

SINAMICS S120

调试手册 · 01/2013
CANopen 接口

SINAMICS

SIEMENS

SIEMENS

SINAMICS

S120 CANopen 接口

开机调试手册

适用于：
固件版本 4.6

IH2, 01/2013
6SL3097-4AA00-0RP0

前言

引言

1

调试的前提条件

2

调试

3

计算/换算设定值/实际值

4

诊断

5

运行模式

6

通讯对象

7

每个驱动对象的 SDO Server

8




附录

A

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

 危险
表示如果不采取相应的小心措施， 将会 导致死亡或者严重的人身伤害。
 警告
表示如果不采取相应的小心措施， 可能 导致死亡或者严重的人身伤害。
 小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。
注意
表示如果不采取相应的小心措施，可能导致财产损失。


当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

 警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 ® 的都是西门子股份有限公司的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

前言

SINAMICS 文档

SINAMICS 文档分为以下几个类别：

- 通用文档/产品样本
- 用户文档
- 制造商/服务文档

其它信息

通过访问以下网址，您可以：

- 订购文档/查看印刷品一览
- 进入下载文档的链接
- 在线使用文档（搜索手册或信息）

<http://www.siemens.com/motioncontrol/docu>

如果您对技术文档有疑问（例如：建议或修改），请发送一份电子邮件到下列地址：
docu.motioncontrol@siemens.com

我的文档管理器

如何在西门子文档内容的基础上创建自定义文档，与自己的机床文档相匹配，请访问以下链接：

<http://www.siemens.com/mdm>

培训

通过以下链接可获取有关 SITRAIN 的信息 - 西门子为自动化产品、系统和解决方案制定的培训：

<http://www.siemens.com/sitrain>

常见问题

常见问题（FAQ）请点击**产品支持**，然后点击右侧的“支持”：

<http://support.automation.siemens.com>

SINAMICS

SINAMICS 的相关信息请参见以下网址：

<http://www.siemens.com/sinamics>

适用范围与其文档/工具（示例）

表格 1 适用范围和可供使用的文档/工具

适用范围	文档/工具
定位	SINAMICS S 销售文档
设计/配置	<ul style="list-style-type: none">选型工具 SIZER电机选型手册
决定/订购	<p>SINAMICS S120 产品样本</p> <ul style="list-style-type: none">SIMOTION、SINAMICS S120 及生产机械电机（产品样本 PM 21）SINAMICS 和用于单轴驱动的电机电机（产品样本 D 31）SINUMERIK & SINAMICS 机床设备（产品样本 NC 61）SINUMERIK 840D sl 1B 型机床设备（产品样本 NC 62）
安装/装配	<ul style="list-style-type: none">SINAMICS S120 控制单元和扩展系统组件手册SINAMICS S120 书本型功率单元手册SINAMICS S120 装机装柜型功率单元手册SINAMICS S120 AC 驱动手册SINAMICS S120M 分布式驱动技术手册
调试	<ul style="list-style-type: none">调试工具 STARTERSINAMICS S120 入门指南SINAMICS S120 调试手册SINAMICS S120 CANopen 调试手册SINAMICS S120 功能手册SINAMICS S120 Safety Integrated 功能手册SINAMICS S120/S150 参数手册

适用范围	文档/工具
使用/操作	<ul style="list-style-type: none">• SINAMICS S120 调试手册• SINAMICS S120/S150 参数手册
维护/维修	<ul style="list-style-type: none">• SINAMICS S120 调试手册• SINAMICS S120/S150 参数手册
文档目录	<ul style="list-style-type: none">• SINAMICS S120/S150 参数手册

目标使用人群

本文档供使用 SINAMICS 驱动系统的机器制造商、调试人员和维修人员使用。

优点

本文档介绍了 SINAMICS S120 调试和维修的必要信息、步骤和操作。

标准功能范畴

本文档描述的功能范畴可能和实际提供的驱动系统的功能范畴有偏差。

- 在驱动系统中可能会执行本文档中未提及的功能。但这并不表示在交付系统时必须提供这些功能以及相关的维修服务。
- 本文档中也可能会描述驱动系统上不存在的功能。提供的驱动系统的功能请参见订货资料。
- 机床制造商增添或者更改的功能，必须由机床制造商进行说明。

同样，为使文档简明清晰，本文档并不包含所有产品类型的所有信息，也不能考虑到订货、销售和维护的每种实际情况。

技术支持

各个国家技术咨询的电话号码请访问下列网址，点击其中的**联系方式**：

<http://www.siemens.com/automation/service&support>

欧盟符合性声明

欧盟 EMC 指令的符合性声明请访问网址：

<http://support.automation.siemens.com>

在网页中输入关键词 **15257461** 或联系您所在地区的西门子办事处。

优点

说明

本 SINAMICS S120 CANopen 调试手册对 SINAMICS 驱动组中 CANopen 接口的调试步骤进行说明。

本调试手册在“以书本型伺服控制为例的首次调试”基础上，补充说明了如何进行 CANopen 通讯接口的首次调试。

- 配备了通讯板 CBC10 的 SINAMICS S120
- “SINAMICS S120 调试手册”对整个 SINAMICS 驱动组的调试进行了详细描述。

ESD 提示

静电敏感元件（ESD）是可能会被静电场或静电放电损坏的单个元件、集成电路、模块或设备。



注意
<p>由电场或静电放电引起的损坏</p> <p>电场或静电放电可能会损坏单个元件、集成电路、模块或设备，从而导致功能故障。</p> <ul style="list-style-type: none">• 仅允许使用原始产品包装或其他合适的包装材料（例如：导电的泡沫橡胶或铝箔）包装、存储、运输和发运电子元件、模块和设备。• 只有采取了以下接地措施之一，才允许接触元件、模块和设备：<ul style="list-style-type: none">– 佩戴防静电腕带– 在带有导电地板的防静电区域中穿着防静电鞋或配带防静电接地带• 电子元件、模块或设备只能放置在导电性的垫板上（带防静电垫板的工作台、导电的防静电泡沫材料、防静电包装袋、防静电运输容器）。

安全提示

**接触带电压部件会引发生命危险**

接触带电压部件会导致死亡或重伤。

- 只有专业人员才允许在电气设备上作业。
- 在所有作业中必须遵守本国的安全规定。

通常有六项安全步骤：

1. 做好断电的准备工作，并通知会受断电影响的组员。
2. 断开设备电源。
 - 关闭设备。
 - 请等待至警告牌上说明的放电时间届满。
 - 确认导线与导线之间和导线与接地线之间无电压。
 - 确认辅助电压回路已断电。
 - 确认电机无法运动。
3. 检查其他所有危险的能源供给，例如：压缩空气、液压、水。
4. 断开所有危险的能源供给，措施比如有：闭合开关、接地或短接或闭合阀门。
5. 确定能源供给不会自动接通。
6. 确保设备已经完全闭锁并选择了正确的设备！

结束作业后以相反的顺序恢复设备的就绪状态。

**危险****一般安全提示**

- 只有确保此处说明的组件将装入的目标设备符合欧盟机械指令时，才允许进行调试。
- 只允许由专业人员对 **SINAMICS** 设备和三相交流电机进行调试。
- 该人员必须参照产品随附的用户技术资料，了解并遵循规定的危险和警告提示。
- 电气设备和电机运行时，电气回路必定具有危险电压。
- 设备运行时，轴运动也可能会带来危险。
- 因此必须在电气设备断电状态下开展所有工作。
- 只有根据 **IEC 61800-5-1** 的规定测试了 **SINAMICS** 设备和 **FI** 保护装置的兼容性后，才允许将带有三相交流电机的 **SINAMICS** 设备通过选择性通断、全电流敏感的故障电流保护装置连接到供电网络。
- 只有规范运输、专业存放、并规范地安装和装配，谨慎小心地操作和维护设备，才能确保设备和电机正常、可靠地运行。
- 对于一些特殊规格的型号，还应参考样本和供货说明。
- 此外，除了随附用户技术文档中的危险和警告提示，还必须遵守本国、本地区以及系统具体的要求和规定。
- 在 **0 V** 到 **48 V** 的所有接口和端子上，只允许连接符合 **EN 60204-1** 的安全低压（**PELV = Protective Extra Low Voltage**：保护特低电压）。

**警告****使用移动无线电装置或移动电话时设备的意外运动会引发生命危险**

在距离本组件大约 **2 m** 的范围内使用发射功率大于 **1 W** 的移动无线电设备或移动电话时，会导致设备功能故障，该故障会对设备功能安全产生影响并能导致人员伤亡或财产损失。

关闭设备附近的无线电设备或移动电话。

目录

前言	3
1 引言	13
1.1 基础知识	13
1.2 SINAMICS CAN 总线通讯结构	14
1.3 CANopen 对象字典	15
1.4 制造商专用对象	17
1.4.1 用于访问 SINAMICS 参数的对象	18
1.4.2 自由对象	21
1.5 可创建 PDO 的数量	22
1.6 过程数据的发送和接收报文	23
1.7 PDO 映射	25
1.8 COB-ID	26
1.9 SDO 服务	28
1.10 PDO 服务	33
1.11 与 CANopen 一同使用的 BICO 互联技术	37
1.12 Bootup 协议	38
1.13 网络管理 (NMT 服务)	39
1.14 CANopen 设备状态机	42
1.15 故障监控	45
1.16 保存参数, 恢复出厂设置	49
1.17 CAN 总线采样时间	51
1.18 可通过 CAN 总线控制的驱动的数量	52
2 调试的前提条件	53
2.1 调试的前提条件	54
2.2 CAN 总线的 CBC10	55
2.2.1 安装 CBC10	57
2.2.2 CANopen 功能	57
2.2.3 诊断 LED “OPT”	59
2.3 调试工具 STARTER	61
2.3.1 在 STARTER 中进行 BICO 互联的基本步骤	61

3	调试	71
3.1	首次调试的基本步骤	72
3.2	配置示例	73
3.3	CBC10 上的硬件设置	74
3.4	使用调试工具 STARTER 配置驱动设备	76
3.4.1	在线搜索驱动设备	76
3.4.2	自动采集驱动设备组件拓扑结构和配置	78
3.4.3	配置电机	80
3.4.4	配置控制单元 CU320-2 上 CBC10 的接口	84
3.4.4.1	CAN 总线接口	85
3.4.4.2	PDO 报文	86
3.4.4.3	监控	87
3.4.5	将项目加载到驱动设备	89
3.4.6	互联电源	92
3.5	配置接收 PDO 报文和发送 PDO 报文	93
3.5.1	驱动对象的预设 COB-ID 和过程数据对象	93
3.5.2	激活“自由 PDO 映射”	96
3.5.3	在自由 PDO 映射中设定 COB-ID 和映射参数	97
3.6	互联接收/发送缓冲器中的过程数据	104
3.6.1	读取接收/发送缓冲器中单个过程数据对象的映射	104
3.6.2	互联传输报文的过程数据	107
3.6.2.1	互联接收缓冲器	108
3.6.2.2	互联发送缓冲器	111
3.6.2.3	互联另一个驱动对象	115
3.7	在线模式下将项目从驱动设备加载到 PC/PG 并保存	116
3.8	SDO 访问 CANopen 过程数据对象	118
3.8.1	标准的 CANopen PZD 对象	118
3.8.2	自由的 CANopen PZD 对象	120
4	计算/换算设定值/实际值	123
4.1	转速设定值设定和转速实际值计算	123
4.2	转矩设定值设定和转矩实际值计算	126
5	诊断	129
5.1	报警对象（紧急对象）	130
5.2	设备内部故障列表（预定义故障区）。	131
5.3	故障寄存器（“Error Register”）	132
6	运行模式	133

7	通讯对象	137
7.1	与驱动无关的控制单元通讯对象	138
7.2	和驱动相关的通讯对象	141
7.2.1	接收 PDO 中的通讯对象	141
7.2.2	发送 PDO 中的通讯对象	148
7.2.3	其他通讯对象	155
7.3	自由对象	158
7.4	驱动子协议 DSP402 中的 CANopen 对象	159
8	每个驱动对象的 SDO Server	163
8.1	映射模型	163
8.2	EDS 文件	167
8.3	SDO Server	170
8.4	制造商专用 EDS 文件的对象	172
A	附录	175
A.1	词汇表	176
	索引	181

引言

1.1 基础知识

为了更好地理解本调试手册，您必须熟悉 **CANopen** 的相关专业术语。

您必须熟悉以下标准

说明

配备了 **CANopen** 的 **SINAMICS** 符合以下标准：

- CiA 301 (Application Layer and Communication Profile)
 - CiA 303-3 (Indicator Specification)
 - CiA 306 (Electronic data sheet specification for CANopen)
 - CiA 402 (Device Profile for Drives and Motion Control)
-

1.2 SINAMICS CAN 总线通讯结构

以 SINAMICS S120 为例，下图显示了调试 CANopen 接口时的硬件和软件配置。

图中显示了：

- CANopen 用户主站应用与 SINAMICS 驱动组之间的连接。
- 通讯板“CBC10”的 CAN 总线接口。
- 控制单元上的相关 CANopen 从站软件，以及调试时用于发送和接收报文的术语“发送”和“接收”的含义。
- 安装了调试工具 STARTER 的 PC，通过以太网接入。

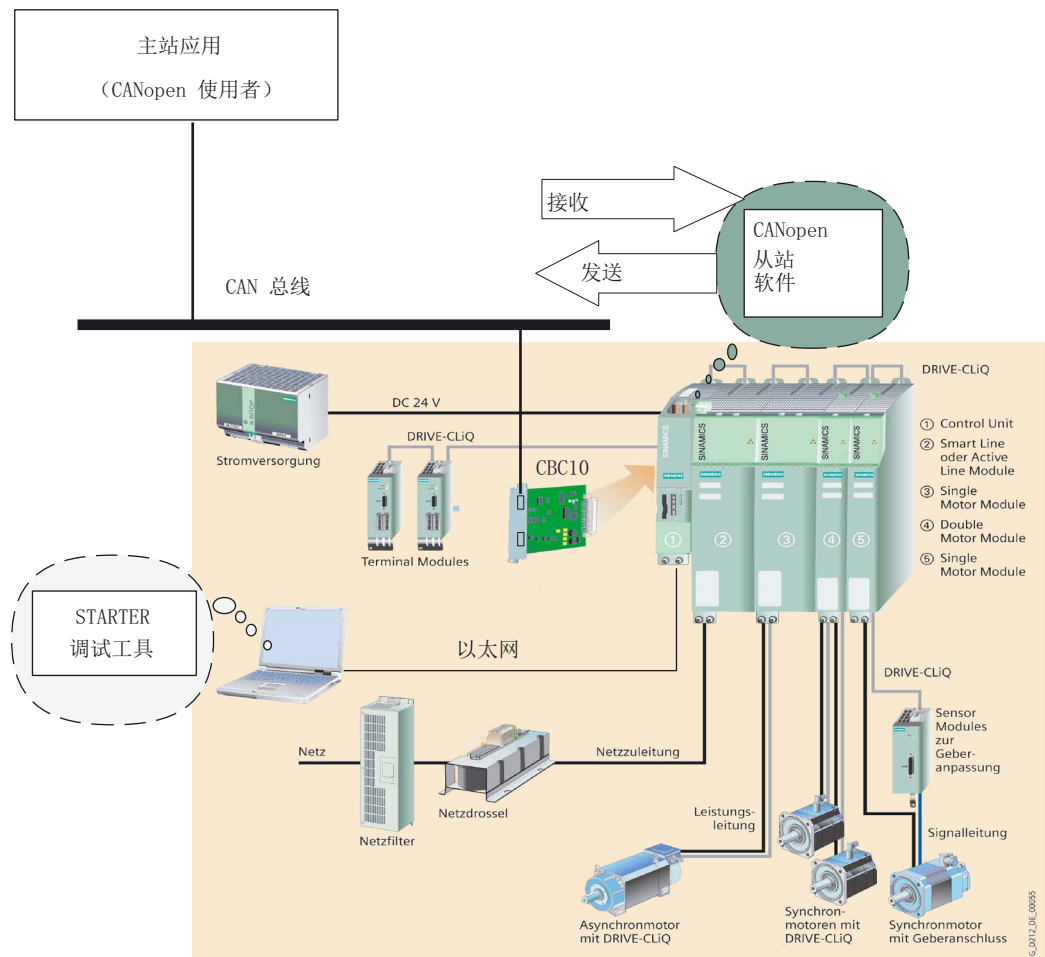


图 1-1 配置了 CAN 总线的 SINAMICS S120 驱动组

1.3 CANopen 对象字典

驱动对象初始化后，在 SINAMICS 驱动组的对象目录（CANopen 从站软件）中对 CANopen 对象进行初始化。

对象字典

下图显示了通讯中 CANopen 对象的分类（值为十六进制值）：

- 控制单元的通讯对象
- 和驱动相关的通讯对象
- 制造商专用对象
- 驱动子协议“DSP 402”的驱动对象

CANopen 最多支持八个驱动对象。

在“通讯对象 (页 137)”章节中以表格形式列出了 SINAMICS 中用于 CANopen 接口通讯的 CANopen 通讯对象和 SINAMICS 参数。

CANopen 对象字典

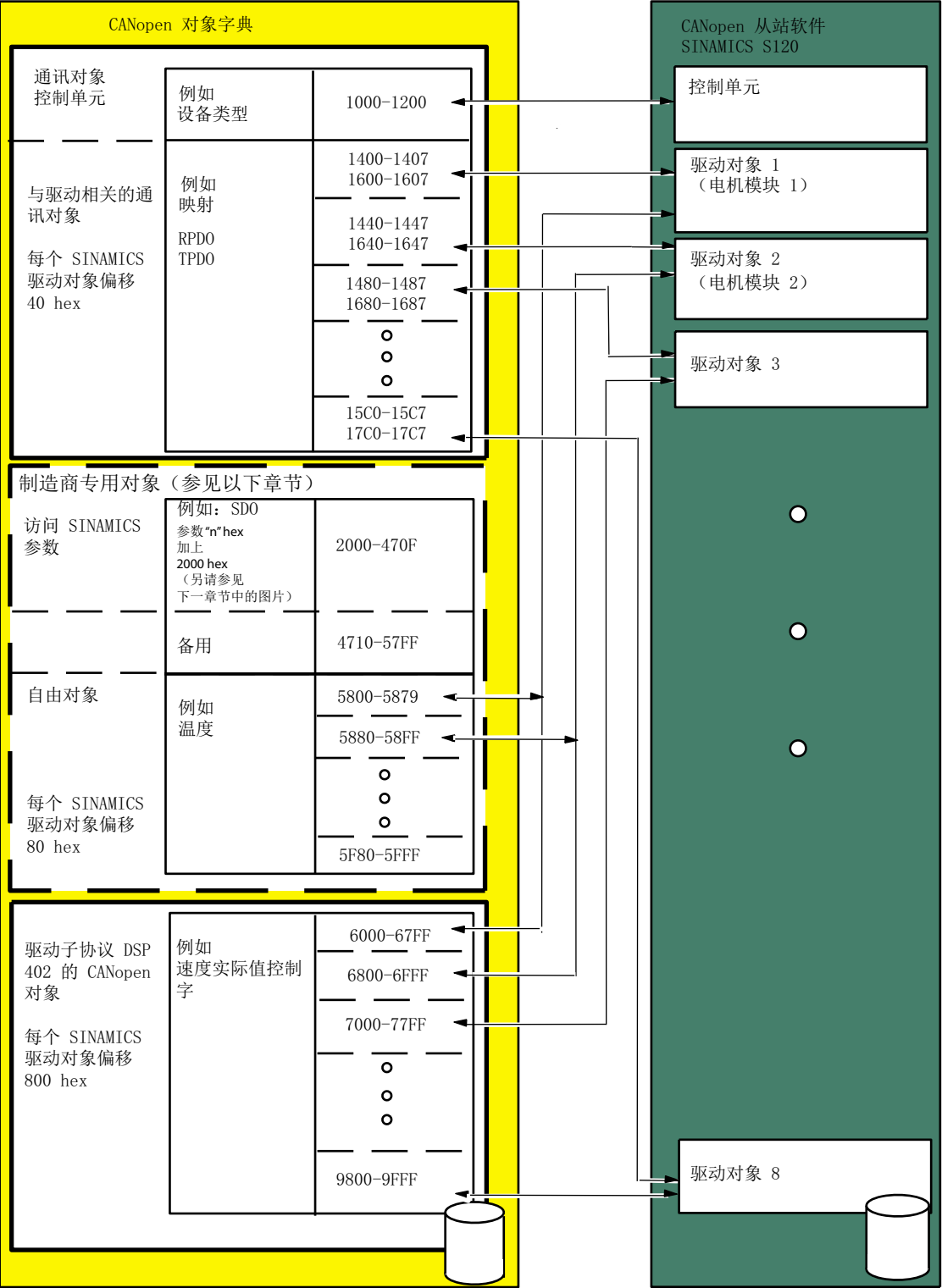


图 1-2 CANopen 对象字典

1.4 制造商专用对象

如上一章的对象字典图所示，CANopen 的对象字典中包含制造商专用对象区域。

制造商专用对象

制造商专用对象的定义为：

- 用于访问 SINAMICS 参数的对象
- 用于发送/接收过程数据的自由对象

对象字典中，制造商专用对象区域从地址“2000 hex”起，到“5FFF hex”结束。

1.4.1 用于访问 SINAMICS 参数的对象

通过对象字典中“2000 hex”至“470F hex”区域内的对象可访问 SINAMICS 参数的数据值。

示例

下图中显示了对对象字典中该对象的分配。

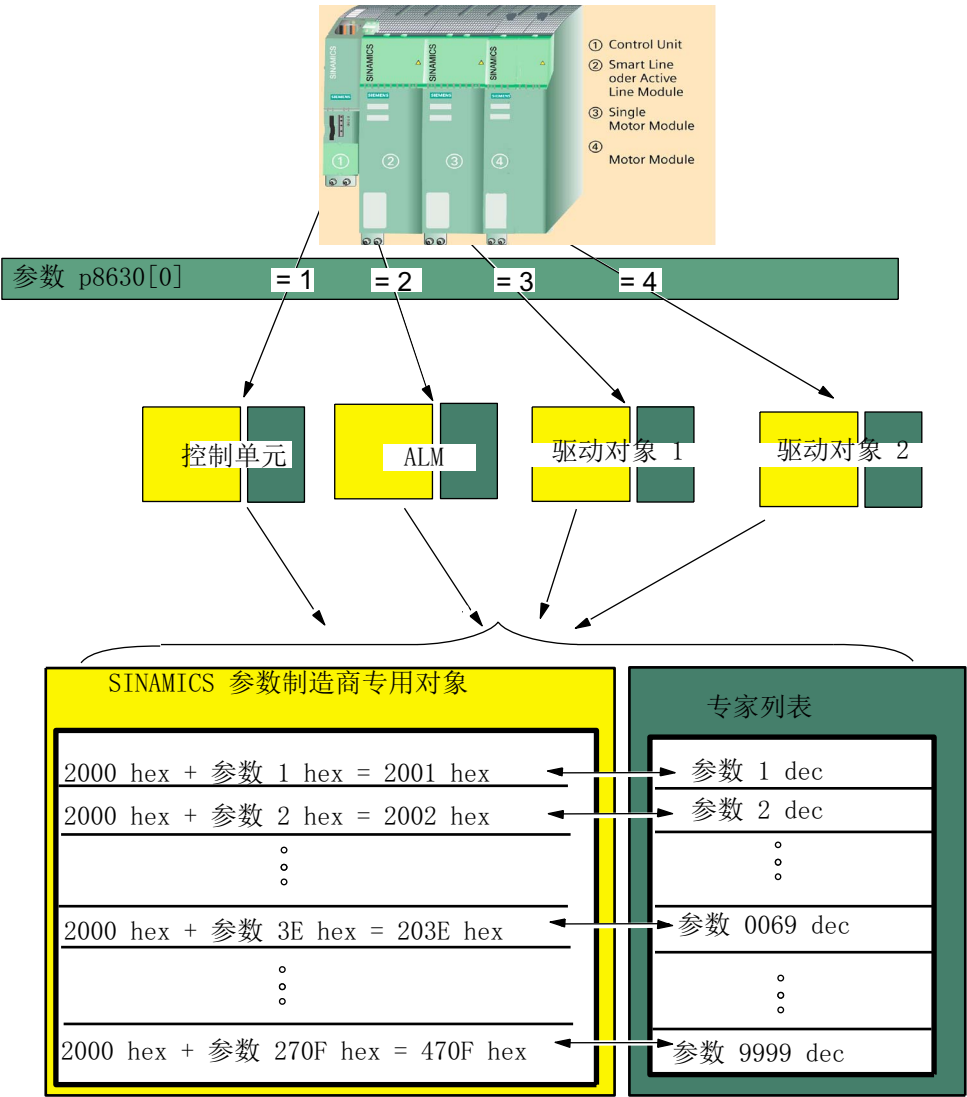


图 1-3 制造商专用对象

参数 p8630[0...2]

通过参数 p8630 索引 0 至 2 可确定 SINAMICS 中驱动对象的访问方式。有以下访问方式：

- 在 p8630[0] 中选择驱动对象
 - 0: 无法访问虚拟对象
 - 1: 控制单元
 - 2...65535: 驱动对象
- p8630[1] 中的子索引范围
 - 0: 0...255
 - 1: 256...511
 - 2: 512...767
 - 3: 768...1023
- p8630[2] 中的参数范围
 - 0: 0...9999
 - 1: 10000...19999
 - 2: 20000...29999
 - 3: 30000...39999
 - 4: 40000...49999
 - 5: 50000...59999

原理

所有 SINAMICS 参数都可通过 SDO 访问。

按以下步骤执行：

- 可通过“2000 hex”到“470F hex”的对象访问所有 SINAMICS 参数。
- SDO 访问将制造商专用对象内部转换为参数。将转换为十六进制值的参数编号加上“2000 hex”。

此数量为 SDO 请求中用于访问 SINAMICS 参数的对象编号。

- 因为 SINAMICS 驱动对象的参数区域已占据了制造商专用区域中参数访问的整个对象区域，因此应在 SINAMICS 参数 p8630[0] 中选择需要访问的驱动对象。
- SINAMICS 参数可为 r 参数或 p 参数。制造商专用数据中包含这些参数的数据值。
- 根据参数 p8630[0] 的开关位置，可读取或写入模块的数据值。

如果需要从驱动对象 1 中读取速度设定值参数 r0062，必须

- 在参数 p8630 [0] 中将开关设为“3”（见前图），
- 将换算为十六进制值的参数 r0062 加上“2000 hex”。十六进制数“203E hex”对应相应的对象编号，可通过 SDO 请求访问参数 r0062。

说明

参数 p8630[0] 中的开关决定驱动组的结构。如果未安装调节型电源模块 (ALM)，则从“2”开始第一个驱动对象的计数。该编号为驱动 ID。

通过参数 p8743[0...7] 可显示每个驱动 ID。在索引“0”中是第一个驱动 ID。

1.4.2 自由对象

可使用对象字典 (OV) 中 “5800 hex” 至 “5FFF hex” 区域的自由对象用于过程数据（参见章节 “CANopen 对象字典” (页 15)）。

自由对象

在对象字典中，每个驱动对象有以下自由连接对象可用：

表格 1- 1 驱动对象 1 的可自由连接对象

对象字典索引 (hex)	描述	PZD 的数据类型
5800 至 580F	16 个可自由连接的接收过程数据对象	Integer 16
5810 至 581F	16 个可自由连接的发送过程数据对象	Integer 16
5820 至 5827	8 个可自由连接的接收过程数据对象	Integer 32
5828 至 582F	备用	–
5830 至 5837	8 个可自由连接的发送过程数据对象	Integer 32
5838 至 5879	备用	–

说明

其它驱动对象的自由对象通过可自由连接对象的对象编号加上 “80 hex” 得出。
例如：驱动对象 2 从 “5880 hex” 开始。

可以通过接收/发送缓冲器的接收字/发送字/双字连接任意过程数据对象。

传输

可通过以下通讯服务访问自由对象：

- PDO
- SDO

1.5 可创建 PDO 的数量

PDO（Process Data Object，过程数据对象）用于传输对所选数据进行快速实时访问的过程数据。特定变量已经预先定义了和特定 PDO 的映射关系。

PDO 可为控制字，如“速度设定值”、“速度实际值”等。

说明

在一个 SINAMICS 驱动对象中最多可定义八个接收 PDO 和八个发送 PDO。SINAMICS 驱动组最多可管理 25 个接收 PDO。

由于每个 SINAMICS 驱动对象中最多可定义八个 PDO，在最大程度利用创建的 RPDO 时，最多可有三个 SINAMICS 驱动对象。还可再创建一个 RPDO。

1.6 过程数据的发送和接收报文

在 SINAMICS 驱动组中使用调试工具 STARTER 调试 CANopen 接口时，有两种调试方案：

1. 通过预定义报文（“Predefined Connection Set”）和 COB-ID。
2. 通过自由 PDO 映射，用户自定义报文。

建议在第一步中激活“预定义连接集”，接下来在自由 PDO 映射中根据应用更新预配置参数。自动调试时“预定义连接集”会自动设置。

SINAMICS 预定义报文

在相应驱动数据的接收和发送报文的“预定义连接集”中，以下过程数据对象已经经过预定义和映射。表格中包含了驱动对象 1 的对象的十六进制值。之后的驱动对象的值依次向后推移“800 hex”。

表格 1-2 预定义连接集中的过程数据对象

类型	过程数据	报文
接收报文	RPDO1	<div>16 位</div> <div>控制字</div> <div>6040 hex</div>
	RPDO2	<div>16 位 32 位</div> <div>控制字 速度设定值</div> <div>6040 hex + 60FF hex</div>
	RPDO3	<div>16 位 16 位</div> <div>控制字 转矩设定值</div> <div>6040 hex + 6071 hex</div>
	RPDO4	<div>16 位 32 位 16 位</div> <div>控制字 速度设定值 转矩设定值</div> <div>6040 hex + 60FF hex + 6071 hex</div>

1.6 过程数据的发送和接收报文

类型	过程数据	报文
发送报文	TPDO1	<div>16 位</div> <div>状态字</div> <div>6041 hex</div>
	TPDO2	<div>16 位32 位</div> <div>状态字速度实际值</div> <div>6041 hex + 606C hex</div>
	TPDO3	<div>16 Bit16 Bit</div> <div>状态字实际转矩</div> <div>6041 hex + 6077 hex</div>
	TPDO4	<div>16 位32 位</div> <div>状态字位置实际值</div> <div>6041 hex + 6063 hex</div>

说明

对于专家列表中的每个驱动对象，用于映射报文的过程数据对象开始于：

- 接收报文从参数 p8710 起
- 发送报文从参数 p8730 起

PZD 接口相应的 BICO 互联必须由用户自行创建。

1.7 PDO 映射

“PDO 映射”可以将驱动对象（过程数据，例如：设定值或实际值）和对象字典中用于 PDO 服务的“自由对象”映射为报文。

PDO 报文传输这些对象的数据值。

识别后每个驱动对象可分配到最多八个接收 PDO 和八个发送 PDO。

一条 CAN 报文可传输最多 8 字节有效载荷数据。 由用户决定在 PDO 中传输哪些有效载荷数据。

示例

下图通过一个示例来描述 PDO 映射（值为十六进制，例如对象大小 “20 hex” = 32 位）：

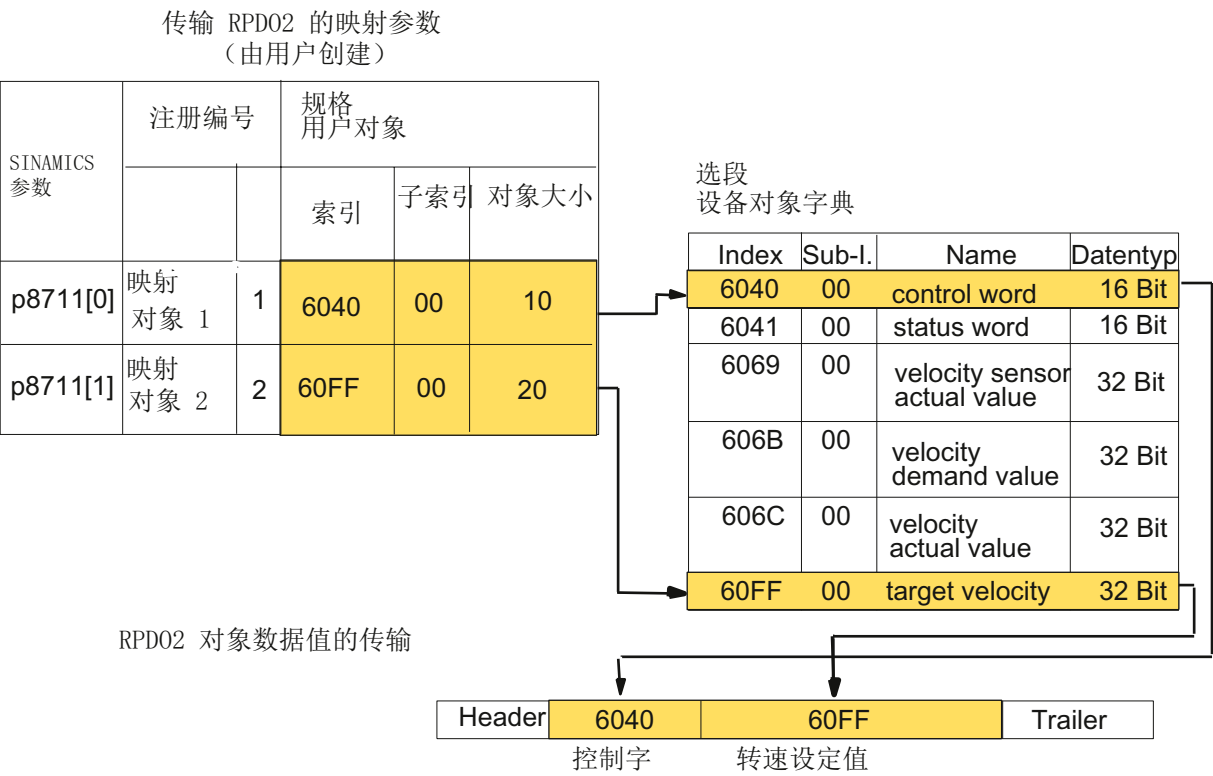


图 1-4 PDO 映射

1.8 COB-ID

每个 COB（通讯对象）都通过标识符确保其唯一性，标识符是 COB 的组成部分。CAN 标准 2.0A 最多支持 2048 个 COB，每个 COB 可通过 11 位的标识符识别。

在相关 SINAMICS 驱动设备的对象目录中可以获取 COB 标识符列表，该表中包含所有可通过 CAN 访问的 COB。

通过 COB-ID 的分配可确定通讯对象之间的优先级。

原理

原则上可自由选择 COB-ID。

CANopen 对通讯对象的标识符分配进行了预设（“Predefined Connection Set”）。下表为缺省标识符分配表。

表格 1- 3 标识符分配

通讯对象	功能码		对应的 COB-ID		对象字典索引(hex) ¹⁾
	dec	bin	hex	注释	
NMT 命令（NMT） ²⁾	0	0000	0	0 dec	–
Sync 消息（SYNC）	1	0001	80	128 dec	1005, 1006, 1007
紧急对象 （EMERGENCY）	1	0001	81–FF	129–255 dec	1014, 1015
Tx-PDO1	3	0011	181–1FF	180 hex + Node-ID	1800
Rx-PDO1	4	0100	201–27F	200 hex + Node-ID	1400
Tx-PDO2	5	0101	281–2FF	280 hex + Node-ID	1801
Rx-PDO2	6	0110	301–37F	300 hex + Node-ID	1401
Tx-PDO3	7	0111	381–3FF	380 hex + Node-ID	1802
Rx-PDO3	8	1000	401–47F	400 hex + Node-ID	1402

Tx-PDO4	9	1001	481-4FF	480 hex + Node-ID	1803
Rx-PDO4	10	1010	501-57F	500 hex + Node-ID	1403
Tx-SDO ²⁾	11	1011	581-5FF	580 hex + Node-ID	1200
Rx-SDO ²⁾	12	1100	601-67F	600 hex + Node-ID	1200
节点监控（NMT 故障控制） ²⁾	14	1110	701-77F	700 hex + Node-ID	100C, 100D

¹⁾ Tx- 和 Rx-PDO 的对象字典索引取决于驱动组中驱动对象的数量。之后的驱动对象的索引依次向后推移 40 hex。（Tx 或 Rx 中的 x 代表相应的 SINAMICS 驱动对象，T = Transmit（发送），R = Receive（接收））

例如：第一个驱动对象的 TPDO 的对象字典索引从 1800 hex 开始，RPDO 从 1400 hex 开始，那么之后的驱动对象的索引依次向后推移 40 hex → 例如第二个驱动对象为 1840 和 1440 hex。

在“预定义连接集”中，之后的 SINAMICS 驱动对象的 COB-ID 依次递增 1。

²⁾ 这些 COB-ID 为固定设置。

1.9 SDO 服务

SDO 服务用于访问连接的驱动设备的对象字典。SDO 连接是 SDO-Client（SDO 客户）与 Server（SDO 服务器）之间的点对点连接。

带对象字典的驱动设备即为 SDO Server。

根据 CANopen 定义驱动设备的第一个 SDO Server 的标识符。

Server 和 Client 之间的通讯为：

- Server 接收：COB-ID 600 hex + Node ID
- Server 发送：COB-ID 580 hex + Node ID

特性

SDO 有以下特性：

- 确定的对象传输
- 传输总是为异步
- 传输数据长度大于 4 字节（一般传输）
- 传输数据长度不大于 4 字节（加速传输）
- 相当于 PROFIBUS 的异步参数通道
- 通过 SDO 可访问驱动设备的所有变量

SDO 协议的结构

SDO 服务根据任务选择适用的协议。下面列出了 SDO 服务的几个最重要的协议。

SDO 写入协议

该协议用于将数据写入驱动设备。

通过信号“Write Response（写入应答）”确认。

表格 1-4 SDO 写入协议

Write Request（写入请求）			
CANopen 主站请求 ----> SINAMICS 信息			
字节 0	字节 1...2	字节 3	字节 4...7
cs = 2F hex	index	sub index	data byte 4
cs = 2B hex	index	sub index	data byte 4...5
cs = 27 hex	index	sub index	data byte 4...6
cs = 23 hex	index	sub index	data byte 4...7
Write Response（写入应答）			
SINAMICS 确认 ----> CANopen 主站应答			
字节 0	字节 1...2	字节 3	字节 4...7
cs = 60	index	sub index	reserved

上表的注释：

- cs: command specifier（特定命令字）
在 cs 中定义有多少发送字节相关数据。其它数据无意义。
- 索引，子索引：
应访问的对象索引和对象子索引。
- 保留：
保留用于以后的应用，总是为 0。

SDO 读取协议

该协议用于读取驱动设备数据。

通过信号“Read Request（读取请求）”请求读取。

表格 1- 5 SDO 读取协议

Read Request（读取请求）			
CANopen 主站请求 ----> SINAMICS 信息			
字节 0	字节 1...2	字节 3	字节 4...7
cs = 40	index	sub index	reserved
SINAMICS 确认 <---- CANopen 主站应答			
Read Response（读取应答）			
字节 0	字节 1...2	字节 3	字节 4...7
cs = 4F hex	index	sub index	data byte 4
cs = 4B hex	index	sub index	data byte 4...5
cs = 47 hex	index	sub index	data byte 4...6
cs = 43 hex	index	sub index	data byte 4...7

- 上表的注释：
- cs:command specifier（特定命令字）
在 cs 中定义有多少发送字节相关数据。 其它数据无意义。
 - 索引，子索引： SDO 索引和子索引。
 - 保留：
保留用于以后的应用，总是为 0。

SDO 传输中止记录协议

该协议用于执行 SDO 服务“传输中止记录”。

表格 1-6 SDO 传输中止记录协议

SINAMICS 请求 ----> CANopen 主站信息			
Error Response (错误应答)			
字节 0	字节 1..2	字节 3	字节 4..7
cs = 80	索引	子索引	中止代码

上表的注释:

- **cs: command specifier** (特定命令字)
该对象的大小总是为 4 字节。
- **索引, 子索引:** SDO 索引和子索引。
- **中止代码:**
包含 4 字节描述中止原因的中止代码。
中止代码是 UNSIGNED32 格式的值。

SDO 中止代码

表格 1-7 SDO 中止代码

中止代码	描述
0503 0000 hex	Toggle bit not alternated.
0504 0000 hex	SDO protocol timed out.
0504 0001 hex	Client/server command specifier not valid or unknown.
0504 0002 hex	Invalid block size (block mode only).
0504 0003 hex	Invalid sequence number (block mode only).
0504 0004 hex	CRC error (block mode only).
0504 0005 hex	Out of memory.
0601 0000 hex	Unsupported access to an object.
0601 0001 hex	Attempt to read a write only object.
0601 0002 hex	Attempt to write a read only object.
0602 0000 hex	Object does not exist in the object dictionary.
0604 0041 hex	Object cannot be mapped to the PDO.

0604 0042 hex	The number and length of the objects to be mapped would exceed PDO length.
0604 0043 hex	General parameter incompatibility reason.
0604 0047 hex	General internal incompatibility in the device.
0606 0000 hex	Access failed due to an hardware error.
0607 0010 hex	Data type does not match, length of service parameter does not match.
0607 0012 hex	Data type does not match, length of service parameter too high.
0607 0013 hex	Data type does not match, length of service parameter too low.
0609 0011 hex	Sub-index does not exist.
0609 0030 hex	Value range of parameter exceeded (only for write access).
0609 0031 hex	Value of parameter written too high.
0609 0032 hex	Value of parameter written too low.
0609 0036 hex	Maximum value is less than minimum value.
060A 0023 hex	Resource not available: SDO connection.
0800 0000 hex	General error.
0800 0020 hex	Data cannot be transferred or stored to the application.
0800 0021 hex	Data cannot be transferred or stored to the application because of local control.
0800 0022 hex	Data cannot be transferred or stored to the application because of the current device state.
0800 0023 hex	Object dictionary dynamic generation failed or no object dictionary is present (e. g. object dictionary is generated from file and generation fails because of a file error).

1.10

PDO 服务

“过程数据对象（PDO）”用于 CANopen 的实时数据传输。

PDO 与对象字典中的条目 (entry) 相关联，是驱动对象的接口。数据类型和驱动对象到 PDO 的映射通过对象字典中的 PDO 映射结构确定。在设备配置期间，PDO 数量和驱动对象到 PDO 的映射会传送到设备。

该传送由对象字典中相应条目上的 SDO 服务来执行。

PDO 分为两种类型。发送 PDO（TPDO）用于发送数据，接收 PDO（RPDO）用于接收数据。支持 TPDO 的 CANopen 设备被称为 PDO Producer（PDO 生产者），支持 PDO 的 CANopen 设备被称为 PDO Consumer（PDO 消费者）。PDO 通过 PDO 通讯参数和 PDO 映射参数标记。下面的两个表格列出了这些参数的结构。

参数

表格 1- 8 PDO 通讯参数 1400 hex ff, 1800 hex ff

子索引	名称	数据类型
00 hex	子索引数量	UNSIGNED8
01 hex	PDO 的 COB-ID	UNSIGNED32
02 hex	PDO 的传输类型	UNSIGNED8
03 hex*	禁止时间	UNSIGNED16
04 hex*	备用	UNSIGNED8
05 hex*	事件计时器	UNSIGNED16

* 只适用于 1800 hex ff

表格 1- 9 PDO 映射参数 1600h ff, 1A00h ff

子索引	名称	数据类型
00 hex	映射到 PDO 的对象数量	UNSIGNED8
01 hex	待映射的第一个对象	UNSIGNED32
02 hex	待映射的第二个对象	UNSIGNED32
03 hex	待映射的第三个对象	UNSIGNED32
04 hex	待映射的第四个对象	UNSIGNED32

PDO 通讯参数描述了 PDO 的通讯方式。PDO 映射参数中包含 PDO 内容的相关信息。

必须为每个 PDO 定义通讯参数和映射参数。

设备子协议中 PDO 的定义总是针对 CANopen 设备中的第一个逻辑设备。如果希望 PDO 定义针对第二个逻辑设备生效，必须按照设备子协议中的 PDO 编号将使用的 CANopen 设备的 PDO 编号增加 64 (40 hex) (见下表)。

表格 1- 10 示例：PDO 编号计算

CANopen 设备中的逻辑设备	CANopen 设备中的 PDO 编号	设备子协议中的 PDO 编号
1. 逻辑设备	PDO 编号 + 0 (PDO1 至 PDO64)	PDO 编号 (PDO1 至 PDO64)
2. 逻辑设备	PDO 编号 + 64 (PDO65 至 PDO128)	PDO 编号 (PDO1 至 PDO64)
第 3 个逻辑设备	PDO 编号 + 128 (PDO129 至 PDO192)	PDO 编号 (PDO1 至 PDO64)
第 4 个逻辑设备	PDO 编号 + 192 (PDO193 至 PDO256)	PDO 编号 (PDO1 至 PDO64)
第 5 个逻辑设备	PDO 编号 + 256 (PDO257 至 PDO320)	PDO 编号 (PDO1 至 PDO64)
第 6 个逻辑设备	PDO 编号 + 320 (PDO321 至 PDO384)	PDO 编号 (PDO1 至 PDO64)
第 7 个逻辑设备	PDO 编号 + 384 (PDO385 至 PDO448)	PDO 编号 (PDO1 至 PDO64)
第 8 个逻辑设备	PDO 编号 + 448 (PDO449 至 PDO512)	PDO 编号 (PDO1 至 PDO64)

对象字典中对应条目的索引通过以下公式计算：

- RPDO 通讯参数索引 = 1400 hex + RPDO 编号 -1
- TPDO 通讯参数索引 = 1800 hex + TPDO 编号 -1
- RPDO 映射参数索引 = 1600 hex + RPDO 编号 -1
- TPDO 通讯参数索引 = 1A00 hex + TPDO 编号 -1

传输方式

可使用以下 PDO 传输方式：

- 同步传输
- 异步传输

为了使通讯设备在传输期间保持同步，必须以特定时间间隔传输同步对象 (SYNC Object)。

下图显示了同步传输和异步传输的原理：

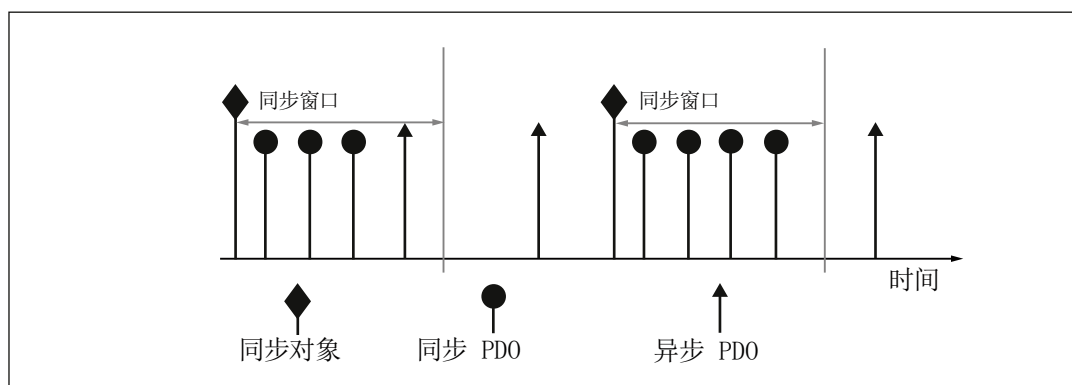


图 1-5 同步传输和异步传输的原理

PDO 传输方式的参数对该传输方式的特性进行描述。

对于同步 TPDO，传输方式也指出了传输速率，即：SYNC 对象“传输周期”的系数。传输方式“1”表示，在每个同步对象周期内传输消息。传输方式“n”表示，以 n 个同步对象周期为间隔传输消息。

传输异步 TPDO 时不考虑 SYNC 信号。

在 SYNC 信号后接收的同步 RPDO 在下一个 SYNC 信号后才传输到应用程序。

SYNC 生产者以特定时间间隔发送 SYNC 对象。SYNC 信号代表基本网络周期。两个 SYNC 信号之间的时间间隔通过缺省参数"通讯循环时间"定义。

SYNC 对象的标识符具有很高的优先级，用于访问 CAN 总线。出厂设置为 80h。该服务的运行不需经过确认。SYNC 对象的标识符可以设为别的值。此时需要对同一条总线上的所有 CANopen 从站进行相应的设置，以保持通讯的正常进行。

SYNC 对象标识符的对象索引为 1005 hex，两个 SYNC 信号之间的时间在对象索引 1006 hex 中定义。

说明

使用 SYNC 信号不能使 SINAMICS 的应用程序同步，只能使得 CANopen 总线上的通讯同步。

异步 RPDO 会直接传输至应用程序。

1.11 与 CANopen 一同使用的 BICO 互联技术

每个驱动设备中都包含大量可连接的输入/输出数据和内部控制数据。

利用 BICO 互联技术(Binector Connector Technology)，您可以对驱动设备功能进行调整，以满足各种应用的要求。

可通过 BICO 参数任意连接的数字和模拟信号，其参数名预设为 BI、BO、CI 或 CO。

这些参数在参数列表或功能图中也具有相应的标记。

包括：

- 二进制互联连接，其中
BI：二进制互联输入，BO：二进制互联输出
- 模拟量互联连接，其中
CI：模拟量互联输入，CO：模拟量互联输出

使用 BICO 技术互联信号

必须将 BICO 输入参数（信号汇点）分配给所需的 BICO 输出参数（信号源），以连接两个信号。

说明

使用 BICO 技术进行信号互联的详细信息请参见 SINAMICS S120 调试手册或 SINAMICS S120/S150 参数手册。

CANopen 参数的 BICO 互联

在 PDO 映射中创建的驱动对象在 CANopen 的接收和发送缓冲器中互联。每个过程数据对象在一个缓冲器中只出现一次。

接收/发送缓冲器过程数据对象的互联会在“互联接收/发送缓冲器中的过程数据 (页 104)”章节中进行描述。

1.12 Bootup 协议

NMT 从站启动后，Bootup 协议会发出信息：NMT 从站已经过初始化（Initialisation）状态，现处于预操作（Pre-Operational）状态。

Bootup 协议 COB-ID = 700h + Node-ID

只传输 1 个数据字节，值为 0。

1.13 网络管理 (NMT 服务)

网络管理的对象为节点，且采用主从结构。

使用 NMT 服务可初始化、启动、监控、复位和停止节点。所有 NMT 服务的 COB-ID = 0。

驱动设备为 NMT 从站。

概览

下图显示了 SINAMICS 连接中 CANopen 节点的状态图。图后的表格中列出了用于控制状态切换的 NMT 服务。

NMT 服务的详细信息请参见 CANopen 标准 “CiA 301 (Application Layer and Communication Profile)”。

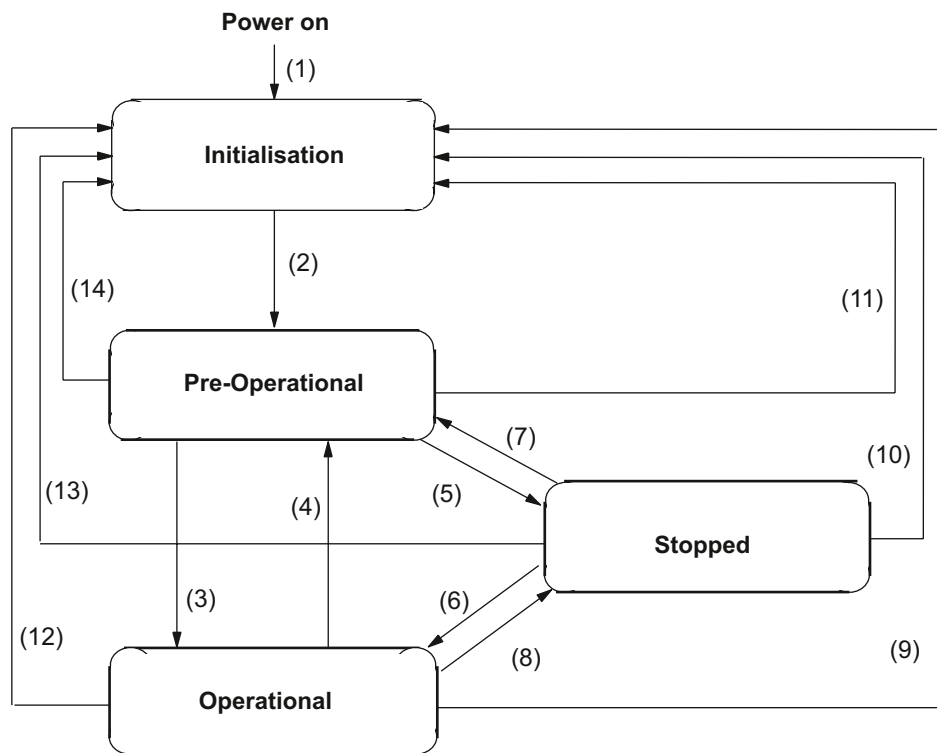


图 1-6 CANopen 节点状态图

说明

在“预操作”状态中只能进行 SDO 通讯，PDO 通讯被禁用。

在“操作”（Operational）状态中也可进行 PDO 通讯。

该状态通过诊断 LED -> 绿色 (CANopen RUN LED) 来显示。

NMT 服务

下表列出了用于控制状态切换的 NMT 服务。

表格 1- 11 状态切换图

切换	服务
(1)	POWER ON 后，控制单元自动进入初始化状态
(2)	在初始化后切换至预操作状态
(3), (6)	Start_Remote_Node 命令 (CS = 1)
(4), (7)	Enter_Pre-Operational_State 命令 (CS = 128)
(5), (8)	Stop_Remote_Node 命令 (CS = 2)
(9), (10), (11)	Reset_Node 命令 (CS = 129)
(12), (13), (14)	Reset_Communication 命令 (CS = 130)

各 NMT 服务的功能如下：

- **Start Remote Node**
从“预操作”状态切换至“操作”状态。在“操作”状态中驱动才能发送和接收过程数据。
- **Stop Remote Node**
从“预操作”或“操作”状态切换至“停止”（Stopped）状态。在“停止”状态下节点才能处理 NMT 命令。
- **Enter Pre-Operational**
从“操作”或“停止”状态切换至“预操作”状态。在“预操作”状态下节点不能处理 PDO。但是可以通过 SDO 进行参数设置和运行。也就是说可以设定设定值。
- **Reset Node**
从“操作”、“预操作”和“停止”状态切换至“初始化”状态。“Reset Node”命令后，所有对象 (1000 hex - 9FFF hex) 复位为 “POWER ON” 后的状态。
- **Reset Communication**
从“操作”、“预操作”和“停止”状态切换至“初始化”状态。“Reset Communication”命令后，所有通讯对象 (1000 hex - 1FFF hex) 复位为原始状态。

NMT 协议

对于下面显示的协议：

NMT 主站可同时向多个 NMT 从站发送请求。

协议显示针对 NMT 主站和 NMT 从站之间的数据交换。所有协议中的 COB-ID = 0。

Node-ID: 1-127, Node-ID = 0: 所有节点

表格 1- 12 COB-ID = 0

NMT 主站请求 ---> NMT 从站信息		
命令	字节 0	字节 1
Start	cs = 1 (01 hex)	Node-ID
Stop	cs = 2 (02 hex)	Node-ID
Enter Pre-Operational	cs = 128 (80 hex)	Node-ID
Reset Node	cs = 129 (81 hex)	Node-ID
Reset Communication	cs = 130 (82 hex)	Node-ID

cs:NMT command specifier

启动后的 NMT 状态

作为 NMT 服务的扩展，也可通过参数设置在 POWER ON 后自动切换至“操作”状态（参见“CANopen 节点状态图”）。

使用参数 p8684 可对 CANopen NMT 状态进行设置，使之在启动后或者通过 NMT 服务“Reset Node”或“Reset Communication”生效。

可使用的值：

- 4: 停止
- 5: 操作
- 127: 预操作（出厂设置）。

根据 CANopen 标准，出厂设置选择的 NMT 状态为“预操作”。

通过参数 p8685 可显示 CANopen NMT 状态或设置所需状态。

1.14 CANopen 设备状态机

CANopen 设备状态机（CANopen Device State Machine）对设备状态和可进行的驱动设备状态切换进行了描述。每种状态都描述了特殊的内部或外部特性。根据驱动设备的状态，只接受特定的切换命令。

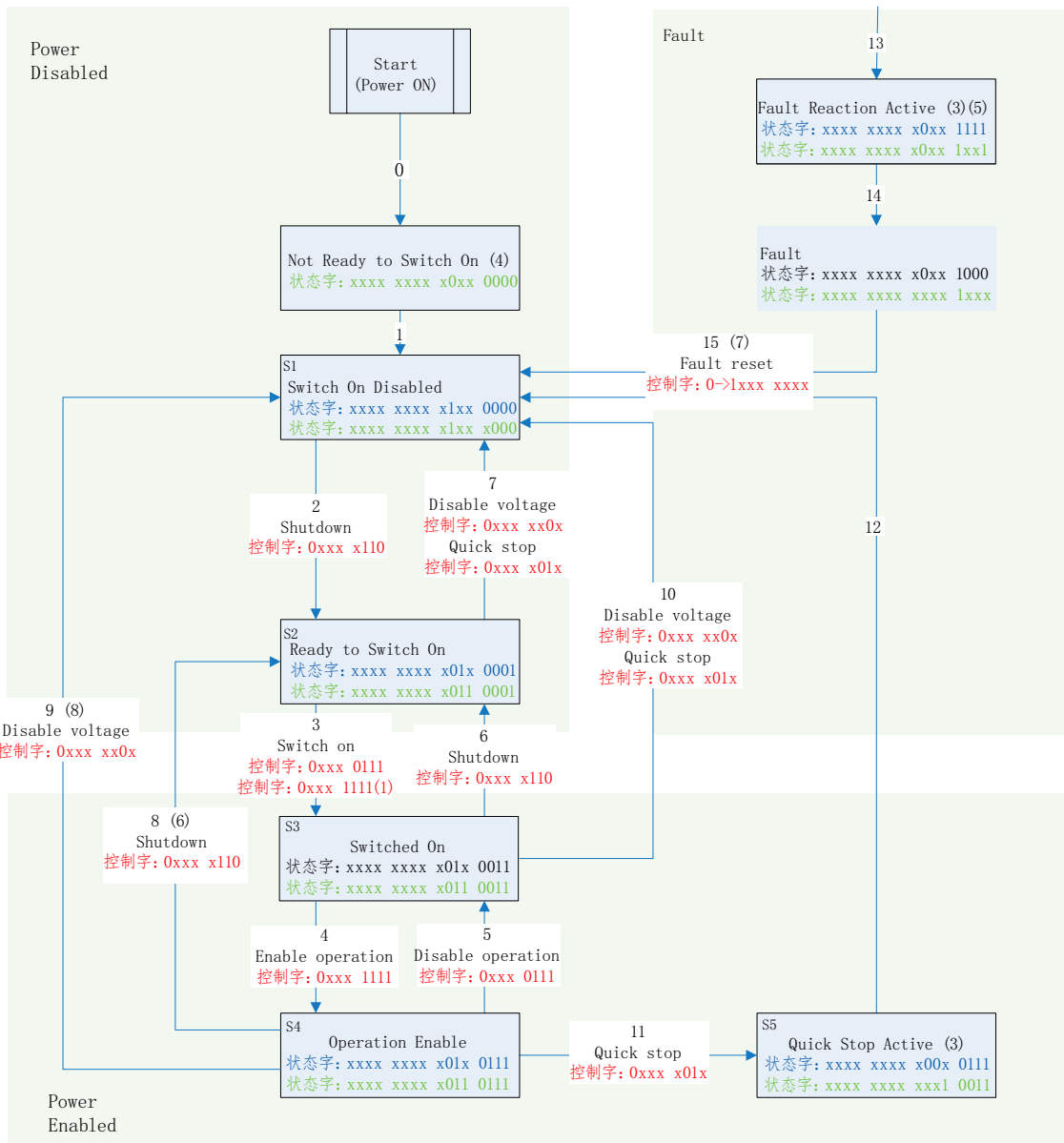
驱动设备的状态通过定义的控制字修改，且/或对应一个内部事件。当前状态可从 CANopen 状态字中读取。

表格 1- 13 CANopen 设备状态机

状态	条件
Not Ready to Switch On	驱动上存在较低功率。 驱动已初始化。 驱动功能已禁用。
Switch On Disabled	驱动初始化已结束。 驱动参数允许被修改。 驱动功能已禁用。 驱动上不允许接入高压。
Ready to Switch On	驱动参数允许被修改。 驱动功能已禁用。 驱动上允许接入高压。
Switched On	驱动参数允许被修改。 驱动功能已禁用。 驱动上已接入了高压。 电流放大器准备就绪。
Operation Enable	没有发现任何错误。 驱动参数允许被修改。 驱动功能激活，电机注入电源。 相当于驱动正常运行。
Quick Stop Active	正在执行急停功能。 驱动参数允许被修改。 驱动功能激活，电机注入电源。

状态	条件
Fault Reaction Active	驱动上出现不太严重的错误。 正在执行急停功能。 驱动参数允许被修改。 驱动功能激活，电机注入电源。
Fault	驱动参数允许被修改。 驱动上出现错误。

CANopen 设备状态机的状态和 SINAMICS 驱动上的映射



- (1) 经由“Switched On”过渡到“Operation Enable”
- (2) 只有当“Quick stop option code” 0x605A = 5 时才允许过渡。但不支持可选对象。
- (3) SINAMICS 顺序控制周期为 2 ms，因此只有在 PZD 采样时间 p8848 ≤ 2 ms 时才能检测到该状态。
- (4) 在该时间点上还没有建立 CANopen 通讯，因此无法在 CANopen 状态字中观察到内部状态。
- (5) 发生故障时，状态一直保持在“Switch On Disabled”。
- (6) 只有在 CBC10 采样时间 p8848 ≤ 2ms 时，才能观察到从“Switched On”到“Ready to Switch On”的自动过渡。
- (7) 只有在 CBC10 采样时间 p8848 ≤ 2 ms 时，才能观察到“Fault Reset”和“Shutdown”指令导致的“Switch On Disabled”到“Ready to Switch On”的自动过渡。
- (8) 功率部件显示实际状态，因此只有在 CBC10 采样时间 p8848 ≤ 2ms 时，才能观察到持续一个周期的状态显示 xx03hex (在控制字 = 0hex 时) 或 xx27hex (在控制字 = 27hex 时)

Sx = PROFIdrive 状态号
 ZSW = CANopen 状态字 (对象 6041)
 ZSW = SINAMICS 状态字
 STW = 控制字 (对象 6040)

图 1-7 CANopen 设备状态机

1.15 故障监控

故障监控服务

故障监控服务可以检测 CAN 网络内部的故障。

设备内部的故障，例如引起复位或状态切换的局部故障不在监控范围内。

故障监控服务以按特定时间间隔发送的 CANopen 设备消息为基础。有两种故障监控方式。

说明

必须激活“寿命保护”和“心跳”两种方式中的一种使故障监控生效。在初始设置中这两种方式均未激活。

寿命保护

NMT 主站通过节点保护协议发送监控请求。如果其中某个 NMT 从站在“节点寿命 (Node life time)”时间段内没有应答，或者 NMT 从站状态发生了改变，则 NMT 主站会向其主站应用程序发送消息。

如果 NMT 从站支持“寿命保护 (Life guarding)”（NMT 从站对 NMT 主站进行监控），它会通过对象字典中的“保护时间 (Guard time)”和“寿命系数 (Life time factor)”确定“节点寿命 (Node life time)”。如果 NMT 从站在“节点寿命 (Node life time)”内没有被 NMT 主站响应/监控，则会通过“寿命保护事件 (Life guarding event)”向其本地应用程序发送消息。如果从站对象字典中“保护时间 (Guard time)”和“寿命系数 (Life time factor)”的条目为“0”（缺省设置），则 NMT 从站不对 NMT 主站进行监控。

当从 NMT 主站通过其 COB-ID 接收到第一个“远程传输请求” (Remote transmit request:RTR) 时，NMT 从站监控启动。这在 Bootup 阶段时常发生。

心跳

“心跳生产者 (Heartbeat producer)”（CANopen 设备）以特定时间间隔发送心跳消息。网络中的一个或多个 CANopen 设备会识别此心跳消息。如果“心跳生产者”的心跳循环停止，心跳消费者会对其作出响应。

故障监控协议

节点保护协议

NMT 主站以特定时间间隔（节点保护时间：Node guard time）向每一个 NMT 从站发送请求。用于各 NMT 从站的时间间隔可不同。从站应答中包含其状态信息。“节点寿命”通过“节点保护时间”与“寿命系数”相乘得出，对于每个 NMT 从站可不同。如果 NMT 从站

在“节点寿命”内未收到来自 NMT 主站的请求，则在 NMT 从站中通过“寿命保护事件”显示“远程节点错误（remote node error）”

在以下情况下会生成“节点保护事件”：

- 在“节点寿命”内未确认 RTR（Remote transmission request，远程传输请求）
- NMT 从站反馈的状态与 NMT 所期望的状态不相符。

如果 CAN 通讯中发生故障，例如出现太多报文故障，则显示故障信息 F08700（详细信息请参见 SINAMICS S120/150 参数手册）。该故障在参数 r0949 中显示。使用 p8641 对驱动的故障响应进行设置。

“节点保护时间”和“寿命系数”的值保存在相应 NMT 从站的对象字典中。

表格 1- 14 节点保护协议 COB-ID = 700h + Node-ID

时间	NMT 主站	通讯		NMT 从站
节点寿命	请求 ---->	远程传输请求		----> 信息
	确认 <----	字节 0, 位 7	字节 0, 位 6..0	<---- 应答
	节点保护时间	t	s	
	请求 ---->	远程传输请求		----> 信息
	确认 <----	字节 0, 位 7	字节 0, 位 6..0	<---- 应答
		t	s	
	“节点寿命”届满后发送信息： “节点保护事件” **			“节点寿命”届满后发送信息： “寿命保护事件” **

缩写：

- ** 对于监控故障
- s NMT 从站状态
- t 触发位

该位的值只能在两个连续的 NMT 从站应答之间改变。“保护协议（Guarding protocol）”激活后，第一个应答的触发位的值必须置 0。当 NMT 状态为“Reset Communication”时，“保护协议”内的触发位应置 0（其它 NMT 状态不会复位触发位）。如果接收应答时触发位的值与上一次应答相同，则将此新应答作为未接收到处理。

心跳协议

心跳协议是不进行 RTR（RTR = Remote transmit request，远程传输请求）信号传输的故障监控服务。

“心跳生产者”会循环传输心跳信息。信息中包含一个或多个“心跳消费者（Heartbeat consumer）”。“生产者”和“消费者”之间的关系通过对象字典控制。

“心跳消费者”监控消费者能否在“心跳消费时间（Heartbeat consumer time）”内接收到“心跳”。如果在“心跳消费时间”内，“心跳消费者”中未接收到“心跳”，则触发“心跳事件（Heartbeat event）”。SINAMICS 驱动仅为“心跳生产者”。

表格 1- 15 心跳协议 COB-ID = 700h + Node-ID

心跳生产者	数据 字节 0		心跳消费者
请求 ---->	字节 0, 位 7	字节 0, 位 6..0	----> 信息
 心跳生产时间 	r	s	 心跳消费时间
请求 ---->	字节 0, 位 7	字节 0, 位 6..0	----> 信息
	r	s	 心跳消费时间
			心跳消费时间届满后：心跳事件

缩写：

r 保留（总为 0）
s “心跳生产者”状态

如果配置了“心跳生产时间”（对象编号 1017），则心跳协议立即启动。此时可从“初始化”状态切换到“预操作”状态。

在这种情况下会将 **Boot-up** 消息作为第一个心跳消息。

说明

不可同时使用节点保护协议和心跳协议。“心跳生产时间”不为零时，自动使用心跳协议。

1.16 保存参数，恢复出厂设置

通过以下控制单元通讯对象可以保存参数和恢复出厂设置。

- 保存参数 -> 通讯对象 1010 hex
- 恢复出厂设置 -> 通讯对象 1011 hex

保存参数（对象 1010 hex）

- 子索引 0: (1010.0):

此子索引显示了对象子索引的数量。

- 子索引 1: (1010.1):

将 ASCII 字符串“evas”（对应十六进制值“65 76 61 73”）写入此子索引后，驱动的所有参数会保存至非易失性存储器（存储卡）。

这相当于将驱动参数 p0977 写为“1”值。

- 子索引 2: (1010.2)**:

将 ASCII 字符串“evas”（对应十六进制值“65 76 61 73”）写入此子索引后，只有驱动的通讯对象（编号为 1000 hex - 1FFF hex 的对象）保存至非易失性存储器（存储卡）。

- 子索引 3: (1010.3)**:

将 ASCII 字符串“evas”（对应十六进制值“65 76 61 73”）写入此子索引后，只有驱动的通讯对象（编号为 6000 hex - 9FFF hex 的对象）保存至非易失性存储器（存储卡）。

通讯对象参数保存在存储卡上\USER\SINAMICS\DATA\目录下的 CCxxxxxn.ACX 文件中。

应用对象参数保存在存储卡上\USER\SINAMICS\DATA\目录下的 CAxxxxxn.ACX 文件中。

“n”指出了参数所属的驱动对象 ID。

只有事先至少一次进行了完整保存（子索引 1）后才能进行部分保存（子索引 2 或 3）。

读取子索引 1...3 时得到值 1，表示设备通过对象写访问保存参数。

恢复参数出厂设置（对象 1011 hex）

- 子索引 0: (1011.0):

此子索引显示了对象子索引的数量。

- 子索引 1: (1011.1):

将 ASCII 字符串“daol”（对应十六进制值“64 61 6F 6C”）写入此子索引后，驱动的所有参数恢复为出厂设置。

这相当于将驱动参数 p0976 写为“1”值。

- 子索引 2: (1011.2):

将 ASCII 字符串“daol”（对应十六进制值“64 61 6F 6C”）写入此子索引后，只有驱动的通讯对象（编号为 1000 hex - 1FFF 的对象）恢复为出厂设置。

- 子索引 3: (1011.3):

将 ASCII 字符串“daol”（对应十六进制值“64 61 6F 6C”）写入此子索引后，只有驱动的应用对象（编号为 6000 hex - 9FFF hex 的对象）恢复为出厂设置。

读取子索引 1...3 时得到值 1，表示设备通过对象写访问将参数恢复为出厂设置。

说明

出厂设置值在成功写入相应的参数后生效。根据 CANopen 协议，出厂设置值要在 NMT 命令“Reset Node”或“Reset Communication”或在开/关后才能生效。

1.17 CAN 总线采样时间

使用参数“p8848 IF2 PZD 采样时间”可设置 CBC10 的采样时间。

出厂设置中采样时间为 4 ms。在 4 ms 的时间内可接收和发送异步报文。

周期时间

- 对于循环接收报文，根据香农采样定理，周期时间必须大于至少两倍的采样时间。如果周期时间足够长，接收报文便不会丢失，报警 A8751 也不会出现。

示例：SYNC 周期时间应为 3 ms。p8848 中的设置 = 1 ms。这样周期时间便大于两倍的采样时间。

- 对于接收报文，根据香农采样定理，其数据变化率没有快于两倍的采样时间时，便可提高周期时间。通过参数 p2118、p2119 将报告类型转换为“不报告”便可不输出提高周期时间时出现的报警 A8751。

通过 p8848 设置 CAN 总线采样时间：

- 设置设备调试参数过滤器 p0009 = 3（驱动基本配置）。
- 修改和保存参数 p8848。
- 修改立即生效。

1.18 可通过 CAN 总线控制的驱动的数量

根据驱动的需求，通过 CAN 总线最多可同时控制四根轴，此时 CANopen 可使用最多四个发送报文和四个接收报文用于通讯。如果驱动使用了八个通讯报文，则只能对三个驱动进行控制 -> $3 * 8 = 24$ 。

可控驱动的最大数量受以下条件限制：

1. 对驱动的需求（例如从动轴需要的报文数量通常少于八个）
2. 通讯报文的最大数量，也就是 25。

伺服控制

- 最多四根轴可控。

矢量控制

- 最多两根轴可控。

调试的前提条件

本章节对调试的前提条件进行描述：

- CBC10
- 调试工具 STARTER

说明

全部的参数、故障、报警以及 SINAMICS 驱动组中 CANopen 的功能图都请参见 SINAMICS S120/S150 参数手册。

2.1 调试的前提条件

调试 SINAMICS 驱动组中的 CAN 总线时，需要以下硬件和软件组件：

- SINAMICS S120:
 - CBC10
 - 带固件的存储卡
- 以太网电缆，将控制单元的以太网接口和 PC/PG 连在一起
- PC/PG 上安装了调试工具 STARTER

说明

对 SINAMICS 驱动组组件的描述，例如布线、连接 PROFIBUS 接口和 PC/PG 以及安装调试工具 STARTER 的相关内容请参见 SINAMICS S120 设备手册（控制单元和附加系统组件）和 SINAMICS S120 设备手册（书本型功率单元）以及 SINAMICS S120 调试手册。

2.2 CAN 总线的 CBC10

CBC10 可将 SINAMICS S120 驱动系统通过 CAN 总线连接到上位自动化系统中。

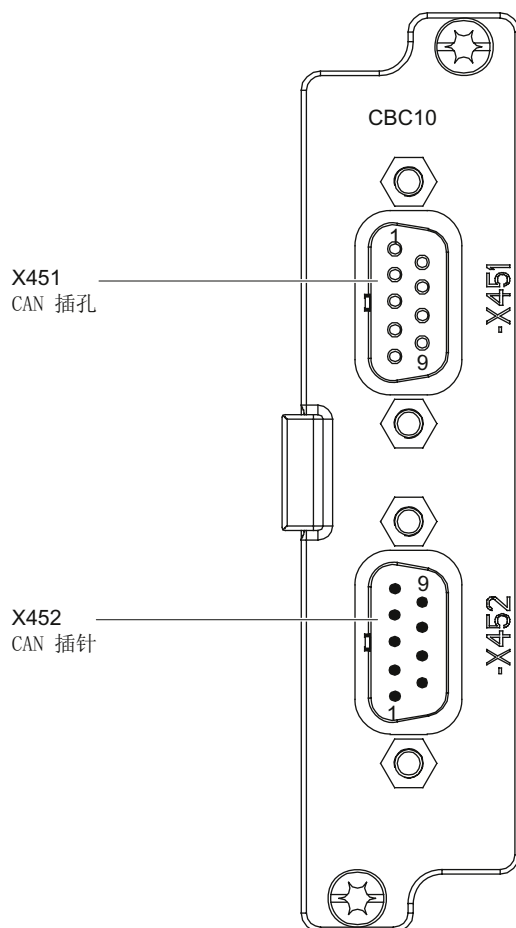


图 2-1 CBC10 视图

CBC10 上配备了两个 9 针 Sub-D 连接器，用于连接 CAN 总线系统。

这些连接器即可以用作输入端，也可以用作输出端。未使用的插口进行了通孔金属化处理。

此外，支持以下几种波特率：10、20、50、125、250、500、800 kBaud 和 1 MBaud。

2.2 CAN 总线的 CBC10

CAN 总线接口 X451 的插口布局如下。

表格 2- 1 CAN 总线接口 X451

	引脚	名称	技术数据
	1	备用	
	2	CAN_L	CAN 信号 (dominant low)
	3	CAN_GND	CAN 接地
	4	备用	
	5	CAN_SHLD	可选屏蔽
	6	GND	CAN 接地
	7	CAN_H	CAN 信号
	8	备用	
	9	备用	
类型： 9 针 SUB-D 母插			

CAN 总线接口 X452 的插口布局如下。

表格 2- 2 CAN 总线接口 X452

	引脚	名称	技术数据
	1	备用	
	2	CAN_L	CAN 信号 (dominant low)
	3	CAN_GND	CAN 接地
	4	备用	
	5	CAN_SHLD	可选屏蔽
	6	GND	CAN 接地
	7	CAN_H	CAN 信号
	8	备用	
	9	备用	
类型： 9 针 SUB-D 公插			

2.2.1 安装 CBC10

安装步骤

如下图所示，按照以下步骤将 CBC10 安装到控制单元 CU320-2 的选件槽。

1. 拧开螺钉并拆下防护盖板。
2. 插入 CBC10。
3. 固定 CBC10。

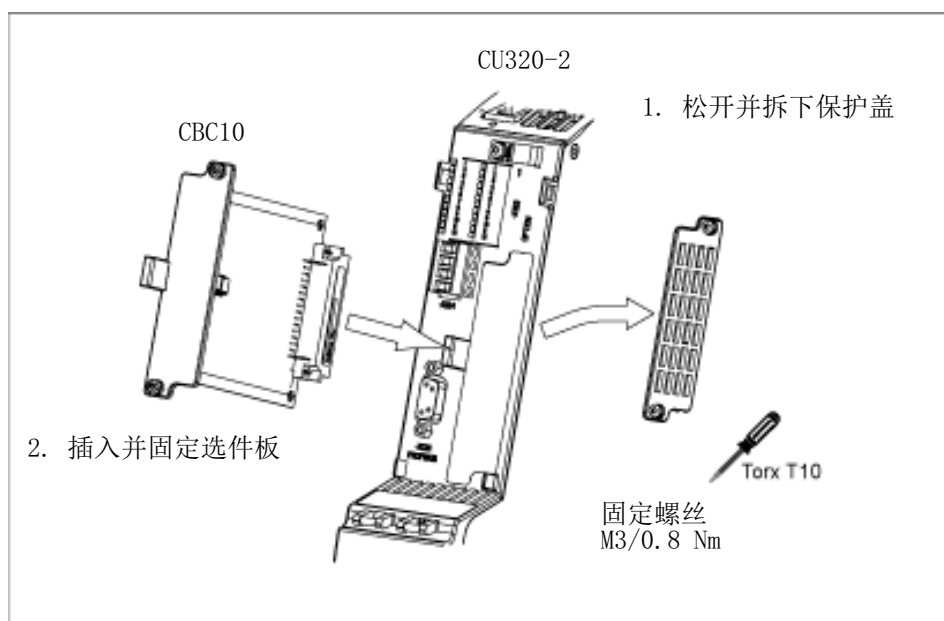


图 2-2 安装 CBC10

2.2.2 CANopen 功能

CBC10 支持通过 SDO（Service Data Objects，服务数据对象）和通过 PDO（Process Data Objects，过程数据对象）进行的 CANopen 传输。

此外 CBC10 支持自由 PDO 映射。

CBC10 支持 CANopen 通信子协议 DS 301 版本 4.0、设备子协议 DSP 402（驱动和运动控制）版本 2.0，以及指示器子协议 DR303-3 版本 1.0。

CBC10 支持用于监控通讯的节点保护和心跳协议（心跳生产者）。

CBC10 提供一个 SDO Server，可通过此 SDO Server 读取和写入所有 SINAMICS 参数。

2.2 CAN 总线的 CBC10

CBC10 的固件支持以下模式：

- 协议速度模式
- 协议转矩模式
- 速度模式

节点保护/寿命保护

SINAMICS 在特定的时间（节点寿命）内等待主站应用程序的报文，并允许特定时间（节点保护时间）内特定数量（寿命系数）的错误。

节点寿命通过节点保护时间和寿命系数相乘得出。

心跳协议

SINAMICS（生产者）将其通讯状态（生命符号）循环（心跳时间）发送至 CAN 总线上的主站应用程序。

协议速度模式

此运行方式允许根据可选的速度协议设定速度设定值。

协议转矩模式

该运行方式支持转矩控制以及为其指定的对象。

速度模式

该运行方式支持采用斜坡的简单速度控制以及相关对象。

参见

运行模式 (页 133)

2.2.3 诊断 LED “OPT”

下图中控制单元 CU320-2 上的诊断 LED “OPT” 显示了设备上 CANopen 节点的状态。

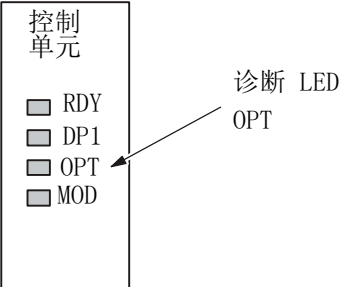


图 2-3 控制单元上的 LED 一览

控制单元上的诊断 LED “OPT” 可显示模块状态和通讯状态，从而向用户提供 CBC10 当前状态的所有必要信息。

原理

通过不同的闪烁频率：

- 诊断 LED OPT -> 红色
存在故障吗？
- 诊断 LED OPT -> 绿色
通讯状态机中的节点处于哪种状态？

诊断 LED OPT -> 红色

表格 2-3 诊断 LED OPT -> 红色（CANopen Error LED）

ERROR LED 闪烁频率	状态	含义
Off	No error	运行就绪
Single flash	Warning limit reached	至少一个 CAN 控制器故障计数器达到了警告阈值“消极故障（Error Passive）”（错误报文过多）。
Double flash	Error Control Event	发生了保护事件。
On	Bus off	CAN 控制器从总线断开。


诊断 LED OPT -> 绿色

表格 2- 4 诊断 LED -> 绿色（CANopen RUN LED）

ERROR LED 闪烁频率	状态	含义
Single flash	Stopped	节点处于“停止”状态。
Blinking	PRE- OPERATIONAL	节点处于“预操作”状态。
On	OPERATIONAL	节点处于“操作”状态。

2.3 调试工具 STARTER

调用 STARTER

1. 点击用户界面上的 STARTER 图标 。
- 或者
2. 在 Windows 开始菜单中选择 “Start > SIMATIC > STEP 7 > STARTER” 来启动 STARTER。

2.3.1 在 STARTER 中进行 BICO 互联的基本步骤

可以在 STARTER 离线模式下、通过 BICO 互联来设置连接的驱动。参数设置可通过以下方式进行：

- 专家列表
- 图形屏幕界面

以下是在调试工具 STARTER 中进行 BICO 互联的基本步骤。

在专家列表中进行 BICO 互联

通过专家列表进行 BICO 互联时，执行以下步骤：

例如需要将控制字参数 p0840 与参数 r8890[0] 互联。

1. 在项目导航器中右击“Drive_1”并通过右键菜单“Expert list”来调用专家列表。
2. 搜索参数 p0840。

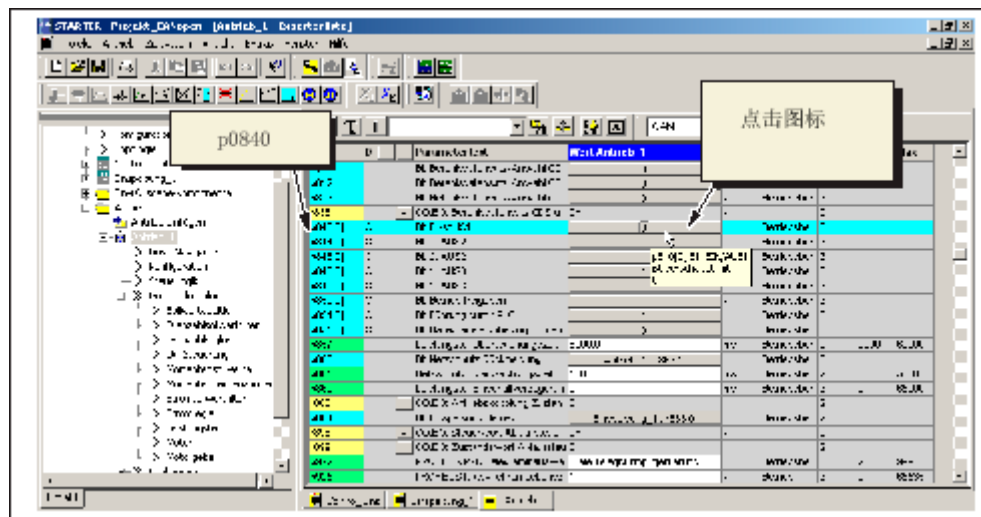


图 2-4 互联 1

3. 点击按钮与参数互联。

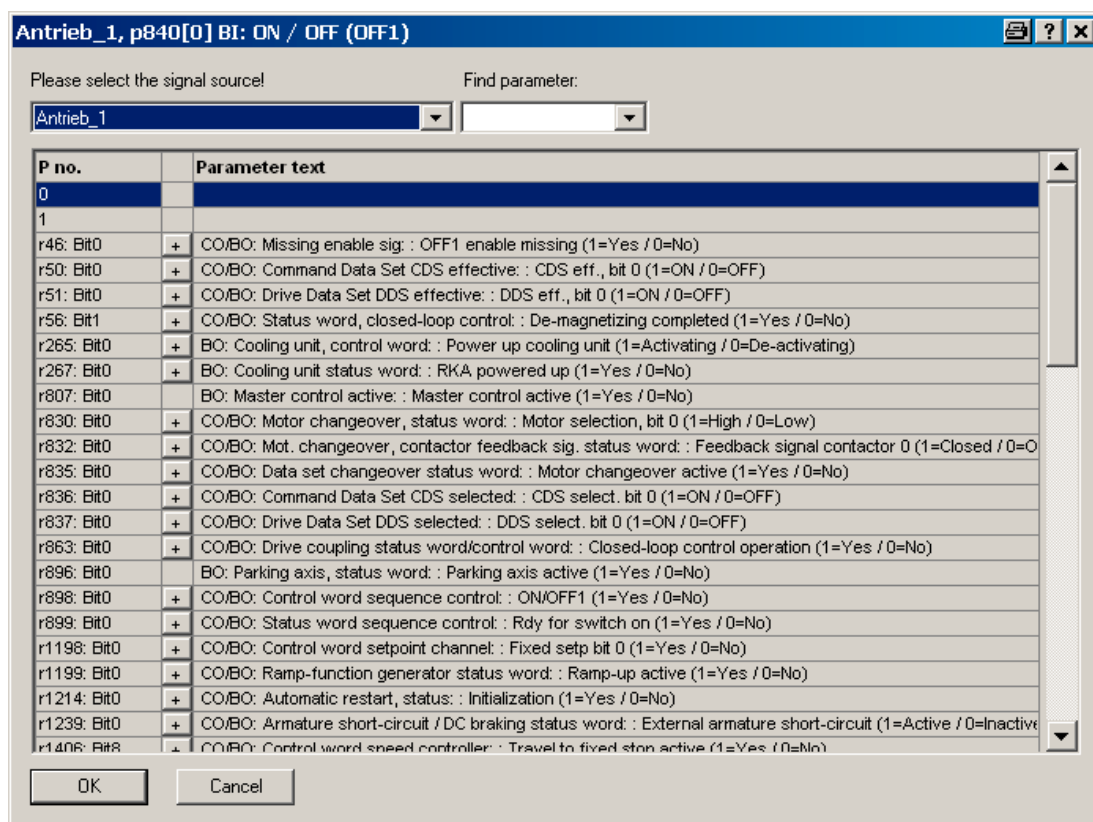


图 2-5 互联 2

此时会显示可选择的 r 参数的列表。

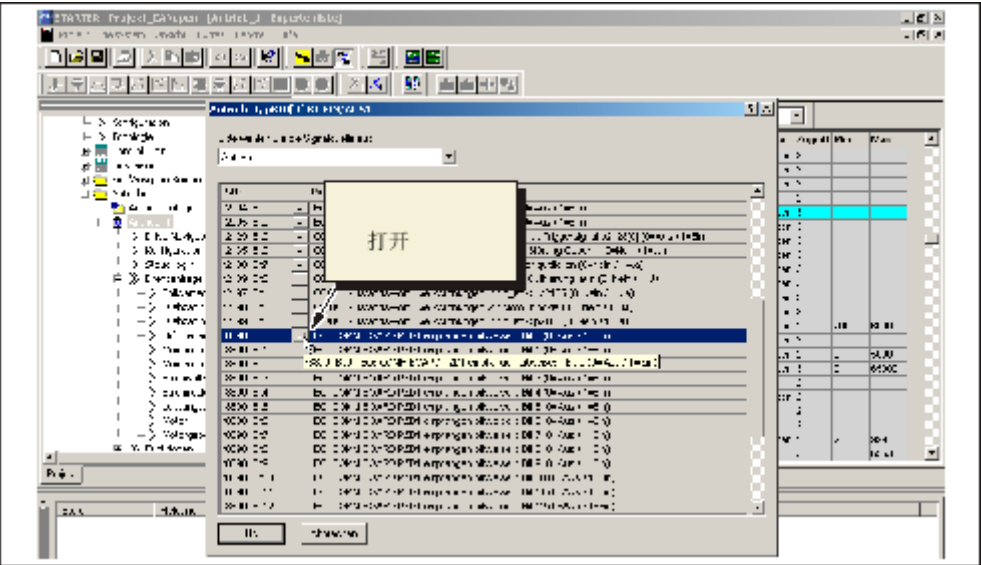


图 2-6 互联 3

- 4. 点击参数 r8890:Bit0 的加号图标。

5. 双击 r8890: Bit0。

现在在专家列表中可以看到，p0840 已经与参数 r8890[0] 互联。

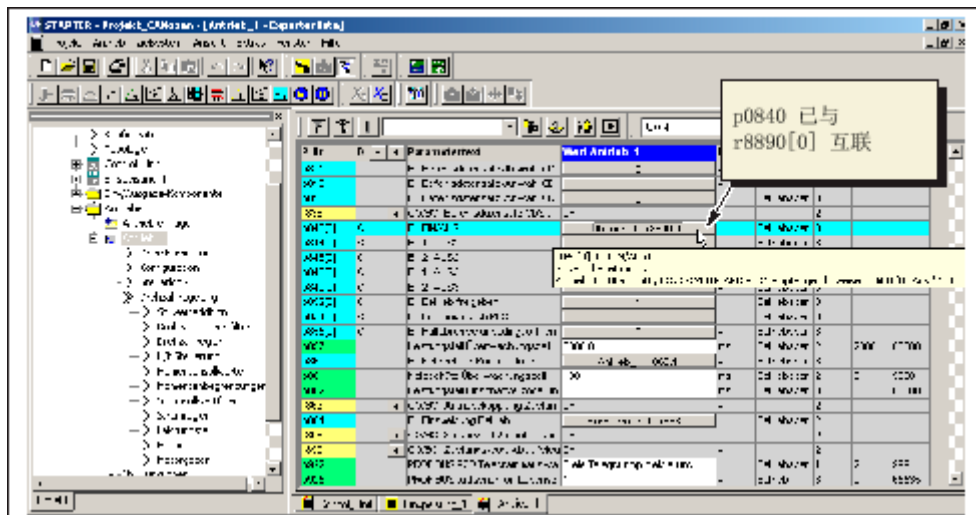


图 2-7 互联 5

2.3 调试工具 STARTER

图形屏幕界面

通过图形屏幕界面进行 BICO 互联时，执行以下步骤：

例如对于速度设定值（数据类型 32 位），需要将“转速设定值 1”的参数 p1155[0] 与参数 r8860[1] 互联。

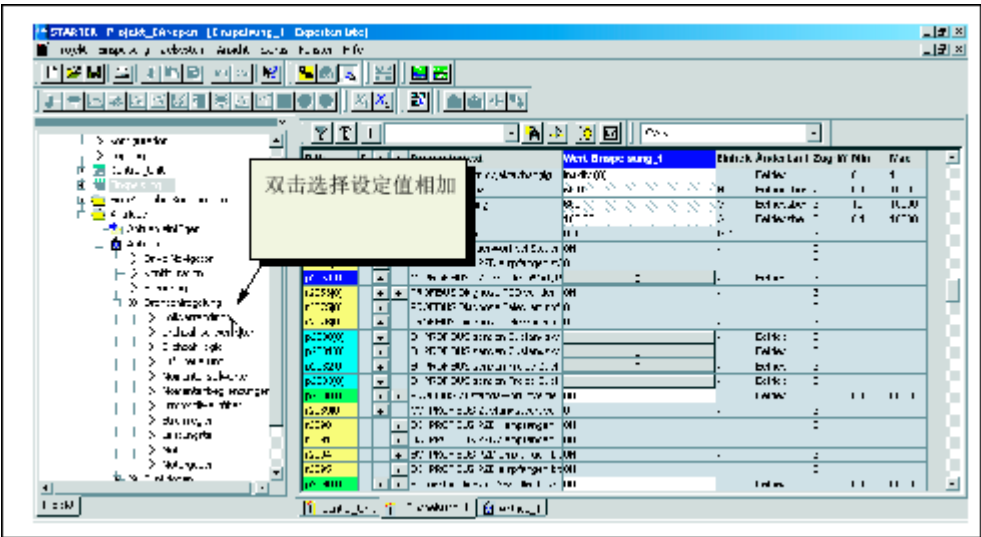


图 2-8 互联界面 1

1. 在项目导航器中，在“Drive_1 > Open-loop/closed-loop control”下双击“Setpoint addition”。

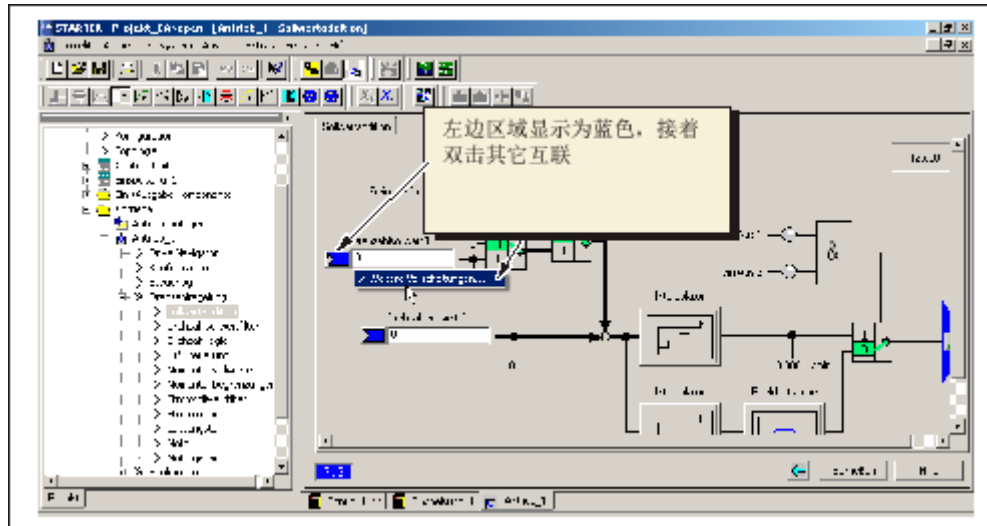


图 2-9 互联界面 2

- 2. 点击“Speed setpoint 1”左侧的蓝色区域，然后点击显示的选项“Further interconnections”。

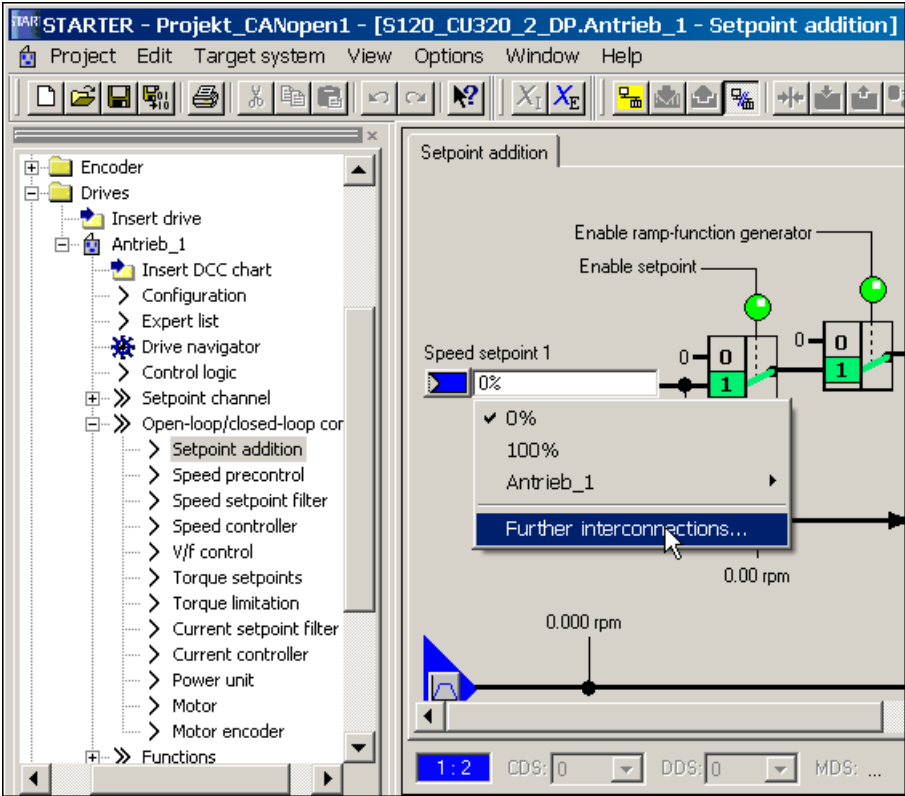


图 2-10 互联界面 3

此时会显示可选择的 r 参数的列表。

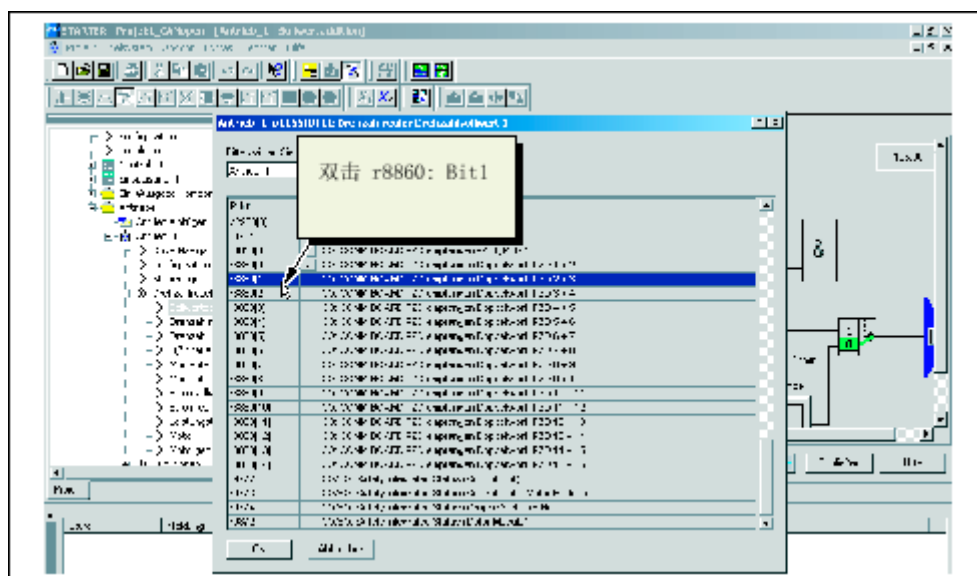


图 2-11 互联界面 4

3. 双击 r8860: Bit1。

现在在图形界面中可以看到，p1155 已经与参数 r8860[1] 互联。

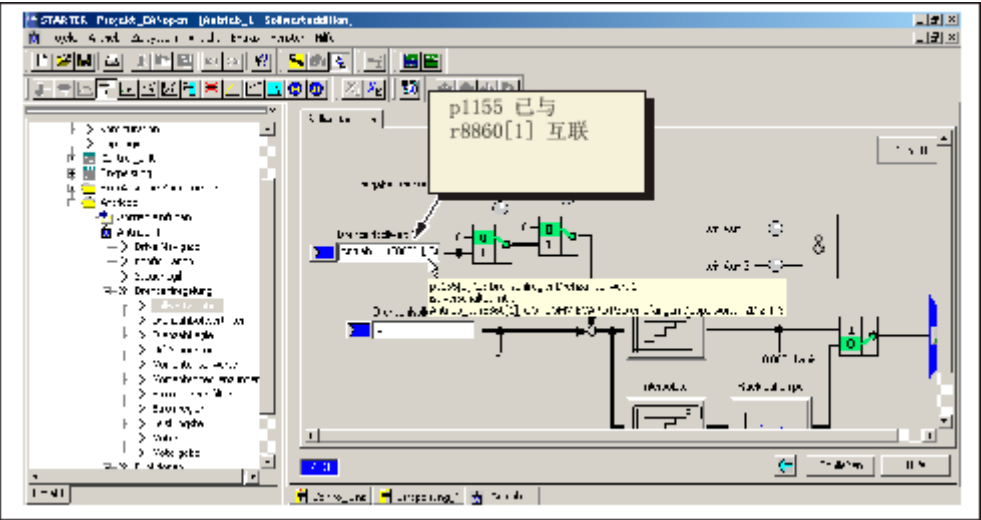


图 2-12 互联界面 5

参见

与 CANopen 一同使用的 BICO 互联技术 (页 37)

互联接收/发送缓冲器中的过程数据 (页 104)

调试

章节内容

本章节将描述如何使用调试工具 **STARTER** 对 **SINAMICS** 驱动组中的 **CANopen** 接口进行首次调试。

首先会说明首次调试时的基本步骤，并给出了一个配置示例。

首次调试的说明针对的是调试工具 **STARTER** 的在线模式。在每一步的末尾，必要时会提示离线调试与在线调试的区别。

前提条件

在按照本章节中描述的步骤进行调试以前，请先阅读“调试的前提条件 (页 53)”章节中的内容。

3.1 首次调试的基本步骤

首次调试

在首次调试 SINAMICS 驱动组中的 CANopen 接口时，需要执行以下基本操作：

1. 进行 CBC10 上的硬件设置。
2. 使用调试工具 STARTER 在线配置驱动设备。
3. 配置接收/发送报文的 COB-ID 和过程数据对象。
4. 互联接收缓冲器和发送缓冲器。
5. 在线模式中，将驱动设备中的项目加载到 PC/PG 并保存。

3.2 配置示例

说明

下面描述的调试步骤以下图中显示的 **SINAMICS S120** 驱动组中的配置示例为基础：

- 一个电源（调节型电源模块）
- 两个驱动（单轴电机模块）

有关该配置的详细信息参见 **SINAMICS S120** 中的章节“书本型伺服控制的首次调试”。

本调试手册在“以书本型伺服控制为例的首次调试”基础上，补充说明了如何进行 **CANopen** 通讯接口（带“CBC10”）的首次调试。

概述

下图显示了驱动对象概览。

CANopen 配置示例：为 SINAMICS 驱动对象分配模块

- 单轴电机模块 1 为第一个 **SINAMICS** 驱动对象
- 单轴电机模块 2 为第二个 **SINAMICS** 驱动对象

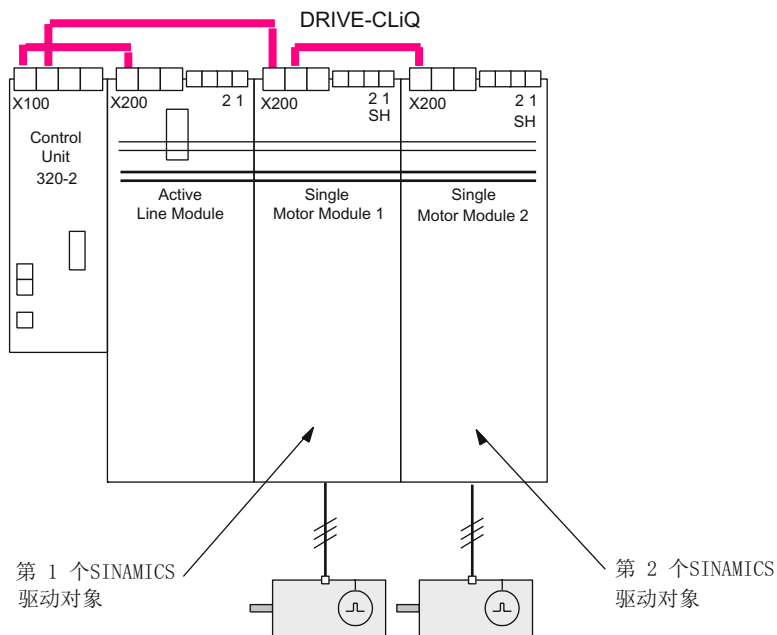


图 3-1 SINAMICS S120 组件（示例）

3.3 CBC10 上的硬件设置

必须正确设置 CBC10 上的开关 S1/S2，以确保数据可通过 CAN 总线安全传输（见下表）。需要设置：

- 总线终端电阻
- 不接地/接地运行

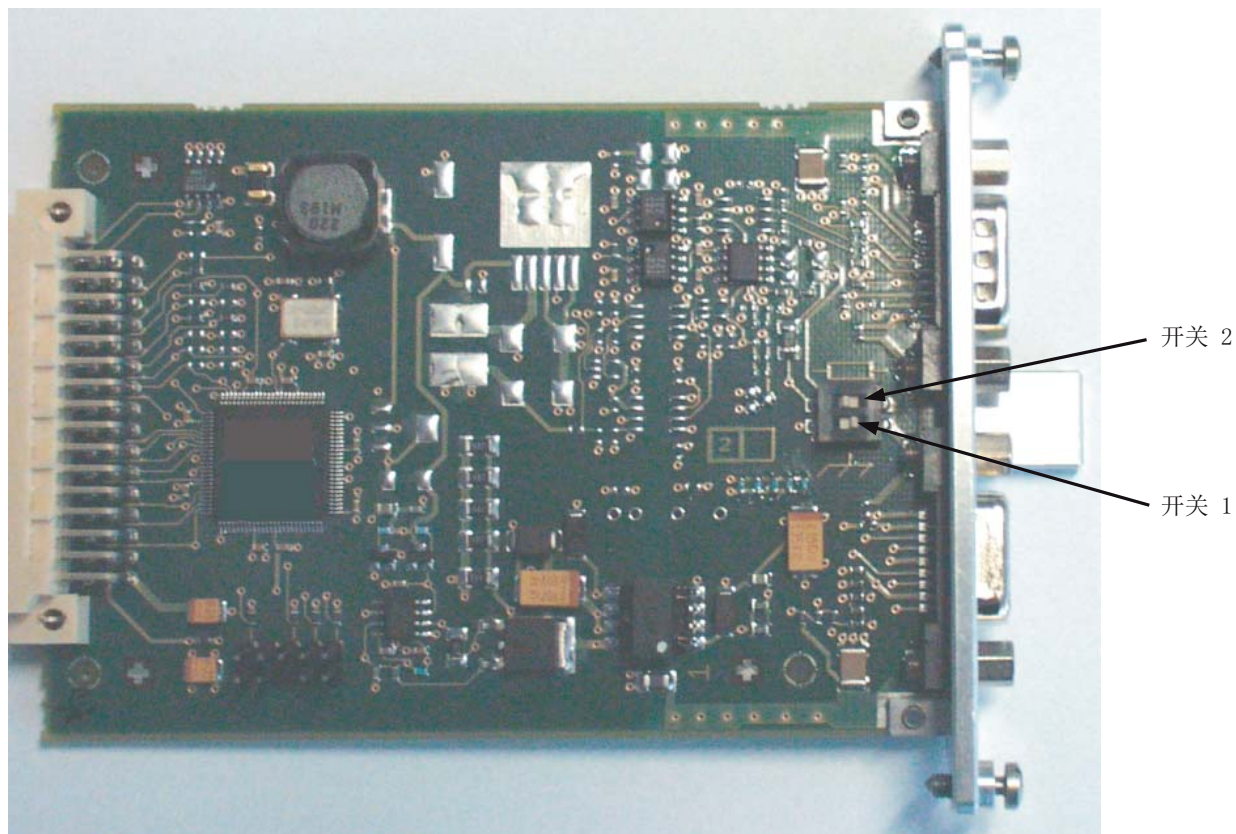
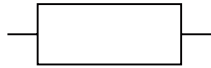
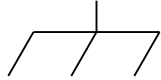


图 3-2 开关 2/1

表格 3-1 2 极 SMD DIL 开关

组件上的标记	开关	功能	开关位置		缺省
	2	总线终端电阻 120 Ω	OFF	无效	OFF
			ON	有效	
	1	不接地/接地 运行	OFF	不接地运行	OFF
			ON	接地运行	

说明

总线支路终端必须要有总线终端电阻，可使用 CBC10 上的总线终端电阻。

3.4 使用调试工具 **STARTER** 配置驱动设备

说明

在执行这些步骤时，请参考 **SINAMICS S120** 调试手册中的提示信息。

在调试工具 **STARTER** 中按以下步骤配置驱动设备：


1. 在线查找驱动设备。
驱动设备的组件拓扑结构和配置是自动确定的。
2. 配置电机。
3. 配置控制单元上“CBC10”的接口。
 - CAN 总线接口
 - PDO 报文
 - 监控
4. 将项目加载到驱动设备中。

3.4.1 在线搜索驱动设备

SINAMICS 固件能自动识别实际拓扑结构，并将其保存至相应的参数。

步骤

用 **STARTER** 打开一个新项目，以确保通过 **STARTER** 自动识别驱动设备配置。执行以下步骤：

1. 点击用户界面上的 **STARTER** 图标  以启动调试工具 **STARTER**。

或者

在 **Windows** 开始菜单中选择 “Start > SIMATIC > STEP 7 > **STARTER**” 来启动 **STARTER**。

STARTER 项目向导打开。

2. 点击按钮“Find drive units online...”。

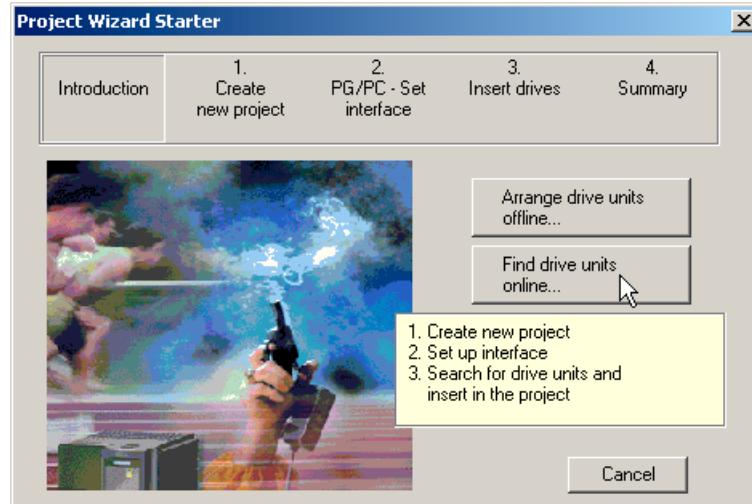


图 3-3 在线搜索驱动设备

项目向导会指引您创建新项目。

3. 在下一个对话框中输入项目名称，例如“Project_CANopen_0”，然后点击“Next >”。

项目向导会在线搜索驱动设备并将其添加至项目。

4. 点击“Continue >”。

项目辅助程序列出了一个项目的配置组合清单。

5. 点击“Complete”。

结果

在调试工具 *STARTER* 中会显示添加了相应驱动设备的新项目。



说明

STARTER 查找驱动设备，也就是控制单元。即：如果系统中存在多个控制单元，便会找到多个驱动设备。此时不显示驱动设备（控制单元、调节型电源模块等）的外部组件。

3.4.2 自动采集驱动设备组件拓扑结构和配置

在创建了项目并在线获取了驱动设备的总线地址（例如“126”）后，必须在线确定相应的驱动设备组件拓扑结构和配置。

步骤

1. 在项目导航器中选中驱动设备“Drive_Unit_Adr126”。
2. 点击图标“Connect to selected target devices” 。
- 建立在线连接并显示对话框“Online/OFFLINE comparison”。
3. 点击“Close”，如必要的话可恢复出厂设置。
4. 在项目导航器中选中驱动设备“Drive_Unit_Adr126”。
5. 点击图标“Restore factory settings” 。
6. 按下“OK”确认以下询问和信息：
 - 对话框“Restore factory settings？”
 - 对话框“The factory settings have been restored”
 - 对话框“The data has been successfully copied from RAM to ROM”
7. 在项目导航器中，双击驱动设备下的选项“Automatic configuration”。

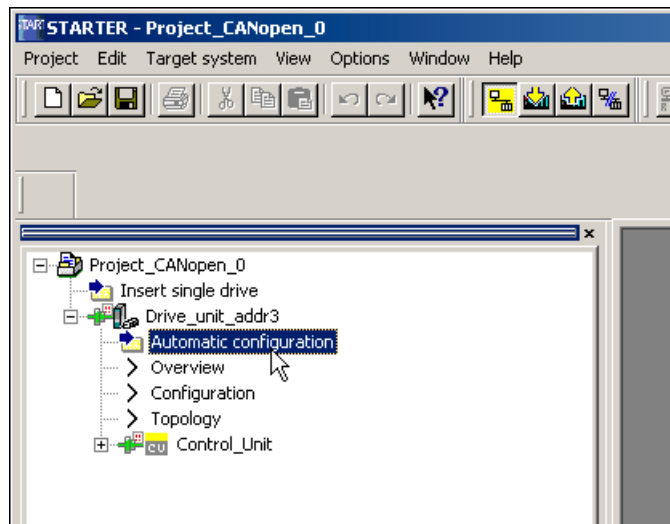


图 3-4 自动配置

- 8. 在“Automatic configuration”对话框中点击 “Configuration”。
调试工具 *STARTER* 会自动搜索所有正确连接的驱动设备组件，并自动将其加载到 *STARTER*。在此示例中识别出两个驱动对象。
- 9. 在对话框 “Drive object type”中选择 “Servo”。

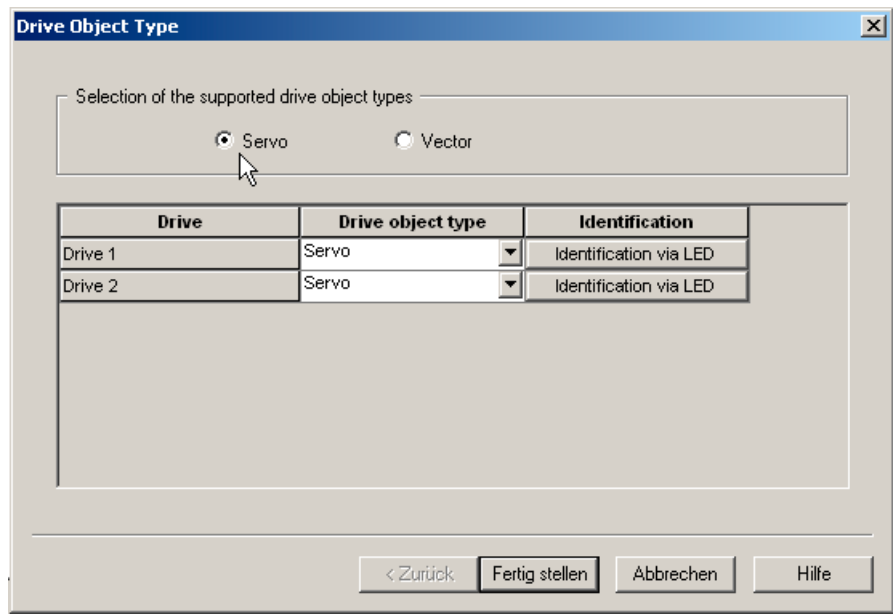


图 3-5 驱动对象类型

说明
在选择“Vector”时，和伺服一样通过协议速度模式和自由过程数据支持 CANopen。

- 10. 点击“Complete”。
系统将数据“从 RAM 加载到 ROM”，并“加载到 PG”。
接着会出现提示信息，提示还需对电机进行配置（参见 配置电机 (页 80)）。
- 11. 点击 “OK”，确认提示。
初始化已结束。
- 12. 在“Automatic configuration”对话框中点击 “Close”。

结果

自动配置完成。在项目导航器中会显示搜索到的所有驱动设备组件，例如“控制单元”、“CBC10”、“电源”和“驱动”。

下图中显示了在调试工具 **STARTER** 中完成了配置的驱动。在工作区中已选择设定和实际拓扑结构的视图。此外已识别出驱动组中的 **CBC10**。

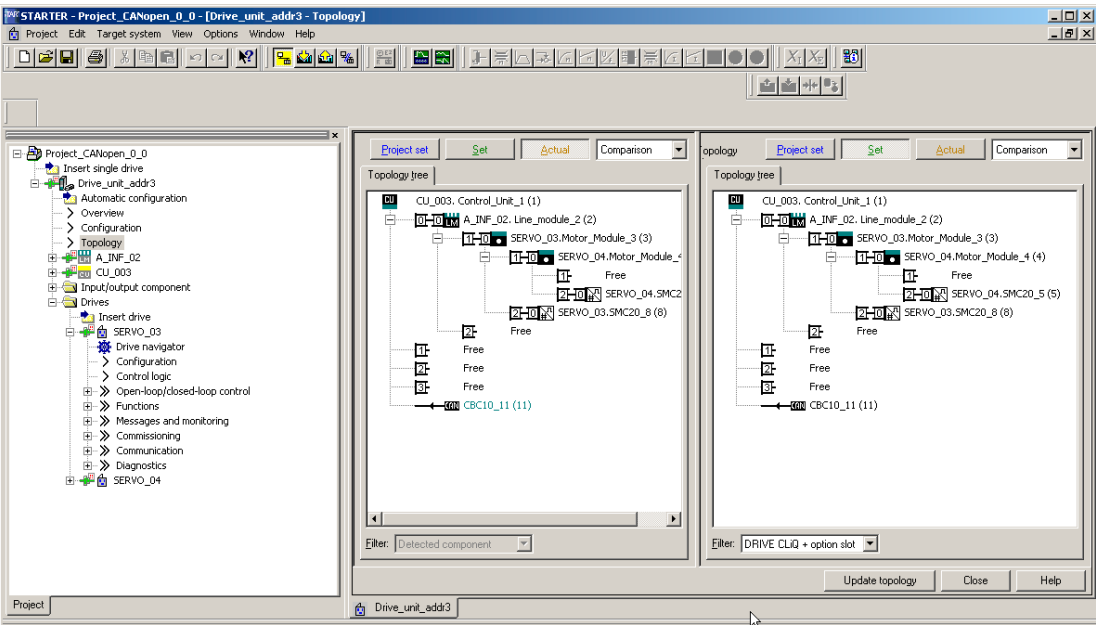


图 3-6 驱动配置的设定和实际拓扑结构（示例）

3.4.3 配置电机

在上一步中，您已经自动采集了驱动设备的组件拓扑结构和配置,并将它添加到 **STARTER** 项目中。

步骤

说明

只能修改电机和编码器的配置。其他组件（如：电源）的配置须保持不变。

下一步对驱动的电机和编码器进行配置。

1. 点击图标“Disconnect from target system...”。

修改的数据将从 RAM 加载到 ROM，并加载到 PG。离线执行电机配置，之后在线将配置加载到目标设备。

2. 在项目导航器中切换至驱动文件夹并双击相应驱动下的“Configuration”。

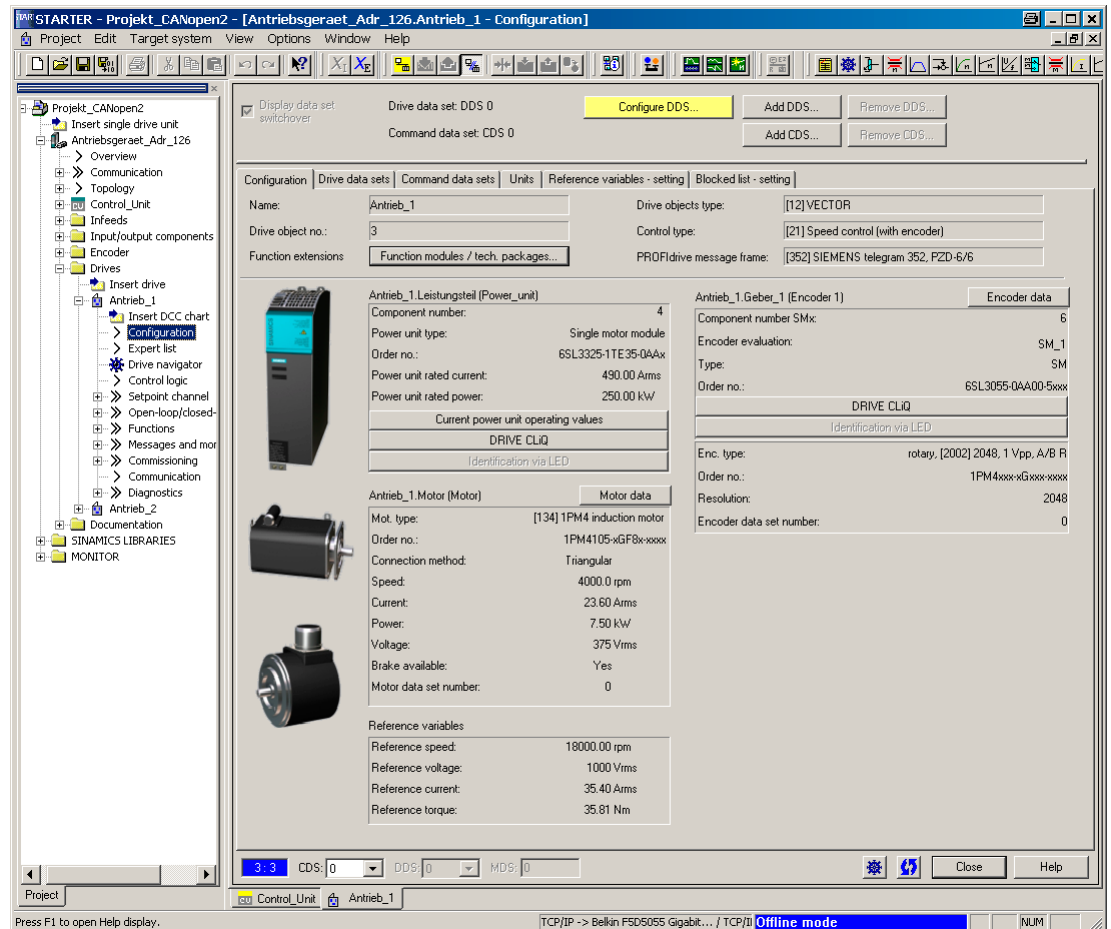


图 3-7 配置驱动（设置电机）

3. 在显示的对话框中点击“Configure DDS...”按钮。

在显示的对话框“configuration ...”中可设定，是否使用扩展设定值通道运行驱动对象（功能模块）。下面描述的调试步骤不使用扩展设定值通道（斜坡函数发生器）。扩展设定值通道一项必须取消选中。

说明

在斜坡函数发生器激活时（带设定值通道），可修改 CI: p2151 = r1119 互联，从而获取斜坡函数发生器前的设定值，用于状态字（r8784）中位 10（已达目标）的检测。
斜坡函数发生器生效时，驱动子协议的对象 6086 hex（运动协议类型）和 6083 hex（协议加速度）生效。

4. 点击“Continue >”，对电机进行配置。

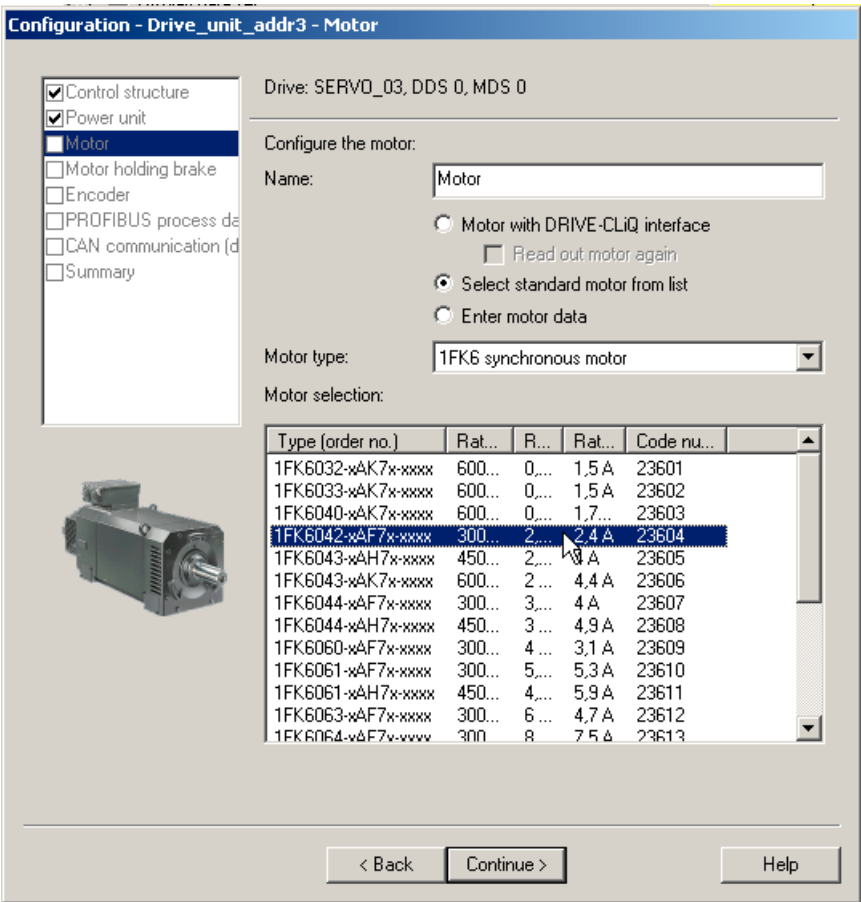


图 3-8 配置电机

5. 选择电机型号，并选择所需电机（订货号）（参见铭牌）。

6. 点击“Continue >”，对编码器进行配置。

7. 选择电机编码器，一直点击“Continue >”，直至显示概要对话框。
8. 点击“Complete”。

结果

第一个驱动的电机和编码器配置完成后。现在使用同样的步骤来配置第二个驱动。

参见

配置控制单元 CU320-2 上 CBC10 的接口 (页 84)

3.4.4 配置控制单元 CU320-2 上 CBC10 的接口

前提条件

已经在调试工具 STARTER 中对安装了 CBC10 的驱动设备进行了配置，并通过 STARTER 离线连接了设备。

步骤

1. 在项目导航器中分别双击“Drive unit_Adr_126”>“Control_Unit”>“CAN Option Board”>“Configuration”。

在之后显示的对话框中对 CBC10 进行配置。

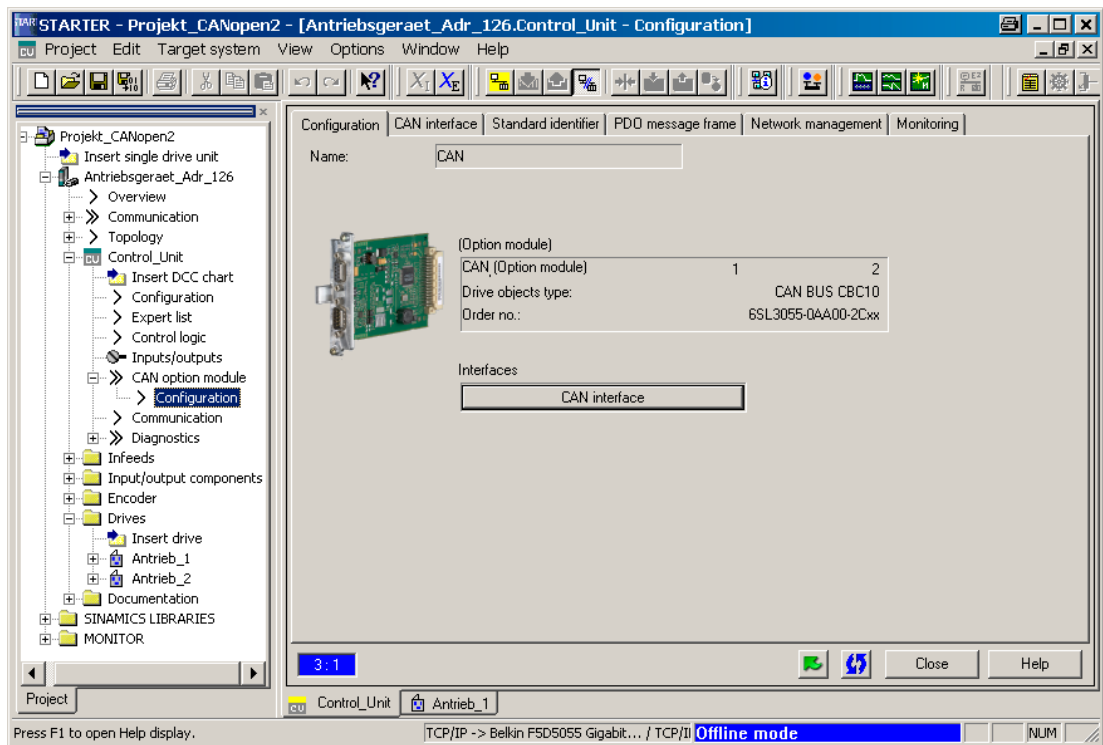


图 3-9 配置 CBC10

2. 在控制单元 CU320-2 上为 CBC10 配置以下传输属性：

- 传输速率
- CAN 总线地址 (页 85)(Node-ID)
- PDO 报文数量 (页 86)
- 节点监控 (页 87) (心跳，寿命保护)

3.4.4.1 CAN 总线接口

可在“CAN interface”标签中设置传输速率和 CAN 总线/Node-ID。

步骤

1. 选择“CAN interface”标签。

传输速率出厂设置为 20 kBit/s。

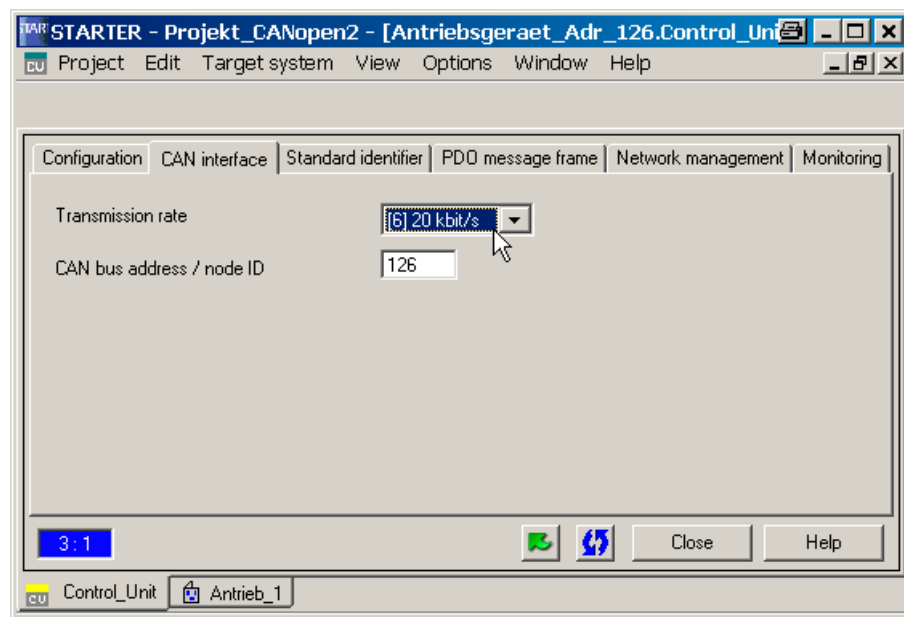


图 3-10 CAN 总线接口

2. 选择另一个调试传输速率(例如：“1 MBit/s”)。

说明

如果在调试期间重启了控制系统或执行了复位，则会恢复为出厂设置。

可通过以下两种方式设置 CAN 总线地址/Node-ID：

- 在此对话框中，如果控制单元上的地址开关 (“DP Address”) 被设为 0，则可设置 1 到 127 之间的值。
- 直接通过控制单元上的地址开关设置。
设置“总线地址/Node-ID”（参见示例）。

示例

通过控制单元上的地址开关设置 CAN 总线地址。

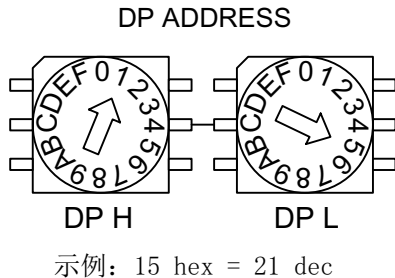


图 3-11 示例: 通过控制单元上的地址开关设置总线地址

说明

请注意以下信息:

- 允许的 CAN 总线地址: 1...127。
- 在开关上设置的地址在 p8620 中显示。
- 只有在重新上电后, 通过地址开关或 p8620 进行的更改才生效。
- SINAMICS 启动时, 系统会首先询问用于设置总线地址的地址开关。如果开关被设为 0, 则可通过参数 p8620 设置地址。
- 如果地址开关被设为生效的节点地址 (1...127), 则该值会传送到 p8620 并显示在其中。

3.4.4.2

PDO 报文

此处说明的首次调试是以具有缺省设置 (预定义连接集)、固定分配的 SDO 和 PDP 的设备为例。此时设备中的每个 SINAMICS 驱动对象 (“单轴电机模块”) 有:

- 四个接收 PDO
- 四个发送 PDO

在 CU320-2 中为每个电机模块缺省设置了四个接收 PDO 和四个发送 PDO。最多可设置八个发送 PDO 和接收 PDO。

激活“预定义连接集”

在离线模式下按以下步骤激活“预定义连接集”：

1. 在项目导航器中选中驱动；例如：“Drive_1”。
2. 通过右键菜单“Expert list”调用驱动的专家列表。
3. 搜索参数 p8744。
4. 选择“Predefined Connection Set (1)”进行设置。
5. 对下一个 SINAMICS 驱动对象“Drive_2”，重复执行步骤 1 至 4。

3.4.4.3 监控

SINAMICS 支持以下可选监控服务，用于确保 CANopen 网络节点的功能：

- “心跳”
SINAMICS（生产者）将其通讯状态循环（心跳时间）发送至 CAN 总线上的主站应用程序。
- “寿命保护”
SINAMICS 在特定的时间（节点寿命）内等待主站应用程序的报文，并允许特定时间（节点保护时间）内特定数量（寿命系数）的故障。
节点寿命通过节点保护时间和寿命系数相乘得出。

说明

对于“心跳”和“寿命保护”这两种节点监控服务，只能激活其中的一种。
两种监控服务相互冲突。

步骤

1. 选择“Monitoring”标签。

传输机制	调试的缺省设置
心跳	100 ms
寿命保护	时间间隔（保护时间） = 100 ms
	故障数量（寿命系数） = 3

2. 确定一种传输机制并进行必要的设置。

故障情况

如果 CAN 通讯中发生故障，例如出现太多报文故障，则显示故障信息 F(A)08700(2)（详细信息参见 SINAMICS S120/S150 参数手册）。该故障在参数 r0949 中显示。使用 p8641 对驱动的故障响应进行设置。使用 p8609 对 CAN 节点的响应进行设置。

说明

参数 p8609 和 p8641 用于定义发生 CAN 通讯故障或设备故障时，驱动以及 CAN 节点的特性。

参数的出厂设置为

p8609 = 1, => 不变更

p8641 = 0, => 不响应

对于各驱动对象，这些参数可设为不同的值。

参数 p8609

设置发生通讯故障或设备故障时 CAN 节点的特性。

- 数值：
 - 0: 预操作
 - 1: 不变更（出厂设置）
 - 2: 停止
- 索引（对应 CANopen 对象 1029 hex）：
 - [0] = 通讯故障时的特性
 - [1] = 设备故障时的特性

参数 p8641

设置 CAN 总线通讯故障时驱动的特性。

- 数值：
 - 0: 不响应（出厂设置）
 - 1: OFF1
 - 2: OFF2
 - 3: OFF3

3.4.5 将项目加载到驱动设备

步骤

1. 点击图标 “Connect to selected target devices” 。

建立在线连接，并进行在线/离线对比。如果识别出不一致性，则会在相应的栏中显示（见下图）。

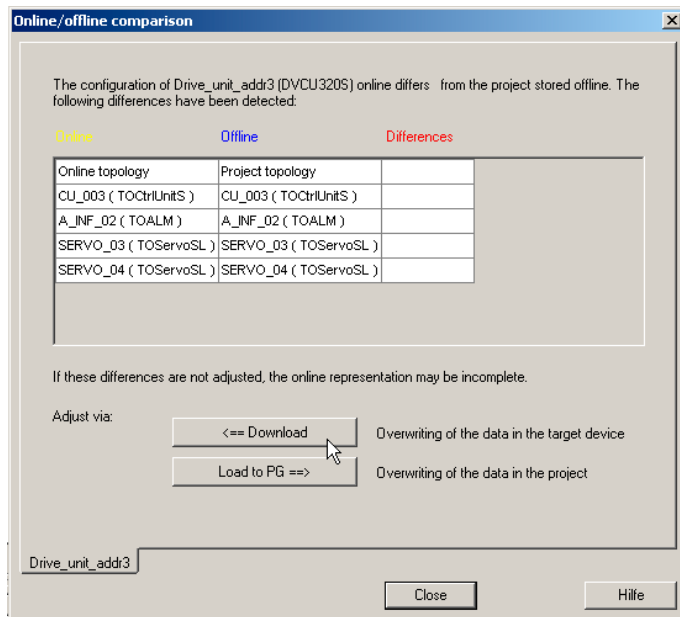


图 3-12 在线/离线对比（示例）

如果离线修改了数据，必须将其加载到目标设备中。

2. 在“ONLINE/OFFLINE comparison”对话框中点击按钮“<-- Download”。
3. 在系统询问“Are you sure?”时点击“Yes”。
4. 在“the data was successfully loaded to the target system”对话框中点击“OK”。
5. 在从 RAM 加载到 ROM 时点击“OK”。

如果在在线/离线对比中再次识别出差异。

6. 点击按钮“Load to PG -->”。

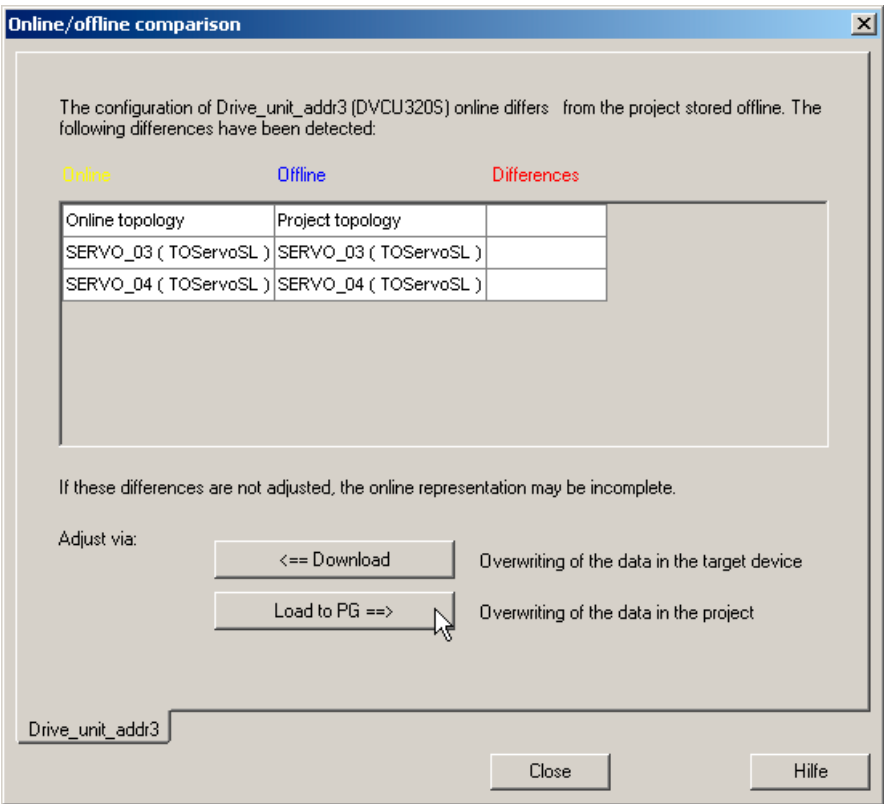


图 3-13 加载到 PG

7. 在系统询问“Are you sure?”时点击“Yes”。

8. 在“the data was successfully loaded to the PG”对话框中点击“OK”。

“ONLINE/OFFLINE comparison”对话框中不再显示差异。

9. 点击“Close”。

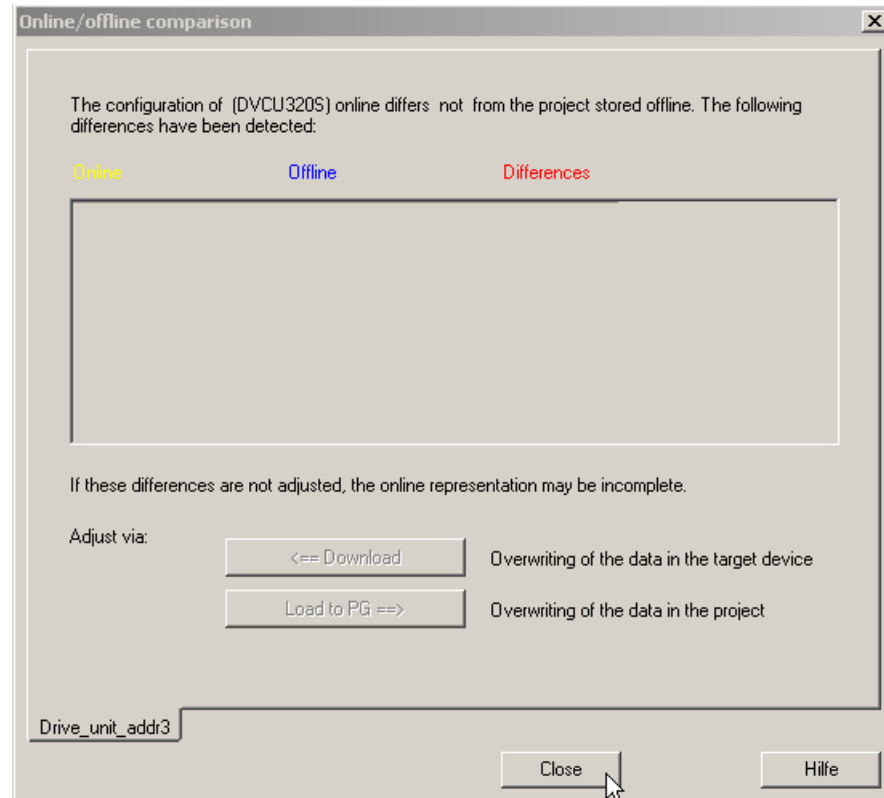


图 3-14 在线/离线对比（数据一致）

结果

具有 CANopen 接口的驱动设备硬件配置完成。在下一步配置接收/发送报文的 COB-ID 和过程数据对象前，必须互联电源。

3.4.6 互联电源

在当前软件版本中，电源（调节型电源模块）不能通过 PDO 直接访问。

为了使用调节型电源模块参数 p0840 (ON/OFF1)通过信号源接通直流母线，可进行以下 BICO 互联：

- 将参数 p0840 与控制单元数字量输入上的一个自由位（例如 r0722 位 15）互联。
- 将参数 p0840 与电机模块 1 控制字的一个自由位（例如 r8890 位 15）互联。

说明

在说明的配置示例中，控制字位于接收缓冲器的 PZD 控制字 1 中（参见附录中的功能图）。

说明

在调试工具 *STARTER* 中进行 BICO 互联的基本步骤请参见“在 *STARTER* 中进行 BICO 互联的基本步骤 (页 61)”章节。

步骤

互联电源参数 p0840 (ON/OFF1)与信号源，例如电机模块 1 (Drive_1)的控制字：

1. 在项目导航器中选中选项“Infeed”。
2. 通过右键菜单菜单“Expert > Expert list”调用专家列表并搜索参数 p0840。
3. 将 p 参数与 r 参数 r8890 位 15，即“Drive_1” (SERVO_03) 的 PZD 1 互联。

说明

此外也请遵循 *SINAMICS S120* 调试手册中的调试提示。

例如在接通电机模块前，必须首先接通直流母线。

3.5 配置接收 PDO 报文和发送 PDO 报文

需要为各驱动对象单独调试通讯和映射参数。此时调试工具 **STARTER** 在线连接至驱动设备。可对各驱动对象的接收/发送报文的以下内容进行自由（自由 PDO 映射）配置：

- 通讯参数
- 映射参数

说明

首次调试时自动预设参数

在 **CANopen** 接口的首次调试时，通常情况下各驱动对象的通讯参数、映射参数会自动预设（“预定义连接集”）。下面描述的调试步骤假设已进行预设，并需要对参数进行修改。如果您不需要参数设置自由 PDO 映射，可跳过此步骤并参考“互联接收/发送缓冲器 (页 104)”章节继续进行调试。

3.5.1 驱动对象的预设 COB-ID 和过程数据对象

打开专家列表

在自由 PDO 映射开始前，按照以下步骤分别打开各驱动对象的专家列表：

1. 在项目导航器中选中驱动；例如：“Drive_1”。
2. 通过右键菜单“Expert list”调用驱动对象 1 的专家列表。
3. 在项目导航器中选中驱动；例如：“Drive_2”。
4. 通过右键菜单“Expert list”调用驱动对象 2 的专家列表。

此时在调试工具 **STARTER** 中显示两个驱动对象的专家列表。可查看“预定义连接集”中的自动预设参数。

说明

在每个驱动对象的专家列表中设定 COB-ID，接收报文从 p8700 起，发送报文从 p8720 起。

在每个驱动对象的专家列表中，接收报文的映射过程数据对象从参数 p8710 起，发送报文从参数 p8730 起。

3.5 配置接收 PDO 报文和发送 PDO 报文

驱动对象 1 的过程数据对象和 COB-ID 之间的映射

下图展示了调试工具 STARTER 专家列表中、驱动对象 1 的发送/接收报文的过程数据对象和预设的 COB-ID 之间的映射。

Expertenliste				Online-Wert SERV0_02
Parameter	D	+	Parametertext	
p8700[0]	-		CBC Receive PDO 1, COB-ID des PDO	20AH
p8700[1]	-		CBC Receive PDO 1, Transmission Type des PDO	FEH
p8701[0]	-		CBC Receive PDO 2, COB-ID des PDO	30AH
p8701[1]	-		CBC Receive PDO 2, Transmission Type des PDO	FEH
p8702[0]	-		CBC Receive PDO 3, COB-ID des PDO	40AH
p8702[1]	-		CBC Receive PDO 3, Transmission Type des PDO	FEH
p8703[0]	-		CBC Receive PDO 4, COB-ID des PDO	50AH
p8703[1]	-		CBC Receive PDO 4, Transmission Type des PDO	FEH
p8704[0]	+		CBC Receive PDO 5, COB-ID des PDO	800006DFH
p8705[0]	+		CBC Receive PDO 6, COB-ID des PDO	800006DFH
p8706[0]	+		CBC Receive PDO 7, COB-ID des PDO	800006DFH
p8707[0]	+		CBC Receive PDO 8, COB-ID des PDO	800006DFH
p8710[0]	-		CBC Receive Mapping für RPDO 1, Gemapptes Objk	60400010H
p8710[1]	-		CBC Receive Mapping für RPDO 1, Gemapptes Objk	0H
p8710[2]	-		CBC Receive Mapping für RPDO 1, Gemapptes Objk	0H
p8710[3]	-		CBC Receive Mapping für RPDO 1, Gemapptes Objk	0H
p8711[0]	-		CBC Receive Mapping für RPDO 2, Gemapptes Objk	60400010H
p8711[1]	-		CBC Receive Mapping für RPDO 2, Gemapptes Objk	60FF0020H
p8711[2]	-		CBC Receive Mapping für RPDO 2, Gemapptes Objk	0H
p8711[3]	-		CBC Receive Mapping für RPDO 2, Gemapptes Objk	0H
p8712[0]	-		CBC Receive Mapping für RPDO 3, Gemapptes Objk	60400010H
p8712[1]	-		CBC Receive Mapping für RPDO 3, Gemapptes Objk	60710010H
p8712[2]	-		CBC Receive Mapping für RPDO 3, Gemapptes Objk	0H
p8712[3]	-		CBC Receive Mapping für RPDO 3, Gemapptes Objk	0H
p8713[0]	-		CBC Receive Mapping für RPDO 4, Gemapptes Objk	60400010H
p8713[1]	-		CBC Receive Mapping für RPDO 4, Gemapptes Objk	60FF0020H
p8713[2]	-		CBC Receive Mapping für RPDO 4, Gemapptes Objk	60710010H
p8713[3]	-		CBC Receive Mapping für RPDO 4, Gemapptes Objk	0H
p8714[0]	+		CBC Receive Mapping für RPDO 5, Gemapptes Objk	0H
p8715[0]	+		CBC Receive Mapping für RPDO 6, Gemapptes Objk	0H
p8716[0]	+		CBC Receive Mapping für RPDO 7, Gemapptes Objk	0H
p8717[0]	+		CBC Receive Mapping für RPDO 8, Gemapptes Objk	0H

图 3-15 驱动对象 1 接收报文的过程数据对象和 COB-ID 之间的映射

Expertenliste				Wert SERV0_03
Parameter	D	+	Parametertext	
p8720[0]	+		CBC Transmít PDO 1, COB-ID des PDO	40000183H
p8721[0]	+		CBC Transmít PDO 2, COB-ID des PDO	40000283H
p8722[0]	+		CBC Transmít PDO 3, COB-ID des PDO	40000383H
p8723[0]	+		CBC Transmít PDO 4, COB-ID des PDO	40000483H
p8724[0]	+		CBC Transmít PDO 5, COB-ID des PDO	C00006E0H
p8725[0]	+		CBC Transmít PDO 6, COB-ID des PDO	C00006E0H
p8726[0]	+		CBC Transmít PDO 7, COB-ID des PDO	C00006E0H
p8727[0]	+		CBC Transmít PDO 8, COB-ID des PDO	C00006E0H
p8730[0]	-		CBC Transmít Mapping für TPDO 1, Gemapptes Objekt 1	60410010H
p8730[1]	-		CBC Transmít Mapping für TPDO 1, Gemapptes Objekt 2	0H
p8730[2]	-		CBC Transmít Mapping für TPDO 1, Gemapptes Objekt 3	0H
p8730[3]	-		CBC Transmít Mapping für TPDO 1, Gemapptes Objekt 4	0H
p8731[0]	-		CBC Transmít Mapping für TPDO 2, Gemapptes Objekt 1	60410010H
p8731[1]	-		CBC Transmít Mapping für TPDO 2, Gemapptes Objekt 2	606C0020H
p8731[2]	-		CBC Transmít Mapping für TPDO 2, Gemapptes Objekt 3	0H
p8731[3]	-		CBC Transmít Mapping für TPDO 2, Gemapptes Objekt 4	0H
p8732[0]	-		CBC Transmít Mapping für TPDO 3, Gemapptes Objekt 1	60410010H
p8732[1]	-		CBC Transmít Mapping für TPDO 3, Gemapptes Objekt 2	60740010H
p8732[2]	-		CBC Transmít Mapping für TPDO 3, Gemapptes Objekt 3	0H
p8732[3]	-		CBC Transmít Mapping für TPDO 3, Gemapptes Objekt 4	0H
p8733[0]	-		CBC Transmít Mapping für TPDO 4, Gemapptes Objekt 1	60410010H
p8733[1]	-		CBC Transmít Mapping für TPDO 4, Gemapptes Objekt 2	60630020H
p8733[2]	-		CBC Transmít Mapping für TPDO 4, Gemapptes Objekt 3	0H
p8733[3]	-		CBC Transmít Mapping für TPDO 4, Gemapptes Objekt 4	0H
p8734[0]	+		CBC Transmít Mapping für TPDO 5, Gemapptes Objekt 1	0H
p8735[0]	+		CBC Transmít Mapping für TPDO 6, Gemapptes Objekt 1	0H
p8736[0]	+		CBC Transmít Mapping für TPDO 7, Gemapptes Objekt 1	0H
p8737[0]	+		CBC Transmít Mapping für TPDO 8, Gemapptes Objekt 1	0H
p8744			CBC PDO Mapping Konfiguration	Freies PDO Mapping (2)

图 3-16 驱动对象 1 发送报文的过程数据对象和 COB-ID 之间的映射

驱动对象 2 的过程数据对象和 COB-ID 之间的映射

下图展示了调试工具 STARTER 专家列表中、驱动对象 2 的发送/接收报文的过程数据对象和预设的 COB-ID 之间的映射。设置第二个驱动对象的过程数据对象时，向后推移 800 hex。

Parameter	D	+	+	Parametertext	Online-Wert SERVO_04
p8700[0]	-			CBC Receive PDO 1, COB-ID des PDO	40000206H
p8700[1]	-			CBC Receive PDO 1, Transmission Type des PDO	FEH
p8701[0]	-			CBC Receive PDO 2, COB-ID des PDO	40000306H
p8701[1]	-			CBC Receive PDO 2, Transmission Type des PDO	FEH
p8702[0]	-			CBC Receive PDO 3, COB-ID des PDO	40000406H
p8702[1]	-			CBC Receive PDO 3, Transmission Type des PDO	FEH
p8703[0]	-			CBC Receive PDO 4, COB-ID des PDO	40000506H
p8703[1]	-			CBC Receive PDO 4, Transmission Type des PDO	FEH
p8704[0]	+			CBC Receive PDO 5, COB-ID des PDO	C00006E0H
p8705[0]	+			CBC Receive PDO 6, COB-ID des PDO	C00006E0H
p8706[0]	+			CBC Receive PDO 7, COB-ID des PDO	C00006E0H
p8707[0]	+			CBC Receive PDO 8, COB-ID des PDO	C00006E0H
p8710[0]	-			CBC Receive Mapping für RPDO 1, Gemapptes Obj	68400010H
p8710[1]	-			CBC Receive Mapping für RPDO 1, Gemapptes Obj	0H
p8710[2]	-			CBC Receive Mapping für RPDO 1, Gemapptes Obj	0H
p8710[3]	-			CBC Receive Mapping für RPDO 1, Gemapptes Obj	0H
p8711[0]	-			CBC Receive Mapping für RPDO 2, Gemapptes Obj	68400010H
p8711[1]	-			CBC Receive Mapping für RPDO 2, Gemapptes Obj	68FF0020H
p8711[2]	-			CBC Receive Mapping für RPDO 2, Gemapptes Obj	0H
p8711[3]	-			CBC Receive Mapping für RPDO 2, Gemapptes Obj	0H
p8712[0]	-			CBC Receive Mapping für RPDO 3, Gemapptes Obj	68400010H
p8712[1]	-			CBC Receive Mapping für RPDO 3, Gemapptes Obj	68710010H
p8712[2]	-			CBC Receive Mapping für RPDO 3, Gemapptes Obj	0H
p8712[3]	-			CBC Receive Mapping für RPDO 3, Gemapptes Obj	0H
p8713[0]	-			CBC Receive Mapping für RPDO 4, Gemapptes Obj	68400010H
p8713[1]	-			CBC Receive Mapping für RPDO 4, Gemapptes Obj	68FF0020H
p8713[2]	-			CBC Receive Mapping für RPDO 4, Gemapptes Obj	68710010H
p8713[3]	-			CBC Receive Mapping für RPDO 4, Gemapptes Obj	0H
p8714[0]	+			CBC Receive Mapping für RPDO 5, Gemapptes Obj	0H
p8715[0]	+			CBC Receive Mapping für RPDO 6, Gemapptes Obj	0H
p8716[0]	+			CBC Receive Mapping für RPDO 7, Gemapptes Obj	0H
p8717[0]	+			CBC Receive Mapping für RPDO 8, Gemapptes Obj	0H

图 3-17 驱动对象 2 接收报文的过程数据对象和 COB-ID 之间的映射

Parameter	D	+	+	Parametertext	Online-Wert SERVO_04
p8720[0]	+			CBC Transmit PDO 1, COB-ID des PDO	40000186H
p8721[0]	+			CBC Transmit PDO 2, COB-ID des PDO	40000286H
p8722[0]	+			CBC Transmit PDO 3, COB-ID des PDO	40000386H
p8723[0]	+			CBC Transmit PDO 4, COB-ID des PDO	40000486H
p8724[0]	+			CBC Transmit PDO 5, COB-ID des PDO	C00006E0H
p8725[0]	+			CBC Transmit PDO 6, COB-ID des PDO	C00006E0H
p8726[0]	+			CBC Transmit PDO 7, COB-ID des PDO	C00006E0H
p8727[0]	+			CBC Transmit PDO 8, COB-ID des PDO	C00006E0H
p8730[0]	-			CBC Transmit Mapping für TPDO 1, Gemapptes Obj	68410010H
p8730[1]	-			CBC Transmit Mapping für TPDO 1, Gemapptes Obj	0H
p8730[2]	-			CBC Transmit Mapping für TPDO 1, Gemapptes Obj	0H
p8730[3]	-			CBC Transmit Mapping für TPDO 1, Gemapptes Obj	0H
p8731[0]	-			CBC Transmit Mapping für TPDO 2, Gemapptes Obj	68410010H
p8731[1]	-			CBC Transmit Mapping für TPDO 2, Gemapptes Obj	686C0020H
p8731[2]	-			CBC Transmit Mapping für TPDO 2, Gemapptes Obj	0H
p8731[3]	-			CBC Transmit Mapping für TPDO 2, Gemapptes Obj	0H
p8732[0]	-			CBC Transmit Mapping für TPDO 3, Gemapptes Obj	68410010H
p8732[1]	-			CBC Transmit Mapping für TPDO 3, Gemapptes Obj	68740010H
p8732[2]	-			CBC Transmit Mapping für TPDO 3, Gemapptes Obj	0H
p8732[3]	-			CBC Transmit Mapping für TPDO 3, Gemapptes Obj	0H
p8733[0]	-			CBC Transmit Mapping für TPDO 4, Gemapptes Obj	68410010H
p8733[1]	-			CBC Transmit Mapping für TPDO 4, Gemapptes Obj	68630020H
p8733[2]	-			CBC Transmit Mapping für TPDO 4, Gemapptes Obj	0H
p8733[3]	-			CBC Transmit Mapping für TPDO 4, Gemapptes Obj	0H
p8734[0]	+			CBC Transmit Mapping für TPDO 5, Gemapptes Obj	0H
p8735[0]	+			CBC Transmit Mapping für TPDO 6, Gemapptes Obj	0H
p8736[0]	+			CBC Transmit Mapping für TPDO 7, Gemapptes Obj	0H
p8737[0]	+			CBC Transmit Mapping für TPDO 8, Gemapptes Obj	0H
p8744				CBC PDO Mapping Konfiguration	Freies PDO Mapping (2)

图 3-18 驱动对象 2 发送报文的过程数据对象和 COB-ID 之间的映射

3.5.2 激活“自由 PDO 映射”

步骤

执行以下步骤，在 SINAMICS 驱动对象 2（单轴电机模块 2）中激活“自由 PDO 映射”：

- 1. 在项目导航器中选中驱动；例如：“Drive_2”。
- 2. 通过右键菜单“Expert list” 调用驱动对象 2 的专家列表。
- 3. 选中参数 p8744。

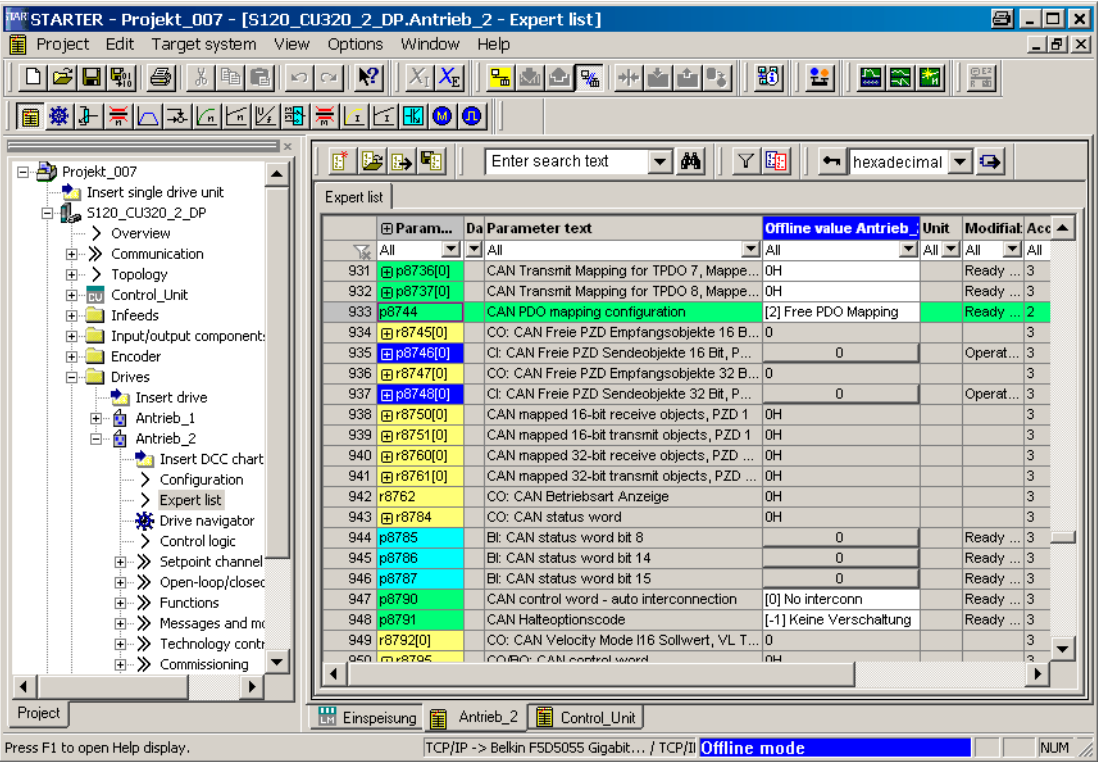


图 3-19 自由 PDO 映射

- 4. 选择设置“Free PDO mapping (2)”。

3.5.3 在自由 PDO 映射中设定 COB-ID 和映射参数

如果处于在线模式并希望自由设定 COB-ID 和映射参数，可进行以下操作：

- COB-ID
在以下参数中，您可自由设定 COB：
 - 参数 p8700 至 p8707 中的接收 (RPDO)
 - 参数 p8720 至 p8727 中的发送 (TPDO)
- 映射参数：
在以下参数中可输入传输报文需要映射的过程数据对象：
 - 接收报文 (RPDO)，从参数 p8710 起
 - 发送报文 (TPDO)，从参数 p8730 起

说明

如果在调试工具 STARTER 在线模式中修改了映射参数，必须在此之前将通讯参数中相关 PDO 的 COB-ID 设置为无效，然后在修改后重新设置为有效。

COB-ID，位 31 = 0 -> COB-ID 有效，

位 31 = 1 -> COB-ID 无效

基本步骤

在线进行自由 PDO 映射时，执行以下步骤：

1. 将相关 RPDO 或 TPDO 的 COB-ID 设置为无效（例如：p8700[0]，PDO 的 COB-ID）。
2. 在相关 RPDO 或 TPDO 中输入过程数据对象作为映射参数（例如：从 p8710[0]起，映射的对象）。
3. 将相关 RPDO 或 TPDO 的 COB-ID 设置为无效。
4. 通过 p8741 激活当前映射配置。

操作步骤

下面将通过示例描述操作步骤。此处的示例为驱动对象 2 的 RPDO1。

将 COB-ID 设置为无效

1. 在项目导航器中选中驱动；例如：“Drive_2”。
2. 通过右键菜单“Expert list”调用单轴电机模块 2 (Drive_2) 的专家列表。
3. 搜索通讯参数 RPDO1 的 COB-ID 参数 p8700[0]。
4. 将调试工具 STARTER 中的十六进制值复制到计算器并输入 **0r 8000 0000 hex**。将结果复制到 STARTER 中的参数。

此时位 31 被设为无效。

现在可匹配映射参数。

匹配映射参数

1. 在表格“驱动子协议 DSP402 中的对象”（参见“通讯对象 (页 159)”章节）中选择需要映射的过程数据对象（例如 RPDO1，控制字 = 6040 hex）。
2. 各个 SINAMICS 驱动对象加上相应的偏移（例如从驱动对象 2 起加上 800 hex）。参见下面的参数表中“表格值”一栏的对象字典索引(hex)（例如 6840 hex）。
3. 将对象字典索引（位 31...16）、子索引（位 15...8）和对象大小（位 7...0）转换为十六进制值（32 位）。参见下面的参数设置表中“得到的十六进制值”一栏（例如 6840 0010 hex）。
4. 将此值输入调试工具 STARTER 中相应的参数。参见下面的参数设置表中“STARTER 中的映射参数”一栏（例如 p8710[0]）。

将 COB-ID 设置为有效

- 1. 在项目导航器中选中驱动；例如：“Drive_2”。
- 2. 通过右键菜单“Expert list”调用单轴电机模块 2 的专家列表。
- 3. 搜索通讯参数 RPDO1 的 COB-ID 参数 p8700[0]。
- 4. 将调试工具 STARTER 中的十六进制值复制到计算器并输入 And 7FFF FFFF hex。
将结果复制到 STARTER 中的参数。此时位 31 被设为有效。
- 5. 将此值复制到 STARTER。

说明

下面所述的所有 RPDO 和 TPDO 都用于描述自由 PDO 映射。可自行选择传输报文需要映射的过程数据对象的方式和方法。

RPDO1 映射

在表格“驱动子协议 DSP402 中的对象”中选择了控制字的过程数据对象后，会得到以下值，这些值应输入到 RPDO1 映射参数 p8710[0] 至 p8710[3] 中，参见下表中的“得到的十六进制值”一栏：

说明

由于是 SINAMICS 驱动对象 2，因此索引上再加上 800 hex。

表格 3-2 RPDO1 时的值

对象名称	“驱动子协议 DSP402 中的对象”表中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	RPDO1 参数
控制字	6840	00	10 hex (16 位)	6840 0010	p8710[0]
无对象	-	-	-	0	p8710[1...3]

RPDO2 映射

在表格“驱动子协议 DSP402 中的对象”中选择了控制字的过程数据对象和转速设定值后，会得到以下数值，这些数值应输入到 RPDO2 映射参数 p8711[0...3] 中，参见下表中的“得到的十六进制值”一栏：

表格 3-3 RPDO2 时的值

对象名称	“驱动子协议 DSP402 中的对象”表中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	RPDO2 参数
控制字	6840	00	10 hex (16 位)	6840 0010	p8711[0]
转速设定值	68FF	00	20 hex (32 位)	68FF 0020	p8711[1]
无对象	-	-	-	0	p8711[2...3]

RPDO3 映射

在表格“驱动子协议 DSP402 中的对象”中选择了控制字的过程数据对象和转矩设定值后，会得到以下数值，这些数值应输入到 RPDO3 映射参数 p8712[0...3] 中，参见下表中的“得到的十六进制值”一栏：

表格 3-4 RPDO3 时的值

对象名称	“驱动子协议 DSP402 中的对象”表中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	RPDO3 参数
控制字	6840	00	10 hex (16 位)	6840 0010	p8712[0]
转矩设定值	6871	00	10 hex (16 位)	6871 0010	p8712[1]
无对象	-	-	-	0	p8712[2...3]

RPDO4 映射

在表格“驱动子协议 DSP402 中的对象”中选择了控制字的过程数据对象以及转速设定值和转矩设定值后，会得到以下数值，这些数值应输入到 RPDO4 映射参数 p8713[0...3] 中，参见下表中的“得到的十六进制值”一栏：

表格 3-5 RPDO4 时的值

对象名称	“驱动子协议 DSP402 中的对象”表 中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	RPDO4 参数
控制字	6840	00	10 hex (16 位)	6840 0010	p8713[0]
转速设定值	68FF	00	20 hex (32 位)	68FF 0020	p8713[1]
转矩设定值	6871	00	10 hex (16 位)	6871 0010	p8713[2]
无对象	-	-	-	0	P8713[3]

TPDO1 映射

在表格“驱动子协议 DSP402 中的对象”中选择了 CBC 状态字的过程数据对象后，会得到以下数值，这些数值应输入到 TPDO1 映射参数 p8730[0...3] 中，参见下表中的“得到的十六进制值”一栏：

表格 3-6 TPDO1 时的值

对象名称	“驱动子协议 DSP402 中的对象”表 中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	TPDO1 参数
CBC 状态字	6841	00	10 hex (16 位)	6841 0010	p8730[0]
无对象	-	-	-	0	p8730[1...3]

TPDO2 映射

在表格“驱动子协议 DSP402 中的对象”中选择了 CBC 状态字的过程数据对象和转速设定值后，会得到以下数值，这些数值应输入到 TPDO2 映射参数 p8731[0...3] 中，参见下表中的“得到的十六进制值”一栏：

表格 3-7 TPDO2 时的值

对象名称	“驱动子协议 DSP402 中的对象”表中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	TPDO2 参数
CBC 状态字	6841	00	10 hex (16 位)	6841 0010	p8731[0]
转速实际值	686C	00	20 hex (32 位)	686C 0020	p8731[1]
无对象	-	-	-	0	p8731[2...3]

TPDO3 映射

在表格“驱动子协议 DSP402 中的对象”中选择了 CBC 状态字的过程数据对象和转矩实际值后，会得到以下数值，这些数值应输入到 TPDO3 映射参数 p8732[0...3] 中，参见下表中的“得到的十六进制值”一栏：

表格 3-8 TPDO3 时的值

对象名称	“驱动子协议 DSP402 中的对象”表中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	TPDO3 参数
CBC 状态字	6841	00	10 hex (16 位)	6841 0010	p8732[0]
转矩实际值	6874	00	10 hex (16 位)	6874 0010	p8732[1]
无对象	-	-	-	0	p8732[2...3]

TPDO4 映射

在表格“驱动子协议 DSP402 中的对象”中选择了 CBC 状态字的过程数据对象和编码器位置实际值后，会得到以下数值，这些数值应输入到 TPDO4 映射参数 p8733[0...3] 中，参见下表中的“得到的十六进制值”一栏：

表格 3-9 TPDO4 时的值

对象名称	“驱动子协议 DSP402 中的对象”表中的值			STARTER 中的映射参数	
	对象字典索引 (hex)	子索引 (hex)	对象大小	得到的十六进制值	TPDO4 参数
CBC 状态字	6841	00	10 hex (16 位)	6841 0010	p8733[0]
位置实际值	6863	00	20 hex (32 位)	6863 0020	p8733[1]
无对象	-	-	-	0	p8733[2...3]

3.6 互联接收/发送缓冲器中的过程数据

传输报文的过程数据对象必须通过 BICO 互联，用于 SINAMICS 和 CAN 总线之间的接口。首次调试时执行以下步骤：

1. 读取接收/发送缓冲器中 PZD 接收/发送字的过程数据对象映射。
2. 将接收缓冲器中 PZD 接收字的源参数与过程数据对象的 SINAMICS 目标参数互联。
3. 将过程数据对象的 SINAMICS 源参数与发送缓冲器中的 PZD 发送字的目标参数互联。

说明

调试工具 STARTER 处于在线模式。

3.6.1 读取接收/发送缓冲器中单个过程数据对象的映射

如果设置了 PDO 映射，则设备会自动识别单个过程数据对象与 PZD 接收/发送字如何分配。

执行以下步骤，可以读取接收/发送缓冲器中 PZD 接收/发送字的单个过程数据对象的映射。

示例

说明

每个驱动对象都有独立的接收和发送缓冲器用于传输报文。

驱动对象的专家列表中包含以下参数的映射：

- 16 位过程数据对象：
 - 接收，从参数 r8750[0] 起
 - 发送，从参数 r8751[0] 起
- 32 位过程数据对象：
 - 接收，从参数 r8760[0] 起
 - 发送，从参数 r8761[0] 起

操作步骤，以驱动对象 2 为例

- 1. 在项目导航器中选中驱动；例如：“Drive_2”。
- 2. 通过右键菜单“Expert list”调用驱动对象 2 的专家列表。
- 3. 搜索参数 r8750[0]，该参数在接收缓冲器中有 16 位过程数据的映射。映射示例见下图：
 - PZD 1 中的控制字（6840 hex），和
 - PZD 4 中的转矩设定值（6871 hex）
- 4. 搜索 r 参数 r8760[0]，该参数在接收缓冲器中有 32 位过程数据的映射。映射示例见下图：
 - PZD 2+3 中的转速设定值（68FF hex）

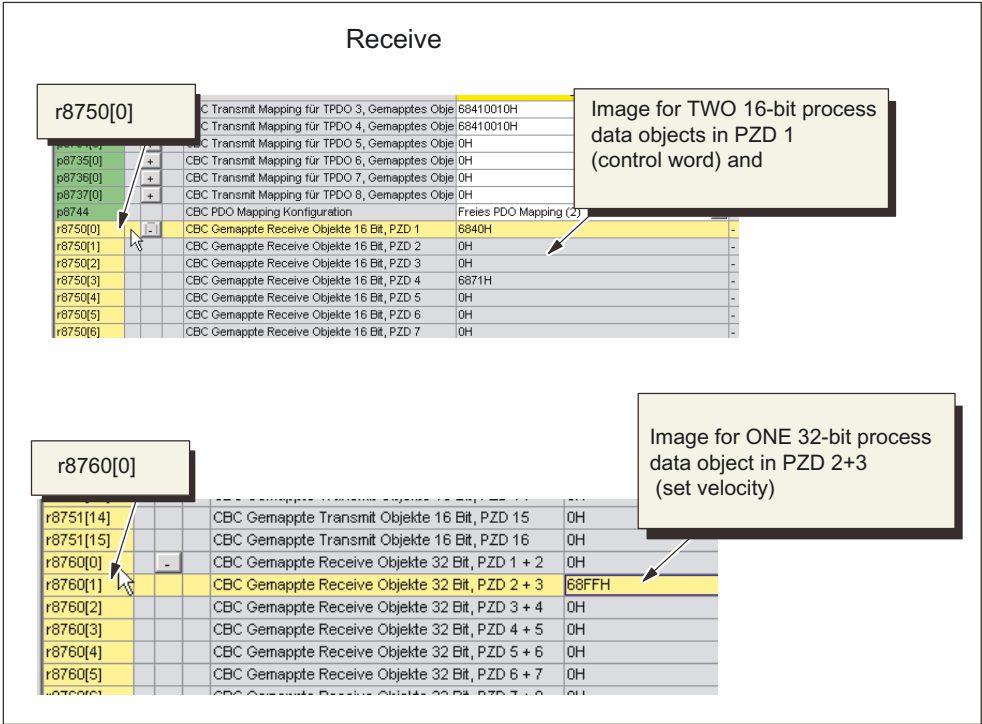


图 3-20 接收缓冲器中过程数据对象的映射

5. 搜索参数 r8751[0]，该参数在发送缓冲器中有 16 位过程数据的映射。映射示例见下图：
 - PZD 1 中的 CBC 状态字 (6841 hex)
 - PZD 4 中的转矩实际值 (6874 hex)
6. 搜索参数 r8761[0]，该参数在发送缓冲器中有 32 位过程数据的映射。映射示例见下图：
 - PZD 2+3 中的转速实际值 (686C hex)
 - PZD 5+6 中的位置实际值 (6863 hex)

Transmit			
r8750[0]	CBC Gemappte Receive Objekte 16 Bit, PZD 10	0H	
r8750[1]	CBC Gemappte Receive Objekte 16 Bit, PZD 11	0H	
r8750[2]	CBC Gemappte Receive Objekte 16 Bit, PZD 12	0H	
r8750[12]	CBC Gemappte Receive Objekte 16 Bit, PZD 13	0H	
r8750[13]	CBC Gemappte Receive Objekte 16 Bit, PZD 14	0H	
r8750[14]	CBC Gemappte Receive Objekte 16 Bit, PZD 15	0H	
r8750[15]	CBC Gemappte Receive Objekte 16 Bit, PZD 16	0H	
r8751[0]	CBC Gemappte Transmit Objekte 16 Bit, PZD 1	6841H	
r8751[1]	CBC Gemappte Transmit Objekte 16 Bit, PZD 2	0H	
r8751[2]	CBC Gemappte Transmit Objekte 16 Bit, PZD 3	0H	
r8751[3]	CBC Gemappte Transmit Objekte 16 Bit, PZD 4	6874H	
r8751[4]	CBC Gemappte Transmit Objekte 16 Bit, PZD 5	0H	
r8751[5]	CBC Gemappte Transmit Objekte 16 Bit, PZD 6	0H	

r8760[8]	CBC Gemappte Receive Objekte 32 Bit, PZD 9 + 10	0H	
r8760[9]	CBC Gemappte Receive Objekte 32 Bit, PZD 10 + 11	0H	
r8760[10]	CBC Gemappte Receive Objekte 32 Bit, PZD 11 + 12	0H	
r8760[11]	CBC Gemappte Receive Objekte 32 Bit, PZD 12 + 13	0H	
r8760[12]	CBC Gemappte Receive Objekte 32 Bit, PZD 13 + 14	0H	
r8760[13]	CBC Gemappte Receive Objekte 32 Bit, PZD 14 + 15	0H	
r8760[14]	CBC Gemappte Receive Objekte 32 Bit, PZD 15 + 16	0H	
r8761[0]	CBC Gemappte Transmit Objekte 32 Bit, PZD 1 + 2	0H	
r8761[1]	CBC Gemappte Transmit Objekte 32 Bit, PZD 2 + 3	686CH	
r8761[2]	CBC Gemappte Transmit Objekte 32 Bit, PZD 3 + 4	0H	
r8761[3]	CBC Gemappte Transmit Objekte 32 Bit, PZD 4 + 5	0H	
r8761[4]	CBC Gemappte Transmit Objekte 32 Bit, PZD 5 + 6	6863H	
r8761[5]	CBC Gemappte Transmit Objekte 32 Bit, PZD 6 + 7	0H	

图 3-21 发送缓冲器中过程数据对象的映射

7. 在项目导航器中选中驱动；例如：“Drive_1”。
8. 通过右键菜单“Expert list”调用驱动对象 1 的专家列表。

同样，此处从参数 r8750[0] 起，有 16 位过程数据对象用于接收；从参数 r8760[0] 起，有 32 位过程数据对象用于接收。

从参数 r8751[0] 起，有 16 位过程数据对象用于发送；从参数 r8761[0] 起，有 32 位过程数据对象用于发送。

9. 读取了两个驱动对象的映射后，执行下一步操作互联接收/发送缓冲器中的过程数据对象。

3.6.2 互联传输报文的过程数据

前提条件

在按照上一章介绍的内容读取了映射后，还须注意以下几点：

下列对象之间可以互联：

- SINAMICS 过程数据对象的源/目标参数
- 接收/发送缓冲器的接收字/发送字

下面的章节中的接收/发送缓冲器功能图节选显示了：

- 接收/发送缓冲器中的过程数据对象与接收/发送字如何分配。
- 哪些相关的接收/发送字的目标/源参数必须互联（以颜色加亮）。

说明

可以从“驱动子协议 DSP402 中的对象”表中的“SINAMICS 参数”一栏获取与接收/发送字互联的 SINAMICS 源/目标参数。

3.6 互联接收/发送缓冲器中的过程数据

3.6.2.1 互联接收缓冲器

在接收缓冲器中需要互联以下对象用于传输报文：

- 控制字（PZD 1）
- 转速设定值（PZD 2+3）
- 转矩设定值（PZD 4）

操作步骤

例如需要互联 PZD 接收字 2+3（32 位）中的转速设定值过程数据对象，按以下步骤执行。互联以下目标参数和源参数：

- 转速设定值的 SINAMICS 目标参数（p1155[0] > 32 位，参见表格“驱动子协议 DSP402 中的 CANopen 对象 (页 159)”）

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
...
60FF		目标速度 转速设定值	无斜坡函数 发生器 -> p1155[0] 使用斜坡函数 发生器 - > p1070	SDO/P DO	Integer 32	-	rw

- 接收缓冲器中源参数 r8860[1] > 32 位（参见下面的转速设定值源参数图，相应的已颜色加亮）。

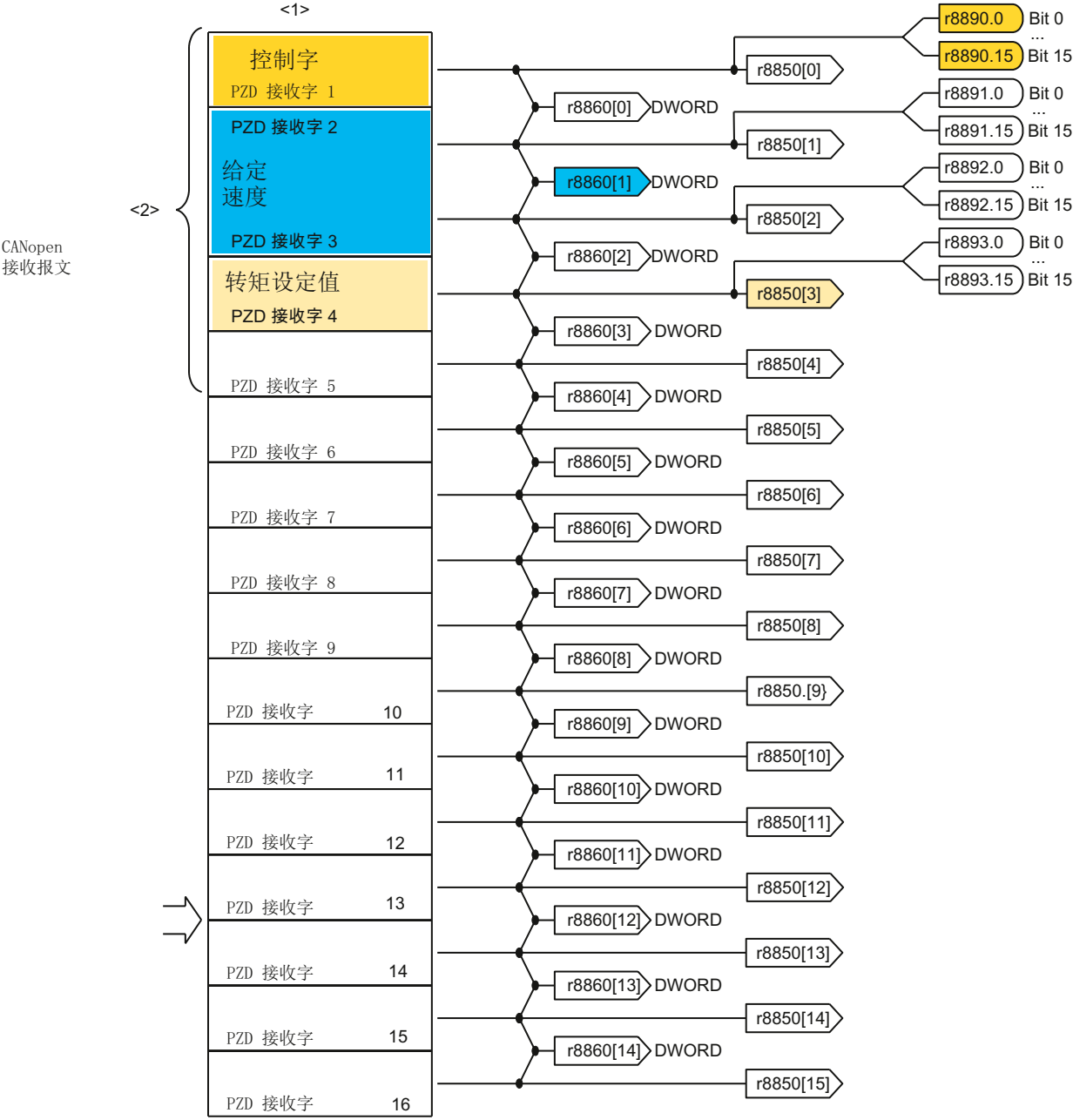


图 3-22 接收缓冲器功能图节选

说明
现在可互联下面列出的过程数据对象。

3.6 互联接收/发送缓冲器中的过程数据

控制字（PZD 1，16 位）

首次调试时，控制字会作为 SINAMICS 目标参数自动与源参数 r8890 互联。

示例：

- 1. 在项目导航器中选中驱动；例如：“Drive_1”。
- 2. 通过右键菜单“Expert list”调用驱动的专家列表。
- 3. 搜索参数 p8790。
- 4. 选中选项“Interconnection (1)”。

退出此区域后，互联建立。

转速设定值（PZD 2+3，32 位）

下表列出了必须与相应的源互联的转速设定值的参数。

表格 3- 10 互联转速设定值

目标 (汇点)	源点 PZD 2+3	含义
p1155[0]	r8860[1]	转速设定值 1

对于转速设定值（数据类型 32 位），需要将参数 p1155[0] 与参数 r8860[1] 互联：

- 1. 在项目导航器中双击“Drive_1”>“Open-loop/closed-loop control”>“Setpoint addition”。
- 2. 点击“Speed setpoint 1”并选择右键菜单“Further interconnections...”。
- 3. 将参数 p1155[0] 与 r8860[1] = PZD 2+3 互联。

转矩设定值（PZD 4，16 位）

将转矩设定值的目标参数 p1513[0] 与源参数 r8850[3] 互联：

表格 3- 11 互联转矩设定值

目标 (汇点)	源点 PZD 4	含义
p1513[0]	r8850[3]	附加转矩

1. 在项目导航器中分别双击“Drive_1” > “Open-loop/closed-loop control” > “torque setpoints”。
2. 选择标签“Supplementary torque”。
3. 点击“Supplementary torque 2” 并选择右键菜单“Further interconnections...”。
4. 将参数 p1513[0] 与 r8850[3] = PZD 4 互联。

为了激活转矩设定值，必须在标签“torque setpoints” 中再进行一次互联。

5. 选择标签“torque setpoints”。
6. 点击“Speed/torque control” 并选择右键菜单“Further interconnections...”。
7. 将参数 p1501[0] 与控制字 (r8890) 的一个自由位（例如：位 14）互联。

3.6.2.2 互联发送缓冲器

在发送缓冲器中需要互联以下对象用于传输报文：

- CBC 状态字 (PZD 1)
- 转速实际值（PZD 2+3）
- 转矩实际值（PZD 4）

3.6 互联接收/发送缓冲器中的过程数据

操作步骤

例如需要互联 PZD 发送字 1（16 位）中的 CBC 状态字过程数据对象，按以下步骤执行。 互联以下源参数和目标参数：

- CBC 状态字的 SINAMICS 源参数（r8784 > 16 位，参见表格“驱动子协议 DSP402 中的 CANopen 对象 (页 159)”）

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
6041		状态字	r8784	PDO/S DO	Unsigne d16	-	ro

- 发送缓冲器中目标参数 p8851[0] > 16 位（参见下面的 CBC 状态字目标参数图，相应的已颜色加亮）。

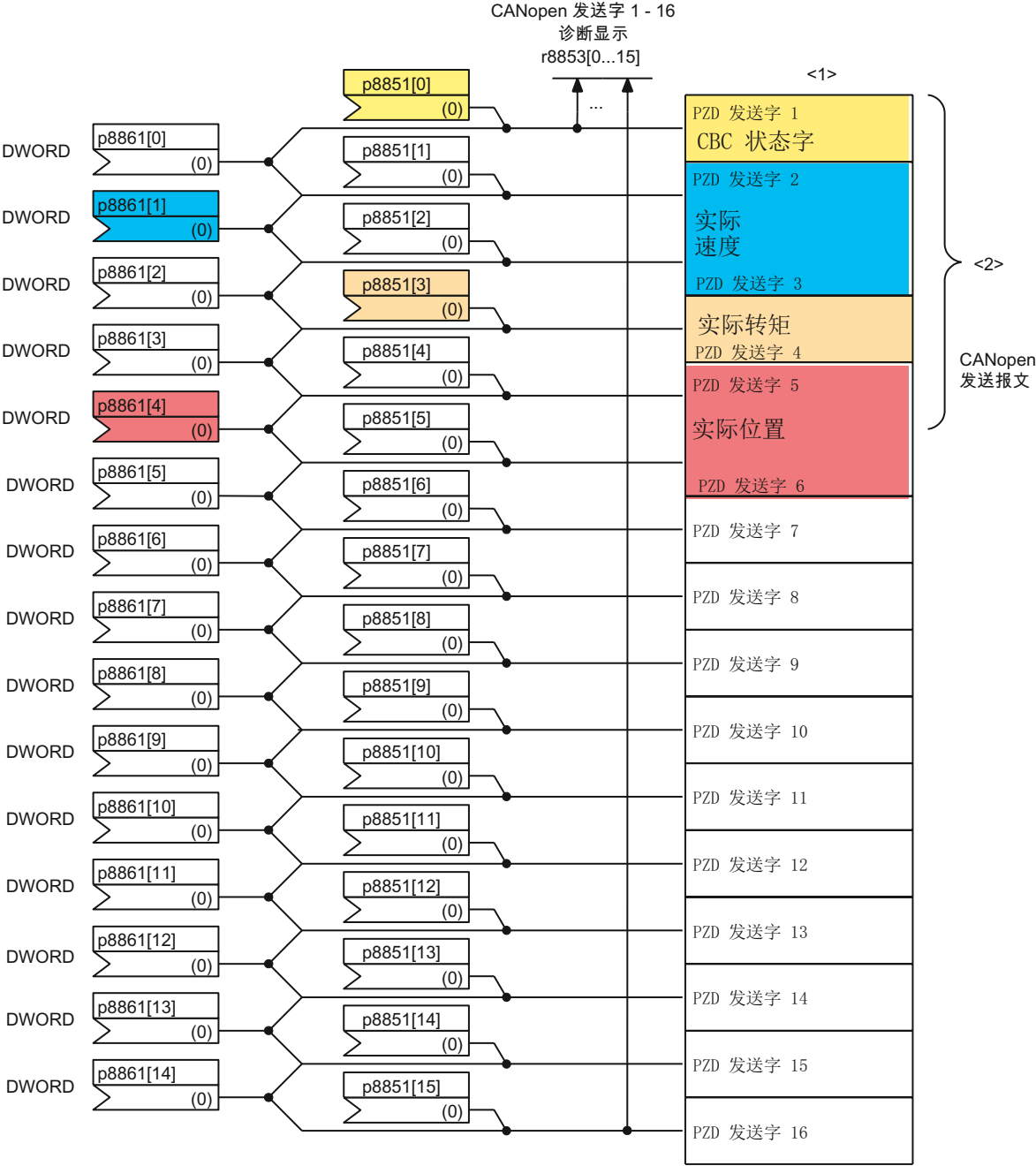


图 3-23 发送缓冲器功能图节选

说明

现在可互联下面列出的过程数据对象。

CBC 状态字（PZD 1，16 位）

将 CBC 状态字的目标参数 p8851[0] 与源参数 r8784 互联：

表格 3- 12 互联 CBC 状态字

目标（汇点） PZD 1	源点	含义
p8851[0]	r8784	CBC 状态字

- 1. 在项目导航器中选中驱动；例如：“Drive_1”。
- 2. 通过右键菜单“Expert list”调用驱动的专家列表。
- 3. 搜索参数 p8851[0]。
- 4. 将参数 p8851[0] = PZD 1 与相应的参数 r8784 互联。

转速实际值（PZD 2+3，32 位）

将转速实际值的目标参数 p8861[1] 与源参数 r0063 互联：

表格 3- 13 互联转速实际值

目标（汇点） PZD 2+3	源点	含义
p8861[1]	r0063	转速实际值

- 1. 在项目导航器中选中驱动；例如：“Drive_1”。
- 2. 通过右键菜单“Expert list”调用驱动的专家列表。
- 3. 搜索参数 p8861[0]。
- 4. 将参数 p8861[1] = PZD 2 + 3 与相应的参数 r0063 互联。

转矩实际值（PZD 4，16 位）

将转矩实际值的目标参数 p8851[3] 与源参数 r0080 互联：

表格 3- 14 互联转矩实际值

目标（汇点） PZD 4	源点	含义
p8851[3]	r0080	转矩实际值

1. 在项目导航器中选中驱动；例如：“Drive_1”。
2. 通过右键菜单“Expert list”调用驱动的专家列表。
3. 搜索参数 p8851[3]。
4. 将参数 p8851[3] = PZD 4 与相应的参数 r0080 互联。

位置实际值（PZD 5+6，32 位）

将位置实际值的目标参数 p8851[3] 与源参数 r0482 互联：

表格 3- 15 互联位置实际值

目标（汇点） PZD 5+6	源点	含义
p8861[4]	r0482	位置实际值

1. 在项目导航器中选中驱动；例如：“Drive_1”。
2. 通过右键菜单“Expert list”调用驱动的专家列表。
3. 搜索参数 p8861[4]。
4. 将参数 p8861[4] = PZD 5+6 与相应编码器的相应参数 r0482[0...2] 互联。

3.6.2.3

互联另一个驱动对象

重复“互联接收缓冲器 (页 108)”和“互联发送缓冲器 (页 111)”中描述的操作步骤，连接下一个驱动对象（单电机模块 2）。

执行这些步骤时，在项目导航器中选择“Drive_2”等。


3.7 在线模式下将项目从驱动设备加载到 PC/PG 并保存

前提条件

处于调试工具 STARTER 在线模式，并且已执行了首次调试步骤。

步骤

执行以下步骤，在线将 STARTER 项目中配置的数据加载到 PG/PC 并保存：

- 1. 在项目导航器中选中驱动“Drive_Unit_Adr126”。
- 2. 点击图标 “Load CPU/drive unit to PG” 。

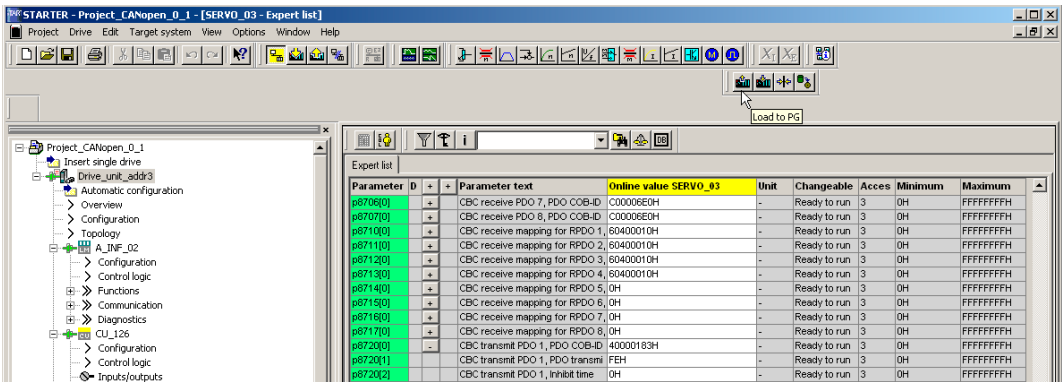



图 3-24 加载到 PG

- 3. 在系统询问 “Start download?”时点击 “Yes”。
- 4. 在 “the data was successfully loaded”对话框中点击 “OK”。
- 5. 点击图标 “Disconnect from target system” 。

6. 如果中途显示询问，依次执行以下操作：

- 点击 “Changes in the drive unit...”。
- 点击 “Save data”，用于 SERVO_3
- 显示信息 “the data was successfully copied from RAM to ROM”时，点击 “OK”。
- 系统询问 “Are you sure that you want the data to be loaded to the PG?”时，点击 “Yes”。
- 显示信息 “the data was successfully loaded to the PG.”时点击 “OK”。
- 点击 “Save data”，用于 SERVO_4。
- 显示信息 “the data was successfully copied from RAM to ROM”时，点击 “OK”。
- 系统询问 “Are you sure that you want the data to be loaded to the PG?”时，点击 “Yes”。
- 显示信息 “the data was successfully loaded to the PG.”时点击 “OK”。

调试工具 STARTER 处于离线模式。

7. 调用菜单 “Project > Save as...”。

说明

CANopen 接口首次调试结束。

3.8 SDO 访问 CANopen 过程数据对象

3.8.1 标准的 CANopen PZD 对象

访问标准的对象

选择合适的访问方式和需要进行的 BICO 互联属于用户的责任。

PDO 访问

对经过映射的标准 CANopen PZD 对象的 PDO 访问是由相应 PZD 接口参数的 BICO 互联实现的。标准的 CANopen PZD 对象指的是制造商专用号段 0x6000 - 0x67FF 内的对象。

SDO 访问

SDO 访问取决于对象是否映射为 PDO：

- 对象已映射为 PDO
可对相应的 PZD 接口参数持续进行 SDO 访问
- 设定值对象未映射为 PDO
可对相应的 PZD 接口参数持续进行 SDO 访问
- 实际值对象未映射为 PDO
可直接对实际值参数进行 SDO 访问。

下图反映是对标准的 CANopen PZD 对象的访问方式。

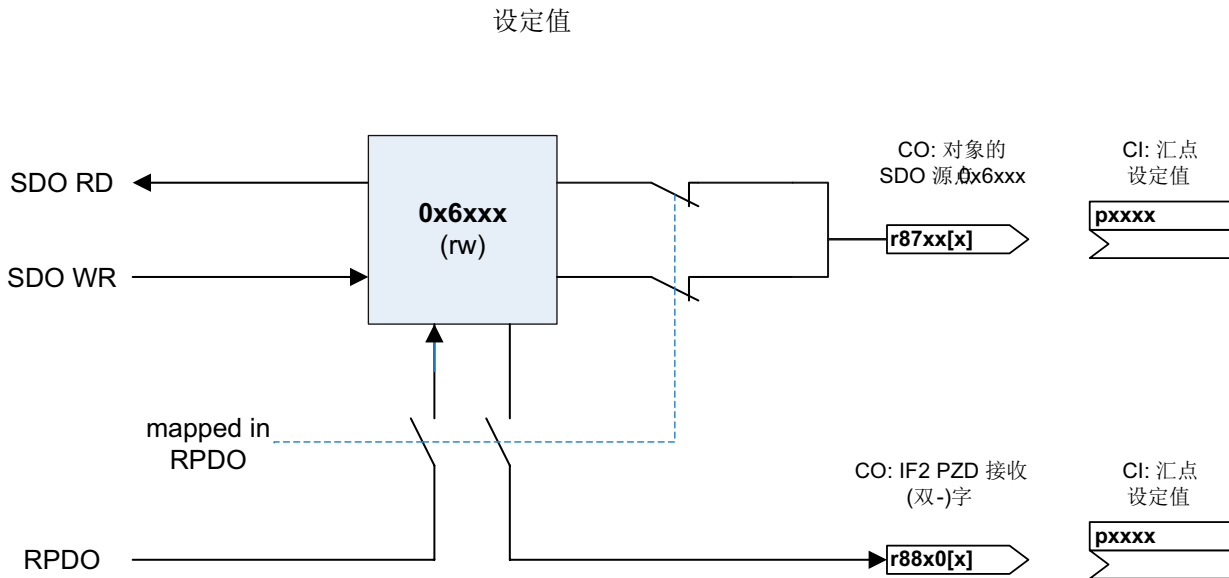


图 3-25 访问标准的 CANopen PZD 设定值对象

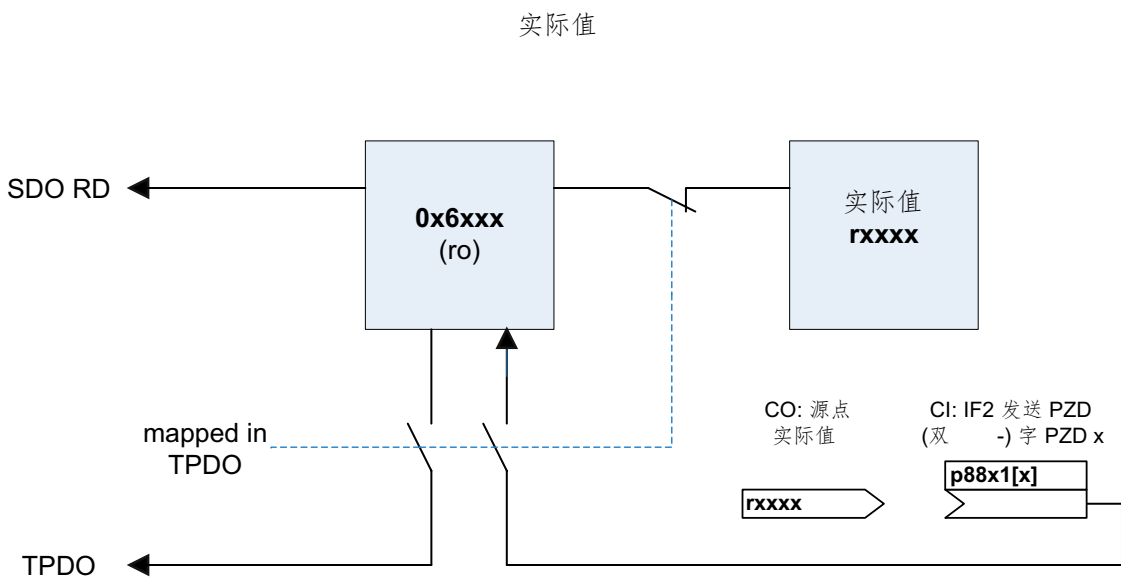


图 3-26 访问标准的 CANopen PZD 实际值对象

3.8.2 自由的 CANopen PZD 对象

访问自由的 CANopen PZD 对象

选择合适的访问方式和需要进行的 BICO 互联属于用户的责任。

PDO 访问

对经过映射的自由 CANopen PZD 对象的 PDO 访问是由相应 PZD 接口参数的 BICO 互联实现的。自由的 CANopen PZD 对象指的是制造商专用号段 0x5800 - 0x58xx 内的对象。

SDO 访问

SDO 访问取决于对象是否映射为 PDO:

- 对象已映射为 PDO

可对相应的 PZD 接口参数持续进行 SDO 访问

- 设定值对象未映射为 PDO

通过互联对应的设定值汇点和 CO 参数实现 SDO 访问。所以每个 PZD 设定值对象都需要一个 CO 参数。如果可以，可在 CANopen 运行模式中对其进行编组。

- 实际值对象未映射为 PDO

通过互联对应的实际值源点和 CI 参数实现 SDO 访问。所以每个自由的 PZD 实际值对象都需要一个 CI 参数。

图反映是对自由的 PZD 对象的访问方式。

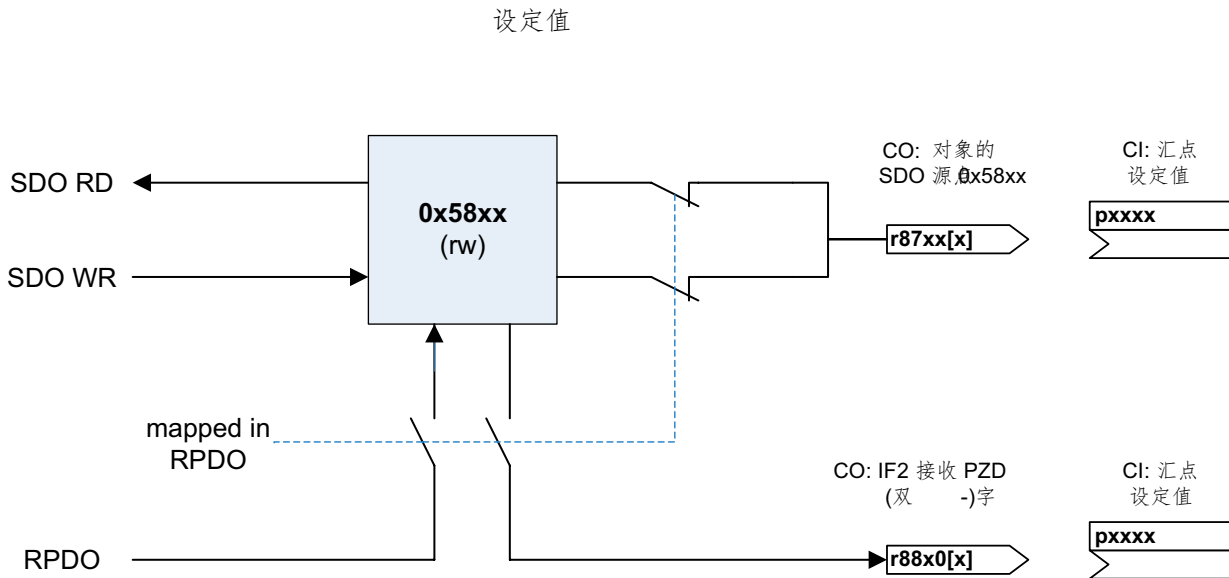


图 3-27 访问自由的 PZD 设定值对象

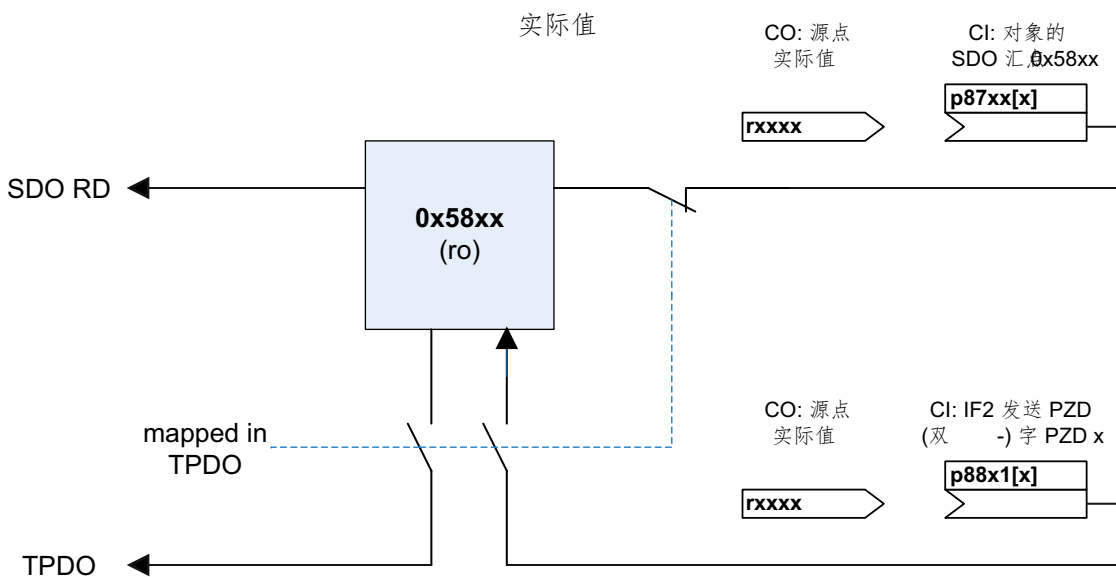


图 3-28 访问自由的 PZD 实际值对象

3.8 SDO 访问 CANopen 过程数据对象

计算/换算设定值/实际值

4.1 转速设定值设定和转速实际值计算

对 CANopen 对象“目标速度 0x60FF”或“速度实际值 0x606C”的访问需要借助 PDO 或 SDO 传输。

CANopen 对象的缺省单位为“增量/秒”。

说明

驱动对象的 CANopen 对象索引通过以下表达式计算：

- 60FF hex + 800 hex * x (x: 驱动编号 0 ... 7)
 - 606C hex + 800 hex * x (x: 驱动编号 0 ... 7)
-

设定转速设定值和计算转速实际值时必须考虑以下参数：

- p8798 = CBC 转速换算系数

参数对应 CANopen 对象 6094 hex。

通过系数将所需的速度单位转换为内部速度单位 (U/s)。

CANopen 速度单位的出厂设置：

- 使用编码器 -> 增量/秒
- 无编码器 -> 转/分钟

内部速度通过以下表达式计算：

转速设定值

$$n_设定_内部[U/s] = \frac{n_设定_总线}{p0408 * 2^{p0418}} * \frac{p8798[0]}{p8798[1]}$$

- p0408 = 线数
- p0418 = 细分分辨率

4.1 转速设定值设定和转速实际值计算

带/不带编码器的转速设定值

- 使用编码器

向总线发送的值通过以下表达式计算：

$$n_设定_总线 = n_设定[rpm] * \frac{1}{60s} * p0408 * 2^{p0418} * \frac{p8798[1]}{p8798[0]}$$

若设定速度值为 3000 rev/min，则在编码器线数为 2048，细分分辨率为 11 时，必须将以下值发送至连接到 SINAMICS 的总线。

$$C800000 \text{ hex} = \frac{3000 [rpm]}{60s} * 2048 * 2^{11} * \frac{1}{1}$$

- 无编码器

$$n_设定_总线 = n_设定[rpm] * \frac{p8798[1]}{p8798[0]}$$

过程数据接口

参数 r8860[x] 中显示的输入转速设定值是通过以下公式计算出的：

$$r8860[x] = \frac{n_总线设定值}{p0408 * 2^{p0418}} * 60s * \frac{4000\ 0000 \text{ hex}}{p2000}$$

$$r8860[x] = n_设定[rpm] * \frac{4000\ 0000 \text{ hex}}{p2000}$$

必须设置 p2000 = p0311 才能在 r8860[x] 中查看到通过总线设定的过程值（相对于 4000 000 hex）。

转速实际值

带/不带编码器的转速实际值

- 使用编码器

$$n_{\text{实际}}[\text{rpm}] = n_{\text{实际_总线}} * 60\text{s} * \frac{1}{p0408 * 2^{p0418}} * \frac{p8798[0]}{p8798[1]}$$

- 无编码器

$$n_{\text{实际}}[\text{rpm}] = n_{\text{实际_总线}} * \frac{p8798[0]}{p8798[1]}$$

过程数据接口

参数 r8863[x] 中显示的输入转速设定值是通过以下公式计算出的：

$$r8863[x] = \frac{n_{\text{实际_总线}}}{p0408 * 2^{p0418}} * 60\text{s} * \frac{4000\ 0000\ \text{hex}}{p2000}$$

$$r8863[x] = n_{\text{实际}}[\text{rpm}] * \frac{4000\ 0000\ \text{hex}}{p2000}$$

必须设置 $p2000 = p0311$ 才能在 r8863[x] 中查看到通过总线设定的过程值（相对于 4000 000 hex）。

说明

但在一些运行模式中还是需要对编码器进行调试，例如：无编码器转速控制 ($p1300 = 20$) 或无编码器转矩控制 ($p1300 = 22$)。在这些运行模式中，转速实际值通过公式“使用编码器”计算得出。

4.2 转矩设定值设定和转矩实际值计算

4.2 转矩设定值设定和转矩实际值计算

对 CANopen 对象“目标转矩 0x6071”或“转矩实际值 0x6077”的访问需要借助 PDO 或 SDO 传输。

CANopen 对象的缺省单位为千分比（1/1000）。

说明

驱动对象的 CANopen 对象索引通过以下表达式计算：

6071 hex + 800 hex * x (x: 驱动编号 0 ... 7)

6077 hex + 800 hex * x (x: 驱动编号 0 ... 7)

设定转矩设定值和计算转矩实际值时必须考虑以下参数：

- r0333 = 电机额定转矩

转矩设定值

向总线发送的值通过以下表达式计算：

$$m_总线设定值 \text{ [千分比]} = \frac{m_设定值 \text{ [Nm]}}{r0333 \text{ [Nm]}} * 1000$$

过程数据接口

参数 r8850[x] 中显示的输入转矩设定值是通过以下公式计算出的（总线设定值[千分比]乘以相对于 p2003 的 r0333 的值）。

示例：

$$r8850[x] = \frac{m_总线设定值 \text{ [千分比]} * r0333 \text{ [Nm]}}{1000} * \frac{4000 \text{ hex}}{p2003 \text{ [Nm]}}$$

必须设置 p2003 = r0333 才能在 r8850[x] 中查看到通过总线设定的百分值 (4000 hex)。

转矩实际值

转矩实际值通过以下表达式计算：

$$m_{\text{实际}}[\text{Nm}] = \frac{m_{\text{实际_总线}}[\text{千分比}]}{1000} * r0333 [\text{Nm}]$$

过程数据接口

参数 r8853[x] 中显示的输入转矩实际值是通过以下公式计算出的：

$$r8853[x] = \frac{m_{\text{实际_总线}}[\text{千分比}] * r0333 [\text{Nm}]}{1000} * \frac{4000 \text{ hex}}{p2003 [\text{Nm}]}$$

$$r8853[x] = \frac{m_{\text{ist}} [\text{Nm}] * 1000}{r0333 [\text{Nm}]} * \frac{4000 \text{ hex}}{p2003 [\text{Nm}]}$$

必须设置 $p2003 = r0333$ 才能在 r8853[x] 中查看到通过总线设定的过程值(4000 hex)。

4.2 转矩设定值设定和转矩实际值计算

诊断

CANopen 通过以下设备支持设备故障的标准化检测、描述和信号化。

- 每个驱动设备的独立报警对象（“紧急对象”）
- 设备内部故障列表（“Predefined Error-Field”，预定义故障区）。
- 故障寄存器（“Error Register”）。

说明

参见标准：

- CiA 301 (Application Layer and Communication Profile)
 - CiA 402 (Device Profile for Drives and Motion Control)
-

5.1 报警对象（紧急对象）

5.1 报警对象（紧急对象）

每个驱动设备都通过高优先级的 8 字节报警对象（紧急对象，故障信息）发送故障状态信息。

各个参数获取方式：

- 对象字典索引 1014 hex (COB-ID EMCY) 和 1015 hex (Inhibit Time EMCY) 中
- SINAMICS 控制单元的参数 p8603 中。

发生故障后，会向对象“1014 hex”中设置的识别符发送故障报文（紧急报文）。

在 CANopen 中每个故障会被分配一个故障代码，故障代码还可以进一步划分为电流故障、电压故障等。

紧急报文

CANopen 驱动设备会在发生故障时自动异步发送紧急报文。紧急报文的结构如下表所示。

表格 5- 1 紧急报文结构

字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7
CANopen 故障代码		CANopen 故障寄存器	SINAMICS 故障号		驱动对象号	备用	备用

字节 0 和 1 为 CANopen 故障代码（参见下面的章节）。

字节 2 中为 CANopen 故障寄存器的编码（参见下面的章节）。

字节 5 中为故障来源的驱动对象号。

不会引起断路的警告不会触发紧急报文，而是会先通过状态字中的警告位或故障位显示。故障会触发紧急报文并导致驱动断开。主站可在自由总线时间读取警告或故障。可通过设置对象 1014 hex 中的位 31 抑制故障报文。

5.2 设备内部故障列表（预定义故障区）。

设备内部故障列表（预定义故障区）可通过以下对象读取：

- 对象字典索引 1003 hex,
- SINAMICS 控制单元参数 p8611

故障列表

列表中包含了驱动设备中经过诊断的、未应答的故障和 CANopen 报警号段 8700-A8799 中存在的警告。故障通过故障代码（Errorcode）和针对特定设备的附加信息以故障发生的时间顺序描述。应答故障或者消除警告后，它从驱动设备内部故障列表中删除。

在对象字典索引 1003 hex 中写入子索引 0，或者在参数 p8611 中写入 0 后，都可以应答所选 CU 的所有驱动对象。

下表中描述了通过 SINAMICS 分析的 CANopen 故障代码（紧急报文中，字节 0/1）。

表格 5- 2 CANopen 故障代码

CANopen 故障代码	含义	通过 SINAMICS 触发
0000 hex	无故障存在	成功应答所有的故障或所有报警都已排除
1000 hex	CAN 故障 1	所有其它 SINAMICS 故障
1001 hex	CAN 故障 2	报警号段 F08700 到 F08799 中的所有其它 CANopen 报警
8110 hex	CAN 溢出，信息丢失	CBC: 报文丢失 (A(N)08751) [报警]
8120 hex	CAN 消极故障 (Error Passive)	CBC: 超出消极故障的故障数 (A08752)[报警]
8130 hex	CAN 寿命保护故障 (CAN Life Guard Error)	CBC: 通讯故障，报警值 2 F08700(A)[故障/报警]

说明

其它 SINAMICS 警告不会发送紧急报文。

5.3 故障寄存器 (“Error Register”)

1 字节故障寄存器 (“Error Register”) 可通过以下对象读取：

- 对象字典索引 1001 hex
- SINAMICS 控制单元参数 r8601

故障寄存器

寄存器可以显示存在的驱动设备故障及其类型。

下表中描述了通过 SINAMICS 分析的 CANopen 故障寄存器（紧急报文中，字节 1 和 2）。

表格 5-3 故障寄存器

故障寄存器	含义	通过 SINAMICS 触发
位 0	generic error	每次 CAN 检测到报警就置位。
位 4	communication error	在发出 CAN 通讯报警（即 08700-08799 号段的报警）时置位。
位 7	manufacturer error	在出现所有 SINAMICS 故障（除了 CAN 通讯报警号段故障）时置位。

运行模式

CANopen 支持与速度和转矩相关的运行模式：

- 速度模式（与速度相关）
采用斜坡的简单速度控制以及相关对象（例如：变频器或换流器）
- 协议速度模式（与速度相关）
速度控制、转速控制以及相关对象。
- 协议转矩模式（与转矩相关）
转矩控制以及相关对象

CANopen 对象和运行模式一览

表格 6-1 CANopen 对象和 CANopen 运行模式

运行模式	通过 CANopen 运 行模式的对象 优先访问	0x6060 运行 模式	0x6061 - 显 示运行模式	0x6502 - 支 持的驱动模式	p1300 - 开环/ 闭环运行模式
速度模式	速度模式	2	2	位 1	0
M* 1		-1	-1	位 16	1
M* 2		-2	-2	位 17	2
M* 3		-3	-3	位 18	3
M* 4		-4	-4	位 19	4
M* 5		-5	-5	位 20	5
M* 6		-6	-6	位 21	6
M* 7		-7	-7	位 22	7
M* 8		-15	-15	位 23	15
M* 9	M*	-18	-18	位 22	18
M* 10	速度模式	-19	-19	位 25	19
M* 11	协议速度模式	-20	-20	位 26	20

运行模式	通过 CANopen 运 行模式的对象 优先访问	0x6060 运行 模式	0x6061 - 显 示运行模式	0x6502 - 支 持的驱动模式	p1300 - 开环/ 闭环运行模式
协议速度 模式		3	3	位 2	21
M* 12	协议转矩模式	-22	-22	位 27	22
协议转矩 模式		4	4	位 3	23
无模式变 化/无指定 模式	-	0	0	-	-
M* = 制造商专用的运行模式					

CANopen 对象 0x6502

CANopen 对象“0x6502 支持的驱动模式”显示了通过对象“0x6060”、SDO 访问或 PDO 访问可以对相应的驱动设备中以及调试状态下的哪些 CANopen 运行模式 (Operation Modes) 进行设置。

CANopen 对象 0x6060

可以通过写入 CANopen 对象“0x6060 Modes of operation”来选择所需的运行模式 (Operation Mode)。

对象始终显示请求的运行模式，即使该运行模式未激活（比如：由于驱动对象不支持该运行模式）。

即使已经在其数据类型内写入一个值，仍一直可以通过 SDO 写入。

如果无法选择一种运行模式（比如：由于不支持该运行模式），那么旧的运行模式继续保持生效。

控制系统必须自己通过读取对象 “0x6061” 来检查是否已选择了所需的运行模式以及该运行模式是否生效。

通过 RPDO 访问

该对象是可映射的，因此可通过 PDO 来传输（不需经过确认）。目前，在 SINAMICS 上，按 CAN 采样周期进行的过程数据访问只对转速控制和转矩控制的切换有用，因为 CANopen 只支持这两种运行模式，而且模式只能在 SINAMICS 运行时切换。

支持的运行模式的所有切换都汇入在后台运行的 SINAMICS 参数通道。

当对象已映射为 RPDO 时，也可对相应的 PZD 接口参数（而不再对 p1300）持续进行 SDO 访问。下图显示了该过程。

其他运行模式的所有切换都汇入在后台运行的 SINAMICS 参数通道。此时进行 PDO 访问没有意义。当对象已映射为 RPDO 时，也可对相应的 PZD 接口参数（而不再对 p1300）持续进行 SDO 访问。下图显示了该过程。

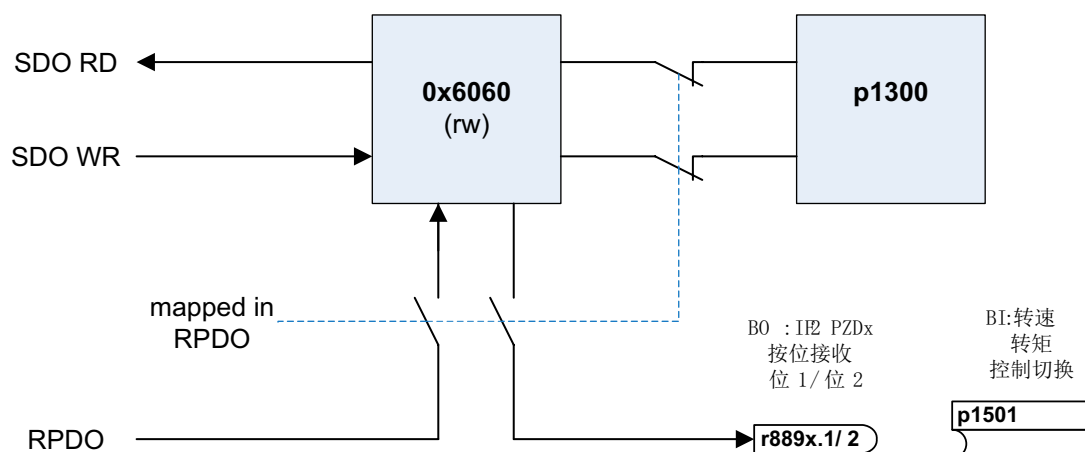


图 6-1 访问运行模式

调试期间选择合适的访问方式属于用户的责任。

通过 SDO 访问

通过 SDO 访问时，会在对象 0x6060 的参数 p1300 中写入一个值。可从 表格 6-1 CANopen 对象和 CANopen 运行模式 (页 133) 中获取相应的值。

CANopen 对象 0x6061

CANopen 对象“0x6061 Modes of operation display”显示的是当前生效的运行模式。

该对象是可映射的，因此可通过 PDO 来传输（不需经过确认）。

当对象已映射为 TPDO 时，也可对相应的 PZD 接口参数持续进行 SDO 访问。在调试时，必须将新的 CO 参数“r8762 CAN 运行模式显示”和 PZD 接口 IF2 相应的汇点连接在一起。

参数 r8762 的值如下得出：

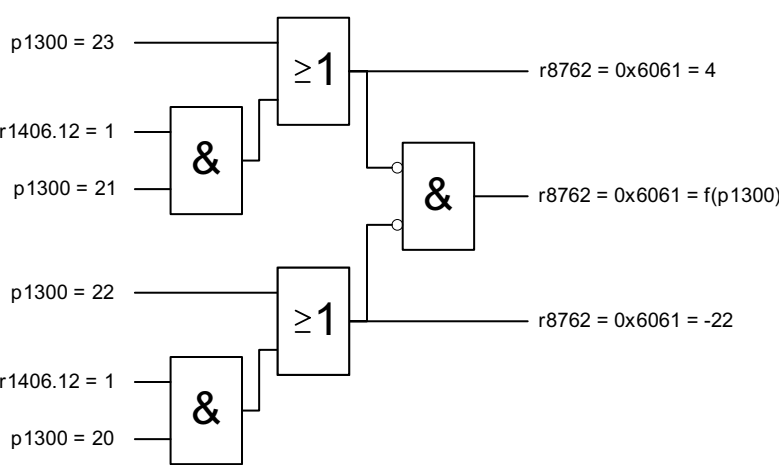


图 6-2 运行模式显示

在将相应的 TPDO 的传输方式配置为异步时，运行模式修改成功后会立即发送该传输方式并将运行模式修改成功的信息报告给控制系统。

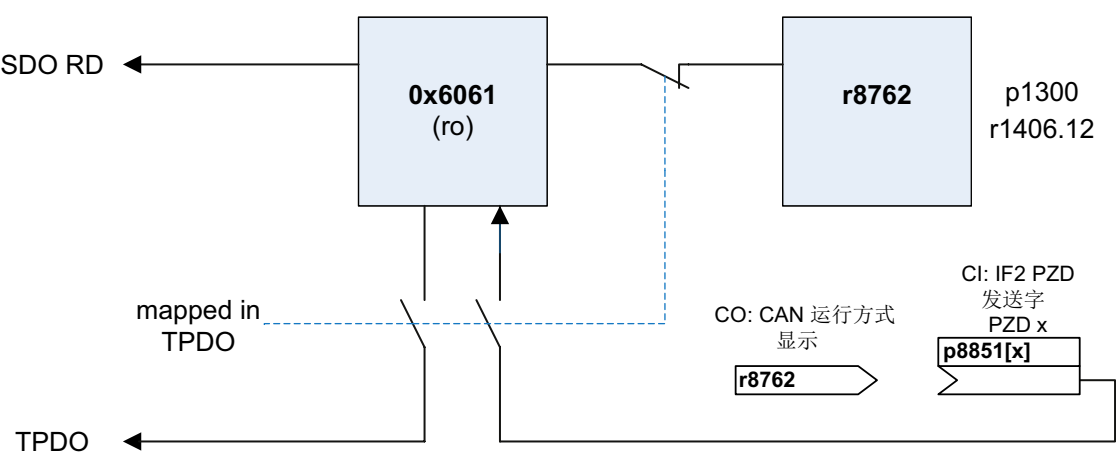


图 6-3 访问运行模式显示

通讯对象

本章节将以表格形式列出 SINAMICS 通过 CANopen 接口通讯时使用的对象（数据值）：

- 与驱动无关的控制单元通讯对象
- 和驱动相关的通讯对象
- 制造商专用对象
- 驱动子协议 DSP402 中的对象

对象字典中的对象

7.1 与驱动无关的控制单元通讯对象

下表中列出了包含了与驱动无关的控制单元对象索引的对象字典。在“SINAMICS 参数”一栏中显示在 SINAMICS 中分配的参数编号。

表格 7-1 与驱动无关的控制单元通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
1000		设备类型	r8600	SDO	Unsigned32	–	ro
1001		故障寄存器	r8601	SDO	Unsigned8	–	ro
1003	0...52 hex	预定义故障区	p8611[0...82]	SDO	Unsigned32	0	ro/rw
	0	故障数量	p8611.0	SDO	Unsigned32	0	rw
	1	模块数量	p8611.1	SDO	Unsigned32	0	ro
	2	模块 1 故障数量	p8611.2	SDO	Unsigned32	0	ro
	3-A	模块 1 的缺省故障区	p8611.3-p8611.10	SDO	Unsigned32	0	ro
	B	模块 2 故障数量	p8611.11	SDO	Unsigned32	0	ro
	C-13	模块 2 的缺省故障区	p8611.12-p8611.19	SDO	Unsigned32	0	ro
	14	模块 3 故障数量	p8611.20	SDO	Unsigned32	0	ro
	15-1C	模块 3 的缺省故障区	p8611.21-p8611.28	SDO	Unsigned32	0	ro
	1D	模块 4 故障数量	p8611.29	SDO	Unsigned32	0	ro
	1E-25	模块 4 的缺省故障区	p8611.30-p8611.37	SDO	Unsigned32	0	ro
	26	模块 5 故障数量	p8611.38	SDO	Unsigned32	0	ro
	27-2E	模块 5 的缺省故障区	p8611.39-p8611.46	SDO	Unsigned32	0	ro
	2F	模块 6 故障数量	p8611.47	SDO	Unsigned32	0	ro
	30-37	模块 6 的缺省故障区	p8611.48-p8611.55	SDO	Unsigned32	0	ro
	38	模块 7 故障数量	p8611.56	SDO	Unsigned32	0	ro

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
	39-40	模块 7 的缺省故障区	p8611.57-p8611.64	SDO	Unsigned32	0	ro
	41	模块 8 故障数量	p8611.65	SDO	Unsigned32	0	ro
	42-49	模块 8 的缺省故障区	p8611.66-p8611.73	SDO	Unsigned32	0	ro
	4A	控制单元的故障数量	p8611.74	SDO	Unsigned32	0	ro
	4B-52	控制单元的缺省故障区	p8611.75-p8611.82	SDO	Unsigned32	0	ro
1005		COB ID SYNC	p8602	SDO	Unsigned32	128	rw
1008		制造商设备名称	–	SDO	–	–	–
100A		制造商软件版本	r0018	SDO	Unsigned32	–	ro
100C		保护时间	p8604.0	SDO	Unsigned16	0	rw
100D		寿命系数	p8604.1	SDO	Unsigned16	0	rw
1010		存储参数	p0977	SDO	Unsigned16	0	rw
	0	支持的最大子索引	–	SDO	–	–	–
	1	保存所有参数	p0977	SDO	Unsigned16	0	rw
	2	保存通讯参数 (0x1000-0x1fff)	p0977	SDO	Unsigned16	0	rw
	3	保存应用相关参数 (0x6000-0x9fff)	p0977	SDO	Unsigned16	0	rw
1011		恢复缺省参数	p0976	SDO	Unsigned16	0	rw
	0	支持的最大子索引		SDO			
	1	恢复所有缺省参数	p0976	SDO	Unsigned16	0	rw
	2	恢复缺省通讯参数 (0x1000-0x1fff)	p0976	SDO	Unsigned16	0	rw
	3	恢复缺省应用参数 (0x6000-0x9fff)	p0976	SDO	Unsigned16	0	rw
1014		COB ID Emergency	p8603	SDO	Unsigned32	0	rw

7.1 与驱动无关的控制单元通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
1017		生产者心跳时间	p8606	SDO	Unsigned16	0	rw
1018		识别对象	r8607[0...3]		Unsigned32	–	ro
	0	条目数量	–	SDO	–	–	–
	1	供应商 ID	r8607.0	SDO	Unsigned32	–	ro
	2	产品代码	r8607.1	SDO	Unsigned32	–	ro
	3	版本号	r8607.2	SDO	Unsigned32	–	ro
	4	序列号	r8607.3	SDO	Unsigned32	0	ro
1027		模块列表	–	–	–	–	–
	0	条目数量	r0102	SDO	Unsigned16	–	ro
	1-8	模块 ID	p0107[0...15]	SDO	Integer16	0	rw
1029		故障特性	–	–	–	–	–
	0	故障级编号	–	SDO	–	–	–
	1	通讯故障	p8609.0	SDO	Unsigned32	1	rw
	2	设备子协议或特定制造商故障	p8609.1	SDO	Unsigned32	1	rw
1200		第 1 服务器 SDO 参数	–	–	–	–	–
	0	条目数量	–	SDO	–	–	–
	1	COB-ID Client -> Server (rx)	r8610.0	SDO	Unsigned32	–	ro
	2	COB-ID Server -> Client (tx)	r8610.1	SDO	Unsigned32	–	ro
1201		驱动对象服务器 SDO 参数	–	–	–	–	–
	0	条目数量	–	SDO	–	–	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw

7.2 和驱动相关的通讯对象

可为每个驱动设置八个发送 PDO/八个接收 PDO，接收 PDO 的总数量不能超过 25。

每个 PDO 包含：

- 通讯参数和
- 映射参数（最大 8 字节/4 字/64 位）。

规则

下表中第一个 PDO 的相关参数会**加粗**显示，表示这些通讯参数和映射参数从属于一个 PDO。

在“预定义连接集”一栏显示“预定义连接集”的预设值。

7.2.1 接收 PDO 中的通讯对象

概述

下表列出了第一个驱动对象的接收 PDO 对象字典，其中包含了各个和驱动相关的通讯对象索引。

表格 7-2 接收 PDO 中和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1400		接收 PDO 1 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro
	1	PDO 使用的 COB-ID	p8700.0	SDO	Unsigned32	200 hex + Node-ID	rw
	2	传输类型	p8700.1	SDO	Unsigned8	FE hex	rw
1401		接收 PDO 2 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro

7.2 和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
	1	PDO 使用的 COB-ID	p8701.0	SDO	Unsigned32	300 hex + Node-ID	rw
	2	传输类型	p8701.1	SDO	Unsigned8	FE hex	rw
1402		接收 PDO 3 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro
	1	PDO 使用的 COB-ID	p8702.0	SDO	Unsigned32	400 hex + Node-ID	rw
	2	传输类型	p8702.1	SDO	Unsigned8	FE hex	rw
1403		接收 PDO 4 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro
	1	PDO 使用的 COB-ID	p8703.0	SDO	Unsigned32	500 hex + Node-ID	rw
	2	传输类型	p8703.1	SDO	Unsigned8	FE hex	rw
1404		接收 PDO 5 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro
	1	PDO 使用的 COB-ID	p8704.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8704.1	SDO	Unsigned8	FE hex	rw
1405		接收 PDO 6 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro
	1	PDO 使用的 COB-ID	p8705.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8705.1	SDO	Unsigned8	FE hex	rw

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1406		接收 PDO 7 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro
	1	PDO 使用的 COB-ID	p8706.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8706.1	SDO	Unsigned8	FE hex	rw
1407		接收 PDO 8 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	2	ro
	1	PDO 使用的 COB-ID	p8707.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8707.1	SDO	Unsigned8	FE hex	rw
1600		接收 PDO 1 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	1	ro
	1	PDO 映射：第一个待映射的应用对象	p8710.0	SDO	Unsigned32	6040 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8710.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8710.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8710.3	SDO	Unsigned32	0	rw

7.2 和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1601		接收 PDO 2 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	2	ro
	1	PDO 映射：第一个待映射的应用对象	p8711.0	SDO	Unsigned32	6040 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8711.1	SDO	Unsigned32	60FF hex	rw
	3	PDO 映射：第三个待映射的应用对象	p8711.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8711.3	SDO	Unsigned32	0	rw
1602		接收 PDO 3 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	2	ro
	1	PDO 映射：第一个待映射的应用对象	p8712.0	SDO	Unsigned32	6040 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8712.1	SDO	Unsigned32	6071 hex	rw
	3	PDO 映射：第三个待映射的应用对象	p8712.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8712.3	SDO	Unsigned32	0	rw

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1603		接收 PDO 4 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	3	ro
	1	PDO 映射：第一个待映射的应用对象	p8713.0	SDO	Unsigned32	6040 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8713.1	SDO	Unsigned32	60FF hex	rw
	3	PDO 映射：第三个待映射的应用对象	p8713.2	SDO	Unsigned32	6071 hex	rw
	4	PDO 映射：第四个待映射的应用对象	p8713.3	SDO	Unsigned32	0	rw
1604		接收 PDO 5 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro
	1	PDO 映射：第一个待映射的应用对象	p8714.0	SDO	Unsigned32	0	rw
	2	PDO 映射：第二个待映射的应用对象	p8714.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8714.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8714.3	SDO	Unsigned32	0	rw

7.2 和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1605		接收 PDO 6 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro
	1	PDO 映射：第一个待映射的应用对象	p8715.0	SDO	Unsigned32	0	rw
	2	PDO 映射：第二个待映射的应用对象	p8715.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8715.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8715.3	SDO	Unsigned32	0	rw
1606		接收 PDO 7 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro
	1	PDO 映射：第一个待映射的应用对象	p8716.0	SDO	Unsigned32	0	rw
	2	PDO 映射：第二个待映射的应用对象	p8716.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8716.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8716.3	SDO	Unsigned32	0	rw

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1607		接收 PDO 8 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro
	1	PDO 映射：第一个待映射的应用对象	p8717.0	SDO	Unsigned32	0	rw
	2	PDO 映射：第二个待映射的应用对象	p8717.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8717.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8717.3	SDO	Unsigned32	0	rw

说明

之后的驱动的索引依次向后推移 40 hex。

7.2.2 发送 PDO 中的通讯对象

概述

下表列出了第一个驱动对象的发送 PDO 对象字典，其中包含了各个和驱动相关的通讯对象索引。

表格 7-3 发送 PDO 中和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1800		发送 PDO 1 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	5	ro
	1	PDO 使用的 COB-ID	p8720.0	SDO	Unsigned32	180 hex + Node-ID	rw
	2	传输类型	p8720.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8720.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8720.3	SDO	Unsigned8	3	rw
	5	事件计时器	p8720.4	SDO	Unsigned16	0	rw
1801		发送 PDO 2 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	5	ro
	1	PDO 使用的 COB-ID	p8721.0	SDO	Unsigned32	280 hex + Node-ID	rw
	2	传输类型	p8721.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8721.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8721.3	SDO	Unsigned8	0	rw
	5	事件计时器	p8721.4	SDO	Unsigned16	0	rw
1802		发送 PDO 3 通讯参数	–	–	–	–	–

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
	0	支持的最大子索引	–	SDO	Unsigned8	5	–
	1	PDO 使用的 COB-ID	p8722.0	SDO	Unsigned32	380 hex + Node-ID	rw
	2	传输类型	p8722.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8722.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8722.3	SDO	Unsigned8	0	rw
	5	事件计时器	p8722.4	SDO	Unsigned16	0	rw
1803		发送 PDO 4 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	5	ro
	1	PDO 使用的 COB-ID	p8723.0	SDO	Unsigned32	480 hex + Node-ID	rw
	2	传输类型	p8723.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8723.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8723.3	SDO	Unsigned8	0	rw
	5	事件计时器	p8723.4	SDO	Unsigned16	0	rw
1804		发送 PDO 5 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	5	ro
	1	PDO 使用的 COB-ID	p8724.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8724.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8724.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8724.3	SDO	Unsigned8	0	rw
	5	事件计时器	p8724.4	SDO	Unsigned16	0	rw

7.2 和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
1805		发送 PDO 6 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	5	ro
	1	PDO 使用的 COB-ID	p8725.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8725.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8725.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8725.3	SDO	Unsigned8	0	rw
	5	事件计时器	p8725.4	SDO	Unsigned16	0	rw
1806		发送 PDO 7 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	5	ro
	1	PDO 使用的 COB-ID	p8726.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8726.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8726.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8726.3	SDO	Unsigned8	0	rw
	5	事件计时器	p8726.4	SDO	Unsigned16	0	rw
1807		发送 PDO 8 通讯参数	–	–	–	–	–
	0	支持的最大子索引	–	SDO	Unsigned8	5	ro
	1	PDO 使用的 COB-ID	p8727.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	传输类型	p8727.1	SDO	Unsigned8	FE hex	rw
	3	禁止时间	p8727.2	SDO	Unsigned16	0	rw
	4	兼容性条目	p8727.3	SDO	Unsigned8	0	rw

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
	5	事件计时器	p8727.4	SDO	Unsigned16	0	rw
1A00		发送 PDO 1 映射参数	—	—	—	—	—
	0	PDO 中映射的应用对象的数量	—	SDO	Unsigned8	1	ro
	1	PDO 映射：第一个待映射的应用对象	p8730.0	SDO	Unsigned32	6041 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8730.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8730.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8730.3	SDO	Unsigned32	0	rw
1A01		发送 PDO 2 映射参数	—	—	—	—	—
	0	PDO 中映射的应用对象的数量	—	SDO	Unsigned8	2	ro
	1	PDO 映射：第一个待映射的应用对象	p8731.0	SDO	Unsigned32	6041 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8731.1	SDO	Unsigned32	606C hex	rw
	3	PDO 映射：第三个待映射的应用对象	p8731.2	SDO	Unsigned32	0	rw

7.2 和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
	4	PDO 映射：第四个待映射的应用对象	p8731.3	SDO	Unsigned32	0	rw
1A02		发送 PDO 3 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	2	ro
	1	PDO 映射：第一个待映射的应用对象	p8732.0	SDO	Unsigned32	6041 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8732.1	SDO	Unsigned32	6077 hex	rw
	3	PDO 映射：第三个待映射的应用对象	p8732.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8732.3	SDO	Unsigned32	0	rw
1A03		发送 PDO 4 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	2	ro
	1	PDO 映射：第一个待映射的应用对象	p8733.0	SDO	Unsigned32	6041 hex	rw
	2	PDO 映射：第二个待映射的应用对象	p8733.1	SDO	Unsigned32	6063 hex	rw

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
	3	PDO 映射：第三个待映射的应用对象	p8733.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8733.3	SDO	Unsigned32	0	rw
1A04		发送 PDO 5 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro
	1	PDO 映射：第一个待映射的应用对象	p8742.0	SDO	Unsigned32	0	rw
	2	PDO 映射：第二个待映射的应用对象	p8742.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8742.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8742.3	SDO	Unsigned32	0	rw
1A05		发送 PDO 6 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro
	1	PDO 映射：第一个待映射的应用对象	p8752.0	SDO	Unsigned32	0	rw

7.2 和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
	2	PDO 映射：第二个待映射的应用对象	p8752.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8752.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8752.3	SDO	Unsigned32	0	rw
1A06		发送 PDO 7 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro
	1	PDO 映射：第一个待映射的应用对象	p8752.0	SDO	Unsigned32	0	rw
	2	PDO 映射：第二个待映射的应用对象	p8752.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8752.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8752.3	SDO	Unsigned32	0	rw
1A07		发送 PDO 8 映射参数	–	–	–	–	–
	0	PDO 中映射的应用对象的数量	–	SDO	Unsigned8	0	ro

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预定义连接集	读写性
	1	PDO 映射：第一个待映射的应用对象	p8752.0	SDO	Unsigned32	0	rw
	2	PDO 映射：第二个待映射的应用对象	p8752.1	SDO	Unsigned32	0	rw
	3	PDO 映射：第三个待映射的应用对象	p8752.2	SDO	Unsigned32	0	rw
	4	PDO 映射：第四个待映射的应用对象	p8752.3	SDO	Unsigned32	0	rw

说明

之后的驱动的索引依次向后推移 40 hex。

7.2.3 其他通讯对象

概述

下表列出了包含与驱动相关的通讯对象索引的对象字典。

表格 7-4 其他通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
1202		驱动对象服务器 SDO 参数	—	—	—	—	—
	0	条目数量	—	SDO	—	—	ro

7.2 和驱动相关的通讯对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1203		驱动对象服务器 SDO 参数	—	—	—	—	—
	0	条目数量	—	SDO	—	—	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1204		驱动对象服务器 SDO 参数	—	—	—	—	—
	0	条目数量	—	SDO	—	—	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1205		驱动对象服务器 SDO 参数	—	—	—	—	—
	0	条目数量	—	SDO	—	—	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1206		驱动对象服务器 SDO 参数	—	—	—	—	—
	0	条目数量	—	SDO	—	—	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1207		驱动对象服务器 SDO 参数	–	–	–	–	–
	0	条目数量	–	SDO	–	–	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1208		驱动对象服务器 SDO 参数	–	–	–	–	–
	0	条目数量	–	SDO	–	–	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1209		驱动对象服务器 SDO 参数	–	–	–	–	–
	0	条目数量	–	SDO	–	–	ro
	1	COB-ID Client -> Server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB-ID Server -> Client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw

7.3 自由对象

对象字典索引 (hex)	描述	PZD 的数据类型	预设值	读写性	SINAMICS 参数
5800 至 580F	16 个可自由连接的接收过程数据对象	Integer16	0	rw	r8745[x]
5810 至 581F	16 个可自由连接的发送过程数据对象	Integer16	0	ro	r8746[x]
5820 至 5827	8 个可自由连接的接收过程数据对象	Integer32	0	rw	r8747[x]
5828 至 582F	备用	–	–	–	–
5830 至 5837	8 个可自由连接的发送过程数据对象	Integer32	0	ro	r8748[x]
5838 至 5879	备用	–	–	–	–

说明

其它驱动对象的自由对象通过可连接对象的对象编号加上 “80 hex” 得出。
例如：驱动对象 2 从 “5880 hex” 开始。

可以通过接收/发送缓冲器的接收字/发送字/双字连接任意过程数据对象。

自由对象过程数据的定标：

- 16 位（字）：4000 hex 对应 100%
- 32 位（字）：4000000 hex 对应 100%

如果过程数据为温度值，则自由对象的定标如下：

- 16 位（字）：4000 hex 对应 100 °C
- 32 位（字）：4000000 hex 对应 100 °C

7.4 驱动子协议 DSP402 中的 CANopen 对象

概述

下表中列出了包含驱动单个对象的索引的对象字典。

在 SINAMICS S120 上，CANopen 支持“协议速度模式”、“协议转矩模式”和“速度模式”。

表格 7-5 驱动子协议 DSP402 中的对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
驱动控制							
6007	–	连接断开选项代码	p8641	SDO	Integer16	3	rw
6040	–	控制字	r8795	PDO/SDO	Unsigned16	–	rw
6041	–	状态字	r8784	PDO/SDO	Unsigned16	–	ro
605D	–	停止选项代码	p8791	SDO	Integer16	–	rw
6060	–	运行模式	p1300	PDO/SDO	Integer8	–	rw
6061	–	运行显示模式	r8762	PDO/SDO	Integer8	–	ro
6502	–	支持的驱动模式	–	SDO	Unsigned32	–	ro
6504	–	驱动制造商	–	SDO	字符串	–	ro
67FF	–	单个设备类型	–	SDO	Unsigned32	–	ro
系数组							
6094	00	速度编码器系数	–	SDO	Unsigned8	2	ro
	01	速度编码器系数的分子	p8798[0]	SDO	Unsigned32	1	rw
	02	速度编码器系数的分母	p8798[1]	SDO	Unsigned32	1	rw
协议速度模式							
6063	–	位置实际值	r0482	SDO/PDO	Integer32	–	ro
6069	–	速度编码器实际值	r0061	SDO/PDO	Integer32	–	ro
606B	–	速度需求值	r1170	SDO/PDO	Integer32	–	ro

7.4 驱动子协议 DSP402 中的 CANopen 对象

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
606C	–	速度实际值 转速实际值	r0063	SDO/PDO	Integer32	–	ro
6083	–	协议加速度	p1082/ p1120	SDO	Unsigned32	–	rw
6084	–	协议减速度	p1082/ p1121	SDO	Unsigned32	–	rw
6085	–	快速停止减速度	p1082/ p1135	SDO	Unsigned32	–	rw
6086	–	运动协议类型	p1115/ p1134	SDO	Integer16	0	rw
60FF	–	目标速度 转速设定值	无斜坡 函数发 生器 -> p1155[0] 使用斜 坡函数 发生器 - > p1070	SDO/PDO	Integer32	–	rw
协议转矩模式							
6071	–	转矩设定值 转矩设定值	r8797	SDO/PDO	Integer16	–	rw
6072	–	最大转矩	p1520	SDO	0	0	0
6074	–	转矩需求值 总转矩设定值	r0079	SDO/PDO	Integer16	–	ro
6077	–	转矩实际值	r0080	SDO/PDO	Integer16	–	ro
速度模式							
6042	–	vl 目标速度	r8792	SDO/PDO	Integer16	–	rw
6043	–	vl 速度设定值	r1170	SDO/PDO	Integer16	–	ro
6044	–	vl 速度实际值	r0063	SDO/PDO	Integer16	–	ro

对象字典索引 (hex)	子索引 (hex)	对象名称	SINAMICS 参数	传输	数据类型	预设值	读写性
6046	0	vl 最小/最大速度	–	SDO	Unsigned8	–	ro
	1	vl 最小速度	p1080	SDO	Unsigned32	–	rw
	2	vl 最大速度	p1082	SDO	Unsigned32	–	rw
6048	0	vl 加速度	–	SDO	Unsigned8	–	ro
	1	Δ 速度	p1082	SDO	Unsigned32	–	rw
	2	Δ 时间	p1120	SDO	Unsigned16	–	rw

说明

对驱动子协议中的后续 SINAMICS 驱动对象进行描述时，需依次向后推移 800 hex。

每个驱动对象的 SDO Server

8.1 映射模型

支持多个映射模型：

- 用一个 SDO Server 映射多驱动对象 SINAMICS 驱动设备
- 用多个 EDS 文件和多个 SDO Server 映射多驱动对象 SINAMICS 驱动设备

用一个 SDO Server 映射多驱动对象 SINAMICS 驱动设备

在该映射模型中，多驱动对象 SINAMICS 驱动设备通过模块映射为一个 SDO Server（缺省 SDO Server）。每个 SINAMICS 驱动设备都有一个模块化的、不含制造商专用的 CANopen 对象的 EDS 文件。

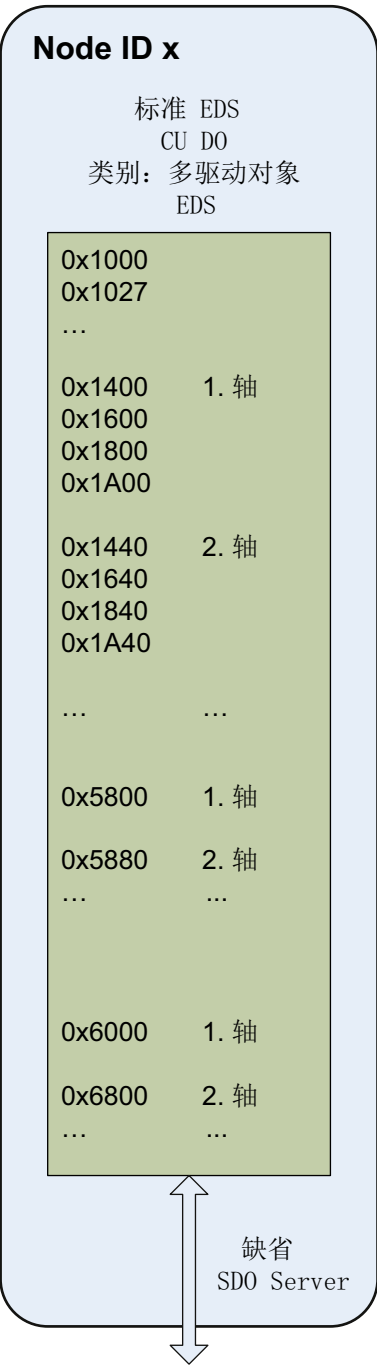


图 8-1 多驱动对象 SINAMICS 驱动设备的建模

用多个 EDS 文件和多个 SDO Server 映射多驱动对象 SINAMICS 驱动设备

在该映射模型中，多驱动对象 SINAMICS 驱动设备映射为 $n+2$ 个 EDS 文件和 $n+2$ 个 SDO Server。

组成部分	注释
$n+2$ 个 EDS 文件	1 个标准 EDS 文件，用于控制单元（驱动对象） 1 个制造商专用的 EDS 文件，用于控制单元（驱动对象） n 个制造商专用的 EDS 文件，用于 CANopen 支持的 n 个轴（ n 为 1 - 8）
$n+2$ 个 SDO 通道	1 个缺省 SDO Server，用于控制单元（驱动对象） 1 个制造商专用的 SDO Server，用于控制单元（驱动对象） n 个制造商专用的 SDO Server，用于 n 个轴（即每个 CANopen 支持的驱动对象， n 为 1 - 8）。

每个附加的 SDO Server 负责一个驱动对象。尤其是负责该驱动对象中制造商专用的对象。缺省 SDO Server 可以访问标准 EDS 文件。

每个 CANopen 驱动对象（包含控制单元）都附加一个单独的制造商专用的 EDS 文件。

8.1 映射模型

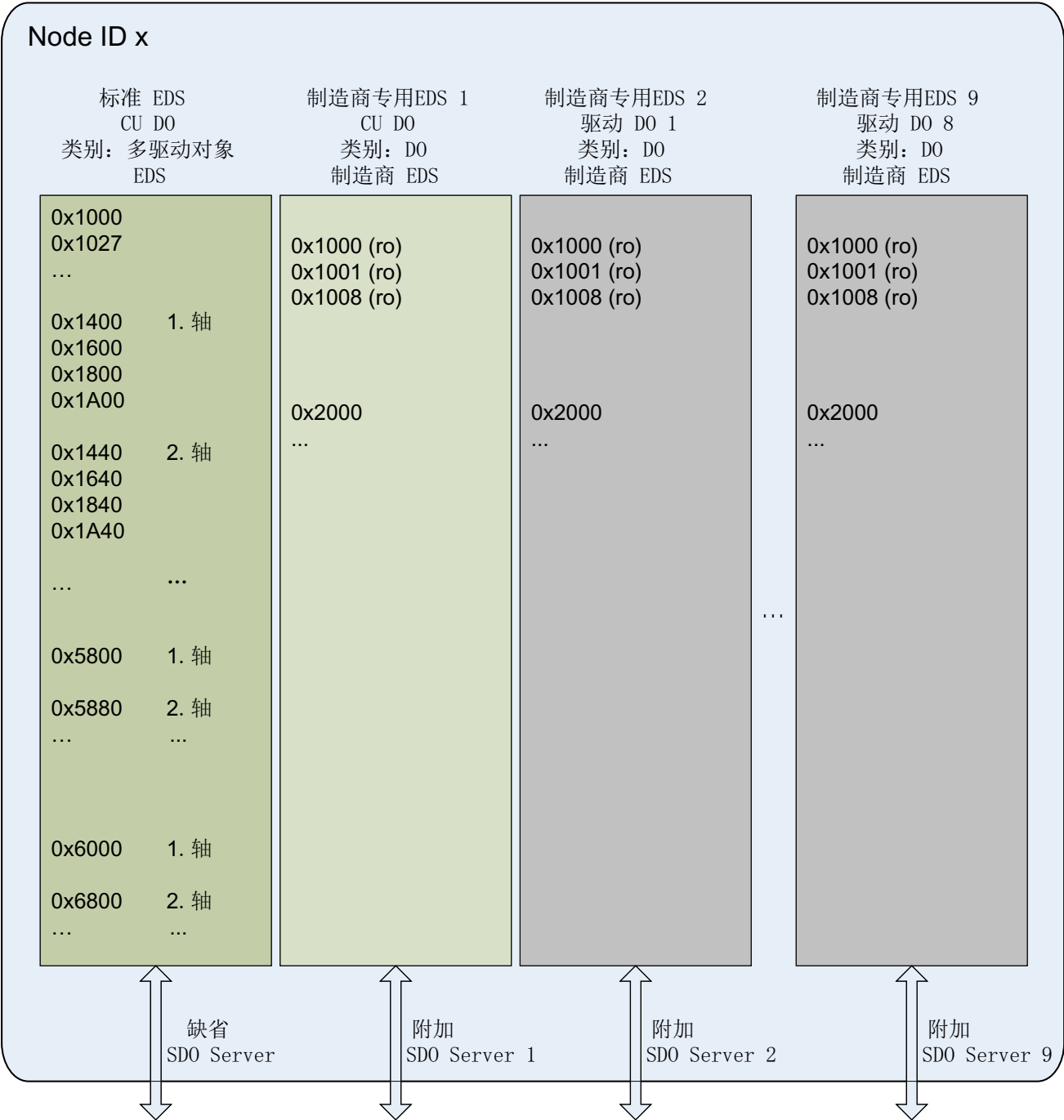


图 8-2 多驱动对象 SINAMICS 驱动设备的建模

8.2 EDS 文件

除了正常 EDS 文件之外，还有用于附加 SDO Server 的标准 CANopen 对象（SDO server parameter）。

此外，还可使用以下 EDS：

- 模块化的、不含制造商专用对象的 EDS 文件
- 用于新模型的两类 EDS 文件（模块化的和制造商专用的）

因此，每个 SINAMICS 驱动设备和 CANopen 支持的每个 SINAMICS 驱动对象各自都有一个 EDS 文件。

所有 SINAMICS 设备的 EDS 文件

Filename	G120P	G120C	G120S_SERVO	G120S_VECTOR	S110	S120	S150	G130	G150	GL150	GM150	SL150	SM120	MV	Kategorie
SINAMICS_G120P.eds	x														Single DO EDS
SINAMICS_G120C.eds		x													Single DO EDS
SINAMICS_G120S_SERVO.eds			x												Single DO EDS
SINAMICS_G120S_VECTOR.eds				x											Single DO EDS
SINAMICS_MV.eds														x	Multi DO EDS
SINAMICS_S120.eds						x									Multi DO EDS
SINAMICS_S110.eds					x										Multi DO EDS
SINAMICS_S150.eds							x								Multi DO EDS
SINAMICS_SL150.eds												x			Multi DO EDS
SINAMICS_SM120.eds													x		Multi DO EDS
SINAMICS_G130.eds								x							Multi DO EDS
SINAMICS_G150.eds									x						Multi DO EDS
SINAMICS_GL150.eds										x					Multi DO EDS
SINAMICS_GM150.eds											x				Multi DO EDS
SINAMICS_S120_CU.eds						x									DO Manufacturer EDS
SINAMICS_S150_CU.eds							x								DO Manufacturer EDS
SINAMICS_G130_CU.eds								x							DO Manufacturer EDS
SINAMICS_G150_CU.eds									x						DO Manufacturer EDS
SINAMICS_MV_CU.eds														x	DO Manufacturer EDS
SINAMICS_SERVO.eds						x									DO Manufacturer EDS
SINAMICS_VECTOR.eds						x	x	x	x						DO Manufacturer EDS
SINAMICS_VECTORMV.eds											x	x		x	DO Manufacturer EDS
SINAMICS_A_INF2C.eds														x	DO Manufacturer EDS

图 8-3 SINAMICS 设备的 EDS 文件

8.2 EDS 文件

Multi DO EDS

设备专用的、不含制造商专用对象的 EDS。

文件名：“SINAMICS_设备名称.eds”

CANopen 对象	名称
通讯子协议中的对象，号段 0x1000 - 0x1FFF:	
0x1000	Device Type
0x1001	Error Register
0x1003	Pre-defined error field
0x1005	COB-ID SYNC message
0x1008	Manufacturer device name
0x100A	Manufacturer software version
0x100C	Guard time
0x100D	Life time factor
0x1010	Store parameters
0x1011	Restore default parameters
0x1014	COB-ID EMCY
0x1017	Producer heartbeat time
0x1018	Identity object
0x1027	Module list
0x1029	Error behavior object
0x1200	SDO server parameter Default SDO channel
0x1201	SDO server parameter CU DO channel
0x1200 + (n+1)	其中，n 为 1 - 8 SDO server parameter Drive DO channels
0x1400 + (n-1) * 0x40 bis 0x1407 + (n-1) * 0x40	其中，n 为 1 - 8 根轴 RPDO communication parameter
0x1600 + (n-1) * 0x40 bis 0x1607 + (n-1) * 0x40	其中，n 为 1 - 8 根轴 RPDO mapping parameter

CANopen 对象	名称
$0x1800 + (n-1) * 0x40$ bis $0x1807 + (n-1) * 0x40$	其中, n 为 1 - 8 根轴 TPDO communication parameter
$0x1A00 + (n-1) * 0x40$ bis $0x1A07 + (n-1) * 0x40$	其中, n 为 1 - 8 根轴 TPDO mapping parameter
所有 n 个逻辑设备的制造商专用号段内的自由过程数据对象 (n 个轴)	
$0x5800 + (n-1) * 0x80$ bis $0x580F + (n-1) * 0x80$	其中, n 为 1 - 8 根轴 16 Bit free receive PDOs
$0x5810 + (n-1) * 0x80$ bis $0x581F + (n-1) * 0x80$	其中, n 为 1 - 8 根轴 16 Bit free transmit PDOs
$0x5820 + (n-1) * 0x80$ bis $0x5827 + (n-1) * 0x80$	其中, n 为 1 - 8 根轴 32 Bit free receive PDOs
$0x5830 + (n-1) * 0x80$ bis $0x5837 + (n-1) * 0x80$	其中, n 为 1 - 8 根轴 32 Bit free transmit PDOs
所有 n 个逻辑设备的标准协议号段内的对象 (n 个轴)	
$0x6000 + (n-1) * 0x800$ bis $0x67FF + (n-1) * 0x800$	其中, n 为 1 - 8 根轴

DO Manufacturer EDS

包含制造商专用对象的驱动对象专用 EDS。

控制单元的 EDS 名: "SINAMICS_设备名称_CU.eds"

DO 文件名: "SINAMICS_DOClassID.eds"

CANopen 对象	名称
通讯子协议号段 0x1000 - 0x1FFF 中的标准对象:	
0x1000 (read only)	Device Type
0x1001 (read only)	Error Register
0x1018 (read only)	Identity Object
制造商专用号段 0x2000 - 0x57FF 中的对象: 换算公式: 制造商专用的对象索引 = SINAMICS 参数编号(hex) + 0x2000	

8.3 SDO Server

缺省 SDO Server

通过缺省 SDO Server 可以访问标准对象，也可以访问“Multi DO EDS”类别中的 EDS 文件。

修改缺省值（p8630[0] >= 1）还可以访问制造商专用的对象。这尤其适用于只具有缺省 SDO Server 的单个驱动对象。

服务器缺省 SDO 参数：

0x1200.0 = 2

0x1200.1 = 0x600 + Node ID

0x1200.2 = 0x580 + Node ID

缺省 SDO Server 的服务器 SDO 参数只读，且在 SINAMICS 参数 r8610 中映射。

附加的 SDO Server

通过缺省 SDO Server 后的下一个 SDO Server 可以访问控制单元的制造商专用对象，也可以访问“DO Manufacturer EDS”类别的 EDS 文件。

控制单元服务器 SDO 参数：

0x1201.0 = 2

0x1201.1 = 0x601 + Node ID

0x1201.2 = 0x581 + Node ID

在 EDS 文件中，对象位于固定的区域中。

通过 n 个其他的 SDO Server 可以访问 CAN 驱动对象的制造商专用的对象（n 为 1 - 8），也可以访问“DO Manufacturer EDS”类别的 EDS 文件。

在 CAN 不支持的驱动对象类别无法进行 SDO 访问。

第 n 个轴的服务器 SDO 参数:

$(0x1201 + n).0 = 2$ 其中, n 为 1 - 8 根轴

$(0x1201 + n).1 = 0x601 + \text{Node ID} + n$ 其中, n 为 1 - 8 根轴

$(0x1201 + n).2 = 0x581 + \text{Node ID} + n$ 其中, n 为 1 - 8 根轴

在 EDS 文件中, 对象分别位于模块化的区域中。

附加的 SDO Server 可用于改善寻址, 而不适用于并行 SDO 访问。访问参数管理器时须考虑到应答时间。

所有附加的 SDO Server 在出厂设置中都是无效的 (通过位 31 设置), 如果需要使用附加的 SDO 通道, 必须手动激活。

附加的 SDO Server 的服务器 SDO 参数可写, 映射在 SINAMICS 参数 r8612 中。

可在 CANopen SDO 号段 0x601 - 0x67F (接收 SDO) 或用于发送 SDO 的 CANopen SDO 号段 0x581 - 0x5FF (发送 SDO) 内自由选择附加的 SDO Server 的 CAN 标识符。

8.4 制造商专用 EDS 文件的对象

在相应的制造商专用 EDS 文件中含有用于一系列 SINAMICS 参数的制造商专用 CANopen 对象：

参数名称	参数参照	CANopen 对象
电流极限	p0640	0x2280
工艺控制器使能	p2200	0x2898
工艺控制器实际值滤波器时间常数	p2265	0x28D9
工艺控制器微分时间常数	p2274	0x28E2
工艺控制器的比例增益	p2280	0x28E8
工艺控制器的最大限制	p2291	0x28F3
工艺控制器的最小限制	p2292	0x28F4
输出频率	r0066	0x2042
经过滤波的转速设定值	r0020	0x2014
经过滤波的输出频率	r0024	0x2018
经过滤波的输出电压	r0025	0x2019
经过滤波的直流母线电压	r0026	0x201A
经过滤波的电流实际值	r0027	0x201B
经过滤波的转矩实际值	r0031	0x201F
有功功率实际值	r0032	0x2020
电机温度	r0035	0x2023
功率单元温度	r0037	0x2025
能量显示	r0039	0x2027
指令数据组 CDS 激活	r0050	0x2032
状态字 2 (ZSW2) 激活	r0053	0x2035
控制字 1 (STW1) 激活	r0054	0x2036
控制单元数字量输入的状态	r0722	0x22D2
控制单元数字量输出的状态	r0747	0x22EB
控制单元模拟量输入电压/电流	r0752	0x22F0
控制单元模拟量输出电压/电流	r0774	0x2306

参数名称	参数参照	CANopen 对象
故障编号	r0947	0x23B3
当前脉冲频率	r1801	0x2709
报警号	r2110	0x283E
斜坡函数发生器后的工艺控制器设定值	r2260	0x28D4
经过滤波的工艺控制器实际值	r2266	0x28DA
工艺控制器的输出信号	r2294	0x28F6
工艺控制器斜升时间	p2257	0x28D1
工艺控制器斜降时间	p2258	0x28D2
工艺控制器积分时间	p2285	0x28ED

制造商专用的 CANopen 对象

8.4 制造商专用 *EDS* 文件的对象

说明

全部的参数、故障、报警以及 SINAMICS 驱动组中 CANopen 的功能图都请参见 SINAMICS S120/S150 参数手册。

A.1 词汇表

在使用基于 CAN 总线的 CANopen 协议时，以下术语和缩写常会出现：

CAL (CAN Application Layer, CAN 应用层)

基于 CAN 总线的高级通信层，是 CAN 总线在开放通信系统中的应用；CAL 由 NMT, DBT, LMT 和 CMS 组成。因为 CAL 范围很广且在使用时具有高度的灵活性，因此 CANopen 通讯子协议 CiA DS 301 定义了用于自动化技术的 CAL 功能子集。

CAN (Controller Area Network, 控制器局域网)

串行总线系统（也称为 CAN 总线）原本开发用于汽车工业，但现在也用于自动化领域。CANopen（见下文）基于 CAN 总线协议定义了更多层。

CAN 控制器

电子模块，其硬件对 CAN 总线协议进行处理。

CAN 标识符

通过将 CAN 标识符分配至 CAN 消息（CANopen: PDO, SDO）来确定相关 CAN 信息之间的优先级。

CANopen

由 CiA 定义的基于 CAN 总线和 CAL 的通讯模型。为了使在一条总线上使用不同制造商的设备能更加简便，CANopen 通信子协议 CiA DS 301 定义了用于自动化工艺的 CAL 功能子集。以此为基础也定义了用于特定设备类型（例如驱动）的其他子协议。

CiA (CAN in Automation international users and manufacturers group)

使用 CAN 接口的设备的制造商和用户协会。

CMS (CAN Message Specification, CAN 信息规范)

CAL 的组成部分，用于定义数据传输的各种机制。

COB (Communication Object, 通讯对象)

CAN 主线上数据以数据包形式传输，也称为通讯对象 (COB) (另一名称为 CAN-Message, CAN 报文)。

CAN 总线上连接的设备可以发送和接收 COB。

COB-ID (COB-Identifier, COB 标识符)

每个 COB 都通过标识符确保其唯一性，标识符是 COB 的组成部分。CAN 标准 2.0A 最多支持 2048 个 COB，每个 COB 可通过 11 位的标识符识别。在本调试手册中，COB-ID 始终以十六进制值给定。

在相关驱动设备的对象字典中可以获取 COB 标识符列表，该表中包含所有可通过 CAN 访问的 COB。

DRIVECOM

开发驱动网络标准 (协议) 的驱动制造商协会。伺服放大器中执行的驱动定位 DRIVECOM 协议 22，是 CiA 开发 CANopen 驱动子协议 CiA DSP 402 的基础。

EDS (Electronic Data Sheet)

电子数据页

EMCY (Emergency, 紧急对象)

SINAMICS 提供紧急对象，用于向 CANopen 总线上的其它节点发送内部设备故障或 CAN 总线故障的信息。EMCY 被设置了高优先级，并发送驱动设备状态的重要信息。

NMT (Network Management, 网络管理)

CAN 的组成部分，用于初始化、配置和故障处理。

Node-ID (Node identification, 节点标识符)

在 CANopen 网络中确定设备的唯一性，即所有设备必须有自己的 Node-ID (总线地址)。COB-ID 的默认分配 (缺省设置) 通过 Node-ID 导出。在本调试手册中，Node-ID 给定为十六进制值。

OD（对象字典）

每个驱动设备都有一个包含所有驱动支持的对象的数据库，即对象字典。对象字典中包含：

- 设备的型号、描述和序列号
- 每个对象的名称、格式、描述 + 索引
- PDO、SDO 列表
- 分配至 PDO 的数据
- 发送 PDO 的时间（SYNC，对象中的修改等）
- 发送紧急信息的时间
- ...

通过对象对所有驱动设备变量进行访问。SDO 和 PDO 的通讯服务访问驱动设备的对象字典。

PDO（Process Data Object，过程数据对象）

用于快速实时访问所选数据。特定变量已经预先定义了和特定 PDO 的映射关系。

SDO 用于访问其它变量。

协议

在总线系统通信领域，协议是用于设备标准化的文档。协议从通讯接口的角度对通信功能（通信子协议）、设备功能（设备子协议）或驱动功能（驱动子协议）进行描述。

RPDO（接收 PDO，接收过程数据对象）

由设备接收的 PDO（例如目标位置）。

SDO（Service Data Object，服务数据对象）

通过 SDO 可访问 CANopen 设备中所有变量；在驱动中即驱动和 CANopen 变量。

SDO 通常用于配置。PDO 用于快速实时访问所选变量。

SYNC（Synchronisation，同步）

SYNC 是用于同步 CAN 设备的专用报文。此报文具有很高的优先级。

TPDO（发送 PDO，发送过程数据对象）

由驱动发送的 PDO（例如位置实际值）。

变量

通过变量访问驱动功能和 CANopen 功能。

可通过 SDO 或 PDO 访问变量。

索引

B

BICO 互联, 37, 61

Bootup 协议, 38

C

CAL, 176

CAN, 176

CAN 标识符, 176

CAN 控制器, 176

CANopen, 176

CANopen 接口, 14

CAN 总线

采样时间, 51

接口 X451, 56

接口 X452, 56

CANopen

BICO 互联, 37

CBC10, 57

EDS 文件, 167

EDS 文件列表, 167

EDS 扩展, 172

EDS 类型, 168

PDO 访问, 118, 120

SDO Server, 170

SDO 访问, 118, 120

支持的运行模式, 133

对象字典, 15

矢量控制, 52

设备状态机, 42

访问自由的 PZD 对象, 120

访问标准的 PZD 对象, 118

伺服控制, 52

制造商专用 EDS 文件, 172

制造商专用对象, 17

映射模型, 163

CANopen 对象, 159

CBC10, 14, 55

CANopen, 57

安装, 57

诊断 LED, 59

CiA, 176

CMS, 176

COB, 177

COB 标识符, 177

COB-ID, 177

D

DRIVECOM, 177

E

EDS, 177

EDS 文件, 167

所有 SINAMICS 设备的列表, 167

EDS 扩展, 172

EDS 类型, 168

DO Manufacturer EDS, 169

EMCY, 177

EMERGENCY, 26

N

NMT, 26, 177

CANopen 接口

开机调试手册, IH2, 01/2013, 6SL3097-4AA00-0RP0

启动后的状态, 41

NMT 从站, 39

NMT 协议, 41

NMT 服务, 40

状态图, 40

Node-ID, 177

O

OD, 178

P

PDO, 22, 178

传输方式, 35

服务, 33

映射参数, 33

通讯参数, 33

R

RPDO, 178

S

SDO, 178

中止代码, 31

写入协议, 29

协议, 29

服务, 28

读取协议, 30

SDO Server, 170

SDO 协议

传输中止记录协议, 30

STARTER, 14

SYNC, 26, 178

服务, 36

T

TPDO, 179

三划

与驱动无关的控制单元通讯对象, 138

四划

专业术语, 176

心跳, 45, 58, 87

心跳协议, 47

计算/换算

转矩设定值设定和转矩实际值计算, 126

转速设定值设定和转速实际值计算, 123

五划

发送, 14

对象字典, 137, 178

术语, 176

矢量, 79

节点寿命, 45

节点保护, 58, 87

节点保护协议, 45

六划

协议, 178

协议转矩模式, 58

协议速度模式, 58

安全提示

概述, 7

网络管理 (NMT 服务), 39

自由 PDO 映射, 96

自由对象, 21, 158

设定值通道

无, 82

有, 82

七划

伺服, 79

寿命系数, 45

寿命保护, 45

状态字, 114

诊断, 129

报警对象, 130

驱动设备内部故障列表, 131

故障寄存器, 132

紧急对象, 130

预定义故障区, 131

诊断 LED, 59

词汇表, 176

运行模式, 133

驱动子协议 DS402, 159

驱动子协议 DS402 的对象字典, 159

八划

参数

出厂设置, 50

保存, 49

参数出厂设置, 50

和驱动相关的通讯对象, 141

转矩设定值, 111, 126

转矩实际值, 115, 127

转速设定值, 110

转速实际值, 114, 125

转速换算系数, 123

九划

保护时间, 45

八划

变量, 179

九划

故障监控协议, 45

故障监控服务, 45

故障寄存器, 132

映射模型

用一个 SDO Server 映射多驱动对象 SINAMICS 驱动设备, 164

用多个 EDS 文件和多个 SDO Server 映射多驱动对象 SINAMICS 驱动设备, 165

首次调试, 72

十划

紧急对象, 130

调试

CBC10 上的硬件设置, 74

互联 CBC 状态字, 114

互联发送缓冲器, 111

互联电源, 92

互联位置实际值, 115

互联转矩设定值, 111

互联转矩实际值, 115

互联转速设定值, 110

互联转速实际值, 114

互联接收缓冲器, 108

互联接收缓冲器和发送缓冲器, 104, 107

加载并保存项目, 116

在线查找驱动设备, 76

自动配置, 78

设置 CU320-2 上的接口 CBC10, 84, 85, 86, 87

访问自由的 PZD 对象, 120

访问标准的 PZD 对象, 118

将项目加载到驱动设备上, 89

- 配置发送报文和接收报文, 93, 96, 97
- 配置电机, 80
- 配置驱动设备, 76
- 控制字互联, 110
- 通讯对象, 137
 - 与驱动无关的, 138
 - 与驱动相关的, 141
 - 发送 PDO, 148
 - 其他, 155
 - 接收 PDO, 141
- 通讯板, 55
- 速度模式, 58
- 预定义故障区, 131

十一划

- 接收, 14
- 控制字, 110
- 斜坡函数发生器, 82

十二划

- 编码器位置实际值, 115

十三划

- 触发位, 47

十四划

- 缩写, 176

Siemens AG
Industry Sector
Drive Technologies
Motion Control Systems
Postfach 3180
91050 ERLANGEN
GERMANY

www.siemens.com/motioncontrol

保留变更权利
© Siemens AG 2004 – 2013