DeviceNet

CFW320-CCAN

User's Guide

DeviceNet User's Guide

Series: CFW320

Language: English

Document: 10009156404 / 00

Build 232

Publication Date: 03/2022

The information below describes the reviews made in this manual.

Version	Revision	Description
-	R00	First edition

ABOUT THE MANUAL

This manual supplies the necessary information for the operation of the CFW320 frequency inverter using the DeviceNet protocol. This manual must be used together with the CFW320 user's manual and programming manual.

ABBREVIATIONS AND DEFINITIONS

ASCII American Standard Code for Information Interchange
CiA CAN in Automation
CIP Common Industrial Protocol
CRC Cycling Redundancy Check
HMI Human-Machine Interface
ODVA Open DeviceNet Vendor Association
PLC Programmable Logic Controller
ro Read only
rw Read/write

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 DeviceNet protocol was developed based on the following specifications and documents:

Document	Version	Source
CAN Specification	2.0	CiA
Volume One - Common Industrial Protocol (CIP) Specification	3.2	ODVA
Volume Three - DeviceNet Adaptation of CIP	1.4	ODVA
Planning and Installation Manual - DeviceNet Cable System	PUB00027R1	ODVA

In order to obtain this documentation, consult ODVA, which is nowadays the organization that keeps, publishes and updates the information related to the DeviceNet network.

1 MAIN CHARACTERISTICS

Below are the main characteristics for communication of the frequency inverter CFW320 with DeviceNet accessory.

- Uses the Set of Predefined Master/Slave Connections (Group 2 Only Server).
- It is supplied with an EDS file for the network master configuration.
- Allows up to 6 input words and 6 output words for cyclic data communication.
- Acyclic data available for parameterization (Explicit Messages).

2 DEVICENET COMMUNICATION INTERFACE

The accessory CFW320 frequency inverter features a CAN interface. It can be used for communication in Devicenet protocol as a network slave. The characteristics of this interface are described below.

2.1 CAN INTERFACE CHARACTERISTICS

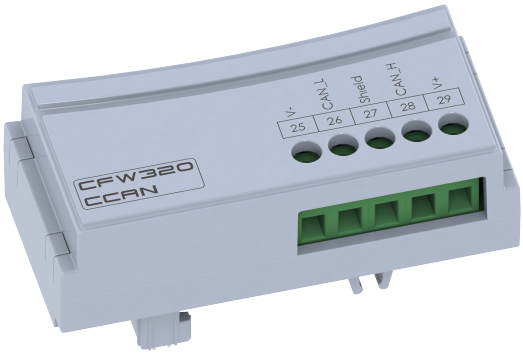


Figure 2.1: *Acessório CCAN*

- Interface galvanically insulated and with differential signal, providing more robustness against electromagnetic interference.
- External power supply of 24 V.
- It allows the connection of up to 64 devices to the same segment. More devices can be connected by using repeaters[1].
- Maximum bus length of 1000 meters.

2.2 PIN ASSIGNMENT OF THE CONNECTOR

The CAN interface has a 5-way connector with the following pin assignment:

Figure 2.2: *Detalhe do conector CAN*

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

Pin	Name	Function
25	V-	Negative pole of the power supply
26	CAN_L	Communication signal CAN_L
27	Shield	Cable shield
28	CAN_H	Communication signal CAN_H
29	V+	Positive pole of the power supply

2.3 POWER SUPLLY

The CAN interfaces require an external power supply between pins 25 and 29 of the network connector. The data for individual consumption and input voltage are shown in the Table 2.2 on page 9.

[1] The maximum number of devices that can be connected to the network also depends on the protocol used.

Table 2.2: Characteristics of the supply for the CAN interface

Power Suplly (Vdc)		
Minimum	Maximum	Recomended
11	30	24
Current (mA)		
Typical		Maximum
30		50

2.4 INDICATIONS

The alarm, fault and status indications of the DeviceNet communication for the CFW320 frequency inverter are made trough the HMI and parameters of the product.

3 DEVICENET NETWORK INSTALLATION

The DeviceNet 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 DeviceNet interface generally allow the configuration of the desired baud rate, ranging from 125 kbit/s até 500 kbit/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 on page 10 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
125 kbit/s	500 m
250 kbit/s	250 m
500 kbit/s	100 m

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

3.2 ADDRESS IN THE DEVICENET NETWORK

Each DeviceNet network device must have an address or MAC ID, and may range from 0 to 63. 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 Table 3.2 on page 10 shows the recommended characteristics for the cable.

Table 3.2: *DeviceNet cable characteristics*

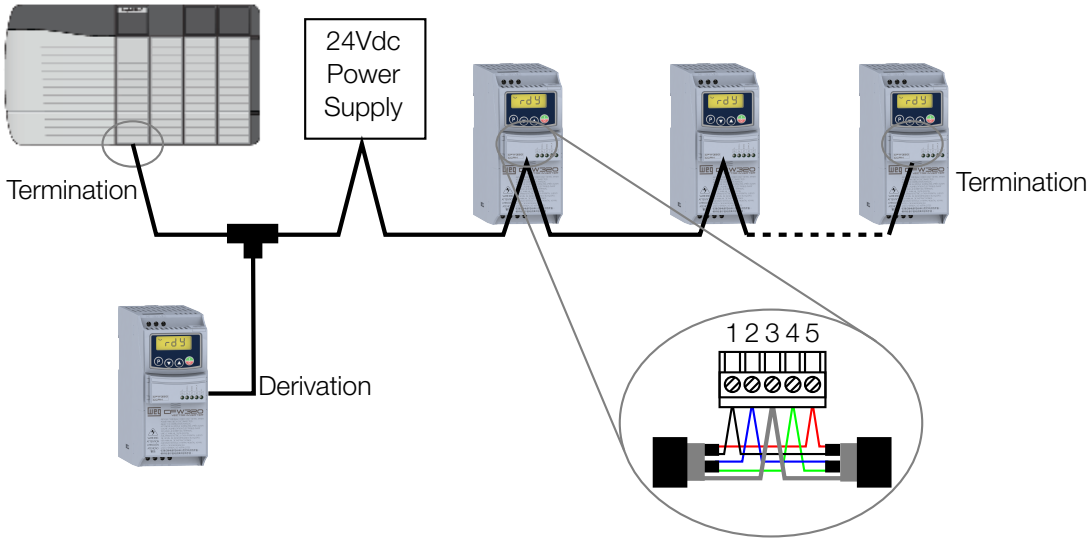
Cable Length (m)	Resistence per Meter (mΩ/m)	Conductor Cross Section (mm^2)
0 ... 40	70	0.25 ... 0.34
40 ... 300	<60	0.34 ... 0.60
300 ... 600	<40	0.50 ... 0.60
600 ... 1000	<26	0.75 ... 0.80

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. If you use derivations, the limits of length for derivation defined by the DeviceNet specification must be observed. 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: *DeviceNet 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 PARAMETERS

4.1 COMMANDS AND COMMUNICATION STATUS

See below the parameters related to the states and commands through the communication networks available for the frequency inverter.

P313 - Action for Communic. Error

Adjustable Range:	0 = Inactive 1 = Ramp Stop 2 = General Disable 3 = Go to LOC 4 = LOC Keep Enab. 5 = Cause Fault	**Factory Setting:**	1

Description:

It allows the selection of the action to be executed by the device, if it is controlled via network and a communication error is detected.

The following events are considered communication errors:

- Alarm A133/Fault F233: CAN interface not powered.
- Alarm A134/Fault F234: bus off.
- Alarm A135/Fault F235: CANopen communication error (Node Guarding/Heartbeat).
- Alarm A136/Fault F236: DeviceNet master in Idle mode.
- Alarm A137/Fault F237: detected timeout in one or more DeviceNet I/O connections.

The actions described in this parameter are executed by means of the automatic writing of the selected actions in the respective bits of the interface control words. Therefore, in order that the commands are effective, it is necessary that the device be programmed to be controlled via the used network interface (with exception of option “Causes a Fault”, which blocks the equipment even if it is not controlled by network). This programming is achieved by means of parameters P220 to P228.

Table 4.1: *P313 options*

Indication	Description
0 = Inactive	No action is taken and the drive remains in the existing status.
1 = Ramp Stop	A stop command with deceleration ramp is executed and the motor stops according to the programmed deceleration ramp.
2 = General Disable	The drive is disabled by removing the General Enabling and the motor coasts to stop.
3 = Go to LOC	The drive commands change to Local.
4 = LOC Keep Enab.	The drive commands change to Local, but the status of the enabling and speed reference commands received via network are kept, providing that the drive has been programmed to use in Local mode the commands via HMI, or 3-wire start/stop and speed reference via either HMI or electronic potentiometer.
5 = Cause Fault	Instead of an alarm, the communication error causes a drive fault, so that a drive fault reset becomes necessary in order to restore normal operation.

P680 - Logical Status

Adjustable Range:	0 to FFFF (hexa) Bit 0 = Reserved Bit 1 = Run Command Bit 2 = Fire Mode Bit 3 to 4 = Reserved Bit 5 = 2nd Ramp Bit 6 = Config. Mode Bit 7 = Alarm Bit 8 = Running Bit 9 = Enabled Bit 10 = Forward Bit 11 = JOG Bit 12 = Remote Bit 13 = Subvoltage Bit 14 = Reserved Bit 15 = Fault	**Factory Setting:**	-
Properties:	ro		

Description:

The inverter status word is unique for all the sources and can only be accessed for reading. It indicates all the relevant operating status and modes of the inverter. The function of each bit of P680 is described in Table 4.2 on page 13.

Table 4.2: *P680 bits function*

Bit	Value/Description
Bit 0 Reserved	-
Bit 1 Run Command	**0:** there was no Run command **1:** there was Run command
Bit 2 Fire Mode	**0:** fire Mode function inactive **1:** fire Mode function active
Bit 3 … 4 Reserved	-
Bit 5 2nd Ramp	**0:** 1st acceleration and deceleration ramp by P100 and P101 **1:** 2nd acceleration and deceleration ramp by P102 and P103
Bit 6 Config. Mode	**0:** inverter operating in normal conditions **1:** inverter in configuration state. It indicates a special condition in which the inverter cannot be enabled, because it has parameterization incompatibility
Bit 7 Alarm	**0:** inverter is not in alarm state **1:** inverter is in alarm state
Bit 8 Running	**0:** motor is stopped **1:** inverter is running according to reference and command
Bit 9 Enabled	**0:** inverter is disabled **1:** inverter is enabled and ready to run the motor
Bit 10 Forward	**0:** motor is running in the reverse direction **1:** motor is running in the forward direction
Bit 11 JOG	**0:** JOG function inactive **1:** JOG function active
Bit 12 Remote	**0:** inverter in Local mode **1:** inverter in Remote mode
Bit 13 Subvoltage	**0:** no undervoltage **1:** with undervoltage
Bit 14 Reserved	-
Bit 15 Fault	**0:** inverter is not in fault state **1:** some fault registered by the inverter

P681 - 13-Bit Speed

Adjustable Range:	0 to FFFF (hexa)	**Factory Setting:**	-
Properties:	ro		

Description:

It defines the 13-bit speed reference. The 13-bit Frequency Reference is a scale based on the motor rated speed (P402) or on the motor rated frequency (P403). In the inverter, parameter P403 is taken as the base to determine the frequency reference.

Thus, the 13-bit frequency value has a range of 16 bits with signal, that is, -32768 to 32767; however, the rated frequency in P403 is equivalent to the value 8192. Therefore, the maximum value in the range 32767 is equivalent to four times P403:

- P681 = 0000h (0 decimal) → motor speed = 0
- P681 = 2000h (8192 decimal) → motor speed = rated frequency

Intermediate or higher frequency values can be obtained by using this scale. E.g., for a 60Hz rated frequency motor, if the value read is 2048 (0800h), then, to obtain the value in Hz one must calculate:

8192 => 60 Hz
2048 => Frequency

$$\text{Frequency} = \frac{2048 \times 60}{8192}$$

Frequency = 15 Hz

Negative values in this parameter indicate that the motor is running in the reverse direction.

NOTE!
The values transmitted over the network have a scale limitation, allowing a maximum of 4 times the rated frequency of the motor, with saturation in 32767 (or -32768).

P684 - CO/DN/DP/ETH Control

Adjustable Range:	0 to FFFF (hexa) Bit 0 = Ramp Enable Bit 1 = General Enable Bit 2 = Run Forward Bit 3 = JOG Enable Bit 4 = Remote Bit 5 = 2nd Ramp Bit 6 = Reserved Bit 7 = Fault Reset Bit 8 to 15 = Reserved	**Factory Setting:**	-
Properties:	ro		

Description:

The inverter control word has read and write access only via network interface, but read only access is permitted for the other sources (keypad, SoftPLC). Each bit function is described as per Table 4.3 on page 15. The value of P684 is indicated in hexadecimal.

Table 4.3: P684 bits function

Bit	Value/Description
Bit 0 Ramp Enable	**0**: stops the motor by deceleration ramp **1**: run the motor according to the acceleration ramp until reaching the speed reference value
Bit 1 General Enable	**0**: disables the inverter, interrupting the power supply to the motor **1**: enables the inverter, allowing the operation of the motor
Bit 2 Run Forward	**0**: run the motor in the opposite direction of the reference signal (reverse) **1**: run the motor in the direction of the reference signal (forward)
Bit 3 JOG Enable	**0**: disable JOG function **1**: enable JOG function
Bit 4 Remote	**0**: inverter goes into Local mode **1**: inverter goes into Remote mode
Bit 5 2nd Ramp	**0**: acceleration and deceleration ramp by P100 and P101 **1**: acceleration and deceleration ramp by P102 and P103
Bit 6 Reserved	-
Bit 7 Fault Reset	**0**: no function **1**: if in fault state, reset the fault
Bit 8 ... 15 Reserved	-

P685 - CO/DN/DP/ETH Speed Ref

Adjustable Range:	0 to FFFF (hexa)	**Factory Setting:**	-
Properties:	ro		

Description:

It allows programming the motor speed reference via communication interfaces only. For other sources (HMI, etc.), it behaves as a read-only parameter.

To enable the use of the reference written in this parameter, the product must be programmed to use the speed reference via communication network. This is programming is done using parameters P221 and P222.

This word uses a 13-bit resolution with signal to represent the motor rated frequency (P403):

- P683 = 0000h (0 decimal) → speed reference = 0.
  P683 = 2000h (8192 decimal) → speed reference = rated frequency (P403).
- P685 = 0000h (0 decimal) → speed reference = 0.
  P685 = 2000h (8192 decimal) → speed reference = rated frequency (P403).

Intermediate or higher reference values can be programmed by using this scale. E.g. 60Hz rated frequency, to obtain a speed reference of 30 Hz one must calculate:

60 Hz => 8192
30 Hz => 13 bits reference

$$13 \text{ bits reference} = \frac{30 \times 8192}{60}$$

13 bits reference = 4096 => Value corresponding to 30 Hz in a 13 bit scale

This parameter also accepts negative values to revert the motor speed direction. The reference speed direction, however, depends also on the control word - P684 bit 2 setting:

- Bit 2 = 1 and P685 > 0: reference for forward direction
- Bit 2 = 1 and P685 < 0: reference for reverse direction
- Bit 2 = 0 and P685 > 0: reference for reverse direction
- Bit 2 = 0 and P685 < 0: reference for forward direction

NOTE!
The values transmitted over the network have a scale limitation, allowing a maximum of 4 times the rated frequency of the motor, with saturation in 32767 (or -32768).

P695 - DOx Value

Adjustable Range:	0 to F (hexa) Bit 0 = DO1 Bit 1 = DO2 Bit 2 = DO3 Bit 3 = DO4	**Factory Setting:**	-
Properties:	ro		

Description:

It provides access for monitoring and controlling the inverter by using the communication interfaces. Each bit represents the value for a digital output. The value written in this parameter is used as the digital output value, providing that the function for the desired digital output be programmed for “P695 value”.

Table 4.4: *P695 bits function*

Bit	Value/Description
Bit 0 DO1	**0**: DO1 output open. **1**: DO1 output closed.
Bit 1 DO2	**0**: DO2 output open. **1**: DO2 output closed.
Bit 2 DO3	**0**: DO3 output open. **1**: DO3 output closed.
Bit 3 DO4	**0**: DO4 output open. **1**: DO4 output closed.

P696 - AOx Value 1

P697 - AOx Value 2

Adjustable Range:	0 to FFFF (hexa)	**Factory Setting:**	-
Properties:	ro		

Description:

It provides access for monitoring and controlling the inverter by using the communication interfaces.

They allow the control of the analog outputs by means of network interfaces (Serial, CAN, etc.). These parameters cannot be changed via HMI.

The value written in these parameters is used as the analog output value, providing that the function for the desired analog output be programmed for “P696 / P697 value”, at the parameters P251, P254.

The value must be written in a 15-bit scale (7FFFh = 32767) to represent 100 % of the output desired value, i.e.:

- P696 = 0000h (0 decimal) → analog output value = 0 %
- P696 = 7FFFh (32767 decimal) → analog output value = 100 %

The showed example was for P696, but the same scale is also used for the parameters P697. For instance, to control the analog output 1 via serial, the following programming must be done:

- Choose a parameter from P696, P697 to be the value used by the analog output 1. For this example, we are going to select P696.
- Program the option “P696 value” as the function for the analog output 1 in P254.
- Using the network interface, write in P696 the desired value for the analog output 1, between 0 and 100 %, according to the parameter scale.

NOTE!
If the analog output is programmed for working from -10 V to 10 V, negative values for this parameter must be used to command the output with negative voltage values, i.e., -32768 to 32767 represent a variation from -10 V to 10 V at the analog output.

4.2 DEVICENET

See below the parameters to configure and operate the CAN interface.

P700 - CAN Protocol

Adjustable Range:	1 = CANopen 2 = DeviceNet	**Factory Setting:**	2

Description:

It allows selecting the desired protocol for the CAN interface. If this parameter is changed, the change takes effect only if the CAN interface is not powered, it is in auto-baud or after the equipment is switched off and on again.

Table 4.5: *P700 options*

Indication	Description
1 = CANopen	Enables the CAN interface with CANopen protocol.
2 = DeviceNet	Enables the CAN interface with DeviceNet protocol.

P701 - CAN Address

Adjustable Range:	0 to 127	**Factory Setting:**	63

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 P700:

- P700 = 1 (CANopen): valid addresses: 1 to 127.
- P700 = 2 (DeviceNet): valid addresses: 0 to 63.

If this parameter is changed, the change takes effect only if the CAN interface is not powered, auto-baud or after the equipment is switched off and on again.

P702 - CAN Baud Rate

Adjustable Range:	0 = 1 Mbps/Auto 1 = Reserved/Auto 2 = 500 Kbps 3 = 250 Kbps 4 = 125 Kbps 5 = 100 Kbps/Auto 6 = 50 Kbps/Auto 7 = 20 Kbps/Auto 8 = 10 Kbps/Auto	**Factory Setting:**	0

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 P700:

- P700 = 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.
- P700 = 2 (DeviceNet): only the 500, 250 and 125 Kbit/s rates are supported. Other options will enable the automatic baud rate detection function – autobaud.

If this parameter is changed, the change takes effect only if the CAN interface is not powered or after the equipment is switched off and on again. After a successful detection, the baud rate parameter (P702) changes automatically to the detected rate. In order to execute the autobaud function again, it is necessary to change the parameter P702 to one of the 'Autobaud' options.

Table 4.6: *P702 options*

Indication	Description
0 = 1 Mbps/Auto	CAN baud rate (automatic detection for DeviceNet).
1 = Reserved/Auto	Automatic detection for DeviceNet.
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).

P703 - Bus Off Reset

Adjustable Range:	0 = Manual 1 = Automatic	**Factory Setting:**	1

Description:

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

Table 4.7: *P703 options*

Indication	Description
0 = Manual	If bus off occurs, the A134/F234 alarm will be indicated on the HMI, theaction programmed in parameter P313 will be executed and the communication will be disabled. In order that the inverter communicates again through the CAN interface, it will be necessary to cycle the power of the inverter.
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 P313.

P705 - CAN Controller Status

Adjustable Range:	0 = Disabled 1 = Auto-baud 2 = CAN Active 3 = Warning 4 = Error Passive 5 = Bus Off 6 = No Bus Power	**Factory Setting:**	-
Properties:	ro		

Description:

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

Table 4.8: P705 options

Indication	Description
0 = Disabled	Inactive CAN interface. It occurs when CAN protocol is not programmed at P705.
1 = Auto-baud	CAN controller is trying to detect baud rate of the network (only for CANopen communication protocol).
2 = CAN Active	CAN interface is active and without errors.
3 = Warning	CAN controller has reached the warning state.
4 = Error Passive	CCAN 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 25 and 29 of the connector.

P706 - RX CAN Telegrams

Adjustable Range:	0 to 9999	**Factory Setting:**	-
Properties:	ro		

Description:

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

P707 - TX CAN Telegrams

Adjustable Range:	0 to 9999	**Factory Setting:**	-
Properties:	ro		

Description:

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

P708 - Bus Off Counter

Adjustable Range:	0 to 9999	**Factory Setting:**	-
Properties:	ro		

Description:

It is a cyclic counter that indicates the number of times the device entered the bus off state in the CAN network. This counter is reset every time the device is switched off, a reset is performed or the parameter maximum limit is reached.

P709 - CAN Lost Messages

Adjustable Range:	0 to 9999	**Factory Setting:**	-
Properties:	ro		

Description:

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

P710 - DeviceNet I/O instances

Adjustable Range:	0 = ODVA Basic 2W 1 = ODVA Extend 2W 2 = Manuf.Spec.2W 3 = Manuf.Spec.3W 4 = Manuf.Spec.4W 5 = Manuf.Spec.5W 6 = Manuf.Spec.6W	**Factory Setting:**	0

Description:

It allows selecting the Assembly class instance for the I/O type communication.

The CFW320 frequency converter has seven setting options. Two of them follow the ODVA AC/DC Drive Profile. The other five represent specific CFW320 frequency converter words. The tables presented next describe each of these control and monitoring words.

NOTE!
If this parameter is changed, it becomes valid only after cycling the power of the product.

0 = Data format for the ODVA Basic Speed (2 words) instances:

Called Basic Speed, these instances represent the simplest operation interface of a device according to the AC/DC Drive Profile. The data mapping is showed below.

Monitoring (Input)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	0						Running1		Faulted
	1	-							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

Control (Output)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	0						Fault Reset		Run Fwd
	1	-							
	2	Speed Reference (low byte)							
	3	Speed Reference (high byte)							

1 = Data format for the ODVA Extended Speed (2 words) instances:

Called Extended Speed, these instances present an equipment operation interface a little bit more refined, which follows the AC/DC Drive Profile. The data mapping is showed below.

Monitoring (Input)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
71	0	At Reference	Ref. from Net	Ctrl from Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

Control (Output)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0		NetRef	NetCtrl		Fault Reset		Warning	Faulted
	1	-							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

The table below presents the meaning of data for the instances 20/70 and 21/71.

Monitoring:

Bit	Valor/Descrição
Bit 0 Faulted	**0**: frequency inverter is not in a fault state **1**: Some fault registered by the frequency inverter. Note: The number of the fault can be read through parameter P049 – Current Fault.
Bit 1 Warning	**0**: frequency inverter is not in an alarm state. **1**: Some alarm registered by the frequency inverter. The number of the alarm can be read by means of parameter P048 – Current Alarm.
Bit 2 Running1 (Fwd)	**0**: The motor is not running forward. **1**: The motor is running forward.
Bit 3 Running2 (Rev)	**0**: The motor is not running in the reverse direction. **1**: The motor is running in the reverse direction.
Bit 4 Ready	**0**: frequency inverter not ready to operate. **1**: frequency inverter ready to operate (Ready, Enabled or Stopping states).
Bit 5 Ctrl from Net	**0**: Drive locally controlled. **1**: Drive remotely controlled.
Bit 6 Ref. from Net	**0**: Speed reference is not being sent via DeviceNet. **1**: Indicates speed reference being sent via DeviceNet.
Bit 7 At Reference	**0**: frequency inverter has not reached the programmed speed yet. **1**: frequency inverter has reached the programmed speed.

- Byte 1 indicates the state of the drive:
  - 0 = Non Existent
  - 1 = Startup
  - 2 = Not Ready
  - 3 = Ready
  - 4 = Enabled
  - 5 = Stopping
  - 6 = Fault Stop
  - 7 = Faulted
- Bytes 2 (low) and 3 (high) represent the effective speed of the motor in RPM.

Control:

Bit	Valor/Descrição
Bit 0 Run Fwd	**0**: Stop motor **1**: The motor runs in the forward direction.
Bit 1 Run Rev	**0**: Stop motor **1**: The motor runs in the reverse direction.
Bit 2 Fault Reset	**0**: Not used. **1**: If in a fault condition, it resets the frequency inverter.
Bits 3 e 4	Reserved.
Bit 5 NetCtrl	**0**: frequency inverter selects the local mode. **1**: frequency inverter selects the remote mode.
Bit 6 NetRef	**0**: Speed reference is not being sent via network. **1**: Speed reference being sent via network.
Bit 7	Reserved.

- Bytes 2 (low) and 3 (high) represent the speed reference of the motor in RPM.

2 = Data format for the Manufacturer Specific 2W (2 words) instances:
3 = Data format for the Manufacturer Specific 3W (3 words) instances:
4 = Data format for the Manufacturer Specific 4W (4 words) instances:
5 = Data format for the Manufacturer Specific 5W (5 words) instances:
6 = Data format for the Manufacturer Specific 6W (6 words) instances:

Called Manufacturer Specific, these instances present the simplest equipment operation interface according to the CFW320 frequency inverter profile. The data mapping is showed below. Besides the command and monitoring words showed above, they make it possible to program up to 6 parameters of the equipment for reading and/or writing via network, through P711 to P718 parameters.

Monitoring (Input)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Functio	Fault	Not Used	Undervoltage	Remote 2	JOG	Reverse	Enabled	Running	Alarm	Config. Mode	2nd Ramp	No Quick Stop	Not Used	Local	Run Command	STO

Control (Output)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	Reserved								Fault Reset	No Quick Stop	2nd Ramp	R1/R2 Mode	Enable JOG	Turn Reverse	General Enable	Enable Ramp

Monitoring (Input)

Table 4.9: *Programming of the I/O words*

	Instance	16 bits word	Function	Option of P710
Const	150	#1	Status Word - P680	2
Programable		#2	Motor Speed - P681	3
		#3	Read #3 DeviceNet	4
		#4	Read #4 DeviceNet	5
		#5	Read #5 DeviceNet	6
		#6	Read #6 DeviceNet	

Control (Output)

Table 4.10: *Programming of the I/O words*

	Instance	16 bits word	Function	Option of P710
Const	100	#1	Control word - P684	2
Programable		#2	Speed reference - P685	3
		#3	Write #3 DeviceNet	4
		#4	Write #4 DeviceNet	5
		#5	Write #5 DeviceNet	6
		#6	Write #6 DeviceNet	

Table 4.11: *P710 options*

Indication	Description
0 = ODVA Basic 2W	Called Basic Speed, these instances represent the simplest operation interface of a device according to the AC/DC Drive Profile
1 = ODVA Extend 2W	Called Extended Speed, these instances present an equipment operation interface a little bit more refined, which follows the AC/DC Drive Profile
2 = Manuf.Spec.2W	These instances represent the operation interface of a device according to the CFW320 frequency converter profile.
3 = Manuf.Spec.3W	These instances represent the operation interface of a device according to the CFW320 frequency converter profile. Besides the control and state words, speed reference and effective value, it is possible to program up to 3 parameters of the equipment itself for reading and/or writing via network.
4 = Manuf.Spec.4W	These instances represent the operation interface of a device according to the CFW320 frequency converter profile. Besides the control and state words, speed reference and effective value, it is possible to program up to 4 parameters of the equipment itself for reading and/or writing via network.
5 = Manuf.Spec.5W	These instances represent the operation interface of a device according to the CFW320 frequency converter profile. Besides the control and state words, speed reference and effective value, it is possible to program up to 5 parameters of the equipment itself for reading and/or writing via network.
6 = Manuf.Spec.6W	These instances represent the operation interface of a device according to the CFW320 frequency converter profile. Besides the control and state words, speed reference and effective value, it is possible to program up to 6 parameters of the equipment itself for reading and/or writing via network.

P711 - DeviceNet Read Word #3

P712 - DeviceNet Read Word #4

P713 - DeviceNet Read Word #5

P714 - DeviceNet Read Word #6

Adjustable Range:	0 to 1199	**Factory Setting:**	0

Description:

It allows the user to program the content of input words (input: slave sends to the master).

Using these parameters, it is possible to program the number of other parameter whose content shall be made available in the input area of the network master.

For instance, in case it is necessary to read the motor current in amperes from the inverter, the value 3 must be programmed in some of these parameters, since the parameter P003 is the parameter that contains this information. Note that the reading value of any parameter is represented with a 16-bit word. Even if the parameter has a decimal resolution value, the value is transferred with no decimal indication. For instance, if the parameter P003 has the value 4.7, the value transferred via network will be 47.

These parameters are used only if the device is programmed in parameter P710 to use options 3 through 6. According to the selected option, it will be available up to six words for reading by the network master.

The first two input words are fixed.

NOTE!
The 0 (zero) value disables the word writing. The amount of input words, however, always remains the same as it was programmed in parameter P710.

P715 - DeviceNet Write Word #3

P716 - DeviceNet Write Word #4

P717 - DeviceNet Write Word #5

P718 - DeviceNet Write Word #6

Adjustable Range:	0 to 1199	**Factory Setting:**	0

Description:

It allow the user to program the content of output words (output: masters sends to the slave).

Using these parameters, it is possible to program the number of other parameter whose content shall be made available in the output area of the network master.

For instance, in case it is necessary to write the acceleration in the device, the value 100 must be programmed in some of these parameters, since the parameter P100 is the parameter where this information is programmed. Note that the written value of any parameter is represented with a 16-bit word. Even if the parameter has a decimal resolution value, the value is transferred with no decimal indication. For instance, if you want to set the parameter P100 with the value 5.0s, the value 50 should be written via network.

These parameters are used only if the device is programmed in parameter P710 to use options 3 through 6. According to the selected option, it will be available up to six words for writing by the network master.

NOTE!
The 0 (zero) value disables the word writing. The amount of input words, however, always remains the same as it was programmed in parameter P710.

P719 - DeviceNet Network Status

Adjustable Range:	0 = Offline 1 = OnLine, Not Conn 2 = OnLine, Connect. 3 = Connection Timed out 4 = Link Failure 5 = Auto-Baud	**Factory Setting:**	-
Properties:	ro		

Description:

It indicates the status of the DeviceNet network. The next table presents a brief description of those states.

Table 4.12: *P719 options*

Indication	Description
0 = Offline	Device without power supply or not online. Communication cannot be established.
1 = OnLine, Not Conn	Device online, but not connected. The slave has successfully completed the MacID verification procedure. This means that the configured baud rate is correct (or it has been detected correctly in case of autobaud) and that there are no other network nodes with the same address. However, there is no communication with the master yet in this stage.
2 = OnLine, Connect.	The device is operational and in normal conditions. The master has allocated a set of I/O type connections with the slave. In this stage the effective exchange of data by means of I/O type connections occurs.
3 = Connection Timed out	One or more I/O type connections have expired.
4 = Link Failure	It indicates that the slave was not able to enter the network due to addressing problems or due to the occurrence of bus off. Make sure the configured address is not used by other device, verify if the chosen baud rate is correct and make sure there are no installation problems.
5 = Auto-Baud	The equipment is executing the autobaud mechanism.

P720 - DeviceNet Master Status

Adjustable Range:	0 = Run 1 = Idle	**Factory Setting:**	-
Properties:	ro		

Description:

It indicates the DeviceNet network master status. It may be in operation mode (Run) or in configuration mode (Idle).

When in Run, reading and writing telegrams are processed normally and updated by the master. When in Idle, only the reading telegrams from the slaves are updated by the master. Writing, in this case, remains disabled.

When communication is disabled this parameter does not represent the actual state of the master.

Table 4.13: *P720 options*

Indication	Description
0 = Run	The master normally processes and updates reading and writing telegrams.
1 = Idle	The master only updates slave reading telegrams. Writing, in this case, is disabled.

5 OPERATION IN THE DEVICENET NETWORK

5.1 CYCLIC DATA

Cyclic data is the data normally used for status monitoring and equipment control. For DeviceNet protocol, the interface supports an I/O connection that allows communication up to 6 input words and 6 output words.

It is necessary the configuration to be made both at the slave and master, i.e., the same amount of input words and output words must be set in the CFW320 and in the master.

I/O connections have different formats, called I/O instances. One of these instances must be selected by the user through parameter P710. For the following examples, we suppose P710 is equal to Manufacturer Specific 3W.

5.1.1 Input words

The CFW320 frequency inverter has a reading area with 6 16-bit words available for cyclic data exchange of communication networks. The data available in the reading area (input) is sent to the network master.

To map an object in the reading area, it is necessary to configure the parameters P711 to P714 that make the data available in the reading words. These parameters must indicate the network addresses (Net Id) of the data to be transmitted in the respective reading words.

Example

The Table 5.1 on page 26 presents a configuration for DeviceNet considering the following parameters to be mapped:

- P680 Status Word.
- P681 Motor Speed.
- P003 Motor Current.

Table 5.1: *Example of reading words configuration.*

Mapped Parameter	Net Id	Size	Qty Mapped Words	Example Value
P680 Status Word	680	16bit	1	786 = 0312h
P681 Motor Speed	681	16bit	1	6500 (65.00 %)
P003 Motor Current	3	16bit	1	23 (2.3 A)

NOTE!
- Mapping of invalid parameters or not available will return zero value.
- The data is transmitted as an integer value, without the indication of the decimal places.
- To obtain the network address (Net Id) of the parameters and the number of decimal places, refer to the programming manual.

5.1.2 Output Words

The CFW320 frequency inverter has a writing area with 6 16-bit words available for cyclic data exchange of communication networks. The data available in the write area (output) is received from the network master.

To map an object in the writing area, it is necessary to configure the parameters P715 to P718 that make the data available in the writing words. These parameters must indicate the network addresses (Net Id) of the data to be transmitted in the respective writing words.

Exemplo

The Table 5.2 on page 27 presents a configuration for DeviceNet considering the following parameters to be mapped:

- P680 Control word.
- P681 Speed reference.
- P100 Acceleration time.

Table 5.2: *Example of configuration of the writing words.*

Mapped Parameter	Net Id	Size	Qty Mapped Words	Example Value
P680 Control word	684	16bit	1	83 = 0053h
P681 Speed reference	685	16bit	1	2500 (25.00) = 9C4h
P100 Acceleration time	100	16bit	1	100 (10.0) = 0064h

NOTE!

- Mapping of readonly parameters (status, diagnostics) or invalid parameters will have no effect.
- Parameters that have the property *Stopped*, when mapped on the writing words, are only changed when the motor is stopped.
- The parameters written using these words are not saved in non-volatile memory. Thus, if the equipment is turned off and back on, these parameters will return to their original value.
- The data is transmitted as an integer value, without the indication of the decimal places.
- To obtain the network address (Net Id) of the parameters, refer to the programming manual.

5.2 ACYCLIC DATA

In addition to the cyclic data, the interface also provides acyclic data via explicit messaging. 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 class, instance, and attribute to the desired parameter. The Section 6.10 on page 32 describe how to address the parameters for CFW320 frequency inverter.

5.3 EDS FILE

Each device on an DeviceNet 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 DeviceNet 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 CFW320 frequency inverter.

6 SUPPORTED OBJECT CLASSES

Any DeviceNet equipment is modeled as a set of objects. The objects are responsible for defining the function that each device will have. In other words, depending on the objects the device implements, it may be a communication adapter, an AC/DC drive, a photoelectric sensor, etc. Mandatory and optional objects are defined for each Device Profile. The CFW320 frequency inverter supports all mandatory classes defined for the AC/DC Device Profile. It also supports Manufacturer Specific classes. The following sections present detailed information about these classes.

6.1 IDENTITY CLASS (01H)

Provides general information about the device identity such as: VendorID, Product Name, Serial Number, etc. The following attributes are implemented:

Table 6.1: *: Identity Class instance attributes*

Attribute	Method	Name	Default	Description
1	GET	Vendor ID	355h	Manufacturer identifier
2	GET	Product Type	2h	Product Type
3	GET	Product Code		Product Code
4	GET	Vendor Revision		Vendor Revision
5	GET	Status		Device status
6	GET	Serial Number		Serial Number
7	GET	Product Name	CFW320	Product Name

6.2 MESSAGE ROUTER CLASS (02H)

Provides information on the explicit message router object. This class does not have any attribute implemented in the CFW320.

6.3 DEVICENET CLASS (03H)

This class is responsible for maintaining the configuration and the state of the physical connections of the DeviceNet node. The following attributes are implemented:

Table 6.2: *DeviceNet Class attributes*

Atributte	Method	Name	Min./Max	Default	Description
1	GET	Revision	1 - 65535		Revision of the DeviceNet Object Class Definition upon which the implementation is based.

Table 6.3: *DeviceNet Class instance attributes*

Atributte	Method	Name	Mín./Max	Default	Description
1	GET/SET	Mac ID	0 - 63	63	Node address.
2	GET/SET	Baud Rate	0 - 2	0	Communication baud rate
3	GET/SET	Bus-Off Interrupt	0 - 1	1	Bus-off reset
4	GET/SET	Bus-Off Counter	0 - 255		Bus-off counter
5	GET/SET	Allocation Information			Information about allocation byte

6.4 ASSEMBLY CLASS (04H)

This class is responsible for grouping several attributes in only one connection. Only the attribute Data (3) is implemented in the CFW320 (Table 6.4 on page 28).

Table 6.4: *Attributes of the instances of the Assembly class*

Atribute	Method	Name	Description
3	GET/SET	Data	Data contained in the assembly object

The Assembly class contains the following instances in the CFW320:

Table 6.5: *Instances of the Assembly class*

Instance	Size	Description
20	2 words	ODVA Basic Speed Control Output.
70	2 words	ODVA Basic Speed Control Output.
21	2 words	ODVA Extended Speed Control Output.
71	2 words	ODVA Extended Speed Control Output.
23	3 words	ODVA Extended Speed/Torque Control Output.
73	3 words	ODVA Extended Speed/Torque Control Output.
100	2 - 14 words	Manufacturer Specific Output
150	2 - 14 words	Manufacturer Specific Input

6.5 CLONNECTION CLASS (05H)

This class allocates and manages the internal resources associated with both I/O and Explicit Messaging Connections. The following methods are implemented:

6.5.1 Instance 1: Explicit Message

Table 6.6: *Classe Connection – Instância 1: Explicit Message*

Attribute	Method	Name	Description
1	GET	State	Object state
2	GET	Instance Type	I/O ou explicit
3	GET	transport Class trigger	Defines the connection behavior
4	GET	Produced Connection ID	CAN ID field for transmission
5	GET	Consumed Connection ID	CAN ID field value representing received msg
6	GET	Initial Comm. Charac.	Defines message groups related to this connection
7	GET	Produced Connection Size	Maximum size (bytes) of this transmission connection
8	GET	Consumed Connection Size	Maximum size (bytes) of this reception connection
9	GET/SET	Expected Packet Rate	Defines timing associated to this connection
12	GET	Watchdog Timeout Action	Action for inactivity/watchdog timeout
13	GET	Produced Connection Path Length	Number of bytes in the producer connection
14	GET	Produced Connection Path	Specifies the path of the data producer objects
15	GET	Consumed Connection Path Length	Number of bytes in the consumer connection
16	GET	Consumed Connection Path	Specifies the path of the data consumer objects
17	GET/SET	Production Inhibit Time	Defines the minimum time between new data production

6.5.2 Instance 2: Polled

Table 6.7: Classe Connection – Instância 2: Polled

Attribute	Method	Name	Description
1	GET	State	Object state
2	GET	Instance Type	I/O ou explicit
3	GET	transport Class trigger	Defines the connection behavior
4	GET	Produced Connection ID	CAN ID field for transmission
5	GET	Consumed Connection ID	CAN ID field value representing received msg
6	GET	Initial Comm. Charac.	Defines message groups related to this connection
7	GET	Produced Connection Size	Maximum size (bytes) of this transmission connection
8	GET	Consumed Connection Size	Maximum size (bytes) of this reception connection
9	GET/SET	Expected Packet Rate	Defines timing associated to this connection
12	GET	Watchdog Timeout Action	Action for inactivity/watchdog timeout
13	GET	Produced Connection Path Length	Number of bytes in the producer connection
14	GET	Produced Connection Path	Specifies the path of the data producer objects
15	GET	Consumed Connection Path Length	Number of bytes in the consumer connection
16	GET	Consumed Connection Path	Specifies the path of the data consumer objects
17	GET/SET	Production Inhibit Time	Defines the minimum time between new data production

6.5.3 Instance 4: Change of State/Cyclic

Table 6.8: Connection class – Instance 4: Change of State/Cyclic

Attribute	Method	Name	Description
1	GET	State	Object state
2	GET	Instance Type	I/O ou explicit
3	GET	transport Class trigger	Defines the connection behavior
4	GET	Produced Connection ID	CAN ID field for transmission
5	GET	Consumed Connection ID	CAN ID field value representing received msg
6	GET	Initial Comm. Charac.	Defines message groups related to this connection
7	GET	Produced Connection Size	Maximum size (bytes) of this transmission connection
8	GET	Consumed Connection Size	Maximum size (bytes) of this reception connection
9	GET/SET	Expected Packet Rate	Defines timing associated to this connection
12	GET	Watchdog Timeout Action	Action for inactivity/watchdog timeout
13	GET	Produced Connection Path Length	Number of bytes in the producer connection
14	GET	Produced Connection Path	Specifies the path of the data producer objects
15	GET	Consumed Connection Path Length	Number of bytes in the consumer connection
16	GET	Consumed Connection Path	Specifies the path of the data consumer objects
17	GET/SET	Production Inhibit Time	Defines the minimum time between new data production

6.6 MOTOR DATA CLASS (28H)

This class stores the information on the motor connected to the product. The following attributes have been implemented:

Table 6.9: Motor Data Class attributes

Attribute	Method	Name	Min/Max	Description
1	GET	Revision	1 - 65535	Revision of the Motor Data Object Class Definition upon which the implementation is based
2	GET	Max Instance	1 - 65535	Maximum instance number

Table 6.10: Motor Data Class instance attributes

Attribute	Method	Name	Min/Max	Unit	Default	Description
3	Get	Motor Type	0 - 10	-	7	0 = Non Standard Motor 1 = PM DC Motor 2 = FC DC Motor 3 = PM Synchronous Motor 4 = FC Synchronous Motor 5 = Switched Reluctance Motor 6 = Wound Rotor Induction Motor 7 = Squirrel Cage Induction Motor 8 = Stepper Motor 9 = Sinusoidal PM BL Motor 10 = Trapezoidal PM BL Motor
6	Get/Set	Rated Current	0-999.9	100mA		Nominal current
7	Get/Set	rated Voltage	0-600	V		Nominal voltage

6.7 MOTOR DATA CLASS (29H)

Responsible for modeling the drive management functions. The following attributes have been implemented:

Table 6.11: Control Supervisor Class attributes

Attribute	Method	Name	Min/Max	Description
1	GET	Revision	1 - 65535	Revision of the Control Supervisor Object Class Definition upon which the implementation is based
2	GET	Max Instance		Maximum instance number

Table 6.12: Atributos da instância da classe Control Supervisor

Attribute	Method	Name	Mín/Max	Default	Description
3	Get/Set	Run 1	0 - 1	-	Run Fwd
4	Get/Set	Run 2	0 - 1	-	Run Rev
5	Get/Set	NetCtrl	0 - 1	0	0 = Local control 1 = Remote control
6	Get	State	0 - 7	-	0 = Vendor specific 1 = Startup 2 = Not Ready 3 = Ready 4 = Enable 5 = Stopping 6 = Fault Stop 7 = Fault
7	Get	Running 1	0 - 1	0	0 = Other state 1 = (Enabled e Run1) or (Stopping e Running1) or (Fault Stop e Running1)
8	Get	Running 2	0 - 1	0	0 = Other state 1 = (Enabled e Run2) or (Stopping e Running2) or (Fault Stop e Running2)
9	Get	Ready	0 - 1	0	0 = Outro estado 1 = Ready or Enabled or Stopping
10	Get	Faulted	0 - 1	0	0 = No error 1 = Error
11	Get	Warning	0 - 1	0	0 = No warnings
12	Get/Set	Fault Reset	0 - 1	0	0 = No action 0 -> 1 = Error reset
15	Get	Ctrl from Net	0 - 1	0	0 = Local control 1 = Remote control

6.8 AC/DC DRIVE CLASS (2AH)

This class is responsible for modeling the management functions of the drive. The following attributes have been implemented: contains specific information of an AC/DC Drive such as operation mode, speed and torque ranges.

Table 6.13: AC/DC Drive Class attributes

Attribute	Method	Name	Min/Max	Description
1	GET	Revision	1 - 65535	Revision of the AC/DC Drive Object Class Definition upon which the implementation is based
2	GET	Max Instance		Maximum instance number

Table 6.14: AC/DC Drive Class instance attributes

Attribute	Method	Name	Mín/Max	Default	Description
4	Get/Set	NetRef 2	0 - 1	0	0 = Local reference 1 = Remote reference
6	Get	DriveMode	1 - 2	-	1 = Speed control (open loop) 2 = Speed control (closed loop)
7	Get	Speed Actual	0 - 9999		Actual speed (best approximation)
8	Get/Set	Speed Ref	0 - 9999	0	Speed reference

NOTE!
The CFW320 will work in speed mode independently of the content of the DriveMode attribute.

6.9 MESSAGE ROUTER CLASS (2BH)

This class is responsible for managing the reception of acknowledgment messages.

Table 6.15: Acknowledge Handler Class instance attributes

Attribute	Method	Name
1	GET/Set	Acknowledge Timer
2	GET	Retry Limit
3	GET	COS Production Connection Instance

6.10 MANUFACTURER APECIFIC CLASSES

The Manufacturer Specific Classes are used for mapping all CFW320 parameters. These classes allow the user to read from and write to any parameter through the network. The Manufacturer Specific Classes use DeviceNet explicit messages. There are separate ranges for each group of parameters, as presented in Table 6.16 on page 32:

Table 6.16: Manufacturer Specific Classes

Class	Name	Range
Class 100 (64h)	VENDOR CLASS F1	Parameters 000 - 099
Class 101 (65h)	VENDOR CLASS F2	Parameters 100 - 199
Class 102 (66h)	VENDOR CLASS F3	Parameters 200 - 299
Class 103 (67h)	VENDOR CLASS F4	Parameters 300 - 399
Class 104 (68h)	VENDOR CLASS F5	Parameters 400 - 499
Class 105 (69h)	VENDOR CLASS F6	Parameters 500 - 599
Class 106 (6Ah)	VENDOR CLASS F7	Parameters 620 - 699
Class 107 (6Bh)	VENDOR CLASS F8	Parameters 700 - 799
Class 108 (6Ch)	VENDOR CLASS F9	Parameters 800 - 899
Class 109 (6Dh)	VENDOR CLASS F10	Parameters 900 - 999
Class 110 (6Eh)	VENDOR CLASS F11	Parameters 1000 - 1099
Class 111 (6Fh)	VENDOR CLASS F12	Parameters 1100 - 1199

Table 6.17: Parameters of the Manufacturer Specific classes

Parameter	Class	Instance	Attribute
P0000	Class 100 (64h)	1	100
P0001	Class 100 (64h)	1	101
P0002	Class 100 (64h)	1	102
...	...	...	...
P0100	Class 101 (65h)	1	100
P0101	Class 101 (65h)	1	101
P0102	Class 101 (65h)	1	102
...	...	...	...
P0200	Class 102 (66h)	1	100
P0201	Class 102 (66h)	1	101
P0202	Class 102 (66h)	1	102
...	...	...	...
P0300	Class 103 (67h)	1	100
P0301	Class 103 (67h)	1	101
P0302	Class 103 (67h)	1	102
...	...	...	...

NOTE!

- The CFW320 uses only instance 1 for Manufacturer Specific Classes.
- In order to access the parameters through the Manufacturer Specific Classes, add the value 100 to the last two digits of any parameter. This new resulting number is known as attribute.

For instance:

Parameter 23: class 64h, instance 1, attribute 123. This path gives access to P023.
Parameter 100: class 65h, instance 1, attribute 100. This path gives access to P100.
Parameter 202: class 66h, instance 1, attribute 102. This path gives access to P202.

7 STARTUP GUIDE

The main steps to start up the CFW320 frequency inverter in DeviceNet network are described below. These steps represent an example of use. Check out the specific chapters for details on the indicated steps.

7.1 INSTALLING THE ACCESSORY

1. Install the communication accessory, as indicated in the installation guide supplied with the accessory.
2. Observe the content of parameter P028. Check if the module was recognized. The detection is done automatically and does not require the user's intervention.
3. Connect the cable to the accessory, considering the recommended instructions in network installation, as described in Section 3 on page 10:
   - Use shielded cable.
   - Properly ground network equipment.
   - Avoid laying communication cables next to power cables.

7.2 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 P222, P226, P227 and P228.
3. Configure communication parameters, such as protocol, address and baudrate in P700, P701 and P702.

   item Program the desired action for the equipment in case of communication fault in parameter P313.
4. Define which data will be read and written at frequency inverter CFW320 using Section 5 on page 26.

7.3 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[2] to the list of devices in the network configuration tool.
2. Select CFW320 frequency inverter 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 the quantity of I/O data communicated between master and slave, as well as the transmission method of these data. The DeviceNet protocol defines different methods of dada exchange, seeing that the module supports the following methods:

   Polled: communication method in which the master sends a telegram to each of the slaves of its list (*scan list*). As soon as it receives the request, the slave immediately answers the request of the master. This process is repeated until all slaves are polled, restarting the cycle.

   Change of State: communication method in which the data exchange between master and slave only occurs when there are changes in the values monitored/controlled up to a certain time limit. When this limit is reached, the transmission and reception will take place even if changes have not occurred.

   Cyclic: another communication method very similar to the previous one. The only difference is the production and consumption of messages. In this type of communication, every data exchange occurs at regular time intervals, no matter if they have been changed or not.

Once configured, the network status P719 indicates OnLine,Conn and the master status P720 indicates Run. It is in this condition that cyclic data exchange effectively occurs between the slave and the master of the network.

[2] 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 CFW320 frequency inverter.

7.4 COMMUNICATION STATUS

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

- The parameters P719 and P720 indicate the status of communication between the device and the network master.

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

7.5 OPERATION USING PROCESS DATA

Once the communication is established, the data mapped in the I/O area is automatically updated between master and slave.

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

7.6 ACCESS TO PARAMETERS – ACYCLIC MESSAGES

Besides the I/O data (cyclic) communication, the DeviceNet protocol also defines a kind of acyclic telegram (*explicit messages*), 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 *explicit messages*. The Section 5.2 on page 27 how to address the parameters of the frequency inverter CFW320 via acyclic messages.

8 QUICK REFERENCE OF ALARMS AND FAULTS

Fault / Alarm	Description	Possible Causes
F032 Comm. Plug-in module comunication Lost	Main control cannot establish the communication link with the communication acccessory.	■ Accessory damaged. ■ Poor connection of the accessory. ■ Problem in the identification of the accessory; refer to P028.
A133 No Power Supply on the CAN Interface	It indicates that the CAN interface has no power supply between pins 25 and 29 of the connector.	■ Measure if there is voltage within the allowed range between pins 25 and 29 of the CAN interface connector. ■ Check if the power supply cables are not misconnected or inverted. ■ Check for contact problems on the cable or connector of the CAN interface.
A134 Bus Off	Bus off error detected on the CAN interface.	■ Check for short circuit on the CAN circuit transmission cable. ■ Check if the cables are not misconnected or inverted. ■ Check if all the network devices use the same baud rate. ■ Check if the termination resistors with the right specification were installed only at the end of the main bus. ■ Check if the CAN network was properly installed.
A135 Node Guarding/ Heartbeat	CANopen communication error control detected communication error using the guarding mechanism.	■ Check the times set on the master and on the slave for message exchange. In order to prevent problems due to transmission delays and time counting, it is recommended that the values set for error detection by the slave be multiples of the times set for message exchange on the master. ■ Check if the master is sending the guarding telegrams in the time set. ■ Check problems in the communication that may cause missing telegrams or transmission delays.
A136 Idle Master	Alarm indicates that the DeviceNet network master is in Idle mode.	■ Set the switch that controls the master operation of the master for Run or the corresponding bit on the configuration word of the master software. If further information is needed, refer to the documentation of the master used.
A137 DeviceNet Connection Timeout	Alarm that indicates that one or more DeviceNet connections timed out.	■ Check the network master status. ■ Check network installation, broken cable or fault/poor contact on the connections with the network.
F233 No Power Supply on the CAN Interface	It indicates that the CAN interface has no power supply between pins V(-) and V(+) of the connector.	■ Measure if there is voltage within the allowed range between pins V(-) and V(+) of the CAN interface connector. ■ Check if the power supply cables are not misconnected or inverted. ■ Check for contact problems on the cable or connector of the CAN interface.
F234 Bus Off	Bus off error detected on the CAN interface.	■ Check for short circuit on the CAN circuit transmission cable. ■ Check if the cables are not misconnected or inverted. ■ Check if all the network devices use the same baud rate. ■ Check if the termination resistors with the right specification were installed only at the end of the main bus. ■ Check if the CAN network was properly installed.
F235 Node Guarding/ Heartbeat	CANopen communication error control detected communication error using the guarding mechanism.	■ Check the times set on the master and on the slave for message exchange. In order to prevent problems due to transmission delays and time counting, it is recommended that the values set for error detection by the slave be multiples of the times set for message exchange on the master. ■ Check if the master is sending the guarding telegrams in the time set. ■ Check problems in the communication that may cause missing telegrams or transmission delays.
F236 Idle Master	Fault indicates that the DeviceNet network master is in Idle mode.	■ Set the switch that controls the master operation for Run or the corresponding bit on the configuration word of the master software. If further information is needed, refer to the documentation of the master used.
F237 DeviceNet Connection Timeout	Fault that indicates that one or more DeviceNet connections timed out.	■ Check the network master status. ■ Check network installation, broken cable or fault/poor contact on the connections with the network.

Fault and alarm operation:

- Faults operate by indicating their occurrence on the HMI, in the frequency inverter status word (P006), in the present fault parameter (P049) and disabling the motor. They can only be reset with a reset command or de-energizing the frequency inverter.
- Alarms operate by indicating their occurrence on the HMI and in the present alarm parameter (P048). They are automatically reset when the alarm condition ceases existing.

WEG Drives & Controls - Automation 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