



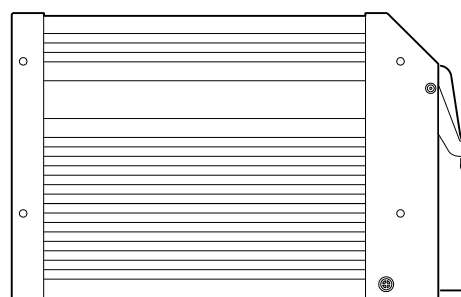
原版编程手册
ClassicController

CR0020

运行时系统 v06
CODESYS® v2.3

中文

7391027_03_CN 2017-07-21



内容

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	CODESYS 编程手册	7
1.4	符号和格式是什么意思？	7
1.5	本文档的结构是怎样的？	8
1.6	说明沿革 (CR0020、CR0505).....	9
2	安全说明	10
2.1	请注意！	10
2.2	需要预先具备哪些知识？	11
2.3	控制器的启动运行状况	12
2.4	注释： 序列号	12
2.5	注释： TEST 输入端	12
3	系统描述	13
3.1	关于装置的信息	13
3.2	硬件说明	13
3.2.1	硬件结构	14
3.2.2	延迟关闭的工作原理	18
3.2.3	继电器： 重要说明！	19
3.2.4	监控概念	20
3.2.5	输入端（技术）	24
3.2.6	输出端（技术）	31
3.2.7	关于配线的注意事项	40
3.2.8	关于簧片继电器的安全说明	40
3.2.9	双向输入端/输出端上的反馈	41
3.2.10	外部供给输出端反馈	43
3.2.11	状态 LED	45
3.3	接口说明	47
3.3.1	串行接口	47
3.3.2	CAN 接口	48
3.4	软件说明	49
3.4.1	装置的软件模块	50
3.4.2	CODESYS 项目的编程说明	53
3.4.3	工作状态	58

内容

3.4.4	工作模式	62
3.4.5	装置的性能极限	64
4	配置	66
4.1	设定运行时系统	66
4.1.1	重新安装运行时系统	67
4.1.2	更新运行时系统	68
4.1.3	检验安装	68
4.2	设定编程系统	69
4.2.1	手动设定编程系统	69
4.2.2	通过模板设定编程系统	72
4.3	一般功能配置	73
4.3.1	配置输入端和输出端（默认设定）	73
4.3.2	系统变量	73
4.4	输入端和输出端功能配置	74
4.4.1	配置输入端	75
4.4.2	配置输出端	79
4.5	变量	85
4.5.1	保持变量	86
4.5.2	网络变量	86
5	IFM 功能元件	87
5.1	针对装置 CR0020 的 IFM 库	87
5.1.1	库 ifm_CR0020_V06yyzz.LIB	88
5.1.2	库 ifm_CR0020_CANopenMaster_V04yyynn.LIB	91
5.1.3	库 ifm_CR0020_CANopenSlave_V04yyynn.LIB	91
5.1.4	库 ifm_CAN1_EXT_Vxxyyzz.LIB	92
5.1.5	库 ifm_J1939_x_Vxxyyzz.LIB	92
5.1.6	库 ifm_hydraulic_16bitOS05_Vxxyyzz.LIB	93
5.2	针对装置 CR0020 的 IFM 功能元件	94
5.2.1	功能元件：CAN 第 2 层	94
5.2.2	功能元件：CANopen 主站	116
5.2.3	功能元件：CANopen 从站	127
5.2.4	功能元件：CANopen SDO	137
5.2.5	功能元件：SAE J1939	142
5.2.6	功能元件：串行接口	160
5.2.7	功能元件：通过处理中断优化 PLC 周期	167
5.2.8	功能元件：处理输入值	173
5.2.9	功能元件：调整模拟值	180
5.2.10	功能元件：针对频率和周期测量的计数器功能	183
5.2.11	功能元件：PWM 功能	198

内容

5.2.12	功能元件： 液压控制.....	212
5.2.13	功能元件： 控制器.....	230
5.2.14	功能元件： 软件复位.....	241
5.2.15	功能元件： 时间测量/设定.....	243
5.2.16	功能元件： 保存、读取和转换内存中的数据.....	246
5.2.17	功能元件： 数据访问和数据检查.....	257
6	诊断和错误处理	265
6.1	诊断.....	265
6.2	故障.....	265
6.3	发生错误时的反应.....	266
6.4	继电器： 重要说明！.....	266
6.5	响应系统错误.....	267
6.6	CAN / CANopen: 错误和错误处理.....	267
7	附录	268
7.1	系统标志.....	268
7.1.1	系统标志： CAN.....	269
7.1.2	系统标志： SAE J1939.....	269
7.1.3	系统标志： 错误标志（标准侧）.....	270
7.1.4	系统标志： 状态 LED（标准侧）.....	271
7.1.5	系统标志： 电压（标准侧）.....	272
7.1.6	系统标志： 输入端 16...40 和输出端 24...0（标准侧）.....	273
7.2	地址分配和 I/O 工作模式.....	274
7.2.1	输入端/输入端地址分配.....	274
7.2.2	可能的输入端/输出端工作模式.....	278
7.2.3	I/O 地址/变量.....	282
7.3	错误表.....	285
7.3.1	错误标志.....	285
7.3.2	错误： CAN / CANopen.....	285
8	Glossary of Terms	287
9	索引	305
10	备注	311
11	ifm weltweit • ifm worldwide • ifm à l'échelle internationale	315

1 关于本手册

内容	
版权	5
概述： ecomatmobile 装置文档模块	6
CODESYS 编程手册	7
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 (CR0020、CR0505)	9

202

1.1 版权

6088

© ifm electronic gmbh 保留所有权利。 未经 ifm electronic gmbh 的同意，不得复制和使用本手册的任何部分。

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

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

1.2 概述：ecomatmobile 装置文档模块

17405

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] > CR0020 > [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 CODESYS 编程手册

17542

在额外的“CODESYS V2.3 编程手册”中，您将可获得更多关于使用编程系统的详情。

本手册可在 **IFM** 网站免费下载：

→ www.ifm.com > [Systems for mobile machines]

您还可通过以下方式获得 **ecomatmobile** 手册和在线帮助：

→ **ecomatmobile** DVD“软件、工具和文档”

1.4 符号和格式是什么意思？

203

以下符号或图表利用图片阐明我们的说明中的注释。

 警告	
可能导致死亡或严重的不可逆伤害。	
 小心	
可能导致轻微的可逆伤害。	
通知	
预计或可能会导致财产损失。	
	关于故障或干扰的重要说明
	其他备注
	操作请求
	反应，结果
	“查看”
abc	交叉引用
123	十进制数
0x123	十六进制数
0b010	二进制数
[...]	按键、按钮或指示标记

1.5 本文档的结构是怎样的？

204
1508

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

本手册的使用方式：

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

若发生故障或有相关疑问，请与制造商联系：

→ www.ifm.com

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

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

→ www.ifm.com > 选择国家 > [技术资料查询] > (货号) > [操作使用说明书]

1.6 说明沿革 (CR0020、CR0505)

9185

本手册更改了哪些内容？概述

日期	主题	更改：
2010-09-09	PID2 (FB)	修正了输入端参数
2010-11-10	终端电阻器	主题 1244 修正
2011-02-14	TIMER_READ_US (FB)	修正了最大计数值转换
2011-04-05	内存 POU、FRAMREAD、FRAMWRITE、FLASHREAD、FLASHWRITE	参数 SRC、LEN、DST 的允许值
2011-04-13	CANopen 概述	新：附录 CANopen 表格
2012-01-09	内存模块 FRAMREAD、FRAMWRITE	更换了表格“允许值”中的参数 SRC、DST
2012-07-16	运行时系统	升级至 v06
2012-10-04	各式各样	修正
2013-06-24	各种	新文档结构
2014-04-28	各种功能块	更准确地说明功能块输入端 CHANNEL
2014-06-30	文档名称	“系统手册”重新命名为“编程手册”
2014-07-31	FB PHASE	修正了输入端参数 C、ET 说明
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
2015-01-13	针对错误代码、系统标志的文档结构	• 错误标志： 现在仅位于附录，章节系统标志 • CAN / CANopen 错误和错误处理： 现在仅位于系统手册“技巧” • 错误代码、EMCY 代码： 现在位于附录，章节错误表
2015-03-10	可用内存	完善了说明
2015-05-22	FB INPUT_ANALOG、INPUT_CURRENT、INPUT_VOLTAGE	允许输入通道
2015-05-26	FB J1939_x_GLOBAL_REQUEST	更准确的说明
2015-06-10	各种功能块	修正了 FB 输入 CHANNEL 说明
2016-04-27	快速输入端的 FB	添加了频率较高时的注意事项

2 安全说明

内容	
请注意！	10
需要预先具备哪些知识？	11
控制器的启动运行状况	12
注释： 序列号	12
注释： TEST 输入端	12

213

2.1 请注意！

214
11212

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

- ▶ 机器/设备的制造商应负责确保机器/设备的安全。
- ▶ 遵守机器/装置投放至市场所在国家的国内和国际法规。

 **警告**

未遵守这些说明可能会导致财产损失或人身伤害。
ifm electronic gmbh 不承担此方面的任何责任。

- ▶ 对此装置和使用其执行工作前，操作者必须已阅读并理解安全说明和本手册中的相应章节。
- ▶ 操作者必须获得对该机器/设备执行工作的授权。
- ▶ 操作者必须已获得执行此工作所需的资质和培训。
- ▶ 遵照装置的技术资料操作！
您可在该处的 **IFM** 主页上找到当前的技术资料：
→ www.ifm.com > 选择您所在国家 > [技术资料查询] > (货号。) > [PDF 技术资料]
- ▶ 注意安装和配线信息，以及装置的功能和特性！
→ 随附的安装说明或在 **IFM** 的主页上：
→ www.ifm.com > 选择您所在国家 > [技术资料查询] > (货号。) > [操作使用说明书]
- ▶ 请注意 **IFM** 网站上提供的现有文档发行说明中的修正内容和注意事项：
→ www.ifm.com > 选择您所在国家 > [技术资料查询] > (货号。) > [操作使用说明书]

通知

串行接口的驱动器模块可能会受损！

运行时断开或连接串行接口可能导致不明确的状态并损坏驱动器模块。

▶ 切勿在运行时断开或连接串行接口。

2.2 需要预先具备哪些知识？

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

若要对 PLC 进行编程，用户还应熟悉 CODESYS 软件。

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

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

请遵守安全说明。

2.3 控制器的启动运行状况

6827
15233
11575

警告

由于机器或设备部分意外和危险启动导致的危险！

- ▶ 创建程序时，程序员必须确保发生故障（例如紧急停机）和随后执行故障排除后，机器或设备部分不会发生任何意外和危险启动！
 - ⇒ 实现重新启动抑制。
- ▶ 发生错误时，在程序中将相关的输出功能设为“错误”！

诸如以下情况可能会导致重新启动：

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

若要确保控制器的安全运行状况：

- ▶ 在应用程序中监控电压供应器。
- ▶ 若发生错误，在应用程序中关闭所有相关的输出功能。
- ▶ 在应用程序（反馈）中监控可能会导致危险移动的执行器。
- ▶ 在应用程序（反馈）中监控可能会导致危险移动的继电器触点。
- ▶ 如有必要，在应用程序中确保焊接的继电器触点无法触发或继续危险移动。

2.4 注释：序列号

20780

- ▶ 在用户的工厂，制作一幅机器控制器网络图。将安装的每一个控制器的序列号输入至网络图。
- ▶ 在下载软件组件之前，读出该序列号并检查网络图，以确保您访问的是正确的控制器。

2.5 注释：TEST 输入端

20781

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

3 系统描述

内容	
关于装置的信息	13
硬件说明	13
接口说明	46
软件说明	49

975

3.1 关于装置的信息

11568

本手册讲述针对 ifm electronic gmbh 移动机器的 ecomatmobile 系列：
ClassicController: CR0020, CR0505 , AI 硬件版本或更高
运行时系统（目标）v06b

3.2 硬件说明

内容	
硬件结构	14
延迟关闭的工作原理	18
继电器：重要说明！	19
监控概念	20
输入端（技术）	24
输出端（技术）	31
关于配线的注意事项	40
关于簧片继电器的安全说明	40
双向输入端/输出端上的反馈	41
外部供给输出端反馈	43
状态 LED	45

14081

3.2.1 硬件结构

内容	
启动条件	14
继电器	14
原理方块图	15
可用内存	16

15332

启动条件

19658

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

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

继电器

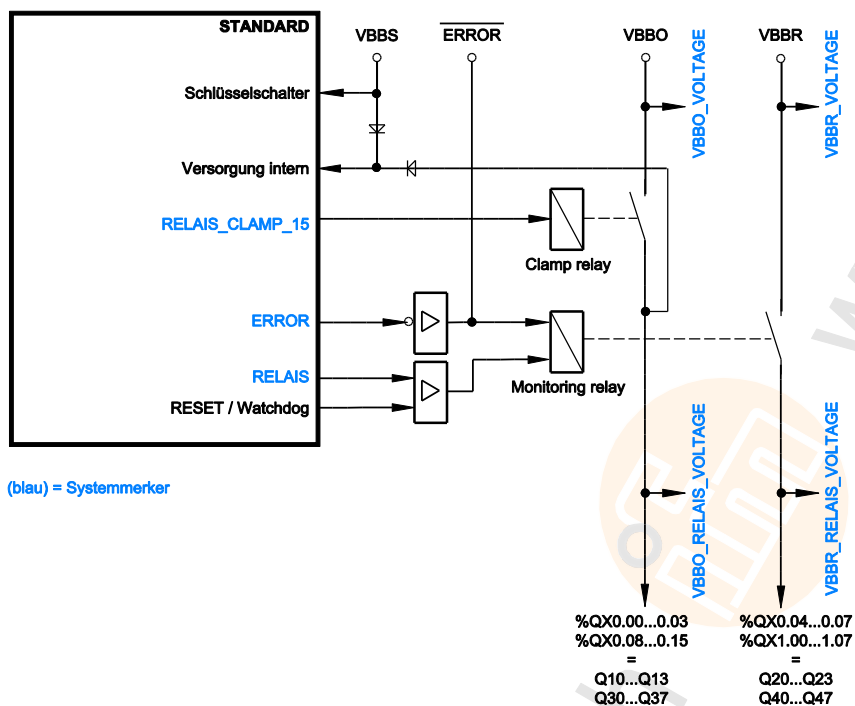
19659

控制器有 2 个内部输出继电器, 可各自将 12 个输出端与端子电压 VBBx (x = o / r) 隔开。
输出继电器 (监控继电器) 通过两个通道由微控制器触发。为此, 一个通道通过固态开关, 由电子狗信号的 AND 功能 (内部微控制器监控) 和系统标志位 RELAIS 触发。另一个通道通过固态开关, 仅由系统标志位 ERROR 触发。开启时, 端子电压 VBBx 通过继电器触点 (非正极引导) 提供给输出端。
夹具继电器通过系统标志 RELAIS_CLAMP_15 利用一个通道启用 (→ 图)。
控制器启动后 RELAIS_CLAMP_15 启用。
夹具继电器切换第二哥输出组的 VBB0 电压。
夹具继电器确保装置内部供电, 只要继续提供 VBB0, 即便 VBBs 被有意或无意中断。
夹具继电器完全在应用程序中控制, 且可通过系统标志 RELAIS_CLAMP_15 的设定/复位命令切换。

原理方块图

19660

下方方块图显示继电器与应用信号的关联以及系统标志的逻辑状态。



图：电源和继电器原理方块图

可用内存

13736

闪存

15366

FLASH 内存 (非易失慢速内存) 装置现有全部	2 M 字节
--------------------------------	--------

由此，如下内存区域将预留用于 ...

应用程序最大大小	704K 字节
除应用程序之外的数据 用户可写入文件、位图、字体等数据	1M 字节
除应用程序之外的数据 数据读取可通过 FLASHREAD (→ 页 251)数据写入可通过 FLASHWRITE (→ 页 252) (文件：就标头而言少 128 字节	64K 字节

剩余的内存保留用于系统内部用途。

SRAM

15906

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。	512K 字节
--	---------

由此，如下内存区域将预留用于 ...

应用程序保留的数据	160K 字节
-----------	---------

剩余的内存保留用于系统内部用途。

FRAM

8002

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。	32K 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	1K 字节
作为剩余定义标志（始于 %MB0...） ► 通过以下 FB 设定内存区域的末尾： MEMORY_RETAIN_PARAM (→ 页 248)!	256 字节
用户可免费使用剩余内存 访问可通过 FRAMREAD (→ 页 254)和 FRAMWRITE (→ 页 255)	16K 字节

剩余的内存保留用于系统内部用途。

3.2.2 延迟关闭的工作原理

993

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

如果未作请求，则必须通过程序关闭控制器。关闭点火开关有助于保存内存状态等。

ClassicController 可利用相应的电源电压输入端连接以及相关系统标志的评估，通过程序来关闭。本章方块图 **硬件结构**（→ 页 14）显示了单个电流回路的来龙去脉。

将端子 VBBs (23) 连接至点火开关。

994

控制器通过端子 23 供给，且可通过点火开关关闭。

在汽车工程中，电势被称为“夹具 15”。

该端子为内部监控。如果不提供电源电压，系统标志 CLAMP_15 则设为 FALSE。标志 CLAMP_15 的复位可通过应用程序监控。

将端子 VBB0 (5) 连接至电池（未开启）。

995

最多有 12 个输出组 VBB0 输出端可通过端子 5 供给。同时，亦可通过该端子锁存控制电子组件。

锁存

996

如果电压施加至 VBB0 并设定系统标志 RELAY_CLAMP_15（以及继电器 [Clamp]），则锁存启用。

如果系统标志 RELAY_CLAMP_15 复位，则 [Clamp] 断电。如果此时无电压施加至端子 23，则移除锁存且控制器完全关闭。

3.2.3 继电器：重要说明！

19480

分配继电器 - 电位：→ 技术资料

单位继电器触点（= 单位输出组）的最大总电流：→ 技术资料

通知

可能会损坏继电器触点！

在紧急情况下，“粘附”继电器触点不再断开输出端的电源！

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

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

► 如果 VBBx 永久连接至电源：

- 另永久连接 VBBO 并
- 在 VBBS（夹具 15）的帮助下通过程序关闭输出端。

3.2.4 监控概念

内容

监控概念的工作原理	21
-----------------	----

991

控制器监控电源电压和系统错误标志。视状态而定...

- 控制器关闭内部继电器
 - > 输出端断电，但保留逻辑状态

或：

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

监控概念的工作原理

997

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

如果在程序处理期间出现了错误，则可利用系统标志位 ERROR 关闭继电器，以断开关键的设备段。复位系统标志位 RELAIS（通过系统标志位 ERROR 或直接复位）即可关闭所有输出端。电流回路 VBBr 中的输出端通过输出端继电器直接断开。因此，电流回路 VBBo 中的输出端仅通过软件断开。

11575

 **警告**

由于机器或设备部分意外和危险启动导致的危险！

- ▶ 创建程序时，程序员必须确保发生故障（例如紧急停机）和随后执行故障排除后，机器或设备部分不会发生任何意外和危险启动！
 - ⇒ 实现重新启动抑制。
- ▶ 发生错误时，在程序中将相关的输出功能设为“错误”！

 如果发生电子狗错误，则程序处理自动中断，控制器复位。而后控制器在通电后再次启动。

监控电源电压 VBBr

20109

通过电势 VBBr，可供给高达 12 个输出组输出端。端子电压受监控：

ERROR_VBBR = TRUE	电源电压缺失或太低
ERROR_VBBR = FALSE	电源电压正常

- ▶ 该信息将在应用程序中加以处理！

监控和保护装置

内容	
施加电源电压后	22
如果运行时系统/应用程序正在运行	22
如果 TEST 插脚未启用	23
一次性机制	23

3926

就这些装置而言，将自动执行以下监控活动：

施加电源电压后

3927

施加电源电压后（控制器在 Bootloader 中），装置将执行以下测试：

- > RAM（测试一次）
- > 电源电压
- > 系统数据一致性
- > Bootloader 的 CRC
- > 如果存在且已启动：运行时系统的 CRC
- > 如果存在且已启动：应用程序的 CRC
- > 内存错误：
 - 如果测试正在运行：标志 ERROR_MEMORY = TRUE（可从第一个周期开始评估）。
 - 如果测试未运行：红色 LED 亮。

如果运行时系统/应用程序正在运行

3928

则循环执行以下测试：

- > 触发电子狗 (100 ms)
而后持续利用程序检查电子狗
- > 持续温度检查
发生故障时：系统标志 ERROR_TEMPERATURE = TRUE
- > 持续电压监控
发生故障时：系统标志 ERROR_POWER = TRUE 或 ERROR_VBBR = TRUE
- > 持续 CAN 总线监控
- > 持续系统数据监控：
 - 加载的程序

系统描述

硬件说明

- RUN / STOP 工作模式，
- 加载的运行时系统。
- 节点 ID，
- CAN 和 RS232 波特率。
- > 在 RUN 工作模式中：
 - 循环 I/O 诊断：
 - 短路，
 - 断线，
 - 输入端和输出端过载（电流）
 - 交叉故障（仅针对 SafetyController）。

如果 TEST 插脚未启用

3929

- > FRAM ¹⁾ 系统数据写入保护，例如：
 - 加载的运行时系统，
 - 校准数据。
 通过软件和硬件执行。
 - > 应用程序（闪存中）写入保护
 - > DEBUG 模式
- ¹⁾ FRAM 在此表示所有类型的非易失快速内存。

一次性机制

3930

- > 下载或上传期间的 CRC 监控。
- > 必须核查运行时系统和应用程序是否分配至同一装置

3.2.5 输入端 (技术)

内容	
模拟输入端.....	24
二进制输入端	25
输入组 I00...I07 / ANALOG0...7	26
输入组 I10...I17 / FRQ0...FRQ3	28
输入组 I20...I27	29
输入组 I30...I37	30
输入组 group I40...I47	30

14090

模拟输入端

2426

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：

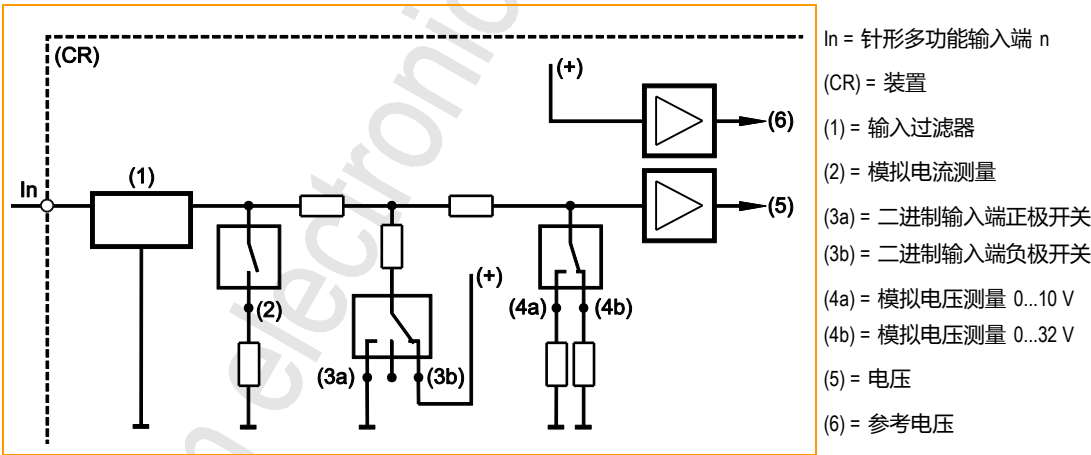
- 电流输入 0...20 mA
- 电压输入 0...10 V
- 电压输入 0...32 V

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

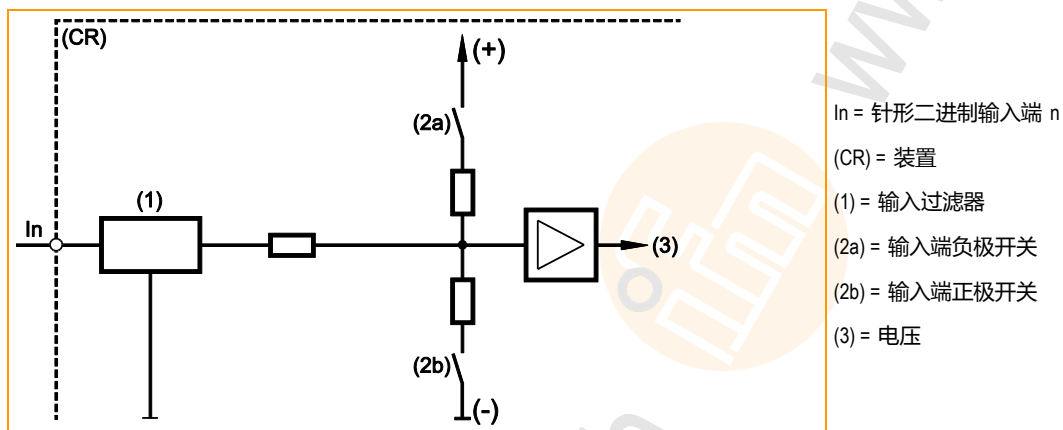
二进制输入端

1015
7345

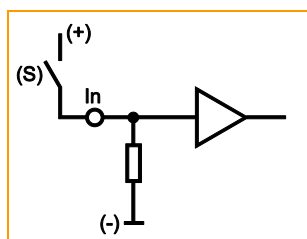
二进制输入端可在以下模式下运行：

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

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



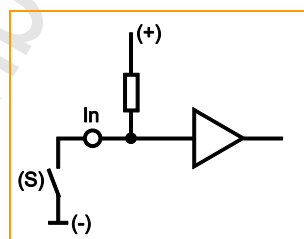
图：针对负极性和正极性传感器信号的二进制输入端负极开关/正极开关基本电路



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 I00...I07 / ANALOG0...7

20389

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）

→ 章节 **可能的输入端/输出端工作模式** (→ 页 278)

可评估包含诊断功能的 NAMUR 传感器。

所有输入端显示关于功能和诊断的相同状况。

 详细说明 → 章节 **输入端/输入端地址分配** (→ 页 274)

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

如果针对电流测量配置模拟输入端，则输入端工作模式切换至安全电压测量范围 (0...30V DC)，超过终值时 (> 21 mA)，在标志字节 ERROR_IO 中设定相应的错误位。如果值再次低于限值，输入端会自动切回至电流测量范围。

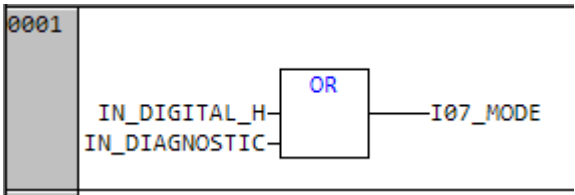
► 每个输入端均可通过应用程序进行配置：

- FB **INPUT_ANALOG** (→ 页 175) > 输入端 MODE
- 配置字节 Ixx_MODE

15380

配置字节 Ixx_MODE 示例：

分配操作将所选输出端设为工作模式
IN_DIGITAL_H（含诊断）。



> 诊断结果可通过以下系统标志显示：

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

非电子开关二进制信号 NAMUR 诊断：
► 为开关配置额外的电阻器连接件！

图：输入端 Inn 非电子开关 S

输入组 I10...I17 / FRQ0...FRQ3

19487

输入端 I10...13

19490

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
- 输出端 (→ 章节 **输出端 (技术)** (→ 页 [31](#)))

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [278](#))

这些输入端不可配置。

输入端 I14...17 / FRQ0...3

19497

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 [278](#))

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：

- 配置字节 Ixx_MODE
- 包含以下 FB 的快速输入端：

FAST_COUNT (→ 页 184)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 186)	测量到达所选通道的信号的频率。
INC_ENCODER (→ 页 188)	针对编码器评估的递增/递减计数器功能
PERIOD (→ 页 192)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 194)	测量所示周期内所示通道频率和以 [μs] 为单位的周期（周期时间）。此外，传号空号比以 [%] 为单位显示。
PHASE (→ 页 196)	读取包含快速输入端的通道对并对比信号的相位

输入组 I20...I27

19489

Inputs I20...23

19499

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
- 输出端 (→ 章节 **输出端 (技术)** (→ 页 [31](#)))

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [278](#))

这些输入端不可配置。

输入端 I24...27 / CYL0...3

19500

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 [278](#))

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：

- 配置字节 Ixx_MODE
- 包含以下 FB 的快速输入端：

FAST_COUNT (→ 页 184)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 186)	测量到达所选通道的信号的频率。
INC_ENCODER (→ 页 188)	针对编码器评估的递增/递减计数器功能
PERIOD (→ 页 192)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 194)	测量所示周期内所示通道频率和以 [μs] 为单位的周期（周期时间）。此外，传号空号比以 [%] 为单位显示。
PHASE (→ 页 196)	读取包含快速输入端的通道对并对比信号的相位

输入组 I30...I37

19501

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）
- 二进制输入端负极开关 (BH)，针对负极性传感器信号
- 输出端（→ 章节 **输出端（技术）**（→ 页 [31](#)）

→ 章节 **可能的输入端/输出端工作模式**（→ 页 [278](#)）

可评估包含诊断功能的 NAMUR 传感器。

所有输入端显示关于功能和诊断的相同状况。

 详细说明 → 章节 **输入端/输入端地址分配**（→ 页 [274](#)）

► 每个输入端均可通过应用程序进行配置：

- 配置字节 Ixx_MODE

输入组 group I40...I47

19502

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）
- 输出端（→ 章节 **输出端（技术）**（→ 页 [31](#)）

→ 章节 **可能的输入端/输出端工作模式**（→ 页 [278](#)）

可评估包含诊断功能的 NAMUR 传感器。

所有输入端显示关于功能和诊断的相同状况。

 详细说明 → 章节 **输入端/输入端地址分配**（→ 页 [274](#)）

► 每个输入端均可通过应用程序进行配置：

- 配置字节 Ixx_MODE

3.2.6 输出端 (技术)

内容	
二进制输出端	31
PWM 输出端	31
输出组 Q10...Q13 / Q20...Q23	32
输出组 Q30...Q37	35
输出组 Q40...Q47	37

14093

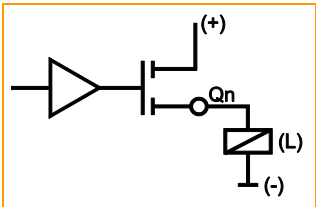
二进制输出端

14094

以下工作模式适用于装置输出端 (→ 技术资料) :

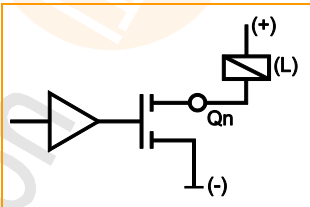
- 二进制输出端, 正极开关 (BH), 含/不含诊断功能
- 二进制输出端, 负极开关 (BL), 不含诊断功能

15450



Qn = 插脚输出端 n
(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号



Qn = 插脚输出端 n
(L) = 负载

输出端负极开关 (BL) 基本电路
针对负极性输出端信号

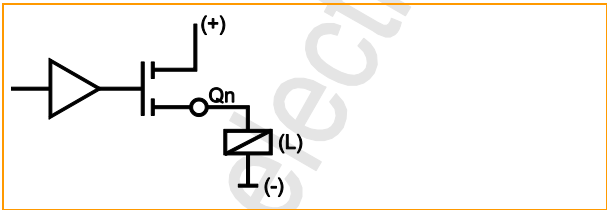
PWM 输出端

14095

以下工作模式适用于装置输出端 (→ 技术资料) :

- PWM 输出端, 正极开关 (BH), 不含诊断功能

15451



Qn = 插脚输出端 n
(L) = 负载

输出端正极开关 (BH) 基本电路
针对正极性输出端信号

输出组 Q10...Q13 / Q20...Q23

19507

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
- 二进制输入端 (→ 章节 **输入端 (技术)** (→ 页 [24](#)))

→ 章节 **可能的输入端/输出端工作模式** (→ 页 [278](#))

如果不作为 PWM 输出端，则可通过集成电流测量通道进行诊断，该通道还可用于电流控制输出功能。

► 每个输出端均可通过应用程序进行配置：

表示负载电流→ FB **OUTPUT_CURRENT** (→ 页 [201](#))

PWM 输出端：→ FB **PWM1000** (→ 页 [210](#))

配置字节 Qxx_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。

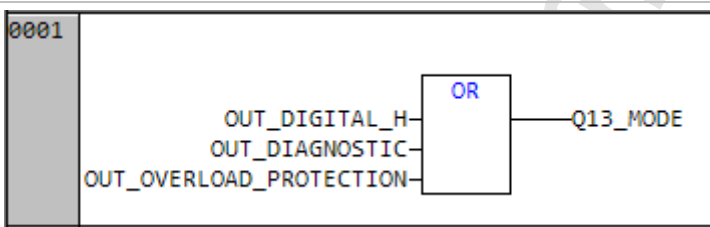
 PWM 模式下的输出端不支持诊断功能。

当作二进制输出端时，可利用系统变量 Q1x_MODE...Q2x_MODE 进行配置。如果要使用诊断，则需额外启用。

输出端信号断线和短路信号分别通过 ERROR_BREAK_Q1Q2 或 ERROR_SHORT_Q1Q2 表示。如有必要，可在应用程序中隐藏单个输出单错误位。

例如：

分配操作将所选输出端设为工作模式 OUT_DIGITAL_H（含诊断）。过载保护已启用（预设）。



❗注意

若为了保护内部测量电阻器，OUT_OVERLOAD_PROTECTION 则应始终保持启用状态（最大测量电流 4.1 A）。

❗ 限制值请务必确保遵循技术资料！

纯粹的 PWM 模式不支持功能 OUT_OVERLOAD_PROTECTION。

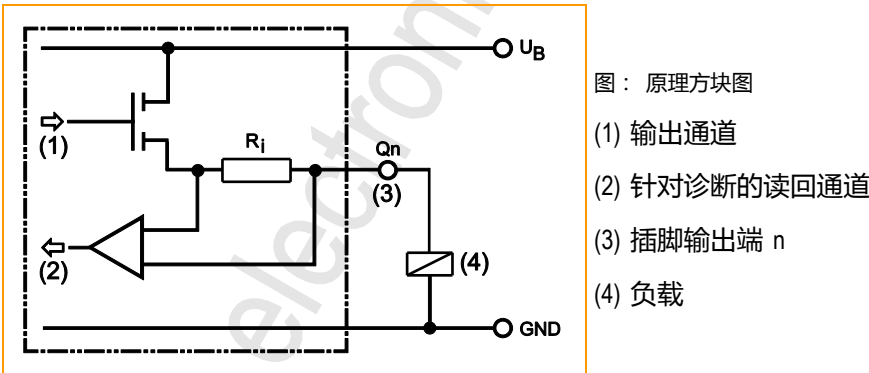
3976

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

诊断：二进制输出端（通过电流测量）

19398
19396

这些输出端的诊断通过输出端的内部电流测量进行：



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。

系统描述

硬件说明

过载即...

“超过最大额定电流的 12.5 %”。

诊断：断线（通过电流测量）

19400

断线检测通过输出端内部的读回通道完成。

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

诊断：短路（通过电流测量）

19401

断线检测通过输出端内部的读回通道完成。

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

输出组 Q30...Q37

19511

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
- 二进制输入端 (→ 章节 **输入端 (技术)** (→ 页 24))

→ 章节 **可能的输入端/输出端工作模式** (→ 页 278)

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。

当作为二进制输出端时,利用系统变量 Qxx_MODE 配置每个输出端。如果要使用诊断，则需额外启用。

注意

为保护内部测量电阻器，OUT_OVERLOAD_PROTECTION 应始终保持启用状态（默认设定）。视所选电流测量范围而定，保护从 2.25 A 或 4.5 A 开始提供。PWM 模式**不支持**该功能，如有必要，可禁用。

 限制值请务必确保遵循技术资料！

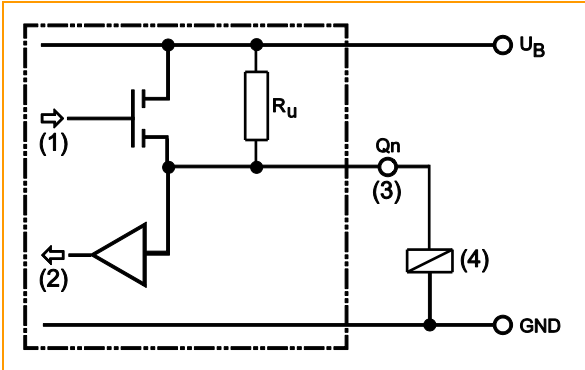
视环境温度而定，不可能通过短路电流确切检测到短路，因为输出端驱动器暂时禁用，以防止受损。

输出端**开启**时，断线和短路检测启用。

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)

输出组 Q40...Q47

19513

这些输出端是一组多功能通道。

这些输出端提供多个功能选项（每个输出端可单独配置）：

- 二进制输出端，正极开关 (BH)，部分还有负极开关 (BL)
- 包含脉冲宽度调制 (PWM) 的模拟输出端（部分作为 H 桥）
- 二进制输入端（→ 章节 **输入端 (技术)**（→ 页 [24](#)）

→ 章节 **可能的输入端/输出端工作模式**（→ 页 [278](#)）

► 每个输出端均可通过应用程序进行配置：

表示负载电流 → FB **OUTPUT_CURRENT**（→ 页 [201](#)）

PWM 输出端：→ FB **PWM1000**（→ 页 [210](#)）

配置字节 Qxx_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。

 PWM 模式下的输出端不支持诊断功能。

当作二进制输出端时,利用系统变量 Qxx_MODE 配置每个输出端。 如果要使用诊断, 则需额外启用。 输出端信号断线和短路信号分别通过 ERROR_BREAK_Qx 或 ERROR_SHORT_Qx 表示 (按照输出组结合)。 如有必要, 可在应用程序中隐藏单个输出单错误位。

❗注意

为保护内部测量电阻器, OUT_OVERLOAD_PROTECTION 应始终保持启用状态 (默认设定)。 视所选电流测量范围而定, 保护从 2.25 A 或 4.5 A 开始提供。 PWM 模式**不支持**该功能, 如有必要, 可禁用。

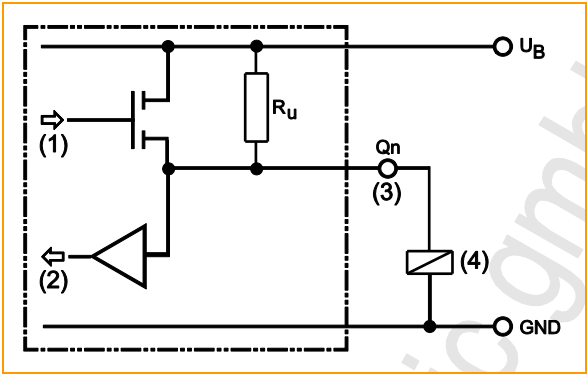
❗ 限制值请务必确保遵循技术资料 !

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

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 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)

3.2.7 关于配线的注意事项

1426

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

单个缩写的含义如下：

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

输入端/输出端通道分配：→ 产品目录、安装说明或技术资料

3.2.8 关于簧片继电器的安全说明

7348

若使用非电子开关，则应注意以下几点：

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

- **补救措施：** 安装针对簧片继电器的串联电阻器：

串联电阻器 = 簧片继电器的最大输入电压/允许电流

例如：32 V / 500 mA = 64 Ohm

- 串联电阻器不得超过装置输入端输入电阻 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

3.2.9 双向输入端/输出端上的反馈

999

有些控制器端子可配置为输入端或输出端 (→ 技术资料)。

通知

如有不允许的反馈，则会损坏输出端！

如果双向输入端/输出端群组同时跟输入端和输出端一起运行，则该输出组的电势 VBB 不可变为无电势。

在以下情况下输出组无电势：

- RELAIS = FALSE 或
- RELAIS_CLAMP_15 = FALSE。

该无电势状态的后果是：如果是在一个输入/输出组中，则电压通过输出端晶体管防护二极管反馈：

- 输入端 (如 I1) = TRUE 且
- 相同组的输出端 (如 Q2) = TRUE。

> 后果：

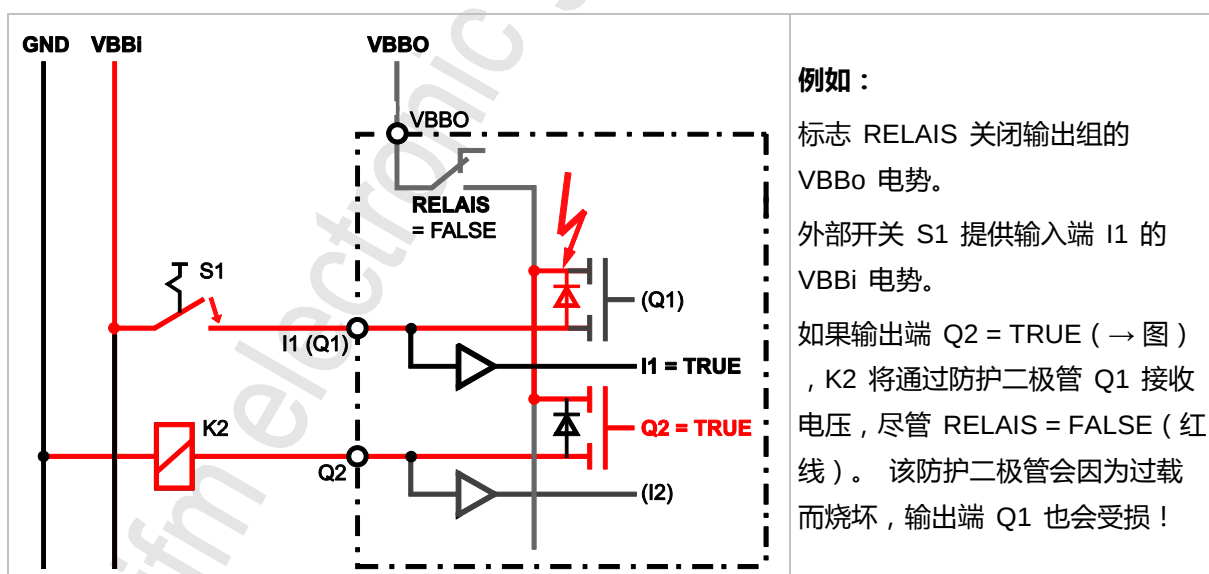
输出端 (Q2) 上的负载通过输入端 (I1) 防护二极管接收电压。当时反馈电流流经的防护二极管以及输出端 (Q1) 会受损。

► 补救措施：

将输入/输出组仅作为输入端或仅作为输出端运行。

或：

遵循以下说明。



图：不允许连接的示例：有反馈危险！



针对组合运行双向输入端/输出端的帮助：

- ▶ 在应用程序中将标志 RELAIS 和/或 RELAIS_CLAMP_15 永久设为 TRUE：
 - TRUE ----- RELAIS
 - TRUE ----- RELAIS_CLAMP_15

3.2.10 外部供给输出端反馈

2422

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

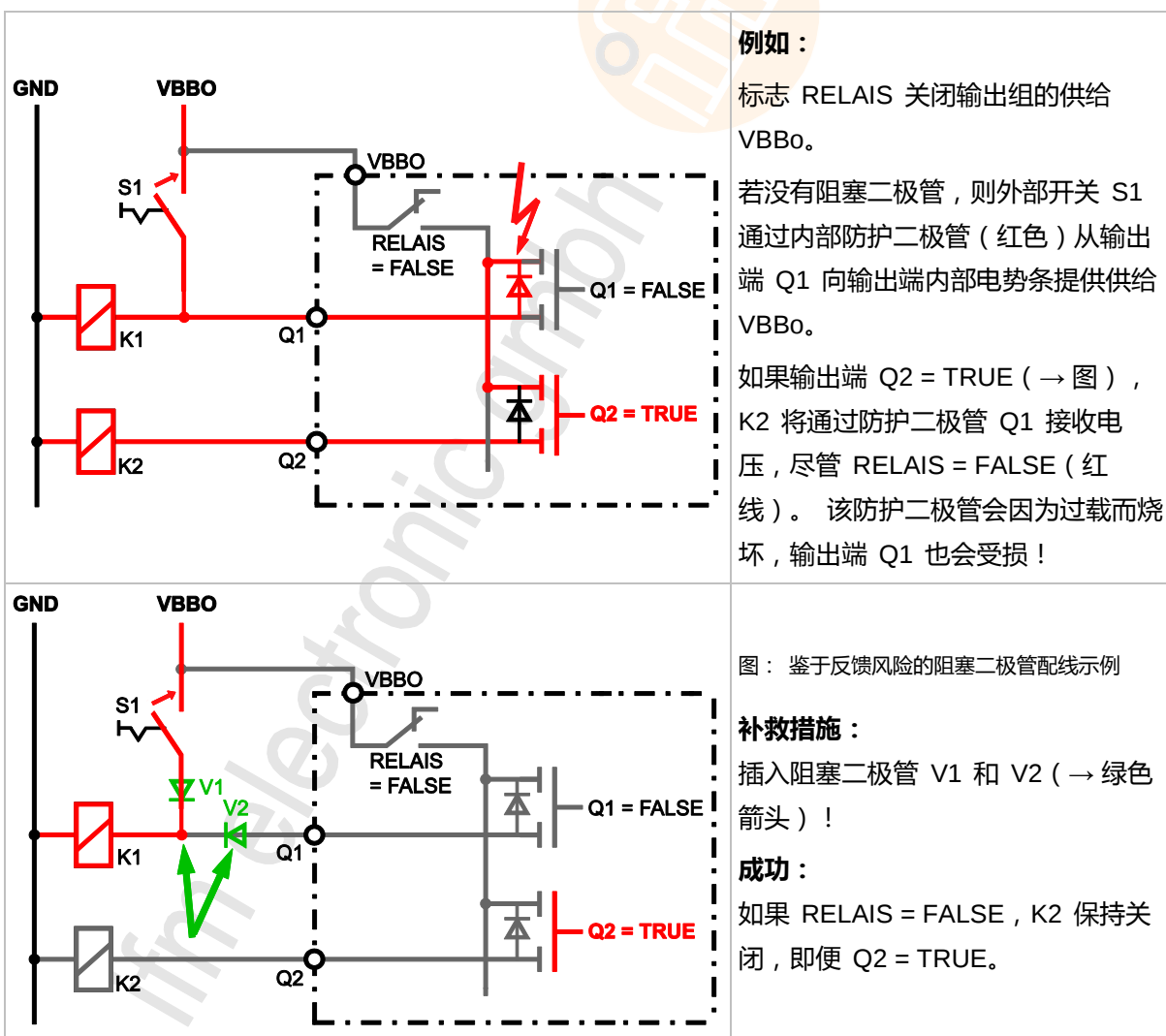
通知

如有不允许的反馈，则会损坏输出端！

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

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

► 通过阻塞二极管保护外部供给的输出端！



ⓘ 注意

针对外部供给输出端的帮助

- ▶ 外部供给输出端必须通过二极管解耦，以便不会有外部电压施加至输入端端子。

3.2.11 状态 LED

1430

集成状态 LED 指示工作状态（默认设定）。

LED 颜色	闪烁频率	说明
关闭	永久关闭	无工作电压
黄色	短暂亮起	初始化或复位检查
绿色/黑色	5 Hz	未加载运行时系统
绿色/黑色	2 Hz	应用程序 RUN
绿色	永久亮起	应用程序 STOP
红色/黑色	2 Hz	应用程序 RUN，但有错误
红色	短暂亮起	严重错误
红色	永久亮起	严重错误（如输入端 TEST = 启用） ERROR STOP / STYSTEM STOP

工作状态 STOP 和 RUN 可通过编程系统更改。

在应用程序中控制 LED

9989

就该装置而言，还可通过应用程序设定状态 LED。为此，需使用以下系统变量（→ **系统标志**（→ 页 268））:

LED	“启用”的 LED 颜色（针对“亮起”）
LED_X	“暂停”的 LED 颜色（针对“熄灭”或其他颜色）
---	来自数据结构 "LED_COLOR" 的颜色常量。允许的条目： LED_GREEN LED_BLUE LED_RED LED_WHITE LED_MAGENTA LED_CYAN LED_YELLOW LED_ORANGE LED_BLACK (= LED 熄灭)
LED_MODE	来自数据结构 "LED_MODES" 的闪烁频率。允许的条目： LED_2HZ LED_1HZ LED_05HZ (= 0.5 Hz) LED_0HZ (常量)

ⓘ注意

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

3.3 接口说明

内容	
串行接口	47
CAN 接口	48

14098

3.3.1 串行接口

14099

本装置包含串行接口。

串行接口一般可搭配以下功能：

- 程序下载
- 调试
- 应用程序的自由使用

12998

❗注意

在默认情况下，串行接口对用户不可用，因为它用于程序下载和调试。

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

连接和数据 → 技术资料

3.3.2 CAN 接口

内容	
CAN: 接口和协议	48
连接和数据 → 技术资料	14101

CAN: 接口和协议

19523
14587

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

- 第 2 层：第 2 级上的 CAN → 章节 **功能元件：CAN 第 2 层** (→ 页 [94](#))
- CANopen 主站 (→ 章节 **功能元件：CANopen 主站** (→ 页 [116](#)))
- CANopen 从站 (→ 章节 **功能元件：CANopen 从站** (→ 页 [127](#)))
- CANopen 网络变量 (通过 CODESYS)
- SAE J1939 (针对驱动管理，→ 章节 **功能元件：SAE J1939** (→ 页 [142](#)))
- 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

11796

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	---		
	SAE J1939	SAE J1939		

标准波特率 = 125 Kbits/s

3.4 软件说明

内容	
装置的软件模块	50
CODESYS 项目的编程说明	53
工作状态	58
工作模式	62
装置的性能极限	64

14107

3.4.1 装置的软件模块

内容	
Bootloader	51
运行时系统.....	51
应用程序	51
库	52

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 [70](#))

我们将在下文将是该软件模块：

Bootloader

14111

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

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

Bootloader 包含基本例程...

- 针对硬件模块之间的通信，
- 针对操作系统的重新加载。

Bootloader 是要保存在装置上的第一个软件模块。

运行时系统

14112

装置中的基本程序，建立装置硬件和应用程序之间的连接。

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

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

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

:

→ www.ifm.com

应用程序

14118

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

8340

警告

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

库

19527

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。例如：

库	使用
ifm_CR0020_Vxxyzz.LIB	装置特定库 必须始终包含于应用程序！
ifm_CR0020_CANopenMaster_Vxxyzz.LIB	(可选) 如果装置 CAN 接口 1 作为 CANopen 主站运行
ifm_CR0020_CANopenSlave_Vxxyzz.LIB	(可选) 如果装置 CAN 接口 1 作为 CANopen 从站运行
ifm_CAN1_EXT_Vxxyzz.LIB	(可选) 如果装置 CAN 接口 1 按照 29 位运行
ifm_CR0020_J1939_x_Vxxyzz.LIB x = 1...2 = CAN 接口编号	(可选) 如果装置 CAN 接口为了与发动机控制器通信

Details: → chapter **针对装置 CR0020 的 IFM 库** (→ 页 [87](#))

3.4.2 CODESYS 项目的编程说明

内容	
CODESYS 中的 FB、FUN、PRG	54
应用程序中的计算和转换	55
注意周期时间！	55
创建应用程序	56
保存启动项目	57
使用 IFM 下载程序	57
使用 IFM 维护工具	57

7426

您可在此了解如何对装置进行编程。

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



CODESYS 中的 FB、FUN、PRG

8473

在 CODESYS 中，我们区分以下类型的功能元件：

FB = 功能块

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

FUN = 功能

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

PRG = 程序

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

ⓘ 注意

不得在功能中调用功能块！

否则：执行期间，应用程序将崩溃。

不得递归或间接调用所有功能元件！

IEC 应用程序最多可包含 8,000 个功能元件！

背景

功能的所有变量...

- 调用时初始化，且
- 返回至调用方后无效。

功能块有 2 种调用：

- 初始化调用以及
- 操作实际调用。

因此，这意味着功能中的 FB 调用。

- 每次有一次额外的初始化调用，且
- 上次调用的数据丢失。

应用程序中的计算和转换

20779

⚠注意

如果应用程序中需要以下元件：

- 数学功能（如 ATAN），
- 计算，
- 转换（如 REAL_TO_BYTE），

则以下适用于相应运算符输入端和输出端的值：

- ▶ 严格遵守每种情况下的允许值范围！
- > 否则控制器可能会出现 FPU 错误。

例如：

20777

目标格式的最大表示值会被超越。

例如：

REAL_TO_INT (12345678.3)
> INT 限于 -32768...+32767（仅整数）

20778

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

注意周期时间！

8006

就 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

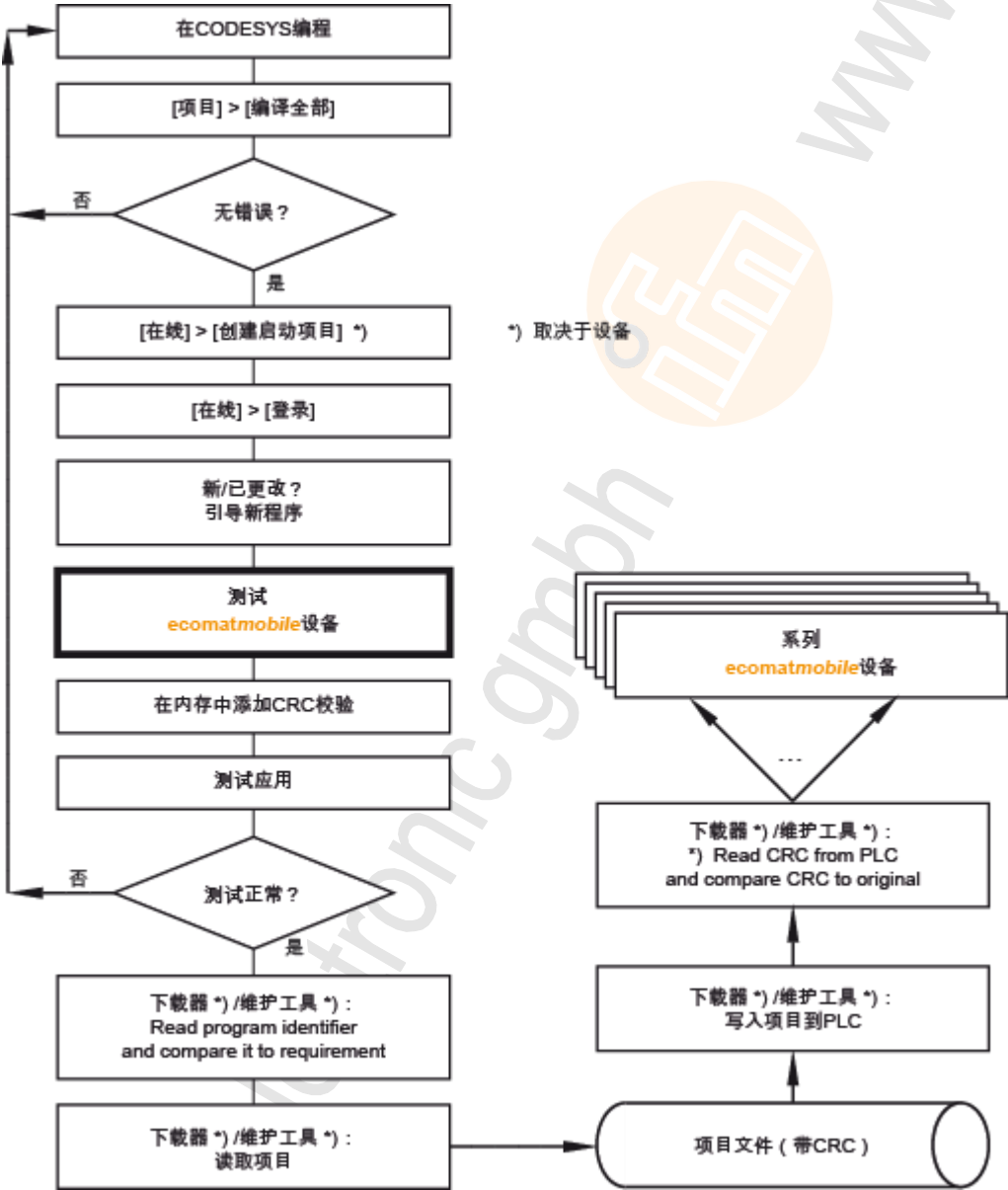
装置有运行过慢的风险！
周期时间不得太长！

- ▶ 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。

创建应用程序

8007

应用程序由 CODESYS 编程系统生成，并在程序开发期间多次加载至控制器，以便进行测试：
在 CODESYS 中：[Online] > [Login] > 加载新的程序。
就每一次经由 CODESYS 的下载而言，会再次转化源代码。结果是，每次控制器内存将形成新的校验和。该进程还适用于安全控制器，直至发布软件。



图：创建和分发软件

保存启动项目

7430

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

❗ 注意

注意：启动项目稍大于实际程序。

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

- ▶ CODESYS 菜单 [Online] > [Create boot project]
每次更改后，该操作是必要的。
- > 重启后，装置将以最后保存的启动项目启动。
- > 如果未保存启动项目：
 - 重启后装置保持 STOP 运行状态。
 - 应用程序不（再）可用。
 - LED 呈绿色亮起。

使用 IFM 下载程序

8008

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

您可在此了解当前的 **IFM** 下载程序（最低 V06.18.26）：

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

使用 IFM 维护工具

8492

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

您可在此了解当前的 **IFM** 维护工具：

→ www.ifm.com > [Systems for mobile machines]

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

3.4.3 工作状态

内容	
工作状态：运行时系统不可用	58
工作状态：应用程序不可用.....	59
工作状态：应用程序可用	60
Bootloader 状态	61
INIT 状态（复位）	61
STOP 状态.....	61
RUN 状态.....	61
SYSTEM STOP 状态	62

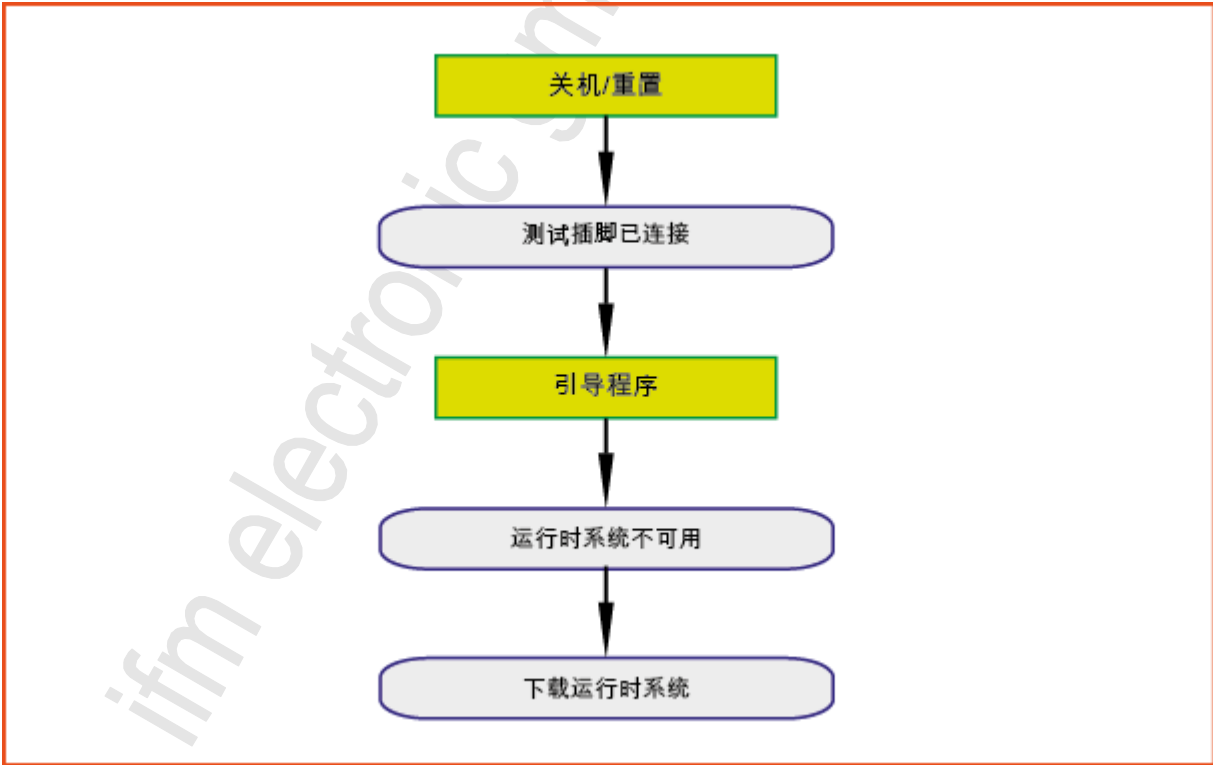
14120

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 （ERROR STOP 之后）

工作状态：运行时系统不可用

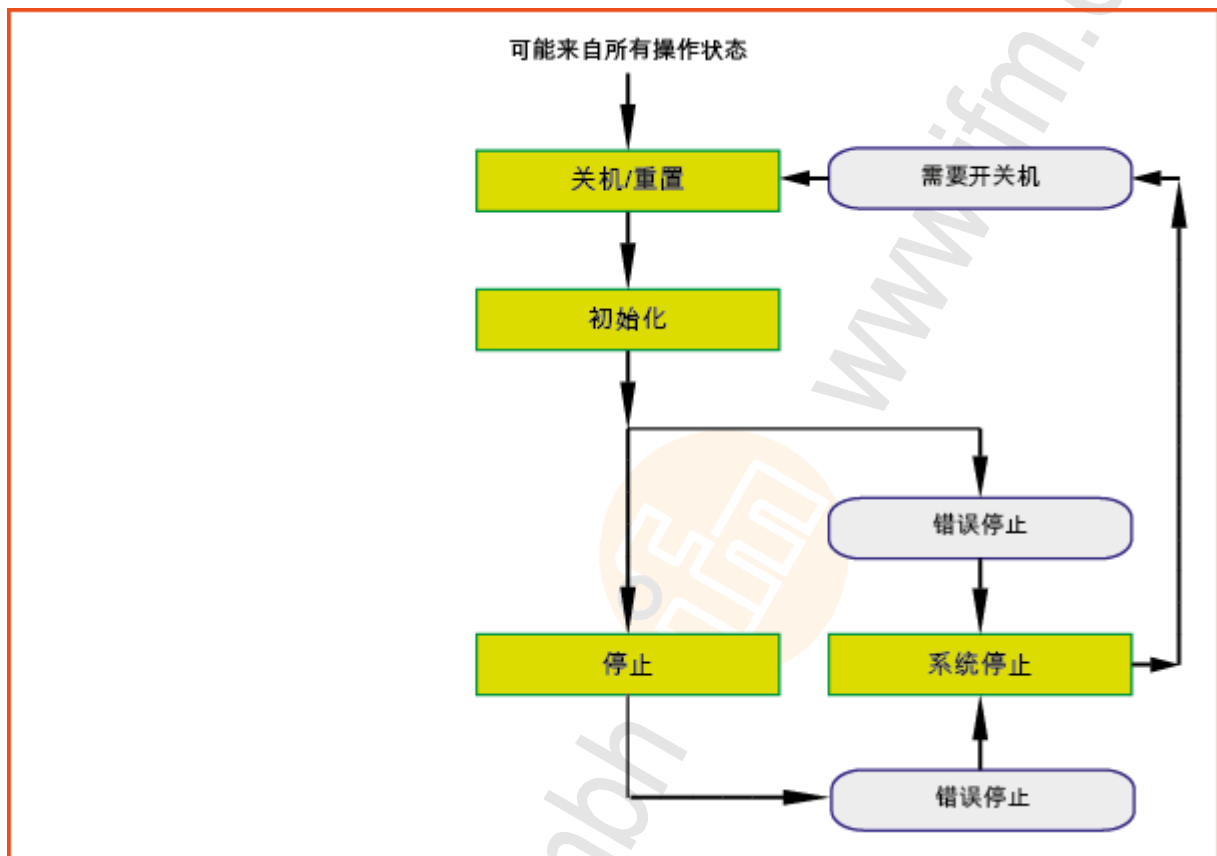
19217



图：工作状态（此时：运行时系统不可用）

工作状态：应用程序不可用

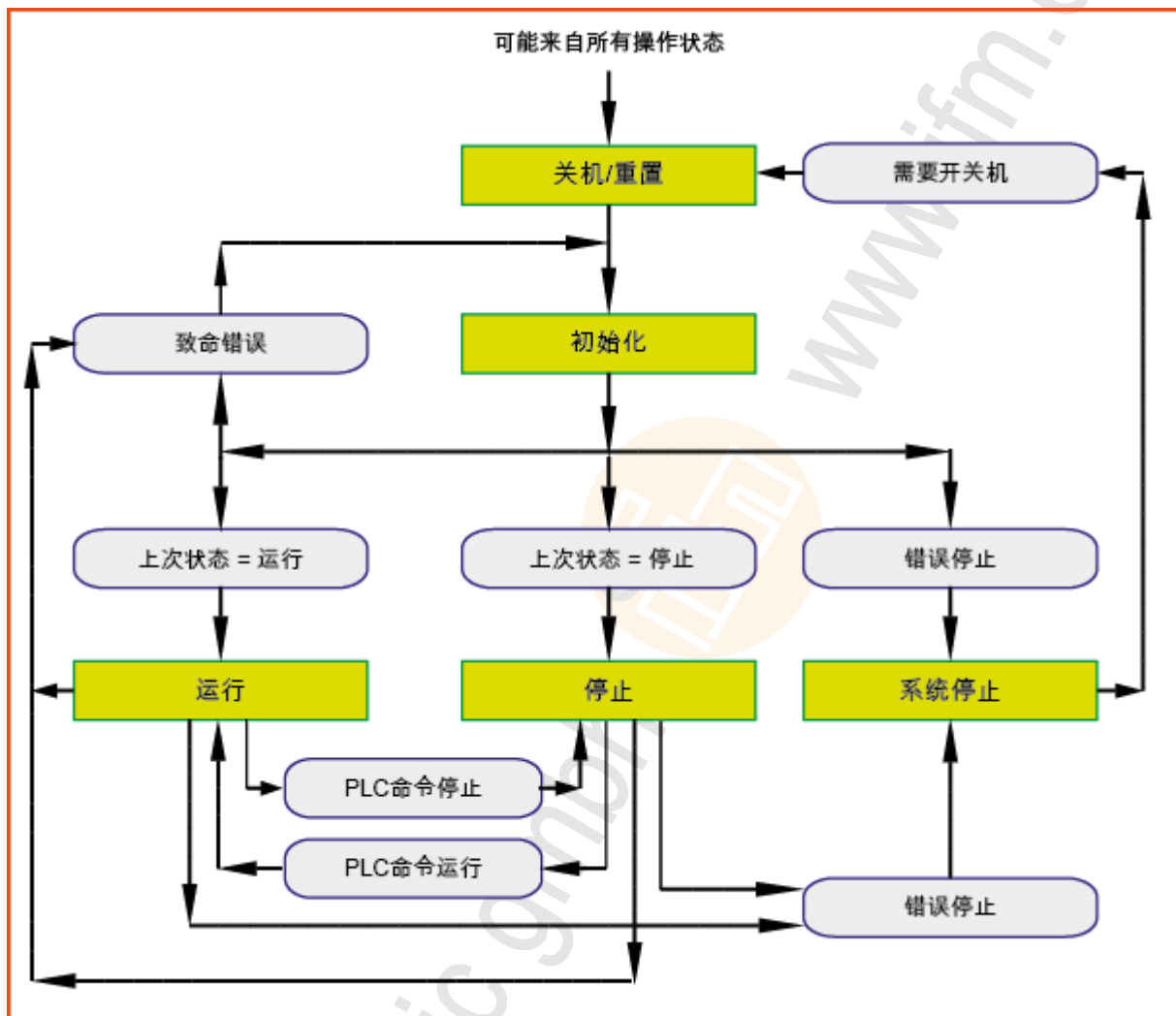
19218



图：工作状态（此时：应用程序不可用）

工作状态：应用程序可用

19219



图：工作状态（此时：应用程序可用）

Bootloader 状态

1080

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

- > LED 闪烁绿色 (5 Hz)。

INIT 状态 (复位)

1076

前提：安装有效的运行时系统。

每次通电复位后均会经历该状态。

- > 运行时系统初始化。
- > 展开各种检查，如等待正确的电源电压。
- > RUN 或 STOP 状态代替该温度状态。
- > LED 呈黄色亮起。

该状态可转换至以下状态之一：

- RUN
- STOP

STOP 状态

1078

在以下情况下可达到该状态：

- 从 RESET 状态转换，前提是：
 - 未加载程序，或
 - RESET 之前的最后状态为 STOP 状态
- 通过 STOP 命令从 RUN 状态转换
 - 仅针对工作模式 = TEST (→ 章节 **TEST 模式** (→ 页 62))
- > LED 呈绿色亮起。

RUN 状态

1077

在以下情况下可达到该状态：

- 从 RESET 状态转换，前提是：
 - RESET 之前的最后状态为 RUN 状态
- 通过 RUN 命令从 STOP 状态转换
 - 仅针对工作模式 = TEST (→ 章节 **TEST 模式** (→ 页 62))
- > LED 闪烁绿色 (2 Hz)。

SYSTEM STOP 状态

19222

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

- > LED 呈红色亮起。

3.4.4 工作模式

1083

ecomatmobile 控制器可在不同模式下运行, 不受工作状态影响。

TEST 模式

1084

通知

存储的软件可能丢失！
在测试模式中, 存储的运行时系统和应用程序软件没有保护。

14892

ⓘ 注意

- > 仅在您连接 OPC 客户端后, 将 TEST 接头连接至电源电压！

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

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

软件仅可在 TEST 模式中下载至控制器。

可通过标志 TEST 查询应用程序的状态。

ⓘ 摘要 测试输入端已启用：

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

注释：TEST 输入端

20781

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

SERIAL_MODE

1085

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

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

→ 章节 **功能元件：串行接口** (→ 页 [160](#))

DEBUG 模式

1086

如果输入端 DEBUG **SET_DEBUG** (→ 页 [262](#)) 设为 TRUE，编程系统或下载程序等则可与控制器通信并执行某些特别的系统命令（如针对服务功能，通过 GSM 调制解调器 CANremote）。

在该工作模式中，不可进行软件下载，因为测试输入端（→ 章节“**TEST 模式** (→ 页 [62](#))”）未连接至电源电压。

3.4.5 装置的性能极限

7358

 注意装置的性能极限！ → 技术资料

高于平均的压力

1488
5023

以下 FB 等利用高于平均水平的系统资源：

功能块	高于平均负载
CYCLE, PERIOD, PERIOD_RATIO, PHASE	使用多个输入频率高的测量通道
OUTPUT_CURRENT_CONTROL, OCC_TASK	同时使用多个电流控制器
CAN 接口	高波特率 (> 250 kbits)，高总线负载
PWM, PWM1000	同时使用多个 PWM 通道。尤其是 4 号以上的通道，时间更为关键。
INC_ENCODER	同时使用多个编码器通道

上述 FB 等可触发系统中断。这意味着每次启用可延长应用程序的周期时间。

以下指示应视作参考值。

FB 使用的限制

1489

电流控制器	最高 8	如有可能，切勿使用任何其他影响性能的功能！
CYCLE, PERIOD, PERIOD_RATIO, PHASE	1 个通道	输入频率 ≤ 10 kHz
	4 个通道	输入频率 ≤ 2 kHz
INC_ENCODER	最高 4	如有可能，切勿使用任何其他影响性能的功能！

1509

通知

控制器有运行过慢的风险！ 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。

电子狗状况

11786

在本装置中，电子狗监控 CODESYS 应用程序的程序运行时。

如果超过最长电子狗时间（大约 100 ms）：

> 装置复位并重启。

这样您可读取标志 LAST_RESET。



4 配置

内容	
设定运行时系统	66
设定编程系统	69
一般功能配置	73
输入端和输出端功能配置	74
变量	85

1016

相应安装说明或本文档 **附录** (→ 页 [268](#))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。

4.1 设定运行时系统

内容	
重新安装运行时系统	67
更新运行时系统	68
检验安装	68

14091

4.1.1 重新安装运行时系统

14092
2733

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

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

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

→ www.ifm.com

2689

ⓘ 注意

必须始终使用适于所选目标的软件版本：

- 运行时系统 (ifm_CR0020_Vxxyyzz.H86),
- PLC 配置 (ifm_CR0020_Vxx.CFG),
- 装置库 (ifm_CR0020_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 "CR0020"）和软件版本号 "xx"（例如 "02"）必须始终是相同的值！否则装置会进入“停止”模式。

"yy"（版本号）和 "zz"（修补号）的值**不必**一致。

4368

ⓘ 还必须加载以下文件：

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

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

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

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

4.1.2 更新运行时系统

13269

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

14158

通知

数据有丢失的风险！
删除或升级运行时系统时，装置上的所有数据和程序将被删除。
▶ 先保存所有所需数据和程序，再删除或升级运行时系统！

就该操作而言，可遵循跟之前章节“重新安装运行时系统”相同的说明。

4.1.3 检验安装

14512
14406

- ▶ 将运行时系统加载至控制器之后：
 - 检查运行时系统是否正确传送！
 - 检查控制器是否加载适当的运行时系统！
- ▶ 第 1 次检查：
利用 **IFM** 下载程序或维护工具核实是否加载正确的运行时系统版本：
 - 读取装置中运行时系统的名称、版本和 CRC ！
 - 手动对比该信息和目标数据！
- ▶ 第 2 次检查（可选）：
在应用程序中核实是否加载适当的运行时系统版本：
 - 读取装置中运行时系统的名称和版本！
 - 对比该数据和特定值！以下 FB 用于读取数据：

GET_IDENTITY (→ 页 260)	读取存储在装置中的特定标识： • 装置的硬件名称和硬件版本 • 装置运行时系统的名称 • 装置运行时系统的版本和版本号 • 应用程序的名称（之前已通过以下方式保存：SET_IDENTITY (→ 页 263))
-------------------------------	--

- ▶ 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.....	69
通过模板设定编程系统.....	72

3968

4.2.1 手动设定编程系统

内容	
设定目标	70
启用 PLC 配置 (如 CR0033	71

3963

设定目标

2687
11379

在 CODESYS 中创建新项目时，必须加载与装置相符的目标文件。

- ▶ 在菜单 [Configuration] 对话框 [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_CR0020_Vxxyzz.LIB 并手动操作！

启用 PLC 配置 (如 CR0033)

15824

在配置编程系统期间 (→ 上一部分), PLC 配置自动执行。

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

- ▶ 单击 CODESYS 中的选项卡 [Resources] :



- ▶ 双击左列的 [PLC Configuration]。

- > 当前 PLC 配置显示屏 (→ 下图) :



用户可根据配置在程序环境中找到以下内容 :

- 所有重要系统和错误标志
根据应用程序和程序, 这些标志必须经过处理和评估。 可通过符号名称访问。

- 输入端和输出端的结构

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



4.2.2 通过模板设定编程系统

13745

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

970

-  安装 **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 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。

4.3.2 系统变量

2252
13519
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。

- > 若要表示和处理电子狗错误或重新启动系统的原因，则设定系统变量 LAST_RESET。
- > 通过模式类型表示所选 I/O 配置

4.4 输入端和输出端功能配置

内容	
配置输入端.....	75
配置输出端.....	79

1375
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_CR0020_Vxxyyzz.LIB) 已预定义常量（如 IN_DIGITAL_H）。
详细信息 → 可能的输入端/输出端工作模式（→ 页 278）。
- 在模板中找到针对每个控制器的程序块，重启控制器后程序块在第一个周期调用。 在此编程的网络仅用于将定义的配置分配给输入端和输出端。

4.4.1 配置输入端

内容	
关于簧片继电器的安全说明	75
二进制输入端	76
快速输入端	77
模拟输入端	78

19567
3973

有效的工作模式 → 章节 可能的输入端/输出端工作模式 (→ 页 278)

- ▶ 每个输入端均可通过应用程序进行配置：
 - FB INPUT_ANALOG (→ 页 175) > 输入端 MODE
 - 配置字节 lxx_MODE

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

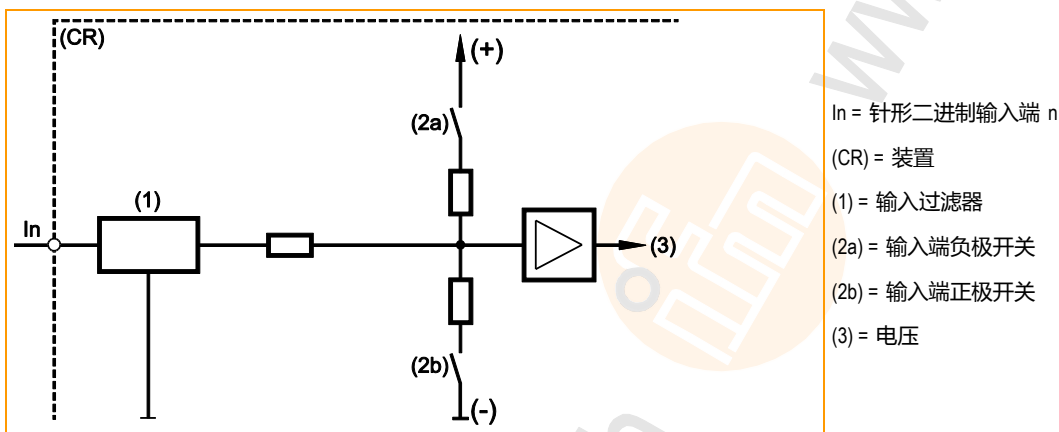
二进制输入端

1015
7345

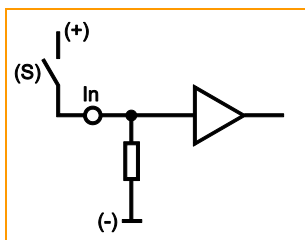
二进制输入端可在以下模式下运行：

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

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



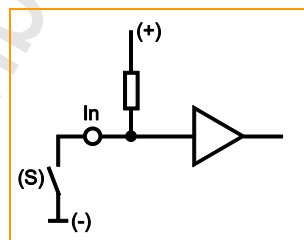
图：针对负极性和正极性传感器信号的二进制输入端负极开关/正极开关基本电路



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。

快速输入端

1018

此外，**ecomatmobile** 控制器有高达 16 个快速计数/脉冲式输入端，输入频率高达 50 kHz（→ 技术资料）。例如，如果机械开关连接至这些输入端，则控制器中可能因为触点弹跳而出现故障信号。如有必要，必须利用应用程序软件过滤这些“故障信号”。

另外，还应注意脉冲输入端是否针对频率测量 (FRQx) 和/或周期测量 (CYLx) 设计（→数据资料）。

适当的功能块包括：

在 FRQx 输入端上： FAST_COUNT (→ 页 184)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 186)	测量到达所选通道的信号的频率。
PERIOD (→ 页 192)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 194)	测量所示周期内所示通道频率和以 [μs] 为单位的周期（周期时间）。此外，传号空号比以 [%] 为单位显示。
PHASE (→ 页 196)	读取包含快速输入端的通道对并对比信号的相位

 使用这些单元时，参数化输入端和输出端自动配置，因此程序员无需进行此操作。

模拟输入端

1369

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

如果在工作模式 "0...32 V" 中读回电源电压，则还可进行比率测量。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。而后电源电压的波动不会对该测量值产生影响。

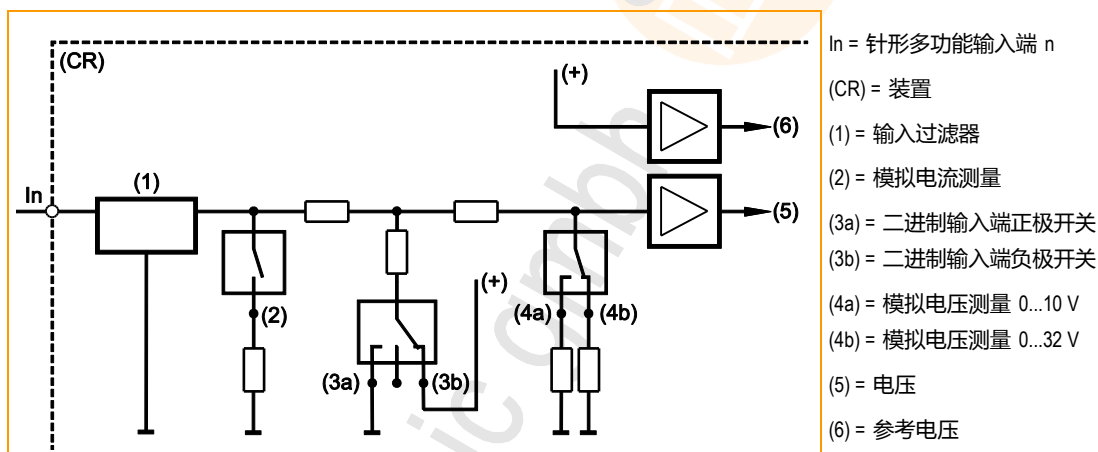
作为备选，还可通过二进制的方式评估模拟通道。

❗注意

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。

► 注意针对二进制评估的更高输入电阻。

8971



图：多功能输出端原理方块图

4.4.2 配置输出端

内容	
二进制和 PWM 输出端	80
输出组 Q1Q2 (Q10...13 / Q20...23)	82
输出组 Q3 (Q30...37)	83
输出组 Q4 (Q40...47)	84

19568
3976

有效的工作模式 → 章节 可能的输入端/输出端工作模式 (→ 页 278)

- ▶ 每个输出端均可通过应用程序进行配置：
表示负载电流→ FB **OUTPUT_CURRENT** (→ 页 201)
PWM 输出端： → FB **PWM1000** (→ 页 210)
配置字节 Qxx_MODE

二进制和 PWM 输出端

1346

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), 含诊断功能和保护
- 包含脉冲宽度调制 (PWM) 的模拟输出端
- PWM 输出端对 H 桥, 不含诊断功能

PWM 可在包含和不含电流控制功能的情况下运行。

 电流控制 PWM 输出端主要用于触发比例液压功能。

 PWM 信号中的中度电流仅可在开启状态下的电流处于测量范围内时通过 FB OUTPUT_CURRENT 正确确定。

14713

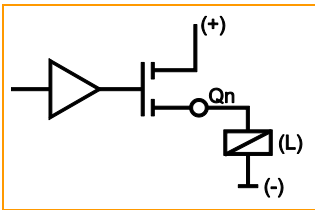
 警告

可能因故障导致财产损失或人身伤害！

以下适用于 PWM 模式下的输出端：

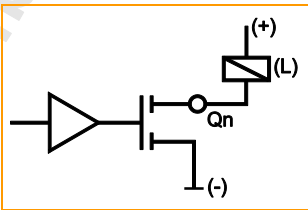
- 无诊断功能
- 未设定 ERROR 标志
- 过载保护 OUT_OVERLOAD_PROTECTION 未启用

15450



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

Qn = 插脚输出端 n
(L) = 负载

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。

短路永久过载或断线时的状况：

(适用于 AH 以上硬件版本, 但不适用于安全模式)

- > 系统标志 ERROR_SHORT_Qx (短路或过载时) 或 ERROR_BREAK_Qx (断线时) 将启用。
- > 仅在短路/过载时：运行时系统禁用受影响的输出端驱动器。
受影响输出端的逻辑保持为 TRUE。
等待时间结束后, 输出端再次启用, 因此定期切换至短路。
等待时间随着输出端负载 (过载) 增加。
短路时开启时间一般为 50 μ s, 显著长于过载的情况。
- ▶ 在应用程序中评估错误标志！
复位输出端逻辑, 必要时停止机器。
如有必要, 通过 RELAY=FALSE (例如通过 ERROR=TRUE) 关闭输出组 VBBr。

排除故障后：

- ▶ 复位错误标志 ERROR_..._Qx。
- > 输出端继电器重新启用输出组 VBBr。
- ▶ 重新设定输出端或重启机器。

输出组 Q1Q2 (Q10...13 / Q20...23)

1378

这些输出端有两种功能。 作为 PWM 输出端时，可通过集成电流测量通道进行诊断，该通道还可用于电流控制输出功能。

使用 **OUTPUT_CURRENT** (→ 页 201)可表示 ≥ 100 mA 的负载电流。

当作为数字输出端时，可利用系统变量 Q1x_MODE...Q2x_MODE 进行配置。 如果要使用诊断，则需额外启用。

输出端信号断线和短路信号分别通过 ERROR_BREAK_Q1Q2 和 ERROR_SHORT_Q1Q2 表示。 如有必要，可在应用程序中隐藏单个输出单错误位。

例如：

分配操作将所选输出端设为工作模式 OUT_DIGITAL_H (含诊断)。 过载保护已启用 (默认状态)。

```
graph LR; A[0001] --- B[OUT_DIGITAL_H]; A --- C[OUT_DIAGNOSTIC]; A --- D[OUT_OVERLOAD_PROTECTION]; B --- E[OR]; C --- E; D --- E; E --- F[Q13_MODE]
```

注意

若要不保护内部测量电阻器，OUT_OVERLOAD_PROTECTION 则应始终保持启用状态 (最大测量电流 4.1 A)。

限制值请务必确保遵循技术资料！

纯粹的 PWM 模式不支持 OUT_OVERLOAD_PROTECTION。

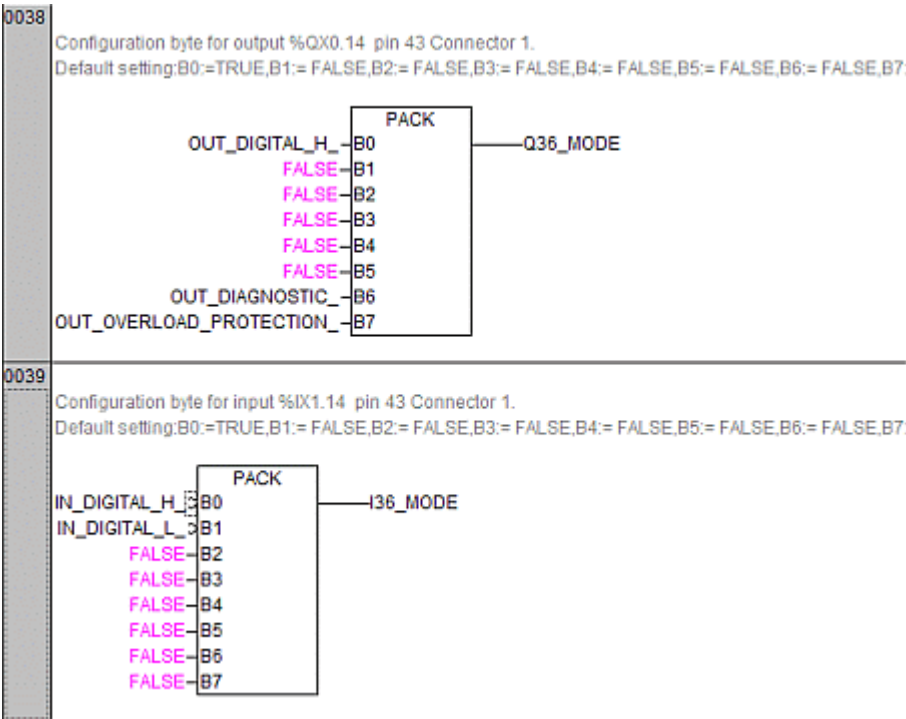
以下情况时断线和短路检测启用...

- 输出端配置为“二进制正极开关 (BH)”，艾尔
- 输出端开启。

输出组 Q3 (Q30...37)

1379

这些输出端可通过系统变量 Q3x_MODE 进行配置。 如果要使用诊断，则需额外启用。 同时还必须将系统标志 I3x_MODE 设为 IN_NOMODE，以禁用相应的输入端。



例如： 右侧分配禁用输入端并将所选输出端设为工作模式“OUT_DIGITAL_H (含诊断)”

输出端信号断线和短路信号分别通过 ERROR_BREAK_Q3 和 ERROR_SHORT_Q3 表示。 如有必要，可在应用程序中隐藏单个输出单错误位。

❗ 限制值请务必确保遵循技术资料！

输出端关闭时，断线检测启用。

输出端开启时，短路检测启用。

输出组 Q4 (Q40...47)

1380

交付时，该输出组禁用，以便通过输入端进行诊断。必须启用输出端以便使用。

这些输出端可通过系统变量 `Q4x_MODE` 进行配置。如果要使用诊断，则需额外启用。同时还必须将系统标志 `I4x_MODE` 设为 `IN_NOMODE`，以禁用相应的输入端。

输出端信号断线和短路信号分别通过 `ERROR_BREAK_Q4` 和 `ERROR_SHORT_Q4` 表示。如有必要，可在应用程序中隐藏单个输出单错误位。

若要执行 H 桥功能，则输出端 `Q41`、`Q42`、`Q45`、`Q46` 可额外切换至模式 `OUT_DIGITAL_L`。

! 限制值请务必确保遵循技术资料！

输出端**关闭**时，断线检测启用。

输出端**开启**时，短路检测启用。

4.5 变量

内容	
保持变量	86
网络变量	86

3130

在本章，您将了解更多关于如何处理变量的信息。

4.5.1 保持变量

3131

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

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）。

声明变量

类别(C)	名称(N)	类型(T)	
VAR	Value_1	WORD	...
变量文件夹(S)	初始值(I)	地址(A)	
CanOpen implicit Vari			...
注释(M):	Counteroperation cycle		

☐ 常量(Q)
☒ 保持变量(R)
☐ 永久变量(P)

确定 取消

4.5.2 网络变量

9856

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

- 将以下库集成至 CODESYS 项目：
- 3S_CANopenNetVar.lib

5 IFM 功能元件

内容	
针对装置 CR0020 的 IFM 库	87
针对装置 CR0020 的 IFM 功能元件	94

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 CR0020 的 IFM 库

内容	
库 ifm_CR0020_V06yyzz.LIB	88
库 ifm_CR0020_CANOpenMaster_V04yynn.LIB	91
库 ifm_CR0020_CANOpenSlave_V04yynn.LIB	91
库 ifm_CAN1_EXT_Vxxyyzz.LIB	92
库 ifm_J1939_x_Vxxyyzz.LIB	92
库 ifm_hydraulic_16bitOS05_Vxxyyzz.LIB	93

14235

5.1.1 库 ifm_CR0020_V06yyzz.LIB

19531

此为装置库。 该 IFM 库包含以下功能块：

功能元件	简短说明
ANALOG_RAW (→ 页 174)	为各个输入端提供模拟/数字转换器非标准值
CAN2 (→ 页 104)	初始化 CAN 接口 x x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
CAN1_BAUDRATE (→ 页 95)	设定 CAN 接口 1 总线参与者的传送速度
CAN1_DOWNLOADID (→ 页 96)	设定 CAN 接口 1 的下载 ID
CANx_ERRORHANDLER (→ 页 105)	在 CAN 接口 x 上执行“手动”总线恢复 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
CANx_EXT_RECEIVE_ALL (→ 页 107)	CAN interface x: 配置所有数据接收对象并读取数据对象的接收缓冲区 x = 2 = CAN 接口编号
CANx_RECEIVE (→ 页 109)	CAN interface x: 配置数据接收对象并读取数据对象的接收缓冲区 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
CANx_RECEIVE_RANGE (→ 页 111)	CAN interface x: 配置一系列数据接收对象并读取数据对象的接收缓冲区 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
CANx_SDO_READ (→ 页 138)	CAN interface x: 读取 SDO, 所示索引来自节点。 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
CANx_SDO_WRITE (→ 页 140)	CAN interface x: 写入 SDO, 索引指定给节点。 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
CANx_TRANSMIT (→ 页 114)	将 CAN 数据对象 (消息) 传输至 CAN 接口 x 以在每次调用时传送 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
CHECK_DATA (→ 页 258)	生成可配置内存区域的校验和 (CRC) 并检查针对不想要的更改的内存区域数据
DELAY (→ 页 232)	延迟输入值的输出, 延迟时间 T (空载时间元件)
FAST_COUNT (→ 页 184)	针对快速输入脉冲的计数功能块
FLASHREAD (→ 页 251)	将不同的数据类型直接从闪存传输至 RAM
FLASHWRITE (→ 页 252)	将不同的数据类型直接写入闪存
FRAMREAD (→ 页 254)	将不同的数据类型直接从 FRAM 内存传输至 RAM FRAM 在此表示所有类型的非易失快速内存。
FRAMWRITE (→ 页 255)	将不同的数据类型直接写入 FRAM 内存 FRAM 在此表示所有类型的非易失快速内存。
FREQUENCY (→ 页 186)	测量到达所选通道的信号的频率。

IFM 功能元件

针对装置 CR0020 的 IFM 库

功能元件	简短说明
GET_IDENTITY (→ 页 260)	读取存储在装置中的特定标识： • 装置的硬件名称和硬件版本 • 装置运行时系统的名称 • 装置运行时系统的版本和版本号 • 应用程序的名称（之前已通过以下方式保存：SET_IDENTITY (→ 页 263)）
GLR (→ 页 233)	同步控制器是包含 PID 特性的控制器
INC_ENCODER (→ 页 188)	针对编码器评估的递增/递减计数器功能
INPUT_ANALOG (→ 页 175)	模拟输入通道：以下对象的备选测量... • 电流 • 电压
INPUT_CURRENT (→ 页 178)	模拟输入通道电流测量
INPUT_VOLTAGE (→ 页 179)	模拟输入通道电压测量
MEMCPY (→ 页 256)	直接写入和读取内存中的不同数据类型
MEMORY_RETAIN_PARAM (→ 页 248)	确定不同事件的剩余数据特性
NORM (→ 页 181)	将定义的限值范围内的值 [WORD] 正常化为新的限值范围内的值
OCC_TASK (→ 页 199)	OCC = Output Current Control PWMi 输出通道的电流控制器 在为时 5 ms 的周期中，每个功能实例调用一次。
OUTPUT_CURRENT (→ 页 201)	测量输出通道上的电流（平均，通过抖动周期）
OUTPUT_CURRENT_CONTROL (→ 页 202)	PWMi 输出通道的电流控制器
PERIOD (→ 页 192)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 194)	测量所示周期内所示通道频率和以 [μs] 为单位的周期（周期时间）。此外，传号空号比以 [%] 为单位显示。
PHASE (→ 页 196)	读取包含快速输入端的通道对对比信号的相位
PID1 (→ 页 235)	PID 控制器
PID2 (→ 页 237)	PID 控制器
PT1 (→ 页 240)	受控系统，一级延迟
PWM (→ 页 204)	初始化并配置具备 PWM 功能的输出通道 通过 RELOAD 定义 PWM 频率
PWM100 (→ 页 208)	初始化并配置具备 PWM 功能的输出通道 以 [Hz] 为单位表示 PWM 频率 按照 1 % 的步距表示传号空号比
PWM1000 (→ 页 210)	初始化并配置具备 PWM 功能的输出通道 传号空号比可按 1‰ 的步距表示
SERIAL_PENDING (→ 页 161)	确定存储在序列接收缓冲区的数据字节的数量
SERIAL_RX (→ 页 162)	读取每次调用时来自序列接收缓冲区的已接收数据字节

IFM 功能元件

针对装置 CR0020 的 IFM 库

功能元件	简短说明
SERIAL_SETUP (→ 页 164)	初始化 RS232 串行接口
SERIAL_TX (→ 页 166)	通过 RS232 串行接口传送一个数据字节
SET_DEBUG (→ 页 262)	安排 DEBUG 模式或监控模式 (视 TEST 输入端而定)
SET_IDENTITY (→ 页 263)	设定应用程序特定程序 ID
SET_INTERRUPT_I (→ 页 168)	通过定义的输入通道请求中断后有条件地执行程序部分
SET_INTERRUPT_XMS (→ 页 171)	以 xms 的时间间隔有条件地执行程序部分
SET_PASSWORD (→ 页 264)	设定程序和内存上传访问控制的用户密码
SOFTRESET (→ 页 242)	使得装置完全重启
TIMER_READ (→ 页 244)	读取当前系统时间, 以 [ms] 为单位 最大值 = 49d 17h 2min 47s 295ms
TIMER_READ_US (→ 页 245)	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_CR0020_CANopenMaster_V04yynn.LIB

18714

该库包含用于将装置作为 CANopen 主站运行的功能块。

库仅适用于第 1 个 CAN 接口。

$x = 1$ = CAN 接口编号

该 IFM 库包含以下功能块：

功能元件	简短说明
CANx_MASTER_EMcy_HANDLER (→ 页 117)	处理 CAN 接口 x 上的 CANopen 主站的装置特定错误状态 $x = 1$ = CAN 接口编号
CANx_MASTER_SEND_EMERGENCY (→ 页 118)	发送 CAN 接口 x 上的 CANopen 主站的应用程序特定错误状态 $x = 1$ = CAN 接口编号
CANx_MASTER_STATUS (→ 页 121)	在 CAN 接口 x 上显示作为 CANopen 主站的装置的状态 $x = 1$ = CAN 接口编号

5.1.3 库 ifm_CR0020_CANopenSlave_V04yynn.LIB

18719

该库包含用于将装置作为 CANopen 从站运行的功能块。

库仅适用于第 1 个 CAN 接口。

$x = 1$ = CAN 接口编号

该 IFM 库包含以下功能块：

功能元件	简短说明
CANx_SLAVE_EMcy_HANDLER (→ 页 128)	处理 CAN 接口 x 上的 CANopen 从站的装置特定错误状态： • 错误寄存器 (索引 0x1001) 和 • CANopen 对象目录的错误字段 (索引 0x1003) $x = 1$ = CAN 接口编号
CANx_SLAVE_NODEID (→ 页 130)	在应用程序运行时间启用 CAN 接口 x CANopen 从站节点 ID 的设定 $x = 1$ = CAN 接口编号
CANx_SLAVE_SEND_EMERGENCY (→ 页 131)	发送 CAN 接口 x 上的 CANopen 从站的应用程序特定错误状态 $x = 1$ = CAN 接口编号
CANx_SLAVE_STATUS (→ 页 134)	在 CAN 接口 x 上显示作为 CANopen 从站的装置的状态 $x = 1$ = CAN 接口编号

5.1.4 库 ifm_CAN1_EXT_Vxxyyzz.LIB

18732

该库包含针对第一个 CAN 接口发动机控制的 互补 POU。
库仅适用于第 1 个 CAN 接口。

该 **IFM** 库包含以下功能块：

功能元件	简短说明
CAN1_EXT (→ 页 97)	初始化同样针对扩展模式的 CAN 接口 1 设定模式和波特率
CAN1_EXT_ERRORHANDLER (→ 页 98)	在 CAN 接口 1 上执行“手动”总线恢复
CAN1_EXT_RECEIVE (→ 页 99)	CAN interface 1: 配置数据接收对象并读取数据对象的接收缓冲区
CANx_EXT_RECEIVE_ALL (→ 页 107)	CAN interface x: 配置所有数据接收对象并读取数据对象的接收缓冲区 x = 1 = CAN 接口编号
CAN1_EXT_TRANSMIT (→ 页 101)	将 CAN 数据对象 (消息) 传输至 CAN 接口 1 以在每次调用时传送

5.1.5 库 ifm_J1939_x_Vxxyyzz.LIB

18722

该库包含用于发动机控制的功能块。

x = 1...2 = CAN 接口编号

该 **IFM** 库包含以下功能块：

功能元件	简短说明
J1939_x (→ 页 143)	CAN interface x: 通信协议 SAE J1939 的协议处理器 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
J1939_x_GLOBAL_REQUEST (→ 页 144)	CAN interface x: 处理 J1939 网络参与者的全局请求和数据接受 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
J1939_x_RECEIVE (→ 页 147)	CAN interface x: 接收单个消息或消息块 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
J1939_x_RESPONSE (→ 页 151)	CAN interface x: 处理针对请求消息的自动响应 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
J1939_x_SPECIFIC_REQUEST (→ 页 154)	CAN interface x: 特定 J1939 网络参与者自动请求单个消息 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)
J1939_x_TRANSMIT (→ 页 157)	CAN interface x: 发送单个消息或消息块 x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

5.1.6 库 ifm_hydraulic_16bitOS05_Vxxyzz.LIB

19535

该库包含用于液压控制的功能块。

该 IFM 库包含以下功能块：

功能元件	简短说明
CONTROL_OCC (→ 页 213)	OCC = Output Current Control 将输入值 [WORD] 换算至所示电流范围。
JOYSTICK_0 (→ 页 216)	将操纵杆的信号 [INT] 换算成明确定义的特性曲线，标准化为 0... 1000
JOYSTICK_1 (→ 页 220)	换算操纵杆 D 的信号 [INT]，标准化为 0... 1000
JOYSTICK_2 (→ 页 224)	将操纵杆的信号 [INT] 换算成可配置的特性曲线；自由选择标准化
NORM_HYDRAULIC (→ 页 227)	将定义的限值范围内的值 [DINT] 正常化为新的限值范围内的值

5.2 针对装置 CR0020 的 IFM 功能元件

内容	
功能元件：CAN 第 2 层	94
功能元件：CANopen 主站	116
功能元件：CANopen 从站	127
功能元件：CANopen SDO	137
功能元件：SAE J1939	142
功能元件：串行接口	160
功能元件：通过处理中断优化 PLC 周期	167
功能元件：处理输入值	173
功能元件：调整模拟值	180
功能元件：针对频率和周期测量的计数器功能	183
功能元件：PWM 功能	198
功能元件：液压控制	212
功能元件：控制器	230
功能元件：软件复位	241
功能元件：时间测量/设定	243
功能元件：保存、读取和转换内存中的数据	246
功能元件：数据访问和数据检查	257

13988
3826

您可在了解适用于本装置的 IFM 功能元件（按主题分类）。

5.2.1 功能元件：CAN 第 2 层

内容	
CAN1_BAUDRATE	95
CAN1_DOWNLOADID	96
CAN1_EXT	97
CAN1_EXT_ERRORHANDLER	98
CAN1_EXT_RECEIVE	99
CAN1_EXT_TRANSMIT	101
CAN2	104
CANx_ERRORHANDLER	105
CANx_EXT_RECEIVE_ALL	107
CANx_RECEIVE	109
CANx_RECEIVE_RANGE	111
CANx_TRANSMIT	114

13754

下文将讲述应用程序中使用的 CAN 功能块（第 2 层）。

CAN1_BAUDRATE

651

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxyyyz.LIB

CODESYS 中的符号：



说明

654

CAN1_BAUDRATE 设定总线参与者的传送速度。

► 为此，可在输入端 BAUDRATE 输入以 Kbits/s 为单位的相应值。

通知

请注意 CR250n、CR0301、CR0302 和 CS0015：

长久使用本单元可能导致 EEPROM 内存模块受损！

► 仅在第一个程序周期初始化时执行该单元一次！

► 之后通过 ENABLE = FALSE 再次禁用该单元！

❗新的波特率将在 RESET（电压 OFF/ON 或软复位）后生效。

ExtendedController：在从站模块中，新的波特率将在电压 OFF/ON 之后生效。

输入端参数

655

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

CAN1_DOWNLOADID

645

= CAN1 下载 ID

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB



说明

648

CAN1_DOWNLOADID 设定第一个 CAN 接口的下载 ID。

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

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

通知

注意 CR250n、CR0301、CR0302 和 CS0015：

如果永久使用该功能，EEPROM 内存模块可能受损。

► 仅在第一个程序周期初始化时执行该 POU 一次！

之后再次禁用 POU (ENABLE = "FALSE")！

输入端参数

649

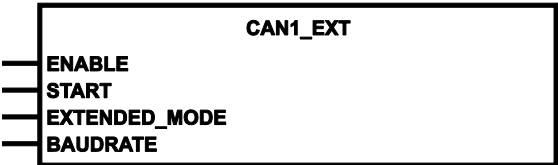
参数	数据类型	说明
ENABLE	BOOL	TRUE（第 1 个周期中）： 采用和启用参数 else: 未执行该功能

CAN1_EXT

4192

单元类型 = 功能块 (FB)
元件类型 ifm_CAN1_EXT_Vxyyyzz.LIB

CODESYS 中的符号：



说明

4333

CAN1_EXT 初始化针对扩展 ID (29 位) 的第一个 CAN 接口。
如果要使用第一个 CAN 接口 (如包含 **SAE J1939** 功能库) , 则需检索 FB。
波特率更改将在电压 OFF/ON 之后生效。
CAN 1 和 CAN 2 的波特率可进行不同的设定。
接口重启期间 , 仅可针对一个周期设定输入端 START !

❗必须先执行 FB , 再使用 CAN1_EXT_ .

输入端参数

4334

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
START	BOOL	TRUE (第 1 个周期中) : 开始 CAN 接口 x 的 CAN 协议 FALSE: 在程序进一步处理期间
EXTENDED_MODE	BOOL := FALSE	TRUE: CAN 接口 ID 为 29 位 FALSE: CAN 接口 ID 为 11 位
BAUDRATE	WORD := 125	波特率 [Kbits/s] 可允许 = 50、100、125、250、500、800、1000

CAN1_EXT_ERRORHANDLER

4195

单元类型 = 功能块 (FB)
元件类型 ifm_CAN1_EXT_Vxxyyzz.LIB

CODESYS 中的符号：



说明

4335

CAN1_EXT_ERRORHANDLER 监控第一个 CAN 接口并评估 CAN 错误。如果出现特定数量的传送错误，CAN 参与者则进入错误认可状态。如果错误频率降低，则参与者再次变回错误激活状态（= 正常状况）。

如果参与者已是错误认可状态，且仍发生传送错误，则断开与总线的连接（= 总线关闭）并设定错误位 CANx_BUSOFF。仅可在排除“总线关闭”条件后重新连接总线（信号 BUSOFF_RECOVER）。

而后必须在应用程序中复位错误位 CANx_BUSOFF。

❗如果要使用自动总线恢复功能（默认设定），则**不得**在程序中集成 CAN1_EXT_ERRORHANDLER 并实例化！

输入端参数

2177

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

CAN1_EXT_RECEIVE

4302

单元类型 = 功能块 (FB)
元件类型 ifm_CAN1_EXT_Vxyyyzz.LIB

CODESYS 中的符号：



说明

4336

CAN1_EXT_RECEIVE 配置数据接收对象并读取数据对象的接收缓冲区。

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

进一步的程序周期调用 CAN1_EXT_RECEIVE 以读取相应的接收缓冲区，如果程序周期长，则多次调用。程序员必须通过评估字节 AVAILABLE 以确保从缓冲区检索新接收的数据对象并进一步处理。

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

评估输出端 OVERFLOW 即可检测数据缓冲器的溢出。如果 OVERFLOW = TRUE，则至少丢失 1 个数据对象。

❗如果要使用该单元，则必须先针对扩展 ID 初始化第 1 个 CAN 接口并利用 **CAN1_EXT** (→ 页 97)。

输入端参数

2172

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

输出端参数

19810

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

CAN1_EXT_TRANSMIT

4307

单元类型 = 功能块 (FB)
元件类型 ifm_CAN1_EXT_Vxxyyzz.LIB

CODESYS 中的符号：



说明

4337

CAN1_EXT_TRANSMIT 将 CAN 数据对象（消息）传送至 CAN 控制器以便传送。

在程序周期中，每个数据对象均调用 FB；如果程序周期长，则多次调用。程序员必须通过评估输出端 RESULT 确保已接受其传送命令。简单来说，若速度为 125 kbits/s，则每 1 ms 可执行一个传送命令。

可通过输入端 ENABLE = FALSE 暂时阻止执行 FB。这样可防止总线过载等。

如果标志分配给各个数据对象并通过 ENABLE 输入端控制 FB 的执行，则几乎可同时传送多个数据对象。

❗如果要使用该单元，则必须先针对扩展 ID 初始化第 1 个 CAN 接口并利用 **CAN1_EXT** (→ 页 97).

输入端参数

4380

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



输出端参数

614

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

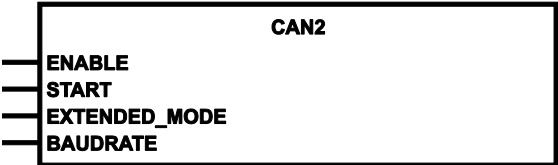


CAN2

639

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

642

CAN2 初始化第 2 个 CAN 接口。

如果要使用第 2 个 CAN 接口，则必须调用 FB。

波特率更改将在电压 OFF/ON 之后生效。

CAN 1 和 CAN 2 的波特率可进行不同的设定。

接口重启期间，仅可针对一个周期设定输入端 START ！

就第 2 个 CAN 接口而言，针对 **SAE J1939** 和 **CAN 接口使用（按照 ISO 11992）** 等的库可用。符合 ISO 11992 的 FB 仅适用于第 2 个 CAN 接口上的 CR2501。

❗必须先执行 FB，再使用 CAN2。

输入端参数

643

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
START	BOOL	TRUE (第 1 个周期中) : 开始 CAN 接口 x 的 CAN 协议 FALSE: 在程序进一步处理期间
EXTENDED_MODE	BOOL := FALSE	TRUE: CAN 接口 ID 为 29 位 FALSE: CAN 接口 ID 为 11 位
BAUDRATE	WORD := 125	波特率 [Kbits/s] 可允许 = 50、100、125、250、500、800、1000

CANx_ERRORHANDLER

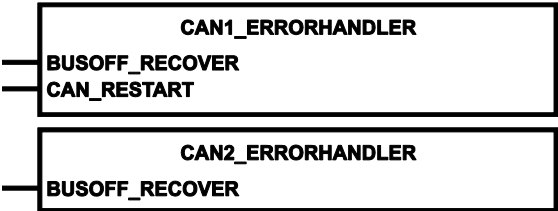
633

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号 :



说明

636

监控 CAN 接口的错误程序

CANx_ERRORHANDLER 监控 CAN 接口并评估 CAN 错误。 如果出现特定数量的传送错误, CAN 参与者则进入错误认可状态。 如果错误频率降低, 则参与者再次变回错误激活状态 (= 正常状况)。

如果参与者已是错误认可状态, 且仍发生传送错误, 则断开与总线的连接 (= 总线关闭) 并设定错误位 CANx_BUSOFF。 仅可在排除“总线关闭”条件后重新连接总线 (信号 BUSOFF_RECOVER)。

输入端 CAN_RESTART 用于修正其他 CAN 错误。 CAN 接口重新初始化。

而后必须在应用程序中复位错误位。

重启接口的步骤各不相同 :

- 针对 CAN 接口 1 或仅包含一个 CAN 接口的装置 :
设定输入端 CAN_RESTART = TRUE (仅 1 个周期)
- 针对 CAN 接口 2 :
设定输入端 START = TRUE (仅 1 个周期) **CAN2** (→ 页 104)

❗注意

原则上, 必须执行 CAN2 以初始化第二个 CAN 接口, 而后 FB 方可用于该接口。

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

输入端参数

637

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

CANx_EXT_RECEIVE_ALL

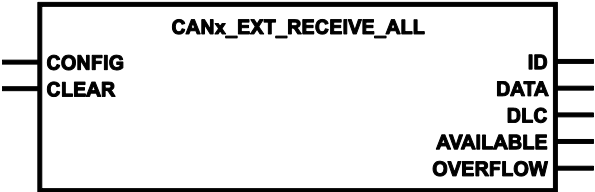
4183

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CANx_EXT_Vxxyyzz.LIB

CODESYS 中的符号：



说明

4326

CANx_EXT_RECEIVE_ALL 配置所有数据接收对象并读取数据对象的接收缓冲区。

初始化期间必须调用一次 FB，以向 CAN 控制器告知数据对象的标识符。

在进一步的程序周期中，调用 CANx_EXT_RECEIVE_ALL 以读取相应的接收缓冲区，若是长程序周期，还可重复。程序员必须通过评估字节 AVAILABLE 以确保从缓冲区检索新接收的数据对象并进一步处理。

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

评估输出端 OVERFLOW 即可检测数据缓冲器的溢出。如果 OVERFLOW = TRUE，则至少丢失 1 个数据对象。

接收缓冲区：单位标识符最多 16 个软件缓冲区。

输入端参数

4329

参数	数据类型	说明
CONFIG	BOOL	TRUE (第 1 个周期中) : 配置数据对象 FALSE: 在程序进一步处理期间
CLEAR	BOOL	TRUE: 删除接收缓冲区 FALSE: function element is not executed

输出端参数

2292

参数	数据类型	说明
ID	DWORD	数据对象 ID 编号
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
DLC	BYTE	包含 SRDO 的 DATA 数组接收字节的数量 允许 : 0...8
AVAILABLE	BYTE	已从接收缓冲器接收但还未读取 (调用 FB 之前) 的 CAN 电报的数量。 可能的值 = 0...16 0 = 无可用的有效数据
OVERFLOW	BOOL	TRUE: 数据缓冲区溢出 ⇒ 数据丢失 ! FALSE: 数据缓冲区无数据丢失

CANx_RECEIVE

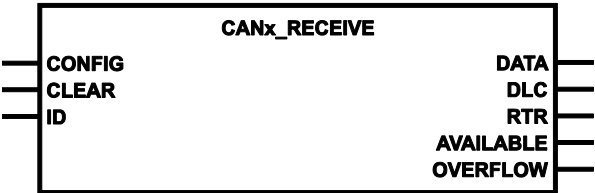
627

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号 :



说明

630

CANx_RECEIVE 配置数据接收对象并读取数据对象的接收缓冲区。

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

在进一步的程序周期中，调用 CANx_RECEIVE 以读取相应的接收缓冲区，若是长程序周期，还可重复。程序员必须通过评估字节 AVAILABLE 以确保从缓冲区检索新接收的数据对象并进一步处理。

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

评估输出端 OVERFLOW 即可检测数据缓冲器的溢出。如果 OVERFLOW = TRUE，则至少丢失 1 个数据对象。

❗如果要使用 CAN2_RECEIVE，则必须先初始化第二个 CAN 接口 **CAN2** (→ 页 104).

输入端参数

631

参数	数据类型	说明
CONFIG	BOOL	TRUE (第 1 个周期中) : 配置数据对象 FALSE: 在程序进一步处理期间
CLEAR	BOOL	TRUE: 删除接收缓冲区 FALSE: function element is not executed
ID	WORD	数据对象 ID 编号 允许值 = 0...2 047

输出端参数

19810

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

CANx_RECEIVE_RANGE

4179

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxyyzz.LIB (xx ≥ 05)

CODESYS 中的符号 :



说明

2295

CANx_RECEIVE_RANGE 配置一系列数据接收对象并读取数据对象的接收缓冲区

就第一个 CAN 接口而言, 每 1 位最多可有 2048 个 ID。

就第二个 CAN 接口而言, 每 11 或 29 位最多可有 256 个 ID。

第二个 CAN 接口需要很长的初始化时间。 为确保电子狗不作出响应, 在范围较大的情况下应将进程分发至几个周期。 →**示例** (→ 页 [113](#)).

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

FB 不可针对同一 CAN 接口的同一 ID 搭配 **CANx_RECEIVE** (→ 页 [109](#)) 或 CANx_RECEIVE_RANGE。

在进一步的程序周期中, 调用 CANx_RECEIVE_RANGE 以读取相应的接收缓冲区, 若是长程序周期, 还可重复。 程序员需通过评估字节 AVAILABLE 以确保从缓冲区 SOFORT 检索新接收的数据对象并进一步处理, 因为数据仅可用于一个周期。

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

评估输出端 OVERFLOW 即可检测数据缓冲器的溢出。 如果 OVERFLOW = TRUE, 则至少丢失 1 个数据对象。

接收缓冲区: 单位标识符最多 16 个软件缓冲区。

输入端参数

2290

参数	数据类型	说明
CONFIG	BOOL	TRUE (第 1 个周期中) : 配置数据对象 FALSE: 在程序进一步处理期间
CLEAR	BOOL	TRUE: 删除接收缓冲区 FALSE: function element is not executed
FIRST_ID	CAN1: WORD CAN2: DWORD	系列第一个数据对象 ID 的编号 正常帧允许值 = 0...2 047 (2 ¹¹) 扩展帧允许值 = 0...536 870 911 (2 ²⁹)
LAST_ID	CAN1: WORD CAN2: DWORD	系列最后个数据对象 ID 的编号 正常帧允许值 = 0...2 047 (2 ¹¹) 扩展帧允许值 = 0...536 870 911 (2 ²⁹) LAST_ID 需大于 FIRST_ID !

输出端参数

4381

参数	数据类型	说明
ID	CAN1: WORD CAN2: DWORD	已传送数据对象 ID
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
DLC	BYTE	从接收缓冲区读取的 CAN 电报字节的数量 允许: 0...8
AVAILABLE	BYTE	已从接收缓冲器接收但还未读取 (调用 FB 之前) 的 CAN 电报的数量。 可能的值 = 0...16 0 = 无可用的有效数据
OVERFLOW	BOOL	TRUE: 数据缓冲区溢出 ⇒ 数据丢失! FALSE: 数据缓冲区无数据丢失

例如：CANx_RECEIVE_RANGE 在 4 个周期中的初始化

2294

```

PLC_PRG (PRG-ST) (+1/181/-1/88)
0001 PROGRAM PLC_PRG
0002 VAR
0003   init : BOOL := FALSE;
0004   initstep : WORD := 1;
0005   can20 : CAN2;
0006   cr2 : CAN2_RECEIVE_RANGE;
0007   cnt : WORD;
0008 END_VAR
0009
0001 (* CAN2 init *)
0002 can20(ENABLE:= TRUE, START:= init, EXTENDED_MODE:= FALSE, BAUDRATE:= 125);
0003
0004 (* CAN2_RECEIVE_RANGE in mehreren Steps initialisieren *)
0005 CASE initstep OF
0006   1:
0007     cr2(CONFIG:= TRUE, CLEAR:= FALSE, FIRST_ID:= 16#100, LAST_ID:= 16#10F, ID:= , DATA:= , DLC:= , AVAILABLE:= , OVERFLOW:= );
0008     initstep := initstep + 1;
0009   2:
0010     cr2(CONFIG:= TRUE, CLEAR:= FALSE, FIRST_ID:= 16#110, LAST_ID:= 16#11F, ID:= , DATA:= , DLC:= , AVAILABLE:= , OVERFLOW:= );
0011     initstep := initstep + 1;
0012   3:
0013     cr2(CONFIG:= TRUE, CLEAR:= FALSE, FIRST_ID:= 16#120, LAST_ID:= 16#12F, ID:= , DATA:= , DLC:= , AVAILABLE:= , OVERFLOW:= );
0014     initstep := initstep + 1;
0015   4:
0016     cr2(CONFIG:= TRUE, CLEAR:= FALSE, FIRST_ID:= 16#130, LAST_ID:= 16#13F, ID:= , DATA:= , DLC:= , AVAILABLE:= , OVERFLOW:= );
0017     initstep := initstep + 1;
0018   ELSE
0019     cr2(CONFIG:= FALSE, CLEAR:= FALSE, FIRST_ID:= 16#100, LAST_ID:= 16#100, ID:= , DATA:= , DLC:= , AVAILABLE:= , OVERFLOW:= );
0020   END_CASE
0021
0022 init := FALSE;
0023
0024 (* Test *)
0025 IF cr2.available > 0 THEN
0026   cnt := cnt + 1;
0027 END_IF

```

CANx_TRANSMIT

609

$x = 1 \dots n$ = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

612

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

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

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

如果标志分配给各个数据对象并通过 ENABLE 输入端控制 FB 的执行, 则几乎可同时传送多个数据对象。

❗ 如果要使用 CAN2_TRANSMIT, 则必须先初始化第二个 CAN 接口 **CAN2** (→ 页 [104](#))。

输入端参数

613

参数	数据类型	说明
ID	WORD	数据对象 ID 编号 允许值 = 0..2 047
DLC	BYTE	从 DATA 数组传送的字节的数量 允许: 0..8
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1..8 字节)
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定

输出端参数

614

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



5.2.2 功能元件：CANopen 主站

内容	
CANx_MASTER_EMCY_HANDLER	117
CANx_MASTER_SEND_EMERGENCY	118
CANx_MASTER_STATUS	121

1870

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



CANx_MASTER_EMCY_HANDLER

13192

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_CANopenMaster_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2009

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

- 错误状态传送至网络 ,
- 应用程序的错误消息存储在对象目录中。

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

❗如果要应用程序特定错误消息存储在对象目录中, 则调用 CANx_MASTER_EMCY_HANDLER 之前必须 (重复) 调用 CANx_MASTER_SEND_EMERGENCY (→ 页 118)Y.

输入端参数

2010

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

输出端参数

2011

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

CANx_MASTER_SEND_EMERGENCY

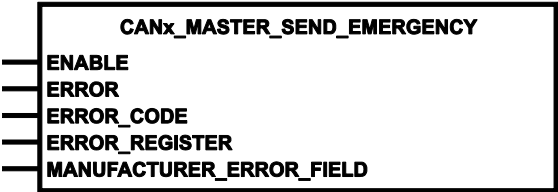
13195

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_CANopenMaster_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2015

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

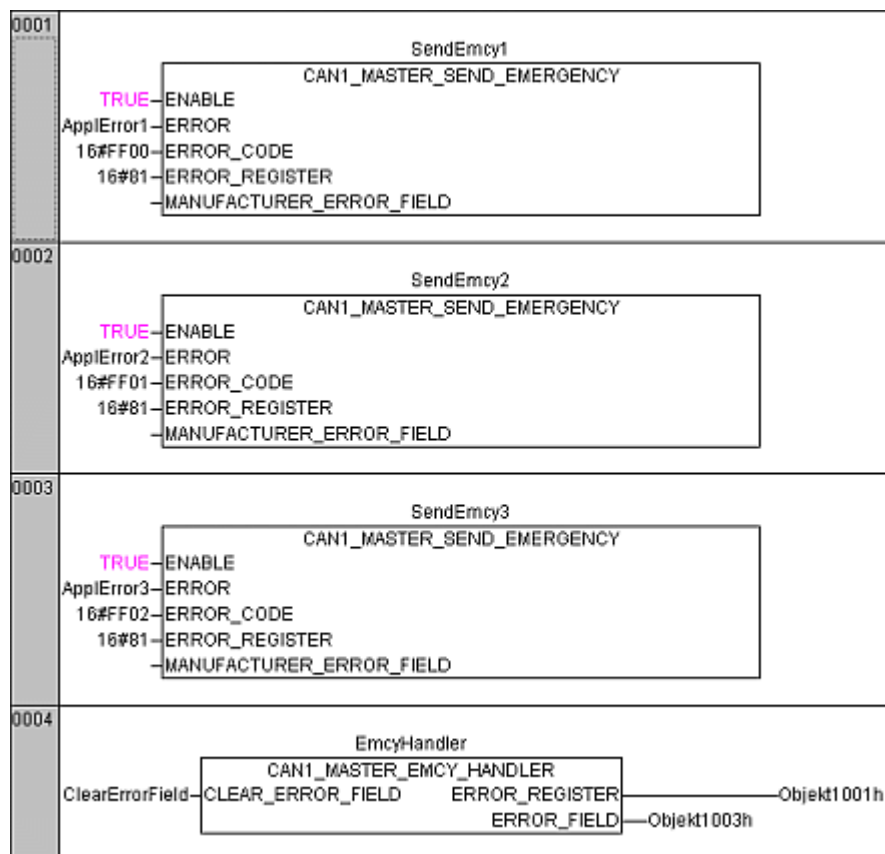
❗ 如果要将应用程序特定错误状态存储在对象目录中, 则必须在调用 **CANx_MASTER_EMCY_HANDLER** (→ 页 117) 之前 (重复) 调用 CANx_MASTER_SEND_EMERGENCY。

输入端参数

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

例如：CANx_MASTER_SEND_EMERGENCY

2018



在该示例中，随后将生成 3 个错误消息：

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

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

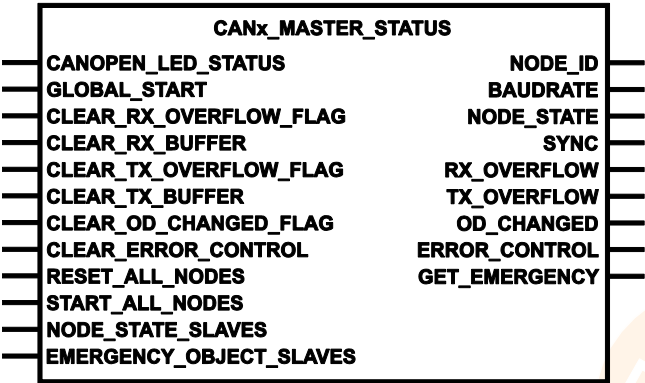
CANx_MASTER_STATUS

2021

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_CANOpenMaster_Vxxyyzz.LIB



说明

2024

用作 CANopen 的装置的状态显示

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

- 监控网络状态
- 监控已连接从站的状态
- 复位或启动网络中的从站。

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

输入端参数

2025

参数	数据类型	说明
CANOPEN_LED_STATUS	BOOL	(输入端不可用于 PDM 装置) TRUE： 控制器状态 LED 切换至 模式 "CANopen" : 闪烁频率 0.5 Hz = PRE-OPERATIONAL 闪烁频率 2.0 Hz = OPERATIONAL 该工作模式不改变其他诊断 LED 信号。
GLOBAL_START	BOOL	TRUE: 所有已连接的网络参与者 (从站) 在网络初始化期间同时启动 (⇒ 状态 OPERATIONAL)。 FALSE: 已连接的网络参与者 先后启动。

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

参数	数据类型	说明
CLEAR_RX_OVERFLOW_FLAG	BOOL	FALSE ⇒ TRUE (edge): 清除错误标志 RX_OVERFLOW else: 未执行该功能
CLEAR_RX_BUFFER	BOOL	FALSE ⇒ TRUE (edge): 删除接收缓冲区中的数据 else: 未执行该功能
CLEAR_TX_OVERFLOW_FLAG	BOOL	FALSE ⇒ TRUE (edge): 清除错误标志 TX_OVERFLOW else: 未执行该功能
CLEAR_TX_BUFFER	BOOL	FALSE ⇒ TRUE (edge): 删除传送缓冲区中的数据 else: 未执行该功能
CLEAR_OD_CHANGED_FLAG	BOOL	FALSE ⇒ TRUE (edge): 删除标志 OD_CHANGED else: 未执行该功能
CLEAR_ERROR_CONTROL	BOOL	FALSE ⇒ TRUE (edge): 删除保护错误列表 (ERROR_CONTROL) else: 未执行该功能
RESET_ALL_NODES	BOOL	FALSE ⇒ TRUE (edge): 所有已连接的网络参与者 (从站) 通过 NMT 命令复位 else: 未执行该功能
START_ALL_NODES	BOOL	FALSE ⇒ TRUE (edge): 所有已连接的网络参与者 (从站) 通过 NMT 命令启动 else: 未执行该功能
NODE_STATE_SLAVES	DWORD	显示所有网络节点的状态。 代码示例 → 章节 例如：CANx_MASTER_STATUS (→ 页 125)
EMERGENCY_OBJECT_SLAVES	DWORD	显示所有网络节点的最新错误消息。  → 章节 访问应用程序运行时结构 (→ 页 126)

输出端参数

2029

参数	数据类型	说明
NODE_ID	BYTE	CANopen 主站当前节点 ID
BAUDRATE	WORD	CANopen 主站的当前波特率, 以 [kBaud] 为单位
NODE_STATE	INT	CANopen 主站的当前状态
SYNC	BOOL	CANopen 主站的 SYNC 信号 TRUE: 已在最后一个周期中发送 SYNC 信号 FALSE: 未在最后一个周期中发送 SYNC 信号
RX_OVERFLOW	BOOL	TRUE: Error: 接收缓冲区溢出 FALSE: 无溢出
TX_OVERFLOW	BOOL	TRUE: Error: 传送缓冲区溢出 FALSE: 无溢出
OD_CHANGED	BOOL	TRUE: CANopen 主站对象目录中的数据已更改 FALSE: 无数据更改
ERROR_CONTROL	ARRAY [0..7] OF BYTE	数组包含遗漏网络节点 (保护或检测信号错误) 的列表 (最多 8 个)
GET_EMERGENCY	STRUCT CANx_EMERGENCY_MESSAGE	在输出端, 结构 CANx_EMERGENCY_MESSAGE 的数据可用。 始终显示 CANopen 网络中最后接收的 EMCY 消息。 若要获取所有发生的错误的列表, 则必须评估数组 EmergencyObjectSlavesArray !

内部结构的参数

2030

以下为该 FB 中使用的数组的结构

参数	数据类型	说明
CANx_EMERGENCY_MESSAGE	STRUCT	NODE_ID: BYTE ERROR_CODE: WORD ERROR_REGISTER: BYTE MANUFACTURER_ERROR_FIELD: ARRAY[0...4] OF BYTE 结构由库 ifm_CR0020_CANopenMaster_Vxxyzz.LIB 的全局变量决定。
CANx_NODE_STATE	STRUCT	NODE_ID: BYTE NODE_STATE: BYTE LAST_STATE: BYTE RESET_NODE: BOOL START_NODE: BOOL PREOP_NODE: BOOL SET_TIMEOUT_STATE: BOOL SET_NODE_STATE: BOOL 结构由库 ifm_CR0020_CANopenMaster_Vxxyzz.LIB 的全局变量决定。

以控制器 CR0020 为例，以下代码段显示 FB CANx_MASTER_STATUS 的使用。

例如：CANx_MASTER_STATUS

2031

从站信息

2033

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

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

❗ 数组元件的数量与节点 ID 不一致。可从 NODE_ID 下的相应结构读取标识符。

```
0001 PROGRAM MasterStatus
0002 VAR
0003   Status: CR0020_MASTER_STATUS;
0004   LedStatus: BOOL := TRUE;
0005   GlobalStartNodes: BOOL := TRUE;
0006   ClearRxOverflowFlag: BOOL;
0007   ClearRxBuffer: BOOL;
0008   ClearTxOverflowFlag: BOOL;
0009   ClearTxBuffer: BOOL;
0010   ClearOdChanged: BOOL;
0011   ClearErrorControl: BOOL;
0012   ResetAllNodes: BOOL;
0013   StartAllNodes: BOOL;
0014   NodeId: BYTE;
0015   Baudrate: WORD;
0016   NodeState: INT;
0017   Sync: BOOL;
0018   RxOverflow: BOOL;
0019   TxOverflow: BOOL;
0020   OdChanged: BOOL;
0021   GuardHeartbeatErrorArray: ARRAY[0..7] OF BYTE;
0022   GetEmergency: EMERGENCY_MESSAGE;
0023 END_VAR
```

节点状态结构

2034

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

Emergency_Message 结构

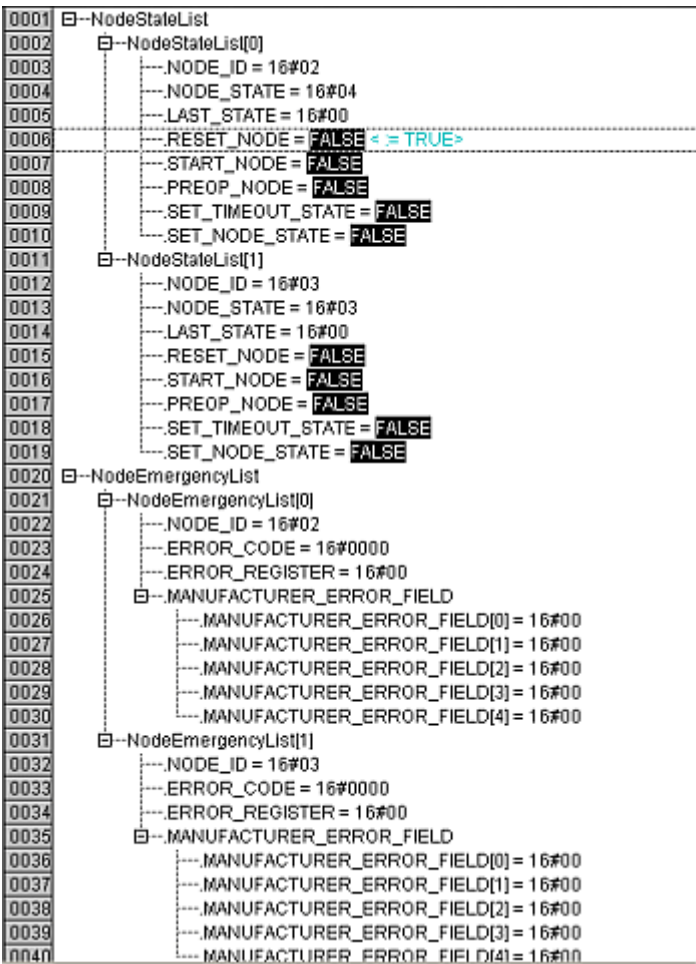
2035

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

访问应用程序运行时结构

2036

在运行时间内，您可通过库的全球变量访问相应的数组元件，进而可读取状态或 EMCY 消息，或复位节点。



在上述示例中，如果 ResetSingleNodeArray[0].RESET_NODE 短暂设为 TRUE，则在配置树中复位第一个节点。

① 关于可能的错误代码→ 系统手册“ecomatmobile 技巧”
→ 章节 CAN / CANopen: 错误和错误处理.

5.2.3 功能元件：CANopen 从站

内容	
CANx_SLAVE_EM CY_HANDLER	128
CANx_SLAVE_NODEID	130
CANx_SLAVE_SEND_EMERGENCY	131
CANx_SLAVE_STATUS	134

1874

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



CANx_SLAVE_EMCY_HANDLER

13199

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_CANopenSlave_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2053

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

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

► 在以下情况下调用功能块 :

- 错误状态要传送至 CAN 网络, 且

应用程序的错误消息要存储在对象目录中。

❗ 您是否想要将错误消息存储在对象目录中 ?

► (重复) 处理 后 **CANx_SLAVE_SEND_EMERGENCY** (→ 页 131) 之后, 调用 CANx_SLAVE_EMCY_HANDLER 一次 !

输入端参数

2054

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

输出端参数

2055

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



CANx_SLAVE_NODEID

13202

= CANx 从站节点 ID

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_CANopenSlave_Vxxyzz.LIB

CODESYS 中的符号 :



说明

2049

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

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

输入端参数

2047

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

CANx_SLAVE_SEND_EMERGENCY

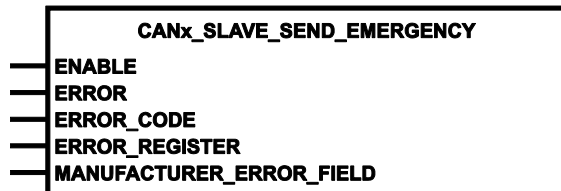
13205

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_CANOpenSlave_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2059

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

- 如果要将错误状态传送至网络中的其他装置, 则调用 FB。

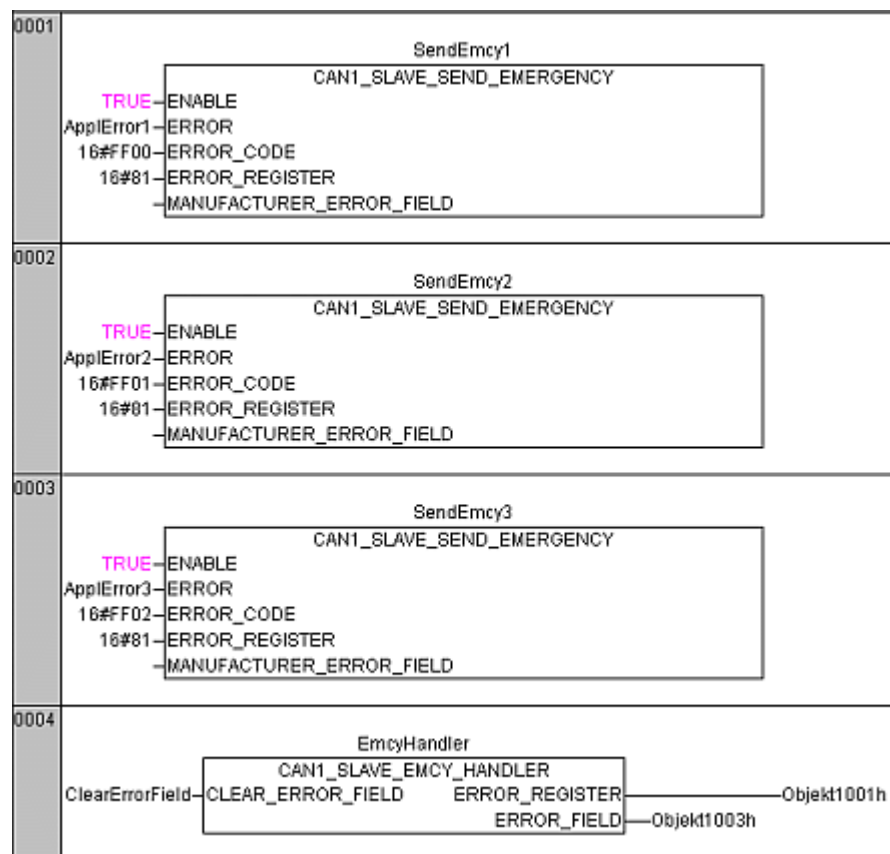
输入端参数

2060

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

例如：CANx_SLAVE_SEND_EMERGENCY

2062



在该示例中，随后将生成 3 个错误消息：

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

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

CANx_SLAVE_STATUS

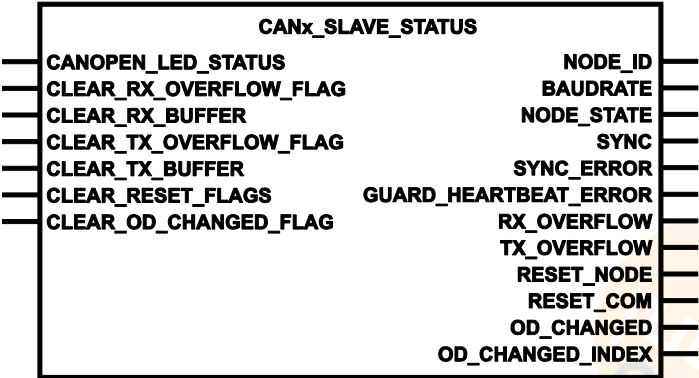
2063

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_CANopenSlave_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2066

CANx_SLAVE_STATUS 显示作为 CANopen 从站的装置的状态。FB 简化 CoDeSys CANopen 从站库的使用。强烈建议通过该功能块评估网络状态。

而后您可在运行时间内访问功能块的单个输出端, 以获取状态概述。

例如 :

```
0001 PROGRAM SlaveStatus
0002 VAR
0003   SlaveStatus: CR0505_SLAVE_STATUS;
0004   LedStatus: BOOL := TRUE;
0005   ClearRxOverflowFlag: BOOL;
0006   ClearRxBuffer: BOOL;
0007   ClearTxOverflowFlag: BOOL;
0008   ClearTxBuffer: BOOL;
0009   ClearResetFlags: BOOL;
0010   ClearOdChanged: BOOL;
0011   NodeId: BYTE;
0012   Baudrate: WORD;
0013   NodeState: BYTE;
0014   Sync: BOOL;
0015   SyncError: BOOL;
0016   GuardHeartbeatError: BOOL;
0017   RxOverflow: BOOL;
0018   TxOverflow: BOOL;
0019   ResetNode: BOOL;
0020   ResetCom: BOOL;
0021   OdChanged: BOOL;
0022   OdChangedIndex: INT;
0023 END_VAR
```

输入端参数

2067

参数	数据类型	说明
CANOPEN_LED_STATUS	BOOL	(输入端不可用于 PDM 装置) TRUE : 控制器状态 LED 切换至模式 "CANopen" : 闪烁频率 0.5 Hz = PRE-OPERATIONAL 闪烁频率 2.0 Hz = OPERATIONAL 该工作模式不改变其他诊断 LED 信号。
GLOBAL_START	BOOL	TRUE: 所有已连接的网络参与者 (从站) 在网络初始化期间同时启动 (⇒ 状态 OPERATIONAL) 。 FALSE: 已连接的网络参与者先后启动。
CLEAR_RX_OVERFLOW_FLAG	BOOL	FALSE ⇒ TRUE (edge): 清除错误标志 RX_OVERFLOW else: 未执行该功能
CLEAR_RX_BUFFER	BOOL	FALSE ⇒ TRUE (edge): 删除接收缓冲区中的数据 else: 未执行该功能
CLEAR_TX_OVERFLOW_FLAG	BOOL	FALSE ⇒ TRUE (edge): 清除错误标志 TX_OVERFLOW else: 未执行该功能
CLEAR_TX_BUFFER	BOOL	FALSE ⇒ TRUE (edge): 删除传送缓冲区中的数据 else: 未执行该功能
CLEAR_RESET_FLAGS	BOOL	FALSE ⇒ TRUE (edge): 清除标志 RESET_NODE 清除标志 RESET_COM else: 未执行该功能
CLEAR_OD_CHANGED_FLAGS	BOOL	FALSE ⇒ TRUE (edge): 清除标志 OD_CHANGED 清除标志 OD_CHANGED_INDEX else: 未执行该功能

输出端参数

2068

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

5.2.4 功能元件：CANopen SDO

内容	
CANx_SDO_READ	138
CANx_SDO_WRITE.....	140

2071

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



CANx_SDO_READ

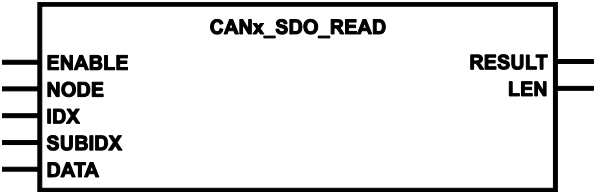
621

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxyxyz.LIB

CODESYS 中的符号 :



说明

624

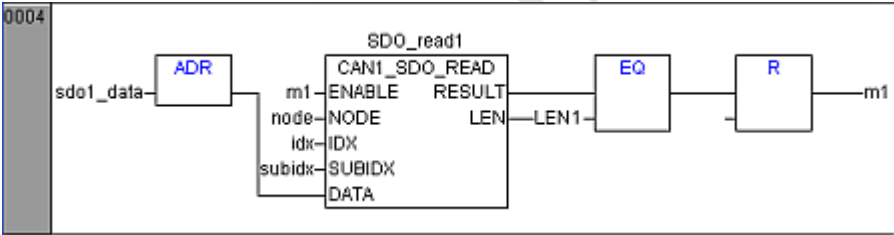
CANx_SDO_READ reads the →SDO (→ 页 297) , 所示索引来自节点。

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

可通过这些方法读取对象目录中的条目。 因此可选择性地读取节点参数。

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

例如 :



输入端参数

625

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

输出端参数

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

RESULT 的可能结果：

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

CANx_SDO_WRITE

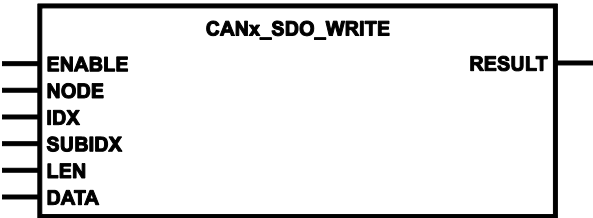
615

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号 :



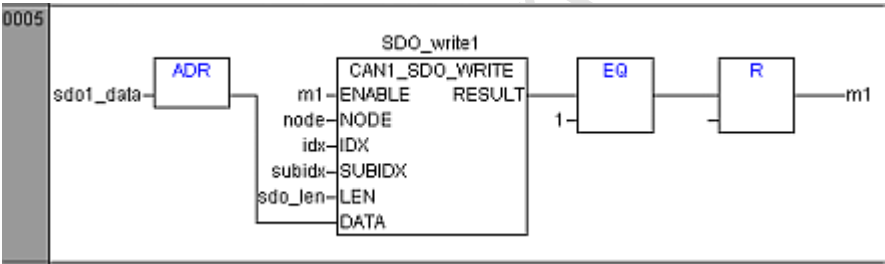
说明

618

CANx_SDO_WRITE 写入 →SDO (→ 页 297) , 索引指定给节点。
前提：节点必须处于 "PRE-OPERATIONAL" 或 "OPERATIONAL" 状态。
可使用该 FB 将条目写入对象目录。 因此可选择性地设定节点参数。

❗ LEN 的值必须低于传送数组的长度。 否则将发送随机数据。

例如 :



输入端参数

619

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

输出端参数

620

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

RESULT 的可能结果：

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

5.2.5 功能元件：SAE J1939

内容	
J1939_x	143
J1939_x_GLOBAL_REQUEST	144
J1939_x_RECEIVE	147
J1939_x_RESPONSE	151
J1939_x_SPECIFIC_REQUEST	154
J1939_x_TRANSMIT	157

2273

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

J1939_x

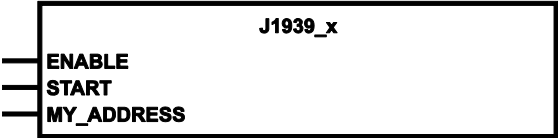
9375

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_x_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

4325

J1939_x 作为通信协议 SAE J1939 的协议处理器。

4313

⚠注意

(仅适用于到 v05 的 RTS)

经由第 1 个 CAN 接口的 J1939 通信 :

- ▶ 先初始化接口, 具体通过 : **CAN1_EXT**
(→ 页 [97](#))!

经由第 2 个 CAN 接口的 J1939 通信 :

- ▶ 先初始化接口, 具体通过 : **CAN2**
(→ 页 [104](#))!

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

如果输入端 START 针对一个周期设为 TRUE, 则启动协议处理器。

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

输入端参数

469

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

J1939_x_GLOBAL_REQUEST

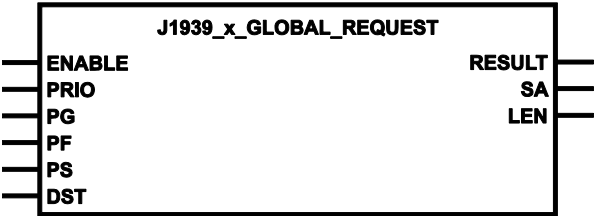
4315

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_x_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2301

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

信息

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

13790

通知

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

- 就每个请求的消息而言, 使用 FB 自身的实例 !
- ▶ 就目标地址而言, DST 适用 :
! 通过运算符 ADR 确定地址并将其分配至 FB !
 - ▶ 此外必须分配优先级 (一般为 3、6 或 7)。

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

- ▶ 鉴于数据请求可通过多个控制周期处理，该进程必须通过 RESULT 字节评估。
 - RESULT = 2: POU 正在等待参与者的数据。
 - RESULT = 1: 参与者接收了数据。
输出端 LEN 显示已接收的数据字节数量。
立即存储/评估该新数据！
接收新消息时，内存地址 DST 中的数据会被覆盖。
 - RESULT = 0: 1.25 秒内总线上无参与者发送回复。
FB 返回至非活跃状态。
仅现在 ENABLE 可再次设为 FALSE！
- ▶ 若要以短时间间隔从多个参与者接收数据：
在同一 PLC 周期中多次调用 POU 并立刻评估！

输入端参数

463

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

 信息

PGN = [Page] + [PF] + [PS]

PDU = [PRIO] + [PGN] + [J1939 address] + [data]

输出端参数

464

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

RESULT 的可能结果：

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

J1939_x_RECEIVE

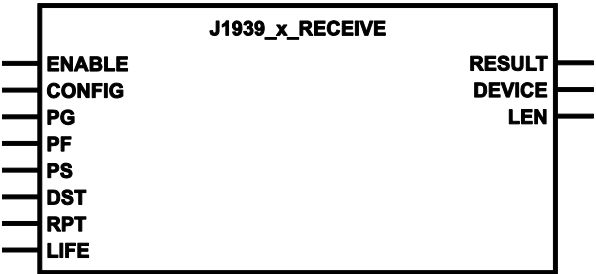
9393

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_x_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2288

J1939_x_RECEIVE 用于接收单个消息或消息块。
为此，必须通过输入端 CONFIG，针对一个周期初始化 FB。初始化期间，分配参数 PG、PF、PS、RPT、LIFE 以及数据组 DST 的内存地址。

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

13790

通知

存在非允许数据覆盖的风险！

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

就目标地址而言，DST 适用：

- !** 通过运算符 ADR 确定地址并将其分配至 FB！
- !** 一旦设定 RPT，则不可再修改！
- ▶ 必须通过 RESULT 字节评估数据的接收。如果 RESULT = 1，则可从通过 DST 分配的内存地址读取数据并进一步处理。
- > 接收新消息时，内存地址 DST 中的数据会被覆盖。
- > 已接收消息字节的数量通过输出端 LEN 显示。
- > 如果 RESULT = 3，则在所示时间窗口内未接收有效的消息 (LIFE • RPT)。

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



输入端参数

457

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

输出端参数

458

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

RESULT 的可能结果：

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

J1939_x_RESPONSE

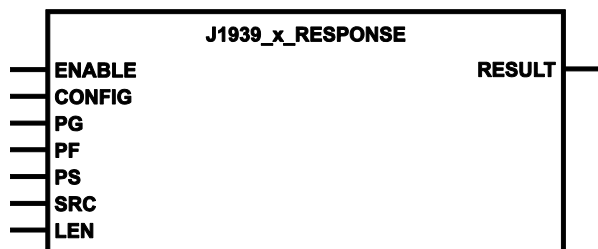
9399

$x = 1 \dots n$ = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_x_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2299

J1939_x_RESPONSE 处理针对请求消息的自动响应。

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

参数 PG、PF、PS、RPT 以及数据组 SRC 地址分配至 FB。

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

输入端参数

451

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CONFIG	BOOL	TRUE (第 1 个周期中): 配置数据对象 FALSE: 在程序进一步处理期间
PG	BYTE	数据页 定义的 PGN (参数组编号) 的值 允许 = 0...1 (正常情况下 = 0)
PF	BYTE	PDU 格式字节 定义的 PGN (参数组编号) 的值 PDU1 (特定) = 0...239 PDU2 (全局) = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN (参数组编号) 的值 如果 PF = PDU1 ⇒ PS = DA (目标地址) (DA = 外部装置的 J1939 地址) 如果 PF = PDU2 ⇒ PS = GE (群组扩展)
SRC	DWORD	源内存中的起始地址 ❗ 通过运算符 ADR 确定地址并将其分配至 FB !
LEN	WORD	待传送数据字节数量 (≥ 1)

输出端参数

13993

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

RESULT 的可能结果：

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

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

值 十进制 十六进制		说明
3	03	错误，数据不能传送



J1939_x_SPECIFIC_REQUEST

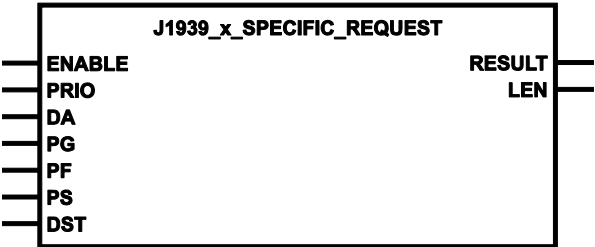
8884

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_x_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2300

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

 信息

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

13790

通知

存在非允许数据覆盖的风险 !

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

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

输入端参数

445

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
PRI0	BYTE	消息优先级 (0...7)
DA	BYTE	所需装置 J1939 地址
PG	BYTE	数据页 定义的 PGN (参数组编号) 的值 允许 = 0...1 (正常情况下 = 0)
PF	BYTE	PDU 格式字节 定义的 PGN (参数组编号) 的值 PDU1 (特定) = 0...239 PDU2 (全局) = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN (参数组编号) 的值 如果 PF = PDU1 ⇒ PS = DA (目标地址) (DA = 外部装置的 J1939 地址) 如果 PF = PDU2 ⇒ PS = GE (群组扩展)
DST	DWORD	目标地址  通过运算符 ADR 确定地址并将其分配至 FB !

 信息

PGN = [Page] + [PF] + [PS]

PDU = [PRI0] + [PGN] + [J1939 address] + [data]

输出端参数

446

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

RESULT 的可能结果：

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

J1939_x_TRANSMIT

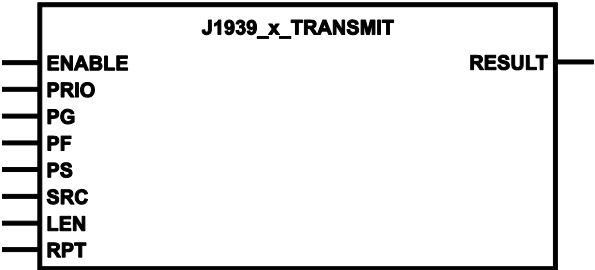
4322

x = 1...n = CAN 接口编号 (视装置而定, → 技术资料)

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_x_Vxxyyzz.LIB

CODESYS 中的符号 :



说明

2298

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

信息

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

- ▶ 就源地址而言, SRC 适用 :
 - ❗ 通过运算符 ADR 确定地址并将其分配至 FB !
- ▶ 此外, 还必须分配待传送数据字节的数量和优先级 (一般为 3、6 或 7) 。
- ▶ 鉴于数据传送通过多个控制周期处理, 进程必须通过 RESULT 字节评估。如果 RESULT = 1, 则所有数据均已传送。
 - ❗ 如果要发送的字节超过 8 个字节, 则进行“多包传输”。

输入端参数

439

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
PRI0	BYTE	消息优先级 (0...7)
PG	BYTE	数据页 定义的 PGN (参数组编号) 的值 允许 = 0...1 (正常情况下 = 0)
PF	BYTE	PDU 格式字节 定义的 PGN (参数组编号) 的值 PDU1 (特定) = 0...239 PDU2 (全局) = 240...255
PS	BYTE	PDU 特定字节 定义的 PGN (参数组编号) 的值 如果 PF = PDU1 ⇒ PS = DA (目标地址) (DA = 外部装置的 J1939 地址) 如果 PF = PDU2 ⇒ PS = GE (群组扩展)
SRC	DWORD	源内存中的起始地址 ! 通过运算符 ADR 确定地址并将其分配至 FB !
LEN	WORD	待传送数据字节数量 允许 = 1...1 785 = 0x0001...0x06F9
RPT	TIME	数据消息循环传送的重复时间 RPT = T#0s ⇒ 仅发送一次

信息

PGN = [Page] + [PF] + [PS]

PDU = [PRI0] + [PGN] + [J1939 address] + [data]

输出端参数

440

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

RESULT 的可能结果：

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

5.2.6 功能元件：串行接口

内容	
SERIAL_PENDING	161
SERIAL_RX	162
SERIAL_SETUP	164
SERIAL_TX	166

1600

ⓘ 注意

原则上，串行接口对用户不可用，因为它用于程序下载和调试。

如果用户将系统标志位 SERIAL_MODE 设为 TRUE，则可自由使用接口。但之后仅可通过 CAN 接口进行程序下载和调试。

可通过以下 FB 在应用程序中使用串行接口。

SERIAL_PENDING

314

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

317

SERIAL_PENDING 确定存储在序列接收缓冲区的数据字节的数量。
不同于 SERIAL_RX (→ 页 162) , 调用该 FB 后缓冲区的内容保持不变。
SERIAL FB 是创建串行接口应用程序特定协议的基础。
为此, 需设定系统标志位 SERIAL_MODE=TRUE !

ⓘ 注意

原则上, 串行接口对用户不可用, 因为它用于程序下载和调试。
如果用户将系统标志位 SERIAL_MODE 设为 TRUE, 则可自由使用接口。 但之后仅可通过 CAN 接口进行程序下载和调试。

输出端参数

319

参数	数据类型	说明
NUMBER	WORD	已接收字节数量

SERIAL_RX

308

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

311

SERIAL_RX 读取每次调用时来自序列接收缓冲区的已接收数据字节。
而后 AVAILABLE 的值减少 1。
如果接收 1000 以上数据字节，则缓冲区溢出，且数据丢失。 该情况通过位 OVERFLOW 显示。
如果采用 7 位数据传输，则第 8 位包含奇偶校验，必要时用户应加以阻止。
SERIAL FB 是创建串行接口应用程序特定协议的基础。
为此，需设定系统标志位 SERIAL_MODE=TRUE ！

①注意

原则上，串行接口对用户不可用，因为它用于程序下载和调试。
如果用户将系统标志位 SERIAL_MODE 设为 TRUE，则可自由使用接口。 但之后仅可通过 CAN 接口进行程序下载和调试。

输入端参数

312

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

输出端参数

313

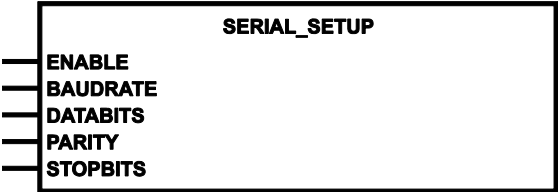
参数	数据类型	说明
Rx	BYTE	从接收缓冲区接收的字节数据
AVAILABLE	WORD	剩余数据字节数量 0 = 无可用的有效数据
OVERFLOW	BOOL	TRUE: 数据缓冲区溢出 ⇒ 数据丢失 ! FALSE: 数据缓冲区无数据丢失

SERIAL_SETUP

302

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

305

SERIAL_SETUP 初始化 RS232 串行接口。
不一定需要执行功能块才能够使用串行接口。若未调用功能块，则以下默认设定适用。
功能块利用针对一个周期的 ENABLE=TRUE，将串行接口设为所示的参数。在功能块的帮助下进行的更改将稳定保存。

ⓘ 注意

原则上，串行接口对用户不可用，因为它用于程序下载和调试。
如果用户将系统标志位 SERIAL_MODE 设为 TRUE，则可自由使用接口。但之后仅可通过 CAN 接口进行程序下载和调试。

5020

通知

串行接口的驱动器模块可能会受损！
运行时断开或连接串行接口可能导致不明确的状态并损坏驱动器模块。
► 切勿在运行时断开或连接串行接口。

输入端参数

6

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

SERIAL_TX

296

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

299

SERIAL_TX 通过 RS232 串行接口传送一个数据字节。
可利用输入端 ENABLE 启用或阻止传送。
SERIAL FB 是创建串行接口应用程序特定协议的基础。
为此，需设定系统标志位 SERIAL_MODE=TRUE ！

ⓘ 注意

原则上，串行接口对用户不可用，因为它用于程序下载和调试。
如果用户将系统标志位 SERIAL_MODE 设为 TRUE，则可自由使用接口。但之后仅可通过 CAN 接口进行程序下载和调试。

输入端参数

300

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

5.2.7 功能元件：通过处理中断优化 PLC 周期

内容	
SET_INTERRUPT_I	168
SET_INTERRUPT_XMS	171
	0965
	8609

您可在此了解优化 PLC 周期的功能。

1599

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

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

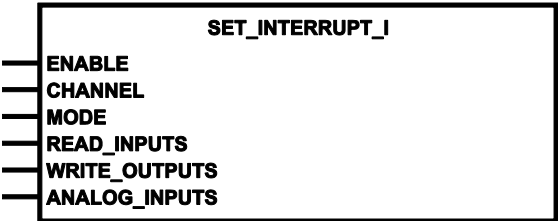
因为 SafetyController 大部分不允许中断功能，因此不可用。

SET_INTERRUPT_I

2381

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

281
11573

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

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

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

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

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

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

ⓘ 注意

应在周期中跳过程序块（安装调用除外），以便亦不会循环调用。

不可在中断程序中初始化和进一步处理监控中断触发的输入端 (CHANNEL)。

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

用户负责主程序和在中断模式下运行的程序部分之间的数据一致性！

9866

中断优先级：

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

971

ⓘ注意

中断程序将影响周期中输入端和输出端的独特性。 因此仅服务部分输入端和输出端。 如果在中断程序中初始化，则将读取或写入以下输入端和输出端。

输入端，数字：

%IX0.0...%IX0.7 (Controller: CR0n3n, CR7n3n)
%IX0.12...%IX0.15、%IX1.4...%IX1.8（所有其他 ClassicController、ExtendedController、SafetyController）
%IX0.0, %IX0.8 (SmartController: CR250n)
IN08...IN11 (CabinetController: CR030n)
IN0...IN3 (PCB 控制器：CS0015)

输入端，模拟：

%IX0.0...%IX0.7 (Controller: CR0n3n, CR7n3n)
所有通道（选择位编码）（所有其他控制器）

输出端，数字：

%QX0.0...%QX0.7 (ClassicController, ExtendedController, SafetyController)
%QX0.0, %QX0.8 (SafetyController: CR7nnn)
OUT00...OUT03 (CabinetController: CR030n)
OUT0...OUT7 (PCB 控制器：CS0015)

如果中断程序在周期中同时访问，则全局变量也不再具有独特性。 较大数量类型（如 DINT）尤其会出现这种问题。

所有其他输入端和输出端照例在周期中处理一次。

输入端参数

2383

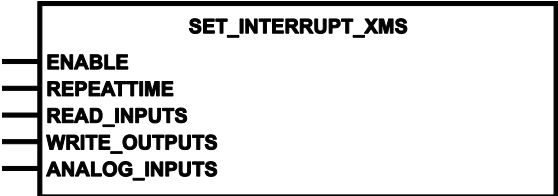
参数	数据类型	说明
ENABLE	BOOL	TRUE (仅针对 1 个周期) : 功能块初始化 FALSE: 单元未执行
CHANNEL	BYTE	中断输入端编号 输入端 IN00...IN07 对应 0...7
MODE	BYTE	触发中断的输入端 CHANNEL 边沿类型 1 = 上升沿 (标准值) 2 = 下降沿 3 = 上升沿和下降沿 > 3 = 标准值
READ_INPUTS	BOOL	TRUE : 先读取输入端 0...7, 再调用程序 然后写入至输入端标志 I00...I07 FALSE : 仅读取 CHANNEL 之下所示的通道 并写入至相应的输入端标志 Ixx
WRITE_OUTPUTS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

SET_INTERRUPT_XMS

272

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

275

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

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

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

还可按照固定时间间隔（每 x ms，还可按照固定时间间隔（每 x ms）调用特定程序部分）调用特定程序部分，且不考虑控制周期。

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

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

①注意

为避免中断程序调用的程序块被额外循环调用，应在周期中跳过（初始化调用除外）。

可启用多个计时器中断程序块。必须计算中断功能的时间要求，以便执行所有调用的功能。这尤其适用于计算、浮点算法或控制器功能。

用户负责主程序和在中断模式下运行的程序部分之间的数据一致性！

请注意：如果是高 CAN 总线活动，设定的 REPEATTIME 会波动。

①注意

中断程序将影响周期中输入端和输出端的独特性。 因此仅服务部分输入端和输出端。 如果在中断程序中初始化，则将读取或写入以下输入端和输出端。

输入端，数字：

%IX0.0...%IX0.7 (Controller: CR0n3n, CR7n3n)
%IX0.12...%IX0.15、%IX1.4...%IX1.8 (所有其他 ClassicController、ExtendedController、SafetyController)
%IX0.0, %IX0.8 (SmartController: CR250n)
IN08...IN11 (CabinetController: CR030n)
IN0...IN3 (PCB 控制器： CS0015)

输入端，模拟：

%IX0.0...%IX0.7 (Controller: CR0n3n, CR7n3n)
所有通道（选择位编码）（所有其他控制器）

输出端，数字：

%QX0.0...%QX0.7 (ClassicController, ExtendedController, SafetyController)
%QX0.0, %QX0.8 (SafetyController: CR7nnn)
OUT00...OUT03 (CabinetController: CR030n)
OUT0...OUT7 (PCB 控制器： CS0015)

如果中断程序在周期中同时访问，则全局变量也不再具有独特性。 较大数量类型（如 DINT）尤其会出现这种问题。

所有其他输入端和输出端照例在周期中处理一次。

输入端参数

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

5.2.8 功能元件：处理输入值

内容	
ANALOG_RAW	174
INPUT_ANALOG.....	175
INPUT_CURRENT	178
INPUT_VOLTAGE.....	179
	1602
	1302

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

①注意

CODESYS PLC 配置所述的模拟原始值直接来自 ADC。它们还未进行修正！
因此，针对同一装置的 PLC 配置中会显示不同原始值。
仅 IFM 功能块可进行错误修正和正常化。功能块提供修正的值。

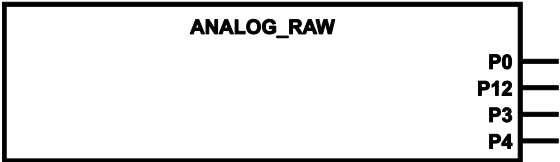
ANALOG_RAW

19589

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

9918

ANALOG_RAW 提供未经筛选的输入端原始模拟信号。

输出端参数

13148

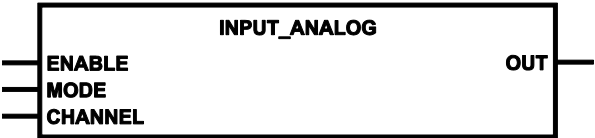
参数	数据类型	说明
P0	ARRAY [0..7] OF WORD	模拟输入端口 0 原始输入值： P0.0 , I00 ... P0.7 , I07
P12	ARRAY [0..7] OF WORD	模拟输入端口 1+2 原始输入值： P12.0 , I14 ... P12.3 , I17 P12.4 , I24 ... P12.7 , I27
P3	ARRAY [0..7] OF WORD	模拟输入端口 3 原始输入值： P3.0 , I30 ... P3.7 , I37
P4	ARRAY [0..7] OF WORD	模拟输入端口 4 原始输入值： P4.0 , I40 ... P4.7 , I47

INPUT_ANALOG

519

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

522

INPUT_ANALOG 有助于进行模拟通道的电流和电压测量。
FB 提供所选模拟通道的当前模拟值。 测量值和输出值源自通过 MODE 指定的工作模式

MODE	输入端工作模式	输出端 OUT	单位
IN_DIGITAL_H	数字输入	0 / 1	---
IN_CURRENT	电流输入	0...20 000	µA
IN_VOLTAGE10	电压输入	0...10 000	mV
IN_VOLTAGE30	电压输入	0...32 000	mV
IN_RATIO	电压输入比率	0...1 000	‰

就工作模式参数设定而言，应使用所示的全局系统变量。 模拟值作为标准值提供。

❗使用该 FB 时，您必须设定系统变量 RELAIS *)。
否则，测量电流时会缺失内部参考电压。

*) Relay exists only in the following devices: CR0020, CRnn32, CRnn33, CR0200, CR0505, CR7nnn

输入端参数

523

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
MODE	BYTE	IN_DIGITAL_H 数字输入 IN_CURRENT 电流输入 0...20 000 µA IN_VOLTAGE10 电压输入 0...10 000 mV IN_VOLTAGE30 电压输入 0...32 000 mV IN_RATIO 比率模拟输入端

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

参数	数据类型	说明
INPUT_CHANNEL	BYTE	输入通道编号 允许 = 0...7



输出端参数

524

参数	数据类型	说明
OUT	WORD	基于 MODE 的输出值 设定无效时：OUT = "0"



INPUT_CURRENT

513

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

516

INPUT_CURRENT 返回模拟电流输入端的实际输入电流（以 [μA] 为单位）

信息

INPUT_CURRENT 是兼容较旧程序的 FB。在新程序中，应使用更强大的 **INPUT_ANALOG** (→ 页 175)。

输入端参数

517

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
INPUT_CHANNEL	BYTE	输入通道编号 允许 = 0...7

输出端参数

518

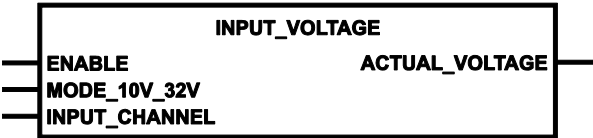
参数	数据类型	说明
ACTUAL_CURRENT	WORD	输出电流，以 [μA] 为单位

INPUT_VOLTAGE

507

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

510

INPUT_VOLTAGE 处理模拟通道上测量的模拟电压。

> FB 将返回所选模拟通道上的当前输入电压（以 [mV] 为单位）。测量适用于通过 MODE_10V_32V（10 000 mV 或 32 000 mV）定义的电压范围。

信息

INPUT_VOLTAGE 是兼容较旧程序的 FB。在新程序中，应使用更强大的 **INPUT_ANALOG** (→ 页 175)。

输入端参数

511

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
MODE_10V_32V	BOOL	TRUE : 电压范围 0...32 V FALSE : 电压范围 0...10 V
INPUT_CHANNEL	BYTE	输入通道编号 允许 = 0...7

输出端参数

512

参数	数据类型	说明
ACTUAL_VOLTAGE	WORD	输入电压，以 [mV] 为单位

5.2.9 功能元件：调整模拟值

内容	
NORM.....	181

1603

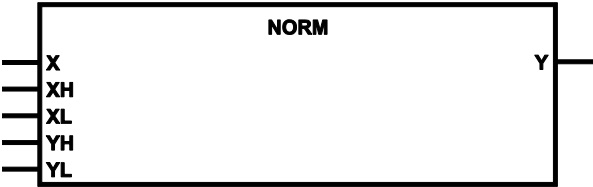
如果调整模拟输入值或模拟功能的结果，则以下 FB 可帮到您。

NORM

401

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxyyyzz.LIB

CODESYS 中的符号：



说明

404

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

ⓘ 注意

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

输入端参数

405

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

输出端参数

406

参数	数据类型	说明
Y	WORD	输出值

例如：NORM (1)

407

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

而后 FB 转换输入信号，如下所示：

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

例如：NORM (2)

408

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

而后 FB 转换输入信号，如下所示：

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

5.2.10 功能元件：针对频率和周期测量的计数器功能

内容

FAST_COUNT	184
FREQUENCY	186
INC_ENCODER	188
PERIOD	192
PERIOD_RATIO	194
PHASE	196

18818

控制器支持高达 4 个快速输入端，可处理高达 30 kHz 的输入频率。除了频率测量之外，FRQ 输入端还可用于评估增量编码器（计数器功能）。

因为测量方法不同，确定频率时可能会出现错误。

以下 FB 适用于简单评估：

功能元件	允许值	说明
FAST_COUNT	0...30 000 Hz	快速脉冲计数
FREQUENCY	0.1...30 000 Hz	测量所示通道上的频率。频率高时测量错误减少
INC_ENCODER	0...30 000 Hz	针对编码器评估的递增/递减计数器功能
PERIOD	0...5 000 Hz	测量所示通道上的频率和周期（周期时间）
PERIOD_RATIO	0...5 000 Hz	测量所示通道上的频率和周期（周期时间）以及传号空号比 [%]
PHASE	0...5 000 Hz	读取通道对对比信号的相位

❗ 将快速输入端作为“正常”数字输入端时的重要事项：

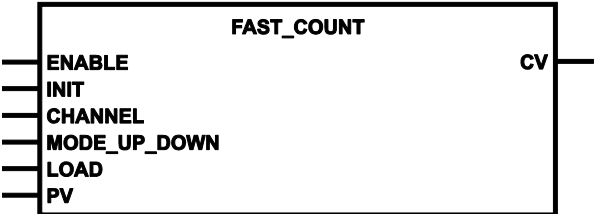
- ▶ 必须考虑提高的噪声脉冲敏感度（例如机械触点的触点弹跳）。
- 标准数字输入端可评估高达 50 Hz 的信号。

FAST_COUNT

20430

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxyxyz.LIB

CODESYS 中的符号：



说明

570

FAST_COUNT 作为快速输入脉冲的计数器块运行。
该 FB 检测 FRQ 输入通道 0...3 的快速脉冲。有了 FRQ 输入通道 0，FAST_COUNT 跟功能块 CTU 一样运行。最大输入频率 → 技术资料。

❗ 鉴于技术设计的原因，ecomatmobile 控制器通道 0 仅可作为递增计数器。通道 1...3 可作为递增和递减计数器。

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

20433

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 计数器停止
INIT	BOOL	FALSE ⇒ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	快速输入通道编号 输入端 I14...I17 对应 0...3
MODE_UP_DOWN	BOOL	TRUE : 计数器向下计数 FALSE : 计数器向上计数
LOAD	BOOL	TRUE : 起始值在 CV 中加载 FALSE: function element is not executed
PV	DWORD	计数器初始值 (预设值)

输出端参数

572

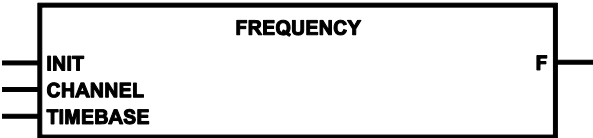
参数	数据类型	说明
CV	DWORD	当前计数值 过溢时的状况： • 向下计数时计数器停止在 0 的位置 • 向上计数时出现过溢

FREQUENCY

20604

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

540

FREQUENCY 测量所示通道的信号频率。 最大输入频率 → 技术资料。

该 FB 测量所选 CHANNEL 的信号频率。 为此，需评估上升沿。 可在广阔的数值范围内测量频率，具体视 TIMEBASE 而定。 高频率需要短时基，而低频率相应地就需要较长的时基。 频率直接以 [Hz] 为单位显示。

❗就 FREQUENCY 而言，仅可使用输入端 FRQ0...FRQ3。

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

20610

参数	数据类型	说明
INIT	BOOL	FALSE ⇒ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	快速输入通道编号 输入端 I14...I17 对应 0...3
TIMEBASE	TIME	频率测量时基 (最多 57 s)

8406

❗ 初始化之前 FB 可能提供错误值。

► 切勿在初始化 FB 之前评估输出端。

强烈建议针对待评估的各个通道，对该 FB 的自身实例进行编程。否则可能会提供错误的值。

输出端参数

542

参数	数据类型	说明
F	REAL	输入信号的频率，以 [Hz] 为单位

INC_ENCODER

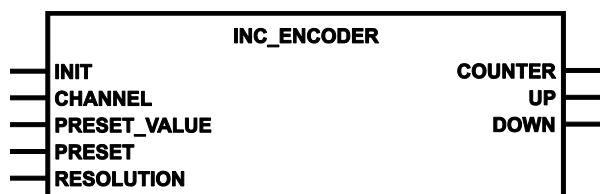
20432

= 增量编码器

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxyyzz.LIB

CODESYS 中的符号：



说明

4330

INC_ENCODER 提供针对编码器评估的递增/递减计数器功能。

每个通过功能块评估的输入对通过两个频率输入端形成。

限制频率 = 30 kHz

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

设定预设值：

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

只要 INIT=FALSE 且 PRESET=FALSE，功能块即对输入端脉冲进行计数。

电流计数值位于输出端 COUNTER。

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

❗切勿搭配使用一个输入端上的该功能块与以下功能块之一！

- **FAST_COUNT** (→ 页 [184](#))
- **FREQUENCY** (→ 页 [186](#))
- **PERIOD** (→ 页 [192](#))
- **PERIOD_RATIO** (→ 页 [194](#))
- **PHASE** (→ 页 [196](#))

在输入端 RESOLUTION，编码器分辨率可成倍评估：

- 1 = 普通分辨率 (跟编码器分辨率相同)，
- 2 = 双倍分辨率评估，

4 = 4 倍分辨率评估。

该输入端的所有其他值表示普通分辨率。

	RESOLUTION = 1 如果为普通分辨率，则仅评估 B 信号的下降沿。
	RESOLUTION = 2 如果为双倍分辨率，则评估 B 信号的上升沿和下降沿。
	RESOLUTION = 4 如果为 4 倍分辨率，则评估 A 信号和 B 信号的上升沿和下降沿。

输入端参数

20438

参数	数据类型	说明
INIT	BOOL	TRUE (仅针对 1 个周期) : 功能块初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	输入通道对编号 0 = 通道对 0 = 输入端 I14 + I15 1 = 通道对 1 = 输入端 I16 + I17 2 = 通道对 2 = 输入端 I24 + I25 3 = 通道对 3 = 输入端 I26 + I27
PRESET_VALUE	DINT	计数器初始值
PRESET	BOOL	FALSE ⇔ TRUE (edge): PRESET_VALUE 加载至 COUNTER TRUE : 计数器忽略输入脉冲 FALSE : 计数器计算输入脉冲
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

530

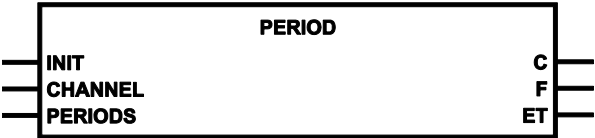
参数	数据类型	说明
COUNTER	DINT	当前计数值
UP	BOOL	TRUE : 计数器在最后一个周期中向上计数 FALSE : 计数器在最后一个周期中未向上计数
DOWN	BOOL	TRUE : 计数器在最后一个周期中向下计数 FALSE : 计数器在最后一个周期中未向下计数

PERIOD

20606

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

373

PERIOD 测量所示通道频率和以 [μs] 为单位的周期（周期时间）。最大输入频率 → 技术资料。

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

如果是低频率，则在使用 FREQUENCY 时会出现误差：为避免这种情况，可使用 PERIOD。周期时间直接以 [μs] 为单位显示。

最大测量范围大约为 71 min。

⚠注意

就 PERIOD 而言，仅可使用输入端 CYL0...CYL3。
针对 PDM360smart: CR1071: 所有输入端。
不再注明 < 0.5 Hz 的频率！

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

20608

参数	数据类型	说明
INIT	BOOL	FALSE ⇔ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	快速输入通道编号 输入端 I24...I27 对应 0...3
PERIODS	BYTE	待对比周期数

8406

❗ 初始化之前 FB 可能提供错误值。

► 切勿在初始化 FB 之前评估输出端。

强烈建议针对待评估的各个通道，对该 FB 的自身实例进行编程。否则可能会提供错误的值。

输出端参数

375

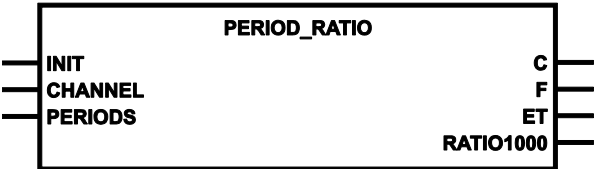
参数	数据类型	说明
C	DWORD	已检查周期的周期时间 [μs] 允许 = 200...10 000 000 = 0xC8...0x989680 (= 10s)
F	REAL	输入信号的频率，以 [Hz] 为单位
ET	TIME	自输入端最后一个上升边沿以来消耗的时间 (可用于非常慢的信号)

PERIOD_RATIO

20441

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

367

PERIOD_RATIO 测量所示周期内所示通道频率和以 [μs] 为单位的周期（周期时间）。此外，传号周期比以 [%] 为单位显示。最大输入频率 → 技术资料。

该 FB 测量所选 CHANNEL 信号的频率和周期时间。若要计算，则评估所有上升沿并利用所示 PERIODS 的数字确定平均值。此外，传号周期比以 [%] 为单位显示。

例如：如果是 25 ms 高电平和 75 ms 低电平的信号比，则值 ATIO1000 为 250 %。
如果是低频率，则在使用 FREQUENCY 时会出现误差：为避免这种情况，可使用 PERIOD_RATIO。周期时间直接以 [μs] 为单位显示。

最大测量范围大约为 71 min。

ⓘ 注意

就 PERIOD_RATIO 而言，仅可使用输入端 CYL0...CYL3。
针对 PDM360smart: CR1071: 所有输入端。
输出端 RATIO1000 针对 100 % 的传号周期比提供值 0（电源电压位置始终有输入信号）。
不再注明 < 0.05 Hz 的频率！

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

20446

参数	数据类型	说明
INIT	BOOL	FALSE ⇔ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	快速输入通道编号 输入端 I24...I27 对应 0...3
PERIODS	BYTE	待对比周期数

8406

❗ 初始化之前 FB 可能提供错误值。

► 切勿在初始化 FB 之前评估输出端。

强烈建议针对待评估的各个通道，对该 FB 的自身实例进行编程。否则可能会提供错误的值。

输出端参数

369

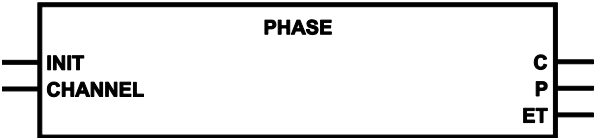
参数	数据类型	说明
C	DWORD	已检查周期的周期时间 [μs] 允许 = 200...10 000 000 = 0xC8...0x989680 (= 10s)
F	REAL	输入信号的频率，以 [Hz] 为单位
ET	TIME	自输入端最后状态更改以来消耗的时间（可用于信号非常缓慢的情况下）
ATIO1000	WORD	传号空号比，以 [%] 为单位 允许 = 1...999 = 0x1...0x3E7 前提： • 针对间隔测量 • 脉冲持续时间 ≥ 100 μs • 频率 < 5 kHz

PHASE

20443

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

361

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

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

❗就低于 15 Hz 的频率而言，显示 0 周期或相移。

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

20444

参数	数据类型	说明
INIT	BOOL	FALSE ⇔ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
CHANNEL	BYTE	输入通道对编号 0 = 通道对 0 = 输入端 I14 + I15 1 = 通道对 1 = 输入端 I16 + I17

8406



初始化之前 FB 可能提供错误值。

▶ 切勿在初始化 FB 之前评估输出端。

强烈建议针对待评估的各个通道，对该 FB 的自身实例进行编程。 否则可能会提供错误的值。

输出端参数

363

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

5.2.11 功能元件：PWM 功能

内容	
OCC_TASK	199
OUTPUT_CURRENT	201
OUTPUT_CURRENT_CONTROL	202
PWM	204
PWM100	208
PWM1000	210

13758

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

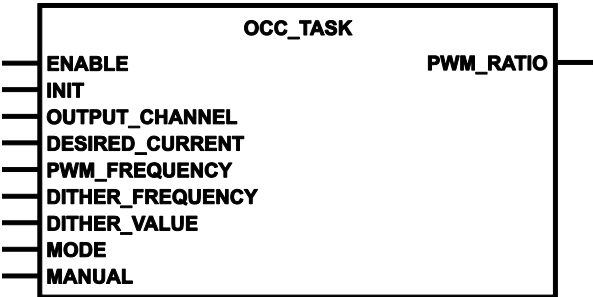


OCC_TASK

20619

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

391

OCC_TASK 作为 PWM 输出端的电流控制器运行。

控制器设计为自适应控制器以便进行自我优化。 如果无需自我优化性能，则可通过输入端 MANUAL 传送 > 0 的值（自我优化性能禁用）。 数值代表补偿值，将影响控制器的积分分量和微分分量。 若要确定 MANUAL 模式下的最佳控制器设定，则可使用值 50。 根据所需的控制器特性，之后可逐步增加（控制器变得更强/更快）或减少（控制器变得更弱/更慢）值。

如果输入端 MANUAL 设为 0，则控制器始终自我优化。 控制系统的性能永久受监控，且在每个周期中，更新的补偿值自动永久存储。 控制系统的更改会即刻加以识别和修正。

ⓘ 注意

OCC_TASK 以 5 ms 的固定周期时间工作。 无需输入实际值，因为 FB 会内部检测。

OCC_TASK 基于 PWM (→ 页 204).

如果 OUTPUT_CURRENT_CONTROL 用于输出端 4...7,则仅可在此使用 PWM FB，前提是同时使用 PWM 输出端 8...11。

► 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...100 % 之间：

- PWM 比率 + DITHER_VALUE < 100 且
- PWM 比率 - DITHER_VALUE > 0 。

输入端参数

20620

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
INIT	BOOL	FALSE ⇔ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
OUTPUT_CHANNEL	BYTE	电流控制输出通道 (0...7) 编号 输出端 Q10...Q13 对应 0...3 输出端 Q20...Q23 对应 4...7
DESIRED_CURRENT	WORD	输出端所需电流值, 以 [mA] 为单位
PWM_FREQUENCY	WORD	输入端负载 PWM 频率 [Hz]
		抖动频率, 以 [Hz] 为单位 值范围 = $0 \dots \text{FREQUENCY} / 2$ $\text{FREQUENCY} / \text{DITHER_FREQUENCY}$ 必须为偶数! FB 可将所有其他至增加到下一个匹配值。
DITHER_VALUE	BYTE	抖动峰间值, 以 [%] 为单位 允许值 = $0 \dots 100 = 0x00 \dots 0x64$
MODE	BYTE	控制器特性: 0 = 增加非常缓慢, 无过冲 1 = 增加缓慢, 无过冲 2 = 最低过冲 3 = 允许中等过冲
MANUAL	BYTE	值 = 0: 控制器按照自我优化的方式运行 值 > 0: 闭环回路控制器的自我优化性能将被覆盖 (一般: 50)

输出端参数

393

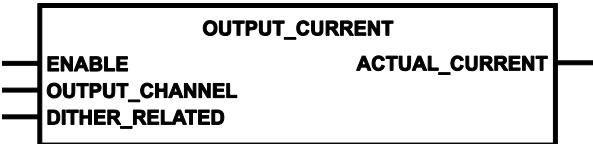
参数	数据类型	说明
PWM_RATIO	BYTE	针对监控目的: 显示 PWM 脉冲比 0...100%

OUTPUT_CURRENT

20449

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

385

OUTPUT_CURRENT 处理协同有效 PWM 通道的电流测量。

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

输入端参数

20451

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
OUTPUT_CHANNEL	BYTE	电流控制输出通道 (0...7) 编号 输出端 Q10...Q13 对应 0...3 输出端 Q20...Q23 对应 4...7
DITHER_RELATED	BOOL	通过以下方式确定电流的平均值... TRUE : 一个抖动周期 FALSE : 一个 PWM 周期

输出端参数

387

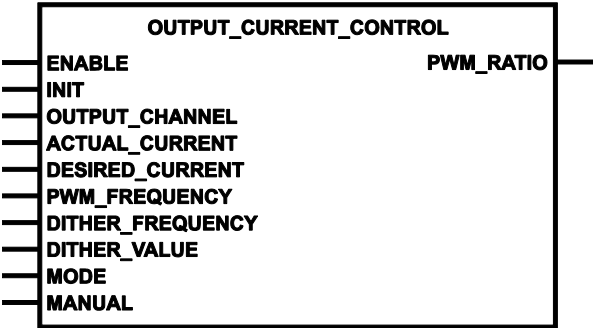
参数	数据类型	说明
ACTUAL_CURRENT	WORD	输出电流，以 [mA] 单位

OUTPUT_CURRENT_CONTROL

20453

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

20454

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

控制器设计为自适应控制器以便进行自我优化。 如果无需该自我优化性能，则可通过输入端 MANUAL 传送 > 0 的值；而后自我优化性能禁用。 数值代表补偿值，将影响控制器的积分分量和微分分量。

若要确定 MANUAL 模式下的最佳控制器设定，则可使用值 50。 根据所需的控制器特性，之后可逐步增加（控制器变得更强/更快）或减少（控制器变得更弱/更慢）值。

如果输入端 MANUAL 设为 0，则控制器始终自我优化。 控制系统的性能永久受监控，且在每个周期中，更新的补偿值自动永久存储。 控制系统的更改会即刻加以识别和修正。

ⓘ 注意

若要获得稳定的输出值，则需定期循环调用 OUTPUT_CURRENT_CONTROL。

OUTPUT_CURRENT_CONTROL 基于 PWM (→ 页 204)。

如果 OUTPUT_CURRENT_CONTROL 用于输出端 4...7,则仅可在此使用 PWM FB，前提是同时使用 PWM 输出端 8...11。

► 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...100 % 之间：

- PWM 比率 + DITHER_VALUE < 100 且
- PWM 比率 - DITHER_VALUE > 0 。

输入端参数

20455

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
INIT	BOOL	FALSE $\Rightarrow$ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
OUTPUT_CHANNEL	BYTE	电流控制输出通道 (0...7) 编号 输出端 Q10...Q13 对应 0...3 输出端 Q20...Q23 对应 4...7
ACTUAL_CURRENT	WORD	PWM 输出端实际电流, 以 [mA] 单位 ► 将 OUTPUT_CURRENT ($\rightarrow$ 页 201) 的输出值传输至输入端 ACTUAL_CURRENT。
DESIRED_CURRENT	WORD	输出端所需电流值, 以 [mA] 为单位
PWM_FREQUENCY	WORD	输入端负载 PWM 频率 [Hz]
		抖动频率, 以 [Hz] 为单位 值范围 = $0 \dots \text{FREQUENCY} / 2$ $\text{FREQUENCY} / \text{DITHER_FREQUENCY}$ 必须为偶数! FB 可将所有其他至增加到下一个匹配值。
DITHER_VALUE	BYTE	抖动峰间值, 以 [%] 为单位 允许值 = $0 \dots 100 = 0x00 \dots 0x64$
MODE	BYTE	控制器特性: 0 = 增加非常缓慢, 无过冲 1 = 增加缓慢, 无过冲 2 = 最低过冲 3 = 允许中等过冲
MANUAL	BYTE	值 = 0: 控制器按照自我优化的方式运行 值 > 0: 闭合回路控制器的自我优化性能将被覆盖 (一般: 50)

输出端参数

381

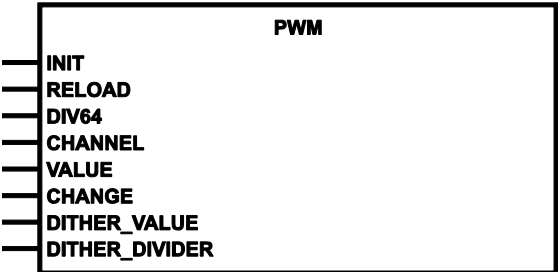
参数	数据类型	说明
PWM_RATIO	BYTE	针对监控目的: 显示 PWM 脉冲比 0...100%

PWM

20457

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

20467

PWM 用于 PWM 输出端的初始化和参数设定。

PWM 有更具技术性的背景。因为其结构的原因，PWM 值可精细分级。因此该 FB 适于控制器用途。

应用程序初始化期间，每个通道调用一次 PWM。此时，输入端 INIT 必须设为 TRUE。初始化期间还分配参数 RELOAD。

❗注意

通道 4...11 的值 RELOAD 必须相同。

就这些通道而言，PWM 和 **PWM1000** (→ 页 [210](#))不得混淆。

PWM 频率 (以及 RELOAD 值) 内部限定为 5 kHz。

输入端 DIV64 必须设为 FALSE (0) 或 TRUE (1)，具体看需要高 PWM 频率还是低 PWM 频率。

在程序循环处理期间，INIT 设为 FALSE。FB 调用并分配新的 PWM 值。如果输入端 CHANGE = TRUE，则采用值。

可实施已初始化 PWM 通道的电流测量：

- 通过 **OUTPUT_CURRENT** (→ 页 [201](#))
- 或例如使用 **IFM** 装置 EC2049 (电流测量系列元件)。

应用程序初始化期间，每个通道调用一次 PWM_Dither。此时，输入端 INIT 必须设为 TRUE。初始化期间，分配确定抖动频率和 VALUE 的 DIVIDER。

❗参数 DITHER_FREQUENCY 和 DITHER_VALUE 可针对每个通道单独设定。

输入端参数

20458

参数	数据类型	说明
INIT	BOOL	FALSE $\Rightarrow$ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
RELOAD	WORD	确定 PWM 频率的值 ($\rightarrow$ 章节 计算 RELOAD 值 ($\rightarrow$ 页 206))
DIV64	BOOL	CPU 周期/ 64
CHANNEL	BYTE	PWM 输出通道编号 输出端 Q10...Q13 对应 0...3 输出端 Q20...Q23 对应 4...7 输出端 Q40...Q47 对应 8...15
VALUE	WORD	当前 PWM 值 允许 = 0...RELOAD 0 = 开启时间 100 % RELOAD = 开启时间 0 %
CHANGE	BOOL	TRUE: 采用以下位置的新值: • VALUE: 当前 PWM 周期结束后 • DITHER_VALUE: 当前抖动周期结束后 FALSE: 更改的 PWM 值对输出端无影响
DITHER_VALUE	WORD	抖动峰间值, 以 [%] 为单位 允许值 = 0...1 000 = 0000...03E8
DITHER_DIVIDER	WORD	抖动频率 = PWM 频率/ DIVIDER * 2

PWM 频率

1529

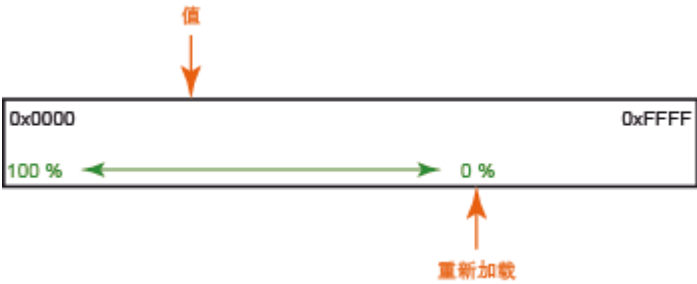
需相应的 PWM 频率, 具体视值类型而定。就 PWM 功能而言, PWM 频率通过重新加载值 (PWM) 传送, 或直接作为以 [Hz] 为单位的数值 (PWM1000) 传送。PWM 输出端的工作原理应控制器的不同而有所差异, 但效果是一样的。

PWM 频率通过内部运行的计数器 (衍生于 CPU 脉冲) 测量。该计数器通过 PWM 初始化启动。它从 0xFFFF 向后计数或从 0x0000 向前计数, 具体视 PWM 输出组 (0...3 和/或 4...7 或 4...11) 而定。如果达到传送的对比值 (VALUE), 则设定输出端。如果计数器过溢 (计数器读数从 0x0000 变为 0xFFFF 或从 0xFFFF 变为 0x0000), 则复位输出端并重启操作。

如果该内部计数器的读数不在 0x0000 和 0xFFFF 之间, 则可针对内部计数器传送其他预设值 (RELOAD)。此时, PWM 频率提高。对比值必须在当前指定的范围内。

计算 RELOAD 值

1531



图：PWM 通道 0...3 的 RELOAD 值

内部 PWM 计数器的 RELOAD 值根据参数 DIV64 和 CPU 频率计算，如下所示：

	• CabinetController: CR0303 • ClassicController: CR0020, CR0505 • ExtendedController: CR0200 • SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506	• CabinetController: CR0301, CR0302 • SmartController: CR250n • PCB 控制器：CS0015 • PDM360smart: CR1071
DIV64 = 0	RELOAD = 20 MHz / f _{PWM}	RELOAD = 10 MHz / f _{PWM}
DIV64 = 1	RELOAD = 312.5 kHz / f _{PWM}	RELOAD = 156.25 kHz / f _{PWM}

输入端 DIV64 必须设为 FALSE (0) 或 TRUE (1)，具体看需要高 PWM 频率还是低 PWM 频率。如果频率分别低于 305 Hz 和 152（根据控制器），则 DIV64 必须设为 "1" 以确保 RELOAD 值不会大于 0xFFFF。

RELOAD 值计算示例

1532

• CabinetController: CR0303 • ClassicController: CR0020, CR0505 • ExtendedController: CR0200 • SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506	• CabinetController: CR0301, CR0302 • SmartController: CR250n • PCB 控制器：CS0015 • PDM360smart: CR1071
PWM 频率应为 400 Hz。 20 MHz _____ = 50 000 = 0xC350 = RELOAD 400 Hz	PWM 频率应为 200 Hz。 10 MHz _____ = 50 000 = 0xC350 = RELOAD 200 Hz
因此 PWM 值的允许范围为 0x0000...0xC350。 这样，输出端切换的对比值必须在 0x0000 和 0xC350 之间。	

因此传号空号比如下：

传号空号比	开启时间	传号空号比的值
最小值	0 %	50 000 = 0xC350
最大值	100 %	0 = 0x0000

可在最小值和最大值之间触发 50 000 中间值（PWM 值）。

PWM 抖动频率

1534

就特定液压阀类型而言，必须在 PWM 频率上额外叠加所谓的抖动频率。如果在较长的周期内通过恒定的 PWM 值触发阀门，则可能会因为系统温度高而阻塞。

为避免此种情况，在抖动频率的基础上通过定义值 (DITHER_VALUE) 提高或降低 PWM 值。因此，抖动频率和振幅的振动 DITHER_VALUE 叠加在恒定的 PWM 值之上。抖动频率显示为 PWM 频率的比率（除法器， $DITHER_DIVIDER \cdot 2$ ）。

斜坡功能

1535

为防止突然从一个 PWM 值变为另一个 PWM 值，如从 15 % ON 变为 70 % ON，可延迟增加，具体为使用 **PT1** (→ 页 240)。PWM 斜坡功能基于 CODESYS 库 UTIL.LIB。这样有助于顺利启动液压系统等。

❗ 注意

安装 ecomatmobile DVD“软件、工具和文档”时，包含示例的项目已存储于您的 PC 的程序目录：

...\ifm electronic\CoDeSys V...\Projects\DEMO_PLC_DVD_V... (针对控制器) 或

...\ifm electronic\CoDeSys V...\Projects\DEMO_PDM_DVD_V... (针对 PDM)。

就该主题而言，您还可了解包含示例的项目。强烈建议遵循所示的步骤。

❗ 控制器 PWM 功能为处理器提供的硬件功能。PWM 功能保持设定状态，直至在控制器上进行硬件复位（断电通电）。

PWM100

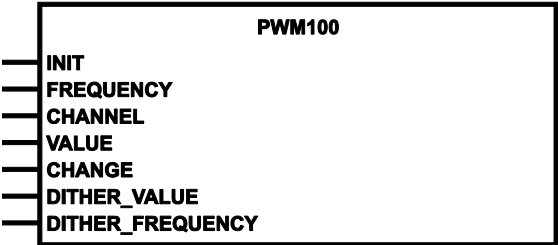
20461

单元类型 = 功能块 (FB)

 新的 **ecomatmobile** 控制器仅支持 **PWM1000** (→ 页 [210](#)).

元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

20462

PWM100 处理 PWM 输出端的初始化和参数设定。

FB 有助于简单应用 **ecomatmobile** 控制器中的 PWM FB。PWM 频率可直接以 [Hz] 为单位显示，传号空号比以 1 % 为单位增减。该 FB **不适于**控制器用途，因为分级较粗。

应用程序初始化期间，每个通道调用一次 FB。就此而言，输入端 INIT 必须设为 TRUE。初始化期间还分配参数 FREQUENCY。

 **注意**

通道 4...11 的值 FREQUENCY 必须相同。

就这些通道而言，**PWM** (→ 页 [204](#))和 PWM100 不得混淆。

PWM 频率内部限制为 5 kHz。

在程序循环处理期间，INIT 设为 FALSE。FB 调用并分配新的 PWM 值。如果输入端 CHANGE = TRUE，则采用值。

可实施已初始化 PWM 通道的电流测量：

- 通过 **OUTPUT_CURRENT** (→ 页 [201](#))
- 或例如使用 **IFM** 装置 EC2049 (电流测量系列元件)。

应用程序初始化期间，每个通道调用一次 DITHER。此时，输入端 INIT 必须设为 TRUE。初始化期间，传送确定抖动频率和抖动值 (VALUE) 的值 FREQUENCY。

 参数 DITHER_FREQUENCY 和 DITHER_VALUE 可针对每个通道单独设定。

输入端参数

20463

参数	数据类型	说明
INIT	BOOL	FALSE $\Rightarrow$ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
FREQUENCY	WORD	PWM 频率, 以 [Hz] 为单位 允许 = 20...250 = 0x0014...0x00FA
CHANNEL	BYTE	PWM 输出通道编号 输出端 Q10...Q13 对应 0...3 输出端 Q20...Q23 对应 4...7 输出端 Q40...Q47 对应 8...15
VALUE	BYTE	当前 PWM 值
CHANGE	BOOL	TRUE : 采用以下位置的新值 : • VALUE : 当前 PMW 周期结束后 • DITHER_VALUE : 当前抖动周期结束后 FALSE : 更改的 PWM 值对输出端无影响
DITHER_VALUE	BYTE	抖动峰间值, 以 [%] 为单位 允许值 = 0...100 = 0x00...0x64
		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。

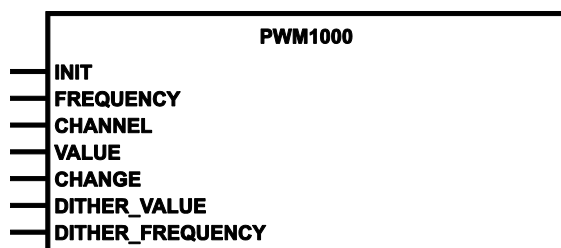
PWM1000

20465

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

20466

PWM1000 处理 PWM 输出端的初始化和参数设定。

FB 有助于简单使用 **ecomatmobile** 装置中的 PWM FB。PWM 频率可直接以 [Hz] 为单位显示，传号空号比以 1 % 为单位增减。

应用程序初始化期间，每个通道调用一次 FB。此时，输入端 INIT 必须设为 TRUE。初始化期间还分配参数 FREQUENCY。

ⓘ 注意

通道 4...11 的值 FREQUENCY 必须相同。

就这些通道而言，**PWM** (→ 页 [204](#))和 PWM1000 不得混淆。

PWM 频率内部限制为 5 kHz。

在程序循环处理期间，INIT 设为 FALSE。FB 调用并分配新的 PWM 值。如果输入端 CHANGE = TRUE，则采用值。

可实施已初始化 PWM 通道的电流测量：

- 通过 **OUTPUT_CURRENT** (→ 页 [201](#))
- 或例如使用 **IFM** 装置 EC2049 (电流测量系列元件)。

应用程序初始化期间，每个通道调用一次 DITHER。此时，输入端 INIT 必须设为 TRUE。初始化期间，传送确定抖动频率和抖动值 (VALUE) 的值 FREQUENCY。

ⓘ 参数 DITHER_FREQUENCY 和 DITHER_VALUE 可针对每个通道单独设定。

输入端参数

20471

参数	数据类型	说明
INIT	BOOL	FALSE $\Rightarrow$ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
FREQUENCY	WORD	PWM 频率, 以 [Hz] 为单位 允许 = 20...250 = 0x0014...0x00FA
CHANNEL	BYTE	PWM 输出通道编号 输出端 Q10...Q13 对应 0...3 输出端 Q20...Q23 对应 4...7 输出端 Q40...Q47 对应 8...15
VALUE	WORD	PWM 值 (传号空号比), 以 [%] 为单位 允许 = 0...1 000 = 0x0000...0x03E8 值 > 1 000 视作 = 1 000
CHANGE	BOOL	TRUE: 采用以下新值: • FREQUENCY: 当前 PWM 周期结束后 • VALUE: 当前 PWM 周期结束后 • DITHER_VALUE: 当前抖动周期结束后 • DITHER_FREQUENCY: 当前抖动周期结束后 FALSE: 更改的 PWM 值对输出端无影响
DITHER_VALUE	WORD	抖动峰间值, 以 [%] 为单位 允许值 = 0...1 000 = 0000...03E8
		抖动频率, 以 [Hz] 为单位 值范围 = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY 必须为偶数! FB 可将所有其他至增加到下一个匹配值。

5.2.12 功能元件：液压控制

内容	
CONTROL_OCC.....	213
JOYSTICK_0.....	216
JOYSTICK_1.....	220
JOYSTICK_2.....	224
NORM_HYDRAULIC.....	227

19540

库 ifm_hydraulic_16bitS05_Vxxyyzz.LIB 包含以下功能块：

CONTROL_OCC (→ 页 213)	OCC = Output Current Control 将输入值 [WORD] 换算至所示电流范围。
JOYSTICK_0 (→ 页 216)	将操纵杆的信号 [INT] 换算成明确定义的特性曲线，标准化为 0... 1000
JOYSTICK_1 (→ 页 220)	换算操纵杆 D 的信号 [INT]，标准化为 0... 1000
JOYSTICK_2 (→ 页 224)	将操纵杆的信号 [INT] 换算成可配置的特性曲线；自由选择标准化
NORM_HYDRAULIC (→ 页 227)	将定义的限值范围内的值 [DINT] 正常化为新的限值范围内的值

需要库 UTIL.Lib (CODESYS 程序包) 中的以下功能块：

- RAMP_INT
- CHARCURVE

这些功能块可自动调用并用过液压库的功能块配置。

The following function blocks are needed from the library:ifm_CR0020_Vxxyyzz.LIB

OUTPUT_CURRENT (→ 页 201)	测量输出通道上的电流 (平均，通过抖动周期)
OUTPUT_CURRENT_CONTROL (→ 页 202)	PWMi 输出通道的电流控制器

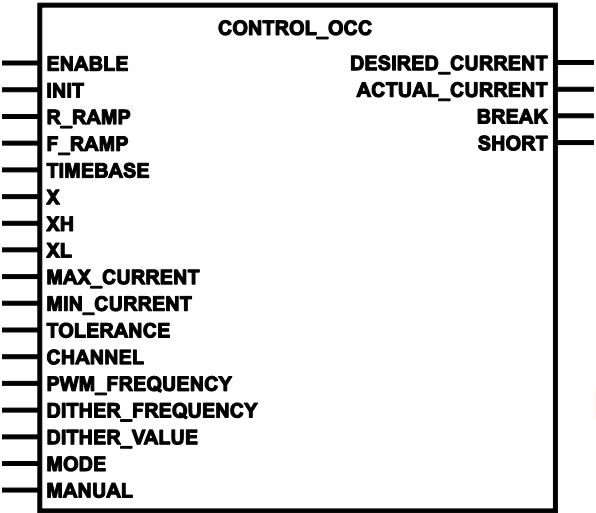
这些功能块可自动调用并用过液压库的功能块配置。

CONTROL_OCC

6245

单元类型 = 功能块 (FB)
元件类型 ifm_HYDRAULIC_16bit0505_Vxxyzz.Lib

CODESYS 中的符号：



说明

600

CONTROL_OCC 将输入值 X 换算至特定电流范围。

在每个 PLC 周期中，FB 的每个实例调用一次。FB 使用 **OUTPUT_CURRENT_CONTROL** (→ 页 202)和 **OUTPUT_CURRENT** (→ 页 201) (来自库 ifm_CR0020_Vxxyzz.LIB. 控制器设计为自适应控制器以便进行自我优化。

如果无需该自我优化性能，则可通过输入端 MANUAL 传输 > 0 的值：→ 自我优化性能禁用。

MANUAL 中的数值代表补偿值，将影响控制器的**积分**分量和**微分**分量。若要确定 MANUAL 模式下的最佳控制器设定，则可使用值 50。

增加值 MANUAL：→ 控制器变得更敏感/更快
减小值 MANUAL：→ 控制器变得不太敏感/更慢

如果输入端 MANUAL 设为 0，则控制器始终自我优化。控制系统的性能永久受监控，且在每个周期中，更新的补偿值自动永久存储。控制系统的更改会即刻加以识别和修正。

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

输入端参数

6247

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
INIT	BOOL	FALSE ⇌ TRUE (edge): 单元初始化 FALSE: 在程序进一步处理期间
R_RAMP	INT	斜坡上升沿 以 [increments/PLC cycle] 或 [increments/TIMEBASE] 为单位 0 = 无斜坡
F_RAMP	INT	斜坡下降沿 以 [increments/PLC cycle] 或 [increments/TIMEBASE] 为单位 0 = 无斜坡
TIMEBASE	TIME	斜坡上升沿和下降沿参考： t#0s = 上升沿/下降沿，以 [increments/PLC cycle] 为单位 ⚠ 快速控制器的周期时间非常短！ 否则 = 上升沿/下降沿，以 [increments/TIMEBASE] 为单位
X	WORD	输入值
XH	WORD	输入值范围的上限值 [increments]
XL	WORD	输入值范围的下限值 [increments]
MAX_CURRENT	WORD	Max. valve current in [mA]
MIN_CURRENT	WORD	最低阀门电流，以 [mA] 为单位
TOLERANCE	BYTE	最低阀门电流公差，以 [increments] 为单位 超过公差时会跳至 MIN_CURRENT
CHANNEL	BYTE	电流控制输出通道编号 输出端 Q10...Q13 对应 0...3 输出端 Q20...Q23 对应 4...7
PWM_FREQUENCY	WORD	输入端负载 PWM 频率 [Hz]
		抖动频率，以 [Hz] 为单位 值范围 = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY 必须为偶数！ FB 可将所有其他至增加到下一个匹配值。
DITHER_VALUE	BYTE	抖动峰间值，以 [%] 为单位 允许值 = 0...100 = 0x00...0x64

IFM 功能元件
针对装置 CR0020 的 IFM 功能元件

参数	数据类型	说明
MODE	BYTE	控制器特性： 0 = 增加非常缓慢，无过冲 1 = 增加缓慢，无过冲 2 = 最低过冲 3 = 允许中等过冲
MANUAL	BYTE	值 = 0：控制器按照自我优化的方式运行 值 > 0：闭合回路控制器的自我优化性能将被覆盖（一般：50）

输出端参数

602

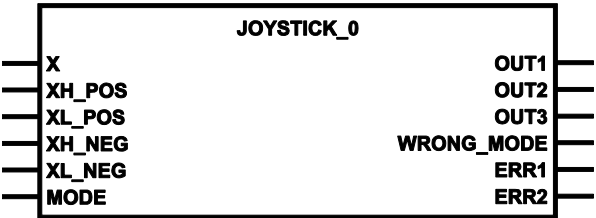
参数	数据类型	说明
DESIRED_CURRENT	WORD	OCC 所需电流值，以 [mA] 为单位 (针对监控目的)
ACTUAL_CURRENT	WORD	输出电流，以 [mA] 单位
BREAK	BOOL	错误：输出端电缆中断
SHORT	BOOL	错误：输出端电缆短路

JOYSTICK_0

13224

单元类型 = 功能块 (FB)
元件类型 ifm_hydraulic_16bit0S05_Vxxyyzz.Lib

CODESYS 中的符号：



说明

432

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

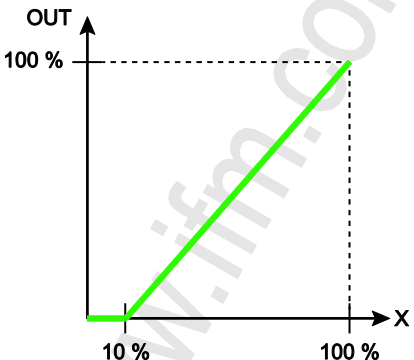
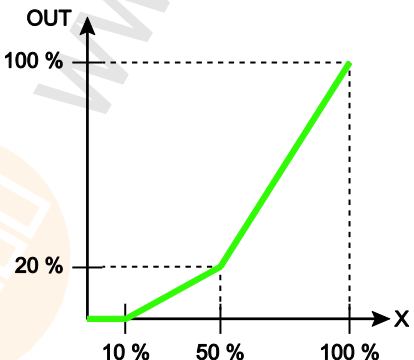
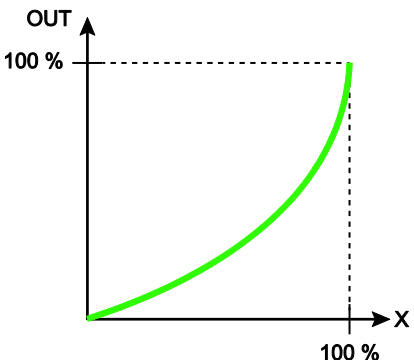
就此 FB 而言，特性曲线值是指定的（→ 图）：

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

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

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

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

输入端参数

3

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

输出端参数

6252

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

ERR1 和 ERR2 的可能结果：

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

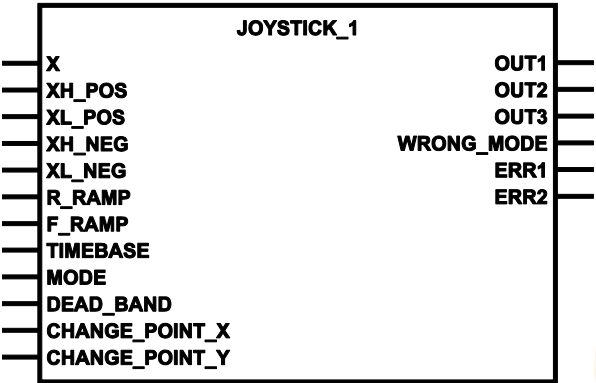
JOYSTICK_1

13227

单元类型 = 功能块 (FB)

元件类型 ifm_hydraulic_16bit0S05_Vxxyyzz.Lib

CODESYS 中的符号：

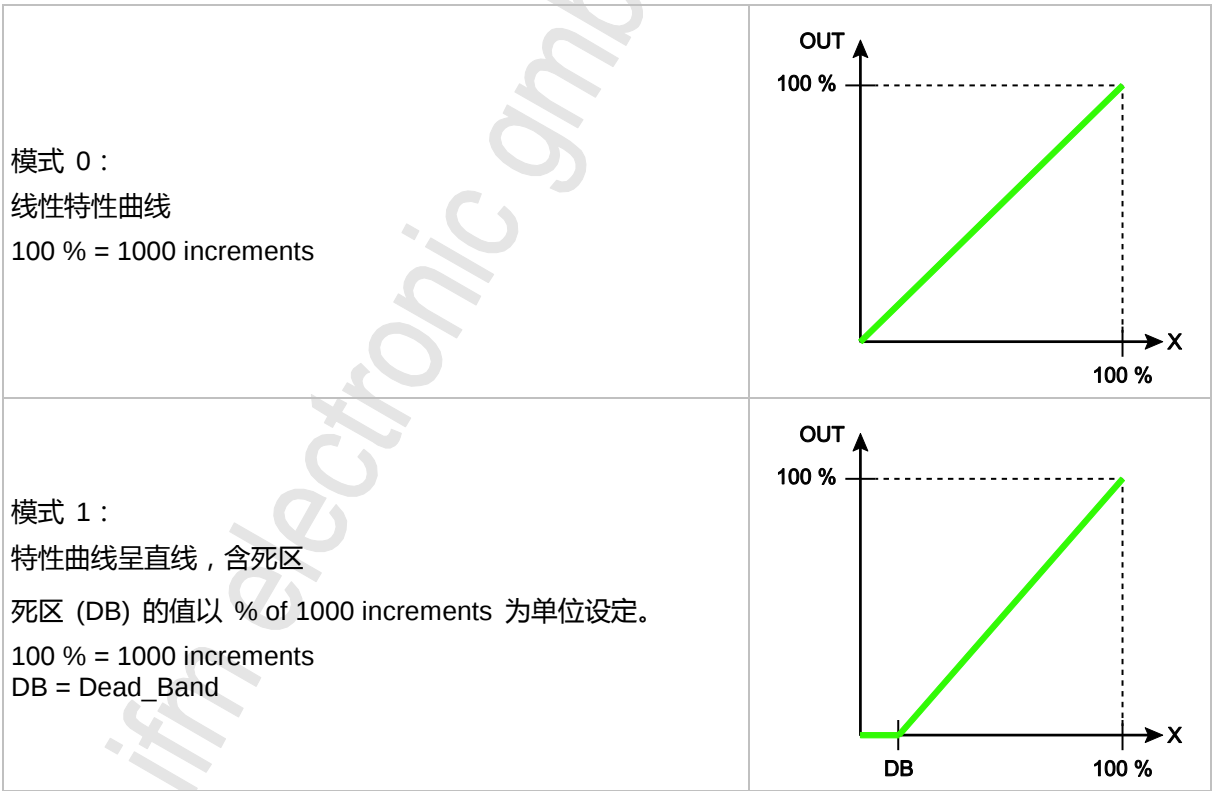


说明

425

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

就此 FB 而言，特性曲线值可配置（→ 图）：



模式 2 : 2 步式线性特性曲线, 含死区 值可配置为 : 死区 : 0...DB, 以 % of 1000 increments 为单位 步骤 : X = CPX, 以 % of 1000 increments 为单位 Y= CPY, 以 % of 1000 increments 为单位 100 % = 1000 increments DB = Dead_Band CPX = Change_Point_X CPY = Change_Point_Y	
特性曲线模式 3 : 曲线上升 (直线固定)	

输入端参数

6256

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

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

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

输出端参数

6252

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

ERR1 和 ERR2 的可能结果：

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

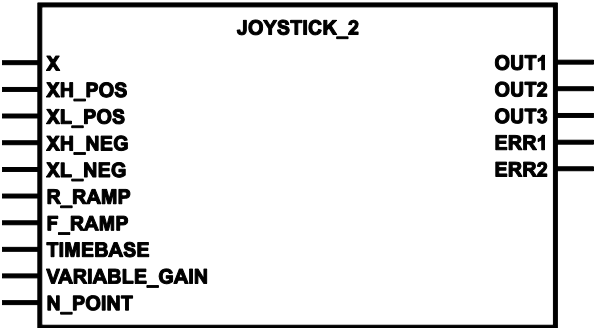


JOYSTICK_2

13228

单元类型 = 功能块 (FB)
元件类型 ifm_hydraulic_16bit0S05_Vxxyyzz.Lib

CODESYS 中的符号：

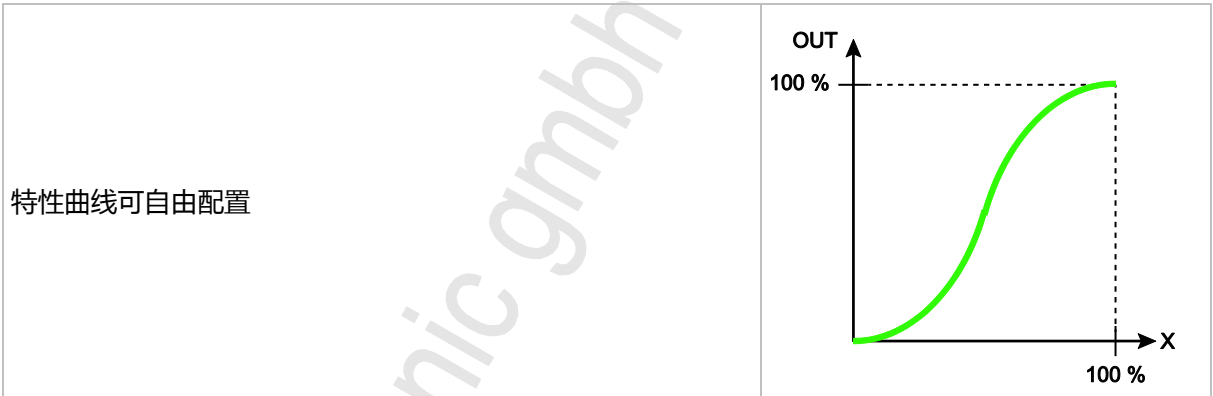


说明

418

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

就此 FB 而言，特性曲线值可自由配置（→ 图）：



输入端参数

261

参数	数据类型	说明
X	INT	输入值 [increments]
XH_POS	INT	正向最大预设值 [increments] (亦允许负值)
XL_POS	INT	正向最小预设值 [increments] (亦允许负值)
XH_NEG	INT	负向最大预设值 [increments] (亦允许负值)

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

参数	数据类型	说明
XL_NEG	INT	负向最小预设值 [increments] (亦允许负值)
R_RAMP	INT	斜坡上升沿, 以 [increments/PLC cycle] 为单位 0 = 无斜坡
F_RAMP	INT	斜坡下降沿, 以 [increments/PLC cycle] 为单位 0 = 无斜坡
TIMEBASE	TIME	斜坡上升沿和下降沿参考: t#0s = 上升沿/下降沿, 以 [increments/PLC cycle] 为单位 ! 快速控制器的周期时间非常短! 否则 = 上升沿/下降沿, 以 [increments/TIMEBASE] 为单位
VARIABLE_GAIN	ARRAY [0..10] OF POINT	描述曲线的数值对 使用 N_POINT 所示的第一个数值对。n = 2...11 例如: 9 个数值对声明为变量 VALUES: VALUES: ARRAY [0..10] OF POINT := (X:=0,Y:=0), (X:=200,Y:=0), (X:=300,Y:=50), (X:=400,Y:=100), (X:=700,Y:=500), (X:=1000,Y:=900), (X:=1100,Y:=950), (X:=1200,Y:=1000), (X:=1400,Y:=1050); 值之间可能有空白。
N_POINT	BYTE	定义曲线特性的点号 (VARIABLE_GAIN 数值对) n = 2...11

输出端参数

420

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

ERR1 和 ERR2 的可能结果：

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

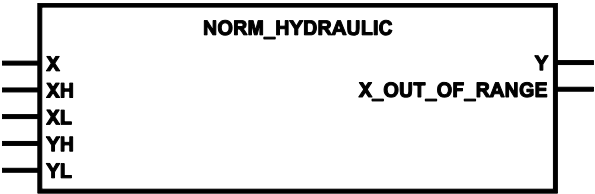


NORM_HYDRAULIC

13232

单元类型 = 功能块 (FB)
元件类型 ifm_hydraulic_16bit0S05_Vxxyyzz.Lib

CODESYS 中的符号：



说明

397

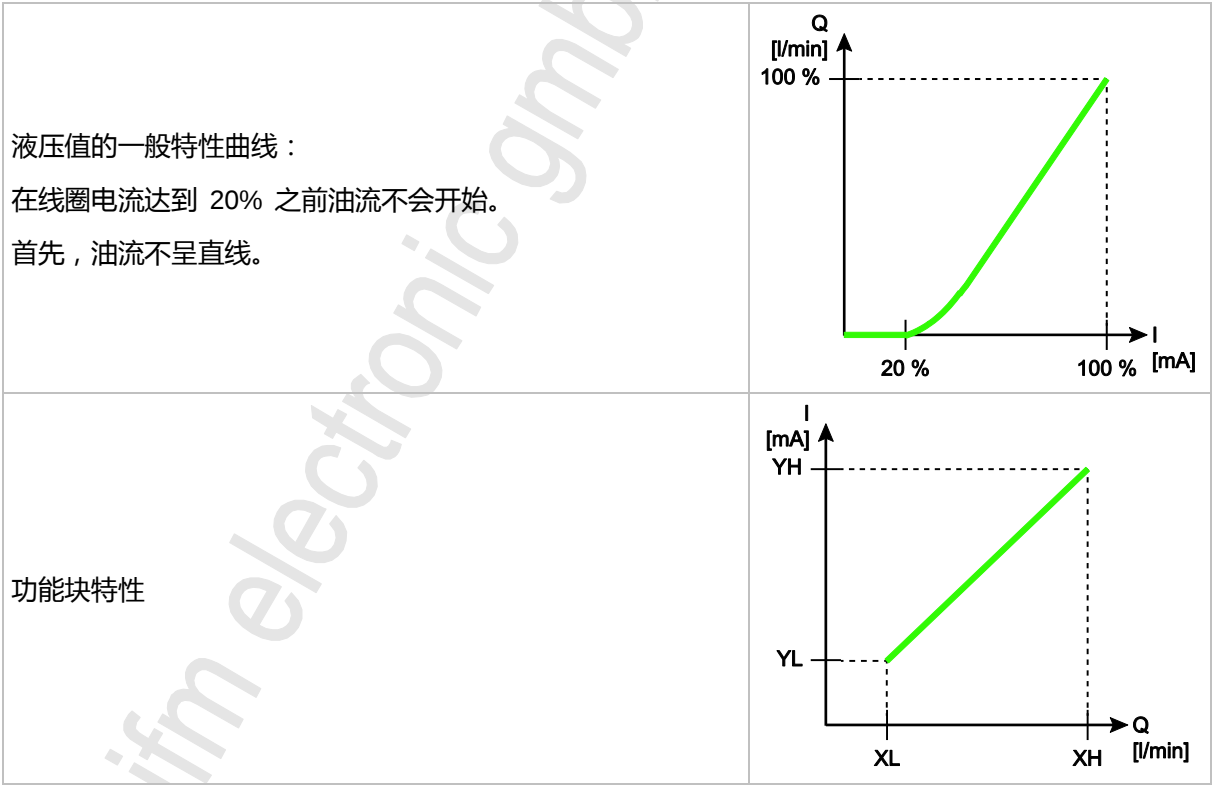
NORM_HYDRAULIC 将包含固定限值的输入值标准化为包含新限值的值。

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

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

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

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



输入端参数

398

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

输出端参数

399

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

例如： NORM_HYDRAULIC

400

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

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

5.2.13 功能元件：控制器

内容

控制器设定规则	230
DELAY	232
GLR	233
PID1	235
PID2	237
PT1	240

1634

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

控制器设定规则

1627

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

设定控制

1628

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

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

而后需确定平稳振荡的时间周期 $T_{critical}$ 。

如有必要，添加微分分量。

T_N 应比 T_v 大 2...10 倍左右。

K_P 应等于 K_D 。

控制系统理想化设定：

控制器	$K_P = K_D$	T_N	T_v
P	$2.0 \cdot K_{P_{critical}}$	—	—
PI	$2.2 \cdot K_{P_{critical}}$	$0.83 \cdot T_{critical}$	—
PID	$1.7 \cdot K_{P_{critical}}$	$0.50 \cdot T_{critical}$	$0.125 \cdot T_{critical}$

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

过冲阻尼

1629

若要抑制过冲，则可使用 **PT1** (→ 页 [240](#)) (低通)。就此而言，PT1 连接指令抑制预设值 XS，而后再提供给控制器功能。

设定变量 T1 应比控制器 TN 大 4...5 倍左右。

DELAY

585

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

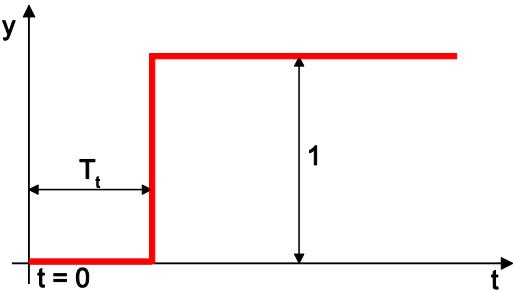
CODESYS 中的符号：



说明

588

DELAY 延迟输入值的输出，延迟时间 T（空载时间元件）。



图：DELAY 的时间特性

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

❗ 为确保 FB 正常运行：必须在每个周期中调用 FB。

输入端参数

589

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

输出端参数

590

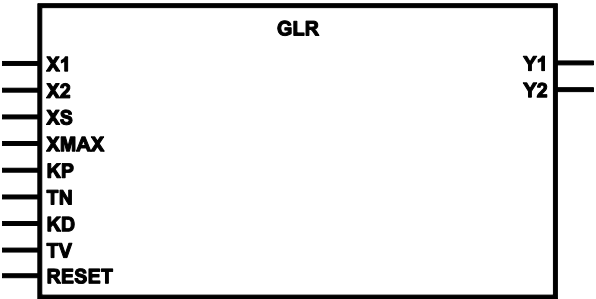
参数	数据类型	说明
Y	WORD	输入值，延迟时间 T

GLR

531

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

534

GLR 处理同步控制器。
同步控制器是包含 PID 特性的控制器。
输入端 KP 和 KD 输入的值在内部除以 10。因此分级更精细（如：KP = 17，对应 1.7）。
被称为较大实际值的被控变量将相应增加。
被称为较小实际值的被控变量对应参考变量。
参考变量 = 65 536 – (XS / XMAX * 65 536)。

⚠注意

被控变量 Y1 和 Y2 已被标准化为 PWM FB
(RELOAD 值 = 65 535)。 注意逆序逻辑：
65 535 = 最小值
0 = 最大值。
注意，输入值 KD 取决于周期时间。 为获得稳定、可重复的控制特性，应按照时间控制的方式调用 FB。

输入端参数

535

参数	数据类型	说明
X1	WORD	通道 1 实际值
X2	WORD	通道 2 实际值
XS	WORD	预设值
XMAX	WORD	最大预设值
KP	Byte	比例分量常数 (/10) (仅限正值!)
TN	TIME	积分作用时间 (积分分量)
KD	BYTE	微分分量 (/10) (仅限正值!)
TV	TIME	微分作用时间 (微分分量)
RESET	BOOL	TRUE: 复位功能元件 FALSE: function element is not executed

输出端参数

536

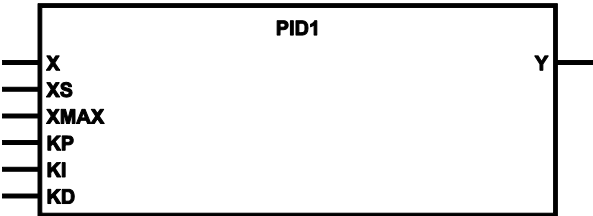
参数	数据类型	说明
Y1	WORD	通道 1 被控变量
Y2	WORD	通道 2 被控变量

PID1

351

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

354

PID1 处理 PID 控制器。

PID 控制器被控变量的改变包含比例 (P)、积分 (I) 和微分 (D) 分量。被控变量先改变一个量，具体取决于输入值的变化速度 (D 分量)。在微分的时间结束后，被控变量返回至与比例范围一致的值，然后根据复位时间改变。

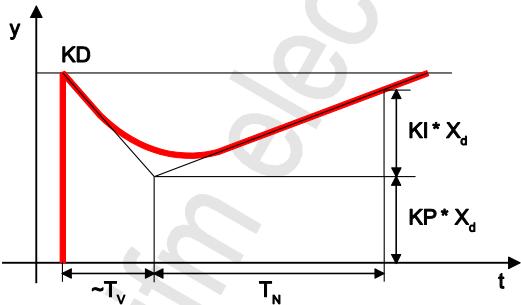
❗ 注意

被控变量 Y 已标准化为 PWM FB (RELOAD 值 = 65,535)。注意逆序逻辑：
65,535 = 最小值
0 = 最大值。

注意，输入值 KI 和 KD 取决于周期时间。为获得稳定、可重复的控制特性，应按照时间控制的方式调用 FB。

如果 $X > X_S$ ，则提高被控变量。
如果 $X < X_S$ ，则降低被控变量。

被控变量 Y 有以下时间特性：



图：PID 控制器的一般阶跃响应

输入端参数

355

参数	数据类型	说明
X	WORD	输入值
XS	WORD	预设值
XMAX	WORD	最大预设值
KP	BYTE	输出信号的比例分量
KI	BYTE	输出信号的积分分量
KD	BYTE	输出信号的微分分量

输出端参数

356

参数	数据类型	说明
Y	WORD	被控变量 (0...1000 ‰)

建议的设定

357

KP = 50

KI = 30

KD = 5

控制器根据上述值快速稳定运行。 在该设定下，控制器不会波动。

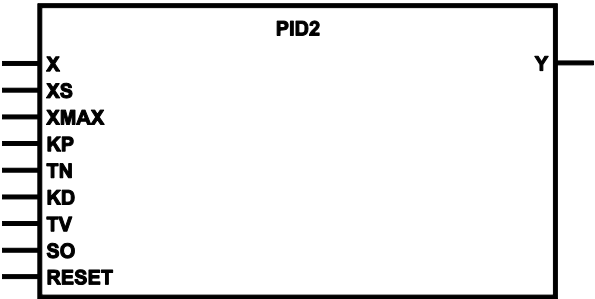
► 若要优化控制器，则后续可逐渐更改值。

PID2

9167

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

347

PID2 处理包含自我优化功能的 PID 控制器。

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

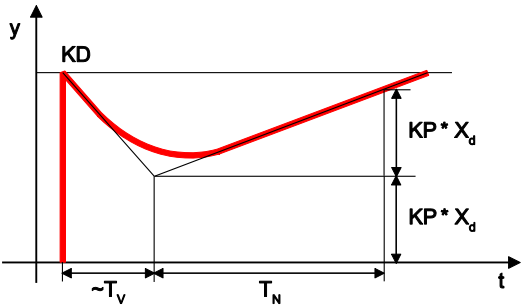
输入端 KP 和 KD 输入的值在内部除以 10。因此分级更精细 (如：KP = 17，对应 1.7)。

⚠注意

被控变量 Y 已标准化为 PWM FB (RELOAD 值 = 65,535)。注意逆序逻辑：
65,535 = 最小值
0 = 最大值。

注意，输入值 KD 取决于周期时间。为获得稳定、可重复的控制特性，应按照时间控制的方式调用 FB。

如果 $X > X_S$, 则提高被控变量。
如果 $X < X_S$, 则降低被控变量。
参考变量内部添加至被控变量。
 $Y = Y + 65,536 - (X_S / X_{MAX} * 65,536)$ 。
被控变量 Y 有以下时间特性。



图：PID 控制器的一般阶跃响应

输入端参数

348

参数	数据类型	说明
X	WORD	输入值
X _S	WORD	预设值
X _{MAX}	WORD	最大预设值
K _P	Byte	比例分量常数 (/10) (仅限正值！)
T _N	TIME	积分作用时间 (积分分量)
K _D	BYTE	微分分量 (/10) (仅限正值！)
T _V	TIME	微分作用时间 (微分量)
S _O	BOOL	TRUE： 自我优化已启用 FALSE： 自我优化未启用
RESET	BOOL	TRUE： 复位功能元件 FALSE: function element is not executed

输出端参数

349

参数	数据类型	说明
Y	WORD	被控变量 (0...1000 ‰)

建议的设置

9127
350

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

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

功能输入端 SO（自我优化）显著改善控制性能。实现理想特性的前提：

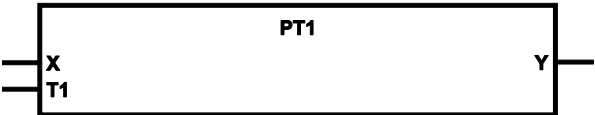
- 控制器采用 I 分量 ($TN \geq 50 \text{ ms}$)
- 参数 KP 以及 TN 已根据实际控制系统进行充分调整，尤其是 TN。
- 采用 ± 127 的控制范围 (X - XS) (如有必要，通过乘以 X、XS 和 XMAX 来扩大控制范围)。
- ▶ 完成参数设定后，可设定 SO = TRUE。
- > 这将显著改善控制性能，尤其是减少过冲。

PT1

338

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



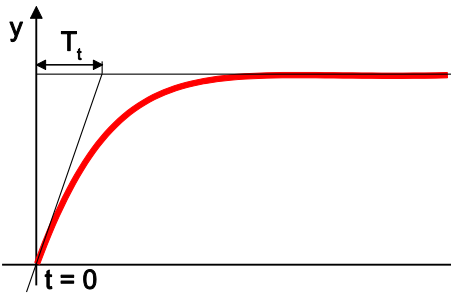
说明

341

PT1 处理包含一级时间延迟的控制系统。
该 FB 为包含时间延迟的比例控制系统。 例如，可用于在使用 PWM FB 时生成斜坡。

❗如果 T1 比 SPS 周期时间短，FB 输出端则会变得不稳定。

低通过滤器输出变量 Y 具备以下时间特性（单位阶跃）：



图：PT1 的时间特性

输入端参数

342

参数	数据类型	说明
X	INT	输入值 [increments]
T1	TIME	延迟时间（时间常数）

输出端参数

343

参数	数据类型	说明
Y	INT	输出值

5.2.14 功能元件：软件复位

内容

SOFTRESET	242
-----------------	-----

1594

可使用该 FB，利用应用程序中的命令重启控制器。

SOFTRESET

260

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

263

SOFTRESET 使得装置完全重启。

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

在重启之前，存储保持变量。

错误内存记录重启。

ⓘ 进行有效通信时：必须考虑长时间的复位周期，否则会发送保护信号。

输入端参数

264

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

5.2.15 功能元件：时间测量/设定

内容	
TIMER_READ	244
TIMER_READ_US	245

1601

通过以下 IFM 功能块，您可...

- 在应用程序中测量时间并加以评估，
- 必要时更改时间值。

TIMER_READ

236

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

239

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

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

输出端参数

241

参数	数据类型	说明
T	TIME	当前系统时间 [ms]

TIMER_READ_US

657

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

660

TIMER_READ_US 读取当前系统时间，以 [μs] 为单位
施加电源电压时，装置生成时钟脉冲，并在寄存器中递增计数。该寄存器可通过 FB 调用读取，且还可用于时间测量等。

 信息

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

输出端参数

662

参数	数据类型	说明
TIME_US	DWORD	当前系统时间 [μs]

5.2.16 功能元件：保存、读取和转换内存中的数据

内容	
数据备份存储类型.....	246
数据自动保存.....	247
手动数据存储.....	250

13795

数据备份存储类型

13805

装置提供以下内存类型：

闪存

13803

属性：

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

FRAM 内存

13802

FRAM 在此表示所有类型的非易失快速内存。

属性：

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

数据自动保存

内容	
MEMORY_RETAIN_PARAM	248

2347

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

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

MEMORY_RETAIN_PARAM (→ 页 [248](#)).

详细信息 → 章节 **变量** (→ 页 [85](#))

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

MEMORY_RETAIN_PARAM

2372

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

2374

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

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

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

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

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

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

配置后仍保留在装置中，即便重新加载应用程序或运行时系统。

输入端参数

2375

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

手动数据存储

内容	
FLASHREAD	251
FLASHWRITE	252
FRAMREAD	254
FRAMWRITE.....	255
MEMCPY	256

13801

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

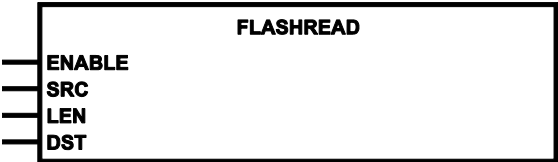
 程序员可通过存储分区 (→ 章节 **可用内存** (→ 页 [16](#))) 了解哪些内存区域可用。

FLASHREAD

561

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

564

- FLASHREAD 有助于直接从闪存读取不同类型的数据。
- > FB 从闪存读取从 SRC 地址开始的内容。且此时，传送的字节数量跟 LEN 所示一样。
 - > 调用 FB 的周期内可充分读取内容。
 - ▶ 请确保 RAM 中的目标内存区域充足。
 - ▶ 就目标地址而言，DST 适用：
 - ❗ 通过运算符 ADR 确定地址并将其分配至 FB！

输入端参数

565

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
SRC	WORD	内存中的相对源起始地址 允许 = 0...65 535 = 0x0000...0xFFFF
LEN	WORD	数据字节数量 允许 = 0...65 535 = 0x0000...0xFFFF
DST	DWORD	目标变量的起始地址 ❗ 通过运算符 ADR 确定地址并将其分配至 FB！

FLASHWRITE

555

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

558

 警告

无法控制流程操作可能造成危险！
执行 FLASHWRITE 时输入端/输出端状态“冻结”。
▶ 机器正在运行时切勿执行该 FB！

- FLASHWRITE 有助于将不同的数据类型直接写入闪存。
- 设定期间可利用该 FB 存储大数据量，且在进程中仅可进行读取访问。
- ▶ 如果已写入页面（即便仅部分），需先删除整个闪存区域，而后再对该页面进行写入访问。对地址 0 进行写入访问即可完成该操作。
 - ▶ 切入多次写入至页面！始终先删除所有内容！
否则会出现陷阱或电子狗错误。
 - ▶  切勿删除闪存区域超过 100 次。否则将无法保证其他闪存区域的数据一致性。
 - ▶ 在每个 SPS 周期内，FLASHWRITE 即可启动一次！
 - ▶ 就源开始地址而言，SRC 适用：
 通过运算符 ADR 确定地址并将其分配至 FB！
- > FB 将 SRC 地址内容写入闪存。且此时，传送的字节数量跟 LEN 所示一样。
-  若目标开始地址 DST 超出允许范围：无数据传输！

输入端参数

559

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	WORD	内存中的相对目标起始地址 允许 = 0...65 535 = 0x0000...0xFFFF
LEN	WORD	数据字节数量 允许 = 0...65 535 = 0x0000...0xFFFF
SRC	DWORD	源变量的起始地址  通过运算符 ADR 确定地址并将其分配至 FB !

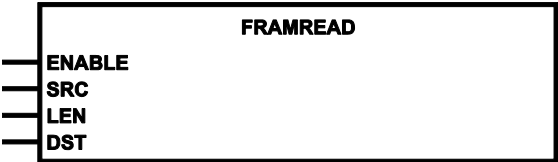
FRAMREAD

549

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

552

FRAMREAD 有助于从 FRAM 中直接快速读取各种数据类型 ¹⁾。

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

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

¹⁾ FRAM 在此表示所有类型的非易失快速内存。

输入端参数

553

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
SRC	WORD	内存中的相对源起始地址 允许 = 0...1 023 = 0x0000...0x03FF
LEN	WORD	数据字节数量
DST	DWORD	目标变量的起始地址 ❗ 通过运算符 ADR 确定地址并将其分配至 FB！

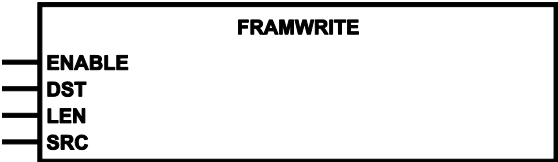
FRAMWRITE

543

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyzz.LIB

CODESYS 中的符号：



说明

546

FRAMWRITE 有助于将各种数据类型直接快速写入 FRAM 内存¹⁾。
FB 将 SRC 地址内容写入非易失 FRAM 内存。且此时，传送的字节数量跟 LEN 所示一样。
如果超过 FRAM 内存区域所示的字节数量，则最多可写入 FRAM 末端的数据。

- ▶ 就源地址而言，SRC 适用：
 - ❗ 通过运算符 ADR 确定地址并将其分配至 FB！
- ❗ 若目标地址 DST 超出允许范围：无数据传输！

¹⁾ FRAM 在此表示所有类型的非易失快速内存。

输入端参数

547

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	WORD	内存中的相对目标起始地址 允许 = 0...1 023 = 0x0000...0x03FF
LEN	WORD	数据字节数量
SRC	DWORD	源变量的起始地址 ❗ 通过运算符 ADR 确定地址并将其分配至 FB！

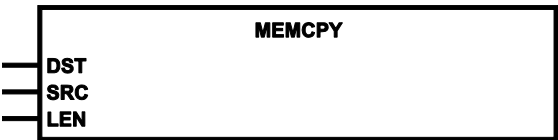
MEMCPY

409

= 内存复制

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyyzz.LIB



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 FB！

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

输入端参数

413

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

5.2.17 功能元件：数据访问和数据检查

内容	
CHECK_DATA	258
GET_IDENTITY	260
SET_DEBUG	262
SET_IDENTITY	263
SET_PASSWORD	264

1598

本章所述 FB 控制数据访问且可启用数据检查。

CHECK_DATA

603

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

606

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

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

不需要的更改： 错误！

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

需要的更改：

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

输入端参数

607

参数	数据类型	说明
STARTADR	DINT	受监控数据内存的起始地址 (WORD 地址，从 %MW0 起)  通过运算符 ADR 确定地址并将其分配至 FB！
LENGTH	WORD	受监控数据内存长度，以 [byte] 为单位
UPDATE	BOOL	TRUE： 可更改数据 FALSE： 不允许更改数据

输出端参数

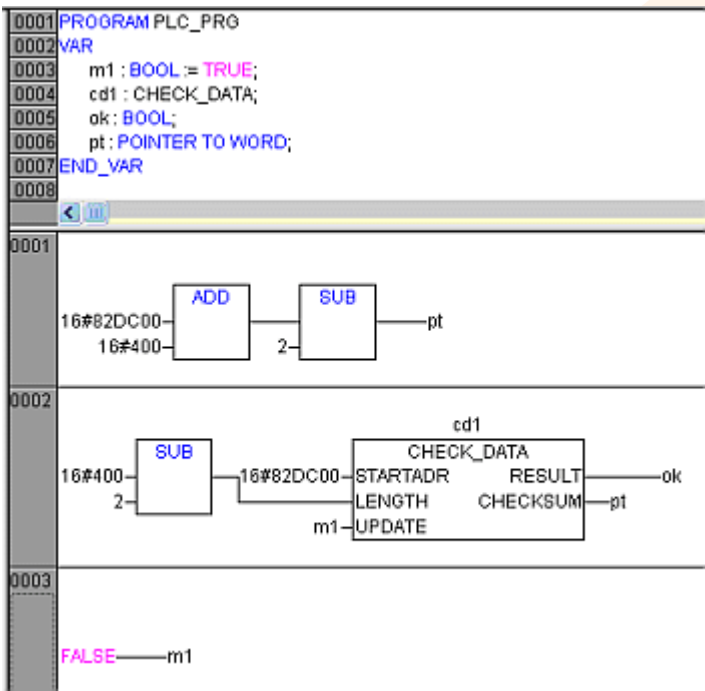
608

参数	数据类型	说明
RESULT	BOOL	TRUE : CRC 校验和正常 FALSE : CRC 校验和错误 (数据已修改)
CHECKSUM	DWORD	当前 CRC 校验和

例如：CHECK_DATA

4168

在以下示例中，程序决定校验和并通过指针 pt 存储在 RAM 中：



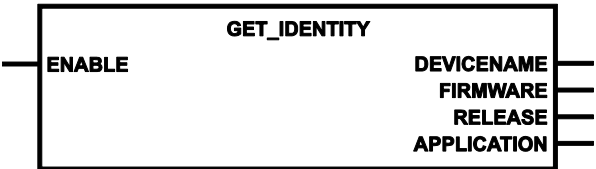
GET_IDENTITY

2212

单元类型 = 功能块 (FB)

元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

2344

GET_IDENTITY 读取存储在装置中的特定标识：

- 装置的硬件名称和硬件版本
- 装置运行时系统的名称
- 装置运行时系统的版本和版本号
- 应用程序的名称（之前已通过以下方式保存：**SET_IDENTITY** (→ 页 263))

输入端参数

2609

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

输出端参数

2610

参数	数据类型	说明
DEVICENAME	STRING(31)	硬件名称 作为最多 31 个字符的字符串，例如："CR0403"
FIRMWARE	STRING(31)	装置运行时系统的名称 作为最多 31 个字符的字符串 示例："CR0403"
RELEASE	STRING(31)	软件版本 作为最多 31 个字符的字符串

IFM 功能元件

针对装置 CR0020 的 IFM 功能元件

参数	数据类型	说明
APPLICATION	STRING(79)	应用程序名称 作为最多 79 个字符的字符串 示例: "Crane1704"



SET_DEBUG

290

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

293

SET_DEBUG 处理 DEBUG 模式而无需启用测试输入端 (→ 章节 **TEST 模式** (→ 页 [62](#))).
如果 FB 的输入端 DEBUG 设为 TRUE，编程系统或下载程序等则可与装置通信并执行某些特别的系统命令 (如针对服务功能，通过 GSM 调制解调器 CANremote)。

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

输入端参数

294

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

SET_IDENTITY

284

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

287

SET_IDENTITY 设定应用程序特定程序 ID。

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

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



*) 'Nozzle in front' 在此代替定制文本。

输入端参数

288

参数	数据类型	说明
ID	STRING(80)	最大长度为 80 个字符的任何文本

SET_PASSWORD

266

单元类型 = 功能块 (FB)
元件类型 ifm_CR0020_Vxxyyzz.LIB

CODESYS 中的符号：



说明

269

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

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

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

仅可在复位之前密码后设定新密码。

⚠ 加载新的应用程序以作为启动项目时，密码复位。

输入端参数

270

参数	数据类型	说明
ENABLE	BOOL	TRUE（仅针对 1 个周期）： 使用新参数 FALSE: 单元未执行
PASSWORD	STRING(16)	密码 如果 PASSWORD = ""，则无需输入密码即可访问

6 诊断和错误处理

内容	
诊断	265
故障	265
发生错误时的反应	266
继电器：重要说明！	266
响应系统错误	267
CAN / CANopen: 错误和错误处理	267

19598

运行时系统 (RTS) 通过内部错误检查功能检查装置：

- 启动阶段（复位阶段）
 - 应用程序执行期间
- 章节 **工作状态** (→ 页 [58](#))
- 此时尽可能地实现高工作可靠性。

6.1 诊断

19601

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

还可监控输入端和输入端是否正常运行，具体视装置而定。

- 断线，
- 短路，
- 值在范围之外。

就诊断而言，可使用装置“正常”运行期间创建的配置和日志数据。

初始化和启动期间可监控系统组件是否正确启动。 日志文件记录错误。

就进一步诊断而言，还可进行自检。

6.2 故障

19602

故障即无法执行所需功能的项目状态，不包括预防性维护或其他计划措施期间或因为缺乏外部资源而导致的失效状态。

故障通常是项目本身失灵的结果，也可能在之前没有失灵的情况下存在。

在 →ISO 13849-1 中，“故障”指“随机故障”。

6.3 发生错误时的反应

19603
12217

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

- > 工作 LED 呈红色亮起，
- > 输出端继电器关闭，
- > 继电器保护的输出端断开电源，
- > 输出端逻辑信号状态保持不变。

❗注意

如果继电器关闭输出端，则逻辑信号状态保持不变。

- ▶ 程序员必须评估 ERROR 位，因此还需在出现故障时复位输出逻辑。

① 装置特定错误代码和诊断消息的完整列表

→ 章节 **系统标志** (→ 页 [268](#)).

6.4 继电器：重要说明！

1446

通过系统标志 RELAIS 或 RELAY_CLAMP_15 使用逻辑功能 (→ 章节 **锁存** (→ 页 [18](#))) 亦可关闭所有其他输出端。

视应用程序而定，现在需决定是否通过复位系统标志位 ERROR 再次开启继电器 - 以及输出端。

此外，还可通过应用程序将系统标志位 ERROR 设为“已定义错误”。

通知

继电器触点可能过早磨损。

- ▶ 仅在紧急情况下将该功能用于一般的输出端关闭。
- ▶ 在正常操作中，仅关闭无负载的继电器！
为此，先通过应用程序关闭输出端！

6.5 响应系统错误

2258

❗ 程序员单独负责应用程序数据处理的安全性。

- ▶ 在应用程序中处理特定错误标志！
错误说明通过错误标志提供。
如有必要，该等错误标志可进一步加以处理。

如果出现严重错误，则可额外设定系统标志位 ERROR。

同时 ERROR = TRUE 会导致以下情况：

- 通过应用程序将所有相关输出端设为 FALSE，
- 工作 LED 呈红色亮起，
- ERROR 输出端设为 FALSE，且
- 输出端继电器 关闭。
- 因此通过这些继电器保护的输出端关闭。

分析和排除错误原因后：

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

6.6 CAN / CANopen: 错误和错误处理

19604

→ 系统手册“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	274
错误表	285

1664

除技术资料指示外，您还可了解附录中的摘要表。

7.1 系统标志

内容	
系统标志： CAN	269
系统标志： SAE J1939	269
系统标志： 错误标志（标准侧）	270
系统标志： 状态 LED（标准侧）	271
系统标志： 电压（标准侧）	272
系统标志： 输入端 16...40 和输出端 24...0（标准侧）	273

12167

 如果扩展 PLC 配置，则系统标志的地址可更改。

▶ 编程时仅可使用系统标志的符号名称！

→ 系统手册“ecomatmobile 技巧”

→ 章节 **错误代码和诊断信息**

7.1.1 系统标志：CAN

19555

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

x = 1...2 = CAN 接口编号

7.1.2 系统标志：SAE J1939

19556

系统标志 (符号名称)	类型	说明
J1939_TASK	BOOL	使用 J1939_TASK 即可满足发送 J1939 消息的时间要求。 如果 J1939 消息要在重复时间 ≤ 50 ms 的情况下发送, 则运行时系统自动设定 J1939_TASK=TRUE。 就时间要求为 ≥ PLC 周期时间的应用而言: ► 通过 J1939_TASK=FALSE 减少系统负载! TRUE: J1939 任务启用 (= 初始值) 每 2 ms 调用任务。 J1939 堆栈在所需时间帧内 发送其消息 FALSE: J1939 任务未启用

7.1.3 系统标志：错误标志（标准侧）

19557

系统标志（符号名称）	类型	说明
ERROR	BOOL	TRUE = set group error message, switch off relay
ERROR_BREAK_Qx (0...x, 值视装置而定→技术资料)	WORD	输出字 x：断线错误 [Bit 0 for output 0] ... [bit z for output z] of this group 位 = TRUE：错误 位 = FALSE：无错误
ERROR_Ix (0...x, 值视装置而定→技术资料)	BYTE	输入字节 x：外围故障 本组 [Bit 0 for input 0] ... [bit z for input z] 位 = TRUE：错误 位 = FALSE：无错误
ERROR_MEMORY	BOOL	内存错误
ERROR_POWER	BOOL	VBBS / 夹具 15 的电压错误： TRUE：值超出范围 或：差值 (VBB15 - VBBS) 太大 > 一般错误 FALSE：值正常
ERROR_SHORT_Qx (0...x, 值视装置而定→技术资料)	WORD	输出字 x：短路错误 [Bit 0 for output 0] ... [bit z for output z] of this group 位 = TRUE：错误 位 = FALSE：无错误
ERROR_TEMPERATURE	BOOL	温度错误 TRUE：值超出范围 > 一般错误 FALSE：值正常
ERROR_VBBx	BOOL	VBBx (x = o r) 电源电压错误： TRUE：值超出范围 > 一般错误 FALSE：值正常

7.1.4 系统标志：状态 LED (标准侧)

12817

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

7.1.5 系统标志：电压（标准侧）

19558

系统标志（符号名称）	类型	说明
CLAMP_15	BOOL	夹具 15 上的电压监控
RELAIS	BOOL	TRUE：继电器通电 输出端供应电压 FALSE：继电器断电 输出端无电压
RELAIS_CLAMP_15	BOOL	继电器夹具 15（插脚 5）
SERIAL_BAUDRATE	WORD	RS232 接口波特率
SERIAL_MODE	BOOL	启用串行接口 (RS232) 以用于应用程序 TRUE： RS232 接口可用于应用程序，但不再可用于装置的编程、调试或监控。 FALSE： RS232 接口不可用于应用程序。装置可进行编程、调试和监控。
SUPPLY_VOLTAGE	WORD	VBBs 上的电源电压，以 [mV] 为单位
TEST	BOOL	TRUE：测试输入端已启用： • 编程模式已启用 • 可下载软件 • 可查询应用程序的状态 • 无法保护存储的软件 FALSE：应用程序在运行中

7.1.6 系统标志：输入端 16...40 和输出端 24...0 (标准侧)

19560

系统标志 (符号名称)	类型	说明
ANALOGx x = 0...7	WORD	模拟输入端 xx : 已过滤 A/D 转换器原始值 (12 位), 无校准或标准化
Ixx xx = 00...07 / 10...17 / 20...27 / 30...37 / 40...47	BOOL	二进制输入端 xx 上的状态 要求: 输入端配置为二进制输入端 (MODE = IN_DIGITAL_H 或 IN_DIGITAL_L) TRUE: 二进制输入端上的电压 > VBBs 的 70 % FALSE: 二进制输入端上的电压 < VBBs 的 30 % 或: 未配置为二进制输入端 或: 错误的配置
Ixx_MODE xx = 00...07 / 14...17 / 24...27 / 30...37 / 40...47	BYTE	输入端 Ixx 工作模式 → 章节 可能的输入端/输出端工作模式 (→ 页 278)
Qxx xx = 10...13 / 20...23 / 30...37 / 40...47	BOOL	二进制输出端 xx 上的状态: 要求: 输出端配置为二进制输出端 TRUE: 输出端启用 FALSE: 输出端禁用 (= 初始值) 或: 未配置为二进制输出端
Qxx_MODE xx = 10...13 / 20...23 / 30...37 / 40...47	BYTE	输出端 Qxx 工作模式 → 章节 可能的输入端/输出端工作模式 (→ 页 278)

7.2 地址分配和 I/O 工作模式

内容	
输入端/输入端地址分配	274
可能的输入端/输出端工作模式.....	278
I/O 地址/变量.....	282

1656

→ 另有技术资料

7.2.1 输入端/输入端地址分配

内容	
输入端：地址分配（标准侧）（输入端 16...40）	275
输出端：地址分配（标准侧）（输出端 0...24）	277

2371

输入端：地址分配（标准侧）（输入端 16...40）

19572

缩写词 → 章节 **关于配线的注意事项** (→ 页 [40](#))输入端/输出端工作模式 → 章节 **可能的输入端/输出端工作模式** (→ 页 [278](#))

IEC 地址	符号地址
%IX0.00 %IW03	I00 ANALOG0
%IX0.01 %IW3	I01 ANALOG1
%IX0.02 %IW4	I02 ANALOG2
%IX0.03 %IW5	I03 ANALOG3
%IX0.04 %IW6	I04 ANALOG4
%IX0.05 %IW7	I05 ANALOG5
%IX0.06 %IW8	I06 ANALOG6
%IX0.07 %IW9	I07 ANALOG7
%IX0.08	I10
%IX0.09	I11
%IX0.10	I12
%IX0.11	I13
%IX0.12	I14
%IX0.13	I15
%IX0.14	I16
%IX0.15	I17
%IX1.00	I20
%IX1.01	I21
%IX1.02	I22
%IX1.03	I23
%IX1.04	I24
%IX1.05	I25
%IX1.06	I26
%IX1.07	I27
%IX1.08	I30
%IX1.09	I31
%IX1.10	I32
%IX1.11	I33
%IX1.12	I34
%IX1.13	I35

附录

地址分配和 I/O 工作模式

IEC 地址	符号地址
%IX1.14	I36
%IX1.15	I37
%IX2.00	I40
%IX2.01	I41
%IX2.02	I42
%IX2.03	I43
%IX2.04	I44
%IX2.05	I45
%IX2.06	I46
%IX2.07	I47

输出端：地址分配（标准侧）（输出端 0...24）

19573

缩写词 → 章节 **关于配线的注意事项** (→ 页 [40](#))输入端/输出端工作模式 → 章节 **可能的输入端/输出端工作模式** (→ 页 [278](#))

IEC 地址	符号地址
%QX0.00	Q10
%QX0.01	Q11
%QX0.02	Q12
%QX0.03	Q13
%QX0.04	Q20
%QX0.05	Q21
%QX0.06	Q22
%QX0.07	Q23
%QX0.08	Q30
%QX0.09	Q31
%QX0.10	Q32
%QX0.11	Q33
%QX0.12	Q34
%QX0.13	Q35
%QX0.14	Q36
%QX0.15	Q37
%QX1.00	Q40
%QX1.01	Q41
%QX1.02	Q42
%QX1.03	Q43
%QX1.04	Q44
%QX1.05	Q45
%QX1.06	Q46
%QX1.07	Q47

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（标准侧）（输入端 16...40）	278
输出端： 工作模式（标准侧）（输出端 0...24）	280

2386

输入端： 工作模式（标准侧）（输入端 16...40）

19575

添加配置值可创建可能的配置组合（如允许）。

= 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
I00...I07	IN_NOMODE	Off	INPUT_ANALOG	MODE	0	00
	IN_DIGITAL_H	正	INPUT_ANALOG	MODE	1	01
	IN_CURRENT	0...20 000 µA	INPUT_ANALOG	MODE	4	04
	IN_VOLTAGE10	0...10 000 mV	INPUT_ANALOG	MODE	8	08
	IN_VOLTAGE30	0...32 000 mV	INPUT_ANALOG	MODE	16	10
	IN_RATIO	0...1 000 ‰	INPUT_ANALOG	MODE	32	20
	IN_DIAGNOSTIC	含 IN_DIGITAL_H	INPUT_ANALOG	MODE	64	40
I10...I13	IN_NOMODE	Off	INPUT_ANALOG	MODE	0	00
	IN_DIGITAL_H	正	INPUT_ANALOG	MODE	1	01
I14...I17	IN_NOMODE	关闭	INPUT_ANALOG	MODE	0	00
	IN_DIGITAL_H	正	INPUT_ANALOG	MODE	1	01
	IN_DIAGNOSTIC	含 IN_DIGITAL_H	INPUT_ANALOG	MODE	64	40
	IN_FAST	含 IN_DIGITAL_H	内部		128	80
	IN_FAST	0...30 000 Hz	FREQUENCY PHASE	频率测量		
		0.1...5 000 Hz	PERIOD	周期测量		
		0.1...5 000 Hz	PERIOD_RATIO	周期和比率测量		
		0...30 000 Hz	FAST_COUNT	计数器		
I20...I23		0...30 000 Hz	INC_ENCODER	检测编码器值		
	IN_NOMODE	关闭	INPUT_ANALOG	MODE	0	00

附录

地址分配和 I/O 工作模式

	IN_DIGITAL_H	正	INPUT_ANALOG	MODE	1	01
I24...I27	IN_NOMODE	关闭	INPUT_ANALOG	MODE	0	00
	IN_DIGITAL_H	正	INPUT_ANALOG	MODE	1	01
	IN_DIGITAL_L	负	INPUT_ANALOG	MODE	2	02
	IN_DIAGNOSTIC	含 IN_DIGITAL_H	INPUT_ANALOG	MODE	64	40
	IN_FAST	含 IN_DIGITAL_H	内部		128	80
	IN_FAST	0...1 000 Hz	FREQUENCY PHASE	频率测量		
		0.1...1 000 Hz	PERIOD	周期测量		
		0.1...1 000 Hz	PERIOD_RATIO	周期和比率测量		
		0...1 000 Hz	FAST_COUNT	计数器		
		0...1 000 Hz	INC_ENCODER	检测编码器		
I30...I37	IN_NOMODE	关闭	INPUT_ANALOG	MODE	0	00
	IN_DIGITAL_H	正	INPUT_ANALOG	MODE	1	01
	IN_DIGITAL_L	负	INPUT_ANALOG	MODE	2	02
	IN_DIAGNOSTIC	含 IN_DIGITAL_H	INPUT_ANALOG	MODE	64	40
I40...I47	IN_NOMODE	关闭	INPUT_ANALOG	MODE	0	00
	IN_DIGITAL_H	正	INPUT_ANALOG	MODE	1	01
	IN_DIAGNOSTIC	含 IN_DIGITAL_H	INPUT_ANALOG	MODE	64	40

利用以下功能块设定工作模式：

FAST_COUNT (→ 页 184)	针对快速输入脉冲的计数功能块
FREQUENCY (→ 页 186)	测量到达所选通道的信号的频率。
INC_ENCODER (→ 页 188)	针对编码器评估的递增/递减计数器功能
INPUT_ANALOG (→ 页 175)	模拟输入通道：以下对象的备选测量... • 电流 • 电压
PERIOD (→ 页 192)	测量所示通道频率和以 [μs] 为单位的周期（周期时间）
PERIOD_RATIO (→ 页 194)	测量所示周期内所示通道频率和以 [μs] 为单位的周期（周期时间）。此外，传号空号比以 [%] 为单位显示。
PHASE (→ 页 196)	读取包含快速输入端的通道对并对比信号的相位

输出端：工作模式（标准侧）（输出端 0...24）

19576

添加配置值可创建可能的配置组合（如允许）。

 = 该配置值为默认值

输出端	可能的工作模式		通过以下方式设定	值	
				十进制	十六进制
Q10...Q13	OUT_NOMODE	关闭	Qxx_MOD	0	00
	OUT_DIGITAL_H	正	Qxx_MODE	1	01
	OUT_CURRENT		Qxx_MODE	4	04
	OUT_DIAGNOSTIC		Qxx_MODE	64	40
	OUT_OVERLOAD_PROTECTION		Qxx_MODE	128	80
Q20...Q23	OUT_NOMODE	Off	Qxx_MODE	0	00
	OUT_DIGITAL_H	正	Qxx_MODE	1	01
	OUT_CURRENT		Qxx_MODE	4	04
	OUT_DIAGNOSTIC		Qxx_MODE	64	40
	OUT_OVERLOAD_PROTECTION		Qxx_MODE	128	80
Q30...Q37	OUT_NOMODE	Off	Qxx_MODE	0	00
	OUT_DIGITAL_H	正	Qxx_MODE	1	01
	OUT_DIAGNOSTIC		Qxx_MODE	64	40
	OUT_OVERLOAD_PROTECTION		Qxx_MODE	128	80
Q40, Q43, Q44, Q47	OUT_NOMODE	Off	Qxx_MODE	0	00
	OUT_DIGITAL_H	正	Qxx_MODE	1	01
	OUT_DIAGNOSTIC		Qxx_MODE	64	40
Q41, Q42, Q45, Q46	OUT_NOMODE	关闭	Qxx_MODE	0	00
	OUT_DIGITAL_H	正	Qxx_MODE	1	01
	OUT_DIGITAL_L	负	Qxx_MODE	2	02
	OUT_DIAGNOSTIC		Qxx_MODE	64	40

详细信息 → 章节 **输出端 Q10...Q13、Q20...Q23、Q30...Q37、Q40...Q47：允许的工作模式**

输出端 Q10...Q13、Q20...Q23、Q30...Q37、Q40...Q47：允许的工作模式

77

工作模式		Q10	Q11	Q12	Q13	Q20	Q21	Q22	Q23
OUT_NOMODE	关闭	X	X	X	X	X	X	X	X
OUT_DIGITAL_H	正	X	X	X	X	X	X	X	X
OUT_CURRENT		X	X	X	X	X	X	X	X
OUT_DIAGNOSTIC		X	X	X	X	X	X	X	X
OUT_OVERLOAD_PROTECTION		X	X	X	X	X	X	X	X
PWM		X	X	X	X	X	X	X	X

工作模式		Q30	Q31	Q32	Q33	Q34	Q35	Q36	Q37
OUT_NOMODE	关闭	X	X	X	X	X	X	X	X
OUT_DIGITAL_H	正	X	X	X	X	X	X	X	X
OUT_DIAGNOSTIC		X	X	X	X	X	X	X	X

工作模式		Q40	Q41	Q42	Q43	Q44	Q45	Q46	Q47
OUT_NOMODE	关闭	X	X	X	X	X	X	X	X
OUT_DIGITAL_H	正	X	X	X	X	X	X	X	X
OUT_DIGITAL_L	负	--	X	X	--	--	X	X	--
OUT_DIAGNOSTIC		X	X	X	X	X	X	X	X
OUT_OVERLOAD_PROTECTION		X	X	X	X	X	X	X	X
PWM		X	--	--	X	X	--	--	X
H 桥		--	X	X	--	--	X	X	--

7.2.3 I/O 地址/变量

内容	
输入端：地址和变量（标准侧）（输入端 16...40）	282
输出端：地址和变量（标准侧）（输出端 0...24）	284

2376

输入端：地址和变量（标准侧）（输入端 16...40）

19579

IEC 地址	I/O 变量	备注
%QB8	I00_MODE	%IX0.00 (I00) 配置字节
%QB9	I01_MODE	%IX0.01 (I01) 配置字节
%QB10	I02_MODE	%IX0.02 (I02) 配置字节
%QB11	I03_MODE	%IX0.03 (I03) 配置字节
%QB12	I04_MODE	%IX0.04 (I04) 配置字节
%QB13	I05_MODE	%IX0.05 (I05) 配置字节
%QB14	I06_MODE	%IX0.06 (I06) 配置字节
%QB15	I07_MODE	%IX0.07 (I07) 配置字节
%QB16	I14_MODE	%IX0.14 (I14) 配置字节
%QB17	I15_MODE	%IX0.15 (I15) 配置字节
%QB18	I16_MODE	%IX0.16 (I16) 配置字节
%QB19	I17_MODE	%IX0.17 (I17) 配置字节
%QB20	I24_MODE	%IX1.04 (I24) 配置字节
%QB21	I25_MODE	%IX1.05 (I25) 配置字节
%QB22	I26_MODE	%IX1.06 (I26) 配置字节
%QB23	I27_MODE	%IX1.07 (I27) 配置字节
%QB24	I30_MODE	%IX1.08 (I30) 配置字节
%QB25	I31_MODE	%IX1.09 (I31) 配置字节
%QB26	I32_MODE	%IX1.10 (I32) 配置字节
%QB27	I33_MODE	%IX1.11 (I33) 配置字节
%QB28	I34_MODE	%IX1.12 (I34) 配置字节
%QB29	I35_MODE	%IX1.13 (I35) 配置字节

附录

地址分配和 I/O 工作模式

IEC 地址	I/O 变量	备注
%QB30	I36_MODE	%IX1.14 (I36) 配置字节
%QB31	I37_MODE	%IX1.15 (I37) 配置字节
%QB32	I40_MODE	%IX2.00 (I40) 配置字节
%Q33	I41_MODE	%IX2.01 (I41) 配置字节
%QB34	I42_MODE	%IX2.02 (I42) 配置字节
%QB35	I43_MODE	%IX2.03 (I43) 配置字节
%QB36	I44_MODE	%IX2.04 (I44) 配置字节
%QB37	I45_MODE	%IX2.05 (I45) 配置字节
%QB38	I46_MODE	%IX2.06 (I46) 配置字节
%QB39	I47_MODE	%IX2.07 (I47) 配置字节
%IW3	ANALOG0	I00 模拟值
%IW4	ANALOG1	I01 模拟值
%IW5	ANALOG2	I02 模拟值
%IW6	ANALOG3	I03 模拟值
%IW7	ANALOG4	I04 模拟值
%IW8	ANALOG5	I05 模拟值
%IW9	ANALOG6	I06 模拟值
%IW10	ANALOG7	I07 模拟值

输出端：地址和变量（标准侧）（输出端 0...24）

19580

IEC 地址	I/O 变量	备注
%QB40	Q10_MODE	%QX0.00 (Q10) 配置字节
%QB41	Q11_MODE	%QX0.01 (Q11) 配置字节
%QB42	Q12_MODE	%QX0.02 (Q12) 配置字节
%QB43	Q13_MODE	%QX0.03 (Q13) 配置字节
%QB44	Q20_MODE	%QX0.04 (Q20) 配置字节
%QB45	Q21_MODE	%QX0.05 (Q21) 配置字节
%QB46	Q22_MODE	%QX0.06 (Q22) 配置字节
%QB47	Q23_MODE	%QX0.07 (Q23) 配置字节
%QB48	Q30_MODE	%QX0.08 (Q30) 配置字节
%QB49	Q31_MODE	%QX0.09 (Q31) 配置字节
%QB50	Q32_MODE	%QX0.10 (Q32) 配置字节
%QB51	Q33_MODE	%QX0.11 (Q33) 配置字节
%QB52	Q34_MODE	%QX0.12 (Q34) 配置字节
%QB53	Q35_MODE	%QX0.13 (Q35) 配置字节
%QB54	Q36_MODE	%QX0.14 (Q36) 配置字节
%QB55	Q37_MODE	%QX0.15 (Q37) 配置字节
%QB56	Q40_MODE	%QX1.00 (Q40) 配置字节
%QB57	Q41_MODE	%QX1.01 (Q41) 配置字节
%QB58	Q42_MODE	%QX1.02 (Q42) 配置字节
%QB59	Q43_MODE	%QX1.03 (Q43) 配置字节
%QB60	Q44_MODE	%QX1.04 (Q44) 配置字节
%QB61	Q45_MODE	%QX1.05 (Q45) 配置字节
%QB62	Q46_MODE	%QX1.06 (Q46) 配置字节
%QB63	Q47_MODE	%QX1.07 (Q47) 配置字节

7.3 错误表

内容	
错误标志	285
错误： CAN / CANopen	285

19606

7.3.1 错误标志

19608

→ 章节 **系统标志** (→ 页 [268](#))

7.3.2 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					说明
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	
00	80	11	--	--	--	--	--	CANx 监控 SYNC 错误 (仅从站)
00	81	11	--	--	--	--	--	CANx 警告阈值 (> 96)
10	81	11	--	--	--	--	--	CANx 接收缓冲区超限
11	81	11	--	--	--	--	--	CANx 传送缓冲区超限
30	81	11	--	--	--	--	--	CANx 保护/检测信号错误 (仅从站)

EMCY 代码：I/O，系统（标准侧）

19552

在以下情况下，以下 EMCY 消息自动发送：

- 作为 CANopen 主站：如果 **CANx_MASTER_EMCY_HANDLER** (→ 页 [117](#)) 循环调用
- 作为 CANopen 从站：如果 **CANx_SLAVE_EMCY_HANDLER** (→ 页 [128](#)) 循环调用

EMCY 代码 对象 0x1003		对象 0x1001	制造商特定信息					说明
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	
00	21	03	I0	I1	I2	I3	I4	输出端诊断
00	23	03	Q1Q2	Q3	Q4			中断时的输出端诊断
02	23	03	Q1Q2	Q3	Q4			短路时的输出端诊断
00	31	05						端子电压 VBB0/VBBs
00	33	05						输出电压 VBBr
00	42	09						过温
00	61	11						内存错误

8 Glossary of Terms

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，错误通过消息发送。

Ethernet

以太网是广泛使用且独立于制造商的技术，有助于在网络中以 10...10 000 Mbps 的速度传送数据。以太网属于非专属传送媒介中所谓的“最优数据传送”系列。该概念形成于 1972 年并在 1985 年确定为 IEEE 802.3。

EUC

EUC = **受控设备**。

EUC 即用于制造、处理、运输、医疗或其他活动的设备、机械或装置 (→ IEC 61508-4，第 3.2.3 节)。因此，EUC 是所有需要安全相关系统以防发生危害的设备、机械或装置的集合。

如果任何合理预见的行为或不作为导致→危害且无法容忍的风险源于 EUC，则需要使用安全功能以达到或维护 EUC 的安全状态。这些安全功能通过一个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**

Glossary of Terms

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 [291](#))

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 = **协议数据单元**。

PDU 是 →CAN 协议 →SAE J1939 的一项。PDU 表示目标或源地址的一部分。

PES

可编程电子系统...

- 用于控制、保护或监控，
- 其工作依赖于一个或多个可编程电子装置，
- 包括输入和输出装置等所有系统元件。

PGN

PGN = **参数组编号**

PGN = PDU 格式 (PF) + PDU 源 (PS)

参数组编号是 →CAN 协议 →SAE J1939 的一项。PGN 收集地址部分 PF 和 PS。

PID 控制器

PID 控制器 (比例-积分-微分) 包含以下单元：

- P = 比例单元
- I = 积分单元
- D = 微分单元 (但不适用于控制器 CR04nn、CR253n)。

PLC 配置

CODESYS 用户界面的部分

- ▶ 程序员告知编程系统哪些硬件有待编程。
- > CODESYS 加载相应的库。
- > 可读取和写入外围状态（输入端/输出端）。

PWM

PWM = 脉冲宽度调制

PWM 输出信号即 GND 和电源电压之间的脉冲信号。

在确定的周期（PWM 频率）内，传号空号比是不同的。连接的负载决定相应的 RMS 电流，具体视传号空号比而定。

R

RAW-CAN

RAW-CAN 指在 CAN 总线（ISO/OSI 第 2 层）上没有额外通信协议的情况下行之有效的纯粹 CAN 协议。CAN 根据 ISO 11898-1 在国际范围内定义，此外还根据 ISO 16845，确保 CAN 芯片的可交换性。

ro

RO = 只读

单向数据传送：数据仅可读取，不可更改。

RTC

RTC = 实时时钟

提供（电池支持）当前的日期和时间。常用于存储错误消息协议。

rw

RW = 读取/写入

双向数据传送：数据既可读取，也可更改。

S

SAE J1939

网络协议 SAE J1939 讲述针对诊断数据（如发动机速度、温度）和控制信息传送的商业车辆 →CAN 总线通信。

标准：串行控制和通信车辆网络建议实践

- 第 2 部分：农林越野机械控制和通信网络
- 第 3 部分：车载诊断实施指南
- 第 5 部分：船尾驱动器和船内火花点火发动机车载诊断实施指南
- 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 数据连接均有一个发送器和一个接收器。该原理为连接导向数据传输。在 TCP/IP 协议系列中，作为连接导向协议，TCP 承担数据保护、数据流控制的任务，并在数据丢失时采取措施。（比较：→UDP）

U

UDP

UDP (**用户数据报协议**) 是最低限度无连接网络协议，属于网络协议系列中的传输层。UDP 的任务是确保通过互联网传输的数据传送给适当的应用。

目前可实施基于 →CAN 和 UDP 的网络变量。变量的值根据广播消息自动更换。在 UDP 中，它们作为广播消息实施，在 CAN 中则作为 →PDO 实施。

根据协议，这些服务为未经确认的数据传送：未检查接收器是否接收消息。网络变量交换对应“1 到 n 个连接”（1 个变送器到 n 个接收器）。

三划

已停止

CANopen 参与者的工作状态。在该模式中仅可传输 →NMT 命令。

四划

从站

总线上的被动参与者，仅依赖→主站的请求。在总线中，从站有明确定义且唯一的→地址。

比率式

还可进行比率测量。如果传感器输出信号与其电源电压成比例，则可通过比率测量（= 与电源成比例的测量）减少电源波动的影响，在理想情况下，甚至可避免该影响。

→ 模拟输入端

五划

主站

处理总线的整体安排。主站决定总线访问时间并定期→轮询从站。

对象目录

包含装置的所有 CANopen 通信参数以及装置特定参数和数据。

电子狗

一般而言，“电子狗”一词表示监控其他组件功能的系统组件。如果检测到可能的故障，则向其发送信号或启用相应的程序分支。信号或程序分支作为其他协作系统组件的触发器，以解决问题。

目标

目标包含 CODESYS 目标装置的硬件说明，如：输入端和输出端、内存、文件位置。
对应电子技术资料。

节点

即网络中的参与者。

节点保护

节点 = 此处：网络参与者

每个从站的可配置循环→监控可相应地加以配置。 →主站核查从站是否及时响应。 从站核查主站是否定期发送请求。 这样即可快速识别和报告出现故障的网络参与者。

闪存

快闪 ROM (或快闪 EPROM 或闪存) 将半导体内存和硬盘的优势相结合。 但是跟硬盘类似，数据在高达 64、128、256、1024... 字节的数据块中同时按区块写入和删除。

闪存的优势

- 即便无电源电压亦可保留存储的数据。
- 因为没有移动部件，闪存无噪音且对冲击和磁场不敏感。

闪存的劣势

- 存储单元可容许有限数量的写入和删除进程：
 - 多级单元：一般为 10 000 个周期
 - 单级单元：一般为 100 000 个周期
- 鉴于写入进程同时写入 16 和 128 K 字节之间的内存块，还可使用无需更改的内存单元。

六划

地址

这是总线参与者的“名称”。 所有参与者需要唯一的地址，以便无忧更换信号。

夹具 15

在车辆中，夹具 15 是通过点火锁开关的正极电缆。

自检

主动检验组件或装置的检验程序。 程序由用户启动，且将花费一定的时间。 结果为显示检验内容以及结果为正面还是负面的检验协议（日志文件）。

七划

应用程序软件

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

抖动频率

抖动频率是 →PWM 信号的组成部分，用于控制液压阀。 对液压阀的电磁驱动器来说，如果 PWM 频率的特定频率叠加控制信号（PWM 脉冲），则控制阀门要容易得多。 该抖动频率必须是 PWM 频率的整数部分。

系统变量

可通过 IEC 地址或符号名称从 PLC 访问的变量。

诊断

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

还可监控输入端和输入端是否正常运行，具体视装置而定。

- 断线，
- 短路，
- 值在范围之外。

就诊断而言，可使用装置“正常”运行期间创建的配置和日志数据。

初始化和启动期间可监控系统组件是否正确启动。 日志文件记录错误。

就进一步诊断而言，还可进行自检。

运行

CANopen 参与者的工作状态 在该模式中可传输→SDO、→NMT 命令和 →PDO。

运行时系统

装置中的基本程序，建立装置硬件和应用程序之间的连接。

进程图像

进程图像即在一个→周期内 PLC 据其运行的输入端和输出端状态。

- 在周期开始时，PLC 将所有输入端的状况读取至进程图像。
周期期间，PLC 无法检测输入端的更改。
- 周期期间，仅可对输出端进行虚拟更改（在进程图像中）。
- 在周期结束时，PLC 将虚拟输出状态写入实际输出端。

连接指令

连接指令是指对文件中其他部分或外部文件的交叉引用。

八划

使用，指定

根据用途说明所载的信息使用产品。

周期时间

即周期的时间。PLC 程序展开一次完整的运行。

这可能需要更长或更短的时间，具体视程序中的事件控制分支而定。

图标

图标即通过简化的图示传达信息的象征性符号。（→ 章节 **符号和格式是什么意思？**（→ 页 [7](#)））

波特

波特 (Baud)，缩写：Bd = 数据传送速度单位。切勿将波特与“位/秒 (bps、bits/s)”混淆。波特表示在一定传送长度内每秒的状态（步骤、周期）更改量。但未定义每步传送的位数。Baud 一词可追溯到法国发明家 J. M. Baudot,其编码用于电传机器。

1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

九划

总线

同一电缆上多个参与者的串行数据传送。

指定用途

根据用途说明所载的信息使用产品。

架构

系统硬件和/或软件的特定配置。

误用

未按照设计方指定的方式使用产品。

产品制造商应在用户信息中针对可预见的误用提出警告。

说明

以下术语之一的上义词：

安装说明、技术资料、用户信息、操作说明、装置手册、安装信息、联机帮助、系统手册、编程手册等。

十划

预运行

预运行 = PRE-OPERATIONAL 模式

CANopen 参与者的工作状态。施加电源电压后，每个参与者自动进入该状态。在 CANopen 网络中，仅→SDO 和 →NMT 命令可在该模式下传输，但无进程数据。

十一划

检测信号

参与者定期发送短信号。这样其他参与者则可核实参与者是否出现故障。

符号

图标即通过简化的图示传达信息的象征性符号。（→ 章节 **符号和格式是什么意思？**（→ 页 [7](#)））

十二划

剩余

电源故障时剩余数据不会丢失。

当电源电压跌破关键值时→运行时系统等即会自动将剩余数据复制到→闪存。 如果电源电压再次可用，则运行时系统将剩余数据加载回 RAM 内存。

但是控制器 RAM 内存中的数据易失，通常会在电源故障时丢失。

嵌入式软件

装置中的系统软件、基本程序，实际上→是运行时系统中。

固件建立装置硬件和应用程序之间的连接。 固件作为系统的一部分由控制器制造商提供，且用户不可更改。

十三划

数据类型

可存储不同大小的值，具体式数据类型而定。

数据类型	最小值	最大值	内存大小
BOOL	FALSE	TRUE	8 位 = 1 个字节
BYTE	0	255	8 位 = 1 个字节
WORD	0	65 535	16 位 = 2 个字节
DWORD	0	4 294 967 295	32 位 = 4 个字节
SINT	-128	127	8 位 = 1 个字节
USINT	0	255	8 位 = 1 个字节
INT	-32 768	32 767	16 位 = 2 个字节
UINT	0	65 535	16 位 = 2 个字节
DINT	-2 147 483 648	2 147 483 647	32 位 = 4 个字节
UDINT	0	4 294 967 295	32 位 = 4 个字节
REAL	$-3.402823466 \cdot 10^{38}$	$3.402823466 \cdot 10^{38}$	32 位 = 4 个字节
ULINT	0	18 446 744 073 709 551 615	64 位 = 8 个字节
STRING			字符数量 + 1

十四划

模板

模板可填入内容。

此处： 预配置软件元件结构，作为应用程序的基础。

索引

9 索引

A

ANALOG_RAW174

B

Bootloader.....51, 287

Bootloader 状态.....61

C

CAN287

接口和协议.....48

CAN / CANopen

错误和错误处理.....267

CAN 堆栈.....287

CAN 接口.....48

CAN1_BAUDRATE.....95

CAN1_DOWNLOADID96

CAN1_EXT97

CAN1_EXT_ERRORHANDLER.....98

CAN1_EXT_RECEIVE99

CAN1_EXT_TRANSMIT.....101

CAN2104

CANx_ERRORHANDLER.....105

CANx_EXT_RECEIVE_ALL.....107

CANx_MASTER_EM CY_HANDLER117

CANx_MASTER_SEND_EMERGENCY.....118

CANx_MASTER_STATUS121

CANx_RECEIVE.....109

CANx_RECEIVE_RANGE.....111

CANx_SDO_READ.....138

CANx_SDO_WRITE.....140

CANx_SLAVE_EM CY_HANDLER.....128

CANx_SLAVE_NODEID.....130

CANx_SLAVE_SEND_EMERGENCY.....131

CANx_SLAVE_STATUS134

CANx_TRANSMIT.....114

CHECK_DATA.....258

CiA287

CiA DS 304.....287

CiA DS 401.....287

CiA DS 402.....287

CiA DS 403.....288

CiA DS 404.....288

CiA DS 405.....288

CiA DS 406.....288

CiA DS 407.....288

COB ID.....288

CODESYS288

CODESYS 中的 FB、FUN、PRG54

CODESYS 项目的编程说明.....53

CODESYS 编程手册.....7

CONTROL_OCC213

CSV 文件.....289

D

DC.....289

DEBUG 模式.....63

DELAY232

DLC.....289

DRAM289

DTC.....289

E

ECU289

EDS 文件.....290

EMC.....290

EMCY.....290

EMCY 代码：CANx.....285

EMCY 代码：I/O，系统（标准侧）.....286

Emergency_Message 结构.....126

Ethernet290

EUC290

F

FAST_COUNT.....184

FB 使用的限制.....64

FiFo.....290

FLASHREAD.....251

FLASH-Speicher.....16

FLASHWRITE.....252

FRAM.....17, 291

FRAM 内存.....246

FRAMREAD.....254

FRAMWRITE.....255

FREQUENCY186

G

GET_IDENTITY260

GLR.....233

H

HMI291

I

I/O 地址/变量.....282

ID291

IEC 61131.....291

IEC 用户周期.....291

IFM 功能元件.....87

INC_ENCODER.....188

INIT 状态（复位）.....61

索引

INPUT_ANALOG	175
INPUT_CURRENT	178
INPUT_VOLTAGE	179
Inputs I20...23	29
IP 地址	291
ISO 11898	292
ISO 11992	292
ISO 16845	292

J

J1939	292
J1939_x	143
J1939_x_GLOBAL_REQUEST	144
J1939_x_RECEIVE	147
J1939_x_RESPONSE	151
J1939_x_SPECIFIC_REQUEST	154
J1939_x_TRANSMIT	157
JOYSTICK_0	216
JOYSTICK_1	220
JOYSTICK_2	224

L

LED	45, 292
LSB	292

M

MAC-ID	292
MEMCPY	256
MEMORY_RETAIN_PARAM	248
MMI	294
MRAM	294
MSB	294

N

NMT	294
NORM	181
NORM_HYDRAULIC	227

O

Obj /对象	294
OBV	294
OCC_TASK	199
OPC	294
OUTPUT_CURRENT	201
OUTPUT_CURRENT_CONTROL	202

P

PC 卡	294
PCMCIA 卡	295
PDM	295
PDO	295
PDU	295
PERIOD	192

PERIOD_RATIO	194
PES	295
PGN	295
PHASE	196
PID 控制器	235, 237, 295
PID1	235
PID2	237
PLC 配置	70, 296
PT1	240
PWM	204, 296
PWM 抖动频率	207
PWM 输出端	31, 80
PWM 频率	205
PWM100	208
PWM1000	210

R

RAW-CAN	296
RELOAD 值计算示例	206
ro	296
RTC	296
RUN 状态	61
rw	296

S

SAE J1939	142, 297
SD 卡	297
SDO	297
SERIAL_MODE	63
SERIAL_PENDING	161
SERIAL_RX	162
SERIAL_SETUP	164
SERIAL_TX	166
SET_DEBUG	262
SET_IDENTITY	263
SET_INTERRUPT_I	168
SET_INTERRUPT_XMS	171
SET_PASSWORD	264
SOFTRESET	242
SRAM	16
STOP 状态	61
SYSTEM STOP 状态	62

T

TCP	297
TEST 模式	62
TIMER_READ	244
TIMER_READ_US	245

U

UDP	298
-----------	-----

索引**一划**

一次性机制	23
一般功能配置	73

二划

二进制和 PWM 输出端	80
二进制输入端	25, 76
二进制输出端	31

三划

工作状态	58
工作状态：运行时系统不可用	58
工作状态：应用程序不可用	59
工作状态：应用程序可用	60
工作原理	21
工作模式	62
已停止	298

四划

无运行时系统	61
比率式	298
中断	64
内存，可用	16
内部结构的参数	124
手动	250
手动设定编程系统	69
手动数据存储	250
从站	298
从站信息	125
计算 RELOAD 值	206
双向输入端/输出端上的反馈	41

五划

功能元件：CAN 第 2 层	94
功能元件：CANopen 从站	127
功能元件：PWM 功能	198
功能元件：处理输入值	173
功能元件：时间测量/设定	243
功能元件：串行接口	160
功能元件：针对频率和周期测量的计数器功能	183
功能元件：保存、读取和转换内存中的数据	246

功能元件：调整模拟值	180
功能元件：通过处理中断优化 PLC 周期	167
功能元件：控制器	230
功能元件：液压控制	212
功能配置	73, 74
节点	299
节点状态结构	125
节点保护	299
本文档的结构是怎样的？	8
可用内存	16
可能的输入端/输出端工作模式	278
目标	299
电子狗	65, 298
电子狗状况	65
处理输入值	173
外部供给输出端反馈	43
主站	298
闪存	246, 299
发生错误时的反应	266
对象目录	298

六划

地址	299
地址分配	274
地址分配和 I/O 工作模式	274
过冲阻尼	231
在应用程序中控制 LED	46
夹具 15	299
网络变量	86
延迟关闭的工作原理	18
自检	300
创建应用程序	56
关于本手册	5
关于装置的信息	13
关于簧片继电器的安全说明	40, 75
安全说明	10
设定目标	70
设定运行时系统	66
设定规则	230
设定控制	230
访问应用程序运行时结构	126
如果 TEST 插脚未启用	23

索引

如果运行时系统/应用程序正在运行22

七划

进程图像301

运行61, 300

运行时系统51, 300

抖动频率207, 300

更新运行时系统68

连接指令301

串行接口47, 160

串行模式63

针对装置 CR0020 的 IFM 功能元件94

针对装置 CR0020 的 IFM 库87

系统时间243

系统变量73, 300

系统标志268

系统标志：CAN269

系统标志：SAE J1939269

系统标志：电压（标准侧）272

系统标志：状态 LED（标准侧）271

系统标志：输入端 16...40 和输出端 24...0（标准侧）273

系统标志：错误标志（标准侧）270

系统描述13

状态 LED45

库52

库 ifm_CAN1_EXT_Vxxyzz.LIB92

库 ifm_CR0020_CANOpenMaster_V04yynn.LIB91

库 ifm_CR0020_CANOpenSlave_V04yynn.LIB91

库 ifm_CR0020_V06yzz.LIB88

库 ifm_hydraulic_16bitOS05_Vxxyzz.LIB93

库 ifm_J1939_x_Vxxyzz.LIB92

应用程序51

应用程序中的计算和转换55

应用程序软件300

快速输入端77

启用 PLC 配置（如 CR003371

启动条件14

诊断265, 300

诊断：二进制输出端（通过电压测量）36, 38

诊断：二进制输出端（通过电流测量）33

诊断：过载36, 38

诊断：过载（通过电流测量）33

诊断：断线（通过电压测量）36, 39

诊断：断线（通过电流测量）34

诊断：短路（通过电压测量）36, 39

诊断：短路（通过电流测量）34

诊断和错误处理265

附录268

八划

软件复位241

软件控制器配置70

图标7, 301

使用 IFM 下载程序57

使用 IFM 维护工具57

使用，指定301

例如：CANx_MASTER_SEND_EMERGENCY120

例如：CANx_MASTER_STATUS125

例如：CANx_RECEIVE_RANGE 在 4 个周期中的初始化113

例如：CANx_SLAVE_SEND_EMERGENCY133

例如：CHECK_DATA259

例如：NORM (1)182

例如：NORM (2)182

例如：NORM_HYDRAULIC229

版权5

周期时间301

备注310

注释：TEST 输入端12, 62

注释：序列号12

注意周期时间！55

波特301

建议的设定236, 239

九划

指定用途302

故障265

响应系统错误267

重新安装运行时系统67

复位61

保存、读取和转换数据246

保存启动项目57

保持变量86

将端子 VBB0 (5) 连接至电池（未开启）。18

将端子 VBBs (23) 连接至点火开关。18

索引

施加电源电压后	22
总线	301
测试	62
误用	302
说明 ... 95, 96, 97, 98, 99, 101, 104, 105, 107, 109, 111, 114, 117, 118, 121, 128, 130, 131, 134, 138, 140, 143, 144, 147, 151, 154, 157, 161, 162, 164, 166, 168, 171, 174, 175, 178, 179, 181, 184, 186, 188, 192, 194, 196, 199, 201, 202, 204, 208, 210, 213, 216, 220, 224, 227, 232, 233, 235, 237, 240, 242, 244, 245, 248, 251, 252, 254, 255, 256, 258, 260, 262, 263, 264, 302	
说明沿革 (CR0020、CR0505)	9
架构	302

十划

配线	40
配置	66
配置输入端和输出端（默认设定）	73
配置输出端	79
原理方块图	15
监控电源电压 VBBr	21
监控和保护装置	22
监控概念	20
监控概念的工作原理	21
高于平均的压力	64
请注意！	10
调试	63
调整过程值	180
通过模板设定编程系统	72
预先具备的知识	11
预运行	302
继电器	14
继电器：重要说明！	19, 266

十一划

接口说明	47
控制器设定规则	230
控制器的启动运行状况	12
检测信号	302
检验安装	68
符号	302
符号和格式是什么意思？	7
停止	61
斜坡功能	207

十二划

硬件说明	13
硬件结构	14
嵌入式软件	303
锁存	18
剩余	302
装置的软件模块	50
装置的性能极限	64

十三划

概述：ecomatmobile 装置文档模块	6
输入组 group I40...I47	30
输入组 I00...I07 / ANALOG0...7	26
输入组 I10...I17 / FRQ0...FRQ3	28
输入组 I20...I27	29
输入组 I30...I37	30
输入端 I10...13	28
输入端 I14...17 / FRQ0...3	28
输入端 I24...27 / CYL0...3	29
输入端（技术）	24
输入端/输入端地址分配	274
输入端/输出端工作模式	278
输入端：工作模式（标准侧）（输入端 16...40）	278
输入端：地址分配（标准侧）（输入端 16...40）	275
输入端：地址和变量（标准侧）（输入端 16...40）	282
输入端地址分配	275
输入端参数 95, 96, 97, 98, 99, 101, 104, 106, 107, 109, 112, 114, 117, 119, 121, 128, 130, 132, 135, 139, 141, 143, 145, 149, 152, 155, 158, 162, 165, 166, 170, 172, 175, 178, 179, 181, 185, 187, 190, 193, 195, 196, 200, 201, 203, 205, 209, 211, 214, 218, 221, 224, 228, 232, 234, 236, 238, 240, 242, 249, 251, 253, 254, 255, 256, 258, 260, 262, 263, 264	
输出组 Q10...Q13 / Q20...Q23	32
输出组 Q1Q2 (Q10...13 / Q20...23)	82
输出组 Q3 (Q30...37)	83
输出组 Q30...Q37	35
输出组 Q4 (Q40...47)	84
输出组 Q40...Q47	37
输出端 Q10...Q13、Q20...Q23、Q30...Q37、Q40...Q47：允许的工作模式	281
输出端：工作模式（标准侧）（输出端 0...24）	280
输出端：地址分配（标准侧）（输出端 0...24）	277
输出端：地址和变量（标准侧）（输出端 0...24）	284

索引

输出端上的反馈	41
输出端地址分配	277
输出端参数 100, 103, 108, 110, 112, 115, 117, 123, 129, 136, 139, 141, 146, 150, 152, 156, 159, 161, 163, 174, 177, 178, 179, 181, 185, 187, 191, 193, 195, 197, 200, 201, 203, 215, 218, 222, 225, 228, 232, 234, 236, 238, 240, 244, 245, 259, 260	
错误：CAN / CANopen	285
错误表	285
错误标志	285
数字输出端	80
数据自动保存	247
数据访问和数据检查	257
数据备份存储类型	246
数据类型	303

十四划

模拟输入端	24, 78
模板	303
需要预先具备哪些知识？	11

10 备注

备注	错误表
----	-----

11 ifm weltweit • ifm worldwide • ifm à l'échelle internationale

Version: 2016-11-29

8310

ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen

www.ifm.com • Email: info@ifm.com

Service hotline: 0800 / 16 16 16 (only Germany, Mo-Fr 07.00...18.00 h)

ifm Niederlassungen • Sales offices • Agences

D	Niederlassung Nord • 31135 Hildesheim • Tel. 0 51 21 / 76 67-0 Niederlassung West • 45128 Essen • Tel. 02 01 / 3 64 75 -0 Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 0 23 51 / 43 01-0 Niederlassung Süd-West • 64646 Heppenheim • Tel. 0 62 52 / 79 05-0 Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 0 70 21 / 80 86-0 Niederlassung Bayern • 82178 Puchheim • Tel. 0 89 / 8 00 91-0 Niederlassung Ost • 07639 Tautenhain • Tel. 0 36 601 / 771-0
A, SL	ifm electronic gmbh • 1120 Wien • Tel. +43 16 17 45 00
AUS	ifm efector Pty Ltd. • Mulgrave Vic 3170 • Tel. +61 3 00 365 088
B, L	ifm electronic N.V. • 1731 Zellik • Tel. +32 2 / 4 81 02 20
BG	ifm electronic eood • 1202 Sofia • Tel. +359 2 807 59 69
BR	ifm electronic Ltda. • 03337-000, Sao Paulo SP • Tel. +55 11 / 2672-1730
CH	ifm electronic ag • 4 624 Härkingen • Tel. +41 62 / 388 80 30
CL	ifm electronic SpA • Oficina 5032 Comuna de Conchalí • Tel. +55 11 / 2672-1730
CN	ifm electronic (Shanghai) Co. Ltd. • 201203 Shanghai • Tel. +86 21 / 3813 4800
CND	ifm efector Canada inc. • Oakville, Ontario L6K 3V3 • Tel. +1 800-441-8246
CZ	ifm electronic spol. s.r.o. • 25243 Průhonice • Tel. +420 267 990 211
DK	ifm electronic a/s • 2605 BROENDBY • Tel. +45 70 20 11 08
E	ifm electronic s.a. • 08820 El Prat de Llobregat • Tel. +34 93 479 30 80
F	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • Tél. +33 0820 22 30 01
FIN	ifm electronic oy • 00440 Helsinki • Tel. +358 75 329 5000
GB, IRL	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • Tel. +44 208 / 213-0000
GR	ifm electronic Monoprosopi E. 7 E. • 15125 Amaroussio • Tel. +30 210 / 6180090
H	ifm electronic kft. • 9028 Győr • Tel. +36 96 / 518-397
I	ifm electronic s.a. • 20041 Agrate-Brianza (MI) • Tel. +39 039 / 68.99.982
IL	Astragal Ltd. • Azur 58001 • Tel. +972 3 -559 1660
IND	ifm electronic India Branch Office • Kolhapur, 416234 • Tel. +91 231-267 27 70
J	efector co., ltd. • Chiba-shi, Chiba 261-7118 • Tel. +81 043-299-2070
MAL	ifm electronic Pte. Ltd • 47100 Puchong Selangor • Tel. +603 8063 9522
MEX	ifm efector S. de R. L. de C. V. • Monterrey, N. L. 64630 • Tel. +52 81 8040-3535
N	Sivilingeniør J. F. Knudtzen A/S • 1396 Billingstad • Tel. +47 66 / 98 33 50
NA	ifm electronic (pty) Ltd • 25 Dr. W. Kulz Street Windhoek • Tel. +264 61 300984
NL	ifm electronic b.v. • 3843 GA Harderwijk • Tel. +31 341 / 438 438
NZ	ifm efector Pty Ltd • 930 Great South Road Penrose, Auckland • Tel. +64 95 79 69 91
P	ifm electronic s.a. • 4410-136 São Félix da Marinha • Tel. +351 223 / 71 71 08
PL	ifm electronic S 7 z o.o. • 40-106 Katowice • Tel. +48 32-608 74 54
RA, ROU	ifm electronic s.r.l. • 1107 Buenos Aires • Tel. +54 11 / 5353 3436
RO	ifm electronic s.r.l. • Sibiu 557260 • Tel. +40 269 224550
ROK	ifm electronic Ltd. • 140-884 Seoul • Tel. +82 2 / 790 5610
RUS	ifm electronic • 105318 Moscow • Tel. +7 495 921-44-14
S	ifm electronic a b • 41250 Göteborg • Tel. +46 31 / 750 23 00
SGP	ifm electronic Pte. Ltd. • Singapore 609 916 • Tel. +65 6562 8661/2/3
SK	ifm electronic s.r.o. • 835 54 Bratislava • Tel. +421 2 / 44 87 23 29
THA	SCM Alliances Co., Ltd. • Bangkok 10 400 • Tel. +66 02 615 4888
TR	ifm electronic Ltd. Sti. • 34381 Sisli/Istanbul • Tel. +90 212 / 210 50 80
UA	TOV ifm electronic • 02660 Kiev • Tel. +380 44 501 8543
USA	ifm efector inc. • Exton, PA 19341 • Tel. +1 610 / 5 24-2000
VN	ifm electronic • Ho Chi Minh city 700000 • Tel. +84-8-35125177
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • Tel. +27 12 345 44 49

Technische Änderungen behalten wir uns ohne vorherige Ankündigung vor.

We reserve the right to make technical alterations without prior notice.

Nous nous réservons le droit de modifier les données techniques sans préavis.

我们保留事先不予通知，即执行技术改造的权利。