CFW-08 Frequency Inverter Communication Manual DeviceNet Slave

03/2006

Series: CFW-08
Software: version 4.6X
0899.5336 E/1

Important

This complementary manual describes the functions of the DeviceNet protocol for the CFW-08 Variable Frequency Drive. The CFW-08 drives with support to this protocol shall have in their product code the version "A4" of the control board as described in the following example:

MOD.: CFW080040B2024P0**A4**Z

The information described in the CFW-08 user's guide for version "A1" of the control board (CFW-08 Plus) is also valid for version "A4", however, particular attention is needed for the following modifications:

- Incorporation of the DeviceNet communication protocol.
- Exclusion of serial communication protocols (WEG protocol, Modbus-RTU and serial remote keypad).

A detailed description of these modifications is presented in the following sections.

Contents

List of Tables

List of Figures

About this manual

This manual describes the operation of the DeviceNet protocol for the CFW-08 Variable Frequency Drive. This manual shall be used along with the CFW-08 User's Guide.

Abbreviations and definitions

ADR Automatic Device Replacement
ASCII American Standard Code for Information Interchange
CAN Controller Area Network
PLC Programmable Logic Controller
MS Module Status
NS Network Status
ODVA Open DeviceNet Vendor Association

Numerical Representation
Decimal numbers are represented as digits without suffix (10). Hexadecimal numbers are represented with the 'h' letter after the number (10h).

Documents

The DeviceNet protocol for the CFW-08 has been developed based on the following specifications and documents:

Document	***Version***	***Source***
DeviceNet Volume I DeviceNet Communication Model and Protocol	2.0	ODVA
DeviceNet Volume II DeviceNet Device Profiles and Object Library	2.0	ODVA

Table 1: DeviceNet technical documentation

Refer to ODVA, which is the organization that currently maintain, publish and update the information related to the DeviceNet protocol, in order to obtain this documentation.

1 Installation

The DeviceNet network, as well as other industrial communication networks, demands special installation precautions so that communication error rate is kept as low as possible. Special attention is required because the industrial environment is usually exposed to extreme and harsh conditions such as electromagnetic interference. The installation must follow the general recommendations of any communication network: do not install data communications cables in a conduit or raceway that contains power conductors, proper size and install a good grounding system, and use only cables and connectors of good quality. The following sections provide basic recommendations for the installation of the CFW-08 variable frequency drive in a DeviceNet network.

1.1 Communication kit

In order to use the CFW-08 variable frequency drive in a DeviceNet network, it is necessary to install the KFB-DN-CFW-08 kit (WEG item 417118222). This kit is composed of the DeviceNet communication module, which contains the buttons and display as in the standard product keypad, a five-way connector, and status/error indication LEDs (refer to item 1.2). The kit is also supplied with a leaflet containing detailed instruction on how to install and remove the communication module. In case the drive comes with the communication board already installed, read this manual and carefully follow all of the instructions herein for proper configuration and operation of the device on the network.

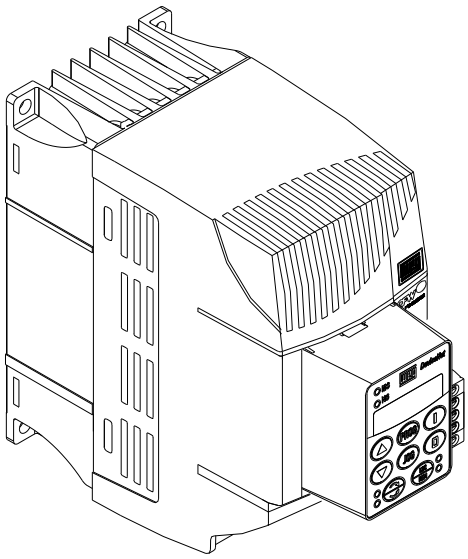

Figure 1: KFB-DN-CFW-08 kit installed on the drive.

1.2 Network connection

The CFW-08 DeviceNet network interface is available at connector XC14 of the communication module. Figure 2 shows where the connector is located on the communication module and Table 2 describes the function of each pin of the connector XC14.

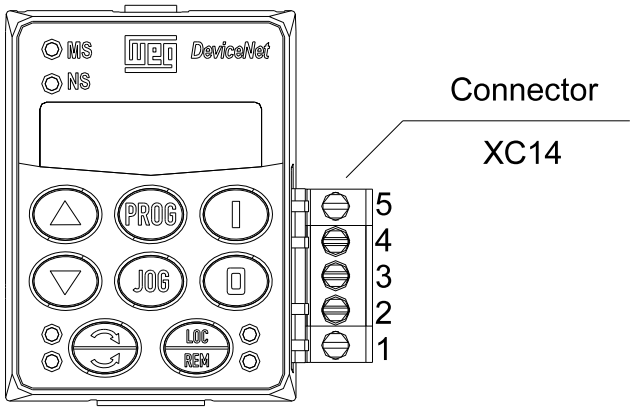

Figure 2: DeviceNet communication module.

Conector XC14			
5	V+	red	**power**
4	CAN_H	white	**signal**
3	Shield	-	
2	CAN_L	blue	**signal**
1	V-	black	**power**

Table 2: Pinout of connector XC14.

Make sure to connect the communication signals according to table 2. Each connector terminal of the several devices connected to the network must share the same signal (V- con V-, CAN_L con CAN_L, etc.). The cable shield (terminal number 3) must not be forgotten. This pin is connected to the internal ground through a parallel RC circuit.

NOTE!
Make sure to connect terminal number 5 (GND) of the connector XC1 to the grounding point (PE) located at the product heat sink.

This connection configuration has been designed aiming to allow the removal of any device from the network without causing problems to the remaining nodes. The addition/removal of nodes to/from the network must be totally transparent.

1.3 Power supply

In order to supply power to the circuit responsible for the CFW-08 DeviceNet communication, connect a power supply to terminals 1 and 5 of the network connector. In order to avoid problems with common mode voltages, ground the network to only one point (V-) and share the supply voltage (V+) between all network devices, as presented in figure 4. If you use a single power supply, add the current requirements of all devices (the consumption of each device shall be known previously) drawing power from the network. This is the minimum nameplate current rating that the power supply should have. If you use more than one power supply, all of them shall share the same reference point (ground). The position of the power supply on the installation affects the load distribution over the network, i.e., it is necessary to balance the output current of each power supply on the network.

There is no specific type of power supply to be used here. Any 24V power supply may be used, since it is able to provide the current drawn by the devices connected to the network and complies with the requirements listed below. Nevertheless, several manufacturers provide power supplies certified by ODVA. For further information, refer to the product catalog available at ODVA[1].

The main requirements for these power supplies are listed below:

- Current capability according to the requirements of all installed devices, respecting the limits of the cable used on the installation.
- Isolated DC output (from the AC input).
- Overcurrent protection.
- Good line regulation 24V +/- 4%.

In addition, it is recommended to install fuses in each network segment fed by the power supply.

Further details about this subject are available in chapter 10 of the document *DeviceNet Volume I*.

Table 3 shows the voltage and current levels required by the CFW-08 drive.

	Minimum	***Maximum***	
Voltage (Vcc)	11	30	24 (recommended)
Current (mA)	20	55	35 (average)

Table 3: Information for the network power supply

1.4 Cables and termination

It is highly recommended to use shielded cable with four wires - a pair for pins 2 and 4 (CAN_L and CAN_H) and another pair for pins 1 and 5 (V- and V+). Figure 3 shows the cable connection according to the pinout and color codes presented in table 2.

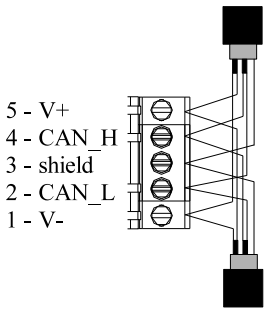

Figure 3: Typical connection of a DeviceNet cable

To interconnect several network nodes, it is recommended to attach them directly to the main communication bus (trunk-line). Nevertheless, drop-lines are allowed. Table 4 presents

[1] http://www.odva.org

the limits for the drop-lines length as a function of the communication baud rate. During network installation, do not install communication cables in a conduit or raceway that contains power conductors in order to avoid electromagnetic interference and consequently to reduce the occurrence of communication errors. Ground all devices connected to the network to the same ground reference in order to avoid ground loop current due to transient differences in ground potentials. The CFW-08 has an internal parallel RC circuit that avoids the ground loop current problem by interconnecting terminal number 3 of the DeviceNet connector to the local ground.

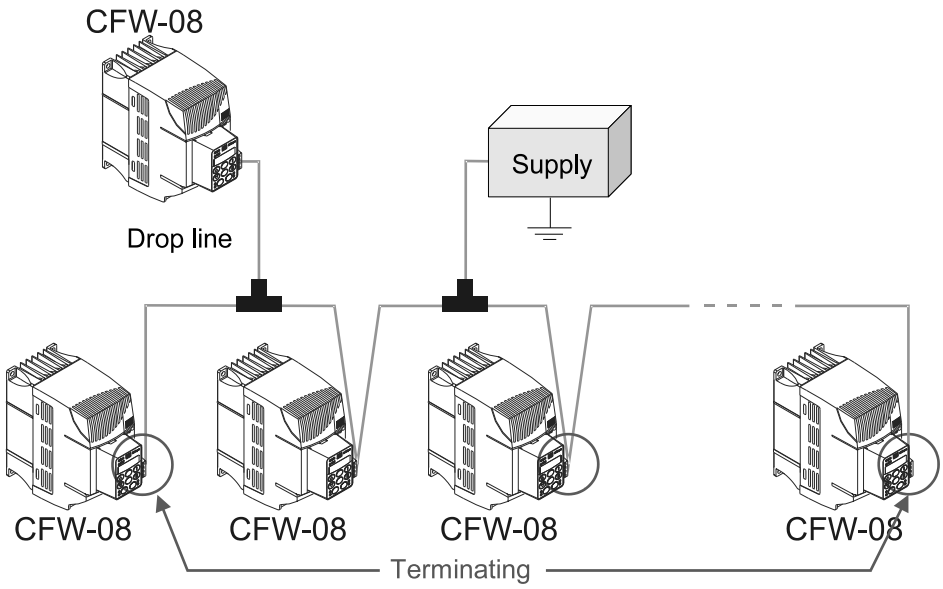

Figure 4: DeviceNet network

The cable used for interconnecting signals CAN_L and CAN_H must have a characteristic impedance of approximately 120Ω. The maximum cable length is a function of the communication baud rate, as presented in table 4.

Baud rate	***Network length (maximum)***	***Drop-line***	
		Maximum	*Cumulative*
125kbps	500m	6m	156m
250kbps	250m		78m
500kbps	100m		39m

Table 4: Network length x Communication baud rate

Both network ends must be fitted with a termination resistor in order to avoid signal reflections. Lack of the termination resistor may cause periodic errors on the network. Observe that the termination resistor must be connected directly across the CAN_H and CAN_L wires and not to the device itself. The resistor must have the following characteristics:

- 121Ω
- 0,25W
- 1% de tolerance

1.5 Configuration file

Any DeviceNet equipment must be previously registered at the network configuration software in order to work properly. To add the CFW-08 to the list of registered devices, use the ASCII file with an EDS extension that is provided in the CD-ROM that comes with the product or download the file from the WEG website[2]. This file contains instructions used by the network master to communicate with the slave. Each network configuration software has a different registration process. Therefore, refer to the software documentation before attempting to register this device.

NOTE!
The user must not modify the EDS file.

[2] http://www.weg.net

2 Description of the main CFW-08 characteristics

The CFW-08 variable frequency drive functions as a slave/server[3] in a DeviceNet network, i.e., the drive receives requests (output data) from a master[4], processes the data from these requests, and sends the data back to the master (input data). This is a cyclic process that takes place while the network is active.

The CFW-08 is not able to directly communicate with other slaves by using the peer-to-peer architecture. The network must have a master in charge of managing the communication. Therefore, the CFW-08 works as a Group 2 Only Server[5] device. In other words, the CFW-08 does not support the services related to the Unconnected Message Manager (UCMM).

The CFW-08 drive follows the device profile of an AC/DC Drive (AC/DC Device Profile). The device profile specifies the format of the data exchanged with the master and represents the operation interface with the drive.

2.1 Supported message types

The DeviceNet protocol defines two message types for the operation and monitoring of devices:

I/O : Synchronous message that provides a dedicated communication path for exchanging priority data between a producer and one or more consumers. They can still be subdivided in different data exchange methods such as: Polled, Bit-Strobe, Change of State, Cyclic and Multicast.

Explicit : General-purpose message type used mainly for non-priority asynchronous tasks such as node configuration and device setting.

The CFW-08 supports both message types: explicit messages are mainly used for drive configuration through the network, while I/O messages are mainly used for exchanging data related to the device operation. The CFW-08 supports the following data exchange methods: Polled, Change of State and Cyclic. The drive can produce data using one or more of these message types. The user should evaluate the options and use the most efficient method for its particular application.

2.2 Status/error indication

The indication of the device status/error on the DeviceNet network is made in two different ways: through display messages and through two bicolor indication LEDs - MS (Module Status) and NS (Network Status) - located in the product keypad, as presented in figure 5.

The **MS** bicolor LED indicates the status of the device:

[3] Also known as node.
[4] The master is usually composed of a scanner module installed in a PLC.
[5] This and other terms are explained in the Glossary.

Status	*Description*
Off	No power supply.
Solid green	Operational.
Flashing red/green	Performing self-diagnosis at power-up.

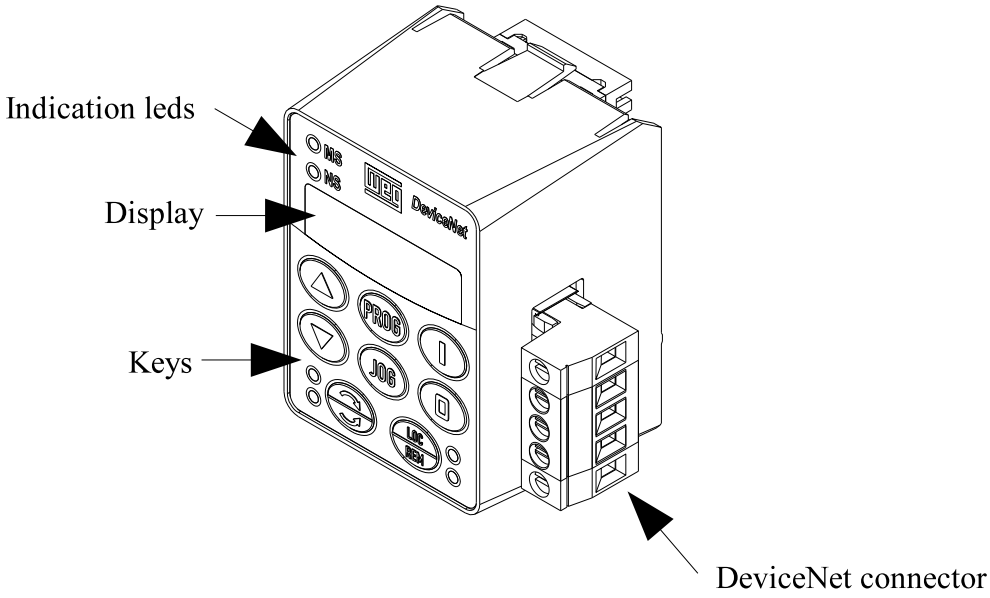

Figure 5: Details of the communication module.

On the other hand, the **NS** bicolor LED indicates the status of the DeviceNet network:

Status	*Description*
Off	No power supply or offline. Communication has not been established.
Flashing green	Device is online but not connected. Slave has successfully completed the Mac ID verification procedure. This means that the communication baud rate is correct (or has been correctly detected, if the auto-baud function is being used) and that there are no other network nodes configured with the same address. However, at this stage, the communication was still not established with the master.
Solid green	Device is operational and operating under normal conditions. Master has allocated a set of I/O connections for the slave. At this stage, data is effectively exchanged by using I/O messages.
Flashing red	One or more I/O connection have expired
Solid red	Indicates that the slave could not connect to the network due to addressing problems or bus-off error. Check if another network device is already using the configured address and if the communication baud rate is properly set.
Flashing red/green	Device is performing self-diagnosis. Performed at the device power-up.

In addition to the **MS** and **NS** indication LEDs, another information source for the status of the device/network are the messages shown at the keypad display. These messages are used for indicating the DeviceNet protocol errors. The four main errors that may appear are:

E33 : CAN controller is not powered up. Check the power supply of the DeviceNet network.

E34 : Indicates that the CAN controller reached the bus-off state. Make sure the slave communication baud rate matches the master baud rate.

E36 : Indicates that the network master is in the idle state.

E37 : This error occurs when one or more allocated I/O connections timed-out.

Lastly, there are specific parameters on the device that indicate its status and error condition. Further information is available in subsections 4.1.1 and 4.1.2.

3 I/O Messages Format

3.1 Control and monitoring

Each I/O instance is composed of control and monitoring bits that cover the most common operations. The CFW-08 supports instances 20/70 and 21/71 defined by ODVA. In addition to these instances, the CFW-08 also supports the WEG-specific instances 100/150. The choice of which instances to use is made at parameter P710. The user should consider the application characteristics when deciding about the instances.

3.1.1 Monitoring options

The following information is available to the user through the network[6]:

- *Faulted* [bit 0, byte 0]: this bit is set when the drive is under a fault condition.
- *Warning* [bit 1, byte 0]: this bit has no function for the CFW-08. Therefore, it is always set to zero.
- *Running1(fwd)* [bit 2, byte 0]: this bit is set when the motor is running forward.
- *Running2(rev)* [bit 3, byte 0]: this bit is set when the motor is running reverse.
- *Ready* [bit 4, byte 0]: this bit is set it the drive is in one of the following states: Ready, Enabled or Stopping.
- *Ctrl from Net* [bit 5, byte 0]: this bit is set when the drive is controlled from the DeviceNet network.
- *Ref from Net* [bit 6, byte 0]: this bit is set when the speed reference is given through the DeviceNet network.
- *At Reference* [bit 7, byte 0]: this bit is set when the speed reference has been reached, i.e., when the difference between the actual speed and the speed reference is less than 1Hz.
- *Drive State* [byte 1]: indicates the status of the drive:

  0: Non-Existant

  1: Startup

  2: Not_Ready

  3: Ready

  4: Enabled

  5: Stopping

  6: Fault_Stop

  7: Faulted

- *Speed Actual* (*RPM*) [bytes 2 e 3]: word that contains the actual speed of the motor in RPM.

[6] Valid for instances 20/70 and 21/71.

3.1.2 Control options

The following control options are available to the drive through the DeviceNet network[7]:

- *Run Fwd* [bit 0, byte 0]: the drive runs forward when this bit is set to 1.
- *Run Rev* [bit 1, byte 0]: the drive runs reverse when this bit is set to 1.
- *Fault Reset* [bit 2, byte 0]: a positive transition (from 0 to 1) on this bit resets the drive when it is under a fault condition.
- *NetCtrl* [bit 5, byte 0]: the CFW-08 is controlled via DeviceNet network (bits 0,1 and 2) when this bit is set to 1.
- *NetRef* (*RPM*) [bit 6, byte 0]: the drive uses the network speed reference (bytes 2 and 3) when this bit is set to 1.
- *Speed Reference* bytes 2 e 3]: sets the motor speed reference in RPM.

3.2 Instances 20/70 (Polled, Change of State and Cyclic)

These instances, called Basic Speed, represent the simplest operation interface available for a device that complies with the AC/DC Device Profile. The data mapping is shown in table 5.

Monitoring (*Input*)

Instance	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 (byte low)							
	3	Speed Actual (byte high)							

Control (*Output*)

Instance	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 (byte low)							
	3	Speed Reference (byte high)							

Table 5: Instances 20/70 defined by ODVA

[7] Valid for instances 20/70 and 21/71.

3.3 Instances 21/71 (Polled, Change of State and Cyclic)

These instances, called Extended Speed, represent a more refined operation interface for a device that complies with the AC/DC Device Profile. The data mapping is shown in table 6.

Monitoring (*Input*)

Instance	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 (byte low)							
	3	Speed Actual (byte high)							

Control (*Output*)

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1	-							
	2	Speed Reference (byte low)							
	3	Speed Reference (byte high)							

Table 6: Instances 21/71 defined by ODVA

3.4 Instances 100/150 (Polled, Change of State and Cyclic)

These WEG-specific instances have been created to allow the user to control the drive, monitor its status, and set any CFW-08 parameters. The data mapping is shown in table 7.

Monitoring (*Input*)

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
150	0	Logical status (error code)							
	1	Error		Undervoltage	Local or Remote	JOG	Direction of Rotation	General Enable	Start/ Stop
	2	Speed Actual (byte low)							
	3	Speed Actual (byte high)							

Start/Stop : 0 = stopped and 1 = running.

General Enable : 0 = disabled and 1 = enabled.

Direction of Rotation : 0 = reverse and 1 = forward.

JOG : 0 = Jog disabled and 1 = Jog enabled.

Local or Remote : 0 = local mode and 1 = remote mode.

Undervoltage : 0 = no undervoltage and 1 = undervoltage.

Error : 0 = no error and 1 = error.

Control (*Output*)

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
100	0	Error Reset			Local or Remoto	JOG	Direction of Rotation	General Enable	Start/ Stop
	1	-							
	2	Speed Reference (byte low)							
	3	Speed Reference (byte high)							

Table 7: Instances 100/150 defined by WEG

Start/Stop : 0 = stop and 1 = start.

General Enable : 0 = disable drive and 1 = enable drive.

Direction of Rotation : 0 = run forward and 1 = run reverse.

JOG : 0 = deactivate Jog function and 1 = activate Jog function.

Local or Remote : 0 = change the drive to local mode and 1 = change the drive to remote mode.

Error Reset : a positive transition (from 0 to 1) on this bit resets the CFW-08 when it is under a fault condition.

NOTE!
The drive will be offline during its reset.

4 DeviceNet-related parameters

The CFW-08 variable frequency drive has a set of parameters (described in next sections) that are used for configuring the device on the network and for monitoring its status.

Parameters that are not mentioned here do not have a direct relation with the DeviceNet communication, however, they may be important for the regular operation of the CFW-08 drive. Therefore, the user should be familiar with all drive's parameters since they may be used during the operation of the drive in the DeviceNet network. Refer to the CFW-08 user's guide for a complete list of the CFW-08 parameters.

4.1 New parameters

Parameters presented in this section are only available for the version "A4" of the control board.

4.1.1 P070 - Status of the CAN controller

This parameter provides information on the status of the device with respect to the CAN bus. This parameter indicates if the controller is working properly or if it is under a communication fault condition.

Range	***Default***	***Access***
0 = Communication is not enabled 1 = Auto-baud Running 2 = Error Active (No error) 3 = Warning 4 = Error Passive 5 = Bus-Off 6 = No power supply	-	read

These errors are a function of the number of invalid received/transmitted messages from/to the network. The error passive state, for instance, is reached when only one device is connected to the network (it transmits messages without receiving an acknowledgment from other devices). The bus-off state, for instance, may be reached when devices configured with different communication baud rates are connected to the same network or when installation problems are encountered (lack of termination resistor, etc.).

4.1.2 P081 - Status of the network master

Indicates the status of the network master.

Range	***Default***	***Access***
0 = Run 1 = Idle	-	read

4.1.3 P313 - Action for communication error

If a communication error occurs (broken cable, lost of the input power supply, etc.) while the CFW-08 is enabled and being controlled through the network, no command to the drive will work. Depending on the application, this may cause a catastrophic situation. In order to avoid any damage, it is possible to set at parameter P313 an action the drive will automatically take in case of a communication problem.

Range	***Default***	***Access***
0 = Stop (Ramp stop) 1 = General disable (Coast stop) 2 = No action 3 = Change to local mode	2	read/write

NOTE!
This action will be executed only if that specific command is enabled through the DeviceNet network.

4.1.4 P700 - CAN protocol

Parameter P700 allows selecting the application layer protocol for the CAN bus on the CFW-08. Option number '2' shall be selected to enable the drive to communicate in a DeviceNet network.

Range	***Default***	***Access***
0 = Disabled 1 = CANopen 2 = DeviceNet	0	read/write

NOTE!
The modification of the device application protocol will be valid only after the CFW-08 is reset.

4.1.5 P701 - CAN node address

This parameter sets the CFW-08 address (Mac ID) on the DeviceNet network.

Range	***Default***	***Access***
0-63	63	read/write

Each device on the network needs to have a different Mac ID. Therefore, the maximum number of devices on the network is 64. There is no reserved address, even if value '0' is frequently used for addressing the network master (DeviceNet scanner) and value '63' is often used for identifying new devices on the network.

NOTE!
The modification of the device address will be valid only after the CFW-08 is reset.

4.1.6 P702 - Communication baud rate

This parameter sets the communication baud rate for the device.

Range	*Default*	*Access*
0 = Auto-baud 1 = Auto-baud **2 = 500 kbit/s** **3 = 250 kbit/s** **4 = 125 kbit/s** 5 = Auto-baud 6 = Auto-baud 7 = Auto-baud 8 = Auto-baud	0	read/write

Set the same baud rate to all devices connected to the network to allow them to communicate. Remember also that the communication baud rate is limited to the cable length used on the installation (refer to table 4). Keep in mind that the communication must be active on the network, i.e., the master shall be exchanging messages with at least one device on the network, so that the auto-baud detection mechanism is operational.

NOTE!
The modification of the communication baud rate will be valid only after the CFW-08 is reset.

4.1.7 P703 - Bus-off reset

When the number of transmission/reception errors exceeds a certain limit, the device may enter into the bus-off state, where it stops accessing the network. Use parameter P703 to set the action CFW-08 shall take in case of a bus-off error: remain in the bus-off state or to automatically reset the error and restart the communication.

Range	*Default*	*Access*
0 = Remain in bus-off state 1 = Automatically reset	0	read/write

4.1.8 P710 - I/O instances

This parameter sets the I/O instances according to the user application. All instances supported by the CFW-08 have a two-word size (4 bytes). In such way, 4 input and 4 output bytes shall be reserved on the master memory, regardless of the communication mode used (Polled, Change of State or Cyclic). This configuration is usually performed through

the network configuration software. For additional information, please, refer to the controller documentation.

The description of each one of these instances is available in section 3.

Range	***Default***	***Access***
0 = Instance 20/70 (2 words I/O) 1 = Instance 21/71 (2 words I/O) 2 = Instance 100/150 (2 words I/O)	0	read/write

NOTE!
The modification of parameter P710 will be valid only after the CFW-08 is reset.

4.2 Modified parameters

The following parameters had modifications in their functions. The modifications are valid only for the version "A4" of the control board.

4.2.1 P220 - Local/remote selection source

Range	***Default***	***Access***
0 = Always local 1 = Always remote 2 = Keypad HMI-CFW08-P or HMI-CFW08-RP (default: local) 3 = Keypad HMI-CFW08-P or HMI-CFW08-RP (default: remote) 4 = DI2 to DI4 5 = DeviceNet (default: local) 6 = DeviceNet (default: remote)	2	read/write

4.2.2 P221 - Speed reference selection - local mode

Range	***Default***	***Access***
0 = Keys ▲ and ▼ HMI 1 = AI1 2,3 = AI2 4 = E.P. 5 = DeviceNet 6 = Multispeed 7 = Sum AI $\geq$ 0 8 = Sum AI	0	read/write

4.2.3 P222 - Speed reference selection - remote mode

Range	*Default*	*Access*
0 = Keys ▲ and ▼ HMI 1 = AI1 2,3 = AI2 4 = E.P. 5 = DeviceNet 6 = Multispeed 7 = Sum AI $\geq$ 0 8 = Sum AI	1	read/write

4.2.4 P229 - Command selection - local mode

Range	*Default*	*Access*
0 = Keypad HMI-CFW08-P or HMI-CFW08-RP 1 = Digital Input 2 = DeviceNet	0	read/write

4.2.5 P230 - Command selection - remote mode

Range	*Default*	*Access*
0 = Keypad HMI-CFW08-P or HMI-CFW08-RP 1 = Digital Input 2 = DeviceNet	1	read/write

4.2.6 P231 - Direction of rotation - local and remote mode

Range	*Default*	*Access*
0 = Forward 1 = Reverse 2 = Commands	2	read/write

5 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 CFW-08 drive supports all mandatory classes defined for the AC/DC Device Profile. In addition to them, the CFW-08 also supports the WEG Specific classes. The following sections present detailed information about these classes.

5.1 Identity class (01h)

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

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		Firmware version
5	Get	Status		Device status
6	Get	Serial Number		Serial number
7	Get	Product Name	CFW-08	Product name

Table 8: Identity class

5.2 Message Router class (02h)

Provides information on the explicit message router object. This class does not have any attribute implemented in the CFW-08.

5.3 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:

Attribute	***Method***	***Name***	***Min./Max***	***Default***	***Description***
1	Get/Set	Mac ID	0-63	63	Node address
2	Get/Set	Baud rate	0-2	0	Communication baud rate
4	Get/Set	Bus-Off Counter	0-255		Bus-off counter
5	Get	Allocation Information			Information about allocation byte

Table 9: DeviceNet class

5.4 Assembly class (04h)

This class is responsible for grouping several attributes in only one connection. Only the attribute Data (3) is implemented in the CFW-08 (table 10).

Attribute	*Method*	*Name*	*Description*
3	Get/Set	Data	Data contained in the assembly object

Table 10: Attributes of the instances of the Assembly class

The Assembly class contains the following instances in the CFW-08. For further information, please, refer to section 3

Instances	*Size*	*Description*
20	2 words	DeviceNet AC/DC Profile
21	2 words	DeviceNet AC/DC Profile
100	2 words	WEG specific
70	2 words	DeviceNet AC/DC Profile
71	2 words	DeviceNet AC/DC Profile
150	2 words	WEG specific

Table 11: Instances of the Assembly class

5.5 Connection class (05h)

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

5.5.1 Instance 1: *Explicit Message*

Attribute	***Method***	***Name***	***Description***
1	Get	State	Object state
2	Get	Instance Type	I/O or 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

Table 12: Connection class - Instance 1:*Explicit Message*

5.5.2 Instance 2: *Polled*

Attribute	*Method*	*Name*	*Description*
1	Get	State	Object state
2	Get	Instance Type	I/O or 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

Table 13: Connection class - Instance 2:*Polled*

5.5.3 Instance 4: *Change of State/Cyclic*

Attribute	***Method***	***Name***	***Description***
1	Get	State	Object state
2	Get	Instance Type	I/O or 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

Table 14: Connection class - Instance 4:*Change of State/Cyclic*

5.6 Motor Data class (28h)

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

Attribute	***Method***	***Name***	***Mín./Max***	***Unidade***	***Default***	***Description***
3	Get/Set	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	220	Nominal voltage

Table 15: Motor Data class

5.7 Control Supervisor class (29h)

Responsable por modelar funciones de control del *drive*. Entonces implementados los siguientes atributos:

Attribute	***Method***	***Name***	***Mín./Max***	***Default***	***Description***
3	Set	Run1	0-1		Run Fwd
4	Set	Run2	0-1		Run Rev
5	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 = Enabled 5 = Stopping 6 = Fault Stop 7 = Fault
7	Get	Running1	0-1	0	0 = Other state 1 = (Enabled and Run1) or (Stopping and Running1) or (Fault Stop and Running1)
8	Get	Running2	0-1	0	0 = Other state 1 = (Enabled and Run2) or (Stopping and Running2) or (Fault Stop and Running2)
9	Get	Ready	0-1	0	0 = Other state 1 = Ready or Enabled or Stopping
10	Get	Faulted	0-1	0	0 = No error 1 = Error
11	Get	Warning	0	0	0 = No warnings
12	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

Table 16: Control Supervisor class

5.8 AC/DC Drive class (2Ah)

This class is responsible for modeling the management functions of the drive. The following attributes have been implemented:

Attribute	*Method*	*Name*	*Mín./Max*	*Default*	*Description*
4	Get/Set	NetRef	0-1	0	0 = Local reference 1 = Remote reference
6	Get/Set	DriveMode	1-2	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

Table 17: AC/DC Drive class

NOTE!
The CFW-08 will work in speed mode (V/Hz or vector control) independently of the content of the DriveMode attribute.

5.9 Acknowledge Handler class (2Bh)

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

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

Table 18: Acknowledge Handler class

5.10 WEG classes

The WEG specific classes are used for mapping all CFW-08 parameters. These classes allow the user to read from and write to any drive parameter through the network. The WEG specific classes use DeviceNet explicit messages.

There are separate ranges for each group of parameters, as presented in table 19:

Range	***Class***	***Name***
Parameters 000-099	Class 100 (64h)	WEG_CLASS_F1
Parameters 100-199	Class 101 (65h)	WEG_CLASS_F2
Parameters 200-299	Class 102 (66h)	WEG_CLASS_F3
Parameters 300-399	Class 103 (67h)	WEG_CLASS_F4
Parameters 400-499	Class 104 (68h)	WEG_CLASS_F5
Parameters 500-599	Class 105 (69h)	WEG_CLASS_F6
Parámetros 600-699	Class 106 (6Ah)	WEG_CLASS_F7
Parameters 700-799	Class 107 (6Bh)	WEG_CLASS_F8

Table 19: WEG classes

Parameters	***Class***	***Instance***	***Attribute***
P000	Class 100 (64h)	1	100
P001	Class 100 (64h)	1	101
P002	Class 100 (64h)	1	102
⋮	⋮	⋮	⋮
P100	Class 101 (65h)	1	100
P101	Class 101 (65h)	1	101
P102	Class 101 (65h)	1	102
⋮	⋮	⋮	⋮
P200	Class 102 (66h)	1	100
P201	Class 102 (66h)	1	101
P202	Class 102 (66h)	1	102
⋮	⋮	⋮	⋮
P300	Class 103 (67h)	1	100
P301	Class 103 (67h)	1	101
P302	Class 103 (67h)	1	102
⋮	⋮	⋮	⋮

Table 20: Parameters of the WEG classes

NOTE!
The CFW-08 uses only instance 1 for WEG classes.

NOTE!
In order to access the parameters through the WEG classes, add the value 100 to the last two digits of any drive parameter. This new resulting number is known as attribute.
For instance:

Parameter 23 : class 64h, instance 1, attribute 123.

Parameter 100 : class 65h, instance 1, attribute 100.

Parameter 202 : class 66h, instance 1, attribute 102.

Parameter 432 : class 68h, instance 1, attribute 132.

6 Additional Function

6.1 Automatic Device Replacement function

This resource, when available at the network master, downloads and stores the configuration of each slave into the internal memory of the master (configuration recovery). This allows replacing a failed device with a spare from the factory with no additional reconfiguration.

The ADR also uses the auto-address recovery, which is the ability of the master to change a device's node address from '63' to any other intended. The DeviceNet protocol defines that all new devices shall be set to address '63'. If the master loses the communication with any node of its scan list, it will continually query the device's identity at address '63' and compares its attributes to the characteristics of the failed device. If the master finds an identical device at address '63', the original configured device parameters will be automatically transferred to the new device (automatic reconfiguration). The communication restarts just after the stored configuration is download to the replacement device resulting in a reduced downtime.

The CFW-08 is ready to use this function with no user intervention. To make this resource active, use the network programming and configuration software. Refer to the documentation of the DeviceNet scanner module.

NOTE!
To make this resource active, set the application protocol to DeviceNet (P700 = 2).

Glossary

Predefined Master/Slave Connection Set: set of connections that make the communication easier. Usually found in master/slave applications. Many of the steps for creating and configuring the Master/Slave Connection Set have been removed from the definition. Thus, the communication can be established with less network and device resources.

UCMM (Unconnected Message Manager):): this object is available in some DeviceNet equipment and is capable of establishing peer-to-peer communication without the support of a master.

Group 2 Only Server: slave (server) that does not have an UCMM object and shall use the Predefined Master/Slave Connection Set to establish the communication. A Group 2 Only Server can transmit and receive only those identifiers defined by the Predefined Master/Slave Connection Set. The CFW-08 drive functions as a Group 2 Only Server.

Group 2 Server: slave (server) that has a UCMM object and is able to establish peer-to-peer connections with other slaves.

Scan List: list of slaves controlled by the network master. Only the slaves enumerated in this list can communicate with the master by using I/O messages.

DeviceNet Master: device that gathers and shares information of the network slaves by using the scan list. The communication always starts with the master. The master function is usually performed by a scanner module in a PLC.

DeviceNet Slave: device that returns information for a master request. The CFW-08 functions as a slave on a DeviceNet network.

Mac ID: attribute that represents the node address on the network.

Polled: type of I/O message where the slave will receive "output" data from the master and respond to that immediately. This process is continuously repeated for all the slaves on the master's scan list.

Change of State: another communication mechanism, where the data is produced whenever the monitored values change, or at a base heartbeat rate (timer). When the timer expires, the data is transmitted and received even if it has not changed. The timer is configured at the network configuration software.

Cyclic: communication mechanism very similar to the before mentioned. The only difference regards the production and consumption of messages. With this type of I/O messaging the data exchange takes place at a precisely defined interval, even if the data has not changed. The time interval is also set at the network configuration software.

Assembly: class which function is to group attributes from multiple objects in only one connection.

AC/DC Device Profile: profile definition of an AC/DC device. Provides a standard interface for presenting the data to the user.