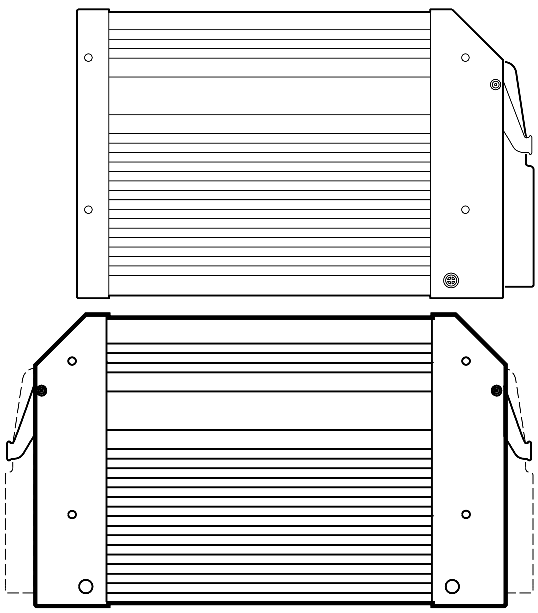
原版编程手册
ClassicController

CR0033
带I / O模块：**CR0133**

运行时系统 v02.00.03
CODESYS® V2.3

中文



7391022_03_CN 2017-07-25

内容

1 关于本手册

内容

202

1.1 版权

6088

我们页面上使用的所有产品名称、图片、公司或其他品牌均是相应的权利所有者之财产：

- AS-i 是 AS-International Association (→ www.as-interface.net) 的财产
- CAN 是德国 CiA (CAN in Automation e.V.) (→ www.can-cia.org) 的财产
- CODESYS™ 是德国 3S – Smart Software Solutions GmbH (→ www.codesys.com) 的财产
- DeviceNet™ 是美国 ODVA™ (Open DeviceNet Vendor Association) (→www.odva.org) 的财产
- EtherNet/IP® 是 →ODVA™ 的财产
- IO-Link® www.io-link.com 是德国 → PROFIBUS Nutzerorganisation e.V. 的财产
- ISOBUS 是德国 AEF – Agricultural Industry Electronics Foundation e.V.
- Microsoft Corporation (→ www.microsoft.com) 的财产
- PROFIBUS® 是德国 PROFIBUS Nutzerorganisation e.V. (→www.profibus.com) 的财产
- PROFINET® 是 →德国 PROFIBUS Nutzerorganisation e.V. 的财产
- Windows® 是→美国 Microsoft Corporation 的财产

1.2 概述： ecomatmobile 装置文档模块

22954
17405

ecomat*mobile* 装置文档包含以下模块：

1.	**技术资料**
内容	表格中的技术数据
来源	→ www.ifm.com > 选择所在国家/地区 > [Data sheet search] > CR0033 > [Technical data in PDF format]

2.	**安装说明/操作说明**
内容	关于安装、电气安装、（调试*）、技术数据的说明
来源	设备随附说明。 IFM 主页亦可提供。

*) 括号中的描述仅包含于特定装置的说明中。

3.	**编程手册 + 联机帮助**
内容	装置软件配置和功能的说明
来源	→ www.ifm.com

4.	**系统手册“ecomatmobile 专门知识”**
内容	关于以下主题的专门知识： • 概述模板和演示程序 • CAN、 CANopen • 控制输出端 • 用户闪存 • 可视化 • 使用的文件和库的概述
来源	→ www.ifm.com

5.	**系统手册“IFM 控制器 ISOBUS”**
内容	装置 ISOBUS 软件配置和功能的说明
来源	→ www.ifm.com > 因此，"CRnnnn" 表示预装 ISOBUS 的装置的产品编号。

1.3 符号和格式是什么意思？

203

以下符号或图表利用图片阐明我们的说明中的注释。

⚠ 警告

可能导致死亡或严重的不可逆伤害。

⚠ 小心

可能导致轻微的可逆伤害。

通知

预计或可能会导致财产损失。

!	关于故障或干扰的重要说明
ℹ	其他备注
▶ ...	操作请求
> ...	反应，结果
→ ...	“查看”
abc	交叉引用
123 0x123 0b010	十进制数 十六进制数 二进制数
[...]	按键、按钮或指示标记

1.4 本文档的结构是怎样的？

16416
1508

本文档融合了各类手册。 它适用于初学者，亦可作为高级用户的参考。 本文档针对应用程序的程序员。

本手册的使用方式：

- 请参阅目录以选择具体的主题。
- 您还可利用索引快速找到您正在查找的术语。
- 在章节的开始，我们将简单地讲述其内容。
- 缩写词和技术术语→ 附录

若发生故障或有相关疑问，请与制造商联系：
→ www.ifm.com

我们想要变得更好！ 每个单独的部分在右上角有 ID 编号。 如果您想要告知我们任何不一致的情况，则跟本文档的标题和语言一起指出该编号。 由衷地感谢您的支持！

我们保留作出修改的权利，因此文档的内容可能会更改。 您可在 **IFM** 网站找到当前的版本：
→ www.ifm.com

⇒ 我们的在线帮助通常会第一时间更新。

⇒ PDF 仅定期更新。

16420

注意

这些说明适用于不含和包含集成 I/O 模块的装置。

► 在两种情况下，均确保针对装置 CR0033 设定 PLC 配置！

您可在以下位置找到更多关于集成 I/O 模块的信息：
→ 章节 **集成 I/O 模块： 说明** (→ 页 294)，本文档附录。

1.5 说明沿革 (CR0033 + CR0133)

15794

本手册更改了哪些内容？ 概述

日期	主题	更改：
2013-06-24	各种	新文档结构
2014-02-03	集成 I/O 模块	CR0133 说明已添加
2014-04-28	各种功能块	更准确地说明功能块输入端 CHANNEL
2014-06-24	FB PID2	图形已修正
2014-06-30	文档名称	“系统手册”重新命名为“编程手册”
2014-07-04	装置输出端 ERROR（夹具 13）	输出端不可用。 参考说明已移除。
2014-07-24	FB INPUT_ANALOG FB SET_INPUT_MODE FB SET_OUTPUT_MODE	针对 ERROR 输出端的可能错误代码已添加
2014-07-31	章节“错误代码、诊断”	错误代码表已添加
2014-07-31	FB PHASE	修正了输入端参数 C、ET 说明
2014-07-31	FB OUTPUT_CURRENT_CONTROL	如果预设值 = 0 mA >>，在“100 ms 之内”控制到 0，“而非立刻”
2014-08-08	章节“集成 I/O 模块输入端”	通过“模拟输入端”和“二进制输入端” 部分完成
2014-08-08	章节“集成 I/O 模块的对象目录”	标题中，“对象目录”替换“SDO”
2014-08-26	输入端、输出端的说明	正极开关/负极开关替代高侧/低侧
2014-08-29	快速输入端	关于自动设定输入电阻的说明
2014-09-29	标志 OUT_OVERLOAD_PROTECTION	一般替换为“过载保护”
2014-11-12	章节“输出端（技术）”	部分“二进制输出端诊断”已增补或修正
2015-01-13	针对错误代码、系统标志的文档结构	• 错误标志： 现在仅位于附录，章节**系统标志** • CAN / CANopen 错误和错误处理： 现在仅位于系统手册“技巧” • 错误代码、EMCY 代码： 现在位于附录，章节**错误表**
2015-03-10	可用内存	完善了说明
2015-05-26	FB J1939_x_GLOBAL_REQUEST	更准确的说明
2015-06-10	各种功能块	修正了 FB 输入 CHANNEL 说明
2015-07-20	输入端 IN12...IN15	现无工作模式 19
2015-07-27	FB GET_IDENTITY	已通过输出端 SERIALNUMBER 添加

2015-08-04	输出端 OUT00...01	通过电流和电压测量诊断
2015-08-24	I/O 模块 ExB01	现删除未实现的 I/O 模式
2015-10-22	系统标志位 SERIAL_MODE	不可通过 USB 调试应用程序
2016-04-27	快速输入端的 FB	添加了频率较高时的注意事项
2017-01-04	FB INC_ENCODER	可搭配**一个**功能块 FAST_COUNT、FREQUENCY、FREQUENCY_PERIOD、PERIOD、PERIOD_RATIO、PHASE，在装置标准侧同一个输出端使用。
2017-01-13	CODESYS 2.3 软件手册	从 IFM 主页下载的提示已移除
2017-01-13	ISOBUS 系统手册	从 IFM 主页下载的提示已添加
2017-02-22	FB INC_ENCODER	可搭配**一个**功能块 PERIOD、PERIOD_RATIO、PHASE，在装置标准侧同一个输出端使用。

2 安全说明

内容

213

2.1 请注意！

214
11212

本手册中提供信息、注意事项和示例并不对任何特性作出担保。 并不通过提供的图纸、展示和示例，为系统承担任何责任，且提供的图纸、展示和示例也未将任何特定应用的特殊性考虑在内。

► 机器/设备的制造商应负责确保机器/设备的安全。

► 遵守机器/装置投放至市场所在国家的国内和国际法规。

⚠ 警告

未遵守这些说明可能会导致财产损失或人身伤害。

ifm electronic gmbh 不承担此方面的任何责任。

► 对此装置和使用其执行工作前，操作者必须已阅读并理解安全说明和本手册中的相应章节。

► 操作者必须获得对该机器/设备执行工作的授权。

► 操作者必须已获得执行此工作所需的资质和培训。

► 遵照装置的技术资料操作！
您可在此处的 **IFM** 主页上找到当前的技术资料：
→ www.ifm.com > 选择您所在国家 > [技术资料查询] > (货号。) > [PDF 技术资料]

► 注意安装和配线信息，以及装置的功能和特性！
→ 随附的安装说明或在 **IFM 的**主页上：
→ www.ifm.com > 选择您所在国家 > [技术资料查询] > (货号。) > [操作使用说明书]

► 请注意 **IFM** 网站上提供的现有文档发行说明中的修正内容和注意事项：
→ www.ifm.com > 选择您所在国家 > [技术资料查询] > (货号。) > [操作使用说明书]

5020

通知

串行接口的驱动器模块可能会受损！

运行时断开或连接串行接口可能导致不明确的状态并损坏驱动器模块。

► 切勿在运行时断开或连接串行接口。

2.2 需要预先具备哪些知识？

215

本文件针对了解控制技术以及根据 IEC 61131-3 进行 PLC 编程的人群。

若要对 PLC 进行编程，用户还应熟悉 CODESYS 软件。

本文档供专业人士使用。 专业人士是指经过专业技能培训有丰富的实践经验，能够预见和避免在操作和维护产品期间的风险及危险。 本文档包含正确操作产品的相关信息。

使用产品前请阅读本文档，以了解操作条件、安装和操作。 使用装置期间，请始终妥善保管本文档。

请遵守安全说明。

2.3 控制器的启动运行状况

6827
15233
11575

⚠ 警告

由于机器或设备部分意外和危险启动导致的危险！

► 创建程序时，程序员必须确保发生故障（例如紧急停机）和随后执行故障排除后，机器或设备部分不会发生任何意外和危险启动！

  ➩ 实现重新启动抑制。

► 发生错误时，在程序中将相关的输出功能设为“错误”！

诸如以下情况可能会导致重新启动：

- 供电中断后电压恢复
- 由于周期时间过长，电子狗反应后复位
- 紧急停机后的错误排除

若要确保控制器的安全运行状况：

► 在应用程序中监控电压供应器。

► 若发生错误，在应用程序中关闭所有相关的输出功能。

► 在应用程序（反馈）中监控可能会导致危险移动的执行器。

► 在应用程序（反馈）中监控可能会导致危险移动的继电器触点。

► 如有必要，在应用程序中确保焊接的继电器触点无法触发或继续危险移动。

2.4 注释： 序列号

20780

► 在用户的工厂，制作一幅机器控制器网络图。 将安装的每一个控制器的序列号输入至网络图。

► 在下载软件组件之前，读出该序列号并检查网络图，以确保您访问的是正确的控制器。

2.5 注释： TEST 输入端

20781

► 机器中所有控制器的 TEST 输入端应单独配线并清晰标注，以便能够正确配置至控制器。

► 服务访问期间，仅启用待访问控制器的 TEST 输入端。

3 系统描述

内容

975

3.1 关于装置的信息

10415

本手册讲述针对 **ifm electronic gmbh** 移动机器的 **ecomat*mobile*** 系列：

- ClassicController: CR0033

3.2 硬件说明

内容

14081

3.2.1 硬件结构

内容

15332

启动条件

19658

为电源接头 VBBs（如标准侧继电器供给）和夹具 15 提供充足的电压之前，装置不会启动。
在车辆中，夹具 15 是通过点火锁开关的正极电缆。

- 允许的工作电压 = 8...32 V
- 启动条件： VBBs > 10 V

继电器

19661

ClassicController 有 2 个内部输出继电器，可各自将 8 个输出端与端子电压 VBBx (x = o | r) 隔开。
继电器仅在以下条件下启用：

- 全局位 ERROR = FALSE

且

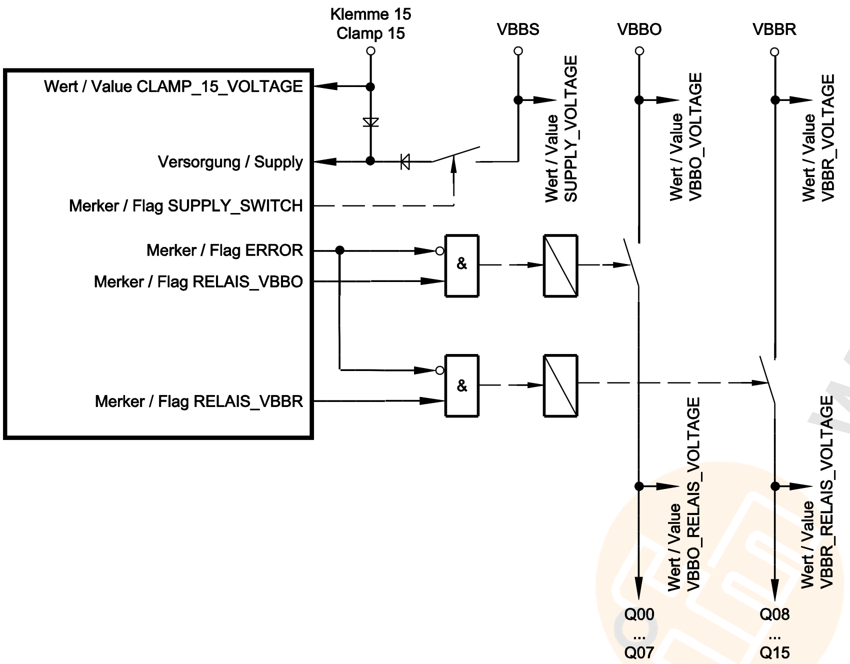
- 位 RELAIS_VBBx = TRUE

在有效条件下，继电器触点将输出端连接至端子电压 VBBx。

(!)启用相应的输出端，时间不早于继电器通电后 ≥ 45 ms！

原理方块图

19662



图： 电源和继电器原理方块图

可用内存

13736

闪存

8136

FLASH 内存（非易失慢速内存） 装置现有全部	2 176 K 字节

由此，如下内存区域将预留用于 ...

应用程序最大大小	1 280K 字节
除应用程序之外的数据 用户可写入文件、位图、字体等数据	128K 字节
除应用程序之外的数据 数据读取可通过 ***FLASHREAD*** (→ 页 244)数据写入可通过 ***FLASHWRITE*** (→ 页 245) （文件： 就标头而言少 128 字节	64K 字节

剩余的内存保留用于系统内部用途。

SRAM

8360

SRAM（易失快速内存） 装置现有全部 SRAM 在此表示所有类型的易失快速内存。	2 216K 字节

由此，如下内存区域将预留用于 ...

应用程序保留的数据	192K 字节

剩余的内存保留用于系统内部用途。

FRAM

20794

FRAM（非易失快速内存） 装置现有全部 FRAM 在此表示所有类型的非易失快速内存。	128K 字节

由此，如下内存区域将预留用于 ...

应用程序中的变量，声明为 VAR_RETAIN	4K 字节
作为剩余定义标志（始于 %MB0...） ▶ 通过以下 FB 设定内存区域的末尾： ***MEMORY*_RETAIN_PARAM** (→ 页 241)!	4K 字节
用户可免费使用剩余内存 访问可通过 ***FRAMREAD*** (→ 页 248)和 ***FRAMWRITE*** (→ 页 249)	16K 字节
用户可免费使用剩余内存 可通过地址运算符访问。	64K 字节

剩余的内存保留用于系统内部用途。

3.2.2 延迟关闭的工作原理

993

如果 **ecomat*mobile*** 控制器断开电源电压（点火开关关闭），则所有输出端均立刻正常关闭，输入端信号不再被读取，控制器软件（运行时系统和应用程序）的处理中断。 该情况与控制器当前的程序步骤无关。

如果未作请求，则必须通过程序关闭控制器。 关闭点火开关有助于保存内存状态等。

ClassicController 可利用相应的电源电压输入端连接以及相关系统标志的评估，通过程序来关闭。 本章方块图 **硬件结构** (→ 页 17) 显示了单个电流回路的来龙去脉。

将端子 VBB15 连接至点火开关。

2418

如果在端子 VBBs 提供电源电压，则内部 PLC 电子组件通过端子 VBB15 初始化。

这些端子 VBB15 和 VBBs 在内部监控。 提供的端子电压 VBB15 可通过系统标志 CLAMP_15_VOLTAGE 监控。 提供的端子电压 VBBs 可通过系统标志 SUPPLY_VOLTAGE 监控。

锁存

2419

控制器的通电：

- 电压通过点火开关提供给 VBB15（夹具 15*）。
- 系统标志 CLAMP_15_VOLTAGE 识别已提供的电压并启用系统标志 SUPPLY_SWITCH。
- SUPPLY_SWITCH activates the connection to the potential VBBs.

\> 点火开关忽略。 控制电压的锁存确定。

通过夹具 15 关闭控制器电源：

- 系统标志 CLAMP_15_VOLTAGE 识别端子 VBB15 上电源电压的关闭。

► 在应用程序中复位系统标志 SUPPLY_SWITCH。

\> 经由 VBBs 的锁存移除，控制器完全关闭。

*) 在车辆中，夹具 15 是通过点火锁开关的正极电缆。

3.2.3 继电器： 重要说明！

12976

分配继电器 - 电位： → 数据表
单位继电器触点（= 单位输出组）的最大总电流： → 数据表

通知

可能会损坏继电器触点！
在紧急情况下，“粘附”继电器触点不再断开输出端的电源！

如果 VBBS (VBBrel) 和夹具 15 同时断开电源，但电位 VBBx 保持与其连接，那么继电器会停止，甚至在系统禁用输出端之前就停止。

在此情况下，继电器分隔输出端和电源**欠载**。 这样会显著减少继电器的寿命周期。

► 如果 VBBx 永久连接至电源：
  • 另永久连接 VBBS (VBBrel) 并
  • 在夹具 15 的帮助下通过程序关闭输出端。

3.2.4 监控概念

内容

991

控制器监控电源电压和系统错误标志。 视状态而定...

- 控制器关闭内部继电器
  - > 输出端断电，但保留逻辑状态

或：

- 运行时系统禁用控制器
  - > 程序停止
  - > 输出端更改为逻辑 "0"
  - > 状态 LED 熄灭

监控电源电压 VBBx

6752

若出现故障，则区分 2 种情况：

端子电压 VBBx 降至限值 5.25 V 以下

15752

> 控制器检测欠压。 端子电压 VBBx 供电的输出端禁用。

> 如果端子电压恢复并返回至正常范围 (> 10 V)，则输出端再次启用。

13975

⚠ 警告

可能存在危险的重启！
存在人身伤害的风险！ 存在机器/设备材料损坏的风险！

如果在故障情况下输出端通过硬件关闭，应用程序生成的逻辑状态不会改变。

► 补救措施：
- 复位应用程序中的输出逻辑！
- 排除故障！
- 根据状况复位输出端。

端子电压 VBBs 降至限值 10 V 以下

20638

> 控制器继续运行，直至电压下降到由其形成的内部电压也下降。

ⓘ 低于 10 V 时，不保存保留数据。 → 标志 RETAIN_WARNING

> 如果内部电压下降，控制器则转至复位。
运行时和应用程序的执行中断。
该情况与 PLC 当前的程序步骤无关。

> 在电源电压再次高于限值之前，不执行控制器重启操作。

监控概念的工作原理

2421

⚠ 警告

意外关闭所有输出端会造成危险！

如果监控程序检测到系统错误：

> 装置会禁用所有输出端的电源。

程序处理期间，输出继电器完全由用户通过软件控制。 因此，安全链的平行触点等可评估为输入信号，且可相应地关闭输出继电器。 为了安全，必须遵守相应的适用国家法规。

如果在程序处理期间出现了错误，则可利用系统标志位 ERROR 关闭继电器，以断开关键的设备段。

❗手动设定标志位 ERROR_VBB 不会对继电器造成影响！

11575

⚠ 警告

由于机器或设备部分意外和危险启动导致的危险！

► 创建程序时，程序员必须确保发生故障（例如紧急停机）和随后执行故障排除后，机器或设备部分不会发生任何意外和危险启动！
  ➪ 实现重新启动抑制。

► 发生错误时，在程序中将相关的输出功能设为“错误”！

❗如果出现电子狗错误，…

> 程序处理自动中断
> 输出端变成无电流状态并转至逻辑 "0"
> 控制器复位
> 而后控制器在通电后再次启动。

参考电压输出端

13934

参考电压输出端用于为传感器提供不受电源电压波动影响的稳定电压。

13402

通知

参考电压输出端可能受损！

► 切勿施加任何外部电压！

电压通过二进制系统变量 REFERENCE_VOLTAGE_5 或 REFERENCE_VOLTAGE_10 在参考电压输出端 [V_{REF} OUT] 上设定。

REFERENCE_VOLTAGE_10	REFERENCE_VOLTAGE_5	参考电压 [V_{REF} OUT]
FALSE	FALSE	0 V
FALSE	TRUE	5 V
TRUE	FALSE	10 V
TRUE	TRUE	0 V

► 如果选择参考电压 = 10 V：
最低为控制器提供 13 V 电压！

► 通过系统变量 REF_VOLTAGE 在参考电压输出端监控电压。

3.2.5 输入端（技术）

内容

14090

模拟输入端

15446

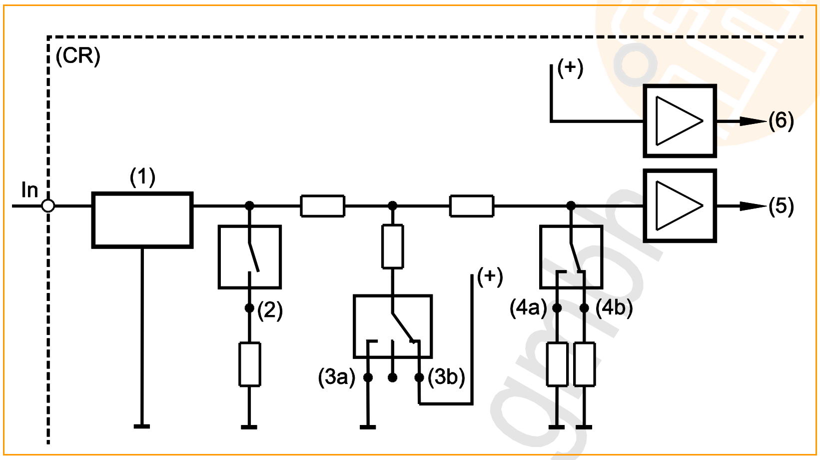
模拟输入端可通过应用程序配置。 可按以下方式设定测量范围：
• 电流输入 0...20 mA
• 电压输入 0...10 V
• 电压输入 0...32 V
• 电阻测量 16...30 000 Ω / 3...680 Ω（GND 测量）

还可进行电压比率测量（0...1000 ‰，可通过功能块调整）。 也就是说，可在没有额外参考电压的情况下评估电位计和操纵杆。 电源电压的波动不会对该测量值产生影响。

作为备选，还可通过二进制的方式评估模拟通道。

ⓘ 如果是比率测量，则连接的传感器应供给装置 VBBs。 因此，避免了偏移电压导致的错误测量。

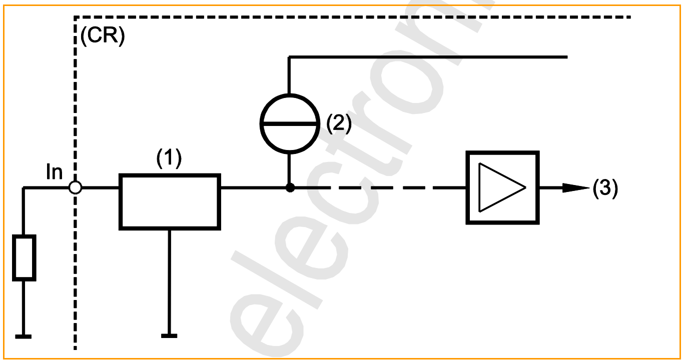
8971



In = 针形多功能输入端 n
(CR) = 装置
(1) = 输入过滤器
(2) = 模拟电流测量
(3a) = 二进制输入端正极开关
(3b) = 二进制输入端负极开关
(4a) = 模拟电压测量 0...10 V
(4b) = 模拟电压测量 0...32 V
(5) = 电压
(6) = 参考电压

图： 多功能输出端原理方块图

8972



图： 电阻器测量输入端方块图

In = 针形电阻器输入端
(CR) = 装置
(1) = 输入过滤器
(2) = 恒流源
(3) = 电压

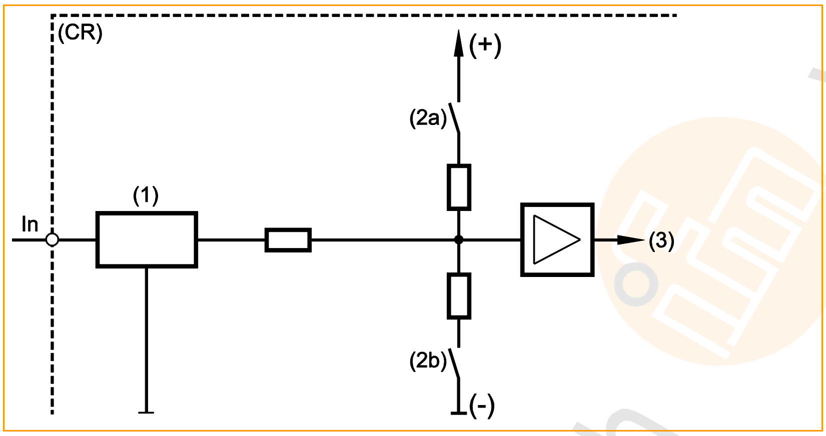
二进制输入端

1015
7345

二进制输入端可在以下模式下运行：

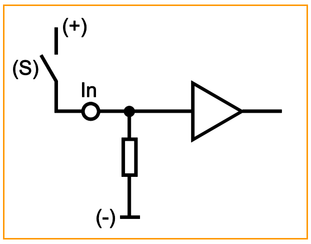
- 二进制输入端正极开关 (BL)，针对正极性传感器信号
- 二进制输入端负极开关 (BH)，针对负极性传感器信号

根据装置，而仅是输入端可进行不同的配置。 除了针对干扰的保护机制之外，二进制输入端还通过模拟阶段进行内部评估。 这样有助于诊断输入端信号。 但在应用程序软件中，开关信可号直接用作位信息。



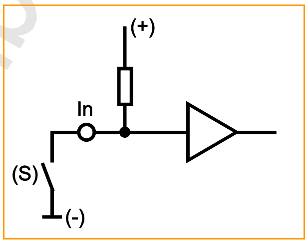
In = 针形二进制输入端 n
(CR) = 装置
(1) = 输入过滤器
(2a) = 输入端负极开关
(2b) = 输入端正极开关
(3) = 电压

图： 针对负极性和正极性传感器信号的二进制输入端负极开关/正极开关基本电路



In = 针形二进制输入端 n
(S) = 传感器

二进制输入端正极开关 (BL) 基本电路
针对正极性传感器信号：
输入端 = 开启 ⇨ 信号 = 低 (GND)



In = 针形二进制输入端 n
(S) = 传感器

二进制输入端负极开关 (BH) 基本电路
针对负极性传感器信号：
输入端 = 开启 ⇨ 信号 = 高 (电源)

就这些输入端中的某些而言 (→ 技术资料) ，可选择要切换到的电势。

输入组 I00...I11

19315

这些输入端是一组多功能通道。

这些输入端用途如下（每个输入端均可单独配置）：

- 模拟输入端 0...20 mA
- 模拟输入端 0...10 V
- 模拟输入端 0...32 V
- 电压比率测量 0...1000 ‰
- 二进制输入端正极开关 (BL)，针对正极性传感器信号（含/不含诊断）
- 二进制输入端负极开关 (BH)，针对负极性传感器信号
- 针对递增编码器和频率或时间间隔测量等的快速输入端

→ 章节 **可能的输入端/输出端工作模式** (→ 页 289)

可评估包含诊断功能的 NAMUR 传感器。

所有输入端显示关于功能和诊断的相同状况。但是：
I00...I07: 频率测量值受电源电压影响的多功能输入端.
I08...I11: 频率测量值固定的多功能输入端.

详细说明 → 章节 **输入端/输入端地址分配**

► 每个输入端均可通过应用程序进行配置：
- FB **INPUT_ANALOG (→ 页 157)** > 输入端 MODE 或：
- FB **SET_INPUT_MODE** (→ 页 161)> 输入端 MODE
- 包含以下 FB 的快速输入端：

FAST_COUNT (→ 页 171)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 173)	测量到达所选通道的信号的频率。
FREQUENCY_PERIOD (→ 页 175)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）
INC_ENCODER (→ 页 178)	针对编码器评估的递增/递减计数器功能
INC_ENCODER_HR (→ 页 181)	针对编码器高分辨率评估的递增/递减计数器功能
PERIOD (→ 页 183)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）
PHASE (→ 页 187)	读取包含快速输入端的通道对并对比信号的相位

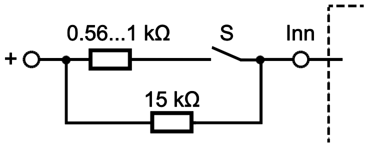
> 如果针对电流测量配置模拟输入端，则装置切换至安全电压测量范围 (0...32V DC)，超过终值时 (> 21.7 mA)，在标志字节 ERROR_CURRENT_Ix 中设定相应的错误位。
如果电流值再次低于限值，则装置每秒钟检查一次。 当值再次低于限值时，输入端会自动切回至电流测量范围。

► 针对 NAMUR： 如果要使用诊断功能，则额外启用该模式：
- 功能块 SET_INPUT_MODE > 设定输入端 DIAGNOSTICS。

- 针对 NAMUR： 开关或传感器的有用信号不得高于系统电源电压。

非电子开关二进制信号 NAMUR 诊断：

► 为开关配置额外的电阻器连接件！



图： 输入端 Inn 非电子开关 S

13956

> 诊断结果可通过以下系统标志显示：

系统标志（符号名称）	类型	说明
ERROR_BREAK_Ix （0...x，值视装置而定→ 技术 资料）	DWORD	输入双字 x： 断线错误 或（电阻输入）： • 电源短路 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE： 错误 位 = FALSE： 无错误
ERROR_SHORT_Ix （0...x，值视装置而定→ 技术 资料）	DWORD	输入双字 x： 短路错误 仅当输入模式 = IN_DIGITAL_H 时 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE： 错误 位 = FALSE： 无错误

> 在应用程序中，系统变量 ANALOG00...ANALOGxx 可用于客户特定诊断。

输入组 I12...I15

10424

这些输入端是一组多功能通道。

这些输入端用途如下（每个输入端均可单独配置）：

• 二进制输入端正极开关 (BL)，针对正极性传感器信号（含/不含诊断）

• 电阻测量输入端（例如温度传感器或燃油传感器）

→ 章节 **可能的输入端/输出端工作模式** (→ 页 289)

可评估包含诊断功能的 NAMUR 传感器。

► 每个输入端均可通过应用程序进行配置：

• FB **INPUT_ANALOG** (→ **页 157**) > 输入端 MODE 或：

• FB **SET_INPUT_MODE** (→ 页 161)> 输入端 MODE

电阻测量

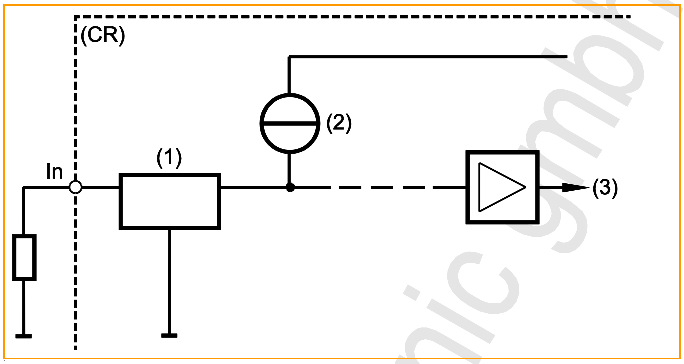
9773

这些输入端上的一般传感器：

• 液位

• 温度（PT1000、NTC）

8972



图： 电阻器测量输入端方块图

In = 针形电阻器输入端

(CR) = 装置

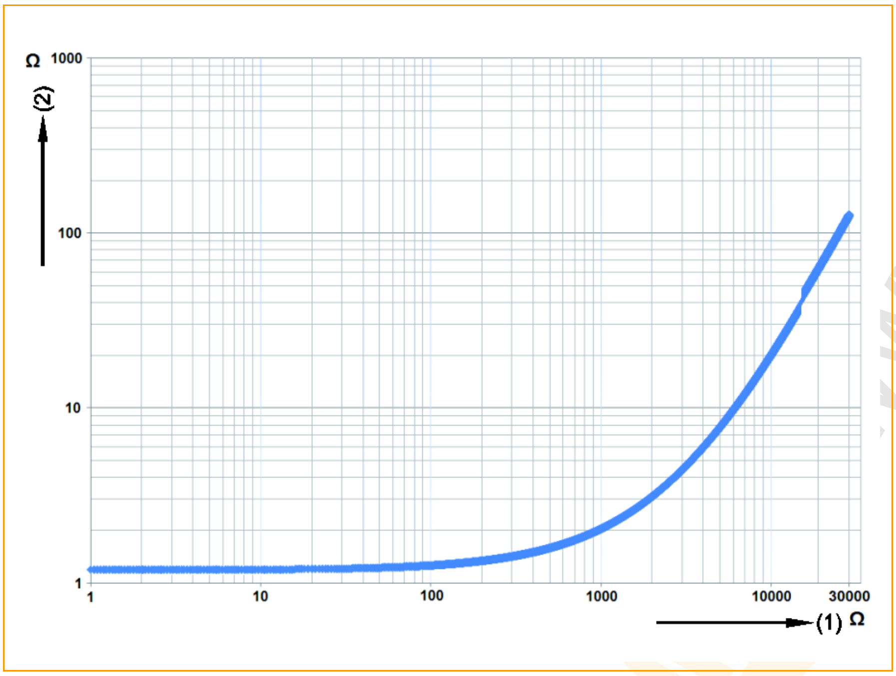
(1) = 输入过滤器

(2) = 恒流源

(3) = 电压

8970

该装置的电阻与电阻值不呈线性关系，→ 图：



图： 与电阻值相关的分辨率

(1) = 输出端电阻值

(2) = 分辨率

当输入端上的 A/D 转换器信号更改 1 个单位时，测量值更改多少 ohm？ 示例：
- 在 1...100 Ω 的范围时，分辨率为 1.2 Ω。
- 在 1 kΩ 的范围时，分辨率大约为 2 Ω。
- 在 2 kΩ 的范围时，分辨率大约为 3 Ω。
- 在 3 kΩ 的范围时，分辨率大约为 6 Ω。
- 在 6 kΩ 的范围时，分辨率大约为 10 Ω。
- 在 10 kΩ 的范围时，分辨率为 11 Ω。
- 在 20 kΩ 的范围时，分辨率大约为 60 Ω。

3.2.6 输出端（技术）

内容

14093

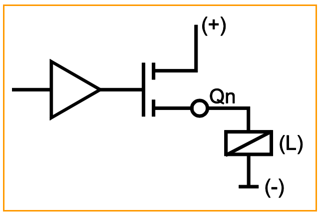
二进制输出端

14094

以下工作模式适用于装置输出端（(→ 技术资料）：

- 二进制输出端，正极开关 (BH)，含/不含诊断功能
- 二进制输出端，负极开关 (BL)，不含诊断功能

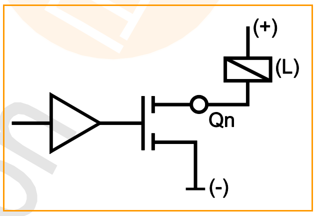
15450



Qn = 插脚输出端 n

(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号



Qn = 插脚输出端 n

(L) = 负载

输出端负极开关 (BL) 基本电路
针对负极性输出端信号

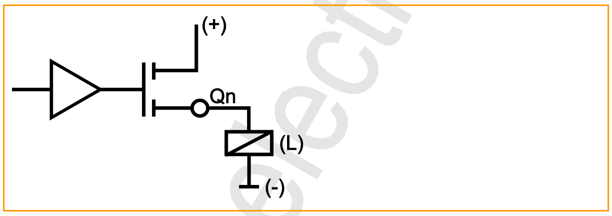
PWM 输出端

14095

以下工作模式适用于装置输出端（(→ 技术资料）：

- PWM 输出端，正极开关 (BH)，不含诊断功能

15451



Qn = 插脚输出端 n

(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号

输出端的保护功能

15248

在特定范围内，本装置的输出端有过载和短路保护。
→ 数据表

定义： 过载

15249

仅可在包含电流测量的输出端检测到过载。

过载即…
“超过最大额定电流的 12.5 %”。

定义： 短路

15250

所有包含诊断功能的输出端均可检测短路。
前提： 输出端未针对电流测量进行配置。

短路即…
“输出电压降至相应电源电压的 88%（测量值的 ± 2.5%）以下。”

> 仅可在输出端 = TRUE 时检测到接地故障。

输出端对过载或短路的反应

15251

输出端的自我保护

15333

硬件有自我保护功能，与输出端和故障检测的工作模式无关。 如果热负载太高（因短路或过载导致），输出驱动器开始计时。
如果输出端计时过长（几个小时），驱动器可能受损。

因此，我们建议：
在以下模式中操作包含诊断功能的装置输出端，因为在此情况下，软件可通过关闭以下对象来额外保护驱动器：
- 功能块 **SET_OUTPUT_MODE** (→ 页 191)> 输入端 DIAGNOSTICS = TRUE 且
- 功能块 SET_OUTPUT_MODE > 输入端 PROTECTION = TRUE

基于输出端工作模式的反应

15479

如果出现过载或短路，则输出端的状况取决于其工作模式（→ FB **SET_OUTPUT_MODE** (→ 页 191)> 输入端 DIAGNOSTICS 和 PROTECTION）：

- DIAGNOSTICS = FALSE 且 PROTECTION = FALSE：
  > 输出端继续运行。
- DIAGNOSTICS = TRUE 且 PROTECTION = FALSE：
  > 检测到错误并以错误代码的形式发出通知（→ 章节 **错误代码** (→ 页 373)).
  这取决于输出端的类型以及输出端的电流或电压。
  程序员可对程序中的错误作出反应。
- DIAGNOSTICS = TRUE 且 PROTECTION = TRUE：
  > 检测到错误并以错误代码的形式发出通知（→ 章节“错误代码”）。
  > 相应的输出端关闭。
  > 输出端逻辑状态不受此影响！
  > 如果错误已排除，控制器每秒钟检查。
  如果错误排除： 控制器再次开启输出端。

使用 PWM1000、OUTPUT_CURRENT_CONTROL、OUTPUT_BRIDGE 时的反应

15480

使用以下 FB 时有所不同：
- **PWM1000** (→ 页 203)
- **OUTPUT_CURRENT_CONTROL** (→ 页 200)
- **OUTPUT_BRIDGE** (→ 页 195)

无诊断。
该 **输出端的自我保护** (→ 页 36)启用。

针对包含电流反馈的输出端的反应

20641

- 针对包含电流反馈的输出端：
  在应用程序中针对输出端请求一般电流！
  应用程序的程序员有责任对事件作出反应。

输出组 Q00...Q15

10445

这些输出端是一组多功能通道。

这些输出端提供多个功能选项（每个输出端可单独配置）：
- 二进制输出端，正极开关 (BH)，部分还有负极开关 (BL)
- 模拟电流控制输出端 (PWMi)
- 包含脉冲宽度调制 (PWM) 的模拟输出端（部分作为 H 桥）

→ 章节 **可能的输入端/输出端工作模式** (→ 页 289)

► 每个输出端均可通过应用程序进行配置：
  → 功能块 **SET_OUTPUT_MODE** (→ 页 191)> 输入端 MODE
  表示负载电流→ FB **OUTPUT_CURRENT** (→ 页 199)
  PWM 输出端： → FB **PWM1000** (→ 页 203)
  PWMi 输出端： → FB **OUTPUT_CURRENT_CONTROL** (→ 页 200)
  控制 H 桥→ FB **OUTPUT_BRIDGE** (→ 页 195)

► 控制输出端 Q00…Q03 和 Q08…Q11 的电流测量范围
  （2 A 或 4 A）：
  → 功能块 SET_OUTPUT_MODE > 输入端 CURRENT_RANGE

使用 H 桥时，电流控制不受支持。

注意

为保护内部测量电阻器，应始终启用过载保护（默认设定）。 提供以下保护：
- 电流测量范围 = 2 A： 保护起始点 2.25 A
- 电流测量范围 = 3 A： 保护起始点 3.375 A
- 电流测量范围 = 4 A： 保护起始点 4.5 A。

在 PWM 模式下，功能**不**受支持。
如有必要，可禁用功能。

限制值请务必确保遵循技术资料！

以下情况时断线和短路检测启用…
- 输出端配置为“二进制正极开关 (BH)”，艾尔
- 输出端开启。

► 使用输出开关聚集时，连接负载上的电源电压不得高于输出组电源电压！

► 如果要避免输出组在继电器禁用时继续运行，应通过针对相同输出组电源的输出开关为针对地面的输出开关提供负载。

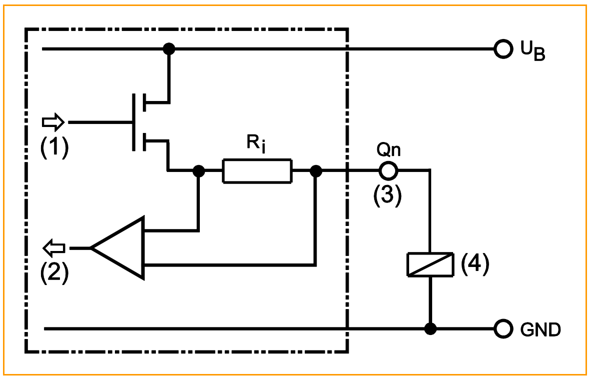
3976

视环境温度而定，不可能通过短路电流确切检测到短路，因为输出端驱动器暂时禁用，以防止受损。

诊断： 二进制输出端（通过电流测量）

19398
19396

这些输出端的诊断通过输出端的内部电流测量进行：



图： 原理方块图

(1) 输出通道

(2) 针对诊断的读回通道

(3) 插脚输出端 n

(4) 负载

诊断： 过载（通过电流测量）

19437
15249

仅可在包含电流测量的输出端检测到过载。

过载即…
“超过最大额定电流的 12.5 %”。

诊断： 断线（通过电流测量）

19400

断线检测通过输出端内部的读回通道完成。

诊断的前提：	输出端 = TRUE
诊断 = 断线：	电阻器 Ri 上无电流（无电压降落） 若未断线，则负载电流流经串联电阻器，形成电压降落，电压降落可通过读回通道评估。

诊断： 短路（通过电流测量）

19401

断线检测通过输出端内部的读回通道完成。

诊断的前提：	输出端 = TRUE
诊断 = 对 GND 短路	电源电压降落超过串联电阻器

3.2.7 关于配线的注意事项

1426

配线图（→ 装置安装说明，章节“配线”）讲述标准装置配置。 配线图帮助将输入端和输出端通道分配至 IEC 地址和装置端子。

单个缩写的含义如下：

A	模拟输入
BH	二进制高侧输入端： 负极性传感器信号负极开关 二进制高侧输出端 正极性输出端信号正极开关
BL	二进制低侧输入端： 正极性传感器信号正极开关 二进制低侧输出端： 负极性输出端信号负极开关
CYL	输入端周期测量
ENC	输出端编码器信号
FRQ	频率输入端
H 桥	包含 H 桥功能的输出端
PWM	**脉**冲**宽**度**调**制信号
PWMi	包含电流测量的 PWM 输出端
IH	脉冲/计数输入端，高侧： 负极性传感器信号负极开关
IL	脉冲/计数输入端，低侧： 正极性传感器信号正极开关
R	针对一个输出端的读回通道

输入端/输出端通道分配： → 产品目录、安装说明或技术资料

3.2.8 关于簧片继电器的安全说明

7348

若使用非电子开关，则应注意以下几点：

> ❗如果在没有串联继电器的情况下连接至装置输入端，则簧片继电器触点可能阻塞。

▶ **补救措施：** 安装针对簧片继电器的串联电阻器：
串联电阻器 = 簧片继电器的最大输入电压/允许电流
例如： 32 V / 500 mA = 64 Ohm

▶ 串联电阻器不得超过装置输入端输入电阻 RE 的 5 % （→ 技术资料）。 否则，信号将不会检测为 TRUE。
例如：
RE = 3 000 Ohm
⇒ 最大串联电阻器 = 150 Ohm

3.2.9 外部供给输出端反馈

2422

在有些应用程序中，执行器不仅由 PLC 输入端控制，还额外由外部开关控制。 在该等情况下，必须利用阻塞二极管保护外部供给的输出端（→ 请参阅下图）。

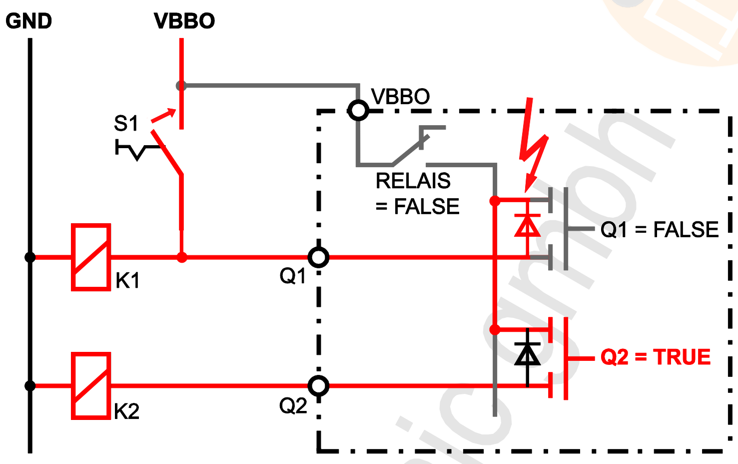
通知

如有不允许的反馈，则会损坏输出端！

如果执行器为外部控制，则同一输出组的电势条不可变为无电势（例如针对 RELAIS = FALSE）。

否则，端子电压 VBBx 则通过集成至外部连接输出端输出驱动器的防护二极管反馈至输入组电势条。因此，该组的其他固定输出端可能会触发其连接的负载。 负载电流会损坏提供反馈的输出端。

▶ 通过阻塞二极管保护外部供给的输出端！

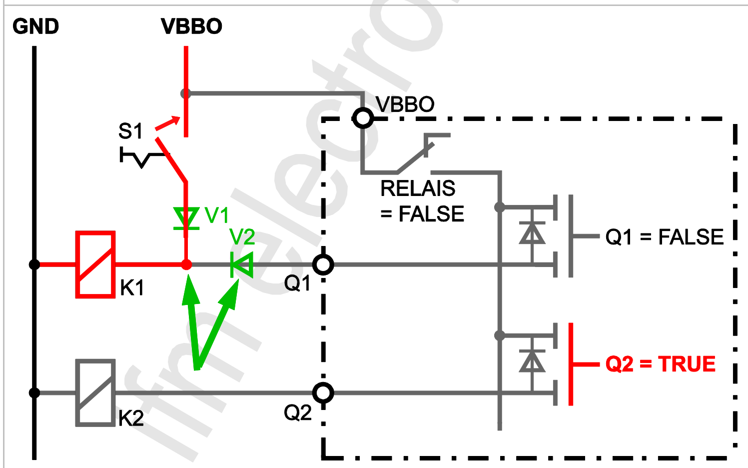


例如：

标志 RELAIS 关闭输出组的供给 VBBo。

若没有阻塞二极管，则外部开关 S1 通过内部防护二极管（红色）从输出端 Q1 向输出端内部电势条提供供给 VBBo。

如果输出端 Q2 = TRUE（→ 图），K2 将通过防护二极管 Q1 接收电压，尽管 RELAIS = FALSE（红线）。 该防护二极管会因为过载而烧坏，输出端 Q1 也会受损！



图： 鉴于反馈风险的阻塞二极管配线示例

补救措施：

插入阻塞二极管 V1 和 V2（→ 绿色箭头）！

成功：

如果 RELAIS = FALSE，K2 保持关闭，即便 Q2 = TRUE。

ⓘ注意

针对外部供给输出端的帮助

► 外部供给输出端必须通过二极管解耦，以便不会有外部电压施加至输入端端子。

3.2.10 状态 LED

20774

集成状态 LED 指示工作状态（默认设定）。

LED 颜色	显示屏	说明
熄灭	永久熄灭	无工作电压
	 t	
黄色	短暂亮起	初始化或复位检查
	 t	（时间帧 = 200 ms）
橙色	以 0.2 Hz 的频率闪烁	TEST=FALSE: 未加载运行时系统
	 t	（时间帧 = 1 s）
绿色	以 5 Hz 的频率闪烁	TEST=TRUE: 未加载运行时系统
	 t	（时间帧 = 200 ms）
绿色	以 2 Hz 的频率闪烁	应用程序 = RUN
	 t	（时间帧 = 200 ms）
绿色	永久亮起	应用程序 = STOP 或： 未加载应用程序
	 t	
红色	以 2 Hz 的频率闪烁	TEST=TRUE: 应用程序 = RUN 和 ERROR STOP / FATAL ERROR
	 t	（时间帧 = 200 ms）
红色	短暂亮起	陷阱错误或： TEST=FALSE: FATAL ERROR
	 t	（时间帧 = 200 ms）
红色	永久亮起	TEST=FALSE: ERROR STOP
	 t	

状态 LED 可通过工作状态 STOP 和 RUN 的编程系统更改。

在应用程序中控制 LED

20775

有了该装置，还可通过应用程序设定状态 LED。 为此，需使用以下系统变量（→ 章节 **系统标志** (→ 页 273)):

系统标志（符号名称）	类型	说明
LED	WORD	“LED 开启”的 LED 颜色 0x0000 = LED_GREEN（预设） 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_X	WORDWORD	“LED 关闭”的 LED 颜色 0x0000 = LED_GREEN 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK（预设） 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_MODE	WORD	LED 闪烁频率： 0x0000 = LED_2HZ（以 2 Hz 的频率闪烁； 预设） 0x0001 = LED_1HZ（以 1 Hz 的频率闪烁） 0x0002 = LED_05HZ（以 0.5 Hz 的频率闪烁） 0x0003 = LED_0HZ（按照 LED 中的值永久亮起）

注意

- ► 切勿在应用程序中使用 LED 颜色 RED。

> 如果出现错误，运行时系统设定 LED 颜色 RED。
但是： 如果在应用程序中更改颜色和/或闪烁模式，包含默认设定的上述表格则不再有效。

3.3 接口说明

内容

14098

3.3.1 串行接口

14099

本装置包含串行接口。

串行接口一般可搭配以下功能：

- 程序下载
- 调试
- 应用程序的自由使用

12998

注意

在默认情况下，串行接口对用户不可用，因为它用于程序下载和调试。

如果用户设定系统标志位 SERIAL_MODE=TRUE，则可自由使用接口。 而后仅可通过全部 4 个 CAN 接口中的一个调试应用程序。

连接和数据 → 技术资料

3.3.2 USB 接口

14100

该装置包含用于程序下载和调试的 USB 接口。

连接和数据 → 技术资料
在 PC 上安装 USB 驱动程序 → 安装说明/操作说明

通过 USB 在 CODESYS 中设定 [Online] > [Communication Parameters...] ：

装置	运行时系统版本	参数	值
CR0032	< V03.00.00	波特率	115200
CR0032	≥ V03.00.01	波特率	4800...57600
CR0033, CR0133	≤ V02.00.01	波特率	115200
CR0033, CR0133	≥ V02.00.02	波特率	4800...57600
CR0232, CR0233	全部	波特率	115200
CR0234, CR0235	全部	波特率	4800...57600
CR7n32	≤ V01.00.04	波特率	115200
CR7n32	≥ V01.00.05	波特率	4800...57600
CR0n3n, CR7n32	全部	摩托罗拉字节顺序	否
CR0n3n, CR7n32	全部	流控制	开启

3.3.3 CAN 接口

内容

14101

连接和数据 → 技术资料

CAN: 接口和协议

13820
14587

装置配有多个 CAN 接口，具体视硬件设计而定。 基本说来，所有接口均可搭配以下功能，彼此不受影响：

- 第 2 层： 第 2 级上的 CAN → 章节 **功能元件： CAN 第 2 层** (→ 页 91))
- CANopen 主站（→ 章节 **功能元件： CANopen 主站** (→ 页 102))
- CANopen 从站（→ 章节 **功能元件： CANopen 从站** (→ 页 114))
- CANopen 网络变量（通过 CODESYS）
- SAE J1939（针对驱动管理，→ 章节 **功能元件： SAE J1939** (→ 页 129))
- 总线负载检测
- 错误帧计数
- 下载接口
- 100 % 总线负载，无数据包丢失

11793

以下 CAN 接口和 CAN 协议可用于本 ecomat*mobile* 装置：

CAN 接口	CAN 1	CAN 2	CAN 3	CAN 4
默认下载 ID	ID 127	ID 126	ID 125	ID 124
CAN 协议	CAN 第 2 层	CAN 第 2 层	CAN 第 2 层	CAN 第 2 层
	CANopen	CANopen	CANopen	CANopen
	SAE J1939	SAE J1939	SAE J1939	SAE J1939

标准波特率 = 125 Kbits/s

ℹ 哪个 CANopen 兼容接口搭配哪个 CANopen 协议，取决于您在 PLC 配置中附加子元件的顺序。
CODESYS > [PLC Configuration] > [CR0033 Configuration Vxx] > [Append subelement] > [CANopen master] 或 [CANopen slave]

3.4 软件说明

内容

14107

3.4.1 装置的软件模块

内容

14110

本装置中的软件与以下硬件进行通信：

软件模块	用户可更改模块吗？	通过什么工具？
应用程序 包含库	是	CODESYS, 维护工具
运行时系统 *)	升级 是 降级 是	维护工具
Bootloader	否	---
（硬件）	否	---

*) 运行时系统版本号必须与 CODESYS 目标系统设定中的目标版本号一致。
→ 章节 **设定目标** (→ 页 68)

我们将在下文将是该软件模块：

Bootloader

14111

交付时 **ecomatmobile** 控制器仅包含 Bootloader。

Bootloader 是有助于将运行时系统和应用程序再次加载至装置的启动程序。

Bootloader 包含基本例程...

- 针对硬件模块之间的通信，
- 针对操作系统的重新加载。

Bootloader 是要保存在装置上的第一个软件模块。

运行时系统

14112

装置中的基本程序，建立装置硬件和应用程序之间的连接。

交付时，控制器中一般未加载运行时系统（LED 以 5 Hz 的频率闪烁绿色）。 在该工作模式中，仅 Bootloader 启用。 它提供加载运行时系统的最低限度功能，以及其他接口（如 CAN）支持。

正常而言，只需下载一次运行时系统。 而后应用程序可加载至控制器（也可多次），不会影响运行时系统。

运行时系统随附本文档，位于单独的数据载体。 此外，可从 **ifm electronic gmbh** 网站下载当前版本：

→ www.ifm.com

应用程序

14118

针对应用程序的软件，由机器制造商实施，一般包含控制相应输入端、输出端计算和决策的逻辑序列、限制和表达式。

8340

⚠ 警告

用户应对其设计创建的应用程序的可靠功能负责。 如有必要，必须请相应的监管和测试机构，按照国家法规额外执行批准测试。

库

14117

ifm electronic 提供多个库 (*.LIB) 以匹配各个包含针对应用程序的程序模块的装置。示例：

库	使用
ifm_CR0033_Vxxyyzz.LIB	装置特定库 必须始终包含于应用程序！
ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB x = 1...4 = CAN 接口编号	（可选） 如果装置 CAN 接口作为 CANopen 主站运行
ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB x = 1...4 = CAN 接口编号	（可选） 如果装置 CAN 接口作为 CANopen 从站运行
ifm_CR0033_J1939_Vxxyyzz.LIB	（可选） 如果装置 CAN 接口是为了与柴油发动机通信

Details: → chapter **针对装置 CR0033 的 IFM 库** (→ 页 85)

3.4.2 CODESYS 项目的编程说明

内容

7426

您可在此了解如何对装置进行编程。

► 请参阅 CODESYS 编程手册中的说明
→ www.ifm.com
→ ecomatmobile DVD“软件、工具和文档”.

CODESYS 中的 FB、FUN、PRG

8473

在 CODESYS 中，我们区分以下类型的功能元件：

FB = 功能块

• FB 可拥有多个输入端和多个输出端。
• 在一个项目中，可多次调用 FB。
• 每次调用必须声明实例。
• 允许： 调用 FB 中的 FB 和 FUN。

FUN = 功能

• 功能可拥有多个输入端，但仅有一个输出端。
• 输出端跟功能本身拥有相同的数据类型。

PRG = 程序

• PRG 可拥有多个输入端和多个输出端。
• 在一个项目中，仅可调用一次 PRG。
• 允许： 调用 PRG 中的 PRG、FB 和 FUN。

注意

不得在功能中调用功能块！
否则： 执行期间，应用程序将崩溃。

不得递归或间接调用所有功能元件！

IEC 应用程序最多可包含 8,000 个功能元件！

背景

功能的所有变量…
- 调用时初始化，且
- 返回至调用方后无效。

功能块有 2 种调用：
- 初始化调用以及
- 操作实际调用。

因此，这意味着功能中的 FB 调用。
- 每次有一次额外的初始化调用，且
- 上次调用的数据丢失。

应用程序中的计算和转换

20779

注意

如果应用程序中需要以下元件：
- 数学功能（如 ATAN），
- 计算，
- 转换（如 REAL_TO_BYTE），

则以下适用于相应运算符输入端和输出端的值：

► 严格遵守每种情况下的允许值范围！

> 否则控制器可能会出现 FPU 错误。

例如：

20777

目标格式的最大表示值会被超越。

例如：

REAL_TO_INT (12345678.3)

> INT 限于 -32768...+32767（仅整数）

20778

现有实际值显然在目标格式的值范围内。
但实际上，数字不在目标格式范围内（因为实际值的内部表示）。
例如：
DW := REAL_TO_DWORD (4294967295.0);
> 在 REAL 中，4294967295 的最准确表示是 4.294967296E9
> 因此，值在目标格式最大允许值的基础上 + 1。
> DWORD 限于 0...4294967295.

注意周期时间！

8006

就 **ecomat*mobile*** 系列控制器的可编程装置而言，有很多功能有助于在一系列应用中使用装置。
因为这些元件根据其复杂性使用较多或较少的系统资源，所以并非总是能够同时并多次使用所有元件。

通知

装置有运行过慢的风险！
周期时间不得太长！

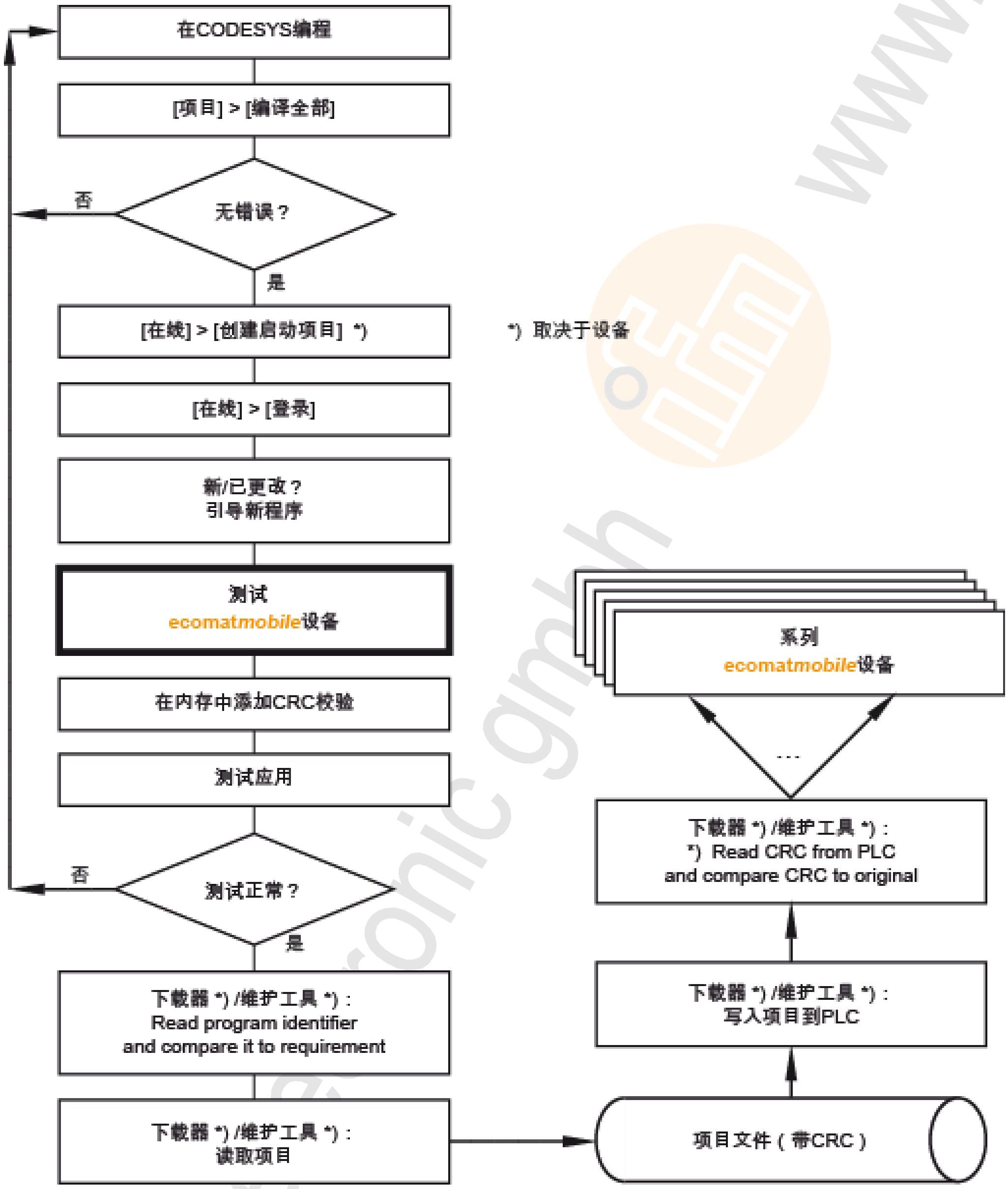
- ► 设计应用程序时必须遵循上述建议并加以测试。
- ► 如有必要，必须通过软件重组和系统设定来优化周期时间。

创建应用程序

8007

应用程序由 CODESYS 编程系统生成，并在程序开发期间多次加载至控制器，以便进行测试：
在 CODESYS 中： [Online] > [Login] > 加载新的程序。

就每一次经由 CODESYS 的下载而言，会再次转化源代码。 结果是，每次控制器内存将形成新的校验和。 该进程还适用于安全控制器，直至发布软件。



图： 创建和分发软件

保存启动项目

7430

始终将相关启动项目和装置中的应用程序项目保存在一起。 只有这样，应用程序才可在装置电源故障后可用。

> **注意**
>
> 注意： 启动项目稍大于实际程序。
>
> 但是： 如果启动项目大于可用的 IEC 代码内存范围，则无法将启动项目保存在装置中。 通电后，启动项目会被删除或无效。

► CODESYS 菜单 [Online] > [Create boot project]
每次更改后，该操作是必要的。

> 重启后，装置将以最后保存的启动项目启动。

> 如果未保存启动项目：

- 重启后装置保持 STOP 运行状态。
- 应用程序不（再）可用。
- LED 呈绿色亮起。

使用 IFM 下载程序

8008

IFM 下载程序用于将程序代码从编程站轻松传输至控制器。 原则上来说，每个应用程序软件均可利用 IFM 下载程序复制到控制器。 优点： 无需包含 CODESYS 许可证的编程系统。

您可在此了解当前的 IFM 下载程序（最低 V06.18.26）：

ecomatmobile DVD“软件、工具和文档”在选项卡 'R360 tools [D/E]' 下方

使用 IFM 维护工具

8492

IFM 维护工具用于将程序代码从编程站轻松传输至控制器。 原则上来说，每个应用程序软件均可利用 IFM 维护工具复制到控制器。 优点： 无需包含 CODESYS 许可证的编程系统。

您可在此了解当前的 IFM 维护工具：

→ www.ifm.com > [移动车辆和工程机械系统]

→ ecomatmobile DVD“软件、工具和文档”在选项卡 'R360 tools [D/E]' 下方

3.4.3 工作状态

内容

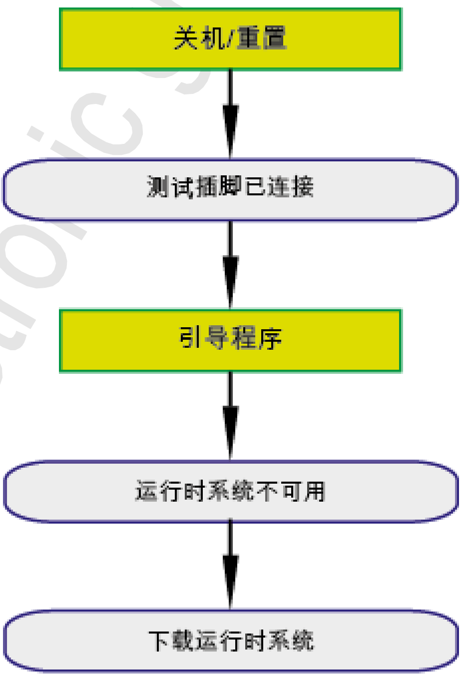
14120

通电后，**ecomat*mobile*** 装置可处于五种可能的工作状态之一：

- BOOTLOADER
- INIT
- STOP
- RUN
- SYSTEM STOP （ERROR STOP 之后）

工作状态： 运行时系统不可用

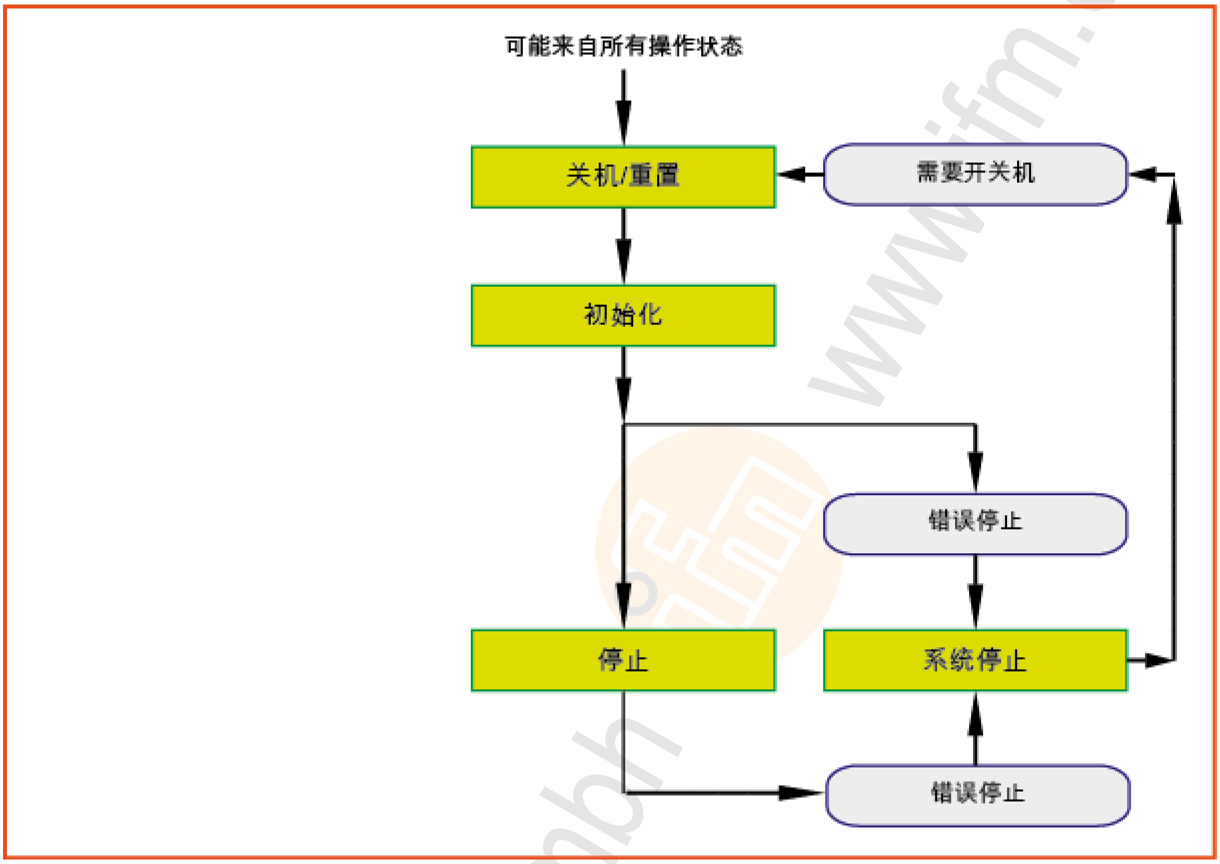
19217



图： 工作状态（此时： 运行时系统不可用）

工作状态： 应用程序不可用

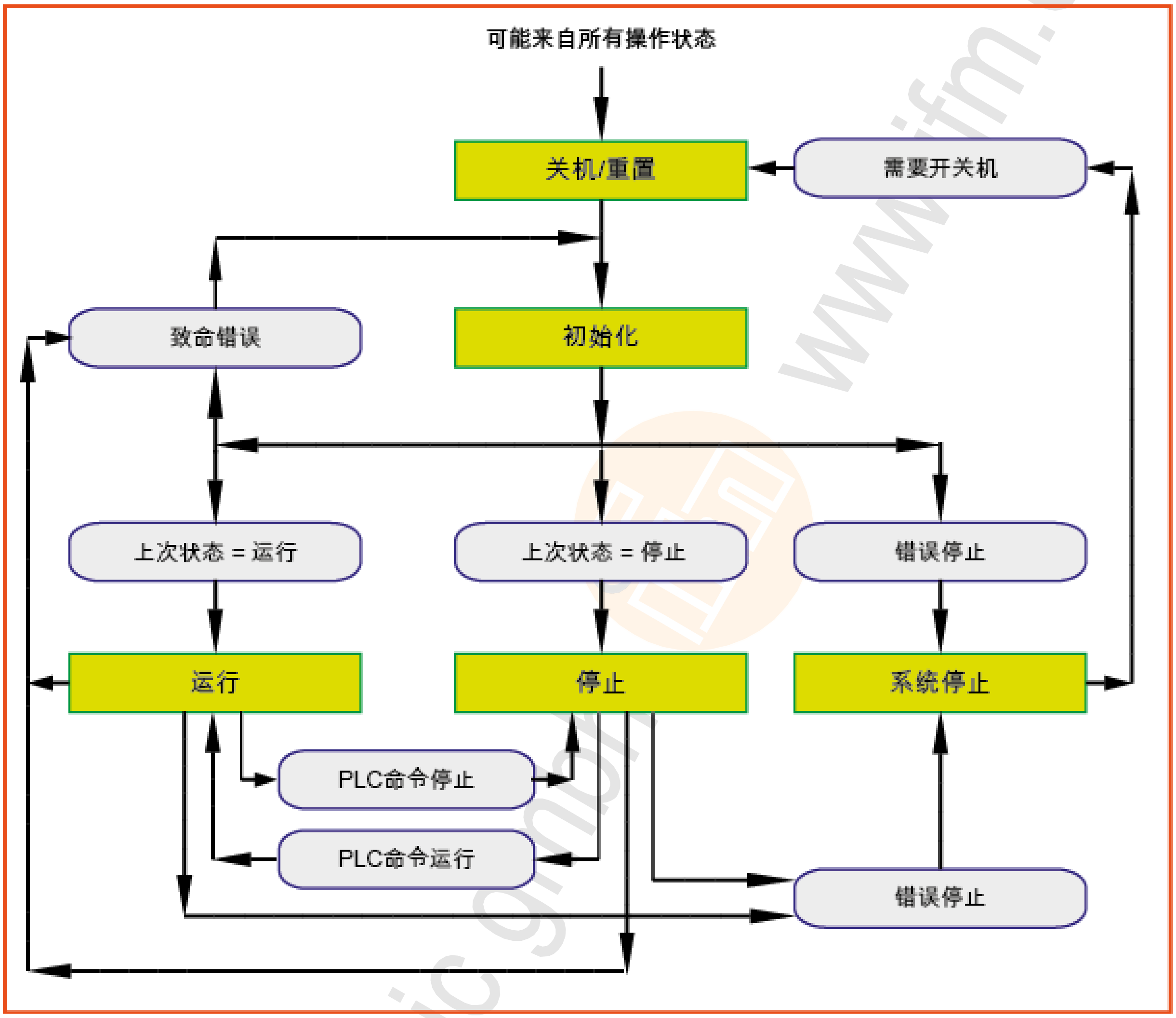
19218



图： 工作状态（此时： 应用程序不可用）

工作状态： 应用程序可用

19219



图： 工作状态（此时： 应用程序可用）

Bootloader 状态

1080

未加载运行时系统。 ecomat*mobile* 控制器处于启动加载状态。 在加载应用程序软件之前，必须先下载运行时系统。

> LED 闪烁绿色 (5 Hz)。

INIT 状态（复位）

1076

前提： 安装有效的运行时系统。

每次通电复位后均会经历该状态。

> 运行时系统初始化。

> 展开各种检查，如等待正确的电源电压。

> RUN 或 STOP 状态代替该温度状态。

> LED 呈黄色亮起。

该状态可转换至以下状态之一：
• RUN
• STOP

STOP 状态

1078

在以下情况下可达到该状态：

- 从 RESET 状态转换，前提是：
  • 未加载程序，或
  • RESET 之前的最后状态为 STOP 状态
- 通过 STOP 命令从 RUN 状态转换
  • 仅针对工作模式 = TEST（→ 章节 **TEST 模式** (→ 页 63))

> LED 呈绿色亮起。

RUN 状态

1077

在以下情况下可达到该状态：

- 从 RESET 状态转换，前提是：
  • RESET 之前的最后状态为 RUN 状态
- 通过 RUN 命令从 STOP 状态转换
  • 仅针对工作模式 = TEST（→ 章节 **TEST 模式** (→ 页 63))

> LED 闪烁绿色 (2 Hz)。

SYSTEM STOP 状态

19222

如果发现不可容忍的错误 (ERROR STOP)，**ecomat*mobile*** 控制器则切换至该状态。 若要改变该状态，则仅可进行断电通电复位操作。

> LED 呈红色亮起。

3.4.4 工作模式

1083

ecomat*mobile* 控制器可在不同模式下运行，不受工作状态影响。

TEST 模式

1084

通知

存储的软件可能丢失！

在测试模式中，存储的运行时系统和应用程序软件没有保护。

14892

注意

> 仅在您连接 OPC 客户端后，将 TEST 接头连接至电源电压！

将电源电压施加至测试输入端即可进入该工作模式
（→ 安装说明 > 章节“技术资料” > 章节“配线”）。

ecomat*mobile* 现在可在 RUN 或 STOP 模式下通过其中一个接口接收命令，如与编程系统进行通信。

软件仅可在 TEST 模式中下载至控制器。

可通过标志 TEST 查询应用程序的状态。

摘要 测试输入端已启用：

- 编程模式已启用
- 可下载软件
- 可查询应用程序的状态
- 无法保护存储的软件

注释： TEST 输入端

20781

- ▶ 机器中所有控制器的 TEST 输入端应单独配线并清晰标注，以便能够正确配置至控制器。
- ▶ 服务访问期间，仅启用待访问控制器的 TEST 输入端。

SERIAL_MODE

2548

串行接口可用于应用程序中的数据交换。 而后仅可通过全部 4 个 CAN 接口调试应用程序软件。

该功能关闭以作为标准 (FALSE)。 状态可通过标志 SERIAL_MODE 控制并通过应用程序或编程系统查询。

→ 章节 **功能元件： 串行接口** (→ 页 144)

DEBUG 模式

1086

如果输入端 DEBUG ***SET_DEBUG*** (→ 页 257)设为 TRUE，编程系统或下载程序等则可与控制器通信并执行某些特别的系统命令（如针对服务功能，通过 GSM 调制解调器 CANremote）。

在该工作模式中，不可进行软件下载，因为测试输入端（→ 章节“ **TEST 模式** (→ 页 63)”）未连接至电源电压。

3.4.5 装置的性能极限

7358

注意装置的性能极限！ → 技术资料

电子狗状况

11786

在本装置中，电子狗监控 CODESYS 应用程序的程序运行时。

如果超过最长电子狗时间（大约 100 ms）：

> 装置复位并重启。

这样您可读取标志 LAST_RESET。

CODESYS 功能

2254

您应注意以下限制：

- 最多可支持 2 048 个 块（PB、FB...）。
- 适用于用户的标志 → 章节 **可用内存** (→ 页 19).
  保留标志说明 → 针对相应 FB。

4 配置

内容

18065
1016

相应安装说明或本文档 **附录** (→ 页 273)所述的装置配置用于标准装置（库存产品）。 它们满足大多数应用程序要求的规格。

但视客户的系列使用要求而定，还可使用其他装置配置，如与输入端/输出端和模拟通道相关的配置。

16420

注意

这些说明适用于不含和包含集成 I/O 模块的装置。

► 在两种情况下，均确保针对装置 CR0033 设定 PLC 配置！

您可在以下位置找到更多关于集成 I/O 模块的信息：

→ 章节 **集成 I/O 模块： 说明** (→ 页 294)，本文档附录。

4.1 设定运行时系统

内容

14091

4.1.1 重新安装运行时系统

14092
2733

交付 **ecomat*mobile*** 装置时，一般未加载运行时系统（LED 以 5 Hz 的频率闪烁绿色）。 在该工作模式中，仅 Bootloader 启用。 它提供加载运行时系统（如 RS232、CAN）的最低限度功能。

正常而言，只需下载一次运行时系统。 应用程序可加载至控制器（也可多次），不会影响运行时系统。

运行时系统随附本文档，位于单独的数据载体。 此外，可从 **ifm electronic gmbh** 网站下载当前版本：

→ www.ifm.com

2689

注意

必须始终使用适于所选目标的软件版本：

- 运行时系统 (`ifm_CR0033_Vxxyyzz.H86`),
- PLC 配置 (`ifm_CR0033_Vxx.CFG`)，
- 装置库 (`ifm_CR0033_Vxxyyzz.LIB`) 和
- 进一步的文件。

V	版本
xx: 00...99	目标版本号
yy: 00...99	版本号
zz: 00...99	修补号

基本文件名称（例如 "CR0033"）和软件版本号 "xx"（例如 "02"）必须始终是相同的值！ 否则装置会进入“停止”模式。

"yy"（版本号）和 "zz"（修补号）的值**不必**一致。

4368

还必须加载以下文件：

- 项目所需的内部库（在 IEC 1131 中创建），
- 配置文件 (`*.CFG`) 以及
- 目标文件 (`*.TRG`)。

鉴于您目前安装的 CODESYS 版本，目标系统可能无法编程或只能部分编程。 在此情况下，请联系 **IFM** 技术支持部门。

运行时系统通过单独的程序“**IFM** 下载程序”传输至装置。 （下载程序位于 **ecomat*mobile*** DVD“软件、工具和文档”或者，如有必要，可从 **IFM** 网站下载： → www.ifm.com.

正常情况下，应用程序通过编程系统加载至状态。 但如果首次从装置读取，还可利用 **IFM** 加载程序加载（→ 上传）。

4.1.2 更新运行时系统

13269

装置上已安装较旧版本的运行时系统、。 现在，您是否想要更新装置上的运行时系统？

14158

通知

数据有丢失的风险！

删除或升级运行时系统时，装置上的所有数据和程序将被删除。

▶ 先保存所有所需数据和程序，再删除或升级运行时系统！

就该操作而言，可遵循跟之前章节“重新安装运行时系统”相同的说明。

4.1.3 检验安装

19517
14406

▶ 将运行时系统加载至控制器之后：
- 检查运行时系统是否正确传送！
- 检查控制器是否加载适当的运行时系统！

▶ 第 1 次检查：
利用 IFM 下载程序或维护工具核实是否加载正确的运行时系统版本：
- 读取装置中运行时系统的名称、版本和 CRC！
- 手动对比该信息和目标数据！

▶ 第 2 次检查（可选）：
在应用程序中核实是否加载适当的运行时系统版本：
- 读取装置中运行时系统的名称和版本！
- 对比该数据和特定值！

以下 FB 用于读取数据：

GET_IDENTITY (→ 页 255)	读取存储在装置中的特定标识： • 装置的硬件名称和硬件版本 • 装置的序列号 • 装置运行时系统的名称 • 装置运行时系统的版本和版本号 • 应用程序的名称（之前已通过以下方式保存： SET_IDENTITY (→ 页 258))

▶ 如果应用程序检测到运行时系统的版本不正确：
将所有安全功能调至安全状态。

4.2 设定编程系统

内容

3968

4.2.1 手动设定编程系统

内容

3963

设定目标

2687
11379

在 CODESYS 中创建新项目时，必须加载与装置相符的目标文件。

► 在菜单 [Configuration] 对话窗口 [Target Settings] 中选择所需目标文件。

> 目标文件构成编程系统硬件的接口。

> 同时，选择目标时加载多个重要的库和 PLC 配置。

► 如有必要，进入窗口 [Target settings] > 选项卡 [Network functionality] > 启用 [Support parameter manager] 并/或启用 [Support network variables]。

► 如有必要，移除加载的 (3S) 库或通过进一步的 (IFM) 库加以补充。

► 始终补充相应的装置库 `ifm_CR0033_Vxxyyzz.LIB` 并手动操作！

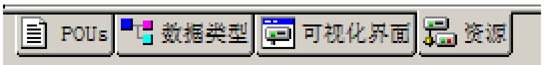
启用 PLC 配置（如 CR0033

15824

在配置编程系统期间（→ 上一部分），PLC 配置自动执行。

► 通过选项卡 [Resources] 可前往菜单项 [PLC Configuration]。
双击 [PLC Configuration] 以打开相应的窗口。

► 单击 CODESYS 中的选项卡 [Resources]：

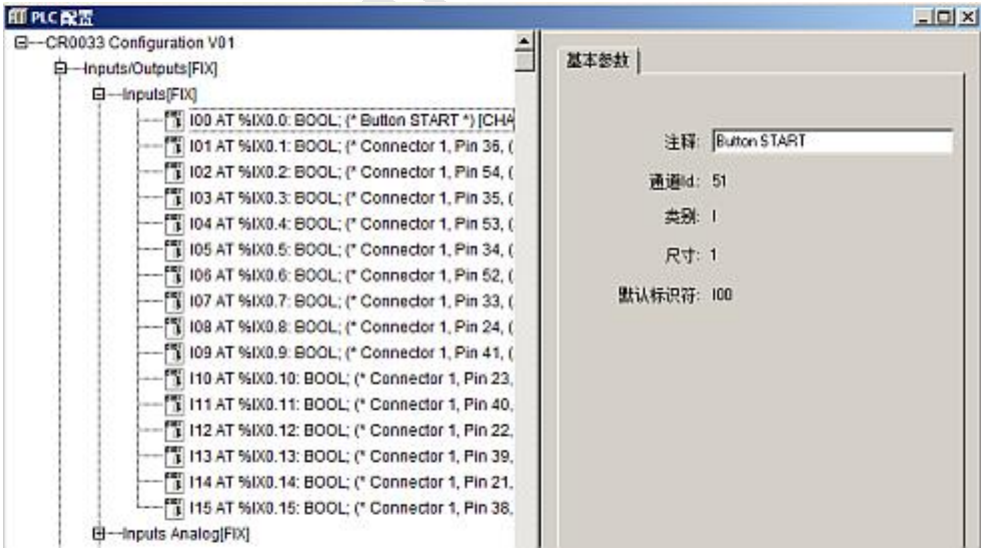


► 双击左列的 [PLC Configuration]。

> 当前 PLC 配置显示屏（→ 下图）：

用户可根据配置在程序环境中找到以下内容：

- 所有重要系统和错误标志
  根据应用程序和程序，这些标志必须经过处理和评估。 可通过符号名称访问。
- 输入端和输出端的结构
  这些可在窗口 [PLC Configuration]（→ 下图）中通过符号直接指定（强烈建议！），且可作为 [Global Variables] 适用于整个项目。



4.2.2 通过模板设定编程系统

13745

IFM 提供即用型模板（程序模板），利用该模板可轻松、快速和充分地设定编程系统。

970

安装 **ecomat*mobile*** DVD“软件、工具和文档”时，包含模板的项目已存储于您的 PC 的程序目录：

`…\ifm electronic\CoDeSys V…\Projects\Template_DVD_V…`

► 通过以下路径在 CODESYS 中打开所需模板：
[File] > [New from template…]

> CODESYS 创建新项目并显示基本程序结构。 强烈建议遵循所示的步骤。

4.3 一般功能配置

3971

4.3.1 配置输入端和输出端（默认设定）

20784

输入端（预设）

20785

- 处于二进制模式（正极开关）
- 未启用诊断功能

包含电流测量的输出端（默认设定）

20786

- 处于二进制模式（正极开关）
- 启用诊断功能
- 启用过载保护

不含电流测量的输出端（默认设定）

20787

- 处于二进制模式（正极开关）
- 未启用诊断功能
- 无过载保护

4.3.2 系统变量

13519
15576

所有系统变量（→ 章节 **系统标志** (→ 页 273)）均已定义地址且不可更改。

> 若要表示和处理电子狗错误或重新启动系统的原因，则设定系统变量 LAST_RESET。

4.4 输入端和输出端功能配置

内容

1394

就 **ecomat*mobile*** 控制器系列的某些装置而言，可启用针对输入端和输出端的额外诊断功能。 因此，可监控相应的输入端和输出端信号，出现故障时，应用程序可作出响应。

使用诊断功能时需考虑特定的边界条件，具体视输入端和输出端而定：

► 必须通过技术资料检查使用的装置是否有所述的输入组和输出组（→ 数据资料）。

- 针对输入端和输出端配置的装置库 (`ifm_CR0033_Vxxyyzz.LIB`) 已预定义常量（如 IN_DIGITAL_H）。
  详细信息 → **可能的输入端/输出端工作模式** (→ 页 289).

4.4.1 配置输入端

内容

3973

有效的工作模式 → 章节 **可能的输入端/输出端工作模式** (→ 页 289)

关于簧片继电器的安全说明

7348

若使用非电子开关，则应注意以下几点：

如果在没有串联继电器的情况下连接至装置输入端，则簧片继电器触点可能阻塞。

- **补救措施：** 安装针对簧片继电器的串联电阻器：
  串联电阻器 = 簧片继电器的最大输入电压/允许电流
  例如： 32 V / 500 mA = 64 Ohm
- 串联电阻器不得超过装置输入端输入电阻 RE 的 5 % （→ 技术资料）。 否则，信号将不会检测为 TRUE。
  例如：
  RE = 3 000 Ohm
  ⇒ 最大串联电阻器 = 150 Ohm

配置输入端软件过滤器

6883

可通过系统变量 Ixx_FILTER 配置过滤模拟输入端测量输入电压的软件过滤器。 如果是阶跃响应，过滤器则像传统低通滤波器一样运行，极限频率通过系统变量中输入的值设定。 可选择值 0...8。

表格： 模拟输入端软件低通滤波器极限频率

Ixx_FILTER	过滤频率 [Hz]	信号上升时间	备注
0	过滤禁用		
1	390	1 ms	
2	145	2.5 ms	
3	68	5 ms	
4	34	10 ms	建议，默认设定
5	17	21 ms	
6	8	42 ms	
7	4	84 ms	
8	2	169 ms	
≥ 9	34	10 ms	→ 默认设定

12969

更改过滤器设定后，该输入端或输出端的值不会立刻正确输出。 值仅会在信号上升时间（→ 表格）结束之后再次修正。

信号上升时间即采用输入步骤时过滤器输出端信号从终值的 10 % 上升到 90 % 所需的时间。 信号下降时间即信号从 90 % 降至 10 % 所需的时间。

模拟输入端： 配置和诊断

13960

► 每个输入端均可通过应用程序进行配置：
  • FB **INPUT_ANALOG** (→ 页 157) > 输入端 MODE 或：
  • FB **SET_INPUT_MODE** (→ 页 161)> 输入端 MODE

> 如果针对电流测量配置模拟输入端，则装置切换至安全电压测量范围 (0...32V DC)，超过终值时 (> 21.7 mA)，在标志字节 ERROR_CURRENT_Ix 中设定相应的错误位。
如果电流值再次低于限值，则装置每秒钟检查一次。 当值再次低于限值时，输入端会自动切回至电流测量范围。

> 在应用程序中，系统变量 ANALOG00...ANALOGxx 可用于客户特定诊断。

18414

如果未使用输入端 I15：

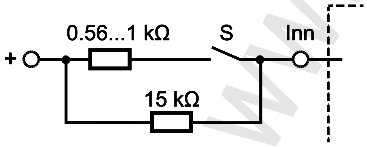
► 将输入端 I15 配置为二进制输入端！

二进制输入端： 配置和诊断

14516

▶ 每个输入端均可通过应用程序进行配置：
- FB **INPUT_ANALOG** (→ 页 157) > 输入端 MODE 或：
- FB **SET_INPUT_MODE** (→ 页 161)> 输入端 MODE

▶ 针对 NAMUR： 如果要使用诊断功能，则额外启用该模式：
- 功能块 SET_INPUT_MODE > 设定输入端 DIAGNOSTICS。

非电子开关二进制信号 NAMUR 诊断： ▶ 为开关配置额外的电阻器连接件！	 图： 输入端 Inn 非电子开关 S

13956

> 诊断结果可通过以下系统标志显示：

系统标志（符号名称）	类型	说明
ERROR_BREAK_Ix （0...x，值视装置而定→ 技术 资料）	DWORD	输入双字 x： 断线错误 或（电阻输入）： • 电源短路 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE： 错误 位 = FALSE： 无错误
ERROR_SHORT_Ix （0...x，值视装置而定→ 技术 资料）	DWORD	输入双字 x： 短路错误 仅当输入模式 = IN_DIGITAL_H 时 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE： 错误 位 = FALSE： 无错误

> 在应用程序中，系统变量 ANALOG00...ANALOGxx 可用于客户特定诊断。

快速输入端

19318

装置处理输入频率高达 30 kHz 的快速计算/脉冲输入端（→ 数据表）。

快速输入端的输入电阻根据应用的模式或功能块自动切换：

输入电阻	针对模式/FB
3.2 kohms	（标准）FAST_COUNT, FREQUENCY, INC_ENCODER, PERIOD 和类似 FB
50.7 kohms	包含 32 V 固定开关值的输入端

例如，如果机械开关连接至这些输入端，则控制器中可能因为触点弹跳而出现故障信号。

▶ 如有必要，利用过滤器 Ixx_DFILTER 过滤这些"故障信号"。
（→ 章节 **系统标志** (→ 页 273)）（并非适用于所有输入端）

适当的功能块包括：

FAST_COUNT (→ 页 171)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 173)	测量到达所选通道的信号的频率。
FREQUENCY_PERIOD (→ 页 175)	测量所示通道频率和以 [µs] 为单位的周期（周期时间）
INC_ENCODER (→ 页 178)	针对编码器评估的递增/递减计数器功能
INC_ENCODER_HR (→ 页 181)	针对编码器高分辨率评估的递增/递减计数器功能
PERIOD (→ 页 183)	测量所示通道频率和以 [µs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 185)	测量所示周期内所示通道频率和以 [µs] 为单位的周期（周期时间）。 此外，传号空号比以 [‰] 为单位显示。
PHASE (→ 页 187)	读取包含快速输入端的通道对并对比信号的相位

使用这些单元时，参数化输入端和输出端自动配置，因此程序员无需进行此操作。

配置硬件过滤器

19320

可通过系统变量 Ixx_DFILTER 在快速计数器和脉冲输入端上配置数字硬件过滤器。 以 µs 为单位（最大 100 000）的值表示二进制电平在被采用之前需无中断应用的时间。 默认 = 0 µs。

> ❗输入信号的电平更改按照过滤器中设定的值延迟。

过滤器仅影响针对以下功能块的已检测信号：

FAST_COUNT (→ 页 171)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 173)	测量到达所选通道的信号的频率。
FREQUENCY_PERIOD (→ 页 175)	测量所示通道频率和以 [µs] 为单位的周期（周期时间）
INC_ENCODER (→ 页 178)	针对编码器评估的递增/递减计数器功能
INC_ENCODER_HR (→ 页 181)	针对编码器高分辨率评估的递增/递减计数器功能
PERIOD (→ 页 183)	测量所示通道频率和以 [µs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 185)	测量所示周期内所示通道频率和以 [µs] 为单位的周期（周期时间）。 此外，传号空号比以 [‰] 为单位显示。

数字过滤器不适用于所有快速计数器和脉冲输入端。

作为二进制输入端使用

3804

允许的高输入频率还可确保检测到故障信号，例如机械开关弹跳的触点。

► 如有必要，在应用程序中解决故障信号！

4.4.2 配置输出端

内容

3976

有效的工作模式 → 章节 **可能的输入端/输出端工作模式** (→ 页 289)

配置输出端软件过滤器

6882

可通过系统变量 Qxx_FILTER 配置过滤测量电流值的软件过滤器。

- 如果是阶跃响应，过滤器则像传统低通滤波器一样运行，极限频率通过系统变量中输入的值设定。
- 电流测量期间，过滤设定会影响诊断时间。

表格： 针对输出端电流测量的软件低通过滤器极限频率

Qxx_FILTER	过滤频率 [Hz]	信号上升时间	备注
0	过滤禁用		
1	580	0.6 ms	
2	220	1.6 ms	
3	102	3.5 ms	
4	51	7 ms	建议，默认设定
5	25	14 ms	
6	12	28 ms	
7	6	56 ms	
8	3	112 ms	
≥ 9	51	7 ms	→ 默认设定

12969

❗更改过滤器设定后，该输入端或输出端的值不会立刻正确输出。 值仅会在信号上升时间（→ 表格）结束之后再次修正。

ℹ信号上升时间即采用输入步骤时过滤器输出端信号从终值的 10 % 上升到 90 % 所需的时间。 信号下降时间即信号从 90 % 降至 10 % 所需的时间。

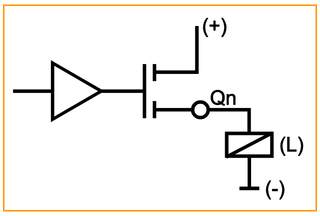
二进制输出端： 配置和诊断

15754

以下工作模式适用于装置输出端（(→ 技术资料）：

- 二进制输出端，正极开关 (BH)，含/不含诊断功能
- 二进制输出端，负极开关 (BL)，不含诊断功能

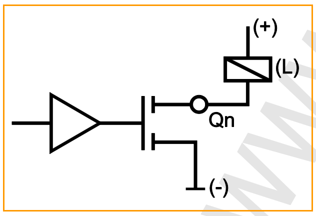
15450



Qn = 插脚输出端 n

(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号



Qn = 插脚输出端 n

(L) = 负载

输出端负极开关 (BL) 基本电路
针对负极性输出端信号

13975

⚠ 警告

可能存在危险的重启！
存在人身伤害的风险！ 存在机器/设备材料损坏的风险！

如果在故障情况下输出端通过硬件关闭，应用程序生成的逻辑状态不会改变。

► 补救措施：
- 复位应用程序中的输出逻辑！
- 排除故障！
- 根据状况复位输出端。

二进制输出端： 配置

15868

► 每个输出端均可通过应用程序进行配置：
→ 功能块 **SET_OUTPUT_MODE** (→ 页 191)> 输入端 MODE
允许值 → 章节 **可能的输入端/输出端工作模式** (→ 页 289)

二进制输出端： 诊断

15762

如果要使用诊断，则需额外启用。

前提： 将二进制输出端配置为正极开关

► 将输出端作为包含诊断的二进制输出端（→ 数据表）：
→ FB SET_OUTPUT_MODE > 输入端 DIAGNOSTICS = TRUE

> 系统标志 ERRORCODE 显示诊断消息：

ERRORCODE	DWORD	写入内部错误列表的最后一个错误 列表包含出现的所有错误代码。

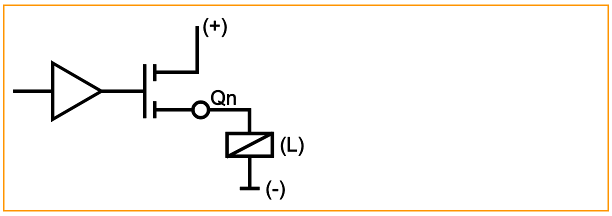
更多消息通过各种错误标志显示。
→ 章节 **系统标志** (→ 页 273)

PWM 输出端

14717

以下工作模式适用于装置输出端（(→ 技术资料）：
• PWM 输出端，正极开关 (BH)，不含诊断功能
• PWM 输出端对 H 桥，不含诊断功能

15451



Qn = 插脚输出端 n
(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号

16253

⚠ 警告

可能因故障导致财产损失或人身伤害！

针对 PWM 模式下的输出端：
• 无诊断功能

9980

注意

PWM 输出端不可并联运行，以便增加最大输出电流等。 输出端不同步运行。
否则整体负载电流可能仅流经一个输出端。 电流测量将不再正常进行。

- PWM 可在包含和不含电流控制功能的情况下运行。
  电流控制 PWM 输出端主要用于触发比例液压功能。
  PWM 信号中的中度电流仅可在开启状态下的电流处于测量范围内时通过 FB OUTPUT_CURRENT 正确确定。

PWM 可用性

15885

包含 PWM 功能的输出端 → 数据表

针对 PWM 功能的 FB

14710

以下功能块可用于 PWM 功能：

OUTPUT_BRIDGE (→ 页 195)	PWM 通道对上的 H 桥
OUTPUT_CURRENT (→ 页 199)	测量输出通道上的电流（平均，通过抖动周期）
OUTPUT_CURRENT_CONTROL (→ 页 200)	PWMi 输出通道的电流控制器
PWM1000 (→ 页 203)	初始化并配置具备 PWM 功能的输出通道 传号空号比可按 1 ‰ 的步距表示

输出端的可能特殊功能：

→ 章节 **功能块： 液压控制** (→ 页 205)
→ 章节 **功能块： 控制器** (→ 页 223)

通过 PWM 控制电流 (= PWMi)

13829

可通过控制器中集成的电流测量通道对线圈电流展开测量。 这样的话，如果线圈升温，则可重新调整电流。 因此，系统中的液压关系保持不变。

原则上，电流控制输出端有短路保护。

4.5 变量

内容

3130

在本章，您将了解更多关于如何处理变量的信息。

4.5.1 保留变量

15454

声明为 `RETAIN` 的变量生成剩余数据。 当装置开启/关闭或进行在线复位时，保留变量将数值保存其中。

❗如果断电期间装置处于“停止”状态，则保留变量的内容丢失！

14166

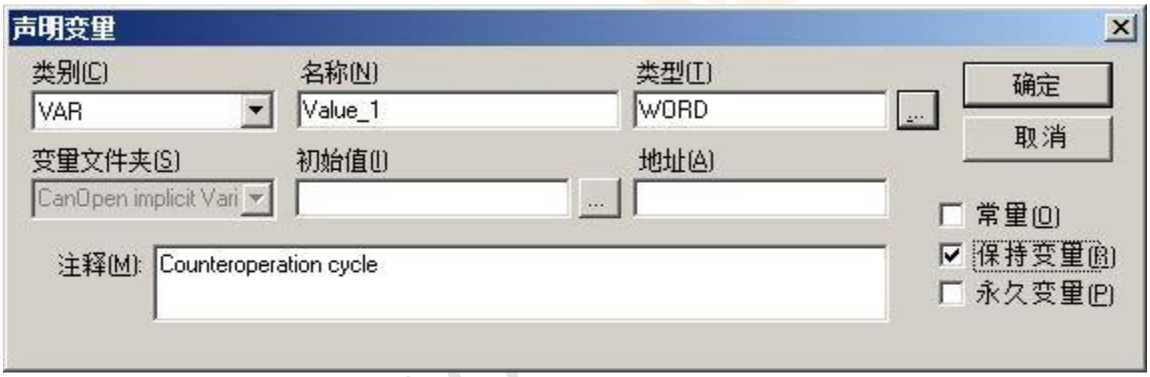
保持变量的一般应用如下：

- 机器运行时递增和保持的工作时间，
- 递增编码器的位置值，
- 监控器中输入的预设值，
- 机器参数，

即装置关闭时其值不得丢失的所有变量。

所有变量类型，还有复杂结构（如计时器），均可声明为保持变量。

► 为此，可启用变量声明中的控制字段 [RETAIN]（→ 窗口）。



4.5.2 网络变量

9856

全局网络变量用于网络中控制器之间的数据交换。 如果变量包含于其声明列表中，则全局网络变量的值可用于整个网络中的所有 CODESYS 项目。

► 将以下库集成至 CODESYS 项目：

  - `3S_CANopenNetVar.lib`

5 IFM 功能元件

内容

13586

所有 CODESYS 功能元件（FB、PRG、FUN）均存储在库中。 下文列出了所有 IFM 库的列表，您可将这些库用于本装置。

然后还有按主题分类的功能元件的说明。

5.1 针对装置 CR0033 的 IFM 库

内容

14235

5.1.1 库 ifm_CR0033_V02yyzz.LIB

19426

此为装置库。 该 IFM 库包含以下功能块：

功能元件	简短说明
CANx (→ 页 92)	初始化 CAN 接口 x x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_BAUDRATE (→ 页 93)	设定 CAN 接口 x 总线参与者的传送速度 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_BUSLOAD (→ 页 94)	确定 CAN 接口 x 的当前总线负载并计算出现的错误帧 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_DOWNLOADID (→ 页 96)	设定 CAN 接口 x 的下载 ID x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_ERRORHANDLER (→ 页 97)	在 CAN 接口 x 上执行"手动"总线恢复 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_RECEIVE (→ 页 98)	CAN interface x: 配置数据接收对象并读取数据对象的接收缓冲区 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_SDO_READ (→ 页 125)	CAN interface x: 读取 SDO，所示索引来自节点。 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_SDO_WRITE (→ 页 127)	CAN interface x: 写入 SDO，索引指定给节点。 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_TRANSMIT (→ 页 100)	将 CAN 数据对象（消息）传输至 CAN 接口 x 以在每次调用时传送 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CHECK_DATA (→ 页 253)	生成可配置内存区域的校验和 (CRC) 并检查针对不想要的更改的内存区域数据
DELAY (→ 页 225)	延迟输入值的输出，延迟时间 T（空载时间元件）
ERROR_REPORT (→ 页 262)	将应用程序特定错误报告至系统
ERROR_RESET (→ 页 264)	复位收到的错误消息
FAST_COUNT (→ 页 171)	针对快速输入脉冲的计数功能块
FLASHREAD (→ 页 244)	将不同的数据类型直接从闪存传输至 RAM
FLASHWRITE (→ 页 245)	将不同的数据类型直接写入闪存
FRAMREAD (→ 页 248)	将不同的数据类型直接从 FRAM 内存传输至 RAM FRAM 在此表示所有类型的非易失快速内存。
FRAMWRITE (→ 页 249)	将不同的数据类型直接写入 FRAM 内存 FRAM 在此表示所有类型的非易失快速内存。
FREQUENCY (→ 页 173)	测量到达所选通道的信号的频率。
FREQUENCY_PERIOD (→ 页 175)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）

功能元件	简短说明
GET_IDENTITY (→ 页 255)	读取存储在装置中的特定标识： • 装置的硬件名称和硬件版本 • 装置的序列号 • 装置运行时系统的名称 • 装置运行时系统的版本和版本号 • 应用程序的名称（之前已通过以下方式保存： SET_IDENTITY (→ 页 258))
INC_ENCODER (→ 页 178)	针对编码器评估的递增/递减计数器功能
INC_ENCODER_HR (→ 页 181)	针对编码器高分辨率评估的递增/递减计数器功能
INPUT_ANALOG (→ 页 157)	模拟输入通道： 以下对象的备选测量... • 电流 • 电压
MEMCPY (→ 页 250)	直接写入和读取内存中的不同数据类型
MEMORY_RETAIN_PARAM (→ 页 241)	确定不同事件的剩余数据特性
MEMSET (→ 页 251)	写入特定数据区
NORM (→ 页 165)	将定义的限值范围内的值 [WORD] 正常化为新的限值范围内的值
NORM_DINT (→ 页 167)	将定义的限值范围内的值 [DINT] 正常化为新的限值范围内的值
NORM_REAL (→ 页 168)	将定义的限值范围内的值 [REAL] 正常化为新的限值范围内的值
OUTPUT_BRIDGE (→ 页 195)	PWM 通道对上的 H 桥
OUTPUT_CURRENT (→ 页 199)	测量输出通道上的电流（平均，通过抖动周期）
OUTPUT_CURRENT_CONTROL (→ 页 200)	PWMi 输出通道的电流控制器
PACK_ERRORCODE (→ 页 267)	帮助将针对以下对象的字节打包为 ERRORCODE： • 错误类别 • 应用程序特定错误 • 错误来源 • 错误原因
PERIOD (→ 页 183)	测量所示通道频率和以 [µs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 185)	测量所示周期内所示通道频率和以 [µs] 为单位的周期（周期时间）。 此外，传号空号比以 [‰] 为单位显示。
PHASE (→ 页 187)	读取包含快速输入端的通道对并对比信号的相位
PID1 (→ 页 226)	PID 控制器
PID2 (→ 页 228)	PID 控制器
PT1 (→ 页 230)	受控系统，一级延迟
PWM1000 (→ 页 203)	初始化并配置具备 PWM 功能的输出通道 传号空号比可按 1 ‰ 的步距表示
SERIAL_PENDING (→ 页 145)	确定存储在序列接收缓冲区的数据字节的数量
SERIAL_RX (→ 页 146)	读取每次调用时来自序列接收缓冲区的已接收数据字节

功能元件	简短说明
SERIAL_SETUP (→ 页 148)	初始化 RS232 串行接口
SERIAL_TX (→ 页 150)	通过 RS232 串行接口传送一个数据字节
SET_DEBUG (→ 页 257)	安排 DEBUG 模式或监控模式（视 TEST 输入端而定）
SET_IDENTITY (→ 页 258)	设定应用程序特定程序 ID
SET_INPUT_MODE (→ 页 161)	将工作模式分配至输入通道
SET_INTERRUPT_I (→ 页 152)	通过定义的输入通道请求中断后有条件地执行程序部分
SET_INTERRUPT_XMS (→ 页 154)	以 xms 的时间间隔有条件地执行程序部分
SET_OUTPUT_MODE (→ 页 191)	设定所选输出通道的工作模式
SET_PASSWORD (→ 页 260)	设定程序和内存上传访问控制的用户密码
SHOW_ERROR_LIST (→ 页 268)	读取当前错误代码
SOFTRESET (→ 页 232)	使得装置完全重启
TEMPERATURE (→ 页 237)	读取装置中的当前温度
TIMER_READ (→ 页 234)	读取当前系统时间，以 [ms] 为单位 最大值 = 49d 17h 2min 47s 295ms
TIMER_READ_US (→ 页 235)	读取当前系统时间，以 [µs] 为单位 最大值 = 1h 11min 34s 967ms 295µs
UNPACK_ERRORCODE (→ 页 269)	帮助将 ERRORCODE 转换为针对以下对象的字节： • 错误类别 • 应用程序特定错误 • 错误来源 • 错误原因

5.1.2 库 ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB

13707

x = 1...4 = CAN 接口编号

该库包含用于将装置作为 CANopen 主站运行的功能块。

该 IFM 库包含以下功能块：

功能元件	简短说明
CANx_MASTER_EMCY_HANDLER (→ 页 103)	处理 CAN 接口 x 上 CANopen 主站的装置特定错误状态 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_MASTER_SEND_EMERGENCY (→ 页 104)	发送 CAN 接口 x 上 CANopen 主站的应用特定错误状态 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_MASTER_STATUS (→ 页 107)	在 CAN 接口 x 上显示作为 CANopen 主站的装置的状态 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

5.1.3 库 ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB

13709

x = 1...4 = CAN 接口编号

该库包含用于将装置作为 CANopen 从站运行的功能块。

该 IFM 库包含以下功能块：

功能元件	简短说明
CANx_SLAVE_EMCY_HANDLER (→ 页 115)	处理 CAN 接口 x 上 CANopen 从站的装置特定错误状态： • 错误寄存器（索引 0x1001）和 • CANopen 对象目录的错误场（索引 0x1003） x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_SLAVE_NODEID (→ 页 116)	在应用程序运行时间启用 CAN 接口 x CANopen 从站节点 ID 的设定 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_SLAVE_SEND_EMERGENCY (→ 页 117)	发送 CAN 接口 x 上 CANopen 从站的应用特定错误状态 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_SLAVE_SET_PREOP (→ 页 120)	在 CANopen 接口 x 上将该 CANopen 从站的工作模式从 "OPERATIONAL" 切换至 "OPERATIONAL" x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
CANx_SLAVE_STATUS (→ 页 121)	在 CAN 接口 x 上显示作为 CANopen 从站的装置的状态 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

5.1.4 库 ifm_CR0033_J1939_Vxxyyzz.LIB

13711

该库包含用于发动机控制的功能块。

该 IFM 库包含以下功能块：

功能元件	简短说明
J1939_x (→ 页 130)	CAN interface x: 通信协议 SAE J1939 的协议处理器 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
J1939_x_GLOBAL_REQUEST (→ 页 131)	CAN interface x: 处理 J1939 网络参与者的全局请求和数据接受 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
J1939_x_RECEIVE (→ 页 134)	CAN interface x: 接收单个消息或消息块 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
J1939_x_RESPONSE (→ 页 137)	CAN interface x: 处理针对请求消息的自动响应 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
J1939_x_SPECIFIC_REQUEST (→ 页 139)	CAN interface x: 特定 J1939 网络参与者自动请求单个消息 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）
J1939_x_TRANSMIT (→ 页 142)	CAN interface x: 发送单个消息或消息块 x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

5.1.5 库 ifm_hydraulic_32bit_Vxxyyzz.LIB

13729

该库包含用于液压控制的功能块。

该 IFM 库包含以下功能块：

功能元件	简短说明
CONTROL_OCC (→ 页 206)	OCC = Output Current Control 将输入值 [WORD] 换算至所示电流范围。
JOYSTICK_0 (→ 页 209)	将操纵杆的信号 [INT] 换算成明确定义的特性曲线，标准化为 0... 1000
JOYSTICK_1 (→ 页 213)	换算操纵杆 D 的信号 [INT]，标准化为 0... 1000
JOYSTICK_2 (→ 页 217)	将操纵杆的信号 [INT] 换算成可配置的特性曲线； 自由选择标准化
NORM_HYDRAULIC (→ 页 220)	将定义的限值范围内的值 [DINT] 正常化为新的限值范围内的值

5.2 针对装置 CR0033 的 IFM 功能元件

内容

13988
3826

您可在此了解适用于本装置的 IFM 功能元件（按主题分类）。

5.2.1 功能元件： CAN 第 2 层

内容

13754

下文将讲述应用程序中使用的 CAN 功能块（第 2 层）。

CANx

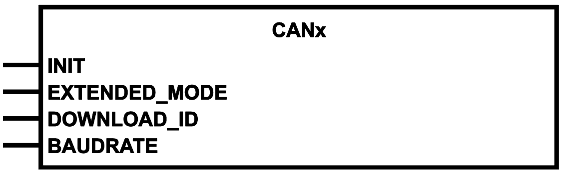
2159

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2162

CANx 初始化第 x 个 CAN 接口

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）.

每个接口的下载 ID 必须是不同的。

单个 CANx 的波特率可设为不同的值。

► 接口重启期间，仅可针对一个周期设定输入端 INIT！

! 下载 ID 和/或波特率的更改仅可在断电/通电后生效。

如果不执行元件，则接口适用于 11 位标识符。

输入端参数

2163

参数	数据类型	说明
INIT	BOOL	TRUE（第 1 个周期中）： 功能块初始化 FALSE: 在程序进一步处理期间
EXTENDED_MODE	BOOL := FALSE	TRUE: CAN 接口 ID 为 29 位 FALSE: CAN 接口 ID 为 11 位
DOWNLOAD_ID	BYTE	CAN 接口 x 的下载 ID x = 1...n = CAN 接口编号（视装置而定， → 技术资料） 有效 = 1...127 预设 = 127 - (x-1)
BAUDRATE	WORD := 125	波特率 [kbits/s] 有效 = 20、50、100、125、250、500、1000

CANx_BAUDRATE

11834

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：

说明

11839

CANx_BAUDRATE sets the transmission rate for the bus participant.

功能块用于设定装置的传送速度。 为此，可在输入端 BAUDRATE 输入以 Kbits/s 为单位的相应值。

> The new value will become effective on RESET (voltage OFF/ON or soft reset).

输入端参数

655

参数	数据类型	说明
ENABLE	BOOL	TRUE（第 1 个周期中）： 采用和启用参数 else: 未执行该功能
BAUDRATE	WORD := 125	波特率 [kbits/s] 有效 = 20、50、100、125、250、500、1000

CANx_BUSLOAD

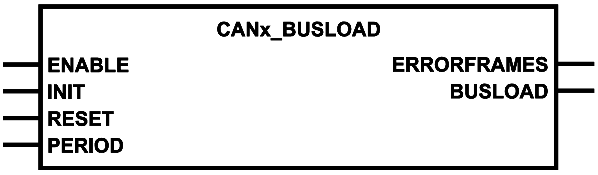
2178

x = 1...n = CAN 接口编号（视装置而定，→ 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2180

确定 CAN 总线上的当前总线负载并计算出现的错误帧。

CANx_BUSLOAD 考虑当前波特率，通过 PERIOD 所示时间内通过 CAN 总线传输的消息的数量和长度来确定总线负载。 PERIOD 所示时间结束后，值 BUSLOAD 会更新。

如果位 RESET 永久为 FALSE，则显示自上次 RESET 以来发生的错误帧的数量。

> **注意**
>
> 如果通过 CANopen 协议展开 CAN 总线通信，则可将 PERIOD 的值设为 SYNC 周期的持续时间。
>
> 测量周期不与 CANopen SYNC 周期同步。

输入端参数

2181

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
INIT	BOOL	TRUE（仅针对 1 个周期）： 配置测量持续时间 PERIOD FALSE: 在程序进一步处理期间
RESET	BOOL	TRUE: 将 ERRORFRAME 设为 "0" FALSE: function element is not executed
PERIOD	WORD	确定总线负载的时间，以 [ms] 为单位 允许 = 20...1 000 ms

输出端参数

2182

参数	数据类型	说明
ERRORFRAMES	WORD	自上次复位以来 CAN 总线上发生的错误帧的数量
BUSLOAD	BYTE	当前总线负载，按 [%] 计算

CANx_DOWNLOADID

11841

= CANx 下载 ID

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

说明

11846

CANx_DOWNLOADID sets the download identifier for the CAN interface x.

功能块可用于设定针对程序下载和调试的通信标识符。 输出端 ENABLE 设为 TRUE 时输入新值。

> ! The new value will become effective on RESET (voltage OFF/ON or soft reset).

输入端参数

649

参数	数据类型	说明
ENABLE	BOOL	TRUE（第 1 个周期中）： 采用和启用参数 else: 未执行该功能

CANx_ERRORHANDLER

2174

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

CANx_ERRORHANDLER
BUSOFF_RECOVER

说明

2329
13991

> ⓘ如果要使用自动总线恢复功能（默认设定），则**不得**在程序中集成功能 CANx_ERRORHANDLER 并实例化！

CANx_ERRORHANDLER 在 CAN 接口 x 上执行“手动”总线恢复。

► 进行 CAN 总线关闭识别后，通过 BUSOFF_RECOVER = TRUE 调用针对一个周期的功能块，以确保控制器可再次在 CAN 总线上发送和接收。

► 而后在应用程序中针对该 CAN 接口复位错误位 CANx_BUSOFF。

\> CAN 接口再次运行。

输入端参数

2177

参数	数据类型	说明
BUSOFF_RECOVER	BOOL	TRUE（仅 1 个周期）： 修正“总线关闭”状态 重启 CAN 接口 x FALSE: function element is not executed

CANx_RECEIVE

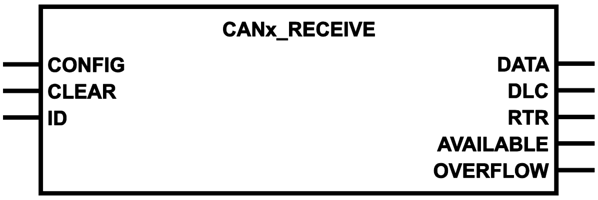
627

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

13338

CANx_RECEIVE 配置数据接收对象并读取数据对象的接收缓冲区。

► 初始化期间必须针对每个数据对象调用一次 FB，以向 CAN 控制器告知数据对象的标识符。

► 在进一步的程序周期中，调用 CANx_RECEIVE 以读取相应的接收缓冲区，若是长程序周期，还可重复。

► FB CANx_RECEIVE 最多允许 256 个实例，具体视 CAN 接口而定。

► 在标准模式中，所有 2048 个 ID 均可同时使用
在扩展模式中，仅 256 个 ID（任意）可同时使用

► 每个 ID（标准或扩展）仅可分配至一个 FB 实例。
若多次使用 ID： 调用最后的实例。

► 如果 CANx_RECEIVE 接收正常或扩展帧，则在 FB CANx 中设定。

> 如果 CANx_RECEIVE 配置为接收正常帧，则包含该 ID 的帧不会传输至 CANopen 堆栈（如可用）。

> 如果在允许范围之外设定 ID（视 CANx 中的设定而定），则不会执行功能块。

► 评估输出端 AVAILABLE，以便从缓冲区读取新接收的数据对象并及时处理。
接收缓冲区： 单位标识符最多 16 个软件缓冲区。

> 每次调用 FB，字节 AVAILABLE 的值会减少 1。
如果 AVAILABLE = 0，则缓冲区内无数据。

► 评估输出端 OVERFLOW，以检测数据缓冲区的溢出。
如果 OVERFLOW = TRUE，则至少丢失 1 个数据对象。

输入端参数

2172

参数	数据类型	说明
CONFIG	BOOL	TRUE（第 1 个周期中）： 配置数据对象 FALSE: 在程序进一步处理期间
CLEAR	BOOL	TRUE: 删除接收缓冲区 FALSE: function element is not executed
ID	DWORD	数据对象 ID 编号： 正常帧（2^{11} 个 ID）： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧（2^{29} 个 ID）： 0...536 870 911 = 0x0000 0000...0x1FFF FFFF

输出端参数

19810

参数	数据类型	说明
DATA	ARRAY [0..7] OF BYTE	已接收数据，（1...8 字节）
OVERFLOW	BOOL	TRUE: 数据缓冲区溢出 ⇨ 数据丢失！ FALSE: 数据缓冲区无数据丢失
AVAILABLE	BYTE	已从接收缓冲器接收但还未读取（调用 FB 之前）的 CAN 电报的数量。 可能的值 = 0...16 0 = 无可用的有效数据
DLC	BYTE	从接收缓冲区读取的 CAN 电报字节的数量 允许： 0...8
RTR	BOOL = FALSE	接收的消息为**远程传送**请求 (wird hier nicht unterstützt)

CANx_TRANSMIT

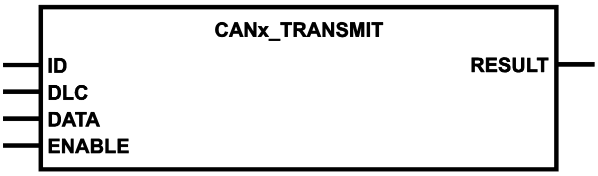
609

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2166

CANx_TRANSMIT 将 CAN 数据对象（消息）传送至 CAN 控制器以便传送。

在程序周期中，每个数据对象均调用 FB，若是长程序周期，还可重复。 程序员必须通过评估输出端 RESULT 确保已接受其传送命令。 简言之，若速度为 125 kbits/s，则每 1 ms 可执行一个传送命令。

可通过输入端 ENABLE 暂时阻止执行 FB (ENABLE = FALSE)。 因此，例如可阻止总线过载。

简单来说，若速度为 125 kbits/s，则每 1 ms 可执行一个传送命令。

如果标志分配给各个数据对象并通过 ENABLE 输入端控制 FB 的执行，则几乎可同时传送包含相同或不同 ID 的多个数据对象。

传送缓冲区： 针对所有标识符，最多 16 个软件缓冲区和 1 个硬件缓冲区

输入端参数

2167

参数	数据类型	说明
ID	DWORD	数据对象 ID 编号： 正常帧（2^{11} 个 ID）： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧（2^{29} 个 ID）： 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
DLC	BYTE	包含 SRDO 的 DATA 数组接收字节的数量 允许： 0...8
DATA	ARRAY [0..7] OF BYTE	待传送数据（1...8 字节）
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定

输出端参数

2168

参数	数据类型	说明
RESULT	BOOL	TRUE（仅针对 1 个周期）： 功能块接受了传送命令 FALSE: 未接受传送命令

5.2.2 功能元件： CANopen 主站

内容

1870

ifm electronic 提供一系列针对 CANopen 主站的 FB（下文将解释）。

CANx_MASTER_EMCY_HANDLER

2006

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB`

CODESYS 中的符号：

说明

2009

CANx_MASTER_EMCY_HANDLER 管理主站的装置特定错误状态。 必须在以下情况下调用 FB：

- 错误状态传送至网络，
- 应用程序的错误消息存储在对象目录中。

错误寄存器（索引 0x1001/01）中的当前值和 CANopen 对象目录中的错误字段（索引 0x1003/0-5）可通过 FB 读取。

> ❗如果要将应用程序特定错误消息存储在对象目录中，则调用 CANx_MASTER_EMCY_HANDLER **之前**必须（重复）调用 **CANx_MASTER_SEND_EMERGENCY** (→ 页 104)Y.

输入端参数

2010

参数	数据类型	说明
CLEAR_ERROR_FIELD	BOOL	FALSE ⇨ TRUE（边沿）： • 将 ERROR_FIELD 的内容传送至功能块输出端 • 删除对象目录中的 ERROR_FIELD 内容 else: 未执行该功能

输出端参数

2011

参数	数据类型	说明
ERROR_REGISTER	BYTE	显示 OBV 索引 0x1001（错误寄存器）的内容
ERROR_FIELD	ARRAY [0..5] OF WORD	显示 OBV 索引 0x1003（错误栏位）的内容 ERROR_FIELD[0]: 已存储错误的数量 ERROR_FIELD[1...5]: 已存储错误，索引 [1] 显示最近的错误

CANx_MASTER_SEND_EMERGENCY

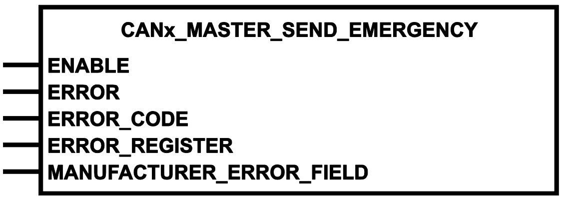
2012

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2015

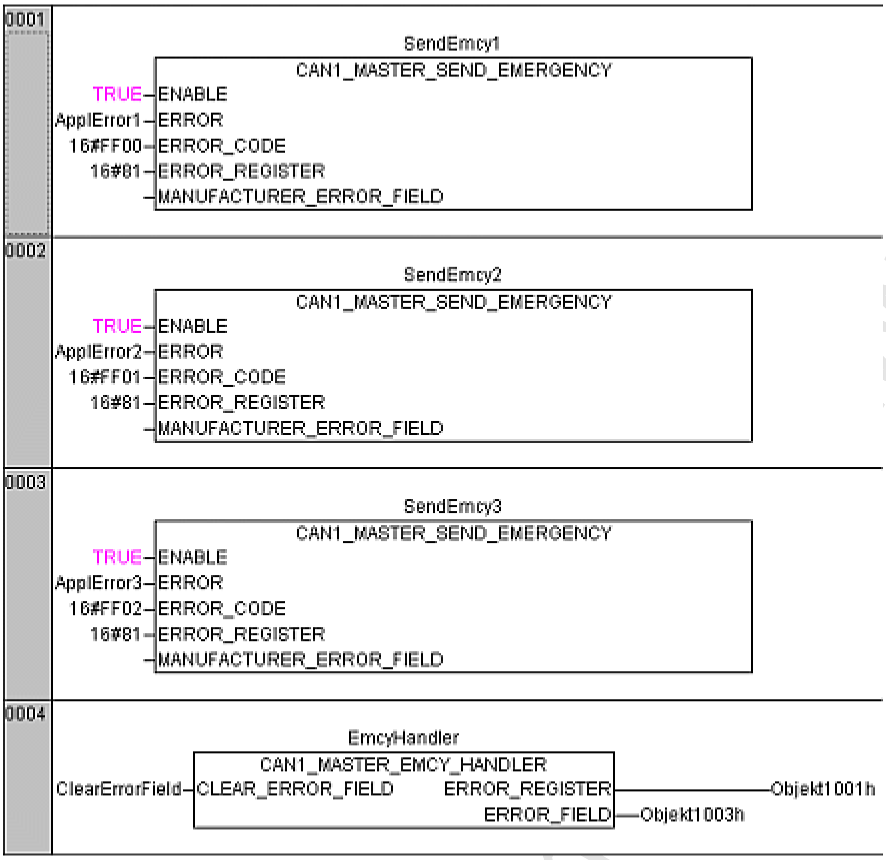
CANx_MASTER_SEND_EMERGENCY 传送应用程序特定错误状态。 如果要将错误状态传送至网络中的其他装置，则调用 FB。

> ! 如果要将应用程序特定错误状态存储在对象目录中，则必须在调用 **CANx_MASTER_EMCY_HANDLER** (→ 页 103) **之前**（重复）调用 CANx_MASTER_SEND_EMERGENCY。

输入端参数

2016 参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
ERROR	BOOL	与已配置错误代码相关的错误当前是否存在的信息通过该输入端传送。 FALSE ⇨ TRUE (edge): 发送下一个错误代码 如果在最后一秒输入端不为 TRUE TRUE ⇨ FALSE（边沿） 且不在显示故障： 延迟大约 1 s 之后： > 未发送错误消息 else: 未执行该功能
ERROR_CODE	WORD	错误代码提供关于已检测错误的详细信息。 应根据 CANopen 规范输入值。
ERROR_REGISTER	BYTE	ERROR_REGISTER 显示错误类型。 此处显示的值通过所有其他当前生效的错误消息的“或”运算来逐位链接。 生成的值写入错误寄存器（索引 1001_{16}/00）并通过 EMCY 消息传送。 应根据 CANopen 规范输入值。
MANUFACTURER_ERROR_FIELD	ARRAY [0..4] OF BYTE	在此可输入高达 5 字节的应用程序特定错误信息。 格式可以自由选择。

例如： CANx_MASTER_SEND_EMERGENCY



在该示例中，随后将生成 3 个错误消息：

1. ApplError1，代码 = 0xFF00，位于错误寄存器 0x81
2. ApplError2，代码 = 0xFF01，位于错误寄存器 0x81
3. ApplError3，代码 = 0xFF02，位于错误寄存器 0x81

CAN1_MASTER_EMCY_HANDLER 将错误消息发送至错误数组 "Object 0x1003" 中的错误寄存器 "Object 0x1001"。

CANx_MASTER_STATUS

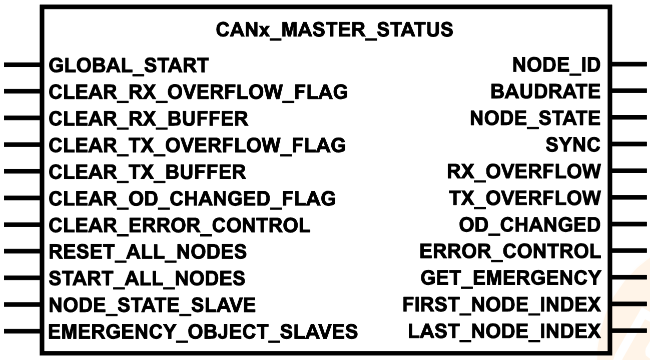
2692

x = 1...n = CAN 接口编号 (视装置而定， → 技术资料)

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2024

用作 CANopen 的装置的状态显示

CANx_MASTER_STATUS 限制作为 CANopen 主站的装置的状态。 其他可能性：

- 监控网络状态
- 监控已连接从站的状态
- 复位或启动网络中的从站。

FB 简化 CODESYS CANopen 主站库的使用。 强烈建议通过该 FB 评估网络状态和错误消息。

输入端参数

19861

参数	数据类型	说明
GLOBAL_START	BOOL	TRUE: 所有已连接的网络参与者 (从站) 在网络初始化期间同时启动 (⇨ 状态 OPERATIONAL) 。 FALSE: 已连接的网络参与者先后启动。
CLEAR_RX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (edge): 清除错误标志 RX_OVERFLOW else: 未执行该功能
CLEAR_RX_BUFFER	BOOL	FALSE ⇨ TRUE (edge): 删除接收缓冲区中的数据 else: 未执行该功能

参数	数据类型	说明
CLEAR_TX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (edge): 清除错误标志 TX_OVERFLOW else: 未执行该功能
CLEAR_TX_BUFFER	BOOL	FALSE ⇨ TRUE (edge): 删除传送缓冲区中的数据 else: 未执行该功能
CLEAR_OD_CHANGED_FLAG	BOOL	FALSE ⇨ TRUE (edge): 删除标志 OD_CHANGED else: 未执行该功能
CLEAR_ERROR_CONTROL	BOOL	FALSE ⇨ TRUE (edge): 删除保护错误列表 (ERROR_CONTROL) else: 未执行该功能
RESET_ALL_NODES	BOOL	FALSE ⇨ TRUE (edge): 所有已连接的网络参与者（从站） 通过 NMT 命令复位 else: 未执行该功能
START_ALL_NODES	BOOL	FALSE ⇨ TRUE (edge): 所有已连接的网络参与者（从站） 通过 NMT 命令启动 else: 未执行该功能
NODE_STATE_SLAVES	DWORD	数组指针地址 [0.. MAX_NODEINDEX] of CANx_NODE_STATE CANopen 网络中从站的状态信息将写入该数组。 从站的状况可通过访问特定值来控制。 MAX_NODEINDEX 为常数，应用程序编译期间可通过 CODESYS 计算。 ! 通过运算符 ADR 确定地址并将其分配至 FB！ 代码示例 → 章节 **例如： CANx_MASTER_STATUS** (→ 页 112)
EMERGENCY_OBJECT_SLAVES	DWORD	数组指针地址 [0.. MAX_NODEINDEX] of CANx_EMERGENCY_MESSAGE 显示所有网络节点的最新错误消息。 ! 通过运算符 ADR 确定地址并将其分配至 FB！

输出端参数

2696

参数	数据类型	说明
NODE_ID	BYTE	CANopen 从站当前节点 ID
BAUDRATE	WORD	CANopen 节点的当前波特率，以 [kBaud] 为单位
NODE_STATE	INT	CANopen 主站的当前状态

参数	数据类型	说明
SYNC	BOOL	CANopen 主站的 SYNC 信号 TRUE: 已在最后一个周期中发送 SYNC 信号 FALSE: 未在最后一个周期中发送 SYNC 信号
RX_OVERFLOW	BOOL	TRUE: Error: 接收缓冲区溢出 FALSE: 无溢出
TX_OVERFLOW	BOOL	TRUE: Error: 传送缓冲区溢出 FALSE: 无溢出
OD_CHANGED	BOOL	TRUE: CANopen 主站对象目录中的数据已更改 FALSE: 无数据更改
ERROR_CONTROL	ARRAY [0..7] OF BYTE	数组包含遗漏网络节点（保护或检测信号错误）的列表（最多 8 个）
GET_EMERGENCY	STRUCT CANx_EMERGENY_MESSAGE	在输出端，结构 CANx_EMERGENCY_MESSAGE 的数据可用。 始终显示 CANopen 网络中最后接收的 EMCY 消息。 若要获取所有发生的错误的列表，则必须评估数组 EmergencyObjectSlavesArray！
FIRST_NODE_INDEX	INT	显示连接至该 CAN 总线的节点（从站）的节点编号
LAST_NODE_INDEX	INT	

内部结构参数

2698

您可在此了解该功能块使用的数组的结构。

以控制器 CR0032 为例，以下代码段显示功能块 CANx_MASTER_STATUS 的使用 → 章节 **例如：CANx_MASTER_STATUS** (→ 页 112).

CANx_EMERGENCY_MESSAGE 结构

13996

结构由库 `ifm_CR0033_CANopenMaster_Vxxyyzz.LIB` 的全局变量决定。

参数	数据类型	说明
NODE_ID	BYTE	发送 EMCY 的参与者的节点 ID
ERROR_CODE	WORD	显示已出现错误的错误代码 → CANopen 规格 CiA 草案标准 301 第 4 版
ERROR_REGISTER	BYTE	发送参与者错误寄存器（索引 0x1001/00）中的值
MANUFACTURER_ERROR_FIELD	ARRAY [0..4] OF BYTE	EMCY 消息中的制造商特定数据字段

CANx_NODE_STATE 结构

13997

结构由库 ifm_CR0033_CANopenMaster_Vxxyyzz.LIB 的全局变量决定。

参数	数据类型	说明
NODE_ID	BYTE	状态信息和结构中配置标志所在的 CANopen 从站的节点 ID
NODE_STATE	BYTE	从 CANopen 主站 CANopen 堆栈角度所见的 CANopen 从站的当前状态
LAST_STATE	BYTE	CANopen 从站的最后已知状态 0 = 从 CANopen 从站接收启动消息 4 = CANopen 从站处于 PRE-OPERATIONAL 状态且通过 SDO 访问配置 5 = CANopen 从站处于 OPERATIONAL 状态 127 = CANopen 从站处于 PRE-OPERATIONAL 状态
RESET_NODE	BOOL	手动复位 CANopen 从站的标志 （NMT 命令 = Reset_Node）
START_NODE	BOOL	手动启动 CANopen 从站的标志 （NMT 命令 = start）
PREOP_NODE	BOOL	手动将 CANopen 从站设定为 PRE-OPERATIONAL 状态的标志 NMT 命令 = enter PRE-OPERATIONAL）
SET_TIMEOUT_STATE	BOOL	以下情况适用时用于手动跳过 CANopen 从站初始化的标志： • 网络中不存在从站 • **和** 从站未配置为可选
SET_NODE_STATE	BOOL	手动初始化 CANopen 从站的标志 访问对象 0x1000 时，从站已将自身识别为除控制器 CODESYS 配置 EDS 文件所示之外的装置类型。

例如： CANx_MASTER_STATUS

2031

从站信息

2699

若要能够访问单个 CANopen 节点的信息，则必须针对相应的结构创建数组。 结构包含于库中。 您可在库管理器中的 [Data types] 之下看到。

数组元件的数量由通过 CANopen 堆栈自动生成的全局变量 MAX_NODEINDEX 决定。 它包含从站数量减 1，如网络配置器所示。

> ! 数组元件的数量与节点 ID **不**一致。 可从 NODE_ID 下的相应结构读取标识符。

CAN1_MASTER_STATUS 程序示例

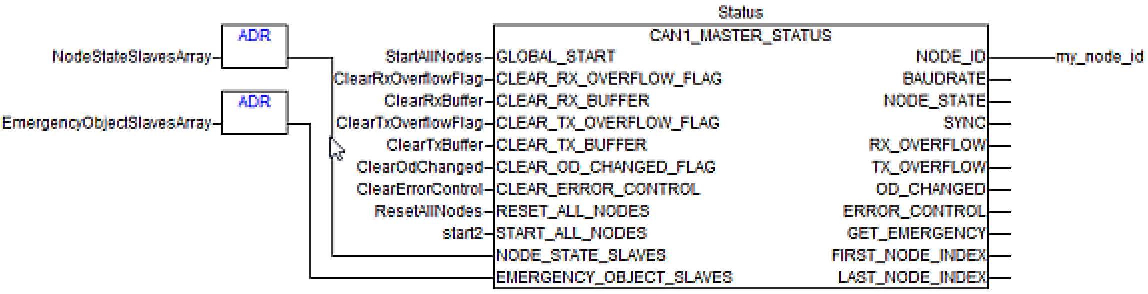
20651

变量声明：

```
VAR
    Status: CAN1_MASTER_STATUS;


    LedStatus: BOOL:= TRUE;
    StartAllNodes: BOOL:= TRUE;
    ClearRxOverflowFlag: BOOL;
    ClearRxBuffer: BOOL;
    ClearTxOverflowFlag: BOOL;
    ClearTxBuffer: BOOL;
    ClearOdChanged: BOOL;
    ClearErrorControl: BOOL;
    ResetAllNodes: BOOL;
    NodeStateSlavesArray: ARRAY [0..MAX_NODEINDEX] OF CAN1_NODE_STATE;
    EmergencyObjectSlavesArray: ARRAY[0..MAX_NODEINDEX] OF CAN1_EMERGENCY_MESSAGE;
    my_node_id: BYTE;
    my_baudrate: WORD;
    my_node_state: INT;
    Sync: BOOL;
    RxOverflow: BOOL;
    TxOverflow: BOOL;
    OdChanged: BOOL;
    GuardHeartbeatErrorArray: ARRAY[0..7] OF BYTE;
    GetEmergency: CAN1_EMERGENCY_MESSAGE;
    start2: BOOL;
    Ency_handler: CAN1_MASTER_EMCY_HANDLER;
    reset_emcy: BOOL;
END_VAR
```

程序示例：



节点状态结构

2034

```
TYPE CAN1_NODE_STATE :
STRUCT
    NODE_ID: BYTE;
    NODE_STATE: BYTE;
    LAST_STATE: BYTE;
    RESET_NODE: BOOL;
    START_NODE: BOOL;
    PREOP_NODE: BOOL;
    SET_TIMEOUT_STATE: BOOL;
    SET_NODE_STATE: BOOL;
END_STRUCT
END_TYPE
```

Emergency_Message 结构

2035

```
TYPE CAN1_EMERGENCY_MESSAGE :
STRUCT
    NODE_ID: BYTE;
    ERROR_CODE: WORD;
    ERROR_REGISTER: BYTE;
    MANUFACTURER_ERROR_FIELD: ARRAY[0..4] OF BYTE;
END_STRUCT
END_TYPE
```

5.2.3 功能元件： CANopen 从站

内容

1874

ifm electronic 提供一系列针对 CANopen 从站的 FB（下文将解释）。

CANx_SLAVE_EMCY_HANDLER

2050

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB

CODESYS 中的符号：

说明

2053

CANx_SLAVE_EMCY_HANDLER 处理 CANopen 从站的装置特定错误状态：

- 错误寄存器（索引 0x1001) 和
- CANopen 对象目录的错误字段（索引 0x1003）

► 在以下情况下调用功能块：

- 错误状态要传送至 CAN 网络，且

应用程序的错误消息要存储在对象目录中。

您是否想要将错误消息存储在对象目录中？ ► （重复）处理 后 **CANx_SLAVE_SEND_EMERGENCY (→ 页 117)** 之后，调用 CANx_SLAVE_EMCY_HANDLER 一次！

输入端参数

2054

参数	数据类型	说明
CLEAR_ERROR_FIELD	BOOL	FALSE ⇨ TRUE（边沿）： • 将 ERROR_FIELD 的内容传送至功能块输出端 • 删除对象目录中的 ERROR_FIELD 内容 else: 未执行该功能

输出端参数

2055

参数	数据类型	说明
ERROR_REGISTER	BYTE	显示 OBV 索引 0x1001（错误寄存器）的内容
	ARRAY [0..5] OF WORD	显示 OBV 索引 0x1003（错误栏位）的内容 ERROR_FIELD[0]: 已存储错误的数量 ERROR_FIELD[1...5]: 已存储错误，索引 [1] 显示最近的错误

CANx_SLAVE_NODEID

2044

= CANx 从站节点 ID

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB

CODESYS 中的符号：

说明

2049

CANx_SLAVE_NODEID 有助于在应用程序运行时间内设定 CANopen 从站节点 ID。

一般而言，在第一个周期，FB 在控制器初始化期间调用一次。 之后，输入端 ENABLE 再次设为 FALSE。

输入端参数

2047

参数	数据类型	说明
ENABLE	BOOL	FALSE ⇨ TRUE (edge): 采用和启用参数 else: 未执行该功能
NODEID	BYTE	节点 ID = 节点的 ID 允许值 = 1...127

CANx_SLAVE_SEND_EMERGENCY

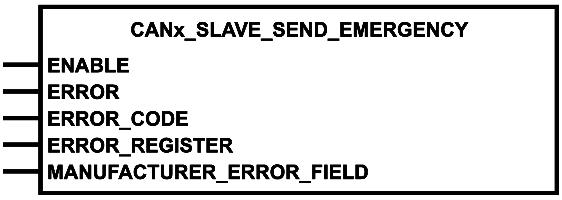
2056

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2059

CANx_SLAVE_SEND_EMERGENCY 传送应用程序特定错误状态。 它们是除装置内部错误消息（输出端短路）之外的待发送错误消息。

► 如果要将错误状态传送至网络中的其他装置，则调用 FB。

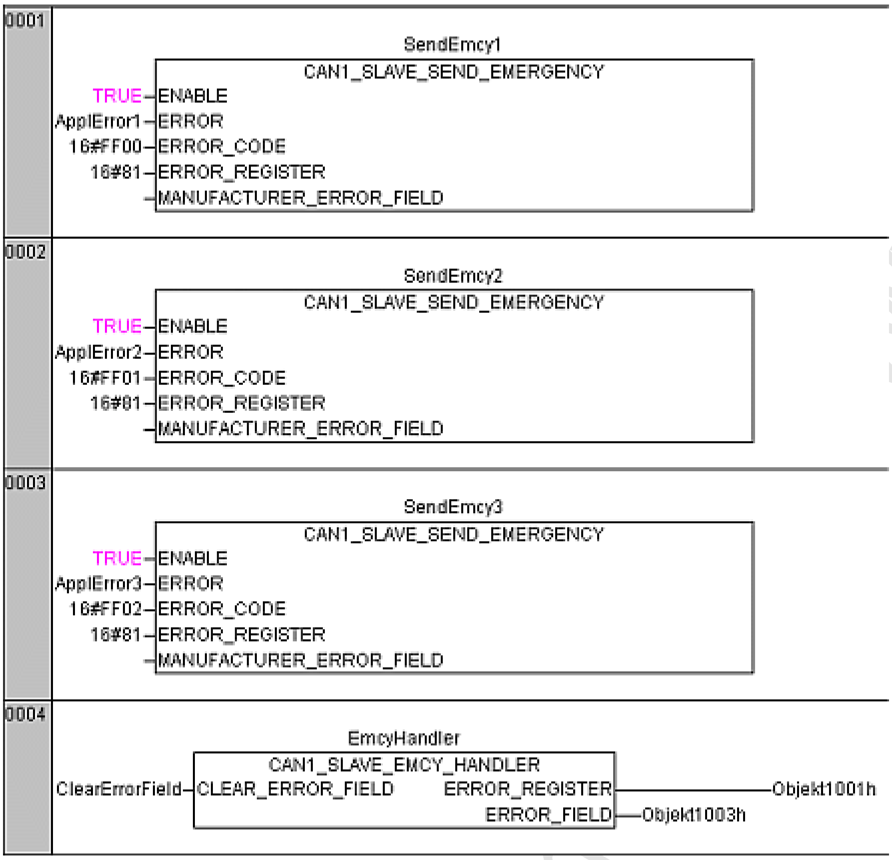
输入端参数

2060

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
ERROR	BOOL	与已配置错误代码相关的错误当前是否存在的信息通过该输入端传送。 FALSE ⇨ TRUE (edge): 发送下一个错误代码 如果在最后一秒输入端不为 TRUE TRUE ⇨ FALSE（边沿） 且不在显示故障： 延迟大约 1 s 之后： > 未发送错误消息 else: 未执行该功能
ERROR_CODE	WORD	错误代码提供关于已检测错误的详细信息。 应根据 CANopen 规范输入值。
ERROR_REGISTER	BYTE	ERROR_REGISTER 显示错误类型。 此处显示的值通过所有其他当前生效的错误消息的“或”运算来逐位链接。 生成的值写入错误寄存器（索引 1001_{16}/00）并通过 EMCY 消息传送。 应根据 CANopen 规范输入值。
MANUFACTURER_ERROR_FIELD	ARRAY [0..4] OF BYTE	在此可输入高达 5 字节的应用程序特定错误信息。 格式可以自由选择。

例如： CANx_SLAVE_SEND_EMERGENCY

2062



在该示例中，随后将生成 3 个错误消息：

1. ApplError1，代码 = 0xFF00，位于错误寄存器 0x81
2. ApplError2，代码 = 0xFF01，位于错误寄存器 0x81
3. ApplError3，代码 = 0xFF02，位于错误寄存器 0x81

CAN1_SLAVE_EMCY_HANDLER 将错误消息发送至错误数组 "Object 0x1003" 中的错误寄存器 "Object 0x1001"。

CANx_SLAVE_SET_PREOP

2700

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB

CODESYS 中的符号：

说明

2703

CANx_SLAVE_SET_PREOP 将该 CANopen 从站的工作模式从 "OPERATIONAL" 切换至 "PRE-OPERATIONAL"。

正常情况下，在出现故障时，控制器切换如下：

- FATAL ERROR 导致控制器 SOFT RESET
- ERROR STOP 导致 SYSTEM STOP

特定情况下，应用程序应将作为从站运行的装置的工作模式设为 "PRE-OPERATIONAL"。 该步骤可通过此处所述 FB 完成。

输入端参数

2704

参数	数据类型	说明
ENABLE	BOOL	FALSE ⇨ TRUE (edge): 将从站设为 PRE-OPERATIONAL else: 未执行该功能

CANx_SLAVE_STATUS

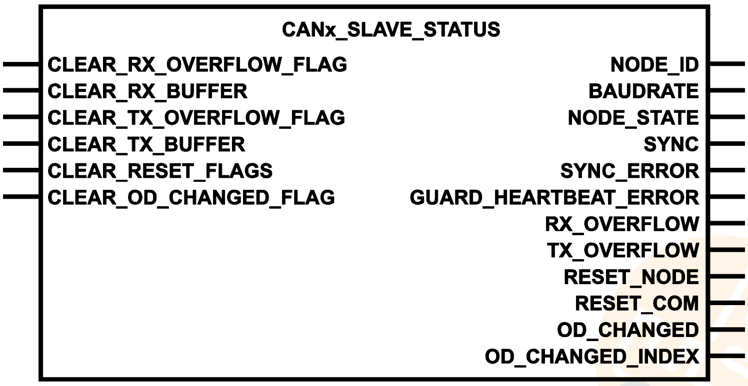
2706

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2707

CANx_SLAVE_STATUS 显示作为 CANopen 从站的装置的状态。

❗强烈建议通过该功能块评估网络状态。

输入端参数

2708

参数	数据类型	说明
CLEAR_RX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (edge): 清除错误标志 RX_OVERFLOW else: 未执行该功能
CLEAR_RX_BUFFER	BOOL	FALSE ⇨ TRUE (edge): 删除接收缓冲区中的数据 else: 未执行该功能
CLEAR_TX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (edge): 清除错误标志 TX_OVERFLOW else: 未执行该功能
CLEAR_TX_BUFFER	BOOL	FALSE ⇨ TRUE (edge): 删除传送缓冲区中的数据 else: 未执行该功能
CLEAR_RESET_FLAGS	BOOL	FALSE ⇨ TRUE (edge): 清除标志 RESET_NODE 清除标志 RESET_COM else: 未执行该功能
CLEAR_OD_CHANGED_FLAGS	BOOL	FALSE ⇨ TRUE (edge): 清除标志 OD_CHANGED 清除标志 OD_CHANGED_INDEX else: 未执行该功能

输出端参数

2068

参数	数据类型	说明
NODE_ID	BYTE	CANopen 从站当前节点 ID
BAUDRATE	WORD	CANopen 节点的当前波特率，以 [kBaud] 为单位
NODE_STATE	BYTE	CANopen 从站的当前状态 0 = 启动消息已发送 4 = CANopen 从站处于 PRE-OPERATIONAL 状态且通过 SDO 访问配置 5 = CANopen 从站处于 OPERATIONAL 状态 127 = CANopen 从站处于 PRE-OPERATIONAL 状态
SYNC	BOOL	CANopen 主站的 SYNC 信号 TRUE: 已在最后一个周期中接收 SYNC 信号 FALSE: 未在最后一个周期中接收 SYNC 信号
SYNC_ERROR	BOOL	TRUE: Error: 未接收主站的 SYNC 信号，或接收得太迟（ComCyclePeriod 过期之后） FALSE: 无 SYNC 错误
GUARD_HEARTBEAT_ERROR	BOOL	TRUE: Error: 未收到主站保护或检测信号错误，或接收得太迟 FALSE: 无保护或检测信号错误
RX_OVERFLOW	BOOL	TRUE: Error: 接收缓冲区溢出 FALSE: 无溢出
TX_OVERFLOW	BOOL	TRUE: Error: 传送缓冲区溢出 FALSE: 无溢出
RESET_NODE	BOOL	TRUE: 从站 CANopen 堆栈已通过从站复位 FALSE: 从站 CANopen 堆栈未复位
RESET_COM	BOOL	TRUE: CAN 堆栈的通信接口通过主站复位 FALSE: 通信接口未复位
OD_CHANGED	BOOL	TRUE: CANopen 主站对象目录中的数据已更改 FALSE: 无数据更改
OD_CHANGED_INDEX	INT	最后更改的对象目录条目的索引

5.2.4 功能元件： CANopen SDO

内容

2071

您可在此了解针对 CANopen 服务数据对象 (SDO) 处理的 **IFM** 功能元件。

CANx_SDO_READ

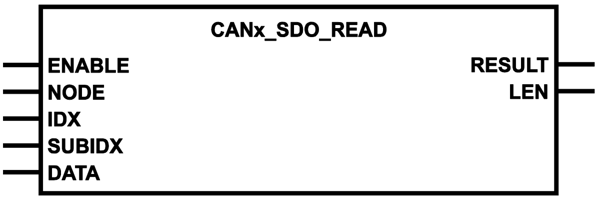
621

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

624

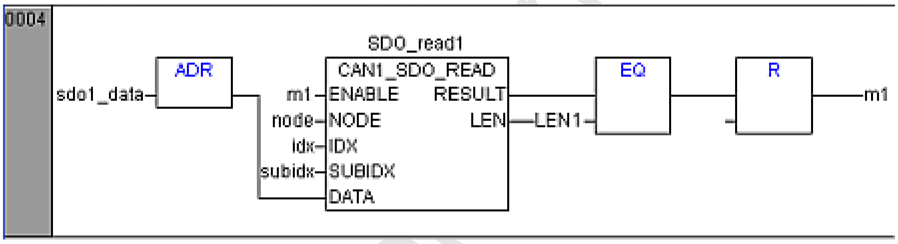
CANx_SDO_READ reads the →**SDO** (→ 页 395)，所示索引来自节点。

前提： 节点必须处于 "PRE-OPERATIONAL" 或 "OPERATIONAL" 模式。

可通过这些方法读取对象目录中的条目。 因此可选择性地读取节点参数。

> ⓘ 数据有丢失的危险！
> 针对所需 SDO 分配足够的内存空间！
> 否则之后的数据会被覆盖。

例如：



输入端参数

625

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
NODE	BYTE	节点 ID 允许值 = 1...127 = 0x01...0x7F
IDX	WORD	对象字典的索引
SUBIDX	BYTE	子索引参考对象目录中的索引
DATA	DWORD	接收数据数组地址 有效长度 = 0...255 通过运算符 ADR 确定地址并将其分配至 FB！

输出端参数

参数	数据类型	说明
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）
LEN	WORD	"字节数量"条目长度 LEN 的值不得大于接收数组的大小。 否则，将会覆盖应用程序中的数据。

RESULT 的可能结果：

值 十进制	十六进制	说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
2	02	功能块已启用（操作还未完成）
3	03	错误，监控时间内未接收数据

CANx_SDO_WRITE

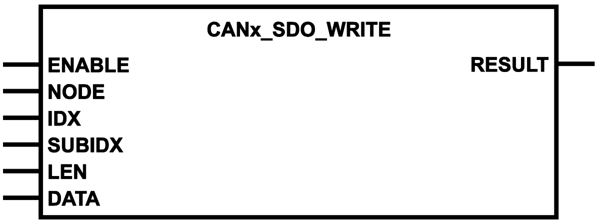
615

x = 1...n = CAN 接口编号（视装置而定，→ 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

618

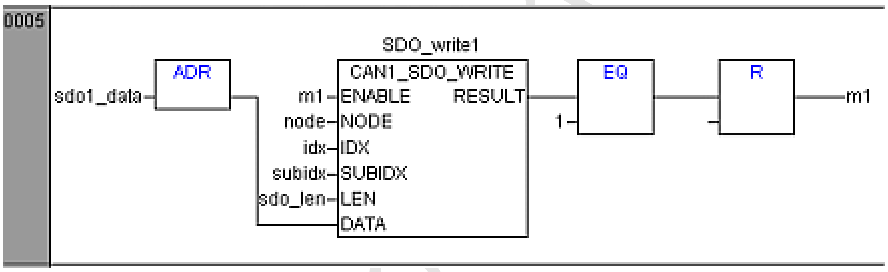
CANx_SDO_WRITE 写入 →**SDO** (→ 页 395)，索引指定给节点。

前提： 节点必须处于 "PRE-OPERATIONAL" 或 "OPERATIONAL" 状态。

可使用该 FB 将条目写入对象目录。 因此可选择性地设定节点参数。

❗LEN 的值必须低于传送数组的长度。 否则将发送随机数据。

例如：



输入端参数

619

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
NODE	BYTE	节点 ID 允许值 = 1...127 = 0x01...0x7F
IDX	WORD	对象字典的索引
SUBIDX	BYTE	子索引参考对象目录中的索引
LEN	WORD	“字节数量”条目长度 LEN 的值不得大于传送数组的大小。 否则，将会发送任意数据。
DATA	DWORD	传送数据数组地址 允许长度 = 0...255 通过运算符 ADR 确定地址并将其分配至 FB !

输出端参数

620

参数	数据类型	说明
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）

RESULT 的可能结果：

值 十进制	十六进制	说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
2	02	功能块已启用（操作还未完成）
3	03	错误，数据不能传送

5.2.5 功能元件： SAE J1939

内容

2273

就 SAE J1939 而言，**ifm electronic** 提供一系列功能元件，将在下文中解释。

J1939_x

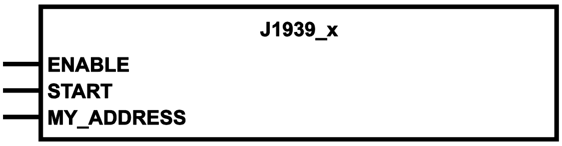
2274

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_J1939_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2276

J1939_x serves as protocol handler for the communication profile SAE J1939.

若要处理通信，必须在每个程序周期中调用协议处理器。 为此，输入端 ENABLE 设为 TRUE。

! 设定后，ENABLE 必须保持为 TRUE！

如果输入端 START 针对一个周期设为 TRUE，则启动协议处理器。

装置地址利用 MY_ADDRESS 分配至控制器。 必须不同于其他 J1939 总线参与者的地址。 而后可由其他总线参与者读取。

输入端参数

469

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
START	BOOL	TRUE（仅针对 1 个周期）： 开始 CAN 接口 x 的 J1939 协议 FALSE: 在程序进一步处理期间
MY_ADDRESS	BYTE	装置 J1939 地址

J1939_x_GLOBAL_REQUEST

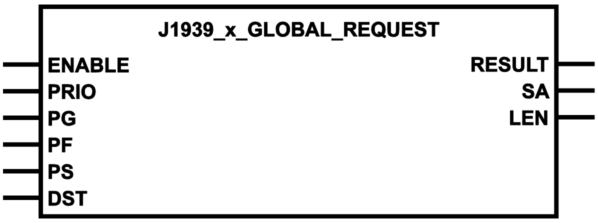
2282

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_J1939_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2301

J1939_x_GLOBAL_REQUEST 负责从所有（全局）有效的 J1939 网络参与者自动请求单个消息。 为此，参数 PG、PF、PS 以及存储已接收数据的数组 DST 地址均分配至 FB。

信息

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939 address] + [data]

13790

通知

存在非允许数据覆盖的风险！

- ▶ 创建大小为 1 785 字节的接收数组。
  这是 J1939 消息的最大大小。
- ▶ 检查已接收数据的数量：
  值不得超过为接收数据而创建的数组的大小！

就每个请求的消息而言，使用 FB 自身的实例！

- ▶ 就目标地址而言，DST 适用：
  ! 通过运算符 ADR 确定地址并将其分配至 FB！
- ▶ 此外必须分配优先级（一般为 3、6 或 7）。

- ▶ 鉴于数据请求可通过多个控制周期处理，该进程必须通过 RESULT 字节评估。
- RESULT = 2: POU 正在等待参与者的数据。
- RESULT = 1: 参与者接收了数据。
  输出端 LEN 显示已接收的数据字节数量。

立即存储/评估该新数据！
接收新消息时，内存地址 DST 中的数据会被覆盖。

- RESULT = 0: 1.25 秒内总线上无参与者发送回复。
  FB 返回至非活跃状态。
  仅现在 ENABLE 可再次设为 FALSE！

► 若要以短时间间隔从多个参与者接收数据：
在同一 PLC 周期中多次调用 POU 并立刻评估！

输入端参数

463

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
PRIO	BYTE	消息优先级 (0...7)
PG	BYTE	数据页 定义的 PGN（参数组编号）的值 允许 = 0...1（正常情况下 = 0)
PF	BYTE	PDU 格式字节 定义的 PGN（参数组编号）的值 PDU2（全局） = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN（参数组编号）的值 GE（群组扩展） = 0...255
DST	DWORD	目标地址 通过运算符 ADR 确定地址并将其分配至 FB！

信息

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939 address] + [data]

输出端参数

20789

参数	数据类型	说明
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）
SA	BYTE	响应装置 J1939 地址
LEN	WORD	已接收字节数量

RESULT 的可能结果：

值 十进制	十六进制	说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
2	02	功能块已启用（操作还未完成）

J1939_x_RECEIVE

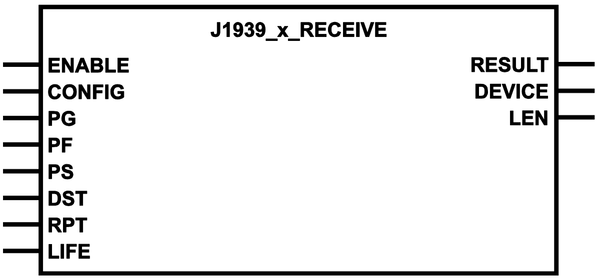
2278

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_J1939_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2288

J1939_x_RECEIVE 用于接收单个消息或消息块。

为此，必须通过输入端 CONFIG，针对一个周期初始化 FB。 初始化期间，分配参数 PG、PF、PS、RPT、LIFE 以及数据组 DST 的内存地址。

! 配置以下参数后不可在应用程序中加以修改。 PG, PF, PS, RPT, LIFE, DST.

13790

通知

存在非允许数据覆盖的风险！

- ▶ 创建大小为 1 785 字节的接收数组。
  这是 J1939 消息的最大大小。
- ▶ 检查已接收数据的数量：
  值不得超过为接收数据而创建的数组的大小！

就目标地址而言，DST 适用：

! 通过运算符 ADR 确定地址并将其分配至 FB！

! 一旦设定 RPT，则不可再修改！

▶ 必须通过 RESULT 字节评估数据的接收。 如果 RESULT = 1，则可从通过 DST 分配的内存地址读取数据并进一步处理。

> 接收新消息时，内存地址 DST 中的数据会被覆盖。

> 已接收消息字节的数量通过输出端 LEN 显示。

> 如果 RESULT = 3，则在所示时间窗口内未接收有效的消息 (LIFE • RPT)。

❗如果利用 FB J1939_..._REQUEST 请求消息，则还需使用该功能块。

输入端参数

457

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CONFIG	BOOL	TRUE（第 1 个周期中）： 配置数据对象 FALSE: 在程序进一步处理期间
PG	BYTE	数据页 定义的 PGN（参数组编号）的值 允许 = 0...1（正常情况下 = 0)
PF	BYTE	PDU 格式字节 定义的 PGN（参数组编号）的值 PDU1（特定） = 0...239 PDU2（全局） = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN（参数组编号）的值 如果 PF = PDU1 ⇨ PS = DA（目标地址） （DA = 外部装置的 J1939 地址） 如果 PF = PDU2 ⇨ PS = GE（群组扩展）
DST	DWORD	目标地址 ❗ 通过运算符 ADR 确定地址并将其分配至 FB！
RPT	TIME	监控时间 在该时间窗口内，必须循环接收消息。 > 否则将会出现错误消息。 RPT = T#0s ⇨ 无监控 ❗一旦设定 RPT，则不可再修改！
LIFE	BYTE	未接收允许的 J1939 消息数量

输出端参数

458

参数	数据类型	说明
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）
DEVICE	BYTE	发送器 J1939 地址
LEN	WORD	已接收字节数量

RESULT 的可能结果：

值 十进制	十六进制	说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
3	03	错误，监控时间内未接收数据

J1939_x_RESPONSE

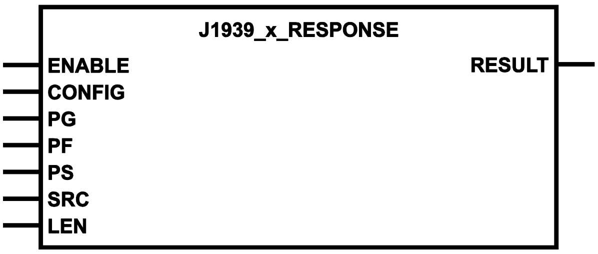
2280

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_J1939_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2299

J1939_x_RESPONSE 处理针对请求消息的自动响应。

该 FB 负责向“全局请求”和“特定请求”自动发送消息。 为此，必须通过输入端 CONFIG，针对一个周期初始化 FB。

参数 PG、PF、PS、RPT 以及数据组 SRC 地址分配至 FB。

- ► 就源地址而言，SRC 适用：
  通过运算符 ADR 确定地址并将其分配至 FB！
- ► 此外，还分配待传送数据字节的数量。

输入端参数

451

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CONFIG	BOOL	TRUE（第 1 个周期中）： 配置数据对象 FALSE: 在程序进一步处理期间
PG	BYTE	数据页 定义的 PGN（参数组编号）的值 允许 = 0...1（正常情况下 = 0)

参数	数据类型	说明
PF	BYTE	PDU 格式字节 定义的 PGN（参数组编号）的值 PDU1（特定） = 0...239 PDU2（全局） = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN（参数组编号）的值 如果 PF = PDU1 ➪ PS = DA（目标地址） （DA = 外部装置的 J1939 地址） 如果 PF = PDU2 ➪ PS = GE（群组扩展）
SRC	DWORD	源内存中的起始地址 通过运算符 ADR 确定地址并将其分配至 FB！
LEN	WORD	待传送数据字节数量 (≥ 1)

输出端参数

13993

参数	数据类型	说明
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）

RESULT 的可能结果：

值 十进制	十六进制	说明
0	00	FB 已禁用
1	01	数据传输完成且无错误
2	02	功能块已启用（操作还未完成）
3	03	错误，数据不能传送

J1939_x_SPECIFIC_REQUEST

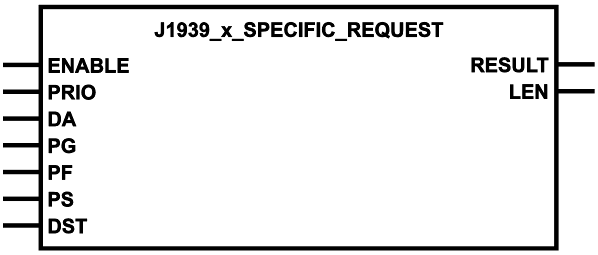
2281

x = 1...n = CAN 接口编号 (视装置而定， → 技术资料)

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_J1939_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2300

J1939_x_SPECIFIC_REQUEST 负责从特定 J1939 网络参与者自动请求单个消息。 为此，逻辑装置地址 DA，参数 PG、PF、PS 以及存储已接收数据的数组 DST 地址均分配至 FB。

信息

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939 address] + [data]

13790

通知

存在非允许数据覆盖的风险！

- ▶ 创建大小为 1 785 字节的接收数组。
  这是 J1939 消息的最大大小。
- ▶ 检查已接收数据的数量：
  值不得超过为接收数据而创建的数组的大小！

- ▶ 就目标地址而言，DST 适用：
  通过运算符 ADR 确定地址并将其分配至 FB！
- ▶ 此外必须分配优先级 (一般为 3、6 或 7)。
- ▶ 鉴于数据请求可通过多个控制周期处理，该进程必须通过 RESULT 字节评估。 如果 RESULT = 1，则所有数据均已接收。

> 输出端 LEN 显示已接收的数据字节数量。

输入端参数

445

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
PRIO	BYTE	消息优先级 (0...7)
DA	BYTE	所需装置 J1939 地址
PG	BYTE	数据页 定义的 PGN（参数组编号）的值 允许 = 0...1（正常情况下 = 0)
PF	BYTE	PDU 格式字节 定义的 PGN（参数组编号）的值 PDU1（特定） = 0...239 PDU2（全局） = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN（参数组编号）的值 如果 PF = PDU1 ⇨ PS = DA（目标地址） （DA = 外部装置的 J1939 地址） 如果 PF = PDU2 ⇨ PS = GE（群组扩展）
DST	DWORD	目标地址 ! 通过运算符 ADR 确定地址并将其分配至 FB！

信息

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939 address] + [data]

输出端参数

446

参数	数据类型	说明
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）
LEN	WORD	已接收字节数量

RESULT 的可能结果：

值 十进制	十六进制	说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
2	02	功能块已启用（操作还未完成）
3	03	错误

J1939_x_TRANSMIT

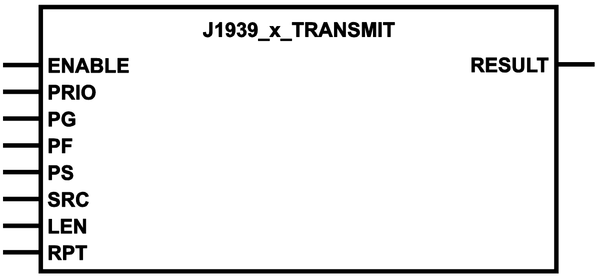
279

x = 1...n = CAN 接口编号（视装置而定， → 技术资料）

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_J1939_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2298

J1939_x_TRANSMIT 负责传送单个消息或消息块。 为此，参数 PG、PF、PS、RPT 以及数据组 SRC 地址分配至 FB。

信息

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939 address] + [data]

► 就源地址而言，SRC 适用：
 通过运算符 ADR 确定地址并将其分配至 FB！

► 此外，还必须分配待传送数据字节的数量和优先级（一般为 3、6 或 7）。

► 鉴于数据传送通过多个控制周期处理，进程必须通过 RESULT 字节评估。 如果 RESULT = 1，则所有数据均已传送。

如果要发送的字节超过 8 个字节，则进行“多包传输”。

输入端参数

439

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
PRIO	BYTE	消息优先级 (0...7)

参数	数据类型	说明
PG	BYTE	数据页 定义的 PGN（参数组编号）的值 允许 = 0...1（正常情况下 = 0)
PF	BYTE	PDU 格式字节 定义的 PGN（参数组编号）的值 PDU1（特定） = 0...239 PDU2（全局） = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN（参数组编号）的值 如果 PF = PDU1 ⇨ PS = DA（目标地址） （DA = 外部装置的 J1939 地址） 如果 PF = PDU2 ⇨ PS = GE（群组扩展）
SRC	DWORD	源内存中的起始地址 通过运算符 ADR 确定地址并将其分配至 FB！
LEN	WORD	待传送数据字节数量 允许 = 1...1 785 = 0x0001...0x06F9
RPT	TIME	数据消息循环传送的重复时间 RPT = T#0s ⇨ 仅发送一次

信息

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939 address] + [data]

输出端参数

440

参数	数据类型	说明
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）

RESULT 的可能结果：

值 十进制	十六进制	说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
2	02	功能块已启用（操作还未完成）
3	03	错误，数据不能传送

5.2.6 功能元件：串行接口

内容

13011

12998

注意

在默认情况下，串行接口对用户不可用，因为它用于程序下载和调试。

如果用户设定系统标志位 SERIAL_MODE=TRUE，则可自由使用接口。 而后仅可通过全部 4 个 CAN 接口中的一个调试应用程序。

下述功能块可供您在应用程序中使用串行接口。

SERIAL_PENDING

314

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：

SERIAL_PENDING
NUMBER

说明

12994

SERIAL_PENDING 确定存储在序列接收缓冲区的数据字节的数量。

跟 SERIAL_RX 相反，调用该功能后缓冲区的内容保持不变。

SERIAL FB 是创建串行接口应用程序特定协议的基础。
为此，需设定系统标志位 SERIAL_MODE=TRUE！

12998

> **注意**
>
> 在默认情况下，串行接口对用户不可用，因为它用于程序下载和调试。
>
> 如果用户设定系统标志位 SERIAL_MODE=TRUE，则可自由使用接口。 而后仅可通过全部 4 个 CAN 接口中的一个调试应用程序。

输出端参数

12996

参数	数据类型	说明
NUMBER	WORD	已接收数据字节数量 (1...1 000)

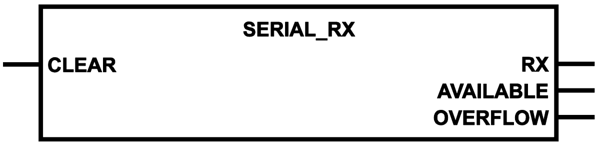
SERIAL_RX

308

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

12997

SERIAL_RX 读取每次调用时来自序列接收缓冲区的已接收数据字节。

如果接收 1000 以上数据字节，则缓冲区溢出，且数据丢失。 该情况通过位 OVERFLOW 显示。

如果采用 7 位数据传输，则第 8 位包含奇偶校验，必要时用户应加以阻止。

SERIAL FB 是创建串行接口应用程序特定协议的基础。
为此，需设定系统标志位 SERIAL_MODE=TRUE！

12998

> **注意**
>
> 在默认情况下，串行接口对用户不可用，因为它用于程序下载和调试。
>
> 如果用户设定系统标志位 SERIAL_MODE=TRUE，则可自由使用接口。 而后仅可通过全部 4 个 CAN 接口中的一个调试应用程序。

输入端参数

312

参数	数据类型	说明
CLEAR	BOOL	TRUE: 删除接收缓冲区 FALSE: function element is not executed

输出端参数

12931

参数	数据类型	说明
Rx	BYTE	从接收缓冲区接收的字节数据
AVAILABLE	WORD	调用功能块之前接收缓冲区可用的已接收字节数量： 0 = 未接收数据 1...1 000 = 接收缓冲区字节数量
OVERFLOW	BOOL	TRUE: 数据缓冲区溢出 ⇨ 数据丢失！ FALSE: 数据缓冲区无数据丢失

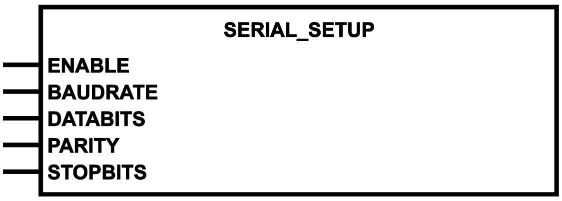
SERIAL_SETUP

302

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

13000

SERIAL_SETUP 初始化 RS232 串行接口。

不一定需要执行功能块才能够使用串行接口。 若未调用功能块，则最后的预设值适用。

功能块利用针对一个周期的 ENABLE=TRUE，将串行接口设为所示的参数。 在功能块的帮助下进行的更改将稳定保存。

12998

注意

在默认情况下，串行接口对用户不可用，因为它用于程序下载和调试。

如果用户设定系统标志位 SERIAL_MODE=TRUE，则可自由使用接口。 而后仅可通过全部 4 个 CAN 接口中的一个调试应用程序。

输入端参数

13002

参数	数据类型	说明
ENABLE	BOOL	TRUE（仅针对 1 个周期）： 初始化接口 FALSE: 在程序进一步处理期间
BAUD RATE	DWORD	波特率 允许值 → 数据表 预设值 → 数据表
DATABITS	BYTE := 8	数据位数量 允许 = 7 或 8
PARITY	BYTE := 0	奇偶校验 允许： 0 = 无，1 = 偶，2 = 奇 若参数设定 DATABITS = 7 且 PARITY = 0： 功能块在 PARITY = 1 时运行
STOPBITS	BYTE := 1	停止位数量 允许 = 1 或 2

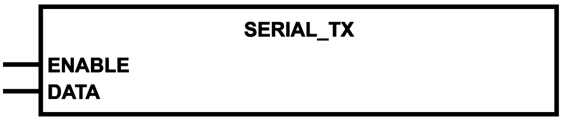
SERIAL_TX

296

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

13003

SERIAL_TX 通过 RS232 串行接口传送一个数据字节。

FiFo 传送内存包含 1 000 字节。

可利用输入端 ENABLE 启用或禁用传送。

SERIAL FB 是创建串行接口应用程序特定协议的基础。
为此，需设定系统标志位 SERIAL_MODE=TRUE！

12998

注意

在默认情况下，串行接口对用户不可用，因为它用于程序下载和调试。

如果用户设定系统标志位 SERIAL_MODE=TRUE，则可自由使用接口。 而后仅可通过全部 4 个 CAN 接口中的一个调试应用程序。

输入端参数

300

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
DATA	BYTE	待传送值

5.2.7 功能元件： 通过处理中断优化 PLC 周期

内容

0965
8609

您可在此了解优化 PLC 周期的功能。

1599

PLC 完整循环处理存储的应用程序。 周期时间会应程序分支而有所不同，具体视外部事件（= 条件转移）等而定。 这样可能对特定功能有不利影响。

可通过循环程序的系统中断，按照固定时间期限或在发生特定事件时调用时间关键型进程，而不考虑周期。

因为 SafetyController 大部分不允许中断功能，因此不可用。

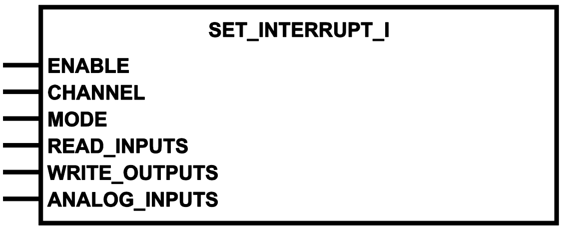
SET_INTERRUPT_I

2381

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

19234
11573

SET_INTERRUPT_I 通过输入通道，利用中断请求处理程序部分的执行。

在传统 PLC 中，周期时间决定实时监控。 因此，PLC 相比客户特定控制器处于不利地位。 当整个应用程序在一个单块中（不可更改）运行时，即便是“实时操作系统”亦无法改变这一事实。

可能的解决方案是尽量缩短周期时间。 这样的结果通常是，将应用程序分成若干控制周期。 但这样会使得编程变得复杂和困难。

还有一个可能性就是，在独立于控制周期之外的输入脉冲请求后，调用特定的程序部分：

用户在 PROGRAM (PRG) 类型程序块中集成应用程序的时间关键型部分。 调用 SET_INTERRUPT_I 一次（初始化期间）即可将该程序块声明为中断程序。 因此，如果输入端 CHANNEL 检测到边沿，则始终执行该程序块。 如果在该程序部分使用输入端和输出端，则亦可在中断程序中读取和写入，并通过输入边沿触发。 可通过 FB 输入端 READ_INPUTS, WRITE_OUTPUTS 和 ANALOG_INPUTS 停止读取和写入。

因此，在程序块中，可通过关联输出或全局变量和写入输出来处理所有时间关键型事件。 因此仅可在输入信号实际调用的情况下执行 FB。

> **❗注意**
>
> 应在周期中跳过程序块（安装调用除外），以便亦不会循环调用。
>
> 不可在中断程序中初始化和进一步处理监控中断触发的输入端 (CHANNEL)。
>
> 主要周期运行时加上通过中断程序调用的所有程序部分的持续时间总和必须始终在最大允许周期时间范围内！
>
> 用户负责主程序和在中断模式下运行的程序部分之间的数据一致性！

19866

中断优先级：

- 所有通过中断程序调用的程序部分拥有相同的执行优先级。 多个同时发生的中断可按照其发生的顺序连续处理。
- 如果在执行程序块（通过中断程序调用）时在相同输入端上检测到其他边沿，则列出中断程序以便处理，完成后直接再次调用程序。 作为选择，可消抖过滤器以过滤干扰多脉冲。
- 拥有更高优先级的中断程序（如 CAN）可打断在中断模式下运行的程序。
- 如果同一通道上有多个中断程序，则最后初始化的 FB（或 PRG）将分配至通道。 而后不再调用之前定义的 FB（或 PRG），且不再提供数据。

> ! 切**勿**搭配使用一个输入端上的该功能块与以下功能块之一！
> • INC_ENCODER_HR (→ 页 181)

输入端参数

2383

参数	数据类型	说明
ENABLE	BOOL	TRUE（仅针对 1 个周期）： 功能块初始化 FALSE: 单元未执行
CHANNEL	BYTE	中断输入端编号 输入端 IN00...IN07 对应 0...7
MODE	BYTE	触发中断的输入端 CHANNEL 边沿类型 1 = 上升沿（标准值） 2 = 下降沿 3 = 上升沿和下降沿 > 3 = 标准值
READ_INPUTS	BOOL	TRUE： 先读取输入端 0...7，再调用程序 然后写入至输入端标志 I00...I07 FALSE： 仅读取 CHANNEL 之下所示的通道 并写入至相应的输入端标志 Ixx
WRITE_OUTPUTS	BOOL	TRUE： 完成程序序列后将输出端标志 Q00...Q07 的当前值写入输出端 FALSE： 不写入输出端
ANALOG_INPUTS	BOOL	TRUE: 读取输入端 0...7 并将未经筛选和校准的 模拟值写入标志 ANALOG_IRQ00...07 FALSE: 切勿写入标志 ANALOG_IRQ00...07

SET_INTERRUPT_XMS

272

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

```
SET_INTERRUPT_XMS
ENABLE
REPEATTIME
READ_INPUTS
WRITE_OUTPUTS
ANALOG_INPUTS
```

说明

19363

SET_INTERRUPT_XMS 以 x ms 的时间间隔处理程序部分的执行。

在传统 PLC 中，周期时间决定实时监控。 因此，PLC 相比客户特定控制器处于不利地位。 当整个应用程序在一个单块中（不可更改）运行时，即便是“实时操作系统”亦无法改变这一事实。

可能的解决方案是尽量缩短周期时间。 这样的结果通常是，将应用程序分成若干控制周期。 但这样会使得编程变得复杂和困难。

还有一个可能性就是，按照独立于控制周期之外的固定时间间隔（每 x ms）调用特定的程序部分：

用户在 PROGRAM (PRG) 类型程序块中集成应用程序的时间关键型部分。 调用 SET_INTERRUPT_XMS 一次（初始化期间）即可将该程序块声明为中断程序。 因此，该程序块始终在 REPEATTIME 结束后（每 x ms）加以处理。 如果在该程序部分使用输入端和输出端，则亦可在定义的周期中读取和写入。 可通过 FB 输入端 READ_INPUTS, WRITE_OUTPUTS 和 ANALOG_INPUTS 停止读取和直接写入。

因此，在程序块中，可通过关联输出或全局变量和写入输出来处理所有时间关键型事件。 因此可比在“正常周期”中更准确地监控计时器。

注意

- 在整个程序中仅调用一次 SET_INTERRUPT_XMS，且仅在第一个周期中！
- 为避免中断程序调用的程序块被额外循环调用，应在周期中跳过（初始化调用除外）。

可启用多个计时器中断程序块。

允许时间的限制：

- 在中断模式下运行的程序部分的运行时 < REPEATTIME
- 主要周期运行时加上通过中断程序调用的所有程序部分的持续时间总和必须始终在最大允许周期时间范围内！

用户负责主程序和在中断模式下运行的程序部分之间的数据一致性！

请注意： 如果是高 CAN 总线活动，设定的 REPEATTIME 会波动。

每个中断模式运行的程序部分（作为 SET_INTERRUPT_XMS）的计数限制在 16。 除 SET_INTERRUPT_XMS 之外的中断模式运行的程序部分的更多定义将忽略且无效。

中断优先级：

- 所有通过中断程序调用的程序部分拥有相同的执行优先级。 多个同时发生的中断可按照其发生的顺序连续处理。
- 拥有更高优先级的中断程序（如 CAN）可打断在中断模式下运行的程序。

输入端参数

2382

参数	数据类型	说明
ENABLE	BOOL	TRUE（仅针对 1 个周期）： 功能块初始化 FALSE: 单元未执行
REPEATTIME	TIME	程序结束和重启之间的持续时间，以 [ms] 为单位 两次调用之间的持续时间确定为 REPEATTIME 总和，程序的运行时通过中断程序调用。
READ_INPUTS	BOOL	TRUE： 先读取输入端 0...7，再调用程序 然后写入至输入端标志 I00...I07 FALSE： 输入端无更新
WRITE_OUTPUTS	BOOL	TRUE： 完成程序序列后将输出端标志 Q00...Q07 的当前值写入输出端 FALSE： 不写入输出端
ANALOG_INPUTS	BOOL	TRUE: 读取输入端 0...7 并将未经筛选和校准的 模拟值写入标志 ANALOG_IRQ00...07 FALSE: 切勿写入标志 ANALOG_IRQ00...07

5.2.8 功能元件： 处理输入值

内容

1602
1302

在本章，我们将讲述可供您读取和处理装置输入端模拟或数字信号的 **IFM** 功能块。

注意

CODESYS PLC 配置所述的模拟原始值直接来自 ADC。 它们还未进行修正！

因此，针对同一装置的 PLC 配置中会显示不同原始值。

仅 IFM 功能块可进行错误修正和正常化。 功能块提供修正的值。

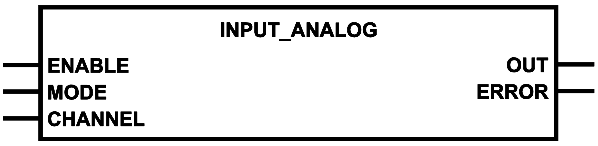
INPUT_ANALOG

15916

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

12912
12916

INPUT_ANALOG enables the following operating modes at the input channels.
详细信息 →章节 **可能的输入端/输出端工作模式** (→ 页 289)

功能块提供所选模拟通道的当前模拟值。 模拟值作为标准值提供。 同时，未校准的原始值通过系统标志 ANALOGxx 提供。

► 针对频率和周期测量以及计数器功能： 设定 MODE=1 (= IN_DIGITAL_H)！

测量值和输出值源自通过 MODE 显示的工作模式：

19260

MODE 十进制 | 十六进制		输入端工作模式	全局变量	FB 输出端 OUT	单位
0	0000	禁用	IN_NOMODE	---	---
1	0001	二进制输入端，负极开关 (BH)	IN_DIGITAL_H	0 / 1	---
2	0002	二进制输入端，正极开关 (BL)	IN_DIGITAL_L	0 / 1	---
4	0004	电流输入端	IN_CURRENT	0…20 000	µA
8	0008	电压输入端	IN_VOLTAGE10	0…10 000	mV
16	0010	电压输入端	IN_VOLTAGE30	0…32 000	mV
32	0020	电压输入端，比率	IN_RATIO	0…1 000	‰
64	0040	---	---	---	---
128	0080	---	---	---	---
256	0100	---	---	---	---
512	0200	电阻输入端	IN_RESISTANCE	3...680 16...30 000	Ω

20790

输出端的这些和其他工作模式还可通过以下方式配置：

SET_INPUT_MODE (→ 页 161)	将工作模式分配至输入通道

输出端的这些和其他工作模式： → FB SET_INPUT_MODE (→ 页 161).

- 如果 VBBs < 4 V （针对模式 1、2、32）： 不读取值
- 高于所示的值亦会记录（包括比率）
- 电流测量期间，过载保护始终启用

18414

如果未使用输入端 I15：

▶ 将输入端 I15 配置为二进制输入端！

13020

注意

在运行时期间切换至其他模式时，需先经过几个周期，而后输出值方可再次正确显示。

如果在运行时期间同一输入通道配置不同，则最近的配置适用。

输入端参数

11937

参数	数据类型	说明		
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定		
MODE	WORD	输如通道 CHANNE 工作模式：		
		0 = 0x0000	IN_NOMODE	（关闭； 预设生效）
		1 = 0x0001	IN_DIGITAL_H	预设
		2 = 0x0002	IN_DIGITAL_L	
		4 = 0x0004	IN_CURRENT	0...20 000 µA
		8 = 0x0008	IN_VOLTAGE10	0...10 000 mV
		16 = 0x0010	IN_VOLTAGE30	0...32 000 mV
		32 = 0x0020	IN_RATIO	0...1 000 ‰
		64 = 0x0040	---	
		128 = 0x0080	---	
		256 = 0x0100	---	
		512 = 0x0200	IN_RESISTANCE	3...680 Ω / 16...30 000 Ω
CHANNEL	BYTE	输入通道编号 输入端 I00...I15 对应 0...15 就功能块而言，xxx_E（如有）将适用： 输入端 I00_E...I23_E 对应 0...23		

输出端参数

11938

参数	数据类型	说明
OUT	WORD	基于 MODE 的输出值 设定无效时： OUT = "0"
ERROR	DWORD	来自该功能块调用的错误代码 → **错误代码** (→ 页 373) （可能的消息 → 下表）

ERROR 的可能结果（n= 任何所需值）：

32 位错误代码包含四个 8 位值 (DWORD)。

第 4 个字节	第 3 个字节	第 2 个字节	第 1 个字节
错误类别	应用程序特定错误代码	错误来源	错误原因

值 [hex]	说明
00 00 00 00	无错误
01 00 10+nn 01	输入通道 Inn 断线（nn = HEX 值） 例如： channel I15 ⇨ 10+nn = 1F
01 00 10+nn 02	输入通道 Inn 短路（nn = HEX 值） 例如： channel I15 ⇨ 10+nn = 1F （仅在 IN_DIGITAL_H 和 DIAGNOSTICS = TRUE 时）
01 00 10+nn 04	输入通道 Inn 过载（nn = HEX 值） 例如： channel I15 ⇨ 10+nn = 1F （仅针对电流测量或电阻测量）
01 00 00 F8	错误参数 ⇨ 一般错误

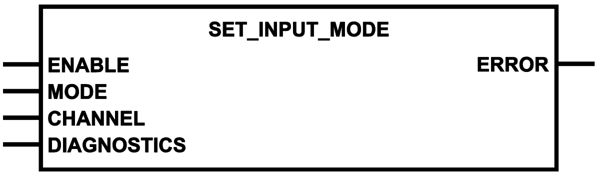
SET_INPUT_MODE

15918

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

11944

使用 SET_INPUT_MODE 以将工作模式分配至输入通道。

→ 章节 **可能的输入端/输出端工作模式** (→ 页 289)

- 高于所示的值亦会记录（包括比率）
- 电流测量期间，过载保护始终启用

运行期间不得更改工作模式。

功能块 **INPUT_ANALOG** (→ 页 157)亦可用于配置输出端工作模式。

18414

如果未使用输入端 I15：

▶ 将输入端 I15 配置为二进制输入端！

13020

注意

在运行时期间切换至其他模式时，需先经过几个周期，而后输出值方可再次正确显示。

如果在运行时期间同一输入通道配置不同，则最近的配置适用。

输入端参数

11945

参数	数据类型	说明
ENABLE	BOOL	FALSE ⇨ TRUE (edge): 初始化功能块（仅 1 个周期） > 读取功能块输入 TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
MODE	WORD	输如通道 CHANNE 工作模式： 0 = 0x0000 IN_NOMODE （关闭； 预设生效） 1 = 0x0001 IN_DIGITAL_H 预设 2 = 0x0002 IN_DIGITAL_L 4 = 0x0004 IN_CURRENT 0…20 000 µA 8 = 0x0008 IN_VOLTAGE10 0…10 000 mV 16 = 0x0010 IN_VOLTAGE30 0…32 000 mV 32 = 0x0020 IN_RATIO 0…1 000 ‰ 64 = 0x0040 --- 128 = 0x0080 ---
		256 = 0x0100 ---
		512 = 0x0200 IN_RESISTANCE 3...680 Ω / 16...30 000 Ω
CHANNEL	BYTE	输入通道编号 输入端 I00...I15 对应 0...15 就功能块而言，xxx_**E**（如有）将适用： 输入端 I00_E...I15_E 对应 0...15
DIAGNOSTICS	BOOL	TRUE: 包含诊断功能的通道 仅适用于 IN_DIGITAL_H 或 IN_RESISTANCE 错误消息： • 断线或接地短路 如果输入电压 < 1V 且持续 > 100 ms • 电源短路 如果输入电压 < 95 % VBBS 且持续 > 100 ms • 电阻器值 > 31 kOhm (I12...I14) 或 > 700 Ohm (I15)

输出端参数

11947

参数	数据类型	说明
ERROR	DWORD	来自该功能块调用的错误代码 → **错误代码** (→ 页 373) （可能的消息 → 下表）

ERROR 的可能结果（n= 任何所需值）：

32 位错误代码包含四个 8 位值 (DWORD)。

第 4 个字节	第 3 个字节	第 2 个字节	第 1 个字节
错误类别	应用程序特定错误代码	错误来源	错误原因

值 [hex]	说明
00 00 00 00	无错误
01 00 00 F8	错误参数 ⇨ 一般错误

5.2.9 功能元件： 调整模拟值

内容

1603

如果调整模拟输入值或模拟功能的结果，则以下 FB 可帮到您。

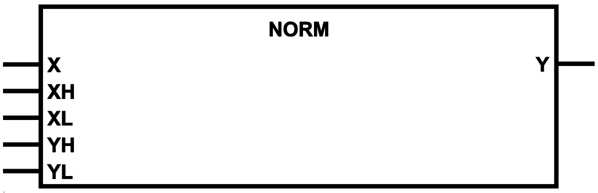
NORM

401

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

404

NORM 将定义的限值范围内的值正常化为新的限值范围内的值。

FB 将 XH 和 XL 限值范围内的类型 WORD 的值正常化为 YH 和 YL 限值范围内的输出值。 例如，该 FB 用于通过模拟输入值生成 PWM 值。

> **注意**
>
> ► X 值必须在 XL 和 XH 之间的定义输入范围！
> X 值无内部可靠性检查。
>
> > 因为舍入误差，正常化值的偏差可能为 1。
>
> > 如果按照倒数的形式定义限值（XH/XL 或 YH/YL），则正常化亦倒转。

输入端参数

405

参数	数据类型	说明
X	WORD	输入值
XH	WORD	输入值范围的上限值 [increments]
XL	WORD	输入值范围的下限值 [increments]
YH	WORD	输出值范围的上限值
YL	WORD	输出值范围的下限值

输出端参数

406

参数	数据类型	说明
Y	WORD	输出值

例如： NORM (1)

407

下限值输入端	0	XL
上限值输入端	100	XH
下限值输出端	0	YL
上限值输出端	2000	YH

而后 FB 转换输入信号，如下所示：

从 X =	50	0	100	75
	↓	↓	↓	↓
到 Y =	1000	0	2000	1500

例如： NORM (2)

408

下限值输入端	2000	XL
上限值输入端	0	XH
下限值输出端	0	YL
上限值输出端	100	YH

而后 FB 转换输入信号，如下所示：

从 X =	1000	0	2000	1500
	↓	↓	↓	↓
到 Y =	50	100	0	25

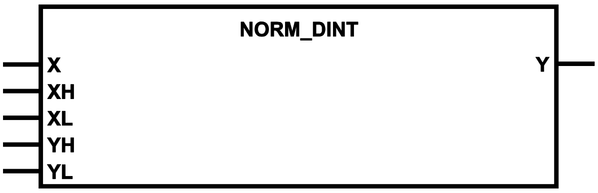
NORM_DINT

2217

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2355

NORM_DINT 将定义的限值范围内的值正常化为新的限值范围内的值。

FB 将 XH 和 XL 限值范围内的类型 DINT 的值正常化为 YH 和 YL 限值范围内的输出值。 例如，该 FB 用于通过模拟输入值生成 PWM 值。

注意

- X 值必须在 XL 和 XH 之间的定义输入范围！
  X 值无内部可靠性检查。
- 计算 (XH-XL)•(YH-YL) 的结果必须保持在数据类型 DINT 的值范围中
  (-2 147 483 648...2 147 483 647)!

\> 因为舍入误差，正常化值的偏差可能为 1。

\> 如果按照倒数的形式定义限值（XH/XL 或 YH/YL），则正常化亦倒转。

输入端参数

2359

参数	数据类型	说明
X	DINT	当前输入值
XH	DINT	输入值范围的上限值
XL	DINT	输入值范围的下限值
YH	DINT	输出值范围的上限值
YL	DINT	输出值范围的下限值

输出端参数

2360

参数	数据类型	说明
Y	DINT	输出值

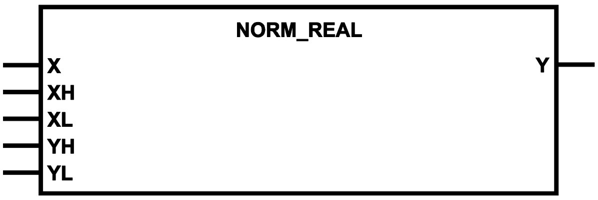
NORM_REAL

2218

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2358

NORM_REAL normalises a value within defined limits to a value with new limits.

FB 将 XH 和 XL 限值范围内的类型 REAL 的值正常化为 YH 和 YL 限值范围内的输出值。 例如，该 FB 用于通过模拟输入值生成 PWM 值。

注意

- ▶ X 值必须在 XL 和 XH 之间的定义输入范围！
  X 值无内部可靠性检查。
- ▶ 计算 (XH-XL)•(YH-YL) 的结果必须保持在数据类型 REAL 的值范围中 ($-3{,}402823466 \cdot 10^{38} \ldots 3{,}402823466 \cdot 10^{38}$)!
- > 因为舍入误差，正常化值的偏差可能为 1。
- > 如果按照倒数的形式定义限值（XH/XL 或 YH/YL），则正常化亦倒转。

输入端参数

2356

参数	数据类型	说明
X	REAL	输入值
XH	REAL	输出值范围的上限值
XL	REAL	输入值范围的下限值
YH	REAL	输出值范围的上限值
YL	REAL	输出值范围的下限值

输出端参数

2357

参数	数据类型	说明
Y	REAL	输出值

5.2.10 功能元件： 针对频率和周期测量的计数器功能

内容

19285

视 **ecomatmobile** 装置而定，可支持高达 16*) 个快速输入端，可处理高达 30 kHz 的输入频率。 除了频率测量之外，输入端还可用于评估增量编码器（计数器功能）。

*) ExtendedController： 高达 32 个快速输入端

因为测量方法不同，确定频率时可能会出现错误。

以下 FB 适用于简单评估：

功能元件	允许值	说明
FREQUENCY	0.1...30 000 Hz	测量所示通道上的频率。 频率高时测量错误减少
PERIOD	0.1...5 000 Hz	测量所示通道上的频率和周期（周期时间）
PERIOD_RATIO	0.1...5 000 Hz	测量所示通道上的频率和周期（周期时间）以及传号空号比 [‰]
FREQUENCY_PERIOD	0.1...30 000 Hz	FB 结合使用两个 FB FREQUENCY 以及 PERIOD 或 PERIOD_RATIO。 在 5 kHz 时自动选择测量方法
PHASE	0.1...5 000 Hz	读取通道对并对比信号的相位
INC_ENCODER	0.1...30 000 Hz	针对编码器评估的递增/递减计数器功能
INC_ENCODER_HR	0.1...5 000 Hz	针对编码器高分辨率评估的递增/递减计数器功能
FAST_COUNT	0.1...30 000 Hz	快速脉冲计数

将快速输入端作为“正常”数字输入端时的重要事项：

- 必须考虑提高的噪声脉冲敏感度（例如机械触点的触点弹跳）。
- 如有必要，必须针对输入信号去抖动！ → 章节 **配置硬件过滤器** (→ 页 77)
- 标准数字输入端可评估高达 50 Hz 的信号（视 IEC 周期和过滤器设定而定）。

FAST_COUNT

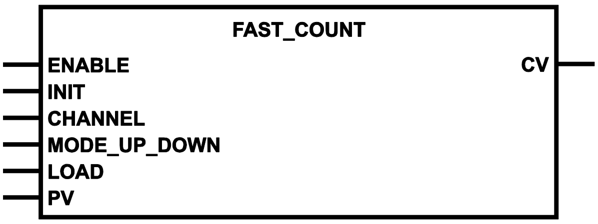
15922

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

20653

FAST_COUNT 作为快速输入脉冲的计数器块运行。

ENABLE=TRUE 期间，功能块检测 FRQ 输入通道的上升沿。

最大输入频率 → 数据表

如果复位和新设 ENABLE，计数器则继续从在最后复位 ENABLE 时有效的值开始计数。

设定 INIT（上升沿）时，计数值 CV 设为 0。

复位参数 INIT 时，计数器从 0 开始计数。

22690

以下适用于装置的标准侧：

该功能块可在同一输出端跟以下功能块一起使用： **INC_ENCODER** (→ 页 178).

22692

► 同时初始化并联运行的功能块。

> 功能块 FAST_COUNT 的后续下载会影响功能块 INC_ENCODER 的值。

切**勿**搭配使用一个输入端上的该功能块与以下功能块之一！

- **FREQUENCY** (→ 页 173)
- **FREQUENCY_PERIOD** (→ 页 175)
- **INC_ENCODER_HR** (→ 页 181)
- **PERIOD** (→ 页 183)
- **PERIOD_RATIO** (→ 页 185)
- **PHASE** (→ 页 187)

14888

注意

如果频率较高（高于 IFM 保证的水平），则会出现以下问题：

- 输出端的开启和关闭时间将变得更重要。
- 组件过度发热。

上述影响取决于每种情况下使用的组件。
这些潜在影响无法准确预测。

输入端参数

19869

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 计数器停止
INIT	BOOL	FALSE ⇨ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	快速输入通道编号 输出端 I00...I11 对应 0...11 就功能块而言，xxx_E（如有）将适用： 输出端 I00_E...I15_E 对应 0...15
MODE_UP_DOWN	BOOL	TRUE： 计数器向下计数 FALSE： 计数器向上计数
LOAD	BOOL	TRUE： 起始值在 CV 中加载 FALSE: function element is not executed
PV	DWORD	计数器初始值（预设值）

输出端参数

572

参数	数据类型	说明
CV	DWORD	当前计数值 过溢时的状况： • 向下计数时计数器停止在 0 的位置 • 向上计数时出现过溢

FREQUENCY

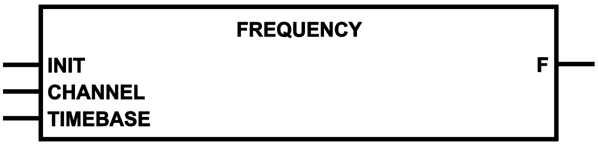
15924

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

ⓘ就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

20656
20675

FREQUENCY 测量到达所选 CHANNEL 的信号的频率。 FB 评估信号的上升沿。

可在广阔的数值范围内测量频率，具体视 TIMEBASE 而定。

- 高频率需要短时间基数
- 低频率需要较长的时间基数

限值：

TIMEBASE	允许，可测量频率
57 000 ms （= 最大值）	1 149 Hz
2 184 ms	30 000 Hz （= 最大值）

待测量频率的时间基数越长，确定的测量值越精确。

频率 = 1 Hz 的示例：

时间基数 [ms]	最高错误率 [%]	测量值 [Hz]
1 000	100	0...2
10 000	10	0.9...1.1

频率直接以 [Hz] 为单位显示。

14888

注意

如果频率较高（高于 IFM 保证的水平），则会出现以下问题：

- 输出端的开启和关闭时间将变得更重要。
- 组件过度发热。

上述影响取决于每种情况下使用的组件。

这些潜在影响无法准确预测。

7321

针对频率测量： 确保功能块在 TIMEBASE 值内未接收 65 535 个以上的上升沿！
否则，内部计数寄存器会溢出并导致结果有误！

22690

以下适用于装置的标准侧：
该功能块可在同一输出端跟以下功能块一起使用： INC_ENCODER (→ 页 178).

22691

▶ 同时初始化并联运行的功能块。

切勿搭配使用一个输入端上的该功能块与以下功能块之一！
- FAST_COUNT (→ 页 171)
- FREQUENCY_PERIOD (→ 页 175)
- INC_ENCODER_HR (→ 页 181)
- PERIOD (→ 页 183)
- PERIOD_RATIO (→ 页 185)
- PHASE (→ 页 187)

输入端参数

19871

参数	数据类型	说明
INIT	BOOL	TRUE（仅针对 1 个周期）： 功能块和接口初始化 FALSE: 测量正在进行中 或者：如果之前 INIT=TRUE，则测量开始
CHANNEL	BYTE	快速输入通道编号 输出端 I00...I11 对应 0...11 就功能块而言，xxx_E（如有）将适用： 输出端 I00_E...I15_E 对应 0...15
TIMEBASE	TIME	频率测量时基（最多 57 s）

输出端参数

542

参数	数据类型	说明
F	REAL	输入信号的频率，以 [Hz] 为单位

FREQUENCY_PERIOD

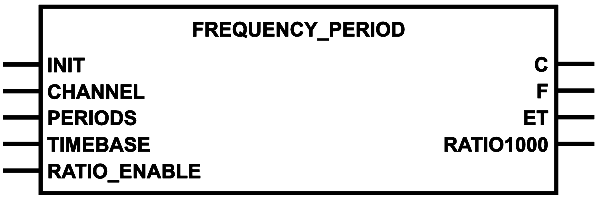
15926

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

20659
20676

FREQUENCY_PERIOD 测量所示通道（针对所有输入端）上的频率和以 [µs] 为单位的周期（周期时间）。 最大输入频率 → 数据表

FB 在通用功能中结合使用 PERIOD/PERIOD_RATIO 和 FREQUENCY。 测量方法在 5 kHz 左右时自动选择：

- 在 5.2 kHz 以下时，FB 像 PERIOD 或 PERIOD_RATIO 一样运行。
- 在 5.5 kHz 以上时，FB 像 FREQUENCY 一样运行。

该 FB 测量所选 CHANNEL 信号的频率和周期时间。 若要计算，则评估所有上升沿并利用所示 PERIODS 的数字确定平均值。

如果输入频率 > 5 kHz 且 FREQUENCY 模式启用，则不可测量比率。

最大测量范围大约为 15 min。

14888

注意

如果频率较高（高于 IFM 保证的水平），则会出现以下问题：

- 输出端的开启和关闭时间将变得更重要。
- 组件过度发热。

上述影响取决于每种情况下使用的组件。
这些潜在影响无法准确预测。

7321

针对频率测量： 确保功能块在 TIMEBASE 值内未接收 65 535 个以上的上升沿！
否则，内部计数寄存器会溢出并导致结果有误！

22690

以下适用于装置的标准侧：
该功能块可在同一输出端跟以下功能块一起使用： INC_ENCODER (→ 页 178).

22691

► 同时初始化并联运行的功能块。

注意

切**勿**搭配使用一个输入端上的该功能块与以下功能块之一！
- FAST_COUNT (→ 页 171)
- FREQUENCY (→ 页 173)
- INC_ENCODER_HR (→ 页 181)
- PERIOD (→ 页 183)
- PERIOD_RATIO (→ 页 185)
- PHASE (→ 页 187)

输入端参数

19872

参数	数据类型	说明
INIT	BOOL	TRUE（仅针对 1 个周期）： 功能块和接口初始化 FALSE: 测量正在进行中 或者：如果之前 INIT=TRUE，则测量开始
CHANNEL	BYTE	快速输入通道编号 输出端 I00...I11 对应 0...11 就功能块而言，xxx_E（如有）将适用： 输出端 I00_E...I15_E 对应 0...15
PERIODS	BYTE	待平均周期数量 (1...16) 0 : 输出端 C 和 F 未更新 > 16 : 限制在 16 以内
TIMEBASE	TIME	频率测量时基（最多 57 s）
RATIO_ENABLE	BOOL	TRUE: 比率测量提供给 RATIO1000 FALSE: 未提供比率测量

输出端参数

2337

参数	数据类型	说明
C	DWORD	已检查周期的周期时间 [µs] 允许 = 33...10 000 000 = 0x21...0x989680
F	REAL	输入信号的频率，以 [Hz] 为单位
ET	TIME	针对间隔测量： （可用于非常慢的信号） RATIO_ENABLE = TRUE: 自输入端边沿最后更改以来消耗的时间 RATIO_ENABLE = FALSE: 自输入端最后一个上升边沿以来消耗的时间 针对其他测量： ET = 0
ATIO1000	WORD	传号空号比，以 [‰] 为单位 允许 = 1...999 = 0x1...0x3E7 前提： • 针对间隔测量 • 脉冲持续时间 ≥ 100 µs • 频率 < 5 kHz

INC_ENCODER

15928

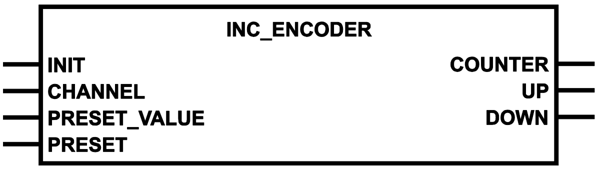
= 增量编码器

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

19302

INC_ENCODER offers up/down counter functions for the evaluation of encoders.

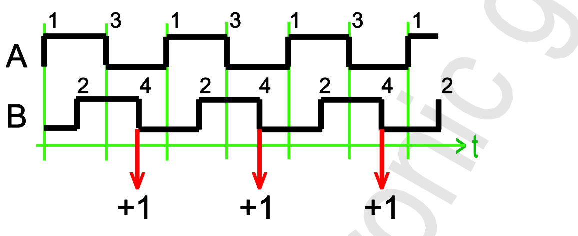
每个通过功能块评估的输入对 CHANNEL 通过两个邻近的频率输入端形成。

限制频率 = 30 kHz

最大待连接元件数量： 4 个编码器（ExtendedController： 最多 8 个编码器）

设定预设值：

1. 在 PRESET_VALUE 中输入值
2. 针对一个周期将 PRESET 设为 TRUE
3. 将 PRESET 复位至 FALSE



只要 INIT=FALSE 且 PRESET=FALSE，功能块在输入对每 4 个脉冲进行计数（与编码器分辨率一致）。

电流计数值位于输出端 COUNTER。

输出端 UP 和 DOWN 表示计数器当前的计数方向。 如果计数器在之前程序周期中按相应的方向计数（即便过溢），则输出端为 TRUE。 如果计数器停止，后续程序周期中的方向输出端亦复位。

22616

以下适用于装置的标准侧：
针对功能块的每个物理输入端 INC_ENCODER (→ 页 178)，可额外之用以下 POU 中的**一个**：

- FAST_COUNT (→ 页 171)
- FREQUENCY (→ 页 173)
- FREQUENCY_PERIOD (→ 页 175)
- PERIOD (→ 页 183)
- PERIOD_RATIO (→ 页 185)
- PHASE (→ 页 187)

22691

► 同时初始化并联运行的功能块。

以下适用于装置的扩展侧：
就功能块 INC_ENCODER_E 的每一个物理输入端而言，**不可**额外使用更多的功能块。

输入端参数

529

参数	数据类型	说明
INIT	BOOL	TRUE（仅针对 1 个周期）： 功能块初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	输入通道对 x 编号 0 = 通道对 0 = 输入端 I00 + I01 ... 7 = 通道对 7 = 输入端 I14 + I15 （0...x，值视装置而定→ 技术 资料） 0 = 通道对 0 = 输入端 I00_E + I01_E ... 7 = 通道对 7 = 输入端 I14_E + I15_E
PRESET_VALUE	DINT	计数器初始值
PRESET	BOOL	FALSE ⇨ TRUE (edge): PRESET_VALUE 加载至 COUNTER TRUE： 计数器忽略输入脉冲 FALSE： 计数器计算输入脉冲

输出端参数

530

参数	数据类型	说明
COUNTER	DINT	当前计数值

参数	数据类型	说明
UP	BOOL	TRUE： 计数器在最后一个周期中向上计数 FALSE： 计数器在最后一个周期中未向上计数
DOWN	BOOL	TRUE： 计数器在最后一个周期中向下计数 FALSE： 计数器在最后一个周期中未向下计数

INC_ENCODER_HR

19225

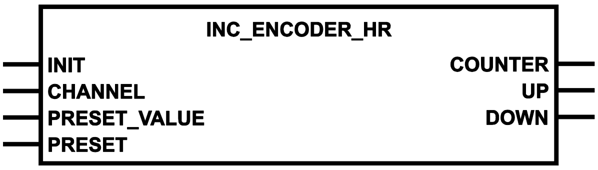
= 增量编码器高分辨率

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。

CODESYS 中的符号：



说明

19231

INC_ENCODER offers up/down counter functions for the evaluation of encoders.

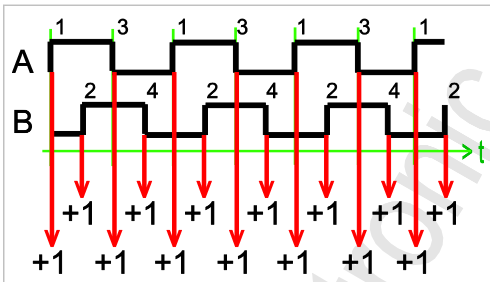
每个通过功能块评估的输入对通过两个邻近的频率输入端形成。

限制频率 = 5 kHz

最大待连接元件数量： 4 个编码器（ExtendedController： 最多 8 个编码器）

设定预设值：

1. 在 PRESET_VALUE 中输入值
2. 针对一个周期将 PRESET 设为 TRUE
3. 将 PRESET 复位至 FALSE

	只要 INIT=FALSE 且 PRESET=FALSE，功能块在输入对每个边沿进行计数（与编码器的 4 倍分辨率一致）。 电流计数值位于输出端 COUNTER。

输出端 UP 和 DOWN 表示计数器当前的计数方向：

- 如果计数器在之前程序周期中按相应的方向计数（即便过溢），则输出端 = TRUE。
- 如果计数器停止，后续程序周期中的方向输出端亦复位。
- 过溢时的不同：
  计数器停止时，方向输出端 = TRUE，直至进行反向计数。

> ❗ 切**勿**搭配使用一个输入端上的该功能块与以下功能块之一！
> • **FAST_COUNT** (→ 页 171)
> • **FREQUENCY** (→ 页 173)
> • **FREQUENCY_PERIOD** (→ 页 175)
> • **INC_ENCODER** (→ 页 178)
> • **PERIOD** (→ 页 183)
> • **PERIOD_RATIO** (→ 页 185)
> • **PHASE** (→ 页 187)
> • **SET_INTERRUPT_I** (→ 页 152)

输入端参数

529

参数	数据类型	说明
INIT	BOOL	TRUE（仅针对 1 个周期）： 功能块初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	输入通道对 x 编号 0 = 通道对 0 = 输入端 I00 + I01 ... 7 = 通道对 7 = 输入端 I14 + I15 （0...x，值视装置而定→ 技术 资料） 0 = 通道对 0 = 输入端 I00_E + I01_E ... 7 = 通道对 7 = 输入端 I14_E + I15_E
PRESET_VALUE	DINT	计数器初始值
PRESET	BOOL	FALSE ⇨ TRUE (edge): PRESET_VALUE 加载至 COUNTER TRUE： 计数器忽略输入脉冲 FALSE： 计数器计算输入脉冲

输出端参数

530

参数	数据类型	说明
COUNTER	DINT	当前计数值
UP	BOOL	TRUE： 计数器在最后一个周期中向上计数 FALSE： 计数器在最后一个周期中未向上计数
DOWN	BOOL	TRUE： 计数器在最后一个周期中向下计数 FALSE： 计数器在最后一个周期中未向下计数

PERIOD

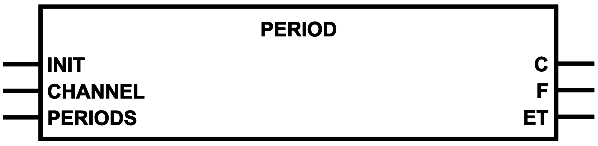
15930

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

20662
20677

PERIOD 测量所示通道（针对所有输入端）上的频率和以 [µs] 为单位的周期（周期时间）。 最大输入频率 → 数据表

该 FB 测量所选 CHANNEL 信号的频率和周期时间。 若要计算，则评估所有上升沿并利用所示 PERIODS 的数字确定平均值。

如果是低频率，则在使用以下对象时会出现误差： FREQUENCY (→ 页 173). 为避免这种情况，可使用 PERIOD。 周期时间直接以 [µs] 为单位显示。

最大测量范围为 10s。

14888

注意

如果频率较高（高于 IFM 保证的水平），则会出现以下问题：

- 输出端的开启和关闭时间将变得更重要。
- 组件过度发热。

上述影响取决于每种情况下使用的组件。
这些潜在影响无法准确预测。

22690

以下适用于装置的标准侧：
该功能块可在同一输出端跟以下功能块一起使用： INC_ENCODER (→ 页 178).

22691

► 同时初始化并联运行的功能块。

切**勿**搭配使用一个输入端上的该功能块与以下功能块之一！
- FAST_COUNT (→ 页 171)
- FREQUENCY (→ 页 173)
- FREQUENCY_PERIOD (→ 页 175)
- INC_ENCODER_HR (→ 页 181)
- PERIOD_RATIO (→ 页 185)
- PHASE (→ 页 187)

输入端参数

19874

参数	数据类型	说明
INIT	BOOL	FALSE ⇨ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	快速输入通道编号 输出端 I00...I11 对应 0...11 就功能块而言，xxx_E（如有）将适用： 输出端 I00_E...I15_E 对应 0...15
PERIODS	BYTE	待平均周期数量 (1...16) 0 : 输出端 C 和 F 未更新 > 16 : 限制在 16 以内

输出端参数

375

参数	数据类型	说明
C	DWORD	已检查周期的周期时间 [µs] 允许 = 200...10 000 000 = 0xC8...0x989680 (= 10s)
F	REAL	输入信号的频率，以 [Hz] 为单位
ET	TIME	自输入端最后一个上升边沿以来消耗的时间 （可用于非常慢的信号）

PERIOD_RATIO

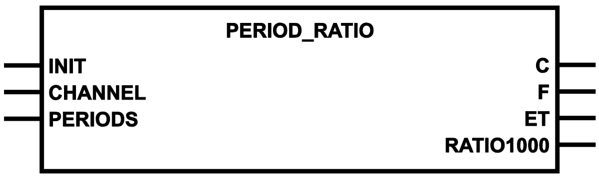
15932

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

ⓘ就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

20665
20678

PERIOD_RATIO 测量在所示周期内所示通道（针对所有输入端）上的频率和以 [µs] 为单位的周期（周期时间）。 此外，传号周期比以 [‰] 为单位显示。 最大输入频率 → 数据表

该 FB 测量所选 CHANNEL 信号的频率和周期时间。 若要计算，则评估所有上升沿并利用所示 PERIODS 的数字确定平均值。 此外，传号周期比以 [‰] 为单位显示。
例如： 如果是 25 ms 高电平和 75 ms 低电平的信号比，则值 ATIO1000 为 250 ‰。

如果是低频率，则在使用以下对象时会出现误差： **FREQUENCY** (→ 页 173). 为避免这种情况，可使用 PERIOD_RATIO。 周期时间直接以 [µs] 为单位显示。

最大测量范围为 10s。

14888

> **ⓘ注意**
>
> 如果频率较高（高于 **IFM** 保证的水平），则会出现以下问题：
>
> - 输出端的开启和关闭时间将变得更重要。
> - 组件过度发热。
>
> 上述影响取决于每种情况下使用的组件。
> 这些潜在影响无法准确预测。

22690

> ⓘ 以下适用于装置的标准侧：
> 该功能块可在同一输出端跟以下功能块一起使用： **INC_ENCODER** (→ 页 178).

22691

> ⓘ ► 同时初始化并联运行的功能块。

切**勿**搭配使用一个输入端上的该功能块与以下功能块之一！
- **FAST_COUNT** (→ 页 171)
- **FREQUENCY** (→ 页 173)
- **FREQUENCY_PERIOD** (→ 页 175)
- **INC_ENCODER_HR** (→ 页 181)
- **PERIOD** (→ 页 183)
- **PHASE** (→ 页 187)

输入端参数

19873

参数	数据类型	说明
INIT	BOOL	FALSE ⇨ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	快速输入通道编号 输出端 I00...I11 对应 0...11 就功能块而言，xxx_E（如有）将适用： 输出端 I00_E...I15_E 对应 0...15
PERIODS	BYTE	待平均周期数量 (1...16) 0 : 输出端 C 和 F 未更新 > 16 : 限制在 16 以内

输出端参数

369

参数	数据类型	说明
C	DWORD	已检查周期的周期时间 [µs] 允许 = 200...10 000 000 = 0xC8...0x989680 (= 10s)
F	REAL	输入信号的频率，以 [Hz] 为单位
ET	TIME	自输入端最后状态更改以来消耗的时间（可用于信号非常缓慢的情况下）
ATIO1000	WORD	传号空号比，以 [‰] 为单位 允许 = 1...999 = 0x1...0x3E7 前提： • 针对间隔测量 • 脉冲持续时间 ≥ 100 µs • 频率 < 5 kHz

PHASE

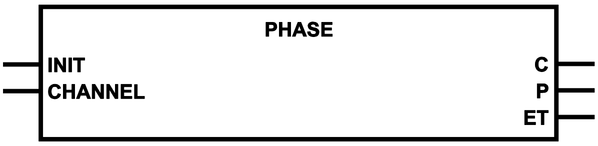
15934

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

ⓘ就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

20668
20679

PHASE 读取包含快速输入端的通道对并对比信号的相位。 最大输入频率 → 数据表

该 FB 对比包含快速输入端的通道对，以便可评估两个信号针对彼此而言的相位。 即便是在以秒为单位的范围内（最长 10s），亦可评估周期。

14888

> **注意**
>
> 如果频率较高（高于 IFM 保证的水平），则会出现以下问题：
>
> - 输出端的开启和关闭时间将变得更重要。
> - 组件过度发热。
>
> 上述影响取决于每种情况下使用的组件。
> 这些潜在影响无法准确预测。

22690

> ⓘ 以下适用于装置的标准侧：
> 该功能块可在同一输出端跟以下功能块一起使用： **INC_ENCODER** (→ 页 178).

22691

> ► 同时初始化并联运行的功能块。

切**勿**搭配使用一个输入端上的该功能块与以下功能块之一！

- FAST_COUNT (→ 页 171)
- FREQUENCY (→ 页 173)
- FREQUENCY_PERIOD (→ 页 175)
- INC_ENCODER_HR (→ 页 181)
- PERIOD (→ 页 183)
- PERIOD_RATIO (→ 页 185)

输入端参数

19875

参数	数据类型	说明
INIT	BOOL	TRUE（仅针对 1 个周期）： 功能块和接口初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	输入通道对编号 0 = 通道对 0 = 输入端 I00 + I01 ... 5 = 通道对 5 = 输入端 I10 + I11 就功能块而言，xxx_E（如有）将适用： 0 = 通道对 0 = 输入端 I00_E + I01_E ... 7 = 通道对 7 = 输入端 I14_E + I15_E

输出端参数

363

参数	数据类型	说明
C	DWORD	第一个输出端通道对信号周期，以 [μs] 为单位
P	INT	相移角 有效测量： 1...358 °
ET	TIME	自通道对第二个脉冲输入端最后一个上升沿以来消耗的时间

5.2.11 功能元件： 一般输出端功能

内容

10462

就此装置而言，您可设定某些或全部输出端的模式。 您可在此了解它的一些功能元件。

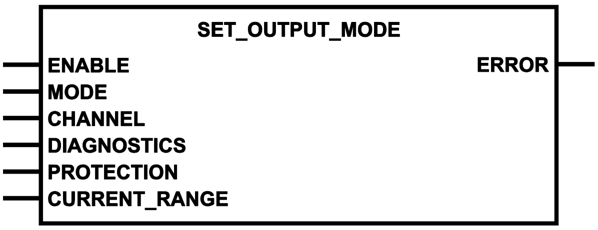
SET_OUTPUT_MODE

15937

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

12094

SET_OUTPUT_MODE 设定所选输出通道的工作模式。

允许的工作模式（→ 数据表）：

MODE	配置值	
	十六进制	十进制
---	---	---
OUT_DIGITAL_H（正极）	0001	1
OUT_DIGITAL_L（负极）	0002	2

CURRENT_RANGE	配置值	
	十六进制	十进制
无电流测量 针对 MODE = 0	00	0
电流测量 2 A (3 A) 针对 MODE = 1 或 2	01	1
电流测量 4 A 针对 MODE = 1 或 2	02	2

❗运行期间不得更改工作模式。

15672

❗注意

运行期间，ACTUAL_CURRENT 的测量范围是否会在功能块 OUTPUT_BRIDGE 的位置加以更改（至 4 A）？

► 就两个相关的输出端而言，在初始化阶段，先调用功能块 SET_OUTPUT_MODE，**再**调用功能块 OUTPUT_BRIDGE！
CURRENT_RANGE = 2 (for 4 A)

输入端参数

12096

参数	数据类型	说明
ENABLE	BOOL	FALSE ⇨ TRUE (edge): 初始化功能块（仅 1 个周期） > 读取功能块输入 TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
MODE	WORD	输出通道 CHANNE 工作模式： 1 = 0x0001 OUT_DIGITAL_H 2 = 0x0002 OUT_DIGITAL_L
CHANNEL	BYTE	输出通道编号 输出端 Q00...Q15 对应 0...15 就功能块而言，xxx_E（如有）将适用： 输出端 Q00_E...Q31_E 对应 0...31
DIAGNOSTICS	BOOL	TRUE: 包含诊断功能的通道 仅适用于 OUT_DIGITAL_H • 包含电流测量的输出端断线： 如果输出端开启，且 如果电流 < 25 mA 且持续 ≥ 66 ms • 不含电流测量的输出端断线： 如果输出端关闭，且 如果输出电压 > 22 % VBBx 且持续 ≥ 100 ms • 过载，包含电流测量： 如果电流 > 测量范围的 112.5 %且持续 ≥ 66 ms • 短路，不含电流测量： 如果 VBBx > 7.3 V，且 如果输出电压 < 88 % VBBx 且持续 ≥ 66 ms 就电流测量而言， 过滤设定 Qxx_FILTER 影响诊断时间 FALSE: 不含诊断功能的通道
PROTECTION	BOOL	TRUE: 过载保护 仅针对 OUT_DIGITAL_H 以及包含电流测量的输出端 如果检测到过载或短路，输出端则关闭 1 s，而后再次开启。 FALSE: function element is not executed
CURRENT_RANGE	BYTE	输出通道 CHANNEL 电流测量：

参数	数据类型	说明		
		0 = 0x00	OUT_CURRENT_RANGE_NONE 仅针对输出端... 不含电流测量 或针对 OUT_DIGITAL_L	（关闭）
		1 = 0x01	OUT_CURRENT_RANGE1 仅针对 OUT_DIGITAL_H	2 A/3 A
		2 = 0x02	OUT_CURRENT_RANGE2 仅针对 OUT_DIGITAL_H	4 A

输出端参数

12102

参数	数据类型	说明
ERROR	DWORD	来自该功能块调用的错误代码 → **错误代码** (→ 页 373)（可能的消息 → 下表）

ERROR 的可能结果（n= 任何所需值）：

32 位错误代码包含四个 8 位值 (DWORD)。

第 4 个字节	第 3 个字节	第 2 个字节	第 1 个字节
错误类别	应用程序特定错误代码	错误来源	错误原因

值 [hex]	说明
00 00 00 00	无错误
01 00 00 F8	错误参数 ➪ 一般错误

5.2.12 功能元件： PWM 功能

内容

13758

您可在此了解可供您运行包含脉冲宽度调制 (PWM) 的输出端的 **IFM** 功能块。

OUTPUT_BRIDGE

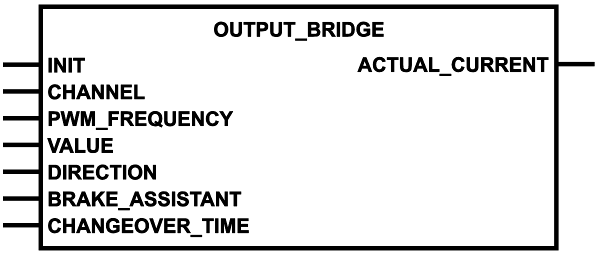
2198

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

ⓘ就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

19298

OUTPUT_BRIDGE 控制 PWM 通道上的 H 桥。

输出端可通过该 FB 轻松地用作 H 桥。 为此，两个连续输出通道通过负极开关驱动器融合至一个桥。如果 DIRECTION = FALSE，就第一个输出端而言，正极开关驱动器通过 PWM 信号触发，第二个输出端的负极开关驱动器切换。

> **注意**
>
> 使用 H 桥时，电流控制不受支持。
>
> 在 PWM 模式下运行的输出端不支持任何诊断功能，且不设定 ERROR 标志。 这是由输出端的结构决定的。
>
> 不得利用配置为 B(L) 的输出端调用该 FB。
>
> 定义为 PWM 输出端的输出端后续不可再作为二进制输出端。
>
> 在该模式下，过载保护功能未启用！
> 功能 OUTPUT_BRIDGE 将复位过载保护功能。
>
> ► 仅在 INIT=FALSE 时在运行期间更改参数。 当前 PWM 周期结束后，方会采用新的参数。
>
> 运行期间更改 PWM 频率：
> 仅在范围 40...2 000 Hz 之间时允许。
>
> ⓘ 如果 VALUE = 0，输出端则不会完全禁用。 原则上，输出端将在 PWM 计时器滴答期间（一般为 50 µs 左右）启用。
>
> ► 必须在每个周期中调用 FB。

分配用作 H 桥的输出通道→ 数据表。

15672

注意

运行期间，ACTUAL_CURRENT 的测量范围是否会在功能块 OUTPUT_BRIDGE 的位置加以更改（至 4 A）？

▶ 就两个相关的输出端而言，在初始化阶段，先调用功能块 SET_OUTPUT_MODE，**再**调用功能块 OUTPUT_BRIDGE！
CURRENT_RANGE = 2 (for 4 A)

ACTUAL_CURRENT 显示 H 桥的电流，以 [mA] 为单位。

测量方法：

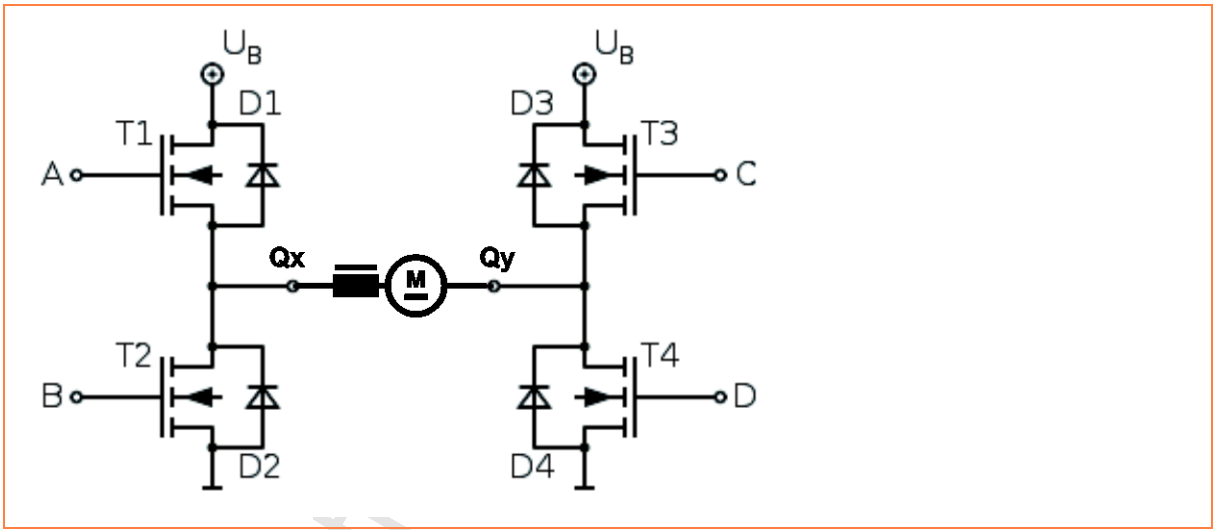
- PWM 频率 < 100 Hz： 取一个 PWM 周期的平均值
- PWM 频率 ≥ 100 Hz： 取一个（PWM 频率/50）PWM 周期的平均值（向下取整）

H 桥的原理

9990
16411

您可在此了解 H 桥如何通过 **IFM** 控制器位置的 PWM 输出端运行。

包含 OWM 控制的 H 桥**基本电路**：



例如，T1 和 T2 合为输出端 Qx。

T3 和 T4 亦互为一体，适用于 Qy。

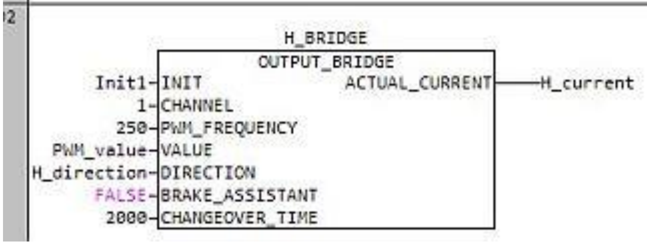
因此，您仅需两个插脚来连接 DC 发动机。

程序示例：

```
VAR
    Init1: BOOL:=TRUE;
    CycleTime:DWORD;
    MaxCycleTime:DWORD;
    ResetMax:BOOL;

    DownloadID: CAN1_DOWNLOADID;

(*****************************************************************************
CHANNEL = 1:    Motor between OUT01 (Pin17) and OUT03(Pin15)
CHANNEL = 2:    Motor between OUT09 (Pin03) and OUT11(Pin05)
*****************************************************************************)
    H_BRIDGE: OUTPUT_BRIDGE;
    PWM_value: WORD := 100;            (* current PWM value - VALUE = 0...1000 *)
    H_direction: BOOL;                 (* TRUE = counter clockwise; FALSE = clockwise *)
    H_current: WORD;                   (* output current in mA *)
    changeover_time: WORD := 500;      (* Space time [ms] during which the motor is not triggered
                                          (> 10 ms) in the case of a change of the rotational direction. *)
END_VAR
```



输入端参数

2204

参数	数据类型	说明
INIT	BOOL	TRUE（仅针对 1 个周期）： 功能块初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	输出对名称： 1 = Q01 + Q03 的桥 1 2 = Q09 + Q11 的桥 2 就功能块而言，xxx_E（如有）将适用： 1 = Q01_E + Q03_E 的桥 1 2 = Q09_E + Q11_E 的桥 2
PWM_FREQUENCY	WORD	输出端负载 PWM 频率 [Hz] > 功能块限于值 20...2 000 = 0x0014...0x07D0 运行期间更改 PWM 频率： 仅在范围 40...2 000 Hz 之间时允许。
VALUE	WORD	PWM 值（传号空号比），以 [‰] 为单位 允许 = 0...1 000 = 0x0000...0x03E8 值 > 1 000 视作 = 1 000

参数	数据类型	说明
DIRECTION	BOOL	发动机旋转方向： TRUE: 逆时针 (ccw)： 桥 1： 电流 Q01(_E) ⇦ Q03(_E) 桥 2： 电流 Q09(_E) ⇦ Q11(_E) FALSE: 顺时针 (cw)： 桥 1： 电流 Q01(_E) ⇨ Q03(_E) 桥 2： 电流 Q09(_E) ⇨ Q11(_E)
BRAKE_ASSISTANT	BOOL	TRUE: 更改旋转方向时： 功能块将两个输出端均切换至接地， 以对发动机进行制动， 但前提是 CHANGEOVER_TIME 处于运行状态。 FALSE: function element is not executed
CHANGEOVER_TIME	WORD	旋转方向更改时发动机未触发的空间时间，以 [ms] 为单位 （≥ 周期时间，至少 10 ms） 值 < 10 ms 视作 = 10 ms

输出端参数

2205

参数	数据类型	说明
ACTUAL_CURRENT	WORD	输出电流，以 [mA] 单位

OUTPUT_CURRENT

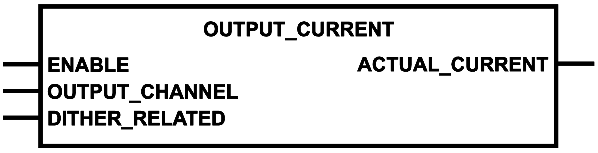
382

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

385

OUTPUT_CURRENT 处理协同有效 PWM 通道的电流测量。

如果输出端用作 PWM 输出端或负极开关，则该 FB 提供电流输出。 装置进行电流测量，即无需外部测量电阻器。

输入端参数

17894

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
OUTPUT_CHANNEL	BYTE	电流控制输出通道 (0...15) 编号 输出端 Q00...Q15 对应 0...15 就功能块而言，xxx_E（如有）将适用： 输出端 Q00_E...Q15_E 对应 0...15
DITHER_RELATED	BOOL	通过以下方式确定电流的平均值... TRUE： 一个抖动周期 FALSE： 一个 PWM 周期

输出端参数

387

参数	数据类型	说明
ACTUAL_CURRENT	WORD	输出电流，以 [mA] 单位

OUTPUT_CURRENT_CONTROL

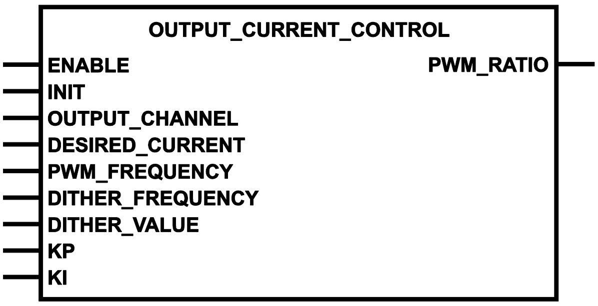
2196

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

2200

OUTPUT_CURRENT_CONTROL 作为 PWM 输出端的电流控制器运行。

设定参数 KI 和 KP 表示控制器的 I 组件和 P 组件。 建议设定 KI=50 和 KP=50 作为初始值，以确定控制器的最佳设定。 根据所需的控制器状况，值可逐步增加（控制器更强/更快）或减少（控制器更弱/更慢）。

在预设值 DESIRED_CURRENT=0 时，输出端在 100 ms 左右内控制到 0 mA，设定参数忽略。

根据使用的控制器硬件，需注意不同的示教性能。

如果在运行期间更改参数，则会出现以下情况：

- 控制器可能完全跳过或
- 控制器需要更长的时间来调整给定值。

▶ 因此验证测量电流，必要时重启控制器。

注意

► 定义参数 DITHER_VALUE 时，确保循环控制工作范围中产生的 PWM 比率保持在 0...1000 % 之间：
• PWM 比率 + DITHER_VALUE < 1000 ‰ 且
• PWM 比率 - DITHER_VALUE > 0 ‰。
若超出该允许范围，则 ITHER_VALUE 暂时内部降低至最大可能值，这样 PWM 比率平均值便可符合所需的值。

> 抖动频率启用时，PWM_FREQUENCY、DITHER_VALUE 和 DITHER_FREQUENCY 的更改仅在当前抖动频率周期结束后生效。

► 仅在 INIT=FALSE 时在运行期间更改参数。 当前 PWM 周期结束后，方会采用新的参数。

► 运行期间更改 PWM 频率：
仅在范围 40...2 000 Hz 之间时允许。

> 如果因为 PWM 比率已经为 100 % 而无法达到参数 DESIRED_CURRENT 所示的电流，则通过系统变量 ERROR_CONTROL_Qx 显示该情况。

> 如果 KI = 0，则无循环控制。

> 如果在循环控制期间，出现 PWM_RATIO = 0，则输出端不会完全禁用。 原则上，输出端将在 PWM 计时器滴答期间（一般为 50 µs 左右）启用。

► 每个 PLC 周期，FB 仅初始化一次 (INIT = TRUE)。

► 不得利用配置为 B(L) 的输出端调用该 FB。

► 定义为 PWM 输出端的输出端后续不可再作为二进制输出端。

> 如果在开启条件下流动电流超过测量范围，则不可再进行控制，因为 AD 转换器在测量范围末端，所以会提供错误的值（最大值）。

输入端参数

2201

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
INIT	BOOL	TRUE（仅针对 1 个周期）： 功能块初始化 FALSE: 在程序进一步处理期间
OUTPUT_CHANNEL	BYTE	电流控制输出通道 (0...15) 编号 输出端 Q00...Q15 对应 0...15 就功能块而言，xxx_E（如有）将适用： 输出端 Q00_E...Q15_E 对应 0...15
DESIRED_CURRENT	WORD	输出端所需电流值，以 [mA] 为单位 允许 = 0...2 000 / 0...4 000 （取决于输出端和配置）
PWM_FREQUENCY	WORD	输出端负载 PWM 频率 [Hz] > 功能块限于值 20...2 000 = 0x0014...0x07D0 运行期间更改 PWM 频率： 仅在范围 40...2 000 Hz 之间时允许。
		抖动频率，以 [Hz] 为单位 值范围 = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY 必须为偶数！ FB 可将所有其他至增加到下一个匹配值。
DITHER_VALUE	WORD	抖动峰间值，以 [‰] 为单位 允许值 = 0...1 000 = 0000...03E8
KP	BYTE	输出信号的比例分量
KI	BYTE	输出信号的积分分量 如果 KI = 0，则无规则

输出端参数

2202

参数	数据类型	说明
PWM_RATIO	WORD	针对监控目的： 显示 PWM 脉冲比 0...999 %

PWM1000

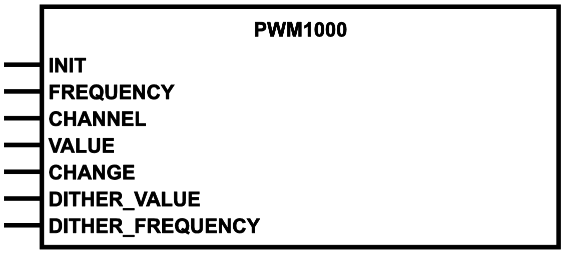
326

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

ⓘ就 ExtendedControllers 的扩展侧而言，FB 名称以 '_E' 结尾。（不适用于 CR0133）

CODESYS 中的符号：



说明

2311

PWM1000 处理 PWM 输出端的初始化和参数设定。

FB 有助于简单使用装置中的 PWM 功能。 就每个通道而言，自身 PWM 频率、传号周期比和抖动频率均可设定。

PWM 频率 FREQUENCY 可直接以 [Hz] 为单位显示，传号周期比 VALUE 以 1 ‰ 为单位增减。

ⓘ 如果 VALUE = 0，输出端则不会完全禁用。 原则上，输出端将在 PWM 计时器滴答期间（一般为 50 µs 左右）启用。

► 定义参数 DITHER_VALUE 时，确保循环控制工作范围中产生的 PWM 比率保持在 0...1000 % 之间：
  • PWM 比率 + DITHER_VALUE < 1000 ‰ 且
  • PWM 比率 - DITHER_VALUE > 0 ‰。
  若超出该允许范围，则 ITHER_VALUE 暂时内部降低至最大可能值，这样 PWM 比率平均值便可符合所需的值。

► 永久启用功能块！

► 不得利用配置为 B(L) 的输出端调用该 FB。

ⓘ注意

运行期间，不可对定义为 PWM 功能的通道执行功能更改。 PWM 功能保持不变，直至在控制器上执行硬件复位 ⇨ 再次断电和通电。

就高 PWM 频率而言，设定比率和输出端比率之间可能因系统而出现差异。

► 仅在 INIT=FALSE 时在运行期间更改参数。 当前 PWM 周期结束后，方会采用新的参数。

► 运行期间更改 PWM 频率：
  仅在范围 40...2 000 Hz 之间时允许。

运行时期间的变化：

始终在输入端 CHANGE 设为 TRUE 时，FB 采用以下值，且需满足相应条件...

- FREQUENCY，当前 PWM 周期结束后
- VALUE，当前 PWM 周期结束后
- DITHER_VALUE，当前抖动频率周期结束后
- DITHER_FREQUENCY，当前抖动频率周期结束后

输入端参数

2312

参数	数据类型	说明
INIT	BOOL	TRUE（仅 1 个周期）： 功能块初始化 采用 FREQUENCY 的新值 FALSE: 在程序进一步处理期间
FREQUENCY	WORD	PWM 频率，以 [Hz] 为单位 > 功能块限于值 20...2 000 = 0x0014...0x07D0 运行期间更改 PWM 频率： 仅在范围 40...2 000 Hz 之间时允许。
CHANNEL	BYTE	PWM 输出通道编号 输出端 Q00...Q15 对应 0...15 就功能块而言，xxx_E（如有）将适用： 输出端 Q00_E...Q15_E 对应 0...15
VALUE	WORD	PWM 值（传号空号比），以 [‰] 为单位 允许 = 0...1 000 = 0x0000...0x03E8 值 > 1 000 视作 = 1 000
CHANGE	BOOL	TRUE: 采用以下新值： • FREQUENCY： 当前 PWM 周期结束后 • VALUE： 当前 PWM 周期结束后 • DITHER_VALUE： 当前抖动周期结束后 • DITHER_FREQUENCY： 当前抖动周期结束后 FALSE： 更改的 PWM 值对输出端无影响
DITHER_VALUE	WORD	抖动峰间值，以 [‰] 为单位 允许值 = 0...1 000 = 0000...03E8
		抖动频率，以 [Hz] 为单位 值范围 = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY 必须为偶数！ FB 可将所有其他至增加到下一个匹配值。

5.2.13 功能元件： 液压控制

内容

13760

库 ifm_HYDRAULIC_32bit_Vxxyyzz.Lib 包含以下功能块：

CONTROL_OCC (→ 页 206)	OCC = Output Current Control 将输入值 [WORD] 换算至所示电流范围。
JOYSTICK_0 (→ 页 209)	将操纵杆的信号 [INT] 换算成明确定义的特性曲线，标准化为 0... 1000
JOYSTICK_1 (→ 页 213)	换算操纵杆 D 的信号 [INT]，标准化为 0... 1000
JOYSTICK_2 (→ 页 217)	将操纵杆的信号 [INT] 换算成可配置的特性曲线； 自由选择标准化
NORM_HYDRAULIC (→ 页 220)	将定义的限值范围内的值 [DINT] 正常化为新的限值范围内的值

需要库 UTIL.Lib（CODESYS 程序包）中的以下功能块：
• RAMP_INT
• CHARCURVE
这些功能块可自动调用并用过液压库的功能块配置。

The following function blocks are needed from the library:ifm_CR0033_Vxxyyzz.LIB

OUTPUT_CURRENT (→ 页 199)	测量输出通道上的电流（平均，通过抖动周期）
OUTPUT_CURRENT_CONTROL (→ 页 200)	PWMi 输出通道的电流控制器

这些功能块可自动调用并用过液压库的功能块配置。

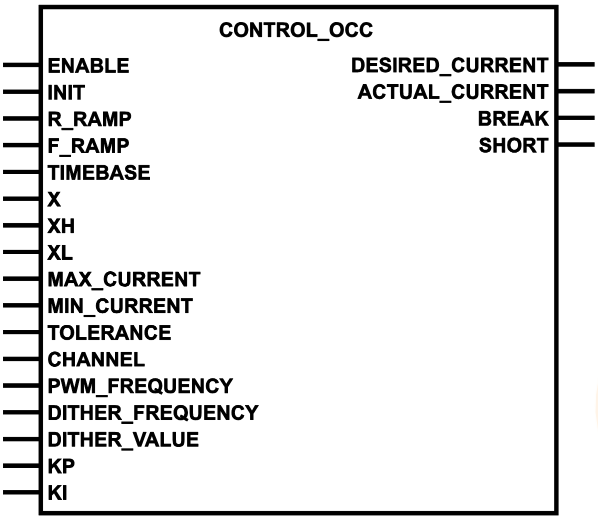
CONTROL_OCC

2735

单元类型 = 功能块 (FB)

元件类型 `ifm_HYDRAULIC_32bit_Vxxyyzz.Lib`

CODESYS 中的符号：



说明

2737

CONTROL_OCC 将输入值 X 换算至特定电流范围。

在每个 PLC 周期中，FB 的每个实例调用一次。

该功能块使用库中的以下功能块：`ifm_CR0033_Vxxyyzz.LIB`

- OUTPUT_CURRENT_CONTROL (→ 页 200)
- OUTPUT_CURRENT (→ 页 199)

控制器根据 PWM 信号的周期控制。

参数设定 KI 和 KP 表示控制器的**积分**和**比例**分量。 建议设定 KI=50 和 KP=50 作为初始值，以确定控制器的最佳设定。

► 增加 KI 和/或 KP 的值： ➩ 控制器变得更敏感/更快
减少 KI 和/或 KP 的值： ➩ 控制器变得不太敏感/更慢

> 在输出端 DESIRED_CURRENT=0 时，输出端**立即**切换至 0 mA，**不会**根据设定的参数调整到 0 mA。

控制器针对电源电压的电压下降有快速补偿机制。 除了控制器的状况之外，基于电压下降，PWM 比率将提高，以尽快达到所需的值。

ⓘ CONTROL_OCC 的输入值 X 应通过 JOYSTICK FB 的输出提供。

输入端参数

2739

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
INIT	BOOL	FALSE ⇨ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
R_RAMP	INT	斜坡上升沿，以 [increments/PLC cycle] 为单位 0 = 无斜坡
F_RAMP	INT	斜坡下降沿，以 [increments/PLC cycle] 为单位 0 = 无斜坡
TIMEBASE	TIME	斜坡上升沿和下降沿参考： t#0s = 上升沿/下降沿，以 [increments/PLC cycle] 为单位 快速控制器的周期时间非常短！ 否则 = 上升沿/下降沿，以 [increments/TIMEBASE] 为单位
X	WORD	输入值
XH	WORD	输入值范围的上限值 [increments]
XL	WORD	输入值范围的下限值 [increments]
MAX_CURRENT	WORD	Max. valve current in [mA]
MIN_CURRENT	WORD	最低阀门电流，以 [mA] 为单位
TOLERANCE	BYTE	最低阀门电流公差，以 [increments] 为单位 超过公差时会跳至 MIN_CURRENT
CHANNEL	BYTE	电流控制输出通道编号 输出端 Q00...Q15 对应 0...15 就功能块而言，xxx_**E**（如有）将适用： 输出端 Q00_E...Q15_E 对应 0...15
PWM_FREQUENCY	WORD	输入端负载 PWM 频率 [Hz]
		抖动频率，以 [Hz] 为单位 值范围 = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY 必须为偶数！ FB 可将所有其他至增加到下一个匹配值。
DITHER_VALUE	BYTE	抖动峰间值，以 [%] 为单位 允许值 = 0...100 = 0x00...0x64
KP	BYTE	输出信号的比例分量
KI	BYTE	输出信号的积分分量

就 KP 而言，KI 适用： 建议初始值 = 50

输出端参数

602

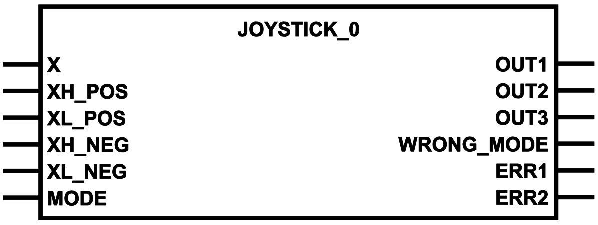
参数	数据类型	说明
DESIRED_CURRENT	WORD	OCC 所需电流值，以 [mA] 为单位 （针对监控目的）
ACTUAL_CURRENT	WORD	输出电流，以 [mA] 单位
BREAK	BOOL	错误： 输出端电缆中断
SHORT	BOOL	错误： 输出端电缆短路

JOYSTICK_0 6250

单元类型 = 功能块 (FB)

元件类型 `ifm_hydraulic_32bit_Vxxyyzz.Lib`

CODESYS 中的符号：

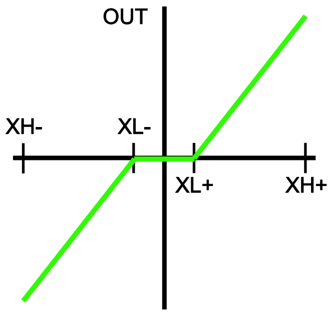
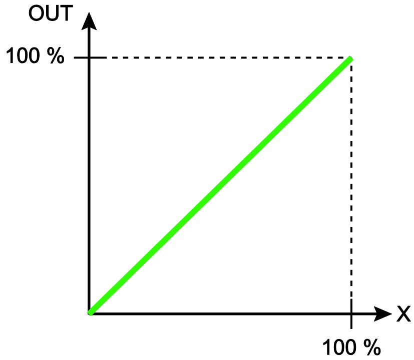


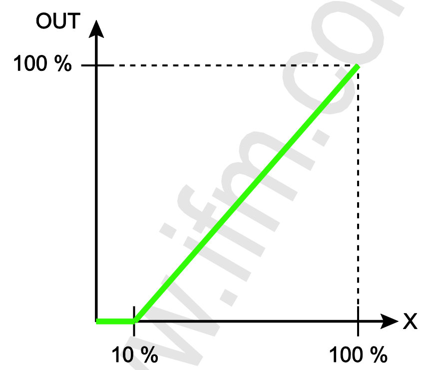
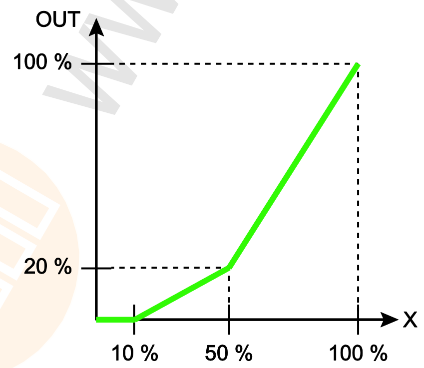
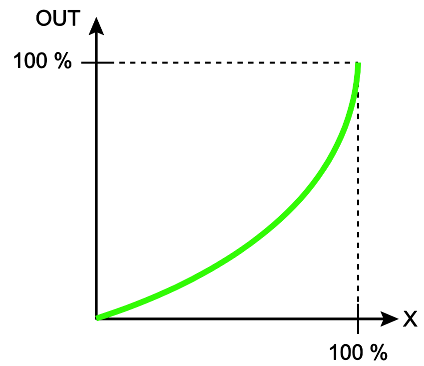
说明 432

JOYSTICK_0 将操纵杆的信号换算成明确定义的特性曲线，标准化为 0...1000。

就此 FB 而言，特性曲线值是指定的（→ 图）：

- 斜坡上升沿 = 5 increments/PLC cycle
  ❗快速控制器的周期时间非常短！
- 斜坡下降沿 = 无斜坡

参数 XL_POS (XL+)、XH_POS (XH+)、XL_NEG (XL-) 和 XH_NEG (XH-) 仅用于评估所需区域内的操纵杆运动。 正区和负区的值会有所不同。 XL_NEG 和 XH_NEG 的值此处为负。	
模式 0： 从范围 XL 到 XH，特性曲线呈直线	

模式 1： 特性曲线呈直线，含死区 值固定为： 死区： 0…10% of 1000 increments	
模式 2： 2 步式线性特性曲线，含死区 值固定为： 死区： 0…10% of 1000 increments 步骤： X = 50 % of 1000 increments Y = 20 % of 1000 increments	
特性曲线模式 3： 曲线上升（直线固定）	

输入端参数

3

参数	数据类型	说明
X	INT	输入值 [increments]
XH_POS	INT	正向最大预设值 [increments] （亦允许负值）
XL_POS	INT	正向最小预设值 [increments] （亦允许负值）
XH_NEG	INT	负向最大预设值 [increments] （亦允许负值）
XL_NEG	INT	负向最小预设值 [increments] （亦允许负值）
MODE	BYTE	模式选择特性曲线： 0 = 线性 (X|OUT = 0|0 ... 1000|1000) 1 = 线性，含死区 (X|OUT = 0|0 ... 100|0 ... 1000|1000) 2 = 2 步式线性，含死区 (X|OUT = 0|0 ... 100|0 ... 500|200 ... 1000|1000) 3 = 曲线上升（直线固定）

输出端参数

6252

参数	数据类型	说明
OUT1	WORD	标准化输出值： 0...1000 增量 如针对左阀门
OUT2	WORD	标准化输出值： 0...1000 增量 如针对右阀门
OUT3	INT	标准化输入值 -1000...0...1000 增量 如针对输入模块上的阀门（如 CR2011 或 CR2031）
WRONG_MODE	BOOL	错误： 无效的模式
ERR1	BYTE	上升沿错误代码 （指来自 `util.lib` 的内部使用功能块 CHARCURVE 和 RAMP_INT） （可能的消息 → 下表）
ERR2	BYTE	下降沿错误代码 （指来自 `util.lib` 的内部使用功能块 CHARCURVE 和 RAMP_INT） （可能的消息 → 下表）

ERR1 和 ERR2 的可能结果：

值 十进制	十六进制	说明
0	00	无错误
1	01	数组错误： 错误序列
2	02	错误： 数组值范围不含输入值 IN。
4	04	错误： 数字 N 对数组无效

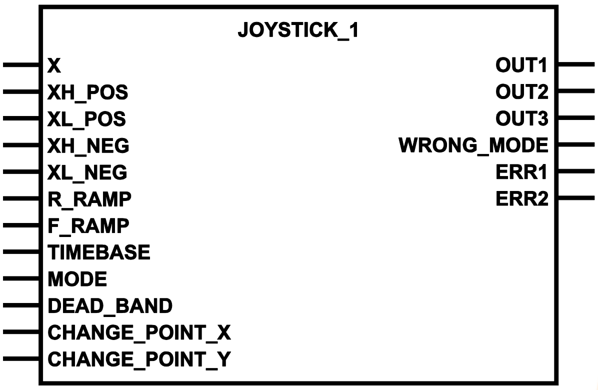
JOYSTICK_1

6255

单元类型 = 功能块 (FB)

元件类型 ifm_hydraulic_32bit_Vxxyyzz.Lib

CODESYS 中的符号：

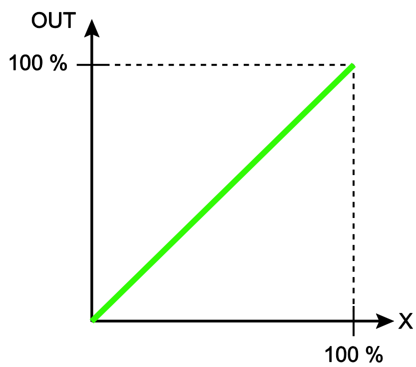
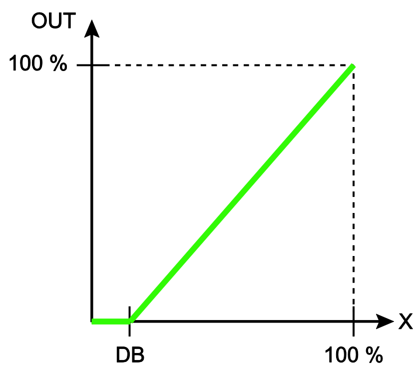


说明

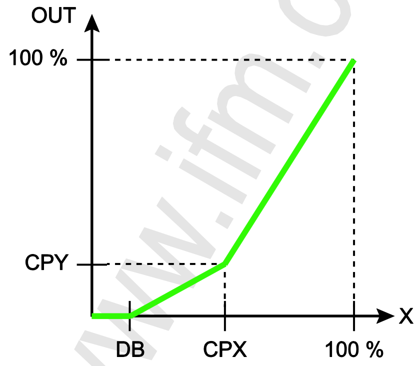
425

JOYSTICK_1 将操纵杆的信号换算成可配置的特性曲线，标准化为 0...1000。

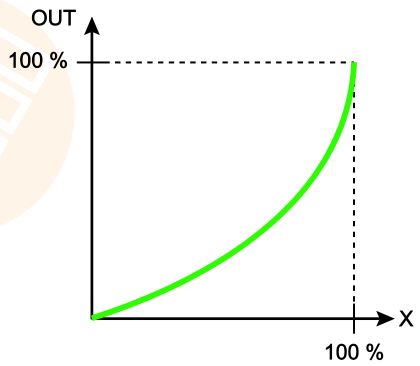
就此 FB 而言，特性曲线值可配置（→ 图）：

模式 0： 线性特性曲线 100 % = 1000 increments	
模式 1： 特性曲线呈直线，含死区 死区 (DB) 的值以 % of 1000 increments 为单位设定。 100 % = 1000 increments DB = Dead_Band	

模式 2：
2 步式线性特性曲线，含死区
值可配置为：
死区：
0…DB，以 % of 1000 increments 为单位
步骤：
X = CPX，以 % of 1000 increments 为单位
Y= CPY，以 % of 1000 increments 为单位
100 % = 1000 increments
DB = Dead_Band
CPX = Change_Point_X
CPY = Change_Point_Y



特性曲线模式 3：
曲线上升（直线固定）



输入端参数

6256

参数	数据类型	说明
X	INT	输入值 [increments]
XH_POS	INT	正向最大预设值 [increments] （亦允许负值）
XL_POS	INT	正向最小预设值 [increments] （亦允许负值）
XH_NEG	INT	负向最大预设值 [increments] （亦允许负值）
XL_NEG	INT	负向最小预设值 [increments] （亦允许负值）
R_RAMP	INT	斜坡上升沿，以 [increments/PLC cycle] 为单位 0 = 无斜坡
F_RAMP	INT	斜坡下降沿，以 [increments/PLC cycle] 为单位 0 = 无斜坡

参数	数据类型	说明
TIMEBASE	TIME	斜坡上升沿和下降沿参考： t#0s = 上升沿/下降沿，以 [increments/PLC cycle] 为单位 快速控制器的周期时间非常短！ 否则 = 上升沿/下降沿，以 [increments/TIMEBASE] 为单位
MODE	BYTE	模式选择特性曲线： 0 = 线性 (X|OUT = 0|0 ... 1000|1000) 1 = 线性，含死区 (X|OUT = 0|0 ... DB|0 ... 1000|1000) 2 = 2 步式线性，含死区 (X|OUT = 0|0 ... DB|0 ... CPX|CPY ... 1000|1000) 3 = 曲线上升（直线固定）
DEAD_BAND	BYTE	可调死区 以 [% of 1000 increments] 为单位
CHANGE_POINT_X	BYTE	针对模式 2： 斜坡步幅，X 值 以 [% of 1000 increments] 为单位
CHANGE_POINT_Y	BYTE	针对模式 2： 斜坡步幅，Y 值 以 [% of 1000 increments] 为单位

输出端参数

6252

参数	数据类型	说明
OUT1	WORD	标准化输出值： 0...1000 增量 如针对左阀门
OUT2	WORD	标准化输出值： 0...1000 增量 如针对右阀门
OUT3	INT	标准化输入值 -1000...0...1000 增量 如针对输入模块上的阀门（如 CR2011 或 CR2031）
WRONG_MODE	BOOL	错误： 无效的模式
ERR1	BYTE	上升沿错误代码 （指来自 `util.lib` 的内部使用功能块 CHARCURVE 和 RAMP_INT） （可能的消息 → 下表）
ERR2	BYTE	下降沿错误代码 （指来自 `util.lib` 的内部使用功能块 CHARCURVE 和 RAMP_INT） （可能的消息 → 下表）

ERR1 和 ERR2 的可能结果：

值 十进制	十六进制	说明
0	00	无错误
1	01	数组错误： 错误序列
2	02	错误： 数组值范围不含输入值 IN。
4	04	错误： 数字 N 对数组无效

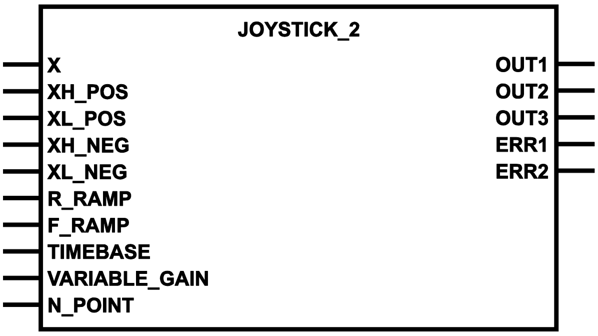
JOYSTICK_2

6258

单元类型 = 功能块 (FB)

元件类型 `ifm_hydraulic_32bit_Vxxyyzz.Lib`

CODESYS 中的符号：

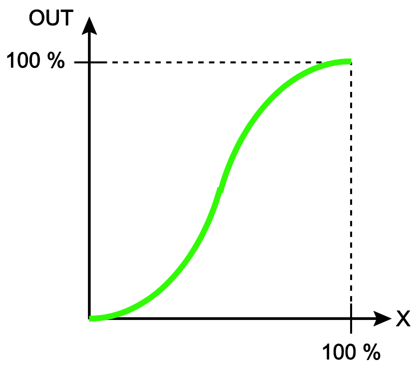


说明

418

JOYSTICK_2 将操纵杆的信号换算成可配置的特性曲线。 自由选择标准化。

就此 FB 而言，特性曲线值可自由配置（→ 图）：

特性曲线可自由配置	

输入端参数

261

参数	数据类型	说明
X	INT	输入值 [increments]
XH_POS	INT	正向最大预设值 [increments] （亦允许负值）
XL_POS	INT	正向最小预设值 [increments] （亦允许负值）
XH_NEG	INT	负向最大预设值 [increments] （亦允许负值）
XL_NEG	INT	负向最小预设值 [increments] （亦允许负值）
R_RAMP	INT	斜坡上升沿，以 [increments/PLC cycle] 为单位 0 = 无斜坡
F_RAMP	INT	斜坡下降沿，以 [increments/PLC cycle] 为单位 0 = 无斜坡
TIMEBASE	TIME	斜坡上升沿和下降沿参考： t#0s = 上升沿/下降沿，以 [increments/PLC cycle] 为单位 快速控制器的周期时间非常短！ 否则 = 上升沿/下降沿，以 [increments/TIMEBASE] 为单位

VARIABLE_GAIN	ARRAY [0..10] OF POINT	描述曲线的数值对
		使用 N_POINT 所示的第一个数值对。n = 2…11
		例如： 9 个数值对声明为变量 VALUES：
		`VALUES: ARRAY [0..10] OF POINT :=`
		`(X:=0,Y:=0),(X:=200,Y:=0), (X:=300,Y:=50),`
		`(X:=400,Y:=100), (X:=700,Y:=500),`
		`(X:=1000,Y:=900), (X:=1100,Y:=950),`
		`(X:=1200,Y:=1000), (X:=1400,Y:=1050);`
		值之间可能有空白。

N_POINT	BYTE	定义曲线特性的点号（VARIABLE_GAIN 数值对） n = 2…11

输出端参数

420

参数	数据类型	说明
OUT1	WORD	标准化输出值： 0...1000 增量 如针对左阀门
OUT2	WORD	标准化输出值： 0...1000 增量 如针对右阀门
OUT3	INT	标准化输入值 -1000...0...1000 增量 如针对输入模块上的阀门（如 CR2011 或 CR2031）
ERR1	BYTE	上升沿错误代码 （指来自 `util.lib` 的内部使用功能块 CHARCURVE 和 RAMP_INT） （可能的消息 → 下表）
ERR2	BYTE	下降沿错误代码 （指来自 `util.lib` 的内部使用功能块 CHARCURVE 和 RAMP_INT） （可能的消息 → 下表）

ERR1 和 ERR2 的可能结果：

值 十进制	十六进制	说明
0	00	无错误
1	01	数组错误： 错误序列
2	02	错误： 数组值范围不含输入值 IN。
4	04	错误： 数字 N 对数组无效

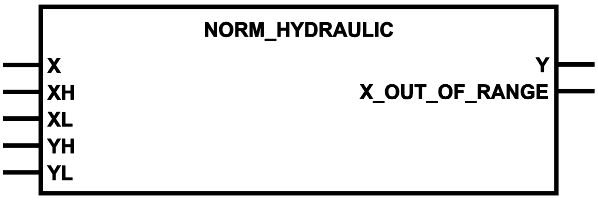
NORM_HYDRAULIC

394

单元类型 = 功能块 (FB)

元件类型 `ifm_hydraulic_32bit_Vxxyyzz.Lib`

CODESYS 中的符号：



说明

397

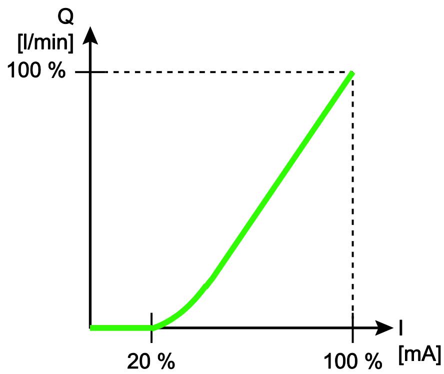
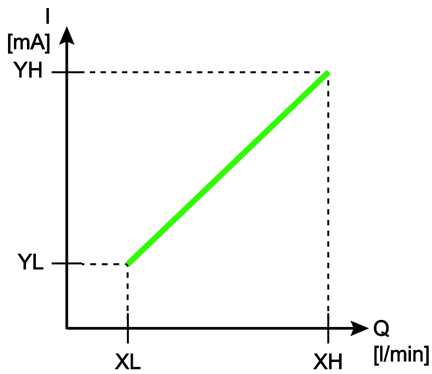
NORM_HYDRAULIC 将包含固定限值的输入值标准化为包含新限值的值。

ⓘ该功能块对应 CODESYS 库 `UTIL.Lib` 中的 NORM_DINT。

功能块将类型 DINT （即 XH 和 XL 限值范围内）的值标准化为 YH 和 YL 限值范围内的输出值。

因为舍入误差，标准化值的偏差可能为 1。 如果按照倒数的形式表示限值（XH/XL 或 YH/YL），则标准化亦倒转。

如果 X 在限值 XL...XH 范围之外，则错误消息将为 X_OUT_OF_RANGE = TRUE。

液压值的一般特性曲线： 在线圈电流达到 20% 之前油流不会开始。 首先，油流不呈直线。	
功能块特性	

输入端参数

398

参数	数据类型	说明
X	DINT	当前输入值
XH	DINT	最大输入值 [increments]
XL	DINT	最小输入值 [increments]
YH	DINT	最大输出值 [increments]，例如： 阀门电流 [mA]/流速 [l/min]
YL	DINT	最小输出值 [increments]，例如： 阀门电流 [mA]，流速 [l/min]

输出端参数

399

参数	数据类型	说明
Y	DINT	输出值
X_OUT_OF_RANGE	BOOL	错误： X 超过 XH 和 XL 限值范围

例如： NORM_HYDRAULIC

400

参数	情况 1：	情况 2：	情况 3：
上限值输入端 XH	100	100	2000
下限值输入端 XL	0	0	0
上限值输出端 YH	2000	0	100
下限值输出端 YL	0	2000	0
非标准化值 X	20	20	20
标准化值 Y	400	1600	1

- 情况 1：
  分辨率相对低的输入端。
  高分辨率输出端。
  1 单位 X 增量导致 20 单位 Y 增量。
- 情况 2：
  分辨率相对低的输入端。
  高分辨率输出端。
  1 单位 X 增量导致 20 单位 Y 增量。
  相比输入信号，输出信号倒转。
- 情况 3：
  高分辨率输入端。
  分辨率相对低的输出端。
  20 单位 X 增量导致 1 单位 Y 增量。

5.2.14 功能元件： 控制器

内容

1634

以下部分详细讲述为 **ecomat*mobile*** 装置软件控制器设定而提供的元件。 该等元件还可作为控制功能开发的基础。

控制器设定规则

1627

就时间常量未知的控制系统而言，封闭控制回路 Ziegler 和 Nickols 设定程序是有利的。

设定控制

1628

开始时，控制系统作为单纯的 P 控制系统运行。 就缓慢的系统而言，微分时间 T_V 设为 0，复位时间 T_N 则非常高（理想情况下为 ∞）。 就快速控制的系统而言，应选择小的 T_N。

之后，增量 KP 增加，直至控制偏差和调整偏差以恒定的振幅（$KP = KP_{critical}$）平稳振荡。 之后达到稳定极限。

而后需确定平稳振荡的时间周期 $T_{critical}$。

如有必要，添加微分分量。

T_N 应比 T_V 大 2...10 倍左右。

KP 应等于 KD。

控制系统理想化设定：

控制器	KP = KD	TN	TV
P	$2.0 \cdot KP_{critical}$	—	—
PI	$2.2 \cdot KP_{critical}$	$0.83 \cdot T_{critical}$	—
PID	$1.7 \cdot KP_{critical}$	$0.50 \cdot T_{critical}$	$0.125 \cdot T_{critical}$

! 就该设定进程而言，需注意产生的振荡不会损坏控制系统。 就灵敏的控制系统而言，

过冲阻尼

1629

若要抑制过冲，则可使用 ***PT1*** (→ 页 230)（低通）。 就此而言，PT1 连接指令抑制预设值 XS，而后再提供给控制器功能。

设定变量 T1 应比控制器 TN 大 4...5 倍左右。

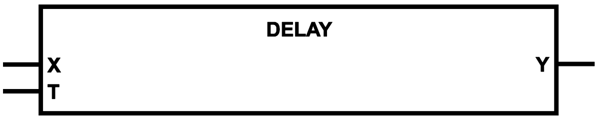
DELAY

585

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

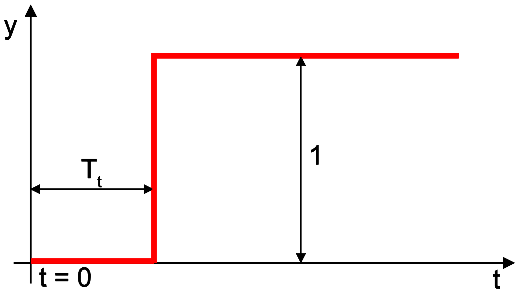
CODESYS 中的符号：



说明

588

DELAY 延迟输入值的输出，延迟时间 T（空载时间元件）。



图： DELAY 的时间特性

空载时间受 PLC 周期持续时间的影响。
空载时间不得超过 100 • PLC 周期时间（内存限制）。
如果设定更长的延迟，则 FB 输出端值的分辨率将更糟糕，这样可能导致短值更改丢失。

❗为确保 FB 正常运行： 必须在每个周期中调用 FB。

输入端参数

2615

参数	数据类型	说明
X	REAL	输入值
T	TIME	延迟时间（空载时间） 允许： 0...100 • 周期时间

输出端参数

2616

参数	数据类型	说明
Y	REAL	输入值，延迟时间 T

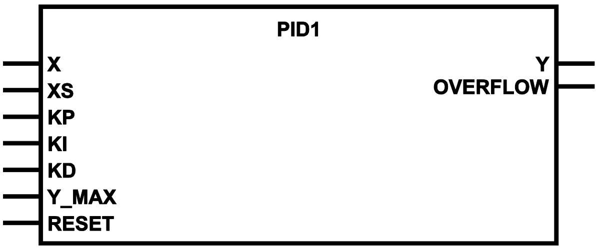
PID1

19235

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

19237

PID1 处理 PID 控制器。

PID 控制器被控变量的改变包含比例 (**P**)、积分 (**I**) 和微分 (**D**) 分量。

当 'I' 部分因为无法修正控制偏差而达到内部限制时，将通过信号表示 OVERFLOW = TRUE。
只要限制有效，OVERFLOW 即保持为 TRUE。

输入端参数

19238

参数	数据类型	说明
X	REAL	输入值
XS	REAL	预设值
KP	REAL	输出信号的比例分量 （仅允许正值）
KI	REAL	输出信号的积分分量 （仅允许正值）
KD	REAL	输出信号的微分分量 （仅允许正值）
Y_MAX	REAL	最大控制值
RESET	BOOL	TRUE： 复位功能元件 FALSE: function element is not executed

输出端参数

19241

参数	数据类型	说明
Y	REAL	输出值
OVERFLOW	BOOL	TRUE: 数据缓冲区溢出 ⇨ 数据丢失！ FALSE: 数据缓冲区无数据丢失

建议的设定

19242

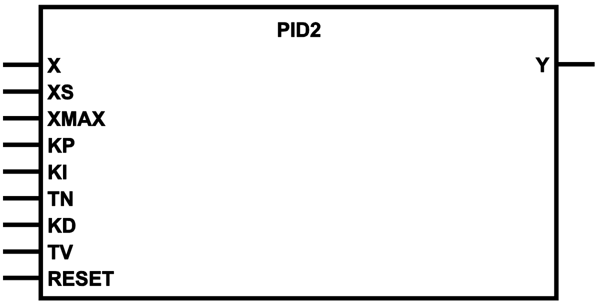
- ▶ 初始值：
  KP = 0
  KD = 0
- ▶ 针对进程调整 KI。
- ▶ 而后逐步修改 KP 和 KI。

PID2 344

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明 6262

PID2 处理 PID 控制器。

PID 控制器被控变量的改变包含比例 (**P**)、积分 (**I**) 和微分 (**D**) 分量。 被控变量先改变一个量，具体取决于输入值的变化速度 (D 分量) 。 在微分酌时间时间 TV 结束后，被控变量返回至与比例分量一致的值，然后根据复位时间 TN 改变。

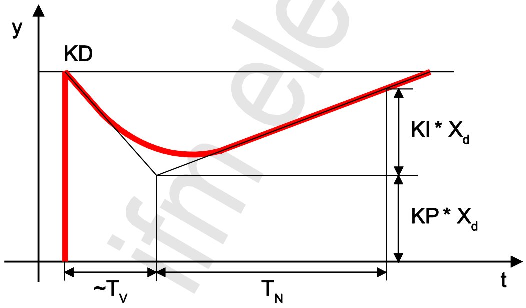
被控变量 Y 已经标准化为 **PWM1000** (→ 页 203).

规则：

- 不允许 KP、KI 和 KD 的负值。
- 如果 TN = 0，则不计算 I 值
- 如果 XS > XMAX，XS 则限制为 XMAX。
- 如果 X > XMAX，Y 则设为 0。
- 如果 X > XS，则提高被控变量。
- 如果 X < XS，则降低被控变量。

参考变量内部添加至被控变量。
Y = Y + 65 536 – (XS / XMAX • 65 536).

被控变量 Y 有以下时间特性。



图： PID 控制器的一般阶跃响应

输入端参数

12963

参数	数据类型	说明
X	WORD	输入值
XS	WORD	预设值
XMAX	WORD	最大预设值
KP	REAL	输出信号的比例分量 （仅允许正值）
KI	REAL	输出信号的积分分量 （仅允许正值）
TN	TIME	积分作用时间（积分分量）
KD	REAL	输出信号的微分分量 （仅允许正值）
TV	TIME	微分作用时间（微分分量）
RESET	BOOL	TRUE： 复位功能元件 FALSE: function element is not executed

输出端参数

349

参数	数据类型	说明
Y	WORD	被控变量 (0...1000 ‰)

建议的设定

350

- 根据系统时间特性选择 TN：
  快速系统 = 小 TN
  慢速系统 = 大 TN
- 逐步缓慢增加 KP，高达仍明确不会产生波动的值。
- 如有必要重新调整 TN。
- 仅在必要时添加**微分**分量：
  选择大约比 TN 小 1/10...1/2 的 TV 值。
  选择跟 KP 相当的 KD 值。

注意，最大控制偏差为 + 127。 为达到良好的控制特性，不得超过该范围，但应尽可能地加以利用。

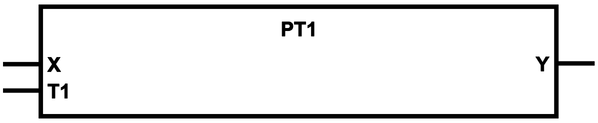
PT1

338

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

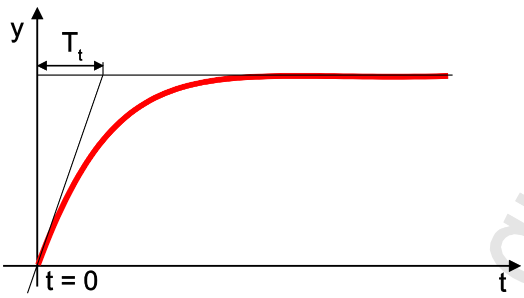
341

PT1 处理包含一级时间延迟的控制系统。

该 FB 为包含时间延迟的比例控制系统。 例如，可用于在使用 PWM FB 时生成斜坡。

❗如果 T1 比 SPS 周期时间短，FB 输出端则会变得不稳定。

低通过滤器输出变量 Y 具备以下时间特性（单位阶跃）：



图： PT1 的时间特性

输入端参数

2618

参数	数据类型	说明
X	DINT	当前输入值
T1	TIME	延迟时间（时间常数）

输出端参数

2619

参数	数据类型	说明
Y	DINT	输出值

5.2.15 功能元件：软件复位

内容

1594

可使用该 FB，利用应用程序中的命令重启控制器。

SOFTRESET

260

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

说明

263

SOFTRESET 使得装置完全重启。

例如，如果要复位节点，则 FB 可搭配 CANopen 使用。 FB SOFTRESET 执行控制器即时重启。 当前周期未执行。

在重启之前，存储保持变量。
错误内存记录重启。

ⓘ进行有效通信时： 必须考虑长时间的复位周期，否则会发送保护信号。

输入端参数

264

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定

5.2.16 功能元件： 时间测量/设定

内容

1601

通过以下 **IFM** 功能块，您可…

- 在应用程序中测量时间并加以评估，
- 必要时更改时间值。

TIMER_READ

236

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：

说明

239

TIMER_READ 读取当前系统时间。

施加电源电压时，装置生成时钟脉冲，并在寄存器中递增计数。 该寄存器可通过 FB 调用读取，且还可用于时间测量等。

> 系统计时器最大可达到 0xFFFF FFFF（对应 49d 17h 2min 47s 295ms），而后会再次从 0 开始计数。

输出端参数

241

参数	数据类型	说明
T	TIME	当前系统时间 [ms]

TIMER_READ_US

657

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：

说明

660

TIMER_READ_US 读取当前系统时间，以 [µs] 为单位

施加电源电压时，装置生成时钟脉冲，并在寄存器中递增计数。 该寄存器可通过 FB 调用读取，且还可用于时间测量等。

> **信息**
>
> 系统计时器的计数值可高达 4 294 967 295 µs，而后会再次从 0 开始计数。
>
> 4 294 967 295 µs = 1h 11min 34s 967ms 295µs

输出端参数

662

参数	数据类型	说明
TIME_US	DWORD	当前系统时间 [µs]

5.2.17 功能元件： 装置温度

内容

2364

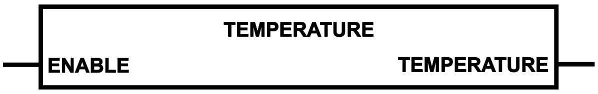
TEMPERATURE

2216

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2365

TEMPERATURE reads the current temperature in the device.

FB 可循环调用，它在其输出端显示当前装置温度 (-40...125 °C)。

输入端参数

2366

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定

输出端参数

2367

参数	数据类型	说明
TEMPERATURE	INT	装置当前内部温度 [°C]

5.2.18 功能元件： 保存、读取和转换内存中的数据

内容

13795

数据备份存储类型

13805

装置提供以下内存类型：

闪存

13803

属性：

• 非易失内存
• 写入相对较慢，且仅可逐块写入
• 在重新写入之前，必须删除内存内容
• 快速读取
• 写入和读取频率有限
• 仅适用于存储大数据量
• 通过 FLASHWRITE 保存数据
• 通过 FLASHREAD 读取数据

FRAM 内存

13802

FRAM 在此表示所有类型的非易失快速内存。

属性：

• 快速写入和读取
• 写入和读取频率无限制
• 可选择任何内存区域
• 通过 FRAMWRITE 保存数据
• 通过 FRAMREAD 读取数据

文件系统

2690

文件系统协调信息在内存中的存储。 文件系统的大小为 128 K 字节。

数据系统的文件名称有限制：
以下对象的最大长度： Controller: CR0n3n, CR7n3n: 15 个字符
所有其他元件的最大长度： 11 个字符

文件系统状况，针对 Controller: CR0n3n, CR7n3n**:**

- 控制器始终尝试写入文件，即便已存在相同的文件名称。 文件可多次保存。 仅使用当前文件。 通过下载（见下文）可防止此类多次归档。
- 单个文件无法覆盖或删除。
- 每次下载（启动项目下载或 RAM 下载）时完全删除文件系统。 而后可写入符号文件或项目文件（CODESYS 中的 FB）等。
- [Reset (Original)]（菜单 [Online] 中的 CODESYS 功能）期间还可删除文件系统。

自动数据备份

内容

14168
2347

ecomat*mobile* 控制器有助于将数据（BOOL、BYTE、WORD、DWORD）保存在内存中且具有非易失性（= 电压故障时保存）。 如果电源电压降低，则备份操作自动启动。 因此有必要将数据定义为 RETAIN 变量 (→ CODESYS)。

声明为 RETAIN 的变量和可通过以下功能块配置为剩余块的标志区域变量存在差异：

***MEMORY*_RETAIN_PARAM** (→ 页 241).

详细信息 → 章节 **变量** (→ 页 83)

自动备份的优势包括，在电压突然下降或电源电压中断时触发存储操作，从而保存当前的数据值（如计数器的值）。

> 如果电源电压 < 8 V，则不再备份保留数据！
> 在此情况下，标记 RETAIN_WARNING = TRUE。

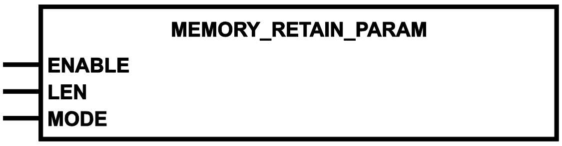
MEMORY_RETAIN_PARAM

2372

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2374

MEMORY_RETAIN_PARAM 确定不同事件的剩余数据特性。 CODESYS 中声明为 VAR_RETAIN 的变量从一开始就具备剩余特性。

剩余数据在不受控制的终止以及控制器正常关闭和开启之后保持其值不变（因为变量声明为 VAR_RETAIN）。 重启后，程序继续按照存储的值运行。

就可选择（通过 MODE）的事件群组而言，该功能块决定多少 (LEN) 数据字节（来自标志类型 %MB0）应具备保持特性，即便未明确声明为 VAR_RETAIN。

事件	MODE = 0	MODE = 1	MODE = 2	MODE = 3
电源 OFF ⇨ ON	数据为新初始化	数据为剩余	数据为剩余	数据为剩余
软件复位	数据为新初始化	数据为剩余	数据为剩余	数据为剩余
冷复位	数据为新初始化	数据为新初始化	数据为剩余	数据为剩余
复位（默认）	数据为新初始化	数据为新初始化	数据为剩余	数据为剩余
加载应用程序	数据为新初始化	数据为新初始化	数据为剩余	数据为剩余
加载运行时系统	数据为新初始化	数据为新初始化	数据为新初始化	数据为剩余

如果 MODE = 0，则仅那些数据具有保持特性，跟明确声明为 VAR_RETAIN 的 MODE=1 一样。

如果决不调用 FB，则标志字节按照 MODE = 0 运行。 超过配置区域的标志字节亦按照 MODE = 0 运行。

配置后仍保留在装置中，即便重新加载应用程序或运行时系统。

输入端参数

2375

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
LEN	WORD	从标志地址 %MB0 起显示剩余特性的数据字节数量 允许 = 0...4 096 = 0x0...0x1000 LEN > 4 096 将自动修正为 LEN = 4 096
MODE	BYTE	这些变量应具有保持特性的事件 （0...3； → 上表） 就 MODE > 3 而言，最后的有效设定将保留

手动数据存储

内容

13801

除了可自动存储数据之外，还可通过功能块调用手动将用户数据存储在集成内存中并可从中读取。

ⓘ程序员可通过存储分区（→ 章节 **可用内存** (→ 页 19)）了解哪些内存区域可用。

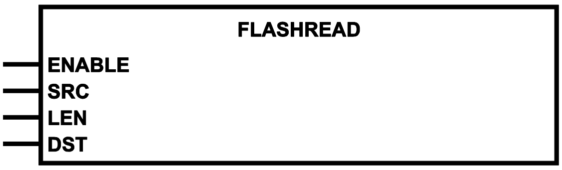
FLASHREAD

561

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

564

FLASHREAD 有助于直接从闪存读取不同类型的数据。

> FB 从闪存读取从 SRC 地址开始的内容。 且此时，传送的字节数量跟 LEN 所示一样。

> 调用 FB 的周期内可充分读取内容。

► 请确保 RAM 中的目标内存区域充足。

► 就目标地址而言，DST 适用：
(!) 通过运算符 ADR 确定地址并将其分配至 FB！

输入端参数

2318

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
SRC	DWORD	内存中的相对起始地址 允许 = 0...65 535 = 0x0...0x0000 FFFF (!) 若起始地址超出允许范围： > 无数据传输
LEN	DWORD	数据字节数量（最多 65 536 = 0x0001 0000） (!)如果所示字节数量超过闪存空间，则数据仅传送至闪存空间的末尾。
DST	DWORD	目标地址 (!) 通过运算符 ADR 确定地址并将其分配至 FB！

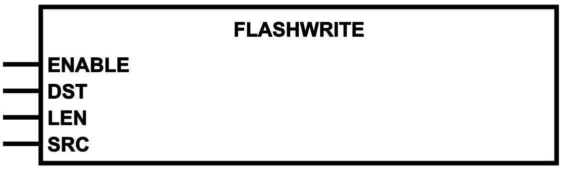
FLASHWRITE

555

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB

CODESYS 中的符号：



说明

19245

► 启用 TEST 输入端以使用功能块！ 否则会出现电子狗错误。

测试输入端已启用：

- 编程模式已启用
- 可下载软件
- 可查询应用程序的状态
- 无法保护存储的软件

558

⚠ 警告

无法控制流程操作可能造成危险！

执行 FLASHWRITE 时输入端/输出端状态“冻结”。

► 机器正在运行时切勿执行该 FB！

FLASHWRITE 有助于将不同的数据类型直接写入闪存。

设定期间可利用该 FB 存储大数据量，且在进程中仅可进行读取访问。

► 如果已写入页面（即便仅部分），需先删除整个闪存区域，而后再对该页面进行写入访问。 对地址 0 进行写入访问即可完成该操作。

► 切入多次写入至页面！ 始终先删除所有内容！
否则会出现陷阱或电子狗错误。

► ⓘ切勿删除闪存区域超过 100 次。 否则将无法保证其他闪存区域的数据一致性。

► 在每个 SPS 周期内，FLASHWRITE 即可启动一次！

► 就源开始地址而言，SRC 适用：
ⓘ 通过运算符 ADR 确定地址并将其分配至 FB！

> FB 将 SRC 地址内容写入闪存。 且此时，传送的字节数量跟 LEN 所示一样。

ⓘ 若目标开始地址 DST 超出允许范围： 无数据传输！

输入端参数

2603

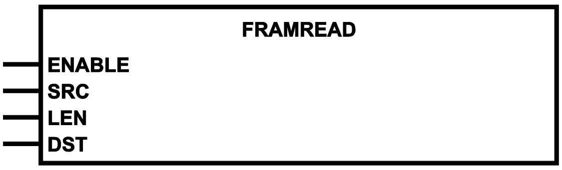
参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
DST	DWORD	内存中的相对起始地址 允许 = 0...65 535 = 0x0...0x0000 FFFF ! 通过运算符 ADR 确定地址并将其分配至 FB！
LEN	DWORD	数据字节数量（最多 65 536 = 0x0001 0000 ） ! 如果所示字节数量超过闪存空间，则数据仅传送至闪存空间的末尾。
SRC	DWORD	源地址

FRAMREAD 549

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明 552

FRAMREAD 有助于从 FRAM 中直接快速读取各种数据类型 [1])。

FB 从 FRAM 内存读取从 SRC 地址开始的内容。 且此时，传送的字节数量跟 LEN 所示一样。
如果超过 FRAM 内存区域所示的字节数量，则最多可读取到 FRAM 末端的数据。

► 就目标地址而言，DST 适用：
  [!] 通过运算符 ADR 确定地址并将其分配至 FB！

[1]) FRAM 在此表示所有类型的非易失快速内存。

输入端参数 2606

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
SRC	DWORD	内存中的相对起始地址 允许 = 0... 16 383 = 0x0000 0000...0x0000 3FFF
LEN	DWORD	数据字节数量 允许 = 0...16 384 = 0x0000 0000...0x0000 4000
DST	DWORD	目标地址 [!] 通过运算符 ADR 确定地址并将其分配至 FB！

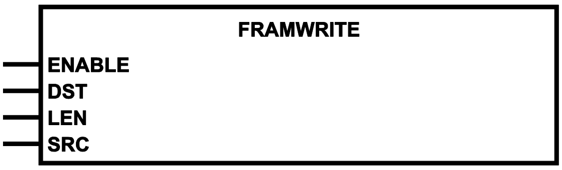
FRAMWRITE

543

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

546

FRAMWRITE 有助于将各种数据类型直接快速写入 FRAM 内存 [1])。

FB 将 SRC 地址内容写入非易失 FRAM 内存。 且此时，传送的字节数量跟 LEN 所示一样。

如果超过 FRAM 内存区域所示的字节数量，则最多可写入 FRAM 末端的数据。

► 就源地址而言，SRC 适用：
  ⓘ 通过运算符 ADR 确定地址并将其分配至 FB！

ⓘ 若目标地址 DST 超出允许范围： 无数据传输！

[1]) FRAM 在此表示所有类型的非易失快速内存。

输入端参数

2605

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
DST	DWORD	内存中的相对起始地址 允许 = 0... 16 383 = 0x0...0x0000 3FFF
LEN	DWORD	数据字节数量 允许 = 0...16 384 = 0x0000 0000...0x0000 4000
SRC	DWORD	源内存中的起始地址 ⓘ 通过运算符 ADR 确定地址并将其分配至 FB！

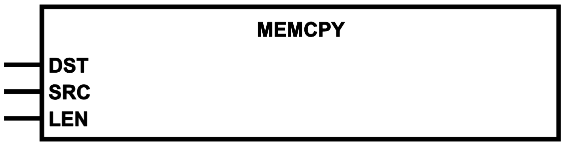
MEMCPY

409

= 内存复制

单元类型 = 功能块 (FB)

元件类型 ifm_CR0033_Vxxyyzz.LIB



说明

15944
412

MEMCPY 有助于直接写入和读取内存中各种类型的数据。

FB 将 SRC 地址内容写入地址 DST。

- 就地址而言，SRC 和 DST 适用：
  ⓘ 通过运算符 ADR 确定地址并将其分配至 FB！

\> 且此时，传送的字节数量跟 LEN 所示一样。 因此还可准确传送一个字变量的一个字节。

\> 如果数据所复制到的内存区域并非完全在允许的内存区域内，那么数据将不会复制，且会通过信号显示参数错误。

DST 内存区域	装置	内存大小
应用程序数据	（全部）	192K 字节
保留数据	CR0032, CR0232, CR0234, CR7n32	32K 字节
	CR0033, CR0133, CR0233, CR0235	64K 字节

表格“可用内存” → 章节 **可用内存** (→ 页 19)

输入端参数

413

参数	数据类型	说明
DST	DWORD	目标地址 ⓘ 通过运算符 ADR 确定地址并将其分配至 FB！
SRC	DWORD	源内存中的起始地址 ⓘ 通过运算符 ADR 确定地址并将其分配至 FB！
LEN	WORD	待传送数据字节数量 (≥ 1)

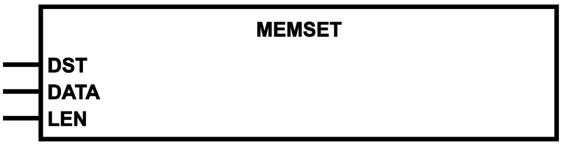
MEMSET

2348

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

2350

MEMSET 可供写入至定义的数据区域。

FB 将从 DST 地址开始的 DATA 内容写入内存，字节数量跟 LEN 所示一样。

► 就目标地址而言，DST 适用：
  ! 通过运算符 ADR 确定地址并将其分配至 FB！

> 如果数据所复制到的内存区域并非完全在允许的内存区域内，那么数据将不会复制，且会通过信号显示参数错误。

DST 内存区域	装置	内存大小
应用程序数据	（全部）	192K 字节
保留数据	CR0032, CR0232, CR0234, CR7n32	32K 字节
	CR0033, CR0133, CR0233, CR0235	64K 字节

输入端参数

2351

参数	数据类型	说明
DST	DWORD	目标地址 ! 通过运算符 ADR 确定地址并将其分配至 FB！
DATA	BYTE	待写入值
LEN	WORD	通过 DATA 覆盖的数据字节数量

5.2.19 功能元件： 数据访问和数据检查

内容

1598

本章所述 FB 控制数据访问且可启用数据检查。

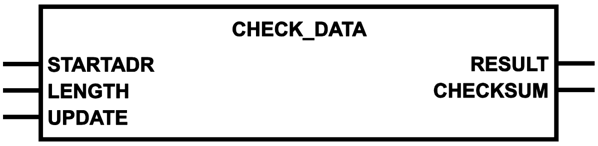
CHECK_DATA

603

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

606

CHECK_DATA 生成可配置内存区域的校验和 (CRC) 并检查针对不想要的更改的内存区域数据

- ► 针对每个待监控的内存区域创建单独的功能块实例。
- ► 通过运算符 ADR 确定地址并将其分配至 FB！
- ► 此外，指出数据字节数 LENGTH（长度从 STARTADR 开始）。

不需要的更改： 错误！
如果输入端 UPDATE = FALSE 且内存数据非故意更改，则 RESULT = FALSE。 而后结果可用于进一步操作（如禁用输出端）。

需要的更改：
内存（如应用程序或 **ecomat*mobile*** 装置的内存）数据更改仅在输出端 UPDATE 设为 TRUE 时是允许的。 而后重新计算校验和。 输出端 RESULT 再永久为 TRUE。

输入端参数

2612

参数	数据类型	说明
STARTADR	DWORD	受监控数据内存的起始地址 （WORD 地址，从 %MW0 起） 通过运算符 ADR 确定地址并将其分配至 FB！
LENGTH	DWORD	受监控数据内存长度，以 [byte] 为单位
UPDATE	BOOL	TRUE: 数据已更改 > 功能块计算新的校验和 FALSE: 数据未更改 > 功能块检查内存区域

输出端参数

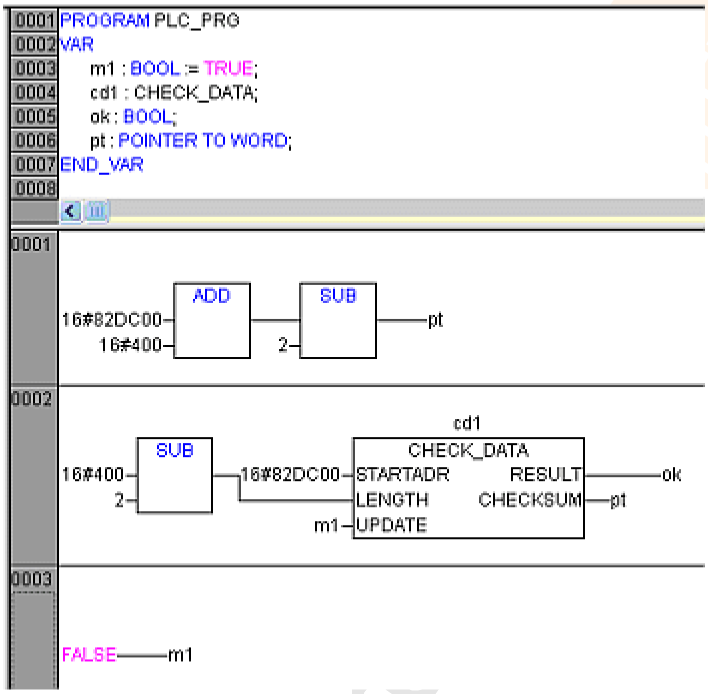
2613

参数	数据类型	说明
RESULT	BOOL	TRUE: CRC 校验和正常： 有意的数据更改或无更改 FALSE: CRC 校验和错误： 数据非故意更改
CHECKSUM	DWORD	当前 CRC 校验和

例如： CHECK_DATA

4168

在以下示例中，程序决定校验和并通过指针 pt 存储在 RAM 中：



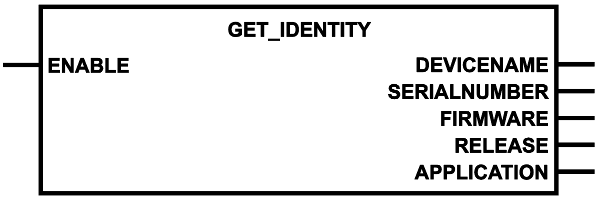
GET_IDENTITY

14505

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

14507

GET_IDENTITY reads the specific identifications stored in the device:

- 装置的硬件名称和硬件版本
- 装置的序列号
- 装置运行时系统的名称
- 装置运行时系统的版本和版本号
- 应用程序的名称（之前已通过以下方式保存： **SET_IDENTITY** (→ 页 258))

输入端参数

2609

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定

输出端参数

14508

参数	数据类型	说明
DEVICENAME	STRING(31)	硬件名称 作为最多 31 个字符的字符串，例如： "CR0403"
SERIALNUMBER	STRING(31)	装置的序列号 作为最多 31 个字符的字符串 例如： "12345678"
FIRMWARE	STRING(31)	装置运行时系统的名称 作为最多 31 个字符的字符串 示例： "CR0403"
RELEASE	STRING(31)	软件版本 作为最多 31 个字符的字符串
APPLICATION	STRING(79)	应用程序名称 作为最多 79 个字符的字符串 示例： "Crane1704"

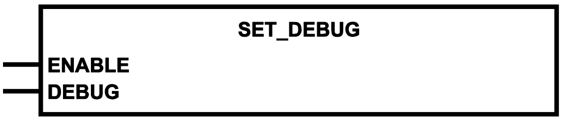
SET_DEBUG

290

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

293

SET_DEBUG 处理 DEBUG 模式而无需启用测试输入端 (→ 章节 **TEST 模式** (→ 页 63)).

如果 FB 的输入端 DEBUG 设为 TRUE，编程系统或下载程序等则可与装置通信并执行某些特别的系统命令 (如针对服务功能，通过 GSM 调制解调器 CANremote) 。

> ! 在该工作模式中，不可进行软件下载，因为测试输入端未连接至电源电压。 仅可进行读取访问。

输入端参数

294

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
DEBUG	BOOL	TRUE： 可通过接口调试 FALSE： 不可通过接口调试

SET_IDENTITY

11927

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

说明

287

SET_IDENTITY 设定应用程序特定程序 ID。

应用程序可利用该 FB 创建程序 ID。 该 ID（即软件版本）可通过软件工具 DOWNLOADER.EXE 读取，以识别下载的程序。

下图显示不同软件工具所示不同 ID 的相互联系。 （示例： ClassicController CR0020):

*) 'Nozzle in front' 在此代替定制文本。

输入端参数

11928

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
ID	STRING(79)	任何所需文本，最大长度为 79 个字符

SET_PASSWORD

266

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

说明

269

SET_PASSWORD 设定通过 DOWNLOADER 进行程序和内存上传的用户密码。

如果密码启用，则仅可在输入正确的密码时通过软件工具 DOWNLOADER 读取应用程序或数据内存。

如果空字串（默认条件）分配至输入端 PASSWORD，则可随时上传应用程序软件或数据内存。

仅可在复位之前密码后设定新密码。

> ⓘ 加载新的应用程序以作为启动项目时，密码复位。

输入端参数

2353

参数	数据类型	说明
ENABLE	BOOL	FALSE ⇨ TRUE (edge): 初始化功能块（仅 1 个周期） > 读取功能块输入 TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
PASSWORD	STRING(16)	密码 如果 PASSWORD = ""，则无需输入密码即可访问

5.2.20 功能元件： 管理错误消息

内容

19229

我们将在此讲述可供您进行以下操作的功能：

- 生成应用程序特定错误代码
- 列出或删除错误代码

ERROR_REPORT

12357

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

说明

12364

应用程序通过 ERROR_REPORT 将应用程序特定错误发送给系统。

► 将错误条件的结果编程至输入端 ENABLE。
如果 ENABLE=TRUE，则功能块输入错误列表中的错误代码。
• 利用以下功能块查看当前错误列表： **SHOW_ERROR_LIST** (→ 页 268)
• 利用以下功能块删除错误列表中的错误条目 **ERROR_RESET** (→ 页 264)

► 将相应的错误代码编程至输入端 ERRORCODE：
结构： yy xx 00 00 → 章节 **错误代码** (→ 页 373)
xx = 应用程序特定错误代码
yy = 错误类别
此处，功能块 **PACK_ERRORCODE** (→ 页 267)

> 功能块输出端发送该功能块调用是否正确配置的信号。
! 功能块不分析错误代码是否有意义。

输入端参数

12363

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 错误生效 用信号发送相应的 ERRORCODE FALSE: 单元未执行 故障（不再）有效
ERRORCODE	DWORD	出现该错误代码时，配置的状况将适用。 → 章节 **错误代码** (→ 页 373) ! 功能块不分析错误代码是否有意义。

输出端参数

19255

参数	数据类型	说明
ERROR	DWORD	来自该功能块调用的错误代码 → **错误代码** (→ 页 373) （可能的消息 → 下表）

ERROR 的可能结果（n= 任何所需值）：

32 位错误代码包含四个 8 位值 (DWORD)。

第 4 个字节	第 3 个字节	第 2 个字节	第 1 个字节
错误类别	应用程序特定错误代码	错误来源	错误原因

值 [hex]	说明
00 00 00 00	无错误
01 00 00 F8	错误参数 ⇨ 一般错误

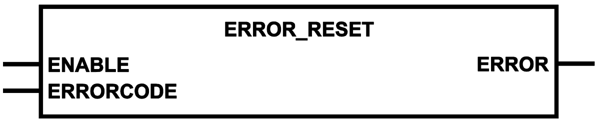
ERROR_RESET

12376

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

12378

应用程序可通过 ERROR_RESET 复位即将到来的错误消息：

- 单一错误消息
- 一组相同类型的错误消息（相同来源或相同原因）
- 应用程序特定错误：
  - 删除所有包含所示 USER 错误代码的错误。
  - 忽略错误类别。
  - 错误原因和错误来源必须包含值 '0'。
- 如果复位类别 3 或 4 的错误，则…
  - 设定标志 ERROR=FALSE
  - 亦删除所有其他类别 3 和 4 的错误。
- 所有错误

根据输入的 ERRORCODE，ENABLE=TRUE 时出现以下情况：

ERRORCODE	注意	说明
0xCL 00 SO **00**	错误原因 = 0	复位所有包含相同错误来源的错误
0xCL 00 **00** CA	错误来源 = 0	复位所有包含相同错误原因的错误
0xXX **AS** SO CA	应用程序特定错误	复位包含相同编号的错误（与错误类别无关）
0x**00 00 00 00**	错误代码 = 0	复位所有错误

说明：
CL = 错误类别的代码
AS = 应用程序特定错误的代码
SO = 错误来源的代码
CA = 错误原因的代码
XX = 无任何作用的代码值

> 只要错误生效，复位时会出现以下情况：
- 故障暂时复位。
- 当诊断时间在又一次程序周期后结束： 再次发送故障的信号。

► FB 的两次复位操作之间：
设定 ENABLE=FALSE（至少针对一次 PLC 周期）！

输入端参数

12379

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
ERRORCODE	DWORD	出现该错误代码时，配置的状况将适用。 → 章节 **错误代码** (→ 页 373) ❗功能块不分析错误代码是否有意义。

输出端参数

19257

参数	数据类型	说明
ERROR	DWORD	来自该功能块调用的错误代码 → **错误代码** (→ 页 373) （可能的消息 → 下表）

ERROR 的可能结果（n= 任何所需值）：

32 位错误代码包含四个 8 位值 (DWORD)。

第 4 个字节	第 3 个字节	第 2 个字节	第 1 个字节
错误类别	应用程序特定错误代码	错误来源	错误原因

值 [hex]	说明
00 00 00 00	无错误
01 00 00 F8	错误参数 ➪ 一般错误

例如： ERROR_RESET

13054

如果所有错误“过载”要通过错误类别“一般错误”复位，则需指明 ERRORCODE=0x01000004：

第 4 个字节	第 3 个字节	第 2 个字节	第 1 个字节
错误类别	应用程序特定错误代码	错误来源	错误原因
0x01	0x00	0x00	0x04
一般错误	无应用程序特定错误	所有错误来源	过载

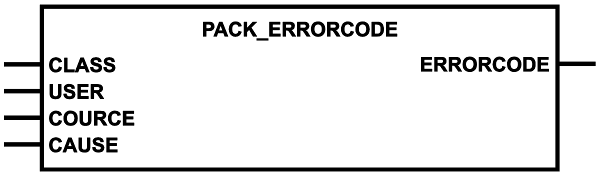
PACK_ERRORCODE

12382

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

12384

PACK_ERRORCODE 帮助将针对以下对象的组件打包为 ERRORCODE：

- 错误类别
- 应用程序特定错误
- 错误来源
- 错误原因

(结构 → 章节 **错误代码** (→ 页 373)).

❗功能块不分析错误代码是否有意义。

输入端参数

12385

参数	数据类型	说明
CAUSE	BYTE	错误原因的代码
SOURCE	BYTE	错误来源的代码
USERCODE	BYTE	应用程序特定错误的代码
CLASS	BYTE	错误类别的代码

输出端参数

12390

参数	数据类型	说明
ERRORCODE	DWORD	创建的错误代码 → 章节 **错误代码** (→ 页 373) ❗功能块不分析错误代码是否有意义。

SHOW_ERROR_LIST

12360

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：

说明

12367

功能块 SHOW_ERROR_LIST 用于读取当前显示的错误代码。

功能块可通过 ENABLE=TRUE 创建包含高达 64 个当前存在的错误代码的列表。
ENABLE=FALSE 保存未经更改的最近列表版本。

如果列表已满，则不接受更多错误代码。

输入端参数

12368

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定

输出端参数

12369

参数	数据类型	说明
ERRORS	ARRAY [0..63] OF DWORD	当前显示的错误代码的列表 → 章节 **错误代码** (→ 页 373)

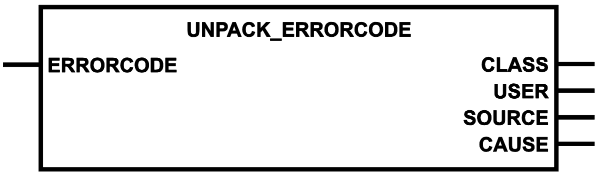
UNPACK_ERRORCODE

13650

单元类型 = 功能块 (FB)

元件类型 `ifm_CR0033_Vxxyyzz.LIB`

CODESYS 中的符号：



说明

13653

UNPACK_ERRORCODE 将 ERRORCODE 分成几个组件：

- 错误类别
- 应用程序特定错误
- 错误来源
- 错误原因

(结构 → 章节 **错误代码** (→ 页 373)).

❗功能块不分析错误代码是否有意义。

输入端参数

13654

参数	数据类型	说明
ERRORCODE	DWORD	错误代码 →章节 **错误代码** (→ 页 373) ❗功能块不分析错误代码是否有意义。

输出端参数

13675

参数	数据类型	说明
CLASS	BYTE	错误类别的代码
USERCODE	BYTE	应用程序特定错误的代码
SOURCE	BYTE	错误来源的代码
CAUSE	BYTE	错误原因的代码

6 诊断和错误处理

内容

19598

运行时系统 (RTS) 通过内部错误检查功能检查装置：

- 启动阶段（复位阶段）
- 应用程序执行期间

→ 章节 ***工作状态*** (→ 页 58)

此时尽可能地实现高工作可靠性。

6.1 诊断

19601

诊断期间，检查装置的“健康状态”。 从而将发现装置是否存在→以及存在何种故障。

还可监控输入端和输入端是否正常运行，具体视装置而定。

- 断线，
- 短路，
- 值在范围之外。

就诊断而言，可使用装置“正常”运行期间创建的配置和日志数据。
初始化和启动期间可监控系统组件是否正确启动。 日志文件记录错误。
就进一步诊断而言，还可进行自检。

6.2 故障

19602

故障即无法执行所需功能的项目状态，不包括预防性维护或其他计划措施期间或因为缺乏外部资源而导致的失效状态。
故障通常是项目本身失灵的结果，也可能在之前没有失灵的情况下存在。
在 →ISO 13849-1 中，“故障”指“随机故障”。

6.3 发生错误时的反应

19603
12217

检测到错误时，还可在应用程序中设定系统标志 ERROR。 因此，在出现故障时，控制器反应如下：

> 工作 LED 呈红色亮起，

> 输出端继电器关闭，

> 继电器保护的输出端断开电源，

> 输出端逻辑信号状态保持不变。

注意

如果继电器关闭输出端，则逻辑信号状态保持不变。

► 程序员必须评估 ERROR 位，因此还需在出现故障时复位输出逻辑。

装置特定错误代码和诊断消息的完整列表
→ 章节 **系统标志** (→ 页 273).

6.4 继电器： 重要说明！

14034

通知

继电器触点可能过早磨损。

► 在正常操作中，仅切换无负载的继电器！
就此而言，通过应用程序将所有相关输出端设为 FALSE！

6.5 响应系统错误

14033

> (!) 程序员单独负责应用程序数据处理的安全性。

► 在应用程序中处理特定错误标志和/或错误代码！
错误说明通过错误标志/错误代码提供。
如有必要，该错误标志/错误代码可进一步加以处理。

分析和排除错误原因后：

► 作为一般规则，通过应用程序复位所有错误标志。
在未明确复位错误标志的情况下，标志保持设定状态并对应用程序产生相应的作用。

6.6 CAN / CANopen: 错误和错误处理

19604

→ 系统手册“ecomat*mobile* 技巧”
→ 章节 **CAN / CANopen： 错误和错误处理**

7 附录

内容

1664

除技术资料指示外，您还可了解附录中的摘要表。

7.1 系统标志

内容

12167

如果扩展 PLC 配置，则系统标志的地址可更改。

► 编程时仅可使用系统标志的符号名称！

→ 系统手册“ecomat*mobile* 技巧”

→ 章节 **错误代码和诊断信息**

7.1.1 系统标志： CAN

12820

系统标志（符号名称）	类型	说明
CANx_BAUDRATE	WORD	CAN interface x: 设定以 [kBaud] 为单位的波特率
CANx_BUSOFF	BOOL	CAN interface x: 错误"CAN 总线关闭" ℹ 复位错误代码亦会复位标志
CANx_DOWNLOADID	BYTE	CAN 接口 x： 设定下载 ID
CANx_ERRORCOUNTER_RX	BYTE	CAN 接口 x： 错误计数接收器 ℹ 标志可通过写入访问复位
CANx_ERRORCOUNTER_TX	BYTE	CAN 接口 x： 错误计数传送 ℹ 标志可通过写入访问复位
CANx_LASTERROR	BYTE	CAN interface x: 最后 CAN 传送的错误编号 0 = 无错误：初始值 1 = 填充错误：总线上连续有 5 个以上相同的位 2 = 格式错误：接收的消息格式错误 3 = 应答错误：发送的消息未确认 4 = bit1 错误：在仲裁区域之外发送了隐性位，但在总线上读取了显性位。 5 = bit0 错误：尝试了发送显性位，但读取了隐性位 或： 总线关闭恢复期间读取了包含 11 个隐性位的序列 6 = CRC 错误：已接收消息的校验错误
CANx_WARNING	BOOL	CAN 接口 x： 达到了警告阈值 (≥ 96) ℹ 标志可通过写入访问复位

CANx 代表 x = 1...4 = CAN 接口编号

7.1.2 系统标志： SAE-J1939

12815

系统标志（符号名称）	类型	说明
J1939_RECEIVE_OVERWRITE	BOOL	设定仅适用于未通过 J1939 传输协议传送的 J1939 数据。 TRUE: 如果旧数据还未从功能块实例读取，则新数据覆盖旧数据 FALSE: 只要旧数据还未从功能块实例读取，则拒绝新数据 如果 IEC 周期长于 J1939 数据的刷新速度，则新数据可在旧数据被读取之前到达。
J1939_TASK	BOOL	使用 J1939_TASK 即可满足发送 J1939 消息的时间要求。 如果 J1939 消息要在重复时间 ≤ 50 ms 的情况下发送，则运行时系统自动设定 J1939_TASK=TRUE。 就时间要求为 ≥ PLC 周期时间的应用而言： ▶ 通过 J1939_TASK=FALSE 减少系统负载！ TRUE： J1939 任务启用（= 初始值） 每 2 ms 调用任务。 J1939 堆栈在所需时间帧内 发送其消息 FALSE： J1939 任务未启用

7.1.3 系统标志： 错误标志（标准侧）

2379

系统标志（符号名称）	类型	说明
ERROR	BOOL	TRUE: 假定安全状态 所有输出端 = OFF 输出端继电器 = OFF （如严重错误/Error-Stop） FALSE: 未发生严重错误
ERROR_BREAK_Ix （0...x，值视装置而定→ 技术 资料）	DWORD	输入双字 x： 断线错误 或（电阻输入）： • 电源短路 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE： 错误 位 = FALSE： 无错误
ERROR_BREAK_Qx （0...x，值视装置而定→ 技术 资料）	DWORD	输出双字 x： 断线错误 [Bit 0 for output 0] ... [bit z for output z] of this group 位 = TRUE： 错误 位 = FALSE： 无错误
ERROR_CONTROL_Qx （0...x，值视装置而定→ 技术 资料）	DWORD	输出双字 x：电流控制错误 无法达到最终值 [Bit 0 for output 0] ... [bit z for output z] of this group 位 = TRUE： 错误 位 = FALSE： 无错误
ERROR_CURRENT_Ix （0...x，值视装置而定→ 技术 资料）	DWORD	输入双字 x： 过电流错误 仅当 Ixx_MODE = IN_CURRENT 时 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE： 错误 位 = FALSE： 无错误
ERROR_POWER	BOOL	VBBs /夹具 15 的过电压错误： TRUE： 值超出范围 或： 差值 (VBB15 - VBBs) > 1 V > 一般错误 > 应用程序停止 > 输出端 = 未启用 > 无通信 > 消息 "Overvoltage clamp 15" FALSE： 值正常
ERROR_SHORT_Ix （0...x，值视装置而定→ 技术 资料）	DWORD	输入双字 x： 短路错误 仅当输入模式 = IN_DIGITAL_H 时 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE： 错误 位 = FALSE： 无错误

系统标志（符号名称）	类型	说明
ERROR_SHORT_Qx （0...x，值视装置而定→ 技术 资料）	DWORD	输出双字 x： 短路错误或过载错误 [Bit 0 for output 0] ... [bit z for output z] of this group 位 = TRUE： 错误 位 = FALSE： 无错误
ERROR_TEMPERATURE	BOOL	温度错误 TRUE： 值超出范围 > 一般错误 FALSE： 值正常
ERROR_VBBx	BOOL	VBBx (x = o | r) 电源电压错误： TRUE： 值超出范围 > 一般错误 FALSE： 值正常
ERRORCODE	DWORD	写入内部错误列表的最后一个错误 列表包含出现的所有错误代码。
LAST_RESET	BYTE	上次复位的原因： 00 = 应用程序复位 01 = 通电复位 02 = 电子狗复位 03 = 软复位 04 = 未知原因

7.1.4 系统标志： 状态 LED（标准侧）

12817

系统标志（符号名称）	类型	说明
LED	WORD	“LED 开启”的 LED 颜色 0x0000 = LED_GREEN（预设） 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_X	WORDWORD	“LED 关闭”的 LED 颜色 0x0000 = LED_GREEN 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK（预设） 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_MODE	WORD	LED 闪烁频率： 0x0000 = LED_2HZ（以 2 Hz 的频率闪烁； 预设） 0x0001 = LED_1HZ（以 1 Hz 的频率闪烁） 0x0002 = LED_05HZ（以 0.5 Hz 的频率闪烁） 0x0003 = LED_0HZ（按照 LED 中的值永久亮起）

7.1.5 系统标志： 电压（标准侧）

12135

系统标志（符号名称）	类型	说明
CLAMP_15_VOLTAGE	WORD	给夹具 15 施加的电压，以 [mV] 为单位
REF_VOLTAGE	WORD	参考电压输出端电压，以 [mV] 为单位
REFERENCE_VOLTAGE_5	BOOL	5 V 参考电压输出端启用
REFERENCE_VOLTAGE_10	BOOL	10 V 参考电压输出端启用
RELAIS_VBBy y = o | r	BOOL	TRUE: VBBy 继电器启用 电压施加至输出组 x (x = 1 | 2) FALSE: VBBy 继电器禁用 无电压施加至输出组 x
SERIAL_MODE	BOOL	启用串行接口 (RS232) 以用于应用程序 TRUE： RS232 接口可用于应用程序，但不再可用于装置的编程、调试或监控。 FALSE： RS232 接口不可用于应用程序。 装置可进行编程、调试和监控。
SUPPLY_SWITCH	BOOL	关闭电源封锁 VBBs 的位。 如果夹具 15 < 4 V，系统仅可接受复位标志，否则标志再次启用。 VBBs 分离在下一次 PLC 周期开始之前完成。 装置关闭可能需要一些时间，具体视内部电容器的充电状态而定。 TRUE: 通过 VBBs 为装置供电启用 FALSE: 通过 VBBs 为装置供电禁用
SUPPLY_VOLTAGE	WORD	VBBs 上的电源电压，以 [mV] 为单位
TEST	BOOL	TRUE: 测试输入端已启用： • 编程模式已启用 • 可下载软件 • 可查询应用程序的状态 • 无法保护存储的软件 FALSE: 应用程序在运行中
VBBx_RELAIS_VOLTAGE x = o | r	WORD	VBBx 到继电器触点的电源电压，以 [mV] 为单位
VBBx_VOLTAGE x = o | r	WORD	VBBx 上的电源电压，以 [mV] 为单位

7.1.6 系统标志： 16 个输入端和 16 个输出端（标准侧）

13121

系统标志（符号名称）	类型	说明
ANALOGxx xx = 00...15	WORD	模拟输入端 xx： 已过滤 A/D 转换器原始值（12 位），无校准或标准化
ANALOG_IRQxx xx = 00...07	WORD	模拟输入端 xx： 未过滤 A/D 转换器原始值（12 位） 无校准或标准化 用于 FB SET_INTERRUPT_I (→ 页 152)或 SET_INTERRUPT_XMS (→ 页 154)
CURRENTxx xx = 00...15	WORD	PWM 输出端 xx： 电流测量已过滤 A/D 转换器原始值（12 位），无校准或标准化
Ixx xx = 00...15	BOOL	二进制输入端 xx 上的状态 条件： 输入端配置为二进制输入端 （MODE = IN_DIGITAL_H 或 IN_DIGITAL_L） TRUE: 二进制输入端上的电压 > VBBS 的 70 % FALSE: 二进制输入端上的电压 < VBBS 的 30 % 或： 未配置为二进制输入端 或： 错误的配置
Ixx_DFILTER xx = 00...11	DWORD	脉冲输入端 xx： 脉冲持续时间，以 [µs] 为单位，作为小故障忽略不计。 输入信号的获得按照设定的时间延迟。 允许 = 0...100 000 µs 预设 = 0 µs = 无过滤器
Ixx_FILTER xx = 00...15	BYTE:=4	二进制和模拟输入端 xx： 第一级软件低通过滤器的极限频率（或信号上升时间） 0 = 0x00 = 无过滤器 1 = 0x01 = 390 Hz (1 ms) 2 = 0x02 = 145 Hz (2.5 ms) 3 = 0x03 = 68 Hz (5 ms) 4 = 0x04 = 34 Hz (10 ms) (preset) 5 = 0x05 = 17 Hz (21 ms) 6 = 0x06 = 8 Hz (42 ms) 7 = 0x07 = 4 Hz (84 ms) 8 = 0x08 = 2 Hz (169 ms) 更高 = → 预设值
Qxx xx = 00...15	BOOL	二进制输出端 xx 上的状态： 条件： 输出端配置为二进制输出端 TRUE: 输出端启用 FALSE: 输出端禁用（= 初始值） 或： 未配置为二进制输出端

系统标志（符号名称）	类型	说明
Qxx_FILTER xx = 00...15	BYTE	输出端 xx： 电流测量第一级软件低通过滤器的极限频率 仅当 Qxx_MODE = OUT_DIGITAL_H 时 PWM 模式则不可 0 = 0x00 = 无过滤器 1 = 0x01 = 580 Hz (0.6 ms) 2 = 0x02 = 220 Hz (1.6 ms) 3 = 0x03 = 102 Hz (3.5 ms) 4 = 0x04 = 51 Hz (7 ms) (preset) 5 = 0x05 = 25 Hz (14 ms) 6 = 0x06 = 12 Hz (28 ms) 7 = 0x07 = 6 Hz (56 ms) 8 = 0x08 = 3 Hz (112 ms) 更高 =→ 预设值

7.2 地址分配和 I/O 工作模式

内容

1656

→ 另有技术资料

7.2.1 I/O 地址/变量

内容

2376

输入端： 地址和变量（标准侧）（16 个输入端）

10432

缩写词 →章节 **关于配线的注意事项** (→ 页 42)
输入端/输出端工作模式 →章节 **可能的输入端/输出端工作模式** (→ 页 289)

IEC 地址	I/O 变量	备注
%IB0		输入字节 0 (%IX0.0...%IX0.7)
%IB1		输入字节 1 (%IX0.8...%IX0.15)
%IW0		输入字 (%IX0...%IX0.15)
%IX0.0	I00	二进制输入通道 0
%IX0.1	I01	二进制输入通道 1
%IX0.2	I02	二进制输入通道 2
%IX0.3	I03	二进制输入通道 3
%IX0.4	I04	二进制输入通道 4
%IX0.5	I05	二进制输入通道 5
%IX0.6	I06	二进制输入通道 6
%IX0.7	I07	二进制输入通道 7
%IX0.8	I08	二进制输入通道 8
%IX0.9	I09	二进制输入通道 9
%IX0.10	I10	二进制输入通道 10
%IX0.11	I11	二进制输入通道 11
%IX0.12	I12	二进制输入通道 12
%IX0.13	I13	二进制输入通道 13
%IX0.14	I14	二进制输入通道 14
%IX0.15	I15	二进制输入通道 15
%IW2	ANALOG00	模拟输入通道 0
%IW3	ANALOG01	模拟输入通道 1
%IW4	ANALOG02	模拟输入通道 2
%IW5	ANALOG03	模拟输入通道 3
%IW6	ANALOG04	模拟输入通道 4
%IW7	ANALOG05	模拟输入通道 5
%IW8	ANALOG06	模拟输入通道 6
%IW9	ANALOG07	模拟输入通道 7
%IW10	ANALOG08	模拟输入通道 8

IEC 地址	I/O 变量	备注
%IW11	ANALOG09	模拟输入通道 9
%IW12	ANALOG10	模拟输入通道 10
%IW13	ANALOG11	模拟输入通道 11
%IW14	ANALOG12	模拟输入通道 12
%IW15	ANALOG13	模拟输入通道 13
%IW16	ANALOG14	模拟输入通道 14
%IW17	ANALOG15	模拟输入通道 15
%IW18	CURRENT00	Q00 上的输出电流（原始值）
%IW19	CURRENT01	Q01 上的输出电流（原始值）
%IW20	CURRENT02	Q02 上的输出电流（原始值）
%IW21	CURRENT03	Q03 上的输出电流（原始值）
%IW22	CURRENT04	Q04 上的输出电流（原始值）
%IW23	CURRENT05	Q05 上的输出电流（原始值）
%IW24	CURRENT06	Q06 上的输出电流（原始值）
%IW25	CURRENT07	Q07 上的输出电流（原始值）
%IW26	CURRENT08	Q08 上的输出电流（原始值）
%IW27	CURRENT09	Q09 上的输出电流（原始值）
%IW28	CURRENT10	Q10 上的输出电流（原始值）
%IW29	CURRENT11	Q11 上的输出电流（原始值）
%IW30	CURRENT12	Q12 上的输出电流（原始值）
%IW31	CURRENT13	Q13 上的输出电流（原始值）
%IW32	CURRENT14	Q14 上的输出电流（原始值）
%IW33	CURRENT15	Q15 上的输出电流（原始值）
%IW34	SUPPLY_VOLTAGE	VBBs 上的电源电压，以 [mV] 为单位
%IW35	CLAMP_15_VOLTAGE	电压夹具 15
%IW36	VBBO_VOLTAGE	VBBo 上的电源电压，以 [mV] 为单位
%IW37	VBBR_VOLTAGE	VBBr 上的电源电压，以 [mV] 为单位
%IW38	VBBO_RELAIS_VOLTAGE	VBBo 到继电器触点的电源电压，以 [mV] 为单位
%IW39	VBBR_RELAIS_VOLTAGE	VBBr 到继电器触点的电源电压，以 [mV] 为单位
%IW40	REF_VOLTAGE	参考输出插脚 51 上的电压
%IW41	ANALOG_IRQ00	模拟输入通道 0 中断

IEC 地址	I/O 变量	备注
%IW42	ANALOG_IRQ01	模拟输入通道 1 中断
%IW43	ANALOG_IRQ02	模拟输入通道 2 中断
%IW44	ANALOG_IRQ03	模拟输入通道 3 中断
%IW45	ANALOG_IRQ04	模拟输入通道 4 中断
%IW46	ANALOG_IRQ05	模拟输入通道 5 中断
%IW47	ANALOG_IRQ06	模拟输入通道 6 中断
%IW48	ANALOG_IRQ07	模拟输入通道 7 中断
%MB7960	ERROR_CURRENT_I0	错误 DWORD 过电流
%MB7964	ERROR_SHORT_I0	错误 DWORD 短路
%MB7968	ERROR_BREAK_I0	错误 DWORD 断线

输出端： 地址和变量（标准侧）（16 个输出端）

10433

缩写词 →章节 **关于配线的注意事项** (→ 页 42)
输入端/输出端工作模式 →章节 **可能的输入端/输出端工作模式** (→ 页 289)

IEC 地址	I/O 变量	备注
%QW0		输出字 0 (%QX0.0...%QX0.15)
%QB0		输出字节 0 (%QX0.0...%QX0.7)
%QB1		输出字节 1 (%QX0.8...%QX0.15)
%QX0.0	Q00	二进制输出/PWM 输出通道 0
%QX0.1	Q01	二进制输出/PWM 输出通道 1
%QX0.2	Q02	二进制输出/PWM 输出通道 2
%QX0.3	Q03	二进制输出/PWM 输出通道 3
%QX0.4	Q04	二进制输出/PWM 输出通道 4
%QX0.5	Q05	二进制输出/PWM 输出通道 5
%QX0.6	Q06	二进制输出/PWM 输出通道 6
%QX0.7	Q07	二进制输出/PWM 输出通道 7
%QX0.8	Q08	二进制输出/PWM 输出通道 8
%QX0.9	Q09	二进制输出/PWM 输出通道 9
%QX0.10	Q10	二进制输出/PWM 输出通道 10
%QX0.11	Q11	二进制输出/PWM 输出通道 11
%QX0.12	Q12	二进制输出/PWM 输出通道 12
%QX0.13	Q13	二进制输出/PWM 输出通道 13
%QX0.14	Q14	二进制输出/PWM 输出通道 14
%QX0.15	Q15	二进制输出/PWM 输出通道 15
%QB2	REFERENCE_VOLTAGE_5	启用 5 V 参考电压输出端
%QB3	REFERENCE_VOLTAGE_10	启用 10 V 参考电压输出端
%QB68	I00_FILTER	针对 %IX0.0 / %IW2 的过滤字节
%QB69	I01_FILTER	针对 %IX0.1 / %IW3 的过滤字节
%QB70	I02_FILTER	针对 %IX0.2 / %IW4 的过滤字节
%QB71	I03_FILTER	针对 %IX0.3 / %IW5 的过滤字节
%QB72	I04_FILTER	针对 %IX0.4 / %IW6 的过滤字节
%QB73	I05_FILTER	针对 %IX0.5 / %IW7 的过滤字节

IEC 地址	I/O 变量	备注
%QB74	I06_FILTER	针对 %IX0.6 / %IW8 的过滤字节
%QB75	I07_FILTER	针对 %IX0.7 / %IW9 的过滤字节
%QB76	I08_FILTER	针对 %IX0.8 / %IW2 的过滤字节
%QB77	I09_FILTER	针对 %IX0.9 / %IW3 的过滤字节
%QB78	I10_FILTER	针对 %IX0.10 / %IW4 的过滤字节
%QB79	I11_FILTER	针对 %IX0.11 / %IW5 的过滤字节
%QB80	I12_FILTER	针对 %IX0.12 / %IW6 的过滤字节
%QB81	I13_FILTER	针对 %IX0.13 / %IW7 的过滤字节
%QB82	I14_FILTER	针对 %IX0.14 / %IW8 的过滤字节
%QB83	I15_FILTER	针对 %IX0.15 / %IW9 的过滤字节
%QB84	Q00_FILTER	针对 %QX0.0 的过滤字节
%QB85	Q01_FILTER	针对 %QX0.1 的过滤字节
%QB86	Q02_FILTER	针对 %QX0.2 的过滤字节
%QB87	Q03_FILTER	针对 %QX0.3 的过滤字节
%QB88	Q04_FILTER	针对 %QX0.4 的过滤字节
%QB89	Q05_FILTER	针对 %QX0.5 的过滤字节
%QB90	Q06_FILTER	针对 %QX0.6 的过滤字节
%QB91	Q07_FILTER	针对 %QX0.7 的过滤字节
%QB92	Q08_FILTER	针对 %QX0.8 的过滤字节
%QB93	Q09_FILTER	针对 %QX0.9 的过滤字节
%QB94	Q10_FILTER	针对 %QX0.10 的过滤字节
%QB95	Q11_FILTER	针对 %QX0.11 的过滤字节
%QB96	Q12_FILTER	针对 %QX0.12 的过滤字节
%QB97	Q13_FILTER	针对 %QX0.13 的过滤字节
%QB98	Q14_FILTER	针对 %QX0.14 的过滤字节
%QB99	Q15_FILTER	针对 %QX0.15 的过滤字节
%QD25	I00_DFILTER	过滤值计数/脉冲输入端 0
%QD26	I01_DFILTER	过滤值计数/脉冲输入端 1
%QD27	I02_DFILTER	过滤值计数/脉冲输入端 2
%QD28	I03_DFILTER	过滤值计数/脉冲输入端 3
%QD29	I04_DFILTER	过滤值计数/脉冲输入端 4

IEC 地址	I/O 变量	备注
%QD30	I05_DFILTER	过滤值计数/脉冲输入端 5
%QD31	I06_DFILTER	过滤值计数/脉冲输入端 6
%QD32	I07_DFILTER	过滤值计数/脉冲输入端 7
%QD33	I08_DFILTER	过滤值计数/脉冲输入端 8
%QD34	I09_DFILTER	过滤值计数/脉冲输入端 9
%QD35	I10_DFILTER	过滤值计数/脉冲输入端 10
%QD36	I11_DFILTER	过滤值计数/脉冲输入端 11
%MB7948	ERROR_SHORT_Q0	错误 DWORD 短路
%MB7952	ERROR_BREAK_Q0	错误 DWORD 断线
%MB7956	ERROR_CONTROL_Q0	错误 DWORD 电流控制

7.2.2 可能的输入端/输出端工作模式

内容

2386

输入端： 工作模式（标准侧）（16 个输入端）

19368

= 该配置值为默认值

输入端	可能的工作模式		利用功能块设定	功能块输入端	值	
					十进制	十六进制
I00...I11	IN_NOMODE	关闭	INPUT_ANALOG SET_INPUT_ANALOG	MODE	0	0000
	IN_DIGITAL_H	正	INPUT_ANALOG SET_INPUT_ANALOG	MODE	1	0001
	IN_DIGITAL_L	负	INPUT_ANALOG SET_INPUT_ANALOG	MODE	2	0002
	IN_CURRENT	0...20 000 µA	INPUT_ANALOG SET_INPUT_ANALOG	MODE	4	0004
	IN_VOLTAGE10	0...10 000 mV	INPUT_ANALOG SET_INPUT_ANALOG	MODE	8	0008
	IN_VOLTAGE30	0...32 000 mV	INPUT_ANALOG SET_INPUT_ANALOG	MODE	16	0010
	IN_RATIO	0...1 000 ‰	INPUT_ANALOG SET_INPUT_ANALOG	MODE	32	0020
	诊断	针对 IN_DIGITAL_H	SET_INPUT_MODE	DIAGNOSTICS	TRUE	
	频率测量 间隔测量 相位测量	0...30 000 Hz	FREQUENCY FREQUENCY_PERIOD PHASE			
	间隔测量	0,1...5 000 Hz	PERIOD			
	周期和比率测量	0,1...5 000 Hz	PERIOD_RATIO			
	计数器	0...30 000 Hz	FAST_COUNT			
I00...I07	检测编码器	0...30 000 Hz 0...5 000 Hz	INC_ENCODER INC_ENCODER_HR			
I12...I14	IN_NOMODE	关闭	INPUT_ANALOG SET_INPUT_ANALOG	MODE	0	0000
	IN_DIGITAL_H	正	INPUT_ANALOG SET_INPUT_ANALOG	MODE	1	0001
	诊断	针对 IN_DIGITAL_H	SET_INPUT_MODE	DIAGNOSTICS	TRUE	
	IN_RESISTANCE	16...30 000 Ω	INPUT_ANALOG SET_INPUT_ANALOG	MODE	512	0200
I15	IN_NOMODE	关闭	INPUT_ANALOG SET_INPUT_ANALOG	MODE	0	0000
	IN_DIGITAL_H	正	INPUT_ANALOG SET_INPUT_ANALOG	MODE	1	0001
	诊断	针对 IN_DIGITAL_H	SET_INPUT_MODE	DIAGNOSTICS	TRUE	
	IN_RESISTANCE	3...680 Ω	INPUT_ANALOG SET_INPUT_ANALOG	MODE	512	0200

利用以下功能块设定工作模式：

FAST_COUNT (→ 页 171)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 173)	测量到达所选通道的信号的频率。
FREQUENCY_PERIOD (→ 页 175)	测量所示通道频率和以 [µs] 为单位的周期（周期时间）
INC_ENCODER (→ 页 178)	针对编码器评估的递增/递减计数器功能
INC_ENCODER_HR (→ 页 181)	针对编码器高分辨率评估的递增/递减计数器功能
INPUT_ANALOG (→ 页 157)	模拟输入通道： 以下对象的备选测量... • 电流 • 电压 • 电阻
PERIOD (→ 页 183)	测量所示通道频率和以 [µs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 185)	测量所示周期内所示通道频率和以 [µs] 为单位的周期（周期时间）。 此外，传号空号比以 [‰] 为单位显示。
SET_INPUT_MODE (→ 页 161)	将工作模式分配至输入通道

输出端： 工作模式（标准侧）（16 个输出端）

15523

= 该配置值为默认值

输出端	可能的工作模式		利用功能块设定	功能块输入端	值	
					十进制	十六进制
Q00...Q15	OUT_DIGITAL_H	正	SET_OUTPUT_MODE	MODE	1	0001
	OUT_DIGITAL_L	负	SET_OUTPUT_MODE	MODE	2	0002
	诊断	针对 OUT_DIGITAL_H 通过电流测量	SET_OUTPUT_MODE	DIAGNOSTICS	TRUE	
	过载保护	针对 OUT_DIGITAL_H 包含电流测量	SET_OUTPUT_MODE	PROTECTION	TRUE	
	电流测量范围	无电流测量	SET_OUTPUT_MODE	CURRENT_RANGE	0	00
		2 A/3 A	SET_OUTPUT_MODE	CURRENT_RANGE	1	01
		4 A	SET_OUTPUT_MODE	CURRENT_RANGE	2	02

详细信息 → 章节 **输出端 Q00...Q15： 允许的工作模式** (→ 页 293)

利用以下功能块设定工作模式：

OUTPUT_BRIDGE (→ 页 195)	PWM 通道对上的 H 桥
OUTPUT_CURRENT_CONTROL (→ 页 200)	PWMi 输出通道的电流控制器
PWM1000 (→ 页 203)	初始化并配置具备 PWM 功能的输出通道 传号空号比可按 1‰ 的步距表示
SET_OUTPUT_MODE (→ 页 191)	设定所选输出通道的工作模式

输出端 Q00...Q15： 允许的工作模式

15525

工作模式		Q00	Q01	Q02	Q03	Q04	Q05	Q06	Q07
OUT_DIGITAL_H	正	X	X	X	X	X	X	X	X
OUT_DIGITAL_L	负	--	X	--	X	--	--	--	--
诊断	针对 OUT_DIGITAL_H 通过电流测量	X	X	X	X	X	X	X	X
过载保护	针对 OUT_DIGITAL_H 包含电流测量	X	X	X	X	X	X	X	X
电流测量范围	2 A	X	X	X	X	--	--	--	--
	3 A	--	--	--	--	X	X	X	X
	4 A	X	X	X	X	--	--	--	--
PWM		X	X	X	X	X	X	X	X
PWMi		X	X	X	X	X	X	X	X
H 桥		--	X	--	X	--	--	--	--

工作模式		Q08	Q09	Q10	Q11	Q12	Q13	Q14	Q15
OUT_DIGITAL_H	正	X	X	X	X	X	X	X	X
OUT_DIGITAL_L	负	--	X	--	X	--	--	--	--
诊断	针对 OUT_DIGITAL_H 通过电流测量	X	X	X	X	X	X	X	X
过载保护	针对 OUT_DIGITAL_H 包含电流测量	X	X	X	X	X	X	X	X
电流测量范围	2 A	X	X	X	X	--	--	--	--
	3 A	--	--	--	--	X	X	X	X
	4 A	X	X	X	X	--	--	--	--
PWM		X	X	X	X	X	X	X	X
PWMi		X	X	X	X	X	X	X	X
H 桥		--	X	--	X	--	--	--	--

7.3 集成 I/O 模块： 说明

内容

16418

7.3.1 系统说明 I/O 模块 ExB01

内容

16422

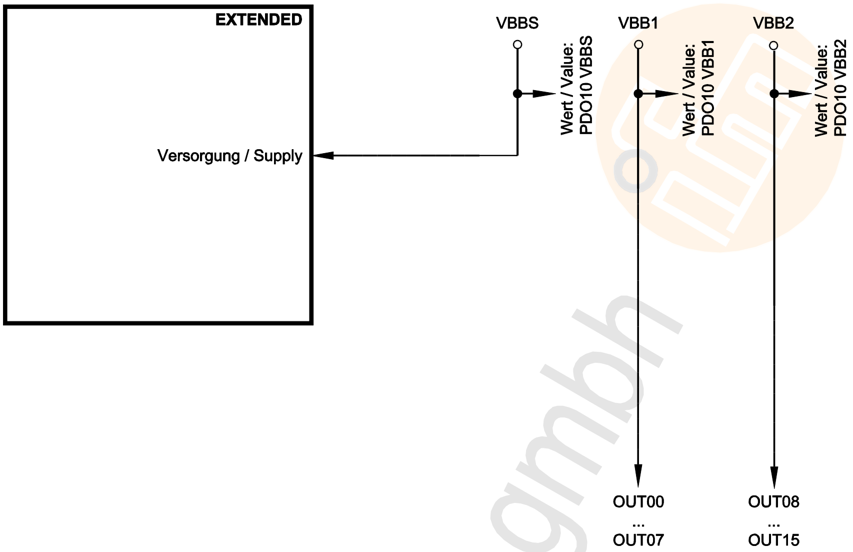
硬件说明 I/O 模块

内容

16423

硬件结构 I/O 模块

16425



图： 电源方块图

状态 LED I/O 模块

16414

集成状态 LED 指示工作状态（默认设定）。

LED 颜色	显示屏	说明
熄灭	永久熄灭	无工作电压
		
黄色	短暂亮起	Status = INIT
		（时间帧 = 200 ms）
绿色	永久亮起	状态 = PRE-OPERATIONAL
		
绿色	以 2 Hz 的频率闪烁	状态 = OPERATIONAL
		（时间帧 = 200 ms）
绿色	闪烁，1 次脉冲	状态 = STOP
		（时间帧 = 200 ms）
红色	永久亮起	错误： CAN 总线关闭
		
红色	闪烁，1 次脉冲	EMCY: CAN 错误警告
		（时间帧 = 200 ms）
红色	闪烁，2 次脉冲	EMCY: 保护/检测信号
		（时间帧 = 200 ms）
红色	闪烁，3 次脉冲	EMCY: 同步错误
		（时间帧 = 200 ms）

集成 I/O 模块 ExB01 输入端

内容

16229

模拟输入端

15444

模拟输入端可通过应用程序配置。 可按以下方式设定测量范围：

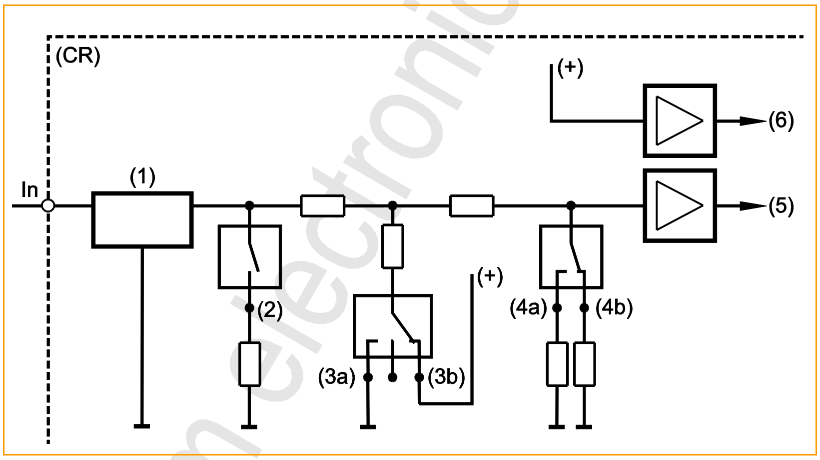
- 电流输入 0...20 mA
- 电压输入 0...10 V
- 电压输入 0...32 V
- 电阻测量 16...30 000 Ω（GND 测量）

还可进行电压比率测量（0...1000 ‰，可通过功能块调整）。 也就是说，可在没有额外参考电压的情况下评估电位计和操纵杆。 电源电压的波动不会对该测量值产生影响。

作为备选，还可通过二进制的方式评估模拟通道。

> ! 如果是比率测量，则连接的传感器应供给装置 VBBs。 因此，避免了偏移电压导致的错误测量。

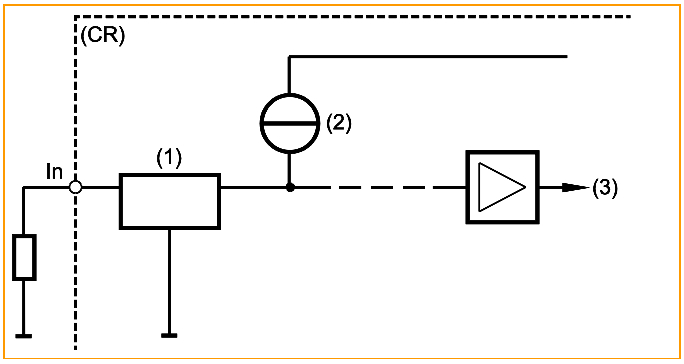
8971



In = 针形多功能输入端 n
(CR) = 装置
(1) = 输入过滤器
(2) = 模拟电流测量
(3a) = 二进制输入端正极开关
(3b) = 二进制输入端负极开关
(4a) = 模拟电压测量 0...10 V
(4b) = 模拟电压测量 0...32 V
(5) = 电压
(6) = 参考电压

图： 多功能输出端原理方块图

8972



图： 电阻器测量输入端方块图

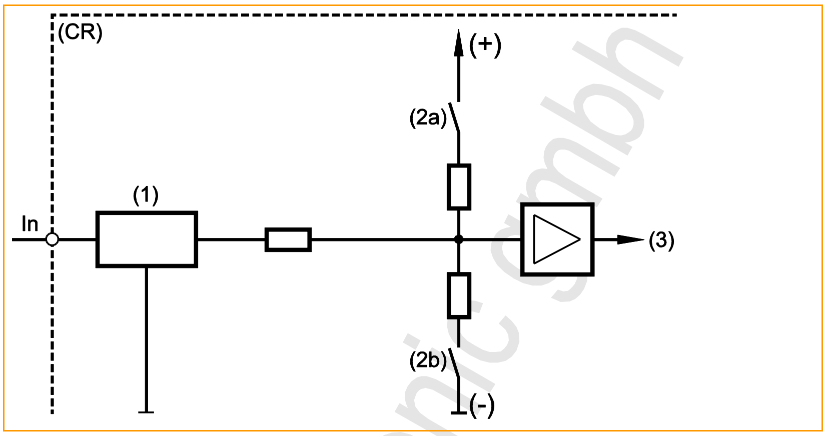
In = 针形电阻器输入端
(CR) = 装置
(1) = 输入过滤器
(2) = 恒流源
(3) = 电压

二进制输入端

1015
7345

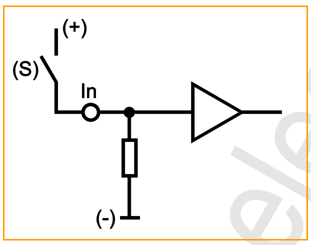
二进制输入端可在以下模式下运行：
- 二进制输入端正极开关 (BL)，针对正极性传感器信号
- 二进制输入端负极开关 (BH)，针对负极性传感器信号

根据装置，而仅是输入端可进行不同的配置。 除了针对干扰的保护机制之外，二进制输入端还通过模拟阶段进行内部评估。 这样有助于诊断输入端信号。 但在应用程序软件中，开关信可号直接用作位信息。



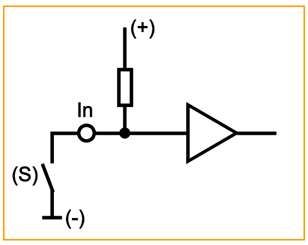
In = 针形二进制输入端 n
(CR) = 装置
(1) = 输入过滤器
(2a) = 输入端负极开关
(2b) = 输入端正极开关
(3) = 电压

图： 针对负极性和正极性传感器信号的二进制输入端负极开关/正极开关基本电路



In = 针形二进制输入端 n
(S) = 传感器

二进制输入端正极开关 (BL) 基本电路
针对正极性传感器信号：
输入端 = 开启 ⇨ 信号 = 低 (GND)



In = 针形二进制输入端 n
(S) = 传感器

二进制输入端负极开关 (BH) 基本电路
针对负极性传感器信号：
输入端 = 开启 ⇨ 信号 = 高 (电源)

就这些输入端中的某些而言（→ 技术资料），可选择要切换到的电势。

I/O 模块输入组 IN00...IN03

15801

这些输入端是一组多功能通道。

这些输入端用途如下（每个输入端均可单独配置）：

- 模拟输入端 0...20 mA
- 模拟输入端 0...10 V
- 模拟输入端 0...32 V
- 二进制输入端正极开关 (BL)，针对正极性传感器信号（含/不含诊断）
- 二进制输入端负极开关 (BH)，针对负极性传感器信号

→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

所有输入端显示关于功能和诊断的相同状况。

► 每个输入端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输入端** (→ 页 312)

> 如果针对电流测量配置模拟输入端，且如果超过最终值（23 mA 且持续 ≥ 40 ms），则装置切换至安全电压测量范围 (0...32 V DC)。 在此情况下，PDO1 发送消息"过电流"。 在大约一秒钟后，输入端自动切回至电流测量范围。

I/O 模块输入组 IN04...IN05

15803

这些输入端是一组多功能通道。

这些输入端用途如下（每个输入端均可单独配置）：

- 二进制输入端正极开关 (BL)，针对正极性传感器信号（含/不含诊断）
- 电阻测量输入端（例如温度传感器或燃油传感器）

→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

► 每个输入端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输入端** (→ 页 312)

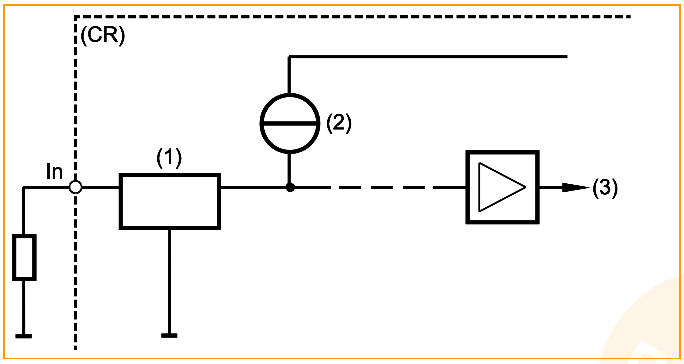
电阻测量

9773

这些输入端上的一般传感器：

- 液位
- 温度（PT1000、NTC）

8972



图： 电阻器测量输入端方块图

In = 针形电阻器输入端

(CR) = 装置

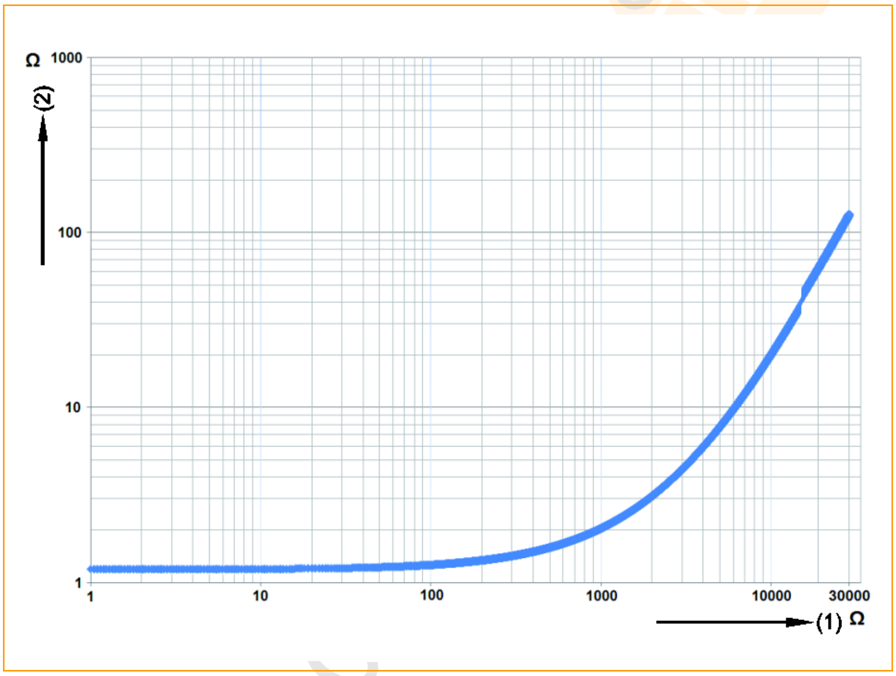
(1) = 输入过滤器

(2) = 恒流源

(3) = 电压

8970

该装置的电阻与电阻值不呈线性关系，→ 图：



图： 与电阻值相关的分辨率

(1) = 输出端电阻值

(2) = 分辨率

当输入端上的 A/D 转换器信号更改 1 个单位时，测量值更改多少 ohm？ 示例：

- 在 1...100 Ω 的范围时，分辨率为 1.2 Ω。
- 在 1 kΩ 的范围时，分辨率大约为 2 Ω。
- 在 2 kΩ 的范围时，分辨率大约为 3 Ω。
- 在 3 kΩ 的范围时，分辨率大约为 6 Ω。
- 在 6 kΩ 的范围时，分辨率大约为 10 Ω。
- 在 10 kΩ 的范围时，分辨率为 11 Ω。
- 在 20 kΩ 的范围时，分辨率大约为 60 Ω。

I/O 模块输入组 IN06...IN11

15804

这些输入端是一组多功能通道。

这些输入端用途如下（每个输入端均可单独配置）：
• 二进制输入端正极开关 (BL)，针对正极性传感器信号（含/不含诊断）
→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

可评估包含诊断功能的 NAMUR 传感器。

► 每个输入端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输入端** (→ 页 312)

I/O 模块输入组 IN12...IN15

15805

这些输入端是一组多功能通道。

这些输入端用途如下（每个输入端均可单独配置）：
• 二进制输入端正极开关 (BL)，针对正极性传感器信号
• 针对递增编码器和频率或时间间隔测量等的快速输入端
→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

► 每个输入端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输入端** (→ 页 312)

集成 I/O 模块输出端 ExB01

内容

16234

I/O 模块输出组 OUT0、OUT1

15806

这些输出端是一组多功能通道。

这些输出端提供多个功能选项（每个输出端可单独配置）：

- 二进制输出端，正极开关 (BH)，含诊断功能和保护
- 模拟电流控制输出端 (PWMi)
- 包含脉冲宽度调制 (PWM) 的模拟输出端

→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

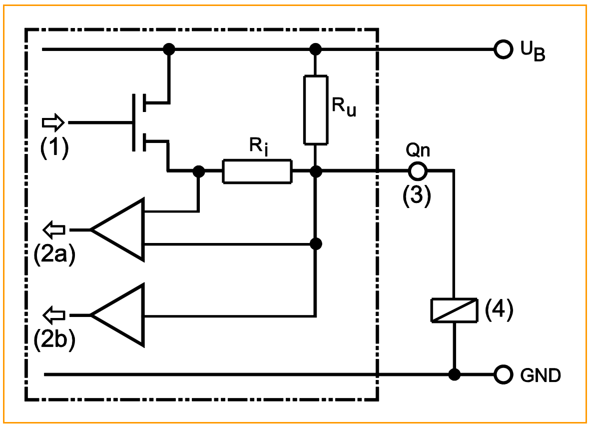
▶ 每个输出端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输出端** (→ 页 315)

▶ 限制值请务必确保遵循技术资料！

诊断： 二进制输出端（通过电流和电压测量）

19433
19434

这些输出端的诊断通过输出端的内部电流和电压测量进行：



图： 原理方块图

(1) 输出通道

(2a) 读回通道以通过电流测量诊断

(2b) 读回通道以通过电压测量诊断

(3) 插脚输出端 n

(4) 加载

诊断： 过载（通过电流测量）

19437
15249

仅可在包含电流测量的输出端检测到过载。

过载即…
“超过最大额定电流的 12.5 %”。

诊断： 断线（通过电压测量）

19436
19404

断线检测通过输出端内部的读回通道完成。

诊断的前提：	输出端 = FALSE
诊断 = 断线：	电阻器 Ru 将读回通道切换至 HIGH 电势（电源）。 若未断线，则低电阻负载 (RL < 10 kΩ) 强制切换至 LOW（逻辑 0）。

诊断： 短路（通过电压测量）

19405

断线检测通过输出端内部的读回通道完成。

诊断的前提：	输出端 = TRUE
诊断 = 对 GND 短路	读回通道切换至 LOW 电势 (GND)

I/O 模块输出组 OUT02...OUT07

15808

这些输出端是一组多功能通道。

这些输出端提供多个功能选项（每个输出端可单独配置）：
- 二进制输出端，正极开关 (BH)，含/不含诊断功能
- 包含脉冲宽度调制 (PWM) 的模拟输出端

→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

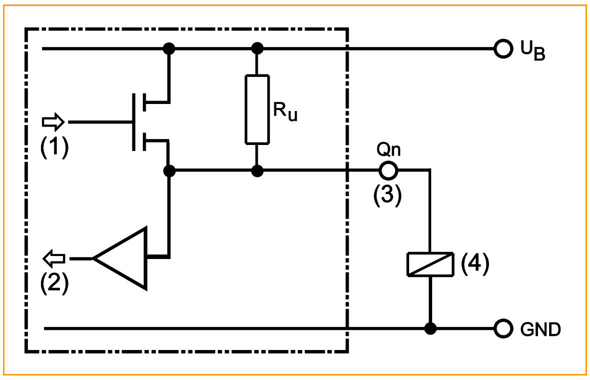
► 每个输出端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输出端** (→ 页 315)

► (!) 限制值请务必确保遵循技术资料！

诊断： 二进制输出端（通过电压测量）

19403
19397

这些输出端的诊断通过输出端的内部电压测量进行：



图： 原理方块图

(1) 输出通道

(2) 针对诊断的读回通道

(3) 插脚输出端 n

(4) 负载

诊断： 过载

19448

输出端无电流测量和过载检测。

诊断： 断线（通过电压测量）

19404

断线检测通过输出端内部的读回通道完成。

诊断的前提：	输出端 = FALSE
诊断 = 断线：	电阻器 Ru 将读回通道切换至 HIGH 电势（电源）。 若未断线，则低电阻负载 (RL < 10 kΩ) 强制切换至 LOW（逻辑 0）。

诊断： 短路（通过电压测量）

19405

断线检测通过输出端内部的读回通道完成。

诊断的前提：	输出端 = TRUE
诊断 = 对 GND 短路	读回通道切换至 LOW 电势 (GND)

I/O 模块输出组 OUT08...OUT09

15809

这些输出端是一组多功能通道。

这些输出端提供多个功能选项（每个输出端可单独配置）：
- 二进制输出端，正极开关 (BH)
- 包含脉冲宽度调制 (PWM) 的模拟输出端
- 包含脉冲宽度调制 (PWM) 的模拟输出端，电压控制

→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

► 每个输出端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输出端** (→ 页 315)

► 限制值请务必确保遵循技术资料！

I/O 模块输出组 OUT10...OUT11

15810

这些输出端是一组多功能通道。

这些输出端提供多个功能选项（每个输出端可单独配置）：
- 二进制输出端，正极开关 (BH)
- 包含脉冲宽度调制 (PWM) 的模拟输出端

→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

► 每个输出端均可通过 PLC 配置进行配置：
→ 章节 **配置集成 I/O 模块的输出端** (→ 页 315)

► 限制值请务必确保遵循技术资料！

I/O 模块输出组 OUT12...OUT15

15811

这些输出端是包含单一特定功能的通道组。

这些输出端具有以下固定设定：
- 二进制输出端，正极开关 (BH)

→ 章节 **I/O 模块可能的工作模式** (→ 页 319)

► 限制值请务必确保遵循技术资料！

接口说明 I/O 模块

内容

16426

CAN 接口 I/O 模块

内容

16608

连接和数据 → 技术资料

CAN: 接口和协议： CR0133 中的 I/O 模块

15833
15835

以下 CAN 接口和 CAN 协议可用于装置的集成 I/O 模块：

CAN 接口	CAN 1	CAN 2	CAN 3	CAN 4
默认下载 ID	ID 123	ID 122	---	---
CAN 协议	---	CANopen 从站	---	---

标准波特率 = 125 Kbits/s

CAN: 接口和协议： CR2532 中的 I/O 模块

16429
16435

以下 CAN 接口和 CAN 协议可用于装置的集成 I/O 模块：

CAN 接口	CAN 1	CAN 2	CAN 3	CAN 4
默认下载 ID	ID 125	ID 124	---	---
CAN 协议	---	CANopen 从站	---	---

标准波特率 = 250 Kbits/s

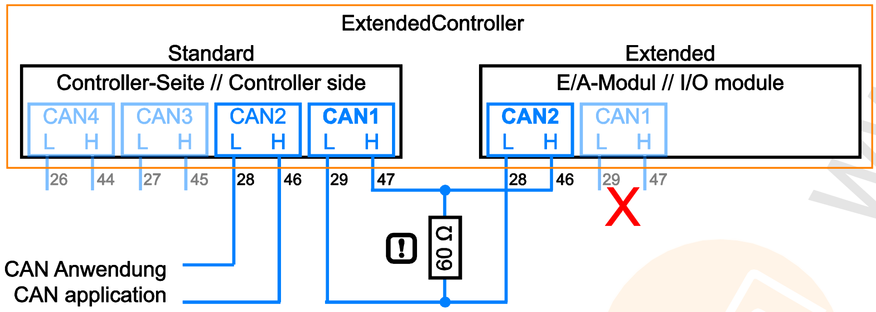
连接集成 I/O 模块 ExB01 以作为 CANopen 从站

15829

装置的集成 I/O 模块以 Smart Controller CR2530 为基础：

- 该侧预设为 CANopen 从站 ExB01
- 将该侧作为 I/O 模块。

建议采用以下连接方法：



- I/O 模块的 CAN1 仅作为检修或维护接口！

► 仅针对集成 I/O 模块使用针对控制器标准侧的所示连接！
切勿将该等连接用于其他目的！

► 就应用程序中的 CAN 网络而言，仅使用标准侧 ≥ CAN2 接口！

7.3.2 I/O 模块的配置

内容

16427

设定编程系统（I/O 模块）

内容

16609

手动设定编程系统（I/O 模块）

内容

16610

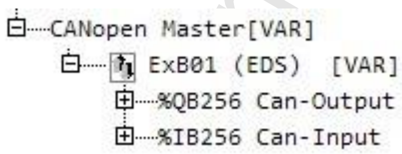
集成内部 I/O 模块 ExB01

15828

❗通过 CODESYS 控制配置集成作为 CANopen 从站的装置的内部 I/O 模块！
为此，集成外部 I/O 模块 采用相同的方法：

► 左击 CODESYS 控制配置中的第一行 (`CR0033 Configuration Vxx`)，以高亮显示。
► 单击鼠标右键即可打开上下文菜单。
► 在该位置选择 [`Append Subelement`]。
► 在上下文菜单中选择 [`CANopen master...`]。
   ℹ 建议针对 CAN1 配置第一个 CANopen 主站。
► 右击 [`CANopen master`] 以再次打开上下文菜单。
► 在该位置选择 [`Append Subelement`]。
► 在上下文菜单中选择针对装置集成 I/O 模块的 EDS 文件：
   [`ExB01_Vxxyyzz.EDS`]。

\> 结果：



❗CAN-Input 和 CAN-Output 的 IEC 地址通过以下详情确定：
- 作为 CANopen 主站的装置的类型，
- I/O 模块在 CANopen 背后的位置，
- 分配的节点 ID。

❗I/O 模块使用三个连续节点 ID。 规则：
⇨ [node ID of the following CAN slave] ≥ [node ID of the I/O module] + 3

► 设定 CAN 参数：
- 节点 ID
- 节点保护
- 检测信号设定

► I/O 模块输入端和输出端的参数设定：
   → 章节 **集成 I/O 模块的对象目录** (→ 页 325)

通过模板设定编程系统（I/O 模块）

16611
13745

IFM 提供即用型模板（程序模板），利用该模板可轻松、快速和充分地设定编程系统。

970

安装 ecomat*mobile* DVD“软件、工具和文档”时，包含模板的项目已存储于您的 PC 的程序目录：

`…\ifm electronic\CoDeSys V…\Projects\Template_DVD_V…`

► 通过以下路径在 CODESYS 中打开所需模板：
[File] > [New from template…]

> CODESYS 创建新项目并显示基本程序结构。 强烈建议遵循所示的步骤。

I/O 模块输入端和输出端的功能配置

内容

16430

配置集成 I/O 模块的输入端

内容

16244

配置输入端的软件过滤器（I/O 模块）

15898

软件过滤器预设且不可更改：

表格： 模拟输入端软件低通滤波器极限频率

FILTER	过滤频率 [Hz]	针对以下范围的阶跃响应 [ms]			备注
		0...70 %	0...90 %	0...99 %	
固定	10	19	36	72	

模拟输入端： 配置和诊断（I/O 模块 ExB01）

15894

每个输入端均可通过 PLC 配置进行配置：

- 点击 [CANopen Master] 下方的 行 [ExB01 (EDS)]
- 点击 [Service Data Objects] 选项卡
- 选择所需参数的索引/子索引
- 单击 [Value] 列中的现有值
- 更改值并通过 [ENTER] 确认

允许值 → 章节 **输入端： 工作模式（I/O 模块）** (→ 页 322)

> 如果针对电流测量配置模拟输入端，且如果超过最终值（23 mA 且持续 ≥ 40 ms），则装置切换至安全电压测量范围 (0...32 V DC)。 在此情况下，PDO1 发送消息“过电流”。 在大约一秒钟后，输入端自动切回至电流测量范围。

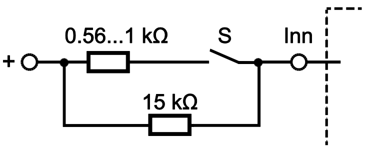
二进制输入端： 配置和诊断（I/O 模块 ExB01）

15896

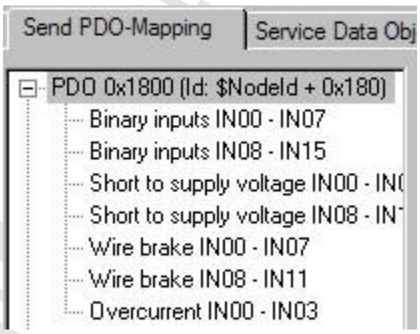
► 每个输入端均可通过 PLC 配置进行配置：

- 点击 [CANopen Master] 下方的 行 [ExB01 (EDS)]
- 点击 [Service Data Objects] 选项卡
- 选择所需参数的索引/子索引
- 单击 [Value] 列中的现有值
- 更改值并通过 [ENTER] 确认

允许值 → 章节 **输入端： 工作模式（I/O 模块）** (→ 页 322)

非电子开关二进制信号 NAMUR 诊断： ► 为开关配置额外的电阻器连接件！	 图： 输入端 Inn 非电子开关 S

> 诊断结果为 DO 1



快速输入端：I/O 模块 ExB01

15869

装置处理输入频率高达 30 kHz 的快速计算/脉冲输入端（→ 数据表）。

❗ 例如，如果机械开关连接至这些输入端，则控制器中可能因为触点弹跳而出现故障信号。

每个输入端均可通过 PLC 配置进行配置：

允许值 → 章节 **输入端：工作模式（I/O 模块）** (→ 页 322)

3804

允许的高输入频率还可确保检测到故障信号，例如机械开关弹跳的触点。

► 如有必要，在应用程序中解决故障信号！

配置集成 I/O 模块的输出端

内容

16248

配置输出端的软件过滤器（I/O 模块）

15900

对于 I/O 模块，以下要求适用：

软件过滤器预设且不可更改。

表格： PWM 输出端软件低通滤波器极限频率

FILTER	过滤频率 [Hz]	针对以下范围的阶跃响应 [ms]			备注
		0...90 %	0...95 %	0...99 %	
固定	52	7.2	9.4	14.4	

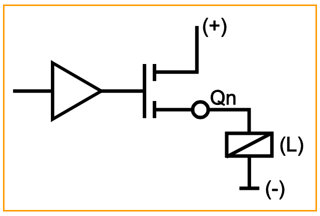
二进制输出端： 配置和诊断（I/O 模块 ExB01）

15882

以下工作模式适用于装置输出端（(→ 技术资料）：

• 二进制输出端，正极开关 (BH)，含/不含诊断功能

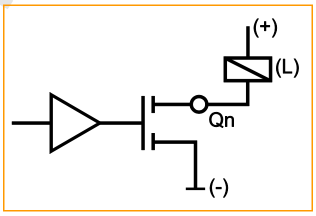
15450



Qn = 插脚输出端 n

(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号



Qn = 插脚输出端 n

(L) = 负载

输出端负极开关 (BL) 基本电路
针对负极性输出端信号

13975

⚠ **警告**

可能存在危险的重启！

存在人身伤害的风险！ 存在机器/设备材料损坏的风险！

如果在故障情况下输出端通过硬件关闭，应用程序生成的逻辑状态不会改变。

► 补救措施：
  • 复位应用程序中的输出逻辑！
  • 排除故障！
  • 根据状况复位输出端。

二进制输出端： 配置 (I/O 模块 ExB01)

15887

► 每个输出端均可通过 PLC 配置进行配置：

- 点击 [CANopen Master] 下方的 行 [ExB01 (EDS)]
- 点击 [Service Data Objects] 选项卡
- 选择所需参数的索引/子索引
- 单击 [Value] 列中的现有值
- 更改值并通过 [ENTER] 确认

允许值 → 章节 **输出端： 工作模式 (I/O 模块)** (→ 页 323)

二进制输出端： 诊断 (I/O 模块 ExB01)

15889

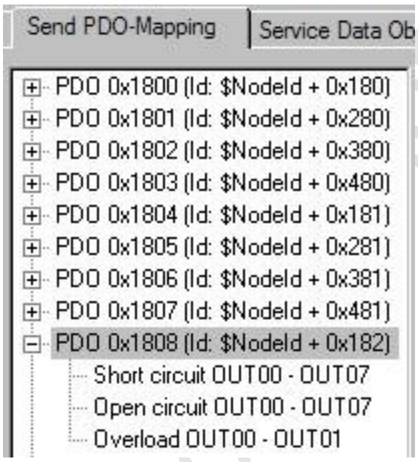
► 每个输出端均可通过 PLC 配置进行配置：

通过以下方式启用诊断…

- 模式 = 15 (OUT_BINARY_HIGH_DIAG) 或
- Mode = 16 (OUT_BINARY_HIGH_DIAG_PROT)

允许值 → 章节 **输出端： 工作模式 (I/O 模块)** (→ 页 323)

\> 结果显示 PDO 9：

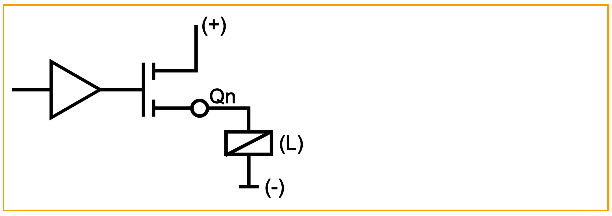


PWM 输出端： I/O 模块 ExB01

16415

以下工作模式适用于装置输出端 ((→ 技术资料) ：
• PWM 输出端，正极开关 (BH)，不含诊断功能

15451



Qn = 插脚输出端 n
(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号

16253

⚠ 警告

可能因故障导致财产损失或人身伤害！

针对 PWM 模式下的输出端：
• 无诊断功能

9980

注意

PWM 输出端不可并联运行，以便增加最大输出电流等。 输出端不同步运行。

否则整体负载电流可能仅流经一个输出端。 电流测量将不再正常进行。

- PWM 可在包含和不含电流控制功能的情况下运行。
  电流控制 PWM 输出端主要用于触发比例液压功能。
  PWM 信号中的中度电流仅可在开启状态下的电流处于测量范围内时正确确定。

PWM 可用性

16364

装置	可用 PWM 输出端的数量	电流控制 (PWMi) 数量	PWM 频率 [Hz]
集成 I/O 模块 ExB01	12	2	20...250

配置针对 PWM 功能的输出端

15888

以下设定适用于输出端的 PWM 功能：
• 模式 = 4 (OUT_PWM) 或
• Mode = 5 (OUT_CURRENT)
允许值 → 章节 **输出端： 工作模式 (I/O 模块)** (→ 页 323)

通过 PWM 控制电流 (= PWMi)

14722

可通过控制器中集成的电流测量通道对线圈电流展开测量。 这样，如果线圈升温，则可重新调整电流，等等。 系统保持液压状况。

原则上，电流控制输出端有短路保护。

I/O 模块可能的工作模式

内容

16440

概述

15859

就输入端和输出端而言，可采用以下工作模式（详细说明： → 后续页面）：

SDO 值 十进制 | 十六进制		Mode	输入端 输出端	说明
0	0x00	关闭	输入端 输出端	关闭，无功能
1	0x01	IN_BINARY_LOW_DIGITAL	输入端	二进制正极开关，数字检测 （如有可能，换用模式 10！）
2	0x02	OUT_BINARY_HIGH	输出端	二进制正极开关： 输出端 = FALSE ⇨ 0 V 输出端 = TRUE ⇨输出端电源电压
3	0x03	IN_VOLTAGE_10V	输入端	10 V 测量范围的模拟电压测量
4	0x04	OUT_PWM	输出端	PWM 模式
5	0x05	OUT_CURRENT	输出端	电流控制
6	0x06	IN_VOLTAGE_RATIO	输入端	模拟电压测量，针对电源电压 VBBs 的比率
7	0x07	IN_CURRENT	输入端	模拟电流测量（高达 23 mA）
8	0x08	---	---	已保留
9	0x09	IN_VOLTAGE_32	输入端	32 V 测量范围的模拟电压测量
10	0x0A	IN_BINARY_LOW	输入端	二进制正极开关（取决于电源电压 VBBs） （模拟或数字检测）
11	0x0B	IN_BINARY_LOW_DIAG	输入端	二进制正极开关，包含诊断（模拟检测） 取决于电源电压 VBBs VBBs 或 GND 短路诊断
12	0x0C	IN_BINARY_HIGH	输入端	二进制负极开关（模拟检测） 取决于电源电压 VBBs
13	0x0D	---	---	已保留
14	0x0E	IN_FREQUENCY	输入端	频率测量（数字检测）
15	0x0F	OUT_BINARY_HIGH_DIAG	输出端	二进制正极开关： 输出端 = FALSE ⇨ 0 V 输出端 = TRUE ⇨输出端电源电压 断线和短路诊断
16	0x10	OUT_BINARY_HIGH_DIAG_PROT	输出端	二进制正极开关： 输出端 = FALSE ⇨ 0 V 输出端 = TRUE ⇨输出端电源电压 断线和短路诊断 发生短路时输出端关闭。

SDO 值 十进制 | 十六进制		Mode	输入端 输出端	说明
17	0x11	---	---	已保留
18	0x12	IN_RESISTOR	输入端	电阻测量（模拟检测）
19	0x13	---	---	已保留
20	0x14	IN_PERIOD_RATIO	输入端	周期比率测量（数字检测）
21	0x15	---	---	已保留
22	0x16	---	---	已保留
23	0x17	---	---	已保留
24	0x18	---	---	已保留

输入端： 工作模式（I/O 模块）

15965

► 每个输入端均可通过 PLC 配置进行配置：
- 点击 [CANopen Master] 下方的 行 [ExB01 (EDS)]
- 点击 [Service Data Objects] 选项卡
- 选择所需参数的索引/子索引
- 单击 [Value] 列中的现有值
- 更改值并通过 [ENTER] 确认

= 该配置值为默认值

输入端	可能的工作模式		对象目录中索引	子索引	值	
					十进制	十六进制
IN00...IN03	关闭		0x2000	0x01...0x04	0	0x00
	电压输入端	0...10 000 mV	0x2000	0x01...0x04	3	0x03
	电压输入比率	0...1 000 ‰	0x2000	0x01...0x04	6	0x06
	电流输入端	0...20 000 µA	0x2000	0x01...0x04	7	0x07
	电压输入端	0...32 000 mV	0x2000	0x01...0x04	9	0x09
	二进制输入端	正极开关	0x2000	0x01...0x04	10	0x0A
	二进制输入端，含诊断 (Namur)	正极开关	0x2000	0x01...0x04	11	0x0B
	二进制输入端	负极开关	0x2000	0x01...0x04	12	0x0C
IN04...IN05	关闭		0x2000	0x05...0x06	0	0x00
	二进制输入端	正极开关	0x2000	0x05...0x06	10	0x0A
	二进制输入端，含诊断 (Namur)	正极开关	0x2000	0x05...0x06	11	0x0B
	电阻输入端	16...30 000 Ohm	0x2000	0x05...0x06	18	0x12
IN06...IN11	关闭		0x2000	0x07...0x0C	0	0x00
	二进制输入端	正极开关	0x2000	0x07...0x0C	10	0x0A
	二进制输入端，含诊断 (Namur)	正极开关	0x2000	0x07...0x0C	11	0x0B
IN12...IN15	关闭		0x2000	0x0D...0x10	0	0x00
	二进制输入端，数字评估	正极开关	0x2000	0x0D...0x10	1	0x01
	频率测量	0...30 000 Hz	0x2000	0x0D...0x10	14	0x0E
	周期和比率测量	0.1...3 000 Hz	0x2000	0x0D...0x10	20	0x14

输出端： 工作模式（I/O 模块）

15966

► 每个输出端均可通过 PLC 配置进行配置：

- 点击 [CANopen Master] 下方的 行 [ExB01 (EDS)]
- 点击 [Service Data Objects] 选项卡
- 选择所需参数的索引/子索引
- 单击 [Value] 列中的现有值
- 更改值并通过 [ENTER] 确认

(绿色) = 该配置值为默认值

输出端	可能的工作模式		对象目录中索引	子索引	值	
					十进制	十六进制
OUT00 ...OUT01	关闭		0x2000	0x11...0x12	0	0x00
	二进制输出端高侧	正	0x2000	0x11...0x12	2	0x02
	模拟输出端，含脉冲宽度调制		0x2000	0x11...0x12	4	0x04
	模拟电流控制输出		0x2000	0x11...0x12	5	0x05
	二进制输出端高侧，含诊断	正	0x2000	0x11...0x12	15	0x0F
	二进制输出端高侧，含诊断和保护	正	0x2000	0x11...0x12	16	0x10
OUT02 ...OUT07	关闭		0x2000	0x13...0x18	0	0x00
	二进制输出端高侧	正	0x2000	0x13...0x18	2	0x02
	模拟输出端，含脉冲宽度调制		0x2000	0x13...0x18	4	0x04
	二进制输出端高侧，含诊断	正	0x2000	0x13...0x18	15	0x0F
	二进制输出端高侧，含诊断和保护	正	0x2000	0x13...0x18	16	0x10
OUT08 ...OUT09	关闭		0x2000	0x19...0x1A	0	0x00
	二进制输出端高侧	正	0x2000	0x19...0x1A	2	0x02
	模拟输出端，含脉冲宽度调制		0x2000	0x19...0x1A	4	0x04
	模拟输出端，含脉冲宽度调制，电压控制	（位于插脚 25 + 43）	0x2000	0x19...0x1A	4	0x04

输出端	可能的工作模式		对象目录中索引	子索引	值	
					十进制	十六进制
OUT10 ...OUT11	关闭		0x2000	0x1B...0x1C	0	0x00
	二进制输出端高侧	正	0x2000	0x1B...0x1C	2	0x02
	模拟输出端，含脉冲宽度调制		0x2000	0x1B...0x1C	4	0x04
OUT12 ...OUT15	关闭		0x2000	0x1D...0x20	0	0x00
	二进制输出端高侧	正	0x2000	0x1D...0x20	2	0x02

7.3.3 集成 I/O 模块的对象目录

内容

15837

对象目录参数表格，概述

内容

15977

概要

15967

► 通过条目 [Save Parameter] 启用或禁用通信和装置参数的自动保存（→ 对象目录，索引 0x1010）：

- 如果子索引 0x1 = 0x02：
  更改时自动保存所有参数
- 如果子索引 0x1 = 0x00：
  不自动保存参数。
  仅更改的参数有效。
  - 直至装置关闭
  - 直至下次复位。

► 通过功能 [Restore] 还原预设的参数值（→ 对象目录，索引 0x1011）（不适用于波特率和节点 ID）。 下一次通电时，这些值会生效。

EDS 文件中的数据类型

16409

EDS 数据类型	IEC 数据类型	最小值	最大值	内存大小
	BOOL	FALSE	TRUE	8 位 = 1 个字节
	BYTE	0	255	8 位 = 1 个字节
	WORD	0	65 535	16 位 = 2 个字节
	DWORD	0	4 294 967 295	32 位 = 4 个字节
	SINT	-128	127	8 位 = 1 个字节
0x0005	USINT	0	255	8 位 = 1 个字节
0x0003	INT	-32 768	32 767	16 位 = 2 个字节
0x0006	UINT	0	65 535	16 位 = 2 个字节
	DINT	-2 147 483 648	2 147 483 647	32 位 = 4 个字节
0x0007	UDINT	0	4 294 967 295	32 位 = 4 个字节
0x0008	REAL	-3.402823466 • 1038	3.402823466 • 1038	32 位 = 4 个字节
	ULINT	0	18 446 744 073 709 551 615	64 位 = 8 个字节
0x0009	STRING			字符数量 + 1

对象目录强制对象（索引 0x1000...0x1FFF），概述

15979

对象目录		参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
索引	子索引					
0x1000		装置类型	装置	0xF0191	是	立刻（通过 CAN 堆栈）
0x1001		错误寄存器	装置	---	是	立刻（通过 CAN 堆栈）
0x1018		装置识别	装置	---	--	--
	0x1	供应商 ID	装置	6907501	是	制造后
	0x2	产品代码	装置	0	是	制造后
	0x3	修订号	装置	0	是	制造后
	0x4	序列号	装置	0	是	制造后

对象目录可选对象（索引 0x1000...0x1FFF），概述

15980

对象目录 索引	对象目录 子索引	参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
0x1003	0x1...0x5	预定义错误字段	CANopen 基本配置	0	是	立刻（通过 CAN 堆栈）
0x1005		COB ID 同步消息	CANopen 基本配置	0x80	是	立刻（通过 CAN 堆栈）
0x1006		通信周期	CANopen 基本配置	0	是	立刻
0x1008		制造商装置名称	CANopen 基本配置	ExB01	是	立刻
0x1009		制造商硬件版本	CANopen 基本配置	V00.00.00	是	立刻
0x100A		制造商软件版本	CANopen 基本配置	V00.00.00	是	立刻
0x100C		保护时间	CANopen 基本配置	0	是	立刻
0x100D		使用期限系数	CANopen 基本配置	0	是	立刻
0x1010		存储参数	CANopen 基本配置		是	立刻
	0x1	保存所有参数	CANopen 基本配置	1	是	立刻
0x1011		恢复默认参数	CANopen 基本配置		否	复位后
	0x1	恢复所有默认参数	CANopen 基本配置	1	否	复位后
0x1014		COB ID 紧急	CANopen 基本配置	0x80 + 节点 ID	是	立刻
0x1016		用户检测信号时间	CANopen 基本配置		--	--
	0x1	用户检测信号时间	CANopen 基本配置	0	是	立刻
0 1017		生产商检测信号时间	CANopen 基本配置	0	是	立刻
0 1400		接收 PDO 通信参数	配置 接收 PDO 1		--	--
	0x1	PDO 使用的 COB ID	配置 接收 PDO 1	0x0200 + 节点 ID	是	预运行后

对象目录		参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
索引	子索引					
	0x2	传送类型	配置 接收 PDO 1	1	是	立刻
0x1401		接收 PDO 通信参数	配置 接收 PDO 2		--	--
	0x1	PDO 使用的 COB ID	配置 接收 PDO 2	0x0300 + 节点 ID	是	预运行后
	0x2	传送类型	配置 接收 PDO 2	1	是	立刻
0x1402		接收 PDO 通信参数	配置 接收 PDO 3		--	--
	0x1	PDO 使用的 COB ID	配置 接收 PDO 3	0x0400 + 节点 ID	是	预运行后
	0x2	传送类型	配置 接收 PDO 3	1	是	立刻
0x1403		接收 PDO 通信参数	配置 接收 PDO 4		--	--
	0x1	PDO 使用的 COB ID	配置 接收 PDO 4	0x0500 + 节点 ID	是	预运行后
	0x2	传送类型	配置 接收 PDO 4	1	是	立刻
0x1600		接收 PDO 映射	映射 接收 PDO 1		是	预运行后
	0x1	PDO 映射	映射 接收 PDO 1	0x6200 0108	是	预运行后
	0x2	PDO 映射	映射 接收 PDO 1	0x6200 0208	是	预运行后
0x1601		接收 PDO 映射	映射 接收 PDO 2		是	预运行后
	0x1	PDO 映射	映射 接收 PDO 2	0x6414 0110	是	预运行后
	0x2	PDO 映射	映射 接收 PDO 2	0x6414 0210	是	预运行后
	0x3	PDO 映射	映射 接收 PDO 2	0x6414 0310	是	预运行后
	0x4	PDO 映射	映射 接收 PDO 2	0x6414 0410	是	预运行后

对象目录 索引	子索引	参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
0x1602		接收 PDO 映射	映射 接收 PDO 3		是	预运行后
	0x1	PDO 映射	映射 接收 PDO 3	0x6414 0510	是	预运行后
	0x2	PDO 映射	映射 接收 PDO 3	0x6414 0610	是	预运行后
	0x3	PDO 映射	映射 接收 PDO 3	0x6414 0710	是	预运行后
	0x4	PDO 映射	映射 接收 PDO 3	0x6414 0810	是	预运行后
0x1603		接收 PDO 映射	映射 接收 PDO 4		是	预运行后
	0x1	PDO 映射	映射 接收 PDO 4	0x6414 0910	是	预运行后
	0x2	PDO 映射	映射 接收 PDO 4	0x6414 0A10	是	预运行后
	0x3	PDO 映射	映射 接收 PDO 4	0x6414 0B10	是	预运行后
	0x4	PDO 映射	映射 接收 PDO 4	0x6414 0C10	是	预运行后
	0x5	PDO 映射	映射 接收 PDO 4	0x00	是	预运行后
0x1800		传送 PDO 通信参数	配置 传送 PDO 1		--	--
	0x1	PDO 使用的 COB ID	配置 传送 PDO 1	0x180 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 1	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 1	0	是	立刻
	0x4	已保留	配置 传送 PDO 1	0	否	--
	0x5	事件时间	配置 传送 PDO 1	0	是	立刻
0x1801		传送 PDO 通信参数	配置 传送 PDO 2		--	--

对象目录		参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
索引	子索引					
	0x1	PDO 使用的 COB ID	配置 传送 PDO 2	0x280 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 2	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 2	0	是	立刻
	0x4	已保留	配置 传送 PDO 2	0	否	--
	0x5	事件时间	配置 传送 PDO 2	0	是	立刻
0x1802		传送 PDO 通信参数	配置 传送 PDO 3		--	--
	0x1	PDO 使用的 COB ID	配置 传送 PDO 3	0x380 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 3	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 3	0	是	立刻
	0x4	已保留	配置 传送 PDO 3	0	否	--
	0x5	事件时间	配置 传送 PDO 3	0	是	立刻
0x1803		传送 PDO 通信参数	配置 传送 PDO 4		--	--
	0x1	PDO 使用的 COB ID	配置 传送 PDO 4	0x480 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 4	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 4	0	是	立刻
	0x4	已保留	配置 传送 PDO 4	0	否	--
	0x5	事件时间	配置 传送 PDO 4	0	是	立刻
0x1804		传送 PDO 通信参数	配置 传送 PDO 5		--	--

对象目录		参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
索引	子索引					
	0x1	PDO 使用的 COB ID	配置 传送 PDO 5	0x181 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 5	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 5	0	是	立刻
	0x4	已保留	配置 传送 PDO 5	0	否	--
	0x5	事件时间	配置 传送 PDO 5	0	是	立刻
0x1805		传送 PDO 通信参数	配置 传送 PDO 6		--	--
	0x1	PDO 使用的 COB ID	配置 传送 PDO 6	0x281 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 6	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 6	0	是	立刻
	0x4	已保留	配置 传送 PDO 6	0	否	--
	0x5	事件时间	配置 传送 PDO 6	0	是	立刻
0x1806		传送 PDO 通信参数	配置 传送 PDO 7		--	--
	0x1	PDO 使用的 COB ID	配置 传送 PDO 7	0x381 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 7	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 7	0	是	立刻
	0x4	已保留	配置 传送 PDO 7	0	否	--
	0x5	事件时间	配置 传送 PDO 7	0	是	立刻
0x1807		传送 PDO 通信参数	配置 传送 PDO 8		--	--

对象目录 索引	子索引	参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
	0x1	PDO 使用的 COB ID	配置 传送 PDO 8	0x481 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 8	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 8	0	是	立刻
	0x4	已保留	配置 传送 PDO 8	0	否	--
	0x5	事件时间	配置 传送 PDO 8	0	是	立刻
0x1808		传送 PDO 通信参数	配置 传送 PDO 9		--	--
	0x1	PDO 使用的 COB ID	配置 传送 PDO 9	0x182 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 9	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 9	0	是	立刻
	0x4	已保留	配置 传送 PDO 9	0	否	--
	0x5	事件时间	配置 传送 PDO 9	0	是	立刻
0x1809		传送 PDO 通信参数	配置 传送 PDO 10		--	--
	0x1	PDO 使用的 COB ID	配置 传送 PDO 10	0x282 + 节点 ID	--	--
	0x2	传送类型	配置 传送 PDO 10	1	是	立刻
	0x3	禁止时间	配置 传送 PDO 10	0	是	立刻
	0x4	已保留	配置 传送 PDO 10	0	否	--
	0x5	事件时间	配置 传送 PDO 10	0	是	立刻
0x1A00		传送 PDO 映射	映射 传送 PDO 1		是	预运行后

对象目录 索引	子索引	参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
	0x1	PDO 映射	映射 传送 PDO 1	0x6000 0108	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 1	0x6000 0208	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 1	0x2020 0108	是	预运行后
	0x4	PDO 映射	映射 传送 PDO 1	0x2020 0208	是	预运行后
	0x5	PDO 映射	映射 传送 PDO 1	0x2021 0108	是	预运行后
	0x6	PDO 映射	映射 传送 PDO 1	0x2021 0208	是	预运行后
	0x7	PDO 映射	映射 传送 PDO 1	0x2025 0108	是	预运行后
0x1A01		传送 PDO 映射	映射 传送 PDO 2		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 2	0x6404 0110	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 2	0x6404 0210	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 2	0x6404 0310	是	预运行后
	0x4	PDO 映射	映射 传送 PDO 2	0x6404 0410	是	预运行后
0x1A02		传送 PDO 映射	映射 传送 PDO 3		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 3	0x2030 0110	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 3	0x2030 0210	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 3	0x2002 0110	是	预运行后
	0x4	PDO 映射	映射 传送 PDO 3	0x2002 0210	是	预运行后
	0x5	PDO 映射	映射 传送 PDO 3	0	是	预运行后

对象目录 索引	子索引	参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
0x1A03		传送 PDO 映射	映射 传送 PDO 4		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 4	0x2012 0120	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 4	0x2012 0220	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 4	0	是	预运行后
0x1A04		传送 PDO 映射	映射 传送 PDO 5		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 5	0x2012 0320	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 5	0x2012 0420	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 5	0	是	预运行后
0x1A05		传送 PDO 映射	映射 传送 PDO 6		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 6	0x2014 0110	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 6	0x2014 0210	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 6	0x2014 0310	是	预运行后
	0x4	PDO 映射	映射 传送 PDO 6	0x2014 0410	是	预运行后
	0x5	PDO 映射	映射 传送 PDO 6	0	是	预运行后
0x1A06		传送 PDO 映射	映射 传送 PDO 7		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 7	0x2015 0120	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 7	0x2015 0220	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 7	0	是	预运行后

对象目录 索引	子索引	参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
0x1A07		传送 PDO 映射	映射 传送 PDO 8		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 8	0x2015 0320	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 8	0x2015 0420	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 8	0	是	预运行后
0x1A08		传送 PDO 映射	映射 传送 PDO 9		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 9	0x2022 0108	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 9	0x2023 0108	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 9	0x2024 0108	是	预运行后
	0x4	PDO 映射	映射 传送 PDO 9	0	是	预运行后
0x1A09		传送 PDO 映射	映射 传送 PDO 10		是	预运行后
	0x1	PDO 映射	映射 传送 PDO 10	0x2040 0110	是	预运行后
	0x2	PDO 映射	映射 传送 PDO 10	0x2041 0110	是	预运行后
	0x3	PDO 映射	映射 传送 PDO 10	0x2041 0210	是	预运行后
	0x4	PDO 映射	映射 传送 PDO 10	0x2050 0010	是	预运行后
	0x5	PDO 映射	映射 传送 PDO 10	0	是	预运行后

使用期限系数 0 被视为 1。

第一个保护协议被视为“启动保护”，即使当时保护尚未启动（保护时间为 ＝0）。

对象目录制造商特定对象（索引 0x2000...0x6FFF），概述

15978

对象目录索引	参数说明	参数针对的对象	预设参数值	更改自动保存？	更改何时生效？
0x2000	I/O 配置	IN00…IN11 IN12…IN15	10 01	是	预运行后
0x2001	PWM 频率	OUT00…OUT11	100	是	预运行后
0x2002	电流值	OUT00…OUT01	0	是	预运行后
0x2004	P 值	OUT00…OUT01	30	是	预运行后
0x2005	I 值	OUT00…OUT01	20	是	预运行后
0x2006	PWM 抖动频率	OUT00…OUT11	0	是	预运行后
0x2007	PWM 抖动值	OUT00…OUT11	0	是	预运行后
0x2012	输入端周期	IN12…IN15	0	是	预运行后
0x2013	周期数量	IN12…IN15	0	是	预运行后
0x2014	周期比率值	IN12…IN15	0	是	预运行后
0x2015	输入端频率	IN12…IN15	0.0	是	预运行后
0x2016	时间基数频率	IN12…IN15	50	是	预运行后
0x2020	输入端 VBBS 短路	IN00…IN11	0	是	预运行后
0x2021	输入端断线	IN00…IN11	0	是	预运行后
0x2022	输出端短路	OUT00…OUT07	0	是	预运行后
0x2023	输出端开路	OUT00…OUT07	0	是	预运行后
0x2024	输出端过载	OUT00…OUT01	0	是	预运行后
0x2025	输入端过电流	IN00…IN03	0	是	预运行后
0x2030	输入电阻	IN04…IN05	0	是	预运行后
0x2040	电源电压	VBBS	0	是	预运行后
0x2041	电源电压	VBB1, VBB2	0	是	预运行后
0x2050	装置温度	装置	0	是	预运行后
0x20F0 != 0x20F1 *)	节点 ID	装置	124	如果两者相同	复位后
0x20F2 != 0x20F3 *)	波特率	装置	3	如果两者相同	复位后
0x20F4	自动启动	装置	0	是	立刻
0x6000	二进制输入端	IN00…IN07 IN08…IN15	0	是	预运行后
0x6200	二进制输出端	OUT00…OUT07 OUT08…OUT15	0	是	预运行后
0x6404	模拟输入端	IN00…IN03	---	---	---
0x6414	模拟输出端	OUT00…OUT11	---	---	---

*) 值必须相同！

对象目录参数表格，详细信息

内容

15982

对象目录强制对象（索引 0x1000...0x1FFF），详细信息

15985

索引	S-Idx	参数名称	数据类型		默认	详细信息
0x1000		装置类型	ro	UDINT	0x000F 0191	装置类型
0x1001		错误寄存器	ro	USINT	0	错误寄存器数位编码至配置文件 301 允许值 = 0b0000 0000 = 无错误 0b0000 0001 = 一般错误 0b0001 0000 = 通信错误 0b1000 0000 = 制造商特定错误
0x1018	0x0	装置 ID 条目数量	ro	USINT	0x04	装置识别
	0x1	供应商 ID	ro	UDINT	0x0690 7501	基于 CiA 规范的供应商 ID
	0x2	产品代码	ro	STRING	0	装置的产品代码
	0x3	修订号	ro	UDINT	0	装置的修订号
	0x4	序列号	ro	UDINT	0	装置的序列号

说明：
数据类型： ro = 只读 / rw = 读写 / wo = 只写

对象目录可选对象（索引 0x1000...0x10FFF），详细信息

16603

索引	S-Idx	参数名称		数据类型	默认	详细信息
0x1003	0x0	预定义错误字段 条目数量	rw	UDINT	0	支持包含 4 个条目的错误列表
	0x1	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码 子索引 1 表示最后一个错误
	0x2	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码
	0x3	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码
	0x4	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码
	0x5	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码
0x1005		COB-ID 同步消息	rw	UDINT	0x0000 0080	同步消息标识符 第 30 位 = 0 ⇨ 装置未生成同步消息 第 30 位 = 1 ⇨ 装置生成一条同步消息 第 29 位 = 0 ⇨ 11 位 ID 第 29 位 = 1 ⇨ ID = 0x80 + 节点 ID
0x1006		通信周期	rw	UDINT	0	2 个同步对象之间的最长时间，以 [µs] 为单位 控制分辨率 = 1 ms
0x1008		制造商装置名称	ro	STRING	EXB01	装置名称
0x1009		制造商硬件版本	ro	STRING	V00.00.00	硬件版本
0x100A		制造商软件版本	ro	STRING	V00.00.00	软件版本
0x100C		保护时间	rw	UINT	0	在以 [ms] 为单位的该时间内，预计装置对系统主站进行“节点保护”。 0 = 不支持该功能 通过“节点保护”或“检测信号”监控节点仅作为备选方案。
0x100D		使用期限系数	rw	USINT	0	如果“保护时间”•“使用期限”未受到“节点保护”，则装置关闭输出端。 装置将 CANopen 状态切换至 PREOP。 默认： “保护时间”•“使用期限”= 0...65535
0x1010	0x0	存储参数 受支持的最大子索引	ro	USINT	0x01	“保存选项”数量
	0x1	保存所有参数	rw	UDINT	2	自动保存所有已更改的参数 0 = 自动保存关闭 2 = 自动保存开启
0x1011	0x0	恢复默认参数 受支持的最大子索引	ro	USINT	0x01	“恢复选项”数量

索引	S-Idx	参数名称		数据类型	默认	详细信息
	0x1	恢复所有默认参数	rw	UDINT	0x01	如果在此输入字符串“负载”，则恢复出厂设定的默认参数，并在下次复位时生效。
0x1014		COBId 紧急	rw	UDINT	0x80 + 节点 ID	第 31 位 = 0 ⇨ EMCY 有效 第 31 位 = 1 ⇨EMCY 无效 第 29 位 = 0 ⇨ 11 位 ID 第 29 位 = 1 ⇨ ID = 0x80 + 节点 ID 用户可更改 CAN ID。
0x1016	0x0	用户检测信号时间 用户检测信号时间数量	ro	USINT	0x01	节点检测信号监控时间 已检测装置数量 = 1
	0x1	用户检测信号时间	rw	UDINT	0	节点检测信号监控时间 格式： 0x0nntttt tttt = 监控时间 [ms] nn = 节点数量 如果 nn=0 或 tttt=0 ⇨，则无监控 通过“节点保护”或“检测信号”监控节点仅作为备选方案。
0x1017		生产商检测信号时间	rw	UINT	0	装置生成生产商检测信号的时间间隔 [ms]

说明：
数据类型： ro = 只读 / rw = 读写 / wo = 只写

对象目录可选对象（索引 0x1400...0x14FF），详细信息

16604

接收 PDO 通信参数

索引	S-Idx	参数名称		数据类型	默认	详细信息
0x1400	0x0	接收 PDO 通信参数 条目数量	ro	USINT	0x02	接收 PDO 1： 二进制输出端 条目数量 = 2
	0x1	PDO 使用的 COBID	rw	UDINT	0x200 + 节点 ID	第一个读取的 PDO 的 CAN-ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 输出端仅在 "n" 个同步对象之后更新。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 输出端立即更新 0xFF = 异步装置配置文件事件； 输出端立即更新
0x1401	0x0	接收 PDO 通信参数 条目数量	ro	USINT	0x02	接收 PDO 2： PWM 输出端 条目数量 = 2
	0x1	PDO 使用的 COBID	rw	UDINT	0x300 + 节点 ID	第一个读取的 PDO 的 CAN-ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 输出端仅在 "n" 个同步对象之后更新。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 输出端立即更新 0xFF = 异步装置配置文件事件； 输出端立即更新
0x1402	0x0	接收 PDO 通信参数 条目数量	ro	USINT	0x02	接收 PDO 3： PWM 输出端 条目数量 = 2
	0x1	PDO 使用的 COBID	rw	UDINT	0x400 + 节点 ID	第一个读取的 PDO 的 CAN-ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效

索引	S-Idx	参数名称	数据类型		默认	详细信息
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 输出端仅在 "n" 个同步对象之后更新。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 输出端立即更新 0xFF = 异步装置配置文件事件； 输出端立即更新
0x1403	0x0	接收 PDO 通信参数 条目数量	ro	USINT	0x02	接收 PDO 4： PWM 输出端 条目数量 = 2
	0x1	PDO 使用的 COBID	rw	UDINT	0x500 + 节点 ID	第一个读取的 PDO 的 CAN-ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 输出端仅在 "n" 个同步对象之后更新。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 输出端立即更新 0xFF = 异步装置配置文件事件； 输出端立即更新

说明：
数据类型： ro = 只读 / rw = 读写 / wo = 只写

对象目录可选对象（索引 0x1600...0x16FF），详细信息

16605

接收 PDO 映射

索引	S-Idx	参数名称		数据类型	默认	详细信息
0x1600	0x0	接收 PDO 映射 PDO 映射对象的数量	rw	USINT	0x02	映射读取 PDO 1： 二进制输出端 集成应用程序对象数量 = 2

	0x1	PDO 映射	ro	UDINT	0x6200 0108	索引 0x6200 子索引 01 中的 1 个字节
						二进制输入端 IN00...IN07
						`0b---- ---X = IN00`
						`0b---- --X- = IN01`
						`0b---- -X-- = IN02`
						`0b---- X--- = IN03`
						`0b---X ---- = IN04`
						`0b--X- ---- = IN05`
						`0b-X-- ---- = IN06`
						`0bX--- ---- = IN07`

	0x2	PDO 映射	ro	UDINT	0x6200 0208	索引 0x6200 子索引 02 中的 1 个字节
						二进制输入端 IN08...IN15
						`0b---- ---X = IN08`
						`0b---- --X- = IN09`
						`0b---- -X-- = IN10`
						`0b---- X--- = IN11`
						`0b---X ---- = IN12`
						`0b--X- ---- = IN13`
						`0b-X-- ---- = IN14`
						`0bX--- ---- = IN15`

0x1601	0x0	接收 PDO 映射 PDO 映射对象的数量	rw	USINT	0x04	映射读取 PDO 2： PWM 输出端 OUT00...OUT03 集成应用程序对象数量 = 4
	0x1	PDO 映射	rw	UDINT	0x6414 0110	PWM 输出端 OUT00 索引 0x6414 子索引 0x1 包含 PWM 输出端 OUT00 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x2	PDO 映射	rw	UDINT	0x6414 0210	PWM 输出端 OUT01 索引 0x6414 子索引 0x2 包含 PWM 输出端 OUT01 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x3	PDO 映射	rw	UDINT	0x6414 0310	PWM 输出端 OUT02 索引 0x6414 子索引 0x3 包含 PWM 输出端 OUT02 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x4	PDO 映射	rw	UDINT	0x6414 0410	PWM 输出端 OUT03 索引 0x6414 子索引 0x4 包含 PWM 输出端 OUT03 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。

索引	S-Idx	参数名称		数据类型	默认	详细信息
0x1602	0x0	接收 PDO 映射 PDO 映射对象的数量	rw	USINT	0x04	映射读取 PDO 3： PWM 输出端 OUT04...OUT07 集成应用程序对象数量 = 4
	0x1	PDO 映射	rw	UDINT	0x6414 0510	PWM 输出端 OUT04 索引 0x6414 子索引 0x5 包含 PWM 输出端 OUT04 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x2	PDO 映射	rw	UDINT	0x6414 0610	PWM 输出端 OUT05 索引 0x6414 子索引 0x6 包含 PWM 输出端 OUT05 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x3	PDO 映射	rw	UDINT	0x6414 0710	PWM 输出端 OUT06 索引 0x6414 子索引 0x7 包含 PWM 输出端 OUT06 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x4	PDO 映射	rw	UDINT	0x6414 0810	PWM 输出端 OUT07 索引 0x6414 子索引 0x8 包含 PWM 输出端 OUT07 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x5	PDO 映射	rw	UDINT	0	保留
0x1603	0x0	接收 PDO 映射 PDO 映射对象的数量	rw	USINT	0x04	映射读取 PDO 4： PWM 输出端 OUT08...OUT11 集成应用程序对象数量 = 4
	0x1	PDO 映射	rw	UDINT	0x6414 0910	PWM 输出端 OUT08 索引 0x6414 子索引 0x9 包含 PWM 输出端 OUT08 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x2	PDO 映射	rw	UDINT	0x6414 0A10	PWM 输出端 OUT09 索引 0x6414 子索引 0xA 包含 PWM 输出端 OUT09 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x3	PDO 映射	rw	UDINT	0x6414 0B10	PWM 输出端 OUT10 索引 0x6414 子索引 0xB 包含 PWM 输出端 OUT10 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x4	PDO 映射	rw	UDINT	0x6414 0C10	PWM 输出端 OUT11 索引 0x6414 子索引 0xC 包含 PWM 输出端 OUT11 的预设值，值被视为以 ‰ 为单位的忙闲度或目标电流值（取决于配置索引 0x2000）。
	0x5	PDO 映射	rw	UDINT	0	保留

说明：
数据类型： ro = 只读 / rw = 读写 / wo = 只写

对象目录可选对象（索引 0x1800...0x18FF），详细信息

16606

传送 PDO 通信参数

索引	S-Idx	参数名称	数据类型		默认	详细信息
0x1800	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 1 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x180 + 节点 ID	传送 PDO 1 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型"异步"的延迟时间。 (0...65535 • 100 µs)
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型"异步"的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1801	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 2 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x280 + 节点 ID	传送 PDO 2 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型"异步"的延迟时间。 (0...65535 • 100 µs)

索引	S-Idx	参数名称	数据类型		默认	详细信息
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型“异步”的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1802	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 3 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x380 + 节点 ID	传送 PDO 3 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型“异步”的延迟时间。 (0...65535 • 100 µs)
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型“异步”的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1803	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 4 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x480 + 节点 ID	传送 PDO 4 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型“异步”的延迟时间。 (0...65535 • 100 µs)

索引	S-Idx	参数名称		数据类型	默认	详细信息
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型"异步"的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1804	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 5 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x181 + 节点 ID	传送 PDO 5 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型"异步"的延迟时间。 (0...65535 • 100 µs)
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型"异步"的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1805	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 6 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x281 + 节点 ID	传送 PDO 6 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型"异步"的延迟时间。 (0...65535 • 100 µs)

索引	S-Idx	参数名称	数据类型		默认	详细信息
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型“异步”的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1806	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 7 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x381 + 节点 ID	传送 PDO 7 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型“异步”的延迟时间。 (0...65535 • 100 µs)
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型“异步”的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1807	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 8 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x481 + 节点 ID	传送 PDO 8 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型“异步”的延迟时间。 (0...65535 • 100 µs)

索引	S-Idx	参数名称		数据类型	默认	详细信息
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型"异步"的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1808	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 9 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x181 + 节点 ID	传送 PDO 9 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型"异步"的延迟时间。 (0...65535 • 100 µs)
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型"异步"的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。
0x1809	0x0	传送 PDO 通信参数 条目数量	ro	USINT	0x05	配置传送 PDO 10 条目数量 = 5
	0x1	PDO 使用的 COBID	rw	UDINT	0x281 + 节点 ID	传送 PDO 10 的 CAN ID 第 31 位 = 0 ⇨ PDO 有效 第 31 位 = 1 ⇨ PDO 无效
	0x2	传送类型	rw	USINT	0x01	0x00 = 同步非循环 0x01...0xF0 = 同步循环； 值仅在 "n" 个同步对象之后传送。 n = 1...240 = 0x01...0xF0 0xFC/0xFD 未实施 0xFE = 异步手动指定事件； 值立即传输 0xFF = 异步装置配置文件事件； 值立即传输
	0x3	禁止时间	rw	UINT	0	PDO 最早再次传送之前传送类型"异步"的延迟时间。 (0...65535 • 100 µs)

索引	S-Idx	参数名称	数据类型		默认	详细信息
	0x4	已保留	rw	USINT	0	保留
	0x5	事件时间	rw	UINT	0	传送类型“异步”的最长传输间隔时间 (0...65535 ms) 当该时间结束时传输 PDO，即便未发生应用程序事件。

说明：
数据类型： ro = 只读 / rw = 读写 / wo = 只写

对象目录可选对象（索引 0x1A00...0x1AFF），详细信息

16607

传送 PDO 映射

索引	S-Idx	参数名称	数据类型		默认	详细信息
0x1A00	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x07	映射传送 PDO 1 集成应用程序对象数量 = 7
	0x1	PDO 映射	rw	UDINT	0x6000 0108	索引 0x6000 子索引 0x1 二进制输入端 00...07： 实际值（数位编码）
	0x2	PDO 映射	rw	UDINT	0x6000 0208	索引 0x6000 子索引 0x2 二进制输入端 08...15： 实际值（数位编码）
	0x3	PDO 映射	rw	UDINT	0x2020 0108	索引 0x2020 子索引 0x1 输入端 00...07： 标志"短路"（数位编码）
	0x4	PDO 映射	rw	UDINT	0x2020 0208	索引 0x2020 子索引 0x2 输入端 08...11： 标志"短路"（数位编码）
	0x5	PDO 映射	rw	UDINT	0x2021 0108	索引 0x2021 子索引 0x1 输入端 00...07： 标志"断线"（数位编码）
	0x6	PDO 映射	rw	UDINT	0x2021 0208	索引 0x2021 子索引 0x2 输入端 08...11： 标志"断线"（数位编码）
	0x7	PDO 映射	rw	UDINT	0x2025 0108	索引 0x2025 子索引 0x1 输入端 00...03： 标志"过载"（数位编码）
0x1A01	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x04	映射传送 PDO 2（模拟输入端） 集成应用程序对象数量 = 4
	0x1	PDO 映射	rw	UDINT	0x6404 0110	索引 0x6404 子索引 0x1 模拟输入端 00： 实际值 （取决于配置 0x2000）
	0x2	PDO 映射	rw	UDINT	0x6404 0210	索引 0x6404 子索引 0x2 模拟输入端 01： 实际值 （取决于配置 0x2000）
	0x3	PDO 映射	rw	UDINT	0x6404 0310	索引 0x6404 子索引 0x3 模拟输入端 02： 实际值 （取决于配置 0x2000）
	0x4	PDO 映射	rw	UDINT	0x6404 0410	索引 0x6404 子索引 0x4 模拟输入端 03： 实际值 （取决于配置 0x2000）
0x1A02	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x04	映射传送 PDO 3 集成应用程序对象数量 = 4
	0x1	PDO 映射	rw	UDINT	0x2030 0110	索引 0x2030 子索引 0x1 输入端 04： 实际电阻器值

索引	S-Idx	参数名称		数据类型	默认	详细信息
	0x2	PDO 映射	rw	UDINT	0x2030 0210	索引 0x2030 子索引 0x2 输入端 05： 实际电阻器值
	0x3	PDO 映射	rw	UDINT	0x2002 0110	索引 0x2002 子索引 0x1 输出端 00： 实际电流值
	0x4	PDO 映射	rw	UDINT	0x2002 0210	索引 0x2002 子索引 0x2 输出端 01： 实际电流值
	0x5	PDO 映射	rw	UDINT	0	保留
0x1A03	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x02	映射传送 PDO 4（周期时间 IN12...IN13） 集成应用程序对象数量 = 2
	0x1	PDO 映射	rw	UDINT	0x2012 0120	索引 0x2012 子索引 0x1 频率输入端 12： 信号周期时间
	0x2	PDO 映射	rw	UDINT	0x2012 0220	索引 0x2012 子索引 0x2 频率输入端 13： 信号周期时间
	0x3	PDO 映射	rw	UDINT	0	保留
0x1A04	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x02	映射传送 PDO 5（周期时间 IN14...IN15） 集成应用程序对象数量 = 2
	0x1	PDO 映射	rw	UDINT	0x2012 0320	索引 0x2012 子索引 0x3 频率输入端 14： 信号周期时间
	0x2	PDO 映射	rw	UDINT	0x2012 0420	索引 0x2012 子索引 0x4 频率输入端 15： 信号周期时间
	0x3	PDO 映射	rw	UDINT	0	保留
0x1A05	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x04	映射传送 PDO 6（频率输入端 IN12...IN15 信号忙闲度） 集成应用程序对象数量 = 4
	0x1	PDO 映射	rw	UDINT	0x2014 0110	索引 0x2014 子索引 0x1 频率输入端 12： 信号忙闲度，以 ‰ 为单位
	0x2	PDO 映射	rw	UDINT	0x2014 0210	索引 0x2014 子索引 0x2 频率输入端 13： 信号忙闲度，以 ‰ 为单位
	0x3	PDO 映射	rw	UDINT	0x2014 0310	索引 0x2014 子索引 0x3 频率输入端 14： 信号忙闲度，以 ‰ 为单位
	0x4	PDO 映射	rw	UDINT	0x2014 0410	索引 0x2014 子索引 0x4 频率输入端 15： 信号忙闲度，以 ‰ 为单位
	0x5	PDO 映射	rw	UDINT	0	保留
0x1A06	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x02	映射传送 PDO 7（IN12...IN13 上的频率） 集成应用程序对象数量 = 2
	0x1	PDO 映射	rw	UDINT	0x2015 0120	索引 0x2015 子索引 0x1 频率输入端 12： 信号频率值，以 Hz 为单位

索引	S-Idx	参数名称		数据类型	默认	详细信息
	0x2	PDO 映射	rw	UDINT	0x2015 0220	索引 0x2015 子索引 0x2 频率输入端 13： 信号频率值，以 Hz 为单位
	0x3	PDO 映射	rw	UDINT	0	保留
0x1A07	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x02	映射传送 PDO 8（IN14...IN15 上的频率） 集成应用程序对象数量 = 2
	0x1	PDO 映射	rw	UDINT	0x2015 0320	索引 0x2015 子索引 0x3 频率输入端 14： 信号频率值，以 Hz 为单位
	0x2	PDO 映射	rw	UDINT	0x2015 0420	索引 0x2015 子索引 0x4 频率输入端 15： 信号频率值，以 Hz 为单位
	0x3	PDO 映射	rw	UDINT	0	保留
0x1A08	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x03	映射传送 PDO 9（错误标志 OUT00...OUT07） 集成应用程序对象数量 = 3
	0x1	PDO 映射	rw	UDINT	0x2022 0108	索引 0x2022 子索引 0x1 OUT00...OUT07: 标志“短路”（数位编码）
	0x2	PDO 映射	rw	UDINT	0x2023 0108	索引 0x2023 子索引 0x1 OUT00...OUT07: 标志“断线”（数位编码）
	0x3	PDO 映射	rw	UDINT	0x2024 0108	索引 0x2024 子索引 0x1 OUT00...OUT01: 标志“过载”（数位编码）
	0x4	PDO 映射	rw	UDINT	0	保留
0x1A09	0x0	传送 PDO 映射 PDO 映射对象的数量	rw	USINT	0x04	映射传送 PDO 10（系统标志） 集成应用程序对象数量 = 4
	0x1	PDO 映射	rw	UDINT	0x2040 0110	索引 0x2040 子索引 0x1 系统电源电压 VBBS
	0x2	PDO 映射	rw	UDINT	0x2041 0110	索引 0x2041 子索引 0x1 输出端电源电压 VBB1
	0x3	PDO 映射	rw	UDINT	0x2041 0210	索引 0x2041 子索引 0x2 输出端电源电压 VBB2
	0x4	PDO 映射	rw	UDINT	0x2050 0010	索引 0x2050 子索引 0x0 系统温度，以 °C 为单位
	0x5	PDO 映射	rw	UDINT	0	保留

说明：
数据类型： ro = 只读 / rw = 读写 / wo = 只写

对象目录制造商特定对象（索引 0x2000...0x6FFF），详细信息

15983

索引	S-Idx	参数名称	数据类型		默认	详细信息	
0x2000	0x0	IO 配置 受支持的最大子索引	ro	USINT	32	配置输入端/输出端 最大受支持子索引 = 32	
	0x1	配置 IN00	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	关闭 输入端 IN00 0...10 000 mV 比率 0...1000 ‰ 0...20 000 µA 0...32 000 mV 二进制正极开关 二进制正极开关，含诊断 二进制负极开关
	0x2	配置 IN01	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	关闭 输入端 IN01 0...10 000 mV 比率 0...1000 ‰ 0...20 000 µA 0...32 000 mV 二进制正极开关 二进制正极开关，含诊断 二进制负极开关
	0x3	配置 IN02	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	关闭 输入端 IN02 0...10 000 mV 比率 0...1000 ‰ 0...20 000 µA 0...32 000 mV 二进制正极开关 二进制正极开关，含诊断 二进制负极开关
	0x4	配置 IN03	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	关闭 输入端 IN03 0...10 000 mV 比率 0...1000 ‰ 0...20 000 µA 0...32 000 mV 二进制正极开关 二进制正极开关，含诊断 二进制负极开关
0x2000	0x5	配置 IN04	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B 18 = 0x12	关闭 输入端 IN04 二进制正极开关 二进制正极开关，含诊断 16…30 000 Ohm
	0x6	配置 IN05	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B 18 = 0x12	关闭 输入端 IN05 二进制正极开关 二进制正极开关，含诊断 16…30 000 Ohm
0x2000	0x7	配置 IN06	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	关闭 输入端 IN06 二进制正极开关 二进制正极开关，含诊断

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	0x8	配置 IN07	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	关闭 输入端 IN07 二进制正极开关 二进制正极开关，含诊断
	0x9	配置 IN08	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	关闭 输入端 IN08 二进制正极开关 二进制正极开关，含诊断
	0xA	配置 IN09	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	关闭 输入端 IN09 二进制正极开关 二进制正极开关，含诊断
	0xB	配置 IN10	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	关闭 输入端 IN10 二进制正极开关 二进制正极开关，含诊断
	0xC	配置 IN11	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	关闭 输入端 IN11 二进制正极开关 二进制正极开关，含诊断
0x2000	0xD	配置 IN12	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	关闭 输入端 IN12 二进制正极开关，数字监控 频率 0...30 000 Hz 周期 周期比率 0...1 000 ‰ 向上计数 向下计数 增量编码器
	0xE	配置 IN13	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	关闭 输入端 IN13 二进制正极开关，数字监控 频率 0...30 000 Hz 周期 周期比率 0...1 000 ‰ 向上计数 向下计数 增量编码器
	0xF	配置 IN14	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	关闭 输入端 IN14 二进制正极开关，数字监控 频率 0...30 000 Hz 周期 周期比率 0...1 000 ‰ 向上计数 向下计数 增量编码器

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	0x10	配置 IN15	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	关闭 输入端 IN15 二进制正极开关，数字监控 频率 0...30 000 Hz 周期 周期比率 0...1 000 ‰ 向上计数 向下计数 增量编码器
0x2000	0x11	配置 OUT00	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 5 = 0x05 15 = 0x0F 16 = 0x10	关闭 输入端 OUT00 二进制正极开关 PWM 输出端 电流控制 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护
	0x12	配置 OUT01	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 5 = 0x05 15 = 0x0F 16 = 0x10	关闭 输入端 OUT01 二进制正极开关 PWM 输出端 电流控制 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护
0x2000	0x13	配置 OUT02	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	关闭 输入端 OUT02 二进制正极开关 PWM 输出端 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护
	0x14	配置 OUT03	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	关闭 输入端 OUT03 二进制正极开关 PWM 输出端 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护
	0x15	配置 OUT04	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	关闭 输入端 OUT04 二进制正极开关 PWM 输出端 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护
	0x16	配置 OUT05	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	关闭 输入端 OUT05 二进制正极开关 PWM 输出端 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	0x17	配置 OUT06	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	关闭 输入端 OUT06 二进制正极开关 PWM 输出端 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护
	0x18	配置 OUT07	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	关闭 输入端 OUT07 二进制正极开关 PWM 输出端 二进制正极开关，含诊断 二进制正极开关，含诊断 + 保护
0x2000	0x19	配置 OUT08	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	关闭 输入端 OUT08 二进制正极开关 PWM 输出端 + PWM 输出端，电压控制
	0x1A	配置 OUT09	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	关闭 输入端 OUT09 二进制正极开关 PWM 输出端 + PWM 输出端，电压控制
0x2000	0x1B	配置 OUT10	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	关闭 输入端 OUT10 二进制正极开关 PWM 输出端
	0x1C	配置 OUT11	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	关闭 输入端 OUT11 二进制正极开关 PWM 输出端
0x2000	0x1D	配置 OUT12	rw	USINT	2	0 = 0x00 2 = 0x02	关闭 输入端 OUT12 二进制正极开关
	0x1E	配置 OUT13	rw	USINT	2	0 = 0x00 2 = 0x02	关闭 输入端 OUT13 二进制正极开关
	0x1F	配置 OUT14	rw	USINT	2	0 = 0x00 2 = 0x02	关闭 输入端 OUT14 二进制正极开关
	0x20	配置 OUT15	rw	USINT	2	0 = 0x00 2 = 0x02	关闭 输入端 OUT15 二进制正极开关
0x2001	0x0	PWM 频率	ro	USINT	12	受支持的最大子索引	
	0x1	PWM 频率 OUT00	rw	UINT	100	20…250	OUT00 PWM 频率 [Hz]
	0x2	PWM 频率 OUT01	rw	UINT	100	20…250	OUT01 PWM 频率 [Hz]
	0x3	PWM 频率 OUT02	rw	UINT	100	20…250	OUT02 PWM 频率 [Hz]
	0x4	PWM 频率 OUT03	rw	UINT	100	20…250	OUT03 PWM 频率 [Hz]
	0x5	PWM 频率 OUT04	rw	UINT	100	20…250	OUT04 PWM 频率 [Hz]
	0x6	PWM 频率 OUT05	rw	UINT	100	20…250	OUT05 PWM 频率 [Hz]
	0x7	PWM 频率 OUT06	rw	UINT	100	20…250	OUT06 PWM 频率 [Hz]

索引	S-Idx	参数名称		数据类型	默认	详细信息	
	0x8	PWM 频率 OUT07	rw	UINT	100	20…250	OUT07 PWM 频率 [Hz]
	0x9	PWM 频率 OUT08	rw	UINT	100	20…250	OUT08 PWM 频率 [Hz]
	0xA	PWM 频率 OUT09	rw	UINT	100	20…250	OUT09 PWM 频率 [Hz]
	0xB	PWM 频率 OUT10	rw	UINT	100	20…250	OUT10 PWM 频率 [Hz]
	0xC	PWM 频率 OUT11	rw	UINT	100	20…250	OUT11 PWM 频率 [Hz]
0x2002	0x0	电流值	ro	USINT	2	受支持的最大子索引	
	0x1	电流值 OUT00	ro	UINT	0	0…2 000	OUT00 输出电流 [mA]
	0x2	电流值 OUT01	ro	UINT	0	0…2 000	OUT01 输出电流 [mA]
0x2004	0x0	P 值	ro	USINT	2	受支持的最大子索引	
	0x1	P 值 OUT00	rw	USINT	30	0…255	电流控制 OUT00 P 值
	0x2	P 值 OUT01	rw	USINT	30	0…255	电流控制 OUT01 P 值
0x2005	0x0	I 值	ro	USINT	2	受支持的最大子索引	
	0x1	I 值 OUT00	rw	USINT	20	0…255	电流控制 OUT00 I 值
	0x2	I 值 OUT01	rw	USINT	20	0…255	电流控制 OUT01 I 值
0x2006	0x0	PWM 抖动频率	ro	USINT	12	受支持的最大子索引	
	0x1	PWM 抖动频率 OUT00	rw	UINT	0	0…PWMfreq / 2	OUT00 PWM 抖动频率 [Hz]
	0x2	PWM 抖动频率 OUT01	rw	UINT	0	0…PWMfreq / 2	OUT01 PWM 抖动频率 [Hz]
	0x3	PWM 抖动频率 OUT02	rw	UINT	0	0…PWMfreq / 2	OUT02 PWM 抖动频率 [Hz]
	0x4	PWM 抖动频率 OUT03	rw	UINT	0	0…PWMfreq / 2	OUT03 PWM 抖动频率 [Hz]
	0x5	PWM 抖动频率 OUT04	rw	UINT	0	0…PWMfreq / 2	OUT04 PWM 抖动频率 [Hz]
	0x6	PWM 抖动频率 OUT05	rw	UINT	0	0…PWMfreq / 2	OUT05 PWM 抖动频率 [Hz]
	0x7	PWM 抖动频率 OUT06	rw	UINT	0	0…PWMfreq / 2	OUT06 PWM 抖动频率 [Hz]
	0x8	PWM 抖动频率 OUT07	rw	UINT	0	0…PWMfreq / 2	OUT07 PWM 抖动频率 [Hz]
	0x9	PWM 抖动频率 OUT08	rw	UINT	0	0…PWMfreq / 2	OUT08 PWM 抖动频率 [Hz]
	0xA	PWM 抖动频率 OUT09	rw	UINT	0	0…PWMfreq / 2	OUT09 PWM 抖动频率 [Hz]
	0xB	PWM 抖动频率 OUT10	rw	UINT	0	0…PWMfreq / 2	OUT10 PWM 抖动频率 [Hz]
	0xC	PWM 抖动频率 OUT11	rw	UINT	0	0…PWMfreq / 2	OUT11 PWM 抖动频率 [Hz]
0x2007	0x0	PWM 抖动值	ro	USINT	12	受支持的最大子索引	
	0x1	PWM 抖动值 OUT00	rw	UINT	0	0…1 000	OUT00 PWM 抖动值 [‰]
	0x2	PWM 抖动值 OUT01	rw	UINT	0	0…1 000	OUT01 PWM 抖动值 [‰]
	0x3	PWM 抖动值 OUT02	rw	UINT	0	0…1 000	OUT02 PWM 抖动值 [‰]
	0x4	PWM 抖动值 OUT03	rw	UINT	0	0…1 000	OUT03 PWM 抖动值 [‰]

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	0x5	PWM 抖动值 OUT04	rw	UINT	0	0...1 000	OUT04 PWM 抖动值 [‰]
	0x6	PWM 抖动值 OUT05	rw	UINT	0	0...1 000	OUT05 PWM 抖动值 [‰]
	0x7	PWM 抖动值 OUT06	rw	UINT	0	0...1 000	OUT06 PWM 抖动值 [‰]
	0x8	PWM 抖动值 OUT07	rw	UINT	0	0...1 000	OUT07 PWM 抖动值 [‰]
	0x9	PWM 抖动值 OUT08	rw	UINT	0	0...1 000	OUT08 PWM 抖动值 [‰]
	0xA	PWM 抖动值 OUT09	rw	UINT	0	0...1 000	OUT09 PWM 抖动值 [‰]
	0xB	PWM 抖动值 OUT10	rw	UINT	0	0...1 000	OUT10 PWM 抖动值 [‰]
	0xC	PWM 抖动值 OUT11	rw	UINT	0	0...1 000	OUT11 PWM 抖动值 [‰]
0x2012	0x0	周期输入端	ro	USINT	4	受支持的最大子索引	
	0x1	周期 IN12	ro	UDINT	0	IN12 周期 [µs]	
	0x2	周期 IN13	ro	UDINT	0	IN13 周期 [µs]	
	0x3	周期 IN14	ro	UDINT	0	IN14 周期 [µs]	
	0x4	周期 IN15	ro	UDINT	0	IN15 周期 [µs]	
0x2013	0x0	周期输入端 平均周期数量	ro	USINT	4	受支持的最大子索引	
	0x1	周期数量 IN12	rw	USINT	4	1...255	IN12 周期数量
	0x2	周期数量 IN13	rw	USINT	4	1...255	IN13 周期数量
	0x3	周期数量 IN14	rw	USINT	4	1...255	IN14 周期数量
	0x4	周期数量 IN15	rw	USINT	4	1...255	IN15 周期数量
0x2014	0x0	周期输入端 – 比率值	ro	USINT	4	受支持的最大子索引	
	0x1	周期比率值 IN12	ro	UINT	0	0...1 000	IN12 传号空号比 [‰]
	0x2	周期比率值 IN13	ro	UINT	0	0...1 000	IN13 传号空号比 [‰]
	0x3	周期比率值 IN14	ro	UINT	0	0...1 000	IN14 传号空号比 [‰]
	0x4	周期比率值 IN15	ro	UINT	0	0...1 000	IN15 传号空号比 [‰]
0x2015	0x0	频率输入端	ro	USINT	4	受支持的最大子索引	
	0x1	频率 IN12	ro	REAL	1	0...30 000	IN12 频率 [Hz]
	0x2	频率 IN13	ro	REAL	1	0...30 000	IN13 频率 [Hz]
	0x3	频率 IN14	ro	REAL	1	0...30 000	IN14 频率 [Hz]
	0x4	频率 IN15	ro	REAL	1	0...30 000	IN15 频率 [Hz]
0x2016	0x0	时间基数	ro	USINT	4	受支持的最大子索引	
	0x1	时间基数 IN12	rw	UINT	50	0...2 000	IN12 时间基数 [ms]
	0x2	时间基数 IN12	rw	UINT	50	0...2 000	IN13 时间基数 [ms]

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	0x3	时间基数 IN12	rw	UINT	50	0...2 000	IN14 时间基数 [ms]
	0x4	时间基数 IN12	rw	UINT	50	0...2 000	IN15 时间基数 [ms]
0x2020	0x0	输入端 - 电源电压短路	ro	USINT	2	受支持的最大子索引	

	0x1	电源电压短路
		IN00...IN07
			ro	USINT	0	0 = 正常
						1 = 短路
							通道（数位编码）
							`0b---- ---X = IN00`
							`0b---- --X- = IN01`
							`0b---- -X-- = IN02`
							`0b---- X--- = IN03`
							`0b---X ---- = IN04`
							`0b--X- ---- = IN05`
							`0b-X-- ---- = IN06`
							`0bX--- ---- = IN07`

	0x2	电源电压短路
		IN08...IN11
			ro	USINT	0	0 = 正常
						1 = 短路
							通道（数位编码）
							`0b---- ---X = IN08`
							`0b---- --X- = IN09`
							`0b---- -X-- = IN10`
							`0b---- X--- = IN11`
							`0b---X ---- = IN12`
							`0b--X- ---- = IN13`
							`0b-X-- ---- = IN14`
							`0bX--- ---- = IN15`

0x2021	0x0	输入端 - 断线	ro	USINT	2	受支持的最大子索引	

	0x1	断线 IN00...IN07	ro	USINT	0	0 = 正常
						1 = 断线
							通道（数位编码）
							`0b---- ---X = IN00`
							`0b---- --X- = IN01`
							`0b---- -X-- = IN02`
							`0b---- X--- = IN03`
							`0b---X ---- = IN04`
							`0b--X- ---- = IN05`
							`0b-X-- ---- = IN06`
							`0bX--- ---- = IN07`

	0x2	断线 IN08...IN11	ro	USINT	0	0 = 正常
						1 = 断线
							通道（数位编码）
							`0b---- ---X = IN08`
							`0b---- --X- = IN09`
							`0b---- -X-- = IN10`
							`0b---- X--- = IN11`
							`0b---X ---- = IN12`
							`0b--X- ---- = IN13`
							`0b-X-- ---- = IN14`
							`0bX--- ---- = IN15`

0x2022	0x0	输出端 – 短路	ro	USINT	1	受支持的最大子索引	

	0x1	短路 OUT00...OUT07	ro	USINT	0	0 = 正常
						1 = 短路
							通道（数位编码）
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`
							`0b---- -X-- = OUT02`
							`0b---- X--- = OUT03`
							`0b---X ---- = OUT04`
							`0b--X- ---- = OUT05`
							`0b-X-- ---- = OUT06`
							`0bX--- ---- = OUT07`

0x2023	0x0	输出端 – 开路	ro	USINT	1	受支持的最大子索引	

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	0x1	开路 OUT00...OUT07	ro	USINT	0	0 = 正常 1 = 开路	通道（数位编码） 0b---- ---X = OUT00 0b---- --X- = OUT01 0b---- -X-- = OUT02 0b---- X--- = OUT03 0b---X ---- = OUT04 0b--X- ---- = OUT05 0b-X-- ---- = OUT06 0bX--- ---- = OUT07
0x2024	0x0	输出端 – 过载	ro	USINT	1	受支持的最大子索引	
	0x1	过载 OUT00...OUT01	ro	USINT	0	0 = 正常 1 = 过载	通道（数位编码） 0b---- ---X = OUT00 0b---- --X- = OUT01
0x2025	0x0	输入端模拟 – 过电流	ro	USINT	1	受支持的最大子索引	
	0x1	过电流 IN00...IN03	ro	USINT	0	0 = 正常 1 = 过电流	通道（数位编码） 0b---- ---X = IN00 0b---- --X- = IN01 0b---- -X-- = IN02 0b---- X--- = IN03
0x2030	0x0	输入端电阻器	ro	USINT	2	受支持的最大子索引	
	0x1	电阻 IN04	ro	UINT	0	0...30 000	IN04 电阻 [Ohms]
	0x2	电阻 IN05	ro	UINT	0	0...30 000	IN05 电阻 [Ohms]
0x2040	0x0	系统电源电压 VBBS	ro	USINT	1	受支持的最大子索引	
	0x1	VBBS	ro	USINT	0	VBBS 电压 [mV]	
0x2041	0x0	输出端电源电压	ro	USINT	2	受支持的最大子索引	
	0x1	VBB1	ro	UINT	0	VBB1 电压 [mV]	
	0x2	VBB2	ro	UINT	0	VBB2 电压 [mV]	
0x2050		装置温度	ro	INT	0	温度 [°C]	
0x20F0		节点 ID	rw	USINT	124	1...125	节点 ID [!] 值 (0x20F0) != 值 (20F1)
0x20F1		节点 ID	rw	USINT	124	1...125	节点 ID [!] 值 (0x20F0) != 值 (20F1)
0x20F2		波特率	rw	USINT	–	波特率 [!] 值 (0x20F2) != 值 (20F3)	
0x20F3		波特率	rw	USINT	–	波特率 [!] 值 (0x20F2) != 值 (20F3)	
0x20F4		自动启动	rw	UINT	0	未使用	
0x6000	0x0	二进制输入端 受支持的最大子索引	ro	USINT	0x02	二进制输入端 00...07 最大受支持子索引 = 2	

索引	S-Idx	参数名称	数据类型		默认	详细信息
	0x1	二进制输入端 IN00 - IN07	ro	USINT	0	二进制输入端 IN00...IN07 0b---- ---X = IN00 0b---- --X- = IN01 0b---- -X-- = IN02 0b---- X--- = IN03 0b---X ---- = IN04 0b--X- ---- = IN05 0b-X-- ---- = IN06 0bX--- ---- = IN07
	0x2	二进制输入端 IN08 - IN15	ro	USINT	0	二进制输入端 IN08...IN15 0b---- ---X = IN08 0b---- --X- = IN09 0b---- -X-- = IN10 0b---- X--- = IN11 0b---X ---- = IN12 0b--X- ---- = IN13 0b-X-- ---- = IN14 0bX--- ---- = IN15
0x6200	0x0	二进制输出端 受支持的最大子索引	ro	USINT	0x02	二进制输出端 最大受支持子索引 = 2
	0x1	二进制输出端 OUT00 - OUT07	wo	USINT	0	二进制输出端 OUT00...OUT07 0b---- ---X = OUT00 0b---- --X- = OUT01 0b---- -X-- = OUT02 0b---- X--- = OUT03 0b---X ---- = OUT04 0b--X- ---- = OUT05 0b-X-- ---- = OUT06 0bX--- ---- = OUT07
	0x2	二进制输出端 OUT08 - OUT15	wo	USINT	0	二进制输出端 OUT08...OUT15 0b---- ---X = OUT08 0b---- --X- = OUT09 0b---- -X-- = OUT10 0b---- X--- = OUT11 0b---X ---- = OUT12 0b--X- ---- = OUT13 0b-X-- ---- = OUT14 0bX--- ---- = OUT15
0x6404	0x0	模拟输入 受支持的最大子索引	ro	USINT	0x04	模拟输入端 最大受支持子索引 = 4
	0x1	模拟输入端 IN00	ro	UINT	--	输入端 IN00 模拟值
	0x2	模拟输入端 IN01	ro	UINT	--	输入端 IN01 模拟值
	0x3	模拟输入端 IN02	ro	UINT	--	输入端 IN02 模拟值
	0x4	模拟输入端 IN03	ro	UINT	--	输入端 IN03 模拟值
0x6414	0x0	PWM 输出端 受支持的最大子索引	ro	USINT	0x12	PWM 输出端 最大受支持子索引 = 12
	0x1	PWM 输出端 OUT00	wo	UINT	--	PWM 输出端 OUT00 的值
	0x2	PWM 输出端 OUT01	wo	UINT	--	PWM 输出端 OUT01 的值
	0x3	PWM 输出端 OUT02	wo	UINT	--	PWM 输出端 OUT02 的值

索引	S-Idx	参数名称	数据类型		默认	详细信息
	0x4	PWM 输出端 OUT03	wo	UINT	--	PWM 输出端 OUT03 的值
	0x5	PWM 输出端 OUT04	wo	UINT	--	PWM 输出端 OUT04 的值
	0x6	PWM 输出端 OUT05	wo	UINT	--	PWM 输出端 OUT05 的值
	0x7	PWM 输出端 OUT06	wo	UINT	--	PWM 输出端 OUT06 的值
	0x8	PWM 输出端 OUT07	wo	UINT	--	PWM 输出端 OUT07 的值
	0x9	PWM 输出端 OUT08	wo	UINT	--	PWM 输出端 OUT08 的值
	0xA	PWM 输出端 OUT09	wo	UINT	--	PWM 输出端 OUT09 的值
	0xB	PWM 输出端 OUT10	wo	UINT	--	PWM 输出端 OUT10 的值
	0xC	PWM 输出端 OUT11	wo	UINT	--	PWM 输出端 OUT11 的值

说明：
数据类型： ro = 只读 / rw = 读写 / wo = 只写

7.3.4 I/O 模块的运行

内容

16433

输入端：PDO 映射（I/O 模块）

15968

下表包含控制配置的以下条目：

- CAN 输入端
- 传送 PDO 映射

数位编码：

```
0b---- ---X = IN00 (IN08)
...
0bX--- ---- = IN07 (IN15)
```

TX-PDO	变量类型	COB ID = 节点 ID + ...	备注
1	USINT	0x180	输入字节 0 (IN00…IN07)
1	USINT	0x180	输入字节 1 (IN08…IN15)
1	USINT	0x180	VBBS 和输入端 (IN00...IN07) 之间短路
1	USINT	0x180	VBBS 和输入端 (IN08…IN15) 之间短路
1	USINT	0x180	输入端 (IN00…IN07) 断线
1	USINT	0x180	输入端 (IN08…IN15) 断线
1	USINT	0x180	输入端 (IN00...IN03) 过电流
2	UINT	0x280	模拟输入端 IN00
2	UINT	0x280	模拟输入端 IN01
2	UINT	0x280	模拟输入端 IN02
2	UINT	0x280	模拟输入端 IN03
3	UINT	0x380	电阻测量输入端 IN04
3	UINT	0x380	电阻测量输入端 IN05
3	UINT	0x380	OUT00 输出电流
3	UINT	0x380	OUT01 输出电流
4	UDINT	0x480	IN12 周期，以 [µs] 为单位
4	UDINT	0x480	IN13 周期，以 [µs] 为单位

TX-PDO	变量类型	COB ID = 节点 ID + ...	备注
5	UDINT	0x181	IN14 周期，以 [µs] 为单位
5	UDINT	0x181	IN15 周期，以 [µs] 为单位
6	UINT	0x281	N12 传号空号比，以 [‰] 为单位
6	UINT	0x281	N13 传号空号比，以 [‰] 为单位
6	UINT	0x281	N14 传号空号比，以 [‰] 为单位
6	UINT	0x281	N15 传号空号比，以 [‰] 为单位
7	USINT	0x381	IN12 频率，以 [Hz] 为单位
7	REAL	0x381	IN13 频率，以 [Hz] 为单位
8	REAL	0x481	IN14 频率，以 [Hz] 为单位
8	REAL	0x481	IN15 频率，以 [Hz] 为单位
9	USINT	0x182	输出端 (OUT00...OUT07) 短路
9	USINT	0x182	输出端 (OUT00...OUT07) 断线
9	USINT	0x182	输出端 (OUT00...OUT01) 过电流
10	UINT	0x282	VBBS 上的电源电压，以 [mV] 为单位
10	UINT	0x282	VBB1 上的电源电压，以 [mV] 为单位
10	UINT	0x282	VBB2 上的电源电压，以 [mV] 为单位
10	UINT	0x282	装置中的温度

输出端： PDO 映射（ I/O 模块）

15969

下表包含控制配置的以下条目：
- CAN 输出端
- 接收 PDO 映射

数位编码：
```
0b---- ---X = OUT00 (OUT08)
...
0bX--- ---- = OUT07 (OUT15)
```

RX-PDO	变量类型	COB ID = 节点 ID + ...	备注
1	USINT	0x200	输出字节 0 (OUT00...OUT07)
1	USINT	0x200	输出字节 1 (OUT08...OUT15)
2	UINT	0x300	PWM 输出端 OUT00
2	UINT	0x300	PWM 输出端 OUT01
2	UINT	0x300	PWM 输出端 OUT02
2	UINT	0x300	PWM 输出端 OUT03
3	UINT	0x400	PWM 输出端 OUT04
3	UINT	0x400	PWM 输出端 OUT05
3	UINT	0x400	PWM 输出端 OUT06
3	UINT	0x400	PWM 输出端 OUT07
4	UINT	0x500	PWM 输出端 OUT08
4	UINT	0x500	PWM 输出端 OUT09
4	UINT	0x500	PWM 输出端 OUT10
4	UINT	0x500	PWM 输出端 OUT11

7.3.5 针对集成 ExB01 I/O 模块的系统标志

内容

16270

系统标志（ExB01 I/O 模块）

15957

没有针对装置集成 I/O 模块的系统标志。
针对进程数据对象 (PDO) 的反馈通过 EDS 文件提供。
→ 章节 **输入端： PDO 映射（I/O 模块）** (→ 页 366)

7.3.6 I/O 模块错误消息

内容

15891

EMCY 对象

15981

基于 DSP-401 或 DSP-301 的以下错误消息代码受支持：

EMCY 代码	错误寄存器	其他代码	说明
0x6100	0x11	0x00	内部软件 Rx 队列溢出 例如，Rx PDO 频率太高 仅通过索引 0x1003 子索引 00 中的条目外部复位
0x6101	0x11	0x00	内部软件 Tx 队列溢出 例如，装置不与总线通信 仅通过索引 0x1003 子索引 00 中的条目外部复位
0x8100	0x11	0x00	监控（保护错误） 未接收针对"保护时间"x"使用期限系数"的保护对象 通过下一次通信复位。
0x8200	0x11	0x00	监控（同步错误） 就"通信周期"而言，未接收同步对象 仅在 OPERATIONAL 时 通过下一次同步 OBJ 或 PREOP 复位

CANopen 不考虑待连续发送的两个相同 EMCY 对象。

SDO 错误消息

15951

出现错误时创建以下消息：

索引	SubIdx	参数名称	数据类型		默认	详细信息	
0x1001		错误寄存器	ro	USINT	0	错误寄存器数位编码至配置文件 301 允许值 = `0b0000 0000` = 无错误 `0b0000 0001` = 一般错误 `0b0001 0000` = 通信错误 `0b1000 0000` = 制造商特定错误	
0x1003	0x0	预定义错误字段 条目数量	rw	UDINT	0	支持包含 4 个条目的错误列表	
	0x1	错误历史记录	ro	UDINT	0	出现了错误； 根据 EMCY 列表编码 子索引 1 表示最后一个错误	
	0x2	错误历史记录	ro	UDINT	0	出现了错误； 根据 EMCY 列表编码	
	0x3	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码	
	0x4	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码	
	0x5	错误历史记录	ro	UDINT	0	出现了错误，根据 EMCY 编码	
0x2020	0x0	输入端 - 电源电压短路	ro	USINT	2	受支持的最大子索引	

	0x1	电源电压短路
		IN00...IN07
			ro	USINT	0	0 = 正常
						1 = 短路
							通道（数位编码）
							`0b---- ---X = IN00`
							`0b---- --X- = IN01`
							`0b---- -X-- = IN02`
							`0b---- X--- = IN03`
							`0b---X ---- = IN04`
							`0b--X- ---- = IN05`
							`0b-X-- ---- = IN06`
							`0bX--- ---- = IN07`

	0x2	电源电压短路
		IN08...IN11
			ro	USINT	0	0 = 正常
						1 = 短路
							通道（数位编码）
							`0b---- ---X = IN08`
							`0b---- --X- = IN09`
							`0b---- -X-- = IN10`
							`0b---- X--- = IN11`
							`0b---X ---- = IN12`
							`0b--X- ---- = IN13`
							`0b-X-- ---- = IN14`
							`0bX--- ---- = IN15`

0x2021	0x0	输入端 - 断线	ro	USINT	2	受支持的最大子索引	

	0x1	断线 IN00...IN07	ro	USINT	0	0 = 正常
						1 = 断线
							通道（数位编码）
							`0b---- ---X = IN00`
							`0b---- --X- = IN01`
							`0b---- -X-- = IN02`
							`0b---- X--- = IN03`
							`0b---X ---- = IN04`
							`0b--X- ---- = IN05`
							`0b-X-- ---- = IN06`
							`0bX--- ---- = IN07`

索引	SubIdx	参数名称	数据类型		默认	详细信息	

	0x2	断线 IN08...IN11	ro	USINT	0	0 = 正常
						1 = 断线
							通道（数位编码）
							`0b---- ---X = IN08`
							`0b---- --X- = IN09`
							`0b---- -X-- = IN10`
							`0b---- X--- = IN11`
							`0b---X ---- = IN12`
							`0b--X- ---- = IN13`
							`0b-X-- ---- = IN14`
							`0bX--- ---- = IN15`

0x2022	0x0	输出端 – 短路	ro	USINT	1	受支持的最大子索引	

	0x1	短路 OUT00...OUT07	ro	USINT	0	0 = 正常
						1 = 短路
							通道（数位编码）
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`
							`0b---- -X-- = OUT02`
							`0b---- X--- = OUT03`
							`0b---X ---- = OUT04`
							`0b--X- ---- = OUT05`
							`0b-X-- ---- = OUT06`
							`0bX--- ---- = OUT07`

0x2023	0x0	输出端 – 开路	ro	USINT	1	受支持的最大子索引	

	0x1	开路 OUT00...OUT07	ro	USINT	0	0 = 正常
						1 = 开路
							通道（数位编码）
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`
							`0b---- -X-- = OUT02`
							`0b---- X--- = OUT03`
							`0b---X ---- = OUT04`
							`0b--X- ---- = OUT05`
							`0b-X-- ---- = OUT06`
							`0bX--- ---- = OUT07`

0x2024	0x0	输出端 – 过载	ro	USINT	1	受支持的最大子索引	

	0x1	过载 OUT00...OUT01	ro	USINT	0	0 = 正常
						1 = 过载
							通道（数位编码）
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`

0x2025	0x0	输入端模拟 – 过电流	ro	USINT	1	受支持的最大子索引	

	0x1	过电流 IN00...IN03	ro	USINT	0	0 = 正常
						1 = 过电流
							通道（数位编码）
							`0b---- ---X = IN00`
							`0b---- --X- = IN01`
							`0b---- -X-- = IN02`
							`0b---- X--- = IN03`

说明

数据类型： ro = 只读 / rw = 读写 / wo = 只写

7.4 错误表

内容

19606

7.4.1 错误代码

内容

12334

通过某些功能块输出的错误代码的概述。

32 位错误代码包含四个 8 位值 (DWORD)。

第 4 个字节	第 3 个字节	第 2 个字节	第 1 个字节
错误类别	应用程序特定错误代码	错误来源	错误原因

错误原因（第 1 个字节）

19273

值 十进制	十六进制	说明
0	00	无错误原因 或： 应用程序特定错误
1	01	断线
2	02	短路
4	04	过载
5	05	欠电压
6	06	过电压
7	07	电流控制
24	18	温度
26	20	内存测试
27	21	内存测试
51	33	整数溢出或： 除以零
56	38	FPU 下溢
57	39	FPU 过溢
58	3A	FPU 除以零
59	3B	FPU 不明错误
128	80	CRC
129	81	损坏数据
130	82	内存保护
131	83	无
144	90	电子狗
145	91	陷阱
147	93	断言失败
194	C2	CAN 总线关闭
224	E0	板连接警告 (ExtendedController)
225	E1	板连接错误 (ExtendedController)
240	F0	序列号

值 十进制	十六进制	说明
241	F1	运行时系统过期
248	F8	错误的参数

错误来源（第 2 个字节）

18660

值		说明
十进制	十六进制	
0	00	无错误来源 或： 应用程序特定错误
1	01	CPU
2	02	外围处理机
3	03	协处理器
8	08	浮点单元
16...31	10...1F	输入端 0...15（标准侧）
32...63	20...3F	输入端 0...31（扩展侧）
64...79	40...4F	输出端 0...15（标准侧）
80..111	50...6F	输出端 0...31（扩展侧）
128...131	80...83	CAN 1...4
144	90	继电器电压 VBBo（标准侧）
145	91	继电器电压 VBBr（标准侧）
146	92	VBBo（标准侧）
147	93	VBBr（标准侧）
148	94	VBBs（标准侧）
149	95	夹具 15
150	96	继电器电压 VBB1（扩展侧）
151	97	继电器电压 VBB2（扩展侧）
152	98	继电器电压 VBB3（扩展侧）
153	99	继电器电压 VBB4（扩展侧）
154	9A	VBBrel（扩展侧）
155	9B	VBB1（扩展侧）
156	9C	VBB2（扩展侧）
157	9D	VBB3（扩展侧）
158	9E	VBB4（扩展侧）
160	A0	模拟多路复用器
161	A1	模拟参考

值 十进制	十六进制	说明
176	B0	内部闪存
177	B1	外部闪存
178	B2	内部 RAM
179	B3	外部 RAM
192	C0	代码 Startupper
193	C1	代码 Bootloader
194	C2	代码运行时系统
196	C4	启动项目
197	C5	代码应用程序
198	C6	暂用 RAM
224	E0	系统数据
225	E1	系统设定
226	E2	系统信息
227	E3	校准数据
228	E4	FRAM / MRAM（用户区域）

应用程序特定错误代码（第 3 个字节

12338

如果是应用程序特定错误，则要求如下：
ERRORCODE byte 1 = error cause = 0x00
ERRORCODE byte 2 = error source = 0x00

► 利用以下功能块将应用程序特定错误的信号发送至控制器： ERROR_REPORT (→ 页 262).

值 十进制	十六进制	说明
0	00	无应用程序特定错误
> 0	> 00	应用程序特定错误

错误类别（第 4 个字节

19271

该信息仅适用于输入端 TEST = FALSE。

值 十进制	十六进制	说明
0	00	无错误
1	01	一般错误，与安全无关 > 错误标志，错误代码 > 可重置错误
3	03	ERROR STOP > 错误代码 > 应用程序中止 > 安全状态 ▶ 需断电通电复位
4	04	严重错误 > 错误代码 > 应用程序中止 > 安全状态 > 装置重启（软复位）

14025

严重错误和 ERROR STOP 仅在以下条件下显示于应用程序中：TEST 输入端已在发生错误时启用。

如果 TEST 输入端 = FALSE：

- ERROR STOP 导致控制器停止
- 严重错误导致控制器重启

错误代码： 示例

19274

Byte 2	▼ 第 1 个字节 ▼		
错误来源 [hex]	**错误原因 [hex]**	**说明**	**功能块**
10...1F	01	断线 Ixx	INPUT_ANALOG
20...3F (Ex)	01	断线 Ixx_E	INPUT_ANALOG_E
40...4F	01	断线 Qxx	
50...6F (Ex)	01	断线 Qxx_E	
10...1F	02	短路 Ixx	INPUT_ANALOG
20...3F (Ex)	02	短路 Ixx_E	INPUT_ANALOG_E
40...4F	02	短路 Qxx	
50...6F (Ex)	02	短路 Qxx_E	
10...1F	04	过电流 Ixx	INPUT_ANALOG
20...3F (Ex)	04	过电流 Ixx_E	INPUT_ANALOG_E
40...4F	04	过载 Qxx	
50...6F (Ex)	04	过载 xx_E	
90	05	欠电压继电器电压 VBBs	
91	05	欠电压继电器电压 VBBr	
92	05	欠电压 VBBo	
93	05	欠电压 VBBr	
94	05	欠电压 VBBs	
95	05	欠电压夹具 15	
96...99 (Ex)	05	欠电压继电器电压 VBBx	
9A (Ex)	05	欠电压 VBBrel	
9B...9E (Ex)	05	欠电压 VBBx	
90	06	过电压继电器电压 VBBs	
91	06	过电压继电器电压 VBBr	
92	06	过电压 VBBo	
93	06	过电压 VBBr	
94	06	过电压 VBBs > 32 V	
94	06	过电压 VBBs > 34 V	
95	06	过电压夹具 15	

Byte 2	▼ 第 1 个字节 ▼		
错误来源 [hex]	错误原因 [hex]	说明	功能块
96...99 (Ex)	06	过电压继电器电压 VBBx	
9A (Ex)	06	过电压 VBBrel	
9B...9E (Ex)	06	过电压 VBBx	
40...4F	07	电流控制 Qxx	
50...6F (Ex)	07	电流控制 Qxx_E	
10...1F	08 (safe)	电流输入端上的安全诊断	
10...1F	09 (safe)	电压输入端上的安全诊断	
40...4F	0A (safe)	已启用输出端上的安全诊断（停留在 1，交叉故障）	
10...17	0B (safe)	安全开关上的安全诊断	SAFETY_SWITCH
A0	0C (safe)	模拟值监控/多路复用器	SAFETY_SWITCH
40...4F	0D (safe)	已禁用输出端上的安全诊断（停留在 1）	
00	18	温度错误	
90...91	19 (safe)	触点故障继电器 VBBo / VBBr	
B3	20	RAM 内存测试失败	
E4	20	FRAM/MRAM 内存测试失败	
B3	21 (safe)	RAM 地址测试失败	
01	30 (safe)	错误/缺少中断	
01	31 (safe)	CPU 错误时基	
08	39	浮点过溢	
08	3A	浮点除以 0	
08	3B	不明浮点错误	
C2	80	运行时系统代码（IFM 代码）中的校验和错误	
C3 (safe)	80	PCP 数据 RAM 中的校验和错误	
C4	80	启动项目中的校验和错误	
C5	80	应用程序代码中的校验和错误	
C6	80	SP-RAM 中的校验和错误	
C7 (safe)	80	PCP 代码 RAM 中的校验和错误	
E0	80	系统数据中的校验和错误	
E1	80	系统变量中的校验和错误	
E2	80	系统参数中的校验和错误	

Byte 2	▼ 第 1 个字节 ▼		
错误来源 [hex]	错误原因 [hex]	说明	功能块
E3	80	校准数据中的校验和错误	
B3	81	RAM 中的损坏数据	
E3	83	损坏校准数据	
04	92 (safe)	安全核心已停止	
05	92 (safe)	错误安全代码	
80...83	C1	CANx 警告	
80...83	C2	CANx 总线关闭	
80...83	C3 (safe)	CAN 安全接收错误	
80...83	C4 (safe)	CAN 安全传送错误	
80, 82	C5 (safe)	CAN 安全配置损坏	
00 (Ex)	E1	板连接错误	
00	F0	序列号错误	
00	F1	运行时系统过期	
00	F8	参数错误	所有包含 ERROR 输出的 FB
10...1F	F8	参数错误 Ixx	INPUT_ANALOG SET_INPUT_MODE
20...3F (Ex)	F8	参数错误 Ixx_E	INPUT_ANALOG_E SET_INPUT_MODE_E
40...4F	F8	参数错误 Qxx	SET_OUTPUT_MODE
50...6F (Ex)	F8	参数错误 Qxx_E	SET_OUTPUT_MODE_E
05 (safe)	ErrorCode	安全代码错误	
06 (safe)	TrapID	安全代码陷阱错误	

说明：

(Ex) = 仅适用于 ExtendedController

(Ex) = 仅适用于 SafetyController

产生的错误类别（= 第 4 个字节）由相应的状况和参数设定决定。 此时，第 3 个字节（应用程序特定错误代码）始终 = 0。

7.4.2 错误标志

19608

→ 章节 **系统标志** (→ 页 273)

7.4.3 错误： CAN / CANopen

19610
19604

→ 系统手册“ecomat*mobile* 技巧”
→ 章节 **CAN / CANopen ： 错误和错误处理**

EMCY 代码： CANx

13094

CANx 指示还适用于各个 CAN 接口。

EMCY 代码 对象 0x1003		对象 0x1001	制造商特定信息					
第 0 个字节 [hex]	第 1 个字节 [hex]	第 2 个字节 [hex]	第 3 个字节	第 4 个字节	第 5 个字节	第 6 个字节	第 7 个字节	说明
00	80	11	--	--	--	--	--	CANx 监控 SYNC 错误（仅从站）
00	81	11	--	--	--	--	--	CANx 警告阈值 (> 96)
10	81	11	--	--	--	--	--	CANx 接收缓冲区超限
11	81	11	--	--	--	--	--	CANx 传送缓冲区超限
30	81	11	--	--	--	--	--	CANx 保护/检测信号错误（仅从站）

EMCY 代码： I/O，系统（标准侧）

2668

在以下情况下，以下 EMCY 消息自动发送：
- 作为 CANopen 主站： 如果 **CANx_MASTER_EMCY_HANDLER** (→ 页 103)循环调用
- 作为 CANopen 从站： 如果 **CANx_SLAVE_EMCY_HANDLER (→ 页 115)** 循环调用

EMCY 代码 对象 0x1003		对象 0x1001	制造商特定信息					
第 0 个字节 [hex]	第 1 个字节 [hex]	第 2 个字节 [hex]	第 3 个字节	第 4 个字节	第 5 个字节	第 6 个字节	第 7 个字节	说明
00	21	03	I07...I00	I15...I08				输入端中断
08	21	03	I07...I00	I15...I08				输入端短路
10	21	03	I07...I00	I15...I08				过电流 0...20 mA
00	23	03	Q07...Q00	Q15...Q08				输出端中断
08	23	03	Q07...Q00	Q15...Q08				输出端短路
00	31	05						端子电压 VBBs

EMCY 代码 对象 0x1003		对象 0x1001	制造商特定信息					
第 0 个字节 [hex]	第 1 个字节 [hex]	第 2 个字节 [hex]	第 3 个字节	第 4 个字节	第 5 个字节	第 6 个字节	第 7 个字节	说明
00	33	05						端子电压 VBBo
08	33	05						端子电压 VBBr
00	42	09						过温

8 Glossary of Terms

B

Bootloader

交付时 **ecomat*mobile*** 控制器仅包含 Bootloader。

Bootloader 是有助于将运行时系统和应用程序再次加载至装置的启动程序。

Bootloader 包含基本例程…

- 针对硬件模块之间的通信，
- 针对操作系统的重新加载。

Bootloader 是要保存在装置上的第一个软件模块。

C

CAN

CAN = **控**制器**局**域**网**络

CAN 是针对大数据量的优先级控制现场总线系统。 有几个基于 CAN 的较高级协议，如 'CANopen' 或 'J1939'。

CAN 堆栈

CAN 堆栈 = 负责处理 CAN 消息的软件组件。

CiA

CiA = CAN in Automation e.V.

德国/埃朗根的用户和制造商组织。 CAN 和 CAN 网络协议的定义和控制体。

主页 → www.can-cia.org

CiA DS 304

DS = **标**准**草**案

针对安全通信的 CANopen 装置行规

CiA DS 401

DS = **标**准**草**案

针对二进制和模拟 I/O 模块的 CANopen 装置行规

CiA DS 402

DS = **标**准**草**案

针对驱动器的 CANopen 装置行规

CiA DS 403

DS = **标准草**案

针对 HMI 的 CANopen 装置行规

CiA DS 404

DS = **标准草**案

针对测量和控制技术的 CANopen 装置行规

CiA DS 405

DS = **标准草**案

可编程控制器 (IEC 61131-3) 接口 CANopen 规范

CiA DS 406

DS = **标准草**案

针对编码器的 CANopen 装置行规

CiA DS 407

DS = **标准草**案

针对本地公共交通的 CANopen 应用行规

COB ID

COB = **通信对**象

ID = **标**识符

CANopen 通信对象 ID

相当于 CAN 消息的标识符，通过 CAN 总线铜通信项目一起发送。

CODESYS

CODESYS® 是德国 3S（即 Smart Software Solutions GmbH）的注册商标。

“自动化联盟 CODESYS”将自动化行业利用广泛使用的 IEC 61131-3 开发工具 CODESYS® 对其所有硬件装置进行编程的公司联合起来。

主页 → www.codesys.com

CSV 文件

CSV = **逗号分隔值**（又称： **字符分隔值**）
CSV 文件是存储或交换简单结构数据的文本文件。
文件扩展名为 .csv。

例如： 包含数值的来源表：

值 1.0	值 1.1	值 1.2	值 1.3
值 2.0	值 2.1	值 2.2	值 2.3
值 3.0	值 3.1	值 3.2	值 3.3

因此 CSV 文件如下：

```
值 1.0；值 1.1；值 1.2；值 1.3
值 2.0；值 2.1；值 2.2；值 2.3
值 3.0；值 3.1；值 3.2；值 3.3
```

D

DC

直流电

DLC

数据长度码 = CANopen 消息数据字节的数量。
针对 →SDO： DLC = 8

DRAM

DRAM = **动**态**随**机**存**取**内**存。
随机存取电子内存 （RAM)模块技术 内存元件是可充电或放电的电容器。 可通过开关晶体管访问，可读取或由新内容覆盖。 内存内容易失： 如果工作电压缺失或重启太迟，则存储的信息丢失。

DTC

DTC = **诊**断**故**障**码** = 错误代码
在协议 J1939 中，故障和错误有效管理，且可通过指定的编号 (DTC) 报告。

E

ECU

(1) **电**子**控**制**器** = 控制装置或微控制器
(2) **发**动机**控**制**器** = 发送机控制装置

EDS 文件

EDS = **电子数据表**，例如针对以下方面：
• CANopen 主站对象目录文件，
• CANopen 装置说明。
装置和程序可通过 EDS 交换其规范并简单加以考虑。

EMC

EMC = **电磁兼容性**。
根据关于电磁兼容性的 EC 指令 (2004/108/EEC)（简称为“EMC 指令”），电气电子装置、设备、系统或组件需满足一定要求，方可在现有电磁环境下正常运行。 装置不得干扰其环境，且不可受到外部电磁干扰的不利影响。

EMCY

紧急情况的缩写
CANopen 协议中的消息，错误通过消息发送。

Ethernet

以太网是广泛使用且独立于制造商的技术，有助于在网络中以 10...10 000 Mbps 的速度传送数据。 以太网属于非专属传送媒介中所谓的“最优数据传送”系列。 该概念形成于 1972 年并在 1985 年确定为 IEEE 802.3。

EUC

EUC = **受控设备**。
EUC 即用于制造、处理、运输、医疗或其他活动的设备、机械或装置（→ IEC 61508-4，第 3.2.3 节）。 因此，EUC 是所有需要安全相关系统以防发生危害的设备、机械或装置的集合。
如果任何合理预见的行为或不作为导致→危害且无法容忍的风险源于 EUC，则需要使用安全功能以达到或维护 EUC 的安全状态。 这些安全功能通过一个多个安全相关系统执行。

F

FiFo

FIFO（**先入先出**）= 堆栈内存的工作原理： 先写入堆栈内存的数据包也将先读取。 每个标识符均有这样的一个缓冲区（队列）。

FRAM

FRAM，又称 FeRAM，指**铁**电**随**机**存**取**内**存。 存储和清除操作可通过铁电层的极化变化进行。
FRAM 相比传统只读内存的优势：
• 非易失性，
• 可兼容常见的 EEPROM，但是：
• 存取时间大约为 100 ns，
• 存取周期几乎无限制。

H

HMI

HMI = **人机界**面

I

ID

ID = **标**识符

区分连接至系统的装置/参与者或在参与者之间传送的消息报的名称。

IEC 61131

标准： 编程逻辑控制器基础知识
• 第 1 部分： 一般信息
• 第 2 部分： 生产设备要求和测试
• 第 3 部分： 编程语言
• 第 5 部分： 通信
• 第 7 部分： 模拟控制编程

IEC 用户周期

IEC 用户周期 = CODESYS 应用程序中的 PLC 周期。

IP 地址

IP = **互联**网**协**议。
IP 地址是有助于清楚识别互联网参与者的编号。 为清晰起见，按照 4 个十进制值写入编号，如 127.215.205.156。

ISO 11898

标准： 道路车辆 - 控制器局域网络

- 第 1 部分： 数据链路层和物理信号发送
- 第 2 部分： 高速媒介访问单元
- 第 3 部分： 低速容错的媒介依赖型接口
- 第 4 部分： 时间触发通信
- 第 5 部分： 低功率模式高速媒介访问单元

ISO 11992

标准： 拖吊和被拖吊车辆之间电气连接数字信息的交换

- 第 1 部分： 物体和数据链路层
- 第 2 部分： 针对制动器和驱动装置的应用层
- 第 3 部分： 针对除制动器和驱动装置之外的设备的应用层
- 第 4 部分： 诊断

ISO 16845

标准： 道路车辆 - 控制器局域网络 (CAN) - 符合性测试计划

J

J1939

→ SAE J1939

L

LED

LED = **发光二**极管。

发光二极管，体积小、功耗可忽略不计的高彩色亮度电子元件。

LSB

最低**有**效**位**/字节

M

MAC-ID

MAC = **制**造商**地**址**代**码

= 制造商序列号。

→ID = **标**识符

每个网卡均有 MAC 地址，即清楚定义的全球唯一数字码，或称序列号。 该 MAC 地址由 6 个十进制数字组成，如 "00-0C-6E-D0-02-3F"。

MMI

→ **HMI** (→ 页 389)

MRAM

MRAM = **磁**阻**随**机**存**取**内**存
信息通过磁存储器存储。 特定材料的性质用于在接触磁场时更改其电阻。
MRAM 相比传统 RAM 内存的优势：

- 非易失性（如 FRAM），但是
- 存取时间仅为 35 ns 左右，
- 存取周期无限制。

MSB

最高**有**效**位**/字节

N

NMT

NMT = **网**络**管**理 =（此处； CANopen 协议中）。
NMT 主站控制 NMT 从站的工作状态。

O

Obj /对象

即可在 CANopen 网络中交换的数据/消息。

OBV

包含装置的所有 CANopen 通信参数以及装置特定参数和数据。

OPC

OPC = **流**程**控**制 **O**LE
独立于制造商的自动化技术通信的标准化软件接口
OPC 客户端（如参数设定或编程装置）在连接后自动登录至 OPC 服务器（如自动化装置）并与其进行通信。

P

PC 卡

→PCMCIA 卡

PCMCIA 卡

PCMCIA = 个人电脑存储卡国际协会标准，针对移动计算机扩展卡的标准。
自 1995 年推出卡总线标准以来，PCMCIA 卡还被称为 PC 卡。

PDM

PDM = **进**程和**对**话**模**块。
操作员机器/设备通信装置。

PDO

PDO = **进**程**数**据**对**象。
时间关键型进程数据通过“进程数据对象 (PDO)”传输。 PDO 可在单个节点之间自由更换（PDO 连接）。 此外，还定义了数据交换要根据事件控制（异步）还是同步。 正确选择传送类型有助于显著减轻 →CAN 总线的压力，具体视待传输数据的类型而定。
根据协议，这些服务为未经确认的数据传送： 未检查接收器是否接收消息。 网络变量交换对应“1 到 n 个连接”（1 个变送器到 n 个接收器）。

PDU

PDU = **协**议**数**据**单**元。
PDU 是→CAN 协议→SAE J1939 的一项。 PDU 表示目标或源地址的一部分。

PES

可编程**电**子**系**统...
• 用于控制、保护或监控，
• 其工作依赖于一个或多个可编程电子装置，
• 包括输入和输出装置等所有系统元件。

PGN

PGN = **参数组编**号
PGN = PDU 格式 (PF) + PDU 源 (PS)
参数组编号是→CAN 协议→SAE J1939 的一项。 PGN 收集地址部分 PF 和 PS。

PID 控制器

PID 控制器（比例-积分-微分）包含以下单元：
• P = 比例单元
• I = 积分单元
• D = 微分单元（但不适用于控制器 CR04nn、CR253n）。

PLC 配置

CODESYS 用户界面的部分

► 程序员告知编程系统哪些硬件有待编程。

> CODESYS 加载相应的库。

> 可读取和写入外围状态（输入端/输出端）。

PWM

PWM = 脉冲宽度调制

PWM 输出信号即 GND 和电源电压之间的脉冲信号。

在确定的周期（PWM 频率）内，传号空号比是不同的。 连接的负载决定相应的 RMS 电流，具体视传号空号比而定。

R

RAW-CAN

RAW-CAN 指在 CAN 总线（ISO/OSI 第 2 层）上没有额外通信协议的情况下行之有效的纯粹 CAN 协议。 CAN 根据 ISO 11898-1 在国际范围内定义，此外还根据 ISO 16845，确保 CAN 芯片的可交换性。

ro

RO = 只读

单向数据传送： 数据仅可读取，不可更改。

RTC

RTC = **实**时**时钟**

提供（电池支持）当前的日期和时间。 常用于存储错误消息协议。

rw

RW = 读取/写入

双向数据传送： 数据既可读取，也可更改。

S

SAE J1939

网络协议 SAE J1939 讲述针对诊断数据（如发动机速度、温度）和控制信息传送的商业车辆 →CAN 总线通信。

标准： 串行控制和通信车辆网络建议实践

- 第 2 部分： 农林越野机械控制和通信网络
- 第 3 部分： 车载诊断实施指南
- 第 5 部分： 船尾驱动器和船内火花点火发动机船载诊断实施指南
- 第 11 部分： 物理层 – 250 kBits/s，屏蔽双绞线
- 第 13 部分： 场外诊断连接器
- 第 15 部分： 简化物理层，250 kBits/s，非屏蔽双绞线 (UTP)
- 第 21 部分： 数据链路层
- 第 31 部分： 网络层
- 第 71 部分： 车辆应用层
- 第 73 部分： 应用层 - 诊断
- 第 81 部分： 网络管理协议

SD 卡

SD 内存卡（**安**全**数**字内存卡的简称）是根据→闪速存储原理运行的数字存储媒介。

SDO

SDO = **服**务**数**据**对**象。

SDO 用于访问 CANopen 对象目录中的对象。 “客户端”向“服务器”请求所需数据。 SDO 始终包含 8 个字节。

例如：

- 在系统启动时通过 →SDO 自动配置所有从站，
- 读取 →对象目录中的错误消息。

每个 SDO 均接受响应监控，如果从站未在监控时间内响应，则重复 SDO。

T

TCP

传输控制协议是 TCP/IP 协议系列的一部分。 每个 TCP/IP 数据连接均有一个变送器和一个接收器。 该原理为连接导向数据传输。 在 TCP/IP 协议系列中，作为连接导向协议，TCP 承担数据保护、数据流控制的任务，并在数据丢失时采取措施。 （比较： →UDP）

U

UDP

UDP（**用**户**数**据报**协**议）是最低限度无连接网络协议，属于网络协议系列中的传输层。 UDP 的任务是确保通过互联网传输的数据传送给适当的应用。
目前可实施基于 →CAN 和 UDP 的网络变量。 变量的值根据广播消息自动更换。 在 UDP 中，它们作为广播消息实施，在 CAN 中则作为 →PDO 实施。
根据协议，这些服务为未经确认的数据传送： 未检查接收器是否接收消息。 网络变量交换对应“1 到 n 个连接”（1 个变送器到 n 个接收器）。

三划

已停止

CANopen 参与者的工作状态。 在该模式中仅可传输 →NMT 命令。

四划

从站

总线上的被动参与者，仅依赖→主站的请求。 在总线中，从站有明确定义且唯一的→地址。

比率式

还可进行比率测量。 如果传感器输出信号与其电源电压成比例，则可通过比率测量（= 与电源成比例的测量）减少电源波动的影响，在理想情况下，甚至可避免该影响。
→ 模拟输入端

五划

主站

处理总线的整体安排。 主站决定总线访问时间并定期→轮询从站。

对象目录

包含装置的所有 CANopen 通信参数以及装置特定参数和数据。

电子狗

一般而言，“电子狗”一词表示监控其他组件功能的系统组件。 如果检测到可能的故障，则向其发送信号或启用相应的程序分支。 信号或程序分支作为其他协作系统组件的触发器，以解决问题。

目标

目标包含 CODESYS 目标装置的硬件说明，如： 输入端和输出端、内存、文件位置。
对应电子技术资料。

节点

即网络中的参与者。

节点保护

节点 = 此处： 网络参与者
每个从站的可配置循环→监控可相应地加以配置。 →主站核查从站是否及时响应。 从站核查主站是否定期发送请求。 这样即可快速识别和报告出现故障的网络参与者。

闪存

快闪 ROM（或快闪 EPROM 或闪存）将半导体内存和硬盘的优势相结合。 但是跟硬盘类似，数据在高达 64、128、256、1024... 字节的数据块中同时按区块写入和删除。

闪存的优势

- 即便无电源电压亦可保留存储的数据。
- 因为没有移动部件，闪存无噪音且对冲击和磁场不敏感。

闪存的劣势

- 存储单元可容许有限数量的写入和删除进程：
  - 多级单元： 一般为 10 000 个周期
  - 单级单元： 一般为 100 000 个周期
- 鉴于写入进程同时写入 16 和 128 K 字节之间的内存块，还可使用无需更改的内存单元。

六划

地址

这是总线参与者的“名称”。 所有参与者需要唯一的地址，以便无忧更换信号。

夹具 15

在车辆中，夹具 15 是通过点火锁开关的正极电缆。

自检

主动检验组件或装置的检验程序。 程序由用户启动，且将花费一定的时间。 结果为显示检验内容以及结果为正面还是负面的检验协议（日志文件）。

七划

应用程序软件

针对应用程序的软件，由机器制造商实施，一般包含控制相应输入端、输出端计算和决策的逻辑序列、限制和表达式。

抖动频率

抖动频率是 →PWM 信号的组成部分，用于控制液压阀。 对液压阀的电磁驱动器来说，如果 PWM 频率的特定频率叠加控制信号（PWM 脉冲），则控制阀门要容易得多。 该抖动频率必须是 PWM 频率的整数部分。

系统变量

可通过 IEC 地址或符号名称从 PLC 访问的变量。

诊断

诊断期间，检查装置的“健康状态”。 从而将发现装置是否存在→以及存在何种故障。

还可监控输入端和输入端是否正常运行，具体视装置而定。
- 断线，
- 短路，
- 值在范围之外。

就诊断而言，可使用装置“正常”运行期间创建的配置和日志数据。
初始化和启动期间可监控系统组件是否正确启动。 日志文件记录错误。
就进一步诊断而言，还可进行自检。

运行

CANopen 参与者的工作状态 在该模式中可传输→SDO、→NMT 命令和 →PDO。

运行时系统

装置中的基本程序，建立装置硬件和应用程序之间的连接。

进程图像

进程图像即在一个→周期内 PLC 据其运行的输入端和输出端状态。

- 在周期开始时，PLC 将所有输入端的状况读取至进程图像。
  周期期间，PLC 无法检测输入端的更改。
- 周期期间，仅可对输出端进行虚拟更改（在进程图像中）。
- 在周期结束时，PLC 将虚拟输出状态写入实际输出端。

连接指令

连接指令是指对文件中其他部分或外部文件的交叉引用。

八划

使用，指定

根据用途说明所载的信息使用产品。

周期时间

即周期的时间。 PLC 程序展开一次完整的运行。

这可能需要更长或更短的时间，具体视程序中的事件控制分支而定。

图标

图标即通过简化的图示传达信息的象征性符号。 （→ 章节 **符号和格式是什么意思？** (→ 页 8))

波特

波特 (Baud)，缩写： Bd = 数据传送速度单位。 切勿将波特与“位/秒 (bps、bits/s)”混淆。 波特表示在一定传送长度内每秒的状态（步骤、周期）更改量。 但未定义每步传送的位数。 Baud 一词可追溯到法国发明家 J. M. Baudot,其编码用于电传机器。

1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

九划

总线

同一电缆上多个参与者的串行数据传送.

指定用途

根据用途说明所载的信息使用产品。

架构

系统硬件和/或软件的特定配置。

误用

未按照设计方指定的方式使用产品。
产品制造商应在用户信息中针对可预见的误用提出警告。

说明

以下术语之一的上义词：
安装说明、技术资料、用户信息、操作说明、装置手册、安装信息、联机帮助、系统手册、编程手册等。

十划

预运行

预运行 = PRE-OPERATIONAL 模式
CANopen 参与者的工作状态。 施加电源电压后，每个参与者自动进入该状态。 在 CANopen 网络中，仅→ SDO 和 →NMT 命令可在该模式下传输，但无进程数据。

十一划

检测信号

参与者定期发送短信号。 这样其他参与者则可核实参与者是否出现故障。

符号

图标即通过简化的图示传达信息的象征性符号。 （→ 章节 **符号和格式是什么意思？** (→ 页 8))

十二划

剩余

电源故障时剩余数据不会丢失。
当电源电压跌破关键值时→运行时系统等即会自动将剩余数据复制到→闪存。 如果电源电压再次可用，则运行时系统将剩余数据加载回 RAM 内存。
但是控制器 RAM 内存中的数据易失，通常会在电源故障时丢失。

嵌入式软件

装置中的系统软件、基本程序，实际上→是运行时系统中。
固件建立装置硬件和应用程序之间的连接。 固件作为系统的一部分由控制器制造商提供，且用户不可更改。

十三划

数据类型

可存储不同大小的值，具体式数据类型而定。

数据类型	最小值	最大值	内存大小
BOOL	FALSE	TRUE	8 位 = 1 个字节
BYTE	0	255	8 位 = 1 个字节
WORD	0	65 535	16 位 = 2 个字节
DWORD	0	4 294 967 295	32 位 = 4 个字节
SINT	-128	127	8 位 = 1 个字节
USINT	0	255	8 位 = 1 个字节
INT	-32 768	32 767	16 位 = 2 个字节
UINT	0	65 535	16 位 = 2 个字节
DINT	-2 147 483 648	2 147 483 647	32 位 = 4 个字节
UDINT	0	4 294 967 295	32 位 = 4 个字节
REAL	$-3.402823466 \cdot 10^{38}$	$3.402823466 \cdot 10^{38}$	32 位 = 4 个字节
ULINT	0	18 446 744 073 709 551 615	64 位 = 8 个字节
STRING			字符数量 + 1

十四划

模板

模板可填入内容。

此处： 预配置软件元件结构，作为应用程序的基础。

9 索引

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

一划

二划

三划

四划

五划

六划

七划

八划

九划

十划

十一划

十二划

十三划

十四划

10 备注

11 ifm weltweit • ifm worldwide • ifm à l’échelle internationale

Version: 2016-11-29 8310

ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen

www.ifm.com • Email: info@ifm.com

Service hotline: 0800 / 16 16 16 (only Germany, Mo-Fr 07.00...18.00 h)

ifm Niederlassungen • Sales offices • Agences

D	Niederlassung Nord • 31135 Hildesheim • Tel. 0 51 21 / 76 67-0
	Niederlassung West • 45128 Essen • Tel. 02 01 / 3 64 75 -0
	Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 0 23 51 / 43 01-0
	Niederlassung Süd-West • 64646 Heppenheim • Tel. 0 62 52 / 79 05-0
	Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 0 70 21 / 80 86-0
	Niederlassung Bayern • 82178 Puchheim • Tel. 0 89 / 8 00 91-0
	Niederlassung Ost • 07639 Tautenhain • Tel. 0 36 601 / 771-0
A, SL	ifm electronic gmbh • 1120 Wien • Tel. +43 16 17 45 00
AUS	ifm efector pty ltd. • Mulgrave Vic 3170 • Tel. +61 3 00 365 088
B, L	ifm electronic N.V. • 1731 Zellik • Tel. +32 2 / 4 81 02 20
BG	ifm electronic eood • 1202 Sofia • Tel. +359 2 807 59 69
BR	ifm electronic Ltda. • 03337-000, Sao Paulo SP • Tel. +55 11 / 2672-1730
CH	ifm electronic ag • 4 624 Härkingen • Tel. +41 62 / 388 80 30
CL	ifm electronic SpA • Oficina 5032 Comuna de Conchalí • Tel. +55 11 / 2672-1730
CN	ifm electronic (Shanghai) Co. Ltd. • 201203 Shanghai • Tel. +86 21 / 3813 4800
CND	ifm efector Canada inc. • Oakville, Ontario L6K 3V3 • Tel. +1 800-441-8246
CZ	ifm electronic spol. s.r.o. • 25243 Průhonice • Tel. +420 267 990 211
DK	ifm electronic a/s • 2605 BROENDBY • Tel. +45 70 20 11 08
E	ifm electronic s.a. • 08820 El Prat de Llobregat • Tel. +34 93 479 30 80
F	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • Tél. +33 0820 22 30 01
FIN	ifm electronic oy • 00440 Helsinki • Tel . +358 75 329 5000
GB, IRL	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • Tel. +44 208 / 213-0000
GR	ifm electronic Monoprosopi E.P.E. • 15125 Amaroussio • Tel. +30 210 / 6180090
H	ifm electronic kft. • 9028 Györ • Tel. +36 96 / 518-397
I	ifm electronic s.a. • 20041 Agrate-Brianza (MI) • Tel. +39 039 / 68.99.982
IL	Astragal Ltd. • Azur 58001 • Tel. +972 3 -559 1660
IND	ifm electronic India Branch Office • Kolhapur, 416234 • Tel. +91 231-267 27 70
J	efector co., ltd. • Chiba-shi, Chiba 261-7118 • Tel. +81 043-299-2070
MAL	ifm electronic Pte. Ltd • 47100 Puchong Selangor • Tel. +603 8063 9522
MEX	ifm efector S. de R. L. de C. V. • Monterrey, N. L. 64630 • Tel. +52 81 8040-3535
N	Sivilingeniør J. F. Knudtzen A/S • 1396 Billingstad • Tel. +47 66 / 98 33 50
NA	ifm elctronic (pty) Ltd • 25 Dr. W. Kulz Street Windhoek • Tel. +264 61 300984
NL	ifm electronic b.v. • 3843 GA Harderwijk • Tel. +31 341 / 438 438
NZ	ifm efector pty ltd • 930 Great South Road Penrose, Auckland • Tel. +64 95 79 69 91
P	ifm electronic s.a. • 4410-136 São Félix da Marinha • Tel. +351 223 / 71 71 08
PL	ifm electronic Sp. z o.o. • 40-106 Katowice • Tel. +48 32-608 74 54
RA, ROU	ifm electronic s.r.l. • 1107 Buenos Aires • Tel. +54 11 / 5353 3436
RO	ifm electronic s.r.l • Sibiu 557260 • Tel. +40 269 224550
ROK	ifm electronic Ltd. • 140-884 Seoul • Tel. +82 2 / 790 5610
RUS	ifm electronic • 105318 Moscow • Tel. +7 495 921-44-14
S	ifm electronic a b • 41250 Göteborg • Tel. +46 31 / 750 23 00
SGP	ifm electronic Pte. Ltd. • Singapore 609 916 • Tel. +65 6562 8661/2/3
SK	ifm electronic s.r.o. • 835 54 Bratislava • Tel. +421 2 / 44 87 23 29
THA	SCM Allianze Co., Ltd. • Bangkok 10 400 • Tel. +66 02 615 4888
TR	ifm electronic Ltd. Sti. • 34381 Sisli/Istanbul • Tel. +90 212 / 210 50 80
UA	TOV ifm electronic • 02660 Kiev • Tel. +380 44 501 8543
USA	ifm efector inc. • Exton, PA 19341 • Tel. +1 610 / 5 24-2000
VN	ifm electronic • Ho Chi Minh city 700000 • Tel. +84-8-35125177
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • Tel. +27 12 345 44 49