

使用 Unity Pro 的 Modicon M340 CANopen

用户手册

目录



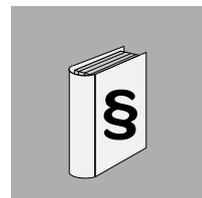
关于本书	9
重要信息	7
部分 I CANopen 通讯概述	11
概览	11
章 1 CANopen 通讯概述	13
概览	13
原理	14
CAN 概览	15
CANopen 现场总线的一般架构	18
拓扑结构	20
CANopen 网络的长度限制	23
符合性分类	25
部分 II CANopen 硬件实施	27
概览	27
章 2 BMX P34 处理器的硬件实现	29
概览	29
处理器描述: BMX P34 2010/2030	30
安装	31
CANopen 处理器的可视化诊断	32
章 3 CANopen 设备简介	35
概览	35
CANopen 设备	36
CANopen 运动命令设备	37
CANopen 输入 / 输出设备	43
其他设备	47

部分 III	CANopen 通讯的软件实现	53
	概览	53
章 4	一般信息	55
	概览	55
	实施原理	56
	实现方法	57
	性能	58
	设备性能	61
章 5	CANopen 总线上的通讯配置	65
	概览	65
5.1	一般信息	66
	一般信息	67
5.2	总线配置	68
	概览	68
	如何访问 CANopen 总线配置屏幕	69
	CANopen 总线编辑器	71
	如何在总线上添加设备	73
	如何删除 / 移动 / 复制总线设备	75
	在项目浏览器中查看 CANopen 总线	77
5.3	设备配置	78
	概览	78
	从站功能	79
	使用 Unity 进行配置	83
	使用外部工具的配置：配置软件	89
	手动配置	92
5.4	主站配置	94
	概览	94
	如何访问 CANopen 主站配置屏幕	95
	CANopen 主站配置屏幕	97
	主站配置屏幕描述	99
章 6	Catalog Manager 软件实现	103
	概览	103
6.1	Catalog Manager 概述	104
	概览	104
	Catalog Manager 描述	105
	Catalog Manager 内容	107
6.2	使用 Catalog Manager	111
	概览	111
	如何启动 Catalog Manager	112
	如何将设备添加到 Catalog Manager	113
	如何在设备上添加功能	116
	基本配置参数	117
	Expert Mode 配置参数	121
	如何复制或删除功能	129

	如何导入 / 导出或删除一个或多个用户设备.....	130
	如何关闭 Catalog Manager	133
	如何创建专用和优化 STB 岛的示例	134
6.3	Catalog Manager 故障排除	135
	故障排除	136
章 7	编程.....	141
	概览	141
	使用 PDO 进行交换	142
	使用 SDO 进行交换	147
	通讯功能示例	150
	Modbus 请求示例	157
章 8	在 CANopen 总线上调试通讯.....	159
	概览	159
	如何访问远程设备的调试屏幕	160
	CANopen 主站的调试屏幕	161
	从站调试屏幕	163
章 9	诊断.....	165
	概览	165
	如何执行诊断	166
	主站诊断	167
	从站诊断	168
章 10	语言对象.....	171
	概览	171
10.1	CANopen 通讯的语言对象和 IODDT	172
	概览	172
	CANopen 通讯语言对象简介	173
	与应用专用功能关联的隐式交换语言对象	174
	与应用专用功能关联的显式交换语言对象	175
	使用显式对象管理交换和报告	177
10.2	适用于所有通讯协议的语言对象和一般 IODDT	179
	概览	179
	T_COM_STS_GEN 类型的 IODDT 隐式交换对象的详细信息	180
	T_COM_STS_GEN 类型的 IODDT 显式交换对象的详细信息	181
10.3	CANopen 专用 IODDT 的语言对象	183
	概览	183
	T_COM_CO_BMX 类型 IODDT 的隐式交换对象的详细信息	184
	T_COM_CO_BMX 类型 IODDT 的显式交换对象的详细信息	194
	与配置关联的语言对象	196
10.4	紧急对象	197
	紧急对象	198
10.5	适用于所有模块的 IODDT 类型的 T_GEN_MOD	201
	T_GEN_MOD 类型 IODDT 的语言对象的详细信息	202

部分 IV	快速入门: CANopen 实现示例	203
	概览	203
章 11	应用程序描述	205
	应用程序概述	206
章 12	使用 Unity Pro 安装应用程序	209
	概览	209
12.1	所采用的解决方案简介	210
	概览	210
	可选用的技术	211
	使用 Unity Pro 的过程中的各个步骤	212
12.2	开发应用程序	213
	概览	213
	创建项目	214
	CANopen 总线配置	215
	CANopen 主站配置	220
	设备配置	221
	变量声明	224
	在 SFC 中创建用于管理移动序列的程序	227
	使用 LD 创建程序以用于应用程序的执行	231
	使用 LD 为操作员屏幕动态显示创建程序	233
	使用 ST 为 Lexium 配置创建程序	234
	创建动态数据表	238
	创建操作员屏幕	240
章 13	启动应用程序	243
	在标准模式下执行应用程序	244
	附录	251
章 14	CANopen 主站本地对象字典条目	253
	概览	253
	符合配置文件 DS301 的对象字典条目	254
	符合配置文件 DS302 的对象字典条目	259
	中型制造商专用对象字典条目	261
章 15	PDO 和 STB 变量之间的关系	267
	STB 岛配置	268
章 16	动作和转换	271
	概览	271
	转换	272
	操作	273
术语	Modicon M340 的 CANopen 术语	275

安全信息



重要信息

声明

在尝试安装、操作或维护设备之前，请仔细阅读下述说明并通过查看来熟悉设备。下述特别信息可能会在本文其他地方或设备上出现，提示用户潜在的危險，或者提醒注意有关阐明或简化某一过程的信息。



在“危險”或“警告”安全标签上添加此符号表示存在触电危險，如果不遵守使用说明，将导致人身伤害。



这是提醒注意安全的符号。提醒用户可能存在人身伤害的危險。请遵守所有带此符号的安全注意事项，以避免可能的人身伤害甚至死亡。

危險

“危險”表示极可能存在危險，如果不遵守说明，可**导致**严重的人身伤害甚至死亡。

警告

“警告”表示可能存在危險，如果不遵守说明，可**导致**严重的人身伤害甚至死亡，或设备损坏。

 **注意**

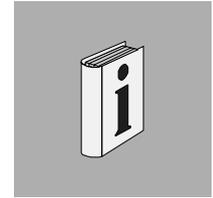
“注意”表示可能存在危险，如果不遵守说明，可 **导致**严重的人身伤害或设备损坏。

请注意

电气设备的安装、操作、维修和维护工作仅限于合格人员执行。对于使用本资料所引发的任何后果，Schneider Electric 概不负责。

(c) 2008 Schneider Electric. 保留所有权利。

关于本书



浏览

文档范围

本手册描述 Modicon M340 系列 PLC 上的 CANopen 网络实施。

有效性

此文档中给出的数据和示意图并不是一成不变的。我们保留根据持续产品开发策略修改我们的产品的权利。

本文档中的信息如有更改，恕不另行通知，并且不应理解为 Schneider Electric 承担的义务。

与产品相关的警告

警告

意外的设备操作

应用此产品要求在控制系统的设计和编程方面有经验。只允许具有此类经验的人士编程、安装、改动和应用此产品。

请遵守所有当地和国家 / 地区的安全法规和标准。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

对于本文档中可能出现的任何错误，Schneider Electric 概不负责。如果您有关于改进或更正此出版物的任何建议，或者从中发现错误，请通知我们。

未经 Schneider Electric 明确书面许可，不得以任何形式、通过任何电子或机械手段（包括复印）复制本文档的任何部分。

在安装和使用本产品时，必须遵守国家 / 地区、区域和当地的所有相关的安全法规。

出于安全方面的考虑和为了确保符合归档的系统数据，只允许制造商对各个组件进行维修。

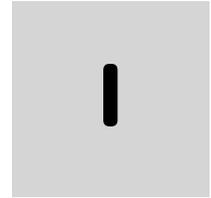
当控制器用于具有技术安全要求的应用时，请遵守有关的使用说明。

不遵守本产品的相关警告可能导致人身伤害或设备损坏。

用户意见

欢迎对本书提出意见。您可以给我们发邮件，我们的邮件地址是
techpub@schneider-electric.com

CANopen 通讯概述



概览

本部分的目标 本部分介绍在 CANopen 现场总线上进行的通讯。

本部分包含了哪些内容? 本部分包括以下内容:

章	章节标题	文件集
1	CANopen 通讯概述	13

CANopen 通讯概述



概览

本章目标

本章描述 CANopen 通讯的主要技术特性。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
原理	14
CAN 概览	15
CANopen 现场总线的一般架构	18
拓扑结构	20
CANopen 网络的长度限制	23
符合性分类	25

原理

简介

CAN 通讯总线最初是为车载系统开发的，现在被用于许多领域，包括：

- 运输
- 移动设备
- 医疗设备
- 建筑
- 工业控制

CAN 系统的长处在于：

- 总线分配系统
 - 错误检测
 - 数据交换的可靠性
-

主站 / 从站结构

CAN 总线具有主站 / 从站总线管理结构。

主站管理：

- 从站的初始化
 - 通讯错误
 - 从站的状态
-

波特率

波特率取决于总线长度（见 CANopen 网络的长度限制页 23）和拓扑结构。

点到点通讯

总线上的通讯按点到点的方式工作。

每个设备都随时可以向总线发送请求，相关的设备会进行响应。

总线中流经的请求的优先级由每个消息的标识符确定。

总线的设计原理

可修改 CANopen 总线，例如通过连接到其他设备或分支器盒来调整电缆长度。

在设计 CANopen 总线时，必须遵守下列规则：

- 确定总线最远端的节点间的距离
 - 确认每个段的长度以及连接到其上的节点数
 - 确认分支器的长度和密度
 - 确认所有段在每端都必须配有一个线路终结器
-

CAN 概览

概览

CANopen 是一种用于工业监控系统的标准现场总线协议。它尤其适合实时 PLC 使用，因为其高效、低成本的解决方案最适用于嵌入式工业应用程序。

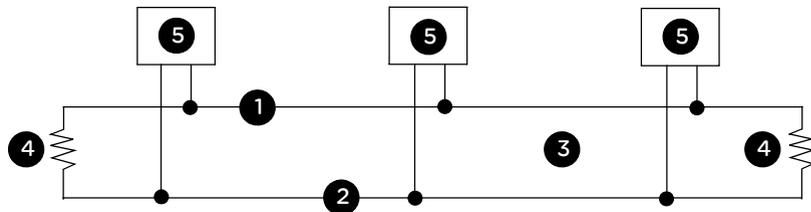
CANopen 协议

CANopen 协议是从 CAN 的一个子集设计而成的协议。从配置文件定义方面来看，它尤其适合在标准工业组件中使用。CANopen 是 CiA (CAN in Automation) 协会的一个标准，从其一面市就立刻备受瞩目。在欧洲，CANopen 被认为是基于 CAN 概念设计的工业系统标准参照。

物理层

CAN 使用以差分方式控制的两线总线（公共回路）。CAN 信号为 CAN-high 和 CAN-low 之间的电压差。（请参见下图。）

下图显示 CAN 两线总线的物理层组件：



说明

否	说明
1	CAN-High 线
2	CAN-Low 线
3	CAN-high/CAN-low 信号的电位差
4	120Ω 电阻块
5	元件

根据电磁兼容性要求，可以以并行、绞合或加强方式布置总线。单一线路结构可减少反射。

CANopen 配置文件

通讯配置文件

CANopen 配置文件系列基于“通讯配置文件”，通讯配置文件指定主要通讯机制及其描述 (DS301)。

设备配置文件

“设备配置文件”中介绍了工业机器人技术中使用的最重要的设备类型。并对其功能进行了定义。

介绍的标准设备的示例为：

- 数字量和模拟量输入 / 输出分配器 (DS401)
- 电机 (DS402)
- 命令设备 (DSP403)
- 回路控制器 (DSP404)
- PLC (DS405)
- 编码设备 (DS406)

通过 CAN 总线 配置设备

使用 CAN 总线配置设备的功能是制造商（通过配置文件系列）所需的独立性的基本组成部分。

CANopen 配置 文件的一般特征

CANopen 是 CAN 系统的一组配置文件，具有以下规格：

- 开放总线系统
- 无协议过载的实时数据交换
- 开放总线系统
- 可修改大小的模块化概念
- 设备的互连和可交换性
- 众多国际制造商的支持
- 标准化网络配置
- 访问所有设备参数
- 循环过程数据和 / 或事件驱动数据的同步和循环（系统响应时间短的可能性）

CANopen 产品认证

市场上所有提供经过认证的 CANopen 产品的制造商均为 CiA 协会的成员。作为此组织 (CiA) 中的活跃成员，Schneider Electric Industries SAS 遵从此协会的标准化建议开发自己的产品。

CAN 标准 CANOpen 规格是由 CiA 协会定义的，通过 www.can-cia.com 网站可访问其部分内容。许多供应商都可提供主站设备和从站设备的源代码。

注意：要查找有关 CANOpen 规格和标准机制的详细信息，请访问 CiA 的主页 (<http://www.can-cia.de>)。

CANopen 网络上的通讯 通讯配置文件基于 CAL 服务和协议。它允许用户使用两种类型的交换：SDO 和 PDO：打开电源时，设备进入初始化阶段，在初始化结束后，设备进入预操作状态。在此阶段，只允许通过 SDO 进行通讯。在收到启动命令后，设备进入操作状态。然后启动 PDO 交换，此时仍可通过 SDO 进行通讯。

PDO PDO 是与过程数据的通讯接口对象，它允许实时数据交换。CANOpen 设备的所有 PDO 描述此设备与其通讯伙伴之间在网络上的隐式交换。PDO 交换在设备进入“操作”模式时立即获得授权。

SDO SDO 允许通过显式请求访问设备数据。SDO 服务在设备处于“操作”或“预操作”状态时可用。

CANopen 现场总线的一般架构

概览

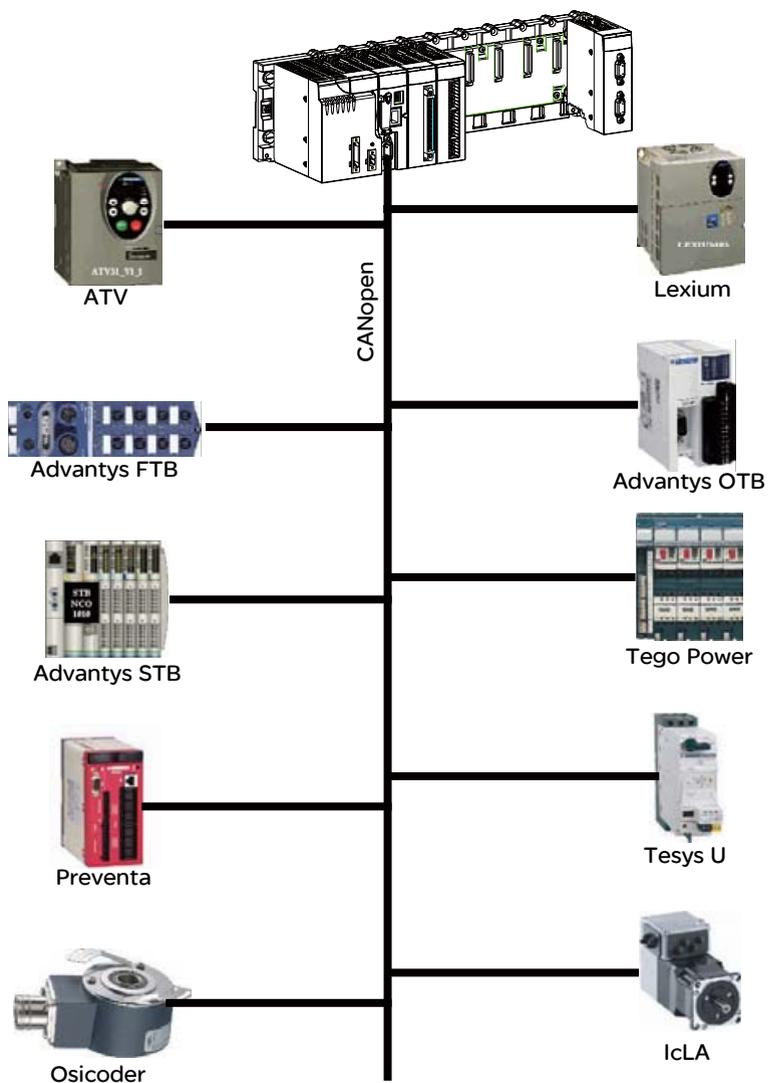
CANopen 架构包括:

- 总线主站
- 从站设备

注意: CANopen 主站的地址为节点编号 127。

架构示例

下图给出了一个 CANopen 架构示例：



拓扑结构

简介

CANopen 现场总线始终包含一个主站：BMX P34 2010/2030 处理器。

通过总线编辑器，您可以声明网络设备并将这些设备与唯一地址关联。

设备的类型有两种：

- **一体型元件：**由单一模块组成。
- **模块化元件：**由通讯器及一个或多个模块组成。

例如，模块化设备可以是 STB 岛（见使用外部工具的配置：配置软件页 89）或 OTB 设备。

CANopen 拓扑结构

可将设备连接到总线。

- **子站：**使用连接到单端口或多端口分流箱的节点。
- **链接：**使用单连接器或双连接器。

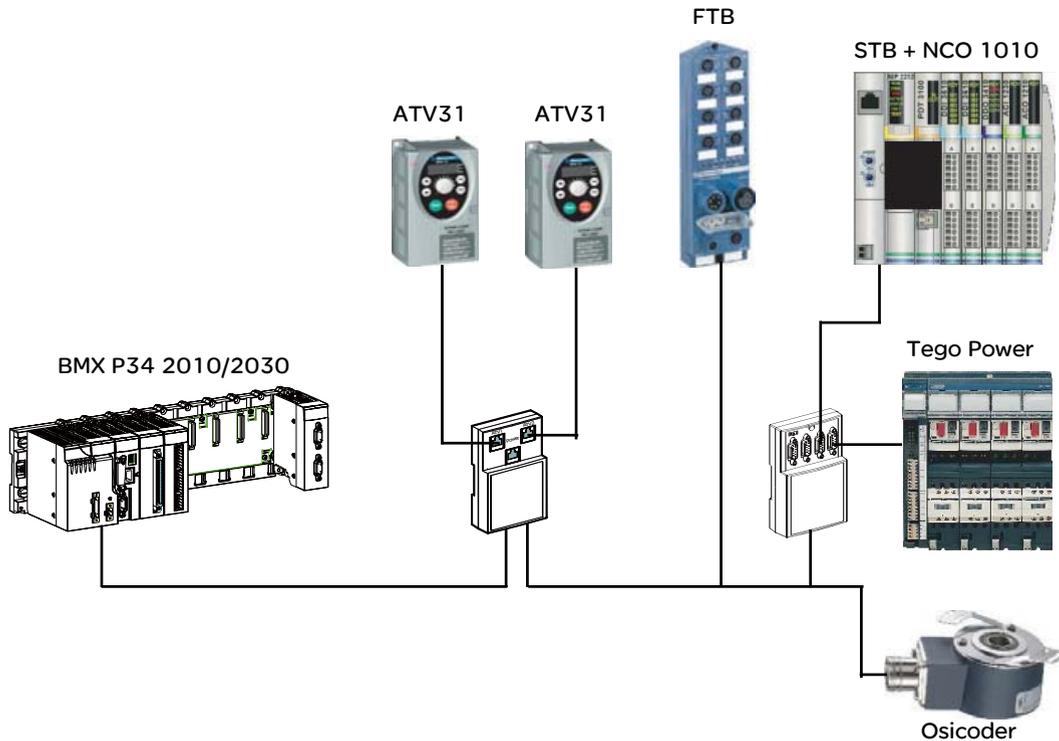
无论选择何种拓扑结构类型，都必须考虑长度限制（见 CANopen 网络的长度限制页 23）的问题。

这些限制涉及：

- 总线长度，即 2 个节点之间的最大距离
- 段长度
- 分支器长度

所有段在每端都必须配有一个线路端接器。

下图显示一个分流拓扑结构示例：



线路端接器

为了最大程度地减小线路末端的反射，必须在每个总线末端附近放置一个线路端接器。每个线路端接器都必须连接在线路 CAN_H 和 CAN_L 之间。这些端接器的电阻为 $120\ \Omega$ ， $1/4\ W\ 5\%$ 。

注意：某些情况下，线路端接器包含在 CANopen 设备中。

段中的设备数

理论上同一段上最多可连接 63 个设备。尽管如此，上述拓扑结构限制意味着实际上可连接的设备数要少于此数字。要增加总线上的设备数，同时又保持相同的流速，可以用“桥”来连接不同的段。

在所有情况下，一个 M340 CANopen 主站都不能处理超过 63 个从站设备。

CANopen 网络的长度限制

简介

CANopen 网络最多允许将 63 个设备和一个主站连接到总线。

总线长度、段和分支器均有限制，详细信息如下面的表所示。

总线长度

为总线选择的数据流速率决定了网络总体最大长度：

波特率	最大长度
1 Mbit/s	4 米
500 Kbit/s	100米
250 Kbit/s	250米
125 Kbit/s	500米
50 Kbit/s	1000米
20 Kbit/s	2500米

注意：总线长度的确定还必须考虑是否使用中继器，中继器会增加总线上信息的传播延迟。中继器在总线上增加传播延迟将减少总线的最大网络长度。5 纳秒的传播延迟相当于长度缩减 1 米。举例来说，带 150 纳秒延迟的中继器将减少 30 米的总线长度。

段长度

与数据流速率无关，不使用中继器时，连接数以及所用电缆类型将限制段的长度。

	电阻	Node_16	Node_32	Node_64
大截面电缆 AWG 18	33 欧姆 / 千米	575 米	530 米	460 米
AWG 电缆：22	70 欧姆 / 千米	270 米	250 米	215 米
小截面电缆 AWG 24	93 欧姆 / 千米	205 米	185 米	160 米
	88 欧姆 / 千米	215 米	200 米	170 米
AWG 电缆：26	157 欧姆 / 千米	120 米	110 米	95 米

子站长度

必须考虑到有关接头的长度限制并通过以下参数进行修正：

波特率	1 Mbit/s	500 Kbit/s	250 Kbit/s	125 Kbit/s	50 Kbit/s	20 Kbit/s
L_{max} (1)	0.3 米	5 米	5 米	5 米	60 米	150 米
Σ L_{max} 本地星形 (2)	0.6 米	10 米	10 米	10 米	120 米	300 米
最短间隔 0,6x Σ L 本地 (3)	-	6 米	6 米	6 米	72 米	180 米
Σ L_{max} (在所有总线上)	1.5 米	30 米	60 米	120 米	300 米	750 米

(1) L_{max}：一个接头的最大长度。

(2) Σ L_{max} 本地星形：使用多端口分支器创建本地星形连接时，同一点上各个接头的最大累计长度。

(3) 最短间隔：两个分支器之间的最短距离。同一点上分接盒的最大长度值。对于各分接盒，可以分别计算：两个分接盒之间的最短间隔是同一点上分接盒的累计长度的 60%。

(4) Σ L_{max}（在所有总线上）：所有总线上各个接头的最大累计长度。

有关详细信息，请参考文档 "CANopen 硬件实施手册"。

符合性分类

概览

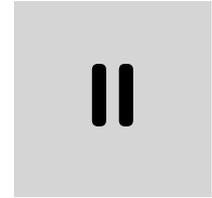
CANopen 通讯端口符合 Schneider M20 类。

		类 M20
分层配置	从站标识	1-63
	二进制流 (Kb/s)	50、125、250、500、1000
支持的设备		63
NMT	NMT 主站	NMT 主站符合 DS301 标准
	启动过程	符合 DSP 302
SDO	SDO 客户端	1
	SDO 服务器	1
	SDO 数据传输	已发送, 分段传输
PDO	COB-ID	读 \ 写
	PDO TT	0, 1-240, 254, 255
	PDO 抑制时间	TPDO (读 \ 写)
	PDO 事件定时器	TPDO (读 \ 写)
SYNC	SYNC	生产
EMCY		消费者 / 生产者
HEALTH	心跳	63 个消费者 1 个生产者
	节点防护	是
参数	保存参数	是

注意：支持的 PDO 数如下所示：

- 接收 256 (RxPDO)
- 256 传输 (TxPDO)

CANopen 硬件实施



概览

本部分主题

本部分描述 CANopen 总线架构的各种可能的硬件配置。

本部分包含了哪些内容?

本部分包括以下内容:

章	章节标题	文件集
2	BMX P34 处理器的硬件实现	29
3	CANopen 设备简介	35

BMX P34 处理器的硬件实现

2

概览

本章目标

本章介绍配备了 CANopen 端口的 BMX P34 处理器及其实现。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
处理器描述：BMX P34 2010/2030	30
安装	31
CANopen 处理器的可视化诊断	32

处理器描述：BMX P34 2010/2030

概览

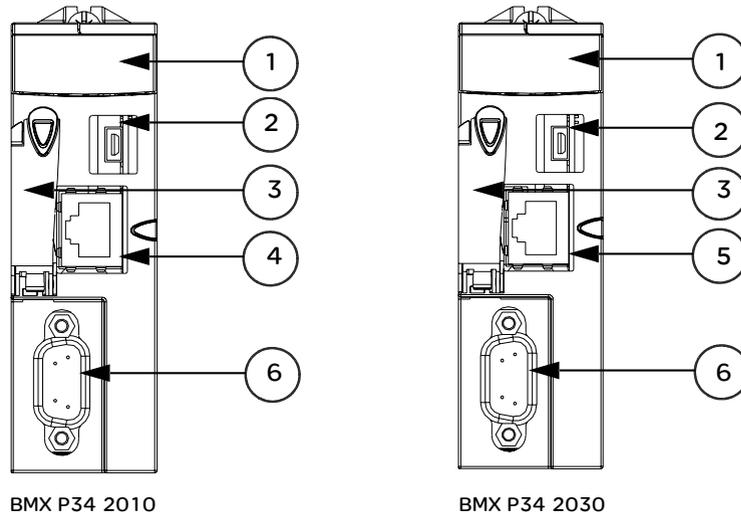
每个 PLC 工作站都配备了一个 BMX P34 处理器。

Modicon M340 系列中有两种处理器具有 CANopen 端口：

- BMX P34 2010, 它还具有 USB 端口和串行口
- BMX P34 2030, 它还具有 USB 端口和以太网端口

BMX P34 处理器设计简单, 并包含一个存储卡插槽。

下图显示了 BMX P34 2010/2030 的正面视图：



BMX P34 2010

BMX P34 2030

编号	名称
1	显示面板
2	USB 端口
3	SD 卡插槽
4	串行口
5	以太网端口
6	CANopen 端口

这些处理器都是总线主站；不能作为从站使用。它们通过 SUB-D 9 针连接器的连接点链接，并允许连接支持 CANopen 协议的从站设备。

注意：总线上只有一个 BMX P34 主站。

安装

概览

配备了 CANopen 端口的 BMX P34 2010/2030 处理器安装在由 BMX CPS 模块馈送的 BMX XBP 机架上。

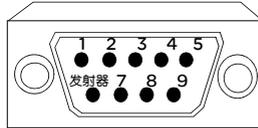
注意: 当处理器正在运行时拔出 / 插入处理器, 总线将不再正常操作。为了重新启动总线, 必须重新初始化电源。

CANopen 连接器

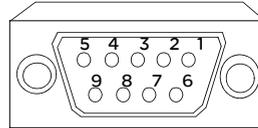
CANopen 处理器端口配有 SUB-D9 连接。

下图显示用于模块 (凸型) 和电缆 (凹型) 的 CANopen 连接器。

凸型连接器



凹型连接器



引脚	信号	描述
1	-	保留
2	CAN_L	CAN_L 总线 (低)
3	CAN_GND	CAN 线束
4	-	保留
5	保留	CAN 可选保护
6	GND	可选线束
7	CAN_H	CAN_H 总线 (高)
8	-	保留
9	保留	CAN 外部电源。 (专供光耦合器电源和发射器 - 接收器使用。) 可选

注意: CAN_SHLD 和 CAN_V+ 未安装在 Modicon M340 系列处理器上。这些是保留的连接。

CANopen 处理器的可视化诊断

概览

Modicon M340 系列中的 BMX P34 处理器配备了若干模块状态可视化 LED。

配备 CANopen 端口的 BMX P34 2010/2030 处理器在其正面具有 2 个 LED, 用于指示总线状态:

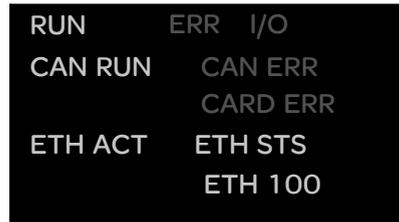
- 绿色 CAN RUN LED
- 红色 CAN ERR LED

在正常运行状态下, CAN ERR LED 熄灭, 而 CAN RUN LED 亮起。

下图显示模块正面的 LED:



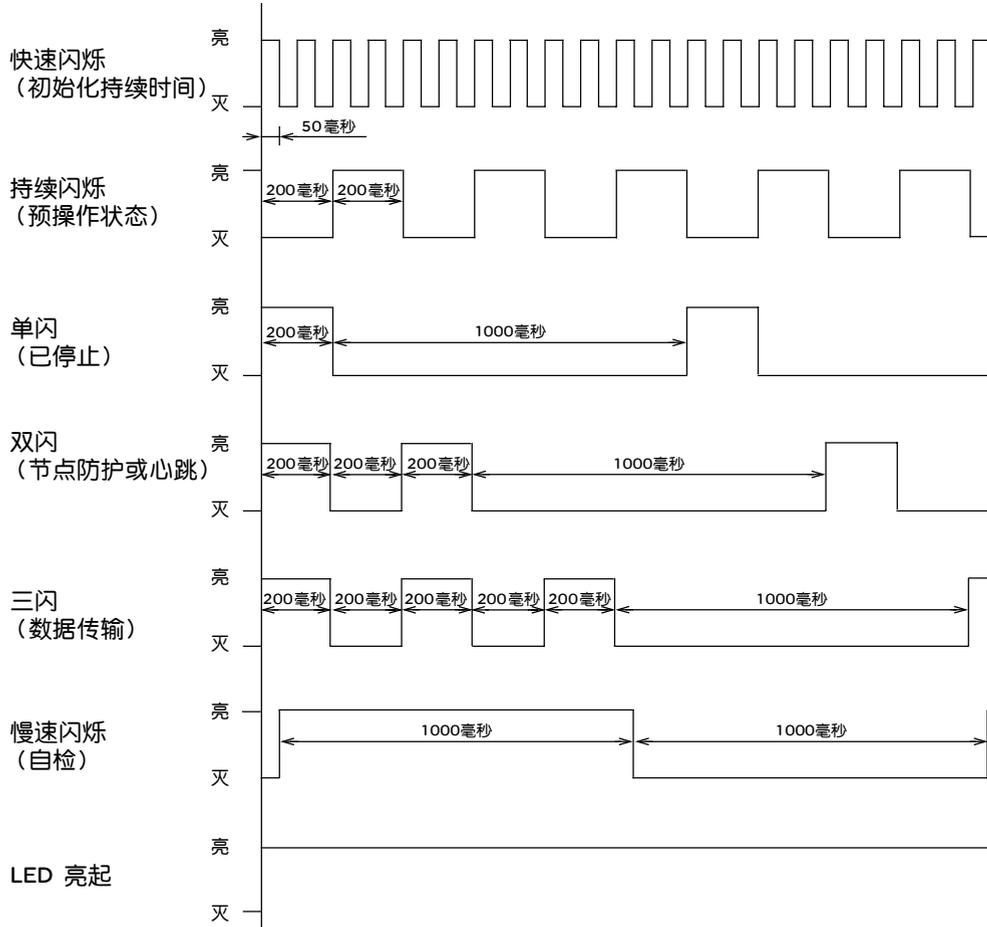
BMX P34 2010 的可视化屏幕



BMX P34 2030 的可视化屏幕

LED 状态

下面的趋势图显示各个 LED 的各种可能状态：



描述

下表描述 CAN RUN 和 CAN ERR LED 的作用：

显示 LED	亮 	快速闪烁 	闪烁 	持续闪烁 	灭 	慢速闪烁 
CAN RUN (绿色)	主站正常工作。	正在进行初始化。	单闪 ：主站停止运行。 三闪 ：正在加载 CANopen 固件。	主站处于预操作状态。	-	正在启动 CANopen 主站自检。
CAN ERR (红色)	总线已停止。CAN 控制器状态为 "BUS OFF" 。	正在进行初始化。	单闪 ：至少一个错误计数器已接近、达到或超过警报级别。 双闪 ：监控故障（节点防护或心跳）	配置无效。	无错误。	CANopen 组件无法启动。

CANopen 设备简介

3

概览

本节主题

本节介绍不同的 CANopen 设备。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
CANopen 设备	36
CANopen 运动命令设备	37
CANopen 输入 / 输出设备	43
其他设备	47

CANopen 设备

概览

能够连接到 CANopen 总线的设备和能够在 Unity Pro V4.0 中配置的设备根据其功能分为以下几组：

- 运动命令设备
- 输入 / 输出设备
- 其他设备

注意：只有 Hardware Catalog Manager 中的设备才可与 Unity Pro 一起使用。

运动命令设备

使用运动命令设备可控制电机。

这些设备包括：

- Altivar
 - Lexium
 - ICLA
 - Osicoder
 - Telsys T
 - SD328A 步进器驱动器
-

输入 / 输出设备

输入 / 输出模块作为远程模块工作。这些设备包括：

- Tego Power 设备
 - Advantys FTB
 - Advantys OTB
 - Advantys FTM
 - Preventa 设备
-

其他设备

它们是：

- Advantys 岛 STB
- Tesys U
- Festo 阀门终端
- Parker Moduflex

使用 STB 岛还可监控输入 / 输出。

CANopen 运动命令设备

概览

使用运动命令设备可控制电机。

这些设备包括：

- Altivar
 - Lexium
 - IcLA
 - Osicoder
 - Tesys T
 - SD328A 步进器驱动器
-

ATV 设备

ATV 设备能够通过流量矢量控制来控制电机的速度。

下图给出了 ATV 设备的一个示例：



注意：对于 ATV31、ATV61 和 ATV71，建议的最低固件版本为 V1.1。

注意：不支持 ATV31 V1.7。但是，使用 ATV31 1.2 配置文件进行配置时可以使用它。在这种情况下，只能使用 ATV31 V1.2 功能。

注意：ATV71：如果您必须将它从 CANopen 总线断开连接，请关闭设备电源，否则，当将其与总线重新连接时，将导致总线严重错误。此问题已在 ATV71 固件版本 V1.2 及以上版本中得到解决。

注意：ATV61：如果您必须将它从 CANopen 总线断开连接，请关闭设备电源，否则，当将其与总线重新连接时，将导致总线严重错误。此问题已在 ATV61 固件版本 V1.4 及以上版本中得到解决。

Lexium 设备

借助于与 BSH 伺服电机兼容的 Lexium 05 伺服驱动器系列, 可以构成具有各种功率 (0.4...6 kW) 和电源电压范围的各种机器的一体型和动态组合。

Lexium 05 伺服驱动器的一体型设计和集成组件 (线路过滤器、制动电阻器和安全功能) 可以最大程度地减少开关机柜中所需的空间。它集成了切断电源安全功能, 可以防止电机意外启动。

伺服驱动器 Lexium 05 的另一个优势是提供了丰富的应用程序选项:

- 通过模拟量输入充当扭矩或速度控制器
- 通过 RS422 接口充当电子变速箱
- 通过现场总线接口充当定位或速度控制器

伺服驱动器可用于以下四种电压类型:

- 115 VAC 单相
- 230 VAC 单相和三相
- 400/480 VAC 三相

下图给出了 Lexium 设备的一个示例:



注意: 对于 Lexium05 设备, 建议的最低固件版本为 V1.120

注意: 对于 Lexium 15 LP, 建议的最低固件版本为 V1.45

注意: 对于 Lexium 15 MH, 建议的最低固件版本为 V6.64

IcLA 设备

IcLA 设备为智能一体型设备。这些设备集成了运动任务所需的一切：定位控制器、电源设备和伺服、EC 或步进电机。

下图给出了 IcLA 设备的一个示例：



⚠ 警告

不支持的 ICIA IFA 版本。

从最低固件版本 V1.105 开始就保证了可靠的操作。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

⚠ 警告

不支持的 ICIA IFE 版本。

从最低固件版本 V1.104 开始就保证了可靠的操作。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

⚠ 警告

不支持的 ICIA IFS 版本。

从最低固件版本 V1.107 开始就保证了可靠的操作。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

Osicoder 设备

Osicoder 设备是一个角度位置传感器。

它在机械结构上与机器的驱动心轴紧密结合，编码器的轴旋转一个圆盘，该圆盘由一系列不透明区域和透明区域组成。LED 发出的光线穿过圆盘的透明区域，并由光敏二极管进行检测。接着，光敏二极管生成一个电子信号，该信号经过放大并转换为数字信号，然后传输到处理系统或电子变速驱动器。因此，编码器的电子输出以数字形式表示输入轴的角度位置。

下图给出了 Osicoder 设备的一个示例：



注意：对于 Osicoder 设备，建议的最低固件版本为 V1.0。

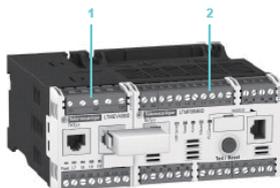
Tesys T 电机管理系统

Tesys T 是一种电机管理系统，可向最大 810 A 的单相和三相恒定速度交流电机提供保护、测量和监控功能。

当将其用于电机控制面板中时，可以：

- 提高安装的操作可用性，
- 改善从项目设计直至实施的整个过程的灵活性，
- 通过提供运行系统所需的全部信息，提高生产率。

下图给出了 Tesys T 设备的一个示例：



1 LTM EV40BD extension module
2 LTM R08MBD controller

SD328A 步进器 驱动器

SD328A 是一种通用步进器驱动器。

它提供了非常紧凑但功能强大的驱动器系统，并且选择了由 Berger Lahr 提供的步进器驱动器。

此设备提供了一个输出，用于直接连接可选的恒速制动。

下图给出了 SD328A 步进器驱动器设备的一个示例：



CANopen 输入 / 输出设备

概览

输入 / 输出模块作为远程模块工作。

这些设备包括：

- Tego Power 设备
- Advantys FTB
- Advantys OTB
- Advantys FTM
- Preventa 设备

Tego Power 设备

Tego Power 是一个模块化系统，它利用预接线的控制和电源电路标准化和简化了电机起动器的实现。此外，此系统还允许在以后对电机起动器进行自定义，通过减少端子数和中间接口以及接线管道数量，缩短维修时间并优化面板的空间。

下图给出了 Tego Power 设备的一个示例：



注意： TegoPower APP_1CC00 和 TegoPower APP_1CC02 的最低版本为 V1.0。

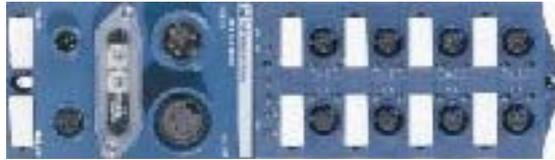
**Advantys FTB
设备**

Advantys FTB 分配器由几个允许连接采集器和激活器的输入 / 输出组成。

注意：FTB 的最低固件版本为 V1.7

注意：对于 FTB 1CN16CM0，从最低固件版本 V1.5 开始就保证了可靠的操作。

下图给出了 Advantys FTB 设备的一个示例：



Advantys OTB 设备

Advantys OTB 设备允许您组成离散量输入 / 输出岛（边界中最多有 132 个通道）或模拟量（最多 48 个通道）IP20，并将它们连接到靠近活动采集器的位置。

下图给出了 Advantys OTB 设备的一个示例：



注意： OTB 的最低固件版本为 V2.0。

警告

不支持的 OTB 版本

从最低固件版本 V2.0 开始就保证了可靠的操作。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

Advantys FTM CANopen

借助于 Advantys FTM 模块化系统，可以使用单个通讯接口（现场总线模块）连接不同数量的输入 / 输出分离器盒。

这些分离器盒通过混合线缆连接到模块，其中包括内部总线和电源（内部、传感器和执行器）。

输入 / 输出分离器盒独立于现场总线类型，这样就减少了分离器盒参考号的数量。一旦安装，本系统就可以运行。

下图给出了 Advantys FTM CANopen 设备的一个示例：



Preventa 设备

Preventa 设备是用于监控安全功能的电子安全控制器。

下图给出了 Preventa 设备的一个示例：



其他设备

概览

这些设备包括：

- STB 岛
- Tesys U
- Festo 阀门终端
- Parker Moduflex

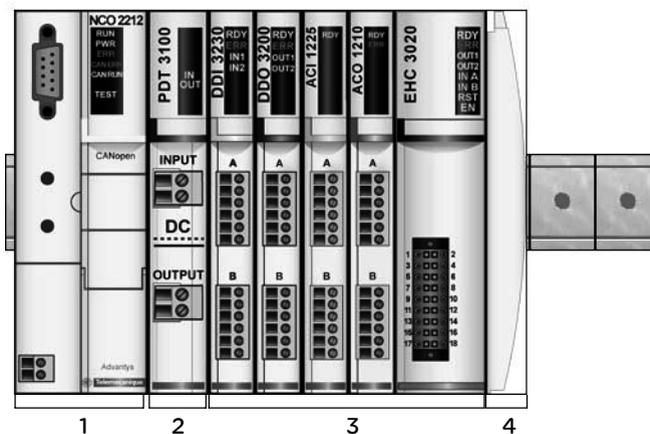
STB 岛

Advantys STB 岛由几个输入 / 输出模块组成。

岛的模块化元件通过 CANopen 本地总线使用网络接口模块 NIM 连接。

STB 模块只能在 STB 岛中使用。

下图给出了岛的一个示例：



描述：

编号	说明
1	网络接口模块。
2	电源分配模块。
3	分配的输入 / 输出模块。这些模块可以是： <ul style="list-style-type: none"> ● 数字量输入 / 输出模块， ● 模拟量输入 / 输出模块， ● 特殊用途模块。
4	岛总线的端接板。

Tesys U 设备

TeSys U-Line 电机启动器为各种设备提供电机控制，从具有固态热过载保护功能的基本电机启动器，到在网上通讯并具有可编程电机保护功能的复杂电机控制器，不一而足。

此设备执行以下功能：

- 单相或三相电机的保护和控制：
 - 隔离断开功能
 - 电子短路保护
 - 电子过载保护
 - 电源切换
- 应用程序控制：
 - 报警（警告保护功能报警，例如过载挂起）
 - 状态监控（运行、就绪、故障，等等）
 - 应用监控（运行时间、故障数目和电机电流值）
 - 故障记录（保存最后 5 个故障，以及电机参数值）

下图给出了 Tesys U 设备的一个示例：



Festo 阀门终端

CPV 直接:

CPV 阀是串联汇流阀，除了阀门功能之外，它们还包括所有用于进气线、排气线和工作线的气压管道。

进气管道是由阀片组成的中心组件，允许气流直接通过阀片。这有助于实现最大流速。所有阀门都具有气压导航控制以优化性能。

现场总线节点直接集成在阀门终端的电气接口中，因此只占用极少的空间。

通过可选的串扩展，可以将另一个阀门终端以及 I/O 模块连接到 Fieldbus Direct 现场总线节点。

CPV 阀门终端具有三种尺寸：

- CPV10
- CPV14
- CPV18

下图给出了 Festo 阀门终端设备的一个示例：



CPX 终端:

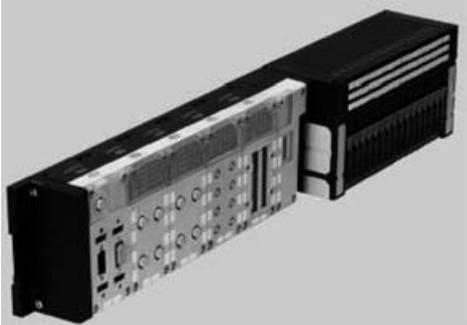
电气终端 CPX 是一种适用于阀门终端的模块化外设系统。此系统经过专门设计，以使阀门终端可适合不同的应用领域。

适用于阀门终端气压组件的不同连接选件 (MPA/CPA/VTSA)

适用于传感器和执行器的灵活的电气连接技术

还可以在不将阀门用作远程 I/O 系统的情况下使用 CPX 终端。

下图给出了 CPX 终端设备的一个示例：



Parker Moduflex

Parker Moduflex 阀门系统提供了灵活的气压自动化功能。

根据应用领域，您可以组装不同长度的岛（最大 16 路输出）。IP 65-67 防水和防尘功能使得您能够将阀门安装在气缸附近，以缩短响应时间和降低耗气量。Parker Moduflex 阀门系统 CANopen 模块 (P2M2HBVC11600) 可在 Modicon M340 配置中用作一个增强的 CANopen 设备。

P2M2HBVC11600 的固件版本必须为 1.4 版或更高版本。

有关 P2M2HBVC11600 接线、LED 模式、安装过程和功能的详细描述，请参阅由 Parker 提供的用户文档。

"S" 系列独立阀门：

对于机器上隔离的气缸，最好找到附近的阀门。因此，独立模块是理想之选，可将响应时间和耗气量降至最小。可以将外设模块直接安装到阀门中。

下图给出了 "S" 系列单螺线管设备的一个示例：



下图给出了 "S" 系列单动空气导向设备的一个示例：



"T" 系列阀门岛模块

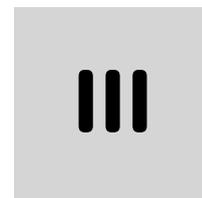
适用于需要短的定位阀门岛的气缸组。

可以在同一个岛式模型中组合具有不同功能和流管的模块，以便提供极大的灵活性以满足各种机器要求。

下图给出了 "T" 系列阀门岛模块设备的一个示例：



CANopen 通讯的软件实现



概览

本部分主题

本部分描述 CANopen 应用程序中的软件配置、编程和诊断的不同可能情况。

本部分包含了哪些内容?

本部分包括以下内容:

章	章节标题	文件集
4	一般信息	55
5	CANopen 总线上的通讯配置	65
6	Catalog Manager 软件实现	103
7	编程	141
8	在 CANopen 总线上调试通讯	159
9	诊断	165
10	语言对象	171

一般信息

4

概览

本章主题

本章描述 Modicon M340 总线上的 CANopen 软件实施原理。

本章包含了哪些内容?

本章包含了以下主题:

主题	文件集
实施原理	56
实现方法	57
性能	58
设备性能	61

实施原理

概览

为实施 CANopen 总线，必须定义在其中集成总线的应用的物理环境（机架、电源、处理器、模块），然后确保实施必要的软件。

使用 Unity Pro 可通过两种方式实施软件：

- 离线模式
- 在线模式

实施原理

下表显示了不同的实施阶段：

模式	阶段	说明
离线	配置	输入配置参数。
离线或在线	用符号表示	用符号表示与 BMX P34 处理器的 CANopen 端口关联的变量。
	编程	给特定功能编程： <ul style="list-style-type: none"> ● 位对象或关联的字 ● 特定指令
在线	传输	将应用程序传输到 PLC 。
	调试 诊断	可以使用不同的资源调试应用程序、控制输入 / 输出和诊断故障： <ul style="list-style-type: none"> ● 语言对象或 IODDT ● Unity Pro 调试屏幕 ● 通过 LED 指示
离线或在线	文档	打印与 CANopen 端口配置有关的各种信息。

注意：上述顺序仅供参考。Unity Pro 软件允许您以需要的交互方式顺序使用编辑器。

⚠ 危险

应用程序的意外行为

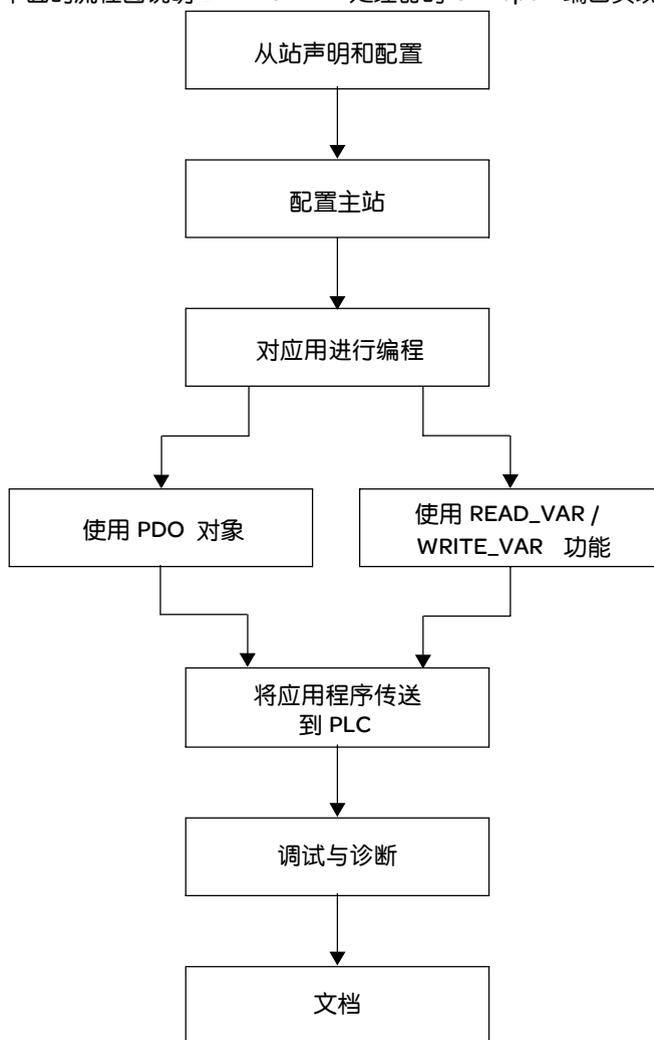
使用诊断系统信息并监控通讯的响应时间。在通讯受到干扰的情况下，响应时间可能会特别长。

如果不遵守这个警告会导致死亡，严重伤害。

实现方法

概述

下面的流程图说明 BMX P34 处理器的 CANopen 端口实现方法:



性能

简介

CANopen 的各种性能详细描述如下。

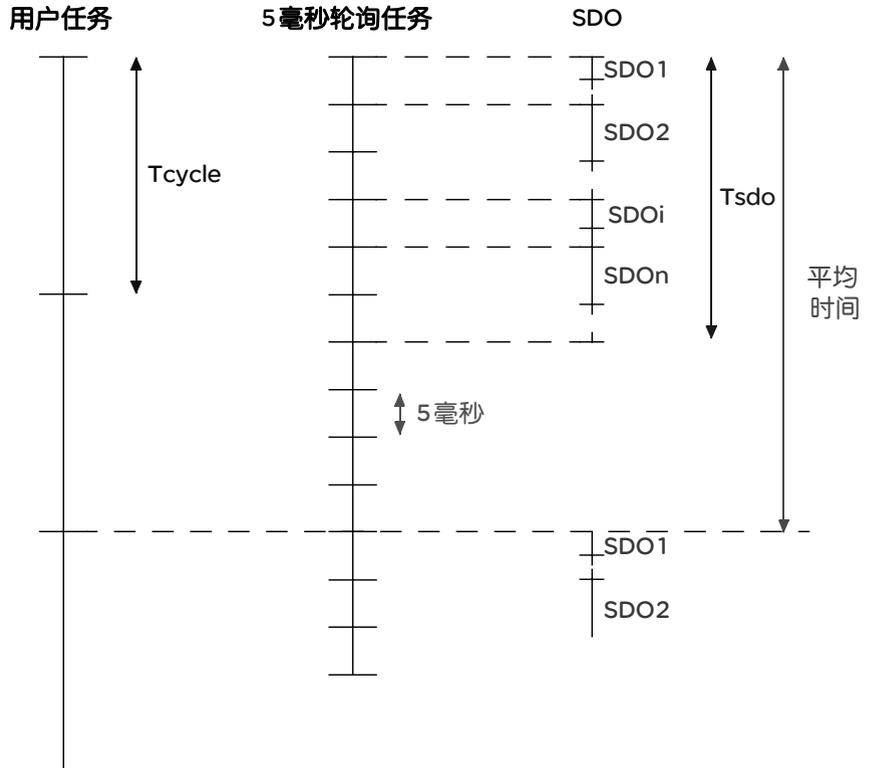
对任务循环时间的 影响

为每个任务循环指定的时间如下所示：

任务	最短
CANopen 输入	10 微秒 / PDO
CANopen 输出	80 微秒 + 15 微秒 / PDO
诊断	120 微秒

通过 SDO 通讯

下图概述了 SDO 管理：



下表定义了描述“通过 SDO 通讯”图形时使用的术语：

术语	定义
Tcycle	用户任务循环
n	要并行执行的 SDO 数量
Tsdo	处理 n 个 SDO 的时间（由于是轮询任务，所以是 5 毫秒的倍数）
平均时间	执行所有 SDO（从 SDO1 到 SDO _n ）的平均时间。 平均时间取决于 Tcycle、n 和 Tsdo: <ul style="list-style-type: none"> ● 如果 Tcycle > Tsdo，则平均时间 = Tcycle ● 如果 Tcycle < Tsdo，则平均时间 = NB * Tcycle 并且 NB = Tsdo / (Tcycle + 1)

注意：轮询任务在每个任务循环中每 5 毫秒运行一次，以便检查交换的终点。这对运行许多 SDO 的用户是十分有用的。

示例：如果任务循环为 50 毫秒，每个主循环 10 个 SDO，并且 SDO 交换时间为 3 毫秒。

采用轮询任务，我们可以每 5 毫秒处理 1 个 SDO。

因此，我们可以启动 10 个 SDO/ 任务循环。

总线启动

CANopen 总线启动时间取决于设备数量。

启动 CANopen 总线的最短时间为 7 秒。

配置一台设备所需的时间大约为 0.8 秒。

配有 64 台设备的 CANopen 总线的启动时间大约为 1 分钟。

**断开 / 重新连接
设备**

断开连接:

检测到设备断开连接的时间取决于错误控制:

错误控制	说明
防护时间	检测到断开连接的时间为 防护时间 * 使用寿命因数
心跳	检测到断开连接的时间为 心跳生成器时间 + (心跳生成器时间 / 2)

重新连接:

设备上的主站每秒轮询设备一次，以检查设备是否重新连接。如果设备并非单独位于总线上，则重新连接此设备所需的时间大约为 1 秒。

如果设备单独位于总线上，则设备断开连接将采用与总线完全断开连接相同的方式设置主站。在此状态后，主站重新启动总线，设备重新连接所需的时间大约为 7 秒。

设备性能

概览

下表描述了每个设备允许的 PDO 最大数量：

系列	设备	F*	Tx PDO	Rx PDO	Tx Cob Id	Rx Cob Id	Extra Cob Id	%MW IN	%MW OUT	%M IN	%MW OUT
电机控制	APP_1CC00		5	5	4	4	2	4	2	0	0
	APP_1CC02		5	5	4	4	2	8	6	0	0
	TeSysT_MMC_L		4	4	4	4	0	46	8	0	0
	TeSysT_MMC_L_ EV40		4	4	4	4	0	62	12	0	0
	TeSysT_MMC_R		4	4	4	4	0	46	8	0	0
	TeSysT_MMC_R_ EV40		4	4	4	4	0	62	12	0	0
	TeSysU_C_Ad		4	4	4	4	0	16	8	0	0
	TeSysU_C_Mu_L		4	4	4	4	0	50	10	0	0
	TeSysU_C_Mu_R		4	4	4	4	0	38	12	0	0
	TeSysU_Sc_Ad		4	4	4	4	0	14	10	0	0
	TeSysU_Sc_Mu_L		4	4	4	4	0	48	10	0	0
	TeSysU_Sc_Mu_R		4	4	4	4	0	36	12	0	0
	TeSysU_Sc_St		4	4	4	4	0	14	10	0	0
检测	Oscoder		2	0	2	0	0	2	0	0	0
分布式 I/O	FTB_1CN08E08CMO		2	2	2	2	0	2	0	40	8
	FTB_1CN08E08SPO		2	2	2	2	0	2	0	0	8

系列	设备	F*	Tx PDO	Rx PDO	Tx Cob Id	Rx Cob Id	Extra Cob Id	%MW IN	%MW OUT	%M IN	%MW OUT	
	FTB_1CN12E04SPO		2	2	2	2	0	2	0	28	4	
	FTB_1CN16CM0		2	2	2	2	0	2	0	56	16	
	FTB_1CN16CP0		2	2	2	2	0	2	0	56	16	
	FTB_1CN16EM0		2	2	2	2	0	2	0	24	0	
	FTB_1CN16EP0		2	2	2	2	0	2	0	24	0	
	FTM_1CN10		5	5	4	4	2	54	50	0	0	
	OTB 岛		2	8	8	4	4	8	68	20	0	0
			1	6	8	4	4	6	102	54	0	0
	OTB_1CO_DM9LP		8	8	4	4	8	38	10	0	0	
	STB_NCO_1010		3	32	32	4	4	56	132	96	0	0
			1	32	32	4	4	56	228	192	0	0
	STB_NCO_2212		3	32	5	4	4	56	132	96	0	0
			1	32	32	4	4	56	228	192	0	0
			9	32	32	4	4	56	278	244	0	0
4			32	32	4	4	56	694	484	0	0	
运动和 驱动器	ATV31_V1_1	5	2	2	2	2	0	4	4	0	0	
		2	2	2	2	2	0	6	10	0	0	
		1	2	2	2	2	0	20	16	0	0	
	ATV31_V1_2	5	2	2	2	2	0	4	4	0	0	
		2	2	2	2	2	0	6	10	0	0	
		1	2	2	2	2	0	20	16	0	0	
		6	2	2	2	2	0	2	2	0	0	
	ATV31_V1_7	5	2	2	2	2	0	4	4	0	0	
		2	2	2	2	2	0	6	10	0	0	
1		2	2	2	2	0	20	16	0	0		

系列	设备	F*	Tx PDO	Rx PDO	Tx Cob Id	Rx Cob Id	Extra Cob Id	%MW IN	%MW OUT	%M IN	%MW OUT
	ATV31T_V1_3	5	2	2	2	2	0	4	4	0	0
		2	2	2	2	2	0	6	10	0	0
		1	2	2	2	2	0	20	16	0	0
	ATV61_V1_1	5	3	3	3	3	0	8	8	0	0
		2	3	3	3	3	0	32	20	0	0
		1	3	3	3	3	0	70	62	0	0
		7	3	3	3	3	0	76	62	0	0
	ATV71_V1_1	5	3	3	3	3	0	8	8	0	0
		2	3	3	3	3	0	16	10	0	0
		1	3	3	3	3	0	22	14	0	0
		7	3	3	3	3	0	80	58	0	0
		6	3	3	3	3	0	6	6	0	0
	IclA_IFA	8	1	1	1	1	0	8	10	0	0
		6	1	1	1	1	0	6	6	0	0
	IclA_IFE	8	1	1	1	1	0	8	10	0	0
		6	1	1	1	1	0	6	6	0	0
	IclA_IFS	8	1	1	1	1	0	8	10	0	0
		6	1	1	1	1	0	6	6	0	0
	LXM05_MFB		4	4	4	4	0	10	10	0	0
	LXM05_V1_12		4	4	4	4	0	24	26	0	0
LXM15LP_V1_45		4	4	4	4	0	8	10	0	0	
LXM15MH_V6_64	8	4	4	4	4	0	96	134	0	0	
	6	4	4	4	4	0	8	10	0	0	
SD3_28		4	4	4	4	0	22	20	0	0	
安全	XPSMC16ZC		4	0	4	0	0	28	0	0	0
	XPSMC32ZC		4	0	4	0	0	28	0	0	0
第三方 设备	CPV_C02	5	1	1	1	1	0	8	4	0	0
		9	1	1	1	1	0	10	6	0	0
		10	1	1	1	1	0	10	4	0	0
	CPX_FB14	11	4	4	4	4	0	56	50	0	0
		12	4	4	4	4	0	26	20	0	0
		9	4	4	4	4	0	72	66	0	0
P2M2HBVC11600		1	1	1	1	0	2	2	0	0	

F* 的图例	
1	扩展
2	标准
3	简化
4	大型
5	基础
6	MFB
7	控制器
8	缺省值
9	高级
10	CP 扩展
11	仅基础 DIO
12	一般 DIO AIO

F*: 函数

CANopen 总线上的通讯配置

5

概览

本章目标

本章介绍 CANopen 现场总线的配置以及总线主站和从站的配置。

本章包含了哪些内容？

本章包含了以下部分：

段	段	文件集
5.1	一般信息	66
5.2	总线配置	68
5.3	设备配置	78
5.4	主站配置	94

5.1 一般信息

概览

本节主题 本节介绍 CANopen 通讯的配置。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
一般信息	67

一般信息

简介

CANopen 架构的配置与 Unity Pro 完全集成。

当对 CANopen 主站的通道进行了配置后，系统会自动在项目浏览器中创建一个节点。然后，可以使用此节点启动总线编辑器，以定义总线的拓扑结构和配置 CANopen 元件。

注意：不能在已连接模式下修改 CANopen 总线的配置。

5.2 总线配置

概览

本节主题 本节介绍 CANopen 总线的配置。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
如何访问 CANopen 总线配置屏幕	69
CANopen 总线编辑器	71
如何在总线中添加设备	73
如何删除 / 移动 / 复制总线设备	75
在项目浏览器中查看 CANopen 总线	77

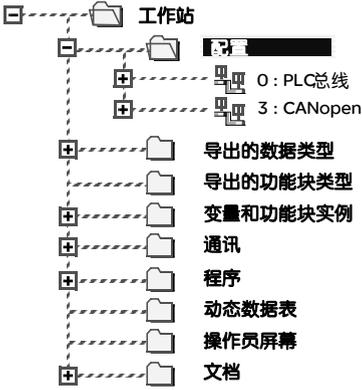
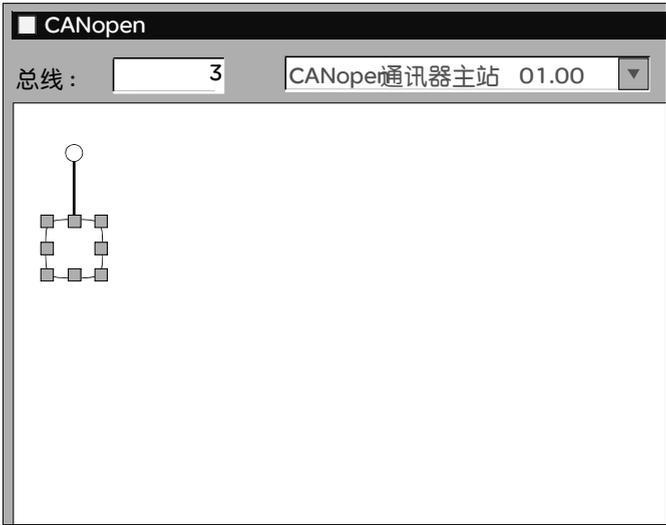
如何访问 CANopen 总线配置屏幕

概览

本部分描述如何访问带内置 CANopen 链路的 Modicon M340 PLC 的 CANopen 总线配置屏幕。

过程

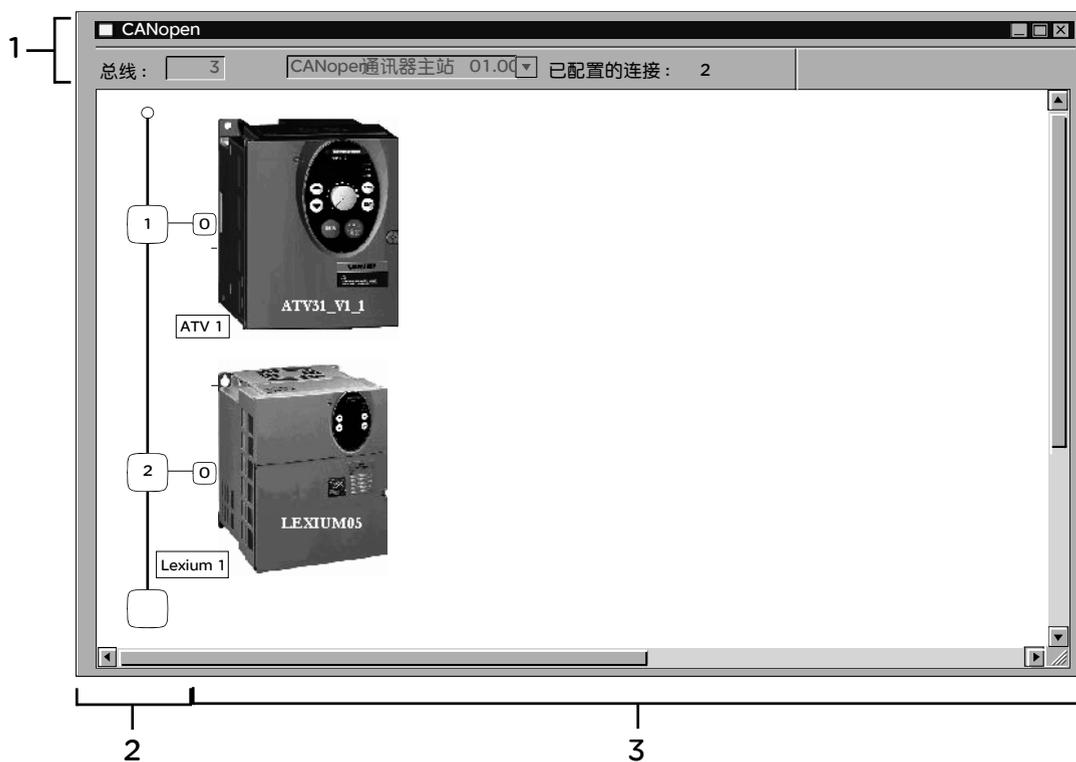
要访问 CANopen 现场总线，请执行以下操作：

步骤	操作
1	<p>在项目导航器中，展开配置目录。</p> <p>结果：将出现以下屏幕：</p> 
2	<p>要打开 CANopen 总线屏幕，请选择下列方法之一：</p> <ul style="list-style-type: none"> ● 双击 CANopen 目录。 ● 选择 CANopen 子目录并在上下文菜单中选择打开。 <p>结果：显示 CANopen 窗口：</p> 

CANopen 总线编辑器

概览 此屏幕用于声明连接到总线的设备。

示意图 CANopen 总线编辑器如下所示：



元素和功能

下表对构成配置屏幕的各个区域进行了描述：

编号	元素	功能
1	总线	总线编号。
	已配置的连接	指示已配置的连接点数。
2	逻辑地址区域	此区域包含连接到总线的设备的地址。
3	模块区域	此区域包含连接到总线的设备。

空白方块指示可用的连接点。

如何在总线上添加设备

过程 使用此操作，可通过软件添加连接到 CANopen 总线的设备：

步骤	操作																																		
1	访问 CANopen (见 如何访问 CANopen 总线配置屏幕页 69) 配置屏幕。																																		
2	<p>双击应连接模块的位置。 结果： 将显示新设备屏幕。</p>  <table border="1" data-bbox="299 597 1008 1015"> <thead> <tr> <th>参考号</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>[-] CANopen 子站</td> <td></td> </tr> <tr> <td>[-] 其他</td> <td></td> </tr> <tr> <td>[-] 运动</td> <td></td> </tr> <tr> <td>---- ATV31_V1_1</td> <td>ATV 31 CANopen 从站 DSP402 (TEATV3111E.eds)</td> </tr> <tr> <td>---- ATV31_V1_2</td> <td>ATV 31 CANopen 从站 DSP402 (TEATV3112E.eds)</td> </tr> <tr> <td>---- ATV31T_V1_3</td> <td>ATV 31 CANopen 从站 DSP402 (TEATV3112E.eds)</td> </tr> <tr> <td>---- ATV61_V1_1</td> <td>ATV61 (TEATV6111E.eds)</td> </tr> <tr> <td>---- ATV71_V1_1</td> <td>ATV71 (TEATV7111E.eds)</td> </tr> <tr> <td>---- IclA_IFA</td> <td>IclA-IFA CANopen (IclA-IFA.eds)</td> </tr> <tr> <td>---- IclA_IFE</td> <td>IclA-IFE CANopen (IclA-IFE.eds)</td> </tr> <tr> <td>---- IclA_IFS</td> <td>IclA-IFS CANopen (IclA-IFS.eds)</td> </tr> <tr> <td>---- IclA_NO65</td> <td>基于配置文件 DS301V4.0和 DSP402V2的 IclAN065 ...</td> </tr> <tr> <td>---- Lexium05</td> <td>DCX170 CANopen (TEDCX170_0100E.eds)</td> </tr> <tr> <td>---- Lexium05_MFB</td> <td>LXM05A PLCopen (LEXIUM05_MFB.EDS)</td> </tr> <tr> <td>---- Lexium15_HP</td> <td>LEXIUM 15 HP伺服驱动器 (Lexium 15 MP HP.eds)</td> </tr> <tr> <td>---- Lexium15_MP</td> <td>LEXIUM 15 MP伺服驱动器 (Lexium 15 MP HP.eds)</td> </tr> </tbody> </table> <p>子站端通讯器</p>	参考号	描述	[-] CANopen 子站		[-] 其他		[-] 运动		---- ATV31_V1_1	ATV 31 CANopen 从站 DSP402 (TEATV3111E.eds)	---- ATV31_V1_2	ATV 31 CANopen 从站 DSP402 (TEATV3112E.eds)	---- ATV31T_V1_3	ATV 31 CANopen 从站 DSP402 (TEATV3112E.eds)	---- ATV61_V1_1	ATV61 (TEATV6111E.eds)	---- ATV71_V1_1	ATV71 (TEATV7111E.eds)	---- IclA_IFA	IclA-IFA CANopen (IclA-IFA.eds)	---- IclA_IFE	IclA-IFE CANopen (IclA-IFE.eds)	---- IclA_IFS	IclA-IFS CANopen (IclA-IFS.eds)	---- IclA_NO65	基于配置文件 DS301V4.0和 DSP402V2的 IclAN065 ...	---- Lexium05	DCX170 CANopen (TEDCX170_0100E.eds)	---- Lexium05_MFB	LXM05A PLCopen (LEXIUM05_MFB.EDS)	---- Lexium15_HP	LEXIUM 15 HP伺服驱动器 (Lexium 15 MP HP.eds)	---- Lexium15_MP	LEXIUM 15 MP伺服驱动器 (Lexium 15 MP HP.eds)
参考号	描述																																		
[-] CANopen 子站																																			
[-] 其他																																			
[-] 运动																																			
---- ATV31_V1_1	ATV 31 CANopen 从站 DSP402 (TEATV3111E.eds)																																		
---- ATV31_V1_2	ATV 31 CANopen 从站 DSP402 (TEATV3112E.eds)																																		
---- ATV31T_V1_3	ATV 31 CANopen 从站 DSP402 (TEATV3112E.eds)																																		
---- ATV61_V1_1	ATV61 (TEATV6111E.eds)																																		
---- ATV71_V1_1	ATV71 (TEATV7111E.eds)																																		
---- IclA_IFA	IclA-IFA CANopen (IclA-IFA.eds)																																		
---- IclA_IFE	IclA-IFE CANopen (IclA-IFE.eds)																																		
---- IclA_IFS	IclA-IFS CANopen (IclA-IFS.eds)																																		
---- IclA_NO65	基于配置文件 DS301V4.0和 DSP402V2的 IclAN065 ...																																		
---- Lexium05	DCX170 CANopen (TEDCX170_0100E.eds)																																		
---- Lexium05_MFB	LXM05A PLCopen (LEXIUM05_MFB.EDS)																																		
---- Lexium15_HP	LEXIUM 15 HP伺服驱动器 (Lexium 15 MP HP.eds)																																		
---- Lexium15_MP	LEXIUM 15 MP伺服驱动器 (Lexium 15 MP HP.eds)																																		
3	<p>输入对应于此地址的连接点编号。 缺省情况下， Unity Pro 软件提供第一个空闲的连续地址。</p>																																		
4	<p>在通讯器字段中，选择可启用 CANopen 总线上的通讯的元素类型。 对于带有内置通讯器的模块，不显示此窗口。</p>																																		

步骤	操作
5	<p>验证后单击确定。</p> <p>结果：此模块被声明。</p> 

如何删除 / 移动 / 复制总线设备

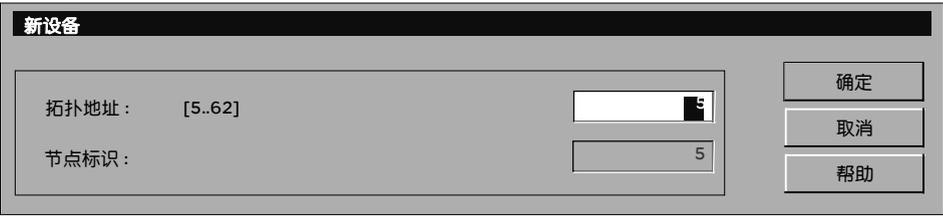
删除设备的过程 使用此操作，可通过软件删除连接到 CANopen 总线的设备：

步骤	操作
1	访问 CANopen 配置屏幕。
2	右键单击要删除的设备的连接点，然后单击 删除子站 。

移动设备的过程 移动设备不是在总线上进行物理移动，而是变更设备地址逻辑。移动会触发程序中输入 / 输出对象的地址修改以及与这些对象关联的变量的移动。

步骤	操作
1	访问 CANopen 配置屏幕。
2	选择要移动的连接点（会有一个框环绕着选定的连接点）。
3	拖放要移动至空连接点的连接点。 结果： 将显示 移动设备 屏幕： <div data-bbox="240 738 1179 954" style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> </div>
4	输入目标连接点的编号。
5	通过按 确定 确认新的连接点。 结果： 将显示 移动设备 屏幕： <div data-bbox="240 1071 1083 1291" style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> </div>
6	通过按 是 确认修改，以修改程序中输入 / 输出对象的地址并移动与这些对象关联的变量。

复制设备的过程 此功能类似于移动设备的功能：

步骤	操作
1	访问 CANopen 配置屏幕。
2	右键单击要复制的设备，然后单击 复制 。
3	右键单击所需的连接点，然后单击 粘贴 。 结果： 将显示 新设备 屏幕： 
4	输入目标连接点的编号。
5	通过按 确定 确认新的连接点。

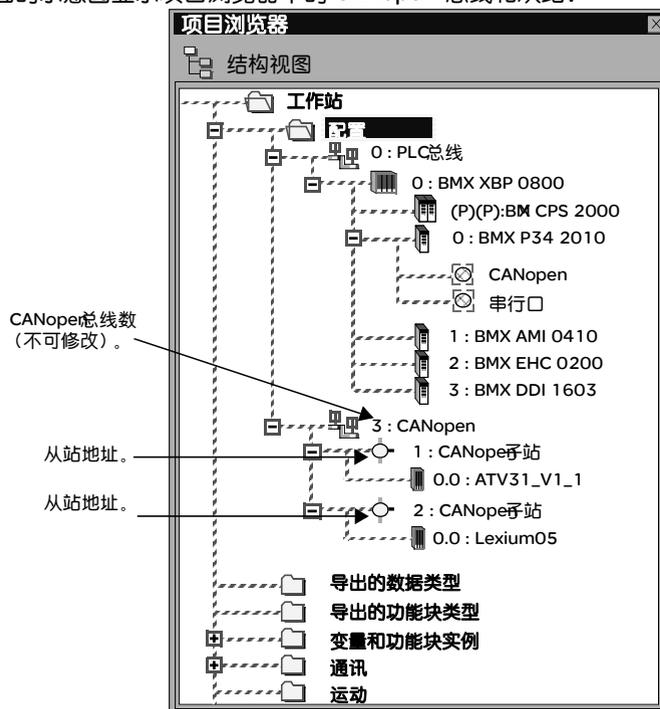
在项目浏览器中查看 CANopen 总线

概览

CANopen 总线显示在项目浏览器的配置目录中。总线的数目是由 Unity Pro 自动计算的。

注意：总线的数目值不能修改。

下面的示意图显示项目浏览器中的 CANopen 总线和从站：



5.3 设备配置

概览

本节主题

本节介绍如何配置 CANopen 设备的初始参数。

配置初始参数有三种方法：

- 使用 Unity 配置
- 使用外部工具配置
- 手动配置

注意：在配置设备之前，强烈建议您选择功能（如果可用）。

本章节包含了哪些内容？

本章节包含了以下主题：

主题	文件集
从站功能	79
使用 Unity 进行配置	83
使用外部工具的配置：配置软件	89
手动配置	92

从站功能

概览

为便于配置使用，某些 CANopen 设备使用功能来表示。

每个功能定义了预映射的 PDO，以及某些可映射的调试变量（从站配置屏幕的 PDO 选项卡）。

注意：应在配置从站前选择功能。

可用功能

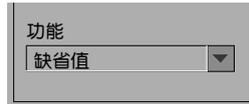
可用功能如下：

功能	描述	涉及的设备
基本	此功能允许对速度进行简单的控制。	ATV
标准	此功能允许控制速度和 / 或扭矩。 所有可映射的参数都在补充的 PDO 中映射，用于： <ul style="list-style-type: none"> ● 调整操作参数（加速长度） ● 额外监视（当前值 ...） ● 额外控制（PID、输出命令 ...） 	
高级	此功能允许控制速度和 / 或扭矩。 某些参数可配置并可在 PDO 中映射，用于： <ul style="list-style-type: none"> ● 调整操作参数（加速长度） ● 额外监视（当前值 ...） ● 额外控制（PID、输出命令 ...） 	

功能	描述	涉及的设备
简单	<p>如果岛不包含高精度模拟量 I/O 模块或 TeSys U STB 模块，则使用此配置文件。此配置文件包括：</p> <ul style="list-style-type: none"> ● NIM 诊断信息（索引 4000 -索引 4006）， ● 8 位离散量信息（索引 6000）， ● 16 位离散量信息（索引 6100）， ● 8 位离散量输出信息（索引 6200）， ● 16 位离散量输出信息（索引 6300）， ● 低精度模拟量输入信息（索引 6401）， ● 低精度模拟量输出信息（索引 6411）。 <p>此配置文件将任何上述对象的索引或子索引条目的数量限制为 32。如果岛配置超过此限制，请使用较大的配置文件。</p>	STB NCO1010 & NCO2212
扩展	<p>如果岛包含高精度模拟量 I/O 模块或 TeSys U STB 模块，则使用此配置文件。此配置文件包括：</p> <ul style="list-style-type: none"> ● NIM 诊断信息（索引 4000 -索引 4006）， ● 8 位离散量信息（索引 6000）， ● 16 位离散量信息（索引 6100）， ● 8 位离散量输出信息（索引 6200）， ● 16 位离散量输出信息（索引 6300）， ● 低精度模拟量输入信息（索引 6401）， ● 低精度模拟量输出信息（索引 6411）， ● 高精度模拟量输入信息（索引 2200-221F）， ● 高精度模拟量输入信息（索引 3200-221F）， ● TeSys U 输入信息（索引 2600-261F）， ● TeSys U 输出信息（索引 3600-361F）。 <p>此配置文件将任何上述对象的索引或子索引条目的数量限制为 32。如果岛配置超过此限制，请使用较大的配置文件。</p>	

功能	描述	涉及的设备
高级	<p>如果岛包含增强的 CANopen 设备、作为运行时参数的特殊特性及高精度模拟量 I/O 模块或 HMI 或 TeSys U STB 模块，则请使用此配置文件。</p> <p>此配置文件包括：</p> <ul style="list-style-type: none"> ● NIM 诊断信息（索引 4000 -索引 4006）， ● 8 位离散量信息（索引 6000）， ● 16 位离散量信息（索引 6100）， ● 8 位离散量输出信息（索引 6200）， ● 16 位离散量输出信息（索引 6300）， ● 低精度模拟量输入信息（索引 6401）， ● 低精度模拟量输出信息（索引 6411）， ● 高精度模拟量输入信息（索引 2200-221F）， ● 高精度模拟量输入信息（索引 3200-221F）， ● TeSys U 输入信息（索引 2600-261F）， ● TeSys U 输出信息（索引 3600-361F）， ● 第三方 CANopen 设备（索引 2000-201F）， ● RTP 信息（索引 4100 &索引 4101）。 <p>此配置文件将任何上述对象的索引或子索引条目的数量限制为 32。如果岛配置超过此限制，请使用较大的配置文件。</p>	STB NCO2212
较大配置文件	如果岛配置不适合于上述任何配置文件，请使用此配置文件。此配置文件包含可用于 STB 岛的所有对象，因此，其将在 CANopen 主站中消耗更多的存储器地址位置。	STB NCO1010 & NCO2212
控制	此功能是专门为带内置控制器卡和所有应用程序卡（泵控制 ...）的 CANopen 通讯设计的。	ATV 61/71
基本	基本级别用于配置无 CP 扩展的阀门终端。	Festo CPV
CP_Extension	此级别用于配置阀门终端（包括 CP 扩展）。	
Basic_DIO_only	此基本级别仅用于配置带有气压阀门和数字量 I/O 的 CPX。	
Generic_DIO_AIO	一般 DS401 级别用于配置 CPX 阀门和 I/O（包括模拟量 I/O 模块）。	Festo CPX
高级	此高级级别用于配置最大 I/O 数和完全参数集。	
缺省值	此功能是某些设备的缺省功能。可修改此功能。	除 ATV 和 Lexium 外 的所有从站

注意：某些设备只能处理一种功能。在这种情况下，功能会以灰色显示，不能对其进行修改。



使用 Unity 进行配置

概览

有些设备可直接使用 Unity 进行配置：

- FTB
- Osicoder
- OTB
- Preventa
- STB NCO 1010
- Tego Power
- Festo 阀门终端
- Parker Moduflex

过程

要配置从站，请执行下列操作：

步骤	操作
1	访问 CANopen (见 如何访问 CANopen 总线配置屏幕页 69) 总线配置屏幕。
2	双击要配置的从站。
3	使用 配置 选项卡配置用法功能。
4	使用 PDO 选项卡配置 PDO 。
5	使用 错误控制 选项卡选择错误控制。

"配置" 选项卡 下图显示从站配置屏幕的示例：



下表显示配置屏幕的各个元素及其功能：

编号	元素	功能
1	选项卡	前景中的选项卡指示显示的屏幕类型。本例显示的是配置屏幕。
2	模块区域	提供设备的简称。

编号	元素	功能
3	通道区域	<p>可在此区域选择要配置的通讯通道。</p> <p>单击设备可显示以下选项卡：</p> <ul style="list-style-type: none"> ● 描述：给出设备的特性 ● CANopen：用于访问 SDO（在在线模式下） ● I/O 对象：允许预先用符号表示输入 / 输出对象 ● 故障：只能在在线模式下访问 <p>单击通道可显示以下选项卡：</p> <ul style="list-style-type: none"> ● PDO（输入 / 输出对象） ● 错误控制 ● 配置 ● 调试，只能在在线模式下访问 ● 诊断，只能在在线模式下访问
4	常规参数区域	此字段用于选择从站功能。
5	配置区域	<p>此区域用于设置设备的通道。</p> <p>有些设备可通过外部工具进行配置。在此例中，配置存储在设备中，您不能输入配置参数，因为此字段为空。</p>

注意：有关常规、配置、调整和调试参数的信息，请参考每种设备的文档。

注意：设备采用其配置时，所有参数都不会发送。CPU 只发送与缺省值不同的参数。

PDO 选项卡

通过 PDO 可以管理 CANopen 主站与从站之间的通讯流。PDO 选项卡可用来配置 PDO。

此屏幕分为 3 部分：

PDO	Tr.Ty...	抑制 ...	偶 ...	符号	拓扑地址	%M...	CO...	索引
PDO 1(...)	255	0	0				16#181	
状态 ...				lexium...	%IW\3.1\0.0...	%MW16		6041:...
PDO 2(...)	255	0	100				16#281	
状态 ...				lexium...	%IW\3.1\0.0...	%MW16		6041:...
位置 ...				lexium...	%ID\3.1\0.0.0...	%MW8		6064:...
PDO 3(...)	255	0	100				-	
状态 ...				lexium...	%IW\3.1\0.0...	%MW16		6041:...
速度 ...				lexium...	%ID\3.1\0.0.0...	%MW10		606C:...
PDO 4(...)	254	0	0				-	

PDO	Tr.Ty...	抑制 ...	偶 ...	符号	拓扑地址	%M...	CO...	索引
PDO 1(...)	255						16#281	
控制 ...				lexium...	%QW\3.1\0.0...	%MW42		6040:...
PDO 2(...)	255						16#381	
控制 ...				lexium...	%QW\3.1\0.0...	%MW42		6040:...
目标 ...				lexium...	%QD\3.1\0.0...	%MW41		607A:...
PDO 3(...)	255						-	
控制 ...				lexium...	%QW\3.1\0.0...	%MW42		6040:...
目标 ...				lexium...	%QD\3.1\0.0...	%MW41		60FF:...
PDO 4(...)	254						-	

参数名称	索引 ...
RAMPsym	3006:01
_IO_act	3008:01
ANA1_act	3009:01
ANA2_act	3009:05
PLCopenRx1	301B:05
PLCopenRx2	301B:06
PLCopenTx1	301B:07
PLCopenTx2	301B:08
JOGactivate	301B:09
_actionStatus	301C:04
_p_actRAMPusr	301F:02
CUR_I_target	3020:04
SPEEDn_target	3021:04
PTPp_abs	3023:01
PTPp_relpref	3023:03
PTPp_target	3023:05
PTPp_relpact	3023:06
GEARdenom	3026:03
GEARnum	3026:04
控制字	6040:00
状态字	6041:00
位置实际值 ...	6063:00

- **传输 PDO**：由从站向主站传输的信息。
- **接收 PDO**：从站从主站接收到的信息。
- **变量**：可映射到 PDO 的变量。要将变量分配给 PDO，请将变量拖放至目标 PDO 中。不能为静态 PDO 分配任何变量。

注意：要配置 STB NCO 1010，需要确定对此设备有效的所有对象并在 PDO 中手动对其进行配置。

有关所关联对象的列表的更多信息，请参考 STB 用户手册。

有关如何使用 PDO 的更多信息，请参见 [...]。

"错误控制" 选项卡 CANopen 从站模块的**错误控制**选项卡允许您配置故障监控。



可以使用以下两种选择：

- **心跳：**心跳机制的原理是循环发送由心跳生成器生成的存在性消息。心跳传输器（生成器）周期性地发送消息。发送时间可通过节点心跳生成器时间值配置。网络上连接的一个或多个元件接收此消息。心跳消费者对心跳消息接收进行观察。如果持续时间超出心跳消费者时间（ $1.5 * \text{生成器心跳时间}$ ），则创建心跳事件，并且设备进入缺省状态。

如果 CANopen 总线上使用了 M340 主站 PLC，则所有使用心跳控制模式的节点都是生成器。主站会对消息的传输和接收进行观察，并且它是节点发送的心跳消息的唯一接收者

主站可向从站发送心跳消息。主站心跳生成器时间设置为 300 毫秒，不可修改。

- **节点防护：**节点防护是监控网络节点的过程。NMT（网络管理）主站定期（此周期被为“防护时间”）发送 RTR（远程传输请求），相关的节点必须在给定的时间（节点使用寿命等于防护时间乘以使用寿命因数）内进行应答。使用寿命值设置为 2，不可修改。

注意：有些设备只支持心跳或节点防护。对于支持心跳和节点防护的设备，在 Unity Pro 中只能选择心跳机制。

使用外部工具的配置：配置软件

概览

要配置 STB NCO 2212、Lexium 05/15 IclA、Tesys U 或 ATV61/71 设备，需要使用外部工具：

- 用于 STB 的 Advantys 配置软件
- 用于 Lexium 05 的 PowerSuite for Lexium 05 V2.2.0 补丁 V2.2.0B 软件
- 用于 ATV31、ATV61、ATV71 和 Tesys U 的 Powersuite V2.0 软件
- 用于 Lexium 15 LP 的 UNILINK V1.5
- 用于 Lexium 15 MH 的 UNILINK V4.0
- 用于 IclA N065 的 ICLA CCT
- 用于 ICLA_IFA、ICLA_IFE、ICLA_IFS 的 EasyIclA V1.104

注意：对于运动和驱动设备，强烈建议将软件与 Unity MFB 配合使用，以便于配置和编程。

Advantys 配置软件

必须使用 Advantys 配置软件（2.5 版本）来配置 STB NCO 2212。Advantys 配置软件会验证配置并创建一个 DCF 文件，此文件包含按照适当顺序排列的配置中使用的对象。可从 Unity Pro 导入 DCF 文件。

注意：只能使用 Advantys 完整版创建 DCF 文件。



意外操作的风险

在配置 STB Island 期间由 Advantys 生成的符号文件 *.xsy 不得用于 Unity Pro 中。During an 将 *.xsy 文件从 Advantys 导入 Unity Pro for CAN 期间，不支持开放设备。

在 PDO 表中指定的 %MW 对象与在 CAN 开放头的配置中定义的对象不属于同一范围。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

向 CANopen 总线添加岛的过程如下所示：

步骤	操作
1	在 Advantys 配置软件（2.2 版或更高版本）中，创建一个新岛。
2	选择 STB NCO 2212 网络接口模块。
3	选择将在应用程序中使用的模块。
4	配置岛。
5	配置完成后，单击 文件 / 导出 将岛导出为 DCF 格式。 此时将显示以下窗口：
	
6	单击 确定 以进行确认。
7	文件导出后，启动 Unity Pro 并打开将在其中使用该岛的项目。
8	向总线编辑器添加一个 STB 设备（请参见如何在总线上添加设备）。
9	右键单击 STB 设备，然后单击 打开模块 。
10	在 PDO 选项卡上，单击按钮 导入 DCF 。
11	单击 确定 以进行确认。PDO 自动进行配置。

注意：修改岛的拓扑结构需要重新执行此过程。
有关 STB 配置的更多信息，请参考 STB 用户手册。

Powersuite 软件 PowerSuite 软件是为实现后续 ATV 速度驱动器而开发的工具。应使用此软件配置 ATV31/61/71 Tesys U 或 Lexium O5 设备 (Powersuite 2)

该软件集成了各种功能，以供实现阶段使用，如：

- 配置准备
- 开始工作
- 维护

配置直接存储在设备中。

有关使用 Powersuite 软件配置 ATV31/61/71和 Tesys U 以及有关使用 PowerSuite 2 配置 Lexium O5 的详细信息，请参考设备用户手册。

UNILINK 软件 UNILINK 提供了用于 Lexium O5 伺服驱动器的简化参数设置。它可用于根据相关的 SER/BPH 无刷电机和应用要求配置、设置和调整 Lexium MMDA/MHDS 驱动器。

有关使用 UNILINK 配置 Lexium 15 的更多信息，请参考 Lexium 用户手册。

ICLA CCT 软件 ICLA CCT 软件用于配置 ICL A N065。它包含一个图形用户界面，可用于试运行、诊断和测试。

ICLA CCT 提供以下功能：

- 输入和显示设备参数
- 归档和复制设备参数
- 显示状态和设备信息
- 定位电机与 PC
- 初始化参考运动
- 访问文档中收录的所有参数
- 诊断操作故障

有关使用 ICLA CCT 配置 ICL A N065 的更多信息，请参考该软件的用户手册。

注意：要使用 ICLA CCT 配置 ICL A N065，需要 USB/CANopen 转换器。

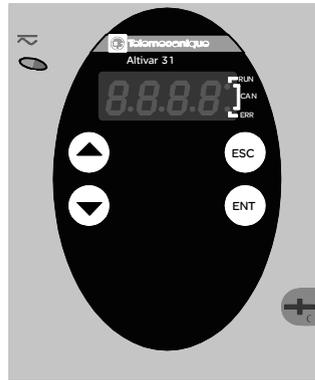
手动配置

概览

可以通过其前面板手动配置 ATV 31 设备和 Icla (N065 除外) 设备。

配置 ATV 31

下图显示 ATV 31 伺服驱动器的不同前面板。



ATV31xxxxx



ATV31xxxxxA

ATV 31 可配置如下：

步骤	操作
1	按 "ENT" 键进入 ATV31 配置菜单。
2	使用 "箭头" 键选择 "COM" 通讯菜单，然后使用 "ENT" 键确认。
3	使用 "箭头" 键选择 "AdCO" 菜单，然后使用 "ENT" 键确认。 输入一个值（CANopen 总线上的地址）。 使用 "ENT" 键确认，然后使用 "ESC" 键退出菜单。
4	使用 "箭头" 键选择 "bdCO" 菜单，然后使用 "ENT" 键确认。 输入一个值（CANopen 总线上的速度）。 使用 "ENT" 键确认，然后使用 "ESC" 键退出菜单。
5	通过数次按 "ESC" 键退出配置菜单。

注意：只能在电机停止并且变速控制器锁定（外罩闭合）时修改配置。输入的任何修改都将在速度控制器的一个“关/开”循环后生效。

有关 ATV31 配置的更多信息，请参考 ATV 速度驱动器用户手册。

Icla 设备配置 Icla 设备（除 Icla N065 外）具有可以配置地址和速度的开关。

注意：有关 STB 配置的更多信息，请参考 STB 用户手册。

5.4 主站配置

概览

本节主题 本节介绍主站配置。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
如何访问 CANopen 主站配置屏幕	95
CANopen 主站配置屏幕	97
主站配置屏幕描述	99

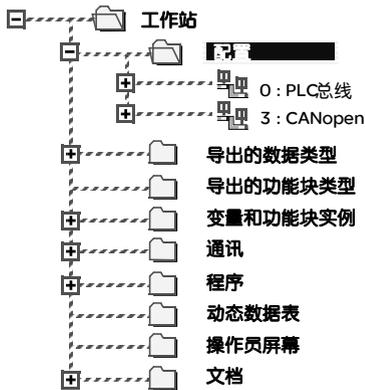
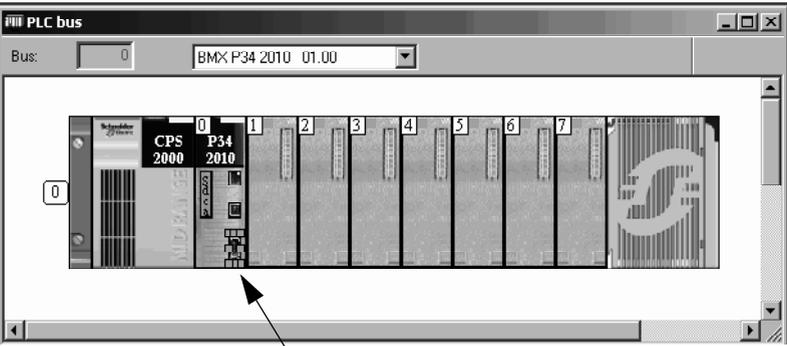
如何访问 CANopen 主站配置屏幕

概览

本章描述如何访问带内置 CANopen 链路的 Modicon M340 PLC 的主站配置屏幕。

过程

要访问主站，请执行以下操作：

步骤	操作
1	<p>在项目导航器中，展开配置目录。</p> <p>结果：将出现以下屏幕：</p> 
2	<p>双击 PLC 总线子目录。</p> <p>结果：将出现以下屏幕：</p>  <p style="text-align: center;">CANopen端口</p> <p>双击处理器的 CANopen 端口。</p>

步骤	操作
3	<p>主站配置屏幕出现：</p> 

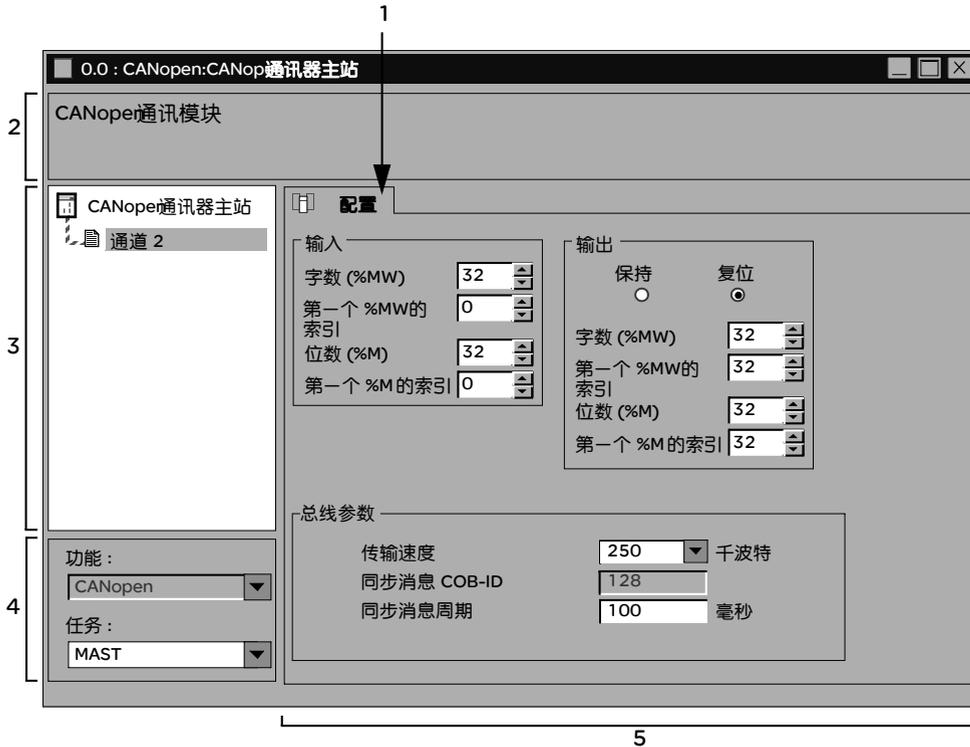
CANopen 主站配置屏幕

概览

此屏幕用于从 Modicon M340 PLC 工作站声明和配置 CANopen 网络的主站。

示意图

主站的配置屏幕如下所示：



元素和功能

下表对构成主站配置屏幕的各个区域进行了描述：

读取	编号	功能
1	选项卡	前景中的选项卡指示显示的屏幕类型。本例显示的是配置屏幕。
2	模块	此区域由配备 CANopen 端口的处理器的缩写标题组成。
3	通道	<p>可在此区域选择要配置的通讯通道。</p> <p>单击设备可显示以下选项卡：</p> <ul style="list-style-type: none"> ● 描述：给出内置 CANopen 端口的特性 ● 输入 / 输出对象：允许预先用符号表示输入 / 输出对象 <p>单击通道可显示以下选项卡：</p> <ul style="list-style-type: none"> ● 配置：允许您声明和配置 CANopen 主站 ● 调试：仅在在线模式下可用 ● 故障：只能在在线模式下访问
4	常规参数	<p>通过此字段可以：</p> <ul style="list-style-type: none"> ● 选择通讯功能（不可修改） ● 将 CANopen 总线与应用程序任务相关联： <ul style="list-style-type: none"> ● MAST 是主任务 ● FAST 是快速任务 <p>任务与总线上的交换并不同步。</p>
5	配置	<p>通过此字段可以：</p> <ul style="list-style-type: none"> ● 配置 PLC 内部存储器地址（将从此地址周期性复制 CANopen 设备的输入）。 ● 配置 PLC 内部存储器地址（将从该地址周期性读取来自 CANopen 设备的输出）。 ● 配置 CANopen 总线的参数。

主站配置屏幕描述

概览

配置屏幕允许您配置总线参数以及输入和输出。

输入

下图显示了输入配置区域：

输入

字数 (%MW) 32

第一个 %MW 的索引 0

位数 (%M) 32

第一个 %M 的索引 0

要配置总线从站的输入，必须指定将输入定期复制到其中的存储区。要定义此区域，必须指定：

- 字数 (%MW)：0 至 32,464
- 第一个字的地址：0 至 32,463
- 位数 (%M)：0 至 32,634
- 第一个位的地址：0 至 32,633

输出

下图显示了输出配置区域：

输出

保持 复位

字数 (%MW) 32

第一个 %MW 的索引 32

位数 (%M) 32

第一个 %M 的索引 32

故障预置模式（维护 / 复位）允许定义设备在 CPU 处于 " 停止 " 或 " 暂停 " 时的行为：

- **维护**：维护输出（保持值）
- **复位**：复位输出（值被设定为 0）

要配置输出，与配置输入类似，必须指定将包含总线从站输出值的字和位表：

- 字数 (%MW)：1 至 32,464
- 第一个字的地址：0 至 32,463
- 位数 (%M)：1 至 32,634
- 第一个位的地址：0 至 32,633

注意：字表和位表是在 PLC 内部存储器中找到的。每个表的两个区域之间禁止出现任何交叉。输入的位区域不能与输出的位区域有重叠。输入的字区域不能与输出的字区域有重叠。

警告

意外的设备操作

在安装中注意采取每项安全预防措施，以便输出的位置在 CANopen 总线停止时保持安全。CANopen 总线停止时，行为特定于所连接的设备。请参见这些设备的用户手册。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

总线参数

下图显示了总线参数配置区域：

总线参数	
传输速度	250 千波特
同步消息 COB-ID	128
同步消息周期	100 毫秒

要配置总线，必须指定：

- 传输速度（见总线长度页 23）：250 千波特（缺省值）
- 同步消息的 COB-ID：128（缺省值）
- 同步消息周期：100 毫秒（缺省值）

语言对象

下面列出的参数用 %KW 语言对象的形式表示：

读取	参数	语言对象
输入	字数 %MW	%KW8
	第一个字的索引	%KW10
	位数 %M	%KW4
	第一个位的索引	%KW6
输出	故障预置模式	%KW0
	字数 %MW	%KW9
	第一个字的索引	%KW11
	位数 %M	%KW5
	第一个位的索引	%KW7
总线参数	传输速度	%KW1
	同步消息 COB-ID	%KW2
	同步消息周期	%KW3

Catalog Manager 软件实现



概览

本章主题

本章描述了 Catalog Manager 软件实现。

本章包含了哪些内容?

本章包含了以下部分:

段	段	文件集
6.1	Catalog Manager 概述	104
6.2	使用 Catalog Manager	111
6.3	Catalog Manager 故障排除	135

6.1 Catalog Manager 概述

概览

本节主题 本节提供了 Catalog Manager 概述

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
Catalog Manager 描述	105
Catalog Manager 内容	107

Catalog Manager 描述

概述

Catalog Manager 是一种软件工具，它简化了对 Unity Pro 目录数据库中 CANopen 设备的管理。

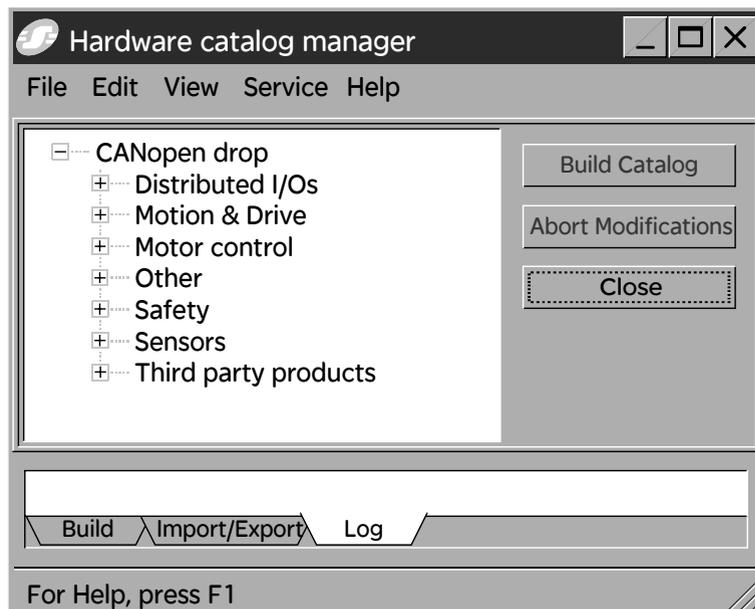
Catalog Manager 是一种独立的软件，可用来：

- 集成第三方产品。
- 添加、删除和配置对现场总线上 CANopen 设备的访问。
- 将为指定设备保留的 CPU 存储器的大小降至最低值。
- 自定义用户界面。

Catalog Manager 在一个目录数据库上工作，后者随 Unity Pro 一起安装并使用一些来自 Unity Pro 安装的基本服务。因此 Unity Pro 软件必须安装在您的工作站上，这样 Catalog Manager 才能正常工作。

设备将被添加到 Unity Pro 的标准目录中，在项目中使用这些设备的方式与使用随 Unity Pro 提供的所有设备都相同。

下图显示了 Catalog Manager 的主屏幕。



输出窗口的上下文菜单包含了下列信息：

- Build：用于显示与生成进度有关的信息。
- Import / export：用于显示与导入/导出有关的信息。
- Log：用于在“Add Device”或“Add Function”期间显示信息、警告和错误。

注意： Catalog Manager 只有英语版本。
一次只能打开 Catalog Manager 的一个实例。

Catalog Manager 内容

概述

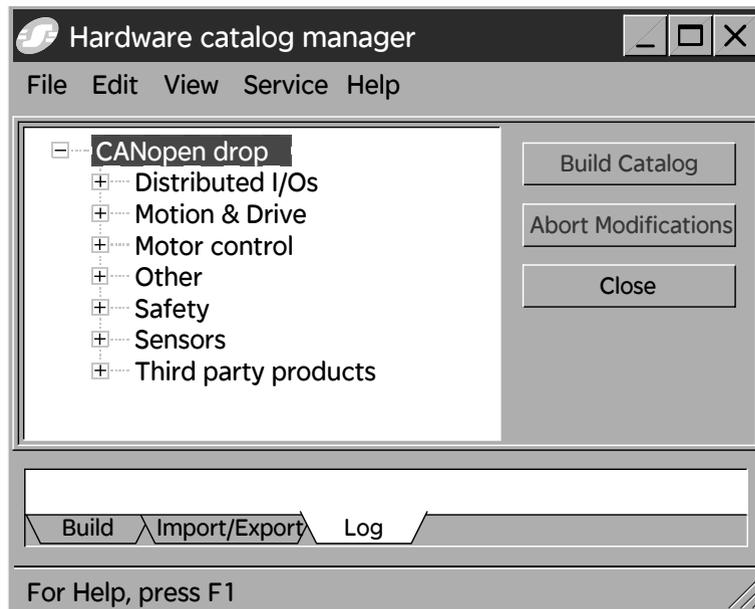
Catalog Manager 由三个级别的数据组成：

- 设备系列
- 设备
- 功能

设备系列

设备系列包含特定于每个系列的所有设备：分布式 I/O、运动驱动器、运动控制、其他、安全、传感器和第三方产品。

下图显示了不同的设备系列。



项目“View”是一个带下列信息的上下文菜单：

- Status Bar：显示 / 隐藏状态栏。
- Output Window：显示 / 隐藏输出窗口。
- View Function：显示选定的功能。

设备

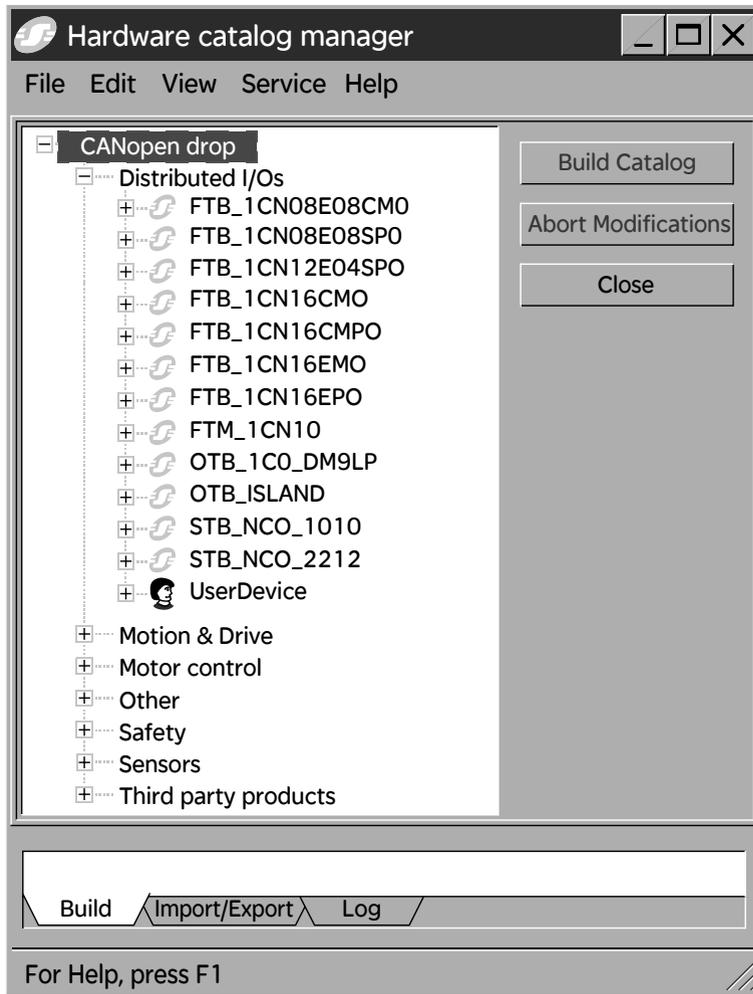
设备是独立的外部单元，可以提供一个或多个不同的功能。

设备在目录中以其名称来标识。缺省名称源自 EDS 文件，但可以对其进行修改。设备名称在目录中必须唯一。

有两种类型的设备：

- 预编程的设备
有关这些设备的信息随 Unity Pro 软件一起提供。
用户不能修改预编程设备的列表。Schneider 设备名称的左侧会显示一个 Schneider 徽标图标。
- 用户设备
不属于任何 Schneider 产品的设备都被视为“用户设备”。
可以在 Catalog Manager 中删除或重新配置用户设备。已经添加但尚未生成的用户设备将显示为带一个  图标。已经生成的设备将由  图标表示。

下图显示了一组分布式 I/O 设备和一个用户设备。



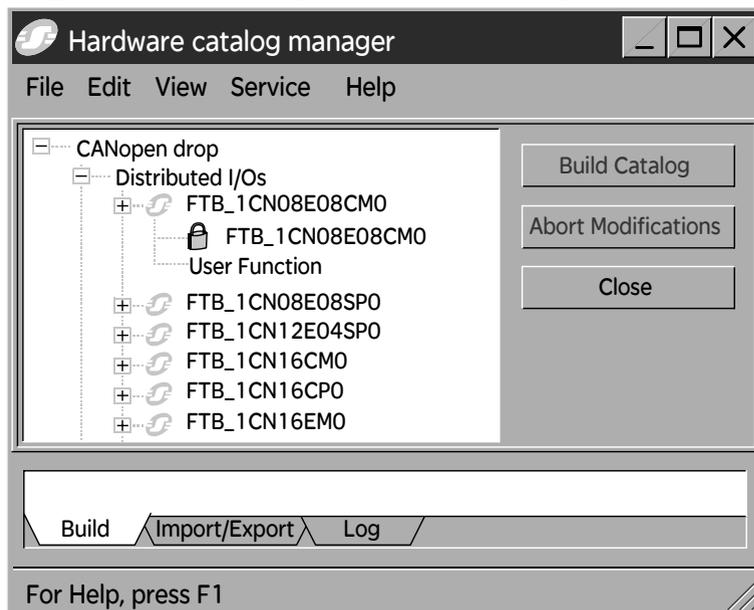
功能

功能是设备内的一个子类别。用户创建功能的目的是仅使用该设备的有限的功能子集。将为第一个功能提议缺省名称，但该名称可以修改。功能名称在给定的设备中必须唯一。每个功能都有自己的 IODDT。

功能可分为以下两种类型：

- 预编程的功能
 - 有关这些功能的信息随 Unity Pro 软件一起提供。
 - 用户不能修改预编程功能，这些功能通过其左侧的挂锁图标指出。
- 用户功能
 - 不属于任何 Schneider 产品的功能都被视为“用户功能”。
 - 可以在 Catalog Manager 中删除或重新配置用户功能。

下图显示了一个同时带预编程功能和用户功能的预编程设备。



示例：

设备系列：分布式 I/O

设备：FTB_1CN08E0BCMO

功能：

缺省（Schneider 功能，写保护）

TestFunction（用户功能）

6.2 使用 Catalog Manager

概览

本节主题 本节提供了使用 Catalog Manager 的不同步骤。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
如何启动 Catalog Manager	112
如何将设备添加到 Catalog Manager	113
如何在设备上添加功能	116
基本配置参数	117
Expert Mode 配置参数	121
如何复制或删除功能	129
如何导入 / 导出或删除一个或多个用户设备	130
如何关闭 Catalog Manager	133
如何创建专用和优化 STB 岛的示例	134

如何启动 Catalog Manager

概览

这是启动 Catalog Manager 的过程。

注意： Catalog Manager 和 Unity Pro 软件不能同时一起运行。

过程

下表显示了启动 Catalog Manager 的过程

步骤	操作
1	检验 Unity Pro 软件是否正在运行 <ul style="list-style-type: none">● 如果 Unity Pro 软件正在运行，关闭它● 如果 Unity Pro 软件没有运行，转到下一步
2	选择： 开始 -> 程序 -> Schneider Electric->Unity Pro->Hardware Catalog Manager

如何将设备添加到 Catalog Manager

概览

这是将设备添加到 Catalog Manager 的过程。



第三方设备集成

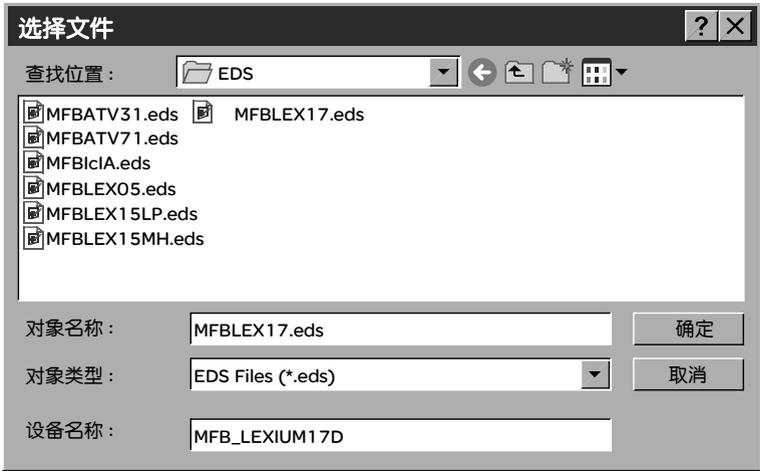
如果某个新设备尚不属于 UnityPro 目录，那么可能会因为设备本身、EDS 文件或两者之间的任何不匹配而导致一些问题。请联系 Schneider 支持团队以便我们帮助您找到适当的解决方案。

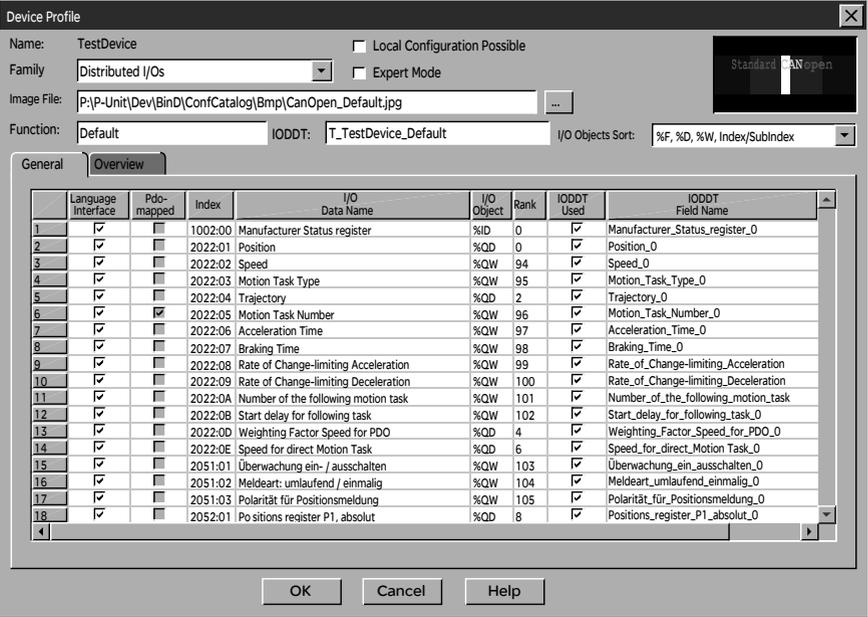
如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

过程

已使用 EDS 文件将新设备添加到 Catalog Manager 中。

在执行下表中的过程之前，您必须知道与要添加的新设备对应的 EDS 文件的名称和所在的位置。

步骤	操作
1	打开 Catalog Manager 。
2	<ul style="list-style-type: none"> ● 选择 Edit->Add Device 或者 <ul style="list-style-type: none"> ● 右键单击设备系列。 ● 选择 “Add Device”
3	显示一个 Windows 对话框，选择要导入的 EDS 文件： 
4	<ul style="list-style-type: none"> ● 选择与要添加的设备对应的 EDS 文件名称 ● 输入设备的唯一名称（可选）。 一次只能选择一个 EDS 文件，不允许多选。 在缺省情况下，EDS 文件名称将被用作设备的名称，但可以对其进行修改。 在后面的步骤中将无法更改设备名称。

步骤	操作																																																																																																																																																																											
5	<p>出现一个屏幕，显示新设备参数。下面是一个示例：</p>  <p>The screenshot shows the 'Device Profile' dialog box. The 'Name' field is 'TestDevice'. The 'Family' is 'Distributed I/Os'. The 'Image File' is 'P:\P-Unit\Dev\BinD\ConfCatalog\Bmp\CanOpen_Default.jpg'. The 'Function' is 'Default' and the 'IODDT' is 'T_TestDevice_Default'. The 'I/O Objects Sort' is '%F, %D, %W, Index/SubIndex'. The 'General' tab is selected, showing a table of I/O objects:</p> <table border="1" data-bbox="330 428 1144 781"> <thead> <tr> <th></th> <th>Language Interface</th> <th>Pdo-mapped</th> <th>Index</th> <th>I/O Data Name</th> <th>I/O Object</th> <th>Rank</th> <th>IODDT Used</th> <th>IODDT Field Name</th> </tr> </thead> <tbody> <tr><td>1</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>1002:00</td><td>Manufacturer Status register</td><td>%ID</td><td>0</td><td><input checked="" type="checkbox"/></td><td>Manufacturer_Status_register_0</td></tr> <tr><td>2</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:01</td><td>Position</td><td>%QD</td><td>0</td><td><input checked="" type="checkbox"/></td><td>Position_0</td></tr> <tr><td>3</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:02</td><td>Speed</td><td>%QW</td><td>94</td><td><input checked="" type="checkbox"/></td><td>Speed_0</td></tr> <tr><td>4</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:03</td><td>Motion Task Type</td><td>%QW</td><td>95</td><td><input checked="" type="checkbox"/></td><td>Motion_Task_Type_0</td></tr> <tr><td>5</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:04</td><td>Trajectory</td><td>%QD</td><td>2</td><td><input checked="" type="checkbox"/></td><td>Trajectory_0</td></tr> <tr><td>6</td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td>2022:05</td><td>Motion Task Number</td><td>%QW</td><td>96</td><td><input checked="" type="checkbox"/></td><td>Motion_Task_Number_0</td></tr> <tr><td>7</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:06</td><td>Acceleration Time</td><td>%QW</td><td>97</td><td><input checked="" type="checkbox"/></td><td>Acceleration_Time_0</td></tr> <tr><td>8</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:07</td><td>Braking Time</td><td>%QW</td><td>98</td><td><input checked="" type="checkbox"/></td><td>Braking_Time_0</td></tr> <tr><td>9</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:08</td><td>Rate of Change-limiting Acceleration</td><td>%QW</td><td>99</td><td><input checked="" type="checkbox"/></td><td>Rate_of_Change-limiting_Acceleration</td></tr> <tr><td>10</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:09</td><td>Rate of Change-limiting Deceleration</td><td>%QW</td><td>100</td><td><input checked="" type="checkbox"/></td><td>Rate_of_Change-limiting_Deceleration</td></tr> <tr><td>11</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:0A</td><td>Number of the following motion task</td><td>%QW</td><td>101</td><td><input checked="" type="checkbox"/></td><td>Number_of_the_following_motion_Task</td></tr> <tr><td>12</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:0B</td><td>Start delay for following task</td><td>%QW</td><td>102</td><td><input checked="" type="checkbox"/></td><td>Start_delay_for_following_task_0</td></tr> <tr><td>13</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:0D</td><td>Weighting Factor Speed for PDO</td><td>%QD</td><td>4</td><td><input checked="" type="checkbox"/></td><td>Weighting_Factor_Speed_for_PDO_0</td></tr> <tr><td>14</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2022:0E</td><td>Speed for direct Motion Task</td><td>%QD</td><td>6</td><td><input checked="" type="checkbox"/></td><td>Speed_for_direct_Motion_Task_0</td></tr> <tr><td>15</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2051:01</td><td>Überwachung ein- / ausschalten</td><td>%QW</td><td>103</td><td><input checked="" type="checkbox"/></td><td>Überwachung_ein_ausschalten_0</td></tr> <tr><td>16</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2051:02</td><td>Meldeart: umlaufend / einmalig</td><td>%QW</td><td>104</td><td><input checked="" type="checkbox"/></td><td>Meldeart_umlaufend_einmalig_0</td></tr> <tr><td>17</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2051:03</td><td>Polarität für Positionsmeldung</td><td>%QW</td><td>105</td><td><input checked="" type="checkbox"/></td><td>Polarität_für_Positionsmeldung_0</td></tr> <tr><td>18</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td>2052:01</td><td>Positions register P1, absolut</td><td>%QD</td><td>8</td><td><input checked="" type="checkbox"/></td><td>Positions_register_P1_absolut_0</td></tr> </tbody> </table>		Language Interface	Pdo-mapped	Index	I/O Data Name	I/O Object	Rank	IODDT Used	IODDT Field Name	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1002:00	Manufacturer Status register	%ID	0	<input checked="" type="checkbox"/>	Manufacturer_Status_register_0	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:01	Position	%QD	0	<input checked="" type="checkbox"/>	Position_0	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:02	Speed	%QW	94	<input checked="" type="checkbox"/>	Speed_0	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:03	Motion Task Type	%QW	95	<input checked="" type="checkbox"/>	Motion_Task_Type_0	5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:04	Trajectory	%QD	2	<input checked="" type="checkbox"/>	Trajectory_0	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2022:05	Motion Task Number	%QW	96	<input checked="" type="checkbox"/>	Motion_Task_Number_0	7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:06	Acceleration Time	%QW	97	<input checked="" type="checkbox"/>	Acceleration_Time_0	8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:07	Braking Time	%QW	98	<input checked="" type="checkbox"/>	Braking_Time_0	9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:08	Rate of Change-limiting Acceleration	%QW	99	<input checked="" type="checkbox"/>	Rate_of_Change-limiting_Acceleration	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:09	Rate of Change-limiting Deceleration	%QW	100	<input checked="" type="checkbox"/>	Rate_of_Change-limiting_Deceleration	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0A	Number of the following motion task	%QW	101	<input checked="" type="checkbox"/>	Number_of_the_following_motion_Task	12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0B	Start delay for following task	%QW	102	<input checked="" type="checkbox"/>	Start_delay_for_following_task_0	13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0D	Weighting Factor Speed for PDO	%QD	4	<input checked="" type="checkbox"/>	Weighting_Factor_Speed_for_PDO_0	14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0E	Speed for direct Motion Task	%QD	6	<input checked="" type="checkbox"/>	Speed_for_direct_Motion_Task_0	15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2051:01	Überwachung ein- / ausschalten	%QW	103	<input checked="" type="checkbox"/>	Überwachung_ein_ausschalten_0	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2051:02	Meldeart: umlaufend / einmalig	%QW	104	<input checked="" type="checkbox"/>	Meldeart_umlaufend_einmalig_0	17	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2051:03	Polarität für Positionsmeldung	%QW	105	<input checked="" type="checkbox"/>	Polarität_für_Positionsmeldung_0	18	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2052:01	Positions register P1, absolut	%QD	8	<input checked="" type="checkbox"/>	Positions_register_P1_absolut_0
	Language Interface	Pdo-mapped	Index	I/O Data Name	I/O Object	Rank	IODDT Used	IODDT Field Name																																																																																																																																																																				
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1002:00	Manufacturer Status register	%ID	0	<input checked="" type="checkbox"/>	Manufacturer_Status_register_0																																																																																																																																																																				
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:01	Position	%QD	0	<input checked="" type="checkbox"/>	Position_0																																																																																																																																																																				
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:02	Speed	%QW	94	<input checked="" type="checkbox"/>	Speed_0																																																																																																																																																																				
4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:03	Motion Task Type	%QW	95	<input checked="" type="checkbox"/>	Motion_Task_Type_0																																																																																																																																																																				
5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:04	Trajectory	%QD	2	<input checked="" type="checkbox"/>	Trajectory_0																																																																																																																																																																				
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2022:05	Motion Task Number	%QW	96	<input checked="" type="checkbox"/>	Motion_Task_Number_0																																																																																																																																																																				
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:06	Acceleration Time	%QW	97	<input checked="" type="checkbox"/>	Acceleration_Time_0																																																																																																																																																																				
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:07	Braking Time	%QW	98	<input checked="" type="checkbox"/>	Braking_Time_0																																																																																																																																																																				
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:08	Rate of Change-limiting Acceleration	%QW	99	<input checked="" type="checkbox"/>	Rate_of_Change-limiting_Acceleration																																																																																																																																																																				
10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:09	Rate of Change-limiting Deceleration	%QW	100	<input checked="" type="checkbox"/>	Rate_of_Change-limiting_Deceleration																																																																																																																																																																				
11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0A	Number of the following motion task	%QW	101	<input checked="" type="checkbox"/>	Number_of_the_following_motion_Task																																																																																																																																																																				
12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0B	Start delay for following task	%QW	102	<input checked="" type="checkbox"/>	Start_delay_for_following_task_0																																																																																																																																																																				
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0D	Weighting Factor Speed for PDO	%QD	4	<input checked="" type="checkbox"/>	Weighting_Factor_Speed_for_PDO_0																																																																																																																																																																				
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2022:0E	Speed for direct Motion Task	%QD	6	<input checked="" type="checkbox"/>	Speed_for_direct_Motion_Task_0																																																																																																																																																																				
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2051:01	Überwachung ein- / ausschalten	%QW	103	<input checked="" type="checkbox"/>	Überwachung_ein_ausschalten_0																																																																																																																																																																				
16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2051:02	Meldeart: umlaufend / einmalig	%QW	104	<input checked="" type="checkbox"/>	Meldeart_umlaufend_einmalig_0																																																																																																																																																																				
17	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2051:03	Polarität für Positionsmeldung	%QW	105	<input checked="" type="checkbox"/>	Polarität_für_Positionsmeldung_0																																																																																																																																																																				
18	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2052:01	Positions register P1, absolut	%QD	8	<input checked="" type="checkbox"/>	Positions_register_P1_absolut_0																																																																																																																																																																				
6	<p>用户可以更改设备配置文件屏幕上的一些参数：</p> <ul style="list-style-type: none"> ● Family：从列表中选择设备系列。 ● Image File：对要在 Catalog Manager 配置编辑器中显示的设备，选择与其关联的图像文件 (BMP 或 JPG 格式)。图像的大小没有限制。 ● Local Configuration Possible：如果必须在本地用外部工具配置设备，请选择此选项 (PLC 不会向设备发送配置参数)。 ● Expert Mode：专用模式配置参数 (见配置参数页 121)。 <p>配置功能参数、基本配置参数 (见基本配置参数页 117) 和专用模式配置参数 (见配置参数页 121)。对所需的所有配置参数进行设置后，单击“确定”。</p>																																																																																																																																																																											
7	<p>返回到 Catalog Manager 主屏幕，单击按钮 'Build Catalog' 保存任何改动，并且将新设备加载到 Catalog Manager 数据库中。</p> <p>会出现一个窗口，显示目录生成的进度。</p> <p>生成完成后，设备将用  图标表示。</p>																																																																																																																																																																											

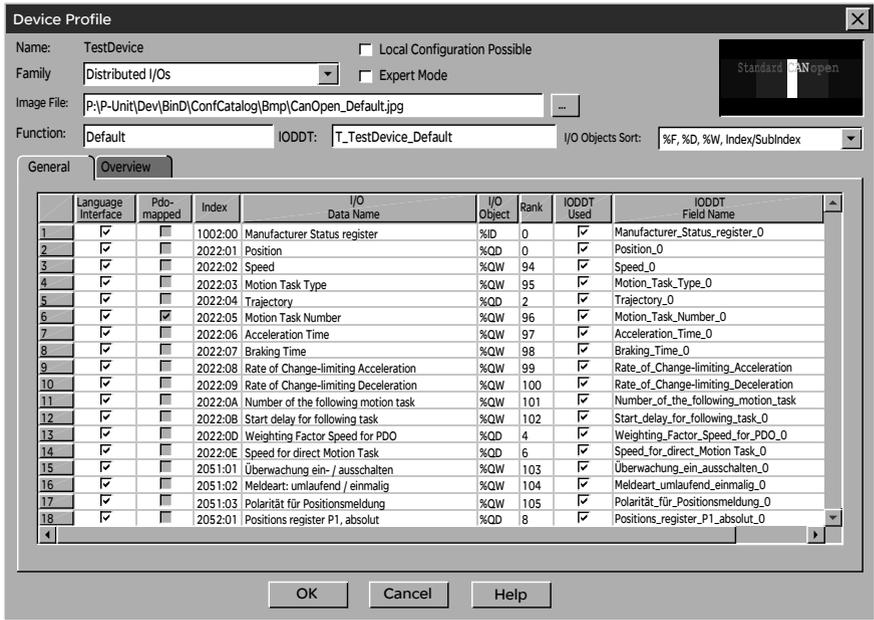
如何在设备上添加功能

概览

本过程描述了如何将功能添加到 Catalog Manager 中的现有设备上。在设备上添加功能可以让用户访问设备上有限的功能子集。

过程

下表显示了添加功能和访问配置屏幕的过程：

步骤	操作
1	打开 Catalog Manager 。
2	访问配置屏幕有以下几种不同的方式： <ul style="list-style-type: none"> ● 右键单击设备。 选择 "Add Function" 或者 ● 选择 Catalog Manager 中的某个设备。 选择 Edit->Add Function
3	<p>出现下面的窗口，显示基本的配置参数：</p>  <p>配置功能参数、基本配置参数（见基本配置参数页 117）、专用模式配置参数（见配置参数页 121）。</p>
4	配置完毕后，单击“确定”。 返回到 Catalog Manager 主屏幕，单击按钮 'Build Catalog' 保存任何改动，并且将新功能加载到 Catalog Manager 数据库中。

基本配置参数

概览

基本配置参数可以在 Device Profile 屏幕中设置，方法是使用 General 和 Overview 选项卡。

功能参数

用户可以自定义已添加到 Catalog Manager 中的功能，以便选择要求的特定能力。

开始配置之前，请注意以下规则：

- 一个设备中每个功能的名称必须是唯一的。缺省功能的名称是“Default”，但用户可以修改它。
- 如果检查到 IODDT 中至少使用了网格中的一个变量，则会为该功能生成一个 IODDT。
- IODDT 名称在整个目录中必须是唯一的。在缺省情况下，它是由设备名称和功能名称组成的，名称的开头为“T_”。用户可以修改 IODDT 名称，但新名称必须是唯一的。
- 不得重复使用预定义的 IODDT 或用户定义的 IODDT。但是，可以通过使用新名称来复制它们。

使用下拉式菜单可以按 I/O 对象将配置参数排序。下面是可用的排序模式：

- 对象排序 1 - 类型和索引 / 子索引：
首先根据类型 (%F、%D、%W)，然后根据每种类型中的索引和子索引将 I/O 排序。
- 对象排序 2 - PDO 顺序：
首先根据类型 (%F、%D、%W)，然后根据每种类型中未映射对象的 PDO 顺序将 I/O 排序。
- 对象排序 3 - 索引 / 子索引：
根据索引和子索引对 I/O 排序。

下图显示了上述三种排序类型的 I/O 对象分配示例：

Objects Dictionary	Type	PDO Mapping	I/O Object	Sort 1	I/O Object	Sort 2	I/O Object	Sort 3
0x2004:06	%IW	Tx1.1	0x3000:04	%IF 0	0x2004:06	%IW 0	0x2004:06	%IW 0
0x2004:07	%IW	Tx1.2	0x3000:03	%ID 4	0x2004:07	%IW 1	0x2004:07	%IW 1
0x2008:01	%QW	Rx1.1	0x2004:06	%IW 6	0x3000:04	%IF 2	0x3000:03	%ID 2
0x2008:05	%QW	Rx1.2	0x2004:07	%IW 7	0x3000:03	%ID 6	0x3000:04	%IF 4
0x3000:02	%QF	Rx4.1	0x3000:02	%QF 0	0x2008:01	%QW 0	0x2008:01	%QW 0
0x3000:01	%QD	Rx4.2	0x3000:01	%QD 4	0x2008:05	%QW 1	0x2008:05	%QW 1
0x3000:04	%IF	Tx4.1	0x2008:01	%QW 6	0x3000:02	%QF 2	0x3000:01	%QD 2
0x3000:03	%ID	Tx4.2	0x2008:05	%QW 7	0x3000:01	%QD 6	0x3000:02	%QF 4

注意：

Tx 1.2: 对象被映射到传输 PDO 编号 1，位置 2。

Rx 1.2: 对象被映射到接收 PDO 编号 1，位置 2。

General 选项卡 General 选项卡显示了 EDS 文件中所有可映射的变量。

General		Overview							
Language Interface	Pdo-mapped	Index	I/O Data Name	I/O Object	Rank	IODDT Used	IODDT Field Name		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2000:01	Parameter Input/Diagnostic at Pin 2	%QW	0	<input checked="" type="checkbox"/>	Parameter_Input/Diagnostic_at_Pin_2_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2001:01	Parameter Input/Output at Pin 4	%QW	1	<input checked="" type="checkbox"/>	Parameter_Input/Output_at_Pin_4_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:01	Common diagnosis	%IW	0	<input checked="" type="checkbox"/>	Common_diagnosis_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:02	Sensor short circuit	%IW	1	<input checked="" type="checkbox"/>	Sensor_short_circuit_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:03	Actuator shutdown Pin 4	%IW	2	<input checked="" type="checkbox"/>	Actuator_shutdown_Pin_4_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:05	Actuator warning Pin 4	%IW	3	<input checked="" type="checkbox"/>	Actuator_warning_Pin_4_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:07	Desina inputs	%IW	4	<input checked="" type="checkbox"/>	Desina_inputs_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6000:01	Digital Input 8 Bits Pin4	%IW	5	<input checked="" type="checkbox"/>	Digital_Input_8_Bits_Pin4_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6000:02	Digital Input 8 Bits Pin2	%IW	6	<input checked="" type="checkbox"/>	Digital_Input_8_Bits_Pin2_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6100:01	Digital Input 16 Bits	%IW	7	<input checked="" type="checkbox"/>	Digital_Input_16_Bits_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6102:01	Polarity 16 Inputs	%QW	2	<input checked="" type="checkbox"/>	Polarity_16_Inputs_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6200:01	Write Outputs 1 to 8	%QW	3	<input checked="" type="checkbox"/>	Write_Outputs_1_to_8_0		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6300:01	Digital Output 16 Bits	%QW	4	<input checked="" type="checkbox"/>	Digital_Output_16_Bits_0		

会列出每个变量的下列信息：

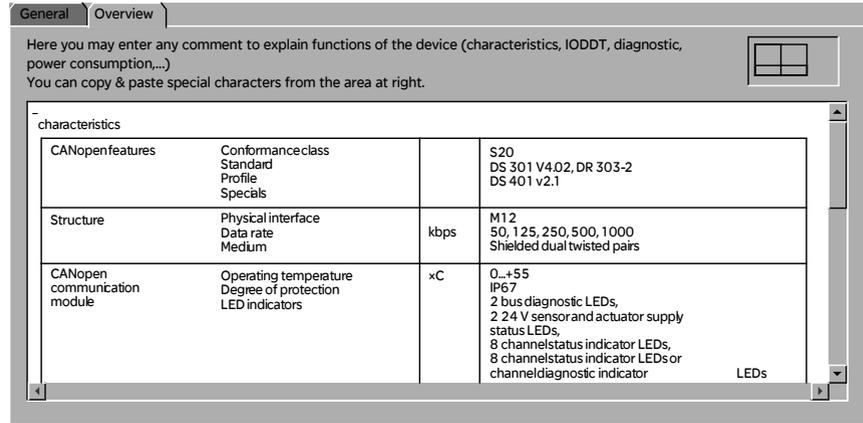
名称	说明
行号	<p>如果用户需要更多信息，可以右键单击“行号”列：</p> <ul style="list-style-type: none"> ● Set as parameter：将变量转换为参数。 ● Set Bit Mapping：打开对话框，定义 bit mapping。 bit mapping 对 %I 或 %Q 拓扑变量上映射的 CANopen 对象创建一个布尔视图。 <div data-bbox="514 889 916 1084" data-label="Image"> </div> <ul style="list-style-type: none"> ● Reset Bit Mapping：复位变量的 bit mapping。
Language Interface	<p>如果选中该复选框，变量将拥有一个语言界面。这样，该变量就可以在程序中使用，并且它的值将显示在调试屏幕中。如果不选中该复选框，变量将不可用。</p> <p>您可以通过取消选中用户不需要的变量来最小化功能所需的存储器量。在缺省情况下，所有变量都是选中的。</p> <p>用户可以通过右键单击来选中或取消选中列中的所有变量。所有“语言界面”变量都具有在目录中定义的静态语言界面。</p>

名称	说明
PDO Mapped (不可修改)	指示变量当前是否已映射到 PDO。它可以在专用模式下更改（请参阅专用模式配置参数）。 缺省功能使用 EDS 文件中定义的 PDO mapping。 'PDO Mapped' 变量的复选框不能由用户直接修改，如果用户在“语言界面”列表中取消选中某个变量，那么 PDO-Mapped' 复选框就得到刷新，并且成为未选中状态。
Index (不可修改)	指示 CANopen 索引参数。
I/O Data Name (不可修改)	EDS 文件中的参数名称。
I/O Object	拓扑语法中的访问类型（%I、%Q）、数据类型（W、D、F）和 I/O 数据的序号。序号由语言界面变量的列表计算而得，并在每次语言界面变量的列表更改时更新。如果 "I/O Object" 复选框未选中，那么 "Language Interface" 字段则为空。
IODDT Used	如果选中，则该数据将成为相关 IODDT 的字段。只有在 "Language interface" 复选框选中时此复选框才可修改。否则它处于未选中状态。 用户可以通过右键单击来选择或取消选择列中的所有变量。
IODDT Field Name	IODDT 字段的名称。在缺省情况下，这是 I/O 数据的名称，其中的空格和特殊字符由下划线代替。如果名称在 IODDT 中不唯一，则会在名称中添加数字。显示此名称，并且只有在 "IODDT Used" 选中时才可以由用户修改。

注意：当 "Device Function" 屏幕打开时，所有 PDO 可映射的变量都作为 "Language Interface" 选中，并用于 IODDT。只有在 EDS 文件中缺省映射的变量才作为 "PDO-mapped" 选中。

Overview 选项卡 在这个屏幕中，用户可以输入设备和相关功能的文字描述。它将显示在 Unity 模块编辑器中。

下图显示了一个 overview 选项卡示例：



General Overview

Here you may enter any comment to explain functions of the device (characteristics, IODDT, diagnostic, power consumption,...)
You can copy & paste special characters from the area at right.

characteristics			
CANopenfeatures	Conformanceclass Standard Profile Specials		S20 DS 301 V4.02, DR 303-2 DS 401 v2.1
Structure	Physical interface Data rate Medium	kbps	M12 50, 125, 250, 500, 1000 Shielded dual twisted pairs
CANopen communication module	Operating temperature Degree of protection LED indicators	°C	0...+55 IP67 2 bus diagnostic LEDs, 2 24 V sensor and actuator supply status LEDs, 8 channel status indicator LEDs, 8 channel status indicator LEDs or channel diagnostic indicator LEDs

输入字段可以由用户填充。用户可通过从窗口右上方区域中复制特殊字符来绘制网格线。

Expert Mode 配置参数

概览

本部分介绍 Expert Mode 配置参数。

要获取 Expert Mode 配置参数，必须选中 Expert Mode 复选框。

Expert Mode 配置参数用于：

- 删除变量
- 更改变量映射
- 修改变量参数
- 以只读模式显示用于生成设备的 EDS 的内容

Expert Mode 配置参数由三个选项卡组成：

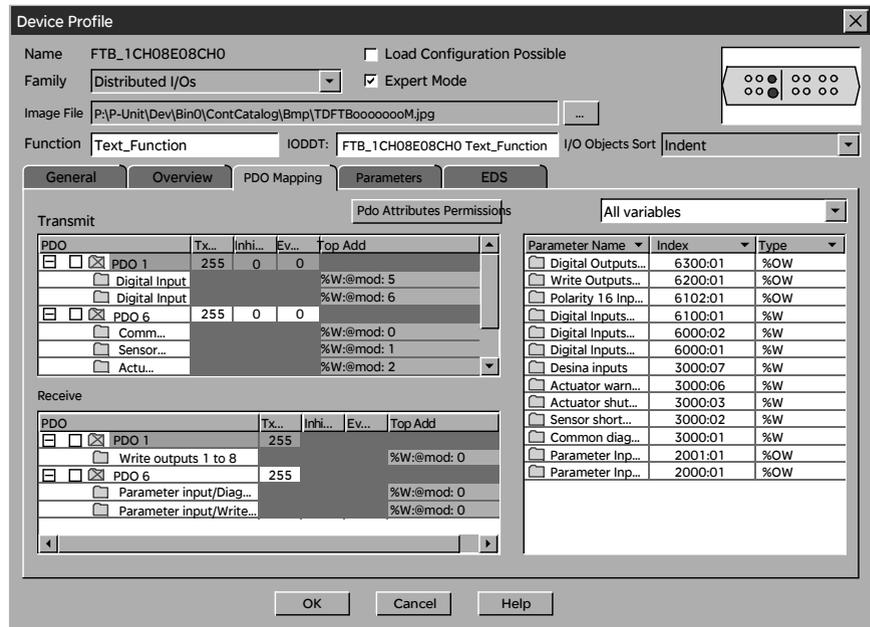
- PDO mapping 选项卡
 - Parameters 选项卡
 - EDS 选项卡
-

PDO Mapping 通过 PDO mapping 选项卡可打开 CANOpen 专用屏幕进行 PDO 配置。

PDO mapping 配置由三个部分组成：

- Transmit PDOs 。
- Receive PDOs 。
- Variables 。

下图描述 PDO Mapping 选项卡：



用于传输和接收 PDO 的 Parameters 在初始化时使用 EDS 中的缺省 PDO mapping 。

用户可以更改这个缺省配置：

- 通过选中 / 取消选中来启用或禁用 PDO 。
- Variables 可以映射或取消映射，方法是使用 PDO 和 variable 窗口之间的拖放机制。
- 每个 PDO 的 Transmission type、Inhibit time、Event timer 都可以更改。如果 PDO 的属性不可用，则单元格将为深灰色且不可编辑。如果用户输入的传输值未经授权，则会显示一个消息框。

传输的隐性规则:

- 介于 241-251 之间的值不可用, 因为它们是保留值。
- 不支持值 252/253。
- 如果设备不支持同步通讯, 那么介于 0-240 之间的值就不可用。

注意: 在接收时 PDO 的 Inhibit time 和 Event timer 一直禁用。

用户不能更改 Topo.Addr 字段。

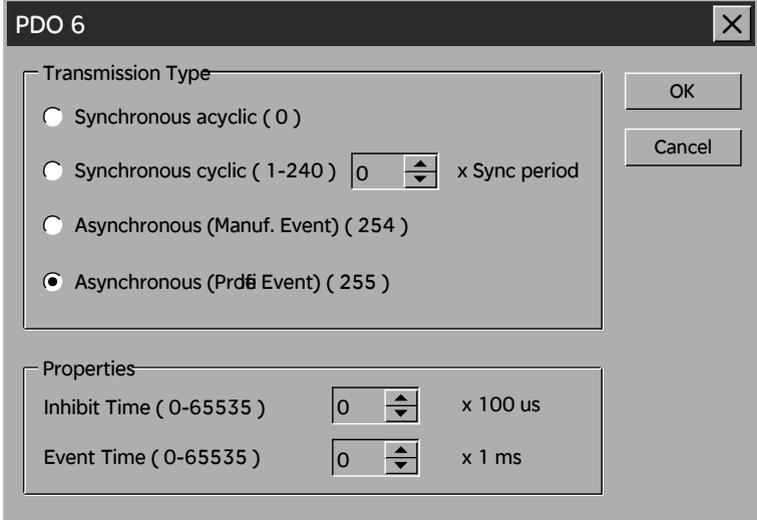
Catalog Manager 计算已映射变量的拓扑地址的固定部分:

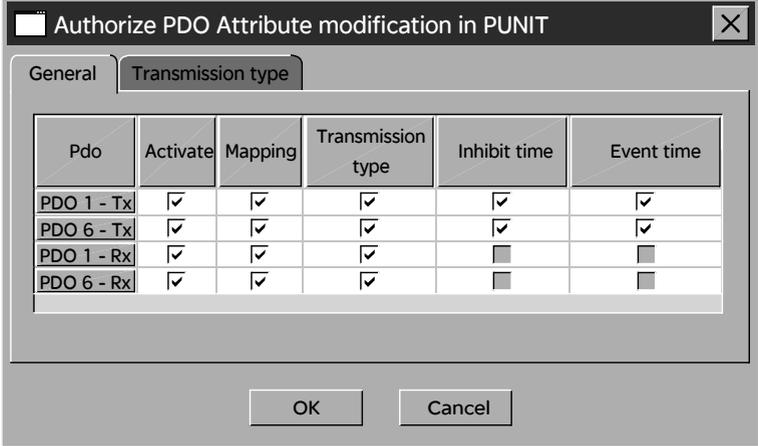
- Access type。
- Data type。
- Channel number。
- Rank。

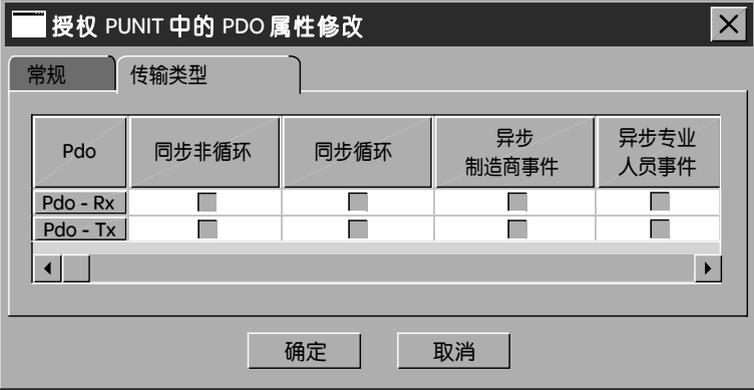
只有当设备在某个应用程序中实例化后模块地址才可知。因此, 目录专用屏幕显示的是 “<@mod>”, 而不是真正的模块地址。

注意: 即便是在 PDO 被禁用的情况下, 变量仍然可以映射到该 PDO。在这种情况下, 这些变量仍然有存储器地址 (静态存储器映射), 但是它们不会在总线上进行交换, 并且在分析应用程序时会出现警告消息。

用户可以在 PDO Mapping 选项卡上访问不同的菜单：

名称	说明
1. 拖放变量	<p>在 PDO Mapping 选项卡上拖放变量的方式：</p> <ul style="list-style-type: none"> ● 拖放到同一 PDO 中的另一个位置。 ● 从一个 PDO 拖放到另一个同样类型的 PDO (Transmit 和 Receive)。 ● 从一个 PDO 拖放到变量窗口 (取消该变量的映射)。 ● 如果访问类型与 PDO 类型兼容, 并且 PDO 所需的存储器量不超过 8 字节这个限制, 则可以从变量窗口拖放到一个 PDO。 ● 拖放到同一 PDO 中的另一个位置。 ● 拖放到同一 PDO 中的另一个位置。
2. Transmission type, Inhibit time 和 Event timer	<p>在这些列上双击可以打开下列屏幕：</p>  <p>只有在选择了某个异步传输类型时才可以访问 Event Time 和 Event Time。 ‘Transmission Type’ 的未授权的值必须根据 PDO 属性而禁用。</p>

名称	说明
3. PDO attributes permissions	<p>用于定义 PDO 的一般访问权限属性。这些属性在 Unity Pro CANopen 屏幕中和 Catalog Manager 中使用，授权修改 PDO 或取消授权。</p> <p>下图显示了常规 PDO attributes 屏幕：</p>  <p>网格用设备中所有 PDO 的列表初始化，并显示每个 PDO 的下列信息：</p> <ul style="list-style-type: none"> ● PDO: 可用 PDO (TX 和 RX) 的列表，以及它是否可修改。 ● Activate: 如果选中，则激活 PDO。因此可在 Unity Pro CANopen 屏幕中启用或禁用关联的 PDO。 ● Mapping: 如果选中，则可在 UnityPro CANopen 屏幕中编辑映射。 ● 传输类型: 如果选中，则可在 Unity Pro CANopen screen 中编辑 'Transmission type' 的值。其他约束可在 'Transmission Type' 选项卡中定义。 ● Inhibit Time: 如果选中，则可在 Unity Pro CANopen 屏幕中编辑 'Inhibit time' 的值。 ● Event time: 如果选中，则可在 Unity Pro CANopen 屏幕中编辑 'Event time' 的值。

名称	说明
4.PDO-Tx 和 PDO-Rx 的 'Transmission type' 范围	<p data-bbox="294 207 884 232">Tx PDO 和 Rx PDO 的 'Transmission type' 范围可由用户定义。</p> <p data-bbox="294 240 761 264">下图显示了 PDO 属性的 Transmission type 屏幕：</p> <div data-bbox="303 277 1057 667" style="border: 1px solid black; padding: 5px;">  </div> <p data-bbox="294 683 1012 708">表格用 Tx PDO 和 Rx PDO 的属性初始化，并且为每个 PDO 显示下列信息：</p> <ul data-bbox="294 716 1203 928" style="list-style-type: none"> ● PDO:RX PDO 和 Tx PDO 。不可修改。 ● Synchronous acyclic：如果选中，那么同步非循环传输类型范围将在 Unity Pro 中成为可用。 ● Synchronous cyclic：如果选中，那么同步循环传输类型范围将在 Unity Pro 中成为可用。 ● Asynchronous Manuf Event：如果选中，则异步制造商。事件传输类型范围将在 Unity Pro 中成为可用。 ● Asynchronous Profile Event：如果选中，那么异步配置文件事件传输类型范围将在 Unity Pro 中成为可用。

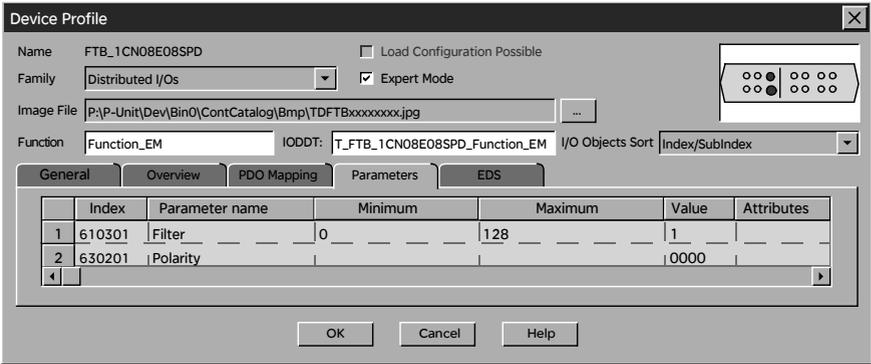
Parameters
选项卡

parameters 选项卡用于修改参数的属性。

参数的属性如下：

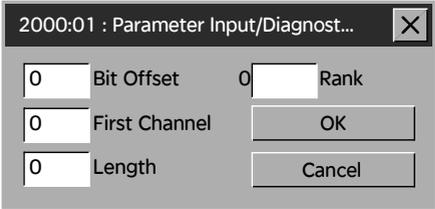
- Index：指示参数的索引和子索引（不可编辑）。
- Parameter name：指示参数的名称（不可编辑）。
- Minimum/Maximum：指示授权值的范围（不可编辑）。
- Value：指示参数值（可编辑），用 EDS 中的缺省值初始化。
- Attribute
 - Editable：参数可在 Unity Pro CANopen 屏幕中编辑（缺省值）。
 - Read only：参数在 Unity Pro 中可见，但不可编辑。
 - Hide：参数在 Unity Pro 中不可见，但值已发送到设备。

下图显示了 parameters 选项卡：



用户可以通过右键单击“行号”列来获得更多信息：

- Set as parameter：将变量转换为参数。
- Set Bit Mapping：打开对话框以定义 bit mapping。
bit mapping 为 %I 或 %Q 拓扑变量上映射的 CANopen 对象创建一个布尔视图。

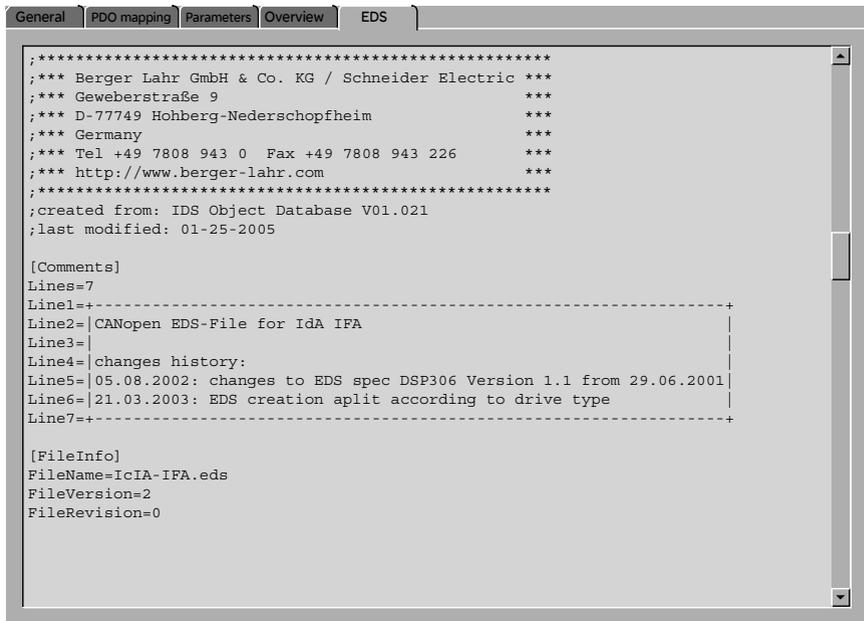


- Reset Bit Mapping：复位变量的 bit mapping。
- Move up and Move down：用来将 Unity Pro 配置屏幕中的参数排序。

EDS 选项卡

EDS 选项卡以只读模式显示用于生成设备的 EDS 的内容。

下图显示了 EDS 选项卡：



```
*****
;*** Berger Lahr GmbH & Co. KG / Schneider Electric ***
;*** Geweberstraße 9 ***
;*** D-77749 Hohberg-Nederschopfheim ***
;*** Germany ***
;*** Tel +49 7808 943 0 Fax +49 7808 943 226 ***
;*** http://www.berger-lahr.com ***
;*****
;created from: IDS Object Database V01.021
;last modified: 01-25-2005

[Comments]
Lines=7
Line1=+-----+
Line2=|CANopen EDS-File for IdA IFA
Line3=|
Line4=|changes history:
Line5=|05.08.2002: changes to EDS spec DSP306 Version 1.1 from 29.06.2001
Line6=|21.03.2003: EDS creation aplit according to drive type
Line7=+-----+

[FileInfo]
FileName=IcIA-IFA.eds
FileVersion=2
FileRevision=0
```

如何复制或删除功能

概览	功能可以在设备之间复制，或者从 Catalog Manager 数据库中删除。
Copy	<p>只有在 Catalog Manager 主屏幕中选择了设备的“功能”项时 Copy 功能才可用。</p> <p>请按照下面的指示来复制功能：</p> <ul style="list-style-type: none">● 选择 Edit->Copy。● 右键单击功能名称，然后选择 'Copy'。 <p>功能参数存储在剪贴板中，随后可以将其粘贴到兼容设备粘贴功能（见页 129）。</p>
Paste	<p>只有当某个“功能”存在于剪贴板中，并且在 Catalog Manager 主屏幕中选择了设备时 Paste 功能才可用。</p> <p>请按照下面的指示来粘贴功能：</p> <ul style="list-style-type: none">● 选择 Edit->Paste。● 右键单击设备名称，然后选择 'Paste'。 <p>无需再次指定 EDS 文件。完整的 EDS 文件已经随设备存储了。</p> <p>"Device Function" 屏幕在功能粘贴之后出现。新功能将用原功能的数据初始化。系统建议的新功能的名称与原功能相同，用户必须对其进行修改，然后才能保存该功能。</p> <p>用户可以修改和保存新功能的所有其他信息。</p>
Delete	<p>用户可以删除功能：</p> <p>请按照下面的指示来删除功能：</p> <ul style="list-style-type: none">● 从设备中选择一个功能。● 右键单击功能名称，然后选择 'Delete'。 <p>或者</p> <p>选择 Edit->Delete。</p> <ul style="list-style-type: none">● 将显示一则消息，请求您进行确认，单击“确定”继续。● 要保存改动，单击 'Build Catalog' 按钮。

如何导入 / 导出或删除一个或多个用户设备

概览

本节介绍如何在 Catalog Manager 中导入、导出或删除一个或多个用户设备。

Export User Devices

‘Export User Devices’ 启用的前提条件是：

- 如果没有作任何修改，或者如果已在目录数据库中生成所有修改。
- 目录中至少存在一个用户设备。

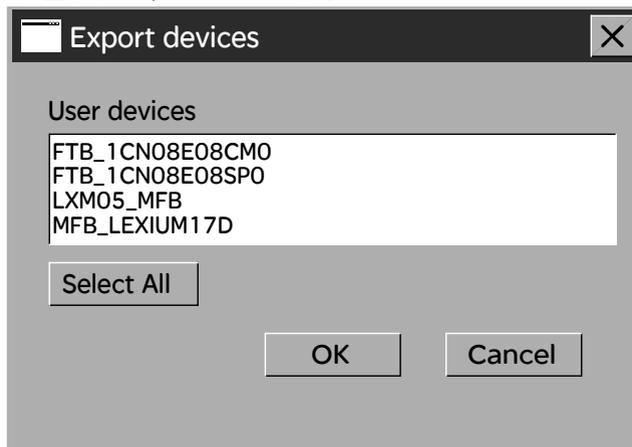
这与树控件中当前选择的项目无关。

用户必须遵循下面的指示导出用户设备：

选择 File->Export User Devices 。

屏幕上显示带用户功能的设备列表（用户设备和带用户功能的预编程设备）。列表框支持多选。通过普通的 Windows 方式（即 Ctrl-单击或 Shift-单击）便可选择要导出的设备。或者，也可以通过单击 "Select All" 按钮来选择所有设备。

下图显示了 ‘export devices’ 屏幕：



在验证此对话框时，如果选择了一个或多个设备，则会显示标准的 Windows “保存”对话框，允许用户选择文件名和导出文件的位置。

文件名的扩展名为 .cpx。

导出文件是一个 zip 文件，其中包括导出设备的所有目录源文件。

用户不能只导出一个功能，而必须导出设备中的所有用户功能。

在验证此对话框时，会从数据库中提取 *.cpx 目录源文件并加以保存。

Import User Devices

'Import User Devices' 启用的前提条件是：

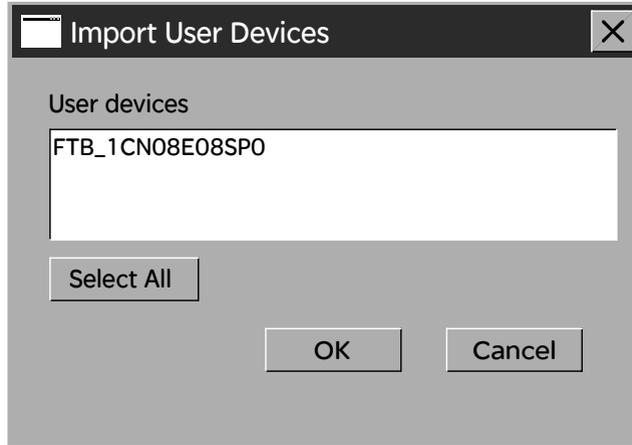
- 如果没有作任何修改，或者如果已在目录数据库中生成所有修改。
- 这与树控件中当前选择的项目无关。

用户必须遵循下面的指示导入用户设备：

选择 File->Import User Devices 。

一个标准的 Windows « 打开 » 屏幕出现。只接受 *.cpx 文件。

打开 cpx 文件时，会显示下面的模式对话框，允许用户选择 cpx 文件中的设备子集：



此对话框的工作方式与 "Export User Devices" （多选列表框）相同。

验证时，将从 *.cpx 文件中提取所有必须的目录源文件，并会生成一个新的目录数据库。

这个新数据库包括：

- 原数据库中的所有 Schneider 核心设备。
- 原数据库中的所有用户设备。
- 在此对话框中选择导入的所有用户设备。

所有用户功能必须从设备导入，不能只导入一个功能。

在出现冲突的情况下（名称相同但是 ID 不同的设备或功能已经存在于数据库中），设备将不会导入，并且出现一个消息框。

对于不需要确认的消息，将使用一个输出单。

有关可能的冲突和如何解决这些冲突的详细信息，请参阅“故障排除”一章。

在导入功能的最后，主窗口将用新数据库刷新。

Delete User
Devices

用户设备可以由用户删除。

用户必须遵循下面的指示删除用户设备：

- 从设备系列中选择用户设备。
 - 右键单击将其删除。
 - 将会出现确认消息框，单击“确定”。
 - 要保存任何改动，单击‘Build Catalog’按钮。
-

如何关闭 Catalog Manager

概览

这是关闭 Catalog Manager 的过程。

过程

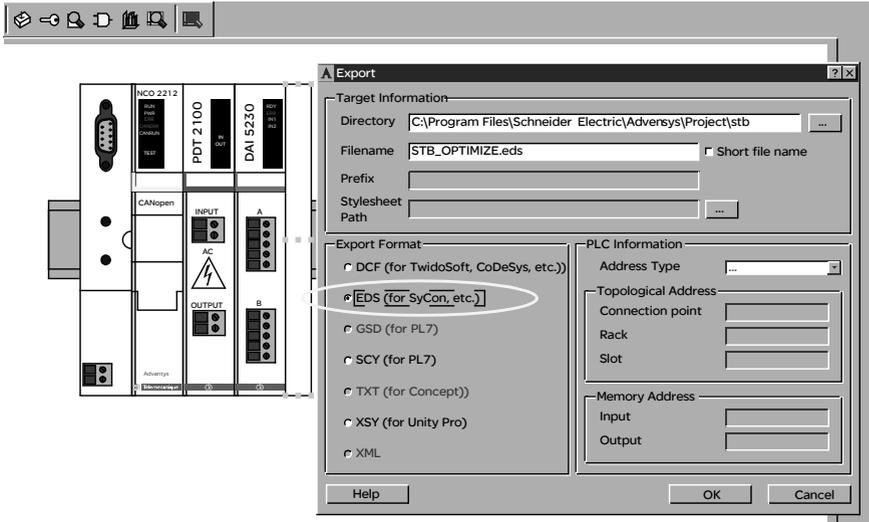
下表显示了关闭 Catalog Manager 的过程：

步骤	操作
1	检查您是否已保存了更改： <ul style="list-style-type: none">● 如果有改动，请单击 ' Build Catalog'。会现一个窗口，显示目录生成的进度。生成完毕后，继续下一步。● 如果没有任何改动，继续下一步。
2	选择 File->Exit 或者 选择主窗口上的 "Close"。

如何创建专用和优化 STB 岛的示例

概览

下面的过程描述了如何创建专用和优化的 STB 岛

步骤	操作
1	启动 Advantys 软件。
2	用需要的配置创建优化的 STB 岛。
3	<p>导出 EDS 文件。</p> <p>下图描述了“导出 EDS 文件”步骤：</p> 
4	打开 Catalog Manager 软件。
5	<p>将设备添加到 Catalog Manager。</p> <p>如何将设备添加到目录管理器（见如何将设备添加到页 113）。</p>

6.3 Catalog Manager 故障排除

概览

本节主题 本节介绍 Catalog Manager 故障排除。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
故障排除	136

故障排除

概览

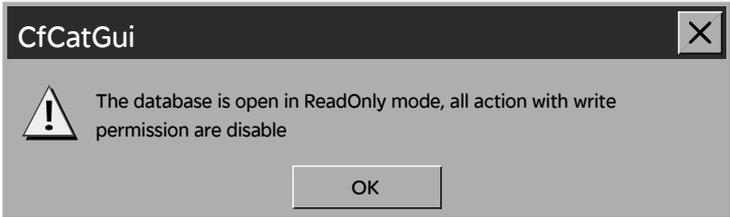
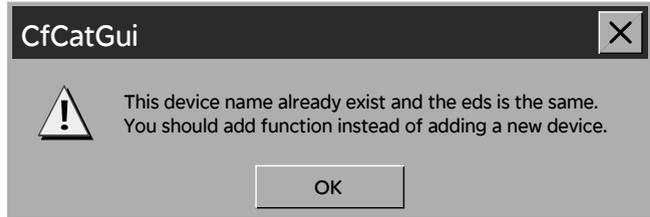
本节提供了使用目录管理器时可能会遇到的任何问题的解决方法。

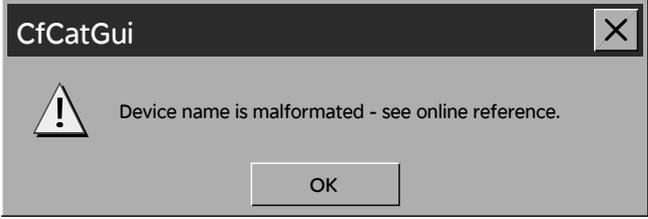
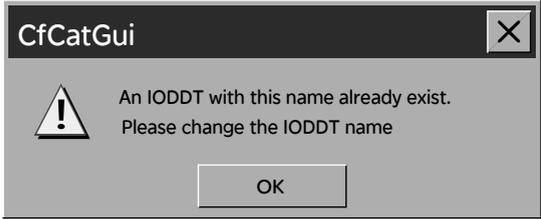
需要用户确认的消息显示在消息框中。所有其他消息显示在“输出”窗口中，并使用与消息类型相对应的文本颜色：

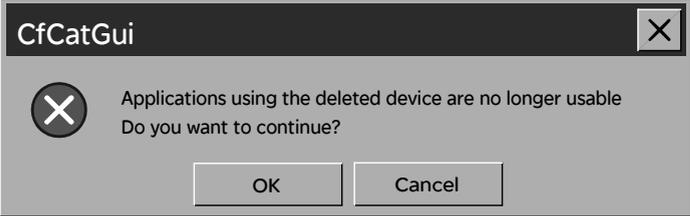
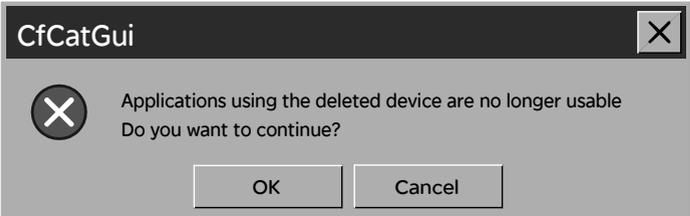
- 红色表示在分析过程中检测到错误
 - 橙色表示必须检查的重要信息
 - 蓝色表示信息型消息
-

故障排除

下表说明了目录管理器的故障排除：

问题	解决方法
<p>仅在只读模式下可用</p>	<p>如果 Unity Pro 打开，则目录管理器将在只读模式下打开。因此，将禁用修改目录管理器的所有功能。</p> <p>屏幕上将出现一条消息，通知用户有关行为的信息：</p>  <p>按照这些说明解决此问题：</p> <ul style="list-style-type: none"> ● 首先关闭 Unity Pro，然后再打开目录管理器。Unity Pro 和目录管理器不能同时启动。
<p>设备名称已存在</p>	<p>当用户希望在目录管理器数据库中添加或导入设备时，必须使用 EDS 文件。</p> <ul style="list-style-type: none"> ● 如果某一设备名称已在目录中存在，则将出现提示，询问用户是否要向该设备添加新功能。 <ul style="list-style-type: none"> ● 如果选择是，则将在主屏幕中选中现有设备并自动调用“添加功能”服务。 ● 如果选择否，则拒绝“添加设备”服务。 <p>按照这些说明解决此问题：</p> <ul style="list-style-type: none"> ● 重命名该设备。 ● 如果拒绝 EDS 文件，则消息中将显示出原因，“打开”屏幕将保持打开。用户可选择另一个 EDS 文件或取消操作。 <p>屏幕上将出现一条消息，通知用户有关行为的信息：</p> 

问题	解决方法
设备名称的格式不正确	<p>该名称的格式不正确：</p> <ul style="list-style-type: none">● 字符数大于 24 个。● 该名称不符合命名 Unity Pro 变量的规则。字符必须位于 A-Z、a-z 或 0-9 和下划线范围内。 <p>按照这些说明解决此问题：</p> <ul style="list-style-type: none">● 减少字符数并使用如上所述的有效字符。 <p>屏幕上将出现一条消息，通知用户有关行为的信息：</p> 
不支持该文件。是否要使用缺省图像？	未找到该文件或格式不受支持。仅支持 .bmp 和 .jpg 格式。
XML 文件格式不正确	<p>下列消息会出现在输出窗口中：</p> <ul style="list-style-type: none">● xml 文件的语法不正确。● xml 不一致。● 生成错误的数据库。 <p>按照这些说明解决此问题：</p> <ul style="list-style-type: none">● 如何在目录管理器中添加设备（见如何将设备添加到页 113）。● 如何在设备上添加功能（见如何在设备上添加功能页 116）。
具有此名称的 IODDT 已存在	<p>下列消息会出现在输出窗口中。按照这些说明解决此问题：</p> <ul style="list-style-type: none">● “功能名称已存在”。必须重命名该功能名称以解决此问题。● “IODDT 名称已存在”。必须重命名 IODDT 以解决此问题。 <p>显示以下消息：</p> 

问题	解决方法
文件格式错误	<p>执行“导出用户设备”操作时出现以下消息：</p> <ul style="list-style-type: none"> 文件格式错误。 <p>按照这些说明解决此问题：</p> <ul style="list-style-type: none"> 如何导入/导出或删除一个或多个用户设备(见如何导入/导出或删除一个或多个用户设备页130)。
删除设备 - 使用已删除设备的应用程序不再可用	<p>仅当树控件中所选项目的列表只包含“用户设备”项目时，才会启用此功能。当满足“删除设备”功能的所有条件时：主菜单项“编辑 / 删除”或一个或多个用户设备上某一选项的“上下文”菜单“删除”。用户需要确认此操作。该确认消息将通知有关可能后果的信息：使用已删除设备的应用程序不再可用。不能打开这些应用程序。</p> <p>显示以下消息：</p>  <p>按照这些说明解决此问题：</p> <ul style="list-style-type: none"> 如何导入/导出或删除一个或多个用户设备(见如何导入/导出或删除一个或多个用户设备页130)。
删除功能 - 使用已删除设备的应用程序不再可用	<p>此功能仅在树控件中所选项目的列表只包含“用户功能”项目时启用。无法删除设备的缺省功能。当满足“删除功能”服务的所有条件时：主菜单项“编辑 / 删除”或一个或多个用户功能的某一选项上“上下文”菜单“删除”。用户需要确认此操作。该确认消息将通知有关可能后果的信息：使用已删除功能的应用程序不再可用。不能打开这些应用程序。</p> <p>显示以下消息：</p>  <p>按照这些说明解决此问题：</p> <ul style="list-style-type: none"> 如何复制或删除功能(见如何复制或删除功能页129)。

概览

简介

本节描述 CANopen 架构的编程。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
使用 PDO 进行交换	142
使用 SDO 进行交换	147
通讯功能示例	150
Modbus 请求示例	157

使用 PDO 进行交换

概览

PDO 使用拓扑地址（%I、%IW、%Q、%QW）和内部变量（%M 或 %MW）。

传输 (%I)

PDO	Tr.Ty...	抑制 ...	偶 ...	符号	拓扑地址	%M...
<input checked="" type="checkbox"/> PDO 1(...)	255	0	0			
--> 状态 ...				lexium...	%IW\3.1\0.0.0.16	%MW16
<input checked="" type="checkbox"/> PDO 2(...)	255	0	100			
--> 状态 ...				lexium...	%IW\3.1\0.0.0.16	%MW16
--> 位置 ...				lexium...	%D\3.1\0.0.0.8	%MW8
<input checked="" type="checkbox"/> PDO 3(...)	255	0	100			
--> 状态 ...				lexium...	%IW\3.1\0.0.0.16	%MW16
--> 速度 ...				lexium...	%D\3.1\0.0.0.10	%MW10
<input checked="" type="checkbox"/> PDO 4(...)	254	0	0			

接收 (%Q)

PDO	Tr.Ty...	抑制 ...	偶 ...	符号	拓扑地址	%M...
<input checked="" type="checkbox"/> PDO 1(...)	255	0	0			
--> 状态 ...				lexium...	%QW\3.1\0.0.0.16	%MW16
<input checked="" type="checkbox"/> PDO 2(...)	255	0	100			
--> 状态 ...				lexium...	%QW\3.1\0.0.0.16	%MW16
--> 位置 ...				lexium...	%QD\3.1\0.0.0.8	%MW8
<input checked="" type="checkbox"/> PDO 3(...)	255	0	100			
--> 状态 ...				lexium...	%QW\3.1\0.0.0.16	%MW16
--> 速度 ...				lexium...	%QD\3.1\0.0.0.10	%MW10
<input checked="" type="checkbox"/> PDO 4(...)	254	0	0			

变量

参数名称	索引	...
RAMPsym	3006:01	
_IO_act	3008:01	
ANA1_act	3009:01	
ANA2_act	3009:05	
PLCopenRx1	301B:05	
PLCopenRx2	301B:06	
PLCopenTx1	301B:07	
PLCopenTx2	301B:08	
JOActivate	301B:08	
_actionStatus	301C:04	
_p_actRAMPusr	301F:02	
CUR_l_target	3020:04	
SPEEDn_target	3021:04	
PTPp_abs	3023:01	
PTPp_relpref	3023:03	
PTPp_target	3023:05	
PTPp_relpact	3023:06	
GEARdenom	3026:03	
GEARnum	3026:04	
控制字	6040:00	
状态字	6041:00	
位置实际值	6063:00	

拓扑地址

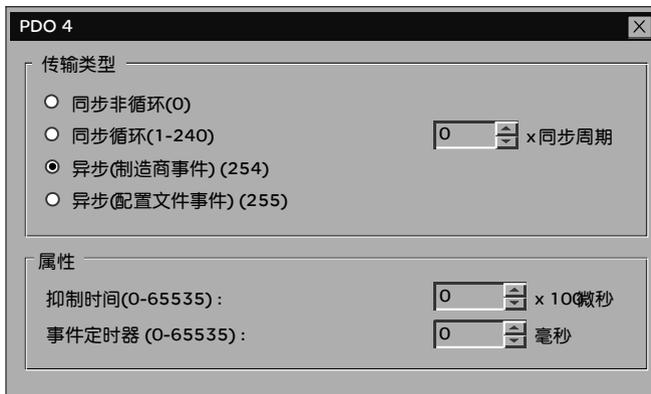
内部变量

拓扑地址与内部变量之间存在对应关系。例如，在上图中，拓扑地址 %IW\3.1\0.0.0.16 等价于 PDO 1 的 %MW16。

可启用或禁用 PDO。

根据 EDS 文件，某些 PDO 已进行了映射。

双击传输类型列，系统显示下面的窗口：



此窗口可用于进行以下配置：

- 传输类型：
 - 同步非循环：传输类型 0 表示应将消息与 SYNC 消息同步传输，但不根据值进行周期性传输。
 - 同步循环：介于 1 和 240 之间的值表示以同步和循环的方式传输 PDO，传输类型值指示两个 PDO 传输间的 SYNC 消息数。
 - 异步 PDO：传输类型 254 表示以异步方式传输 PDO。这种传输方法完全取决于设备中的实施。它主要用于数字 I/O。
 - 同步 PDO：传输类型 255 表示 PDO 在值发生变化时以异步方式传输。验证选定的设备是否支持所配置的传输类型。
- 抑制时间：在此时间段屏蔽通讯。
- 事件定时器：（为启动 PDO 而管理事件所需的时间）。

注意：只能使用 Unity Pro 对 PDO 进行配置。

拓扑地址的结构 CANopen 总线从站的输入 / 输出对象的拓扑地址的结构如下：

% I, Q X, W, D, F \ b.e \ r . m . c . d

系列	元素	值	含义
符号	%	-	指示 IEC 对象。
对象类型	I	-	输入对象。
	Q	-	输出对象。
格式（大小）	X	8 位 (Ebool)	Ebool 类型布尔值（非强制）。
	W	16 位	16 位 WORD 类型的字。
	D	32 位	32 位 DINT 类型的字。
	F	32 位	32 位 REAL 类型的字。
模块 / 通道地址和连接点	b	3 到 999	总线编号。
	e	1 到 63	连接点编号（CANopen 从站号）。
机架号	r	0	虚拟机架号，始终为 0。
模块编号	米	0	虚拟模块编号，始终为 0。
通道编号	c	除 FTB 外的所有设备，都等于 0（通道从 0 到 7 编号，然后从 10 到 17 编号）。	通道编号。
数据在通道中的序号	d	0...999	从站的数据数字。 此数字可以在 0 到 999 之间，因为从站最多只能有 1000 个输入和输出字。

拓扑寻址示例

与 CANopen 的 3 号总线的点 4 连接的某个项的拓扑寻址示例：

带布尔显示的数字量 /TOR 自主模块	
%I\3.4\0.0.5	布尔值输入至通道 5 (忽略了梯级 0)。
数字量标准模块	
%IW\3.4\0.0.0.2.5	布尔值输入至唯一的通道 0, 序号 2, 位 5。 导入 DCF 文件时, 会给出映射。
Advantys STB 岛上的数字量模块	
%IW\3.4\0.0.0.3.2	字 3, 位 2, 数据由 Advantys 配置软件提供。

编号开始于：

- 0 (对于通道)
- 0 (对于序号)

注意：虚拟对象 (机架、模块) 的序号编号总是为 0。

CANopen 数字量输入/输出的对象寻址遵循的规则与机架上的数字量输入/输出的对象寻址相同：字、双字和浮点字位于同一个块中。

示例：位于 CANopen 的 3 号总线 0 通道的连接点 4 的设备，具有：

数据类型	拓扑地址：
2 个输入字	%IW \3.4\0.0.0.0 或 %IW \3.4\0.0.0.1
1 个双字输入	%ID \3.4\0.0.0.2
1 个浮点输入	%IF \3.4\0.0.0.4
1 个输出字	%QW\3.4\0.0.0.6

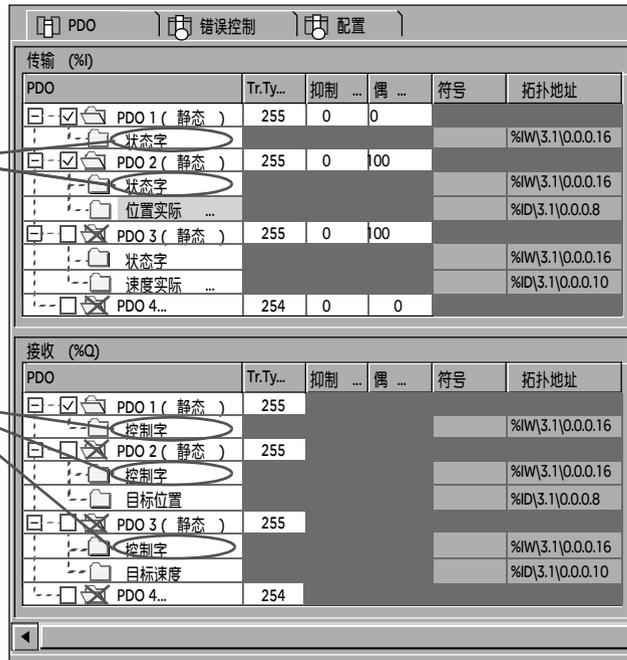
每个对象在 PDO 中只能映射一次。如果同一个对象在相同的 PDO 中映射多次，Unity Pro 将显示错误消息。

如果存在多个具有相同映射对象的 PDO，则只能启用一个 PDO。如果同时启用多个具有相同映射对象的 PDO，则在重新生成应用程序时，Unity Pro 将显示一条错误消息。

Lexium 05 示例：

错误：同一对象映射到两个启用的 PDO 中

无错误：仅启用一个 PDO。



使用 SDO 进行交换

概览

CANopen 总线上的消息的显式交换是通过读 / 写协议完成的。

有 3 种方式可以访问 SDO :

- 使用通讯功能 READ_VAR 和 WRITE_VAR
- 使用 Unity Pro 调试屏幕
- 使用请求 ModBus FC43/0xD



意外的设备操作

在修改变量时，请在特定目标 CANopen 设备的文档中查看 SDO 命令的结果。

如果不遵守这个警告将会导致 死亡，严重伤害，或设备损坏。

通讯功能

使用通讯功能 READ_VAR 和 WRITE_VAR 可以访问 SDO 。

注意：最多可同时发送 16 个 READ_VAR/WRITE_VAR。轮询任务在每个任务循环中每 5 毫秒运行一次，以便检查交换的终点。这对在一个任务循环中运行许多 SDO 的用户是十分有用的。

有关如何使用通讯功能的详细信息，请参见[通讯功能示例](#)。

注意：通过写入 SDO 更改设备的输出对于 %QW 没有任何影响。

Unity Pro

使用 SDO 对象可以访问变量。

在在线模式下，通过 CANopen 屏幕（请参见从站诊断）可以访问：

- 处于读 / 写模式下的各种设备对象（只能通过列表框）
- 变量的描述
- 重复通讯

CANopen 屏幕如下所示：



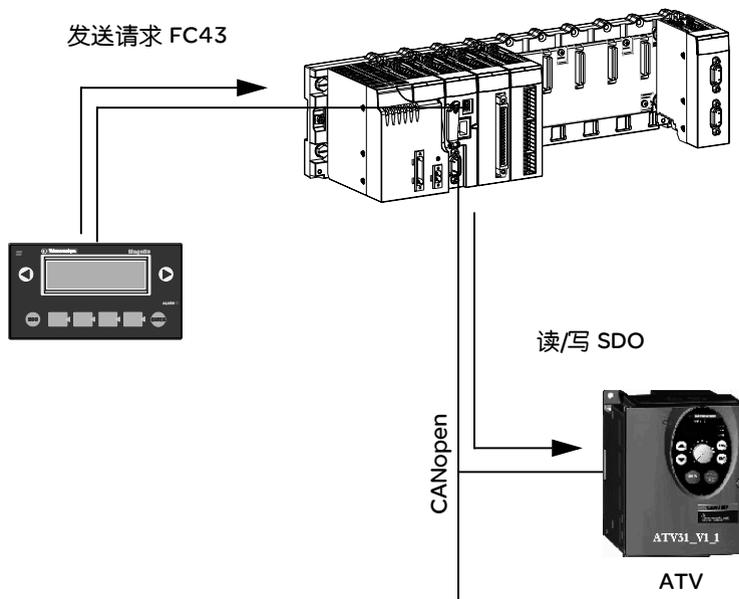
值的显示方式如下：



以字节格式显示 SDO 信息（读取或写入）。对于 16 位或 32 位信息，首先显示低字节（例如：十六进制 0102 将显示为 02 01）。

Modbus 请求

从人机界面（例如：XBT），可以使用 Modbus FC43 请求访问 SDO。
BMX P34 2010/2030



有关如何使用 Modbus 请求 FC43/0xD 的详细信息，请参见 Modbus 请求示例。

SDO 超时

实施了各种超时。超时取决于对象的类型和访问的类型（读 / 写）：

对象	超时
1010h	15 秒
1011h	3 秒
2000h到 6000h	8 秒
所有其他对象	
- SDO 读取	1 秒
- SDO 写入	2 秒

通讯功能示例

概览

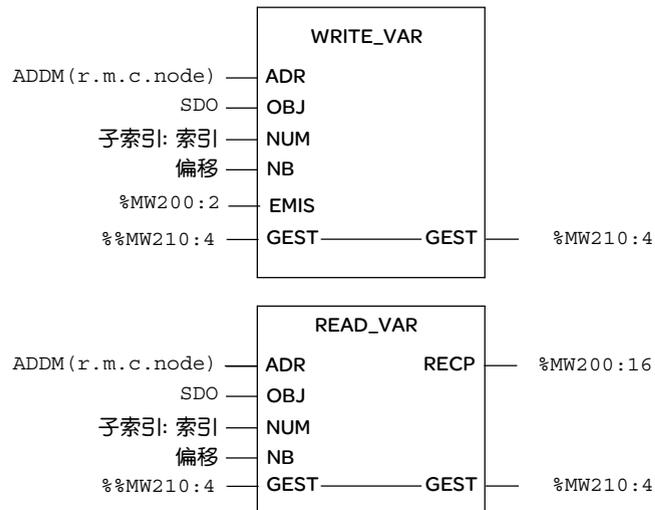
使用通讯功能 `READ_VAR` 和 `WRITE_VAR` 可访问 SDO。

有 3 种可能的表示形式：

- FBD 表示形式
- 梯形图表示形式
- IL 表示形式

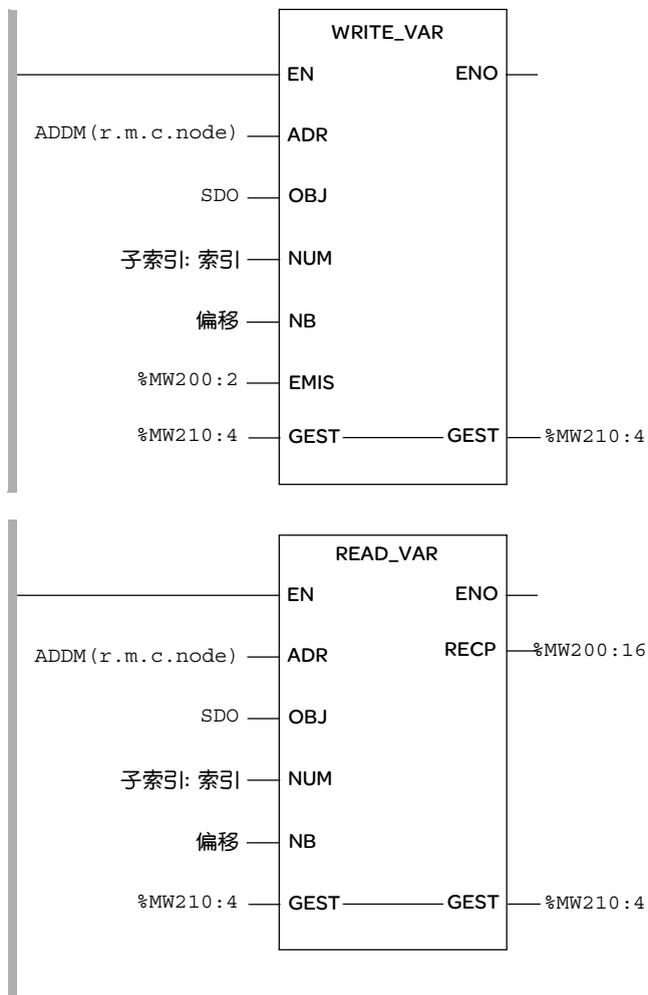
FBD 表示形式

通讯功能的 FBD 表示形式如下所示：



梯形图表示形式

通讯功能的梯形图表示形式如下所示：



IL 表示形式

通讯功能的语法如下:

```
ADDM(  
    IN := ' 0.0.2.2'  
)  
ST %MW2100:8  
  
LD 50  
ST %MW2182 (* 超时 5 秒钟 *)  
LD 2  
ST %MW2183 (* 长度 *)  
  
(* 读取 "Vendor ID" 对象, 从站 @2, CANopen 网络 *)  
READ_VAR (  
    ADR := %MW2100:8,  
    OBJ := 'SDO',  
    NUM := 16#00011018,  
    NB := 0,  
    GEST := %MW2120:4,  
    RECP := %MW2110:4  
)  
  
(* 写入值 16#FFFF, 从站 @2 输出, CANopen 网络 *)  
LD 16#ffff  
ST %MW2200  
  
WRITE_VAR (  
    ADR := %MW2100:8  
    OBJ := 'SDO',  
    NUM := 16#00016300,  
    NB := 0,  
    EMIS := %MW2200:1,  
    GEST := %MW2180:4  
)
```

注意: `offset` 参数必须设置为 0。

注意: `subindex : index` 参数按简单字编码 (`subindex` 是高字节)。

WRITE_VAR

功能的参数描述

下表列出了 WRITE_VAR 功能的各个参数:

参数	说明
ADDM('r.m.c.node')	交换操作的目标实体的地址: <ul style="list-style-type: none"> ● r: 处理器机架号 ● m: 机架中的处理器插槽 (0) ● c: 通道 (CANopen 只使用通道 2) ● node: CANopen 总线上的传输设备的标识符
'SDO'	SDO 对象类型。
subindex:index	标识 CANopen SDO 索引或子索引的双字或立即值: 组成双字的最高有效字包含子索引, 最低有效字则包含索引。 示例: 如果使用双字 subindex:index: <ul style="list-style-type: none"> ● 最高有效 16 位包含子索引 ● 最低有效 16 位包含索引
EMIS	包含要发送的 SDO 数据的字表 (%MW200:2)。 WRITE_VAR 功能的接收缓冲区必须大于 SDO。设备文档中有 SDO 的长度信息。
GEST	带 4 个输入的字表 (%MW210:4)。

READ_VAR 功能的参数描述 下表列出了 READ_VAR 功能的各个参数：

参数	说明
ADDM('r.m.c.node')	交换操作的目标实体的地址： <ul style="list-style-type: none"> ● r: 处理器机架号 ● m: 机架中的处理器插槽 (0) ● c: 通道 (CANopen 只使用通道 2) ● node: 总线上的目标设备的标识符
'SDO'	SDO 对象类型。
subindex:index	标识 CANopen SDO 索引或子索引的双字或立即值： 组成双字的最高有效字包含子索引，最低有效字则包含索引。 示例： 如果使用双字 subindex:index： <ul style="list-style-type: none"> ● 最高有效 16 位包含子索引 ● 最低有效 16 位包含索引
GEST	带 4 个输入的字表 (%MW210:4)。
RECP	至少带一个用于接收 SDO 数据的输入的字表 (%MW200:16)。 READ_VAR 功能的接收缓冲区必须大于 SDO。设备文档中有 SDO 的长度信息。

控制块字描述

下表描述控制块的各个字：

字段	字	类型	说明
控制字节	0 (最低有效字节)	BYTE	位 0 = 活动位 位 1 = 取消位
交换 ID	0 (最高有效字节)	BYTE	单个数字，交换的标识符。
ComState	1 (最低有效字节)	BYTE	0x00 = 交换完成 0x01 = 超时 0x02 = 用户取消 0x03 = 地址格式不正确 0x04 = 目标地址不正确 0x06 = Com Fb 参数不正确 0x07 = 一般传输问题 0x09 = 接收的缓冲区太小 0x0B = 无系统资源 0xFF = 网络交换错误
ExchState	1 (最高有效字节)	BYTE	如果 ComState = 0x00: 0x00: 请求已处理 0x01: 无法处理 0x02: 响应不正确 如果 ComState = 0xFF 0x07: 一般交换错误 0x0B: 目标设备没有更多资源。 0x0D: 无法访问设备。 0x2B: SDO 交换错误
超时	2	WORD	超时值 (x 100 毫秒)
长度	3	WORD	长度 (以字节为单位)

ST 语言示例

```
(* 读取节点 5 SDO, 索引 1018, 子索引 3 *)
if (%M400) then
    subindex_index := 16#00031018 ;
    %MW1052 := 50; (* timeout 5 secondes *)

    READ_VAR(ADDM('0.0.2.5'),'SDO',subindex_index,0,%MW1050:4,%M
W1100:2);

    %M400:= 0;
end_if;

(* Write the node 31 SDO, index 203C, subindex 2 *)
if (%M401) then
    subindex_index := 16#0002203C;
    %MW1152 := 50; (* timeout 5 secondes *)
    %MW1153 := 2; (* length 2 bytes *)
    %MW1200 := 16#03E8; (* value of object *)
    WRITE_VAR(ADDM('0.0.2.31'),'SDO',subindex_index,0,
%MW1200:1,%MW1150:4);

    %M401:= 0;
end_if;
```

Modbus 请求示例

概览

从人机界面（例如：XBT），可以使用 Modbus FC43 请求访问 SDO

SDO 读取示例

节点读取 1F，对象 1005，子索引 00，长度 8 个字节

FC	MEI	Prot	Nid	索引	子索引	偏移	长度
2B	0D	00	1F	10 05	00	00 00	00 08

响应正确：接收 4 个字节

FC	MEI	Prot	Nid	索引	子索引	偏移	长度	对象值
2B	0D	00	1F	10 05	00	00 00	00 04	80 00 00 00

故障：SDO 取消代码

FC	MEC	扩展长度	MEI	例外代码	SDO 中止代码
AB	FF	00 06	0D	EC	06 02 00 00

写入 SDO 示例

节点读取 1F，对象 203C，子索引 02，长度 2 个字节 03 E8

FC	MEI	Prot	Nid	索引	子索引	偏移	长度	数据
2B	0D	01	1F	20 C3	02	00 00	00 02	03 E8

响应正确：接收 4 个字节

FC	MEI	Prot	Nid	索引	子索引	偏移	长度
2B	0D	00	1F	20 3C	02	00 00	00 00

故障：SDO 取消代码

FC	MEC	扩展长度	MEI	例外代码	SDO 中止代码
AB	FF	00 06	0D	EC	06 02 00 00

在 CANopen 总线上调试通讯

8

概览

本章目标

本章介绍 CANopen 总线主站和从站的调试。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
如何访问远程设备的调试屏幕	160
CANopen 主站的调试屏幕	161
从站调试屏幕	163

如何访问远程设备的调试屏幕

概览

下面的操作介绍如何访问 CANopen 网络元素的不同调试屏幕。

注意：只能在在线模式下访问调试屏幕。

主站调试屏幕

要访问主站调试屏幕，请执行以下操作：

步骤	操作
1	连接到管理器 PLC。
2	访问 CANopen 主站配置屏幕（见如何访问 CANopen 主站配置屏幕页 95）。
3	选择 调试 选项卡。

从站调试屏幕

要访问从站调试屏幕，请执行以下操作：

步骤	操作
1	连接到管理器 PLC。
2	访问 CANopen 从站配置屏幕（见使用 Unity 进行配置页 83）。
3	选择 调试 选项卡。

CANopen 主站的调试屏幕

概览

此屏幕只能在在线模式下使用。

示意图

下图显示了主站调试屏幕：



元素和功能

下表对构成主站调试屏幕的各个区域进行了描述：

读取	编号	通道
1	选项卡	前景中的选项卡指示显示的屏幕类型。这里显示的是调试屏幕。
2	模块	此区域由配备有 CANopen 端口的模块的缩写标题和指示模块状态的 3 个 LED 组成。
3	通道	<p>此区域允许您选择要调试的通讯通道。</p> <p>单击设备可显示以下选项卡：</p> <ul style="list-style-type: none"> ● 描述：给出内置 CANopen 端口的特性 ● 输入 / 输出对象：允许预先用符号表示输入 / 输出对象 <p>单击通道可显示以下选项卡：</p> <ul style="list-style-type: none"> ● 配置：允许您声明和配置 CANopen 主站 ● 调试：只能在在线模式下访问 ● 故障：只能在在线模式下访问 <p>此区域还包含一个指示通道状态的 LED。</p>
4	常规参数	<p>此区域用于查看：</p> <ul style="list-style-type: none"> ● 通讯功能 ● 与 CANopen 总线关联的任务
5	显示和命令	<p>此区域由 3 个窗口组成，可用于了解：</p> <ul style="list-style-type: none"> ● CANopen 从站状态 ● CANopen 主站状态 ● 错误计数器状态

从站调试屏幕

概览

此屏幕只能在在线模式下使用。

示意图

下图显示了从站调试屏幕：



标准设备的调试屏幕描述

下表介绍调试屏幕的各个部分及其功能。

编号	元素	功能
1	选项卡	前景中的选项卡指示显示的屏幕类型。这里显示的是调试屏幕。
2	模块区域	包含模块的缩写标题。 同一区域中有两个 LED： <ul style="list-style-type: none"> ● 一个绿色 LED，指示设备在工作（亮/灭） ● 一个红色 LED，指示紧急情况（亮/灭）
3	通道区域	此区域允许您选择要调试的通讯通道。 单击设备可显示以下选项卡： <ul style="list-style-type: none"> ● 描述：给出内置 CANopen 端口的特性。 ● 输入 / 输出对象：允许预先用符号表示输入 / 输出对象。 ● CANopen：允许读 / 写 SDO。 ● 缺省：只能在在线模式下访问 单击通道可显示以下选项卡： <ul style="list-style-type: none"> ● PDO：允许您配置 PDO。 ● 配置：允许您声明和配置 CANopen 主站。 ● 调试：只能在在线模式下访问。 ● 错误控制：只能在在线模式下访问。 此区域还包含一个指示通道状态的 LED。
4	常规参数区域	重新调用与通道关联的功能。
5	正在运行的参数区域	此区域显示所有通道的输入 / 输出数据信息。此区域分为 3 列： <ul style="list-style-type: none"> ● 参数列显示输入 / 输出数据所映射的输入 / 输出对象和未标记的对象 ● 标签列显示输入 / 输出数据的名称 ● 值列显示输入 / 输出数据的值

注意：对于标准设备，值按以下格式显示：

- 十进制（缺省）
- 十六进制
- 二进制

要选择格式，请在调试屏幕中右键单击值，然后选择**显示模式**。

对于带布尔显示的设备 (FTB)，可强制该值。

注意：在**值列**，如果变量显示为红色，则指示值超出范围。单击变量可查看变量的范围。范围显示在状态栏中。

诊断

9

概览

本章目标

本节介绍 CANopen 总线的诊断方法。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
如何执行诊断	166
主站诊断	167
从站诊断	168

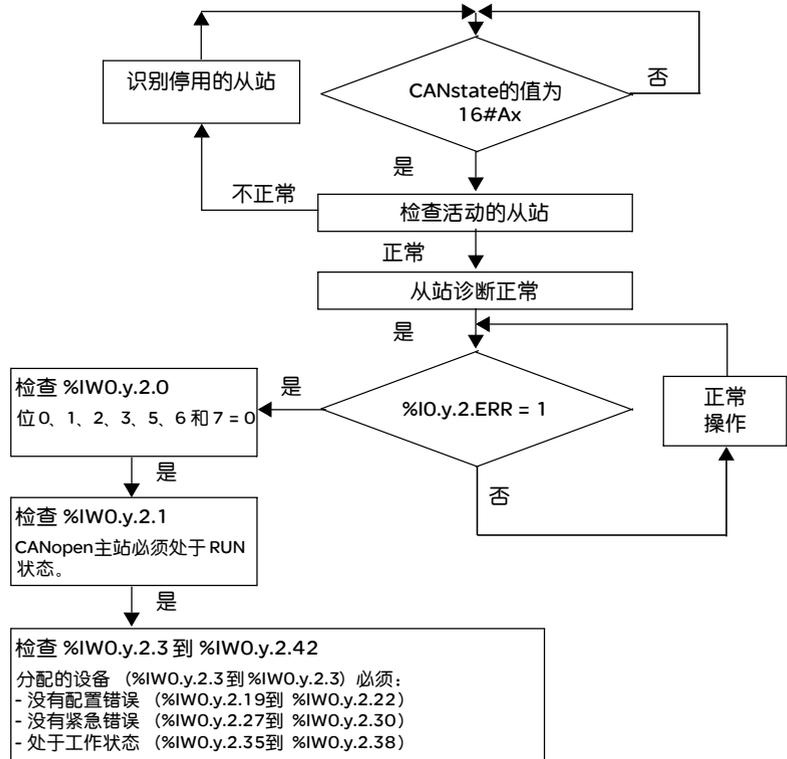
如何执行诊断

概览

开始时，可以使用位于处理器正面的 LED 确定 CANopen 总线上的故障（请参见 CANopen 处理器的可视化诊断）。接下来，可以按照下面描述的过程操作，该过程详细介绍了总线启动管理以及使用 PLC 提供的语言对象执行的检查。

过程

下图指出了此过程的各个阶段：



如何检查 %IW0.y.2

要了解 %IW 的不同状态，请参见 T_COM_CO_BMX 类型 IODDT 的隐式交换对象的详细信息。

主站诊断

概览

可在以下级别对 CANopen 总线主站进行诊断：

- 模块级别
- 通道级别

模块诊断

模块诊断屏幕显示当前的错误，按照错误类别可分为：

- 内部错误
- 外部错误
- 其他错误

通道诊断

通道诊断屏幕显示当前的错误，按照错误类别可分为：

- 外部错误
- 其他错误

下表显示 CANopen 功能可能出现的错误：

错误类型	错误	语言对象
外部	CANopen 主站未运行。	%MWr.m.c.2.0
	一个或几个从站出错或未运行。	%MWr.m.c.2.1
其他	配置错误。	%MWr.m.c.2.3
	低优先级接收队列溢出。	%lWr.m.c.0.0
	CAN 控制器溢出。	%lWr.m.c.0.1
	CAN 控制器已从总线断开连接。	%lWr.m.c.0.2
	CAN 控制器出错。	%lWr.m.c.0.3
	CAN 控制器已不在错误模式。	%lWr.m.c.0.4
	低优先级传输队列溢出。	%lWr.m.c.0.5
	高优先级接收队列溢出。	%lWr.m.c.0.6
	高优先级传输队列溢出。	%lWr.m.c.0.7
	任务循环时间大于 CANopen 主站循环时间。	%lWr.m.c.0.8

从站诊断

概览

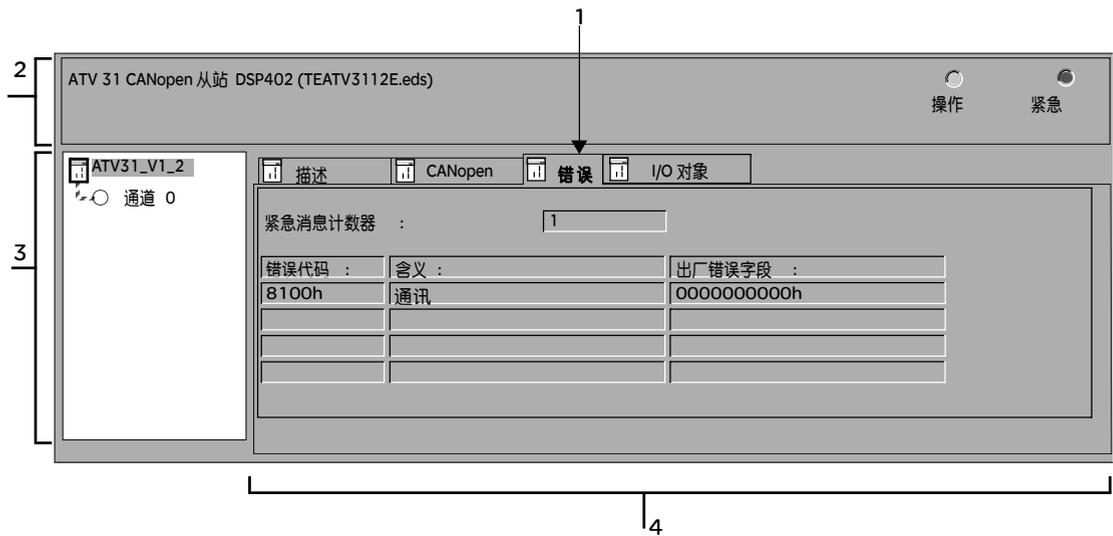
从站诊断只在设备级别进行。

从站诊断屏幕显示：

- 接收到的紧急消息计数器
- 按时间顺序收到的最后四个紧急消息（见紧急对象页 198）

示意图

下图显示了从站诊断屏幕：



元素和功能

下表对构成主站调试屏幕的各个区域进行了描述：

读取	编号	通道
1	选项卡	前景中的选项卡指示显示的屏幕类型。这里显示的是诊断屏幕。
2	模块	此区域由配备有 CANopen 端口的模块的缩写标题和指示模块状态的 2 个 LED 组成。
3	通道	<p>此区域允许您选择要调试的通讯通道。</p> <p>单击设备可显示以下选项卡：</p> <ul style="list-style-type: none"> ● 描述：给出设备的特性。 ● CANopen：允许读 / 写 SDO（只限在线模式）。 ● 故障：允许您查看从站模块生成的最后 4 个错误代码（该选项卡只在在线模式下可用）（请参见制造商的文档）。 ● I/O 对象：允许预先用符号表示输入 / 输出对象。 <p>此区域还包含一个指示通道状态的 LED。</p>
4	显示	<p>此区域包含：</p> <ul style="list-style-type: none"> ● 错误计数器 ● 最后 4 条错误消息（最后接收到的消息位于上部行中）

注意： 错误计数器不能被复位为 0。

概览

本章的目标

本章描述与嵌入到 CPU 模块中的 CANopen 主站关联的隐式和显式语言对象。

注意：系统位 %S9 和系统字 %SW8 及 %SW9 不适用于 CANopen。

注意：有关特定的 CANopen 主站对象的详细信息，请参见 CANopen 专用 IODDT 的语言对象

本章包含了哪些内容？

本章包含了以下部分：

段	段	文件集
10.1	CANopen 通讯的语言对象和 IODDT	172
10.2	适用于所有通讯协议的语言对象和一般 IODDT	179
10.3	CANopen 专用 IODDT 的语言对象	183
10.4	紧急对象	197
10.5	适用于所有模块的 IODDT 类型的 T_GEN_MOD	201

10.1 CANopen 通讯的语言对象和 IODDT

概览

本节主题 本章描述 CANopen 通讯的语言对象和 IODDT 。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
CANopen 通讯语言对象简介	173
与应用专用功能关联的隐式交换语言对象	174
与应用专用功能关联的显式交换语言对象	175
使用显式对象管理交换和报告	177

CANopen 通讯语言对象简介

总则 IODDT 是由制造商预定义的，并包含属于特定应用模块的通道的输入/输出语言对象。

CANopen 通讯拥有一个关联的 IODDT：

- 由所有通讯协议使用的 T_COM_STS_GEN

注意：IODDT 类型变量的创建按两种方式进行：

- I/O 对象选项卡
- 数据编辑器

语言对象类型 每种 IODDT 都包含用于对其进行控制和检查其操作的一组语言对象。

语言对象有两种类型：

- 隐式交换对象，在与模块关联的任务的每个循环中自动交换。
- 显式交换对象，在应用程序请求时使用显式交换指令交换。

隐式交换涉及到模块状态、通讯信号、从站等。

显式交换允许使用模块参数设置和诊断。

注意：每个从站设备都有一个 IODDT（FTB 除外）。有关更多信息，请参考相关设备的用户手册。

与应用专用功能关联的隐式交换语言对象

概览

集成的应用专用接口或额外的模块可以自动增强用于对此接口或模块进行编程的语言对象应用。

这些对象对应于输入 / 输出图像和模块或集成应用专用接口的软件数据。

提示

当 PLC 处于运行或停止模式时，将在任务开始时，在 PLC 存储器中更新模块输入（%I 和 %IW）。

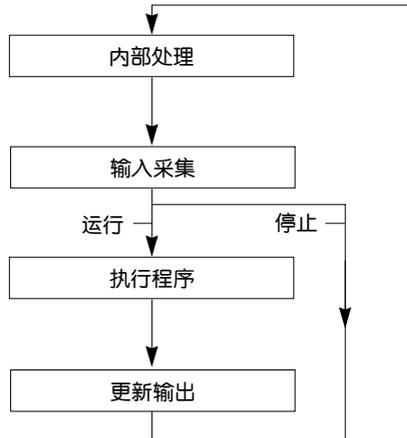
仅当 PLC 处于运行模式时，才会在任务结束时更新模块输出（%Q 和 %QW）。

注意：对于 BMX P34 处理器，当任务出现在 STOP 模式时，根据所选择的配置：

- 输出设置为故障预置位置（故障预置模式）；
- 输出保持其最后一个值（维护模式）。

图

下图显示了 PLC 任务的操作循环（循环执行）。



与应用专用功能关联的显式交换语言对象

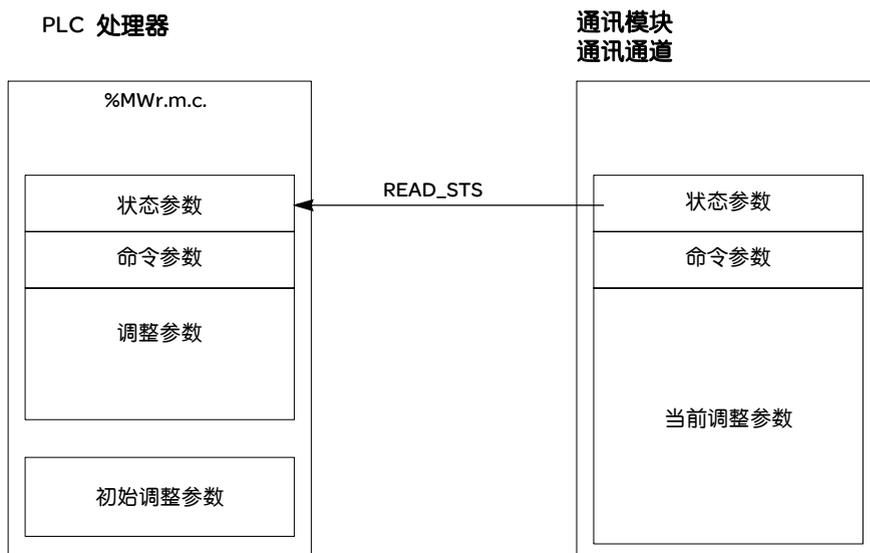
概览

显式交换是根据用户程序请求，使用 `READ_STS` 指令（读取状态字）执行的交换。这些交换适用于属于一个通道的一组相同类型（状态）的 `%MW` 对象。

注意：这些对象提供有关模块的信息（例如，通道上的故障类型）。

使用显式指令的一般原则

下图显示了可以在处理器和模块之间执行的各种类型的显式交换。

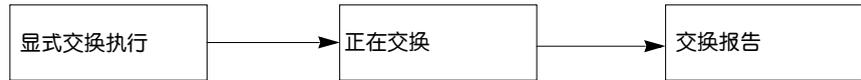


管理交换

在显式交换期间，必须检查交换的性能，以便只在正确执行交换后才考虑数据。
为此提供了以下两种类型的信息：

- 与正在进行的交换有关的信息
- 交换报告

下图说明了管理交换的原理



注意：为了避免同一通道同时发生多个显式交换，在使用此通道调用任何 EF 之前，需要测试与该通道关联的 IODDT 的 EXCH_STS (%MWI.r.m.c.0) 字的值。

使用显式对象管理交换和报告

概览

当在 PLC 存储器与模块之间交换数据时，模块可能需要多个任务循环以确认此信息。所有 IODDT 均使用以下两个字来管理交换：

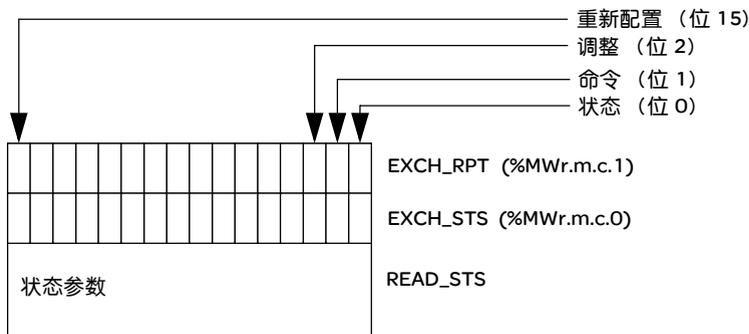
- EXCH_STS (%MWr.m.c.0)：正在交换。
- EXCH_RPT (%MWr.m.c.1)：报告。

注意：应用程序将不检测显式交换的管理（例如，%MW0.0.MOD.0.0），这具体取决于模块的位置：

- 对于机架内模块，显式交换是立即在本地总线上进行的并在执行任务结束之前已完成，因此，例如，在应用程序检查 %MW.0.0.MOD.0.0 位之前，READ_STS 始终已完成。
- 对于远程总线（例如，Fipio），显式交换与执行任务不同步，因此，应用程序可进行检测。

示意图

下图显示了用于管理交换的各个有效位：



有效位的描述

字 EXCH_STS (%MWr.m.c.0) 和 EXCH_RPT (%MWr.m.c.1) 的序号为 0 的位与状态参数关联：

- STS_IN_PROGR 位 (%MWr.m.c.0.0) 指示状态字的读请求是否正在进行。
- STS_ERR 位 (%MWr.m.c.1.0) 指定状态字的读请求是否被模块通道接受。

显式交换的执行指示器：EXCH_STS

下表显示 EXCH_STS (%MWr.m.c.0) 显式交换控制位：

标准符号	类型	访问	含义	地址
STS_IN_PROGR	BOOL	R	正在读取通道状态字	%MWr.m.c.0.0

注意：如果模块不存在或已断开连接，则不会将显式交换对象（如 READ_STS）发送到模块 (STS_IN_PROG (%MWr.m.c.0.0) = 0)，但会刷新这些字。

显式交换报告：EXCH_RPT

下表介绍 EXCH_RPT (%MWr.m.c.1) 报告位：

标准符号	类型	访问	含义	地址
STS_ERR	BOOL	R	读取通道状态字时出错 (1 = 故障)	%MWr.m.c.1.0

10.2 适用于所有通讯协议的语言对象和一般 IODDT

概览

本节的目标 本节介绍适用于所有通讯协议的语言对象和一般 IODDT 。

本章节包含了哪些内容? 本章节包含了以下主题：

主题	文件集
T_COM_STS_GEN 类型的 IODDT 隐式交换对象的详细信息	180
T_COM_STS_GEN 类型的 IODDT 显式交换对象的详细信息	181

T_COM_STS_GEN 类型的 IODDT 隐式交换对象的详细信息

概览

下表显示适用于所有通讯协议（Fipio 除外）的 T_COM_STS_GEN 类型的 IODDT 隐式交换对象。

错误位

下表介绍错误位 CH_ERROR(%lr.m.c.ERR) 的含义。

标准符号	类型	访问	含义	地址
CH_ERROR	EBOOL	读	通讯通道错误位。	%lr.m.c.ERR

T_COM_STS_GEN 类型的 IODDT 显式交换对象的详细信息

概览 本节介绍适用于所有通讯协议（Fipio 除外）的 T_COM_STS_GEN 类型的 IODDT 显式交换对象。本节还包括其位具有特定含义的字类型对象。下面将详细介绍这些对象。
示例变量声明：IODDT_VAR1，类型为 T_COM_STS_GEN。

注意

- 通常情况下，位含义是针对位状态为 1 给出的。特定情况下，会针对位的每个状态给出解释。
- 不是所有位都会用到。

显式交换的执行标志：EXCH_STS 下表介绍了通道 EXCH_STS (%MWr.m.c.0) 中各个通道交换控制位的含义：

标准符号	类型	访问	含义	地址
STS_IN_PROGR	BOOL	读	正在读取通道状态字。	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	读	正在交换当前参数。	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	读	正在交换调整参数。	%MWr.m.c.0.2

显式交换报告：EXCH_RPT 下表介绍交换报告字 EXCH_RPT (%MWr.m.c.1) 各个位的含义。

标准符号	类型	访问	含义	地址
STS_ERR	BOOL	读	读取通道状态字时出错。	%MWr.m.c.1.0
CMD_ERR	BOOL	读	交换命令参数时出错。	%MWr.m.c.1.1
ADJ_ERR	BOOL	读	交换调整参数时出错。	%MWr.m.c.1.2

标准通道故障, CH_FLT 下表介绍了状态字 CH_FLT (%MWr.m.c.2) 的各个位的含义。
由 READ_STS (IODDT_VAR1) 执行读取。

标准符号	类型	访问	含义	地址
NO_DEVICE	BOOL	读	通道上没有运行的设备。	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	读	通道上有一个故障设备。	%MWr.m.c.2.1
BLK	BOOL	读	端子块故障（未连接）。	%MWr.m.c.2.2
TO_ERR	BOOL	读	超时错误（接线有缺陷）。	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	读	内部错误或通道自检。	%MWr.m.c.2.4
CONF_FLT	BOOL	读	硬件和软件配置不同。	%MWr.m.c.2.5
COM_FLT	BOOL	读	与 PLC 通讯时出现问题。	%MWr.m.c.2.6
APPLI_FLT	BOOL	读	应用程序错误（调整或配置错误）。	%MWr.m.c.2.7

10.3 CANopen 专用 IODDT 的语言对象

概览

本节主题 本节介绍 CANopen 专用 IODDT T_COM_CO_BMX 的隐式和显式语言对象。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
T_COM_CO_BMX 类型 IODDT 的隐式交换对象的详细信息	184
T_COM_CO_BMX 类型 IODDT 的显式交换对象的详细信息	194
与配置关联的语言对象	196

T_COM_CO_BMX 类型 IODDT 的隐式交换对象的详细信息

概览

隐式交换对象在与通道关联的任务的每个循环中自动交换。这些对象是 %I、%IW、%Q 和 %QW。

下表介绍了 IODDT T_COM_CO_BMX 的各个隐式交换对象。

下面的表中显示的参数 r、m 和 c 表示模块的拓扑寻址。每个参数的含义如下：

- r 表示机架编号
- m 表示模块编号
- c 表示通道编号

通道错误

下表介绍位 %Ir.m.c.ERR：

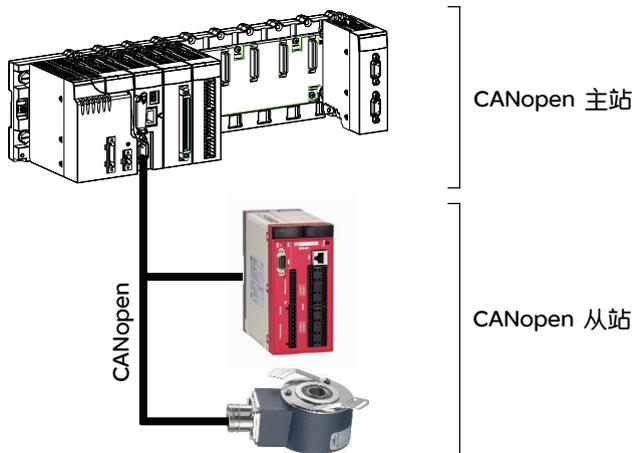
标准符号	类型	访问	说明	地址
CH_ERROR	BOOL	R	通道错误	%Ir.m.c.ERR

主站状态和事件指示器

下表介绍字 %IW_{r.m.c.0} 到 %IW_{r.m.c.2}：

标准符号	类型	访问	说明	地址
COMM_STS	INT	R	主站通讯状态	%IW _{r.m.c.0}
CAN_STS	INT	R	CANopen 主站的状态	%IW _{r.m.c.1}
EVT_STS	INT	R	事件指示器	%IW _{r.m.c.2}

下图给出了一个主站状态指示器示例。



在此示例中，字 `%IW0.0.2.1` 给出了 CANopen 主站的状态。参数如下所示：

- r: '0',
- m: '0',
- c: '2' (CANopen 通道)。

最后一个参数 ('1') 指示已使用的字 (`CAN_STS`)。

下表介绍来自主站和事件指示器的各个状态字中各个位的含义：

地址	说明	位含义
<code>%IW0.m.c.0</code>	主站通讯状态	<p>位 0=1: 低优先级接收队列溢出。CANopen 主站正在通过低优先级队列接收“心跳”和“节点防护”消息以及 SSDO 和 CSDO。</p> <p>位 1=1: CAN 控制器的 FIFO 覆盖。</p> <p>位 2=1: CAN 控制器状态为“BUS OFF”。</p> <p>位 3=1: CAN 控制器未正确运行和停止。故障清除时位将复位。</p> <p>位 4=1: CAN 控制器已脱离非正常状态。</p> <p>位 5=1: 低优先级发送队列溢出。CANopen 主站正在通过低优先级发送队列发送“心跳”和“节点防护”消息以及 SSDO 和 CSDO。</p> <p>位 6=1: 高优先级接收队列溢出。CANopen 主站正在通过高优先级接收队列接收 RPDO、NMT 命令、消息同步以及紧急消息。</p> <p>位 7=1: 高优先级接收队列溢出。CANopen 主站正在通过高优先级队列发送 TPDO、NMT 命令、消息同步以及紧急消息。</p> <p>位 8=1: 指示任务循环比 CANopen 主站循环快（可覆盖输出）。为避免覆盖输出，建议您设置比 CANopen 循环更长的任务循环时间。循环值可从字 <code>%IW0.m.c.59</code> 到 <code>%IW0.m.c.61</code> 中获取。</p>

地址	说明	位含义
%IW.r.m.c.1	CANopen 主站的状态	<p>0x00:INIT；CANopen 主站未初始化。这与 CANopen 模块的“初始化”状态相对应。在此状态下，CANopen 主站无法与网络通讯。</p> <p>0x40:RESET；在“NMT 启动”过程中，CANopen 主站被配置为主站。通过 CAN 总线和 SDO 命令界面，SDO 可配置 CANopen 主站的对象字典。应用程序具有通过 SDO 命令读 / 写对象字典的访问权限。网络管理器的初始化尚未启动。</p> <p>=0x60:NET -INIT；根据 CIA DSP-302 启动。CANopen 主站正在检查从站的分配。</p> <p>=0x61:NET RESET；网络由 NMT 命令“复位所有节点的通讯”重新初始化</p> <p>=0x62:NET -WAIT；CANopen 主站正在等待模块以便运行命令“复位通讯”。</p> <p>0x64:BOOT -CONF；CANopen 主站正在根据 CIA DSP-302 运行各模块的初始化。</p> <p>0x8x:CLEAR；已对网络进行了扫描。主站正在等待启动命令（“启动 CANopen 主站 / 管理器”或“启动网络”）。</p> <p>0xAx:RUN 网络处于“正常操作”状态。</p> <p>0xCx:STOP 网络处于“停止”状态。</p> <p>0xEx:PREOPERATIONAL；网络处于“预操作”状态。</p> <p>0x9x:FATAL ERROR；出现意外行为。必须重新初始化 CANopen 主站。</p> <p>已对网络进行了扫描。状态变量的 4 个最高有效位指示网络的状态（CLEAR、RUN、STOP 和 PREOPERATIONAL）。4 个较低有效位则包含下列其他信息：</p> <p>位 0: 可选模块的错误位</p> <ul style="list-style-type: none"> ● =0: 无错误。 ● =1: 至少有一个可选模块不符合预期网络的配置。 <p>位 1: 必选模块的错误位</p> <ul style="list-style-type: none"> ● =0: 无错误。 ● =1: 至少有一个必选模块未处于预期状态。 <p>位 2: “正常操作”位</p> <ul style="list-style-type: none"> ● =0: 包括 CANopen 主站在内，没有模块处于 CANopen “正常操作”状态。 ● =1: 至少有一个模块处于“正常操作”状态（CANopen 主站除外）。 <p>位 3: CANopen 主站的“正常操作”位</p> <ul style="list-style-type: none"> ● =0: CANopen 主站未处于“正常操作”状态。 ● =1: CANopen 主站处于“正常操作”状态。

地址	说明	位含义
%IWwr.m.c.2	事件指示器	<p>位 0=1 检测到网络通讯错误后，此位仍保持为 1。CANopen 主站的通讯状态会给出确切的错误原因。（CANopen 主站无法正确运行和停止）。</p> <p>位 1=1 模块正在使用 CANopen 主站的节点编号。（CANopen 主站无法正确运行和停止）。</p> <p>位 2=1：必选模块在网络监控方面出现故障（心跳或节点防护）。此状态引起的后果取决于“NMT 启动”对象的配置。如果“NMT 启动”对象未造成总线复位，那么此位有效。</p> <p>位 3=1 模块标识错误或来自必选对象的对象字典 DCF 错误。（CANopen 主站无法正确运行和停止）。</p> <p>位 4=1：可选模块无法通讯。相关模块处于“停止”状态。</p> <p>位 5=1：在自配置阶段未成功创建过程映像和配置 PDO。（CANopen 主站无法正确运行和停止）。</p> <p>位 6=1：在自动配置阶段网络监控操作不当。检测最近的设备连接。</p> <p>位 7=1：如果位列表中的一个位改变状态，此位仍保持为 1。</p> <p>位 8=1：启动过程开始时，CANopen 主站会检查每个从站。如果主站不支持设备的某些功能（例如：对象 1F81h 的位 4 到 6），则此位设置为 1。</p> <p>位 9=1：CANopen 主站接收到的 RPDO 数据太少。（CANopen 主站无法正确运行和停止）。</p> <p>位 10=1：无法正确配置设备。例如：设备不支持对象。（CANopen 主站无法正确运行和停止）。</p> <p>位 11=1：此位指示 SDO 接口专用的应用程序队列溢出。</p> <p>位 12=1：主站循环时间大于 256 毫秒。</p> <p>位 13=1：保留。</p> <p>位 14=1：保留。</p> <p>位 15=1：主站单独存在于总线上（检查电缆是否连接）。</p>

分配的从站

下表介绍字 %IWwr.m.c.3 到 %IWwr.m.c.6：

标准符号	类型	访问	说明	地址
SLAVE_ASSIGNED_1_16	INT	R	用于分配的从站 1 到 16	%IWwr.m.c.3
SLAVE_ASSIGNED_17_32	INT	R	用于分配的从站 17 到 32	%IWwr.m.c.4
SLAVE_ASSIGNED_33_48	INT	R	用于分配的从站 33 到 48	%IWwr.m.c.5
SLAVE_ASSIGNED_49_64	INT	R	用于分配的从站 49 到 63	%IWwr.m.c.6

如果此位等于 0，则未为此位分配从站。

如果此位等于 1，则为此位分配了从站。

节点编号对应于位数 + 1。

配置的从站下表介绍字 %IW_r.m.c.11 到 %IW_r.m.c.14:

标准符号	类型	访问	说明	地址
SLAVE_CONF_1_16	INT	R	用于配置的从站 1 到 16	%IW _r .m.c.11
SLAVE_CONF_17_32	INT	R	用于配置的从站 17 到 32	%IW _r .m.c.12
SLAVE_CONF_33_48	INT	R	用于配置的从站 33 到 48	%IW _r .m.c.13
SLAVE_CONF_49_64	INT	R	用于配置的从站 49 到 63	%IW _r .m.c.14

如果此位等于 0，则从站未配置且无法启动。

如果此位等于 1，则从站已配置，能够启动。

节点编号对应于位数 + 1。

存在配置故障的从站下表介绍字 %IW_r.m.c.19 到 %IW_r.m.c.22:

标准符号	类型	访问	说明	地址
SLAVE_FLT_1_16	INT	R	存在配置故障的从站 1 到 16	%IW _r .m.c.19
SLAVE_FLT_17_32	INT	R	存在配置故障的从站 17 到 32	%IW _r .m.c.20
SLAVE_FLT_33_48	INT	R	存在配置故障的从站 33 到 48	%IW _r .m.c.21
SLAVE_FLT_49_64	INT	R	存在配置故障的从站 49 到 63	%IW _r .m.c.22

如果此位等于 0，则分配的从站与配置相对应。

如果此位等于 1，则分配的从站与配置不对应。

节点编号对应于位数 + 1。

故障从站下表介绍字 %IW_r.m.c.27 到 %IW_r.m.c.30:

标准符号	类型	访问	说明	地址
SLAVE_EM CY_1_16	INT	R	故障从站 1 到 16	%IW _r .m.c.27
SLAVE_EM CY_17_32	INT	R	故障从站 17 到 32	%IW _r .m.c.28
SLAVE_EM CY_33_48	INT	R	故障从站 33 到 48	%IW _r .m.c.29
SLAVE_EM CY_49_64	INT	R	故障从站 49 到 63	%IW _r .m.c.30

如果此位等于 0，则从站没有错误。

如果此位等于 1，则从站有错误。

节点编号对应于位数 + 1。

**正常工作的
从站 1 到 16**

下表介绍字 %IW_r.m.c.35:

标准符号	类型	访问	说明	地址
SLAVE_ACTIV_1	BOOL	R	总线上工作的从站: 设备 1	%IW _r .m.c.35.0
SLAVE_ACTIV_2	BOOL	R	总线上工作的从站: 设备 2	%IW _r .m.c.35.1
SLAVE_ACTIV_3	BOOL	R	总线上工作的从站: 设备 3	%IW _r .m.c.35.2
SLAVE_ACTIV_4	BOOL	R	总线上工作的从站: 设备 4	%IW _r .m.c.35.3
SLAVE_ACTIV_5	BOOL	R	总线上工作的从站: 设备 5	%IW _r .m.c.35.4
SLAVE_ACTIV_6	BOOL	R	总线上工作的从站: 设备 6	%IW _r .m.c.35.5
SLAVE_ACTIV_7	BOOL	R	总线上工作的从站: 设备 7	%IW _r .m.c.35.6
SLAVE_ACTIV_8	BOOL	R	总线上工作的从站: 设备 8	%IW _r .m.c.35.7
SLAVE_ACTIV_9	BOOL	R	总线上工作的从站: 设备 9	%IW _r .m.c.35.8
SLAVE_ACTIV_10	BOOL	R	总线上工作的从站: 设备 10	%IW _r .m.c.35.9
SLAVE_ACTIV_11	BOOL	R	总线上工作的从站: 设备 11	%IW _r .m.c.35.10
SLAVE_ACTIV_12	BOOL	R	总线上工作的从站: 设备 12	%IW _r .m.c.35.11
SLAVE_ACTIV_13	BOOL	R	总线上工作的从站: 设备 13	%IW _r .m.c.35.12
SLAVE_ACTIV_14	BOOL	R	总线上工作的从站: 设备 14	%IW _r .m.c.35.13
SLAVE_ACTIV_15	BOOL	R	总线上工作的从站: 设备 15	%IW _r .m.c.35.14
SLAVE_ACTIV_16	BOOL	R	总线上工作的从站: 设备 16	%IW _r .m.c.35.15

节点编号对应于位数 + 1。

正常工作的从站 17 到 32 下介绍字 %IW_r.m.c.36:

标准符号	类型	访问	说明	地址
SLAVE_ACTIV_17	BOOL	R	总线上工作的从站: 设备 17	%IW _r .m.c.36.0
SLAVE_ACTIV_18	BOOL	R	总线上工作的从站: 设备 18	%IW _r .m.c.36.1
SLAVE_ACTIV_19	BOOL	R	总线上工作的从站: 设备 19	%IW _r .m.c.36.2
SLAVE_ACTIV_20	BOOL	R	总线上工作的从站: 设备 20	%IW _r .m.c.36.3
SLAVE_ACTIV_21	BOOL	R	总线上工作的从站: 设备 21	%IW _r .m.c.36.4
SLAVE_ACTIV_22	BOOL	R	总线上工作的从站: 设备 22	%IW _r .m.c.36.5
SLAVE_ACTIV_23	BOOL	R	总线上工作的从站: 设备 23	%IW _r .m.c.36.6
SLAVE_ACTIV_24	BOOL	R	总线上工作的从站: 设备 24	%IW _r .m.c.36.7
SLAVE_ACTIV_25	BOOL	R	总线上工作的从站: 设备 25	%IW _r .m.c.36.8
SLAVE_ACTIV_26	BOOL	R	总线上工作的从站: 设备 26	%IW _r .m.c.36.9
SLAVE_ACTIV_27	BOOL	R	总线上工作的从站: 设备 27	%IW _r .m.c.36.10
SLAVE_ACTIV_28	BOOL	R	总线上工作的从站: 设备 28	%IW _r .m.c.36.11
SLAVE_ACTIV_29	BOOL	R	总线上工作的从站: 设备 29	%IW _r .m.c.36.12
SLAVE_ACTIV_30	BOOL	R	总线上工作的从站: 设备 30	%IW _r .m.c.36.13
SLAVE_ACTIV_31	BOOL	R	总线上工作的从站: 设备 31	%IW _r .m.c.36.14
SLAVE_ACTIV_32	BOOL	R	总线上工作的从站: 设备 32	%IW _r .m.c.36.15

正常工作的从站 下表介绍字 %IW_r.m.c.37:
33 到 48

标准符号	类型	访问	说明	地址
SLAVE_ACTIV_33	BOOL	R	总线上工作的从站: 设备 33	%IW _r .m.c.37.0
SLAVE_ACTIV_34	BOOL	R	总线上工作的从站: 设备 34	%IW _r .m.c.37.1
SLAVE_ACTIV_35	BOOL	R	总线上工作的从站: 设备 35	%IW _r .m.c.37.2
SLAVE_ACTIV_36	BOOL	R	总线上工作的从站: 设备 36	%IW _r .m.c.37.3
SLAVE_ACTIV_37	BOOL	R	总线上工作的从站: 设备 37	%IW _r .m.c.37.4
SLAVE_ACTIV_38	BOOL	R	总线上工作的从站: 设备 38	%IW _r .m.c.37.5
SLAVE_ACTIV_39	BOOL	R	总线上工作的从站: 设备 39	%IW _r .m.c.37.6
SLAVE_ACTIV_40	BOOL	R	总线上工作的从站: 设备 40	%IW _r .m.c.37.7
SLAVE_ACTIV_41	BOOL	R	总线上工作的从站: 设备 41	%IW _r .m.c.37.8
SLAVE_ACTIV_42	BOOL	R	总线上工作的从站: 设备 42	%IW _r .m.c.37.9
SLAVE_ACTIV_43	BOOL	R	总线上工作的从站: 设备 43	%IW _r .m.c.37.10
SLAVE_ACTIV_44	BOOL	R	总线上工作的从站: 设备 44	%IW _r .m.c.37.11
SLAVE_ACTIV_45	BOOL	R	总线上工作的从站: 设备 45	%IW _r .m.c.37.12
SLAVE_ACTIV_46	BOOL	R	总线上工作的从站: 设备 46	%IW _r .m.c.37.13
SLAVE_ACTIV_47	BOOL	R	总线上工作的从站: 设备 47	%IW _r .m.c.37.14
SLAVE_ACTIV_48	BOOL	R	总线上工作的从站: 设备 48	%IW _r .m.c.37.15

正常工作的从站 下表介绍字 %IW_r.m.c.38:
49 到 64

标准符号	类型	访问	说明	地址
SLAVE_ACTIV_49	BOOL	R	总线上工作的从站: 设备 49	%IW _r .m.c.38.0
SLAVE_ACTIV_50	BOOL	R	总线上工作的从站: 设备 50	%IW _r .m.c.38.1
SLAVE_ACTIV_51	BOOL	R	总线上工作的从站: 设备 51	%IW _r .m.c.38.2
SLAVE_ACTIV_52	BOOL	R	总线上工作的从站: 设备 52	%IW _r .m.c.38.3
SLAVE_ACTIV_53	BOOL	R	总线上工作的从站: 设备 53	%IW _r .m.c.38.4
SLAVE_ACTIV_54	BOOL	R	总线上工作的从站: 设备 54	%IW _r .m.c.38.5
SLAVE_ACTIV_55	BOOL	R	总线上工作的从站: 设备 55	%IW _r .m.c.38.6
SLAVE_ACTIV_56	BOOL	R	总线上工作的从站: 设备 56	%IW _r .m.c.38.7
SLAVE_ACTIV_57	BOOL	R	总线上工作的从站: 设备 57	%IW _r .m.c.38.8
SLAVE_ACTIV_58	BOOL	R	总线上工作的从站: 设备 58	%IW _r .m.c.38.9
SLAVE_ACTIV_59	BOOL	R	总线上工作的从站: 设备 59	%IW _r .m.c.38.10
SLAVE_ACTIV_60	BOOL	R	总线上工作的从站: 设备 60	%IW _r .m.c.38.11
SLAVE_ACTIV_61	BOOL	R	总线上工作的从站: 设备 61	%IW _r .m.c.38.12
SLAVE_ACTIV_62	BOOL	R	总线上工作的从站: 设备 62	%IW _r .m.c.38.13
SLAVE_ACTIV_63	BOOL	R	总线上工作的从站: 设备 63	%IW _r .m.c.38.14

处于停止状态的从站 下表介绍字 %IW_r.m.c.43 到 %IW_r.m.c.46:

标准符号	类型	访问	说明	地址
SLAVE_STOPPED_1_16	INT	R	停止的从站 1 到 16	%IW _r .m.c.43
SLAVE_STOPPED_17_32	INT	R	停止的从站 17 到 32	%IW _r .m.c.44
SLAVE_STOPPED_33_48	INT	R	停止的从站 33 到 48	%IW _r .m.c.45
SLAVE_STOPPED_49_64	INT	R	停止的从站 49 到 63	%IW _r .m.c.46

预操作从站下表介绍字 %IW_r.m.c.51 到 %IW_r.m.c.54:

标准符号	类型	访问	说明	地址
SLAVE_PREOP_1_16	INT	R	预操作从站 1 到 16	%IW _r .m.c.51
SLAVE_PREOP_17_32	INT	R	预操作从站 17 到 32	%IW _r .m.c.52
SLAVE_PREOP_33_48	INT	R	预操作从站 33 到 48	%IW _r .m.c.53
SLAVE_PREOP_49_64	INT	R	预操作从站 49 到 63	%IW _r .m.c.54

主站循环时间

下表介绍与主站循环时间相关的状态字的含义:

地址	说明	含义
%IW _r .m.c.59	最短主站循环时间	CANopen 主站循环时间的最小值 (以毫秒为单位)。
%IW _r .m.c.60	当前主站循环时间	CANopen 主站循环时间的当前值 (以毫秒为单位)。
%IW _r .m.c.61	最长主站循环时间	CANopen 主站循环时间的最大值 (以毫秒为单位)。

复位紧急故障

下表介绍 CANopen 主站命令字的含义:

地址	说明	位含义
%QW _r .m.c.0	CANopen 主站的命令字	<p>位 0=1: 复位紧急从站位列表。位列表复位后, 此位被设置为零。</p> <p>位 1=1: 复位主站的通讯状态的位 8 (%IW₀.0.2.0)。位 8 指示任务循环比 CANopen 主站循环快。位 8 复位后, 位 1 被设置为零。</p> <p>位 2=1: 复位事件指示器 (%IW₀.0.2.2) 的位 7 (更改位列表)。复位后该位将被设置为 0。</p> <p>位 5=1: 复位 CANopen 主站 (可以在发生严重错误时重启主站, 无需关闭和开启电源)。复位主站后该位将被设置为 0。</p> <p>位 6 到 位 15: 保留。</p>

T_COM_CO_BMX 类型 IODDT 的显式交换对象的详细信息

概览

本部分显示 CANopen 主站的显式交换语言对象。

这些对象根据应用程序的请求使用指令 READ_STS 进行交换。

下面的表中显示的参数 r、m 和 c 表示模块的拓扑地址。每个参数的含义如下：

- r：表示机架编号
- m：表示模块在机架中的位置
- c：表示通道编号

执行指示器： EXCH_STS

下表介绍了通道 EXCH_STS (%MWr.m.c.0) 中各个通道交换控制位的含义：

符号	类型	访问	说明	编号
STS_IN_PROGR	BOOL	R	正在读取状态参数	%MWr.m.c.0.0

交换报告： EXCH_RPT

下表介绍通道 EXCH_RPT (%MWr.m.c.1) 的各个运行报告位的含义：

符号	类型	访问	说明	编号
STS_ERR	BOOL	R	读取通道状态时出错	%MWr.m.c.1.0

标准通道故障： 下表说明 CH_FLT (%MWr.m.c.2) 状态字位的含义。由 READ_STS 执行读取：
CH_FLT

对象	函数	含义
%MWr.m.c.2	CANopen 主站的状态	<p>位 0=1: CANopen 主站未处于操作状态。</p> <p>位 1=1: 从站有错误，一个或多个从站有错误或未处于操作状态。</p> <p>位 2: 保留。</p> <p>位 3=1: 配置错误。</p> <p>位 4 到 位 7: 保留。</p> <p>位 8 到 位 10: CAN ERR LED :</p> <ul style="list-style-type: none"> ● 000 = 关闭 ● 001 = 单闪 ● 010 = 双闪 ● 011 = 三闪 ● 111 = 打开 <p>位 11 到 位 13: CAN RUN LED :</p> <ul style="list-style-type: none"> ● 001 = 单闪 ● 100 = 闪烁 ● 111 = 打开 <p>位 14 到 位 15: 保留。</p>
%MWr.m.c.3	一般错误计数	收到的代码为 10xxH 的紧急消息数
%MWr.m.c.4	设备硬件错误计数	收到的代码为 50xxH 的紧急消息数
%MWr.m.c.5	设备软件错误计数	收到的代码为 60xxH 的紧急消息数
%MWr.m.c.6	通讯错误计数	收到的代码为 81xxH 的紧急消息数
%MWr.m.c.7	协议错误计数	收到的代码为 82xxH 的紧急消息数
%MWr.m.c.8	外部错误计数	收到的代码为 90xxH 的紧急消息数
%MWr.m.c.9	设备专用	收到的代码为 FFxxH 的紧急消息数

与配置关联的语言对象

概览

CANopen 主站的配置存储在配置常量 (%KW) 中。

下面的表中显示的参数 r、m 和 c 表示模块的拓扑地址。每个参数的含义如下：

- r: 表示机架编号
- m: 表示模块在机架中的位置
- c: 表示通道编号

配置对象

下表列出了与 CANopen 网络的配置关联的所有过程控制语言对象：

编号	类型	函数	说明
%KW.r.m.c.0	INT	系统使用的常量值	最低有效字节：16#00: <ul style="list-style-type: none"> ● 0: 复位 ● 1: 保持 最高有效字节：16#37。
%KW.r.m.c.1	INT	波特率 (见 CANopen 网络的长度限制页 23)	值已经过编码: <ul style="list-style-type: none"> ● 0 = 1000 千波特 ● 2 = 500 千波特 ● 3 = 250 千波特 ● 4 = 125 千波特 ● 5 = 50 千波特 ● 6 = 20 千波特
%KW.r.m.c.2	INT	COB-ID 同步	缺省值：0080h
%KW.r.m.c.3	INT	同步周期	1 .. 1000 毫秒。
%KW.r.m.c.4	INT	配置位	存储器中输入映像区域 TOR 的大小 (按位数计算)。
%KW.r.m.c.5	INT	配置位	存储器中输出映像区域 TOR 的大小 (按位数计算)。
%KW.r.m.c.6	INT	配置位	输入映像区域 TOR 的开始地址 (%M)。
%KW.r.m.c.7	INT	配置位	输出映像区域 TOR 的开始地址 (%M)。
%KW.r.m.c.8	INT	配置位	存储器中输入映像区域的大小 (按字数计算)。
%KW.r.m.c.9	INT	配置位	存储器中输出映像区域的大小 (按字数计算)。
%KW.r.m.c.10	INT	配置位	输出映像区域的开始地址 (%MW)。
%KW.r.m.c.11	INT	配置位	输出映像区域的开始地址 (%MW)。

10.4 紧急对象

概览

本节主题 本节介绍 CANopen 通讯的紧急对象。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
紧急对象	198

紧急对象

概览

已为 CANopen 定义了用于诊断应用程序的紧急对象 (EMCY)。

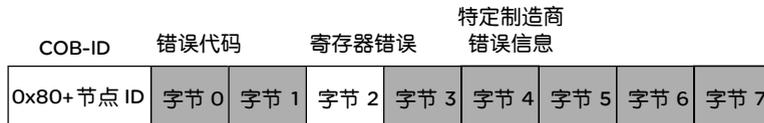
这些对象的 COB-ID 包含生成紧急消息的设备的节点标识。紧急对象的 COB-ID 按以下方式构造：

$$\text{COB-ID}_{\text{EMCY}} = 0x80 + \text{节点标识}$$

EMCY 对象的数据字段由 8 个字节组成，包含：

- 紧急错误代码（2 个字节）
- 错误寄存器（1 个字节）
- 出厂特定错误信息（5 个字节）

下图显示了 EMCY 对象的结构：



注意： 错误代码和错误寄存器的内容由 CiA 指定。

使用错误选项卡（请参见从站诊断），您可查看按时间顺序收到的最后 4 条紧急消息。

错误代码 00xx

下表描述错误代码 00xx 的内容：

错误代码（十六进制）	说明
00xx	错误复位为零或没有错误

错误代码 10xx

下表描述错误代码 10xx 的内容：

错误代码（十六进制）	说明
10xx	一般错误

错误代码 2xxx

下表描述错误代码 2xxx 的内容:

错误代码 (十六进制)	说明
20xx	电流
21xx	设备输入端的电流
22xx	设备的内部电流
23xx	设备输出端的电流

错误代码 3xxx

下表描述错误代码 3xxx 的内容:

错误代码 (十六进制)	说明
30xx	电压
31xx	主电压
32xx	设备的内部电压
33xx	输出电压

错误代码 4xxx

下表描述错误代码 4xxx 的内容:

错误代码 (十六进制)	说明
40xx	温度
41xx	环境温度
42xx	设备温度

错误代码 50xx

下表描述错误代码 50xx 的内容:

错误代码 (十六进制)	说明
50xx	设备硬件

错误代码 6xxx

下表描述错误代码 6xxx 的内容:

错误代码 (十六进制)	说明
60xx	设备软件
61xx	内部软件
62xx	用户软件
63xx	数据集

错误代码 70xx

下表描述错误代码 70xx 的内容:

错误代码 (十六进制)	说明
70xx	附加模块

错误代码 8xxx

下表描述错误代码 8xxx 的内容:

错误代码 (十六进制)	说明
80xx	监控
81xx	通讯
8110	CAN 溢出 (对象丢失)
8120	CAN 处于被动错误模式
8130	寿命保护错误或心跳错误
8140	已从总线恢复
8150	COB-ID 传输过程中出现冲突
82xx	协议错误
8210	PDO 由于长度错误, 未被处理
8220	PDO 长度超出限制

错误代码 90xx

下表描述错误代码 90xx 的内容:

错误代码 (十六进制)	说明
90xx	外部错误

错误代码 Fxxx

下表描述错误代码 Fxxx 的内容:

错误代码 (十六进制)	说明
F0xx	附加功能
FFxx	特定于设备

10.5 适用于所有模块的 IODDT 类型的 T_GEN_MOD

概览

本节的目标 本节介绍适用于所有模块的 IODDT 类型 T_GEN_MOD 及关联的语言对象。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
T_GEN_MOD 类型 IODDT 的语言对象的详细信息	202

T_GEN_MOD 类型 IODDT 的语言对象的详细信息

简介 Modicon M340 PLC 的所有模块都具有关联的 T_GEN_MOD 类型的 IODDT。

注意 通常情况下，位含义是针对位状态为 1 给出的。特定情况下，会针对位的每个状态给出解释。

某些位未使用。

对象列表 下表显示 IODDT 的对象。

标准符号	类型	访问	含义	地址
MOD_ERROR	BOOL	R	模块错误位	%I.r.m.MOD.ERR
EXCH_STS	INT	R	模块交换控制字	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	正在读取模块的状态字	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	交换报告字	%MWr.m.MOD.1
STS_ERR	BOOL	R	读取模块状态字时发生事件	%MWr.m.MOD.1.0
MOD_FLT	INT	R	模块的内部错误字	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	内部错误，模块无法操作	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	故障通道	%MWr.m.MOD.2.1
BLK	BOOL	R	端子块接线错误	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	硬件或软件配置错误	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	模块缺失或不工作	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	模块的内部错误字（仅限 Fipio 扩展）	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	内部检测到的故障，模块无法使用 （仅限 Fipio 扩展）	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	故障通道（仅限 Fipio 扩展）	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	端子块接线错误（仅限 Fipio 扩展）	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	硬件或软件配置错误（仅限 Fipio 扩展）	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	模块缺失或不工作（仅限 Fipio 扩展）	%MWr.m.MOD.2.14

快速入门：CANopen 实现示例



IV

概览

概述

本节提供一个 CANopen 实现示例。

本部分包含了哪些内容？

本部分包括以下内容：

章	章节标题	文件集
11	应用程序描述	205
12	使用 Unity Pro 安装应用程序	209
13	启动应用程序	243

应用程序描述

11

概览

本章主题

本章描述应用程序规范，并演示使用操作员屏幕示例所得到的最终结果。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
应用程序概述	206

应用程序概述

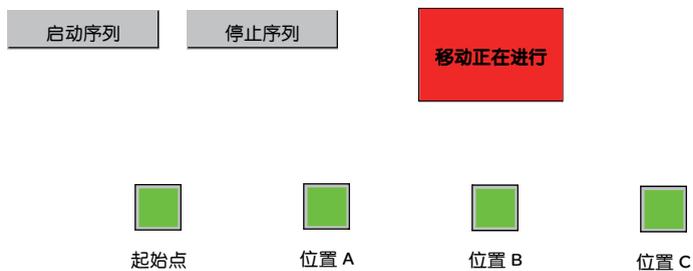
概览

本文中描述的应用程序用于驱动工作的移动设备。

移动设备按照定义的位置顺序移入不同的工作位置。移动设备在这些位置停止几秒。应用程序的控制资源基于操作员屏幕，它显示各个位置传感器的状态和实际的移动位置值。移动设备进行移动时，系统会闪烁警告消息。

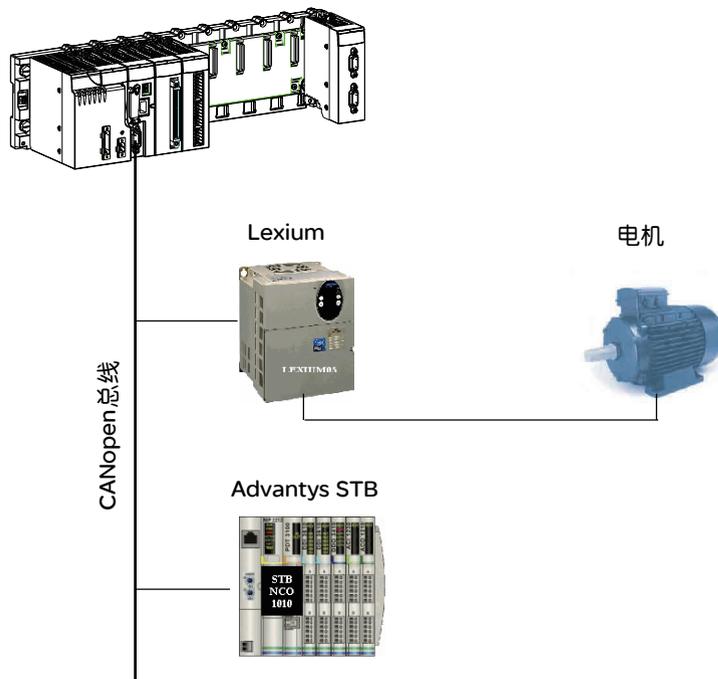
应用示意图

下面是应用程序的最终操作员屏幕：



设备可按如下方式连接：

BMX P34 2010



操作模式

操作模式如下所示：

- **启动序列**按钮用于启动定义的序列。
 - 在此示例中，移动设备首先进入 B 位置，然后进入 A 位置，最终到达 C 位置，在回到起始点之前，等待新的启动请求。
 - 移动设备在每个位置停止几秒，用于仿真操作时间。
 - **停止序列**按钮可中止移动设备序列。移动设备停在最后一个目标位置并返回起始点，等待新的启动请求。
-

使用 Unity Pro 安装应用程序

12

概览

本章主题

本章描述创建所述应用程序的过程。它详略得当地介绍了创建应用程序的各个组件的步骤。

本章包含了哪些内容？

本章包含了以下部分：

段	段	文件集
12.1	所采用的解决方案简介	210
12.2	开发应用程序	213

12.1 所采用的解决方案简介

概览

本节主题 本节介绍用于开发应用程序的解决方案。它介绍怎样选择技术，并给出创建应用程序的时间线。

本章节包含了哪些内容? 本章节包含了以下主题：

主题	文件集
可选用的技术	211
使用 Unity Pro 的过程中的各个步骤	212

可选用的技术

概览

使用 Unity Pro 编写移动驱动应用程序的方法有多种。建议采用的方法是在 CANopen 网络上使用 Lexium 05 伺服驱动器和安装 Advantys STB 岛。

可选技术

下表显示用于应用程序的可选技术：

对象	选用的技术
Lexium 操作模式	使用定位模式。此模式允许您通过 CANopen 网络将目标位置发送至 Lexium 05 伺服驱动器。
传感器接口	使用 STB Advantys。此设备是分布式 I/O、电源及其他模块的组合，用作开放式现场总线网络中的岛节点
监控屏幕	使用库中的元素和新对象。
主监控程序	此程序是使用顺序功能图 (SFC) 开发的，也称为 GRAFCET。各个段和转换是以梯形图 (LD) 语言和结构化文本 (ST) 语言创建的。

注意：此示例显示了 PDO 和 SDO 朝向速度驱动器的交换。但是，为了进行速度驱动器配置和控制，建议使用运动功能块。

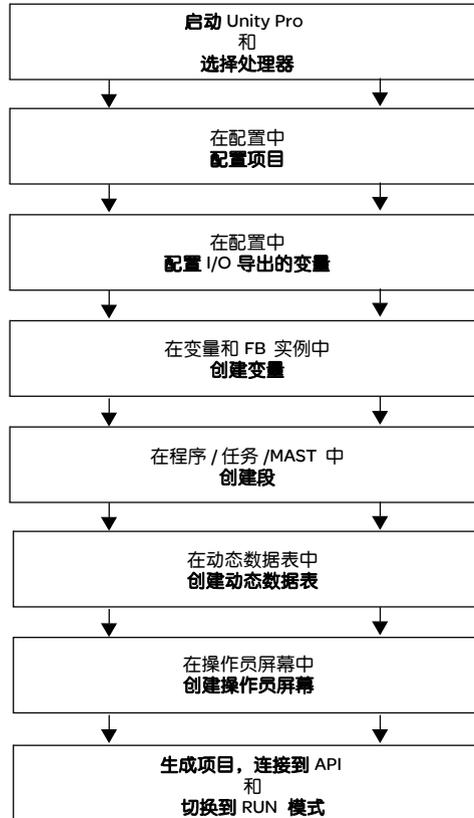
使用 Unity Pro 的过程中的各个步骤

概览

下面的逻辑图显示创建应用程序时要遵循的各个步骤。为了能够正确地定义所有的应用程序元素，必须按照先后顺序进行。

描述

各种类型的描述如下：



12.2 开发应用程序

概览

本节主题 本节分步描述如何使用 Unity Pro 创建应用程序。

本章节包含了哪些内容? 本章节包含了以下主题:

主题	文件集
创建项目	214
CANopen 总线配置	215
CANopen 主站配置	220
设备配置	221
变量声明	224
在 SFC 中创建用于管理移动序列的程序	227
使用 LD 创建程序以用于应用程序的执行	231
使用 LD 为操作员屏幕动态显示创建程序	233
使用 ST 为 Lexium 配置创建程序	234
创建动态数据表	238
创建操作员屏幕	240

创建项目

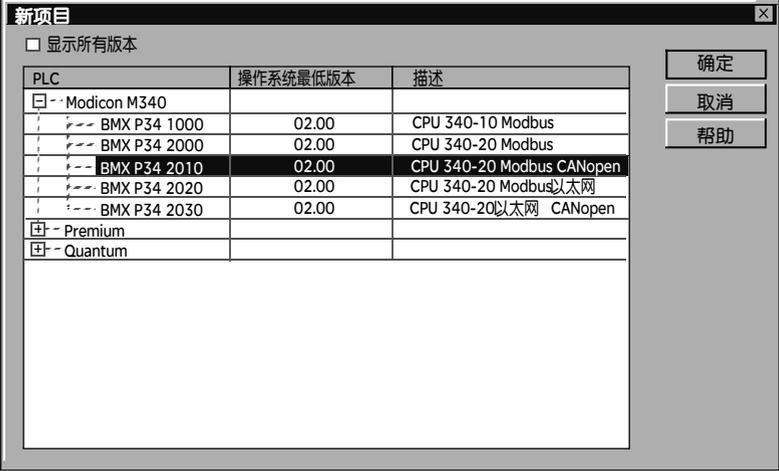
概览

使用 Unity Pro 开发应用程序时，需要创建与 PLC 关联的项目。

注意：有关更多信息，请参见 Unity Pro 联机帮助（依次单击 ?、Unity、Unity Pro、操作模式和项目配置）。

创建项目的过程

下表显示使用 Unity Pro 创建项目的过程。

步骤	操作																											
1	启动 Unity Pro 软件（在本例中为 XL 版本）。																											
2	依次单击“文件”、“新建”，然后选择 CANopen 主站 PLC（如 BMX P34 2010）：  <table border="1" data-bbox="466 695 1094 1079"> <thead> <tr> <th>PLC</th> <th>操作系统最低版本</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>[-] Modicon M340</td> <td></td> <td></td> </tr> <tr> <td> [+] BMX P34 1000</td> <td>02.00</td> <td>CPU 340-10 Modbus</td> </tr> <tr> <td> [+] BMX P34 2000</td> <td>02.00</td> <td>CPU 340-20 Modbus</td> </tr> <tr> <td> [+] BMX P34 2010</td> <td>02.00</td> <td>CPU 340-20 Modbus CANopen</td> </tr> <tr> <td> [+] BMX P34 2020</td> <td>02.00</td> <td>CPU 340-20 Modbus以太网</td> </tr> <tr> <td> [+] BMX P34 2030</td> <td>02.00</td> <td>CPU 340-20以太网 CANopen</td> </tr> <tr> <td>[+] Premium</td> <td></td> <td></td> </tr> <tr> <td>[+] Quantum</td> <td></td> <td></td> </tr> </tbody> </table>	PLC	操作系统最低版本	描述	[-] Modicon M340			[+] BMX P34 1000	02.00	CPU 340-10 Modbus	[+] BMX P34 2000	02.00	CPU 340-20 Modbus	[+] BMX P34 2010	02.00	CPU 340-20 Modbus CANopen	[+] BMX P34 2020	02.00	CPU 340-20 Modbus以太网	[+] BMX P34 2030	02.00	CPU 340-20以太网 CANopen	[+] Premium			[+] Quantum		
PLC	操作系统最低版本	描述																										
[-] Modicon M340																												
[+] BMX P34 1000	02.00	CPU 340-10 Modbus																										
[+] BMX P34 2000	02.00	CPU 340-20 Modbus																										
[+] BMX P34 2010	02.00	CPU 340-20 Modbus CANopen																										
[+] BMX P34 2020	02.00	CPU 340-20 Modbus以太网																										
[+] BMX P34 2030	02.00	CPU 340-20以太网 CANopen																										
[+] Premium																												
[+] Quantum																												
3	单击“确定”确认。																											

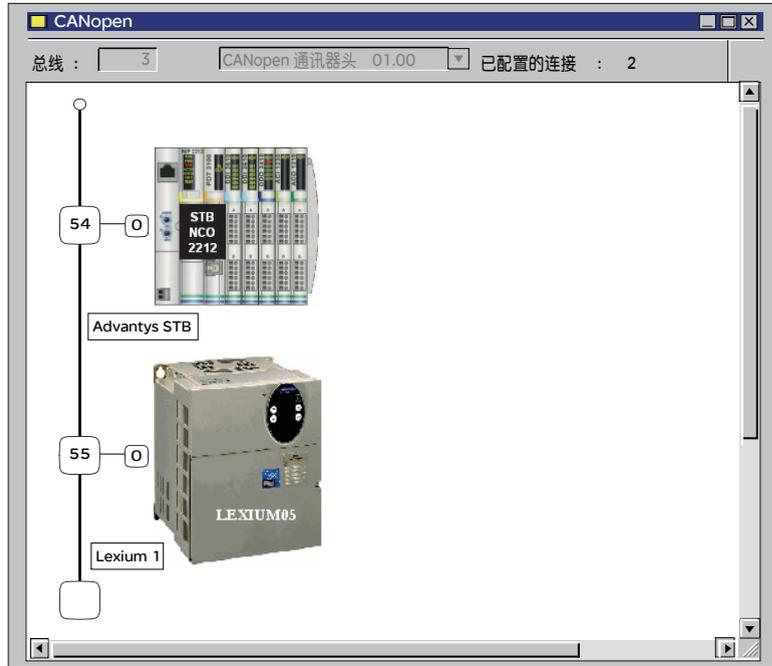
CANopen 总线配置

概览

开发 CANopen 应用程序涉及到选择正确的从站设备和适当的配置。

CANopen 总线示意图

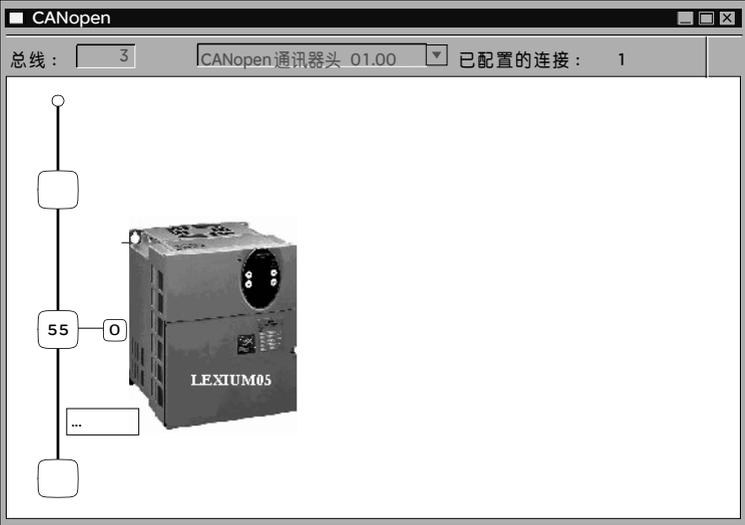
下面的屏幕显示已配置的 CANopen 总线：



**CANopen
总线配置**

下表介绍选择 CANopen 从站的过程：

步骤	操作
1	在项目浏览器中，双击配置，然后双击 3 : CANopen。将打开 CANopen 窗口。
2	<p>在 CANopen 窗口中，双击从站必须链接到的节点。 结果：将打开下面的窗口。</p> 
3	在新设备窗口中，输入节点号 (55)，然后双击运动并选择 Lexium05。

步骤	操作
4	<p>单击 "确定" 以确认。 结果：从站模块被声明。</p> 
5	<p>按照同样的过程声明 Advantys STB 岛。在新设备窗口中，输入节点号 (54)，然后双击其他并选择 STB_NCO_2212。</p>

注意：此示例显示了 PDO 和 SDO 朝向速度驱动器的交换。但是，为了进行速度驱动器配置和控制，建议使用运动功能块。

注意：必须使用 Advantys 配置软件设置此 Advantys STB 岛的配置。

STB 岛配置

下表介绍使用 Advantys 配置软件配置 STB 岛的过程：

步骤	操作
1	打开 Advantys 配置软件版本 2.2.0.2 并创建新的 STB 岛。
2	在该岛上插入一个 STB NCO2212 电源模块、一个 STB DDI3420 离散量输入模块和一个 STB DD03410 离散量输出模块。
3	保存配置并单击文件 / 导出，从而以 DCF 格式导出岛。 将打开导出窗口：
	
4	单击确定。
5	启动 Unity Pro 并打开将在其中使用 STB 岛的项目。
6	在总线编辑器中添加 STB 设备。
7	右键单击 STB 设备，然后单击打开模块。
8	在 PDO 选项卡中，单击导入 DCF 按钮（请参见 STB 配置）。
9	单击 " 确定 " 进行确认。

 **警告**

意外操作的风险

在配置 STB Island 期间由 Advantys 生成的符号文件 *.xsy 不得用于 Unity Pro 中。During an 将 *.xsy 文件从 Advantys 导入 Unity Pro for CAN 期间，不支持开放设备。

在 PDO 表中指定的 %MW 对象与在 CAN 开放头的配置中定义的对象不属于同一范围。

如果不遵守这个警告将会导致 死亡，严重伤害，或设备损坏。

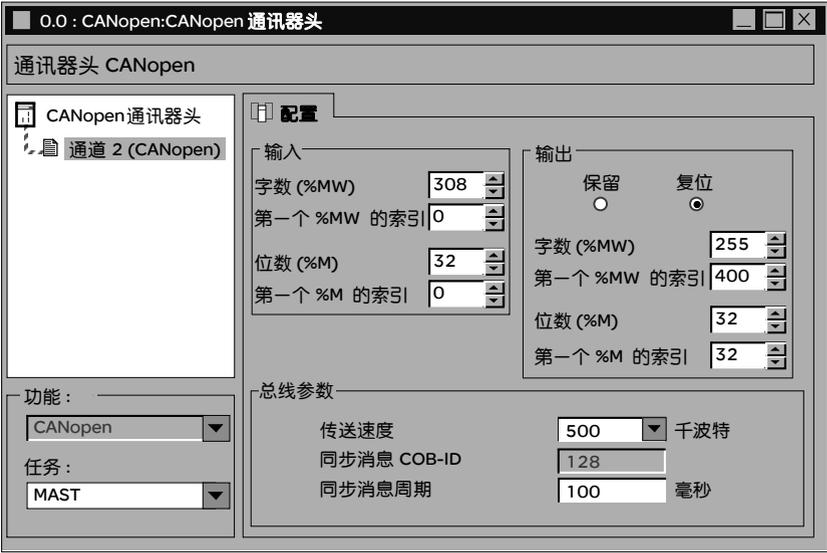
CANopen 主站配置

概览

开发 CANopen 应用程序涉及到选择正确的 CANopen 主站 PLC 配置。

CANopen 主站 PLC 配置

下表介绍配置 CANopen 主站 PLC 的过程：

步骤	操作
1	在项目浏览器中，依次双击配置 °Φ0:BMS XBP 0800 和 0:BMX P34 2010。双击 CANopen 以便访问 CANopen 通讯头窗口。
2	在输入和输出配置区域，输入第一个字（%MW）的索引以及所需的字数。
3	在总线参数区域，选择应用程序传送速度。此例中，选择 500 千波特。 
4	单击工具栏上的 <input checked="" type="checkbox"/> 按钮以验证配置。

注意：生成项目时，输出窗口中可能会显示警告和错误消息。如果未显示，请单击视图 / 输出窗口。

警告消息指示配置的字数比总线上需要的字数多。

错误消息指示缺少配置的字。

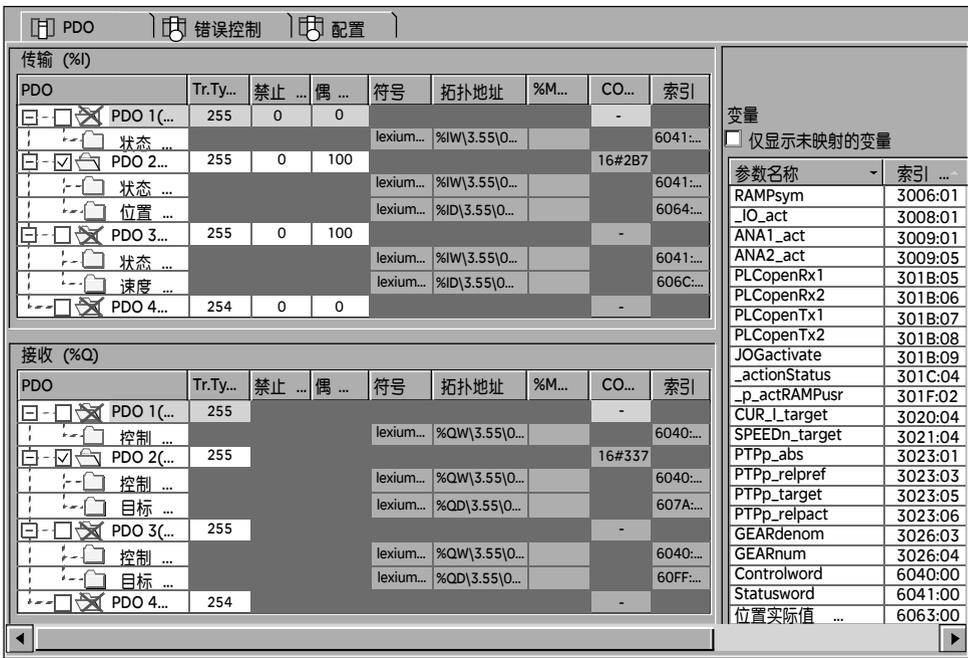
设备配置

概览

声明了从站后，即可访问其配置窗口。

Lexium 伺服 驱动器配置

下表显示 Lexium 的配置过程：

步骤	操作																																																																																																																																																	
1	在项目浏览器中，双击配置，然后双击 3：CANopen。																																																																																																																																																	
2	在 CANopen 窗口中，双击 Lexium 图标。Lexium 配置窗口打开。																																																																																																																																																	
3	单击 PDO 选项卡查看 PDO 配置、变量及其拓扑地址。																																																																																																																																																	
4	<p>对于此示例，在传输 (%I) 和接收 (%Q) 窗口中选择 PDO2 (静态)。</p>  <p>The screenshot shows the Lexium configuration interface with the following details:</p> <ul style="list-style-type: none"> Transmission (%I) Table: <table border="1"> <thead> <tr> <th>PDO</th> <th>Tr.Ty...</th> <th>禁止</th> <th>偶 ...</th> <th>符号</th> <th>拓扑地址</th> <th>%M...</th> <th>CO...</th> <th>索引</th> </tr> </thead> <tbody> <tr> <td>PDO 1(...)</td> <td>255</td> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td><input checked="" type="checkbox"/> PDO 2(...)</td> <td>255</td> <td>0</td> <td>100</td> <td>lexium...</td> <td>%IW\3.55\0...</td> <td></td> <td>16#2B7</td> <td>6041:...</td> </tr> <tr> <td><input type="checkbox"/> PDO 3(...)</td> <td>255</td> <td>0</td> <td>100</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td><input type="checkbox"/> PDO 4(...)</td> <td>254</td> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> Reception (%Q) Table: <table border="1"> <thead> <tr> <th>PDO</th> <th>Tr.Ty...</th> <th>禁止</th> <th>偶 ...</th> <th>符号</th> <th>拓扑地址</th> <th>%M...</th> <th>CO...</th> <th>索引</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> PDO 1(...)</td> <td>255</td> <td></td> <td></td> <td>lexium...</td> <td>%QW\3.55\0...</td> <td></td> <td></td> <td>6040:...</td> </tr> <tr> <td><input checked="" type="checkbox"/> PDO 2(...)</td> <td>255</td> <td></td> <td></td> <td>lexium...</td> <td>%QW\3.55\0...</td> <td></td> <td>16#337</td> <td>6040:...</td> </tr> <tr> <td><input checked="" type="checkbox"/> PDO 3(...)</td> <td>255</td> <td></td> <td></td> <td>lexium...</td> <td>%QD\3.55\0...</td> <td></td> <td></td> <td>607A:...</td> </tr> <tr> <td><input checked="" type="checkbox"/> PDO 4(...)</td> <td>254</td> <td></td> <td></td> <td>lexium...</td> <td>%QW\3.55\0...</td> <td></td> <td></td> <td>6040:...</td> </tr> <tr> <td><input checked="" type="checkbox"/> PDO 4(...)</td> <td>254</td> <td></td> <td></td> <td>lexium...</td> <td>%QD\3.55\0...</td> <td></td> <td></td> <td>60FF:...</td> </tr> </tbody> </table> Variable List: <table border="1"> <thead> <tr> <th>参数名称</th> <th>索引</th> </tr> </thead> <tbody> <tr><td>RAMPsym</td><td>3006:01</td></tr> <tr><td>_IO_act</td><td>3008:01</td></tr> <tr><td>ANA1_act</td><td>3009:01</td></tr> <tr><td>ANA2_act</td><td>3009:05</td></tr> <tr><td>PLCopenRx1</td><td>301B:05</td></tr> <tr><td>PLCopenRx2</td><td>301B:06</td></tr> <tr><td>PLCopenTx1</td><td>301B:07</td></tr> <tr><td>PLCopenTx2</td><td>301B:08</td></tr> <tr><td>JOgactivate</td><td>301B:09</td></tr> <tr><td>_actionStatus</td><td>301C:04</td></tr> <tr><td>_p_actRAMPusr</td><td>301F:02</td></tr> <tr><td>CUR_L_target</td><td>3020:04</td></tr> <tr><td>SPEEDn_target</td><td>3021:04</td></tr> <tr><td>PTPp_abs</td><td>3023:01</td></tr> <tr><td>PTPp_relpref</td><td>3023:03</td></tr> <tr><td>PTPp_target</td><td>3023:05</td></tr> <tr><td>PTPp_relpact</td><td>3023:06</td></tr> <tr><td>GEARdenom</td><td>3026:03</td></tr> <tr><td>GEARnum</td><td>3026:04</td></tr> <tr><td>Controlword</td><td>6040:00</td></tr> <tr><td>Statusword</td><td>6041:00</td></tr> <tr><td>位置实际值</td><td>6063:00</td></tr> </tbody> </table> 	PDO	Tr.Ty...	禁止	偶 ...	符号	拓扑地址	%M...	CO...	索引	PDO 1(...)	255	0	0						<input checked="" type="checkbox"/> PDO 2(...)	255	0	100	lexium...	%IW\3.55\0...		16#2B7	6041:...	<input type="checkbox"/> PDO 3(...)	255	0	100						<input type="checkbox"/> PDO 4(...)	254	0	0						PDO	Tr.Ty...	禁止	偶 ...	符号	拓扑地址	%M...	CO...	索引	<input checked="" type="checkbox"/> PDO 1(...)	255			lexium...	%QW\3.55\0...			6040:...	<input checked="" type="checkbox"/> PDO 2(...)	255			lexium...	%QW\3.55\0...		16#337	6040:...	<input checked="" type="checkbox"/> PDO 3(...)	255			lexium...	%QD\3.55\0...			607A:...	<input checked="" type="checkbox"/> PDO 4(...)	254			lexium...	%QW\3.55\0...			6040:...	<input checked="" type="checkbox"/> PDO 4(...)	254			lexium...	%QD\3.55\0...			60FF:...	参数名称	索引	RAMPsym	3006:01	_IO_act	3008:01	ANA1_act	3009:01	ANA2_act	3009:05	PLCopenRx1	301B:05	PLCopenRx2	301B:06	PLCopenTx1	301B:07	PLCopenTx2	301B:08	JOgactivate	301B:09	_actionStatus	301C:04	_p_actRAMPusr	301F:02	CUR_L_target	3020:04	SPEEDn_target	3021:04	PTPp_abs	3023:01	PTPp_relpref	3023:03	PTPp_target	3023:05	PTPp_relpact	3023:06	GEARdenom	3026:03	GEARnum	3026:04	Controlword	6040:00	Statusword	6041:00	位置实际值	6063:00
PDO	Tr.Ty...	禁止	偶 ...	符号	拓扑地址	%M...	CO...	索引																																																																																																																																										
PDO 1(...)	255	0	0																																																																																																																																															
<input checked="" type="checkbox"/> PDO 2(...)	255	0	100	lexium...	%IW\3.55\0...		16#2B7	6041:...																																																																																																																																										
<input type="checkbox"/> PDO 3(...)	255	0	100																																																																																																																																															
<input type="checkbox"/> PDO 4(...)	254	0	0																																																																																																																																															
PDO	Tr.Ty...	禁止	偶 ...	符号	拓扑地址	%M...	CO...	索引																																																																																																																																										
<input checked="" type="checkbox"/> PDO 1(...)	255			lexium...	%QW\3.55\0...			6040:...																																																																																																																																										
<input checked="" type="checkbox"/> PDO 2(...)	255			lexium...	%QW\3.55\0...		16#337	6040:...																																																																																																																																										
<input checked="" type="checkbox"/> PDO 3(...)	255			lexium...	%QD\3.55\0...			607A:...																																																																																																																																										
<input checked="" type="checkbox"/> PDO 4(...)	254			lexium...	%QW\3.55\0...			6040:...																																																																																																																																										
<input checked="" type="checkbox"/> PDO 4(...)	254			lexium...	%QD\3.55\0...			60FF:...																																																																																																																																										
参数名称	索引																																																																																																																																																	
RAMPsym	3006:01																																																																																																																																																	
_IO_act	3008:01																																																																																																																																																	
ANA1_act	3009:01																																																																																																																																																	
ANA2_act	3009:05																																																																																																																																																	
PLCopenRx1	301B:05																																																																																																																																																	
PLCopenRx2	301B:06																																																																																																																																																	
PLCopenTx1	301B:07																																																																																																																																																	
PLCopenTx2	301B:08																																																																																																																																																	
JOgactivate	301B:09																																																																																																																																																	
_actionStatus	301C:04																																																																																																																																																	
_p_actRAMPusr	301F:02																																																																																																																																																	
CUR_L_target	3020:04																																																																																																																																																	
SPEEDn_target	3021:04																																																																																																																																																	
PTPp_abs	3023:01																																																																																																																																																	
PTPp_relpref	3023:03																																																																																																																																																	
PTPp_target	3023:05																																																																																																																																																	
PTPp_relpact	3023:06																																																																																																																																																	
GEARdenom	3026:03																																																																																																																																																	
GEARnum	3026:04																																																																																																																																																	
Controlword	6040:00																																																																																																																																																	
Statusword	6041:00																																																																																																																																																	
位置实际值	6063:00																																																																																																																																																	
5	单击错误控制选项卡并将节点心跳生成器时间设置为 300 毫秒。																																																																																																																																																	
6	单击工具栏上的 <input checked="" type="checkbox"/> 按钮以验证配置。																																																																																																																																																	
7	关闭该窗口。																																																																																																																																																	

STB 配置 下表显示加载使用 Advantys 配置软件定义的配置的过程：

步骤	操作																																																																																																													
1	在项目浏览器中，双击配置，然后双击 3：CANopen。																																																																																																													
2	在 CANopen 窗口中，双击 Advantys STB 图标。STB NCO2212 配置窗口打开。																																																																																																													
3	<p>在功能区域，选择自动配置。</p> <div style="text-align: center;">  <p>功能： 自动配置</p> </div> <p>在本例中，我们使用自动配置功能，因为 STB 岛上插入了可自动配置的模块（请参见 Advantys STB 配置）。</p>																																																																																																													
4	单击 PDO 选项卡查看 PDO 配置、变量及其拓扑地址。单击水平滚动条的向右按钮，可以看到导入 DCF 按钮。																																																																																																													
5	<p>单击导入 DCF 按钮，加载使用 Advantys 配置软件生成的 DCF 配置文件。</p>  <p>The screenshot shows the configuration window with tabs for PDO, 错误控制 (Error Control), and 配置 (Configuration). The PDO configuration is split into '传输 (%I)' (Transmit) and '接收 (%Q)' (Receive) sections. The '传输 (%I)' section has a table with columns: PDO, Tr.Ty..., 禁止 ... (Prohibit), 偶 ... (Even), 符号 (Symbol), 拓扑地址 (Topology Address), %M..., CO..., and 索引 (Index). The '接收 (%Q)' section has a similar table. On the right, there is a '变量' (Variables) section with a checkbox '仅显示未映射的变量' (Show only unmapped variables) and a list of parameter names and indices.</p> <table border="1" style="margin-left: 20px;"> <caption>传输 (%I)</caption> <thead> <tr> <th>PDO</th> <th>Tr.Ty...</th> <th>禁止 ...</th> <th>偶 ...</th> <th>符号</th> <th>拓扑地址</th> <th>%M...</th> <th>CO...</th> <th>索引</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> PDO 1</td> <td>255</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> <td>16#182</td> <td></td> </tr> <tr> <td>数字量 8 ...</td> <td></td> <td></td> <td></td> <td></td> <td>%IW\3.2\0.0...</td> <td>%MW184</td> <td></td> <td>6000:01</td> </tr> <tr> <td>数字量 8 ...</td> <td></td> <td></td> <td></td> <td></td> <td>%IW\3.2\0.0...</td> <td>%MW185</td> <td></td> <td>6000:02</td> </tr> </tbody> </table> <table border="1" style="margin-left: 20px;"> <caption>接收 (%Q)</caption> <thead> <tr> <th>PDO</th> <th>Tr.Ty...</th> <th>禁止 ...</th> <th>偶 ...</th> <th>符号</th> <th>拓扑地址</th> <th>%M...</th> <th>CO...</th> <th>索引</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> PDO 1</td> <td>255</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> <td>16#202</td> <td></td> </tr> <tr> <td>数字量 8 ...</td> <td></td> <td></td> <td></td> <td></td> <td>%QW\3.2\0.0...</td> <td>%MW559</td> <td></td> <td>6200:01</td> </tr> </tbody> </table> <table border="1" style="margin-left: 20px;"> <caption>变量</caption> <thead> <tr> <th>参数名称</th> <th>索引</th> </tr> </thead> <tbody> <tr><td>岛诊断 :...</td><td>4000:00</td></tr> <tr><td>岛诊断 :!...</td><td>4001:00</td></tr> <tr><td>已配置节点 1...</td><td>4002:01</td></tr> <tr><td>已配置节点 3...</td><td>4002:02</td></tr> <tr><td>已配置节点 4...</td><td>4002:03</td></tr> <tr><td>已配置节点 6...</td><td>4002:04</td></tr> <tr><td>已配置节点 8...</td><td>4002:05</td></tr> <tr><td>已配置节点 9...</td><td>4002:06</td></tr> <tr><td>已配置节点 1...</td><td>4002:07</td></tr> <tr><td>已配置节点 1...</td><td>4002:08</td></tr> <tr><td>可选节点 1...</td><td>4003:01</td></tr> <tr><td>可选节点 3...</td><td>4003:02</td></tr> <tr><td>可选节点 4...</td><td>4003:03</td></tr> <tr><td>可选节点 6...</td><td>4003:04</td></tr> <tr><td>可选节点 8...</td><td>4003:05</td></tr> <tr><td>可选节点 9...</td><td>4003:06</td></tr> <tr><td>可选节点 1...</td><td>4003:07</td></tr> <tr><td>可选节点 1...</td><td>4003:08</td></tr> <tr><td>有错误的节点 16...</td><td>4004:01</td></tr> <tr><td>有错误的节点 32...</td><td>4004:02</td></tr> <tr><td>有错误的节点 48...</td><td>4004:03</td></tr> <tr><td>有错误的节点 64...</td><td>4004:04</td></tr> </tbody> </table>	PDO	Tr.Ty...	禁止 ...	偶 ...	符号	拓扑地址	%M...	CO...	索引	<input checked="" type="checkbox"/> PDO 1	255	0					16#182		数字量 8 ...					%IW\3.2\0.0...	%MW184		6000:01	数字量 8 ...					%IW\3.2\0.0...	%MW185		6000:02	PDO	Tr.Ty...	禁止 ...	偶 ...	符号	拓扑地址	%M...	CO...	索引	<input checked="" type="checkbox"/> PDO 1	255	0					16#202		数字量 8 ...					%QW\3.2\0.0...	%MW559		6200:01	参数名称	索引	岛诊断 :...	4000:00	岛诊断 :!...	4001:00	已配置节点 1...	4002:01	已配置节点 3...	4002:02	已配置节点 4...	4002:03	已配置节点 6...	4002:04	已配置节点 8...	4002:05	已配置节点 9...	4002:06	已配置节点 1...	4002:07	已配置节点 1...	4002:08	可选节点 1...	4003:01	可选节点 3...	4003:02	可选节点 4...	4003:03	可选节点 6...	4003:04	可选节点 8...	4003:05	可选节点 9...	4003:06	可选节点 1...	4003:07	可选节点 1...	4003:08	有错误的节点 16...	4004:01	有错误的节点 32...	4004:02	有错误的节点 48...	4004:03	有错误的节点 64...	4004:04
PDO	Tr.Ty...	禁止 ...	偶 ...	符号	拓扑地址	%M...	CO...	索引																																																																																																						
<input checked="" type="checkbox"/> PDO 1	255	0					16#182																																																																																																							
数字量 8 ...					%IW\3.2\0.0...	%MW184		6000:01																																																																																																						
数字量 8 ...					%IW\3.2\0.0...	%MW185		6000:02																																																																																																						
PDO	Tr.Ty...	禁止 ...	偶 ...	符号	拓扑地址	%M...	CO...	索引																																																																																																						
<input checked="" type="checkbox"/> PDO 1	255	0					16#202																																																																																																							
数字量 8 ...					%QW\3.2\0.0...	%MW559		6200:01																																																																																																						
参数名称	索引																																																																																																													
岛诊断 :...	4000:00																																																																																																													
岛诊断 :!...	4001:00																																																																																																													
已配置节点 1...	4002:01																																																																																																													
已配置节点 3...	4002:02																																																																																																													
已配置节点 4...	4002:03																																																																																																													
已配置节点 6...	4002:04																																																																																																													
已配置节点 8...	4002:05																																																																																																													
已配置节点 9...	4002:06																																																																																																													
已配置节点 1...	4002:07																																																																																																													
已配置节点 1...	4002:08																																																																																																													
可选节点 1...	4003:01																																																																																																													
可选节点 3...	4003:02																																																																																																													
可选节点 4...	4003:03																																																																																																													
可选节点 6...	4003:04																																																																																																													
可选节点 8...	4003:05																																																																																																													
可选节点 9...	4003:06																																																																																																													
可选节点 1...	4003:07																																																																																																													
可选节点 1...	4003:08																																																																																																													
有错误的节点 16...	4004:01																																																																																																													
有错误的节点 32...	4004:02																																																																																																													
有错误的节点 48...	4004:03																																																																																																													
有错误的节点 64...	4004:04																																																																																																													
6	单击错误控制选项卡并将节点心跳生成器时间设置为 300 毫秒。																																																																																																													
7	单击工具栏上的 <input checked="" type="checkbox"/> 按钮以验证配置。																																																																																																													
8	<p>关闭该窗口。</p> <p>有关 STB 配置的更多信息，请参见 STB 岛配置。</p>																																																																																																													

声明 I/O 对象

下表显示加载使用 Advantys 配置软件定义的配置的过程：

步骤	操作
1	单击 CANopen 窗口中的 Lexium 模块图标，以打开 \3.55\0.0 : Lexium05 窗口。单击 Lexium05，然后单击 I/O 对象选项卡。
2	单击 I/O 对象前缀地址 %CH，然后单击更新网格按钮，通道地址会出现在 I/O 对象网格中。
3	单击行 %CH\3.55\0.0，然后在 I/O 对象创建窗口中的名称前缀区域输入通道名称，如 Lexium。
4	现在，单击不同的隐式 I/O 对象前缀地址，然后单击更新网格按钮查看隐式 I/O 对象的名称和地址。

地址	名称
1 %CH\3.55\0.0.0	Lexium
2 %ID\3.55\0.0.0	Lexium.Cap1Pos
3 %ID\3.55\0.0.0.2	Lexium.Cap2Pos
4 %ID\3.55\0.0.0.4	Lexium.param27_
5 %ID\3.55\0.0.0.6	Lexium.param27_
6 %ID\3.55\0.0.0.8	Lexium.p_actRAM
7 %ID\3.55\0.0.0.10	Lexium.position_a
8 %ID\3.55\0.0.0.12	Lexium.position_a
9 %ID\3.55\0.0.0.14	Lexium.Velocity_a
10 %IW\3.55\0.0.0.16	Lexium.IO_act
11 %IW\3.55\0.0.0.17	Lexium.ANA1_act
12 %IW\3.55\0.0.0.18	Lexium.ANA2_act
13 %IW\3.55\0.0.0.19	Lexium.Cap1Cou
14 %IW\3.55\0.0.0.20	Lexium.Cap2Cou
15 %IW\3.55\0.0.0.21	Lexium.actionStat
16 %IW\3.55\0.0.0.22	Lexium.Statuswo
17 %OD\3.55\0.0.0	Lexium.param27_
18 %OD\3.55\0.0.0.2	Lexium.param27_
19 %OD\3.55\0.0.0.4	Lexium.param35_
20 %OD\3.55\0.0.0.6	Lexium.param35_
21 %OD\3.55\0.0.0.8	Lexium.param35_
22 %OD\3.55\0.0.0.10	Lexium.GEARden
23 %OD\3.55\0.0.0.12	Lexium.GEARnu
24 %OD\3.55\0.0.0.14	Lexium.Target_P
25 %OD\3.55\0.0.0.16	Lexium.Profile_Ve
26 %OD\3.55\0.0.0.18	Lexium.Target_Ve
27 %QW\3.55\0.0.0.20	Lexium.Param_6
28 %QW\3.55\0.0.0.21	Lexium.JOGactiva
29 %QW\3.55\0.0.0.22	Lexium.CUR_L ta
30 %QW\3.55\0.0.0.23	Lexium.SPEEDn
31 %QW\3.55\0.0.0.24	Lexium.param35_
32 %QW\3.55\0.0.0.25	Lexium.Controlw

注意：重复相同的过程创建名为 BusMaster (%CH0.0.2) 的 CANopen I/O 对象。在 PLC 总线窗口中，双击 CANopen 端口，然后单击 CANopen 通讯头可访问 I/O 对象选项卡。

变量声明

概览

在程序的不同段中使用的所有变量必须进行声明。

未声明的变量不能在程序中使用。

注意：有关更多信息，请参见 [Unity Pro 联机帮助](#)（依次单击“？”、Unity、Unity Pro、操作模式和数据编辑器）。

声明变量的过程

下表显示了声明应用程序变量的过程：

步骤	操作
1	在项目浏览器 / 变量和 FB 实例中，双击基本变量。
2	在数据编辑器窗口中，选中“名称”列中的框，然后输入第一个变量的名称。
3	然后，选择此变量的类型。
4	声明了所有变量之后，可以关闭窗口。

应用程序使用的 变量

下表显示了在应用程序中使用的变量的详细信息：

变量	类型	定义
Action_Time	TIME	移动设备在每个位置的停止时间。
Configuration_Done	BOOL	完成 Lexium 配置。
Homing_Done	BOOL	完成原点定义。
index_subindex	DINT	WRITE_VAR 功能块的 CANopen 参数地址。
Lexium_Config_Step	INT	配置步骤（程序）。
Lexium_Disabling	INT	关闭命令。
Lexium_operation_enable	INT	用于启动 Lexium 驱动器的命令。
Mobile _at_Position_A	BOOL	移动设备位于 A 位置。
Mobile _at_Position_B	BOOL	移动设备位于 B 位置。
Mobile _at_Position_C	BOOL	移动设备位于 C 位置。
Mobile_at_start_position	BOOL	移动设备位于起始位置。
Mobile_in_Progress	BOOL	移动设备正在移动。
New_SetPoint	BOOL	开始下一个移动。
Operation_done	BOOL	移动设备操作完成。
Position_A	DINT	第一个定位值。
Position_B	DINT	第二个定位值。
Position_C	DINT	第三个定位值。
Ready_For_Stop	BOOL	移动设备移动到停止应用程序前指示的最后一个目标位置。然后它会回到起始位置。
Run	BOOL	序列的开始。
Sequence_Number	INT	移动设备给出的放置编号。
Start_Configuration	EBOOL	开始 Lexium 配置。
Stop	BOOL	移动设备停止序列并返回起始点。
Target_Reached	BOOL	到达目标位置。

下面的屏幕显示使用数据编辑器创建的应用程序变量：

名称	类型	地址	值	注释
Action_Time	TIME		t#3s	
Configuration_Done	BOOL...			
Homing_Done	BOOL	%IW\3.55\0.0.0.22.14		
index_subindex	DINT			
Lexium_Config_Step	INT			
Lexium_Disabling	INT		6	
Lexium_operation_enable	INT		15	
Mobile_at_Position_A	BOOL			
Mobile_at_Position_B	BOOL			
Mobile_at_Position_C	BOOL			
Mobile_at_Start_Position	BOOL			
Mobile_In_Progress	BOOL			
New_SetPoint	BOOL	%QW\3.55\0.0.0.25.4		
Operation_Done	BOOL			
Position_A	DINT		50000	
Position_B	DINT		100000	
Position_C	DINT		200000	
Ready_For_Stop	BOOL			
Run	BOOL			
Sequence_Number	INT			
Start_Configuration	EBOOL			
Stop	BOOL			
Target_Reached	BOOL	%IW\3.55\0.0.0.22.10		

注意：启动时，Lexium 05 位于就绪，可以启动状态（显示 rdy）。要驱动电机，Lexium 必须位于操作启用状态。要切换至此状态，总线命令会将 Lexium 控制字的后 4 位设置为 "1"（00001111（二进制）= 15（十进制））。要将 Lexium 05 切换为就绪，可以启动状态，总线命令会将 Lexium 控制字的第六和第七位设置为 "1"（00000110（二进制）= 6（十进制））。有关 Lexium 控制字的更多信息，请参考 Lexium 制造商手册。

在 SFC 中创建用于管理移动序列的程序

概览

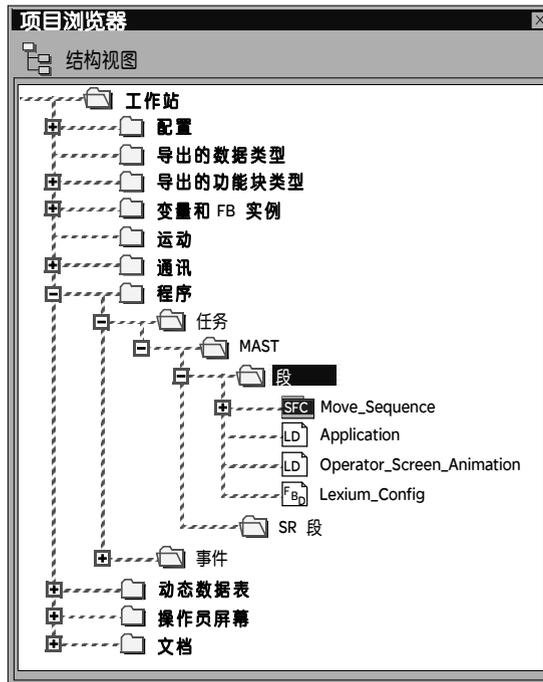
主程序是用 SFC (Grafcet) 编写的。Grafcet 步和转换的各个段是用 LD 编写的。此程序是在 MAST 任务中声明的，并取决于布尔型变量的状态。

SFC 语言的主要优点是，通过其图形化动态显示，可以实时监控应用程序的执行。

在 MAST 任务中声明了以下几个段：

- **Move_Sequence**（请参见 Move_Sequence 段示意图页 229）段，该段用 SFC 编写，用于描述操作模式。
- **Application**（请参见使用 LD 创建程序以用于应用程序的执行）段，该段用 LD 编写，用于执行移动设备操作延迟时间并复位定位起始位 New_Setpoint。
- **Operator_Screen_Animation**（请参见使用 LD 为操作员屏幕动态显示创建程序）段，该段用 LD 编写，用于动态显示操作员屏幕。
- **Lexium_Config**（请参见使用 ST 为 Lexium 配置创建程序）段，该段用 ST 编写，用于描述 Lexium 配置的不同步骤。

这些段在项目浏览器中显示如下：



注意：在应用程序中使用的 LD、SFC 和 FBD 类型的段必须在在线模式（请参见在标准模式下执行应用程序）下（且 PLC 处于 RUN 状态）动态显示。

注意：如果任务循环比 CANopen 主站循环快，输出将被覆盖。为避免发生这种情况，建议将任务循环时间设定为长于 CANopen 主站循环时间。

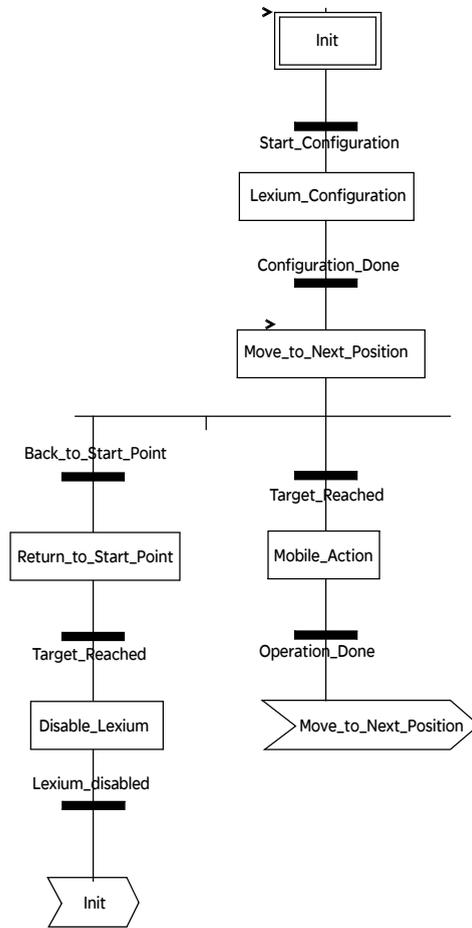
创建 SFC 段的过程

下表显示了为应用程序创建 SFC 段的过程。

步骤	操作
1	在项目浏览器 \ 程序 \ 任务中，双击 MAST。
2	右键单击段，然后选择新建段。为段指定名称（对于 SFC 段，为 Movement_sequence），然后选择 SFC 语言。
3	将显示段的名称，此时可以双击它以进行编辑。
4	<p>SFC 编辑工具出现在窗口中，您可以使用这些工具创建 Grafcet。</p> <p>例如，创建包含转换的步：</p> <ul style="list-style-type: none"> 若要创建该步，请单击 ，然后将其放在编辑器中。 若要创建转换，请单击 ，然后将其放在编辑器中（通常放在上一步之下）。

Move_Sequence 段示意图

下面的屏幕显示应用程序 Grafcet。此处没有定义任何条件：



有关 grafcet 中使用的操作和转换，请参见 [动作和转换](#)

注意：有关创建 SFC 段的更多信息，请参见 [Unity Pro 联机帮助](#)（依次单击“？”、Unity、Unity Pro、操作模式、编程和 SFC 编辑器）。

Move_Sequence 下表描述 Move_Sequence Grafcet 的各个步和转换：
段的描述

步 / 转换	描述
Init	这是初始状态。
Start_Configuration	在变量满足以下条件时，该转换为活动状态： <ul style="list-style-type: none"> ● Stop = 0 ● Run = 1
Lexium_Configuration	Lexium 05 已启用并且定义了 0 位置（使用 Lexium 的 Homing 功能）。
Configuration_done	在 Lexium 初始化后，该转换处于活动状态。
Move_to_next_position	下一个目标位置被加载到 Lexium 05 中。当此步激活时，序列编号会递增。
Target_reached	到达目标位置时，Lexium 5 将此变量设置为 "1"。
Mobile_action	移动设备位于目标位置并且正在执行操作。
Operation_done	移动设备操作结束时，此转换变为活动状态。
Back_to_start_point	在序列结束或者发出停止请求时，此转换变为活动状态。
Return_to_start_point	起始点定义在目标位置。
Disable_Lexium	Lexium 05 驱动器被禁用。
Lexium_disabled	禁用 Lexium 时，此转换有效。

注意：通过单击 SFC 段名称前面的 ，可以看到 SFC 的所有步、操作和转换。

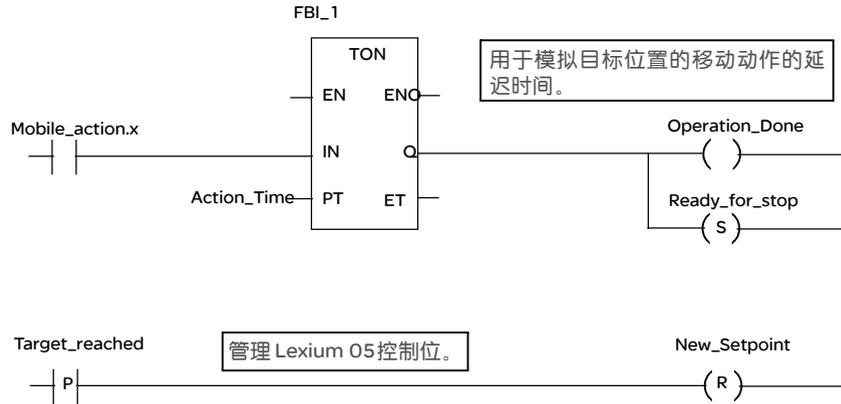
使用 LD 创建程序以用于应用程序的执行

概览

本节执行移动设备动作延迟时间并复位定位起始位 `New_Setpoint`。

Application 段示意图

下面的段是 MAST 任务的一部分。它没有定义任何条件，因此会永久地执行：



Application 段的 描述

- 第一行用于模拟移动设备位于目标位置的动作时间。Mobile_Action 步变为活动状态时，会触发 TON 定时器。到达 PT 时间时，TON 输出会切换为 "1"，验证转换变量 Operation_done，并设置 Ready_for_stop 变量。
- 第二行复位 Target_reached 正转换上的 New_Setpoint 变量。

创建 LD 段的过程 下表描述创建 Application 段的一部分的过程。

步骤	操作
1	在项目浏览器 \ 程序 \ 任务中，双击 MAST。
2	右键单击段，然后选择新建段。将此段命名为 Application，然后选择语言类型 LD。将打开编辑窗口。
3	要创建触点 Action_Mobile.x，请单击  ，然后将其放在编辑器中。双击此触点，然后输入步的名称（该名称应以后缀 ".x" 结尾，表示 SFC 段的步）。单击 "确定" 确认。
4	要使用 TON 功能块，必须将它实例化。在编辑器中单击鼠标右键，然后单击数据选择，再单击  。单击功能和功能块类型选项卡。单击库集，在列表中选择 TON 功能块，然后按 "确定" 确认并放置该功能块。要将 Action_Mobile.x 触点链接到 TON 功能块的输入，请将触点和输入水平对齐，单击  并将链接放置在触点和输入之间。

注意：有关创建 LD 段的更多信息，请参见 Unity Pro 联机帮助（依次单击 "?"、Unity、Unity Pro、操作模式、编程和 LD 编辑器）。

使用 LD 为操作员屏幕动态显示创建程序

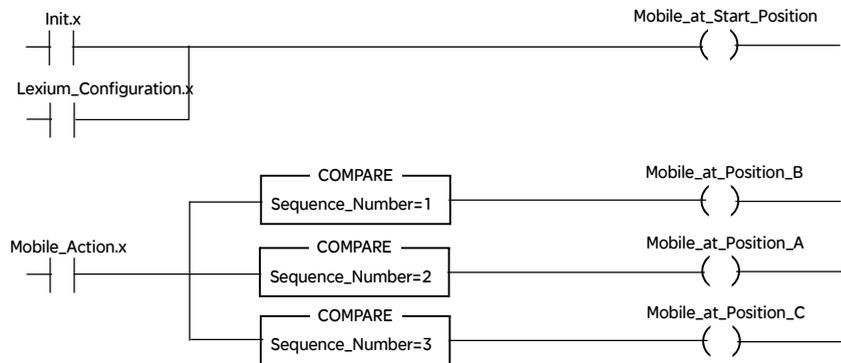
概览

本节动态显示操作员屏幕。

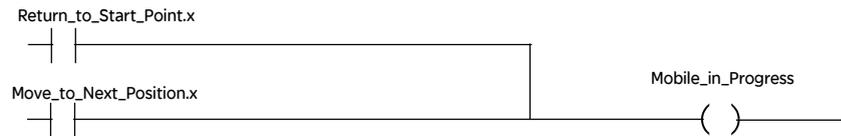
Operator_Screen _Animation 段示意图

下面的段是 MAST 任务的一部分。它没有定义任何条件，因此会永久地执行：

操作员屏幕动态显示。



管理操作员屏幕上的警告消息。



创建 LD 段的过程

有关创建 LD 段的信息，请参见创建 LD 段的过程。

使用 ST 为 Lexium 配置创建程序

概览

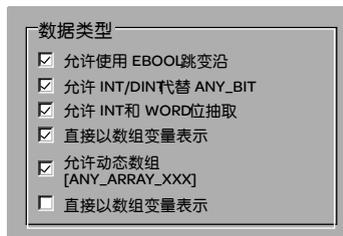
本节介绍 Lexium 配置的各个步。仅当 grafcet 中到达 Lexium_Configuration 步时本节内容才适用（请参见 Move_Sequence 段示意图）。

编程结构

编程结构如下：

步编号	步描述
0	Lexium 的启动命令。
10	如果 Lexium 处于 运行 状态，则使用 WRITE_VAR 功能切换为 回归 模式。
20	如果 WRITE_VAR 的结果是确定的，则转到步 30。
30	使用 WRITE_VAR 功能的 回归 方法定义。有关参考移动方法的更多信息，请参考 Lexium 用户手册。
40	如果 WRITE_VAR 的结果是确定的，则转到步 50。
50	启动 回归 方法。
60	回归完成。
70	Lexium 使用 WRITE_VAR 功能切换为 定位 模式。
80	如果 WRITE_VAR 的结果是确定的，则 Lexium 配置完成。

注意：要进行正确的变量声明，请单击**工具/项目设置/语言扩展**，然后选中"直接以数组变量表示"和"允许动态数组"。



ST 程序

下面是 ST 结构化文本语言编程的一个示例。专用段位于相同的主任务 (MAST) 中。

```
CASE Lexium_Config_Step OF
0: (* Lexium 处于 "准备好开启" 位置 *)
  IF (Lexium.statusword.0) THEN
    Lexium.controlword:=Lexium_operation_enable;
    Lexium_Config_Step := 10;
  END_IF;

10: (* Lexium 处于 "运行" 位置 *)
  IF (Lexium.statusword.2) THEN (* 操作模式: 回归 *)
    index_subindex:=16#00006060 (*CANopen 参数地址 *)
    %MW200:=6; (*Lexium 功能的定义: 回归 *)
    %MW162:=5; (* 超时 500 毫秒 *)
    %MW163:=1; (* 长度为 1 个字节 *)
    WRITE_VAR(ADDM('0.0.2.55'),'SDO',index_subindex,0,%MW200:1,
%MW160:4);
    Lexium_Config_Step:=20;
  END_IF;

20: (* 测试 WRITE_VAR 功能的结果 *)
  IF (NOT %MW160.0) THEN (* 测试活动位 *)
    IF (%MW161=0) THEN (* 交换正确 *)
      Lexium_Config_Step := 30;
    END_IF;
  END_IF;

30: (* 回归方法: 设置尺寸 *)
  index_subindex:=16#00006098
  %MW150:=35; (* 回归方法定义 *)
  %MW252:=5; (* 超时 500 毫秒 *)
  %MW253:=1; (* 长度为 1 个字节 *)
```

```
WRITE_VAR(ADDM('\0.0.2.55'),'SDO',index_subindex,0,%MW150:1,
%MW250:4);
Lexium_Config_Step:=40;

40: (* 测试 WRITE_VAR 功能的结果 *)
IF (NOT %MW250.0) THEN (* 测试活动位 *)
  IF (%MW251=0) THEN (* 交换正确 *)
    New_Setpoint:=0;
    Lexium_Config_Step := 50;
  END_IF;
END_IF;

50: (* 触发回归 *)
New_Setpoint :=1;
Lexium_Config_Step:=60;

60: (* 回归完成 *)
IF (Target_Reached) AND (Homing_Done) THEN
  New_Setpoint:=0;
  Lexium_Config_Step:=70;
END_IF;

70: (* 操作模式: 定位 *)
index_subindex:=16#00006060
%MW450:=1; (* 定位方法定义 *)
%MW352:=5; (* 超时 500 毫秒 *)
%MW353:=1; (* 长度为 1 个字节 *)
WRITE_VAR(ADDM('\0.0.2.55'),'SDO',index_subindex,0,%MW450:1,
%MW350:4);
Lexium_Config_Step:=80;
```

```
80: (* 测试 WRITE_VAR 功能的结果 *)
    IF (NOT %MW350.0) THEN (* 测试活动位 *)
        IF (%MW351=0) THEN (* 交换正确 *)
            Configuration_Done := 1;
        END_IF;
    END_IF;
END_CASE;
```

创建动态数据表

概览

动态数据表用于监控变量值以及修改和 / 或强制这些值。只有在变量和 FB 实例中声明的变量才能添加到动态数据表

注意：注：有关更多信息，请参见 Unity Pro 联机帮助（依次单击 "?"、Unity、Unity Pro、操作模式、调试和调整、查看和调整变量和动态数据表）。

创建动态数据表的过程

下表显示创建动态数据表的过程。

步骤	操作
1	在项目浏览器中，右键单击动态数据表，然后单击新建动态数据表。 将打开编辑窗口。
2	单击 "名称" 列中的第一个单元格，然后单击  按钮，添加所需变量。

为应用程序创建的 动态数据表

下面的屏幕显示应用程序使用的动态数据表：

名称	值	类型	注释
<input type="radio"/> Mobile_at_position_A		BOOL	
<input type="radio"/> Mobile_at_position_B		BOOL	
<input type="radio"/> Mobile_at_position_C		BOOL	
<input type="radio"/> Mobile_at_Start_Position		BOOL	
<input type="radio"/> Run		BOOL	
<input type="radio"/> Stop		BOOL	
<input type="radio"/> New_Setpoint		BOOL	
<input type="radio"/> Target_Reached		BOOL	
<input checked="" type="radio"/> Lexium.Target_position		DINT	
<input type="radio"/> Lexium.Position_actual_value		DINT	
<input checked="" type="radio"/> Lexium.controlword		INT	
<input checked="" type="radio"/> Lexium.Statusword		INT	
<input checked="" type="radio"/> BusMaster.COMM_STS		INT	
<input checked="" type="radio"/> BusMaster.CAN_STS		INT	
<input checked="" type="radio"/> BusMaster.EVT_STS		INT	

有关创建 Lexium 和 BusMaster 对象的更多信息，请参见声明 I/O 对象。

注意：只有在在线模式下，动态数据表才是动态的（显示变量值）。

注意：COMM_STS、CAN_STS 和 EVT_STS 字用于检查应用程序是否工作良好。有关更多信息，请参考 CANopen 用户手册。

注意：要快速填充动态数据表，请通过按住 Ctrl 键选择多个变量。

创建操作员屏幕

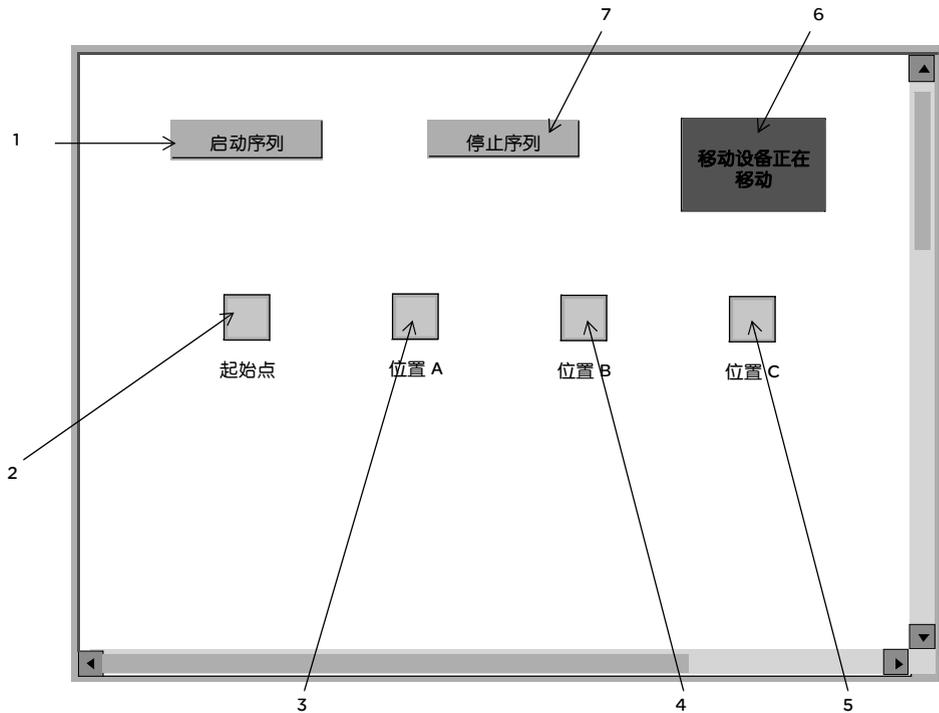
概览

操作员屏幕用于动态显示用符号表示应用程序的图形对象。这些对象可以来自 Unity Pro 库，也可以使用图形编辑器进行创建。

注意：有关更多信息，请参见 Unity Pro 联机帮助（依次单击“?”、Unity、Unity Pro、操作模式和操作员屏幕）。

操作员屏幕示意图

下图显示了应用程序的操作员屏幕：



关联变量如下表所示：

编号	描述	关联变量
1	启动按钮	Run
2	"起始点" 指示灯	Mobile_At_Start_Position
3	"位置 A" 指示灯	Mobile_At_Position_A
4	"位置 B" 指示灯	Mobile_At_Position_B
5	"位置 C" 指示灯	Mobile_At_Position_C
6	"移动设备正在移动" 指示灯	Mobile_in_Progress
7	停止按钮	Stop

注意：若要在在线模式下动态显示对象，必须单击 。通过单击此按钮，可以验证写入的内容。

创建操作员屏幕的过程

下表介绍创建启动按钮的过程。

步骤	操作
1	在项目浏览器中，右键单击操作员屏幕，然后单击新建屏幕。将出现操作员屏幕编辑器。
2	单击  并将新的按钮放置到操作员屏幕上。双击此按钮并在控制选项卡中，通过单击按钮  选择 Run 变量并按 "确定" 进行确认。然后，在文本区输入按钮的名称。

下表介绍插入并动态显示指示灯的过程。

步骤	操作
1	在 "工具" 菜单中，选择操作员屏幕库。双击显示单元，然后单击指示灯。从运行时屏幕选择动态绿灯并复制 (Ctrl+C) 粘贴 (Ctrl+V) 到操作员屏幕编辑器的绘图中。
2	此时，指示灯便显示在您的操作员屏幕中。选择指示灯并单击  。按 Enter，将打开对象属性窗口。选择动态显示选项卡，然后通过单击  (位于 %MW1.0 处) 输入相关的变量。 单击  并输入相同的变量。
3	单击 "应用" 和 "确定" 进行确认。

概览

本章主题

本章说明启动应用程序的过程，还描述了执行应用程序的各种方法

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
在标准模式下执行应用程序	244

在标准模式下执行应用程序

概览

要在标准模式下工作，需要将定义的变量与在 CANopen 总线上声明的设备的 PDO 地址关联。

注意：有关寻址的更多信息，请参见 Unity Pro 联机帮助（依次单击“?”、Unity、Unity Pro、语言参考、数据描述和数据实例）。

变量分配

下表显示对变量进行直接寻址的过程：

步骤	操作
1	在项目浏览器的“变量和 FB 实例”中，双击基本变量。
2	在“地址”列中，输入与变量关联的地址，格式为“\总线.节点\机架.模块.通道.数据”。 
3	对所有定位变量重复相同的过程。

已分配变量的示意图

下面的屏幕显示了应用程序变量分配：

数据编辑器

变量 | DDT类型 | 功能块 | DFB类型

过滤器 名称 * EDT DDT IODDT

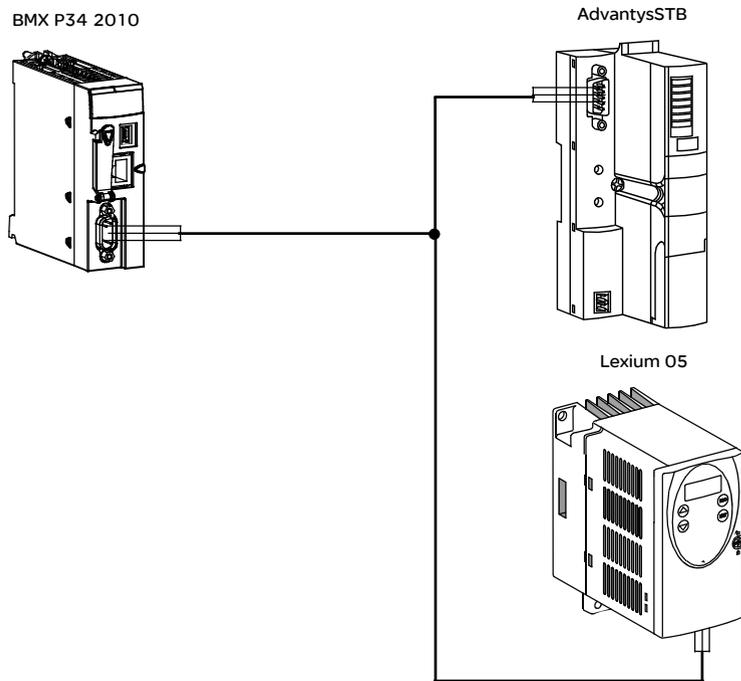
名称	类型	地址	值	注释
<input type="radio"/> Action_Time	TIME		t#3s	
<input type="radio"/> Configuration_Done	BOOL...			
<input type="radio"/> Homing_Done	BOOL			
<input type="radio"/> index_subindex	DINT			
<input type="radio"/> Lexium_Enable	INT		55	
<input type="radio"/> Lexium_operation_enable	INT		15	
<input type="radio"/> Mobile_at_Position_A	BOOL	%IW\3.2\0.0.0.167.7		
<input type="radio"/> Mobile_at_Position_B	BOOL	%IW\3.2\0.0.0.167.6		
<input type="radio"/> Mobile_at_Position_C	BOOL	%IW\3.2\0.0.0.167.5		
<input type="radio"/> Mobile_at_Start_Position	BOOL	%IW\3.2\0.0.0.167.4		
<input type="radio"/> Mobile_In_Progress	BOOL			
<input type="radio"/> New_SetPoint	BOOL	%QW\3.1\0.0.0.25.4		
<input type="radio"/> Operation_Done	BOOL			
<input type="radio"/> Position_A	DINT		1000	
<input type="radio"/> Position_B	DINT		2000	
<input type="radio"/> Position_C	DINT		4000	
<input type="radio"/> Run	BOOL			
<input type="radio"/> Sequence_Number	INT			
<input type="radio"/> Start_Configuration	EBOOL			
<input type="radio"/> Stop	BOOL			
<input type="radio"/> Target_Reached	BOOL	%IW\3.1\0.0.0.16.10		
<input type="radio"/>				

变量分配描述

- 前四个布尔变量分配给 STB DDI 3420 模块的四个离散量输入。
- New_Setpoint 分配给 Lexium 05 控制位。此位的正转换会触发新的定位。
- Target_Reached 分配给 Lexium 05 状态位，此状态位在达到目标时设置为 "1"。

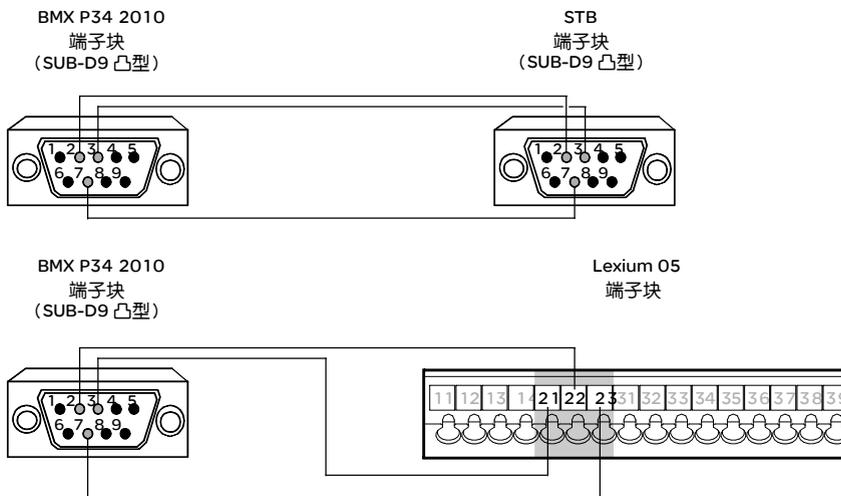
CANopen
总线接线

CANopen 总线按如下方式连接:



注意: Lexium 05 位于 CANopen 总线的末端。将终端电阻器 CAN 开关设置为 "1"。

连接器的引脚分配如下所示：



BMX P34 2010 端子块描述：

引脚号	符号	描述
1	-	保留
2	CAN_L	CAN_L 总线 (低)
3	CAN_GND	CAN 接地
4	-	保留
5	保留	可选 CAN 保护
6	(GND)	可选接地
7	CAN_H	CAN_H 总线 (高)
8	-	保留
9	保留	CAN 外部正电源 (可选)

STB 端子块描述:

引脚号	符号	描述
1	-	保留
2	CAN_L	CAN_L 总线 (低)
3	CAN_GND	CAN 接地
4	-	保留
5	(CAN_SHLD)	可选 CAN 保护
6	(GND)	可选接地
7	CAN_H	CAN_H 总线 (高)
8	-	保留
9	-	保留

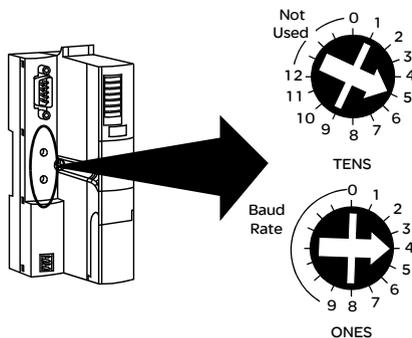
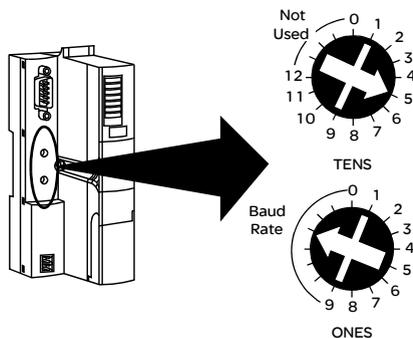
Lexium 05 端子块描述:

引脚号	符号	描述
21	CAN_GND	CAN 接地
22	CAN_L	CAN_L 总线 (低)
23	CAN_H	CAN_H 总线 (高)

Advantys STB 配置

下表显示 Lexium 05 的配置过程：

步骤	操作
1	关闭 STB 。
2	使用旋转开关（位于 CANopen NIM 的前面）配置波特率。旋转开关的定位如下（5 = 500 kbits/s）：
3	启动然后关闭 STB 。
4	使用旋转开关配置 STB 的地址。例如，如果设备的节点编号为 "54"，那么旋转开关定位如下：
5	启动 STB 并按住 STB NCO 模块上的复位按钮 5 秒钟。
6	这会自动配置 STB 。



Lexium 配置

下表显示 Lexium 05 的配置过程：

步骤	操作
1	启动 Lexium 05。界面上会显示 RDY。
2	按 Enter。
3	按向下箭头键，直到显示 COM-。然后按 Enter。
4	按向下箭头键，直到显示 CoAD（CANopen 地址）。然后按 Enter。
5	使用箭头键配置节点编号。然后按 Esc。
6	按向下箭头键，直到显示 CoBD（CANopen 波特率）。然后按 Enter。
7	使用箭头键配置波特率 (500)。然后按 Esc。
8	按 Esc，直到显示 RDY。

概览

概述

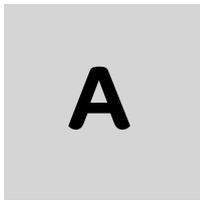
这些附录包含了对应用程序编程很有用的信息。

本附录包含了哪些内容？

本附录包含了以下章节：

章	章节标题	文件集
A	CANopen 主站本地对象字典条目	253
B	PDO 和 STB 变量之间的关系	267
C	动作和转换	271

CANopen 主站本地对象字典条目



概览

本章主题

本章包含 CANopen 主站的本地对象字典条目。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
符合配置文件 DS301 的对象字典条目	254
符合配置文件 DS302 的对象字典条目	259
中型制造商专用对象字典条目	261

符合配置文件 DS301 的对象字典条目

对象字典条目 下表介绍符合配置文件 DS301 的对象字典条目。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
1000		设备类型	VAR	Unsigned32	0x000F 0191
1001		错误寄存器	VAR	Unsigned8	
1005		COB-ID SYNC	VAR	Unsigned32	
1006		通讯循环周期	VAR	Unsigned32	
1007		同步窗口长度	VAR	Unsigned32	
1008		制造商设备名称	VAR	String	BMX CPU 20x0
1009		制造商硬件版本	VAR	String	MIDRANGE BASIC
100A		制造商软件版本	VAR	String	COMM_FW_01_xx
1012		COB-ID 时标消息	VAR	Unsigned32	
1016		消费者心跳时间	ARRAY		
	0	条目数: 64		Unsigned8	
	1	消费者心跳时间		Unsigned32	
	...			Unsigned32	
	64			Unsigned32	
1017		生产者心跳时间	VAR	Unsigned16	
1018		标识对象	RECORD		
	0	条目数		Unsigned8	4
	1	供应商 ID		Unsigned32	0x0600 005A
	2	产品代码		Unsigned32	0x3300 FFFF
	3	修订号		Unsigned32	0xyyyy xxxx
	4	序列号		Unsigned32	0x0
				Unsigned32	
1020		验证配置	ARRAY		
	0	条目数: 2		Unsigned8	
	1	配置日期		Unsigned32	
	2	配置时间		Unsigned32	
1200		1. 服务器 SDO	RECORD		
	0	条目数		Unsigned8	

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
	1	COB-ID 客户端 -> 服务器 (Rx)		Unsigned32	600H + 节点 ID
	2	COB-ID 服务器 -> 客户端 (Tx)		Unsigned32	580H + 节点 ID
1280		1. 客户端 SDO	RECORD		
	0	条目数		Unsigned8	
	1	COB-ID 客户端 -> 服务器 (Rx)		Unsigned32	
	2	COB-ID 服务器 -> 客户端 (Tx)		Unsigned32	
	3	服务器 SDO 的节点 ID		Unsigned8	
1281		2. 客户端 SDO	RECORD		
	0	条目数		Unsigned8	
	1	COB-ID 客户端 -> 服务器 (Rx)		Unsigned32	
	2	COB-ID 服务器 -> 客户端 (Tx)		Unsigned32	
	3	服务器 SDO 的节点 ID		Unsigned8	
1282		3. 客户端 SDO	RECORD		
	0	条目数		Unsigned8	
	1	COB-ID 客户端 -> 服务器 (Rx)		Unsigned32	
	2	COB-ID 服务器 -> 客户端 (Tx)		Unsigned32	
	3	服务器 SDO 的节点 ID		Unsigned8	
1400		1. 接收 PDO	RECORD		
	0	支持的最大子索引		Unsigned8	
	1	PDO 使用的 COB-ID		Unsigned32	
	2	传输类型		Unsigned8	
	3			Unsigned16	
	4			Unsigned8	
	5	事件定时器		Unsigned16	
1401		2. 接收 PDO	RECORD		
	0	支持的最大子索引		Unsigned8	
	1	PDO 使用的 COB-ID		Unsigned32	
	2	传输类型		Unsigned8	
	3			Unsigned16	
	4			Unsigned8	
	5	事件定时器		Unsigned16	

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
.....					
14FF		256.接收 PDO	RECORD		
	0	支持的最大子索引		Unsigned8	
	1	PDO 使用的 COB-ID		Unsigned32	
	2	传输类型		Unsigned8	
	3			Unsigned16	
	4			Unsigned8	
	5	事件定时器		Unsigned16	
1600		1.接收 PDO 映射			
	0	PDO 中映射的应用程序对象数		Unsigned8	取决于应用程序的 PDO 映射
	1	将被映射的 1. 应用程序对象的 PDO 映射		Unsigned32	索引 (16 位) 子索引 (8 位) 长度 (8 位)
	2	2. 应用程序对象的 PDO 映射		Unsigned32	
				
	8	8. 应用程序对象的 PDO 映射		Unsigned32	
1601		2.接收 PDO 映射			
	0	PDO 中映射的应用程序对象数		Unsigned8	取决于应用程序的 PDO 映射
	1	将被映射的 1. 应用程序对象的 PDO 映射		Unsigned32	索引 (16 位) 子索引 (8 位) 长度 (8 位)
	2	2. 应用程序对象的 PDO 映射		Unsigned32	
				
	8	8. 应用程序对象的 PDO 映射		Unsigned32	
.....					
16FF		256.接收 PDO 映射			
	0	PDO 中映射的应用程序对象数		Unsigned8	取决于应用程序的 PDO 映射
	1	将被映射的 1. 应用程序对象的 PDO 映射		Unsigned32	索引 (16 位) 子索引 (8 位) 长度 (8 位)
	2	2. 应用程序对象的 PDO 映射		Unsigned32	
				
	8	8. 应用程序对象的 PDO 映射		Unsigned32	

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
1800		1. 传输 PDO	RECORD		
	0	支持的最大子索引		Unsigned8	
	1	PDO 使用的 COB-ID		Unsigned32	
	2	传输类型		Unsigned8	
	3	禁止时间		Unsigned16	
	4	保留		Unsigned8	
	5	事件定时器		Unsigned16	
1801		2. 传输 PDO	RECORD		
	0	支持的最大子索引		Unsigned8	
	1	PDO 使用的 COB-ID		Unsigned32	
	2	传输类型		Unsigned8	
	3	禁止时间		Unsigned16	
	4	保留		Unsigned8	
	5	事件定时器		Unsigned16	
.....					
18FF		256. 传输 PDO	RECORD		
	0	支持的最大子索引		Unsigned8	
	1	PDO 使用的 COB-ID		Unsigned32	
	2	传输类型		Unsigned8	
	3	禁止时间		Unsigned16	
	4	保留		Unsigned8	
	5	事件定时器		Unsigned16	
1A00		1. 传输 PDO 映射			
	0	PDO 中映射的应用程序对象数		Unsigned8	取决于应用程序的 PDO 映射
	1	将被映射的 1. 应用程序对象的 PDO 映射		Unsigned32	索引 (16 位) 子索引 (8 位) 长度 (8 位)
	2	2. 应用程序对象的 PDO 映射		Unsigned32	
				
	8	8. 应用程序对象的 PDO 映射		Unsigned32	
1A01		2. 传输 PDO 映射			

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
	0	PDO 中映射的应用程序对象数		Unsigned8	取决于应用程序的 PDO 映射
	1	将被映射的 1. 应用程序对象的 PDO 映射		Unsigned32	索引 (16 位) 子索引 (8 位) 长度 (8 位)
	2	2. 应用程序对象的 PDO 映射		Unsigned32	
				
	8	8. 应用程序对象的 PDO 映射		Unsigned32	
.....					
1AFF		256. 传输 PDO 映射			
	0	PDO 中映射的应用程序对象数		Unsigned8	取决于应用程序的 PDO 映射
	1	将被映射的 1. 应用程序对象的 PDO 映射		Unsigned32	索引 (16 位) 子索引 (8 位) 长度 (8 位)
	2	2. 应用程序对象的 PDO 映射		Unsigned32	
				
	8	8. 应用程序对象的 PDO 映射		Unsigned32	

符合配置文件 DS302 的对象字典条目

对象字典条目 下表介绍符合配置文件 DS302 的对象字典条目。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
1F22		简短 DCF	ARRAY		
	0	条目数	VAR	Unsigned8	
	1	节点 ID 为 1 的设备	VAR	DOMAIN	
	...				
	127	节点 ID 为 127 的设备		DOMAIN	
1F26		预计配置日期	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 为 1 的设备		Unsigned32	
	...				
	127	节点 ID 为 127 的设备		Unsigned32	
1F27		预计配置时间	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 为 1 的设备		Unsigned32	
	...				
	127	节点 ID 为 127 的设备		Unsigned32	
1F80		NMT 启动	VAR	Unsigned32	
1F81	...	从站分配	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 为 1 的设备		Unsigned32	
	...				
	127	节点 ID 为 127 的设备		Unsigned32	
1F82	...	请求 NMT	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 1 的请求 NMT		Unsigned8	
	...				
	128	所有节点的请求 NMT		Unsigned8	
1F84	...	设备类型标识	ARRAY		
	0	条目数		Unsigned8	

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
	1	节点 ID 为 1 的设备		Unsigned32	
	...				
	127	节点 ID 为 127 的设备		Unsigned32	
1F85	...	供应商标识	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 为 1 的设备		Unsigned32	
	...				
	127	节点 ID 为 127 的设备		Unsigned32	
1F86	...	产品代码	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 为 1 的设备		Unsigned32	
	...				
	127	节点 ID 为 127 的设备		Unsigned32	
1F87	...	修订号	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 为 1 的设备		Unsigned32	
	...				
	127	节点 ID 为 127 的设备		Unsigned32	

中型制造商专用对象字典条目

项目数据 下表介绍对象条目 2010（项目数据）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
2010		项目数据	RECORD		
	0	条目数		Unsigned8	
	1	当前字节长度		Unsigned16	只读访问 由主站管理器更新
	2	项目数据域		DOMAIN	

**CANopen 主站
定时控制** 下表介绍对象条目 2100（CANopen 主站定时控制）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
2100		CANopen 主站定时控制	ARRAY		
	0	条目数		Unsigned8	
	1	一个循环中传输的最大 TPDO 数		Unsigned8	
	2	一个循环中的最大高优先级接收队列访问数（RPDO、EMCY）		Unsigned8	
	3	一个循环中的最大低优先级接收队列访问数（SDO、心跳/防护）		Unsigned8	

**CANopen
主站状态**

下表介绍对象条目 4100 (CANopen 主站状态)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4100		CANopen 主站状态	ARRAY		
	0	条目数		Unsigned8	
	1	全局事件		Unsigned16	
	2	通讯状态		Unsigned8	
	3	通讯诊断		Unsigned8	
	4	配置位		Unsigned16	
	5	LED 控制		Unsigned16	
	6	最短循环时间		Unsigned8	
	7	最长循环时间		Unsigned8	

Nd_asg

下表介绍对象条目 4101 (Nd_asg)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4101		Nd_asg	ARRAY		
	0	条目数		Unsigned8	
	1	Nd_asg[0,1,2,3		Unsigned32	
	2	Nd_asg[4,5,6,7		Unsigned32	
	3	Nd_asg[8,9,10,11		Unsigned32	
	4	Nd_asg[12,13,14,15		Unsigned32	

Nd_cfg

下表介绍对象条目 4102 (Nd_cfg)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4102		Nd_cfg	ARRAY		
	0	条目数		Unsigned8	
	1	Nd_cfg[0,1,2,3		Unsigned32	
	2	Nd_cfg[4,5,6,7		Unsigned32	
	3	Nd_cfg[8,9,10,11		Unsigned32	
	4	Nd_cfg[12,13,14,15		Unsigned32	

Nd_asf 下表介绍对象条目 4103 (Nd_asf)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4103		Nd_asf	ARRAY		
	0	条目数		Unsigned8	
	1	Nd_asf[0,1,2,3		Unsigned32	
	2	Nd_asf[4,5,6,7		Unsigned32	
	3	Nd_asf[8,9,10,11		Unsigned32	
	4	Nd_asf[12,13,14,15		Unsigned32	

Nd_oper 下表介绍对象条目 4104 (Nd_oper)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4104		Nd_oper	ARRAY		
	0	条目数		Unsigned8	
	1	Nd_oper[0,1,2,3		Unsigned32	
	2	Nd_oper[4,5,6,7		Unsigned32	
	3	Nd_oper[8,9,10,11		Unsigned32	
	4	Nd_oper[12,13,14,15		Unsigned32	

Nd_stop 下表介绍对象条目 4105 (Nd_stop)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4105		Nd_stop	ARRAY		
	0	条目数		Unsigned8	
	1	Nd_stop[0,1,2,3		Unsigned32	
	2	Nd_stop[4,5,6,7		Unsigned32	
	3	Nd_stop[8,9,10,11		Unsigned32	
	4	Nd_stop[12,13,14,15		Unsigned32	

Nd_preop 下表介绍对象条目 4106 (Nd_preop)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4106		Nd_preop	ARRAY		
	0	条目数		Unsigned8	
	1	Nd_preop[0,1,2,3		Unsigned32	
	2	Nd_preop[4,5,6,7		Unsigned32	
	3	Nd_preop[8,9,10,11		Unsigned32	
	4	Nd_preop[12,13,14,15		Unsigned32	

Nd_err 下表介绍对象条目 4107 (Nd_err)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4107		Nd_err	ARRAY		
	0	条目数		Unsigned8	
	1	Nd_err[0,1,2,3		Unsigned32	
	2	Nd_err[4,5,6,7		Unsigned32	
	3	Nd_err[8,9,10,11		Unsigned32	
	4	Nd_err[12,13,14,15		Unsigned32	

节点错误计数 下表介绍对象条目 4110 (节点错误计数)。

索引 (十六进制)	子索引	描述	对象类型	数据类型	注释
4110		节点错误计数	ARRAY		
	0	条目数		Unsigned8	
	1	接收到节点号 1 的紧急消息数		Unsigned8	
	...				
	127	接收到节点号 127 的紧急消息数		Unsigned8	

错误代码专用错误计数器 下表介绍对象条目 4111 到 4117（错误代码专用错误计数器）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
4111		一般错误计数（代码 10xxH）	VAR	Unsigned8	
4112		设备硬件错误计数（代码 50xxH）	VAR	Unsigned8	
4113		设备软件错误计数（代码 60xxH）	VAR	Unsigned8	
4114		通讯错误计数（代码 81xxH）	VAR	Unsigned8	
4115		协议错误计数（代码 82xxH）	VAR	Unsigned8	
4116		外部错误计数（代码 90xxH）	VAR	Unsigned8	
4117		设备专用（代码 FFxxH）	VAR	Unsigned8	

紧急历史记录 下表介绍对象条目 4118（紧急历史记录）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
4118		紧急历史记录	ARRAY		
	0	条目数		Unsigned8	
	1	节点号 1 的紧急历史记录		Domain	
	...				
	127	节点号 127 的紧急历史记录		Domain	

输入过程映像 下表介绍对象条目 4200（输入过程映像）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
4200		输入过程映像	RECORD		
	0	条目数		Unsigned8	
	1	当前字节长度		Unsigned16	只读访问 由主站管理器更新

输出过程映像 下表介绍对象条目 4201（输出过程映像）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
4201		输出过程映像	RECORD		
	0	条目数		Unsigned8	
	1	当前字节长度		Unsigned16	只读访问 由主站管理器更新

其他主站信息 下表介绍对象条目 4205（其他主站信息）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
4205		其他主站信息	RECORD		
	0	条目数		Unsigned8	ro
	1	耦合器 (CPU) 类型		Unsigned8	rw
	2	CAN 波特率表索引		Unsigned8	ro
	3	使用的最高节点 ID		Unsigned8	ro
	4	使用的 RxPDO 数		Unsigned16	ro
	5	使用的 TxPDO 数		Unsigned16	ro
	6	映射的对象 PI 输入数		Unsigned16	ro
	7	映射的对象 PI 输出数		Unsigned16	ro
	8	简短 DCF 所用的字节数		Unsigned8	ro
	9	简短 DCF 缓冲区的字节大小		Unsigned16	ro
	10	配置签名		Unsigned16	rw
	11	控制		Unsigned16	rw

访问类型：ro（只读），rw（读/写）。

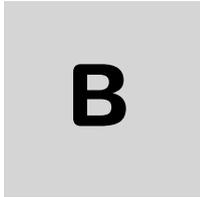
其他从站分配 下表介绍对象条目 4250（其他主站分配）。

索引（十六进制）	子索引	描述	对象类型	数据类型	注释
4250		其他从站分配	ARRAY		
	0	条目数		Unsigned8	
	1	节点 ID 1 的启动行为		Unsigned8	
	...				
	127	节点 ID 127 的启动行为		Unsigned8	

位 0 = 0: 符合 DS-302 的启动。

位 1 = 1: 启动不会覆盖配置参数。

PDO 和 STB 变量之间的关系



B

概览

本章主题

本章描述 Unity Pro 上的 PDO 配置以及 Advantys 配置软件上的 STB 变量之间的等价关系。

本章包含了哪些内容?

本章包含了以下主题:

主题	文件集
STB 岛配置	268

STB 岛配置

概览

使用 COBid，可以在 PDO 和 STB 变量之间建立链接。

STB 岛可以通过以下软件进行配置：

- Advantys 配置软件 (STB NCO 2212)。
- Unity Pro 软件 (STB NCO 2212 和 NCO 1010)。

使用 Advantys 配置岛进行配置

使用 Advantys 配置软件配置 STB 岛的过程如下。该过程仅涉及 STB NCO 2212 模块：

步骤	操作																																																																																																																																																																																																																																														
1	在 Advantys 配置软件（2.2.0.2 版或更高版本）中，创建一个新岛。																																																																																																																																																																																																																																														
2	选择 STBNCO2212 网络接口模块																																																																																																																																																																																																																																														
3	选择将在应用程序中使用的模块																																																																																																																																																																																																																																														
4	<p>在菜单中单击岛和 I/O 映像概述</p>  <table border="1" data-bbox="386 748 1114 1179"> <thead> <tr> <th>对象</th> <th>15</th> <th>14</th> <th>13</th> <th>12</th> <th>11</th> <th>10</th> <th>9</th> <th>8</th> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>6100:01</td> <td>9</td> </tr> <tr> <td>6000:01</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>2</td> <td>2</td> <td>2</td> <td>2</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>6000:02</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>4</td> <td>4</td> <td>4</td> <td>4</td> <td>3</td> <td>3</td> <td>3</td> <td>3</td> </tr> <tr> <td>6000:03</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>6</td> <td>6</td> <td>6</td> <td>6</td> <td>5</td> <td>5</td> <td>5</td> <td>5</td> </tr> <tr> <td>6000:04</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>8</td> <td>8</td> <td>8</td> <td>8</td> <td>7</td> <td>7</td> <td>7</td> <td>7</td> </tr> <tr> <td>6000:05</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> </tr> <tr> <td>6000:06</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> <td>10</td> </tr> <tr> <td>6401:01</td> <td>10</td> </tr> <tr> <td>6401:02</td> <td>10</td> </tr> <tr> <td>2600:00</td> <td>32</td> </tr> <tr> <td></td> <td>32</td> </tr> <tr> <td>2601:00</td> <td>32</td> </tr> <tr> <td></td> <td>32</td> </tr> </tbody> </table> <p>此窗口显示离线时的 I/O 映像概述。变量索引与 Unity Pro 软件中相同。可在此窗口中方便快捷地找到 PDO 的内容。</p>	对象	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	6100:01	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	6000:01	-	-	-	-	-	-	-	-	2	2	2	2	1	1	1	1	6000:02	-	-	-	-	-	-	-	-	4	4	4	4	3	3	3	3	6000:03	-	-	-	-	-	-	-	-	6	6	6	6	5	5	5	5	6000:04	-	-	-	-	-	-	-	-	8	8	8	8	7	7	7	7	6000:05	-	-	-	-	-	-	-	-	10	10	10	10	10	10	10	10	6000:06	-	-	-	-	-	-	-	-	10	10	10	10	10	10	10	10	6401:01	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	6401:02	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	2600:00	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32		32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	2601:00	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32		32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
对象	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																															
6100:01	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9																																																																																																																																																																																																																															
6000:01	-	-	-	-	-	-	-	-	2	2	2	2	1	1	1	1																																																																																																																																																																																																																															
6000:02	-	-	-	-	-	-	-	-	4	4	4	4	3	3	3	3																																																																																																																																																																																																																															
6000:03	-	-	-	-	-	-	-	-	6	6	6	6	5	5	5	5																																																																																																																																																																																																																															
6000:04	-	-	-	-	-	-	-	-	8	8	8	8	7	7	7	7																																																																																																																																																																																																																															
6000:05	-	-	-	-	-	-	-	-	10	10	10	10	10	10	10	10																																																																																																																																																																																																																															
6000:06	-	-	-	-	-	-	-	-	10	10	10	10	10	10	10	10																																																																																																																																																																																																																															
6401:01	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10																																																																																																																																																																																																																															
6401:02	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10																																																																																																																																																																																																																															
2600:00	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32																																																																																																																																																																																																																															
	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32																																																																																																																																																																																																																															
2601:00	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32																																																																																																																																																																																																																															
	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32																																																																																																																																																																																																																															
5	配置完成后，单击“文件”/“导出”以 DCF 格式导出岛，供以后导入到 Unity Pro 中。																																																																																																																																																																																																																																														

 **警告****意外操作的风险**

配置 STB Island 期间，由 Advantys 生成的符号文件 *.xsy 不得用于 Unity Pro 中。During an 将 *.xsy 文件从 Advantys 导入 Unity Pro for CAN 期间，不支持开放设备。

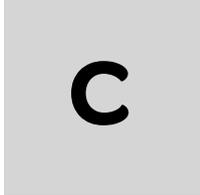
在 PDO 表中指定的 %MW 对象与在 CAN 开放头的配置中定义的对象不属于同一范围。

如果不遵守这个警告将会导致死亡，严重伤害，或设备损坏。

使用 Unity Pro 使用 Unity Pro 软件配置 STB 岛的过程如下：
软件进行配置

步骤	操作
1	在项目浏览器中，双击配置，然后双击 3 : CANopen。
2	在 CANopen 窗口中，双击 Advantys STB 图标。STB 配置窗口打开。
3	在功能区域，选择高级。 
4	单击 PDO 选项卡查看 PDO 配置、变量及其拓扑地址。 
5	在窗口的右侧有一个列表，其中列出了映射或未映射的 STB 变量。其中的索引与 Advantys 配置软件中相同。可以方便快捷地找到变量。 将变量拖放到右边的 PDO 以配置 STB 岛。

动作和转换



概览

本章主题

本章包含 grafcet 中所用的动作和转换（请参见在 SFC 中创建用于管理移动序列的程序）。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
转换	272
操作	273

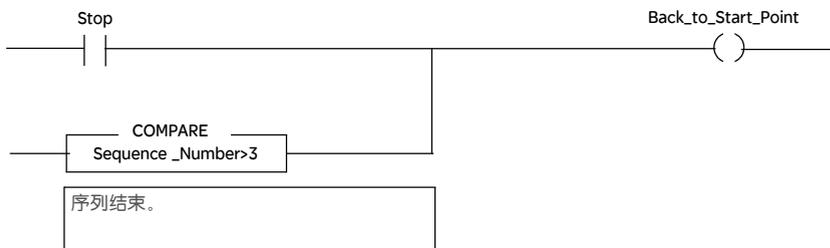
转换

概览

下面的用 LD 编写的任务可在 grafcet 的不同转换中使用。

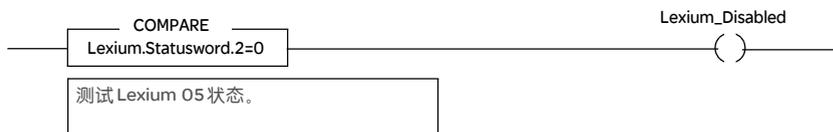
Back_to_Start_Point 转换

与 **Back_to_Start_Point** 转换关联的操作如下：



Lexium_Disabled 转换

与 **Lexium_Disabled** 转换关联的操作如下：



操作

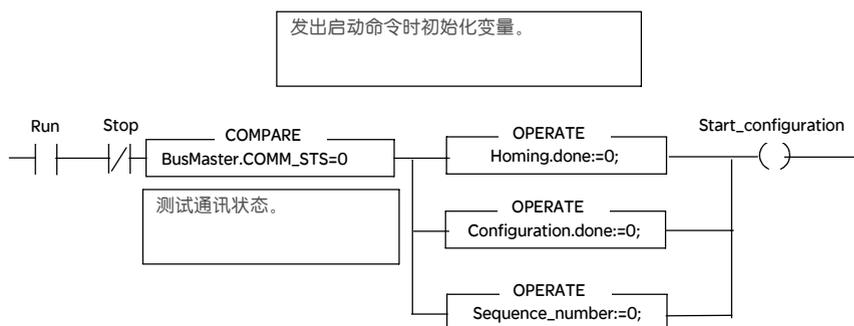
概览

下面的用 LD 和 ST 编写的任务可在 grafcet 的不同步骤中使用。

注意: 要使用以下操作, 请在工具/ 项目设置/ 语言扩展中选择允许动态数组和直接以数组变量表示选项。

Init 步

与 Init 步关联的操作如下:



Move_to_Next_Position 步

有两个操作与 **Move_to_Next_Position** 步关联。

第一个操作如下：

```
(* 定义目标位置 *)
CASE Sequence_number OF
  1: Lexium.Target_Position:=Position_B;
  2: Lexium.Target_Position:=Position_A;
  3: Lexium.Target_Position:=Position_C;
END_CASE;
IF (Sequence_number<4) AND NOT (Stop) THEN
(* 开始新的定位 *)
  New_SetPoint:=1;
  Ready_for_Stop:=0;
END IF;
```

第二个操作如下：

```
(* 在开始新的移动前递增 *)
INC(Sequence_Number);
```

注意：对于递增操作，限定符必须定位于 P（上升沿）上。

Return_to_Start_Point 步

与 **Return_to_Start_Point** 步关联的操作如下：

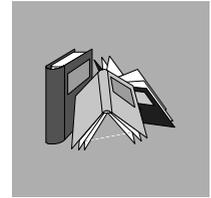
```
(* 加载目标位置 *)
Lexium.Target_Position:=0;
(* 开始新的定位 *)
New_Setpoint:=1;
```

Disable_Lexium

与 **Disable_Lexium** 步关联的操作如下：

```
(* Lexium 电压禁用 *)
Lexium.Controlword:=Lexium_disabling;
```

Modicon M340 的 CANopen 术语



ADVANTYS	用于 PLC 岛的 Schneider CANopen 配置工具。
BOOL	布尔型。
CAN	控制器区域网络：现场总线，最初为汽车应用领域开发，现在用于很多领域。
COB	通讯对象 ：CANopen 总线上的传输单元。COB 由唯一标识符 [0, 2047] 进行标识，标识符采用 11 位编码。一个 COB 最多包含 8 个数据字节。COB 的传输优先级由其标识符确定。标识符越低，相关 COB 的优先级越高。
COB-ID	COB 标识符 ：CANopen 网络上的 COB 的唯一标识符。标识符确定 COB 的优先级。
CSDO	SDO 客户端。
CiA	自动化中的 CAN：CAN 设备的用户和制造商的国际组织。
DINT	双精度整数 ：32 位字。
DS	草案标准 ：CIA 组织所创建的规范文档。
Discrete Module (离散量模块)	Tout Ou Rien。
EBOOL	带跳变沿检测和强制可能性的布尔类型。
EDM	多语言电子数据表 ：EDS 文件的扩展版。扩展包括对欧洲多语言的支持和对设备物理特征的描述。

EDS **电子数据表:** CANopen 设备配置文件的描述, 描述按照 DSP306 CiA 规格进行了规格化。

EMCY **紧急:** 由内部错误 / 故障生成的触发事件。此对象随每个新错误传输, 因为错误代码是独立的机制。

ETS **空终端支持:** 其他信息存储在 PLC 应用程序中以供上载。

HEALTH
(运行状况) **值为 1 的位:** 模式正常工作
值为 0 的位:

- 配置错误, 或者
- 模块已配置, 但找不到, 或者
- 模块已配置, 但与现有模块地址相同, 或者
- 无通讯

INT **整型:** 整型 16 位字。

IODDT **输入 / 输出导出的数据类型。**

Mapping (映射) 对特殊格式和不同格式的数据进行转换。

NIM **网络接口模块:** 设备和现场总线间的通讯。

NMT **网络管理:** 负责在 CAN 网络中管理执行、配置和错误。

PDO **过程数据对象:** 在 CANopen 中不同元素间进行数据交换的对象。

PROCESS
IMAGE
(过程映像) 系统存储器部分, 用于存储来自 CANopen 总线上的 PDO 交换的 E/S 值。此段由 CANopen 栈管理。
在每个任务循环开始时, 输入都被复制到用户应用程序存储器中, 在每个任务循环结束时, 都对输出进行复制。

REAL 实数。

RPDO 收到的 PDO 。

SDO	服务数据对象 ：对等通讯，可以访问 CANopen 总线元素的字典对象。
SSDO	SDO 服务器
STB	小型端子块。
SYNC	同步对象。

TPDO	PDO 传输。
-------------	---------

UDINT	无符号双精度整数 ：无符号双精度整数。
UINT	无符号整数 ：无符号整数。

施耐德电气公司
Schneider Electric China
www.schneider-electric.cn

北京市朝阳区将台路2号和
乔丽晶中心施耐德电气大厦
邮编: 100016
电话: (010) 8434 6699
传真: (010) 8450 1130

Schneider Electric Building, Chateau
Regency, No.2 Jiangtai Road, Chaoyang
District, Beijing 100016 China.
Tel: (010) 8434 6699
Fax: (010) 8450 1130

由于标准和材料的变更，文中所述特性和本资料中的图像
只有经过我们的业务部门确认以后，才对我们有约束。



本手册采用生态纸印刷