

■



知识 ecomatmobile

CODESYS® V2.3

中文

7391020\_00\_CN 2017-07-25

© ifm electronic gmbh

## 内容

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>关于本手册</b>                             | <b>5</b>  |
| 1.1      | 版权 .....                                 | 5         |
| 1.2      | 概述： ecomatmobile 装置文档模块 .....            | 6         |
| 1.3      | CODESYS 编程手册 .....                       | 7         |
| 1.4      | 本手册中描述了哪些装置？ .....                       | 7         |
| 1.5      | 符号和格式是什么意思？ .....                        | 8         |
| 1.6      | 本文档的结构是怎样的？ .....                        | 9         |
| 1.7      | 说明沿革 (App-C).....                        | 9         |
| <b>2</b> | <b>模板和演示程序</b>                           | <b>10</b> |
| 2.1      | 简介 .....                                 | 10        |
| 2.1.1    | ifm 模板是什么？ .....                         | 10        |
| 2.1.2    | ifm 演示程序是什么？ .....                       | 11        |
| 2.2      | 通过模板设置编程系统 .....                         | 12        |
| 2.2.1    | 关于 IFM 模板.....                           | 13        |
| 2.2.2    | 怎样快速、简单地设置编程系统？（例如，CR2500） .....         | 15        |
| 2.2.3    | 插入 CANopen 从站（例如：CR2500 <-- CR2011）..... | 16        |
| 2.2.4    | 为项目补充更多功能.....                           | 17        |
| 2.3      | ifm 演示程序 .....                           | 20        |
| 2.3.1    | 控制器的演示程序.....                            | 20        |
| <b>3</b> | <b>使用 CAN – 描述</b>                       | <b>23</b> |
| 3.1      | CAN 接口 .....                             | 23        |
| 3.1.1    | CAN: 接口和协议 .....                         | 23        |
| 3.2      | CAN: 交换数据 .....                          | 24        |
| 3.2.1    | 数据接收 .....                               | 24        |
| 3.2.2    | 传输数据 .....                               | 25        |
| 3.3      | 有关 CANopen 的技术详情 .....                   | 26        |
| 3.3.1    | CANopen 网络配置、状态和错误处理 .....               | 26        |
| 3.3.2    | CoDeSys 支持 CANopen .....                 | 27        |
| 3.3.3    | CANopen 主站 .....                         | 28        |
| 3.3.4    | CANopen 从站 .....                         | 51        |
| 3.3.5    | CANopen 表 .....                          | 63        |
| 3.4      | CANopen 网络变量 .....                       | 74        |
| 3.4.1    | 一般信息 .....                               | 74        |
| 3.4.2    | CANopen 网络变量的配置 .....                    | 75        |

## 内容

|          |                             |            |
|----------|-----------------------------|------------|
| 3.4.3    | 网络变量的特殊性.....               | 82         |
| 3.5      | CAN / CANopen / 网络变量摘要..... | 83         |
| 3.6      | 用于驱动工程的 CAN.....            | 84         |
| 3.6.1    | 按照 SAE J1939 的标识符.....      | 85         |
| 3.6.2    | 例如：详细报文文档.....              | 86         |
| 3.6.3    | 例如：短报文文档.....               | 87         |
| 3.7      | CAN / CANopen: 错误和错误处理..... | 89         |
| 3.7.1    | CAN 错误.....                 | 90         |
| 3.7.2    | CANopen 错误.....             | 93         |
| <b>4</b> | <b>控制输出端 – 描述</b>           | <b>103</b> |
| 4.1      | PWM 功能 – 描述.....            | 103        |
| 4.1.1    | PWM 信号处理 – 描述.....          | 104        |
| 4.1.2    | 使用 PWMi 进行液压控制.....         | 111        |
| 4.2      | 控制器 – 描述.....               | 113        |
| 4.2.1    | 自衡过程.....                   | 113        |
| 4.2.2    | 无自平衡的受控系统.....              | 114        |
| 4.2.3    | 带延迟的受控系统.....               | 114        |
| <b>5</b> | <b>使用用户闪存</b>               | <b>115</b> |
| 5.1      | 闪存 – 什么是闪存？.....            | 115        |
| 5.2      | 什么是 CSV 文件？.....            | 116        |
| 5.3      | CSV 文件和 ifm 维护工具.....       | 117        |
| 5.3.1    | CSV 文件的要求.....              | 118        |
| 5.3.2    | 使用电子数据表程序创建 CSV 文件.....     | 119        |
| 5.3.3    | 使用编辑器创建 CSV 文件.....         | 121        |
| 5.3.4    | 使用维护工具传输 CSV 文件.....        | 123        |
| 5.3.5    | 访问闪存数据：功能块.....             | 124        |
| <b>6</b> | <b>装置中的可视化</b>              | <b>125</b> |
| 6.1      | 概要.....                     | 125        |
| 6.2      | 用户界面建议.....                 | 126        |
| 6.2.1    | 用户友好的产品设计的建议.....           | 127        |
| 6.2.2    | 您是否了解未来的用户？.....            | 128        |
| 6.2.3    | 检查适用性.....                  | 129        |
| 6.2.4    | 语言障碍.....                   | 130        |
| 6.2.5    | 文化底蕴通常不可转换.....             | 132        |
| 6.2.6    | 指令和标准.....                  | 135        |
| 6.3      | 有关颜色和位图图形的基本信息.....         | 142        |
| 6.3.1    | 图像大小矢量图形 / 像素图形.....        | 143        |

| 内容        |   |
|-----------|---|
| 6.3.2     | 位图图形的颜色 ..... 145   |
| 6.3.3     | 显示哪些颜色？ ..... 145   |
| 6.4       | 有关位图图形的特殊信息 ..... 146   |
| 6.4.1     | 加色法混色 ..... 146   |
| 6.4.2     | 哪些图形适合哪些 PDM 以及必须执行哪些步骤？ ..... 147  |
| <b>7</b>  | <b>使用的文件和库概述</b> ..... <b>148</b>   |
| 7.1       | 一般概述 ..... 148  |
| 7.2       | 单个文件和库有什么用途？ ..... 150  |
| 7.2.1     | 用于运行时系统的文件 ..... 150  |
| 7.2.2     | 目标文件 ..... 150  |
| 7.2.3     | PLC 配置文件 ..... 150  |
| 7.2.4     | ifm 装置库 ..... 151   |
| 7.2.5     | ifm CANopen 库主站/从站 ..... 151  |
| 7.2.6     | CODESYS CANopen 库 ..... 152   |
| 7.2.7     | 特定 ifm 库 ..... 153  |
| <b>8</b>  | <b>专业术语</b> ..... <b>155</b>  |
| <b>9</b>  | <b>索引</b> ..... <b>173</b>  |
| <b>10</b> | <b>备注</b> ..... <b>178</b>  |
| <b>11</b> | <b>ifm weltweit • ifm worldwide • ifm à l'échelle internationale</b> ..... <b>183</b> |

# 1 关于本手册

## 内容

|                              |   |
|------------------------------|---|
| 版权 .....                     | 5 |
| 概述：ecomatmobile 装置文档模块 ..... | 6 |
| CODESYS 编程手册 .....           | 7 |
| 本手册中描述了哪些装置？ .....           | 7 |
| 符号和格式是什么意思？ .....            | 8 |
| 本文档的结构是怎样的？ .....            | 9 |
| 说明沿革 (App-C) .....           | 9 |

202

## 1.1 版权

6088

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

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

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

|                                |   |
|--------------------------------|---|
| <b>1. 技术资料</b>                 |   |
| 内容                             | 表格中的技术数据  |
| 来源                             | → <a href="http://www.ifm.com">www.ifm.com</a> > 选择所在国家/地区 > [Data sheet search] > CRnnnn > [Technical data in PDF format]  |
| <b>安装说明/操作说明</b>               |   |
| 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。<br>IFM 主页亦可提供。  |
| <b>编程手册 + 联机帮助</b>             |   |
| 内容                             | 装置软件配置和功能的说明  |
| 来源                             | → <a href="http://www.ifm.com">www.ifm.com</a>  |
| <b>系统手册“ecomatmobile 专门知识”</b> |   |
| 内容                             | 关于以下主题的专门知识： <ul style="list-style-type: none"> <li>• 概述模板和演示程序</li> <li>• CAN、CANopen</li> <li>• 控制输出端</li> <li>• 用户闪存</li> <li>• 可视化</li> <li>• 使用的文件和库的概述</li> </ul> |
| 来源                             | → <a href="http://www.ifm.com">www.ifm.com</a>  |
| <b>系统手册“IFM 控制器 ISOBUS”</b>    |   |
| 内容                             | 装置 ISOBUS 软件配置和功能的说明  |
| 来源                             | → <a href="http://www.ifm.com">www.ifm.com</a><br>因此，“CRnnnn”表示预装 ISOBUS 的装置的产品编号。  |

\*) 括号中的描述仅包含于特定装置的说明中。

## 1.3 CODESYS 编程手册

17542

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

本手册可在 **IFM** 网站免费下载：

→ [www.ifm.com](http://www.ifm.com) > [移动车辆和工程机械系统]

您还可通过以下方式获得 *ecomatmobile* 手册和在线帮助：

→ *ecomatmobile* DVD“软件、工具和文档”

## 1.4 本手册中描述了哪些装置？

14403

不同装置的技术和方法可能不同。

这些说明适用于以下装置：

- 所有 *ecomatmobile* 控制器
- PDM: CR10nn
- PCB 控制器：CS0015

## 1.5 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。

|                     |              |
|---------------------|--------------|
| <b>⚠ 警告</b>         |              |
| 可能导致死亡或严重的不可逆伤害。    |              |
| <b>⚠ 小心</b>         |              |
| 可能导致轻微的可逆伤害。        |              |
| <b>通知</b>           |              |
| 预计或可能会导致财产损失。       |              |
| ❗                   | 关于故障或干扰的重要说明 |
| ℹ                   | 其他备注         |
| ▶ ...               | 操作请求         |
| > ...               | 反应, 结果       |
| → ...               | “查看”         |
| <a href="#">abc</a> | 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   |



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

## 1.7 说明沿革 (App-C)

14337

本手册更改了哪些内容？概述

| 日期 | 主题 | 更改： |
|----|----|-----|
|    |    |     |

## 2 模板和演示程序

### 内容

|                 |    |
|-----------------|----|
| 简介 .....        | 10 |
| 通过模板设置编程系统..... | 12 |
| ifm 演示程序.....   | 20 |

13747

### 2.1 简介

11646

#### 2.1.1 ifm 模板是什么？

11647

它们是为了 CODESYS 应用程序的模板。

这些模板分别可用于所有可编程 **ecomatmobile** 装置。

文件名的结构：

ifm\_template\_CRnnnn(CAN)\_(V1)\_(V2).pro

括号中的术语具有以下含义：

|       |  |
|-------|--|
| (CAN) | CAN 协议：<br><ul style="list-style-type: none"> <li>• 第 2 层</li> <li>• CANopen 主站</li> <li>• CANopen 从站</li> </ul> |
| (V1)  | CRnnnn 装置运行时系统的版本 (Vxyyzz)   |
| (V2)  | 模板的版本 (Vnn)  |

**!** 模板中的货号必须与要编程的装置的货号完全相同！ → 装置手册，章节“有关软件的信息”

### 快速参考指南：ifm 模板

18057

下面是您如何找到 **ifm** 模板：

- ▶ 在 CODESYS 菜单 [Datei] 中 > 打开 [Neu aus Vorlage...].
- > 对话框 [Öffnen] 将显示。
- ▶ 在目录树中选择以下路径：  
 (程序盘符) > [Programme] > [ifm electronic] > [CoDeSys (Version)] > [Projects] > (aktuelle 模板 DVD) > (请求的模板)
- ▶ 使用 [Öffnen] 确认选择。

- > 新的 CODESYS 项目即创建。  
此项目包含可在所选装置上运行的项目的所有必要元件和参数设置。
- ▶ 根据应用手动调整此项目。  
必要时，集成单个 ifm 演示 (→ 章节 ifm 演示程序 (→ 页 20))。

## 2.1.2 ifm 演示程序是什么？

11648

ifm 演示程序是针对各个功能的 CODESYS 示例。

在大多数情况下，这些示例不适用于特定 ifm 装置，直到未指定任何其他信息。

文件名的结构：

(device)demo\_(V1)\_(V2).pro

括号中的术语具有以下含义：

|      |               |
|------|---------------|
| (装置) | 示例装置的货号       |
| (V1) | 演示类型          |
| (V2) | 演示程序的版本 (Vnn) |

### 快速参考指南：ifm 演示程序

18058

下面是您如何找到 ifm 演示程序：

- ▶ 在 CODESYS 菜单中打开 [Projekt] > [öffnen]。
- > 对话框 [Öffnen] 将显示。
- ▶ 在目录树中选择以下路径：  
(程序盘符) > [Programme] > [ifm electronic] > [CoDeSys (版本)] > [Projects] > (请求的演示目录) > (请求的演示项目)
- ▶ 使用 [Öffnen] 确认选择。
- > 窗口 [Objekte kopieren] 显示。
- ▶ 突出显示只包含请求的功能的元件。
- ▶ 使用 [OK] 确认选择。
- > 演示项目中突出显示的元件即插入到当前项目中。
- ▶ 调整应用的元件并例如将其添加到模块 PLC\_PRG 中。

## 2.2 通过模板设置编程系统

### 内容

|   |    |
|---|----|
| 关于 IFM 模板 .....                           | 13 |
| 怎样快速、简单地设置编程系统？（例如，CR2500） .....          | 15 |
| 插入 CANopen 从站（例如：CR2500 <-- CR2011） ..... | 16 |
| 为项目补充更多功能 .....                           | 17 |

18051

ifm 提供随时可用的模板（程序模板），以便快速、简单和完整地设置编程系统。

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。

## 2.2.1 关于 IFM 模板

### 内容

|                          |    |
|--------------------------|----|
| 通常的文件夹结构.....            | 13 |
| 模板 (C) 的文件夹中的程序和功能 ..... | 14 |

3981

通常，为每个单元提供以下模板：

- ifm\_template\_CRnnnnLayer2\_Vxxyyzz.pro  
用于带有 CAN 第 2 层的单元的操作
- ifm\_template\_CRnnnnMaster\_Vxxyyzz.pro  
用于 CANopen 主单元的操作
- ifm\_template\_CRnnnnSlave\_Vxxyyzz.pro  
用于 CANopen 从单元的操作

此处所述的模板用于：

- 版本 2.3.9.6 之后的 CODESYS
- **ecomatmobile** DVD“软件、工具和文档”上，版本 020000 之后

所有模板都具有相同的结构。

已经为 CAN 操作选择此程序模板是程序正常运行的重要基础。

### 通常的文件夹结构

3978

功能元件按以下文件夹排序：

| 文件夹                  | 描述  |
|----------------------|---|
| CAN_OPEN             | 用于控制器和 PDM，<br>作为主站或从站的 CAN 操作：<br>包含 CANopen 的 FB。                 |
| I_O_CONFIGURATION    | 用于控制器，<br>使用第 2 层或作为主站或从站的 CAN 操作：<br>用于设置输入端和输出端的运行模式参数的 FB。       |
| PDM_COM_LAYER2       | 用于控制器，<br>作为第 2 层或作为从站的 CAN 操作：<br>用于 PLC 和 PDM 之间通过第 2 层进行基础通信的 FB |
| CONTROL_CR10nn       | 用于 PDM，<br>使用第 2 层或作为主站或从站的 CAN 操作：<br>包含操作期间进行图像和键控制的 FB。          |
| PDM_DISPLAY_SETTINGS | 用于 PDM，<br>使用第 2 层或作为主站或从站的 CAN 操作：<br>包含用于调整监视器的 FB。               |

## 模板 (C) 的文件夹中的程序和功能

18048

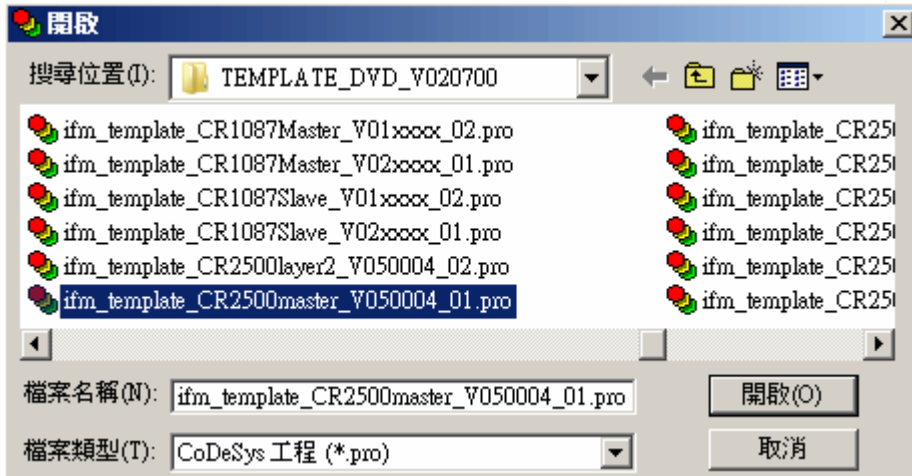
上述文件夹包含以下程序和功能块（所有 = 功能元件）：

| 文件夹中的功能元件                | 描述   |
|--------------------------|--|
| <b>CAN_OPEN</b>          |  |
| CANopen                  | 作为主站的 CAN 操作：<br>包含以下参数化功能元件：<br>• CAN1_MASTER_EMCY_HANDLER,<br>• CAN1_MASTER_STATUS,<br>• SELECT_NODESTATE (→ 关闭)。  |
| CANopen                  | 作为从站的 CAN 操作：<br>包含以下参数化功能元件：<br>• CAN1_SLAVE_EMCY_HANDLER,<br>• CAN1_SLAVE_STATUS,<br>• SELECT_NODESTATE (→ 关闭)。  |
| Objekt1xxxh              | 作为从站的 CAN 操作：<br>包含以下参数的值 [STRING]：<br>• ManufacturerDeviceName, 例如：'CR1051'<br>• ManufacturerHardwareVersion, 例如：'HW_Ver 1.0'<br>• ManufacturerSoftwareVersion, 例如：'SW_Ver 1.0' |
| <b>I_O_CONFIGURATION</b> |  |
| CONF_IO_CRnnnn           | 使用第 2 层或作为主站或从站的 CAN 操作：<br>参数化输入端和输出端的运行模式。   |
| <b>PDM_COM_LAYER2</b>    |  |
| PLC_TO_PDM               | 使用第 2 层或作为从站的 CAN 操作：<br>组织从控制器到 PDM 的通信：<br>• 监控传输时间，<br>• 传输图像更改的控制数据、输入值等。  |
| TO_PDM                   | 使用第 2 层或作为从站的 CAN 操作：<br>组织控制器和 PDM 之间的 LED 和键的信号。<br>包含以下参数化功能元件：<br>• PACK (→ 3S),<br>• PLC_TO_PDM (→ up),<br>• UNPACK (→ 3S).   |
| <b>根目录中的功能元件</b>         |  |
| PLC_CYCLE                | 使用第 2 层或作为主站或从站的 CAN 操作：<br>确定单元中 PLC 的周期时间。   |
| PLC_PRG                  | 使用第 2 层或作为主站或从站的 CAN 操作：<br>主程序 其中包含更多程序元件。  |

## 2.2.2 怎样快速、简单地设置编程系统？（例如，CR2500）

18052

- ▶ 在 CODESYS 菜单中，选择：[File] > [New from template...]
- ▶ 选择当前 DVD 的目录，例如 ...\Projects\TEMPLATE\_DVD020000.
- ▶ 在列表中找到装置的货号，例如，作为 CANopen 主站的 CR2500：



- ▶ 记住正确的程序版本！
- ▶ 如何组织 CAN 网络？  
是否想要在第 2 层的基础上工作，或者一个主站是否具有多个从站（对于 CANopen）？
- ▶ 使用 [Open] 确认选择。
- > 即会使用以下文件夹结构（左侧）生成新的 CODESYS 项目：

|                            |                               |
|----------------------------|-------------------------------|
| 作为 CANopen 主站的 CR2500 的示例： | 作为 CANopen 从站的 CR1051 的另一个示例： |
|                            |                               |

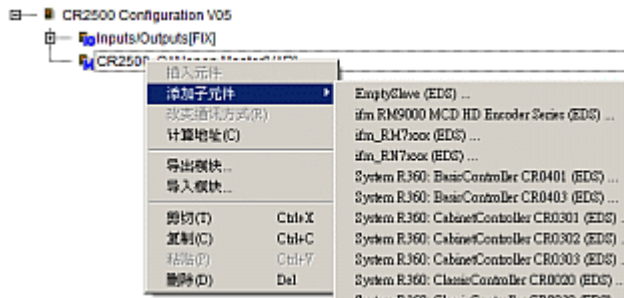
（通过模板中的文件夹结构 → 部分 **关于 IFM 模板**（→ 页 13））。

- ▶ 使用 [file] > [Save as...] 保存新项目，并定义合适的目录和项目名。

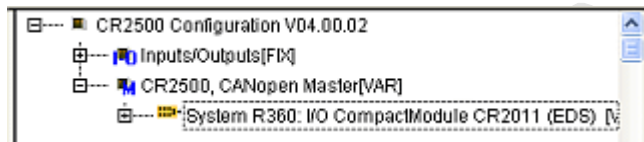
### 2.2.3 插入 CANopen 从站 (例如: CR2500 ← CR2011)

18053

- ▶ 在项目中配置 CAN 网络：  
在 CODESYS 项目中，双击制表键 [resources] 上方的元素 [PLC configuration]。
- ▶ 在条目 [CR2500, CANopen Master] 中单击鼠标右键。
- ▶ 单击上下文菜单 [Append subelement]：



- > 展开的上下文菜单中将显示所有可用 EDS 文件的列表。
- ▶ 选择请求的元件，例如，"System R360": I/O CompactModule CR2011 (EDS)".  
EDS 文件位于目录 C:\...\CoDeSys V...\Library\PLCConf\ 中。
- > 窗口 [PLC configuration] 的变化如下：



- ▶ 根据要求为输入的从站设置 CAN 参数、PDO 映射和 SDO。  
ⓘ 最好取消选择 [Create all SDOs]。
- ▶ 在有更多从站的情况下，继续执行上述步骤。
- ▶ 保存项目！

这应该是您的项目的充分描述。您是否想要使用其他元件和功能补充此项目？

→ 章节 **为项目补充更多功能** (→ 页 17)



## 2.2.4 为项目补充更多功能

3987

您已使用 ifm 模板创建一个项目，并且已经定义了 CAN 网络。现在，您想要向此项目中添加更多功能。

例如，我们将 CabinetController CR2500 当作连接了 I/O CabinetModule CR2011 和 I/O CompactModule 作为从站的 CAN open 主站：



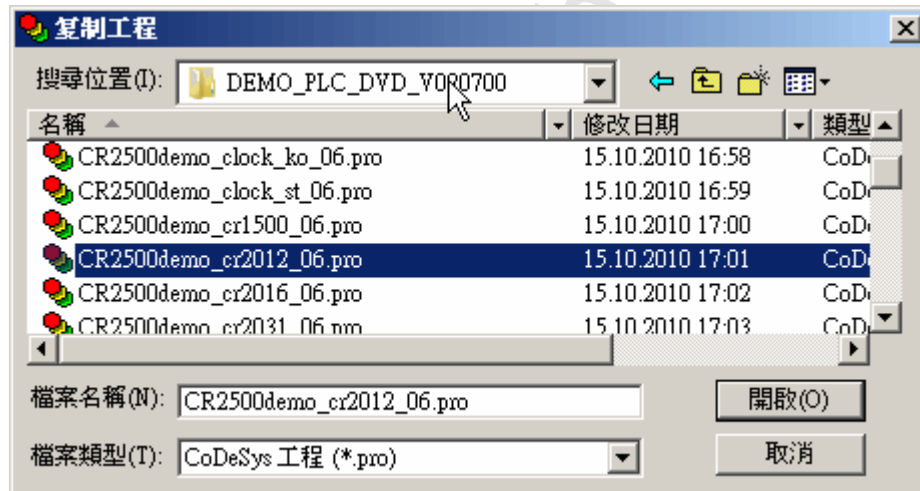
例如：PLC 配置

CR2012 上连接了一个控制杆，CR2012 又在 CR2032 上触发 PWM 输出端。如何以快速、简单的方式实现此目标？

- ▶ 保存 CODESYS 项目！
- ▶ 在 CODESYS 中，使用 [Project] > [Copy...] 打开包含请求的功能的项目：

例如，CR2500Demo\_CR2012\_02.pro

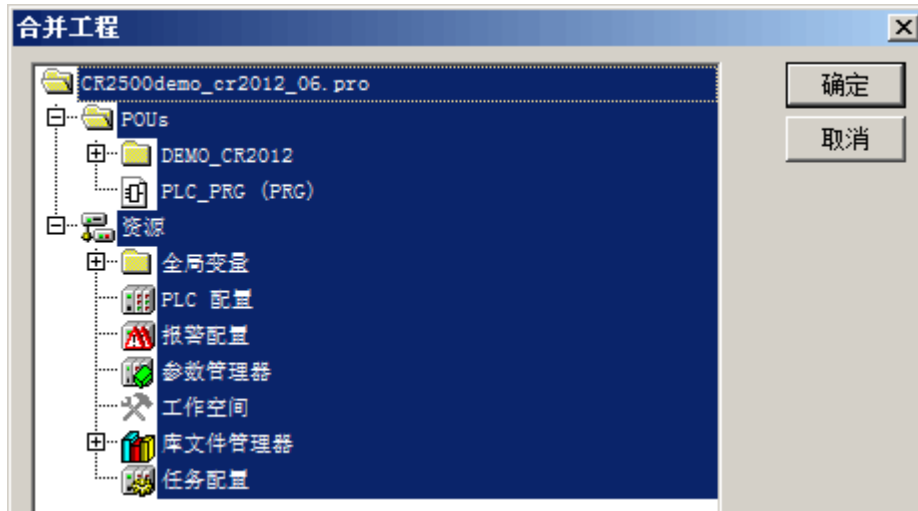
通过 C:\...\CoDeSys V\Projects\ 下的目录 DEMO\_PLC\_DVD\：



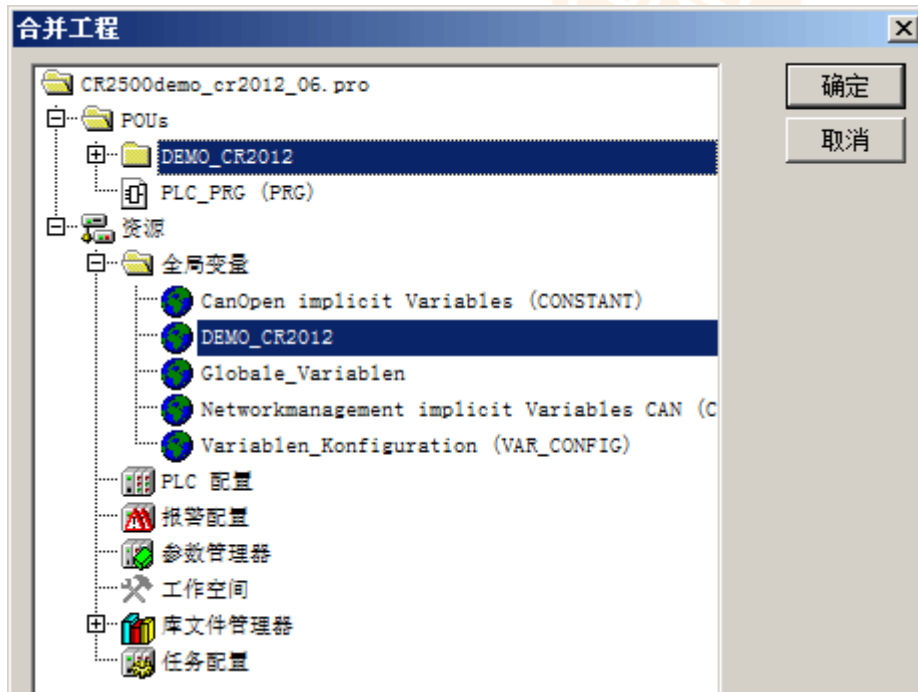
- ▶ 使用 [Open] 确认选择。

- ▶ 可忽略“Error when loading the PLC configuration”报文。

> [Copy objects] 窗口显示：



- ▶ 突出显示仅包含请求的功能的元件，例如，在本例中为：



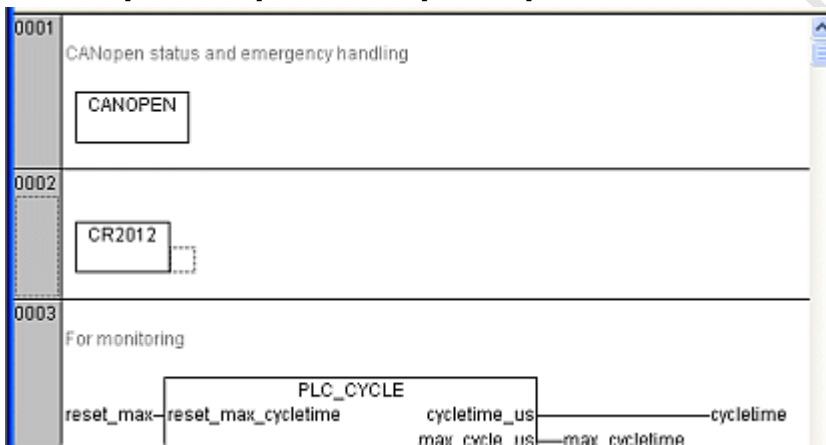
<sup>①</sup> 在其他情况下，可能需要库和/或可视化。

- ▶ 使用 [OK] 确认选择。

> 在我们的示例项目中，已添加了演示项目中选择的元件：

| POU : | 资源 : |
|-------|------|
|       |      |

▶ 在主程序 [PLC\_PRG] 中插入程序 [CR2012]，例如：



- ▶ 功能元件和全局变量的注释通常包含有关如何配置、包含或排除单个元件的信息。 必须遵守此信息。
- ▶ 根据您的条件调整输入端和输出端变量以及参数和可能的可视化。
- ▶ [Project] > [Save] 和 [Project] > [Rebuild all].
- ▶ 在进行可能需要的纠正以及添加缺少的库之后（→ 重建之后的错误报文），再次保存项目。
- ▶ 遵循这一原则逐步 (!) 从其他项目添加更多功能并检查结果。
- ▶ [Project] > [Save] 和 [Project] > [Rebuild all].

## 2.3 ifm 演示程序

### 内容

控制器的演示程序..... 20

3982

在目录

- DEMO\_PLC\_DVD... (针对控制器) 或
- DEMO\_PDM\_DVD... (针对 PDM) 中

位于 C:\...\CoDeSys V...\Projects\ 下

我们说明了测试的演示程序中的某些功能。如果需要，可以在自己的项目中实现这些功能。 ifm 演示程序的结构和变量与 ifm 模板中的结构和变量匹配。

每个演示程序都只显示一个主题。对于控制器，也会显示某些可视化，这会在 PC 屏幕上显示测试的功能。

功能元件和变量列表中的注释可帮助您根据您的项目调整演示程序。

如果没有特别注明，演示程序适用于所有控制器或所有 PDM。

此处描述的演示程序适用于：

- 版本 2.3.9.6 之后的 CODESYS
- ecomatmobile DVD“软件、工具和文档”上，版本 020000 之后

### 2.3.1 控制器的演示程序

3995

| 演示程序  | 功能   |
|---|--|
| CR2500Demo_CanTool_xx.pro   | PDM360、PDM360compact、PDM360smart 和控制器的该演示程序的功能不同：<br>包含要设置的 FB 并分析 CAN 接口。 |
| CR2500Demo_ClockFu_xx.pro<br>CR2500Demo_ClockKo_xx.pro<br>CR2500Demo_ClockSt_xx.pro | 控制器的时钟发生器，以模拟输入端上的值为函数：<br>Fu = 采用功能块图<br>K0 = 采用梯形图<br>St = 采用结构文本        |
| CR2500Demo_CR1500_xx.pro  | 连接小键盘模块 Cr1500，作为控制器 (CANopen 主站) 的从站。                                     |
| CR2500Demo_CR2012_xx.pro  | I/O 柜内用模块 CR2012，作为控制器 (CANopen 主站) 的从站，<br>连接带方向开关和参考中压的控制杆。              |

| 演示程序                                     | 功能   |
|--|--|
| CR2500Demo_CR2016_xx.pro                 | I/O 柜内用模块 CR2016, 作为控制器 (CANopen 主站) 的从站,<br>4 个频率输入端,<br>4 个数字输入端减开关,<br>4 个数字输入端加开关,<br>4 个模拟输入端比率计,<br>4 个 PWM1000 输出端和<br>12 个数字输出端。 |
| CR2500Demo_CR2031_xx.pro                 | I/O 紧凑型模块 CR2031, 作为控制器 (CANopen 主站) 的从站,<br>PWM 输出端上的当前测量值  |
| CR2500Demo_CR2032_xx.pro                 | I/O 紧凑型模块 CR2032, 作为控制器 (CANopen 主站) 的从站,<br>4 个数字输入端,<br>4 个数字输入端模拟评估,<br>4 个数字输出端,<br>4 个 PWM 输出端。                                     |
| CR2500Demo_CR2033_xx.pro                 | I/O 紧凑型模块 CR2033, 作为控制器 (CANopen 主站) 的从站,<br>4 个数字输入端,<br>4 个数字输入端模拟评估,<br>4 个数字输出端,   |
| CR2500Demo_CR2101_xx.pro                 | 倾斜传感器 CR2101, 作为控制器 (CANopen 主站) 的从站。  |
| CR2500Demo_CR2102_xx.pro                 | 倾斜传感器 CR2102, 作为控制器 (CANopen 主站) 的从站。  |
| CR2500Demo_CR2511_xx.pro                 | I/O 智能模块 CR2511, 作为控制器 (CANopen 主站) 的从站,<br>8 个 PWM 输出端电流控制。   |
| CR2500Demo_CR2512_xx.pro                 | I/O 智能模块 CR2512, 作为控制器 (CANopen 主站) 的从站,<br>8 个 PWM 输出端。<br>显示每个通道对的当前电流。  |
| CR2500Demo_CR2513_xx.pro                 | I/O 智能模块 CR2513, 作为控制器 (CANopen 主站) 的从站,<br>4 个数字输入端,<br>4 个数字输出端,<br>4 个模拟输入端 0...10 V。   |
| CR2500demo_input_from_pdm_CANopen_xx.pro | 通过 CANopen 使用系统变量 :<br>• HANDLE,<br>• INPUT_VALUE,<br>• LENGHT   |

| 演示程序  | 功能  |
|---|---|
| CR2500demo_input_from_pdm_Layer2_xx.pro                                 | 通过 CAN-Layer2 使用系统变量 :<br><ul style="list-style-type: none"> <li>• HANDLE,</li> <li>• INPUT_VALUE,</li> <li>• LENGHT</li> </ul>           |
| CR2500Demo_Interrupt_xx.pro   | 以 SET_INTERRUPT_XMS 为例。   |
| CR2500Demo_Operating_hours_xx.pro                                       | 带有 PDM 接口的工作时间计数器示例   |
| CR2500Demo_PWM_xx.pro   | 使用以下功能元件将输入端中的电位计值转换为输出端中的额定值 :<br><ul style="list-style-type: none"> <li>• INPUT_VOLTAGE,</li> <li>• NORM,</li> <li>• PWM100.</li> </ul> |
| CR2500Demo_RS232_xx.pro   | 串行接口借助 Windows 超级终端接收数据的示例。   |
| StartersetDemo.pro<br>StartersetDemo2.pro<br>StartersetDemo2_fertig.pro | 使用初学者学习箱 EC2074 进行的各种在线学习练习   |

\_xx = indication of the demo version

### 3 使用 CAN – 描述

| 内容                           |    |
|------------------------------|----|
| CAN 接口 .....                 | 23 |
| CAN: 交换数据 .....              | 24 |
| 有关 CANopen 的技术详情 .....       | 26 |
| CANopen 网络变量 .....           | 74 |
| CAN / CANopen / 网络变量摘要 ..... | 83 |
| 用于驱动工程的 CAN .....            | 84 |
| CAN / CANopen: 错误和错误处理 ..... | 89 |

13743

#### 3.1 CAN 接口

| 内容               |    |
|------------------|----|
| CAN: 接口和协议 ..... | 23 |

14101

连接和数据 → 技术资料

##### 3.1.1 CAN: 接口和协议

18060

设备配备多个 CAN 接口，具体视硬件设计而定。基本上，所有接口均可与以下功能一起使用，彼此独立：

- 第 2 层：CAN 位于第 2 级
- CANopen 主站
- CANopen 从站
- CANopen 网络变量 (通过 CODESYS)
- SAE J1939 (用于驱动器管理)
- 总线负载检测
- 错误帧计数器
- 下载接口
- 100 % 总线负载，无包损失

如果有多个支持 CANopen 的接口可用，则以下适用于将 CANopen 协议分配给 CAN 接口 (视装置而定)：

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于： | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用： |
|----------------------------------|-----------------------------------|
| 您在控制器配置中添加子元件的顺序                 | 选择您添加子元件的 CAN 接口                  |

## 3.2 CAN: 交换数据

19015

CAN 数据通过国际标准 ISO 11898 中指定的七层 ISO/OSI 参考模型的链路层 (第 2 层) 的 CAN 协议交换。

每个总线参与方都能传输报文 (多主站能力)。交换数据功能与无线电类似。数据在总线上传输, 无发射器或地址。数据仅由标识符标记。每个参与方都负责接收传输的数据, 以及借助标识符检查数据是否与此参与方相关。此过程由 CAN 控制器与运行时系统一起自动执行。

对于正常 CAN 数据交换, 编程人员只需在设计软件时让系统知道数据对象的标识符。这通过用于 CAN 传输和 CAN 接收的功能块实现:

- 基于第 2 层的 CAN POU (RAW CAN):  
简单功能。
- 基于 SAE J1939 的 CAN POU:  
用于发电机管理的高级功能。
- 基于 CANopen 的 CAN POU:  
复杂 CAN 功能。
- CANopen 安全 POU (可选):  
用于具有 SafetyControllers 的安全应用 CAN 功能。

使用这些功能块将以下单元结合到一个数据对象中:

- 有用数据,
- 帧类型 (可选),
- 所选标识符 (ID)。

这些数据对象参与通过 CAN 总线交换数据。传输和接收对象可通过所有有效 IEC 数据类型 (例如, BOOL、WORD、INT、ARRAY) 定义。

CAN 报文包括 COB 标识符 (**COB-ID**) 和最多 8 个数据字节。此 ID 不代表传输或接收模块, 但标识报文。要传输数据, 必须在传输模块中声明传输对象, 以及必须在至少一个其他模块中声明接收对象。这两个声明必须分配给相同的标识符和相同的报文类型 (基本或扩展)。

### 3.2.1 数据接收

19016

原则上, 接收的数据对象自动存储在缓冲区 (即, 不影响用户)。

每个标识符都有一个这样的缓冲区 (队列)。根据应用软件, 此缓冲区按照 FIFO 原则 (先进先出 (First In, First Out)) 通过装置特定 CAN 接收功能块清空。



### 3.2.2 传输数据

19017

通过调用装置特定 CAN 传输功能块，应用程序会恰好向缓冲区（队列）传输一条 CAN 报文。作为反馈，会通知您缓冲区中是否仍有足够的空间，以及报文传输是否成功。缓冲区会将报文自主传输到 CAN 控制器，CAN 控制器在总线上传输该报文。

如果缓冲区已满，则会拒绝传输命令。然后，应用程序必须重复该传输命令。需要向编程人员说明一点此信息。

在缓冲区中，没有基于 COB IDs 向报文分配传输优先级。因此，编程人员必须认真分配其传输报文的顺序。

### 3.3 有关 CANopen 的技术详情

#### 内容

|                            |    |
|----------------------------|----|
| CANopen 网络配置、状态和错误处理 ..... | 26 |
| CoDeSys 支持 CANopen .....   | 27 |
| CANopen 主站 .....           | 28 |
| CANopen 从站 .....           | 51 |
| CANopen 表 .....            | 63 |

13822

#### 3.3.1 CANopen 网络配置、状态和错误处理

13824

相连装置的网络配置和参数设置通过编程软件 CODESYS 执行。

对于某些装置，错误报文只能通过 CANopen 堆栈中的嵌套变量结构到达。

下面的文档为您显示网络配置的结构和用途。

以下章节介绍 CODESYS CANopen 堆栈的内部功能元件及其用途。另外，还提供如何使用网络配置工具的信息。

#### ⓘ 注意

仅使用相应的装置特定库是绝对必要的。通过装置的集成货号可以看到上下文。

以 CAN 接口 1 的 CANopen 主站的 CR0032 为例：

→ ifm\_CR0032\_CANopen1Master\_Vxxyyzz.lib

→ 装置手册，章节“设置目标”

当使用其他库时，装置无法再正常发挥功能。

### 3.3.2 CoDeSys 支持 CANopen

1857

#### 有关带有 CODESYS 的 CANopen 的一般信息

13826

CODESYS 是用于按照国际标准 IEC 61131 编程控制系统的领先系统之一。为使 CODESYS 对用户更有吸引力，许多重要功能已集成到编程系统中，其中包括一个用于 CANopen 的配置工具。此 CANopen 配置工具允许您在 CODESYS 下配置 CANopen 网络（但具有某些限制）。

**Ecomatmobile** 控制器可用作 CANopen 主站和 CANopen 从站。

#### ⚠注意

对于所有 **ecomat mobile** 控制器和 PDM360smart，您必须使用 3S CANopen 库，并添加：  
"OptTableEx"

如果创建新项目，通常会自动加载这些库。如果您通过库管理器添加库，您必须确保进行正确选择。  
未添加此项的 CANopen 库用于其他可编程 ifm 装置。

#### CANopen 术语和实现

1858

对于 CANopen 中的过程数据传输，CAN 网络中没有主站和从站。然而，在 CANopen 中，存在用于网络管理 (NMT) 和用于配置的主/从架构。

CAN 协议（位于 CANopen 协议下）不了解任何主/从关系。

实现假定 CAN 网络用作 CODESYS 可编程控制器的外围。

主站是 NMT 主站和配置主站。通常，主站可确保网络运行。主站主动启动通过配置得知的单个节点（= 网络节点）。这些节点称为从站。

为使主站更接近 CANopen 从站的状态，为主站引入了对象目录。主站也可充当 SDO 服务器（SDO = 服务数据对象），而不仅仅是从站配置阶段的 SDO 客户端。

### 3.3.3 CANopen 主站

#### 内容

|                        |    |
|------------------------|----|
| CANopen 库.....         | 28 |
| 创建 CANopen 项目 .....    | 30 |
| 添加和配置 CANopen 从站 ..... | 35 |
| 主站运行时.....             | 40 |
| 启动 CANopen 网络 .....    | 43 |
| 网络状态 .....             | 45 |

1859

### CANopen 库

#### 内容

|                       |    |
|-----------------------|----|
| 库：CANopen 的系统需要 ..... | 28 |
| CANopen 库的功能.....     | 29 |

18033

### 库：CANopen 的系统需要

14356

当使用 CANopen 功能时，以下库自动集成到 CODESYS 项目中：

- CODESYS 库 3S\_CanDrvOptTableEx.LIB
- CODESYS 库 3S\_CANopenMasterOptTableEx.LIB
- CODESYS 库 3S\_CANopenManagerOptTableEx.LIB
- CODESYS 库 3S\_CanOpenDeviceOptTableEx.LIB
- CODESYS 库 3S\_CanOpenNetVarOptTableEx.LIB
- CODESYS 库 SysLibCallback.LIB

❗不能在应用程序的代码中直接调用包含的功能块和功能！

## CANopen 库的功能

1990

目前, CODESYS CANopen 库支持 CANopen 中定义的以下功能:

- **传输 PDO**: 主站传输到从站(从站 = 节点, 装置)
  - 传输受事件控制(即, 发生更改时),
  - 传输受时间控制(RepeatTimer), 或
  - 作为同步 PDO 传输, 即, 始终在主站传输 SYNC 时传输。也可以使用外部 SYNC 源启动同步 PDO 传输。
- **接收 PDO**: 主站从从站接收  
视从站而定: 事件控制、请求控制、非周期和周期性。
- **PDO 映射**  
CANopen 从站传输/接收的本地对象目录和 PDO 之间的分配(如果从站支持)。
- **传输和接收 SDO**(未分段, 即, 对象目录中每个条目 4 个字节)
  - 在系统启动时通过 SDO 自动配置所有从站。
  - 应用控制的将 SDO 传输到已配置从站和从其接收 SDO。
- **同步**  
CANopen 主站自动传输 SYNC 报文。
- **节点保护**  
为相应配置的每一个从站自动传输保护报文和寿命监控。  
① 建议: 最好为从那时起总线负载较低的当前装置使用检测信号功能。
- **检测信号**  
自动传输和监控检测信号报文。

目前, CODESYS CANopen 库**不支持** CANopen 中定义的以下功能:

- 动态标识符分配
- 动态 SDO 连接
- 逐块 SDO 传输(可借助各自的 ifm 装置库中的功能块为某些 ifm 装置实现)
- 分段 SDO 传输(可借助各自的 ifm 装置库中的功能块实现)
- 上面未提及的 CANopen 协议的所有选项。

**ifm** CANopen 库支持以下功能:

- **紧急**  
从已配置从站和报文存储接收紧急报文。
- 在从站中设置**节点 ID** 和**波特率**。  
通过调用简单的功能, 可以在应用的运行时设置从站的节点 ID 和波特率。

## 创建 CANopen 项目

19021

下面逐步描述了创建具有 CANopen 主站的新项目。假设您已在处理器上安装 CODESYS，并且也已正确安装或复制目标和 EDS 文件。


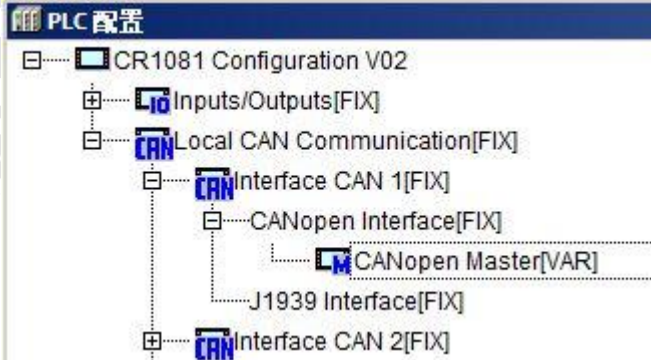
下面详细介绍控制器和 CANopen 配置对话框的调整和应用：

- 在 CODESYS 手册中，参阅 [Ressourcen] > [Steuerungskonfiguration]
- 在 CODESYS 联机帮助中。

► 在创建了新项目之后，在控制器配置中添加 CANopen 主站（→ 设备手册，“设置目标”章节）：

| 对于所有控制器（CR04nn，CR253n 除外）适用于：   | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：  |
|---|--|
| <ul style="list-style-type: none"> <li>► 在第一行上单击鼠标右键 ("CRnnnn Configuration Vnn")</li> <li>► [Unterelement anhängen] &gt; [CANopen Master]</li> </ul> | <ul style="list-style-type: none"> <li>► 在所需 CAN 接口上：<br/>右击 [CANopen Interface]</li> <li>► [Unterelement anhängen] &gt; [CANopen Master]</li> </ul> |

如果有多个支持 CANopen 的接口可用，则以下适用于将 CANopen 协议分配给 CAN 接口（视装置而定）：

| 对于所有控制器（除 CR04nn、CR253n 外），以下所述适用：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：  |
|---|--|
| 您在控制器配置中添加子元件的顺序  | 选择您添加子元件的 CAN 接口   |
| <p>CR0033 示例：</p>  | <p>CR1081 示例：</p>  |

> 以下库和软件模块会自动集成：

| 对于所有控制器 ( CR04nn, CR253n 除外 ) 适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：          |
|---|--|
| STANDARD.LIB，提供用于 IEC 61131 中定义的控制器的标准功能，   | STANDARD.LIB，提供用于 IEC 61131 中定义的控制器的标准功能，  |
| 3S_CanOpenManager.LIB，提供 CANopen 基本功能 ( 如果需要，3S_CanOpenManagerOptTable.LIB，用于 C167 控制器 )，                                   | -  |
| 一个或多个库 3S_CANopenNetVar.LIB、3S_CANopenDevice.LIB 和 3S_CANopenMaster.LIB ( 如果需要，3S_...OptTable.LIB，用于 C167 控制器 )，具体取决于请求的功能， | -  |
| 系统库，例如：SysLibSem.LIB 和 SysLibCallback.LIB。  | 系统库，例如：SysLibSem.LIB 和 SysLibCallback.LIB。 |

- ▶ 要使用准备好的网络状态、状态和 EMCY 功能，在库管理器中手动添加 ifm CANopen 主站库 ( 或 ifm CANopen\_NT 库 )。没有此库，必须直接从 3S CANopen 库的嵌套结构直接读取网络信息。
- ▶ 此外，集成以下库和软件模块：
  - 相应硬件的装置库，例如 ifm\_CR0032\_Vxxyyzz.LIB。此库提供所有装置特定功能。
  - 要在网络上运行的所有从站的 EDS 文件。所有 ifm CANopen 从站的 EDS 文件都由 ifm electronic gmbh 提供。

**对于其他制造商节点的 EDS 文件，请联系相应制造商。**

## CANopen 主站：[CAN parameters] 选项卡

1967

- ▶ 在此对话框窗口中，设置主站最重要的参数。



例如：CAN 接口 1 上的 CR0233 CANopen 主站的控制配置

说明：

### CAN 参数：波特率

10028

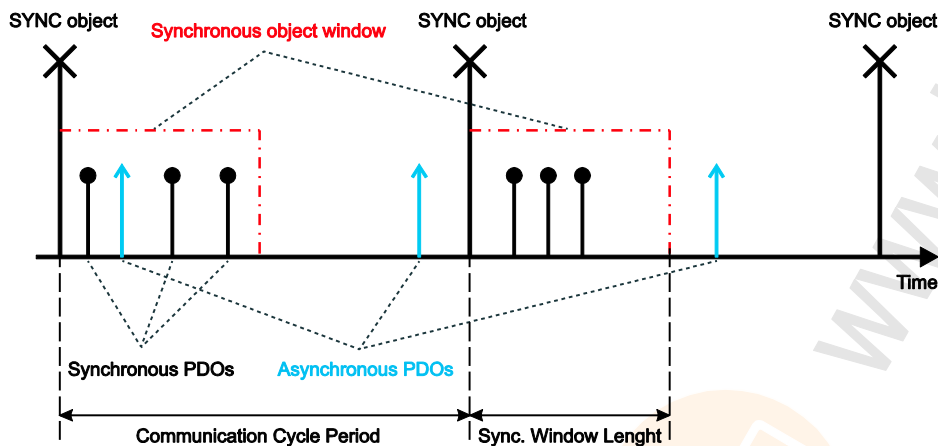
- ▶ 选择主站的波特率。
- 波特率必须与其他网络参与者的传输速度对应。



**CAN 参数：通讯周期间隔/同步 窗口长度**

10029

在 [Communication Cycle Period] 到期后，主站会传输 SYNC 报文。



[Sync. Window Length] 表示由其他网络参与者传输且主站必须接收的同步 PDO 的时间。

正如在大多数应用中，对 SYNC 对象没有特殊要求，可以为 [Communication Cycle Period] 和 [Sync. Window Length] 设置相同的时间。

请确保以 [μs] 为单位输入时间（值 50 000 对应于 50 ms）。

**CAN 参数：同步 COB ID**

10030

在此字段中，可以设置 SYNC 报文的标识符。它始终在通信周期间隔已过之后传输。默认值为 128，且通常不得更改。要启用 SYNC 报文传输，必须设置 [activate] 复选框。

**ⓘ 注意**

始终在程序周期开始时生成 SYNC 报文。然后，读取输入端，处理程序，写入输出端以及传输所有同步 PDO。

请注意，如果设置的 SYNC 时间比程序周期时间短，则 SYNC 时间会更长。

**例如：** 通信周期间隔 = 10 ms，程序周期时间 = 30 ms。

仅在 30 ms 之后传输 SYNC 报文。

**CAN 参数：节点 ID**

10031

► 在此字段中输入主站的节点编号（而不是下载 ID！）。

该节点编号只能在网络中出现一次，否则会干扰通信。

**CAN 参数：自动启动**

10032

在成功配置之后，网络和相连节点会设置为 [operational] 状态，然后启动。

如果未选中此复选框，则必须手动启动网络。

**CAN 参数：检测信号**

10033

如果网络中的其他参与方支持检测信号，则可以选择 [support DSP301, V4.01...] 选项。

必要时，主站可在设置的时间已过之后生成器自己的检测信号信号。



## 添加和配置 CANopen 从站

### 内容

|  |    |
|--|----|
| CANopen 从站： [CAN parameters] 选项卡 .....               | 35 |
| [Receive PDO-Mapping] 和 [Send PDO-Mapping] 选项卡 ..... | 38 |
| CANopen 从站： 注册 [Service Data Objects] .....          | 39 |

1861

接下来，您可以添加 CANopen 从站。为此，您必须再次调用控制器配置 [Insert] > [Append subelement] 中的对话框。存储在目录 PLC\_CONF 中的 CANopen 装置描述 ( EDS 文件 ) 列表可用。通过选择相应的装置，将其直接添加到控制器配置树中。



例如：带有相连 I/O CompactModule 的 CR0020 CANopen 主站的 PLC 配置

### ⚠注意

如果通过 CoDeSys 中的配置对话框添加从站，源代码会为每一个节点动态集成到应用程序中。同时，额外插入的每一个从站都会延长应用程序的周期时间。这意味着：在具有许多从站的网络中，主站无法进一步处理时间关键任务（例如，FB OCC\_TASK）。

具有 27 个从站的网络的基本周期时间为 30 ms。

请注意，不得超过 PLC 周期的最大时间（大约 50 ms）（监视时间：100 ms）。

## CANopen 从站： [CAN parameters] 选项卡

1968

### CAN 参数：节点 ID

10036

节点 ID 用于明确标识 CAN 模块，并与介于 1 至 127 之间的模块集上的编号对应。

此 ID 以十进制形式输入，并在添加新的模块后自动增加 1。

**CAN 参数：编写 DCF**

10037

如果启用 [Write DCF]，则会在将 EDS 文件添加到编译文件的集目录中之后，创建一个 DCF 文件。该 DCF 文件的名称由 EDS 文件的名称和附加节点 ID 组成。

**CAN 参数：创建所有 SDO**

10038

如果启用此选项，则会为所有通信对象生成 SDO。

不会再编写默认值！

**CAN 参数：节点复位**

10039

在重新启动主站后，会在初始化 CANopen 网络期间复位从站（“load”和 NMT 命令“Reset Node”）。然后，配置从站。

**CAN 参数：可选装置**

10040

如果启用 [optional device] 选项，主站会只尝试从此节点读取一次。如果缺少响应，该节点会被忽略，主站进入正常运行状态。

如果从站连接到网络，并在稍后实时检测到，它会自动启动。

为此，您必须在主站的 CAN 参数中已选中 [Automatic startup] 选项。

**CAN 参数：无初始化**

10041

如果启用此选项，主站会立即运行节点，而不传输配置 SDO。（不过，会生成 SDO 数据并存储在控制器中。）

**CAN 参数：节点保护 / 检测信号设置**

10042

根据您可以选择的装置：

- [nodeguarding] 和 [life time factor] 必须设置一项
- 必须设置 [heartbeat]。

如果都设置，则只执行检测信号。

建议：最好为从那时起总线负载较低的当前装置使用检测信号功能。

**CAN 参数：紧急报文**

10043

通常，选中此选项。EMCY 报文使用指定标识符传送。

**CAN 参数：通信周期**

10044

在特殊应用中，可在此处设置由主站生成的 SYNC 报文的监控时间。

请注意，此时间必须大于主站的 SYNC 时间。必须通过实验确定最佳值。

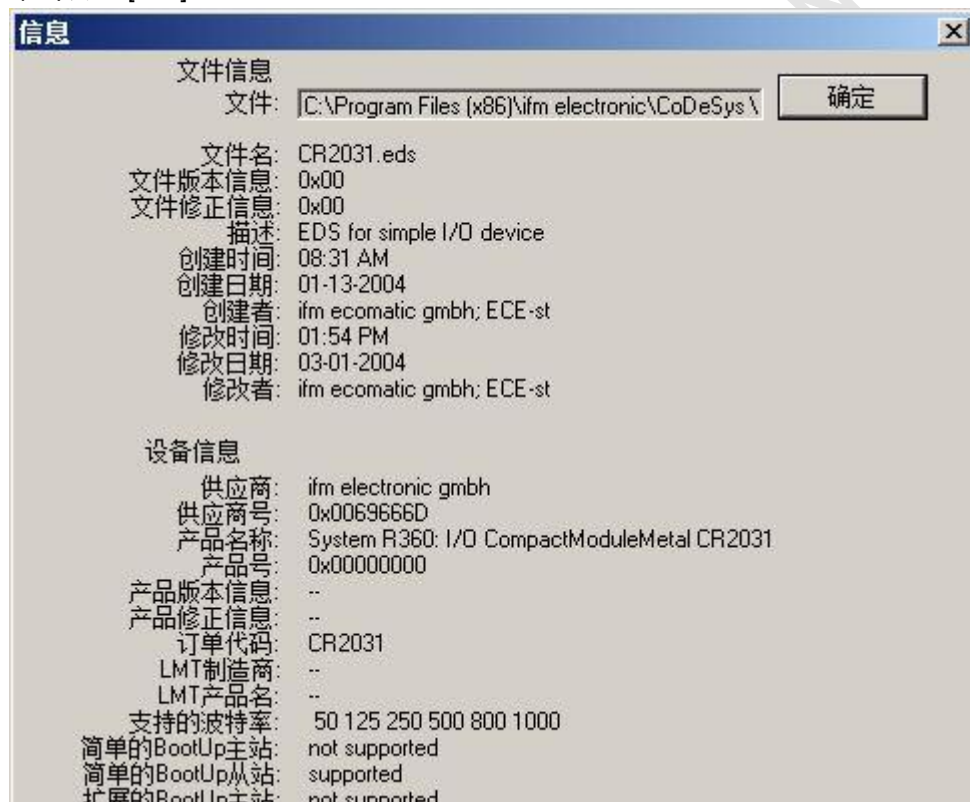
在大多数情况下，进行节点监控有节点保护或检测信号足以。

**CAN 参数：信息**

18062

在 EDS 文件中显示此从站的描述：

- ▶ 单击按钮 [Info]：



例如：有关从站 CR2031 的信息

- ▶ 再次使用 [OK] 关闭显示屏。

## [Receive PDO-Mapping] 和 [Send PDO-Mapping] 选项卡

1969

使用 CAN 模块的配置对话框中的 [Receive PDO-Mapping] 和 [Send PDO-Mapping] 选项卡，可更改 EDS 文件中描述的模块映射（CANopen 从站传输/接收的本地对象目录和 PDO 之间的分配）（如果 CAN 模块支持）。

EDS 文件的所有 [mappable] 对象位于左侧，并可添加到右侧的 PDO（过程数据对象）中，或从中删除。

可添加 [StandardDataTypes] 以在 PDO 中生成空格。

## PDO 映射：插入

10046

使用 [Insert] 按钮，您可生成更多 PDO 并插入相应对象。输入端和输出端通过插入的 PDO 分配给 IEC 地址。

在控制器配置中，可在关闭对话框之后看到进行的设置。可为单个对象提供符号名称。

## PDO 映射：属性

10047

标准中定义的 PDO 属性可在对话框中通过属性编辑。

|        |   |
|--------|---|
| COB-ID | 每一个 PDO 报文都需要一个明确的 COB Id（通信对象标识符）。如果模块不支持某个选项，或者不能更改值，则该字段未灰色且无法编辑。  |
| 禁止时间   | 禁止时间（100 μs）是此 PDO 的两条报文之间的最短时间，以便在更改此值时传输的报文的传输频率不会过高。单位为 100 μs。  |
| 传输类型   | <p>对于传输类型，您会收到此模块的可能的传输模式选择：</p> <p><b>非周期 - 同步</b><br/>更改后，随下一次 SYNC 传输 PDO。</p> <p><b>周期性 - 同步</b><br/>同步传输 PDO。[Number of SYNCs] 表示此 PDO 的两次传输之间的同步报文数量。</p> <p><b>异步 - 装置配置文件特定</b><br/>在发生事件时传输 PDO，例如，当值更改时。装置配置文件定义可以此方式传输的数据。</p> <p><b>异步 - 制造商特定</b><br/>在发生事件时传输 PDO，例如，当值更改时。装置制造商定义以此方式传输的数据。</p> <p><b>（异步）同步 - 仅限 RTR</b><br/>不实现这些服务。</p> <p><b>SYNC 数</b><br/>根据传输类型，可编辑此字段以输入随后再次传输 PDO 的同步报文数量（[Com. Cycle Period]、[Sync Window Length]、[Sync. COB ID] 的 CAN 参数对话框中的定义）。</p> <p><b>事件时间</b><br/>根据传输类型，此字段中指示两次 PDO 传输之间所需的最大期限（以毫秒 [ms] 为单位）。</p> |

## CANopen 从站：注册 [Service Data Objects]

18036

| 索引号      | 名称                | 值    | type       | 缺省值  |
|----------|-------------------|------|------------|------|
| 2000sub1 | chan 1            | 0x02 | Unsigned8  | 0x02 |
| 2000sub2 | chan 2            | 0x02 | Unsigned8  | 0x02 |
| 2000sub3 | chan 3            | 0x02 | Unsigned8  | 0x02 |
| 2000sub4 | chan 4            | 0x02 | Unsigned8  | 0x02 |
| 2000sub5 | chan 5            | 0x02 | Unsigned8  | 0x02 |
| 2000sub6 | chan 6            | 0x02 | Unsigned8  | 0x02 |
| 2000sub7 | chan 7            | 0x02 | Unsigned8  | 0x02 |
| 2000sub8 | chan 8            | 0x02 | Unsigned8  | 0x02 |
| 2001     | PWM frequency     | 100  | Unsigned8  | 100  |
| 2004sub1 | P - value         | 50   | Unsigned8  | 50   |
| 2004sub2 | I - value         | 20   | Unsigned8  | 20   |
| 2004sub3 | Max current       | 4000 | Unsigned16 | 4000 |
| 2005sub1 | P - value         | 50   | Unsigned8  | 50   |
| 2005sub2 | I - value         | 20   | Unsigned8  | 20   |
| 2005sub3 | Max current       | 4000 | Unsigned16 | 4000 |
| 2006sub1 | P - value         | 50   | Unsigned8  | 50   |
| 2006sub2 | I - value         | 20   | Unsigned8  | 20   |
| 2006sub3 | Max current       | 4000 | Unsigned16 | 4000 |
| 2007sub1 | P - value         | 50   | Unsigned8  | 50   |
| 2007sub2 | I - value         | 20   | Unsigned8  | 20   |
| 2007sub3 | Max current       | 4000 | Unsigned16 | 4000 |
| 20f0     | Node id switch a  | 0x20 | Unsigned8  | 0x20 |
| 20f1     | Node id switch b  | 0x20 | Unsigned8  | 0x20 |
| 20f2     | Bit rate switch a | 0x04 | Unsigned8  | 0x04 |
| 20f3     | Bit rate switch b | 0x04 | Unsigned8  | 0x04 |
| 6200sub1 | binary outputs 1  |      | Unsigned8  |      |
| 6411sub1 | chan 1            |      | Integer16  |      |
| 6411sub2 | chan 2            |      | Integer16  |      |
| 6411sub3 | chan 3            |      | Integer16  |      |
| 6411sub4 | chan 4            |      | Integer16  |      |

这里列出了 EDS 或 DCF 文件的所有对象，它们位于索引 0x2000...0x9FFF 的范围内并定义为可写。

这里指出了索引、名称、当前值、类型和默认值。

只有 [Wert] 中的值可更改。

## SDOs: 更改值

18039

[Wert] 中的值可更改：

- ▶ 双击请求的条目。
- ▶ 输入新的值。
- ▶ 使用 [Eingabe] 确认更改，或使用 [ESC] 拒绝。

在 CAN 初始化期间：

- > 不同于默认值的值以 SDO (服务数据对象) 形式传输到 CAN 模块。
- > 因此，这些值对 CANopen 从站的对象目录具有直接影响。
- > 通常，在每次启动应用程序时会重新编写这些值 – 而无论它们是否永久存储在 CANopen 从站中。
- > 如果值被删除而未输入新值，则将在初始化期间传输默认值。



## 主站运行时

### 内容

|                            |    |
|----------------------------|----|
| 在系统启动时，在总线上复位所有已配置从站 ..... | 40 |
| 轮询从站装置类型.....              | 41 |
| 配置所有正确检测到的装置 .....         | 41 |
| 自动配置从站 .....               | 41 |
| 启动所有配置正确的从站.....           | 41 |
| 周期性传输 SYNC 报文 .....        | 42 |
| 带寿命监控的节点保护.....            | 42 |
| 从主站到从站的检测信号.....           | 42 |
| 接收紧急报文 .....               | 43 |

8569

这里介绍有关 CANopen 主站库在运行时的功能的信息。

CANopen 主站库为 CODESYS 应用提供对大多数应用来说足够的隐式服务。 这些服务以透明方式为用户集成，并在应用中可用，无需额外调用。 以下描述假设 CANopenMaster 库（或 CANopen\_NT 库）被手动添加到库管理器中以使用网络诊断、状态和 EMCY 功能。

CANopen 主站库的服务：

### 在系统启动时，在总线上复位所有已配置从站

19027

CAN 文档 DSP301 中介绍了各个 NMT 命令。

根据 CANopen，NMT 代表网络管理 (Network Management)。

### 逐个复位从站

19029

要复位从站，分别为每个从站显式使用 NMT 命令“Reset Node”作为标准。

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                                    |
|---|--|
| 使用 FB CANx_MASTER_STATUS :<br>• GLOBAL_START = FALSE<br>• RESET_ALL_NODES = FALSE | 使用 FB CANOPEN_NMTSERVICES :<br>• NODE = 从站的节点 ID<br>• NMTSERVICE = 3 |



## 一次复位所有从站

19031

为避免具有功能不太强大的 CAN 控制器的从站过载，使用“Reset All Nodes”命令一次复位所有相连从站十分有用。

| 对于所有控制器（CR04nn，CR253n 除外）适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                            |
|--|--|
| 使用 FB CANx_MASTER_STATUS：<br>• GLOBAL_START = TRUE<br>• RESET_ALL_NODES = TRUE | 使用 FB CANOPEN_NMTSERVICES：<br>• NODE = 0<br>• NMTSERVICE = 3 |

## 轮询从站装置类型

8021

使用 SDO 轮询从站装置类型（轮询对象 0x1000）并比较配置的从站 ID：

- > 在以下情况下，在 0.5 s 之后重复此请求 ...
  - 未收到装置类型
  - 以及从站未在配置中标识为可选
  - 以及超时时间未过。

指示从其收到错误装置类型的从站的错误状态。

## 配置所有正确检测到的装置

8022

如果从站没有在监控时间内作出响应，则会监控每一个 SDO 的响应并重复 SDO。

## 自动配置从站

8023

在总线运行时，使用 SDO 自动配置从站：

先决条件：从站已通过启动报文记录在主站中。

## 启动所有配置正确的从站

19032

在相应从站配置结束之后，启动所有正确配置的从站：

## 逐个启动从站

19034

通常，NMT 命令“Start Node”用于启动从站。

| 对于所有控制器（CR04nn，CR253n 除外）适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                                   |
|--|---|
| 使用 FB CANx_MASTER_STATUS：<br>• GLOBAL_START = FALSE<br>• START_ALL_NODES = FALSE | 使用 FB CANOPEN_NMTSERVICES：<br>• NODE = 从站的节点 ID<br>• NMTSERVICE = 2 |

### 一次启动所有从站

19036

就像“复位”一样，此命令可以替换为“Start All Nodes”。

| 对于所有控制器 ( CR04nn , CR253n 除外 ) 适用于 :  | 对于所有 CR04nn、CR1nnn、CR253n , 以下所述适用 :                          |
|---|---|
| 使用 FB CANx_MASTER_STATUS :<br>• GLOBAL_START = TRUE<br>• START_ALL_NODES = TRUE | 使用 FB CANOPEN_NMTSERVICES :<br>• NODE = 0<br>• NMTSERVICE = 2 |

### 周期性传输 SYNC 报文

8025

此值只能在配置期间设置。

### 带寿命监控的节点保护

19038

我们的建议：最好为从那时起总线负载较低的当前装置使用检测信号功能。

可以为每一个从站设置带寿命监控的节点保护：

| 对于所有控制器 ( CR04nn , CR253n 除外 ) 适用于 :  | 对于所有 CR04nn、CR1nnn、CR253n , 以下所述适用 :  |
|---|---|
| FB CANx_MASTER_STATUS 显示最多 8 个从站的错误状态：<br>> ERROR_CONTROL = 缺少的网络节点列表 ( 保护或检测信号错误 ) | 使用 FB CANOPEN_GETGUARDBERRLIST 批量列出主站检测到错误的所有节点：<br>保护错误，检测信号错误<br>> N_NODES = 包含检测信号或保护错误的节点的数量<br>> NODEID = 包含检测信号或保护错误的节点 ID 列表 |

### 从主站到从站的检测信号

19039

| 对于所有控制器 ( CR04nn , CR253n 除外 ) 适用于 :  | 对于所有 CR04nn、CR1nnn、CR253n , 以下所述适用 :  |
|---|---|
| FB CANx_MASTER_STATUS 显示最多 8 个从站的错误状态：<br>> ERROR_CONTROL = 缺少的网络节点列表 ( 保护或检测信号错误 ) | 使用 FB CANOPEN_GETGUARDBERRLIST 批量列出主站检测到错误的所有节点：<br>保护错误，检测信号错误<br>> N_NODES = 包含检测信号或保护错误的节点的数量<br>> NODEID = 包含检测信号或保护错误的节点 ID 列表 |

## 接收紧急报文

19042

接收每个从站的紧急报文，存储最后接收的紧急报文：

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：   |
|---|---|
| 使用 FB CANx_MASTER_STATUS 读取错误报文：<br>> EMERGENCY_OBJECT_SLAVES = 当前 EMCY 报文列表<br>> GET_EMERGENCY = 最后生成的 EMCY 报文 | 使用 FB CANOPEN_GETEMCYMESSAGES 读取错误报文：<br>> N_MSGS = 累积报文的数量<br>> EMCY = 紧急报文列表最近的条目位于索引 0 |

## 启动 CANopen 网络

19048

下面介绍有关如何启动 CANopen 网络的信息。

在将项目下载到控制器或复位应用之后，主站会再次启动 CAN 网络。这会始终以相同的操作序列进行：

- 复位所有从站，除非它们在配置工具中标记为 [nicht initialisieren]。它们使用具有从站节点 ID 的“Reset Node”(0x81) 命令逐个复位。  
→ 章节 **启动所有配置正确的从站** (→ 页 41)
- 配置所有从站。为此，轮询从站的对象 0x1000。
  - 如果从站在 0.5 秒的监控时间内响应：
    - > 发送下一个响应配置 SDO。
  - 如果从站配置为 [optional] 且未在响应时间内响应对象 0x1000 的轮询：
    - > 其会标记为“not available”且
    - > 不会向其发送更多 SDO。
  - 如果从站使用与配置的类型不同的类型（以低 16 位）响应对象 0x1000 的轮询：
    - > 将不会配置它，且
    - > 其将标记为“wrong type”。
- 只要在监控时间内看到从站响应，就会重复所有 SDO。应用程序可监控各个从站的启动。
- ▶ 必要时，取消从站的初始化，如果...
  - 从站不存在**以及**
  - 从站未配置为 [optional]：

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用： |
|---|-----------------------------------|
| 在阵列 NODE_STATE_SLAVE 中设置 FB CANx_MASTER_STATUS 的标记 SET_TIMEOUT_STATE = TRUE | 超时结束后，FB 停止等待。                    |

- ▶ 必要时，跳过使用与配置的类型不同的类型响应对象 0x1000 的轮询从站的初始化：

| 对于所有控制器 ( CR04nn, CR253n 除外 ) 适用于 :                                  | 对于所有 CR04nn, CR1nnn, CR253n, 以下所述适用 :                                |
|--|--|
| 在阵列 NODE_STATE_SLAVE 中设置 FB<br>CANx_MASTER_STATUS 的标记 SET_NODE_STATE | 使用 FB CANOPEN_NMTSERVICES :<br>• NODE = 从站的节点 ID<br>• NMTSERVICE = 1 |

- > 因此，将从站设置为 PRE-OPERATIONAL 状态。
- 如果稍后将从站设置为 OPERATIONAL 状态：
  - > 主站不会向该从站发送任何 PDO
  - > 该从站发送的 PDO 均被忽略。
- 如果主站配置的检测信号时间不是 0：
  - > 会在主控制器启动之后立即生成检测信号。
- 在所有从站获取其配置 SDO 之后：
  - > 使用配置的节点保护开始保护从站。
- 如果主站配置为 [automatisch starten]：
  - > 则会使用 NMT 命令“Start Remote Node”(0x01)
  - > 每个从站都有主站单独启动。
- 如果设置了标记 GLOBAL\_START：
  - > NMT 命令与节点 ID 0 一起使用
  - > 会使用“Start all Nodes”启动所有从站。
- 至少发送一次所有已配置的 TX-PDO  
(对于主站，这些是 RX-PDO)。
- ▶ 其 [automatisch starten] 被禁用，单独启动每个从站：
  - 通过 NODE\_STATE\_SLAVE 阵列中的标记 START\_NODE，或
  - → 章节 **启动所有配置正确的从站** (→ 页 41)。

## 网络状态

### 内容

|                                   |    |
|-----------------------------------|----|
| CANopen 主站启动 .....                | 45 |
| CANopen 从站启动 .....                | 46 |
| 启动网络, 无 [Automatic startup] ..... | 46 |
| CANopen 主站的对象目录 .....             | 48 |

19050

下面介绍如何解释 CANopen 网络的状态以及如何作出反应。

对于启动 CANopen 网络 (→ 章节 **启动网络** (→ 页 43)) 以及在操作期间, 库的单个功能块会经历不同状态。

### 注意

在 CODESYS 的监控模式 (在线模式) 下, 可以在全局变量列表“CANOpen implicit variables”中看到 CAN 网络的状态。这需要准确了解 CANopen 以及 CODESYS CANopen 库的结构。

为便于访问, 以下 FB 在 CANopen 主站库中可用 (特定于装置和 CAN 通道) :

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于 : | 对于所有 CR04nn、CR1nnn、CR253n, 以下所述适用 : |
|-----------------------------------|-------------------------------------|
| CANx_MASTER_STATUS                | CANOPEN_GETSTATE                    |

## CANopen 主站启动

19056

在 CAN 网络启动期间, 主站会经历这里列出的不同状态 :

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于 :                        | 对于所有 CR04nn、CR1nnn、CR253n, 以下所述适用 :   |
|--|---|
| FB CANx_MASTER_STATUS:<br>> NODE_STATE = CANopen 主站的当前状态 | FB CANOPEN_GETSTATE :<br>> MASTERSTATE = 主站内部状态<br>> CANSTATE = CANopen 网络的状态 |

详细信息→ 章节 **CANopen 主站的 NMT 状态** (→ 页 71)

每当从站没有对 SDO 请求 (上传或下载) 作出响应时, 请求会重复。主站离开状态 3 (如上所述), 但不是在所有 SDO 均已成功传输之前。因此, 可以检测是否缺少从站或者从站是否未正确接收所有 SDO。从站是通过确认还是通过中止响应对主站来说并不重要。对于主站来说, 只有其究竟是否收到响应才是重要的。

标记为 [optional] 的从站是例外。仅会对可选从站询问一次其对象 0x1000。如果它们未在 0.5 秒内作出响应, 主站会首先忽略该从站, 且主站会进入状态 5, 不会对此从站作出其他反应。

## CANopen 从站启动

19057

您可以在这里看到从站的状态（从主站的角度）：

| 对于所有控制器（CR04nn，CR253n 除外）适用于：                           | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                      |
|---|--|
| FB CANx_SLAVE_STATUS:<br>> NODE_STATE = CANopen 从站的当前状态 | FB CANOPEN_GETNMTSTATESLAVE:<br>> NMTSTATE = 节点的网络运行状态 |

详细信息 → 章节 **CANopen 从站的 NMT 状态** (→ 页 [71](#))

## 启动网络，无 [Automatic startup]

8583

有时，应用必须确定启动 CANopen 从站的时间。为此，必须在配置中禁用 CANopen 主站的 [Automatic startup] 选项。然后，由应用启动从站。

## 使用 GLOBAL\_START 启动网络

19073

在具有许多参与方（在大多数情况下超过 8 个）的 CAN 网络中，通常会发生并非所有（主要是较慢）IO 节点（例如，CompactModules Cr2013）都会检测到连续 NMT 报文的情况。原因是这些节点必须侦听具有 ID 0 的所有报文。传输间隔过短的 NMT 报文会使此类节点的接收缓存过载。

减少连续 NMT 报文的数量对此有帮助：

| 对于所有控制器（CR04nn，CR253n 除外）适用于：                      | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：               |
|--|---|
| 使用 FB CANx_MASTER_STATUS：<br>▶ GLOBAL_START = TRUE | 使用 FB CANOPEN_SETSTATE：<br>▶ GlobalStart = TRUE |

- > CANopen 主站库使用命令“Start All Nodes”，而不是使用命令“Start Node”单独启动所有节点。
- > GLOBAL\_START 仅在网络初始化时执行一次。
- > 如果设置了此输入端，则控制器也会启动具有状态 98（见上文）的节点。但是，这些节点的 PDO 保持禁用。

### 使用 START\_ALL\_NODES 启动网络

19074

如果没有使用 GLOBAL\_START 自动启动网络，则可逐个启动每个节点。

如果未请求，选项如下所示：

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于：   | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                                   |
|--|---|
| 使用 FB CANx_MASTER_STATUS：<br>• GLOBAL_START = FALSE<br>• START_ALL_NODES = FALSE | 使用 FB CANOPEN_NMTSERVICES：<br>• NODE = 从站的节点 ID<br>• NMTSERVICE = 2 |

- START\_ALL\_NODES 通常由应用程序在运行时设置。
- > 如果设置了此输入端，则也会启动具有状态 98（见上文）的节点。但是，这些节点的 PDO 保持禁用。

### 使用 RESET\_ALL\_NODES 初始化网络

19075

适用于命令“Start All Nodes”的原因也同样适用于 NMT 命令“Reset All Nodes”（而不是针对每个单个节点的命令“Reset Nodes”）。

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于：   | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                            |
|--|--|
| 使用 FB CANx_MASTER_STATUS：<br>• GLOBAL_START = TRUE<br>• RESET_ALL_NODES = TRUE | 使用 FB CANOPEN_NMTSERVICES：<br>• NODE = 0<br>• NMTSERVICE = 3 |

- > 这会一次同时复位所有节点。

### 访问 CANopen 主站的状态

19076

您应轮询主站的状态，以便在 IO 网络就绪之前不会处理应用代码。以下代码片段示例显示一个选项：

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：  |
|---|--|
| <b>变量声明</b><br>VAR<br>FB_MasterStatus := CR0020_MASTER_STATUS;<br>;<br>END_VAR            | <b>变量声明</b><br>VAR<br>FB_MasterStatus : CANOPEN_GETSTATE;<br>...<br>END_VAR            |
| <b>程序代码</b><br>IF FB_MasterStatus.NODE_STATE = 5 THEN<br><application code><br>END_IF     | <b>程序代码</b><br>IF FB_MasterStatus.MASTERSTATE = 5 THEN<br><application code><br>END_IF |
| 通过在 FB CANx_MASTER_STATUS 的阵列 NODE_STATE_SLAVE 中设置标记 TIME_OUT_STATE，例如，应用可以响应以及跳过不可配置的节点。 | 通过设置 FB CANOPEN_GETSTATE 的输入端 NODE 的值，例如，应用可以响应以及跳过无法配置的节点。                            |

## CANopen 主站的对象目录

### 内容

|                      |    |
|----------------------|----|
| 访问对象目录 ( 控制器 ) ..... | 48 |
| 访问对象目录 ( 其他 ) .....  | 50 |

19156

在某些情况下，如果 CANopen 主站具有其自己的对象目录将十分有用。例如，这会实现应用与其他 CAN 节点的数据交换。

主站的对象目录在编译期间使用名为 CRnnnnMasterODEntry.EDS 的 EDS 文件生成并提供默认值。此 EDS 文件存储在目录 CoDeSys Vn\Library\PLCconf 中。此 EDS 文件的内容可以通过配置窗口 [CAN parameters] 中的按钮 [EDS...] 查看。

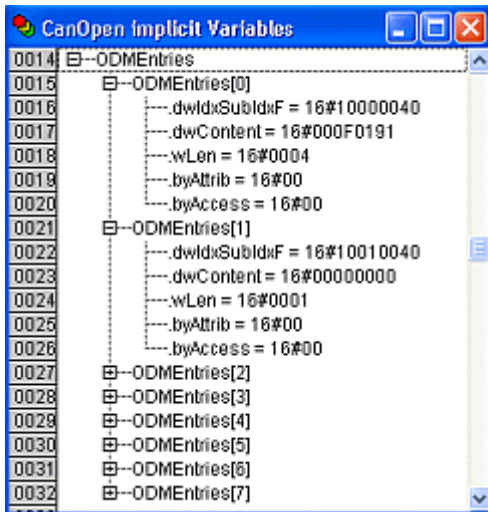
即使对象目录不可用，也可无限制地使用主站。

### 访问对象目录 ( 控制器 )

19157

对于所有控制器 ( CR04nn , CR253n 除外 ) 适用于 :

对象目录由应用通过具有以下结构的阵列访问 :





| 结构元素          | 描述  |
|---------------|---|
| .dwIdxSubIdxF | 组件 0xiiiiissff 的结构：<br>iiiii – index (2 bytes, bits 16...31), ldx<br>ss – 子索引 (1 个字节, 第 8...15 位), SubIdx<br>ff – 标记 (1 个字节, 第 0...7 位), F<br><br>标记位的含义：<br>第 0 位：写入<br>第 1 位：内容是指向某个地址的指针<br>第 2 位：可映射<br>第 3 位：swap<br>第 4 位：带有符号的值<br>第 5 位：浮点<br>第 6 位：包含更多子索引 |
| .dwContent    | 包含条目的内容   |
| .wLen         | 数据的长度   |
| .byAttrib     | 最初设计作为访问授权<br>可由主站的应用自由使用   |
| .byAccess     | 在过去的访问授权中<br>可由主站的应用自由使用  |

在平台上，CODESYS 没有用于此对象目录的编辑器。

EDS 文件仅会确定用于创建对象目录的对象。条目始终是使用长度 4 生成的，且标记（对象目录条目 .dwIdxSubIdxF 的组件的最低有效字节）始终具有值 1。这意味着两个字节的值均为 0x41。

如果对象目录在主站中可用，则主站可在网络中充当 SDO 服务器。每当客户端通过写入访问该对象目录的某个条目时，就会在 CANx\_MASTER\_STATUS 中通过标记 OD\_CHANGED 通知应用。在评估之后，必须复位此标记。

应用可通过直接写入或读取条目，或通过将条目指向 IEC 变量，来使用该对象目录。这意味着：当读取/写入另一个节点时，可直接访问这些 IEC 变量。

如果对象目录的索引和子索引已知，可按如下方式寻址条目：

```
I := GetODMEntryValue(16#iiiiiss00, pCanOpenMaster[0].wODMFirstIdx,
pCanOpenMaster[0].wODMFirstIdx + pCanOpenMaster[0].wODMCount;
```

对于“iiii”，必须使用索引，对于“ss”，必须使用子索引（作为十六进制值）。

阵列条目的编号在 I 中可用。您现在可以直接访问该条目的组件。

输入地址、长度和标记足以，这样可以将此条目直接传输到 IEC 变量：

```
ODMEntries[I].dwContent := ADR(<variable name>);
ODMEntries[I].wLen := sizeof(<variable name>);
ODMEntries[I].dwIdxSubIdxF := ODMEntries[I].dwIdxSubIdxF OR OD_ENTRYFLG_WRITE OR
OD_ENTRYFLG_ISPOINTER;
```

更改“.dwContent”的内容足以仅更改该条目的内容。

## 访问对象目录 (其他)

19158

对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：

对象目录由应用通过功能块访问：

- CANOPEN\_GETODCHANGEDFLAG,
- CANOPEN\_READOBJECTDICT,
- CANOPEN\_WRITEOBJECTDICT.

在平台上，CODESYS 没有用于此对象目录的编辑器。

EDS 文件仅会确定用于创建对象目录的对象。

如果对象目录在主站中可用，则主站可在网络中充当 SDO 服务器。每当客户端通过写入访问该对象目录的某个条目时，就会在 CANOPEN\_GETODCHANGEDFLAG 中通过标记 ODCHANGED 通知应用。在评估之后，必须通过输入端 RESETFLAG=TRUE 复位此标记。

应用可通过直接写入或读取条目来使用该对象目录。

### 3.3.4 CANopen 从站

#### 内容

|                        |    |
|------------------------|----|
| CANopen 从站库的功能.....    | 51 |
| 配置 CANopen 从站 .....    | 53 |
| 运行时访问 CANopen 从站 ..... | 61 |

1865

CODESYS 可编程控制器也可以是 CAN 网络中的 CANopen 从站。

#### CANopen 从站库的功能

19161

CANopen 从站库与 CANopen 配置工具相结合为用户提供以下选项：

- 在 CODESYS 中：配置设备运行的节点保护/检测信号、紧急、节点 ID 和波特率的属性。
- 连同 CODESYS 中的参数管理器，可以创建一个默认 PDO 映射，该映射可由主站在运行时更改。PDO 映射的配置由主站在配置阶段期间更改。借助映射，可以将应用的 IEC 变量映射到 PDO。这意味着会将 IEC 变量分配给 PDO，以便能够在应用程序中轻松评估它们。
- CANopen 从站库提供一个对象目录。此对象目录的大小在编译 CODESYS 时定义。此目录包含描述 CANopen 从站的所有对象以及参数管理器定义的对象。在参数管理器中，仅列表类型参数和变量可用于 CANopen 从站。
- CANopen 从站会管理对该对象目录的访问，例如，它充当总线上的 SDO 服务器。
- CANopen 从站会监控节点保护或检测信号花费时间（始终仅一个创建器），并为应用设置相应的错误标记。
- 可以生成一个 EDS 文件，描述 CANopen 从站的已配置属性，以便可以集成装置并配置为 CANopen 主站下的从站。

CANopen 从站库不会显式提供 CANopen 中描述的以下功能（此处未指出或上面的部分中未实现的 CANopen 协议中的所有选项）：

- 动态 SDO 和 PDO 标识符
- SDO 块传输

- 自动生成紧急报文。 紧急报文必须始终由应用程序生成。 为此，CANopen 从站库提供以下 FB：

- 对于所有控制器 (CR04nn, CR253n除外) 适用于：

|                                  |   |
|----------------------------------|---|
| <b>CANx_SLAVE_EMCY_HANDLER</b>   | 处理 CAN 接口 x 上 CANopen 从站的装置特定错误状态：<br>• 错误寄存器 (索引 0x1001) 和<br>• CANopen 对象目录的错误场 (索引 0x1003)<br>x = 1...n = CAN 接口编号 (视装置而定， → 技术资料) |
| <b>CANx_SLAVE_SEND_EMERGENCY</b> | 发送 CAN 接口 x 上 CANopen 从站的应用特定错误状态<br>x = 1...n = CAN 接口编号 (视装置而定， → 技术资料)   |

- 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：

|                                 |   |
|---------------------------------|---|
| <b>CANOPEN_GETERRORREGISTER</b> | = 获取 CANopen 错误寄存器<br>从控制器读取错误寄存器 0x1001 和 0x1003<br>寄存器可通过设置相应的输入端复位。    |
| <b>CANOPEN_GETEMCYMESSAGES</b>  | = 获取 CANopen 紧急报文<br>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。<br>该列表可通过设置相应的输入端复位。 |
| <b>CANOPEN_SENDEMCMESAGE</b>    | = CANopen 发送紧急报文<br>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                 |

- 当前仅在 StartNode NMT 报文到达时接受 PDO 属性的动态更改，而不使用 CANopen 中定义的机制。

## 配置 CANopen 从站

### 内容

|                                |    |
|--------------------------------|----|
| [Base settings] 选项卡.....       | 53 |
| [CAN settings] 选项卡.....        | 57 |
| [Default PDO mapping] 选项卡..... | 58 |
| 通过主站配置更改标准映射.....              | 60 |

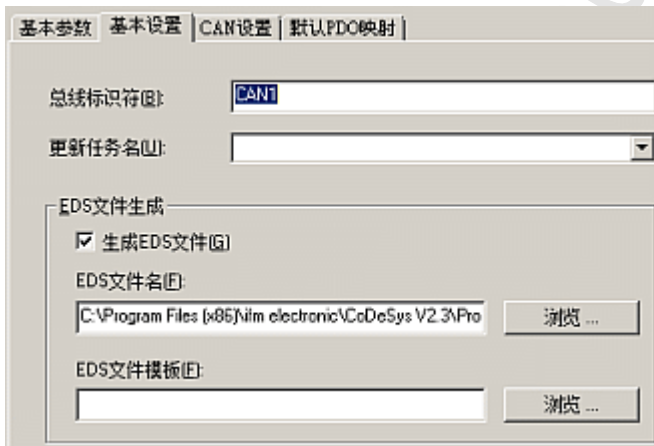
19163  
1980

请求：将控制器用作 CANopen 从站：

- ▶ 在 PLC 配置中，必须添加从站：
  - 右键单击第一行 ("CRnnnn Configuration Vnn")
  - [Append Subelement] > [CANopen Slave...]
- > 如果有多个支持 CANopen 的接口可用，则以下适用于将 CANopen 协议分配给 CAN 接口（视装置而定）：
  - 章节“CAN 接口和 CAN 协议”
- > 所有需要的库都会自动添加到库管理器中。

### [Base settings] 选项卡

1981



#### 基本设置：总线标识符

10049

当前未使用参数。

#### 基本设置：更新任务的名称

10050

调用 CANopen 从站的任务的名称。

### 基本设置：生成 EDS 文件

10051

如果能够通过将 CANopen 从站添加到任何主站配置中的设置生成 EDS 文件，则必须启用 [Generate EDS file] 选项并指定文件名。作为一个选项，可指示其条目添加到 CANopen 从站的 EDS 文件中的模板文件。如果重叠，不会改写模板定义。



## 对象目录示例

1991

例如，对象目录中可能存在以下条目：

```
[FileInfo]
FileName=D:\CoDeSys\lib2\plcconf\MyTest.eds
FileVersion=1
FileRevision=1
Description=EDS for CoDeSys-Project: D:\CoDeSys\CANopenTestprojekte\TestHeartbeatODsettings_Device.pro
CreationTime=13:59
CreationDate=09-07-2005
CreatedBy=CoDeSys
ModificationTime=13:59
ModificationDate=09-07-2005
ModifiedBy=CoDeSys

[DeviceInfo]
VendorName=3S Smart Software Solutions GmbH
ProductName=TestHeartbeatODsettings_Device
ProductNumber=0x33535F44
ProductVersion=1
ProductRevision=1
OrderCode=xxxx.yyyy.zzzz
LMT_ManufacturerName=3S GmbH
LMT_ProductName=3S_Dev
BaudRate_10=1
BaudRate_20=1
BaudRate_50=1
BaudRate_100=1
BaudRate_125=1
BaudRate_250=1
BaudRate_500=1
BaudRate_800=1
BaudRate_1000=1
SimpleBootUpMaster=1
SimpleBootUpSlave=0
ExtendedBootUpMaster=1
ExtendedBootUpSlave=0
...

[1018sub0]
ParameterName=Number of entries
ObjectType=0x7
DataType=0x5
AccessType=ro
DefaultValue=2
PDOMapping=0

[1018sub1]
ParameterName=VendorID
ObjectType=0x7
DataType=0x7
AccessType=ro
DefaultValue=0x0
PDOMapping=0

[1018sub2]
ParameterName=Product Code
ObjectType=0x7
DataType=0x7
AccessType=ro
DefaultValue=0x0
PDOMapping=0
```

有关单个对象的含义，请参阅 CANopen 规范 DS301。

除规定条目外，EDS 文件还包含 SYNC、保护、紧急和检测信号的定义。如果不使用这些对象，则可将值设置为 0（预设值）。但是，由于这些对象在运行时在从站的对象目录中存在，因此会将它们写入 EDS 文件。

这同样适用于通信和映射参数的条目。映射对象 0x16nn 或 0x1Ann 的所有 8 个可能的子索引均存在，但可能不包含在子索引 0 中。

**!**库不支持位映射！



## [CAN settings] 选项卡

1982

在此选项卡中可以设置**节点 ID** 和**波特率**。

**装置类型**

(这是在 EDS 中输入的对象 0x1000 的默认值) 具有 0x191 作为默认值 (标准 IO 装置), 并且可以自由更改。

CAN 控制器的索引来自 CANopen 从站在控制器配置中的位置。

**节点保护**参数、**检测信号**参数和**紧急 COB ID** 也可以在此选项卡中定义。 只能为监控检测信号配置 CANopen 从站。

建议: 最好为从那时起总线负载较低的当前装置使用检测信号功能。

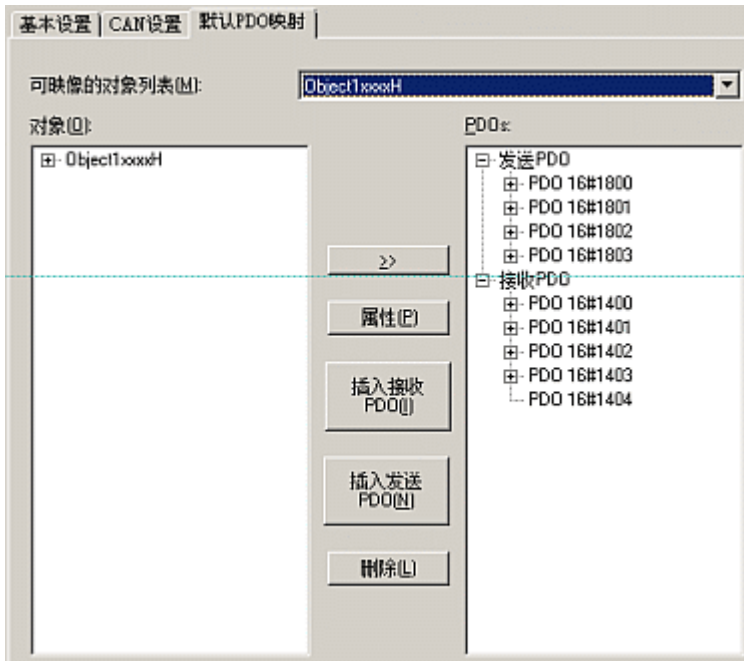
**注意**

在应用防护或检测信号时, 以及在创建 EDS 文件以便与 CANopen 主站集成时:

- ▶ 输入保护时间 = 0
- 输入寿命因素 = 0
- 输入检测信号时间 = 0
- > 为 CANopen 主站设置的值在配置期间传输到 CANopen 从站。 因此, CANopen 主站已为此节点安全启用保护或检测信号。

## [Default PDO mapping] 选项卡

1983



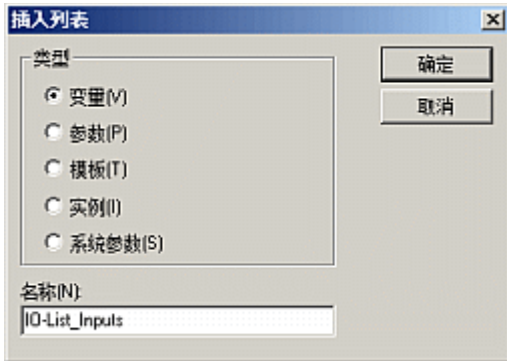
在此选项卡中，可定义 CANopen 从站传输/接收的本地对象目录（OD 编辑器）和 PDO 之间的分配。此类分配称为“映射”。

在使用的对象目录条目（变量 OD）中，在对象索引/子索引之间建立应用变量的连接。您只需确保包含多个子索引的索引的子索引 0 有关子索引数量的信息。

**例如：变量列表**

10052

在 CANopen 从站的第一个接收 PDO ( COB ID = 512 + 节点 Id ) 上, 应收到变量 PLC\_PRG.a 的数据。

**信息**

[Variables] 和 [parameters] 可选择作为列表类型。

为数据交换（例如，通过 PDO 或对象目录中的其他条目），创建一个变量列表。

如果您不想将对象目录条目与应用变量关联，则应使用参数列表。对于参数列表，当前预定义索引 1006<sub>16</sub> / 子索引 0。在此条目中，主站可输入“Com. Cycle Period”的值。这会发出不存在 SYNC 报文信号。

因此，您必须在对象目录（参数管理器）中创建一个变量列表，并将索引/子索引与变量 PLC\_PRG.a 关联。

- ▶ 为此，在变量列表中添加一行（单击鼠标右键可打开上下文菜单），并输入变量名称（任意名称）以及索引和子索引。
- ▶ 接收 PDO 的唯一允许的访问权限是 [write only]。
- ▶ 在 [variable] 列中输入“PLC\_PRG.a”或按 [F2] 并选择变量。

**注意**

CANopen 主站要读取的数据（例如，输入端、系统变量）必须具有 [read only] 访问权限。

CANopen 主站要写入的数据（例如，从站中的输出端）必须具有 [write only] 访问权限。

CANopen 主站要写入以及同时要从从站应用读取并写入到其中的 SDO 参数必须具有 [read-write] 访问权限。

要能够打开参数管理器，必须在 [Network functionality] 下的目标设置中启用参数管理器。索引/子索引区域已包含合理的值，且不得更改。

| 参数管理器          |        |         |          |             |             |
|----------------|--------|---------|----------|-------------|-------------|
|                | Name   | Index   | SubIn... | Accessright | Variable    |
| IO-List_Inputs | Input3 | 16#2000 | 16#3     | read only   | PLC_PRG.IN3 |
| RxPdoList      | Input2 | 16#2000 | 16#2     | read only   | PLC_PRG.IN2 |
| TxPdoList      | Input1 | 16#2000 | 16#1     | read only   | PLC_PRG.IN1 |
| Object_1006h   |        |         |          |             |             |
| Object1xxxxH   |        |         |          |             |             |

同步动作

在 CANopen 从站的默认 PDO 映射中，然后将索引/子索引条目作为映射条目分配给接收 PDO。通过从以下章节得知的对话框可定义 PDO 属性：**添加和配置 CANopen 从站** (→ 页 35)。

只有来自参数管理器的属性为 [read only] 或 [write only] 的对象才会在可能生成的 EDS 文件中标记为可映射 (= 可分配)，并出现在可映射对象列表中。所有其他对象在 EDS 文件中都不会标记为可映射。

### 通过主站配置更改标准映射

1984

您可在主站的某些限制内更改默认 PDO 映射 (在 CANopen 从站配置中)。

以下几点将适用：

CANopen 从站只能在标准映射 (CANopen 从站配置中的默认 PDO 映射) 已可用的对象目录中重新创建条目。

例如，对于包含默认 PDO 映射中的已映射对象的 PDO，不能在主站配置中映射第二个对象。

因此，主站配置更改的映射最多可包含标准映射中可用的 PDO。在这些 PDO 内，有 8 个映射条目 (子索引)。

可能发生的错误不显示，例如，将多余 PDO 定义 / 多余映射条目当作不存在那样处理。

在主站中，PDO 必须始终按如下方式创建：

- 从 0x1400 (接收 PDO 通信参数) 开始，或
- 从 0x1800 (传输 PDO 通信参数) 开始
- 并且一个接一个，无中断。

## 运行时访问 CANopen 从站

1985

### 设置 CANopen 从站的节点编号

19165

可以在应用程序的运行时期间，在 CANopen 从站上设置节点编号：

| 对于所有控制器 ( CR04nn, CR253n 除外 ) 适用于：     | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                         |
|--|---|
| 使用来自 CANopen 从站库的 FB CANx_SLAVE_NODEID | 来自库的 FB CANOPEN_SETSTATE<br>ifm_CANopen_NT_Vxxyyzz.LIB 面板 |

### 设置 CANopen 从站的波特率

19166

可以在应用程序的运行时期间，在 CANopen 从站上设置波特率：

| 对于所有控制器 ( CR04nn, CR253n 除外 ) 适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：                     |
|---|---|
| 从装置 ifm_CRnnnn_Vxxyyzz.LIB 库使用以下 FB 之一：<br>• CAN1_BAUDRATE 或<br>• CAN1_EXT 或<br>• CANx. | 从 ifm_CANopen_NT_Vxxyyzz.LIB 库使用 FB<br>CANOPEN_ENABLE |

### 由应用程序访问 OD 条目

19167

作为标准，对象目录中存在映射到变量（参数管理器）的条目。

但是，还有自动生成的无法通过参数管理器映射到变量的内容的 CANopen 从站条目。下面提供了这些条目：

| 对于所有控制器 ( CR04nn, CR253n 除外 ) 适用于：     | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：   |
|--|---|
| 使用来自 CANopen 从站库的 FB CANx_SLAVE_STATUS | • CANOPEN_READOBJECTDICT<br>• CANOPEN_WRITEOBJECTDICT<br>来自库 ifm_CANopen_NT_Vxxyyzz.LIB |

### 运行时更改 PDO 属性

1988

如果要在运行时更改 PDO 的属性，则按 CANopen 所述由另一个节点通过 SDO 写访问权限来完成。

作为替代方案，可以直接编写新的属性，例如，发的 PDO 的“event time”，然后将命令“StartNode-NMT”传输到节点，尽管它已启动。因此，装置会在对象目录中重新解释值。

## 通过应用程序发送紧急报文

19168

从应用程序发送紧急报文：

- 对于所有控制器 ( CR04nn , CR253n 除外 ) 适用于：

|                                  |   |
|----------------------------------|---|
| <b>CANx_SLAVE_EMCY_HANDLER</b>   | 处理 CAN 接口 x 上 CANopen 从站的装置特定错误状态：<br>• 错误寄存器 ( 索引 0x1001 ) 和<br>• CANopen 对象目录的错误场 ( 索引 0x1003 )<br>x = 1...n = CAN 接口编号 ( 视装置而定， → 技术资料 ) |
| <b>CANx_SLAVE_SEND_EMERGENCY</b> | 发送 CAN 接口 x 上 CANopen 从站的应用特定错误状态<br>x = 1...n = CAN 接口编号 ( 视装置而定， → 技术资料 )   |

- 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：

|                                 |   |
|---------------------------------|---|
| <b>CANOPEN_GETERRORREGISTER</b> | = 获取 CANopen 错误寄存器<br>从控制器读取错误寄存器 0x1001 和 0x1003<br>寄存器可通过设置相应的输入端复位。    |
| <b>CANOPEN_GETEMCYMESSAGES</b>  | = 获取 CANopen 紧急报文<br>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。<br>该列表可通过设置相应的输入端复位。 |
| <b>CANOPEN_SENDEMCMYMESSAGE</b> | = CANopen 发送紧急报文<br>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                 |

### 3.3.5 CANopen 表

| 内容                  |    |
|---------------------|----|
| CANopen 报文的结构 ..... | 63 |
| 启动报文 .....          | 68 |
| 网络管理 (NMT) .....    | 69 |

9941

以下表告知您有关 CANopen 接口的重要值和设置的信息。

#### CANopen 报文的结构

| 内容                  |    |
|---------------------|----|
| COB ID 的结构 .....    | 64 |
| 功能代码 / 预定义连接组 ..... | 65 |
| SDO 命令字节 .....      | 66 |
| SDO 中止代码 .....      | 67 |

9971

CANopen 报文由 COB ID 和高达 8 个字节的数据组成：

| COB ID |   |   | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|--------|---|---|-----|--------|--------|--------|--------|--------|--------|--------|--------|
| X      | X | X | X   | X      | X      | X      | X      | X      | X      | X      | X      |

以下章节提供详细信息。

⚠ 请注意字节顺序颠倒！ (⇒ Little Endian 或 Intel 格式)

示例：

| 值 [hex]  | 数据类型  | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|----------|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| 12       | BYTE  | 1      | 2      | -      | -      | -      | -      | -      | -      |
| 1234     | WORD  | 3      | 4      | 1      | 2      | -      | -      | -      | -      |
| 12345678 | DWORD | 7      | 8      | 5      | 6      | 3      | 4      | 1      | 2      |

### COB ID 的结构

9972

报文的第一部分是 COB ID。11 位 COB ID 的结构：

| 半字节 0 |      |   |   | 半字节 1 |       |   |   | 半字节 2 |   |   |   |
|-------|------|---|---|-------|-------|---|---|-------|---|---|---|
| 11    | 10   | 9 | 8 | 7     | 6     | 5 | 4 | 3     | 2 | 1 | 0 |
| --    | 3    | 2 | 1 | 0     | 6     | 5 | 4 | 3     | 2 | 1 | 0 |
| --    | 功能代码 |   |   |       | 节点 ID |   |   |       |   |   |   |

COB ID 包括 **功能代码 / 预定义连接组** (→ 页 65) 和节点 ID。

例如：

通信对象 = TPDO1 (TX)

装置的代码编号 = 0x020 = 32

计算：

通信对象 TPDO1 的功能代码 = 0x03

11 位 COB ID 中功能代码的意义 = 0x03 · 0x80 = 0x180

添加节点编号 (0x020) ⇒ COB ID 为：0x1A0

| 1  |          |   |   | A |            |   |   | 0 |   |   |   |
|----|----------|---|---|---|------------|---|---|---|---|---|---|
| 3  | 2        | 1 | 0 | 3 | 2          | 1 | 0 | 3 | 2 | 1 | 0 |
| 0  | 0        | 0 | 1 | 1 | 0          | 1 | 0 | 0 | 0 | 0 | 0 |
| -- | 0x03 = 3 |   |   |   | 0x020 = 32 |   |   |   |   |   |   |



## 功能代码 / 预定义连接组

9966

“CANopen 预定义连接组”中预定义了某些功能代码。

当使用预定义连接组时，您可运行高达 127 个参与方的 CANopen 网络，而无双重分配 COB ID 的风险。

广播或组播报文：

| 通信对象 | 功能代码 [hex] | COB ID [hex] | 相关参数对象 [hex]           |
|------|------------|--------------|------------------------|
| NMT  | 0          | 000          |                        |
| SYNC | 1          | 080          | 1005, 1006, 1007, 1028 |
| TIME | 2          | 100          | 1012, 1013             |

点对点报文：

| 通信对象         | 功能代码 [hex] | COB ID [hex]  | 相关参数对象 [hex] |
|--------------|------------|---------------|--------------|
| EMERGENCY    | 1          | 080 + 节点 ID   | 1014, 1015   |
| TPDO1 (TX)   | 3          | 180 + 节点 ID   | 1800         |
| RPDO1 (RX)   | 4          | 20016 + 节点 ID | 1400         |
| TPDO2 (TX)   | 5          | 280 + 节点 ID   | 1801         |
| RPDO2 (RX)   | 6          | 30016 + 节点 ID | 1401         |
| TPDO3 (TX)   | 7          | 380 + 节点 ID   | 1802         |
| RPDO3 (RX)   | 8          | 400 + 节点 ID   | 1402         |
| TPDO4 (TX)   | 9          | 480 + 节点 ID   | 1803         |
| RPDO4 (RX)   | A          | 500 + 节点 ID   | 1403         |
| 默认 SSDO (TX) | B          | 58016 + 节点 ID | 1200         |
| 默认 CSDO (RX) | C          | 60016 + 节点 ID | 1280         |
| NMT 错误控制     | E          | 70016 + 节点 ID | 1016, 1017   |

TX = 从站发送到主站

RX = 从站从主站接收

SSDO = 服务器 SDO

CSDO = 客户端 SDO

## SDO 命令字节

9968

SDO 报文的结构：

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5         | Byte 6 | Byte 7 | Byte 8 |
|--------|-----|--------|--------|--------|--------|----------------|--------|--------|--------|
| XXX    | 8   | 命令     | 索引     |        | 子索引    | 数据<br>取决于带传输数据 |        |        |        |

**!** 请注意字节顺序颠倒！ (⇒ Little Endian 或 Intel 格式)

SDO COB ID 包括：

| CANopen |                   |
|---------|-------------------|
| 节点 ID   | COB ID SDO        |
| 1...127 | TX: 0x580 + 节点 ID |
|         | 0x600 + 节点 ID     |

TX = 从站发送到主站

RX = 从站从主站接收

**!** DLC = **数据长度码** = CANopen 消息数据字节的数量。

针对 →SDO： DLC = 8

SDO 命令字节：

| 命令<br>十六进制   十<br>进制 | 报文 | 数据长度      | 描述          |
|----------------------|----|-----------|-------------|
| 21   33              | 请求 | 超过 4 个字节  | 将数据发送到从站    |
| 22   34              | 请求 | 1...4 个字节 | 将数据发送到从站    |
| 23   35              | 请求 | 4 个字节     | 将数据发送到从站    |
| 27   39              | 请求 | 3 个字节     | 将数据发送到从站    |
| 2B   43              | 请求 | 2 个字节     | 将数据发送到从站    |
| 2F   47              | 请求 | 1 个字节     | 将数据发送到从站    |
| 40   64              | 请求 | ---       | 从从站接收数据     |
| 42   66              | 响应 | 1...4 个字节 | 将数据从从站发送到主站 |
| 43   67              | 响应 | 4 个字节     | 将数据从从站发送到主站 |
| 47   71              | 响应 | 3 个字节     | 将数据从从站发送到主站 |
| 4B   75              | 响应 | 2 个字节     | 将数据从从站发送到主站 |
| 4F   79              | 响应 | 1 个字节     | 将数据从从站发送到主站 |

|    |     |    |       |   |
|----|-----|----|-------|---|
| 60 | 96  | 响应 | ---   | 数据传输支持：<br>将接收确认从从站发送到主站  |
| 80 | 128 | 响应 | 4 个字节 | 数据传输失败<br>将中止报文从从站发送到主站<br>→ 章节 <b>SDO 中止代码</b> (→ 页 <a href="#">67</a> ) |

## SDO 中止代码

9970

❗ SDO 中止代码不是紧急报文的一部分！

| 中止代码<br>[hex] | 描述                       |
|---------------|--------------------------|
| 0503 0000     | 触发位没有交替改变                |
| 0504 0000     | SDO 协议超时                 |
| 0504 0001     | 无效或未知的 client/server 命令字 |
| 0504 0002     | 无效的块大小 (仅限块模式)           |
| 0504 0003     | 无效的序号 (仅限块模式)            |
| 0504 0004     | CRC 错误 (仅限块模式)           |
| 0504 0005     | 内存溢出                     |
| 0601 0000     | 对象不支持访问                  |
| 0601 0001     | 试图读只写对象                  |
| 0601 0002     | 试图写只读对象                  |
| 0602 0000     | 对象在对象字典中不存在              |
| 0604 0041     | 对象无法映射到 PDO              |
| 0604 0042     | 要映射的对象数量和长度超出 PDO 长度     |
| 0604 0043     | 一般参数不兼容原因                |
| 0604 0047     | 一般装置内部不兼容                |
| 0606 0000     | 由于硬件错误导致访问失败             |
| 0607 0010     | 数据类型不匹配, 服务参数长度不匹配       |
| 0607 0012     | 数据类型不匹配, 服务参数长度过长        |
| 0607 0013     | 数据类型不匹配, 服务参数长度过短        |
| 0609 0011     | 子索引不存在                   |
| 0609 0030     | 超出参数的值范围 (仅限写访问)         |
| 0609 0031     | 写入的参数值过大                 |
| 0609 0032     | 写入的参数值过小                 |
| 0609 0036     | 最大值小于最小值                 |

| 中止代码<br>[hex] | 描述  |
|---------------|---|
| 0800 0000     | 一般错误  |
| 0800 0020     | 数据无法传输或存储到应用  |
| 0800 0021     | 由于本地控制导致数据无法传输或存储到应用                                  |
| 0800 0022     | 由于当前装置状态导致数据无法传输或存储到应用                                |
| 0800 0023     | 对象字典动态生成失败或对象字典不存在<br>(例如, 对象字典是通过文件生成的, 而文件错误导致生成失败) |

## 启动报文

9961

启动后, CAN 参与方会发送一次启动报文:

|      | COB ID        | DLC | Byte 1 |
|------|---------------|-----|--------|
| 十六进制 | 0x700 + 节点 ID | 0x1 | 0x00   |
| 十进制  | 1 792 + 节点 ID | 1   | 0      |

现在, 参与方能够在 CAN 网络中进行通信。

DLC = **数据长度码** = CANopen 消息数据字节的数量。

针对 →SDO: DLC = 8

### 例如:

参与方的节点 ID 为 0x7D = 125。

启动消息的 COB ID 为: 0x77D = 1 917

### 偏差:

❗ 有些装置无法发送 [0x700 + 节点 ID] (这些装置是在 CANopen 规范的版本 4 之前构建的)。

这些装置会发送以下启动报文并无状态:

|      | COB ID        | Byte 1 |
|------|---------------|--------|
| 十六进制 | 0x080 + 节点 ID | 0x0    |
| 十进制  | 128 + 节点 ID   | 0      |

## 网络管理 (NMT)

### 内容

|              |    |
|--------------|----|
| 网络管理命令 ..... | 69 |
| NMT 状态 ..... | 70 |

9974

## 网络管理命令

9962

使用以下网络管理命令，用户可以影响单个或所有 CAN 参与方的运行模式。 结构：

| COB ID | DLC | Byte 1 | Byte 2 |
|--------|-----|--------|--------|
| 0x000  | X   | 命令     | 节点 ID  |

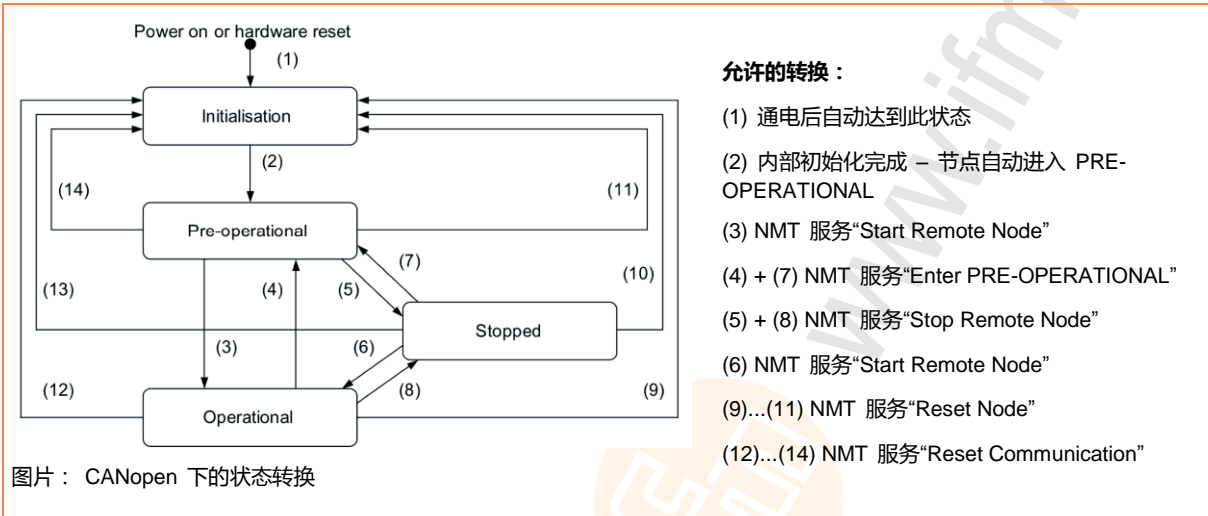
节点 ID = 00 ⇒ 命令对于网络中的所有节点同时有效

| COB ID | NMT 命令     |       | 描述                    |                           |
|--------|------------|-------|-----------------------|---------------------------|
| 0x000  | 0x01 = 01  | 节点 ID | start_remode_node     | 将节点设置为“operational”状态     |
| 0x000  | 0x02 = 02  | 节点 ID | stop_remode_node      | 将节点设置为“STOPPED”状态         |
| 0x000  | 0x80 = 128 | 节点 ID | enter_pre-operational | 将节点设置为“PRE-OPERATIONAL”状态 |
| 0x000  | 0x81 = 129 | 节点 ID | reset_node            | 节点复位                      |
| 0x000  | 0x82 = 130 | 节点 ID | reset_communication   | 复位节点的 CAN 通信              |

### NMT 状态

9963

状态字节告知 CAN 参与方的状态。



图片：CANopen 下的状态转换

© ifm electronic gmbh

### CANopen 主站的 NMT 状态

9964

**!** 这些状态显示 CANopen 主堆栈的内部状态。它们不是由 CANopen 规范定义的。

| 状态<br>十六进制   十进制 |   | 描述   |
|------------------|---|--|
| 00               | 0 | 未定义  |
| 01               | 1 | 主站等待节点的启动报文。<br>或：主站等待给定保护时间到期。  |
| 02               | 2 | <ul style="list-style-type: none"> <li>主站等待 300 ms。</li> <li>主站请求对象 0x1000。</li> <li>然后状态设置为 3。</li> </ul> |
| 03               | 3 | 主站配置其从站。为此，配置工具生成的所有 SDO 均会逐个传输到从站。然后，主站更改为状态 5。   |
| 05               | 5 | 在将所有 SDO 传输到从站之后，主站进入状态 5 并保持此状态。状态 5 是主站的正常运行状态。  |

从 FB 中读取节点状态：

| 使用的功能块             | 此处发现的节点状态       |
|--------------------|-----------------|
| CANx_MASTER_STATUS | 输出端 NODE_STATE  |
| CANOPEN_GETSTATE   | 输出端 MASTERSTATE |

### CANopen 从站的 NMT 状态

9965

| 状态<br>十六进制   十进制 |    | 描述  |
|------------------|----|---|
| FF               | -1 | 从站通过 NMT 报文“Reset Node”复位并自动进入状态 1。               |
| 00               | 0  | 未定义   |
| 01               | 1  | 状态 = 正在等待 BOOTUP<br>从站在最多 2 s 之后或收到启动报文后立即进入状态 2。 |
| 02               | 2  | 状态 = BOOTUP<br>从站在延迟 0.5 s 之后自动进入状态 3。            |

| 状态<br>十六进制   十进制 |    | 描述   |
|------------------|----|--|
| 03               | 3  | <p>状态 = PREPARED</p> <p>从站在状态 3 中配置。只要从站已收到配置工具生成的所有 SDO，它就会保持在状态 3 中。无论在从站配置期间对 SDO 传输的响应是否中止（错误），或者无论对所有 SDO 传输的响应是否无错误，这都不重要。重要的只是从站收到响应 – 而不是其内容。</p> <p>如果在配置工具中，已启用“Reset node”选项，则会在传输随后包含值“load”的对象 0x1011 子索引 1 之后执行新的节点复位。然后，通过上传对象 0x1000 再次轮询从站。</p> <p>在配置阶段期间出现问题的从站保持状态 3，或者在配置阶段之后直接进入错误状态（状态 &gt; 5）。</p> |
| 04               | 4  | <p>状态 = PRE-OPERATIONAL</p> <p>节点始终进入状态 4，以下情况下除外：</p> <ul style="list-style-type: none"> <li>它是“可选”从站，且在总线上被检测为不可用（轮询对象 0x1000），或者：</li> <li>从站存在，但使用非配置工具预期的低 16 位类型对对象 0x1000 轮询作出反应。</li> </ul>  |
| 05               | 5  | <p>状态 = OPERATIONAL</p> <p>状态 5 是从站的正常运行状态：[Normal Operation]。</p> <p>如果主站配置为 [Automatic startup]，则从站在状态 4 下启动（即，生成“启动节点”NMT 报文），且从站自动进入状态 5。</p> <p>如果设置了标记 GLOBAL_START，则主站会等到所有从站都处于状态 4。然后，通过 NMT 命令 [Start All Nodes] 启动所有从站。</p>   |
| 61               | 97 | <p>如果节点为可选（CAN 配置中的可选装置）且没有对对象 0x1000 的 SDO 轮询作出反应，则该节点会进入状态 97。</p> <p>如果从站连接到网络，并在稍后实时检测到，它会自动启动。为此，您必须在主站的 CAN 参数中已选中 [Automatic startup] 选项。</p>   |
| 62               | 98 | <p>如果装置类型（对象 0x1000）没有对已配置的类型作出响应，则节点进入状态 98。</p>  |
| 63               | 99 | <p>如果节点保护超时，则从站设置为状态 99。</p> <p>只要从站再次对节点保护请求作出响应，且选项 [Automatic startup] 启用，则主站会自动启动它。根据对节点保护请求的响应中包含的状态，新配置或仅启动节点。</p> <p>要手动启动从站，采用方法 [NodeStart] 足以。</p>   |

在以下情况下，将节点保护报文传输到从站...

- 如果从站处于状态 4 或更高，以及
- 如果配置了节点保护。



从 FB 中读取节点状态：

| 使用的功能块                                  | 此处发现的节点状态      |
|---|----------------|
| CANx_MASTER_STATUS<br>CANx_SLAVE_STATUS | 输出端 NODE_STATE |
| CANOPEN_GETSTATE                        | 输出端 NODESTATE  |

### 节点的 CANopen 状态

1973

基于 CANopen 的节点状态（通过这些值，状态还可通过相应报文中的节点编码）。

| 状态<br>十六进制   十<br>进制 | CANopen 状态：     | 描述                               |
|----------------------|-----------------|----------------------------------|
| 00   0               | BOOTUP          | 节点的 BOOTUP 报文                    |
| 04   4               | STOPPED         | 节点处于 STOPPED 状态。无数据交换，而且也无法配置节点。 |
| 05   5               | OPERATIONAL     | 节点处于 OPERATIONAL 状态并参与正常数据交换。    |
| 7F   127             | PRE-OPERATIONAL | 节点处于 PRE-OPERATIONAL 状态并可由主站配置。  |

如果节点保护处于活动状态：最高有效状态位在报文之间切换。

从功能块读取节点状态：

| 使用的功能块             | 此处发现的节点状态                              |
|--------------------|--|
| CANx_MASTER_STATUS | 来自阵列 NODE_STATE_SLAVE 的结构元素 LAST_STATE |
| CANx_SLAVE_STATUS  | 输出端 NODE_STATE                         |
| CANOPEN_GETSTATE   | 输出端 LASTNODESTATE                      |

## 3.4 CANopen 网络变量

### 内容

|                       |    |
|-----------------------|----|
| 一般信息 .....            | 74 |
| CANopen 网络变量的配置 ..... | 75 |
| 网络变量的特殊性 .....        | 82 |

1868

### 3.4.1 一般信息

2076

**CAN 网络变量**是一个用于在两个或多个控制器之间交换数据的选项。对于用户，该机制应易于使用。目前，网络变量基于 CAN 和以太网 (UDP/IP) 实现。


变量值基于广播报文自动交换。

- 在 UDP 中，作为广播报文，
- 在 CAN 中，作为与 CANopen 的 PDO 相当的报文。

根据协议，这些服务为未确认的数据传输：

它不会检查接收方是否收到报文。

网络变量的交换“1 对 n 连接”（1 个传送方对 n 个接收方）。

 CANopen 对象目录是另一个用于交换变量的选项。这是使用确认协议的 1 对 1 连接。用户可以检查报文是否到达接收方。交换不会自动执行，但是通过来自应用程序的 FB 调用执行。

→ 章节 **CANopen 主站的对象目录** (→ 页 [48](#))

### 3.4.2 CANopen 网络变量的配置

#### 内容

|                 |    |
|-----------------|----|
| 目标设置中的设置.....   | 76 |
| 全局变量列表中的设置..... | 77 |

1869

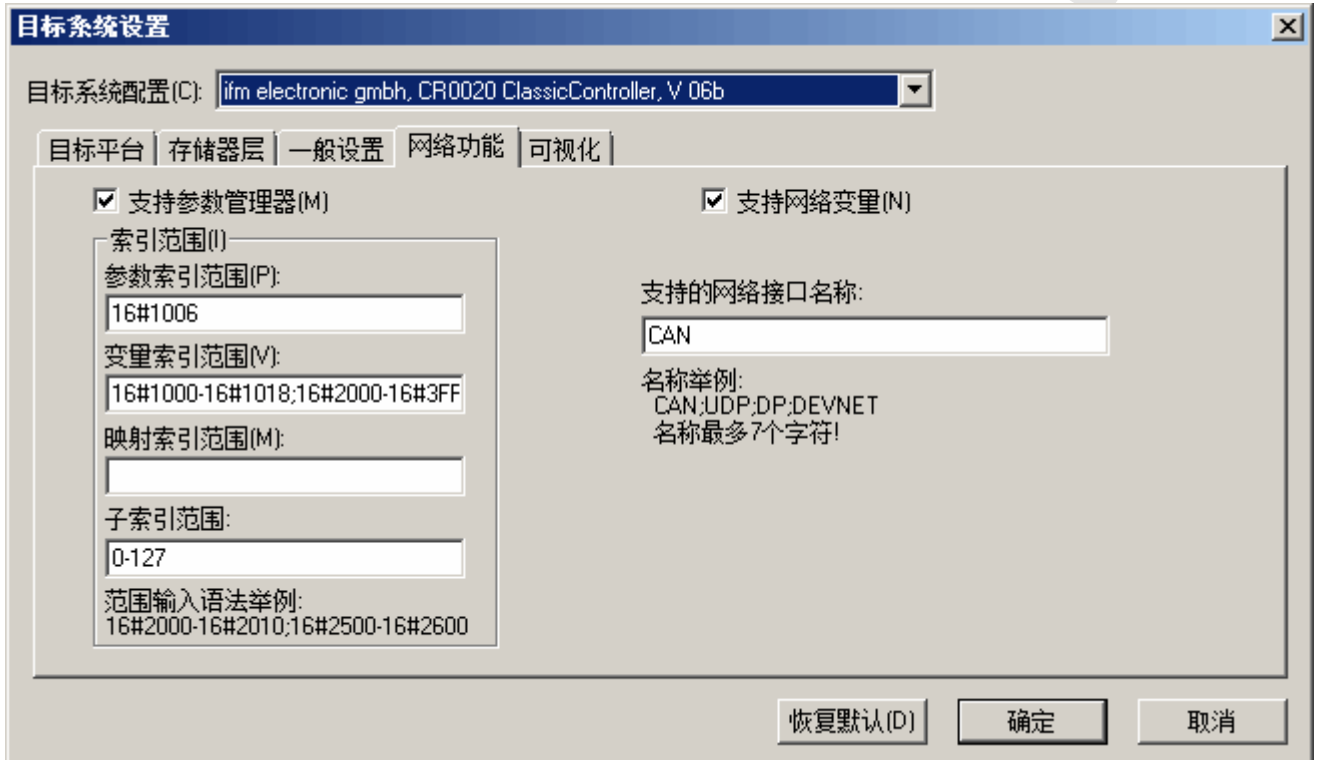
要将网络变量与 CODESYS 一起使用，您需要以下库：

- 3s\_CanDrv.lib
- 3S\_CANopenManager.lib
- 3S\_CANopenNetVar.lib
- SysLibCallback.lib。

CODESYS 会在循环开始和结束时自动生成所需初始化代码，以及网络块调用。

## 目标设置中的设置

1994



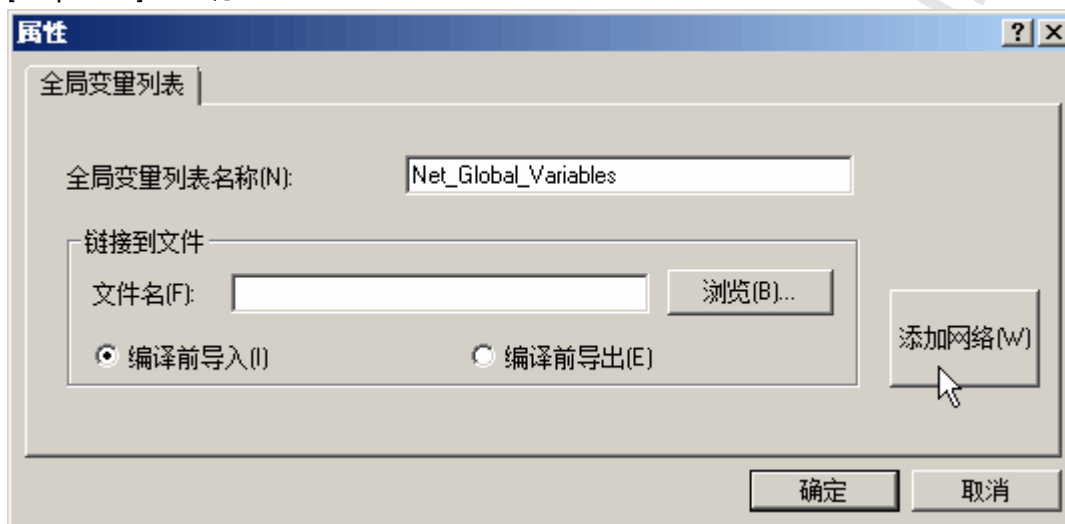
例如：ClassicController CR0020 的目标设置

- ▶ 选择 [Target settings] 对话框。
- ▶ 选择 [Network functionality] 选项卡。
- ▶ 选中 [Support network variables] 复选框。
- ▶ 在 [Names of supported network interfaces] 中输入请求的网络的名称（本例中为 CAN）。
- ▶ 要使用网络变量，您还必须向控制器配置中添加一个 CANopen 主站或 CANopen 从站（装置）。
- ▶ 请注意使用相应装置类型的网络变量时的特殊性。  
→ 章节 **网络变量的特殊性** (→ 页 82)

### 全局变量列表中的设置

1995

- ▶ 创建新的全局变量列表。在此列表中定义要与其他控制器交换的变量。
- ▶ 使用 [Object Properties] 菜单打开对话框。
- > [Properties] 窗口将显示：



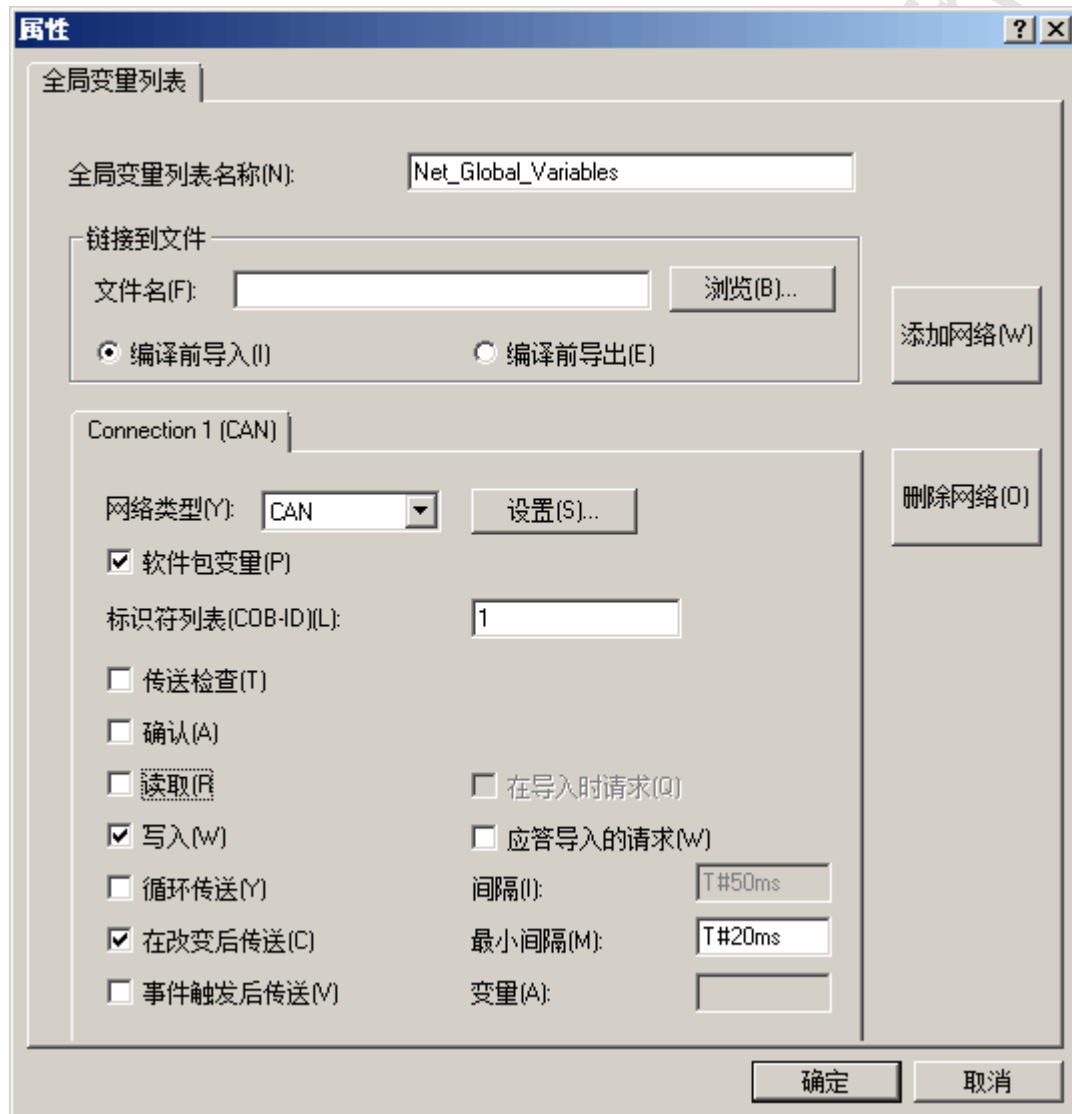
© ifm electronic gmbh

如果您想要定义网络属性：

- ▶ 单击 [Add network] 按钮。

如果您已配置多个网络连接，您也可在此处为每个变量列表配置多个连接。

- > [Properties] 窗口扩展如下：



各个选项的含义：

### 全局变量列表：网络类型

作为网络类型，您可输入目标设置中指示的其中一个网络名称。

如果您单击它旁边的 [Settings] 按钮，则可选择 CAN 接口：

1. CAN 接口：值 = 0
  2. CAN 接口：值 = 1
- 等等。

### 全局变量列表：包变量

10056

如果使用 [v] 启用此选项，则会将变量组合到一个传输单元中（若可能）。对于 CAN，传输单元的大小为 8 个字节。

如果一个传输单元中无法包括列表中的所有变量，则会为此列表生成多个传输单元。

如果未启用此选项，则每个变量都具有其自己的传输单元。

如果配置了 [Transmit on change]，则会单独检查每一个传输单元是否已更改并且必须重新传输。

### 全局变量列表：列表标识符 (COB-ID)

10057

基本标识符用作交换不同项目的变量列表的唯一标识。交换具有相同基本标识符的变量列表。

▶ 请注意，具有相同基本标识符的变量列表的定义在不同的项目中匹配。

#### ⚠注意

在 CAN 网络中，COB ID 直接用作 CAN 报文的标识符。不会检查该标识符是否也用于其余 CAN 配置。

为确保在两个控制器之间正确交换数据，两个项目中的全局变量列表必须匹配。您可使用 [Link to file] 功能实现此目标。一个项目可在编译前导出变量列表文件，其他项目应在编译之前导入此文件。

除简单的数据类型外，变量列表也可包含结构和阵列。这些组合数据类型的元素单独传输。

▶ 字符串不能通过网络变量传输！  
否则，将发生运行时错误，并启用监视器。

如果变量列表大于相应网络的 PDO，则数据将被拆分为多个 PDO。因此，无法确保在一个周期中接收变量列表的所有数据。部分变量列表可在不同的周期中接收。具有结构和阵列类型的变量也可能如此。

### 全局变量列表：传输校验和

10058

不支持此选项。

### 全局变量列表：确认

10059

不支持此选项。

### 全局变量列表：读取

10060

读取一个（或多个）控制器的变量值。

© ifm electronic gmbh



www.ifm.com



## 全局变量列表：写入

10061

将此列表的变量传输到其他控制器。

### ⓘ 注意

您仅应为每一个变量列表选择其中一个选项，例如，只读或只写。

如果您想要读取或写入某个项目的多个变量，请使用多个变量列表（一个用于读取，一个用于写入）。

要为两个参与方之间的通信获取相同的数据结构，您应将变量列表从一个控制器复制到另一个控制器。

在一个网络中，仅应在两个参与方之间交换相同的变量列表。

## 全局变量列表：周期性传输

10062

仅当启用 [write] 时才有效。以指定 [interval] 传输值，而无论值是否已更改。

## 全局变量列表：更改时传输

10063

仅当其中一个变量值已更改时才传输值。使用 [Minimum gap]（值 > 0）可定义报文包之间的最短时间。

## 全局变量列表：发生事件时传输

13327

如果选择此选项，则仅在指示的二进制 [variable] 设置为 TRUE 时，才传输 CAN 报文。无法通过输入帮助从已定义变量列表中选择此变量。

### 3.4.3 网络变量的特殊性

11811

以下接口上支持网络变量：

- CAN 1 (value = 0)
- CAN 2 (值 = 1)
- CAN 3 (值 = 2)
- CAN 4 (值 = 3)

11818

#### ⓘ 注意

- ▶ 以十进制值形式输入网络变量和接收 PDO 的标识符！

### 3.5 CAN / CANopen / 网络变量摘要

7946

- 网络变量的 COB ID 必须不同于控制器配置中的 CANopen 从站 ID 以及收发块的 ID !
- 如果一个 COB ID 中放入超过 8 个字节的网络变量, CODESYS 会将数据包自动扩展为多个连续 COB ID。这可导致与手动定义的 COB ID 冲突!
- 网络变量无法传输任何字符串变量。
- 在以下情况下可传输网络变量...
  - 如果变量变为 TRUE (事件),
  - 如果网络变量发生数据更改, 或
  - 周期性计时器间隔已过时
- 间隔时间是选择周期性传输后两次传输之间的时间。如果变量经常变化, 则最短距离是两次传输之间的等待时间。
- 为降低总线负载, 通过网络变量拆分报文, 或使用多个事件将 CANx\_TRANSMIT 拆分为多个 PLC 周期。
- 在控制器配置中, [Com Cycle Period] 和 [Sync. Window Length] 的值应该相同。
- 如果为从站选择 [Com Cycle Period], 则从站会恰好在这段时间内搜索主站的同步对象。这是为什么 [Com Cycle Period] 的值必须高于 [Master Synch Time] 的原因。
- 我们建议为从站选择“可选启动”, 为网络选择“自动启动”。这会降低不必要的总线负载, 并允许短暂丢失的从站再次集成到网络中。
- 我们建议将模拟输入端设置为“同步传输”以避免总线过载。
- 二进制输入端 (尤其是不定期切换的输入端) 应最好使用事件计时器设置为“异步传输”。
- 在监控从站状态期间要考虑的问题:
  - 在从站启动后, 从站正常运行需要一段时间。
  - 当系统关闭时, 从站可指示由于早期电压损失导致的错误状态更改。

### 3.6 用于驱动工程的 CAN

#### 内容

|                        |    |
|------------------------|----|
| 按照 SAE J1939 的标识符..... | 85 |
| 例如：详细报文文档.....         | 86 |
| 例如：短报文文档.....          | 87 |

7678

采用标准 J1939，SAE 为用户提供用于驱动工程的 CAN 总线协议。CAN 报文使用 29 位标识符传送。由于标识符较长，许多报文可直接分配给标识符。

编写协议时利用了此优势且某些报文合并 ID 组中。ID 分配在标准 SAE J1939 和 ISO 11992 中指定。ISO 11992 的协议基于 SAE J1939 的协议。

| 标准        | 应用范围     |
|-----------|----------|
| SAE J1939 | 发动机管理    |
| ISO 11992 | 牵引车和挂车接口 |

就软件协议而言，这两个标准不同，因为 ISO 11992 基于 SAE J1939。但是，在硬件接口方面，有一点差异：ISO 11992 的电压水平更高。

**!** 要使用 SAE J1939 / ISO 11992 的功能，肯定需要组合装置制造商（例如，发动机、传动装置）的协议描述。对于总控制装置中实现的报文，必须使用此描述，因为不是每一个制造商都会实现所有报文，或者实现并非对所有组合装置都有用。

要为 SAE J1939 的功能开发程序，应具有以下信息和工具：

- 组合装置要使用的数据列表
- 组合装置制造商的概述列表以及所有相关数据
- 具有 29 位支持的 CAN 监视器
- 如果需要，标准 SAE J1939

### 3.6.1 按照 SAE J1939 的标识符

7675

对于使用 SAE J1939 进行的数据交换，29 位标识符是决定因素。此标识符见如下示意图：

|          |             |          |        |        |                      |        |        |        |        |        |             |             |             |                                 |          |        |        |        |        |        |        |        |   |   |   |             |   |   |   |   |   |   |   |   |   |
|----------|-------------|----------|--------|--------|----------------------|--------|--------|--------|--------|--------|-------------|-------------|-------------|---------------------------------|----------|--------|--------|--------|--------|--------|--------|--------|---|---|---|-------------|---|---|---|---|---|---|---|---|---|
| <b>A</b> | S<br>O<br>F | 标识符 11 位 |        |        |                      |        |        |        |        |        |             |             | S<br>R<br>R | I<br>D<br>E                     | 标识符 18 位 |        |        |        |        |        |        |        |   |   |   | R<br>T<br>R |   |   |   |   |   |   |   |   |   |
| <b>B</b> | S<br>O<br>F | 优先级      | R      | D<br>P | PDU 格式 (PF)<br>6+2 位 |        |        |        |        |        | S<br>R<br>R | I<br>D<br>E | 静止<br>PF    | PDU 细节 (PS)<br>目标地址<br>组外部变量或专用 |          |        |        |        |        | 源地址    |        |        |   |   |   | R<br>T<br>R |   |   |   |   |   |   |   |   |   |
| <b>C</b> |             | 1        | 3      | 2      | 1                    | 1      | 1      | 8      | 7      | 6      | 5           | 4           | 3           | 1                               | 1        | 1      | 1      | 1      | 1      | 1      | 2      | 2      | 2 | 2 | 2 | 2           | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| <b>D</b> | -           | 2<br>8   | 2<br>7 | 2<br>6 | 2<br>5               | 2<br>4 | 2<br>3 | 2<br>2 | 2<br>1 | 2<br>0 | 1<br>9      | 1<br>8      | -           | -                               | 1<br>7   | 1<br>6 | 1<br>5 | 1<br>4 | 1<br>3 | 1<br>2 | 1<br>1 | 1<br>0 | 9 | 8 | 7 | 6           | 5 | 4 | 3 | 2 | 1 | 0 | - |   |   |

说明：

A = CAN 扩展报文格式

B = J1939 报文格式

C = J1939 报文位位置

D = CAN 29 位 ID 地址

SOF = 帧起始 (Start of frame, SOF)

SRR = 替代远程请求 (Substitute remote request, SRR)

IDE = 标识符扩展标记 (Identifier extension flag, IDE)

RTR = 远程传输请求 (Remote transmission request, RTR)

PDU = 协议数据单元 (Protocol Data Unit, PDU)

PGN = 参数组编号 (Parameter Group Number, PGN) = PDU 格式 (PF) + PDU 源 (PS)

(→ **COB-ID**)

为此，基本上要遵守 SAE J1939 的 3 种通信方式：

- 与 PDU1 ( PDU 格式 0...239 ) 的目标特定通信
- 与 PDU2 ( PDU 格式 240...255 ) 的广播通信
- 与 PDU1 或 PDU2 的专用通信

### 3.6.2 例如：详细报文文档

7679

ETC1: 电子变速器控制器 #1 (3.3.5)

值 = 0x0CF00203

产生以下细节：

| 名称          | 参数         | 上述示例中的值                           |
|-------------|------------|-----------------------------------|
| 传输重复率       | RPT        | 10 ms                             |
| 数据长度        | LEN        | 8 字节                              |
| PDU 格式      | PF         | 240                               |
| PDU 细节      | PS         | 2                                 |
| 默认优先级       | PRIO       | 3                                 |
| 数据页         | PG         | 0                                 |
| 源地址<br>目标地址 | SA<br>DA   | 3                                 |
| 参数组编号       | PGN        | 0x00F002                          |
| 标识符         | ID         | 0x0CF00203                        |
| 数据场         | SRC<br>DST | 阵列地址<br>(数据字节 1...8 的含义 à 制造商的文档) |

比如已经准备好所有相关数据的制造商，可将其直接传输到功能块。

根据所需功能设置相应值。对于场 SA / DA 或 SRC / DST，含义（而不是值）根据接收或传输功能改变。

单个数据字节必须从阵列读取并根据其含义处理。

### 3.6.3 例如：短报文文档

7680

但是，即使组合装置制造商仅提供短文档，也只能从标识符衍生 FB 参数。除了 ID 之外，“传输重复率”和数据场的含义也始终需要。

如果协议报文不是制造商特定，则标准 SAE J1939 或 ISO 11992 也可用作信息源。

标识符 0x0CF00203 的结构：

| DATA | PRI0、保留、 PG | PF  | PS | SA / DA |
|------|-------------|-----|----|---------|
| 十六进制 | 0C          | F0  | 02 | 03      |
| 十进制  | → 下表        | 240 | 2  | 3       |

由于这些值是有时需要单个位的十六进制数字，因此必须进一步分解这些数字：

| DATA | PRI0、保留、 PG |
|------|-------------|
| 十六进制 | 0C          |
| 二进制  | 0000 1100   |

被拆分为：

| DATA | 无关  | PRI0 | 已保留 | PG |
|------|-----|------|-----|----|
| 二进制  | 000 | 011  | 0   | 0  |
| 十进制  | --- | 3    | 0   | 0  |

## “PRIO、保留., PG”的其他典型组合

0x18:

| DATA | PRIO、保留、 PG |
|------|-------------|
| 十六进制 | 18          |
| 二进制  | 0001 1000   |

被拆分为：

| DATA | 无关  | PRIO | 已保留 | PG |
|------|-----|------|-----|----|
| 二进制  | 000 | 110  | 0   | 0  |
| 十进制  | --- | 6    | 0   | 0  |

0x1C:

| DATA | PRIO, reserv., PG |
|------|-------------------|
| 十六进制 | 1c                |
| 二进制  | 0001 1100         |

被拆分为：

| DATA | 无关  | PRIO | 已保留 | PG |
|------|-----|------|-----|----|
| 二进制  | 000 | 111  | 0   | 0  |
| 十进制  | --- | 7    | 0   | 0  |



### 3.7 CAN / CANopen: 错误和错误处理

#### 内容

|                  |    |
|------------------|----|
| CAN 错误 .....     | 90 |
| CANopen 错误 ..... | 93 |

1171

所述的错误机制由集成在控制器中的 CAN 控制器自动处理。这不能受用户影响。（根据应用）编程人员应对应用软件中已报告的错误作出反应。

CAN 错误机制的目标：

- 确保整个 CAN 网络中的数据对象一致
- 网络的永久功能，即使在 CAN 参与方故障时也不受影响
- CAN 参与方的临时干扰和永久干扰之间的差异
- 采用 2 步实现故障参与方的定位和自禁用：
  - 被动错误
  - 从总线断开连接（总线离线）

这为暂时干扰的参与方提供“休息”。

为了在发生错误时为感兴趣的用户提供 CAN 控制器的行为概述，下面简单明了地描述了错误处理。在检测到错误之后，会自动准备信息并在应用软件中作为 CAN 错误位提供给编程人员。

### 3.7.1 CAN 错误

8589

#### 错误报文

1172

如果总线参与方检测到错误状态，它会立即传输错误报文。然后，中止传输或拒绝其他参与方已收到的正确报文。这会确保为所有参与方提供正确、一致的数据。由于错误报文是直接传输的，发送方可立即开始重复干扰的报文，这与其他现场总线系统（它们会等到定义的确证时间已过）不同。这是 CAN 最重要的功能之一。

串行数据传输的一个基本问题是永久干扰或故障的总线参与方可拦截整个系统。CAN 的错误处理可增加此类风险。为排除这种情况，需要一种机制来检测参与方的故障，并在必要时将此参与方从总线断开。

#### 错误计数器

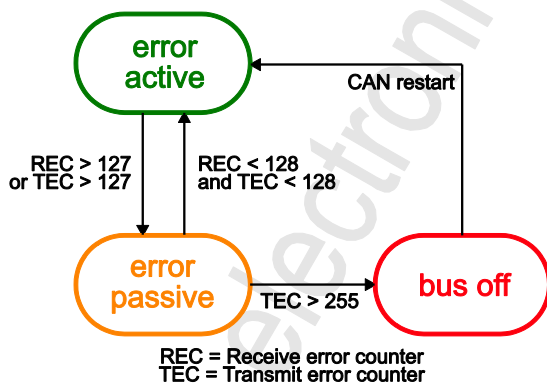
1173

传输和接收错误计数器集成在 CAN 控制器中。它们的数值随每一次故障传输或接收操作递增。如果传输正确，这些计数器的数值会递减。

但是，错误计数器在发生错误时的数值递增大于成功时的数值递减。在定义的期限内，这可导致计数大量增加，即使不受干扰的报文数量大于受干扰的报文数量。不受干扰的周期延长会慢慢降低计数。因此，这些计数表示受干扰报文的相对频率。

如果参与方本身是最早检测到错误的（= 自己造成的错误），则此参与方因此错误受到的“惩罚”比其他总线参与方更严重。为此，计数器递增数值更大。

如果参与方的计数超出定义的值，则可以假定此参与方故障。为防止此参与方通过主动错误报文（主动错误）干扰总线通信，将其切换到“被动错误”。



图：错误计数器的机制

主动错误

→ 参与方，主动错误 (→ 页 91)

被动错误

→ 参与方，被动错误 (→ 页 91)

总线断开

→ 参与方，总线断开 (→ 页 92)

CAN 重启

→ 参与方，总线断开

## 参与方，主动错误

1174

主动错误参与方参与总线通信不受限制，并被允许通过传输主动错误报文报告检测到的错误。如前所述，传输的报文被销毁。

## 参与方被动错误

19173

被动错误参与方仍可无限制地进行通信。然而，仅允许通过被动错误标记标识检测到的错误，这不干扰总线通信。如果处于定义的计数值之下，被动错误参与方在此变为主动错误。

关于应用程序中的反应：

| 对于所有控制器 (CR04nn, CR253n 除外) 适用于：                  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：  |
|---|--|
| 如果错误计数器值 $\geq 96$ ：<br>系统变量 CANx_WARNING = TRUE。 | 如果超出 TX 错误计数器的警告阈值：<br>在 FB CAN_STATUS 中，输出端 WARNING_TX 变为 = TRUE<br>如果超出 RX 错误计数器的警告阈值：<br>在 FB CAN_STATUS 中，输出端 WARNING_RX 变为 = TRUE |

在此状态下，参与方为被动错误。

## 参与方总线断开

19174

如果错误计数值继续增大，参与方会在超出最大计数值之后从总线断开（总线断开）。

关于应用程序中的反应：

| 对于所有控制器（CR04nn，CR253n 除外）适用于：  | 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：  |
|--|--|
| 系统变量 CANx_BUSOFF = TRUE  | 如果超出 TX 错误计数器的上限：<br>在 FB CAN_STATUS 中，输出端 BUSOFF 变为 = TRUE<br>如果超出 RX 错误计数器的上限：<br>在 FB CAN_STATUS 中，输出端 BUSOFF 变为 = TRUE |
| > 错误 CANx_BUSOFF 会由运行时系统自动处理和复位。<br>如果要通过应用程序进行准确的错误处理和评估：<br>▶ 使用 FB CANx_ERRORHANDLER！ | > 错误 BUSOFF 会由运行时系统自动处理（恢复）。尝试重新启动相应的 CAN 接口，每秒最多 4 次。<br>> 如果在第 4 次尝试之后总线断开未修复，则装置本身会从接口断开并不再参与总线流量。                      |
| ▶ 始终在应用程序中复位总线错误指示：  | ▶ 在 FB CAN_STATUS 中设置输入端 CLEAR = TRUE。<br>> 错误指示即会复位。<br>如果这些错误最初不再在下一个周期设置：<br>> 总线修复<br>> 设备再次 ERROR ACTIVE，即它通常参与总线通信。  |
| ▶ 在应用程序中显式复位错误 CANx_BUSOFF！  |  |

### 3.7.2 CANopen 错误

**内容**

EMCY 报文的结构..... 93  
 制造商特定信息 ..... 99

11670

#### EMCY 报文的结构

**内容**

以下错误有所不同 : ..... 94  
 紧急报文 ..... 94  
 标识符 ..... 94  
 EMCY 错误代码 ..... 95  
 CANopen 错误代码概述..... 95  
 对象 0x1001 ( 错误寄存器 ) ..... 97  
 对象 0x1003 ( 错误场 ) ..... 97  
 发出装置错误信号..... 98

19177

在 CANopen 下，错误状态通过简单的标准化机制通知。对于 CANopen 装置，每次发生错误都会通过详细描述错误的特殊报文进行通知。

如果错误或其原因为在某个时间之后消失，此事件也会通过 EMCY 报文进行通知。最近发生的错误存储在对象目录（对象 0x1003）中，并且可通过 SDO 访问读取（→ **CANx\_SDO\_READ**）。此外，当前错误情况在错误寄存器（对象 0x1001）中反映。

通过 SDO 访问读取错误：

- 对于所有控制器（CR04nn，CR253n 除外）适用于：

|                      |   |
|----------------------|---|
| <b>CANx_SDO_READ</b> | CAN interface x: 读取 SDO，所示索引来自节点。<br>x = 1...n = CAN 接口编号（视装置而定，→ 技术资料） |
|----------------------|---|

- 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：

|                             |   |
|-----------------------------|---|
| <b>CANOPEN_SDOREAD</b>      | = CANopen 读取 SDO<br>读取 "Expedited SDO" = 加速服务数据对象 |
| <b>CANOPEN_SDOREADBLOCK</b> | = CANopen 读取 SDO 块<br>通过 SDO 块传输读取网络节点对象目录中所示的条目  |
| <b>CANOPEN_SDOREADMULTI</b> | = CANopen 读取 SDO multi<br>读取网络节点对象目录中所示的条目        |

**以下错误有所不同：**

8046

**通信错误**（例如：）

- CAN 控制器发出 CAN 错误信号。  
（频繁发生此错误表示存在物理问题。 这些错误可显著影响传输行为以及网络的数据速率。）
- 寿命保护或检测信号错误
- SYNC 错误（仅限从站）

**应用错误**（例如：）

- 短路或断线
- 温度过高

**紧急报文**

9973

从站中的装置错误或 CAN 总线中的问题会触发紧急报文：

| COB ID       | DLC | Byte 0 | Byte 1 | Byte 2       | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------------|-----|--------|--------|--------------|--------|--------|--------|--------|--------|
| 0x80 + 节点 ID |     | 错误代码   |        | 对象<br>0x1001 | 装置特定   |        |        |        |        |

**!** 请注意字节顺序颠倒！（⇒ Little Endian 或 Intel 格式）

**标识符**

8048

错误报文的标识符由以下元素的总和组成：

EMCY 默认标识符 128 (0x80)

+

节点 ID

## EMCY 错误代码

8049

提供有关所发生错误的详细信息。通信配置文件中已经定义了可能的错误代码列表。仅适用于某个装置类的错误代码在此装置类的相应装置配置文件中定义。

## CANopen 错误代码概述

545

| 错误代码 (十六进制) | 含义            |
|-------------|---------------|
| 00xx        | 复位或无错误        |
| 10xx        | 一般性错误         |
| 20xx        | 电流            |
| 21xx        | 电流, 装置输入端     |
| 22xx        | 电流, 装置内部      |
| 23xx        | 电流, 装置输出端     |
| 30xx        | 电压            |
| 31xx        | 电源电压          |
| 32xx        | 装置内部的电压       |
| 33xx        | 输出电压          |
| 40xx        | 温度            |
| 41xx        | 环境温度          |
| 42xx        | 装置温度          |
| 50xx        | 装置硬件          |
| 60xx        | 装置软件          |
| 61xx        | 内部软件          |
| 62xx        | 用户软件          |
| 63xx        | 数据集           |
| 70xx        | 其他模块          |
| 80xx        | 监控            |
| 81xx        | 通信            |
| 8110        | CAN 溢出 - 对象丢失 |
| 8120        | CAN 处于错误被动模式下 |
| 8130        | 寿命保护错误或检测信号错误 |
| 8140        | 从总线离线状态恢复     |

| 错误代码 (十六进制) | 含义            |
|-------------|---------------|
| 8150        | 传输 COB-ID 冲突  |
| 82xx        | 协议错误          |
| 8210        | PDO 因长度错误而未处理 |
| 8220        | 已超过 PDO 长度    |
| 90xx        | 外部错误          |
| F0xx        | 其他功能          |
| FFxx        | 装置特定          |



## 对象 0x1001 ( 错误寄存器 )

8547

此对象反映了 CANopen 装置的一般错误状态。如果对象 0x1001 不再发出任何错误信号，则装置视为无错误。

| 位 | 含义 (Bedeutung) |
|---|----------------|
| 0 | 一般性错误          |
| 1 | 电流             |
| 2 | 电压             |
| 3 | 温度             |
| 4 | 特性错误           |
| 5 | 装置配置文件特定       |
| 6 | 保留 – 始终为 0     |
| 7 | 制造商特定          |

对于错误报文，可同时在错误寄存器中设置多个位。

例如：Cr2033，通道 2 上的“wire break”报文（→ 装置的安装手册）：

| COB-ID       | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|--------------|-----|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x80 + 节点 ID |     | 00     | FF     | 81     | 10     | 00     | 00     | 00     | 00     |

错误代码 = 0xFF00

错误寄存器 = 0x81 = 0b1000 0001，因此它包括以下错误：

- 一般性错误
- 制造商特定

相关通道 = 0x0010 = 0b0000 0000 0001 0000 = 断线通道 2

## 对象 0x1003 ( 错误场 )

8050

对象 0x1003 代表装置的错误内存。子索引包含最近发生的触发了错误报文的错误。

如果发生新的错误，其 EMCY 错误代码会始终存储在子索引 0x01 中。所有其他更早的错误在错误内存中后移一个位置，即，子索引加 1。如果使用所有受支持的子索引，则删除最早的错误。子索引 0x00 会增加到存储的错误数。在已纠正了所有错误之后，值“0”会写入子索引 0x01 的错误场中。

要删除错误内存，值“0”可写入子索引 0x00。不能输入其他值。

## 发出装置错误信号

19178

如前所述，在装置中发生错误时会传输 EMCY 报文。与可编程装置相比，分散的输入/输出模块（例如，CompactModules CR2033）会自动传输错误报文。

相应错误代码 → 相应装置手册。

仅当应用程序中集成了以下 FB 之一时，可编程装置才会自动生成 EMCY 报文（例如，对于“short circuit on output Q07”）：

- 对于所有控制器（CR04nn，CR253n 除外）适用于：

|                                 |  |
|---------------------------------|--|
| <b>CANx_MASTER_EMCY_HANDLER</b> | 处理 CAN 接口 x 上 CANopen 主站的装置特定错误状态<br>x = 1...n = CAN 接口编号（视装置而定，→ 技术资料）  |
| <b>CANx_SLAVE_EMCY_HANDLER</b>  | 处理 CAN 接口 x 上 CANopen 从站的装置特定错误状态：<br>• 错误寄存器（索引 0x1001）和<br>• CANopen 对象目录的错误场（索引 0x1003）<br>x = 1...n = CAN 接口编号（视装置而定，→ 技术资料） |

- 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：

|                                 |   |
|---------------------------------|---|
| <b>CANOPEN_GETERRORREGISTER</b> | = 获取 CANopen 错误寄存器<br>从控制器读取错误寄存器 0x1001 和 0x1003<br>寄存器可通过设置相应的输入端复位。    |
| <b>CANOPEN_GETEMCYMESSAGES</b>  | = 获取 CANopen 紧急报文<br>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。<br>该列表可通过设置相应的输入端复位。 |

使用 CODESYS 可编程的所有 ecomatmobile 装置的自动传输的 EMCY 错误代码的概述 → 章节 **CANopen 错误代码概述** (→ 页 95).

如果应用程序要另外传输应用特定错误，则使用以下 FB 之一：

- 对于所有控制器（CR04nn，CR253n 除外）适用于：

|                                   |   |
|-----------------------------------|---|
| <b>CANx_MASTER_SEND_EMERGENCY</b> | 发送 CAN 接口 x 上 CANopen 主站的应用特定错误状态<br>x = 1...n = CAN 接口编号（视装置而定，→ 技术资料） |
| <b>CANx_SLAVE_SEND_EMERGENCY</b>  | 发送 CAN 接口 x 上 CANopen 从站的应用特定错误状态<br>x = 1...n = CAN 接口编号（视装置而定，→ 技术资料） |

- 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：

|                              |   |
|------------------------------|---|
| <b>CANOPEN_SENDEMCMESAGE</b> | = CANopen 发送紧急报文<br>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 |
|------------------------------|---|

## 制造商特定信息

8548

装置制造商可指出其他错误信息。 格式可以自由选择。

### 例如：

装置中发生两个错误，并通过总线发出信号：

#### - 输出端短路：

错误代码 0x2308，  
将值 0x03 (0b0000 0011) 输入到对象 0x1001 中  
(一般性错误和电流错误)

#### - CAN 溢出：

错误代码 0x8110，  
将值 0x13 (0b0001 0011) 输入到对象 0x1001 中  
(一般性错误、电流错误和通信错误)

#### >> CAN 溢出已处理：

错误代码 0x0000，  
将值 0x03 (0b0000 0011) 输入到对象 0x1001 中  
(一般性错误、电流错误、通信错误复位)

只能通过不再存在通信错误的此信息看到。

## CANopen EMCY 代码概述 (标准页)

18071

- 对于所有控制器 (CR04nn, CR253n 除外) 适用于:  
当周期性调用 FB CANx\_MASTER\_EMCY\_HANDLER 时, 会自动发送以下 EMCY 报文。
- 对于所有 CR04nn、CR1nnn、CR253n, 以下所述适用:  
在 CANopen 堆栈中, 这些 EMCY 代码尚没有固定的实现。建议:  
    ▮ 使用 FB CANOPEN\_SENDEMCMYMESSAGE 生成这些 EMCY 代码。

| EMCY 代码对象 0x1003 |               | 对象 0x1001     | 制造商特定信息 |         |         |         |         | 说明            |
|------------------|---------------|---------------|---------|---------|---------|---------|---------|---------------|
| 第 0 个字节 [hex]    | 第 1 个字节 [hex] | 第 2 个字节 [hex] | 第 3 个字节 | 第 4 个字节 | 第 5 个字节 | 第 6 个字节 | 第 7 个字节 |               |
| 00               | 21            | 03            |         |         |         |         |         | 断线, 输入端       |
| 08               | 21            | 03            |         |         |         |         |         | 短路, 输入端       |
| 10               | 21            | 03            |         |         |         |         |         | 过电流 0...20 mA |
| 00               | 23            | 03            |         |         |         |         |         | 断线, 输出端       |
| 08               | 23            | 03            |         |         |         |         |         | 短路, 输出端       |
| 10               | 23            | 03            |         |         |         |         |         | 过载, 输出端       |
| 00               | 31            | 05            |         |         |         |         |         | 电源电压 VBBS     |
| 00               | 33            | 05            |         |         |         |         |         | 输出电压 VBBO     |
| 08               | 33            | 05            |         |         |         |         |         | 输出电压 VBRR     |
| 00               | 42            | 09            |         |         |         |         |         | 温度过高          |

字节 3...7 的条目取决于装置的输入端和输出端的具体分布 (→ 编程手册)。

## CANopen EMCY 代码概述 (扩展页)

18072

- 对于所有控制器 (CR04nn, CR253n 除外) 适用于:  
当周期性调用 FB CANx\_MASTER\_EMCY\_HANDLER 时, 会自动发送以下 EMCY 报文。
- 对于所有 CR04nn、CR1nnn、CR253n, 以下所述适用:  
在 CANopen 堆栈中, 这些 EMCY 代码尚没有固定的实现。建议:  
    ▮ 使用 FB CANOPEN\_SENDEMCMYMESSAGE 生成这些 EMCY 代码。

| 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 |               |
| 01               | 21            | 03            |         |         |         |         |         | 断线, 输入端       |
| 09               | 21            | 03            |         |         |         |         |         | 短路, 输入端       |
| 11               | 21            | 03            |         |         |         |         |         | 过电流 0...20 mA |
| 01               | 23            | 03            |         |         |         |         |         | 断线, 输出端       |
| 09               | 23            | 03            |         |         |         |         |         | 短路, 输出端       |
| 10               | 23            | 03            |         |         |         |         |         | 过载, 输出端       |
| 10               | 33            | 05            |         |         |         |         |         | 输出电压 VBB1     |
| 11               | 33            | 05            |         |         |         |         |         | 输出电压 VBB2     |
| 12               | 33            | 05            |         |         |         |         |         | 输出电压 VBB3     |
| 13               | 33            | 05            |         |         |         |         |         | 输出电压 VBB4     |
| 18               | 33            | 05            |         |         |         |         |         | 继电器电源         |


字节 3...7 的条目取决于装置的输入端和输出端的具体分布 (→ 编程手册)。

### CANopen EMCY 代码概述 (CANx)

18073

- **对于所有控制器 ( CR04nn , CR253n 除外 ) 适用于 :**  
当周期性调用 FB CANx\_MASTER\_EMCY\_HANDLER 时, 会自动发送以下 EMCY 报文。
- **对于所有 CR04nn、CR1nnn、CR253n , 以下所述适用 :**  
在 CANopen 堆栈中, 这些 EMCY 代码尚没有固定的实现。 建议 :  
    ↳ 使用 FB CANOPEN\_SENDEMCMYMESSAGE 生成这些 EMCY 代码。

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 )  |

## 4 控制输出端 - 描述

### 内容

|                  |     |
|------------------|-----|
| PWM 功能 - 描述..... | 103 |
| 控制器 - 描述 .....   | 113 |

13741

### 4.1 PWM 功能 - 描述

#### 内容

|                     |     |
|---------------------|-----|
| PWM 信号处理 - 描述 ..... | 103 |
| 使用 PWMi 进行液压控制..... | 111 |

13831

本章节介绍有关 *ecomatmobile* 装置中的脉冲宽度调制的更多信息。

PWM 或 PWMi 的可用性：

- 装置的数据表
- 装置手册

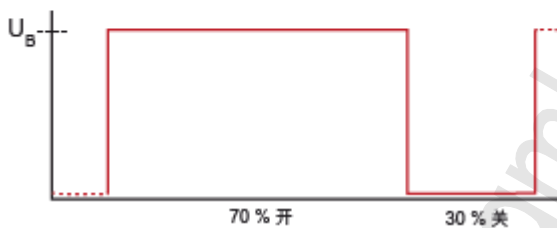
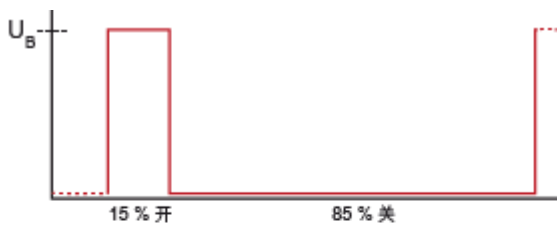
## 4.1.1 PWM 信号处理 - 描述

### 内容

|                           |     |
|---------------------------|-----|
| PWM: PWM 输出端的作用是什么？ ..... | 105 |
| PWM: 什么是抖动？ .....         | 106 |
| PWM: 功能块 .....            | 109 |
| PWM: 参数描述 .....           | 109 |

13832  
6889

缩写 PWM 代表脉冲宽度调制 (**p**ulse **w**idth **m**odulation)。它主要用于触发移动和鲁棒控制器应用的比例阀 (PWM 阀)。此外, 使用 PWM 输出端的其他组件 (附件), 可以将脉冲宽度调制的输出信号转换为模拟输出电压。



图：PWM 原理

PWM 输出信号是 GND 和电源电压之间的脉冲信号。在定义的期限 (PWM 频率) 内, 传号空号比会不同。根据传号空号比, 相连负载会确定相应的 RMS 电流。

**ecomatmobile** 控制器的 PWM 功能是由处理器提供的硬件功能。要使用控制器的集成 PWM 输出端, 必须在应用程序中对它们进行初始化并根据请求的输出信号进行参数化。



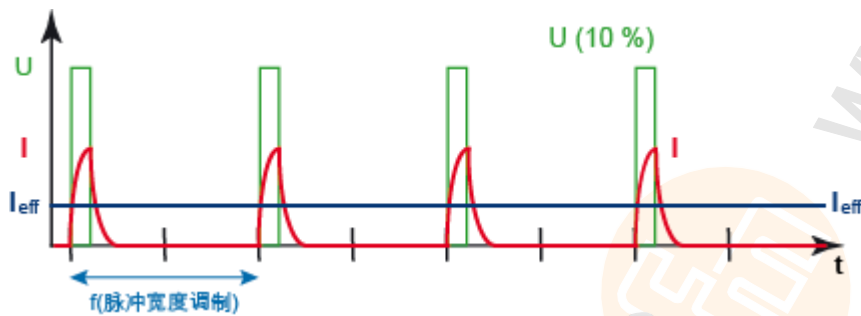
## PWM: PWM 输出端的作用是什么？

1563

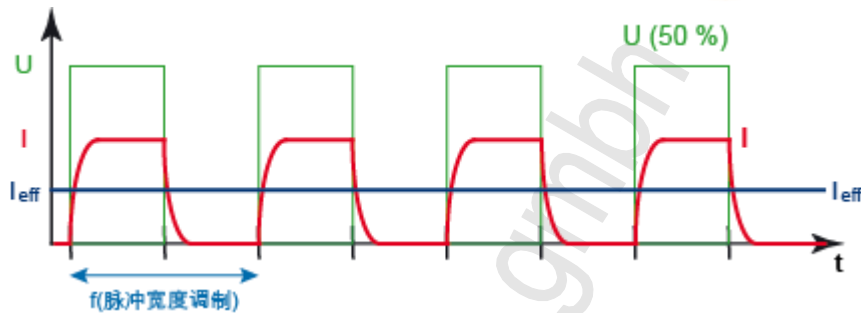
PWM 代表脉冲宽度调制 (pulse width modulation)，意味着以下原理：

通常，只要打开，数字输出端就会提供固定输出电压。在这里，输出电压的值无法更改。但是，PWM 输出端将电压分为一个快速的由多个方波脉冲串组成的序列。脉冲持续时间 [switched on] / 脉冲持续时间 [switched off] 比率决定请求的输出电压的有效值。这称为接通时间，以 [%] 为单位。

① 在下面的示意图中，电流分布显示为格式化的直线。实际上，电流呈 e 函数。



图：10 % 接通时间下，PWM 电压 U 和线圈电流 I 的分布：  
有效线圈电流  $I_{\text{eff}}$  也为 10 %



图：50 % 接通时间下，PWM 电压 U 和线圈电流 I 的分布：  
有效线圈电流  $I_{\text{eff}}$  也为 50 %



图：100 % 接通时间下，PWM 电压 U 和线圈电流 I 的分布：  
有效线圈电流  $I_{\text{eff}}$  也为 100 %

## PWM: 什么是抖动？

1564

如果控制液压比例阀，其活塞不会立即运动，而且起初与线圈电流不成比例。由于这一“滑黏效应”——一种“开启扭矩”，阀门起初需要稍微较高的电流才能产生将活塞从其关闭位置移开所需的动力。每次阀门活塞的位置改变也会发生同样的情况。此效应反映为阀门活塞的急动，尤其是在操纵速度极低的情况下。

通过使阀门活塞稍微前后运动（抖动）来从技术上解决这一问题。活塞持续振动，无法“黏住”。此外，现在也会执行微小的位置改变，无任何延迟，可以说是“快速接合”。

优点：以这种方式控制的液压缸的运动可以更敏感。

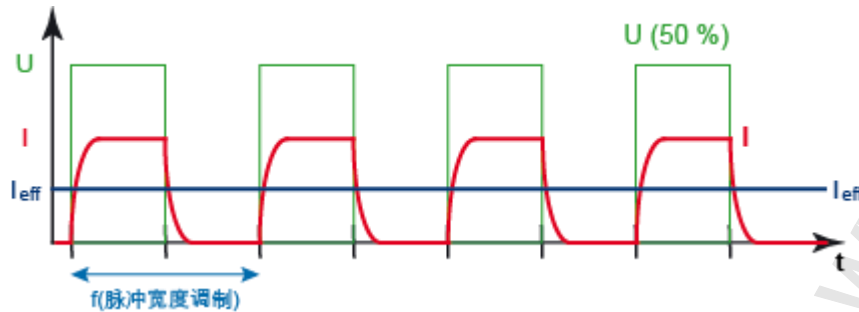
缺点：由于阀门线圈现在是持续工作，阀门在抖动时测得的温度高于不抖动时。

这意味着必须找到“黄金分割”。

## 抖动在什么时候有用？

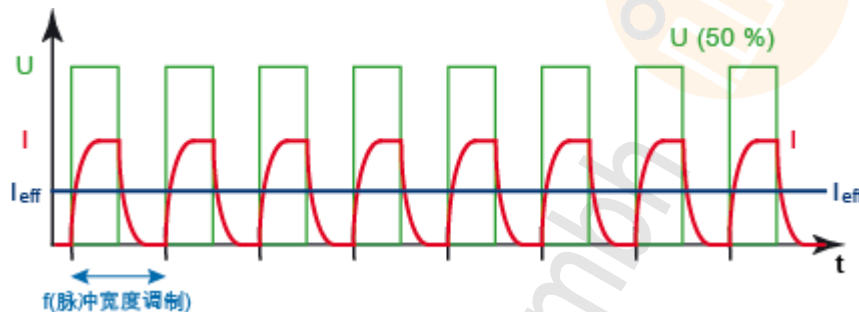
1565

当 PWM 输出端提供足够小的脉冲频率（标准值：高达 250 Hz）时，这样阀门活塞以最低冲程持续运动，因此不需要抖动（→ 下图）：

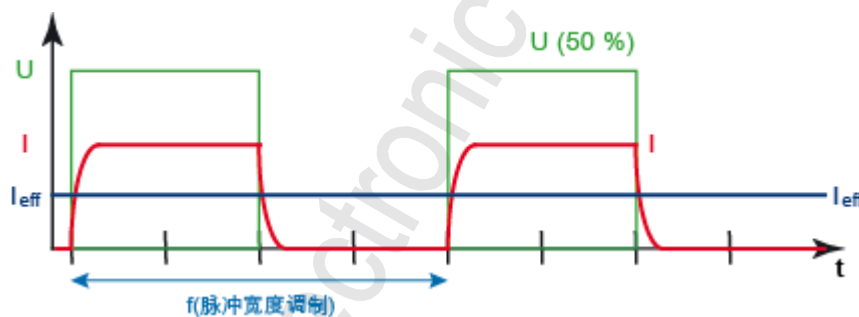


图：平衡的 PWM 信号；无需抖动。

在更高的 PWM 频率（标准值 250 Hz，最高达 1 kHz）下，阀门活塞的剩余运动较短或者较慢，因此这会有效地导致静止，这样阀门活塞必定会再次卡在其当前位置！（→ 下面的图）：



图：PWM 信号频率高会导致线圈中几乎是直流电流。阀门活塞不会再进行足够的运动。每次信号变化，阀门活塞都必须再次克服开启扭矩。



图：PWM 信号的频率过低仅允许阀门活塞进行极少的急动。每个脉冲都会将阀门活塞再次从其关闭位置移开；每一次阀门活塞都必须再次克服开启扭矩。

❗ 在 PWM 接通时间低于 10 % 和高于 90 % 时，绝对需要为 PWM 信号附加抖动信号。

## 抖动频率和幅度

1566

PWM 输出信号的传号空号比（接通时间）通过抖动频率切换。抖动频率决定两个抖动半波的接通时间差异。

❗ 抖动频率必须是 PWM 频率的整数部分。否则，液压系统不会平稳工作，而是将振荡。

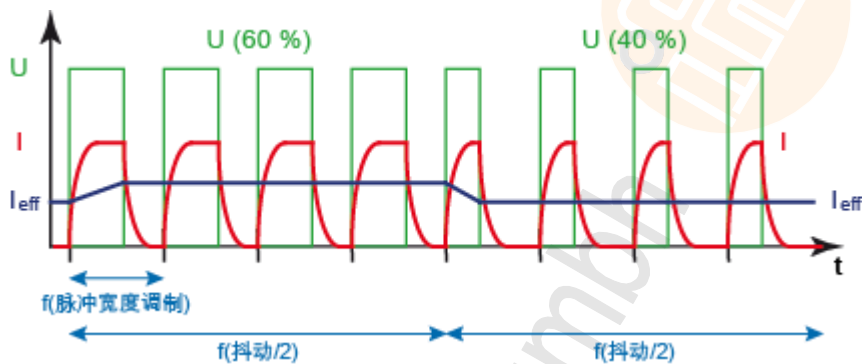
## 抖动示例

1567

抖动频率是 PWM 频率的 1/8。

抖动幅度为 10 %。

在图中 50 % 的接通时间下，4 个脉冲的实际接通时间为 60 %，接下来 4 个脉冲的实际接通时间为 40 %，这表示平均接通时间为 50 %。产生的有效线圈电流将是最大线圈电流的 50 %。



其结果是，阀门活塞始终在其关闭位置周围振荡，以准备好通过下一个信号改变进入新位置，而不必克服之前的开启扭矩。

## PWM: 功能块

19183

您可以使用以下功能块访问 PWM 兼容输出端的 PWM 功能：

❗ 此处所列的某些功能块仅可用于单个装置。

- 对于所有控制器（CR04nn，CR253n 除外）适用于：

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

- 对于所有 CR04nn、CR253n，以下所述适用：

|                 |   |
|-----------------|---|
| CURRENT_CONTROL | PWMi 输出通道的电流控制                            |
| H_BRIDGE        | PWM 通道对上的 H 桥                             |
| PWM1000         | 初始化并配置具备 PWM 功能的输出通道<br>传号空号比可按 1‰ 的步距表示  |
| PWM1000_LOW     | 初始化并配置支持 PWM 的输出通道减开关<br>可以 1‰ 的增量表示传号空号比 |

## PWM: 参数描述

13833

FB PWM... 包含一组参数。这里详细介绍了其中的一些参数。

### PWM 频率

2304

根据阀门类型，需要响应的 PWM 频率。对于 PWM1000，PWM 频率以数字值形式直接传输，单位为 [Hz]。

所有 PWM 通道的行为方式都相同。每一个 PWM 通道都可以单独设置为其自己的频率。PWM 频率位于范围 20...250 Hz 内。

### PWM 抖动

2306

对于某些液压阀类型，必须在 PWM 频率上额外加上所谓的抖动频率。如果这些阀门长期由固定 PWM 值触发，它们会由于较高系统温度而阻塞。

为防止这种情况发生，在抖动频率的基础上将 PWM 值增加或减少定义的值 (DITHER\_VALUE)。因此，在固定 PWM 值上加上具有抖动频率和幅度 DITHER\_VALUE 的振动。抖动频率必须是 PWM 频率的整数部分。

© ifm electronic gmbh

## 4.1.2 使用 PWMi 进行液压控制

### 内容

此库的用途是什么？ - 简介 ..... 111

19185  
1559

ifm electronic 为用户提供特殊功能，以使用 PWM (= PWMi) 控制作为电流调整的特殊场的液压系统

液压库仅可用于控制器，但不可用于 CR04nn、CR253n。

### 此库的用途是什么？ - 简介

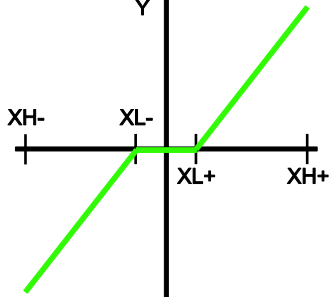
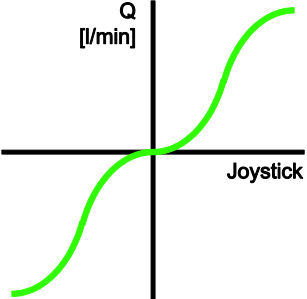
1560

多亏了此库的 FB，您可以完成以下任务：

### 标准化控制杆的输出信号

1561

控制杆的整个运动区域并非始终都会影响机器运动。

|  |   |
|--|---|
| <p>通常，控制杆空档位置周围的区域是空闲的，因为控制杆不会在此空档位置可靠地供应 0 V。<br/>在此图中，XL- 和 XL+ 之间的区域是空闲的。</p> |  |
| <p>此库的 FB 使您能够根据要求调整特性 - 根据要求，甚至是可自由配置：</p>                                      |  |

## 带有电流控制输出端的液压控制阀

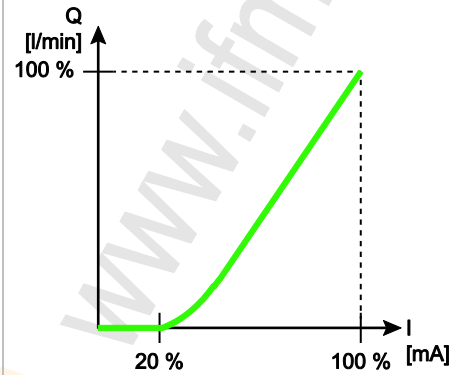
1562

通常，液压阀没有完全线性的特性：

液压阀的典型特性曲线：

液压油在线圈电流达到大约 20 % 时开始流动。 初始油流量不是线性的。

计算线圈电流的预设值时必须考虑这一点。 此库的 FB 在这里支持您。

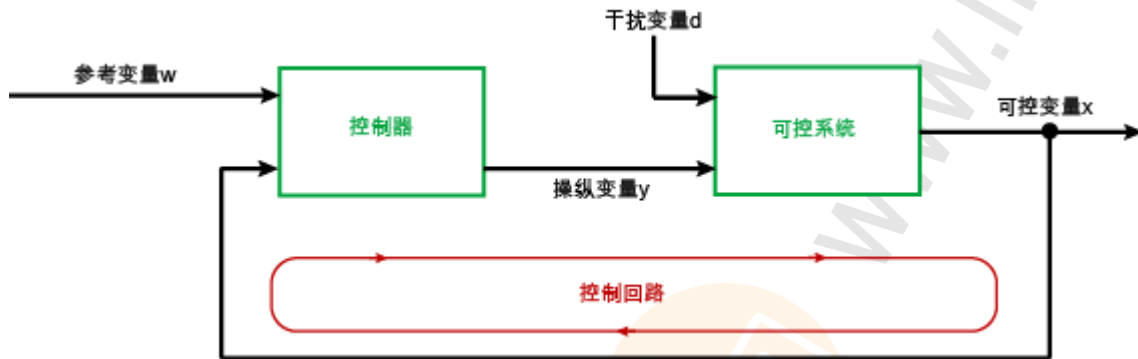




## 4.2 控制器 - 描述

13830  
1623

控制是持续检测要控制的单元（控制变量  $x$ ）并与基准变量  $w$  进行比较的过程。根据这一对比结果，控制变量会受到影响以适应基准变量。



图：控制原理

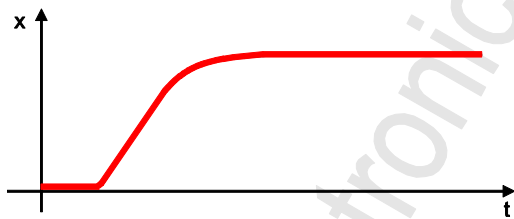
选择合适的控制装置及其最佳设置需要准确指出受控制系统的稳态行为和动态行为。在大多数情况下，这些特性值只能通过实验确定并很难受到影响。

受控系统可以分为三种类型：

### 4.2.1 自衡过程

1624

对于自衡过程，控制变量  $x$  在某个操纵变量（稳态）之后变为新的最终值。这些受控系统的决定性因素是放大（稳态传输因子  $K_S$ ）。放大比例越小，系统控制效果越好。这些受控系统称为 P 系统（ $P = \text{比例}$ ）。



图：P 控制器 = 自衡过程

## 4.2.2 无自平衡的受控系统

1625

具有指向无穷的受控系统称为无自平衡的受控系统。通常,这是由于整合性能造成的。结果是,控制变量在操纵变量已更改或受到干扰因素影响之后不断增大。由于此行为,它决不会达到最终值。这些受控系统称为 I 系统 (I = 完整)。

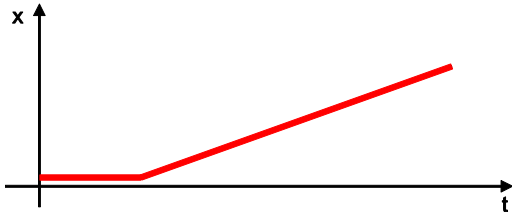


图: I 控制器 = 无自平衡的受控系统

## 4.2.3 带延迟的受控系统

1626

大多数受控系统相当于 P 系统的串联系统 (带补偿的系统) 和一个或多个 T1 系统 (带惯性的系统)。例如,一级受控系统由节流点和后续内存的串联连接组成。

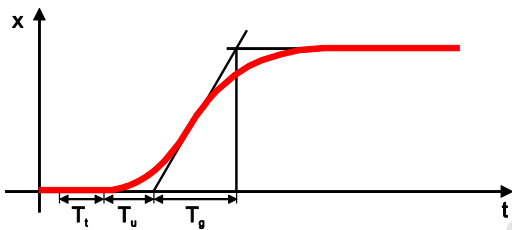


图: PT 系统 = 带延迟的受控系统

对于带死区时间的受控系统,控制变量不会在死区时间  $T_d$  过去之前,对控制变量的变化作出反应。死区时间  $T_d$  或  $T_d + T_u$  的总和与系统的可控性相关。系统的可控性越高,  $T_g/T_u$  比率越大。

集成在库中的控制器是前面的基本功能的汇总。它取决于各个受控系统所使用的功能以及组合方式。

## 5 使用用户闪存

### 内容

|                        |     |
|------------------------|-----|
| 闪存 – 什么是闪存？ .....      | 115 |
| 什么是 CSV 文件？ .....      | 116 |
| CSV 文件和 ifm 维护工具 ..... | 117 |

11607

某些 ifm 装置具有用户闪存。该闪存区域设计用于客户的应用数据。

应用示例：

- 报文文本（可以选择多种语言），在 PDM 中和显示屏上显示
- 负载限制值表，例如，升降机、起重机和转台云梯

编程人员可以创建列表或表。

① 用于创建列表或表的程序必须能够将源文件转换为 CSV 文件。

例如，合适的程序包括电子数据表，如 Microsoft Excel 或 OpenOffice Calc。

### ① 注意

CSV 文件不得包含任何安全相关的数据。  
未提供合适的备份措施。

### 5.1 闪存 – 什么是闪存？

11608

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。

## 5.2 什么是 CSV 文件？

11627

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

⚠ 请注意：

- 下载程序和维护工具需要源表的列之间有分隔符，例如，分号 (;)。
- CODESYS 需要一个空字节 (NUL) 作为字符串的终止符。
- 每个数据集 (要传输的每个表行) 都应具有相同数量的表列。

### 5.3 CSV 文件和 ifm 维护工具

#### 内容

|                         |     |
|-------------------------|-----|
| CSV 文件的要求.....          | 118 |
| 使用电子数据表程序创建 CSV 文件..... | 119 |
| 使用编辑器创建 CSV 文件.....     | 121 |
| 使用维护工具传输 CSV 文件.....    | 123 |
| 访问闪存数据：功能块.....         | 124 |

11619

以下装置可以与 ifm 维护工具进行通信：

通过 AddIn BasicSystem

- BasicController: CR040n, CR041n, CR043n
- BasicDisplay: CR045n
- SmartController: CR253n

通过 Addin R360System

- Controller: CR0n3n, CR7n3n
- 控制器：CR0020, CR0200, CR0505
- CabinetController: CR0303
- SmartController: CR2500

### 5.3.1 CSV 文件的要求

11630

- ▶ CSV 文件必须具有特定标题结构。  
所有标题数据都以“#”开头。
  - 第 1 行：CSV 文件类型  
例如：#File Type=0  
允许：0/1
  - 第 2 行（可选）：CSV 文件的项目名称  
例如：#Name=Demo Textmessages  
允许：0..20 个字符
  - 第 3 行（可选）：CSV 文件的版本  
例如：#Version=V01.00.00  
允许：0..12 个字符
- ifm 维护工具知道用户闪存的开始地址。  
CSV 文件中不必指出地址。
- ▶ 数据必须直接跟在标题数据行后，无任何间隙！  
**结构：**  
relative address;date 或 text;data type;{comment}  
**例如：**  
31;excess temperature;string(20);text 02  
数据类型后面必须带分号 (;)。
- ifm 维护工具本身会从数据类型生成正确的数据长度。  
因此，CSV 文件中不必详细地指出字符串数据。
- ▶ 分号 (;) 用作字段分隔符。

### 5.3.2 使用电子数据表程序创建 CSV 文件

11639

例如：

- BasicDisplay CR0451 的双语报文文本
- 20 个文本，每个高达 30 个字符，采用德语
- 20 个文本，每个高达 30 个字符，采用英语

|    | A                         | B | C          | D       |
|----|---------------------------|---|------------|---------|
| 1  | #FileType=0               |   |            |         |
| 2  | #Name=Messages D/E CR0451 |   |            |         |
| 3  | #Version=V01.00.00        |   |            |         |
| 4  | 0 Dies ist Text 1         |   | String(30) | Text 01 |
| 5  | 31 Dies ist Text 2        |   | String(30) | Text 02 |
| 6  | 62 Dies ist Text 3        |   | String(30) | Text 03 |
| 7  | 93 Dies ist Text 4        |   | String(30) | Text 04 |
| 8  | 124 Dies ist Text 5       |   | String(30) | Text 05 |
| 9  | 155 Dies ist Text 6       |   | String(30) | Text 06 |
| 10 | 186 Dies ist Text 7       |   | String(30) | Text 07 |
| 11 | 217 Dies ist Text 8       |   | String(30) | Text 08 |
| 12 | 248 Dies ist Text 9       |   | String(30) | Text 09 |
| 13 | 279 Dies ist Text 10      |   | String(30) | Text 10 |
| 14 | 310 Dies ist Text 11      |   | String(30) | Text 11 |
| 15 | 341 Dies ist Text 12      |   | String(30) | Text 12 |
| 16 | 372 Dies ist Text 13      |   | String(30) | Text 13 |
| 17 | 403 Dies ist Text 14      |   | String(30) | Text 14 |
| 18 | 434 Dies ist Text 15      |   | String(30) | Text 15 |
| 19 | 465 Reserve               |   | String(30) | Text 16 |
| 20 | 496 Reserve               |   | String(30) | Text 17 |
| 21 | 527 Reserve               |   | String(30) | Text 18 |
| 22 | 558 Reserve               |   | String(30) | Text 19 |
| 23 | 589 Reserve               |   | String(30) | Text 20 |
| 24 | 620 This is text 1        |   | String(30) | Text 21 |
| 25 | 651 This is text 2        |   | String(30) | Text 22 |
| 26 | 682 This is text 3        |   | String(30) | Text 23 |
| 27 | 713 This is text 4        |   | String(30) | Text 24 |
| 28 | 744 This is text 5        |   | String(30) | Text 25 |
| 29 | 775 This is text 6        |   | String(30) | Text 26 |
| 30 | 806 This is text 7        |   | String(30) | Text 27 |
| 31 | 837 This is text 8        |   | String(30) | Text 28 |
| 32 | 868 This is text 9        |   | String(30) | Text 29 |
| 33 | 899 This is text 10       |   | String(30) | Text 30 |
| 34 | 930 This is text 11       |   | String(30) | Text 31 |
| 35 | 961 This is text 12       |   | String(30) | Text 32 |
| 36 | 992 This is text 13       |   | String(30) | Text 33 |
| 37 | 1023 This is text 14      |   | String(30) | Text 34 |
| 38 | 1054 This is text 15      |   | String(30) | Text 35 |
| 39 | 1085 reserve              |   | String(30) | Text 36 |
| 40 | 1116 reserve              |   | String(30) | Text 37 |
| 41 | 1147 reserve              |   | String(30) | Text 38 |
| 42 | 1178 reserve              |   | String(30) | Text 39 |
| 43 | 1209 reserve              |   | String(30) | Text 40 |

说明：


| 字段编号    | 描述   |
|---------|--|
| A1...A3 | 标题； 条目以“#”开头   |
| A1      | #FileType=<br>0: 当编译 CSV 文件时，设置的参数直接以给定顺序存储在用户闪存中。<br>1: 当编译 CSV 文件时，设置的参数以可以直接使用 CoDeSys 结构读取数据的方式存储在用户闪存中。 |
| A2      | #Name=<br>用于定义表以及在应用程序中查找表的名称<br>长度 = 0...20 个字符   |

| 字段编号                  | 描述   |
|-----------------------|--|
| A3                    | #Version=<br>表的版本 (例如, 针对不同车辆)<br>长度 = 0...12 个字符            |
| A4...A43              | 报文文本开头的字节号   |
| A4                    | 第一种语言的第一个文本 (字节号 = 0)  |
| A24                   | 此处: 第二种语言的第一个文本 (字节号 = 620 = 补偿)                             |
| B4...B23<br>B24...B43 | 报文文本 (第一种语言)<br>相同报文文本 (第二种语言)<br><b>ifm</b> 维护工具仅将这些文本传输到装置 |
| C4...C43              | 装置类型, 此处: string(30)   |
| D4...D43              | 注释 (可选)<br>• 仅供在创建表时参考<br>• 注释不会传输到装置                        |

▶ 必须使用此结构, CoDeSys 才能理解生成的 CSV 文件。

▶ 保存电子数据表:

选择内存位置并输入文件名。

 选择合适的文件名, 以便稍后能够找到正确文件。

▶ 将电子数据表转换为 CSV 文件。

选择分号“;”作为列分隔符。

对于 **Excel**: [Save As...] > [Save as type:] = CSV 文件

对于 **OpenOffice**: [Save As...] > [Save as type] = 文本 CSV > [Keep Current Format]

在 [Export of text files] 窗口中:

- 将分号设置为字段分隔符
- 设置文本分隔符 = (空)

▶ 确认警告 (关于格式丢失)。

▶ 关闭电子数据表程序。

▶ 使用编辑器打开生成的 CSV 文件:

```
#FileType=0;::
#Name=Messages D/E CR0451;::
#version=v01.00.00;::
0;Dies 1st Text 1;String(30);Text 01
31;Dies 1st Text 2;String(30);Text 02
62;Dies 1st Text 3;String(30);Text 03
93;Dies 1st Text 4;String(30);Text 04
124;Dies 1st Text 5;String(30);Text 05
155;Dies 1st Text 6;String(30);Text 06
186;Dies 1st Text 7;String(30);Text 07
217;Dies 1st Text 8;String(30);Text 08
248;Dies 1st Text 9;String(30);Text 09
279;Dies 1st Text 10;String(30);Text 10
```

▶ 删除标题行 (以“#”开头) 后面的所有分号。

▶ 关闭编辑器并保存文件。





### 5.3.3 使用编辑器创建 CSV 文件

11640

- ▶ 手动输入请求的 CSV 文件内容。
- ▶ 将文件另存为 CSV 文件。  
文件类型 = ANSI

例如：

- BasicDisplay CR0451 的双语报文文本
- 20 个文本，每个高达 30 个字符，采用德语
- 20 个文本，每个高达 30 个字符，采用英语

```
#FileType=0
#Name=Messages D/E CR0451
#Version=V01.00.00
0;Dies ist Text 1;String(30);
11;Dies ist Text 2;String(30);
62;Dies ist Text 3;String(30);
93;Dies ist Text 4;String(30);
124;Dies ist Text 5;String(30);
155;Dies ist Text 6;String(30);
186;Dies ist Text 7;String(30);
217;Dies ist Text 8;String(30);
248;Dies ist Text 9;String(30);
279;Dies ist Text 10;String(30);
310;Dies ist Text 11;String(30);
341;Dies ist Text 12;String(30);
372;Dies ist Text 13;String(30);
403;Dies ist Text 14;String(30);
434;Dies ist Text 15;String(30);
465;Reserve;String(30);
496;Reserve;String(30);
527;Reserve;String(30);
558;Reserve;String(30);
589;Reserve;String(30);
620;This is text 1;String(30);
651;This is text 2;String(30);
682;This is text 3;String(30);
713;This is text 4;String(30);
744;This is text 5;String(30);
775;This is text 6;String(30);
806;This is text 7;String(30);
837;This is text 8;String(30);
868;This is text 9;String(30);
899;This is text 10;String(30);
930;This is text 11;String(30);
961;This is text 12;String(30);
992;This is text 13;String(30);
1023;This is text 14;String(30);
1054;This is text 15;String(30);
1085;reserve;String(30);
1116;reserve;String(30);
1147;reserve;String(30);
1178;reserve;String(30);
1209;reserve;String(30);
```

说明：

| 字段编号        | 描述   |
|-------------|--|
| 行 1...3     | 标题； 条目以“#”开头   |
| 行 1         | #FileType=<br>0: 当编译 CSV 文件时，设置的参数直接以给定顺序存储在用户闪存中。<br>1: 当编译 CSV 文件时，设置的参数以可以直接使用 CoDeSys 结构读取数据的方式存储在用户闪存中。 |
| 行 2<br>(可选) | #Name=<br>用于定义表以及在应用程序中查找表的名称<br>长度 = 0...20 个字符   |
| 行 3<br>(可选) | #Version=<br>表的版本（例如，针对不同车辆）<br>长度 = 0...12 个字符  |

| 字段编号           | 描述   |
|----------------|--|
| 0; 31; 62; ... | 报文文本开头的字节号 <ul style="list-style-type: none"> <li>• 以相对地址 0 开头</li> <li>• 以下地址，以 ( 行长度加 1 ) 个字节递增</li> </ul>                                       |
| ;text;         | 报文文本 ( 或保留 ) <ul style="list-style-type: none"> <li>• 每行正好 20 个字符 ( 填充任何字符，此处：点 )</li> <li>• 数据类型根据行长度自动生成</li> <li>• 仅这些数据稍后传输到装置中</li> </ul>     |
| String(30)     | 数据类型 <p> 数据类型后面必须带分号 (;)。</p> <p>ifm 维护工具会基于此信息自动生成正确长度的报文文本。</p> |

### 5.3.4 使用维护工具传输 CSV 文件

19189

- ▶ 将 ifm 装置的编程接口连接到 PC。
- ▶ 如果尚未完成，则使用 CODESYS 将应用程序（作为启动项目）传输到 ifm 装置。
- ▶ 启动 ifm 维护工具。

例如，对于 BasicSystem：

- ▶ 如果尚未完成，打开以下菜单：  
[ecomat mobile] > [CAN] > [Basic System]
- ▶ 在用户界面的左侧栏，在维护工具中选择以下菜单：  
[ecomat mobile] > [CAN] > [Basic System]
- ▶ 在用户界面的中间栏中选择以下菜单：  
[Basic System] > [System information] > [Identity]
- > 单击 [Read] 后，用户界面的右侧显示装置信息。

如果显示正确的装置数据：

- ▶ 在用户界面的中间栏中选择以下菜单：  
[Basic System] > [Software] > [Load]
- ▶ 单击用户界面右侧的 [to Basic System]。
- ▶ 在 字段 [Load software] 中单击 [Import \*.csv file...]
- > 窗口 [Load software] 显示。
- ▶ 选择内存位置和文件，并使用 [Open] 确认。
- > 具有以下信息的信息窗口显示：
  - 内存位置、路径
  - 内存：已使用 n 个字节，共 m 字节
  - 文件类型
  - 名称（来自 CSV 标题数据，可编辑）
  - 版本（来自 CSV 标题数据，可编辑）
- ▶ 使用 [Import] 将文件导入到要传输的文件列表。
- ▶ 标记要传输的文件（或：全部）。
- ▶ 使用 [Load] 将 CSV 文件传输到 ifm 装置。
- > 进度条指示过程的进度情况。
- > Then a finished message appears.
- ▶ 重新启动装置。

### 5.3.5 访问闪存数据：功能块

19190

应用程序只能使用以下功能块访问数据：

- 对于所有控制器（CR04nn，CR253n 除外）适用于：

|            |                      |
|------------|----------------------|
| FLASHREAD  | 将不同的数据类型直接从闪存传输至 RAM |
| FLASHWRITE | 将不同的数据类型直接写入闪存       |

- 对于所有 CR04nn、CR1nnn、CR253n，以下所述适用：

|            |  |
|------------|--|
| FLASH_INFO | 从用户闪存读取信息： <ul style="list-style-type: none"> <li>• 内存区域的名称（用户定义），</li> <li>• 软件版本，</li> <li>• 起始地址（用于使 IEC 结构简单读取）</li> </ul> |
| FLASH_READ | 直接将不同数据类型从闪存传输到 RAM  |

## 6 装置中的可视化

### 内容

|                      |     |
|----------------------|-----|
| 概要 .....             | 125 |
| 用户界面建议 .....         | 126 |
| 有关颜色和位图图形的基本信息 ..... | 142 |
| 有关位图图形的特殊信息 .....    | 146 |


3111

本章节介绍有关 CODESYS 可视化中的位图图形的重要信息。

### 6.1 概要

10465  
10464

除使用 CODESYS 可视化编辑器创建的图形元素外，您还可以集成使用其他程序创建的图形。例如，此类图形文件可以是象形图、徽标或更小的图像。但是，在您集成此类“外部图形”之前，必须考虑某些基本信息，这将在下面的章节中介绍。

 下面提供更多信息：

- 可视化的创建和参数设置：
  - CODESYS 编程手册 (→ **ecomatmobile** DVD“软件、工具和文档”)
  - **ifm** 手册“PDM – Handbuch zur Einführung”
- 请参阅 **限制和编程说明!**

#### 注意

▶ 在 PC 上的 CODESYS 下立即以某个项目名称保存新项目或通过模板创建的项目！  
如果将没有文件名的项目加载到装置中，则没有可视化显示。装置没有文件名，因此可视化无法启动。

## 6.2 用户界面建议

### 内容

|                    |     |
|--------------------|-----|
| 用户友好的产品设计的建议 ..... | 127 |
| 您是否了解未来的用户? .....  | 128 |
| 检查适用性.....         | 129 |
| 语言障碍 .....         | 130 |
| 文化底蕴通常不可转换.....    | 132 |
| 指令和标准.....         | 135 |

7435

用户友好性是接受并使用技术产品的决定性标准！

在本章节中，我们将提供有关如何尽可能用户友好地设计机器的用户界面（也称为“人机界面”（Human-Machine-Interface, HMI））的一些建议。

## 6.2.1 用户友好的产品设计的建议

7436

人机之间的所有重要界面都由用户平台和设计决定。设计人机界面的重要标准包括...

- 明确情况：
  - 清晰说明每个功能。
  - 按预期进行设计，了解到的内容保持不变
- 可读性：
  - 考虑环境（照明、阅读距离）。
- 直观的处理：
  - 操作元件 / 功能必须显而易见。
  - 用户界面必须能一目了然。
- 感官性
  - 操作元件必须用户友好。
  - 明确区分其他显示屏和操作元件。
- 反馈
  - 对用户活动作出快速反应。
  - 消息的原因必须十分明确。
- 产品的环境，由以下因素导致的干扰或刺激...
  - 噪声
  - 光线不足
  - 光反射
  - 振动
  - 极端温度

从制造商的角度来看，以下功能也十分重要：

- 显示屏作为品牌特定的特性。
- 显示屏必须符合标准。

## 6.2.2 您是否了解未来的用户？

7444

应了解产品未来的用户：

- 年龄
- 性别
- 感官能力：
  - 视力
  - 听力
  - 习惯用手（左或右撇子）
  - 触觉能力
- 培训和教育：
  - 一般教育水平
  - 特定培训研讨会和经验
- 积极性和认知能力：
  - 感知（感觉器官）：并非所有可用信息都会使用，但会在进入意识之前以许多方式进行大规模过滤、集成和更改。
  - 思考：对信息进行知识处理的工作记忆的容量非常小。
  - 学习：存储在长期记忆中的信息通常会提前（例如，由于期待）以及随后（例如，被后续信息）改变。
  - 回忆：长期记忆中“实际”存在的信息通常无法获取。
  - 积极性和专注力：疲劳、倦怠、注意力分散等可以影响认知能力。
- 对问题或应用领域熟悉：
  - 能够识别危险
  - 知道行动后会发生什么情况
- 应用的强度（产品的使用频率和强度）
- 文化，例如：
  - 语言
  - 颜色和符号的含义
  - 阅读方向



## 6.2.3 检查适用性

7422

在许多情况下，具有潜在用户的测试装置可以提供要使产品在市场上取得成功需要/必须进行的改进以及改进方式的重要结果。

对于此“可用性测试”，必须执行以下步骤：

- 确定用户群（目标群体）：
  - 谁要使用产品？
- 准备访谈指南：
  - 使用哪种方法访谈哪些用户（操作员、安装工、维护人员）？
  - 想要通过访谈实现什么目标？（改进潜力）
- 执行访谈并评估。
- 创建情境场景：
  - 创建可评估的测试环境。
  - 确定关键用户场景。
- 执行可用性测试：
  - 测试人员如何在测试装置中处理产品？
  - 在哪些地方需要对产品执行哪些纠正措施？
- 如果需要，在优化了产品之后重复测试。

## 6.2.4 语言障碍

7454

要生产满足全球最终用户的装置，必须要考虑语言。如果操作员不能理解屏幕上的说明，他就无法有效执行此任务。考虑到世界上有许多种不同语言，制造商仍在尝试解决这一问题。下面列出了一些语言：

### 中文字符

中文字符（也称为汉字）是一种语标符号，即可以代表一个词。《康熙字典》中收录的汉字数超过 47000 个，但是在中国，认识三到四千个字就够了。在现代，汉字已经被极大地简化并在中国大陆使用，而香港和台湾仍在使用繁体中文。汉字已经拉丁化，称为“拼音”，也在中国广泛使用。

### 日文字符

现代日语书写系统使用三种主要手写体：

- 日本汉字是来自汉字的象形文字
- 平假名用于本土日语词，以及
- 片假名用于外来词
- 拉丁化的日本字（称为罗马字）也在日文中使用。

### 韩文

现代韩文书写系统称为“韩语”，在北朝鲜和南朝鲜正式使用。此外，还使用韩文汉字，称为从中文借用的字符。

## 阿拉伯字母

此书写体用于书写亚洲（例如，中东、巴基斯坦）和非洲（例如，阿拉伯和乌尔都）的许多语言。它采用草书风格从右向左书写，并包括 28 个字母。

## Unicode

Unicode 是一种一致地表示和使用世界书写系统中的字符的标准。根据计算机调整语言并不容易，部分原因是某些语言的字符数量较大。可以仅使用一个字节编码一个英文字符，因为编写的英文只需要少量字符。这不适用于像日文、中文或韩文之类的语言，它们拥有超过 256 个字符，因此需要双字节或多字节编码。有许多编码方式在使用，Unicode 似乎是最常用的方式。它显然会编码成世界上的所有语言。

例如，Han unification（中日韩越统一表意文字，缩写为“Unihan”）是 Unicode 和通用字符集（根据 ISO 10646）用于在单个统一字符集中映射中文、日文和韩文的多个字符集的方法。

阿拉伯字符可以使用 Unicode 版本 5.0 或更高版本（多个字符集和 ISO 8859-6）进行编码。

ISO 10646 指定通用多八位编码字符集。它用于表示、交换、处理、存储和输入世界上的语言的书写形式，以及其他符号。

Unicode 标准版本 4...6 全部符合 ISO 10646 标准。

## 象形图

这是一种图形符号（也称为“象形文字”），通过图示代表某个概念、对象、事件或活动。象形图已经使用了几千年。它们在当今世界存在语言障碍和文盲的情况下仍十分重要，并且用作图形标志、代表性标志、说明或统计图。由于其图形性质，它们被用于生活中的不同方面。例如，为指示洗手间和机场，ISO 7001 标准“图形符号 - 公共信息符号”中定义了一组标准象形图。

象形图已发展成为针对具有认知问题的人的功能性视觉语言。每个图像代表一个词或概念。它由两个元素（绘图和文字）组成。这些符号通常为黑底方块白色图形。

## 6.2.5 文化底蕴通常不可转换

7461

源文本中应避免国家/地区、文化或语言特定的底蕴，因为通常不必使用，而且根据目标文化进行的调整会花费大量时间。在大多数情况下，作者不知道他的文本或图形在文化或语言方面具有特征，或者由于其他设计相关的决策导致本地化问题。例如，在以下情况下，可能会发生问题：

- 颜色
- 符号
- 图示
- 阅读方向

### 颜色

7464

选择“正确”的颜色是文本和产品设计的重要要素。许多颜色具有文化特异性，如果使用不当可导致误解，甚至是由于处理故障导致产品图像丢失。

示例：

| 颜色 | 在欧洲和美国的含义                        | 在其他文化中的含义   |
|----|----------------------------------|---|
| 红色 | 戏剧性、骚乱、血（争斗、复仇和死亡）、爱、危险、高贵       | 中国：幸运、欢快<br>俄罗斯：美丽<br>埃及：死亡<br>印度：生命、有创造力<br>日本：生气、危险       |
| 黄色 | 注意、警告、阳光、永恒、嫉妒、仇恨                | 中国：声势显赫、健康、力量<br>埃及：快乐、富有<br>印度：成功<br>日本：高贵                 |
| 绿色 | 自然、环保、希望、不朽、财富                   | 中国：永恒、家庭、和谐、健康、和平、后代<br>埃及：富饶、强大<br>印度：富有、富饶<br>日本：未来、年轻、活力 |
| 蓝色 | 水、天空、忠诚、自由、可靠、欢乐、友好、有男子气概        | 亚洲：富有、强大<br>埃及：贞洁、信仰、真理                                     |
| 白色 | 光、纯洁、聪明、生命、完美、理想、好、实事求是、纯净、清白、诚实 | 亚洲：死亡、悲伤、纯洁<br>埃及：欢乐  |
| 黑色 | 死亡、悲伤、黑暗、邪恶 以及：友爱、权力和团结          | （在佛教不代表悲伤）<br>埃及：复活   |
| 灰色 | 智慧和成熟                            | 亚洲：令人鼓舞   |

© ifm electronic gmbh



www.ifm.com

## 符号

7465

由于符号的产生通常是类似于具有文化特异性的概念，或使用源文化的熟悉领域相关的典故，它们会为本地化带来问题。

例如：



用于代表开始的房屋符号无法让人清晰地理解，因为无法毫无问题地转换英语“home”一词。

## 图示

7466

图像不总是文字的明智替代品。

可能无法表示更复杂的过程。例如，如何用图示说明“按下按钮，直到您感到轻微的阻力”请求？

即使图示可以很好地表示某种事实，但在国际级别使用它要好好考虑好。仅当图示与文化无关，即，可以在所有预期目标国家/地区使用而无需调整时，用图像代替文字才是明智的，而且会降低成本。许多对我们来说不言自明的事在其他文化中并非如此。

用图示代表人可导致以下问题：这个人必须或者可能具有哪种性别？是什么肤色？多大年龄？最后，所有目标国家/地区中的适用对象感觉受到公平对待。在西欧不显眼的服饰可能会在阿拉伯或非洲国家/地区导致恼怒。不得用图示代表手势和单个身体部位（尤其是手部和眼睛），因为它们经常触发攻击性或侮辱联想。

## 阅读方向

7468

在大多数文化中，是从左向右，从上至下进行阅读的。

但是，某些亚洲文化是从下至上，从后向前阅读。

许多阿拉伯文化从右向左阅读。

图形说明必须考虑这些特殊性！

## 6.2.6 指令和标准

| 内容  |     |
|---|-----|
| ISO 7001 _ 图形符号 – 公共信息符号 .....            | 135 |
| ISO 9126 _ 软件工程 – 产品质量.....               | 136 |
| ISO 9241 _ 人-系统交互人类工效学 .....              | 137 |
| ISO 10646 _ 信息技术 – 通用多八位编码字符集 (UCS) ..... | 139 |
| ISO 13406 _ 平板视觉显示工作的人类工效学要求.....         | 140 |
| ISO 13407 _ 以人为中心的交互系统设计过程 .....          | 140 |
| ISO 20282 _ 日常用品的易操作性 .....               | 141 |

7445

下面的列表只是选择一部分，并非全部。

### ISO 7001 \_ 图形符号 – 公共信息符号

7456

图形符号（也称为“象形文字”）通过图示代表某个概念、对象、事件或活动。象形图已经使用了几千年。它们在当今世界存在语言障碍和文盲的情况下仍十分重要，并且用作图形标志、代表性标志、说明或统计图。由于其图形性质，它们被用于生活中的不同方面。

例如：



## ISO 9126 \_ 软件工程 – 产品质量

7446

此标准描述以下标准：

**功能：** 软件拥有的所需功能达到什么程度？

- 适合性：功能针对指定任务的适合性，例如，通过子功能组成面向任务的功能。
- 正确性：提供正确或约定的结果或效果，例如，计算值的必要精度。
- 互操作性：能够与指定系统交互。
- 安全性：能够防止未经授权访问（无意或有意）程序和数据。
- 合规性：软件功能使软件符合应用特定标准或协议或法律规定和类似法规。

**可靠性：** 软件是否能够在规定期限内规定条件下保持指定性能水平？

- 成熟度：错误状态下故障频率低。
- 容错度：能够在软件错误或不符其指定接口时保持指定性能水平。
- 稳健性：能够在输入端异常时确保稳定的系统。软件能够经得住“新手”操作。
- 恢复能力：能够在发生故障时恢复性能水平以及检索直接相关的数据。必须考虑时间和所需投入水平。
- 合规性：软件符合可靠性标准或协议的程度。

**可用性：** 用户使用软件所需的投入水平以及如何评估？

- 易懂性：用户理解概念及其应用所需的投入水平。
- 易学性：用户了解应用（例如，处理、输入端、输出端）所需的投入水平。
- 可用性：用户处理应用所需的投入水平。
- 吸引力：应用对用户的吸引力。
- 合规性：软件符合可用性标准或协议的程度。

**效率：** 软件的性能水平与所使用装置之间的关系怎么样？

- 时间行为：执行功能时的响应和处理时间以及数据处理速度。
- 消费行为：执行功能所需操作元件的数量和触动时间。资源消耗，如 CPU 时间、硬盘访问等。
- 合规性：软件符合效率标准或协议的程度。

**可变性：** 在软件中进行定义的更改需要哪种投入水平？更改可能包括根据环境、要求或功能规格的改变进行的纠正、改进或调整。

- 可析性：诊断瑕疵或故障原因或者确定需要更改的部分所需的投入水平。
- 可修改性：根据改变的环境执行改进、消除故障或进行调整所需的投入水平。
- 稳定性：出现意外更改影响的可能性。
- 可测试性：测试更改的软件所需的投入水平。

**可转换性：** 是否能够将软件轻松地转换到其他环境中？可能是组织环境、硬件环境或软件环境。

- 适应性：软件适应不同环境的能力。
- 可安装性：在指定环境中安装软件所需的投入水平。
- 共存：软件能够与具有类似或相同功能的其他软件并行运行的能力。
- 可换性：能够在另一个指定软件的环境中取代该软件使用此软件的可能性以及为此所需的投入水平。
- 合规性：软件符合可转换性标准或协议的程度。



## ISO 9241 \_ 人-系统交互人类工效学

7447

ISO 9241 标准是一种介绍人和计算机之间的交互指南的国际标准。该标准系列介绍工作环境、硬件和软件的要求。该指南的目标是避免计算机工作场所的健康损害，以及使用户易于执行其任务。

以下部分（不完整列表）是该标准的一部分：

- 第 1 部分：一般说明
- 第 2 部分：任务要求指南
- 第 3 部分：视觉显示要求
- 第 4 部分：键盘要求
- 第 5 部分：工作台布置及姿势要求
- 第 6 部分：工作环境指南
- 第 7 部分：反射显示的要求
- 第 8 部分：显示色要求
- 第 9 部分：非键盘输入装置的要求  
(第 10 部分：对话原则(过时,被 2006 中的第 110 部分取代))
- 第 11 部分：使用指南
- 第 12 部分：信息表达
- 第 13 部分：用户指南
- 第 14 部分：菜单对话框
- 第 15 部分：命令对话框
- 第 16 部分：直接操作对话框
- 第 17 部分：表格填写对话框
- 第 110 部分：对话框原则(取代第 10 部分)
- 第 151 部分：全球网用户接口指南
- 第 171 部分：软件可访问性指南(于 2008 年 10 月发布)
- 第 300 部分：电子视觉显示要求概述
- 第 302 部分：电子视觉显示术语(目前处于草案阶段)
- 第 303 部分：电子视觉显示要求(目前处于草案阶段)
- 第 304 部分：用户性能试验方法
- 第 305 部分：电子视觉显示光学实验室试验方法(目前处于草案阶段)
- 第 306 部分：电子视觉显示现场评估方法(目前处于草案阶段)
- 第 307 部分：电子视觉显示分析和符合性试验方法(目前处于草案阶段)
- 第 400 部分：物理输入装置原则和要求
- 第 410 部分：物理输入装置设计标准(目前处于草案阶段)

第 5 部分和第 6 部分涉及工作环境。第 3 部分、第 4 部分、第 7 部分、第 8 部分和第 9 部分涉及硬件要求，第 11...17 部分和第 110 部分涉及软件人类工效学方面。基本上，**ISO 9241-110“对话框原则”**(→ 页 138)和 **ISO 9241-11“使用指南”**(→ 页 138)部分包含一些交互系统的人类工效学设计标准。

## ISO 9241-11 \_ 使用指南

7448

软件的可用性取决于其使用情境。ISO 9241 的第 11 部分定义了软件使用的三种主要标准：

- 解决任务的有效性
- 使用系统的效率
- 软件用户的满意度

## ISO 9241-110 \_ 对话框原则

7450

交互系统的用户界面（如，网站或软件）应易于使用。ISO 9241 的第 110 部分介绍了用户和系统之间的界面（对话框设计）的以下设计和评估原则：

- 任务适用性  
合适的功能、最大限度减少不必要的交互
- 自描述性  
借助支持/反馈提供易懂性
- 适合学习  
用户指南、适合的隐喻、目标：最大限度缩短学习时间
- 可控性  
由用户进行对话框控制
- 符合用户预期  
一致性、适合用户模型
- 个体适合性  
可根据用户及其工作环境进行调整
- 容错度  
优先提供智能对话框原则，以使用户避免错误。其他方面：检测到的用户错误不会阻止用户目标。  
未检测到的错误：由用户进行轻微调整。

### ISO 10646 \_ 信息技术 – 通用多八位编码字符集 (UCS)

7455

通用字符集 (UCS) 是国际标准 ISO 10646 中定义的标准字符集。实际上，与 Unicode 相同。

每个字符使用 2 个字节的内存空间。Unicode 是一个 16 位代码，代表  $2^{16} = 65536$  个字符。第一个目标是清楚、标准化地编码所有国家语言的字符。

并非所有这 65536 个字符地址都会使用。用户定义的区域启用大约 2000 个带用户特定字符的地址。

通过组合两个 16 位代码可以编码其他 1408576 个字符。希望能够涵盖存在或曾存在的所有字符。

此外，还会映射技术符号、音乐符号、音标等。然而，离使用所有字符地址还很远。

例如：

|   | 000 | 001 | 002 | 003 | 004 | 005 | 006 | 007 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | NUL | DLE | SP  | 0   | @   | P   | ~   | p   |
| 1 | SOH | DC1 | !   | 1   | A   | Q   | a   | q   |
| 2 | STX | DC2 | "   | 2   | B   | R   | b   | r   |
| 3 | ETX | DC3 | #   | 3   | C   | S   | c   | s   |
| 4 | EOT | DC4 | \$  | 4   | D   | T   | d   | t   |
| 5 | ENO | NAK | %   | 5   | E   | U   | e   | u   |
| 6 | ACK | SYN | &   | 6   | F   | V   | f   | v   |
| 7 | BEL | ETB | '   | 7   | G   | W   | g   | w   |
| 8 | BS  | CAN | (   | 8   | H   | X   | h   | x   |
| 9 | HT  | EM  | )   | 9   | I   | Y   | i   | y   |
| A | LF  | SUB | *   | :   | J   | Z   | j   | z   |
| B | VT  | ESC | +   | ;   | K   | [   | k   | {   |
| C | FF  | FS  | ,   | <   | L   | \   | l   |     |
| D | CR  | GS  | -   | =   | M   | ]   | m   | }   |
| E | SO  | RS  | .   | >   | N   | ^   | n   | ~   |
| F | SI  | US  | /   | ?   | O   | _   | o   | DEL |

Unicode：控制字符和基本字符

|   | 219 | 21A | 21B | 21C | 21D | 21E | 21F |
|---|-----|-----|-----|-----|-----|-----|-----|
| 0 | ←   | →   | ↖   | →   | ←   | ←   | ↗   |
| 1 | ↑   | ↓   | ↖   | →   | ↑   | ↑   | ↖   |
| 2 | →   | ↖   | ↓   | ↓   | ⇒   | ⇒   | ↘   |
| 3 | ↓   | ↗   | ↘   | ↓   | ↓   | ↓   | ↕   |
| 4 | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   |
| 5 | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   |
| 6 | ↖   | ↖   | ↖   | ↖   | ↖   | ↖   | ↖   |
| 7 | ↗   | ↗   | ↗   | ↗   | ↗   | ↗   | ↗   |
| 8 | ↘   | ↘   | ↘   | ↘   | ↘   | ↘   | ↘   |
| 9 | ↙   | ↙   | ↙   | ↙   | ↙   | ↙   | ↙   |
| A | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   |
| B | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   |
| C | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   |
| D | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   |
| E | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   |
| F | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   | ↔   |

Unicode：箭头

## ISO 13406 \_ 平板视觉显示工作的人类工效学要求

7453

### 第 2 部分：平板显示人类工效学要求

根据国际标准 ISO 13406-2，根据以下标准对 LCD 屏幕进行分类：

- 从观看者的方向测得的亮度、对比度和颜色
- 存在入射光时的反光和对比度
- 图像准备时间
- 故障（像素故障）

## ISO 13407 \_ 以人为中心的交互系统设计过程

7452

ISO 13407 是一种介绍典型以人为中心的软件开发过程的标准。如果特殊开发过程符合标准建议，则可以将其视为符合标准。

该标准将以人为中心的设计表示为涉及人为因素和人类工效学信息和技术知识的跨学科活动。ISO 过程包括四个重要的子活动：

- 了解使用情境：  
此活动的结果是相关用户、其任务及其环境的文档描述。
- 指定要求：  
在此阶段期间，从现有文档折衷地扣除目标。以下任务中定义了系统任务的划分...
  - 由人执行的任务
  - 由技术执行的任务
- 生成解决方案：  
这可以在原型开发或其他迭代过程之后完成。这些原型可以是纸质草稿（模拟），也可以是可执行程序版本。如果用户界面存在公司内部设计规则，应使用这些规则。
- 评估解决方案：  
检查解决方案是否符合定义的要求。为此，可以进行专家评估、可用性测试、访谈或这几项的组合。评估确定偏差的相关性，并将其作为开发过程下一次迭代的起点。

此方法与软件开发的现有过程模型互补。根据此标准，以人为中心的设计过程应在项目的最早阶段中开始，并且应重复直到系统符合要求。以人为中心的设计的重要性和所需投入水平取决于要开发产品的大小和类型。对于小型项目，由个人控制。

## ISO 20282 \_ 日常用品的易操作性

7443

此草案包括

- 第 1 部分：针对使用情境和用户特征的设计要求  
其中介绍了以下标准：
  - 范围
  - 用户界面
  - 用户
  - 心理和社会特征
  - 物理和社会环境因素
  - 身体和感官特征
- 第 2 部分：上坡用产品的试验方法  
本部分是试验方法的技术规格。



### 6.3 有关颜色和位图图形的基本信息

|                       |     |
|-----------------------|-----|
| 内容                    |     |
| 图像大小矢量图形 / 像素图形 ..... | 143 |
| 位图图形的颜色 .....         | 145 |
| 显示哪些颜色？ .....         | 145 |

3112

对于图形和图像文件，区分了两种基本类型：

|          | 矢量图形  | 像素图形   |
|----------|---|--|
| 例如：      | CAD 程序图纸<br>字符集类型 TrueType、PostScript 或 OpenType  | 数码相片<br>来自扫描仪或捕获程序的文件  |
| 原理：      | 矢量图形基于准确定义用于生成图像的对象<br>的图像描述。<br>例如，通过中心的位置（坐标）、半径、线粗细<br>和颜色定义一个圆。                                   | 光栅图形（也称为像素图形或位图）是一种描述<br>由光栅型像素分配（每种像素分配一种颜色）组<br>成的图像的方式。因此，光栅图形的主要特征<br>是像素宽度和高度（图像分辨率），以及颜色深<br>度。                    |
| 所需内存空间：  | 所需内存空间相对较小  | 根据分辨率，所需内存空间较高或极高：文件<br>随着每额外存储一个像素而增大。  |
| 缩放时损失：   | 可以对任何图像大小进行无损重采样（缩放）  | 在大多数情况下，对其他图像大小进行重采样<br>（缩放）会导致质量损失。   |
| 硬件性能：    | 由于显示器原则上基于光栅矩阵，因此必须对<br>所有图形进行单个像素重采样（= 光栅化）以在显示<br>器上显示。<br>根据图形的复杂性，需要功能十分强大的计算机<br>以实现快速处理和显示。     | 要求相对较低   |
| 典型文件扩展名： | *.cdr (Corel Draw)<br>*.dwg (AutoCAD)<br>*.ai (Adobe Illustrator)<br>*.svg (Scalable Vector Graphics) | *.bmp (Bitmap)<br>*.gif (CompuserV GIF)<br>*.jpg (Joint Photographic Experts Group)<br>*.png (Portable Network Graphics) |

### 6.3.1 图像大小 矢量图形 / 像素图形

7380

| 矢量图形   | 像素图形  |
|--|---|
| 图形元素被描述为矢量：有关起点和终点、线粗细和颜色、可能的填充图案和颜色渐变的信息。   | 现代数码相机的像素图形具有 500 万及以上的像素（分辨率 = 500 万像素）。特殊数据压缩会尝试减小所需高内存空间。遗憾的是，压缩会导致质量不佳。   |
| 减小或放大十分容易，并且不会导致质量损失（→ 示例如下）。  | 放大会导致图形分块或图像模糊（→ 示例如下）。缩小这样的百万像素图像会导致图像信息损失大。   |
| 例如：<br><br>原始直径 10 mm / 放大 5 倍<br>EPS 文件 35 kB | 例如：<br><br>原始 30 x 30 px / 放大 5 倍<br>BMP 文件 3 kB / 62 kB |

#### 例如：缩小 Cr108n 的像素图像

19193  
7402

**任务：**例如，分辨率为 500 万像素的现有数码照片的图像大小为 2560 x 1920 像素（= 4,915,200 像素）。

此照片要以仅 800 x 480 像素（= 显示器尺寸 PDM360NG）的图像大小显示。

**问题 1：**源照片的宽高比为 4:3 (1.33:1)，但目标照片为 15:9 (1.66:1)。

**解决方案（各向异性）：**以不同的比例缩放图像的高度和宽度，以在显示屏上显示无失真图像。

对于均匀（各向同性）缩放，与原始图像相比，图像会失真。

**问题 2：**缩放后，仅有 384,000 像素（= 7.8% 的原始图像），其他 4,531,200 像素不再可用。

换言之：水平方向，仅每隔 2 个像素使用一个，垂直方向，仅每隔 3 个像素使用一个。

因此，这样转换的照片不再具有原始照片的质量。重要信息丢失，且图像失真。

► **❗ 解决方案：**从一开始就以所需尺寸和分辨率创建图像。

该问题相应地适用于具有不同显示器尺寸的其他装置。

## 调整位图图形

9996

您可以借助常用图形软件调整现有位图图形。

请问问您的 **ecomatmobile** 专家！



© ifm electronic gmbh



## 6.3.2 位图图形的颜色

3121

第二个重要因素是为每一个像素存储的颜色信息 ( RGB 值 )。

RGB 代表红色、绿色和蓝色。对于这三种主要颜色中的每种颜色，有 255 个强度水平。通过以不同的强度混合这三种主要颜色，大约可以创建 1660 万种颜色，方法是：**加色法混色** (→ 页 146)。要表现这一数量，需要合适的显示器和功能强大的处理器。

## 6.3.3 显示哪些颜色？

19194

### Cr108n 中的颜色

19195  
7381

显示屏可以每种主要颜色表现 6 位颜色深度，即，64 种颜色等级。因此，从共 256 个连续可寻址颜色等级中，仅可每隔三种颜色使用一种：

| 颜色 | 允许的颜色值  |
|----|---|
| 红色 | R = 0, 4, 8, 12, 16, ..., 236, 240, 244, 248, 252 |
| 绿色 | G = 0, 4, 8, 12, 16, ..., 236, 240, 244, 248, 252 |
| 蓝色 | B = 0, 4, 8, 12, 16, ..., 236, 240, 244, 248, 252 |

不符合此样式的值不会显示。

### Cr045n 中的颜色

19196  
8367

装置可以表现 8 位颜色深度，即，一共 256 种颜色等级。从每个颜色通道的共 256 个连续可寻址颜色等级 (= 16,777,216 种颜色) 中，仅可每隔 65,535 种颜色使用一种：

| 颜色 | 允许的颜色值  |
|----|---|
| 红色 | R (3 位 = 8 个等级) = 0, 32, 64, 96, 128, 160, 192, 224 |
| 绿色 | G (3 位 = 8 个等级) = 0, 32, 64, 96, 128, 160, 192, 224 |
| 蓝色 | B (2 位 = 4 个等级) = 0, 64, 128, 192                   |

调色板在工厂指定，并永久存储在装置中。

不符合此样式的值不会显示。

## 6.4 有关位图图形的特殊信息

### 内容

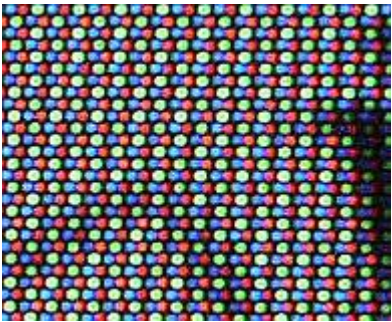
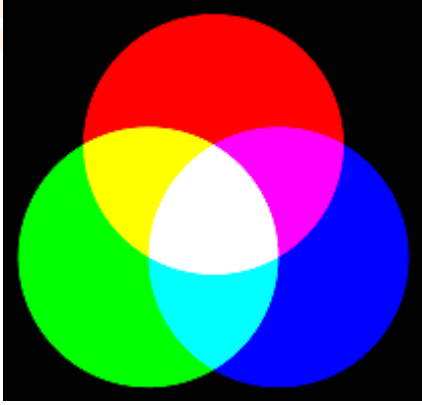
|                               |     |
|-------------------------------|-----|
| 加色法混色.....                    | 146 |
| 哪些图形适合哪些 PDM 以及必须执行哪些步骤？..... | 147 |

3113









感兴趣的读者可以在这里找到有关位图图形的更多详细信息。

### 6.4.1 加色法混色

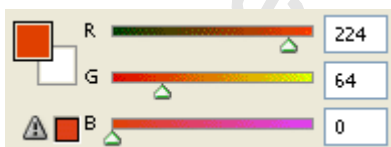
3123

| RGB   | 混色   |
|---|--|
| 显示器和许多打印机通过 3 种主要颜色（红色、绿色和蓝色）进行混色。  | 通过按所需混合比率添加颜色来进行混色。因此，此方法称为加色法混色。  |
|  |  |
| 照片：显示器的 RGB 光栅，大幅放大   | 图：加色法混色  |

表：颜色混合示例

|          |   |            |  |             |   |
|----------|---|------------|--|-------------|---|
| 100 % 红色 |  | + 100 % 绿色 |  | = 100 % 黄色  |  |
| 100 % 绿色 |  | + 100 % 蓝色 |  | = 100 % 蓝绿色 |  |
| 100 % 蓝色 |  | + 100 % 红色 |  | = 100 % 洋红色 |  |

更少份额的各个主要颜色导致的色彩饱和度细微差别：



屏幕截图：Photoshop 上的 RGB 颜色混合；100 %  $\Rightarrow$  255<sub>dez</sub> = FF<sub>hex</sub>

## 6.4.2 哪些图形适合哪些 PDM 以及必须执行哪些步骤？

7387

并非所有位图都适合在 PDM 上显示。

- 原则上，应转换照片，这样在以指定分辨率和颜色深度显示时会对它们进行优化。
- 对比度较低的图像不合适，因为 PDM 上无法显示色差。
- 如果需要，应优化徽标和符号，以便在 PDM 上显示或重新绘制。



# 7 使用的文件和库概述

## 内容

一般概述 ..... 148  
 单个文件和库有什么用途？ ..... 150

2711

(截至 2014-06-25)

根据单元和所需功能，使用不同的库和文件。有些库和文件会自动加载，其他的必须由编程人员插入或加载。

## 7.1 一般概述

2712

| 文件名   | 描述和内存位置 <sup>1)</sup>   |
|---|---|
| ifm_CRnnnn_Vxx.CFG                              | PLC 配置<br>每个装置仅 1 个装置特定文件<br>包括：输入端和输出端、标记字节以及内存分配的 IEC 和符号地址<br>...\CoDeSys V*\Targets\ifm\ifm_CRnnnncfg\Vxxyzz  |
| CAA-*.CHM                                       | 联机帮助<br>每个装置仅 1 个装置特定文件<br>包括：本装置的联机帮助<br>...\CoDeSys V*\Targets\ifm\Help\... (语言)                                |
| ifm_CRnnnn_Vxxyzz.H86<br>ifm_CRnnnn_Vxxyzz.RESX | 运行时系统<br>(首次使用时必须加载到控制器/显示器中)<br>每个装置仅 1 个装置特定文件<br>...\CoDeSys V*\Targets\ifm\Library\ifm_CRnnnn                 |
| ifm_Browser_CRnnnn.INI                          | CODESYS 浏览器命令<br>(CODESYS 需要此文件才能启动项目)<br>每个装置仅 1 个装置特定文件<br>包括：在 CODESYS 中用于浏览器的命令<br>...\CoDeSys V*\Targets\ifm |
| ifm_Errors_CRnnnn.INI                           | CODESYS 错误文件<br>(CODESYS 需要此文件才能启动项目)<br>每个装置仅 1 个装置特定文件<br>包括：CODESYS 发出的装置特定错误报文<br>...\CoDeSys V*\Targets\ifm  |
| ifm_CRnnnn_Vxx.TRG                              | 目标文件<br>每个装置仅 1 个装置特定文件<br>包括：CODESYS 的硬件描述，例如：内存、文件位置<br>...\CoDeSys V*\Targets\ifm                              |
| ifm_*_Vxxyzz.LIB                                | 一般库<br>每个装置可能有多个文件<br>...\CoDeSys V*\Targets\ifm\Library  |

| 文件名                     | 描述和内存位置 <sup>1)</sup>   |
|-------------------------|---|
| ifm_CRnnnn_Vxxyzz.LIB   | 装置特定库<br>每个装置仅 1 个装置特定文件<br>包括：此装置的功能元件<br>...\\CoDeSys V*\Targets\ifm\Library\ifm_CRnnnn |
| ifm_CRnnnn_*_Vxxyzz.LIB | 装置特定库<br>每个装置可能有多个文件<br>→ 以下表<br>...\\CoDeSys V*\Targets\ifm\Library\ifm_CRnnnn           |

说明：

- \* 任何标志
- CRnnnn 控制器/显示器的货号
- V\* CODESYS 版本
- Vxx ifm 软件的版本号
- yy ifm 软件的发行版号
- zz ifm 软件的补丁号

1) 文件的内存位置：

系统盘符 (C:/ D:)\ program folder\ ifm electronic

## 7.2 单个文件和库有什么用途？

### 内容

|                          |     |
|--------------------------|-----|
| 用于运行时系统的文件.....          | 150 |
| 目标文件 .....               | 150 |
| PLC 配置文件.....            | 150 |
| ifm 装置库 .....            | 151 |
| ifm CANopen 库主站/从站 ..... | 151 |
| CODESYS CANopen 库 .....  | 152 |
| 特定 ifm 库.....            | 153 |

2713

以下概述显示哪些文件/库可以以及可能会与哪些单元一起使用？本列表中未指出的文件/库可能只能在某些条件下使用，或者功能可能尚未经过测试。

### 7.2.1 用于运行时系统的文件

2714

| 文件名   | 功能            | 可用于：  |
|---|---------------|---|
| ifm_CRnnnn_Vxxyzz.H86<br>ifm_CRnnnn_Vxxyzz.RESX | 运行时系统         | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• BasicDisplay: CR045n</li> <li>• PDM: CR10nn</li> </ul> |
| ifm_Browser_CRnnnn.INI                          | CODESYS 浏览器命令 | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• PDM: CR10nn</li> </ul>                                 |
| ifm_Errors_CRnnnn.INI                           | CODESYS 错误文件  | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• PDM: CR10nn</li> </ul>                                 |

### 7.2.2 目标文件

2715

| 文件名                | 功能             | 可用于：  |
|--------------------|----------------|---|
| ifm_CRnnnn_Vxx.TRG | CODESYS 中的目标文件 | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• BasicDisplay: CR045n</li> <li>• PDM: CR10nn</li> </ul> |

### 7.2.3 PLC 配置文件

2716

| 文件名                   | 功能                | 可用于：  |
|-----------------------|-------------------|---|
| ifm_CRnnnn_Vxxyzz.CFG | CODESYS 中的 PLC 配置 | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• BasicDisplay: CR045n</li> <li>• PDM: CR10nn</li> </ul> |

## 7.2.4 ifm 装置库

2717

| 文件名                         | 功能           | 可用于：   |
|-----------------------------|--------------|--|
| ifm_CRnnnn_Vxxyzz.LIB       | 装置特定库        | <ul style="list-style-type: none"> <li>• 所有 ecomatmobile 控制器</li> <li>• BasicDisplay: CR045n</li> <li>• PDM: CR10nn</li> </ul> |
| ifm_CR0200_MSTR_Vxxyzz.LIB  | 无扩展功能的库      | <ul style="list-style-type: none"> <li>• ExtendedController: CR0200</li> </ul>   |
| ifm_CR0200_SMALL_Vxxyzz.LIB | 无扩展功能、功能减少的库 | <ul style="list-style-type: none"> <li>• ExtendedController: CR0200</li> </ul>   |

## 7.2.5 ifm CANopen 库主站/从站

2718

这些库基于 CODESYS 库（3S CANopen 功能元件），并以简单的方式使功能对用户可用。

| 文件名   | 功能  | 可用于：  |
|---|---|---|
| ifm_CRnnnn_CANopenMaster_Vxxyzz.LIB<br>ifm_CRnnnn_CANopenXMaster_Vxxyzz.LIB<br>ifm_CRnnnn_CANxopenMaster_Vxxyzz.LIB | CANopen master emergency and status handler | <ul style="list-style-type: none"> <li>• 所有 ecomatmobile 控制器 *)</li> <li>• PDM: CR10nn *)</li> </ul>  |
| ifm_CRnnnn_CANopenSlave_Vxxyzz.LIB<br>ifm_CRnnnn_CANopenXSlave_Vxxyzz.LIB<br>ifm_CRnnnn_CANxopenSlave_Vxxyzz.LIB    | CANopen 从站紧急和状态处理程序                         | <ul style="list-style-type: none"> <li>• 所有 ecomatmobile 控制器 *)</li> <li>• PDM: CR10nn *)</li> </ul>  |
| ifm_CANx_SDO_Vxxyzz.LIB   | CANopen SDO 读取和 SDO 写入                      | <ul style="list-style-type: none"> <li>• PDM360: CR1050, CR1051</li> <li>• PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul>   |
| ifm_CANopen_NT_Vxxyzz.LIB   | CAN 堆栈中的 CANopen 功能元件                       | <ul style="list-style-type: none"> <li>• BasicController: CR040n, CR041n, CR043n</li> <li>• BasicDisplay: CR045n</li> <li>• PDM360 NG: CR108n, CR120n</li> <li>• SmartController: CR253n</li> </ul> |

\*) 但不可用于...

- BasicController: CR040n, CR041n, CR043n
- BasicDisplay: CR045n
- PDM360 NG: CR108n, CR120n
- SmartController: CR253n

## 7.2.6 CODESYS CANopen 库

2719

对于以下装置，这些库不可用：

- BasicController: CR040n, CR041n, CR043n
- BasicDisplay: CR045n
- PDM360 NG: CR108n, CR120n
- SmartController: CR253n

| 文件名                                 | 功能             | 可用于：  |
|-------------------------------------|----------------|---|
| 3S_CanDrvOptTableEx.LIB             | CANopen 驱动程序   | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• PDM360smart: CR1070, CR1071</li> </ul>               |
| 3S_CanDrv.LIB <sup>1)</sup>         |                | <ul style="list-style-type: none"> <li>• PDM360: CR1050, CR1051</li> <li>• PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul> |
| 3S_CANopenDeviceOptTableEx.LIB      | CANopen 从站驱动程序 | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• PDM360smart: CR1070, CR1071</li> </ul>               |
| 3S_CANopenDevice.LIB <sup>1)</sup>  |                | <ul style="list-style-type: none"> <li>• PDM360: CR1050, CR1051</li> <li>• PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul> |
| 3S_CANopenManagerOptTableEx.LIB     | CANopen 网络管理器  | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• PDM360smart: CR1070, CR1071</li> </ul>               |
| 3S_CANopenManager.LIB <sup>1)</sup> |                | <ul style="list-style-type: none"> <li>• PDM360: CR1050, CR1051</li> <li>• PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul> |
| 3S_CANopenMasterOptTableEx.LIB      | CANopen 主站     | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• PDM360smart: CR1070, CR1071</li> </ul>               |
| 3S_CANopenMaster.LIB <sup>1)</sup>  |                | <ul style="list-style-type: none"> <li>• PDM360: CR1050, CR1051</li> <li>• PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul> |
| 3S_CANopenNetVarOptTableEx.LIB      | 用于网络变量的驱动程序    | <ul style="list-style-type: none"> <li>• 所有 <b>ecomatmobile</b> 控制器</li> <li>• PDM360smart: CR1070, CR1071</li> </ul>               |
| 3S_CANopenNetVar.LIB <sup>1)</sup>  |                | <ul style="list-style-type: none"> <li>• PDM360: CR1050, CR1051</li> <li>• PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul> |

<sup>1)</sup> 对于以下装置：此库没有用作占位符的功能：

- BasicController: CR040n, CR041n, CR043n
- BasicDisplay: CR045n
- PDM360 NG: CR108n, CR120n
- SmartController: CR253n



## 7.2.7 特定 ifm 库

2720

| 文件名                          | 功能                                 | 可用于：  |
|------------------------------|------------------------------------|---|
| ifm_RawCAN_NT_Vxxyyzz.LIB    | CAN 堆栈中的 CANopen 功能元件，<br>基于第 2 层  | <ul style="list-style-type: none"> <li>BasicController: CR040n, CR041n, CR043n</li> <li>BasicDisplay: CR045n</li> <li>PDM360 NG: CR108n, CR120n</li> <li>SmartController: CR253n</li> </ul>   |
| ifm_J1939_NT_Vxxyyzz.LIB     | CAN 堆栈中的 J1939 通信功能元件              | <ul style="list-style-type: none"> <li>BasicController: CR040n, CR041n, CR043n</li> <li>BasicDisplay: CR045n</li> <li>PDM360 NG: CR108n, CR120n</li> <li>SmartController: CR253n</li> </ul>   |
| ifm_NetVarLib_NT_Vxxyyzz.lib | 用于网络变量的附加驱动程序                      | <ul style="list-style-type: none"> <li>BasicController: CR040n, CR041n, CR043n</li> <li>BasicDisplay: CR045n</li> <li>PDM360 NG: CR108n, CR120n</li> <li>SmartController: CR253n</li> </ul>   |
| ifm_J1939_Vxxyyzz.LIB        | J1939 通信功能元件                       | 高达运行时系统 V04： <ul style="list-style-type: none"> <li>CabinetController: CR0303</li> <li>ClassicController: CR0020, CR0505</li> <li>ExtendedController: CR0200</li> <li>SafetyController: CR7020, CR7200, CR7505</li> <li>SmartController: CR2500</li> <li>PDM360smart: CR1070, CR1071</li> </ul>   |
| ifm_J1939_x_Vxxyyzz.LIB      | J1939 通信功能元件                       | 从运行时系统 V05 开始： <ul style="list-style-type: none"> <li>CabinetController: CR0303</li> <li>ClassicController: CR0020, CR0505</li> <li>ExtendedController: CR0200</li> <li>SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506</li> <li>SmartController: CR2500</li> <li>sPDM360smart: CR1070, CR1071</li> </ul>                        |
| ifm_CRnnnn_J1939_Vxxyyzz.LIB | J1939 通信功能元件                       | <ul style="list-style-type: none"> <li>Controller: CR0n3n, CR7n3n</li> </ul>  |
| ifm_PDM_J1939_Vxxyyzz.LIB    | J1939 通信功能元件                       | <ul style="list-style-type: none"> <li>PDM360: CR1050, CR1051</li> <li>PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul>   |
| ifm_CANx_LAYER2_Vxxyyzz.LIB  | CAN 功能元件，基于第 2 层：<br>CAN 传输、CAN 接收 | <ul style="list-style-type: none"> <li>PDM360: CR1050, CR1051</li> <li>PDM360compact: CR1052, CR1053, CR1055, CR1056</li> </ul>   |
| ifm_CAN1E_Vxxyyzz.LIB        | 将 CAN 总线从 11 位更改为 29 位             | 高达运行时系统 V04： <ul style="list-style-type: none"> <li>PDM360smart: CR1070, CR1071</li> </ul>  |
| ifm_CAN1_EXT_Vxxyyzz.LIB     | 将 CAN 总线从 11 位更改为 29 位             | 从运行时系统 V05 开始： <ul style="list-style-type: none"> <li>CabinetController: CR030n</li> <li>ClassicController: CR0020, CR0505</li> <li>ExtendedController: CR0200</li> <li>PCB 控制器：CS0015</li> <li>SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506</li> <li>SmartController: CR250n</li> <li>PDM360smart: CR1070, CR1071</li> </ul> |
| ifm_CAMERA_O2M_Vxxyyzz.LIB   | 摄像头功能元件                            | <ul style="list-style-type: none"> <li>PDM360：CR1051CR1051</li> </ul>   |
| CR2013AnalogConverter.LIB    | I/O 模块 CR2013 的模拟值转换               | <ul style="list-style-type: none"> <li>所有 ecomatmobile 控制器</li> <li>PDM: CR10nn</li> </ul>  |

## 使用的文件和库概述

## 单个文件和库有什么用途？

| 文件名                                   | 功能   | 可用于：   |
|---------------------------------------|--|--|
| ifm_Hydraulic_16bit0S04_Vxxyyzz.LIB   | R360 控制器的液压功能元件                                    | 高达运行时系统 V04：<br>• ClassicController: CR0020, CR0505<br>• ExtendedController: CR0200<br>• SafetyController: CR7020, CR7200, CR7505<br>• SmartController: CR250n                           |
| ifm_Hydraulic_16bit0S05_Vxxyyzz.LIB   | R360 控制器的液压功能元件                                    | 从运行时系统 V05 开始：<br>• ClassicController: CR0020, CR0505<br>• ExtendedController: CR0200<br>• SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506<br>• SmartController: CR250n |
| ifm_Hydraulic_32bit_Vxxyyzz.LIB       | R360 控制器的液压功能元件                                    | • Controller: CR0n3n, CR7n3n   |
| ifm_Hydraulic_CR0303_Vxxyyzz.LIB      | R360 控制器的液压功能元件                                    | • CabinetController: CR0303  |
| ifm_SafetyIO_Vxxyyzz.LIB              | 安全功能元件   | • SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506   |
| ifm_SafetyPLCopen_Vxxyyzz.LIB         | 安全功能元件   | • SafetyController: CR7032, CR7132   |
| ifm_PDM_UTIL_Vxxyyzz.LIB              | 辅助功能 PDM   | • PDM360: CR1050, CR1051<br>• PDM360compact: CR1052, CR1053, CR1055, CR1056  |
| ifm_PDMng_UTIL_Vxxyyzz.LIB            | 辅助功能 PDM   | • PDM360 NG: CR1082, CR1083, CR1084, CR1085  |
| ifm_PDMsmart_UTIL_Vxxyyzz.LIB         | 辅助功能 PDM   | • PDM360smart: CR1070, CR1071  |
| ifm_PDM_Input_Vxxyyzz.LIB             | 替代输入功能元件 PDM                                       | • PDM: CR10nn  |
| ifm_CR107n_Init_Vxxyyzz.LIB           | 初始化功能元件 PDM360smart                                | • PDM360smart: CR1070, CR1071  |
| ifm_PDM_File_Vxxyyzz.LIB              | 文件功能元件 PDM360                                      | • PDM360: CR1050, CR1051<br>• PDM360compact: CR1052, CR1053, CR1055, CR1056<br>• PDM360 NG: CR1082, CR1083, CR1084, CR1085   |
| ifm_PDM360NG_linux_syscall_asynch_LIB | 将 Linux 命令发送至系统                                    | • PDM360 NG: CR1082, CR1083, CR1084, CR1085  |
| ifm_PDM360NG_USB_Vxxyyzz.LIB          | 通过 USB 接口管理装置                                      | • PDM360 NG: CR1082, CR1083, CR1084, CR1085  |
| ifm_PDM360NG_USB_LL_Vxxyyzz.LIB       | auxiliary library for ifm_PDM360NG_USB_Vxxyyzz.LIB | • PDM360 NG: CR1082, CR1083, CR1084, CR1085  |
| Instrumente_x.LIB                     | 预定义的显示仪器   | • PDM: CR10nn  |
| Symbols_x.LIB                         | 预定义符号  | • PDM360: CR1050, CR1051<br>• PDM360compact: CR1052, CR1053, CR1055, CR1056  |
| Segment_x.LIB                         | 预定义的 7 段显示屏  | • PDM360: CR1050, CR1051<br>• PDM360compact: CR1052, CR1053, CR1055, CR1056  |

可按要求提供其他库。

## 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](http://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](http://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"。



© ifm electronic gmbh

## MMI

→ HMI (→ 页 [159](#))

## 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船载诊断实施指南
- 第 11 部分：物理层 – 250 kBits/s，屏蔽双绞线
- 第 13 部分：场外诊断连接器
- 第 15 部分：简化物理层，250 kBits/s，非屏蔽双绞线 (UTP)
- 第 21 部分：数据链路层
- 第 31 部分：网络层
- 第 71 部分：车辆应用层
- 第 73 部分：应用层 - 诊断
- 第 81 部分：网络管理协议

### SD 卡

SD 内存卡（**安全数字**内存卡的简称）是根据 → 闪速存储原理运行的数字存储媒介。

### SDO

SDO = **服务数据对象**。

SDO 用于访问 CANopen 对象目录中的对象。“客户端”向“服务器”请求所需数据。SDO 始终包含 8 个字节。

**例如：**

- 在系统启动时通过 →SDO 自动配置所有从站，
- 读取 →对象目录中的错误消息。

每个 SDO 均接受响应监控，如果从站未在监控时间内响应，则重复 SDO。

## T

### TCP

**传输控制协议**是 TCP/IP 协议系列的一部分。每个 TCP/IP 数据连接均有一个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

处理整线的整体安排。主站决定总线访问时间并定期→轮询从站。

### 对象目录

包含装置的所有 CANopen 通信参数以及装置特定参数和数据。

### 电子狗

一般而言, “电子狗”一词表示监控其他组件功能的系统组件。如果检测到可能的故障, 则向其发送信号或启用相应的程序分支。信号或程序分支作为其他协作系统组件的触发器, 以解决问题。

## 目标

目标包含 CODESYS 目标装置的硬件说明，如：输入端和输出端、内存、文件位置。  
对应电子技术资料。

## 节点

即网络中的参与者。

## 节点保护

节点 = 此处：网络参与者

每个从站的可配置循环→监控可相应地加以配置。 →主站核查从站是否及时响应。 从站核查主站是否定期发送请求。 这样即可快速识别和报告出现故障的网络参与者。

## 闪存

快闪 ROM (或快闪 EPROM 或闪存) 将半导体内存和硬盘的优势相结合。 但是跟硬盘类似，数据在高达 64、128、256、1024... 字节的数据块中同时按区块写入和删除。

### 闪存的优势

- 即便无电源电压亦可保留存储的数据。
- 因为没有移动部件，闪存无噪音且对冲击和磁场不敏感。

### 闪存的劣势

- 存储单元可容许有限数量的写入和删除进程：
  - 多级单元：一般为 10 000 个周期
  - 单级单元：一般为 100 000 个周期
- 鉴于写入进程同时写入 16 和 128 K 字节之间的内存块，还可使用无需更改的内存单元。

## 六划

### 地址

这是总线参与者的“名称”。 所有参与者需要唯一的地址，以便无忧更换信号。

### 夹具 15

在车辆中，夹具 15 是通过点火锁开关的正极电缆。

## 自检

主动检验组件或装置的检验程序。程序由用户启动，且将花费一定的时间。结果为显示检验内容以及结果为正面还是负面的检验协议（日志文件）。

## 七划

### 应用程序软件

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

### 抖动频率

抖动频率是 →PWM 信号的组成部分，用于控制液压阀。对液压阀的电磁驱动器来说，如果 PWM 频率的特定频率叠加控制信号（PWM 脉冲），则控制阀门要容易得多。该抖动频率必须是 PWM 频率的整数部分。

### 系统变量

可通过 IEC 地址或符号名称从 PLC 访问的变量。

## 诊断

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

还可监控输入端和输入端是否正常运行，具体视装置而定。

- 断线，
- 短路，
- 值在范围之外。

就诊断而言，可使用装置“正常”运行期间创建的配置和日志数据。

初始化和启动期间可监控系统组件是否正确启动。日志文件记录错误。

就进一步诊断而言，还可进行自检。

## 运行

CANopen 参与者的工作状态 在该模式中可传输→SDO、→NMT 命令和 →PDO。

### 运行时系统

装置中的基本程序，建立装置硬件和应用程序之间的连接。



## 进程图像

进程图像即在一个→周期内 PLC 据其运行的输入端和输出端状态。

- 在周期开始时，PLC 将所有输入端的状况读取至进程图像。周期期间，PLC 无法检测输入端的更改。
- 周期期间，仅可对输出端进行虚拟更改（在进程图像中）。
- 在周期结束时，PLC 将虚拟输出状态写入实际输出端。

## 连接指令

连接指令是指对文件中其他部分或外部文件的交叉引用。

## 八划

### 使用，指定

根据用途说明所载的信息使用产品。

### 周期时间

即周期的时间。PLC 程序展开一次完整的运行。

这可能需要更长或更短的时间，具体视程序中的事件控制分支而定。

### 图标

图标即通过简化的图示传达信息的象征性符号。（→ 章节 **符号和格式是什么意思？**（→ 页 8））

### 波特

波特 (Baud)，缩写：Bd = 数据传送速度单位。切勿将波特与“位/秒 (bps、bits/s)”混淆。波特表示在一定传送长度内每秒的状态（步骤、周期）更改量。但未定义每步传送的位数。Baud 一词可追溯到法国发明家 J. M. Baudot,其编码用于电传机器。

1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

## 九划

### 总线

同一电缆上多个参与者的串行数据传送。

## 指定用途

根据用途说明所载的信息使用产品。

## 架构

系统硬件和/或软件的特定配置。

## 误用

未按照设计方指定的方式使用产品。

产品制造商应在用户信息中针对可预见的误用提出警告。

## 说明

以下术语之一的上义词：

安装说明、技术资料、用户信息、操作说明、装置手册、安装信息、联机帮助、系统手册、编程手册等。

## 十划

### 预运行

预运行 = PRE-OPERATIONAL 模式

CANopen 参与者的工作状态。施加电源电压后，每个参与者自动进入该状态。在 CANopen 网络中，仅 → SDO 和 → NMT 命令可在该模式下传输，但无进程数据。

## 十一划

### 检测信号

参与者定期发送短信号。这样其他参与者则可核实参与者是否出现故障。

### 符号

图标即通过简化的图示传达信息的象征性符号。（→ 章节 **符号和格式是什么意思？**（→ 页 8））

## 十二划

### 剩余

电源故障时剩余数据不会丢失。

当电源电压跌破关键值时→运行时系统等即会自动将剩余数据复制到→闪存。如果电源电压再次可用，则运行时系统将剩余数据加载回 RAM 内存。

但是控制器 RAM 内存中的数据易失，通常会在电源故障时丢失。

### 嵌入式软件

装置中的系统软件、基本程序，实际上→是运行时系统中。

固件建立装置硬件和应用程序之间的连接。固件作为系统的一部分由控制器制造商提供，且用户不可更改。

## 十三划

### 数据类型

可存储不同大小的值，具体式数据类型而定。

| 数据类型   | 最小值                          | 最大值                         | 内存大小         |
|--------|------------------------------|-----------------------------|--------------|
| BOOL   | FALSE                        | TRUE                        | 8 位 = 1 个字节  |
| BYTE   | 0                            | 255                         | 8 位 = 1 个字节  |
| WORD   | 0                            | 65 535                      | 16 位 = 2 个字节 |
| DWORD  | 0                            | 4 294 967 295               | 32 位 = 4 个字节 |
| SINT   | -128                         | 127                         | 8 位 = 1 个字节  |
| USINT  | 0                            | 255                         | 8 位 = 1 个字节  |
| INT    | -32 768                      | 32 767                      | 16 位 = 2 个字节 |
| UINT   | 0                            | 65 535                      | 16 位 = 2 个字节 |
| DINT   | -2 147 483 648               | 2 147 483 647               | 32 位 = 4 个字节 |
| UDINT  | 0                            | 4 294 967 295               | 32 位 = 4 个字节 |
| REAL   | $-3.402823466 \cdot 10^{38}$ | $3.402823466 \cdot 10^{38}$ | 32 位 = 4 个字节 |
| ULINT  | 0                            | 18 446 744 073 709 551 615  | 64 位 = 8 个字节 |
| STRING |                              |                             | 字符数量 + 1     |

## 十四划

### 模板

模板可填入内容。

此处：预配置软件元件结构，作为应用程序的基础。

# 9 索引

## 符号

[Base settings] 选项卡 ..... 53  
 [CAN settings] 选项卡 ..... 57  
 [Default PDO mapping] 选项卡 ..... 58  
 [Receive PDO-Mapping] 和 [Send PDO-Mapping] 选项卡 ..... 38

## B

Bootloader ..... 155

## C

CAN ..... 83, 155  
     交换数据 ..... 24  
     接口和协议 ..... 23  
 CAN / CANopen  
     错误和错误处理 ..... 89  
 CAN / CANopen / 网络变量摘要 ..... 83  
 CAN 波特率 ..... 32  
 CAN 参数：无初始化 ..... 36  
 CAN 参数：节点 ID ..... 33, 35  
 CAN 参数：节点复位 ..... 36  
 CAN 参数：节点保护 / 检测信号设置 ..... 36  
 CAN 参数：可选装置 ..... 36  
 CAN 参数：同步 COB ID ..... 33  
 CAN 参数：自动启动 ..... 34  
 CAN 参数：创建所有 SDO ..... 36  
 CAN 参数：波特率 ..... 32  
 CAN 参数：信息 ..... 37  
 CAN 参数：紧急报文 ..... 36  
 CAN 参数：通讯周期间隔/同步 窗口长度 ..... 33  
 CAN 参数：通信周期 ..... 37  
 CAN 参数：检测信号 ..... 34  
 CAN 参数：编写 DCF ..... 36  
 CAN 堆栈 ..... 155  
 CAN 接口 ..... 23  
 CAN 错误 ..... 89, 90  
 CANopen ..... 83  
 CANopen EMCY 代码概述 (CANx) ..... 102  
 CANopen EMCY 代码概述 (扩展页) ..... 101  
 CANopen EMCY 代码概述 (标准页) ..... 100  
 CANopen 从站 ..... 51  
 CANopen 从站：[CAN parameters] 选项卡 ..... 35

CANopen 从站：注册 [Service Data Objects] ..... 39  
 CANopen 从站库的功能 ..... 51  
 CANopen 从站启动 ..... 46  
 CANopen 从站的 NMT 状态 ..... 71  
 CANopen 术语和实现 ..... 27  
 CANopen 主站 ..... 28  
 CANopen 主站：[CAN parameters] 选项卡 ..... 32  
 CANopen 主站启动 ..... 45  
 CANopen 主站的 NMT 状态 ..... 71  
 CANopen 主站的对象目录 ..... 48  
 CANopen 网络变量 ..... 74  
 CANopen 网络变量的配置 ..... 75  
 CANopen 网络配置、状态和错误处理 ..... 26  
 CANopen 报文的结构 ..... 63  
 CANopen 库 ..... 28  
 CANopen 库的功能 ..... 29  
 CANopen 表 ..... 63  
 CANopen 错误 ..... 93  
 CANopen 错误代码概述 ..... 95  
 CiA ..... 155  
 CiA DS 304 ..... 155  
 CiA DS 401 ..... 155  
 CiA DS 402 ..... 155  
 CiA DS 403 ..... 156  
 CiA DS 404 ..... 156  
 CiA DS 405 ..... 156  
 CiA DS 406 ..... 156  
 CiA DS 407 ..... 156  
 COB ID ..... 156  
 COB ID 的结构 ..... 64  
 CODESYS ..... 156  
 CODESYS CANopen 库 ..... 152  
 CoDeSys 支持 CANopen ..... 27  
 CODESYS 编程手册 ..... 7  
 Cr045n 中的颜色 ..... 145  
 Cr108n 中的颜色 ..... 145  
 CSV 文件 ..... 157  
 CSV 文件和 ifm 维护工具 ..... 117  
 CSV 文件的要求 ..... 118

## D

DC ..... 157  
 DLC ..... 157  
 DRAM ..... 157  
 DTC ..... 157

|   |     |                      |     |
|---|-----|----------------------|-----|
| <b>E</b>  |     | <b>M</b>             |     |
| ECU .....   | 157 | MAC-ID .....         | 160 |
| EDS 文件 .....  | 158 | MMI .....            | 162 |
| EMC .....   | 158 | MRAM .....           | 162 |
| EMCY .....  | 158 | MSB .....            | 162 |
| EMCY 报文的结构 .....  | 93  | <b>N</b>             |     |
| EMCY 错误代码 .....   | 95  | NMT .....            | 162 |
| Ethernet .....  | 158 | NMT 状态 .....         | 70  |
| EUC .....   | 158 | <b>O</b>             |     |
| <b>F</b>  |     | Obj/对象 .....         | 162 |
| FiFo .....  | 158 | OBV .....            | 162 |
| FRAM .....  | 159 | OPC .....            | 162 |
| <b>H</b>  |     | <b>P</b>             |     |
| HMI .....   | 159 | PC 卡 .....           | 162 |
| <b>I</b>  |     | PCMCIA 卡 .....       | 163 |
| ID .....  | 159 | PDM .....            | 163 |
| IEC 61131 .....   | 159 | PDO .....            | 163 |
| IEC 用户周期 .....  | 159 | PDO 映射：插入 .....      | 38  |
| ifm CANopen 库主站/从站 .....  | 151 | PDO 映射：属性 .....      | 38  |
| ifm weltweit • ifm worldwide • ifm à l'échelle internationale ..... | 183 | PDU .....            | 163 |
| ifm 装置库 .....   | 151 | PES .....            | 163 |
| ifm 模板是什么？ .....  | 10  | PGN .....            | 163 |
| ifm 演示程序 .....  | 20  | PID 控制器 .....        | 163 |
| ifm 演示程序是什么？ .....  | 11  | PLC 配置 .....         | 164 |
| IP 地址 .....   | 159 | PLC 配置文件 .....       | 150 |
| ISO 10646_ 信息技术 - 通用多八位编码字符集 (UCS) .....                            | 139 | PWM .....            | 164 |
| ISO 11898 .....   | 160 | PWM 输出端的作用是什么？ ..... | 105 |
| ISO 11992 .....   | 160 | 什么是抖动？ .....         | 106 |
| ISO 13406_ 平板视觉显示工作的人类工效学要求 .....                                   | 140 | 功能块 .....            | 109 |
| ISO 13407_ 以人为本的交互系统设计过程 .....                                      | 140 | 参数描述 .....           | 109 |
| ISO 16845 .....   | 160 | PWM 功能 - 描述 .....    | 103 |
| ISO 20282_ 日常用品的易操作性 .....  | 141 | PWM 抖动 .....         | 109 |
| ISO 7001_ 图形符号 - 公共信息符号 .....                                       | 135 | PWM 信号处理 - 描述 .....  | 104 |
| ISO 9126_ 软件工程 - 产品质量 .....   | 136 | PWM 频率 .....         | 109 |
| ISO 9241_ 人-系统交互人类工效学 .....   | 137 | <b>R</b>             |     |
| ISO 9241-11_ 使用指南 .....   | 138 | RAW-CAN .....        | 164 |
| ISO 9241-110_ 对话框原则 .....   | 138 | ro .....             | 164 |
| <b>J</b>  |     | RTC .....            | 164 |
| J1939 .....   | 160 | rw .....             | 164 |
| <b>L</b>  |     | <b>S</b>             |     |
| LED .....   | 160 | SAE J1939 .....      | 165 |
| LSB .....   | 160 | 标识符 .....            | 85  |
|   |     | SD 卡 .....           | 165 |

**索引**

**单个文件和库有什么用途？**

SDO .....165  
 SDO 中止代码 .....67  
 SDO 命令字节 .....66  
 SDOs  
     更改值 .....39

**T**

TCP .....165

**U**

UDP .....166

**一划**

一次启动所有从站 .....42  
 一次复位所有从站 .....41  
 一般信息 .....74  
 一般概述 .....148

**三划**

已停止 .....166

**四划**

开启扭矩 .....106  
 无自平衡的受控系统 .....114  
 不带补偿的受控系统 .....114  
 比率式 .....166  
 什么是 CSV 文件？ .....116  
 从主站到从站的检测信号 .....42  
 从站 .....166  
 文化底蕴通常不可转换 .....132  
 为项目补充更多功能 .....17  
 以下错误有所不同： .....94

**五划**

功能代码 / 预定义连接组 .....65  
 节点 .....167  
 节点的 CANopen 状态 .....73  
 节点保护 .....167  
 本手册中描述了哪些装置？ .....7  
 本文档的结构是怎样的？ .....9  
 目标 .....167  
 目标文件 .....150  
 目标设置中的设置 .....76  
 电子狗 .....166

由应用程序访问 OD 条目 .....61  
 用于运行时系统的文件 .....150  
 用于驱动工程的 CAN .....84  
 用户友好的产品设计的建议 .....127  
 用户界面建议 .....126  
 处理 .....89  
 主站 .....166  
 主站运行时 .....40  
 闪存 .....167  
 闪存 - 什么是闪存？ .....115  
 加色法混色 .....146  
 发出装置错误信号 .....98  
 对象 0x1001 ( 错误寄存器 ) .....97  
 对象 0x1003 ( 错误场 ) .....97  
 对象目录 .....166  
 对象目录示例 .....55

**六划**

地址 .....167  
 在系统启动时，在总线上复位所有已配置从站 .....40  
 有关 CANopen 的技术详情 .....26  
 有关位图图形的特殊信息 .....146  
 有关带有 CODESYS 的 CANopen 的一般信息 .....27  
 有关颜色和位图图形的基本信息 .....142  
 夹具 15 .....167  
 此库的用途是什么？ - 简介 .....111  
 网络状态 .....45  
 网络变量 .....83  
 网络变量的特殊性 .....82  
 网络管理 (NMT) .....69  
 网络管理命令 .....69  
 传输数据 .....25  
 自动配置从站 .....41  
 自检 .....168  
 自衡过程 .....113  
 全局变量列表：包变量 .....79  
 全局变量列表：写入 .....81  
 全局变量列表：发生事件时传输 .....81  
 全局变量列表：列表标识符 (COB-ID) .....79  
 全局变量列表：网络类型 .....78  
 全局变量列表：传输校验和 .....79  
 全局变量列表：更改时传输 .....81  
 全局变量列表：周期性传输 .....81

**索引**

**单个文件和库有什么用途？**

全局变量列表：读取.....79  
 全局变量列表：确认.....79  
 全局变量列表中的设置.....77  
 创建 CANopen 项目.....30  
 关于 IFM 模板.....13  
 关于本手册.....5  
 设置 CANopen 从站的节点编号.....61  
 设置 CANopen 从站的波特率.....61  
 访问 CANopen 主站的状态.....47  
 访问闪存数据：功能块.....124  
 访问对象目录（其他）.....50  
 访问对象目录（控制器）.....48

**七划**

进程图像.....169  
 运行.....168  
 运行时访问 CANopen 从站.....61  
 运行时更改 PDO 属性.....61  
 运行时系统.....168  
 抖动示例.....108  
 抖动在什么时候有用？.....107  
 抖动频率.....106, 109, 168  
 抖动频率和幅度.....108  
 连接指令.....169  
 位图图形的颜色.....145  
 系统变量.....168  
 库：CANopen 的系统需要.....28  
 应用程序软件.....168  
 快速参考指南：ifm 模板.....10  
 快速参考指南：ifm 演示程序.....11  
 启动 CANopen 网络.....43  
 启动网络，无 [Automatic startup].....46  
 启动报文.....68  
 启动所有配置正确的从站.....41  
 诊断.....168

**八划**

轮询从站装置类型.....41  
 图示.....134  
 图标.....8, 169  
 图像大小矢量图形 / 像素图形.....143  
 制造商特定信息.....99  
 使用 CAN - 描述.....23

使用 GLOBAL\_START 启动网络.....46  
 使用 PWMi 进行液压控制.....111  
 使用 RESET\_ALL\_NODES 初始化网络.....47  
 使用 START\_ALL\_NODES 启动网络.....47  
 使用，指定.....169  
 使用电子数据表程序创建 CSV 文件.....119  
 使用用户闪存.....115  
 使用的文件和库概述.....148  
 使用维护工具传输 CSV 文件.....123  
 使用编辑器创建 CSV 文件.....121  
 例如：变量列表.....59  
 例如：详细报文文档.....86  
 例如：短报文文档.....87  
 例如：缩小 Cr108n 的像素图像.....143  
 版权.....5  
 周期时间.....169  
 周期性传输 SYNC 报文.....42  
 备注.....178  
 变量列表示例.....59  
 单个文件和库有什么用途？.....150  
 波特.....169  
 参与方，主动错误.....91  
 参与方总线断开.....92  
 参与方被动错误.....91

**九划**

指令和标准.....135  
 指定用途.....170  
 按照 SAE J1939 的标识符.....85  
 带有电流控制输出端的液压控制阀.....112  
 带延迟的受控系统.....114  
 带延迟的受控系统.....114  
 带寿命监控的节点保护.....42  
 带补偿的受控系统.....113  
 标识符.....94  
 标准化控制杆的输出信号.....111  
 显示哪些颜色？.....145  
 哪些图形适合哪些 PDM 以及必须执行哪些步骤？.....147  
 怎样快速、简单地设置编程系统？（例如，CR2500）.....15  
 脉冲宽度调制.....105  
 总线.....169  
 语言障碍.....130  
 误用.....170



|                    |     |
|--------------------|-----|
| 说明 .....           | 170 |
| 说明沿革 (App-C) ..... | 9   |
| 架构 .....           | 170 |

## 十划

|                     |     |
|---------------------|-----|
| 配置 CANopen 从站 ..... | 53  |
| 配置所有正确检测到的装置 .....  | 41  |
| 逐个启动从站 .....        | 41  |
| 逐个复位从站 .....        | 40  |
| 紧急报文 .....          | 94  |
| 特定 ifm 库 .....      | 153 |
| 阅读方向 .....          | 134 |
| 调整位图图形 .....        | 144 |
| 通过主站配置更改标准映射 .....  | 60  |
| 通过应用程序发送紧急报文 .....  | 62  |
| 通过模板设置编程系统 .....    | 12  |
| 通常的文件夹结构 .....      | 13  |
| 预运行 .....           | 170 |
| 预定义连接组 .....        | 65  |

## 十一划

|                        |          |
|------------------------|----------|
| 接收紧急报文 .....           | 43       |
| 控制输出端 - 描述 .....       | 103      |
| 控制器 - 描述 .....         | 113      |
| 控制器的演示程序 .....         | 20       |
| 基本设置：生成 EDS 文件 .....   | 54       |
| 基本设置：更新任务的名称 .....     | 53       |
| 基本设置：总线标识符 .....       | 53       |
| 检查适用性 .....            | 129      |
| 检测信号 .....             | 170      |
| 符号 .....               | 134, 170 |
| 符号和格式是什么意思？ .....      | 8        |
| 您是否了解未来的用户？ .....      | 128      |
| 添加和配置 CANopen 从站 ..... | 35       |

## 十二划

|  |     |
|--|-----|
| 插入 CANopen 从站 (例如：CR2500 <-- CR2011) ..... | 16  |
| 嵌入式软件 .....                                | 171 |
| 幅度 .....                                   | 108 |
| 剩余 .....                                   | 171 |
| 装置中的可视化 .....                              | 125 |

## 十三划

|                              |         |
|------------------------------|---------|
| 概述：ecomatmobile 装置文档模块 ..... | 6       |
| 概要 .....                     | 27, 125 |
| 频率 .....                     | 108     |
| 错误                           |         |
| -计数器 .....                   | 90      |
| 错误计数器 .....                  | 90      |
| 错误报文 .....                   | 90      |
| 简介 .....                     | 10      |
| 数据类型 .....                   | 171     |
| 数据接收 .....                   | 24      |

## 十四划

|                          |     |
|--------------------------|-----|
| 模板 .....                 | 172 |
| 模板 (C) 的文件夹中的程序和功能 ..... | 14  |
| 模板和演示程序 .....            | 10  |

## 十五划

|          |     |
|----------|-----|
| 颜色 ..... | 132 |
|----------|-----|

## 10 备注



© ifm electronic gmbh

www.ifm.com

© ifm electronic gmbh

© ifm electronic gmbh

© ifm electronic gmbh



www.ifm.com

© ifm electronic gmbh

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

Version: 2016-11-29

ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen

[www.ifm.com](http://www.ifm.com) • Email: [info@ifm.com](mailto: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<br>Niederlassung West • 45128 Essen • Tel. 02 01 / 3 64 75 -0<br>Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 0 23 51 / 43 01-0<br>Niederlassung Süd-West • 64646 Heppenheim • Tel. 0 62 52 / 79 05-0<br>Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 0 70 21 / 80 86-0<br>Niederlassung Bayern • 82178 Puchheim • Tel. 0 89 / 8 00 91-0<br>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 ą 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.