

SoMachine

CANopen 管理功能

CAA CiA 405 库指南

05/2010

本文档中提供的信息包含有关此处所涉及产品的性能的一般说明和 / 或技术特性。本文档并非用于（也不代替）确定这些产品针对特定用户应用的适用性或可靠性。任何此类用户或集成者都有责任就相关特定应用或使用方面对产品执行适当且完整的风险分析、评估和测试。Schneider Electric 或是其任何附属机构或子公司对于误用此处包含的信息而产生的后果概不负责。如果您有关于改进或更正此出版物的任何建议，或者从中发现错误，请通知我们。

未经 Schneider Electric 明确书面许可，不得以任何形式、通过任何电子或机械手段（包括影印）复制本文档的任何部分。

在安装和使用本产品时，必须遵守国家、地区和当地的所有相关的安全法规。出于安全方面的考虑和为了帮助确保符合归档的系统数据，只允许制造商对各个组件进行维修。

当设备用于具有技术安全要求的应用时，必须遵守有关的使用说明。

如果在我们的硬件产品上不正确地使用 Schneider Electric 软件或认可的软件，则可能导致人身伤害、损害或不正确的操作结果。

不遵守此信息可能导致人身伤害或设备损坏。

© 2010 Schneider Electric。保留所有权利。

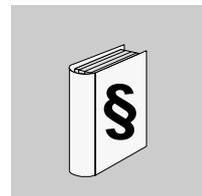
目录



安全信息	5
关于本书	7
部分 I 简介	9
章 1 CANopen 协议简介	11
简介	12
NMT 协议	13
PDO 协议	14
SDO 协议	15
EMCY 协议	16
SYNC 协议	17
错误控制协议	18
章 2 CiA405 简介	21
简介	21
部分 II 功能块描述	23
章 3 功能块通用 I/O 和行为	25
3.1 通用 I/O 和行为	25
通用 I/O 描述	26
CANopen 内核检测到的错误代码	28
功能块执行图	29
章 4 网络管理功能块	31
4.1 CIA405 .NMT : 设备 NMT 状态管理	32
设备 NMT 状态管理	32
4.2 CIA405 .RECV_EM CY : EMCY 消息扫描	35
EMCY 消息扫描	35
4.3 CIA405 .RECV_EM CY_DEV : 获取设备 EMCY 消息	36
获取设备 EMCY 消息	36
章 5 自有节点 ID 功能块	39
5.1 CIA405 .GET_LOCAL_NODE_ID : 获取控制器 CANopen 节点 ID	39
获取控制器 CANopen 节点 ID	39

章 6 查询状态功能块	41
6.1 CIA405.GET_CANOPEN_KERNEL_STATE : 获取 CANopen 内核状态 获取 CANopen 内核状态	42 42
6.2 CIA405.GET_STATE : 获取 CANopen 设备状态 获取 CANopen 设备状态	43 43
章 7 SDO 访问功能块	45
7.1 CIA405.SDO_READ : 读取任意大小的 CANopen 对象 读取任意大小的 CANopen 对象	46 46
7.2 CIA405.SDO_READ4 : 读取最多 4 个字节的 CANopen 对象 读取最多 4 个字节的 CANopen 对象	49 49
7.3 CIA405.SDO_WRITE : 写入任意大小的 CANopen 对象 写入任意大小的 CANopen 对象	52 52
7.4 CIA405.SDO_WRITE4 : 写入最多 4 个字节的 CANopen 对象 写入最多 4 个字节的 CANopen 对象	57 57
术语表	59
索引	63

安全信息



重要信息

声明

在尝试安装、操作或维护设备之前，请仔细阅读下述说明并通过查看来熟悉设备。下述特别信息可能会在本文其他地方或设备上出现，提示用户潜在的危險，或者提醒注意有关阐明或简化某一过程的信息。



在“危險”或“警告”安全标签上添加此符号表示存在触电危險，如果不遵守使用说明，将导致人身伤害。



这是提醒注意安全的符号。提醒用户可能存在人身伤害的危險。请遵守所有带此符号的安全注意事项，以避免可能的人身伤害甚至死亡。

危險

“危險”表示极可能存在危險，如果不遵守说明，可导致严重的人身伤害甚至死亡。

警告

“警告”表示可能存在危險，如果不遵守说明，可导致严重的人身伤害甚至死亡，或设备损坏。

▲ 注意

“注意”表示可能存在危险，如果不遵守说明，可导致严重的人身伤害或设备损坏。

注意

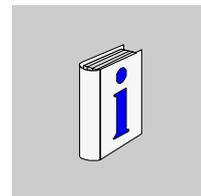
注意(无安全警告符号)，表示存在潜在的危险，如果忽视，可能导致设备损坏。

请注意

电气设备的安装、操作、维修和维护工作仅限于合格人员执行。对于使用本资料所引发的任何后果，Schneider Electric 概不负责。

专业人员是指掌握与电气设备的制造和操作相关的技能和知识的人员，他们经过安全培训能够发现和避免相关的危险。

关于本书



概览

文档范围

本文档介绍用于从控制器应用程序管理和监控 CANopen 网络和设备的 SoMachine CAA CiA 405 库功能块。

有效性说明

本文档已随 SoMachine V2.0 的发布进行了更新。

关于产品的资讯

警告

意外的设备操作

- 仅使用 Schneider Electric 认可的可与本设备配合使用的软件。
- 每次更改物理硬件配置后，请更新应用程序。

如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

警告

失去控制

- 任何控制方案的设计者都必须考虑到控制路径可能出现故障的情况，并为某些关键控制功能提供一种方法，使其在出现路径故障时，以及出现路径故障后恢复至安全状态。紧急停止和越程停止、断电和重启都属于关键控制功能。
- 对于关键控制功能，必须提供单独或冗余的控制路径。
- 系统控制路径可包括通讯链路。必须对暗含的无法预料的传输延迟或链接失效问题加以考虑。
- 遵守所有事故预防规定和当地的安全指南。¹
- 为了保证正确运行，在投入使用前，必须对设备的每次执行情况分别进行全面测试。

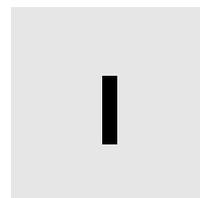
如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

¹ 有关详细信息，请参阅 NEMA ICS 1.1（最新版）中的“安全指导原则 - 固态控制器的应用、安装和维护”以及 NEMA ICS 7.1（最新版）中的“结构安全标准及可调速驱动系统的选择、安装与操作指南”或您特定地区的类似规定。

用户意见

欢迎对本书提出意见。您可以给我们发邮件，我们的邮件地址是 techcomm@schneider-electric.com。

简介



概述

本部分概述 CANopen 中使用的各种协议，并介绍 IEC 61131-3 可编程控制器 (CiA405) 的 CANopen 接口和设备配置文件。

注意：本简介的部分内容摘录并改编自 CAN in Automation (CiA) 网站 (www.can-cia.org)。CiA? 和 CANopen® 是 CAN in Automation e.V 的欧盟注册商标。

关于 CiA：CAN in Automation (CiA) 是控制器区域网络 (CAN) 用户和制造商联合组成的国际性组织。该组织是一家非盈利性协会，提供 CAN（国际标准为 ISO 11898 系列）相关的技术、产品和市场信息。其目标是提升 CAN 的形象，并为 CAN 技术的未来发展指明方向。该组织的一项重要工作是开发和维护 CANopen 规范，以及为所有其他基于 CAN 的国际标准化高层次协议提供支持。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
1	CANopen 协议简介	11
2	CiA405 简介	21

CANopen 协议简介



1

本章包含了哪些内容？

本章包含了以下主题：

主题	页
简介	12
NMT 协议	13
PDO 协议	14
SDO 协议	15
EMCY 协议	16
SYNC 协议	17
错误控制协议	18

简介

CANopen 协议

CANopen 协议为实时数据（过程数据对象 (PDO)）、配置数据（服务数据对象 (SDO)）和网络管理数据（启动消息、NMT 消息和错误控制消息）以及其他功能（时间标记、SYNC 消息、紧急消息）提供标准化的通讯对象 (COB)。所有通讯对象都可通过设备对象字典中的 CAN 网络进行访问。这些对象可通过 16 位索引进行寻址。对于数组和记录对象，还存在另一个 8 位子索引。通讯对象在 CANopen 网络上由专用且唯一的标识符 (COB ID) 来标识。

设备对象字典

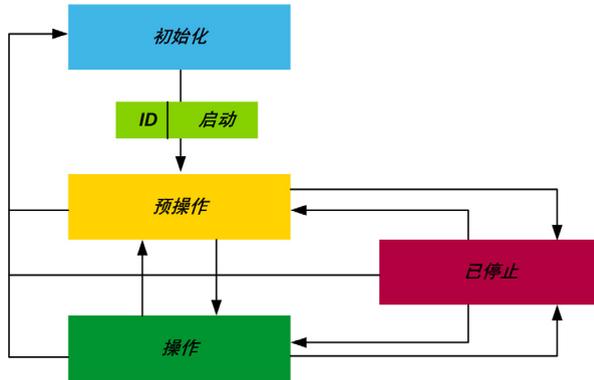
CANopen 设备的对象字典条目在电子数据表 (EDS) 文件中介绍。CANopen 主站和从站参数可以在 SoMachine CANopen 配置程序中进行调整。有关 CANopen 配置的详细信息，请参阅在线帮助 CoDeSys 部分的“编辑器 / 设备编辑器 / CANbus 配置编辑器”一章。

NMT 协议

网络管理 (NMT) 协议

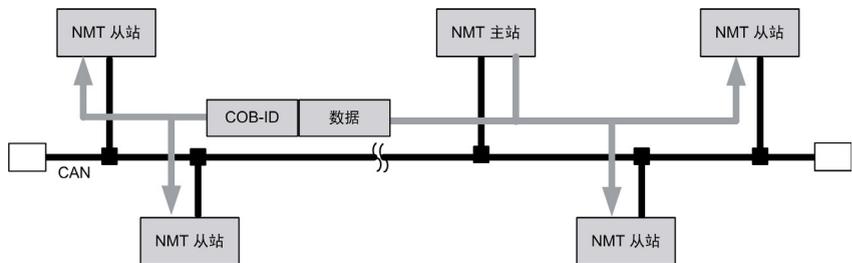
网络管理 (NMT) 协议用于控制 CANopen NMT 从站设备的网络行为。单个专用网络组成部分或所有网络组成部分均可以通过其 NMT 从站状态机中的 NMT 协议进行交换。

下图显示的是 CANopen NMT 从站状态机。



基于主站 - 从站关系的 NMT 服务

所有 CANopen 设备都会计算传入的 NMT 命令（NMT 消息中的命令说明符 (CS) 代码）。只有具有 NMT 主站功能的 CANopen 设备才能传输 NMT 消息。在 CANopen 网络中，只有一个 NMT 主站处于活动状态。



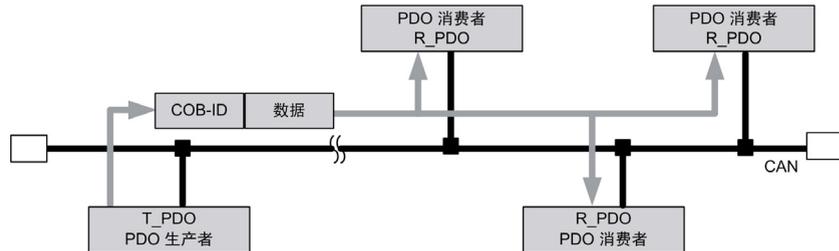
PDO 协议

过程数据对象 (PDO) 协议

过程数据对象 (PDO) 是通过广播传输的高优先级短（最多 8 个字节）CAN 消息。因此，PDO 适用于实时数据（如驱动器或 I/O 模块的控制和状态信息、传感器提供的测量值等）的传输。

PDO 以未确认方式进行传输。这表示即使有特定的网络组成部分接收到信息，也不会对接收进行确认。

下图显示了从生产者（传输 PDO）到消费者（接收 PDO）的 PDO 传输。



PDO 传输的触发事件

下面定义了 PDO 传输的触发事件。

触发事件	定义
事件或定时器驱动	设备内部事件可触发 PDO 传输（例如，值更改、温度值超出特定限制、事件定时器已过时等）
远程请求	收到远程传输请求 (RTR)，请求进行 PDO 传输。
循环同步	PDO 的传输与 SYNC 消息的接收紧密相关。
非循环同步	定义的设备特定事件可触发 PDO，而 PDO 在接收到下一个 SYNC 消息时传输。

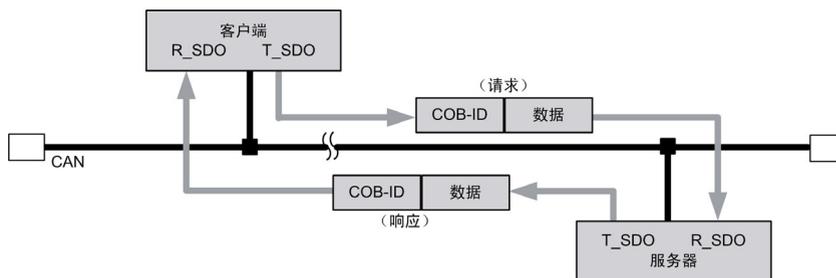
SDO 协议

服务数据对象 (SDO) 协议

通过服务数据对象 (SDO) 可以访问 CANopen 对象字典 (OD) 的任何条目。SDO 将在两个设备之间建立 P2P 通讯通道。此外，通过 SDO 协议还可采用分段方式传输任何数据量。因此，SDO 协议主要用于与配置数据通讯。

通过分别使用客户端通道配置相关 SDO 服务器，可建立两个设备之间的 SDO 连接。SDO 传输是一项确认服务。

下图显示的带有请求和响应的 SDO 交换。

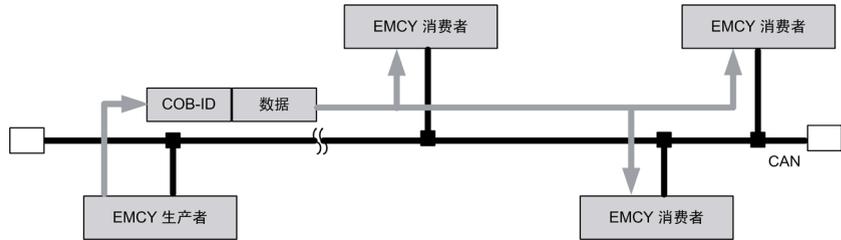


EMCY 协议

紧急 (EMCY) 协议

紧急 (EMCY) 对象可以让设备指示检测到设备内部错误。接收到此信号后，其他网络组成部分可以评估接收到的信息，并启动相应的特定于制造商的对策。

下图显示的是通过 EMCY 对象检测到的错误消息。



EMCY 消息

从其他 CANopen 设备接收到的 EMCY 消息存储在 EMCY 存储表中。对于每个 EMCY 生产者，如果没有收到 EMCY 消息，或者只要最新的 EMCY 消息是*无错误*消息，则 EMCY 消费者会认为 EMCY 生产者未检测到内部错误。

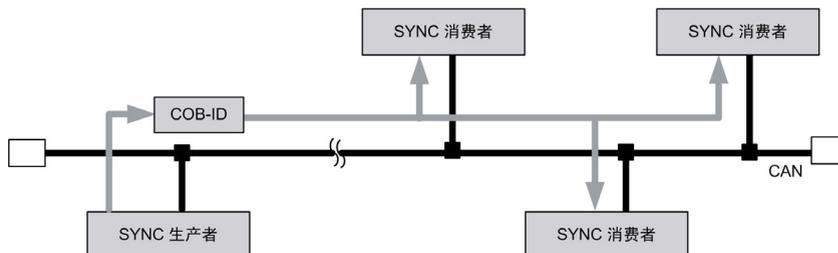
SYNC 协议

同步 (SYNC) 协议

使用 SYNC 协议可以实现同步网络行为。循环传输的 SYNC 消息表示消费者启动了其特定于应用程序的行为，该行为与 SYNC 消息的接收紧密相关。

一旦收到数据，同步 PDO 会在收到 SYNC 消息时认为该数据有效。

下图显示的是 SYNC 协议的 SYNC 消息传输。

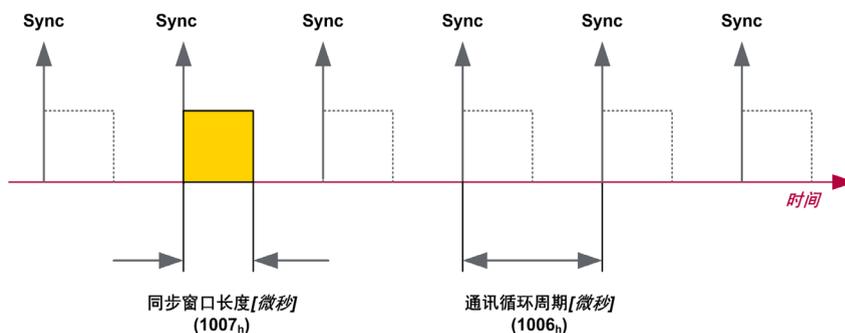


SYNC 行为的控制变量

两个连续 SYNC 消息之间的时间段称为 *通讯循环周期*，可以在 SYNC 生产者的对象字典（对象索引为十六进制的 1006）中进行调整。

同步传输 PDO 在接收到 SYNC 消息后的指定时间范围内开始传输。此时间范围称为 *同步窗口长度*，可在所有必须传输同步 PDO 的设备的对象字典（对象索引为十六进制的 1007）中进行配置。

下图显示的是同步窗口长度和通讯循环周期。



错误控制协议

错误控制协议

心跳或保护服务是用于检查网络组成部分是否存在并获取其状态的网络管理 (NMT) 服务。

CANopen 中存在两种类型的错误控制协议：

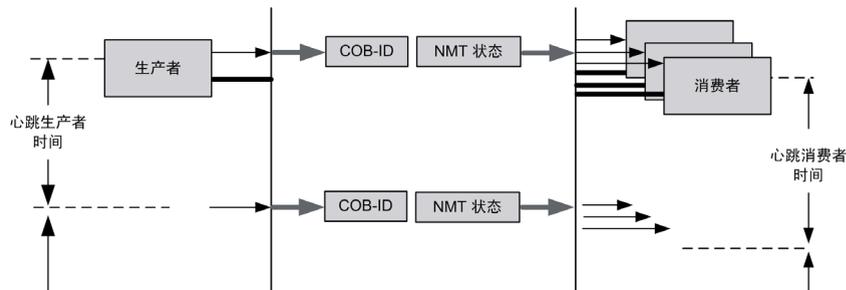
- 心跳
- 节点保护 / 寿命保护

心跳协议是较灵活的协议，在没有远程传输请求 (RTR) 的情况下也可以工作，因此，只要 CANopen 设备支持此协议，就应使用该协议。

心跳

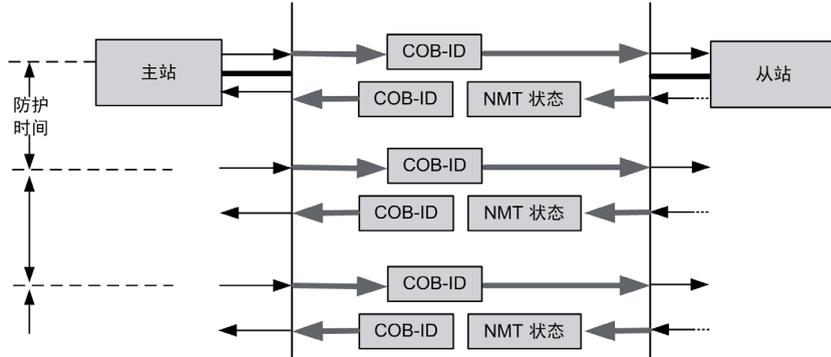
CANopen 设备将循环传输其心跳。此循环时间称为 *心跳生产者时间*。接收到心跳消息后，心跳消费者（例如 CANopen 管理器）将评估相关设备是否仍在正常运行以及是否处于正确的 NMT 状态。在特定时间段内，消费者必须从相关的心跳生产者至少接收一条心跳消息，该时间段（心跳消费者时间）在消费者中进行配置。

下图显示的是错误控制协议的心跳与时间间隔。



节点保护 / 寿命保护

CANopen 管理器通过远程传输请求 (RTR) 来请求错误控制消息。受保护的设备会回复一个 CAN 数据帧，以指示当前的 NMT 状态。



主站： 主站监控从站的 NMT 状态（节点保护）

从站： 从站监控主站的寿命信号（寿命保护）

CiA405 简介

2

简介

访问 CANopen 网络的方法

IEC 61131-3 可编程控制器 (CiA405) 的 CANopen 接口和设备配置文件介绍了两种从控制器访问 CANopen 网络的方法：

- 网络变量
- 功能块

网络变量

网络变量通常会映射到要接收或要传输的 PDO 中。在对象字典中，可在定义的索引范围内访问 IEC 61131-3 变量。

功能块

配置文件还定义了一些特定于 CANopen 的功能块，例如 SDO、NMT 和紧急通讯服务。

CAA CiA 405 库

CAA CiA 405 库提供了一组满足 CiA405 要求的功能块，用于从控制器（CANopen 主站）的应用程序（IEC61131-3 程序）访问 CANopen 网络。将 CANopen 管理器添加到控制器 CAN 总线接口后，控制器库管理器会自动声明该库。

在该库中，按如下方式组织功能块：

- 网络管理 功能块：
 - CIA405 .NMT : 控制 CANopen 设备 NMT 状态
 - CIA405 .RECV_EM CY : 扫描所有设备的 EMCY 存储
 - CIA405 .RECV_EM CY_DEV : 获取指定设备的最后一条存储的 EMCY 消息
- 自有节点 id 功能块：
 - CIA405 .GET_LOCAL_NODE_ID : 获取控制器 CANopen 管理器节点 ID
- 查询状态 功能块：

- CIA405 .GET_CANOPEN_KERNEL_STATE : 获取 CANopen 内核当前状态
- CIA405 .GET_STATE : 获取指定设备的当前状态
- SDO 访问 功能块:
 - CIA405 .SDO_READ : 读取指定设备的任意大小的对象
 - CIA405 .SDO_READ4 : 读取指定设备的最多 4 个字节的对象
 - CIA405 .SDO_WRITE : 写入指定设备的任意大小的对象
 - CIA405 .SDO_WRITE4 : 写入指定设备的最多 4 个字节的对象

注意: 要在 CANopen 上控制 Altivar 驱动器和 Lexium 运动驱动器, 最好使用专用 PLCopen 功能块 (请参阅 *Altivar*、*集成 Lexium* 和 *Lexium 库指南*)。

命名空间

CAA CiA 405 库命名空间为 CIA405 。库功能块、变量和常量必须与库命名空间一起使用。

功能块实例和变量声明示例:

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

功能块描述



概述

本部分概述用于从控制器应用程序管理和监控 CANopen 网络和设备的 CAA CiA 405 库的各个功能块。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
3	功能块通用 I/O 和行为	25
4	网络管理功能块	31
5	自有节点 ID 功能块	39
6	查询状态功能块	41
7	SDO 访问功能块	45

功能块通用 I/O 和行为

3

3.1 通用 I/O 和行为

本节包含了哪些内容？

本节包含了以下主题：

主题	页
通用 I/O 描述	26
CANopen 内核检测到的错误代码	28
功能块执行图	29

通用 I/O 描述

简介

本主题以 CIA405 .RECV_EMCY 功能块为示例，介绍 CAA CiA 405 库功能块的常规管理和执行。下文介绍了通用于所有功能块的 I/O。这些功能块继承自 CiA405Base 内部隐藏功能块。

图形表示形式

下图中突出显示了 CAA CiA 405 库中所有功能块共有的参数：



通用输入变量描述

下表介绍了 CAA CiA 405 库的所有功能块共有的输入参数。

输入	数据类型	说明
网络	USINT	必须在其上执行所请求服务的 CAN 通道。 1（缺省值）= 第一个 CAN 总线接口 2 = 第二个 CAN 总线接口（如果存在）
ENABLE	BOOL	启用功能块的执行。 在上升沿：执行开始。 在下降沿：如果执行未完成，则取消执行。否则，输出数据会复位为 0。
TIMEOUT	UDINT	最大执行时间（以毫秒为单位）。如果在收到响应之前达到超时，则执行会因超时错误而中止。 0（出厂设置）= 禁用超时。 1...65535 = 超时值（以毫秒为单位）。

通用输出变量描述

下表介绍了 CAA CiA 405 库的所有功能块共有的输出变量。

输出	数据类型	说明
CONFIRM	BOOL	成功完成执行时为 TRUE。
错误	CANOPEN_KERNEL_ERROR (USINT)	包含 CANopen 内核返回的检测到错误代码的执行。 00 (十六进制) = 未检测到任何执行错误。 01 (十六进制) = 检测到错误, 但是代码在另一个 功能块输出上提供 (而不是来自 CANopen 内核)。 02..FF (十六进制) = CANopen 内核检测到的错 误代码。

CANopen 内核检测到的错误代码

说明

库提供一个包含预定义错误代码变量的 CIA405
`.CANOPEN_KERNEL_ERROR_CODES` 全局变量列表。

检测到的错误代码

下面提供了错误代码及关联全局变量和说明。

检测到的错误代码	说明
CANOPEN_KERNEL_NO_ERROR = 00 (十六进制)	CANopen 内核未检测到错误。
CANOPEN_KERNEL_OTHER_ERROR = 01 (十六进制)	<p>如果 ERROR 输出 = 01 (十六进制)，则已检测到错误；如果在功能块上存在其他错误代码输出，则该输出包含更加具体的信息。</p> <p>示例：</p> <ul style="list-style-type: none"> ● CIA405 .SDO_READ 和 CIA405 .SDO_WRITE 功能块：ERRORINFO 输出包含 SDO 中止消息的内容。 ● CIA405 .RECV_EM CY 和 CIA405 .RECV_EM CY_DEV 功能块：ERRORINFO 输出包含接收到的 EM CY 消息的内容。如果 ERRORINFO 输出中提供了紧急消息，则 ERROR 输出 = 1 且 CONFIRM 输出 = 0。 ● CIA405 .GET_CANOPEN_KERNEL_STATE 功能块：绝不会提供十六进制 01 值，因为不存在任何其他错误代码输出。
CANOPEN_KERNEL_DATA_OVERFLOW = 02 (十六进制)	CANopen 对象的发送缓冲区或接收缓冲区溢出。
CANOPEN_KERNEL_TIMEOUT = 03 (十六进制)	发生功能块执行超时。
CANOPEN_KERNEL_CANBUS_OFF = 10 (十六进制)	CANopen 节点从 CAN 总线断开连接。
CANOPEN_KERNEL_CAN_ERROR_PASSIVE = 11 (十六进制)	CANopen 节点处于错误被动状态：节点能够通讯，但是不允许在检测到错误时发送活动错误标志。
CANOPEN_INTERNAL_FB_ERROR = 21 (十六进制)	<p>特定制造商错误代码。</p> <p>注意： 21 (十六进制) 到 FF (十六进制) 为设备制造商专用值。</p>

功能块执行图

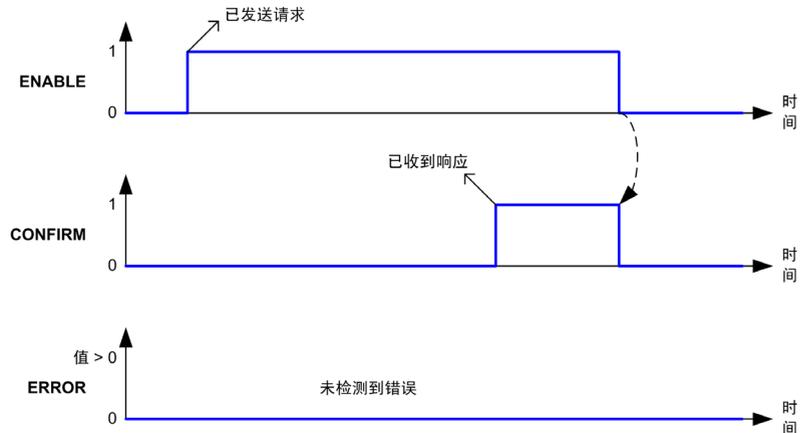
控制信号的行为

下面介绍了控制信号 `ENABLE`、`CONFIRM` 和 `ERROR` 的三个典型行为：

- 执行在没有检测到错误的情况下结束
- 执行被应用程序取消
- 执行在检测到错误时中止或结束

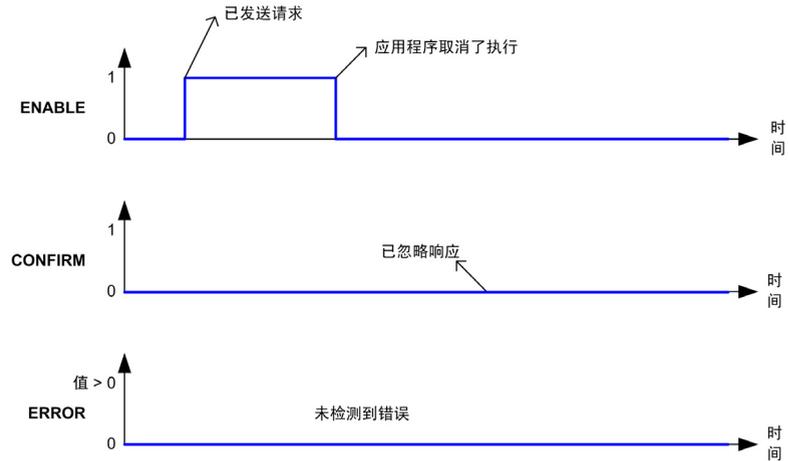
执行在没有检测到错误的情况下结束

一旦对当前请求做出响应，并且没有检测到任何错误，输出 `CONFIRM` 便会立即设置为 `TRUE`，并保留 `TRUE` 值（前提是在输入 `ENABLE` 为 `TRUE` 的情况下调用了该功能块）。如果在 `ENABLE` 被复位为 `FALSE` 的情况下调用了该功能块，则 `CONFIRM` 会复位为 `FALSE`，并且功能块可以开始新执行。



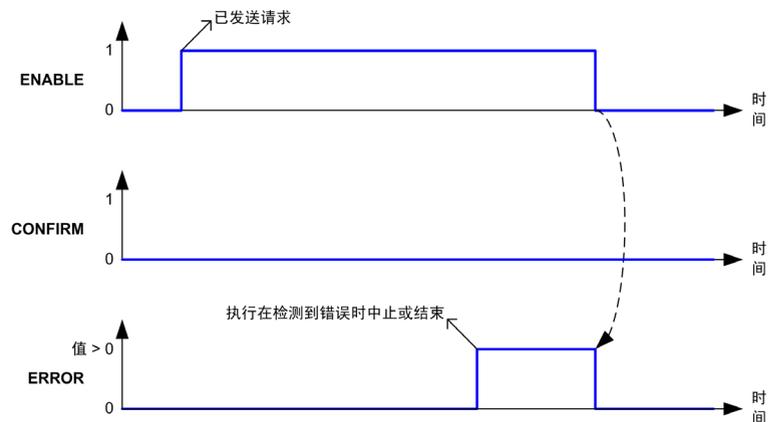
执行被应用程序取消

如果在当前执行结束之前，在 `ENABLE` 复位为 `FALSE` 的情况下调用该功能块，则会取消功能块执行。可能目前已对取消的请求做出回应，或该响应稍后将到达，但这些回应此时会被忽略。



执行在检测到错误时中止或结束

一旦当前执行由于收到 SDO 中止消息而中止，或由于检测到错误而结束，输出 `ERROR` 就会被设置为除 0 以外的值（有关检测到的错误代码的更多信息，请参阅 `CANopen` 内核检测到的错误代码（参见第 28 页））。如果在 `ENABLE` 复位为 `FALSE` 的情况下调用该功能块，则输出 `ERROR` 会复位为 0，并且功能块可以开始新执行。



网络管理功能块

4

本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
4.1	CIA405 .NMT : 设备 NMT 状态管理	32
4.2	CIA405 .RECV_EM CY : EMCY 消息扫描	35
4.3	CIA405 .RECV_EM CY_DEV : 获取设备 EMCY 消息	36

4.1 CIA405 .NMT : 设备 NMT 状态管理

设备 NMT 状态管理

功能块描述

使用 CIA405 .NMT 功能块可以从控制器应用程序控制 CANopen 设备的 NMT 状态。该功能块通过执行对 CANopen 目标设备的 NMT 服务请求，来执行请求的 NMT 状态转换。

图形表示形式



特定输入变量描述

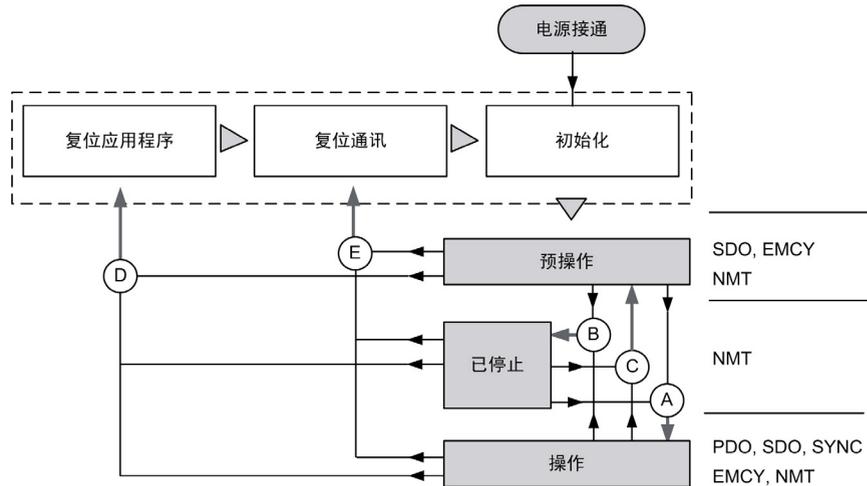
输入	数据类型	说明
DEVICE	DEVICE (USINT)	CANopen 目标设备节点 ID。 0 (缺省值) = 所有 NMT 从站设备 1...127 = 目标设备节点 ID
STATE	TRANSITION_STATE 。有关详细信息， 请参阅下文	请求的 NMT 状态转换。

注意：有关功能块通用 I/O 描述和执行行为的信息，请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

CIA405 .TRANSITION_STATE ENUM

NMT 状态机描述主要操作中的 NMT 从站的初始化和状态。

下图显示的是 NMT 状态、关联的可用通讯对象（PDO、SDO、SYNC、EMCY 和 NMT）和 5 种状态转换（A 到 E）。

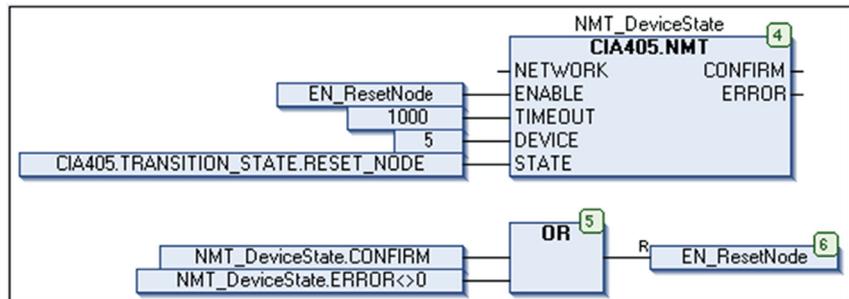


CIA405 .TRANSITION_STATE 枚举类型包含下表中介紹的 NMT 状态转换命令。

枚举器	值（十六进制）	说明
STOP_REMOTE_NODE	0004	切换到 <i>停止</i> 状态。 (转换 B)
START_REMOTE_NODE	0005	切换到 <i>正常操作</i> 状态。 (转换 A)
RESET_NODE	0006	切换到 <i>复位应用程序</i> 状态。加载设备配置文件的已保存数据，并自动从 <i>复位通讯</i> 切换到 <i>预操作</i> 状态。 (转换 D)
RESET_COMMUNICATION	0007	切换到 <i>复位通讯</i> 状态。加载通讯配置文件的已存储数据，并自动切换到 <i>预操作</i> 状态。 (转换 E)
ENTER_PRE_OPERATIONAL	007F	切换到 <i>预操作</i> 状态。 (转换 C)
ALL_EXCEPT_NMT_AND_SENDER	0800	未实现（无效参数）

示例

以下示例显示了如何在超时 1 秒（1000 毫秒）的情况下，将“复位节点”命令发送到连接到第一个 CAN 总线接口的 CANopen 节点 5。该命令在布尔变量 EN_ResetNode 设置为 TRUE（由用户在线设置或应用程序设置）时发送。当执行成功结束（输出 CONFIRM = TRUE）或检测到错误（输出 ERROR <> 0）时，EN_ResetNode 命令会复位为 FALSE。



4.2 CIA405 .RECV_EMCY : EMCY 消息扫描

EMCY 消息扫描

功能块描述

CIA405.RECV_EMCY 功能块会在一个循环中对所有现有 CANopen 设备的紧急 (EMCY) 消息存储进行扫描，并返回找到的 EMCY 消息。

启用该功能块后，EMCY 存储扫描将从上一个扫描停止点开始启动：

- 如果找到一条无错误之外的 EMCY 消息，则扫描会在此点结束，功能块返回该 EMCY 消息及关联设备节点 ID。
- 如果找到一条无错误 EMCY 消息，但该消息以前不是无错误消息，则扫描会在此点结束，功能块返回值为 0 的 EMCY 消息及关联设备节点 ID。
- 如果没有无错误之外的 EMCY 消息，且在一个完整扫描过程中未找到新的无错误 EMCY 消息，则功能块会返回值为 0 的 EMCY 消息和值为 0 的设备节点 ID。

图形表示形式



特定输出变量描述

输出	数据类型	说明
DEVICE	DEVICE (USINT)	与返回的 EMCY 消息关联的 CANopen 设备的节点 ID。 0 = 未找到任何 EMCY 消息 1...127 = 设备节点 ID
ERRORINFO	CS.EMCY_ERROR。 有关详细信息，请参阅 CS.EMCY_ERROR 结构 (参见第 37 页)	从节点 ID 为 DEVICE 的 CANopen 设备接收的最后一个 EMCY 消息。

注意：有关功能块通用 I/O 描述和执行行为的信息，请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

4.3 CIA405 .RECV_EMCY_DEV : 获取设备 EMCY 消息

获取设备 EMCY 消息

功能块描述

CIA405 .RECV_EMCY_DEV 功能块返回从指定 CANopen 设备接收的最后一存储的紧急 (EMCY) 消息。

图形表示形式



特定输入变量描述

输入	类型	说明
DEVICE	DEVICE (USINT)	要检查的 CANopen 设备的节点 ID。 0 (缺省值) = 本地设备 (控制器) 1..127 = 设备节点 ID

特定输出变量描述

输出	类型	说明
ERRORINFO	CS.EMCY_ERROR。(有关详细信息, 请参阅下文)	从节点 ID 为 <i>DEVICE</i> 的 CANopen 设备接收的最后一个 EMCY 消息。

注意: 有关功能块通用 I/O 描述和执行行为的信息, 请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

CS.EMCY_ERROR 结构

CS.EMCY_ERROR 是与 EMCY 消息内容关联的结构（有关具体 EMCY 消息内容的更多信息，请参阅您的设备 CANopen 指南）。CS.EMCY_ERROR 包含以下元素。

元素	类型	说明
EMCY_ERROR_CODE	字	EMCY 消息的错误代码
ERROR_REGISTER	字节	EMCY 消息的错误寄存器（位字段）
ERROR_FIELD	ARRAY[1...5] OF BYTE	EMCY 消息的特定设备制造商错误字段

注意：此结构在 CAA CANopen 栈库（命名空间 = CS）中声明。这是为什么要使用结构全名（<命名空间>.<数据类型>）的原因。CAA CiA 405 库命名空间为 CIA405。

5.1 CIA405 .GET_LOCAL_NODE_ID : 获取控制器 CANopen 节点 ID

获取控制器 CANopen 节点 ID

功能块描述

CIA405 .GET_LOCAL_NODE_ID 功能块返回指定 CAN 总线接口上的控制器 CANopen 节点 ID。

图形表示形式



特定输出变量描述

输出	类型	说明
DEVICE	DEVICE (USINT)	控制器的 CANopen 节点 ID。 0 = 无效 1...127 = 控制器节点 ID

注意：有关功能块通用 I/O 描述和执行行为的信息，请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

查询状态功能块

6

本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
6.1	CIA405.GET_CANOPEN_KERNEL_STATE : 获取 CANopen 内核状态	42
6.2	CIA405.GET_STATE : 获取 CANopen 设备状态	43

6.1 CIA405.GET_CANOPEN_KERNEL_STATE : 获取 CANopen 内核状态

获取 CANopen 内核状态

功能块描述

CIA405.GET_CANOPEN_KERNEL_STATE 功能块返回控制器 CANopen 内核的当前状态。

图形表示形式



特定输出变量描述

输出	类型	说明
STATE	CANOPEN_KERNEL_ERROR (有关详细信息, 请参阅 CANopen 内核检测到的错误代码 (参见第 28 页))	控制器 CANopen 内核的当前状态
DEVICE	DEVICE (USINT)	控制器的 CANopen 节点 ID。 0 = 无效 1...127 = 控制器节点 ID

注意: 有关功能块通用 I/O 描述和执行行为的信息, 请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

6.2 CIA405.GET_STATE : 获取 CANopen 设备状态

获取 CANopen 设备状态

功能块描述

CIA405.GET_STATE 功能块在心跳或节点保护处于活动状态时，返回指定 CANopen 设备的当前 NMT 状态。

图形表示形式



特定输入变量描述

输入	类型	说明
DEVICE	DEVICE (USINT)	要检查的 CANopen 设备的节点 ID。 0 (缺省值) = 本地设备 (控制器) 1...127 = 设备节点 ID

特定输出变量描述

输出	类型	说明
STATE	DEVICE_STATE (有关详细信息, 请参阅下文)	CANopen 设备的 NMT 状态。

注意: 有关功能块通用 I/O 描述和执行行为的信息, 请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

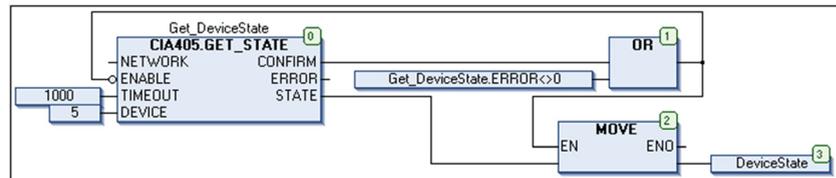
CIA405.DEVICE_STATE ENUM

CIA405 .DEVICE_STATE 枚举类型包含 CANopen 设备 NMT 状态的列表。

枚举器	值	说明
INIT	0	初始化状态
RESET_COMM	1	复位通讯状态
RESET_APP	2	复位应用程序状态
PRE_OPERATIONAL	3	预操作状态
STOPPED	4	停止状态
OPERATIONAL	5	正常操作状态
UNKNOWN	6	未知 NMT 状态。 所选设备的节点保护或心跳不处于活动状态，或者控制器不是心跳消费者。
NOT_AVAIL	7	NMT 状态不可用。 所选设备的节点保护或心跳处于活动状态，但是该设备在超时之前未正确报告其 NMT 状态。

示例

以下示例显示了如何在超时 1 秒（1000 毫秒）的情况下，获取连接到第一个 CAN 总线接口的 CANopen 节点 5 的状态。CIA405.GET_STATE 功能会自动执行，以进行连续的状态读取。设备 NMT 状态会复制到 CIA405.DEVICE_STATE 类型的 DeviceState 变量中。



SDO 访问功能块



本章包含了哪些内容?

本章包含了以下部分:

节	主题	页
7.1	CIA405.SDO_READ : 读取任意大小的 CANopen 对象	46
7.2	CIA405.SDO_READ4 : 读取最多 4 个字节的 CANopen 对象	49
7.3	CIA405.SDO_WRITE : 写入任意大小的 CANopen 对象	52
7.4	CIA405.SDO_WRITE4 : 写入最多 4 个字节的 CANopen 对象	57

7.1 CIA405.SDO_READ : 读取任意大小的 CANopen 对象

读取任意大小的 CANopen 对象

功能块描述

CIA405 .SDO_READ 功能块用于通过 SDO 消息读取指定设备的任意大小的 CANopen 对象。

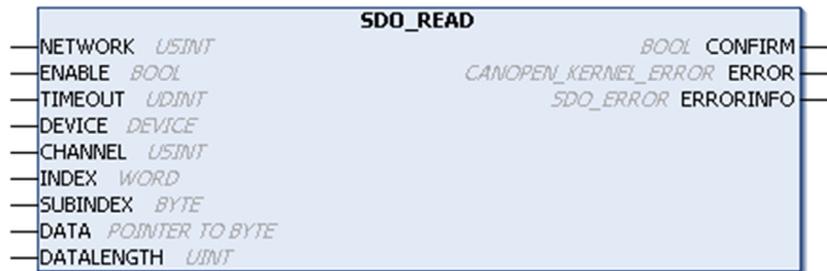
必须向该功能块传递以下这些特定参数：

- 设备节点 ID
- SDO 客户端 / 服务器通道（缺省情况下，仅定义一个通道）
- CANopen 对象索引 / 子索引
- 用来存储对象值的数据缓冲区的指针
- 数据缓冲区大小

如果读取成功结束，则该功能块会返回读取的对象大小。数据在数据缓冲区中提供。

注意：如果要读取的对象的大小小于或等于 4 个字节，最好使用 CIA405 .SDO_READ4 功能块。

图形表示形式



特定输入变量描述

输入	类型	说明
DEVICE	DEVICE (USINT)	CANopen 设备的节点 ID。 0 (缺省值) = 本地设备 (控制器) 1...127 = 设备节点 ID
CHANNEL	USINT	SDO 通道编号。 缺省值 = 1
INDEX	WORD	对象索引。 范围: 0000 (十六进制) ...FFFF (十六进制)
SUBINDEX	BYTE	对象子索引。 范围: 00 (十六进制) ..FF (十六进制)
DATA	POINTER TO BYTE	接收从设备对象读取的数据的数据缓冲区地址。 必须使用 ADR 标准功能定义关联指针。

特定输出变量描述

输出	类型	说明
ERRORINFO	SDO_ERROR (UDINT)	当 ERROR 输出 = 1 时, 返回 SDO 中止消息内容 (大小为 4 个字节)。

特定输入 / 输出变量描述

输入 / 输出	类型	说明
DATALENGTH	UINT	作为输入: 数据缓冲区的大小 (以字节为单位)。 注意: 要确保 DATALENGTH 输入正确初始化 (当功能块执行在 ENABLE 输入上升沿上开始时) 为数据缓冲区大小, 请使用 SIZEOF 标准功能。 作为输出: 读取的对象的大小 (以字节为单位)。

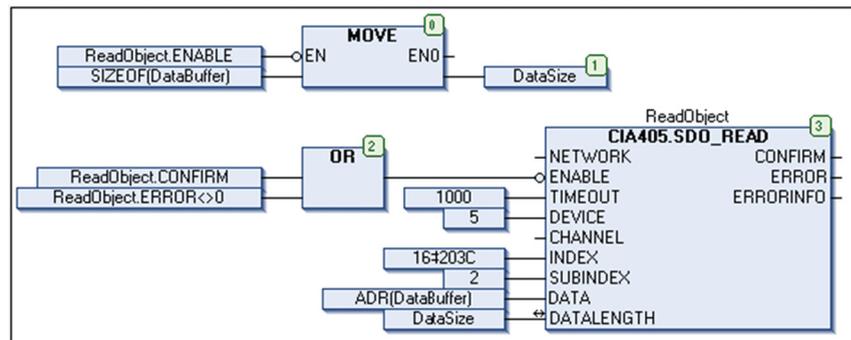
注意: 有关功能块通用 I/O 描述和执行行为的信息, 请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

示例

以下示例显示了如何在超时 1 秒（1000 毫秒）的情况下，读取连接到第一个 CAN 总线接口的 CANopen 节点 5 的对象索引 203C（十六进制）/子索引 02（十六进制）。CIA405.SDO_READ 功能块实例 (ReadObject) 会自动执行，以进行连续读取。

变量 DataSize（UINT 类型）：

- 初始化为数据缓冲区大小（DataBuffer: N 字节的数组）（当功能块的 ENABLE 输入为 FALSE 时，且在启动下一个执行之前）。
- 包含在功能块输出 CONFIRM 的上升沿上读取的数据的大小（以字节为单位）（该示例不演示如何从数据缓冲区提取值，也不演示如何管理错误检测）。



7.2 CIA405.SDO_READ4 : 读取最多 4 个字节的 CANopen 对象

读取最多 4 个字节的 CANopen 对象

功能块描述

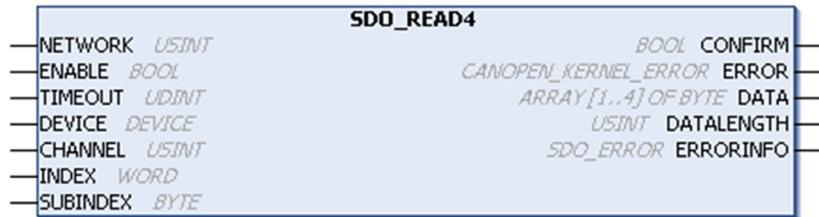
CIA405.SDO_READ4 功能块用于通过 SDO 消息读取指定设备的最多 4 个字节的 CANopen 对象。

必须向该功能块传递以下这些特定参数：

- 设备节点 ID
- SDO 客户端 / 服务器通道（缺省情况下，仅定义一个通道）
- CANopen 对象索引 / 子索引

如果读取成功完成，则该功能块会返回读取的对象大小。数据在一个 4 字节数组中提供。

图形表示形式



特定输入变量描述

输入	类型	说明
DEVICE	DEVICE (USINT)	CANopen 设备的节点 ID。 0 (缺省值) = 本地设备 (控制器) 1...127 = 设备节点 ID
CHANNEL	USINT	SDO 通道编号。 缺省值 = 1
INDEX	字	对象索引。 范围: 0000 (十六进制) ...FFFF (十六进制)
SUBINDEX	字节	对象子索引。 范围: 00 (十六进制) ..FF (十六进制)

特定输出变量描述

输出	类型	说明
DATA	ARRAY[1...4] OF BYTE	接收从设备对象读取的数据的数据数组。
DATALength	USINT	读取的对象的大小 (以字节为单位)。
ERRORINFO	SDO_ERROR (UDINT)	当 ERROR 输出 = 1 时, 返回 SDO 中止消息内容 (大小为 4 个字节)。

注意: 有关功能块通用 I/O 描述和执行行为的信息, 请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

下表显示对象大小及相应的 DATA 数组内容。

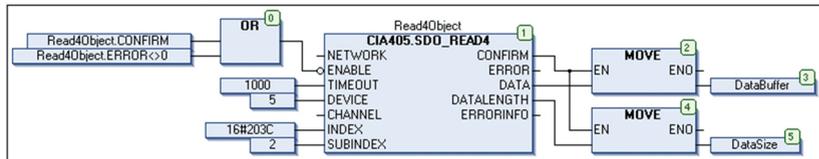
对象大小 示例	DATALENGTH	DATA(1)	DATA(2)	DATA(3)	DATA(4)
1 字节 01 (十六进制)	1	01 (十六进制)	无效	无效	无效
2 个字节 01 23 (十六进制)	2	LSB 23 (十六进制)	MSB 01 (十六进制)	无效	无效
3 个字节 01 23 45 (十六进制)	3	LSB 45 (十六进制)	23 (十六进制)	MSB 01 (十六进制)	无效
4 个字节 01 23 45 67 (十六进制)	4	LSB 67 (十六进制)	45 (十六进制)	23 (十六进制)	MSB 01 (十六进制)

LSB = 低有效字节

MSB = 最高有效字节

示例

以下示例显示了如何在超时 1 秒（1000 毫秒）的情况下，读取连接到第一个 CAN 总线接口的 CANopen 节点 5 的对象索引 203C（十六进制）/子索引 02（十六进制）。CIA405.SDO_READ4 功能块实例 (Read4Object) 会自动执行，以进行连续读取。变量 DataBuffer（4 字节数组）包含最后读取的数据的值。变量 DataSize（USINT 类型）包含最后读取的数据的大小（最大 4 字节）。该示例不演示如何管理错误检测。



7.3 CIA405.SDO_WRITE : 写入任意大小的 CANopen 对象

写入任意大小的 CANopen 对象

功能块描述

CIA405 .SDO_WRITE 功能块用于通过 SDO 消息写入指定设备的任意大小的 CANopen 对象。

必须向该功能块传递以下这些特定参数：

- 设备节点 ID
- SDO 客户端 / 服务器通道（缺省情况下，仅定义一个通道）
- CANopen 对象索引 / 子索引
- SDO 模式（定义数据传输模式，有关详细信息，请参阅 SDO_MODE ENUM（参见第 54 页））
- 用来存储要写入的对象值的数据缓冲区的指针
- 要写入的字节数

注意：如果要写入的对象的大小小于或等于 4 个字节，则使用 CIA405 .SDO_WRITE4 功能块。

图形表示形式



特定输入变量描述

输入	类型	说明
DEVICE	DEVICE (USINT)	CANopen 设备的节点 ID。 0 (缺省值) = 本地设备 (控制器) 1...127 = 设备节点 ID
CHANNEL	USINT	SDO 通道编号。 缺省值 = 1
INDEX	字	对象索引。 范围: 0000 (十六进制) ..FFFF (十六进制)
SUBINDEX	字节	对象子索引。 范围: 00 (十六进制) ...FF (十六进制)
MODE	SDO_MODE (有关详细信息, 请参阅下文)	数据传输模式。 0 (缺省值) = AUTO (表示自动模式选择)
DATA	POINTER TO BYTE	用来存储要写入的对象值的数据缓冲区的地址。 必须使用 ADR 标准功能定义关联指针。
DATALLENGTH	UINT	要写入的对象的大小 (以字节为单位)。 注意: 使用 SIZEOF 标准功能可定义数据缓冲区大小。

特定输出变量描述

输出	类型	说明
ERRORINFO	SDO_ERROR (UDINT)	当 ERROR 输出 = 1 时, 返回 SDO 中止消息内容 (大小为 4 个字节)。

注意: 有关功能块通用 I/O 描述和执行行为的信息, 请参阅功能块通用 I/O 和行为 (参见第 25 页) 一章。

CIA405 .SDO_MODE ENUM

CIA405 .SDO_MODE 枚举类型包含 SDO 传输模式的列表。

枚举器	值	说明
AUTO	0	自动模式选择。
EXPEDITED	1	用于最多 4 个字节的数据的 SDO 加速模式。 数据在一个 SDO 请求中发送。
SEGMENTED	2	用于超过 4 个字节的数据的 SDO 分段模式。 数据被分割为 7 字节的段，通过连续的 SDO 确认请求发送。
BLOCK	3	用于超过 4 个字节的数据的 SDO 块模式。 通过连续帧发送的数据被分割为 7 字节的数据块，这些块不会进行确认。接收到所有块后，接收方会发送确认。 注意： 这是较快的传输模式，但是您的设备可能不支持，因为此模式是近期的 CANopen 配置附加项。

示例

采用 ST 语言在 CANopen 设备中读取和写入数据的程序示例：

在节点地址 1 上声明一个 ATV71

在节点地址 2 上声明一个 ATV31

要写入或读取的数据为：

INDEX:=16#203C,

SUBINDEX:=2,

写入的值 (SDOWrite_data) 为在每个循环上递增的 0 到 500 的斜坡。

```
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 : 写入最多 4 个字节的 CANopen 对象

写入最多 4 个字节的 CANopen 对象

功能块描述

CIA405.SDO_WRITE4 功能块用于通过 SDO 消息写入指定设备的最多 4 个字节的 CANopen 对象。

必须向该功能块传递以下这些特定参数：

- 设备节点 ID
- SDO 客户端 / 服务器通道（缺省情况下，仅定义一个通道）
- CANopen 对象索引 / 子索引
- 要写入的值
- 要写入的字节数（对象大小）

图形表示形式



特定输入变量描述

输入	类型	说明
DEVICE	DEVICE (USINT)	CANopen 设备的节点 ID。 0（缺省值）= 本地设备（控制器） 1...127 = 设备节点 ID
CHANNEL	USINT	SDO 通道编号。 缺省值 = 1

输入	类型	说明
INDEX	WORD	对象索引。 范围：0000（十六进制）..FFFF（十六进制）
SUBINDEX	BYTE	对象子索引。 范围：00（十六进制）..FF（十六进制）
DATA	ARRAY[1...4] OF BYTE	用来存储要写入的对象值的数据数组。
DATALength	USINT	要写入的对象的大小（以字节为单位）。

特定输出变量描述

输出	类型	说明
ERRORINFO	SDO_ERROR (UDINT)	当 ERROR 输出 = 1 时，返回 SDO 中止消息内容（大小为 4 个字节）。

注意：有关功能块通用 I/O 描述和执行行为的信息，请参阅功能块通用 I/O 和行为（参见第 25 页）一章。

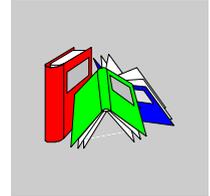
下表显示对象大小及相应的 DATA 数组内容。

对象大小 示例	DATALength	DATA(1)	DATA(2)	DATA(3)	DATA(4)
1 字节 01（十六进制）	1	01（十六进制）	无效	无效	无效
2 字节 01 23（十六进制）	2	LSB 23（十六进制）	MSB 01（十六进制）	无效	无效
3 字节 01 23 45（十六进制）	3	LSB 45（十六进制）	23（十六进制）	MSB 01（十六进制）	无效
4 字节 01 23 45 67（十六进制）	4	LSB 67（十六进制）	45（十六进制）	23（十六进制）	MSB 01（十六进制）

LSB = 低有效字节

MSB = 最高有效字节

术语



0-9

协议

设备在网络上进行相互通讯时使用的一组规则

毫秒

毫秒

C

CAN

控制器区域网络 (CAN) 是一种串行总线系统，其目的是实现嵌入式控制。

CANopen

CANopen 是用于嵌入式控制系统的符合国际标准 (EN 50325-4) 的基于 CAN 的高层次协议。

CiA

CAN in Automation (CiA) 是用户和制造商组成的国际组织，致力于开发和支持 CANopen 及其他基于 CAN 的高层次协议。

CiA405

IEC 61131-3 可编程控制器的 CANopen 接口和设备配置文件

COB

CANopen 协议通讯对象

E

EDS

现场总线设备描述的电子数据表文件。

EMCY

紧急

H

hex

十六进制

I

I/O

输入 / 输出

ID

标识符

N

NMT

CANopen 网络管理

O

OD

CANopen 协议对象字典

P

PDO

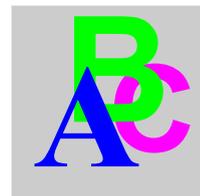
CANopen 协议过程数据对象

S

SDO

CANopen 协议服务数据对象

索引



SDO 访问功能块

- CIA405.SDO_READ: 读取任意大小的 CANopen 对象, 46
- CIA405.SDO_READ4: 读取最多 4 个字节的 CANopen 对象, 49
- CIA405.SDO_WRITE: 写入任意大小的 CANopen 对象, 52
- CIA405.SDO_WRITE: 写入最多 4 个字节的 CANopen 对象, 57

功能块描述, 23

- SDO 访问功能块, 45
- 功能块通用 I/O 和行为, 25
- 查询状态功能块, 41
- 网络管理功能块, 31
- 自有节点 ID 功能块, 39

功能块通用 I/O 和行为

- 通用 I/O 和行为, 25

查询状态功能块

- CIA405.GET_CANOPEN_KERNEL_STATE: 获取 CANopen 内核状态, 42
- CIA405.GET_STATE: 获取 CANopen 设备状态, 43

简介, 9

- CANopen 协议简介, 11
- CiA405 简介, 21

网络管理功能块

- CIA405.NMT: 设备 NMT 状态管理, 32
- CIA405.RECV_EMCY_DEV: 获取设备 EMCY 消息, 36
- CIA405.RECV_EMCY: EMCY 消息扫描, 35

自有节点 ID 功能块

- CIA405.GET_LOCAL_NODE_ID: 获取控制器 CANopen 节点 ID, 39

