CANopen

SSW900-CAN-W

User's Guide

CANopen User's Guide

Series: SSW900

Software version: 1.2X

Language: English

Document: 10006223748 / 00

Build 5251

Publication Date: 01/2019

The information below describes the reviews made in this manual.

Version	Revision	Description
V1.2X	R00	First edition

CONTENTS

ABOUT THE MANUAL

This manual supplies the necessary information for the operation of the SSW900 soft-starter using the CANopen protocol. This manual must be used together with the SSW900 user's manual and programming manual.

ABBREVIATIONS AND DEFINITIONS

ASCII American Standard Code for Information Interchange
CAN Controller Area Network
CiA CAN in Automation
CIP Common Industrial Protocol
CRC Cycling Redundancy Check
HMI Human-Machine Interface
ISO International Organization for Standardization
ODVA Open DeviceNet Vendor Association
OSI Open Systems Interconnection
PLC Programmable Logic Controller
ro Read only
rw Read/write
RTR Remote Transmission Request

NUMERICAL REPRESENTATION

Decimal numbers are represented by means of digits without suffix. Hexadecimal numbers are represented with the letter 'h' after the number. Binary numbers are represented with the letter 'b' after the number.

DOCUMENTS

The CANopen protocol was developed based on the following specifications and documents:

Document	Version	Source
CAN Specification	2.0	CiA
CiA DS 301 CANopen Application Layer and Communication Profile	4.02	CiA
CiA DRP 303-1 Cabling and Connector Pin Assignment	1.1.1	CiA
CiA DSP 306 Electronic Data Sheet Specification for CANopen	1.1	CiA
CiA DSP 402 Device Profile Drives and Motion Control	2.0	CiA
Planning and Installation Manual - DeviceNet Cable System	PUB00027R1	ODVA

1 MAIN CHARACTERISTICS

Below are the main characteristics for communication of the soft-starter SSW900 with CANopen accessory.

- Network management task (NMT).
- 4 transmission PDOs.
- 4 reception PDOs.
- Heartbeat Consumer.
- Heartbeat Producer.
- Node Guarding.
- SDO Client.
- SYNC producer/consumer.
- It is supplied with an EDS file for the network master configuration.
- Acyclic data available for parameterization.

2 INTERFACE DESCRIPTION

The SSW900 soft-starter has two Slots for accessories (Figura 2.1). Parameters S3.5.1 and S3.5.2 present which accessory was recognized by Slot.

The accessories can be connected to any Slot, but only one type of each communication accessory is allowed.

Read the user's manual of the SSW900 soft-starter before installing or using this accessory.

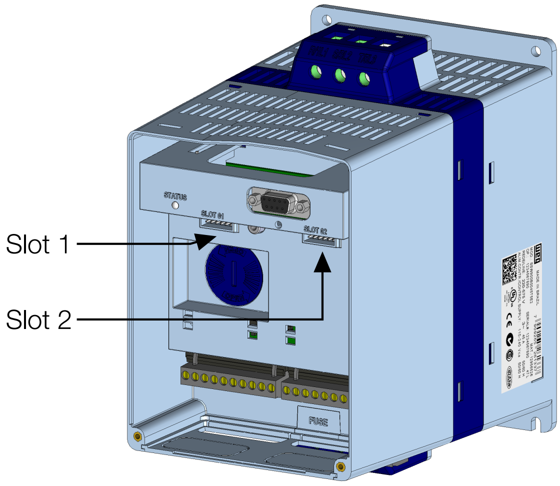
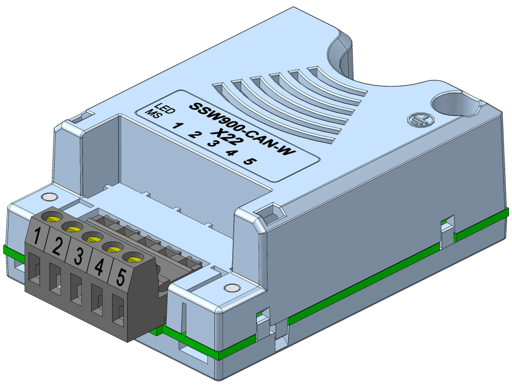


Figure 2.1: *Slots for accessories*

2.1 CANOPEN ACCESSORY



SSW900-CAN-W:

- Supplied items:
  - Installation guide.
  - CANopen/DeviceNet communication module.

2.2 CONNECTOR

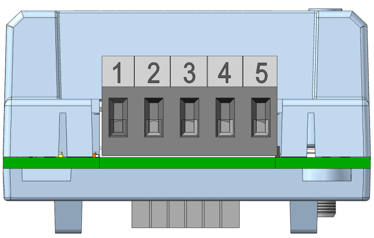


Table 2.1: *Pin assignment of connector for CANopen interface*

Pin	Name	Function
1	V-	Negative pole of the power supply
2	CAN_L	Communication signal CAN_L
3	Shield	Cable shield
4	CAN_H	Communication signal CAN_H
5	V+	Positive pole of the power supply

2.3 POWER SUPLLY

The power supply of the network must be able to supply enough current to power up the equipments and interfaces connected to the network. The data for individual consumption and input voltage are presented in tables 2.2 and 2.3.

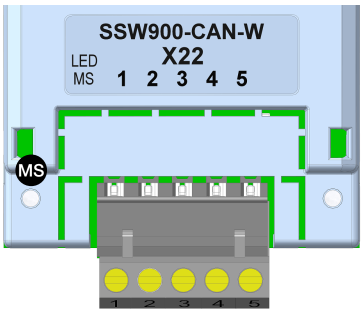
Table 2.2: *Power Suplly (Vdc)*

Minimum	Maximum	Recommended
11 V	30 V	24 V

Table 2.3: *Current*

Typical	Maximum
30 mA	50 mA

2.4 INDICATION LED



The MS LED indicates the conditions of the module itself. That is, whether it is able to work or not. The table below shows the possible states.

Table 2.4: *State of the CANopen module*

Status	Description	Comments
Off	No power	-
Green	Module operating and in normal conditions	-
Red	Module in error	Reinitializing the equipment is required.
Flashing green/red	Equipment performing self-diagnosis	It occurs during initialization.

3 CANOPEN NETWORK INSTALLATION

The CANopen network, such as several industrial communication networks, for being many times applied in aggressive environments with high exposure to electromagnetic interference, requires that certain precautions be taken in order to guarantee a low communication error rate during its operation. Recommendations to perform the connection of the product in this network are presented next.

NOTE!
Detailed recommendations on how to perform the installation are available at document ”Planning and Installation Manual” (item DOCUMENTS).

3.1 BAUD RATE

Equipments with CANopen interface generally allow the configuration of the desired baud rate, ranging from 10 kbit/s to 1 Mbit/s. The baud rate that can be used by the equipment depends on the length of the cable used in the installation. The table 3.1 shows the baud rates and the maximum cable length that can be used in the installation, according to the protocol recommendation.

Table 3.1: *Supported baud rates and cable length*

Baud Rate	Cable length
10 kbit/s	1000 m
20 kbit/s	1000 m
50 kbit/s	1000 m
100 kbit/s	600 m
125 kbit/s	500 m
250 kbit/s	250 m
500 kbit/s	100 m
800 kbit/s	50 m
1 Mbit/s	25 m

All network equipment must be programmed to use the same communication baud rate.

3.2 ADDRESS IN THE CANOPEN NETWORK

Each CANopen network device must have an address or Node-ID, and may range from 1 to 127. This address must be unique for each equipment.

3.3 TERMINATION RESISTOR

The use of termination resistors at the ends of the bus is essential to avoid line reflection, which can impair the signal and cause communication errors. Termination resistors of 121 Ω | 0.25 W must be connected between the signals CAN_H and CAN_L at the ends of the main bus.

3.4 CABLE

The connection of CAN_L and CAN_H signals must be done with shielded twisted pair cable. The following table shows the recommended characteristics for the cable.

Table 3.2: CANopen cable characteristics

Cable Length (m)	Resistence per Meter (mΩ/m)	Conductor Cross Section (mm^2)
0 ... 40	70	0.25 ... 0.34
40 ... 300	<60	0.34 ... 0.60
300 ... 600	<40	0.50 ... 0.60
600 ... 1000	<26	0.75 ... 0.80

It is necessary to use a twisted pair cable to provide additional 24Vdc power supply to equipments that need this signal. It is recommended to use a certified DeviceNet cable.

3.5 CONNECTION IN THE NETWORK

In order to interconnect the several network nodes, it is recommended to connect the equipment directly to the main line without using derivations. During the cable installation the passage near to power cables must be avoided, because, due to electromagnetic interference, this makes the occurrence of transmission errors possible.

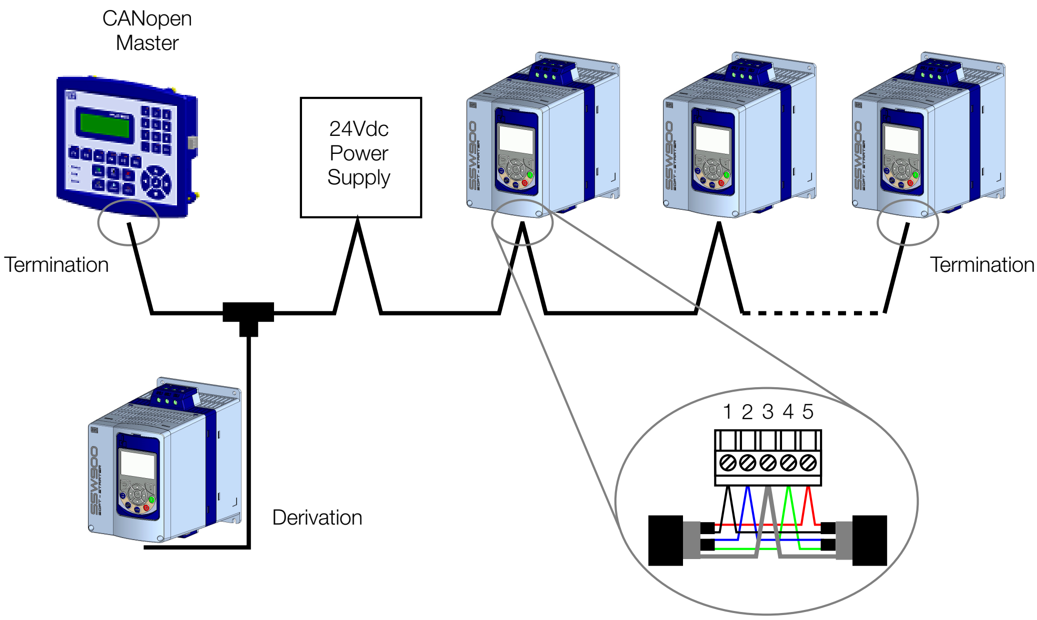


Figure 3.1: CANopen network installation example

In order to avoid problems with current circulation caused by difference of potential among ground connections, it is necessary that all the devices be connected to the same ground point.

To avoid voltage difference problems between the power supplies of the network devices, it is recommended that the network is fed by only one power supply and the signal is provided to all devices through the cable. If it is required more than one power supply, these should be referenced to the same point. Use the power supply to power the bus cable system only.

The maximum number of devices connected to a single segment of the network is limited to 64. Repeaters can be used for connecting a bigger number of devices.

4 S STATUS

Allows viewing of the SSW reading variables.

S5 COMMUNICATIONS

HMI monitoring parameters of the communication interfaces.

For a detailed description, refer to the Anybus-CC, CANopen, DeviceNet and Modbus RTU User's Manuals of the SSW according to the interface used.

S5.1 Status Word

.1 SSW	0 ... 15 Bit

Description:
Word of SSW status.

.1 SSW Word of SSW status.

Bit	Value/Description
Bit 0 Running	0: The motor is not enabled. 1: The motor is enabled.
Bit 1 Gener. Enabled	0: When it is general disabled by any mean. 1: When it is general enabled by all the means.
Bit 2 JOG	0: The JOG function is inactive. 1: The JOG function is active.
Bit 3 Initial Test	0: None. 1: During the initial tests before the motor starting.
Bit 4 Ramp Up	0: It is not accelerating. 1: During the whole acceleration.
Bit 5 Full Voltage	0: There is no full voltage applied to the motor. 1: Full voltage is being applied to the motor.
Bit 6 Bypass	0: With open bypass. 1: With closed bypass.
Bit 7 Ramp Down	0: It is not decelerating. 1: During the whole deceleration.
Bit 8 Remote	0: Local. 1: Remote.
Bit 9 Braking	0: It is not executing braking. 1: During the braking process.
Bit 10 FWD/REV	0: It is not reverting the rotation direction. 1: During the rotation reversion process.
Bit 11 Reverse	0: Forward rotation. 1: Reverse rotation.
Bit 12 Ton	0: None. 1: Time before start (C5.7.2).
Bit 13 Toff	0: None. 1: Time after stop (C5.7.3).
Bit 14 Alarm	0: The SSW is not in alarm condition. 1: The SSW is in alarm condition. Note: The active alarm codes can be read by means of the menu D2.1.
Bit 15 Fault	0: The SSW is not in fault condition. 1: The SSW is in fault condition. Note: The active fault code can be read by means of the menu D1.1.

S5.2 Command Word

.5 Slot1	0 ... 15 Bit
.6 Slot2	0 ... 15 Bit

Description:
Command word of all sources of the SSW. The RUN/STOP and JOG commands of the sources which are not active will be reset.

.5 Slot1 Control word via any communication accessory connected to Slot 1.

.6 Slot2 Command word via any communication accessory connected to Slot 2.

Bit	Value/Description
Bit 0 Start/Stop	**0**: stopping by ramp. **1**: starting by ramp.
Bit 1 Gener. Enabled	**0**: general disable. **1**: general enable.
Bit 2 JOG	**0**: no JOG. **1**: with JOG.
Bit 3 FWD/REV	**0**: clockwise CW. **1**: counterclockwise CCW.
Bit 4 LOC/REM	**0**: local. **1**: remote.
Bit 5 … 6 Reserved	
Bit 7 Reset	**0** → **1**: execute fault reset (if a fault is active). Note: Only in the 0 to 1 transition command.
Bit 8 … 15 Reserved	

NOTE!
If the RUN/STOP and JOG commands are by a certain source and it is active, only these commands can be viewed in S5.2. For security reasons, all the other commands of the other sources which are not active will be reset.

S5.3 Value for Outputs

.1 DO Value 0 ... 15 Bit

Description:
Value for digital and analog outputs via serial communication.

.1 DO Value Value for the digital outputs via network interfaces.

Bit	Value/Description
Bit 0 DO1	**0**: Inactive. **1**: Active.
Bit 1 DO2	**0**: Inactive. **1**: Active.
Bit 2 DO3	**0**: Inactive. **1**: Active.
Bit 3 … 15 Reserved	

S5.3.2 Value for AO

.1 AO in 10 bits 0 ... 1023

Description:
Value for the analog output via network interfaces.

.1 AO in 10 bits Value for the analog output via network interfaces: 0...1023. 0=0% and 1023=100%.

S5.7 CANopen/DeviceNet

.1 CAN Controller Status	0 ... 6
.2 Received Telegram	0 ... 65535
.3 Transmitted Telegram	0 ... 65535
.4 Bus Off Counter	0 ... 65535
.5 Lost Messages	0 ... 65535
.6 CANopen Comm. Status	0 ... 5
.7 CANopen Node State	0 ... 4

Description:
Status of the CAN communication accessory and the protocols that use this interface.

.1 CAN Controller Status It allows identifying if the CAN interface board is properly installed and if the communication presents errors.

Indication	Description
0 = Disabled	Inactive CAN interface. It occurs when CAN protocol is not programmed at C8.4.1.
1 = Auto-baud	CAN controller is trying to detect baud rate of the network (only for DeviceNet communication protocol).
2 = CAN Enabled	CAN interface is active and without errors.
3 = Warning	CAN controller has reached the warning state.
4 = Error Passive	CAN controller has reached the error passive state.
5 = Bus Off	CAN controller has reached the bus off state.
6 = No Bus Power	CAN interface does not have power supply between the pins 1 and 5 of the connector.

.2 Received Telegram This parameter works as a cyclic counter that is incremented every time a CAN telegram is received. It informs the operator if the device is being able to communicate with the network.

.3 Transmitted Telegram This parameter works as a cyclic counter that is incremented every time a CAN telegram is transmitted. It informs the operator if the device is being able to communicate with the network.

.4 Bus Off Counter It is a cyclic counter that indicates the number of times the device entered the bus off state in the CAN network.

.5 Lost Messages It is a cyclic counter that indicates the number of messages received by the CAN interface, but could not be processed by the device. In case that the number of lost messages is frequently incremented, it is recommended to reduce the baud rate used in the CAN network.

NOTE!
This counter is reset every time the device is switched off, a reset is performed or the parameter maximum limit is reached.

.6 CANopen Comm. Status It indicates the board state regarding the CANopen network, informing if the protocol has been enabled and if the error control service is active (Node Guarding or Heartbeat).

Indication	Description
0 = Disabled	CANopen protocol disabled.
1 = Reserved	
2 = Comm. Enabled	Communication enabled.
3 = ErrorCtrl.Enab	Communication enabled and error control service enabled (Node Guarding/Heartbeat).
4 = Guarding Error	Node Guarding error occurred.
5 = HeartbeatError	Heartbeat error occurred.

.7 CANopen Node State It operates as a slave of the CANopen network, and as such element it has a state machine that controls its behavior regarding the communication. This parameter indicates in which state the device is.

Indication	Description
0 = Disabled	CANopen protocol disabled.
1 = Initialization	Communication with the device is not possible during this stage, which is concluded automatically.
2 = Stopped	Only the NMT object is available.
3 = Operational	All the communication objects are available.
4 = PreOperational	It is already possible to communicate with the slave but its PDOs are not yet available for operation.

5 C CONFIGURATIONS

This menu allows the programming of all SSW configuration parameters.

C8 COMMUNICATION

To change information via communication network, the SSW has several standard protocols.

The following necessary accessories and protocols are available:

Protocol	Accessory
CANopen	SSW900-CAN-W
DeviceNet	SSW900-CDN-N, SSW900-CAN-W
EtherNet/IP	SSW900-CETH-IP-N
Modbus RTU	SSW900-CRS485-W
Modbus TCP	SSW900-CMB-TCP-N
Profibus DP	SSW900-CPDP-N
PROFINET IO	SSW900-CPN-IO-N

For further details regarding the SSW configuration to operate these protocols, refer to the SSW Communication Manual.

C8.4 CANopen/DeviceNet

Configuration for the SSW900-CAN-W communication accessory and protocols that use this interface.

C8.4 CANopen/DeviceNet
C8.4.1 Protocol
Range: 0 ... 2 **Default:** 2
Properties:

Description:
It allows selecting the desired protocol for the CAN interface.

Indication	Description
0 = Disabled	Disable CAN interface.
1 = CANopen	Enable CAN interface with CANopen protocol.
2 = DeviceNet	Enable CAN interface with DeviceNet protocol.

C8.4 CANopen/DeviceNet
C8.4.2 Address
Range: 0 ... 127 **Default:** 63
Properties:

Description:
It allows programming the address used for the CAN communication. It is necessary that each element of the network has an address different from the others. The valid addresses for this parameter depend on the protocol programmed in P0700:

- P0700 = 1 (CANopen): valid addresses: 1 to 127.
- P0700 = 2 (DeviceNet): valid addresses: 0 to 63.

NOTE!
After changing this configuration, for the modification to be effective, the change takes effect only if the CAN interface is not exchanging cyclic data with the network.

C8.4 CANopen/DeviceNet

C8.4.3 Baud Rate

Range: 0 ... 8 **Default:** 0
Properties:

Description:
It allows programming the desired baud rate for the CAN interface, in bits per second. This rate must be the same for all the devices connected to the network. The supported bauld rates for the device depend on the protocol programmed in the parameter C8.4.1:

- C8.4.1 = 1 (CANopen): It is possible to use any rate specified in this parameter, but it does not have the automatic baud rate detection function – autobaud.
- C8.4.1 = 2 (DeviceNet): only the 500, 250 and 125 Kbit/s rates are supported. Other options will enable the automatic baud rate detection function – autobaud.

After a successful detection, the baud rate parameter (C8.4.3) changes automatically to the detected rate. In order to execute the autobaud function again, it is necessary to change the parameter C8.4.3 to one of the 'Autobaud' options.

Indication	Description
0 = 1 Mbps/Auto	CAN baud rate (automatic detection for DeviceNet).
1 = Reserved	Reserved
2 = 500 Kbps	CAN baud rate.
3 = 250 Kbps	CAN baud rate.
4 = 125 Kbps	CAN baud rate.
5 = 100 Kbps/Auto	CAN baud rate (automatic detection for DeviceNet).
6 = 50 Kbps/Auto	CAN baud rate (automatic detection for DeviceNet).
7 = 20 Kbps/Auto	CAN baud rate (automatic detection for DeviceNet).
8 = 10 Kbps/Auto	CAN baud rate (automatic detection for DeviceNet).

NOTE!
After changing this configuration, for the modification to be effective, the change takes effect only if the CAN interface is not exchanging cyclic data with the network.

C8.4 CANopen/DeviceNet

C8.4.4 Bus Off Reset

Range: 0 ... 1 **Default:** 1
Properties:

Description:
It allows programming the inverter behavior when detecting a bus off error at the CAN interface.

Indication	Description
0 = Manual	If bus off occurs, the A134/F134 alarm will be indicated on the HMI and the communication will be disabled. In case of alarm, the action programmed in parameter C8.4.5.2 will be executed. In order that the inverter communicates again through the CAN interface, it will be necessary to disable and enable the interface, or reinitiate the device.
1 = Automatic	If bus off occurs, the communication will be reinitiated automatically and the error will be ignored. In this case the alarm will not be indicated on the HMI and the inverter will not execute the action programmed in C8.4.5.2.

C8.4.5 CAN Error

Protection against interruption in the CAN communication.

If for some reason there is an interruption in the CAN communication, a communication error will be indicated, alarm A133...A137 or fault F133...F137 will be shown on the HMI, depending on the programming of C8.4.5.1, and the action programmed in C8.4.5.2 will be executed.

It only occurs after the equipment is online. This error is only generated for the SSW900-CAN-W module.

C8.4.5 CAN Error

C8.4.5.1 Mode

Range: 0 ... 2 **Default:** 0

Properties:

Description:
It allows configuring the tripping mode of the protection against interruption in the CAN communication.

Indication	Description
0 = Inactive	No tripping.
1 = Fault	Trips as fault. Disables the motor.
2 = Alarm	Trips as alarm. Action described in C8.4.5.2.

C8.4.5 CAN Error

C8.4.5.2 Alarm Action

Range: 0 ... 4 **Default:** 0

Properties:

Description:
Action for the CAN communication interruption alarm.

The actions described in this parameter are executed through the writing of the respective bits in the control word of the SLOT to which the accessory SSW900-CAN-W is connected. Thus, for the commands to be effective, the equipment must be programmed to be controlled by the network interface used. This programming is done through menu C3.

Indication	Description
0 = Indicates Only	No action is taken; the equipment remains in the current state.
1 = Ramp Stop	The stop by ramp command is executed, and the motor stops according to the programmed deceleration ramp.
2 = General Disable	The equipment is general disabled, and the motor stops by inertia.
3 = Change to LOC	The equipment is commanded to local mode.
4 = Change to REM	The equipment is commanded to remote mode.

NOTE!
The alarm action will only have a function if the error tripping mode C8.4.5.1 is programmed for Alarm.

6 OPERATION IN THE CANOPEN NETWORK

6.1 ACCESS TO THE DATA

Each slave of the CANopen network has a list called object dictionary that contains all the data accessible via network. Each object of this list is identified with an index, which is used during the equipment configuration as well as during message exchanges. This index is used to identify the object being transmitted.

6.2 CYCLIC DATA

Cyclic data is the data normally used for status monitoring and equipment control. For CANopen protocol, the interface supports 4 receive PDOs and 4 transmit PDOs.

It is necessary the configuration to be made both at the slave and master.

6.3 ACYCLIC DATA

In addition to the cyclic data, the interface also provides acyclic data via SDO. Using this type of communication, you can access any equipment parameter. Access to this type of data is commonly done using instructions for reading or writing data, which should indicate the index and sub-index to the desired parameter. The item 7.4 describes how to address the parameters for SSW900 soft-starter.

6.4 COMMUNICATION OBJECTS - COB

There is a specific set of objects that are responsible for the communication among the network devices. Those objects are divided according to the type of data and the way they are sent or received by a device. The following communication objects (COBs) are described by the specification:

Table 6.1: *Types of Communication Objects (COBs)*

Type of object	Description
Service Data Object (SDO)	SDO are objects responsible for the direct access to the object dictionary of a device. By means of messages using SDO, it is possible to indicate explicitly (by the object index) what data is being handled. There are two SDO types: Client SDO, responsible for doing a read or write request to a network device, and the Server SDO, responsible for taking care of that request. Since SDO are usually used for the configuration of a network node, they have less priority than other types of message.
Process Data Object (PDO)	PDO are used for accessing equipment data without the need of indicating explicitly which dictionary object is being accessed. Therefore, it is necessary to configure previously which data the PDO will be transmitting (data mapping). There are also two types of PDO: Receive PDO and Transmit PDO. They are usually utilized for transmission and reception of data used in the device operation, and for that reason they have higher priority than the SDO.
Emergency Object (EMCY)	This object is responsible for sending messages to indicate the occurrence of errors in the device. When an error occurs in a specific device (EMCY producer), it can send a message to the network. In the case that any network device be monitoring that message (EMCY consumer), it can be programmed so that an action be taken (disabling the other devices, error reset, etc.).
Syncronization Object (SYNC)	In the CANopen network, it is possible to program a device (SYNC producer) to send periodically a synchronization message for all the network devices. Those devices (SYNC consumers) will then be able, for instance, to send a certain datum that needs to be made available periodically.
Network Management (NMT)	Every CANopen network needs a master that controls the other devices (slaves) in the network. This master will be responsible for a set of services that control the slave communications and their state in the CANopen network. The slaves are responsible for receiving the commands sent by the master and for executing the requested actions. The protocol describes two types of service: device control service, with which the master controls the state of each network slave, and error control service (Node Guarding an Heartbeat), with which the device sends periodic messages to inform that the connection is active.

All the communication of the slave with the network is performed using those objects, and the data that can be accessed are the existent in the device object dictionary.

6.5 COB-ID

A telegram of the CANopen network is always transmitted by a communication object (COB). Every COB has an identifier that indicates the type of data that is being transported. This identifier, called COB-ID has an 11 bit size, and it is transmitted in the identifier field of a CAN telegram. It can be subdivided in two parts:

Function Code				Address						
bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- Function Code: indicates the type of object that is being transmitted.
- Adrress: indicates with which network device the telegram is linked.

A table with the standard values for the different communication objects is presented next. Notice that the standard value of the object depends on the slave address, with the exception of the COB-ID for NMT and SYNC, which are common for all the network elements. Those values can also be changed during the device configuration stage.

Table 6.2: *COB-ID for the different objects*

COB	Function Code (bits 10-7)	COB-ID Resultant COB-ID (function + address)
NMT	0000	0
SYNC	0001	128 (80h)
EMCY	0001	129 - 255 (81h - FFh)
PDO1 (tx)	0011	385 - 511 (181h - 1FFh)
PDO1 (rx)	0100	513 - 639 (201h - 27Fh)
PDO2 (tx)	0101	641 - 767 (281h - 2FFh)
PDO2 (rx)	0110	769 - 895 (301h - 37Fh)
PDO3 (tx)	0111	897 - 1023 (381h - 3FFh)
PDO3 (rx)	1000	1025 - 1151 (401h - 47Fh)
PDO4 (tx)	1001	1153 - 1279 (481h - 4FFh)
PDO4 (rx)	1010	1281 - 1407 (501h - 57Fh)
SDO (tx)	1011	1409 - 1535 (581h - 5FFh)
SDO (rx)	1100	1537 - 1663 (601h - 67Fh)
Node Guarding/Heartbeat	1110	1793 - 1919 (701h - 77Fh)

6.6 EDS FILE

Each device on an CANopen network has an EDS configuration file, which contains information about the device functions on the network. This file is used by a master or configuration software to program devices present at CANopen network.

The EDS file is available from WEG website (http://www.weg.net). It is important to note if the EDS configuration file is compatible with the firmware version of the SSW900 soft-starter.

7 OBJECT DICTIONARY

The object dictionary is a list containing several equipment data which can be accessed via CANopen network. 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.

The CiA DS 301 document defines a set of minimum objects that every CANopen network slave must have. The objects available in that list are grouped according to the type of function they execute. The objects are arranged in the dictionary in the following manner:

Table 7.1: *Object dictionary groupings*

Index	Objects	Description
0001h - 025Fh	Data type definition	Used as reference for the data type supported by the system.
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.

7.1 DICTIONARY STRUCTRE

The general structure of the dictionary has the following format:

Index	Object	Name	Type	Access

- **Index:** indicates directly the object index in the dictionary.
- **Object:** describes which information the index stores (simple variable, array, record, etc.).
- **Name:** contains the name of the object in order to facilitate its identification.
- **Type:** indicates directly the stored data type. For simple variables, this type may be an integer, a float, etc. For arrays, it indicates the type of data contained in the array. For records, it indicates the record format according to the types described in the first part of the object dictionary (indexes 0001h – 0360h).
- **Access:** informs if the object in question is accessible only for reading (ro), for reading and writing (rw), or if it is a constant (const).

For objects of the array or record type, a sub-index that is not described in the dictionary structure is also necessary.

7.2 DATA TYPE

The first part of the object dictionary (index 0001h – 025Fh) describes the data types that can be accessed at a CANopen network device. They can be basic types, as integers and floats, or compound types formed by a set of entries, as records and arrays.

7.3 COMMUNICATION PROFILE - COMMUNICATION OBJECTS

The indexes from 1000h to 1FFFh in the object dictionary correspond to the part responsible for the CANopen network communication configuration. Those objects are common to all the devices, however only a few are obligatory. A list with the objects of this range that are supported by the soft-starter SSW900 is presented next.

Table 7.2: *Object list – Communication Profile*

Index	Object	Name	Type	Access
1000h	VAR	device type	UNSIGNED32	ro
1001h	VAR	error register	UNSIGNED8	ro
1005h	VAR	COB-ID SYNC	UNSIGNED32	rw
100Ch	VAR	quard time	UNSIGNED16	rw
100Dh	VAR	life time factor	UNSIGNED8	rw
1016h	ARRAY	consume heartbeat time	UNSIGNED32	rw
1017h	VAR	producer heartbeat time	UNSIGNED16	rw
1018h	RECORD	Identity Object	Identity	ro
Server SDO Parameter				
1200h	RECORD	1st Server SDO parameter	SDO Parameter	ro
Receive PDO Communication Parameter				
1400h	RECORD	1st receive PDO Parameter	PDO CommPar	rw
1401h	RECORD	2nd receive PDO Parameter	PDO CommPar	rw
1402h	RECORD	3rd receive PDO Parameter	PDO CommPar	rw
1403h	RECORD	4th receive PDO Parameter	PDO CommPar	rw
Receive PDO Mapping Parameter				
1600h	RECORD	1st receive PDO mapping	PDO Mapping	rw
1601h	RECORD	2nd receive PDO mapping	PDO Mapping	rw
1602h	RECORD	3rd receive PDO mapping	PDO Mapping	rw
1603h	RECORD	4th receive PDO mapping	PDO Mapping	rw
Transmit PDO Communication Parameter				
1800h	RECORD	1st transmit PDO Parameter	PDO CommPar	rw
1801h	RECORD	2nd transmit PDO Parameter	PDO CommPar	rw
1802h	RECORD	3rd transmit PDO Parameter	PDO CommPar	rw
1803h	RECORD	4th transmit PDO Parameter	PDO CommPar	rw
Transmit PDO Mapping Parameter				
1A00h	RECORD	1st transmit PDO mapping	PDO Mapping	rw
1A01h	RECORD	2nd transmit PDO mapping	PDO Mapping	rw
1A02h	RECORD	3rd transmit PDO mapping	PDO Mapping	rw
1A03h	RECORD	4th transmit PDO mapping	PDO Mapping	rw

These objects can only be read and written via the CANopen network, it is not available via the keypad (HMI) or other network interface. The network master, in general, is the equipment responsible for setting up the equipment before starting the operation. The EDS configuration file brings the list of all supported communication objects.

Refer to item 8 for more details on the available objects in this range of the objects dictionary.

7.4 MANUFACTURER SPECIFIC OBJECTS

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 SSW900, the whole list of parameters was made available in this object range. It is possible to operate the SSW900 by means of these parameters, carrying out any function that the inverter can execute. The parameters were made available starting from the index 2000h, and by adding their Net Id to this index their position in the dictionary is obtained. To identify how the parameters are distributed in the object dictionary, refer to the Appendix A.

In order to be able to program the SSW900 operation correctly via the CANopen network, it is necessary to know its operation through the parameters.

Refer to the SSW900 soft-starter programming manual for a complete list of the parameters and their detailed description.

8 COMMUNICATION OBJECTS DESCRIPTION

This item describes in detail each of the communication objects available for the SSW900 soft-starter. It is necessary to know how to operate these objects to be able to use the available functions for the SSW900 soft-starter communication.

8.1 IDENTIFICATION OBJECT

There is a set of objects in the dictionary which are used for equipment identification; however, they do not have influence on their behavior in the CANopen network.

8.1.1 Object 1000h - Device Type

This object gives a 32-bit code that describes the type of object and its functionality.

Table 8.1: *Object 1000h - Device Type*

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1000h	0	Device Type	UNSIGNED32	RO	No	0

This code can be divided into two parts: 16 low-order bits describing the type of profile that the device uses, and 16 high-order bits indicating a specific function according to the specified profile.

8.1.2 Object 1001h - Error Register

This object indicates whether or not an error in the device occurred. The type of error registered for the equipment follows what is described in the table 8.2.

Table 8.2: *Object 1001h - Error Register*

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1001h	0	Error register	UNSIGNED8	RO	yes	0

Table 8.3: *Structure of the object Error Register*

Bit	Meaning
0	Generic error
1	Currrent
2	Voltage
3	Temperature
4	Communication
5	Reservaded (always 0)
6	Reservaded (always 0)
7	Specific of the manufacturer

If the device presents any error, the equivalent bit must be activated. The first bit (generic error) must be activated with any error condition.

8.1.3 Object 1018h - Identity Object

It brings general information about the device.

Table 8.4: *Object 1018h - Identity Object*

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1018h	0	Number of the last sub-index	UNSIGNED8	RO	No	4
	1	Vendor ID	UNSIGNED32	RO	No	0000.0123h
	2	Product code	UNSIGNED32	RO	No	0000.1200h
	3	Revision number	UNSIGNED32	RO	No	According to the equipment firmware version
	4	Serial number	UNSIGNED32	RO	No	Different for each SSW900

The vendor ID is the number that identifies the manufacturer at the CiA. The product code is defined by the manufacturer according to the type of product. The revision number represents the equipment firmware version. The sub-index 4 is a unique serial number for each soft-starter SSW900 in CANopen network.

8.2 SERVICE DATA OBJECTS - SDOS

The SDOs are responsible for the direct access to the object dictionary of a specific device in the network. They are used for the configuration and therefore have low priority, since they do not have to be used for communicating data necessary for the device operation.

There are two types of SDOs: client and server. Basically, the communication initiates with the client (usually the master of the network) making a read (upload) or write (download) request to a server, and then this server answers the request.

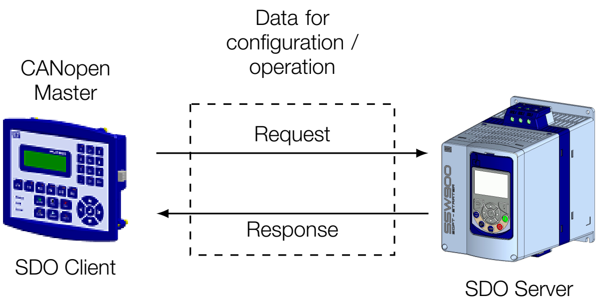


Figure 8.1: *Communication between SDO client and server*

8.2.1 Object 1200h - SDO Server

The soft-starter SSW900 soft-starter has only one SDO of the server type, which makes it possible the access to its entire object dictionary. Through it, an SDO client can configure the communication, the parameters and the the SSW900 operation. Every SDO server has an object, of the SDO_PARAMETER type, for its configuration, having the following structure:

Table 8.5: *Objet 1200h - SDO Server*

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1200h	0	Number of the last sub-index	UNSIGNED8	RO	No	2
	1	COB-ID Client - Server (rx)	UNSIGNED32	RO	No	600h + Node-ID
	2	COB-ID Server - Client (tx)	UNSIGNED32	RO	No	580h + Node-ID

8.2.2 SDOs Operation

A telegram sent by an SDO has an 8 byte size, with the following structure:

Identifier	8 data bytes							
11 bits	Command	Index		Subindex	Object data			
	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7

The identifier depends on the transmission direction (rx or tx) and on the address (or Node-ID) of the destination server. For instance, a client that makes a request to a server which Node-ID is 1, must send a message with the identifier 601h. The server will receive this message and answer with a telegram which COB-ID is equal to 581h.

The command code depends on the used function type. For the transmissions from a client to a server, the following commands can be used:

Table 8.6: *Command codes for SDO client*

Command	Function	Description	Object Data
22h	Download	Write object	Not defined
23h	Download	Write object	4 byte
2Bh	Download	Write object	2 byte
2Fh	Download	Write object	1 byte
40h	Upload	Read object	Not used
60h ou 70h	Upload segment	Segmented read	Not used

When making a request, the client will indicate through its COB-ID, the address of the slave to which this request is destined. Only one slave (using its respective SDO server) will be able to answer the received telegram to the client. The answer telegram will have also the same structure of the request telegram, the commands however are different:

Table 8.7: *Command codes for SDO server*

Command	Function	Description	Object Data
60h	Download	Write object	Not used
43h	Upload	Write object	4 byte
4Bh	Upload	Write object	2 byte
4Fh	Upload	Write object	1 byte
41h	Upload segment	Initiates segmented response for read	4 byte
01h ou 0Dh	Upload segment	Last data segment for read	8 ... 2 bytes

For readings of up to four data bytes, a single message can be transmitted by the server; for the reading of a bigger quantity of bytes, it is necessary that the client and the server exchange multiple telegrams.

A telegram is only completed after the acknowledgement of the server to the request of the client. If any error is detected during telegram exchanges (for instance, no answer from the server), the client will be able to abort the process by means of a warning message with the command code equal to 80h.

NOTE!
When the SDO is used for writing in objects that represent the SSW900 parameters (objects starting from the index 2000h), this value is saved in the nonvolatile frequency inverter memory. Therefore, the configured values are not lost after the equipment is switched off or reset. For all the other objects these values are not saved automatically, so that it is necessary to rewrite the desired values.

E.g.: A client SDO requests for a slave at address 1 the reading of the object identified by the index 2000h, sub-index 0 (zero), which represents an 16-bit integer. The master telegram has the following format:

Identifier	Command	Index		Subindex	Data			
601h	40h	00h	20h	00h	00h	00h	00h	00h

The slave responds to the request indicating that the value of the referred object is equal to 999 [1]:

Identifier	Command	Index		Subindex	Data			
581h	4Bh	00h	20h	00h	E7h	03h	00h	00h

8.3 PROCESS DATA OBJECTS - PDOS

The PDOs are used to send and receive data used during the device operation, which must often be transmitted in a fast and efficient manner. Therefore, they have a higher priority than the SDOs.

In the PDOs only data are transmitted in the telegram (index and sub-index are omitted), and in this way it is possible to do a more efficient transmission, with larger volume of data in a single telegram. However it is necessary to configure previously what is being transmitted by the PDO, so that even without the indication of the index and sub-index, it is possible to know the content of the telegram.

There are two types of PDOs, the receive PDO and the transmit PDO. The transmit PDOs are responsible for sending data to the network, whereas the receive PDOs remain responsible for receiving and handling these data. In this way it is possible to have communication among slaves of the CANopen network, it is only necessary to configure one slave to transmit information and one or more slaves to receive this information.

Figure 8.2: *Communication using PDOs*

NOTE!
PDOs can only be transmitted or received when the device is in the operational state.

8.3.1 PDO Mapping Objects

In order to be able to be transmitted by a PDO, it is necessary that an object be mapped into this PDO content. In the description of communication objects (1000h – 1FFFh), the filed "PDO Mapping" informs this possibility. Usually only information necessary for the operation of the device can be mapped, such as enabling commands, device status, reference, etc. Information on the device configuration are not accessible through PDOs, and if it is necessary to access them one must use the SDOs.

For the manufacturer's specific objects (2000h – 5FFFh), the table A.2 presents some PDO mapping objects. Read-only parameters (ro) can be used only by transmit PDOs, whereas the other parameters can be used only by receive PDOs.

[1] Do not forget that for any integer type of data, the byte transfer order is from the least significant to the most significant.

The EDS file brings the list of all objects available, informing whether the object can be mapped or not.

8.3.2 Receive PDOs

The receive PDOs, or RPDOs, are responsible for receiving data that other devices send to the CANopen network. The soft-starter SSW900 has 4 receive PDOs, each one being able to receive up to 8 bytes. Each RPDO has two parameters for its configuration, a PDO_COMM_PARAMETER and a PDO_MAPPING, as described next.

PDO_COMM_PARAMETER

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1400h até 1403h	0	Number of the last sub-index	UNSIGNED8	RO	No	4
	1	COB-ID used by the PDO	UNSIGNED32	RW	No	200h / 300h 400h / 500h + Node-ID
	2	Transmission Type	UNSIGNED8	RW	No	254

The sub-index 1 contains the receive PDO COB-ID. Every time a message is sent to the network, this object will read the COB-ID of that message and, if it is equal to the value of this field, the message will be received by the device. This field is formed by an UNSIGNED32 with the following structure:

Table 8.8: *COB-ID description*

Bit	Value	Description
31 (MSB)	0	PDO is enabled
	1	PDO is disabled
30	0	RTR permitted
29	0	Identifier size = 11 bits
28 - 11	0	Not used, always 0
10 - 0 (LSB)	X	11-bit COB-ID

The bit 31 allows enabling or disabling the PDO. The bits 29 and 30 must be kept in 0 (zero), they indicate respectively that the PDO accepts remote frames (RTR frames) and that it uses an 11-bit identifier. Since the SSW900 does not use 29-bit identifiers, the bits from 28 to 11 must be kept in 0 (zero), whereas the bits from 10 to 0 (zero) are used to configure the COB-ID for the PDO.

The sub-index 2 indicates the transmission type of this object, according to the next table.

Table 8.9: *Transmission type description*

Type of transmission	PDOs transmission				
	Cyclic	Acyclic	Synchronous	Asynchronous	RTR
0		•	•		
1 - 240	•		•		
241 - 251	Reserved				
252			•		•
253				•	•
254				•	
255				•	

- **Values 0 – 240:** any RPDO programmed in this range presents the same performance. When detecting a message, it will receive the data; however it won't update the received values until detecting the next SYNC telegram.
- **Values 252 e 253:** not allowed for receive PDOs.
- **Values 254 e 255:** they indicated that there is no relationship with the synchronization object. When receiving a message, its values are updated immediately.

PDO_MAPPING

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1600h até 1603h	0	Number of mapped objects	0 = disable 1-4=number of mapped objects	RO	No	0
	1 up to 4	1 up to 4 object mapped in the PDO	UNSIGNED32	RW	No	According EDS file

This parameter indicates the mapped objects in the SSW900 soft-starter receive PDOs. The default value of these objects is indicated in the product's EDS file. It is possible to map up to 4 different objects for each RPDO, provided that the total length does not exceed eight bytes. The mapping of an object is done indicating its index, sub-index[2] and size (in bits) in an UNSIGNED32, field with the following format:

UNSIGNED32		
Index (16 bits)	Sub-index (8 bits)	Objects size (8 bits)

For instance, analyzing the receive PDO standard mapping, we have:

- **Sub-index 0 = 2:** This PDO has two mapped objects.
- **Sub-index 1 = 22AD.0010h:** the first mapped object has an index equal to 22ADh, sub-index 0 (zero), and a size of 16 bits. This object corresponds to the parameter S5.2.5 Command Word Slot1.
- **Sub-index 2 = 22B8.0010h:** the second mapped object has an index equal to 22B8h, sub-index 0 (zero), and a size of 16 bits. This object corresponds to the parameter S5.3.2.1 Value for AO AO in 10 bits.

It is possible to modify this mapping by changing the quantity or the number of mapped objects. Remembering that only 4 objects or 8 bytes can be mapped at maximum.

NOTE!

- In order to change the mapped objects in a PDO, it is first necessary to write the value 0 (zero) in the sub-index 0 (zero). In that way the values of the sub-indexes 1 to 4 can be changed. After the desired mapping has been done, one must write again in the sub-index 0 (zero) the number of objects that have been mapped, enabling again the PDO.
- Do not forget that PDOs can only be received if the device is in the operational state.

8.3.3 Transmit PDOs

The transmit PDOs, or TPDOs, as the name says, are responsible for transmitting data for the CANopen network. The soft-starter SSW900 soft-starter has 4 transmit PDOs, each one being able to transmit up to 8 data bytes. In a manner similar to RPDOs, each TPDO has two parameters for its configuration, a PDO_COMM_PARAMETER and a PDO_MAPPING, as described next.

PDO_COMM_PARAMETER

[2] If the object is of the VAR type and does not have sub-index, the value 0 (zero) must be indicated for the sub-index.

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1800h-1803h	0	Number of the last sub-index	UNSIGNED8	RO	No	5
	1	COB-ID used by the PDO	UNSIGNED32	RW	No	180h / 280h / 380h / 480h + Node-ID
	2	Transmission Type	UNSIGNED8	RW	No	254
	3	Time between transmissions	UNSIGNED16	RW	No	-
	4	Reserved	UNSIGNED8	RW	No	-
	5	Event timer	UNSIGNED16	RW	No	0

The sub-index 1 contains the transmit PDO COB-ID. Every time this PDO sends a message to the network, the identifier of that message will be this COB-ID. The structure of this field is described in table 8.8.

The sub-index 2 indicates the transmission type of this object, which follows the table 8.9 description. Its working is however different for transmit PDOs:

- **Value 0:** indicates that the transmission must occur immediately after the reception of a SYNC telegram, but not periodically.
- **Values 1 – 240:** the PDO must be transmitted at each detected SYNC telegram (or multiple occurrences of SYNC, according to the number chosen between 1 and 240).
- **Value 252:** indicates that the message content must be updated (but not sent) after the reception of a SYNC telegram. The transmission of the message must be done after the reception of a remote frame (RTR frame).
- **Value 253:** the PDO must update and send a message as soon as it receives a remote frame.
- **Value 254:** The object must be transmitted according to the timer programmed in sub-index 5.
- **Value 255:** the object is transmitted automatically when the value of any of the objects mapped in this PDO is changed. It works by changing the state (Change of State). This type does also allow that the PDO be transmitted according to the timer programmed in sub-index 5.

In the sub-index 3 it is possible to program a minimum time (in multiples of 100 μs) that must elapse after the a telegram has been sent, so that a new one can be sent by this PDO. The value 0 (zero) disables this function.

The sub-index 5 contains a value to enable a timer for the automatic sending of a PDO. Therefore, whenever a PDO is configured as the asynchronous type, it is possible to program the value of this timer (in multiples of 1 ms), so that the PDO is transmitted periodically in the programmed time.

NOTE!

- The value of this timer must be programmed according to the used transmission rate. Very short times (close to the transmission time of the telegram) are able to monopolize the bus, causing indefinite retransmission of the PDO, and avoiding that other less priority objects transmit their data
- The minimum time allowed for this Function in the soft-starter SSW900 is 2 ms.
- It is important to observe the time between transmissions programmed in the sub-index 3, especially when the PDO is programmed with the value 255 in the sub-index 2 (Change of State).
- Do not forget that PDOs can only be received if the slave is in the operational state.

PDO_MAPPING

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1A00h-1A03h	0	Number of the last sub-index	UNSIGNED8	RO	No	0
	1 - 4	1 up to 4 object mapped in the PDO	UNSIGNED32	RW	No	0

The PDO MAPPING for the transmission works in similar way than for the reception, however in this case the data to be transmitted by the PDO are defined. Each mapped object must be put in the list according to the description showed next:

UNSIGNED32		
Index (16 bits)	Sub-index (8 bits)	Object size (8 bits)

For instance, analyzing the standard mapping of the fourth transmit PDO, we have:

- **Sub-índice 0 = 2:** This PDO has two mapped objects.
- **Sub-índice 1 = 22A8.0010h:** the first mapped object has an index equal to 22A8h, sub-index 0 (zero), and a size of 16 bits. This object corresponds to the parameter S5.1.1 Status Word SSW.
- **Sub-índice 2 = 2018.0020h:** the second mapped object has an index equal to 2018h, sub- index 0 (zero), and a size of 32 bits. This object corresponds to the parameter S1.1.4 Current Average.

It is possible to modify this mapping by changing the quantity or the number of mapped objects. Remember that a maximum of 4 objects or 8 bytes can be mapped.

NOTE!
In order to change the mapped objects in a PDO, it is first necessary to write the value 0 (zero) in the sub-index 0 (zero). In that way the values of the sub-indexes 1 to 4 can be changed. After the desired mapping has been done, one must write again in the sub-index 0 (zero) the number of objects that have been mapped, enabling again the PDO.

8.4 SYNCHRONIZATION OBJECT - SYNC

This object is transmitted with the purpose of allowing the synchronization of events among the CANopen network devices. It is transmitted by a SYNC producer, and the devices that detect its transmission are named SYNC consumers.

The soft-starter SSW900 has the function of a SYNC consumer and, therefore, it can program its PDOs to be synchronous. Synchronous PDOs are those related to the synchronization object, thus they can be programmed to be transmitted or updated based in this object.

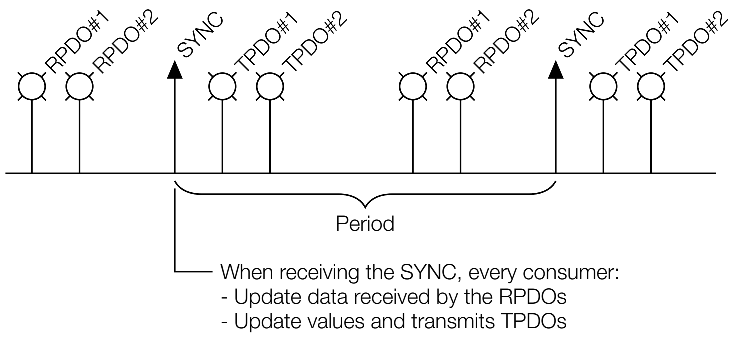


Figure 8.3: *SYNC*

The SYNC message transmitted by the producer does not have any data in its data field, because its purpose is to provide a time base for the other objects. The following object is available for the configuration of the SYNC consumer:

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1015h	0	COB-ID SYNC	UNSIGNED32	RW	No	80h

NOTE!
The period of the SYNC telegrams must be programmed in the producer according to the transmission rate and the number of synchronous PDOs to be transmitted. There must be enough time for the transmission of these objects, and it is also recommended that there is a tolerance to make it possible the transmission of asynchronous messages, such as EMCY, asynchronous PDOs and SDOs.

8.5 NETWORK MANAGEMENT - NMT

The network management object is responsible for a series of services that control the communication of the device in a CANopen network. For this object, the services of node control and error control are available (using Node Guarding or Heartbeat).

8.5.1 Slave State Control

With respect to the communication, a CANopen network device can be described by the following state machine:

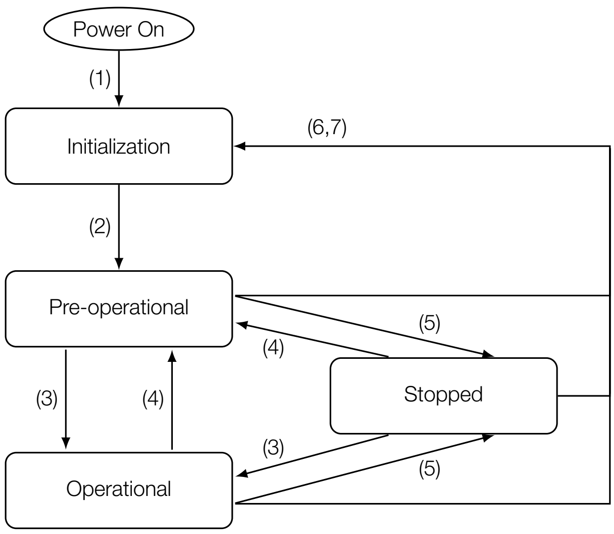


Figure 8.4: *CANopen node state diagram*

Table 8.10: *Transitions Description*

Transition	Description
1	The device is switched on and initiates the initialization (automatic)
2	Initialization concluded, it goes to the preoperational state (automatic)
3	It receives the Start Node command for entering the operational state
4	It receives the Enter Pre-Operational command, and goes to the preoperational state
5	It receives the Stop Node command for entering the stopped state
6	It receives the Reset Node command, when it executes the device complete reset
7	It receives the Reset Communication command, when it reinitializes the object values and the CANopen device communication

During the initialization the Node-ID is defined, the objects are created and the interface with the CAN network is configured. Communication with the device is not possible during this stage, which is concluded automatically. At the end of this stage the slave sends to the network a telegram of the Boot-up Object, used only to indicate that the initialization has been concluded and that the slave has entered the preoperational state. This telegram has the identifier 700h + Node-ID, and only one data byte with value equal to 0 (zero).

In the preoperational state it is already possible to communicate with the slave. But its PDOs are not yet available for operation. In the operational state all the objects are available, whereas in the stopped state only the NMT object can receive or transmit telegrams to the network. The next table shows the objects available for each state.

Table 8.11: *Objects accessible in each state*

	Initialization	Préoperational	Operational	Stopped
PDO			•	
SDO		•	•	
SYNC		•	•	
EMCY		•	•	
Boot-up	•			
NMT		•	•	•

This state machine is controlled by the network master, which sends to each slave the commands so that the desired state change be executed. These telegrams do not have confirmation, what means that the slave does only receive the telegram without returning an answer to the master. The received telegrams have the following structure:

Identifier	byte 1	byte 2
00h	Command Code	Destination Node-ID

Table 8.12: *Commands for the state transition*

Command Code	Destination Node ID
1 = START node (transition 3) 2 = STOP node (transition 4) 128 = Enter preoperational (transition 5) 129 = Reset node (transition 6) 130 = Reset comunication (transition 7)	0 = All the slaves 1 ... 127 = Specific slave

The transitions indicated in the command code correspond to the state transitions executed by the node after receiving the command (according to the figure 8.4). The Reset node command makes the slave execute a complete reset of the device, while the Reset communication command causes the device to reinitialize only the objects pertinent to the CANopen communication.

8.5.2 Error Control - Node Guarding

This service is used to make it possible the monitoring of the communication with the CANopen network, both by the master and the slave as well. In this type of service the master sends periodical telegrams to the slave, which responds to the received telegram. If some error that interrupts the communication occurs, it will be possible to identify this error, because the master as well as the slave will be notified by the Timeout in the execution of this service. The error events are called Node Guarding for the master and Life Guarding for the slave.

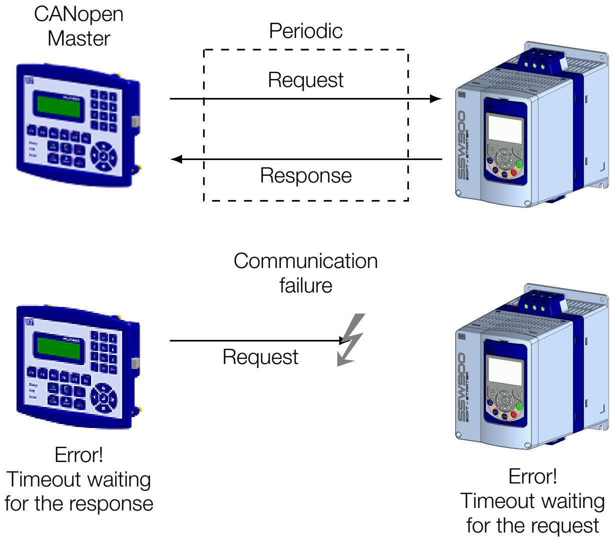


***Figure 8.5:** Error control service – Node Guarding*

There are two objects of the dictionary for the configuration of the error detection times for the Node Guarding service:

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
100Ch	0	Guard Time	UNSIGNED32	RW	No	0

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
100Dh	0	Life Time Factor	UNSIGNED8	RW	No	0

The 100Ch object allows programming the time necessary (in milliseconds) for a fault occurrence being detected, in case the slave does not receive any telegram from the master. The 100Dh object indicates how many faults in sequence are necessary until it be considered that there was really a communication error. Therefore, the multiplication of these two values will result in the total necessary time for the communication error detection using this object. The value 0 (zero) disables this function.

Once configured, the slave starts counting these times starting from the first Node Guarding telegram received from the network master. The master's telegram is of the remote type, not having data bytes. The identifier is equal to 700h + Node-ID of the destination slave. However the slave response telegram has 1 data byte with the following structure:

Identificador	byte 1	
	bit 7	bit 6 ... 0
700h + Node ID	Toogle	Estado do Escravo

This telegram has one single data byte. This byte contains, in the seven least significant bits, a value to indicate the slave state (4 = stopped, 5 = operational and 127 = preoperational), and in the eighth bit, a value that must be changed at every telegram sent by the slave (toggle bit).

If the soft-starter SSW900 detects an error using this mechanism, it will turn automatically to the preoperational state and indicate A135/F135 on its HMI.

NOTE!

- This object is active even in the stopped state (see table 8.11).
- The value 0 (zero) in any of these two objects will disable this function.
- If after the error detection the service is enabled again, then the error indication will be removed from the HMI.
- The minimum value accepted by the SSW900 soft-starter is 2 ms. But considering the transmission rate and the number of nodes in the network, the times programmed for this function must be consistent, so that there is enough time for the transmission of the telegrams and also that the rest of the communication be able to be processed.
- For any slave only one of the two services - Heartbeat or Node Guarding – can be enabled.

8.5.3 Error Control - Heartbeat

The error detection through the Heartbeat mechanism is done using two types of objects: the Heartbeat producer and the Heartbeat consumer. The producer is responsible for sending periodic telegrams to the network, simulating a heartbeat, indicating that the communication is active and without errors. One or more consumers can monitor these periodic telegrams, and if they cease occurring, it means that any communication problem occurred.

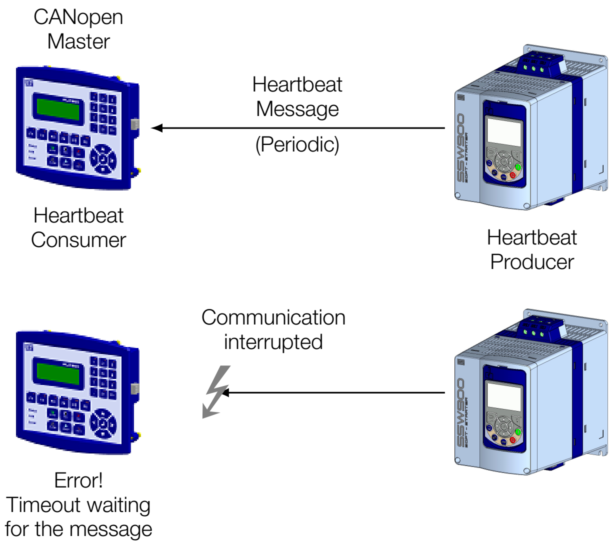


Figure 8.6: *Error control service – Heartbeat*

One device of the network can be both producer and consumer of heartbeat messages. For example, the network master can consume messages sent by a slave, making it possible to detect communication problems with the master, and simultaneously the slave can consume heartbeat messages sent by the master, also making it possible to the slave detect communication fault with the master.

The SSW900 soft-starter has the producer and consumer of heartbeat services. As a consumer, it is possible to program up to 4 different producers to be monitored by the equipment:

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1016h	0	Number of the last sub-index	UNSIGNED8	RO	No	2
	1 - 2	Consumer Heartbeat Time 1 – 2	UNSIGNED32	RW	No	0

At sub-indexes 1 to 2, it is possible to program the consumer by writing a value with the following format:

UNSIGNED32		
Reserved (8 bits)	Node-ID (8 bits)	HeartBeat time (16 bits)

- Node-ID: it allows programming the Node-ID for the heartbeat producer to be monitored.
- Heartbeat time: it allows programming the time, in 1 millisecond multiples, until the error detection if no message of the producer is received. The value 0 (zero) in this field disables the consumer.

Once configured, the heartbeat consumer initiates the monitoring after the reception of the first telegram sent by the producer. In case that an error is detected because the consumer stopped receiving messages from the heartbeat producer, it will turn automatically to the preoperational state and indicate A135/F135 on the HMI display.

As a producer, the SSW900 soft-starter has an object for the configuration of that service:

Index	Sub-index	Name	Type	Access	PDO Mapping	Value
1017h	0	Producer Heartbeat Time	UNSIGNED8	RW	No	0

The 1017h object allows programming the time in milliseconds during which the producer has to send a heartbeat telegram to the network. Once programmed, the device initiates the transmission of messages with the following format:

Identifier	byte 1	
	bit 7	bit 6 ... 0
700h + Node ID	Always 0	Slave State

NOTE!

- This object is active even in the stopped state (see table 8.11).
- The value 0 (zero) in any of these two objects will disable this function.
- If after the error detection the service is enabled again, then the error indication will be removed from the HMI display.
- The minimum value accepted by the SSW900 soft-starter is 2 ms. But considering the transmission rate and the number of nodes in the network, the times programmed for this function must be consistent, so that there is enough time for the transmission of the telegrams and also that the rest of the communication be able to be processed.
- For any slave only one of the two services - Heartbeat or Node Guarding – can be enabled.

8.6 INITIALIZATION PROCEDURE

Once the operation of the objects available for the SSW900 soft-starter is known, then it becomes necessary to program the different objects to operate combined in the network. In a general manner, the procedure for the initialization of the objects in a CANopen network follows the description of the next flowchart:

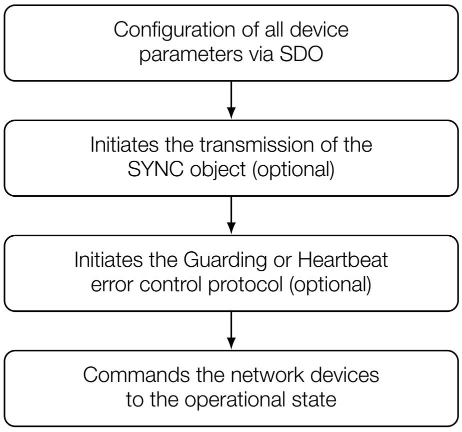


Figure 8.7: *Initialization process flowchart*

It is necessary to observe that the SSW900 soft-starter communication objects (1000h to 1FFFh) are not stored in the nonvolatile memory. Therefore, every time the equipment is reset or switched off, it is necessary to redo the communication objects parameter setting. The manufacturer specific objects (starting from 2000h that represents the parameters), they are stored in the nonvolatile memory and, thus, could be set just once.

9 STARTUP GUIDE

The main steps to start up the SSW900 soft-starter in CANopen network are described below. These steps represent an example of use. Check out the specific chapters for details on the indicated steps.

9.1 INSTALLING THE ACCESSORY

1. Install the communication accessory, as indicated in the installation guide supplied with the accessory.
2. With the module installed, during the recognition stage, the MS LED test routine will be performed. After this stage, the MS LED must turn on in green.
3. Connect the cable to the accessory, considering the recommended instructions in network installation, as described in item 3.5:
   - Use shielded cable.
   - Properly ground network equipment.
   - Avoid laying communication cables next to power cables.

9.2 CONFIGURING THE EQUIPMENT

1. Follow the recommendations described in the user manual to program the device parameters related to the motor parameterization, desired functions for the I/O signals, etc.
2. Program the command sources as desired for the application in menu C3.
3. Configure communication parameters, such as address and baudrate in C8.4.
4. Program the desired action for the equipment in case of communication fault in C8.4.5.

9.3 CONFIGURING THE MASTER

The way the network configuration is done depends greatly on the used client and the configuration tool. It is essential to know the tools used to perform this activity. In general, the following steps are necessary to perform the network configuration.

1. Load the EDS file[3] to the list of devices in the network configuration tool.
2. Select SSW900 soft-starter from the available list of devices on the network configuration tool. This can be done manually or automatically, if allowed by the tool.
3. During the configuration of the network, it is necessary to define which data will be read and written at soft-starter SSW900 by configuring the transmission and reception PDOs as described in item 8.3. Among the main parameters that can be used to control the device, we can mention:
   - S5.1.1 Status Word SSW (read)
   - S5.2.5 Command Word Slot1 (write)
   - S5.2.6 Command Word Slot2 (write)
4. Configure error control using the Node Guarding or Heartbeat services as described in item 8.5.

Once configured, the network status S5.7.6 indicates Comm. Enabled or ErrorCtrl.Enab and the node state S5.7.7 indicates Operational. It is in this condition that PDO transmission and reception effectively occurs.

[3] The EDS file is available from WEG website (http://www.weg.net). It is important to note if the EDS configuration file is compatible with the firmware version of the SSW900 soft-starter.

9.4 COMMUNICATION STATUS

Once the network is assembled and the client programmed, it is possible to use the MS LED and parameters of the equipment to identify some status related to the communication.

- The MS LED provides information about the status of the interface.
- The parameters S5.7.6 and S5.7.7 indicate the status of CANopen communication.

The master of the network must also supply information about the communication with the slave.

9.5 OPERATION USING PROCESS DATA

Once the communication is established, the data mapped in the PDOs is automatically updated. Among the main parameters that can be used to control the device, we can mention:

- S5.1.1 Status Word SSW
- S5.2.5 Command Word Slot1
- S5.2.6 Command Word Slot2

It is important to know these parameters to program the master as desired for the application.

9.6 ACCESS TO PARAMETERS – ACYCLIC MESSAGES

Besides the cyclic communication using PDOs, the CANopen protocol also defines a kind of acyclic message via SDO, used especially in asynchronous tasks, such as parameter setting and configuration of the equipment.

The EDS file provides the full parameter list of the equipment, which can be accessed via SDO. The item 7.4 describes how to address the parameters of the soft-starter SSW900 via SDO.

10 FAULTS AND ALARMS

Fault/Alarm	Description	Possible Causes
F133/A133: CAN Interface without Power Supply	It indicates that the CAN interface does not have power supply between the pins 1 and 5 of the connector. It actuates when the CAN interface is connected to the power supply and the absence of power is detected.	- Measure the voltage between the pins 1 and 5 of the CAN interface connector. - Verify if the power supply cables have not been changed or inverted. - Make sure there is no contact problem in the cable or in the CAN interface connector.
F134/A134: Bus Off	The bus off error in the CAN interface has been detected. If the number of reception or transmission errors detected by the CAN interface is too high, the CAN controller can be taken to the bus off state, where it interrupts the communication and disables the CAN interface. In order that the communication be reestablished, it will be necessary to cycle the power of the product, or remove the power supply from the CAN interface and apply it again, so that the communication be reinitiated.	- Verify if there is any short-circuit between the CAN circuit transmission cables. - Verify if the cables have not been changed or inverted. - Verify if all the network devices use the same baud rate. - Verify if termination resistors with the correct values were installed only at the extremes of the main bus. - Verify if the CAN network installation was carried out in proper manner.
F135/A135: CANopen Offline	It occurs when CANopen node state changes from operational to pre-operational.	- Verify the error control mechanisms operation (Heartbeat/Node Guarding). - Verify if the master is sending the guarding/heartbeat telegrams in the programmed time. - Verify communication problems that can cause telegram losses or transmission delays.

APPENDIX A

	Level 1		Level 2		Level 3	Page
S	Status	S1	Measurements	S1.1	Current	42
				S1.2	Main Line Voltage	
				S1.3	Output Voltage	
				S1.4	SCR Blocking Voltage	
				S1.5	Output Power & P.F.	
				S1.6	P.L.L.	
				S1.7	Motor Torque	
				S1.8	Control Voltage	
		S2	I/O	S2.1	Digital	42
				S2.2	Analog Output	
		S3	SSW900	S3.1	SSW Status	43
				S3.2	Software Version	
				S3.3	SSW Model	
				S3.4	Fan Status	
				S3.5	Accessories	
		S4	Temperatures	S4.1	SCRs Temperature	45
				S4.2	Thermal Class Status	
				S4.3	Motor Temperature	
		S5	Communications	S5.1	Status Word	45
				S5.2	Command Word	
				S5.3	Value for Outputs	
				S5.4	RS485 Serial	
				S5.5	Anybus-CC	
				S5.6	Configuration Mode	
				S5.7	CANopen/DeviceNet	
				S5.9	Bluetooth	
		S6	SoftPLC	S6.1	SoftPLC Status	49
				S6.2	Scan Cycle Time	
				S6.3	Value for Outputs	
				S6.4	Parameter	
D	Diagnostics	D1	Fault	D1.1	Actual	50
				D1.2	Fault History	
		D2	Alarms	D2.1	Actual	50
				D2.2	Alarm History	
		D3	Events			50
		D4	Motor On	D4.1	Start Current	50
				D4.2	Real Start Time	
				D4.3	Current Full Voltage	
				D4.4	Main Line Voltage	
				D4.5	Main Line Frequency	
				D4.6	kWh Counter	
				D4.7	Number Start	
		D5	Temperatures	D5.1	SCRs Maximum	51
				D5.2	Motor Maximum	
		D6	Hours Control			51
		D7	Changed Parameters			51

	Level 1		Level 2		Level 3	Page
C	Configurations	C1	Starting and Stopping			51
		C2	Nominal Motor Data			52
		C3	LOC/REM Selection			52
		C4	I/O	C4.1	Digital Inputs	53
				C4.2	Digital Outputs	
				C4.3	Analog Output	
		C5	Protections	C5.1	Voltage Protections	56
				C5.2	Current Protections	
				C5.3	Torque Protections	
				C5.4	Power Protections	
				C5.5	Phase Sequence	
				C5.6	Bypass Protections	
				C5.7	Time Protections	
				C5.8	Motor Thermal Protection	
				C5.9	Motor Thermal Class	
				C5.10	SSW Short Circuit	
				C5.11	Fault Auto-Reset	
		C6	HMI	C6.1	Password	61
				C6.2	Language	
				C6.3	Date and Time	
				C6.4	Main Screen	
				C6.5	LCD Backlight	
				C6.6	Communication Timeout	
		C7	Special Functions	C7.1	Forward/Reverse	62
				C7.2	Kick Start	
				C7.3	Jog	
				C7.4	Braking	
		C8	Communication	C8.1	I/O Data	63
				C8.2	RS485 Serial	
				C8.3	Anybus-CC	
				C8.4	CANopen/DeviceNet	
				C8.6	Bluetooth	
		C9	SSW900	C9.1	Nominal Data	67
				C9.2	Types of Connections	
				C9.3	Accessories Config.	
				C9.4	Fan Configuration	
		C10	Load / Save Parameters	C10.1	Load / Save User	68
				C10.2	Copy Function HMI	
				C10.3	Erase Diagnostics	
				C10.4	Load Factory Default	
				C10.5	Save Changed Param.	
		C11	SoftPLC	C11.3	Parameter	69
A	Assistant	A1	Oriented Start-up			70

Table A.2: *Characteristics of the parameters for the communication protocol*

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
S1 Status\Measurements						
S1.1	Current					
S1.1.1	R Phase	0.0 to 14544.0 A	1	201Ah	26	32bit
S1.1.2	S Phase	0.0 to 14544.0 A	1	201Ch	28	32bit
S1.1.3	T Phase	0.0 to 14544.0 A	1	201Eh	30	32bit
S1.1.4	Average	0.0 to 14544.0 A	1	2018h	24	32bit
S1.1.5	Motor %In	0.0 to 999.9 %	1	2002h	2	16bit
S1.1.6	SSW %In	0.0 to 999.9 %	1	2001h	1	16bit
S1.2	Main Line Voltage					
S1.2.1	R-S Line	0.0 to 999.9 V	1	2021h	33	16bit
S1.2.2	S-T Line	0.0 to 999.9 V	1	2022h	34	16bit
S1.2.3	T-R Line	0.0 to 999.9 V	1	2023h	35	16bit
S1.2.4	Average	0.0 to 999.9 V	1	2004h	4	16bit
S1.2.5	Motor %Vn	0.0 to 999.9 %	1	2003h	3	16bit
S1.2.6	SSW %Vn	0.0 to 999.9 %	1	2005h	5	16bit
S1.3	Output Voltage					
S1.3.1	Average	0.0 to 999.9 V	1	2007h	7	16bit
S1.3.2	Motor %Vn	0.0 to 999.9 %	1	2006h	6	16bit
S1.4	SCR Blocking Voltage					
S1.4.1	R-U Blocking	0.0 to 999.9 V	1	2015h	21	16bit
S1.4.2	S-V Blocking	0.0 to 999.9 V	1	2016h	22	16bit
S1.4.3	T-W Blocking	0.0 to 999.9 V	1	2017h	23	16bit
S1.5	Output Power & P.F.					
S1.5.1	Active	0.0 to 11700.0 kW	1	200Ah	10	32bit
S1.5.2	Apparent	0.0 to 11700.0 kVA	1	200Ch	12	32bit
S1.5.3	Reactive	0.0 to 11700.0 kVAr	1	200Eh	14	32bit
S1.5.4	P. F.	0.0 to 1.0	2	2008h	8	8bit
S1.6	P.L.L.					
S1.6.1	Status	0 = Off 1 = Ok		2010h	16	enum
S1.6.2	Frequency	0.0 to 99.9 Hz	1	2011h	17	16bit
S1.6.3	Sequence	0 = Invalid 1 = RST / 123 2 = RTS / 132		2012h	18	enum
S1.7	Motor Torque					
S1.7.1	Motor %Tn	0.0 to 999.9 %	1	2009h	9	16bit
S1.8	Control Voltage					
S1.8.1	Input	0.0 to 999.9 V	1	2047h	71	16bit
S1.8.2	+5V	0.0 to 9.99 V	2	2048h	72	16bit
S1.8.3	+12V	0.0 to 99.9 V	1	2049h	73	16bit
S1.8.4	+Vbat	0.0 to 9.99 V	2	204Bh	75	16bit
S1.8.5	+48V	0.0 to 99.9 V	1	204Ch	76	16bit
S2 Status\I/O						
S2.1	Digital					

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
S2.1.1	Inputs	Bit 0 = DI1 Bit 1 = DI2 Bit 2 = DI3 Bit 3 = DI4 Bit 4 = DI5 Bit 5 = DI6 Bit 6 ... 15 = Reserved		22A5h	677	16bit
S2.1.2	Outputs	Bit 0 = DO1 Bit 1 = DO2 Bit 2 = DO3 Bit 3 ... 15 = Reserved		22A6h	678	16bit
S2.2	Analog Output					
S2.2.1	Percent	0.0 to 100.0 %	2	22A1h	673	16bit
S2.2.2	Current	0.0 to 20.0 mA	3	22A2h	674	16bit
S2.2.3	Voltage	0.0 to 10.0 V	3	22A3h	675	16bit
S2.2.4	10 bits	0 to 1023	0	22A4h	676	16bit
S3 Status\SSW900						
S3.1	SSW Status					
S3.1.1	Actual	0 = Ready 1 = Initial Test 2 = Fault 3 = Ramp Up 4 = Full Voltage 5 = Bypass 6 = Reserved 7 = Ramp Down 8 = Braking 9 = FWD/REV 10 = Jog 11 = Start Delay 12 = Re-start Delay 13 = General Disabled 14 = Configuration		22A7h	679	enum
S3.1.2	Active Command Source	0 = HMI Keys LOC 1 = HMI Keys REM 2 = DIx LOC 3 = DIx REM 4 = USB LOC 5 = USB REM 6 = SoftPLC LOC 7 = SoftPLC REM 8 = Slot 1 LOC 9 = Slot 1 REM 10 = Slot 2 LOC 11 = Slot 2 REM		20E8h	232	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
S3.1.3	Status Word					
S3.1.3.1	SSW	Bit 0 = Running Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = Initial Test Bit 4 = Ramp Up Bit 5 = Full Voltage Bit 6 = Bypass Bit 7 = Ramp Down Bit 8 = Remote Bit 9 = Braking Bit 10 = FWD/REV Bit 11 = Reverse Bit 12 = Ton Bit 13 = Toff Bit 14 = Alarm Bit 15 = Fault		22A8h	680	16bit
S3.1.4	Configuration Mode					
S3.1.4.1	Status	Bit 0 = System Initialization Bit 1 = Firmware Download Bit 2 = Oriented Start-Up Bit 3 = Incompatible Bit 4 = Reset Needs Bit 5 = Copy HMI Bit 6 ... 15 = Reserved		22B4h	692	16bit
S3.2	Software Version					
S3.2.1	Package	0.0 to 99.99	2	2148h	328	16bit
S3.2.2	Details					
S3.2.2.1	Control 1 V	0.0 to 99.99	2	214Ah	330	16bit
S3.2.2.2	Control 1 rev.	-32768 to 32767	0	2147h	327	s16bit
S3.2.2.3	Bootloader V	0.0 to 99.99	2	2149h	329	16bit
S3.2.2.4	Bootloader rev.	-32768 to 32767	0	2143h	323	s16bit
S3.2.2.5	HMI rev.	-32768 to 32767	0	2142h	322	s16bit
S3.2.2.6	Control 2 V	0.0 to 99.99	2	214Bh	331	16bit
S3.2.2.7	Control 2 rev.	-32768 to 32767	0	2146h	326	s16bit
S3.2.2.8	Accessory 1 V	0.0 to 99.99	2	214Dh	333	16bit
S3.2.2.9	Accessory 1 rev.	-32768 to 32767	0	2144h	324	s16bit
S3.2.2.10	Accessory 2 V	0.0 to 99.99	2	214Eh	334	16bit
S3.2.2.11	Accessory 2 rev.	-32768 to 32767	0	2145h	325	s16bit
S3.3	SSW Model					
S3.3.1	Current	0 = 10 to 30 A 1 = 45 to 105 A 2 = 130 to 200 A 3 = 255 to 412 A 4 = 480 to 670 A 5 = 820 to 950 A		2126h	294	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		6 = 1100 to 1400 A				
S3.3.2	Voltage	0 = 220 to 575 V 1 = 400 to 690 V		2128h	296	enum
S3.3.3	Control Voltage	0 = 110 to 240 V 1 = 110 to 130 V 2 = 220 to 240 V 3 = 24 Vcc		2129h	297	enum
S3.3.4	Serial Number	0 to 4294967295	0	212Ah	298	32bit
S3.4	Fan Status					
S3.4.1	Actual	0 = Off 1 = On		2125h	293	enum
S3.5	Accessories					
S3.5.1	Slot 1	0 = Without 1 = Anybus-CC 2 = RS-485 3 = PT100 4 = I/Os Exp. 5 = Profibus 6 = CAN 7 = Ethernet 8 = External Current Acqu.		214Fh	335	enum
S3.5.2	Slot 2	0 = Without 1 = Anybus-CC 2 = RS-485 3 = PT100 4 = I/Os Exp. 5 = Profibus 6 = CAN 7 = Ethernet 8 = External Current Acqu.		2150h	336	enum
		S4 Status\Temperatures				
S4.1	SCRs Temperature					
S4.1.1	Actual	-22 to 260 $^\circ C$	0	203Ch	60	s16bit
S4.2	Thermal Class Status					
S4.2.1	Of Maximum	0.0 to 100.0 %	1	2032h	50	16bit
S4.3	Motor Temperature					
S4.3.1	Channel 1	-20 to 260 $^\circ C$	0	203Fh	63	s16bit
S4.3.2	Channel 2	-20 to 260 $^\circ C$	0	2040h	64	s16bit
S4.3.3	Channel 3	-20 to 260 $^\circ C$	0	2041h	65	s16bit
S4.3.4	Channel 4	-20 to 260 $^\circ C$	0	2042h	66	s16bit
S4.3.5	Channel 5	-20 to 260 $^\circ C$	0	2043h	67	s16bit
S4.3.6	Channel 6	-20 to 260 $^\circ C$	0	2044h	68	s16bit
		S5 Status\Communications				

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
S5.1	Status Word					
S5.1.1	SSW	Bit 0 = Running Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = Initial Test Bit 4 = Ramp Up Bit 5 = Full Voltage Bit 6 = Bypass Bit 7 = Ramp Down Bit 8 = Remote Bit 9 = Braking Bit 10 = FWD/REV Bit 11 = Reverse Bit 12 = Ton Bit 13 = Toff Bit 14 = Alarm Bit 15 = Fault		22A8h	680	16bit
S5.2	Command Word					
S5.2.1	DIx	Bit 0 = Start/Stop Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = FWD/REV Bit 4 = LOC/REM Bit 5 ... 6 = Reserved Bit 7 = Reset Bit 8 = Brake Bit 9 = Emergency Start Bit 10 ... 15 = Reserved		22ABh	683	16bit
S5.2.2	HMI Key	Bit 0 = Start/Stop Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = FWD/REV Bit 4 = LOC/REM Bit 5 ... 6 = Reserved Bit 7 = Reset Bit 8 ... 15 = Reserved		22A9h	681	16bit
S5.2.3	USB	Bit 0 = Start/Stop Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = FWD/REV Bit 4 = LOC/REM Bit 5 ... 6 = Reserved Bit 7 = Reset Bit 8 ... 15 = Reserved		22AAh	682	16bit
S5.2.4	SoftPLC	Bit 0 = Start/Stop		22ACh	684	16bit

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = FWD/REV Bit 4 = LOC/REM Bit 5 ... 6 = Reserved Bit 7 = Reset Bit 8 ... 15 = Reserved				
S5.2.5	Slot1	Bit 0 = Start/Stop Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = FWD/REV Bit 4 = LOC/REM Bit 5 ... 6 = Reserved Bit 7 = Reset Bit 8 ... 15 = Reserved		22ADh	685	16bit
S5.2.6	Slot2	Bit 0 = Start/Stop Bit 1 = Gener. Enabled Bit 2 = JOG Bit 3 = FWD/REV Bit 4 = LOC/REM Bit 5 ... 6 = Reserved Bit 7 = Reset Bit 8 ... 15 = Reserved		22AEh	686	16bit
S5.3	Value for Outputs					
S5.3.1	DO Value	Bit 0 = DO1 Bit 1 = DO2 Bit 2 = DO3 Bit 3 ... 15 = Reserved		22B7h	695	16bit
S5.3.2	Value for AO					
S5.3.2.1	AO in 10 bits	0 to 1023	0	22B8h	696	16bit
S5.4	RS485 Serial					
S5.4.1	Interface Status	0 = Off 1 = On 2 = Timeout Error		22DFh	735	enum
S5.4.2	Received Telegram	0 to 65535	0	22E0h	736	16bit
S5.4.3	Transmitted Telegram	0 to 65535	0	22E1h	737	16bit
S5.4.4	Telegram with Error	0 to 65535	0	22E2h	738	16bit
S5.4.5	Reception Errors	0 to 65535	0	22E3h	739	16bit
S5.5	Anybus-CC					
S5.5.1	Identification	0 = Disabled 1 ... 15 = Reserved 16 = Profibus DP 17 = DeviceNet 18 = Reserved		22EEh	750	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		19 = EtherNet/IP 20 = Reserved 21 = Modbus TCP 22 = Reserved 23 = PROFINET IO 24 ... 25 = Reserved				
S5.5.2	Comm. Status	0 = Setup 1 = Init 2 = Wait Comm 3 = Idle 4 = Data Active 5 = Error 6 = Reserved 7 = Exception 8 = Access Error		22EFh	751	enum
S5.6	Configuration Mode					
S5.6.1	Status	Bit 0 = System Initialization Bit 1 = Firmware Download Bit 2 = Oriented Start-Up Bit 3 = Incompatible Bit 4 = Reset Needs Bit 5 = Copy HMI Bit 6 ... 15 = Reserved		22B4h	692	16bit
S5.6.2	Control	Bit 0 = Abort Startup Bit 1 ... 15 = Reserved		22B5h	693	16bit
S5.7	CANopen/DeviceNet					
S5.7.1	CAN Controller Status	0 = Disabled 1 = Auto-baud 2 = CAN Enabled 3 = Warning 4 = Error Passive 5 = Bus Off 6 = No Bus Power		22C1h	705	enum
S5.7.2	Received Telegram	0 to 65535	0	22C2h	706	16bit
S5.7.3	Transmitted Telegram	0 to 65535	0	22C3h	707	16bit
S5.7.4	Bus Off Counter	0 to 65535	0	22C4h	708	16bit
S5.7.5	Lost Messages	0 to 65535	0	22C5h	709	16bit
S5.7.6	CANopen Comm. Status	0 = Disabled 1 = Reserved 2 = Comm. Enabled 3 = ErrorCtrl.Enab 4 = Guarding Error 5 = HeartbeatError		22D1h	721	enum
S5.7.7	CANopen Node State	0 = Disabled		22D2h	722	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		1 = Initialization 2 = Stopped 3 = Operational 4 = PreOperational				
S5.7.8	DNet Network Status	0 = Offline 1 = OnLine,NotConn 2 = OnLine,Conn 3 = Conn.Timed-out 4 = Link Failure 5 = Auto-Baud		22CCh	716	enum
S5.7.9	DeviceNet Master Status	0 = Run 1 = Idle		22CDh	717	enum
S5.9	Bluetooth					
S6 Status\SoftPLC						
S6.1	SoftPLC Status					
S6.1.1	Actual	0 = No Application 1 = Install. App. 2 = Incompat. App. 3 = App. Stopped 4 = App. Running		244Ch	1100	enum
S6.2	Scan Cycle Time					
S6.2.1	Actual	0 to 65535 ms	0	244Eh	1102	16bit
S6.3	Value for Outputs					
S6.3.1	DO Value	Bit 0 = DO1 Bit 1 = DO2 Bit 2 = DO3 Bit 3 ... 15 = Reserved		22B9h	697	16bit
S6.3.2	AO Value					
S6.3.2.1	AO in 10 bits	0 to 1023	0	22BAh	698	16bit
S6.4	Parameter					
S6.4.1	User #1	-10000 to 10000	0	2456h	1110	s32bit
S6.4.2	User #2	-10000 to 10000	0	2458h	1112	s32bit
S6.4.3	User #3	-10000 to 10000	0	245Ah	1114	s32bit
S6.4.4	User #4	-10000 to 10000	0	245Ch	1116	s32bit
S6.4.5	User #5	-10000 to 10000	0	245Eh	1118	s32bit
S6.4.6	User #6	-10000 to 10000	0	2460h	1120	s32bit
S6.4.7	User #7	-10000 to 10000	0	2462h	1122	s32bit
S6.4.8	User #8	-10000 to 10000	0	2464h	1124	s32bit
S6.4.9	User #9	-10000 to 10000	0	2466h	1126	s32bit
S6.4.10	User #10	-10000 to 10000	0	2468h	1128	s32bit
S6.4.11	User #11	-10000 to 10000	0	246Ah	1130	s32bit
S6.4.12	User #12	-10000 to 10000	0	246Ch	1132	s32bit
S6.4.13	User #13	-10000 to 10000	0	246Eh	1134	s32bit
S6.4.14	User #14	-10000 to 10000	0	2470h	1136	s32bit
S6.4.15	User #15	-10000 to 10000	0	2472h	1138	s32bit

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
S6.4.16	User #16	-10000 to 10000	0	2474h	1140	s32bit
S6.4.17	User #17	-10000 to 10000	0	2476h	1142	s32bit
S6.4.18	User #18	-10000 to 10000	0	2478h	1144	s32bit
S6.4.19	User #19	-10000 to 10000	0	247Ah	1146	s32bit
S6.4.20	User #20	-10000 to 10000	0	247Ch	1148	s32bit
S6.4.21	User #21	-10000 to 10000	0	247Eh	1150	s32bit
S6.4.22	User #22	-10000 to 10000	0	2480h	1152	s32bit
S6.4.23	User #23	-10000 to 10000	0	2482h	1154	s32bit
S6.4.24	User #24	-10000 to 10000	0	2484h	1156	s32bit
S6.4.25	User #25	-10000 to 10000	0	2486h	1158	s32bit
S6.4.26	User #26	-10000 to 10000	0	2488h	1160	s32bit
S6.4.27	User #27	-10000 to 10000	0	248Ah	1162	s32bit
S6.4.28	User #28	-10000 to 10000	0	248Ch	1164	s32bit
S6.4.29	User #29	-10000 to 10000	0	248Eh	1166	s32bit
S6.4.30	User #30	-10000 to 10000	0	2490h	1168	s32bit
S6.4.31	User #31	-10000 to 10000	0	2492h	1170	s32bit
S6.4.32	User #32	-10000 to 10000	0	2494h	1172	s32bit
S6.4.33	User #33	-10000 to 10000	0	2496h	1174	s32bit
S6.4.34	User #34	-10000 to 10000	0	2498h	1176	s32bit
S6.4.35	User #35	-10000 to 10000	0	249Ah	1178	s32bit
S6.4.36	User #36	-10000 to 10000	0	249Ch	1180	s32bit
S6.4.37	User #37	-10000 to 10000	0	249Eh	1182	s32bit
S6.4.38	User #38	-10000 to 10000	0	24A0h	1184	s32bit
S6.4.39	User #39	-10000 to 10000	0	24A2h	1186	s32bit
S6.4.40	User #40	-10000 to 10000	0	24A4h	1188	s32bit
S6.4.41	User #41	-10000 to 10000	0	24A6h	1190	s32bit
S6.4.42	User #42	-10000 to 10000	0	24A8h	1192	s32bit
S6.4.43	User #43	-10000 to 10000	0	24AAh	1194	s32bit
S6.4.44	User #44	-10000 to 10000	0	24ACh	1196	s32bit
S6.4.45	User #45	-10000 to 10000	0	24AEh	1198	s32bit
S6.4.46	User #46	-10000 to 10000	0	24B0h	1200	s32bit
S6.4.47	User #47	-10000 to 10000	0	24B2h	1202	s32bit
S6.4.48	User #48	-10000 to 10000	0	24B4h	1204	s32bit
S6.4.49	User #49	-10000 to 10000	0	24B6h	1206	s32bit
S6.4.50	User #50	-10000 to 10000	0	24B8h	1208	s32bit
D1 Diagnostics\Fault						
D1.1	Actual					
D1.1.1	Fxxx	0 to 999	0	205Ah	90	16bit
D1.2	Fault History					
D2 Diagnostics\Alarms						
D2.1	Actual					
D2.1.1	Axxx 1	0 to 999	0	205Bh	91	16bit
D2.1.2	Axxx 2	0 to 999	0	205Ch	92	16bit
D2.1.3	Axxx 3	0 to 999	0	205Dh	93	16bit
D2.1.4	Axxx 4	0 to 999	0	205Eh	94	16bit
D2.1.5	Axxx 5	0 to 999	0	205Fh	95	16bit
D2.2	Alarm History					
D3 Diagnostics\Events						
D4 Diagnostics\Motor On						

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
D4.1	Start Current					
D4.1.1	Maximum	0.0 to 14544.0 A	1	2024h	36	32bit
D4.1.2	Average	0.0 to 14544.0 A	1	2026h	38	32bit
D4.2	Real Start Time					
D4.2.1	Actual	0 to 999 s	0	2030h	48	16bit
D4.2.2	Final	0 to 999 s	0	2031h	49	16bit
D4.3	Current Full Voltage					
D4.3.1	Maximum	0.0 to 14544.0 A	1	2028h	40	32bit
D4.4	Main Line Voltage					
D4.4.1	Maximum	0.0 to 999.9 V	1	2036h	54	16bit
D4.4.2	Minimun	0.0 to 999.9 V	1	2037h	55	16bit
D4.5	Main Line Frequency					
D4.5.1	Maximum	0.0 to 99.9 Hz	1	2038h	56	16bit
D4.5.2	Minimum	0.0 to 99.9 Hz	1	2039h	57	16bit
D4.6	kWh Counter					
D4.6.1	Total	0.0 to 214748364.7 kWh	1	2034h	52	32bit
D4.7	Number Start					
D4.7.1	Total	0 to 65535	0	203Bh	59	16bit
D5 Diagnostics\Temperatures						
D5.1	SCRs Maximum					
D5.1.1	Total	-22 to 260 $°C$	0	204Dh	77	s16bit
D5.2	Motor Maximum					
D5.2.1	Channel 1	-20 to 260 $°C$	0	2050h	80	s16bit
D5.2.2	Channel 2	-20 to 260 $°C$	0	2051h	81	s16bit
D5.2.3	Channel 3	-20 to 260 $°C$	0	2052h	82	s16bit
D5.2.4	Channel 4	-20 to 260 $°C$	0	2053h	83	s16bit
D5.2.5	Channel 5	-20 to 260 $°C$	0	2054h	84	s16bit
D5.2.6	Channel 6	-20 to 260 $°C$	0	2055h	85	s16bit
D6 Diagnostics\Hours Control						
D6.1	Powered	0 to 4294967295 s	0	202Ah	42	TIME
D6.2	Enabled	0 to 4294967295 s	0	202Ch	44	TIME
D6.3	Fan ON	0 to 4294967295 s	0	202Eh	46	TIME
D7 Diagnostics\Changed Parameters						
C1 Configurations\Starting and Stopping						
C1.1	Types of Control	0 = Voltage Ramp 1 = Voltage Ramp + Current Limit 2 = Current Limit 3 = Current Ramp 4 = Pump Control 5 = Torque Control 6 = D.O.L. SCR		20CAh	202	enum
C1.2	Initial Start Voltage	25 to 90 %	0	2065h	101	8bit
C1.3	Maximum Start Time	1 to 999 s	0	2066h	102	16bit
C1.4	Start End Detection	0 = Time 1 = Automatic		206Ah	106	enum
C1.5	Initial Current Ramp	150 to 500 %	0	206Fh	111	16bit

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C1.6	Current Ramp Time	1 to 99 %	0	2070h	112	8bit
C1.7	Current Limit	150 to 500 %	0	206Eh	110	16bit
C1.8	Start Torque Chara.	1 = Constant 2 = Linear 3 = Square		2078h	120	enum
C1.9	Initial Start Torque	10 to 300 %	0	2079h	121	16bit
C1.10	End Start Torque	10 to 300 %	0	207Ah	122	16bit
C1.11	Minimun Start Torque	10 to 300 %	0	207Bh	123	16bit
C1.12	Min.Start Torq. Time	1 to 99 %	0	207Ch	124	8bit
C1.13	Stop Time	0 to 999 s	0	2068h	104	16bit
C1.14	Step Down Volt. Stop	60 to 100 %	0	2067h	103	8bit
C1.15	End Voltage Stop	30 to 55 %	0	2069h	105	8bit
C1.16	Stop Torque Characte.	1 = Constant 2 = Linear 3 = Square		207Dh	125	enum
C1.17	End Stop Torque	10 to 100 %	0	207Eh	126	8bit
C1.18	Minimum Stop Torque	10 to 100 %	0	207Fh	127	8bit
C1.19	Min. Stop Torque Time	1 to 99 %	0	2080h	128	8bit
C2 Configurations\Nominal Motor Data						
C2.1	Voltage	1 to 999 V	0	2190h	400	16bit
C2.2	Current	0.1 to 2424.0 A	1	2191h	401	16bit
C2.3	Speed	1 to 3600 rpm	0	2192h	402	16bit
C2.4	Power	0.1 to 1950.0 kW	1	2194h	404	16bit
C2.5	P.F. Power Factor	0.01 to 1.0	2	2195h	405	8bit
C2.6	S.F. Service Factor	0.01 to 1.5	2	2196h	406	8bit
C3 Configurations\LOC/REM Selection						
C3.1	Mode	0 = Always LOC 1 = Always REM 2 = HMI LR Key LOC 3 = HMI LR Key REM 4 = DIx 5 = USB LOC 6 = USB REM 7 = SoftPLC LOC 8 = SoftPLC REM 9 = Slot 1 LOC 10 = Slot 1 REM 11 = Slot 2 LOC 12 = Slot 2 REM		20DCh	220	enum
C3.2	LOC Command	0 = HMI Keys 1 = DIx 2 = USB 3 = SoftPLC 4 = Slot 1 5 = Slot 2		20E5h	229	enum
C3.3	REM Command			20E6h	230	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		0 = HMI Keys 1 = DIx 2 = USB 3 = SoftPLC 4 = Slot 1 5 = Slot 2				
C3.4	Commands Copy	0 = No 1 = Yes		20E7h	231	enum
		C4 Configurations\I/O				
C4.1	Digital Inputs					
C4.1.1	DI1	0 = Not Used 1 = Start / Stop 2 = Start (3 Wires) 3 = Stop (3 Wires) 4 = General Enable 5 = LOC / REM 6 = JOG 7 = FWD / REV 8 = No External Fault 9 = No External Alarm 10 = Brake 11 = Reset 12 = Load User 1/2 13 ... 16 = Reserved		2107h	263	enum
C4.1.2	DI2	0 = Not Used 1 = Start / Stop 2 = Start (3 Wires) 3 = Stop (3 Wires) 4 = General Enable 5 = LOC / REM 6 = JOG 7 = FWD / REV 8 = No External Fault 9 = No External Alarm 10 = Brake 11 = Reset 12 = Load User 1/2 13 ... 16 = Reserved		2108h	264	enum
C4.1.3	DI3	0 = Not Used 1 = Start / Stop 2 = Start (3 Wires) 3 = Stop (3 Wires) 4 = General Enable 5 = LOC / REM 6 = JOG 7 = FWD / REV		2109h	265	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		8 = No External Fault 9 = No External Alarm 10 = Brake 11 = Reset 12 = Load User 1/2 13 = Reserved 14 = Emergency Start 15 ... 16 = Reserved				
C4.1.4	DI4	0 = Not Used 1 = Start / Stop 2 = Start (3 Wires) 3 = Stop (3 Wires) 4 = General Enable 5 = LOC / REM 6 = JOG 7 = FWD / REV 8 = No External Fault 9 = No External Alarm 10 = Brake 11 = Reset 12 = Load User 1/2 13 ... 16 = Reserved		210Ah	266	enum
C4.1.5	DI5	0 = Not Used 1 = Start / Stop 2 = Start (3 Wires) 3 = Stop (3 Wires) 4 = General Enable 5 = LOC / REM 6 = JOG 7 = FWD / REV 8 = No External Fault 9 = No External Alarm 10 = Brake 11 = Reset 12 = Load User 1/2 13 ... 16 = Reserved		210Bh	267	enum
C4.1.6	DI6	0 = Not Used 1 = Start / Stop 2 = Start (3 Wires) 3 = Stop (3 Wires) 4 = General Enable 5 = LOC / REM 6 = JOG 7 = FWD / REV 8 = No External Fault 9 = No External Alarm 10 = Brake 11 = Reset		210Ch	268	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		12 = Load User 1/2 13 ... 14 = Reserved 15 = Mot. Thermistor A032 16 = Mot. Thermistor F032				
C4.2	Digital Outputs					
C4.2.1	DO1	0 = Not Used 1 = Running 2 = Full Voltage 3 = Bypass 4 = FWD / REV K1 5 = DC Braking 6 = Without Fault 7 = With Fault 8 = Without Alarm 9 = With Alarm 10 = No Fault / Alarm 11 = SoftPLC 12 = Communication 13 = I motor % > Value 14 = Breaker Shunt Trip		2113h	275	enum
C4.2.2	DO2	0 = Not Used 1 = Running 2 = Full Voltage 3 = Bypass 4 = FWD / REV K2 5 = DC Braking 6 = Without Fault 7 = With Fault 8 = Without Alarm 9 = With Alarm 10 = No Fault / Alarm 11 = SoftPLC 12 = Communication 13 = I motor % > Value 14 = Breaker Shunt Trip		2114h	276	enum
C4.2.3	DO3	0 = Not Used 1 = Running 2 = Full Voltage 3 = Bypass 4 = Not Used 5 = DC Braking 6 = Without Fault 7 = With Fault 8 = Without Alarm 9 = With Alarm 10 = No Fault / Alarm 11 = SoftPLC		2115h	277	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		12 = Communication 13 = I motor % > Value 14 = Breaker Shunt Trip				
C4.2.4	DO Comparison Value	10.0 to 500.0 %	1	2116h	278	16bit
C4.3	Analog Output					
C4.3.1	Function	0 = Not Used 1 = SSW Current % 2 = Line Voltage % 3 = Output Voltage % 4 = Power Factor 5 = Thermal Class Prot. 6 = Output Power W 7 = Output Power VA 8 = Motor Torque % 9 = Value to AO 10 = SCRs Temperature 11 = SoftPLC		20FBh	251	enum
C4.3.2	Gain	0.0 to 9.999	3	20FCh	252	16bit
C4.3.3	Signal	0 = 0 to 20mA 1 = 4 to 20mA 2 = 20mA to 0 3 = 20 to 4mA 4 = 0 to 10V 5 = 10V to 0		20FDh	253	enum
C5 Configurations\Protections						
C5.1	Voltage Protections					
C5.1.1	Motor Undervoltage					
C5.1.1.1	Mode	0 = Inactive 1 = Fault F002 2 = Alarm A002		2384h	900	enum
C5.1.1.2	Level	0 to 30 %Vn	0	2385h	901	8bit
C5.1.1.3	Time	0.1 to 10.0 s	1	2386h	902	8bit
C5.1.2	Motor Overvoltage					
C5.1.2.1	Mode	0 = Inactive 1 = Fault F016 2 = Alarm A016		2387h	903	enum
C5.1.2.2	Level	0 to 20 %Vn	0	2388h	904	8bit
C5.1.2.3	Time	0.1 to 10.0 s	1	2389h	905	8bit
C5.1.3	Motor Voltage Imbalance					
C5.1.3.1	Mode	0 = Inactive 1 = Fault F001 2 = Alarm A001		238Ah	906	enum
C5.1.3.2	Level	0 to 30 %Vn	0	238Bh	907	8bit
C5.1.3.3	Time	0.1 to 10.0 s	1	238Ch	908	8bit

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C5.2	Current Protections					
C5.2.1	Motor Undercurrent					
C5.2.1.1	Mode	0 = Inactive 1 = Fault F065 2 = Alarm A065		238Eh	910	enum
C5.2.1.2	Level	0 to 99 %In	0	238Fh	911	8bit
C5.2.1.3	Time	1 to 99 s	0	2390h	912	8bit
C5.2.2	Motor Overcurrent					
C5.2.2.1	Mode	0 = Inactive 1 = Fault F066 2 = Alarm A066		2391h	913	enum
C5.2.2.2	Level	0 to 99 %In	0	2392h	914	8bit
C5.2.2.3	Time	1 to 99 s	0	2393h	915	8bit
C5.2.3	Current Imbalance					
C5.2.3.1	Mode	0 = Inactive 1 = Fault F074 2 = Alarm A074		2394h	916	enum
C5.2.3.2	Level	0 to 30 %In	0	2395h	917	8bit
C5.2.3.3	Time	1 to 99 s	0	2396h	918	8bit
C5.3	Torque Protections					
C5.3.1	Undertorque					
C5.3.1.1	Mode	0 = Inactive 1 = Fault F078 2 = Alarm A078		23B6h	950	enum
C5.3.1.2	Level	0 to 99 %Tn	0	23B7h	951	8bit
C5.3.1.3	Time	1 to 99 s	0	23B8h	952	8bit
C5.3.2	Overtorque					
C5.3.2.1	Mode	0 = Inactive 1 = Fault F079 2 = Alarm A079		23B9h	953	enum
C5.3.2.2	Level	0 to 99 %Tn	0	23BAh	954	8bit
C5.3.2.3	Time	1 to 99 s	0	23BBh	955	8bit
C5.4	Power Protections					
C5.4.1	Underpower					
C5.4.1.1	Mode	0 = Inactive 1 = Fault F080 2 = Alarm A080		23C0h	960	enum
C5.4.1.2	Level	0 to 99 %Pn	0	23C1h	961	8bit
C5.4.1.3	Time	1 to 99 s	0	23C2h	962	8bit
C5.4.2	Overpower					
C5.4.2.1	Mode			23C3h	963	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		0 = Inactive 1 = Fault F081 2 = Alarm A081				
C5.4.2.2	Level	0 to 99 %Pn	0	23C4h	964	8bit
C5.4.2.3	Time	1 to 99 s	0	23C5h	965	8bit
C5.5	Phase Sequence					
C5.5.1	Mode	0 = Inactive 1 = RST - Fault F067 2 = RTS - Fault F068		23A2h	930	enum
C5.6	Bypass Protections					
C5.6.1	Undercurrent	0 = Inactive 1 = Fault F076		2397h	919	enum
C5.6.2	Overcurrent	0 = Inactive 1 = Fault F063		2398h	920	enum
C5.6.3	Closed	0 = Inactive 1 = Fault F077		2399h	921	enum
C5.7	Time Protections					
C5.7.1	Before Start	0.5 to 999.9 s	1	23A3h	931	16bit
C5.7.2	After Stop	2.0 to 999.9 s	1	23A4h	932	16bit
C5.7.3	Between Start	2 to 9999 s	0	23A5h	933	16bit
C5.8	Motor Thermal Protection					
C5.8.1	Ch1 Installed Sensor					
C5.8.1.1	Mode	0 = Off 1 = On 2 = On Stator		23EEh	1006	enum
C5.8.2	Ch1 Sensor Fault					
C5.8.2.1	Mode	0 = Fault F109 and F117 1 = Alarm A109 and A117		23E6h	998	enum
C5.8.3	Ch1 Overtemperature					
C5.8.3.1	Mode	0 = Fault F101 1 = Alarm A101 2 = F101 and A101		23C6h	966	enum
C5.8.3.2	Fault Level	0 to 250 °C	0	23C7h	967	8bit
C5.8.3.3	Alarm Level	0 to 250 °C	0	23C8h	968	8bit
C5.8.3.4	Alarm Reset	0 to 250 °C	0	23C9h	969	8bit
C5.8.4	Ch2 Installed Sensor					
C5.8.4.1	Mode	0 = Off 1 = On 2 = On Stator		23EFh	1007	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C5.8.5	Ch2 Sensor Fault					
C5.8.5.1	Mode	0 = Fault F110 and F118 1 = Alarm A110 and A118		23E7h	999	enum
C5.8.6	Ch2 Overtemperature					
C5.8.6.1	Mode	0 = Fault F102 1 = Alarm A102 2 = F102 and A102		23CAh	970	enum
C5.8.6.2	Fault Level	0 to 250 $°C$	0	23CBh	971	8bit
C5.8.6.3	Alarm Level	0 to 250 $°C$	0	23CCh	972	8bit
C5.8.6.4	Alarm Reset	0 to 250 $°C$	0	23CDh	973	8bit
C5.8.7	Ch3 Installed Sensor					
C5.8.7.1	Mode	0 = Off 1 = On 2 = On Stator		23F0h	1008	enum
C5.8.8	Ch3 Sensor Fault					
C5.8.8.1	Mode	0 = Fault F111 and F119 1 = Alarm A111 and A119		23E8h	1000	enum
C5.8.9	Ch3 Overtemperature					
C5.8.9.1	Mode	0 = Fault F103 1 = Alarm A103 2 = F103 and A103		23CEh	974	enum
C5.8.9.2	Fault Level	0 to 250 $°C$	0	23CFh	975	8bit
C5.8.9.3	Alarm Level	0 to 250 $°C$	0	23D0h	976	8bit
C5.8.9.4	Alarm Reset	0 to 250 $°C$	0	23D1h	977	8bit
C5.8.10	Ch4 Installed Sensor					
C5.8.10.1	Mode	0 = Off 1 = On 2 = On Stator		23F1h	1009	enum
C5.8.11	Ch4 Sensor Fault					
C5.8.11.1	Mode	0 = Fault F112 and F120 1 = Alarm A112 and A120		23E9h	1001	enum
C5.8.12	Ch4 Overtemperature					
C5.8.12.1	Mode	0 = Fault F104 1 = Alarm A104 2 = F104 and A104		23D2h	978	enum
C5.8.12.2	Fault Level	0 to 250 $°C$	0	23D3h	979	8bit
C5.8.12.3	Alarm Level	0 to 250 $°C$	0	23D4h	980	8bit
C5.8.12.4	Alarm Reset	0 to 250 $°C$	0	23D5h	981	8bit
C5.8.13	Ch5 Installed Sensor					

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C5.8.13.1	Mode	0 = Off 1 = On 2 = On Stator		23F2h	1010	enum
C5.8.14	Ch5 Sensor Fault					
C5.8.14.1	Mode	0 = Fault F113 and F121 1 = Alarm A113 and A121		23EAh	1002	enum
C5.8.15	Ch5 Overtemperature					
C5.8.15.1	Mode	0 = Fault F105 1 = Alarm A105 2 = F105 and A105		23D6h	982	enum
C5.8.15.2	Fault Level	0 to 250 $°C$	0	23D7h	983	8bit
C5.8.15.3	Alarm Level	0 to 250 $°C$	0	23D8h	984	8bit
C5.8.15.4	Alarm Reset	0 to 250 $°C$	0	23D9h	985	8bit
C5.8.16	Ch6 Installed Sensor					
C5.8.16.1	Mode	0 = Off 1 = On 2 = On Stator		23F3h	1011	enum
C5.8.17	Ch6 Sensor Fault					
C5.8.17.1	Mode	0 = Fault F114 and F122 1 = Alarm A114 and A122		23EBh	1003	enum
C5.8.18	Ch6 Overtemperature					
C5.8.18.1	Mode	0 = Fault F106 1 = Alarm A106 2 = F106 and A106		23DAh	986	enum
C5.8.18.2	Fault Level	0 to 250 $°C$	0	23DBh	987	8bit
C5.8.18.3	Alarm Level	0 to 250 $°C$	0	23DCh	988	8bit
C5.8.18.4	Alarm Reset	0 to 250 $°C$	0	23DDh	989	8bit
C5.9	Motor Thermal Class					
C5.9.1	Programming Mode	0 = Standard 1 = Custom		23A6h	934	enum
C5.9.2	Action Mode	0 = Inactive 1 = Fault F005 2 = Alarm A005 3 = F005 and A005		23A7h	935	enum
C5.9.3	Alarm Level	0 to 100 %	0	23A8h	936	8bit
C5.9.4	Alarm Reset	0 to 100 %	0	23A9h	937	8bit
C5.9.5	Motor Temperature	0 = T.C. + PT100 1 = T.C. + Th.Im.		23AAh	938	enum
C5.9.6	Thermal Class			23ABh	939	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		0 = Automatic 1 = Class 10 2 = Class 15 3 = Class 20 4 = Class 25 5 = Class 30 6 = Class 35 7 = Class 40 8 = Class 45				
C5.9.7	Motor Data					
C5.9.7.1	Insulation Class	0 = Class A 105°C 1 = Class E 120°C 2 = Class B 130°C 3 = Class F 155°C 4 = Class H 180°C 5 = Class N 200°C 6 = Class R 220°C 7 = Class S 240°C 8 = Class 250°C		23ACh	940	enum
C5.9.7.2	Temperature Rise	0 to 200 $°C$	0	23AEh	942	8bit
C5.9.7.3	Ambient Temperature	0 to 200 $°C$	0	23ADh	941	8bit
C5.9.7.4	Locked Rotor Time	1 to 100 s	0	23AFh	943	8bit
C5.9.7.5	Locked Rotor Current	2.0 to 10.0 x	1	23B0h	944	8bit
C5.9.7.6	Heating Time Constant	1 to 2880 min	0	23B1h	945	16bit
C5.9.7.7	Cooling Time Constant	1 to 8640 min	0	23B2h	946	16bit
C5.9.8	Thermal Image					
C5.9.8.1	Reset	0 to 8640 min	0	23B3h	947	16bit
C5.10	SSW Short Circuit					
C5.10.1	Motor Off	0 = Inactive 1 = Fault F019		239Ah	922	enum
C5.10.2	Motor On	0 = Inactive 1 = Fault F020		239Bh	923	enum
C5.11	Fault Auto-Reset					
C5.11.1	Mode	0 = Off 1 = On		20CFh	207	enum
C5.11.2	Time	3 to 600 s	0	20D0h	208	16bit
		C6 Configurations\HMI				
C6.1	Password					
C6.1.1	Password	0 to 9999	0	20D2h	210	16bit
C6.1.2	Password Options	0 = Off 1 = On 2 = Change Password		20C8h	200	enum
C6.2	Language					

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C6.2.1	Language	0 = Português 1 = English 2 = Español		20C9h	201	enum
C6.3	Date and Time					
C6.3.2	Day of the Week	0 = Sunday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday		20C3h	195	enum
C6.4	Main Screen					
C6.5	LCD Backlight					
C6.5.1	Level	1 to 15	0	20DAh	218	8bit
C6.6	Communication Timeout					
C6.6.1	Mode	0 = Inactive 1 = Fault F127 2 = Alarm A127		20BEh	190	enum
C6.6.2	Alarm Action	0 = Indicates Only 1 = Ramp Stop 2 = General Disable 3 = Change to LOC 4 = Change to REM		20BFh	191	enum
C6.6.3	Time	1 to 999 s	0	20C0h	192	16bit
C7 Configurations\Special Functions						
C7.1	Forward/Reverse					
C7.1.1	Mode	0 = Inactive 1 = By Contactor 2 = Only for JOG		20E4h	228	enum
C7.2	Kick Start					
C7.2.1	Mode	0 = Off 1 = On		2208h	520	enum
C7.2.2	Time	0.1 to 2.0 s	1	2209h	521	8bit
C7.2.3	Voltage	70 to 90 %	0	220Ah	522	8bit
C7.2.4	Current	300 to 700 %	0	220Bh	523	16bit
C7.3	Jog					
C7.3.1	Mode	0 = Off 1 = On		21FEh	510	enum
C7.3.2	Level	10 to 100 %	0	21FFh	511	8bit
C7.4	Braking					

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C7.4.1	Mode	0 = Inactive 1 = Reverse 2 = Optimal 3 = DC		21F4h	500	enum
C7.4.2	Time	1 to 299 s	0	21F5h	501	16bit
C7.4.3	Level	30 to 70 %	0	21F6h	502	8bit
C7.4.4	End	0 = Inactive 1 = Automatic		21F7h	503	enum
		C8 Configurations\Communication				
C8.1	I/O Data					
C8.1.1	Data Read					
C8.1.1.1	Slot 1 1st Word	1 to 50	0	22C8h	712	8bit
C8.1.1.2	Slot 1 Quantity	1 to 50	0	22C9h	713	8bit
C8.1.1.3	Slot 2 1st Word	1 to 50	0	22F1h	753	8bit
C8.1.1.4	Slot 2 Quantity	1 to 50	0	22F2h	754	8bit
C8.1.1.5	Word #1	0 to 65535	0	2514h	1300	16bit
C8.1.1.6	Word #2	0 to 65535	0	2515h	1301	16bit
C8.1.1.7	Word #3	0 to 65535	0	2516h	1302	16bit
C8.1.1.8	Word #4	0 to 65535	0	2517h	1303	16bit
C8.1.1.9	Word #5	0 to 65535	0	2518h	1304	16bit
C8.1.1.10	Word #6	0 to 65535	0	2519h	1305	16bit
C8.1.1.11	Word #7	0 to 65535	0	251Ah	1306	16bit
C8.1.1.12	Word #8	0 to 65535	0	251Bh	1307	16bit
C8.1.1.13	Word #9	0 to 65535	0	251Ch	1308	16bit
C8.1.1.14	Word #10	0 to 65535	0	251Dh	1309	16bit
C8.1.1.15	Word #11	0 to 65535	0	251Eh	1310	16bit
C8.1.1.16	Word #12	0 to 65535	0	251Fh	1311	16bit
C8.1.1.17	Word #13	0 to 65535	0	2520h	1312	16bit
C8.1.1.18	Word #14	0 to 65535	0	2521h	1313	16bit
C8.1.1.19	Word #15	0 to 65535	0	2522h	1314	16bit
C8.1.1.20	Word #16	0 to 65535	0	2523h	1315	16bit
C8.1.1.21	Word #17	0 to 65535	0	2524h	1316	16bit
C8.1.1.22	Word #18	0 to 65535	0	2525h	1317	16bit
C8.1.1.23	Word #19	0 to 65535	0	2526h	1318	16bit
C8.1.1.24	Word #20	0 to 65535	0	2527h	1319	16bit
C8.1.1.25	Word #21	0 to 65535	0	2528h	1320	16bit
C8.1.1.26	Word #22	0 to 65535	0	2529h	1321	16bit
C8.1.1.27	Word #23	0 to 65535	0	252Ah	1322	16bit
C8.1.1.28	Word #24	0 to 65535	0	252Bh	1323	16bit
C8.1.1.29	Word #25	0 to 65535	0	252Ch	1324	16bit
C8.1.1.30	Word #26	0 to 65535	0	252Dh	1325	16bit
C8.1.1.31	Word #27	0 to 65535	0	252Eh	1326	16bit
C8.1.1.32	Word #28	0 to 65535	0	252Fh	1327	16bit
C8.1.1.33	Word #29	0 to 65535	0	2530h	1328	16bit
C8.1.1.34	Word #30	0 to 65535	0	2531h	1329	16bit
C8.1.1.35	Word #31	0 to 65535	0	2532h	1330	16bit
C8.1.1.36	Word #32	0 to 65535	0	2533h	1331	16bit
C8.1.1.37	Word #33	0 to 65535	0	2534h	1332	16bit

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C8.1.1.38	Word #34	0 to 65535	0	2535h	1333	16bit
C8.1.1.39	Word #35	0 to 65535	0	2536h	1334	16bit
C8.1.1.40	Word #36	0 to 65535	0	2537h	1335	16bit
C8.1.1.41	Word #37	0 to 65535	0	2538h	1336	16bit
C8.1.1.42	Word #38	0 to 65535	0	2539h	1337	16bit
C8.1.1.43	Word #39	0 to 65535	0	253Ah	1338	16bit
C8.1.1.44	Word #40	0 to 65535	0	253Bh	1339	16bit
C8.1.1.45	Word #41	0 to 65535	0	253Ch	1340	16bit
C8.1.1.46	Word #42	0 to 65535	0	253Dh	1341	16bit
C8.1.1.47	Word #43	0 to 65535	0	253Eh	1342	16bit
C8.1.1.48	Word #44	0 to 65535	0	253Fh	1343	16bit
C8.1.1.49	Word #45	0 to 65535	0	2540h	1344	16bit
C8.1.1.50	Word #46	0 to 65535	0	2541h	1345	16bit
C8.1.1.51	Word #47	0 to 65535	0	2542h	1346	16bit
C8.1.1.52	Word #48	0 to 65535	0	2543h	1347	16bit
C8.1.1.53	Word #49	0 to 65535	0	2544h	1348	16bit
C8.1.1.54	Word #50	0 to 65535	0	2545h	1349	16bit
C8.1.2	Data Write					
C8.1.2.1	Slot 1 1st Word	1 to 20	0	22CAh	714	8bit
C8.1.2.2	Slot 1 Quantity	1 to 20	0	22CBh	715	8bit
C8.1.2.3	Slot 2 1st Word	1 to 20	0	22F3h	755	8bit
C8.1.2.4	Slot 2 Quantity	1 to 20	0	22F4h	756	8bit
C8.1.2.5	Update Delay	0.0 to 999.9 s	1	2383h	899	16bit
C8.1.2.6	Word #1	0 to 65535	0	2578h	1400	16bit
C8.1.2.7	Word #2	0 to 65535	0	2579h	1401	16bit
C8.1.2.8	Word #3	0 to 65535	0	257Ah	1402	16bit
C8.1.2.9	Word #4	0 to 65535	0	257Bh	1403	16bit
C8.1.2.10	Word #5	0 to 65535	0	257Ch	1404	16bit
C8.1.2.11	Word #6	0 to 65535	0	257Dh	1405	16bit
C8.1.2.12	Word #7	0 to 65535	0	257Eh	1406	16bit
C8.1.2.13	Word #8	0 to 65535	0	257Fh	1407	16bit
C8.1.2.14	Word #9	0 to 65535	0	2580h	1408	16bit
C8.1.2.15	Word #10	0 to 65535	0	2581h	1409	16bit
C8.1.2.16	Word #11	0 to 65535	0	2582h	1410	16bit
C8.1.2.17	Word #12	0 to 65535	0	2583h	1411	16bit
C8.1.2.18	Word #13	0 to 65535	0	2584h	1412	16bit
C8.1.2.19	Word #14	0 to 65535	0	2585h	1413	16bit
C8.1.2.20	Word #15	0 to 65535	0	2586h	1414	16bit
C8.1.2.21	Word #16	0 to 65535	0	2587h	1415	16bit
C8.1.2.22	Word #17	0 to 65535	0	2588h	1416	16bit
C8.1.2.23	Word #18	0 to 65535	0	2589h	1417	16bit
C8.1.2.24	Word #19	0 to 65535	0	258Ah	1418	16bit
C8.1.2.25	Word #20	0 to 65535	0	258Bh	1419	16bit
C8.2	RS485 Serial					
C8.2.1	Serial Protocol	0 ... 1 = Reserved 2 = Modbus RTU		22DAh	730	enum
C8.2.2	Address	1 to 247	0	22DBh	731	8bit
C8.2.3	Baud Rate	0 = 9600 bits/s		22DCh	732	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		1 = 19200 bits/s 2 = 38400 bits/s 3 = 57600 bits/s				
C8.2.4	Bytes Config.	0 = 8 bits, no, 1 1 = 8 bits, even,1 2 = 8 bits, odd, 1 3 = 8 bits, no, 2 4 = 8 bits, even,2 5 = 8 bits, odd, 2		22DDh	733	enum
C8.2.5	Timeout					
C8.2.5.1	Mode	0 = Inactive 1 = Fault F128 2 = Alarm A128		22E4h	740	enum
C8.2.5.2	Alarm Action	0 = Indicates Only 1 = Ramp Stop 2 = General Disable 3 = Change to LOC 4 = Change to REM		22E5h	741	enum
C8.2.5.3	Timeout	0.0 to 999.9 s	1	22DEh	734	16bit
C8.3	Anybus-CC					
C8.3.1	Update Configuration	0 = Normal Operation 1 = Update configuration		22EDh	749	enum
C8.3.2	Address	0 to 255	0	22F5h	757	8bit
C8.3.3	Baud Rate	0 = 125 kbps 1 = 250 kbps 2 = 500 kbps 3 = Autobaud		22F6h	758	enum
C8.3.4	IP Address Configuration	0 = Parameters 1 = DHCP 2 = DCP		22F8h	760	enum
C8.3.5	IP Address	0.0.0.0 to 255.255.255.255		22FAh	762	ip_address
C8.3.6	CIDR	0 = Reserved 1 = 128.0.0.0 2 = 192.0.0.0 3 = 224.0.0.0 4 = 240.0.0.0 5 = 248.0.0.0 6 = 252.0.0.0 7 = 254.0.0.0 8 = 255.0.0.0 9 = 255.128.0.0 10 = 255.192.0.0		22F9h	761	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		11 = 255.224.0.0 12 = 255.240.0.0 13 = 255.248.0.0 14 = 255.252.0.0 15 = 255.254.0.0 16 = 255.255.0.0 17 = 255.255.128.0 18 = 255.255.192.0 19 = 255.255.224.0 20 = 255.255.240.0 21 = 255.255.248.0 22 = 255.255.252.0 23 = 255.255.254.0 24 = 255.255.255.0 25 = 255.255.255.128 26 = 255.255.255.192 27 = 255.255.255.224 28 = 255.255.255.240 29 = 255.255.255.248 30 = 255.255.255.252 31 = 255.255.255.254				
C8.3.7	Gateway	0.0.0.0 to 255.255.255.255		22FEh	766	ip_address
C8.3.8	Station Name Suffix	0 to 254	0	2302h	770	8bit
C8.3.9	Modbus TCP Timeout					
C8.3.9.1	Mode	0 = Inactive 1 = Fault F131 2 = Alarm A131		2303h	771	enum
C8.3.9.2	Alarm Action	0 = Indicates Only 1 = Ramp Stop 2 = General Disable 3 = Change to LOC 4 = Change to REM		2304h	772	enum
C8.3.9.3	Modbus TCP Timeout	0.0 to 999.9 s	1	22F7h	759	16bit
C8.3.10	Off Line Error					
C8.3.10.1	Mode	0 = Inactive 1 = Fault F129 2 = Alarm A129		2381h	897	enum
C8.3.10.2	Alarm Action	0 = Indicates Only 1 = Ramp Stop 2 = General Disable 3 = Change to LOC 4 = Change to REM		2382h	898	enum
C8.4	CANopen/DeviceNet					
C8.4.4	Bus Off Reset	0 = Manual		22BFh	703	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		1 = Automatic				
C8.4.5	CAN Error					
C8.4.5.1	Mode	0 = Inactive 1 = Fault 2 = Alarm		22D3h	723	enum
C8.4.5.2	Alarm Action	0 = Indicates Only 1 = Ramp Stop 2 = General Disable 3 = Change to LOC 4 = Change to REM		22D4h	724	enum
C8.6	Bluetooth					
C8.6.1	Mode	0 = Off 1 = On		2320h	800	enum
		C9 Configurations\SSW900				
C9.1	Nominal Data					
C9.1.1	Current	0 = 10 A 1 = 17 A 2 = 24 A 3 = 30 A 4 = 45 A 5 = 61 A 6 = 85 A 7 = 105 A 8 = 130 A 9 = 171 A 10 = 200 A 11 = 255 A 12 = 312 A 13 = 365 A 14 = 412 A 15 = 480 A 16 = 604 A 17 = 670 A 18 = 820 A 19 = 950 A 20 = 1100 A 21 = 1400 A		2127h	295	enum
C9.2	Types of Connections					
C9.2.1	Delta Inside	0 = Off 1 = On		2096h	150	enum
C9.2.2	External Bypass	0 = Without 1 = With		208Ch	140	enum
C9.3	Accessories Config.					

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C9.3.1	Slot 1	0 = Automatic 1 = Anybus-CC 2 = RS-485 3 = PT100 4 = I/Os Exp. 5 = Profibus 6 = CAN 7 = Ethernet 8 = External Current Acqu.		2151h	337	enum
C9.3.2	Slot 2	0 = Automatic 1 = Anybus-CC 2 = RS-485 3 = PT100 4 = I/Os Exp. 5 = Profibus 6 = CAN 7 = Ethernet 8 = External Current Acqu.		2152h	338	enum
C9.4	Fan Configuration					
C9.4.1	Mode	0 = Always Off 1 = Always On 2 = Controlled		20CBh	203	enum
C10 Configurations\Load / Save Parameters						
C10.1	Load / Save User					
C10.1.1	Mode	0 = Not Used 1 = Load User 1 2 = Load User 2 3 = Reserved 4 = Save User 1 5 = Save User 2 6 = Reserved		20CEh	206	enum
C10.2	Copy Function HMI					
C10.2.1	Mode	0 = Off 1 = SSW -> HMI 2 = HMI -> SSW		213Fh	319	enum
C10.3	Erase Diagnostics					
C10.3.1	Mode	0 ... 1 = Not Used 2 = Fault 3 = Alarms 4 = Events 5 = Motor ON 6 = Temperaturas 7 = Hours Control		20CDh	205	enum

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
		8 = Thermal Class Status				
C10.4	Load Factory Default					
C10.4.1	Mode	0 = No 1 = Yes		20CCh	204	enum
C10.5	Save Changed Param.					
C10.5.1	Mode	0 = No 1 = Yes		20D1h	209	enum
C11 Configurations\SoftPLC						
C11.1	Mode	0 = Stop Program 1 = Run Program		244Dh	1101	enum
C11.2	Action App. Not Running	0 = Inactive 1 = Alarm A708 2 = Fault F708		244Fh	1103	enum
C11.3	Parameter					
C11.3.1	User #1	-10000 to 10000	0	2456h	1110	s32bit
C11.3.2	User #2	-10000 to 10000	0	2458h	1112	s32bit
C11.3.3	User #3	-10000 to 10000	0	245Ah	1114	s32bit
C11.3.4	User #4	-10000 to 10000	0	245Ch	1116	s32bit
C11.3.5	User #5	-10000 to 10000	0	245Eh	1118	s32bit
C11.3.6	User #6	-10000 to 10000	0	2460h	1120	s32bit
C11.3.7	User #7	-10000 to 10000	0	2462h	1122	s32bit
C11.3.8	User #8	-10000 to 10000	0	2464h	1124	s32bit
C11.3.9	User #9	-10000 to 10000	0	2466h	1126	s32bit
C11.3.10	User #10	-10000 to 10000	0	2468h	1128	s32bit
C11.3.11	User #11	-10000 to 10000	0	246Ah	1130	s32bit
C11.3.12	User #12	-10000 to 10000	0	246Ch	1132	s32bit
C11.3.13	User #13	-10000 to 10000	0	246Eh	1134	s32bit
C11.3.14	User #14	-10000 to 10000	0	2470h	1136	s32bit
C11.3.15	User #15	-10000 to 10000	0	2472h	1138	s32bit
C11.3.16	User #16	-10000 to 10000	0	2474h	1140	s32bit
C11.3.17	User #17	-10000 to 10000	0	2476h	1142	s32bit
C11.3.18	User #18	-10000 to 10000	0	2478h	1144	s32bit
C11.3.19	User #19	-10000 to 10000	0	247Ah	1146	s32bit
C11.3.20	User #20	-10000 to 10000	0	247Ch	1148	s32bit
C11.3.21	User #21	-10000 to 10000	0	247Eh	1150	s32bit
C11.3.22	User #22	-10000 to 10000	0	2480h	1152	s32bit
C11.3.23	User #23	-10000 to 10000	0	2482h	1154	s32bit
C11.3.24	User #24	-10000 to 10000	0	2484h	1156	s32bit
C11.3.25	User #25	-10000 to 10000	0	2486h	1158	s32bit
C11.3.26	User #26	-10000 to 10000	0	2488h	1160	s32bit
C11.3.27	User #27	-10000 to 10000	0	248Ah	1162	s32bit
C11.3.28	User #28	-10000 to 10000	0	248Ch	1164	s32bit
C11.3.29	User #29	-10000 to 10000	0	248Eh	1166	s32bit
C11.3.30	User #30	-10000 to 10000	0	2490h	1168	s32bit
C11.3.31	User #31	-10000 to 10000	0	2492h	1170	s32bit

Parameter	Description	Range of values	Decimal places	Index	Net Id	Size
C11.3.32	User #32	-10000 to 10000	0	2494h	1172	s32bit
C11.3.33	User #33	-10000 to 10000	0	2496h	1174	s32bit
C11.3.34	User #34	-10000 to 10000	0	2498h	1176	s32bit
C11.3.35	User #35	-10000 to 10000	0	249Ah	1178	s32bit
C11.3.36	User #36	-10000 to 10000	0	249Ch	1180	s32bit
C11.3.37	User #37	-10000 to 10000	0	249Eh	1182	s32bit
C11.3.38	User #38	-10000 to 10000	0	24A0h	1184	s32bit
C11.3.39	User #39	-10000 to 10000	0	24A2h	1186	s32bit
C11.3.40	User #40	-10000 to 10000	0	24A4h	1188	s32bit
C11.3.41	User #41	-10000 to 10000	0	24A6h	1190	s32bit
C11.3.42	User #42	-10000 to 10000	0	24A8h	1192	s32bit
C11.3.43	User #43	-10000 to 10000	0	24AAh	1194	s32bit
C11.3.44	User #44	-10000 to 10000	0	24ACh	1196	s32bit
C11.3.45	User #45	-10000 to 10000	0	24AEh	1198	s32bit
C11.3.46	User #46	-10000 to 10000	0	24B0h	1200	s32bit
C11.3.47	User #47	-10000 to 10000	0	24B2h	1202	s32bit
C11.3.48	User #48	-10000 to 10000	0	24B4h	1204	s32bit
C11.3.49	User #49	-10000 to 10000	0	24B6h	1206	s32bit
C11.3.50	User #50	-10000 to 10000	0	24B8h	1208	s32bit
A1 Assistant\Oriented Start-up						
A1.1	Mode	0 = No 1 = Yes		213Dh	317	enum

Table A.3: *Description of the parameter data types*

Data Type	Description
enum	Enumerated type (unsigned 8-bit) contains a list of values with function description for each item.
8bit	Unsigned 8-bit integer, ranges from 0 to 255.
16bit	Unsigned 16-bit integer, ranges from 0 to 65,535.
s16bit	Signed 16-bit integer, ranges from -32,768 to 32,767.
32bit	Unsigned 32-bit integer, ranges from 0 to 4,294,967,295.
s32bit	Signed 32-bit integer, ranges from -2,147,483,648 to 2,147,483,647.
date	Displays the date and time value in the format below: second (1 byte) minute (1 byte) hour (1 byte) day (1 byte) month (1 byte) reserved (1 byte) year (2 bytes)
TIME	Displays the time in the format hh:mm:ss. For network protocols, this data type is transferred as an unsigned 32-bit integer value representing the number of seconds.
ip_address	Unsigned 32-bit integer representing the octets of the IP address.
MAC_ADDRESS	48-bit identifier displayed in XX:XX:XX:XX:XX:XX format.
STRING_ASCII	Text string. For network protocols, this data type is transferred as a string filled with zeros (\0) to the end (maximum parameter size plus one).

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