



编程手册

SmartController

CR2530

CR2532

运行系统 $\geq$ V03.00.03

CODESYS[®] $\geq$ V2.3.9.33

简体中文

内容

1	关于本手册	5
1.1	版权	5
1.2	概述：ecomatmobile 装置文档模块	6
1.3	符号和格式是什么意思？	7
1.4	本文档的结构是怎样的？	8
1.5	说明沿革 (CR253n)	10
2	安全说明	11
2.1	请注意！	11
2.2	需要预先具备哪些知识？	12
2.3	控制器的启动运行状况	12
2.4	注意事项：序列号	12
3	系统描述	13
3.1	关于装置的信息	13
3.2	硬件说明	14
3.2.1	硬件结构	14
3.2.2	输入端（技术）	17
3.2.3	输出端（技术）	23
3.2.4	关于配线的注意事项	30
3.2.5	关于簧片继电器的安全说明	30
3.2.6	状态 LED	31
3.3	接口说明	32
3.3.1	CAN 接口	33
3.4	软件说明	34
3.4.1	装置的软件模块	35
3.4.2	CODESYS 项目的编程说明	37
3.4.3	工作状态	41
3.4.4	装置的性能极限	44
4	配置	47
4.1	设定运行时系统	48
4.1.1	重新安装运行时系统	48
4.1.2	更新运行时系统	50
4.1.3	检验安装	50

4.2	设定编程系统.....	51
4.2.1	手动设定编程系统.....	51
4.2.2	通过模板设定编程系统.....	55
4.3	一般功能配置.....	55
4.3.1	系统变量.....	55
4.4	输入端和输出端功能配置	56
4.4.1	配置输入端和输出端（默认设定）.....	56
4.4.2	配置输入端	57
4.4.3	配置输出端	62
4.5	变量.....	67
4.5.1	保留变量.....	67
4.5.2	网络变量.....	68
5	IFM 功能元件	69
5.1	针对装置 CR2530 的 IFM 库.....	69
5.1.1	库 ifm_CR2530_V03yyzz.LIB	70
5.1.2	库 ifm_RAWCan_NT_Vxxyzz.LIB	71
5.1.3	库 ifm_CANopen_NT_Vxxyzz.LIB.....	72
5.1.4	库 ifm_J1939_NT_Vxxyzz.LIB	73
5.2	针对装置 CR2530 的 IFM 功能元件.....	75
5.2.1	输出端功能元件.....	75
5.2.2	功能元件：RAW-CAN（第 2 层）.....	77
5.2.3	功能元件：CANopen	115
5.2.4	功能元件：SAE J1939.....	172
5.2.5	功能元件：处理输入值.....	216
5.2.6	功能元件：输出端功能.....	229
5.2.7	功能元件：系统.....	238
6	诊断和错误处理	265
6.1	诊断.....	265
6.2	故障.....	265
6.3	响应系统错误.....	266
6.3.1	响应错误消息的进程示例.....	266
6.4	CAN / CANopen: 错误和错误处理	266
7	附录	267
7.1	系统标志.....	268
7.2	地址分配和 I/O 工作模式.....	269
7.2.1	I/O 地址/变量.....	269
7.2.2	可能的输入端/输出端工作模式.....	272

7.3	集成 I/O 模块：说明	276
7.3.1	系统说明 I/O 模块 ExB01.....	276
7.3.2	I/O 模块的配置	291
7.3.3	集成 I/O 模块的对象目录	305
7.3.4	I/O 模块的运行	346
7.3.5	针对集成 ExB01 I/O 模块的系统标志.....	349
7.3.6	I/O 模块错误消息	350
7.4	错误表	353
7.4.1	错误标志	353
7.4.2	错误：CAN / CANopen	353
8	专业术语	355

9	指数	372
----------	-----------	------------

1 关于本手册

内容

版权.....	5
概述：ecomatmobile 装置文档模块	6
符号和格式是什么意思？	7
本文档的结构是怎样的？	8
说明沿革 (CR253n)	10

202

1.1 版权

46224

© ifm electronic gmbh 保留所有权利。 未经 ifm electronic gmbh 同意，不得复制或使本手册的任何部分。

我方页面上使用的所有产品名称、图片、公司或其他品牌均为各自权利所有者的资产：

- AS-i 是 AS 国际协会的资产，(→ www.as-interface.net)
- CAN 是德国 CiA (CAN in Automation e.V.) 的资产 (→ www.can-cia.org)
- CODESYS™ 是德国 3S - 即 Smart Software Solutions GmbH 的资产 (→ www.codesys.com)
- DeviceNet™ 是美国 ODVA™ (开放式设备网络供应商协会) 的资产 (→ www.odva.org)
- EtherNet/IP® 是 →ODVA™ 的资产
- EtherCAT® 由德国倍福自动化有限公司授权，是德国倍福自动化有限公司的注册商标和专利技术
- IO-Link® (→ www.io-link.com) 是德国 →PROFIBUS Nutzerorganisation e.V. 的资产 德国
- ISOBUS 是 AEF - 即农业电子基金会的资产 Deutschland (→ www.aef-online.org)
- Microsoft® 是美国微软公司的资产 (→ www.microsoft.com)
- Modbus® 是法国施耐德电气公司的资产 (→ www.schneider-electric.com)
- PROFIBUS® 是德国 PROFIBUS Nutzerorganisation e.V. 的资产 (→ www.profibus.com)
- PROFINET®→是德国PROFIBUS Nutzerorganisation e.V. 的资产
- Windows® 是美国→微软公司的资产

1.2 概述： ecomatmobile 装置文档模块

ecomatmobile 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] > CR2530 > [Technical data in PDF format]
	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。
	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
	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
	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。

*) 括号中的描述仅包含于特定装置的说明中。

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。	
	重要说明 如不遵守，可能导致故障或干扰
	信息 补充说明
 ...	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：

联系方式 → 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

50762

命令号 [hex]	命令号 [dec]	说明
0101	257	快速设定 AS-i Master 1/2
0103	259	更改用户语言
0104	260	更改显示设定
0105	261	设定输出控制
0106	262	设定 SPS 工作模式
0109	265	设定日期/时间
010A	266	配置 NTP 服务器设定
010B	267	读取日期/时间/NTP 设定
010C	268	重新启动系统
010D	269	读取现场总线信息（仅可在 CODESYS 中执行！）
010F	271	读取 OSC 条目的文本
0110	272	显示目标可视化

ⓘ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 CR2530 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 [276](#))，本文档附录。

1.5 说明沿革 (CR253n)

15326

本手册更改了哪些内容？概述

日期	主题	更改：
2014-02-03	集成 I/O 模块	CR2532 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
2014-04-29	FB CAN_REMOTE_RESPONSE	功能块 ENABLE 的更多精确说明
2014-05-12	CAN 限制	已针对 CAN、CANopen 和 CAN J1939 添加限制
2014-06-30	文档名称	“系统手册”重新命名为“编程手册”
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”
2014-08-08	章节“输入组 I1 (IN04...IN05)”	替换为涨价“电阻测量”
2014-08-08	FB PERIOD	已通过工作模式“相位测量”完成 (LZS V03.02.zz 或更高)
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
2014-11-12	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
2015-06-10	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
2015-08-04	输出端 OUT00...01	通过电流和电压测量诊断
2015-10-22	针对输入 IN12...IN15 的工作模式	二进制输入 = 模式 01 (而不是模式 10)
2016-03-02	FB OUTPUT、PWM1000	FB 输入通道说明已自改
2016-04-18	FB INPUT	针对电阻输入而修正的值 (MODE=18)
2016-04-27	快速输入端的 FB	添加了频率较高时的注意事项
2017-01-13	CODESYS 2.3 软件手册	从 IFM 主页下载的提示已移除
2018-07-09	ifm 分支机构列表	已移除
2018/10/10	集成 I/O 模块 ExB01 的输出	关于添加的 VBB1/VBB2 的提示

2 安全说明

内容

请注意！	11
需要预先具备哪些知识？	12
控制器的启动运行状况	12
注意事项：序列号	12

43976

2.1 请注意！

6091
11779

本手册中提供的信息、注意事项和范例，并不对特性作出保证。 图纸、表示方式和范例不对系统的假设负责，且不予考虑应用特定的特殊性。

- ▶ 机器/设备的制造商有责任确保机器/设备的安全性。
- ▶ 遵循机器/安装所要上市的国家/地区的国家和国际规定！

警告

不遵守这些说明可能造成财产损失或人员受伤。

ifm electronic gmbh 拒绝为这些方面负责。

- ▶ 在针对本装置或借助本装置进行工作前，涉事员工必须通读并理解安全说明和本手册中的对应章节。
- ▶ 涉事员工必须获得使用机器/设备的授权。
- ▶ 涉事员工必须持有进行该工作所需的相关资质，经过所需的培训。
- ▶ 遵循控制器的技术数据！
您可在 **ifm** 网站找到当前的技术数据。
- ▶ 请注意了解控制器的功能和特性的同时，同样留意控制器的安装和接线！
→ 提供的安装说明或在 **ifm** 网站上。

主页 → www.ifm.com

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”！

例如，重新启动可能由如下原因造成：

- 断电后的供电恢复
- 由于周期时间过长，在看门狗响应后复位
- E-stop 后错误排除

为确保安全的控制器行为：

- ▶ 在应用程序中监测电源供应。
- ▶ 如有错误，在应用程序中关闭所有相关输出。
- ▶ 在应用中对可能导致危险运动的执行器进行额外监控（反馈）。
- ▶ 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 注意事项：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。

3 系统描述

内容	
关于装置的信息	13
硬件说明	14
接口说明	32
软件说明	34

975

3.1 关于装置的信息

15329

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

- SmartController: CR2530

3.2 硬件说明

内容	
硬件结构	14
输入端（技术）	17
输出端（技术）	23
关于配线的注意事项	30
关于簧片继电器的安全说明	30
状态 LED	31

14081

3.2.1 硬件结构

内容	
条件.....	15
原理方块图.....	15
可用内存	16

15332

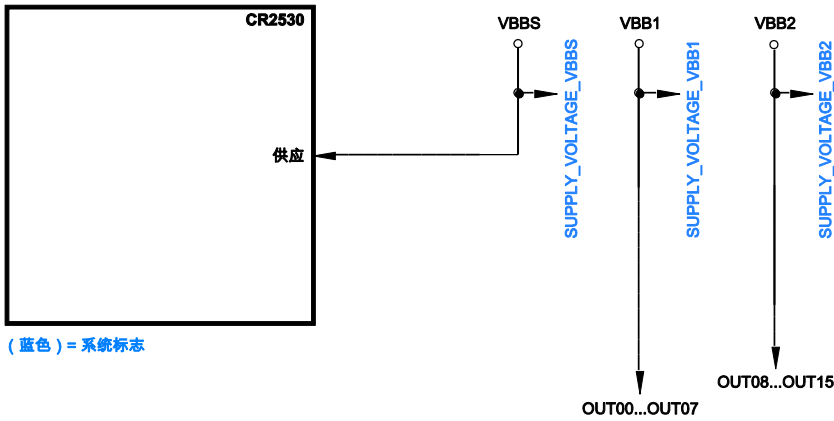
条件

1376

为电源接头 VBBs 提供充足的电压前，装置不会启动。
电压 > 8 V 即视为充足。
允许的工作电压 → 数据表

原理方块图

1377



图：电源方块图

可用内存

13736

FLASH-Speicher

13053

FLASH 内存（非易失慢速内存） 装置现有全部	1 536 千字节
-----------------------------	-----------

由此，如下内存区域将预留用于 ...

应用程序最大大小	128 千字节
除应用程序之外的数据 通过 FB 读取数据 FLASH_READ (→ 页 241) (文件：就标头而言少 128 字节)	64 千字节

剩余的内存保留用于系统内部用途。

SRAM

14027

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。	592K 字节
---	---------

由此，如下内存区域将预留用于 ...

应用程序保留的数据	128K 字节
-----------	---------

剩余的内存保留用于系统内部用途。

FRAM

2262

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。	2K 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	128 字节
固定为剩余定义标志 (%MB0...127)	128 字节

剩余的内存保留用于系统内部用途。

3.2.2 输入端 (技术)

内容	
模拟输入端.....	18
二进制输入端	19
输入组 IN00...IN03	20
输入组 IN04...IN05	20
输入组 IN06...IN11	22
输入组 IN12...IN15	22

14090

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：

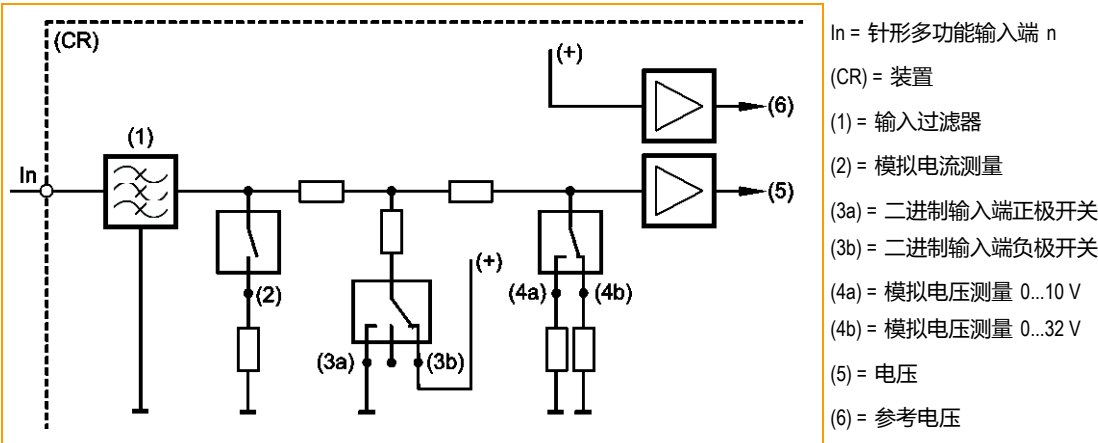
- 电流输入 0...20 mA
- 电压输入 0...10 V
- 电压输入 0...32 V
- 电阻测量 16...30 000 Ω (GND 测量)

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。

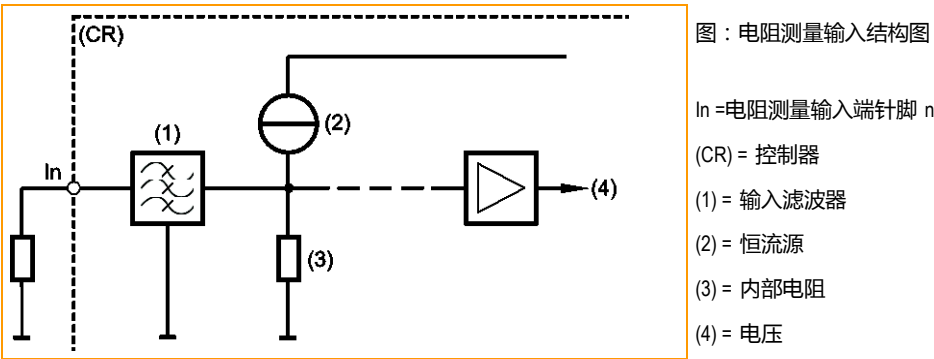
⚠ 如果是比率测量, 则连接的传感器应供给装置 VBBs。 因此, 避免了偏移电压导致的错误测量。

8971



图：多功能输出端原理方块图

8972



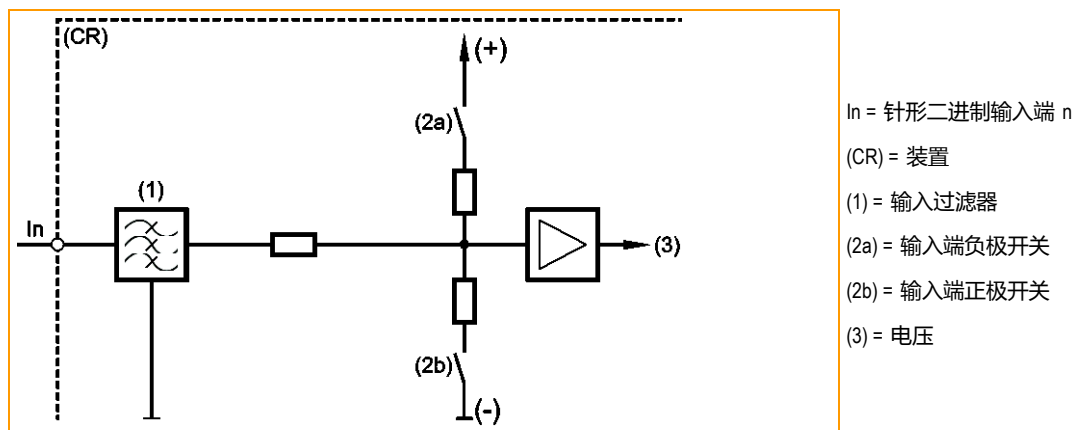
图：电阻测量输入结构图

二进制输入端

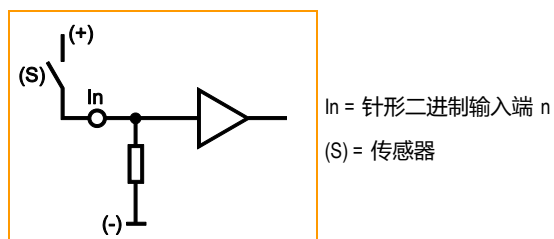
二进制输入端可在以下模式下运行：

- 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，开关信号可直接用作位信息。



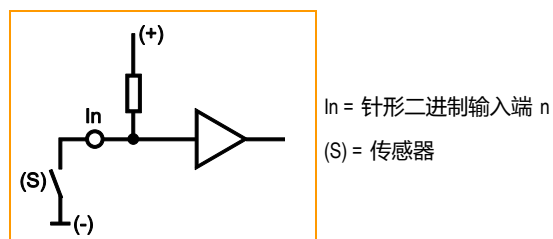
图：针对负极性和正极性传感器信号的二进制输入端负极开关/正极开关基本电路



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 $\Rightarrow$ 信号 = 低 (GND)



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 $\Rightarrow$ 信号 = 高 (电源)

就这些输入端中的某些而言（ $\rightarrow$ 技术资料），可选择要切换到的电势。

输入组 IN00...IN03

15339

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

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

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应用程序进行配置：

- FB **INPUT** (→ 页 [223](#)) > 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...32 VDC)，超过终值 (23 mA，且持续 ≥ 40 ms) 时，在功能块 INPUT 中相应地设定输出端 RESULT。在大约一秒钟后，输入端自动切回至电流测量范围。

输入组 IN04...IN05

15341

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）（含/不含诊断）

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

► 每个输入端均可通过应用程序进行配置：

- FB **INPUT** (→ 页 [223](#)) > 输入端 MODE

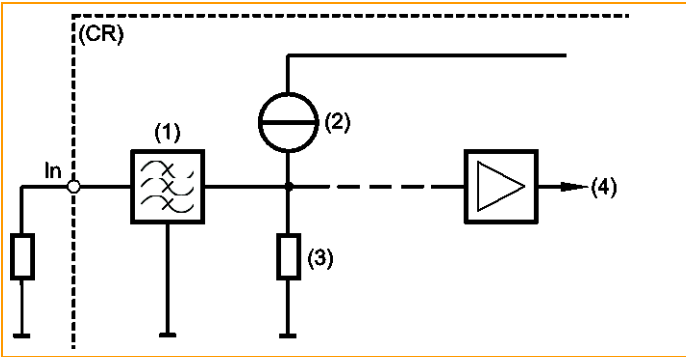
电阻测量

9773

这些输入端上的一般传感器：

- 液位
- 温度 (PT1000、NTC)

8972



图：电阻测量输入结构图

In =电阻测量输入端引脚 n

(CR) = 控制器

(1) = 输入滤波器

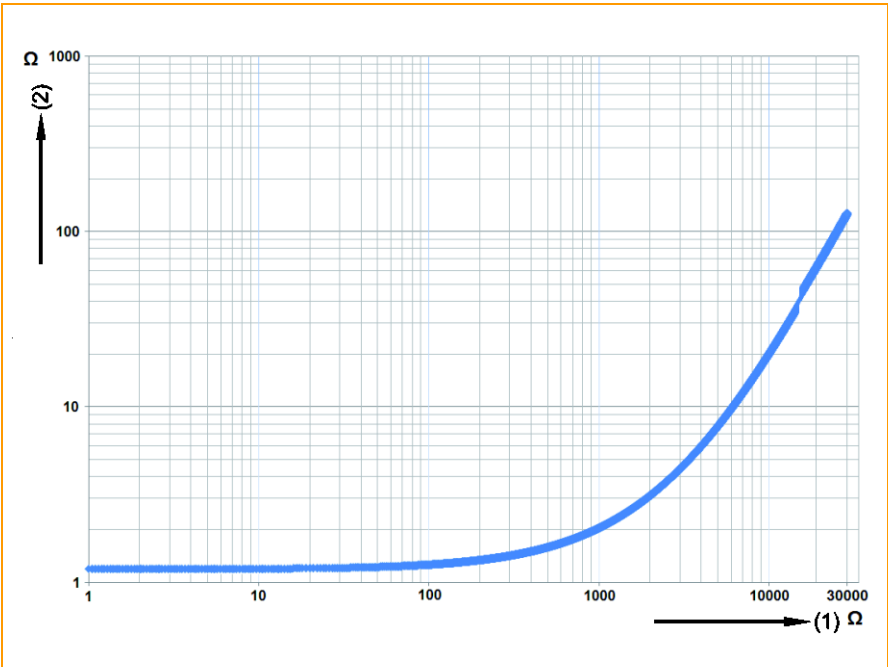
(2) = 恒流源

(3) = 内部电阻

(4) = 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。

输入组 IN06...IN11

15344

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 **可能的输入端/输出端工作模式** (→ 页 [272](#))

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** (→ 页 [223](#)) > 输入端 MODE

输入组 IN12...IN15

15346

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 **可能的输入端/输出端工作模式** (→ 页 [272](#))

所有输入端显示关于功能和诊断的相同状况。

 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** (→ 页 [223](#)) > 输入端 MODE

3.2.3 输出端（技术）

内容	
输出端的保护功能.....	23
输出组 OUT0、OUT1	25
输出组 OUT02...OUT07	27
输出组 OUT08...OUT09	28
输出组 OUT10...OUT11	28
输出组 OUT12...OUT15	29

14093

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

15644

拥有诊断能力的所有输出端均可检测短路。

诊断可通过电压反馈和电流反馈进行。

短路的定义为...

“输出电压下降至相应供电电压的 93.5% 以下（测量值 ± 2.0%）。”

> 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

15253

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：

- FB **OUTPUT** (→ 页 233) > 输入端 MODE = 16

这也只默认设定，但前提条件是使用控制配置中的标志。

根据输出端工作模式的反应

15252

出现过载或短路时，输出端的行为会根据其工作模式而定(→ 功能块 **OUTPUT** (→ 页 233) > 输入模式)：

- MODE=2：高电位开关量输出：无诊断且无保护
> 输出端持续工作。
- MODE=15：带诊断的高电位开关量输出
> 检测到错误，且赋值给功能块 OUTPUT 的输出变量 RESULT：
例如：RESULT = 128、141、142 或 145。
结果根据输出端类型及输出端的电流和电压而定。
程序员可在程序中对错误进行反应。
- MODE=16：带诊断和保护的高电位开关量输出
> 检测到错误，且赋值给功能块 OUTPUT 的输出变量 RESULT。
> 相应输出端关闭。
> ❗输出端的逻辑状态不受该操作的影响！
▶ 重启：开关量输出逻辑上打开和关闭！

使用 PWM 或 CURRENT_CONTROL 时的反应

15254

使用 FB PWM 或 CURRENT_CONTROL 时，情况不同：

无诊断。该 **输出端的自我保护** (→ 页 24) 启用。

- ▶ 针对包含电流反馈的输出端：
在应用程序中针对输出端查询一般电流！
程序员有责任对事件作出反应。

输出组 OUT0、OUT1

15351

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

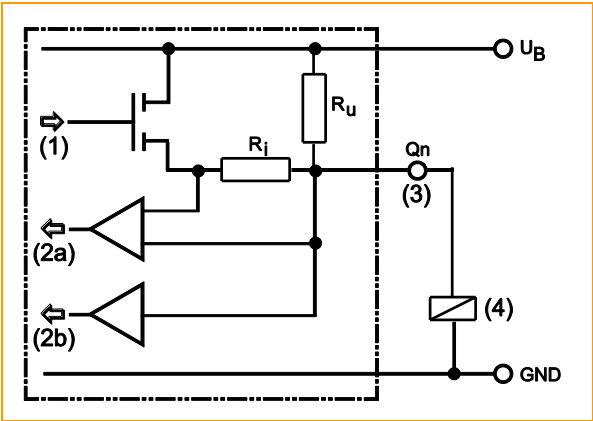
→ 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

- ▶ 每个输出端均可通过应用程序进行配置：
 - FB **OUTPUT** (→ 页 [233](#))> 输入端 MODE
 - PWM 输出端：→ FB **PWM1000** (→ 页 [236](#))
 - 电流控制和负载电流指示 → FB **CURRENT_CONTROL** (→ 页 [230](#))
- ▶  关于限值，请确保遵循技术资料！

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 R_u 将读回通道切换至 HIGH 电势（电源）。 若未断线，则低电阻负载 ($R_L < 10\text{ k}\Omega$) 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)

输出组 OUT02...OUT07

15353

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

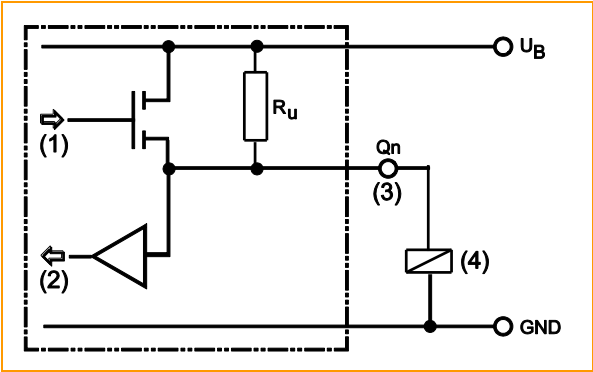
→ 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

- ▶ 每个输出端均可通过应用程序进行配置：
 - FB **OUTPUT** (→ 页 [233](#))> 输入端 MODE
 - PWM 输出端：→ FB **PWM1000** (→ 页 [236](#))
- ▶  关于限值，请确保遵循技术资料！

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)

输出组 OUT08...OUT09

15355

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
- 这些输出端不含诊断功能。

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

► 每个输出端均可通过应用程序进行配置：

→ FB **OUTPUT** (→ 页 [233](#))> 输入端 MODE

PWM 输出端：→ FB **PWM1000** (→ 页 [236](#))

►  关于限值，请确保遵循技术资料！

输出组 OUT10...OUT11

15362

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
- 这些输出端不含诊断功能。

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

► 每个输出端均可通过应用程序进行配置：

→ FB **OUTPUT** (→ 页 [233](#))> 输入端 MODE

PWM 输出端：→ FB **PWM1000** (→ 页 [236](#))

►  关于限值，请确保遵循技术资料！

输出组 OUT12...OUT15

15364

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)
- 这些输出端不含诊断功能。

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

►  关于限值，请确保遵循技术资料！

3.2.4 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

3.2.5 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 R_E 的 5 %（→ 技术资料）。否则，信号将不会检测为 TRUE。

例如：

$R_E = 3\,000\text{ Ohm}$

⇒ 最大串联电阻器 = 150 Ohm

3.2.6 状态 LED

20646

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
		
橙色	短暂亮起	初始化或复位检查
	 （时间帧 = 200 ms）	
绿色	以 5 Hz 的频率闪烁	未加载运行时系统
	 （时间帧 = 200 ms）	
绿色	以 2 Hz 的频率闪烁	应用程序 = RUN
	 （时间帧 = 200 ms）	
红色	以 5 Hz 的频率闪烁	应用程序 = 因电压过低而停止
	 （时间帧 = 200 ms）	
红色	永久亮起	系统错误 (FATAL ERROR) ; 应用程序 = STOP
		

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

15481

可通过 SET_LED 在应用程序中更改状态 LED 的频率和颜色。

❗ 在应用程序中使用 LED 功能块，则无需在 RUN 状态下对状态 LED 进行系统设定。

3.3 接口说明

内容	
CAN 接口	33

14098

3.3.1 CAN 接口

内容

CAN: 接口和协议	33
------------------	----

14101

连接和数据 → 技术资料

CAN: 接口和协议

14589
15238

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：

- RAW-CAN (第 2 层) 第 2 级上的 CAN → 章节 **功能元件：RAW-CAN (第 2 层)** (→ 页 [77](#))
- CANopen 主站/CANopen 从站 (→ 章节 **功能元件：CANopen** (→ 页 [115](#)))
- CANopen 网络变量 (通过 CODESYS) (→ 章节 **网络变量** (→ 页 [68](#)))
- SAE J1939 (针对驱动管理，→ 章节 **功能元件：SAE J1939** (→ 页 [172](#)))
- 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

14591

以下 CAN 接口和 CAN 协议可用于本 **ecomatmobile** 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	接口不存在	接口不存在
	CANopen	CANopen		
	SAE J1939	SAE J1939		

标准波特率 = 250 Kbits/s

 所有 CAN 接口均可同时兼容所有 CAN 协议。使用的 ID 不得彼此冲突！

3.4 软件说明

内容	
装置的软件模块	35
CODESYS 项目的编程说明	37
工作状态	41
装置的性能极限	44

14107

3.4.1 装置的软件模块

内容

Bootloader	35
操作系统	36
应用程序	36
库	36

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 [52](#))

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。

操作系统

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。

→ 章节 **装置的软件模块** (→ 页 35)

交付时，控制器中一般未加载操作系统 (LED 以 5 Hz 的频率闪烁绿色)。 在该工作模式中，引导程序启用。 它提供加载操作系统所需的 (如 CAN) 的最低限度功能。

通常只需下载一次操作系统。 应用程序可下载至控制器 (也可多次)，不会影响操作系统。

操作系统随文档一起提供，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

15409

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_CR2530_Vxxyzz.LIB	装置特定库 必须始终包含于应用程序！
ifm_RawCAN_NT_Vxxyzz.LIB	(可选) 装置 CAN 接口搭配 CAN 第 2 层运行时
ifm_CANopen_NT_Vxxyzz.LIB	(可选) 装置 CAN 接口作为 CANopen 主站或 CANopen 从站运行时
ifm_J1939_NT_Vxxyzz.LIB	(可选) 装置 CAN 接口为了与发动机控制器通信时

Details: → chapter **针对装置 CR2530 的 IFM 库** (→ 页 69)

3.4.2 CODESYS 项目的编程说明

内容	
CODESYS 中的 FB、FUN、PRG	38
注意周期时间！	39
关于装置编程的重要说明	39
创建应用程序	40
使用 ifm maintenance 工具	41
应用程序的分发	41

7426

您可在此了解如何对控制器进行编程。

► 请参阅 CODESYS 编程手册中的说明。

CODESYS 中的 FB、FUN、PRG

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 应用程序最多包含 8000 个功能元件；在本装置中，最多 512 个功能元件！

背景

功能的所有变数...

- 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，这意味着功能中的功能块调用。

- 每次有一次额外的初始化调用，且
- 上次调用的数据丢失。

注意周期时间！

8006

就 **ecomatmobile** 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。

关于装置编程的重要说明

20763

适用于以下装置：

- BasicController relay CR0431
- ▶ 编程时，连连接头 B:1 (VBB15) 和 B:8 (VBBS)。否则，不可编程。

背景

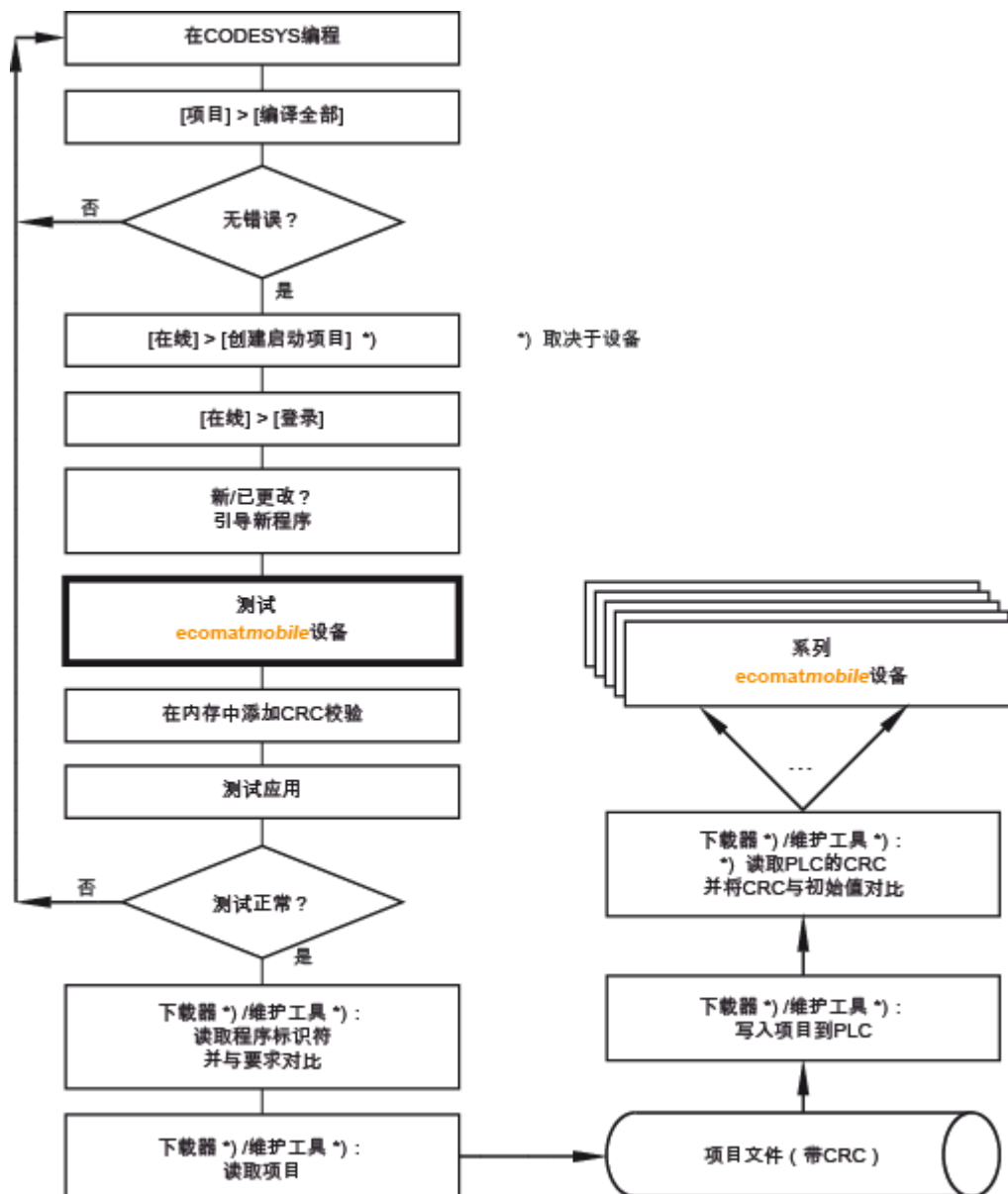
- 编程开始时，控制器复位所有输出端，还有 SUPPLY_SWITCH。
- 若无 VBB15，控制器将断开电源电压并关闭。
- 当控制器再次开启时，装置处于 Bootloader 模式。
程序员需再次将 BasicSystem 加载至装置。
而后，再次将应用程序加载至装置。

创建应用程序

应用程序由 CODESYS 2.3 编程系统生成，并在程序开发期间多次下载至控制器，以便进行测试：

在 CODESYS 中：[Online] > [Login] > 下载新的程序。

就每一次经由 CODESYS 2.3 的下载而言，会再次转化源代码。结果是，每次在控制器内存中形成新的校验和。在软件发布之前，安全控制器也允许此过程。



图形：创建和分发软件

使用 ifm maintenance 工具

8492

ifm maintenance 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** maintenance 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** maintenance 工具：

主页 → www.ifm.com

应用程序的分发

8493

如果要应用程序软件复制到量产机器并使用，则建议采用以下顺序：

- 保存软件
完成程序开发后，利用 **IFM** 维护工具加载至控制器的应用程序最新版本需从控制器读取并利用名称 `project_file.RESX` 保存在数据载体上。仅该进程可确保应用程序软件及其校验和得以存储。
- 下载软件。
若要为所有量产机器配备相同的软件，则仅可利用 **IFM** 维护工具将该文件加载至控制器。
- 利用 **IFM** 维护工具再次加载时，集成校验和会自动识别该文件数据的错误。

3.4.3 工作状态

1075

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

- BOOTLOADER
- INIT
- STOP
- RUN
- SYSTEM STOP

INIT 状态（复位）

20647

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 呈橙色亮起。

该状态可转换至以下状态之一：

- RUN
- STOP

STOP 状态

8288

在以下情况下可转换至该状态：

- 如果未加载任何应用程序，则可从 INIT 状态转换。
- 如果满足以下条件，则可从 RUN 状态转换：
 - STOP 命令通过 CODESYS 接口发送。

在 STOP 状态中：

- > 装置的输入端关闭。
- > 应用程序的处理停止。
- > LED 呈绿色亮起。

该状态可转换至以下状态之一：

- RUN
- ERROR
- FATAL ERROR
- INIT（通电复位后）

RUN 状态

8287

在以下情况下可转换至该状态：

- 如果满足以下条件，则可从 INIT 状态（自动启动）转换：
 - 工作电压达到最小值。且：
 - 应用程序存在。
- 从 STOP 状态：
 - 通过 CODESYS 命令 RUN。
 - 工作电压达到或超过最小值。

在 RUN 状态中：

- > 运行时系统正在运行。
- > 应用程序正在运行。
- > LED 以 2 Hz 的频率闪烁绿色。

应用程序可对 LED 进行不同的控制 → FB **SET_LED** (→ 页 [259](#)).

该状态可转换至以下状态之一：

- INIT（通电复位后）
- STOP
- ERROR
- FATAL ERROR

ERROR 状态

8290

在以下情况下可转换至该状态：

- 如果电源电压太低。

在 ERROR 状态中：

- > 装置的输入端关闭。
- > 应用程序的处理停止。
- > 系统参数保存。
- > LED 以 5 Hz 的频率闪烁红色。

该状态可转换至以下状态之一：

- INIT（通电复位后）
- RUN
- STOP
- FATAL ERROR

FATAL ERROR 状态

8289

在以下情况下可转换至该状态：

- 内存错误（RAM/闪存）
- 异常错误
- 运行时系统错误

在 FATAL ERROR 状态中：

- > 装置的输入端关闭。
- > 应用程序终止。
- > 运行时系统终止。
- > LED 呈红色亮起。

该状态可转换至以下状态之一：

- INIT（通电复位后）

3.4.4 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

15365

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 (100 ms)：

- > 装置切换至“超时错误”状态
- > 所有进程停止（复位）
- > 所有输出关闭
- > 状态 LED 以 10 Hz 的频率闪烁红色

排除故障：

- ▶ 删除应用程序！
- ▶ 通电复位
- ▶ 重新将应用程序加载至装置

如果相应的电子狗出现故障：

- > 第二个电子狗会让装置切换至“严重错误”状态
- > LED 呈红色亮起

排除故障：

- ▶ 通电复位

如果不成功：

- ▶ 转至 Bootloader
- ▶ 通电复位
- ▶ 重新将运行时系统加载至装置
- ▶ 重新将应用程序加载至装置

如果不成功：

- ▶ 硬件错误：将装置寄至 IFM ！

本装置 CAN 的限制

17975

① 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）。

有些 Raw-CAN 功能元件有助于传送和接收一个 PLC 周期中的多个消息，消息暂时存储在 FiFo 中。

- CAN_TX..., → 功能元件：传送 RAW-CAN 数据
- **CAN_RX_ENH_FIFO** (→ 页 [91](#))
- **CAN_RX_RANGE_FIFO** (→ 页 [97](#))

FiFo 消息的数量是有限的。控制器存在如下限制：

标准	控制器	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n ioControl: CR205n SmartController: CR253n	PDM360 NG: CR108n, CR120n
最大 FiFo 传输			
- 采用功能库 CAN_TX...		4 条消息	4 条消息
- 采用功能库 CAN_TX_ENH...		16 条消息	16 条消息
最大 FiFo 接收			
- 采用功能块 CAN_RX_..._FIFO		32 条消息	32 条消息

本装置 CANopen 的限制

17976

控制器存在如下限制：

标准	控制器	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n ioControl: CR205n SmartController: CR253n	PDM360 NG: CR108n, CR120n
最大警告错误		32 条消息	128 条消息
最大 SDO 数据		2 048 字节	2 048 字节

本装置 CAN J1939 的限制

17977

控制器存在如下限制：

控制器	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n ioControl: CR205n SmartController: CR253n	PDM360 NG: CR108n, CR120n
标准		
最大 FiFo 传输 - 采用功能块 J1939_TX - 采用功能块 J1939_TX_ENH	4 条消息 16 条消息	4 条消息 16 条消息
最大 FiFo 接收 - 采用功能块 J1939_RX_FIFO	32 条消息	32 条消息
最大 DTCs	64 条消息	64 条消息
最大数据 J1939	1 785 字节	1 785 字节

4 配置

内容

设定运行时系统	48
设定编程系统	51
一般功能配置	55
输入端和输出端功能配置	56
变量.....	67

18065
1016

相应安装说明或本文档 **附录** (→ 页 [267](#))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

①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 CR2530 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 [276](#))，本文档附录。

4.1 设定运行时系统

内容	
重新安装运行时系统	48
更新运行时系统	50
检验安装	50

14091

4.1.1 重新安装运行时系统

14635
8486

交付 **ecomatmobile** 控制器时，一般未加载操作系统（LED 以 5 Hz 的频率闪烁绿色）。在该工作模式中，仅引导程序启用。它提供加载操作系统所需的（如 RS232、CAN）的最低限度功能。

通常只需下载一次操作系统。应用程序可下载至控制器（也可多次），不会影响操作系统。

操作系统随文档一起提供，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

8651

通知

数据有丢失的风险！

如在数据传输中发生电源故障，数据可能丢失，致使控制器不再可使用。维修仅可由 **ifm electronic** 进行。

► 请确保在数据传输时电源不间断！

①注意

必须始终使用适于所选目标的软件版本：

- 操作系统 (ifm_CR2530_Vxxyyzz.RESX) ,
- PLC 配置 (ifm_CR2530_Vxx.CFG) ,
- 控制器库 (ifm_CR2530_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 "CR2530") 和软件版本号 "xx" (例如 "01") 必须始终是相同的值！否则控制器会进入“停止”模式。

"yy" (版本号) 和 "zz" (修补号) 的值**不必**一致。

① 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)。

② 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。

联系方式 → www.ifm.com

操作系统通过单独的“Maintenance”工具软件传输至控制器。（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下载到控制器。但如果事先从控制器上传后，还可利用“Maintenance”工具加载。

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

14637

- ▶ 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！
 - 检查是否将正确的运行时系统加载至控制器！
- ▶ 第 1 次测试：
利用 IFM 维护工具测试是否加载正确的运行时系统版本：
 - 在装置中读取运行时系统的名称和版本！
 - 手动对比该信息和目标数据！
- ▶ 第 2 次测试（可选）：
在应用程序中检查是否加载正确的运行时系统版本：
 - 在装置中读取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_SW_INFO (→ 页 248)	传送关于装置系统软件的信息： • 软件名称， • 软件版本， • 版本号， • 版本日期
-----------------------	--

- ▶ 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	
手动设定编程系统.....	51
通过模板设定编程系统.....	55

14461

4.2.1 手动设定编程系统

内容	
设定目标	52
启用 PLC 配置	53
CAN 声明（例如 CR1080）	54

3963

设定目标

13136
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] 对话框 [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_CR2530_Vxxyyzz.LIB 并手动操作！

8485

ⓘ注意

必须始终使用适于所选目标的软件版本：

- 操作系统 (ifm_CR2530_Vxxyyzz.RESX) ,
- PLC 配置 (ifm_CR2530_Vxx.CFG) ,
- 控制器库 (ifm_CR2530_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 "CR2530"）和软件版本号 "xx"（例如 "01"）必须始终是相同的值！ 否则控制器会进入“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。

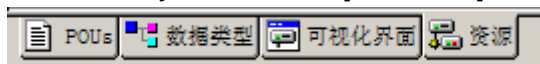
联系方式 → www.ifm.com

启用 PLC 配置

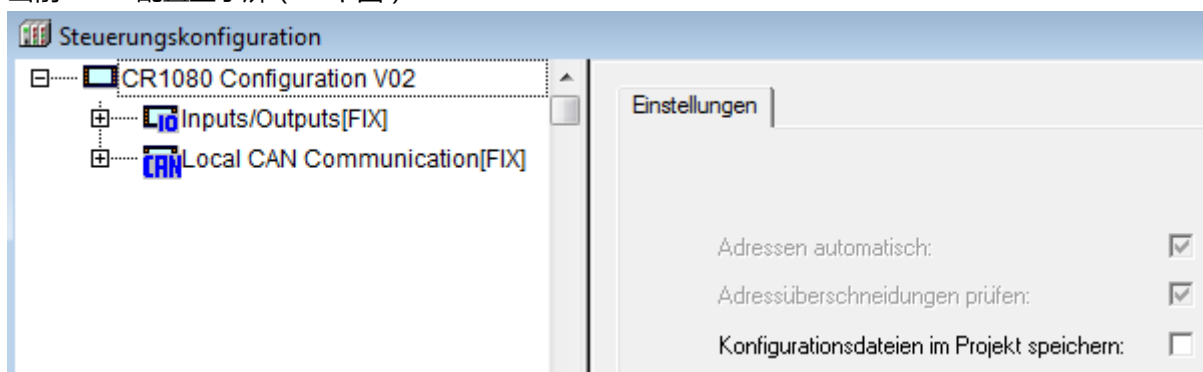
PLC 配置通过目标系统自动加载。PLC 配置映射 CODESYS 中文件 CR2530.cfg 的内容。程序员可这样轻松访问预定义的系统 and 错误标志，输入端和输出端，以及装置的 CAN 接口。

访问 PLC 配置（如 CR1080）：

- ▶ 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）：



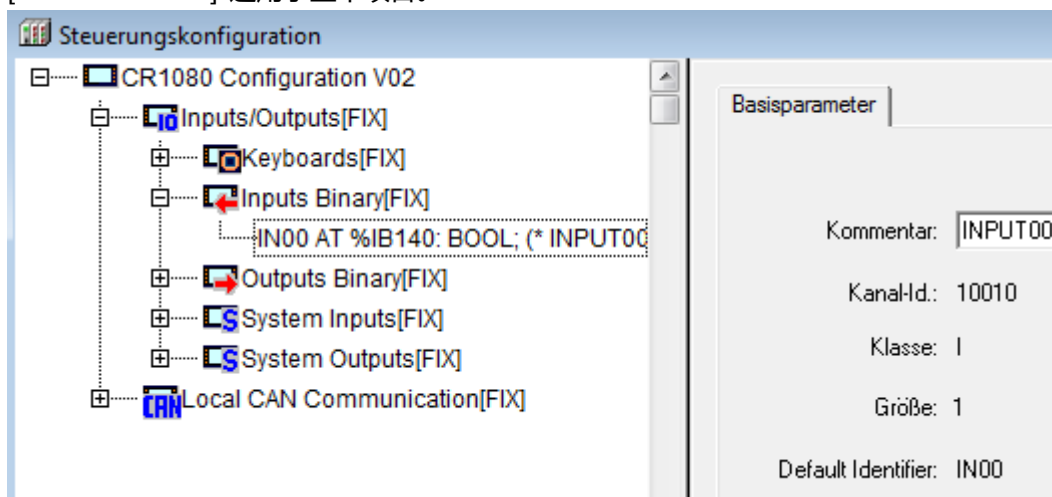
- > 根据配置，以下适用于针对用户的程序环境：

- 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

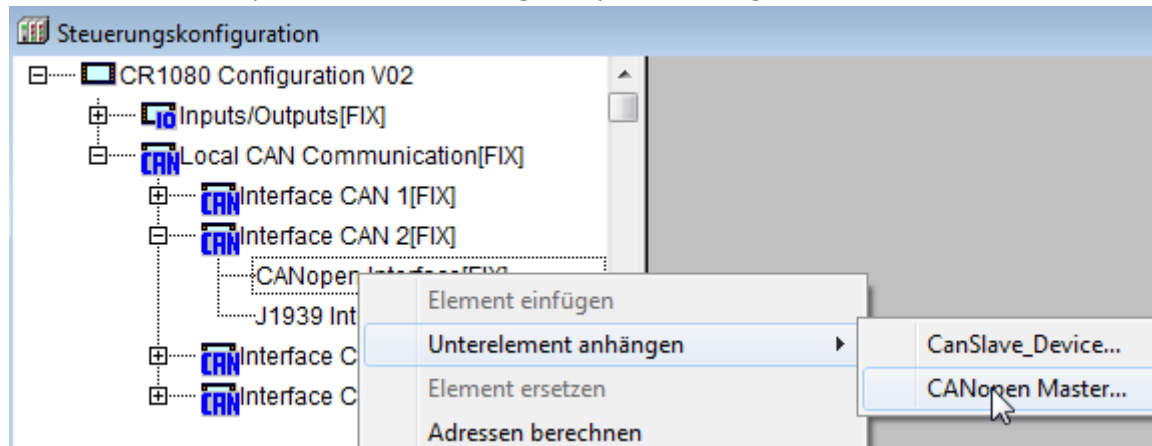
这些可在窗口 [PLC Configuration]（例如 → 下图）中通过符号直接指定（强烈建议！），且可作为 [Global Variables] 适用于整个项目。



CAN 声明（例如 CR1080）

在 CODESYS PLC 配置中，您现在需要声明 CAN 接口。

- ▶ 右击 PLC 配置的名称。所需 CAN 接口的 [CANopen Interface [FIX]]。
- ▶ 单击 [Append Subelement]。
- ▶ 即便装置作为 CANopen 从站运行：单击 [CANopen Master...]：

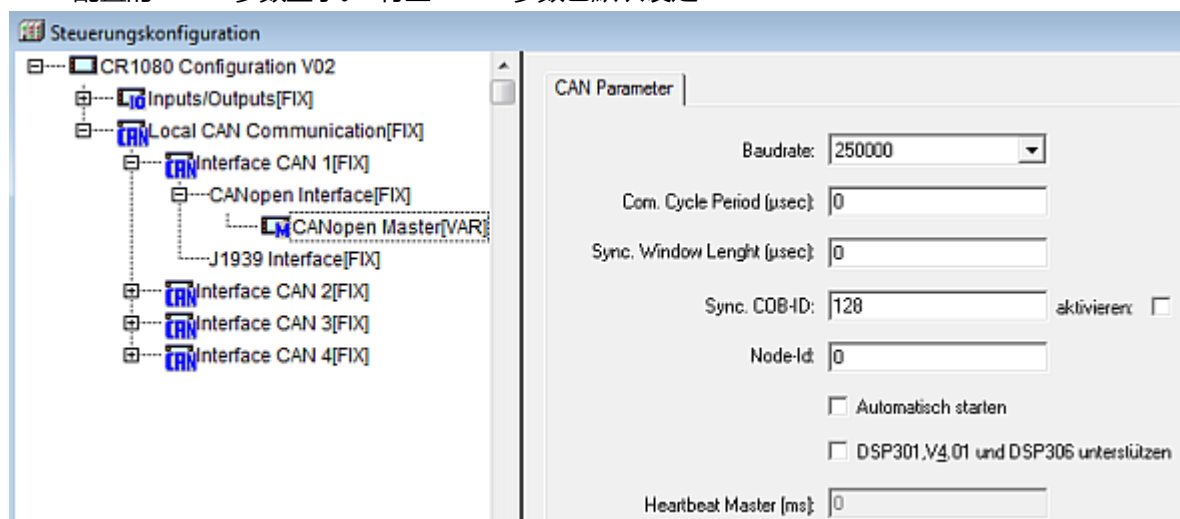


信息

如果装置作为从站运行，则亦可选择 [CanSlave_Device]。

为使得作为主站的配置更简单，亦可使用所有 CAN 第 2 层和网络变量功能。

- > PLC 配置的 CAN 参数显示。有些 CAN 参数已默认设定：



- ▶ 如果装置在 CAN 第 2 层运行，或通过网络变量或 CAN_RX / CAN_TX 作为从站运行：
 - ❗ 检查是否针对装置设定正确的波特率（所有参与者的波特率必须相同）。
- ▶ 如果装置作为 CANopen 主站运行：
 - 检查所有参数设定。

- ▶ 关闭窗口 [PLC Configuration]。
- ▶ 在菜单 [File] > [Save as...] 中 为项目指定合理的名称并将其保存在所需目录中。
- ▶  在应用程序中始终针对 CAN 接口 **CANOPEN_ENABLE** (→ 页 [116](#)) 调用 FB 自身的实例！

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 **ecomatmobile** 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 系统变量

15576

所有系统变量（→ 章节 **系统标志** (→ 页 [268](#))）均已定义地址且不可更改。

4.4 输入端和输出端功能配置

内容	
配置输入端和输出端（默认设定）	56
配置输入端.....	57
配置输出端.....	62

7995
1394

就 **ecomatmobile** 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_CR2530_Vxxyzz.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 [272](#)).

4.4.1 配置输入端和输出端（默认设定）

2249

- 交付时所有输入端和输出端处于二进制模式（正极开关！）。
- 诊断功能未启用。
- 过载保护启用。

4.4.2 配置输入端

内容	
关于簧片继电器的安全说明	57
模拟输入端： 配置和诊断	58
开关量输入：配置和诊断.....	60
快速输入端.....	61

3973

有效的工作模式 → 章节 **可能的输入端/输出端工作模式** (→ 页 [272](#))

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

模拟输入端： 配置和诊断

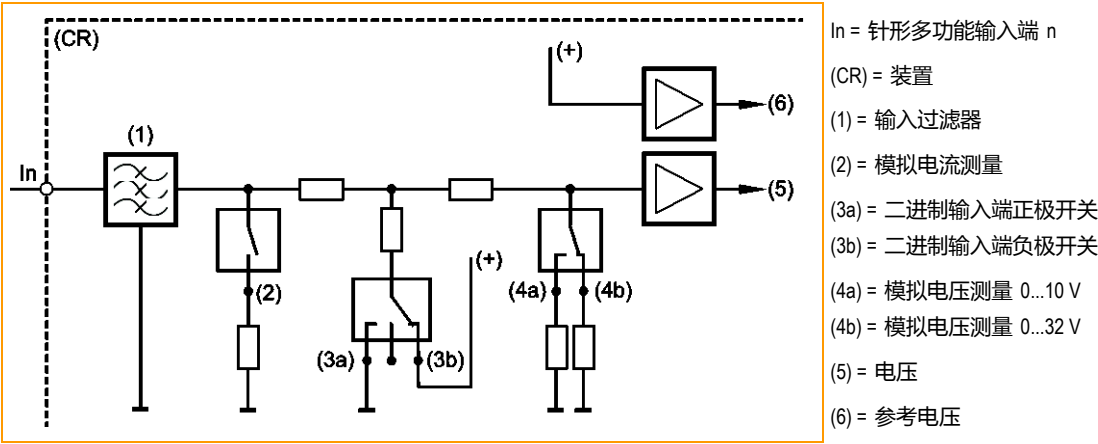
14656

每个输入端均可通过应用程序进行配置：

- FB **INPUT** (→ 页 223) > 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...32 VDC)，超过终值 (23 mA，且持续 ≥ 40 ms) 时，在功能块 INPUT 中相应地设定输出端 RESULT。 在大约一秒钟后，输入端自动切回至电流测量范围。

作为备选，还可通过二进制的方式评估模拟通道。

8971



图： 多功能输出端原理方块图

配置输入端软件过滤器

15418

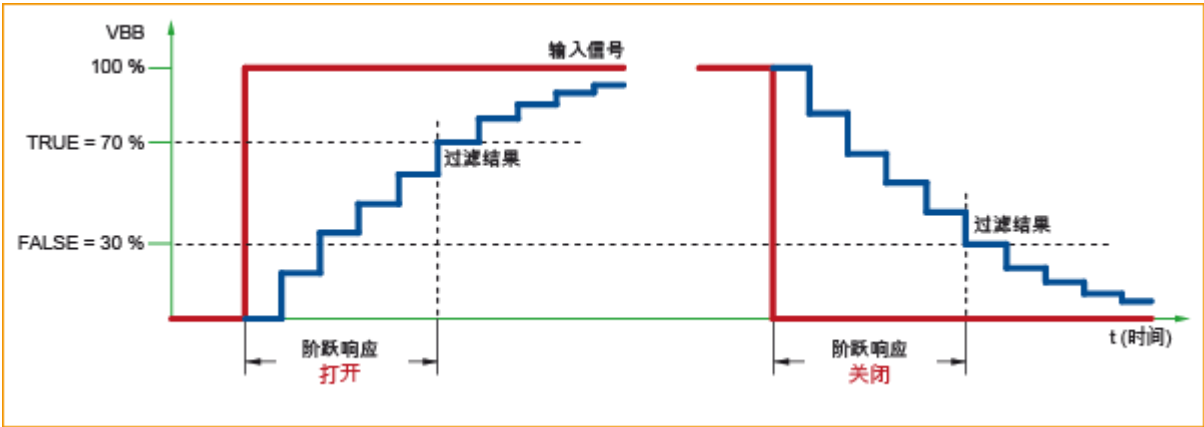
可通过 FB 中的输入端 **FILTER INPUT** (→ 页 223) 配置软件过滤器，过滤模拟输入端的测量输入电压。过滤器跟低通滤波器一样工作；过滤频率通过 FILTER 输入的值设定。就 FILTER 而言，0...8 的值是允许的。

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 %	
0	过滤禁用				
1	120	2	4	7	
2	47	5	9	17	
3	22	10	18	35	
4	10	19	36	72	推荐
5	5	38	73	146	
6	2.5	77	147	293	
7	1.2	154	294	588	
8	0.7	308	589	1177	

以下阶跃响应的表现相关：

- 模拟输入端：0...90 % 和 0...99 %
- 二进制输入端：0...70 %



图：开启/关闭后输入端二进制信号的时间进程

开关量输入：配置和诊断

14672

每个输入端均可通过应用程序进行配置：

- FB **INPUT** (→ 页 223) > 输入端 MODE

MODE	BYTE	输入通道工作模式：	
		0 = 0x00	关闭
		3 = 0x03	电压输入 0...10 000 mV
		6 = 0x06	电压输入比率计 0...1 000 ‰
		7 = 0x07	电流输入 0...20 000 µA
		9 = 0x09	电压输入 0...32 000 mV
		10 = 0x0A	(仅针对模拟评估输入) 二进制输入，正极开关 (BL)
		11 = 0x0B	(仅针对模拟评估输入) 二进制输入，正极开关 (BI)，包含诊断功能 (Namur)
		12 = 0x0C	二进制输入，负极开关 (BH)
		18 = 0x12	电阻输入 16...30 000 Ω

启用输入端诊断

7352

如果要使用诊断，则需额外启用。

- 通过功能块的输入端 MODE 设定输入端的模式 **INPUT** (→ 页 223).
- > FB **INPUT** (→ 页 223) 在 RESULT 输出端提供输入端的诊断消息。

非电子开关的 NAMUR 型带诊断的开关量输入：
► 为开关配置额外的电阻连接件！

图：输入端 Inn 非电子开关 S

 这些输入端可使用包含诊断功能的 NAMUR 传感器。
在此情况下，无需额外的电阻器连接。

快速输入端

8292

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）。

14677

❗ 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。

适当的功能块包括：

FASTCOUNT (→ 页 217)	针对快速输入脉冲的计数功能块
INC_ENCODER (→ 页 219)	针对编码器评估的递增/递减计数器功能
PERIOD (→ 页 226)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）

🔧 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。

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！

4.4.3 配置输出端

内容	
配置输出端软件过滤器.....	63
二进制输出端： 配置和诊断.....	64
PWM 输出端.....	65

3976

有效的工作模式 → 章节 可能的输入端/输出端工作模式 (→ 页 [272](#))

配置输出端软件过滤器

15421

可通过 FB 中的输入端 FILTER **OUTPUT** (→ 页 233) 配置软件过滤器，过滤 PWM 输出端的测量输出电流。

FILTER 字节仅适于包含电流测量的输出端。

针对不含电流测量的输出端：设定 FILTER = 0 !

输出端电流会在一个 PWM 周期内加以平均。

如果抖动设定，则电流在抖动周期内加以平均。

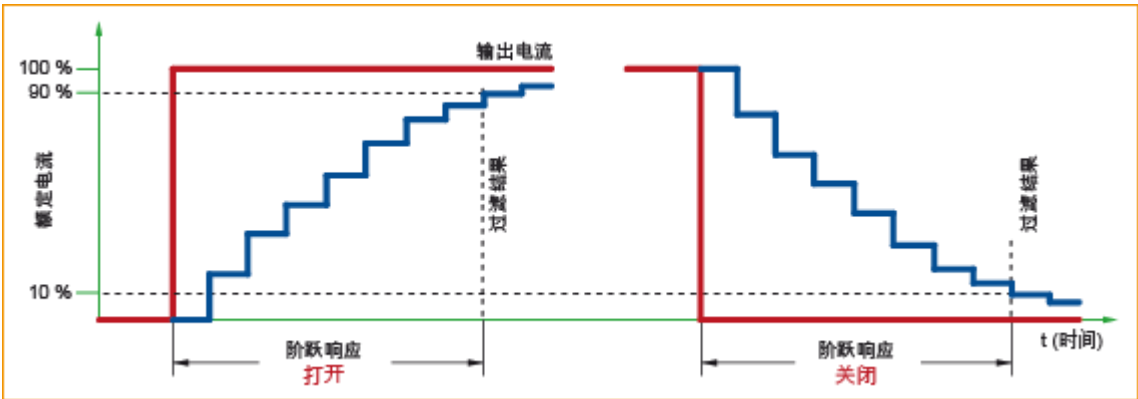
过滤器跟低通滤波器一样工作；极限频率通过 FILTER 输入的值设定。就 FILTER 而言，0...8 的值是允许的。

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 %	
0	过滤禁用				不含电流测量的输出端
1	600	0.8	1.0	1.4	
2	233	1.8	2.2	3.4	
3	109	3.6	4.6	7.0	
4	52	7.2	9.4	14.4	推荐
5	26	14.6	19.0	29.2	
6	13	29.4	38.2	58.6	
7	6	58.8	76.4	117.6	
8	4	117.8	153.2	235.4	

以下阶跃响应的表现相关：

- 输出电流： 0...90 % and 0...99 %



图：开启/关闭后输出端二进制电流信号时间序列

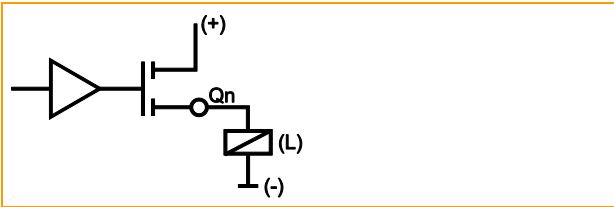
二进制输出端：配置和诊断

14689

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

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

- 每个输出端均可通过应用程序进行配置：
→ FB **OUTPUT** (→ 页 [233](#))> 输入端 MODE.

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。

输出端诊断的配置

8301

如果要使用诊断，则需额外启用。

- 如果将输出端作为包含诊断的二进制输出端（→ 数据表）：
→ FB OUTPUT > 输入端 MODE = 15 或 18
- > FB **OUTPUT** (→ 页 [233](#)) 在 RESULT 输出端提供输出端的诊断消息。

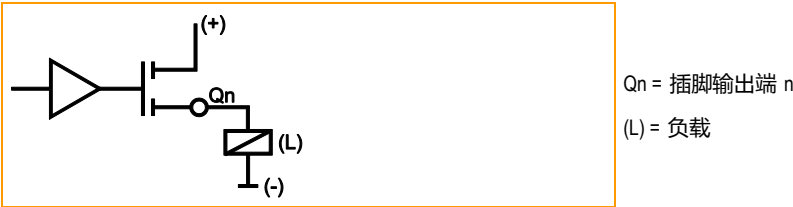
PWM 输出端

14705

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



输出端正极开关 (BH) 基本电路
针对正极性输出端信号

15414

 **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
- 过载保护 OUT_OVERLOAD_PROTECTION 未启用

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 可用性

15371

装置	可用 PWM 输出端的数量	电流控制 (PWMi) 数量	PWM 频率 [Hz]
SmartController: CR2530	12	2	20...250

针对 PWM 功能的 FB

14718

以下功能块适用于输出端的 PWM 功能：

CURRENT_CONTROL (→ 页 230)	PWMi 输出通道的电流控制
PWM1000 (→ 页 236)	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

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。

4.5 变量

内容

保留变量	67
网络变量	68

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

8672

保留变量可自动保存在受保护的内存区域，重启期间可自动再次加载。

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）。



保存保留变量

9853

在装置中，数据类型 RETAIN 仅在运行时期间储存在易失内存 (RAM) 中。

为永久保存数据，会在每个周期结束时自动保存于 FRAM 内存 ¹⁾。

¹⁾ FRAM 在此表示所有类型的非易失快速内存。

ⓘ 注意

在本装置中，切勿使用 3S 库 SysLibPlcCtrl.lib 中的以下功能：

- FUN SysSaveRetains
- FUN SysRestoreRetains

读回保留变量

9854

通电后以及第一个程序周期之间，装置自动将保存的数据写回至运行的内存一次。为此，无需将额外的 FB 集成于应用程序。

⚠注意

在本装置中，切勿使用 3S 库 SysLibPlcCtrl.lib 中的以下功能：

- FUN SysSaveRetains
- FUN SysRestoreRetains

4.5.2 网络变量

15242
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
- ifm_NetVarLib_NT_Vxxyyzz.lib

5 IFM 功能元件

内容

针对装置 CR2530 的 IFM 库	69
针对装置 CR2530 的 IFM 功能元件	75

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 CR2530 的 IFM 库

内容

库 ifm_CR2530_V03yyzz.LIB	70
库 ifm_RAWCan_NT_Vxxyzz.LIB	71
库 ifm_CANopen_NT_Vxxyzz.LIB	72
库 ifm_J1939_NT_Vxxyzz.LIB	73

14235

5.1.1 库 ifm_CR2530_V03yyzz.LIB

此为装置库。该 **IFM** 库包含以下功能块：

功能元件	简短说明
FASTCOUNT (→ 页 217)	针对快速输入脉冲的计数功能块
FLASH_READ (→ 页 241)	直接将不同数据类型从闪存传输到 RAM
GET_APP_INFO (→ 页 243)	传送关于存储在装置上的应用程序的信息： • 应用程序名称， • 应用程序版本， • 唯一的 CODESYS 版本号， • CODESYS 版本日期
GET_HW_INFO (→ 页 245)	传送关于装置硬件的信息： • IFM 产品编号（例如 CR0403）， • 产品名称， • 明确的序列号， • 硬件版本， • 生产日期
GET_IDENTITY (→ 页 247)	读取存储在装置上的应用程序的 ID (之前已通过以下方式保存： SET_IDENTITY (→ 页 258))
GET_SW_INFO (→ 页 248)	传送关于装置系统软件的信息： • 软件名称， • 软件版本， • 版本号， • 版本日期
GET_SW_VERSION (→ 页 250)	传送关于存储在装置上的软件版本的信息： • BasicSystem 版本， • Bootloader 版本， • SIS 版本， • 应用程序版本， • 用户闪存版本
INC_ENCODER (→ 页 219)	针对编码器评估的递增/递减计数器功能
INPUT (→ 页 223)	将工作模式分配至输入通道 提供所选通道的当前状态
MEM_ERROR (→ 页 252)	利用信号通知某些参数或内存中的错误 (再次) 初始化系统资源
MEMCPY (→ 页 254)	直接写入和读取内存中的不同数据类型
OHC (→ 页 256)	可调工作时间计数器 (0...3)
OUTPUT (→ 页 233)	将工作模式分配至输出通道 提供所选通道的当前状态
PERIOD (→ 页 226)	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
PWM1000 (→ 页 236)	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_IDENTITY (→ 页 258)	设定应用程序特定程序 ID
SET_LED (→ 页 259)	在应用程序中更改状态 LED 的频率和颜色
SET_PASSWORD (→ 页 262)	设定程序和内存上传访问控制的用户密码
TIMER_READ_US (→ 页 264)	读取当前系统时间，以 [μs] 为单位 最大值 = 1h 11min 34s 967ms 295μs

5.1.2 库 ifm_RAWCan_NT_Vxxyyzz.LIB

14715

该 **IFM** 库包含以下功能块：

功能元件	简短说明
CAN_ENABLE (→ 页 78)	初始化所示 CAN 接口 配置 CAN 波特率
CAN_RECOVER (→ 页 80)	启用/禁用自动总线关闭处理 总线关闭时重新启动 CAN 接口
CAN_REMOTE_REQUEST (→ 页 110)	发送相应的请求并将其他装置的响应作为结果返回
CAN_REMOTE_RESPONSE (→ 页 112)	将数据提供给装置中的 CAN 控制器，该数据作为对远程消息的请求的响应自动发送。
CAN_RX (→ 页 86)	配置数据接收对象并读取数据对象的接收缓冲区
CAN_RX_ENH (→ 页 88)	- 配置数据接收对象并读取数据对象的接收缓冲区 - 帧类型和掩码可选
CAN_RX_ENH_FIFO (→ 页 91)	- 配置数据接收对象并读取数据对象的接收缓冲区 - 帧类型和掩码可选 - 每个周期会有几条 CAN 消息
CAN_RX_RANGE (→ 页 94)	- 配置一系列数据接收对象并读取数据对象的接收缓冲区 - 帧类型和掩码可选
CAN_RX_RANGE_FIFO (→ 页 97)	- 配置一系列数据接收对象并读取数据对象的接收缓冲区 - 帧类型和掩码可选 - 每个周期会有几条 CAN 消息
CAN_SETDOWNLOADID (→ 页 81)	= 设定 CAN 下载 ID 设定 CAN 接口的下载 ID
CAN_STATUS (→ 页 83)	获取关于所选 CAN 总线的状态信息： BAUDRATE、DOWNLOAD_ID、BUSOFF、WARNING_RX、WARNING_TX、VERSION、BUSLOAD 并根据需要复位：BUSOFF, WARNING_RX, WARNING_TX

功能元件	简短说明
CAN_TX (→ 页 101)	将 CAN 数据对象 (消息) 传输至配置的 CAN 接口以在每次调用时传送
CAN_TX_ENH (→ 页 103)	将 CAN 数据对象 (消息) 传输至配置的 CAN 接口以在每次调用时传送 CAN 特定特性可设定
CAN_TX_ENH_CYCLIC (→ 页 106)	以循环的方式将 CAN 数据对象 (消息) 传输至配置的 CAN 接口以便传送 CAN 特定特性可设定

5.1.3 库 ifm_CANopen_NT_Vxxyyzz.LIB

14914

该 **IFM** 库包含以下功能块：

功能元件	简短说明
CANOPEN_ENABLE (→ 页 116)	初始化所示 CANopen 主站接口 配置 CAN 波特率
CANOPEN_GETBUFFERFLAGS (→ 页 118)	= CANopen 获取缓冲标志 提供关于缓冲标志的信息 标志可通过可选输入端复位。
CANOPEN_GETEMCYMESSAGES (→ 页 165)	= 获取 CANopen 紧急报文 列出自上次删除消息以来控制器从网络其他节点接收的所有紧急报文。 该列表可通过设置相应的输入端复位。
CANOPEN_GETERRORREGISTER (→ 页 167)	= 获取 CANopen 错误寄存器 从控制器读取错误寄存器 0x1001 和 0x1003 寄存器可通过设置相应的输入端复位。
CANOPEN_GETGUARDHBERRLIST (→ 页 160)	= 获取 CANopen 保护和检测信号错误列表 批量列出主站检测到错误的所有节点： 保护错误，检测信号错误 该列表可通过设定相应的输入端复位。
CANOPEN_GETGUARDHBSTATSLV (→ 页 162)	= CANopen 从站获取保护和检测信号状态 将以下状态的信号发送至从属运行的控制器： 节点保护监控，检测信号监控 通过信号发送的错误可通过设定相应的输入端复位。
CANOPEN_GETNMTSTATESLAVE (→ 页 126)	= CANopen 从站获取网络管理状态 发送节点网络运行状态的信号
CANOPEN_GETODCHANGEDFLAG (→ 页 132)	= 获取对象目录更改标志 报告特定对象目录条目的任何值更改
CANOPEN_GETSTATE (→ 页 120)	= CANopen 设定状态 请求主站、从属装置或网络中特定节点的参数
CANOPEN_GETSYNCSTATE (→ 页 155)	= CANopen 获取 SYNC 状态 • 读取 SYNC 功能的设定 (启用/未启用) • 读取 SYNC 功能的错误状态 (SyncError)

功能元件	简短说明
CANOPEN_NMTSERVICES (→ 页 128)	= CANopen 网络管理服务 更新内部节点状态 且视 NMT 命令条目而定 : • 触发 NMT 命令或 • 触发节点初始化
CANOPEN_READOBJECTDICT (→ 页 134)	= CANopen 读取对象目录 从装置的对象目录读取配置数据
CANOPEN_SDOREAD (→ 页 139)	= CANopen 读取 SDO 读取 "Expedited SDO" = 加速服务数据对象
CANOPEN_SDOREADBLOCK (→ 页 141)	= CANopen 读取 SDO 块 通过 SDO 块传输读取网络节点对象目录中所示的条目
CANOPEN_SDOREADMULTI (→ 页 144)	= CANopen 读取 SDO multi 读取网络节点对象目录中所示的条目
CANOPEN_SDOWRITE (→ 页 146)	= SDO 写入 写入 "Expedited SDO" = 加速服务数据对象
CANOPEN_SDOWRITEBLOCK (→ 页 148)	= CANopen 写入 SDO block 通过 SDO 块传输写入网络节点对象目录中所示的条目
CANOPEN_SDOWRITEMULTI (→ 页 151)	= CANopen 写入 SDO multi 写入网络节点对象目录中所示的条目
CANOPEN_SENDEMCMYMESSAGE (→ 页 169)	= CANopen 发送紧急报文 发送 EMCY 报文 报文通过相应的参数集合而成并输入寄存器 0x1003
CANOPEN_SETSTATE (→ 页 123)	= CANopen 设定状态 设定主站、从属装置或网络中特定节点的参数
CANOPEN_SETSYNCSTATE (→ 页 157)	= CANopen 设定 SYNC 状态 开启和关闭 SYNC 功能
CANOPEN_WRITEOBJECTDICT (→ 页 136)	= CANopen 写入对象目录 将配置数据写入装置的对象目录

5.1.4 库 ifm_J1939_NT_Vxxyyzz.LIB

14912

该 **IFM** 库包含以下功能块：

功能元件	简短说明
J1939_DM1RX (→ 页 207)	J1939 诊断消息 1 RX 从其他 ECU 接收诊断消息 DM1 或 DM2
J1939_DM1TX (→ 页 210)	J1939 诊断消息 1 TX 将有效错误消息传送至 CAN 堆栈
J1939_DM1TX_CFG (→ 页 213)	J1939 诊断消息 1 TX 可配置 CAN 堆栈不发送循环 DM1 "zero active faults" 消息

功能元件	简短说明
J1939_DM3TX (→ 页 215)	J1939 诊断消息 3 TX 删除装置上的无效 DTC (DM2)
J1939_ENABLE (→ 页 173)	初始化 J1939 堆栈
J1939_GETDABYNAME (→ 页 175)	= 获取目标任意名称 通过名称信息确定一个或多个参与者的目标地址
J1939_NAME (→ 页 178)	向装置提供名称以在网络中识别
J1939_RX (→ 页 190)	接受单帧消息 显示 CAN 总线最后读取的消息
J1939_RX_FIFO (→ 页 192)	= 包含 FIFO 的 J1939 RX 接受所有特定消息并接着从 FiFo 读取
J1939_RX_MULTI (→ 页 194)	= J1939 RX 多帧消息 接收多帧消息
J1939_SPEC_REQ (→ 页 185)	= J1939 特定请求 请求和接收来自其他控制器的特定消息
J1939_SPEC_REQ_MULTI (→ 页 187)	= J1939 特定请求多帧消息 请求和接收来自其他控制器的特定多帧消息
J1939_STATUS (→ 页 182)	显示 J1939 堆栈的相关信息
J1939_TX (→ 页 197)	发送单个单帧消息
J1939_TX_ENH (→ 页 199)	= J1939 TX 增强 发送单个单帧消息 还可设定：传输优先级、数据长度
J1939_TX_ENH_CYCLIC (→ 页 201)	= J1939 TX 增强循环 循环发送单帧消息 还可设定：传输优先级、数据长度、周期
J1939_TX_ENH_MULTI (→ 页 203)	= J1939 TX 增强多帧消息 发送单个多帧消息

5.2 针对装置 CR2530 的 IFM 功能元件

内容

输出端功能元件	75
功能元件：RAW-CAN (第 2 层)	77
功能元件：CANopen	115
功能元件：SAE J1939	172
功能元件：处理输入值	216
功能元件：输出端功能	229
功能元件：系统	238

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 输出端功能元件

8354
7556

有些功能元件返回 RESULT 消息。

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1...31		全局返回值；示例：
1	01	FB 执行完成且无错误 – 数据有效
4	04	FB 正在处理中 - 数据循环处理
5	05	FB 正在处理中 - 仍在接收
6	06	FB 正在处理中 - 仍在发送
7	07	FB 正在处理中 - ID 远程启用中
8	08	功能块启用中
14	0E	FB 已启用 CANopen 管理器配置装置并发送 SDO
15	0F	FB 已启用 CANopen 管理器启动
32 ₁₀ ...63		FB 特定返回值
64 ₁₀ ...127		FB 特定错误消息
128 ₁₀ ...255		全局错误消息；示例：

值 十进制 十六进制		说明
238	EE	错误：CANopen 配置太大，无法启动
239	EF	错误：CANopen 管理器无法启动
240	F0	错误：多个模式输入端已启用 例如 CANopen NTM 服务
241	F1	错误：CANopen 状态转换未获准许
242	F2	错误：不可设定
247	F7	错误：超过了内存（长度大于数组）
250	FA	错误：FIFO 已满 – 数据丢失
252	FC	错误：CAN 多帧传送失败
253	FD	错误：CAN 传送失败。 数据无法传送。
255	FF	错误：内存不足以使用多帧

5.2.2 功能元件：RAW-CAN (第 2 层)

内容	
功能元件：RAW-CAN 状态	77
功能元件：接收 RAW-CAN 数据	85
功能元件：传送 RAW-CAN 数据	100
功能元件：RAW-CAN 远程	109

15051

我们在此讲述应用程序中使用的易福门电子 RAW-CAN 功能块 (CAN 第 2 层)。

功能元件：RAW-CAN 状态

内容	
CAN_ENABLE	78
CAN_RECOVER	80
CAN_SETDOWNLOADID	81
CAN_STATUS	83

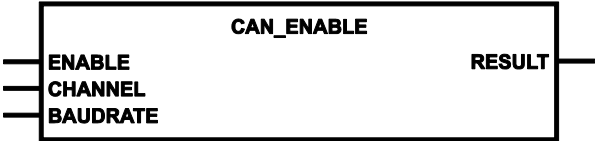
15049

CAN_ENABLE

7492

单元类型 = 功能块 (FB)
元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7494

CAN 硬件通过 CAN_ENABLE 初始化。若无此调用，不可在 RAW-CAN 中进行其他调用，否则会返回一个错误。

若要更改波特率，则需遵循以下步骤：

- ▶ 在一个周期内保持功能块满足 ENABLE=FALSE。
- > 所有协议复位。
- > 重新初始化 CAN 接口以及在该接口上运行的 CAN 协议。任何可用于循环传送的信息亦丢失，必须重新创建。
- > ENABLE=TRUE 更新时，采用新的比特率。

输入端参数

7495

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: 启用 CAN 接口 FALSE: 禁用 CAN 接口
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
波特率	WORD := 250	波特率 [kbits/s] 可允许 = 20、50、100、125、250、500、1000

输出端参数

8530

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
8	08	功能块启用中
9	09	CAN 未启用
242	F2	错误：不可设定

CAN_RECOVER

7512

单元类型 = 功能块 (FB)
元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7513

CAN_RECOVER has the following tasks:

- 启用/禁用自动总线关闭处理
- 总线关闭时重新启动 CAN 接口

> 如果总线关闭：CAN 控制器删除所有缓冲区（包括其他协议的缓冲区）

如果 CAN_RECOVER 未使用 (ENABLE=FALSE)：

- > 如果总线关闭，则在 1 秒后自动进行恢复尝试。
- > 连续 4 次恢复尝试后，受影响的 CAN 接口禁用。

输入端参数

7514

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: CAN 总线关闭后无自动恢复 FALSE: CAN 总线关闭后自动恢复
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
RECOVER_EXECUTE	BOOL	TRUE（仅针对 1 个周期）： CAN 接口重新启动 修正总线关闭状况 FALSE: 功能元件未执行
InhibitTime (参数可选)	TIME := T#1s	总线关闭和 CAN 接口重启之间的等待时间

CAN_SETDOWNLOADID

7516

= 设定下载 ID

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7517

连接运行时系统和 CODESYS 开发环境时，数据交换需要下载 ID。 启动装置时，利用硬件配置中的默认值设定下载 ID。

该值可通过 CAN_SETDOWNLOADID 在 PLC 程序中设定（如利用特定输入端）。 更改的 ID 亦可写入硬件配置。

输入端参数

7519

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
DOWNLOAD_ID	BYTE	1...127 = 设定下载 ID 0 = 读取下载 ID

输出端参数

7520

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	功能块执行完成且无错误
8	08	功能块启用中
242	F2	错误：不可设定

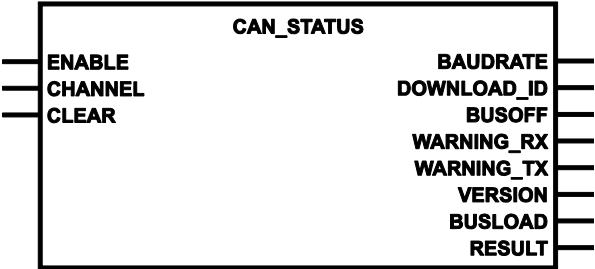
CAN_STATUS

7499

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7501

CAN_STATUS provides information on the chosen CAN bus.

若无硬件初始化，则可将以下标志复位为 FALSE：

- BUSOFF
- WARNING_RX
- WARNING_TX

输入端参数

7502

参数	数据类型	说明
ENABLE	BOOL := FALSE	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
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
CLEAR	BOOL := FALSE	TRUE: 复位以下标志： • WARNING_RX • WARNING_TX • BUSOFF FALSE: 功能元件未执行

输出端参数

7504

参数	数据类型	说明
BAUDRATE	WORD	CANopen 节点的当前波特率，以 [kBaud] 为单位
DOWNLOAD_ID	BYTE	当前下载 ID
BUSOFF	BOOL	接口错误 CAN BUS OFF
WARNING_RX	BOOL	接口超过接收警告阈值
WARNING_TX	BOOL	接口超过传送警告阈值
VERSION	DWORD	IFM CAN 堆栈库的 vabvab
BUSLOAD	BYTE	当前总线负载，按 [%] 计算
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	功能块执行完成且无错误
8	08	功能块启用中
9	09	CAN 未启用
242	F2	错误：不可设定

功能元件：接收 RAW-CAN 数据

内容	
CAN_RX	86
CAN_RX_ENH	88
CAN_RX_ENH_FIFO	91
CAN_RX_RANGE	94
CAN_RX_RANGE_FIFO	97

15050

CAN_RX

7586

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7588

CAN_RX 用于接收消息。
FB 仅限几个功能，所需内存空间小。
CAN_RX 针对设定的标识符进行过滤。 如果在一个周期中接收几个包含相同标识符的 CAN 消息，仅最后/最近的消息可用。

输入端参数

7589

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
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
ID	DWORD	数据对象标识符编号： 正常帧 (2 048 个 ID)： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (536 868 864 个 ID)： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF

输出端参数

7590

参数	数据类型	说明
DATA	ARRAY [0..7] OF BYTE	已接收数据，（ 1...8 字节）
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	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定

CAN_RX_ENH

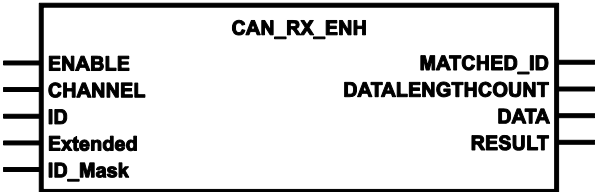
7606

= CAN RX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7608

此外，CAN_RX_ENH 提供以下可能性（完全不同于 CAN_RX (→ 页 86)):

- 选择帧类型（11 或 29 位），
- 定义掩码以评估 CAN ID。

ID 和掩码 的位比较：	如果 ID_MASK 位 = 0，那么 CAN-ID 位可能 = 0 或 1。 如果 ID_MASK 位 = 1，那么 CAN-ID 位必须 = ID 位。
-----------------	---

可通过掩码将几个标识符定义为过滤器。

例如：

ID =	0x100 = 0b0001 0000 0000
ID_MASK =	0x1F1 = 0b0001 1111 0001
结果	评估包含以下位模式的 CAN ID： 0bxxx1 0000 xxx0（x = 任何），即就此例而言（[hex] 中的全部）： 100, 102, 104, 106, 108, 10A, 10C, 10E, 300, 302, 304, 306, 308, 30A, 30C, 30E, 500, 502, 504, 506, 508, 50A, 50C, 50E, 700, 702, 704, 706, 708, 70A, 70C, 70E

输入端参数

7609

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
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
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
ID_Mask (参数可选)	DWORD := 0	标识符过滤器掩码 如果 ID_MASK 位 = 0, CAN ID 位可能 = 0 或 1 如果 ID_MASK 位 = 1, CAN ID 位必须 = ID 位

输出端参数

7613

参数	数据类型	说明
MATCHED_ID	DWORD	数据对象标识符编号
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据，（ 1...8 字节 ）
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	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误： 不可设定

CAN_RX_ENH_FIFO

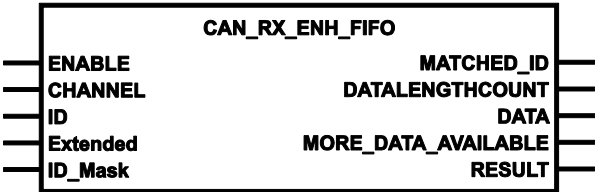
7615

= 包含 FiFo 的 CAN RX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7616

此外，CAN_RX_ENH_FIFO 提供针对已接收数据的 FiFo (完全不同于 **CAN_RX_ENH** (→ 页 88)). 因此，在一个周期中可接收多个 CAN 消息。

❗ FiFo 已满时不会出现覆盖。 入站消息将丢失。

在此事件中：

- ▶ 禁用 FB 并通过 ENABLE 重新启用。
- > FiFo 删除，且可重新填充。

过滤器掩码说明： →**CAN_RX_ENH** (→ 页 88) > 章节 **说明**

输入端参数

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
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
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0... $2^{29}-1$) FALSE: 正常帧 (ID = 0... $2^{11}-1$)
ID_Mask (参数可选)	DWORD := 0	标识符过滤器掩码 如果 ID_MASK 位 = 0, CAN ID 位可能 = 0 或 1 如果 ID_MASK 位 = 1, CAN ID 位必须 = ID 位

输出端参数

7617

参数	数据类型	说明
MATCHED_ID	DWORD	数据对象标识符编号
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
MORE_DATA_AVAILABLE	BOOL	TRUE: FiFo 中有进一步接收的数据 FALSE: FiFo 中无进一步接收的数据
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	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定
250	FA	错误：FiFo 已满 – 数据丢失

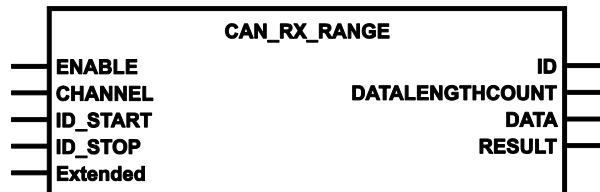
CAN_RX_RANGE

7592

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7594

CAN_RX_RANGE provides the following settings:

- 选择消息类型 (11 或 29 位) ,
- 定义标识符范围。

CAN_RX 针对设定的标识符进行过滤。 如果在一个周期中接收几个包含相同标识符的 CAN 消息, 仅最后/最近的消息可用。

输入端参数

7595

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID_START	DWORD	数据对象标识符范围的起始编号 : 正常帧 (2 ¹¹) : 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹) : 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	数据对象标识符范围的结束编号 : 正常帧 (2 ¹¹) : 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹) : 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)

输出端参数

7598

参数	数据类型	说明
ID	DWORD	数据对象标识符编号： 正常帧（2 048 个 ID）： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧（536 868 864 个 ID）： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据，（1..8 字节）
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	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定

CAN_RX_RANGE_FIFO

7601

= 包含 FiFo 的 CAN RX 范围

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7603

CAN_RX_RANGE_FIFO basically works like **CAN_RX_RANGE** (→ 页 94).

此外，CAN_RX_RANGE_FIFO 提供针对已接收数据的 FiFo。因此，在一个周期中可接收多个 CAN 消息。

❗ FiFo 已满时不会出现覆盖。入站消息将丢失。

在此事件中：

- 使用 ENABLE 以禁用和重新启用功能。
- > FiFo 删除，且可重新填充。

输入端参数

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID_START	DWORD	数据对象标识符范围的起始编号 : 正常帧 (2 ¹¹) : 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹) : 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	数据对象标识符范围的结束编号 : 正常帧 (2 ¹¹) : 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹) : 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)

输出端参数

7604

参数	数据类型	说明
ID	DWORD	数据对象标识符编号： 正常帧（2 048 个 ID）： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧（536 868 864 个 ID）： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据，（1..8 字节）
MORE_DATA_AVAILABLE	BOOL	TRUE: FiFo 中有进一步接收的数据 FALSE: FiFo 中无进一步接收的数据
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	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定
250	FA	错误：FiFo 已满 – 数据丢失

功能元件： 传送 RAW-CAN 数据

内容	
CAN_TX.....	101
CAN_TX_ENH.....	103
CAN_TX_ENH_CYCLIC.....	106

15055

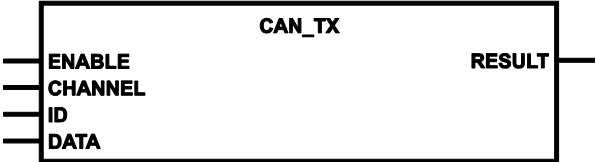
CAN_TX

7522

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7523

CAN_TX 会每周期发送一条标准消息。
FB 仅限几个功能，所需内存空间小。
> 如果在一个周期内多次调用该 FB 的实例，数据亦会发送多次。
如果是简单的功能 CAN_TX 和 CAN_RX，则通过 ID 确定发送标准帧还是扩展帧。若有增强版本，则通过输入端 EXTENDED 设定。因此 ID 范围 0...2047 中的扩展帧不可通过简单的功能发送。

输入端参数

7524

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
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
ID	DWORD	数据对象标识符编号： 正常帧 (2 048 个 ID)： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (536 868 864 个 ID)： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7527

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	功能块执行完成且无错误
242	F2	错误：不可设定
250	FA	错误：FIFO 已满 – 数据丢失

CAN_TX_ENH

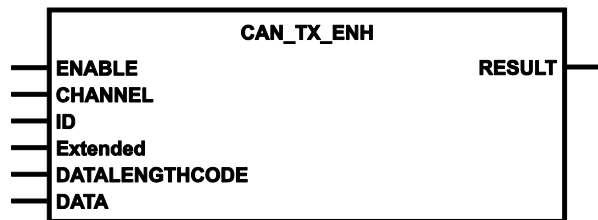
7558

= CAN TX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7559

其他设定选项通过 CAN_TX_ENH (针对: 增强) 提供。在此, 所有 CAN 特定特性均可单独设定, 例如

- 它是 11 位还是 29 位标识符?
- 可预设其他输入端, 以便 **CAN_TX** (→ 页 [101](#)) 不需要。
- > 如果在一个周期内多次调用该 FB 的实例, 数据亦会发送多次。

输入端参数

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: 执行该功能元件 边缘: 输入新的值。 FALSE: 单元未执行 复位内部远程值
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
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
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
DATALengthCODE	BYTE	= 数据长度码 待发送数据字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7527

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	功能块执行完成且无错误
242	F2	错误：不可设定
250	FA	错误：FiFo 已满 – 数据丢失

CAN_TX_ENH_CYCLIC

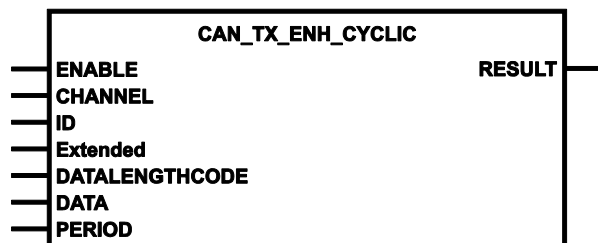
7568

= CAN TX 增强循环

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7569

CAN_TX_ENH_CYCLIC 用于次循环传送 CAN 消息。

否则, FB 相当于 **CAN_TX_ENH** (→ 页 [103](#)).

► 通过参数 PERIOD 设定周期。

❗ 如果周期太短, 则可能导致总线负载高, 进而影响整个系统的性能。

输入端参数

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
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
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
DataLengthCode (参数可选)	BYTE := 8	待发送数据的长度 (0...8 字节)
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)
PERIOD	TIME	周期

输出端参数

7510

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
8	08	功能块启用中
9	09	CAN 未启用
242	F2	错误：不可设定
250	FA	错误：FIFO 已满 – 数据丢失

功能元件： RAW-CAN 远程

内容	
CAN_REMOTE_REQUEST	110
CAN_REMOTE_RESPONSE.....	112

15057

CAN_REMOTE_REQUEST

7625

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7627

若要请求远程消息，则可通过 CAN_REMOTE_REQUEST 发送相应的要求，其他装置的响应将作为结果发回。

输入端参数

7628

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
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
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)

输出端参数

7629

参数	数据类型	说明
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据，（1...8 字节）
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	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定

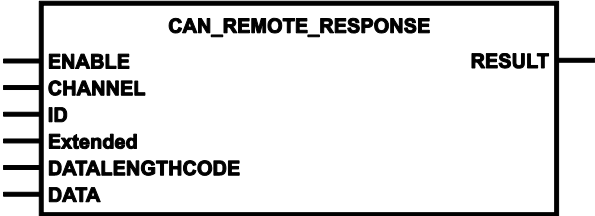
CAN_REMOTE_RESPONSE

7631

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7633

CAN_REMOTE_RESPONSE 将数据提供给装置中的 CAN 控制器，该数据在请求远程消息后自动发送。

该 FB 很大程度上取决于装置类型。 仅可设定有限的远程消息。

BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n	最多 40 个远程消息
PDM360 NG: CR108n, CR120n	最多 100 个远程消息

输入端参数

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: 执行该功能元件 边缘: 输入新的值。 FALSE: 单元未执行 复位内部远程值
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
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
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
DATALengthCODE	BYTE	= 数据长度码 待发送数据字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7636

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
6	06	FB 正在处理中 - ID 远程未启用
7	07	FB 正在处理中 - ID 远程启用中
8	08	功能块启用中
10	0A	没有进一步的远程条目可用
242	F2	错误：不可设定

5.2.3 功能元件：CANopen

内容

功能元件：CANopen 状态.....	115
功能元件：CANopen 网络管理	125
功能元件：CANopen 对象目录	131
功能元件：CANopen SDO.....	138
功能元件：CANopen SYNC	154
功能元件：CANopen 保护.....	159
功能元件：CANopen 紧急.....	164

15059

就 CANopen 而言，IFM 电子提供一系列功能元件，将在下文中解释。

功能元件：CANopen 状态

内容

CANOPEN_ENABLE.....	116
CANOPEN_GETBUFFERFLAGS	118
CANOPEN_GETSTATE.....	120
CANOPEN_SETSTATE	123

15061

CANOPEN_ENABLE

7785

单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7787

CANOPEN_ENABLE 允许开启或关闭 CANopen 主站。

► **!** 在应用程序中始终针对 CAN 接口 **CANOPEN_ENABLE** (→ 页 [116](#)) 调用 FB 自身的实例！

! 为避免保护或检测信号错误，必须先通过相应的序列“关闭”节点。

如果主站在停止后重新启动，所有其他连接的节点亦需要重新初始化。

若没有 CANOPEN_ENABLE，CANopen 主站可自动启动，但前提是已在配置中选择。

配置的波特率仅在 **CAN_ENABLE** (→ 页 [78](#)) 之前未启用的情况下予以采用。

输入端参数

7788

参数	数据类型	说明
ENABLE	BOOL := TRUE	TRUE: • 启用针对所选通道的 CANopen • 根据配置设定启动 CANopen 管理器或 CANopen 装置 FALSE: • 禁用针对所选通道的 CANopen • 终止 CANopen 管理器或 CANopen 装置
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
BaudRate (参数可选)	WORD := 0	波特率 [kbits/s] 允许值 = 20, 50, 100, 125, 250, 500, 800, 1 000 0 = 使用 PLC 配置的设置

输出端参数

7789

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	功能块执行完成且无错误
14	0E	FB 已启用 CANopen 管理器配置装置并发送 SDO
15	0F	FB 已启用 CANopen 管理器启动
238	EE	错误：CANopen 配置太大，无法启动
239	EF	错误：CANopen 管理器无法启动
242	F2	错误：不可设定

CANOPEN_GETBUFFERFLAGS

7890

= 获取缓冲标志
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7892

CANOPEN_GETBUFFERFLAGS 提供关于缓冲标志的信息。
标志可通过可选输入端复位。
功能块返回溢出标志的状态。

输入端参数

7893

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ResetRXFlags (参数可选)	BOOL := FALSE	TRUE: 提供输出端标志状态, 然后复位 FALSE: 功能元件未执行
ResetTXFlags (参数可选)	BOOL := FALSE	TRUE: 提供输出端标志状态, 然后复位 FALSE: 功能元件未执行

输出端参数

7894

参数	数据类型	说明
RXOVFL	BOOL	RX 溢出标志状况 TRUE: 接收缓冲区溢出 FALSE: 接收缓冲区无溢出
RXWARN	BOOL	RX 溢出警告标志状况 TRUE: 接收缓冲区程度处于临界状态 FALSE: 接收缓冲区程度处于非临界状态
TXOVFL	BOOL	TX 溢出标志状况 TRUE: 传送缓冲区溢出 FALSE: 传送缓冲区无溢出
TXWARN	BOOL	TX 溢出警告标志状况 TRUE: 传送缓冲区程度处于临界状态 FALSE: 传送缓冲区程度处于非临界状态
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	功能块执行完成且无错误
8	08	功能块还未执行
242	F2	错误：不可设定

CANOPEN_GETSTATE

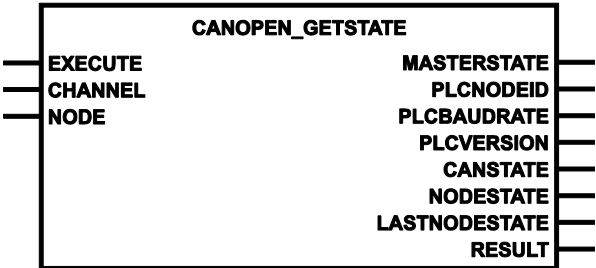
7865

= 获取状态

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7867

可通过 CANOPEN_GETSTATE 设定主站、从属装置或网络中特定节点参数。

输入端参数

7868

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
NODE	BYTE	Node ID = 节点的 ID (0...127) 装置作为 CANopen 主站： 值 = 0： 输出端仅可返回装置本身的状态信息。 包含节点信息的输出端无效。 值不为 0： 网络节点的 Node ID。 就此节点以及装置的节点而言，输出端返回状态。 装置作为 CANopen 从站： 值 = 0 (预设)： 输出端返回从站的状态信息。 值不为 0： 无操作

输出端参数

7869

参数	数据类型	说明
MASTERSTATE	BYTE	主站状态 = 主站内部状态： 0 = 0x00 = 主站启动 4 = 0x04 = 节点配置运行 5 = 0x05 = 主站正常运行状态 255 = 0xFF = PLC 作为从站运行
PLCNODEID	BYTE	PLC Node ID = 程序运行的 PLC 的 Node ID 值 = 0...127 = 0x00...0x7F
PLCBAUDRATE	DWORD	PLC 波特率
PLCVERSION	DWORD	PLC 版本
CANSTATE	BYTE	CANopen 网络的状态 装置作为主站运行： Node ID = 0 (装置亦同) 0 = 0x00 = OK 128 = 0x80 = BUSOFF Node ID ≠ 0 (节点)： 0 = 0x00 = OK 1 = 0x01 = 节点上的保护或检测信号错误 128 = 0x80 = BUSOFF 装置作为从站运行： 0 = 0x00 = OK 1 = 0x01 = 保护或检测信号错误 128 = 0x80 = BUSOFF
NODESTATE	BYTE	节点状态 = 从主站视角所见从站内部节点状态。输入端 NODEID 识别节点。 -1 = 0xFF = ResetNode 之后复位 1 = 0x01 = 等待 BOOTUP 2 = 0x02 = 接收 BOOTUP 消息之后 3 = 0x03 = 还未配置：STOPPED 4 = 0x04 = 通过 SDO 配置之后：PRE-OPERATIONAL 5 = 0x05 = 启动节点之后：OPERATIONAL 97 = 0x61 = 可选节点 98 = 0x62 = 除在 0x1000 中配置之外的其他装置类型 99 = 0x63 = 节点保护
LASTNODESTATE	BYTE	最后节点状态 基于 CANopen 的节点状态（通过这些值，状态还可在与节点相对应的消息中编码）。 0 0x00 BOOTUP 4 0x04 STOPPED 5 0x05 OPERATIONAL 127 0x7F PRE-OPERATIONAL

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：

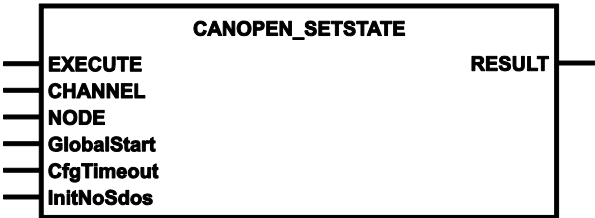
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
8	08	FB 已启用 – 还未处理
242	F2	错误：不可设定

CANOPEN_SETSTATE

7858

= 设定状态
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7860

可通过 CANOPEN_SETSTATE 设定主站、从属装置或网络中节点的参数。
主站、节点或装置的 NMT 状态在 CAN 堆栈中进行处理，或通过 FB 的命令进行处理
CANOPEN_NMTSERVICES (→ 页 [128](#)). 同时，还进行容许性检查。因为一致性的原因，未针对该目的提供输入端。

输入端参数

7861

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
NODE	BYTE	Node ID = 节点的 ID (0...127) 装置作为 CANopen 主站： 值 = 0： 更改仅指装置本身。 值不为 0： 网络节点 Node ID，其参数待更改。 既定的设定仅适用于该节点（而不是装置） 装置作为 CANopen 从站： 在从站模式中，从站的 Node ID 可通过该输入端设定。 值 = 0： 无操作 值不为 0： 功能块将该值作为装置的新 Node ID。

参数	数据类型	说明
GlobalStart (参数可选)	BOOL := TRUE	要求：启动 IEC 程序后必须立即调用 FB。该设定覆盖配置的设定。 TRUE: 同时启动所有参与者 FALSE: 逐个启动所有参与者
CfgTimeout (参数可选)	TIME := T#0ms	设定节点的配置超时： 值 = 0： 无操作 - 保留配置数据 值不为 0： 利用新值覆盖配置中的数据
InitNoSdos (参数可选)	BOOL := FALSE	对于 NODE 所示节点，初始化期间，... TRUE: 切勿发送配置数据 FALSE: 发送配置的 SDO

输出端参数

7862

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
8 08	FB 已启用 - 还未处理
242 F2	错误：不可设定

功能元件：CANopen 网络管理

内容	
CANOPEN_GETNMTSTATESLAVE	126
CANOPEN_NMTSERVICES.....	128

15063

CANOPEN_GETNMTSTATESLAVE

7851

= 获取网络管理状态从站

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7853

► 仅在装置作为 CANopen 从站运行时使用 FB !

如果请求了无效的状态转换，仅可通过 CANOPEN_GETNMTSTATESLAVE 将基于 CANopen 的运行状态和错误消息报告给应用程序。

输入端参数

7854

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定

输出端参数

7855

参数	数据类型	说明
NMTSTATE	BYTE	节点的网络运行状态 0 = INIT 1 = OPERATIONAL 2 = PRE-OPERATIONAL 3 = 已停止
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	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
242	F2	错误：不可设定

CANOPEN_NMTSERVICES

7843

= 网络管理服务

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7844

CANOPEN_NMTSERVICES 触发 NMT 命令或节点的初始化，具体视 NMT 命令条目而定。

 NMT = **网络管理**

功能块更新内部节点状态。如果 CANopen 状态转换（见统手册“ecomatmobile 技巧”> **NMT 状态**）未获允许，则不执行命令。

CANopen 装置可通过 FB 自动更改 CANopen 状态：
preoperational ⇌ operational

输入端参数

7847

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
NODE	BYTE	CANopen 节点 ID 允许值 = 0...127 = 0x00...0x7F NODE = 0: 命令适用于网络中的所有节点 NODE = 装置节点 ID : 命令同样适用于装置
NMTSERVICE	BYTE	网络命令 0 = 初始化节点 (主站除外) 1 = 进入 PRE-OPERATIONAL 2 = 启动节点 3 = 复位节点 4 = 复位通信 5 = 停止节点
Timeout (参数可选)	TIME := T#0ms	针对初始化的 FB 等待时间 时间结束后 FB 停止等待。 0 = 使用配置中的值

输出端参数

7848

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	功能块执行完成且无错误
8	08	功能块启用中
35	23	至少 1 个配置 SDO 未成功
36	24	节点已初始化
37	25	请求初始化时，节点不处于 PRE-OPERATIONAL 模式
43	2B	主站/从站未初始化
241	F1	错误：CANopen 状态转换未获准许
242	F2	错误：不可设定

功能元件：CANopen 对象目录

内容	
CANOPEN_GETODCHANGEDFLAG	132
CANOPEN_READOBJECTDICT	134
CANOPEN_WRITEOBJECTDICT	136

15065

CANOPEN_GETODCHANGEDFLAG

7927

= 获取对象目录更改标志

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7928

CANOPEN_GETODCHANGEDFLAG 报告特定对象目录条目的任何值更改。

输入端参数

7930

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引

输出端参数

7931

参数	数据类型	说明
DATA	DWORD	参数值
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	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
242	F2	错误：不可设定

CANOPEN_READOBJECTDICT

7933

= 读取对象目录
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7935

CANOPEN_READOBJECTDICT 从装置对象目录读取高达 4 字节的配置数据以用于应用程序。

输入端参数

7936

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引

输出端参数

7937

参数	数据类型	说明
DATA	DWORD	参数值
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	功能块执行完成且无错误
8	08	功能块还未执行
40	28	对象目录条目无效
242	F2	错误：不可设定

CANOPEN_WRITEOBJECTDICT

7940

= 写入对象目录

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7942

CANOPEN_WRITEOBJECTDICT 将配置数据写入控制器的对象目录。

通知

这样可能导致重要系统设定篡改，例如：

- 保护时间
- 检测信号时间

► 谨慎核实输入参数！

输入端参数

7943

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引
DATA	DWORD	参数值

输出端参数

7945

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	功能块执行完成且无错误
8	08	功能块还未执行
40	28	对象目录条目无效
242	F2	错误：不可设定

功能元件： CANopen SDO

内容	
CANOPEN_SDOREAD	139
CANOPEN_SDOREADBLOCK.....	141
CANOPEN_SDOREADMULTI	144
CANOPEN_SDOWRITE	146
CANOPEN_SDOWRITEBLOCK	148
CANOPEN_SDOWITEMULTI	151

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。

CANOPEN_SDOREAD

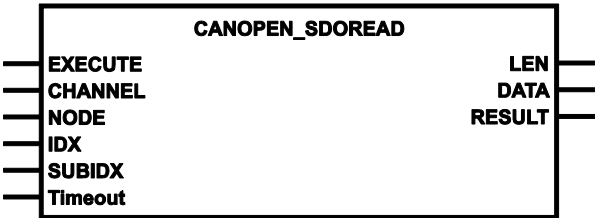
7791

= SDO 读取

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7793

CANOPEN_SDOREAD 是编辑 "Expedited SDO" 的简单功能块，即最多包含 4 字节用户数据的 SDO。该类型通常代表 SDO 通信的更大部分。

 Expedited SDO = 加速服务数据对象

因为数据量限制为最多 4 字节用户数据，可节省大量内存空间，因为该 FB 仅需保留 4 作为缓冲存储，且本身不会创建大数据组。

输入端参数

7794

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
NODE	BYTE	节点 ID 允许值 = 1...127 = 0x01...0x7F
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引
Timeout (参数可选)	TIME := T#10ms	针对响应的 FB 等待时间 时间结束后 FB 停止等待。 值 = 0： 使用配置中的值

输出端参数

7795

参数	数据类型	说明
LEN	BYTE	接收字节数量 (1...4)
DATA	DWORD	接收数据值 (高达 4 字节)
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
5	05	FB 启用中 – 还未接收到数据
32	20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
242	F2	错误：不可设定
255	FF	缓冲区溢出 – 接收了太多数据字节

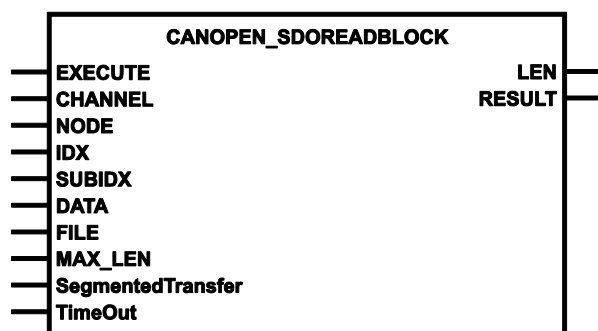
CANOPEN_SDOREADBLOCK

14942

= SDO 读取功能块

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:**说明**

14943

CANOPEN_SDOREADBLOCK 通过 SDO 块传输读取网络节点对象目录中所示的条目。

- > 如果阶段不支持块传输，则 FB 自动切换至“分段传输”。您还可通过输入端直接切换至“分段传输”。
- > 针对 SDO 的 COB ID 利用已传送节点 ID 计算。

多帧 SDO 的长度一般无限制。

就不包含文件系统的系统（例如 BasicController CR04nn）而言，以下几点适用：

- ▶ 将地址传送至 FB（通过指针访问以便写入）。由起始地址 DATA 和数据量 MAX_LEN 确定的内存区域必须可用！
- > 如果数据量大于所示，则通过 RESULT 停止传输并发送信号。

就包含文件系统的系统（例如 PDM360NG CR108n）而言，以下几点适用：

- ▶ 将文件路径和名称传送至 FB（数据按二进制格式保存于其中）。
- > 输出端 RESULT 提供关于 SDP 传送状态的信息。

输入端参数

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
NODE	BYTE	(Node ID) 节点的 ID 允许 = 1...127 = 0x01...0x7F  SDO 的 COB ID 利用节点 ID + 0x600 计算
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引
DATA	DWORD	存储已接收数据的数据区的地址  输入端没有针对包含文件系统的装置 (Linux) 的功能。
FILE	STRING(80)	存储二进制接收数据的路径和文件名称  输入端没有针对不含文件系统的装置 (BasicSystem) 的功能。
MAX_LEN	DWORD	可接收的最大允许字节数量
SegmentedTransfer (参数可选)	BOOL := FALSE	TRUE: 分段 SDO 传输 FALSE: SDO 块
Timeout (参数可选)	TIME := T#10ms	针对响应的 FB 等待时间 时间结束后 FB 停止等待。 值 = 0: 使用配置中的值

输出端参数

14951

参数	数据类型	说明
LEN	DWORD	已接收数据字节数量
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
5	05	FB 启用中 – 还未接收到数据
16	10	传输作为分段下载启用
17	11	传输作为块下载启用
32	20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
64	40	错误：写入指针在允许的数据范围之外
65	41	错误：文件无法打开
66	42	写入至文件时出错
242	F2	错误：不可设定

CANOPEN_SDOREADMULTI

7806

= SDO 读取 multi
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7808

CANOPEN_SDOREADMULTI 读取网络节点对象目录中所示的条目。针对 SDO 的 COB ID 根据 CANopen 约定，利用已传送节点 ID 计算。

输入端参数

7809

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
NODE	BYTE	(Node ID) 节点的 ID 允许 = 1...127 = 0x01...0x7F ① SDO 的 COB ID 利用节点 ID + 0x600 计算
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引
Timeout (参数可选)	TIME := T#10ms	针对响应的 FB 等待时间 时间结束后 FB 停止等待。 值 = 0： 使用配置中的值

输出端参数

7810

参数	数据类型	说明
LEN	DWORD	已接收字节数量 允许值 = 1...2 048 = 0x0000 0001...0x0000 0800
DATA	ARRAY [0..SDOMAXDATA] OF BYTE	SDO 数据传输用户数据缓冲区内存
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
5	05	FB 启用中 – 还未接收到数据
32	20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
242	F2	错误：不可设定
255	FF	错误：内存不足以使用多帧

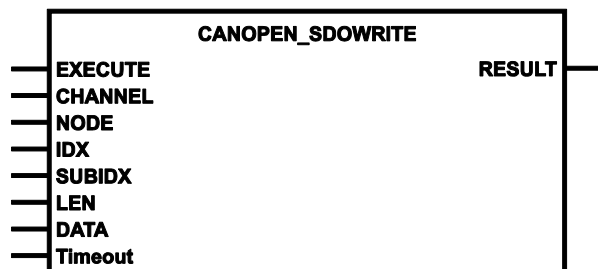
CANOPEN_SDOWRITE

7825

= SDO 写入

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:**说明**

7826

CANOPEN_SDOWRITE 是编辑 "Expedited SDO" 的简单功能块，即最多包含 4 字节用户数据的 SDO。该类型通常代表 SDO 通信的更大部分。

 Expedited SDO = 加速服务数据对象

因为数据量限制为最多 4 字节用户数据，可节省大量内存空间，因为该 FB 仅需保留 4 作为缓冲存储，且本身不会创建大数据组。

输入端参数

7828

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
NODE	BYTE	节点 ID 允许值 = 1...127 = 0x01...0x7F
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引
LEN	BYTE	待传送数据字节数量 允许值 = 1...4 = 0x01...0x04
DATA	ARRAY [0..3] OF BYTE	数据区 (1...4 字节)
Timeout (参数可选)	TIME := T#10ms	针对响应的 FB 等待时间 时间结束后 FB 停止等待。 值 = 0： 使用配置中的值

输出端参数

7829

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
8 08	功能块启用中
32 20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33 21	TIMEOUT 已过
242 F2	错误： 不可设定

CANOPEN_SDOWRITEBLOCK

14961

= SDO 写入功能块

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:**说明**

14963

CANOPEN_SDOWRITEBLOCK 通过 SDO 块传输写入网络节点对象目录中所示的条目。

如有必要，您可通过 FB 输入端切换至分段传输。

- > 针对 SDO 的 COB ID 利用已传送节点 ID 计算。
- > 输出端 RESULT 提供关于 SDP 传送状态的信息。

多帧 SDO 的长度一般无限制。

就不包含文件系统的系统（例如 BasicController CR04nn）而言，以下几点适用：

- ▶ 将地址传送至 FB（通过指针访问以便读取）。

就包含文件系统的系统（例如 PDM360NG CR108n）而言，以下几点适用：

- ▶ 将文件路径和名称传送至 FB（数据按二进制格式从中读取）。

输入端参数

14964

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
NODE	BYTE	(Node ID) 节点的 ID 允许 = 1...127 = 0x01...0x7F  SDO 的 COB ID 利用节点 ID + 0x600 计算
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引
LEN	DWORD	DATA 待传送数据字节数量 允许 = 1...2 048 = 0x0000 0001...0x0000 0800
DATA	DWORD	读取待传送数据的数据区的地址  输入端没有针对包含文件系统的装置 (Linux) 的功能。
FILE	STRING(80)	读取二进制传送数据的路径和文件名称  输入端没有针对不含文件系统的装置 (BasicSystem) 的功能。
SegmentedTransfer (参数可选)	BOOL := FALSE	TRUE: 分段 SDO 传输 FALSE: SDO 块
Timeout (参数可选)	TIME := T#10ms	针对响应的 FB 等待时间 时间结束后 FB 停止等待。 值 = 0 : 使用配置中的值

Parameters of the outputs

14968

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：

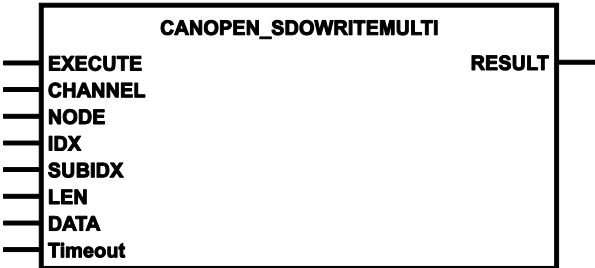
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
8	08	FB 已启用 – 还未处理
16	10	传输作为分段下载启用
17	11	传输作为块下载启用
32	20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
65	41	错误：文件无法打开
242	F2	错误：不可设定

CANOPEN_SDOWRITEMULTI

7832

= SDO 写入 multi
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7834

CANOPEN_SDOWRITEMULTI 写入网络节点对象目录中所示的条目。 针对 SDO 的 COB ID 根据 CANopen 约定，利用已传送节点 ID 计算。

输入端参数

7835

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
NODE	BYTE	节点 ID 允许值 = 1...127 = 0x01...0x7F
IDX	WORD	对象字典的索引 y
SUBIDX	BYTE	子索引参考对象目录中的索引
LEN	DWORD	待传送数据字节数量 允许值 = 1...2 048 = 0x0000 0001...0x0000 0800
DATA	ARRAY [0..SDOMAXDATA] OF BYTE	SDO 数据传输用户数据缓冲区内存
Timeout (参数可选)	TIME := T#10ms	针对响应的 FB 等待时间 时间结束后 FB 停止等待。 值 = 0： 使用配置中的值

输出端参数

7836

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
8	08	功能块启用中
32	20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
242	F2	错误：不可设定

功能元件： CANopen SYNC

内容	
CANOPEN_GETSYNCSTATE.....	155
CANOPEN_SETSYNCSTATE.....	157

15069

CANOPEN_GETSYNCSTATE

7871

= 获取 SYNC 状态
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7872

CANOPEN_GETSYNCSTATE reads...

- SYNC 功能的设定（启用/未启用），
- SYNC 功能的错误状态 (SyncError)。

如果 PLC CAN 作为 CANopen 从站运行，则通过该 FB 以信号通知 SYNC 信号不存在还是定期显示。

同步 PDOS 等在 CAN 堆栈中处理。但 CANOPEN_GETSYNCSTATE 提供错误状态，以便应用程序能够相应地作出响应。

输入端参数

7874

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定

输出端参数

7875

参数	数据类型	说明
SYNC	BOOL	SYNC 功能的状态 TRUE: SYNC 已启用： 在 主站模式 中，SYNC 电报根据配置中的设定生成，同步 PDO 传送和接收。 在 从站模式 中，SYNC 电报接收并相应地处理。 FALSE: SYNC 未启用
SYNCERROR	BYTE	(同步错误) SYNC 错误消息 0 = 无错误 >0 = SYNC 错误 (从站模式)
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	功能块执行完成且无错误
8	08	功能块还未执行
242	F2	错误：不可设定

CANOPEN_SETSYNCSTATE

7883

= 设定 SYNC 状态

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7884

SYNC 通过 CANOPEN_SETSYNCSTATE 开启和关闭。

输入端参数

7886

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
SYNC	BOOL	SYNC 功能的状态 TRUE: SYNC 已启用： 在 主站模式 中，SYNC 电报根据配置中的设定生成，同步 PDO 传送和接收。 在 从站模式 中，SYNC 电报接收并相应地处理。 FALSE: SYNC 未启用

输出端参数

7887

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	功能块执行完成且无错误
8	08	功能块还未执行
38	26	SYNC 不可启用
242	F2	错误：不可设定

功能元件：CANopen 保护

内容	
CANOPEN_GETGUARDHBERRLIST	160
CANOPEN_GETGUARDHBSTATSLV	162

15071

CANOPEN_GETGUARDHBERRLIST

7896

= 获取保护和检测信号错误列表

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7898

CANOPEN_GETGUARDHBERRLIST 批量列出主站检测到错误的所有节点：

- 保护错误
- 检测信号错误

输入端参数

7899

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
ResetList (参数可选)	BOOL := FALSE	复位错误列表 TRUE: 提供错误列表以及输出端故障 节点数量，然后复位。 FALSE: 功能元件未执行

输出端参数

7900

参数	数据类型	说明
N_NODES	WORD	包含检测信号或保护错误的节点的数量 0 = 无节点有防护或检测信号错误
NODEID	ARRAY [0..MAXGUARDERROR] OF BYTE	包含检测信号或保护错误的 Node ID 列表 最近的条目位于索引 0。 MAXGUARDERROR 取决于装置 → 章节 本装置 CANopen 的限制 (→ 页 45)
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	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
242	F2	错误：不可设定

CANOPEN_GETGUARDHBSLVLV

7902

= 获取保护和检测信号状态从站

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7904

CANOPEN_GETGUARDANDHBSTATESLAVE 将以下状态报告至从属运行的控制器：

- 监控节点保护
- 监控检测信号

控制器可能是检测信号生成器或检测信号消耗器。

输入端参数

7905

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
Reset (参数可选)	BOOL := FALSE	TRUE: 提供输出端当前状态 然后复位为“无错误” FALSE: 功能元件未执行

输出端参数

7906

参数	数据类型	说明
GUARDSTATE	BYTE	节点保护状态 0 = 0x00 = 无错误（或：未启用） 1 = 0x01 = 超时（配置） 127 = 0x7F = 未接收到保护消息
PROD_HBSTATE	BYTE	控制器作为检测信号生成器： 0 = 0x00 = 未启用 1 = 0x01 = 启用
CONS_HBSTATE	BYTE	控制器作为检测信号消耗器： 0 = 0x00 = 无故障 1 = 0x01 = 超时（配置） 127 = 0x7F = 还未接收到检测信号消息
CONS_HBCOBID	WORD	控制器消耗器检测信号对其作出响应的检测信号消息的 COB-ID（配置）
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	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
242	F2	错误：不可设定

功能元件：CANopen 紧急

内容	
CANOPEN_GETEMCYMESSAGES.....	165
CANOPEN_GETERRORREGISTER.....	167
CANOPEN_SENDEMCMYMESSAGE	169

15073

CANOPEN_GETEMCYMESSAGES

7921

= 获取紧急消息
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7923

CANOPEN_GETEMCYMESSAGES 返回自上次删除消息以来控制器从网络其他节点接收的所有紧急消息。
该列表可通过设定相应的输入端复位。 MAXEMCYMSGs 消息最大值存储。 每条消息包含关于从哪个节点发送的信息。 最近的消息位于索引 0。

输入端参数

7924

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
RstList (参数可选)	BOOL := FALSE	TRUE: 提供包含输出端累积 CAN 消息 的列表, 然后删除。 FALSE: 功能元件未执行

输出端参数

参数	数据类型	说明								
N_MSGS	DWORD	累积消息的数量								
EMCY	ARRAY [0..MAXEMCYMSGS] OF T_EM CY	紧急消息 最近的条目位于索引 0。 T_EM CY 结构： <table><tr><td>.NODEID</td><td>消息来源节点的 ID</td></tr><tr><td>.EEC</td><td>紧急错误代码</td></tr><tr><td>.ER</td><td>错误寄存器</td></tr><tr><td>.MSEF</td><td>制造商特定错误代码</td></tr></table> MAXEMCYMSG = 10	.NODEID	消息来源节点的 ID	.EEC	紧急错误代码	.ER	错误寄存器	.MSEF	制造商特定错误代码
.NODEID	消息来源节点的 ID									
.EEC	紧急错误代码									
.ER	错误寄存器									
.MSEF	制造商特定错误代码									
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	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
242	F2	错误：不可设定

CANOPEN_GETERRORREGISTER

7915

= 获取错误寄存器
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7917

CANOPEN_GETERRORREGISTER 从控制器读取错误寄存器 0x1001 和 0x1003。

输入端参数

7918

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
Reset_1001 (参数可选)	BOOL := FALSE	TRUE: 复位错误寄存器 0x1001 FALSE: 功能元件未执行
Reset_1003 (参数可选)	BOOL := FALSE	TRUE: 复位错误寄存器 0x1003 将条目数量设为 0 FALSE: 功能元件未执行 输入端保持不变。

输出端参数

7919

参数	数据类型	说明
ER	BYTE	错误寄存器 0x1001 的内容
ERROR_FIELD	ARRAY [0..MAXERR] OF DWORD	错误寄存器 0x1003 的内容 索引 0 = 已存储错误的数量 索引 1..MAXERR = 已存储错误 最近的错误位于索引 1。 预设：MAXERR = 5
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	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
242	F2	错误：不可设定

CANOPEN_SENDEMCMYMESSAGE

7908

= 发送紧急消息
单元类型 = 功能块 (FB)
元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7910

CANOPEN_SENDEMCMYMESSAGE 发送 EMCY 消息。消息通过相应的参数集合而成并输入寄存器 0x1003。紧急消息的 COB ID 通过配置数据确定。

输入端参数

7911

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ERRORACTIVE	BOOL	FALSE ⇔ TRUE (edge): 发送下一个错误代码 TRUE ⇔ FALSE (边沿) : 若不再提供错误, 则在延迟 1 秒后发送 “无错误”的消息。
EEC	WORD	EEC = Emergency Error Code
ER (参数可选)	BYTE := 0	0 = 使用错误寄存器 0x1001 的值
MSEF	ARRAY [0..4] OF BYTE	MSEF = 制造商特定错误代码 = 由制造商定义的其他错误代码。 值来自应用程序。
Write1003 (参数可选)	BOOL := FALSE	TRUE: 在对象 0x1003 中输入该 EMCY 消息 FALSE: 功能元件未执行
SendSysStatus (参数可选)	BOOL := FALSE	发送系统状态 TRUE: 系统状态会予以检查, 且若出现 错误状态, 会传送至网络。 FALSE: 功能元件未执行

输出端参数

7912

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	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
39	27	配置中无对象 1001 ₁₆
242	F2	错误：不可设定

5.2.4 功能元件：SAE J1939

内容

功能元件：SAE J1939 状态	172
功能元件：SAE J1939 请求	184
功能元件：接收 SAE J1939	189
功能元件：传送 SAE J1939	196
功能元件：SAE J1939 诊断	206

2273

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

功能元件：SAE J1939 状态

内容

J1939_ENABLE.....	173
J1939_GETDABYNAME	175
J1939_NAME	178
J1939_STATUS.....	182

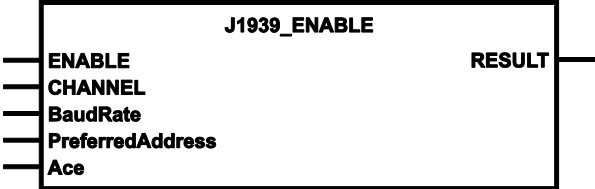
15077

J1939_ENABLE

7641

单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7642

就 J1939 堆栈初始化而言，J1939_ENABLE 设为 TRUE=1。
> 该 FB 也会促使启动 CFG 文件软输入/输出。
> 如果 CAN_ENABLE 之前未启用，则仅可采用不同的波特率。

ACE = 地址声明启用：

- 如果 IFM 控制器通过 J1939 与唯一发动机控制器通信：
设定 ACE = FALSE。
- 但如果同一个总线上有几个发动机控制器在工作：
设定 ACE = TRUE。
在此情况下，发动机控制器必须支持地址声明！
否则可能与后续系统故障存在地址重叠的风险。

输入端参数

7643

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: 启用 J1939 通道 Ace=TRUE: 地址声明生效 FALSE: 阻止 J1939 通道
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
Baudrate (参数可选)	WORD := 250	波特率 [Kbits/s] 允许值： 20, 50, 100, 125, 250, 500, 800, 1 000
PreferredAddress (参数可选)	BYTE = 252	首选源地址
Ace (参数可选)	BOOL := TRUE	地址声明启用 TRUE: 地址声明启用 (控制设备自动配置) FALSE: 无地址声明

输出端参数

8542

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：

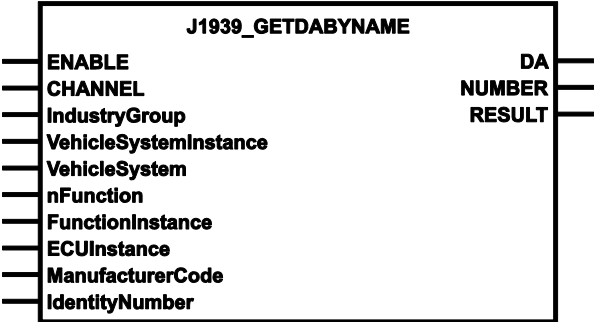
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	功能块启用中
9	09	CAN 未启用
242	F2	错误：不可设定

J1939_GETDABYNAME

7664

= 获取目标任意名称
单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7665

通过 J1939_GETDABYNAME，可利用名称信息确定一个或多个参与者的目标地址。

- 如果在可选输入端未设定特定值：
⇒ 结果列表仅显示包含该特定值的参与者。
- 如果在可选输入端未设定值或默认值：
⇒ 列表过滤期间不考虑该条目。

输入端参数

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
IndustryGroup (参数可选)	BYTE = 0xFF	industry group = 装置的行业组 允许值 = 0...7 255 = 0xFF = 针对所有的过滤
VehicleSystemInstance (参数可选)	BYTE := 0xFF	车辆系统实例 允许值 = 0...15 = 0x00...0x0F 255 = 0xFF = 针对所有的过滤
VehicleSystem (参数可选)	BYTE := 0xFF	车辆系统 允许值 = 0...127 = 0x00...0x7F 255 = 0xFF = 针对所有的过滤
nFunction (参数可选)	WORD := 0xFFFF	装置的功能 允许值 = 0...255 = 0x0000...0x00FF 65 535 = 0xFFFF = 针对所有项的过滤
FunctionInstance (参数可选)	BYTE := 0xFF	功能的实例 允许值 = 0...31 = 0x00...0x1F 255 = 0xFF = 针对所有的过滤
ECUInstance (参数可选)	BYTE := 0xFF	控制装置的实例 允许值 = 0...7 255 = 0xFF = 针对所有的过滤
ManufacturerCode (参数可选)	WORD := 0xFFFF	制造商代码 (必须从 SAE 申请) 允许值 = 0...2047 ($2^{11}-1$) = 0x0000...0x07FF 65 535 = 0xFFFF = 针对所有项的过滤
IdentityNumber (参数可选)	DWORD := 0xFFFF FFFF	装置的序列号 (不得覆盖) 允许值 = 0...2047 ($2^{11}-1$) = 0x0000 0000...0x0000 07FF 4 294 967 295 = 0xFFFF FFFF = 针对所有项的过滤

输出端参数

7668

参数	数据类型	说明
DA	ARRAY [0..254] OF BYTE	已找到参与者列表 255 = 未通过该数字找到参与者
NUMBER	BYTE	已找到总线参与者的数量。
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

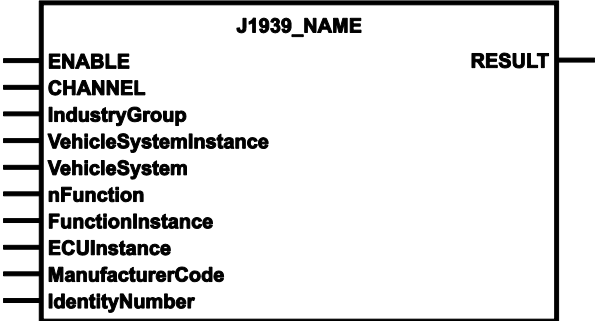
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
8	08	功能块启用中
242	F2	错误：不可设定

J1939_NAME

7646

单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7648

可通过 J1939_NAME 向装置提供名称以在网络中识别。

默认情况下，使用 **IFM** 名称。

用户有以下选项可更改装置的名称：

- ▶ 使用 CFG 文件的信息或
- ▶ 通过 J1939_NAME 覆盖所需数据。
- > 如果在可选输入端未设定值或默认值：
 - ⇒ 未覆盖预设值。

>

下表显示符合 SAE J1939-81 的 64 位 NAME 信息的构成：

参数	数据类型	说明
arbitrary address capable	1 位	提供任何需要的地址
industry group	3 位	装置的行业组
vehicle system instance	4 位	车辆系统实例
vehicle system	7 位	车辆系统
reserved	1 位	已保留
function	8 位	装置的功能
function instance	5 位	功能的实例
ECU instance	3 位	控制器的实例
manufacturer code	11 位	制造商代码（必须适用于 SAE）
identify number	21 位	装置的序列号（不得覆盖）

表格：符合 SAE J1939-81 的 64 位 NAME 信息的构成

输入端参数

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: 提供任何需要的地址 FALSE: 固定地址
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
IndustryGroup (参数可选)	BYTE = 0xFF	industry group = 装置的行业组 允许值 = 0...7 255 = 0xFF = 针对所有的过滤
VehicleSystemInstance (参数可选)	BYTE := 0xFF	车辆系统实例 允许值 = 0...15 = 0x00...0x0F 255 = 0xFF = 针对所有的过滤
VehicleSystem (参数可选)	BYTE := 0xFF	车辆系统 允许值 = 0...127 = 0x00...0x7F 255 = 0xFF = 针对所有的过滤
nFunction (参数可选)	WORD := 0xFFFF	装置的功能 允许值 = 0...255 = 0x0000...0x00FF 65 535 = 0xFFFF = 针对所有项的过滤
FunctionInstance (参数可选)	BYTE := 0xFF	功能的实例 允许值 = 0...31 = 0x00...0x1F 255 = 0xFF = 针对所有的过滤
ECUInstance (参数可选)	BYTE := 0xFF	控制装置的实例 允许值 = 0...7 255 = 0xFF = 针对所有的过滤
ManufacturerCode (参数可选)	WORD := 0xFFFF	制造商代码 (必须从 SAE 申请) 允许值 = 0...2047 (2 ¹¹ -1) = 0x0000...0x07FF 65 535 = 0xFFFF = 针对所有项的过滤
IdentityNumber (参数可选)	DWORD := 0xFFFF FFFF	装置的序列号 (不得覆盖) 允许值 = 0...2047 (2 ¹¹ -1) = 0x0000 0000...0x0000 07FF 4 294 967 295 = 0xFFFF FFFF = 针对所有项的过滤

输出端参数

7661

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：

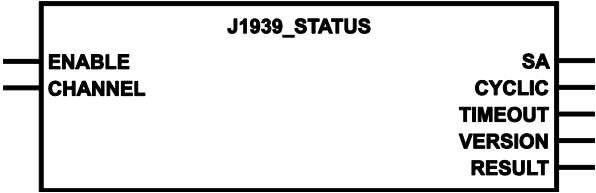
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	功能块启用中
242	F2	错误：不可设定

J1939_STATUS

7670

单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7672

通过 J1939_STATUS , 相关信息可读回至 J1939 堆栈。

输入端参数

7673

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
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定

输出端参数

7674

参数	数据类型	说明
SA	BYTE	已声明源地址
CYCLIC	WORD	循环消息的数量
TIMEOUT	BYTE	未及时针对处理图像提供数据的节点的源地址 255 = 0xFF = 所有节点及时发送
VERSION	DWORD	IFM CAN 堆栈库的 vabvab
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	协议已启用
2	02	协议未启用
3	03	源地址已请求
4	04	地址丢失
242	F2	错误：不可设定

功能元件： SAE J1939 请求

内容	
J1939_SPEC_REQ	185
J1939_SPEC_REQ_MULTI	187

15079

J1939_SPEC_REQ

15023

= J1939 特定请求
单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

15026

J1939_SPECIFIC_REQUEST 请求和接收来自其他控制器的特定消息。
如果请求多帧消息：
• FB 提供前 8 字节的数据
• RESULT 显示错误

输入端参数

15028

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
DA	BYTE	所需装置 J1939 地址

输出端参数

15029

参数	数据类型	说明
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	接收字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
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
5	05	FB 启用中 – 还未接收到数据
64	40	错误：接收多帧
242	F2	错误：不可设定

J1939_SPEC_REQ_MULTI

15033

= J1939 特定请求多帧消息

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

15036

J1939_SPECIFIC_REQUEST 请求和接收来自其他控制器的特定多帧消息。

输入端参数

15037

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
DA	BYTE	所需装置 J1939 地址

输出端参数

15038

参数	数据类型	说明
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	待传送数据字节数量 允许 = 1...1 785 = 0x0001...0x06F9
DATA	ARRAY [0..1784] OF BYTE	已接收数据 (1...1785 字节)
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
5	05	FB 启用中 – 还未接收到数据
242	F2	错误：不可设定

功能元件： 接收 SAE J1939

内容	
J1939_RX	190
J1939_RX_FIFO.....	192
J1939_RX_MULTI	194

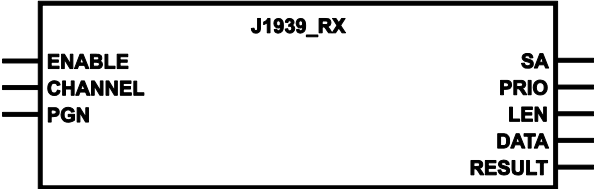
15081

J1939_RX

7724

单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7725

J1939_RX 是最简单的单帧消息接收方式。最后在 CAN 总线上读取的消息返回。

输入端参数

7726

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
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF

❗ PGN = 0 未使用。

输出端参数

7727

参数	数据类型	说明
SA	BYTE	变送器源地址
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	接收字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
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	功能块执行完成且无错误
5	05	FB 启用中 – 还未接收到数据
9	09	CAN 未启用
242	F2	错误：不可设定

J1939_RX_FIFO

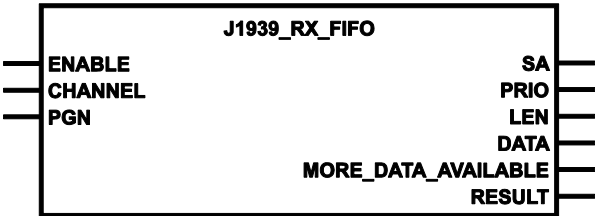
7732

= 包含 FIFO 的 J1939 RX

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7733

J1939_RX_FIFO 供接收所有特定消息并连续从 FIFO 读取。

输入端参数

7734

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
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF

! PGN = 0 未使用。

输出端参数

7735

参数	数据类型	说明
SA	BYTE	变送器源地址
PRI0	BYTE	消息优先级 (0...7)
LEN	BYTE	接收字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	已接收数据，(1...8 字节)
MORE_DATA_AVAILABLE	BOOL	TRUE: FiFo 中有进一步接收的数据 FALSE: FiFo 中无进一步接收的数据
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
5	05	FB 启用中 – 还未接收到数据
242	F2	错误：不可设定
250	FA	错误：FiFo 已满 – 数据丢失

J1939_RX_MULTI

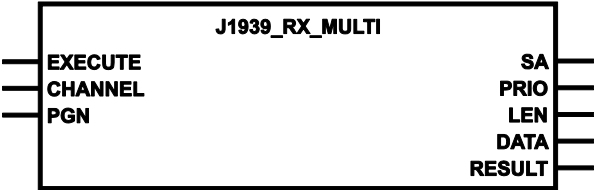
7736

= J1939 RX 多帧消息

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7741

J1939_RX_MULTI 供接收多帧消息。

输入端参数

7743

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF

! PGN = 0 未使用。

输出端参数

7744

参数	数据类型	说明
SA	BYTE	变送器源地址
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	已接收字节数量 允许值 = 0...1 785 = 0x0000 0000...0x0000 06F9
DATA	ARRAY [0..1784] OF BYTE	待传送数据 (1...1785 字节)
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
5	05	FB 启用中 – 还未接收到数据
242	F2	错误：不可设定

功能元件： 传送 SAE J1939

内容	
J1939_TX	197
J1939_TX_ENH.....	199
J1939_TX_ENH_CYCLIC	201
J1939_TX_ENH_MULTI.....	203

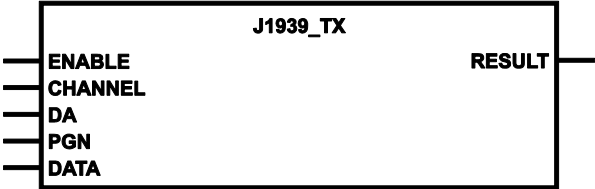
15083

J1939_TX

7688

单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7689

J1939_TX 是最简单的单帧消息传送方式。

输入端参数

7690

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
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
DA	BYTE := 249	DA = ECU 目标地址 PGN > 61139: 忽略参数 DA
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7693

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	功能块执行完成且无错误
242	F2	错误：不可设定
250	FA	错误：FIFO 已满 – 数据丢失

J1939_TX_ENH

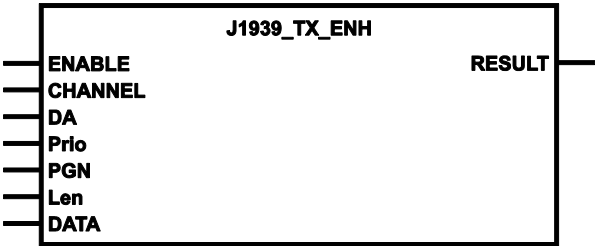
7696

= J1939 TX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7697

其他设定选项由针对单帧消息的 J1939_TX_ENH (针对：增强) 提供：

- 传送优先级
- 数据长度

多帧消息 → J1939_TX_ENH_MULTI (→ 页 [203](#)).

输入端参数

7702

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE := 249	DA = ECU 目标地址 PGN > 61139: 忽略参数 DA
Prio (参数可选)	BYTE := 3	消息优先级 允许值 = 0...7
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
Len (参数可选)	BYTE := 8	待传送字节数量 允许值 = 0...8
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7969

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：

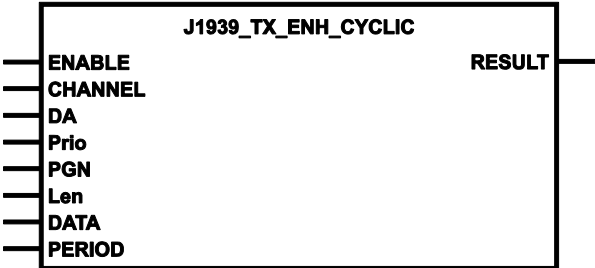
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
242	F2	错误：不可设定
250	FA	错误：FIFO 已满 – 数据丢失

J1939_TX_ENH_CYCLIC

7716

= J1939 TX 增强循环
单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7718

J1939_TX_ENH_CYCLIC 用于次循环传送 CAN 消息。
否则，FB 相当于 **J1939_TX_ENH** (→ 页 [199](#)).
► 通过参数 PERIOD 设定周期。

! 如果周期太短，则可能导致总线负载高！
总线负载可影响整个系统的性能。

输入端参数

7719

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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE := 249	DA = ECU 目标地址 PGN > 61139: 忽略参数 DA
Prio (参数可选)	BYTE := 3	消息优先级 允许值 = 0...7
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
Len (参数可选)	BYTE := 8	待传送字节数量 允许值 = 0...8
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)
PERIOD	TIME	周期

输出端参数

7720

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
8	08	功能块启用中
242	F2	错误：不可设定

J1939_TX_ENH_MULTI

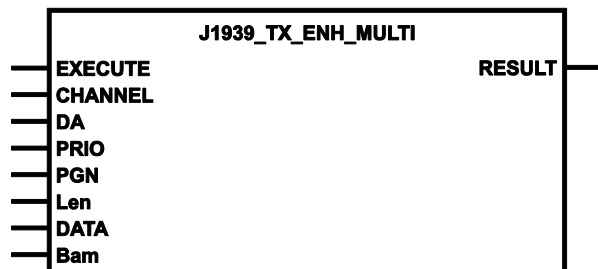
7699

= J1939 TX 增强多帧消息

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7705

多帧消息的传送通过 J1939_TX_ENH_MULTI 执行。

FB 相当于 **J1939_TX_ENH** (→ 页 [199](#)). 此外, 还可确定是否应作为 BAM (**广播公告消息**) 执行传送。

输入端参数

7712

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE := 249	DA = ECU 目标地址 PGN > 61139: 忽略参数 DA
Prio (参数可选)	BYTE := 3	消息优先级 允许值 = 0...7
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
Len (参数可选)	BYTE := 8	待传送字节数量 允许值 = 0...8
DATA	ARRAY [0..1784] OF BYTE	待传送数据 (1...1785 字节)
Bam (参数可选)	BOOL := FALSE	BAM = 广播公告消息 = 面向所有参与者的消息 TRUE: 作为 BAM 的多帧传送 面向所有参与者的消息 FALSE: 自动; 仅面向目标地址的消息

输出端参数

7714

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	功能块执行完成且无错误
8	08	功能块启用中
65	41	错误：不可传送
242	F2	错误：不可设定

功能元件： SAE J1939 诊断

内容	
J1939_DM1RX	207
J1939_DM1TX.....	210
J1939_DM1TX_CFG	213
J1939_DM3TX.....	215

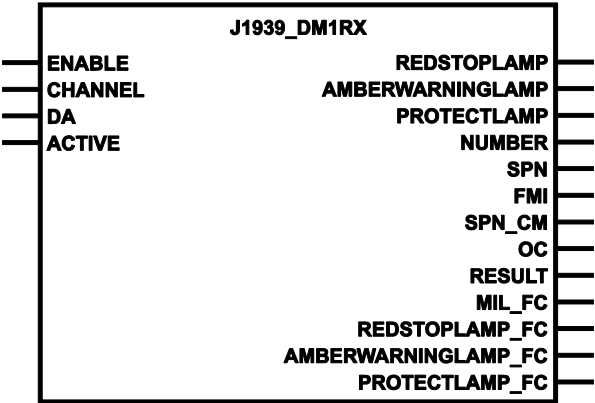
15085

J1939_DM1RX

14977

= J1939 诊断消息 1 RX
单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7761

J1939_RX_DM1 从其他 ECU 接收诊断消息 DM1 或 DM2。

输入端参数

14979

参数	数据类型	说明
ENABLE	BOOL := FALSE	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
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE	DA = 可检索 DTC 的 ECU 的目标地址。 DA = 254: 从装置本身读取 DTC
ACTIVE	BOOL	TRUE: 读取有效 DTC (DM1) FALSE: 读取之前有效的 DTC (DM2)

输出端参数

14980

参数	数据类型	说明
REDSTOPLAMP	BOOL	红色停止灯（仅限较早的项目） TRUE: 开启 FALSE: 熄灭
AMBERWARNINGLAMP	BOOL	琥珀色警告灯（仅限较早的项目） TRUE: 开启 FALSE: 熄灭
PROTECTLAMP	BOOL	保护灯（仅限较早的项目） TRUE: 开启 FALSE: 熄灭
NUMBER	BYTE	接收 DTC 数量 (0..8)
SPN	WORD	可疑参数编号（→ J1939 规格）
FMI	BYTE	故障模式指示器（→ J1939 规格） 允许值 = 0...31 = 0x00...0x1F
SPN_CM	BOOL	转换方式（→ J1939 规格）
OC	BYTE	发生次数计数
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）
MIL_FC	BYTE	电子组件状态 故障指示灯状态和闪烁代码： 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁
REDSTOPLAMP_FC	BYTE	电子组件状态 红色停止灯状态和闪烁代码： 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁
AMBERWARNINGLAMP_FC	BYTE	电子组件状态 黄色警告灯状态和闪烁代码： 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁

参数	数据类型	说明
PROTECTLAMP_FC	BYTE	电子组件状态 保护灯状态和闪烁代码 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁

RESULT 的可能结果：

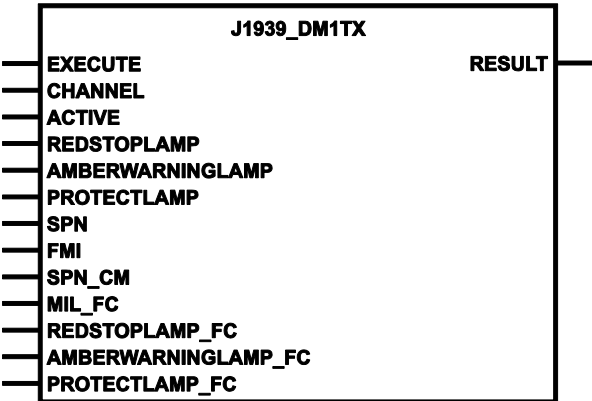
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
8	08	FB 启用中 – 未接收到数据
242	F2	错误：不可设定

J1939_DM1TX

14993

= J1939 诊断消息 1 TX
单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7747

控制器仅可通过 J1939_TX_DM1 (DM = 诊断消息) 将一条有效错误消息传送至 CAN 堆栈。

- > 该消息存储在硬件配置中。
 - > 消息标记为“有效”并每秒作为 DM1 传送一次。
 - > 如果已发生错误，事件计数器则会增加数值。
 ❗ 事件计数器由 CAN 堆栈管理。
 - > 故障代码的所有位执行分离。 只要在其中一个故障代码中设定一位，则同样在灯状态中设定。
- 在请求到达 DM2 后，CAN 堆栈可从硬件配置读取相应的信息并传送。
- > 当 DM3 消息到达时，硬件配置中的错误内存会删除所有无效错误。

输入端参数

14995

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ACTIVE	BOOL	TRUE: DTC 已启用 作为 DM1 循环传送 (1x 每秒) FALSE: DTC 不再启用 保存在硬件配置中 请求时作为 DM2 传送
REDSTOPLAMP	BOOL	红色停止灯 (仅限较早的项目) TRUE: 开启 FALSE: 熄灭
AMBERWARNINGLAMP	BOOL	琥珀色警告灯 (仅限较早的项目) TRUE: 开启 FALSE: 熄灭
PROTECTLAMP	BOOL	保护灯 (仅限较早的项目) TRUE: 开启 FALSE: 熄灭
SPN	WORD	可疑参数编号 (→ J1939 规格)
FMI	BYTE	故障模式指示器 (→ J1939 规格) 允许值 = 0...31 = 0x00...0x1F
SPN_CM	BOOL	转换方式 (→ J1939 规格)
MIL_FC	BYTE	电子组件状态 故障指示灯状态和闪烁代码： 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁
REDSTOPLAMP_FC	BYTE	电子组件状态 红色停止灯状态和闪烁代码： 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁

参数	数据类型	说明
AMBERWARNINGLAMP_FC	BYTE	电子组件状态 黄色警告灯状态和闪烁代码： 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁
PROTECTLAMP_FC	BYTE	电子组件状态 保护灯状态和闪烁代码 0 = 熄灭 1 = 开启 2 = 缓慢闪烁 3 = 快速闪烁

输出端参数

7750

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	数据在错误内存中标记为“有效”
242	F2	错误：不可设定

J1939_DM1TX_CFG

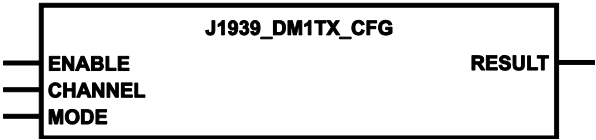
15424

= J1939 诊断消息 1 TX 可配置

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_V02.00.02.LIB 或更高

CODESYS 中的符号:



说明

15426

从运行时系统 V03.00.03 开始，只要针对相应的 CAN 接口调用 FB，**J1939_ENABLE** (→ 页 173) CAN 堆栈就每秒钟自动发送 DM1 消息。

► 如果不想 CAN 堆栈自动循环传送 DM1 消息，则使用 FB J1939_DM1TX_CFG。

FB 提供以下循环传送 DM1 消息的模式：

MODE = 0 (预设)	CAN 堆栈每秒发送符合标准的 DM1 "zero active faults" 消息。 可通过 FB 手动传送 DM1 J1939_DM1TX (→ 页 210)消息。
MODE = 1	CAN 堆栈不发送 DM1 "zero active faults" 消息。 DM2 请求自动响应。 可通过 FB 手动传送 DM1 J1939_DM1TX (→ 页 210) 消息。
MODE = 2	CAN 堆栈不发送循环 DM1 "zero active faults" 消息 CAN 堆栈也不会自动响应 DM2 请求。

输入端参数

15427

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
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
MODE	BYTE := 0	功能块的工作模式 允许 = 0...2 (→ FB 说明)

输出端参数

15429

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：

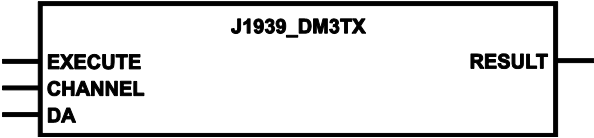
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
242	F2	错误：不可设定

J1939_DM3TX

15002

= J1939 诊断消息 3 TX
单元类型 = 功能块 (FB)
元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

15004

您可通过 J1939_DM3TX (DM = **诊断消**) 删除其他装置上的无效 DTC。
> 接收 DM3 消息后，即会在硬件配置中删除错误内存中的所有无效错误。

输入端参数

15006

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n) , 视装置而定
DA	BYTE	DA = 可删除 DTC 的 ECU 的 目标地址 。 DA = 254: 删除装置本身中的 DTC (DM2)

输出端参数

15008

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	功能块执行完成且无错误
242	F2	错误： 不可设定

5.2.5 功能元件：处理输入值

内容	
FASTCOUNT	217
INC_ENCODER	219
INPUT	223
PERIOD	226

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。

FASTCOUNT

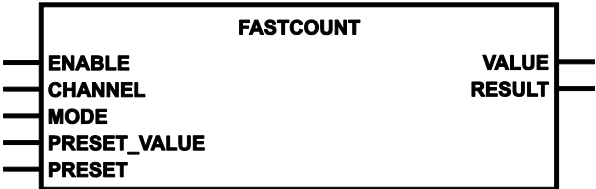
8112

= 快速计数

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyyzz.LIB

CODESYS 中的符号:



说明

8114

FASTCOUNT 作为快速输入脉冲（高达 30 kHz）的计数功能块运行。

该 FB 检测快速输入通道的脉冲（→ 数据表）。

❗ 未检测到计数值的过溢或下溢。

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

16729

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 开始的进程在后台继续 > FB 输出端未更新
CHANNEL	BYTE	快速输入通道编号 输入端 IN12...IN15 对应 12...15
	BYTE	功能块的工作模式： 0 = 0x00 = 停止计数器 21 = 0x15 = 上升沿计数器 22 = 0x16 = 下降沿计数器
PRESET_VALUE	DWORD	计数器初始值
PRESET	BOOL	TRUE (仅针对 1 个周期)： 加载初始值 PRESET_VALUE FALSE: 计数器已启用

输出端参数

8116

参数	数据类型	说明
VALUE	DWORD	输出值
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	功能块已启用 - 还没有可用的有效值
130	82	通道设定无效
132	84	模式设定无效

INC_ENCODER

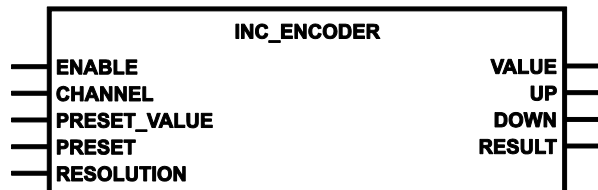
8134

= 增量编码器

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8135

INC_ENCODER 处理针对编码器评估的递增/递减计数器功能。

两个频率输入端构成输入端对且通过 FB 加以评估。

允许的极限频率 = 0...1 000 Hz

计数器可通过 PRESET_VALUE 设为预设值。如果 PRESET 设为 TRUE，则采用值。之后，必须再次将 PRESET 设为 FALSE，以便计时器再次启用。

电流计数值位于输出端 VALUE。输出端 UP 和 DOWN 表示计数器最后的计数方向：如果计时器已在相应的方向计数，则输出端为 TRUE。如果自上次调用 FB 以来未更改计数器，则两个输出端均为 FALSE。

在输入端 RESOLUTION，编码器分辨率可成倍评估：

1 = 普通分辨率 (-536 870 912...536 870 911，跟编码器分辨率相同)，

2 = 双倍分辨率评估 (-1 073 741 824...1 073 741 823),

4 = 四倍分辨率评估 (-2 147 483 648...2 147 483 647).

该输入端的所有其他值表示普通分辨率。

	RESOLUTION = 1 每四个边沿进行计数 (= 编码器分辨率)。
	RESOLUTION = 2 每两个边沿进行计数
	RESOLUTION = 4 每个上升和下降边沿进行计数

输入端参数

10259

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 开始的进程在后台继续 > FB 输出端未更新
CHANNEL	BYTE	输入通道对编号 12 = 通道对 0 = 输入端 I12 + I13 14 = 通道对 1 = 输入端 I14 + I15
PRESET_VALUE	DINT	计数器初始值
PRESET	BOOL	TRUE (仅针对 1 个周期): 加载初始值 PRESET_VALUE FALSE: 计数器已启用
RESOLUTION	BYTE	评估编码器分辨率: 01 = 每四个边沿进行计数 (= 编码器分辨率) 02 = 每两个边沿进行计数 04 = 每个上升和下降边沿进行计数 所有其他值计为 "01"。

输出端参数

8138

参数	数据类型	说明
VALUE	DINT	如果 RESOLUTION = 1: VALUE = -536 870 912...536 870 911 (= DINT 的 ¼ 范围) 如果 RESOLUTION = 2: VALUE = -1 073 741 824...1 073 741 823 (= DINT 的 ½ 范围) 如果 RESOLUTION = 4: VALUE = -2 147 483 648...2 147 483 647 (= DINT 的范围)
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
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	功能块已启用 - 还没有可用的有效值
130	82	通道设定无效
138	8A	分辨率设定无效

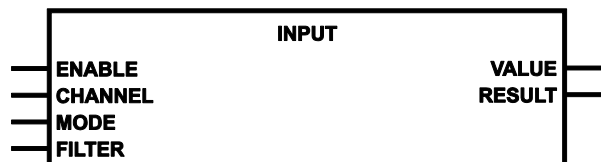
INPUT

8103

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8104

INPUT 有助于确定输入通道的状态 (→ 数据表)。FB 提供所选通道的当前状态。

测量值和输出值源自通过 MODE 显示的工作模式:

- 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
- 模拟输入端 0...20 mA
- 模拟输入端 0...10 V
- 模拟输入端 0...32 V
- 模拟输入端 比率计 0...32 V
- 模拟输入端 电阻测量 16...30 000 Ω (CR04nn, 来自 HW 版本 AD: 16...30 000 Ω)

! 运行期间不得更改工作模式。

模拟值作为标准值提供。

输入端参数

15879

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																														
CHANNEL	BYTE	输入通道编号 输入端 IN00...IN15 对应 0...15																														
MODE	BYTE	输入通道工作模式 : <table> <tr> <td>0 = 0x00</td><td>关闭</td><td></td></tr> <tr> <td>1 = 0x01</td><td colspan="2">二进制输入端, 正极开关 (BL), 数字评估</td></tr> <tr> <td>3 = 0x03</td><td>电压输入端</td><td>0...10 000 mV</td></tr> <tr> <td>6 = 0x06</td><td>电压输入端, 比率</td><td>0...1 000 ‰</td></tr> <tr> <td>7 = 0x07</td><td>电流输入端</td><td>0...20 000 µA</td></tr> <tr> <td>9 = 0x09</td><td>电压输入端</td><td>0...32 000 mV</td></tr> <tr> <td>10 = 0x0A</td><td colspan="2">二进制输入端, 正极开关 (BL) (预设)</td></tr> <tr> <td>11 = 0x0B</td><td colspan="2">二进制输入端, 正极开关 (BL), 含诊断 (Namur)</td></tr> <tr> <td>12 = 0x0C</td><td colspan="2">二进制输入端, 负极开关 (BH)</td></tr> <tr> <td>18 = 0x12</td><td>电阻输入端</td><td>16...30 000 Ω</td></tr> </table>	0 = 0x00	关闭		1 = 0x01	二进制输入端, 正极开关 (BL), 数字评估		3 = 0x03	电压输入端	0...10 000 mV	6 = 0x06	电压输入端, 比率	0...1 000 ‰	7 = 0x07	电流输入端	0...20 000 µA	9 = 0x09	电压输入端	0...32 000 mV	10 = 0x0A	二进制输入端, 正极开关 (BL) (预设)		11 = 0x0B	二进制输入端, 正极开关 (BL), 含诊断 (Namur)		12 = 0x0C	二进制输入端, 负极开关 (BH)		18 = 0x12	电阻输入端	16...30 000 Ω
0 = 0x00	关闭																															
1 = 0x01	二进制输入端, 正极开关 (BL), 数字评估																															
3 = 0x03	电压输入端	0...10 000 mV																														
6 = 0x06	电压输入端, 比率	0...1 000 ‰																														
7 = 0x07	电流输入端	0...20 000 µA																														
9 = 0x09	电压输入端	0...32 000 mV																														
10 = 0x0A	二进制输入端, 正极开关 (BL) (预设)																															
11 = 0x0B	二进制输入端, 正极开关 (BL), 含诊断 (Namur)																															
12 = 0x0C	二进制输入端, 负极开关 (BH)																															
18 = 0x12	电阻输入端	16...30 000 Ω																														
FILTER	BYTE	针对输入端测量的过滤器 : 有效 = 0...8 建议 = 4 → 章节 配置输入端软件过滤器 (→ 页 59)																														

输出端参数

8106

参数	数据类型	说明
VALUE	WORD	输入通道的当前值或状态 (根据所选操作模式)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

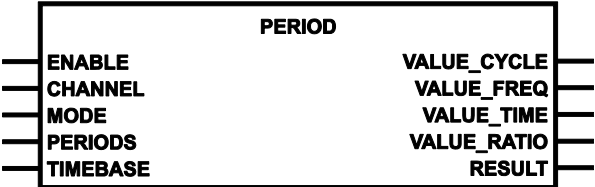
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 - 数据有效
2	02	功能块已启用 (操作还未完成)
3	03	功能块已启用 - 还没有可用的有效值
130	82	通道设定无效
132	84	模式设定无效
136	88	过滤设定无效
141	8D	发生了断线
142	8E	发生了电源电压短路
144	90	输入端电流太高

PERIOD

8122

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

15850

PERIOD 测量特定通道上的频率 (以 [Hz] 为单位)、周期 (周期时间, 以 [μs] 为单位) 或相移 (以 [°] 为单位, 具体视设定的模式而定。

MODE 十进制 十六进制		说明
0	00	无测量
14	0E	频率测量 在特定时间内对上升沿进行计数。
19	13	周期测量 (以 MODE = 20 代替为佳!) 测量两个上升沿之间的时间间隔。 指定特定数量的周期内的平均值。
20	14	周期和比率测量 测量两个上升沿之间的时间间隔。 指定特定数量的周期内的平均值。
25	19	(LZS 版本 03.02.zz 或更高) 输入通道对通道 A 和通道 B 之间的相移 (0...359°) (仅系统不会出现 > 179°的大幅跳跃时, 消息有意义)

❗运行期间不得更改工作模式。

❗ 如果 MODE=19 或 MODE=20 或 MODE=25 :
允许的输入频率 = 0.1...3 000 Hz
如果负载太高, 周期时间则可能长到不可接受。
→ 章节 **装置的性能极限** (→ 页 [44](#))

①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

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 开始的进程在后台继续 > FB 输出端未更新
CHANNEL	BYTE	(MODE = 14 / 20) 快速输入通道编号 输入端 IN12...IN15 对应 12...15 (MODE = 25) 快速输入 A 通道编号 输入端 IN12 / IN14 对应 12 / 14 B 通道 = A 通道 + 1
	BYTE	功能块的工作模式： 0 = 0x00 = 无测量 14 = 0x0E = 频率测量 19 = 0x13 = 间隔测量 20 = 0x14 = 间隔和比率测量 25 = 0x19 = 两个输入信号的相位差
PERIODS	BYTE	待平均周期数 (1...4) • 若 MODE = 19 / 20 ⇒ 算数平均 • 若 MODE = 25 ⇒ 几何平均 • 若 PERIODS = 1 ⇒ 不作平均
TIMEBASE	TIME	(仅针对 MODE = 14 时) 对沿进行计数的时间，单位 [ms] 允许值 = 1...2 000

输出端参数

8125

参数	数据类型	说明
VALUE_CYCLE	DWORD	输入端周期时间，单位 [μs] 输入组 A 通道周期时间，单位 [μs]
VALUE_FREQ	REAL	输入端已检测周期的频率，单位 [Hz] 输入组 A 通道已检测周期的频率，单位 [Hz]
VALUE_TIME	TIME	值 = 0 (未检测) 最后一个上升沿以来消耗的时间 最后一次有效测量以来消耗的时间
VALUE_RATIO	WORD	值 = 0 (未检测) 输入信号占空比，单位 [%] 相位差，单位 [°]
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	功能块已启用 - 还没有可用的有效值
130	82	通道设定无效
132	84	模式设定无效
137	89	PERIODS 或 TIMEBASE 的值无效
146	92	周期太长

5.2.6 功能元件：输出端功能

内容	
CURRENT_CONTROL	230
OUTPUT	233
PWM1000	236

15075
10462

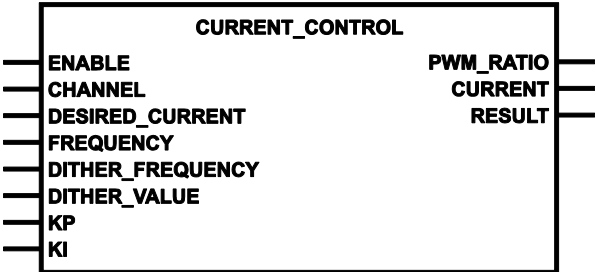
就此装置而言，您可设定某些或全部输出端的模式。 您可在此了解它的一些功能元件。

CURRENT_CONTROL

8082

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8086

CURRENT_CONTROL 作为 PWMI 输出端的电流控制器运行。

控制器的运行与 PWM 信号的持续周期无关。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。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）。
- > 在所需值 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。

根据使用的控制器硬件，需注意不同的示教性能。

⚠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 PWM 频率低于 100 Hz，另加额外的抖动频率，电流控制不可能再达到所示的准确度（→ 数据表）。

输入端参数

17890

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 开始的进程在后台继续 > FB 输出端未更新
CHANNEL	BYTE	电流控制输出通道编号 输出端 OUT00...OUT01 对应 0...1
DESIRED_CURRENT	WORD	输出端所需电流值，以 [mA] 为单位
FREQUENCY	WORD	允许的输出端 PWM 频率，以 [Hz] 为单位 允许 = 20...250 = 0x0014...0x00FA
		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

输出端参数

8088

参数	数据类型	说明
PWM_RATIO	WORD	针对监控目的：显示 PWM 脉冲比 0...1000 ‰
CURRENT	WORD	仅适用于电流控制输出端： 电流输出端电流，以 [mA] 为单位
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	功能块已启用 - 还没有可用的有效值
128	80	VBBx 欠电压
129	81	VBBx 过电压
130	82	通道设定无效
131	83	DESIRED_CURRENT 的值无效
134	86	抖动设定无效

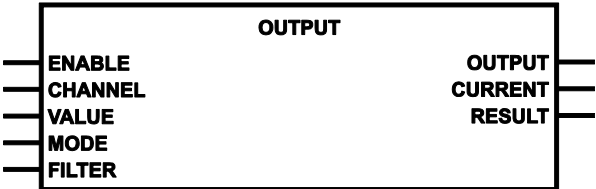
OUTPUT

8078

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8079

OUTPUT 将工作模式分配至输出通道 (→ 数据表)。 FB 有助于检测所选输入通道的状态。

测量值和输出值源自通过 MODE 显示的工作模式:

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

❗运行期间不得更改工作模式。

输入端参数

17871

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
CHANNEL	BYTE	输出通道编号 输出端 OUT00...OUT07 对应 0...7
VALUE	BOOL	TRUE: 启用输出端 FALSE: 禁用输出端
MODE	BYTE	输出端工作模式： 0 = 0x00 = 关闭 2 = 0x02 = 二进制输出端正极开关 15 = 0x0F = 二进制输出端正极开关，含诊断 16 = 0x10 = 二进制输出端正极开关，含诊断和保护
FILTER	BYTE	❗ 仅针对包含电流反馈的输出端： 针对输出端测量的过滤器： 有效 = 0...8 建议 = 4 → 章节 配置输出端软件过滤器 (→ 页 63) ❗ 针对不含电流反馈的输出端： FILTER = 0 或：切勿设定参数 FILTER !

❗ 运行期间不得更改工作模式。

输出端参数

8081

参数	数据类型	说明
OUTPUT	BOOL	TRUE: 输出端启用 FALSE: 输出端禁用
CURRENT	WORD	仅适用于电流控制输出端： 电流输出端电流，以 [mA] 为单位
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

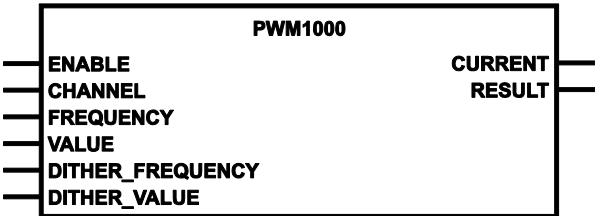
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
2	02	功能块已启用 (操作还未完成)
3	03	功能块已启用 - 还没有可用的有效值
128	80	VBBx 欠电压
129	81	VBBx 过电压
130	82	通道设定无效
132	84	模式设定无效
136	88	过滤设定无效
141	8D	检测到了断线 (针对二进制输出端, 正极开关 (BH), 含诊断)
142	8E	检测到了短路 (针对二进制输出端正极开关 (BH), 含诊断)
145	91	输出端电流太高 (针对二进制输出端正极开关 (BH), 含诊断和保护)

PWM1000

8060

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8062

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 FB。就每个通道而言，自身 PWM 频率、传号空号比和抖动频率均可设定。
PWM 频率 FREQUENCY 可直接以 [Hz] 为单位显示，传号空号比 VALUE 以 1 % 为单位增减。

输入端参数

17875

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 开始的进程在后台继续 > FB 输出端未更新
CHANNEL	BYTE	PWM 输出通道编号 输出端 OUT00...OUT11 对应 0...11
FREQUENCY	WORD	PWM 频率，以 [Hz] 为单位 允许 = 20...250 = 0x0014...0x00FA
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
		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

输出端参数

8523

参数	数据类型	说明
CURRENT	WORD	仅适用于电流控制输出端： 电流输出端电流，以 [mA] 为单位
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	功能块已启用 - 还没有可用的有效值
128	80	VBBx 欠电压
130	82	通道设定无效
131	83	VALUE 的值无效
133	85	FREQUENCY 的值无效
134	86	抖动设定无效

5.2.7 功能元件：系统

内容	
FLASH_INFO	239
FLASH_READ	241
GET_APP_INFO	243
GET_HW_INFO	245
GET_IDENTITY	247
GET_SW_INFO	248
GET_SW_VERSION	250
MEM_ERROR	252
MEMCPY	254
OHC	256
SET_IDENTITY	258
SET_LED	259
SET_PASSWORD	262
TIMER_READ_US	264

15067

我们将在此讲述可供您进行以下操作的 IFM 功能：

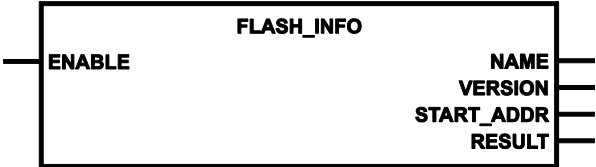
- 管理内存内容
- 读取软件和硬件信息
- 设定或读取各种数据和参数

FLASH_INFO

11580

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

11588

FLASH_INFO 从用户闪存读取信息：

- 内存区域名称（用户定义），
- 软件版本，
- 起始地址（通过 IEC 结构简单读取）。

输入端参数

11589

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

11590

参数	数据类型	说明
NAME	STRING(24)	内存区域名称（用户定义）
VERSION	STRING(24)	软件版本
START_ADDR	DWORD	数据的起始地址
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

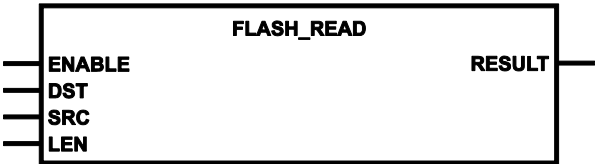
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
157	9D	软件标题无效（CRC 错误）

FLASH_READ

8147

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

11579

FLASH_READ 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。

- ▶ 通过 SRC + LEN 产生的地址必须 ≤ 65 408。
- ▶ 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 POU !

输入端参数

8148

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	目标地址 ❗ 通过运算符 ADR 确定地址并将其分配至 POU !
SRC	DWORD	内存中的相对起始地址 有效 = 0...65 407 = 0x0000 0000...0x0000 FF7F
LEN	WORD	待传送数据字节数量 (≥ 1)

输出端参数

8152

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
152	98	不允许的内存区域： - 无效的源地址 - 无效的目标地址 - 无效的字节数量

GET_APP_INFO

11581

= 获取应用程序信息
单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

11593

GET_APP_INFO 提供关于存储在装置上的应用程序软件的信息：

- 名称 (= CODESYS 项目文件名称) ，
- 版本 (= 来自 CODESYS 菜单 [Project] > [Project Info] > [Version]) ，
- 明确的 CoDeSys 版本号 ，
- CoDeSys 版本日期。

输入端参数

11594

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

11595

参数	数据类型	说明
NAME	STRING(24)	应用程序名称
VERSION	STRING(24)	应用程序的版本
BUILD_NUM	STRING(24)	唯一的 CODESYS 版本号 (例如 : "45")
BUILD_DATE	STRING(24)	CODESYS 版本日期 (例如 : "20111006123800")
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果 :

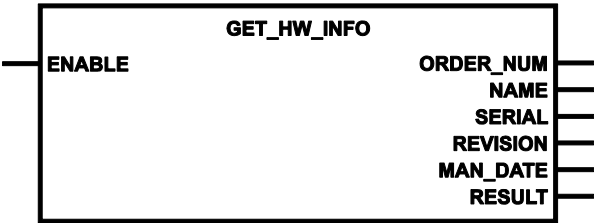
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效

GET_HW_INFO

11582

= 获取硬件信息
单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyyzz.LIB

CODESYS 中的符号:



说明

1599

GET_HW_INFO 提供关于装置硬件的信息：

- **IFM** 产品编号（例如 CR0403），
- 产品名称，
- 明确的序列号，
- 硬件版本，
- 生产日期。

输入端参数

11600

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

11601

参数	数据类型	说明
ORDER_NUM	STRING(24)	IFM 产品编号（例如：CR0403）
NAME	STRING(24)	产品名称（例如："BasicController 12/12"）
SERIAL	STRING(24)	装置的序列号（例如："000045784"）
REVISION	STRING(24)	装置的硬件修订版本（例如："V01.00.01"）
MAN_DATE	STRING(24)	装置制造日期（例如："20111007123800"）
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

GET_IDENTITY

8166

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

15411

GET_IDENTITY 读取装置中存储的 ID (之前已通过以下方式保存 : SET_IDENTITY (→ 页 [258](#))).

输入端参数

8167

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

8168

参数	数据类型	说明
APP_IDENT	STRING(80)	应用程序 ID 作为最多 80 个字符的字符串，例如： "Crane1704"
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
155	9B	不可读取值

GET_SW_INFO

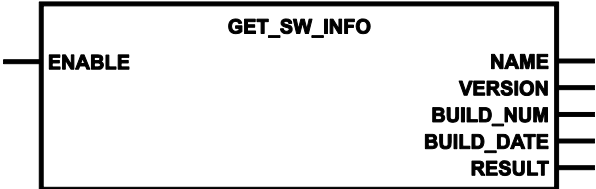
11583

= 获取软件信息

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

11596

GET_SW_INFO 提供关于装置系统软件的信息：

- 软件名称，
- 软件版本，
- 版本号，
- 版本日期。

输入端参数

11597

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

11598

参数	数据类型	说明
NAME	STRING(24)	系统软件的名称（例如：“BasicSystem”）
VERSION	STRING(24)	系统软件的版本（例如：“V02.00.03”）
BUILD_NUM	STRING(24)	系统软件的版本号（例如：“45”）
BUILD_DATE	STRING(24)	系统软件的版本日期（例如：“20111006123800”）
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

GET_SW_VERSION

14763

= 获取软件版本
单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyyzz.LIB

CODESYS 中的符号:



说明

14765

GET_SW_VERSION 提供关于装置中的软件的信息：

- BasicSystem 版本
- Bootloader 版本
- SIS 版本
- IEC 应用程序版本
- IEC 用户闪存版本

输入端参数

14766

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

14767

参数	数据类型	说明
BS_VERSION	STRING(24)	BasicSystem 版本
BL_VERSION	STRING(24)	Bootloader 版本
SIS_VERSION	STRING(24)	SIS 版本 (SIS = 系统信息服务)
IEC_VERSION	STRING(24)	IEC 应用程序版本
USER_VERSION	STRING(24)	IEC 用户闪存版本
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

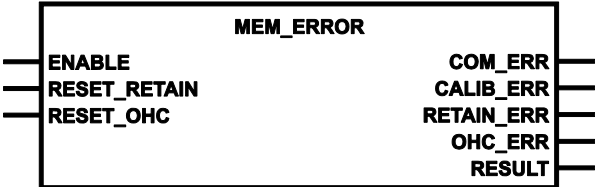
值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效

MEM_ERROR

14770

= 内存错误
单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyyzz.LIB

CODESYS 中的符号:



说明

14772

MEM_ERROR 利用信号通知某些参数或内存中的错误。
内存区域可通过相应的 FB 输入端删除。

输入端参数

14773

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
RESET_RETAIN	BOOL	TRUE: 删除非易失保留内存 FALSE: 内存内容无更改
RESET_OHC	BOOL	TRUE: 删除非易失 OHC 内存 FALSE: 内存内容无更改

输出端参数

14774

参数	数据类型	说明
COM_ERR	BOOL	下载 ID 和波特率设为默认值 (下载参数丢失)
CALIB_ERR	BOOL	校准值无效 (模拟输入端、PWM 输出端、系统电压)
RETAIN_ERR	BOOL	保留内存无效 (例如因为强磁场而部分删除)
OHC_ERR	BOOL	OHC 值无效 (例如因为强磁场而部分删除)
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

MEMCPY

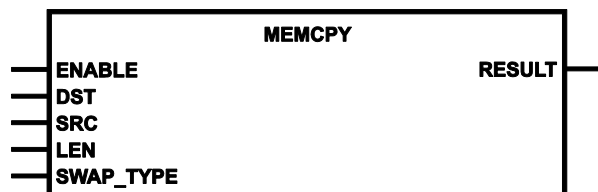
8160

= 内存复制

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

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 POU！

> 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。

输入端参数

8162

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	目标地址  通过运算符 ADR 确定地址并将其分配至 POU !
SRC	DWORD	源地址
LEN	WORD	待传送数据字节数量 (≥ 1)
SWAP_TYPE	BYTE	交换字节顺序 0 = 无交换 例如 : 1A 2B 3C 4D ⇨ 1A 2B 3C 4D 1 = 交换 2 个字节 (WORD、INT ...) 例如 : 1A 2B 3C 4D ⇨ 2B 1A 4D 3C  LEN 必须为 2 的倍数 ! 2 = 交换 4 个字节 (DWORD、DINT、REAL、TIME ...) 例如 : 1A 2B 3C 4D ⇨ 4D 3C 2B 1A  LEN 必须为 4 的倍数 !

输出端参数

8163

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
152	98	不允许的内存区域： • 无效的源地址 • 无效的目标地址 • 无效的字节数量
156	9C	不允许的值： • 针对 SWAP_TYPE 的无效值 • LEN 不匹配 SWAP_TYPE

OHC

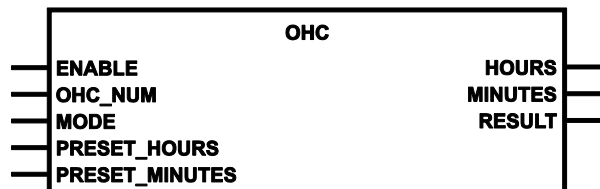
14777

= 工作时间计数器

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

14778

OHC 提供 4 个通用的工作时间计数器。

但如果硬件版本 < AD：仅可使用 2 个工作时间计数器。

有效的计数范围：0:00...4 294 967 295:59 小时 (= 490 293 年 25 天 15 小时)

❗ 如果装置硬件版本 < AD：

复位针对 OHC 的内存区域一次：

► 在 FB 中 **MEM_ERROR** (→ 页 [252](#))，设定输入端 RESET_OHC = TRUE！

> 仅现在可使用工作时间计数器。

输入端参数

14779

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 开始的进程在后台继续 > FB 输出端未更新
OHC_NUM	BYTE	工作时间计数器 计数器编号 (0...3)
MODE	BYTE	计数器工作模式 允许值 = 0 = 停止计数器 1 = 在最后存储值位置继续计数 2 = 复位计数器 3 = 通过以下值预设计数器
PRESET_HOURS	DWORD	预设小时 (0...4 294 967 295 = 0x0000 0000...0xFFFF FFFF)
PRESET_MINUTES	BYTE	预设分钟 (0...59 = 0x00...0x3B)

输出端参数

14780

参数	数据类型	说明
HOURS	DWORD	小时值计数 (0...4 294 967 295 = 0x0000 0000...0xFFFF FFFF)
MINUTES	BYTE	分钟值计数 (0...59 = 0x00...0x3B)
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
130	82	OHC_NUM 中的计数器数字无效
131	83	预设值无效
132	84	模式设定无效
158	9E	剩余内存无效 (CRC 错误)

SET_IDENTITY

8174

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8535

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。

► 可读取该 ID 以识别加载的程序：

- 通过软件“维护工具”
- 通过 FB 在应用程序中 **GET_IDENTITY** (→ 页 [247](#))

输入端参数

8175

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
APP_IDENT	STRING(80)	应用程序 ID 作为最多 80 个字符的字符串，例如：“Crane1704” 通过 APP_IDENT = "" 复位

输出端参数

8176

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

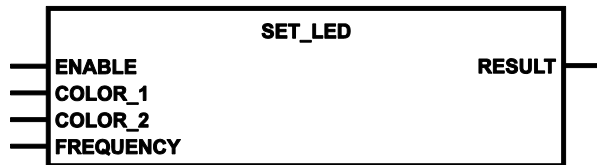
SET_LED

8052

单元类型 = 功能块 (FB)

元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8054

可通过 SET_LED 在应用程序中更改状态 LED 的频率和颜色。

❗ 如果在应用程序中更改闪烁模式，则默认设定表不再有效 (→ 章节 **状态 LED** (→ 页 [31](#))).

输入端参数

8223

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
COLOR_1	BYTE	“开启”的 LED 颜色 来自数据结构 "System LED Color" 的颜色常量；允许： 00 = LED_BLACK (= LED 熄灭) 02 = LED_GREEN
COLOR_2	BYTE	“关闭”的 LED 颜色 来自数据结构 "System LED Color" 的颜色常量；允许： 00 = LED_BLACK (= LED 熄灭) 02 = LED_GREEN
FREQUENCY	BYTE	LED 闪烁频率 来自数据结构 "System LED Frequency" 的频率常量；允许： 00 = LED_0HZ = 永久亮起 01 = LED_05HZ = 以 0.5 Hz 的频率闪烁 02 = LED_1HZ = 以 1 Hz 的频率闪烁 04 = LED_2HZ = 以 2 Hz 的频率闪烁 10 = LED_5HZ = 以 5 Hz 的频率闪烁

输出端参数

8227

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	功能块执行完成且无错误
2	02	功能块已启用 (操作还未完成)
133	85	FREQUENCY 的值无效
151	97	颜色值无效

SET_PASSWORD

8178

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.LIB

CODESYS 中的符号:



说明

8179

SET_PASSWORD 设定通过维护工具进行程序和内存上传的用户密码。

如果用户密码启用，则仅可在输入正确的密码时通过维护工具读取应用程序或数据内存。

如果将空串（默认条件）分配至 PASSWORD 输入端，则密码复位。 应用程序软件或数据内存可随时上传。

❗ 加载新的应用程序时，密码复位。

输入端参数

8180

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
PASSWORD	STRING(16)	密码 如果 PASSWORD = ""，则无需输入密码即可访问

输出端参数

8181

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

TIMER_READ_US

8219

单元类型 = 功能块 (FB)
元件类型 ifm_CR2530_Vxxyzz.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

8220

参数	数据类型	说明
TIME_US	DWORD	当前系统时间 [μs]
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

6 诊断和错误处理

内容	
诊断.....	265
故障.....	265
响应系统错误	266
CAN / CANopen: 错误和错误处理	266

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 [41](#))

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 响应系统错误

8504

原则上，程序员有责任对应用程序中的错误消息采取措施。错误说明通过错误消息提供。

- > 只要不再显示错误导致状态，系统即会复位错误消息。

6.3.1 响应错误消息的进程示例

8505

运行时系统循环写入系统标志 TEMPERATURE。

应用程序通过检索 INT 变量来检测装置温度。

如果超过或未达到应用程序的允许值：

- > 应用程序禁用输出端。
- ▶ 修正错误的原因。
- > 应用程序检测返回至正常的温度值：
机器/系统可重启或操作可继续。

6.4 CAN / CANopen: 错误和错误处理

19604

→ 系统手册“ecomatmobile 技巧”

→ 章节 **CAN / CANopen：错误和错误处理**

7 附录

内容	
系统标志	268
地址分配和 I/O 工作模式	269
集成 I/O 模块：说明	276
错误表	353

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

8374



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！

系统标志 (符号名称)	类型	说明
SUPPLY_VOLTAGE_VBBx	WORD	VBBx 上的电源电压，以 [mV] 为单位 CR040n: x = 1, 2 CR041n: x = 1, 2 CR253n: x = 1, 2
SUPPLY_VOLTAGE_VBBS	WORD	VBBS 上的电源电压，以 [mV] 为单位
SUPPLY_VOLTAGE_VU	WORD	内部电源电压，以 [mV] 为单位
TEMPERATURE	INT	控制器内部温度，单位 [°C]

7.2 地址分配和 I/O 工作模式

内容	
I/O 地址/变量	269
可能的输入端/输出端工作模式.....	272

1656

→ 另有技术资料

7.2.1 I/O 地址/变量

内容	
输入端： 地址和变量	270
输出端： 地址和变量	271

2376

输入端：地址和变量

21043

IEC 地址	I/O 变量	备注
%IB0	IN00	二进制输入/模拟输入通道 0
%IB1	IN01	二进制输入/模拟输入通道 1
%IB2	IN02	二进制输入/模拟输入通道 2
%IB3	IN03	二进制输入/模拟输入通道 3
%IB4	IN04	二进制输入/电阻输入通道 4
%IB5	IN05	二进制输入/电阻输入通道 5
%IB6	IN06	二进制输入通道 6
%IB7	IN07	二进制输入通道 7
%IB8	IN08	二进制输入通道 8
%IB9	IN09	二进制输入通道 9
%IB10	IN10	二进制输入通道 10
%IB11	IN11	二进制输入通道 11
%IB12	IN12	二进制输入/快速输入通道 12
%IB13	IN13	二进制输入/快速输入通道 13
%IB14	IN14	二进制输入/快速输入通道 14
%IB15	IN15	二进制输入/快速输入通道 15
%IW8	SUPPLY_VOLTAGE_VBBS	电源 VBBS 电压，以 [mV] 为单位
%IW9	SUPPLY_VOLTAGE_VU	电源 VU 电压，以 [mV] 为单位
%IW10	SUPPLY_VOLTAGE_VBB1	电源 VBB1 电压，以 [mV] 为单位
%IW11	SUPPLY_VOLTAGE_VBB2	电源 VBB2 电压，以 [mV] 为单位
%IW12	TEMPERATURE	温度，以 [°C] 为单位

输出端：地址和变量

21044

IEC 地址	I/O 变量	备注
%QB0	OUT00	二进制输出/PWM 输出通道 0
%QB1	OUT01	二进制输出/PWM 输出通道 1
%QB2	OUT02	二进制输出/PWM 输出通道 2
%QB3	OUT03	二进制输出/PWM 输出通道 3
%QB4	OUT04	二进制输出/PWM 输出通道 4
%QB5	OUT05	二进制输出/PWM 输出通道 5
%QB6	OUT06	二进制输出/PWM 输出通道 6
%QB7	OUT07	二进制输出/PWM 输出通道 7
%QB8	OUT08	二进制输出/PWM 输出通道 8
%QB9	OUT09	二进制输出/PWM 输出通道 9
%QB10	OUT10	二进制输出/PWM 输出通道 10
%QB11	OUT11	二进制输出/PWM 输出通道 11
%QB12	OUT12	二进制输出通道 12
%QB13	OUT13	二进制输出通道 13
%QB14	OUT14	二进制输出通道 14
%QB15	OUT15	二进制输出通道 15

7.2.2 可能的输入端/输出端工作模式

内容

输入端：工作模式	272
输出端：工作模式	274

2386

输入端：工作模式

15746

 = 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	FB 输入端	值	
				十进制	十六进制
IN00...IN03	关闭	INPUT	MODE	0	00
	电压输入端 0...10 000 mV	INPUT	MODE	3	03
	电压输入比率 0...1 000 ‰	INPUT	MODE	6	06
	电流输入端 0...20 000 µA	INPUT	MODE	7	07
	电压输入端 0...32 000 mV	INPUT	MODE	9	09
	二进制输入端 正极开关	INPUT	MODE	10	0A
	二进制输入端，含诊断 (Namur) 正极开关	INPUT	MODE	11	0B
	二进制输入端 负极开关	INPUT	MODE	12	0C
IN04...IN05	关闭	INPUT	MODE	0	00
	二进制输入端 正极开关	INPUT	MODE	10	0A
	二进制输入端，含诊断 (Namur) 正极开关	INPUT	MODE	11	0B
	电阻输入端 16...30 000 ohms	INPUT	MODE	18	12
IN06...IN11	关闭	INPUT	MODE	0	00
	二进制输入端 正极开关	INPUT	MODE	10	0A
	二进制输入端，含诊断 (Namur) 正极开关	INPUT	MODE	11	0B
IN12...IN15	关闭	INPUT	MODE	0	00
	二进制输入端，数字评估 正极开关	INPUT	MODE	1	01
	频率测量 0...30 000 Hz	PERIOD	MODE	14	0E
	周期和比率测量 0.1...3 000 Hz	PERIOD	MODE	20	14

输入端	可能的工作模式	利用功能块设定	FB 输入端	值	
	相位差 0...359°	PERIOD	MODE	25	19
	递增计数器 递减计数器 0...30 000 Hz	FASTCOUNT	MODE	21 22	15 16
	检测编码器 0...1 0000 Hz	INC_ENCODER			

利用以下功能块设定工作模式：

INPUT (→ 页 223)	将工作模式分配至输入通道 提供所选通道的当前状态
FASTCOUNT (→ 页 217)	针对快速输入脉冲的计数功能块
INC_ENCODER (→ 页 219)	针对编码器评估的递增/递减计数器功能
PERIOD (→ 页 226)	- 在所示通道测量： 频率和周期长度（周期时间），以 [μs] 为单位 - 在所示通道对测量： 通道 A 和通道 B 之间的相移，以 [°] 为单位

输出端：工作模式

15747

= 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	FB 输入端	值	
					十进制	十六进制
OUT00 ...OUT01	关闭		OUTPUT	MODE	0	00
	二进制输出端	正极开关	OUTPUT	MODE	2	02
	二进制输出端，含诊断	正极开关	OUTPUT	MODE	15	0F
	二进制输出端，含诊断和保护	正极开关	OUTPUT	MODE	16	10
	模拟输出端，含脉冲宽度调制		PWM1000			
	模拟电流控制输出		CURRENT_CONTROL			
OUT02 ...OUT07	关闭		OUTPUT	MODE	0	00
	二进制输出端	正极开关	OUTPUT	MODE	2	02
	二进制输出端，含诊断	正极开关	OUTPUT	MODE	15	0F
	二进制输出端，含诊断和保护	正极开关	OUTPUT	MODE	16	10
	模拟输出端，含脉冲宽度调制		PWM1000			
OUT08 ...OUT09	关闭		OUTPUT	MODE	0	00
	二进制输出端	正极开关	OUTPUT	MODE	2	02
	模拟输出端，含脉冲宽度调制		PWM1000			
	模拟输出端，含脉冲宽度调制，电压控制	(位于插脚 25 + 43)	PWM1000			
OUT10 ...OUT11	关闭		OUTPUT	MODE	0	00
	二进制输出端	正极开关	OUTPUT	MODE	2	02
	模拟输出端，含脉冲宽度调制		PWM1000			
OUT12 ...OUT15	二进制输出端	正极开关	---	---	---	---

利用以下功能块设定工作模式：

OUTPUT (→ 页 233)	将工作模式分配至输出通道 提供所选通道的当前状态
PWM1000 (→ 页 236)	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
CURRENT_CONTROL (→ 页 230)	PWMi 输出通道的电流控制

7.3 集成 I/O 模块：说明

内容	
系统说明 I/O 模块 ExB01.....	276
I/O 模块的配置	291
集成 I/O 模块的对象目录	305
I/O 模块的运行	346
针对集成 ExB01 I/O 模块的系统标志	349
I/O 模块错误消息.....	350

16418

7.3.1 系统说明 I/O 模块 ExB01

内容	
硬件说明 I/O 模块	276
接口说明 I/O 模块	289

16422

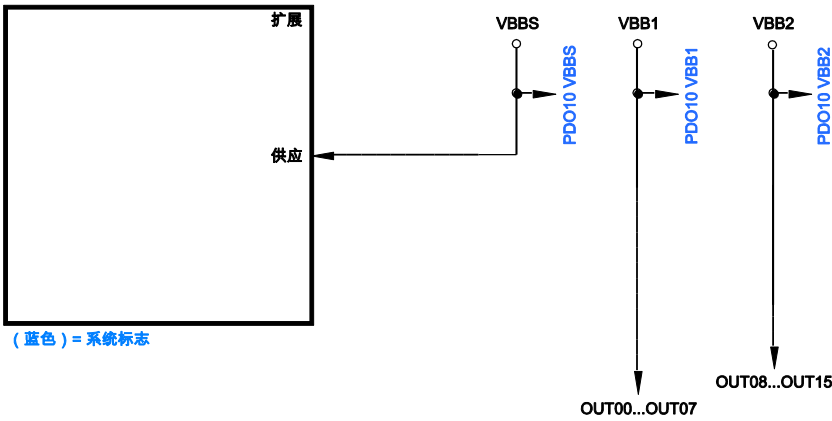
硬件说明 I/O 模块

内容	
硬件结构 I/O 模块	277
状态 LED I/O 模块	278
集成 I/O 模块 ExB01 输入端.....	279
集成 I/O 模块输出端 ExB01	285

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	
模拟输入端.....	280
二进制输入端	281
I/O 模块输入组 IN00...IN03	282
I/O 模块输入组 IN04...IN05	282
I/O 模块输入组 IN06...IN11	284
I/O 模块输入组 IN12...IN15	284

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：

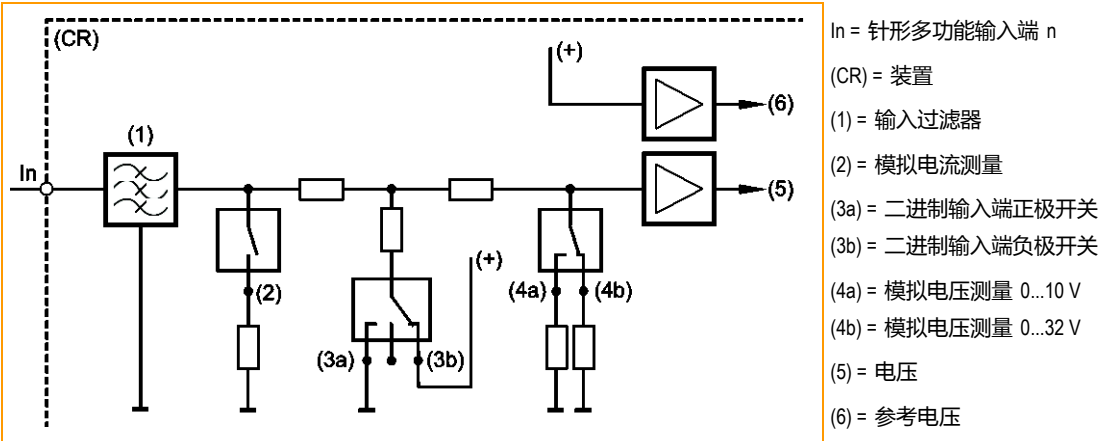
- 电流输入 0...20 mA
- 电压输入 0...10 V
- 电压输入 0...32 V
- 电阻测量 16...30 000 Ω (GND 测量)

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。

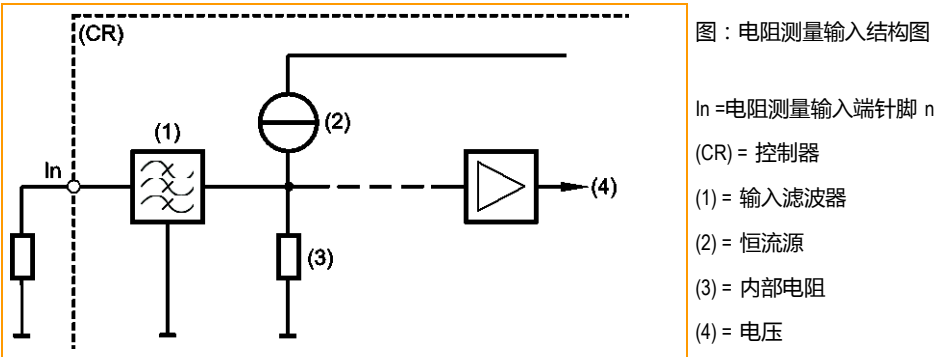
❗如果是比率测量, 则连接的传感器应供给装置 VBBs。 因此, 避免了偏移电压导致的错误测量。

8971



图：多功能输出端原理方块图

8972



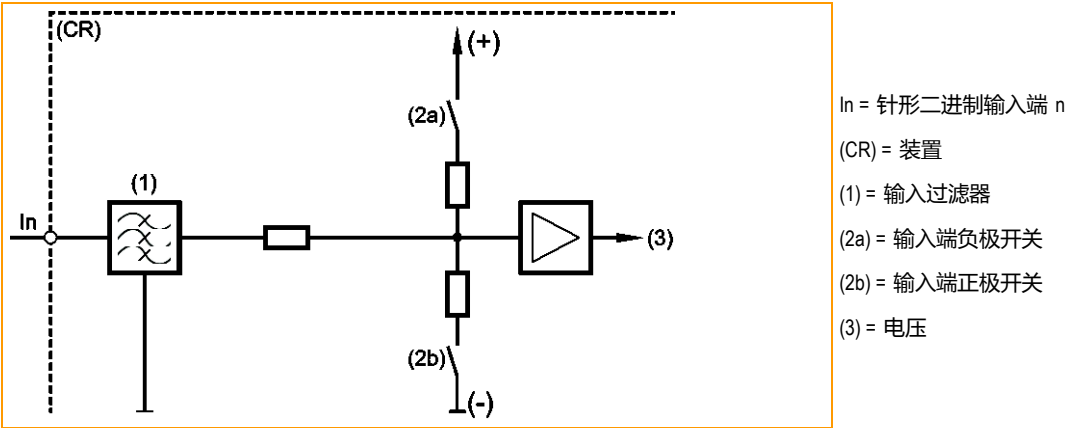
图：电阻测量输入结构图

二进制输入端

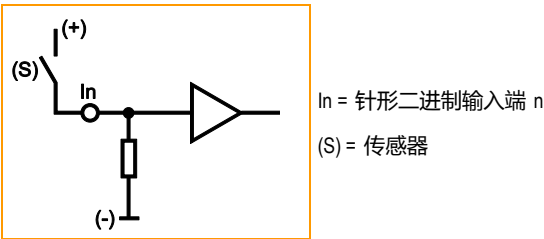
二进制输入端可在以下模式下运行：

- 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。



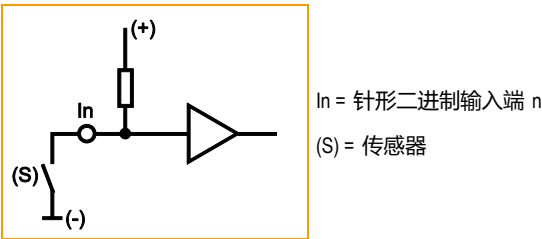
图：针对负极性和正极性传感器信号的二进制输入端负极开关/正极开关基本电路



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)



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 [300](#))

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 [294](#))

- > 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 [300](#))

► 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 [294](#))

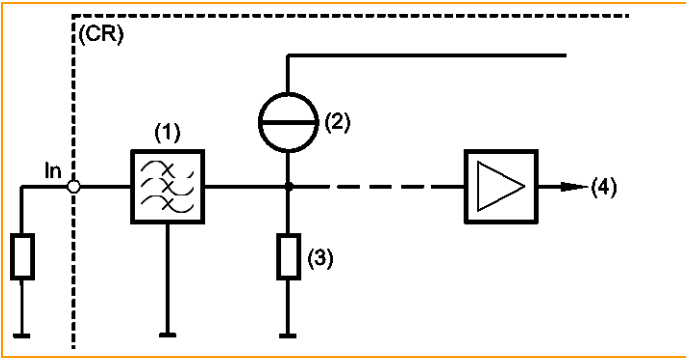
电阻测量

9773

这些输入端上的一般传感器：

- 液位
- 温度 (PT1000、NTC)

8972



图：电阻测量输入结构图

In =电阻测量输入端引脚 n

(CR) = 控制器

(1) = 输入滤波器

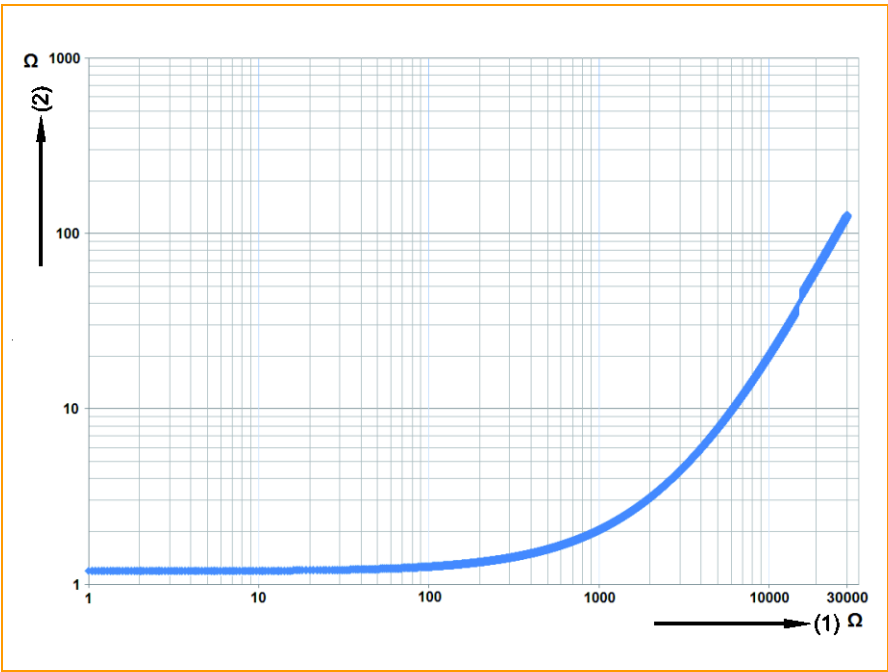
(2) = 恒流源

(3) = 内部电阻

(4) = 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 [300](#))

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 [294](#))

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 [300](#))

► 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 [294](#))

集成 I/O 模块输出端 ExB01

内容	
E/A-Modul Ausgangsgruppe OUT0, OUT1	285
I/O 模块输出组 OUT02...OUT07	287
I/O 模块输出组 OUT08...OUT09	288
I/O 模块输出组 OUT10...OUT11	288
I/O 模块输出组 OUT12...OUT15	288

16234

E/A-Modul Ausgangsgruppe OUT0, OUT1

50805
50805



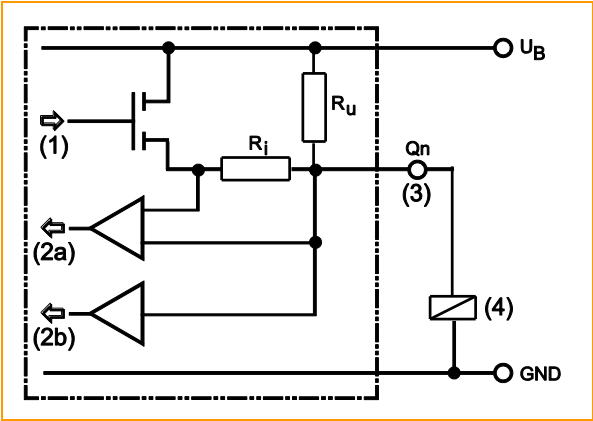
如果电源 VBB1 和 VBB2 没接则所有输出关闭

- 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 [300](#))
- 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 [296](#))
-  关于限值，请确保遵循技术资料！

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 ($R_L < 10\text{ k}\Omega$) 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

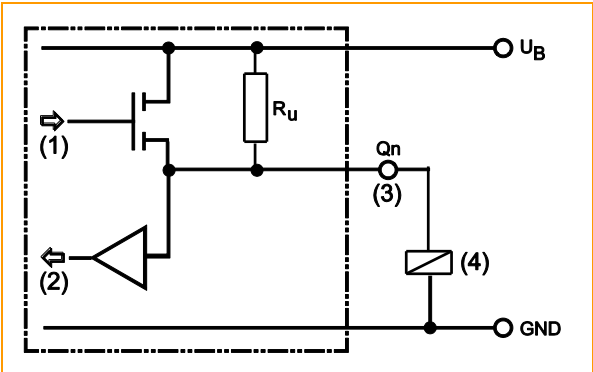
15808

- 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 [300](#))
- 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 [296](#))
- **!** 关于限值，请确保遵循技术资料！

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 ($R_L < 10\text{ k}\Omega$) 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 [300](#))

- ▶ 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 [296](#))
- ▶  关于限值，请确保遵循技术资料！

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 [300](#))

- ▶ 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 [296](#))
- ▶  关于限值，请确保遵循技术资料！

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 [300](#))

- ▶  关于限值，请确保遵循技术资料！

接口说明 I/O 模块

内容	
CAN 接口 I/O 模块	289

16426

CAN 接口 I/O 模块

内容	
CAN: 接口和协议： CR0133 中的 I/O 模块	290
CAN: 接口和协议： CR2532 中的 I/O 模块	290
连接集成 I/O 模块 ExB01 以作为 CANopen 从站	290

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

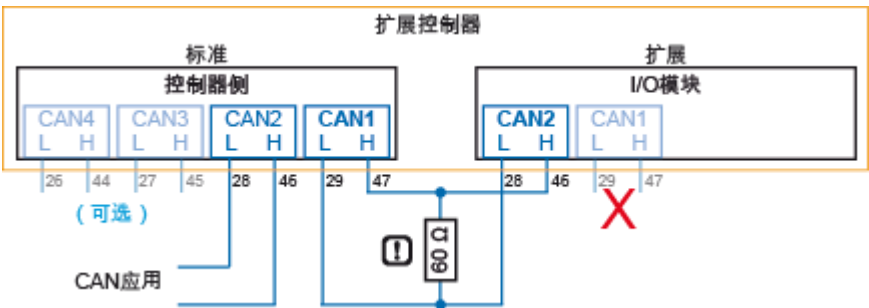
连接集成 I/O 模块 ExB01 以作为 CANopen 从站

15829

装置的集成 I/O 模块以 Smart Controller CR2530 为基础：

- 该侧预设作为 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	
设定编程系统（I/O 模块）	291
I/O 模块输入端和输出端的功能配置.....	294
I/O 模块可能的工作模式.....	300

16427

设定编程系统（I/O 模块）

内容	
手动设定编程系统（I/O 模块）	291
通过模板设定编程系统（I/O 模块）	293

16609

手动设定编程系统（I/O 模块）

内容	
集成内部 I/O 模块 ExB01.....	292

16610

集成内部 I/O 模块 ExB01

①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 (CR2530 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...]

① 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_Vxyyyz.EDS]。

> 结果：

```

 白---CANopen Master[VAR]
      白--- ExB01 (EDS) [VAR]
            白--- %QB256 Can-Output
            白--- %IB256 Can-Input
  
```

①CAN-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。

①I/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 [305](#))

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 **ecomatmobile** 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

配置集成 I/O 模块的输入端.....	294
配置集成 I/O 模块的输出端.....	296

16430

配置集成 I/O 模块的输入端

内容

配置输入端的软件过滤器 (I/O 模块)	294
模拟输入端：配置和诊断 (I/O 模块 ExB01)	294
二进制输入端：配置和诊断 (I/O 模块 ExB01)	295
快速输入端：I/O 模块 ExB01.....	295

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 [302](#))

- > 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

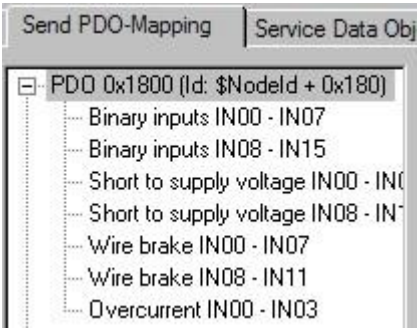
允许值 → 章节 **输入端：工作模式（I/O 模块）**（→ 页 [302](#)）

非电子开关的 NAMUR 型带诊断的开关量输入：

▶ 为开关配置额外的电阻连接件！

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）。

14677

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 [302](#)）

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

配置输出端的软件过滤器（I/O 模块）	296
二进制输出端：配置和诊断（I/O 模块 ExB01）	296
PWM 输出端：I/O 模块 ExB01	298

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

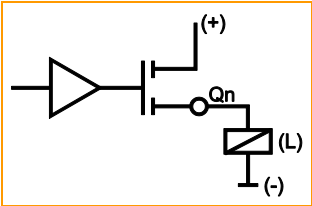
FILTER	过滤频率 [Hz]	针对以下范围的阶跃响应 [ms]			备注
		0...90 %	0...95 %	0...99 %	
固定	52	7.2	9.4	14.4	

二进制输出端：配置和诊断（I/O 模块 ExB01）

15882

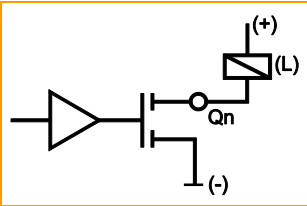
以下工作模式适用于装置输出端（→ 技术资料）：
• 二进制输出端，正极开关（BH），含/不含诊断功能

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 [303](#)）

二进制输出端：诊断（I/O 模块 ExB01）

15889

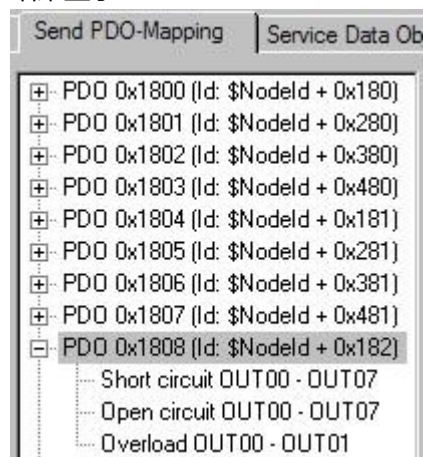
► 每个输出端均可通过 PLC 配置进行配置：

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 [303](#)）

> 结果显示 PDO 9：



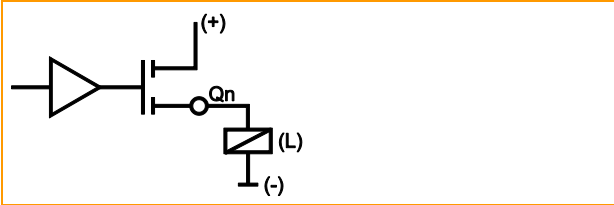
PWM 输出端： I/O 模块 ExB01

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 [303](#))

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

概述.....300

输入端： 工作模式（ I/O 模块 ）302

输出端： 工作模式（ I/O 模块 ）303

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 ）

SDO 值 十进制 十六进制		Mode	输入端 输出端	说明
15	0x0F	OUT_BINARY_HIGH_DIAG	输出端	二进制正极开关： 输出端 = FALSE $\Rightarrow$ 0 V 输出端 = TRUE $\Rightarrow$ 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 $\Rightarrow$ 0 V 输出端 = TRUE $\Rightarrow$ 输出端电源电压 断线和短路诊断 发生短路时输出端关闭。
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

 = 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
OUT10 ...OUT11	关闭		0x2000	0x1B...0x1C	0	0x00

输出端	可能的工作模式		对象目录中	子索引	值	
	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

对象目录参数表格，概述.....	305
对象目录参数表格，详细信息.....	319

15837

对象目录参数表格，概述

内容

概要.....	305
EDS 文件中的数据类型.....	306
对象目录强制对象（索引 0x1000...0x1FFF），概述.....	307
对象目录可选对象（索引 0x1000...0x1FFF），概述.....	308
对象目录制造商特定对象（索引 0x2000...0x6FFF），概述.....	317

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 • 10 ³⁸	3.402823466 • 10 ³⁸	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		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					
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					
0x1401	0x2	传送类型	配置 接收 PDO 1	1	是	立刻
		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
0x1402	0x2	传送类型	配置 接收 PDO 2	1	是	立刻
		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
0x1403	0x2	传送类型	配置 接收 PDO 3	1	是	立刻
		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
0x1600	0x2	传送类型	配置 接收 PDO 4	1	是	立刻
		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
0x1601	0x2	PDO 映射	映射 接收 PDO 1	0x6200 0208	是	预运行后
		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		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					
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		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 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		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 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		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					
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		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					
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

对象目录强制对象（索引 0x1000...0x1FFF），详细信息.....	319
对象目录可选对象（索引 0x1000...0x10FFF），详细信息.....	320
对象目录可选对象（索引 0x1400...0x14FF），详细信息.....	322
对象目录可选对象（索引 0x1600...0x16FF），详细信息.....	324
对象目录可选对象（索引 0x1800...0x18FF），详细信息.....	327
对象目录可选对象（索引 0x1A00...0x1AFF），详细信息.....	333
对象目录制造商特定对象（索引 0x2000...0x6FFF），详细信息	336

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：0x0nnntttt 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 关闭 输入端 IN00 3 = 0x03 0...10 000 mV 6 = 0x06 比率 0...1000 % 7 = 0x07 9 = 0x09 0...20 000 µA 10 = 0x0A 0...32 000 mV 11 = 0x0B 二进制正极开关 12 = 0x0C 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 关闭 输入端 IN01 3 = 0x03 0...10 000 mV 6 = 0x06 比率 0...1000 % 7 = 0x07 9 = 0x09 0...20 000 µA 10 = 0x0A 0...32 000 mV 11 = 0x0B 二进制正极开关 12 = 0x0C 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 关闭 输入端 IN02 3 = 0x03 0...10 000 mV 6 = 0x06 比率 0...1000 % 7 = 0x07 9 = 0x09 0...20 000 µA 10 = 0x0A 0...32 000 mV 11 = 0x0B 二进制正极开关 12 = 0x0C 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 关闭 输入端 IN03 3 = 0x03 0...10 000 mV 6 = 0x06 比率 0...1000 % 7 = 0x07 9 = 0x09 0...20 000 µA 10 = 0x0A 0...32 000 mV 11 = 0x0B 二进制正极开关 12 = 0x0C 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 关闭 输入端 IN04 10 = 0x0A 二进制正极开关 11 = 0x0B 二进制正极开关，含诊断 18 = 0x12 16...30 000 Ohm
	0x6	配置 IN05	rw	USINT	10	0 = 0x00 关闭 输入端 IN05 10 = 0x0A 二进制正极开关 11 = 0x0B 二进制正极开关，含诊断 18 = 0x12 16...30 000 Ohm
0x2000	0x7	配置 IN06	rw	USINT	10	0 = 0x00 关闭 输入端 IN06 10 = 0x0A 二进制正极开关 11 = 0x0B 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 关闭 输入端 IN15 1 = 0x01 二进制正极开关, 数字监控 14 = 0x0E 频率 0...30 000 Hz 19 = 0x13 周期 20 = 0x14 周期比率 0...1 000 ‰ 21 = 0x15 向上计数 22 = 0x16 向下计数 23 = 0x17 增量编码器
0x2000	0x11	配置 OUT00	rw	USINT	2	0 = 0x00 关闭 输入端 OUT00 2 = 0x02 二进制正极开关 4 = 0x04 PWM 输出端 5 = 0x05 电流控制 15 = 0x0F 二进制正极开关, 含诊断 16 = 0x10 二进制正极开关, 含诊断 + 保护
	0x12	配置 OUT01	rw	USINT	2	0 = 0x00 关闭 输入端 OUT01 2 = 0x02 二进制正极开关 4 = 0x04 PWM 输出端 5 = 0x05 电流控制 15 = 0x0F 二进制正极开关, 含诊断 16 = 0x10 二进制正极开关, 含诊断 + 保护
0x2000	0x13	配置 OUT02	rw	USINT	2	0 = 0x00 关闭 输入端 OUT02 2 = 0x02 二进制正极开关 4 = 0x04 PWM 输出端 15 = 0x0F 二进制正极开关, 含诊断 16 = 0x10 二进制正极开关, 含诊断 + 保护
	0x14	配置 OUT03	rw	USINT	2	0 = 0x00 关闭 输入端 OUT03 2 = 0x02 二进制正极开关 4 = 0x04 PWM 输出端 15 = 0x0F 二进制正极开关, 含诊断 16 = 0x10 二进制正极开关, 含诊断 + 保护
	0x15	配置 OUT04	rw	USINT	2	0 = 0x00 关闭 输入端 OUT04 2 = 0x02 二进制正极开关 4 = 0x04 PWM 输出端 15 = 0x0F 二进制正极开关, 含诊断 16 = 0x10 二进制正极开关, 含诊断 + 保护
	0x16	配置 OUT05	rw	USINT	2	0 = 0x00 关闭 输入端 OUT05 2 = 0x02 二进制正极开关 4 = 0x04 PWM 输出端 15 = 0x0F 二进制正极开关, 含诊断 16 = 0x10 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]
	0x8	PWM 频率 OUT07	rw	UINT	100	20...250	OUT07 PWM 频率 [Hz]

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	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 [‰]
	0x5	PWM 抖动值 OUT04	rw	UINT	0	0...1 000	OUT04 PWM 抖动值 [‰]

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	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]
	0x3	时间基数 IN12	rw	UINT	50	0...2 000	IN14 时间基数 [ms]

索引	S-Idx	参数名称	数据类型		默认	详细信息	
	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
	0x4	PWM 输出端 OUT03	wo	UINT	--	PWM 输出端 OUT03 的值

索引	S-Idx	参数名称	数据类型		默认	详细信息
	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

输入端：PDO 映射 (I/O 模块)	346
输出端：PDO 映射 (I/O 模块)	348

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）

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	
系统标志 (ExB01 I/O 模块)	349

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 [346](#))

7.3.6 I/O 模块错误消息

内容

EMCY 对象.....	350
SDO 错误消息.....	351

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	nw	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

错误标志353

错误： CAN / CANopen.....353

19606

7.4.1 错误标志

19608

→ 章节 **系统标志** (→ 页 [268](#))

7.4.2 错误： CAN / CANopen

19610
19604

→ 系统手册“ecomatmobile 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 发送缓冲区超限	
30	81	11	--	--	--	--	--	CANx 保护/检测信号错误 (仅从站)	

EMCY 代码：I/O，系统

8412

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	I0 LSB	I0 MSB				输入端中断
08	21	03	I0 LSB	I0 MSB				输入端短路
10	21	03	I0 LSB	I0 MSB				超出电流信号 4...20 mA 范围
00	23	03	Q0 LSB	Q0 MSB				输出端中断
08	23	03	Q0 LSB	Q0 MSB				输出端短路
00	31	05						电源 VBBS
00	33	05						端子电压 VU
08	33	05						输出驱动电源 VBB1，VBB2
00	42	09						温度过高

在 CANopen 堆栈中，这些 EMCY 代码均无固定实现。 建议：

- 使用 CANOPEN_SENDEMCMYMESSAGE 产生这些 EMCY 代码。

制造商特定信息（详细信息）

24551

用于输入端和输出端（如可用）的制造商特定信息 EMCY 代码，按如下方式进行分布：

字节	BYTE 3							
位	7	6	5	4	3	2	1	0
I0 LSB	IN07	IN06	IN05	IN04	IN03	IN02	IN01	IN00
Q0 LSB	OUT07	OUT06	OUT05	OUT04	OUT03	OUT02	OUT01	OUT00

字节	BYTE 4							
位	7	6	5	4	3	2	1	0
I0 MSB	IN15	IN14	IN13	IN12	IN11	IN10	IN09	IN08
Q0 MSB	OUT15	OUT14	OUT13	OUT12	OUT11	OUT10	OUT09	OUT08

8 专业术语

B

Bootloader

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。

C

CAN

CAN = **控制器局域网**

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** = 发动机控制装置

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，该信息标明错误信息。

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或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

- 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 (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 [359](#))

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 OLE**

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 = **Protocol Data Unit** = 协议数据单元。

PDU 是→CAN 协议→SAE J1939 的一个术语。它指的是目标地址的组件 (PDU 格式 1, 连接导向) 或群组扩展 (PDU 格式 2, 消息导向)。

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 = 参数群组编号

PGN = 6 个零位 + 1 个预留位 + 1 个数据页位 + 8 位 PDU 格式 (PF) + 8 位 PDU 特定域 (PS)

参数组编号是→CAN 协议→SAE J1939 的一项术语。

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车载诊断实施指南
- 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

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

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。

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。

五划

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。

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。

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。

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。

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。

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）。

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。

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。

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。

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。

系统变量

可通过 IEC 地址或符号名称从 PLC 访问的变量。

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。

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。

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 [7](#)））

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。

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。

$1 \text{ MBd} = 1024 \times 1024 \text{ Bd} = 1\,048\,576 \text{ Bd}$

指定用途

根据用途说明所载的信息使用产品。

总线

同一电缆上多个参与者的串行数据传送。

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。

架构

系统硬件和/或软件的特定配置。

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。

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 [7](#)））

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。

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。

数据类型

可存储不同大小的值，具体数据类型而定。

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

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。

操作系统

装置中的基本程序，建立装置硬件和应用程序之间的连接。

→ 章节 **装置的软件模块** (→ 页 [35](#))

9 指数

B

Bootloader 35, 355

C

CAN 355

 接口和协议 33

 接口和协议：CR0133 中的 I/O 模块 290

 接口和协议：CR2532 中的 I/O 模块 290

CAN / CANopen

 错误和错误处理 266

CAN 声明（例如 CR1080） 54

CAN 堆栈 355

CAN 接口 33

CAN 接口 I/O 模块 289

CAN_ENABLE 78

CAN_RECOVER 80

CAN_REMOTE_REQUEST 110

CAN_REMOTE_RESPONSE 112

CAN_RX 86

CAN_RX_ENH 88

CAN_RX_ENH_FIFO 91

CAN_RX_RANGE 94

CAN_RX_RANGE_FIFO 97

CAN_SETDOWNLOADID 81

CAN_STATUS 83

CAN_TX 101

CAN_TX_ENH 103

CAN_TX_ENH_CYCLIC 106

CANOPEN_ENABLE 116

CANOPEN_GETBUFFERFLAGS 118

CANOPEN_GETEMCYMESSAGES 165

CANOPEN_GETERRORREGISTER 167

CANOPEN_GETGUARDHBERRLIST 160

CANOPEN_GETGUARDHBSTATSLV 162

CANOPEN_GETNMTSTATESLAVE 126

CANOPEN_GETODCHANGEDFLAG 132

CANOPEN_GETSTATE 120

CANOPEN_GETSYNCSTATE 155

CANOPEN_NMTSERVICES 128

CANOPEN_READOBJECTDICT 134

CANOPEN_SDOREAD 139

CANOPEN_SDOREADBLOCK 141

CANOPEN_SDOREADMULTI 144

CANOPEN_SDOWRITE 146

CANOPEN_SDOWRITEBLOCK 148

CANOPEN_SDOWRITEMULTI 151

CANOPEN_SENDEMCMYMESSAGE 169

CANOPEN_SETSTATE 123

CANOPEN_SETSYNCSTATE 157

CANOPEN_WRITEOBJECTDICT 136

CiA 355

CiA DS 304 355

CiA DS 401 355

CiA DS 402 355

CiA DS 403 356

CiA DS 404 356

CiA DS 405 356

CiA DS 406 356

CiA DS 407 356

COB ID 356

CODESYS 356

CODESYS 中的 FB、FUN、PRG 38

CODESYS 项目的编程说明 37

CSV 文件 357

CURRENT_CONTROL 230

D

DC 357

DLC 357

DRAM 357

DTC 357

E

E/A-Modul Ausgangsgruppe OUT0, OUT1 285

ECU 358

EDS 文件 358

EDS 文件中的数据类型 306

EMC 358

EMCY 358

EMCY 代码：CANx 353

EMCY 代码：I/O，系统 354

EMCY 对象 350

ERROR 状态 43

Ethernet 358

EUC 358

F

FASTCOUNT 217

FATAL ERROR 状态 43

FiFo 359

FLASH_INFO 239

FLASH_READ 241

FLASH-Speicher 16

FRAM 16, 359

G

GET_APP_INFO 243

GET_HW_INFO 245

GET_IDENTITY 247

GET_SW_INFO 248

GET_SW_VERSION 250

H

HMI 359

I

I/O 地址/变量	269
I/O 模块可能的工作模式	300
I/O 模块的运行	346
I/O 模块的配置	291
I/O 模块输入组 IN00...IN03	282
I/O 模块输入组 IN04...IN05	282
I/O 模块输入组 IN06...IN11	284
I/O 模块输入组 IN12...IN15	284
I/O 模块输入端和输出端的功能配置	294
I/O 模块输出组 OUT02...OUT07	287
I/O 模块输出组 OUT08...OUT09	288
I/O 模块输出组 OUT10...OUT11	288
I/O 模块输出组 OUT12...OUT15	288
I/O 模块错误消息	350
ID	359
IEC 61131	359
IEC 用户周期	359
IFM 功能元件	69
INC_ENCODER	219
INIT 状态 (复位)	41
INPUT	223
IP 地址	360
ISO 11898	360
ISO 11992	360
ISO 16845	360

J

J1939	360
J1939_DM1RX	207
J1939_DM1TX	210
J1939_DM1TX_CFG	213
J1939_DM3TX	215
J1939_ENABLE	173
J1939_GETDABYNAME	175
J1939_NAME	178
J1939_RX	190
J1939_RX_FIFO	192
J1939_RX_MULTI	194
J1939_SPEC_REQ	185
J1939_SPEC_REQ_MULTI	187
J1939_STATUS	182
J1939_TX	197
J1939_TX_ENH	199
J1939_TX_ENH_CYCLIC	201
J1939_TX_ENH_MULTI	203

L

LED	360
LSB	360

M

MAC-ID	361
MEM_ERROR	252
MEMCPY	254
MMI	361
MRAM	361
MSB	361

N

NMT	361
-----------	-----

O

Obj /对象	361
OBV	361
OHC	256
OPC	362
OUTPUT	233

P

Parameters of the outputs	150
PC 卡	362
PCMCIA 卡	362
PDM	362
PDO	362
PDU	362
PERIOD	226
PES	363
PGN	363
PID 控制器	363
PLC 配置	363
PWM	363
PWM 可用性	66, 298
PWM 输出端	65
PWM 输出端: I/O 模块 ExB01	298
PWM1000	236

R

RAW-CAN	364
ro	364
RTC	364
RUN 状态	42
rw	364

S

SAE J1939	364
SD 卡	365
SDO	365
SDO 错误消息	351
SET_IDENTITY	258
SET_LED	259
SET_PASSWORD	262

SRAM	16
STOP 状态	42

T

TCP	365
TIMER_READ_US	264

U

UDP	365
-----------	-----

一划

一般功能配置	55
--------------	----

二划

二进制输入端	19, 281
二进制输入端：配置和诊断（I/O 模块 ExB01）	295
二进制输出端：诊断（I/O 模块 ExB01）	297
二进制输出端：配置（I/O 模块 ExB01）	297
二进制输出端：配置和诊断	64
二进制输出端：配置和诊断（I/O 模块 ExB01）	296

三划

工作状态	41
已停止	365

四划

开关量输入：配置和诊断	60
比率式	366
手动设定编程系统	51
手动设定编程系统（I/O 模块）	291
从站	366

五划

功能元件：CANopen	115
功能元件：CANopen SDO	138
功能元件：CANopen SYNC	154
功能元件：CANopen 对象目录	131
功能元件：CANopen 网络管理	125
功能元件：CANopen 状态	115
功能元件：CANopen 保护	159
功能元件：CANopen 紧急	164
功能元件：RAW-CAN 远程	109
功能元件：RAW-CAN 状态	77
功能元件：RAW-CAN（第 2 层）	77

功能元件：SAE J1939	172
功能元件：SAE J1939 状态	172
功能元件：SAE J1939 诊断	206
功能元件：SAE J1939 请求	184
功能元件：处理输入值	216
功能元件：传送 RAW-CAN 数据	100
功能元件：传送 SAE J1939	196
功能元件：系统	238
功能元件：接收 RAW-CAN 数据	85
功能元件：接收 SAE J1939	189
功能元件：输出端功能	229
节点	366
节点保护	366
本文档的结构是怎样的？	8
本装置 CAN J1939 的限制	46
本装置 CAN 的限制	45
本装置 CANopen 的限制	45
可用内存	16
可能的输入端/输出端工作模式	272
目标	367
电子狗	367
电子狗状况	44
电阻测量	21, 283
主站	367
闪存	367
对象目录	367
对象目录可选对象（索引 0x1000...0x10FFF），详细信息	320
对象目录可选对象（索引 0x1000...0x1FFF），概述	308
对象目录可选对象（索引 0x1400...0x14FF），详细信息	322
对象目录可选对象（索引 0x1600...0x16FF），详细信息	324
对象目录可选对象（索引 0x1800...0x18FF），详细信息	327
对象目录可选对象（索引 0x1A00...0x1AFF），详细信息	333
对象目录制造商特定对象（索引 0x2000...0x6FFF），详细信息	336
对象目录制造商特定对象（索引 0x2000...0x6FFF），概述	317
对象目录参数表格，详细信息	319
对象目录参数表格，概述	305
对象目录强制对象（索引 0x1000...0x1FFF），详细信息	319
对象目录强制对象（索引 0x1000...0x1FFF），概述	307

六划

地址	367
地址分配和 I/O 工作模式	269

在应用程序中控制 LED	31
夹具 15	367
网络变量	68
自检	368
创建应用程序	40
关于本手册	5
关于配线的注意事项	30
关于装置的信息	13
关于装置编程的重要说明	39
关于簧片继电器的安全说明	30, 57
安全说明	11
设定目标	52
设定运行时系统	48
设定编程系统	51
设定编程系统 (I/O 模块)	291

七划

进程图像	368
运行	368
抖动频率	368
更新运行时系统	50
连接指令	368
连接集成 I/O 模块 ExB01 以作为 CANopen 从站	290
针对 PWM 功能的 FB	66
针对集成 ExB01 I/O 模块的系统标志	349
针对装置 CR2530 的 IFM 功能元件	75
针对装置 CR2530 的 IFM 库	69
作为二进制输入端使用	61
条件	15
系统变量	55, 368
系统标志	268
系统标志 (ExB01 I/O 模块)	349
系统说明 I/O 模块 ExB01	276
系统描述	13
状态 LED	31
状态 LED I/O 模块	278
库	36
库 ifm_CANopen_NT_Vxxyzz.LIB	72
库 ifm_CR2530_V03yzz.LIB	70
库 ifm_J1939_NT_Vxxyzz.LIB	73
库 ifm_RAWCan_NT_Vxxyzz.LIB	71
应用程序	36
应用程序软件	368

应用程序的分发	41
快速输入端	61
快速输入端： I/O 模块 ExB01	295
启用 PLC 配置	53
启用输入端诊断	60
诊断	265, 369
诊断：二进制输出端 (通过电压测量)	27, 287
诊断：二进制输出端 (通过电流和电压测量)	25, 286
诊断：过载	27, 287
诊断：过载 (通过电流测量)	25, 286
诊断：断线 (通过电压测量)	26, 27, 286, 287
诊断：短路 (通过电压测量)	26, 28, 286, 288
诊断和错误处理	265
附录	267

八划

软件说明	34
图标	369
制造商特定信息 (详细信息)	354
使用 PWM 或 CURRENT_CONTROL 时的反应	24
使用，指定	369
使用 ifm maintenance 工具	41
版权	5
周期时间	369
变量	67
注意事项：序列号	12
注意周期时间！	39
波特	369
定义：过载	23
定义：短路	23

九划

指定用途	369
故障	265
响应系统错误	266
响应错误消息的进程示例	266
重新安装运行时系统	48
保存保留变量	67
保留变量	67
总线	369
误用	370
说明 ... 78, 80, 81, 83, 86, 88, 91, 94, 97, 101, 103, 106, 110, 112, 116, 118, 120, 123, 126, 128, 132, 134, 136, 139, 141, 144, 146, 148, 151, 155, 157, 160, 162, 165, 167, 169, 173, 175, 178, 182, 185, 187, 190,	

192, 194, 197, 199, 201, 203, 207, 210, 213, 215, 217, 219, 223, 226, 230, 233, 236, 239, 241, 243, 245, 247, 248, 250, 252, 254, 256, 258, 259, 262, 264, 370	
说明沿革 (CR253n).....	10
架构.....	370

十划

根据输出端工作模式的反应.....	24
配置.....	47
配置针对 PWM 功能的输出端.....	299
配置集成 I/O 模块的输入端.....	294
配置集成 I/O 模块的输出端.....	296
配置输入端.....	57
配置输入端软件过滤器.....	59
配置输入端和输出端（默认设定）.....	56
配置输入端的软件过滤器（I/O 模块）.....	294
配置输出端.....	62
配置输出端软件过滤器.....	63
配置输出端的软件过滤器（I/O 模块）.....	296
原理方块图.....	15
请注意！.....	11
读回保留变量.....	68
通过 PWM 控制电流 (= PWMi).....	66, 299
通过模板设定编程系统.....	55
通过模板设定编程系统（I/O 模块）.....	293
预运行.....	370

十一划

接口说明.....	32
接口说明 I/O 模块.....	289
控制器的启动运行状况.....	12
检测信号.....	370
检验安装.....	50
符号.....	370
符号和格式是什么意思？.....	7

十二划

硬件说明.....	14
硬件说明 I/O 模块.....	276
硬件结构.....	14
硬件结构 I/O 模块.....	277
嵌入式软件.....	370
剩余.....	370
集成 I/O 模块 ExB01 输入端.....	279

集成 I/O 模块：说明.....	276
集成 I/O 模块的对象目录.....	305
集成 I/O 模块输出端 ExB01.....	285
集成内部 I/O 模块 ExB01.....	292
装置的软件模块.....	35
装置的性能极限.....	44

十三划

概述.....	300
概述：ecomatmobile 装置文档模块.....	6
概要.....	305
输入组 IN00...IN03.....	20
输入组 IN04...IN05.....	20
输入组 IN06...IN11.....	22
输入组 IN12...IN15.....	22
输入端（技术）.....	17
输入端：PDO 映射（I/O 模块）.....	346
输入端：工作模式.....	272
输入端：工作模式（I/O 模块）.....	302
输入端：地址和变量.....	270
输入端和输出端功能配置.....	56
输入端参数.. 78, 80, 81, 83, 86, 89, 92, 101, 104, 107, 110, 113, 116, 118, 120, 123, 126, 129, 132, 134, 136, 139, 142, 144, 147, 149, 152, 155, 157, 160, 162, 165, 167, 170, 173, 176, 180, 182, 185, 187, 190, 192, 194, 197, 199, 202, 204, 207, 211, 213, 215, 218, 221, 224, 227, 231, 234, 236, 239, 241, 243, 245, 247, 248, 250, 252, 255, 257, 258, 260, 262	
输入端参数.....	95
输入端参数.....	98
输出组 OUT0、OUT1.....	25
输出组 OUT02...OUT07.....	27
输出组 OUT08...OUT09.....	28
输出组 OUT10...OUT11.....	28
输出组 OUT12...OUT15.....	29
输出端（技术）.....	23
输出端：PDO 映射（I/O 模块）.....	348
输出端：工作模式.....	274
输出端：工作模式（I/O 模块）.....	303
输出端：地址和变量.....	271
输出端功能元件.....	75
输出端对过载或短路的反应.....	23
输出端诊断的配置.....	64
输出端的自我保护.....	24
输出端的保护功能.....	23

输出端参数.....	79, 82, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114, 117, 119, 121, 124, 127, 130, 133, 135, 137, 140, 143, 145, 147, 153, 156, 158, 161, 163, 166, 168, 171, 174, 177, 181, 183, 186, 188, 191, 193, 195, 198, 200, 202, 205, 208, 212, 214, 215, 218, 222, 225, 228, 232, 235, 237, 240, 242, 244, 246, 247, 249, 251, 253, 255, 257, 258, 261, 263, 264
错误： CAN / CANopen.....	353
错误表.....	353
错误标志	353
数据类型	371

十四划

模拟输入端	18, 280
模拟输入端： 配置和诊断.....	58
模拟输入端： 配置和诊断 (I/O 模块 ExB01)	294
模板	371
需要预先具备哪些知识？	12

十六划

操作系统	36, 371
------------	---------