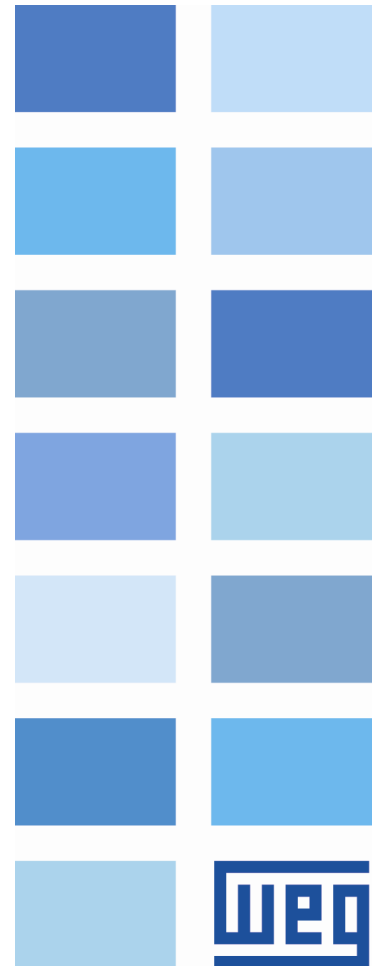


DeviceNet

SCA06

Manual del Usuario





Manual del Usuario de DeviceNet

Serie: SCA06

Idioma: Español

Nº del Documento: 10003341887 / 00

Build 345*

Fecha de la Publicación: 03/2015

Índice

A Respecto del Manual	5
Abreviaciones y Definiciones	5
Representación Numérica	5
Documentos	5
1 Introducción a la Comunicación DeviceNet	6
1.1 CAN	6
1.1.1 Frame de Datos	6
1.1.2 Frame Remoto	6
1.1.3 Acceso a la Red	6
1.1.4 Control de Errores	7
1.1.5 CAN y DeviceNet	7
1.2 Características de la Red DeviceNet	7
1.3 Medio Físico	7
1.3.1 Capada de Enlace de Datos	8
1.3.2 Capada de Transporte y Red	9
1.3.3 Capada de Aplicación – Protocolo CIP	9
1.3.4 Archivo de Configuración EDS	10
1.3.5 Modos de Comunicación	10
1.3.6 Conjunto de Conexiones Predefinidas Maestro/Esclavo	11
2 Interfaz de Comunicación DeviceNet	12
2.1 Características de la interfaz CAN	12
2.2 Terminales del Conector	12
2.3 Fuente de Alimentación	13
2.4 Señalizaciones	13
3 Instalación en Red DeviceNet	14
3.1 Tasa de Comunicación	14
3.2 Dirección en la Red DeviceNet	14
3.3 Resistor de Terminación	14
3.4 Cable	14
3.5 Conexión con la Red	15
4 Parametrización	16
4.1 Símbolos para Descripción de las Propiedades	16
P0070 - Estado del Controlador CAN	16
P0071 - Contador de Telegramas CAN Recibidos	16
P0072 - Cotador de Telegrams CAN transmitidos	16
P0073 – Contador de Errores de Bus Off.....	17
P0074 - Contador de Mensajes CAN Perdidas	17
P0077 - Estado de la Red DeviceNet	17
P0078 - Estado del Maestro Devicenet.....	18
P0202 - Modo de Operación.....	18
P0662 – Acción para Error de Comunicación	19
P0700 - Protocolo CAN	19
P0701 - Dirección CAN	20
P0702 - Tasa de Comunicación CAN	20

P0703 - Reset de Bus Off	20
P0710 – Instancias de I/O DeviceNet	21
P0711 – Lectura #3 DeviceNet	25
P0712 – Lectura #4 DeviceNet	25
P0713 – Lectura #5 DeviceNet	25
P0714 – Lectura #6 DeviceNet	25
P0715 – Lectura #7 DeviceNet	25
P0716 – Lectura #8 DeviceNet	25
P0717 – Lectura #9 DeviceNet	25
P0718 – Lectura #10 DeviceNet	25
P0719 – Lectura #11 DeviceNet	25
P0720 – Lectura #12 DeviceNet	25
P0721 – Lectura #13 DeviceNet	25
P0722 – Lectura #14 DeviceNet	25
P0723 – Escrita #3 DeviceNet	26
P0724 – Escrita #4 DeviceNet	26
P0725 – Escrita #5 DeviceNet	26
P0726 – Escrita #6 DeviceNet	26
P0727 – Escrita #7 DeviceNet	26
P0728 – Escrita #8 DeviceNet	26
P0729 – Escrita #9 DeviceNet	26
P0730 – Escrita #10 DeviceNet	26
P0731 – Escrita #11 DeviceNet	26
P0732 – Escrita #12 DeviceNet	26
P0733 – Escrita #13 DeviceNet	26
P0734 – Escrita #14 DeviceNet	26
P0735 - Palabra de Control	27
P0736 - Estado Lógico	27
5 Clases de Objetos Soportadas	29
5.1 Clase Identity (01H)	29
5.2 Clase Message Router (02H)	29
5.3 Clase DeviceNet (03H)	29
5.4 Clase Assembly (04H)	30
5.5 Clase Connection (05H)	30
5.5.1 Instancia 1: Explicit Messag	30
5.5.2 Instancia 2: Polled	31
5.5.3 Instancia 4: Change of State/Cyclic	31
5.6 Clase Motor Data (28H)	31
5.7 Clase Motor Data (29H)	32
5.8 Clase AC/DC Drive (2AH)	32
5.9 Clase Message Router (2BH)	33
5.10 Clases Específicas del Fabricante	33
6 Fallos y Alarmas	35
A133/F233 - Sin Alimentación en el Interfaz CAN	35
A134/F234 - Bus Off	35
A136/F236 - Maestro en Idle	35
A137/F237 - Timeout en la Conexión DeviceNet	36

A RESPECTO DEL MANUAL

Este manual provee la descripción necesaria para la operación del servoconvertidor SCA06 utilizando el protocolo DeviceNet. Este manual debe ser utilizado en conjunto con el manual del usuario y manual de programación del SCA06.

ABREVIACIONES Y DEFINICIONES

ASCII	American Standard Code for Information Interchange
CRC	Cycling Redundancy Check
CiA	CAN in Automation
CIP	Common Industrial Protocol
PLC	Programmable Logic Controller
HMI	Human-Machine Interface
ODVA	Open DeviceNet Vendor Association
ro	Read only (solamente de 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. Números binarios son representados con la letra 'b' luego del número.

DOCUMENTOS

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

Documento	Versión	Fuente
CAN Specification	2.0	CiA
Volume One - Common Industrial Protocol (CIP) Specification	3.2	ODVA
Volume Three - DeviceNet Adaptation of CIP	1.4	ODVA

Para obtener esta documentación, de debe consultar la ODVA, que actualmente es la organización que mantiene, promociona y actualiza las informaciones relativas a la red DeviceNet.

1 INTRODUCCIÓN A LA COMUNICACIÓN DEVICENET

Para la operación en red DeviceNet, es necesario conocer el modo como la comunicación es hecha. Para es, este ítem trae una descripción general del funcionamiento del protocolo DeviceNet, conteniendo las funciones utilizadas por el SCA06. Para una descripción detallada, consulte la especificación del protocolo.

1.1 CAN

La red DeviceNet 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. El protocolo DeviceNet no utiliza este tipo de frame.

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.

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

²La 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.

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, él 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.
- **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 DeviceNet

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 DeviceNet 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 DEVICENET

Presentado en 1994, DeviceNet es una implementación del protocolo Common Industrial Protocol (CIP) para redes de comunicación industrial. Desarrollado originalmente por la Allen-Bradley, donde ha tenido su tecnología transferida para la ODVA que, desde entonces, mantiene, divulga y promueve el DeviceNet y otras redes basadas en el protocolo CIP³. Además de eso utiliza el protocolo Controller Area Network (CAN) para enlace de datos y acceso al medio, capas 2 y 1 del modelo OSI/ISO, respectivamente.

Utilizado principalmente en la interconexión de controladores industriales y dispositivos de entrada/salida (I/O), el protocolo sigue el modelo productor-consumidor, soporta múltiples modos de comunicación y posee prioridad entre mensajes.

Es un sistema que puede ser configurado para operar tanto en una arquitectura maestro-esclavo cuanto en una arquitectura distribuida punto a punto. Además de eso, define dos tipos de mensajes, I/O (datos de proceso) y explicit (configuración y parametrización). Posee también mecanismos de detección de dirección dobles y aislamiento de los nudos en caso de fallas críticas.

Una red DeviceNet puede contener hasta 64 dispositivos, con dirección de 0 a 63. Cualquier uno de estos puede ser utilizado. No hay cualquier restricción, sin embargo se debe evitar el 63, pues esta dirección es común de ser utilizada para fines de pruebas (comisionamiento / puesta en marcha).

1.3 MEDIO FÍSICO

DeviceNet usa una topología de red del tipo tronco/derivaciones que permite que tanto el cableado de la señal cuanto de alimentación se encuentre presentes en el mismo cable. Esta alimentación, suministrada por una fuente

³CIP representa, en la realidad, una familia de redes. DeviceNet, EtherNet/IP y ControlNet utilizan CIP en la camada de aplicación. La diferencia entre ellos está primordialmente en las capas de enlace de datos y física.

conectada directamente en la red, alimenta los transceivers CAN de los nudos, y posee las siguientes características:

- 24 Vdc;
- Salida DC aislada de la entrada AC;
- Capacidad de corriente compatible con los equipamientos instalados.

El tamaño total de la red varía de acuerdo con la tasa de transmisión utilizada, conforme presentado en la tabla 1.1

Tabla 1.1: Tamanho da rede x Taxa de transmissão

Taxa de transmissão	Tamanho da rede	Derivação	
125 Kbps	500 m	6	156 m
250 Kbps	250 m	6	78 m
500 Kbps	100 m	6	39 m

Para evitar reflexiones de la señal en la línea, se recomienda la instalación de resistores de terminación en las extremidades de la red, pues la falta de estos puede provocar errores intermitentes. Este resistor debe poseer las siguientes características, conforme especificación del protocolo:

- 121Ω;
- 0,25 W;
- 1 % de tolerancia.

En DeviceNet, diversos tipos de conectores pueden ser utilizados, tanto sellados cuanto abiertos. La definición del tipo a ser utilizado dependerá de la aplicación y del ambiente de operación del equipamiento. El SCA06 utiliza un conector de 5 vías cuyos terminales está presentado en la sección 3. Para una descripción completa de los conectores utilizados por el DeviceNet consulte la especificación del protocolo.

1.3.1 Camada de Enlace de Datos

La camada de enlace de datos del DeviceNet es definida por la especificación del CAN, el cual define dos estados posibles; dominante (nivel lógico 0) y recesivo (nivel lógico 1). Un nudo puede llevar la red al estado dominante si transmitir alguna información. Así, el bus solamente estará en el estado recesivo si no se tiene nudos transmisores en el estado dominante.

CAN utiliza el CSMA/NBA para acceder el medio físico. Eso significa que un nudo, antes de transmitir, debe comprobar si el bus está libre. Caso este, entonces elle podrá iniciar la transmisión de su telegrama. Caso no este, deberá aguardar. Si más de un nudo acceder la red simultáneamente, un mecanismo basado en prioridad de mensaje entrará en acción para decidir cual de ellos tendrá prioridad sobre los otros. Este mecanismo es no destructivo, o sea, el mensaje es preservado mismo que ocurran choques entre dos o más telegramas.

CAN define cuatro tipos de telegramas (data, remote, overload, error). De estos telegramas, DeviceNet utiliza solo el frame de datos (data frame) y el frame de errores (error frame).

Datos son trasladados utilizándose el frame de datos. La estructura de este frame es presentado en la figura 1.1.

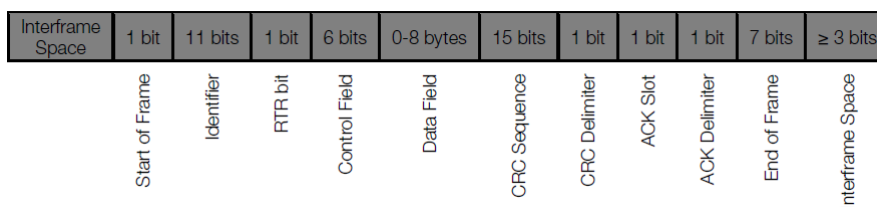


Figura 1.1: Frame de datos CAN

Ya los errores son señalados a través del frame de errores. CAN posee una verificación y un confinamiento de errores bastante robusto. Eso garantiza que un nudo con problemas no perjudique la comunicación en la red.

Para una descripción completa de los errores, consulte la especificación del CAN.

1.3.2 Camada de Transporte y Red

DeviceNet requiere que una conexión sea establecida antes de haber intercambio de datos con el dispositivo. Para establecer esta conexión, cada nudo DeviceNet debe implementar el Unconnected Message Manager (UCMM) o el Group 2 Unconnected Port. Estos dos mecanismos de posicionamiento utilizan mensajes del tipo explicit para establecer la conexión, que a seguir será utilizada para el intercambio de datos de proceso entre un nudo y otro. Este intercambio de datos utiliza mensajes del tipo I/O (consultar ítem 1.3.5).

Los telegramas DeviceNet son clasificados en grupos, le cual definen funciones y prioridades específicas. Estos telegramas utilizan el campo identificador (11 bits) del frame de datos CAN para identificar únicamente cada un de los mensajes, garantizando así el mecanismo de prioridades CAN.

Un nudo DeviceNet puede ser cliente, servidor o los dos. Además de eso, clientes y servidores pueden ser productores y/o consumidores de mensajes. En un típico nudo cliente, por ejemplo, su conexión producirá requisiciones y consumirá respuestas. Otras conexiones de clientes o servidores solo consumirán mensajes. O sea, el protocolo prevé diversas posibilidades de conexión entre los dispositivos.

El protocolo dispone también de un recurso para detección de nudos con direcciones (Mac ID) duplicados. Evite que ocurra direcciones duplicadas, en general, es más eficiente que intentar ubicarlas después.

1.3.3 Camada de Aplicación – Protocolo CIP

DeviceNet utiliza el Common Industrial Protocol (CIP) en la camada de aplicación. Se trata de un protocolo estrictamente orientado a objetos, también es utilizado por el ControlNet y por el EtherNet/IP. O sea, él es independiente del medio físico y de la camada de enlace de datos. La figura 1.1 presenta la estructura de este protocolo.

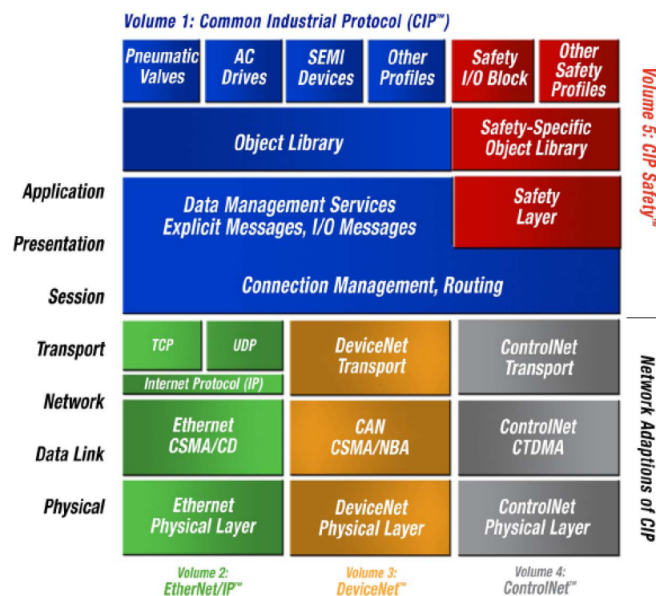


Figura 1.2: Estructura en capas del protocolo CIP

CIP tiene dos objetivos principales:

- Transporte de datos de control de los dispositivos de I/O;

- Transporte de informaciones de configuración y diagnóstico del sistema que esta siendo controlado.

Un nudo (maestro o esclavo) DeviceNet es entonces modelado por un conjunto de objetos CIP, los cuales encapsulan datos y servicios y determinan así su comportamiento.

Existen objetos obligatorios (todo dispositivo debe contener) y objetos opcionales. Objetos opcionales son aquellos que moldean el dispositivo de acuerdo con la categoría (llamado de perfil) que pertenecen, tales como: AC/DC Drive, lector de código de barras o válvula neumática. Por sierran diferentes, cada un de estos contendrá un conjunto también diferente de objetos.

Para más informaciones, consulte la especificación del DeviceNet. Ella presenta un listado completo de los perfiles de dispositivos ya padronizados por la ODVA, bien como los objetos que la componen.

1.3.4 Archivo de Configuración EDS

Cada dispositivo en una red DeviceNet posee un archivo de configuración EDS, que contiene las informaciones sobre el funcionamiento del dispositivo y debe ser registrado en el software de configuración, para programación de los dispositivos presentes en la red DeviceNet.

El archivo de configuración EDS 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 equipamiento, para utilizar un archivo EDS que sea compatible con esta versión.

1.3.5 Modos de Comunicación

El protocolo DeviceNet posee dos tipos básicos de mensajes, I/O y explicit. Cada un de ellos es adecuado a un determinado tipo de dato, conforme presentado abajo:

- I/O: tipo de telegrama síncrono dedicado al traslado de datos prioritarios entre un productor y un o más consumidores. Se dividen de acuerdo con el método de intercambio de datos. Los principales son:
 - Polled: método de comunicación en que el maestro envía un telegrama a cada un de los esclavos de su lista (scan list). Así que recibe la solicitud, el esclavo contesta prontamente la solicitud del maestro. Este proceso es repetido hasta que todos sean consultados, reiniciando el ciclo;
 - Bit-strobe: método de comunicación donde el maestro envía para la red un telegrama conteniendo 8 bytes de datos. Cada bit de estos 8 bytes representa un esclavo que, cuando es llamado, contesta de acuerdo con el programado;
 - Change of State: método de comunicación donde el intercambio de datos entre maestro y esclavo ocurre solo cuando se tiene cambios en los valores monitoreados/controlados, hasta un cierto límite de tiempo. Cuando este límite es alcanzado, la transmisión y recepción ocurrirán mismo que no se tenga ocurrido alteraciones. La configuración de esta variable de tiempo es hecha en el programa de configuración de la red;
 - Cyclic: otro método de comunicación muy semejante al anterior. La única diferencia se queda por cuenta de la producción y del consumo de mensajes. En este tipo, todo intercambio de datos ocurre en intervalos regulares de tiempo, independiente de haberes sido modificados o no. Este período también es ajustado en el software de configuración de red.
- Explicit: tipo de telegrama de uso general y no prioritario. Utilizado principalmente en tareas asíncronas tales como parametrización y configuración do equipamiento.

**¡NOTA!**

el servoconvertidor SCA06 no disponza el método de comunicación Bit-strobe.

1.3.6 Conjunto de Conexiones Predefinidas Maestro/Esclavo

DeviceNet utiliza fundamentalmente un modelo de mensajes punto a punto. Sin embargo, es bastante común utilizar un esquema predefinido de comunicación basado en el mecanismo maestro/esclavo.

Esta estructura utiliza un movimiento simplificado de mensajes del tipo I/O muy común en aplicaciones de control. Las ventajas de este método están en los requisitos necesarios para ejecutarlo, en general menores si comparados al UCMM. Hasta mismo dispositivos simples con recursos limitados (memoria, procesador de 8 bits) son capaces de ejecutar el protocolo.

2 INTERFAZ DE COMUNICACIÓN DEVICENET

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

2.1 CARACTERÍSTICAS DE LA INTERFAZ CAN

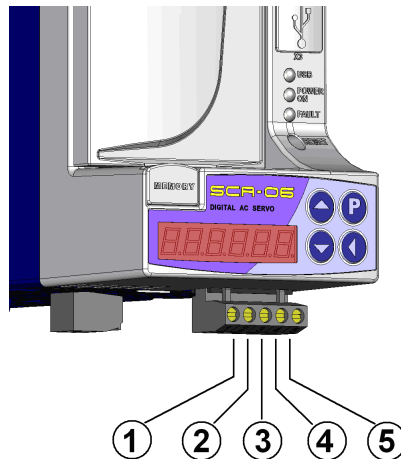


Figura 2.1: Detalle do conector CAN na parte inferior do produto

- Interfaz aislada galvánicamente y con señal diferencial, otorgando mayor robustez contra interferencia electro-magnética.
- 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 repetidores⁴.
- 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:

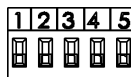


Tabla 2.1: Terminales del conector X4 para interfaz CAN

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

⁴El número límite de equipamientos que pueden ser conectados en la red, también depende del protocolo utilizado.

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 (Vcc)		
Mínimo	Máximo	Sugerido
11	30	24
Corriente (mA)		
Típico		Máximo
30		50

2.4 SEÑALIZACIONES

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

3 INSTALACIÓN EN RED DEVICENET

La red DeviceNet, 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 DeviceNet en general permiten configurar la tasa de comunicación deseada, pudiendo variar de 125Kbit/s até 500Kbit/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 especificación del protocolo ⁵.

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

Tasa de comunicación	Longitud del cable
125 Kbps	500 m
250 Kbps	250 m
500 Kbps	100 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 parametro P0702.

3.2 DIRECCIÓN EN LA RED DEVICENET

Cada dispositivo de la red CANopen precisa tener una dirección, o Node ID, entre 0 y 63. Esta dirección debe ser única para cada equipamiento. Para el servoconvertidor SCA06, la configuración de la dirección es hecha a través del parametro P0701.

3.3 RESISTOR DE TERMINACIÓN

La utilización de resistencias de terminación en las extremidades del bus CAN es fundamental para evitar reflexión de línea, que puede perjudicar la señal transmitida y ocasionar errores en la comunicación. Las extremidades del bus CAN deben poseer un resistor de terminación en el valor de 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 DeviceNet

Longitu del Cable (m)	Resistencia por metro (Ω m)	Area del Conductor (mm2)
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

⁵Diferentes productos pueden tener diferentes longitudes máximas de cable permitida para la instalación.

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

3.5 CONEXIÓN CON LA RED

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

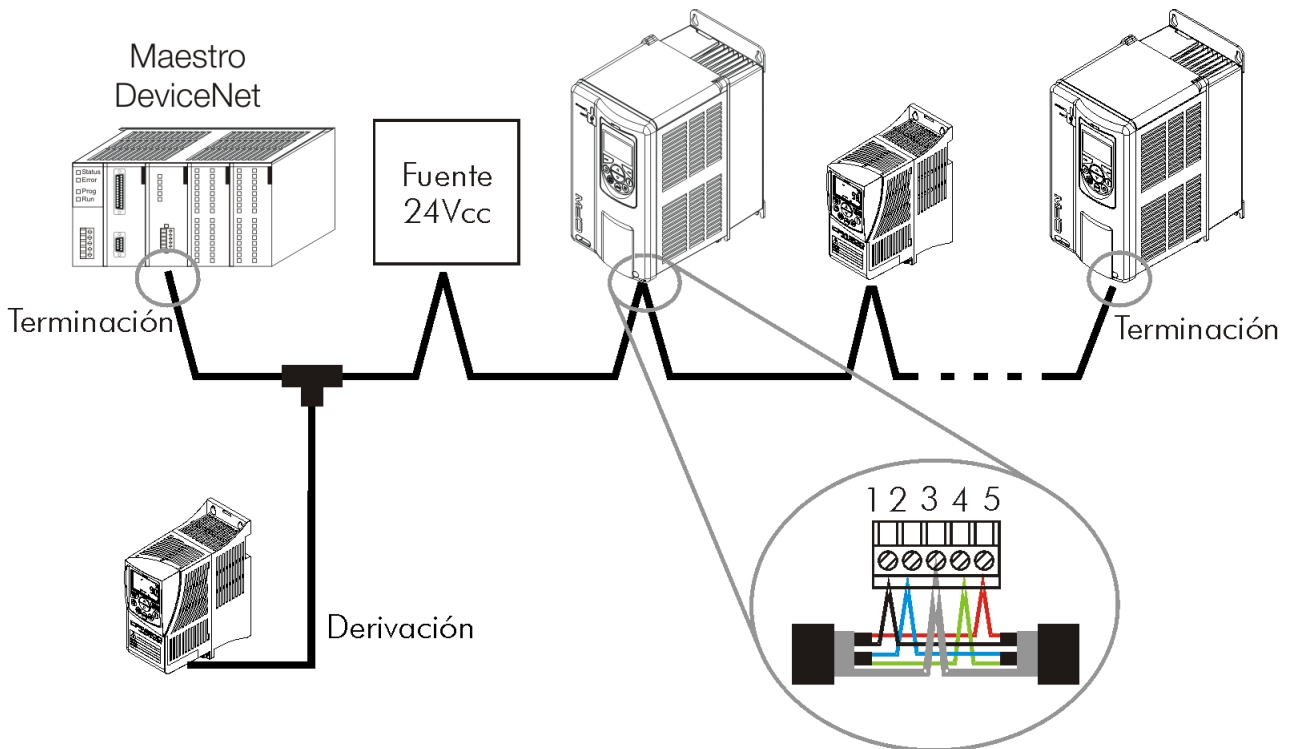


Figura 3.1: Ejemplo de instalación en red DeviceNet

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 es presentado solo os parámetros del servoconvertidor SCA06 que poseen relación con la comunicación DeviceNet.

4.1 SÍMBOLOS PARA DESCRIPCIÓN DE LAS PROPIEDADES

- **RO** Parámetro solamente de lectura
- **RW** Parámetro de lectura y escrita
- **CFG** Parámetro solamente modificado con el motor parado

P0070 - ESTADO DEL CONTROLADOR CAN

Rango de Valores:	0 = Deshabilitado 1 = Autobaud 2 = Interfaz CAN Activa 3 = Warning 4 = Error Passive 5 = Bus Off 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 P0070.
1 = Autobaud	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 = Warning	Controlador CAN atingiu o estado de warning.
4 = Error PAssive	Controlador CAN alcanzo el estado de error passive.
5 = Bus Off	Controlador CAN alcanzo el estado de bus off.
6 = Sin Alimentación	Interfaz CAN no posee alimentación entre los terminales 1 y 5 del conector.

P0071 - CONTADOR DE TELEGRAMAS CAN RECEBIDOS

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 - COTADOR DE TELEGRAMS 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.

P0077 - ESTADO DE LA RED DEVICENET

Rango de Valores:	0 = Offline 1 = Online, No Conectado 2 = Online, Conectado 3 = Conexión Caduco 4 = Falla en la Conexión 5 = Autobaud	Padrón: -
Propiedades:	RO	

Descripción:

Indica el estado de la red DeviceNet. La tabla que sigue presenta una breve descripción de estos estados.

Tabla 4.2: Valores para el parámetro P0077

Estado	Descripción
Offline	Sin alimentación o no online. Comunicación no puede ser establecida.
Online, No Conectado	Dispositivo online, más no conectado. Esclavo completó con suceso el procedimiento de verificación del MacID. Eso significa que la tasa de comunicación configurada está correcta (o fue detectada correctamente en el caso de la utilización del autobaud) y que no hay otros nodos en la red con la misma dirección. Sin embargo, en esta etapa, todavía no hay comunicación con el maestro.
Online, Conectado	Dispositivo operacional e em condições normais. Mestre alocou um conjunto de conexões do tipo I/O com o escravo. Nesta etapa ocorre efetivamente a troca de dados através de conexões do tipo I/O.
Conexión Caduco	Una o más conexiones del tipo I/O caducaran.
Falla en la Conexión	Indica que el esclavo no puede entrar en la red debido a problemas de dirección o entonces debido a la ocurrencia de bus off. Verifique si la dirección configurada ya no está siendo utilizada por otro equipamiento, si la tasa de comunicación elegida está correcta o si existen problemas en la instalación.
Autobaud	Equipamiento ejecutando rutina del mecanismo de autobaud.

P0078 - ESTADO DEL MAESTRO DEVICENET

Rango de Valores:	0 = RUN 1 = IDLE	Padrón: -
Propiedades:	RO	

Descripción:

Indica el estado del maestro de la red DeviceNet. Este puede estar en el modo de operación (Run) o en el modo de configuración (Idle).

Cuando en Run, telegramas de lectura y escrita son procesados y actualizados normalmente por el maestro. Cuando en Idle, solamente telegramas de lectura de los esclavos son actualizados por el maestro. La escrita, en este caso, se queda deshabilitada.

Cuando la comunicación estuviera deshabilitada este parámetro no representa el estado real del maestro.

P0202 - MODO DE OPERACIÓN

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

Descripción:

Este parámetro define el modo de operación del servoconvertidor SCA06, permitiendo programar qué variable se desea controlar en el motor y en la fuente de comandos, para ejecución de las funciones.

Para que el equipo sea controlado a través de la red DeviceNet, es necesario utilizar el modo 5 = DeviceNet. En caso de que este modo esté programado, serán dados, vía red DeviceNet, comandos y referencias para operación del producto.


¡NOTA!

- El control del equipo, a través de los objetos para drives, solamente es posible seleccionando la opción deseada en este parámetro, no obstante, la comunicación DeviceNet puede ser utilizada en cualquier modo de operación.
- La interfaz DeviceNet permite el control de velocidad y de torque del servoconvertidor SCA06. Para realizar funciones de posicionamiento se debe utilizar el modo de operación Ladder, elaborando un programa aplicativo en ladder y utilizando parámetros del usuario como interfaz con el maestro de la red, para control y monitoreo del equipo.

P0662 – ACCIÓN PARA ERROR DE COMUNICACIÓN

Rango de Valores:	0 = Muestra Alarme 1 = Genera Falla 2 = Ejecuta función STOP 3 = Deshabilita drive	Padrón: -
Propiedades:	RW	

Descripción:

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

Tabla 4.3: Opciones para el parámetro P0662

Opción	Descripción
0 = Muestra Alarma	Solamente indica alarma en la HMI en caso de error de comunicación. Si la comunicación es restablecida, la indicación de alarma será retirada automáticamente.
1 = Causa Falla	En lugar de alarma, la falla en el equipo es causada por un error de comunicación, siendo necesario ejecutar el reset de fallas para el retorno de su operación normal.
2 = Ejecuta función STOP	Será hecha la indicación de alarma y la ejecución del comando STOP. Para que el drive salga de esta condición, será necesario realizar el reset de fallas o deshabilitar el drive.
3 = Deshabilita drive	Será hecha la indicación de alarma y la ejecución del comando deshabilita.

Los siguientes eventos son considerados errores de comunicación:

Comunicación DeviceNet:

- Alarma A133/Falla F233: Sin alimentación en la interfaz CAN.
- Alarma A134/Falla F234: bus off.
- Alarma A136/Falha F236: Maestro DeviceNet en Idle.
- Alarma A137/Falha F237: Timeout en la conexión DeviceNet.

P0700 - PROTOCOLO CAN

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

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 Valores:	0 a 127	Padrón: 63
Propiedades:	RW	

Descripción:

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

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

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

P0702 - TASA DE COMUNICACIÓN CAN

Rango de Valores:	0 = 1 Mbit/s / Autobaud 1 = 800 Kbit/s / Autobaud 2 = 500 Kbit/s 3 = 250 Kbit/s 4 = 125 Kbit/s 5 = 100 Kbit/s / Autobaud 6 = 50 Kbit/s / Autobaud 7 = 20 Kbit/s / Autobaud 8 = 10 Kbit/s / Autobaud	Padrón: 0
Propiedades:	RW	

Descripción:

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

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

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

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

P0703 - RESET DE BUS OFF

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

Descripción:

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

Tabla 4.4: Opciones para el parámetro P0703

Opción	Descripción
0 = Reset Manual	Caso ocurra bus off, 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 bus off, 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.

P0710 – INSTANCIAS DE I/O DEVICENET

Rango de Valores:	0 = ODVA Basic Speed (2 palabras) 1 = ODVA Extended Speed (2 palabras) 2 = ODVA Extended Speed/Torque (3 palabras) 3 = Especific.Fab 2W (2 palabras) 4 = Especific.Fab 3W (3 palabras) 5 = Especific.Fab 4W (4 palabras) 6 = Especific.Fab 5W (5 palabras) 7 = Especific.Fab 6W (6 palabras) 8 = Especific.Fab 7W (7 palabras) 9 = Especific.Fab 8W (8 palabras) 10 = Especific.Fab 9W (9 palabras) 11 = Especific.Fab 10W (10 palabras) 12 = Especific.Fab 11W (11 palabras) 13 = Especific.Fab 12W (12 palabras) 14 = Especific.Fab 13W (13 palabras) 15 = Especific.Fab 14W (14 palabras)	Padrón: 0
Propiedades:	RW	

Descripción:

Permite seleccionar la instancia de la clase Assembly para comunicación do tipo I/O.

El servoconvertidor SCA06 posee siete opciones de ajustes. Dos de ellas siguen el padrón definido en el perfil AC/DC Drive Profile de la ODVA. As otras cinco, representan palabras específicas del servoconvertidor SCA06. Las tablas presentadas en seguida describen cada una de estas palabras de control y de monitoreo.


¡NOTA!

Caso este parámetro sea modificado, elle solamente será válido luego que el producto es desenergizado y energizado nuevamente.

0 = Formato de los datos para las instancias ODVA Basic Speed (2 palabras):

Llamada de Basic Speed, estas instancias representan lo más sencillo interfaz de operación de un equipamiento segundo el perfil AC/DC Drive Profile. El mapa de los datos es presentado abajo.

Monitoreo (Entrada)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	0						Running1		Faulted
	1	-							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

Control (Salida)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	0						Fault Reset		Run Fwd
	1	-							
	2	Speed Reference (low byte)							
	3	Speed Reference (high byte)							

1 = Formato de los datos para las instancias ODVA Extended Speed (2 palabras):

Llamada de Extended Speed, estas instancias representan un interfaz un poco más mejorada de operación del equipamiento que sigue el perfil AC/DC Drive Profile. El mapa de los datos es presentado abajo.

Monitoreo (Entrada)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
71	0	At Reference	Ref. from Net	Ctrl from Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

Control (Salida)

Instância	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0		NetRef	NetCtrl		Fault Reset		Warning	Faulted
	1	-							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							

2 = Formato de los datos para las instancias ODVA Extended Torque (3 palabras):

Llamadas de Extended Torque, estas instancias representan una interfaz muy semejante a la Extended Speed, teniendo como única diferencia la posibilidad de envío del límite de torque. Abajo es mostrado el mapeo de los datos.

Monitoreo (Entrada)

Instancia	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
73	0	At Reference	Ref. from Net	Ctrl from Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (low byte)							
	3	Speed Actual (high byte)							
	4	Torque Actual (low byte)							
	5	Torque Actual (high byte)							

Control (Salida)

Instancia	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
23	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1	-							
	2	Speed Reference (low byte)							
	3	Speed Reference (high byte)							
	4	Torque Limit (low byte)							
	5	Torque Limit (low byte)							

En el cuadro siguiente se presenta el significado de los datos para las instancias 20/70, 21/71 y 23/73.

Monitoreo:

Bit	Valor/Descrição
Bit 0 Faulted	0: servoconvertidor no está en estado de falla. 1: Alguna falla registrada por el servoconvertidor. Obs.: El número de la alarma puede ser leído a través del parámetro P0049 – Falla Actual.
Bit 1 Warning	0: servoconvertidor no está en estado de alarma. 1: Alguna alarma registrada por el servoconvertidor. Obs.: El número de la alarma puede ser leído a través del parámetro P0048 – Alarma Actual.
Bit 2 Running1 (Fwd)	0: El motor no está girando en sentido horario. 1: El motor está girando en sentido horario.
Bit 3 Running2 (Rev)	0: El motor no está girando en sentido antihorario. 1: El motor está girando en sentido antihorario.
Bit 4 Ready	0: servoconvertidor no está pronto para operar. 1: servoconvertidor está pronto para operar (estados Ready, Enabled o Stopping).
Bit 5 Ctrl from Net	0: Drive controlado localmente. 1: Drive controlado remotamente.
Bit 6 Ref. from Net	0: La referencia de velocidad no está siendo enviada vía red DeviceNet. 1: Indica referencia de velocidad siendo enviada vía red DeviceNet.
Bit 7 At Reference	0: servoconvertidor aún no alcanzó la velocidad programada. 1: servoconvertidor alcanzó la velocidad programada.

- Byte 1 indica el estado del drive:
 - 0 = Non Existent
 - 1 = Startup
 - 2 = Not Ready
 - 3 = Ready
 - 4 = Enabled
 - 5 = Stopping
 - 6 = Fault Stop
 - 7 = Faulted
- Bytes 2 (low) y 3 (high) representan la velocidad real del motor en RPM.
- Bytes 4 (low) y 5 (high) representan el valor real de la corriente del motor, proporcional al torque.

Control:

Bit	Valor/Descripción
Bit 0 Run Fwd	0: Para motor. 1: Gira motor en sentido horario.
Bit 1 Run Rev	0: Para motor. 1: Gira motor en sentido antihorario.
Bit 2 Fault Reset	0: Sin función. 1: En estado de falla, ejecuta el reset del servoconvertidor.
Bits 3 e 4	Reservado.
Bit 5 NetCtrl	0: servoconvertidor selecciona el modo local. 1: servoconvertidor selecciona el modo remoto.
Bit 6 NetRef	0: La referencia de velocidad no está siendo enviada vía red. 1: El envío de la referencia de velocidad sea hecho vía red.
Bits 7	Reservado.

- Bytes 2 (low) y 3 (high) representan la referencia de velocidad del motor en RPM.
- Bytes 4 (low) y 5 (high) representan la limitación de la corriente real del motor, proporcional al torque. Ese valor es escrito en los parámetros P0131 y P0132, límite de corriente negativo y positivo respectivamente.

3 = Formato de los datos para las instancias Manufacturer Specific 2W (2 palabras):
4 = Formato de los datos para las instancias Manufacturer Specific 3W (3 palabras):
5 = Formato de los datos para las instancias Manufacturer Specific 4W (4 palabras):
6 = Formato de los datos para las instancias Manufacturer Specific 5W (5 palabras):
7 = Formato de los datos para las instancias Manufacturer Specific 6W (6 palabras):
8 = Formato de los datos para las instancias Manufacturer Specific 7W (7 palabras):
9 = Formato de los datos para las instancias Manufacturer Specific 8W (8 palabras):
10 = Formato de los datos para las instancias Manufacturer Specific 9W (9 palabras):
11 = Formato de los datos para las instancias Manufacturer Specific 10W (10 palabras):
12 = Formato de los datos para las instancias Manufacturer Specific 11W (11 palabras):
13 = Formato de los datos para las instancias Manufacturer Specific 12W (12 palabras):
14 = Formato de los datos para las instancias Manufacturer Specific 13W (13 palabras):
15 = Formato de los datos para las instancias Manufacturer Specific 14W (14 palabras):

Llamadas de Manufacturer Specific, estas instancias representan la más simple interfaz de operación del equipo según el perfil del servoconvertidor SCA06. El mapeo de los datos es mostrado abajo. Además de las palabras de control, estado, referencia y valor actual de la velocidad, es posible programar hasta 24 parámetros del propio equipo para lectura y/o escritura vía red, a través de los parámetros P0711 a P0734.

Monitoreo (Entrada)

Tabla 4.5: Programación de las palabras de I/O

Instancia	Palabras de 16 bits (word)	Función	Opción del P0710					
			3	4	5	6	15	
Fixo Programable 150	#1	Palabra de estado (P0736)	3 4 5 6 15					
	#2	Velocidad/corriente Actual(*)						
	#3	Lectura #3 DeviceNet						
	#4	Lectura #4 DeviceNet						
	#5	Lectura #5 DeviceNet						
	⋮	⋮						
	#14	Lectura #14 DeviceNet						

Control (Salida)

Tabla 4.6: Programación de las palabras de I/O

Instancia	Palabras de 16 bits (word)	Función
	100	#1
#2		Referencia velocidad/corriente(*)
#3		Escritura #3 DeviceNet
#4		Escritura #4 DeviceNet
#5		Escritura #5 DeviceNet
⋮		⋮
#14		Escritura #14 DeviceNet

* El contenido de esta palabra es definido conforme el modo de operación programado en los bits 8 a 10, en la palabra de control (P0735)::

- Modo velocidad: la referencia de velocidad y la velocidad del motor son informadas en RPM.
- Modo torque: la referencia de corriente (proporcional al torque) y la corriente del motor son informadas conforme la escala del P0003.

P0711 – LECTURA #3 DEVICENET
P0712 – LECTURA #4 DEVICENET
P0713 – LECTURA #5 DEVICENET
P0714 – LECTURA #6 DEVICENET
P0715 – LECTURA #7 DEVICENET
P0716 – LECTURA #8 DEVICENET
P0717 – LECTURA #9 DEVICENET
P0718 – LECTURA #10 DEVICENET
P0719 – LECTURA #11 DEVICENET
P0720 – LECTURA #12 DEVICENET
P0721 – LECTURA #13 DEVICENET
P0722 – LECTURA #14 DEVICENET

Rango de Valores:	0 a 1249	Padrón: 0
Propiedades:	RW	

Descripción:

Estos parámetros permiten programar el contenido de las palabras 3 a 14 de entrada (input: esclavo envía para el maestro). Utilizando estos parámetros, es posible programar el número de otro parámetro cuyo contenido debe ser disponibilizado en el área de entrada del maestro de la red.

Por ejemplo, caso se desee leer del drive la corriente del motor en amperes, se debe programar en alguno de los parámetros el valor 3, pues el parámetro P0003 es el parámetro que contiene esta información. Vale recordar que el valor leído de cualquier parámetro es representado con una palabra de 16 bits. Mismo que el parámetro posea

resolución decimal, el valor es transmitido sin la indicación de los decimales. Por ejemplo, si el parámetro P0003 posee el valor 4,7 A, el valor suministrado vía red será 47.

Estos parámetros son utilizados solamente si el drive fuera programado en el parámetro P0710 para utilizar las opciones 3 hasta 15. De acuerdo con la opción seleccionada, son disponibilizadas hasta 14 palabras para lectura por el maestro de la red.

Las dos primeras palabras de entrada son fijas.


¡NOTA!

El valor 0 (cero) deshabilita la escritura en la palabra. La cantidad de palabras de entrada, por otro lado, permanece siempre igual a lo que fuera programado en el parámetro P0710.

P0723 – ESCRITA #3 DEVICENET
P0724 – ESCRITA #4 DEVICENET
P0725 – ESCRITA #5 DEVICENET
P0726 – ESCRITA #6 DEVICENET
P0727 – ESCRITA #7 DEVICENET
P0728 – ESCRITA #8 DEVICENET
P0729 – ESCRITA #9 DEVICENET
P0730 – ESCRITA #10 DEVICENET
P0731 – ESCRITA #11 DEVICENET
P0732 – ESCRITA #12 DEVICENET
P0733 – ESCRITA #13 DEVICENET
P0734 – ESCRITA #14 DEVICENET

Rango de Valores:	0 a 1249	Padrón: 0
Propiedades:	RW	

Descripción:

Estos parámetros permiten programar el contenido de las palabras 3 a 14 de salida (output: maestro envía para el esclavo). Utilizando estos parámetros, es posible programar el número de otro parámetro cuyo contenido debe ser disponibilizado en el área de salida del maestro de la red.

Por ejemplo, caso se desee escribir en el drive la rampa de aceleración, se debe programar en alguno de los parámetros el valor 100, pues el parámetro P0100 es el parámetro donde esta información es programada. Vale recordar que el valor escrito de cualquier parámetro es representado con una palabra de 16 bits. Aunque el parámetro posea resolución decimal, el valor es transmitido sin la indicación de los decimales. Por ejemplo, caso se desee programar el parámetro P0100 con el valor 5,0s, el valor escrito vía red deberá ser 50.

Estos parámetros son utilizados solamente si el drive fuera programado en el parámetro P0710 para utilizar las opciones 3 hasta 15. De acuerdo con la opción seleccionada, son disponibilizadas hasta 14 palabras para escritura

por el maestro de la red.

Las dos primeras palabras de salida son fijas.


¡NOTA!

El valor 0 (cero) deshabilita la escritura en la palabra. La cantidad de palabras de entrada, por otro lado, permanece siempre igual a lo que fuera programado en el parámetro P0710.

P0735 - PALABRA DE CONTROL

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

Descripción:

Palabra de comando del drive vía interfaz DeviceNet. Este parámetro solamente puede ser alterado vía interfaz DeviceNet. Para las demás fuentes (HMI, etc.) éste se comporta como un parámetro de solamente lectura.

Para que los comandos escritos en este parámetro sean ejecutados, es necesario que el equipo esté programado para ser controlado vía DeviceNet. Esta programación es hecha a través del parámetro P0202.

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

Bits	15 a 11	10 a 8	7	6	5 a 3	2	1	0
Função	Reservado	Modo de operación	Reset Falla	Reset Drive	Reservado	Sentido de Giro	STOP	Habilita

Bit	Valor/Descripción
Bit 0 Habilita	0: Deshabilita drive. 1: Habilita drive.
Bit 1 STOP	0: No ejecuta función STOP. 1: Ejecuta función STOP.
Bit 2 Sentido de Giro	0: Forward 1: Reverse
Bits 3 a 5	Reservado.
Bit 6 Reset Drive	0: Sin función. 1: Ejecuta Reset del drive.
Bit 7 Reset de Falha	0: Sin función. 1: En estado de falla ejecuta el Reset de la falla.
Bits 8 a 10 Modo de Opeação	1: Modo Torque 2: Modo Velocidad
Bits 11 a 15	Reservado.

P0736 - ESTADO LÓGICO

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

Descripción:

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

Bits	15 a 11	10 a 8	7	6	5 a 3	2	1	0
Función	Reservado	Modo de operación	Falla	Atingiu Ref.	Reservado	Sentido de Giro	STOP	Habilitado

Bit	Valor/Descripción
Bit 0 Habilitado	0: El Drive no está habilitado. 1: El Drive está habilitado.
Bit 1 STOP	0: Sin función. 1: Ejecutando la función STOP.
Bit 2 Sentido de Giro	0: Horario. 1: Antihorario.
Bits 3 a 5	Reservado.
Bit 6 Atingiu Ref.	0: Referencia no alcanza. 1: Referencia alcanza.
Bit 7 Falha	0: Drive sin falla. 1: El Drive está en falla.
Bits 8 a 10 Modo de Opeación	1: Modo Torque. 2: Modo Velocidad.
Bits 11 a 15	Reservado.

5 CLASES DE OBJETOS SOPORTADAS

Todo dispositivo DeviceNet es modelado por un conjunto de objetos. Son ellos los responsables por definir que función, determinado equipamiento tendrá. O sea, de acuerdo con los objetos implementados, este equipamiento podrá ser un adaptador de comunicación, un drive AC/DC, un sensor fotoeléctrico, etc.. Objetos obligatorios y opcionales son definidos en cada uno de estos perfiles de dispositivos (Device Profile). El servoconvertidor SCA06 soporta todas las clases obligatorias del perfil AC/DC Drive Profile. Soporta también clases específicas del fabricante. Detalles de cada una de ellas son presentados en las secciones a seguir.

5.1 CLASE IDENTITY (01H)

Suministra informaciones generales sobre la identidad del dispositivo, tales como VendorID, Product Name, Serial Number, etc. Están implementados los siguientes atributos:

Tabla 5.1: Atributos de la instancia de la Clase Identity

Atributo	Método	Nombre	Padrón	Descripción
1	GET	Vendor ID	355h	Identificador del fabricante
2	GET	Product Type	2h	Tipo del producto
3	GET	Product Code		Código del producto
4	GET	Vendor Revision		Revisión del firmware
5	GET	Status		Estado actual del dispositivo
6	GET	Serial Number		Número serial
7	GET	Product Name	SCA-06	Nombre del producto

5.2 CLASE MESSAGE ROUTER (02H)

Suministra informaciones sobre el objeto roteador de mensajes del tipo explicit. En el SCA06, esta clase no tiene cualquier atributo implementado.

5.3 CLASE DEVICENET (03H)

Responsable por mantener la configuración y el estado de las conexiones físicas del nodo DeviceNet. Están implementados los siguientes atributos:

Tabla 5.2: Atributos de la Clase DeviceNet

Atributo	Metodo	Nombre	Min./Max	Padrón	Descripción
1	GET	Revision	1 - 65535		Revisión de la definición del Objeto de Clase DeviceNet sobre cual la implementación fue basada.

Tabla 5.3: Atributos de la instancia de la Clase DeviceNet

Atributo	Metodo	Nombre	Min./Max	Padron	Descripción
1	GET/SET	Mac ID	0 - 63	63	Dirección del nodo
2	GET/SET	Baud Rate	0 - 2	0	Tasa de transmisión
3	GET/SET	Bus-Off Interrupt	0 - 1	1	Reset de bus-off
4	GET/SET	Bus-Off Counter	0 - 255		Contador de bus-off
5	GET/SET	Allocation Information			Información sobre el allocation byte

5.4 CLASE ASSEMBLY (04H)

Clase cuya función es juntar diversos atributos en una sola conexión. En el SCA06 solamente el atributo Data (3) está implementado (5.4)

Tabla 5.4: Atributos de las instancias de la Clase Assembly

Atributo	Metodo	Nombre	Descripción
3	GET/SET	Data	Datos de la instancia

En el SCA06, la clase Assembly contiene las siguientes instancias:

Tabla 5.5: Instancias de la Clase Assembly

Instancia	Tamaño	Descripción
20	2 palabras	ODVA Basic Speed Control Output.
70	2 palabras	ODVA Basic Speed Control Output.
21	2 palabras	ODVA Extended Speed Control Output.
71	2 palabras	ODVA Extended Speed Control Output.
23	3 palabras	ODVA Extended Speed/Torque Control Output.
73	3 palabras	ODVA Extended Speed/Torque Control Output.
100	2 - 14 palabras	Especifica de lo Fabricante Salida
150	2 - 14 palabras	Especifica de lo Fabricante Entrada

5.5 CLASE CONNECTION (05H)

Instancia conexiones del tipo I/O y explicit. Están implementados los siguientes atributos:

5.5.1 Instancia 1: Explicit Messag

Tabla 5.6: Clase Connection – Instancia 1: Explicit Message

Atributo	Método	Nombre	Descripción
1	GET	State	Estado del objeto
2	GET	Instance Type	I/O o explicit
3	GET	transport Class trigger	Define comportamiento de la conexión
4	GET	Produced Connection ID	Identificador CAN de transmisión
5	GET	Consumed Connection ID	Identificador CAN de recepción
6	GET	Initial Comm. Charac.	Define el grupo de mensajes asociado a esta conexión
7	GET	Produced Connection Size	Tamaño en bytes de esta conexión de transmisión
8	GET	Consumed Connection Size	Tamaño en bytes de esta conexión de recepción
9	GET/SET	Expected Packet Rate	Define valores de tiempo utilizado internamente
12	GET	Watchdog Timeout Action	Define como tratar contador Inactivity/Watchdog
13	GET	Produced Connection Path Length	Número de bytes de la conexión productora
14	GET	Produced Connection Path	Camino de los objetos productores de datos
15	GET	Consumed Connection Path Length	Número de bytes de la conexión consumidora
16	GET	Consumed Connection Path	Camino de los objetos consumidores de datos
17	GET/SET	Production Inhibit Time	Define el tiempo mínimo para nueva producción de datos

5.5.2 Instancia 2: Polled

Tabla 5.7: Clase Connection – Instância 2: Polled

Atributo	Método	Nombre	Descripción
1	GET	State	Estado del objeto
2	GET	Instance Type	I/O o explicit
3	GET	transport Class trigger	Define comportamiento de la conexión
4	GET	Produced Connection ID	Identificador CAN de transmisión
5	GET	Consumed Connection ID	Identificador CAN de recepción
6	GET	Initial Comm. Charac.	Define el grupo de mensajes asociado a esta conexión
7	GET	Produced Connection Size	Tamaño en bytes de esta conexión de transmisión
8	GET	Consumed Connection Size	Tamaño en bytes de esta conexión de recepción
9	GET/SET	Expected Packet Rate	Define valores de tiempo utilizado internamente
12	GET	Watchdog Timeout Action	Define como tratar contador Inactivity/Watchdog
13	GET	Produced Connection Path Length	Número de bytes de la conexión productora
14	GET	Produced Connection Path	Camino de los objetos productores de datos
15	GET	Consumed Connection Path Length	Número de bytes de la conexión consumidora
16	GET	Consumed Connection Path	Camino de los objetos consumidores de datos
17	GET/SET	Production Inhibit Time	Define el tiempo mínimo para nueva producción de datos

5.5.3 Instancia 4: Change of State/Cyclic

Tabla 5.8: Clase Connection – Instance 4: Change of State/Cyclic

Atributo	Método	Nombre	Descripción
1	GET	State	Estado del objeto
2	GET	Instance Type	I/O o explicit
3	GET	transport Class trigger	Define comportamiento de la conexión
4	GET	Produced Connection ID	Identificador CAN de transmisión
5	GET	Consumed Connection ID	Identificador CAN de recepción
6	GET	Initial Comm. Charac.	Define el grupo de mensajes asociado a esta conexión
7	GET	Produced Connection Size	Tamaño en bytes de esta conexión de transmisión
8	GET	Consumed Connection Size	Tamaño en bytes de esta conexión de recepción
9	GET/SET	Expected Packet Rate	Define valores de tiempo utilizado internamente
12	GET	Watchdog Timeout Action	Define como tratar contador Inactivity/Watchdog
13	GET	Produced Connection Path Length	Número de bytes de la conexión productora
14	GET	Produced Connection Path	Camino de los objetos productores de datos
15	GET	Consumed Connection Path Length	Número de bytes de la conexión consumidora
16	GET	Consumed Connection Path	Camino de los objetos consumidores de datos
17	GET/SET	Production Inhibit Time	Define el tiempo mínimo para nueva producción de datos

5.6 CLASE MOTOR DATA (28H)

Clase que almacena datos de placa del motor conectado al producto. Están implementados los siguientes atributos:

Tabla 5.9: Atributos da classe Motor Data

Atributo	Método	Nombre	Min/Max	Descripción
1	GET	Revision	1 - 65535	Revisão da definição do Objeto de Classe Motor Data sobre qual a implementação foi baseada.
2	GET	Max Instance	1 - 65535	Número máximo de instancias.

Tabla 5.10: Atributos da instância da classe Motor Data

Atributo	Método	Nombre	Mín/Max	Unidade	Padrón	Descripción
3	Get	Motor Type	0 - 10	-	7	0 = Non Standard Motor 1 = PM DC Motor 2 = FC DC Motor 3 = PM Synchronous Motor 4 = FC Synchronous Motor 5 = Switched Reluctance Motor 6 = Wound Rotor Induction Motor 7 = Squirrel Cage Induction Motor 8 = Stepper Motor 9 = Sinusoidal PM BL Motor 10 = Trapezoidal PM BL Motor
6	Get/Set	Rated Current	0-999.9	100mA		Corriente nominal
7	Get/Set	rated Voltage	0-600	V		Tensión nominal

5.7 CLASE MOTOR DATA (29H)

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

Tabla 5.11: Atributos de la Clase Control Supervisor

Atributo	Método	Nombre	Min/Max	Descripción
1	GET	Revision	1 - 65535	Revisión de la definición del Objeto de Clase Control Supervisor sobre cual la implementación fue basada
2	GET	Max Instance		Número máximo de instancias.

Tabla 5.12: Atributos de la instancia de la Clase Control Supervisor

Atributo	Método	Nombre	Min/Max	Padrão	Descripción
3	Get/Set	Run 1	0 - 1	-	Run Fwd
4	Get/Set	Run 2	0 - 1	-	Run Rev
5	Get/Set	NetCtrl	0 - 1	0	0 = Control local 1 = Control vía red
6	Get	State	0 - 7	-	0 = Vendor specific 1 = Startup 2 = Not Ready 3 = Ready 4 = Enable 5 = Stopping 6 = Fault Stop 7 = Fault
7	Get	Running 1	0 - 1	0	0 = Otro estado 1 = (Enabled y Run1) o (Stopping y Running1) o (Fault Stop y Running1)
8	Get	Running 2	0 - 1	0	0 = Otro estado 1 = (Enabled y Run2) o (Stopping y Running2) o (Fault Stop y Running2)
9	Get	Ready	0 - 1	0	0 = Otro estado 1 = Ready o Enabled o Stopping
10	Get	Faulted	0 - 1	0	0 = Sin fallos 1 = Fallo ocurrido
11	Get	Warning	0 - 1	0	0 = Sin warnings
12	Get/Set	Fault Reset	0 - 1	0	0 = Sin acción 0 ->1 = Reset de errores
15	Get	Ctrl from Net	0 - 1	0	0 = Control es local 1 = Control es vía red

5.8 CLASE AC/DC DRIVE (2AH)

Contiene informaciones específicas de un AC/DC Drive tales como modo de operación y escalas de velocidad y torque. Están implementados los siguientes atributos:

Tabla 5.13: Atributos de la Clase AC/DC Drive

Atributo	Método	Nombre	Min/Max	Descripción
1	GET	Revision	1 - 65535	Revisión de la definición del Objeto de Clase AC/DC Drive sobre cual la implementación fue basada
2	GET	Max Instance		Número máximo de instancias.

Tabla 5.14: Atributos de la instancia de la Clase AC/DC Drive

Atributo	Método	Nombre	Min/Max	Padrón	Descripción
4	Get/Set	NetRef 2	0 - 1	0	0 = Referencia local 1 = Referencia via rede
6	Get	DriveMode	1 - 2	-	1 = Speed control (open loop) 2 = Speed control (closed loop)
7	Get	Speed Actual	0 - 9999		Velocidad real (mejor aproximación)
8	Get/Set	Speed Ref	0 - 9999	0	Referência de velocidad


¡NOTA!

El SCA06 operará en modo de velocidad independiente del contenido del atributo DriveMode.

5.9 CLASE MESSAGE ROUTER (2BH)

La función de esta clase es controlar la recepción de mensajes de reconocimiento (acknowledge).

Tabla 5.15: Acknowledge Handler Class instance attributes

Atributo	Método	Nombre
1	GET/Set	Acknowledge Timer
2	GET	Retry Limit
3	GET	COS Production Connection Instance

5.10 CLASES ESPECÍFICAS DEL FABRICANTE

Las clases específicas del fabricante son utilizadas para definir todos los parámetros del SCA06. Ellas permiten que el usuario lea y escriba en cualquier parámetro a través de la red. Para esto, mensajes DeviceNet del tipo explicit son utilizadas. Hay rangos separadas para cada grupo de parámetros, conforme presentado en la Tabla 5.16:

Tabla 5.16: Clases Específicas del Fabricante

Clase	Nombre	Rango de valores
Clase 100 (64h)	VENDOR CLASS F1	Parametro 000 - 099
Clase 101 (65h)	VENDOR CLASS F2	Parametro 100 - 199
Clase 102 (66h)	VENDOR CLASS F3	Parametro 200 - 299
Clase 103 (67h)	VENDOR CLASS F4	Parametro 300 - 399
Clase 104 (68h)	VENDOR CLASS F5	Parametro 400 - 499
Clase 105 (69h)	VENDOR CLASS F6	Parametro 500 - 599
Clase 106 (6Ah)	VENDOR CLASS F7	Parametro 620 - 699
Clase 107 (6Bh)	VENDOR CLASS F8	Parametro 700 - 799
Clase 108 (6Ch)	VENDOR CLASS F9	Parametro 800 - 899
Clase 109 (6Dh)	VENDOR CLASS F10	Parametro 900 - 999
Clase 110 (6Eh)	VENDOR CLASS F11	Parametro 1000 - 1099
Clase 111 (6Fh)	VENDOR CLASS F12	Parametro 1100 - 1199

Tabla 5.17: *Parámetros das Clases específicas del fabricante*

Parametro	Clase	Instancia	Atributo
P0000	Clase 100 (64h)	1	100
P0001	Clase 100 (64h)	1	101
P0002	Clase 100 (64h)	1	102
...
P0100	Clase 101 (65h)	1	100
P0101	Clase 101 (65h)	1	101
P0102	Clase 101 (65h)	1	102
...
P0200	Clase 102 (66h)	1	100
P0201	Clase 102 (66h)	1	101
P0202	Clase 102 (66h)	1	102
...
P0300	Clase 103 (67h)	1	100
P0301	Clase 103 (67h)	1	101
P0302	Clase 103 (67h)	1	102
...


¡NOTA!

- Para estas clases, el SCA06 utiliza solamente la instancia 1.
- También para estas clases, los parámetros son accedidos adicionando el valor decimal 100 a los dígitos de la decena de cualquier parámetro. Este nuevo número es llamado de atributo.

Por ejemplo:

Parámetro 23: Clase 64h, instancia 1, atributo 123. Este camino da acceso al P0023.

Parámetro 100: Clase 65h, instancia 1, atributo 100. Este camino da acceso al P0100.

Parámetro 202: Clase 66h, instancia 1, atributo 102. Este camino da acceso al P0202.

6 FALLOS Y ALARMAS

A133/F233 - SIN ALIMENTACIÓN EN EL INTERFAZ CAN

Descripción:

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

Actuación

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

Si la interfaz CAN estuviera alimentada y si fuera detectada la falta de alimentación en el interfaz CAN, será señalizada a través del HMI el mensaje de alarma A133 – o falla 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/F234 - BUS OFF

Descripción:

Detectado error de bus off en el interfaz CAN.

Actuación:

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

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

Posibles Causas/Corrección:

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

A136/F236 - MAESTRO EN IDLE

Descripción:

Alarma que señala que el maestro de la red DeviceNet está en modo Idle.

Actuación: Actúa cuando el SCA06 detecta que el maestro de la red fue para el modo Idle. En este modo, solamente las variables leídas del esclavo continúan siendo actualizadas en la memoria del maestro. Ninguno de los comandos enviados al esclavo es procesado.

En este caso será señalado a través del HMI el mensaje de alarma A136 – o falla F236 dependiendo de la programación hecha en el parámetro P0313. Para la condición de alarma, caso el maestro regrese nuevamente para el modo Run (estado normal de operación del equipamiento), la señalización de alarma desaparecerá automáticamente del HMI.

Posibles Causas/Corrección:

- Ajuste la llave que comanda el modo de operación del maestro para ejecución (Run) o entonces el bit correspondiente en la palabra de configuración del software del maestro. En caso de dudas, consulte la documentación del maestro en uso.

A137/F237 - TIMEOUT EN LA CONEXIÓN DEVICENET
Descripción:

Alarma que señala que una o más conexiones I/O DeviceNet caducaran.

Actuación:

Ocurre cuando, por algún motivo, luego que iniciada la comunicación cíclica del maestro con el producto, esta comunicación fuera interrumpida.

En este caso será señalizada a través del HMI el mensaje de alarma A137 – o falla F237 dependiendo de la programación hecha en el parámetro P0313. Para la condición de alarma, caso la conexión con el maestro sea restablecida, la señalización de alarma desaparecerá automáticamente del HMI.

Posibles Causas/Corrección:

- Verificar el estado del maestro de la red.
- Verificar instalación de la red, cable roto o falla/mal contacto en las conexiones con la red.



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