DeviceNet

CFW-11

Communication Manual
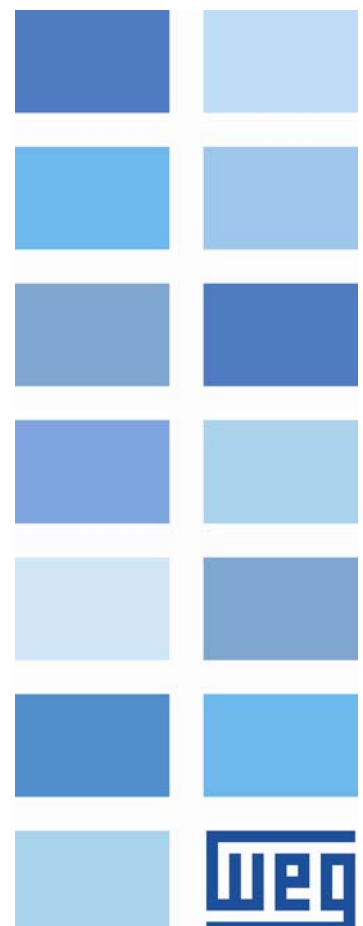
Language: English

DeviceNet Communication Manual

Summary

About this Manual

This manual provides the necessary information for the operation of the CFW-11 frequency inverter using the DeviceNet protocol. This manual must be used together with the CFW-11 user manual.

Abbreviations and Definitions

ASCII   American Standard Code for Information Interchange
CAN   Controller Area Network
CIP   Common Industrial Protocol
PLC   Programmable Logic Controller
HMI   Human-Machine Interface
ODVA   Open DeviceNet Vendor Association

Numerical Representation

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

Documents

The DeviceNet protocol for the CFW-11 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

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 Introduction to the DeviceNet Communication

In order to operate the CFW-11 frequency inverter in a DeviceNet network, it is necessary to know the manner this communication is performed. Therefore, this section brings a general description of the DeviceNet protocol operation, containing the functions used by the CFW-11. Refer to the DeviceNet documentation indicated above for a detailed description of the protocol.

1.1 CAN

DeviceNet is a network based on CAN, i.e., it uses CAN telegrams for exchanging data in the network.

The CAN protocol is a serial communication protocol that describes the services of layer 2 of the ISO/OSI model (data link layer)[1]. This layer defines the different types of telegrams (frames), the error detection method, the validation and arbitration of messages.

1.1.1 Data Frame

CAN network data is transmitted by means of a data frame. This frame type is composed mainly by an 11 bit[2] identifier (arbitration field), and by a data field that may contain up to 8 data bytes.

Identifier	8 data bytes							
11 bits	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7

1.1.2 Remote Frame

Besides the data frame, there is also the remote frame (RTR frame). This type of frame does not have a data field, but only the identifier. It works as a request, so that another network device transmits the desired data frame.

1.1.3 Access to the Network

Any device in a CAN network can make an attempt to transmit a frame to the network in a certain moment. If two devices try to access the network simultaneously, the one that sends the message with the highest priority will be able to transmit. The message priority is defined by the CAN frame identifier, the smaller the value of this identifier, the higher the message priority. The telegram with the identifier 0 (zero) is the one with the highest priority.

1.1.4 Error Control

The CAN specification defines several error control mechanisms, which makes the network very reliable and with a very low undetected transmission error rate. Every network device must be able to identify the occurrence of these errors, and to inform the other elements that an error was detected.

A CAN network device has internal counters that are incremented every time a transmission or reception error is detected, and are decremented when a telegram is successfully transmitted or received. If a considerable amount of errors occurs, the device can be led to the following states:

☑ *Warning*: when the counter exceeds a defined limit, the device enters the *warning* state, meaning the occurrence of a high error rate.
☑ *Error Passive*: when this value exceeds a higher limit, the device enters the *error passive* state, and it stops acting in the network when detecting that another device sent a telegram with an error.
☑ *Bus Off*: finally, we have the *buss off* state, in which the device will not send or receive telegrams any more.

---

[1] In the CAN protocol specification, the ISO11898 standard is referenced as the definition of the layer 1 of this model (physical layer).
[2] The CAN 2.0 specification defines two data frame types, standard (11 bit) and extended (29 bit). For the CFW-11 DeviceNet protocol, only the standard frames are accepted.

1.1.5 CAN and DeviceNet

Only the definition of how to detect errors, create and transmit a frame, are not enough to define a meaning for the data transmitted via the network. It is necessary to have a specification that indicates how the identifier and the data must be assembled and how the information must be exchanged. Thus, the network elements can interpret the transmitted data correctly. In that sense, the DeviceNet specification defines exactly how to exchange data among the devices and how every one must interpret these data.

There are several other protocols based on CAN, as CANopen, J1939, etc., which do also use CAN frames for the communication. However, those protocols cannot be used together in the same network.

1.2 DeviceNet

The next sections present, in a succinct form, the DeviceNet protocol.

1.2.1 Introduction

Introduced in 1994, DeviceNet is an implementation of the Common Industrial Protocol (CIP) for industrial communication networks. Developed originally by Allen-Bradley, it had its technology transferred to the ODVA that, since then, keeps, publishes and promotes DeviceNet and other networks based on the CIP[3] protocol. Furthermore, it uses the Controller Area Network (CAN) protocol for the data link and access to the medium, layers 2 and 1 of the OSI/ISO model, respectively.

Used mainly for the connection of industrial controllers and I/O devices, the protocol follows the model producer-consumer, supports multiple communication modes and has priority between messages.

It is a system that can be configured to operate in master-slave architecture as well as in a distributed point-to-point architecture. Besides, it defines two kinds of messages, I/O (process data) and explicit (configuration and parameter setting). It also has mechanisms to detect duplicated addresses and for node isolation in case of critical faults.

A DeviceNet network can have up to 64 devices, addressed from 0 to 63. Any of them can be used. There is no restriction, although the 63 should be avoided because it is usually used for commissioning.

1.2.2 Physical Layer

DeviceNet uses a network topology of the trunk/derivation type that allows the signal wiring as well as the power wiring to be present in the same cable. This power is supplied by a power supply connected directly to the network, which feeds the CAN transceivers of the nodes, and has the following characteristics:

☑ 24Vdc;
☑ DC output isolated from the AC input;
☑ Current capacity compatible with the installed equipment.

The used Baud rate depends on the size (cable length) of the network, as showed in the table below.

Table 0.1 - Network size x Baud rate

Baud Rate	Network Size	Derivation	
		Maximum	Total
125kbps	500m		156m
250kbps	250m	6m	78m
500kbps	100m		39m

In order to avoid reflections in the line, it is recommended the installation of termination resistors at the line extremes, because the absence of them may cause intermittent errors. This resistor must have the following characteristics, according to the protocol specification:

---

[3] CIP actually represents a family of networks. DeviceNet, EtherNet/IP and ControlNet use CIP in the application layer. The difference among them is primordially in the data link and physical layers.

☑ 121Ω;
☑ 0,25W;
☑ 1% tolerance.

For DeviceNet, several types of connectors can be used, sealed ones as well as open ones. The definition of the type to be used depends on the application and on the equipment operation environment. The CFW-11 uses a 5 wire plug-in connector, and its pinout is showed in the section 2. For a complete description of the connectors used with DeviceNet, consult the protocol specification.

1.2.3 Data Link Layer

The DeviceNet data link layer is defined by the CAN specification, which defines two possible states; dominant (logic level 0) and recessive (logic level 1). A node can bring the network to the dominant state if it transmits any information. Thus, the bus will only be in the recessive state if there where no transmitting nodes in the dominant state.

CAN uses the CSMA/NBA to access the physical medium. This means that a node, before transmitting, must verify if the bus is free. In case it is, then the node can initiate the transmission of its telegram. In case it is not, then the node must await. If more than one node access the network simultaneously, a priority mechanism takes action to decide which one will have priority over the others. This mechanism is not destructive, i.e., the message is preserved even if there is a collision between two or more telegrams.

CAN defines four types of telegrams (data, remote, overload and error). Among them, DeviceNet uses only the data frame and the error frame.

Data is moved using the data frame. This frame structure is showed in the figure 1.1.

Errors, however, are indicated by means of the error frames. CAN has a very robust error verification and confinement. This assures that a node with problems does not impair the communication in the network.

For a complete description of the errors, consult the CAN specification.

Interframe Space	1 bit	11 bits	1 bit	6 bits	0-8 bytes	15 bits	1 bit	1 bit	1 bit	7 bits	≥ 3 bits
	Start of Frame	Identifier	RTR bit	Control Field	Data Field	CRC Sequence	CRC Delimiter	ACK Slot	ACK Delimiter	End of Frame	Interframe Space
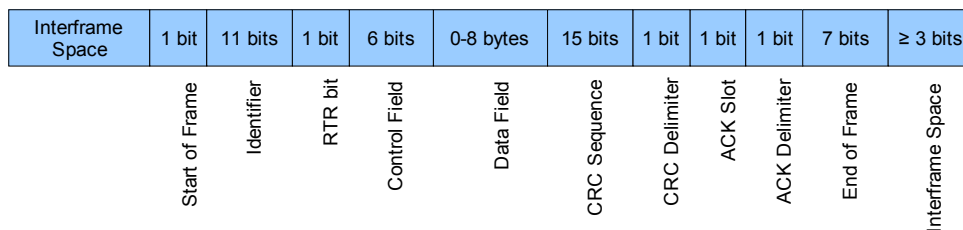
Figure 0.1 - CAN data frame

1.2.4 Network and Transport Layer

DeviceNet requires that a connection be established before data exchange with the device takes place. In order to establish this connection each DeviceNet node must implement the Unconnected Message Manager (UCMM) or the Group 2 Unconnected Port. These two allocation mechanisms use messages of the explicit type to establish a connection, which will then be used for process data exchange between one node and the other. This data exchange uses messages of the I/O type (refer to item 1.2.7).

The DeviceNet telegrams are classified in groups, which define specific functions and priorities. Those telegrams use the identifier field (11 bits) of the CAN data frame to uniquely identify each one of the messages, thus assuring the CAN priority mechanism.

A DeviceNet node can be a client, a server or both. Furthermore, clients and servers can be producers and/or consumers of messages. In a typical client node, for instance, its connection will produce requests and will consume answers. Other client or server connections will only consume messages. In other words, the protocol allows several connection possibilities among the devices.

The protocol also has a resource for detection of nodes with duplicated addresses (Mac ID). Avoiding that duplicated addresses occur is, in general, more efficient than trying to locate them later.

1.2.5 Application Layer – CIP Protocol

In the application layer, DeviceNet uses the Common Industrial Protocol (CIP). It is a protocol strictly orientated to objects, used also by ControlNet and EtherNet/IP. In other words, it is independent from the physical medium and from the data link layer. The Figure 1.2 presents the structure of this protocol.
The CIP has two main purposes:

☑ Transport of I/O devices control data.
☑ Transport of configuration and diagnosis information of the system being controlled.
A DeviceNet node (master or slave) is then molded by a set of CIP objects, which encapsulate data and services, thus determining its behavior.

There are obligatory objects (each device must have) and optional objects. Optional objects are those that mold the device according to the category (called profile) to which they belong, as: AC/DC drive, bar code reader or pneumatic valve. For being different, each one of these will contain a group of also different objects.

For more information refer to the DeviceNet specification. It presents the complete list of devices already standardized by the ODVA, as well as the objects that compose them.

1.2.6 Configuration File

Every DeviceNet node has a configuration file associated[4] . This file contains important information about the device operation and must be registered in the network configuration software.
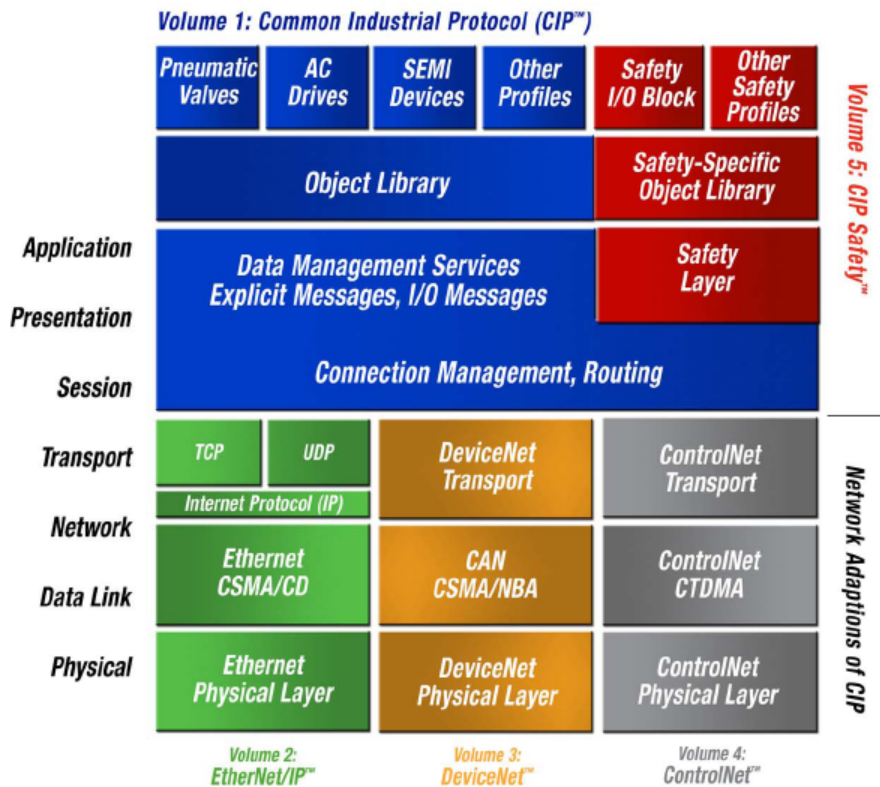
Figure 0.2 - CIP protocol structure in layers

---

[4] Known as EDS file.

1.2.7 Communication Modes

The DeviceNet protocol presents two basic types of messages, I/O and explicit. Each one of them is adequate to a specific kind of data, as described below:

☑ I/O: It is a kind of synchronous telegram dedicated to the movement of priority data between one producer and one or more consumers. They are divided according to the data exchange method. The main types are:

- *Polled:* It is a communication method where the master sends one telegram to each of the slaves of its list (scan list). As soon as receiving the request, the slave responds promptly to the request from the master. This process is repeated until all be consulted, restarting the cycle.

- *Bit-strobe: It* is a communication method where the master sends to the network a telegram containing 8 data bytes. Each bit from those 8 bytes represents a slave that, if addressed, responds according to the programmed.

- *Change of State:* It is a communication method where the data exchange between master and slave occurs only when changes in the monitored/controlled values happened, until a certain time limit. When this limit is reached, the transmission and reception will occur even if there were no changes. The configuration of this time variable is done in the network configuration program.

- *Cyclic:* It is another communication method very similar to the previous one. The only difference stays in the production and consume of messages. In this type, every data exchange occurs in regular time intervals, whether or not they had been changed. This time period is also adjusted in the network configuration software.

☑ Explicit: It is a kind of general purpose telegram and without priority. It is mainly used for asynchronous tasks like the parameter settings and the configuration of the equipment.

1.2.8 Set of Predefined Master/Slave Connections

DeviceNet uses fundamentally a point-to-point message model. However, it is quite common to use a predefined communication model based on the master/slave mechanism.

This model uses a simplified message movement of the I/O type, very common in control applications. An advantage of this method is that the necessary requests to run it are generally less than for the UCMM. Even simple devices with limited resources (memory, 8 bit processor) are capable of executing the protocol.

2   Accessory Kits

In order to make the DeviceNet communication possible with the CFW-11, it is necessary to use one of the CAN communication kits described next. Information on the installation of these modules in the inverter can be obtained in the guide that comes with the kits.

2.1   CAN Interfaces

2.1.1 CAN-01 Kit
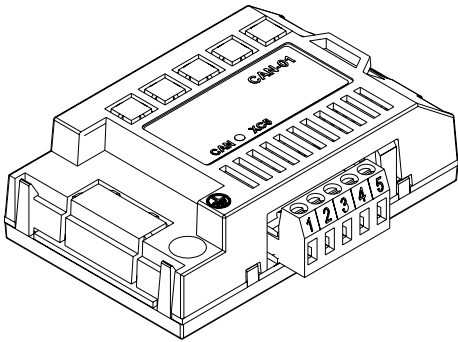
- ☑ WEG part number: 10051961.
- ☑ Composed by the CAN communication module (drawing at the left), mounting instruction and fixing screw.
- ☑ The interface is electrically isolated and with differential signal, which grants more robustness against electromagnetic interference.
- ☑ External 24V supply via the DeviceNet network cable.

2.1.2 CAN/RS485-01 Kit
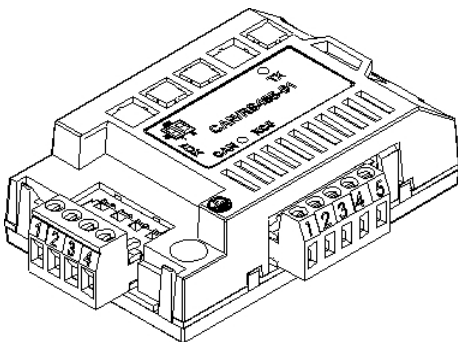
- ☑ WEG part number: 10051960.
- ☑ Composed by the CAN/RS485-01 communication module (drawing at the left), mounting instruction and fixing screw.
- ☑ It has the same characteristics as the CAN-01 interface, plus an RS485 interface, for applications where the operation with both interfaces is necessary.

2.1.3 Connector Pinout

The CAN communication module presents a 5-wire plug-in connector (XC5) with the following pinout:
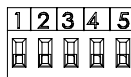
Table 2.1 – CAN interface XC5 connector pinout

Pin	Name	Function
1	V-	Power supply negative pole
2	CAN_L	CAN_L communication signal
3	Shield	Cable shield
4	CAN_H	CAN_H communication signal
5	V+	Power supply positive pole

2.1.4 Power Supply

The CFW-11 CAN interface needs an external power supply between the pins 1 and 5 of the network connector. In order to avoid problems of potential difference among the network devices, it is recommended that the network be supplied at only one point, and that the power supply be taken to all the devices trough the cable. If more than one power supply is necessary, they must be referenced to the same point. The individual consumption and input voltage data are presented in the next table.

Table 2.2 - *CAN interface supply characteristics*

Supply Voltage (V_{DC})		
Minimum	Maximum	Recommended
11	30	24
Current (mA)		
Minimum	Maximum	Average
20	50	30

2.2 Connection with the Network

For the connection of the inverter using the active DeviceNet interface the following points must be observed:

☑ The use of cables specific for CAN/DeviceNet networks is recommended.
☑ Grounding of the cable shield at only one point, thus avoiding current loops. This point is usually at the power supply. If there is more than one power supply, only one of them must be connected to the protective earth.
☑ The termination resistors must be installed only at extremes of the main bus, even if there are derivations.
☑ The network power supply must be able to deliver enough current to feed all the equipment transceivers. The CFW-11 DeviceNet module consumes approximately 50mA.

2.3 Module Configuration

In order to configure the DeviceNet module follow the steps indicated below:

☑ With the inverter powered off install the module on the XC43 connector (slot 3 – green).
☑ Make sure it is properly fitted into the connector and secured by the screw.
☑ Power up the inverter.
☑ Verify the content of the parameter P0028 making sure the accessory was correctly recognized. If necessary, refer to the installation guide.
☑ Set the CAN protocol for DeviceNet by means of the parameter P0700.
☑ Adjust the address of the inverter in the network through the parameter P0701.
  - Valid values: 0 to 63.
☑ Set the Baud Rate in P0702. Valid values:
  - 0 = Auto
  - 1 = Auto
  - 2 = 500kbps
  - 3 = 250kbps
  - 4 = 125kbps
  - 5 = Auto
  - 6 = Auto
  - 7 = Auto
  - 8 = Auto
  At the parameter P0710, configure the most suitable I/O instance for the application (this choice will impact the number of words exchanged with the network master). The very same number of words must be adjusted at the network master. Finally, program a value different from -1 in the parameters P0711 to P0718 (refer to the section 3).
  - Valid values: 0 to 7.
☑ Cycle the power of the CFW-11, so that the changes become effective.
☑ Connect the network cable to the module.
☑ Register the configuration file (EDS file) in the network configuration software.
☑ Add the CFW-11 to the scan list of the master.

☑ In the network configuration software, choose a method of data exchange with the master, i.e., *polled*, *change of state, cyclic* or *bit-strobe*. The CFW-11 DeviceNet module supports all these I/O data types, besides the *explicit* (acyclic data).

☑ If everything is configured correctly the parameter P0719 will indicate the "*Online* Not Connected" or "*Online* Connected" state. Observe also the parameter that indicates the network master status, P0720. There will only be effective data exchange when the master status is *Run*.

Refer to the section 3 for more information on the parameters mentioned above.

2.4 Access to the Parameters

After the EDS file registration in the network configuration software, the user will get access to the equipment complete parameter list, which can be accessed via explicit messages.
This means that it is possible to perform the drive programming and configuration through the network configuration software.

In order to get application details of this resource, refer to the network master programming documentation (PLC, PC, etc.).

3 Inverter Programming

Next, only the CFW-11 frequency inverter parameters related to the DeviceNet communication will be presented.

3.1 Symbols for Propriety Description

RO	Read-only parameter.
CFG	Parameter that can be changed only with a stopped motor.
Net	Parameter visible on the HMI if the inverter has the network interface installed – RS232, RS485, CAN, Anybus-CC, Profibus – or if the USB interface is connected.
Serial	Parameter visible on the HMI if the inverter has the RS232 or RS485 interface installed.
CAN	Parameter visible on the HMI if the inverter has the CAN interface installed.

P0105 – 1st/2nd Ramp Selection

P0220 – LOCAL/REMOTE Selection Source

P0221 – Speed Reference Selection – LOCAL Situation

P0222 – Speed Reference Selection – REMOTE Situation

P0223 – FORWARD/REVERSE Selection - LOCAL Situation

P0224 – Run/Stop Selection – LOCAL Situation

P0225 – JOG Selection – LOCAL Situation

P0226 – FORWARD/REVERSE Selection - REMOTE Situation

P0227 – Run/Stop Selection - REMOTE Situation

P0228 – JOG Selection - REMOTE Situation

These parameters are used in the configuration of the command source for the CFW-11 inverter local and remote situations. In order that the inverter be controlled through the DeviceNet interface, the options "CANopen/DNet" available in these parameters, must be selected.

The detailed description of these parameters is found in the CFW-11 Programming Manual.

P0313 – Communication Error Action

Adjustable Range:	0 = Inactive 1 = Disable via Run/Stop 2 = Disable via General Enable 3 = Change to LOCAL 4 = Change to LOCAL keeping the commands and the reference 5 = Fault trip	Factory Setting:	0

Proprieties: CFG, Net

Access groups via HMI:

01 PARAMETER GROUPS
∟ 49 Communication
∟ 111 Status/Commands

Description:
This parameter allows selecting which action must be executed by the inverter, in case that a communication error is detected.

Table 3.1 - Values for the parameter P0313

Options	Description
0 = Inactive	No action is taken and the inverter remains in the existing status
1 = Disable via Run/Stop	A stop command with deceleration ramp is executed and the motor stops according to the programmed deceleration ramp
2 = Disable via General Enable	The inverter is disabled by removing the General Enabling and the motor coasts to stop
3 = Change to LOCAL	The inverter commands change to LOCAL
4 = Change to LOCAL keeping the commands and the reference	The inverter is changed to the local mode; However, the enabling and reference commands received via the network , in case the inverter had been programmed for start/stop via HMI or 3-wire and reference via HMI or electronic potentiometer, are kept in the local mode.
5 = Fault Trip	Instead of an alarm, a communication error causes a fault at the inverter, so that it becomes necessary to perform the inverter fault reset in order to get it back to normal operation.

For the CAN interface used with the DeviceNet protocol, the following events are considered communication errors:

☑ A133 Alarm/F233 fault: CAN interface without power supply.
☑ A134 Alarm/F233 fault: bus off.
☑ A136 Alarm/F233 fault: Idle master.
☑ A137 Alarm/F233 fault: timeout in one or more I/O connections has occurred.

The description of these alarms/faults is presented in the section 4.

The actions described in this parameter are performed by means of the automatic writing of the respective bits on the Control Word via CAN parameter – P0684. In order that the commands written in this parameter be executed, it is necessary that the inverter be programmed to be commanded via DeviceNet. This programming is done by means of parameters P0220 to P0228.

P0680 – Logical Status

Adjustable Range:	0000h – FFFFh	Factory Setting:	-

Proprieties: RO

Access groups via HMI:

01 PARAMETER GROUPS
∟ 49 Communication
∟ 111 Status/Commands

Description:
It allows the monitoring of the inverter status. Each bit corresponds to one state:

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3 to 0
Function	Fault condition	Manual/ Automatic	Undervoltage	LOC/REM	JOG	Speed Direction	General Enabling active	Ramp enabled	In Alarm condition	In configuration mode	Second Ramp	Quick Stop Activated	Reserved

Table 3.2 - Parameter P0680 bit functions

Bits	Values
Bits 0 to 3	Reserved
Bit 4 Quick Stop Activated	**0**: Quick stop command is not activated. **1**: Inverter is executing quick stop command.

Bit 5 Second Ramp	**0**: The inverter is configured to use as acceleration and deceleration ramp for the motor, the first ramp, programmed at the parameters P0100 and P0101. **1**: The inverter is configured to use as acceleration and deceleration ramp for the motor, the second ramp, programmed at the parameters P0102 and P0103.
Bit 6 In Configuration Mode	**0**: Inverter operating normally. **1**: Inverter in configuration mode. Indicates a special condition when the inverter cannot be enabled: ☑ Executing the self tuning routine; ☑ Executing the guided start-up routine; ☑ Executing the HMI copy function; ☑ Executing the flash memory card guided routine; ☑ There is a parameter setting incompatibility. ☑ Without power supply at the inverter power section. **Note**: It is possible to obtain the exact description of the special operation mode at parameter P0692.
Bit 7 Alarm condition	**0**: The inverter is not in alarm condition. **1**: The inverter is in alarm condition. **Note** The alarm number can be read by means of the parameter P0048 – Current Alarm.
Bit 8 Run/Stop	**0**: The motor is stopped. **1**: The inverter is driving the motor at the set point speed, or executing either the acceleration or the deceleration ramp.
Bit 9 General Enabling active	**0**: General Enabling is not active. **1**: General enabling is active and the inverter is ready to run the motor.
Bit 10 Speed Direction	**0**: The motor is rotating in reverse mode. **1**: The motor is rotating in direct mode.
Bit 11 JOG	**0**: JOG function inactive. **1**: JOG function active.
Bit 12 LOC/REM	**0**: Inverter in LOCAL situation. **1**: Inverter in REMOTE situation.
Bit 13 Undervoltage	**0**: No Undervoltage. **1**: With Undervoltage.
Bit 14 Manual/ Automatic	**0**: PID in manual mode. **1**: PID in Automatic mode.
Bit 15 Fault condition	**0**: The inverter is not in a fault condition. **1**: Any fault has been recorded by the inverter. **Note**: The fault number can be read by means of the parameter P0049 – Current Fault.

P0681 – Motor Speed in 13 bits

Adjustable Range:	-32768 – 32767	Factory Setting:	-

Proprieties: RO

Access groups via HMI:

01 PARAMETER GROUPS
 └ 49 Communication
   └ 111 Status/Commands

Description:

It allows monitoring the motor speed. This word uses 13 bit resolution with signal to represent the motor nominal speed:

☑ P0681 = 0000h (0 decimal)   → motor speed = 0 rpm
☑ P0681 = 2000h (8192 decimal)   → motor speed = synchronous speed

Intermediate or higher speed values in rpm can be obtained by using this scale. E.g., for a 4 pole 1800 rpm synchronous speed motor, if the value read is 2048 (0800h), then, to obtain the speed in rpm one must calculate:

$$8192 – 1800 \text{ rpm} \qquad \text{Speed in rpm} = \frac{1800 \times 2048}{8192}$$
$$2048 – \text{value read in P0681}$$

Speed in rpm = 450 rpm

Negative values in this parameter indicate motor rotating in counterclockwise.

P0684 – CANopen/DeviceNet Control Word

Adjustable Range:	0000h – FFFFh	Factory Setting:	0000h

Proprieties:	CAN

Access groups via HMI:

01 PARAMETER GROUPS
  └ 49 Communication
      └ 111 Status/Commands

Description:

It is the inverter control word via CAN interface. This parameter can only be changed via the CAN interface (CANopen or DeviceNet protocols). For the other sources (HMI, USB, Serial, etc.) it behaves like a read-only parameter.

In order that the commands written in this parameter be executed, it is necessary that the inverter be programmed to be commanded via CAN. This programming is done by means of parameters P0105 and P0220 to P0228.

Each bit of this word represents a command that can be executed by the inverter.

Bits	15 to 8	7	6	5	4	3	2	1	0
Function	Reserved	Fault reset	Quick Stop	Second Ramp Use	LOC/REM	JOG	Direction of Rotation	General Enabling	Run/Stop

Table 3.3 - Parameter P0684 bit functions

Bits	Values
Bit 0 Run/Stop	**0:** It stops the motor with deceleration ramp. **1:** The motor runs according to the acceleration ramp until reaching the speed reference value.
Bit 1 General Enabling	**0:** It disables the inverter, interrupting the supply for the motor. **1:** It enables the inverter allowing the motor operation.
Bit 2 Direction of Rotation	**0:** To run the motor in a direction opposed to the speed reference. **1:** To run the motor in the direction indicated by the speed reference.
Bit 3 JOG	**0:** It disables the JOG function. **1:** It enables the JOG function.
Bit 4 LOC/REM	**0:** The inverter goes to the LOCAL situation. **1:** The inverter goes to the REMOTE situation.
Bit 5 Second Ramp Use	**0:** The inverter uses as acceleration and deceleration ramp for the motor, the first ramp times, programmed at the parameters P0100 and P0101. **1:** The inverter uses as acceleration and deceleration ramp for the motor, the second ramp times, programmed at the parameters P0102 and P0103.
Bit 6 Quick Stop	**0:** Quick Stop command not activated. **1:** Quick Stop command activated. **Obs.:** when the control modes V/f or VVW are selected, the use of this function is not recommended.
Bit 7 Fault reset	**0:** No function. **1:** If in a fault condition, then it executes the inverter reset.
Bits 8 to 15	Reserved

P0685 – CANopen/DeviceNet Speed Reference

Adjustable Range:	-32768 – 32767	Factory Setting:	0

Proprieties:	CAN

01 PARAMETER GROUPS
  └ 49 Communication
      └ 111 Status/Commands

Description:

It allows the programming of the speed reference for the inverter via CAN interface. This parameter can only be changed via the CAN interface (CANopen or DeviceNet protocols). For the other sources (HMI, USB, Serial, etc.) it behaves like a read-only parameter.

In order that the reference written in this parameter be used, it is necessary that the inverter be programmed for using the speed reference via CAN. This programming is done by means of parameters P0221 and P0222.

This word uses a 13 bit resolution with signal to represent the motor synchronous speed:

☑ P0685 = 0000h (0 decimal)   → speed reference = 0 rpm
☑ P0685 = 2000h (8192 decimal)   → speed reference = synchronous speed

Intermediate or higher speed reference values can be programmed by using this scale. E.g., for a 4 pole 1800 rpm synchronous speed motor, to obtain a speed reference of 900 rpm one must calculate:

$$1800 \text{ rpm} \quad - \quad 8192$$
$$900 \text{ rpm} \quad - \quad 13 \text{ bit reference} \qquad 13 \text{ bit reference} = \frac{900 \times 8192}{1800}$$

13 bit reference = 4096 (value corresponding to 900 rpm 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 bit 2 setting – P0684:

☑ Bit 2 = 1 and P0685 > 0: reference for direct speed rotation
☑ Bit 2 = 1 and P0685 < 0: reference for reverse speed rotation
☑ Bit 2 = 0 and P0685 > 0: reference for reverse speed rotation
☑ Bit 2 = 0 and P0685 < 0: reference for direct speed rotation

P0695 – Digital Output Setting

Adjustable Range:	0000h – FFFFh	Factory Setting:	0000h

Proprieties: Net

Access groups via HMI:

01 PARAMETER GROUPS
  └ 49 Communication
      └ 111 Status/Commands

Description:

It allows the control of the digital outputs by means of the network interfaces (Serial, USB, CAN, etc.). This parameter cannot be changed via HMI.

Each bit of this parameter corresponds to the desired value for a digital output. In order to have the correspondent digital output controlled according to this content, it is necessary that its function be programmed for "P0695 Content" at parameters P0275 to P0280.

Bits	15 to 5	4	3	2	1	0
Function	Reserved	Setting for DO5	Setting for DO4	Setting for DO3 (RL3)	Setting for DO2 (RL2)	Setting for DO1 (RL1)

Table 3.4 - *Parameter P0695 bit functions*

Bits	Values
Bit 0 Setting for DO1 (RL1)	**0**: DO1 output open. **1**: DO1 output closed.
Bit 1 Setting for DO2 (RL2)	**0**: DO2 output open. **1**: DO2 output closed.
Bit 2 Setting for DO3 (RL3)	**0**: DO3 output open. **1**: DO3 output closed.
Bit 3 Setting for DO4	**0**: DO4 output open. **1**: DO4 output closed.
Bit 4 Setting for DO5	**0**: DO5 output open. **1**: DO5 output closed.
Bits 5 to 15	Reserved

P0696 – Analog Output Value 1

P0697 – Analog Output Value 2

P0698 – Analog Output Value 3

P0699 – Analog Output Value 4

Adjustable Range:	-32768 – 32767	Factory Setting:	0

Proprieties: Net

Access groups via HMI:

01 PARAMETER GROUPS
    ∟ 49 Communication
        ∟ 111 Status/Commands

Description:

It allows the control of the analog outputs by means of network interfaces (Serial, USB, CAN, etc.) This parameter 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 "Content P0696/ P0697/ P0698/ P0696", in the parameters P0251, P0254, P0257 or P0260.

The value must be written in a 15 bit scale (7FFFh = 32767)[5] to represent 100% of the output desired value, i.e.:

☑ P0696 = 0000h (0 decimal) → analog output value = 0 %
☑ P0696 = 7FFFh (32767 decimal) → analog output value = 100 %

The showed example was for P0696, but the same scale is also used for the parameters P0697/P0698/P069. For instance, to control the analog output 1 via CAN interface the following programming must be done:

☑ Choose a parameter from P0696 to P0699 to be the value used by the analog output 1. For this example we are going to select P0696.
☑ Program the option "Content P0696" as the function for the analog output 1 in P0254.
☑ Using the CAN interface, write in P0696 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 operating from -10V to +10V, negative values must be used for these parameters to command the output with negative voltage values, i.e, -32768 to 32767 represent a variation from -10V to +10V at the output.

P0700 – CAN Protocol

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

Proprieties: CFG, CAN

---

[5] Refer to the CFW-11 manual for knowing the actual output resolution.

Access groups via HMI:

01 PARAMETER GROUPS
 └ 49 Communication
  └ 112 CANopen/DeviceNet

Description:

It allows selecting the desired protocol for the CAN interface. In order to enable the DeviceNet protocol it is necessary to set this parameter to the option "2 = DeviceNet".

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

P0701 – CAN Address

Adjustable Range:	0 to 127	Factory Setting:	63

Proprieties: CFG, CAN

Access groups via HMI:

01 PARAMETER GROUPS
 └ 49 Communication
  └ 112 CANopen/DeviceNet

Description:

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

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

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

P0702 – CAN Baud Rate

Adjustable Range:	0 = 1 Mbps/Auto 1 = 800 Kbps/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

Proprieties: CFG, CAN

Access groups via HMI:

01 PARAMETER GROUPS
 └ 49 Communication
  └ 112 CANopen/DeviceNet

Description:

It allows programming the desired value for the CAN interface Baud rate, in bits per second. This rate must be the same for all the devices connected to the network. When the option "Auto" (autobaud) is selected, the CFW-11 will adjust itself automatically to the actual network baud rate. In order that this mechanism works it is necessary that at least two or more devices be actively communicating in the network.

After a successful detection, the baud rate parameter (P0702) changes automatically to the detected rate. In order to execute the autobaud function again, it is necessary to change the parameter P0702 to one of the "Auto" options.

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

P0703 – Bus Off Reset

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

Proprieties: CFG, CAN

Access groups via HMI:

01 PARAMETER GROUPS
    └ 49 Communication
        └ 112 CANopen/DeviceNet

Description:

It allows programming the inverter behavior when detecting a bus off error in the CAN interface:

Table 3.5 - *Values for the parameter P0703*

Options	Description
0 = Manual Reset	If *bus off* occurs, the A134/F234 alarm will be indicated on the HMI, the action programmed in parameter P0313 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 Reset	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 P0313.

P0705 – CAN Controller Status

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

Proprieties: RO, CAN

Access groups via HMI:

01 PARAMETER GROUPS
    └ 49 Communication
        └ 112 CANopen/DeviceNet

Description:

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

Table 3.6 - *Values for the parameter P0705*

Options	Description
0 = Inactive	Inactive CAN interface. It occurs when the inverter does not have the CAN interface installed.
1 = Auto-baud	CAN controller is trying to detect baud rate of the network.
2 = Active CAN interface	CAN interface is active and without errors.
3 = *Warning*	The CAN controller has reached the *warning* state.
4 = *Error Passive*	The CAN controller has reached the *error passive* state.
5 = *Bus Off*	The CAN controller has reached the *bus off* state.
6 = No Bus Power	The CAN interface does not have power supply between the pins 1 and 5 of the connector.

P0706 – Received CAN Telegram Counter

| Adjustable Range: | 0 to 65535 | Factory Setting: | - |

Proprieties: RO, CAN

Access groups via HMI:

01 PARAMETER GROUPS
└ 49 Communication
  └ 112 CANopen/DeviceNet

Description:

This parameter works as a cyclic counter that is incremented every time a CAN telegram is received. It gives a feedback to the operator if the device is able to communicate with the network. This counter is reset every time the inverter is switched off, a reset is performed or when the maximum limit of the parameter is reached.

P0707 – Transmitted CAN Telegram Counter

Adjustable Range:	0 to 65535	Factory Setting:	-

Proprieties: RO, CAN

Access groups via HMI:

01 PARAMETER GROUPS
└ 49 Communication
  └ 112 CANopen/DeviceNet

Description:

This parameter works as a cyclic counter that is incremented every time a CAN telegram is transmitted. It gives a feedback to the operator if the device is able to communicate with the network. This counter is reset every time the inverter is switched off, a reset is performed or when the maximum limit of the parameter is reached.

P0708 – Buss Off Error Counter

Adjustable Range:	0 to 65535	Factory Setting:	-

Proprieties: RO, CAN

Access groups via HMI:

01 PARAMETER GROUPS
└ 49 Communication
  └ 112 CANopen/DeviceNet

Description:

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

P0709 – Lost CAN Message Counter

Adjustable Range:	0 to 65535	Factory Setting:	-

Proprieties: RO, CAN

Access groups via HMI:

01 PARAMETER GROUPS
└ 49 Communication
  └ 112 CANopen/DeviceNet

Description:

It is a cyclic counter that indicates the number of messages received by the CAN interface that could not be processed by the inverter. 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 inverter is switched off, a reset is performed or when the maximum limit of the parameter is reached.

P0710 – DeviceNet I/O Instances

Adjustable Range:	0 = ODVA Basic Speed (2 words) 1 = ODVA Extended Speed (2 words) 2 = Manuf.Spec.2W (2 words) 3 = Manuf.Spec.3W (3 words) 4 = Manuf.Spec. 4W (4 words) 5 = Manuf.Spec. 5W (5 words) 6 = Manuf.Spec. 6W (6 words)	**Factory Setting:** 0

Proprieties: CFG, CAN

Access groups via HMI:

01 PARAMETER GROUPS
    └ 49 Communication
        └ 112 CANopen/DeviceNet

Description:

It allows selecting the *Assembly* class instance for the I/O type communication. These instances represent the user interface with the CFW-11 inverter. Each one presents the control and monitoring data in one manner. It is up to the user to choose which option is the best for the application.

The CFW-11 has seven setting options. Two of them follow the ODVA *AC/DC Drive Profile*. The other five represent specific WEG words. The tables presented next describe each of these control and monitoring words.

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

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 Device Profile.* The data mapping is showed below.

Monitoring (Input)

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	0						Running1		Faulted
	1	-							
	2	Actual Speed (low byte)							
	3	Actual Speed (high byte)							

Control (Output)

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	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 Device Profile.* The data mapping is showed below.

Monitoring (Input)

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	At Reference	Reference from Network	Control from Network	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Actual Speed (low byte)							
	3	Actual Speed (high byte)							

Control (Output)

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
71	0		Network Reference	Network Control			Fault Reset	Reverse Run	Forward Run
	1	-							
	2	Speed Reference (low byte)							
	3	Speed Reference (high byte)							

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

Monitoring:

Bits (Byte 0)	Values
Bit 0 Faulted	**0:** The inverter is not in a fault condition. **1:** A fault has been recorded by the inverter. **Note:** The fault number can be read by means of the parameter P0049 – Current Fault.
Bit 1 Warning	**0:** The inverter is not in alarm condition. **1:** The inverter is in alarm condition. **Note** The alarm number can be read by means of the parameter P0048 – Current Alarm.
Bit 2 Running1 (Fwd)	**0:** The motor is not rotating clockwise **1:** The motor is rotating clockwise
Bit 3 Running2 (Rev)	**0:** The motor is not rotating counterclockwise **1:** The motor is rotating counterclockwise
Bit 4 Ready	**0:** The inverter is not ready to operate **1:** The inverter is ready to operate (states *Ready, Enabled* or *Stopping*)
Bit 5 Control from Network	**0:** The drive is controlled locally **1:** The drive is controlled remotely
Bit 6 Reference from Network	**0:** The speed reference is not being sent via the DeviceNet network **1:** It indicates that the speed reference is being sent via the DeviceNet network
Bit 7 At Reference	**0:** The inverter has not yet reached the programmed speed **1:** The inverter has reached the programmed speed

☑ Byte 1 indicates the drive status:
- 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 motor actual speed in rpm.

Control:

Bits (Byte 0)	Values
Bit 0 Run Forward	**0:** It stops the motor **1:** It runs the motor clockwise

Bit 1 Run Reverse	**0:** It stops the motor **1:** It runs the motor counterclockwise
Bit 2 Fault Reset[6]	**0:** no function **1:** If in a fault condition, then it executes the inverter reset.
Bits 3 and 4	Reserved
Bit 5 Network Control[7]	**0:** It selects the local mode **1:** It selects the remote mode
Bit 6 Network Reference	**0:** The speed reference is not being sent via the DeviceNet network **1:** That the speed reference be sent via the network
Bit 7	Reserved

☑ Bytes 2 (low) and 3 (high) (high) represent the motor actual speed in rpm.

2 = Data format for the *WEG Specific 2W* (2 words) instances:

Called *WEG Specific 2W*, these instances present the simplest equipment operation interface according to the WEG profile. The data mapping is showed below.

Monitoring (Input)

Bits	15	14	13	12	11	10	9	8	7	6	5	4 to 0
Function	Faulted	Manual/ Automatic	Undervoltage	LOC/REM	JOG	Speed Direction	General Enable	Running	Alarm	Configuration mode	Second Ramp	Reserved

---

6 After a reset the drive goes to the offline state.
7 In order that this action be executed the parameters P0220 – P0228 must be correctly programmed.

Bits (Byte 0 and 1)	Values
Bits 0 to 4	Reserved.
Bit 5 Second Ramp	**0:** The inverter is configured to use as acceleration and deceleration ramp for the motor, the first ramp, programmed at the parameters P0100 and P0101. **1:** The inverter is configured to use as acceleration and deceleration ramp for the motor, the second ramp, programmed at the parameters P0102 and P0103.
Bit 6 Configuration mode	0: Inverter operating normally. 1: Inverter in configuration mode. Indicates a special condition when the inverter cannot be enabled: ☑ Executing the self tuning routine; ☑ Executing guided start-up routine; ☑ Executing the HMI copy function; ☑ Executing the flash memory card guided routine; ☑ There is a parameter setting incompatibility. ☑ Without power supply at the inverter power section. **Note:** It is possible to obtain the exact description of the special operation mode at parameter P0692.
Bit 7 Alarm	**0:** The inverter is not in alarm condition. **1:** The inverter is in alarm condition. **Note:** The alarm number can be read by means of the parameter P0048 – Current Alarm.
Bit 8 Running	**0:** The motor is stopped. **1:** The inverter is driving the motor at the set point speed, or executing either the acceleration or the deceleration ramp.
Bit 9 General Enable	**0:** General Enabling is not active. **1:** General enabling is active and the inverter is ready to run the motor.
Bit 10 Speed Direction	**0:** The motor is rotating counterclockwise. **1:** The motor is rotating clockwise.
Bit 11 JOG	**0:** JOG function inactive. **1:** JOG function active.
Bit 12 LOC/REM	**0:** Inverter in LOCAL situation. **1:** Inverter in REMOTE situation.
Bit 13 Undervoltage	**0:** No Undervoltage. **1:** With Undervoltage.
Bit 14 Manual/ Automatic	**0:** PID in manual mode. **1:** PID in Automatic mode.
Bit 15 Faulted	**0:** The inverter is not in a fault condition. **1:** A fault has been recorded by the inverter. **Note:** The fault number can be read by means of the parameter P0049 – Current Fault.

Control (Output)

Bits	15 to 8	7	6	5	4	3	2	1	0
Function	Reserved	Fault Reset	Reserved	Second Ramp Use	LOC/REM	JOG	Speed Direction	General Enabling	Start/Stop

Bits (Byte 0 and 1)	Values
Bit 0 Start/Stop	**0:** It stops the motor with deceleration ramp. **1:** The motor runs according to the acceleration ramp until reaching the speed reference value.
Bit 1 General Enabling	**0:** It disables the inverter, interrupting the supply for the motor. **1:** It enables the inverter allowing the motor operation.
Bit 2 Speed Direction	**0:** To run the motor in the direction indicated by the speed reference. **1:** To run the motor in a direction opposed to the speed reference.
Bit 3 JOG	**0:** It disables the JOG function. **1:** It enables the JOG function.
Bit 4 LOC/REM	**0:** The inverter goes to the LOCAL situation. **1:** The inverter goes to the REMOTE situation.
Bit 5 Second Ramp Use	**0:** The inverter uses as acceleration and deceleration ramp for the motor, the first ramp times, programmed at the parameters P0100 and P0101. **1:** The inverter uses as acceleration and deceleration ramp for the motor, the second ramp times, programmed at the parameters P0102 and P0103.
Bit 6	Reserved
Bit 7 Fault reset	**0:** No function. **1:** If in a fault condition, then it executes the inverter reset.
Bits 8 to 15	Reserved

3 = Data format for the *WEG Specific 3W* (3 words) instances:
4 = Data format for the *WEG Specific 4W* (4 words) instances:
5 = Data format for the *WEG Specific 5W* (5 words) instances:

6 = Data format for the *WEG Specific 6W* (6 words) instances:
These instances have the same format as the ***WEG Specific 2W.*** Besides the command and monitoring words showed above, they make it possible to program up to 4 parameters of the equipment for reading and/or writing via network.

Monitoring (Input)

Instance	16 bit words	Function	P0710 Options				
100	1	Monitoring	2	3	4	5	6
	2	Actual Speed (13 bit representation)					
	3	Contend of the parameter P0711					
	4	Contend of the parameter P0712					
	5	Contend of the parameter P0713					
	6	Contend of the parameter P0714					

Control (Output)

Instance	16 bit words	Function	P0710 Options				
150	1	Control	2	3	4	5	6
	2	Speed Reference (13 bit representation)					
	3	Contend of the parameter P0715					
	4	Contend of the parameter P0716					
	5	Contend of the parameter P0717					
	6	Contend of the parameter P0718					

P0712 – DeviceNet Reading Word #4

P0713 – DeviceNet Reading Word #5

P0714 – DeviceNet Reading Word #6

Adjustable Range:	-1 to 1499	Factory Setting:	-1 (disabled)

Proprieties: CFG, CAN

Access groups via HMI:

01 PARAMETER GROUPS
 └ 49 Communication
   └ 112 CANopen/DeviceNet

Description:

These parameters allow the user to program the reading of any other equipment parameter via the network. That is, they contain the number of another parameter.

For instance, P0711 = 5. In this case the content of P0005 (motor frequency) will be sent via network. In this manner, in the network master memory position corresponding to the third reading word, the motor frequency will be read.

P0715 – DeviceNet Writing Word #3

P0716 – DeviceNet Writing Word #4

P0717 – DeviceNet Writing Word #5

P0718 – DeviceNet Writing Word #6

Adjustable Range:	-1 to 1499	Factory Setting:	-1 (disabled)

Proprieties: CFG, CAN

Access groups via HMI:

01 PARAMETER GROUPS
 └ 49 Communication
   └ 112 CANopen/DeviceNet

Description:

These parameters allow the user to program the writing of any other equipment parameter via the network. That is, they contain the number of another parameter.

For instance, P0715 = 100. In this case the content to be written in P0100 will be sent via network. In this manner, the network master memory position corresponding to the third writing word must contain the value for P0100.

P0719 – DeviceNet Network Status

Adjustable Range:	0 = Offline 1 = Online, Not Connected 2 = Online, Connected 3 = Timed-out Connection 4 = Connection Failure 5 = Auto-baud	Factory Setting:	-

Proprieties: RO, CAN

Access groups via HMI:

01 PARAMETER GROUPS
 └ 49 Communication
   └ 112 CANopen/DeviceNet

Description:

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

Status	Description
Offline	Device without power supply or not online. Communication cannot be established,
Online, Not Connected	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.
Online, Connected	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.
Timed-out Connection	One or more I/O type connections have expired.
Connection 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.
Autobaud	The equipment is executing the autobaud mechanism.

P0720 – DeviceNet Master Status

Adjustable Range:	0 = Run 1 = Prog (Idle)	Factory Setting:	-

Proprieties: RO, CAN

Access groups via HMI:

01 PARAMETER GROUPS
   ∟ 49 Communication
      ∟ 112 CANopen/DeviceNet

Description:

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

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

P0799 – I/O Update Delay

Range:	0.0 to 999.0	Default:	0.0

Proprieties: RW

Access groups via HMI:

01 PARAMETER GROUPS.
   ∟ 49 Communication
      ∟ 111 Status/Commands

Description:

It allows setting the delay time for the update of the data mapped in the writing words (data received by the equipment) via Profibus DP, Devicenet, CANopen communication networks and Anybus interface. The delay time is activated in the transition of the equipment status in the network from offline to online[8], as in *figure 3.1*.

---

8 For this function, online represents the state where the exchange of cyclic I/O data occurs.
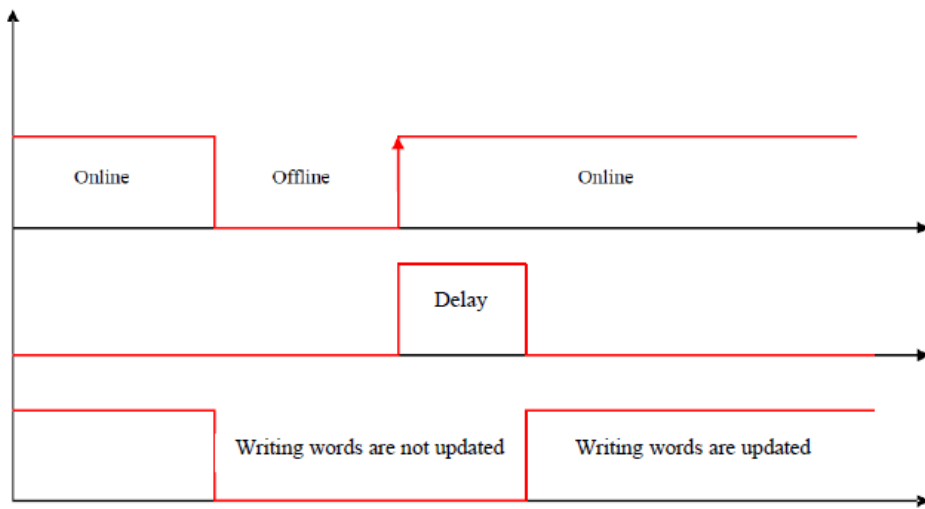
Figure 3.1 Delay in the update of I/O words

4 Faults and Alarms Related to the DeviceNet Communication

A133/F233 – CAN Interface Without Power Supply

Description:
It indicates that the CAN interface does not have power supply between the pins 1 and 5 of the connector.

Actuation:
In order that it be possible to send and receive telegrams through the CAN interface, it is necessary to supply external power to the interface circuit.

If the absence of power supply at the CAN interface is detected, the alarm A133 or the fault F233, depending on the P0313 programming, will be signalized through the HMI. If the circuit power supply is reestablished, the CAN communication will be reinitiated. In case of alarms, the alarm indication will also be removed from the HMI.

Possible Causes/Correction:
- ☑ Measure if there is voltage between the pins 1 and 5 of the CAN interface connector.
- ☑ Verify if the power supply cables have not been changed or inverted.
- ☑ Make sure there is no contact problem in the cable or in the CAN interface connector.

A134/F234 – Bus Off

Description:
The bus off error in the CAN interface has been detected.

Actuation:
If the number of reception or transmission errors detected by the CAN interface is too high, the CAN controller can be taken to the bus off state, where it interrupts the communication and disables the CAN interface.

In this case the alarm A134 or the fault F234, depending on the P0313 programming, will be signalized through the HMI. In order that the communication be reestablished, it will be necessary to cycle the power of the inverter, or remove the power supply from the CAN interface and apply it again, so that the communication be reinitiated.

Possible Causes/Correction:
- ☑ Make sure there is no short-circuit in CAN circuit transmission cables.
- ☑ Verify if the cables have not been changed or inverted.
- ☑ Verify if all the network devices use the same baud rate.
- ☑ Make sure that termination resistors with the correct have been installed only at the extremes of the main bus.
- ☑ Verify if the installation of the CAN network was done in an adequate manner.

A136/F236 – Idle Master

Description:
It is the alarm that indicates that the DeviceNet network master is in the Idle mode.

Actuation:
It acts when the CFW-11 detects that the network master went to the Idle mode. In this mode, only the variables read from the slave continue being updated in the memory of the master. None of the commands sent to the slave is processed.

In this case the alarm A136 or the fault F236, depending on the P0313 programming, will be signalized through the HMI. In case of alarms, If the master is set in the *Run* mode again (normal equipment operation status), the alarm indication will be removed from the HMI.

Possible Causes/Correction:
- ☑ Adjust the switch that commands the master operation mode for execution (Run) or set the correspondent bit in the configuring word of the master software. In case of doubts, referrer to used master documentation.

A137/F237 – DeviceNet Connection Timeout

Description:

It is the alarm that indicates that one or more DeviceNet I/O connections have expired.

Actuation:

It occurs when, for any reason, the master is not able to access information from the slave.

In this case the alarm A137 or the fault F237, depending on the P0313 programming, will be signalized through the HMI. In case of alarms, if the connection with the master is reestablished, the alarm indication will be removed from the HMI.

Possible Causes/Correction:

☑ Verify if the master is present in the network and if it is in the Run mode.