CANopen Programmer's Manual

The ICR SmartActuator which uses this software is a DISCONTINUED Tolomatic Product. This manual is made available for use with legacy ICR.

Copley Controls Corp.

Part Number 95-00271-000

Revision 5

October, 2008

TABLE OF CONTENTS

ABOUT THIS MANUAL

Overview and Scope

This manual describes the CANopen implementation developed by Copley Controls Corporation for the Accelnet, Xenus, R-Series, and Stepnet amplifiers. 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.

Related Documentation

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

Those interested in Running CAM Tables from RAM (p. 199) should also see the *Copley Camming User Guide.*

Information on Copley Controls Software can be found at:
http://www.copleycontrols.com/Motion/Products/Software/index.html

Comments

Copley Controls Corp. 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 Corporation. Accelnet, Stepnet, Xenus, and CME 2 are registered trademarks of Copley Controls Corporation.

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 Corp. Copley Controls Corp. assumes no responsibility for any errors that may appear in this document.

Product Warnings

WARNING

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

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

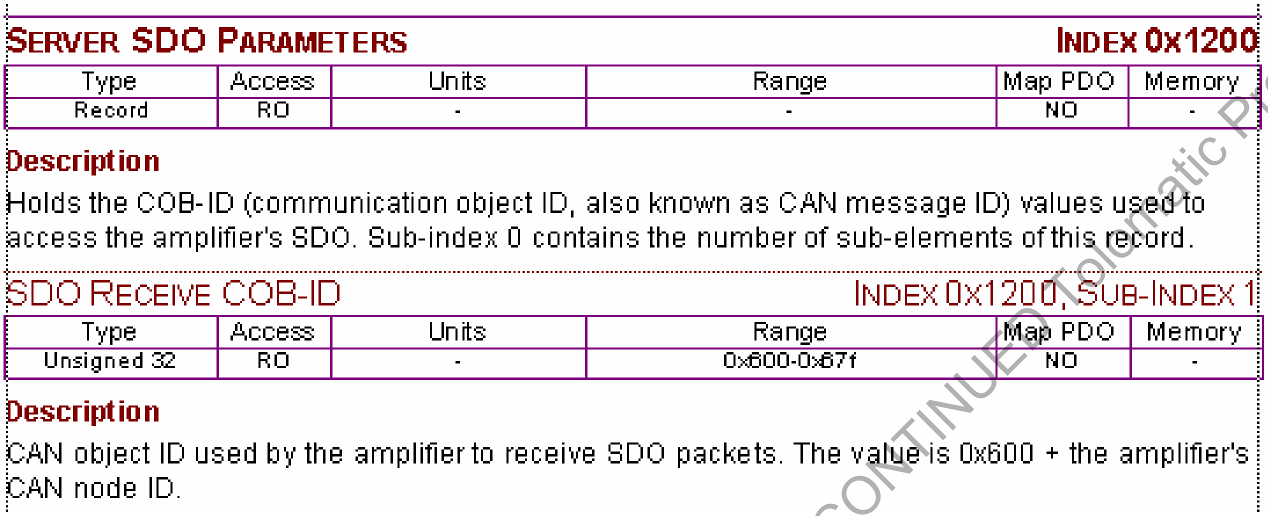
Failure to heed this warning can cause equipment damage, injury, or death.

Revision History

Revision	Date	DECO #	Comments
1.0	Oct, 2002		Initial publication.
2.0	Dec, 2003		Added descriptions of new objects to support stepper mode and profile velocity mode operation, additional homing methods, and amplifier configuration.
2.1	Jan, 2004		Various minor edits and updates.
2.2	March, 2004		Added information about emergency message (EMCY) and memory storage options for objects.
3	June, 2006		Added information on EMCY Message Error Codes (p. 41), a new Camming mode and an object for reading/writing CVM Indexer Program registers (see Alternative Control Sources, p. 191), a new Trace Tool (p. 203), and a new Profile Torque Mode Operation (p. 173). Also, instructions for Ending an Interpolated Position Move (p. 184).
4	June, 2008	16591	Various updates, including Web page references and details on Running CAM Tables from RAM (p. 199).
5	October, 2008	17339	Various updates.

Object Description Conventions

Object descriptions in this manual look like the samples shown below. Each description includes a table of summary information.



SERVER SDO PARAMETERS **INDEX 0x1200**

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

Description

Holds the COB-ID (communication object ID, also known as CAN message ID) values used to access the amplifier'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
Unsigned 32	RO	-	0x600-0x67f	NO	-

Description

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

Sub-Index Object Relationships

This manual describes objects and sub-index objects. Object descriptions are set off by bold type and a heavy separator line. Sub-index object descriptions have regular typeface and a thinner line.

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

Object Summary Description Fields

Field Name	Description
Type	The object type (i.e., Unsigned 32, Integer, String).
Access	The object's access type: • RO for read only • WO for write only • RW for read and write • RC for read and clear
Units	The units used to express the object's value.
Range	The acceptable range of values if less then that specified by Type.
Map PDO	YES if the object can be mapped to a PDO. NO if it cannot. EVENT if the object can be mapped and set to event triggering.
Memory	Some objects can be held in the amplifier's flash memory (F), some in RAM (R), and some in RAM and flash (RF). If an object cannot be stored, or if the object contains sub-index objects, the Memory field contains a dash (-).

CHAPTER

1: INTRODUCTION

This chapter discusses how Copley Controls supports the use of CANopen to provide distributed motion control.

Contents include:

1.1: CAN and CANopen

Contents of this Section

This section describes Copley Controls' use of CANopen and the underlying Controller Area Network (CAN).

Topics include:

Copley Controls Amplifiers in CANopen Networks

Copley’s CANopen Amplifiers

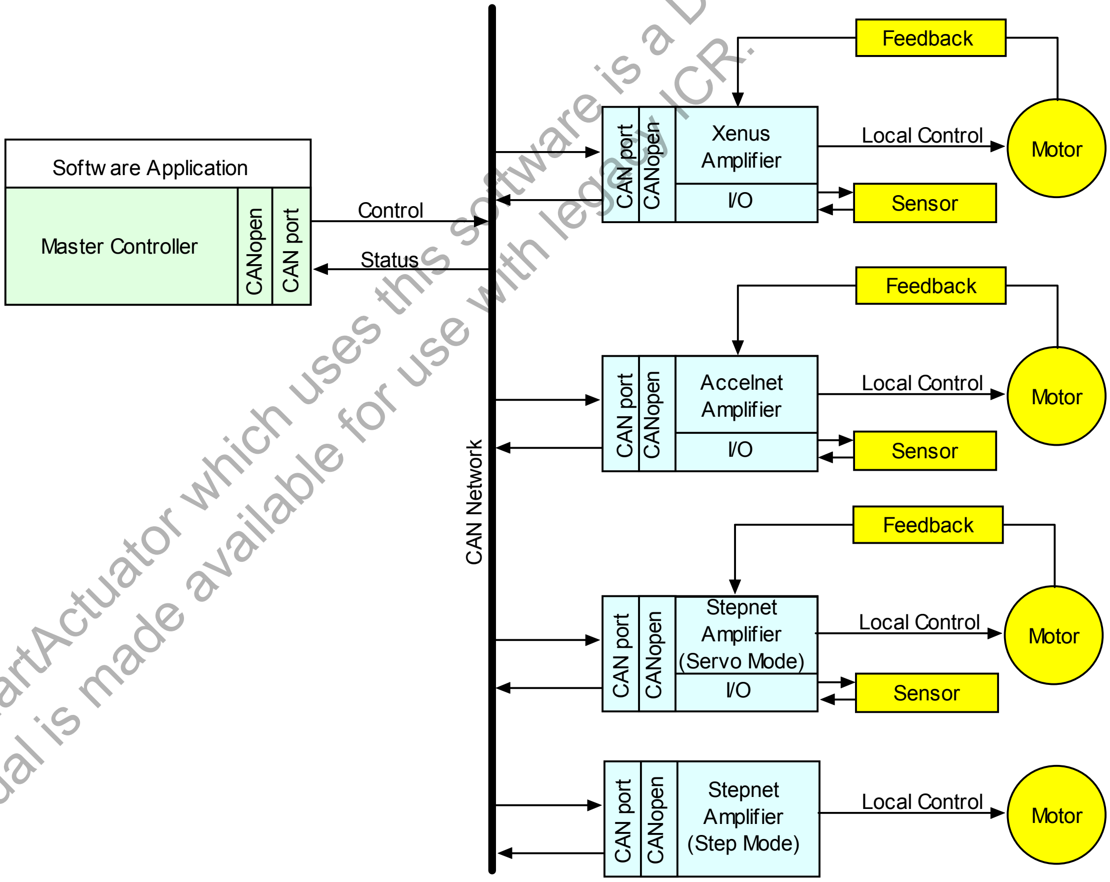
Several lines of Copley Controls amplifiers, including Accelnet, Stepnet, Xenus, and the ruggedized R-Series, offer distributed motion control through support of the Controller Area Network (CAN) and the CANopen network profiles. Using CANopen, the amplifiers can take instruction from a master application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

CAN and 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 amplifiers, 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.



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 and should not be used.)

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 124-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 may be used 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.
- 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:

- They specify methods for packaging multiple CAN messages to send large blocks of data as a single entity.
- They standardize and simplify 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 amplifiers 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

The amplifier's operating mode is set using the Mode Of Operation object (index 0x6060, p. 59).

(The *Profile for Drives and Motion Control* also supports other modes that are not supported by Copley Controls amplifiers at this time.)

1.2: Defining and Accessing CANopen Devices

Contents of this Section

This section describes the objects and methods used to configure and control devices on a CANopen network.

Topics include:

Defining a Device: CANopen Objects and Object Dictionaries

Objects and Dictionaries

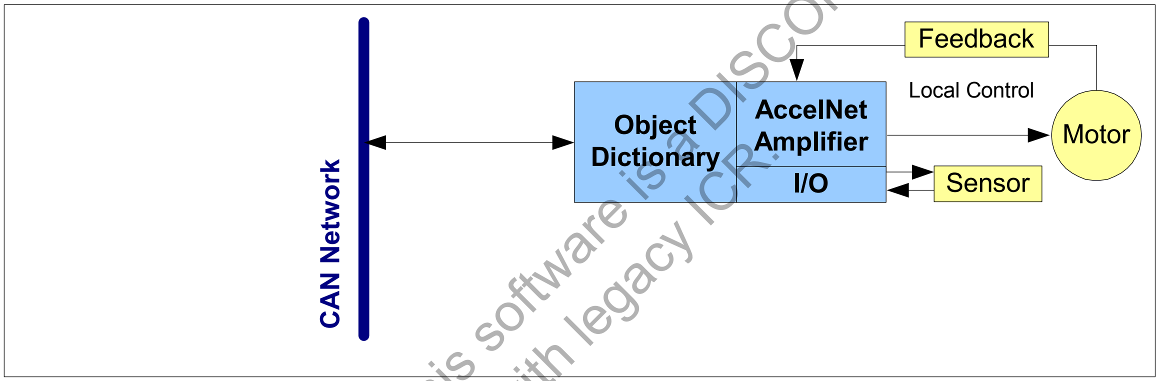
The primary means of controlling a device on a CANopen network is by writing to device parameters, and reading device status information. For this purpose, each device defines a group of parameters that can be written, and status values that can be read. These parameters and status values are collectively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



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.

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 0x6410, p. 81) 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 for future use
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 for further use

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

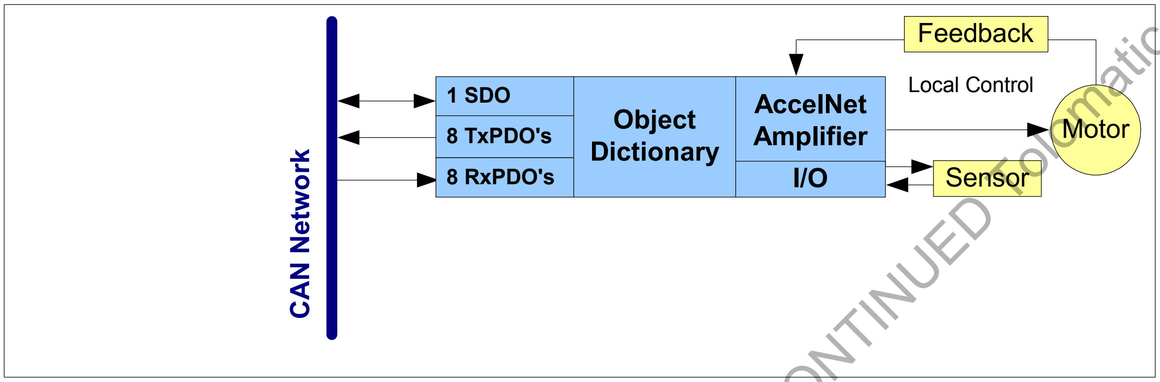
Here are the basic characteristics of PDOs and SDOs.

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, p. 22.	For more information about PDOs, see PDOs: Description and Examples, p. 24.
For help deciding whether to use an SDO or a PDO see SDO vs. PDO: Design Considerations, p. 27.	

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 amplifiers each support 1 SDO and 16 PDOs (eight transmit PDOs and eight receive PDOs).



SDOs: Description and Examples

Overview

Each amplifier 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 amplifier (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 amplifier) 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 amplifier 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 device's receive SDO message identifier. In response, the CANopen master should expect an SDO message whose CAN message ID matches the device's 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

For instance, updating an object that holds an eight-byte long value requires six CAN messages:

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 amplifiers do not require use of the block transfer protocol.

PDOs: Description and Examples

Overview

Each amplifier 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, p. 32, and Transmit PDO Communication Parameters, p. 34).

The first four transmit PDOs and receive PDOs provided in Copley Controls CANopen amplifiers 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:

- **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, p. 40.)
- **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
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.
SYNC message	For synchronous PDOs, the message is transmitted after a specified number of SYNC cycles have occurred.
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 amplifiers 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 amplifier, 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 amplifier 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, p. 187). Used to receive the PVT segments.	4	Trajectory Buffer Status object (index 0x2012, p. 189). 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, p. 45) on the amplifier 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, p. 45) on all but the time-stamp transmitter.
		2	Various status information: Status Word (index 0x6041, p. 55), Manufacturer Status Register object (index 0x1002, p. 56), and Input Pin States (index 0x2190 p. 95). 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.

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. To transfer large amounts of low priority data, use the SDO. Also, if confirmation is absolutely required, use an SDO.
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.	
Asynchronous.	Synchronous or asynchronous. Cyclic or acyclic.	Use PDO when synchronous or broadcast communications are required. 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.
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.	
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.

How to Map (or Remap) a PDO

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.

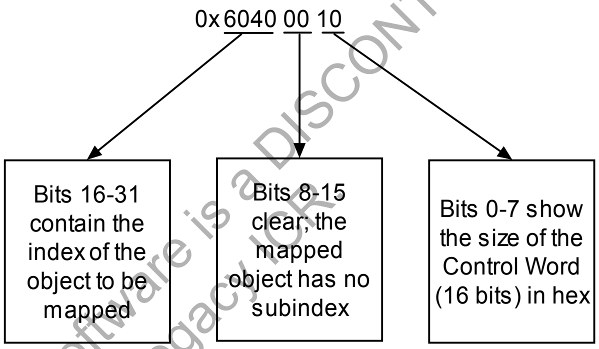
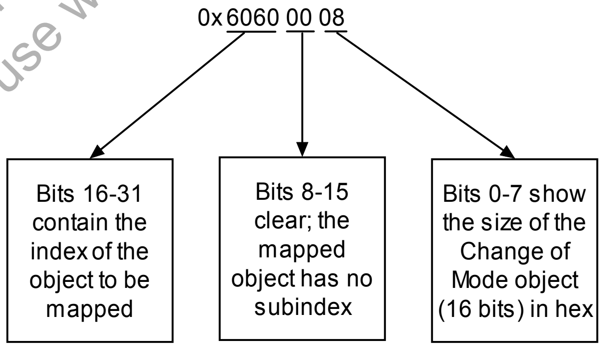
To Map a Receive PDO

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 second receive PDO is mapped to the device’s Control Word object (index 0x6040, p. 54) to receive device state change commands and to the Mode Of Operation object (index 0x6060, p. 59) 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 (16 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.

1.3: Objects that Define SDOs and PDOs

Contents of this Section

This section describes objects and sub-index objects used to configure SDOs and PDOs. They include:

SERVER SDO PARAMETERS INDEX 0X1200

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

Description

Holds the COB-ID (communication object ID, also known as CAN message ID) values used to access the amplifier'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
Unsigned 32	RO	-	0x600-0x67f	NO	-

Description

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

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

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	0x580-0x5ff	NO	-

Description

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

Receive PDO Communication Parameters — Index 0x1400 – 0x1407

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

Description

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 – 7, Sub-Index 1

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	See Default Values, below.	NO	R

Description

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 for future use.
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 + amplifier CAN node ID.
0x1401	0x00000300 + amplifier CAN node ID.
0x1402	0x00000400 + amplifier CAN node ID.
0x1403	0x00000500 + amplifier CAN node ID.
0x1404	0x80000000
0x1405	0x80000000
0x1406	0x80000000
0x1407	0x80000000

PDO Type — Index 0x1400 – 7, Sub-Index 2

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

Description

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	-

Description

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

NUMBER OF MAPPED OBJECTS INDEX 0X1600 – 7, SUB-INDEX 0

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

Description

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 – 7, SUB-INDEX 1 – 4

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	See *Description*, below	NO	R

Description

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.

Transmit PDO Communication Parameters — Index 0x1800 – 0x1807

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

Description

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 – 7, Sub-Index 1

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	See Default Values, below.	NO	R

Description

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 + amplifier CAN node ID.
0x1801	0x00000280 + amplifier CAN node ID.
0x1802	0x00000380 + amplifier CAN node ID.
0x1803	0x00000480 + amplifier CAN node ID.
0x1804	0x80000000
0x1805	0x80000000
0x1806	0x80000000
0x1807	0x80000000

PDO TYPE INDEX 0X1800 – 7, SUB-INDEX 2

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

Description

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	-

Description

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

NUMBER OF MAPPED OBJECTS INDEX 0X1A00 – 7, SUB-INDEX 0

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

Description

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 – 7, Sub-Index 1 – 4

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	See *Description*, below	NO	R

Description

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.

CHAPTER

2: Network Management

This chapter describes the messages, methods, and objects used to manage devices on a CANopen network.

Contents include:

2.1: Network Management Overview

Contents of this Section

This section describes the objects, messages, and methods used to control the CANopen network.

Topics include:

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 amplifiers:

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

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.

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.

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 amplifiers 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 amplifiers can produce the SYNC and high-resolution time stamp messages.

We recommend using an amplifier as the master sync generator. This assures greater timing accuracy and allows the amplifier 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 amplifier. See EMCY Message Error Codes, p. 41.
2	Error register object value See Error Register, p. 62.
3	Reserved for future use (0 for now).
4, 5	Bit mask representing the Copley Controls codes for active error conditions on the amplifier (see EMCY Message Copley-Specific Error Conditions, p. 42).
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 Amplifiers:

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	Amplifier Over Temperature
4300	Motor Temperature Sensor
5080	Amplifier 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 amplifier. The mapped bits have the following meanings:

Bit	Description
0	Output short circuit
1	Amplifier over temperature
2	Amplifier over voltage
3	Amplifier 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	Amplifier internal hardware error (contact Copley Controls customer support)
14	Node guarding error

2.2: Network Management Objects

Contents of this Section

This section describes closely related to network management. They include:

COB-ID Sync Message Index 0x1005

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	See SYNC ID Format, below.	NO	R

Description

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 amplifier is configured as the SYNC message producer. This bit should be set in at most one amplifier on a network.
31	Reserved

Communication Cycle Period Index 0x1006

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	microseconds	-	NO	R

Description

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

An amplifier 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.

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

Guard Time Index 0x100C

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	-	NO	R

Description

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

If the amplifier 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. 45), the amplifier 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

Description

This object gives a multiple of the GUARD Time (index 0x100C, p. 44). The amplifier 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 amplifier 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
Unsigned 32	RW	microseconds	0 - 294,967,295	YES	R

Description

This object holds a time stamp indicating the amplifier'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 amplifier 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 amplifiers to synchronize their clocks across the CANopen network. To enable this feature, one amplifier should be selected as a high-resolution time stamp producer. This amplifier 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 amplifier 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.

PRODUCER HEARTBEAT TIME INDEX 0X1017

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	-	NO	R

Description

This object gives the frequency at which the amplifier 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.

EMERGENCY OBJECT ID INDEX 0X1014

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	-	NO	R

Description

CAN message ID used with the emergency object. See Emergency Messages, p. 40 and the *CANopen Application Layer and Communication Profile (DS 301).*

EMERGENCY OBJECT ID INHIBIT TIME INDEX 0X1015

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	-	NO	R

Description

Inhibit time for the emergency object. See Emergency Messages, p. 40 and the *CANopen Application Layer and Communication Profile (DS 301).*

NETWORK OPTIONS INDEX 0X21B3

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	-	NO	RF

Description

Network options. Configures the amplifier's network.	
CANopen	
Bits	Meaning
0	Must be clear to select CANopen networking.
1-15	Reserved

CHAPTER
3: DEVICE CONTROL, CONFIGURATION, AND STATUS

This chapter describes a wide range of device control, configuration, and status methods and objects.

Contents include:

3.1: Device Control and Status Overview

Contents of this Section

This section describes the objects and functions used to control the status of an amplifier.

Topics include:

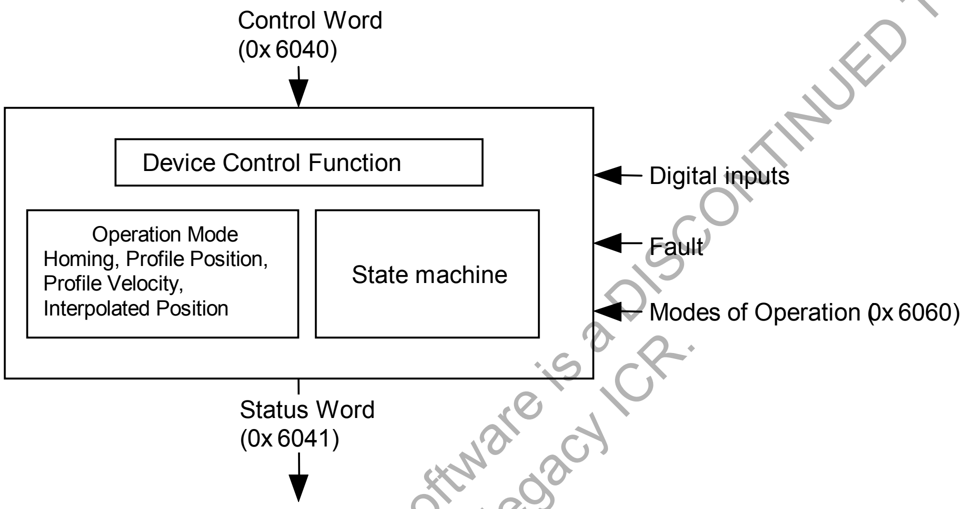
Control Word, Status Word, and Device Control Function

Device Control Function Block

The *CANopen Profile for Drives and Motion Control (DSP 402)* describes control of the amplifier 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. 54) manages device mode and state changes. The Status Word object (index 0x6041, p. 55) identifies the current state of the amplifier. The Mode Of Operation object (index 0x6060, p. 59) sets the amplifier'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. 59), Copley Controls CANopen amplifiers 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.

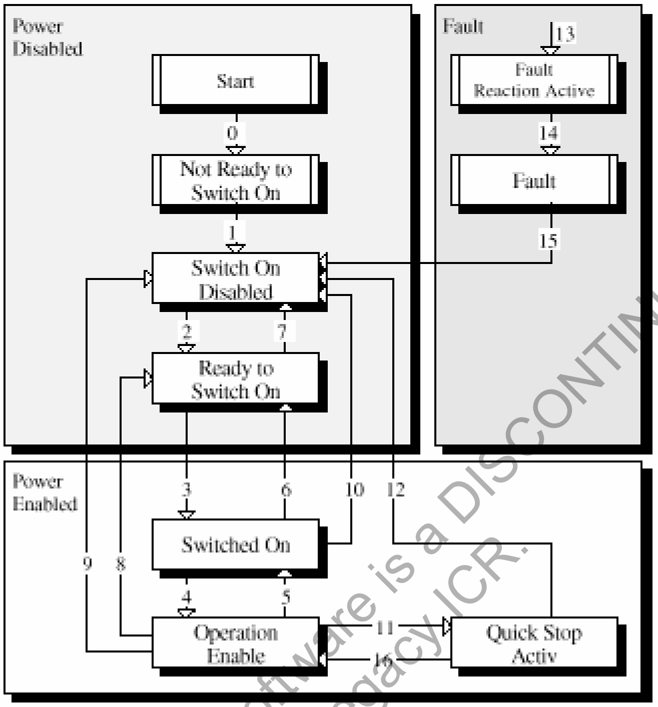
States are described below:

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 amplifier 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 amplifier 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 amplifier. Each transition is numbered and described in the legend below.



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

Continued….

…State Changes Diagram Legend, continued

	From State	To State	Event/Action
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. 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. 58). Action: The drive function is enabled.

3.2: Device Control and Status Objects

Contents of this Section

This section describes the objects used to control the status of an amplifier.

They include:

CONTROL WORD INDEX: 0X6040

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below.	EVENT	R

Description

This object is used to controls the state of the amplifier. It can be used to enable / disable the amplifier 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 amplifier.
1	Enable Voltage. This bit must be set to enable the amplifier.
2	Quick Stop. If this bit is clear, then the amplifier is commanded to perform a quick stop.
3	Enable Operation. This bit must be set to enable the amplifier.
4-6	Operation mode specific. Descriptions appear in the sections that describe the various operating modes. Also see Mode Of Operation (index 0x6060, p. 59).
7	Reset Fault. A low-to-high transition of this bit makes the amplifier attempt to clear any latched fault condition.
8	Halt. If the bit is set, the amplifier will perform a halt.
9-15	Reserved for future use.

STATUS WORD INDEX 0X6041

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	See *Description*, below.	Event	-

Description

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

<table>
<tr><th>Bits</th><th colspan="6">Description</th></tr>
<tr><td>0</td><td colspan="6">Ready to switch on.</td></tr>
<tr><td>1</td><td colspan="6">Switched on.</td></tr>
<tr><td>2</td><td colspan="6">Operation Enabled. Set when the amplifier is enabled.</td></tr>
<tr><td>3</td><td colspan="6">Fault. If set, a latched fault condition is present in the amplifier.</td></tr>
<tr><td>4</td><td colspan="6">Voltage enabled. Set if the amplifier bus voltage is above the minimum necessary for normal operation.</td></tr>
<tr><td>5</td><td colspan="6">Quick Stop. When clear, the amplifier is performing a quick stop.</td></tr>
<tr><td>6</td><td colspan="6">Switch on disabled.</td></tr>
<tr><td>7</td><td colspan="6">Warning. Set if a warning condition is present on the amplifier. Read the Manufacturer Status Register object (index 0x1002, p. 56) for details of what warning is bit indicates.</td></tr>
<tr><td>8</td><td colspan="6">Set if the last trajectory was aborted rather then finishing normally.</td></tr>
<tr><td>9</td><td colspan="6">Remote. Set when the amplifier is being controlled by the CANopen interface. When clear, the amplifier may be monitored through this interface, but some other input source is controlling it. Other input sources include the serial port, amplifier 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 'amplifier 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. 60).</td></tr>
<tr><td>10</td><td colspan="6">Target Reached. This bit is set when the amplifier is finished running a trajectory, and the Position Error (index 0x60F4, p. 116) has been within the Position Tracking Window (index 0x6067, p. 115) for the programmed time. The bit is not cleared until a new trajectory is started.</td></tr>
<tr><td>11</td><td colspan="6">Internal Limit Active. This bit is set when one of the amplifier limits (current, voltage, velocity or position) is active. The specific bits from the Manufacturer Status Register (index 0x1002, p. 56) that cause this bit to be set can be customized by using the mask defined in the Limit Status Mask object (index 0x2184, p. 57).</td></tr>
<tr><td>12-13</td><td colspan="6">The meanings of these bits are operation mode specific:</td></tr>
<tr><td></td><td><b>Bit</b></td><td><b>Profile Position Mode</b></td><td><b>Profile Velocity Mode</b></td><td><b>Profile Torque Mode</b></td><td><b>Homing Mode</b></td><td><b>Interpolated Position Mode</b></td></tr>
<tr><td></td><td>12</td><td>Setpoint acknowledge.</td><td>Speed = 0.</td><td>Reserved</td><td>Homing attained.</td><td>Interpolated pos. mode active.</td></tr>
<tr><td></td><td>13</td><td>Following error.</td><td>Maximum slippage error.</td><td>Reserved.</td><td>Homing error.</td><td>Reserved.</td></tr>
<tr><td></td><td colspan="6">For information on operation modes, see Mode Of Operation (index 0x6060, p. 59).</td></tr>
<tr><td>14</td><td colspan="6">Set when the amplifier 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.</td></tr>
<tr><td>15</td><td colspan="6">Reserved.</td></tr>
</table>

Manufacturer Status Register Index 0x1002

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	See *Description*, below.	EVENT	-

Description

This 32-bit object is a bit-mapped status register with the following fields:

Bit	Description
0	Short circuit detected
1	Amplifier 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
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	Amplifier 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	Amplifier 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. 116) is outside Position Tracking Window (index 0x6067, p. 115).
26	Home switch is active.
27	In motion. This bit is set when the amplifier is finished running a trajectory, and the Position Error (index 0x60F4, p. 116) has been within the Position Tracking Window (index 0x6067, p. 115) for the programmed time. The bit is not cleared until a new trajectory is started.
28	Velocity window. Set if the absolute velocity error exceeds the velocity window value.
29	Phase not yet initialized. Set when the amplifier has not yet initialized its phase while using mode setting 5 (algorithmic phase mode) of Phasing Mode object (0x21C0).
30	Command fault. PWM or other command signal not present.

'STICKY' EVENT STATUS REGISTER INDEX 0X2180

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	-	YES	-

Description

Sticky Amplifier Event Status Register. This read-only parameter is bit-mapped in exactly the same way as the Manufacturer Status Register (index 0x1002, p. 56), but instead of giving the present status of the amplifier, 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. 57), 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	Units	Range	Map PDO	Memory
Unsigned 32	RC	-	-	YES	R

Description

This is a latched version of the Manufacturer Status Register object (index 0x1002, p. 56). Bits are set by the amplifier 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
Unsigned 32	RW	-	-	YES	RF

Description

This parameter defines which bits in the Manufacturer Status Register object (index 0x1002, p. 56) can set the limit bit (bit 11) of the Status Word object (index 0x6041, p. 55). 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.

QUICK STOP OPTION CODE — INDEX 0X605A

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	R

Description

This object defines the behavior of the amplifier when a quick stop command is issued. The following values are defined.

Value	Description
0	Disable the amplifier's outputs
1	Slow down using the normal slow down ramp programmed in Profile Deceleration (index 0x6084, p. 177). When the move has been successfully aborted the amplifier's state will transition to the 'switch on disabled' state.
2	Slow down using the quick stop ramp programmed in Quick Stop Deceleration (index 0x6085, p. 178) 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 amplifier 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
Integer 16	RW	-	See *Description*, below.	NO	R

Description

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

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

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
Integer 16	RW	-	See *Description*, below.	NO	R

Description

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

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

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

Halt Option Code — Index 0x605D

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	R

Description

This object defines the behavior of the amplifier when a halt command is issued. The following values are defined.

Value	Description
0	Disable the amplifier's outputs
1	Slow down using the slow down ramp (i.e. the normal move deceleration value).
2	Slow down using the quick stop ramp.
3	Stop the move abruptly.

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

Mode Of Operation — Index 0x6060

Type	Access	Units	Range	Map PDO	Memory
Integer 8	RW	-	See *Description*, below.	YES	R

Description

This object selects the amplifier'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

The amplifier will not accept other values.

Note that there may be some delay between setting the mode of operation and the amplifier 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
Integer 8	RO	-	See *Description*, below.	EVENT	-

Description

This object displays the current mode of operation. See Mode Of Operation (index 0x6060, p. 59).

DESIRED STATE INDEX 0X2300

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	RF

Description

This object defines what input source controls the amplifier, and what general mode the amplifier 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.
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.
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.
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.
40	In microstepping mode, the amplifier is driven by the CANopen interface.
42	Micro-stepping 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.

3.3: Error Management Objects

Contents of this Section

This section describes objects used to view error status and define error limits and error handling. They include:

Pre-Defined Error Object — Index 0x1003

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

Description

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. 40). 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
Unsigned 8	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
Unsigned 32	RW	-	-	NO	R

Description

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).

Error Register — Index 0x1001

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RO	-	See *Description*, below.	YES	-

Description

This object is a bit-mapped list of error conditions present in the amplifier. 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 amplifier.
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 amplifier is either over or under the amplifier's limits.
3	Temperature error. Either the amplifier or motor is over temperature. Note that the amplifier will only detect a motor over temperature condition if an amplifier input has been configured to detect this condition.
4	Communication error. The amplifier does not presently use this bit.
5-6	Reserved for future use.
7	The following errors cause this bit to be set; Motor phasing error, tracking error, limit switch active.

Tracking Error Window — Index 0x2120

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Counts	0 - 2,147,483,647	YES	RF

Description

Also known as Position Tracking Error Limit. Specifies the maximum absolute Position Error (index 0x60F4, p. 116) 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. 56).Using the Fault Mask object (index 0x2182, p. 63), the tracking error event can be configured to either disable the amplifier immediately, or abort the present move and continue holding position.

FAULT MASK INDEX 0X2182

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	See *Description*, below.	YES	RF

Description

This variable is used to configure which amplifier events cause latching faults. Setting a fault mask bit to 1 causes the associated amplifier 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. 64).

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 amplifier detected corrupted flash data values on startup. The amplifier will remain disabled and indicate a fault condition.
1	Amplifier internal error. This bit is read only and cannot be cleared. It indicates that the amplifier failed its power-on self-test. The amplifier will remain disabled and indicate a fault condition.
2	Short circuit. If set, then the amplifier will latch a fault condition when a short circuit is detected on the motor outputs. If clear, the amplifier will disable its outputs for 100 milliseconds and then re-enable.
3	Amplifier over temperature. If set, this bit will cause an amplifier over temperature condition to act as a latching fault. If clear, the amplifier 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 amplifier to latch a fault condition. If clear, the amplifier 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 amplifier; 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 amplifier is re-enabled when the phasing error is removed.
9	Tracking error. If set, a tracking error will cause the amplifier to latch in the disabled state. If clear, a tracking error will cause the present move to be aborted, but the amplifier 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 amplifier detected an FPGA failure. The amplifier will remain disabled and indicate a fault condition.
12	Command input lost fault. If set: programs the amplifier to latch in the disabled state when the command input is lost. This fault is currently only available on special amplifiers.
13-31	Reserved

LATCHING FAULT STATUS REGISTER INDEX 0X2183

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RC	-	-	YES	R

Description

Bit-mapped to show which latching faults have occurred in the amplifier. When a latching fault has occurred, the fault bit (bit 22) of the Manufacturer Status Register object (index 0x1002, p. 56) is set. The cause of the fault can be read from this register.

To clear a fault condition, write a 1 to the associated bit in this register.

The events that cause the amplifier to latch a fault are programmable. See Fault Mask object (index 0x2182, p. 63) for details.

Latched Faults	
Bit	Fault Description
0	Data flash CRC failure. This fault is considered fatal and cannot be cleared.
1	Amplifier internal error. This fault is considered fatal and cannot be cleared.
2	Short circuit.
3	Amplifier over temperature.
4	Motor over temperature.
5	Over voltage.
6	Under voltage.
7	Feedback fault.
8	Phasing error.
9	Tracking error.
10	Over Current,
11-31	Reserved,

3.4: Basic Amplifier Configuration Objects

Objects described in this section provide access to basic amplifier parameters. They include:

They include:

Device Type Index 0x1000

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	See *Description*, below.	NO	-

Description

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 amplifier 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 amplifiers, this value is 0x0006, indicating that Copley Controls supports servo and stepper devices.

Device Name Index 0x1008

Type	Access	Units	Range	Map PDO	Memory
Visible String	RO	-	-	NO	-

Description

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

Hardware Version String Index 0x1009

Type	Access	Units	Range	Map PDO	Memory
String	RO	-	-	NO	-

Description

Describes amplifier hardware version.

Store Parameters Index 0x1010

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

Description

Allows the current values programmed into the amplifier'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.

Store All Objects Index 0x1010, Sub-Index 1

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	-	NO	R

Description

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 amplifier reset.

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

STORE COMMUNICATION PARAMETERS INDEX 0X1010, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	-	NO	R

Description

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 amplifier reset.

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

STORE DEVICE PROFILE PARAMETERS INDEX 0X1010, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	-	NO	R

Description

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 amplifier reset.

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

STORE MANUFACTURER SPECIFIC PARAMETERS INDEX 0X1010, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	-	NO	R

Description

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 amplifier reset.

Objects in the category are the objects with indexes in the range 0x2000 – 0x5FFF.

SOFTWARE VERSION NUMBER INDEX 0X100A

Type	Access	Units	Range	Map PDO	Memory
Visible String	RO	-	-	NO	-

Description

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

IDENTITY OBJECT INDEX 0X1018

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

Description

This object can uniquely identify an amplifier 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
Unsigned 32	RO	-	0x000000AB	NO	-

Description

A unique identifier assigned to Copley Controls Corp.

The value of this identifier is fixed at: 0x000000AB

PRODUCT CODE INDEX 0X1018, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	See *Description*, below.	NO	-

Description

Identifies the specific amplifier model. Also known as Amplifier Hardware Type. Identical to Amplifier Type Code (index 0x6510, Sub-Index 13, p. 76). The currently defined values for this object are:

Value	Product
0x0200	ACM: Accelnet Module.
0x0201	XSL: Xenus Panel (obsolete).
0x0203	ACP: Accelnet Panel (obsolete).
0x0206	XSL-R: Xenus Panel, resolver version.
0x0207	XSL: Xenus Panel.
0x0209	ACJ: Accelnet Micro Panel.
0x020b	ACP: Accelnet Panel.
0x020c	ACK: Accelnet Micro Module.
0x020e	Special.
0x020f	Special.
0x0210	ACJ-S: Accelnet Micro Panel, analog encoder version.
0x0240	STM: Stepnet Module.
0x0242	STP: Stepnet Panel.
0x0243	STL: Stepnet Micro Module.
0x0300	ASP: Accelnet Panel, dual axis.
0x0310	XSJ(S): Xenus Micro Panel.
0x0320	XTL-R: Xenus Panel, resolver version.
0x0330	XTL(S): Xenus Panel.
0x0340	XSJ-R: Xenus Micro Panel, resolver version
0x0380	AEP: Accelnet EtherCat Panel.
0x0350	STX: Stepnet AC Panel
0x03a0	ADP: Accelnet Panel

REVISION NUMBER INDEX 0X1018, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	-	NO	-

Description

Identifies the revision of the CANopen interface.

SERIAL NUMBER INDEX 0X1018, SUB-INDEX 4

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	-	NO	-

Description

The amplifier's serial number. Holds the same value as Amplifier Serial Number (index 0x6510, Sub-Index 1, p. 74).

AMPLIFIER NAME INDEX 0X21A0

Type	Access	Units	Range	Map PDO	Memory
Visible string	RW	-	-	NO	F

Description

This object may be used to assign a name to an amplifier. The data written here is stored to flash memory and is not used by the amplifier. 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.

MISC AMPLIFIER OPTIONS REGISTER INDEX 0X2420

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	-	YES	RF

Description

Miscellaneous Amplifier Options Register. Bit-mapped as follows:

Bit	Description
0	If set, input pins 1, 2, and 3 are pulled high on the amplifier. If clear the pins are not pulled up. This option is only available on the Junus amplifier.
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.
3-31	Reserved.

CANOPEN NETWORK CONFIGURATION INDEX 0X21B0

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below.	NO	RF

Description

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

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 amplifier 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 CAN Node ID (index 0x21B1, p. 72).
11	This bit is ignored on amplifiers that do not have an address switch. On amplifiers with an address switch, setting this bit programs the amplifier 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
4	125,000
5	50,000
6	20,000
7-15	Reserved for future use

INPUT MAPPING FOR CAN NODE ID INDEX 0X21B1

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	See *Description*, below.	NO	F

Description

When the CANopen Network Configuration object (index 0x21B0, p. 71) 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 amplifier 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.

CAN ID SELECTION SWITCH VALUE INDEX 0X2197

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	-	0 - 15	YES	-

Description

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

MULTI-MODE PORT CONFIGURATION INDEX 0X2241

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	-	YES	RF

Description

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.

Supported Drive Modes Index 0x6502

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	See *Description*, below.	NO	-

Description

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

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.
1	Profile velocity mode.
3	Profile torque mode.
5	Homing mode.
6	Interpolated Position Mode.

The current version of amplifier 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 0x00000061.

Future versions of Copley Controls CANopen amplifier firmware might support additional operating modes. If so, those versions will return additional values.

Amplifier Model Number Index 0x6503

Type	Access	Units	Range	Map PDO	Memory
Visible String	RO	-	-	NO	-

Description

This ASCII string gives the amplifier model number.

Amplifier Manufacturer Index 0x6504

Type	Access	Units	Range	Map PDO	Memory
Visible String	RO	-	-	NO	-

Description

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

Manufacturer's Web Address Index 0x6505

Type	Access	Units	Range	Map PDO	Memory
Visible String	RO	-	-	NO	-

Description

This ASCII string gives the web address of Copley Controls.

Amplifier Data Index 0x6510

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

Description

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

AMPLIFIER SERIAL NUMBER INDEX 0X6510, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	-	-	NO	-

Description

Gives the amplifier serial number.

AMPLIFIER DATE CODE INDEX 0X6510, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Visible string	RO	-	-	NO	-

Description

Date of manufacture of the amplifier.

AMPLIFIER PEAK CURRENT INDEX 0X6510, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	NO	-

Description

The amplifier's peak current rating.

Amplifier Continuous Current

Index 0x6510, Sub-Index 4

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	NO	-

Description

The amplifier's continuous current rating.

Amplifier Peak Current Time

Index 0x6510, Sub-Index 5

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	milliseconds	-	NO	-

Description

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

Amplifier Maximum Voltage

Index 0x6510, Sub-Index 6

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.1 volts	-	NO	-

Description

Maximum bus voltage rating for amplifier.

Amplifier Minimum Voltage

Index 0x6510, Sub-Index 7

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.1 volts	-	NO	-

Description

Minimum bus voltage rating for amplifier.

Amplifier Voltage Hysteresis

Index 0x6510, Sub-Index 8

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.1 volts	-	NO	-

Description

Hysteresis for maximum bus voltage cut-out.

Amplifier Maximum Temperature

Index 0x6510, Sub-Index 9

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	degrees centigrade	-	NO	-

Description

Temperature limit for amplifier.

Amplifier Temperature Hysteresis

Index 0x6510, Sub-Index 10

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	degrees centigrade	-	NO	-

Description

Hysteresis value for amplifier over temperature cut-out.

Amplifier Current Loop Period

Index 0x6510, Sub-Index 11

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	10 nanoseconds	-	NO	-

Description

Current loop update period in 10-nanosecond units.

Amplifier Servo Loop Period Index 0x6510, Sub-Index 12

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	-	-	NO	-

Description

Servo loop update period as a multiple of the current loop period.

Amplifier Type Code Index 0x6510, Sub-Index 13

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	-	See *Description*, below.	NO	-

Description

Identifies the specific amplifier model. Also known as Amplifier Hardware Type. Identical to Product Code (index 0x1018, Sub-index 2, p. 69). The currently defined values for this object are:

Value	Product
0x0200	ACM: Accelnet Module.
0x0201	XSL: Xenus Panel (obsolete).
0x0203	ACP: Accelnet Panel (obsolete).
0x0206	XSL-R: Xenus Panel, resolver version.
0x0207	XSL: Xenus Panel.
0x0209	ACJ: Accelnet Micro Panel.
0x020b	ACP: Accelnet Panel.
0x020c	ACK: Accelnet Micro Module.
0x020e	Special.
0x020f	Special.
0x0210	ACJ-S: Accelnet Micro Panel, analog encoder version.
0x0240	STM: Stepnet Module.
0x0242	STP: Stepnet Panel.
0x0243	STL: Stepnet Micro Module.
0x0300	ASP: Accelnet Panel, dual axis.
0x0310	XSJ(S): Xenus Micro Panel.
0x0320	XTL: Xenus Panel, resolver version.
0x0330	XTL(S): Xenus Panel.
0x0340	XSJ-R: Xenus Micro Panel, resolver version
0x0380	AEP: Accelnet EtherCat Panel.
0x0390	AMP: Accelnet Macro Panel.
0x0350	STX: Stepnet AC Panel
0x03a0	ADP: Accelnet Panel

Current Corresponding to Max A/D Reading Index 0x6510, Sub-Index 14

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	NO	F

Amplifier current corresponding to maximum A/D reading.

Voltage Corresponding to Max A/D Reading Index 0x6510, Sub-Index 15

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.1 volts	-	NO	F

Amplifier voltage corresponding to maximum A/D reading.

Analog Input Scaling Factor Index 0x6510, Sub-Index 16

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	-	-	NO	F

Amplifier analog input scaling factor.

Amplifier Minimum PWM Off Time Index 0x6510, Sub-Index 17

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	10 ns	-	NO	F

This fixed amplifier 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 0x6510, Sub-Index 18

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	CPU cycles	-	NO	F

This fixed amplifier 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 0x6510, Sub-Index 19, p. 77).

PWM Dead Time At Zero Current Index 0x6510, Sub-Index 19

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	CPU cycles	-	NO	F

This fixed amplifier 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 0x6510, Sub-Index 18 p. 77).

Peak Current Internal Regen Resistor Index 0x6510, Sub-Index 20

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	NO	F

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

Continuous Current Internal Regen Resistor Index 0x6510, Sub-Index 21

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	NO	F

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

Time At Peak Current Internal Regen Resistor Index 0x6510, Sub-Index 22

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	ms	-	NO	F

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

Analog Encoder Scaling Factor Index 0x6510, Sub-Index 23

Type	Access	Units	Range	Map PDO	Memory
Integer 16	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 0x6510, Sub-Index 24

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	-	NO	F

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 0x6510, Sub-Index 25

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	-	NO	F

Returns the number of axis implemented by this amplifier.

Internal Regen Current

Index 0x6510, Sub-Index 26

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	mA	-	NO	F

Amplifier internal maximum regen current.

FPGA Image Version

Index 0x6510, Sub-Index 27

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	-	NO	R

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

Secondary Firmware Version

Index 0x6510, Sub-Index 28

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	-	-	NO	R

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

Firmware Version Number (Extended)

Index 0x2422

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	-	-	NO	F

Description

Firmware Version Number (extended). The upper 16 bits give the same major/minor version number as Firmware Version Number (index 0x6510, Sub-Index 24, p. 77). 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
Unsigned 32	RO	-	-	NO	-

Description

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
Unsigned 16	RW	-	See *Description*, below.	NO	RF

Description

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.
8	Status bit, set when bus clamping is active.

.

3.5: Basic Motor Configuration Objects

Contents of this Section

Objects described in this section provide access to basic motor parameters. They include:

MOTOR MODEL NUMBER INDEX 0X6403

Type	Access	Units	Range	Map PDO	Memory
Visible String	RW	-	-	NO	F

Description

The motor's model number.

MOTOR MANUFACTURER INDEX 0X6404

Type	Access	Units	Range	Map PDO	Memory
Visible String	RW	-	-	NO	F

Description

The motor's manufacturer name.

MOTOR DATA INDEX 0X6410

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

Description

This record holds a variety of motor parameters.

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

MOTOR TYPE INDEX 0X6410, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

Defines the type of motor connected to the amplifier:

Type	Description
0	Rotary motor.
1	Linear motor.

MOTOR POLE PAIRS INDEX 0X6410, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 - 32,767	NO	F

Description

Number of motor pole pairs (electrical phases) 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 0x6410, Sub-Index 3

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

Defines the direction of the motor wiring:

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

Hall Sensor Type

Index 0x6410, Sub-Index 4

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

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.

Hall Sensor Wiring

Index 0x6410, Sub-Index 5

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

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

Hall Offset — Index 0x6410, Sub-Index 6

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	degrees	-360 - 360	NO	F

Description

Offset angle to be applied to the Hall sensors.

Motor Resistance — Index 0x6410, Sub-Index 7

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.01 Ohm	0 - 32,767	NO	F

Description

Motor winding resistance, in 0.01-Ohm units.

Motor Inductance — Index 0x6410, Sub-Index 8

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.01 milliHenry	0 - 32,767	NO	F

Description

Motor winding inductance, in 0.01-milliHenry units.

Motor Inertia — Index 0x6410, Sub-Index 9

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Rotary: 0.000001 Kg / cm^2 Linear: 0.0001 Kg.	0 - 2,147,483,647	NO	F

Description

Motor inertia.

Motor Back EMF — Index 0x6410, Sub-Index 10

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Rotary: 0.01 V/KRPM Linear: 0.01 V/mps	0 - 2,147,483,647	NO	F

Description

Motor back-EMF constant.

Motor Maximum Velocity — Index 0x6410, Sub-Index 11

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	0.1 counts / second	0 - 500,000,000	NO	F

Description

Maximum motor velocity.

Motor Torque Constant — Index 0x6410, Sub-Index 12

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Rotary: 0.001 Nm / Amp Linear: 0.00001 N.	0 - 2,147,483,647	NO	F

Description

Motor Torque (Force) constant.

Motor Peak Torque — Index 0x6410, Sub-Index 13

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Rotary: 0.001 Nm Linear: 0.00001 N	0 - 2,147,483,647	NO	F

Description

Motor Peak Torque (Force).

Motor Continuous Torque — Index 0x6410, Sub-Index 14

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Rotary: 0.001 Nm/Amp Linear: 0.00001 N/Amp	0 - 2,147,483,647	NO	F

Description

Motor Continuous Torque (Force).

Motor Temperature Sensor — Index 0x6410, Sub-Index 15

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

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

Motor Has a Brake — Index 0x6410, Sub-Index 16

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

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

Motor Stopping Time — Index 0x6410, Sub-Index 17

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	milliseconds	0 - 10,000	NO	F

Description

Also known as Brake/Stop Delay Time. When the amplifier 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 0x6410, Sub-Index 19, p. 85).

Motor Brake Delay — Index 0x6410, Sub-Index 18

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	milliseconds	0 - 10,000	NO	F

Description

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

Motor Brake Velocity

Index 0x6410, Sub-Index 19

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	0.1 counts / second	0 - 500,000,000	NO	F

Description

During the Motor Stopping Time (index 0x6410, Sub-Index 17, p. 84), if the motor's actual velocity falls below this value the brake output is activated immediately.

Encoder Type Code

Index 0x6410, Sub-Index 20

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

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

Value	Description
0	Primary incremental quadrature encoder.
1	No encoder.
2	Analog encoder.
3	Multi-mode port incremental quadrature encoder
4	Analog Halls used for position feedback.
5	Resolver input.
6	Digital Halls.
7	Analog encoder, special.
8	Yaskawa Sigma-Mini SGMM.
9	Panasonic Minas-A.
10	SPI Command
11	SSI
12	EnDat 2.2

Encoder Units

Index 0x6410, Sub-Index 21

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

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

Value	Description
0	microns
1	nanometers
2	millimeters

Motor Encoder Direction

Index 0x6410, Sub-Index 22

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	F

Description

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

Motor Counts/Rev — Index 0x6410, Sub-Index 23

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts / rev	0 - 2,147,483,647	NO	F

Description

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 0x6410, Sub-Index 24

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	encoder units / count	0 - 32,767	NO	F

Description

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

Motor Electrical Distance — Index 0x6410, Sub-Index 25

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	encoder units / cycle	0 - 2,147,483,647	NO	F

Description

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

Reserved — Index 0x6410, Sub-Index 26-27

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

Description

Reserved.

Analog Encoder Shift — Index 0x6410, Sub-Index 28

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	0 -10	NO	F

Description

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 0x6410, Sub-Index 29

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	microsteps	-	NO	F

Description

Microsteps per revolution for microstepping motors.

LOAD ENCODER TYPE INDEX 0X6410, SUB-INDEX 30

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below.	NO	F

Description

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 0X6410, SUB-INDEX 31

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below.	NO	F

Description

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

LOAD ENCODER RESOLUTION INDEX 0X6410, SUB-INDEX 32

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	encoder units / count	0 - 2,147,483,647	NO	F

Description

Only used with linear motors. Also known as Position Encoder Resolution. Number of *Encoder Units* / encoder count. For information, see Encoder Units (index 0x6410, Sub-Index 21, p. 85).

BI-QUAD FILTER COEFFICIENTS INDEX 0X6410, SUB-INDEX 33

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

Description

Reserved.

NUMBER OF RESOLVER CYCLES/MOTOR REV INDEX 0X6410, SUB-INDEX 34

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	-	-	-	NO	F

Description

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

MOTOR ENCODER WRAP INDEX 0X2220

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Counts	-	NO	RF

Description

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
Integer 32	RW	Counts	-	NO	RF

Description

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
Integer 32	RW	-	-	NO	F

Description

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.

Load Encoder Options Index 0x2223

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	-	-	NO	F

Description

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.

Phasing Mode Index 0x21C0

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	NO	RF

Description

Controls the mechanism used by the amplifier to compute the motor phasing angle. Determines what inputs the amplifier 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 amplifier's 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 amplifier enters a state machine used to initialize its phase. While the amplifier is performing this operation, bit 29 of the Manufacturer Status Register (0x1002) is set.

At the start of the phase init algorithm the amplifier will wait to be enabled. Once enabled, the main algorithm will start. If the amplifier 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 amplifier 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 amplifier 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 amplifier 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
Unsigned 16	RW	0.01 amps	-	YES	RF

Description

See Algorithmic Phase Init Mode Details (p. 89).

Algorithmic Phase Initialization Timeout Index 0x21C3

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	-	YES	RF

Description

See Algorithmic Phase Init Mode Details (p. 89).

Algorithmic Phase Initialization Config Index 0x21C4

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	-	YES	RF

Description

See Algorithmic Phase Init Mode Details (p. 89). 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 Hall Offset (index 0x6410, Sub-Index 6, p. 83), as the initial angle for the first attempt.
3-15	Reserved.

3.6: Real-time Amplifier and Motor Status Objects

Contents of this Section

This section describes the following objects:

ANALOG/DIGITAL REFERENCE INPUT VALUE INDEX 0X2200

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	millivolts	-	YES	-

Description

Most recent value read from the reference A/D input (millivolts). Available on certain amplifiers.

HIGH VOLTAGE REFERENCE INDEX 0X2201

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.1 volts	-	YES	-

Description

The voltage present on the high-voltage bus.

AMPLIFIER TEMPERATURE INDEX 0X2202

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	degrees centigrade	-	YES	R

Description

The amplifier temperature.

SYSTEM TIME INDEX 0X2141

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RO	milliseconds	-	YES	R

Description

Time since startup.

WINDING A CURRENT INDEX 0X2203

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

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
Integer 16	RO	0.01 amps	-	YES	-

Description

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
Integer 16	RO	millivolts	-	YES	-

Description

The voltage present on the analog feedback, sine input (millivolts). Not available on all amplifiers.

COSINE FEEDBACK VOLTAGE INDEX 0X2206

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	millivolts	-	YES	-

Description

Voltage present on the analog feedback, cosine input (millivolts). Available on certain amplifiers.

A/D OFFSET VALUE INDEX 0X2207

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	millivolts	-	YES	-

Description

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 amplifier performs on itself while running.

CURRENT OFFSET A INDEX 0X2210

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

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
Integer 16	RO	0.01 amps	-	YES	-

Description

A calibration offset value, calculated at startup, and applied to the winding B current reading..

MOTOR PHASE ANGLE INDEX 0X2260

Type	Access	Units	Range	Map PDO	Memory
Integer 16	R	degrees	0 - 360	YES	R

Description

Motor phase angle, derived from motor commutation.

MOTOR PHASE ANGLE INDEX 0X2262

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	degrees	0 - 360	YES	R

Description

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
Integer 16	RO	degrees	0 - 360	YES	R

Description

For feedback types, such as resolver, that can also calculate phase angle information. This parameter allows the phase information to be read directly.

HALL STATE INDEX 0X2261

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	-	0 - 7	YES	-

Description

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.

3.7: Digital I/O Configuration Objects

Contents of this Section

This section describes the following objects:

Input Pin States — Index 0x2190

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	R0	-	See *Description*, below	EVENT	-

Description

The 16-bit value returned by this command gives the current state (high/low) of the amplifier'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 Config Register — Index 0x2191

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below	YES	RF

Description

Some amplifiers have one or more pull-up resistors associated with their general-purpose input pins. On these amplifiers, 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 amplifier. 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 amplifiers 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	Units	Range	Map PDO	Memory
Array: Unsigned 16	RW	-	-	YES	-

Description

This object consists of N identical sub-elements, where N is the number of input pins available on the amplifier. 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
Unsigned 16	RW	-	See *Description*, below	YES	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 amplifier on the rising edge of the input.
3	Reset the amplifier on the falling edge of the input.
4	Positive side limit switch. Active high. See Misc Amplifier Options Register (index 0x2420, p. 70).
5	Positive side limit switch. Active low. See Misc Amplifier Options Register (index 0x2420, p. 70).
6	Negative side limit switch. Active high. See Misc Amplifier Options Register (index 0x2420, p. 70).
7	Negative side limit switch. Active low. See Misc Amplifier Options Register (index 0x2420, p. 70).
8	Motor temperature sensor. Active high.
9	Motor temperature sensor. Active low.
10	Disable amplifier when high. Clear latched faults on low to high transition.
11	Disable amplifier when low. Clear latched faults on high to low transition.
12	Reset on rising edge. Disable amplifier when high.
13	Reset on falling edge. Disable amplifier when low.
14	Home switch. Active high.
15	Home switch. Active low.
16	Disable amplifier when high.
17	Disable amplifier when low.
19	PWM synchronization. Only for high speed inputs; see amplifier 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	-	-	YES	-

Description

This object consists of N identical sub-index objects, where N is the number of input pins available on the amplifier. (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
Unsigned 16	RW	milliseconds	0 - 10,000	YES	RF

Description

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 amplifier recognizes the state.

RAW INPUT PIN VALUE INDEX 0X2196

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	See *Description*, below.	YES	-

Description

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. 95).

OUTPUT PIN CONFIGURATION INDEX 0X2193

Type	Access	Units	Range	Map PDO	Memory
Array		-	-	YES	RF

Description

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.

OUTPUT PIN CONFIGURATION INDEX 0X2193, SUB-INDEX 1-N

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

The values programmed into these objects allow the amplifier's digital outputs to be driven by internal amplifier events, or externally driven.

Each output configuration consists of a 16-bit configuration word (bits 0-15), followed by a variable number of words (0-4), 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 amplifier's Manufacturer Status Register object (index 0x1002, p. 56). 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 amplifier's Latched Event Status Register (index 0x2181, p. 57). 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 amplifier's Trajectory Generator Status object (index 0x2252, p. 175). 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.
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
Unsigned 16	RW	-	See *Description*, below.	EVENT	R

Description

When read, this parameter gives the active/inactive state of the amplifier'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 amplifier. 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. 98 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.

DIGITAL CONTROL INPUT CONFIGURATION INDEX 0X2320

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below.	NO	RF

Description

Defines the configuration of the digital control inputs when the amplifier 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
Integer 32	RW	See *Description*, below.	See *Description*, below.	YES	RF

Description

When the amplifier 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
Unsigned 32	RO	-	See *Description*, below.	EVENT	-

Description

This object gives the present value of the digital inputs of the amplifier. 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 amplifier in the same ordering as Input Pin States (index 0x2190, p. 95).

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	Amplifier 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. 95).

3.8: Xenus Regen Resister Objects

Contents of this Section

This section describes the following objects:

Xenus Regen Resister Resistance Index 0x2150

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	0.01 Ω	-	NO	RF

Description

Regen resister resistance.

Xenus Regen Resister Continuous Power Index 0x2151

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	watts	-	NO	RF

Description

Regen resister, continuous power.

Xenus Regen Resister Peak Power Index 0x2152

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	watts	-	NO	RF

Description

Regen resister, peak power.

Xenus Regen Resister Peak Time Index 0x2153

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	-	NO	RF

Description

Regen resister, peak time.

Xenus Regen Resister Turn-On Voltage Index 0x2154

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	0.1 Vdc	-	NO	RF

Description

Regen resister, turn-on voltage.

Xenus Regen Resister Turn-Off Voltage Index 0x2155

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	0.1 Vdc	-	NO	RF

Description

Regen resister, turn-off voltage.

XENUS REGEN RESISTER MODEL STRING INDEX 0X2156

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

Description

Regen resister model number string.

XENUS REGEN RESISTER STATUS INDEX 0X2157

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	See *Description*, below.	YES	-

Description

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 resister settings or the internal amplifier protections.
3-15	Reserved for future use.

CHAPTER

4: CONTROL LOOP CONFIGURATION

This chapter describes nested control loop model used by Copley Controls amplifiers to control the position of the motor.

Contents include:

4.1: Control Loop Configuration Overview

Contents of this Section

This section provides an overview of the control loops.

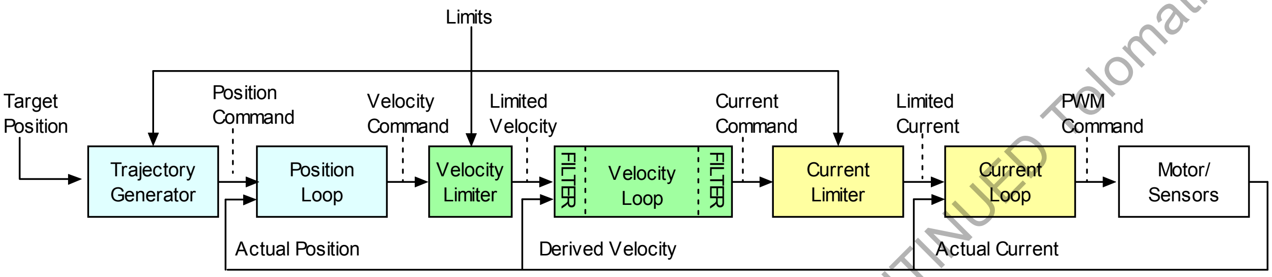
Topics include:

Nested Position, Velocity, and Current Loops

Nesting of Control Loops and Modes

Copley Controls amplifiers use up to three nested control loops - current, velocity, and position - to control a motor in three associated operating modes.

In position mode, the amplifier 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.



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

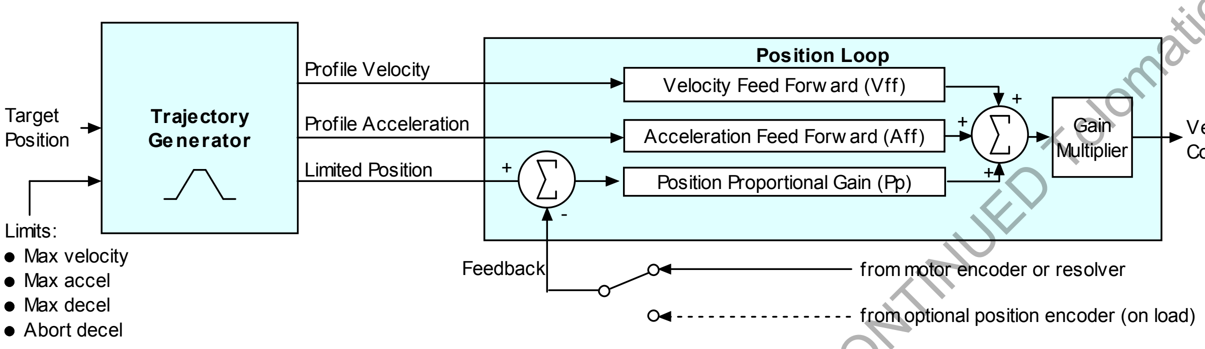
These loops (and servo control loops in general) share several common attributes:

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 amplifier 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 amplifier.

The Position Loop

Position Loop Diagram

The CANopen master provides a target position to the amplifier'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.



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	157
	Homing Speeds / 0x6099	The sub-index objects of 0x6099 hold the two velocities (fast and slow) used when homing.	158
	Homing Acceleration / 0x609A	Defines the acceleration used for all homing moves.	158
	Home Offset / 0x607C	Used in homing mode as an offset between the home sensor position and the zero position.	158
Profile Position	Motion Profile Type / 0x6086	Selects the type of trajectory profile to use. Choices are trapezoidal, S-curve, and velocity.	178
	Target Position / 0x607A	Destination position of the move.	175
	Profile Velocity / 0x6081	The velocity that the trajectory generator attempts to achieve when running in position profile mode.	176
	Profile Acceleration / 0x6083	Acceleration that the trajectory generator attempts to achieve when running in position profile mode	177
	Profile Deceleration / 0x6084	Deceleration that the trajectory generator attempts to achieve at the end of a trapezoidal profile when running in position profile mode.	176
	Trajectory Jerk Limit / 0x2121	Defines the maximum jerk (rate of change of acceleration) for use with S-curve profile moves.	175
Interpolated Position	IP move segment command / 0x2010	Used to send PVT segment data and buffer commands when running in interpolated position mode.	187

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.	114
Instantaneous Commanded Acceleration / 0x2251	Acceleration to which the position loop's acceleration feed forward gain is applied.	114
Position Command Value / 0x6062	Motor position (in units of counts) to which the amplifier is currently trying to move the axis.	114

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. 114).

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. 116) as the difference between the Position Actual Value and the Position Command 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 0x60FB, sub-index 1-6, p. 117).

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 / 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.	120
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.	116

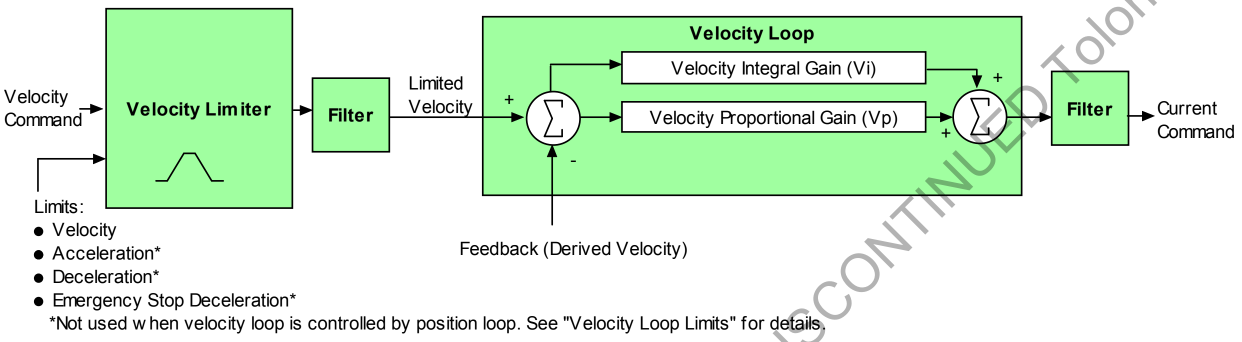
Modulo Count (Position Wrap)

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. 56).

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.


Velocity Command
Velocity Limiter
Filter
Limited Velocity
Velocity Loop
Velocity Integral Gain (Vi)
Velocity Proportional Gain (Vp)
+
+
+
-
∑
∑
Filter
Current Command
Limits:
● Velocity
● Acceleration*
● Deceleration*
● Emergency Stop Deceleration*
Feedback (Derived Velocity)
*Not used w hen velocity loop is controlled by position loop. See "Velocity Loop Limits" for details.

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 Alternative Control Sources Overview, p. 192.) 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)	121
*Velocity Loop Maximum Acceleration / 0x2100 (used only without position loop)	120
*Velocity Loop Maximum Deceleration / 0x2101 (used only without position loop)	121
Velocity Loop Emergency Stop Deceleration / 0x2102 (used only without position loop)	121
*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. 122).

Velocity Loop Gains

The velocity loop uses the following gains. See Velocity Loop Gains (index 0x60F9, p. 123).

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 Co-Efficients (index 0x2106, p. 124) and Velocity Loop Command Filter Co-Efficients (index 0x2108, p. 124).

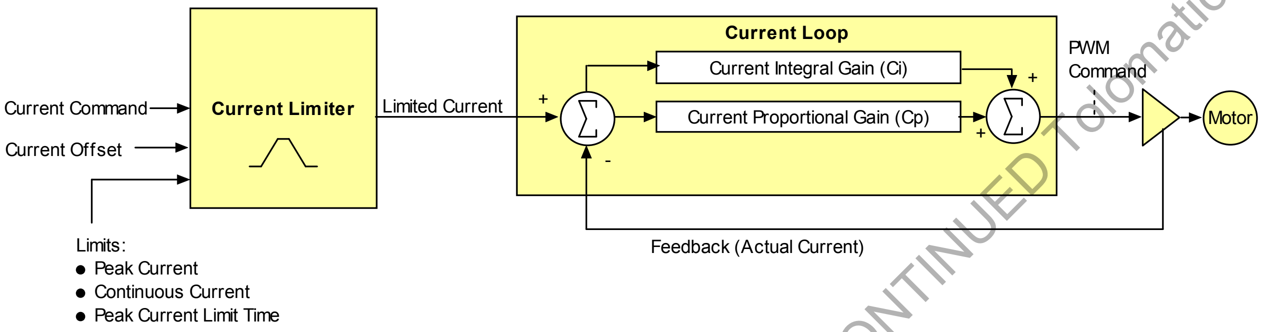
Velocity Loop Output

The output of the velocity loop is accessed in the Commanded Current object (index 0x221D, p. 128).

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 amplifier's power stage.



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 amplifier for a short duration of time. This value cannot exceed the peak current rating of the amplifier.	126
User Continuous Current Limit / 0x2111	Maximum current that can be constantly generated by the amplifier.	126
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.	126

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. 128).

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 0x60F6, p. 128) 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 amplifier.

4.2: Position Loop Configuration Objects

Contents of this Section

This section describes the objects used to configure the position control loop.

They include:

Instantaneous Commanded Velocity Index 0x2250

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 counts / sec	-	YES	-

Description

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 0x60FB, Sub-Index 2, p. 117) is multiplied.

Instantaneous Commanded Acceleration Index 0x2251

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	10 counts / sec^2	-	YES	-

Description

This is the acceleration output from the trajectory generator. It is the acceleration by which the position loop's Position Loop Acceleration Feed Forward gain (index 0x60FB, Sub-Index 3, p. 117) is multiplied.

Position Command Value Index 0x6062

Type	Access	Units	Range	Map PDO	Memory
Integer 32	R0	counts	-	YES	-

Description

This is the motor position (in units of counts) to which the amplifier is currently trying to move the axis. This value is updated every servo cycle based on the amplifier's internal trajectory generator. Identical to Position Command Value (index 0x6062, p. 114).

Position Actual Value Index 0x6063

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	R

Description

This is the actual motor position as calculated by the amplifier 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. 118).

Position Actual Value Index 0x6064

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	R

Description

This object holds the same value as Position Actual Value object (index 0x6063, p. 114).

TRACKING WARNING WINDOW INDEX 0X6065

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	0 - 2,147,483,647	YES	RF

Description

This object holds the maximum position error that the amplifier 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. 56) is set.

Note that this following error window generates a warning, not an amplifier fault. A separate tracking error window may be programmed which will cause an amplifier fault condition if exceeded. See the Tracking Error Window object (index 0x2120, p. 62) for details.

MAXIMUM SLIPPAGE-PROFILE VELOCITY MODE INDEX 0X60F8

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	0 - 2,147,483,647	YES	RF

Description

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. 115). A change to either object is reflected in the other.

POSITION TRACKING WINDOW INDEX 0X6067

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	0 - 2,147,483,647	YES	RF

Description

Size of the amplifier's tracking window. When the absolute position error of the motor is less then 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. 55) when the amplifier 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. 56) 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 amplifier is in motion.

POSITION TRACKING WINDOW TIME INDEX 0X6068

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	0 - 5000	YES	RF

Description

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. 115) for at least this long. The tracking window bit (bit 25) in the Manufacturer Status Register (index 0x1002, p. 56) will not be cleared until the position has been within the position tracking window for at least this long.

POSITION ERROR INDEX 0X60F4

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	counts	-	YES	-

Description

Also known as following error. This object gives the difference, in units of counts, between the Position Actual Value object (index 0x6063, p. 114) and the Position Command Value object (index 0x60FC, p. 117).

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. 117), Position Tracking Window object (index 0x6067, p. 115), and Tracking Error Window object (index 0x2120, p. 62).

POSITION LOOP CONTROL EFFORT INDEX 0X60FA

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 counts/sec	-	YES	-

Description

The position loop produces a commanded velocity as its output. This object gives access to that value. This value also represents the input to the velocity loop.

Position Loop Gains — Index 0x60FB

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	YES	-

Description

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
Integer 16	RW	-	0 –32,767	YES	RF

Description

This gain value is multiplied by the position loop error. The position loop error is the difference between the Position Command Value (index 0x60FC, p. 117) and the Position Actual Value (index 0x6064, p. 114).

Position Loop Velocity Feed Forward — Index 0x60FB, Sub-Index 2

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 –32,767	YES	RF

Description

This value is multiplied by the Instantaneous Commanded Velocity (index 0x2250, p. 114) 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
Integer 16	RW	-	0 –32,767	YES	RF

Description

This value is multiplied by the Instantaneous Commanded Acceleration (index 0x2251, p. 114) 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
Integer 16	RW	-	-	YES	RF

Description

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 Command Value — Index 0x60FC

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	counts	-	YES	-

Description

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 Command Value (index 0x6062, p. 114).

SOFTWARE POSITION LIMITS INDEX 0X607D

Type	Access	Units	Range	Map PDO	Memory
Array	RW	-	-	YES	-

Description

This array holds the two software position limit values Negative Software Limit Position and Positive Software Limit Position.

NEGATIVE SOFTWARE LIMIT POSITION INDEX 0X607D, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	RF

Description

Software limits are only in effect after the amplifier has been referenced (i.e. homing has been successfully completed). Set to less than negative software limit to disable.

POSITIVE SOFTWARE LIMIT POSITION INDEX 0X607D, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	RF

Description

Software limits are only in effect after the amplifier has been referenced (i.e. homing has been successfully completed). Set to greater than positive software limit to disable.

SOFTWARE LIMIT DECELERATION INDEX 0X2253

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	10 counts / sec^2	0 – 100,000,000	YES	RF

Description

The deceleration rate used when approaching a software limit.

MOTOR ENCODER POSITION INDEX 0X2240

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	R

Description

For single-encoder systems, this is the same as the Position Actual Value object (index 0x6063, p. 114). 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. 120).

LOAD ENCODER POSITION INDEX 0X2242

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	R

Description

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. 114). For single encoder systems, this object is not used.

4.3: Velocity Loop Configuration Objects

Contents of this Section

This section describes the objects used to configure the velocity control loop.

They include:

Velocity Command Value Index 0x606B

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 counts/sec	-	YES	-

Description

Also known as commanded velocity. The velocity that the velocity loop is currently trying to attain.

When the amplifier 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 amplifiers support some modes in which the velocity command is produced from a source other then the position loop. In these modes the command velocity comes from the analog reference input, the digital PWM inputs, or the internal function generator.

Actual Velocity Index 0x6069

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 enc counts / sec	-	YES	-

Description

Actual motor velocity.

Actual Velocity Index 0x606C

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 counts/sec	-	YES	-

Description

This object contains exactly the same information as object 0x6069.

Unfiltered Motor Encoder Velocity Index 0x2232

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 enc counts / sec	-	YES	-

Description

Unfiltered motor velocity.

Load Encoder Velocity Index 0x2231

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 counts / sec	-	YES	-

Description

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.

Velocity Loop Maximum Acceleration Index 0x2100

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	1000 enc counts / sec^2	0 – 100,000,000	YES	RF

Description

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

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	1000 enc counts / sec^2	0 – 100,000,000	YES	RF

Description

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).

VELOCITY LOOP EMERGENCY STOP DECELERATION INDEX 0X2102

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	1000 enc counts / sec^2	0 – 100,000,000	YES	RF

Description

The deceleration rate used during the time that the amplifier is trying to actively stop a motor before applying the brake output.

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
Integer 32	RW	0.1 counts/sec	0 – 500,000,000	YES	RF

Description

This velocity value is a limit on the commanded velocity used by the velocity loop.

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
Integer 32	RW	0.1 counts/sec	0 – 500,000,000	YES	RF

Description

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. 56) 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. 62).

VELOCITY ERROR WINDOW – PROFILE VELOCITY INDEX 0X606D

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	0.1 counts/sec	0 – 65,535	YES	RF

Description

Object 606D 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 unsigned 16 and 0x2104 is Integer 32. Changes made to either object affect both.

Velocity Error Window Time Index 0x2105

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	0- 5,000	YES	RF

Description

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. 121) the Velocity Window bit (bit 28) in the Manufacturer Status Register object (index 0x1002, p. 56) is cleared.

Velocity Error Window Time Index 0x606E

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	0- 5,000	YES	RF

Description

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.

Limited Velocity Index 0x2230

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	0.1 counts/sec	-	YES	-

Description

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.

Programmed Velocity Command Index 0x2341

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	0.1 counts/sec	-500,000,000 – 500,000,000	YES	RF

Description

Gives the commanded velocity value when running in programmed velocity mode (see mode 11, Desired State *object*, p. 60, and Alternative Control Sources Overview, p. 192).

Velocity Loop Gains — Index 0x60F9

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	YES	-

Description

This object contains the various gain values used to optimize the velocity control loop.

Velocity Loop Proportional Gain — Index 0x60F9, Sub-Index 1

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 – 32,767	YES	RF

Description

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 0x60F9, Sub-Index 2

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 – 32,767	YES	RF

Description

This gain value is multiplied by the integral of the velocity loop error.

Velocity Loop Acceleration Feed Forward — Index 0x60F9, Sub-Index 3

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 – 32,767	YES	RF

Description

This gain value is multiplied by the Instantaneous Commanded Acceleration (index 0x2251, p. 114) from the trajectory generator. The result is added to the output of the velocity loop.

Velocity Loop Gain Scaler — Index 0x60F9, Sub-Index 4

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 – 32,767	YES	RF

Description

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 0x60F9, Sub-Index 5

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	0 – 32,000	YES	RF

Description

Modifies the effect of velocity loop integral gain. The higher the Vi Drain value, the faster the integral sum is lowered.

Hall Velocity Mode Shift Value — Index 0x2107

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	0 – 32,767	YES	RF

Description

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 OUTPUT FILTER CO-EFFICIENTS INDEX 0X2106

Type	Access	Units	Range	Map PDO	Memory
Octet	RW	-	-	YES	RF

Description

Programs the filter coefficients of a bi-quad filter structure that acts on the velocity loop output. Contact Copley Controls for more information.

VELOCITY LOOP COMMAND FILTER CO-EFFICIENTS INDEX 0X2108

Type	Access	Units	Range	Map PDO	Memory
Octet	RW	-	-	YES	RF

Description

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 CO-EFFICIENTS INDEX 0X2109

Type	Access	Units	Range	Map PDO	Memory
Octet	RW	-	-	YES	RF

Description

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.

4.4: Current Loop Configuration Objects

Contents of this Section

This section describes the objects used to configure the current control loop.

They include:

USER PEAK CURRENT LIMIT INDEX 0X2110

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.01 amps	0 – 32,767	YES	RF

Description

User peak current limit. Known as boost current on stepper amplifiers. This value cannot exceed the peak (or boost) current rating of the amplifier.

USER CONTINUOUS CURRENT LIMIT INDEX 0X2111

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.01 amps	0 – 32,767	YES	RF

Description

User Continuous Current Limit. Also known as Run Current on stepper amplifiers. This value should be less then the User Peak Current Limit. The amplifier uses this value as an input to an I^2T current limiting algorithm to prevent over stressing the load.

USER PEAK CURRENT LIMIT TIME INDEX 0X2112

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	milliseconds	0 – 10,000	YES	RF

Description

Specifies the maximum time at peak current. The amplifier uses this value as an input to an I^2T current limiting algorithm to prevent over stressing the load.

ACTUAL CURRENT, D AXIS INDEX 0X2214

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

Part of the internal current loop calculation.

ACTUAL CURRENT, Q AXIS INDEX 0X2215

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

Part of the internal current loop calculation.

CURRENT COMMAND, D AXIS INDEX 0X2216

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

Part of the internal current loop calculation.

CURRENT COMMAND, Q AXIS INDEX 0X2217

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

Part of the internal current loop calculation.

CURRENT LOOP OUTPUT, D AXIS INDEX 0X2218

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.1 V	-	YES	-

Description

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
Integer 16	RO	0.1 V	-	YES	-

Description

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
Integer 16	RO	0.01 amps	-	YES	-

Description

Actual motor current.

COMMANDED CURRENT INDEX 0X221D

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

Instantaneous commanded current as applied to the current limiter.

LIMITED CURRENT INDEX 0X221E

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	0.01 amps	-	YES	-

Description

Output of the current limiter (input to the current loop).

PROGRAMMED CURRENT COMMAND INDEX 0X2340

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.01 amps	-	YES	RF

Description

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. 60, and Alternative Control Sources Overview, p. 192.)

COMMANDED CURRENT RAMP RATE INDEX 0X2113

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	mA/second	-	YES	RF

Description

Setting this to zero disables slope limiting in Profile Torque mode. It is also used when the amplifier is running in Programmed Current mode (Desired State object [index 0x2300, p. 60] = 1).

CURRENT LOOP GAINS INDEX 0X60F6

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	YES	-

Description

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 0X60F6, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 – 32,767	YES	RF

Description

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 0X60F6, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	0 – 32,767	YES	RF

Description

This gain value is multiplied by the integral of current error.

CURRENT OFFSET INDEX 0X60F6, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.01 amps	-	YES	RF

Description

This offset value is added to the commanded motor current. It can be used to compensate for a directional bias affecting the current loop.

4.4: 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 amplifier 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
Integer 16	RW	-	-	YES	RF

Description

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. 130) as the key.
	2	Use Instantaneous Commanded Velocity (index 0x2250, p. 114).
	3	Use Load Encoder Velocity (index 0x2231, p. 120).
	4	Use Position Command Value object (index 0x60FC, p. 117).
	5	Use Position Actual Value object (index 0x6063, p. 114).
	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).	

GAIN SCHEDULING KEY PARAMETER INDEX 0X2371

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	-	-	YES	R

Description

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.

CHAPTER
5: STEPPER MODE SUPPORT

This chapter describes Copley Controls' support of stepper motor operation over a CANopen network.

Contents include:

5.1: Stepper Mode Operation

Copley Controls Amplifiers and Stepper Mode Operation

Copley Controls supports the use of stepper motors over a CANopen network.

The Stepnet amplifier can drive a two-phase stepper motor in stepper or servo mode.

The Accelnet and Xenus amplifiers can drive a three-phase stepper motor in stepper mode.

Stepper vs. Servo

In a closed-loop servo system, sensors feed back the actual position and/or velocity of the motor, and the amplifier 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 amplifier 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 amplifier 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. 132).

The Accelnet and Xenus amplifiers 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 amplifiers provide a much higher degree of control over a motor's position than does a full stepping system. The microstepping amplifier 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 0x6410, Sub-Index 2, p. 81)	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 0x6410, Sub-Index 29, p. 86)	Microsteps per revolution.

There is virtually no limit on the number of microsteps/rev. Programming a very high value does not mean that the amplifier 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 amplifiers 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 amplifiers 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 amplifier 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 then 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 then 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.

5.2: Stepper Mode Objects

Contents of this Section

This section describes the objects used to support stepper motor operation. Some are also used in servo mode operation.

They include:

Boost Current

Index 0x2110

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.01 amps	0 – 32,767	YES	RF

Description

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
Integer 16	RW	0.01 amps	0 – 32,767	YES	RF

Description

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
Integer 16	RW	milliseconds	0 – 10,000	YES	RF

Description

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 amplifier 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
Unsigned 16	RW	0.01 amps	0 - 32,767	YES	RF

Description

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
Unsigned 16	RW	milliseconds	0 - 10,000	YES	RF

Description

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
Integer 16	RW	-	-	YES	RF

Description

Can be used to reduce detent noise.

Voltage Control Mode Time Delay

Index 0x21D5

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	0 - 10,000	YES	RF

Description

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
Unsigned 16	RW	-	-	YES	RF

Description

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. 116). When this bit is set, the gain value Maximum Velocity Adjustment (index 0x21D5, p. 135) is multiplied by the Position Error, and the result is a velocity that is added to the microstepping position.
2-15	Reserved.

MAXIMUM VELOCITY ADJUSTMENT — INDEX 0x21D8

Type	Access	Units	Range	Map PDO	Memory
Unsigned 32	RW	0.1 steps/sec	-	YES	RF

Description

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. 136) is set.

PROPORTIONAL GAIN FOR STEPPER OUTER LOOP — INDEX 0x21D7

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	-	YES	RF

Description

This parameter gives the gain used for calculating a velocity adjustment based on Position Error (index 0x60F4, p. 116). 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. 136) is set.

CHAPTER

6: Homing Mode Operation

This chapter describes the operation of an amplifier in homing mode.

Contents include:

6.1: Homing Mode Operation Overview

Contents of this Section

This section describes control of the amplifier in homing mode.

Topics include:

Homing 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:

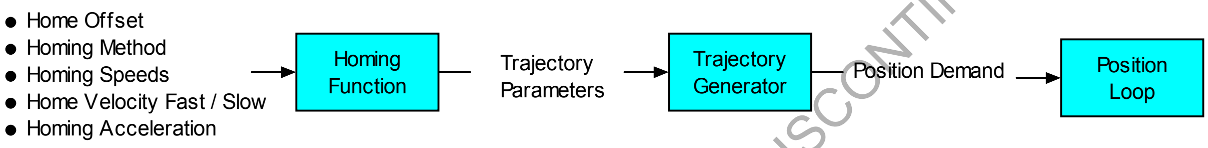
- limit switches at the ends of travel, or
- a dedicated home switch.

Most of the methods also use the index pulse input from an incremental encoder.

The amplifier performs homing operations in Homing Mode (Mode Of Operation [index 0x6060, p. 59] =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.

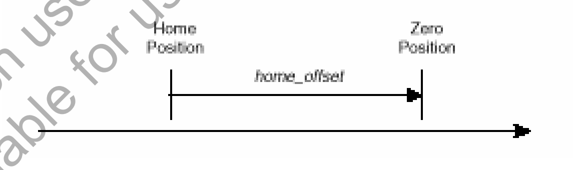


Initiating and Verifying a Homing Sequence

A homing move is started by setting bit 4 of the Control Word object (index 0x6040, p. 54). The results of a homing operation can be accessed in the Status Word (index 0x6041, p. 55).

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.



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. 158])

Homing Acceleration

Homing Acceleration (index 0x609A, p. 158) 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.

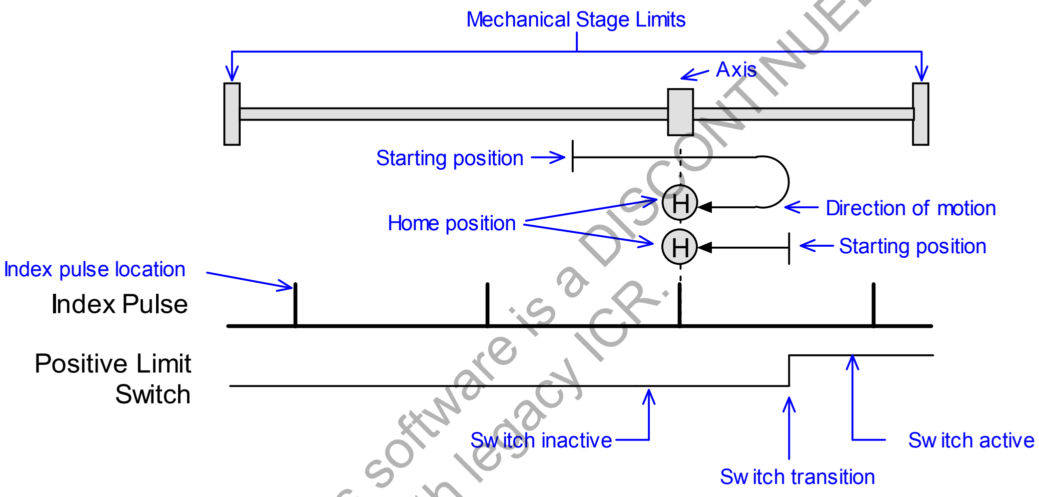
Homing Methods Overview

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.



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. 157) to: 0.

Home is Current Position; Move to New Zero

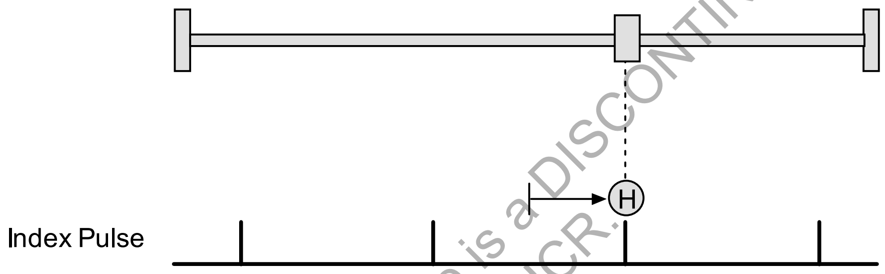
Set current position to home and move to new zero position (including home offset). This is the same as Home is Current Position except that mode 0 does not do the final move to the home position.

Set Homing Method (index 0x6098, p. 157) to: 35.

Next Index

Direction of Motion: Positive

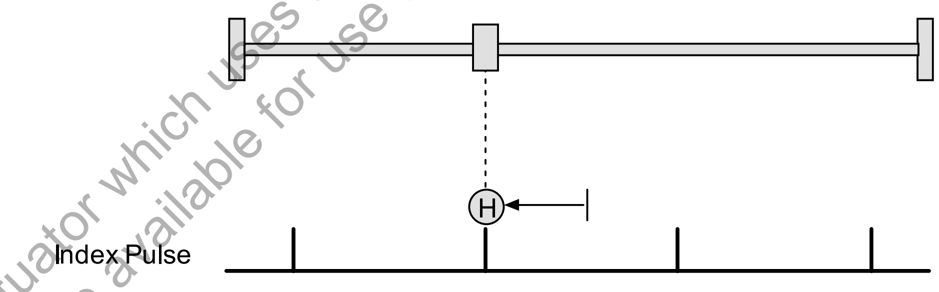
Home is the first index pulse found in the positive direction. Direction of motion is positive. If a positive limit switch is activated before the index pulse, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

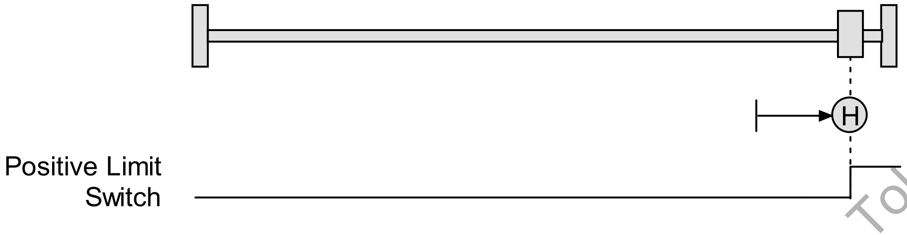


Set Homing Method (index 0x6098, p. 157) to: 33.

Limit Switch

Direction of Motion: Positive

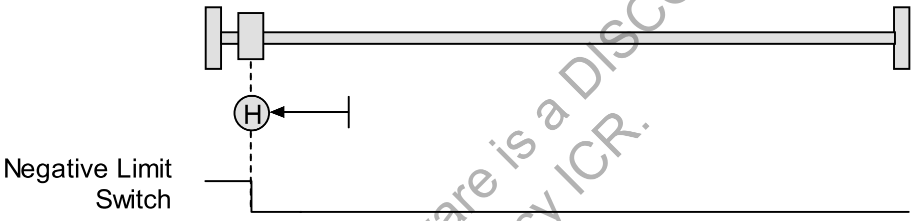
Home is the transition of the positive limit switch. Initial direction of motion is positive if the positive limit switch is inactive.



Set Homing Method (index 0x6098, p. 157) 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.

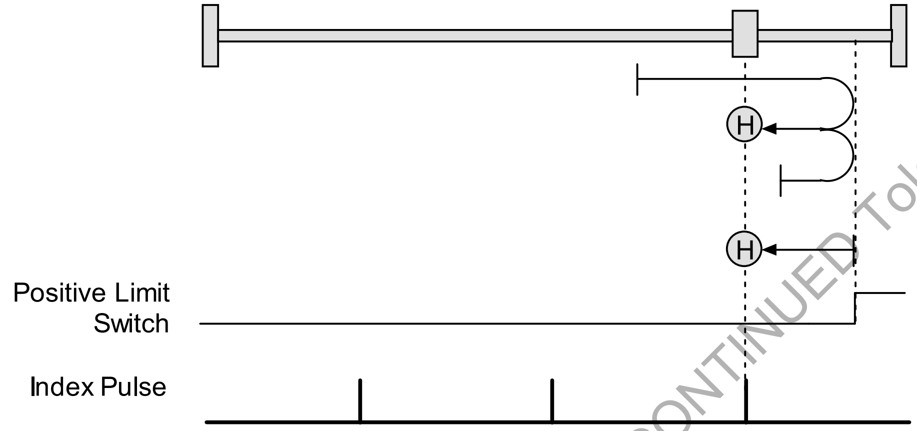


Set Homing Method (index 0x6098, p. 157) to: 17.

Limit Switch Out to Index

Direction of Motion: Positive

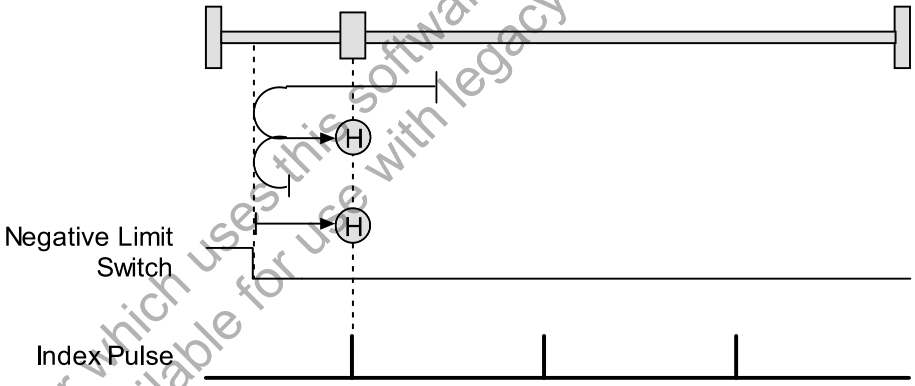
Home is the first index pulse to the negative side of the positive limit switch transition. Initial direction of motion is positive if the positive limit switch is inactive (shown here as low).



Set Homing Method (index 0x6098, p. 157) 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).



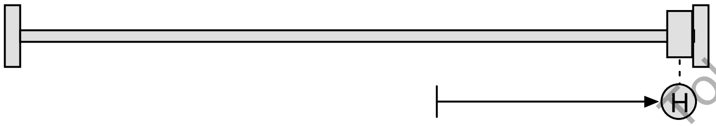
Set Homing Method (index 0x6098, p. 157) to: 1.

Hardstop

Direction of Motion: Positive

Home is the positive hard stop. Direction of motion is positive. The hard stop is reached when the amplifier 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.

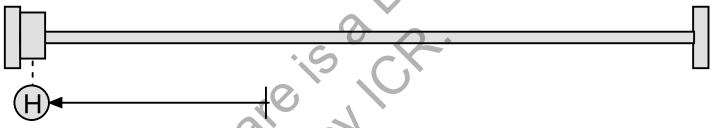
In stepper amplifiers in stepper mode, the hard stop is reached when the following error exceeds the tracking window.



Set Homing Method (index 0x6098, p. 157) to: -1.

Direction of Motion: Negative

Home is the negative hard stop. Direction of motion is negative. The hard stop is reached when the amplifier 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.



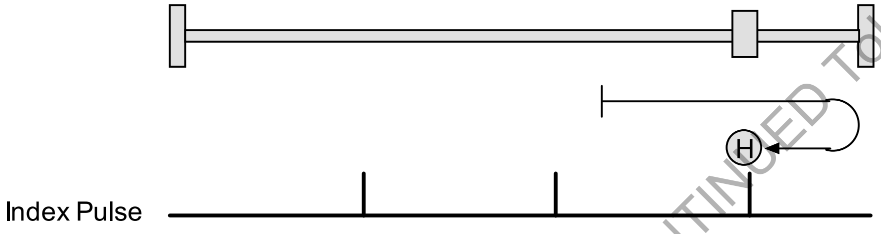
Set Homing Method (index 0x6098, p. 157) to: -2.

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 amplifier 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.

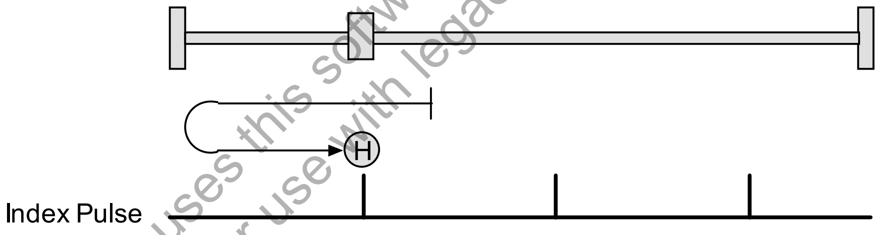
In stepper amplifiers in stepper mode, the hard stop is reached when the following error exceeds the tracking window.



Set Homing Method (index 0x6098, p. 157) 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 amplifier 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.

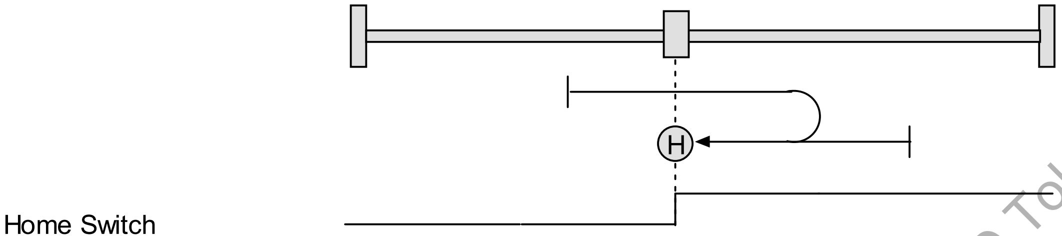


Set Homing Method (index 0x6098, p. 157) to:-3.

Home Switch

Direction of Motion: Positive

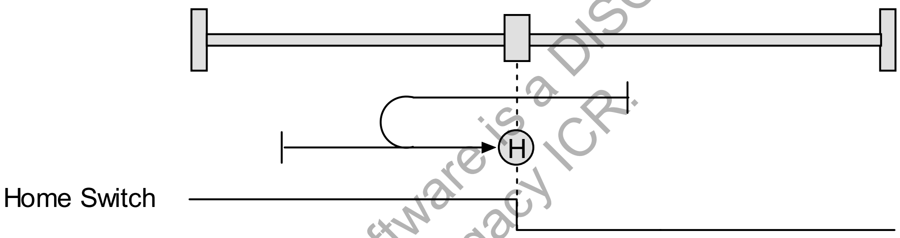
Home is the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

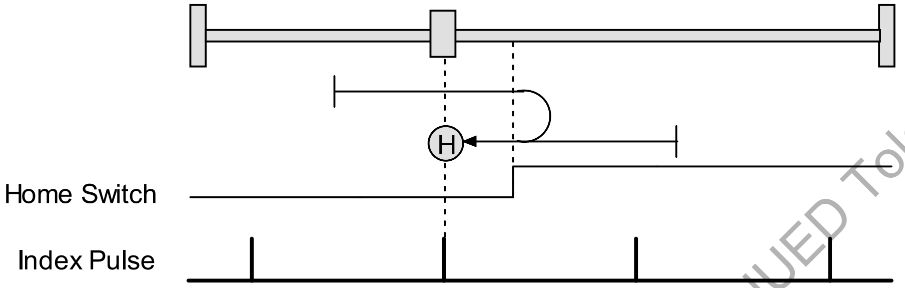


Set Homing Method (index 0x6098, p. 157) to: 21.

Home Switch Out to Index

Direction of Motion: Positive

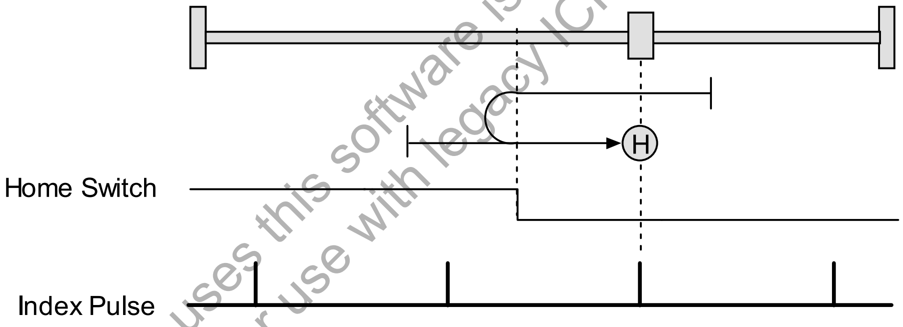
Home is the first index pulse to the negative side of the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

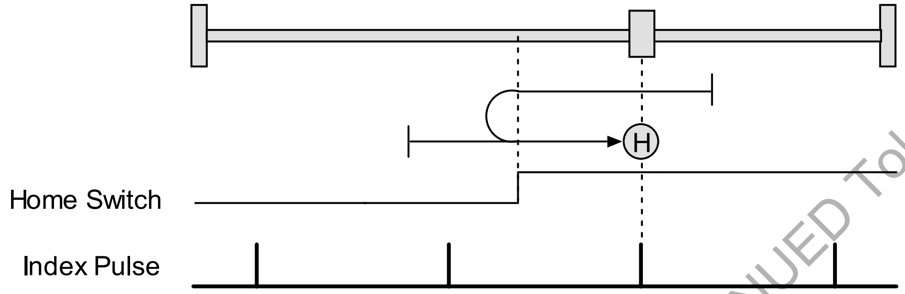


Set Homing Method (index 0x6098, p. 157) to: 5.

Home Switch In to Index

Direction of Motion: Positive

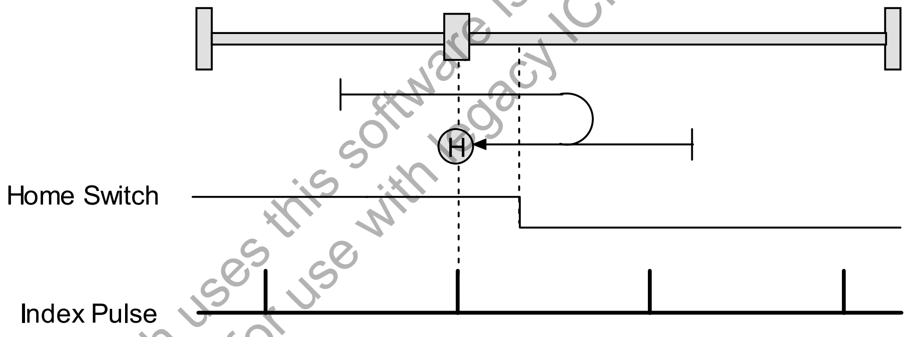
Home is the first index pulse to the positive side of the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

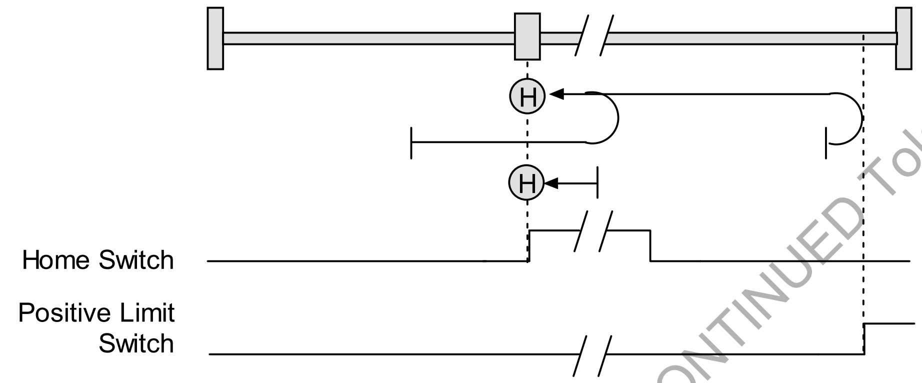


Set Homing Method (index 0x6098, p. 157) to: 6.

Lower Home

Direction of Motion: Positive

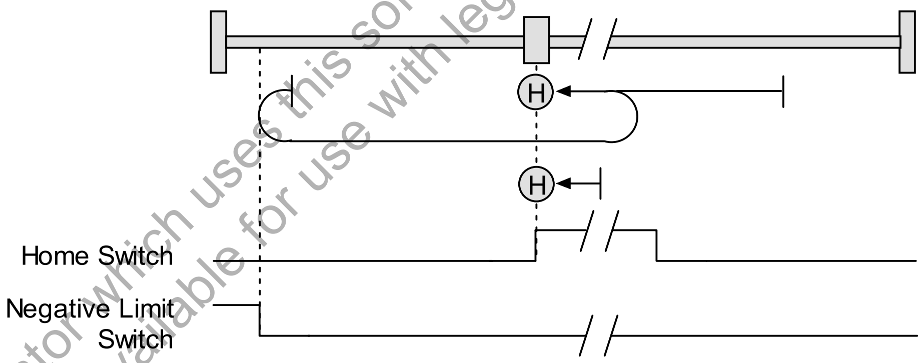
Home is the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. Motion will reverse if a positive limit switch is activated before the home switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

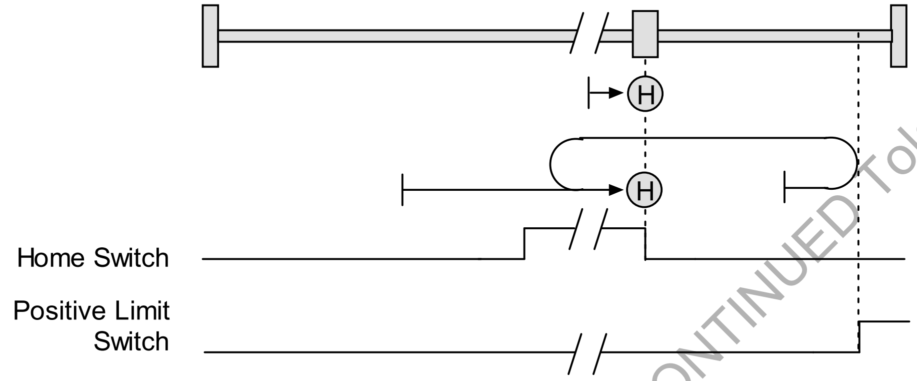


Set Homing Method (index 0x6098, p. 157) to: 29.

Upper Home

Direction of Motion: Positive

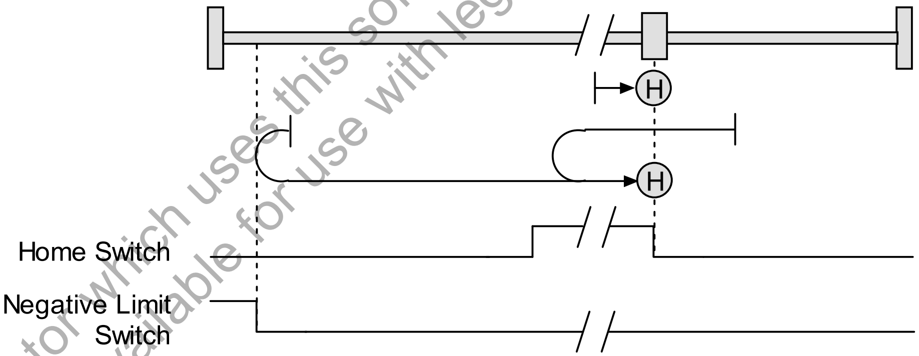
Home is the positive edge of a momentary home switch. Initial direction of motion is positive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

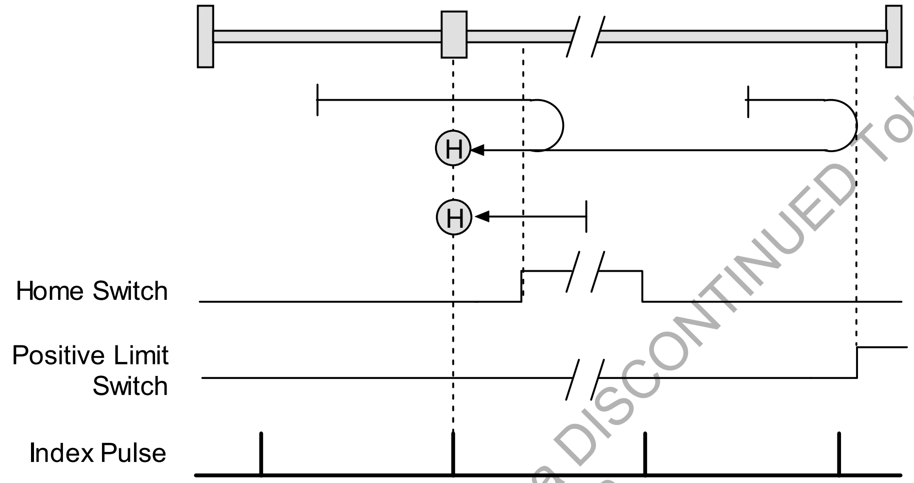


Set Homing Method (index 0x6098, p. 157) to: 27

Lower Home Outside Index

Direction of Motion: Positive

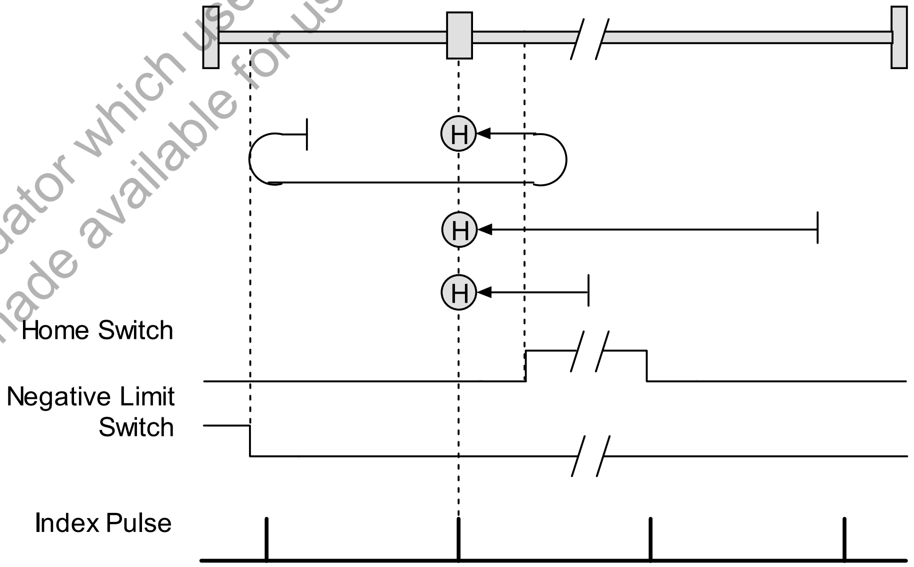
Home is the first index pulse on the negative side of the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

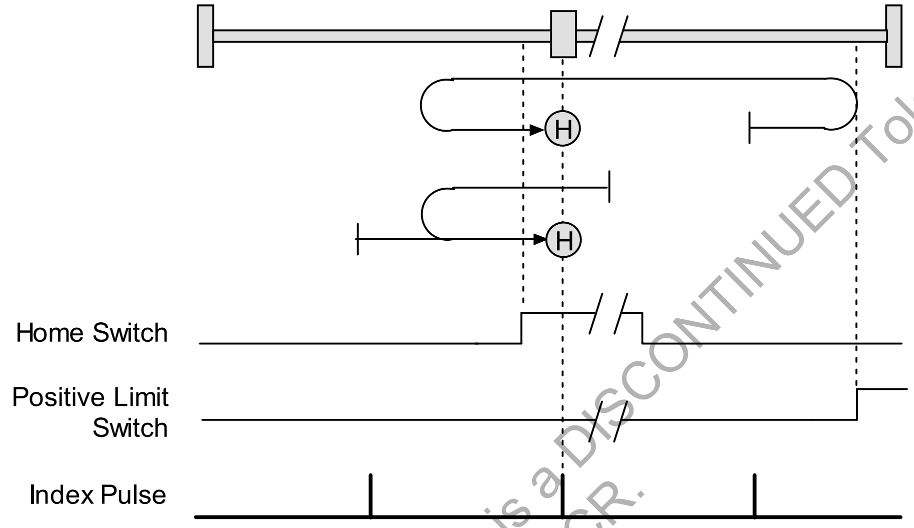


Set Homing Method (index 0x6098, p. 157) to: 14.

Lower Home Inside Index

Direction of Motion: Positive

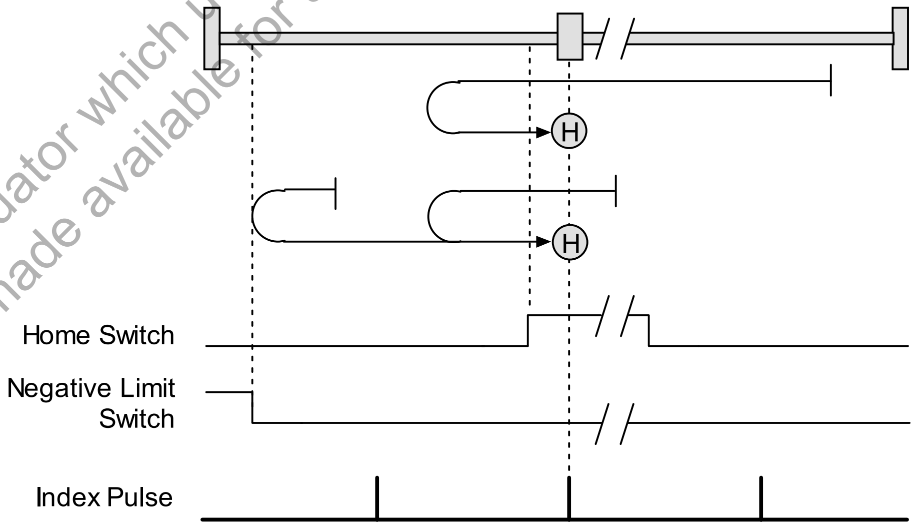
Home is the first index pulse on the positive side of the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

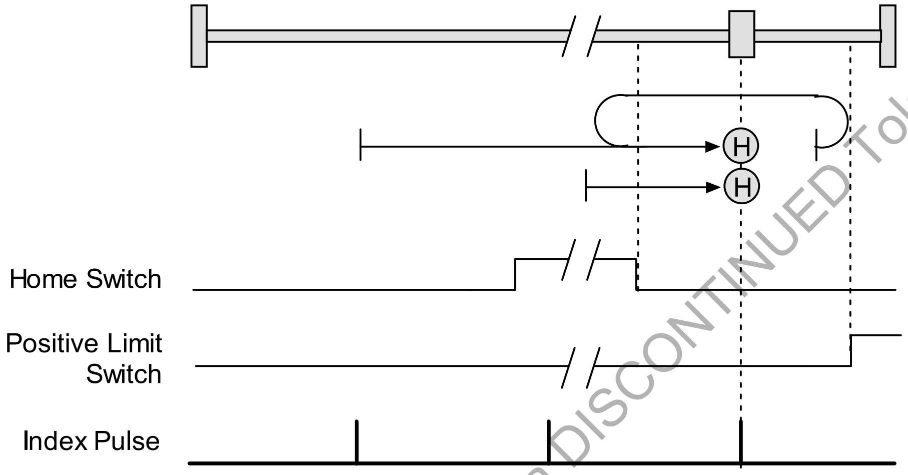


Set Homing Method (index 0x6098, p. 157) to: 13.

Upper Home Outside Index

Direction of Motion: Positive

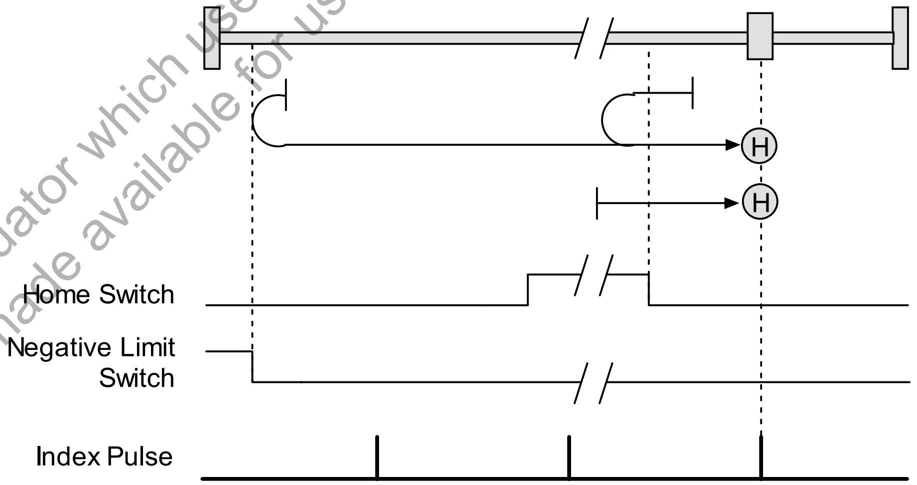
Home is the first index pulse on the positive side of the positive edge of a momentary home switch. Initial direction of motion is positive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.

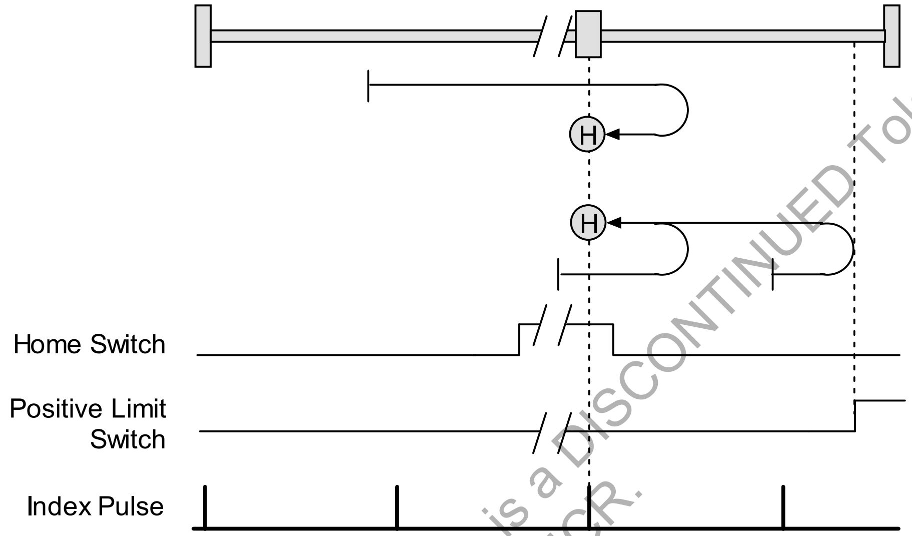


Set Homing Method (index 0x6098, p. 157) to: 11.

Upper Home Inside Index

Direction of Motion: Positive

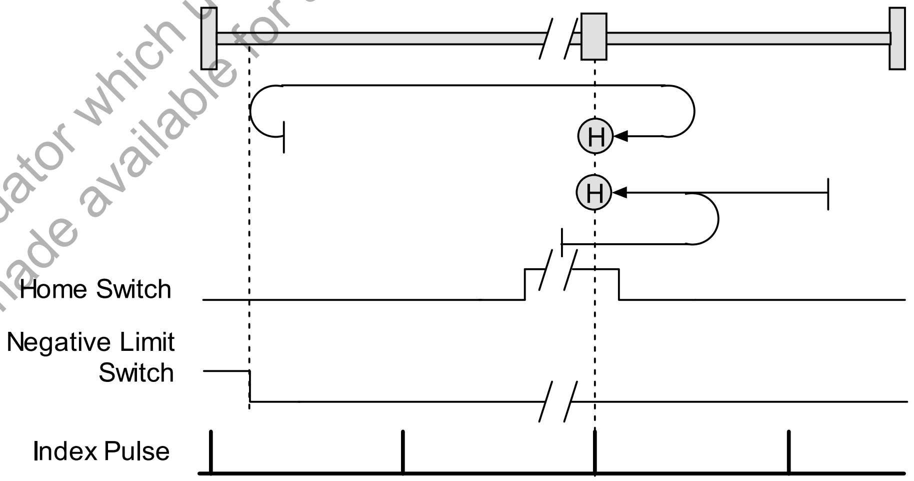
Home is the first index pulse on the negative side of the positive edge of momentary home switch. Initial direction of motion is positive. If initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Set Homing Method (index 0x6098, p. 157) 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.



Set Homing Method (index 0x6098, p. 157) to: 12.

Copley Controls Home Configuration Object for Custom Homing Methods

Copley Controls provides an object that provides access to the amplifier'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.

6.2: Homing Mode Operation Objects

Contents of this Section

This section describes the objects that control the operation of the amplifier in homing mode.

They include:

Homing Method Index 0x6098

Type	Access	Units	Range	Map PDO	Memory
Integer 8	RW	-	See *Description*, below.	YES	RF

Description

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. 145
	Negative	-3	
Hardstop	Negative	-2	p. 144
	Positive	-1	
Home is Current Position	Any	0	p. 141
Home is Current Position; Move to New Zero	Any	35	p. 141
Limit Switch Out to Index	Negative	1	p. 143
	Positive	2	
Home Switch Out to Index	Positive	3	p. 147
	Negative	5	
Home Switch In to Index	Positive	4	p. 148
	Negative	6	
Lower Home Outside Index	Positive	7	p. 151
	Negative	14	
Lower Home Inside Index	Positive	8	p. 152
	Negative	13	
Upper Home Inside Index	Positive	9	p. 154
	Negative	12	
Upper Home Outside Index	Positive	10	p. 153
	Negative	11	
Limit Switch	Negative	17	p. 142
	Positive	18	
Home Switch	Positive	19	p. 146
	Negative	21	
Lower Home	Positive	23	p. 149
	Negative	29	
Upper Home	Positive	25	p. 150
	Negative	27	
Next Index	Positive	34	p. 141
	Negative	33	
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.140.

Homing Speeds Index 0x6099

Type	Access	Units	Range	Map PDO	Memory
Array	RW	-	-	YES	-

Description

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
Integer 32	RW	0.1 counts/sec	0 – 500,000,000	YES	RF

Description

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.

Home Velocity – Slow Index 0x6099, Sub-Index 2

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	0.1 counts/sec	0 – 500,000,000	YES	RF

Description

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.

Homing Acceleration Index 0x609A

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	10 counts/sec^2	0 – 200,000,000	YES	RF

Description

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).

Home Offset Index 0x607C

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	RF

Description

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. 139) for more information.

HARD STOP MODE HOME DELAY — INDEX 0X2351

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	milliseconds	0 - 10,000	YES	RF

Description

Delay used for homing to a hard stop mode.

HARD STOP MODE HOME CURRENT — INDEX 0X2350

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	0.01A	0 - 32,767	YES	RF

Description

Home current in hard stop mode, in which the amplifier drives the motor to the mechanical end of travel (hard stop). End of travel is recognized when the amplifier outputs the Hard Stop Mode Home Current for the Hard Stop Mode Home Delay time (index 0x2351, p. 159).

HOME CONFIG — INDEX 0X2352

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below.	YES	RF

Description

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.
	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. 161) 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.	

Position Capture Control Register Index 0x2400

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See *Description*, below.	YES	RF

Description

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, p. 161) is captured on the falling edge of the index.
1	If set, the Captured Index Position is captured on the rising 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, Home Capture Position (index 0x2403, p. 161) captures falling edges of the home switch input transition; if clear, it captures rising edges.
6	If set, Home Capture Position will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.
8	If set, enable high speed input position capture. The position value is written to Captured Position for High Speed Position Capture (index 0x2405, p. 161).
9	If set, don't overwrite high speed input capture positions.
10	If set, a Captured Position for High Speed Position Capture value will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.
12	Clear actual position on every encoder index pulse.

Position Capture Status Register Index 0x2401

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	See *Description*, below.	YES	-

Description

Shows the current status of the index or home switch capture mechanism. Bit-mapped as follows:

Bit	Description
0	If set, an index position has been captured. Cleared when the captured position is read.
1,2	Reserved.
3	If set, a new index transition occurred when a captured position was already stored. The existing Captured Index Position (index 0x2402, p. 161) will be overwritten or preserved as programmed in bit 2 of the Position Capture Control Register (index 0x2400, p. 160).
4	If set, new home switch transition data has been captured.
5,6	Reserved.
7	If set, a new home switch input transition occurred when a captured position was already stored. The existing Home Capture Position (index 0x2403, p. 161) will be overwritten or preserved as programmed in bit 6 of the Position Capture Control Register.
8	If set, a new high speed input position has been captured. Cleared when the captured position is read.
10	If set, high speed input position overflow.
11	If set, a new high speed input transition occurred when a Captured Position for High Speed Position Capture (index 0x2405, p. 161) was already stored. The existing Captured Position for High Speed Position Capture will be overwritten or preserved as programmed in bit 10 of the Position Capture Control Register.

Captured Index Position — Index 0x2402

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	counts	-	YES	-

Description

Reading this variable resets bits 0 & 3 of the Position Capture Status Register (index 0x2401, p. 160). 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. 160), 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
Integer 32	RO	counts	-	EVENT	-

Description

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. 160), and the status of the captured data can be checked in Position Capture Status Register (index 0x2401, p. 160).

Time Stamp of Last High Speed Position Capture — Index 0x2404

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	microseconds	-	EVENT	R

Description

Provides the time when an input pin configured as a high speed capture input became active (and the axis position was captured).

Captured Position for High Speed Position Capture — Index 0x2405

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	counts	-	EVENT	R

Description

Provides the position that the axis was in when an input pin configured as a high speed capture input became active.

Homing Adjustment — Index 0x2353

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	counts	-	EVENT	R

Description

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 sequence.

CHAPTER

7: PROFILE POSITION, VELOCITY, AND TORQUE MODE OPERATION

This chapter describes the operation of an amplifier in profile position, profile velocity, and profile torque mode.

Contents include:

7.1: Profile Position Mode Operation

Point-to-Point Motion Profiles

In profile position mode, an amplifier 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.

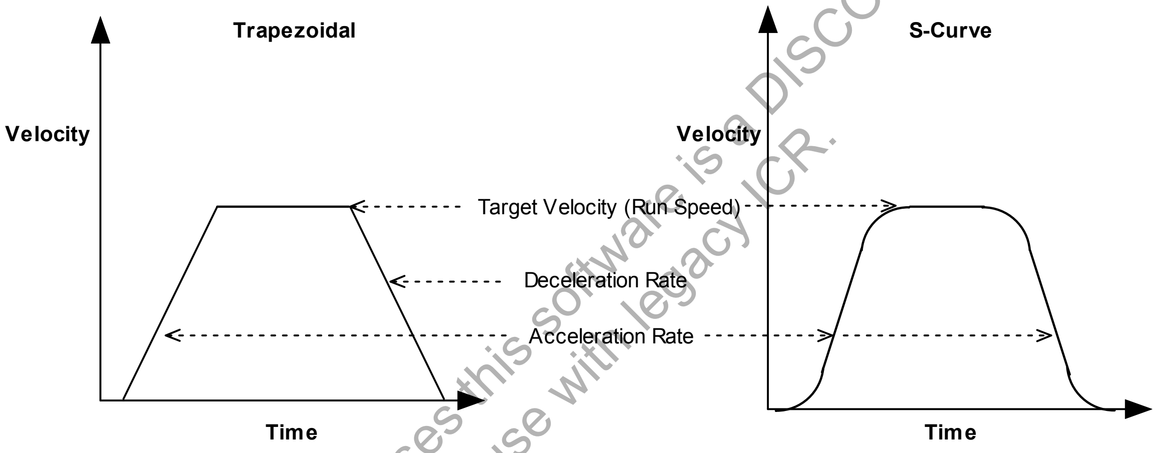
The amplifier performs profile position moves in Profile Position Mode (Mode Of Operation [index 0x6060, p. 59] =1).

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.



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 "smoothes" 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, p. 165.

The Motion Profile Type object (index 0x6086, p. 178) 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: Some Design Considerations, p. 171.

(Copley Controls CANopen amplifiers also support a profile position special velocity mode. This profile type resembles the trapezoidal profile, but with no target position specified. The motion obeys 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

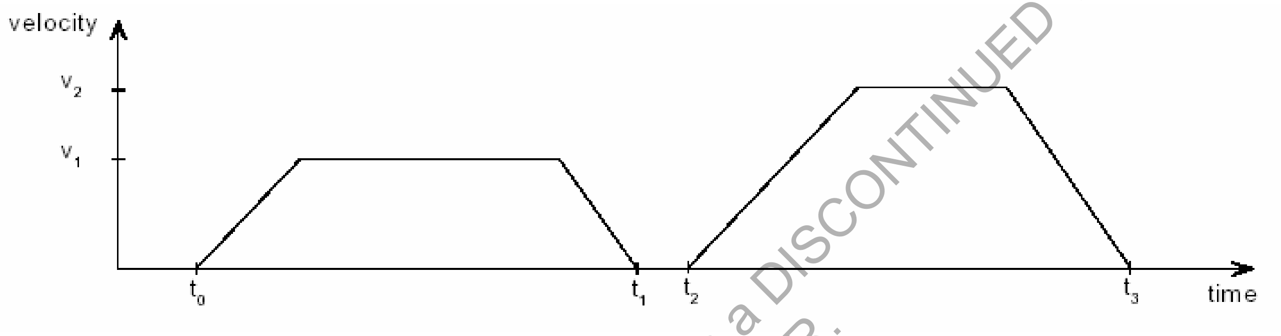
There are two methods for handling a series of point-to-point moves:

- 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.

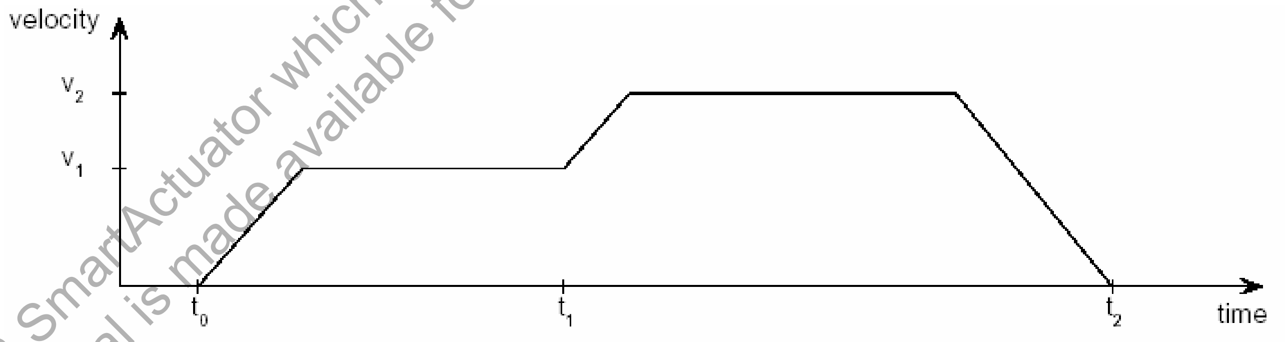


The *CANopen Profile for Drives and Motion Control (DSP 402)* refers to this method as the "single setpoint" method.

Copley Controls CANopen amplifiers 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).



The *CANopen Profile for Drives and Motion Control (DSP 402)* refers to this method as the "set of setpoints" method.

Copley Controls CANopen amplifiers allow use of this method with trapezoidal profile moves only.

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.	175
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.	175
Profile Velocity / 0x6081	Velocity that the trajectory generator will attempt to achieve when running in position profile mode.	176
Profile Acceleration / 0x6083	Acceleration that the trajectory generator attempts to achieve when running in position profile mode.	177
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.	177
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.	178
Motion Profile Type / 0x6086	Trapezoidal, S-curve, or special velocity mode.	178

The Point-to-Point Move Buffer

In profile position mode, the amplifier 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 amplifier'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 setpoint	The transition of bit 4 from 0 to 1 is what causes the amplifier 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: Amplifier 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. 178) = 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	Amplifier sets bit 10 to 1 when target position has been reached. Amplifier clears bit 10 to zero when new target is received. If quick stop option code (p. 58) 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	setpoint acknowledge	Set by the amplifier 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

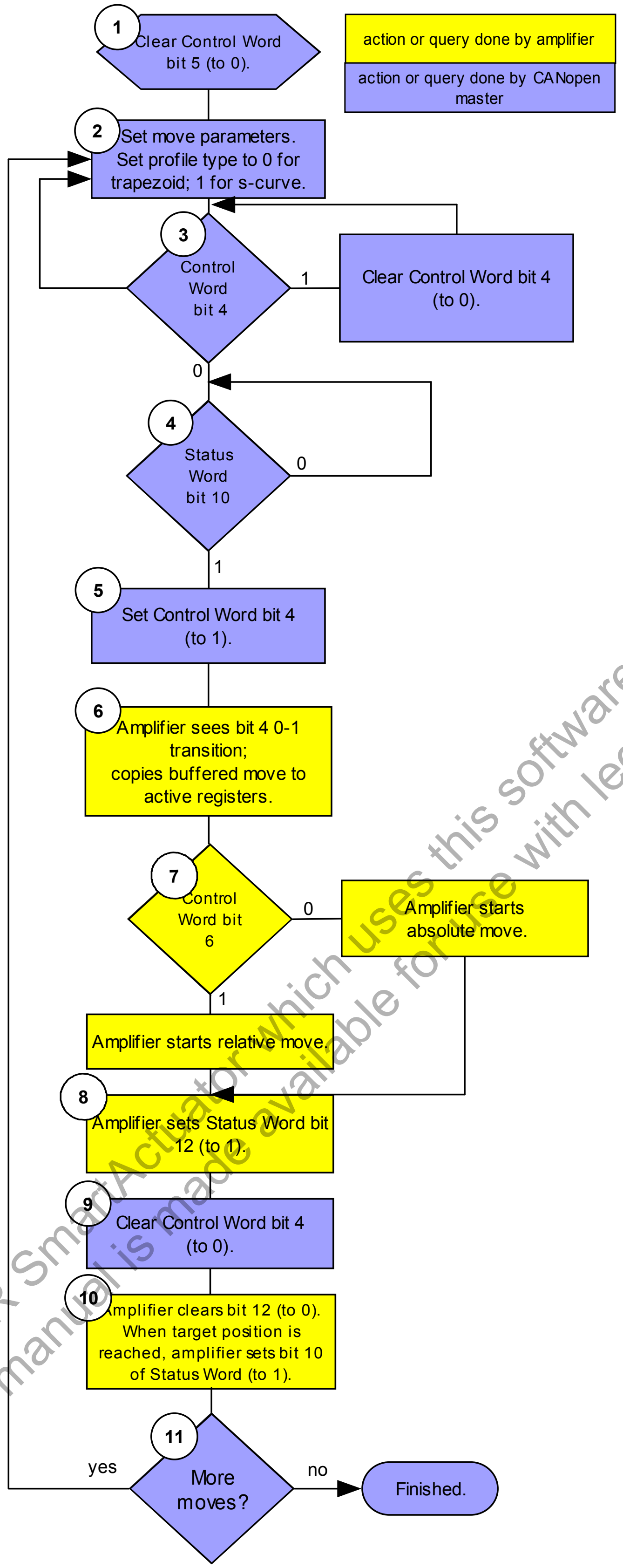
Overview

The following sections illustrate how to perform:

- A series of moves treated as a Series of Discrete Profiles
- A series of trapezoidal or profile position special velocity moves treated as One Continuous Profile

Series of Discrete Profiles

This diagram illustrates how to implement a series of moves as a series of discrete profiles.


action or query done by amplifier
action or query done by CANopen master
1 Clear Control Word bit 5 (to 0).
2 Set move parameters. Set profile type to 0 for trapezoid; 1 for s-curve.
3 Control Word bit 4
1
0
Clear Control Word bit 4 (to 0).
4 Status Word bit 10
0
1
5 Set Control Word bit 4 (to 1).
6 Amplifier sees bit 4 0-1 transition; copies buffered move to active registers.
7 Control Word bit 6
0
1
Amplifier starts absolute move.
Amplifier starts relative move.
8 Amplifier sets Status Word bit 12 (to 1).
9 Clear Control Word bit 4 (to 0).
10 Amplifier clears bit 12 (to 0). When target position is reached, amplifier sets bit 10 of Status Word (to 1).
11 More moves?
yes
no
Finished.

Notes:

1. Control Word bit 5 is "change set immediately." Clearing it tells the amplifier to treat a series of moves as a series of discrete profiles.

2. Move Parameters are described on page 108.

3. Control Word bit 4 is "new setpoint." It needs to be 0 because the move is triggered by a 0->1 transition.

4. Status Word bit 10 is "target reached." Value is 0 when move is in progress; 1 when move is finished.

5. Value of 1 indicates that valid data has been sent to amplifier and new move should begin.

6. Amplifier must detect 0-1 transition to begin move.

7. Control word bit 6: value 0 causes absolute move; value 1 causes relative move.

8. Status Word bit 13 is "setpoint acknowledge." A value of 1 indicates the amplifier has received a setpoint and has started the move.

9. Control Word bit 4 is "new setpoint." It needs to be 0 to allow the next move is triggered by a 0->1 transition. Also, the 1->0 transition causes the amplifier to clear bit 13.

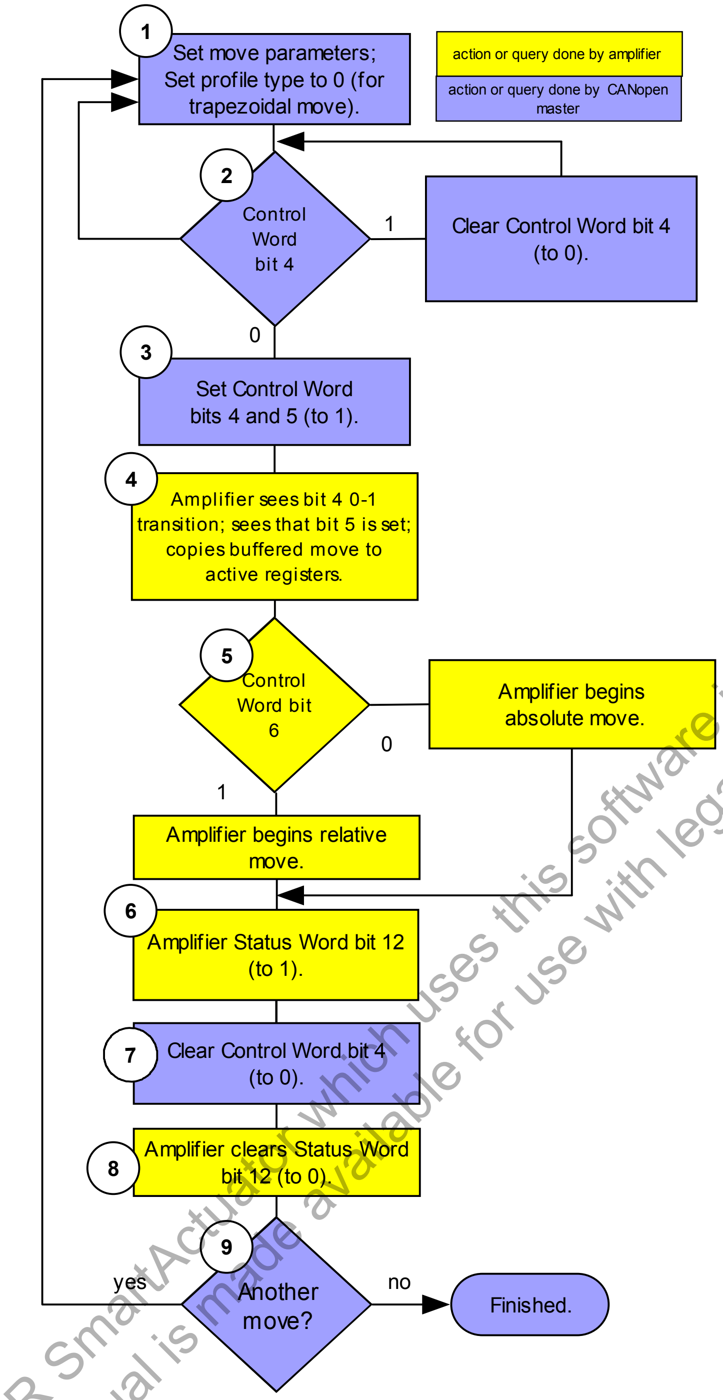
10. Amplifier detects 0->1 transition of Control Word bit 4 and clears bit 13 in response.

When the motor reaches the target position, the amplifier sets Status Word bit 10 ("target reached") to 1.

11. CANopen master returns to step 2 if there are more moves to complete; otherwise, the series of moves is finished.

One Continuous Profile

This diagram illustrates how to implement a series of moves as one continuous profile.


1
Set move parameters; Set profile type to 0 (for trapezoidal move).
action or query done by amplifier
action or query done by CANopen master
2
Control Word bit 4
1
Clear Control Word bit 4 (to 0).
0
3
Set Control Word bits 4 and 5 (to 1).
4
Amplifier sees bit 4 0-1 transition; sees that bit 5 is set; copies buffered move to active registers.
5
Control Word bit 6
0
Amplifier begins absolute move.
1
Amplifier begins relative move.
6
Amplifier Status Word bit 12 (to 1).
7
Clear Control Word bit 4 (to 0).
8
Amplifier clears Status Word bit 12 (to 0).
9
Another move?
yes
no
Finished.

Notes:

1. Move Parameters are described on page 108. This type of move is only supported as a trapezoidal profile.

2. Control Word bit 4 is "new setpoint." 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 "setpoint acknowledge." A value of 1 indicates the amplifier has received a setpoint and has started the move.

7. Control Word bit 4 is "new setpoint." 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.

Trapezoidal vs. S-Curve Profile: Some Design Considerations

Difference Between Trapezoidal and S-Curve Profiles

Here is a review of the differences between trajectory and S-curve profiles, and 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. 175) might notice some slowing. A higher Profile Acceleration can be applied to compensate, but watch out for amplifier and motor limits.

7.2: 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. 176). Profile velocity moves are controlled by some of the same gains and limits objects used in profile position mode.

The amplifier performs profile velocity moves in Profile Velocity Mode (Mode Of Operation [index 0x6060, p. 59] =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. 54). Motion is stopped by a high-to-low transition of the same 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. 175) 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. 176) 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. 176) 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 amplifier will generate a move as long as the halt bit (Control Word bit 8) is not set. If the halt bit is set, the amplifier will stop the move using the deceleration value.

7.3: 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.176). When the amplifier 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. 177). When the amplifier is halted, the torque ramps down at the same rate.

Profile torque moves are controlled by Current Loop Gains (index 0x60F6, p. 128).

The amplifier performs profile torque moves in Profile Torque Mode (Mode Of Operation [index 0x6060, p. 59] =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 amplifier performs calculations based on the motor’s Motor Torque Constant (index 0x6410, Sub-Index 12, p. 83) and Motor Continuous Torque (index 0x6410, Sub-Index 14, p. 84).

Starting and Stopping Profile Torque Moves

To start a move in profile torque mode, set the profile parameters. The amplifier will generate a move as long as the halt bit (Control Word bit 8) is not set. If the halt bit is set, the amplifier will stop the move using the torque_slope value.

7.4: Profile Mode Objects

Contents of this Section

This section describes the objects that control operation of the amplifier in profile position, velocity, and torque modes.

They include:

TRAJECTORY JERK LIMIT INDEX 0X2121

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	100 counts / sec^3	0 – 100,000,000	YES	RF

Description

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 STATUS INDEX 0X2252

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	See *Description*, below.	YES	-

Description

This variable gives status information about the trajectory generator. It is bit-mapped as follows:

Bit	Description
0-10	Reserved for future use.
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 amplifier 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.

TRAJECTORY GENERATOR DESTINATION POSITION INDEX 0X2122

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RO	counts	-	YES	-

Description

The position that the trajectory generator uses as its destination. Mostly useful when driving the amplifier using the pulse & direction inputs.

TARGET POSITION INDEX 0X607A

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	counts	-	YES	RF

Description

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. 178):

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 Profile Position Mode Operation, p. 164, for more information.

Profile Velocity Index 0x6081

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	0.1 counts/sec	0 – 500,000,000	YES	RF

Description

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 Profile Position Mode Operation, p. 164, for more information.

Target Velocity Index 0x60FF

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	0.1 counts/sec	-500,000,000 - 500,000,000	YES	R

Description

In profile velocity mode, this object is an input to the amplifier'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. 54) has been changed from 0 to 1.

Target Torque Index 0x6071

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	rated torque/1000	-32,768 - 32,767	YES	RF

Description

In profile torque mode, this object is an input to the amplifier's internal trajectory generator. Any change to the target torque triggers an immediate update to the trajectory generator.

Torque Command Index 0x6074

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	rated torque/1000	-32,768 - 32,767	YES	RF

Description

Output value of the torque limiting function.

Motor Rated Torque Index 0x6076

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	0.001 Nm	0 - 32,767	YES	RF

Description

Motor's rated torque (see motor name plate or documentation).

Torque Actual Value Index 0x6077

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	rated torque/1000	-32,768 - 32,767	YES	RF

Description

Instantaneous torque in the motor.

TORQUE SLOPE INDEX 0X6087

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	rated torque/1000/second	Positive integer values	YES	RF

Description

Torque acceleration or deceleration.

TORQUE PROFILE TYPE INDEX 0X6088

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RO	--	0-0	YES	RF

Description

Type of torque profile used to perform a torque change. Set to zero to select trapezoidal profile.

PROFILE ACCELERATION INDEX 0X6083

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	10 counts/sec^2	0 - 200,000,000	YES	RF

Description

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 Profile Position Mode Operation, p. 164, for more information.

PROFILE DECELERATION INDEX 0X6084

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	10 counts/sec^2	0 - 200,000,000	YES	RF

Description

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. 177) 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 Profile Position Mode Operation, p. 164, for more information.

Quick Stop Deceleration Index 0x6085

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	10 counts/sec^2	0 - 200,000,000	YES	RF

Description

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.

Motion Profile Type Index 0x6086

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	-	See *Description*, below.	YES	RF

Description

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 amplifier will not accept other values. See Profile Position Mode Operation, p. 164, 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 Profile Position Mode Operation, p. 164, for more information.

Velocity Sensor Selection Index 0x606A

Type	Access	Units	Range	Map PDO	Memory
RESERVED	-	-	0	YES	RF

Description

Reserved for future use. This object specifies how actual velocity is measured. Currently, Copley Controls amplifiers support only the use of position encoders for calculation of actual velocity. Any value other than zero will return an error.

CHAPTER

8: INTERPOLATED POSITION OPERATION

This chapter describes control of an amplifier in interpolated position mode.

Contents include:

8.1: Interpolated Position Mode Overview

Contents of this Section

This section provides an overview of the interpolated position mode.

Topics include:

Coordinated Motion

Interpolated position mode is used to control multiple coordinated axes or a single axis with the need for time-interpolation of setpoint data. In interpolated position mode, the trajectory is calculated by the CANopen master and passed to the amplifier's interpolated position buffer as a set of points. The amplifier reads the points from the buffer and performs linear or cubic interpolation between them.

Copley Controls CANopen amplifiers 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 amplifier 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 amplifier 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 amplifier 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 amplifiers 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. 189) as described here:

IP Submode	Description
0	Linear interpolation with a constant time.
-1	Linear interpolation with variable time.
-2	PVT interpolation.

Linear Interpolation with a Constant Time

In IP submode 0, the trajectory target position of each segment is written to Interpolation Position Target (index 0x60C1, Sub-Index 1, p. 190. Each time Interpolation Position Target is written to, the entire record is written to the amplifier's internal buffers. (In mode 0, Sub-Index 2 and Sub-Index 3 are ignored).

The time interval is set in Interpolation Constant Time Index (index 0x60C2, Sub-Index 1, p. 190).

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 Target, which is Sub-Index 1 of the Interpolation Data Record (index 0x60C1, p. 190). With each update to Interpolation Time (index 0x60C1, Sub-Index 2, p. 190), the entire record is written to the amplifier'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 Target (index 0x60C1, Sub-Index 1, p. 190) 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. 190), the entire record is written to the amplifier'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. 187). 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 PVT Profile Moves Using the Copley Controls Alternative Objects, p. 185.

Interpolated Position Trajectory Buffer

A typical profile contains a large number of segments. These segments must be passed to the amplifier over the CANopen network quickly enough to ensure that the next point is received before the amplifier needs it to calculate the intermediate motor positions.

To reduce the tight timing requirements of sending trajectory segments over the network, the amplifier maintains a buffer of trajectory segments in its memory. This allows the controller to send trajectory segments in bursts, rather then one at a time, as the profile is executing. The amplifier can hold 32 trajectory segments. See the Trajectory Buffer Free Count object (index 0x2011, p. 188).

Guidelines for Buffer Use

The amplifier 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 amplifier needs access to both of these segments to do the interpolation. When that segment is finished (at point P1) the amplifier needs the next segment in order to continue interpolating toward point P2.

So, between P0 and P1, the amplifier does not yet need P2. At P1, the amplifier no longer needs P0, but does need P2 to continue. Strictly speaking, there is no time when the amplifier 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. 54) and Status Word settings (index 0x6041, p. 55) settings. The transition of Control Word bit 4 from 0 to 1 causes the amplifier to start the move using the points stored in the interpolated move trajectory buffer. For an example, see PVT Profile Moves Using the Copley Controls Alternative Objects (below) and Format of Data Bytes in PVT Segment Mode, p. 188.

Ending an Interpolated Position Move

Interpolated position moves can be stopped by adding a zero time value to the buffer. This method allows the amplifier 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 Constant Time Index (index 0x60C2, Sub-Index 1, p. 190) 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. 190).
-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. 187), and Format of Data Bytes in PVT Segment Mode, p. 188.

An Interpolated position move can also be ended in one of several other ways:

- Clear bit 4 of the Control Word (index 0x6040, p. 54).
- 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 amplifier 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. 25, and SYNC and High-resolution Time Stamp Messages, p. 40.)

PVT Profile Moves Using the Copley Controls Alternative Objects

As mentioned earlier, Copley Controls CANopen amplifiers 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. 189). 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. 187).

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. 54).

6 Each time a new Trajectory Buffer Status object (index 0x2012, p. 189) 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 amplifier 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. 187) 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. 187), and Format of Data Bytes in PVT Segment Mode, p. 188.

8.2: Interpolated Position Mode Objects

Contents of this Section

This section describes the objects that control operation of the amplifier in profile position mode.

They include:

IP MOVE SEGMENT COMMAND INDEX 0X2010

Type	Access	Units	Range	Map PDO	Memory
8 Byte array	WO	-	-	YES	R

Overview

This object is used to send PVT segment data and buffer commands in interpolated position mode.

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 then 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.
3-127	Reserved for future use

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 amplifier 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 amplifier receives non-consecutive segments, an error is flagged and no further segments are accepted until the error is cleared. This allows the amplifier 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
Integer 16	RO	-	-	YES	-

Description

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
Unsigned 32	RO	-	See *Description*, below.	EVENT	-

Description

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
Unsigned 16	RO	-	-	YES	R

Description

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
Integer 16	RW	-	-	YES	R

Description

Determines which interpolation submode to use:

Submode	Description
0	Linear interpolation with a constant time.
-1	Linear interpolation with variable time.
-2	Cubic polynomial interpolation, which is also known as position, velocity, and time (PVT) interpolation. 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 Interpolation Submode Select.

INTERPOLATION DATA RECORD INDEX 0X60C1

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	YES	R

Description

This object is used to send interpolation data to the amplifier's interpolation buffer.

INTERPOLATION POSITION TARGET INDEX 0X60C1, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	Counts	-	YES	R

Description

A target position. Used in all three interpolation modes.

INTERPOLATION TIME INDEX 0X60C1, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	milliseconds	-	YES	R

Description

The time interval of the move segment that ends with the Interpolation Position Target (Sub-Index 1). 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.

INTERPOLATION VELOCITY INDEX 0X60C1, SUB-INDEX 3

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	0.1 counts/sec	-	YES	R

Description

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 CONSTANT TIME INDEX 0X60C2

Type	Access	Units	Range	Map PDO	Memory
Record	RW	-	-	YES	R

Description

Used only in interpolation mode 0 (linear interpolation with a constant time). Defines the segment interval.

INTERPOLATION CONSTANT TIME INDEX INDEX 0X60C2, SUB-INDEX 1

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	RW	milliseconds	-	YES	R

Description

This object sets the constant time that is associated with each trajectory segment in interpolation mode 0.

INTERPOLATION CONSTANT TIME UNITS INDEX 0X60C2, SUB-INDEX 2

Type	Access	Units	Range	Map PDO	Memory
Unsigned 8	R0	-	-	YES	R

Description

This object which always return the value -3 indicating that Interpolation Constant Time index is always formatted in units of milliseconds..

APPENDIX

A: ALTERNATIVE CONTROL SOURCES

This chapter describes control of an amplifier by a source other than the CANopen network.

Contents include:

A.1: Alternative Control Sources Overview

Typically, when a Copley amplifier is used on a CANopen network, the CANopen master uses the network to send commands that drive the amplifier's position, velocity, or current loop.

Alternately, an amplifier 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 amplifier's internal generator or a Copley Virtual Machine (CVM) program. Use the Indexer Register Values object (index 0x2600, p. 198) to read and write the CVM Indexer program registers.

An amplifier can also run in camming mode to execute moves programmed in camming tables. The Camming Configuration object (index 0x2360, p. 196) and several other objects described in this chapter are used to configure and operate the amplifier 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 60.

Other objects affect the amplifier under alternative control sources. They are described in the next section.

A.2: Alternative Control Source Objects

Contents of this Section

This section describes objects related to alternate sources of amplifier control.

They include:

Micro-Stepping Rate Index 0x21C1

Type	Access	Units	Range	Map PDO	Memory
Integer 16	RW	degrees / sec	0 - 32,767	YES	RF

Description

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. 60), code 42.

Analog Reference Scaling Factor Index 0x2310

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	See *Description*, below.	-	YES	RF

Description

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. 60, 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
Integer 16	RW	millivolts	-	YES	RF

Description

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
Integer 16	RW	millivolts	-	YES	RF

Description

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
Integer 16	RW	millivolts	-	YES	RF

Description

The analog reference input is subject to a non-linear adjustment to clip reading around zero. This object defines the size of that window.

PWM Input Frequency Index 0x2322

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	10 Hz	-	YES	RF

Description

This is the frequency of the PWM for use only in UV commutation mode (Desired State object [index 0x2300, p. 60[= 5).

Function Generator Configuration — Index 0x2330

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	See Description, below.	YES	RF

Description

Configures the amplifier'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	Describe
0	None (disabled)
1	Square wave.
2	Sine wave.

Note that the amplifier is placed under control of the function generator by setting the Desired State object (index 0x2300, p. 60) to one of the following values:
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
Integer 16	RW	Hz	0 – 32,767	YES	RF

Description

This object gives the frequency of the internal function generator.

Function Generator Amplitude — Index 0x2332

Type	Access	Units	Range	Map PDO	Memory
Integer 32	RW	See *Description*, below	See *Description*, below	YES	RF

Description

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
Integer 16	RW	0.1 percent	0 – 1,000	YES	RF

Description

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
Unsigned 16	RW	-	-	YES	RF

Description

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. 196). If clear, use digital command input as configured in using Copley's CME 2 software camming controls or Input Pin States (index 0x2190, p. 95)	
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 amplifier 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
Unsigned 16	RW	Master Counts.	0 - 32,767	YES	RF

Description

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
Unsigned 16	RW	Master Counts.	0 - 32,767	YES	RF

Description

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
Integer 16	RW	0.1 counts/second	-500,000,000 - 500,000,000	YES	RF

Description

Virtual master encoder velocity for camming mode.

TRACE BUFFER RESERVED SIZE INDEX 0X250A

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	words	0 - 2048	YES	R

Description

The number of RAM words in the amplifier Trace Buffer to reserve for Trace Buffer Data (such as CAM tables).

TRACE BUFFER ADDRESS INDEX 0X250B

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	words	0 - 2048	YES	R

Description

An offset from the beginning of the memory reserved for Trace Buffer Data (index 0x250C, p. 197). 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
Variable length	RW	-	-	YES	R

Description

The first value written to this object will be stored in trace buffer RAM at the location specified by Trace Buffer Address (index 0x250B, p. 197). 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	Units	Range	Map PDO	Memory
Array	RW	-	-	NO	R

Description

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).

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	Units	Range	Map PDO	Memory
Unsigned 32	RW	-	-	YES	R

Description

One sub-index object for each Indexer program register.

A.3: Running CAM Tables from RAM

Normally, Cam Tables are stored in the amplifier'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 amplifier RAM.

(For a full description of camming, see the *Copley Camming User Guide.*)

Cam Tables in Amplifier 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 amplifier 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 amplifier 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. 201).

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 amplifier will look at address 0 for a metadata pair. When configured to run Cam Table 1, the amplifier 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 Trace Buffer Reserved Size (index 0x250A, p. 197) 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. 197) the Cam Table's initial offset value.

Write a series of values to Trace Buffer Data (index 0x250C, p. 197).

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 amplifier increments the offset pointer in Trace Buffer Address.

3. Configure the Camming Parameters

To configure the amplifier to run Cam Tables from RAM, set bit 7 in the Camming Configuration object (index 0x2360, p. 196). Set other parameters as needed.

4. Run a Cam Table from RAM

Set the Desired State object (index 0x2300, p. 60) 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.

APPENDIX
B: TRACE TOOL

This chapter describes the CANopen interface to the Copley Controls Trace Tool.

Contents include:

B.1: Trace Tool Overview

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.

B.2: Trace Tool Objects

Contents of this Section

This section describes objects related to the Trace Tool. They include:

TRACE CHANNELS INDEX 0X2500

Type	Access	Units	Range	Map PDO	Memory
Array	RW	-	See description, below.	NO	-

Description

This object uses 6 sub-indices configure up to 6 trace channels.

TRACE CHANNELS INDEX 0X2500, SUB-INDEX 1-6

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	-	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.
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
Continued…	

…continued:

Code	Trace Variable
36	Motor phase angle (1 degree units)
37	Amplifier temperature (degrees C)
38	Amplifier Manufacturer Status Register (index 0x1002, p. 56)
39	Amplifier event latch word
40	Hall sensor state
41	Position Capture Status Register (index 0x2401, p. 160)
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 amplifiers with analog encoder support.
47	The analog encoder cosine input. Only valid for amplifiers 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 Status

Index 0x2501

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	R0	-	-	YES	R

Description

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
Unsigned 32	RO	nanoseconds	-	NO	R

Description

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 Period

Index 0x2505

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RW	See description.	-	NO	R

Description

The trace period, in integer multiples of the Trace Reference Period (0x2502).

TRACE SAMPLE COUNT INDEX 0X2503

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	-	YES	R

Description

Returns the number of samples collected so far.

TRACE MAX SAMPLES INDEX 0X2504

Type	Access	Units	Range	Map PDO	Memory
Unsigned 16	RO	-	-	NO	R

Description

Get maximum samples. 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. For an accurate value, the trace variables should be set first; then the maximum number of samples available may be requested.

TRACE TRIGGER CONFIGURATION INDEX 0X2506

Type	Access	Units	Range	Map PDO	Memory
Unsigned 48	RW	-	-	NO	R

Description

Set/get the trace trigger configuration.

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>0-3</td><td colspan="2">Channel number to trigger on (if applicable).</td></tr>
<tr><td>4-7</td><td colspan="2">Reserved.</td></tr>
<tr><td>8-11</td><td colspan="2">Trigger type (may be interpreted differently for some trigger types):</td></tr>
<tr><td></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td></td><td>0</td><td>No trigger in use.</td></tr>
<tr><td></td><td>1</td><td>Trigger as soon as the selected channel's input is greater then or equal to the trigger level.</td></tr>
<tr><td></td><td>2</td><td>Trigger as soon as the selected channel's input is less then or equal to the trigger level.</td></tr>
<tr><td></td><td>3</td><td>Trigger when the selected channel's input changes from below to above the trigger level.</td></tr>
<tr><td></td><td>4</td><td>Trigger when the selected channel's input changes from above to below the trigger level.</td></tr>
<tr><td></td><td>5</td><td>Trigger when any selected bits in the channel value are set. The bits are selected using the trigger level value as a mask.</td></tr>
<tr><td></td><td>6</td><td>Trigger when any selected bits in the channel value are clear. The bits are selected using the trigger level value as a mask.</td></tr>
<tr><td></td><td>7</td><td>Trigger any time the selected channel value changes.</td></tr>
<tr><td></td><td>8</td><td>The trigger level mask selects one or more bits in the Manufacturer Status Register (index 0x1002, p. 56). 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.</td></tr>
<tr><td></td><td>9</td><td>Like type 8, but the trigger occurs when the bit(s) change from 1 to 0.</td></tr>
<tr><td></td><td>10</td><td>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.</td></tr>
<tr><td>12-14</td><td colspan="2">Reserved.</td></tr>
<tr><td>15</td><td colspan="2">If set, take one sample per trigger event.</td></tr>
</table>

TRACE DELAY INDEX 0X2507

Type	Access	Units	Range	Map PDO	Memory
Signed 16	RW	-	-	NO	R

Description

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	Units	Range	Map PDO	Memory
Unsigned 16	RW	-	-	YES	R

Description

Write 0 to stop trace collection or a non-zero value to restart it.

TRACE DATA INDEX 0X2509

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

Description

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.

APPENDIX

C: OBJECTS BY FUNCTION

This chapter lists the objects in functional groups. (The groupings match the titles of the sections of this manual that describe the objects.)

They include:

Objects that Define SDOs and PDOs

Network Management Objects

Device Control And Status Objects

Error Management Objects

Basic Amplifier Configuration Objects

Basic Motor Configuration Objects

Real-time Amplifier and Motor Status Objects

Digital I/O Configuration Objects

Position Loop Configuration Objects

Velocity Loop Configuration Objects

Current Loop Configuration Objects

Stepper Mode Objects

Homing Mode Operation Objects

Profile Mode Objects

Interpolated Position Mode Objects

Alternative Control Source Objects

Trace Tool Objects

APPENDIX

D: 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.

CANopen Programmer's Manual
P/N 95-00271-000

Revision 5

October, 2008