOP DECELERATION   INDEX 0x2102

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	1000 enc counts / sec^2	0 to 2^{32}-1	TR	RF

The deceleration rate used during the time that the drive is trying to actively stop a motor before applying the brake output. Also known as the Velocity Loop Fast Stop Ramp. Note that this feature is not used when the position loop is driving the velocity loop. In that case, the trajectory generator's abort acceleration is used.

VELOCITY LOOP – MAXIMUM VELOCITY   INDEX 0x2103

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	0 to 2^{32}-1	TR	RF

This velocity value is a limit on the commanded velocity used by the velocity loop.
Also known as the Velocity Loop Velocity Limit. The velocity loop's commanded velocity can be generated by several sources, including the output of the position loop. Velocity Loop-Maximum Velocity allows that velocity to be limited to a specified amount.

VELOCITY ERROR WINDOW – PROFILE POSITION   INDEX 0x2104

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	0 to 2^{32}-1	TR	RF

Also known as the Velocity Tracking Window, this object defines the velocity loop error window. If the absolute velocity error exceeds this value, then the velocity window bit of the Manufacturer Status Register object (index 0x1002, p. 62) is set. The Velocity Window bit will only be cleared when the velocity error has been within the Velocity Error Window for the timeout period defined in the Velocity Error Window Time object (index 0x2120, p. 71).

VELOCITY ERROR WINDOW TIME   INDEX 0x2105

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to 2^{16}-1	TR	RF

Also known as Velocity Tracking Time. When the absolute velocity error remains below the limit set in the Velocity Error Window – Profile Position object (index 0x2104, p. 151) the Velocity Window bit (bit 28) in the Manufacturer Status Register object (index 0x1002, p. 62) is cleared.

VELOCITY LOOP OUTPUT FILTER COEFFICIENTS   INDEX 0x2106

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[0..13] of UINT	RW	224	-	NO	RF

Programs the filter coefficients of a bi-quad filter structure that acts on the velocity loop output. Contact Copley Controls for more information.

HALL VELOCITY MODE SHIFT VALUE   INDEX 0x2107

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	RF

This parameter is only used in Hall velocity mode. It specifies a left shift value for the position and velocity information calculated in that mode.

VELOCITY LOOP COMMAND FILTER COEFFICIENTS   INDEX 0x2108

Type	Access	Units	Range	Map PDO	Memory
ARRAY[0..13] of UINT	RW	224	-	NO	RF

Programs the filter coefficients of a bi-quad filter structure that acts on the velocity loop input. Contact Copley Controls for more information.

ANALOG INPUT FILTER COEFFICIENTS   INDEX 0x2109

Type	Access	Units	Range	Map PDO	Memory
ARRAY[0..13] of UINT	RW	224	-	NO	F

Programs the filter coefficients of a bi-quad filter structure that acts on the analog reference input at servo loop update rate (3 kHz). Contact Copley Controls for more information.

LIMITED VELOCITY   INDEX 0x2230

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts/sec	-2^{31} to $+2^{31}$-1	T	R

This is the commanded velocity after it passes through the velocity loop limiter and the velocity command filter. It is the velocity value that the velocity loop will attempt to achieve.

LOAD ENCODER VELOCITY   INDEX 0x2231

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts / sec	-2^{31} to $+2^{31}$-1	T	R

Also known as Position Encoder Velocity. Copley Controls supports the use of two encoders on a system, where the motor encoder is on the motor and the load or position encoder is on the load (the device being controlled). In such a system, the actual velocity objects read the motor encoder velocity, and the velocity loop acts on the motor encoder input. Object 0x2231 reads the load encoder velocity.

UNFILTERED MOTOR ENCODER VELOCITY INDEX 0x2232

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}-1$	T	R

Unfiltered motor velocity.

VELOCITY LOOP ERROR INDEX 0x2233

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}-1$	T	R

Velocity loop error.

LOAD ENCODER VELOCITY INDEX 0x2234

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}-1$	T	R

Load encoder velocity.

PROGRAMMED VELOCITY COMMAND INDEX 0x2341

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	-2^{31} to $+2^{31}-1$	TR	RF

Gives the commanded velocity value when running in programmed velocity mode (see mode 11, Desired State *object*, p. 68, and Overview, p. 235).

VELOCITY LOOP GAINS INDEX 0x2381

Type	Access	Bits	Range	Map PDO	Memory
Record	RW	112	-	NO	RF

This object contains the various gain values used to optimize the velocity control loop.

VELOCITY LOOP PROPORTIONAL GAIN INDEX 0x2381, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

This gain value is multiplied by the velocity loop error. The velocity loop error is the difference between the desired and actual motor velocity.

VELOCITY LOOP INTEGRAL GAIN INDEX 0x2381, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

This gain value is multiplied by the integral of the velocity loop error.

VELOCITY LOOP ACCELERATION FEED FORWARD INDEX 0x2381, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

This gain value is multiplied by the Instantaneous Commanded Acceleration (index 0x2251, p. 142) from the trajectory generator. The result is added to the output of the velocity loop.

VELOCITY LOOP GAIN SCALER   INDEX 0x2381, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-1, 0, +8	TR	RF

Velocity loop output is shifted this many times to arrive at the commanded current value. Positive values result in a right shift while negative values result in a left shift. The shift allows the velocity loop gains to have reasonable values for very high- or low-resolution encoders. Recommended values for this parameter are 8, 0 or -1.

VELOCITY LOOP VI DRAIN (INTEGRAL BLEED)   INDEX 0x2381, SUB-INDEX 5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	RF

Modifies the effect of velocity loop integral gain. The higher the Vi Drain value, the faster the integral sum is lowered.

VELOCITY LOOP COMMAND FEED FORWARD   INDEX 0x2381, SUB-INDEX 6

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $+2^{15}$-1	TR	RF

The input command (after limiting) to the velocity loop is scaled by this value and added in to the output of the velocity loop.

ACTUAL MOTOR VELOCITY   INDEX 0x6069

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 enc counts / sec	-2^{31} to $+2^{31}$-1	T	R

Actual motor velocity.

VELOCITY SENSOR SELECTION   INDEX 0x606A

Type	Access	Units	Range	Map PDO	Memory
See description	RW	-	0	TR	R

This object specifies how actual velocity is measured. Currently, Copley Controls drives support only the use of position encoders for calculation of actual velocity. This should be set to zero. Any value other than zero will return an error

VELOCITY COMMAND VALUE   INDEX 0x606B

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts/sec	-2^{31} to $+2^{31}$-1	T	R

Also known as commanded velocity. The velocity that the velocity loop is currently trying to attain. When the drive is running in homing, profile position, or interpolated position mode, the velocity command value is the output of the position loop, and the input to the velocity loop.

Copley Controls CANopen drives support some modes in which the velocity command is produced from a source other than the position loop. In these modes, the command velocity comes from the analog reference input, the digital PWM inputs, or the internal function generator.

User defined units are achievable using the factor group objects.

ACTUAL VELOCITY   INDEX 0x606C

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	0.1 counts/sec	-2^{31} to $+2^{31}-1$	T	R

This object contains exactly the same information as object 0x6069.
User defined units are achievable using the factor group objects.

VELOCITY ERROR WINDOW – PROFILE VELOCITY   INDEX 0x606D

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	0.1 counts/sec	0 to $2^{16}-1$	TR	R

Object 0x606D holds the same value as index 0x2104. It is included because the *CANopen Profile for Drives and Motion Control (DSP 402)* mandates it for support of profile velocity mode operation. In the Copley Controls implementation, 0x2104 and 0x606D differ only in the data type. Object 0x606D is UNSIGNED16 and 0x2104 is INTEGER32. Changes made to either object affect both.

VELOCITY ERROR WINDOW TIME   INDEX 0x606E

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $2^{16}-1$	TR	R

Object 606E holds the same value as 0x2105. It is included because the *CANopen Profile for Drives and Motion Control (DSP 402)* mandates it for support of profile velocity mode operation. Changes made to either 0x606E or 0x2105 affect both objects

VELOCITY THRESHOLD   INDEX 0x606F

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	0.1 enc counts / sec	0 to $2^{16}-1$	R	R

This object determines the threshold to use when considering the state of the
speed=0 bit of the status word. User defined units are achievable using the factor group objects

VELOCITY THRESHOLD TIME   INDEX 0x6070

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	ms	0 to $2^{16}-1$	R	R

This object shall indicate the configured velocity threshold time.

POSITION RANGE LIMIT   INDEX 0x607B

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	80	-	R	R

This object indicates the maximum and minimum position range limits applied to the Position Demand values. On reaching either limit the drive will wrap to the other. Wrap-around can be prevented by setting the Software Position Limits inside the Position Range Limits.
Sub-index 0 contains the number of sub-elements in this record.

MINIMUM POSITION RANGE LIMIT   INDEX 0x607B, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	R	R

User defined units are achievable using the factor group objects.

MAXIMUM POSITION RANGE LIMIT   INDEX 0X607B , SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	R	R

User defined units are achievable using the factor group objects.

SOFTWARE POSITION LIMITS   INDEX 0X607D

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	80	-	R	RF

This object holds the maximum and minimum absolute position limits for the Position Demand value and Position Actual value. They are only in effect after the drive has been referenced (Homing is successful). User defined units are achievable using the factor group objects.

MINIMUM SOFTWARE POSITION LIMIT   INDEX 0X607D, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	TR	RF

Software limits are only in effect after the drive has been referenced (i.e. homing has been successfully completed). Set to less than negative software limit to disable.

MAXIMUM SOFTWARE POSITION LIMIT   INDEX 0X607D, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	TR	RF

Software limits are only in effect after the drive has been referenced (i.e. homing has been successfully completed). Set to greater than positive software limit to disable.

MAXIMUM PROFILE VELOCITY   INDEX 0X607F

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to $2^{32}-1$	R	R

The maximum allowed velocity in either direction during a profile velocity move.

The units are user-defined via Factor Group settings.

VELOCITY LOOP GAINS   INDEX 0X60F9

Type	Access	Units	Range	Map PDO	Memory
RECORD	RW	-	-	YES	R

This object is no longer recommended. Use object 0x2381 (p.153). This object contains the various gain values used to optimize the velocity control loop.

TARGET VELOCITY   INDEX 0X60FF

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	-2^{31} to $+2^{31}-1$	TR	R

In profile velocity mode, this object is an input to the drive's internal trajectory generator. Any change to the target velocity triggers an immediate update to the trajectory generator.

Note that this is different from the way the profile position works. In that mode, changing the trajectory input parameters doesn't affect the trajectory generator until bit 4 of the Control Word object (index 0x6040, p. 61) has been changed from 0 to 1.

User defined units are achievable using the factor group objects

MAXIMUM MOTOR SPEED   INDEX 0x6080

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	RPM	0 to 2^{32}-1	NO	R

The maximum motor speed allowable in either direction. Typically found in the motor's specifications.

PROFILE VELOCITY   INDEX 0x6081

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts/sec	0 to 2^{32}-1	TR	R

In profile position mode, this value is the velocity that the trajectory generator will attempt to achieve. Note that the value programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 200, for more information.

END VELOCITY   INDEX 0x6082

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts/sec	0 to 2^{32}-1	NO	R

This object indicates the targeted velocity that the drive shall have upon reaching the target position. Normally, the drive would use an end velocity of 0.

User defiend units are achievable using the factor group objects.

Note: This object is only supported with the trapazoidal trajectory generator.

PROFILE ACCELERATION   INDEX 0x6083

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts/sec^2	0 to 2^{32}-1	TR	RF

In profile position mode, this value is the acceleration that the trajectory generator attempts to achieve. For S-curve moves, this value is also used to decelerate at the end of the move.

Note that the value programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 200, for more information.

User defined units are achievable using the factor group objects.

PROFILE DECELERATION   INDEX 0x6084

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts/sec^2	0 to 2^{32}-1	TR	RF

Deceleration that the trajectory generator uses at the end of a trapezoidal profile when running in position profile mode.

Note that this value is only used when running trapezoidal or profile position special velocity mode profiles. The S-curve profile generator uses the Profile Acceleration object (index 0x6083, p. 157) as the acceleration target for both the start and end of moves.

Note that the value programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 200, for more information.

User defined units are achievable using the factor group objects.

QUICK STOP DECELERATION   INDEX 0x6085

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts/sec^2	0 to 2^{32}-1	TR	RF

Also known as Trajectory Abort Deceleration. This object gives the deceleration value used when a trajectory needs to be stopped as the result of a quick stop command. When a quick stop command is issued, the command velocity is decreased by this value until it reaches zero. This occurs in all position modes (homing, profile position, and interpolated position modes), and for all trajectory generators (trapezoidal and S-curve).

Note that unlike most trajectory configuration values, this value is NOT buffered. Therefore, if the value of this object is updated during an abort, the new value is used immediately. Also note that setting this object to zero causes the abort to run with unlimited deceleration. The command velocity is immediately set to zero.

User defined units are achievable using the factor group objects.

MOTION PROFILE TYPE   INDEX 0x6086

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	TR	R

This object selects the type of trajectory profile to use when running in profile position mode. The supported values for this object are:

Mode	Description
0	Trapezoidal profile mode.
3	S-curve profile mode (Jerk limited).
-1	Velocity mode.

The drive will not accept other values. See Operation, p. 200, for more information.

Note that the value programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 200, for more information.

PROFILE JERK   INDEX 0x60A4

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	48	-	NO	R

Indicates the configured set of jerk parameters that can be used during profile moves.
Sub-index 0 holds the number of elements in this object.

PROFILE JERK 1   INDEX 0x60A4 SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to 2^{32}-1	NO	R

Operates the same as 0x2121 but uses Factor Group units which are user-defined.

VELOCITY LOOP MAXIMUM ACCELERATION   INDEX 0x60C5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to 2^{32}-1	R	R

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is positive (i.e. the speed is increasing in either direction).

VELOCITY LOOP MAXIMUM DECELERATION   INDEX 0x60C6

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	User Defined	0 to $2^{32}-1$	R	R

This acceleration value limits the maximum rate of change of the commanded velocity input to the velocity loop. This limit only applies when the absolute value of the velocity change is negative (i.e. the speed is decreasing in either direction).

6.5   Current Loop Configuration Objects

USER PEAK CURRENT LIMIT   INDEX 0x2110

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	TR	RF

User peak current limit. Known as boost current on stepper drives. This value cannot exceed the peak (or boost) current rating of the drive.

USER CONTINUOUS CURRENT LIMIT   INDEX 0x2111

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	TR	RF

User Continuous Current Limit (Run Current on stepper drives). This value should be less than the User Peak Current Limit. The drive uses this value as an input to an I2T current limiting algorithm to prevent over stressing the load.

USER PEAK CURRENT LIMIT TIME INDEX 0x2112

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $2^{16}-1$	TR	RF

Specifies the maximum time at peak current. The drive uses this value as an input to an I2T current limiting algorithm to prevent over stressing the load. (Also: Time at Boost Current on stepper drives).

COMMANDED CURRENT RAMP RATE   INDEX 0x2113

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	mA/second	-2^{31} to $+2^{31}-1$	TR	F

Setting this to zero disables slope limiting in Profile Torque mode. It is also used when the drive is running in Programmed Current mode (Desired State object [index 0x2300, p. 68] = 1).

ACTUAL CURRENT, D AXIS INDEX 0x2214

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Part of the internal current loop calculation.

ACTUAL CURRENT, Q AXIS INDEX 0x2215

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Part of the internal current loop calculation.

CURRENT COMMAND, D AXIS   INDEX 0x2216

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Part of the internal current loop calculation.

CURRENT COMMAND, Q AXIS   INDEX 0X2217

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Part of the internal current loop calculation.

CURRENT LOOP OUTPUT, D AXIS INDEX 0X2218

Type	Access	Units	Range	PDO	Memory
INTEGER16	RO	0.1 V	-2^{15} to $+2^{15}-1$	T	R

Part of the internal current loop calculation. Also known as Terminal Voltage Stepper.

CURRENT LOOP OUTPUT, Q AXIS INDEX 0X2219

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.1 V	-2^{15} to $+2^{15}-1$	T	R

Part of the internal current loop calculation. Also known as Terminal Voltage Servo.

ACTUAL MOTOR CURRENT INDEX 0X221C

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Actual motor current.

COMMANDED CURRENT   INDEX 0X221D

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Instantaneous commanded current as applied to the current limiter.

LIMITED CURRENT INDEX 0X221E

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	0.01 amps	-2^{15} to $+2^{15}-1$	T	R

Output of the current limiter (input to the current loop).

PROGRAMMED CURRENT COMMAND   INDEX 0X2340

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	T	RF

This object gives the programmed current value used when running in programmed current mode (mode 1) or diagnostic micro-stepping mode (mode 42). (See Desired State object, p. 68, and Overview).

CURRENT LOOP GAINS   INDEX 0X2380

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	64	-	YES	-

This object contains the various gain values used to optimize the current control loop.
Sub-index 0 contains the number of sub-elements of this record.

CURRENT LOOP PROPORTIONAL GAIN INDEX 0X2380, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	F

This gain value is multiplied by the current error value. The current error is the difference between the desired current and the actual current.

CURRENT LOOP INTEGRAL GAIN   INDEX 0X2380, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to 2^{16}-1	TR	F

This gain value is multiplied by the integral of current error.

CURRENT OFFSET   INDEX 0X2380, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}$-1	TR	RF

This offset value is added to the commanded motor current. It can be used to compensate for a directional bias affecting the current loop.

CURRENT LOOP GAINS   INDEX 0X60F6

Type	Access	Units	Range	Map PDO	Memory
RECORD	RW	-	-	YES	R

This object is for backward compatibility and is no longer recommended.

Use object 0x2380 which has the same content.

6.6   Gain Scheduling Configuration

The *Gain Scheduling* feature allows you to schedule gain adjustments based on changes to a key parameter. For instance, Pp, Vp, and Vi could be adjusted based on changes to commanded velocity.

Gain adjustments are specified in a Gain Scheduling Table. Each table row contains a key parameter value and the corresponding gain settings. The drive uses linear interpolation to make smooth gain adjustments between the programmed settings.

Gain Scheduling Tables are stored in the Copley Virtual Machine (CVM) memory space. They can be created and modified using CME 2 software.

The following objects are used to configure Gain Scheduling.

GAIN SCHEDULING CONFIG   INDEX 0X2370

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	-	YES	RF

Bits	Meaning		
0-2	Key parameter for gain scheduling.		
	Value	Description	
	0	None. Setting the key parameter to zero disables gain scheduling.	
	1	Use value written to Gain Scheduling Key Parameter (index 0x2371, p. 164) as the key.	
	2	Use Instantaneous Commanded Velocity (index 0x2250, p. 142).	
	3	Use Load Encoder Velocity (index 0x2231, p. 152).	
	4	Use Position Demand Internal Value object (index 0x60FC, p. 147).	
	5	Use Position Actual Value object (index 0x6063, p. 142).	
	6-7	Reserved.	
3-7	Reserved.		
8	If set, use the absolute value of key parameter for gain lookup.		
9	If set, disable gain scheduling until the axis is referenced (homed).		
10-15	Reserved.		
16	Table includes position loop PP if set.		
17	Tables includes position loop VP if set.		
18	Table includes velocity loop VI if set.		
19	Table includes current loop offset if set.		
20	Table includes position loop if PI if set.		
21	Table includes position loop PD if set.		
22-31	Reserved.		

GAIN SCHEDULING KEY PARAMETER   INDEX 0x2371

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-2^{31} to $+2^{31}-1$	TR	R

Gain scheduling key parameter value. When gain scheduling is enabled, the current value of the key parameter is stored here. When this parameter is selected as the key parameter for gain scheduling, then it may be written to manually move through entries in the gain scheduling table.

6.7   Chained Biquad Filters

SECOND CHAINED BIQUAD FILTER   INDEX 0x210A

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[[0..13] of UINT	RW	224	-	NO	RF

Second chained biquad filter on output of velocity loop.

THIRD CHAINED BIQUAD FILTER INDEX 0x210B

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[[0..13] of UINT	RW	224	-	NO	RF

Third chained biquad filter on output of velocity loop.

FIRST CHAINED BIQUAD FILTER   INDEX 0x210C

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[[0..13] of UINT	RW	224	-	NO	RF

First chained biquad filter on input of current loop.

SECOND CHAINED BIQUAD FILTER   INDEX 0x210D

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[[0..13] of UINT	RW	224	-	NO	RF

Second chained biquad filter on input of current loop.

7 STEPPER MODE SUPPORT

7.1 Operation

Copley Controls Drives and Stepper Mode Operation

Copley Controls supports the use of stepper motors over a CANopen network. The Stepnet drive can drive a two-phase stepper motor in stepper or servo mode. The Accelnet and Xenus drives can drive a three-phase stepper motor in stepper mode.

Stepper vs. Servo

In a closed-loop servo system, sensors feedback the actual position and/or velocity of the motor, and the drive calculates how much torque to apply to the motor to move it to the target destination.

An open-loop stepper system does not typically have sensors to feed back actual position or velocity information. Nor does it use the position and velocity loops used in servo systems. Instead, the drive moves the motor in steps by applying fixed current to the motor's windings in measured intervals. Position and velocity commands can be derived but not measured.

Microstepping

The type of stepper motor supported by the Copley Controls Stepnet drive has two windings. It can be driven using the simple full stepping method or the more precise microstepping method. Copley Controls supports microstepping as described in Microstepping (p. 166).

The Accelnet and Xenus drives support three-phase, three-winding stepper motors. The Accelnet and Xenus also use microstepping to drive these three-phase stepper motors.

Microstepping

Copley Controls' microstepping drives provide a much higher degree of control over a motor's position than does a full stepping system. The microstepping drive applies varying amounts of current into both windings of the motor at the same time, making it possible to rest the motor not only at the full step locations, but at points between them, and thus allowing a high degree of control over the motor's position.

In microstepping mode, it is necessary to program the following CANopen objects:

Object	Description
Motor Pole Pairs (Index 0x2383, Sub-Index 2, p. 90)	Number of motor pole pairs (electrical phases) per rotation. For example, for a 1.8 deg/step motor, set Motor Pair Polls to 50.
Microsteps/Rev (Index 0x2383, Sub-Index 29, p. 95)	Microsteps per revolution.

There is virtually no limit on the number of microsteps/rev. Programming a very high value does not mean that the drive can actually move the motor to that many distinct positions, because the ability to control current in the windings is limited. The practical limit depends on the motor, but something on the order of 1000 microsteps/electrical cycle is generally

reasonable. It is sometimes advantageous to program a large number of microsteps, so the system works as expected when connected to a high-resolution encoder.

Some drive manufacturers require that the number of microsteps/rev be an integer multiple of the number of electrical cycles. Copley Controls drives do not have such a limitation.

Current Control in Microstepping Mode

Servo systems use their servo loops to determine how much current (and in which direction) to apply to the motor. For a stepper motor, the amount of current is typically a constant value programmed by the user.

In addition, Copley Controls drives use different current values for different states of motor activity. During constant speed moves, the Run Current is applied.

During the acceleration / deceleration portion of the move, the Boost Current is used. After a move completes (the velocity reaches zero) the drive continues to apply the Run Current to the motor for the amount of time programmed in the Run to Hold Time object. Once that timeout has expired, the Hold Current is applied.

While Boost Current is applied to the motor, an I^2T limit is used to protect the motor from overheating. If the move remains in the acceleration phase for longer than the boost current time, then the current applied to the motor falls back to the run current. This allows the system to set the Run Current value equal to the motor's continuous current limit, and set the Boost Current to a value larger than the motor's continuous limit.

Once the move has finished and the holding current has been applied to the motor, an optional voltage control mode of operation can be entered. In this mode of operation, the motor is held in position with extremely low jitter at the expense of a slightly looser control of the current in the motor's windings. The Voltage Control Mode Time Delay object can be programmed to control the delay between entering hold current mode and entering the voltage control mode.

If the Voltage Control Mode Time Delay is set to zero, the voltage control mode is disabled.

7.2 Stepper Mode Objects

BOOST CURRENT   INDEX 0x2110

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	TR	RF

Functions as boost current in stepper mode and peak current in servo mode. Current used during acceleration and deceleration in stepper mode. Specifies a boost or peak current limit in 0.01-amp units.

RUN CURRENT   INDEX 0x2111

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	TR	RF

Functions as run current in stepper mode and continuous current in servo mode. Output of the current limiter (0.01-amp units). This is the current that the current loop will attempt to apply to the stepper motor during continuous velocity portion of moves.

TIME AT BOOST CURRENT   INDEX 0x2112

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $2^{16}-1$	TR	RF

Functions as time at boost current in stepper mode and time at peak current in servo mode. Specifies the maximum time at boost or peak current. The drive uses this value as an input to an I^2T current limiting algorithm to prevent over stressing the load.

HOLD CURRENT   INDEX 0x21D0

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01 amps	-2^{15} to $+2^{15}-1$	TR	RF

Current used to hold the motor at rest. Used in stepper mode only.

RUN TO HOLD TIME INDEX 0x21D1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $2^{16}-1$	TR	RF

The period of time, beginning when a move is completed, during which the output stays at run current level before switching to hold current level. Used in stepper mode only.

DETENT CORRECTION GAIN FACTOR FOR MICROSTEPPING MODE   INDEX 0x21D2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

Can be used to reduce detent noise.

VOLTAGE CONTROL MODE TIME DELAY   INDEX 0x21D5

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $2^{16}-1$	TR	RF

Time delay from entering hold current before entering the special voltage control mode of operation. This mode trades the normal tight control of current for very low jitter on the motor position. Used in stepper mode only. Set to 0 to disable this feature.

STEPPER CONFIGURATION AND STATUS   INDEX 0x21D6

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $+2^{15}-1$	TR	RF

Bit-mapped as follows:

Bit	Description
0	Use the encoder input for phase compensation if enabled. Pure stepper mode if disabled.
1	Use on outer position loop to adjust the stepper position based on Position Error (index 0x60F4, p. 144). When this bit is set, the gain value Maximum Velocity Adjustment (index 0x21D5, p. 168) is multiplied by the Position Error, and the result is a velocity that is added to the microstepping position.
2-15	Reserved.

PROPORTIONAL GAIN FOR STEPPER OUTER LOOP   INDEX 0x21D7

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	-	0 to $2^{16}-1$	TR	RF

This parameter gives the gain used for calculating a velocity adjustment based on Position Error (index 0x60F4, p. 144). This parameter is only used when the stepper outer loop is engaged, which occurs when bit 1 of Stepper Configuration and Status (index 0x21D6, p. 169) is set.

MAXIMUM VELOCITY ADJUSTMENT   INDEX 0x21D8

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 steps/sec	0 to $2^{32}-1$	TR	RF

This is the maximum velocity adjustment made by the stepper outer position loop when enabled. This parameter is only used when the stepper outer loop is engaged, which occurs when bit 1 of Stepper Configuration and Status (index 0x21D6, p. 169) is set.

8 HOMING MODE OPERATION

8.1 Overview

Homing is the method by which a drive seeks the home position (also called the datum, reference point, or zero point). There are various methods of achieving this using limit switches at the ends of travel, or a dedicated home switch.

Most of the methods also use the index pulse input from an incremental encoder.

The drive performs homing operations in Homing Mode (Mode Of Operation [index 0x6060, p. 67] =6).

The Homing Function

The homing function provides a set of trajectory parameters to the position loop, as shown below. The parameters are generated by the homing function and are not directly accessible through CANopen dictionary objects. They include the profile mode and velocity, acceleration, and deceleration data.

- Home Offset
- Homing Method
- Homing Speeds
- Home Velocity Fast / Slow
- Homing Acceleration
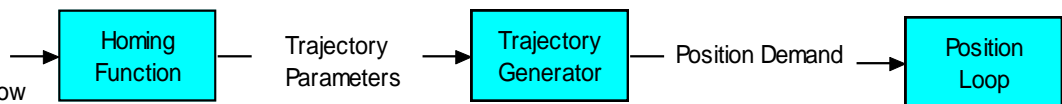
Figure 11—Homing Function

Initiating and Verifying a Homing Sequence

A homing move is started by setting bit 4 of the Control Word object (index 0x6040, p. 61). The results of a homing operation can be accessed in the Status Word (index 0x6041, p. 61).

Home Offset

The *home offset* is the difference between the zero position for the application and the machine home position (found during homing). During homing the home position is found and once the homing is completed the zero position is offset from the home position by adding the Home Offset to the home position. All subsequent absolute moves shall be taken relative to this new zero position. This is illustrated in the following diagram.

Figure 42—Home Offset position

Homing Speeds

There are two homing speeds: fast and slow. The fast speed is used to find the home switch and the slow speed is used to find the index pulse. (See the

Homing Speeds object [index 0x6099, p. 189])

Homing Acceleration

Homing Acceleration (index 0x609A, p. 189) establishes the acceleration to be used for all accelerations and decelerations with the standard homing modes.

Note that in homing, it is not possible to program a separate deceleration rate.

8.2 Homing Methods Overview

There are several homing methods. Each method establishes the:

Home reference (limit or home switch transition or encoder index pulse)

Direction of motion and, where appropriate, the relationship of the index pulse to limit or home switches.

Legend to Homing Method Descriptions

As highlighted in the example below, each homing method diagram shows the starting position on a mechanical stage. The arrow line indicates direction of motion, and the circled H indicates the home position. Solid line stems on the index pulse line indicate index pulse locations. Longer dashed lines overlay these stems as a visual aid. Finally, the relevant limit switch is represented, showing the active and inactive zones and transition.
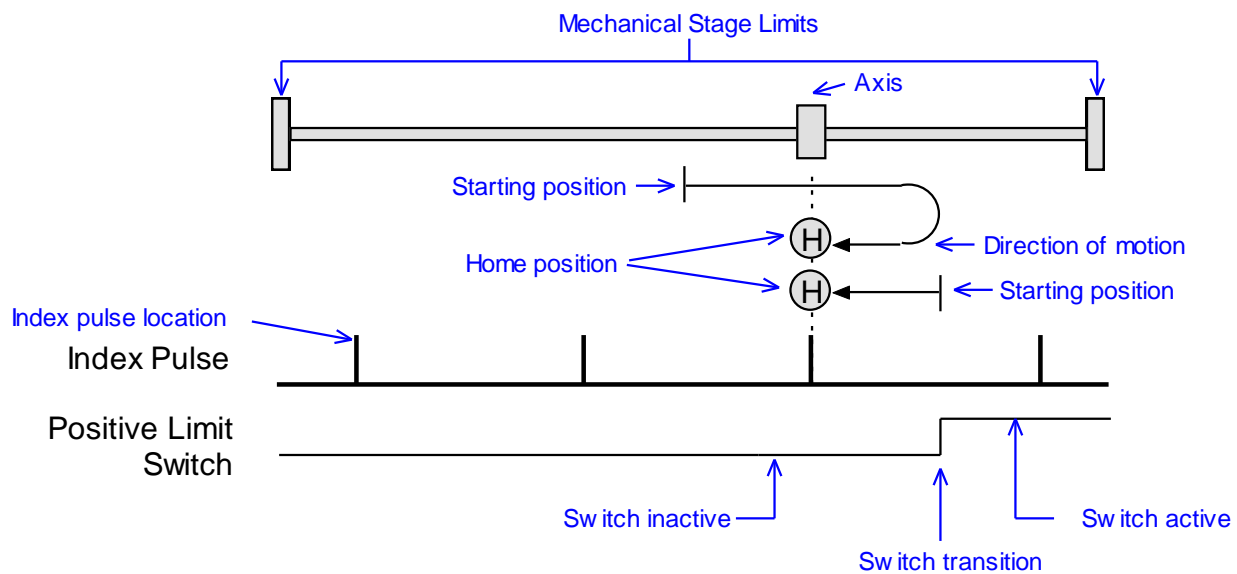
Figure 13—Homing Method Descriptions

Note that in the homing method descriptions, negative motion is leftward and positive motion is rightward.

Home is Current Position

Using this method, home is the current position.

Set Homing Method (index 0x6098, p. 188) to: 0.

Home is Current Position; Move to New Zero

Set current position to home and move to new zero position (including home offset). This is the same as **Error! Reference source not found.** except that mode 0 does not do the final move to the home position.

Set Homing Method (index 0x6098, p. 188) to: 35.

Next Index

Direction of Motion: Positive

Home is the first index pulse found in the positive direction. Direction of motion is positive. If a positive limit switch is activated before the index pulse, an error is generated.
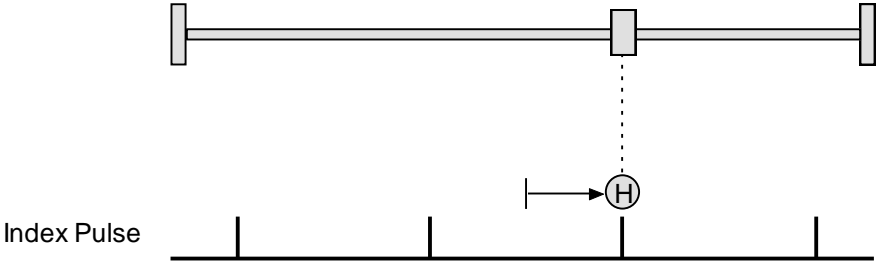
Figure 54—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 34.

Direction of Motion: Negative

Home is the first index pulse found in negative direction. Direction of motion is negative. If a negative limit switch is activated before the index pulse, an error is generated.
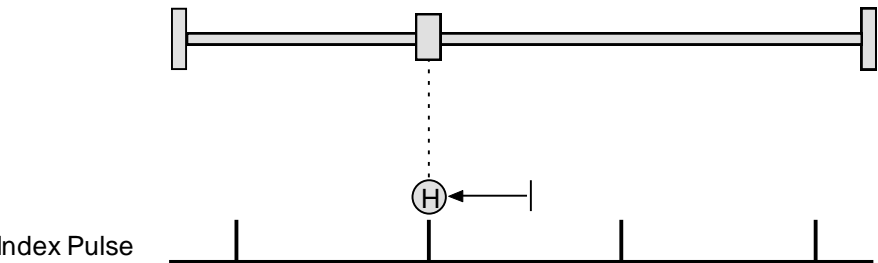
Figure 65—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 33.

Limit Switch

Direction of Motion: Positive

Home is the transition of the positive limit switch. Initial direction of motion is positive if the positive limit switch is inactive.
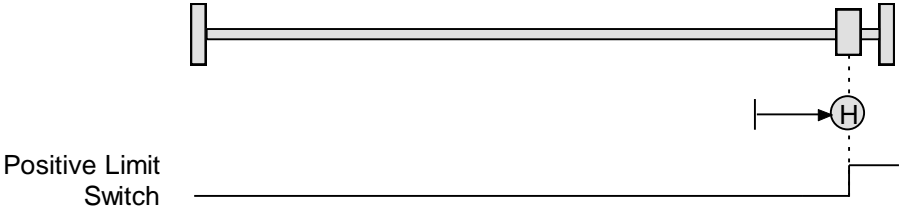
Figure 76—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 18.

Direction of Motion: Negative

Home is the transition of negative limit switch. Initial direction of motion is negative if the negative limit switch is inactive.
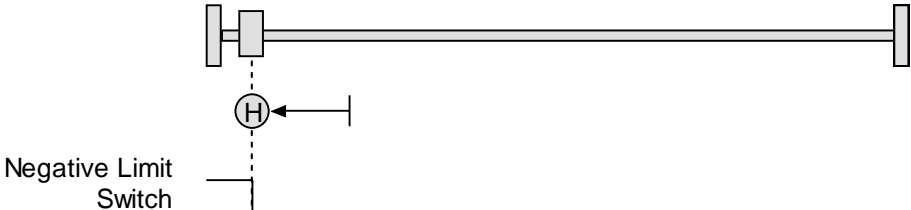
Figure 87—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 17.

Limit Switch Out to Index

Direction of Motion: Positive

Home is the first index pulse to the negative side of the positive limit switch transition. Initial direction of motion is positive if the positive limit switch is inactive (shown here as low).
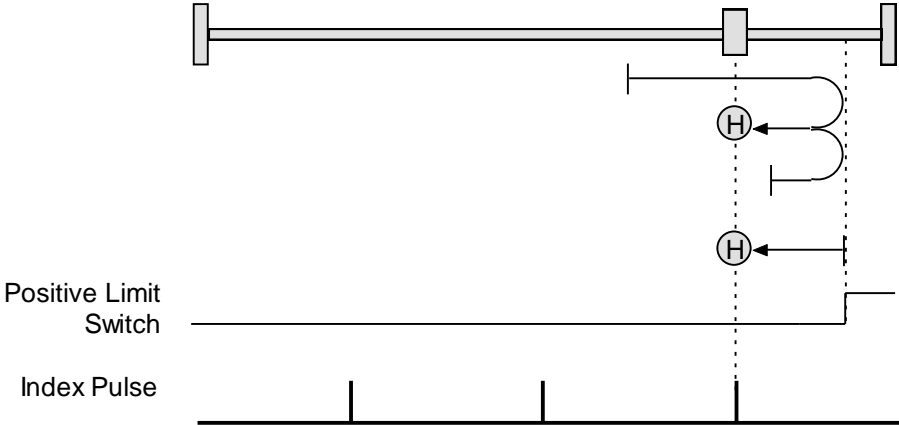
Figure 98—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 2.

Direction of Motion: Negative

Home is the first index pulse to the positive side of the negative limit switch transition. Initial direction of motion is negative if the negative limit switch is inactive (shown here as low).
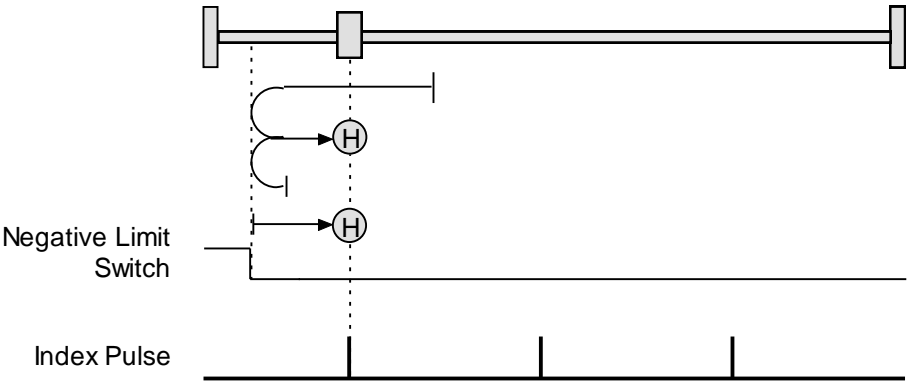
Figure 109—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 1.

Hardstop

Direction of Motion: Positive

Home is the positive hard stop. Direction of motion is positive. The hard stop is reached when the drive outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a positive limit switch is activated before the hard stop, an error is generated.

In stepper drives in stepper mode, the hard stop is reached when the following error exceeds the tracking window.

Figure 20—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: -1.

Direction of Motion: Negative

Home is the negative hard stop. Direction of motion is negative. The hard stop is reached when the drive outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a negative limit switch is activated before the hard stop, an error is generated.
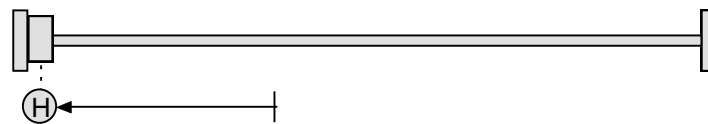
Figure 21—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: -2.

Hardstop Out to Index

Direction of Motion: Positive

Home is the first index pulse on the negative side of the positive hard stop. Initial direction of motion is positive. The hard stop is reached when the drive outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a positive limit switch is activated before the hard stop, an error is generated.

In stepper drives in stepper mode, the hard stop is reached when the following error exceeds the tracking window.
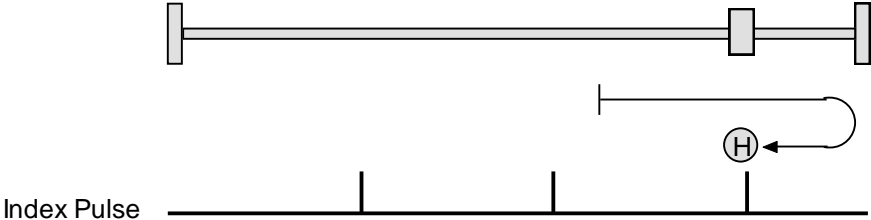
Figure 22—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: -4.

Direction of Motion: Negative

Home is the first index pulse on the positive side of the negative hard stop. Initial direction of motion is negative. The hard stop is reached when the drive outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a negative limit switch is activated before the hard stop, an error is generated.
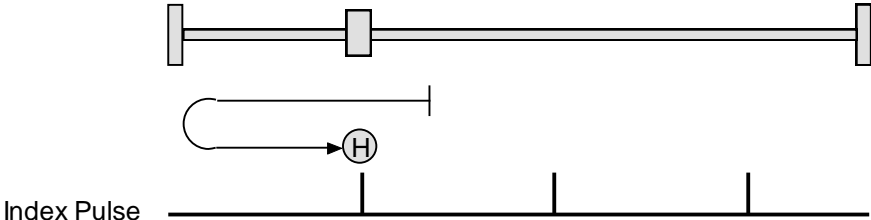
Figure 23—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to:-3.

Home Switch

Direction of Motion: Positive

Home is the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.
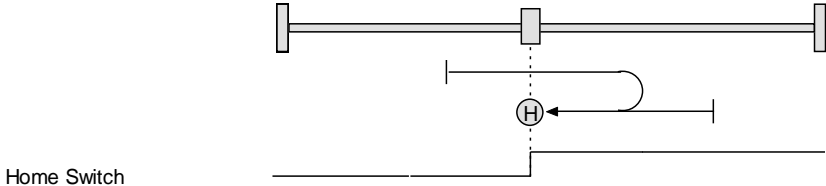
Home Switch

Figure 23—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 19.

Direction of Motion: Negative

Home is the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.
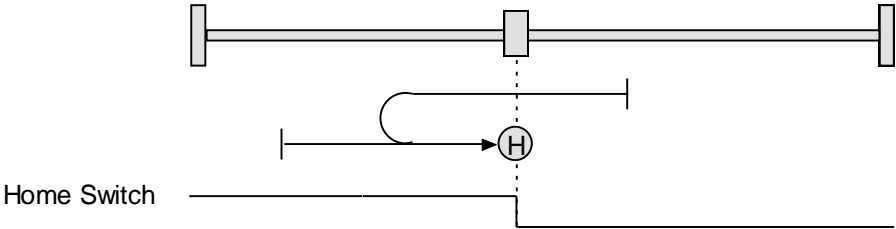
Home Switch

Figure 24—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 21.

Home Switch Out to Index

Direction of Motion: Positive

Home is the first index pulse to the negative side of the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.
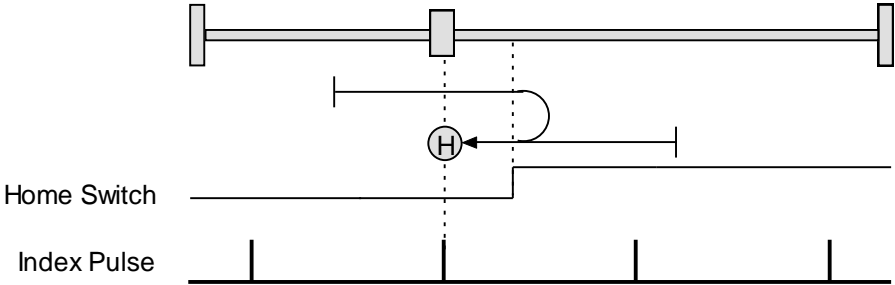
Figure 24—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 3.

Direction of Motion: Negative

Home is the first index pulse to the positive side of the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.
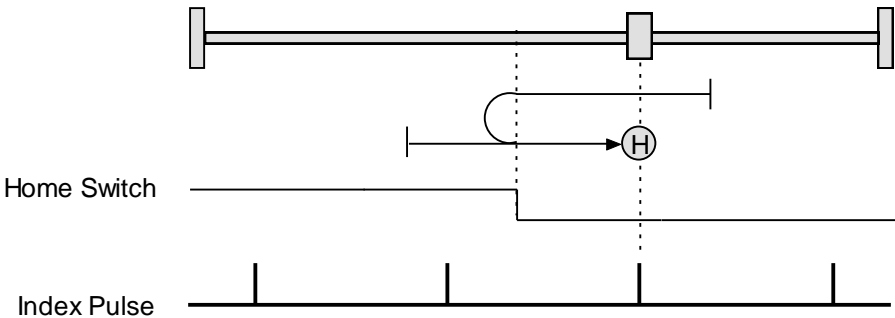
Figure 25—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 5.

Home Switch In to Index

Direction of Motion: Positive

Home is the first index pulse to the positive side of the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.
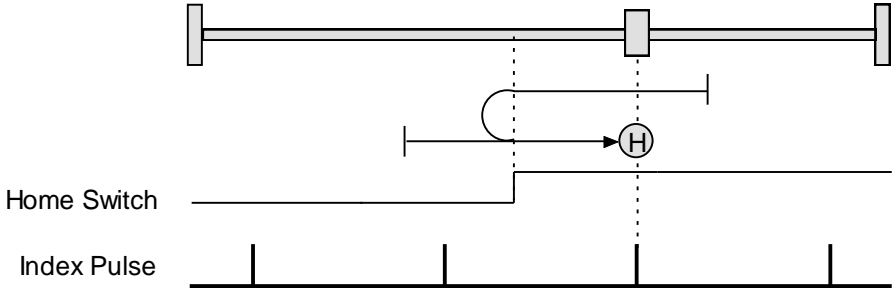
Figure 26—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 4.

Direction of Motion: Negative

Home is the first index pulse to the negative side of the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.
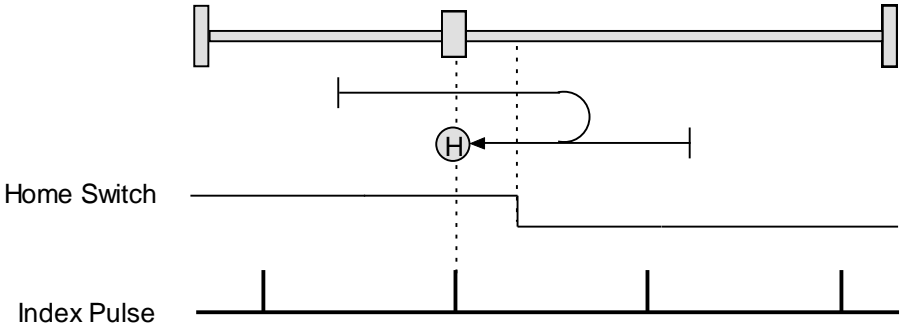
Figure 27—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 6.

Lower Home

Direction of Motion: Positive

Home is the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. Motion will reverse if a positive limit switch is activated before the home switch; then, if a negative limit switch is activated before the home switch, an error is generated.
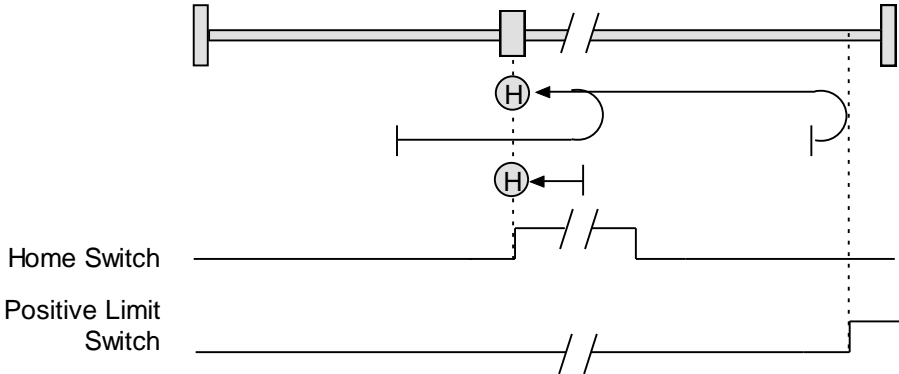
Figure 28—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 23.

Direction of Motion: Negative

Home is the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home switch, an error is generated.
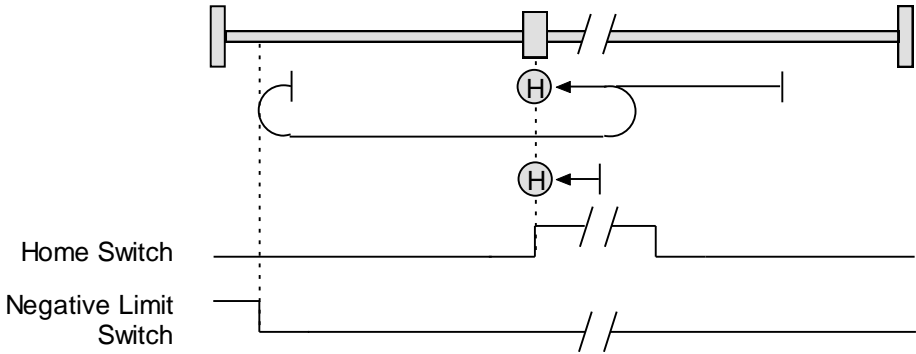
Figure 29—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 29.

Upper Home

Direction of Motion: Positive

Home is the positive edge of a momentary home switch. Initial direction of motion is positive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.
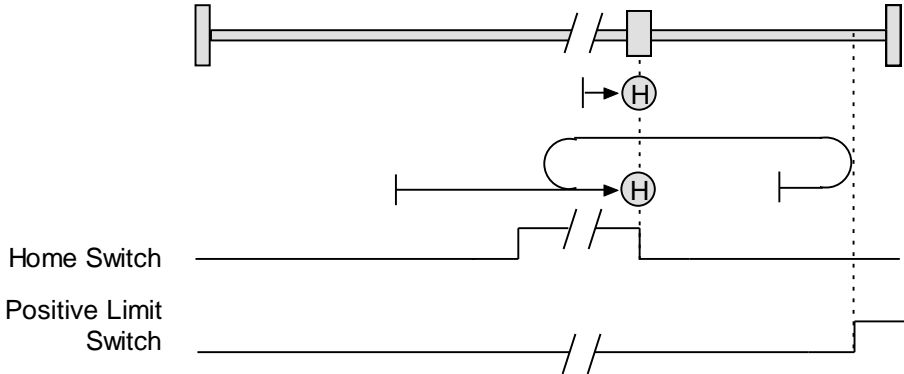
Figure 30—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 25

Direction of Motion: Negative

Home is the positive edge of momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home switch, an error is generated.
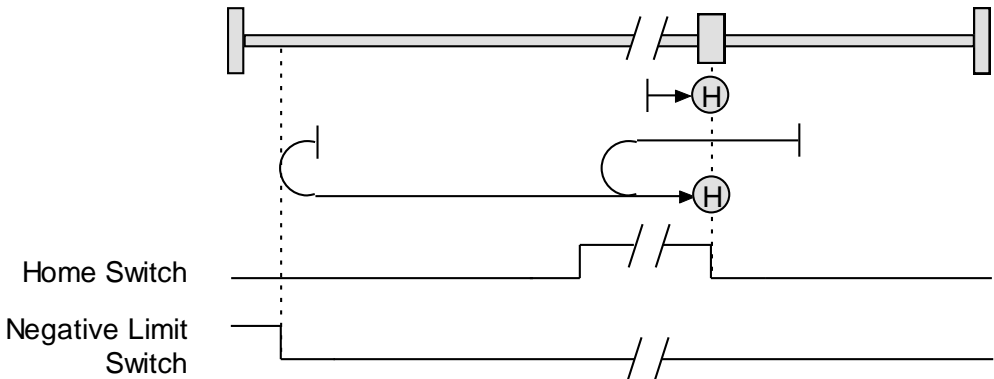
Figure 31—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 27

Lower Home Outside Index

Direction of Motion: Positive

Home is the first index pulse on the negative side of the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.
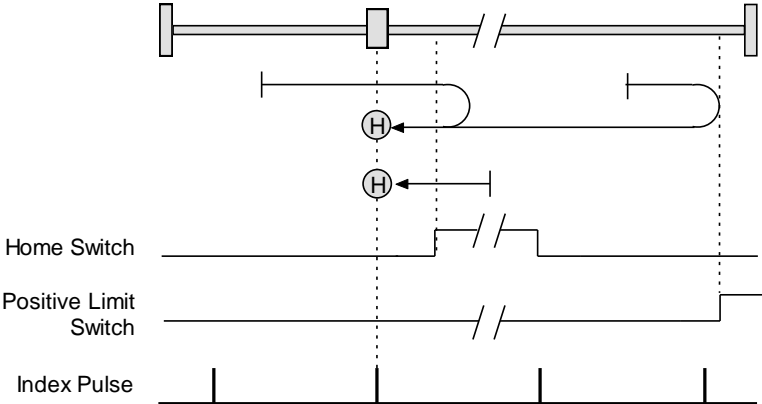
Figure 32—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 7.

Direction of Motion: Negative

Home is the first index pulse on the negative side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.
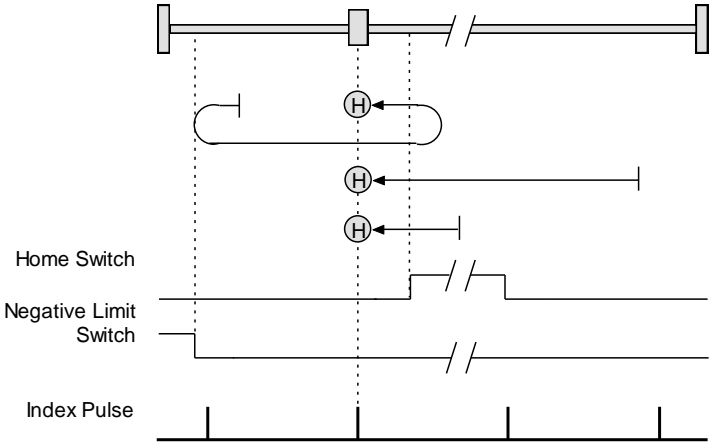
Figure 33—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 14.

Lower Home Inside Index

Direction of Motion: Positive

Home is the first index pulse on the positive side of the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.
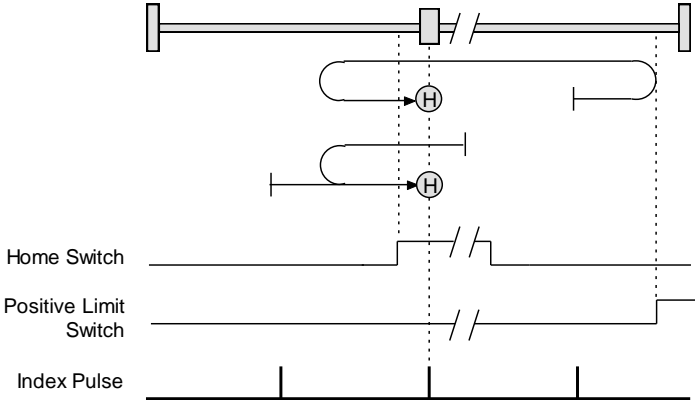
Figure 34—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 8.

Direction of Motion: Negative

Home is the first index pulse on the positive side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.
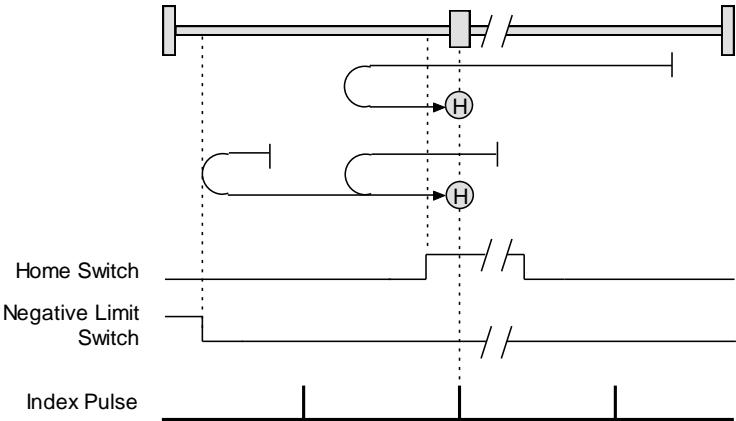
Figure 35—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 13.

Upper Home Outside Index

Direction of Motion: Positive

Home is the first index pulse on the positive side of the positive edge of a momentary home switch. Initial direction of motion is positive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.
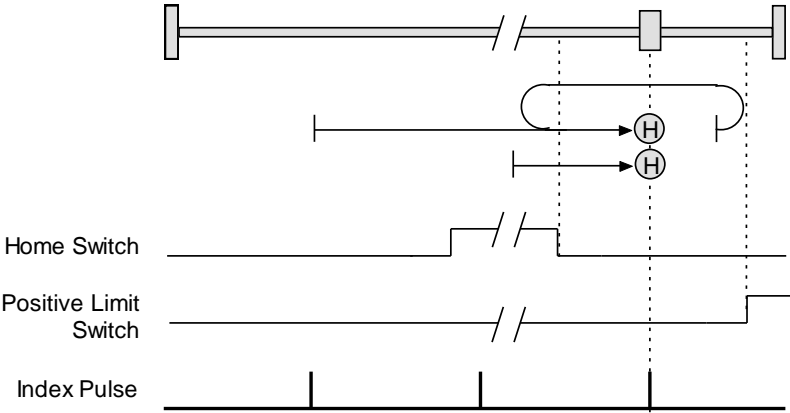
Figure 36—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 10.

Direction of Motion: Negative

Home is the first index pulse on the positive side of the positive edge of a momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial position is right of the home position, the axis reverses on encountering the home switch.
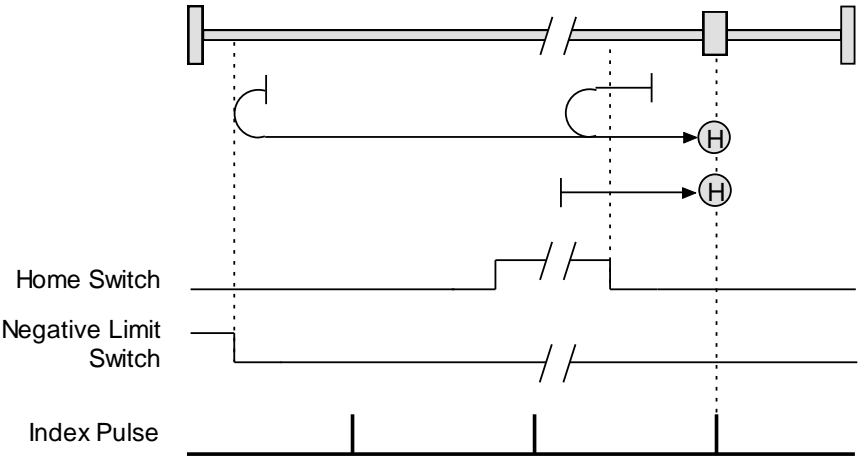
Figure 37—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 11.

Upper Home Inside Index

Direction of Motion: Positive

Home is the first index pulse on the negative side of the positive edge of momentary home switch. Initial direction of motion is positive. If initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.
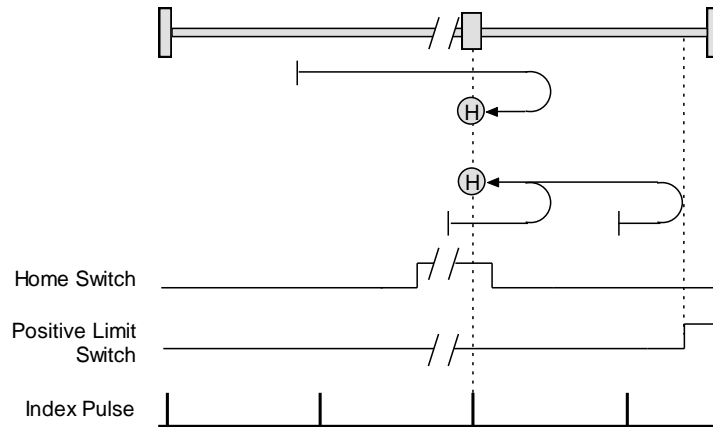
Figure 38—Diagram showing positive motion

Set Homing Method (index 0x6098, p. 188) to: 9.

Direction of Motion: Negative

Home is the first index pulse on the negative side of the positive edge of a momentary home switch. Initial direction of motion is negative if the home switch is inactive. If initial motion leads away from the home switch, the axis reverses on encountering the negative limit; then, if a negative limit switch is activated before the home switch, an error is generated.
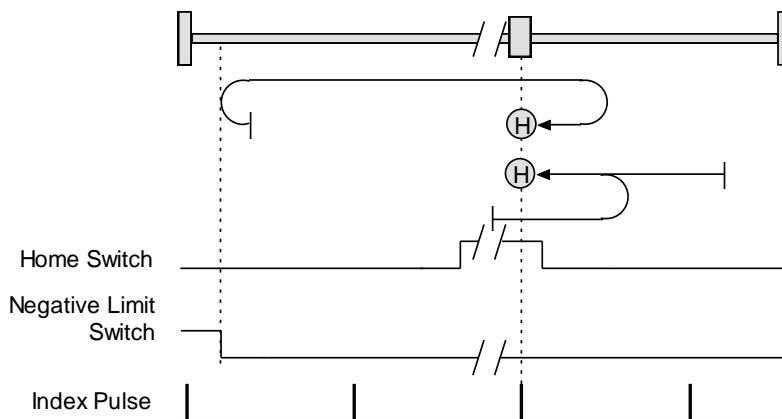
Figure 39—Diagram showing negative motion

Set Homing Method (index 0x6098, p. 188) to: 12.

Immediate Home

Immediate Home with Absolute Encoder

On startup, the drive is immediately homed using the position data from an absolute encoder. This homing method uses Object 0x607C (Home Offset), to apply an offset, if desired.

Set Homing Method (index 0x6098, p. 188) to: 25

Home Configuration Object for Custom Homing Methods

Copley Controls provides an object that provides access to the drive's internal home configuration register. When a standard CANopen homing method is used, the software automatically sets a value in this register.

To specify homing options that are not supported by the standard CANopen methods, the application can directly program this configuration register. This provides finer control of the homing methods then the standard CANopen ones allow.

For example, all of the standard CANopen homing methods will cause a move to the new zero position after it has been found. With a large home offset, this could be a large or slow move. This final move can be avoided by programming the internal home configuration register directly.

8.3   Homing Mode Operation Objects

HOMING METHOD   INDEX 0x6098

Type	Access	Units	Range	Map PDO	Memory
Integer 8	RW	-	See *Description*, below.	YES	RF

The method for finding the motor home position in homing mode. Program a method described below by writing its code to 0x6098. Most of the methods are paired. Each member of a pair uses the same basic method but starts in the opposite direction and has a distinct code. For a full description of any method, see the referenced pages.

Homing Method	Initial Motion	Code	Full Description #
Hardstop Out to Index	Positive	-4	p. 177
	Negative	-3	
Hardstop	Negative	-2	p. 176
	Positive	-1	
Error! Reference source not found.	Any	0	p. **Error! Bookmark not defined.**
Home is Current Position; Move to New Zero	Any	35	p. 172
	Negative	1	p. 175
	Positive	2	
Limit Switch Out to Index			
Error! Reference source not found.	Positive	3	p. **Error! Bookmark not defined.**
	Negative	5	
Home Switch In to Index	Positive	4	p. 180
	Negative	6	
Lower Home Outside Index	Positive	7	p. 183
	Negative	14	
Lower Home Inside Index	Positive	8	p. 184
	Negative	13	
Upper Home Inside Index	Positive	9	p. 186
	Negative	12	
Upper Home Outside Index	Positive	10	p. 185
	Negative	11	
Limit Switch	Negative	17	p. 174
	Positive	18	
	Positive	19	p. 178
	Negative	21	

Homing Method	Initial Motion	Code	Full Description #
Home Switch			
Lower Home	Positive	23	p. 181
	Negative	29	
Upper Home	Positive	25	p. 182
	Negative	27	
Next Index	Positive	34	p. 172
	Negative	33	
Immediate Home	37		
Reserved for future use.	15-16, 20, 22, 24, 26, 28, 30-32		

Note that these homing methods only define the location of the home position. The zero position is always the home position adjusted by the homing offset. See Homing Methods Overview, p.172.

HOMING SPEEDS   INDEX 0x6099

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	80	-	YES	RF

This array holds the two velocities used when homing. Sub-index 0 contains the number of sub-elements of this record.

HOME VELOCITY – FAST   INDEX 0x6099, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts/sec	0 to $2^{32}-1$	TR	RF

This velocity value is used during segments of the homing procedure that may be handled at high speed. Generally, this means move in which the home sensor is being located, but the edge of the sensor is not being found.

User defined units are achievable using the factor group objects.

HOME VELOCITY – SLOW   INDEX 0x6099, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.1 counts/sec	0 to $2^{32}-1$	TR	RF

This velocity value is used for homing segment that require low speed such as cases where the edge of a homing sensor is being sought.

User defined units are achievable using the factor group objects.

HOMING ACCELERATION   INDEX 0x609A

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	10 counts/sec^2	0 to $2^{32}-1$	TR	RF

This value defines the acceleration used for all homing moves. The same acceleration value is used at the beginning and ending of moves (i.e. there is no separate deceleration value).

User defined units are achievable using the factor group objects.

HOME OFFSET   INDEX 0x607C

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-2^{31} to $+2^{31}-1$	TR	RF

The home offset is the difference between the zero position for the application and the machine home position (found during homing). During homing the home position is found. Once the homing is completed the zero position is offset from the home position by adding the Home Offset to the home position. All subsequent absolute moves shall be taken relative to this new zero position.

See Home Offset (p. 170) for more information.

User defined units are achievable using the factor group objects.

HARD STOP MODE HOME CURRENT   INDEX 0x2350

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	0.01A	-2^{15} to $+2^{15}-1$	TR	RF

Home current in hard stop mode, in which the drive drives the motor to the mechanical end of travel (hard stop). End of travel is recognized when the drive outputs the Hard Stop Mode Home Current for the Hard Stop Mode Home Delay time (index 0x2351, p. 190).

Note that the homing current value isn't the current limit that will be used when homing, it's the current threshold. The drive considers the motor to be in a hard stop condition when the actual current exceeds this amount for longer than the homing delay value (index 0x2351). During a home to hard stop move the motor current will be temporarily limited to a value that's 25% higher than this setting.

HARD STOP MODE HOME DELAY   INDEX 0x2351

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	milliseconds	0 to $2^{16}-1$	TR	RF

Delay used for homing to a hard stop mode.

HOME CONFIG   INDEX 0x2352

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	TR	RF

Alternate method for configuring the homing mode. Provides more flexibility than the standard CANopen method does. Bit-mapped as follows:

Bits	Description		
0-3	Home function.		
	Value	Description	
	0	If bit 5 is not set, then just set the current position as home. If bit 5 is set, then move in the direction specified by bit 4 and set the location of the first index pulse as home. Bit 6 is not used in this mode.	
	1	Move in the direction specified by bit 4 until a limit switch is encountered. Then move in the other direction out of limit. If bit 5 is clear, then the edge location is home. If bit 5 is set, then the next index pulse is home. Bit 6 is not used in this mode.	

Bits		Description
	2	Home on a constant home switch. The initial move is made in the direction specified by bit 4. When the home switch is encountered, the direction is reversed. The edge of the home switch is set as home if bit 5 is clear. If bit 5 is set, then an index pulse is used as the home position. Bit 6 is used to define which index pulse is used.
	3	Home on an intermittent home switch. This mode works the same as mode 2 except that if a limit switch is encountered when initially searching for home, then the direction is reversed. In mode 2, hitting a limit switch before finding home would be considered an error. Bit 8 identifies which edge of the home to search for (positive or negative).
	4	Home to a hard stop. This moves in the direction specified in bit 4 until the home current limit is reached. It then presses against the hard stop using that current value until the home delay time expires. If bit 5 (index) is set, drive away from the hard stop until an index is found.
4		Initial move direction (0=positive, 1=negative).
5		Home on index pulse if set.
6		Selects which index pulse to use. If set, use the pulse on the DIR side of the sensor edge. DIR is the direction specified by bit 4 of this word.
7		If set, capture falling edge of index. Capture rising edge if clear.
8		When using a momentary home switch, this bit identifies which edge of the home switch to reference on. If set, then the negative edge is used; if clear the positive edge is used.
9		If set, make a move to the zero position when homing is finished. If clear, the zero position is found, but not moved to.
10		If set, the homing sequence will run as normal, but the actual position will not be adjusted at the end. Note that even though the actual position is not adjusted, the Homing Adjustment (index 0x2353, p. 196) is updated with the size of the adjustment (in counts) that would have been made. Also, if bit 10 is set then no move to zero is made regardless of the setting of bit 9.
11		If this bit is set, at the end of the home routine the home configuration stored in flash will be set to 15, and the home offset stored in flash will be updated to the correct value necessary to calibrate an absolute encoder based on the most recent home operation. This bit is used to automate the calibration of absolute encoders.

8.4   Position Capture

POSITION CAPTURE CONTROL REGISTER INDEX 0x2400

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	See *Description*, below.	TR	RF

Sets up position capture features for the encoder index home switch input and high-speed position capture input. Bit-mapped as follows:

Bit	Description
0	If set, the Captured Index Position (index 0x2402) is captured on the rising edge of the index.
1	If set, the Captured Index Position is captured on the falling edge of the index.
2	If set, a Captured Index Position value will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.
3,4	Reserved.
5	If set, a Home Capture Position (Index 0x2403) is captured on the active > inactive edge of the home switch input. If clear, the position is captured on the inactive > active edge of the input.
6	If set, a Home Capture Position will not be overwritten by a new position until it has been read. If clear, a new position will overwrite the old position.
8	If set, enable high speed input position capture. The position value is written to Position of Last HIGH-SPEED Motor Capture (index 0x2405).
10	If set, a Position of Last HIGH-SPEED Motor Capture value will be latched.
12	If set, clear Position Actual Value (index 0x6063) on every encoder index pulse.

POSITION CAPTURE STATUS REGISTER   INDEX 0x2401

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	-	See *Description*, below.	T	R

Shows the current status of the index or home switch capture mechanism. Bit-mapped as follows:

Bit	Description
0	If set, a new Captured Index Position (p. 179) has been captured. Cleared when the captured position is read.
1,2	Reserved
3	If set, a new Captured Index Position (p. 179) transition occurred when a captured position was already stored. Depending on the setting of bit 2 of 0x2400, the new position may have been discarded, or has overwritten the previously stored position.
4	If set, new Home Capture Position (p. 179) has been captured. This bit is cleared when the captured position is read.
5,6	Reserved
7	If set, a new home switch input transition occurred when a Home Capture Position was already stored. Depending on the setting of bit 6 of 0x2400, the new position has been discarded, or has overwritten the previously stored position.
8	If set, a new high-speed input position has been captured. Cleared when the captured position is read.
10	Reserved
11	If set, a new high speed input transition occurred when a Position of Last HIGH-SPEED Motor Capture was already stored. Depending on the setting of bit 10 of 0x2400, the new position has been discarded, or has overwritten the previously stored position.

CAPTURED INDEX POSITION   INDEX 0x2402

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

Reading this variable resets bits 0 & 3 of the Position Capture Status Register (index 0x2401, p. 192). Provides the position that the axis was in when an index pulse was captured. Configured by setting bits in the Position Capture Control Register (index 0x2400, p. 187), and the status of the captured data can be checked in the Position Capture Status Register. Reading this variable resets bits 0 & 3 of the Position Capture Status Register.

HOME CAPTURE POSITION INDEX 0x2403

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

Provides the position that the axis was in when an input pin configured as a home switch input became active. This function can be configured by setting bits in the Position Capture Control Register (index 0x2400, p. 187), and the status of the captured data can be checked in Position Capture Status Register (index 0x2401, p. 187).

TIME STAMP OF LAST HIGH-SPEED POSITION CAPTURE   INDEX 0x2404

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	microseconds	-2^{31} to $+2^{31}-1$	TR	R

Provides the time when an input pin configured as a high-speed capture input became active (and the axis position was captured).

POSITION OF LAST HIGH-SPEED MOTOR CAPTURE   INDEX 0x2405

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	Counts	-2^{31} to $+2^{31}-1$	T	R

Provides the position that the axis was in when an input pin configured as a high-speed capture input became active.

POSITION OF LAST HIGH-SPEED LOAD CAPTURE   INDEX 0x2406

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

Provides the position that the axis was in when an input pin configured as a high-speed capture input became active.

INPUT CAPTURE CONTROL INDEX 0x2408

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	136	-	NO	R

Provides the position that the axis was in when an input pin configured as a high-speed capture input became active. Sub-index 0 holds the number of elements in this object.

INPUT 1~26 CAPTURE CONTROL   INDEX 2408 SUB-INDEX 1~26

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	-	0 to 2^8-1	NO	R

Configures the capture settings for IN1~IN26.

INPUT CAPTURE STATUS   INDEX 0X2409

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	136	-	TR	R

Provides the status of capture from IN1~IN26.
Sub-index 0 holds the number of elements in this object.

INPUT 1~26 CAPTURE STATUS   INDEX 0X2409 SUB-INDEX 1~26

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	-	0 to 2^8-1	TR	R

Reports the status of the capture for IN1~IN26.

CAPTURED RISING EDGE POSITION   INDEX 0X240A

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	496	-	T	R

Position at the time of capture for IN1~IN26.
Sub-index 0 holds the number of elements in this object.

INPUT 1~26 CAPTURED RISING EDGE POSITION   INDEX 0X240A SUB-INDEX 1~26

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-2^{31} to $+2^{31}$-1	T	R

Each sub-index is the position at the time of capture for the input of the sub-index number.

CAPTURED FALLING EDGE POSITION   INDEX 0X240B

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	496	-	T	R

Position at the time of capture for IN1~IN26.
Sub-index 0 holds the number of elements in this object.

INPUT 1~26 CAPTURED FALLING EDGE POSITION   INDEX 0X240B SUB-INDEX 1~26

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	-	-2^{31} to $+2^{31}$-1	T	R

Each sub-index is the position at the time of capture for the input of the sub-index number.

CAPTURED RISING EDGE TIME   INDEX 0X240C

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	976	-	EVENT	R

The time at the rising edge of the capture for IN1~IN26.
Sub-index 0 holds the number of elements in this object.

INPUT 1~26 CAPTURED RISING EDGE TIME INDEX 0X240C SUB-INDEX 1~26

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED64	RW	-	0 to $+2^{64}$-1	T	R

Each sub-index is the time at rising edge of the capture for the input of the sub-index number.

CAPTURED FALLING EDGE TIME   INDEX 0x240D

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	976	-	T	R

The time at the falling edge of the capture for IN1~IN26.
Sub-index 0 holds the number of elements in this object.

INPUT 1~26 CAPTURED RISING EDGE TIME INDEX 0x240D SUB-INDEX 1~26

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED64	RW	-	0 to $+2^{64}-1$	T	R

Each sub-index is the time at the falling edge of the capture for the input of the sub-index number.

CAPTURED EDGE COUNTS   INDEX 0x240E

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RO	8	-	NO	R

The time at the falling edge of the capture for IN1~IN26. Sub-index 0 holds the number of elements in this object.

INPUT 1~26 CAPTURED EDGE COUNTS   INDEX 0x240E SUB-INDEX 1~26

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	counts	0 to $2^{16}-1$	T	R

Each sub-index is the time at the falling edge of the capture for the input of the sub-index number.

CME2 SOFTWARE USE   INDEX 0x2421

Type	Access	Bits	Range	Map PDO	Memory
STRING(40)	RW	320	-		F

Reserved for use by CME2 software.

TIME OF LAST POSITION SAMPLE INDEX 0x2410

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	ms	0 to $2^{32}-1$	T	R

This object provides a 32-bit time stamp, in microseconds, that corresponds to the time that the encoder was most recently sampled. When this object is read, the position is latched and saved into object 0x2411. If mapping this object and 0x2411 to a PDO, object 0x2410 should be mapped first so the objects match data from one reading.

POSITION DURING TIME SAMPLE READING   INDEX 0x2411

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This object provides the position read the corresponds to the time of the last position sampled, object 0x2410. Object 0x2410 must be read for a new position to be latched.

HOMING ADJUSTMENT   INDEX 0X2353

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This parameter is updated after each successful homing operation. The value it contains is the size of the actual position adjustment made in the last home sequenc

9 TOUCH PROBE OPERATION

TOUCH PROBE FUNCTION   INDEX 0x60B8

Type	Access	Units		Range	Map PDO	Memory
UNSIGNED16	RW	-		See Description, below.	TR	R

This object sets up the function of the touch probes.

Bit	Value	Definition
0	0	Switch off touch probe 1
	1	Enable touch probe 1
1	0	Trigger first event (latching)
	1	Continuous
2,3	0	Use input pin 1 as first touch probe input
	1	Use encoder index as first touch probe
	2	Use value programmed in 0x60D0 sub-index 1 to specify first touch probe input
	3	Reserved
4	0	Switch off sampling at positive edge of touch probe 1
	1	Enable sampling at positive edge of touch probe 1
5	0	Switch off sampling at negative edge of touch probe 1
	1	Enable sampling at negative edge of touch probe 1
6,7	Reserved	Reserved
8	0	Switch off touch probe 2
	1	Enable touch probe 2
9	0	Trigger first event (latching)
	1	Continuous
10,11	0	Use input pin 2
	1	Use encoder index as second touch probe
	2	Use value programmed in 0x60D0 sub-index 2 to specify second touch probe input
	3	Reserved
12	0	Switch off sampling at positive edge of touch probe 2
	1	Enable sampling at positive edge of touch, probe 2
13	0	Switch off sampling at negative edge of touch probe 2
	1	Enabled sampling at negative edge of touch probe 2
14,15	Reserved	Reserved

TOUCH PROBE STATUS   INDEX 0x60B9

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	-	See *Description*, below.	T	R

This object sets up the function of the touch probes.

Bit	Value	Definition
0	0	Touch probe 1 is switched off
	1	Touch probe 1 is enabled
1	0	Touch probe 1 no positive edge value stored
	1	Touch probe 1 positive edge value stored
2	0	Touch probe 1 no negative edge value stored
	1	Touch probe 1 negative edge value stored
3-7	Reserved	Reserved
8	0	Touch probe 2 is switched off
	1	Touch probe 2 is enabled
9	0	Touch probe 2 no positive edge value stored
	1	Touch probe 2 positive edge value stored
10	0	Touch probe 2 no negative edge value stored
	1	Touch probe 2 negative edge value stored
11-15	Reserved	Reserved

TOUCH PROBE POS1 POS VALUE   INDEX 0x60BA

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This object provides the position value of the touch probe 1 at the positive edge. User defined units are achievable using the factor group objects.

TOUCH PROBE POS1 NEG VALUE   INDEX 0x60BB

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This object provides the position value of the touch probe 1 at the negative edge. User defined units are achievable using the factor group objects.

TOUCH PROBE POS2 POS VALUE   INDEX 0x60BC

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This object provides the position value of the touch probe 2 at the positive edge. User defined units are achievable using the factor group objects.

TOUCH PROBE POS2 NEG VALUE   INDEX 0x60BD

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	T	R

This object provides the position value of the touch probe 2 at the negative edge. User defined units are achievable using the factor group objects.

TOUCH PROBE SELECT   INDEX 0x60D0

Type	Access	Bits	Range	Map PDO	Memory
RECORD	RW	48	See *Description*, below.	NO	R

This object is used to select the inputs that are used for the touch probes. Accepted manufacturer specific values -1 for pin 1, -2 for pin 2, etc. This array holds the two values.
Sub-index 0 contains the number of sub-elements for this record.

TOUCH PROBE SELECT – PROBE 1   INDEX 0x60D0, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $+2^{15}-1$	NO	R

TOUCH PROBE SELECT – PROBE 2   INDEX 0x60D0, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	-2^{15} to $+2^{15}-1$	NO	R

TOUCH PROBE TIME 1 POS VALUE   INDEX 0x60D1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	ns	0 to $2^{32}-1$	T	R

This object provides the time value of the touch probe 1 at the positive edge.

TOUCH PROBE TIME 1 NEG VALUE   INDEX 0x60D2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	ns	0 to $2^{32}-1$	T	R

This object provides the time value of the touch probe 1 at the negative edge.

TOUCH PROBE TIME 2 POS VALUE   INDEX 0x60D3

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	ns	0 to $2^{32}-1$	T	R

This object provides the time value of the touch probe 1 at the positive edge.

TOUCH PROBE TIME 2 NEG VALUE   INDEX 0x60D4

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	ns	0 to $2^{32}-1$	T	R

This object provides the time value of the touch probe 1 at the negative edge.

10 PROFILE POSITION, VELOCITY, TORQUE & FACTOR GROUP OPERATION

10.1 Operation

Point-to-Point Motion Profiles

In profile position mode, an drive receives set points from the trajectory generator to define a target position and moves the axis to that position at a specified velocity and acceleration. This is known as a point-to-point move.

The drive performs profile position moves in Profile Position Mode (Mode Of Operation [index 0x6060, p. 67] =1).

Jerk

In a point-to-point move, the rate of change in acceleration is known as jerk. In some applications, high rates of jerk can cause excessive mechanical wear or material damage.

Trapezoidal and S-curve Motion Profiles

To support varying levels of jerk tolerance, the profile position mode supports two motion profiles: the trapezoidal profile, which has unlimited jerk, and the jerk-limited S-curve (sinusoidal) profile.
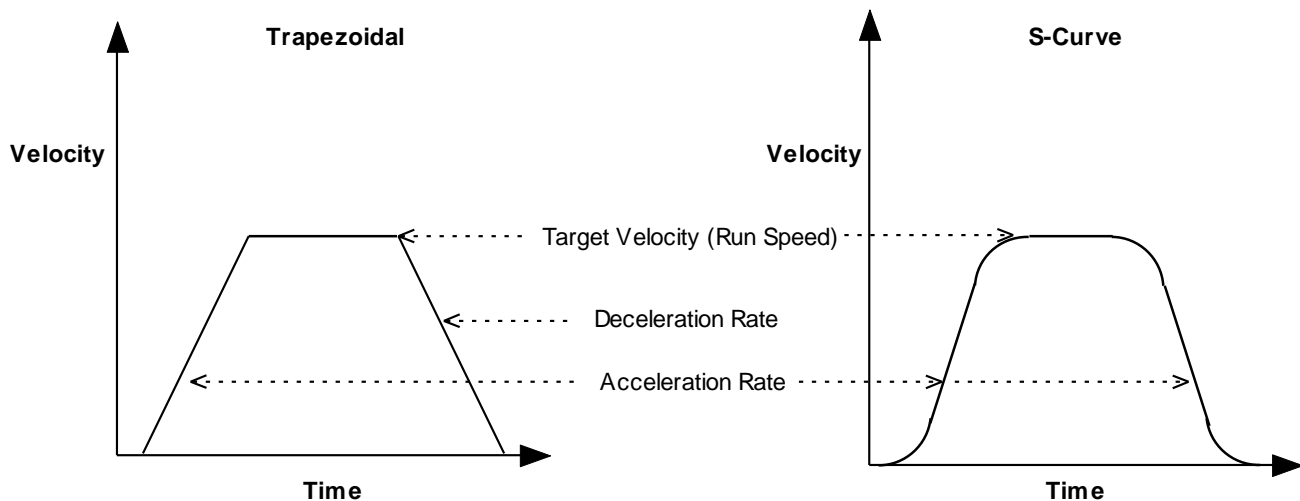
Figure 40—Trapezoidal and S-curve Motion Profiles

In a trapezoidal profile, jerk is unlimited at the corners of the profile (start of the move, when the target velocity is reached, when deceleration begins, and at the end of the move). S-curve profiling limits jerk or "smooths" the motion.

Note that an S-curve profile move does not support an independent deceleration rate. Instead, the acceleration rate is applied to both the acceleration and deceleration of the move. Further, trapezoidal and profile position special velocity mode profiles support changing of the parameters of the current move, whereas an S-curve profile does not. This difference is discussed in Handling a Series of Point-to-Point Moves

The Motion Profile Type object (index 0x6086) controls which type of profile is used.

For guidance in choosing a trapezoidal or S-curve profile, read the following sections and then refer to Trapezoidal vs. S-Curve Profile.

(Copley Controls CANopen drives also support a profile position special velocity mode. This 7obeys the acceleration, deceleration, and velocity limits, but continues to move as though the target position were infinite.)

Relative vs. Absolute Moves

In a relative move, the target position is added to the instantaneous commanded position, and the result is the destination of the move. In an absolute move, the target position is offset from the home position.

Handling a Series of Point-to-Point Moves

There are two methods for handling a series of point-to-point moves:

As a series of discrete profiles (supported in both trapezoidal and S-curve profile moves)

As one continuous profile (supported in trapezoidal profile moves only)

General descriptions of the two methods follow. Detailed procedures and examples appear later in the chapter.

A Series of Discrete Profiles

The simplest way to handle a series of point-to-point moves is to start a move to a particular position, wait for the move to finish, and then start the next move. As shown below, each move is discrete. The motor accelerates, runs at target velocity, and then decelerates to zero before the next move begins.
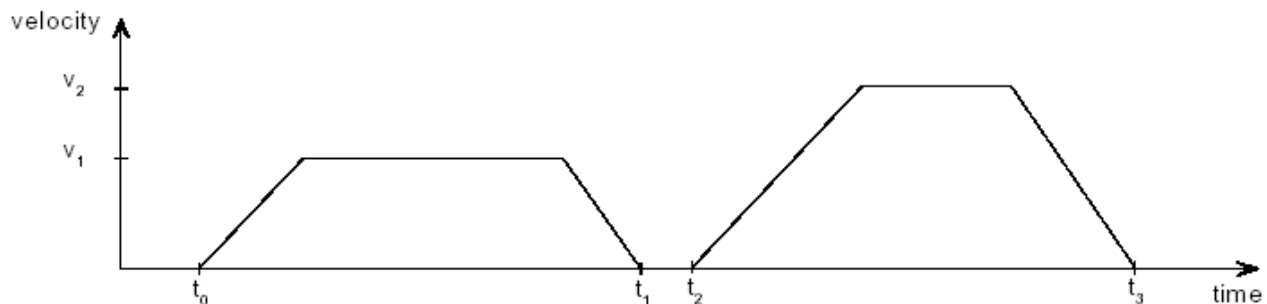
Figure 41—Trapezoidal Discrete Motion Profiles

The *CANopen Profile for Drives and Motion Control (DSP 402)* refers to this method as the "single set point" method.

Copley Controls CANopen drives allow use of this method with both trapezoidal and S-curve profile moves.

One Continuous Profile

Alternately, a series of trapezoidal profile moves can be treated as a continuous move. As shown below, the motor does not stop between moves. Instead, the move parameters (target position, velocity, acceleration, and deceleration) are updated immediately at the end of the previous move (when bit 4 of the Control Word is set, as described later in this section).
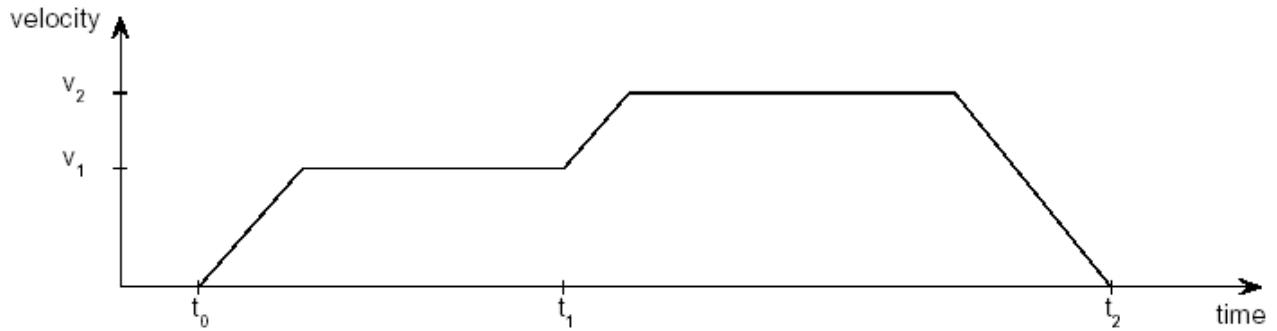
Figure 42—Trapezoidal Continuous Motion Profile

The *CANopen Profile for Drives and Motion Control (DSP 402)* refers to this method as the "set of set points" method.

Copley Controls CANopen drives allow use of this method with trapezoidal profile moves only.

Set of set-points

When a drive is in trapezoidal profile position mode, a feature called *set of set-points* is available. This enables the drive to keep an additional set-point in a buffer, which is executed when the currently executing move is finished. The figure below shows how the feature is used, and how corresponding bits in the status word and control word are used.
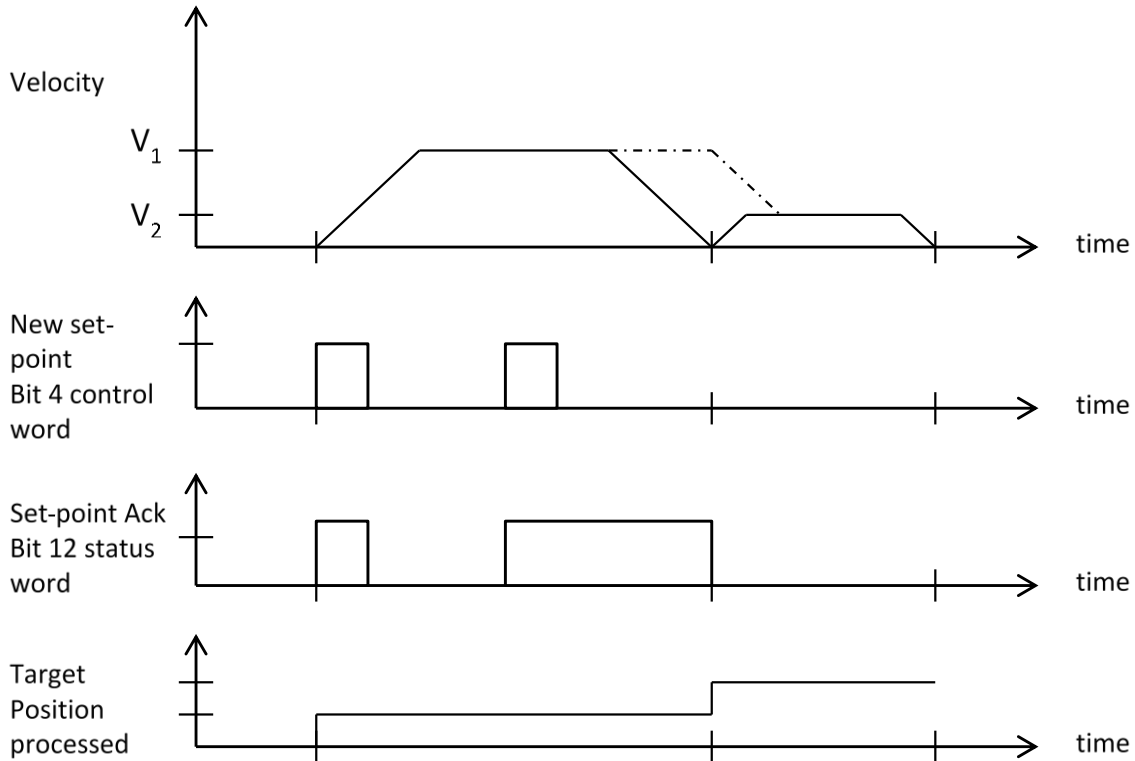
Figure 43—Set of set points

The dashed line in the first graph represents the velocity if the *change of set point bit* (bit 9 of the control word) is set.

Using the end velocity object may be helpful to achieve functionality as show in the graph below.
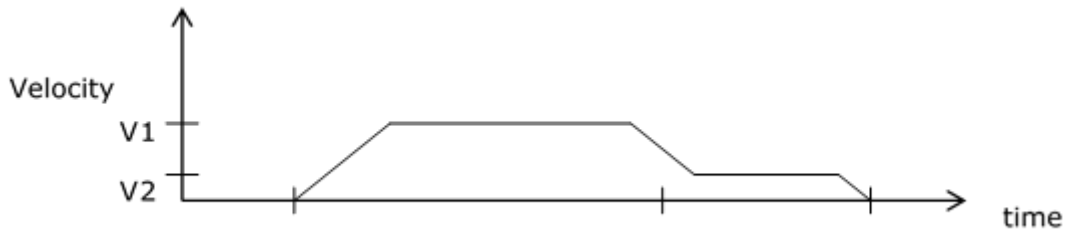
Figure 44—Graph showing funciotnality of end velocity object

Overview of Point-to-Point Move Parameters and Related Data

Move Parameters

Each point-to-point move is controlled by a set of parameters, accessed through the following objects.

Object Name/ID	Description	Page #
Trajectory Jerk Limit / 0x2121	Maximum rate of change of acceleration. Used with S-curve profiles only.	211
Target Position / 0x607A	When running in position profile mode, this object holds the destination position of the trajectory generator. Note that for profile position special velocity mode profiles, the target position only specifies the direction of motion, not a true position.	207
Profile Velocity / 0x6081	Velocity that the trajectory generator will attempt to achieve when running in position profile mode.	157
Profile Acceleration / 0x6083	Acceleration that the trajectory generator attempts to achieve when running in position profile mode.	157
Profile Deceleration / 0x6084	Note that an S-curve profile move does not use a deceleration rate. Instead, the acceleration rate is applied to both the acceleration and deceleration of the move.	157
Quick Stop Deceleration / 0x6085	Deceleration value used when a trajectory needs to be stopped as the result of a quick stop command. Note that unlike most trajectory configuration values, this value is NOT buffered. Therefore, if the value of this object is updated during an abort, the new value is used immediately.	158
Motion Profile Type / 0x6086	Trapezoidal, S-curve, or special velocity mode.	158

The Point-to-Point Move Buffer

In profile position mode, the drive uses a buffer to store the parameters (listed in Move Parameters, above) for the next point-to-point move, or for a modification of the current trapezoidal profile move. The move buffer can be modified at any point before a control sequence (described in following sections) copies the "next-move" parameters to the active move registers.

Move-Related Control Word and Status Word Bit Settings

An drive's Control Word (index 0x6040) and Status Word (index 0x6041) play an important role in the initiation and control of point-to-point move sequences, as described below.

Object Name / Index	Bit #	Bit Name	Description/Comments
Control Word / 0x6040	4	new set point	The transition of bit 4 from 0 to 1 is what causes the drive to copy a set of move parameters from the buffer to the active register, thus starting the next move.
	5	change set immediately	Allows or prevents attempt to perform a series of moves as one continuous profile (change move parameters while move is in progress). Value = 0: Drive will ignore a 0 to 1 transition on bit 4 if there is currently a move in progress. Value = 1 and Motion Profile Type (index 0x6086, p. 158) = trapezoidal or velocity mode: Allow new move to begin immediately after bit 4 low-to-high transition. Value = 1 and Motion Profile Type is S-curve: Ignore update and continue move with old parameters.
	6	absolute/relative	Value = 0: Move is absolute (based on home position). Value = 1: Move is relative (based on current commanded position).
	8	halt	Value = 1: Interrupts the motion of the drive. Wait for release to continue.
Status Word / 0x6041	10	target reached	Drive sets bit 10 to 1 when target position has been reached. Drive clears bit 10 to zero when new target is received. If quick stop option code (p. 65) is 5, 6, 7, or 8, this bit is set when the quick stop operation is finished and the drive is halted. Bit 10 is also set when a Halt occurs.
	12	set point acknowledge	Set by the drive when Control Word bit 4 goes from 0 to 1. Cleared when Control Word bit 4 is cleared. An invalid transition on Control Word bit 4 will not cause this bit to be set. Invalid transitions include those made while drive is in motion and in S-curve mode, or made while drive in motion with Control Word bit 5 not set.

Point-To-Point Move Sequence Examples

Overview

The following diagram illustrates how to implement a series of moves as a series of discrete profiles or as one continuous profile.
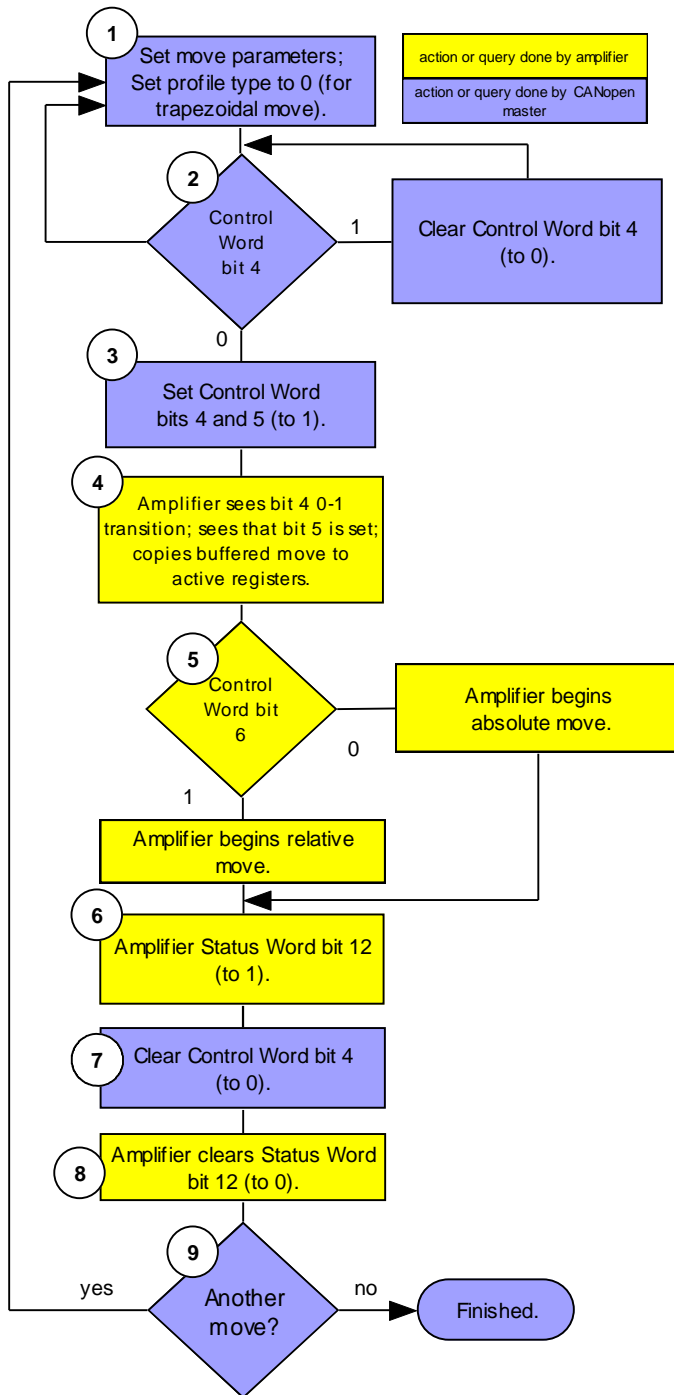
Notes:

1. Move Parameters are described on page 204. This type of move is only supported as a trapezoidal profile.

2. Control Word bit 4 is "new set point." It needs to be 0 because the move will be triggered by a 0->1 transition.

3. Bit 4, value of 1 indicates that valid data has been sent to amplifier and new move should begin.

Bit 5 is "change set immediately." A value of 1 tells the amplifier to update the current profile immediately by copying the contents of the move buffer to the active registers (without waiting for move to finish).

4. Amplifier must detect bit 4 0-1 transition to begin move. Bit 5 value 1 allows immediate update.

5. Control word bit 6: value 0 causes absolute move; value 1 causes relative move.

6. Status Word bit 13 is "set point acknowledge." A value of 1 indicates the amplifier has received a set point and has started the move.

7. Control Word bit 4 is "new set point." It needs to be 0 to allow the next move will be triggered by a 0->1 transition. Also, the 1->0 transition causes the amplifier to clear bit 13.

8. Amplifier detects 0->1 transition of Control Word bit 4 and clears bit 13 in response.

When the motor reaches the target position, the amplifier sets Status Word bit 10 ("target reached") to 1.

9. CANopen master returns to step 1 if there are more moves to complete; otherwise, the series of moves is finished.

Figure 45—Point-to-point diagram

Trapezoidal vs. S-Curve Profile

Difference Between Trapezoidal and S-Curve Profiles

The table below provides a comparison of the differences between trajectory and S-curve profiles, along with including some design considerations indicated by those differences:

Trapezoidal Profile	S-Curve Profile	Design Considerations
Unlimited jerk, operation not as smooth.	Limited jerk, smoother operation.	If the application cannot tolerate jerk, use S-curve. If the application can tolerate jerk, other features available exclusively in trapezoidal profile may indicate its use.
Supports separate acceleration and deceleration rates.	Does not support separate deceleration rate; uses acceleration rate for acceleration and deceleration.	If a separate deceleration rate is critical, the trapezoidal profile is indicated.
Supports modification of current move parameters during current move, allowing the execution of a series of moves as a continuous profile.	Does not support modification of current move. A series of moves requires a series of discrete profiles.	If current move modification is critical, the trapezoidal profile is indicated.
Generally, requires less torque than the S-curve profile to complete an equal move in equal time.	Generally, requires more torque than a trapezoidal profile to complete an equal move in equal time, to make up for time sacrificed for gentler starts and stops.	Designers switching a profile from trapezoidal to S-curve or lowering the value of Trajectory Jerk Limit (index 0x2121, p. 211) might notice some slowing. A higher Profile Acceleration can be applied to compensate, but watch out for drive and motor limits.

TARGET POSITION   INDEX 0x607A

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	counts	-	YES	RF

When running in position profile mode, this object defines the destination of the trajectory generator. The object's meaning varies with the move type, as set in Motion Profile Type (index 0x6086, p. 158):

Move Type	Meaning
Relative	Move distance.
Absolute	Target position.
Velocity	Direction: 1 for positive, -1 for negative.

Note that the target position programmed here is not passed to the internal trajectory generator until the move has been started or updated using the Control Word. See Operation, p. 200, for more information.

10.2 Profile Velocity Mode Operation

Position and Velocity Loops

In profile velocity mode, both the velocity and position loops are used to reach the velocity programmed in the Target Velocity object (index 0x60FF, p. 192). Profile velocity moves are controlled by some of the same gains and limits objects used in profile position mode.

The drive performs profile velocity moves in Profile Velocity Mode (Mode Of Operation [index 0x6060, p. 67] =3).

Stepper Motor Support

The profile velocity mode can be used with a stepper motor.

Encoder Used as Velocity Sensor

The actual velocity is not measured with a velocity sensor. It is derived using position feedback from the encoder.

Starting and Stopping Profile Velocity Moves

As in Profile Position (and Interpolated Position) modes, motion is started by a low-to-high transition of bit 4 of the Control Word (index 0x6040, p. 61). Motion is stopped by a low-to-high transition of bit 8, the Halt bit.

Profile Velocity Mode vs. Profile Position Special Velocity Mode

Profile Position Special Velocity Mode

As described earlier, the profile position mode supports a special velocity mode, in which the velocity trajectory generator takes the place of the trapezoidal generator. The two generators are identical with the exception that in the velocity trajectory generator, the Target Position object (index 0x607A, p. 207) indicates direction, not a target position. Any positive number (including zero) gives positive motion and any negative number gives negative motion. In this special velocity mode, the move continues at the Profile Velocity (index 0x6081, p. 157) until a new target velocity is set or until the move is halted.

To start a move in this mode, program all the profile parameters (trajectory mode, profile velocity, acceleration, deceleration, and direction) and then program a 0-to-1 transition on Control Word bit 4. You can then clear bit 4 without effecting the trajectory, modify any of the parameters (direction, velocity, acceleration, etc.), and set Control Word bit 4 (with bit 5 set also) to update the profile. The normal way to stop motion in this mode is to set a profile velocity of 0.

Profile Velocity Mode

In profile velocity mode, the target velocity is updated as soon as the Target Velocity object (index 0x60FF, p. 156) is set.

In this mode, Control Word bits 4, 5, and 6 are not used.

To start a move in profile velocity mode, set the profile parameters (profile acceleration, profile deceleration, and target velocity). The drive will generate a move as long as the halt bit (Control Word bit 8) is not set. If the halt bit is set, the drive will stop the move using the deceleration value.

10.3 Profile Torque Mode Operation

Current Loop

In profile torque mode, the current loop is used to reach the torque programmed in the Target Torque object (index 0x6071, p.209). When the drive is enabled, or the torque command is changed, the motor torque ramps to the new value at the rate programmed in Torque Slope (index 0x6087, p. 210). When the drive is halted, the torque ramps down at the same rate.

Profile torque moves are controlled by Current Loop Gains (index 0x2380, p. 161).

The drive performs profile torque moves in Profile Torque Mode (Mode Of Operation [index 0x6060, p. 67] =4).

Notes:

1: The profile torque mode cannot be used with a stepper motor.

2: To convert torque commands to the current commands that actually drive the motor, the drive performs calculations based on the motor's Motor Torque Constant (Index 0x2383, Sub-Index 12, p. 92) and Motor Continuous Torque (Index 0x2383, Sub-Index 14

Starting and Stopping Profile Torque Moves

To start a move in profile torque mode, set the profile parameters. The drive will generate a move as long as the halt bit (Control Word bit 8) is not set. If the halt bit is set, the drive will stop the move using the torque_slope value.

TARGET TORQUE   INDEX 0x6071

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	rated torque/1000	-2^{15} to $+2^{15}-1$	TR	R

In profile torque mode, this object is an input to the drive's internal trajectory generator. Any change to the target torque triggers an immediate update to the trajectory generator.

MAX TORQUE INDEX 0x6072

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	rated torque/1000	0 to $2^{16}-1$	T	R

The max torque the drive will exert.

MAX CURRENT   INDEX 0x6073

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	rated current/1000	0 to $2^{16}-1$	T	R

This is the maximum torque-creating current permissible in the motor.

TORQUE DEMAND   INDEX 0x6074

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	rated torque/1000	-2^{15} to $+2^{15}-1$	T	R

Output value of the trajectory generator.

MOTOR RATED CURRENT   INDEX 0x6075

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.001A	0 to $2^{32}-1$	NO	R

The motor's rated current (see motor name plate or motor documentation.)

MOTOR RATED TORQUE   INDEX 0x6076

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	0.001 N-m	0 to $2^{32}-1$	NO	R

Motor's rated torque (see motor name plate or documentation).

TORQUE ACTUAL VALUE   INDEX 0x6077

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	Rated Torque/1000	-2^{15} to $+2^{15}-1$	T	R

Instantaneous torque in the motor.

CURRENT ACTUAL VALUE   INDEX 0x6078

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	Rated Current/1000	-2^{15} to $+2^{15}-1$	T	R

Instantaneous current in the motor.

TORQUE SLOPE   INDEX 0x6087

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Rated Torque/1000 /second	0 to $2^{32}-1$	TR	R

Torque acceleration or deceleration. Set to zero to disable slope limiting for instant response.

TORQUE PROFILE TYPE   INDEX 0x6088

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	0	TR	R

Type of torque profile used to perform a torque change. Set to zero to select trapezoidal profile. No other types are supported.

POSITIVE TORQUE LIMIT   INDEX 0x60E0

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Rated Torque / 1000	0 to $2^{16}-1$	R	R

Motor maximum torque limit in the positive rotation or movement direction.

NEGATIVE TORQUE LIMIT   INDEX 0x60E1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Rated Torque / 1000	0 to $2^{16}-1$	R	R

Motor maximum torque limit in the negative rotation or movement direction.

10.4 Profile Mode Objects

TRAJECTORY JERK LIMIT   INDEX 0x2121

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	100 counts / sec^3	0 to $2^{32}-1$	TR	RF

This object defines the maximum jerk (rate of change of acceleration) for use with
S-curve profile moves. Other profile types do not use the jerk limit.

TRAJECTORY GENERATOR DESTINATION POSITION   INDEX 0x2122

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RO	counts	-2^{31} to $+2^{31}-1$	TR	R

The position that the trajectory generator uses as its destination. Mostly useful when driving the drive using the pulse & direction inputs.

JERK – TRAJECTORY ABORT   INDEX 0x2123

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	100 counts/sec^3	0 to $2^{32}-1$	NO	RF

Jerk value to use during trajectory aborts. If this is zero, then the abort will be calculated without any jerk limits.

TRAJECTORY GENERATOR STATUS INDEX 0x2252

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	Bits	See *Description*, below.	T	R

This variable gives status information about the trajectory generator. It is bit-mapped as follows:

Bit	Description
0-8	Reserved.
9	Cam table underflow.
10	Reserved.
11	Homing error. If set an error occurred in the last home attempt. Cleared by a home command.
12	Referenced. Set if a homing command has been successfully executed. Cleared by a home command.
13	Homing. Set when the drive is running a home command.
14	Set when a move is aborted. This bit is cleared at the start of the next move.
15	In motion bit. If set, the trajectory generator is presently generating a profile.

10.5 Factor Group Objects

Note: *position actual value = position internal value x feed constants*

position encoder resolution x gear ratio

POSITION ENCODER RESOLUTION INDEX 0x608F

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	80		NO	R

This array holds the two values. Sub-index 0 contains the number of sub-elements of this record. Position encoder resolution = encoder increments / motor revolutions. Typical use is with geared rotary motors, or lead-screw linear systems where the position encoder is the load encoder.

Sub-index 0 indicates the highest sub-index supported.

POSITION ENCODER RESOLUTION – ENCODER INCREMENTS INDEX 0x608F, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	counts	0 to $2^{32}-1$	NO	R

The number of encoder counts.

POSITION ENCODER RESOLUTION – MOTOR REVOLUTIONS INDEX 0x608F, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	R

The number of revolutions.

GEAR RATIO INDEX 0x6091

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	80	0 to $2^{32}-1$	NO	R

This array holds the two values. Gear ratio = motor shaft revolutions / driving shaft revolutions
Sub-index 0 contains the number of sub-elements of this record.

RATIO – MOTOR REVOLUTIONS INDEX 0x6091, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Revolutions	0 to $2^{32}-1$	NO	R

Gear box input (motor) shaft revolutions for numerator of gear-ratio fraction.

GEAR RATIO – SHAFT REVOLUTIONS INDEX 0x6091, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	Revolutions	0 to $2^{32}-1$	NO	R

Gear box output (drive) shaft revolutions for denominator of gear-ratio fraction.

FEED CONSTANT INDEX 0x6092

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	80	RECORD	NO	R

The measurement distance for one revolution of the gear box output shaft.
Feed constant = Feed / driving Shaft Revolutions.
This array holds the two values. Sub-index 0 contains the number of sub-elements of this record.

FEED CONSTANT – FEED   INDEX 0x6092, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	R

The measurement distance in user-defined units.

FEED CONSTANT – SHAFT REVOLUTIONS   INDEX 0x6092, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	R

The number of shaft revolutions for the measured distance.

VELOCITY FACTOR   INDEX 0x6096

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	80	RECORD	NO	R

The velocity factor is used to match the velocity units to user defined velocity units.

Velocity Factor = Velocity Units / User Defined Velocity Units.

This array holds the two values. Sub-index 0 contains the number of sub-elements of this record.

Velocity Factor - Numerator Index 0x6096, Sub-Index 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	R

VELOCITY FACTOR - DIVISOR INDEX 0x6096, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	R

ACCELERATION FACTOR   INDEX 0x6097

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	80	RECORD	NO	R

The acceleration factor is used to scale acceleration units in the drive. This array holds the two values. Sub-index 0 contains the number of sub-elements of this record.

NUMERATOR   INDEX 0x6097, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	R

DIVISOR   INDEX 0x6097, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	-	0 to $2^{32}-1$	NO	R

JERK FACTOR INDEX 0x60A2

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	80	RECORD	NO	R

The jerk factor is used to match the jerk units to the defined jerk units. This array holds the two values. Sub-index 0 contains the number of sub-elements of this record.

JERK FACTOR NUMERATOR   INDEX 0x60A2, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	100 Count / s^3	0 to $2^{32}-1$	NO	R

Jerk Units.

JERK FACTOR DIVISOR   INDEX 0x60A2, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RW	100 Count / s^3	0 to $2^{32}-1$	NO	R

Defined Jerk Units.

POLARITY   INDEX 0x607E

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	-	See description	R	R

This object indicates if the position or velocity demand shall be multiplied by 1 or by -1. This polarity is only used in specific modes, including Profile Position, CSP, Profile Velocity and CSV. If value = 0, then multiply by 1. If value = 1 then multiply by -1.

11 INTERPOLATED POSITION OPERATION

11.1 Overview

Coordinated Motion

Interpolated position mode is used to control multiple coordinated axes or a single axis with the need for time-interpolation of set point data. In interpolated position mode, the trajectory is calculated by the CANopen master and passed to the drive's interpolated position buffer as a set of points. The drive reads the points from the buffer and performs linear or cubic interpolation between them.

Copley Controls CANopen drives support three interpolation sub-modes: linear interpolation with constant time, linear interpolation with variable time, and cubic polynomial interpolation, which is also known as position, velocity, and time (PVT) interpolation. The drive can switch between linear and PVT interpolation on the fly.

Linear Interpolation with a Constant Time

In this mode, trajectory position points are assumed to be spaced at a fixed time interval. The drive drives the axis smoothly between two points within the fixed time.

Linear Interpolation with Variable Time

In this linear interpolation mode, each trajectory segment can have a different time interval.

Cubic Polynomial (PVT) Interpolation

In PVT mode, the CANopen master describes the trajectory points as a position, velocity, and time until the next point.

Given two such points, the drive can interpolate smoothly between them by calculating a cubic polynomial function and evaluating it repeatedly until the next point is encountered.

Cubic polynomial interpolation produces much smoother curves than linear interpolation. Thus, it can describe a complex profile with many fewer reference points. This allows a profile to be compressed into a small number of reference points which can be sent over the CAN bus using only a small amount of its total bandwidth.

Standard and Copley Custom Objects for Interpolated Position Mode

Copley Controls CANopen drives provide two sets of objects for performing IP moves:

The CANopen DSP-402 profile standard IP move objects: 0x60C0, 0x60C1, and 0x60C2.

The Copley Controls alternative objects for PVT and linear interpolation with variable time:
0x 2010, 0x 2011, 0x 2012, and 0x 2013. These objects use bandwidth in a more efficient manner and feature an integrity counter to identify lost packets.

CANopen Standard IP Move Objects

When the CANopen DSP-402 profile standard IP move objects are used, the interpolation submode is chosen by setting a code in Interpolation Submode Select (index 0x60C0 p. 222) as described here:

IP Submode	Description
0	Linear interpolation with a constant time. The drive will adjust the constant times of segments to prevent buffer over/underflows. The adjustment can be as much as one servo cycle. This lowers burden of master to maintain accurate constant timings. But interpolated segment can be slightly different than intended due to the adjustments in segment lengths.
-1	Linear interpolation with constant time. Master monitors 0x2012 (buffer status) to maintain points in buffer without overflowing
-2	Linear interpolation with variable times for segments.
-3	Cubic Polynomial (PVT) Interpolation

Linear Interpolation with a Constant Time

In IP submode 0, the trajectory target position of each segment is written to Interpolation Position (index 0x60C1, Sub-Index 1. Each time Interpolation Position is written to, the entire record is written to the drive's internal buffers. (In mode 0, Sub-Index 2 and Sub-Index 3 are ignored).

The time interval is set in Interpolation Time Value (index 0x60C2, Sub-Index 1, p. 224).

Linear Interpolation with Variable Time

In IP submode -1, each trajectory segment can have a different time interval. The trajectory target position of each segment is written to Interpolation Position , which is Sub-Index 1 of the Interpolation Data Record (index 0x60C1, p. 223). With each update to Interpolation Time (index 0x60C1, Sub-Index 2, p. 223), the entire record is written to the drive's internal buffers.
(In mode -1, Sub-Index 3 is ignored.)

Cubic Polynomial (PVT) Interpolation

In IP submode -2, the trajectory target position of each segment is written to Interpolation Position (index 0x60C1, Sub-Index 1, p. 223) and the segment time is written to Interpolation Time (index 0x60C1, Sub-Index 2). When the segment velocity is written to Interpolation Velocity (index 0x60C1, Sub-Index 3, p. 223), the entire record is written to the drive's internal buffers.

Copley Controls Alternative Objects for IP Moves

The Copley Controls alternative objects use bandwidth in a highly efficient manner. They also feature an integrity counter to identify lost packets.

Each profile segment is packed into a single 8-byte object in the object dictionary (IP move segment command, index 0x2010, p. 220). If a PDO is used to transmit the object, then a segment may be transmitted in a single CAN message. For a PVT example, see **Error! R eference source not found.**. PVT Profile Moves Using the Copley Controls Alternative Objects.

Interpolated Position Trajectory Buffer

A typical profile contains a large number of segments. These segments must be passed to the drive over the CANopen network quickly enough to ensure that the next point is received before the drive needs it to calculate the intermediate motor positions.

To reduce the tight timing requirements of sending trajectory segments over the network, the drive maintains a buffer of trajectory segments in its memory. This allows the controller to send trajectory segments in bursts, rather than one at a time, as the profile is executing. The drive can hold 32 trajectory segments. See the Trajectory Buffer Free Count object (index 0x2011, p. 221).

Guidelines for Buffer Use

The drive needs a minimum of 2 trajectory segments to perform interpolation. Thus, a successful move requires at least two segments in the buffer. Generally, it is best to keep the buffer at least one step ahead of the interpolation, so it is best to keep at least three segments in the buffer at any time during a move.

For instance, suppose a PVT trajectory includes the three segments:

P0, V0, T0
P1, V1, T1
P2, V2, T2

While the move is between the points P0 and P1, the drive needs access to both of these segments to do the interpolation. When that segment is finished (at point P1) the drive needs the next segment in order to continue interpolating toward point P2.

So, between P0 and P1, the drive does not yet need P2. At P1, the drive no longer needs P0 but does need P2 to continue. Strictly speaking, there is no time when the drive needs all three segments at once. However, in practice it is best to make sure that P2 is available when the move is getting close to it.

Starting an Interpolated Position Move

An interpolated position move is started using Control Word settings (index 0x6040, p. 61) and Status Word settings (index 0x6041, p. 61) settings. The transition of Control Word bit 4 from 0 to 1 causes the drive to start the move using the points stored in the interpolated move trajectory buffer. For an example, see **Error! Reference source not found.** (below) and

Format of Data Bytes in PVT Segment Mode.

Ending an Interpolated Position Move

Interpolated position moves can be stopped by adding a zero-time value to the buffer. This method allows the drive to reach the present set point before motion stops.

When using the CANopen standard interpolation objects, send the zero-time value using one the methods described below.

IP Submode	Description	Method
0	Linear interpolation with a constant time.	Send a zero value to Interpolation Time Value (index 0x60C2, Sub-Index 1, p. 224) before sending a segment to the buffer.
-1	Linear interpolation with variable time.	Send a zero in Interpolation Time (index 0x60C1, Sub-Index 2, p. 223).
-2	PVT move using standard CANopen objects.	

Sending a segment with a zero-time value is the recommended way to end an interpolation profile that uses the Copley Controls alternate objects. See IP move segment command object (index 0x2010, p. 220), and Format of Data Bytes in PVT Segment Mode.

An Interpolated position move can also be ended in one of several other ways:

- Clear bit 4 of the Control Word (index 0x6040, p. 61).

- Clear the quick stop bit (bit 2) of the Control Word.

- Set the halt bit (bit 8) of the Control Word.

- Stop adding segments to the buffer. This will cause a buffer underflow, stopping interpolation.

Note that each of these methods stops motion immediately, even if the axis has not reached the set point.

Synchronization

An drive can run in synchronized mode or asynchronous mode. Synchronized mode should be used when doing multi-axis interpolated position moves. (See PDO Transmission Modes, p. 24, and SYNC and High-resolution Time Stamp Messages.)

PVT Profile Moves Using the Copley Controls Alternative Objects

As mentioned earlier, Copley Controls CANopen drives provide an alternate set of objects for more efficient execution of PVT moves and linear interpolation moves with variable time.

The basic method for sending PVT profile data over the CANopen network is:

1. Configure a transmit PDO to send out the Trajectory Buffer Status object (index 0x2012, p. 222). The preferred transmit type for this PDO is 255 (event driven). This causes the PDO to be transmitted every time a segment is read from the buffer, or on error.

2. Configure a receive PDO to receive the PVT buffer data via the IP move segment command (index 0x2010, p. 220).

3. Use either PDO or SDO transfers to fill the PVT buffer with the first N points of the profile (where N is the size of the PVT buffer).

4. If using synchronization, start synchronization before starting motion.

5. Start the move by causing a 0-to-1 transition of bit 4 of the Control Word object (index 0x6040, p. 61).

6. Each time a new Trajectory Buffer Status object (index 0x2012, p. 222) is received, first check for error bits. If no errors have occurred, then one or more additional segments of PVT data should be transmitted (until the trajectory has finished).

   If the Trajectory Buffer Status object indicates that an error has occurred, then the reaction of the controller will depend on the type of error:

   Underflow errors indicate that the master controller is not able to keep up with the trajectory information. When an drive detects a buffer underflow condition while executing an interpolated profile, it will immediately abort the profile. In this case, using longer times between segments is advisable.

   Overflow errors indicate an error in the CANopen master software.

   Segment sequencing errors suggest either an error in the CANopen master software or a lost message, possibly due to noise on the bus. Since the next segment identifier value is passed with the PVT status object, it should be possible to resend the missing segments starting with the next expected segment. Note that the sequencing error code must be cleared with the appropriate IP move segment command Buffer Command Mode message (p. 220) before any new segments of PVT data are accepted.

7. End the move by setting the PVT segment time to zero. See IP move segment command object (index 0x2010, p. 220), and Format of Data Bytes in PVT Segment Mode.

11.2 Interpolated Position Mode Objects

IP MOVE SEGMENT COMMAND   INDEX 0x2010

Type	Access	Bytes	Range	Map PDO	Memory
UNSIGNED64	RW	8	RECORD	R	R

Overview

This object is used to send PVT segment data and buffer control commands in interpolated position mode. This object is write only.

Byte 1: Header Byte

The first byte of the object identifies the type of information contained in the rest of the message. Among other things, it determines whether the PVT Segment Command object operates in a PVT buffer command mode or carries a PVT profile segment.

Buffer Command Mode

If the most significant bit of the header byte is set to 1, then the PVT segment command object is a PVT buffer command. In this case, the command code is located in the remaining 7 bits of the header byte and should take one of the following codes:

Code	Description
0	Clear the buffer and abort any move in progress.
1	Pop the N most recently sent segments off the buffer. PVT profiles will continue to run as long as the buffer doesn't underflow. The number of segments to pop (N) is passed in the next byte (byte 1 of the message). If there are less than N segments on the buffer, this acts the same as a buffer clear except that the profile is not stopped except by underflow.
2	Clear buffer errors. The next byte of data gives a mask of the errors to be cleared (any set bit clears the corresponding error). Error bit locations are the same as the top byte of the status value.
3	Reset the segment ID code to zero.
4	No operation. Used with EtherCAT

PVT Segment Mode

If the most significant bit of the first byte of the message is a zero, then the message contains a segment of the PVT profile. The remaining bits of this first byte contain the following values:

Bits	Description
0-2	Segment integrity counter. This three-bit value increases for each segment sent and is used by the drive to identify missing profile segments. More details of the use of this value are provided below.
3-6	These bits hold a buffer format code. This code identifies how the PVT data is packed into the remaining 7 bytes of the message. See the table below for details.
7	Zero. This bit is always zero identifying the message as containing PVT data.

Format of Data Bytes in PVT Segment Mode

Buffer segments hold the PVT information to be added to the buffer. The PVT data is stored in the remaining 7 bytes of the message. The format of this data is indicated by the buffer format code encoded in byte 0.

Code	Description		
0	Bytes	Contents	
	1	The time (in milliseconds) until the start of the next PVT segment. Set to zero to end the move.	
	2-4	A 24-bit absolute position (counts). This is the starting position for this profile segment.	
	5-7	A 24-bit velocity given in 0.1 counts / second units.	
1	Same as for code 0, except velocity is in 10 ct/sec units. This allows greater velocity range with less precision.		
2	Same as for code 0, except the position is relative to the previous segment's position. If this is the first segment of a move, the position is relative to the starting commanded position.		
3	Same as for code 2, except velocity is in 10 ct/sec units.		
4	Bytes 1-4 hold a 32-bit absolute position (counts). This is not a full segment itself but is useful at the start of a move when a full 32-bit position must be specified. If the next segment is a relative position segment (code 2 or 3), its position is relative to this value.		
5	Bytes	Contents	
	1	The time (in milliseconds) until the start of the next linear IP segment. Set to zero to end the move.	
	2-5	A 32-bit absolute position (counts). This is the starting position for this profile segment.	
6	Same as for code 5, except the position is relative to the previous segment's position. If this is the first segment of a move, the position is relative to the starting commanded position.		
7-15	Reserved for future use.		

Segment Integrity Counter

Each segment of a move is given a 16-bit numeric identifier. The first segment is given the identifier 0, and each subsequent segment is given the next higher ID.

The three-bit integrity counter sent in byte zero of the segment should correspond to the lowest three bits of the ID code (i.e. zero for the first segment and increasing by 1 for each subsequent segment). If the drive receives non-consecutive segments, an error is flagged, and no further segments are accepted until the error is cleared. This allows the drive to identify missing segments in the move and stop processing data at that point. Because the PVT buffer status message includes the ID of the next expected segment, it should be possible to clear this error and resend the missing data before the buffer is exhausted.

TRAJECTORY BUFFER FREE COUNT   INDEX 0x2011

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	-	0 to $2^{16}-1$	T	R

This object gives the number of locations in the IP trajectory buffer that are currently available to accept new trajectory segments. It contains the same information as bits 16-23 of the Trajectory Buffer Status object (index 0x2012), below.

TRAJECTORY BUFFER STATUS   INDEX 0x2012

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	-	See *Description*, below.	T	R

This object gives access to status information about the IP trajectory buffer.

The status value is bit-mapped as follows:

Bit(s)	Description
0-15	These bits hold the 16-bit segment identifier of the next IP move segment expected. If a segment error has occurred (i.e. the segment integrity counter of a received message was out of order), then these bits may be consulted to determine the ID of the segment that should have been received.
16-23	The number of free locations in the IP buffer.
24	Set if a segment sequence error is in effect. A segment sequence error occurs when an IP segment is received with the incorrect value in its integrity counter.
25	Set if a buffer overflow has occurred.
26	Set if a buffer underflow has occurred.
27-30	Reserved for future use.
31	This bit is set if the IP buffer is empty.

This object is intended to be read using a PDO and has a PDO event associated with it. The event occurs when one of the error bits (24 – 26) is set, or when the trajectory generator removes a segment from the trajectory buffer.

NEXT TRAJECTORY SEGMENT ID   INDEX 0x2013

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	-	0 to 2^{16}-1	T	R

This object gives the full 16-bit value of the next trajectory segment expected by the buffer interface. It contains the same information as bits 0-15 of the Trajectory Buffer Status object (index 0x2012).

INTERPOLATION SUBMODE SELECT   INDEX 0x60C0

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	-	Bit mapped	TR	R

Determines which interpolation submode to use.

Sub mode	Description
0	Linear interpolation with constant time which is set in 0x60C2. Positions are added to the buffer by writing to 0x60C1, sub-index 0. In this mode the drive will make small adjustments to the constant time value to keep the buffer from under or overflowing. The drive can adjust the constant time of any particular segment by as much as 1 servo cycle.
-1	Linear interpolation with constant time for every point added. In order to prevent over under or overflowing of the buffer, new points must be added while keeping the master and slave synchronized. The master must monitor the PVT buffer status (0x2012) to ensure that there are adequate updates which is likely if there are small differences between the master and drive clocks. Constant timing ensures that the trajectory will follow the path defined by the position updates.
-2	Linear interpolation with variable time. Positions are added to the buffer by writing to 0x60C1, sub-index 0, then time (ms) is written to 0x60C1, sub-index 1.

Sub mode	Description
	Position and time are both added to the buffer when the sub-index 1 is written.
-3	Cubic polynomial interpolation. Position is first written to 0x60C1, sub-index 0, then time is written to 0x60C1, sub-index 1. Velocity is written to 0x60C1, sub-index 2. The new data will be latched when the velocity sub-index is written. NOTE: Copley Controls provides a set of alternate objects (0x 2010, 0x 2011, 0x 2012, and 0x 2013) for efficient PVT move handling. When using the alternate objects, it is not necessary to set a linear interpolation submode using this object 0x60C0.

INTERPOLATION DATA RECORD   INDEX 0x60C1

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	88	RECORD	TR	R

This object is used to send interpolation data to the drive's interpolation buffer.
Sub index 0 contains the number of sub-index entries.

INTERPOLATION POSITION   INDEX 0x60C1, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Counts	0 to 2^{32} -1	TR	R

A target position. Used in all three interpolation modes.

INTERPOLATION TIME   INDEX 0x60C1, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	milliseconds	0 to 2^8 -1	TR	R

The time interval of the move segment that starts with the Interpolation Position (Sub-Index 1), and extends to the next segment. Not used with interpolation mode 0 (linear interpolation with a constant time). In interpolation mode -1 (linear interpolation with variable time), writing to this object causes the entire record to be written to the interpolation buffer. Writing a value of zero to this object indicates the end of the interpolated move.

INTERPOLATION VELOCITY   INDEX 0x60C1, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/sec	0 to 2^{32} -1	TR	R

Used only in interpolation mode -2 (PVT). This is the velocity used to drive the axis to the Interpolation Position Target (Sub-Index 1) within the Interpolation Time (Sub-Index 2). Writing to this object causes the entire record to be written to the interpolation buffer.

INTERPOLATION TIME PERIOD   INDEX 0x60C2

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	32	RECORD	TR	R

Used only in interpolation mode 0 (linear interpolation with a constant time). Defines the segment interval. Sub index 0 contains the number of sub-index entries.

INTERPOLATION TIME VALUE INDEX 0x60C2, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED8	RW	variable	0 to 2^8 -1	TR	R

This object sets the constant time that is associated with each trajectory segment in interpolation mode 0.

INTERPOLATION TIME UNITS INDEX 0x60C2, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
INTEGER8	RW	-	-3 to -6	TR	R

This object indicates the Interpolation Time Units. The formula is, seconds = Interpolation Time Value * Interpolation Time Units	Time Units
-3	0.001
-4	0.0001
-5	0.00001
-6	0.000001

INTERPOLATION DATA CONFIGURATION INDEX 0x60C4

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	136	RECORD		R

Data is used to enable the drive to receive interpolation data before a move. Also, can be used to store positions and other data received from the master.
Sub-index 0 holds the number of elements in this object.

MAXIMUM BUFFER SIZE   INDEX 0x60C4 SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to 2^{32} -1	NO	R

The number of interpolation records that can be stored.

ACTUAL BUFFER SIZE   INDEX 0x60C4 SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to 2^{32} -1	R	R

The number of interpolation records that have been stored.

BUFFER ORGANIZATION   INDEX 0x60C4 SUB-INDEX 3

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RW	8	0 to 2^8 -1	R	R

0x00 = FIFO organization, 0x01 = ring buffer organization. Other values are not allowed.

BUFFER POSITION   INDEX 0x60C4 SUB-INDEX 4

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to 2^{32} -1	R	R

Dimensionless, indicating the next free buffer entry point.

SIZE OF DATA RECORD INDEX 0x60C4 SUB-INDEX 5

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RW	8	0 to 2^8-1	R	R

The number of bytes in the data record.

BUFFER CLEAR INDEX 0x60C4 SUB-INDEX 6

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RW	8	0,1	R	R

0x00 = Clear buffer inputs, disable access, and clear all IP data records.
0x01 = Enable access to input buffers. All other values are not allowed.

12 CYCLIC SYNCHRONOUS MODES

12.1 Cyclic Synchronous Position Mode (CSP)

In this mode the controller generates a trajectory and sends increments of position, along with velocity and current feed-forward values, to the drive.

The primary feedback from the drive is the actual motor position and optionally, actual motor velocity and torque.

Position, velocity, and torque control loops are all closed in the servo drive which acts as a follower for the position commands.

The diagram below shows an overview of the cascading control structure in CSP mode.

Objects in parallelograms are real-time PDO data.

Other objects in rectangles are usually configured SDOs, non-synchronously.
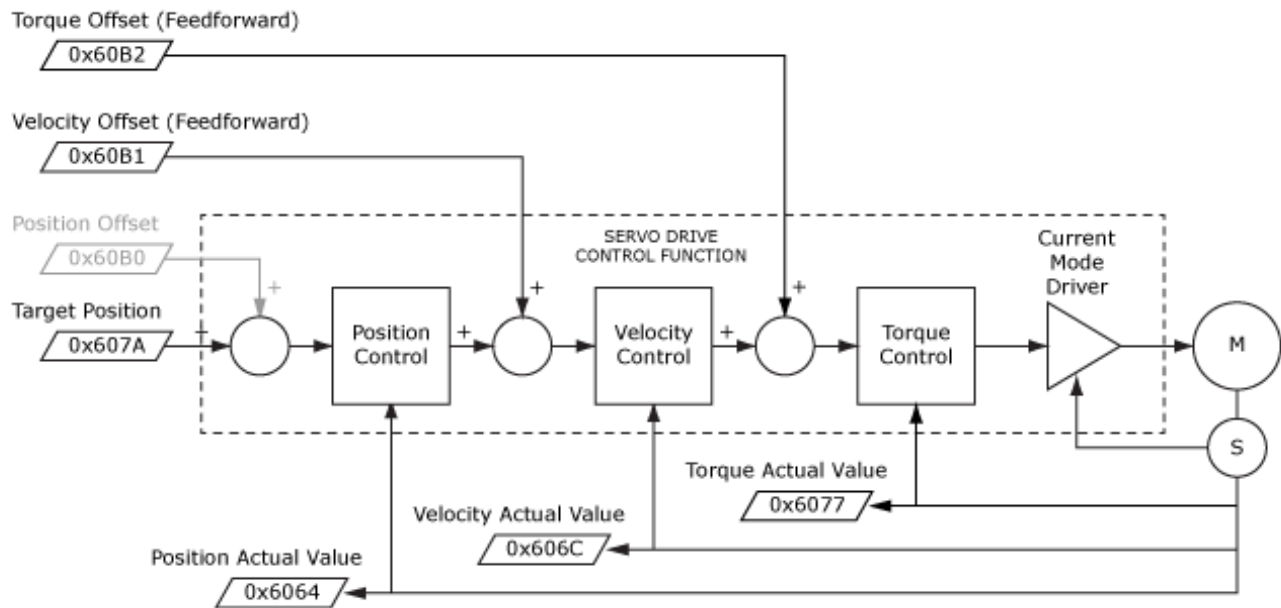
Figure 46—Overview of the cascading control structure in CSP mode

This diagram shows the CSP control function with real-time and other configuration parameters.
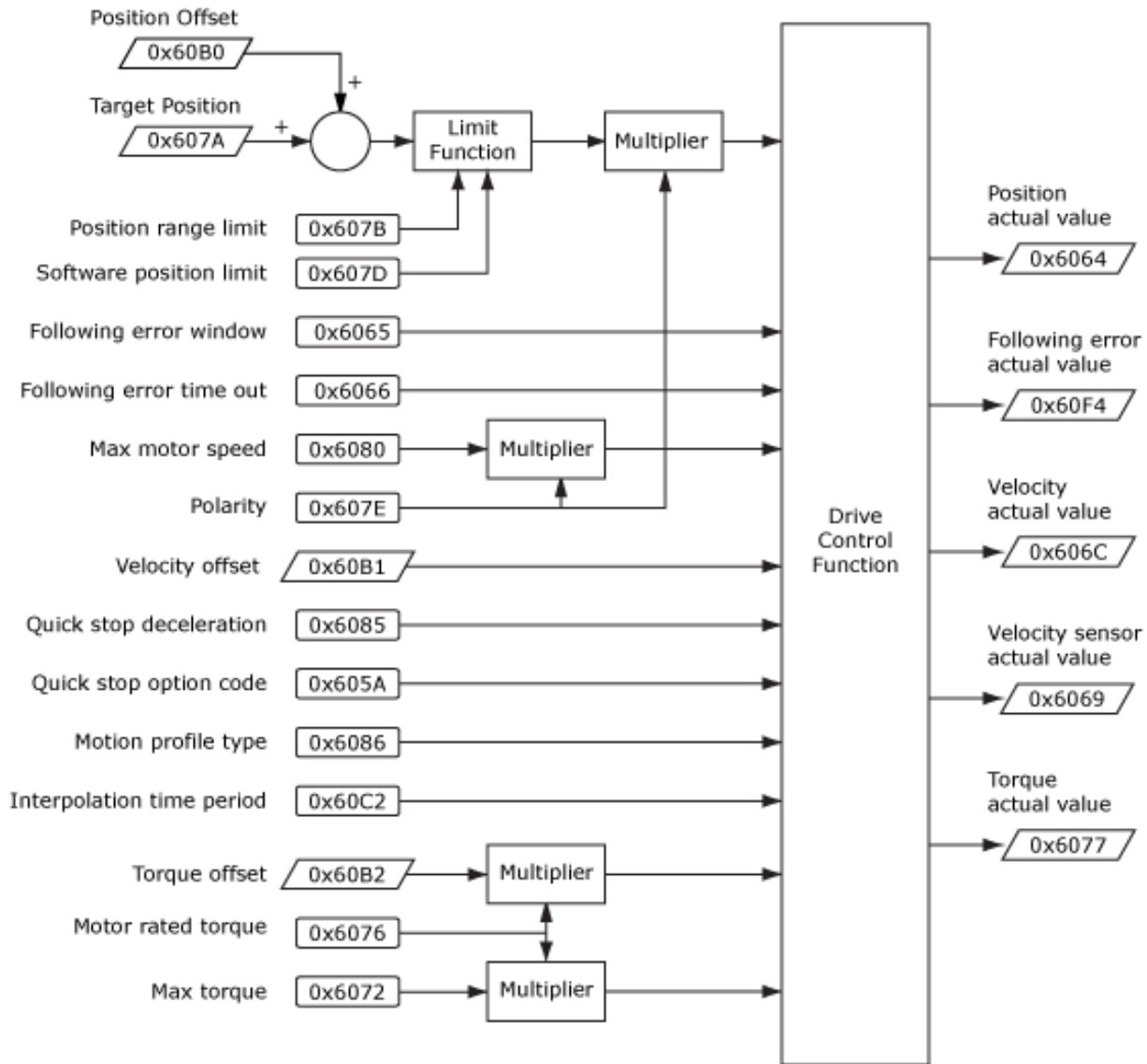
Figure 47-CSP Control Function

12.2 Cyclic Synchronous Velocity Mode (CSV)

CSV mode is frequently used with controllers that close the position loop and use the position error to command the velocity of the servo drive (which can also accept a torque feedward value). Velocity and torque loops are closed in the servo drive.

The diagram below shows an overview of the cascading control structure in cyclic synchronous velocity mode (CSV).
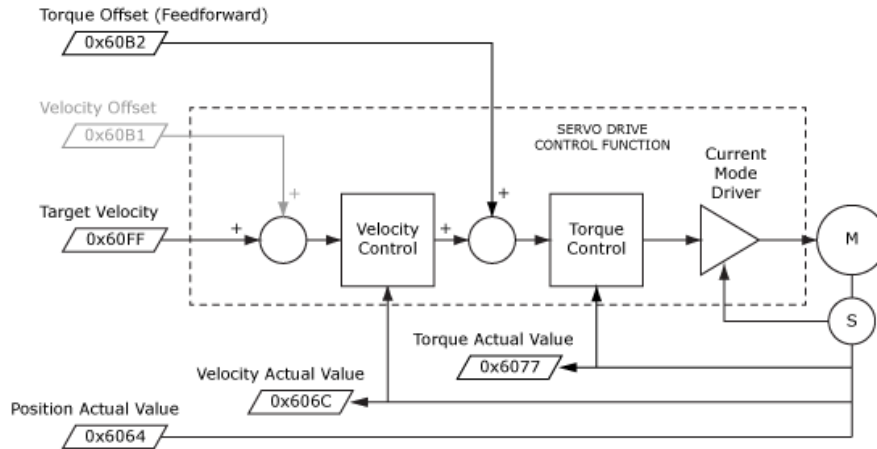
Figure 48—CSV control structure

The diagram below shows the CSV control function with real-time and other configuration parameters.
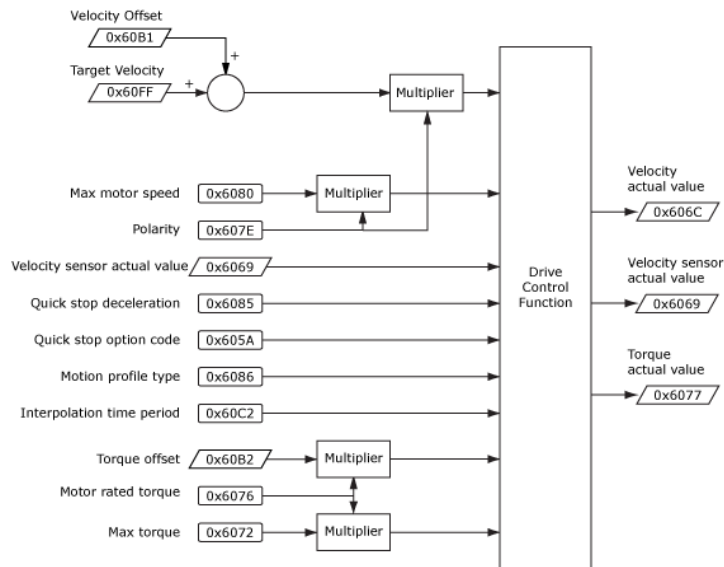
Figure 49—CSV control function

12.3 Cyclic Synchronous Torque Mode (CST)

When the controller has a full PID compensator to control position and velocity, the output is a torque command to the servo drive. A torque offset value is used for vertical loads to balance against gravity so that the torque command from the controller produces symmetrical acceleration up and down.

The diagram below shows an overview of the cascading control structure in cyclic synchronous torque mode (CST).
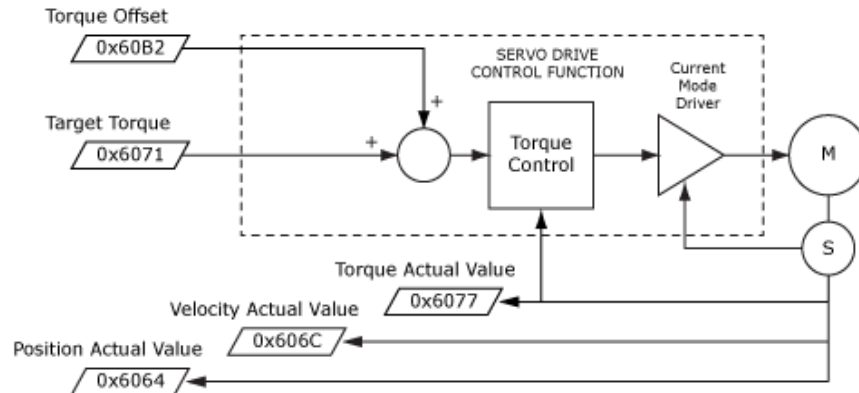
Figure 50—CST control structure

This diagram shows the CST control function with real-time and other configuration parameters.
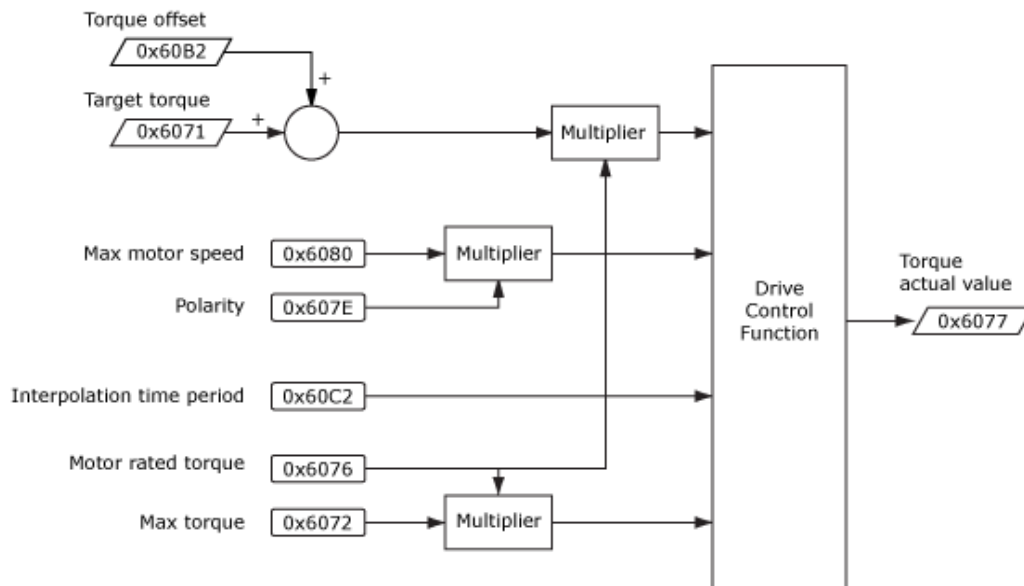
Figure 51—CST control function

13 ONLY FOR ETHERCAT OBJECTS

DEVICE IDENTIFICATION RELOAD OBJECT (DEVICE ID)　　INDEX 0x10E0

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	56	RECORD	NO	R

This object is used to explicitly reload register 0x0012 in the ESC (EtherCAT Slave Controller) that contains the Configured Station Alias (Device ID). Sub-index 0 contains the number of sub-elements of this record.
Register addresses shown below refer to ETG 1100 Slave Controller.

CONFIGURED STATION ALIAS REGISTER VALUE　　INDEX 0x10E0, SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RW	16	0 to 2^{16}-1	NO	R

Write: write a value into register 0x0012. Read: read current value of register 0x0012
This enables changing of the Device ID value without resetting (power-cycle) the drive.

WRITE CONFIGURED STATION ALIAS PERSISTENT　　INDEX 0x10E0, SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RW	8	0,1	NO	R

FALSE(0): Write access to SI 1 will write value to register 0x0012 only

TRUE(1): Write access to SI 1 will write value to register 0x0012 and to the SII (Slave Information Interface)

RELOAD ID-SELECTOR VALUE　　INDEX 0x10E0, SUB-INDEX 3

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RW	16		NO	R

Write: writing 0x0000 to the subindex updates the current ID-selector (switches) value into register 0x0012

Read: read current value of ID-selector

BACKUP PARAMETER INFO INDEX 0x10F0

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RO	48	-	NO	R

Sub-index 0 contains the number of sub-elements of this record.

CRC OF PARAMETER STORAGE　　INDEX 0x10F0, SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UMSIGNED32	RO	32	0 to 4.3G	NO	R

This value will be compared to the CRC of the drive flash memory to verify its integrity after a reset or power-on event. If they are different it will set bit 0 of 0x2183 to indicate a fatal fault.

ETHERCAT ERROR SETTINGS   INDEX 0X10F1

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED8	RO	8	0 to 2^8 -1	NO	R

Sub-index 0 contains the number of sub-elements of this record.

LOCAL ERROR REACTION   INDEX 0X10F1, SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to 2^{32} -1	NO	R

0 = PDO State Slave changes the value of the PDO State Bit (either in the PDO Parameter or if it is part of the process data description then there. Slave remains in OP

1 = Disable SyncManager (outputs) from PDI side and go to ErrSafeOp

2 = Device specific state Input PDOs contains a device specific value, e.g. profile/device specific error code

SYNC ERROR COUNTER LIMIT INDEX 0X10F1, SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RW	16	0 to 2^{16} -1	NO	?

Limit set by the master when slave shall change its EtherCAT State to SAFEOP with AL Status Code 0x1A.
The Slave shall reset its internal error counter when it is synchronized again and confirms the state transition from SAFEOP to OP.

SYNC MANAGER 2 SYNCHRONIZATION   INDEX 0X1C32

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	288	RECORD	NO	R

Sub-index 0 contains the number of sub-elements of this record.

SYNC MANAGER 2, SYNCHRONIZATION TYPE INDEX 0X1C32 SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UINT	RW	16	0~3	NO	R

0: Free Run
1: Synchron with SM 2 event
2: DC Mode, Synchron with SYNC0 event
3: DC Mode, Synchron with SYNC1 event

SYNC MANAGER 2, CYCLE TIME   INDEX 0X1C32 SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32		NO	R

Cycle time in ns:
Free run: cycle time of local timer
Sychron with SM 2 Event: cycle time of the master
DC Mode: SYNC0/SYNC1 Cycle Time

SYNC MANAGER 2, SHIFT TIME   INDEX 0X1C32 SUB-INDEX 3

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	:	NO	R

Time between SYNC0 Event and Outputs Valid (in ns, only in DC-Mode)

SYNC MANAGER 2, SYNC TYPES SUPPORTED INDEX 0X1C32 SUB-INDEX 4

Type	Access	Bits	Range	Map PDO	Memory
UINT	RO	16	Bit mapped	NO	R

Supported synchronization modes:
- Bit 0 = 1: Free Run is supported
- Bit 1 = 1: Synchron with SM 2 Event is supporte
- Bit 2-3 = 01: DC-Mode is supported
- Bit 4-5 = 10: Output Shift with SYNC1 Event (only DC-Mode)
- Bit 14 = 1: dynamic times (could be measured Messen by writing 0x1C32:08)

SYNC MANAGER 2, MINIMUM CYCLE TIME   INDEX 0X1C32 SUB-INDEX 5

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Minimum cycle time supported (in ns).

SYNC MANAGER 2, CALC AND COPY TIME   INDEX 0X1C32 SUB-INDEX 6

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Minimal time between SYNC0 and SYNC1 Event (in ns, only in DC-Mode).

SYNC MANAGER 2, MINIMUM HARDWARE DELAY   INDEX 0X1C32 SUB-INDEX 7

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Time in ns.

SYNC MANAGER 2, RESERVED   INDEX 0X1C32 SUB-INDEX 8

Type	Access	Bits	Range	Map PDO	Memory
UINT	RO	16		NO	R

Reserved for future use.

SYNC MANAGER 2, HARDWARE DELAY TIME INDEX 0X1C32 SUB-INDEX 9

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Time between SYNC1 Event and Outputs Valid (in ns, only in DC-Mode).

SYNC MANAGER 2, SYNC0 CYCLE TIME   INDEX 0X1C32 SUB-INDEX 10

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Sync 0 cycle time.

SYNC MANAGER 3 SYNCHRONIZATION   INDEX 0x1C33

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	-	-	NO	R

Sub-index 0 contains the number of sub-elements of this record.

SYNC MANAGER 3, SYNCHRONIZATION TYPE INDEX 0x1C33 SUB-INDEX 1

Type	Access	Bits	Range	Map PDO	Memory
UINT	RW	16	0~3	NO	R

0: Free Run
1: Synchron with SM 3 event
2: DC Mode, Synchron with SYNC0 event
3: DC Mode, Synchron with SYNC1 event
34: Sychron with SM 2 Event (Outputs available)

SYNC MANAGER 3, CYCLE TIME   INDEX 0x1C33 SUB-INDEX 2

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32		NO	R

Cycle time in ns:

- Free run: cycle time of local timer
- Sychron with SM 3 Event: cycle time of the master
- DC Mode: SYNC0/SYNC1 Cycle Time

SYNC MANAGER 3, SHIFT TIME   INDEX 0x1C33 SUB-INDEX 3

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	:	NO	R

Time between SYNC0 Event and Outputs Valid (in ns, only in DC-Mode)

SYNC MANAGER 3, SYNC TYPES SUPPORTED INDEX 0x1C33 SUB-INDEX 4

Type	Access	Bits	Range	Map PDO	Memory
UINT	RO	16	Bit mapped	NO	R

Supported synchronization modes:
Bit 0 = 1: Free Run is supported
Bit 1 = 1: Synchron with SM 2 Event is supported (Outputs available)
Bit 2-3 = 01: DC-Mode is supported
Bit 4-5 = 01: Input Shift with local event (Outputs available)
Bit 4-5 = 10: Input Shift with SYNC1 Event (no Outputs available)
Bit 14 = 1: dynamic times (could be measured Messen by writing 0x1C32:08 or 0x1C33:08)

SYNC MANAGER 3, MINIMUM CYCLE TIME   INDEX 0x1C33 SUB-INDEX 5

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Minimum cycle time supported (in ns).

SYNC MANAGER 3, CALC AND COPY TIME   INDEX 0X1C33 SUB-INDEX 6

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Time between Input Latch and the availability of the inputs for the master (in ns, only in DC-Mode).

SYNC MANAGER 3, MINIMUM HARDWARE DELAY   INDEX 0X1C33 SUB-INDEX 7

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Time in ns.

Sync Manager 3, Reserved   Index 0x1C33 Sub-Index 8

Type	Access	Bits	Range	Map PDO	Memory
UINT	RO	16		NO	R

Reserved for future use.

SYNC MANAGER 3, HARDWARE DELAY TIME INDEX 0X1C33 SUB-INDEX 9

Type	Access	Bits	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Time between SYNC1 Event and Outputs Valid (in ns, only in DC-Mode)

SYNC MANAGER 3, SYNC0 CYCLE TIME   INDEX 0X1C33 SUB-INDEX 10

Type	Access	Units	Range	Map PDO	Memory
UDINT	RO	32	0~4.295 seconds	NO	R

Time in ns.

NODE GUARDING ERROR ACTION INDEX 0X21B2

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RW	16	0~1	R	R

This parameter can be set to stop and hold or disable the drive when a heartbeat error occurs..
0 = stop and hold, 1 = disable drive.

14 ALTERNATIVE CONTROL SOURCES

14.1 Overview

Typically, when a Copley drive is used on a CANopen network, the CANopen master uses the network to send commands that drive the drive's position, velocity, or current loop.

Alternately, an drive on a CANopen network can accept position, velocity, or current commands over the device's serial port, digital I/O channels, or analog reference inputs, or run under the control of the drive's internal generator or a Copley Virtual Machine (CVM) program. Use the Indexer Register Values object (index 0x2600, p. 239) to read and write the CVM Indexer program registers.

An drive can also run in camming mode to execute moves programmed in camming tables. The Camming Configuration object (index 0x2360, p. 238) and several other objects described in this chapter are used to configure and operate the drive in camming mode.

Even while operating under an alternative control source, a device's status can still be monitored over the CANopen network.

Specify a control source by choosing a mode in the Desired State object (index 0x2300). For more information, see page 68.

Other objects affect the drive under alternative control sources. They are described in the next section.

14.2 Alternative Control Source Objects

MICRO-STEPPING RATE   INDEX 0X21C1

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	degrees / sec	-2^{15} to $+2^{15}-1$	TR	RF

This value is only used when running in diagnostic micro-stepping mode. It gives the step angle update rate. See Desired State object (index 0x2300, p. 68), code 42.

ANALOG REFERENCE SCALING FACTOR   INDEX 0X2310

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	See *Description*, below.	-2^{31} to $+2^{31}-1$	TR	F

When running in a mode that relies on the analog reference as an input, this object defines the scaling that is applied to the analog reference input. See Desired State object (index 0x2300, p. 68, codes 2, 12, 22.

Mode	Scaling
Current	0.01 Amps /10 volt.
Velocity	0.1 counts / second / 10 volt.
Position	1 Count /10 volt.

ANALOG REFERENCE OFFSET   INDEX 0x2311

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	millivolts	-2^{15} to $+2^{15}-1$	TR	F

This is one of two offset values applied to the analog reference input before it is used in calculations.

ANALOG REFERENCE CALIBRATION OFFSET   INDEX 0x2312

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	millivolts	-2^{15} to $+2^{15}-1$	TR	F

This voltage is added to the analog command input and is calibrated at the factory to give a zero reading for zero input voltage. It is one of two offset values applied to the analog reference input before the input is used in calculations.

ANALOG REFERENCE DEADBAND   INDEX 0x2313

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	millivolts	-2^{15} to $+2^{15}-1$	TR	F

The analog reference input is subject to a non-linear adjustment to clip reading around zero. This object defines the size of that window.

SECONDARY ANALOG REFERENCE VALUE INDEX 0x2208

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	mV	-2^{15} to $+2^{15}-1$	T	R

Secondary analog reference value.

MOTOR TEMP SENSOR VOLTAGE   INDEX 0x2209

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RO	mV	-2^{15} to $+2^{15}-1$	T	R

Present voltage at analog motor temperature sensor.

MOTOR TEMP SENSOR LIMIT   INDEX 0x220A

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	mV	-2^{15} to $+2^{15}-1$	TR	F

Limit for analog motor temperature sensor. If this parameter is set to zero, then the analog motor temperature sensor is disabled. If this parameter is set to a positive value, then a motor temperature error will occur any time the voltage on the motor temperature input exceeds this value (in millivolts). If this parameter is negative, then a motor temperature error will occur any time the voltage on the motor temperature input is lower than the absolute value of this limit in millivolts.

PWM INPUT FREQUENCY   INDEX 0x2322

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	10 Hz	0 to $2^{16}-1$	TR	RF

This is the frequency of the PWM for use only in UV commutation mode
(Desired State object [index 0x2300, p. 68].

FUNCTION GENERATOR CONFIGURATION INDEX 0x2330

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Bit mapped	See Description, below.	TR	RF

Configures the drive's internal function generator, which can drive the current, velocity, or position loop. Bit-mapped:

Bits	Description
0-1	Function code.
2-11	Reserved for future use.
12	One-shot mode. If set, the function code is reset to zero (disabled) after one complete waveform.
13	Invert every other waveform if set.
14-15	Reserved for future use.

The function code programmed into bits 0-1 defines the type of waveform to be generated:

Code	Description
0	None (disabled)
1	Square wave.
2	Sine wave.

Note that the drive is placed under control of the function generator by setting the Desired State object (index 0x2300, p. 68) to one of the following values:

Bits	Description
4	Function generator drives current loop.
14	Function generator drives velocity loop.
24	Function generator drives position loop in servo mode.
34	Function generator drives position loop in stepper mode.

FUNCTION GENERATOR FREQUENCY INDEX 0x2331

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Hz	0 to $2^{16}-1$	TR	RF

This object gives the frequency of the internal function generator.

FUNCTION GENERATOR AMPLITUDE INDEX 0x2332

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	See *Description*, below	-2^{31} to $+2^{31}-1$	TR	RF

The amplitude of the signal generated by the internal function generator.
The units depend on the servo operating mode:

Mode	Units
Current	0.01 Amps
Velocity	0.1 counts/second
Position	Counts

FUNCTION GENERATOR DUTY CYCLE   INDEX 0x2333

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	0.1 percent	0 to 1000	TR	RF

This object gives the function generator duty cycle for use with the square wave function.
It has no effect when running the sine function.

CAMMING CONFIGURATION   INDEX 0x2360

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Bits	Bit mapped	TR	RF

Configures Camming Mode operation:

Bits	Description
0-3	ID Number of the Cam Table to use (0-9)
4	Reserved.
5	If set, exit table in forward direction.
6	If set, use the Camming Internal Generator. The internal generator runs at the constant velocity programmed in Cam Master Velocity (index 0x2363, p. 239). If clear, use digital command input as configured in using Copley's CME 2 software camming controls or Input Pin States (index 0x2190, p. 113)
7	If set, run tables stored in RAM. If clear, use tables stored in the flash file system.
8-11	Input number to use as Cam Trigger. Note: a value of 0 selects IN1, value of 1 selects IN2, etc.
12-13	Cam Trigger type:
		Value	Type
		0	None (Continuous): The active Cam Table is repeated continuously.
		1	Use Input, Edge: The active Cam Table begins executing on the rising edge of the input pin selected by bits 8-11.
		2	Use Input, Level: The active Cam Table will run as long as the input selected by bits 8-11 is high.
		3	Use Master (Secondary) Encoder Index: The active Cam Table is executed when the drive receives an index pulse from the Master encoder. Index pulses received during execution are ignored.

CAM DELAY FORWARD   INDEX 0x2361

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Master Counts.	-2^{15} to $+2^{15}-1$	TR	RF

The delay applied before beginning a camming profile after the trigger has been activated, in a forward direction.

CAM DELAY REVERSE   INDEX 0x2362

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Master Counts.	-2^{15} to $+2^{15}-1$	TR	RF

The delay (in master counts) applied before beginning a camming profile after
the trigger has been activated, in a reverse direction.

CAM MASTER VELOCITY   INDEX 0X2363

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	0.1 counts/second	-2^{31} to $+2^{31}-1$	TR	RF

Virtual master encoder velocity for camming mode.

TRACE BUFFER RESERVED SIZE   INDEX 0X250A

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	RAM Words	0 to $2^{16}-1$	TR	R

The number of RAM words in the drive Trace Buffer to reserve for Trace Buffer Data (such as CAM tables).

TRACE BUFFER ADDRESS   INDEX 0X250B

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Data RAM address	0 to $2^{16}-1$	TR	R

An offset from the beginning of the memory reserved for Trace Buffer Data (index 0x250C, p. 239). Designates the location where the next Trace Buffer Data write (such as a CAM table master/slave value pair) will be stored.

TRACE BUFFER DATA   INDEX 0X250C

Type	Access	Units	Range	Map PDO	Memory
INTEGER32	RW	Trace data	-2^{31} to $+2^{31}-1$	TR	R

The first value written to this object will be stored in trace buffer RAM at the location specified by Trace Buffer Address (index 0x250B). On each subsequent write to this object, an internal pointer is incremented, and the value will be written to the next memory location. One use of this data object is the storage of CAM Table master/slave position value pairs.

INDEXER REGISTER VALUES   INDEX 0X2600

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	1040	RECORD	TR	R

This array object holds the values of the 32 programmable registers (0-31) maintained by the CVM Indexer Program. Each sub-index object 1-32 contains the value of an Indexer Program register (sub-index object 1 contains the value of Indexer Program register 0, sub-index object 32 contains the value of register 31). Sub-index 0 contains the number of sub-indexes.
Note: When the CVM Indexer program is started, all registers are initialized to zero.

INDEXER REGISTER VALUES   INDEX 0X2600, SUB-INDEX 1-32

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED32	RW	32	0 to $2^{32}-1$	TR	R

One sub-index object for each Indexer program register.

14.3 Running CAM Tables from RAM

Normally, Cam Tables are stored in the drive's flash memory, allowing the Cam Tables to be uploaded once and persist between power cycles.

In applications where flash storage is not appropriate or optimal, up to 16 Cam Tables can be loaded into and run from drive RAM.

(For a full description of camming, see the *Copley Camming User Guide.*)

Cam Tables in Drive RAM

NOTE: Increments vs. Positions. When entering Cam Table data in CME 2, the user enters pairs of absolute master and slave positions. CME 2 then converts the absolute position values to increment values. When writing Cam Table data to drive RAM, the controller program must write increment values (not absolute position values).

Using the Trace Buffer RAM Area for Cam Tables

Cam tables can be stored in and run from the area of drive RAM called the trace buffer. This RAM area is normally reserved for trace data collected by the CME 2 Scope Tool. When not needed for trace data, it may be used for other purposes, including the storage of Cam Tables.

RAM Cam Table Capacity

The Trace Buffer is 2048 16-bit words long. It can store up to 16 Cam Tables.

The maximum number of master/slave increment value pairs that can be stored in RAM varies. If the master increment is constant, a compressed format can be used.

Furthermore, each Cam Table requires two words of metadata, so using 16 tables would reduce the data allocation by 32 words.

Using one table in compressed format, about 2,000 master/slave increment value pairs can be represented. A typical maximum is about 1000 value pairs.

CAM Table Structure

When used for Cam Tables, the trace buffer begins with Cam Table metadata consisting of up to 16 word-pairs (32 words). The first word in each pair defines the address (offset from the beginning of the buffer). The second word contains the length of the Cam Table.

The metadata is followed by Cam Table data, starting at the address (offset) specified in the metadata.

In standard format, Cam Table data consists of master/slave increment value pairs. The first word in a pair contains a master increment and the second word contains the corresponding slave increment.

A compressed format may be used when the master increment changes at a constant rate as described in Compressed Format for Uniform Master Increments (p. 241).

NOTE: The controller program must make sure that there is a pair of metadata words for each Cam Table. The metadata rows must start at address (offset) 0 and must be in table ID order. For instance, the metadata pair that begins at address 0 defines Cam Table 0, the metadata pair that begins at address 2 defines Cam Table 1, etc. When configured to run Cam Table 0, the drive will look at address 0 for a metadata pair. When configured to run Cam Table 1, the drive will look at address 2, and so on.

Example: Single Cam Table

The following example shows a single Cam Table (identified at run time as Cam Table 0) stored in the trace buffer RAM area. The first pair of words contains the Cam Table's metadata. Word 1 contains the address (offset) to the beginning of Cam Table 0. The second word contains the length of the table.

The remaining words begin at address 2 and contain Cam Table data in the form of master/slave increment value pairs.

Address	Data	Data Description
0	2	Address of the start of Cam Table 0.
1	30	Length of Cam Table 0.
2	100	A master/slave increment value pair. For each 100 master increments, the slave axis is incremented 50 encoder counts.
3	50	
4-31	xxxx	Additional master/slave increment value pairs.

Example: Multiple Cam Tables

The following example shows three Cam Tables stored in the trace buffer RAM area. The first pair of words contains the metadata for Cam Table 0. The second and third word-pairs contain the metadata for Cam Tables 1 and 2, respectively.

The remaining words begin at address 6 and contain Cam Table data, in the form of master/slave increment value pairs, for the three Cam Tables.

Address	Data	Data Description
0	2	Address of the start of the Cam Table 0.
1	30	Length of Cam Table 0.
2	36	Address of the start of Cam Table 1.
3	24	Length of Cam Table 1.
4	60	Address of the start of Cam Table 2.
5	64	Length of Cam Table 2.
6—35	xxxx	Cam Table 0 data in the form of master/slave increment value pairs.
36—59	xxxx	Cam Table 1 data in the form of master/slave increment value pairs.
60—123	xxxx	Cam Table 2 data in the form of master/slave increment value pairs.

Compressed Format for Uniform Master Increments

When the Cam Master increments at a constant rate, a compressed format may be used to save RAM space.

In standard format, each master/slave increment value pair is expressed using two words, one for the master and one for the slave.

In the compressed format, the constant master increment is stored in the table's first data word and the slave increments are stored in the subsequent data words.

To indicate that the compressed format is used, set bit 14 of the first data word (which contains the master increment value). Clear bit 15.

Example: A Table in Compressed Format

Address	Data	Data Description
0	2	Address of the start of Cam Table 0.
1	30	Length of Cam Table 0.
2	50	The constant master increment. To indicate that this is a constant master increment for a compressed table, bit 14 is set and bit 15 is clear.
3—31	xxxx	A series of slave increment values.

Procedures for Running Cam Tables from RAM

Process overview:

1. Allocate RAM for Cam Tables

- Write to coder velocity for camming mode.

- Trace Buffer Reserved Size (index 0x250A, p. 239) the number of memory words to reserve for Cam Tables.

2. Load a Cam Table into RAM

- Write to Trace Buffer Address (index 0x250B, p. 239) the Cam Table's initial offset value.

- Write a series of values to Trace Buffer Data (index 0x250C, p. 239).

- For standard table format, the series starts with a master increment value followed by the corresponding slave increment, and the master/slave pairing sequence is repeated for each row of Cam Table data.

- For compressed table format, the first value is the constant master increment value. Bit 14 of this first word is set, and bit 15 is clear. Subsequent values written to Trace Buffer Data represent the series of slave increments.

- Each time a value is written to or read from Trace Buffer Data, the drive increments the offset pointer in Trace Buffer Address.

3. Configure the Camming Parameters

- To configure the drive to run Cam Tables from RAM, set bit 7 in the Camming Configuration object (index 0x2360, p. 238). Set other parameters as needed.

4. Run a Cam Table from RAM

- Set the Desired State object (index 0x2300, p. 68) to 25 (camming mode).

- The Cam Table selected in bits 0-3 of the Camming Configuration object will be run in response to the trigger events specified in bits 12-13 of the Camming Configuration object.

15 TRACE TOOL

15.1 Overview

The Copley Controls trace tool allows the programmer to configure and monitor up to 6 motion trace channels. Each channel can be configured to monitor any of a number of trace variables. Other configuration choices include the trace period and trace trigger.

TRACE CHANNEL CONFIGURATION INDEX 0X2500

Type	Access	Bits	Range	Map PDO	Memory
ARRAY	RW	112	See Description, below.	NO	R

This object uses 6 sub-indices configure up to 6 trace channels.
Sub-index 0 holds the number of trace channels.

TRACE CHANNELS   INDEX 0X2500, SUB-INDEX 1-6

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	Code values	0 to 55	NO	R

Sub-object x configures trace channel x. Each channel can be configured to monitor one of the trace variables described below by programming the sub-object with the code.

Code	Trace Variable
0	No data. Setting a channel to this value disables it. Disabling unused channels saves space in the trace buffer.
3	Current reading winding A (0.01 amps)
4	Current reading winding B (0.01 amps)
5	Reference A/D reading (millivolts)
6	High voltage reference (0.1 volts)
7	Commanded torque
8	Limited torque
9	Commanded current (D rotor axis) (0.01 amps)
10	Commanded current (Q rotor axis) (0.01 amps)
11	Actual current (X stator axis) (0.01 amps)
12	Actual current (Y stator axis) (0.01 amps)
13	Actual current (D rotor axis) (0.01 amps)
14	Actual current (Q rotor axis) (0.01 amps)
15	Current Error (D rotor axis) (0.01 amps)
16	Current Error (Q rotor axis) (0.01 amps)
17	Current Integral (D rotor axis)
18	Current Integral (Q rotor axis)
19	Current loop output (D rotor axis)
20	Current loop output (Q rotor axis)
21	Current loop output (X stator axis)
22	Current loop output (Y stator axis)
23	Actual motor velocity (0.1 counts/sec or 0.01 RPM if using back EMF velocity estimate).
24	Commanded motor velocity.
25	Limited motor velocity command.
26	Velocity loop error.
27	Velocity loop integral.
28	Actual load position (counts).
29	Commanded position.

Code	Trace Variable
30	Position loop error
31	Motor encoder position (counts)
32	Position loop output velocity
33	Raw input pin readings (no debounce)
34	reserved
35	reserved
36	Motor phase angle (1 degree units)
37	Drive temperature (degrees C)
38	Drive Manufacturer Status Register (index 0x1002, p. 62)
39	Drive event latch word
40	Hall sensor state
41	Position Capture Status Register (index 0x2401, p. 192)
42	Index capture register
43	Load encoder velocity (0.1 counts / second).
44	Velocity command from trajectory generator (0.1 counts/sec)
45	Acceleration command from trajectory generator (10 counts/sec^2)
46	The analog encoder sine input. Only valid for drives with analog encoder support.
47	The analog encoder cosine input. Only valid for drives with analog encoder support.
48	The value of the digital inputs (after debounce)
49	The destination position input to the trajectory generator.
50	Actual motor velocity as seen by velocity loop. This is an unfiltered version of trace variable.
51	Load encoder position (counts).
52	Gain scheduling key parameter value.
53	Position loop P gain
54	Velocity loop P gain
55	Velocity loop I gain

TRACE SYSTEM STATUS   INDEX 0x2501

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	R0	Bit	Bit mapped	T	R

Get trace status:

Bits	Description
0	Set if trace data is currently being collected.
1	Set if trigger has occurred.
2-15	Reserved for future use.

TRACE REFERENCE PERIOD INDEX 0x2502

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED32	RO	nanoseconds	0 to $2^{32}-1$	NO	R

Get fundamental period. Returns a 32-bit value containing the fundamental trace period in units of nanoseconds. The fundamental period is the maximum frequency at which the trace system can sample data. The actual trace period is set in integer multiples of this value using the Trace Period object (0x2505).

TRACE SAMPLE COUNT   INDEX 0x2503

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	Samples	0 to $2^{16}-1$	T	R

Returns the number of samples collected so far.

TRACE MAX SAMPLES   INDEX 0x2504

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RO	Samples	0 to 2^{16}-1	NO	R

The maximum number of samples that the internal trace memory buffer can hold is calculated and returned as a 16-bit value. Note that the maximum number of samples is dependent on the number and type of active trace variables. Set the trace variables first; then read the maximum number of samples available.

TRACE PERIOD   INDEX 0x2505

Type	Access	Units	Range	Map PDO	Memory
UNSIGNED16	RW	See description.	0 to 2^{16}-1	NO	R

The trace period, in integer multiples of the Trace Reference Period (0x2502).

TRACE TRIGGER CONFIGURATION INDEX 0x2506

Type	Access	Units	Range	Map PDO	Memory
ARRAY[0..2] of UNSIGNED16	RW	Bit mapped	RECORD	NO	R

Set/get the trace trigger configuration. Three additional words of data are supplied that identify the type of trigger used to start the trace. The first word has the following format:

Bits	Description	
0-3	Channel number to trigger on (if applicable).	
4-7	Reserved.	
8-11	Trigger type (may be interpreted differently for some trigger types):	
	Type	**Description**
	0	No trigger in use
	1	Trigger as soon as the selected channel's input is greater than or equal to the trigger level
	2	Trigger as soon as the selected channel's input is less than or equal to the trigger level
	3	Trigger when the selected channel's input changes from below to above the trigger level
	4	Trigger when the selected channel's input changes from above to below the trigger level
	5	Trigger when any selected bits in the channel value are set. The bits are selected using the trigger level value as a mask
	6	Trigger when any selected bits in the channel value are clear. The bits are selected using the trigger level value as a mask
	7	Trigger any time the selected channel value changes
	8	The trigger level mask selects one or more bits in the Manufacturer Status Register (index 0x1002, p. 62). The trigger occurs when any of these bits change from to 1. In this mode, the channel number selected by the trigger is not used.
	9	Like type 8, but the trigger occurs when the bit(s) change from 1 to 0
	10	Trigger on the start of the next function generator cycle. This trigger type is only useful when running in function generator mode. The trigger channel number isn't used.
12-14	Reserved	
15	If set, take one sample per trigger event	

Normally, the two following data words specify a 32-bit trigger level (sent high word first).

These data values may be interpreted differently for some trigger types.
The trigger types are shown below:

Type	Description
0	No trigger in use
1	Trigger as soon as the selected channel's input is greater than or equal to the trigger level
2	Trigger as soon as the selected channel's input is less than or equal to the trigger level
3	Trigger when the selected channel's input changes from below to above the trigger level
4	Trigger when the selected channel's input changes from above to below the trigger level
5	Trigger when any selected bits in the channel value are set. The bits are selected using the trigger level value as a mask
6	Trigger when any selected bits in the channel value are clear. The bits are selected using the trigger level value as a mask
7	Trigger any time the selected channel value changes
8	The trigger level mask selects one or more bits in the event status word. The trigger occurs when any of these bits change from 0 to 1. In this mode, the channel number selected by the trigger is not used.
9	Like type 8, but the trigger occurs when the bit(s) change from 1 to 0
10	Trigger on the start of the next function generator cycle. This trigger type is only useful when running in function generator mode. The trigger channel number isn't used.
11	The trigger level mask selects one or more bits in the capture status word (parameter 0x6d). The trigger occurs when any select bit in the status word is set. This also clears the selected bits in the capture status which makes it useful for triggering on a new capture value even if the captured value didn't change.

TRACE TRIGGER DELAY   INDEX 0X2507

Type	Access	Units	Range	Map PDO	Memory
INTEGER16	RW	Trace Period	-2^{15} to $+2^{15}-1$	NO	R

Set/get the delay between the trigger occurring and the start of captured data. The delay is given in units of trace periods (0x2505).

Note that the delay may be either positive or negative. A negative delay means that the data captured will precede the trigger event by the specified number of cycles. Although any input value is accepted, the number of samples preceding the trigger is limited to the length of the trace buffer and the number (and size) of channels being captured.

TRACE START/STOP INDEX 0X2508

Type	Access	Bits	Range	Map PDO	Memory
UNSIGNED16	RW	16	0 to $2^{16}-1$	TR	R

Write 0 to stop trace collection or a non-zero value to restart it.

TRACE DATA INDEX 0X2509

Type	Access	Bits	Range	Map PDO	Memory
ARRAY[0..16383] of UINT	RO	262144	0 to $2^{16}-1$ for each sample	NO	R

After a trace has been collected, the trace data can be downloaded by reading from this object. The downloaded data should be viewed as an array of 32-bit samples.

16 COMMON DEVICE PROFILE OBJECTS

For more information on the Common Device Profile Objects listed in the table, refer to *ETG 5003 Semiconductor Device Profile Part 1 Common Device Profile* (ETG.5003-1 S (R) V1.1.0).

Objects	Name
0xF000	Semiconductor device profile
0xF010	Module profile list
0xF380	Active exception status
0xF381	Active device warning details
0xF383	Active device error details
0xF390	Latched exception status
0xF391	Latched exception warning details
0xF393	Latched exception error details
0xF3A1	Device warning details
0xF3A3	Device error details
0xF6F0	Input latch local timestamp
0xF9F0	Manufacturer serial number
0xF9F1	CDP functional generation number
0xF9F2	SDP functional generation number
0xF9F3	Vendor name
0xF9F4	Semiconductor SDP device name
0xF9F5	Output identifier
0xF9F6	Time since power on
0xF9F8	Firmware update functional generation number
0xFBF0	Device reset command
0xFBF1	Exception reset command
0xFBF2	Store parameters command
0xFBF3	Calculate checksum command
0xFBF4	Load parameters command

17 APPENDIX A: OBJECTS BY FUNCTION

Objects that Define SDOs and PDOs

Device Control and Status Objects

Error Management Objects

Basic Drive Configuration Objects

Basic Motor Configuration Objects

Digital I/O Configuration Objects

Current Loop Configuration Objects

Profile Current Configuration Objects

Interpolated Position Mode Objects

Alternative Control Source Objects

Trace Tool Objects

EtherCAT only Objects

Factor Group Objects

Touch Probe Objects

18 APPENDIX B: OBJECTS BY INDEX ID

This chapter lists the objects in order of index ID. Bold page numbers indicate that the top-level object's general description appears on that page. Regular page numbers indicate that a reference to the object (or one of its sub-objects) appears on that page.

19 APPENDIX C: PDO MAPPING CHART

EtherCAT PDO Mapping Chart

R = Receive (to drive)   T = Transmit (from drive)

Fixed PDO (one slot only)   RxPDO's   TxPDO's   Fixed PDO (one slot only)
High priority slot   0x1C12   0x1C13   High priority slot

0x1700: CSP Mode   0x1600   0x1A00   0x1800
0x1701: CSV Mode   0x1601   0x1A01   0x1801
0x1702: CST Mode   0x1602   0x1A02
0x1703: UV Mode   0x1603   0x1A03
0x1704: CSTCA Mode

Non-fixed PDOs (user mappable)
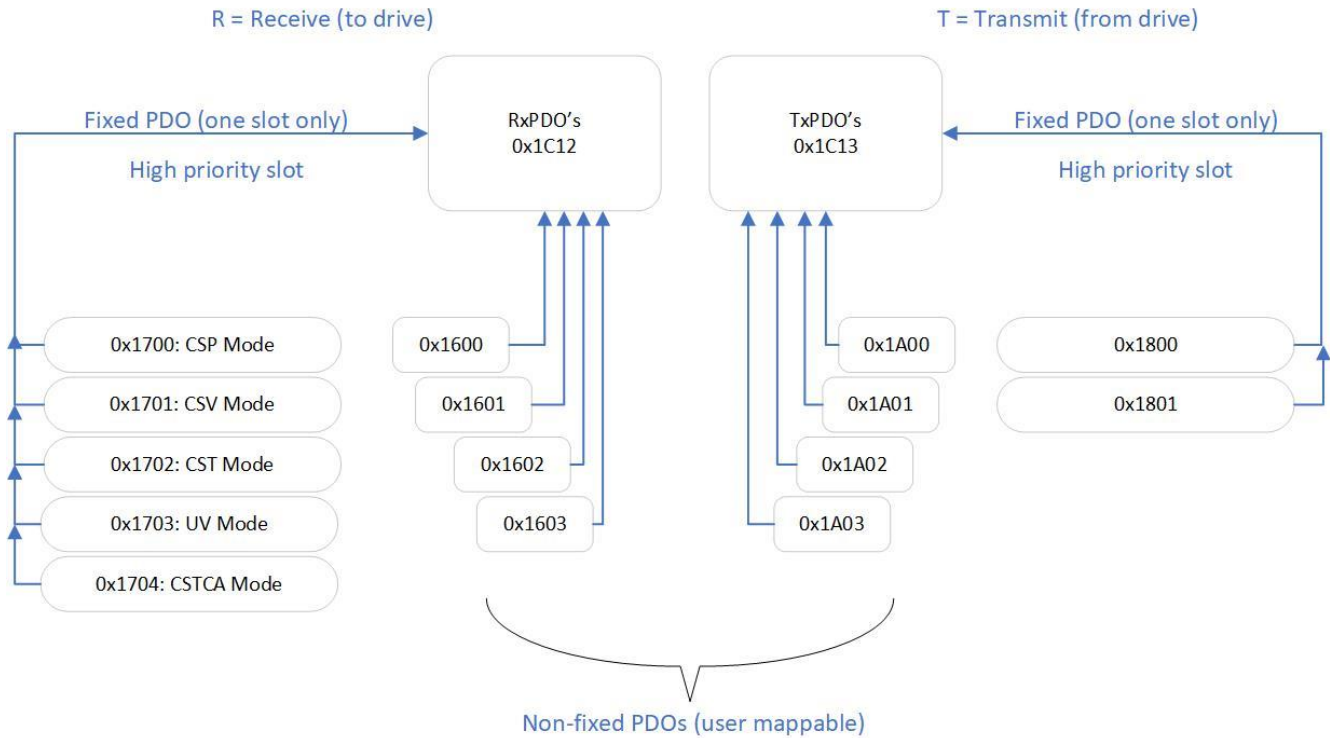
8 non-fixed PDOs that are mappable. 8 per drive.

Pick 1 fixed PDO per axis. Only one. Cannot edit.

CANopen

Programmer's

Manual

P/N 16-01195

Revision 04

August 12, 2021