

CE



原版编程手册
BasicDisplay

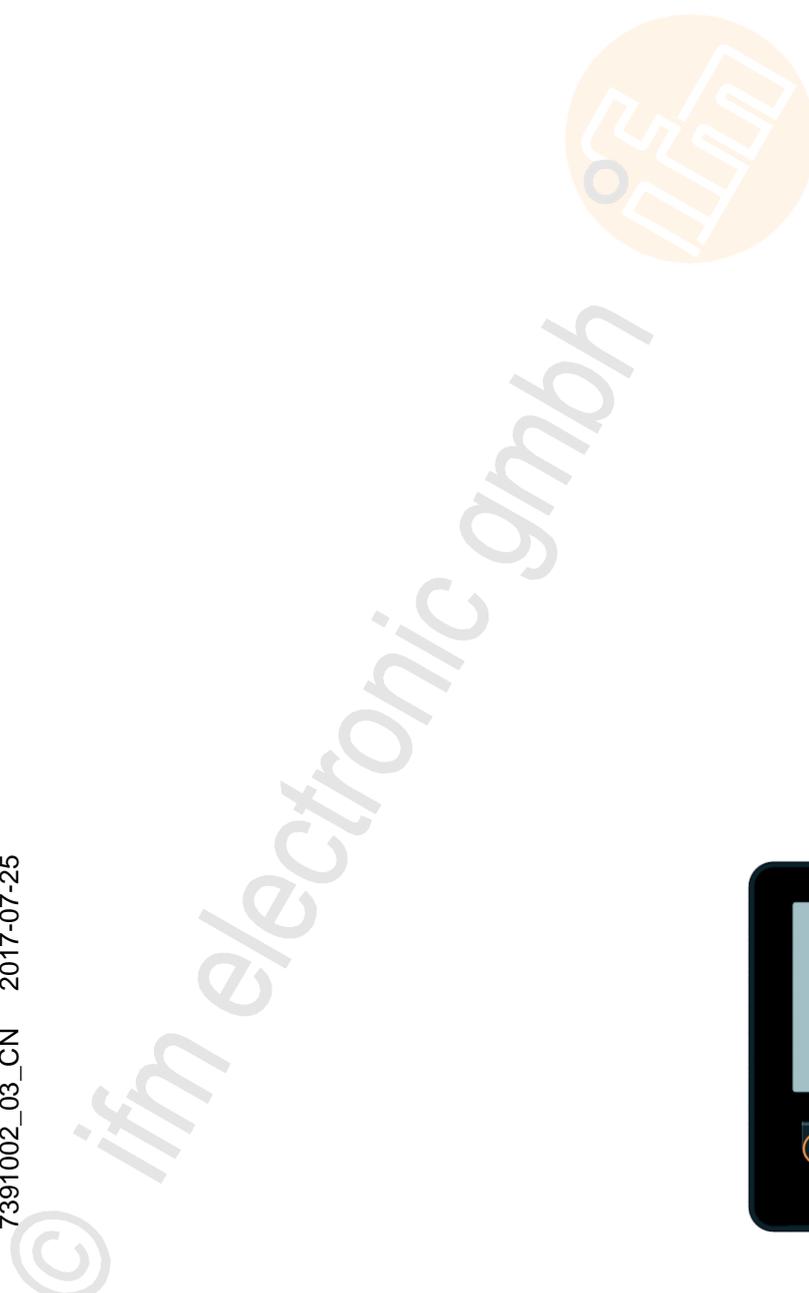
CR0452

运行时系统 v03.02

CODESYS® v2.3

中文

7391002_03_CN 2017-07-25



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	符号和格式是什么意思？	7
1.4	本文档的结构是怎样的？	8
1.5	说明沿革 (CR0452).....	8
2	安全说明	9
2.1	请注意	9
2.2	需要预先具备哪些知识？	10
2.3	控制器的启动运行状况	10
2.4	注释： 序列号	11
2.5	注释： TEST 输入端	11
3	系统描述	12
3.1	关于装置的信息	12
3.1.1	附件	12
3.2	硬件说明	13
3.2.1	硬件设定	13
3.2.2	状态 LED.....	17
3.3	接口说明	18
3.3.1	CAN 接口	18
3.4	软件说明	19
3.4.1	装置的软件模块	19
3.4.2	CODESYS 项目的编程说明	22
3.4.3	工作状态	27
3.4.4	装置的性能极限	30
4	配置	39
4.1	设定运行时系统	39
4.1.1	重新安装运行时系统	40
4.1.2	更新运行时系统	42
4.1.3	检验安装	42
4.2	设定编程系统	43
4.2.1	手动设定编程系统	43

内容

4.2.2	通过模板设定编程系统.....	48
4.3	一般功能配置.....	49
4.3.1	系统变量.....	49
4.4	变量.....	50
4.4.1	保留变量.....	50
4.4.2	网络变量.....	51
5	IFM 功能元件	52
5.1	针对装置 CR0452 的 IFM 库.....	52
5.1.1	所需的库.....	52
5.1.2	库 ifm_CR0452_Vxxyzz.LIB.....	53
5.1.3	库 ifm_CR0452_Init_Vxxyzz.LIB.....	54
5.1.4	库 ifm_PDMsmart_util_Vxxyzz.LIB.....	54
5.1.5	库 ifm_RAWCan_NT_Vxxyzz.LIB.....	54
5.1.6	库 ifm_CANopen_NT_Vxxyzz.LIB.....	55
5.1.7	库 ifm_J1939_NT_Vxxyzz.LIB.....	57
5.2	针对装置 CR0452 的 IFM 功能元件.....	59
5.2.1	输出端功能元件.....	60
5.2.2	功能元件：RAW-CAN (第 2 层).....	61
5.2.3	功能元件：CANopen.....	98
5.2.4	功能元件：SAE J1939.....	155
5.2.5	功能元件：系统.....	198
5.2.6	功能元件：图形.....	225
6	诊断和错误处理	237
6.1	诊断.....	237
6.2	故障.....	237
6.3	响应系统错误.....	238
6.3.1	响应错误消息的进程示例.....	238
6.4	CAN / CANopen: 错误和错误处理.....	238
7	附录	239
7.1	系统标志.....	239
7.2	错误表.....	241
7.2.1	错误标志.....	241
7.2.2	错误：CAN / CANopen.....	241

内容

8	专业术语	243
9	索引	260
10	备注	265
11	ifm weltweit • ifm worldwide • ifm à l'échelle internationale	269



1 关于本手册

内容

版权	5
概述：ecomatmobile 装置文档模块	6
符号和格式是什么意思？	7
本文档的结构是怎样的？	8
说明沿革 (CR0452)	8

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.
- 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] > CR0452 > [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： <ul style="list-style-type: none"> • 概述模板和演示程序 • CAN、CANopen • 控制输出端 • 用户闪存 • 可视化 • 使用的文件和库的概述
来源	→ www.ifm.com
系统手册“IFM 控制器 ISOBUS”	
内容	装置 ISOBUS 软件配置和功能的说明
来源	→ www.ifm.com 因此，“CRnnnn”表示预装 ISOBUS 的装置的产品编号。

*) 括号中的描述仅包含于特定装置的说明中。

1.3 符号和格式是什么意思？

203

以下符号或图表利用图片阐明我们的说明中的注释。

⚠ 警告	
可能导致死亡或严重的不可逆伤害。	
⚠ 小心	
可能导致轻微的可逆伤害。	
通知	
预计或可能会导致财产损失。	
❗	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

1.4 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]

1.5 说明沿革 (CR0452)

15324

本手册更改了哪些内容？概述

日期	主题	更改：
2014-03-24	可视化极限	关于允许绘图区的信息
2014-04-29	FB CAN_REMOTE_RESPONSE	功能块 ENABLE 的更多精确说明
2014-05-12	CAN 限制	已针对 CAN、CANopen 和 CAN J1939 添加限制
2014-06-30	文档名称	“系统手册”重新命名为“编程手册”
2015-01-13	针对错误代码、系统标志的文档结构	<ul style="list-style-type: none"> • 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
2017-01-13	CODESYS 2.3 软件手册	从 IFM 主页下载的提示已移除

2 安全说明

内容

请注意.....	9
需要预先具备哪些知识？.....	10
控制器的启动运行状况.....	10
注释：序列号.....	11
注释：TEST 输入端.....	11

213

2.1 请注意

6091
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]

2.2 需要预先具备哪些知识？

215

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

若要对 PLC 进行编程，用户还应熟悉 CODESYS 软件。

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

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

2.3 控制器的启动运行状况

6827
15233
11575

警告

由于机器或设备部分意外和危险启动导致的危险！

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

诸如以下情况可能会导致重新启动：

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

若要确保控制器的安全运行状况：

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

2.4 注释：序列号

20780

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

2.5 注释：TEST 输入端

20781

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



3 系统描述

内容

关于装置的信息	12
硬件说明	13
接口说明	18
软件说明	19

975

3.1 关于装置的信息

15407

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

- BasicDisplayXL: CR0452

显示屏是以下系列的一部分： BasicController: CR040n, CR041n, CR043n.

3.1.1 附件

15406

一系列附件适用于 BasicDisplay： 示例：

EC0404	CR0452 正面面板安装机架
EC0406	RAM 托架组件，可将 CR0452 作为桌面装置
EC0452	使用罩盖 EC0402 时显示屏和 BasicController 之间的电源线和 CAN 缆线
EC0454	显示屏和 BasicController 之间的 5 m 电源线和 CAN 缆线
---	用于更新固件、运行时系统和应用程序的“维护工具”软件 → ecomatmobile DVD“软件、工具和文档”

您可在 **IFM** 网站找到针对产品的附件：

→ www.ifm.com > 选择您所在国家/地区 > [Data sheet search] > 产品编号 > [Accessories]

3.2 硬件说明

内容

硬件设定	13
状态 LED	17

14081

3.2.1 硬件设定

内容

可用内存	14
接口	15
CR0452 彩色显示屏	15
CR0452 操作元件	16
按键 LED 可调暗	16
外壳后面板上的连接	16

15269

防护等级 IP 65

面板安装：防护等级 IP 67

可用内存

13736

FLASH-Speicher

13053

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

由此，如下内存区域将预留用于 ...

应用程序最大大小	512K 字节
除应用程序之外的数据 通过 FB 读取数据 FLASH_READ (→ 页 201) (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

14027

SRAM (易失快速内存) 装置现有全部 SRAM 在此表示所有类型的易失快速内存。	592K 字节
--	---------

由此，如下内存区域将预留用于 ...

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

剩余的内存保留用于系统内部用途。

FRAM

2262

FRAM (非易失快速内存) 装置现有全部 FRAM 在此表示所有类型的非易失快速内存。	2K 字节
--	-------

由此，如下内存区域将预留用于 ...

应用程序中的变量，声明为 VAR_RETAIN	128 字节
固定为剩余定义标志 (%MB0...127)	128 字节

剩余的内存保留用于系统内部用途。

接口

15383
15272

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	接口不存在	接口不存在	接口不存在
	CANopen			
	SAE J1939			

标准波特率 = 250 Kbits/s

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

CR0452 彩色显示屏

15258

名称	数据
技术	TFT
屏幕对角线	4.3" (10.9 cm)
纵横比	16:9
分辨率	480 x 272 像素
颜色深度	8 位 = 256 种颜色，通过定义的调色板 ▶ 将图像创建为 256 色文件！
背景照明	LED 可调暗，以 1 % 为单位 设定 a) 可临时更改，且 b) 可预设（存储）

CR0452 操作元件

15260

显示屏配有以下操作元件：

- 4 个功能键 [F1]...[F4]
LED 背光照明
- 1 个摇臂开关
作为 4 个独立按键的组合
LED 背光照明
- 1 个 [OK] 键
LED 背光照明
- 1 个 [ESC] 键
LED 背光照明

所有按键均独立工作，彼此不受影响。
装置可检测多个同时按压的键并加以评估。

按键 LED 可调暗

8369

- ▶ 所有操作元件均有 LED 背光照明。

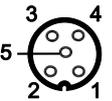
操作元件的**夜间设计**：

- 所有操作元件的 LED 仅可集体调暗：
 - ▶ 标志 KEY_BACKLIGHT_CTRL

外壳后面板上的连接

8351

M12 连接器，编码为 A，针对电源和 CAN：

图解	插脚	名称	注意
	1	n.c.	----
	2	VBB	8...32 V DC
	3	GND	端子 31
	4	CAN_H	
	5	CAN_L	

3.2.2 状态 LED

7998

集成状态 LED 指示工作状态（默认设定）。

LED 颜色	闪烁频率	说明
熄灭	永久熄灭	无工作电压
红色/绿色	短暂亮起	INIT 状态, 复位检查
绿色	5 Hz	未加载运行时系统
绿色	2 Hz	RUN 状态: 应用程序正在运行
绿色	永久亮起	STOP 状态: 应用程序停止
红色	5 Hz	含错误的 STOP 状态: 应用程序停止 原因: 欠电压
红色	10 Hz	含错误的 STOP 状态: 应用程序停止 原因: 超过应用程序或可视化的超时时间: ▶ 删除应用程序! ▶ 通电复位 ▶ 重新将应用程序加载至装置
红色	永久亮起	FATAL ERROR: 应用程序停止 原因: 软件电子狗故障 ▶ 通电复位 如果不成功: ▶ 转至 Bootloader ▶ 通电复位 ▶ 重新将 BasicSystem 加载至装置 ▶ 重新将应用程序加载至装置 如果不成功: ▶ 硬件错误: 将装置寄至 IFM!

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

15481

可通过 SET_LED 在应用程序中更改状态 LED 的频率和颜色。

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

3.3 接口说明

内容

CAN 接口	18
--------------	----

14098

3.3.1 CAN 接口

内容

CAN: 接口和协议	18
------------------	----

14101

连接和数据 → 技术资料

CAN: 接口和协议

15270
15271

装置仅配有一个 CAN 接口：

接口可搭配以下功能：

- RAW-CAN (Layer 2): 第 2 级上的 CAN → 章节 **功能元件：RAW-CAN (Layer 2)** (→ 页 [61](#))
- CANopen 主站/CANopen 从站 (→ 章节 **功能元件：CANopen** (→ 页 [98](#)))
- CANopen 网络变量 (通过 CODESYS) (→ 章节 **Network variables** (→ 页 [51](#)))
- SAE J1939 (针对驱动管理, → 章节 **功能元件：SAE J1939** (→ 页 [155](#)))
- 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

15272

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	接口不存在	接口不存在	接口不存在
	CANopen			
	SAE J1939			

标准波特率 = 250 Kbits/s

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

3.4 软件说明

内容	
装置的软件模块	19
CODESYS 项目的编程说明	22
工作状态	27
装置的性能极限	30

14107

3.4.1 装置的软件模块

内容	
Bootloader	20
运行时系统	20
应用程序	20
库	21

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 44)

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

15274

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。

可视化页面和嵌入式图片是 CODESYS 应用程序的一部分。

库

15409

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

库	使用
ifm_CR0452_Vxxyzz.LIB	装置特定库 必须始终包含于应用程序！
ifm_RawCAN_NT_Vxxyzz.LIB	(可选) 装置 CAN 接口搭配 CAN 第 2 层运行时
ifm_CANopen_NT_Vxxyzz.LIB	(可选) 装置 CAN 接口作为 CANopen 主站或 CANopen 从站运行时
ifm_J1939_NT_Vxxyzz.LIB	(可选) 装置 CAN 接口为了与发动机控制器通信时

Details: → chapter **针对装置 CR0452 的 IFM 库** (→ 页 [52](#))

3.4.2 CODESYS 项目的编程说明

内容

CODESYS 中的 FB、FUN、PRG	23
注意周期时间！	24
关于装置编程的重要说明	24
创建应用程序	25
使用 IFM 维护工具	26
应用程序的分发	26

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

15410

在 CODESYS 中，我们区分以下类型的功能元件：

FB = 功能块

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

FUN = 功能

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

PRG = 程序

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

ⓘ注意

不得在功能中调用功能块！

否则：执行期间，应用程序将崩溃。

不得递归或间接调用所有功能元件！

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

背景

功能的所有变数...

- 调用时初始化，且
- 返回至调用方后无效。

功能块有 2 种调用：

- 初始化调用以及
- 操作实际调用。

因此，这意味着功能中的功能块调用。

- 每次有一次额外的初始化调用，且
- 上次调用的数据丢失。

注意周期时间！

8006

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

通知

装置有运行过慢的风险！

周期时间不得太长！

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

关于装置编程的重要说明

20763

适用于以下装置：

- BasicController relay CR0431
- ▶ 编程时，连通接头 B:1 (VBB15) 和 B:8 (VBBs)。否则，不可编程。

背景

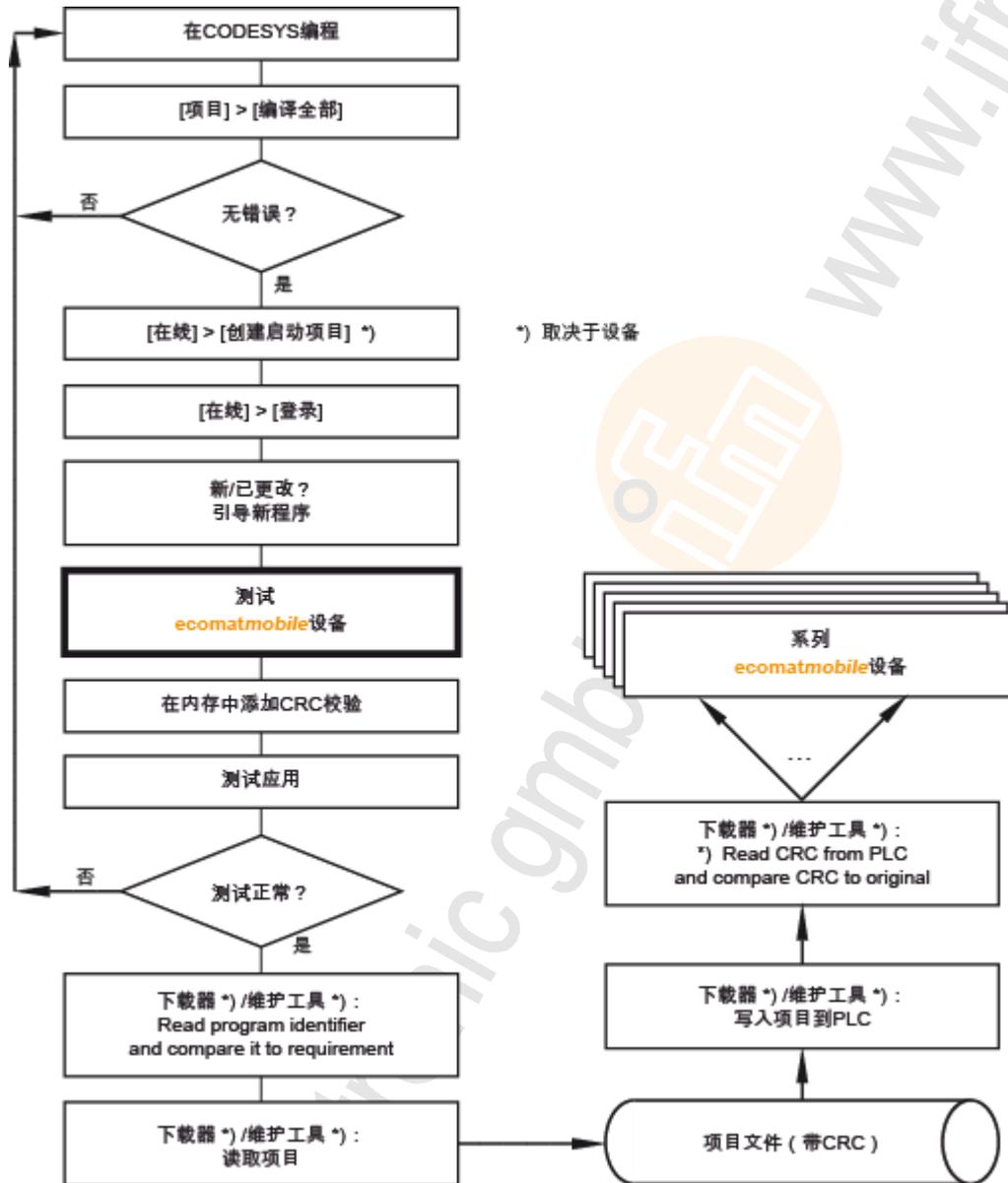
- 编程开始时，控制器复位所有输出端，还有 SUPPLY_SWITCH。
- 若无 VBB15，控制器将断开电源电压并关闭。
- 当控制器再次开启时，装置处于 Bootloader 模式。
程序员需再次将 BasicSystem 加载至装置。
而后，再次将应用程序加载至装置。

创建应用程序

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

使用 IFM 维护工具

8492

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

您可在此了解当前的 **IFM** 维护工具：

→ www.ifm.com > [移动车辆和工程机械系统]

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

应用程序的分发

8493

如果要应用程序软件复制到量产机器并使用，则建议采用以下顺序：

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

3.4.3 工作状态

1075

通电后，*ecomatmobile* 装置可处于五种可能的工作状态之一：

- BOOTLOADER
- INIT
- STOP
- RUN
- SYSTEM STOP

INIT 状态 (复位)

20647

前提：安装有效的运行时系统。

每次通电复位后均会经历该状态。

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

该状态可转换至以下状态之一：

- RUN
- STOP

STOP 状态

8288

在以下情况下可转换至该状态：

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

在 STOP 状态中：

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

该状态可转换至以下状态之一：

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

RUN 状态

8287

在以下情况下可转换至该状态：

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

在 RUN 状态中：

- > 运行时系统正在运行。
- > 应用程序正在运行。
- > LED 以 2 Hz 的频率闪烁绿色。
应用程序可对 LED 进行不同的控制 → FB **SET_LED** (→ 页 [220](#)).

该状态可转换至以下状态之一：

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

ERROR 状态

8290

在以下情况下可转换至该状态：

- 如果电源电压太低。

在 ERROR 状态中：

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

该状态可转换至以下状态之一：

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

FATAL ERROR 状态

8289

在以下情况下可转换至该状态：

- 内存错误（RAM/闪存）
- 异常错误
- 运行时系统错误

在 FATAL ERROR 状态中：

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

该状态可转换至以下状态之一：

- INIT（通电复位后）

3.4.4 装置的性能极限

7358



注意装置的性能极限！ → 技术资料

电子狗状况

15277

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（应用程序：100 ms；可视化：1 200 ms）：

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

排除故障：

- ▶ 删除应用程序！
- ▶ 通电复位
- ▶ 重新将应用程序加载至装置

如果相应的电子狗出现故障：

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

排除故障：

- ▶ 通电复位

如果不成功：

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

如果不成功：

- ▶ 硬件错误：将装置返回至 **IFM**！

可视化极限

8337

ecomatmobile 装置等使用的嵌入式显示屏不可提供所有颜色范围的位图图形，因为动力储备有限。但以下准备有助于在装置中使用位图图像：

- 正确选择图形
- 明智转换颜色，或明智编译调色板并
- 先正确缩放位图，再在装置使用。

→ **装置的性能极限** (→ 页 30)

8465

参数	CR0451 限制	CR0452 限制
文件类型	位图 (*.bmp)	
文件名称	仅小写字母，命名约定 = 8.3	
图像大小	320 x 240 像素	480 x 272 像素
颜色	8 位 = 28 种颜色 = 可代表 256 种颜色	
所需内存空间	≤ 76K 字节，取决于 RLE 压缩的图像内容	

表格：启动图像规格

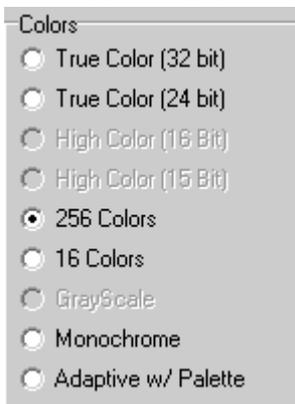
项目中使用的图像可大于给定的图像大小。但此时，仅显示图像的一部分（可选择）。

8464

颜色：

装置支持 2^8 (= 256) 种色差。

▶ 创建包含 256 种颜色的位图 (BMP)：



重新取样/缩放图像

3117

如果装置中加载的图像不满足尺寸或颜色要求，则会先调整尺寸再显示，并“核查”使用的颜色。

每次打开图像时，必须重新取样。这通常会导致要花费更长的时间从一个图像更换至其他图像。修正措施：

- ▶ 先在您的电脑上的图像处理程序中转换所有的位图或图像。
仅针对 BasicDisplay：图像通过 CODESYS 集成至项目时调整颜色调色板。在设备本身上，不会进行任何调整（尺寸、缩放、颜色）。
- ▶ 仅保存装置可视化中适当转换的图像。

可视化限制

8319

名称	限制
字符串长度	≤ 80 个字符
路径名称长度	不相关
可视化页面数量	≤ 15
单位可视化页面的图形对象数量	≤ 20
单位项目的位图数量	≤ 256 取决于应用程序的大小和可用闪存
单位项目的字符集数量	字符集永久存储且不可更改。
单位项目的 POU ¹⁾ 数量	≤ 512

1) POU (程序组织单元) = 功能、功能块或程序块

Because of the limited memory:

- 避免元件分组
- 避免主幻灯片可视化
- 避免主背景幻灯片可视化

CODESYS 可视化元件

453

位图图形 (BMP) → 章节 **可视化极限** (→ 页 [31](#))

! 并非所有 CODESYS 功能均可在本装置上成功执行：

可视化元件	PDM 功能安全
多段线	0 多段线最多包含 5 个锚定点；不可调整 多段线不会围住任何区域。
曲线	-- 不支持
长方形	+ 无已知问题
圆角长方形	-- 不支持
圆形、椭圆形	+ 无已知问题
多边形	0 多边形最多包含 10 个锚定点；不可调整
饼形图	-- 不支持
可视化	-- 不支持
按钮	-- 不支持
表格	-- 不支持
滚动条	-- 不支持
趋势曲线	-- 不支持
警报表格	-- 不支持
刻度	+ 创建刻度以作为 BMP 文件
指针式仪表	+ 作为包含叠加 CoDeSys 多边形的 BMP 文件表示指针式仪表
条形图	+ 创建刻度以作为 BMP 文件 作为叠加的 CoDeSys 长方形表示值
直方图	+ 创建刻度以作为 BMP 文件 作为叠加的 CoDeSys 长方形表示值
图形文件	+ 每个项目最多 256 • BMP ¹⁾ • BMP RLE 压缩 • TIFF ¹⁾ • JPEG ¹⁾
图形缩放模式	0 支持圆形、椭圆形、长方形、线条、多段线、多边形
ActiveX 元件	-- 不支持
指针图	-- 不支持
编辑工具	-- 不支持

+	可无忧使用
o	可有限使用
--	不可使用

1) 集成于项目时，文件转换成 RLE 压缩位图。

从 CODESYS 版本 2.3.9.24 起，集成图像时会打开额外的对话框。可利用该对话框调整文件，以适于针对装置使用的调色板。在颜色转换期间，您可在“最相似的颜色”（禁用 [Dithering]和“抖动”之间作出选择。

绘图区：

- 左上角标记虚拟和实际绘图区的原始位置 (0,0)。
- 虚拟绘图区 = 2,560 x 1,536 像素
(扩大实际绘图区)
- 不计算虚拟绘图区中的元件。

绘图区

15987

- 左上角标记虚拟和实际绘图区的原始位置 (0,0)。
- 虚拟绘图区 (X/Y 坐标) = -32768...+32767
(扩大实际绘图区)
- 包括外部尺寸在内的所有对象必须在虚拟绘图区边界之内，即便是在缩放或移位之后！
否则，可视化将不再正确。
- 不计算虚拟绘图区中的元件。

文本

8436

- 装置上明确可见的最小字体大小为 11 点。
 - 允许的字体：
 - Arial (标准)
 - Lucida Console
 - 允许的字体大小 [Pixel] 和字体粗细
 - Arial : 11 (标准)、16、24、32 (全部仅普通)
 - Lucida Console : 16、24、48*) (全部仅普通)
- *) 字体大小 48 的 Lucida Console 仅包含以下字符：
- 0 到 9 的数字
 - 特殊字符 + - . : %
 - 空格。
- 允许的效果：
 - 无 (标准)
 - 以下文本脚本忽略不计：
 - Western
 - Hebrew
 - Arabic
 - Greek
 - Turkish
 - Baltic
 - Central European
 - Cyrillic
 - Vietnamese

元件的移动

7392

图像和文本元件可通过定义的方式在显示屏上移动。

元件移动	说明
旋转	围绕定义的轴心点旋转元件 显示旋转角度 <ul style="list-style-type: none"> ▪ 旋转角度以 [degree] 为单位 ▪ 正值 = 顺时针旋转 ▪ 负值 = 逆时针旋转
移动	元件移动： <ul style="list-style-type: none"> ▪ 水平 ▪ 垂直 ▪ 仅在绘图区内 ▪ 最大直至离开绘图区
相对移动： <ul style="list-style-type: none"> • 长方形 • 椭圆形/圆形 	元件的每个边缘均可通过 INT 类型变量移动特定的像素点。 <ul style="list-style-type: none"> ▪ 4 个边缘的基本位置 = 0 ▪ 新值会让该边缘移动特定的值 值 > 0 的移动方向： <ul style="list-style-type: none"> ▪ 水平边缘向下 ▪ 垂直边缘向右 值 < 0 的移动方向： <ul style="list-style-type: none"> ▪ 水平边缘向上 ▪ 垂直边缘向左

本装置 CAN 的限制

17975

i FIFO (**先入先出**) = 堆栈内存的工作原理：先写入堆栈内存的数据包也将先读取。每个标识符均有这样的一个缓冲区 (队列)。

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

- CAN_TX..., → 功能元件：传送 RAW-CAN 数据
- **CAN_RX_ENH_FIFO** (→ 页 [75](#))
- **CAN_RX_RANGE_FIFO** (→ 页 [80](#))

FiFo 消息的数量是有限的。以下装置限制有效：

装置	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n, CR120n
标准		
最多 FiFo 传送 - 通过 FB CAN_TX... - 通过 FB CAN_TX_ENH...	4 条消息 16 条消息	4 条消息 16 条消息
最多 FiFo 接收 - 通过 FB CAN_RX_..._FIFO	32 条消息	32 条消息

本装置 CANopen 的限制

17976

以下装置限制有效：

装置	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n, CR120n
标准		
最多保护错误	32 条消息	128 条消息
最多 SDO 数据	2 048 字节	2 048 字节

本装置 CAN J1939 的限制

17977

以下装置限制有效：

标准	装置	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n, CR120n
最多 FiFo 传送 - 通过 FB J1939_TX - 通过 FB J1939_TX_ENH		4 条消息 16 条消息	4 条消息 16 条消息
最多 FiFo 接收 - 通过 FB J1939_RX_FIFO		32 条消息	32 条消息
最多 DTC		64 条消息	64 条消息
最多数据 J1939		1 785 字节	1 785 字节

4 配置

内容

设定运行时系统	39
设定编程系统	43
一般功能配置	49
变量	50

1016

相应安装说明或本文档 **附录** (→ 页 [239](#))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	40
更新运行时系统	42
检验安装	42

14091

4.1.1 重新安装运行时系统

14635
8486

交付 **ecomatmobile** 控制器后，正常情况下无需加载运行时系统（LED 以 5 Hz 的频率闪烁绿色）。在该工作模式中，仅 Bootloader 启用。它提供加载操作系统（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 > [移动车辆和工程机械系统]

通知

数据有丢失的风险！

如果数据传送期间断电，则数据可能丢失，装置无法再正常运行。仅 **IFM** 可进行维修。

▶ 确保在数据传送时电源不间断！

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_CR0452_Vxxyyzz.RESX) ,
- PLC 配置 (ifm_CR0452_Vxx.CFG) ,
- 装置库 (ifm_CR0452_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 "CR0452"）和软件版本号 "xx"（例如 "01"）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)。

① 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“维护工具”传输至装置。（下载器位于 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 > [移动车辆和工程机械系统].

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“维护工具”记载。



4.1.2 更新运行时系统

13269

装置上已安装较旧版本的运行时系统。现在，您是否想要更新装置上的运行时系统？

14158

通知

数据有丢失的风险！

删除或升级运行时系统时，装置上的所有数据和程序将被删除。

▶ 先保存所有所需数据和程序，再删除或升级运行时系统！

就该操作而言，可遵循跟之前章节“重新安装运行时系统”相同的说明。

4.1.3 检验安装

14637

- ▶ 将运行时系统加载至控制器之后：
 - 检查运行时系统是否正确传送！
 - 检查是否将正确的运行时系统加载至控制器！
- ▶ 第 1 次测试：

利用 **IFM** 维护工具测试是否加载正确的运行时系统版本：

 - 在装置中读取运行时系统的名称和版本！
 - 手动对比该信息和目标数据！
- ▶ 第 2 次测试（可选）：

在应用程序中检查是否加载正确的运行时系统版本：

 - 在装置中读取运行时系统的名称和版本！
 - 对比该数据和特定值！

以下 FB 用于读取数据：

GET_SW_INFO (→ 页 [208](#))

传送关于装置系统软件的信息：

- 软件名称，
- 软件版本，
- 版本号，
- 版本日期

- ▶ 如果应用程序检测到运行时系统的版本不正确：

将所有安全功能调至安全状态。

4.2 设定编程系统

内容

手动设定编程系统.....	43
通过模板设定编程系统.....	48

14461

4.2.1 手动设定编程系统

内容

设定目标	44
启用 PLC 配置.....	45
CAN 声明 (例如 CR1080)	46

3963

设定目标

13136
11379

在 CODESYS 中创建新项目时，必须加载与装置相符的目标文件。

- ▶ 在菜单 [Configuration] 对话框 [Target Settings] 中选择所需目标文件。
- > 目标文件构成编程系统硬件的接口。
- > 同时，选择目标时加载多个重要的库和 PLC 配置。
- ▶ 如有必要，进入窗口 [Target settings] > 选项卡 [Network functionality] > 启用 [Support parameter manager] 并/或启用 [Support network variables]。
- ▶ 如有必要，移除加载的 (3S) 库或通过进一步的 (IFM) 库加以补充。
- ▶ 始终补充相应的装置库 ifm_CR0452_Vxxyzz.LIB 并手动操作！

ⓘ 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_CR0452_Vxxyzz.RESX) ，
- PLC 配置 (ifm_CR0452_Vxx.CFG) ，
- 装置库 (ifm_CR0452_Vxxyzz.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 "CR0452"）和软件版本号 "xx"（例如 "01"）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。

启用 PLC 配置

10079

PLC 配置通过目标系统自动加载。PLC 配置映射 CODESYS 中文件 CR0452.cfg 的内容。程序员可这样轻松访问预定义的系统 and 错误标志，输入端和输出端，以及装置的 CAN 接口。

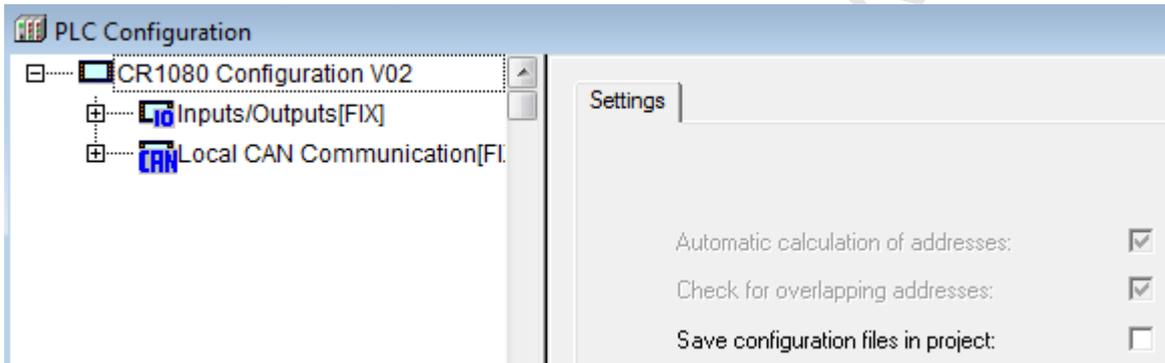
访问 PLC 配置（如 CR1080）：

- ▶ 单击 CoDeSys 中的选项卡 [Resources]：



- ▶ 双击左列的 [PLC Configuration]。

> 当前 PLC 配置显示屏（→ 下图）：



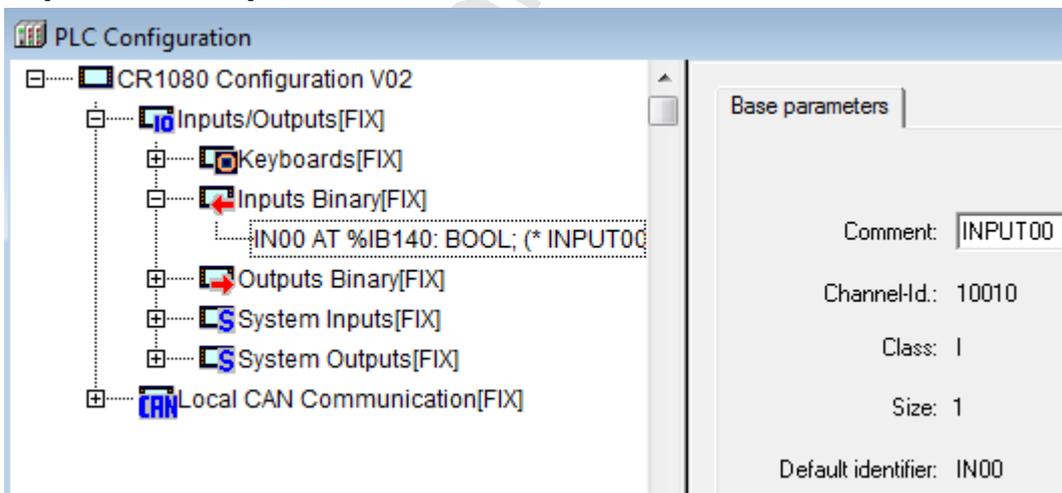
> 根据配置，以下适用于针对用户的程序环境：

- 系统和错误标志

根据应用程序和程序，这些标志必须经过处理和评估。可通过符号名称访问。

- 输入端和输出端的结构

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

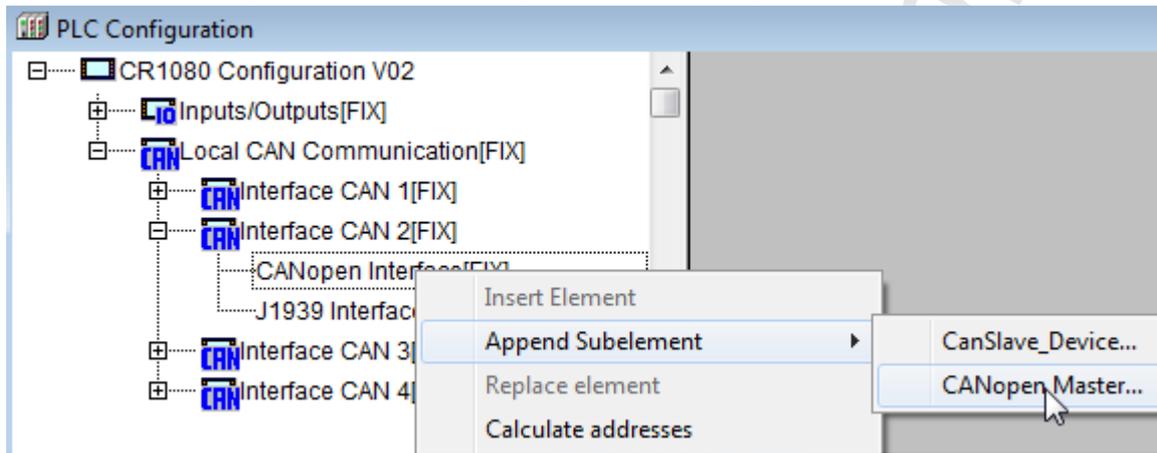


CAN 声明 (例如 CR1080)

10080

在 CODESYS PLC 配置中，您现在需要声明 CAN 接口。

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

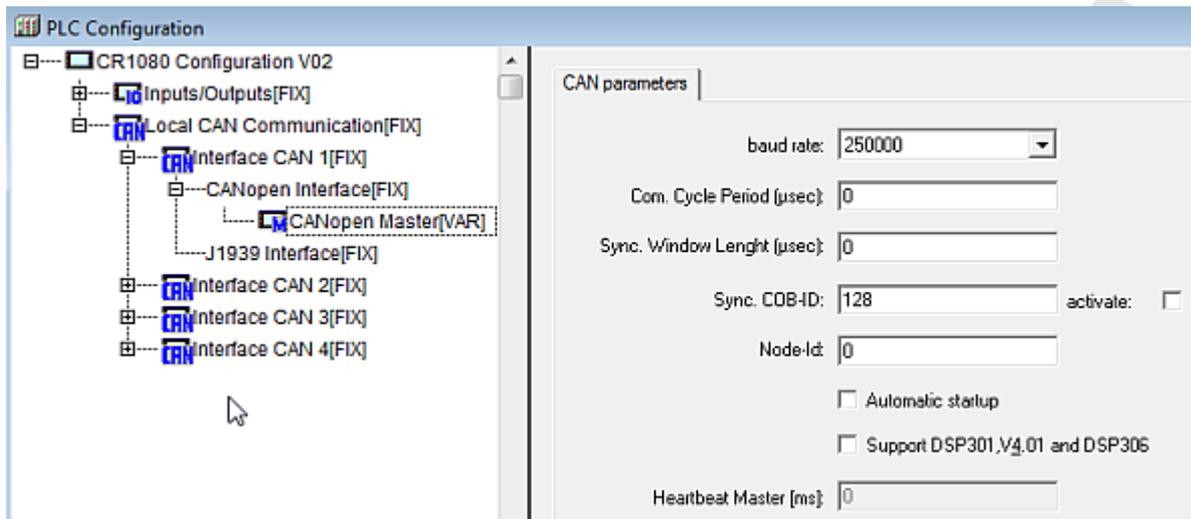


信息

如果装置作为从站运行，则亦可选择 [CanSlave_Device]。

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

- > PLC 配置的 CAN 参数显示。有些 CAN 参数已默认设定：



- ▶ 如果装置在 CAN 第 2 层运行，或通过网络变量或 CAN_RX / CAN_TX 作为从站运行：
 - ❗ 检查是否针对装置设定正确的波特率（所有参与者的波特率必须相同）。
- ▶ 如果装置作为 CANopen 主站运行：
 - 检查所有参数设定。
- ▶ 关闭窗口 [PLC Configuration]。
- ▶ 在菜单 [File] > [Save as...] 中 为项目指定合理的名称并将其保存在所需目录中。
- ▶ ❗ 在应用程序中始终针对 CAN 接口 **CANOPEN_ENABLE** (→ 页 99) 调用 FB 自身的实例！

4.2.2 通过模板设定编程系统

13745

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

970

-  安装 **ecomatmobile** DVD“软件、工具和文档”时，包含模板的项目已存储于您的 PC 的程序目录：
- ...\\ifm electronic\\CoDeSys V...\\Projects\\Template_DVD_V...
- ▶ 通过以下路径在 CODESYS 中打开所需模板：
[File] > [New from template...]
 - > CODESYS 创建新项目并显示基本程序结构。强烈建议遵循所示的步骤。

4.3 一般功能配置

3971

4.3.1 系统变量

15576

所有系统变量 (→ 章节 **系统标志** (→ 页 [239](#))) 均已定义地址且不可更改。

4.4 变量

内容

保留变量	50
网络变量	51

3130

在本章，您将了解更多关于如何处理变量的信息。

4.4.1 保留变量

8672

保留变量可自动保存在受保护的内存区域，重启期间可自动再次加载。

14166

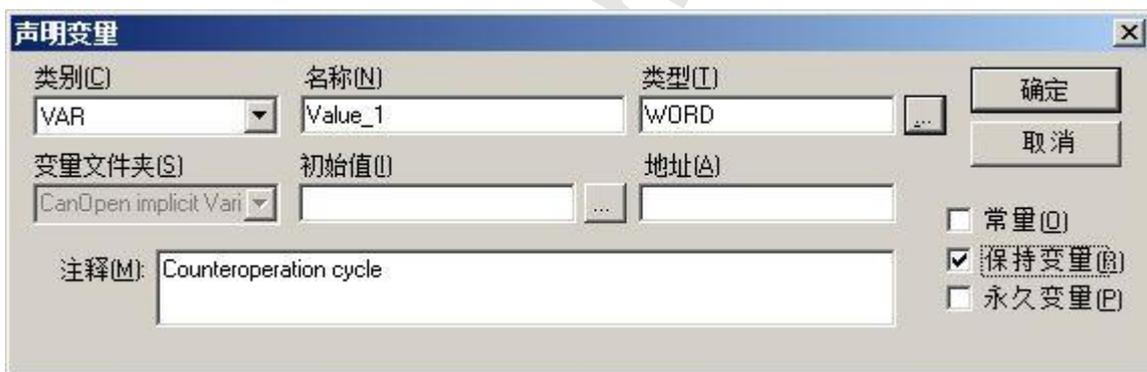
保持变量的一般应用如下：

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

即装置关闭时其值不得丢失的所有变量。

所有变量类型，还有复杂结构（如计时器），均可声明为保持变量。

► 为此，可启用变量声明中的控制字段 [RETAIN]（→ 窗口）。



保存保留变量

9853

在装置中，数据类型 RETAIN 仅在运行时期间储存在易失内存 (RAM) 中。为永久保存数据，会在每个周期结束时自动保存于 FRAM 内存¹⁾。

¹⁾ FRAM 在此表示所有类型的非易失快速内存。

ⓘ 注意

在本装置中，切勿使用 3S 库 SysLibPlcCtr1.lib 中的以下功能：

- FUN SysSaveRetains
- FUN SysRestoreRetains

读回保留变量

9854

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

ⓘ 注意

在本装置中，切勿使用 3S 库 SysLibPlcCtr1.lib 中的以下功能：

- FUN SysSaveRetains
- FUN SysRestoreRetains

4.4.2 网络变量

15242
9856

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

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

5 IFM 功能元件

内容

针对装置 CR0452 的 IFM 库	52
针对装置 CR0452 的 IFM 功能元件	59

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 CR0452 的 IFM 库

内容

所需的库	52
库 ifm_CR0452_Vxxyzz.LIB	53
库 ifm_CR0452_Init_Vxxyzz.LIB	54
库 ifm_PDMsmart_util_Vxxyzz.LIB	54
库 ifm_RAWCan_NT_Vxxyzz.LIB	54
库 ifm_CANopen_NT_Vxxyzz.LIB	55
库 ifm_J1939_NT_Vxxyzz.LIB	57

14235

5.1.1 所需的库

15300

如果您不想将 IFM 模板作为本装置初始编程的基础，则应确保至少将以下的库集成至您的项目：

ifm_CR0452_Vxxyzz.LIB	装置库
ifm_CR0452_Init_Vxxyzz.LIB	初始化装置屏幕

网络变量所需的库

15304

如果您想要使用网络变量，则还需要以下的库：

ifm_NetVarLib_NT_Vxxyzz.LIB	网络变量 (IFM 库) 支持
3S_CANopenNetVar.LIB	网络变量 (3S 库) 支持

5.1.2 库 ifm_CR0452_Vxxyzz.LIB

15284

此为装置库。该 IFM 库包含以下功能块：

功能元件	简短说明
FLASH_INFO (→ 页 199)	从用户闪存读取信息： <ul style="list-style-type: none"> • 内存区域的名称（用户定义）， • 软件版本， • 起始地址（用于使 IEC 结构简单读取）
FLASH_READ (→ 页 201)	直接将不同数据类型从闪存传输到 RAM
GET_APP_INFO (→ 页 203)	传送关于存储在装置上的应用程序的信息： <ul style="list-style-type: none"> • 应用程序名称， • 应用程序版本， • 唯一的 CODESYS 版本号， • CODESYS 版本日期
GET_HW_INFO (→ 页 205)	传送关于装置硬件的信息： <ul style="list-style-type: none"> • IFM 产品编号（例如 CR0403）， • 产品名称， • 明确的序列号， • 硬件版本， • 生产日期
GET_IDENTITY (→ 页 207)	读取存储在装置上的应用程序的 ID (之前已通过以下方式保存： SET_IDENTITY (→ 页 219))
GET_SW_INFO (→ 页 208)	传送关于装置系统软件的信息： <ul style="list-style-type: none"> • 软件名称， • 软件版本， • 版本号， • 版本日期
GET_SW_VERSION (→ 页 210)	传送关于存储在装置上的软件版本的信息： <ul style="list-style-type: none"> • BasicSystem 版本， • Bootloader 版本， • SIS 版本， • 应用程序版本， • 用户闪存版本
MEM_ERROR (→ 页 212)	利用信号通知某些参数或内存中的错误 (再次)初始化系统资源
MEMCPY (→ 页 214)	直接写入和读取内存中的不同数据类型
OHC (→ 页 217)	可调工作时间计数器 (0..3)
SET_IDENTITY (→ 页 219)	设定应用程序特定程序 ID
SET_LED (→ 页 220)	在应用程序中更改状态 LED 的频率和颜色
SET_PASSWORD (→ 页 222)	设定程序和内存上传访问控制的用户密码

功能元件	简短说明
TIMER_READ_US (→ 页 224)	读取当前系统时间，以 [μs] 为单位 最大值 = 1h 11min 34s 967ms 295μs

5.1.3 库 ifm_CR0452_Init_Vxxyzz.LIB

15286

该 **IFM** 库包含以下功能块：

功能元件	简短说明
BASICDISPLAY_INIT (→ 页 234)	在第一个 PLC 周期中初始化 BasicDisplay 的屏幕

5.1.4 库 ifm_PDMsmart_util_Vxxyzz.LIB

15289

该 **IFM** 库包含以下功能块：

功能元件	简短说明
GET_TEXT_FROM_FLASH (→ 页 226)	通过以下功能块从闪存读取类型 STRING 的文本： FLASH_READ (→ 页 201)
NORM_DINT (→ 页 228)	将定义的限值范围内的值 [DINT] 正常化为新的限值范围内的值
NORM_REAL (→ 页 230)	将定义的限值范围内的值 [REAL] 正常化为新的限值范围内的值
PDM_PAGECONTROL (→ 页 235)	控制特定可视化页面的显示
TOGGLE (→ 页 232)	仅通过一个输入位设定和复位布尔变量。

5.1.5 库 ifm_RAWCan_NT_Vxxyzz.LIB

14715

该 **IFM** 库包含以下功能块：

功能元件	简短说明
CAN_ENABLE (→ 页 62)	初始化所示 CAN 接口 配置 CAN 波特率
CAN_RECOVER (→ 页 64)	启用/禁用自动总线关闭处理 总线关闭时重新启动 CAN 接口
CAN_REMOTE_REQUEST (→ 页 93)	发送相应的请求并将其他装置的响应作为结果返回
CAN_REMOTE_RESPONSE (→ 页 95)	将数据提供给装置中的 CAN 控制器，该数据作为对远程消息的请求的响应自动发送。
CAN_RX (→ 页 70)	配置数据接收对象并读取数据对象的接收缓冲区

功能元件	简短说明
CAN_RX_ENH (→ 页 72)	<ul style="list-style-type: none"> 配置数据接收对象并读取数据对象的接收缓冲区 帧类型和掩码可选
CAN_RX_ENH_FIFO (→ 页 75)	<ul style="list-style-type: none"> 配置数据接收对象并读取数据对象的接收缓冲区 帧类型和掩码可选 每个周期会有几条 CAN 消息
CAN_RX_RANGE (→ 页 78)	<ul style="list-style-type: none"> 配置一系列数据接收对象并读取数据对象的接收缓冲区 帧类型和掩码可选
CAN_RX_RANGE_FIFO (→ 页 80)	<ul style="list-style-type: none"> 配置一系列数据接收对象并读取数据对象的接收缓冲区 帧类型和掩码可选 每个周期会有几条 CAN 消息
CAN_SETDOWNLOADID (→ 页 65)	= 设定 CAN 下载 ID 设定 CAN 接口的下载 ID
CAN_STATUS (→ 页 67)	获取关于所选 CAN 总线的状态信息： BAUDRATE、DOWNLOAD_ID、BUSOFF、WARNING_RX、WARNING_TX、VERSION、BUSLOAD 并根据需要复位：BUSOFF、WARNING_RX、WARNING_TX
CAN_TX (→ 页 84)	将 CAN 数据对象（消息）传输至配置的 CAN 接口以在每次调用时传送
CAN_TX_ENH (→ 页 86)	将 CAN 数据对象（消息）传输至配置的 CAN 接口以在每次调用时传送 CAN 特定特性可设定
CAN_TX_ENH_CYCLIC (→ 页 89)	以循环的方式将 CAN 数据对象（消息）传输至配置的 CAN 接口以便传送 CAN 特定特性可设定

5.1.6 库 ifm_CANopen_NT_Vxxyzz.LIB

14914

该 IFM 库包含以下功能块：

功能元件	简短说明
CANOPEN_ENABLE (→ 页 99)	初始化所示 CANopen 主站接口 配置 CAN 波特率
CANOPEN_GETBUFFERFLAGS (→ 页 101)	= CANopen 获取缓冲标志 提供关于缓冲标志的信息 标志可通过可选输入端复位。
CANOPEN_GETEMCYMESSAGES (→ 页 148)	= 获取 CANopen 紧急报文 列出自上次删除消息以来控制器从网络其他节点接收的所有紧急报文。 该列表可通过设置相应的输入端复位。
CANOPEN_GETERRORREGISTER (→ 页 150)	= 获取 CANopen 错误寄存器 从控制器读取错误寄存器 0x1001 和 0x1003 寄存器可通过设置相应的输入端复位。

功能元件	简短说明
CANOPEN_GETGUARDHBERRLIST (→ 页 143)	= 获取 CANopen 保护和检测信号错误列表 批量列出主站检测到错误的所有节点： 保护错误，检测信号错误 该列表可通过设定相应的输入端复位。
CANOPEN_GETGUARDHBSTATSLV (→ 页 145)	= CANopen 从站获取保护和检测信号状态 将以下状态的信号发送至从属运行的控制器： 节点保护监控，检测信号监控 通过信号发送的错误可通过设定相应的输入端复位。
CANOPEN_GETNMTSTATESLAVE (→ 页 110)	= CANopen 从站获取网络管理状态 发送节点网络运行状态的信号
CANOPEN_GETODCHANGEDFLAG (→ 页 116)	= 获取对象目录更改标志 报告特定对象目录条目的任何值更改
CANOPEN_GETSTATE (→ 页 103)	= CANopen 设定状态 请求主站、从属装置或网络中特定节点参数
CANOPEN_GETSYNCSTATE (→ 页 138)	= CANopen 获取 SYNC 状态 • 读取 SYNC 功能的设定 (启用/未启用) • 读取 SYNC 功能的错误状态 (SyncError)
CANOPEN_NMTSERVICES (→ 页 112)	= CANopen 网络管理服务 更新内部节点状态 且视 NMT 命令条目而定： • 触发 NMT 命令或 • 触发节点初始化
CANOPEN_READOBJECTDICT (→ 页 118)	= CANopen 读取对象目录 从装置的对象目录读取配置数据
CANOPEN_SDOREAD (→ 页 123)	= CANopen 读取 SDO 读取 "Expedited SDO" = 加速服务数据对象
CANOPEN_SDOREADBLOCK (→ 页 125)	= CANopen 读取 SDO 块 通过 SDO 块传输读取网络节点对象目录中所示的条目
CANOPEN_SDOREADMULTI (→ 页 128)	= CANopen 读取 SDO multi 读取网络节点对象目录中所示的条目
CANOPEN_SDOWRITE (→ 页 130)	= SDO 写入 写入 "Expedited SDO" = 加速服务数据对象
CANOPEN_SDOWRITEBLOCK (→ 页 132)	= CANopen 写入 SDO block 通过 SDO 块传输写入网络节点对象目录中所示的条目
CANOPEN_SDOWRITEMULTI (→ 页 135)	= CANopen 写入 SDO multi 写入网络节点对象目录中所示的条目
CANOPEN_SENDEMCMYMESSAGE (→ 页 152)	= CANopen 发送紧急报文 发送 EMCY 报文 报文通过相应的参数集合而成并输入寄存器 0x1003

功能元件	简短说明
CANOPEN_SETSTATE (→ 页 106)	= CANopen 设定状态 设定主站、从属装置或网络中特定节点的参数
CANOPEN_SETSYNCSTATE (→ 页 140)	= CANopen 设定 SYNC 状态 开启和关闭 SYNC 功能
CANOPEN_WRITEOBJECTDICT (→ 页 120)	= CANopen 读取对象目录 将配置数据写入装置的对象目录

5.1.7 库 ifm_J1939_NT_Vxxyyzz.LIB

14912

该 **IFM** 库包含以下功能块：

功能元件	简短说明
J1939_DM1RX (→ 页 189)	J1939 诊断消息 1 RX 从其他 ECU 接收诊断消息 DM1 或 DM2
J1939_DM1TX (→ 页 192)	J1939 诊断消息 1 TX 将有效错误消息传送到 CAN 堆栈
J1939_DM1TX_CFG (→ 页 195)	J1939 诊断消息 1 TX 可配置 CAN 堆栈不发送循环 DM1 "zero active faults" 消息
J1939_DM3TX (→ 页 197)	J1939 诊断消息 3 TX 删除装置上的无效 DTC (DM2)
J1939_ENABLE (→ 页 156)	初始化 J1939 堆栈
J1939_GETDABYNAME (→ 页 158)	= 获取目标任意名称 通过名称信息确定一个或多个参与者的目标地址
J1939_NAME (→ 页 161)	向装置提供名称以在网络中识别
J1939_RX (→ 页 173)	接受单帧消息 显示 CAN 总线最后读取的消息
J1939_RX_FIFO (→ 页 175)	= 包含 FIFO 的 J1939 RX 接受所有特定消息并接着从 FiFo 读取
J1939_RX_MULTI (→ 页 177)	= J1939 RX 多帧消息 接收多帧消息
J1939_SPEC_REQ (→ 页 168)	= J1939 特定请求 请求和接收来自其他控制器的特定消息
J1939_SPEC_REQ_MULTI (→ 页 170)	= J1939 特定请求多帧消息 请求和接收来自其他控制器的特定多帧消息
J1939_STATUS (→ 页 165)	显示 J1939 堆栈的相关信息
J1939_TX (→ 页 180)	发送单个单帧消息

功能元件	简短说明
J1939_TX_ENH (→ 页 182)	= J1939 TX 增强 发送单个单帧消息 还可设定：传输优先级、数据长度
J1939_TX_ENH_CYCLIC (→ 页 184)	= J1939 TX 增强循环 循环发送单帧消息 还可设定：传输优先级、数据长度、周期
J1939_TX_ENH_MULTI (→ 页 186)	= J1939 TX 增强多帧消息 发送单个多帧消息

5.2 针对装置 CR0452 的 IFM 功能元件

内容	
输出端功能元件	60
功能元件：RAW-CAN (第 2 层)	61
功能元件：CANopen	98
功能元件：SAE J1939.....	155
功能元件：系统.....	198
功能元件：图形.....	225

13988
3826

您可在此了解适用于本装置的 **IFM** 功能元件（按主题分类）。

5.2.1 输出端功能元件

8354
7556

有些功能元件返回 RESULT 消息。

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1...31		全局返回值； 示例：
1	01	FB 执行完成且无错误 - 数据有效
4	04	FB 正在处理中 - 数据循环处理
5	05	FB 正在处理中 - 仍在接收
6	06	FB 正在处理中 - 仍在发送
7	07	FB 正在处理中 - ID 远程启用中
8	08	功能块启用中
14	0E	FB 已启用 CANopen 管理器配置装置并发送 SDO
15	0F	FB 已启用 CANopen 管理器启动
32 ₁₀ ...63		FB 特定返回值
64 ₁₀ ...127		FB 特定错误消息
128 ₁₀ ...255		全局错误消息； 示例：
238	EE	错误： CANopen 配置太大，无法启动
239	EF	错误： CANopen 管理器无法启动
240	F0	错误： 多个模式输入端已启用 例如 CANopen NTM 服务
241	F1	错误： CANopen 状态转换未获准许
242	F2	错误： 不可设定
247	F7	错误： 超过了内存（长度大于数组）
250	FA	错误： FiFo 已满 - 数据丢失
252	FC	错误： CAN 多帧传送失败
253	FD	错误： CAN 传送失败。 数据无法传送。
255	FF	错误： 内存不足以使用多帧

5.2.2 功能元件：RAW-CAN (第 2 层)

内容

功能元件：RAW-CAN 状态.....	61
功能元件：接收 RAW-CAN 数据.....	69
功能元件：传送 RAW-CAN 数据.....	83
功能元件：RAW-CAN 远程.....	92

15051

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

功能元件：RAW-CAN 状态

内容

CAN_ENABLE.....	62
CAN_RECOVER.....	64
CAN_SETDOWNLOADID.....	65
CAN_STATUS.....	67

15049

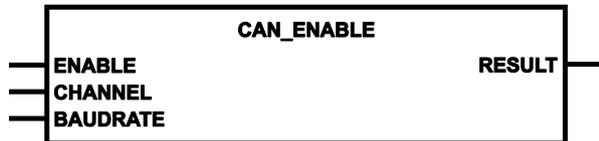
CAN_ENABLE

7492

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7494

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

若要更改波特率，则需遵循以下步骤：

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

输入端参数

7495

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

输出端参数

8530

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

RESULT 的可能结果：

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

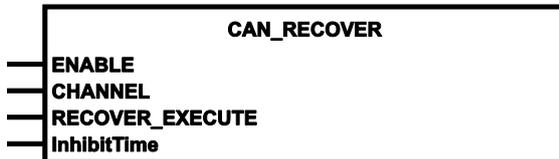
CAN_RECOVER

7512

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7513

CAN_RECOVER has the following tasks:

- 启用/禁用自动总线关闭处理
- 总线关闭时重新启动 CAN 接口
- > 如果总线关闭：CAN 控制器删除所有缓冲区（包括其他协议的缓冲区）

如果 CAN_RECOVER 未使用 (ENABLE=FALSE)：

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

输入端参数

7514

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

CAN_SETDOWNLOADID

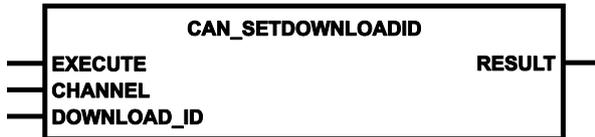
7516

= 设定下载 ID

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7517

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

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

输入端参数

7519

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

输出端参数

7520

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

RESULT 的可能结果：

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

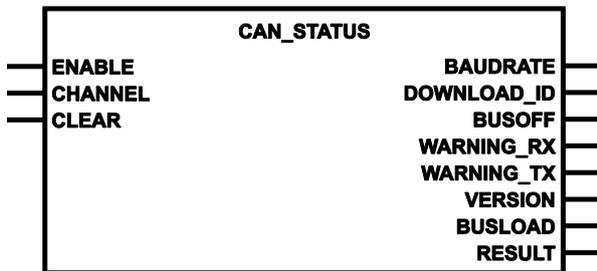
CAN_STATUS

7499

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7501

CAN_STATUS provides information on the chosen CAN bus.

若无硬件初始化，则可将以下标志复位为 FALSE：

- BUSOFF
- WARNING_RX
- WARNING_TX

输入端参数

7502

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1..n), 视装置而定
CLEAR	BOOL := FALSE	TRUE: 复位以下标志： • WARNING_RX • WARNING_TX • BUSOFF FALSE: 功能元件未执行

输出端参数

7504

参数	数据类型	说明
BAUDRATE	WORD	CANopen 节点的当前波特率, 以 [kBaud] 为单位
DOWNLOAD_ID	BYTE	当前下载 ID
BUSOFF	BOOL	接口错误 CAN BUS OFF
WARNING_RX	BOOL	接口超过接收警告阈值
WARNING_TX	BOOL	接口超过传送警告阈值
VERSION	DWORD	IFM CAN 堆栈库的 vabvab
BUSLOAD	BYTE	当前总线负载, 按 [%] 计算
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

功能元件：接收 RAW-CAN 数据

内容	
CAN_RX	70
CAN_RX_ENH	72
CAN_RX_ENH_FIFO	75
CAN_RX_RANGE	78
CAN_RX_RANGE_FIFO	80

15050

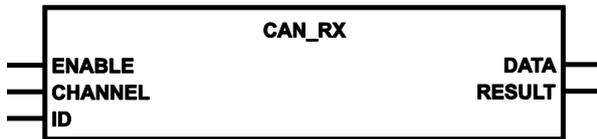
CAN_RX

7586

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7588

CAN_RX 用于接收消息。

FB 仅限几个功能，所需内存空间小。

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

输入端参数

7589

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
ID	DWORD	数据对象标识符编号： 正常帧 (2 048 个 ID)： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (536 868 864 个 ID)： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF

输出端参数

7590

参数	数据类型	说明
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果 :

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误 : 不可设定

CAN_RX_ENH

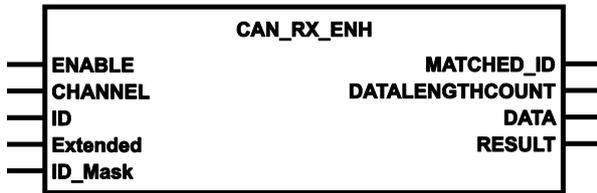
= CAN RX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxyyzz.LIB

7606

CODESYS 中的符号:



说明

7608

此外, CAN_RX_ENH 提供以下可能性 (完全不同于 CAN_RX (→ 页 70)):

- 选择帧类型 (11 或 29 位),
- 定义掩码以评估 CAN ID。

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

可通过掩码将几个标识符定义为过滤器。

例如:

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

输入端参数

7609

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID	DWORD	数据对象 ID 编号: 正常帧 (2 ¹¹ 个 ID): 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹ 个 ID): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
ID_Mask (参数可选)	DWORD := 0	标识符过滤器掩码 如果 ID_MASK 位 = 0, CAN ID 位可能 = 0 或 1 如果 ID_MASK 位 = 1, CAN ID 位必须 = ID 位

输出端参数

7613

参数	数据类型	说明
MATCHED_ID	DWORD	数据对象标识符编号
DATALengthCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1..8 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定

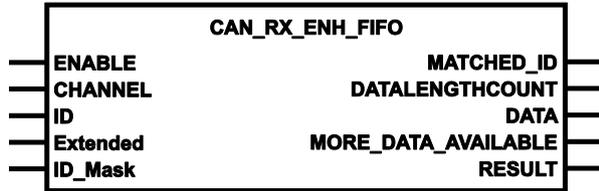
CAN_RX_ENH_FIFO

= 包含 FiFo 的 CAN RX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

此外，CAN_RX_ENH_FIFO 提供针对已接收数据的 FiFo (完全不同于 **CAN_RX_ENH** (→ 页 72)). 因此，在一个周期中可接收多个 CAN 消息。

❗ FiFo 已满时不会出现覆盖。 进站消息将丢失。

在此事件中：

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

过滤器掩码说明： → **CAN_RX_ENH** (→ 页 72) > 章节 **说明**

输入端参数

7609

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID	DWORD	数据对象 ID 编号: 正常帧 (2 ¹¹ 个 ID): 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹ 个 ID): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
ID_Mask (参数可选)	DWORD := 0	标识符过滤器掩码 如果 ID_MASK 位 = 0, CAN ID 位可能 = 0 或 1 如果 ID_MASK 位 = 1, CAN ID 位必须 = ID 位

输出端参数

7617

参数	数据类型	说明
MATCHED_ID	DWORD	数据对象标识符编号
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1..8 字节)
MORE_DATA_AVAILABLE	BOOL	TRUE: FiFo 中有进一步接收的数据 FALSE: FiFo 中无进一步接收的数据
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定
250	FA	错误：FiFo 已满 - 数据丢失

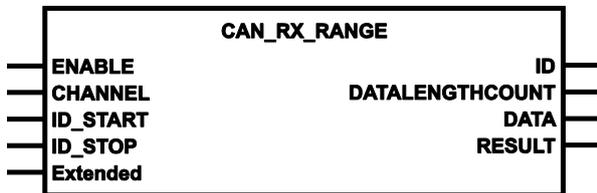
CAN_RX_RANGE

7592

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7594

CAN_RX_RANGE provides the following settings:

- 选择消息类型 (11 或 29 位) ,
- 定义标识符范围。

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

输入端参数

7595

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID_START	DWORD	数据对象标识符范围的起始编号 : 正常帧 (2 ¹¹) : 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹) : 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	数据对象标识符范围的结束编号 : 正常帧 (2 ¹¹) : 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹) : 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)

输出端参数

7598

参数	数据类型	说明
ID	DWORD	数据对象标识符编号： 正常帧（2 048 个 ID）： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧（536 868 864 个 ID）： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据，（1...8 字节）
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定

CAN_RX_RANGE_FIFO

7601

= 包含 FiFo 的 CAN RX 范围

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7603

CAN_RX_RANGE_FIFO basically works like **CAN_RX_RANGE** (→ 页 78).

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

❗ FiFo 已满时不会出现覆盖。 进站消息将丢失。

在此事件中：

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

输入端参数

7595

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID_START	DWORD	数据对象标识符范围的起始编号: 正常帧 (2 ¹¹): 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	数据对象标识符范围的结束编号: 正常帧 (2 ¹¹): 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)

输出端参数

7604

参数	数据类型	说明
ID	DWORD	数据对象标识符编号： 正常帧（2 048 个 ID）： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧（536 868 864 个 ID）： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据，（1...8 字节）
MORE_DATA_AVAILABLE	BOOL	TRUE: FiFo 中有进一步接收的数据 FALSE: FiFo 中无进一步接收的数据
RESULT	BYTE	功能块反馈 （可能的消息 → 下表）

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定
250	FA	错误：FiFo 已满 - 数据丢失

功能元件： 传送 RAW-CAN 数据

内容	
CAN_TX	84
CAN_TX_ENH.....	86
CAN_TX_ENH_CYCLIC	89

15055

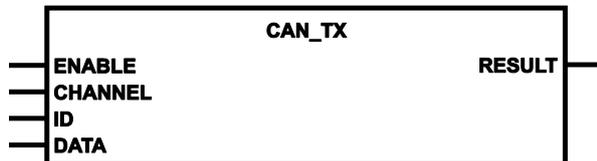
CAN_TX

7522

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7523

CAN_TX 会每周期发送一条标准消息。

FB 仅限几个功能，所需内存空间小。

> 如果在一个周期内多次调用该 FB 的实例，数据亦会发送多次。

如果是简单的功能 CAN_TX 和 CAN_RX，则通过 ID 确定发送标准帧还是扩展帧。若有增强版本，则通过输入端 EXTENDED 设定。因此 ID 范围 0...2047 中的扩展帧不可通过简单的功能发送。

输入端参数

7524

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
ID	DWORD	数据对象标识符编号： 正常帧 (2 048 个 ID)： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (536 868 864 个 ID)： 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7527

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

RESULT 的可能结果：

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

CAN_TX_ENH

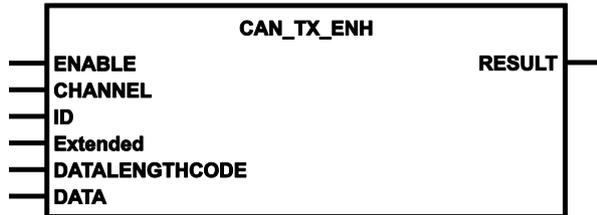
7558

= CAN TX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7559

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

- 它是 11 位还是 29 位标识符?
 - 可预设其他输入端, 以便 **CAN_TX** (→ 页 84) 不需要。
- > 如果在一个周期内多次调用该 FB 的实例, 数据亦会发送多次。

输入端参数

7634

参数	数据类型	说明
ENABLE	BOOL	FALSE ⇔ TRUE (edge): 初始化功能块 (仅 1 个周期) > 读取功能块输入 TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID	DWORD	数据对象 ID 编号: 正常帧 (2 ¹¹ 个 ID): 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹ 个 ID): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
DATALengthCODE	BYTE	= 数据长度码 待发送数据字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7527

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

RESULT 的可能结果：

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

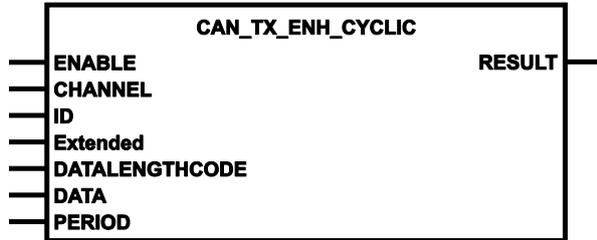
CAN_TX_ENH_CYCLIC

7568

= CAN TX 增强循环

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyyzz.LIB

CODESYS 中的符号:**说明**

7569

CAN_TX_ENH_CYCLIC 用于次循环传送 CAN 消息。

否则, FB 相当于 **CAN_TX_ENH** (→ 页 86).

▶ 通过参数 PERIOD 设定周期。

❗ 如果周期太短, 则可能导致总线负载高, 进而影响整个系统的性能。

输入端参数

7582

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID	DWORD	数据对象 ID 编号: 正常帧 (2 ¹¹ 个 ID): 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹ 个 ID): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
DataLengthCode (参数可选)	BYTE := 8	待发送数据的长度 (0...8 字节)
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)
PERIOD	TIME	周期

输出端参数

7510

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
8	08	功能块启用中
9	09	CAN 未启用
250	FA	错误：FiFo 已满 - 数据丢失

功能元件：RAW-CAN 远程

内容

CAN_REMOTE_REQUEST	93
CAN_REMOTE_RESPONSE	95

15057

CAN_REMOTE_REQUEST

7625

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxyyzz.LIB

CODESYS 中的符号:



说明

7627

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

输入端参数

7628

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇔ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
ID	DWORD	数据对象 ID 编号： 正常帧 (2 ¹¹ 个 ID)： 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹ 个 ID)： 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)

输出端参数

7629

参数	数据类型	说明
DATALENGTHCOUNT	BYTE	数据长度计数 已接收数据字节数量
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
5	05	FB 正在处理中 - 仍在接收
9	09	CAN 未启用
242	F2	错误：不可设定

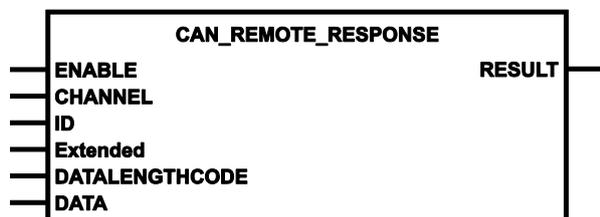
CAN_REMOTE_RESPONSE

7631

单元类型 = 功能块 (FB)

元件类型 ifm_RawCAN_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7633

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

该 FB 很大程度上取决于装置类型。 仅可设定有限的远程消息。

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

输入端参数

7634

参数	数据类型	说明
ENABLE	BOOL	FALSE ⇔ TRUE (edge): 初始化功能块 (仅 1 个周期) > 读取功能块输入 TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
ID	DWORD	数据对象 ID 编号: 正常帧 (2 ¹¹ 个 ID): 0...2 047 = 0x0000 0000...0x0000 07FF 扩展帧 (2 ²⁹ 个 ID): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (参数可选)	BOOL := FALSE	TRUE: 扩展帧 (ID = 0...2 ²⁹ -1) FALSE: 正常帧 (ID = 0...2 ¹¹ -1)
DATALENGTHCODE	BYTE	= 数据长度码 待发送数据字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7636

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
6	06	FB 正在处理中 - ID 远程未启用
7	07	FB 正在处理中 - ID 远程启用中

5.2.3 功能元件：CANopen

内容	
功能元件：CANopen 状态	98
功能元件：CANopen 网络管理	109
功能元件：CANopen 对象目录	115
功能元件：CANopen SDO	122
功能元件：CANopen SYNC	137
功能元件：CANopen 保护	142
功能元件：CANopen 紧急	147

15059

就 CANopen 而言，IFM 电子提供一系列功能元件，将在下文中解释。

功能元件：CANopen 状态

内容	
CANOPEN_ENABLE	99
CANOPEN_GETBUFFERFLAGS	101
CANOPEN_GETSTATE	103
CANOPEN_SETSTATE	106

15061

CANOPEN_ENABLE

7785

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyyz.LIB

CODESYS 中的符号:



说明

7787

CANOPEN_ENABLE 允许开启或关闭 CANopen 主站。

- ▶ **!** 在应用程序中始终针对 CAN 接口 **CANOPEN_ENABLE** (→ 页 99) 调用 FB 自身的实例！

! 为避免保护或检测信号错误，必须先通过相应的序列“关闭”节点。

如果主站在停止后重新启动，所有其他连接的节点亦需要重新初始化。

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

配置的波特率仅在 **CAN_ENABLE** (→ 页 62) 之前未启用的情况下予以采用。

输入端参数

7788

参数	数据类型	说明
ENABLE	BOOL := TRUE	TRUE: <ul style="list-style-type: none"> • 启用针对所选通道的 CANopen • 根据配置设定启动 CANopen 管理器或 CANopen 装置 FALSE: <ul style="list-style-type: none"> • 禁用针对所选通道的 CANopen • 终止 CANopen 管理器或 CANopen 装置
CHANNEL	BYTE	CAN 接口 (1...n)，视装置而定
BaudRate (参数可选)	WORD := 0	波特率 [kbits/s] 允许值 = 20, 50, 100, 125, 250, 500, 800, 1 000 0 = 使用 PLC 配置的设置

输出端参数

7789

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
14	0E	FB 已启用 CANopen 管理器配置装置并发送 SDO
15	0F	FB 已启用 CANopen 管理器启动
238	EE	错误：CANopen 配置太大，无法启动
239	EF	错误：CANopen 管理器无法启动
242	F2	错误：不可设定

CANOPEN_GETBUFFERFLAGS

7890

= 获取缓冲标志

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7892

CANOPEN_GETBUFFERFLAGS 提供关于缓冲标志的信息。

标志可通过可选输入端复位。

功能块返回溢出标志的状态。

输入端参数

7893

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

输出端参数

7894

参数	数据类型	说明
RXOVFL	BOOL	RX 溢出标志状况 TRUE: 接收缓冲区溢出 FALSE: 接收缓冲区无溢出
RXWARN	BOOL	RX 溢出警告标志状况 TRUE: 接收缓冲区程度处于临界状态 FALSE: 接收缓冲区程度处于非临界状态
TXOVFL	BOOL	TX 溢出标志状况 TRUE: 传送缓冲区溢出 FALSE: 传送缓冲区无溢出
TXWARN	BOOL	TX 溢出警告标志状况 TRUE: 传送缓冲区程度处于临界状态 FALSE: 传送缓冲区程度处于非临界状态
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

CANOPEN_GETSTATE

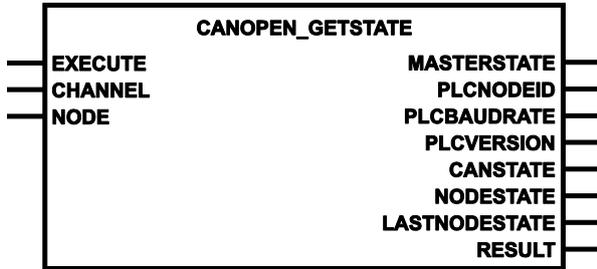
7865

= 获取状态

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7867

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

输入端参数

7868

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

输出端参数

7869

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

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

RESULT 的可能结果：

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

CANOPEN_SETSTATE

7858

= 设定状态

单元类型 = 功能块 (FB)

元件类型 ifm_CANOpen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7860

可通过 CANOPEN_SETSTATE 设定主站、从属装置或网络中节点的参数。

主站、节点或装置的 NMT 状态在 CAN 堆栈中进行处理，或通过 FB 的命令进行处理

CANOPEN_NMTSERVICES (→ 页 [112](#)). 同时，还进行容许性检查。因为一致性的原因，未针对该目的提供输入端。

输入端参数

7861

参数	数据类型	说明
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (edge): 执行一次功能元件 否则： 功能元件不会启用 处理已经启动的功能元件。
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
NODE	BYTE	Node ID = 节点的 ID (0...127) 装置作为 CANopen 主站： 值 = 0： 更改仅指装置本身。 值不为 0： 网络节点 Node ID, 其参数待更改。 既定的设定仅适用于该节点 (而不是装置) 装置作为 CANopen 从站： 在从站模式中, 从站的 Node ID 可通过该输入端设定。 值 = 0： 无操作 值不为 0： 功能块将该值作为装置的新 Node ID。
GlobalStart (参数可选)	BOOL := TRUE	要求： 启动 IEC 程序后必须立即调用 FB。 该设定覆盖配置的设置。 TRUE: 同时启动所有参与者 FALSE: 逐个启动所有参与者
CfgTimeout (参数可选)	TIME := T#0ms	设定节点的配置超时： 值 = 0： 无操作 - 保留配置数据 值不为 0： 利用新值覆盖配置中的数据
InitNoSdos (参数可选)	BOOL := FALSE	对于 NODE 所示节点, 初始化期间, ... TRUE: 切勿发送配置数据 FALSE: 发送配置的 SDO

输出端参数

7862

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

RESULT 的可能结果：

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

功能元件：CANopen 网络管理

内容

CANOPEN_GETNMTSTATESLAVE	110
CANOPEN_NMTSERVICES.....	112

15063

CANOPEN_GETNMTSTATESLAVE

7851

= 获取网络管理状态从站

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7853

▶ 仅在装置作为 CANopen 从站运行时使用 FB !

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

输入端参数

7854

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

输出端参数

7855

参数	数据类型	说明
NMTSTATE	BYTE	节点的网络运行状态 0 = INIT 1 = OPERATIONAL 2 = PRE-OPERATIONAL 3 = 已停止
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

CANOPEN_NMTSERVICES

= 网络管理服务

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

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

 NMT = 网络管理

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

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

输入端参数

7847

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

输出端参数

7848

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	功能块启用中
35	23	至少 1 个配置 SDO 未成功
36	24	节点已初始化
37	25	请求初始化时, 节点不处于 PRE-OPERATIONAL 模式
43	2B	主站/从站未初始化
241	F1	错误: CANopen 状态转换未获准许
242	F2	错误: 不可设定

功能元件：CANopen 对象目录

内容	
CANOPEN_GETODCHANGEDFLAG	116
CANOPEN_READOBJECTDICT	118
CANOPEN_WRITEOBJECTDICT	120

15065

CANOPEN_GETODCHANGEDFLAG

7927

= 获取对象目录更改标志

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7928

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

输入端参数

7930

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

输出端参数

7931

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

RESULT 的可能结果：

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

CANOPEN_READOBJECTDICT

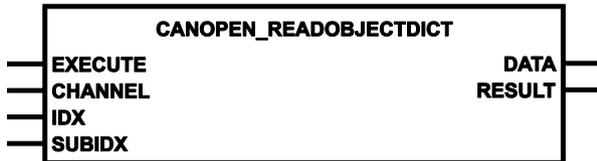
7933

= 读取对象目录

单元类型 = 功能块 (FB)

元件类型 ifm_CANOpen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7935

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

输入端参数

7936

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

输出端参数

7937

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	功能块还未执行
40	28	对象目录条目无效
242	F2	错误：不可设定

CANOPEN_WRITEOBJECTDICT

7940

= 写入对象目录

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7942

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

通知

这样可能导致重要系统设定篡改，例如：

- 保护时间
- 检测信号时间

▶ 谨慎核实输入参数！

输入端参数

7943

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

输出端参数

7945

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	功能块还未执行
40	28	对象目录条目无效
242	F2	错误：不可设定

功能元件：CANopen SDO

内容	
CANOPEN_SDOREAD.....	123
CANOPEN_SDOREADBLOCK.....	125
CANOPEN_SDOREADMULTI.....	128
CANOPEN_SDOWRITE.....	130
CANOPEN_SDOWRITEBLOCK.....	132
CANOPEN_SDOWRITEMULTI.....	135

2071

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

CANOPEN_SDOREAD

7791

= SDO 读取

单元类型 = 功能块 (FB)

元件类型 ifm_CANOpen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7793

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

Expedited SDO = 加速服务数据对象

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

输入端参数

7794

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

输出端参数

7795

参数	数据类型	说明
LEN	BYTE	接收字节数量 (1...4)
DATA	DWORD	接收数据值 (高达 4 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 - 数据有效
5	05	FB 启用中 - 还未接收到数据
32	20	SDO 传送被客户端或服务中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
242	F2	错误：不可设定
255	FF	缓冲区溢出 - 接收了太多数据字节

CANOPEN_SDOREADBLOCK

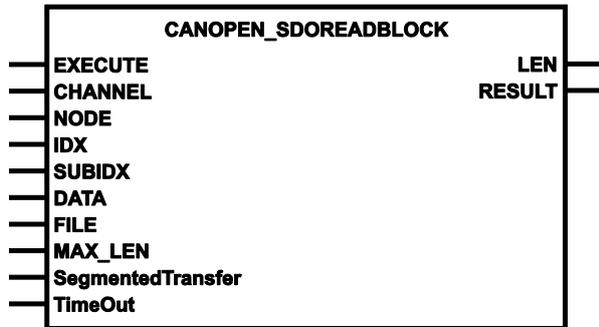
14942

= SDO 读取功能块

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

14943

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

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

多帧 SDO 的长度一般无限制。

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

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

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

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

输入端参数

14945

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

输出端参数

14951

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

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
16	10	传输作为分段下载启用
17	11	传输作为块下载启用
32	20	SDO 传送被客户端或服务中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
64	40	错误：写入指针在允许的数据范围之外
65	41	错误：文件无法打开
66	42	写入至文件时出错
242	F2	错误：不可设定

CANOPEN_SDOREADMULTI

7806

= SDO 读取 multi

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7808

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

输入端参数

7809

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

输出端参数

7810

参数	数据类型	说明
LEN	DWORD	已接收字节数量 允许值 = 1...2 048 = 0x0000 0001...0x0000 0800
DATA	ARRAY [0..SDOMAXDATA] OF BYTE	SDO 数据传输用户数据缓冲区内存
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
5	05	FB 启用中 – 还未接收到数据
32	20	SDO 传送被客户端或服务中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
242	F2	错误：不可设定
255	FF	错误：内存不足以使用多帧

CANOPEN_SDOWNRITE

7825

= SDO 写入

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7826

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

 Expedited SDO = 加速服务数据对象

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

输入端参数

7828

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

输出端参数

7829

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

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
8	08	功能块启用中
32	20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
242	F2	错误：不可设定

CANOPEN_SDOWNWRITEBLOCK

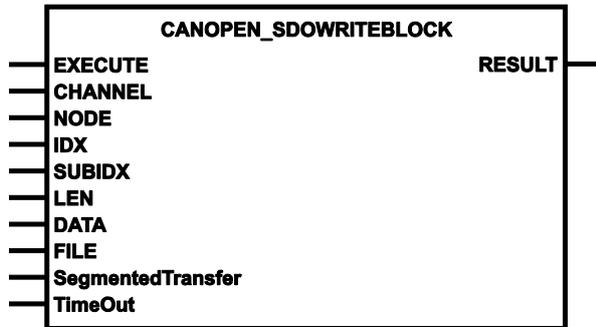
14961

= SDO 写入功能块

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

14963

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

如有必要，您可通过 FB 输入端切换至分段传输。

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

多帧 SDO 的长度一般无限制。

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

- ▶ 将地址传送至 FB（通过指针访问以便读取）。

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

- ▶ 将文件路径和名称传送至 FB（数据按二进制格式从中读取）。

输入端参数

14964

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

输出端参数

14968

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

RESULT 的可能结果：

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

CANOPEN_SDOWRITEMULTI

7832

= SDO 写入 multi

单元类型 = 功能块 (FB)

元件类型 ifm_CANOpen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7834

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

输入端参数

7835

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

输出端参数

7836

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
8	08	功能块启用中
32	20	SDO 传送被客户端或服务器中止 (SDO 中止代码 0x80)
33	21	TIMEOUT 已过
242	F2	错误：不可设定

功能元件：CANopen SYNC

内容

CANOPEN_GETSYNCSTATE.....	138
CANOPEN_SETSYNCSTATE	140

15069

CANOPEN_GETSYNCSTATE

7871

= 获取 SYNC 状态

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7872

CANOPEN_GETSYNCSTATE reads...

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

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

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

输入端参数

7874

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

输出端参数

7875

参数	数据类型	说明
SYNC	BOOL	SYNC 功能的状态 TRUE: SYNC 已启用： 在 主站模式 中，SYNC 电报根据配置中的设定生成，同步 PDO 传送和接收。 在 从站模式 中，SYNC 电报接收并相应地处理。 FALSE: SYNC 未启用
SYNCERROR	BYTE	(同步错误) SYNC 错误消息 0 = 无错误 >0 = SYNC 错误 (从站模式)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

CANOPEN_SETSYNCSTATE

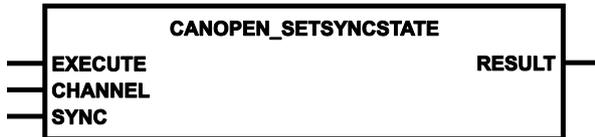
7883

= 设定 SYNC 状态

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7884

SYNC 通过 CANOPEN_SETSYNCSTATE 开启和关闭。

输入端参数

7886

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

输出端参数

7887

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	功能块还未执行
38	26	SYNC 不可启用
242	F2	错误：不可设定

功能元件：CANopen 保护

内容

CANOPEN_GETGUARDHBERRLIST	143
CANOPEN_GETGUARDHBSTATSLV	145

15071

CANOPEN_GETGUARDHBERRLIST

7896

= 获取保护和检测信号错误列表

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7898

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

- 保护错误
- 检测信号错误

输入端参数

7899

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

输出端参数

7900

参数	数据类型	说明
N_NODES	WORD	包含检测信号或保护错误的节点的数量 0 = 无节点有防护或检测信号错误
NODEID	ARRAY [0..MAXGUARDERROR] OF BYTE	包含检测信号或保护错误的 Node ID 列表 最近的条目位于索引 0。 MAXGUARDERROR 取决于装置 → 章节 装置的性能极限 (CANopen) (→ 页 37)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

CANOPEN_GETGUARDHBSTATSLV

7902

= 获取保护和检测信号状态从站

单元类型 = 功能块 (FB)

元件类型 ifm_CANOpen_NT_Vxyzzz.LIB

CODESYS 中的符号:



说明

7904

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

- 监控节点保护
- 监控检测信号

控制器可能是检测信号生成器或检测信号消耗器。

输入端参数

7905

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

输出端参数

7906

参数	数据类型	说明
GUARDSTATE	BYTE	节点保护状态 0 = 0x00 = 无错误 (或: 未启用) 1 = 0x01 = 超时 (配置) 127 = 0x7F = 未接收到保护消息
PROD_HBSTATE	BYTE	控制器作为检测信号生成器: 0 = 0x00 = 未启用 1 = 0x01 = 启用
CONS_HBSTATE	BYTE	控制器作为检测信号消耗器: 0 = 0x00 = 无故障 1 = 0x01 = 超时 (配置) 127 = 0x7F = 还未接收到检测信号消息
CONS_HBCOBID	WORD	控制器消耗器检测信号对其作出响应的检测信号消息的 COB-ID (配置)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果:

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

功能元件：CANopen 紧急

内容

CANOPEN_GETEMCYMESSAGES.....	148
CANOPEN_GETERRORREGISTER.....	150
CANOPEN_SENDEMCMYMESSAGE	152

15073

CANOPEN_GETEMCYMESSAGES

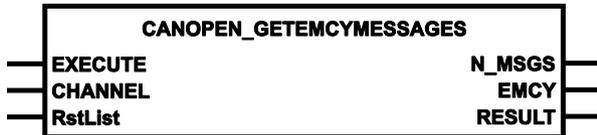
7921

= 获取紧急消息

单元类型 = 功能块 (FB)

元件类型 ifm_CANOpen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7923

CANOPEN_GETEMCYMESSAGES 返回自上次删除消息以来控制器从网络其他节点接收的所有紧急消息。

该列表可通过设定相应的输入端复位。MAXEMCYMSGS 消息最大值存储。每条消息包含关于从哪个节点发送的信息。最近的消息位于索引 0。

输入端参数

7924

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

输出端参数

7925

参数	数据类型	说明								
N_MSGS	DWORD	累积消息的数量								
EMCY	ARRAY [0..MAXEMCYMSG] OF T_EMCY	紧急消息 最近的条目位于索引 0。 T_EMCY 结构： <table border="1" style="margin-left: 20px;"> <tr> <td>.NODEID</td> <td>消息来源节点的 ID</td> </tr> <tr> <td>.EEC</td> <td>紧急错误代码</td> </tr> <tr> <td>.ER</td> <td>错误寄存器</td> </tr> <tr> <td>.MSEF</td> <td>制造商特定错误代码</td> </tr> </table> MAXEMCYMSG = 10	.NODEID	消息来源节点的 ID	.EEC	紧急错误代码	.ER	错误寄存器	.MSEF	制造商特定错误代码
.NODEID	消息来源节点的 ID									
.EEC	紧急错误代码									
.ER	错误寄存器									
.MSEF	制造商特定错误代码									
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)								

RESULT 的可能结果：

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

CANOPEN_GETERRORREGISTER

7915

= 获取错误寄存器

单元类型 = 功能块 (FB)

元件类型 ifm_CANOpen_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7917

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

输入端参数

7918

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

输出端参数

7919

参数	数据类型	说明
ER	BYTE	错误寄存器 0x1001 的内容
ERROR_FIELD	ARRAY [0..MAXERR] OF DWORD	错误寄存器 0x1003 的内容 索引 0 = 已存储错误的数量 索引 1..MAXERR = 已存储错误最近的错误位于索引 1。 预设：MAXERR = 5
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

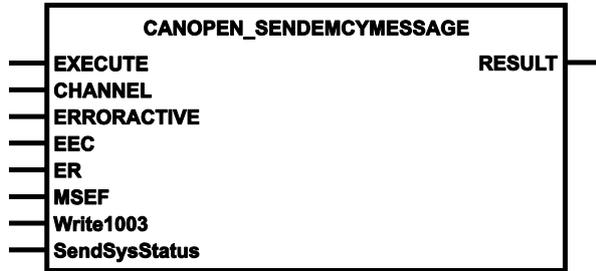
CANOPEN_SENDEMCMYMESSAGE

7908

= 发送紧急消息

单元类型 = 功能块 (FB)

元件类型 ifm_CANopen_NT_Vxxyzz.LIB

CODESYS 中的符号:**说明**

7910

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

输入端参数

7911

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

输出端参数

7912

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	FB 已启用 – 还未处理
39	27	配置中无对象 1001 ₁₆
242	F2	错误：不可设定

5.2.4 功能元件：SAE J1939

内容	
功能元件：SAE J1939 状态.....	155
功能元件：SAE J1939 请求.....	167
功能元件：接收 SAE J1939.....	172
功能元件：传送 SAE J1939.....	179
功能元件：SAE J1939 诊断.....	188

2273

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

功能元件：SAE J1939 状态

内容	
J1939_ENABLE.....	156
J1939_GETDABYNAME.....	158
J1939_NAME.....	161
J1939_STATUS.....	165

15077

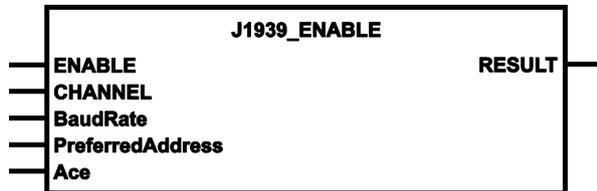
J1939_ENABLE

7641

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7642

就 J1939 堆栈初始化而言, J1939_ENABLE 设为 TRUE=1。

- > 该 FB 也会促使启动 CFG 文件软输入/输出。
- > 如果 CAN_ENABLE 之前未启用, 则仅可采用不同的波特率。

ACE = 地址声明启用:

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

输入端参数

7643

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

输出端参数

8542

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

RESULT 的可能结果：

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

J1939_GETDABYNAME

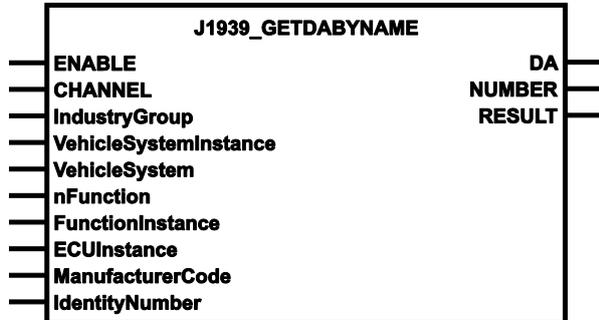
7664

= 获取目标任意名称

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7665

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

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

输入端参数

7667

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
IndustryGroup (参数可选)	BYTE = 0xFF	industry group = 装置的行业组 允许值 = 0...7 255 = 0xFF = 针对所有的过滤
VehicleSystemInstance (参数可选)	BYTE := 0xFF	车辆系统实例 允许值 = 0...15 = 0x00...0x0F 255 = 0xFF = 针对所有的过滤
VehicleSystem (参数可选)	BYTE := 0xFF	车辆系统 允许值 = 0...127 = 0x00...0x7F 255 = 0xFF = 针对所有的过滤
nFunction (参数可选)	WORD := 0xFFFF	装置的功能 允许值 = 0...255 = 0x0000...0x00FF 65 535 = 0xFFFF = 针对所有项的过滤
FunctionInstance (参数可选)	BYTE := 0xFF	功能的实例 允许值 = 0...31 = 0x00...0x1F 255 = 0xFF = 针对所有的过滤
ECUInstance (参数可选)	BYTE := 0xFF	控制装置的实例 允许值 = 0...7 255 = 0xFF = 针对所有的过滤
ManufacturerCode (参数可选)	WORD := 0xFFFF	制造商代码 (必须从 SAE 申请) 允许值 = 0...2047 (2 ¹¹ -1) = 0x0000...0x07FF 65 535 = 0xFFFF = 针对所有项的过滤
IdentityNumber (参数可选)	DWORD := 0xFFFF FFFF	装置的序列号 (不得覆盖) 允许值 = 0...2047 (2 ¹¹ -1) = 0x0000 0000...0x0000 07FF 4 294 967 295 = 0xFFFF FFFF = 针对所有项的过滤

输出端参数

7668

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

RESULT 的可能结果：

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

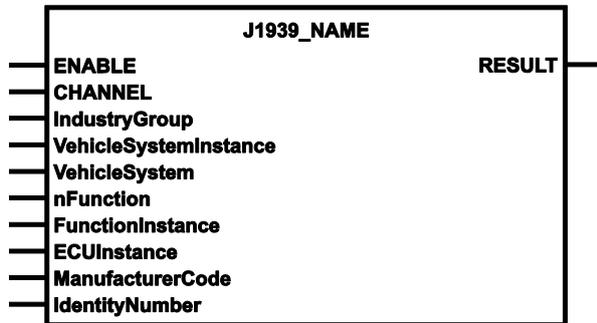
J1939_NAME

7646

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7648

可通过 J1939_NAME 向装置提供名称以在网络中识别。

默认情况下，使用 **IFM** 名称。

用户有以下选项可更改装置的名称：

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

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

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

表格：符合 SAE J1939-81 的 64 位 NAME 信息的构成



输入端参数

7652

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

输出端参数

7661

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

RESULT 的可能结果：

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

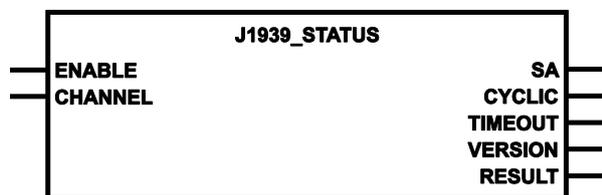
J1939_STATUS

7670

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7672

通过 J1939_STATUS，相关信息可读回至 J1939 堆栈。

输入端参数

7673

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定

输出端参数

7674

参数	数据类型	说明
SA	BYTE	已声明源地址
CYCLIC	WORD	循环消息的数量
TIMEOUT	BYTE	未及时针对处理图像提供数据的节点的源地址 255 = 0xFF = 所有节点及时发送
VERSION	DWORD	IFM CAN 堆栈库的 vabvab
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	协议已启用
2	02	协议未启用
3	03	源地址已请求
4	04	地址丢失
242	F2	错误：不可设定

功能元件：SAE J1939 请求

内容

J1939_SPEC_REQ	168
J1939_SPEC_REQ_MULTI	170

15079

J1939_SPEC_REQ

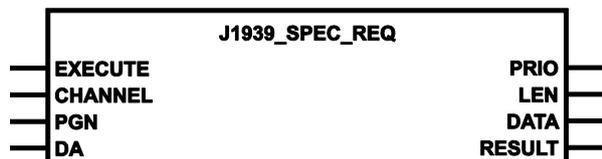
15023

= J1939 特定请求

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

15026

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

如果请求多帧消息：

- FB 提供前 8 字节的数据
- RESULT 显示错误

输入端参数

15028

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

输出端参数

15029

参数	数据类型	说明
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	接收字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
5	05	FB 启用中 – 还未接收到数据
64	40	错误：接收多帧
242	F2	错误：不可设定

J1939_SPEC_REQ_MULTI

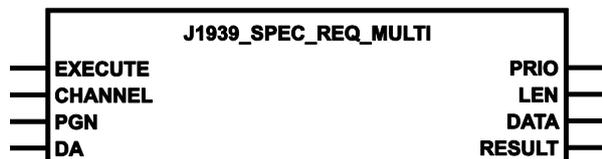
15033

= J1939 特定请求多帧消息

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

15036

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

输入端参数

15037

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

输出端参数

15038

参数	数据类型	说明
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	待传送数据字节数量 允许 = 1...1 785 = 0x0001...0x06F9
DATA	ARRAY [0..1784] OF BYTE	已接收数据 (1...1785 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
5	05	FB 启用中 – 还未接收到数据
242	F2	错误：不可设定

功能元件：接收 SAE J1939

内容	
J1939_RX.....	173
J1939_RX_FIFO.....	175
J1939_RX_MULTI.....	177

15081

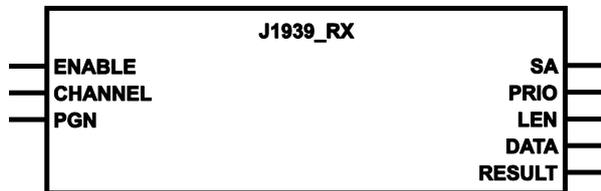
J1939_RX

7724

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7725

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

输入端参数

7726

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF

! PGN = 0 未使用。

输出端参数

7727

参数	数据类型	说明
SA	BYTE	变送器源地址
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	接收字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
5	05	FB 启用中 – 还未接收到数据
9	09	CAN 未启用
242	F2	错误：不可设定

J1939_RX_FIFO

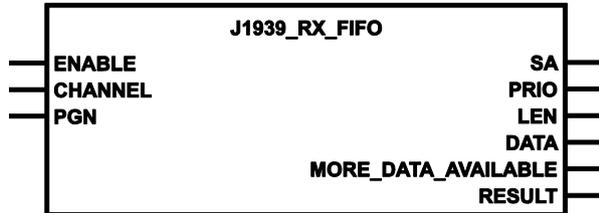
7732

= 包含 FIFO 的 J1939 RX

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7733

J1939_RX_FIFO 供接收所有特定消息并连续从 FIFO 读取。

输入端参数

7734

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1..n), 视装置而定
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF

! PGN = 0 未使用。

输出端参数

7735

参数	数据类型	说明
SA	BYTE	变送器源地址
PRI0	BYTE	消息优先级 (0...7)
LEN	BYTE	接收字节数量 (0...8)
DATA	ARRAY [0..7] OF BYTE	已接收数据, (1...8 字节)
MORE_DATA_AVAILABLE	BOOL	TRUE: FiFo 中有进一步接收的数据 FALSE: FiFo 中无进一步接收的数据
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	FB 执行完成且无错误 - 数据有效
5	05	FB 启用中 - 还未接收到数据
242	F2	错误：不可设定
250	FA	错误：FiFo 已满 - 数据丢失

J1939_RX_MULTI

7736

= J1939 RX 多帧消息

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7741

J1939_RX_MULTI 供接收多帧消息。

输入端参数

7743

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

! PGN = 0 未使用。

输出端参数

7744

参数	数据类型	说明
SA	BYTE	变送器源地址
PRI0	BYTE	消息优先级 (0...7)
LEN	WORD	已接收字节数量 允许值 = 0...1 785 = 0x0000 0000...0x0000 06F9
DATA	ARRAY [0..1784] OF BYTE	待传送数据 (1...1785 字节)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
5	05	FB 启用中 – 还未接收到数据
242	F2	错误：不可设定

功能元件： 传送 SAE J1939

内容	
J1939_TX	180
J1939_TX_ENH.....	182
J1939_TX_ENH_CYCLIC	184
J1939_TX_ENH_MULTI.....	186

15083

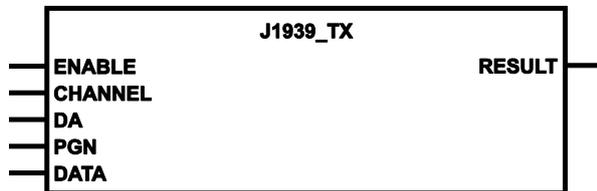
J1939_TX

7688

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7689

J1939_TX 是最简单的单帧消息传送方式。

输入端参数

7690

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE := 249	DA = ECU 目标地址 PGN > 61139: 忽略参数 DA
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1..8 字节)

输出端参数

7693

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

RESULT 的可能结果：

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

J1939_TX_ENH

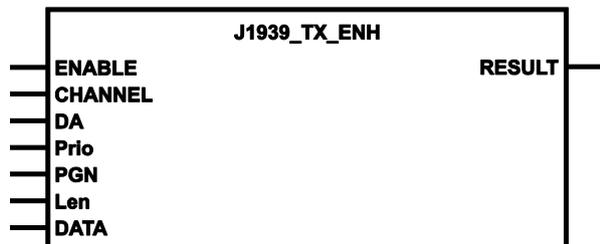
7696

= J1939 TX 增强

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7697

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

- 传送优先级
- 数据长度

多帧消息 → J1939_TX_ENH_MULTI (→ 页 [186](#)).

输入端参数

7702

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE := 249	DA = ECU 目标地址 PGN > 61139: 忽略参数 DA
Prio (参数可选)	BYTE := 3	消息优先级 允许值 = 0...7
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
Len (参数可选)	BYTE := 8	待传送字节数量 允许值 = 0...8
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1...8 字节)

输出端参数

7969

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

RESULT 的可能结果：

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

J1939_TX_ENH_CYCLIC

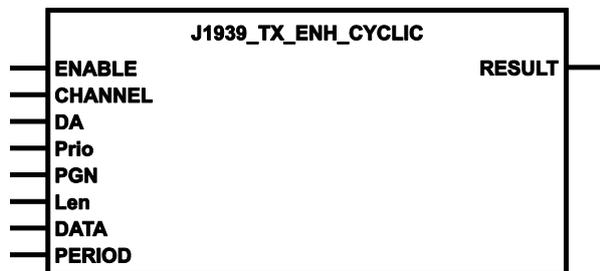
7716

= J1939 TX 增强循环

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7718

J1939_TX_ENH_CYCLIC 用于次循环传送 CAN 消息。

否则, FB 相当于 **J1939_TX_ENH** (→ 页 [182](#)).

▶ 通过参数 PERIOD 设定周期。

❗ 如果周期太短, 则可能导致总线负载高!
总线负载可影响整个系统的性能。

输入端参数

7719

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE := 249	DA = ECU 目标地址 PGN > 61139: 忽略参数 DA
Prio (参数可选)	BYTE := 3	消息优先级 允许值 = 0...7
PGN	DWORD	PGN = 参数群组编号 允许 = 0...262 143 = 0x00000000...0x0003FFFF
Len (参数可选)	BYTE := 8	待传送字节数量 允许值 = 0...8
DATA	ARRAY [0..7] OF BYTE	待传送数据 (1..8 字节)
PERIOD	TIME	周期

输出端参数

7720

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

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
8	08	功能块启用中
242	F2	错误：不可设定

J1939_TX_ENH_MULTI

= J1939 TX 增强多帧消息

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

7699

CODESYS 中的符号:



说明

多帧消息的传送通过 J1939_TX_ENH_MULTI 执行。

FB 相当于 **J1939_TX_ENH** (→ 页 [182](#)). 此外, 还可确定是否应作为 BAM (广播公告消息) 执行传送。

7705

输入端参数

7712

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

输出端参数

7714

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	功能块执行完成且无错误
8	08	功能块启用中
65	41	错误：不可传送
242	F2	错误：不可设定

功能元件：SAE J1939 诊断

内容	
J1939_DM1RX	189
J1939_DM1TX.....	192
J1939_DM1TX_CFG	195
J1939_DM3TX.....	197

15085

J1939_DM1RX

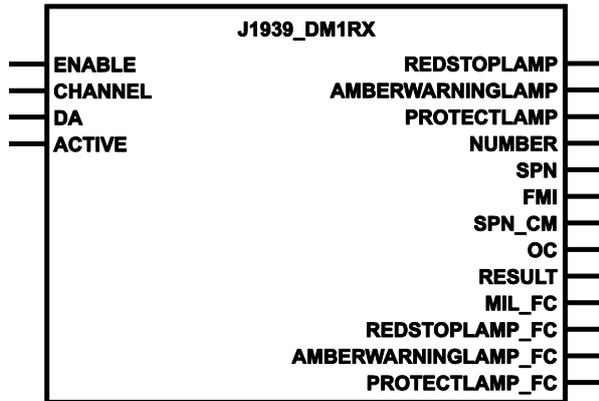
14977

= J1939 诊断消息 1 RX

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

7761

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

输入端参数

14979

参数	数据类型	说明
ENABLE	BOOL := FALSE	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1...n), 视装置而定
DA	BYTE	DA = 可检索 DTC 的 ECU 的目标地址。 DA = 254: 从装置本身读取 DTC
ACTIVE	BOOL	TRUE: 读取有效 DTC (DM1) FALSE: 读取之前有效的 DTC (DM2)

输出端参数

14980

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

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

RESULT 的可能结果：

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

J1939_DM1TX

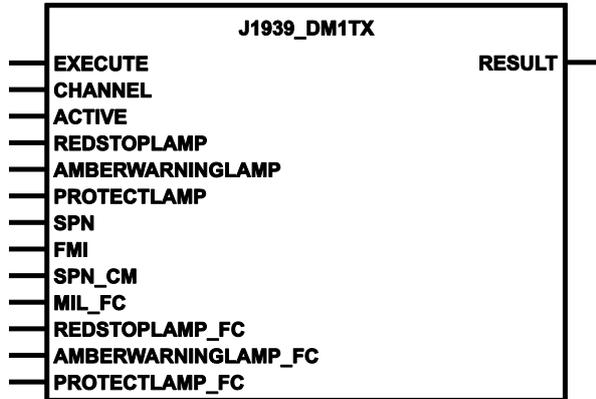
14993

= J1939 诊断消息 1 TX

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyyzz.LIB

CODESYS 中的符号:



说明

7747

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

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

输入端参数

14995

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

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

输出端参数

7750

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

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	数据在错误内存中标记为“有效”
242	F2	错误：不可设定

J1939_DM1TX_CFG

15424

= J1939 诊断消息 1 TX 可配置

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_V02.00.02.LIB 或更高

CODESYS 中的符号:



说明

15426

从运行时系统 V03.00.03 开始，只要针对相应的 CAN 接口调用 FB，**J1939_ENABLE** (→ 页 156) CAN 堆栈就每秒钟自动发送 DM1 消息。

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

FB 提供以下循环传送 DM1 消息的模式：

MODE = 0 (预设)	CAN 堆栈每秒发送符合标准的 DM1 "zero active faults" 消息。 可通过 FB 手动传送 DM1 J1939_DM1TX (→ 页 192)消息。
MODE = 1	CAN 堆栈不发送 DM1 "zero active faults" 消息。 DM2 请求自动响应。 可通过 FB 手动传送 DM1 J1939_DM1TX (→ 页 192) 消息。
MODE = 2	CAN 堆栈不发送循环 DM1 "zero active faults" 消息 CAN 堆栈也不会自动响应 DM2 请求。

输入端参数

15427

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
CHANNEL	BYTE	CAN 接口 (1..n)，视装置而定
MODE	BYTE := 0	功能块的工作模式 允许 = 0..2 (→ FB 说明)

输出端参数

15429

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

RESULT 的可能结果：

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

J1939_DM3TX

15002

= J1939 诊断消息 3 TX

单元类型 = 功能块 (FB)

元件类型 ifm_J1939_NT_Vxxyzz.LIB

CODESYS 中的符号:



说明

15004

您可通过 J1939_DM3TX (DM = 诊断消) 删除其他装置上的无效 DTC。

> 接收 DM3 消息后, 即会在硬件配置中删除错误内存中的所有无效错误。

输入端参数

15006

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

输出端参数

15008

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

RESULT 的可能结果:

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

5.2.5 功能元件：系统

内容	
FLASH_INFO	199
FLASH_READ	201
GET_APP_INFO	203
GET_HW_INFO	205
GET_IDENTITY	207
GET_SW_INFO	208
GET_SW_VERSION	210
MEM_ERROR	212
MEMCPY	214
OHC	217
SET_IDENTITY	219
SET_LED	220
SET_PASSWORD	222
TIMER_READ_US	224

15067

我们将在此讲述可供您进行以下操作的 **IFM** 功能：

- 管理内存内容
- 读取软件和硬件信息
- 设定或读取各种数据和参数

FLASH_INFO

11580

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

11588

FLASH_INFO 从用户闪存读取信息：

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

输入端参数

11589

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

输出端参数

11590

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

RESULT 的可能结果：

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

FLASH_READ

8147

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

11579

FLASH_READ 有助于直接从闪存读取不同类型的数据。

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

- ▶ 通过 SRC + LEN 产生的地址必须 $\leq 65\ 408$ 。
- ▶ 就目标地址而言，DST 适用：
 - ❗ 通过运算符 ADR 确定地址并将其分配至 FB！

输入端参数

8148

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
DST	DWORD	目标地址 ❗ 通过运算符 ADR 确定地址并将其分配至 FB！
SRC	DWORD	内存中的相对起始地址 有效 = 0...65 407 = 0x0000 0000...0x0000 FF7F
LEN	WORD	待传送数据字节数量 (≥ 1)

输出端参数

8152

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
152	98	不允许的内存区域： <ul style="list-style-type: none"> • 无效的源地址 • 无效的目标地址 • 无效的字节数量

GET_APP_INFO

11581

= 获取应用程序信息

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

11593

GET_APP_INFO 提供关于存储在装置上的应用程序软件的信息：

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

输入端参数

11594

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

输出端参数

11595

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

RESULT 的可能结果：

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

GET_HW_INFO

11582

= 获取硬件信息

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

1599

GET_HW_INFO 提供关于装置硬件的信息：

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

输入端参数

11600

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

输出端参数

11601

参数	数据类型	说明
ORDER_NUM	STRING(24)	IFM 产品编号 (例如: CR0403)
NAME	STRING(24)	产品名称 (例如: "BasicController 12/12")
SERIAL	STRING(24)	装置的序列号 (例如: "000045784")
REVISION	STRING(24)	装置的硬件修订版本 (例如: "V01.00.01")
MAN_DATE	STRING(24)	装置制造日期 (例如: "20111007123800")
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

GET_IDENTITY

8166

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

15411

GET_IDENTITY 读取装置中存储的 ID (之前已通过以下方式保存 : SET_IDENTITY (→ 页 219)).

输入端参数

8167

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

输出端参数

8168

参数	数据类型	说明
APP_IDENT	STRING(80)	应用程序 ID 作为最多 80 个字符的字符串, 例如: "Crane1704"
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果 :

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	FB 执行完成且无错误 - 数据有效
155	9B	不可读取值

GET_SW_INFO

11583

= 获取软件信息

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

11596

GET_SW_INFO 提供关于装置系统软件的信息：

- 软件名称，
- 软件版本，
- 版本号，
- 版本日期。

输入端参数

11597

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

输出端参数

11598

参数	数据类型	说明
NAME	STRING(24)	系统软件的名称 (例如: "BasicSystem")
VERSION	STRING(24)	系统软件的版本 (例如: "V02.00.03")
BUILD_NUM	STRING(24)	系统软件的版本号 (例如: "45")
BUILD_DATE	STRING(24)	系统软件的版本日期 (例如: "20111006123800")
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果:

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

GET_SW_VERSION

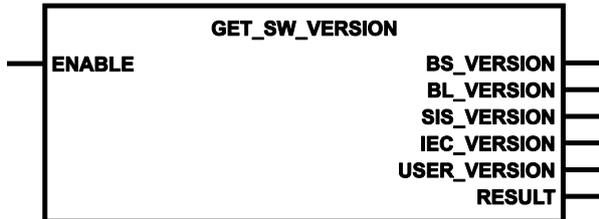
14763

= 获取软件版本

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxyyzz.LIB

CODESYS 中的符号:



说明

14765

GET_SW_VERSION 提供关于装置中的软件的信息：

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

输入端参数

14766

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

输出端参数

14767

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

RESULT 的可能结果：

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

MEM_ERROR

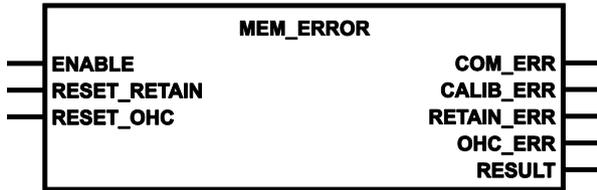
14770

= 内存错误

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

14772

MEM_ERROR 利用信号通知某些参数或内存中的错误。

内存区域可通过相应的 FB 输入端删除。

输入端参数

14773

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
RESET_RETAIN	BOOL	TRUE: 删除非易失保留内存 FALSE: 内存内容无更改
RESET_OHC	BOOL	TRUE: 删除非易失 OHC 内存 FALSE: 内存内容无更改

输出端参数

14774

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

RESULT 的可能结果：

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

MEMCPY

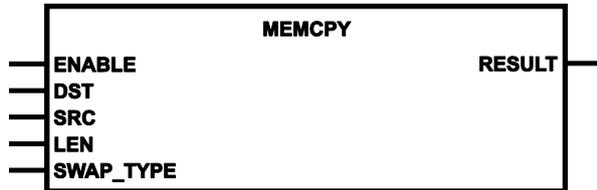
8160

= 内存复制

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

412

MEMCPY 有助于直接写入和读取内存中各种类型的数据。

FB 将 SRC 地址内容写入地址 DST。

► 就地址而言，SRC 和 DST 适用：

❗ 通过运算符 ADR 确定地址并将其分配至 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

8162

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
DST	DWORD	目标地址 ! 通过运算符 ADR 确定地址并将其分配至 FB !
SRC	DWORD	源地址
LEN	WORD	待传送数据字节数量 (≥ 1)
SWAP_TYPE	BYTE	交换字节顺序 0 = 无交换 例如: 1A 2B 3C 4D ⇔ 1A 2B 3C 4D 1 = 交换 2 个字节 (WORD、INT ...) 例如: 1A 2B 3C 4D ⇔ 2B 1A 4D 3C ! LEN 必须为 2 的倍数! 2 = 交换 4 个字节 (DWORD、DINT、REAL、TIME ...) 例如: 1A 2B 3C 4D ⇔ 4D 3C 2B 1A ! LEN 必须为 4 的倍数!

输出端参数

8163

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

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
152	98	不允许的内存区域： <ul style="list-style-type: none"> • 无效的源地址 • 无效的目标地址 • 无效的字节数量
156	9C	不允许的值： <ul style="list-style-type: none"> • 针对 SWAP_TYPE 的无效值 • LEN 不匹配 SWAP_TYPE

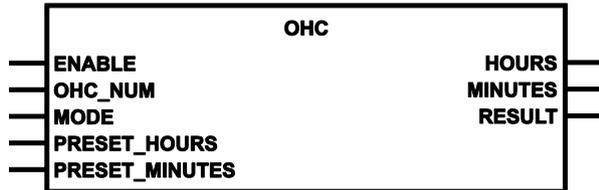
OHC

= 工作时间计数器

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

OHC 提供 4 个通用的工作时间计数器。

但如果硬件版本 < AD：仅可使用 2 个工作时间计数器。

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

❗ 如果装置硬件版本 < AD：

复位针对 OHC 的内存区域一次：

▶ 在 FB 中 **MEM_ERROR** (→ 页 [212](#))，设定输入端 RESET_OHC = TRUE！

> 仅现在可使用工作时间计数器。

输入端参数

14779

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

输出端参数

14780

参数	数据类型	说明
HOURS	DWORD	小时值计数 (0..4 294 967 295 = 0x0000 0000...0xFFFF FFFF)
MINUTES	BYTE	分钟值计数 (0..59 = 0x00...0x3B)
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

值 十进制 十六进制		说明
0	00	FB 已禁用
1	01	FB 执行完成且无错误 – 数据有效
130	82	OHC_NUM 中的计数器数字无效
131	83	预设值无效
132	84	模式设定无效
158	9E	剩余内存无效 (CRC 错误)

SET_IDENTITY

8174

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

8535

SET_IDENTITY 设定应用程序特定程序 ID。

应用程序可利用该 FB 创建程序 ID。

- ▶ 可读取该 ID 以识别加载的程序：
 - 通过软件“维护工具”
 - 通过 FB 在应用程序中 **GET_IDENTITY** (→ 页 207)

输入端参数

8175

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
APP_IDENT	STRING(80)	应用程序 ID 作为最多 80 个字符的字符串，例如：“Crane1704” 通过 APP_IDENT = "" 复位

输出端参数

8176

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

RESULT 的可能结果：

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

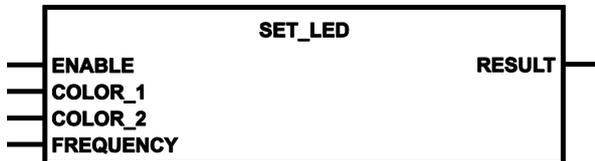
SET_LED

8052

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

8054

可通过 SET_LED 在应用程序中更改状态 LED 的频率和颜色。

! 如果在应用程序中更改闪烁模式，则默认设定表不再有效 (→ 章节 **状态 LED** (→ 页 17)) .

输入端参数

8223

参数	数据类型	说明
ENABLE	BOOL	TRUE: 执行该功能元件 FALSE: 单元未执行 > 功能块输入端未启用 > 功能块输出端未指定
COLOR_1	BYTE	“开启”的 LED 颜色 来自数据结构 "System LED Color" 的颜色常量；允许： 00 = LED_BLACK (= LED 熄灭) 02 = LED_GREEN
COLOR_2	BYTE	“关闭”的 LED 颜色 来自数据结构 "System LED Color" 的颜色常量；允许： 00 = LED_BLACK (= LED 熄灭) 02 = LED_GREEN
FREQUENCY	BYTE	LED 闪烁频率 来自数据结构 "System LED Frequency" 的频率常量；允许： : 00 = LED_0HZ = 永久亮起 01 = LED_05HZ = 以 0.5 Hz 的频率闪烁 02 = LED_1HZ = 以 1 Hz 的频率闪烁 04 = LED_2HZ = 以 2 Hz 的频率闪烁 10 = LED_5HZ = 以 5 Hz 的频率闪烁

输出端参数

8227

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

RESULT 的可能结果：

值		说明
十进制	十六进制	
0	00	FB 已禁用
1	01	功能块执行完成且无错误
2	02	功能块已启用 (操作还未完成)
133	85	FREQUENCY 的值无效
151	97	颜色值无效

SET_PASSWORD

8178

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

8179

SET_PASSWORD 设定通过维护工具进行程序和内存上传的用户密码。

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

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

! 加载新的应用程序时，密码复位。

输入端参数

8180

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

输出端参数

8181

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

RESULT 的可能结果：

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

TIMER_READ_US

8219

单元类型 = 功能块 (FB)

元件类型 ifm_CR0452_Vxxyzz.LIB

CODESYS 中的符号:



说明

660

TIMER_READ_US 读取当前系统时间，以 [μs] 为单位

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

信息

系统计时器的计数值可高达 4 294 967 295 μs，而后会再次从 0 开始计数。

4 294 967 295 μs = 1h 11min 34s 967ms 295μs

输出端参数

8220

参数	数据类型	说明
TIME_US	DWORD	当前系统时间 [μs]
RESULT	BYTE	功能块反馈 (可能的消息 → 下表)

RESULT 的可能结果：

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

5.2.6 功能元件：图形

内容	
功能元件：图形帮助.....	225
功能元件：图形可视化.....	233

15294

功能元件：图形帮助

内容	
GET_TEXT_FROM_FLASH.....	226
NORM_DINT	228
NORM_REAL	230
TOGGLE.....	232

15296

您将在此了解可在图形界面编程时提供支持的更多 IFM 功能元件。

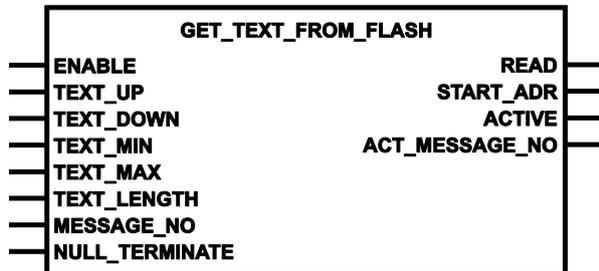
GET_TEXT_FROM_FLASH

3196

单元类型 = 功能块 (FB)

元件类型 ifm_PDMsmart_UTIL_Vxxyyzz.Lib

CODESYS 中的符号:



说明

11651

GET_TEXT_FROM_FLASH controls **FLASH_READ** (→ 页 201) , 以直接读取类型 STRING 的文本。

跟 PDM360 和 PDM360compact 相反, PDM360smart 无文件系统。因此, 建议利用闪存或 FLASH 内存存储文本消息。若要读取这些内存区域, 则需 FLASH_READ。

为确保读取一个或多个文本, 必须计算内存中文本的起始地址。该计算以及 ENABLE 输入端的设定/复位在 GET_TEXT_FROM_FLASH 中进行。

内存中的文本必须根据以下规则进行管理:

文本长度

所有文本的文本长度必须相同, 且最多 30 个字符, 因为装置的显示屏尺寸有限。

文本创建

应利用电子表格程序 (如 Excel) 创建文本并以 CSV 格式保存。该 CSV 文件可通过 **IFM** 维护工具直接加载至所需内存区域。

→ **ecomatmobile** DVD“软件、工具和文档”。

编程系统自动通过 NULL 类型终止 STRING。因此, 包含 30 个字符的文本在内存中使用 31 个字节。FB 在计算时考虑这一点。

从所示装置闪存 (→ 章节 **可用内存** (→ 页 14)) 扣除标题的 128 个字节 因此, 如果文本长度为 30 个字符, 则闪存可保存 $65\,408 / 31 = 2\,109$ 个文本。

输入端参数

3302

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
TEXT_UP	BOOL	边沿 FALSE → TRUE : 读取下一个文本
TEXT_DOWN	BOOL	边沿 FALSE → TRUE : 读取上一个文本
TEXT_MIN	WORD	MESSAGE_NO 下限
TEXT_MAX	WORD	MESSAGE_NO 上限
TEXT_LENGTH	BYTE	文本长度
MESSAGE_NO	WORD	文本数量
NULL_TERMINATE	BOOL	TRUE: 字符串以 0 结尾 FALSE: 字符串不以 0 结尾

输出端参数

15596

参数	数据类型	说明
READ	BOOL	读取命令 ▶ 将该信号设定至 FB FLASH_READ 的输入端 ENABLE !
START_ADR	WORD	计算的起始地址 ▶ 将该信号设定至 FB FLASH_READ 的输入端 SCR !
ACTIV	BOOL	TRUE: FB is active (if input ENABLE = 1)
ACT_MESSAGE_NO	WORD	当前文本数量

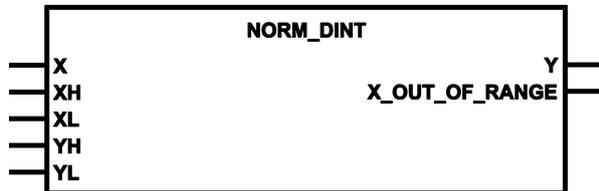
NORM_DINT

13240

单元类型 = 功能块 (FB)

元件类型 ifm_PDMSmart_UTIL_Vxxyyzz.LIB

CODESYS 中的符号:



说明

3307

NORM_DINT 将定义的限值范围内的值正常化为新的限值范围内的值。

FB 将类型 DINT (即 XH 和 XL 限值范围内) 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。
输出 X_OUT_OF_RANGE 在该值范围之外设定。
- ▶ 计算 $(XH-XL) \cdot (YH-YL)$ 的结果必须保持在数据类型 DINT 的值范围中 (-2 147 483 648...2 147 483 647)!
- > 因为舍入误差, 正常化值的偏差可能为 1。
- > 如果按照倒数的形式定义限值 (XH/XL 或 YH/YL), 则正常化亦倒转。

输入端参数

3308

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

输出端参数

3309

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

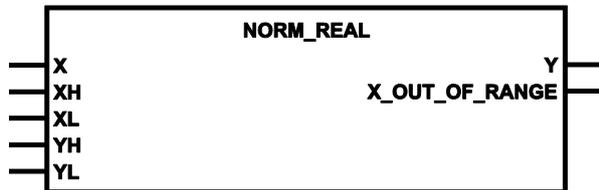
NORM_REAL

13244

单元类型 = 功能块 (FB)

元件类型 ifm_PDsmart_UTIL_Vxyyzz.LIB

CODESYS 中的符号:



说明

3310

NORM_REAL 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 将类型 REAL (即 XH 和 XL 限值范围内) 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。
输出 X_OUT_OF_RANGE 在该值范围之外设定。
- ▶ 计算 $(XH-XL) \cdot (YH-YL)$ 的结果必须保持在数据类型 REAL 的值范围中 $(-3,402823466 \cdot 10^{38} \dots 3,402823466 \cdot 10^{38})!$
- > 因为舍入误差, 正常化值的偏差可能为 1。
- > 如果按照倒数的形式定义限值 (XH/XL 或 YH/YL), 则正常化亦倒转。

输入端参数

3311

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

输出端参数

3312

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

TOGGLE

13248

单元类型 = 功能块 (FB)

元件类型 ifm_PDMsmart_UTIL_Vxxyyzz.LIB

CODESYS 中的符号:



说明

3304

TOGGLE 仅通过一个输入位启用布尔变量的设定和复位。

输入端 IN 的第一个上升沿将输出端 OUT 设为 'TRUE'。

下一个上升沿将输出端复位为 'FALSE'。

等等。

输入端参数

3305

参数	数据类型	说明
IN	BOOL	边沿 FALSE → TRUE : 设定/复位输出端

输出端参数

3306

参数	数据类型	说明
OUT	BOOL	IN 上的第 1 个边沿 ⇒ TRUE IN 上的第 2 个边沿 ⇒ FALSE IN 上的第 3 个边沿 ⇒ TRUE ...

功能元件：图形可视化

内容

BASICDISPLAY_INIT.....	234
PDM_PAGECONTROL.....	235

15298

您可在此了解针对以下目的的 **IFM** 功能元件：

- 初始化装置屏幕
- 显示可视化页面

BASICDISPLAY_INIT

9310

组件类型 = 类型 INT 功 (FUN) BOOL

元件类型 ifm_CRRnnnn_Init_Vxxyzz.LIB

CODESYS 中的符号:



说明

9312

在第一个 PLC 周期中，功能 BASICDISPLAY_INIT 初始化 BasicDisplay 的屏幕。

若无此初始化操作，则屏幕保持暗色。

功能无需参数设定。

- ▶ 仅在第一个 PLC 周期中调用功能！
- 而后跳过调用。
- 下方示例

例如：BasicDisplay_Init

9314

<ul style="list-style-type: none"> ▶ 生成程序 (PRG) INIT_DISPLAY。 ▶ 调用功能 (FUN) BASICDISPLAY_INIT 以便仅在第一个 PLC 周期执行。 	
<ul style="list-style-type: none"> ▶ 调用程序 PLC_PRG 中的程序 INIT_DISPLAY (连同其他初始化，如有)。 	

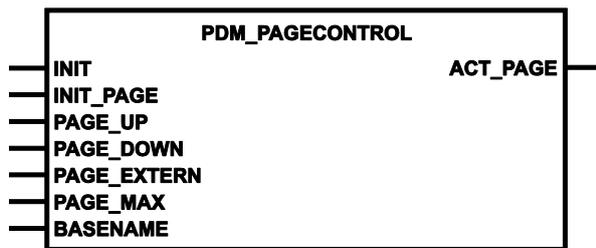
PDM_PAGECONTROL

3186

装置类型 = 程序 (PRG)

元件类型	可用于以下装置：
ifm_PDM_UTIL_Vxxyyzz.LIB	<ul style="list-style-type: none"> • PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
ifm_PDMng_UTIL_Vxxyyzz.LIB	<ul style="list-style-type: none"> • PDM360 NG: CR108n, CR120n
ifm_PDMsmart_UTIL_Vxxyyzz.LIB	<ul style="list-style-type: none"> • BasicDisplay: CR045n • PDM360smart: CR1070, CR1071

CODESYS 中的符号:



说明

3294

PDM_PAGECONTROL 控制特定可视化页面的打开。在 CoDeSys 中，通过系统变量 CurrentVisu (字符串类型 [40]) 打开可视化页面和提供反馈。

还可通过该程序打开所选可视化页面或逐个滚动可视化页面。

如果所有可视化名称对应相同的模式，即基本名 + 5 位数字的组合 (库版本 V04.00.07 或更高；之前：3 位*)，则可确保程序的最佳使用。

BASENAME 示例 = PAGE：

可视化名称 = PAGE00001、PAGE00002、PAGE00003 等等。

就基本名 1...35 而言，可使用大写字母 (无特殊字符)。可视化页面必须连续编号。程序通过参数 BASENAME 和编号创建最终可视化名称，或从当前可视化名称读取编号并在输出端参数 ACT_PAGE 提供。

每个可视化页面还可单独命名，而不是利用基本名和连续编号命名，如 SERVICE1、MOTORDATA2、CONFIGURATION3。但在此情况下，编程更复杂，因为基本名和可视化编号必须单独分配。而后逐个滚动非常受限。

 将字母 P 作为 BASENAME，而后您的程序可与 IFM 模板兼容。

*)  另外，在为现有可视化页面命名时还需注意新的 5 位编号！

输入端参数

3293

参数	数据类型	说明
INIT	BOOL	TRUE (仅针对 1 个周期) : 显示屏通过 INIT_PAGE 所示的初始化予以初始化。 FALSE: 在程序进一步处理期间
INIT_PAGE	WORD	通过 INIT 调用的可视化编号
PAGE_UP	BOOL	边沿 FALSE → TRUE : 增加可视化编号的值
PAGE_DOWN	BOOL	边沿 FALSE → TRUE : 减少可视化编号的值
PAGE_EXTERN	WORD	所示可视化页面直接打开 (与 PAGE_UP / PAGE_DOWN 无关)。 如果 PAGE_EXTERN = ACT_PAGE, 则 PAGE_EXTERN 复位为 "0" !
PAGE_MAX	WORD	可选可视化页面最大数
BASENAME	STRING [35]	可视化页面名称常见部分 可视化页面按其名称编号: 例如: P00001. 以下几点将适用: • "P" = BASENAME (仅限大写字母!) • "00001" = 可视化编号 (5 位!)

输出端参数

3295

参数	数据类型	说明
ACT_PAGE	WORD	当前可视化编号

6 诊断和错误处理

内容

诊断	237
故障	237
响应系统错误	238
CAN / CANopen: 错误和错误处理	238

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 [27](#))

此时尽可能地实现高工作可靠性。

6.1 诊断

19601

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

还可监控输入端和输入端是否正常运行，具体视装置而定。

- 断线，
- 短路，
- 值在范围之外。

就诊断而言，可使用装置“正常”运行期间创建的配置和日志数据。

初始化和启动期间可监控系统组件是否正确启动。日志文件记录错误。

就进一步诊断而言，还可进行自检。

6.2 故障

19602

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

故障通常是项目本身失灵的结果，也可能在之前没有失灵的情况下存在。

在 →ISO 13849-1 中，“故障”指“随机故障”。

6.3 响应系统错误

8504

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

- > 只要不再显示错误导致状态，系统即会复位错误消息。

6.3.1 响应错误消息的进程示例

8505

运行时系统循环写入系统标志 TEMPERATURE。

应用程序通过检索 INT 变量来检测装置温度。

如果超过或未达到应用程序的允许值：

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

6.4 CAN / CANopen: 错误和错误处理

19604

→ 系统手册“ecomatmobile 技巧”

→ 章节 **CAN / CANopen : 错误和错误处理**

7 附录

内容

系统标志	239
错误表	241

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

15309



如果扩展 PLC 配置，则系统标志的地址可更改。

- ▶ 编程时仅可使用系统标志的符号名称！

系统标志 (符号名称)	类型	说明
KEY_F1	BOOL	功能键 1
KEY_F2	BOOL	功能键 2
KEY_F3	BOOL	功能键 3
KEY_F4	BOOL	功能键 4
KEY_UP	BOOL	导航键 [▲]
KEY_DOWN	BOOL	导航键 [▼]
KEY_LEFT	BOOL	导航键 [◀]
KEY_RIGHT	BOOL	导航键 [▶]
KEY_OK	BOOL	导航键 [OK]
KEY_ESC	BOOL	导航键 [ESC]
KEY_CHANGED	BOOL	脉冲：按键状态更改 (按钮已按或松开)
KEY_PRESSED	BOOL	按钮已按
KEY_BACKLIGHT	BYTE	按键背景照明 (0...100 %) 预设 = 100 %
RT_F1	BOOL	脉冲：开启功能键 1
RT_F2	BOOL	脉冲：开启功能键 2
RT_F3	BOOL	脉冲：开启功能键 3
RT_F4	BOOL	脉冲：开启功能键 4
RT_UP	BOOL	脉冲：开启导航键 [▲]

系统标志 (符号名称)	类型	说明
RT_DOWN	BOOL	脉冲: 开启导航键 [▼]
RT_LEFT	BOOL	脉冲: 开启导航键 [◀]
RT_RIGHT	BOOL	脉冲: 开启导航键 [▶]
RT_OK	BOOL	脉冲: 开启导航键 [OK]
FT_F1	BOOL	脉冲: 松开功能键 1
FT_F2	BOOL	脉冲: 松开功能键 2
FT_F3	BOOL	脉冲: 松开功能键 3
FT_F4	BOOL	脉冲: 松开功能键 4
FT_UP	BOOL	脉冲: 松开导航键 [▲]
FT_DOWN	BOOL	脉冲: 松开导航键 [▼]
FT_LEFT	BOOL	脉冲: 松开导航键 [◀]
FT_RIGHT	BOOL	脉冲: 松开导航键 [▶]
FT_OK	BOOL	脉冲: 松开导航键 [OK]
SCREEN_BACKLIGHT	BYTE	LCD 背景照明 (0...100 %) 预设 = 100 %
SUPPLY_VOLTAGE_VBBS	WORD	VBBs 上的电源电压, 以 [mV] 为单位
SUPPLY_VOLTAGE_VU	WORD	内部电源电压, 以 [mV] 为单位
TEMPERATURE	INT	装置中的温度, 以 [°C] 为单位

7.2 错误表

内容	
错误标志	241
错误：CAN / CANopen	241

19606

7.2.1 错误标志

19608

→ 章节 **系统标志** (→ 页 [239](#))

7.2.2 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：系统

8413

EMCY 代码 对象 0x1003		对象 0x1001	制造商特定信息					
第 0 个字节 [hex]	第 1 个字节 [hex]	第 2 个字节 [hex]	第 3 个字节	第 4 个字节	第 5 个字节	第 6 个字节	第 7 个字节	说明
00	31	05						电源电压
00	42	09						过温

8 专业术语

B

Bootloader

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

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

Bootloader 包含基本例程...

- 针对硬件模块之间的通信，
- 针对操作系统的重新加载。

Bootloader 是要保存在装置上的第一个软件模块。

C

CAN

CAN = **控制器局域网**

CAN 是针对大数据量的优先级控制现场总线系统。有几个基于 CAN 的较高级协议，如 'CANopen' 或 'J1939'。

CAN 堆栈

CAN 堆栈 = 负责处理 CAN 消息的软件组件。

CiA

CiA = CAN in Automation e.V.

德国/埃朗根的用户和制造商组织。CAN 和 CAN 网络协议的定义和控制体。

主页 → www.can-cia.org

CiA DS 304

DS = **标准草案**

针对安全通信的 CANopen 装置行规

CiA DS 401

DS = **标准草案**

针对二进制和模拟 I/O 模块的 CANopen 装置行规

CiA DS 402

DS = **标准草案**

针对驱动器的 CANopen 装置行规

CiA DS 403

DS = **标准草案**

针对 HMI 的 CANopen 装置行规

CiA DS 404

DS = **标准草案**

针对测量和控制技术的 CANopen 装置行规

CiA DS 405

DS = **标准草案**

可编程控制器 (IEC 61131-3) 接口 CANopen 规范

CiA DS 406

DS = **标准草案**

针对编码器的 CANopen 装置行规

CiA DS 407

DS = **标准草案**

针对本地公共交通的 CANopen 应用行规

COB ID

COB = **通信对象**

ID = **标识符**

CANopen 通信对象 ID

相当于 CAN 消息的标识符，通过 CAN 总线铜通信项目一起发送。

CODESYS

CODESYS® 是德国 3S (即 Smart Software Solutions GmbH) 的注册商标。

“自动化联盟 CODESYS”将自动化行业利用广泛使用的 IEC 61131-3 开发工具 CODESYS® 对其所有硬件装置进行编程的公司联合起来。

主页 → www.codesys.com

CSV 文件

CSV = **逗号分隔值** (又称 : **字符分隔值**)

CSV 文件是存储或交换简单结构数据的文本文件。

文件扩展名为 .csv。

例如 : 包含数值的来源表 :

值 1.0	值 1.1	值 1.2	值 1.3
值 2.0	值 2.1	值 2.2	值 2.3
值 3.0	值 3.1	值 3.2	值 3.3

因此 CSV 文件如下 :

值 1.0 ; 值 1.1 ; 值 1.2 ; 值 1.3

值 2.0 ; 值 2.1 ; 值 2.2 ; 值 2.3

值 3.0 ; 值 3.1 ; 值 3.2 ; 值 3.3

D

DC

直流电

DLC

数据长度码 = CANopen 消息数据字节的数量。

针对 →SDO : DLC = 8

DRAM

DRAM = **动态随机存取内存**。

随机存取电子内存 (RAM)模块技术 内存元件是可充电或放电的电容器。 可通过开关晶体管访问, 可读取或由新内容覆盖。 内存内容易失 : 如果工作电压缺失或重启太迟, 则存储的信息丢失。

DTC

DTC = **诊断故障码** = 错误代码

在协议 J1939 中, 故障和错误有效管理, 且可通过指定的编号 (DTC) 报告。

E

ECU

(1) **电子控制器** = 控制装置或微控制器

(2) **发动机控制器** = 发送机控制装置

EDS 文件

EDS = **电子数据表**，例如针对以下方面：

- CANopen 主站对象目录文件，
- CANopen 装置说明。

装置和程序可通过 EDS 交换其规范并简单加以考虑。

EMC

EMC = **电磁兼容性**。

根据关于电磁兼容性的 EC 指令 (2004/108/EEC) (简称为“EMC 指令”)，电气电子装置、设备、系统或组件需满足一定要求，方可在现有电磁环境下正常运行。装置不得干扰其环境，且不可受到外部电磁干扰的不利影响。

EMCY

紧急情况的缩写

CANopen 协议中的消息，错误通过消息发送。

Ethernet

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

EUC

EUC = **受控设备**。

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

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

F

FiFo

FIFO (**先入先出**) = 堆栈内存的工作原理：先写入堆栈内存的数据包也将先读取。每个标识符均有这样的一个缓冲区 (队列)。

FRAM

FRAM, 又称 FeRAM, 指**铁电随机存取内存**。存储和清除操作可通过铁电层的极化变化进行。

FRAM 相比传统只读内存的优势:

- 非易失性,
- 可兼容常见的 EEPROM, 但是:
- 存取时间大约为 100 ns,
- 存取周期几乎无限制。

H

HMI

HMI = **人机界面**

I

ID

ID = **标识符**

区分连接至系统的装置/参与者或在参与者之间传送的消息报的名称。

IEC 61131

标准: **编程逻辑控制器基础知识**

- 第 1 部分: 一般信息
- 第 2 部分: 生产设备要求和测试
- 第 3 部分: 编程语言
- 第 5 部分: 通信
- 第 7 部分: 模拟控制编程

IEC 用户周期

IEC 用户周期 = CODESYS 应用程序中的 PLC 周期。

IP 地址

IP = **互联网协议**。

IP 地址是有助于清楚识别互联网参与者的编号。为清晰起见, 按照 4 个十进制值写入编号, 如 127.215.205.156。

ISO 11898

标准：道路车辆 - 控制器局域网

- 第 1 部分：数据链路层和物理信号发送
- 第 2 部分：高速媒介访问单元
- 第 3 部分：低速容错的媒介依赖型接口
- 第 4 部分：时间触发通信
- 第 5 部分：低功率模式高速媒介访问单元

ISO 11992

标准：拖吊和被拖吊车辆之间电气连接数字信息的交换

- 第 1 部分：物体和数据链路层
- 第 2 部分：针对制动器和驱动装置的应用层
- 第 3 部分：针对除制动器和驱动装置之外的设备的应用层
- 第 4 部分：诊断

ISO 16845

标准：道路车辆 - 控制器局域网 (CAN) - 符合性测试计划

J

J1939

→ SAE J1939

L

LED

LED = **发光二极管**。

发光二极管，体积小、功耗可忽略不计的高彩色亮度电子元件。

LSB

最低有效位/字节

M

MAC-ID

MAC = **制造商地址代码**

= 制造商序列号。

→ID = **标识符**

每个网卡均有 MAC 地址，即清楚定义的全局唯一数字码，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 [247](#))

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。

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

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。

9 索引

B

BASICDISPLAY_INIT	234
Bootloader	20, 243

C

CAN	243
接口和协议	18
CAN / CANopen	
错误和错误处理	238
CAN 声明 (例如 CR1080)	46
CAN 堆栈	243
CAN 接口	18
CAN_ENABLE	62
CAN_RECOVER	64
CAN_REMOTE_REQUEST	93
CAN_REMOTE_RESPONSE	95
CAN_RX	70
CAN_RX_ENH	72
CAN_RX_ENH_FIFO	75
CAN_RX_RANGE	78
CAN_RX_RANGE_FIFO	80
CAN_SETDOWNLOADID	65
CAN_STATUS	67
CAN_TX	84
CAN_TX_ENH	86
CAN_TX_ENH_CYCLIC	89
CANOPEN_ENABLE	99
CANOPEN_GETBUFFERFLAGS	101
CANOPEN_GETEMCYMESSAGES	148
CANOPEN_GETERRORREGISTER	150
CANOPEN_GETGUARDBERRLIST	143
CANOPEN_GETGUARDBHSTATSLV	145
CANOPEN_GETNMTSTATESLAVE	110
CANOPEN_GETODCHANGEDFLAG	116
CANOPEN_GETSTATE	103
CANOPEN_GETSYNCSTATE	138
CANOPEN_NMTSERVICES	112
CANOPEN_READOBJECTDICT	118
CANOPEN_SDOREAD	123
CANOPEN_SDOREADBLOCK	125
CANOPEN_SDOREADMULTI	128
CANOPEN_SDOWRITE	130
CANOPEN_SDOWRITEBLOCK	132
CANOPEN_SDOWRITEMULTI	135
CANOPEN_SENDEMCMYMESSAGE	152
CANOPEN_SETSTATE	106
CANOPEN_SETSYNCSTATE	140
CANOPEN_WRITEOBJECTDICT	120
CiA	243
CiA DS 304	243
CiA DS 401	243
CiA DS 402	243

CiA DS 403	244
CiA DS 404	244
CiA DS 405	244
CiA DS 406	244
CiA DS 407	244
COB ID	244
CODESYS	244
CODESYS 中的 FB、FUN、PRG	23
CODESYS 可视化元件	33
CODESYS 项目的编程说明	22
CR0452 彩色显示屏	15
CR0452 操作元件	16
CSV 文件	245

D

DC	245
DLC	245
DRAM	245
DTC	245

E

ECU	245
EDS 文件	246
EMC	246
EMCY	246
EMCY 代码：CANx	241
EMCY 代码：系统	242
ERROR 状态	28
Ethernet	246
EUC	246

F

FATAL ERROR 状态	29
FiFo	246
FLASH_INFO	199
FLASH_READ	201
FLASH-Speicher	14
FRAM	14, 247

G

GET_APP_INFO	203
GET_HW_INFO	205
GET_IDENTITY	207
GET_SW_INFO	208
GET_SW_VERSION	210
GET_TEXT_FROM_FLASH	226

H

HMI	247
-----------	-----

I

ID	247
IEC 61131	247
IEC 用户周期	247
ifm weltweit • ifm worldwide • ifm à l'échelle internationale	269
IFM 功能元件	52
INIT 状态 (复位)	27
IP 地址	247
ISO 11898	248
ISO 11992	248
ISO 16845	248

J

J1939	248
J1939_DM1RX	189
J1939_DM1TX	192
J1939_DM1TX_CFG	195
J1939_DM3TX	197
J1939_ENABLE	156
J1939_GETDABYNAME	158
J1939_NAME	161
J1939_RX	173
J1939_RX_FIFO	175
J1939_RX_MULTI	177
J1939_SPEC_REQ	168
J1939_SPEC_REQ_MULTI	170
J1939_STATUS	165
J1939_TX	180
J1939_TX_ENH	182
J1939_TX_ENH_CYCLIC	184
J1939_TX_ENH_MULTI	186

L

LED	248
Libraries	
required	52
required for network variables	52
LSB	248

M

MAC-ID	248
MEM_ERROR	212
MEMCPY	214
MMI	249
MRAM	249
MSB	249

N

NMT	249
NORM_DINT	228
NORM_REAL	230

O

Obj 对象	249
OBV	249
OHC	217
OPC	249
Operating hours counter	217

P

PC 卡	249
PCMCIA 卡	250
PDM	250
PDM_PAGECONTROL	235
PDO	250
PDU	250
PES	250
PGN	250
PID 控制器	250
PLC 配置	44, 251
PWM	251

R

RAW-CAN	251
ro	251
RTC	251
RUN 状态	28
rw	251

S

SAE J1939	155, 252
SD 卡	252
SDO	252
SET_IDENTITY	219
SET_LED	220
SET_PASSWORD	222
SRAM	14
STOP 状态	27

T

TCP	252
TIMER_READ_US	224
TOGGLE	232

U

UDP	253
-----------	-----

一划

一般功能配置	49
--------------	----

三划

工作状态	27
已停止	253

四划

元件的移动	36
比率式	253
内存, 可用	14
手动设定编程系统	43
从站	253
文本	35

五划

功能元件: CANopen	98
功能元件: CANopen SDO	122
功能元件: CANopen SYNC	137
功能元件: CANopen 对象目录	115
功能元件: CANopen 网络管理	109
功能元件: CANopen 状态	98
功能元件: CANopen 保护	142
功能元件: CANopen 紧急	147
功能元件: RAW-CAN 远程	92
功能元件: RAW-CAN 状态	61
功能元件: RAW-CAN (第 2 层)	61
功能元件: SAE J1939	155
功能元件: SAE J1939 状态	155
功能元件: SAE J1939 诊断	188
功能元件: SAE J1939 请求	167
功能元件: 传送 RAW-CAN 数据	83
功能元件: 传送 SAE J1939	179
功能元件: 系统	198
功能元件: 图形	225
功能元件: 图形可视化	233
功能元件: 图形帮助	225
功能元件: 接收 RAW-CAN 数据	69
功能元件: 接收 SAE J1939	172
节点	254
节点保护	254
本文档的结构是怎样的?	8
本装置 CAN J1939 的限制	38
本装置 CAN 的限制	37
本装置 CANopen 的限制	37

可用内存	14
可视化极限	31
可视化限制	32
目标	254
电子狗	253
电子狗状况	30
外壳后面板上的连接	16
主站	253
闪存	254
对象目录	253

六划

地址	254
在应用程序中控制 LED	17
夹具 15	254
网络变量	51
网络变量所需的库	52
自检	255
创建应用程序	25
关于本手册	5
关于装置的信息	12
关于装置编程的重要说明	24
安全说明	9
设定目标	44
设定运行时系统	39
设定编程系统	43

七划

进程图像	256
运行	255
运行时系统	20, 255
抖动频率	255
更新运行时系统	42
连接指令	256
针对装置 CR0452 的 IFM 功能元件	59
针对装置 CR0452 的 IFM 库	52
系统变量	49, 255
系统标志	239
系统描述	12
状态 LED	17
库	21
库 ifm_CANopen_NT_Vxyzz.LIB	55

库 ifm_CR0452_Init_Vxxyzz.LIB	54
库 ifm_CR0452_Vxxyzz.LIB	53
库 ifm_J1939_NT_Vxxyzz.LIB	57
库 ifm_PDMsmart_util_Vxxyzz.LIB	54
库 ifm_RAWCan_NT_Vxxyzz.LIB	54
应用程序	20
应用程序软件	255
应用程序的分发	26
启用 PLC 配置	45
诊断	237, 255
诊断和错误处理	237
附件	12
附录	239

八划

软件说明	19
软件控制器配置	44
图标	7, 256
使用 IFM 维护工具	26
使用, 指定	256
例如: BasicDisplay_Init	234
版权	5
所需的库	52
周期时间	256
备注	264
变量	50
注释: TEST 输入端	11
注释: 序列号	11
注意周期时间!	24
波特	256

九划

指定用途	257
按键 LED 可调暗	16
故障	237
响应系统错误	238
响应错误消息的进程示例	238
重新安装运行时系统	40
重新取样/缩放图像	32
复位	27
保存保留变量	51
保留变量	50

总线	256
误用	257
说明 62, 64, 65, 67, 70, 72, 75, 78, 80, 84, 86, 89, 93, 95, 99, 101, 103, 106, 110, 112, 116, 118, 120, 123, 125, 128, 130, 132, 135, 138, 140, 143, 145, 148, 150, 152, 156, 158, 161, 165, 168, 170, 173, 175, 177, 180, 182, 184, 186, 189, 192, 195, 197, 199, 201, 203, 205, 207, 208, 210, 212, 214, 217, 219, 220, 222, 224, 226, 228, 230, 232, 234, 235, 257	
说明沿革 (CR0452)	8
架构	257
绘图区	34

十划

配置	39
请注意	9
读回保留变量	51
通过模板设定编程系统	48
预先具备的知识	10
预运行	257

十一划

接口	15
接口说明	18
控制器的启动运行状况	10
检测信号	257
检验安装	42
符号	257
符号和格式是什么意思?	7

十二划

硬件设定	13
硬件说明	13
嵌入式软件	258
剩余	258
装置的软件模块	19
装置的性能极限	30

十三划

概述: ecomatmobile 装置文档模块	6
输入端参数 .. 62, 64, 65, 67, 70, 73, 76, 78, 81, 84, 87, 90, 93, 96, 99, 101, 103, 107, 110, 113, 116, 118, 120, 123, 126, 128, 131, 133, 135, 138, 140, 143, 145, 148, 150, 153, 157, 159, 163, 165, 168, 170, 173, 175, 177, 180, 183, 185, 187, 189, 193, 195, 197, 199, 201, 203, 205, 207, 208, 210, 212, 215, 218, 219, 220, 222, 227, 229, 231, 232, 236	
输出端功能元件	60
输出端参数 .. 63, 66, 68, 71, 74, 77, 79, 82, 85, 88, 91, 94, 97, 100, 102, 104, 108, 111, 114, 117, 119, 121, 124, 127, 129, 131, 134, 136,	

139, 141, 144, 146, 149, 151, 154, 157, 160, 164, 166, 169, 171, 174,
176, 178, 181, 183, 185, 187, 190, 194, 196, 197, 200, 202, 204, 206,
207, 209, 211, 213, 216, 218, 219, 221, 223, 224, 227, 229, 231, 232,
236

错误：CAN / CANopen.....241

错误表.....241

错误标志.....241

数据类型.....258

十四划

模板.....259

需要预先具备哪些知识？.....10

10 备注







11 ifm weltweit • ifm worldwide • ifm à l'échelle internationale

Version: 2016-11-29

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.π.Ε. • 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 厶 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.