

# SoMachine

## Funciones de gestión CANopen Guía de la biblioteca CAA CiA 405

06/2011

EIO0000000498.02

[www.schneider-electric.com](http://www.schneider-electric.com)

**Schneider**  
 **Electric**

---

La información que se ofrece en esta documentación contiene descripciones de carácter general y/o características técnicas sobre el rendimiento de los productos incluidos en ella. La presente documentación no tiene como objetivo sustituir ni debe emplearse para determinar la idoneidad o fiabilidad de dichos productos para aplicaciones de usuario específicas. Los usuarios o integradores tienen la responsabilidad de llevar a cabo un análisis de riesgos adecuado y exhaustivo, así como la evaluación y pruebas de los productos en relación con la aplicación o uso en cuestión de dichos productos. Ni Schneider Electric ni ninguna de sus filiales o asociados asumirán responsabilidad alguna por el uso inapropiado de la información contenida en este documento. Si tiene sugerencias para mejoras o modificaciones o ha hallado errores en esta publicación, le rogamos que nos lo notifique.

No se podrá reproducir este documento de ninguna forma, ni en su totalidad ni en parte, ya sea por medios electrónicos o mecánicos, incluida la fotocopia, sin el permiso expreso y por escrito de Schneider Electric.

Al instalar y utilizar este producto es necesario tener en cuenta todas las regulaciones sobre seguridad correspondientes, ya sean regionales, locales o estatales. Por razones de seguridad y para garantizar que se siguen los consejos de la documentación del sistema, las reparaciones sólo podrá realizarlas el fabricante.

Cuando se utilicen dispositivos para aplicaciones con requisitos técnicos de seguridad, siga las instrucciones pertinentes.

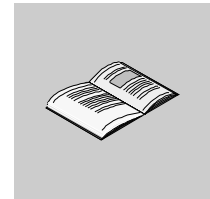
Si con nuestros productos de hardware no se utiliza el software de Schneider Electric u otro software aprobado, pueden producirse lesiones, daños o un funcionamiento incorrecto del equipo.

Si no se tiene en cuenta esta información se pueden causar daños personales o en el equipo.

© 2011 Schneider Electric. Reservados todos los derechos.

---

# Tabla de materias



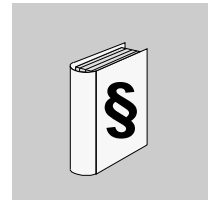
	<b>Información de seguridad</b> .....	<b>5</b>
	<b>Acerca de este libro</b> .....	<b>7</b>
<b>Parte I</b>	<b>Introducción</b> .....	<b>9</b>
<b>Capítulo 1</b>	<b>Introducción al protocolo CANopen</b> .....	<b>11</b>
	Introducción .....	12
	Protocolo NMT .....	13
	Protocolo PDO .....	14
	Protocolo SDO .....	15
	Protocolo EMCY .....	16
	Protocolo SYNC .....	17
	Protocolo de control de errores .....	19
<b>Capítulo 2</b>	<b>Introducción a CiA405</b> .....	<b>21</b>
	Introducción .....	21
<b>Parte II</b>	<b>Descripciones de bloques de funciones</b> .....	<b>23</b>
<b>Capítulo 3</b>	<b>E/S y comportamiento comunes de los bloques de funciones</b> .....	<b>25</b>
3.1	E/S y comportamiento comunes .....	25
	Descripción de E/S común .....	26
	Códigos de error detectados del núcleo CANopen .....	28
	Diagramas de ejecución de bloques de funciones .....	29
<b>Capítulo 4</b>	<b>Bloques de funciones de gestión de red</b> .....	<b>31</b>
4.1	CIA405.NMT: Gestión del estado de NMT de dispositivos .....	32
	Administración del estado NMT para dispositivos .....	32
4.2	CIA405.RECV_EM CY: Exploración de mensajes EMCY .....	35
	Exploración de mensajes EMCY .....	35
4.3	CIA405.RECV_EM CY_DEV: Obtener mensaje EMCY de dispositivo .....	37
	Obtener mensaje EMCY del dispositivo .....	37
<b>Capítulo 5</b>	<b>Bloques de funciones de ID del nodo</b> .....	<b>39</b>
5.1	CIA405.GET_LOCAL_NODE_ID: Obtener ID de nodo CANopen del controlador .....	39
	Obtener ID de nodo CANopen del controlador .....	39

---

<b>Capítulo 6</b>	<b>Bloques de funciones de estado de consulta . . . . .</b>	<b>41</b>
6.1	CIA405.GET_CANOPEN_KERNEL_STATE: Obtener el estado del kernel CANopen . . . . .	42
	Obtener estado del núcleo CANopen . . . . .	42
6.2	CIA405.GET_STATE: Obtener el estado del dispositivo CANopen . . . . .	43
	Obtener estado del dispositivo CANopen . . . . .	43
<b>Capítulo 7</b>	<b>Bloques de funciones de acceso SDO . . . . .</b>	<b>45</b>
7.1	CIA405.SDO_READ: Leer objetos CANopen de cualquier tamaño. . . . .	46
	Leer objetos CANopen de cualquier tamaño . . . . .	46
7.2	CIA405.SDO_READ4: Leer objetos CANopen de hasta 4 bytes. . . . .	49
	Leer objetos CANopen de hasta 4 bytes . . . . .	49
7.3	CIA405.SDO_WRITE: Escribir objetos CANopen de cualquier tamaño . . . . .	52
	Escribir objetos CANopen de cualquier tamaño . . . . .	52
7.4	CIA405.SDO_WRITE4: Escribir objetos CANopen de hasta 4 bytes . . . . .	57
	Escribir objetos CANopen de hasta 4 bytes . . . . .	57
<b>Glosario</b>	. . . . .	<b>59</b>
<b>Índice</b>	. . . . .	<b>63</b>

---

## Información de seguridad



---

### Información importante

#### AVISO

Lea atentamente estas instrucciones y observe el equipo para familiarizarse con el dispositivo antes de instalarlo, utilizarlo o realizar su mantenimiento. Los mensajes especiales que se ofrecen a continuación pueden aparecer a lo largo de la documentación o en el equipo para advertir de peligros potenciales o para ofrecer información que aclara o simplifica los distintos procedimientos.



La inclusión de este icono en una etiqueta de peligro o advertencia indica un riesgo de descarga eléctrica, que puede provocar lesiones si no se siguen las instrucciones.



Éste es el icono de alerta de seguridad. Se utiliza para advertir de posibles riesgos de lesiones. Observe todos los mensajes que siguen a este icono para evitar posibles lesiones o incluso la muerte.

### PELIGRO

**PELIGRO** indica una situación inminente de peligro que, si no se evita, **provocará** lesiones graves o incluso la muerte.

### ADVERTENCIA

**ADVERTENCIA** indica una situación potencialmente peligrosa que, si no se evita, **puede provocar la** muerte o lesiones graves.

---

 <b>AVISO</b>
--

<b>AVISO</b> indica una situación potencialmente peligrosa que, si no se evita, <b>puede provocar</b> lesiones leves o moderadas.
---

<b>AVISO</b>
--------------

<b>AVISO</b> , utilizado sin el símbolo de alerta de seguridad, indica una situación potencialmente peligrosa que, si no se evita, <b>puede provocar</b> daños en el equipo.
--

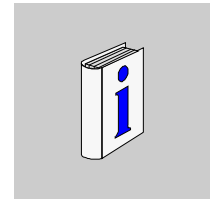
### TENGA EN CUENTA

La instalación, manejo, puesta en servicio y mantenimiento de equipos eléctricos deberán ser realizados sólo por personal cualificado. Schneider Electric no se hace responsable de ninguna de las consecuencias del uso de este material.

Una persona cualificada es aquella que cuenta con capacidad y conocimientos relativos a la construcción, el funcionamiento y la instalación de equipos eléctricos y que ha sido formada en materia de seguridad para reconocer y evitar los riesgos que conllevan tales equipos.

---

## Acerca de este libro



---

### Presentación

#### Objeto

En este documento se describen los bloques de funciones de la biblioteca CAA CiA 405 SoMachine utilizados para gestionar y monitorizar la red y los dispositivos CANopen de la aplicación del controlador.

#### Campo de aplicación

Este documento se ha actualizado con la publicación de SoMachine V3.0.

#### Información relativa al producto

### **ADVERTENCIA**

#### **FUNCIONAMIENTO INESPERADO DEL EQUIPO**

- Utilice sólo software aprobado por Schneider Electric para este equipo.
- Actualice el programa de aplicación siempre que cambie la configuración de hardware física.

**El incumplimiento de estas instrucciones puede causar la muerte, lesiones serias o daño al equipo.**

---

## **ADVERTENCIA**

### **PÉRDIDA DE CONTROL**

- El diseñador del esquema de control debe tener en cuenta los posibles modos de fallo de rutas de control y, para ciertas funciones de control críticas, proporcionar los medios para lograr un estado seguro durante y después de un fallo de ruta. Funciones de control críticas son, por ejemplo, una parada de emergencia y una parada de sobrerrecorrido, un corte de alimentación y un reinicio.
- Para las funciones críticas de control deben proporcionarse rutas de control separadas o redundantes.
- Las rutas de control del sistema pueden incluir enlaces de comunicación. Deben tenerse en cuenta las implicaciones de los retrasos de transmisión no esperados o los fallos en el enlace.
- Tenga en cuenta todas las reglamentaciones para la prevención de accidentes y las directrices de seguridad locales.<sup>1</sup>
- Cada implementación de este equipo debe probarse de forma individual y exhaustiva antes de entrar en servicio.

**El incumplimiento de estas instrucciones puede causar la muerte, lesiones serias o daño al equipo.**

<sup>1</sup> Para obtener información adicional, consulte NEMA ICS 1.1 (última edición), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" (Directrices de seguridad para la aplicación, la instalación y el mantenimiento del control de estado estático) y NEMA ICS 7.1 (última edición), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" (Estándares de seguridad para la construcción y guía para la selección, instalación y utilización de sistemas de unidades de velocidad ajustable) o su equivalente aplicable a la ubicación específica.

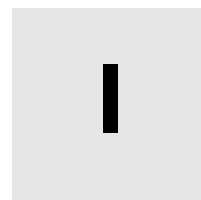
### **Comentarios del usuario**

Envíe sus comentarios a la dirección electrónica [techcomm@schneider-electric.com](mailto:techcomm@schneider-electric.com).



---

# Introducción



---

## Descripción general

En esta sección se proporciona una descripción general de los protocolos utilizados en CANopen, y se describe el perfil de dispositivo y la interfaz de CANopen para los controladores programables IEC 61131-3 (CiA405)

**NOTA:** parte del contenido de esta introducción se ha extraído y adaptado del sitio web de CAN in Automation (CiA) ([www.can-cia.org](http://www.can-cia.org)). CiA® y CANopen® son marcas registradas de la comunidad de CAN in Automation e.V

**Acerca de CiA:** CAN in Automation (CiA) es el grupo internacional de usuarios y fabricantes de CAN (red de área del controlador). La asociación sin ánimo de lucro proporciona información técnica, de productos y de marketing acerca de CAN, estandarizada internacionalmente en la serie ISO 11898. El objetivo es promocionar la imagen de CAN y proporcionar una ruta para futuros desarrollos de la tecnología CAN. Una parte importante del esfuerzo de la organización consiste en desarrollar y mantener las especificaciones de CANopen, así como dar soporte a todos los demás protocolos de capa superior basados en CAN estandarizados internacionalmente.

## Contenido de esta parte

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
1	Introducción al protocolo CANopen	11
2	Introducción a CiA405	21



---

# Introducción al protocolo CANopen



---

## Contenido de este capítulo

Este capítulo contiene los siguiente apartados:

Apartado	Página
Introducción	12
Protocolo NMT	13
Protocolo PDO	14
Protocolo SDO	15
Protocolo EMCY	16
Protocolo SYNC	17
Protocolo de control de errores	19

## Introducción

### Protocolo CANopen

El protocolo CANopen proporciona objetos de comunicación estandarizada (COB) para datos en tiempo real (objetos de datos de proceso (PDO)), datos de configuración (objetos de datos de servicio (SDO)) y datos de gestión de red (mensaje de inicio, mensaje NMT y mensaje de control de errores), así como otras funciones (marca de hora, mensaje SYNC y mensaje de emergencia). Se puede acceder a todos los objetos de comunicación a través de la red CAN en el diccionario de objetos de dispositivo. Estos objetos son direccionables mediante un índice de 16 bits. En el caso de una matriz y objetos de registro, hay un subíndice de 8 bits adicional. Un objeto de comunicación se identifica mediante un identificador dedicado y exclusivo (COB ID) a través de la red CANopen.

### Diccionario de objetos de dispositivo

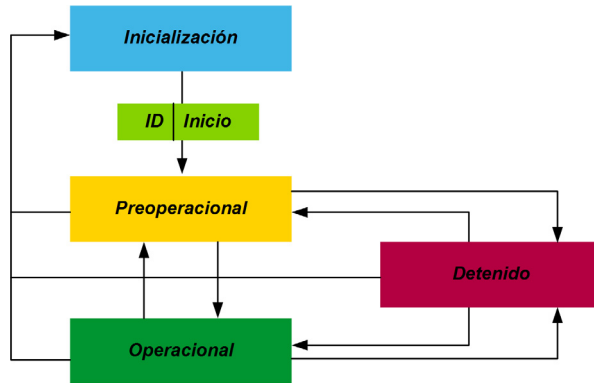
Las entradas del diccionario de objetos de un dispositivo CANopen se describen en un archivo EDS (hoja de datos electrónica). Los parámetros de maestro y esclavos CANopen se pueden ajustar en el configurador de CANopen SoMachine. Para obtener más información sobre la configuración de CANopen, consulte la sección CoDeSys de la ayuda en línea, el capítulo Editores/Editores de dispositivos/Editor de configuración CANbus.

## Protocolo NMT

### Protocolo de gestión de red (NMT)

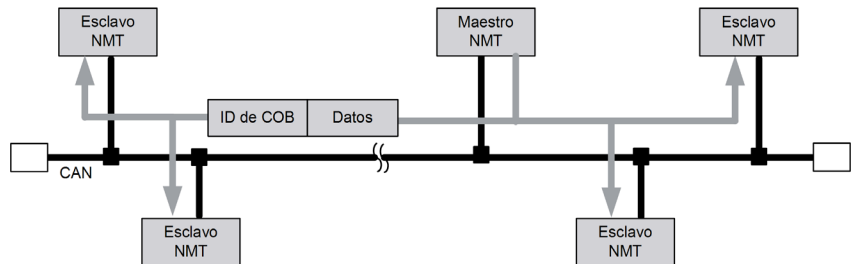
El objetivo del protocolo de gestión de red (NMT) es controlar el comportamiento de red de los dispositivos esclavos CANopen NMT. Un participante dedicado o todos los participantes de red se conectan a través del protocolo NMT en sus máquinas de estado esclavo NMT.

En la siguiente figura se representa la máquina de estado esclavo CANopen NMT.



### Servicios NMT a través de la relación de maestro-esclavo

Todos los dispositivos CANopen evalúan los comandos NMT entrantes (código de especificador de comando (CS) en el mensaje NMT). Sólo un dispositivo CANopen con la capacidad de maestro NMT puede transmitir mensajes NMT. En una red CANopen, sólo hay un maestro NMT activo.



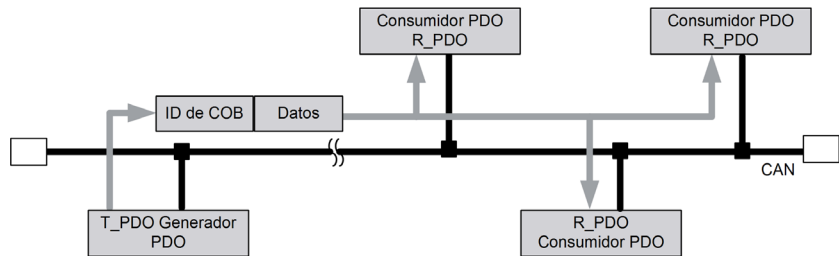
## Protocolo PDO

### Protocolo PDO (Objetos de datos de proceso)

Los objetos de datos de proceso (PDO) son mensajes CAN breves (datos de 8 bytes como máximo) de alta prioridad que se transmiten en modalidad de difusión. Por lo tanto, los PDO son adecuados para la transmisión de datos en tiempo real como por ejemplo la información de control y estado de una unidad o un módulo de E/S, valores medidos proporcionados por sensores, etc.

Los PDO se transmiten sin confirmación. Esto significa que no hay confirmación de recepción, incluso después de que un participante de red específico reciba la información.

En la siguiente figura se representa la transmisión de PDO desde el generador (PDO de transmisión) a los consumidores (PDO de recepción).



### Eventos de activación para la transmisión de PDO

Los eventos de activación para la transmisión de PDO se definen a continuación.

Evento de activación	Definición
Controlado por evento o temporizador	Un evento de dispositivo interno desencadena la transmisión de PDO (por ejemplo, cambio de valor, el valor de la temperatura supera un límite determinado, temporizador de eventos transcurrido, etc.).
Requerido remotamente	Transmisión de PDO requerida por petición de transmisión remota (RTR).
Síncrono cíclico	La transmisión de los PDO se asocia a la recepción del mensaje SYNC.
Síncrono acíclico	Un evento específico de dispositivo definido desencadena el PDO, que se transmite con la recepción del siguiente mensaje SYNC.

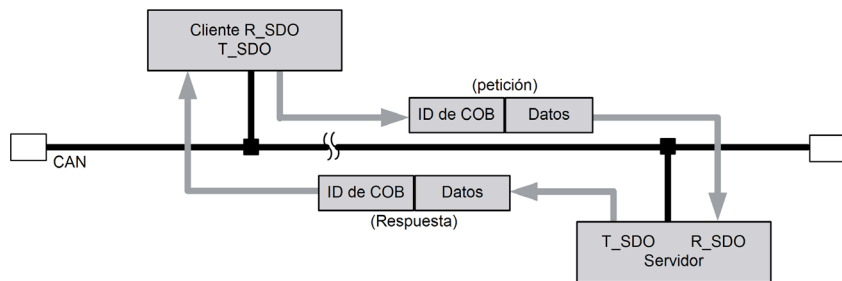
## Protocolo SDO

### Protocolo SDO (Objetos de datos de servicio)

Los objetos de datos de servicio (SDO) permiten acceder a cualquier entrada del diccionario de objetos (OD) CANopen. Un SDO establece un canal de comunicación par a par entre dos dispositivos. Además, el protocolo SDO permite transferir cualquier cantidad de datos de forma segmentada. Por lo tanto, el protocolo SDO se utiliza principalmente para comunicarse con los datos de configuración.

Una conexión SDO entre dos dispositivos se establece configurando el servidor SDO relacionado respectivamente con el canal de cliente. La transmisión SDO es un servicio confirmado.

En la siguiente figura se representa el intercambio de SDO con petición y respuesta.

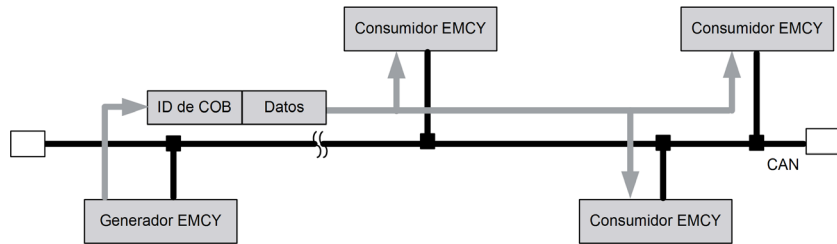


## Protocolo EMCY

### Protocolo de emergencia (EMCY)

El objeto de emergencia (EMCY) permite a los dispositivos indicar errores detectados en su interior. Al recibir esta señal, otros participantes de la red pueden evaluar la información recibida e iniciar acciones de contador apropiadas y específicas del fabricante.

En la siguiente figura se representa el mensaje de error detectado a través de los objetos EMCY.



### Mensajes EMCY

Los mensajes EMCY recibidos de otros dispositivos CANopen se almacenan en una tabla de almacenamiento EMCY. En cada generador EMCY, si no se recibe ningún mensaje EMCY, o siempre que el mensaje EMCY más reciente haya indicado que *no hay ningún error*, los consumidores EMCY consideran que no se ha detectado ningún error interno en el generador EMCY.



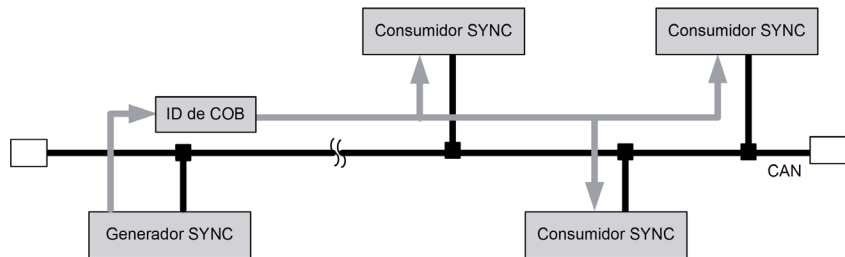
## Protocolo SYNC

### Protocolo síncrono (SYNC)

El comportamiento de red síncrono se puede conseguir con el protocolo SYNC. El mensaje SYNC transmitido cíclicamente indica a los consumidores que inicien su comportamiento específico de aplicación, que se asocia a la recepción del mensaje SYNC.

Una vez recibidos los datos, el PDO síncrono considera que estos son válidos cuando recibe el mensaje SYNC.

En la figura siguiente se representa la transmisión de mensajes SYNC del protocolo SYNC.

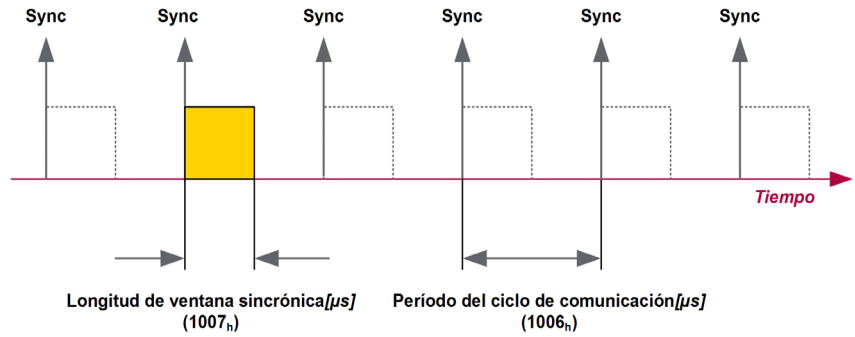


### Variables de control para el comportamiento SYNC

El período de tiempo entre dos mensajes SYNC consecutivos se denomina *período del ciclo de comunicación* y se puede ajustar en el diccionario de objetos del productor SYNC (índice de objeto 1006 hex).

El PDO de transmisión síncrona se transmite en una ventana de tiempo específica después de la recepción del mensaje SYNC. Esta ventana de tiempo se denomina *longitud de ventana síncrona* y se puede configurar en el diccionario de objetos de todos los dispositivos que tienen que transmitir PDO síncronos (índice de objeto 1007 hex).

En la figura siguiente se representa la longitud de ventana síncrona y el período del ciclo de comunicación.



## Protocolo de control de errores

### Protocolo de control de errores

Los servicios heartbeat o de guardia son servicios de gestión de redes (NMT) que se utilizan para comprobar la presencia de participantes de red y conocer su estado.

En CANopen existen dos tipos de protocolos de control de errores:

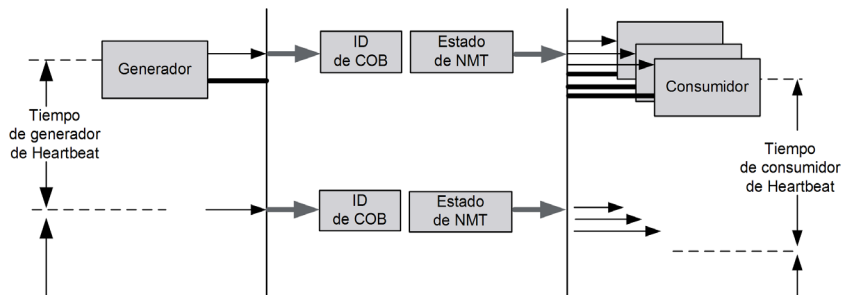
- Heartbeat
- Guardia de nodo/de vida

Dado que el protocolo de heartbeat es el más flexible y funciona sin una petición de transmisión remota (RTR), se recomienda utilizarlo si recibe soporte de los dispositivos CANopen.

### Heartbeat

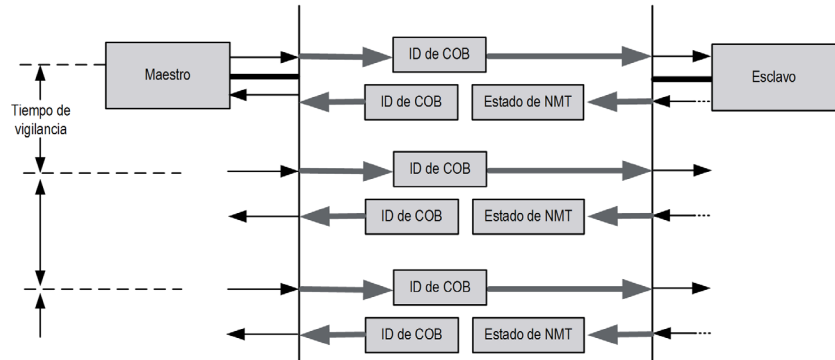
Un dispositivo CANopen transmite su heartbeat cíclicamente. El tiempo de ciclo se denomina *tiempo de generador de heartbeat*. Al recibir el mensaje de heartbeat, los consumidores de heartbeat (por ejemplo, el administrador CANopen) evalúan si el dispositivo relacionado todavía funciona correctamente y si permanece en el estado NMT correcto. El período de tiempo en el que debe recibirse al menos un mensaje de heartbeat del productor de heartbeat relacionado se configura en los consumidores (tiempo de consumidor de heartbeat).

En la siguiente figura se representa el heartbeat con intervalos de tiempo del protocolo de control de errores.



### Guardia de nodo/de vida

El administrador CANopen solicita el mensaje de control de errores a través de una petición de transmisión remota (RTR). El dispositivo que debe guardarse responde con una trama de datos CAN que indica el estado de NMT actual.



**Maestro:** el maestro monitoriza el estado del esclavo NMT (guardia de nodo)

**Esclavo:** el esclavo monitoriza los signos de *vida* (Guardia de vida) del maestro

---

## Introducción

### Métodos de acceso a la red CANopen

La interfaz y el perfil de dispositivo CANopen para los controladores programables (CiA405) IEC 61131-3 describen dos métodos de acceso a la red CANopen desde el controlador:

- Variables de red
- Bloques de funciones

### Variables de red

Las variables de red generalmente se asignan en PDO para ser recibidas o transmitidas. En el diccionario de objetos, se puede acceder a las variables IEC 61131-3 en un rango de índices definido.

### Bloques de funciones

El perfil también define algunos bloques de funciones específicos de CANopen, por ejemplo, los servicios SDO, NMT y de comunicación de emergencias.

### Biblioteca CAA CiA 405

La biblioteca CAA CiA 405 ofrece un conjunto de bloques de funciones para cumplir los requisitos de CiA405 para el acceso a la red CANopen desde la aplicación (programa IEC61131-3) del controlador (maestro CANopen). La biblioteca se declara automáticamente en el administrador de bibliotecas del controlador cuando se añade un administrador CANopen a una interfaz de bus CAN del controlador.

En la biblioteca, los bloques de funciones se organizan de la siguiente manera:

- Bloques de funciones de *gestión de red*:
  - CIA405.NMT: para controlar los estados NMT de los dispositivos CANopen.
  - CIA405.RECV\_EMCY: para explorar los almacenamientos EMCY de todos los dispositivos.
  - CIA405.RECV\_EMCY\_DEV: para obtener el último mensaje EMCY almacenado de un dispositivo especificado.

- **Bloque de funciones del ID del nodo:**
  - `CIA405.GET_LOCAL_NODE_ID`: para obtener el ID de nodo de administrador CANopen del controlador.
- **Bloques de funciones de estado de consulta:**
  - `CIA405.GET_CANOPEN_KERNEL_STATE`: para obtener el estado actual del kernel CANopen.
  - `CIA405.GET_STATE`: para obtener el estado actual de un dispositivo especificado.
- **Bloques de funciones de acceso SDO:**
  - `CIA405.SDO_READ`: para leer objetos de cualquier tamaño de un dispositivo especificado.
  - `CIA405.SDO_READ4`: para leer objetos de hasta 4 bytes de un dispositivo especificado.
  - `CIA405.SDO_WRITE`: para escribir objetos de cualquier tamaño de un dispositivo especificado.
  - `CIA405.SDO_WRITE4`: para escribir objetos de hasta 4 bytes de un dispositivo especificado.

**NOTA:** para el control de las unidades Altivar y de las unidades de movimiento Lexium en CANopen, utilice los bloques de funciones PLCopen dedicados (consulte la guías de la biblioteca *Altivar*, *Lexium integrada* y *Lexium*).

## Namespace

El espacio de nombres de la biblioteca CAA CiA 405 es `CIA405`. Los bloques de funciones de la biblioteca, las variables y las constantes deben utilizarse con el espacio de nombres de la biblioteca.

Ejemplo de instancia de bloque de funciones y declaración de variables:

```
VAR
ReadObject: CIA405.SDO_READ;
SDOabort_info: CIA405.SDO_ERROR;
END_VAR
```

---

# Descripciones de bloques de funciones



---

## Descripción general

Esta sección ofrece una descripción general de los bloques de funciones de la biblioteca CAA CiA 405 que se utilizan para administrar y monitorizar la red y los dispositivos CANopen de las aplicaciones de controlador.

## Contenido de esta parte

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
3	E/S y comportamiento comunes de los bloques de funciones	25
4	Bloques de funciones de gestión de red	31
5	Bloques de funciones de ID del nodo	39
6	Bloques de funciones de estado de consulta	41
7	Bloques de funciones de acceso SDO	45





---

## E/S y comportamiento comunes de los bloques de funciones

# 3

---

### 3.1 E/S y comportamiento comunes

---

#### Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Descripción de E/S común	26
Códigos de error detectados del núcleo CANopen	28
Diagramas de ejecución de bloques de funciones	29

## Descripción de E/S común

### Introducción

En este apartado se describe la gestión y las ejecuciones genéricas de los bloques de funciones de la biblioteca CAA CiA 405 utilizando el bloque de funciones CIA405.RECV\_EMCY como ejemplo. La E/S común a todos los bloques de funciones se describe a continuación. Se hereda del bloque de funciones oculto e interno CiA405Base.

### Representación gráfica

Los parámetros comunes a todos los bloques de funciones de la biblioteca CAA CiA 405 se resaltan en el siguiente gráfico.



### Descripción de variables de entrada comunes

En la siguiente tabla se describen los parámetros de entrada comunes a todos los bloques de funciones de la biblioteca CAA CiA 405.

Entrada	Tipo de datos	Descripción
NETWORK	USINT	Canal CAN en el que debe ejecutarse el servicio solicitado. 1 (valor predeterminado) = primera interfaz de bus CAN 2 = segunda interfaz de bus CAN (si existe)
ENABLE	BOOL	Permite la ejecución del bloque de funciones. En el flanco ascendente: se inicia la ejecución. En el flanco descendente: se cancela la ejecución si no se ha completado. De lo contrario, los datos de salida se restablecen a 0.
TIMEOUT	UDINT	Tiempo de ejecución máximo en ms. Si se alcanza el timeout antes de que haya una respuesta disponible, se cancela la ejecución en error de timeout. 0 (ajuste de fábrica) = el timeout está deshabilitado. 1...65535 = valor de timeout en ms.

**Descripción de variables de salida comunes**

En la siguiente tabla se describen las variables de salida comunes a todos los bloques de funciones de la biblioteca CAA CiA 405.

Salida	Tipo de datos	Descripción
CONFIRM	BOOL	TRUE si la ejecución se completa correctamente.
ERROR	CANOPEN_KERNEL_ERROR (USINT)	Contiene el código de error detectado en la ejecución, devuelto por el kernel CANopen. 00 hex = no se ha detectado ningún error de ejecución. 01 hex = error detectado, pero el código está disponible en otra salida del bloque de funciones (no en el kernel CANopen). 02...FF hex = código de error detectado en el kernel CANopen.

## Códigos de error detectados del núcleo CANopen

### Descripción

La biblioteca ofrece una lista de variables globales `CIA405.CANOPEN_KERNEL_ERROR_CODES` con variables de códigos de error predefinidos.

### Código de error detectado

A continuación se indican los códigos de error con cada variable global asociada y la descripción.

Código de error detectado	Descripción
<code>CANOPEN_KERNEL_NO_ERROR = 00 hex</code>	El kernel CANopen no ha detectado ningún error.
<code>CANOPEN_KERNEL_OTHER_ERROR = 01 hex</code>	Si la salida <code>ERROR = 01 hex</code> , se habrá detectado un error, y si hay otra salida de código de error en el bloque de funciones, esta salida contendrá información más específica. <b>Ejemplos:</b> <ul style="list-style-type: none"> <li>● Bloques de funciones <code>CIA405.SDO_READ</code> y <code>CIA405.SDO_WRITE</code>: la salida <code>ERRORINFO</code> incluye el contenido del mensaje de cancelación de SDO.</li> <li>● Bloques de funciones <code>CIA405.RECV_EMCY</code> y <code>CIA405.RECV_EMCY_DEV</code>: la salida <code>ERRORINFO</code> incluye el contenido del mensaje EMCY recibido. Cuando en la salida <code>ERRORINFO</code> hay un mensaje de emergencia disponible, la salida <code>ERROR = 1</code> y la salida <code>CONFIRM = 0</code>.</li> <li>● Bloque de funciones <code>CIA405.GET_CANOPEN_KERNEL_STATE</code>: el valor hexadecimal <code>01</code> nunca se entrega porque no hay ninguna otra salida de código de error.</li> </ul>
<code>CANOPEN_KERNEL_DATA_OVERFLOW = 02 hex</code>	Desborde en los búferes de envío o los búferes de recepción de los objetos CANopen.
<code>CANOPEN_KERNEL_TIMEOUT = 03 hex</code>	Se ha producido un timeout de ejecución del bloque de funciones.
<code>CANOPEN_KERNEL_CANBUS_OFF = 10 hex</code>	El nodo CANopen está desconectado del bus CAN.
<code>CANOPEN_KERNEL_CAN_ERROR_PASSIVE = 11 hex</code>	El nodo CANopen tiene un error de estado pasivo: el nodo se puede comunicar, pero no puede enviar un indicador de error activo si se detecta un error.
<code>CANOPEN_INTERNAL_FB_ERROR = 21 hex</code>	Código de error específico de fabricante. <b>NOTA:</b> <code>21hex</code> a <code>FFhex</code> son valores dedicados de fabricante de dispositivos.

## Diagramas de ejecución de bloques de funciones

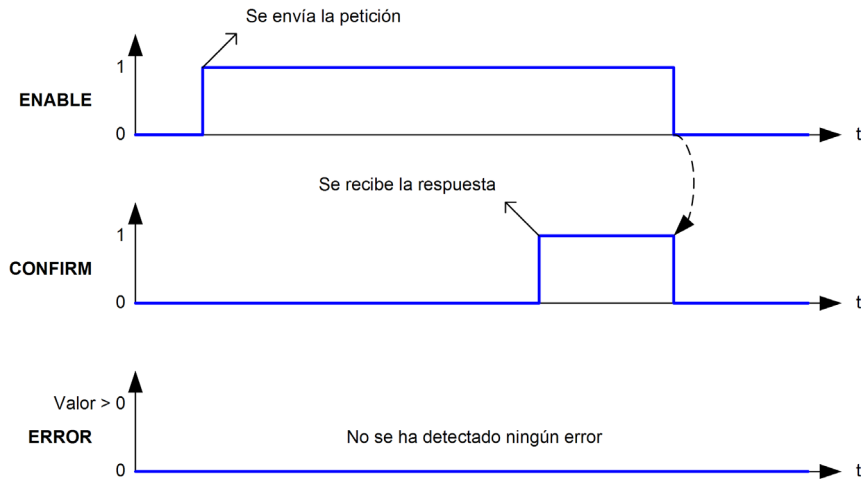
### Comportamiento de las señales de control

A continuación se describen los tres comportamientos típicos de las señales de control `ENABLE`, `CONFIRM` y `ERROR`:

- La ejecución ha terminado sin que se haya detectado ningún error
- La aplicación ha cancelado la ejecución
- La ejecución se ha cancelado o ha terminado al detectarse un error

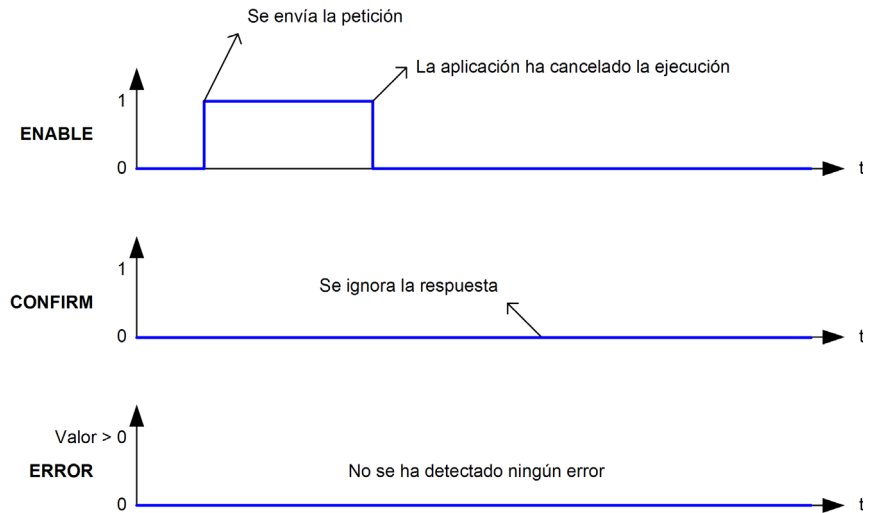
### La ejecución ha terminado sin que se haya detectado ningún error

En cuanto la respuesta a la petición actual está disponible, y si no se ha detectado ningún error, la salida `CONFIRM` se establece en `TRUE` y sigue así hasta que se llama al bloque de funciones con la salida `ENABLE` como `TRUE`. Cuando se llama al bloque de funciones, si `ENABLE` está restablecida en `FALSE`, `CONFIRM` se restablece en `FALSE` y el bloque de funciones está preparado para un nuevo inicio de ejecución.



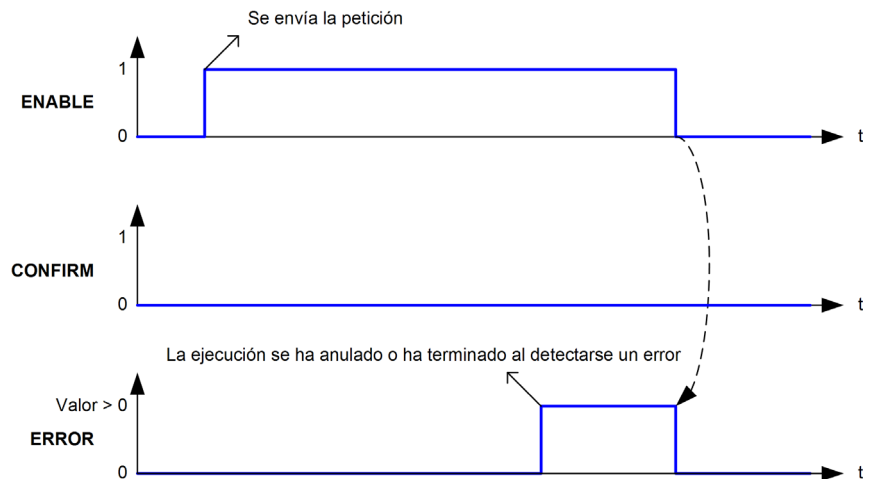
### La aplicación ha cancelado la ejecución

Si se llama al bloque de funciones y `ENABLE` está restablecida en `FALSE` antes de que haya finalizado la ejecución actual, se cancela la ejecución del bloque de funciones. Se ignorará una posible respuesta disponible o la respuesta que llegue posteriormente a la petición cancelada.



### La ejecución se ha cancelado o ha terminado al detectarse un error

Una vez cancelada la ejecución actual mediante un mensaje de cancelación de SDO recibido, o si dicha ejecución ha finalizado al detectarse un error, la salida **ERROR** se establece en un valor distinto al cero 0 (consulte Códigos de error detectados en el kernel CANopen (véase página 28) para obtener más información sobre los códigos de error detectados). La salida **ERROR** se restablece en 0 y el bloque de funciones está preparado para un nuevo inicio de ejecución cuando, al llamar al bloque de funciones, la entrada **ENABLE** está restablecida a FALSE.



---

# Bloques de funciones de gestión de red

# 4

---

## Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
4.1	CIA405.NMT: Gestión del estado de NMT de dispositivos	32
4.2	CIA405.RECV_EM CY: Exploración de mensajes EMCY	35
4.3	CIA405.RECV_EM CY_DEV: Obtener mensaje EMCY de dispositivo	37

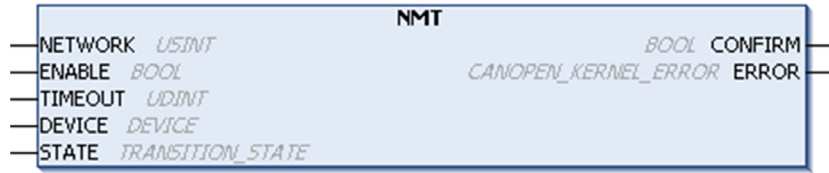
## 4.1 CIA405.NMT: Gestión del estado de NMT de dispositivos

### Administración del estado NMT para dispositivos

#### Descripción del bloque de funciones

El bloque de funciones CIA405.NMT permite el control del estado de NMT de los dispositivos CANOpen de la aplicación de controlador. Realiza la petición de servicio NMT a un dispositivo de destino CANOpen para que realice la transición de estado de NMT solicitada.

#### Representación gráfica



#### Descripción de variables de entrada específicas

Entrada	Tipo de datos	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo de destino CANOpen. 0 (valor predeterminado) = todos los dispositivos esclavos NMT 1...127 = ID de nodo del dispositivo de destino
STATE	TRANSITION_STATE. Consulte a continuación para obtener más información.	Transición de estado de NMT solicitada.

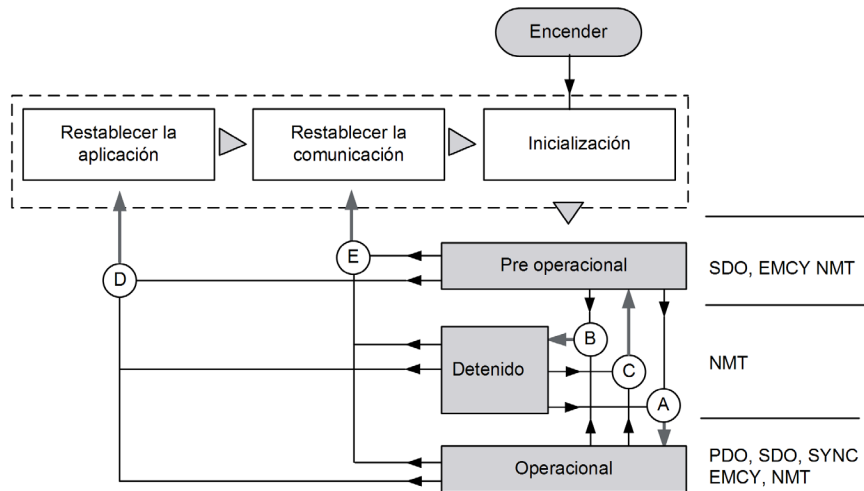
**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (véase página 25).



### CIA405.TRANSITION\_STATE ENUM

La máquina de estado de NMT describe la inicialización y el estado de un esclavo NMT en la operación principal.

La figura que aparece a continuación muestra los estados de NMT, los objetos de comunicación disponibles asociados (PDO, SDO, SYNC, EMCY y NMT) y 5 transiciones de estado (de la A a la E).



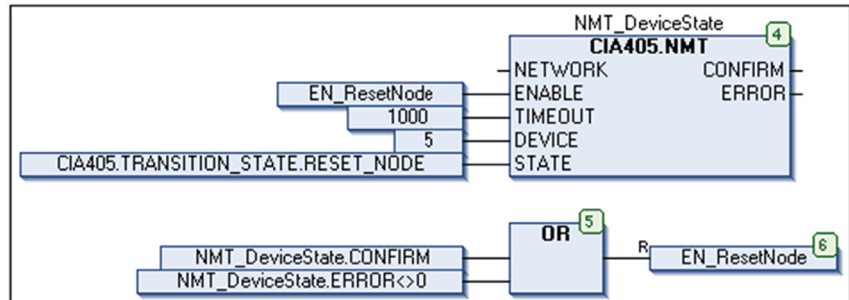
El tipo enumerado CIA405.TRANSITION\_STATE contiene los comandos de transición de estado de NMT, que se describen en la tabla siguiente.

Enumerador	Valor (hexadecimal)	Descripción
STOP_REMOTE_NODE	0004	Cambie al estado de <i>detención</i> . (Transición B)
START_REMOTE_NODE	0005	Cambie al estado <i>operacional</i> . (Transición A)
RESET_NODE	0006	Cambie al estado de <i>restablecimiento de aplicaciones</i> . Cargue los datos guardados de los perfiles de dispositivo y cambie automáticamente al estado <i>pre operacional</i> a través de <i>Restablecer comunicación</i> . (Transición D)

Enumerador	Valor (hexadecimal)	Descripción
RESET_COMMUNICATION	0007	Cambie al estado de <i>restablecimiento de comunicación</i> . Cargue los datos almacenados del perfil de comunicación y cambie automáticamente al estado <i>pre operacional</i> . (Transición E)
ENTER_PRE_OPERATIONAL	007F	Cambie al estado <i>pre operacional</i> . (Transición C)
ALL_EXCEPT_NMT_AND_SENDER	0800	No implementado (parámetro no válido)

### Ejemplo

El ejemplo siguiente muestra cómo enviar un comando Restablecer nodo al nodo 5 CANopen conectado a la primera interfaz de bus CAN con un timeout de 1s (1000 ms). El comando se envía cuando la variable booleana `EN_ResetNode` se establece como TRUE (debe hacerlo el usuario online o la aplicación). El comando `EN_ResetNode` se restablece en FALSE cuando la ejecución ha finalizado correctamente (salida `CONFIRM = TRUE`) o se ha detectado un error (salida `ERROR <> 0`).



## 4.2 CIA405.RECV\_EMCY: Exploración de mensajes EMCY

### Exploración de mensajes EMCY

#### Descripción del bloque de funciones

El bloque de funciones CIA405.RECV\_EMCY explora los almacenamientos de mensajes de emergencia (EMCY) en un bucle en todos los dispositivos CANopen existentes y devuelve los mensajes EMCY encontrados.

Cuando el bloque de funciones está habilitado, se inicia la exploración de almacenamientos EMCY desde el punto de detención de la exploración anterior.

- Si se encuentra un mensaje EMCY distinto a *Sin errores*, la exploración termina en este punto y el bloque de funciones devuelve el mensaje EMCY y el ID de nodo del dispositivo asociado.
- Si se encuentra un mensaje EMCY *Sin errores* que anteriormente era distinto de *Sin errores*, la exploración termina en este punto y el bloque de funciones devuelve un mensaje EMCY = 0 y el ID de nodo del dispositivo asociado.
- Si no hay ningún mensaje EMCY distinto de *Sin errores* y no se encuentra ningún mensaje EMCY nuevo *Sin errores* durante una exploración completa, el bloque de funciones devuelve un mensaje EMCY = 0 y el ID de nodo del dispositivo = 0.

#### Representación gráfica



### Descripción de variables de salida específicas

Salida	Tipo de datos	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo CANopen asociado al mensaje EMCY devuelto. 0 = Ningún mensaje EMCY encontrado 1...127 = ID de nodo del dispositivo
ERRORINFO	CS.EMCY_ERROR. Consulte la estructura CS.EMCY_ERROR (véase página 38) para obtener información.	Último mensaje EMCY recibido del dispositivo CANopen con el ID de nodo <i>DEVICE</i> .

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (véase página 25).

## 4.3 CIA405.RECV\_EMCY\_DEV: Obtener mensaje EMCY de dispositivo

### Obtener mensaje EMCY del dispositivo

#### Descripción del bloque de funciones

El bloque de funciones CIA405.RECV\_EMCY\_DEV devuelve el último mensaje de emergencia (EMCY) almacenado, recibido de un dispositivo CANopen especificado.

#### Representación gráfica



#### Descripción de variables de entrada específicas

Entrada	Tipo	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo CANopen que debe comprobarse. 0 (valor predeterminado) = dispositivo local (controlador) 1...127 = ID de nodo del dispositivo

#### Descripción de variables de salida específicas

Salida	Tipo	Descripción
ERRORINFO	CS.EMCY_ERROR. (Consulte a continuación para obtener más información.)	Último mensaje EMCY recibido del dispositivo CANopen con el ID de nodo <i>DEVICE</i> .

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (véase página 25).

### Estructura CS.EMCY\_ERROR

CS.EMCY\_ERROR es la estructura asociada al contenido de los mensajes EMCY. (Consulte la guía de dispositivos CANopen para obtener más información sobre el contenido específico de los mensajes EMCY.) CS.EMCY\_ERROR contiene los siguientes elementos.

Elemento	Tipo	Descripción
EMCY_ERROR_CODE	WORD	Código de error del mensaje EMCY
ERROR_REGISTER	BYTE	Registro de errores (campo de bit) del mensaje EMCY
ERROR_FIELD	ARRAY [1...5] OF BYTE	Campo de error específico de fabricante de dispositivos del mensaje EMCY

**NOTA:** esta estructura se declara en la biblioteca de pilas CAA CANopen (espacio de nombres = CS). Esta es la razón de que se utilice el nombre completo de la estructura (<espacio de nombres>.<tipo de datos>). El espacio de nombres de la biblioteca CAA CiA 405 es CIA405.

---

## Bloques de funciones de ID del nodo

# 5

---

### 5.1 CIA405.GET\_LOCAL\_NODE\_ID: Obtener ID de nodo CANopen del controlador

---

#### Obtener ID de nodo CANopen del controlador

##### Descripción del bloque de funciones

El bloque de funciones `CIA405.GET_LOCAL_NODE_ID` devuelve el ID de nodo CANopen del controlador en una interfaz de bus CAN especificada.

##### Representación gráfica



##### Descripción de variables de salida específicas

Salida	Tipo	Descripción
<code>DEVICE</code>	<code>DEVICE (USINT)</code>	ID de nodo CANopen del controlador. 0 = no válido 1...127 = ID de nodo del controlador

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (véase página 25).





---

## Bloques de funciones de estado de consulta

# 6

---

### Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
6.1	CIA405.GET_CANOPEN_KERNEL_STATE: Obtener el estado del kernel CANopen	42
6.2	CIA405.GET_STATE: Obtener el estado del dispositivo CANopen	43

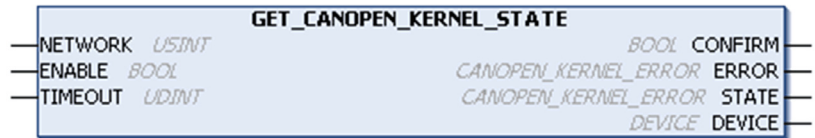
## 6.1 CIA405.GET\_CANOPEN\_KERNEL\_STATE: Obtener el estado del kernel CANopen

### Obtener estado del núcleo CANopen

#### Descripción del bloque de funciones

El bloque de funciones CIA405.GET\_CANOPEN\_KERNEL\_STATE devuelve el estado actual del kernel CANopen del controlador.

#### Representación gráfica



#### Descripción de variables de salida específicas

Salida	Tipo	Descripción
STATE	CANOPEN_KERNEL_ERROR (Consulte Códigos de error detectados en el kernel CANopen (véase página 28) para obtener información.)	Estado actual del kernel CANopen del controlador
DEVICE	DEVICE (USINT)	ID de nodo CANopen del controlador. 0 = no válido 1...127 = ID de nodo del controlador

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (véase página 25).

## 6.2 CIA405.GET\_STATE: Obtener el estado del dispositivo CANopen

### Obtener estado del dispositivo CANopen

#### Descripción del bloque de funciones

El bloque de funciones CIA405.GET\_STATE devuelve el estado de NMT actual del dispositivo CANopen especificado si el heartbeat o la guardia de nodo están activos.

#### Representación gráfica



#### Descripción de variables de entrada específicas

Entrada	Tipo	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo CANopen que debe comprobarse. 0 (valor predeterminado) = dispositivo local (controlador) 1...127 = ID de nodo del dispositivo

#### Descripción de variables de salida específicas

Salida	Tipo	Descripción
STATE	DEVICE_STATE (Consulte a continuación para obtener más información.)	Estado de NMT del dispositivo CANopen.

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (véase página 25).

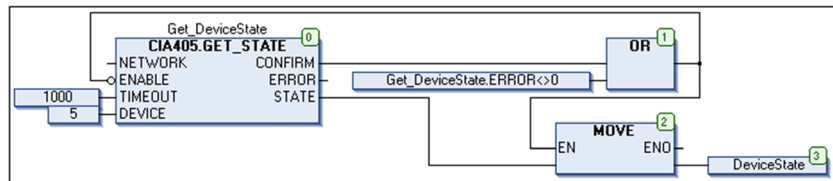
### CIA405.DEVICE\_STATE ENUM

El tipo enumerado CIA405.DEVICE\_STATE contiene la lista de estados de NMT del dispositivo CANopen.

Enumerador	Valor	Descripción
INIT	0	Estado de inicialización
RESET_COMM	1	Estado de restablecimiento de comunicación
RESET_APP	2	Estado de restablecimiento de aplicación
PRE_OPERATIONAL	3	Estado pre operacional
STOPPED	4	Estado detenido
OPERATIONAL	5	Estado operacional
UNKNOWN	6	Estado de NMT desconocido. La guardia de nodo o heartbeat no está activa para el dispositivo seleccionado o el controlador no es el consumidor de heartbeat.
NOT_AVAIL	7	Estado de NMT no disponible. La guardia de nodo o heartbeat está activa para el dispositivo seleccionado, pero el dispositivo no informa de su estado de NMT correctamente antes del timeout.

### Ejemplo

El ejemplo siguiente muestra cómo obtener el estado del nodo 5 CANopen conectado a la primera interfaz de bus CAN con un timeout de 1s (1000 ms). La función CIA405.GET\_STATE se ejecuta automáticamente para una lectura de estado continua. El estado de NMT del dispositivo se copia en la variable DeviceState de tipo CIA405.DEVICE\_STATE.



---

# Bloques de funciones de acceso SDO

# 7

---

## Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
7.1	CIA405.SDO_READ: Leer objetos CANopen de cualquier tamaño	46
7.2	CIA405.SDO_READ4: Leer objetos CANopen de hasta 4 bytes	49
7.3	CIA405.SDO_WRITE: Escribir objetos CANopen de cualquier tamaño	52
7.4	CIA405.SDO_WRITE4: Escribir objetos CANopen de hasta 4 bytes	57

## 7.1 CIA405.SDO\_READ: Leer objetos CANOpen de cualquier tamaño

### Leer objetos CANOpen de cualquier tamaño

#### Descripción del bloque de funciones

El bloque de funciones `CIA405.SDO_READ` se utiliza para leer objetos CANOpen de cualquier tamaño de un dispositivo especificado a través de mensajes SDO.

Estos parámetros específicos se deben pasar al bloque de funciones:

- ID de nodo del dispositivo
- Canal de cliente/servidor SDO (de manera predeterminada, sólo se define un canal)
- Índice/subíndice de objeto CANOpen
- Puntero al búfer de datos en el que se almacenarán valores de objeto
- Tamaño del búfer de datos

El bloque de funciones devuelve el tamaño del objeto de lectura si la lectura ha terminado con éxito. Los datos están disponibles en el búfer de datos.

**NOTA:** si el tamaño del objeto que se va a leer es inferior o igual a 4 bytes, utilice el bloque de funciones `CIA405.SDO_READ4`.

#### Representación gráfica



**Descripción de variables de entrada específicas**

Entrada	Tipo	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo CANopen. 0 (valor predeterminado) = dispositivo local (controlador) 1...127 = ID de nodo del dispositivo
CHANNEL	USINT	Número de canal SDO. Valor predeterminado = 1
INDEX	WORD	Índice de objeto. Rango: 0000 hex ... FFFF hex
SUBINDEX	BYTE	Subíndice de objeto. Rango: 00 hex ... FF hex
DATA	POINTER TO BYTE	Dirección del búfer de datos que recibe la lectura de datos del objeto de dispositivo. La función estándar ADR debe utilizarse para definir el puntero asociado.

**Descripción de variables de salida específicas**

Salida	Tipo	Descripción
ERRORINFO	SDO_ERROR (UDINT)	Cuando la salida ERROR = 1, devuelve el contenido del mensaje de cancelación de SDO (tamaño de 4 bytes).

**Descripción de variables de entrada/salida específicas**

Entrada/Salida	Tipo	Descripción
DATALENGTH	UINT	Como entrada: tamaño del búfer de datos (en bytes). <b>NOTA:</b> Para asegurarse de que la entrada DATALENGTH se inicializa correctamente (en la ejecución del bloque de funciones se inicia en el flanco ascendente de la entrada ENABLE) con el tamaño de búfer de datos, utilice la función estándar SIZEOF.  Como salida: tamaño de la lectura de objetos (en bytes).

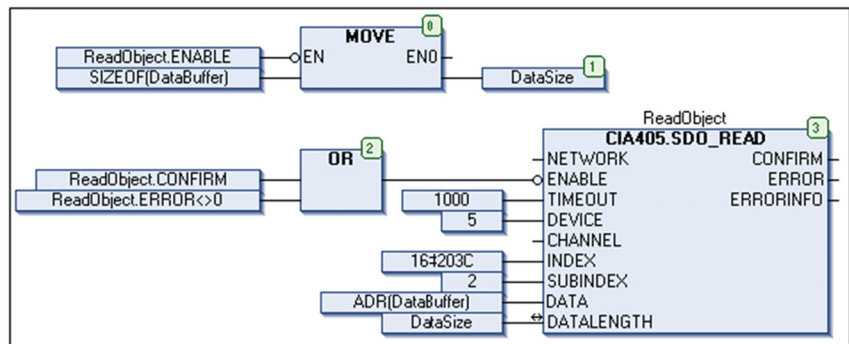
**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (*véase página 25*).

## Ejemplo

El ejemplo siguiente muestra cómo leer el índice de objeto 203C hex/subíndice 02 hex del nodo CANopen 5 conectado a la primera interfaz de bus CAN con un timeout de 1s (1000ms). La instancia del bloque de funciones CIA405.SDO\_READ (ReadObject) se ejecuta automáticamente para una lectura continua.

La variable `DataSize` (tipo `UINT`):

- Se inicializa con el tamaño del búfer de datos (`DataBuffer`: matriz de N bytes) cuando la entrada `ENABLE` del bloque de funciones es `FALSE`, antes de iniciar la siguiente ejecución.
- Contiene el tamaño de la lectura de datos (en bytes) en el flanco ascendente de la salida del bloque de funciones `CONFIRM` (el ejemplo no muestra cómo extraer el valor del búfer de datos ni cómo gestionar las detecciones de errores).





## 7.2 CIA405 .SDO\_READ4: Leer objetos CANopen de hasta 4 bytes

### Leer objetos CANopen de hasta 4 bytes

#### Descripción del bloque de funciones

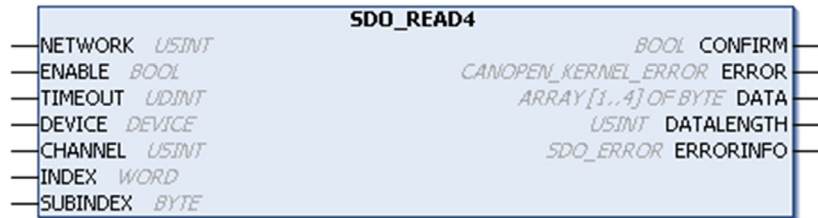
El bloque de funciones `CIA405.SDO_READ4` se utiliza para leer objetos CANopen de un tamaño de hasta 4 bytes de un dispositivo especificado a través de un mensaje SDO.

Estos parámetros específicos se deben pasar al bloque de funciones:

- ID de nodo del dispositivo
- Canal de cliente/servidor SDO (de manera predeterminada, sólo se define un canal)
- Índice/subíndice de objeto CANopen

El bloque de funciones devuelve el tamaño del objeto de lectura si la lectura se ha realizado con éxito. Los datos están disponibles en una matriz de 4 bytes.

#### Representación gráfica



#### Descripción de variables de entrada específicas

Entrada	Tipo	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo CANopen. 0 (valor predeterminado) = dispositivo local (controlador) 1...127 = ID de nodo del dispositivo
CHANNEL	USINT	Número de canal SDO. Valor predeterminado = 1

Entrada	Tipo	Descripción
INDEX	WORD	Índice de objeto. Rango: 0000 hex ... FFFF hex
SUBINDEX	BYTE	Subíndice de objeto. Rango: 00 hex ... FF hex

### Descripción de variables de salida específicas

Salida	Tipo	Descripción
DATA	ARRAY [1...4]OF BYTE	Matriz de datos que recibe la lectura de datos del objeto de dispositivo.
DATALENGTH	USINT	Tamaño del objeto de lectura (en bytes).
ERRORINFO	SDO_ERROR (UDINT)	Cuando la salida <b>ERROR = 1</b> , devuelve el contenido del mensaje de cancelación de SDO (tamaño de 4 bytes).

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (*véase página 25*).

En la siguiente tabla se muestra el tamaño del objeto frente al contenido de la matriz de DATOS.

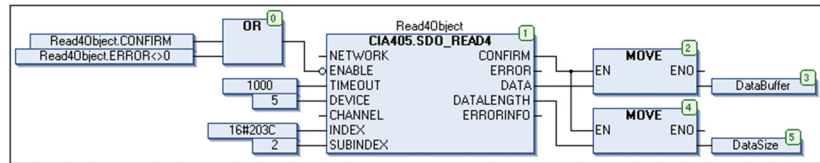
Tamaño del objeto Ejemplo	DATALENGTH	DATA(1)	DATA(2)	DATA(3)	DATA(4)
1 byte <b>01 hex</b>	1	<b>01 hex</b>	no significativo	no significativo	no significativo
2 bytes <b>01 23 hex</b>	2	LSB <b>23 hex</b>	MSB <b>01 hex</b>	no significativo	no significativo
3 bytes <b>01 23 45 hex</b>	3	LSB <b>45 hex</b>	<b>23 hex</b>	MSB <b>01 hex</b>	no significativo
4 bytes <b>01 23 45 67 hex</b>	4	LSB <b>67 hex</b>	<b>45 hex</b>	<b>23 hex</b>	MSB <b>01 hex</b>

LSB = byte poco significativo

MSB = byte más significativo

## Ejemplo

El ejemplo siguiente muestra cómo leer el índice de objeto 203C hex/subíndice 02 hex del nodo CANopen 5 conectado a la primera interfaz de bus CAN con un timeout de 1 s (1.000 ms). La instancia del bloque de funciones CIA405.SDO\_READ4 (Read4Object) se ejecuta automáticamente para una lectura continua. La variable `DataBuffer` (matriz de 4 bytes) contiene el valor de los últimos datos leídos. La variable `DataSize` (tipo `USINT`) contiene el tamaño (4 bytes como máximo) de los últimos datos leídos. En el ejemplo no se muestra cómo gestionar las detecciones de errores.



## 7.3 CIA405.SDO\_WRITE: Escribir objetos CANopen de cualquier tamaño

### Escribir objetos CANopen de cualquier tamaño

#### Descripción del bloque de funciones

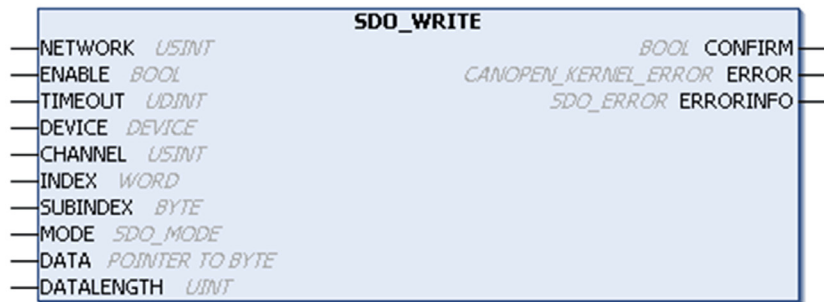
El bloque de funciones `CIA405.SDO_WRITE` se utiliza para escribir objetos CANopen de cualquier tamaño de un dispositivo especificado a través de mensajes SDO.

Estos parámetros específicos se deben pasar al bloque de funciones:

- ID de nodo del dispositivo
- Canal de cliente/servidor SDO (de manera predeterminada, sólo se define un canal)
- Índice/subíndice de objeto CANopen
- Modalidad SDO (define la modalidad de transmisión de datos). Consulte `SDO_MODE` ENUM para obtener más información (*véase página 54*)
- Puntero al búfer de datos en el que se almacenan los valores de objeto que se van a escribir
- Número de bytes que se van a escribir

**NOTA:** Si el tamaño del objeto que se va a escribir es inferior o igual a 4 bytes, utilice el bloque de funciones `CIA405.SDO_WRITE4`.

#### Representación gráfica



### Descripción de variables de entrada específicas

Entrada	Tipo	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo CANopen. 0 (valor predeterminado) = dispositivo local (controlador) 1...127 = ID de nodo del dispositivo
CHANNEL	USINT	Número de canal SDO. Valor predeterminado = 1
INDEX	WORD	Índice de objeto. Rango: 0000 hex ... FFFF hex
SUBINDEX	BYTE	Subíndice de objeto. Rango: 00 hex ... FF hex
MODE	SDO_MODE (Consulte a continuación para obtener más información.)	Modalidad de transmisión de datos. 0 (valor predeterminado) = AUTO para la selección de modalidad automática
DATA	POINTER TO BYTE	Dirección del búfer de datos en el que se almacenan los valores de objeto que se van a escribir. La función estándar ADR debe utilizarse para definir el puntero asociado.
DATALength	UINT	Tamaño del objeto de escritura (en bytes). <b>NOTA:</b> utilice la función estándar <code>SIZEOF</code> para definir el tamaño del búfer de datos.

### Descripción de variables de salida específicas

Salida	Tipo	Descripción
ERRORINFO	SDO_ERROR (UDINT)	Cuando la salida <code>ERROR = 1</code> , devuelve el contenido del mensaje de cancelación de SDO (tamaño de 4 bytes).

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (*véase página 25*).

**CIA405.SDO\_MODE ENUM**

El tipo enumerado CIA405.SDO\_MODE contiene la lista de modalidades de transmisión SDO.

Enumerador	Valor	Descripción
AUTO	0	Selección de modalidad automática.
EXPEDITED	1	Modalidad de envío urgente de SDO para datos de hasta 4 bytes. Datos enviados en una petición SDO.
SEGMENTED	2	Modalidad segmentada SDO para datos de más de 4 bytes. Los datos se segmentan en segmentos de 7 bytes enviados por peticiones confirmadas SDO consecutivas.
BLOCK	3	Modalidad de bloque SDO para datos de más de 4 bytes. Los datos enviados por tramas consecutivas se segmentan en bloques de 7 bytes que no se confirman. El receptor envía una confirmación una vez recibidos todos los bloques. <b>NOTA:</b> Esta modalidad de transmisión es más rápida, pero es posible que su dispositivo no la admita, ya que esta modalidad se ha introducido recientemente en el perfil CANopen.

**Ejemplo**

Ejemplo de programa en lenguaje ST para leer y escribir datos en dispositivos CANopen:

Un ATV71 se declara en la dirección de nodo 1

Un ATV31 se declara en la dirección de nodo 2

Los datos que se deben escribir o leer son:

INDEX:=16#203C ,

SUBINDEX:=2 ,

El valor escrito (SDOwrite\_data) es una rampa de 0 a 500 incrementada en cada ciclo.

```
PROGRAM Read_Write_SDO
VAR
    WriteSDO_ATV71 : CIA405.SDO_WRITE;
    WriteSDO_ATV31 : CIA405.SDO_WRITE;
    ReadSDO_ATV31 : CIA405.SDO_READ;
    SDOwrite_data : UINT;
    SDOread_data : UINT;
    ReadSDO_datalength : UINT:=2;
```

```
start_test : BOOL;
END_VAR
IF start_test THEN
  WriteSDO_ATV31(
    NETWORK:= ,
    ENABLE:= NOT (WriteSDO_ATV31.CONFIRM OR
  (WriteSDO_ATV31.ERROR<>0)) ,
    TIMEOUT:= ,
    DEVICE:=2 ,
    CHANNEL:= ,
    INDEX:=16#203C ,
    SUBINDEX:=2 ,
    MODE:= ,
    DATA:=ADR(SDOWrite_data) ,
    DATALENGTH:=2 ,
    CONFIRM=> ,
    ERROR=> ,
    ERRORINFO=> );
  ReadSDO_ATV31(
    NETWORK:= ,
    ENABLE:=NOT (ReadSDO_ATV31.CONFIRM OR
  (ReadSDO_ATV31.ERROR<>0)) ,
    TIMEOUT:= ,
    DEVICE:=1 ,
    CHANNEL:= ,
    INDEX:=16#203C ,
    SUBINDEX:=2 ,
    DATA:=ADR(SDORead_data) ,
    DATALENGTH:=ReadSDO_datalength ,
    CONFIRM=> ,
    ERROR=> ,
    ERRORINFO=> );
  WriteSDO_ATV71(
```

```
NETWORK:= ,
ENABLE:= NOT (WriteSDO_ATV71.CONFIRM OR
(WriteSDO_ATV71.ERROR<>0)) ,
TIMEOUT:= ,
DEVICE:=1 ,
CHANNEL:= ,
INDEX:=16#203C ,
SUBINDEX:=2 ,
MODE:= ,
DATA:=ADR(SDOWrite_data) ,
DATALENGTH:=2 ,
CONFIRM=> ,
ERROR=> ,
ERRORINFO=> );
IF SDOWrite_data<500
THEN SDOWrite_data:=SDOWrite_data+1;
ELSE SDOWrite_data:=0;
END_IF
END_IF
```



## 7.4 CIA405.SDO\_WRITE4: Escribir objetos CANopen de hasta 4 bytes

### Escribir objetos CANopen de hasta 4 bytes

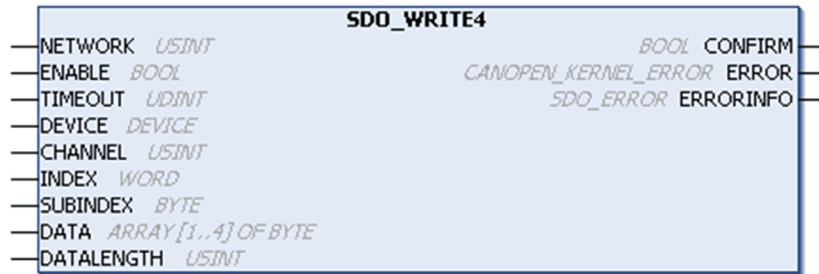
#### Descripción del bloque de funciones

El bloque de funciones CIA405.SDO\_WRITE4 se utiliza para escribir objetos CANopen de un tamaño de hasta 4 bytes de un dispositivo especificado a través de mensajes SDO.

Estos parámetros específicos se deben pasar al bloque de funciones:

- ID de nodo del dispositivo
- Canal de cliente/servidor SDO (de manera predeterminada, sólo se define un canal)
- Índice/subíndice de objeto CANopen
- Valores que se van a escribir
- Número de bytes que se van a escribir (tamaño del objeto)

#### Representación gráfica



#### Descripción de variables de entrada específicas

Entrada	Tipo	Descripción
DEVICE	DEVICE (USINT)	ID de nodo del dispositivo CANopen. 0 (valor predeterminado) = dispositivo local (controlador) 1...127 = ID de nodo del dispositivo
CHANNEL	USINT	Número de canal SDO. Valor predeterminado = 1

Entrada	Tipo	Descripción
INDEX	WORD	Índice de objeto. Rango: 0000 hex ... FFFF hex
SUBINDEX	BYTE	Subíndice de objeto. Rango: 00 hex ... FF hex
DATA	ARRAY[1...4] OF BYTE	Matriz de datos en la que se almacenan los valores de objetos que se van a escribir.
DATALENGTH	USINT	Tamaño del objeto de escritura (en bytes).

### Descripción de variables de salida específicas

Salida	Tipo	Descripción
ERRORINFO	SDO_ERROR (UDINT)	Cuando la salida ERROR = 1, devuelve el contenido del mensaje de cancelación de SDO (tamaño de 4 bytes).

**NOTA:** para ver las descripciones de E/S comunes y el comportamiento de ejecución del bloque de funciones, consulte el capítulo E/S y comportamiento comunes de los bloques de funciones (*véase página 25*).

En la siguiente tabla se muestra el tamaño del objeto frente al contenido de la matriz de DATOS.

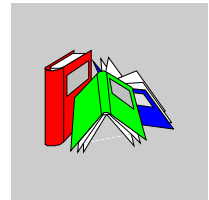
Tamaño del objeto Ejemplo	DATALENGTH	DATA(1)	DATA(2)	DATA(3)	DATA(4)
1 byte <b>01 hex</b>	1	<b>01 hex</b>	no significativo	no significativo	no significativo
2 bytes <b>01 23 hex</b>	2	LSB <b>23 hex</b>	MSB <b>01 hex</b>	no significativo	no significativo
3 bytes <b>01 23 45 hex</b>	3	LSB <b>45 hex</b>	<b>23 hex</b>	MSB <b>01 hex</b>	no significativo
4 bytes <b>01 23 45 67 hex</b>	4	LSB <b>67 hex</b>	<b>45 hex</b>	<b>23 hex</b>	MSB <b>01 hex</b>

LSB = byte poco significativo

MSB = byte más significativo

---

# Glosario



---

## C

### **CAN**

CAN (Red de área del controlador) es un sistema de bus serie para el control incorporado.

### **CANopen**

CANopen es el protocolo de capa superior basado en CAN estandarizado internacionalmente (EN 50325-4) para el sistema de control incorporado.

### **CiA**

CAN in Automation (CiA) es el grupo de usuarios y fabricantes internacionales que desarrolla y apoya CANopen y otros protocolos de capa superior basados en CAN.

### **CiA405**

Interfaz y perfil de dispositivo CANopen para controladores programables IEC 61131-3

### **COB**

Objetos de comunicación del protocolo CANopen

## E

### **E/S**

Entrada/Salida

**EDS**

Archivo de hoja de datos electrónica para la descripción del dispositivo de bus de campo.

**EMCY**

Emergencia

**H**

**hex**

Hexadecimal

**I**

**ID**

Identificador

**M**

**ms**

Milisegundo

**N**

**NMT**

Gestión de red CANopen

**O**

**OD**

Diccionario de objetos del protocolo CANopen

## **P**

### **PDO**

Objetos de datos del proceso de protocolo CANopen

### **Protocolo**

Conjunto de reglas utilizado por los dispositivos para comunicarse entre sí en una red

## **S**

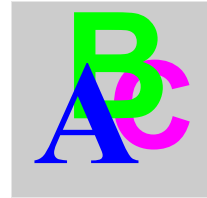
### **SDO**

Objetos de datos del servicio de protocolo CANopen



---

# Índice



## B

### Bloque de funciones

- GET\_CANOPEN\_KERNEL\_STATE, 42
- GET\_LOCAL\_NODE\_ID, 39
- GET\_STATE, 43
- NMT, 32
- RECV\_EMCY, 35
- RECV\_EMCY\_DEV, 37

### Bloques de funciones de acceso SDO

- CIA405.SDO\_READ: Leer objetos CANopen de cualquier tamaño, 46
- CIA405.SDO\_READ4: Leer objetos CANopen de hasta 4 bytes, 49
- CIA405.SDO\_WRITE: Escribir objetos CANopen de cualquier tamaño, 52
- CIA405.SDO\_WRITE: Escribir objetos CANopen de hasta 4 bytes, 57

### Bloques de funciones de estado de consulta

- CIA405.GET\_CANOPEN\_KERNEL\_STATE: Obtener el estado del kernel CANopen, 42
- CIA405.GET\_STATE: Obtener el estado del dispositivo CANopen, 43

### Bloques de funciones de gestión de red

- CIA405.NMT: Gestión del estado de NMT de dispositivos, 32
- CIA405.RECV\_EMCY\_DEV: Obtener mensaje EMCY de dispositivo, 37
- CIA405.RECV\_EMCY: Exploración de mensajes EMCY, 35

### Bloques de funciones de ID del nodo

- CIA405.GET\_LOCAL\_NODE\_ID: Obte-

ner ID de nodo CANopen del controlador, 39

## C

### CANOPEN\_KERNEL\_ERROR\_CODES

Lista de variables globales, 28

## D

### Descripciones de bloques de funciones, 23

Bloques de funciones de acceso SDO, 45

Bloques de funciones de estado de consulta, 41

Bloques de funciones de gestión de red, 31

Bloques de funciones de ID del nodo, 39

E/S y comportamiento comunes de los bloques de funciones, 25

### DEVICE\_STATE

Variables de salida, 44

## E

E/S y comportamiento comunes de los bloques de funciones

E/S y comportamiento comunes, 25

### EMCY\_ERROR

Variables de salida, 38

## **G**

GET\_CANOPEN\_KERNEL\_STATE  
    Bloque de funciones, 42  
GET\_LOCAL\_NODE\_ID  
    Bloque de funciones, 39  
GET\_STATE  
    Bloque de funciones, 43

Variables de salida  
    DEVICE\_STATE, 44  
    EMCY\_ERROR, 38

## **I**

introducción, 9  
Introducción  
    Introducción a CiA405, 21  
introducción  
    introducción al protocolo CANopen, 11

## **L**

Lista de variables globales  
    CANOPEN\_KERNEL\_ERROR\_CODES  
    , 28

## **N**

NMT  
    Bloque de funciones, 32

## **R**

RECV\_EMCY  
    Bloque de funciones, 35  
RECV\_EMCY\_DEV  
    Bloque de funciones, 37

## **T**

TRANSITION\_STATE  
    Variables de entrada, 33

## **V**

Variables de entrada  
    TRANSITION\_STATE, 33