Motors | Automation | Energy | Transmission & Distribution | Coatings

EtherCAT

SCA06

User's Guide

EtherCAT User's Guide

Series: SCA06

Language: English

Document Number: 10005376721 / 00

Build 664

Publication Date: 01/2018

CONTENTS

ABOUT THE MANUAL

This manual supplies the necessary information for the operation of the SCA06 servo drive using the EtherCAT protocol. This manual must be used together with the SCA06 user's manual and programming manual.

1 EQUIPMENT CHARACTERISTICS IN ETHERCAT NETWORK

Following they are presented the main characteristics for the EtherCAT communication accessory for SCA06 servo drive.

- It allows the equipment to operate as slave for the EtherCAT communication.
- It has two Ethernet ports, which allows to connect several equipments in sequence or in a ring topology.
- It uses CoE (*CANopen over EtherCAT*) protocol in the application layer.
- It makes available several objects according to CiA 402 specification - *Device Profile Drives and Motion Control*.
- There is a xml file that is supplied with the product to configure the network master.

2 ETHERCAT OVERVIEW

Following it is presented general information about the EtherCAT tecnology.

2.1 ETHERCAT TECHNOLOGY

EtherCAT (**Ether**net for **C**ontrol **A**utomation **T**echnology) is a Real Time Ethernet technology introduced originally in 2003 by Beckhoff. In 2007, it was integrated into the international fieldbus standards IEC 61158. The EtherCAT Technology Group (ETG) promotes EtherCAT and is responsible for its continued development.

EtherCAT communication technology is based on 100BASE-TX Ethernet physical layer. An EtherCAT bus consists of a master system and up to 65535 slave devices, connected together with standard Ethernet cabling.

In order to improve network performance, instead of sending one Ethernet package at a time to each node, EtherCAT network takes a different approach, employing ”on the fly” processing hardware. The frame, containing data to several slaves, is initially sent by the master to the first slave on the network. The slave device process the incoming Ethernet frames directly, extract or insert relevant data and transfer the frame to the next EtherCAT slave device, without waiting the entire frame to be received. The last slave device in the bus segment sends the fully processed frame back, so that it is returned by the first slave to the master as a kind of response frame, which maximize the use of the full duplex Ethernet bandwidth. This approach makes data update between master and slaves really fast and deterministic, ideal for application fields like machine controls.

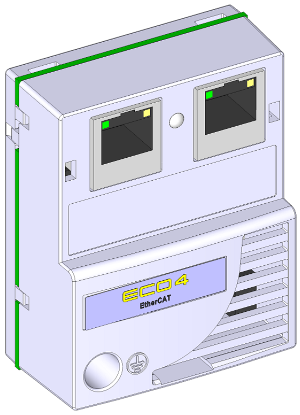
2.2 DEVICE PROFILES

The EtherCAT communication allows different protocols to be used in the application layer. For servo drive SCA06, the CANopen over EtherCAT (CoE) device profile is available, where the CANopen protocol is applied to EtherCAT. Different communication objects available to CANopen are also used via EtherCAT, like Service Data Objects (SDO), Process Data Objects (PDO) and the Object Dictionary structure to manage the parameters.

3 INTERFACES DESCRIPTION

SCA06 servo drive uses ECO4 accessory to provide an EtherCAT interface operating as EtherCAT slave in the product.

3.1 ETHERCAT INTERFACE ACCESSORY



- Supplied items:
  - Installation guide.
  - EtherCAT interface module.
- It has an integrated switch to daisy chain connection.

NOTE!
ECO4 accessory can only be connected at Slot 2.

3.2 CONNECTORS

Two RJ45 connectors are available for network connection:

- XC1 (IN): used to connect to the segment that comes from master side.
- XC2 (OUT): used to connect to the segment that leads to the other slaves (or not connected, if it is the last device on the network).

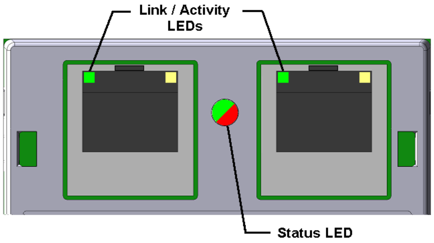


Figure 3.1: *Connectors and LEDs of EtherCAT accessory*

The connectors pinout follows the Fast Ethernet 100BASE-TX standard, using two pairs of cables (standard connection, straight-through) for transmitting and receiving data.

EtherCAT accessory makes the protective earth connection through the fixing screw. The connectors frame, which are usually connected to the cable shield, are short circuited, and a RC circuit connects them to the protective earth.

3.3 INDICATION LEDS

EtherCAT accessory has a LED indicator on each Ethernet port, and a bicolor LED for diagnostics (Status). These LEDs have the following functions and indications:

Table 3.1: *EtherCAT indication LEDs*

LED	Color	Function
Link/Activity (IN/OUT)	Green	Indication of link and activity, one for each Ethernet port.
Status	Bicolor (Green/Red)	Module state; indicate the state of EtherCAT slave.

Table 3.2: *Link/Activity LED*

State	Description
Off	No link or equipment not powered.
Solid green	With link, no activity.
Blinking	With link and with activity.

Table 3.3: *Status LED*

Color	State	Description
Green	Off	INIT (or not powered)
	Alternating on/off	PRE-OPERATIONAL.
	1 blink and 1 interval	SAFE-OPERATIONAL.
	Solid	OPERATIONAL.
Red	Off	No error (or not powered)
	Alternating on/off	Interface initialization error.
	1 blink and 1 interval	Communication error, slave left the operational state.

4 NETWORK INSTALLATION

Following are recommendations related to equipment installation in an EtherCAT network. Details on the characteristics of the components used for installation can be obtained along the ETG.

4.1 CABLE

Recommended characteristics for the cable:

- Standard Ethernet cable, 100Base-TX (FastEthernet), CAT 5e.
- Shielded cable.
- Maximum length between devices: 100m.

4.2 NETWORK TOPOLOGY

To connect SCA06 servo drive in an EtherCAT network, it is necessary to observe the Ethernet connector on the device used for connection.

- The network always start by the EtherCAT master.
- The X1 connector (IN) should always be connected to the network segment leading to the EtherCAT master.
- The X2 connector (OUT) should always be connected to the network segment leading to other EtherCAT slaves.
- When connected in a ring topology for redundancy, the X2 connector (OUT) of the last slave can be connected to the second port of the EtherCAT master, otherwise it should be leave disconnected.

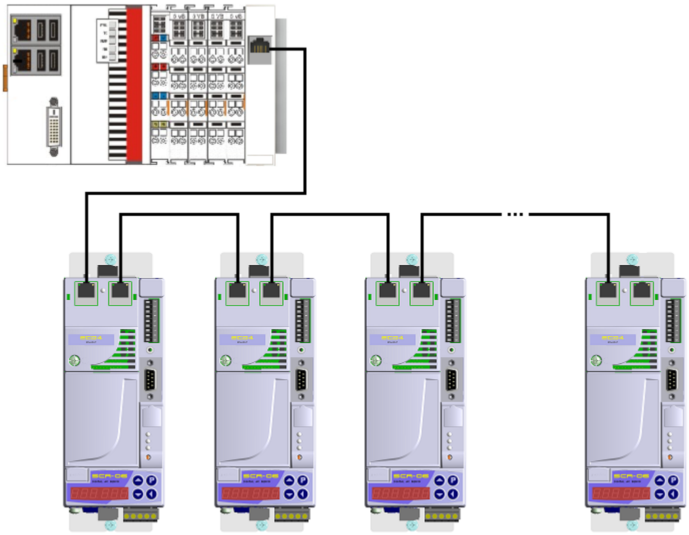


***Figure 4.1:** EtherCAT network topology*

4.3 RECOMMENDATIONS FOR GROUNDING CONNECTION AND CABLE ROUTING

The correct connection with the ground decreases problems caused by interference in an industrial environment. The following are some recommendations about grounding and cable routing:

- Always use shielded twisted pair Ethernet cables and connectors with metallic casing.

- Use clamps on the main grounding point, allowing better contact surface between the cable shield and ground.
- Avoid the cable connection on multiple grounding points, especially where there are grounds with different potentials.
- Pass signal cables and communication cable in dedicated pathways. Prevent the passage of these cables next to power cables.

5 PARAMETERIZATION

Next, the SCA06 servo drive parameters related to the EtherCAT communication will be presented.

5.1 SYMBOLS FOR THE PROPERTIES DESCRIPTION

- **RO** Read-only parameter
- **RW** Read/write parameter
- **CFG** Parameter that can be changed only with a stopped motor
- **ECAT** Parameter visible on the HMI if the product has the EtherCAT interface installed

P0202 – OPERATION MODE

Range:	1 = Torque Mode 2 = Speed Mode 3 = No function 4 = Ladder Mode 5 = CANopen/DeviceNet/EtherCAT 6 = Profibus DP/Ethernet	**Default:** 2
Properties:	RW	

Description:

This parameter defines the operating mode for SCA06 servo drive, allowing to program which mode is desired for the motor control, and what is the control source for the device.

For the equipment to be controlled over EtherCAT network, you must select the option for operation via EtherCAT network in this parameter. If this mode is set, the type of control, as well as commands and references for operation of the product will be given via EtherCAT, using the objects defined in the object dictionary. Among the main objects used for control and monitoring equipment, the following can be mentioned:

- 6040h: ControlWord
- 6041h: StatusWord
- 6060h: Mode of operation
- 6063h: Position actual value
- 607Ah: Target position
- 60FFh: Target velocity
- 6071h: Target Torque

A detailed description of these and other objects can be obtained in section 6. For details on other operation modes, refer to the SCA06 user's manual.

✓ **NOTE!**

- The control of equipment using these objects is only possible by selecting the desired option in this parameter, but EtherCAT communication can be used in any operating mode.
- The CANopen interface uses internally the same objects as EtherCAT interface. Therefore, it is not possible to control the equipment simultaneously by both interfaces.

P0662 – COMMUNICATION ERROR BEHAVIOR

Range:	0 = Show Alarm 1 = Generate Fault 2 = Run STOP function 3 = Disable drive	**Default:** -
Properties:	RW	

Description:

This parameter allows selecting which action should be executed by the equipment, if it is controlled via network and a communication error is detected.

Table 5.1: *Option for parameter P0662*

Option	Description
0 = Show Alarm	If a communication error occurs, only indicates alarm in the keypad. If the communication is reestablished, the alarm indication is automatically removed.
1 = Generate Fault	Instead of an alarm, a communication error causes a fault in the equipment, and it is necessary to reset the failure to return to normal operation.
2 = Run STOP function	It will indicate an alarm along with the execution of the STOP command. To start operation again, it is necessary to run the fault reset or disable the drive.
3 = Disable drive	It will indicate an alarm along with the execution of the disable command.

The following events are considered communication errors:

EtherCAT communication:

- Alarm A145/Fault F45: EtherCAT interface error.
- Alarm A146/Fault F46: communication error with EtherCAT master.

P0850 – FIRMWARE REVISION FOR ETHERCAT ACCESSORY

Range:	0 a 65535	**Default:** -
Properties:	ro, ecat	

Description:

EtherCAT accessory has an internal processor responsible by running the communication routines and perform data exchange with SCA06 servo drive. This parameter identify the firmware revision of this accessory.

P0851 – ETHERCAT ACCESSORY STATUS

Range:	0 = Inactive 1 = Access error 2 = Watchdog error 3 = Offline 4 = Online	**Default:** -
Properties:	ro, ecat	

Description:

Indicates the status of EtherCAT Interface, related to the data exchange between network master and SCA06 servo drive.

Table 5.2: *Indications of parameter P0851*

Value	Description
0 = Inactive	Interface not installed or not recognized by the device.
1 = Access error	Error during the initialization procedure of EtherCAT module.
2 = Watchdog error	Error in the watchdog mechanism between accessory and device, responsible by checking if data update between both are running successfully.
3 = Offline	EtherCAT module running properly, but there is no cyclic data exchange with the network master.
4 = Online	EtherCAT module running properly, with cyclic data exchange with the network master.

P0852 – ETHERCAT LINK STATUS

Range:	0000h ... FFFFh	**Default:** -
Properties:	ro, ecat	

Description:

Parameter that provides information about the state of communication for each available Ethernet port on the module. Uses a binary field, where each bit represents a different information:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	Reserved				Port 2: Communication Established	Port 2: Loop Active	Port 1: Communication Established	Port 1: Loop Active	Reserved		Link 2 OK	Link 1 OK	Reserved			

Table 5.3: *P0852 bits indication*

Bit	Value/Description
Bit 0...3	Reserved.
Bit 4 Link 1 OK	**0**: Port 1: No link. **1**: Port 1: Link active.
Bit 5 Link 2 OK	**0**: Port 2: No link. **1**: Port 2: Link active.
Bit 6...7	Reserved.
Bit 8 Port 1: Loop active	**0**: Port 1: No loop. **1**: Data loop active at port 1, to return response to the master.
Bit 9 Port 1: Communication Established	**0**: No communication at port 1. **1**: Communication established at port 1.
Bit 10 Port 2: Loop active	**0**: Port 2: No loop. **1**: Data loop active at port 2, to return response to the master.
Bit 11 Port 2: Communication Established	**0**: No communication at port 2. **1**: Communication established at port 2.
Bit 12...15	Reserved.

P0853 – ETHERCAT SLAVE STATUS

Range:	0 = Inactive 1 = Initialization 2 = Pre-operational 3 = Reserved 4 = Safe-operational 5 = Reserved 6 = Reserved 7 = Reserved 8 = Operational	**Default:** -
Properties:	ro, ecat	

Description:

Indicate the slave state, according to the EtherCAT state machine.

Table 5.4: *Indications of parameter P0853*

Valor	Descrição
0 = Inactive	Interface is inactive.
1 = Initialization	EtherCAT in the initialization state, waiting configurations and commands from the master to allow data exchange via mailbox.
2 = Pre-operational	Initialization successful, waiting commands from the master to configure the communication and starting cyclic data exchange.
4 = Safe-operational	Master starts cyclic data reading, but data writing is not performed.
8 = Operational	Mastes runs cyclic data update for reading and writing data areas.

P0855 – CONFIGURED TXPDO

Range:	0 to 65535	**Default:** -
Properties:	ro, ecat	

Description:

Indicates the index of the transmission PDO selected for communicate with the equipment. The device has 4 transmit PDOs, responsible for sending data to the master, but only one can be active for communication. Each TxPDO has a default mapping, but the master can also change the PDO mapping as desired.

P0856 – NUMBER OF BYTES CONFIGURED AT TXPDO

Range:	1 to 32	**Default:** -
Properties:	ro, ecat	

Description:

Indicates the data size, in bytes, transmitted to the master via the TxPDO. The number of bytes is formed by adding the size of each object mapped in TxPDO and is updated only when the master initiates the cyclic data communication with the equipment.

P0857 – CONFIGURED RXPDO

Range:	0 to 65535	**Default:** -
Properties:	ro, ecat	

Description:

Indicates the index of the receive PDO selected for communicate with the equipment. The device has 4 receive PDOs, responsible for receiving data from the master, but only one can be active for communication. Each RxPDO has a default mapping, but the master can also change the PDO mapping as desired.

P0858 – NUMBER OF BYTES CONFIGURED AT RXPDO

Range:	1 to 32	**Default:** -
Properties:	ro, ecat	

Description:

Indicates the data size, in bytes, received from the master via the RxPDO. The number of bytes is formed by adding the size of each object mapped in RxPDO and is updated only when the master initiates the cyclic data communication with the equipment.

P0859 – DATA UPDATE INTERVAL

Range:	0.0 a 1000.0 ms	**Default:** -
Properties:	ro, ecat	

Description:

Once in the operational state, the master must send telegrams to update the operating data at regular intervals. This parameter indicates the time between the last two data updates, allowing user to monitor the period for execution of this function.

6 NETWORK OPERATION

Following it shows operating characteristics of the SCA06 servo drive using EtherCAT interface.

6.1 ETHERCAT COMMUNICATION ARCHITECTURE

The following figure presents the SCA06 servo drive architecture for network data exchange.

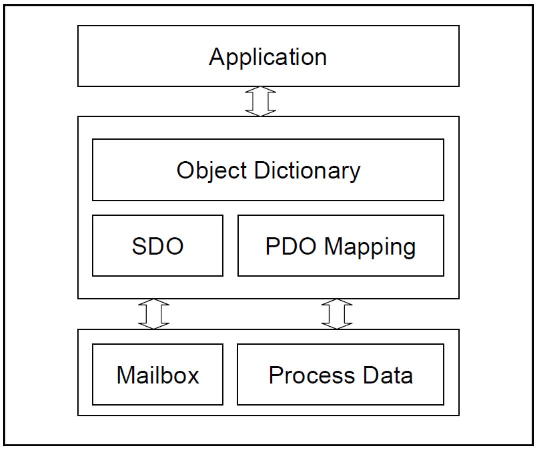


Figure 6.1: *EtherCAT communication architecture*

The application layer protocol used by the EtherCAT is based on CANopen DS301 protocol, called CANopen over EtherCAT (CoE). Among the main structures used in communication, it can be highlighted:

Object Dictionary
All the data exchange is based on the object dictionary, which is a list of available data for communication configuration and operation of the equipment.

PDOs
PDOs are used to define and transmit process data (cyclic data) communicated between master and slave for control and monitoring equipment.

SDO
The equipment has an SDO server that enables the exchange of acyclic data, for setting up communication and parameterization. Data exchanged via SDO utilize the mailbox structure for sending commands and receiving the response by the network master.

6.2 ETHERCAT STATE MACHINE

Every EtherCAT slave has a state machine responsible for defining the activities to be performed between master and slave. After power on, the slave goes to the startup state, and other state transitions typically occur as requested by network master.

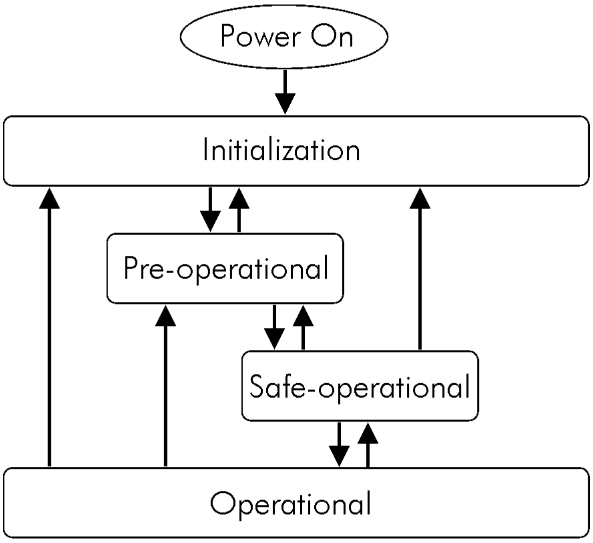


Figure 6.2: *EtherCAT state machine*

Table 6.1: *States and transitions for EtherCAT state machine*

State / Transition	Description
Initialization	▪ Standard state after initialization. ▪ Mailbox communication or process data exchange is not allowed.
Initialization → Pre-operational	▪ Master configures logical address for slave. ▪ Master configures mailbox Sync Manager. ▪ Master initiates synchronizations with slaves (Distributed Clock). ▪ Master requests pre-operational state.
Pre-operational	▪ Mailbox communication is possible, but no process data communication.
Pre-operational → Safe-operational	▪ Master configures Sync Manager and FMMU for process data. ▪ Master sends commands via SDO for slave configuration, including PDO mapping. ▪ Master requests safe-operational state.
Safe-operational	▪ Mailbox communication is possible. ▪ Slave sends input process data to master, but output data from master to slave remains with initial values.
Safe-operational → Operational	▪ Master sends valid output values. ▪ Master requests operational state.
Operational	▪ Mailbox communication is possible. ▪ Process data communication (inputs and outputs) is possible, according to PDO mapping.

6.3 SYNC MANAGERS

Sync Managers are responsible for controlling access to data sent and received over the network, and ensure data consistency. There are 4 Sync Managers, each one responsible for managing a different data type:

- Sync Manager 0: responsible for writing mailbox data.
- Sync Manager 1: responsible for reading mailbox data.
- Sync Manager 2: control access to RxPDO data. Object 1C12h is used to indicate which RxPDO this Sync Manager is associated with.
- Sync Manager 3: control access to TxPDO data. Object 1C13h is used to indicate which TxPDO this Sync Manager is associated with.

6.4 SYNCHRONIZATION MODES

Data update may be performed using two synchronization mechanisms:

- SM-Synchronous: data update is based on Sync Managers 2 and 3. When a telegram is received by interface for data update, at this momment it runs the internal data update routine by the slave.
- DC-Synchronous: data update is synchronous, using distributed clock mechanism.

✓ **NOTE!**
The minimum cycle allowed for SCA06 servo drive data updating over the network is 1ms.

6.5 PROCESS DATA - PDO

The process data (cyclic data), used to control and monitor the equipment during operation, are transmitted and received by the slave using PDOs. There are two types of PDOs:

- Receive PDO - RxPDO: receive data from the master, as commands and references. The device provides four RxPDOs, each with a standard mapping, but only one RxPDO can be enabled at a time.
- Transmit PDO - TxPDO: transmit data to the master, as states and process variables. The device provides four TxPDOs, each with a standard mapping, but only one TxPDO can be enabled at a time.

6.5.1 PDOs Mapping

The mapping defines the content transmitted by each PDO, based on the objects dictionary of the slave. It is possible to map up to 8 objects for each PDO. This mapping is done using the objects 1600h to 1603h for RxPDOs and 1A00h to 1A03h for TxPDOs.

Table 6.2: *Standard mapping for RxPDOs*

RxPDO	Mapped Objects	Objects Description	Size	Total
RxPDO 1 (Object 1600h)	6040h	Control Word	16 bits	13 Bytes
	607Ah	Target Position	32 bits	
	60FFh	Target velocity	32 bits	
	6071h	Target torque	16 bits	
	6060h	Modes of operation	8 bits	
RxPDO 2 (Object 1601h)	6040h	Control Word	16 bits	6 Bytes
	607Ah	Target Position	32 bits	
RxPDO 3 (Object 1602h)	6040h	Control Word	16 bits	6 Bytes
	60FFh	Target velocity	32 bits	
RxPDO 4 (Object 1603h)	6040h	Control Word	16 bits	4 Bytes
	6071h	Target torque	16 bits	

Table 6.3: Standard mapping for TxPDOs

TxPDO	Mapped Objects	Objects Description	Size	Total
TxPDO 1 (Object 1A00h)	6041h	Status Word	16 bits	13 Bytes
	6064h	Position actual value in user units	32 bits	
	606Ch	Velocity actual value	32 bits	
	6077h	Torque actual value	16 bits	
	6061h	Modes of operation display	8 bits	
TxPDO 2 (Object 1A01h)	6041h	Status Word	16 bits	6 Bytes
	6064h	Position actual value in user units	32 bits	
TxPDO 3 (Object 1A02h)	6041h	Status Word	16 bits	6 Bytes
	606Ch	Velocity actual value	32 bits	
TxPDO 4 (Object 1A03h)	6041h	Status Word	16 bits	4 Bytes
	6077h	Torque actual value	16 bits	

6.5.2 PDO Assignment

Besides mapping, the PDO must be assingned to a Sync Manager to perform the data transfer during operation. This is done through the following objects:

- 1C12h - RxPDO assign: program which RxPDO will be assingned to the Sync Manager 2, for process data reception. Must be written the number of the RxPDO object mapping (1600h ... 1603h) to relate the Sync Manager to the PDO.
- 1C13h - TxPDO assign: program which RxPDO will be assingned to the Sync Manager 3, for process data transmission. Must be written the number of the TxPDO object mapping (1A00h ... 1A03h) to relate the Sync Manager to the PDO.

✓ **NOTE!**
Both mapping and selection of PDOs can only be performed during the pre-operational state, before beginning the process data exchange.

6.6 MAILBOX

Besides PDOs, responsible for transmitting and receiving cyclic data, EtherCAT communication also enables the data exchange via mailbox, typically used for acyclic data exchange between master and slave.

Different data sets can be transmitted via mailbox. SCA06 servo drive enables the data exchange using an SDO server that responds to requests from the EtherCAT master as specified by the EtherCAT protocol.

The following services are available via the mailbox:

Table 6.4: Available Services via mailbox

Name	Description
SDO Download Normal	Write value to an object of the dictionary.
SDO Upload Normal	Read value from an object of the dictionary.
Abort SDO Transfer	Aborts execution of the service on failure/invalid data.
Get Object Dictionary List	Read the list of available objects.
Get Object Description	Gets information about an object (index).
Get Entry Description	Gets information about an object (sub-index).

6.7 ETHERCAT SLAVE INFORMATION

Every EtherCAT slave has a configuration file in XML format, containing information about the objects, services and slave configurations. This file, known as EtherCAT Slave Information (ESI) is provided on the CD shipped with the product.

Information such as the list of objects for communication, configuration and FMMUs Sync Managers, standard mapping for PDOs, are present in this file.

✓ **NOTE!**
It is important to note if the XML configuration file is compatible with the firmware version of SCA06 servo drive.

7 OBJECT DICTIONARY

The object dictionary is a list containing several equipment data which can be accessed via EtherCATnetwork. An object of this list is identified by means of a 16-bit index, and it is based in that list that all the data exchange between devices is performed. Any object of this list can be accessed via SDO, and objects related with the control, operation and monitoring of the equipment can be mapped to cyclical communication via PDOs.

The objects are divided into groups with different functions. The main groups are:

Table 7.1: *Groups for Object Dictionary*

Index	Objects	Description
1000h – 1FFFh	Communication objects	They are objects common to all the CANopen devices. They contain general information about the equipment and also data for the communication configuration.
2000h – 5FFFh	Manufacturer specific objects	In this range, each CANopen equipment manufacturer is free to define which data those objects will represent.
6000h – 9FFFh	Standardized device objects	This range is reserved to objects that describe the behavior of similar equipment, regardless of the manufacturer.

The other indexes that are not referred in this list are reserved for future use.

✓ **NOTE!**
The configuration file in XML brings the full list of objects in the equipment, as well as their properties with respect to type, default values, and mapping for PDOs

7.1 MANUFACTURER SPECIFIC

For indexes from 2000h to 5FFFh, each manufacture is free to define which objects will be present, and also the type and function of each one. In the case of the SCA06, the whole list of parameters was made available in this object range. It is possible to operate the SCA06 by means of these parameters, carrying out any function that the drive can execute. The parameters were made available starting from the index 2000h, and by adding their number to this index their position in the dictionary is obtained. The next table illustrates how the parameters are distributed in the object dictionary.

Table 7.2: *Object List – Manufacturer Specific*

Index	Sub-index	Name	Type	Access	PDO Mapping
2000h	0	P0000 – Access parameter	INT16	rw	No
2002h	0	P0002 – Motor speed	INT16	ro	Yes
2003h	0	P0003 – Motor current	INT16	ro	Yes
2004h	0	P0004 – DC link voltage	INT16	ro	Yes
⋮					
2077h	0	P0119 – Current reference	INT16	rw	Yes
2079h	0	P0121 – Speed reference	INT16	rw	Yes
⋮					

In order to be able to program the device operation correctly via the network, it is necessary to know its operation through the parameters.

Besides parameters, the SCA06 servo drive also has the following objects:

- 3000h – digital inputs.
- 3001h – digital outputs.

7.1.1 Object 3000h – Digital inputs

This object allows reading the digital inputs status from SCA06 servo drive. The sub-index is used to identify the desired set of inputs (standard product or expansions), and each bit of the sub-index represents one digital input.

Table 7.3: *Object 3000h - Digital Inputs*

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
3000h	0	Number Of Entries	UINT16	RO	No	4
	1	Digital Inputs - Standard product	UINT16	RO	Yes	-
	2	Digital Inputs - Slot 1	UINT16	RO	Yes	-
	3	Digital Inputs - Slot 2	UINT16	RO	Yes	-
	4	Digital Inputs - Slot 3	UINT16	RO	Yes	-

7.1.2 Object 3001h – Digital outputs

This object allows writing the digital output values to servo drive SCA06. The sub-index is used to identify the desired set of outputs (standard product or expansions), and each bit of the sub-index represents one digital output.

Table 7.4: *Object 3001h - Digital Outputs*

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
3001h	0	Number Of Entries	UINT16	RO	No	4
	1	Digital Outputs - Standard Product	UINT16	RW	Yes	0
	2	Digital Outputs - Slot 1	UINT16	RW	Yes	0
	3	Digital Outputs - Slot 2	UINT16	RW	Yes	0
	4	Digital Outputs - Slot 3	UINT16	RW	Yes	0

7.2 DEVICE PROFILE

The SCA06 servo drive follows the CiA DPS 402 – Device Profile Drives and Motion Control description. This document describes a set of objects that must be common for drives, regardless of the manufacturer. This makes the interaction between devices with the same function easier (as for servo drive), because the data, as well as the device behavior, are made available in a standardized manner. For those objects the indexes from 6000h to 9FFFh were reserved.

Refer to the section 8 for a detailed description of which objects are available for this range of the object dictionary.

8 CIA 402 DRIVE PROFILE

The objects that are common for drives, defined by the CANopen specification in the CiA DSP 402 document, are described in this section

The following table presents the list of the available objects for the servo drive SCA06, divided according to the different operation modes of the device.

Table 8.1: *Object List – Drive Profile*

Index	Sub-index	Name	Type	Access	PDO Mapping
Device Control					
6040h	0	ControlWord	UINT16	rw	Yes
6041h	0	StatusWord	UINT16	ro	Yes
6060h	0	Modes of operation	INT8	rw	Yes
6061h	0	Modes of operation display	INT8	ro	Yes
6502h	0	Supported drives modes	UINT32	ro	Yes
Factor Group					
608Fh	0	Position encoder resolution	UINT32	rw	No
6091h	0	Gear ratio	UINT32	rw	No
6092h	0	Feed constant	UINT32	rw	No
Position Control Function					
6063h	0	Position actual internal value	INT32	ro	Yes
6064h	0	Position actual value	INT32	ro	Yes
Profile Position Mode					
607Ah	0	Target position	INT32	rw	Yes
6081h	0	Profile velocity	UINT32	rw	Yes
6083h	0	Profile acceleration	UINT32	rw	Yes
6084h	0	Profile deceleration	UINT32	rw	Yes
6086h	0	Motion profile type	INT16	rw	Yes
Profile Velocity Profile					
6069h	0	Velocity sensor actual value	INT32	ro	Yes
606Bh	0	Velocity demand value	INT32	ro	Yes
606Ch	0	Velocity actual value	INT32	ro	Yes
60FFh	0	Target velocity	INT32	rw	Yes
Profile Torque Mode					
6071h	0	Target torque	INT16	rw	Yes
6077h	0	Torque actual value	INT16	ro	Yes
6087h	0	Torque slope	UINT32	rw	Yes
6088h	0	Torque profile type	INT16	rw	Yes
Cyclic Synchronous Position mode/Cyclic Synchronous Velocity Position mode					
60B1h	0	Velocity offset	INT32	rw	Sim
60C2h	0	Interpolation time period	Interpolation time period record	ro	Sim

8.1 DEVICE CONTROL

The following figure represents the state machine for controlling the drive.

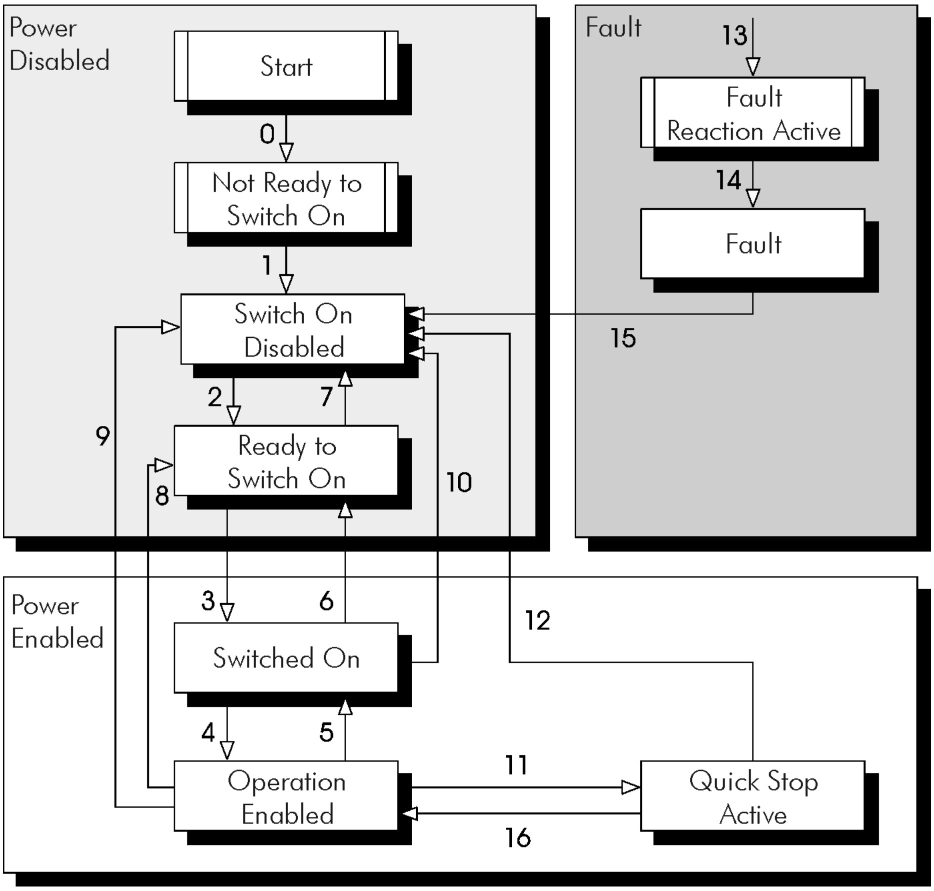


Figure 8.1: *State machine for drives*

States description:

- **Not ready to switch on**: The drive is initializing, it cannot be commanded.
- **Switch on disabled**: Initialization complete, the drive is able to receive commands.
- **Ready to switch on**: Command to allow powering up the drive has been received.
- **Switched on**: command for powering up the drive has been received.
- **Operation enabled**: the drive is enabled, being controlled according to the programmed operation mode. Power is being applied to the motor.
- **Quick stop active**: during the operation, the quick stop command was received. Power is being applied to the motor.
- **Fault reaction active**: a fault has occurred and the drive is performing the action related to the type of fault.
- **Fault**: drive with fault. disabled function, without power being applied to the motor.

Transitions description:

- ✓ **Transition 0**: the drive is switched on and the initialization procedure starts.
- ✓ **Transition 1**: initialization completed (automatic).
- ✓ **Transition 2**: the *Shutdown* command has been received. The state transition is performed, but no action is taken by the drive.
- ✓ **Transition 3**: the *Switch on* command has been received. The state transition is performed, but no action is taken by the drive.
- ✓ **Transition 4**: the *Enable operation* command has been received. The drive is enabled.
- ✓ **Transition 5**: the *Disable operation* command has been received. The drive is disabled.
- ✓ **Transition 6**: the *Shutdown* command has been received. The state transition is performed, but no action is taken by the drive.

- ✓ **Transition 7**: the *Quick stop* and *Disable voltage* commands have been received. The state transition is performed, but no action is taken by the drive.
- ✓ **Transition 8**: the *Shutdown* command has been received. During the drive operation it is disabled, blocking the supply to the motor.
- ✓ **Transition 9**: the *Disable voltage* command has been received. During the drive operation it is disabled, blocking the supply to the motor.
- ✓ **Transition 10**: the *Quick stop* or *Disable voltage* commands have been received. The state transition is performed, but no action is taken by the drive.
- ✓ **Transition 11**: the *Quick stop* command has been received. The drive performs the stopping via ramp function.
- ✓ **Transition 12**: the *Disable voltage* command has been received. The drive is disabled.
- ✓ **Transition 13**: a fault is detected and the drive is disabled.
- ✓ **Transition 14**: after disabling the drive, it goes to the fault state (automatic).
- ✓ **Transition 15**: the *Fault reset* command has been received. The drive performs the fault reset and returns to the disabled and without fault state.
- ✓ **Transition 16**: the *Enable operation* command has been received. The drive performs the start via ramp function.

This state machine is controlled by the object 6040h, and the other states can be monitored by the object 6041h. Both objects are described next.

8.1.1 Object 6040h – Controlword

It controls the drive state. The bits of this word have the following functions:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	Reserved							Halt	Fault reset	Operation mode specific			Enable operation	Quick stop	Enable voltage	Switch on

The bits 0, 1, 2, 3 and 7 allow controlling the drive state machine. The commands for state transitions are given by means of the bit combinations indicated in the table 8.2. The bits marked with “x” are irrelevant for the command execution.

Table 8.2: *Control word commands*

	Control word bits					
Command	**Fault reset**	**Enable operation**	**Quick stop**	**Enable voltage**	**Switch on**	**Transitios**
Shutdown	0	×	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Disable voltage	0	×	×	0	×	7, 9, 10, 12
Quick stop	0	×	0	1	×	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0 → 1	×	×	×	×	15

The bits 4, 5, 6 and 8 have different functions according to the used operation mode.

✓ **NOTE!**
For the commands sent by the control word to be executed by the SCA06 servo drive, it is necessary to programm it for the “CANopen” mode of operation. This programming is done on parameter P0202.

8.1.2 Object 6041h – Statusword

It indicates the drive present state. The bits of this word have the following functions:

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

In this word the bits 0, 1, 2, 3, 5 and 6 indicate the state of the device according to the state machine described in the figure 8.1. The table 8.3 describes the combinations of these bits for the state indications. The bits marked with "x" are irrelevant for the state indication.

Table 8.3: *Drive states indicated through the Statusword*

Value (binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

The other bits indicate a specific condition for the drive.

- **Bit 4 – Voltage enabled**: indicates that the drive power section is powered.
- **Bit 7 – Warning**: indicates that has an active warning. Not used by SCA06.
- **Bit 9 – Remote**: indicates when the drive is in the remote mode and accepts commands via EtherCAT network.
- **Bit 10 – Target reached**: indicates when the drive is operating at the reference value, which depends on the used operation mode. It is also set to 1 when the functions Quick stop or Halt are activated.
- **Bit 11 – Internal limit active**: Not used by SCA06.
- **Bits 12 and 13 – Operation mode specific**: they depend on the drive operation mode.

8.1.3 Object 6060h – Modes of Operation

It allows programming the drive operation mode.

Index	Sub-index	Name	Type	Access	PDO Mapping
6060h	0	Modes of Operation	INT8	rw	Yes

Acceptable values for this object are described in table 8.4. Other values are reserved.

Table 8.4: *Modes of Operation*

Value	Mode of Operation
1	Profile Position Mode
3	Profile Velocity Mode
4	Profile Torque Mode
8	Cyclic sync position mode
9	Cyclic sync velocity mode

8.1.4 Object 6061h – Modes of Operation Display

It indicates the drive operation mode.

Index	Sub-index	Name	Type	Access	PDO Mapping
6061h	0	Modes of Operation Display	INT8	ro	Yes

The value presented at this object follows the same options for object 6060h.

8.1.5 Object 6502h – Supported drives modes

It indicates the modes of operation supported by the drive. Each bit represents a mode of operation, and the value 1 in bit indicates that the mode of operation is supported.

Bit	31 - 16	15-10	9	8	7	6	5	4	3	2	1	0
Mode	Manufacturer specific	Reserved	cst	csv	csp	ip	hm	Reserved	tq	pv	vl	pp

THE SCA06 servo drive features three modes of operation:

- pp: Profile Position mode.
- pv: Profile Velocity mode.
- tq: Torque mode.
- csv: Cyclic sync velocity mode.
- csp: Cyclic sync position mode.

8.2 FACTOR GROUP

This object group allows converting units for objects that represent position values. These values will have their scale and dimension defined according to the programmed notation and dimension values, as described below:

8.2.1 Object 608Fh – Position Encoder Resolution

This object defines the increment of the encoder according to the motor revolution:

Position encoder resolution = encoder increments / motor revolutions

Index	Sub-index	Name	Type	Access	PDO Mapping
608Fh	0	Number Of Entries	UINT8	ro	No
	1	Encoder increments	UINT32	rw	No
	2	Motor revolutions	UINT32	rw	No

Table 8.5: Values for the Encoder Increments Sub-index

Valor	Encoder increments
41h	Degrees
42h	Minutes
43h	Seconds
FFh	Internal unit - 65536 increments by revolution

The sub-index 2 (Motor revolutions) only accepts value equal to 1.

8.2.2 Object 6091h – Gear Ratio

This object indicates the configuration and number of motor shaft revolutions and number of driving shaft revolutions, i.e., it defines the gear ratio. The gear ratio is defined by the following formula:

Gear ratio = motor shaft revolutions / driving shaft revolutions

Index	Sub-index	Name	Type	Access	PDO Mapping
6091h	0	Number Of Entries	UINT8	ro	No
	1	Motor revolutions	UINT32	rw	No
	2	Shaft revolutions	UINT32	rw	No

The only possible value for the sub-index 1 and sub-index 2 is 1.

8.2.3 Object 6092h – Feed constant

This object indicates the distance per one revolution of the motor shaft:

Index	Sub-index	Name	Type	Access	PDO Mapping
6092h	0	Number Of Entries	UINT8	ro	No
	1	Feed	UINT32	rw	No
	2	Shaft revolutions	UINT32	rw	No

Table 8.6: Values for the Feed Sub-index

Valor	Feed
41h	Degrees
42h	Minutes
43h	Seconds
FFh	Internal unit - 65536 increments by revolution

The sub-index 2 (Shaft revolutions) only accepts value equal to 1.

8.3 POSITION CONTROL FUNCTION

This object group is used to describe the operation of the position controller in closed loop.

8.3.1 Object 6063h – Position internal actual value

It represents the actual position of the motor shaft in increments. A complete revolution represents 65536 increments.

Index	Sub-index	Name	Type	Access	PDO Mapping
6063h	0	Position actual value	INT32	ro	Yes

The value of this object always represents the shaft position in a single revolution only. The number of revolutions is not controlled by this object.

8.3.2 Object 6064h – Position Actual Value

It represents the actual position of the motor shaft. The value of this object can be transformed from internal units to user-defined values, according to the settings of the objects 608Fh, 6091h and 6092h, as per table 8.7.

Index	Sub-index	Name	Type	Access	PDO Mapping
6064h	0	Position Actual Value in User Units	INT32	ro	Yes

The higher part of this object represents the number of revolutions and the lower part represents the position of the axis in the current revolution

Table 8.7: *Programming of the Factor Group objects*

Object	608Fh	608Fh	6091h	6091h	6092h	6092h
Unit	sub-index 1	sub-index 2	sub-index 1	sub-index 2	sub-index 1	sub-index 2
Degrees	41h	1	1	1	41h	1
Minutes	42h	1	1	1	42h	1
Seconds	43h	1	1	1	43h	1
Internal unit	FFh	1	1	1	FFh	1

8.4 PROFILE POSITION MODE

This mode of operation allows the control of the SCA06 servo drive by adjusting of the position set-point, which can be executed following two methods:

- single set-point.
- set of set-points.

No matter which method is used, the adjustment of a set-point is performed as follows: first you must write on the Target Position object (607Ah) the desired set-point. Then you must write 1 on the NEW SET POINT bit on the control object (ControlWord – 6041h). The SET-POINT ACKNOWLEDGE bit on the status object (StatusWord – 6040h) will be set indicating that a new set-point was received. If the set-point is accepted, the bit is reset. When the set-point is reached, the TARGET REACHED bit on the status object will be set. Figure 8.2 shows an example of set-point writing.

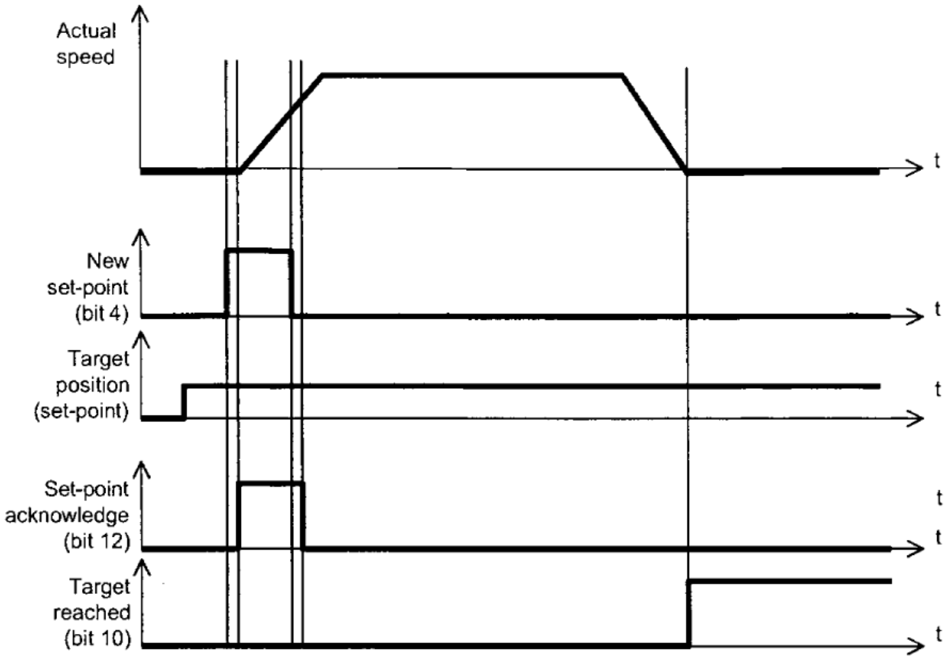


***Figure 8.2:** Adjustment of the position set-point (Source: IEC 61800-7-201)*

Single set-point
The single set-point method is used when you want to execute a new set-point immediately. Figure 8.3 illustrates the method.

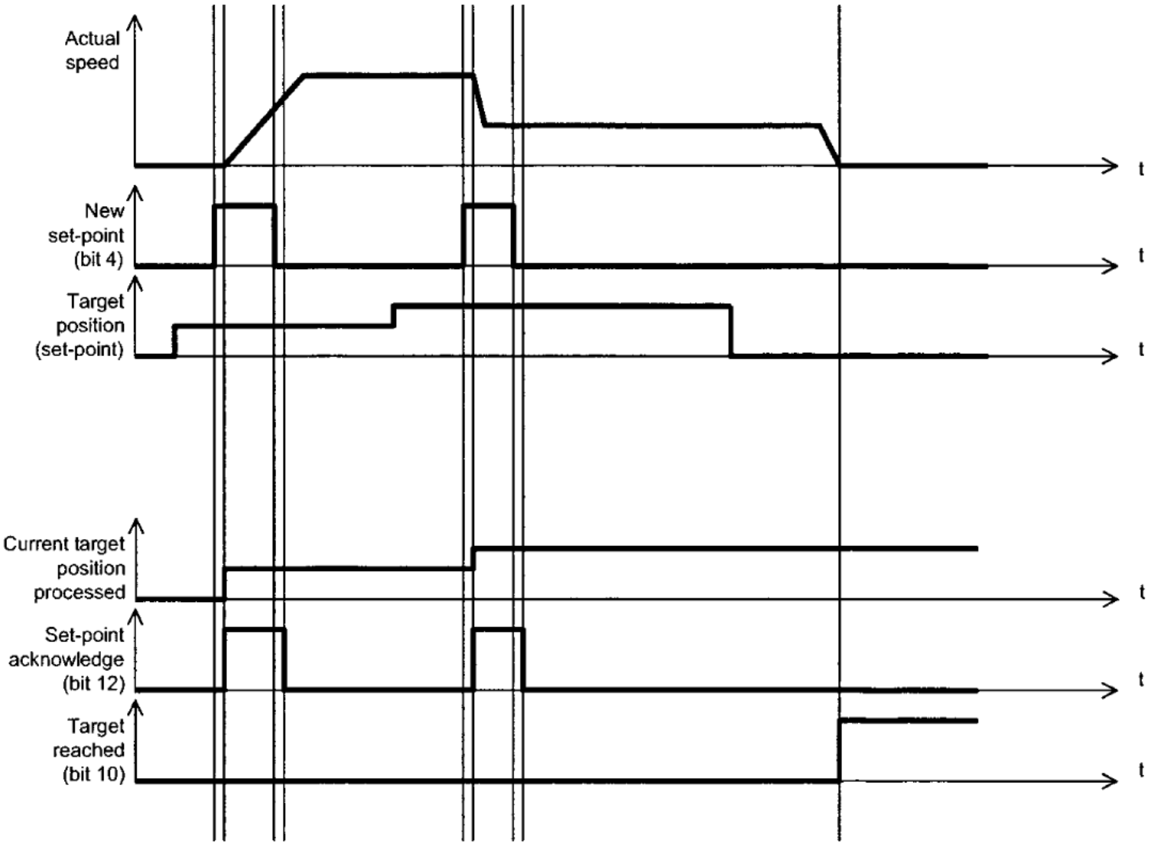

Actual speed
t
New set-point (bit 4)
t
Target position (set-point)
t
Current target position processed
t
Set-point acknowledge (bit 12)
t
Target reached (bit 10)
t

***Figure 8.3:** Single set-point method (Source: IEC 61800-7-201)*

Set of set-point
The set of set-point method is used when you want to execute a new set-point only after the completion of the

previous one. Figure 8.4 illustrates the method.

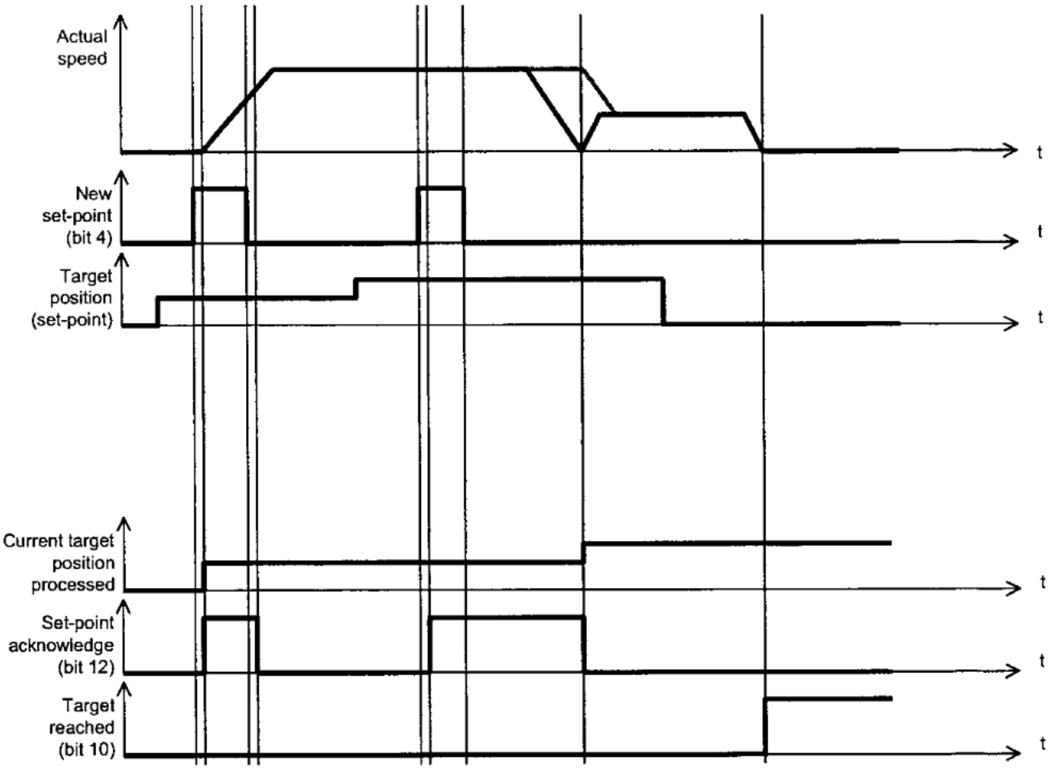

Actual speed
New set-point (bit 4)
Target position (set-point)
Current target position processed
Set-point acknowledge (bit 12)
Target reached (bit 10)
t

Figure 8.4: *Set of set-point method (Source: IEC 61800-7-201)*

The SCA06 servo drive can store two set-points, the one which is in execution and the one that will be executed, as illustrated in figure 8.5.

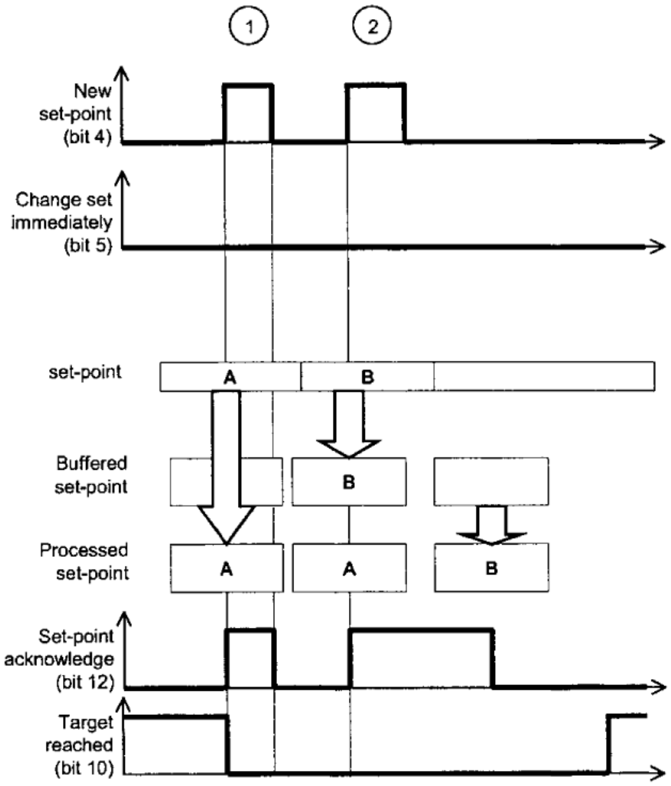


Figure 8.5: *Storage of set-point (Source: IEC 61800-7-201)*

8.4.1 Control and Status Bits

The profile mode position uses some bits of the ControlWord and StatusWord objects to control and monitor its operation.

For the ControlWorld object (6040h) the following bits are used:

- Bit 4 – New set-point.
- Bit 5 – Change set immediately.
- Bit 6 – absolute/relative.
- Bit 8 – Halt.
- Bit 9 – Change on set-point.

Table 8.8 and table 8.9 contain the definition of the control bits.

Table 8.8: *Positioning Mode – definition of the bits 4, 5 and 9*

Bit 9	Bit 5	Bit 4	Definition
0	0	0 → 1	Position must be completed before the next one starts.
X	1	0 → 1	Next position must start immediately.
1	0	0 → 1	Option not implemented on the SCA06.

Table 8.9: *Positioning Mode – definition of bits 6 and 8*

Bit	Value	Definition
6	0	Position reference must be an absolute value.
	1	Position reference must be a relative value.
8	0	Positioning must be executed or continued.
	1	Shaft must be stopped according to object 605Dh.

For the StatusWord object (6041h), the following bits are used:

- Bit 10 – Target reached.
- Bit 12 – Set-point acknowledge.
- Bit 13 – Following error.

Table 8.10 contains the definition of the status bits.

Table 8.10: *Positioning Mode – definition of bits 10, 12 and 13*

Bit	Value	Definition
10	0	Position reference not reached.
	1	Position reference reached.
12	0	Previous position reference already processed, waiting for new position reference.
	1	Previous position reference in process, replacement of position reference will be accepted.
13	0	No Following error.
	1	Following error.

8.4.2 Object 607Ah – Target Position

It allows programming the position reference for the SCA06 servo drive in positioning mode. The 16 most relevant bits inform the number of revolutions and the 16 bits least relevant ones inform the fraction of revolution. The scale used in this object is 65536 for number of revolutions and 65536 increments for one revolution of the shaft. The value of this object must be interpreted as absolute or relative, according to the status of Bit 6 of the ControlWord object (6040h).

Index	Sub-index	Name	Type	Access	PDO Mapping
607Ah	0	Target Position	INT32	rw	Yes

8.4.3 Object 6081h – Profile Velocity

It allows programming the speed normally reached at the end of the acceleration ramp during a movement profile. The value set in this object must be between 0 and 9999 rpm.

Index	Sub-index	Name	Type	Access	PDO Mapping
6081h	0	Profile Velocity	UINT32	rw	Yes

8.4.4 Object 6083h – Profile Acceleration

It allows programming the acceleration ramp until the motor shaft reaches the programmed speed. The scale used is the ms/krpm scale and the values must be between 1 and 32767.

Index	Sub-index	Name	Type	Access	PDO Mapping
6083h	0	Profile Acceleration	UINT32	rw	Yes

8.4.5 Object 6084h – Profile Deceleration

It allows programming the acceleration ramp until the motor shaft reaches the zero speed. The scale used in this object is the same as that of the object 6083h.

Index	Sub-index	Name	Type	Access	PDO Mapping
6084h	0	Profile Deceleration	UINT32	rw	Yes

8.4.6 Object 6086h – Motion Profile Type

It allows programming the profile of the acceleration and deceleration ramp for the drive.

Index	Sub-index	Name	Type	Access	PDO Mapping
6086h	0	Motion Profile Type	INT16	rw	Yes

Table 8.11: *Values for the Motion Profile Type*

Value	Profile
0000h	Linear ramp
FFFFh	No ramp

8.5 PROFILE VELOCITY MODE

This mode of operation allows controlling the drive in a simple way, providing functions, such as:

- Calculation of the reference value.
- Speed capture and monitoring.
- Speed limitation.
- Speed ramps, among other functions.

Those functions are executed based on a set of objects for the configuration of this mode of operation.

8.5.1 Control and Status Bits

Bits 4, 5, 6 and 8 of the control word (object 6040h – Controlword) have the following functions in the speed mode:

Table 8.12: *Profile Velocity Mode – definition of bits 4, 5, 6 and 8*

Bit	Name	Value	Definition
4	Reserved		
5	Reserved		
6	Reserved		
8	Halt	0	Executes movement.
		1	Stops shaft.

For the StatusWord object (6041h), the following bits are used:

- Bit 10 – Target reached;
- Bit 12 – Speed;
- Bit 13 – Max slippage error (not implemented).

Table 8.13: *Profile Velocity Mode – definition of bits 10, 12 and 13*

Bit	Value	Definition
10	0	Halt = 0 – speed reference not reached. Halt = 1 – speed different from 0 (zero).

	1	Halt = 0 – speed reference reached.
		Halt = 1 – speed equal to 0 (zero).

12	0	Speed different from 0 (zero).
	1	Speed equal to 0 (zero).
13	0	Not implemented.
	1	

8.5.2 Object 6069h – Velocity Sensor Actual Value

It allows the reading of the sensor used to measure the motor speed. The SCA06 servo drive uses a resolver as position (the angular speed is obtained by deriving this value in time), so the sensor provides a value proportional to the angular position. The sensor has resolution of 14 bits, and one complete revolution provides 16384 different position values.

Index	Sub-index	Name	Type	Access	PDO Mapping
6069h	0	Velocity Sensor Actual Value	INT32	ro	Yes

8.5.3 Object 606Bh – Velocity Demand Value

It indicates the speed provided by the trajectory generator of the servo drive, used by the speed controller to control the motor. The value provided by this object is given in the internal scale of the SCA06, where 0x7FFF FFFF → 10.000 rpm.

Index	Sub-index	Name	Type	Access	PDO Mapping
606Bh	0	Velocity Demand Value	INT32	ro	Yes

8.5.4 Object 606Ch – Velocity Actual Value

It indicates the motor speed. The value provided by this object is given in the internal scale of the SCA06, where 0x7FFF FFFF → 10.000 rpm.

Index	Sub-index	Name	Type	Access	PDO Mapping
606Ch	0	Velocity Actual Value	INT32	ro	Yes

8.5.5 Object 60FFh – Target Velocity

Allows programming the speed reference for the SCA06 servo drive in speed mode. The value set in this object must observe the internal scale of the SCA06, where 0x7FFF FFFF → 10.000 rpm and 0x8000 0000 → -10.000 rpm.

Index	Sub-index	Name	Type	Access	PDO Mapping
60FFh	0	Target Velocity	INT32	rw	Yes

8.6 PROFILE TORQUE MODE

This mode allows controlling the drive by means of a torque reference received by the EtherCAT network.

Those functions are executed based on a set of objects for the configuration of this mode of operation.

8.6.1 Control and Status Bits

Bits 4, 5, 6 and 8 of the control word (object 6040h – Controlword) have the following functions in the speed mode:

Table 8.14: *Torque Mode – definition of the bits 4, 5, 6 and 8*

Bit	Name	Value	Definition
4	Reserved		
5	Reserved		
6	Reserved		
8	Halt	0	Executes movement.
		1	Stops shaft.

For the StatusWord object (6041h), the following bits are used:

- Bit 10 – Target reached;
- Bit 12 – Reserved;
- Bit 13 – Reserved.

Table 8.15: Torque Mode – definition of bits 10,12 and13

Bit	Value	Definition
10	0	Torque reference not reached.
	1	Torque reference not reached.
12	0	Reserved
	1	
13	0	Reserved
	1	

8.6.2 Object 6071h – Target Torque

It allows programming the torque reference for the SCA06 servo drive in the torque mode. The scale used to write on this object is provided in parts per thousand of the motor rated torque.

Index	Sub-index	Name	Type	Access	PDO Mapping
6071h	0	Target Torque	INT16	rw	Yes

8.6.3 Object 6077h – Torque Actual Value

It indicates the actual motor torque. The value is provided in part per thousand of the rated motor torque.

Index	Sub-index	Name	Type	Access	PDO Mapping
6077h	0	Torque Actual Value	INT16	ro	Yes

8.6.4 Object 6087h – Torque Slope

It allows programming the rate of torque variation in time (torque ramp) for the SCA06 servo drive. The scale used is of parts per thousand of the rated motor torque per second.

Index	Sub-index	Name	Type	Access	PDO Mapping
6087h	0	Torque Slope	INT16	rw	Yes

8.6.5 Object 6088h – Torque Profile Type

It allows programming the torque ramp profile for the drive.

Index	Sub-index	Name	Type	Access	PDO Mapping
6088h	0	Torque Profile Type	INT16	rw	Yes

Table 8.16: Values for the Torque Profile Type

Value	Profile
0000h	Linear ramp of the torque
FFFFh	No ramp

8.7 CYCLIC SINCHRONOUS POSITION MODE

In cyclic synchronous manner, it provides a target position to the drive device, wich performs position control, velocity control and torque control. The overall structure for this mode is shown in Figure 8.6.

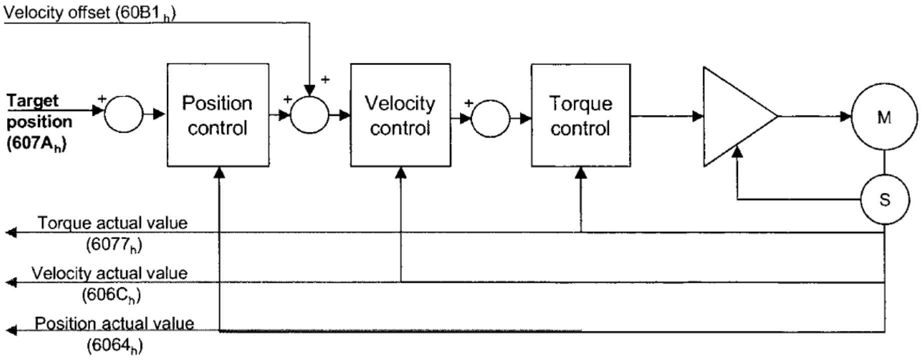


Figure 8.6: *Cyclic synchronous position mode overview*

8.7.1 Control and Status Bits

This mode uses no mode specific bits of the contolword and three bits of the statusword for mode-specific purposes. Table 8.17 defines the values for bit 10, 12 and 13 of the statusword.

Table 8.17: *definition of bit 10, bit 12 and bit 13*

Bit	Value	Definition
10	0	Reserved
	1	Reserved
12	0	Target position is ignored
	1	Target position shall be used as input to position control loop
13	0	No following error
	1	Following error

8.7.2 Object 60B1h – Velocity Offset

This object provide the offset for the velocity value. The offset shall be given in user defined velocity units. This object contains the input value of velocity feed forward.

Index	Sub-index	Name	Type	Access	PDO Mapping
60B1h	0	Velocity Offset	INT32	rw	Yes

8.7.3 Object 60C2h – Interpolation time period

This object shall indicate the configured interpolation cycle time.

Index	Sub-index	Name	Type	Access	PDO Mapping
60C2h	0	Highest sub-index supported	2	ro	No
	1	Interpolation time period value	UINT8	rw	Yes
	2	Interpolation time index	-128 a 63	rw	Yes

8.7.4 Mode Configuration

The objects below will be configured for the drive works in Cyclic Synchronization Position Mode:

- 0x6040 - Controlword
- 0x6060 – Modes of Operation
- 0x60C2 – Interpolation Time Type
- 0x60B1 – Velocity Offset
- 0x6086 – Motion Profile Type
- 0x607A – Target Position;

8.8 CYCLIC SINCHRONOUS VELOCITY MODE

In cyclic synchronous manner, it provides a target velocity to the drive device, wich performs velocity control and torque control. The overall structure for this mode is shown in Figure 8.7.

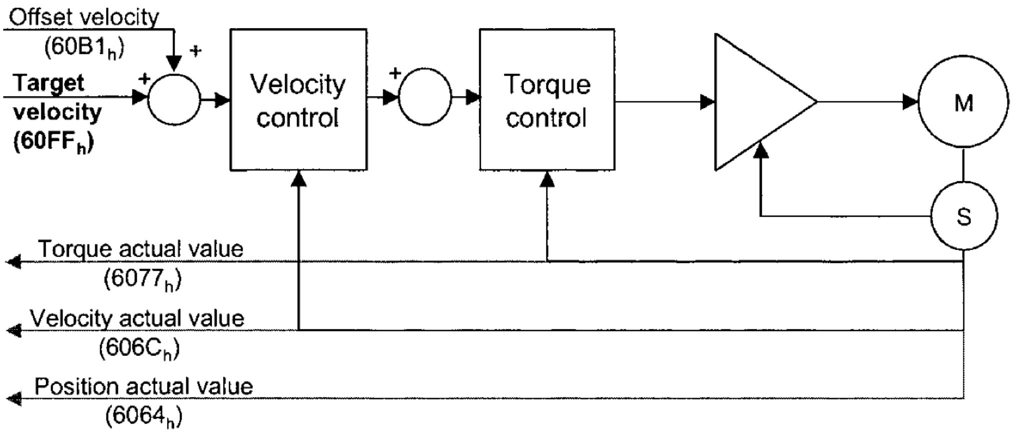


Figure 8.7: Cyclic synchronous velocity mode overview

8.8.1 Control and Status Bits

This mode uses no mode specific bits of the contolword and three bits of the statusword for mode-specific purposes. Table 8.18 defines the values for bit 10, 12 and 13 of the statusword.

Table 8.18: definition of bit 10, bit 12 and bit 13

Bit	Value	Definition
10	0	Reserved
	1	Reserved
12	0	Target velocity ignored
	1	Target velocity shall be used as input to velocity control loop
13	0	Reserved
	1	Reserved

8.8.2 Object 60B1h – Velocity Offset

In cyclic synchronous velocity mode, it contains the command offset of the drive device.

Index	Sub-index	Name	Type	Access	PDO Mapping
60B1h	0	Velocity Offset	INT32	rw	Sim

8.8.3 Object 60C2h – Interpolation time period

According of item 8.7.3.

8.8.4 Mode configuration

The objects below will be configured for the drive works in Cyclic Synchronization Velocity Mode:

- 0x6040 - Controlword
- 0x6060 – Modes of Operation
- 0x60C2 – Interpolation Time Type
- 0x60B1 – Velocity Offset
- 0x6086 – Motion Profile Type
- 0x60FF – Target Velocity;

9 STARTUP GUIDE

The following are the main steps for commissioning of SCA06 servo drive on the EtherCAT network. The steps are an example of usage. Refer to specific chapters for details on the steps.

9.1 INSTALLING THE ETHERCAT MODULE

1. Install the EtherCAT communication module at slot 2, as indicated in the installation guide that came with the module.
2. Connect the Ethernet cable to the module, considering the necessary care in network installation, as described in section 4:
   - Use shielded cable.
   - Perform a properly grounding connection.
   - Avoid the passage of communication cables next to power cables.

9.2 CONFIGURING THE DRIVE

1. Follow the recommendations described in the user's guide to program the device parameter related to motor parameters, desired functions for the I/O signs, etc.
2. Program the desired mode of operation for application in parameter P0202 (P0202 = 5 for control via EtherCAT).
3. Set the desired device action in the event of communication failure, through P0662 (P0662 = 1 to cause equipment failure in the event of loss of communication with the master).
4. Other settings, such as selection of data for communication via EtherCAT, are made by the configuration tool of the network master.

9.3 CONFIGURING THE MASTER

The way you do the setup of the network depends largely on the master and configuration tool. It is important to know the tools used to perform this activity. In general, the following steps are required to perform the network configuration.

1. Load the XML configuration file provided on the CD shipped with the product [1] to the equipment list in the network configuration tool.
2. Select SCA06 servo drive on the list of available products in the network configuration tool. This can be done manually or automatically, if allowed by the tool.
3. Program the transmit and receive PDOs, as well as the data transmitted and received by these PDOs:
   - There are 4 TxPDOs and 4 RxPDOs, but only one can be enabled at a time.
   - TxPDOs are associated with the Sync Manager 2 (outputs), while RxPDOs are associated with the Sync Manager 3 (Inputs). The selection of which receive and transmit PDOs are active is done through objects 1C12h and 1C13 respectively.
   - Each PDO has a default mapping containing a set of predetermined objects for communication. This default mapping is usually sufficient to meet most applications, but you can change the mapping if desired.
4. You must also set the data update cycle. The minimum value supported by the device is 200 us.
5. The equipment allows the master to use two interrupt modes for data update:

[1] It is important to note if the XML configuration file is compatible with SCA06 servo drive firmware version.

- Without synchronized interrupt (SM-Synchron): for each EtherCAT telegram received, an interrupt is generated to update the data process.
- Synchronized network interrupt (DC-Synchron): the equipment also allows to use a distributed clock mechanism, where data is updated at regular intervals by the equipment simultaneously, and these clocks are synchronized by the EtherCAT network.

9.4 COMMUNICATION STATUS

Once the network is installed and master is programmed, you can use the device parameters to identify some states related to communication. The network master must also provide information about communication with slave.

- Parameter P0851 indicates communication status for EtherCAT interface.
- Parameter P0852 provides information about communication link for each Ethernet port.
- Parameter P0853 indicates the slave state, with respect to the EtherCAT state machine.
- Parameters P0855 to P0858 indicate PDO usage and number of mapped data.
- Parameter P0859 indicates the update time for data exchange with the master.

9.5 OPERATION USING PROCESS DATA

Once communication is established, the data mapped in the PDOs are automatically updated between master and slave. Among the main objects that can be used to control the drive, we can mention:

- 6040h: ControlWord
- 6041h: StatusWord
- 6060h: Mode of operation
- 6063h: Position actual value
- 607Ah: Target position
- 60FFh: Target velocity
- 6071h: Target Torque

It is important to know these objects and understand the operation of each mode of operation to program the master as desired for the application.

10 FAULTS AND ALARMS

F0045/A0145 - ETHERCAT INTERFACE ACCESS ERROR

Description:

It indicates data exchange failure between SCA06 servo drive and the EtherCAT accessory.

Actuation:

During initialization and operation, regular data check is performed between device and EtherCAT accessory to assure this connection is correct.

If a problem is identified, it will sign through the display an alarm message A0145 - or failure F0045, depending on P0662 programming. You need to power off and on the device again for a new attempt to access the EtherCAT module.

Possible Causes/Correction:

- Check if the accessory is properly connected.
- Check the firmware version for drive and accessory, to be sure they are compatible.
- Check if the master scan cycle is higher than the minimum allowed by the device.
- Hardware errors caused, for example, from handling or improper installation of accessories may cause this error. If possible, perform tests replacing the accessory communication.

F0046/A0146 - ETHERCAT OFFLINE

Description:

It indicates communication failure between the slave and EtherCAT master.

Actuation:

Acts when, for some reason, there is a break in communication between the slave and the EtherCAT master, after the cyclic communication was initiated.

If a problem is identified, it will sign through the display an alarm message A0146 - or failure F0046, depending on P0662 programming. For alarms, the indication will disappear when the cyclic communication is restored..

Possible Causes/Correction:

- Check whether the network master is properly configured and operating normally.
- Check for short circuit or bad contact in communication cables.
- Check the installation of the entire network - cable routing, grounding.

WEG Drives & Controls - Automação LTDA.
Jaraguá do Sul – SC – Brazil
Phone 55 (47) 3276-4000 – Fax 55 (47) 3276-4020
São Paulo – SP – Brazil
Phone 55 (11) 5053-2300 – Fax 55 (11) 5052-4212
automacao@weg.net
www.weg.net