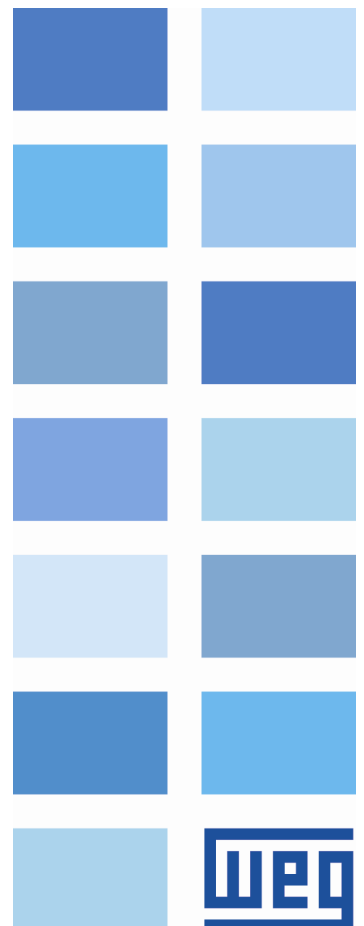


# BACnet

CFW501

## User's Manual





# **BACnet User's Manual**

Series: CFW501

Language: English

Document Number: 10002041146 / 01

Publication Date: 04/2022

# CONTENTS

<b>CONTENTS .....</b>	<b>3</b>
<b>ABOUT THE MANUAL .....</b>	<b>5</b>
<b>ABBREVIATIONS AND DEFINITIONS .....</b>	<b>5</b>
<b>NUMERICAL REPRESENTATION .....</b>	<b>5</b>
<b>DOCUMENTS.....</b>	<b>5</b>
<b>1 INTRODUCTION TO THE SERIAL COMMUNICATION .....</b>	<b>6</b>
<b>2 INTRODUCTION TO THE BACNET COMMUNICATION .....</b>	<b>7</b>
<b>2.1 BACNET MS/TP.....</b>	<b>8</b>
2.1.1 BACnet MS/TP Message Structure.....	8
<b>2.2 ADDRESS .....</b>	<b>10</b>
<b>2.3 BACNET PROFILE.....</b>	<b>10</b>
2.3.1 ReadProperty (DS-RP-B).....	10
2.3.2 WriteProperty (DS-WP-B).....	11
2.3.3 WHO IS / I AM (DM-DDB-B) .....	11
2.3.4 Device Management-Reinitialize Device-B (DM-RD-B).....	11
<b>3 INTERFACE DESCRIPTION.....</b>	<b>12</b>
<b>3.1 PLUG-IN MODULES .....</b>	<b>12</b>
3.1.1 RS485 connector for the standard plug-in module (CFW500-IOS).....	12
<b>3.2 PLUG-IN MODULES WITH RS485 INTERFACE AND ADDITIONAL INTERFACE .....</b>	<b>13</b>
3.2.1 Plug-in module with two RS485 interfaces (CFW500-CRS485-B) .....	13
3.2.2 Plug-in module with RS485 and USB interfaces (CFW500- CUSB) .....	13
3.2.3 Plug-in module with RS485 and RS232 interfaces (CFW500-CRS232) .....	14
<b>3.3 RS485 .....</b>	<b>14</b>
3.3.1 RS485 Interface Characteristics .....	14
3.3.2 Terminating resistor .....	14
3.3.3 Indications .....	15
3.3.4 Connection with the RS485 Network.....	15
<b>4 PROGRAMMING .....</b>	<b>16</b>
<b>4.1 SYMBOLS FOR THE PROPERTIES DESCRIPTION .....</b>	<b>16</b>
P0105 – 1 <sup>ST</sup> /2 <sup>ND</sup> RAMP SELECTION.....	16
P0220 – LOCAL/REMOTE SELECTION SOURCE .....	16
P0221 – SPEED REFERENCE SELECTION – LOCAL SITUATION .....	16
P0222 – SPEED REFERENCE SELECTION – REMOTE SITUATION .....	16
P0223 – FORWARD/REVERSE SELECTION – LOCAL SITUATION .....	16
P0224 – RUN/STOP SELECTION – LOCAL SITUATION .....	16
P0225 – JOG SELECTION – LOCAL SITUATION .....	16
P0226 – FORWARD/REVERSE SELECTION – REMOTE SITUATION .....	16
P0227 – RUN/STOP SELECTION – REMOTE SITUATION .....	16
P0228 – JOG SELECTION – REMOTE SITUATION .....	16
P0308 – SERIAL ADDRESS .....	16
P0310 – SERIAL BAUD RATE.....	17
P0311 – SERIAL INTERFACE BYTE CONFIGURATION .....	17
P0312 – SERIAL PROTOCOL .....	18
P0313 – COMMUNICATION ERROR ACTION .....	19
P0314 – SERIAL WATCHDOG .....	20
P0316 – SERIAL INTERFACE STATUS .....	20
P0680 – STATUS WORD .....	20
P0681 – MOTOR SPEED IN 13 BITS .....	22

P0682 – SERIAL CONTROL WORD .....	22
P0683 – SERIAL SPEED REFERENCE .....	24
P0695 – DIGITAL OUTPUT SETTING .....	24
P0696 – VALUE 1 FOR ANALOG OUTPUTS .....	25
P0697 – VALUE 2 FOR ANALOG OUTPUTS .....	25
P0698 – VALUE 3 FOR ANALOG OUTPUTS .....	25
P0760 – BACNET EQUIPMENT INSTANCE – HIGH PART .....	26
P0761 – BACNET EQUIPMENT INSTANCE – LOW PART .....	26
P0762 – MAXIMUM MASTER NUMBER .....	27
P0763 – MAXIMUM NUMBER OF MS/TP FRAMES .....	28
P0764 – I AM TRANSMISSION .....	28
P0765 – NUMBER OF RECEIVED TOKENS .....	28
<b>5 BACNET OBJECT MODELING .....</b>	<b>29</b>
5.1 BACNET OBJECTS .....	30
5.1.1 ANALOG INPUT OBJECT .....	30
5.1.2 ANALOG OUTPUT OBJECT .....	30
5.1.3 ANALOG VALUE OBJECT .....	31
5.1.4 BINARY INPUT OBJECT .....	31
5.1.5 BINARY OUTPUT OBJECT .....	32
5.1.6 BINARY VALUE OBJECT .....	32
5.1.7 DEVICE OBJECT .....	33
5.1.8 Mailbox .....	33
<b>6 FAULTS AND ALARMS RELATED TO THE BACNET COMMUNICATION .....</b>	<b>35</b>
A128/F228 – TIMEOUT FOR SERIAL COMMUNICATION .....	35

## ABOUT THE MANUAL

This manual provides the necessary information for the operation of the CFW501 frequency inverter using the BACnet protocol. This manual must be used together with the CFW501 user manual.

### ABBREVIATIONS AND DEFINITIONS

ASCII	American Standard Code for Information Interchange
PLC	Programmable Logic Controller
HMI	Human-Machine Interface
ro	Read-only
rw	Read/write

### NUMERICAL REPRESENTATION

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

### DOCUMENTS

The BACnet protocol for the CFW501 was developed based on the following specifications and documents:

Document	Version	Source
Standard 135-2004	1.0	ANSI/ASHRAE/ISO

In order to obtain this documentation, consult BACnet ORG, which is the organization that currently maintains, publishes and updates the information regarding the BACnet network.

## **1 INTRODUCTION TO THE SERIAL COMMUNICATION**

In a serial interface, the data bits are sent sequentially through a communication channel, or busbar. Several technologies use serial communication for data transfer, including the RS232 and RS485 interfaces.

The standards that specify the RS232 and RS485 interfaces, however, do specify neither the format nor the character sequence for data transmission and reception. In this sense, besides the interface, it is also necessary to identify the protocol used for the communication.

The BACnet MS/TP network defines the BACnet message exchange using the RS485 interface as the physical layer.

The characteristics of the RS485 serial interfaces available in the CFW501 frequency inverter, as well as the BACnet protocol, will be presented next.

## 2 INTRODUCTION TO THE BACNET COMMUNICATION

BACnet, acronym for "Building Automation Control Network", is a protocol defined by the ANSI/ASHRAE/ISO Standard 135-2004. The protocol defines a model for building-automation, describing the interaction between devices and systems. The protocol defines:

- Data and commands structured in an object-oriented model;
- Services that describe the access to data;
- A flexible network architecture.

The BACnet standard defines six types of communication networks for transporting BACnet messages, as showed in the Figure 2.1. The type of network defines the physical and data link layers. The six types of networks are:

- BACnet ARCnet;
- BACnet Ethernet;
- BACnet Lontalk;
- BACnet MS/TP;
- BACnet Point-to-Point;
- BACnet IP;

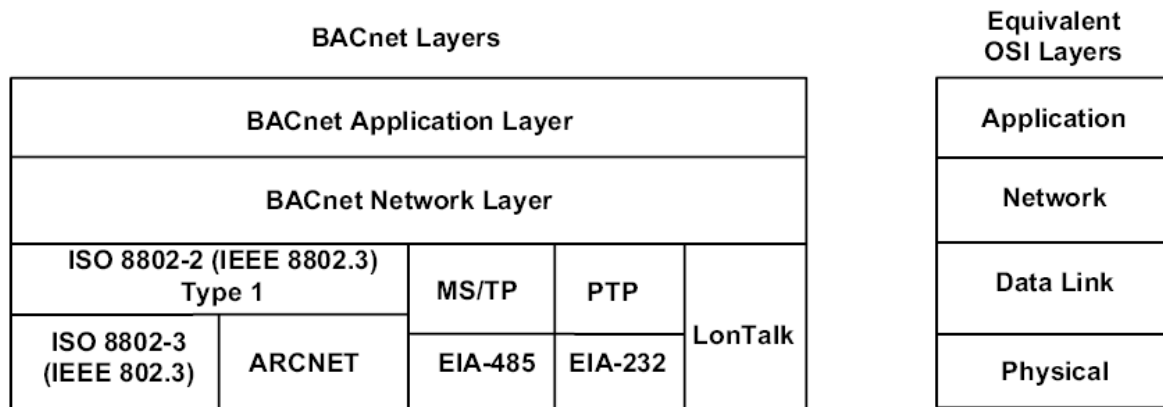


Figure 2.1: BACnet protocol architecture

A BACnet equipment contains an information collection defined as objects and properties.

A BACnet object represents physical or virtual information of the equipment, as a digital or analog input, control variables and parameters. The BACnet standard defines 25 types of objects. Each object is identified by a propriety called Object Identifier, which codifies the object instance type in a 32-bit binary number.

A BACnet property represents characteristics or information of a BACnet object. It is through the properties that other elements can access the equipment information. The property access can be defined as read-only or writing/reading. The BACnet specification defines services that are grouped in five categories:

- Object access
- Device Management
- Alarm and event
- File transfer
- Virtual terminal

BACnet equipments can be classified in six different profiles according to the set of services made available:

- BACnet Operator Workstation (B-OWS)
- BACnet Building Controller (B-BC)

- BACnet Advanced Application Controller (B-AAC)
- BACnet Application Specific Controller (B-ASC)
- BACnet Smart Actuator (B-AS)
- BACnet Smart Sensor (B-SS)

## 2.1 BACNET MS/TP

In the CFW501, the BACnet protocol was developed using the RS485 standard for the physical and data link layers, called BACnet MS/TP (Master Slave/Token Passing). BACnet MS/TP nodes can be divided into two groups, master nodes and slave nodes, according to the node address range.

The access control to the communication mean is performed in two ways:

- **Master/Slave (MS):** It is used in the communication between a master node and a slave node;
- **Token passing (TP):** Communication only among master nodes. A logical ring is defined and the master that has the token can establish communication with slave nodes and other masters.

In a BACnet MS/TP network, the nodes are initialized and enter the IDLE state, waiting for the arrival of a telegram, which can be:

- Invalid frame: it remains in IDLE;
- Not desired frame: it remains in IDLE;
- Token: it enters the USE TOKEN state, executes the necessary communication (with slaves or other masters) and passes the token to the next node;
- Reception of a Poll of Master: it sends a telegram to the node with the address in the Source Address field;
- Reception of a DataNoReplay: it signalizes the reception to the higher layers;
- Reception of a DataNeedingReplay: it signalizes the reception to the higher layers and sends the requested response.

### 2.1.1 BACnet MS/TP Message Structure

The BACnet specification defines that the frame can have from 0 up to 501 bytes (octets), and that each byte is composed by 8 bits without parity, with start and stop bits, as showed in the Figure 2.2.

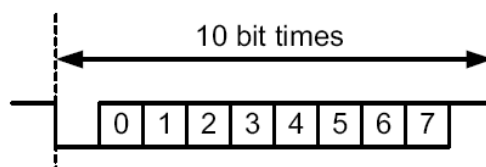


Figure 2.2: Byte structure

**Reception (RX):** The maximum time between bytes ( $T_{framegap}$ ) is of 20-bit times, and the minimum time between frames ( $T_{turnaround}$ ) after the last byte stop bit is of 40-bit times, according to the Figure 2.3.

**Transmission (TX):** The RTS signal must be disabled after the end of the stop bit time ( $T_{postdrive}$ ), which is of 15-bit times.



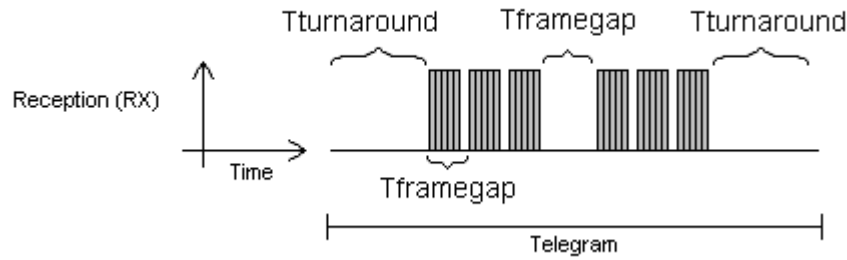


Figure 2.3: BACnet data reception

Header and data, as Figure 2.4 illustrates, form the BACnet data frame.

HEADER								DATA		
0x55	0xFF	Frame type	Destination address	Source address	Length	Length	CRC	Data	CRC	CRC

Figure 2.4: BACnet Frame

**Preamble:** It is formed by two bytes with the 55h and FFh values respectively.

**Frame type:** The BACnet specification defines eight frame types, from 0 to 7. Frame Types 8 through 127 are reserved for the specification improvement, and from 128 through 255 are reserved for each vendor specific frames. The defined types are:

- 0 Token
- 1 Poll for master
- 2 Reply to poll for master
- 3 Test request
- 4 Test response
- 5 BACnet data expecting reply
- 6 BACnet data not expecting reply
- 7 Reply postponed

Only master nodes must acknowledge frame types 0, 1 and 2, the slave nodes must ignore them.

Token (0) frame type: it is used in the relationship between master nodes. It does not present data. The master node that has the token can initiate the communication. After sending the maximum number of data frames (Nmax\_info\_frames) and waiting any expected replies, it must pass the token to the next master.

Poll for Master (1) frame type: It is transmitted periodically during the configuration. It is used to discover the presence of other masters in the network and to determine the token sequence. Master nodes must respond and slave nodes must ignore it. It does not present data.

Reply to Poll for Master (2) frame type: It is the response of the master nodes to the Poll for Master (frame type 1). It does not present data.

Test Request (3) frame type: It is used to start the communication in the MS/TP network. It is applied to send a particular piece of information to a node.

Test Response (4) frame type: It is the response to a Test Request.

BACnet Data Expecting Reply (5) frame type: It is used by master nodes to convey the data parameter of a DL\_UNITDATA.request that presents destination address, data, priority and message code, waiting for a response from the destination node.

**BACnet Data not Expecting Reply (6) frame type:** It is used by master nodes to convey the data parameter of a DL\_UNITDATA.request that presents destination address, data, priority and message code. It does not wait for a response from the destination node.

**Reply Postponed (7) frame type:** It is used by master nodes to indicate that the response to a Data Expecting Reply frame will be sent later. It does not present data.

**Destination and source addresses:** It is formed by two bytes, destination and source, respectively.

**Length:** It is formed by two bytes that inform the number of data bytes in the message.

**Header CRC:** The last part of the header is the fields for checking header transmission errors. The used method is the CRC-8 (Cyclic Redundancy Check).

**Data:** It may present from 0 to 501 bytes, according to the BACnet specification. Data in the CFW501 can present up to 59 bytes.

**Data CRC:** The last part of the telegram is the field for checking data transmission errors. The used method is the CRC-16 (Cyclic Redundancy Check).

## 2.2 ADDRESS

It presents an address range from 0 to 254, where:

- The range from 0 to 127 is reserved for master or slave nodes;
- The range from 128 to 254 is used only by slave nodes.

The broadcast telegram must have FFh (255) in the destination address field.  
The serial address is set using parameter P0308 in the CFW501.

## 2.3 BACNET PROFILE

The BACnet profile developed for the CFW501 is the B-ASC, with communication management services and data sharing that presents the following BIBBs (BACnet interoperability Building Blocks):

- DATA SHARING:
  - DS-RP-B: ReadProperty;
  - DS-WP-B: WriteProperty.
- DEVICE and NETWORK MGMT:
  - DM-DDB-B: WHO IS / I AM;
  - DM-RD-B: Device Management-Reinitialize Device-B.

### 2.3.1 ReadProperty (DS-RP-B)

A BACnet client (node performing a request to a server node) uses the ReadProperty service to obtain a BACnet object property value. This service allows reading access to the properties that have the R (reading) access type.

### **2.3.2 WriteProperty (DS-WP-B)**

A BACnet client uses the WriteProperty service to modify the value of a specific BACnet object property. This service allows writing access to the properties that have the W (write) or C (commandable) access type.

### **2.3.3 WHO IS / I AM (DM-DDB-B)**

The WHO IS / I AM service is used to identify the devices connected to the network. The WHO IS message is sent by the BACnet controller, and the nodes respond with an I AM message, informing their Object Identifier and address. The I AM message is transmitted in broadcast, and can be transmitted during initialization or continuous, according to the parameter P0764.

### **2.3.4 Device Management-Reinitialize Device-B (DM-RD-B)**

The Reinitialize Device service is used to remotely reinitialize the equipment, and uses a password to validate the service execution. The BACnet standard defines that the password is a string (set of ASCII characters) with up to 20 positions. The password used for the CFW501 remote reinitialization is the same one used to allow the access for parameter content modifications, informed in the parameter P0000. This password can be a number between 0000 and 9999.

The BACnet password for the CFW501 is a 4-character string. Therefore, the BACnet password can be a number between 0000 and 9999. E.g., considering that the CFW501 default password is 5, the remote reinitialization service will only be executed if the received password is "0005".

### 3 INTERFACE DESCRIPTION

The interfaces for serial communication RS232, RS485 or USB available for the CFW501 frequency inverter depend on the selected plug-in module for the product. Following are presented information about the connection and installation of the equipment, using different plug-in modules.

#### 3.1 PLUG-IN MODULES

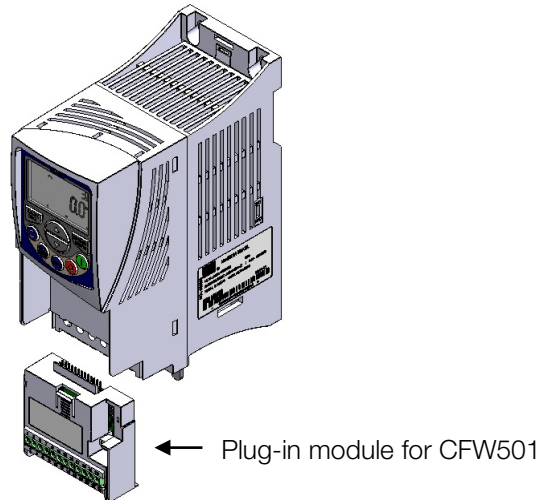


Figure 3.1: Example of CFW501 plug-in module

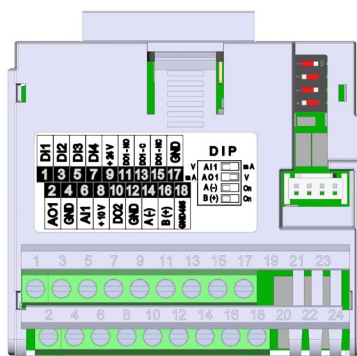
All plug-in modules for CFW501 frequency inverter have at least one standard RS485 interface, identified as Serial (1). This standard RS485 interface has two functions:

- Point to Point Connection with remote keypad.
- Connection via RS485 for network operation.

The selection of the function that will be used for the product is made using parameter P0312.

##### 3.1.1 RS485 connector for the standard plug-in module (CFW500-IOS)

For the standard plug-in module, the connection to the RS485 interface is available via the control terminal using the following pin assignment:



Pin	Name	Function
12	A-Line (-)	RxD/TxD negative
14	B-Line (+)	RxD/TxD positive
16	GND	0V isolated from the RS485 circuit

Table 2.1: RS485 connector pinout for standard plug-in module (CFW500-IOS)

### 3.2 PLUG-IN MODULES WITH RS485 INTERFACE AND ADDITIONAL INTERFACE

Depending on the installed plug-in module, CFW501 has up to two serial interfaces simultaneously, but only one can be the source of commands or references, the other is necessarily inactive or remote keypad as the selection of P0312.

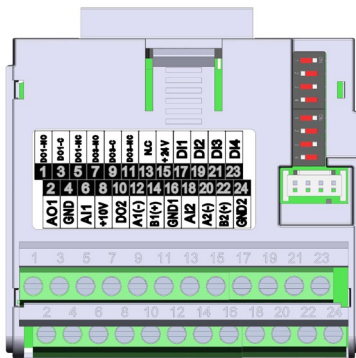
The Serial (1) Interface is the standard interface CFW501 and is present in all plug-in modules via the terminals of the standard RS485 port. The Serial (2) interface is present only in the plug-in modules described below:



**NOTE!**

It is not possible use the serial interfaces for communication with two different networks. The only allowed simultaneous operation is using Serial (1) connected to the remote keypad, and another programmed protocol to Serial (2).

#### 3.2.1 Plug-in module with two RS485 interfaces (CFW500-CRS485-B)



Pin	Name	Function
12	A-Line (-)	RxD/TxD negative – Serial (1)
14	B-Line (+)	RxD/TxD positive – Serial (1)
16	Ref.	0V for the RS485 circuit – Serial (1)

Pin	Name	Function
20	A-Line (-)	RxD/TxD negative – Serial (2)
22	B-Line (+)	RxD/TxD positive – Serial (2)
24	Ref.	0V for the RS485 circuit – Serial (2)

Table 3.2: Plug-in module with additional RS485

For this plug-in module, besides the standard RS485 interface, a second RS485 interface is available. This accessory allows simultaneous connection of a remote HMI in the standard RS485 serial interface, plus a programmable RS485 interface.

#### 3.2.2 Plug-in module with RS485 and USB interfaces (CFW500- CUSB)

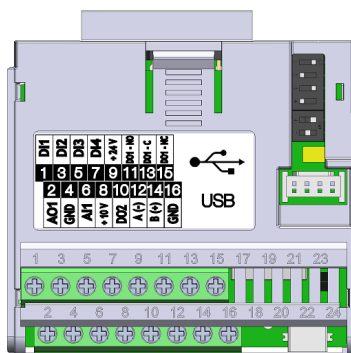
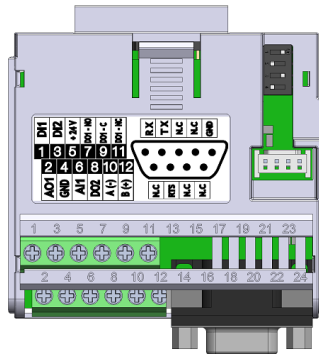


Figure 2.2: Plug-in module with USB connection

For this plug-in module, besides the standard RS485 interface, a USB interface with mini-USB connector is available. When connecting the USB interface, it will be recognized as a USB to serial converter, and a virtual COM port will be created<sup>1</sup>. Thus communication is made with the drive via this COM port. The USB accessory also provides the connection to the remote keypad via standard RS485 interface.

### 3.2.3 Plug-in module with RS485 and RS232 interfaces (CFW500-CRS232)



*Figure 2.3: Plug-in module with RS232 connection*

For this plug-in module, besides the standard RS485 interface, an RS232 interface is available. This accessory allows simultaneous connection of a remote HMI in the standard RS485 serial interface, plus a programmable RS232 interface.

## 3.3 RS485

### 3.3.1 RS485 Interface Characteristics

- The interface follows the EIA-485 standard.
- It allows communication baud rates from 9600 up to 38400 Kbit/s.
- The interface is electrically isolated and with differential signal, which grants more robustness against electromagnetic interference.
- It allows the connection of up to 32 devices to the same segment. More devices can be connected by using repeaters<sup>2</sup>.
- A maximum bus length of 1000 meters.
- 

### 3.3.2 Terminating resistor

It is necessary to enable a terminating resistor at both ends of the main bus for each segment of the RS485 network. There are switches in the CFW501 frequency inverter that can be activated to enable the terminating resistor. Refer to the plug-in installation guide for details.

---

<sup>1</sup> It is necessary to install the USB driver on the CD-ROM supplied with the product. The COM port number created depends on the availability in the operating system and, once connected, consult the hardware resources of the system to identify this port.

<sup>2</sup> The limit number of devices that can be connected to the network depends also on the used protocol.

### **3.3.3 Indications**

Details on the alarms, communications failures and communication states are made through the keypad (HMI) and product parameters.

### **3.3.4 Connection with the RS485 Network**

The following points must be observed for the connection of the device using the RS485 interface:

- It is recommended the use of a shielded cable with a twisted pair of wires.
- It is also recommended that the cable has one more wire for the connection of the reference signal (GND). In case the cable does not have the additional wire, then the GND signal must be left disconnected.
- The cable must be laid separately (and far away if possible) from the power cables.
- All the network devices must be properly grounded, preferably at the same ground connection. The cable shield must also be grounded.
- Enable the termination resistors only at two points, at the extremes of the main bus, even if there are derivations from the bus.

## 4 PROGRAMMING

Next, the CFW501 frequency inverter parameters related to the BACnet communication will be presented.

### 4.1 SYMBOLS FOR THE PROPERTIES DESCRIPTION

RO	Reading only parameter
CFG	Parameter that can be changed only with a stopped motor.

#### P0105 – 1<sup>ST</sup>/2<sup>ND</sup> 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 CFW501 frequency inverter local and remote situations. In order that the device be controlled through the BACnet interface, the options 'serial' available in these parameters, must be selected.

The detailed description of these parameters is found in the CFW501 programming manual.

#### P0308 – SERIAL ADDRESS

<b>Range:</b>	0 to 255	<b>Default:</b> 1
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

#### Description:

It allows programming the address used for the inverter serial communication. It is necessary that each device in the network has an address different from all the others. The valid addresses for this parameter depend on the protocol programmed in P0312:

- HMI → programming needn't address.
- Modbus RTU → valid addresses: 1 to 247.
- BACnet → valid addresses: 0 to 254.
- N2 → valid addresses: 1 to 255.
- SymbiNet → valid addresses: 1 to 63.



**P0310 – SERIAL BAUD RATE**

<b>Range:</b>	0 = 9600 bits/s 1 = 19200 bits/s 2 = 38400 bits/s	<b>Default: 1</b>
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows programming the baud rate for the serial communication interface, in bits per second. This baud rate must be the same for all the devices connected to the network.


**NOTE!**

To use the RS485 interface with the remote keypad it is not necessary to set the baud rate. This setting is necessary only with the other serial protocols, for both the standard interface (1) as the additional interface (2).

**P0311 – SERIAL INTERFACE BYTE CONFIGURATION**

<b>Range:</b>	0 = 8 data bits, no parity, 1 stop bit 1 = 8 data bits, even parity, 1 stop bit 2 = 8 data bits, odd parity, 1 stop bit 3 = 8 data bits, no parity, 2 stop bits 4 = 8 data bits, even parity, 2 stop bits 5 = 8 data bits, odd parity, 2 stop bits	<b>Default: 1</b>
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows programming the number of data bits, parity and stop bits of the serial interface bytes. This configuration must be identical for all the devices connected to the network.


**NOTE!**

The option 0 must be selected for the BACnet and N2 protocol.


**NOTE!**

To use the RS485 interface with the remote keypad it is not necessary to set the byte configuration. This setting is necessary only with the other serial protocols, for both the standard interface (1) as the additional interface (2).

**P0312 – SERIAL PROTOCOL**

<b>Range:</b>	0 = HMI (1) 1 = SymbiNet (1) 2 = Modbus RTU (1) 3 = BACnet (1) 4 = N2 (1) 5 = Master Modbus RTU (1) and Modbus RTU (2) 6 = HMI (1) and Modbus RTU (2) 7 = Modbus RTU (2) 8 = HMI (1) and BACnet (2) 9 = BACnet (2) 10 = HMI (1) and N2 (2) 11 = N2 (2) 12 = HMI (1) and Master Modbus RTU (2) 13 = Modbus RTU (1) and Master Modbus RTU (2) 14 = HMI (1) and SymbiNet (2) 15 = SymbiNet (2)	<b>Default: 2</b>
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows selecting the desired protocol for the serial interface.

*Table 4.1: P0312 options*

Option	Description
0 = HMI (1)	For the standard RS485 interface (1), this option selects the remote HMI communication protocol.
1 = SymbiNet (1)	For the standard serial interface (1), this option selects SymbiNet as the communication protocol.
2 = Modbus RTU (1)	For the standard RS485 interface (1), this option selects Modbus RTU slave as the communication protocol.
3 = BACnet (1)	For the standard RS485 interface (1), this option selects BACnet MS/TP the communication protocol.
4 = N2 (1)	For the standard RS485 interface (1), this option selects N2 the communication protocol.
5 = Master Modbus RTU (1) and Modbus RTU (2)	For the interface modules with more than one serial interface, this option allows to use the standard interface (1) as master Modbus RTU, and simultaneously to use the additional interface as slave Modbus RTU.
6 = HMI (1)/Modbus RTU (2)	For the interface modules with more than one serial interface (example: CFW500-CUSB, etc.), this option allows to use the remote HMI connected to standard interface (1), and simultaneously, to use Modbus RTU slave protocol at the additional serial interface (2).
7 = Modbus RTU (2)	For the additional RS485 interface (2), this option selects Modbus RTU slave communication protocol. The standard serial interface (1) remains disabled.
8 = HMI (1)/BACnet (2)	For the interface modules with more than one serial interface (example: CFW500-CUSB, etc.), this option allows to use the remote HMI connected to standard interface (1), and simultaneously, to use BACnet MS/TP protocol at the additional serial interface (2).
9 = BACnet (2)	For the additional RS485 interface (2), this option selects BACnet MS/TP communication protocol. The standard serial interface (1) remains disabled.
10 = HMI (1)/N2 (2)	For the interface modules with more than one serial interface (example: CFW500-CUSB, etc.), this option allows to use the remote HMI connected to standard interface (1), and simultaneously, to use N2 protocol at the additional serial interface (2).
11 = N2 (2)	For the additional RS485 interface (2), this option selects N2

	communication protocol. The standard serial interface (1) remains disabled.
12 = HMI (1) and Master Modbus RTU (2)	For the interface modules with more than one serial interface, this option allows to use the remote HMI connected to standard interface (1), and simultaneously to use the drive as Master Modbus RTU at the additional serial interface (2).
13 = Modbus RTU (1) and Master Modbus RTU (2)	For the interface modules with more than one serial interface, this option allows to use the standard interface (1) as slave Modbus RTU, and simultaneously to use the drive as Master Modbus RTU at the additional serial interface (2).
14 = HMI (1) and SymbiNet (2)	For the interface modules with more than one serial interface, this option allows to use the remote HMI connected to standard interface (1), and simultaneously to use the additional interface (2) as SymbiNet.
15 = SymbiNet (2)	For the additional serial interface (2), this option selects SymbiNet communication protocol. The standard serial interface (1) remains disabled.

**P0313 – COMMUNICATION ERROR ACTION**

<b>Range:</b> 0 = Inactive 1 = Disable via Run/Stop 2 = Disable via General Enable 3 = Change to Local 4 = Change to Local keeping commands and reference 5 = Causes a Fault  <b>Properties:</b> CFG <b>Access groups via HMI:</b> NET	<b>Default: 1</b>
---	-------------------

**Description:**

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

*Table 4.2: P0313 options*

Options	Description
0 = Inactive	No action is taken and the drive 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 drive is disabled by removing the General Enabling and the motor coasts to stop.
3 = Change to Local	The drive commands change to Local.
4 = Change to Local keeping commands and reference	The drive commands change to Local, but the status of the enabling and speed reference commands received via network are kept, providing that the drive has been programmed to use in Local mode the commands via HMI, or 3-wire start/stop and speed reference via either HMI or electronic potentiometer.
5 = Causes a Fault	Instead of an alarm, the communication error causes an drive fault, so that a drive fault reset becomes necessary in order to restore normal operation.

The following events are considered communication errors:

Serial communication (RS485):

- A128 alarm/F228 fault: Serial communication timeout

The actions described in this parameter are executed by means of the automatic writing of the selected actions in the respective bits of the interface control words. Therefore, in order that the commands written in this parameter be effective, it is necessary that the device be programmed to be controlled via the used network interface (with exception of option “Causes a Fault”, which blocks the equipment even if it is not controlled by network). This programming is achieved by means of parameters P0220 to P0228.

**P0314 – SERIAL WATCHDOG**

<b>Range:</b>	0.0 to 999.0s	<b>Default:</b> 0.0
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows programming a time limit for the detection of serial interface communication error. If the frequency inverter remains without receiving valid telegrams longer than the time programmed in this parameter, it will be considered that a communication error has occurred, the alarm A128 will be showed on the HMI and the option programmed in P0313 will be executed.

After being powered up, the frequency inverter starts counting this time from the first received valid telegram. The value 0.0 disables this function.

**P0316 – SERIAL INTERFACE STATUS**

<b>Range:</b>	0 = Inactive 1 = Active 2 = Watchdog error	<b>Default:</b> -
<b>Properties:</b>	RO	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows identifying whether the RS485 serial interface board is properly installed, and whether the serial communication presents errors.

*Table 4.3: P0316 options*

Options	Description
0 = Inactive	Inactive serial interface. It occurs when the device does not have the RS485 board installed.
1 = Active	Installed and acknowledged RS485 interface board.
2 = Watchdog error	The serial interface is active, but a serial communication error has been detected - A128 alarm/F228 fault.

**P0680 – STATUS WORD**

<b>Range:</b>	0000h to FFFFh	<b>Default:</b> -
<b>Properties:</b>	RO	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows the device status monitoring. Each bit represents a specific status:

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	Fault condition	(PID) Automatic	Undervoltage	LOC/REM	JOG	Speed direction	Active General Enable	Motor Running	Alarm condition	In configuration mode	Second ramp	Active quick stop	Bypass	Fire Mode	Run command	Reserved

**Table 4.4: P0680 parameter bit functions**

<b>Bits</b>	<b>Values</b>
Bit 0	Reserved.
Bit 1 Run command	0: Run command is inactive. 1: Run command is active.  This bit is mapped in the BV1 object
Bit 2 Fire Mode	0: Drive is not in Fire Mode. 1: Drive is in Fire Mode.  This bit is mapped in the BV2 object
Bit 3 Bypass	0: Drive is not in Bypass mode. 1: Drive is in Bypass mode.  This bit is mapped in the B3 object
Bit 4 Active quick stop	0: The fast stop command is not active. 1: The drive is executing the fast stop command.  This bit is mapped in the BV4 object
Bit 5 Second ramp	0: The drive is configured to use the first ramp values, programmed in P0100 and P0101, as the motor acceleration and deceleration ramp times. 1: The drive is configured to use the second ramp values, programmed in P0102 and P0103, as the motor acceleration and deceleration ramp times.  This bit is mapped in the BV5 object
Bit 6 In configuration mode	0: The drive is operating normally. 1: The drive is in the configuration mode. It indicates a special condition during which the drive cannot be enabled: Executing the self-tuning routine Executing the oriented start-up routine Executing the HMI copy function Executing the flash memory card self-guided routine There is a parameter setting incompatibility There is no power at the drive power section  This bit is mapped in the BV6 object
Bit 7 Alarm condition	0: The drive is not in alarm condition. 1: The drive is in alarm condition. Note: The alarm number can be read by means of the parameter P0048 – Present Alarm.  This bit is mapped in the BV7 object
Bit 8 Motor Running	0: The motor is stopped. 1: The drive is running the motor at the set point speed, or executing either the acceleration or the deceleration ramp.  This bit is mapped in the BV8 object
Bit 9 Active General Enable	0: General Enable is not active. 1: General Enable is active and the drive is ready to run the motor.  This bit is mapped in the BV9 object
Bit 10 Speed direction	0: The motor is running in the reverse direction. 1: The motor is running in the forward direction.  This bit is mapped in the BV10 object
Bit 11 JOG	0: Inactive JOG function. 1: Active JOG function.  This bit is mapped in the BV11 object
Bit 12 LOC/REM	0: Drive in Local mode. 1: Drive in Remote mode.  This bit is mapped in the BV12 object
Bit 13 Undervoltage	0: No Undervoltage. 1: With Undervoltage.  This bit is mapped in the BV13 object

Bit 14 Manual/ Automatic	0: PID in manual mode. 1: PID in Automatic mode.  This bit is mapped in the BV14 object
Bit 15 Fault condition	0: The drive is not in a fault condition. 1: The drive has detected a fault. Note: The fault number can be read by means of the parameter P0049 – Present Fault.  This bit is mapped in the BV15 object

### P0681 – MOTOR SPEED IN 13 BITS

<b>Range:</b> - 32768 to 32767	<b>Default:</b> -
<b>Properties:</b> RO	
<b>Access groups via HMI:</b> NET	

**Description:**

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

- P0681 = 0000h (0 decimal) → motor speed = 0
- 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 motor and 1800 rpm of synchronous speed if the value read is 2048 (0800h), then, to obtain the speed in rpm one must calculate:

8192 => 1800 rpm 2048 => Speed in rpm
$\text{Speed in rpm} = \frac{1800 \times 2048}{8192}$
Speed in rpm = 450 rpm

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

This parameter is mapped in the ANV16 object.

### P0682 – SERIAL CONTROL WORD

<b>Range:</b> 0000h to FFFFh	<b>Default:</b> 0000h
<b>Properties:</b> -	
<b>Access groups via HMI:</b> NET	

**Description:**

It is the device BACnet interface control word. This parameter can only be changed via serial interface. For the other sources (HMI, etc.) it behaves like a read-only parameter.

In order to have those commands executed, it is necessary to program the equipment to be controlled via serial. This programming is achieved by means of parameters P0105 and P0220 to P0228.

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

Bits	15	14	13	13 to 8	7	6	5	4	3	2	1	0
Function	Reserved	External PID controller 1	Main PID controller	Reserved	Fault reset	Quick stop	Second ramp	LOC/REM	JOG	Speed direction	General enable	Run/Stop

**Table 4.5: P0682 parameter 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.  This bit is mapped in the object BV16
Bit 1 General enable	0: It disables the drive, interrupting the supply for the motor. 1: It enables the drive allowing the motor operation.  This bit is mapped in the object BV17
Bit 2 Speed direction	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.  This bit is mapped in the object BV18
Bit 3 JOG	0: It disables the JOG function. 1: It enables the JOG function.  This bit is mapped in the object BV19
Bit 4 LOC/REM	0: The drive goes to the Local mode. 1: The drive goes to the Remote mode.  This bit is mapped in the object BV20
Bit 5 Second ramp	0: The drive uses the first ramp values, programmed in P0100 and P0101, as the motor acceleration and deceleration ramp times. 1: The drive is configured to use the second ramp values, programmed in P0102 and P0103, as the motor acceleration and deceleration ramp times.  This bit is mapped in the object BV21
Bit 6 Quick stop	0: It does not execute the quick stop command. 1: It executes the quick stop command. Note: This function is not allowed with control types (P0202) V/f or VVV.  This bit is mapped in the object BV22
Bit 7 Fault reset	0: No function. 1: If in a fault condition, then it executes the reset.  This bit is mapped in the object BV23
Bits 8 to 12	Reserved.
Bit 13 Main PID controller	0: Automatic. 1: Manual.  This bit is mapped in the object BV28
Bit 14 External PID controller 1	0: Automatic. 1: Manual.  This bit is mapped in the object BV29
Bit 15	Reserved.

**P0683 – SERIAL SPEED REFERENCE**

<b>Range:</b>	-32768 to 32767	<b>Default:</b> 0
<b>Properties:</b>	-	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows programming the motor speed reference via the BACnet interface. This parameter can only be changed via serial interface. For the other sources (HMI, etc.) it behaves like a read-only parameter.

In order that the reference written in this parameter be used, it is necessary that the drive be programmed to use the speed reference via serial. This programming is achieved by means of parameters P0221 and P0222.

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

- P0683 = 0000h (0 decimal) → speed reference = 0
- P0683 = 2000h (8192 decimal) → speed reference = synchronous speed

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

$\begin{aligned} 1800 \text{ rpm} &\Rightarrow 8192 \\ 900 \text{ rpm} &\Rightarrow 13 \text{ bit reference} \end{aligned}$
$13 \text{ bit reference} = \frac{900 \times 8192}{1800}$
$13 \text{ bit reference} = 4096 \Rightarrow \text{Value corresponding to 900 rpm in a 13 bit scale}$

This parameter also accepts negative values to revert the motor speed direction. The reference speed direction, however, depends also on the control word - P0682 - bit 2 setting:

- Bit 2 = 1 and P0683 > 0: reference for forward direction
- Bit 2 = 1 and P0683 < 0: reference for reverse direction
- Bit 2 = 0 and P0683 > 0: reference for reverse direction
- Bit 2 = 0 and P0683 < 0: reference for forward direction

This parameter is mapped in the ANV17 object.

**P0695 – DIGITAL OUTPUT SETTING**

<b>Range</b>	0000h to 001Fh	<b>Default:</b> 0000h
<b>Properties:</b>	-	
<b>Access groups via HMI:</b>	NET	

**Description:**

It allows the control of the digital outputs by means of the network interfaces. This parameter cannot be changed via HMI.

Each bit of this parameter corresponds to the desired value for one 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 P0279.



Bits	15 to 5	4	3	2	1	0
Function	Reserved	DO5 setting	DO4 setting	DO3 setting	DO2 setting	DO1 setting

**Table 4.6:** P0695 parameter bit functions

Bits	Values
Bit 0 DO1 setting	0: DO1 output open. 1: DO1 output closed.  This bit is mapped in the BOUT0 object
Bit 1 DO2 setting	0: DO2 output open. 1: DO2 output closed.  This bit is mapped in the BOUT1 object
Bit 2 DO3 setting	0: DO3 output open. 1: DO3 output closed.  This bit is mapped in the BOUT2 object
Bit 3 DO4 setting	0: DO4 output open. 1: DO4 output closed.  This bit is mapped in the BOUT3 object
Bit 4 DO5 setting	0: DO5 output open. 1: DO5 output closed.  This bit is mapped in the BOUT4 object
Bits 5 to 15	Reserved


**NOTE!**

Some of the digital outputs may not be available depending on the plug-in module.

**P0696 – VALUE 1 FOR ANALOG OUTPUTS**
**P0697 – VALUE 2 FOR ANALOG OUTPUTS**
**P0698 – VALUE 3 FOR ANALOG OUTPUTS**

**Range:** -32768 to 32767 **Default:** 0

**Properties:** -

**Access groups via HMI:** NET

**Description:**

They allow the control of the analog outputs by means of network interfaces. These parameters cannot be changed via HMI.

The value written in these parameters is used as the analog output value, providing that the function for the desired analog output be programmed for “P0696 / P0697 / P0698 value”, at the parameters P0251, P0254, P0257.

The value must be written in a 15-bit scale (7FFFh = 32767)<sup>3</sup> to represent 100 % of the output desired value, i.e.:

- P0696 = 0000h (0 decimal) → analog output value = 0 %

<sup>3</sup> For the actual output resolution, refer to the product manual.

- 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. For instance, to control the analog output 1 via serial, the following programming must be done:

- Choose a parameter from P0696, P0697, P0698 to be the value used by the analog output 1. For this example, we are going to select P0696.
- Program the option “P0696 value” as the function for the analog output 1 in P0254.
- Using the network interface, write in P0696 the desired value for the analog output 1, between 0 and 100 %, according to the parameter scale.

BACnet objects of the ANALOG OUTPUT type mold the analog outputs, where:

- ANO0 - P0696.
- ANO1 - P0697.
- ANO2 - P0698.


**NOTE!**

For CFW501 frequency inverter, the analog output 3 represents the frequency output (FO).

**P0760 – BACNET EQUIPMENT INSTANCE – HIGH PART**

<b>Range:</b>	1 a 247	<b>Default: 1</b>
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

**Description:**

It defines the high part of the BACnet equipment instance.


**NOTE!**

Refer to the parameter P0761 description for more details.

**P0761 – BACNET EQUIPMENT INSTANCE – LOW PART**

<b>Range:</b>	0 to 9999	<b>Default: 0</b>
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

**Description:**

It defines the low part of the BACnet equipment instance.

The BACnet standard defines that the equipment instance must be unique in the network and it must present a value between 0 and 4194304. The BACnet instance will compose the Object Identifier property of the DEVICE object, which defines the equipment characteristics in the network.

The BACnet instance can be defined automatically or manually:

**Automatically:**

If the values of parameters P0760 and P0761 are set 0 (default value), the inverter will automatically create the BACnet instance based on the vendor BACnet ID (WEG BACnet ID = 359) and the serial address. For this configuration, the user must only inform the serial address at the parameter P0308.

**BACnet instance = BACnet ID + Serial address**

Example 1: serial address = 102

Instance = 359102

Example 2: serial address = 15

Instance = 359015


**NOTE!**

The instance created automatically is not showed at the parameters P0760 and P0761, which remain with the value 0.

**Manual:**

The BACnet instance is defined by using the parameters P0760 and P0761. The P0760 parameter content is multiplied by 10000 and added to the P0761 parameter content.

Example 1: Instance = 542786

$$542786 / 10000 = 54.2786$$

P0760 = 54 (whole part)

P0761 = 2786 (fractional part)

Example 2: Instance = 66789

$$66789 / 10000 = 6.6789$$

P0760 = 6 (whole part)

P0761 = 6789 (fractional part)

Example 3: Instance = 35478

$$35478 / 10000 = 3.5478$$

P0760 = 3 (whole part)

P0761 = 5478 (fractional part)


**NOTE!**

The parameters P0760 and P0761 allow adjusting a maximum value of 4199999. However, the maximum instance value will be 4194304.


**NOTE!**

The equipment must be initialized when the contents of parameters P0760 and P0761 are changed.

**P0762 – MAXIMUM MASTER NUMBER**

**Range:** 0 to 127

**Default:** 127

**Properties:** CFG

**Access groups via HMI:** NET

**Description:**

It allows programming the longest address used by a master in the BACnet network, optimizing the communication. All the network devices must be set with the same value in this parameter.

With the standard value (127) for this parameter, any address set for the device can participate in the communication. However, that will make the devices in the network send requests searching for devices in all the address range, hindering the data exchange cycle and the entrance of new devices in the network. By limiting the longest address accepted, addresses above this value are ignored, preventing the search for unnecessary addresses and optimizing the communications.

It is recommended that the devices on the network be addressed in sequence from address 1 on, and that this parameter be set with the same value as the last address of the network.


**NOTE!**

The equipment must be initialized when the content of the parameter P0762 is changed.

**P0763 – MAXIMUM NUMBER OF MS/TP FRAMES**

<b>Range:</b>	1 to 65535	<b>Default:</b> 1
<b>Properties:</b>	CFG	
<b>Access groups via HMI:</b>	NET	

**Description:**

It defines the number of telegrams that the node can transmit when it receives the token. Then it must transmit the token to the next node.


**NOTE!**

The equipment must be initialized when the content of the parameter P0763 is changed.

**P0764 – I AM TRANSMISSION**

<b>Range:</b>	0 = Power Up 1 = Continuous	<b>Default:</b> 0
<b>Properties:</b>	RO	
<b>Access groups via HMI:</b>	NET	

**Description:**

The I AM telegram is used to identify the node in the BACnet network. When the option 1, periodically, is selected, then an I AM telegram is sent every 200 ms. When the option 0, power up, is selected, then a I AM telegram is sent in the Power up just.

**P0765 – NUMBER OF RECEIVED TOKENS**

<b>Range:</b>	0 to 65535	<b>Default:</b> -
<b>Properties:</b>	RO	
<b>Access groups via HMI:</b>	NET	

**Description:**

It is the counter of the number of tokens received from other BACnet nodes. It allows the serial communication verification.

## 5 BACNET OBJECT MODELING

A BACnet object represents physical or virtual equipment information, as a digital input or parameters. The CFW501 presents the following object types:

- ANALOG INPUT;
- ANALOG OUTPUT;
- ANALOG VALUE;
- BINARY INPUT;
- BINARY OUTPUT;
- BINARY VALUE;
- DEVICE OBJECT.

Each object type defines a data structure composed by properties that allow the access to the object information. The table 5.1 shows the implemented properties for each CFW501 object type.

*Table 5.1: Properties of the BACnet objects*

Property	DEVICE	ANALOG INPUT	ANALOG OUTPUT	ANALOG VALUE	BINARY INPUT	BINARY OUTPUT	BINARY VALUE
Object Identifier	X	X	X	X	X	X	X
Object Name	X	X	X	X	X	X	X
Object Type	X	X	X	X	X	X	X
System Status	X						
Vendor Name	X						
Vendor Identifier	X						
Model Name	X						
Firmware Revision	X						
Application Software Version	X						
Description	X	X	X	X	X	X	X
Protocol Version	X						
Protocol Revision	X						
Protocol service supported	X						
Protocol object types Supported	X						
Object List	X						
Max APDU Len Accepted	X						
Segmentation Supported	X						
APDU timeout	X						
Number of APDU retries	X						
Max Master	X						
Max info frames	X						
Device Address Binding	X						
Database revision	X						
Present Value		X	X	X	X	X	X
Status Flags		X	X	X	X	X	X
Event State		X	X	X	X	X	X
Out of Service		X	X	X	X	X	X
Units		X	X	X	X	X	X
Priority Array			X	X□		X	X□
Relinquish Default			X	X□		X	X□
Polarity					X	X	

\* The Priority Array and Relinquish Default properties are available for objects with type C (Commandable) access.

Each object presents an identifier unique in the network, called Object Identifier. The Object Identifier property is composed by two parts:



The Object Type values are defined by the BACnet specification, and object instance is defined by the manufacturer for each object available for communication.


Regarding the Present Value property, each object can present one of the following access types:

- R** Read-only
- C** Commandable – uses priority array.
- W** Write-only
- R/W** Read/Write – without priority array

The Commandable (C) access type presents a 16-level priority array, where priority 1 is the highest and 16 the lowest. If all the priorities are disabled (NULL), the value of the Relinquish Default property is given to the Present Value property.

**5.1 BACNET OBJECTS**

The parameters of frequency inverter CFW501 are mapped through BACnet objects which are described below.



**NOTE!**  
Check the product manual for parameters details.

**5.1.1 ANALOG INPUT OBJECT**

It represents an analog input that can have its value read by the controller. CFW501 ANALOG INPUT type objects are described in the table 5.2. The ANALOG INPUT objects are of the REAL type.

*Table 5.2: ANALOG INPUT objects*

Object Identifier	Object name	Related Parameter	Unit	Access type
AI0	AI1 Value	P0018	%	R
AI1	AI2 Value	P0019	%	R
AI2	AI3 Value	P0020	%	R

**5.1.2 ANALOG OUTPUT OBJECT**

It represents an analog output that can have its value written by the controller. CFW501 ANALOG OUTPUT type objects are described in the table 5.3. The ANALOG OUTPUT objects are of the REAL type.

*Table 5.3: ANALOG OUTPUT objects*

Object Identifier	Object name	Related Parameter	Unit	Access type
AO0	AOx Value 1	P0696		C
AO1	AOx Value 2	P0697		C
AO2	AOx Value 3	P0698		C

### 5.1.3 ANALOG VALUE OBJECT

They represent system control parameters that can be read, written or commanded by the controller. CFW501 ANALOG VALUE type objects are described in the table 5.4. The ANALOG VALUE objects are of the REAL type.

*Table 5.4: ANALOG VALUE objects*

Object Identifier	Object name	Related Parameter	Unit	Access type
AV0	Motor Speed	P0002	rpm	R
AV1	Motor Current	P0003	A	R
AV2	DC Link Voltage (Ud)	P0004	V	R
AV3	Motor Frequency	P0005	Hz	R
AV4	Motor Voltage	P0007	V	R
AV5	Motor Torque	P0009	%	R
AV6	Output Power	P0010	kW	R
AV7	Heatsink Temperature	P0030	°C	R
AV9	Time Powered	P0042	h	R
AV10	Time Enabled	P0043	h	R
AV11	kWh Output Energy	P0044	kWh	R
AV12	Present Alarm	P0048		R
AV13	Present Fault	P0049		R
AV14	Acceleration Time	P0100	s	C
AV15	Deceleration Time	P0101	s	C
AV16	Speed in 13 bits	P0681		R
AV17	Serial/USB Speed Ref.	P0683		C
AV18	SoftPLC Parameter 3	P1012		C
AV19	SoftPLC Parameter 4	P1013		C
AV20	SoftPLC Parameter 16	P1025		C
AV21	Main PID Aut. Setpoint	P1011		C
AV22	Main PID Man. Setpoint	P1014	%	C
AV23	Main PID Feedback	P1015		R
AV24	Main PID Output	P1016	%	R
AV25	External PID Auto Setpoint	P1060		C
AV26	External PID Man. Setpoint	P1061	%	C
AV27	External PID Feedback	P1062		R
AV28	External PID Output	P1063	%	R
AV100	Mailbox: param. number	-		R/W
AV101	Mailbox: param. value	-		R/W

The CFW501 programming manual presents the detailed description of each parameter.

### 5.1.4 BINARY INPUT OBJECT

It represents a physical digital input that can have its status read by the controller. CFW501 BINARY INPUT type objects are described in the table 5.5.

*Table 5.5: BINARY INPUT objects*

Object Identifier	Object name	Related Parameter	States Description (1 / 0)	Access type
BI0	DI1	P0012 – Bit 0	On/Off	R
BI1	DI2	P0012 – Bit 1	On/Off	R
BI2	DI3	P0012 – Bit 2	On/Off	R
BI3	DI4	P0012 – Bit 3	On/Off	R
BI4	DI5	P0012 – Bit 4	On/Off	R
BI5	DI6	P0012 – Bit 5	On/Off	R
BI6	DI7	P0012 – Bit 6	On/Off	R
BI7	DI8	P0012 – Bit 7	On/Off	R

### 5.1.5 BINARY OUTPUT OBJECT

It represents a physical digital output that can have its status changed by the controller. CFW501 BINARY OUTPUT type objects are described in the table 5.6.

*Table 5.6: BINARY OUTPUT objects*

Object Identifier	Object name	Related Parameter	States Description (1 / 0)	Access type
BO0	DO1	P0695 – Bit 0	On/Off	C
BO1	DO2	P0695 – Bit 1	On/Off	C
BO2	DO3	P0695 – Bit 2	On/Off	C
BO3	DO4	P0695 – Bit 3	On/Off	C
BO4	DO5	P0695 – Bit 4	On/Off	C

### 5.1.6 BINARY VALUE OBJECT

They represent system control parameter bits that can be read, written or commanded by the controller. CFW501 BINARY VALUE type objects are described in the table 5.7.



*Table 5.7: BINARY VALUE objects*

Object Identifier	Object name	Related Parameter	State (1 / 0)	Access type
BV1	Run Command	P0680 Bit 1	On/Off	R
BV2	Fire mode	P0680 Bit 2	On/Off	R
BV3	Bypass	P0680 Bit 3	On/Off	R
BV4	Quick Stop	P0680 Bit 4	Active/Inactive	R
BV5	2nd Ramp	P0680 Bit 5	On/Off	R
BV6	Config. Mode	P0680 Bit 6	Config/Normal	R
BV7	Alarm	P0680 Bit 7	Alarm/No Alarm	R
BV8	Running	P0680 Bit 8	Running/Stopped	R
BV9	Enabled	P0680 Bit 9	Enabled/Disabled	R
BV10	Forward	P0680 Bit 10	Forward/Reverse	R
BV11	JOG	P0680 Bit 11	On/Off	R
BV12	Remote	P0680 Bit 12	Remote/Local	R
BV13	Subvoltage	P0680 Bit 13	Subvoltage/No	R
BV14	Automatic(PID)	P0680 Bit 14	Auto/Manual	R
BV15	Fault	P0680 Bit 15	Fault/No Fault	R
BV16	Ramp Enable	P0682 Bit 0	Run/Stop	C
BV17	General Enable	P0682 Bit 1	Enable/Disable	C
BV18	Run Forward	P0682 Bit 2	Forward/Reverse	C
BV19	JOG Enable	P0682 Bit 3	On/Off	C
BV20	Remote	P0682 Bit 4	Remote/Local	C
BV21	2nd Ramp	P0682 Bit 5	On/Off	C
BV22	Quick Stop	P0682 Bit 6	On/Off	C
BV23	Fault Reset	P0682 Bit 7	Reset/Off	C
BV29	Intern PID	P0682 Bit 13	Manual/Auto	C
BV30	Extern PID	P0682 Bit 14	Manual/Auto	C
BV100	Mailbox: exec. read	-	On/Off	R/W
BV101	Mailbox: exec. write	-	On/Off	R/W


**NOTE!**

In order to have those commands executed, it is necessary that the inverter be programmed to be controlled via serial. This programming is achieved by means of parameters P0105 and P0220 to P0228.

**5.1.7 DEVICE OBJECT**

The DEVICE object informs the BACnet equipment characteristics. Its properties represent those characteristics and they are described in the table 5.1. There must be only one DEVICE object at each BACnet equipment.

**5.1.8 Mailbox**

It is a structure that allows reading and writing CFW501 parameters. The following objects compose this structure:

*Table 5.8: Mailbox objects*

<b>Object Identifier</b>	<b>Object name</b>	<b>Description</b>	<b>Access type</b>
AV100	Mailbox: param. number	Informs the parameter number	R/W
AV101	Mailbox: param. value	Informs the datum read from or written into the parameter	R/W
BV100	Mailbox: exec. read	Parameter reading command	R/W
BV101	Mailbox: exec. write	Parameter writing command	R/W

Procedure for reading a parameter via Mailbox:

1. Write the parameter number in the AV100 object Present Value property;
2. Write 1 in the BV100 object Present Value property;
3. Read the parameter value in the AV101 object Present Value property.

Procedure for writing a parameter via Mailbox:

1. Write the parameter number in the AV100 object Present Value property;
2. Write the value to the parameter in the AV101 object Present Value property;
3. Write 1 in the BV101 object Present Value property.

## 6 FAULTS AND ALARMS RELATED TO THE BACNET COMMUNICATION

### A128/F228 – TIMEOUT FOR SERIAL COMMUNICATION

**Description:**

It is the only alarm/fault related to the serial communication indicates that the equipment stopped receiving valid serial telegrams for a period longer than the one programmed in P0314.

**Operation:**

The parameter P0314 allows programming a period of time during which the equipment must receive at least one valid telegram via the RS485 serial interface – with address and error-checking field correct – otherwise, it will be considered that there was any problem in the serial communication. The time counting initiates after the reception of the first valid telegram. This function can be used by any serial protocol supported by the equipment.

After the serial communication timeout has been identified, the A128 alarm or F228 fault message will be showed on the HMI, depending on the P0313 programming. For alarms, if the communication is reestablished and new valid telegrams are received, the alarm indication will be removed from the HMI.

**Possible Causes/Correction:**

- Verify factors that could cause failures in the communication (cables, installation, and grounding).
- Make sure that the master sends telegrams to the equipment in intervals shorter than the programmed in P0314.
- Disable this function at P0314.



WEG Drives & Controls - Automação LTDA.  
Jaraguá do Sul – SC – Brazil  
Phone 55 (47) 3276-4000 – Fax 55 (47) 3276-4020  
São Paulo – SP – Brazil  
Phone 55 (11) 5053-2300 – Fax 55 (11) 5052-4212  
automacao@weg.net  
[www.weg.net](http://www.weg.net)