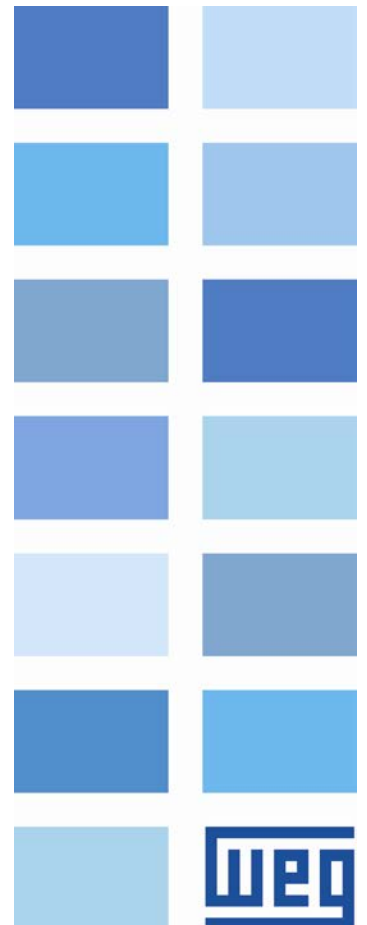


# CANopen

CFW500

## Manual del Usuario





# **Manual del Usuario CANopen**

Serie: CFW500

Idioma: Español

N ° del Documento: 10002253298 / 00

Fecha de la Publicación: 06/2013

# CONTENIDOS

|   |           |
|---|-----------|
| <b>CONTENIDOS.....</b>  | <b>3</b>  |
| <b>A RESPECTO DEL MANUAL.....</b>                               | <b>6</b>  |
| <b>ABREVIACIONES Y DEFINICIONES.....</b>                        | <b>6</b>  |
| <b>REPRESENTACIÓN NUMÉRICA.....</b>                             | <b>6</b>  |
| <b>DOCUMENTOS.....</b>  | <b>6</b>  |
| <b>1 INTRODUCCIÓN A LA COMUNICACIÓN CANOPEN.....</b>            | <b>7</b>  |
| <b>1.1 CAN.....</b>   | <b>7</b>  |
| 1.1.1 Frame de Datos.....                                       | 7         |
| 1.1.2 Frame Remoto.....   | 7         |
| 1.1.3 Acceso a la Red.....                                      | 7         |
| 1.1.4 Control de Errores.....                                   | 7         |
| 1.1.5 CAN y CANopen.....  | 8         |
| <b>1.2 CARACTERÍSTICAS DE LA RED CANOPEN.....</b>               | <b>8</b>  |
| <b>1.3 MEDIO FÍSICO.....</b>                                    | <b>8</b>  |
| <b>1.4 DIRECCIÓN EN LA RED CANOPEN.....</b>                     | <b>8</b>  |
| <b>1.5 ACCESO A LOS DATOS.....</b>                              | <b>8</b>  |
| <b>1.6 TRANSMISIÓN DE DATOS.....</b>                            | <b>8</b>  |
| <b>1.7 OBJETOS RESPONSABLES POR LA COMUNICACIÓN – COBS.....</b> | <b>9</b>  |
| <b>1.8 COB-ID.....</b>  | <b>9</b>  |
| <b>1.9 ARCHIVO EDS.....</b>                                     | <b>10</b> |
| <b>2 ACCESORIO PARA COMUNICACIÓN CANOPEN.....</b>               | <b>11</b> |
| <b>2.1 KIT CAN-01.....</b>                                      | <b>11</b> |
| <b>2.2 TERMINALES DEL CONECTOR.....</b>                         | <b>11</b> |
| <b>2.3 FUENTE DE ALIMENTACIÓN.....</b>                          | <b>11</b> |
| <b>2.4 SEÑALIZACIONES.....</b>                                  | <b>11</b> |
| <b>3 INSTALACIÓN DE LA RED CANOPEN.....</b>                     | <b>12</b> |
| <b>3.1 TASA DE COMUNICACIÓN.....</b>                            | <b>12</b> |
| <b>3.2 DIRECCIÓN EN LA RED CANOPEN.....</b>                     | <b>12</b> |
| <b>3.3 RESISTOR DE TERMINACIÓN.....</b>                         | <b>12</b> |
| <b>3.4 CABLE.....</b>   | <b>12</b> |
| <b>3.5 CONEXIÓN CON LA RED.....</b>                             | <b>13</b> |
| <b>4 PARAMETRIZACIÓN.....</b>                                   | <b>14</b> |
| <b>4.1 SÍMBOLOS PARA DESCRIPCIÓN DE LAS PROPIEDADES.....</b>    | <b>14</b> |
| P0105 – SELECCIÓN 1ª/2ª RAMPA.....                              | 14        |
| P0220 – SELECCIÓN FUENTE LOCAL/REMOTO.....                      | 14        |
| P0221 – SELECCIÓN REFERENCIA LOCAL.....                         | 14        |
| P0222 – SELECCIÓN REFERENCIA REMOTA.....                        | 14        |
| P0223 – SELECCIÓN GIRO LOCAL.....                               | 14        |
| P0224 – SELECCIÓN GIRA/PARA LOCAL.....                          | 14        |
| P0225 – SELECCIÓN JOG LOCAL.....                                | 14        |
| P0226 – SELECCIÓN GIRO REMOTO.....                              | 14        |
| P0227 – SELECCIÓN GIRA/PARA REMOTO.....                         | 14        |
| P0228 – SELECCIÓN JOG REMOTO.....                               | 14        |
| P0313 – ACCIÓN PARA ERROR DE COMUNICACIÓN.....                  | 14        |
| P0680 – ESTADO LÓGICO.....                                      | 15        |
| P0681 – VELOCIDAD DEL MOTOR EN 13 BITS.....                     | 16        |
| P0684 – PALABRA DE CONTROL VÍA CANOPEN.....                     | 17        |

|   |           |
|---|-----------|
| P0685 – REFERENCIA DE VELOCIDAD VÍA CANOPEN.....                | 18        |
| P0695 – VALOR PARA LAS SALIDAS DIGITALES.....                   | 18        |
| P0696 – VALOR 1 PARA SALIDAS ANALÓGICAS.....                    | 19        |
| P0697 – VALOR 2 PARA SALIDAS ANALÓGICAS.....                    | 19        |
| P0698 – VALOR 3 PARA SALIDAS ANALÓGICAS.....                    | 19        |
| P0700 – PROTOCOLO CAN.....                                      | 20        |
| P0701 – DIRECCIÓN CAN.....                                      | 20        |
| P0702 – TASA DE COMUNICACIÓN CAN.....                           | 20        |
| P0703 – RESET DE BUS OFF.....                                   | 21        |
| P0705 – ESTADO DEL CONTROLADOR CAN.....                         | 21        |
| P0706 – CONTADOR DE TELEGRAMAS CAN RECIBIDOS.....               | 22        |
| P0707 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS.....            | 22        |
| P0708 – CONTADOR DE ERRORES DE BUS OFF.....                     | 22        |
| P0709 – CONTADOR DE MENSAJES CAN PERDIDAS.....                  | 22        |
| P0721 – ESTADO DE LA COMUNICACIÓN CANOPEN.....                  | 23        |
| P0722 – ESTADO DEL NUDO CANOPEN.....                            | 23        |
| <b>5 DICCIONARIO DE OBJETOS.....</b>                            | <b>24</b> |
| 5.1 ESTRUCTURA DEL DICCIONARIO.....                             | 24        |
| 5.2 TIPOS DE DATOS.....   | 24        |
| 5.3 COMMUNICATION PROFILE – OBJETOS PARA COMUNICACIÓN.....      | 24        |
| 5.4 MANUFACTURER SPECIFIC – OBJETOS ESPECÍFICOS DEL CFW500..... | 25        |
| 5.5 DEVICE PROFILE – OBJETOS COMUNES PARA DRIVES.....           | 25        |
| <b>6 DESCRIPCIÓN DE LOS OBJETOS DE COMUNICACIÓN.....</b>        | <b>27</b> |
| 6.1 OBJETOS DE IDENTIFICACIÓN.....                              | 27        |
| 6.1.1 Objeto 1000h – Device Type.....                           | 27        |
| 6.1.2 Objeto 1001h – Error Register.....                        | 27        |
| 6.1.3 Objeto 1018h – Identity Object.....                       | 27        |
| 6.2 SERVICE DATA OBJECTS – SDOS.....                            | 28        |
| 6.2.1 Objeto 1200h – Servidor SDO.....                          | 29        |
| 6.2.2 Funcionamiento de los SDOs.....                           | 29        |
| 6.3 PROCESS DATA OBJECTS – PDOS.....                            | 30        |
| 6.3.1 Objetos Mapeables para los PDOs.....                      | 31        |
| 6.3.2 PDOs de Recepción.....                                    | 32        |
| 6.3.3 PDOs de Transmisión.....                                  | 34        |
| 6.4 SYNCHRONIZATION OBJECT – SYNC.....                          | 36        |
| 6.5 NETWORK MANAGEMENT – NMT.....                               | 37        |
| 6.5.1 Control de los Estados del Esclavo.....                   | 37        |
| 6.5.2 Control de Errores – <i>Node Guarding</i> .....           | 38        |
| 6.5.3 Control de Errores – <i>Heartbeat</i> .....               | 40        |
| 6.6 PROCEDIMIENTO DE INICIALIZACIÓN.....                        | 42        |
| <b>7 DESCRIPCIÓN DE LOS OBJETOS PARA DRIVES.....</b>            | <b>43</b> |
| 7.1 DEVICE CONTROL – OBJETOS PARA CONTROL DEL DRIVE.....        | 44        |
| 7.1.1 Objeto 6040h – Controlword.....                           | 45        |
| 7.1.2 Objeto 6041h – Statusword.....                            | 46        |
| 7.1.3 Objeto 6060h – Modes of Operation.....                    | 47        |
| 7.1.4 Objeto 6061h – Modes of Operation Display.....            | 47        |
| 7.2 VELOCITY MODE – OBJETOS PARA CONTROL DEL DRIVE.....         | 47        |
| 7.2.1 Bits de Control y Estado.....                             | 48        |
| 7.2.2 Objeto 6042h – vl Target Velocity.....                    | 48        |
| 7.2.3 Objeto 6043h – vl Velocity Demand.....                    | 48        |
| 7.2.4 Objeto 6044h – vl Control Effort.....                     | 48        |
| 7.2.5 Objeto 6046h – vl Velocity Min Max Amount.....            | 49        |
| 7.2.6 Objeto 6048h – vl Velocity Acceleration.....              | 49        |
| 7.2.7 Objeto 6049h – vl Velocity Deceleration.....              | 50        |

|          |   |           |
|----------|---|-----------|
| <b>8</b> | <b>FALLAS Y ALARMAS RELACIONADAS CON LA COMUNICACIÓN CANOPEN.....</b> | <b>51</b> |
|          | A133/F233 – SIN ALIMENTACIÓN EN EL INTERFAZ CAN.....                  | 51        |
|          | A134/F234 – BUS OFF .....   | 51        |
|          | A135/F235 – NODE GUARDING/HEARTBEAT.....                              | 51        |

## A RESPECTO DEL MANUAL

Este manual suministra la descripción necesaria para la operación del convertidor de frecuencia CFW500 utilizando el protocolo CANopen. Este manual debe ser utilizado en conjunto con el manual del usuario del CFW500.

### ABREVIACIONES Y DEFINICIONES

|        |                                 |
|--------|---------------------------------|
| CAN    | Controller Area Network         |
| CiA    | CAN in Automation               |
| COB    | Communication Object            |
| COB-ID | Communication Object Identifier |
| SDO    | Service Data Object             |
| PDO    | Process Data Object             |
| RPDO   | Receive PDO                     |
| TPDO   | Transmit PDO                    |
| NMT    | Network Management Object       |
| ro     | Read only (solamente lectura)   |
| rw     | Read/write (lectura y escrita)  |

### REPRESENTACIÓN NUMÉRICA

Números decimales son representados a través de dígitos sin sufijo. Números hexadecimales son representados con la letra 'h' luego del número.

### DOCUMENTOS

El protocolo CANopen fue desarrollado con base en las siguientes especificaciones y documentos:

| Documento   | Versión | Fuente |
|---|---------|--------|
| CAN Specification   | 2.0     | CiA    |
| CiA DS 301<br>CANopen Application Layer and Communication Profile | 4.02    | CiA    |
| CiA DRP 303-1<br>Cabling and Connector Pin Assignment             | 1.1.1   | CiA    |
| CiA DSP 306<br>Electronic Data Sheet Specification for CANopen    | 1.1     | CiA    |
| CiA DSP 402<br>Device Profile Drives and Motion Control           | 2.0     | CiA    |

Para obtener esta documentación, se debe consultar la CiA, que actualmente es la organización que mantiene, divulga y actualiza las informaciones relativas a la red CANopen.

# 1 INTRODUCCIÓN A LA COMUNICACIÓN CANOPEN

Para la operación en red CANopen, es necesario conocer el modo como la comunicación es hecha. Para es, este ítem trae una descripción general del funcionamiento del protocolo CANopen, conteniendo las funciones utilizadas por el CFW500. Para una descripción detallada, consulte la especificación del protocolo.

## 1.1 CAN

La red CANopen es una red basada en CAN, lo que significa decir que ella utiliza telegramas CAN para intercambios de datos en la red.

El protocolo CAN es un protocolo de comunicación serial que describe los servicios de la camada 2 del modelo ISO/OSI (camada de enlace de datos)<sup>1</sup>. En esta camada, son definidos los distintos tipos de telegramas (frames), el modo de detección de errores, validación y arbitración de los mensajes.

### 1.1.1 Frame de Datos

Los datos en una red CAN son transmitidos a través de un frame de datos. Este tipo de frame es formado principalmente por un campo identificador de 11 bits<sup>2</sup> (arbitration field), y un campo de datos (data field), que puede contener hasta 8 bytes de datos.

|               |                  |        |        |        |        |        |        |        |
|---------------|------------------|--------|--------|--------|--------|--------|--------|--------|
| Identificador | 8 bytes de datos |        |        |        |        |        |        |        |
| 11 bits       | byte 0           | byte 1 | byte 2 | byte 3 | byte 4 | byte 5 | byte 6 | byte 7 |

### 1.1.2 Frame Remoto

Además del frame de datos, existe también el frame remoto (RTR frame). Este tipo de frame no posee campo de datos, solo el identificador. Elle funciona como una requisición para que otro dispositivo de la red transmita el frame de datos deseado.

### 1.1.3 Acceso a la Red

En una red CAN, cualquier elemento de la red puede intentar transmitir un frame para la red en un determinado instante. Caso dos elementos intenten acceder la red al mismo tiempo, conseguirá transmitir aquello que enviar el mensaje más prioritaria. La prioridad del mensaje es definida por el identificador del frame CAN, cuanto menor el valor de este identificador, mayor es la prioridad del mensaje. El telegrama con el identificador "0" (cero) corresponde al telegrama más prioritario.

### 1.1.4 Control de Errores

La especificación CAN define diversos mecanismos para el control de errores, lo que la torna una red muy confiable y con un índice muy bajo de errores de transmisión que no son detectados. Cada dispositivo de la red debe ser capaz de identificar la ocurrencia de estos errores, e informar los demás elementos que un error fue detectado.

Un dispositivo de la red CAN posee contadores internos que son incrementados toda vez que un error de transmisión o recepción es detectado, y decrementado cuando un telegrama es enviado o recibido con suceso. Caso ocurra una cantidad considerable de errores, el dispositivo puede ser llevado para los siguientes estados:

- **Error Active:** los contadores internos de error están en un nivel bajo y el dispositivo opera normalmente en la red CAN. Puede enviar y recibir telegramas y actuar en la red CAN caso detecte algún error en la transmisión de telegramas.
- **Warning:** cuando ese contador pasa de un determinado límite, el dispositivo entra en el estado de *warning*, significando la ocurrencia de una elevada tasa de errores.
- **Error Passive:** cuando este valor ultrapasa un límite mayor, elle entra en el estado de *error passive*, donde es dispositivo para de actuar en la red al detectar que otro dispositivo ha enviado un telegrama con error.

<sup>1</sup> En la especificación del protocolo CAN, es referenciada a la normativa ISO 11898 como definición de la camada 1 de este modelo (camada física).

<sup>2</sup> La especificación CAN 2.0 define dos tipos de frames de datos: *standard* (11bits) y *extended* (29 bits). Para esta implementación, solamente frames *standard* son aceptos.

- **Bus Off:** por último, tenemos el estado de *bus off*, en el cual el dispositivo no irá más enviar o recibir telegramas. El dispositivo opera como si estuviera desconectado de la red.

### 1.1.5 CAN y CANopen

Solamente la definición de como detectar errores, crear y transmitir un frame no son suficientes para definir un significado para los datos que son enviados vía red. Es necesario que haya una especificación que indique como el identificador y los datos deben ser montados y como las informaciones deben ser intercambiadas. De este modo los elementos de la red pueden interpretar correctamente los datos que son transmitidos. En este sentido, la especificación CANopen define justamente como intercambiar datos entre los equipamientos y como cada dispositivo debe interpretar estos datos.

Existen diversos otros protocolos basados en CAN, como DeviceNet, CANopen, J1939, etc., que utilizan frames CAN para la comunicación. Sin embargo estos protocolos no pueden operar en conjunto en la misma red.

## 1.2 CARACTERÍSTICAS DE LA RED CANOPEN

Por utilizar un bus CAN como modo de transmisión de telegramas, todos los dispositivos de la red CANopen poseen los mismos derechos de acceso a la red, donde la prioridad del identificador es responsable por solucionar problemas de conflicto cuando accesos simultáneos ocurren. Eso trae el beneficio de posibilitar la comunicación directa entre esclavos de la red, además del hecho de que los datos pueden ser dispuestos de manera más optimizada, sin la necesidad de un maestro que haga el control de toda la comunicación, haciendo acceso cíclico a todos los dispositivos de la red para la actualización de los datos.

Otra característica importante es la utilización del modelo productor / consumidor para la transmisión de datos. Eso significa decir que un mensaje que transita en la red no posee una dirección fija en la red como destino. Este mensaje posee un identificador que indica cual el dato que ella está transportando. Cualquier elemento de la red que necesite utilizarse de esta información para la suya lógica de operación, podrá consumirla y, por lo tanto, un mismo mensaje puede ser utilizada por varios elementos de la red al mismo tiempo.

### 1.3 MEDIO FÍSICO

El medio físico para la transmisión de las señales en una red CANopen es especificado por la norma ISO 11898. Ella define como bus de transmisión un par de cables tranzado con señal eléctrico diferencial.

### 1.4 DIRECCIÓN EN LA RED CANOPEN

Toda la red CANopen debe poseer un maestro, responsable por servicios de gestión de la red, y también puede poseer un conjunto de hasta 127 esclavos. Cada dispositivo de la red también puede ser llamado de nudo. Todo esclavo en una red CANopen es identificado a través de su dirección, o Node-ID, que debe ser único para cada esclavo de la red, y puede cambiar de 1 hasta 127.

Para el convertidor de frecuencia CFW500, la dirección del esclavo es programada a través del parámetro P0701.

### 1.5 ACCESO A LOS DATOS

Cada esclavo de la red CANopen posee un listado, denominado diccionario de objetos, que contiene todos los datos que son accesibles vía red. Cada objeto de este listado es identificado a través de un índice, y durante la configuración del equipamiento e intercambio de mensajes, este índice es utilizado para identificar lo que esta siendo transmitido.

### 1.6 TRANSMISIÓN DE DATOS

La transmisión de datos numéricos a través de telegramas CANopen es hecha utilizando la representación hexadecimal del número, y enviando el byte menos significativo del dato primer.

Ejemplo: transmisión de un entero con señal de 32 bits (12345678h = 305419896 decimal), más un entero con señal de 16 bits (FF00h = -256 decimal), en un frame CAN.



| identificador | 6 bytes de datos |        |        |        |                |        |
|---------------|------------------|--------|--------|--------|----------------|--------|
|               | Entero 32 bits   |        |        |        | Entero 16 bits |        |
|               | byte 0           | byte 1 | byte 2 | byte 3 | byte 4         | byte 5 |
| 11 bits       | 78h              | 56h    | 34h    | 12h    | 00h            | FFh    |

## 1.7 OBJETOS RESPONSABLES POR LA COMUNICACIÓN – COBS

Existe un determinado conjunto de objetos que son responsables por la comunicación entre los dispositivos de la red. Estos objetos están divididos de acuerdo con los tipos de datos y el modo como son enviados o recibidos por un dispositivo. El CFW500 soporta los siguientes objetos de comunicación (COBs):

*Tabla 1.1: Tipos de objetos de comunicación (COBs)*

| Tipo de Objeto                | Descripción   |
|-------------------------------|---|
| Service Data Object (SDO)     | Los SDOs son objetos responsables por el acceso directo al diccionario de objetos de un dispositivo. A través de mensajes utilizando los SDOs, es posible indicar explícitamente (a través del índice del objeto), cual el dato que está siendo manipulado. Existen dos tipos de SDOs: Cliente SDO, responsable por hacer una requisición de lectura o de escrita para un dispositivo de la red, y el Servidor SDO, responsable por atender esta requisición. Como los SDOs son utilizados generalmente para configuración de un nudo de la red, son menos prioritarios que otros tipos de mensajes.  |
| Process Data Object (PDO)     | Los PDOs son utilizados para acceder datos del equipamiento sin la necesidad de indicar explícitamente cual es el objeto del diccionario que está siendo accedido. Para eso, es necesario configurar previamente cuales son los datos que el PDO estará transmitiendo (mapeo de los datos). También existen dos tipos de PDOs: PDO de recepción y PDO de transmisión. PDOs usualmente son utilizados para transmisión y recepción de datos utilizados durante la operación del dispositivo, y por eso son más prioritarios que los SDOs.  |
| Emergency Object (EMCY)       | Este objeto es responsable por el envío de mensajes para indicar la ocurrencia de errores en el dispositivo. Cuando un error ocurre en un determinado dispositivo (Productor EMCY), este puede enviar un mensaje para la red. Caso algún dispositivo de la red se encuentre monitoreando este mensaje (Consumidor EMCY), es posible programar para que una acción sea tomada (deshabilitar demás dispositivos de la red, reset de errores, etc.).   |
| Synchronization Object (SYNC) | En la red CANopen es posible programar un dispositivo (Productor SYNC) para enviar, periódicamente, un mensaje de sincronización para todos los dispositivos de la red. Estos dispositivos (Consumidores SYNC) pueden entonces, por ejemplo, enviar un determinado dato que necesita estar disponible periódicamente.   |
| Network Management (NMT)      | Toda la red CANopen precisa tener un maestro que haga el control de los demás dispositivos de la red (esclavos). Este maestro será responsable por un conjunto de servicios que controlan la comunicación de los esclavos y su estado en la red CANopen. Los esclavos son responsables por recibir los comandos enviados por el maestro y ejecutar las acciones solicitadas. El protocolo describe dos tipos de servicios que el maestro puede utilizar: servicio de control del dispositivo, donde el maestro controla el estado de cada esclavo en la red, y servicios de control de errores (Node Guarding), donde el esclavo envía mensajes periódicos para el maestro para informar que la conexión está activa. |

Toda la comunicación del convertidor con la red es hecha utilizándose estos objetos, y los datos que pueden ser accedidos son los existentes en el diccionario de objetos del dispositivo.

## 1.8 COB-ID

Un telegrama de la red CANopen siempre es transmitido por un objeto de comunicación (COB). Todo COB posee un identificador que indica el tipo de dato que está siendo transportado. Este identificador, llamado de COB-ID, posee un tamaño de 11 bits, y es transmitido en el campo identificador de un telegrama CAN. Elle puede ser subdividido en dos partes:

| Código da Función |       |       |       | Dirección del nudo |       |       |       |       |       |       |
|-------------------|-------|-------|-------|--------------------|-------|-------|-------|-------|-------|-------|
| bit 10            | bit 9 | bit 8 | bit 7 | bit 6              | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |

- Código de la función: indica el tipo de objeto que está siendo transmitido.
- Dirección del nudo: indica con cual dispositivo de la red el telegrama está vinculado.

A seguir es presentada una tabla con los valores padrones para los diferentes objetos de comunicación disponibles en el CFW500. Es necesario observar que el valor padrón del objeto depende del enderezo del esclavo, con excepción de los COB-IDs para NMT y SYNC, que son comunes para todos los elementos de la red. Estos valores también pueden ser alterados durante la etapa de configuración del dispositivo.

*Tabla 1.2: COB-ID para los diferentes objetos*

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

## 1.9 ARCHIVO EDS

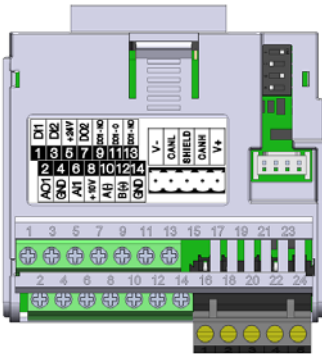
Cada dispositivo en una red CANopen posee un archivo de configuración EDS, que contiene las informaciones sobre el funcionamiento del dispositivo en la red CANopen, bien como la descripción de todos los objetos existentes para comunicación. En general este archivo es utilizado por un maestro o software de configuración, para programación de los dispositivos presentes en la red CANopen.

El archivo de configuración EDS para el CFW500 es suministrado en conjunto con el producto, y también puede ser obtenido a través del sitio web <http://www.weg.net>. Es necesario observar la versión de software del convertidor, para utilizar un archivo EDS que sea compatible con esta versión.

## 2 ACCESORIO PARA COMUNICACIÓN CANOPEN

Para posibilitar la comunicación CANopen en el producto, es necesario utilizar el kit para comunicación CAN descrito a seguir. Informaciones a respecto de la instalación de estos módulos pueden ser obtenidas en el guía rápido que acompaña el kit.

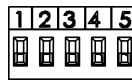
### 2.1 KIT CAN-01



- Ítem WEG: 11593087.
- Formado por el módulo de comunicación CAN (figura al lado) más un guía rápido de montaje.
- Interfaz aislada galvánicamente y con señal diferencial, atribuyendo mayor robustez contra la interferencia electromagnética.
- Alimentación externa de 24V.
- Permite la conexión de hasta 64 dispositivos en el mismo segmento. Una cantidad mayor de dispositivos puede ser conectado con el uso de repetidores<sup>3</sup>.
- Longitud máxima del bus de 1000 metros.

### 2.2 TERMINALES DEL CONECTOR

El módulo para la comunicación CAN posee un conector *plug-in* de 5 vías con los siguientes terminales:



**Tabla 2.1:** Terminales del conector para interfaz CAN

| Terminal | Nombre | Función                                    |
|----------|--------|--|
| 1        | V-     | Polo negativo de la fuente de alimentación |
| 2        | CAN_L  | Señal de comunicación CAN_L                |
| 3        | Shield | Blindaje del cable                         |
| 4        | CAN_H  | Señal de comunicación CAN_H                |
| 5        | V+     | Polo positivo de la fuente de alimentación |



#### ¡NOTA!

Se recomienda conectar a tierra el pin GND del módulo CFW500-CCAN. Esto es necesario para que la pantalla del cable tenga una conexión con la tierra.

### 2.3 FUENTE DE ALIMENTACIÓN

El interfaz CAN necesita de una tensión de alimentación externa entre los terminales 1 y 5 del conector de la red. Los datos para consumo individual y tensión de entrada son presentados en la tabla que sigue.

**Tabla 2.2:** Características de la alimentación para interfase CAN

| Tensión de alimentación ( $V_{cc}$ ) |        |          |
|--------------------------------------|--------|----------|
| Mínimo                               | Máximo | Sugerido |
| 11                                   | 30     | 24       |
| Corriente (mA)                       |        |          |
| Típico                               |        | Máximo   |
| 30                                   |        | 50       |

### 2.4 SEÑALIZACIONES

Indicaciones de alarmas, fallas y estados de la comunicación son realizadas a través de la HMI y de los parámetros del producto.

<sup>3</sup> El número límite de equipos que pueden ser conectados en la red también depende del protocolo utilizado.

### 3 INSTALACIÓN DE LA RED CANOPEN

La red CANopen, como varias redes de comunicación industriales, por el hecho de ser aplicada muchas veces en ambientes agresivos y con alta exposición a la interferencia electromagnética, exige ciertos cuidados que deben ser aplicados para garantizar una baja tasa de errores de comunicación durante su operación. A seguir son presentadas recomendaciones para realizar la conexión del producto en esta red.

#### 3.1 TASA DE COMUNICACIÓN

Equipamientos con interfaz CANopen en general permiten configurar la tasa de comunicación deseada, pudiendo variar de 10Kbit/s hasta 1Mbit/s. La tasa de comunicación (*baud rate*) que puede ser utilizada por un equipamiento depende de la longitud del cable utilizado en la instalación. La tabla que sigue presenta las tasas de comunicación y la longitud máxima de cable que puede ser utilizado en la instalación, de acuerdo con el recomendado por la CiA<sup>4</sup>.

*Tabla 3.1: Tasas de comunicación soportadas y longitud máxima de cable*

| Tasa de comunicación | Longitud del cable |
|----------------------|--------------------|
| 1 Mbit/s             | 25 m               |
| 800 Kbit/s           | 50 m               |
| 500 Kbit/s           | 100 m              |
| 250 Kbit/s           | 250 m              |
| 125 Kbit/s           | 500 m              |
| 100 Kbit/s           | 600 m              |
| 50 Kbit/s            | 1000 m             |
| 20 Kbit/s            | 1000 m             |
| 10 Kbit/s            | 1000 m             |

Todos los equipamientos de la red deben programarse para utilizar la misma tasa de comunicación. Para el convertidor de frecuencia CFW500, la configuración de la tasa de comunicación es hecha a través del parámetro P0702.

#### 3.2 DIRECCIÓN EN LA RED CANOPEN

Cada dispositivo de la red CANopen precisa tener una dirección, o Node ID, entre 1 y 127. Esta dirección debe ser única para cada equipamiento. Para el convertidor de frecuencia CFW500, la configuración de la dirección es hecha a través del parámetro P0701.

#### 3.3 RESISTOR DE TERMINACIÓN

La utilización de resistencias de terminación en las extremidades del bus CAN es fundamental para evitar reflexión de línea, que puede perjudicar la señal transmitida y ocasionar errores en la comunicación. Las extremidades del bus CAN deben poseer un resistor de terminación en el valor de  $121\Omega$  / 0.25W, conectando las señales CAN\_H y CAN\_L.

#### 3.4 CABLE

Para la conexión de las señales CAN\_L y CAN\_H se debe utilizar par tranzado con blindaje. La tabla a seguir presenta las características recomendadas para el cable.

*Tabla 3.2: Propiedades del cable para red CANopen*

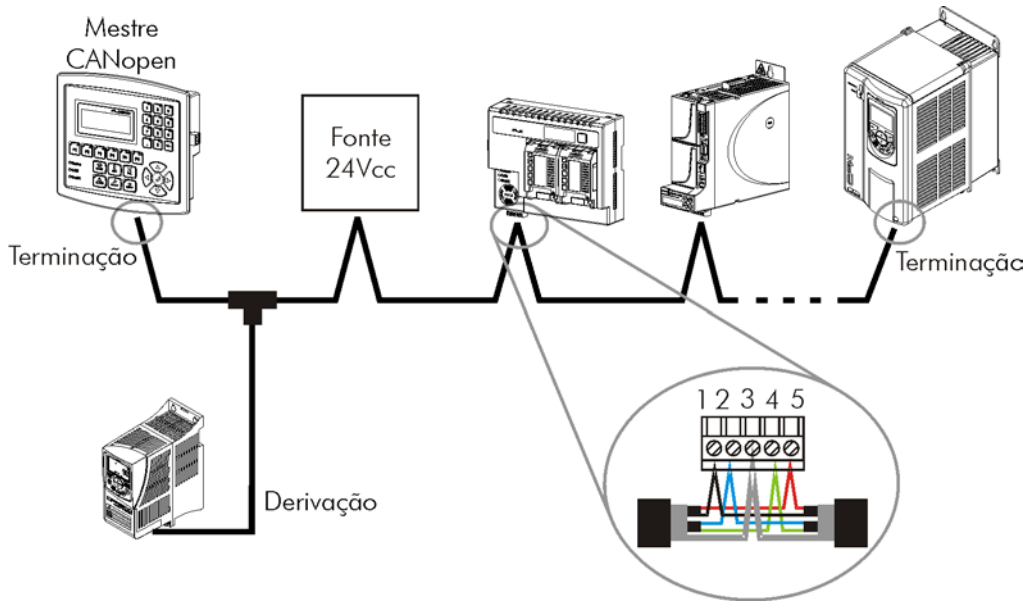
| Longitud del cable (m) | Resistencia por metro (mOhm/m) | Área del conductor (mm <sup>2</sup> ) |
|------------------------|--------------------------------|---------------------------------------|
| 0 ... 40               | 70                             | 0.25 ... 0.34                         |
| 40 ... 300             | <60                            | 0.34 ... 0.60                         |
| 300 ... 600            | <40                            | 0.50 ... 0.60                         |
| 600 ... 1000           | <26                            | 0.75 ... 0.80                         |

<sup>4</sup> Diferentes productos pueden tener diferentes longitudes máximas de cable permitida para la instalación.

También es necesaria la utilización de un par tranzado adicional para llevar la alimentación de 24Vcc para los equipamientos que necesitan de esta señal. Se recomienda usar un cable certificado para red DeviceNet.

### 3.5 CONEXIÓN CON LA RED

Para interconectar los diversos nudos de la red, se recomienda la conexión del equipamiento directamente a partir de la línea principal, sin la utilización de derivaciones. Durante la instalación de los cables, se debe evitar su disposición cerca de los cables de potencia, pues debido a la interferencia electromagnética, eso facilita la ocurrencia de errores durante la transmisión. Para evitar problemas de circulación de corriente por diferencia de potencial entre distintos puntos de puesta a tierra, es necesario que todos los dispositivos estén conectados en el mismo punto de tierra.



**Figura 3.1:** Ejemplo de instalación en red CANopen

Para evitar problemas de diferencia de tensión en la alimentación entre los dispositivos de la red, es recomendado que la red sea alimentada en apenas un punto, y la señal de alimentación sea llevada a todos los dispositivos a través del cable. Caso sea necesaria más de una fuente de alimentación, éstas deben estar referenciadas al mismo punto.

El número máximo de dispositivos conectados en un único segmento de la red es limitado en 64. Repetidores pueden ser utilizados para conectar un número mayor de dispositivos.

## 4 PARAMETRIZACIÓN

A seguir serán presentados solamente los parámetros del convertidor de frecuencia CFW500 que poseen relación directa con la comunicación CANopen.

### 4.1 SÍMBOLOS PARA DESCRIPCIÓN DE LAS PROPIEDADES

|     |   |
|-----|---|
| RO  | Parámetro solamente de lectura  |
| CFG | Parámetro solamente modificado con el motor parado                                |
| CAN | Parámetro visible a través de la IHM si el producto poseer interfaz CAN instalada |

#### P0105 – SELECCIÓN 1ª/2ª RAMPA

#### P0220 – SELECCIÓN FUENTE LOCAL/REMOTO

#### P0221 – SELECCIÓN REFERENCIA LOCAL

#### P0222 – SELECCIÓN REFERENCIA REMOTA

#### P0223 – SELECCIÓN GIRO LOCAL

#### P0224 – SELECCIÓN GIRA/PARA LOCAL

#### P0225 – SELECCIÓN JOG LOCAL

#### P0226 – SELECCIÓN GIRO REMOTO

#### P0227 – SELECCIÓN GIRA/PARA REMOTO

#### P0228 – SELECCIÓN JOG REMOTO

Estos parámetros son utilizados en la configuración de la fuente de los comandos para los modos de operación local y remota del convertidor de frecuencia CFW500. Para que el equipo sea controlado a través de la interfaz CANopen, se debe seleccionar una de las opciones ‘CANopen/DeviceNet/Profibus DP’ disponibles en los parámetros.

La descripción detallada de estos parámetros se encuentra en el manual de programación del convertidor de frecuencia CFW500.

#### P0313 – ACCIÓN PARA ERROR DE COMUNICACIÓN

|                                 |   |                  |
|---------------------------------|---|------------------|
| <b>Rango de</b>                 | 0 = Inactivo  | <b>Padrón: 1</b> |
| <b>Valores:</b>                 | 1 = Para por Rampa                                      |                  |
|                                 | 2 = Deshabilita General                                 |                  |
|                                 | 3 = Va para modo Local                                  |                  |
|                                 | 4 = Va para modo Local y mantiene comandos y referencia |                  |
|                                 | 5 = Causa Falla   |                  |
| <b>Propiedades:</b>             | CFG   |                  |
| <b>Grupo de acceso vía HMI:</b> | NET   |                  |

#### Descripción:

Este parámetro permite seleccionar cual es la acción que debe ser ejecutada por el equipo, caso este sea controlado vía red y un error de comunicación sea detectado.

**Tabla 4.1: Valores de lo parámetro P0313**

| Opciones  | Descripción   |
|---|---|
| 0 = Inactivo  | Ninguna acción es tomada, el equipo permanece en el estado actual.  |
| 1 = Para por Rampa                                      | El comando de parada por rampa es ejecutado, y el motor para de acuerdo con la rampa de desaceleración programada.  |
| 2 = Deshabilita General                                 | El equipo es deshabilitado general, y el motor para por inercia.  |
| 3 = Va para modo Local                                  | El equipo es comandado para el modo local.  |
| 4 = Va para modo Local y mantiene comandos y referencia | El equipo es comandado para el modo local, más los comandos de habilita y de referencia de velocidad recibidos vía red son mantenidos en modo local, desde que el equipo sea programado para utilizar, en modo local, comandos vía HMI o 3 "wire start stop", y la referencia de velocidad vía HMI o potenciómetro electrónico. |
| 5 = Causa Falla   | En el lugar de alarma, un error de comunicación causa una falla en el convertidor de frecuencia; siendo necesario hacer el reset de fallas en el convertidor de frecuencia para que el mismo regrese a su operación normal.   |

Se considera errores de comunicación los siguientes eventos:

Comunicación CANopen/DeviceNet

- Alarma A133/Falla F233: sin alimentación en la interfaz CAN.
- Alarma A134/Falla F234: *bus off*.
- Alarma A135/Falla F235: error de comunicación CANopen (*Node Guarding/Heartbeat*).
- Alarma A136/Falla F236: maestro de la red DeviceNet en modo *Idle*.
- Alarma A137/Falla F237: ocurrió *timeout* en una o más conexiones I/O DeviceNet.

Las acciones descritas en este parámetro son ejecutadas a través de la escrita automática de los respectivos bits en el parámetro de control de la interfaz de red que corresponde a la falla detectada. De esta forma, para que los comandos escritos en este parámetro tengan efecto, es necesario que el equipo se encuentre programado para ser controlado vía la interfaz de red utilizada (a excepción de la opción "Causa Falla", que bloquea el equipo aunque el mismo no sea controlado vía red). Esta programación es hecha a través de los parámetros P0220 hasta P0228.

## P0680 – ESTADO LÓGICO

**Rango de** 0000h a FFFFh

**Padrón:** -

**Valores:**

**Propiedades:** RO

**Grupo de acceso vía HMI:** NET

**Descripción:**

Permite el monitoreo del estado del convertidor de frecuencia. Cada bit representa un estado:

| Bits    | 15       | 14                 | 13         | 12      | 11  | 10              | 9                  | 8             | 7         | 6                        | 5             | 4                    | 3 a 0     |
|---------|----------|--------------------|------------|---------|-----|-----------------|--------------------|---------------|-----------|--------------------------|---------------|----------------------|-----------|
| Función | En Falla | Manual/ Automático | Subtensión | LOC/REM | JOG | Sentido de Giro | Habilitado General | Motor Girando | En Alarma | En modo de configuración | Segunda Rampa | Parada Rápida Activa | Reservado |

**Tabla 4.2:** Función de los bits para el parámetro P0680

| Bits                              | Valores   |
|-----------------------------------|---|
| Bits 0 a 3                        | Reservado.  |
| Bit 4<br>Parada Rápida Activa     | 0: Drive no posee comando de parada rápida activa.<br>1: Drive está ejecutando el comando de parada rápida.   |
| Bit 5<br>Segunda Rampa            | 0: Drive configurado para rampa de aceleración y de desaceleración del motor vía primera rampa; valores programados en los parámetros P0100 y P0101.<br>1: Drive configurado para rampa de aceleración y de desaceleración del motor vía segunda rampa; valores programados en los parámetros P0102 y P0103.  |
| Bit 6<br>En Modo de Configuración | 0: Drive operando normalmente.<br>1: Drive en modo de configuración. Indica una condición especial en la cual el drive no puede ser habilitado:<br>Ejecutando la rutina de autoajuste.<br>Ejecutando la rutina de puesta en marcha (start-up) orientada.<br>Ejecutando la función copy de la HMI.<br>Ejecutando la rutina auto-guiada de la tarjeta de memoria flash.<br>Posee incompatibilidad de parametrización.<br>Sin alimentación en el circuito de potencia del drive. |
| Bit 7<br>En Alarma                | 0: Drive no está en el estado de alarma.<br>1: Drive está en el estado de alarma.<br>Observación: el número de la alarma puede ser leído a través del parámetro P0048 – Alarma Actual.  |
| Bit 8<br>Motor Girando            | 0: Motor está parado.<br>1: Drive está girando el eje del motor a la velocidad de referencia, o ejecutando rampa de aceleración o desaceleración.   |
| Bit 9<br>Habilitado General       | 0: Drive está deshabilitado general.<br>1: Drive está habilitado general y listo para girar el eje del motor.   |
| Bit 10<br>Sentido de Giro         | 0: Motor girando en el sentido reverso.<br>1: Motor girando en el sentido directo.  |
| Bit 11<br>JOG                     | 0: Función JOG inactiva.<br>1: Función JOG activa.  |
| Bit 12<br>LOC/REM                 | 0: Drive en modo local.<br>1: Drive en modo remoto.   |
| Bit 13<br>Subtensión              | 0: Sin subtensión.<br>1: Con subtensión.  |
| Bit 14<br>Manual/ Automático      | 0: En modo manual (función PID).<br>1: En modo automático (función PID).  |
| Bit 15<br>En Falla                | 0: Drive no está en el estado de falla.<br>1: Algún falla registrado por el drive.<br>Observación: El número del falla puede ser leído a través del parámetro P0049 – Falla Actual.   |

### P0681 – VELOCIDAD DEL MOTOR EN 13 BITS

**Rango de** - 32768 a 32767

**Padrón:** -

**Valores:**
**Propiedades:** RO

**Grupo de acceso vía HMI:** NET

**Descripción:**

Permite monitorear la velocidad del motor. Esta palabra utiliza resolución de 13 bits con señal para representar la frecuencia nominal (P0403) del motor:

- P0681 = 0000h (0 decimal) → velocidad del motor = 0
- P0681 = 2000h (8192 decimal) → velocidad del motor = frecuencia nominal

Valores de velocidad intermediarios o superiores pueden ser obtenidos utilizando esta escala. Por ejemplo, 60 Hz de frecuencia nominal, caso el valor leído sea 2048 (0800h), para obtener el valor en Hz se debe calcular:

|   |
|---|
| 8192 => 60 Hz<br>2048 => Frecuencia en Hz |
|---|

|  |
|--|
| Frecuencia en Hz = $\frac{60 \times 2048}{8192}$ |
|--|

|                          |
|--------------------------|
| Frecuencia en Hz = 30 Hz |
|--------------------------|

Valores negativos para este parámetro indican motor girando en el sentido reverso.




**¡NOTA!**

Los valores transmitidos a través de la red presentan una limitación en la escala utilizada, que permite indicar una velocidad máxima de 4 veces la velocidad de sincronismo del motor, con saturación en 32767 (o -32768).

**P0684 – PALABRA DE CONTROL VÍA CANOPEN**
**Rango de** 0000h a FFFFh

**Padrón:** 0000h

**Valores:**
**Propiedades:** NET

**Grupo de acceso vía HMI:** NET

**Descripción:**

Palabra de comando del **convertidor de frecuencia** vía interfaz CANopen. Este parámetro solamente puede ser modificado vía interfaz CANopen/DeviceNet/Profibus DP. Para las demás fuentes (HMI, etc.) ele se comporta como un parámetro solamente de lectura.

Para que los comandos escritos en este parámetro sean ejecutados, es necesario que el equipo se encuentre programado para ser controlado vía CANopen/DeviceNet/Profibus DP. Esta programación es hecha a través de los parámetros P0105 y P0220 hasta P0228.

Cada bit de esta palabra representa un comando que puede ser ejecutado en el equipo.

| Bits    | 15 a 8    | 7               | 6             | 5                     | 4       | 3   | 2               | 1                | 0         |
|---------|-----------|-----------------|---------------|-----------------------|---------|-----|-----------------|------------------|-----------|
| Función | Reservado | Reset de Fallas | Parada Rápida | Utiliza Segunda Rampa | LOC/REM | JOG | Sentido de Giro | Habilita General | Gira/Para |

**Tabla 4.3:** Función de los bits para el parámetro P0684

| Bits                           | Valores  |
|--------------------------------|--|
| Bit 0<br>Gira/Para             | 0: Para el eje del motor por rampa de desaceleración.<br>1: Gira el eje del motor de acuerdo con la rampa de aceleración hasta alcanzar el valor de la referencia de velocidad.  |
| Bit 1<br>Habilita General      | 0: Deshabilita general el drive de frecuencia, interrumpiendo la alimentación para el motor.<br>1: Habilita general el drive, permitiendo la operación del motor.  |
| Bit 2<br>Sentido de Giro       | 0: Girar el eje del motor en el sentido opuesto al de la referencia.<br>1: Girar el eje del motor en el sentido indicado en la referencia.   |
| Bit 3<br>JOG                   | 0: Deshabilita la función JOG.<br>1: Habilita la función JOG.  |
| Bit 4<br>LOC/REM               | 0: Drive va para el modo local.<br>1: Drive va para el modo remoto.  |
| Bit 5<br>Utiliza Segunda Rampa | 0: Drive configurado para rampa de aceleración y de desaceleración del motor vía primera rampa; valores programados en los parámetros P0100 y P0101.<br>1: Drive configurado para rampa de aceleración y de desaceleración del motor vía segunda rampa; valores programados en los parámetros P0102 y P0103. |
| Bit 6<br>Parada Rápida         | 0: No ejecuta el comando de parada rápida.<br>1: Ejecuta el comando de parada rápida.<br>Observación: cuando el tipo de control (P0202) es V/f o VVW no se recomienda la utilización de esta función.  |
| Bit 7<br>Reset de Fallas       | 0: Sin función.<br>1: Si en estado de falla, ejecuta el reset del drive.   |
| Bits 8 a 15                    | Reservado.   |

**P0685 – REFERENCIA DE VELOCIDAD VÍA CANOPEN**

|                                 |                |                  |
|---------------------------------|----------------|------------------|
| <b>Rango de</b>                 | -32768 a 32767 | <b>Padrón:</b> 0 |
| <b>Valores:</b>                 |                |                  |
| <b>Propiedades:</b>             | NET            |                  |
| <b>Grupo de acceso vía HMI:</b> | NET            |                  |

**Descripción:**

Permite programar la referencia de velocidad para el motor vía interfaz CANopen. Este parámetro solamente puede ser modificado vía interfaz CANopen/DeviceNet/Profibus DP. Para las demás fuentes (HMI, etc.) elle se comporta como un parámetro solamente de lectura.

Para que la referencia escrita en este parámetro sea utilizada, es necesario que el equipo se encuentre programado para utilizar la referencia de velocidad vía CANopen/DeviceNet/Profibus DP. Esta programación es hecha a través de los parámetros P0221 y P0222.

Esta palabra utiliza resolución de 13 bits con señal para representar la frecuencia nominal (P0403) del motor:

- P0685 = 0000h (0 decimal) → velocidad del motor = 0
- P0685 = 2000h (8192 decimal) → velocidad del motor = frecuencia nominal (P0403)

Valores de referencias intermedias o superiores pueden ser programados utilizando esta escala. Por ejemplo, 60 Hz de frecuencia nominal, caso se dese una referencia de 30 Hz, se debe calcular:

|   |
|---|
| 60 Hz => 8192<br>30 Hz => Referencia en 13 bits |
|---|

|   |
|---|
| Referencia en 13 bits = $\frac{60 \times 8192}{30}$ |
|---|

|                              |  |
|------------------------------|--|
| Referencia en 13 bits = 4096 | => Valor correspondiente a 30 Hz en la escala de 13 bits |
|------------------------------|--|

Este parámetro también acepta valores negativos para cambiar el sentido de la rotación del motor. El sentido de la rotación de la referencia, sin embargo, depende también del valor del bit 2 de la palabra de control – P0684:

- Bit 2 = 1 y P0685 > 0: referencia para el sentido directo
- Bit 2 = 1 y P0685 < 0: referencia para el sentido reverso
- Bit 2 = 0 y P0685 > 0: referencia para el sentido reverso
- Bit 2 = 0 y P0685 < 0: referencia para el sentido directo


**¡NOTA!**

Los valores transmitidos por la red tienen una limitación debido a la escala utilizada, permitiendo que sea programada una velocidad máxima de referencia de 4 veces la velocidad síncrona del motor.

**P0695 – VALOR PARA LAS SALIDAS DIGITALES**

|                                 |               |                      |
|---------------------------------|---------------|----------------------|
| <b>Rango de</b>                 | 0000h a 001Fh | <b>Padrón:</b> 0000h |
| <b>Valores:</b>                 |               |                      |
| <b>Propiedades:</b>             | NET           |                      |
| <b>Grupo de acceso vía HMI:</b> | NET           |                      |

**Descripción:**

Posibilita el control de las salidas digitales a través de la interfaz de red (Serial, CAN, etc.). Este parámetro no puede ser modificado a través de la HMI.

Cada bit de este parámetro corresponde al valor deseado para una salida digital. Para que la salida digital correspondiente pueda ser controlada de acuerdo con este contenido, es necesaria que su función sea programada para “Contenido P0695”, en los parámetros P0275 a P0279.

| Bits    | 15 a 5    | 4              | 3              | 2              | 1              | 0              |
|---------|-----------|----------------|----------------|----------------|----------------|----------------|
| Función | Reservado | Valor para DO5 | Valor para DO4 | Valor para DO3 | Valor para DO2 | Valor para DO1 |

**Tabla 4.4:** Función de los bits para el parámetro P0695

| Bits                    | Valores  |
|-------------------------|--|
| Bit 0<br>Valor para DO1 | 0: salida DO1 abierta.<br>1: salida DO1 cerrada. |
| Bit 1<br>Valor para DO2 | 0: salida DO2 abierta.<br>1: salida DO2 cerrada. |
| Bit 2<br>Valor para DO3 | 0: salida DO3 abierta.<br>1: salida DO3 cerrada. |
| Bit 3<br>Valor para DO4 | 0: salida DO4 abierta.<br>1: salida DO4 cerrada. |
| Bit 4<br>Valor para DO5 | 0: salida DO5 abierta.<br>1: salida DO5 cerrada. |
| Bits 5 a 15             | Reservado.                                       |



**¡NOTA!**

Algunas de las salidas digitales pueden no estar disponibles dependiendo del módulo plug-in utilizado.

**P0696 – VALOR 1 PARA SALIDAS ANALÓGICAS**

**P0697 – VALOR 2 PARA SALIDAS ANALÓGICAS**

**P0698 – VALOR 3 PARA SALIDAS ANALÓGICAS**

|                                 |                |                  |
|---------------------------------|----------------|------------------|
| <b>Rango de Valores:</b>        | -32768 a 32767 | <b>Padrón:</b> 0 |
| <b>Propiedades:</b>             | NET            |                  |
| <b>Grupo de acceso vía HMI:</b> | NET            |                  |

**Descripción:**

Posibilita el control de las salidas analógicas a través del interfaz de red (Serial, CAN, etc.). Estos parámetros no pueden ser modificados a través de la HMI.

El valor escrito en estos parámetros es utilizado como valor para la salida analógica, desde que la función de la salida analógica deseada sea programada para “Contenido P0696 / P0697”, en los parámetros P0251, P0254, P0257.

El valor debe ser escrito en una escala de 15 bits (7FFFh = 32767)<sup>5</sup> para representar 100 % del valor deseado para la salida, o sea:

- P0696 = 0000h (0 decimal) → valor para la salida analógica = 0 %
- P0696 = 7FFFh (32767 decimal) → valor para la salida analógica = 100 %

En este ejemplo fue presentado el parámetro P0696, más la misma escala es utilizada para los parámetros P0697 / P0698. Por ejemplo, se desea controlar el valor de la salida analógica 1 a través del serial. En este caso se debe proceder la siguiente programación:

- Elegir uno de los parámetros P0696, P0697, P0698 para ser el valor utilizado por la salida analógica 1. En este ejemplo, vamos elegir el P0696.
- Programar, en la función de la salida analógica 1 (P0254), la opción “Contenido P0696”.
- A través del interfaz de red, escribir en el P0696 el valor deseado para la salida analógica 1, entre 0 y 100 %, de acuerdo con la escala del parámetro.

<sup>5</sup> Para la resolución real de la salida, consulte el manual del producto.


**¡NOTA!**

Para el convertidor de frecuencia CFW500, la salida analógica 3 representa la salida en frecuencia (FO).

**P0700 – PROTOCOLO CAN**

|                                 |               |                  |
|---------------------------------|---------------|------------------|
| <b>Rango de</b>                 | 1 = CANopen   | <b>Padrón: 2</b> |
| <b>Valores:</b>                 | 2 = DeviceNet |                  |
| <b>Propiedades:</b>             | CFG           |                  |
| <b>Grupo de acceso vía HMI:</b> | NET           |                  |

**Descripción:**

Permite seleccionar el protocolo deseado para el interfaz CAN. Caso este parámetro fuera alterado, la alteración tendrá efecto solamente si la interfaz CAN estuviera sin alimentación, en *autobaud* o luego que el equipamiento fuera desconectado y conectado nuevamente.

**P0701 – DIRECCIÓN CAN**

|                                 |         |                   |
|---------------------------------|---------|-------------------|
| <b>Rango de</b>                 | 0 a 127 | <b>Padrón: 63</b> |
| <b>Valores:</b>                 |         |                   |
| <b>Propiedades:</b>             | CFG     |                   |
| <b>Grupo de acceso vía HMI:</b> | NET     |                   |

**Descripción:**

Permite programar la dirección utilizada para comunicación CAN del dispositivo. Es necesario que cada equipamiento de la red posea una dirección distinta de las demás. Las direcciones válidas para este parámetro dependen del protocolo programado en el P0700:

- P0700 = 1 (CANopen) → direcciones válidas: 1 a 127.
- P0700 = 2 (DeviceNet) → direcciones válidas: 0 a 63.

Caso este parámetro fuera alterado, la alteración tendrá efecto solamente si la interfaz CAN estuviera sin alimentación, en *autobaud* o luego que el equipamiento fuera desconectado y conectado nuevamente.

**P0702 – TASA DE COMUNICACIÓN CAN**

|                                 |                                  |                  |
|---------------------------------|----------------------------------|------------------|
| <b>Rango de</b>                 | 0 = 1 Mbit/s / <i>Autobaud</i>   | <b>Padrón: 0</b> |
| <b>Valores:</b>                 | 1 = 800 Kbit/s / <i>Autobaud</i> |                  |
|                                 | 2 = 500 Kbit/s                   |                  |
|                                 | 3 = 250 Kbit/s                   |                  |
|                                 | 4 = 125 Kbit/s                   |                  |
|                                 | 5 = 100 Kbit/s / <i>Autobaud</i> |                  |
|                                 | 6 = 50 Kbit/s / <i>Autobaud</i>  |                  |
|                                 | 7 = 20 Kbit/s / <i>Autobaud</i>  |                  |
|                                 | 8 = 10 Kbit/s / <i>Autobaud</i>  |                  |
| <b>Propiedades:</b>             | CFG                              |                  |
| <b>Grupo de acceso vía HMI:</b> | NET                              |                  |

**Descripción:**

Permite programar el valor deseado para la tasa de comunicación del interfaz CAN, en bits por segundo. Esta tasa debe ser la misma para todos los equipamientos conectados en la red. Las tasas de comunicación soportadas para el dispositivo dependen del protocolo programado en el P0700:

- P0700 = 1 (CANopen): se puede utilizar cualquier tasa indicada en este parámetro, mas no posee la función de detección automática de tasa – *autobaud*.
- P0700 = 2 (DeviceNet): solamente las tasas de 500, 250 y 125 Kbit/s son soportadas. Demás opciones habilitan la función de detección automática de tasa – *autobaud*.

Caso este parámetro fuera alterado, la alteración tendrá efecto solamente si la interfaz CAN estuviera sin alimentación o luego que el equipamiento fuera desconectado y conectado nuevamente.

Luego de una detección con suceso, el parámetro de la tasa de comunicación (P0702) modificase automáticamente para la tasa seleccionada. Para ejecutar nuevamente la función de *autobaud*, es necesario modificar el parámetro P0702 para una de las opciones '*Autobaud*'.

**P0703 – RESET DE BUS OFF**

|                                 |                |                  |
|---------------------------------|----------------|------------------|
| <b>Rango de</b>                 | 0 = Manual     | <b>Padrón:</b> 0 |
| <b>Valores:</b>                 | 1 = Automático |                  |
| <b>Propiedades:</b>             | CFG            |                  |
| <b>Grupo de acceso vía HMI:</b> | NET            |                  |

**Descripción:**

Permite programar cual es el comportamiento del convertidor al detectar un error de *bus off* en el interfaz CAN.

*Tabla 4.5: Opciones para el parámetro P0703*

| Opción               | Descripción   |
|----------------------|---|
| 0 = Reset Manual     | Caso ocurra <i>bus off</i> , será señalado en el HMI la alarma A134/F234, la acción programada en el parámetro P0313 será ejecutada y la comunicación será deshabilitada. Para que el convertidor vuelva a se comunicar a través del interfaz CAN, será necesario desenergizar y energizar nuevamente el convertidor. |
| 1 = Reset Automático | Caso ocurra <i>bus off</i> , la comunicación será reiniciada automáticamente y el error será ignorado. En este caso, no será hecha la señalización de alarma en el HMI y el convertidor no ejecutará la acción descrita en el P0313.  |

**P0705 – ESTADO DEL CONTROLADOR CAN**

|                                 |  |                  |
|---------------------------------|--|------------------|
| <b>Rango de</b>                 | 0 = Deshabilitado  | <b>Padrón:</b> - |
| <b>Valores:</b>                 | 1 = <i>Autobaud</i><br>2 = Interfaz CAN activa<br>3 = <i>Warning</i><br>4 = <i>Error Passive</i><br>5 = <i>Bus Off</i><br>6 = Sin alimentación |                  |
| <b>Propiedades:</b>             | RO   |                  |
| <b>Grupo de acceso vía HMI:</b> | NET  |                  |

**Descripción:**

Permite identificar si la tarjeta de interfaz CAN está debidamente instalada, y si la comunicación presenta errores.

*Tabla 4.6: Valores para el parámetro P0705*

| Valor                    | Descripción   |
|--------------------------|---|
| 0 = Deshabilitado        | Interfaz CAN inactiva. Ocurre cuando el convertidor no posee tarjeta de interfaz CAN instalada.               |
| 1 = <i>Autobaud</i>      | Ejecutando función para detección automática de la tasa de comunicación (apenas para el protocolo DeviceNet). |
| 2 = Interfaz CAN activa  | Interfaz CAN activa y sin errores.  |
| 3 = <i>Warning</i>       | Controlador CAN alcanzo el estado de <i>warning</i> .   |
| 4 = <i>Error Passive</i> | Controlador CAN alcanzo el estado de <i>error passive</i> .   |
| 5 = <i>Bus Off</i>       | Controlador CAN alcanzo el estado de <i>bus off</i> .   |
| 6 = Sin alimentación     | Interfaz CAN no posee alimentación entre los terminales 1 y 5 del conector.                                   |

**P0706 – CONTADOR DE TELEGRAMAS CAN RECIBIDOS**

|                                 |           |                  |
|---------------------------------|-----------|------------------|
| <b>Rango de Valores:</b>        | 0 a 65535 | <b>Padrón:</b> - |
| <b>Propiedades:</b>             | RO        |                  |
| <b>Grupo de acceso vía HMI:</b> | NET       |                  |

**Descripción:**

Este parámetro funciona como un contador cíclico, que es incrementado toda vez que un telegrama CAN es recibido. Suministra un retorno para el operador si el dispositivo está consiguiendo comunicarse con la red. Este contador es puesto a cero siempre que el equipo es desenergizado, hecho el reset o al alcanzar el límite máximo del parámetro.

**P0707 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS**

|                                 |           |                  |
|---------------------------------|-----------|------------------|
| <b>Rango de Valores:</b>        | 0 a 65535 | <b>Padrón:</b> - |
| <b>Propiedades:</b>             | RO        |                  |
| <b>Grupo de acceso vía HMI:</b> | NET       |                  |

**Descripción:**

Este parámetro funciona como un contador cíclico, que es incrementado toda vez que un telegrama CAN es transmitido. Suministra un retorno para el operador si el dispositivo está consiguiendo comunicarse con la red. Este contador es puesto a cero siempre que el equipo fuera desenergizado, hecho el reset o al alcanzar el límite máximo del parámetro.

**P0708 – CONTADOR DE ERRORES DE BUS OFF**

|                                 |           |                  |
|---------------------------------|-----------|------------------|
| <b>Rango de Valores:</b>        | 0 a 65535 | <b>Padrón:</b> - |
| <b>Propiedades:</b>             | RO        |                  |
| <b>Grupo de acceso vía HMI:</b> | NET       |                  |

**Descripción:**

Contador cíclico que indica el número de veces que el equipo ha entrado en el estado de *bus off* en la red CAN. Este contador es puesto a cero siempre que el equipo es desenergizado, hecho el reset o al alcanzar el límite máximo del parámetro.

**P0709 – CONTADOR DE MENSAJES CAN PERDIDAS**

|                                 |           |                  |
|---------------------------------|-----------|------------------|
| <b>Rango de Valores:</b>        | 0 a 65535 | <b>Padrón:</b> - |
| <b>Propiedades:</b>             | RO        |                  |
| <b>Grupo de acceso vía HMI:</b> | NET       |                  |

**Descripción:**

Contador cíclico que indica el número de mensajes recibidas por la interfaz CAN, más que no podrán ser procesadas por el equipo. Caso el número de mensajes perdidos sea incrementado con frecuencia, recomendase disminuir la tasa de comunicación utilizada para la red CAN. Este contador es puesto a cero siempre que el equipo es desenergizado, hecho el reset o al alcanzar el límite máximo del parámetro.

**P0721 – ESTADO DE LA COMUNICACIÓN CANOPEN**

|                                 |                                   |                  |
|---------------------------------|-----------------------------------|------------------|
| <b>Rango de</b>                 | 0 = Deshabilitado                 | <b>Padrón:</b> - |
| <b>Valores:</b>                 | 1 = Reservado                     |                  |
|                                 | 2 = Comunicación Habilitada       |                  |
|                                 | 3 = Control de Errores Habilitado |                  |
|                                 | 4 = Erro de Guarding              |                  |
|                                 | 5 = Erro de Heartbeat             |                  |
| <b>Grupo de acceso vía HMI:</b> | NET                               |                  |

**Descripción:**

Indica el estado de la tarjeta con relación a la red CANopen, informando si el protocolo fue habilitado y si el servicio de control de errores está activo (*Node Guarding* o *Heartbeat*).

**P0722 – ESTADO DEL NUDO CANOPEN**

|                                 |                    |                  |
|---------------------------------|--------------------|------------------|
| <b>Rango de</b>                 | 0 = Deshabilitado  | <b>Padrón:</b> - |
| <b>Valores:</b>                 | 1 = Inicialización |                  |
|                                 | 2 = Parado         |                  |
|                                 | 3 = Operacional    |                  |
|                                 | 4 = Preoperacional |                  |
| <b>Grupo de acceso vía HMI:</b> | NET                |                  |

**Descripción:**

El equipo opera como esclavo de la red CANopen, y como tal posee una máquina de estados que hace el control de su comportamiento con relación a la comunicación. Este parámetro indica en cual estado se encuentra el dispositivo.

## 5 DICCIONARIO DE OBJETOS

El diccionario de objetos es un listado con los diversos datos del equipamiento que son accedidos a través de la red CANopen. Un objeto de este listado es identificado a través de un índice de 16 bits, y es basado en este listado que todo el intercambio de datos entre los dispositivos es efectuado.

El documento CiA DS 301 define un conjunto mínimo de objetos que todo el esclavo de la red CANopen debe poseer. Los objetos disponibles en este listado son agrupados de acuerdo con el tipo de función que elle ejecuta. Los objetos son dispuestos en el diccionario de la siguiente manera:

*Tabla 5.1: Agrupamientos del diccionario de objetos*

| Índice        | Objetos                                  | Descripción  |
|---------------|--|--|
| 0001h – 025Fh | Definición de los tipos de datos         | Utilizado como referencia para los tipos de datos soportados por el sistema.   |
| 1000h – 1FFFh | Objetos de comunicación                  | Son objetos comunes a todos los dispositivos CANopen. Contiene informaciones generales a respecto del equipo y también datos para la configuración de la comunicación. |
| 2000h – 5FFFh | Objetos específicos del fabricante       | En este rango, cada fabricante de equipos CANopen es libre para definir cuales datos estos objetos irán representar.   |
| 6000h – 9FFFh | Objetos estandarizados para dispositivos | Este rango es reservado para objetos que describen el comportamiento de equipos similares, independiente del fabricante.   |

Demás índices no referenciados en este listado son reservados para uso futuro.

### 5.1 ESTRUCTURA DEL DICCIONARIO

La estructura general del diccionario de objetos posee el siguiente formato:

| Índice | Objeto | Nombre | Tipo | Acceso |
|--------|--------|--------|------|--------|
|--------|--------|--------|------|--------|

- **Índice:** indica directamente el índice del objeto en el diccionario.
- **Objeto:** describe que información el índice almacena (variables simples, *array*, *record*, etc.)
- **Nombre:** contiene el nombre del objeto para facilitar su identificación.
- **Tipo:** indica directamente el tipo de dato almacenado. Para variables simples, este tipo puede ser un entero, un *float*, etc. Para *arrays*, indica el tipo del dato contenido en el *array*. Para *records*, indica el formato del *record*, de acuerdo con los tipos descriptos en la primera parte del diccionario de objetos (índices 0001h – 0360h).
- **Acceso:** informa si el objeto en cuestión está accesible solamente para lectura (ro), para lectura y escrita (rw), o es una constante (const).

Para objetos del tipo array o records, todavía es necesario un subíndice, que no es descrito en la estructura del diccionario.

### 5.2 TIPOS DE DATOS

La primera parte del diccionario de objetos (índices 0001h – 025Fh) describe los tipos de datos que pueden ser accedidos en un dispositivo en la red CANopen. Estos pueden ser tipos básicos, como enteros y floats, o tipos compuestos, formados por un conjunto de entradas, como records y arrays.

### 5.3 COMMUNICATION PROFILE – OBJETOS PARA COMUNICACIÓN

Los índices de 1000h hasta 1FFFh corresponden, en el diccionario de objetos, la parte responsable por las configuraciones de la comunicación en la red CANopen. Estos objetos son comunes a todos los dispositivos, sin embargo solamente algunos son obligatorios. A seguir es presentado uno listado con los objetos de este rango soportados por el convertidor de frecuencia CFW500.



**Tabla 5.2:** Listado de objetos – Communication Profile

| Índice                               | Objeto | Nombre                     | Tipo          | Acceso |
|--------------------------------------|--------|----------------------------|---------------|--------|
| 1000h                                | VAR    | device type                | UNSIGNED32    | ro     |
| 1001h                                | VAR    | error register             | UNSIGNED8     | ro     |
| 1005h                                | VAR    | COB-ID SYNC                | UNSIGNED32    | rw     |
| 100Ch                                | VAR    | guard time                 | UNSIGNED16    | rw     |
| 100Dh                                | VAR    | life time factor           | UNSIGNED8     | rw     |
| 1016h                                | ARRAY  | Consumer heartbeat time    | UNSIGNED32    | rw     |
| 1017h                                | VAR    | Producer heartbeat time    | UNSIGNED16    | rw     |
| 1018h                                | RECORD | Identity Object            | Identity      | ro     |
| Server SDO Parameter                 |        |                            |               |        |
| 1200h                                | RECORD | 1st Server SDO parameter   | SDO Parameter | ro     |
| Receive PDO Communication Parameter  |        |                            |               |        |
| 1400h                                | RECORD | 1st receive PDO Parameter  | PDO CommPar   | rw     |
| 1401h                                | RECORD | 2nd receive PDO Parameter  | PDO CommPar   | rw     |
| Receive PDO Mapping Parameter        |        |                            |               |        |
| 1600h                                | RECORD | 1st receive PDO mapping    | PDO Mapping   | rw     |
| 1601h                                | RECORD | 2nd receive PDO mapping    | PDO Mapping   | rw     |
| Transmit PDO Communication Parameter |        |                            |               |        |
| 1800h                                | RECORD | 1st transmit PDO Parameter | PDO CommPar   | rw     |
| 1801h                                | RECORD | 2nd transmit PDO Parameter | PDO CommPar   | rw     |
| Transmit PDO Mapping Parameter       |        |                            |               |        |
| 1A00h                                | RECORD | 1st transmit PDO mapping   | PDO Mapping   | rw     |
| 1A01h                                | RECORD | 2nd transmit PDO mapping   | PDO Mapping   | rw     |

Estos objetos solamente pueden leerse y escribirse a través de la red CANopen, no están disponibles vía HMI u otra interfaz de red. El maestro de la red, en general, es el equipamiento responsable por la configuración del equipamiento antes de iniciar la operación. El archivo de configuración EDS trae la lista de todos los objetos de comunicación soportados.

Para una descripción detallada de cuales objetos están disponibles en este rango del diccionario de objetos, consulte el ítem 6.

#### 5.4 MANUFACTURER SPECIFIC – OBJETOS ESPECÍFICOS DEL CFW500

En los índices de 2000h hasta 5FFFh, cada fabricante es libre para definir cuales objetos estarán presentes, el tipo y la función de cada objeto. El convertidor de frecuencia CFW500, en este rango de objetos fue proporcionado todo el listado de parámetros. A través de estos parámetros es posible operar el CFW500, ejecutando cualquier función que el convertidor pueda realizar. Los parámetros fueran dispuestos a partir del índice 2000h, y con el número del parámetro sumado a este índice para obtener su posición en el diccionario. La tabla que sigue presenta como están distribuidos los parámetros en el diccionario de objetos.

**Tabla 5.3:** Listado de objetos del CFW500 – Manufacturer Specific

| Índice | Objeto | Nombre                           | Tipo      | Acceso |
|--------|--------|----------------------------------|-----------|--------|
| 2000h  | VAR    | P0000 – Parámetro de acceso      | INTEGER16 | rw     |
| 2001h  | VAR    | P0001 – Referencia de Velocidad  | INTEGER16 | ro     |
| 2002h  | VAR    | P0002 – Velocidad del motor      | INTEGER16 | ro     |
| 2003h  | VAR    | P0003 – Corriente del motor      | INTEGER16 | ro     |
| 2004h  | VAR    | P0004 – Tensión CC               | INTEGER16 | ro     |
| ...    | ...    | ...                              | ...       | ...    |
| 2064h  | VAR    | P0100 – Tiempo de aceleración    | INTEGER16 | rw     |
| 2065h  | VAR    | P0101 – Tiempo de desaceleración | INTEGER16 | rw     |
| ...    | ...    | ...                              | ...       | ...    |

Para el listado completo y una descripción detallada de los parámetros, consulte el manual del CFW500. Es necesario reconocer la operación del convertidor a través de los parámetros para que se pueda programar correctamente su operación vía red CANopen.

#### 5.5 DEVICE PROFILE – OBJETOS COMUNES PARA DRIVES

La documentación CANopen también incluí propuestas para la estandarización de determinados tipos de dispositivos. El convertidor de frecuencia CFW500 sigue el descrito por la CiA DPS 402 – Device Profile Drives and Motion Control. Este documento describe un conjunto de objetos que deben ser comunes para drives, independiente del fabricante. Eso facilita la interoperabilidad entre dispositivos con la misma función (como

convertidores de frecuencia), pues tanto los datos cuanto el comportamiento del dispositivo están dispuestos de una forma padrón.

Para estos objetos fueran reservados los índices de 6000h hasta 9FFFh. Es posible operar el convertidor a través de la red CANopen, tanto a través de los parámetros (ubicados a partir del índice 2000h) cuanto a través de estos objetos estandarizados.

Para una descripción detallada de cuales objetos están disponibles en este rango del diccionario de objetos, consulte el ítem 6.

## 6 DESCRIPCIÓN DE LOS OBJETOS DE COMUNICACIÓN

En este ítem son descriptos detalladamente cada uno de los objetos de comunicación disponibles para el convertidor de frecuencia CFW500. Es necesario conocer como estos objetos son operados para utilizar las funciones disponibles para la comunicación del drive.

### 6.1 OBJETOS DE IDENTIFICACIÓN

Existe un conjunto de objetos en el diccionario que son utilizados para la identificación del equipamiento, sin embargo no poseen influencia en su comportamiento en la red CANopen.

#### 6.1.1 Objeto 1000h – Device Type

Este objeto suministra un código en 32 bits que describe el tipo de objeto y su funcionalidad.

|              |             |
|--------------|-------------|
| Índice       | 1000h       |
| Nombre       | Device type |
| Objeto       | VAR         |
| Tipo         | UNSIGNED32  |
| Acceso       | ro          |
| Mapeable     | No          |
| Rango        | UNSIGNED32  |
| Valor Padrón | 0001.0192h  |

Este código puede ser dividido en dos partes: 16 bits inferiores, describiendo el tipo de perfil (*profile*) que el dispositivo utiliza, y 16 bits superiores, indicando una función específica, de acuerdo con el perfil especificado.

#### 6.1.2 Objeto 1001h – Error Register

Este objeto indica la ocurrencia o no de error en el dispositivo. El tipo de error registrado para el CFW500 es descrito conforme tabla 6.1.

|              |                |
|--------------|----------------|
| Índice       | 1001h          |
| Nombre       | Error register |
| Objeto       | VAR            |
| Tipo         | UNSIGNED8      |
| Acceso       | ro             |
| Mapeable     | Sí             |
| Rango        | UNSIGNED8      |
| Valor Padrón | 0              |

**Tabla 6.1:** Estructura del objeto Error Register

| Bit | Significado               |
|-----|---------------------------|
| 0   | Error genérico            |
| 1   | Corriente                 |
| 2   | Tensión                   |
| 3   | Temperatura               |
| 4   | Comunicación              |
| 5   | Reservado (siempre 0)     |
| 6   | Reservado (siempre 0)     |
| 7   | Específico del fabricante |

Caso el dispositivo presente algún error, el bit equivalente debe ser activado. El primer bit (error genérico) deberá ser activado en cualquier situación de error.

#### 6.1.3 Objeto 1018h – Identity Object

Trae informaciones generales a respecto del dispositivo.

|        |                 |
|--------|-----------------|
| Índice | 1018h           |
| Nombre | Identity object |
| Objeto | Record          |
| Tipo   | Identity        |

|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número del último subíndice |
| Acceso       | RO                          |
| Mapeable     | No                          |
| Rango        | UNSIGNED8                   |
| Valor Padrón | 4                           |

|              |            |
|--------------|------------|
| Subíndice    | 1          |
| Descripción  | Vendor ID  |
| Acceso       | RO         |
| Mapeable     | No         |
| Rango        | UNSIGNED32 |
| Valor Padrón | 0000.0123h |

|              |                     |
|--------------|---------------------|
| Subíndice    | 2                   |
| Descripción  | Código del producto |
| Acceso       | RO                  |
| Mapeable     | No                  |
| Rango        | UNSIGNED32          |
| Valor Padrón | 0000.0A00h          |

|              |  |
|--------------|--|
| Subíndice    | 3  |
| Descripción  | Número de la revisión                                  |
| Acceso       | RO   |
| Mapeable     | No   |
| Rango        | UNSIGNED32   |
| Valor Padrón | De acuerdo con la versión de firmware del equipamiento |

|              |                            |
|--------------|----------------------------|
| Subíndice    | 4                          |
| Descripción  | Número serial              |
| Acceso       | RO                         |
| Mapeable     | No                         |
| Rango        | UNSIGNED32                 |
| Valor Padrón | Diferente para cada CFW500 |

El Vendor ID es un número que identifica el fabricante junto a la CiA. El código del producto es definido por el fabricante de acuerdo con el tipo de producto. El número de la revisión representa la versión de firmware del equipamiento. El subíndice 4 es un número serial único para cada convertidor de frecuencia CFW500 en red CANopen.

## 6.2 SERVICE DATA OBJECTS – SDOs

Los SDOs son responsables por el acceso directo al diccionario de objetos de un determinado dispositivo en la red. Ellos son utilizados para la configuración y, por lo tanto, poseen baja prioridad, ya que no deben ser utilizados para comunicar datos necesarios para la operación del dispositivo.

Existen dos tipos de SDOs: cliente y servidor. Básicamente, la comunicación inicia con el cliente (usualmente el maestro de la red) haciendo una requisición de lectura (*upload*) o escrita (*download*) para un servidor, y este contesta al que fue solicitado.

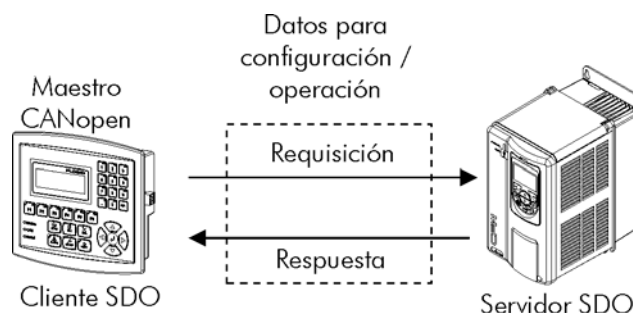


Figura 6.1: Comunicación entre cliente y servidor SDO

### 6.2.1 Objeto 1200h – Servidor SDO

El convertidor de frecuencia CFW500 posee un único SDO del tipo servidor, que posibilita el acceso a todo el su diccionario de objetos. A través de el, un cliente SDO puede configurar la comunicación, parámetros y modos de operación del convertidor. Todo el servidor SDO posee un objeto, del tipo SDO\_PARAMETER, para la su configuración, poseyendo la siguiente estructura:

|        |                      |
|--------|----------------------|
| Índice | 1200h                |
| Nombre | Server SDO Parameter |
| Objeto | Record               |
| Tipo   | SDO Parameter        |

|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número del último subíndice |
| Acceso       | RO                          |
| Mapeable     | No                          |
| Rango        | UNSIGNED8                   |
| Valor Padrón | 2                           |

|              |                                |
|--------------|--------------------------------|
| Subíndice    | 1                              |
| Descripción  | COB-ID Cliente - Servidor (rx) |
| Acceso       | RO                             |
| Mapeable     | No                             |
| Rango        | UNSIGNED32                     |
| Valor Padrón | 600h + Node-ID                 |

|              |                                |
|--------------|--------------------------------|
| Subíndice    | 2                              |
| Descripción  | COB-ID Servidor - Cliente (tx) |
| Acceso       | RO                             |
| Mapeable     | No                             |
| Rango        | UNSIGNED32                     |
| Valor Padrón | 580h + Node-ID                 |

### 6.2.2 Funcionamiento de los SDOs

Un telegrama enviado por un SDO posee 8 bytes de tamaño, con la siguiente estructura:

| Identificador | 8 bytes de datos |        |        |           |                  |        |        |        |
|---------------|------------------|--------|--------|-----------|------------------|--------|--------|--------|
| 11 bits       | Comando          | Índice |        | Subíndice | Datos del objeto |        |        |        |
|               | byte 0           | byte 1 | byte 2 | byte 3    | byte 4           | byte 5 | byte 6 | byte 7 |

El identificador depende del sentido de la transmisión (rx o tx) y de la dirección (o Node-ID) del servidor destino. Por ejemplo, un cliente que hace una requisición para un servidor cuyo Node-ID es 1, debe enviar un mensaje con el identificador igual a 601h. El servidor irá recibir esto mensaje y contestar con un telegrama cuyo COB-ID es igual a 581h.

El código del comando depende del tipo de función utilizada. Para las transmisiones de un cliente para un servidor, pueden ser utilizados los siguientes comandos:

*Tabla 6.2: Código de los comandos para cliente SDO*

| Comando   | Función        | Descripción        | Datos del objeto |
|-----------|----------------|--------------------|------------------|
| 22h       | Download       | Escrita en objeto  | Indefinido       |
| 23h       | Download       | Escrita en objeto  | 4 bytes          |
| 2Bh       | Download       | Escrita en objeto  | 2 bytes          |
| 2Fh       | Download       | Escrita en objeto  | 1 byte           |
| 40h       | Upload         | Lectura de objeto  | No utilizado     |
| 60h o 70h | Upload segment | Lectura segmentada | No utilizado     |

Al hacer la requisición, el cliente indicará a través de su COB-ID, cual es la dirección del esclavo para el cual esta requisición se destina. Solamente un esclavo (usando su respectivo servidor SDO) podrá contestar para el cliente el telegrama recibido. El telegrama de respuesta poseerá también la misma estructura del telegrama de requisición, sin embargo los comandos serán diferentes:

**Tabla 6.3:** Código de los comandos para servidor SDO

| Comando     | Función        | Descripción                              | Datos del objeto |
|-------------|----------------|--|------------------|
| 60h         | Download       | Respuesta para escrita en objeto         | No utilizado     |
| 43h         | Upload         | Respuesta para lectura de objeto         | 4 bytes          |
| 4Bh         | Upload         | Respuesta para lectura de objeto         | 2 bytes          |
| 4Fh         | Upload         | Respuesta para lectura de objeto         | 1 byte           |
| 41h         | Upload segment | Inicia respuesta segmentada para lectura | 4 bytes          |
| 01h ... 0Dh | Upload segment | Último segmento de datos para lectura    | 8 ... 2 bytes    |

Para lecturas que involucran hasta cuatro bytes de datos, un único mensaje puede ser transmitido por el servidor; para lectura de una cantidad mayor de bytes, es necesario que cliente y servidor intercambien múltiples telegramas.

Un telegrama solamente es completo luego de la confirmación del servidor para la requisición hecha por el cliente. Caso algún error sea detectado durante el intercambio de telegramas (por ejemplo, no hay respuesta del servidor), el cliente podrá abortar el proceso con un mensaje de aviso con el código del comando igual a 80h.



**¡NOTA!**

Cuando el SDO es utilizado para escrita en los objetos que representan los parámetros del CFW500 (objetos a partir del índice 2000h), este valor es guardado en la memoria no volátil del convertidor de frecuencia. De esta forma, después de apagado o hecho el reset del equipamiento, los valores configurados no son perdidos. Para los demás objetos, estos valores no son guardados automáticamente, de manera que es necesario rescribir los valores deseados.

Ejemplo: un cliente SDO solicita para un CFW500 en la dirección 1, la lectura del objeto identificado por el índice 2000h, subíndice 0 (cero), que representa un entero de 16 bits. El telegrama del maestro posee la siguiente forma:

| Identificador | Comando | Índice |     | Subíndice | Datos |     |     |     |
|---------------|---------|--------|-----|-----------|-------|-----|-----|-----|
| 601h          | 40h     | 00h    | 20h | 00h       | 00h   | 00h | 00h | 00h |

El CFW500 contesta a la requisición, indicando que el valor para el referido objeto es igual a 999<sup>6</sup>:

| Identificador | Comando | Índice |     | Subíndice | Datos |     |     |     |
|---------------|---------|--------|-----|-----------|-------|-----|-----|-----|
| 581h          | 4Bh     | 00h    | 20h | 00h       | E7    | 03h | 00h | 00h |

### 6.3 PROCESS DATA OBJECTS – PDOS

Los PDOS son utilizados para enviar y recibir datos utilizados durante la operación del dispositivo, que muchas veces precisan ser transmitidos de forma rápida y eficiente. Por eso, ellos poseen una prioridad mayor del que los SDOs.

En los PDOS, solamente los datos son transmitidos en el telegrama (índices y subíndices son omitidos), y de esta forma es posible hacer una transmisión más eficiente, con mayor volumen de datos en un único telegrama. Sin embargo es necesario configurar previamente el que está siendo transmitido por el PDO, de forma que, mismo sin la indicación del índice y subíndice, sea posible saber el contenido del telegrama.

Existen dos tipos de PDOS, los PDOS de recepción y los PDOS de transmisión. Los PDOS de transmisión son responsables por enviar datos para la red, mientras que los PDOS de recepción se quedan responsables por recibir y tratar estos datos. De esta forma es posible que haya comunicación entre esclavos de la red CANopen, desde que sea configurado un esclavo para transmitir una información, y un o más esclavos para recibir esta información.

<sup>6</sup> No olvidar que cualquier dato del tipo entero, el orden de transferencia de los bytes va del menos significativo hasta el más significativo.

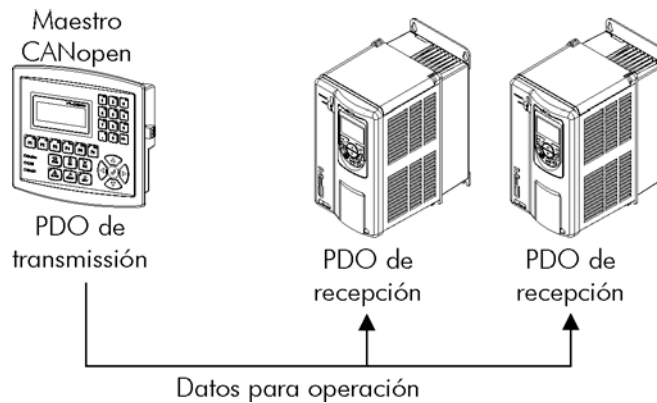


Figura 6.2: Comunicación utilizando PDOs



**¡NOTA!**

PDOs solamente pueden ser transmitidos o recibidos cuando el dispositivo está en el estado operacional. La figura 6.4 presenta los estados disponibles para un nudo de la red CANopen.

### 6.3.1 Objetos Mapeables para los PDOs

Para un objeto poder ser transmitido a través de un PDO, es necesario que él sea mapeable para el contenido del PDO. En la descripción de los objetos de comunicación (1000h – 1FFFh), el campo “Mapeable” informa esta condición. Usualmente, solo informaciones necesarias para la operación del dispositivo son mapeables, como comandos para habilitación, status del dispositivo, referencias, etc. Informaciones para configuración del dispositivo no son accedidas a través de PDOs, e caso sea necesario accederlas vía red se debe utilizar los SDOs.

Para los objetos específicos del fabricante (2000h – 5FFFh), la tabla que sigue presenta algunos objetos mapeables para los PDOs. Parámetros con acceso solo para lectura (ro) pueden ser utilizados solo por PDOs de transmisión, mientras que los demás parámetros pueden ser utilizados solo por PDOs de recepción. El archivo EDS del equipo trae un listado de todos los objetos disponibles para el convertidor, informando si el objeto es mapeable o no.

Tabla 6.4: Ejemplos de parámetros mapeables para PDOs

| Índice | Objeto | Nombre                         | Tipo       | Acceso |
|--------|--------|--------------------------------|------------|--------|
| 2002h  | VAR    | P0002 – Velocidad del Motor    | UNSIGNED16 | ro     |
| 2003h  | VAR    | P0003 – Corriente del Motor    | UNSIGNED16 | ro     |
| 2004h  | VAR    | P0004 – Tensión Bus CC (Ud)    | UNSIGNED16 | ro     |
| 2005h  | VAR    | P0005 – Frecuencia del Motor   | UNSIGNED16 | ro     |
| 2006h  | VAR    | P0006 – Estado del Convertidor | UNSIGNED16 | ro     |
| 2007h  | VAR    | P0007 – Tensión de Salida      | UNSIGNED16 | ro     |
| 2009h  | VAR    | P0009 – Par en el Motor        | INTEGER16  | ro     |
| 200Ah  | VAR    | P0010 – Potencia de Salida     | UNSIGNED16 | ro     |
| 200Ch  | VAR    | P0012 – Estado DI1 a DI8       | UNSIGNED16 | ro     |
| 200Dh  | VAR    | P0013 – Estado DO1 a RL3       | UNSIGNED16 | ro     |
| 2012h  | VAR    | P0018 – Valor de AI1           | INTEGER16  | ro     |
| 2013h  | VAR    | P0019 – Valor de AI2           | INTEGER16  | ro     |
| 2014h  | VAR    | P0020 – Valor de AI3           | INTEGER16  | ro     |
| 2015h  | VAR    | P0021 – Valor de AI4           | INTEGER16  | ro     |
| 2064h  | VAR    | P0100 – Tiempo Aceleración     | UNSIGNED16 | rw     |
| 2065h  | VAR    | P0101 – Tiempo Desaceleración  | UNSIGNED16 | rw     |
| 22A8h  | VAR    | P0680 – Estado Lógico          | UNSIGNED16 | ro     |
| 22A9h  | VAR    | P0681 – Velocidad 13 bits      | UNSIGNED16 | ro     |
| 22ACh  | VAR    | P0684 – Control CANopen/DNet   | UNSIGNED16 | rw     |
| 22ADh  | VAR    | P0685 – Ref. Vel. CANop./DNet  | UNSIGNED16 | rw     |

El archivo EDS del equipamiento trae la lista de todos los objetos disponibles, informando si el objeto es mapeable o no.

## 6.3.2 PDOs de Recepción

Los PDOs de recepción, o RPDOs, son responsables por recibir datos que otros dispositivos envían para la red CANopen. El convertidor de frecuencia CFW500 posee 2 PDOs de recepción, cada uno pudiendo recibir hasta 8 bytes de datos. Cada RPDO posee dos parámetros para su configuración, un PDO\_COMM\_PARAMETER y un PDO\_MAPPING, conforme presentado a seguir.

### PDO\_COMM\_PARAMETER

|        |                                     |
|--------|-------------------------------------|
| Indice | 1400h hasta 1401h                   |
| Nombre | Receive PDO communication parameter |
| Objeto | Record                              |
| Tipo   | PDO COMM PARAMETER                  |

|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número del último subíndice |
| Acceso       | ro                          |
| Mapeable     | No                          |
| Rango        | UNSIGNED8                   |
| Valor Padrón | 2                           |

|              |  |
|--------------|--|
| Subíndice    | 1  |
| Descripción  | COB-ID usado por el PDO                        |
| Acceso       | rw   |
| Mapeable     | No   |
| Rango        | UNSIGNED32                                     |
| Valor Padrón | 1400h: 200h + Node-ID<br>1401h: 300h + Node-ID |

|              |                     |
|--------------|---------------------|
| Subíndice    | 2                   |
| Descripción  | Tipo de transmisión |
| Acceso       | rw                  |
| Mapeable     | No                  |
| Rango        | UNSIGNED8           |
| Valor Padrón | 254                 |

El subíndice 1 contiene el COB-ID del PDO de recepción. Siempre que un mensaje es enviado para la red, este objeto irá leer cual es el COB-ID de este mensaje, e caso este sea igual a valor de este campo, el mensaje será recibido por el dispositivo. Este campo es formado por un UNSIGNED32 con la siguiente estructura:

**Tabla 6.5:** Descripción del COB-ID

| Bit          | Valor | Descripción                        |
|--------------|-------|------------------------------------|
| 31 (MSB)     | 0     | PDO está habilitado                |
|              | 1     | PDO está deshabilitado             |
| 30           | 0     | RTR permitido                      |
| 29           | 0     | Tamaño del identificador = 11 bits |
| 28 – 11      | 0     | No utilizado, siempre 0            |
| 10 – 0 (LSB) | X     | COB-ID de 11 bits                  |

El bit 31 permite habilitar o deshabilitar el PDO. Los bits 30 y 29, que deben ser mantenidos en 0 (cero), indican respectivamente que el PDO acepta frames remotos (RTR frames) y que utiliza identificador de 11 bits. Como el convertidor de frecuencia CFW500 no utiliza identificadores de 29 bits, los bits de 28 hasta 11 deben ser mantenidos en 0 (cero), mientras que los bits de 10 hasta 0 (cero) son usados para configurar el COB-ID para el PDO.

El subíndice 2 indica el tipo de transmisión de este objeto, de acuerdo con la tabla que sigue.



**Tabla 6.6:** Descripción del tipo de transmisión

| Tipo de transmisión | Transmisión de PDOs |          |            |           |     |
|---------------------|---------------------|----------|------------|-----------|-----|
|                     | Cíclico             | Acíclico | Sincrónico | Asíncrono | RTR |
| 0                   |                     | •        | •          |           |     |
| 1 – 240             | •                   |          | •          |           |     |
| 241 – 251           | Reservado           |          |            |           |     |
| 252                 |                     |          | •          |           | •   |
| 253                 |                     |          |            | •         | •   |
| 254                 |                     |          |            | •         |     |
| 255                 |                     |          |            | •         |     |

- **Valores 0 – 240:** cualquier RPDOs programado en este rango posee el mismo funcionamiento. Al detectar un mensaje, él irá recibir los datos, sin embargo no actualizará los valores recibidos hasta detectar el próximo telegrama SYNC.
- **Valores 252 y 253:** no permitido para PDOs de recepción.
- **Valores 254 y 255:** indica que no posee relación con el objeto de sincronización. Al recibir unos mensajes, sus valores serán actualizados inmediatamente.

## PDO\_MAPPING

|        |                     |
|--------|---------------------|
| Índice | 1600h hasta 1601h   |
| Nombre | Receive PDO mapping |
| Objeto | Record              |
| Tipo   | PDO MAPPING         |

|              |   |
|--------------|---|
| Subíndice    | 0   |
| Descripción  | Número de objetos mapeados                                |
| Acceso       | RO  |
| Mapeable     | No  |
| Rango        | 0 = deshabilitado<br>1 ... 4 = número de objetos mapeados |
| Valor Padrón | 0   |

|              |                                   |
|--------------|-----------------------------------|
| Subíndice    | 1 hasta 4                         |
| Descripción  | 1º hasta 4º objeto mapeado no PDO |
| Acceso       | Rw                                |
| Mapeable     | No                                |
| Rango        | UNSIGNED32                        |
| Valor Padrón | Indicado en el archivo EDS        |

Este parámetro indica los objetos mapeados en los PDOs de recepción el convertidor de frecuencia CFW500. Para cada RPDO, es posible mapear hasta 4 objetos diferentes, desde que el tamaño total no ultrapase ocho bytes. El mapeado de un objeto es hecho indicando su índice, subíndice<sup>7</sup> y tamaño (en bits) en un campo UNSIGNED32, con el siguiente formato:

| UNSIGNED32          |                       |                               |
|---------------------|-----------------------|-------------------------------|
| Índice<br>(16 bits) | Subíndice<br>(8 bits) | Tamaño del objeto<br>(8 bits) |

Por ejemplo, analizando el mapeado padrón del PDO de recepción, tenemos:

- **Subíndice 0 = 2:** el RPDO posee dos objetos mapeados.
- **Subíndice 1 = 22AC.0010h:** el primero objeto mapeado posee índice igual a 22ACh, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0684 del convertidor, que representa la palabra de control vía CANopen.
- **Subíndice 2 = 22AD.0010h:** el segundo objeto mapeado posee índice igual a 22ADh, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0685 del convertidor, que representa la referencia de velocidad.

Es posible modificar este mapeado, modificando la cantidad o el número de los objetos mapeados. Recordar que en el máximo pueden ser mapeados hasta 4 objetos o 8 bytes.

<sup>7</sup> Si el objeto es del tipo VAR y no tiene sub-índice, el valor 0 (cero) debe ser indicado para el subíndice.



## ¡NOTA!

- Para poder modificar los objetos mapeados en un PDO, primero es necesario escribir el valor 0 (cero) en el subíndice 0 (cero). De este modo, los valores de los subíndices 1 hasta pueden ser modificados. Después de hecho el mapeado deseado, se debe escribir nuevamente en el subíndice 0 (cero) el número de objetos que fueran mapeados, habilitando nuevamente el PDO.
- Los valores recibidos a través de estos objetos no son guardados en la memoria no volátil del convertidor. De este modo, luego de un comando para apagar o de reset del equipamiento, los objetos modificados por un RPDO vuelven para el su valor padrón.
- No olvidar que los PDOs solamente pueden ser recibidos caso el CFW500 se encuentre en el estado operacional.

### 6.3.3 PDOs de Transmisión

Los PDOs de transmisión, o TPDOs, como el nombre dice, son responsables por transmitir datos para la red CANopen. El convertidor de frecuencia CFW500 posee 2 PDOs de transmisión, cada un pudiendo transmitir hasta 8 bytes de datos. De modo semejante a los RPDOs, cada TPDO posee dos parámetros para su configuración, un PDO\_COMM\_PARAMETER y un PDO\_MAPPING, conforme presentado a seguir.

#### PDO\_COMM\_PARAMETER

|        |                        |
|--------|------------------------|
| Índice | 1800h hasta 1801h      |
| Nombre | Transmit PDO Parameter |
| Objeto | Record                 |
| Tipo   | PDO COMM PARAMETER     |

|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número del último subíndice |
| Acceso       | ro                          |
| Mapeable     | No                          |
| Rango        | UNSIGNED8                   |
| Valor Padrón | 5                           |

|              |  |
|--------------|--|
| Subíndice    | 1  |
| Descripción  | COB-ID usado por el PDO                        |
| Acceso       | rw   |
| Mapeable     | No   |
| Rango        | UNSIGNED32                                     |
| Valor Padrón | 1800h: 180h + Node-ID<br>1801h: 280h + Node-ID |

|              |                     |
|--------------|---------------------|
| Subíndice    | 2                   |
| Descripción  | Tipo de transmisión |
| Acceso       | rw                  |
| Mapeable     | No                  |
| Rango        | UNSIGNED8           |
| Valor Padrón | 254                 |

|              |                            |
|--------------|----------------------------|
| Subíndice    | 3                          |
| Descripción  | Tiempo entre transmisiones |
| Acceso       | rw                         |
| Mapeable     | No                         |
| Rango        | UNSIGNED16                 |
| Valor Padrón | -                          |

|              |           |
|--------------|-----------|
| Subíndice    | 4         |
| Descripción  | Reservado |
| Acceso       | rw        |
| Mapeable     | No        |
| Rango        | UNSIGNED8 |
| Valor Padrón | -         |

|              |                                 |
|--------------|---------------------------------|
| Subíndice    | 5                               |
| Descripción  | Temporizador de eventos         |
| Acceso       | rw                              |
| Mapeable     | No                              |
| Rango        | 0 = deshabilitado<br>UNSIGNED16 |
| Valor Padrón | 0                               |

El subíndice 1 contiene el COB-ID del PDO de transmisión. Siempre que este PDO enviar un mensaje para la red, el identificador de este mensaje será este COB-ID. La estructura de este campo es descrita en la Tabla 6.5.

El subíndice 2 indica el tipo de transmisión de este objeto, que sigue descrito por la Tabla 6.6. Sin embargo, su funcionamiento es distinto para PDOs de transmisión:

- **Valor 0:** indica que la transmisión debe ocurrir inmediatamente luego de la recepción de un telegrama SYNC, más no periódicamente.
- **Valores 1 – 240:** el PDO debe ser transmitido a cada telegrama SYNC detectado (u ocurrencias múltiples de SYNC, de acuerdo con el número elegido entre 1 e 240).
- **Valor 252:** indica que el contenido del mensaje debe ser actualizado (más no enviado), luego de la recepción de un telegrama SYNC. El envío del mensaje debe ser hecho luego de la recepción de un frame remoto (RTR frame).
- **Valor 253:** el PDO debe actualizar y enviar un mensaje así que recibir un frame remoto.
- **Valores 254:** el objeto debe ser transmitido de acuerdo con el timer programado en el subíndice 5.
- **Valores 255:** el objeto es transmitido automáticamente cuando el valor de algún de los objetos mapeados en este PDO fuera modificado. Funciona por modificación de estado (*Change Of State*). Este tipo también permite que el PDO sea transmitido de acuerdo con el timer programado en el subíndice 5.

En el subíndice 3 es posible programar un tiempo mínimo (en múltiplos de 100us) que debe transcurrir para que, después de transmitido un telegrama, un nuevo telegrama pueda ser enviado por este PDO. El valor 0 (cero) deshabilita esta función.

El subíndice 5 contiene un valor para habilitar un temporizador para el envío automático de un PDO. De este modo, siempre que un PDO es configurado para el tipo asíncrono, es posible programar el valor de este temporizador (en múltiplos de 1ms), para que el PDO sea transmitido periódicamente en el tiempo programado.



### ¡NOTA!

- Se debe observar el tiempo programado en este temporizador, de acuerdo con la tasa de transmisión utilizada. Tiempos muy pequeños (próximos al tiempo de transmisión del telegrama) pueden monopolizar el bus, causando la retransmisión indefinida del PDO e impidiendo que otros objetos menos prioritarios posan transmitir sus datos.
- El tiempo mínimo permitido para esta función en el convertidor de frecuencia CFW500 es 2ms.
- Es importante observar el tiempo entre transmisiones programado en el subíndice 3 principalmente cuando el PDO es programado con el valor 255 en el subíndice 2 (Change Of State).
- No olvidar que los PDOs solamente pueden ser transmitidos caso el esclavo se encuentra en el estado operacional.

### PDO\_MAPPING

|        |                      |
|--------|----------------------|
| Índice | 1A00h hasta 1A01h    |
| Nombre | Transmit PDO mapping |
| Objeto | Record               |
| Tipo   | PDO MAPPING          |

|              |   |
|--------------|---|
| Subíndice    | 0   |
| Descripción  | Número del último subíndice                               |
| Acceso       | ro  |
| Mapeable     | No  |
| Rango        | 0 = deshabilitado<br>1 ... 4 = número de objetos mapeados |
| Valor Padrón | 0   |

|              |                                   |
|--------------|-----------------------------------|
| Subíndice    | 1 hasta 4                         |
| Descripción  | 1° hasta 4° objeto mapeado no PDO |
| Acceso       | rw                                |
| Mapeable     | No                                |
| Rango        | UNSIGNED32                        |
| Valor Padrón | 0                                 |

El PDO MAPPING para la transmisión funciona de modo semejante que para la recepción, sin embargo en este caso son definidos los datos que serán transmitidos por el PDO. Cada objeto mapeado debe ser colocado en el listado conforme presentado a seguir:

| UNSIGNED32          |                       |                               |
|---------------------|-----------------------|-------------------------------|
| Índice<br>(16 bits) | Subíndice<br>(8 bits) | Tamaño del objeto<br>(8 bits) |

Por ejemplo, analizando el mapeado padrón del cuarto PDO de transmisión, tenemos:

- **Subíndice 0 = 2:** este PDO posee dos objetos mapeados.
- **Subíndice 1 = 22A8.0010h:** el primero objeto mapeado posee índice igual a 22A8h, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0680 del convertidor, que representa el estado del convertidor.
- **Subíndice 2 = 22A9.0010h:** el segundo objeto mapeado posee índice igual a 22A9h, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0681 del convertidor, que representa la velocidad del motor.

Es posible modificar este mapeado, alterando la cantidad o el número de los parámetros mapeados. Recordar que en el máximo pueden ser mapeados 4 objetos o 8 bytes.



**¡NOTA!**

Para poder modificar los objetos mapeados en un PDO, primero es necesario escribir el valor 0 (cero) en el subíndice 0 (cero). De este modo, los valores de los subíndices 1 hasta 4 pueden ser modificados. Después de hecho el mapeado deseado, se debe escribir nuevamente en el subíndice 0 (cero) el número de objetos que fueran mapeados, habilitando nuevamente el PDO.

## 6.4 SYNCHRONIZATION OBJECT – SYNC

Este objeto es transmitido con el objetivo de permitir la sincronización de eventos entre los dispositivos de la red CANopen. Elle es transmitido por un productor SYNC, y los dispositivos que detectan su transmisión son denominados consumidores SYNC.

El convertidor de frecuencia CFW500 posee la función de consumidor SYNC y, por lo tanto, puede programar sus PDOs para que sean sincrónicos. PDOs sincrónicos son aquellos relacionados con el objeto de sincronización, y por lo tanto pueden ser programados para que sean transmitidos o actualizados con base en este objeto.

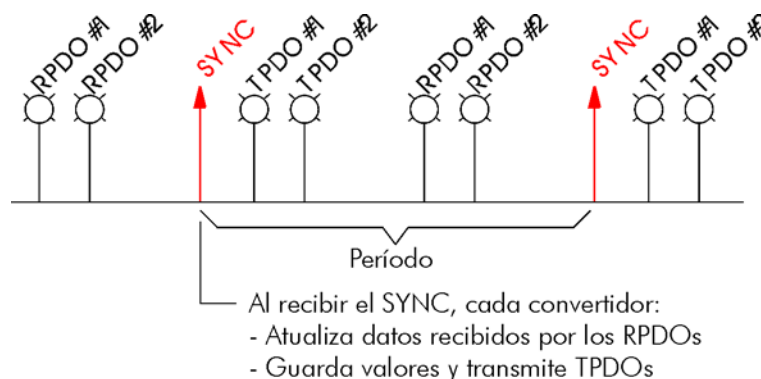


Figura 6.3: SYNC

El mensaje SYNC transmitido por el productor no posee dato alguno en su campo de datos, pues su objetivo es suministrar una base de tiempo para los demás objetos. En el convertidor de frecuencia CFW500, existe un objeto para la configuración del COB-ID del consumidor SYNC.

|        |             |
|--------|-------------|
| Indice | 1015h       |
| Nombre | COB-ID SYNC |
| Objeto | VAR         |
| Tipo   | UNSIGNED32  |

|              |            |
|--------------|------------|
| Acceso       | rw         |
| Mapeable     | No         |
| Rango        | UNSIGNED32 |
| Valor Padrón | 80h        |



**¡NOTA!**

Se debe observar el tiempo programado en el productor para el período de los telegramas SYNC, de acuerdo con la tasa de transmisión utilizada y el número de PDOs sincrónicos a ser transmitidos. Es necesario que se tenga tiempo suficiente para la transmisión de estos objetos, y también es recomendado que se tenga holgura para posibilitar el envío de mensajes asíncronas, como EMCY, PDOs asíncronos y SDOs.

## 6.5 NETWORK MANAGEMENT – NMT

El objeto de gestión de la red es responsable por un conjunto de servicios que controlan la comunicación del dispositivo en la red CANopen. Para el CFW500 están disponibles los servicios de control del nudo u de control de errores (utilizando *Node Guarding* o *Heartbeat*).

### 6.5.1 Control de los Estados del Esclavo

Con relación a la comunicación, un dispositivo de la red CANopen puede ser descrito por la siguiente máquina de estados:

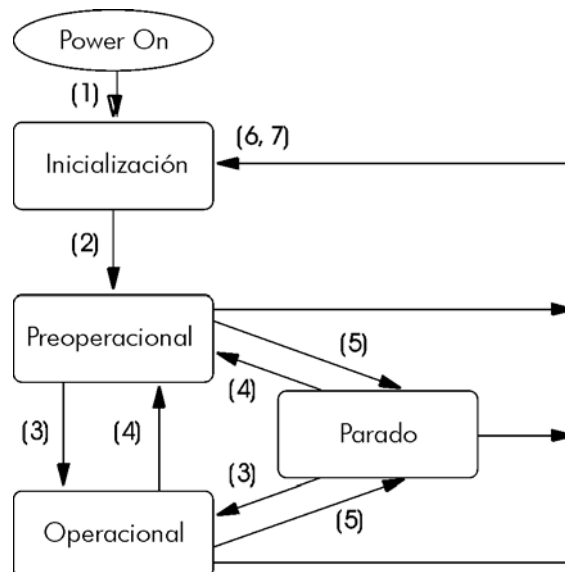


Figura 6.4: Diagrama de estados del nudo CANopen

Tabla 6.7: Descripción de las transiciones

| Transición | Descripción   |
|------------|---|
| 1          | Dispositivo es encendido y empieza la inicialización (automático).  |
| 2          | Inicialización concluida, va para el estado preoperacional (automático).  |
| 3          | Recibe comando Start Node para entrar en el estado operacional.   |
| 4          | Recibe comando Enter Pre-Operational, y va para el estado preoperacional.   |
| 5          | Recibe comando Stop Node para entrar en el estado parado.   |
| 6          | Recibe comando Reset Node, donde ejecuta el reset completo del dispositivo.   |
| 7          | Recibe comando Reset Communication, donde reinicializa el valor de los objetos y la comunicación CANopen del dispositivo. |

Durante la inicialización, es definido el Node-ID, creados los objetos y configurada la interface con la red CAN. No es posible se comunicar con el dispositivo en esta etapa, que es concluida automáticamente. En el final de esta etapa, el esclavo envía para la red un telegrama del objeto Boot-up, utilizado solo para indicar que la inicialización fue concluida y que el esclavo entro en el estado preoperacional. Este telegrama posee identificador 700h + Node-ID, y solo un byte de datos con valor igual a 0 (cero).

En el estado preoperacional, ya es posible se comunicar con el esclavo, sin embargo los PDOs todavía no están disponibles para operación. En el estado operacional, todos los objetos están disponibles, mientras que en el estado parado, solo el objeto NMT puede recibir o transmitir telegramas para la red. La tabla que sigue presenta los objetos disponibles para cada estado.

**Tabla 6.8:** Objetos accesibles en cada estado

|         | Inicialización | Preoperacional | Operacional | Parado |
|---------|----------------|----------------|-------------|--------|
| PDO     |                |                | •           |        |
| SDO     |                | •              | •           |        |
| SYNC    |                | •              | •           |        |
| EMCY    |                | •              | •           |        |
| Boot-up | •              |                |             |        |
| NMT     |                | •              | •           | •      |

Esta máquina de estados es controlada por el maestro de la red, que envía para cada esclavo, comandos para que sea ejecutado la transición de estados deseado. Estos telegramas no poseen confirmación, lo que significa que el esclavo solo recibe el telegrama sin retornar respuesta para el maestro. Los telegramas recibidos poseen la siguiente estructura:

| Identificador | byte 1             | byte 2          |
|---------------|--------------------|-----------------|
| 00h           | Código del comando | Node-ID destino |

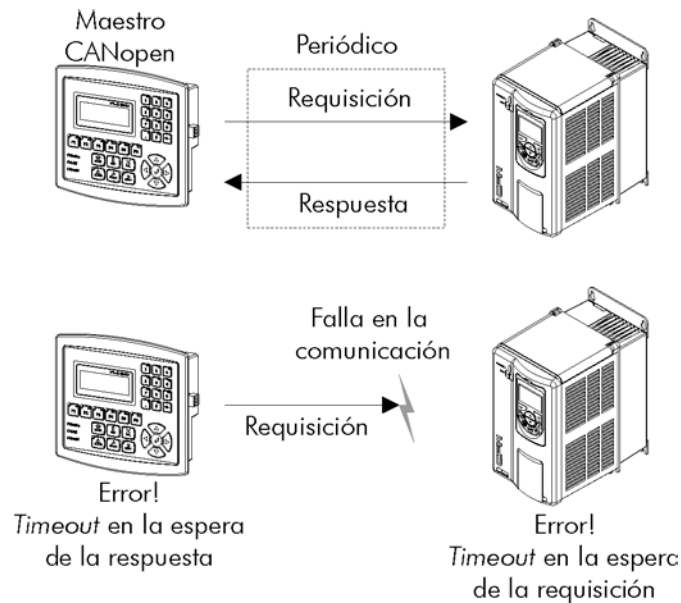
**Tabla 6.9:** Comandos para la transición de estados

| Código del comando                         | Node-ID destino                |
|--|--------------------------------|
| 1 = START node (transición 3)              | 0 = Todos los esclavos         |
| 2 = STOP node (transición 4)               | 1 ... 127 = Esclavo específico |
| 128 = Enter pre-operational (transición 5) |                                |
| 129 = Reset node (transición 6)            |                                |
| 130 = Reset communication (transición 7)   |                                |

Las transiciones indicadas en el código del comando equivalen a las transiciones de estado ejecutadas por el nudo luego de recibir el comando. El comando *Reset node* hace con que El convertidor de frecuencia CFW500 ejecute un reset completo del dispositivo, mientras que el comando *Reset communication* hace con que el dispositivo reinicialice solo os objetos relativos a la comunicación CANopen.

## 6.5.2 Control de Errores – Node Guarding

Este servicio es utilizado para posibilitar el monitoreo de la comunicación con la red CANopen, tanto por el maestro cuanto por el esclavo. En este tipo de servicio, el maestro envía telegramas periódicos para el esclavo, que contesta el telegrama recibido. Caso ocurra algún error que interrumpa la comunicación, será posible identificar este error, pues tanto el maestro cuanto el esclavo serán notificados por el *timeout* en la ejecución de este servicio. Los eventos de error son llamados de *Node Guarding* para el maestro, y de *Life Guarding* para el esclavo.



**Figura 6.5:** Servicio de control de errores – Node Guarding

Para el servicio de *Node Guarding*, existen dos objetos del diccionario para configuración de los tiempos para detección de errores de comunicación:

|        |            |
|--------|------------|
| Índice | 100Ch      |
| Nombre | Guard Time |
| Objeto | VAR        |
| Tipo   | UNSIGNED16 |

|              |            |
|--------------|------------|
| Acceso       | rw         |
| Mapeable     | No         |
| Rango        | UNSIGNED16 |
| Valor Padrón | 0          |

|        |                  |
|--------|------------------|
| Índice | 100Dh            |
| Nombre | Life Time Factor |
| Objeto | VAR              |
| Tipo   | UNSIGNED8        |

|              |           |
|--------------|-----------|
| Acceso       | rw        |
| Mapeable     | No        |
| Rango        | UNSIGNED8 |
| Valor Padrón | 0         |

El objeto 100Ch permite programar el tiempo necesario (en milisegundos) para que una ocurrencia de falla sea detectada, caso El convertidor de frecuencia CFW500 no reciba ninguno telegrama del maestro. El objeto 100Dh indica cuantas fallas en secuencia son necesarias hasta que se considere que ocurrió realmente un error de comunicación. Por lo tanto, la multiplicación de estos dos valores suministrará el tiempo total necesario para detección de errores de comunicación utilizando este objeto. El valor 0 (cero) deshabilita esta función.

Una vez configurado, El convertidor de frecuencia CFW500 empieza a contar estos tiempos a partir del primero telegrama *Node Guarding* recibido del maestro de la red. El telegrama del maestro es del tipo remoto, no poseyendo bytes de datos. El identificador es igual a 700h + Node-ID del esclavo destino. Ya el telegrama de respuesta del esclavo posee 1 byte de datos con la siguiente estructura:

| Identificador  | byte 1 |                    |
|----------------|--------|--------------------|
|                | bit 7  | bit 6 ... bit 0    |
| 700h + Node-ID | Toggle | Estado del esclavo |

Este telegrama posee un único byte de datos. Este byte contiene, en los siete bits menos significativos, un valor para indicar el estado del esclavo (4 = Parado, 5 = Operacional y 127 = Preoperacional), y en el octavo bit, un valor que debe ser modificado a cada telegrama enviado por el esclavo (*toggle bit*).

Caso El convertidor de frecuencia CFW500 detecte un error utilizando este mecanismo, él irá automáticamente para el estado preoperacional y indicará alarma A135 en su HMI.



**¡NOTA!**

- Este objeto está activo mismo en el estado parado (mirar tabla 6.8).
- El valor 0 (cero) en un de los dos objetos deshabilita esta función.
- Después de detectado el error, caso el servicio sea habilitado más una vez, la indicación del error es retirada de la HMI.
- El valor mínimo acepto para convertidor de frecuencia CFW500 es de 2ms, más llevándose en cuenta la tasa de transmisión y el número de puntos en la red, los tiempos programados para esa función deben ser coherentes, de manera que haya tiempo suficiente para transmisión de los telegramas y también para que el resto de la comunicación pueda ser procesada.
- Para cada esclavo, solamente un de los servicios – Heartbeat o Node Guarding – puede ser habilitado.

### 6.5.3 Control de Errores – Heartbeat

La detección de errores a través del mecanismo de *heartbeat* es hecha utilizando dos tipos de objetos: el productor *heartbeat* y el consumidor *heartbeat*. El productor es responsable por enviar telegramas periódicos para la red, simulando un batido del corazón, indicando que la comunicación está activa y sin errores. Un o más consumidores pueden monitorear estos telegramas periódicos y, caso estos telegramas dejen de ocurrir, significa que algún problema de comunicación ha ocurrido.

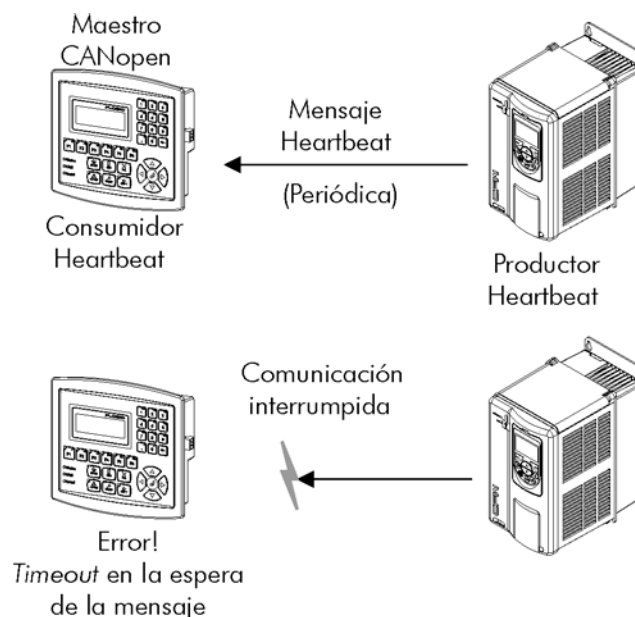


Figura 6.6: Servicio de control de errores – Heartbeat

Un mismo dispositivo de la red puede ser productor y consumidor de mensajes *heartbeat*. Por ejemplo, el maestro de la red puede consumir mensajes enviadas por un esclavo, permitiendo detectar problemas de comunicación con el esclavo, y al mismo tiempo el esclavo puede consumir mensajes *heartbeat* enviadas por el maestro, también posibilitando al esclavo detectar fallas en la comunicación con el maestro.

El convertidor de frecuencia CFW500 posee los servicios de productor y consumidor *heartbeat*. Como consumidor, es posible programar hasta 4 diferentes productores para que sean monitoreados por el convertidor:

|        |                         |
|--------|-------------------------|
| Indice | 1016h                   |
| Nombre | Consumer Heartbeat Time |
| Objeto | ARRAY                   |
| Tipo   | UNSIGNED32              |



|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número del último subíndice |
| Acceso       | ro                          |
| Mapeable     | No                          |
| Rango        | -                           |
| Valor Padrón | 4                           |

|              |                               |
|--------------|-------------------------------|
| Subíndice    | 1 – 4                         |
| Descripción  | Consumer Heartbeat Time 1 – 4 |
| Acceso       | rw                            |
| Mapeable     | No                            |
| Rango        | UNSIGNED32                    |
| Valor Padrón | 0                             |

En los subíndices de 1 hasta 4, es posible programar el consumidor escribiendo un valor en el siguiente formato:

| UNSIGNED32            |                     |                             |
|-----------------------|---------------------|-----------------------------|
| Reservado<br>(8 bits) | Node-ID<br>(8 bits) | Heartbeat time<br>(16 bits) |

- **Node-ID:** permite programar el Node-ID del productor *heartbeat* el cual se desea monitorear.
- **Heartbeat time:** permite programar el tiempo, en múltiplos de 1 milisegundo, hasta la detección de error, caso ningún mensaje del productor sea recibida. El valor 0 (cero) en este campo deshabilita el consumidor.

Después de configurado, el consumidor *heartbeat* inicia el monitoreo luego del primero telegrama enviado por el productor. Caso sea detectado error por el hecho del consumidor dejar de recibir mensajes del productos *heartbeat*, el convertidor irá automáticamente para el estado preoperacional e indicará A135 en la HMI.

Como productor, el convertidor de frecuencia CFW500 posee un objeto para configuración de este servicio:

|        |                         |
|--------|-------------------------|
| Índice | 1017h                   |
| Nombre | Producer Heartbeat Time |
| Objeto | VAR                     |
| Tipo   | UNSIGNED16              |

|              |           |
|--------------|-----------|
| Acceso       | rw        |
| Mapeable     | No        |
| Rango        | UNSIGNED8 |
| Valor Padrón | 0         |

El objeto 1017h permite programar el tiempo en milisegundos en el cual el productor envíe un telegrama *heartbeat* para la red. Una vez programado, el convertidor inicia la transmisión de mensajes con el siguiente formato:

| Identificador  | byte 1    |                    |
|----------------|-----------|--------------------|
|                | bit 7     | bit 6 ... bit 0    |
| 700h + Node-ID | Siempre 0 | Estado del esclavo |

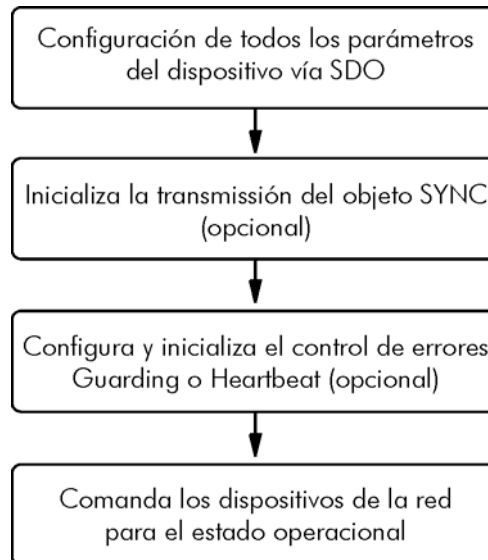


**¡NOTA!**

- Este objeto está activo mismo en el estado parado (ver tabla 6.8).
- El valor 0 (cero) en un de los objetos deshabilita esta función.
- Después de detectado el error, caso el servicio sea habilitado más una vez, la indicación del error es quitada de la HMI.
- El valor de tiempo programado para el consumidor debe ser mayor del que el programado para el respectivo productor. Recomendase inclusive programar el consumidor con valores múltiplos del utilizado para el productor.
- Para cada esclavo, solamente un de los servicios – Heartbeat o Node Guarding – puede ser habilitado.

## 6.6 PROCEDIMIENTO DE INICIALIZACIÓN

Una vez conocido el funcionamiento de los objetos disponibles para el convertidor de frecuencia CFW500, es necesario ahora programar los diferentes objetos para operaren en conjunto en la red. De modo general, el procedimiento para la inicialización de los objetos en una red CANopen sigue es siguiente diagrama de flujo:



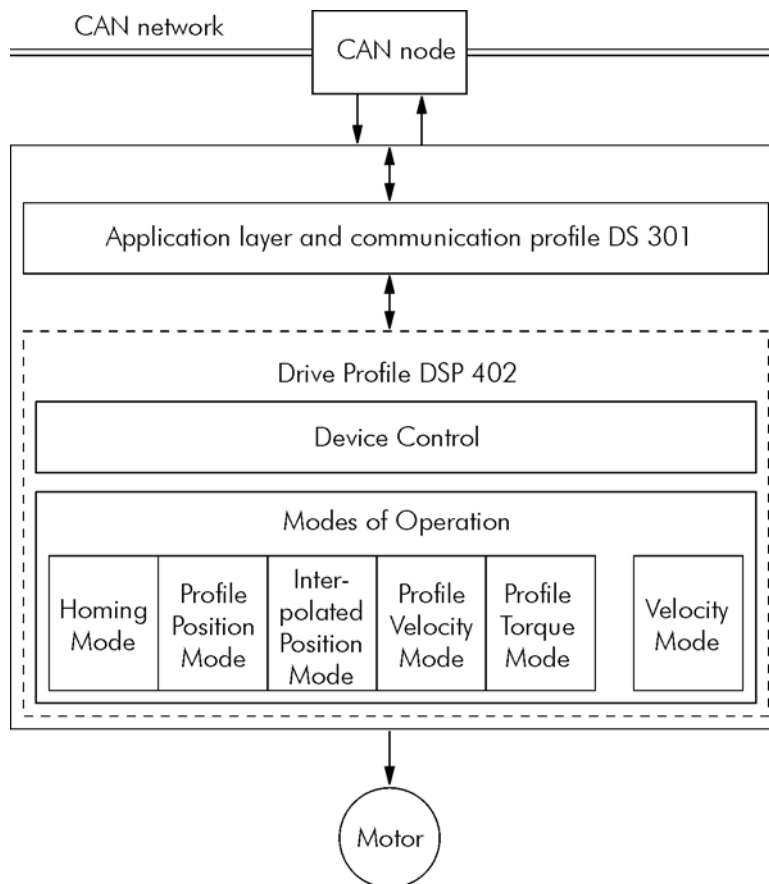
**Figura 6.7:** Diagrama de flujo del proceso de inicialización

Es necesario observar que los objetos de comunicación del convertidor de frecuencia CFW500 (1000h hasta 1FFFh) no son almacenados en la memoria no volátil. De este modo, siempre que fuera hecho el reset o apagado el equipo, es necesario rehacer la parametrización de los objetos de comunicación. Para los objetos específicos del fabricante (a partir de 2000h, que representan los parámetros), estos son almacenados en la memoria no volátil y, por lo tanto, pueden ser parametrizados una sola vez.

## 7 DESCRIPCIÓN DE LOS OBJETOS PARA DRIVES

En este ítem serán descritos los objetos comunes para drives, definidos por la especificación CANopen, en el documento CiA DSP 402. Los objetos mencionados aquí poseen descripción y operación semejantes, independiente del fabricante del drive. Eso facilita la interoperabilidad e intercambiabilidad entre diferentes dispositivos.

La figura 7.1 presenta un diagrama con la arquitectura lógica de funcionamiento de un drive a través de la red CANopen, con los diferentes modos de operación definidos en esta especificación. Cada modo de operación posee un conjunto de objetos que permite la configuración y operación del drive en la red.



**Figura 7.1:** Arquitectura de comunicación para un drive en la red CANopen

Para el CFW500, solamente el Velocity Mode es soportado. La tabla 7.1 presenta el listado de los objetos disponibles para el CFW500, divididos de acuerdo con los diferentes modos de operación del convertidor.

**Tabla 7.1:** Listado de objetos – Drive Profile

| Índice         | Objeto | Nombre                     | Tipo                        | Acceso | Mapeable |
|----------------|--------|----------------------------|-----------------------------|--------|----------|
| Device Control |        |                            |                             |        |          |
| 6040h          | VAR    | Controlword                | UNSIGNED16                  | rw     | Sí       |
| 6041h          | VAR    | Statusword                 | UNSIGNED16                  | ro     | Sí       |
| 6060h          | VAR    | Modes of Operation         | INTEGER8                    | rw     | Sí       |
| 6061h          | VAR    | Modes of Operation Display | INTEGER8                    | ro     | Sí       |
| 6502h          | VAR    | Supported drive modes      | INTEGER8                    | ro     | Sí       |
| Velocity Mode  |        |                            |                             |        |          |
| 6042h          | VAR    | vl target velocity         | INTEGER16                   | rw     | Sí       |
| 6043h          | VAR    | vl velocity demand         | INTEGER16                   | ro     | Sí       |
| 6044h          | VAR    | vl control effort          | INTEGER16                   | ro     | Sí       |
| 6046h          | ARRAY  | vl velocity min max amount | UNSIGNED32                  | rw     | Sí       |
| 6048h          | RECORD | vl velocity acceleration   | vl vel. accel. decl. record | rw     | Sí       |
| 6049h          | RECORD | vl velocity deceleration   | vl vel. accel. decl. record | rw     | Sí       |

Siempre que un objeto de este listado fuera leído o escrito, el CFW500 irá mapear sus funciones en los parámetros del convertidor. De este modo, a operar el sistema a través de estos objetos, el valor de los

parámetros puede ser modificado de acuerdo con la función utilizada. En los ítems siguientes, es hecho una descripción detallada de cada un de estos objetos, donde son indicados los parámetros del convertidor que son utilizados para ejecutar las funciones de estos objetos.

## 7.1 DEVICE CONTROL – OBJETOS PARA CONTROL DEL DRIVE

Todo el drive que opera en una red CANopen siguiendo la especificación DSP 402 debe estar de acuerdo con el descrito en la siguiente máquina de estados:

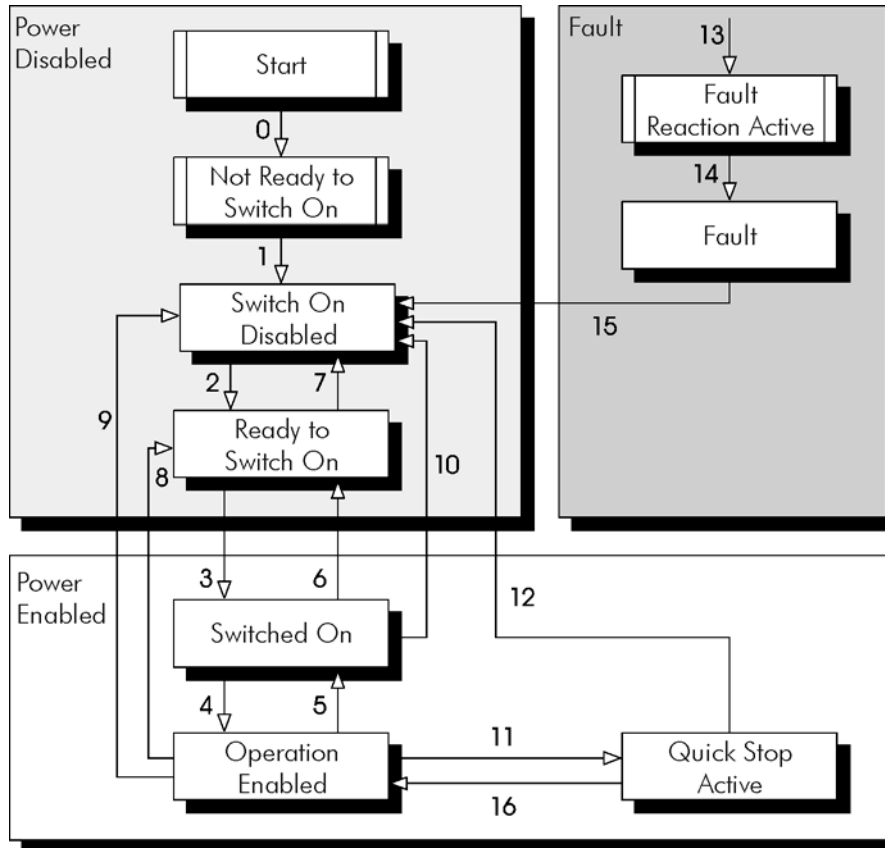


Figura 7.2: Máquina de estado para drives

Descripción de los estados:

- **Not ready to switch on:** el convertidor está inicializando, no puede ser comandado.
- **Switch on disabled:** inicialización completa, convertidor puede recibir comandos.
- **Ready to switch on:** comando para permitir la alimentación del drive fue recibido.
- **Switched on:** comando para energizar la potencia del drive fue recibido.
- **Operation enabled:** el drive está habilitado, sendo controlado de acuerdo con el modo de operación programado. Potencia está sendo aplicada al motor.
- **Quick stop active:** durante la operación, el comando de *quick stop* fue recibido. Potencia está sendo aplicada al motor.
- **Fault reaction active:** una falla ocurrió y el drive está ejecutando la acción relativa al tipo de error.
- **Fault:** drive con error. Función deshabilitada, sin potencia sendo aplicada en el motor.



### ¡NOTA!

El convertidor de frecuencia CFW500 no posee llave para bloqueo / habilitación de la alimentación de potencia del equipamiento. De esto modo, los estados descritos en el grupo *Power disabled* fueran implementados por cuestiones de compatibilidad con la máquina de estados descrita, sin embargo la alimentación de potencia del dispositivo permanece activa mismo en estos estados.

Descripción de las transiciones:

- **Transición 0:** el drive es encendido e inicia el procedimiento de inicialización. La alimentación de potencia del drive está activa.
- **Transición 1:** Inicialización completa (automático).
- **Transición 2:** Comando “*Shutdown*” recibido. Es hecha la transición de estados, más ninguna acción es tomada por el CFW500.
- **Transición 3:** Comando “*Switch on*” recibido. Es hecha la transición de estados, más ninguna acción es tomada por el CFW500.
- **Transición 4:** Comando “*Enable operation*” recibido. El drive es habilitado. Semejante a activar el bit 1 de la palabra de control vía CAN del convertidor – P0684.
- **Transición 5:** Comando “*Disable operation*” recibido. El drive es deshabilitado. Semejante a poner a cero el bit 1 de la palabra de control vía CAN del convertidor – P0684.
- **Transición 6:** Comando “*Shutdown*” recibido. Es hecha la transición de estados, más ninguna acción es tomada por el CFW500.
- **Transición 7:** Comandos “*Quick stop*” y “*Disable voltage*” recibidos. Es hecha la transición de estados, más ninguna acción es tomada por el CFW500.
- **Transición 8:** Comando “*Shutdown*” recibido. Durante la operación del drive este es deshabilitado, bloqueando la alimentación para el motor. Semejante a poner a cero el bit 1 de la palabra de control vía CAN del convertidor – P0684.
- **Transición 9:** Comando “*Shutdown*” recibido. Durante la operación del drive este es deshabilitado, bloqueando la alimentación para el motor. Semejante a poner a cero el bit 1 de la palabra de control vía CAN del convertidor – P0684.
- **Transición 10:** Comando “*Quick stop*” o “*Disable voltage*” recibido. Es hecha la transición de estados, más ninguna acción es tomada por el CFW500.
- **Transición 11:** Comando “*Quick stop*” recibido. Convertidor ejecuta la función de parada por rampa. Semejante a poner a cero el bit 0 de la palabra de control vía CAN del convertidor – P0684.
- **Transición 12:** Comando “*Disable voltage*” recibido. El drive es deshabilitado. Semejante a poner a cero el bit 1 de la palabra de control vía CAN del convertidor – P0684.
- **Transición 13:** Error es detectado y el drive es deshabilitado.
- **Transición 14:** Después de deshabilitar el drive, este va para el estado de error (automático).
- **Transición 15:** Comando “*Fault reset*” recibido. El convertidor ejecuta el reset del error y retorna para el estado deshabilitado y sin falla.
- **Transición 16:** Comando “*Enable operation*” recibido. Convertidor ejecuta la función de arranque por rampa. Semejante a activar el bit 0 de la palabra de control vía CAN del convertidor – P0684.

Esta máquina de estado es controlada por el objeto 6040h, y los estados pueden ser monitoreados por el objeto 6041h. Ambos los objetos son presentados a seguir.

### 7.1.1 Objeto 6040h – Controlword

Hace el control del estado del convertidor.

|                       |             |
|-----------------------|-------------|
| Índice                | 6040h       |
| Nombre                | Controlword |
| Objeto                | VAR         |
| Tipo                  | UNSIGNED16  |
| Parámetros utilizados | P0684       |

|              |            |
|--------------|------------|
| Acceso       | rw         |
| Mapeable     | Sí         |
| Rango        | UNSIGNED16 |
| Valor Padrón | -          |

Los bits de esta palabra poseen las siguientes funciones:

| 15 – 9    | 8    | 7           | 6 – 4                   | 3                | 2          | 1              | 0         |
|-----------|------|-------------|-------------------------|------------------|------------|----------------|-----------|
| Reservado | Halt | Fault reset | Operation mode specific | Enable operation | Quick stop | Enable voltage | Switch on |

Los bits 0, 1, 2, 3 y 7 permiten hacer el control de la máquina de estados del drive. Los comandos para transición de estados son hechos a través de las combinaciones de bits indicados en la Tabla 7.2. Los bits marcados con ‘x’ son irrelevantes para la ejecución del comando.

**Tabla 7.2: Comandos de la palabra de control**

| Comando           | Bits de la palabra de control |                  |            |                |           | Transiciones |
|-------------------|-------------------------------|------------------|------------|----------------|-----------|--------------|
|                   | Fault reset                   | Enable operation | Quick stop | Enable voltage | Switch on |              |
| Shutdown          | 0                             | x                | 1          | 1              | 0         | 2, 6, 8      |
| Switch on         | 0                             | 0                | 1          | 1              | 1         | 3            |
| Disable voltage   | 0                             | x                | x          | 0              | x         | 7, 9, 10, 12 |
| Quick stop        | 0                             | x                | 0          | 1              | x         | 7, 10, 11    |
| Disable operation | 0                             | 0                | 1          | 1              | 1         | 5            |
| Enable operation  | 0                             | 1                | 1          | 1              | 1         | 4, 16        |
| Fault reset       | 0 → 1                         | x                | x          | x              | x         | 15           |

Los bits 4, 5, 6 y 8 poseen diferentes funciones de acuerdo con el modo de operación utilizado. La descripción detallada de la función de estos bits para el modo velocidad (velocity mode) está descrita en el ítem 7.2.1.

Los bits 11, 12, 13, 14 y 15 están reservados para uso específico por el fabricante. Para convertidor de frecuencia CFW500, bit 11 se utiliza para seleccionar el modo de operación de local (0) o remoto (1).

### 7.1.2 Objeto 6041h – Statusword

Indica el estado actual del CFW500.

|                       |            |
|-----------------------|------------|
| Índice                | 6041h      |
| Nombre                | Statusword |
| Objeto                | VAR        |
| Tipo                  | UNSIGNED16 |
| Parámetros utilizados | P0680      |

|              |            |
|--------------|------------|
| Acceso       | ro         |
| Mapeable     | Sí         |
| Rango        | UNSIGNED16 |
| Valor Padrón | -          |

Los bits de esta palabra poseen las siguientes funciones:

**Tabla 7.3: Función de los bits de la palabra de estado (Statusword)**

| Bit     | Descripción             |
|---------|-------------------------|
| 0       | Ready to switch on      |
| 1       | Switched on             |
| 2       | Operation enabled       |
| 3       | Fault                   |
| 4       | Voltage enabled         |
| 5       | Quick stop              |
| 6       | Switch on disabled      |
| 7       | Warning                 |
| 8       | Reservado               |
| 9       | Remote                  |
| 10      | Target reached          |
| 11      | Internal limit active   |
| 12 – 13 | Operation mode specific |
| 14 – 15 | Reservado               |

En esta palabra, los bits 0, 1, 2, 3, 5 y 6 indican el estado del dispositivo de acuerdo con la máquina de estados presentada en la figura 7.2. La tabla 7.4 describe las combinaciones de estos bits para indicación de los estados. Los bits marcados con 'x' son irrelevantes para la indicación del estado.

**Tabla 7.4:** Estados del drive indicados a través de la palabra de estado

| Valor (binario)     | Estado                 |
|---------------------|------------------------|
| xxxx xxxx x0xx 0000 | Not ready to switch on |
| xxxx xxxx x1xx 0000 | Switch on disabled     |
| xxxx xxxx x01x 0001 | Ready to switch on     |
| xxxx xxxx x01x 0011 | Switched on            |
| xxxx xxxx x01x 0111 | Operation enabled      |
| xxxx xxxx x00x 0111 | Quick stop active      |
| xxxx xxxx x0xx 1111 | Fault reaction active  |
| xxxx xxxx x0xx 1000 | Fault                  |

Demás bits indican una condición específica para el drive.

- **Bit 4 – Voltage enabled:** indica que la potencia del drive está sendo alimentada.
- **Bit 7 – Warning:** No utilizado para el CFW500.
- **Bit 9 – Remote:** indica cuando el drive está en el modo remoto y acepta comandos vía red CANopen<sup>8</sup>. Representa el valor del bit 4 de la palabra de estado – P0680.
- **Bit 10 – Target reached:** indica cuando el drive está operando en el valor de la referencia, que depende del modo de operación utilizado. También es colocado en 1 cuando las funciones “quick stop” o “halt” son accionadas.
- **Bit 11 – Internal limit active:** no utilizado para el convertidor de frecuencia CFW500.
- **Bits 12 y 13 – Operation mode specific:** depende del modo de operación del drive.

### 7.1.3 Objeto 6060h – Modes of Operation

Permite programar el modo de operación del CFW500.

|                       |                    |
|-----------------------|--------------------|
| Índice                | 6060h              |
| Nombre                | Modes of operation |
| Objeto                | VAR                |
| Tipo                  | INTEGER8           |
| Parámetros utilizados | -                  |
|                       |                    |
| Acceso                | rw                 |
| Mapeable              | Sí                 |
| Rango                 | INTEGER8           |
| Valor Padrón          | -                  |

El único modo soportado por el convertidor CFW500 es el “*Velocity Mode*”, representado por el valor 2.

### 7.1.4 Objeto 6061h – Modes of Operation Display

Indica el modo de operación del CFW500.

|                       |                            |
|-----------------------|----------------------------|
| Índice                | 6061h                      |
| Nombre                | Modes of operation display |
| Objeto                | VAR                        |
| Tipo                  | INTEGER8                   |
| Parámetros utilizados | -                          |
|                       |                            |
| Acceso                | rw                         |
| Mapeable              | Sí                         |
| Rango                 | INTEGER8                   |
| Valor Padrón          | -                          |

El valor presentado en este objeto sigue las mismas opciones para objeto 6060h.

## 7.2 VELOCITY MODE – OBJETOS PARA CONTROL DEL DRIVE

Este modo de operación permite el control del convertidor de modo simple, disponiendo funciones del tipo:

<sup>8</sup> Depende de la programación del convertidor de frecuencia.

- Cálculo del valor de referencia.
- Captura y monitoreo de la velocidad.
- Limitación de velocidad.
- Rampas de velocidad, dentro otras funciones.

Estas funciones son ejecutadas con base en un conjunto de objetos para configuración de este modo de operación.

## 7.2.1 Bits de Control y Estado

Los bits 4, 5, 6 y 8 de la palabra de control (objeto 6040h – Controlword) poseen las siguientes funciones en el modo velocidad:

| Bit | Nombre                          | Valor | Descripción                            |
|-----|---------------------------------|-------|--|
| 4   |                                 |       | Reservado                              |
| 5   |                                 |       | Reservado                              |
| 6   | Reference ramp (habilita rampa) | 0     | Deshabilita Rampa – bit 0 do P0684 = 0 |
|     |                                 | 1     | Habilita Rampa – bit 0 do P0684 = 1    |
| 8   | Halt                            | 0     | Habilita Rampa – bit 0 do P0684 = 1    |
|     |                                 | 1     | Deshabilita Rampa – bit 0 do P0684 = 0 |

Para que el motor gire de acuerdo con la rampa de aceleración, es necesario que el bits 4 sea activado.

Para la palabra de status, los bits específicos del modo de operación (bits 12 y 13) son reservados para uso futuro.

## 7.2.2 Objeto 6042h – vl Target Velocity

Permite programar el valor de la referencia de velocidad para el convertidor, en rpm:

|                       |                    |
|-----------------------|--------------------|
| Índice                | 6042h              |
| Nombre                | vl target velocity |
| Objeto                | VAR                |
| Tipo                  | INTEGER16          |
| Parámetros utilizados | P0684, P0685       |

|              |           |
|--------------|-----------|
| Acceso       | rw        |
| Mapeable     | Sí        |
| Rango        | INTEGER16 |
| Valor Padrón | 0         |

El objeto vl target velocity permite la escritura de valores negativos para indicar referencia de velocidad para el motor en el sentido de giro antihorario. Esta modificación en el sentido de giro es ejecutada a través de la escritura del bit 2 en la palabra de comando vía CAN – P0684.

## 7.2.3 Objeto 6043h – vl Velocity Demand

Indica el valor de la referencia de velocidad luego de la rampa, en rpm:

|                       |                    |
|-----------------------|--------------------|
| Índice                | 6043h              |
| Nombre                | vl velocity demand |
| Objeto                | VAR                |
| Tipo                  | INTEGER16          |
| Parámetros utilizados | -                  |

|              |           |
|--------------|-----------|
| Acceso       | ro        |
| Mapeable     | Sí        |
| Rango        | INTEGER16 |
| Valor Padrón | -         |

## 7.2.4 Objeto 6044h – vl Control Effort

Indica el valor de la velocidad de acuerdo con el medido el motor, en rpm. Para los modos de control sin realimentación, este objeto posee el mismo valor que el objeto 6043h.



|                       |                   |
|-----------------------|-------------------|
| Índice                | 6044h             |
| Nombre                | vl control effort |
| Objeto                | VAR               |
| Tipo                  | INTEGER16         |
| Parámetros utilizados | P0681             |

|              |           |
|--------------|-----------|
| Acceso       | ro        |
| Mapeable     | Sí        |
| Rango        | INTEGER16 |
| Valor Padrón | -         |

### 7.2.5 Objeto 6046h – vl Velocity Min Max Amount

Permite programar el valor de la velocidad mínima y máxima para el convertidor. Solamente valores positivos son aceptos, sin embargo los valores programados también son válidos para el sentido de giro antihorario. Los valores son escritos en “rpm”.

|                       |                            |
|-----------------------|----------------------------|
| Índice                | 6046h                      |
| Nombre                | vl velocity min max amount |
| Objeto                | ARRAY                      |
| Tipo                  | UNSIGNED32                 |
| Parámetros utilizados | P0133, P0134               |

|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número do último sub-índice |
| Acceso       | ro                          |
| Mapeable     | No                          |
| Rango        | 2                           |
| Valor Padrón | 2                           |

|              |                        |
|--------------|------------------------|
| Subíndice    | 1                      |
| Descripción  | vl velocity min amount |
| Acceso       | rw                     |
| Mapeable     | Sí                     |
| Rango        | UNSIGNED32             |
| Valor Padrón | -                      |

|              |                        |
|--------------|------------------------|
| Subíndice    | 2                      |
| Descripción  | vl velocity max amount |
| Acceso       | rw                     |
| Mapeable     | Sí                     |
| Rango        | UNSIGNED32             |
| Valor Padrón | -                      |

### 7.2.6 Objeto 6048h – vl Velocity Acceleration

Permite programar la rampa de aceleración del convertidor.

|                       |  |
|-----------------------|--|
| Índice                | 6048h  |
| Nombre                | vl velocity acceleration                     |
| Objeto                | RECORD                                       |
| Tipo                  | VI velocity acceleration deceleration record |
| Parámetros utilizados | P100   |

|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número do último sub-índice |
| Acceso       | ro                          |
| Mapeable     | No                          |
| Rango        | 2                           |
| Valor Padrón | 2                           |

|              |             |
|--------------|-------------|
| Subíndice    | 1           |
| Descripción  | Delta speed |
| Acceso       | rw          |
| Mapeable     | Sí          |
| Rango        | UNSIGNED32  |
| Valor Padrón | -           |

|              |            |
|--------------|------------|
| Subíndice    | 2          |
| Descripción  | Delta time |
| Acceso       | rw         |
| Mapeable     | Sí         |
| Rango        | UNSIGNED32 |
| Valor Padrón | -          |

El valor de la aceleración es calculado dividiendo la velocidad programada en el subíndice 1 por el tiempo programado en el subíndice 2. Los valores programados deben respetar el rango de valores para el parámetro P100.

### 7.2.7 Objeto 6049h – VI Velocity Deceleration

Permite programar la rampa de desaceleración del convertidor.

|                       |  |
|-----------------------|--|
| Índice                | 6049h  |
| Nombre                | VI velocity deceleration                     |
| Objeto                | RECORD                                       |
| Tipo                  | VI velocity acceleration deceleration record |
| Parámetros utilizados | P101   |

|              |                             |
|--------------|-----------------------------|
| Subíndice    | 0                           |
| Descripción  | Número do último sub-índice |
| Acceso       | ro                          |
| Mapeable     | No                          |
| Rango        | 2                           |
| Valor Padrón | 2                           |

|              |             |
|--------------|-------------|
| Subíndice    | 1           |
| Descripción  | Delta speed |
| Acceso       | rw          |
| Mapeable     | Sí          |
| Rango        | UNSIGNED32  |
| Valor Padrón | -           |

|              |            |
|--------------|------------|
| Subíndice    | 2          |
| Descripción  | Delta time |
| Acceso       | rw         |
| Mapeable     | Sí         |
| Rango        | UNSIGNED32 |
| Valor Padrón | -          |

El valor de la desaceleración es calculada dividiendo la velocidad programada en el subíndice 1 por el tiempo programado en el subíndice 2. Los valores programados deben respetar el rango de valores para el parámetro P0101.

## 8 FALLAS Y ALARMAS RELACIONADAS CON LA COMUNICACIÓN CANOPEN

### A133/F233 – SIN ALIMENTACIÓN EN EL INTERFAZ CAN

**Descripción:**

Indica que el interfaz CAN no posee alimentación entre los terminales 1 y 5 del conector.

**Actuación:**

Para que sea posible enviar y recibir telegramas a través del interfaz CAN, es necesario suministrar alimentación externa para el circuito de interfaz.

Si la interfaz CAN estuviera alimentada y si fuera detectada la falta de alimentación en el interfaz CAN, será señalizada a través del HMI el mensaje de alarma A133 – o falla F233; dependiendo de la programación hecha en el parámetro P0313. Si la alimentación del circuito es restablecida, la comunicación CAN será reiniciada. Para la condición de alarma, la señalización desaparecerá automáticamente del HMI en el momento que la alimentación del circuito es reestablecida.

**Posibles Causas/Corrección:**

- Medir si hay tensión entre los terminales 1 y 5 del conector del interfaz CAN.
- Verificar si los cables de alimentación no están cambiados o invertidos.
- Verificar problemas de contacto en el cable o en el conector de la interfaz CAN.

### A134/F234 – BUS OFF

**Descripción:**

Detectado error de *bus off* en el interfaz CAN.

**Actuación:**

Caso el número de errores de recepción o transmisión detectados por el interfaz CAN sea muy alto<sup>9</sup>, el controlador CAN puede ser llevado al estado de *bus off*, donde él interrumpe la comunicación y deshabilita la interfaz CAN.

En este caso será señalizada a través del HMI el mensaje de alarma A134 – o falla F234, dependiendo de la programación hecha en el parámetro P0313. Para que la comunicación sea restablecida es necesario interrumpir y regresar nuevamente la alimentación para el producto o interrumpir y regresar nuevamente la alimentación del interfaz CAN; para que la comunicación sea reiniciada.

**Posibles Causas/Corrección:**

- Verificar cortocircuito en los cables de transmisión del circuito CAN.
- Verificar si los cables no están cambiados o invertidos.
- Verificar si todos los dispositivos de la red utilizan la misma tasa de comunicación.
- Verificar si resistores de terminación con valores correctos fueran colocados solamente en los extremos del bus principal.
- Verificar si la instalación de la red CAN fue hecha de manera adecuada.

### A135/F235 – NODE GUARDING/HEARTBEAT

**Descripción:**

Control de errores de la comunicación CANopen detecto error de comunicación utilizando el mecanismo de guarding.

**Actuación:**

Utilizando los mecanismos de control de error – Node Guarding o Heartbeat – el maestro y el esclavo pueden intercambiar telegramas periódicos, en un período predeterminado. Caso la comunicación sea interrumpida por

<sup>9</sup> Para más informaciones a respecto de la detección de errores, consultar la especificación CAN.

algún motivo, tanto el maestro cuanto el esclavo podrán detectar el error en la comunicación por el timeout en el intercambio de estos mensajes.

En este caso será señalizado a través del HMI el mensaje de alarma A135 – o falla F235, dependiendo de la programación hecha en el parámetro P0313. Para la condición de alarma, caso este control de errores sea nuevamente habilitado, la señalización de alarma desaparecerá automáticamente del HMI.

### Posibles Causas/Corrección:

- Verificar los tiempos programados en el maestro y en el esclavo para intercambio de los mensajes. Para evitar problemas debido al retraso en la transmisión y diferencias en el conteo de los tiempos, recomendase que los valores programados para intercambio de los mensajes en el maestro sea un poco menor que los tiempos programados para detección de errores por el esclavo.
- Verificar si el maestro está enviando los telegramas de “*guarding*” en el tiempo programado.
- Verificar problemas en la comunicación que puedan ocasionar pérdida de telegramas o retrasos en la transmisión.



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