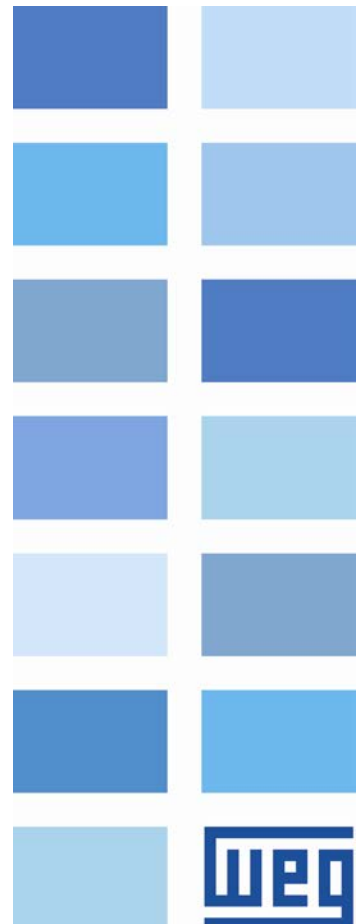


CANopen

SCA06

Manual del Usuario





Manual del Usuario CANopen

Serie: SCA06

Idioma: Español

N ° del Documento: 10001615309 / 01

Fecha de la Publicación: 08/2014

CONTENIDOS

CONTENIDOS.....	3
A RESPECTO DEL MANUAL	6
ABREVIACIONES Y DEFINICIONES	6
REPRESENTACIÓN NUMÉRICA.....	6
DOCUMENTOS.....	6
1 INTRODUCCIÓN A LA COMUNICACIÓN CANOPEN	7
1.1 CAN.....	7
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
1.2 CARACTERÍSTICAS DE LA RED CANOPEN.....	8
1.3 MEDIO FÍSICO	8
1.4 DIRECCIÓN EN LA RED CANOPEN	8
1.5 ACCESO A LOS DATOS.....	8
1.6 TRANSMISIÓN DE DATOS	8
1.7 OBJETOS RESPONSABLES POR LA COMUNICACIÓN – COBS.....	9
1.8 COB-ID	9
1.9 ARCHIVO EDS	10
2 INTERFAZ DE COMUNICACIÓN CANOPEN	11
2.1 CARACTERÍSTICAS DE LA INTERFAZ CAN.....	11
2.2 TERMINALES DEL CONECTOR.....	11
2.3 FUENTE DE ALIMENTACIÓN	11
2.4 SEÑALIZACIONES.....	12
3 INSTALACIÓN DE LA RED CANOPEN	13
3.1 TASA DE COMUNICACIÓN	13
3.2 DIRECCIÓN EN LA RED CANOPEN	13
3.3 RESISTOR DE TERMINACIÓN	13
3.4 CABLE	13
3.5 CONEXIÓN CON LA RED.....	14
4 PARAMETRIZACIÓN.....	15
4.1 SÍMBOLOS PARA DESCRIPCIÓN DE LAS PROPIEDADES	15
P0070 – ESTADO DEL CONTROLADOR CAN.....	15
P0071 – CONTADOR DE TELEGRAMAS CAN RECIBIDOS	15
P0072 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS	15
P0073 – CONTADOR DE ERRORES DE BUS OFF.....	16
P0074 – CONTADOR DE MENSAJES CAN PERDIDAS.....	16
P0075 – ESTADO DE LA COMUNICACIÓN CANOPEN	16
P0076 – ESTADO DEL NUDO CANOPEN	16
P0202 – MODO DE OPERACIÓN.....	17
P00662 – ACCIÓN PARA ERROR DE COMUNICACIÓN	17
P0700 – PROTOCOLO CAN	18
P0701 – DIRECCIÓN CAN.....	18
P0702 – TASA DE COMUNICACIÓN CAN	18
P0703 – RESET DE BUS OFF	19
P0704 – FOLLOW.....	19

P0705 – COB ID FOLLOW	20
P0706 – PERÍODO FOLLOW	20
4.2 FUNCIÓN FOLLOW VÍA CANOPEN	20
4.2.1 Follow Programado por Parámetros	20
4.2.2 Follow Programado por el Software WSCAN	21
5 DICCIONARIO DE OBJETOS	22
5.1 ESTRUCTURA DEL DICCIONARIO	22
5.2 TIPOS DE DATOS	22
5.3 COMMUNICATION PROFILE – OBJETOS PARA COMUNICACIÓN	22
5.4 MANUFACTURER SPECIFIC – OBJETOS ESPECÍFICOS DEL SCA06	23
5.4.1 Objeto 3000h – Entradas Digitales	24
5.4.2 Objeto 3001h – Salidas Digitales	24
5.4.3 Objetos 3002h a 3009h – Follow	25
5.5 DEVICE PROFILE – OBJETOS COMUNES PARA DRIVES	25
6 DESCRIPCIÓN DE LOS OBJETOS DE COMUNICACIÓN	26
6.1 OBJETOS DE IDENTIFICACIÓN	26
6.1.1 Objeto 1000h – Device Type	26
6.1.2 Objeto 1001h – Error Register	26
6.1.3 Objeto 1018h – Identity Object	27
6.2 SERVICE DATA OBJECTS – SDOS	27
6.2.1 Objeto 1200h – Servidor SDO	28
6.2.2 Funcionamiento de los SDOs	28
6.3 PROCESS DATA OBJECTS – PDOS	29
6.3.1 Objetos Mapeables para los PDOs	30
6.3.2 PDOs de Recepción	31
6.3.3 PDOs de Transmisión	33
6.4 SYNCHRONIZATION OBJECT – SYNC	35
6.5 NETWORK MANAGEMENT – NMT	36
6.5.1 Control de los Estados del Esclavo	36
6.5.2 Control de Errores – <i>Node Guarding</i>	37
6.5.3 Control de Errores – <i>Heartbeat</i>	39
6.6 PROCEDIMIENTO DE INICIALIZACIÓN	41
7 DESCRIPCIÓN DE LOS OBJETOS PARA DRIVES	42
7.1 DEVICE CONTROL – OBJETOS PARA CONTROL DEL DRIVE	43
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
7.1.5 Objeto 6502h – Supported Drive Modes	47
7.2 FACTOR GROUP – OBJETOS PARA CONVERSIÓN DE UNIDADES	48
7.2.1 Objeto 608Fh – Position Encoder Resolution	48
7.2.2 Objeto 6091h – Gear Ratio	48
7.2.3 Objeto 6092h – Feed Constant	49
7.3 POSITION CONTROL FUNCTION – CONTROLADOR DE POSICIÓN	50
7.3.1 Objeto 6063h – Position Actual Value	50
7.3.2 Objeto 6064h – Position Actual Value in User Units	50
7.4 PROFILE POSITION MODE – OBJETOS PARA CONTROL DEL DRIVE	51
7.4.1 Bits de Control y Estado	53
7.4.2 Objeto 607Ah – Target Position	54
7.4.3 Objeto 6081h – Profile Velocity	54
7.4.4 Objeto 6083h – Profile Acceleration	54
7.4.5 Objeto 6084h – Profile Deceleration	55
7.4.6 Objeto 6086h – Motion Profile Type	55
7.5 PROFILE VELOCITY MODE – OBJETOS PARA CONTROL DEL DRIVE	55

7.5.1	Bits de Control y Estado	55
7.5.2	Objeto 6069h – Velocity Sensor Actual Value	56
7.5.3	Objeto 606Bh – Velocity Demand Value	56
7.5.4	Objeto 606Ch – Velocity Actual Value.....	56
7.5.5	Objeto 60FFh – Target Velocity	57
7.6	PROFILE TORQUE MODE – OBJETOS PARA CONTROL DEL DRIVE	57
7.6.1	Bits de Control y Estado	57
7.6.2	Objeto 6071h – Target Torque	58
7.6.3	Objeto 6077h – Torque Actual Value.....	58
7.6.4	Objeto 6087h – Torque Slope.....	58
7.6.5	Objeto 6088h – Torque Profile Type.....	58
8	OPERACIÓN EN LA RED CANOPEN – MODO MAESTRO	60
8.1	HABILITACIÓN DE LA FUNCIÓN CANOPEN MAESTRO.....	60
8.2	CARACTERÍSTICAS DEL MAESTRO CANOPEN.....	60
8.3	OPERACIÓN DEL MAESTRO	60
8.4	BLOQUES PARA EL MAESTRO CANOPEN	61
8.4.1	CANopen SDO – Lectura/Escritura de Datos vía SDO.....	61
9	MARCADORES DE SISTEMA PARA CAN/CANOPEN	63
9.1	WORDS DE LECTURA DE ESTADO	63
9.2	WORDS DE ESCRITURA DE COMANDO	63
10	FALLAS Y ALARMYS RELACIONADAS CON LA COMUNICACIÓN CANOPEN .65	
	A133/F33 – SIN ALIMENTACIÓN EN EL INTERFAZ CAN	65
	A134/F34 – BUS OFF	65
	A135/F35 – NODE GUARDING/HEARTBEAT	65

A RESPECTO DEL MANUAL

Este manual suministra la descripción necesaria para la operación del convertidor de frecuencia SCA06 utilizando el protocolo CANopen. Este manual debe ser utilizado en conjunto con el manual del usuario del SCA06.

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 CANopen Application Layer and Communication Profile	4.02	CiA
CiA DRP 303-1 Cabling and Connector Pin Assignment	1.1.1	CiA
CiA DSP 306 Electronic Data Sheet Specification for CANopen	1.1	CiA
CiA DSP 402 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 SCA06. 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)¹. 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 bits² (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.

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

² 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 servoconvertidor SCA06, 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					
11 bits	Entero 32 bits			Entero 16 bits		
	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	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 SCA06 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 SCA06. 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

COB	Código de la Función (bits 10 – 7)	COB-ID Resultante (función + dirección)
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/ 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 SCA06 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 INTERFAZ DE COMUNICACIÓN CANOPEN

El servoconvertidor SCA06 posee, de manera estándar, en el producto, una interfaz CAN. La misma puede ser utilizada para comunicación en el protocolo CANopen como maestro o esclavo de la red. Las características de esta interfaz son descritas a seguir.

2.1 CARACTERÍSTICAS DE LA INTERFAZ CAN

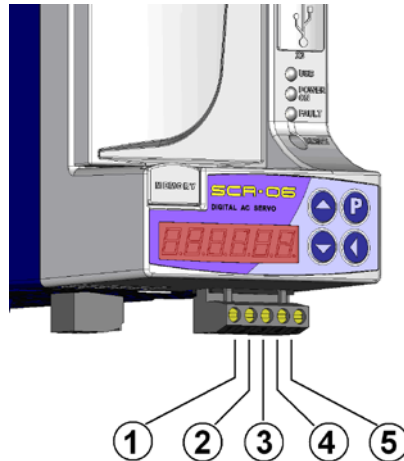


Figura 2.1: Detalle del conector CAN en la parte inferior del producto

- Interfaz aislada galvánicamente y con señal diferencial, otorgando mayor robustez contra interferencia electromagnética.
- Alimentación externa de 24 V.
- Permite la conexión de hasta 64 dispositivos en el mismo segmento. Una cantidad mayor de dispositivos puede ser conectada, también con uso de repetidores³.
- Longitud máxima del embarrado de 1000 metros.

2.2 TERMINALES DEL CONECTOR

La interfaz CAN posee un conector *plug-in* de 5 vías (X4) con los siguientes terminales:

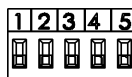


Tabla 2.1: Terminales del conector X4 para interfaz CAN

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

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

³ El número límite de equipamientos que pueden ser conectados en la red, también depende del protocolo utilizado.

2.4 SEÑALIZACIONES

Las indicaciones de alarmas, fallas y estados de la comunicación CANopen para el servoconvertidor SCA06 son hechas a través de la HMI y de los parámetros del producto.

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 CiA⁴.

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 servoconvertidor SCA06, la configuración de la tasa de comunicación es hecha a través del .

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 servoconvertidor SCA06, la configuración de la dirección es hecha a través del .

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 121Ω / 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 ²)
0 ... 40	70	0.25 ... 0.34
40 ... 300	<60	0.34 ... 0.60
300 ... 600	<40	0.50 ... 0.60
600 ... 1000	<26	0.75 ... 0.80

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.

⁴ Diferentes productos pueden tener diferentes longitudes máximas de cable permitida para la instalación.

3.5 CONEXIÓN CON LA RED

Para interconectar los diversos nodos 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.

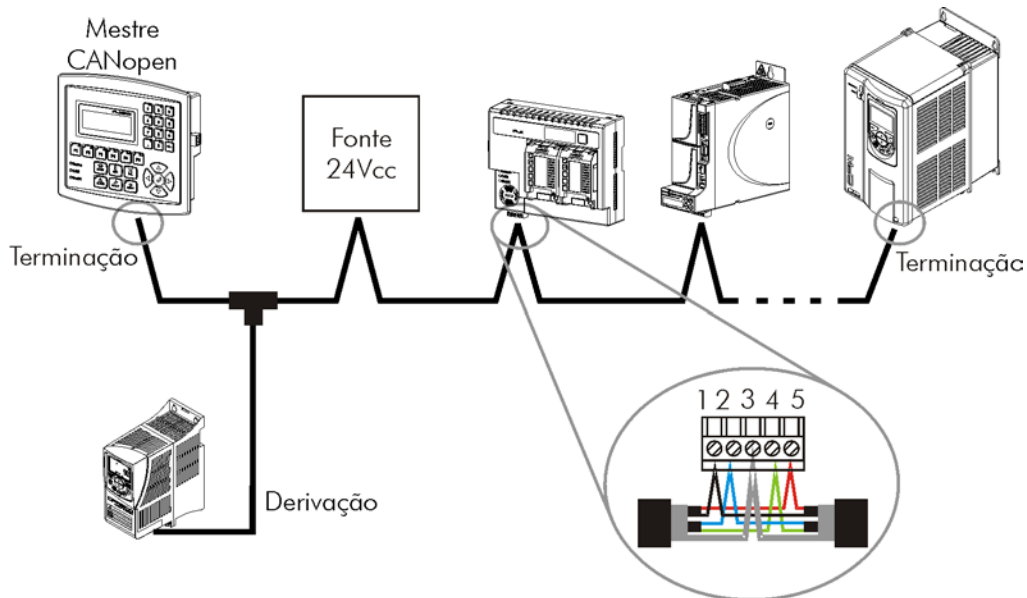


Figura 3.1: Ejemplo de instalación en red CANopen

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 solo los parámetros del servoconvertidor SCA06 que poseen relación con la comunicación CANopen.

4.1 SÍMBOLOS PARA DESCRIPCIÓN DE LAS PROPIEDADES

RO	Parámetro solamente de lectura.
CFG	Parámetro solamente modificable con el motor parado.
DP	Parámetro visibles a través de la HMI si el servoconvertidor poseer interfaz CANopen instalada.

P0070 – ESTADO DEL CONTROLADOR CAN

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

Descripción:

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

Tabla 4.1: Valores para el parámetro P0070

Valor	Descripción
0 = Deshabilitado	Interfaz CAN inactiva. Ocurre cuando el equipo no tiene protocolo CAN programado en P0700.
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.

P0071 – CONTADOR DE TELEGRAMAS CAN RECIBIDOS

Rango de Valores:	0 a 65535	Padrón: -
Propiedades:	RO	

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.

P0072 – CONTADOR DE TELEGRAMAS CAN TRANSMITIDOS

Rango de Valores:	0 a 65535	Padrón: -
Propiedades:	RO	

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.

P0073 – CONTADOR DE ERRORES DE BUS OFF

Rango de Valores:	0 a 65535	Padrón: -
Propiedades:	RO	

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.

P0074 – CONTADOR DE MENSAJES CAN PERDIDAS

Rango de Valores:	0 a 65535	Padrón: -
Propiedades:	RO	

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.

P0075 – ESTADO DE LA COMUNICACIÓN CANOPEN

Rango de Valores:	0 = Deshabilitado 1 = Reservado 2 = Comunicación Habilitada 3 = Control de Errores Habilitado 4 = Erro de Guarding 5 = Erro de Heartbeat	Padrón: -
Propiedades:	RO, CAN	

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*).

P0076 – ESTADO DEL NUDO CANOPEN

Rango de Valores:	0 = Deshabilitado 1 = Inicialización 2 = Parado 3 = Operacional 4 = Preoperacional	Padrón: -
Propiedades:	RO, CAN	

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.

P0202 – MODO DE OPERACIÓN

Rango de Valores:	1 = Modo Torque 2 = Modo Velocidad 3 = Reservado 4 = Modo Ladder 5 = CANopen 6 = Profibus DP	Padrón: 2
Propiedades:	CFG	

Descripción:

Este parámetro define el modo de operación del servoconvertidor SCA06. Para que el equipamiento sea controlado a través de la red CANopen, es necesario utilizar el modo 5 = CANopen. En caso de que este modo esté programado, serán dados comandos y referencias para operación del producto vía red CANopen, utilizando los objetos definidos en el diccionario de objetos.

Entre los principales objetos utilizados para control y monitoreo del equipamiento, podemos citar:

- 6040h: ControlWord
- 6041h: StatusWord
- 6060h: Mode of operation
- 6063h: Position actual value
- 607Ah: Target position
- 60FFh: Target velocity
- 6071h: Target Torque

La descripción detallada de éstos y demás objetos es hecha en el ítem 7. Para detalles sobre los modos de operación de 1 a 4, consulte el manual del usuario del servoconvertidor SCA06.


NOTA:

- El control del equipamiento a través de los objetos para drives solamente es posible para el modo de operación 5, no obstante, la comunicación CANopen puede ser utilizada en cualquier modo de operación.
- Para el SCA06 operando como esclavo de la función Follow, se debe programar el modo de operación 4 (Ladder), y utilizar el bloque MC_GearInPos.

P00662 – ACCIÓN PARA ERROR DE COMUNICACIÓN

Rango de Valores:	0 = Inactivo 1 = Causa Falla 2 = Causa alarma y Stop 3 = Causa alarme y deshabilita drive	Padrón: 0
Propiedades:	RW	

Descripción:

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

Tabla 4.2: Opciones para el parámetro P00662

Opción	Descripción
0 = Solamente indica alarma	Sólo muestra el código de alarma en la HMI del servoconvertidor.
1 = Causa Falla	Causa falla y el servoconvertidor sólo vuelve a operar en caso de que sea hecho reset de fallas.
2 = Indica alarma y ejecuta STOP	Será hecha la indicación de alarma junto con la ejecución del comando STOP. Para que el servo salga de esta condición, será necesario realizar el reset de fallas o deshabilitar el drive.
3 = Indica alarma y deshabilita drive	Será hecha la indicación de alarma junto con la ejecución del comando "deshabilita".

Son considerados errores de comunicación los siguientes eventos:

Comunicación Serial (RS232/RS485):

- Alarma A00128/Falla F00028: *timeout* de la interfaz serial.

Comunicación CANopen:

- 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*).

P0700 – PROTOCOLO CAN

Rango de	0 = Deshabilitado	Padrón: 0
Valores:	1 = CANopen 2 = Reservado 3 = CANespecial 1	
Propiedades:		

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

Rango de	0 a 127	Padrón: 63
Valores:		
Propiedades:		

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.

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

Rango de	0 = 1 Mbit/s / <i>Autobaud</i>	Padrón: 0
Valores:	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>	
Propiedades:		

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*.

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.

P0703 – RESET DE BUS OFF

Rango de	0 = Manual	Padrón: 0
Valores:	1 = Automático	
Propiedades:		

Descripción:

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

Tabla 4.3: Opciones para el parámetro P0703

Opción	Descripción
0 = Reset Manual	Caso ocurra <i>bus off</i> , será señalado en el HMI la alarma A134/F34, la acción programada en el parámetro P0662 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 P0662.

P0704 – FOLLOW

Rango de	0 = Deshabilitado	Padrón: 0
Valores:	1 = Maestro Follow Real 2 = Maestro Follow Virtual 3 = Esclavo Follow	
Propiedades:	CFG	

Descripción:

Permite habilitar la función Follow vía CANopen, además de definir si el equipamiento deberá ser maestro (productor) o esclavo (consumidor) Follow.

Tabla 4.4: Opciones para el parámetro P0704

Opción	Descripción
0 = Deshabilitado	No envía mensaje Follow
1 = Maestro Follow Real	Envía telegramas Follow conteniendo posición y velocidad del eje real.
2 = Maestro Follow Virtual	Envía telegramas Follow contendo posición y velocidad del eje virtual.
3 = Esclavo Follow	Recibe telegramas Follow. Necesita bloque MC_GearInPos para ejecutar función Follow

Una vez programado como maestro o esclavo, el servoconvertidor SCA06 deberá entrar automáticamente en modo operacional, en la red CANopen, de forma de posibilitar el intercambio de PDOs entre los equipamientos de la red.

Para detalles sobre el funcionamiento de la función Follow consulte el ítem 4.2.1.

P0705 – COB ID FOLLOW

Rango de	385 – 511	Padrón: 0
Valores:		
Propiedades:	CFG	

Descripción:

Define el COB ID (Communication Object Identifier) del PDO Follow. El rango de valores 385 (181h) a 511 (1FFh) es definido por la especificación CANopen como rango estándar para el TPDO1. Tanto el maestro como los esclavos deben utilizar el mismo COB ID.

La función del TPDO1 (maestro) y del RPDO1 (esclavos) es dedicada para esta función, por tanto, estos PDOs no deben ser configurados para comunicación de otros datos.

Para detalles sobre el funcionamiento de la función Follow consulte el ítem 4.2.1.

P0706 – PERÍODO FOLLOW

Rango de	0,2 a 5,0 ms	Padrón: 1,0
Valores:		
Propiedades:	CFG	

Descripción:

Permite programar el período de transmisión del telegrama Follow por el maestro de la red. No es utilizado por los esclavos.

Cuanto menor es el período, más rápidamente las referencias son transmitidas y más preciso será el sincronismo, no obstante, el tiempo de ocupación del embarado también será mayor, lo que puede dificultar la comunicación en caso de que existan otros datos a ser comunicados por la red CANopen. Este período también debe ser programado en función de la tasa de comunicación. A 1 Mbit/s, un telegrama Follow lleva aproximadamente 100 us para ser transmitido. A medida que la tasa de comunicación queda más lenta, este tiempo de transmisión aumenta proporcionalmente.

Si apenas la función Follow es utilizada en la red, se puede programar el menor período posible, una vez que no existirán otros telegramas transmitidos en la red. Sin embargo, si la función de maestro CANopen es utilizada en paralelo, es importante que haya tiempo disponible para transmisión de los demás telegramas CANopen. A título de recomendación, cuando el maestro CANopen es utilizado, los telegramas Follow deben ocupar cerca de 10 a 20 % del tiempo en el embarado.

Para detalles sobre el funcionamiento de la función Follow consulte el ítem 4.2.1.

4.2 FUNCIÓN FOLLOW VÍA CANOPEN

La función Follow permite realizar el sincronismo de posición entre dos o más servomotores. El sincronismo es establecido a través del envío de telegramas de tipo PDO, donde el maestro Follow envía valores de posición y velocidad del motor, que serán utilizados como referencia por uno o más esclavos Follow.

La función Follow puede ser programada por dos fuentes diferentes: por parámetros o por el software WSCAN.

4.2.1 Follow Programado por Parámetros

En caso de que no se desee utilizar la función de maestro CANopen disponible para el servoconvertidor SCA06, se puede programar la función Follow apenas utilizando parámetros. En este caso, los parámetros P0704, P0705 y P0706 son utilizados, y es necesario hacer la programación/configuración de los siguientes elementos:

- **Red:** para utilización de la función Follow, es necesario primeramente configurar la interfaz CANopen, definiendo protocolo, dirección y tasa de comunicación, así como realizar la instalación necesaria para la comunicación – cables, alimentación, resistores de terminación, etc.

- **Maestro:** para el maestro (o productor), es necesario habilitar la función Follow, seleccionar el eje de referencia, y programar el COB ID y el período de transmisión.
- **Esclavos:** en los esclavos (o consumidores), se debe habilitar la función Follow en el modo esclavo y programar el COB ID, igual al programado para el maestro. Para que los esclavos reciban y utilicen estos valores de velocidad y posición, es necesario también que los mismos sean programados para el modo Ladder (P0204 = 4), y se debe utilizar el bloque MC_GearInPos. Más detalles sobre la configuración del bloque MC_GearInPos en el Help del software WLP.

4.2.2 Follow Programado por el Software WSCAN

Otra opción para programación de la función Follow es a través del software WSCAN. En este caso, los parámetros P0704, P0705 y P0706 no deben ser programados, y la habilitación de la función es realizada a través de la ventana de configuración del servoconvertidor SCA06 en el software WSCAN.

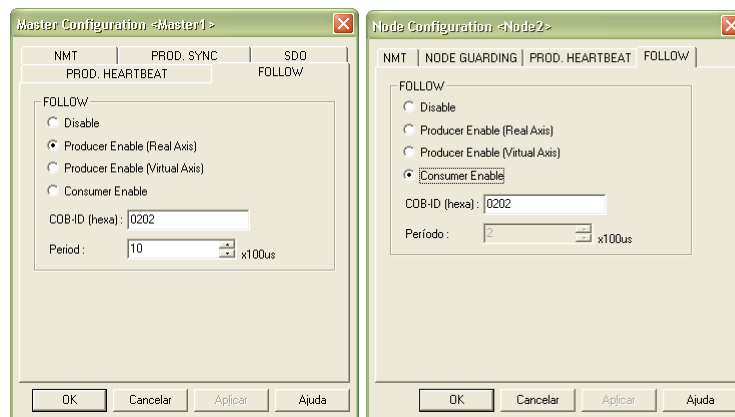


Figura 4.1: Ventanas de configuración para maestro y esclavo Follow

Una vez programado, durante la inicialización de los esclavos con la función Follow, el maestro de red CANopen hará la programación de la función utilizando objetos internos del equipamiento. La programación es realizada online, y no es necesario reiniciar el equipamiento para que los nuevos valores sean utilizados.

Junto con la programación de la función Follow, también es posible programar otros servicios para la red CANopen, como PDOs, SDOs, Node Guarding, etc., observando siempre que el tiempo de ocupación del embarrado CAN, por los telegramas Follow, debe permitir la utilización de estos otros servicios.

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 servoconvertidor SCA06, operando en modo esclavo.

Tabla 5.2: Listado de objetos – Communication Profile

Índice	Objeto	Nome	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
...				
1407h	RECORD	8th 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
...				
1607h	RECORD	8th 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
...				
1807h	RECORD	8th 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
...				
1A07h	RECORD	8th 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 SCA06

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 servoconvertidor SCA06, en este rango de objetos fue proporcionado todo el listado de parámetros. A través de estos parámetros es posible operar el SCA06, 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 – Manufacturer Specific

Índice	Objeto	Nombre	Tipo	Acceso
2000h	VAR	P0000 – Parámetro de acceso	INTEGER16	rw
2002h	VAR	P0002 – Velocidad del motor	INTEGER16	ro
2003h	VAR	P0003 – Corriente del motor	INTEGER16	ro
2004h	VAR	P0004 – Tensión CC	INTEGER16	ro
...
2077h	VAR	P0119 – Referencia de Corriente	INTEGER16	rw
2079h	VAR	P0121 – Referencia de Velocidad	INTEGER16	rw
...

Para el listado completo y una descripción detallada de los parámetros, consulte el manual del SCA06. Es necesario reconocer la operación del convertidor a través de los parámetros para que se pueda programar correctamente su operación vía red CANopen.

Además de los parámetros, el SCA06 presenta los siguientes objetos:

- 0x3000h – entradas digitales.
- 0x3001h – salidas digitales.
- 0x3002h – Follow Position Actual Value.
- 0x3003h – Follow Velocity Actual Value.
- 0x3004h – Follow Target Position.
- 0x3005h – Follow Target Velocity.
- 0x3008h – Follow Type.
- 0x3009h – Follow Period.

5.4.1 Objeto 3000h – Entradas Digitales

Permite acceso al estado de las entradas digitales del servoconvertidor SCA06.

Índice	3000h
Nombre	Entradas Digitales
Objeto	Array
Tipo	UNSIGNED16

Sub-índice	0
Descripción	Number of Entries
Acceso	ro
Mapeable	No
Rango	UNSIGNED16

Sub-índice	1
Descripción	Entrada digital estándar
Acceso	ro
Mapeable	Sí
Rango	UNSIGNED16

Sub-índice	2
Descripción	Entrada digital Slot 1
Acceso	Ro
Mapeable	Sí
Rango	UNSIGNED16

Sub-índice	3
Descripción	Entrada digital Slot 2
Acceso	Ro
Mapeable	Sí
Rango	UNSIGNED16

Sub-índice	4
Descripción	Entrada digital Slot 3
Acceso	Ro
Mapeable	Sí
Rango	UNSIGNED16

5.4.2 Objeto 3001h – Salidas Digitales

Permite controlar el estado de las salidas digitales del servoconvertidor SCA06.

Índice	3001h
Nombre	Salidas Digitales
Objeto	Array
Tipo	UNSIGNED8

Sub-índice	0
Descripción	Number of Entries
Acceso	ro
Mapeable	No
Rango	UNSIGNED8

Sub-índice	1
Descripción	Salida digital estándar
Acceso	Ro
Mapeable	Sí
Rango	UNSIGNED8

Sub-índice	2
Descripción	Salida digital Slot 1
Acceso	Ro
Mapeable	sí
Rango	UNSIGNED8

Sub-índice	3
Descripción	Salida digital Slot 2
Acceso	ro
Mapeable	sí
Rango	UNSIGNED8

Sub-índice	4
Descripción	Salida digital Slot 3
Acceso	Ro
Mapeable	Sí
Rango	UNSIGNED8

5.4.3 Objetos 3002h a 3009h – Follow

Los objetos 3002h a 3009h son objetos de uso exclusivo de la función Follow.

5.5 DEVICE PROFILE – OBJETOS COMUNES PARA DRIVES

La documentación CANopen también incluí propuestas para la estandarización de determinados tipos de dispositivos. El servoconvertidor SCA06 sigue el descrito por la CiA DPS 402 – *Device Profile Drives and Motion Contrd.* 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 servoconvertidores), 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

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 descritos detalladamente cada uno de los objetos de comunicación disponibles para el servoconvertidor SCA06 operando en modo esclavo. Es necesario conocer como estos objetos son operados para utilizar las funciones disponibles para la comunicación del drive.



¡NOTA!

El servoconvertidor SCA06 puede operar como maestro o esclavo de la red CANopen. Los objetos descritos a seguir describen la operación del equipamiento como esclavo de la red CANopen. Para la descripción de las características del producto operando como maestro de la red CANopen, consulte el ítem 8 junto con el software de configuración de la red CANopen WSCAN.

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	0002.0192h

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 SCA06 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	0000.0700h

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 SCA06

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 servoconvertidor SCA06 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.

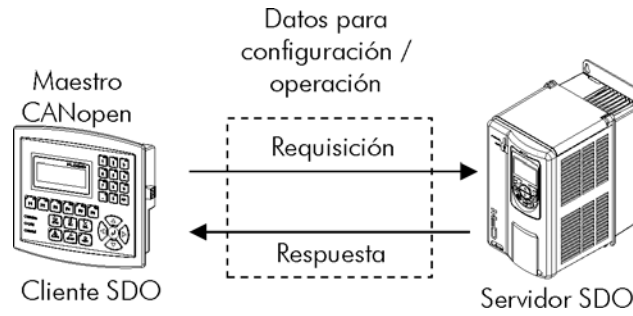


Figura 6.1: Comunicación entre cliente y servidor SDO

6.2.1 Objeto 1200h – Servidor SDO

El servoconvertidor SCA06 operando en modo esclavo 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 11 bits	8 bytes de datos							
	Comando byte 0	Índice byte 1 byte 2		Subíndice byte 3	Datos del objeto 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 SCA06 (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 SCA06 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 SCA06 contesta a la requisición, indicando que el valor para el referido objeto es igual a 999⁵:

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.

⁵ No olvidar que cualquier dato del tipo entero, el orden de transferencia de los bytes va del menos significativo hasta el más significativo.

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.

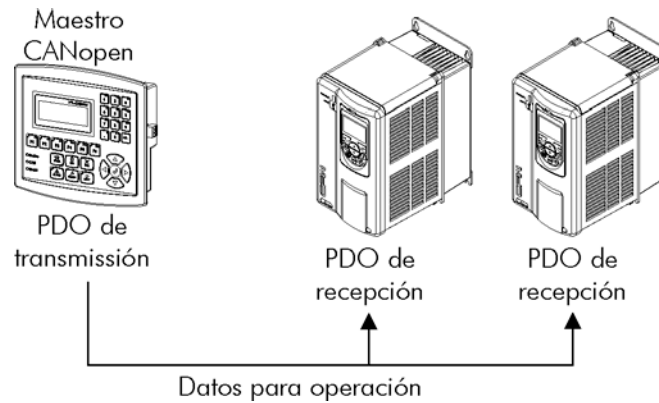


Figura 6.2: Comunicación utilizando PDOs



¡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	INTEGER16	ro
2003h	VAR	P0003 – Corriente del Motor	INTEGER16	ro
2004h	VAR	P0004 – Tensión Bus CC (Ud)	UNSIGNED16	ro
2008h	VAR	P0008 – Estado DI1 a DI3	UNSIGNED16	ro
2063h	VAR	P0099 – Habilita	UNSIGNED16	rw
2077h	VAR	P0119 – Referencia de Corriente	INTEGER16	rw
2079h	VAR	P0121 – Referencia de Velocidad	INTEGER16	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 servoconvertidor SCA06 operando en modo esclavo posee 8 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

Índice	1400h hasta 1407h
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 1401h: 300h + Node-ID 1402h: 400h + Node-ID 1403h: 500h + Node-ID 1404h – 1407h: 0

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

O 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 recibida 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 servoconvertidor SCA06 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 1607h
Nombre	Receive PDO mapping
Objeto	Record
Tipo	PDO MAPPING

Subíndice	0
Descripción	Número de objetos mapeados
Acceso	RO
Mapeable	No
Rango	0 = deshabilitado 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 servoconvertidor SCA06. 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, subíndice⁶ y tamaño (en bits) en un campo UNSIGNED32, con el siguiente formato:

UNSIGNED32		
Índice (16 bits)	Subíndice (8 bits)	Tamaño del objeto (8 bits)

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

- **Subíndice 0 = 2:** el RPDO posee dos objetos mapeados.
- **Subíndice 1 = 2063.0010h:** el primero objeto mapeado posee índice igual a 2063h, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0099 del drive, que representa el comando de habilitación.
- **Subíndice 2 = 2079.0010h:** el segundo objeto mapeado posee índice igual a 2079h, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0121 del drive, 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.

⁶ 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 SCA06 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 servoconvertidor SCA06 posee 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 1807h
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 1801h: 280h + Node-ID 1802h: 380h + Node-ID 1803h: 480h + Node-ID 1804h – 1807h: 0

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 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 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 servoconvertidor SCA06 es 1ms.
- 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 1A07h
Nombre	Transmit PDO mapping
Objeto	Record
Tipo	PDO MAPPING

Subíndice	0
Descripción	Número del último subíndice
Acceso	ro
Mapeable	No
Rango	0 = deshabilitado 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 (16 bits)	Subíndice (8 bits)	Tamaño del objeto (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 = 2002.0010h:** el primero objeto mapeado posee índice igual a 2002h, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0002 del drive, que representa la velocidad del motor.
- **Subíndice 2 = 2023.0010h:** el segundo objeto mapeado posee índice igual a 2023h, subíndice 0 (cero), y tamaño igual a 16 bits. Este objeto corresponde al parámetro P0023 del drive, que representa la falla actual.

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 servoconvertidor SCA06 operando en modo esclavo 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.

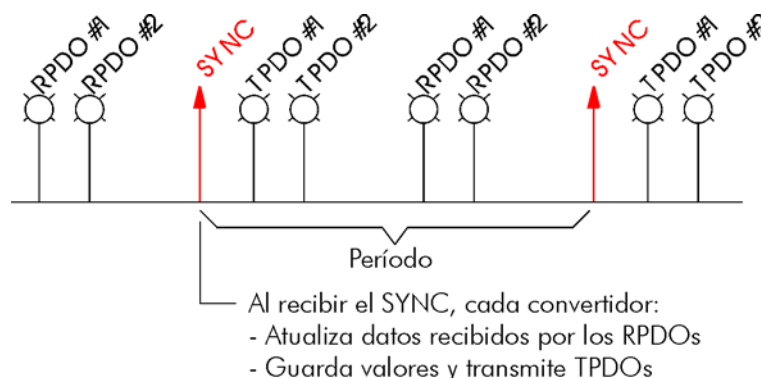


Figura 6.3: SYNC

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 servoconvertidor SCA06, existe un objeto para la configuración del COB-ID del consumidor SYNC.

Índice	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 SCA06 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:

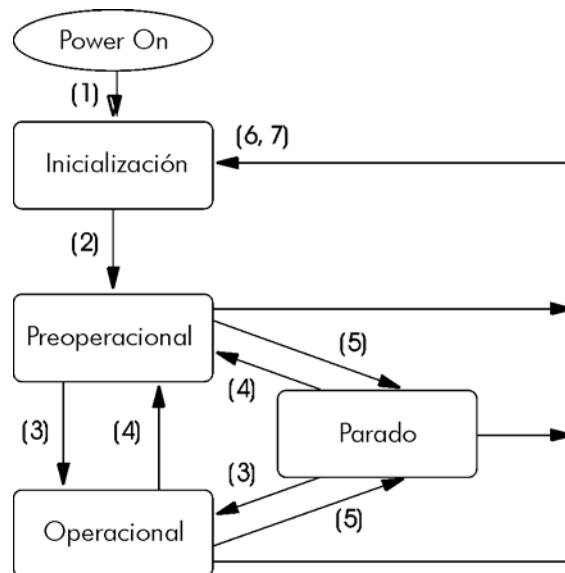


Figura 6.4: Diagrama de estados del nudo CANopen

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 servoconvertidor SCA06 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.

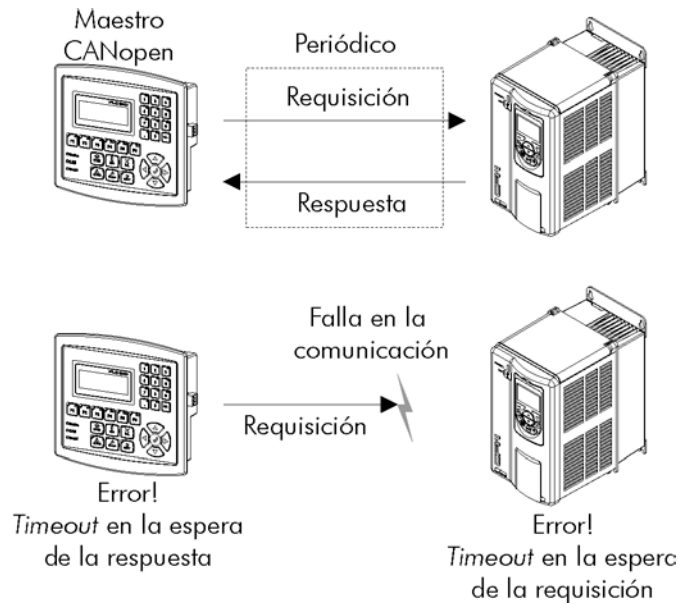


Figura 6.5: Servicio de control de errores – Node Guarding

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 servoconvertidor SCA06 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 servoconvertidor SCA06 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 servoconvertidor SCA06 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 servoconvertidor SCA06 es de 1ms, 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.

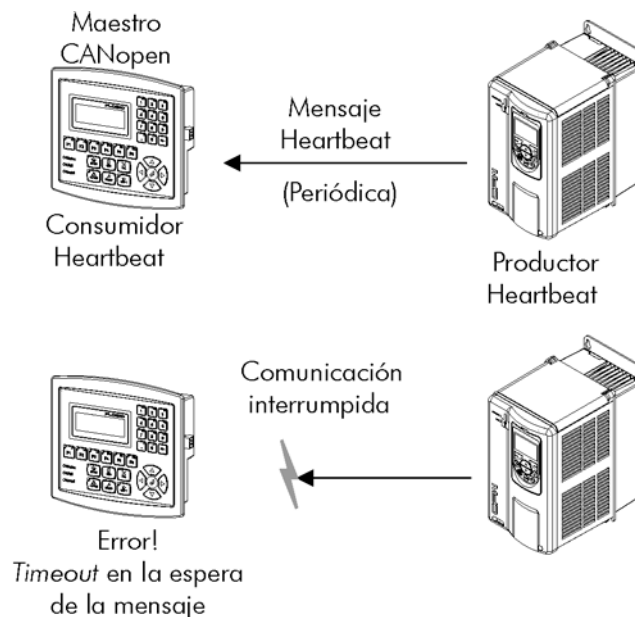


Figura 6.6: Servicio de control de errores – Heartbeat

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 servoconvertidor SCA06 posee los servicios de productor y consumidor *heartbeat*. Como consumidor, es posible programar hasta 4 diferentes productores para que sean monitoreados por el convertidor:

Índice	1016h
Nombre	Consumer Heartbeat Time
Objeto	ARRAY
Tipo	UNSIGNED32

Subíndice	0
Descripción	Número del último subíndice
Acceso	ro
Mapeable	No
Rango	-
Valor Padrón	8

Subíndice	1 – 8
Descripción	Consumer Heartbeat Time 1 – 8
Acceso	rw
Mapeable	No
Rango	UNSIGNED32
Valor Padrón	0

En los subíndices de 1 hasta 8, es posible programar el consumidor escribiendo un valor en el siguiente formato:

UNSIGNED32		
Reservado (8 bits)	Node-ID (8 bits)	Heartbeat time (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 servoconvertidor SCA06 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 servoconvertidor SCA06, 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:

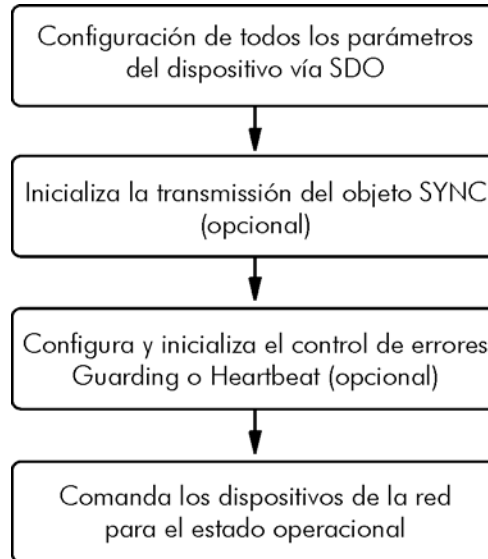


Figura 6.7: Diagrama de flujo del proceso de inicialización

Es necesario observar que los objetos de comunicación del servoconvertidor SCA06 (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.

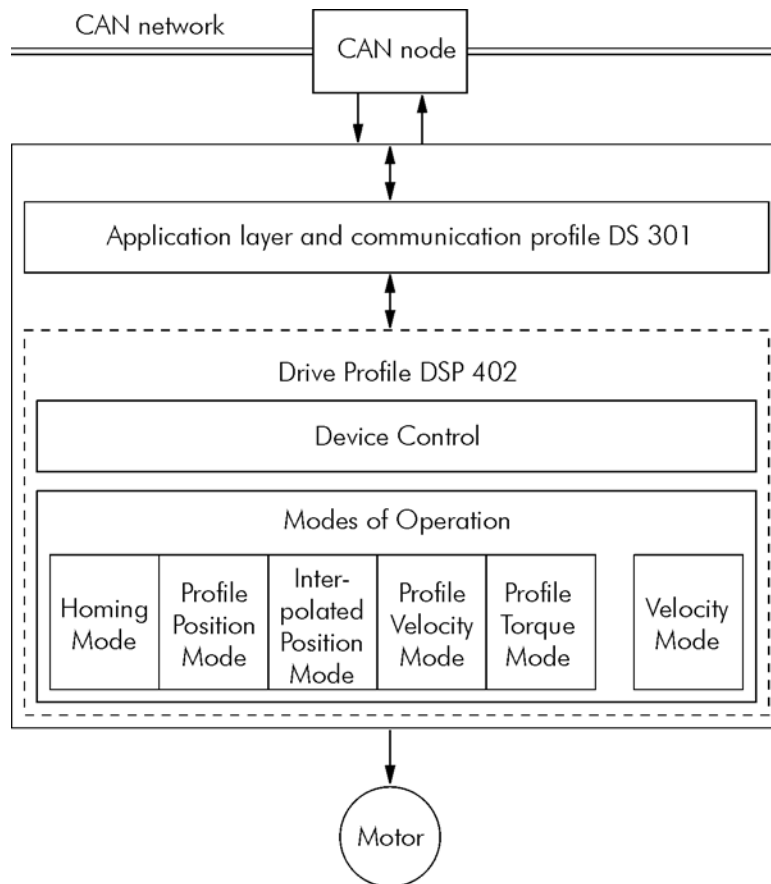


Figura 7.1: Arquitectura de comunicación para un drive en la red CANopen

Para el SCA06, solamente el Velocity Mode es soportado. La siguiente tabla presenta el listado de los objetos disponibles para el SCA06, 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
Control Device					
6040h	VAR	ControlWord	Unsigned16	rw	Sí
6041h	VAR	StatusWord	Unsigned16	ro	Sí
6060h	VAR	Mode of operation	Integer8	rw	Sí
6061h	VAR	Modes of operation display	Integer8	ro	Sí
6502h	VAR	Supported drives modes	Unsigned32	ro	Sí
Factor Group					
608Fh	VAR	Position encoder resolution	Unsigned32	rw	No
6091h	VAR	Gear ration	Unsigned32	rw	No
6092h	VAR	Feed constant	Unsigned32	rw	No
Position Control Function					
6063h	VAR	Position actual value	Integer32	ro	Sí
6064h	VAR	Position actual value in user units	Integer32	ro	Sí
Profile Position Mode					
607Ah	VAR	Target position	Integer32	rw	Sí
6081h	VAR	Profile velocity	Unsigned32	rw	Sí
6083h	VAR	Profile acceleration	Unsigned32	rw	Sí
6084h	VAR	Profile deceleration	Unsigned32	rw	Sí
6086h	VAR	Motion profile type	Integer16	rw	Sí
Profile Velocity Profile					
6069h	VAR	Velocity sensor actual value	Integer32	ro	Sí
606Bh	VAR	Velocity demand value	Integer32	ro	Sí
606Ch	VAR	Velocity actual value	Integer32	ro	Sí
60FFh	VAR	Target velocity	Integer32	rw	Sí
Profile Torque Mode					
6071h	VAR	Target torque	Integer16	rw	Sí
6077h	VAR	Torque actual value	Integer16	ro	Sí
6087h	VAR	Torque slope	Unsigned32	rw	Sí
6088h	VAR	Torque profile type	Integer16	rw	Sí

Siempre que un objeto de este listado fuera leído o escrito, el SCA06 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:

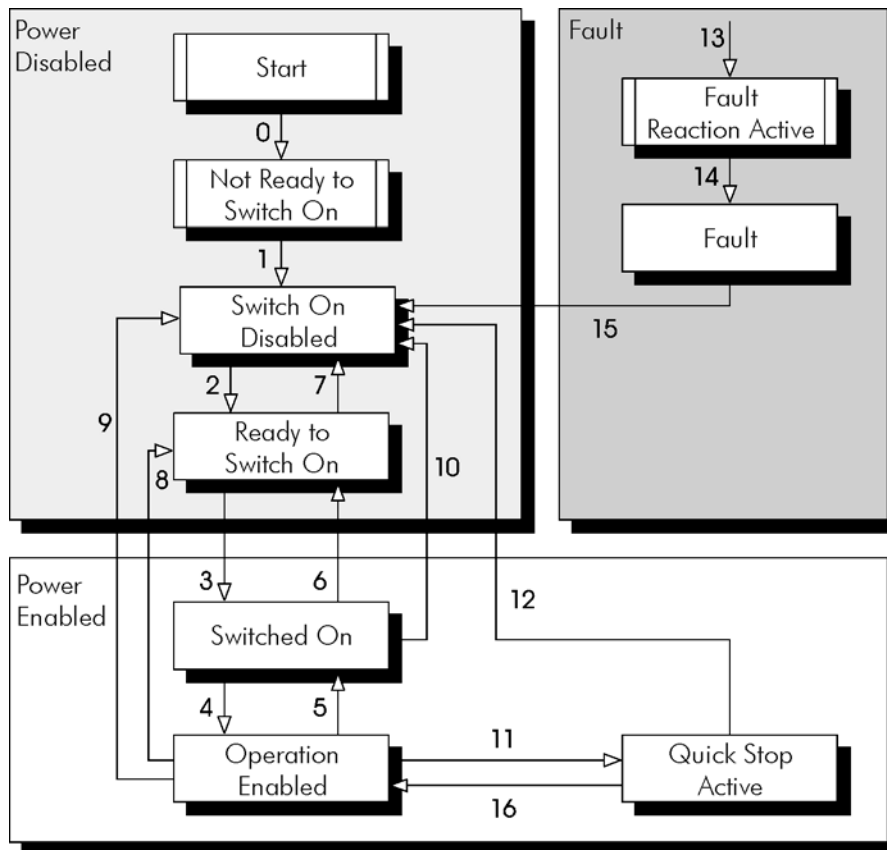


Figura 7.2: Máquina de estado para drives

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 SCA06 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 SCA06.
- **Transición 3:** Comando “*Switch on*” recibido. Es hecha la transición de estados, más ninguna acción es tomada por el SCA06.
- **Transición 4:** Comando “*Enable operation*” recibido. El drive es habilitado.
- **Transición 5:** Comando “*Disable operation*” recibido. El drive es deshabilitado.

- **Transición 6:** Comando “*Shutdown*” recibido. Es hecha la transición de estados, más ninguna acción es tomada por el SCA06.
- **Transición 7:** Comandos “*Quick stop*” y “*Disable voltage*” recibidos. Es hecha la transición de estados, más ninguna acción es tomada por el SCA06.
- **Transición 8:** Comando “*Shutdown*” recibido. Durante la operación del drive este es deshabilitado, bloqueando la alimentación para el motor.
- **Transición 9:** Comando “*Shutdown*” recibido. Durante la operación del drive este es deshabilitado, bloqueando la alimentación para el motor.
- **Transición 10:** Comando “*Quick stop*” o “*Disable voltage*” recibido. Es hecha la transición de estados, más ninguna acción es tomada por el SCA06.
- **Transición 11:** Comando “*Quick stop*” recibido. Convertidor ejecuta la función de parada por rampa.
- **Transición 12:** Comando “*Disable voltage*” recibido. El drive es deshabilitado.
- **Transición 13:** Error es detectado y el drive es deshabilitado.
- **Transición 14:** Después de deshabilitar el drive, él 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.

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.



¡NOTA!

Para que los comandos enviados por la palabra de control sean ejecutados por el servoconvertidor SCA06, es necesario que el drive sea programado para el modo de operación “CANopen”. Esta programación es hecha en el parámetro P0202.

7.1.2 Objeto 6041h – Statusword

Indica el estado actual del SCA06.

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 SCA06.
- **Bit 9 – Remote:** indica cuando el drive está en el modo remoto y acepta comandos vía red CANopen⁷.

⁷ Depende de la programación del convertidor de frecuencia.

- **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 servoconvertidor SCA06.
- **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 SCA06.

Índice	6060h
Nombre	Modes of operation
Objeto	VAR
Tipo	INTEGER8
Parámetros utilizados	-

Acceso	rw
Mapeable	Sí
Rango	INTEGER8
Valor Padrón	-

Los valores aceptables para este objeto se describen en la tabla 7.1. Otros valores están reservados.

Tabla 7.1: Modos de operación para el servoconvertidor SCA06

Valor	Modo de operación
1	Profile Position Mode
3	Profile Velocity Mode
4	Profile Torque Mode

7.1.4 Objeto 6061h – Modes of Operation Display

Indica el modo de operación del SCA06.

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.

7.1.5 Objeto 6502h – Supported Drive Modes

Indica los modos de operación soportados por el drive. Cada bit representa un modo de operación, y el valor 1 en el bit indica que el modo de operación es soportado.

31 – 15	15 – 7	9	8	7	6	5	4	3	2	1	0
Manufacturer specific	reserved	cst	csv	csp	ip	hm	reserved	tq	pv	vl	PP

El servoconvertidor SCA06 presenta 3 modos de operación:

- pp: Profile Position mode.
- pv: Profile Velocity mode.
- tq: Torque mode.

Conociendo los modos soportados en el SCA06, se define el valor 0Dh para este objeto.

Índice	6502h
Nombre	Supported drive modes
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acceso	ro
Mapeable	Sí
Rango	UNSIGNED8
Valor Estándar	0Dh

7.2 FACTOR GROUP – OBJETOS PARA CONVERSIÓN DE UNIDADES

Este grupo de objetos permite hacer la conversión de unidades para objetos que representan valores de posición. Estos valores tendrán su escala y dimensión definida de acuerdo con los valores de notación y dimensión programados, conforme es descrito a seguir:

7.2.1 Objeto 608Fh – Position Encoder Resolution

Este objeto define el incremento del encoder de acuerdo con la rotación del motor.

Position encoder resolution = encoder increments / motor revolutions

Índice	608Fh
Nombre	Position encoder resolution
Objeto	ARRAY
Tipo	UNSIGNED32

Sub-índice	0
Descripción	Número del último sub-índice
Acceso	ro
Mapeable	No
Rango	UNSIGNED8
Valor Estándar	02h

Sub-índice	1
Descripción	Encoder increments
Acceso	rw
Mapeable	No
Rango	UNSIGNED32
Valor Estándar	FFh

Sub-índice	2
Descripción	Motor revolutions
Acceso	rw
Mapeable	No
Rango	UNSIGNED32
Valor Estándar	01h

Valores posibles para el sub-índice 1 (Encoder increments):

Tabla 7.2: Valores para el Sub-índice Encoder Increments

Valor	Encoder increments
41	Grados
42	Minutos
43	Segundos
FF	Unidad interna – 65536 incrementos por vuelta

El sub-índice 2 (Motor revolutions) acepta solamente valor igual a 1.

7.2.2 Objeto 6091h – Gear Ratio

Este objeto indica la configuración del número de rotaciones del eje del motor y el número de rotaciones del eje motriz, o sea, define la relación de transmisión. La relación de transmisión es definida por la siguiente fórmula:

Gear ratio = motor shaft revolutions / driving shaft revolutions

Índice	6091h
Nombre	Gear ratio
Objeto	ARRAY
Tipo	UNSIGNED32

Sub-índice	0
Descripción	Número del último sub-índice
Acceso	Ro
Mapeable	No
Rango	UNSIGNED8
Valor Estándar	02h

Sub-índice	1
Descripción	Motor revolutions
Acceso	rw
Mapeable	No
Rango	UNSIGNED32
Valor Estándar	01h

Sub-índice	2
Descripción	Shaft revolutions
Acceso	rw
Mapeable	No
Rango	UNSIGNED32
Valor Estándar	01h

El único valor posible para el sub-índice 1 y sub-índice 2 es 1.

7.2.3 Objeto 6092h – Feed Constant

Este objeto indica la distancia por 1 vuelta del eje del motor.

Índice	6092h
Nombre	Feed constant
Objeto	ARRAY
Tipo	UNSIGNED32

Sub-índice	0
Descripción	Número del último sub-índice
Acceso	ro
Mapeable	No
Rango	UNSIGNED8
Valor Estándar	02h

Sub-índice	1
Descripción	Feed
Acceso	rw
Mapeable	No
Rango	UNSIGNED32
Valor Estándar	FFh

Sub-índice	2
Descripción	Shaft revolutions
Acceso	rw
Mapeable	No
Rango	UNSIGNED32
Valor Estándar	01h

Valores posibles para el sub-índice 1 (Feed):

Tabla 7.3: Valores para el Sub-índice Feed

Valor	Modo de operación
41	Grados
42	Minutos
43	Segundos
FF	Unidad interna – 65536 incrementos por vuelta

El sub-índice 2 (Shaft revolutions) acepta solamente valor igual a 1.

7.3 POSITION CONTROL FUNCTION – CONTROLADOR DE POSICIÓN

Este grupo de objetos es utilizado para describir el funcionamiento del controlador de posición en malla cerrada.

7.3.1 Objeto 6063h – Position Actual Value

Representa la posición actual del eje del motor en incrementos. Una vuelta completa representa 65536 incrementos.

Índice	6063h
Nombre	Position actual value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acceso	ro
Mapeable	Sí
Rango	INTEGER32
Valor Estándar	-

El valor de este objeto representa siempre la posición de eje solamente en una vuelta . El número de vueltas no es controlado por este objeto.

7.3.2 Objeto 6064h – Position Actual Value in User Units

Representa la posición actual del eje del motor. El valor de este objeto puede ser transformado de unidades internas a valores definidos por el usuario, de acuerdo con lo programado en los objetos 608Fh, 6091h y 6092h, conforme Tabla 7.4.

Índice	6064h
Nombre	Position actual value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acceso	ro
Mapeable	Sí
Rango	INTEGER32
Valor Estándar	-

El valor de este objeto representa siempre la posición de eje solamente en una vuelta. El número de vueltas no es controlado por este objeto.

Tabla 7.4: Programación de los objetos Factor Group

Objeto Unidad	608Fh sub-índice 1	608Fh sub-índice 2	6091h sub-índice 1	6091h sub-índice 2	6092h sub-índice 1	6092h sub-índice 2
Grados	41h	1	1	1	41h	1
Minutos	42h	1	1	1	42h	1
Segundos	43h	1	1	1	43h	1
Unidad interna	FFh	1	1	1	FFh	1

7.4 PROFILE POSITION MODE – OBJETOS PARA CONTROL DEL DRIVE

Este modo de operación permite el control del servoconvertidor SCA06 a través del ajuste de set-point de posición, que pueden ser ejecutados siguiendo dos métodos:

- single set-point.
- set of set-points.

Independientemente del método utilizado, deben ser configurados los siguientes objetos:

- 0x6081 – Profile Velocity.
- 0x6083 – Profile Acceleration.
- 0x6084 – Profile Deceleration.
- 0x6086 – Motion Profile Type.
- 0x607A – Target Position.

Luego del ajuste de la velocidad, de la aceleración, y del set-point, se debe realizar el siguiente procedimiento:

- Habilitar el drive escribiendo 15 en el objeto 0x6041 - ControlWord;
- Escribir en los bits 9, 8, 6, 5 y 4 del objeto 0x6041 – ControlWord, conforme la Tabla 7.9 y Tabla 7.10;

El estado de la ejecución del posicionamiento puede ser verificado en el objeto 0x6040 – StatusWord conforme la Tabla 7.11. El bit SET-POINT ACKNOWLEDGE en el objeto de status (StatusWord – 6040h) será ajustado indicando que un nuevo set-point fue recibido. Si el set-point es aceptado, el bit será ajustado. Cuando el set-point sea alcanzado, el bit TRAGET REACHED, en el objeto de status, será ajustado. La figura 7.3 ilustra un ejemplo de escritura de set-point.

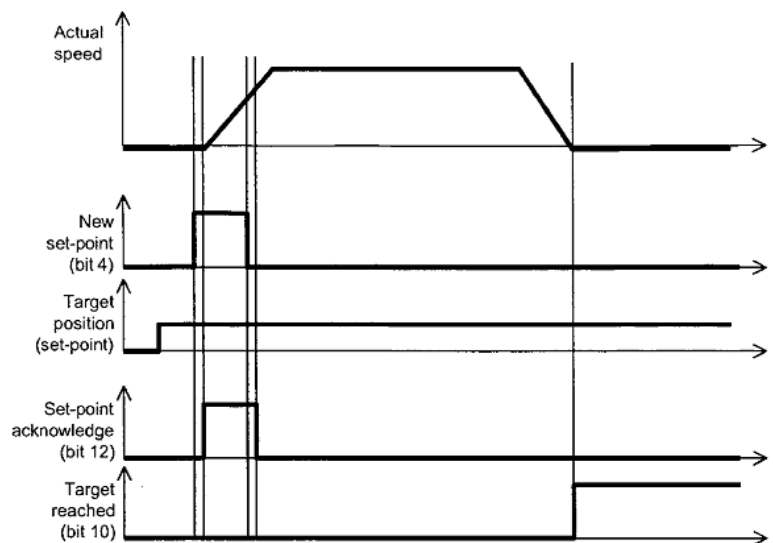


Figura 7.3: Ajuste de set-point de posición (Fuente: IEC 61800-7-201)

Single set-point

El método set-point único es utilizado cuando se desea ejecutar un nuevo set-point inmediatamente. La figura 7.4 ilustra el funcionamiento del método.

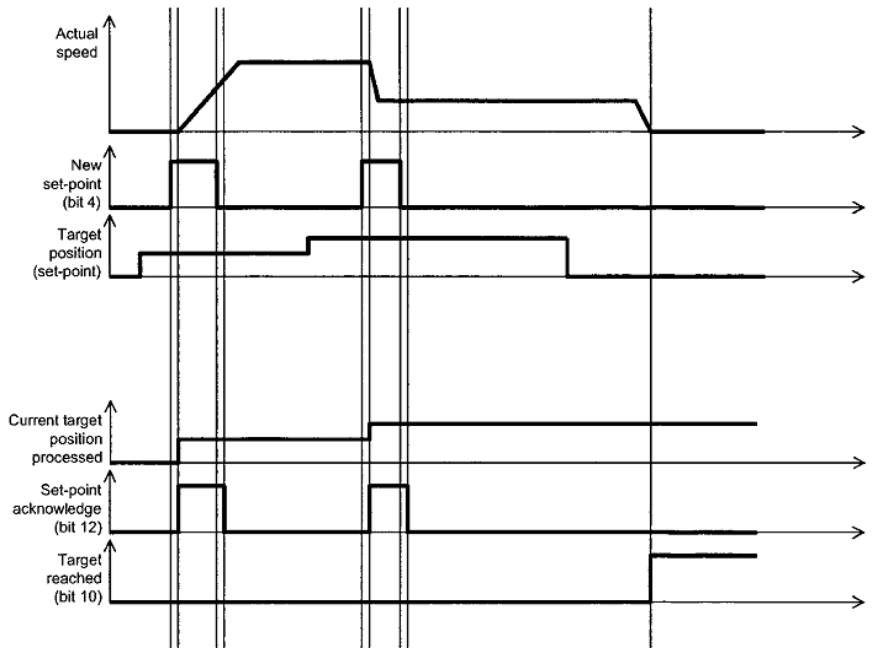


Figura 7.4: Método single set-point (Fuente: IEC 61800-7-201)

Set of set-point

El método conjunto de set-point es utilizado cuando se desea ejecutar un nuevo set-point, solamente tras la finalización del anterior. La figura 7.5 ilustra el funcionamiento del método.

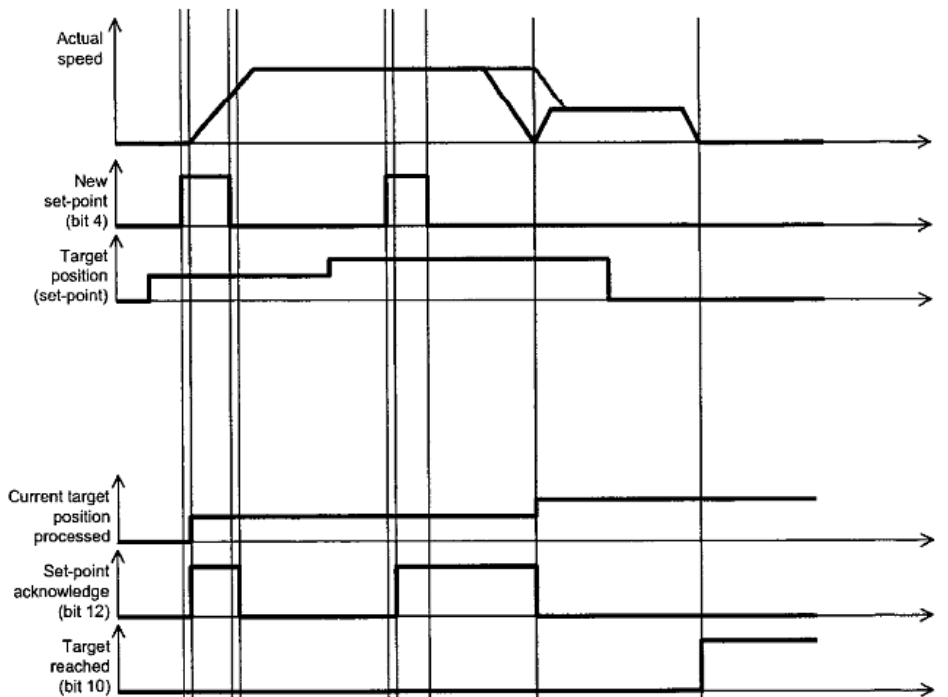


Figura 7.5: Método set of set-point (Fuente: IEC 61800-7-201)

El servoconvertidor SCA06 puede almacenar dos set-points, el que está en ejecución y el que será ejecutado, como ilustra la figura 7.6.

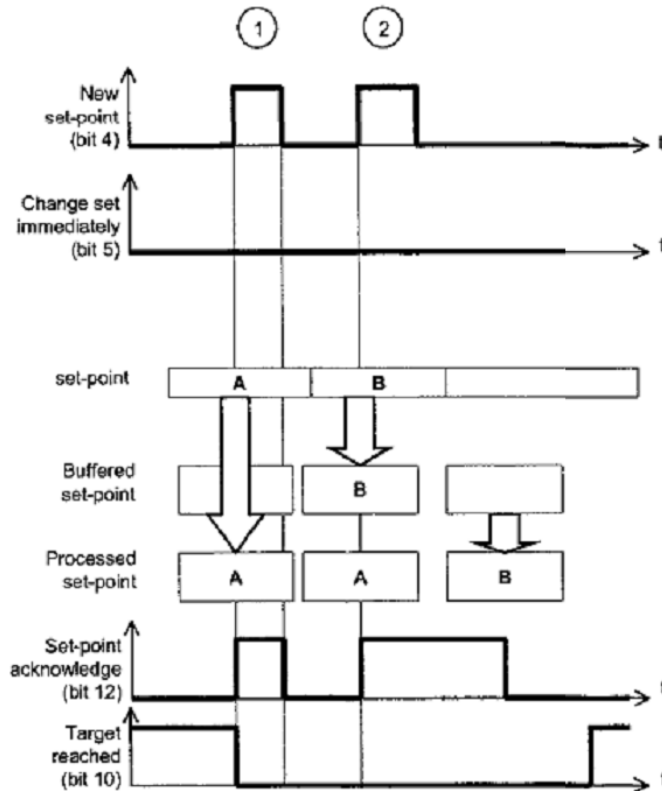


Figura 7.6: Almacenamiento de set-point (Fuente: IEC 61800-7-201)

7.4.1 Bits de Control y Estado

El profile mode position utiliza algunos bits de los objetos ControlWord y StatusWord para controlar y monitorear su funcionamiento. Para el objeto ControlWord son utilizados los siguientes bits:

- Bit 4 – New set-point.
- Bit 5 – Change set immediatel.
- Bit 6 – absolute/relative.
- Bit 8 – Halt (no implementado en el SCA06).
- Bit 9 – Change on set-point.

Las tabla 7.5 y tabla 7.6 informan la definición de los bits de control.

Tabla 7.5: Modo Posicionamiento – definición de los bits 4, 5 y 9

Bit 9	Bit 5	Bit 4	Definición
0	0	0 → 1	Posición debe ser concluida antes de que la próxima inicie.
X	1	0 → 1	Próxima posición debe ser iniciada inmediatamente.
1	0	0 → 1	Opción no implementada en el SCA06.

Tabla 7.6: Modo Posicionamiento – definición de los bits 6 y 8

Bit	Valor	Definición
6	0	La referencia de posición debe ser un valor absoluto.
	1	La referencia de posición debe ser un valor relativo.
8	0	El posicionamiento debe ser ejecutado o continuado.
	1	El eje debe ser parado conforme objeto 605Dh.

Para el objeto StatusWord son utilizados los siguientes bits:

- Bit 10 – Target reached.
- Bit 12 – Set-point acknowledge.
- Bit 13 – Following error.

La Tabla 7.7 informa la definición de los bits de status.

Tabla 7.7: Modo Posicionamiento – definición de los bits 10,12 y13

Bit	Valor	Definición
10	0	Referencia de posición no alcanzada.
	1	Referencia de posición alcanzada.
12	0	Referencia de posición anterior ya procesada, aguardando nueva referencia de posición.
	1	Referencia de posición anterior en procesamiento, sustitución de referencia de posición será aceptada.
13	0	Sin error de Following
	1	Error de Following

7.4.2 Objeto 607Ah – Target Position

Permite programar la referencia de posición para el servoconvertidor SCA06 en el modo posicionamiento. Los 16 bits más significativos informan el número de vuelta y los 16 bits menos significativos informan la fracción de vuelta. La escala utilizada en este objeto es 65536 para número de vueltas y 65536 incrementos para una vuelta del eje. El valor de este objeto debe ser interpretado como absoluto o relativo, conforme el estado del Bit 6 del objeto ControlWord.

Número de vuelta 16MSB	Fracción de vuelta 16LSB
---------------------------	--------------------------------

Índice	607Ah
Nombre	Target position
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	INTEGER32
Valor Estándar	0000 0000h

7.4.3 Objeto 6081h – Profile Velocity

Permite programar la velocidad normalmente alcanzada al final de la rampa de aceleración durante un perfil de movimiento. El valor a ser programado en este objeto debe estar entre 0 y 9999 rpm.

Índice	6081h
Nombre	Profile Velocity
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acceso	Rw
Mapeable	Sí
Rango	UNSIGNED32
Valor Estándar	0000 0000h

7.4.4 Objeto 6083h – Profile Acceleration

Permite programar la rampa de aceleración hasta que el eje del motor alcance la velocidad programada. La escala utilizada es la escala ms/krpm y los valores deben estar entre 1 y 32767

Índice	6083h
Nombre	Profile Acceleration
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	UNSIGNED32
Valor Estándar	0000 0001h

7.4.5 Objeto 6084h – Profile Deceleration

Permite programar la rampa de desaceleración hasta que el eje del motor alcance la velocidad cero. La escala utilizada en este objeto es la misma del objeto 6083h.

Índice	6084h
Nombre	Profile deceleration
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	UNSIGNED32
Valor Estándar	0000 0001h

7.4.6 Objeto 6086h – Motion Profile Type

Permite programar el perfil de la rampa de aceleración y desaceleración para el drive.

Índice	6086h
Nombre	Motion profile type
Objeto	VAR
Tipo	INTERGER32

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	INTERGER16
Valor Estándar	FFFFh

Valores posibles para este objeto:

Tabla 7.8: Valores para el Sub-índice Motion Profile Type

Valor	Definición
0000h	Rampa linear
FFFFh	Sin rampa

7.5 PROFILE VELOCITY MODE – OBJETOS PARA CONTROL DEL DRIVE

Este modo de operación permite el control del drive de forma simple, poniendo a disposición funciones del tipo:

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

Estas funciones son ejecutadas tomando como base un conjunto de objetos para configuración de este modo de operación.

7.5.1 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:

Tabla 7.9: Modo Velocidad – definición de los bits 4, 5, 6 y 8

Bit	Nombre	Valor	Descripción
4			Reservado
5			Reservado
6			Reservado
8	Halt	0	Ejecuta movimiento
		1	Para el eje

Para el objeto StatusWord son utilizados los siguientes bits:

- Bit 10 – Target reached.
- Bit 12 – Speed.
- Bit 13 – Max slippage error (no implementado).

Tabla 7.10: Modo Velocidad – definición de los bits 10,12 y13

Bit	Valor	Definición
10	0	Halt = 0 - referencia de velocidad no alcanzada. Halt = 1 - velocidad diferente de 0 (cero)
	1	Halt = 0 - referencia de velocidad alcanzada. Halt = 1 - velocidad igual a 0 (cero).
12	0	Velocidad diferente de 0(cero).
	1	Velocidad igual a 0 (cero).
13	0	No implementado
	1	

7.5.2 Objeto 6069h – Velocity Sensor Actual Value

Permite la lectura del sensor utilizado para medir la velocidad del motor. El servoconvertidor SCA06 utiliza un resolver como posición (la velocidad angular es obtenida derivando este valor en el tiempo), luego el sensor provee un valor proporcional a la posición angular. El sensor posee resolución de 14 bits, y una vuelta completa provee 16384 valores diferentes de posición.

Índice	6069h
Nombre	Velocity sensor actual valor
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acceso	RO
Mapeable	Sí
Rango	INTERGER32
Valor Estándar	-

7.5.3 Objeto 606Bh – Velocity Demand Value

Indica la velocidad suministrada por el generador de trayectoria del servoconvertidor, utilizada por el regulador de velocidad para control del motor. El valor suministrado por este objeto es dado en la escala interna del SCA06 donde, 0x7FFF FFFF → 10.000 rpm.

Índice	606Bh
Nombre	Velocity demand value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acceso	Ro
Mapeable	Sí
Rango	INTERGER32
Valor Estándar	-
Valor mínimo	0x8000 0001
Valor máximo	0x7FFF FFFF

7.5.4 Objeto 606Ch – Velocity Actual Value

Indica la velocidad del motor. El valor suministrado por este objeto es dado en la escala interna del SCA06 donde, 0x7FFF FFFF → 10.000 rpm.

Índice	606Ch
Nombre	Velocity actual value
Objeto	VAR
Tipo	INTEGER32

Sub-índice	0
Acceso	ro
Mapeable	Sí
Rango	INTERGER32
Valor Estándar	-

7.5.5 Objeto 60FFh – Target Velocity

Permite programar la referencia de velocidad para el servoconvertidor SCA06 en el modo velocidad. El valor a ser programado en este objeto debe respetar la escala interna del SCA06 donde 0x7FFF FFFF → 10.000 rpm y 0x8000 0000 → -10.000 rpm

Índice	60FFh
Nombre	Target Velocity
Objeto	VAR
Tipo	INTERGER32

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	INTERGER32
Valor Estándar	0000 0000h

7.6 PROFILE TORQUE MODE – OBJETOS PARA CONTROL DEL DRIVE

Este modo posibilita el control del drive vía una referencia de torque recibida por la red CANopen.

Estas funciones son ejecutadas tomando como base un conjunto de objetos para configuración de este modo de operación.

7.6.1 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:

Tabla 7.11: Modo Torque – definición de los bits 4, 5, 6 y 8

Bit	Nombre	Valor	Descripción
4			Reservado
5			Reservado
6			Reservado
8	Halt	0	Ejecuta movimiento
		1	Para el eje

Para el objeto StatusWord son utilizados los siguientes bits:

- Bit 10 – Target reached.
- Bit 12 – Reservado.
- Bit 13 – Reservado.

Tabla 7.12: Modo Torque – definición de los bits 10,12 y13

Bit	Valor	Definición
10	0	Referencia de torque no alcanzada.
	1	Referencia de torque alcanzada.
12	0	Reservado
	1	
13	0	Reservado
	1	

7.6.2 Objeto 6071h – Target Torque

Permite programar la referencia de torque para el servoconvertidor SCA06 en el modo torque. La escala utilizada para escritura en este objeto es suministrada en partes por mil del torque nominal del motor.

Índice	6071h
Nombre	Target Torque
Objeto	VAR
Tipo	INTERGER16

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	INTERGER16
Valor Estándar	0000h

7.6.3 Objeto 6077h – Torque Actual Value

Indica el torque actual del motor. El valor es suministrado en parte por mil del torque nominal del motor.

Índice	6077h
Nombre	Torque actual value
Objeto	VAR
Tipo	INTERGER16

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	INTERGER16
Valor Estándar	0

7.6.4 Objeto 6087h – Torque Slope

Permite programar la tasa de variación del torque en el tiempo (rampa de torque) para el servoconvertidor SCA06. La escala utilizada es de partes por mil del torque nominal del motor por segundo.

Índice	6087h
Nombre	Torque slope
Objeto	VAR
Tipo	UNSIGNED32

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	UNSIGNED32
Valor Estándar	0

7.6.5 Objeto 6088h – Torque Profile Type

Indica el torque actual del motor. El valor es suministrado en parte por mil del torque nominal del motor.

Índice	6077h
Nombre	Torque Profile type
Objeto	VAR
Tipo	INTERGER16

Sub-índice	0
Acceso	rw
Mapeable	Sí
Rango	INTERGER16
Valor Estándar	FFFFh

Valores posibles para este objeto:

Tabla 7.13: Valores para el Sub-índice Torque Profile Type

Valor	Definición
0000h	Rampa linear de torque
FFFFh	Sin rampa

8 OPERACIÓN EN LA RED CANOPEN – MODO MAESTRO

Además de la operación como esclavo, el servoconvertidor SCA06 también permite la operación como maestro de la red CANopen. A seguir serán descritas características y funciones del SCA06 como maestro de la red CANopen.

8.1 HABILITACIÓN DE LA FUNCIÓN CANOPEN MAESTRO

Como estándar, el servoconvertidor SCA06 está programado para operar como esclavo de la red CANopen. La programación del equipamiento como maestro de la red debe ser hecha utilizando el software WSCAN, que permite también la configuración de toda la red CANopen. La descripción detallada de las ventanas y funciones del software WSCAN debe ser obtenida en el menú “Ayuda” del propio software.

Luego de ser elaborada la configuración del maestro, es necesario hacer el download⁸ de las configuraciones, utilizando una de las interfaces de programación del producto – consulte el manual del usuario para mayores informaciones. Una vez programado como maestro de la red, en caso de que sea necesario borrar estas configuraciones, la función para borrar el programa del usuario – a través del P00204 – también borra las configuraciones del maestro CANopen.



¡NOTA!

La red CANopen es una red flexible que permite varias formas de configuración y operación. No obstante, esta flexibilidad exige que el usuario tenga un buen conocimiento de las funciones y objetos de comunicación utilizados para configuración de la red, así como el conocimiento del software de programación WSCAN.

8.2 CARACTERÍSTICAS DEL MAESTRO CANOPEN

El servoconvertidor SCA06 permite controlar un conjunto de hasta 8 esclavos, utilizando los siguientes servicios y recursos de comunicación:

- Servicio de gestor de la red (NMT).
- 8 PDOs de transmisión.
- 8 PDOs de recepción.
- 8 Consumidores Heartbeat.
- Productor Heartbeat.
- Cliente SDO.
- Productor/consumidor SYNC.
- Mapeado en los PDOs realizado utilizando parámetros del usuario.

Las características físicas – instalación, conector, cable, etc. – son las mismas, tanto para el SCA06 operando como maestro como para cuando opera como esclavo. Las configuraciones de dirección y tasa de comunicación también son necesarias para la operación como maestro, pero estas configuraciones son programadas por el software WSCAN de acuerdo con las propiedades definidas para el maestro en el propio software.

8.3 OPERACIÓN DEL MAESTRO

Una vez programado para operar como maestro, el servoconvertidor SCA06 ejecutará las siguientes etapas para realizar la inicialización, en secuencia, para cada uno de los esclavos:

- 1^a: enviado el comando de reset de la comunicación para toda la red, para que los esclavos inicien con valores conocidos para los objetos de comunicación.
- 2^a: Identificación del equipamiento en la red, a través de la lectura vía SDO del objeto 1000h/00h – Object Identification.
- 3^a: Escritura vía SDO de todos los objetos programados para el esclavo, que usualmente incluye la configuración y mapeo de los TPDOs y RPDOs, node guarding, heartbeat, además de los objetos específicos del fabricante, en caso de que sean programados.

⁸ Durante el download de las configuraciones, la comunicación CANopen será deshabilitada, siendo reiniciada al término de la operación.

- 4ª: Iniciado servicio de control de errores – node guarding o heartbeat – en caso de que sean programados.
- 5ª: envío del esclavo para modo operacional.

Si una de estas etapas falla, será indicado error de comunicación con el esclavo. Dependiendo de las configuraciones, la inicialización del esclavo será abortada, y el maestro hará la inicialización del esclavo siguiente, retornando al esclavo con error tras intentar inicializar todos los demás esclavos de la red.

De forma semejante, si, durante la operación de un esclavo, es identificado error en el servicio de control de errores, dependiendo de las configuraciones hechas para el maestro, el esclavo será automáticamente reseteado y el procedimiento de inicialización será ejecutado nuevamente.



¡NOTA!

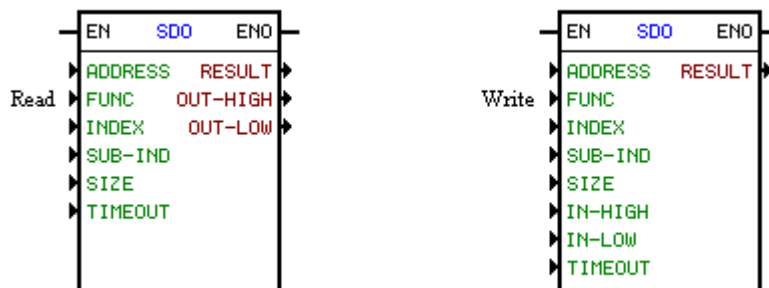
El estado de la comunicación y el estado de cada esclavo pueden ser observados en marcadores de sistema.

8.4 BLOQUES PARA EL MAESTRO CANOPEN

Además de los objetos de comunicación y de las configuraciones hechas en el software WSCAN, también están disponibles bloques para monitoreo y envío de comandos, que pueden ser utilizados durante la elaboración del aplicativo en ladder para el servoconvertidor SCA06. No es necesario utilizar estos bloques durante la operación del equipamiento, pero su uso otorga mayor flexibilidad y facilita el diagnóstico de problemas de comunicación durante la operación del servoconvertidor SCA06.

8.4.1 CANopen SDO – Lectura/Escritura de Datos vía SDO

Bloque para lectura o escritura de datos vía SDO de un esclavo remoto. Permite realizar la lectura o escritura de objetos en la red con tamaño de hasta 4 bytes.



DESCRIPCIÓN:

Está compuesto por 1 entrada EN, 1 salida ENO y 9 argumentos, siendo los mismos:

- ADDRESS : Dirección del nudo de la red CANopen
- FUNC : Función (lectura o escritura)
- INDEX : Índice del objeto que se desea leer o escribir (decimal)
- SUB-IND : Sub-índice del objeto que se desea leer o escribir (decimal)
- SIZE : Tamaño del objeto que se desea leer o escribir (bytes)
- TIMEOUT : Tiempo en ms de espera para lectura o escritura del valor
- RESULT : Resultado de la ejecución del bloque
 - 0 = ejecutado con éxito
 - 1 = la tarjeta no puede ejecutar la función (ejemplo: maestro no habilitado)
 - 2 = timeout en la espera de respuesta del maestro
 - 3 = esclavo retornó error
- OUT-HIGH : Valor más significativo del objeto leído (word)
- OUT-LOW : Valor menos significativo del objeto leído (word)
- IN-HIGH : Valor más significativo a ser escrito en el objeto (word)
- IN-LOW : Valor menos significativo a ser escrito en el objeto (word)

La entrada EN es responsable por la habilitación del bloque.
La salida ENO queda en 1 tras ejecutar el bloque

FUNCIONAMIENTO:

Si la entrada EN es cero, el bloque no es ejecutado.

Si la entrada EN sufre una transición de 0 para 1, la tarjeta envía un mensaje vía red CANopen hacia un esclavo de la red, de acuerdo con los argumentos programados. Si el bloque está programado para lectura, la tarjeta hará la requisición para el esclavo, y el valor informado por el esclavo será guardado en los argumentos de salida. Si el bloque está programado para escritura, los argumentos de entrada serán escritos en el objeto correspondiente del esclavo. Tras la ejecución del bloque, la salida ENO queda en 1 y sólo retorna a cero luego que la entrada EN quede en cero.

9 MARCADORES DE SISTEMA PARA CAN/CANOPEN

Para interfaz CAN y comunicación CANopen, fueron puestos a disposición los siguientes marcadores de lectura (%RS) y marcadores de escritura (%WC), para control y monitoreo de esta interfaz:

9.1 WORDS DE LECTURA DE ESTADO

Estado del Maestro y Esclavos CANopen: conjunto de marcadores de lectura para indicar informaciones sobre el estado general del maestro CANopen y el estado de la comunicación entre el maestro y cada uno de los esclavos.	
Marcador	Descripción
%RS4000	Estado do maestro CANopen: Bit 0: todos los esclavos fueron contactados. Bit 1: download de las configuraciones de los esclavos realizada. Bit 2: control de errores de los esclavos iniciado. Bit 3: fin de la inicialización de los esclavos. Bit 4: detectado error en la inicialización de por lo menos un esclavo. Bit 5: detectado error en el servicio de control de erros de por lo menos un esclavo. Bits 6 y 7: reservado Bit 8: asume el valor del toggle bit (ver %CD3200) luego de que el maestro envíe comando NMT. Bits 9 ... 12: reservado Bit 13: interfaz CAN en estado de bus off. Bit 14: sin alimentación en la interfaz CAN. Bit 15: comunicación deshabilitada.
%RS4001 ... %RS4127	Estado de los esclavos CANopen. Son 127 marcadores de Word, donde cada marcador está asociado a una dirección en la red CANopen, e indica el estado del esclavo en la dirección: Bit 0: maestro contactó esclavo con éxito. Bit 1: download de las configuraciones del esclavo realizada con éxito. Bit 2: control de errores del esclavo iniciado. Bit 3: fin de la inicialización del esclavo. Bit 4: detectado error en la inicialización del esclavo. Bit 5: detectado error en el servicio de control de errores del esclavo. Bits 6 ... 15: reservado

Último Error en el Cliente SDO: conjunto de marcadores de lectura para informar datos sobre errores en el cliente SDO. En caso de que alguna requisición sea hecha al cliente SDO y el esclavo no responda, o responda con error, los datos relativos al último error detectado por el cliente SDO son guardados en estos marcadores.	
Marcador	Descripción
%RS4128	Dirección del esclavo destino, para la cual la requisición SDO fue enviada.
%RS4129	Índice del objeto acceso vía SDO.
%RS4130	Sub-índice del objeto accedido.
%RS4131	Tipo de acceso realizado: 1 = lectura, 2 = escritura.
%RS4132 ... %RS4133	Para accesos de escritura, indica el valor escrito.
%RS4134 ... %RS4135	Indica el código del error recibido, de acuerdo con los errores de comunicación vía SDO de la especificación del protocolo CANopen.

Último EMCY detectado: conjunto de marcadores de lectura para informar datos sobre errores reportados por productores EMCY. El maestro CANopen no posee consumidor EMCY. Telegramas EMCY enviados por esclavos de la red, no obstante, son capturados por el maestro, y las informaciones del último EMCY detectado son guardadas en estos marcadores.	
Marcador	Descripción
%RS4136	Dirección del esclavo que reportó el EMCY.
%RS4137... % RS4140	Ocho bytes de datos del telegrama EMCY, con informaciones sobre el código de error reportado por el esclavo.

9.2 WORDS DE ESCRITURA DE COMANDO

Control del Maestro CANopen: conjunto de marcadores de escritura para controlar el maestro CANopen.	
Marcador	Descripción
%WC4142	Comando para control del maestro CANopen y envío de telegrama NMT. Bits 0 ... 7: código del comando NMT: 1 = START 2 = STOP 128 = ENTER PRE-OPERATIONAL 129 = RESET NODE 130 = RESET COMMUNICATION Bit 8: toggle bit, siempre que el valor de este bit sea alterado enviará el comando programado. Bits 9 ... 14: reservado Bit 15: deshabilita comunicación CANopen
%WC4143	Bits 16 ... 23: dirección del esclavo destino para envío del comando NMT.

10 FALLAS Y ALARMYS RELACIONADAS CON LA COMUNICACIÓN CANOPEN

A133/F33 – 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 F33; dependiendo de la programación hecha en el parámetro P0662. 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/F34 – 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 alto⁹, 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 F34, dependiendo de la programación hecha en el parámetro P0662. 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/F35 – 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

⁹ 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 señalado a través del HMI el mensaje de alarma A135 – o falla F35, dependiendo de la programación hecha en el parámetro P0662. 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 Equipamentos Elétricos S.A.
Jaraguá do Sul – SC – Brasil
Fone 55 (47) 3276-4000 – Fax 55 (47) 3276-4020
São Paulo – SP – Brasil
Fone 55 (11) 5053-2300 – Fax 55 (11) 5052-4212
automacao@weg.net
www.weg.net