POWER ELECTRONICS

SD750

ACCESSORIES MANUAL

ETHERNET/IP BOARD

LOW VOLTAGE VARIABLE SPEED DRIVE

SD750

LOW VOLTAGE VARIABLE SPEED DRIVE

Accessories Manual

Ethernet/IP board

Edition: February 2022

SD75MA01CI Rev. C

ABOUT THIS MANUAL

PURPOSE

This manual contains important instructions for the installation, configuration and use of the Ethernet/IP optional board for Power Electronics' SD750 variable speed drives. From now on, this manual refers to XMV660 with the term "board" or "equipment".

Power Electronics reserves the right to modify product features.

TARGET AUDIENCE

This manual is intended for qualified customers who will install, operate and maintain Power Electronics SD750 variable speed drives.

Only qualified technical personnel validated by Power Electronics may install and start up the chargers.

REFERENCE MANUALS

The following reference documents are available for SD750 variable speed drives:

- Hardware and Installation Manual.
- Programming and Software Manual.
- Maintenance Manual.
- Pumps Application Manual.

POWER ELECTRONICS CONTACT INFORMATION

Power Electronics España, S.L.
Polígono Industrial Carrases
Ronda del Camp d'Aviació nº 4
46160, Llíria (Valencia)
ESPAÑA
Teléfono: (+34) 96 136 65 57
Página web: www.power-electronics.com

Power Electronics USA Inc.
1510 N. Hobson Street, Gilbert,
Phoenix
AZ 85233
ESTADOS UNIDOS DE AMÉRICA
Ventas US: 602-354-4890 / (480) 519-5977

REVISIONS CONTROL		
DATE (DD/MM/YYYY)	**REVISION**	**DESCRIPTION**
12/10/2019	A	First Edition.
06/03/2020	B	Second Edition.
22/02/2022	C	Commissioning. Faults and warnings mapping.

EN

TABLE OF CONTENTS

SAFETY SYMBOLS

Always follow safety instructions to prevent accidents and potential hazards from occurring.

In this manual, safety messages are classified as follows:

 **WARNING**	Identifies potentially hazardous situations where dangerous voltage may be present, which if not avoided, could result in minor personal injury, serious injury or death. Be extremely careful and follow the instructions to avoid the risk of electrical shocks.
 **CAUTION**	Identifies potentially hazardous situations, which if not avoided, could result in product damage, or minor or moderate personal injury. Read the message and follow the instructions carefully.
 **NOTICE**	Identifies important measures to take in order to prevent damage equipment and warranty lost, as well as encouraging good use and environmental practices.

EN

Other symbols used in this manual for safety messages are the following:

	Hot surface. Be careful and follow the instructions to avoid burns and personal injuries.
	Risk of fire. Be careful and follow the instructions to prevent causing an unintentional fire.
	Caution, risk of electric shock. Energy storage timed discharge. Wait for the indicated time to avoid electrical hazards.
	Caution, risk of hearing damage. Wear hearing protection.

SAFETY INSTRUCTIONS

IMPORTANT!

Read carefully this manual to maximize the performance of the product and to ensure its safe installation and use.

In order to appropriately use the drive, please, follow all instructions described in the *Hardware and Installation Manual* which refer to transportation, installation, electrical connection and commissioning of the equipment.

For maintenance operations, follow the instructions from the *Maintenance Manual.*

Power Electronics accepts no responsibility for any damages resulting from incorrect use of equipment.

CAUTION

Read carefully the *Hardware and Installation Manual,* the *Maintenance Manual* and all documentation related to the drive to guarantee its safe use and avoid the risk of personal injuries and damages to the equipment.

Ensure compliance with local and national regulations of the installation site.

NOTICE

CAUTION IN CONNECTIONS

Use conductive paste between plates in every electrical connection. Otherwise, resistance will increase and an overheat in the contact zone of the conductors may occur.

INTRODUCTION

1

SD750 drives are compatible with several optional boards:

- Communication boards (Ethernet/IP, Profinet, CANopen…).
- Encoder board.
- Digital and analog I/O expansion boards.
- Optical fiber board

…among others. Up to three optional boards can be connected, maximum two of the same type.

EN

This manual focuses on the optional communication board Ethernet/IP. This board allows connecting SD750 drives to an Ethernet network (LAN – Local Area Net). It supports the standard TCP/IP communication protocol and the industrial application layer protocol Ethernet/IP for industrial automation applications.

Thanks to this board, the drive can be controlled and monitored through the network, either by the user or through a PLC sequence program or any master device (client).

Ethernet board specifications

- Device type: Network adapter.
- Shape factor: Inserted board.
- Wiring type: Ethernet 10Base-T, Ethernet 100Base-TX.
- Data exchange protocol: Modbus TCP/IP, Ethernet/IP.
- Auto-addressing Protocol DHCP supported.
- Data transmission speed: 10Mbps, 100Mbps, auto-negotiation 10 / 100.
- Standards: IEEE 802.3, IEEE 802.3u (only for 100Base-TX).

LED indicators

The Ethernet/IP board includes 5 LEDs (status, run, ready, Eth1 and Eth2) that provide information about the power supply of the board, network detection and communication status.

The LED indicators on the Ethernet/IP expansion board provide information about the board and communication status. Please notice that some LEDs are bicolor and will change their color and frequency according to the situation. Refer to the following table for the description of all possible colors and frequency of each LED indicator, as well as what they represent.

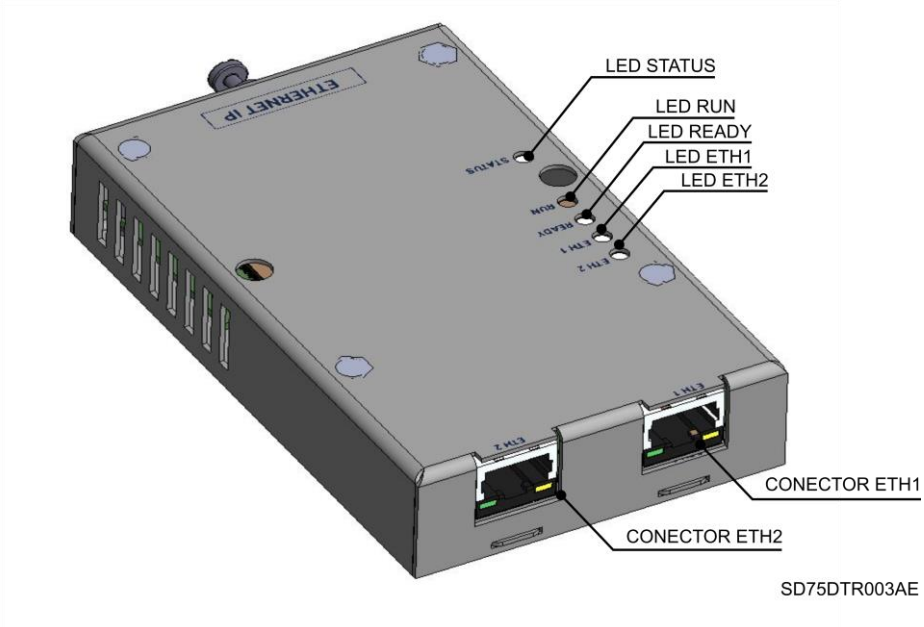
LED	COLOR / FREQUENCY	DESCRIPTION
STATUS	Red, steady	Hardware failure or inability to install the protocol. The board probably has to be repaired.
	Red, slow flashing	The board cannot stablish communication with the network controller or initialize the protocol.
	Red, fast flashing	The board cannot stablish communication with the SD750 central board.
	Green, slow flashing	The system is operating correctly.
	Green, fast flashing	Test mode. User has set the board to test mode.
RUN	Green	User application is running without errors.
READY	Green	Board's operating system is working correctly.
ETH1	Green, steady	**Device operational:** The devices is operating correctly
	Green, flashing	**Standby:** The device has not been configured.
	Red / green, flashing	**Self-test:** The device is performing its power up testing.
	Red, flashing	**Minor fault:** the device has detected a recoverable minor fault. E.g., an incorrect or inconsistent configuration can be considered as a minor fault.
	Red, steady	**Major fault:** The device has detected a non-recoverable major fault.
	Off	**No power:** The power supply to the device is missing.

LED	COLOR / FREQUENCY	DESCRIPTION
ETH2	Green, steady	**Connected:** The device has at least one established connection (even to the Message Router).
	Green, flashing	**No connections:** The device has no established connections but has obtained an IP address.
	Red / green, flashing	**Self-test:** The device is performing its power up testing.
	Red, flashing	**Connection timeout:** One or more of the connections in which this device is the target has timed out. This status will be finished only if all timed out connections are reestablished or if the device is reset.
	Red, steady	**Duplicate IP:** the device has detected that its IP address is already in use.
	Off	**Not powered, no IP address:** The device does not have an IP address (or is powered off).

Note: "Device" refers to the communication slave.

EN

Finally, the two RJ45 connectors ETH1 and ETH2 allow ring communication in the Ethernet/IP network (the board can be connected to two different networks).



Requirements

To establish communication with SD750 drives via Ethernet/IP, the user should have one client which supports Ethernet/IP protocol that supports at the same time:

- Explicit connection messaging: data without temporal relevance (configuration, diagnosis, data collection).
- Connected I/O messaging: I/O real-time data, functional safety data, motion control data.
- Unconnected messaging: accessing to data without establishing a connection.

Refer to section 2 for more information about Ethernet/IP protocol.

ETHERNET/IP PROTOCOL

2

Ethernet/IP introduction

Ethernet/IP is an application layer protocol for industrial automation applications. It uses the standard protocols TCP/IP and Ethernet to configure, access and control industrial automation devices.

Ethernet/IP provides a total integrated system from the industrial floor to the company network. It is a network protocol highly appropriate for the industrial environment because it allows establishing real-time communication between workstations and I/O devices.

Ethernet/IP adapts the Common Industrial Protocol (CIP) to standard Ethernet. CIP is based on abstract object modeling. Every device in a CIP network is modeled as a collection of objects. This offers several advantages for users and automation manufacturers, such as low-cost product development, usability, simple integration of devices and networks, and interoperability among suppliers.

According to the CIP specification, an object provides an abstract representation of a particular component within a product. Therefore, anything not described in object form is not visible through CIP.

CIP objects can have the following structured elements:

- Classes
- Instances
- Attributes

Furthermore, objects may contain **services** offering a well-defined functionality. CIP (and thus Ethernet/IP) separates between two standard types of messaging:

- **Explicit Messaging** (Class 3 and UCMM). Explicit messages are used within CIP for point-to-point and client/server connections. They contain addressing and service information causing execution of a specific service on a specific part of the network node. Services are only available in this type of messaging.
- **Implicit Messaging** (Class 1). Implicit messages do not contain any transmission protocol in their IO data, for instance there is not any address and/or service information. A dynamically generated unique connection ID allows reliable identification. The data format has already been specified in the EDS (Electronic data sheet) file previously. Thus, the efficiency of data transmission is improved as the meaning of the data is already known.

Explicit messaging (class 3 and UCMM)

Explicit messaging is used for point to point messaging that typically takes place only once (or at least not very frequently). Explicit messaging is typically used for non-real time data. The messaging uses the request/response mechanism based on the client-server model.

In explicit messaging, objects have associated functions called **services**. Services are identified by their service codes defining the kind of action to take place when an object is entirely or partly addressed through explicit messages according to the addressing scheme. The implemented services are:

Code	Function	Description
01	Get_Attributtes_All	To be used at instance level; returns a list with the value of all the attributes
0E	Get_Attribute_Single	To be used at attribute level; returning the current value of the attribute
10	Set_Attribute_Single	To be used at attribute level; setting a value to the attribute

EN

Implicit messaging (class 1)

Implicit messaging is used for cyclic communication, i.e., for periodically repeated transmission of data with the same structure. Implicit messages are based on the producer-consumer model, which supports multicast and unicast (Point-2-Point) messaging. When opening a CIP I/O connection a scanner usually connects to a pair of **assembly instances**, also called connection points. Each assembly instance comes with a specific data structure. For example, the data of an assembly instance can combine attributes of other objects.

CIP data types

Data type	Description	Number of bytes	Code
BOOL	Boolean	1-bit encoded into 1-byte	0xC1
BYTE	Bit String- 8 bits	1 byte	0xD1
USINT	Unsigned short integer	1 byte	0xC6
SINT	Short integer	1 byte	0xC2
WORD	Bit string - 16 bits	2 bytes	0xD2
UINT	Unsigned integer	2 bytes	0xC7
INT	Integer	2 bytes	0xC3
DWORD	Bit string - 32 bits	4 bytes	0xD3
UDINT	Unsigned Double Integer	4 bytes	0xC8
DINT	Double Integer	4 bytes	0xC4
SHORT_STRING	Character string (1 byte per character, 1-byte length indicator)	1 + n (first byte indicates length)	0xDA

EDS (Electronic Data Sheet) file

An EDS is a simple ASCII text file that can be generated on any ASCII editor. The CIP specification lays down a set of rules for the overall design and syntax of an EDS which makes configuration of devices much easier. The main purpose of the EDS is to give information on several aspects of the device's capabilities, the most important ones being the I/O Connections it supports and what parameters for visualization or configuration exist within the device.

The EDS file is structured into sections, each of which starts with a section name in square brackets []:

- **[Device]:** Is equivalent to the Identity Object information and is used to match an EDS to a section.
- **[Device Classification]:** Describes what network the device can be connected to.
- **[Params]:** Identifies all configuration parameters in the device.
- **[Assembly]:** Describes the structure of data items.
- **[Connection Manager]:** Describes connections supported by the device.
- **[Capacity]:** Specifies the communication capacity of Ethernet/IP devices.

CIP classes

The table below lists all default CIP object classes available within the CoIn Ethernet/IP stack that are implemented in the SD750 drive. They will be described later on this section:

Name	Class ID
Object Identity	0x01
Object Message Router	0x02
Object Assembly	0x04
Object Connection Manager	0x06
Object Parameter	0x0F
Object Parameter Group	0x10
Object Motor Data	0x28
Object Control Supervisor	0x29
Object AC Drive	0x2A
Object PE Status (Power Electronics)	0x65
Object TCP/IP	0xF5
Object Ethernet Link	0xF6

EN

For further details about objects and attributes, refer to the CIP protocol specification.

Object Identity

The Identity Object provides identification and general information about the device. The **first and only instance** identifies the whole device. It is used for electronic keying and by applications wishing to determine what devices are on the network.

Instance attributes:

Attribute ID	Name	Data type	Default	Services
1	Vendor ID	UINT	1104: Power Electronics	Get
2	Device Type	UINT	2: AC drive	Get
3	Product Code	UINT	750	Get
4	Revision	Struct	1,1	Get
5	Status	WORD	-	
6	Serial number	UDINT	-	Get
7	Product name	SHORT_STRING	PESD750Drive	Get

Object Message Router

The Message Router Object provides a messaging connection point through which a client may address a service to any object class or instance residing in the physical device. Since the message router does not have any class or instance attributes, there are no services supported.

Object Assembly

This object binds attributes of multiple objects, which allows data to or from each object to be sent or received over a single connection. Assembly objects can be used to bind input data or output data. The terms "input" and "output" are defined from the network's point of view (PLC). An input will produce data on the network and an output will consume data from the network.

Following the AC/DC Drive profile, the following instances are implemented:

Instance Number		Type	Size (bytes)	Name
Decimal	Hex.			
20	14	Output	4	Basic Speed Control
21	15	Output	4	Extended Speed Control
22		Output		Speed and torque control Output
23		Output		Ext Speed and torque control Output
24		Output		Process control Output
25		Output		Ext Process control Output
70	46	Input	4	Basic Speed Control Status
71	47	Input	4	Extended Speed Control Status
72		Input		Speed and torque control Input
73		Input		Ext Speed and torque control Input
74		Input		Process control Input
75		Input		Ext Process control Input
100	64	Output	4	Custom modbus 4 bytes Output
101	65	Output	8	Custom modbus 8 bytes Output
102		Output	12	Custom modbus 12 bytes Output
103		Output	16	Custom modbus 16 bytes Output
104		Output	32	Custom modbus 32 bytes Output
150	96	Input	4	Custom modbus 4 bytes Input
151	97	Input	8	Custom modbus 8 bytes Input
152	98	Input	12	Custom modbus 12 bytes Input
153		Input	16	Custom modbus 16 bytes Input
154		Input	32	Custom modbus 32 bytes Input

The format of the attributes is given below:

Instance	Bit / Byte	7	6	5	4	3	2	1	0
20	0						Fault Reset		Run Fwd
	1								
	2	Speed Reference (Low Byte) – RPM							
	3	Speed Reference (High Byte) – RPM							
21	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1								
	2	Speed Reference (Low Byte) – RPM							
	3	Speed Reference (High Byte) – RPM							
22	0						Fault Reset		Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							

Instance	Bit / Byte	7	6	5	4	3	2	1	0
	4	Actual Torque (Low Byte)							
	5	Actual Torque (High Byte)							
23	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
	4	Actual Torque (Low Byte)							
	5	Actual Torque (High Byte)							
24	0						Fault Reset		Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
	4	Actual Torque (Low Byte)							
	5	Actual Torque (High Byte)							
25	0	At Ref	Ref from Net	Ctrl from Net			Running 1 (Fwd)	Warning	Faulted
	1	Mode							
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
	4	Process Reference (Low Byte)							
	5	Process Reference (High Byte)							
70	0						Running 1 (Fwd)		Faulted
	1								
	2	Actual Speed (Low Byte) – RPM							
	3	Actual Speed (High Byte) – RPM							
71	0	At Ref	Ref from Net	Ctrl from Net	Ready	Running 2 (Rev)	Running 1 (Fwd)	Warning	Faulted
	1	Drive Status							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
72	0						Running 1 (Fwd)		Faulted
	1								
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Actual Torque (Low Byte)							
	5	Actual Torque (High Byte)							
73	0	At Ref	Ref from Net	Ctrl from Net	Ready	Running 2 (Rev)	Running 1 (Fwd)	Warning	Faulted
	1	Drive state							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Actual Torque (Low Byte)							
	5	Actual Torque (High Byte)							
74	0						Running 1 (Fwd)		Faulted
	1								
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Actual Torque (Low Byte)							
	5	Actual Torque (High Byte)							

EN

Instance	Bit / Byte	7	6	5	4	3	2	1	0
75	0	At Ref	Ref from Net	Ctrl from Net	Ready	Running 2 (Rev)	Running 1 (Fwd)	Warning	Faulted
	1	Drive state							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Actual Torque (Low Byte)							
	5	Actual Torque (High Byte)							
100	0	Custom Modbus Value 1 (Low Byte)							
	1	Custom Modbus Value 1 (High Byte)							
	2	Custom Modbus Value 2 (Low Byte)							
	3	Custom Modbus Value 2 (High Byte)							
101	0	Custom Modbus Value 1 (Low Byte)							
	1	Custom Modbus Value 1 (High Byte)							
	2	Custom Modbus Value 2 (Low Byte)							
	3	Custom Modbus Value 2 (High Byte)							
	4	Custom Modbus Value 3 (Low Byte)							
	5	Custom Modbus Value 3 (High Byte)							
	6	Custom Modbus Value 4 (Low Byte)							
	7	Custom Modbus Value 4 (High Byte)							
102	0	Custom Modbus Value 1 (Low Byte)							
	1	Custom Modbus Value 1 (High Byte)							
	2	Custom Modbus Value 2 (Low Byte)							
	3	Custom Modbus Value 2 (High Byte)							
	4	Custom Modbus Value 3 (Low Byte)							
	5	Custom Modbus Value 3 (High Byte)							
	6	Custom Modbus Value 4 (Low Byte)							
	7	Custom Modbus Value 4 (High Byte)							
	8	Custom Modbus Value 5 (Low Byte)							
	9	Custom Modbus Value 5 (High Byte)							
	10	Custom Modbus Value 6 (Low Byte)							
	11	Custom Modbus Value 6 (High Byte)							
103	0	Custom Modbus Value 1 (Low Byte)							
	1	Custom Modbus Value 1 (High Byte)							
	2	Custom Modbus Value 2 (Low Byte)							
	3	Custom Modbus Value 2 (High Byte)							
	4	Custom Modbus Value 3 (Low Byte)							
	5	Custom Modbus Value 3 (High Byte)							
	6	Custom Modbus Value 4 (Low Byte)							
	7	Custom Modbus Value 4 (High Byte)							
	8	Custom Modbus Value 5 (Low Byte)							
	9	Custom Modbus Value 5 (High Byte)							
	10	Custom Modbus Value 6 (Low Byte)							
	11	Custom Modbus Value 6 (High Byte)							
	12	Custom Modbus Value 7 (Low Byte)							
	13	Custom Modbus Value 7 (High Byte)							
	14	Custom Modbus Value 8 (Low Byte)							
	15	Custom Modbus Value 8 (High Byte)							
104	0	Custom Modbus Value 1 (Low Byte)							
	1	Custom Modbus Value 1 (High Byte)							
	2	Custom Modbus Value 2 (Low Byte)							

Instance	Bit / Byte	7	6	5	4	3	2	1	0
	3	Custom Modbus Value 2 (High Byte)							
	4	Custom Modbus Value 3 (Low Byte)							
	5	Custom Modbus Value 3 (High Byte)							
	6	Custom Modbus Value 4 (Low Byte)							
	7	Custom Modbus Value 4 (High Byte)							
	8	Custom Modbus Value 5 (Low Byte)							
	9	Custom Modbus Value 5 (High Byte)							
	10	Custom Modbus Value 6 (Low Byte)							
	11	Custom Modbus Value 6 (High Byte)							
	12	Custom Modbus Value 7 (Low Byte)							
	13	Custom Modbus Value 7 (High Byte)							
	14	Custom Modbus Value 8 (Low Byte)							
	15	Custom Modbus Value 8 (High Byte)							
	16	Custom Modbus Value 9 (Low Byte)							
	17	Custom Modbus Value 9 (High Byte)							
	18	Custom Modbus Value 10 (Low Byte)							
	19	Custom Modbus Value 10 (High Byte)							
	20	Custom Modbus Value 11 (Low Byte)							
	21	Custom Modbus Value 11 (High Byte)							
	22	Custom Modbus Value 12 (Low Byte)							
	23	Custom Modbus Value 12 (High Byte)							
	24	Custom Modbus Value 13 (Low Byte)							
	25	Custom Modbus Value 13 (High Byte)							
	26	Custom Modbus Value 14 (Low Byte)							
	27	Custom Modbus Value 14 (High Byte)							
	28	Custom Modbus Value 15 (Low Byte)							
	29	Custom Modbus Value 15 (High Byte)							
	30	Custom Modbus Value 16 (Low Byte)							
	31	Custom Modbus Value 16 (High Byte)							
150	0	Custom Modbus Value 17 (Low Byte)							
	1	Custom Modbus Value 17 (High Byte)							
	2	Custom Modbus Value 18 (Low Byte)							
	3	Custom Modbus Value 18 (High Byte)							
	4	Custom Modbus Value 19 (Low Byte)							
151	0	Custom Modbus Value 17 (Low Byte)							
	1	Custom Modbus Value 17 (High Byte)							
	2	Custom Modbus Value 18 (Low Byte)							
	3	Custom Modbus Value 18 (High Byte)							
	4	Custom Modbus Value 19 (Low Byte)							
	5	Custom Modbus Value 19 (High Byte)							
	6	Custom Modbus Value 20 (Low Byte)							
	7	Custom Modbus Value 20 (High Byte)							
152	0	Custom Modbus Value 17 (Low Byte)							
	1	Custom Modbus Value 17 (High Byte)							
	2	Custom Modbus Value 18 (Low Byte)							
	3	Custom Modbus Value 18 (High Byte)							
	4	Custom Modbus Value 19 (Low Byte)							
	5	Custom Modbus Value 19 (High Byte)							
	6	Custom Modbus Value 20 (Low Byte)							

EN

Instance	Bit / Byte	7	6	5	4	3	2	1	0
	7	Custom Modbus Value 20 (High Byte)							
	8	Custom Modbus Value 21 (Low Byte)							
	9	Custom Modbus Value 21 (High Byte)							
	10	Custom Modbus Value 22 (Low Byte)							
	11	Custom Modbus Value 22 (High Byte)							
153	0	Custom Modbus Value 17 (Low Byte)							
	1	Custom Modbus Value 17 (High Byte)							
	2	Custom Modbus Value 18 (Low Byte)							
	3	Custom Modbus Value 18 (High Byte)							
	4	Custom Modbus Value 19 (Low Byte)							
	5	Custom Modbus Value 19 (High Byte)							
	6	Custom Modbus Value 20 (Low Byte)							
	7	Custom Modbus Value 20 (High Byte)							
	8	Custom Modbus Value 21 (Low Byte)							
	9	Custom Modbus Value 21 (High Byte)							
	10	Custom Modbus Value 22 (Low Byte)							
	11	Custom Modbus Value 22 (High Byte)							
	12	Custom Modbus Value 23 (Low Byte)							
	13	Custom Modbus Value 23 (High Byte)							
	14	Custom Modbus Value 24 (Low Byte)							
	15	Custom Modbus Value 24 (High Byte)							
154	0	Custom Modbus Value 17 (Low Byte)							
	1	Custom Modbus Value 17 (High Byte)							
	2	Custom Modbus Value 18 (Low Byte)							
	3	Custom Modbus Value 18 (High Byte)							
	4	Custom Modbus Value 19 (Low Byte)							
	5	Custom Modbus Value 19 (High Byte)							
	6	Custom Modbus Value 20 (Low Byte)							
	7	Custom Modbus Value 20 (High Byte)							
	8	Custom Modbus Value 21 (Low Byte)							
	9	Custom Modbus Value 21 (High Byte)							
	10	Custom Modbus Value 22 (Low Byte)							
	11	Custom Modbus Value 22 (High Byte)							
	12	Custom Modbus Value 23 (Low Byte)							
	13	Custom Modbus Value 23 (High Byte)							
	14	Custom Modbus Value 24 (Low Byte)							
	15	Custom Modbus Value 24 (High Byte)							
	16	Custom Modbus Value 25 (Low Byte)							
	17	Custom Modbus Value 25 (High Byte)							
	18	Custom Modbus Value 26 (Low Byte)							
	19	Custom Modbus Value 26 (High Byte)							
	20	Custom Modbus Value 27 (Low Byte)							
	21	Custom Modbus Value 27 (High Byte)							
	22	Custom Modbus Value 28 (Low Byte)							
	23	Custom Modbus Value 28 (High Byte)							
	24	Custom Modbus Value 29 (Low Byte)							
	25	Custom Modbus Value 29 (High Byte)							

Instance	Bit / Byte	7	6	5	4	3	2	1	0
	26	Custom Modbus Value 30 (Low Byte)							
	27	Custom Modbus Value 30 (High Byte)							
	28	Custom Modbus Value 31 (Low Byte)							
	29	Custom Modbus Value 31 (High Byte)							
	30	Custom Modbus Value 32 (Low Byte)							
	31	Custom Modbus Value 32 (High Byte)							

Each Assembly instance has the following attributes:

Attribute ID	Name	Data Type	Default value
1	Data	Array of Bytes	-
2	Size	UINT	-

Object Connection Manager

The object Connection Manager allocates and manages the **internal resources** associated with both I/ O and Explicit Messaging Connections.

Object Parameter

Use of the object Parameter provides a known, public interface to a device's configuration data. There must be one instance of this object class for each of the device's configurable parameters. The instances must start at instance one and increment by one with no gaps in the instances.

The implemented instances of object Parameter can be found at the EDS file.

Instance attributes:

Attribute ID	Name	Data type	Default	Services
1	Value	Variable	-	Get/Set (*)
2	Link Path Size	USINT	7	Get
3	Link Path (**)	EPATH	-	Get
5	Data type	USINT	-	Get
6	Data Size	USINT	-	Get

(*) Parameter value can be set if the instance belongs to a R/W variable.

(**) Link path: CIP path to the object from where this parameter's value is retrieved. The packed PATH of the parameter object has the following structure:

Class Segment	Class ID	Instance Segment	Instance ID 1	Instance ID 2	Attribute Segment	Attribute ID
0x20	0x0F	0x25	InstanceID LSB	InstanceID MSB	0x30	0x01

In the case of the Ethernet/IP board for SD750 drives, object Parameter will have as many instances implemented as configuration and visualization parameters exist in the drive. They will be organized consecutively, and the first instance will be number 1. Refer to the *Software and Programming Manual* of the SD750 drive to consult details about each parameter, as well as their Modbus address.

NOTICE

Object Parameter is a list of all configuration and visualization parameters of the SD750 drive. Each parameter is an instance, starting with the first configuration parameter.

This list will very according to the software version of the drive microprocessor. Customer must ensure that the *Software and Programming Manual* matches the software version installed. For further information, contact Power Electronics.

Object Motor Data

This object serves as a database for motor parameters. It has **only one instance,** with the following attributes:

Attribute ID	Name	Data type	Default	Units	Description	Services
3	Motor type	USINT	0		0 - Non-standard motor	Get
6	Rated current	UINT	-	100mA	Rated stator current	Get/Set
7	Rated voltage	UINT	-	V	Rated base voltage	Get/Set
8	Rated power	UDINT	-	W	Rated power at rated freq	Get/Set
9	Rated freq	UINT	-	Hz	Rated electrical frequency	Get/Set
11	Max speed	UINT	-	rpm	Max allowed motor speed	Get/Set

Object Control Supervisor

This object models all the management functions for devices within the "Hierarchy of Motor Control Devices". It **only has one instance**. Instance attributes

Attribute ID	Name	Data type	Default	Description	Services
3	Run1	BOOL	-	Run forward control	Get/Set
4	Run2	BOOL	-	Run reverse control	Get/Set
5	NetCtrl	BOOL	-	RUN/STOP Control 0: Local Control 1: Network Control	Get/Set
6	State	USINT	0	Drive status 0: Vendor specific 1: Start-up 2: Not Ready 3: Ready 4: Enabled 5: Stopping 6: Fault Stop 7: Faulted	Get
7	Running1	BOOL	-	Running in forward direction	Get
8	Running2	BOOL	-	Running in reverse direction	Get
9	Ready	BOOL	-	1: Drive Ready or Enabled or Stopping 0: Other status	Get
10	Faulted	BOOL	-	1: Fault (latched) 0: No Faults Present	Get
11	Warning	BOOL	-	1: Warning (not latched) 0: No Warnings Present	Get

Attribute ID	Name	Data type	Default	Description	Services
12	FaultRst	BOOL	-	0 ➔ 1: Fault Reset 0: No action	Get/Set
13	FaultCode	UINT	-	Fault code See section 5.	Get
14	WarnCode	UINT	-	Warning code See section 5.	Get
15	CtrlFromNet	BOOL	-	Status of control 0: Local control 1: Control from network	Get
16	ForceFault	USINT	-	Action on loss of CIP network 0: Fault + Stop. 1: Ignore (Warning optional). 2: Vendor specific.	Get
17	ForceStatus	BOOL	-	Status of the forced fault 0: Not forced 1: Forced	Get/Set

EN

Object AC Drive

This object models the functions specific to an AC or DC drive, e.g., speed ramp, torque control etc. It has **only one instance,** with the following attributes:

Attribute ID	Name	Data type	Units	Description	Services
3	AtReference	BOOL	-	Drive working at current reference (speed or torque, depending on the working mode)	Get
4	NetRef	BOOL	-	Configuration of torque or speed reference mode setting (local or from network). 0: Local reference setting 1: Reference setting from network	Get/Set
5	NetProc	BOOL	-	Configuration of process control setting (local or from network). 0: Local reference setting 1: Reference setting from network	Get/Set
6	DriveMode	USINT	-	Drive mode 0: Vendor specific 1: Open loop speed (frequency) 2: Closed loop speed control 3: Torque control 4: Process control 5: Position control	Get
7	SpeedActual	INT	rpm	Actual drive speed[1]	Get
8	SpeedRef	INT	rpm	Speed reference[1]	Get/Set
9	CurrentActual	INT	100mA	Actual motor phase current[1]	Get
10	CurrentLimit	INT	100mA	Motor phase current limit[1]	Get
11	TorqueActual	INT	[0.01%]	Actual torque applied to the motor[1]	Get
12	TorqueRef	INT	[0.01%]	Torque reference[1]	Get/Set
13	ProcessActual	INT	[%]	Actual process control value[1]	Get
14	ProcessRef	INT	[%]	Process control reference set point[1]	Get/Set
15	PowerActual	INT	W	Actual output power[1]	Get
16	InputVoltage	INT	V	Input voltage[1]	Get

[1] **Note:** Attributes **7 to 21** (both inclusive) are scaled to 2 ^ scale factor. This scale factor is taken from attributes **22 to 28**; but if its value is fixed at 0 no scaling will be applied.

Attribute ID	Name	Data type	Units	Description	Services
17	Output voltage	INT	V	Output voltage	Get
18	AccelTime	UINT	[ms]	Acceleration time	Get/Set
19	DecelTime	UINT	[ms]	Deceleration time	Get/Set
20	LowSpdLimit	UINT	rpm	Minimum speed limit	Get/Set
21	HighSpdLimit	UINT	rpm	Maximum speed limit	Get/Set
22	SpeedScale	INT	-	Speed scaling factor[1] - Scaled speed = rpm / 2^SpeedScale - Range: -128 to 127	Get/Set
23	CurrentScale	INT	-	Current scaling factor [1] - Scaled current = A / 2^CurrentScale - Range: -128 to 127	Get/Set
24	TorqueScale	INT	-	Torque scaling factor [1] - Scaled torque= Nm / 2^TorqueScale - Range: -128 to 127	Get/Set
25	ProcessScale	INT	-	Process scaling factor - Scaled process = % / 2^ProcessScale - Range: -128 to 127	Get/Set
26	PowerScale	INT	-	Power scaling factor [1] - Scaled power = W / 2^PowerScale - Range: -128 to 127	Get/Set
27	VoltageScale	INT	-	Voltage scaling factor [1] - Scaled voltage = V / 2^VoltageScale - Range: -128 to 127	Get/Set
28	TimeScale	INT	-	Time scaling factor [1] - Scaled time = ms / 2^TimeScale - Range: -128 to 127	Get/Set
29	RefFromNet	BOOL	-	Status of torque/speed reference 0: Local torque/speed reference 1: Torque/speed reference from network	Get
30	ProcFromNet	BOOL	-	Status of process control reference	Get

Object Custom Modbus

This is a customized object, with a **single instance**. Each attribute represents a custom Modbus value (1 ... 32). Attributes 1 to 16 are read-only for the master (Get) and from 17 to 32 are write-only (Set).

Note: For the correct operation of this object, user must configure the custom parameters in groups G21.3.2 and G21.3.3 of the SD750. Attributes of the object Custom Modbus make use of these parameters so the user is capable of configuring them with the variables from the SD750 drive that are most relevant to him.

Attribute ID	Name	Data type
1	Modbus Var 1	UINT
2	Modbus Var 2	UINT
3	Modbus Var 3	UINT
4	Modbus Var 4	UINT
5	Modbus Var 5	UINT
6	Modbus Var 6	UINT
7	Modbus Var 7	UINT
8	Modbus Var 8	UINT

[1] **Note:** Attributes **7 to 21** (both included) have a scale 2^scale factor. This scale factor is taken from attributes 22 to 28, but if their value is set to 0 no scaling will be applied.

Attribute ID	Name	Data type
9	Modbus Var 9	UINT
10	Modbus Var 10	UINT
11	Modbus Var 11	UINT
12	Modbus Var 12	UINT
13	Modbus Var 13	UINT
14	Modbus Var 14	UINT
15	Modbus Var 15	UINT
16	Modbus Var 16	UINT
17	Modbus Var 31	UINT
18	Modbus Var 32	UINT
19	Modbus Var 33	UINT
20	Modbus Var 34	UINT
21	Modbus Var 35	UINT
22	Modbus Var 36	UINT
23	Modbus Var 37	UINT
24	Modbus Var 38	UINT
25	Modbus Var 39	UINT
26	Modbus Var 40	UINT
27	Modbus Var 41	UINT
28	Modbus Var 42	UINT
29	Modbus Var 43	UINT
30	Modbus Var 44	UINT
31	Modbus Var 45	UINT
32	Modbus Var 46	UINT

EN

Object TCP/IP

The TCP/IP interface object provides the mechanism to configure a device's TCP/IP network interface. Examples of configurable items include the device's IP Address, Network Mask, and Gateway Address.

The Ethernet/IP Adapter stack supports exactly **one instance** of the TCP/IP interface object.

Attribute ID	Name	Data type	Description	Services
1	Status	DWORD	Interface status. Bit 0: Not configured Bit 1: Configured	Get
2	Config capability	DWORD	Interface capability flags. Only bit 2 is used: 0: Disable DHCP 1: Enable DHCP	Get
3	Config control	DWORD	Interface control flags. **Note:** This attribute is defined, but not implemented.	Get
4	Physical link path	Struct	Path to physical link object. **Note:** This attribute is defined, but not implemented.	Get
5	Interface config	Struct	Interface configuration structure: • IP Address (UDINT) • Network Mask (UDINT) • Gateway Address (UDINT)	Get/Set
6	Host name	STRING	Device's host name. **Note:** This attribute is defined, but not implemented.	Get

Object Ethernet Link

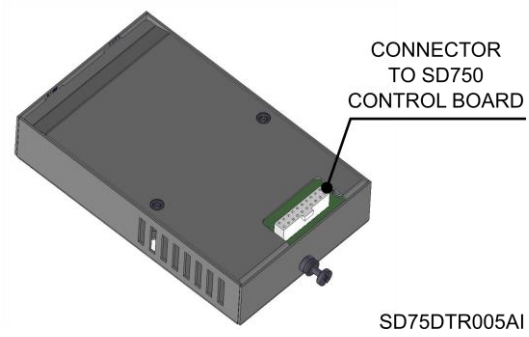
The Ethernet Link object maintains link-specific status information for the Ethernet communications interface. Since the CoIn expansion board is a multi-port device, it holds more than one instance of this object. Usually, when using the 2-port switch, instance 1 is assigned Ethernet port 0 and instance 2 is assigned Ethernet port 1.

Attribute ID	Name	Data type	Description	Services
1	Interface speed	UDINT	Speed in Mbps	Get
2	Interface flags	DWORD	Interface status flags Bit 0: Connection status Bit 1: 0: Half Duplex 1: Full Duplex	Get
3	MAC address	ARRAY	Array of 6 UINTs with MAC address	Get
6	Interface control	Struct	Configuration for physical interface	Get
10	Interface label	SHORT_STRING	Human readable identification	Get

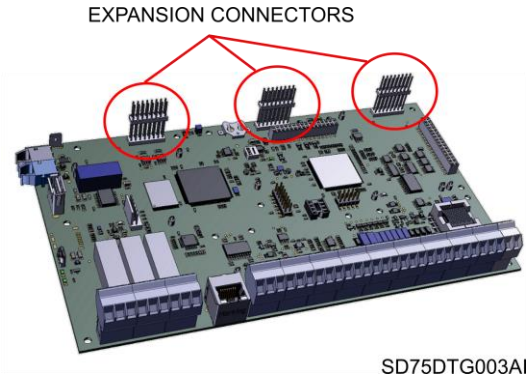
CONNECTION TO THE DRIVE

3

The Ethernet/IP board can be connected directly, through the connector on its back side, to any of the three expansion connectors of the SD750 drive central control board. Once connected, it allows integrating the drive in an Ethernet local area network (LAN) with TCP/IP or Ethernet/IP network protocol. One Ethernet/IP board will be necessary for each equipment which is going to be connected to such network.



SD75DTR005AI



SD75DTG003AI

EN

CAUTION

Power Electronics' SD750 drives operate with a high electric energy.

Make sure the power supply has been disconnected and wait for at least 10 minutes to guarantee that DC bus is discharged before installing the Ethernet/IP board. Otherwise, there is a risk of personal injuries or accidents.

COMMISSIONING

4

The Ethernet/IP expansion board allows configuring the SD750 drive as an Ethernet/IP industrial communication slave. Once loaded the EDS[1] in the Ethernet/IP master, the objects to monitor must be configured.

NOTICE

Consult implemented objects and their attributes in section CIP classes.

The following steps must be followed:

1. Connect the expansion board and ensure communication with the SD750 is correct (G23). Configure action in case of communication fault.
2. Load EDS file in the PLC master.
3. Configure network parameters (G21) so they match with the data expected by the master.
4. Configure in the master the objects to exchange between master and slave. In case object Custom Modbus is used, configure variables in groups G21.3.2 and G21.3.3.
5. Verify communication is stablished. Configure action in case of communication faults with the master.
6. Configure control parameters G21.3.4 - G21.3.6 based on the application.

The rest of the commissioning depends on the PLC, the program used, etc.

[1] Electronic Data Sheet. It is the file that defines a device.

Parameters setting

Up next, the relevant parameters are summarized. **For details about the range of values and Modbus addresses, refer to the Software and Programming Manual for SD750 drives.**

Subgroup 20.6: Custom Modbus configuration

Subgroup G20.6 of the SD750 drive allows configuring consecutive registers of the Modbus map:

Screen	Range	Function	Set on RUN
G20.6.1 Custom modbus addr 1 = 0 **G20.6.2 Custom modbus addr 2 = 0** ... **G20.6.120 Custom modbus addr 120 = 0**	0 to 65535	These parameters allow configuring 120 consecutive registers (4500 to 4619) variables from the Modbus map as required. This is particularly useful when designing a SCADA, so that the client can consult several registers in a single reading operation. They are grouped as follows: • Subgroup 20.6.1: Values 1 to 30 • Subgroup 20.6.2: Values 31 to 60 • Subgroup 20.6.3: Values 61 to 90 • Subgroup 20.6.4: Values 91 to 120 In parameters G20.6.x, user must enter the Modbus registers (Modbus address – 40001) that will be pointed to. Once configured, parameters G20.7.x can be used to read or write the value of each register. **Example**: Let us suppose we want to store the local speed reference (G3.3, Modbus 40053). We must configure register 52 (40053 – 1) in G20.6.1, at *Custom modbus addr1*. Then, in G20.7.1, *Custom modbus val 1* we will read the current value of the local speed reference. To modify it, we must enter the new value and save changes. **Note**: When reading or writing a variable, keep in mind the type of variable and its Modbus range to ensure values are interpreted correctly.	YES

Subgroup 20.7: Custom Modbus values

Subgroup G20.7 of the SD750 drive allows reading and writing the registers of consecutive G20.6 group records:

Screen	Range	Function	Set on RUN
G20.7.1 Custom modbus val1 = 0 **G20.7.2 Custom modbus val2 = 0** ... **G20.7.120 Custom modbus val30 = 0**	0 to 65535	These parameters can be used to read and write the values of the registers that were previously configured in G20.6. They are grouped as follows: • Subgroup 20.7.1: Values 1 to 30 • Subgroup 20.7.2: Values 31 to 60 • Subgroup 20.7.3: Values 61 to 90 • Subgroup 20.7.4: Values 91 to 120 **Note**: When reading or writing a variable, keep in mind the type of variable and its Modbus range to ensure values are interpreted correctly.	YES

EN

Subgroup 21.1: Ethernet

Sub-group G21 of the SD750 drive allows you to configure and display the status of the Ethernet IP board

<table>
<tr><th>Screen</th><th>Range</th><th>Function</th><th>Set on RUN</th></tr>
<tr><td>G21.1.1 Automatic IP = No</td><td>No
Yes</td><td>Allows assigning the parameters automatically.
<table><tr><th>OPT.</th><th>FUNCTION</th></tr><tr><td>No</td><td>Drive will take the IP, subnet mask and gateway addresses set by the user in [G21.1.2, G21.1.3 and G21.1.4].</td></tr><tr><td>Yes</td><td>The drive requests and receives the parameters of the IP, Subnet Mask and Gateway addresses from the Network Server. DHCP protocol is used.</td></tr></table></td><td>YES</td></tr>
<tr><td>G21.1.1.1 Assigned IP = 0.0.0.0</td><td>0 to 255</td><td>Shows the drive IP address, regardless of whether it was assigned automatically or by the user in parameter [G21.1.2].</td><td>YES</td></tr>
<tr><td>G21.1.1.2 Assigned subnet = 0.0.0.0</td><td>0 to 255</td><td>Shows the drive subnet mask, regardless of whether it was assigned automatically or by the user in parameter [G21.1.3].</td><td>YES</td></tr>
<tr><td>G21.1.1.3 Assigned gateway = 0.0.0.0</td><td>0 to 255</td><td>Shows the drive gateway address, regardless of whether it was assigned automatically or by the user in parameter [G21.1.4].
Format of Gateway Address is: A.B.C.D.</td><td>YES</td></tr>
<tr><td>G21.1.2 IP address = 192.168.1.143</td><td>0 to 255</td><td>Sets the IP address of the equipment in the user local network. This address must be provided by the local network administrator.
Format of the IP address is: A.B.C.D.
To configure the address, enter a value in each of the four parameters that compose it.</td><td>NO</td></tr>
<tr><td>G21.1.3 Subnet Mask = 255.255.255.0</td><td>0 to 255</td><td>Sets the subnet mask address of the equipment in the user local network. This address must be provided by the local network administrator.
Format of the subnet mask is: A.B.C.D.
To configure the address, enter a value in each of the four parameters that compose it.</td><td>NO</td></tr>
<tr><td>G21.1.4 Gateway = 0.0.0.0</td><td>0 to 255</td><td>Sets the gateway address of the equipment in the user local network. This address must be provided by the local network administrator.
Format of the gateway address is: A.B.C.D.
To configure the address, enter a value in each of the four parameters that compose it.</td><td>NO</td></tr>
<tr><td>G21.1.5 MAC address= A.B.C.D.E.F</td><td>0x00 to 0xFF</td><td>Sets the MAC address. This address must be unique and exclusive and depends on the LAN board. This address must be provided by Power Electronics.
Format of the MAC address is: A.B.C.D.E. F.
To configure the address, enter a value in each of the six parameters that compose it.</td><td>NO</td></tr>
</table>

Subgroup 21.3: Ethernet/IP

Subgroup 21.3 of the SD750 drive allows configuring Ethernet/IP.

Subgroup 21.3.1: Ethernet / IP

G21.3.1: Network parameters configuration group.

Screen	Range	Function	Set on RUN
G21.3.1.1 Automatic IP = No	No Yes	Allows the possibility of assigning the parameters automatically. **OPT.** / **FUNCTION** No: Drive will take the IP, subnet mask and gateway addresses set by the user in [G21.3.1.2, G21.3.1.3 and G21.3.1.4]. Yes: The drive requests and receives the parameters of the IP, Subnet Mask and Gateway addresses from the Network Server. DHCP protocol is used.	YES
G21.3.1.2 Assigned IP = 0.0.0.0	0 to 255	Shows the assigned IP address. **Available if G23.1.3.1 = Yes.**	YES
G21.3.1.3 Assigned subnet = 0.0.0.0	0 to 255	Shows the assigned subnet mask. **Available if G23.1.3.1 = Yes.**	YES
G21.3.1.4 Assigned gateway = 0.0.0.0	0 to 255	Shows the assigned gateway address. **Available if G23.1.3.1 = Yes**.	YES
G21.3.1.2 IP address = 192.168.1.143	0 to 255	**Available if G23.1.3.1 = No.** Sets the IP address of the equipment in the user local network. This address must be provided by the local network administrator. Format of the IP address is: A.B.C.D. To configure the address, enter a value in each of the four parameters that compose it.	NO
G21.3.1.3 Subnet Mask = 255.255.255.0	0 to 255	**Available if G23.1.3.1 = No.** Sets the subnet mask address of the equipment in the user local network. This address must be provided by the local network administrator. Format of the subnet mask is: A.B.C.D. To configure the address, enter a value in each of the four parameters that compose it.	NO
G21.3.1.4 Gateway = 0.0.0.0	0 to 255	**Available if G23.1.3.1 = No.** Sets the gateway address of the equipment in the user local network. This address must be provided by the local network administrator. Format of the gateway address is: A.B.C.D. To configure the address, enter a value in each of the four parameters that compose it.	NO
G21.3.1.5 MAC address= A.B.C.D.E.F	0x00 to 0xFF	Sets the MAC address. This address must be unique and exclusive and depends on the LAN board. This address must be provided by Power Electronics. Format of the MAC address is: A.B.C.D.E. F. To configure the address, enter a value in each of the six parameters that compose it.	NO

Note: Please notice parameters G21.3.1.2, G21.3.1.3 and G21.3.1.4 vary depending on whether automatic IP is enabled or not (parameter G21.3.1.1).

Subgroup 21.3.2: Master's input

G21.3.2: Configuration group of the master's input variables for object Custom Modbus (PE Status). The number of variables to configure will depend on the number of variables selected in the master.

Subgroup 21.3.2.1: Addresses

Screen	Range	Function	Set on RUN
Custom modbus addr1: 0 **Custom modbus addr2: 0** ... **Custom modbus addr16: 0**	0 to 65535	See "Subgroup 20.6: Custom Modbus address". **Note:** If these values are modified, the configuration of G20.6, G20.7, G21.3.3, G21.4.2 and G21.4.3 will change.	YES

Subgroup 21.3.2.2: Values

Screen	Range	Function	Set on RUN
Custom modbus val1: 0 **Custom modbus val2: 0** ... **Custom modbus val16: 0**	0 to 65535	See "Subgroup 20.7: Custom Modbus values". **Note:** If these values are modified, the configuration of G20.6, G20.7, G21.3.3, G21.4.2 and G21.4.3 will change.	YES

Subgroup 21.3.3: Master's output

G21.3.3: Configuration group of the master's output variables for object Custom Modbus (PE Status). The number of variables to configure will depend on the number of variables selected in the master

Subgroup 21.3.3.1: Addresses

Screen	Range	Function	Set on RUN
Custom modbus addr31: 0 **Custom modbus addr32: 0** ... **Custom modbus addr46: 0**	0 to 65535	See "Subgroup 20.6: Custom Modbus address". **Note:** If these values are modified, the configuration of G20.6, G20.7, G21.3.2, G21.4.2 and G21.4.3 will change.	YES

Subgroup 21.3.3.2: Values

Screen	Range	Function	Set on RUN
Custom modbus val31: 0 **Custom modbus val32: 0** ... **Custom modbus val46: 0**	0 to 65535	See "Subgroup 20.7: Custom Modbus values". **Note:** If these values are modified, the configuration of G20.6, G20.7, G21.3.2, G21.4.2 and G21.4.3 will change.	YES

Others

Screen	Range	Function	Set on RUN
G21.3.4 Control mode = Local	Local Network Net decides	Allows defining who controls the equipment (sends start, stop, fault and reset commands). **OPT.** / **FUNCTION** Local — Control mode depends on G4.1.1 and G4.1.2 settings. The PLC will never be able decide what to do, even if it is communicating with the drive. Network — The network commands the drive. Net decides — The PLC decides which of the above configurations to use. **Note:** When network mode is enabled, the value of parameters G4.1.1 and G4.1.2 will be forced to "Ethernet IP". If configuration changes, these parameters will return to their previous value.	NO

<table>
<tr><th>Screen</th><th>Range</th><th>Function</th><th>Set on RUN</th></tr>
<tr><td>G21.3.5 Reference mode = Local</td><td>Local
Network
Net decides</td><td>Allows defining who sets the direct reference (V or P) for the equipment.

<table><tr><th>OPT.</th><th>FUNCTION</th></tr><tr><td>Local</td><td>Ethernet/IP master cannot modify the references. Reference will be set through local control.</td></tr><tr><td>Network</td><td>Reference will be given by the network.</td></tr><tr><td>Net decides</td><td>The PLC decides which of the above configurations to use.</td></tr></table>
Note: When network mode is enabled, the value of parameters G3.1, G3.2, G3.4 and G3.5 will be forced to “Ethernet IP”. If configuration changes, these parameters will return to their previous value.</td><td>NO</td></tr>
<tr><td>G21.3.6 PID mode = Local</td><td>Local
Network
Net decides</td><td>Allows defining who sets the direct reference (V or P) for the PID.

<table><tr><th>OPT.</th><th>FUNCTION</th></tr><tr><td>Local</td><td>Ethernet/IP master cannot modify modify the parameters of the PID. Those will be set through local control.</td></tr><tr><td>Network</td><td>PID parameters will be given by the network.</td></tr><tr><td>Net decides</td><td>The PLC decides which of the above configurations to use.</td></tr></table>
Note: When network mode is enabled, the value of parameter G6.1 will be forced to “Ethernet IP”. If configuration changes, this parameter will return to its previous value.</td><td>NO</td></tr>
<tr><td>G21.3.7 Connector 1 status = Off</td><td>Off
On</td><td>Read-only parameter, shows whether the Ethernet/IP connector 1 is off or on.</td><td>NO</td></tr>
<tr><td>G21.3.8 Fault mode c1 = Fault[1]</td><td>Fault
Ignore</td><td>Enables the fault associated to the Ethernet/IP connector 1.

<table><tr><th>OPT.</th><th>FUNCTION</th></tr><tr><td>Fault</td><td>If the timeout is overcome, fault F60:Lost CIP c1 comms will be triggered.</td></tr><tr><td>Ignore</td><td>Fault F60 will not be triggered.</td></tr></table></td><td>YES</td></tr>
<tr><td>G21.3.9 Connector 2 status = Off</td><td>Off
On</td><td>Read-only parameter, shows whether the Ethernet/IP connector 2 is off or on.</td><td>NO</td></tr>
<tr><td>G21.3.10 Fault mode c2 = Fault[1]</td><td>Fault
Ignore</td><td>Enables the fault associated to the Ethernet/IP connector 1.

<table><tr><th>OPT.</th><th>FUNCTION</th></tr><tr><td>Fault</td><td>If the timeout is overcome, fault F112:Lost CIP c2 comms will be triggered.</td></tr><tr><td>Ignore</td><td>Fault F112 will not be triggered.</td></tr></table></td><td>YES</td></tr>
<tr><td>G21.3.11 Client comms status = OK</td><td>Ok
Timeout</td><td>Read-only parameter, shows whether the client Ethernet/IP communication status is working properly or not.</td><td>YES</td></tr>
<tr><td>G21.3.12 Client comms fault = Fault</td><td>Fault
Ignore</td><td>Enables timeout fault associated to the client Ethernet/IP communication.

<table><tr><th>OPC.</th><th>FUNCTION</th></tr><tr><td>Fault</td><td>If the timeout is overcome, fault F62:CIP client comm lost will be triggered.</td></tr><tr><td>Ignore</td><td>Fault F62 will not be triggered.</td></tr></table></td><td>YES</td></tr>
<tr><td>G21.3.13 Client comms timeout = 4 s</td><td>1 to 60 s</td><td>Sets the timeout before G21.3.11 Client comms status changes to Timeout.</td><td>YES</td></tr>
</table>

EN

[1] The status of connectors indicates the of communication status with the Ethernet/IP master. In case the application does not require ring connection, the SD750 will have to be configured to ignore the fault of the unused connector(F60: connector 1, F112: connector 2).

Subgroup G23.3: Communications

Subgroup G23.3 allows visualizing the status of communication between the COIN and the SD750.

- Warning 49: Communication COIN – SD750
- Fault 109: Communication COIN – SD750

<table>
<tr><th>Screen</th><th>Range</th><th>Function</th><th>Set on RUN</th></tr>
<tr><td>G23.3.4 EthernetIP board state = Off</td><td>Off
On</td><td>Shows the status of the Ethernet/IP board.
<table><tr><th>OPT.</th><th>FUNCTION</th></tr><tr><td>Off</td><td>The board is not connected.</td></tr><tr><td>On</td><td>The board is connected</td></tr></table></td><td>NO</td></tr>
<tr><td>G23.3.5 EthernetIP board test = No</td><td>No
Yes</td><td>Enables the LED fast blinking. This is useful to locate the board in case several boards of the same type are connected.
Note: This parameter will only appear if an Ethernet/IP board has been connected.</td><td>NO</td></tr>
<tr><td>G23.3.6 EthernetIP Com Error = Fault</td><td>Off
Warning
Fault</td><td>Allows defining the behavior of the drive in case communication with the Ethernet/IP board is lost.
<table><tr><th>OPT.</th><th>FUNCTION</th></tr><tr><td>Off</td><td>Drive will remain operating normally.</td></tr><tr><td>Warning</td><td>Warning “W49:EthernetIP” expansion will be triggered.</td></tr><tr><td>Fault</td><td>Fault “F109:Exp EthernetIP comm” will be triggered and the drive will stop.</td></tr></table></td><td>NO</td></tr>
</table>

Configuration parameters

The configuration parameters for the Ethernet board are summarized below:

Parameter	Screen	Range	Register range	Access [1]
G20.6.1 to G20.6.120	Custom Modbus addresses 1 to 120	0 to 65535	0 to 65535	RW
G20.7.1 to G20.7.120	Values of custom Modbus registers 1 to 120	0 to 65535	0 to 65535	RW
G21.1.1	Automatic IP = No	No Yes	0 to 1	RW
G21.1.1.1	Assigned IP = 0.0.0.0 (A.B.C.D)	0 to 255	0 to 255	RO
G21.1.1.2	Assigned subnet = 0.0.0.0 (A.B.C.D)	0 to 255	0 to 255	RO
G21.1.1.3	Assigned gateway = 0.0.0.0 (A.B.C.D)	0 to 255	0 to 255	RO
G21.1.2	IP address = 192.168.1.143 (A.B.C.D)	0 to 255	0 to 255	RW
G21.1.3	Subnet Mask = 255.255.255.0 (A.B.C.D)	1 to 255	1 to 255	RW
G21.1.4	Gateway = 0.0.0.1 (A.B.C.D)	0 to 255	0 to 255	RW
G21.1.5	MAC address = 0.27.119.129.238.66 (A.B.C.D.E.F)	0 to 255	0 to 255	RW
G21.3.1.1	Automatic IP = No	No Yes	0 to 1	RW
G21.3.1.2	Assigned IP = 0.0.0.0 (A.B.C.D)	0 to 255	0 to 255	RO
G21.3.1.3	Assigned subnet = 0.0.0.0 (A.B.C.D)	0 to 255	0 to 255	RO
G21.3.1.4	Assigned gateway = 0.0.0.0 (A.B.C.D)	0 to 255	0 to 255	RO
G21.3.1.2	IP address = 192.168.1.143 (A.B.C.D)	0 to 255	0 to 255	RW
G21.3.1.3	Subnet Mask = 255.255.255.0 (A.B.C.D)	1 to 255	1 to 255	RW
G21.3.1.4	Gateway = 0.0.0.0 (A.B.C.D)	0 to 255	0 to 255	RW
G21.3.1.5	MAC address = 00-1B-77-81-EE-42 (A.B.C.D.E.F)	0 to 255	0 to 255	RW
G21.3.2.1.1 to G21.3.2.1.16	Custom Modbus addresses 1 to 16	0 a 65535	0 a 65535	RW
G21.3.2.2.1 to G21.3.2.2.16	Values of custom Modbus registers 1 to 16	0 a 65535	0 a 65535	RW
G21.3.3.1.1 to G21.3.3.1.16	Custom Modbus addresses 31 to 46	0 a 65535	0 a 65535	RW
G21.3.3.2.1 to G21.3.3.2.16	Values of custom Modbus registers 31 to 46	0 a 65535	0 a 65535	RW
G21.3.4	Control mode = Local	Local Network Network decides	0 1 2	RW
G21.3.5	Reference mode = Local	Local Network Network decides	0 1 2	RW
G21.3.6	PID mode = Local	Local Network Network decides	0 1 2	RW
G21.3.7	Connector 1 status = Off	Off On	0 to 1	RO
G21.3.8	Fault mode c1 = Fault	Fault Ignore	0 to 1	RW
G21.3.9	Connector 2 status = Off	Off On	0 to 1	RO
G21.3.10	Fault mode c2 = Fault	Fault Ignore	0 to 1	RW

EN

Parameter	Screen	Range	Register range	Access [1]
G21.3.11	Client comms status = OK	OK Timeoutt	0 to 1	RO
G21.3.12	Client comms fault = Fault	Fault Ignore	0 to 1	RW
G21.3.13	Client comms timeout = 4 s	1 to 60 s	1 to 60	RW
G23.3.4	EthernetIP board state = Off	Off On	0 to 1	RO
G23.3.5	EthernetIP board test = No	No Yes	0 to 1	RW
G23.3.6	EthernetIP Com Error = Fault	Off Warning Fault	0 1 2	RW

[1] **Access**: **RW**: Read and write. **RO**: Read only

Custom Modbus

Custom Modbus comes by default as detailed in the tables below. **For more information on Modbus addresses, refer to the Software and Programming Manual of the SD750 drive.**

Master´s input

Position	Variable	Modbus Address	Modbus Range
1	Comms Status	43585	Real Value= Modbus Value
2	Motor speed (%)	42003	Real Value= (Modbus Value / 100)
3	Motor current	42007	Real Value= (Modbus Value / 10)
4	Motor power	42010	Real Value= Modbus Value
5	Motor torque	42008	Real Value= (Modbus Value / 100)
6	Motor frequency	42005	Real Value= (Modbus Value / 10)
7	Motor voltage	42006	Real Value= Modbus Value
8	Motor phi cosine	42009	Real Value= (Modbus Value / 100)
9	DC bus voltage	42035	Real Value= Modbus Value
10	Speed reference	42001	Real Value= (Modbus Value / 100)
11	Drive temperature	42039	Real Value= Modbus Value
12	IGBT temperature	42040	Real Value= Modbus Value
13	L1-L2 supply voltage	42031	Real Value= Modbus Value
14	L2-L3 supply voltage	42032	Real Value= Modbus Value
15	AI1 percentage (%)	42062	Real Value= (Modbus Value / 100)
16	AI2 percentage (%)	42065	Real Value= (Modbus Value / 100)

Master´s output

Position	Variable	Modbus Address	Modbus Range
1	Host Start Control	43586	Real Value= Modbus Value
2	Host Comms Control (Ref)	43570	Real Value= Modbus Value
3	Host Reset Control	43588	Real Value= Modbus Value
4	Host Trip Control	43589	Real Value= Modbus Value
5	Acceleration rate 1	40181	Real Value= (Modbus Value / 100)
6	Deceleration rate 1	40182	Real Value= (Modbus Value / 100)
7	Main start mode	40224	Real Value= Modbus Value
8	Main stop mode	40221	Real Value= Modbus Value
9	Minimum speed limit 1min	40401	Real Value= (Modbus Value / 100)
10	Maximum speed limit 1min	40402	Real Value= (Modbus Value / 100)
11	Speed ref 1 source	40051	Real Value= Modbus Value
12	Torque ref 1 source	40054	Real Value= Modbus Value
13	Main control mode	40071	Real Value= Modbus Value
14	Current limit	40405	Real Value= (Modbus Value / 10000)
15	Torque limit	40409	Real Value= (Modbus Value / 100)
16	Regeneration I limit	40417	Real Value= (Modbus Value / 10000)

EN

FAULTS AND WARNINGS LIST

5

Please, consult the full list of fault and warning messages in the *Software and Programming Manual* for SD750 drives.

Description of Fault List

DISPLAY	DESCRIPTION
F60:Lost CIP c1 comms	The connection of the equipment (server) to the Ethernet / IP client (PLC) in connector 1 of the board has been lost. If there is no configuration to indicate another option, by default, the CIP standard forces the drive to stop the motor and trip Timeout fault.
F61:EIP Fault	Failure in the Ethernet/IP board. This fault is triggered by the PLC.
F62:CIP client comm lost	Client communications timeout exceeded.
F109:Exp EthernetIP comm	Failure in communication with the Ethernet/IP board.
F112:Lost CIP c2 comms	Failure in communication with the connector 1 of the Ethernet/IP board.
F123:Ethernet IP Exp Version	Expansion card version is prior to 2.2.0

Troubleshooting

DISPLAY	POSSIBLE CAUSE	ACTIONS
F60:Lost CIP c1 comms	The active connection with the Ethernet/IP Client has been lost.	Check the Ethernet/IP connection of the client (PLC, PC).
F61:EIP Fault	The PLC has detected a fault in the Ethernet/IP board.	Verify the cable of the board connector is properly connected. Verify the board is connected correctly.
F62:CIP client comm lost	Reserved.	Contact with Power Electronics.
F109:Exp EthernetIP comm	Communication with the Ethernet/IP board is not correct.	Verify the board is connected correctly. Consult with Power Electronics.
F3:PDINT	Communication with the connector 2 of the Ethernet/IPboard is not correct.	
F123:Ethernet IP Exp Version	Software of the connected board is out of date.	Update the expansion board software to the latest available version.

List of warning messages

Warning	Acronym	Name	Description
W49	**EIPE**	EthernetIP expansion	There is a communications problem with the Ethernet/IP board.

To map fault or warning codes, the code must be converted to hexadecimal. That will be the corresponding CIP code.

Examples:

Fault code PE	Display	CIP code (Hex)
0	No faults	0000
23	F23:Min speed limit	0017
60	F60:Lost CIP c1 comms	003C
61	F61:EIP Fault	003D
62	F62:CIP client comm lost	003E
109	F109:Exp EthernetIP comm	006D
112	F112:Lost CIP c2 comms	0070
123	F123:Ethernet IP Exp Version	007B

Warning code PE	Display	Name	CIP code (Hex)
1	MOL	Motor overload	0001
13	SLMAX	Max speed limit	000D
49	EIPE	EthernetIP expansion	0031

EN

POWER ELECTRONICS

24H TECHNICAL ASSISTANCE 365 DAYS A YEAR

FIND YOUR NEAREST DELEGATION
POWER-ELECTRONICS.COM/CONTACT/