SIEMENS

SIMATIC

S7-1500, ET 200MP, ET 200SP

CM PtP - 点□点□接的□□

功能手□

版本 08/2024

siemens.com

SIEMENS

SIMATIC

S7-1500 / ET 200MP / ET 200SP CM PtP - 点对点连接的组态

功能手册

08/2024

A5E37509416-AF

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 Siemens Aktiengesellschaft 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。

Siemens Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
德国

A5E37509416-AF
Ⓟ 09/2024 本公司保留更改的权利

前言

本文档的用途

本文档中介绍了有关组态和调试 S7-1500 (ET 200MP) 和 ET 200SP 点对点通信模块的重要信息。

所需基本知识

理解本文档中的内容，需要具备以下知识：

- 有关自动化技术的基本知识
- SIMATIC 工业自动化系统基本知识
- Windows 计算机操作技能
- 熟练掌握 STEP 7

本文档的适用范围

本文档适用于在 STEP 7 (TIA Portal) V12 和更高版本中操作的所有 S7-1500 (ET 200MP) 点对点通信模块和 ET 200SP 。

约定

本手册中使用的术语“CPU”既可指代 S7-1500 的 CPU，也可指代诸如 IM 155-5 等分布式 I/O 系统的接口模块。

另请遵循以下标注的注意事项：

说明

这些注意事项中包含有关本文档中所述产品、产品操作或应特别关注部分的重要信息。

回收和处置

为了以环保且可持续的方式回收和处置您的旧设备，请与经认证的电子废品处理公司联系，并根据您所在国家/地区的适用法规来处置设备。

更多支持

有关各种 SIMATIC 产品与系统的技术文档，敬请访问 Internet (http://www.siemens.com/simatic-tech-doku-portal)。

Siemens 工业在线支持

在此处可轻松快速地获取以下主题的最新信息：

- 产品支持

  提供了产品的所有信息和广泛的专有知识、技术规范、常见问题与解答、证书、下载资料和手册。

- 应用示例

  提供了解决自动化任务所使用的工具以及相关示例，还提供了函数块、性能信息以及视频。

- 服务

  介绍了行业服务、现场服务、技术支持、备件和培训提供情况的相关信息。

- 论坛

  提供了自动化技术相关的答疑和解决方案。

- 我的技术支持

  该部分是您在工业在线支持中的个人工作区，其中提供了消息、支持查询和可组态的文档。

由 Internet (https://support.industry.siemens.com) 上的西门子工业在线支持提供这部分信息。

网上商城

网上商城即为 Siemens AG 基于全集成自动化 (TIA) 和全集成能源管理 (TIP) 的自动化与驱动器解决方案领域的目录和订购系统。

Internet (https://mall.industry.siemens.com) 和信息和下载中心 (https://www.siemens.com/automation/infocenter)提供了自动化和驱动器领域的所有产品目录。

安全信息

西门子的产品及解决方案中包含工业网络安全功能，可确保工厂、系统、机器和网络的安全运行。

为了保护工厂、系统、机器和网络防止受到网络攻击，需要实施并持续维护先进的全方位工业网络安全保护措施。西门子的产品和解决方案是这个概念的一个要素。

客户有责任防止其工厂、系统、机器和网络遭受未经授权的访问。只有在必要时并采取了适当的安全措施（例如防火墙和/或网络分段）的情况下，系统、机器和组件才能连接到企业网络或互联网。

有关实施保护性工业网络安全措施的更多信息，请访问此处 (http://www.siemens.com/cybersecurity-industry)。

Siemens 不断对产品和解决方案进行开发和完善以提高安全性。西门子强烈建议您及时更新产品并始终使用最新产品版本。如果使用的产品版本不再受支持，或者未能应用最新的更新程序，客户遭受网络攻击的风险会增加。

要随时了解有关产品更新的信息，请订阅 Siemens Industrial Cybersecurity RSS Feed：网址 (https://www.siemens.com/cert)。

目录

功能手册文档指南

1

1.1 功能手册文档指南

SIMATIC S7-1500 自动化系统、基于 SIMATIC S7-1500 和 SIMATIC ET 200MP 的 1513/1516pro-2 PN, SIMATIC Drive Controller CPU、ET 200SP、ET 200AL 和 ET 200eco PN 分布式 I/O 系统的文档分为 3 个部分。

用户可根据需要快速访问所需内容。

相关文档，可从 Internet 免费下载。
(https://support.industry.siemens.com/cs/cn/zh/view/109742705)

基本信息

系统手册和入门指南中详细描述了 SIMATIC S7-1500, SIMATIC Drive Controller, ET 200MP、ET 200SP、 ET 200AL 和 ET 200eco PN 系统的组态、安装、接线和调试。对于 1513/1516pro-2 PN CPU，可参见相应的操作说明。

STEP 7 在线帮助为用户提供了组态和编程方面的支持。

示例：

- S7-1500 入门指南
- 系统手册
- ET 200pro 和 1516pro-2 PN CPU 操作说明
- TIA Portal 在线帮助

设备信息

设备手册中包含模块特定信息的简要介绍，如特性、接线图、功能特性和技术规范。

示例：

- CPU 设备手册
- “接口模块”设备手册
- “数字量模块”设备手册
- “模拟量模块”设备手册
- “通信模块”设备手册

- “工艺模块”设备手册
- “电源模块”设备手册
- BaseUnit 设备手册

常规信息

功能手册中包含有关 SIMATIC Drive Controller 和 S7-1500 自动化系统的常规主题的详细描述。

示例：

- 《诊断》功能手册
- 《通信》功能手册
- 《运动控制》功能手册
- 《Web 服务器》功能手册
- 《周期和响应时间》功能手册
- PROFINET 功能手册
- PROFIBUS 功能手册

产品信息

产品信息中记录了对这些手册的更改和补充信息。本产品信息的优先级高于设备手册和系统手册。

有关产品信息的最新版本，敬请访问 Internet：

- S7-1500/ET 200MP (https://support.industry.siemens.com/cs/cn/zh/view/68052815/en)
- SIMATIC Drive Controller (https://support.industry.siemens.com/cs/de/zh/view/109772684/zh)
- 运动控制 (https://support.industry.siemens.com/cs/de/zh/view/109794046/zh)
- ET 200SP (https://support.industry.siemens.com/cs/cn/zh/view/73021864)
- ET 200eco PN (https://support.industry.siemens.com/cs/cn/zh/view/109765611)

手册集

手册集中包含系统的完整文档，这些文档收集在一个文件中。

可以在 Internet 上找到手册集：

- S7-1500/ET 200MP/SIMATIC Drive Controller (https://support.industry.siemens.com/cs/cn/zh/view/86140384)
- ET 200SP (https://support.industry.siemens.com/cs/cn/zh/view/84133942)
- ET 200AL (https://support.industry.siemens.com/cs/cn/zh/view/95242965)
- ET 200eco PN (https://support.industry.siemens.com/cs/cn/zh/view/109781058)

1.2 SIMATIC 技术文档

附加的 SIMATIC 文档将完善信息。可通过以下链接和 QR 代码获取这些文档及其用途。

借助“工业在线技术支持”，可获取所有主题的相关信息。应用示例用于帮助用户实施相应的自动化任务。

SIMATIC 技术文档概述

可以在此处找到西门子工业在线技术支持中可用的 SIMATIC 文档的概述：

工业在线技术支持（国际）
(https://support.industry.siemens.com/cs/cn/zh/view/109742705)

观看此短视频，了解在西门子工业在线技术支持中可以直接找到概述的位置以及如何在移动设备上使用西门子工业在线技术支持：

每个视频快速介绍自动化产品的技术文档
(https://support.industry.siemens.com/cs/cn/zh/view/109780491)

YouTube 视频：西门子自动化产品 - 技术文档一览
(https://youtu.be/TwLSxxRQQsA)

保留文档

保留本文档供以后使用。

对于以数字形式提供的文档：

1. 在收到您的产品后和初始安装/调试之前下载关联的文档。使用以下下载选项：
   - 工业在线技术支持（国际）：(https://support.industry.siemens.com)

     订货号用于将文档分配给产品。订货号标记在产品和包装标签上。具有新的、不兼容功能的产品会被分配一个新的订货号和文档。
   - ID 链接：

     产品可能具有 ID 链接。ID 链接是二维码，其中带有边框且右下角为黑色。通过 ID 链接可访问产品的数字铭牌。使用智能手机摄像头、条形码扫描仪或阅读器应用程序扫描产品或包装标签上的二维码，即可调用 ID 链接。
2. 保留此版本文档。

更新文档

产品的文档以数字形式更新。特别是在功能扩展的情况下，新的性能特征会在更新版本中提供。

1. 根据上述描述，通过工业在线支持或 ID 链接下载当前版本。
2. 同时保留此版本文档。

我的技术支持

通过"我的技术支持"，可以最大程度善用您的工业在线支持服务。

注册	要使用"我的技术支持"中的所有功能，必须先进行注册。注册后，可以在个人工作区中创建过滤器、收藏夹和选项卡。
支持申请	支持申请页面还支持用户资料自动填写，用户可随时查看当前的所申请的支持请求。
文档	在"文档"(Documentation) 区域中，可以构建您的个人库。
收藏夹	可使用"添加到我的技术支持收藏夹"(Add to mySupport favorites) 来标记特别感兴趣或经常需要的内容。在"收藏夹"(Favorites) 下，会显示所标记条目的列表。

最近查看的文章	“我的技术支持”中最近查看的页面位于“最近查看的文章”(Recently viewed articles) 下。
CAx 数据	借助 CAx 数据区域，可以访问 CAx 或 CAe 系统的最新产品数据。仅需单击几次，用户即可组态自己的下载包： • 产品图片、二维码、3D 模型、内部电路图、EPLAN 宏文件 • 手册、功能特性、操作手册、证书 • 产品主数据

有关“我的技术支持”，敬请访问 Internet。
(https://support.industry.siemens.com/My/ww/zh)

应用示例

应用示例中包含有各种工具的技术支持和各种自动化任务应用示例。自动化系统中的多个组件完美协作，可组合成各种不同的解决方案，用户无需再关注各个单独的产品。

有关应用示例，敬请访问 Internet。
(https://support.industry.siemens.com/cs/ww/zh/ps/ae)

1.3 工具支持

下面介绍的工具在所有步骤中都会为您提供支持：从规划到调试，再到系统分析。

TIA Selection Tool

TIA Selection Tool 工具可在为 Totally Integrated Automation (TIA) 选择、组态和订购设备时提供支持。

作为 SIMATIC Selection Tools 的后继产品，TIA Selection Tool 将已知的自动化技术组态器组装到一个工具中。

借助 TIA Selection Tool，用户可基于产品选型或产品组态生成完整的订单表。

有关 TIA Selection Tool，敬请访问 Internet。
(https://support.industry.siemens.com/cs/cn/zh/view/109767888)

SINETPLAN

SINETPLAN (Siemens Network Planner) 是西门子公司推出的一种网络规划工具，用于对基于 PROFINET 的自动化系统和网络进行规划设计。使用该工具时，在规划阶段即可对 PROFINET 网络进行预测型的专业设计。此外，SINETPLAN 还可用于对网络进行优化，检测网络资源并合理规划资源预留。这将有助于在早期的规划操作阶段，有效防止发生调试问题或生产故障，从而大幅提升工厂的生产力水平和生产运行的安全性。

优势概览：

- 端口特定的网络负载计算方式，显著优化网络性能
- 优异的现有系统在线扫描和验证功能，生产力水平大幅提升
- 通过导入与仿真现有的 STEP 7 系统，极大提高调试前的数据透明度
- 通过实现长期投资安全和资源的合理应用，显著提高生产效率

SINETPLAN 可从 Internet 上下载。
(https://new.siemens.com/global/en/products/automation/industrial-communication/profinet/sinetplan.html)

参见

PRONETA Professional (https://support.industry.siemens.com/cs/ww/zh/view/109781283)

简介

2

2.1 通信模块概述

自动化系统包含各式各样的组件。其中还包括通信模块。串行通信通过点对点连接提供简单的数据交换功能。

通过在 OSI 层模型中的较低层设置通信参数，便可以自定义各种通信伙伴（请参见传输安全性 (页 25)部分）。

与 S7-1500、ET 200MP 和 ET 200SP 的点对点连接通信完全依靠具有串行接口的通信模块 (CM) 进行。

就此应用，SIMATIC S7 有许多模块可提供物理接口和基本协议机制。

- RS232：该接口通过附加伴随信号协调伙伴之间的通信。
- RS422/RS485：该接口可通过使用差分电压作为传输技术来延长线路，还可以通过总线结构实现包含 2 台以上设备的结构 (RS485)。

执行 CPU 和 CM 之间协调工作的指令可用于从 CPU 向相应模块传送数据。它们将向用户程序通知是否已成功发送或接收新数据。在没有 SIMATIC CPU 的系统中，用户必须自行编写这些指令的功能 (https://support.industry.siemens.com/cs/ww/zh/view/59062563)。

本功能手册介绍了 PtP 通信模块的功能和用途。

组件和订货号概述

通信模块及其应用适用性的表格概述

通信模块	S7-1500	ET 200MP	ET 200SP	订货号
CM PtP RS232 BA [1]	X	X	-	6ES7540-1AD01-0AA0
CM PtP RS422/485 BA	X	X	-	6ES7540-1AB01-0AA0
CM PtP RS232 HF [2]	X	X	-	6ES7541-1AD01-0AB0
CM PtP RS422/485 HF	X	X	-	6ES7541-1AB01-0AB0
CM PtP (ET 200SP)	-	-	X	6ES7137-6AA01-0BA0

[1] BA = Basic

[2] HF = High Feature

说明

带 IM 155-6 MF HF 的 CM PtP (ET 200SP)

通过 IM 155-6 MF HF 接口模块 (6ES7155-6MU00-0CN0)，可使用除 PROFIBUS/PROFINET 之外的不同现场总线协议。在这种情况下，不能使用指令库 PtP Communication。请注意编程手册《在没有 SIMATIC 系统指令的情况下，运行 CM PtP (https://support.industry.siemens.com/cs/ww/zh/view/59062563)》中的信息。另请参见接口模块的设备手册，可从 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109773210) 上下载。LMf 库多现场总线可实现基于 Modbus/TCP 协议的 ET 200SP IM155-6 MF 到 SIMATIC S7-1500 的连接，更多信息，请参见。

组件和接口概述

通信模块及其功能的表格概述。

通信模块	接口	协议					连接技术		诊断，通过 Event-TracePtP
		Free-port	3964(R)	Modbus 主站	Modbus 从站	USS 主站	D-sub 9 针	D-sub 15 针	固件版本
CM PtP RS232 BA	RS232	X	X	-	-	X	X	-	V2.0 及以上版本
CM PtP RS422/485 BA	RS422	X	X	-	-	X	-	X	V2.0 及以上版本
	RS485	X	-	-	-	X	-	X	V2.0 及以上版本
CM PtP RS232 HF	RS232	X	X	X	X	X	X	-	V2.0 及以上版本
CM PtP RS422/485 HF	RS422	X	X	X	X	X	-	X	V2.0 及以上版本
	RS485	X	-	X	X	X	-	X	V2.0 及以上版本
CM PtP (ET 200SP)	RS232	X	X	X	X	X	ET 200SP BaseUnit [1]		-
	RS422 [2]	X	X	X	X	X			
	RS485	X	-	X	X	X			

[1] 带有端子（而非 D-sub）的 BaseUnit；分配取决于传输物理介质

[2] CM PtP 通信模块还可用于 RS422 模式下的多点连接

组件和数据传输速率概述

通信模块可采用不同的数据传输速率发送和接收数据。下表列出了单个通信模块的分配。

通信模块	数据传输速率 (bps)											
	300	600	1200	2400	4800	9600	19200	38400	57600	76800	115200	250000[1]
CM PtP RS232 BA	X	X	X	X	X	X	X	-	-	-	-	-
CM PtP RS422/485 BA	X	X	X	X	X	X	X	-	-	-	-	-
CM PtP RS232 HF	X	X	X	X	X	X	X	X	X	X	X	-
CM PtP RS422/485 HF	X	X	X	X	X	X	X	X	X	X	X	X
CM PtP (ET 200SP)	X	X	X	X	X	X	X	X	X	X	X	X

[1] 专用于 RS485 接口使用 DMX512 协议时

组件和接收缓冲区大小概述

每个通信模块都有一个缓冲区来缓存接收到的帧。下表显示了单个帧的最大大小分配以及单个通信模块的存储器大小。

模块	接收缓冲区大小 KB	最大帧长度 KB	可缓冲帧
CM PtP RS232 BA	2	1	255
CM PtP RS422/485 BA	2	1	255
CM PtP RS232 HF	8	4	255
CM PtP RS422/485 HF	8	4	255
CM PtP (ET 200SP)	4	2	255

伴随信号和数据流控制

- 使用 XON/XOFF 进行软件数据流控制

  自由口协议支持使用 XON/XOFF 通过 RS232 和 RS422 接口进行数据流控制。

- 使用 RTS/CTS 进行硬件数据流控制

  自由口协议支持使用 RTS/CTS 通过 RS232 接口进行数据流控制。

- 伴随信号的自动操作

  RS232 伴随信号可使用 Freeport、Modbus 主站和 Modbus 从站协议通过 RS232 接口进行控制。（仅在硬件数据流控制激活时可用。）

通信模块的协议

根据所使用的通信模块，可建立具有不同协议的通信连接：

- Freeport：传输不带有指定协议格式的 ASCII 字符串
- 3964(R)：可编程逻辑控制器之间的通信（主站/主站通信）
- Modbus RTU：可编程逻辑控制器之间的通信（主站/主站通信）通信模块既可作为主站，也可作为从站。
- USS：可编程控制器与变频器之间的通信（主站/从站通信）。通信根据变频器技术要求而定制。通信模块只能作为主站。

2.2 处理步骤概述

点对点连接

系统提供了多种在两个或更多通信伙伴之间进行数据交换的联网选项。最简单形式的数据互换是通过两个通信伙伴之间的点对点连接进行的。

通信模块 (CM) 构成一个可编程逻辑控制器和一个或多个通信伙伴之间的接口。数据通过与通信模块的点对点连接以串行模式发送。

组态/参数分配

通信模块的组态包括在 STEP 7 (TIA Portal) 的设备组态中安排通信模块，以及在通信模块的属性对话框中设置特定协议参数（静态组态）。

编程

编程包括通过用户程序将通信模块与相应的 CPU 进行程序特定的连接。使用 STEP 7 (TIA Portal) 对通信模块进行编程。

CPU、通信模块和通信伙伴之间的通信通过指令 (页 87)进行。大量指令适用于 S7-1500 和 S7-1200 自动化系统。可以在用户程序中使用这些指令启动和控制通信，以及改变运行组态（动态组态）。

有关详细信息，请参见指令概述 (页 21)和 STEP 7 (TIA Portal) 在线帮助。

2.3 指令概述

数据通信

可在硬件配置中使用模块的"针对许多短报文进行性能优化"参数来定义数据交换的类型。只要不超过传入/传出报文的最大长度，建议使用性能优化选项。

CPU 与通信模块之间的数据交换有两种类型：

- 异步数据交换

  点对点指令通过读取或写入数据集与通信模块异步通信（相对于应用周期）。数据传输需要经过多个应用周期。最大报文长度需符合模块技术规范。

- 同步数据交换（性能优化选项 (页 47)）

  点对点指令通过通信模块的 IO 数据基于应用周期与通信模块同步进行通信。

  传入报文的最大长度为 24 个字节，传出报文的最大长度为 30 个字节。通过基于应用周期同步使用数据，响应时间得到优化，尤其是在并行使用多个 CM PtP 时。

说明

性能优化选项适用于 V4.0 及更高版本的 PtP Communication 指令库。

指令概述

CPU、通信模块和通信伙伴之间的通信通过支持相应通信模块的特殊指令和协议进行。

PtP Communication 指令库中的指令处理 CPU 与通信模块之间的数据交换。必须从用户程序中调用一次或循环调用。指令会自动检测性能优化选项是否处于活动状态并调整数据交换的方法。

这些指令是 STEP 7 (TIA Portal) 的一部分。它们位于"通信 > 通信处理器"(Communication > Communication processor) 下的"指令"(Instructions) 任务卡中。如果支持所需功能，则适用于所有列出的通信模块。

点对点 指令	含义
Send_P2P (页 110)	将数据发送给通信伙伴。
Receive_P2P (页 114)	接收来自通信伙伴的数据。
Receive_Reset (页 117)	清除通信模块的接收缓冲区。

点对点 指令	含义
Port_Config (页 94)	动态分配基本接口参数。
Send_Config (页 98)	发送参数分配；动态分配端口的串行发送参数。
Receive_Config (页 100)	接收参数分配；动态分配端口的串行接收参数。
P3964_Config (页 107)	协议组态；动态组态程序 3964(R) 的参数。
Signal_Get (页 118)	读取 RS232 伴随信号。
Signal_Set (页 119)	设置 RS232 伴随信号。
Get_Features (页 122)	读取通信模块支持的扩展功能。
Set_Features (页 124)	激活通信模块支持的扩展功能。

USS 指令	含义
USS_Port_Scan (页 186)	通过 USS 网络进行通信。
USS_Drive_Control (页 190)	与驱动器交换数据。
USS_Read_Param (页 195)	从驱动器读取参数。
USS_Write_Param (页 197)	更改驱动器中的参数。

MODBUS 指令	含义
Modbus_Master (页 147)	通过 PtP 端口作为 Modbus 主站进行通信。
Modbus_Slave (页 155)	通过 PtP 端口作为 Modbus 从站进行通信。
Modbus_Comm _Load (页 141)	为 Modbus RTU 组态通信模块的端口。

串行通信的基本知识 3

3.1 串行数据传输

在串行数据传输期间，要传输的信息字符的各个位均按照所定义的顺序依次发送。

双向数据传输 - 工作模式

对于双向数据传输，通信模块具有两种工作模式：

- 半双工操作

  数据在通信伙伴之间在两个方向上交替地进行交换。在半双工工作中，一个通信伙伴发送数据，与此同时，另一个通信伙伴接收数据。在此过程中，一条线路交替着用于发送或接收。

- 全双工操作

  数据在一个或多个通信伙伴之间同时双向交换，也就是说可以同时发送和接收。该过程要求一条线路用于发送，一条线路用于接收。

数据传输

仅在字符传输期间才支持所谓的时基同步（在固定字符串传输时使用的固定计时码）。每个要发送的字符前附加一个同步脉冲，也称为起始位。起始位传输的长度确定时钟脉冲。字符传输结束由一个或两个停止位构成。

声明

除起始位和停止位外，还必须先在发送和接收伙伴之间做进一步声明，然后才能进行串行传输。这些声明包括：

- 数据传输速率
- 帧的开始和结束标准（例如，字符延迟时间）
- 奇偶校验
- 数据位个数（7 或 8 个位/字符）
- 停止位个数（1 或 2 个）

3.2 传输安全性

传输安全性在数据传输和传输程序选择上起着重要作用。一般而言，使用参考模型的层数越多，传输安全性越高。

现有协议的分类

下图说明了通信模块的协议与参考模型的匹配情况。

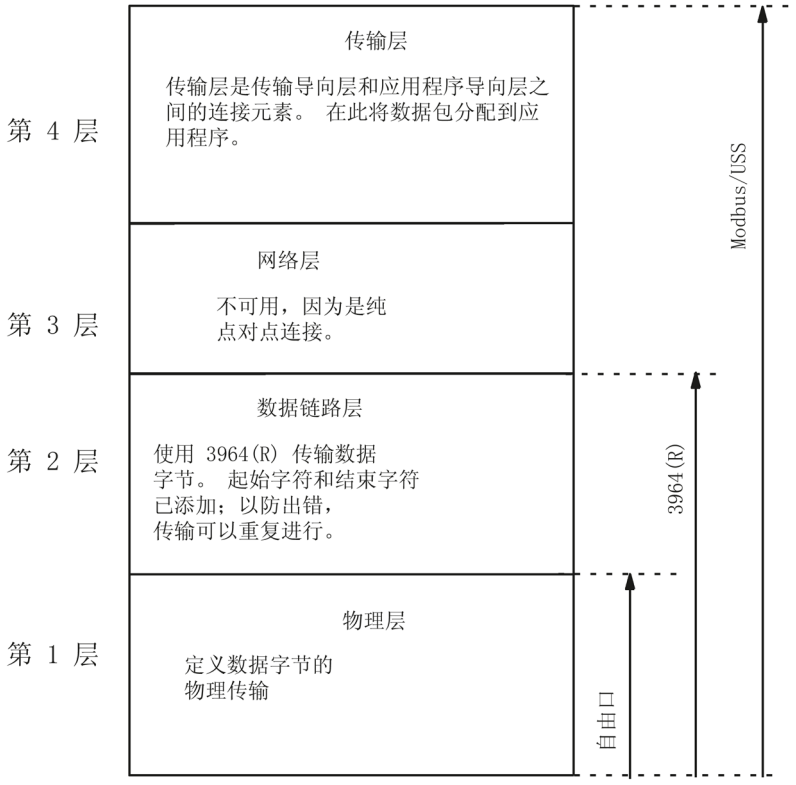


图 3-1 参考模型中现有通信模块协议的分类

使用自由口时的传输安全性

使用自由口时的传输安全性：

- 使用自由口发送数据时，除了使用奇偶校验位外，没有其它的数据保护措施。这意味着使用自由口传输数据非常有效，但数据安全性却无法保证。可通过帧起始条件和帧结束条件的参数分配来实现某种程度上的数据安全性。
- 使用奇偶校验位可确保能够识别出要发送字符中的位的反转。然而，如果字符中有两位或更多位被反转，则无法确保仍能检测到这些错误。
- 例如，为了提高传输安全性，您可以执行检验和、帧长度规范或可组态结束条件。这些措施必须由用户执行。
- 通过对发送或接收帧进行响应的确认帧，可以进一步增强数据安全性。这适用于使用高级协议进行数据通信的情况（ISO 7 层参考模型）。

使用 3964(R) 的传输安全性

奇偶校验位用于提高数据安全；根据组态情况将待传送的数据位数转换为奇数或偶数。

使用奇偶校验位可确保能够识别出要发送字符中的位的反转。然而，如果字符中有两位或更多位被反转，则无法再可靠地检测到这些错误。

如果将奇偶校验设置为“无”，将不传输奇偶校验位。此设置会降低传输安全性。

可以使用两种不同的程序进行数据传输，即使用或不使用块校验字符的数据传输：

- 不带有块校验字符 (BCC) 块检查字符的数据传输：**3964**

  可通过指定的帧结构、帧分解和帧重复来实现传输安全性。

- 带有块检查字符的数据传输：**3964R**

  可通过指定的帧结构和帧分解、帧重复并使用 block check character (BCC) 来实现高度的传输安全性。

在本手册中，当说明和注释提及两个数据传输模式时会使用术语 3964(R)。

Modbus 和 USS 的传输完整性

奇偶校验位用于提高传输安全性；它会根据组态情况将待传送的数据位数转换为奇数或偶数。

使用奇偶校验位可确保能够识别出要发送字符中的位的反转。然而，如果字符中有两位或更多位被反转，则不再能清楚地检测到该错误。

如果将奇偶校验设置为"无"，将不传输奇偶校验位。此设置会降低传输安全性。

Modbus 会额外使用循环冗余校验 (CRC)。使用这种方法时，将在传输数据之前以所谓的 CRC 值形式向用户数据的每个数据块添加附加冗余。这是一个使用特定程序计算的检查值，它可以用来检测传输过程中可能发生的任何错误。

USS 额外使用 BCC（block check character，块检查字符）。块检查字符在接收时形成，并在读入整个帧后与收到的 BCC 进行比较。如果这两个字符不匹配，则不对帧进行评估。如果有一个字符传输不正确，则能够可靠地检出错误。如果有偶数个字符传输不正确，则无法再可靠地检出错误。

3.3 RS232 模式

以下通信模块支持 RS232 模式：

- CM PtP RS232 BA
- CM PtP RS232 HF
- CM PtP (ET 200SP)

在 RS232 模式中，可通过两条线路发送数据。单独的线路可用于发送方向和接收方向。发送和接收可同时进行 (full duplex)。

RS232 信号

除了 TXD、RXD 和 GND 信号外，使用 RS232 硬件时，通信模块还提供额外的 RS232 信号：

TXD	输出	发送的数据； 接口正在发送
RXD	输入	接收的数据； 接口正在接收
GND		共用接地参考（地）； 隔离
DCD	输入	数据载体检测； 连接调制解调器时的载体信号。通信伙伴指明其已识别到进入数据。
DTR	输出	数据终端就绪； DTR 设置为“ON”：通信模块已开启，准备运行 DTR 设置为“OFF”：通信模块未开启，未运行就绪
DSR	输入	数据集准备就绪； DSR 设置为“ON”：通信伙伴发出已准备好运行的信号 DSR 设置为“OFF”：通信方没有开启，未准备好运行
RTS	输出	请求发送； RTS 设置为“ON”：通信模块准备发送；向通信伙伴发出信号，告知数据已准备好发送 RTS 设置为“OFF”：通信模块尚未准备好发送

CTS	输入	清除发送； 通信伙伴可接收来自通信模块的数据（响应通信模块的 RTS = ON） CTS 设置为"ON"：向通信伙伴发出已准备好接收的信号 CTS 设置为"OFF"：向通信伙伴发出"未准备好接收"信号
RI	输入	用于连接调制解调器的呼入（振铃指示器）

通信模块上电后，输出信号处于 OFF 状态（未激活）。

可在通信模块的用户界面中组态 DTR/DSR 和 RTS/CTS 控制信号的操作。

在以下情况中，RS232 信号不会受到影响：

- 已组态的数据流量控制 "Hardware RTS always switched"

  （对应于伴随信号的自动操作）
- 已组态的数据流控制 "Hardware RTS always ON"

  （对应于使用 RTS/CTS 的硬件流量控制）
- 已组态的数据流控制"硬件 RTS 始终开启，忽略 DTR/DSR"(Hardware RTS always ON, ignore DTR/DSR)

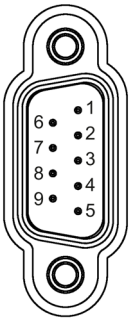
有关详细信息，请参见 握手程序 (页 41) 一章。

连接电缆

下列不同长度的标准连接电缆可用于连接同样具有 9 针 D-sub 连接器的通信伙伴：

订货号	**6ES7902-1AB00-0AA0**	**6ES7902-1AC00-0AA0**	**6ES7902-1AD00-0AA0**
产品型号 名称	S7 连接电缆 RS232		
电缆长度	5 m	10 m	15 m

下表显示了 CM PtP RS232 BA/HF 的 9 针 D-sub 连接器的引脚分配。

引脚连接器	引脚	标识	输入/ 输出	自制连接电缆必选/可选
	1	DCD	输入	可选
	2	RXD	输入	必选
	3	TXD	输出	必选
	4	DTR	输出	可选
	5	GND	—	必选
	6	DSR	输入	可选
	7	RTS	输出	可选
	8	CTS	输入	可选
	9	RI	输入	可选
正视图				

上述列出的连接电缆的电缆或连接器不能作为单独的产品进行订购。如果您自己制作连接电缆，请记住通信伙伴处未连接的输入必须连接至开路电位。

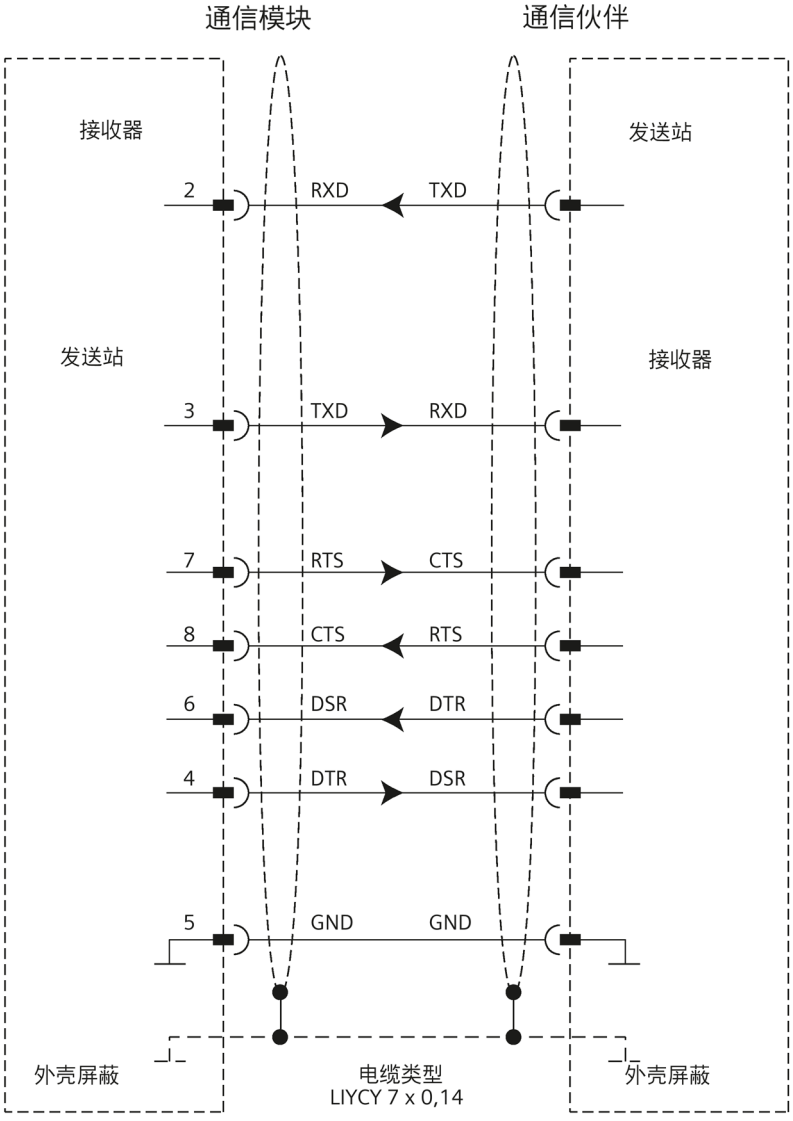
请注意，只能使用带屏蔽的连接器外壳。电缆屏蔽层两侧必须有较大的表面积与连接器外壳接触。

⚠小心

请勿将电缆屏蔽层与 GND 连接

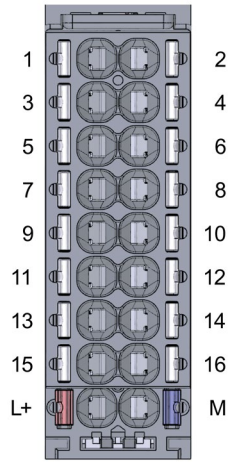
请勿将电缆屏蔽层与 GND 连接，因为这可能会损坏接口。必须始终将 GND 连接在两侧（针脚 5），否则还可能损坏接口模块。

下图说明了在通信模块和通信伙伴之间进行点对点连接时使用的电缆。



BaseUnit

在下表中，可找到 CM PtP (ET 200SP) 的 BaseUnit 引脚分配。

BaseUnit	引脚	标识	输入/输出
	1	TXD	输出
	2	RXD	输入
	3	RTS	输出
	4	CTS	输入
	5	DTR	输出
	6	DSR	输入
	7	DCD	输入
	8	RI	输入
	9	GND	---
	10		
正视图			

⚠小心

请勿将电缆屏蔽层与 GND 连接

请勿将电缆屏蔽层与 GND 连接，因为这可能会损坏接口。必须始终将 GND 连接在两侧（针脚 5），否则还可能损坏接口模块。

3.4 RS422 模式

以下通信模块支持 RS422 模式：

- CM PtP RS422/485 BA
- CM PtP RS422/485 HF
- CM PtP (ET 200SP)

在 RS422 模式下，数据通过两对线（四线制模式）传输。单独的线对可用于发送方向和接收方向。发送和接收可同时进行 (full duplex)。

可在两个或多个通信伙伴之间同时进行数据交换。在 RS422 多点模式中，只有一个从站可在指定时间发送数据。

接口工作模式

下表总结了各种通信模块和协议的接口操作模式。

该通信模块可在 RS422 模式下用于以下拓扑：

- 两个节点之间的连接：点对点连接
- 多个节点之间的连接：多点连接
  （仅适用于 CM PtP (ET 200SP)）

工作模式	说明
全双工 (RS422) 四线制模式（点对点连接）	两个设备在此工作模式中具有相同的优先级。
全双工 (RS422) 四线制模式（多点主站）	该通信模块可用作多点主站。
全双工 (RS422) 四线制模式（多点从站）	该通信模块可用作多点从站。

以下各项适用于 RS422 模式中的多点主站/从站拓扑结构：

- 主站的发送端与所有从站的接收端互连。
- 从站的发送端与主站的接收端互连。
- 只有主站的接收端和一个从站的接收端具有默认设置。所有其他从站在无默认设置的情况下运行。

RS422 信号

使用 RS422 硬件时，通信模块上存在以下信号：

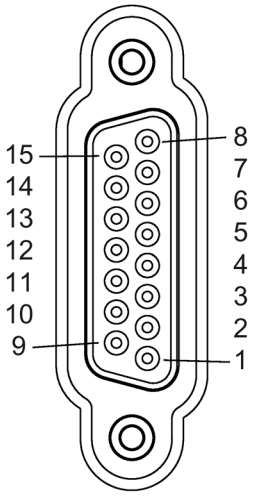
T (A) -	输出	发送的数据
T (B) +	输出	发送的数据
R (A) -	输入	接收的数据
R (B) +	输入	接收的数据
GND		公共接地参考端（地）；浮动

连接电缆

下列不同长度的标准连接电缆可用于连接同样具有 15 针 D-sub 插座的通信伙伴：

订货号	**6ES7902-3AB00-0AA0**	**6ES7902-3AC00-0AA0**	**6ES7902-3AG00-0AA0**
产品型号 名称	S7 连接电缆 RS422		
电缆长度	5 m	10 m	50 m

下表显示了 CM PtP RS232 BA/HF 的 15 针 D-sub 插座的引脚分配。

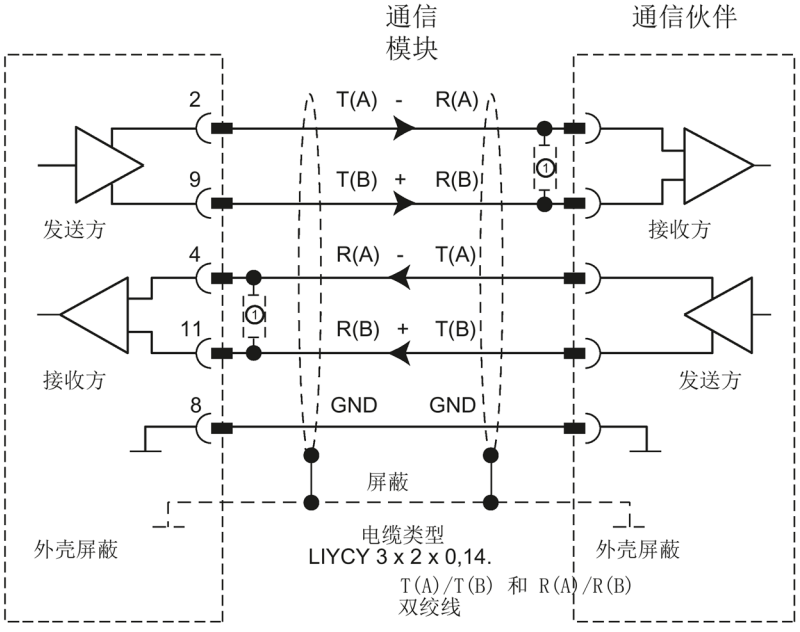
插口	引脚	标识	输入/输出
	1	-	-
	2	T (A) -	输出
	3	-	-
	4	R (A) -	输入
	5	-	-
	6	-	-
	7	-	-
	8	GND	-
	9	T (B) +	输出
	10	-	-
	11	R (B) +	输入
	12	-	-
	13	-	-
	14	-	-
	15	-	-
正视图			

上述列出的连接电缆的电缆或连接器不能作为单独的产品进行订购。如果您自己制作连接电缆，请记住通信伙伴处未连接的输入必须连接至开路电位。

请注意，只能使用带屏蔽的连接器外壳。电缆屏蔽层两侧必须有较大的表面积与连接器外壳接触。

⚠小心
请勿将电缆屏蔽层与 GND 连接 请勿将电缆屏蔽层与 GND 连接，因为这可能会损坏接口。必须始终将 GND 连接在两端（针脚 8），否则可能损坏接口模块。

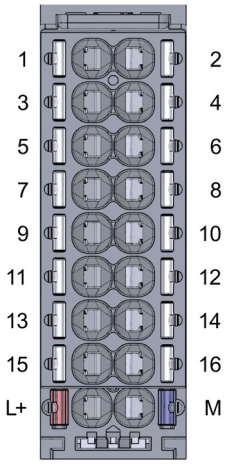
下图说明了在通信模块和通信伙伴之间进行点对点连接时使用的电缆。



①对于长度超过 50 m 的电缆，需要在接收端焊接一个约 330 Ω 的终端电阻以确保无干扰数据通信。

BaseUnit

在下表中，可找到 CM PtP (ET 200SP) 的 BaseUnit 引脚分配。

BaseUnit	引脚	标识	**输入/输出**
	11	T (A) -	输出
	12	R (A) - T(A)/R(A)	输入 输入/输出
	13	T (B) +	输出
	14	R (B) + T(B)/R(B)	输入 输入/输出
	15	GND	---
	16		
正视图			

⚠小心

请勿将电缆屏蔽层与 GND 连接

请勿将电缆屏蔽层与 GND 连接，因为这可能会损坏接口。必须始终将 GND 连接在两侧（针脚 5），否则还可能损坏接口模块。

3.5 RS485 模式

以下通信模块支持 RS485 模式：

- CM PtP RS422/485 BA
- CM PtP RS422/485 HF
- CM PtP (ET 200SP)

在 RS485 模式中，可通过一个线对（两线制操作）传输数据。此线对可交替用于发送和接收方向。发送和接收可交替进行 (half duplex)。完成发送操作后，操作将立即切换到接收模式（准备好接收）。在接收到新的发送作业后会立即再次重置发送模式。

接口工作模式

下表总结了各种通信模块和协议的接口操作模式。

工作模式	说明
半双工 (RS485) 两线制操作	两线制模式下，点对点连接或多点连接（多点）的工作模式。通信模块既可作为主站，也可作为从站。

如果在 RS485 模式（半双工、两线制操作）下运行自由口，则必须在用户程序中采取相应措施，以确保在任意指定时间都只有一个设备在发送数据。如果多个设备同时发送数据，则帧会被破坏。

Modbus 自动确保只有一台设备正在发送。

半双工模式下 RS485 通信模块的切换时间

为发送与接收之间的切换设置最长 0.1 ms 的切换时间。

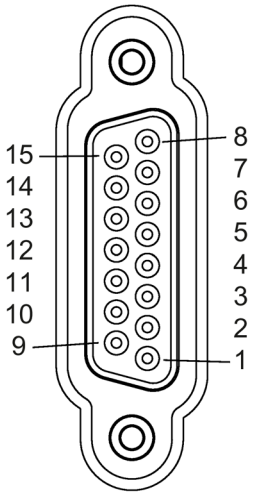
RS485 信号

使用 RS485 硬件时，通信模块上存在以下信号：

R (A)/T (A) -	输入/输出	接收的/发送的数据
R (B)/T (B) +	输入/输出	接收的/发送的数据
GND		公共接地参考端（地）；浮动

连接电缆

下表显示了相应通信模块的 15 针 D-sub 插座的引脚分配。

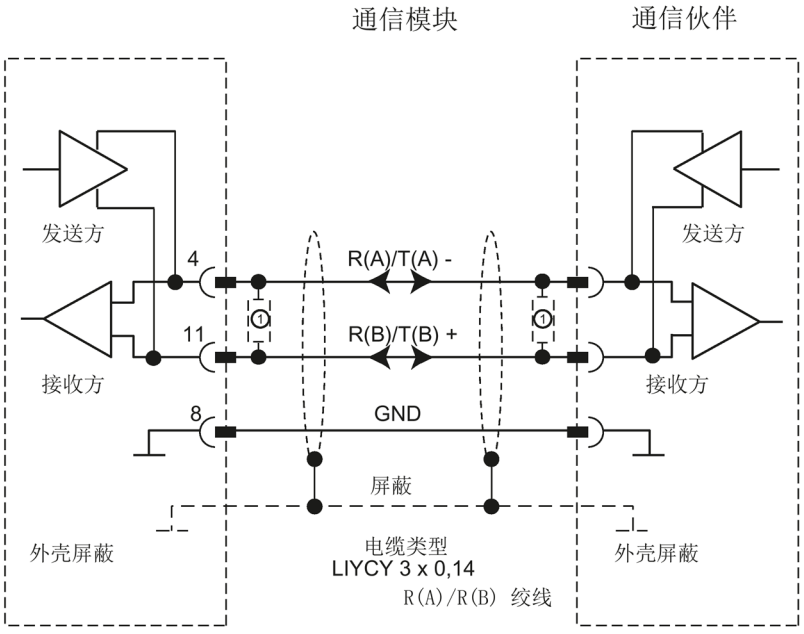
插口	引脚	标识	输入/输出
	1	-	-
	2	-	-
	3	-	-
	4	R (A)/T (A) -	输入/输出
	5	-	-
	6	-	-
	7	-	-
	8	GND	-
	9	-	-
	10	-	-
	11	R (B)/T (B) +	输入/输出
	12	-	-
	13	-	-
	14	-	-
	15	-	-
正视图			

制作连接电缆时，请记住通信伙伴处未连接的输入必须连接至开路电位。

请注意，只能使用带屏蔽的连接器外壳。电缆屏蔽层两侧必须有较大的表面积与连接器外壳接触。

⚠小心
请勿将电缆屏蔽层与 GND 连接 请勿将电缆屏蔽层与 GND 连接，因为这可能会损坏接口。必须始终将 GND 连接在两端（针脚 8），否则可能损坏接口模块。

下图说明了在通信模块和通信伙伴之间进行点对点连接时使用的电缆。



① 对于长度超过 50 m 的电缆，需要在接收端焊接一个约 330 Ω 的终端电阻以确保无干扰数据通信。

3.6 握手程序

简介

握手用于控制两个通信伙伴之间的数据流。如果设备以不同的速度操作，则使用握手方法可以防止在传输期间产生的数据丢失。

我们可从根本上区别以下方法：

表格 3- 1　方法和接口概述

方法	RS232	RS422	RS485
软件数据流控制 XON/XOFF	X	X	-
硬件数据流控制 (RTS/CTS)	X	-	-
伴随信号的自动操作	X	-	-

软件数据流控制

对于通信模块，按照以下方式执行软件数据流控制：

- **XON/XOFF**
  - 通过参数分配将通信模块设置为"XON/XOFF"操作模式后，会发送 XON 字符，从而允许通信伙伴发送数据。
  - 接收缓冲区溢出之前达到组态的最大报文数或 16 个字符时，通信模块将发送字符 XOFF，从而请求通信伙伴中断传输。如果通信伙伴仍然继续发送数据，则在接收缓冲区上溢时将生成一条错误消息。在最后一个帧中接收到的数据将被丢弃。
  - CPU 获取一个帧，并且接收缓冲区准备好再次接收数据后，通信模块会发送 XON 字符。
  - 如果通信模块在发送期间接收到 XOFF 字符，则会取消当前的发送操作，直到再次从通信伙伴接收到 XON。如果在特定的可组态时间内未接收到 XON，则会取消发送操作，并输出相应的错误消息。

说明

可为 XON 和 XOFF 组态字符（任何 ASCII 字符）。

在 XON/XOFF 软件数据流控制的参数分配期间，用户数据不可包含任何已组态的 XON 或 XOFF 字符。

硬件数据流控制

说明

无需为 "Hardware RTS always ON, ignore DTR/DSR" 参数分配连接 DTR/DSR 信号。

如果组态了 "Hardware RTS always ON"，则必须将所使用的接口信号完全连接起来。请确保本地 RTS（输出）与通信伙伴的 CTS （输入）相连接，而本地 CTS 与通信伙伴的 RTS 相连接。相应地，本地 DTR 必须与通信伙伴的 DSR 相连接，而本地 DSR 与通信模块的 DTR 相连接。

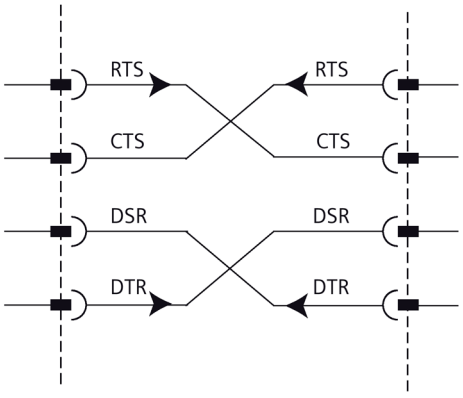


图 3-2　　接口信号的接线

- **硬件 RTS 始终切换，忽略 DTR/DSR**
  - 通过参数分配将通信模块设置为"硬件 RTS 始终开启"的操作模式后，会向通信伙伴输出 RTS = ON 信号以指示其就绪状态。
  - 只要在缓冲区上溢前达到所组态的最大帧数或 16 个字符，RTS 就将被设置为 OFF。
    如果通信伙伴仍继续发送数据，则在接收缓冲区上溢时会生成一条错误消息。在最后一个帧中接收到的数据将被丢弃。
  - 只要 CPU 提取帧并且接收缓冲区已准备好再次接收数据，RTS 就会被设置为 ON。
  - 如果在发送操作期间 CTS 切换到 OFF，则通信模块会中断发送操作，直到 CTS 复位为 ON。如果 CTS 未在特定的可组态时间内重置为 ON，则会取消发送操作，并输出一条相应的错误消息。

- **硬件 RTS 始终开启**

  "Hardware RTS always ON" 模式对应于 "Hardware RTS always ON, ignore DTR/DSR" 模式。但还需要连接 DTR 和 DSR。

  – 通过参数分配将通信模块设置为“硬件 RTS 始终开启”的操作模式后，会设置 DTR = ON 和 RTS = ON，以向通信伙伴发出其常规就绪状态的信号。

  – 只要在缓冲区上溢前达到所组态的最大帧数或 16 个字符，RTS 就将被设置为 OFF。
  如果通信伙伴仍继续发送数据，则在接收缓冲区上溢时会生成一条错误消息。在最后一个帧中接收到的数据将被丢弃。

  – 只要 CPU 提取帧并且接收缓冲区已准备好再次接收数据，RTS 就会被设置为 ON。

  – 如果在发送操作期间 CTS 切换到 OFF，则通信模块会中断发送操作，直到 CTS 复位为 ON。如果 CTS 未在特定的可组态时间内重置为 ON，则会取消发送操作，并输出一条相应的错误消息。

  – 从 DSR = ON 切换为 DSR = OFF 将取消激活的发送作业并触发错误消息。

伴随信号的自动操作

- **硬件 RTS 始终处于切换状态**

  对于通信模块，“硬件 RTS 始终处于切换状态”按如下方式实现：

  – 通过参数分配将通信模块设置为“硬件 RTS 始终切换”操作模式后，会将线路 RTS 设置为 OFF，将 DTR 设置为 ON（通信模块准备好运行）。

    在将 DSR 线路设置为 ON 后才能发送帧。只要将 DSR 设置为 OFF，就无法通过 RS232C 接口发送数据。取消发送作业并生成一条相应的错误消息。

  – 发送作业未决时，RTS 会设置为 ON，并且启动组态的 RTS 接通延迟。数据输出时间结束后，系统会检查通信伙伴是否已将 CTS 设置为 ON。如果已设置为 ON，则会通过 RS232 接口发送数据。

  – 如果 CTS 线路在 RTS 接通延迟范围内未设置为 ON，或在传输期间 CTS 切换为 OFF，则发送作业会被中止，并生成一条错误消息。

  – 一旦数据发送完毕且超过组态的清除 RTS 关断延迟，RTS 线路将立即设置为 OFF。系统不会等待 CTS 更改为 OFF。

– 始终都可通过 RS232 接口接收数据。如果通信模块的接收缓冲区有溢出的风险，将无响应。

– 从 DSR = ON 切换为 DSR = OFF 将取消激活的发送作业并触发错误消息。

说明

设置"RTS 接通延迟"(RTS ON delay)，使通信伙伴能够在此时间结束之前进入准备好接收状态。

设置"RTS 关断延时"(RTS OFF delay)，使通信伙伴能够在 RTS 设置为 OFF 以及取消发送请求前完整接收帧的最后字符。

说明

组态自动使用 RS232 信号后，将不能通过相应指令控制 RTS 和 DTR！

时序图

下图显示了在已组态数据流控制"硬件 RTS 始终切换"(Hardware RTS always switched)下，发送作业的时间顺序：

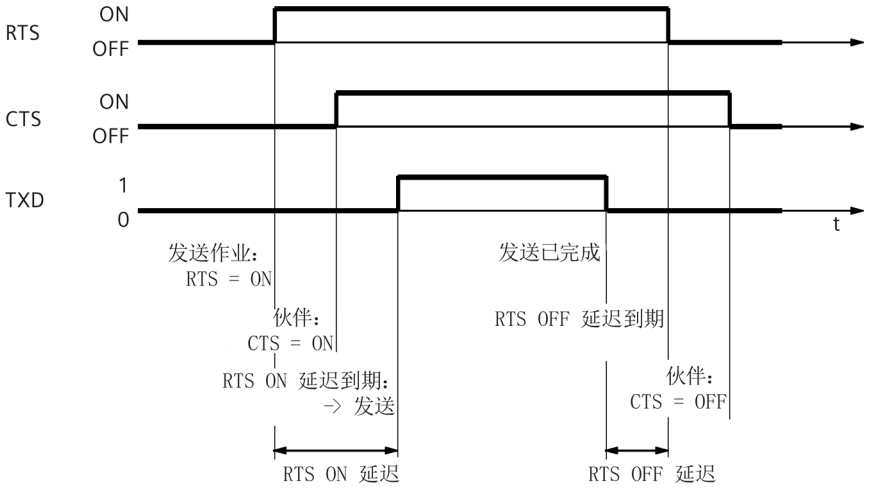


图 3-3 "硬件 RTS 始终切换"(Hardware RTS always switched) 的时序图

更多信息

说明

通信模块通过以下设置接受 DTR/DSR 或 RTS/CTS 操作：

- 硬件 RTS 始终切换，忽略 DTR/DSR
- 硬件 RTS 始终开启
- 硬件 RTS 始终处于切换状态

组态/参数分配

4

4.1 通信模块的组态/参数分配

以下部分包含下列协议及其参数的说明：

- 使用自由口通信 (页 48)
- 使用 3964(R) 通信 (页 58)
- 通过 Modbus RTU 通信 (页 64)
- 使用 USS 通信 (页 73)

执行参数分配以及随后根据所用协议对通信进行编程时需要用到此信息。

组态和参数分配在 STEP 7 (TIA Portal) 的设备视图和通信模块的属性对话框中进行。在运行期间，有些组态也可通过相应的"Config"指令更改（Port_Config、Send_Config、Receive_Config、P3964_Config）。

说明

PROFIBUS DP 的 GSD 文件

有关使用 PROFIBUS DP 的 GSD 文件时的限制信息，请参见相应设备手册的"参数设置"部分。

设置点对点通信的步骤

该过程不依赖于所使用的通信模块。

1. 在 STEP 7 (TIA Portal) 硬件编辑器的设备视图中，组态一个带有 CPU 和通信模块的 S7-1500 结构。
2. 在"属性"(Properties) 选项卡的"常规"(General) 区域中分配通信模块接口的参数（协议、协议参数、地址）。

4.2 关于使用性能优化选项的特殊功能

从通信模块的固件版本 V2.0 起，可使用性能优化选项。如果仅使用多个通信模块发送和接收短帧，则此选项很适用。

以下概述显示了不使用和使用该选项之间的主要区别：

不使用性能优化选项	使用性能优化选项
根据通信模块将报文长度限制为 1、2 或 4 KB	对于接收帧，将报文长度限制为 24 个字节；对于发送帧，则限制为 30 个字节。更长的帧会被拒绝。
传输一份报文需要 CPU 的多个应用周期。周期数随通过数据记录进行通信的通信模块数的增加而增加。	帧的传输需要 CPU 的一个应用周期，并且可并行使用多个通信模块（优化响应时间，改善时序行为）。
地址分配范围为 8 字节输入数据和 0 字节输出数据	地址分配范围为 32 字节输入数据和 32 字节输出数据
自通信模块的固件版本 V1.0 起可用	自通信模块的固件版本 V2.0 起可用
组态和参数分配在 STEP 7 (TIA Portal) 的设备视图和通信模块的"属性"(Properties) 对话框中进行。性能优化选项不能使用"Config"指令 (Port_Config, Send_Config,Receive_Config, P3964_Config) 进行更改。	
自指令库 PtP Communication, USS Communication 和 MODBUS (RTU) 版本 V1.0 起支持	自指令库 PtP Communication 版本 V4.0 以及指令库 USS Communication 和 MODBUS (RTU) 版本 V5.0 起支持

说明

Modbus RTU

在激活性能优化选项的情况下通过 Modbus RTU 进行通信时，传输数据的数量结构存在限制 (页 70)。

4.3 使用自由口通信

4.3.1 与自由口建立串行连接的程序

要求

- 已设置硬件并且存在到链路伙伴的电气连接。
- 已在 STEP 7 (TIA Portal) 中创建项目并且 CPU 已插入到硬件组态中。

步骤 - 硬件组态

1. 将 CM PtP 通信模块插入硬件组态中。
2. 根据链路伙伴设置通信参数：

   例如，传输速度、字符帧、帧开始和帧结束

   每次 CPU 启动时，会将这些参数传送到 CM PtP 通信模块。

步骤 - 编程

1. 创建数据结构，该结构包括要传送的数据。

发送数据

1. 插入来自 PtP 通信库的指令：用于发送数据的 Send_P2P
2. 将指令的输入参数和输出参数互联，例如：
   - PORT 输入处系统变量的 HWID
   - BUFFER 输入处包含要发送数据的数据结构

   注意：运行期间，REQ 输入处的每个上升沿将发送指定的数据区一次。必须调用该块，直到 DONE 指示已将数据传送到模块。

   发生错误时，如果设置 ERROR 一次并在 STATUS 中显示相应的信息，则表示没有传送数据。

接收数据：

1. 插入来自 PtP 通信库的指令：用于接收数据的 Receive_P2P
2. 将指令的输入参数和输出参数互联，例如：
   - PORT 输入处系统变量的 HWID
   - BUFFER 输入处用于存储已接收数据的数据结构

   注意：运行期间 NDR 输出处的高电平表示新数据已接收并存储在指定的数据区。必须调用该块，直到 NDR = TRUE。然后，可以分析接收的数据并且可以再次调用 RECEIVE_P2P。

可选附件

- 可以选择使用以 _Config 结尾的指令以在用户程序运行期间更改硬件组态的参数。不会在硬件组态中保存这些更改。下次重启时会将其覆盖。
- 如果自动操作不是一个合适的选择，指令 Signal_Set 和 Signal_Get 可用于单独地控制 RS232 随附信号。

4.3.2 使用自由口的数据传输

简介

自由口是可自由编程的基于帧的协议，也称为 ASCII 协议。

自由口协议通过通信模块和通信伙伴之间的点对点连接控制数据传输。自由口协议包含物理层（第 1 层）。

自由口协议支持发送和接收任何结构的消息（从 00H 到 FFH（对于 8 个数据位的字符帧）或从 00H 到 7FH（对于 7 个数据位的字符帧）的所有字符）。

必须为发送方向和接收方向组态帧的起始和结束标准。可以组态不同的开始和结束标准。

可将指令用于与通信伙伴之间的通信（请参见 PtP 编程概述）。

4.3.3 使用自由口发送数据

指定发送设置

若发送消息，必须通知通信伙伴消息发送的开始和结束。这些设置可在硬件配置中永久设置，也可使用指令 Send_Config 在运行期间进行调整。可以选择下列选项之一或各选项的组合：

- 在帧开始前发送中断

  可指定在 RTS 接通延迟时间结束后，于每条消息传输开始时发送附加中断。
  “中断”(Break) 的持续时间可以位时间指定。

  如果使用其他机制进行同步，则可取消激活与发送中断的一致性。

- 发送空闲线路

  可指定附加 "Idle Line" 信号是每个消息传输开始时的输出。
  “空闲线路”(Idle Line) 的持续时间可以位时间指定。

  如果使用其他机制进行同步，则可取消激活与发送中断的一致性。

- RTS 接通延迟

  可以组态在实际数据传输开始前、RTS（发送请求）之后必须经过的时间（仅限 RS232）。

- RTS 关断延迟

  可以组态在取消激活 RTS 信号前、传输完成后必须经过的时间（仅限 RS232）。

- 到（包括）结束符时停止发送

  可组态结束符的个数（1 或 2）及其值。

  发送结束符前的所有数据，与所选帧的长度无关。将要发送的数据中必须包括结束符。即时指定的数据长度更长，发送的数据也仅到分隔符为止并包含该分隔符。

- 已添加字符数

  已添加字符数的输入。按组态的长度发送数据。结束符是自动添加的。根据结束符的数量，将向伙伴发送比指令中所指定数量多一到五个的字符。

说明

如果组合“在帧开始前发送中断”(Send break before frame start)、“发送空闲线路”(Send idle line) 和“RTS 接通延迟”(RTS ON delay)，处理顺序将为“RTS 接通延迟”(RTS ON delay)、“在帧开始前发送中断”(Send break before frame start)、“发送空闲线路”(Send idle line)。

4.3.4 使用自由口接收数据

指定消息的开始

对于使用自由口的数据传输，可在多种不同的开始标准中进行选择。开始标准可定义帧的开始时间。一旦符合指示消息开始的标准，就将扫描数据流的消息结束标准。在此选择与发送通信伙伴属性对应的设置。

有两种不同的方法可用于检测消息的开始：

- **以任意字符开始**

  任意字符均可用于定义消息的开始（默认设置）。

  这意味着在通信开始时发送的第一个字符，或检测到帧结束后的第一个字符将被识别为消息的第一个字符。

- **在特殊条件时开始**

  基于以下特定条件检测到消息的开始。

  - **检测到换行符后**

    除非事先接收到中断，否则不会接受帧开始，也就是说，伙伴必须在发送帧之前先发送中断。

  - **检测到空闲线路后**

    经组态的 idle line 持续时间结束后才会接受帧开始。此程序需要两个帧之间的最小间隔。

  - **接收到起始字符后**

    在识别到经组态的 start character 后检测到帧开始。

  - **在检测到一个或多个 start sequence 后**

    在识别到长度达到五个字符的已组态字符串时检测到帧开始。最多可组态 4 个 start sequence。最长为 5 个字符的开始序列也可包含“don't care characters”。

示例：

表格 4- 1　组态开始条件

开始条件	第 1 个字符	第 2 个字符	第 3 个字符	第 4 个字符	第 5 个字符
1	0x68	xx	xx	0x68	xx
2	0x10	0xaa	xx	xx	xx
3	0xdc	0xaa	xx	xx	xx
4	0xe5	xx	xx	xx	xx
:					

已收到如下消息：68 10 aa 68 bb 10 aa 16

开始标准的评估从接收到第一个字符 0x68 开始。
第 2 个和第 3 个字符无约束。
当接收到第 4 个字符（第二个 0x68）时，满足第一个开始条件，将开始下一个消息评估。

指定消息结束

使用自由口协议进行数据传输时，可从多种不同的结束标准中进行选择。结束标准可用于定义完整接收帧的位置。

可组态的结束标准有：

- **按"消息超时"(message timeout) 来识别消息结束**
- **按"响应超时"(response timeout) 来识别消息结束**
- **character delay time 结束后（默认设置）**
- **在接收到固定帧长度后**
- **接收到最大字符数后**
- **读取消息中的消息长度**
- **接收到结束序列后**

消息超时

接收数据时，在用于传输帧的已组态时间结束后检测到帧结束。时间测量从满足开始标准后开始。

响应超时

响应时间用来监视通信伙伴的响应行为。如果在发送作业完成后没有识别到有效的帧开始，则通过相应的消息确认发送作业。

需额外组态实际结束标准。

字符延时时间结束

接收数据时，在超出后续字符间的已组态最长时间（字符延迟时间）时检测到帧结束。该值以位时间为单位。

在这种情况下，必须设置字符延时时间以使其可在两个相邻帧之间结束。不过，该时间应该足够长，以便通信伙伴在一个帧内执行传输暂停时，不会错误地识别该帧已结束。

说明

要实现较高的数据传输速度，建议采用至少 100 个位时间的值。

固定帧长度

接收数据时，在达到已组态帧的长度后识别帧结束。

如果字符延时时间（如果已激活）在达到固定帧长度之前结束，则会输出一条错误消息并丢弃该帧。

如果所接收字符的帧长度与组态的固定帧长度不匹配，则请注意以下情况：

- 在达到组态的固定帧长度之后接收到的所有字符都将被丢弃，直至检测到新的开始标准。
- 如果在达到固定帧长度之前满足另一个（已激活的）结束标准，则会输出一条错误消息并丢弃该帧。

最大字符数

接收数据时，达到所声明的字符数之后识别为帧结束。

此设置可与 "Character delay time" 设置结合使用。如果出现了另一个结束条件，则还认为所接收的帧无错误，而无论是否已达到最大字符数。

如果所接收字符的帧长度与所组态的最大帧长度不匹配，则请注意以下情况：

- 在达到组态的最大字符数后接收到的所有字符都将被丢弃，直至检测到新的开始标准（例如“Idle Line”）。
- 如果在达到组态的最大字符数之前满足不同（已激活）的结束标准，则此“帧组成部分”会被评估为有效帧，而伙伴会等待新的开始标准。在满足新开始标准之前接收到的所有字符都将被丢弃。

说明

如果未激活更多结束标准，则固定帧长度和最大字符数将以相同的方式响应。

消息中的消息长度

接收数据时，如果已接收帧的长度达到已发送的帧长度，则将检测到帧结束。

以下参数可定义用于评估消息长度的字符：

- **消息中长度字段的偏移量**

  在消息中，该值可用于定义将用于确定消息长度的字符的位置。

  可根据缓冲区的大小在 0 到 4095 个字符之间对值进行设置。

- **长度字段的大小**

  该值可用于自将用于确定消息长度的第一个评估位置起指定字符的数量。

  可设置 1、2 和 4 个字符的值。

- **未在长度规范中计数的字符数**

  添加到帧且未对帧长度计数的字符数。该值可用于定义不应包括在消息长度评估中的帧结束时的字节数。

  可设置 0 到 255 个字符的值。

示例：

“消息中的消息长度”(Message length in the message) 的参数分配

消息中长度字段的偏移量：	第 3 个字节（必须将“2”组态为偏移量）
“长度字段的大小”(Size of length field)：	1 个字节
“长度规范中未计字符数”(Number of characters not counted in length specification)：	3 个字节

消息					未在长度规范中计数的字符数		
起始字符	地址	字段长度			校验和	结束符	
字节 1	字节 2	字节 3	字节 ...	字节 X	字节 X+1	字节 X+2	字节 X+3

结束序列

接收数据时，在接收到组态的 end sequence（最多 5 个字符）后识别帧结束。最长为 5 个字符的结束序列也可包含“don't care characters”。CPU 可应用所接收的数据，包括 end sequence。

如果您正在使用 end sequence，则传输为非代码透明的，并且必须排除用户数据中所存在的结束代码。

说明

帧结束序列
如果只有一个结束符，则该条目必须在第 5 行执行。
如果有两个结束符，则这些条目必须在第 4 行和第 5 行执行（无间隙）。
使用其它字符时也是如此。

4.3.5 明码性

明码性

代码透明是指用户数据中可以包含任意字符组合，而无需识别结束标准。

程序的明码性取决于所组态的结束标准和流控制的选择：

- 具有指定结束序列或使用 XON/XOFF 流控制
  - 非代码透明
- 结束标准 character delay time、fixed frame length、maximum frame length、message timeout 或 response timeout 和 message length in the message
  - 代码透明

4.3.6 接收缓冲区

模块的接收缓冲区

通信模块具有接收缓冲区，用于临时存储接收到的帧，直到其被传输到 CPU。该接收缓冲区作为环形缓冲区实现，这意味着帧按照接收顺序传输到 CPU 中，直到接收缓冲区已满。如果缓冲区已满后接收更多帧，则最早接收的帧会被覆盖。如果组态了“禁止覆盖”(Prevent overwriting)，则在接收缓冲区已满时，将生成相应的消息。在接收缓冲区准备接收新帧之前，将拒绝所有其它帧。

分配参数期间，可指定启动期间是否删除接收缓冲区。也可以为缓冲的接收帧数指定值的范围（1 至 255）。

根据所使用的通信模块，模块的接收缓冲区最多可容纳 8 kB（请参见“简介 (页 16)”部分）。帧的最大长度为 4 KB。这意味着每个通信模块都能缓冲至少两帧。

如果您始终要将最后接收到的帧传输到 CPU，则必须为缓冲的帧数分配值 "1"，并取消激活覆盖保护。

说明

如果暂停循环调用 Receive_P2P 一段时间，再次调用 Receive_P2P 可能会导致先接收到来自模块的旧报文，然后才接收到来自 CPU 的最新报文。中断时，旧帧已从通信模块的接收缓冲区传输，准备传输到 CPU。

4.3.7 通过 DMX512 进行通信

可使用 ET 200SP CM PtP（固件版本 V1.0.5 及更高）通信模块通过 DMX512（数字多路复用）进行通信。要通过 DMX512 进行通信，也可使用性能优化选项，但前提是要使用最大值 29_D 作为最高地址。

有关建立 DMX512 连接的更多信息，请参见西门子工业在线支持中常见问题解答的条目 ID 109778975 (https://support.industry.siemens.com/cs/ww/zh/view/109778975)。

4.4 使用 3964(R) 通信

4.4.1 与 3964(R) 建立串行连接的程序

要求

- 已设置硬件并且存在到链路伙伴的电气连接。
- 已在 STEP 7 (TIA Portal) 中创建项目并且 CPU 已插入到硬件组态中。

步骤 - 硬件组态

1. 将 CM PtP 通信模块插入硬件组态中。
2. 根据链路伙伴设置通信参数：

   例如，传输速度、字符帧、帧开始和帧结束

   每次 CPU 启动时，会将这些参数传送到 CM PtP 通信模块。

步骤 - 编程

1. 创建数据结构，该结构包括要传送的数据。

发送数据：

1. 插入来自 PtP 通信库的指令：用于发送数据的 Send_P2P
2. 将指令的输入参数和输出参数互联，例如：
   - PORT 输入处系统变量的 HWID
   - BUFFER 输入处包含要发送数据的数据结构

   注意：运行期间，REQ 输入处的每个上升沿将发送指定的数据区一次。必须调用该块，直到 DONE 指示已将数据传送到模块。

   发生错误时，如果设置 ERROR 一次并在 STATUS 中显示相应的信息，则表示没有传送数据。

接收数据：

1. 插入来自 PtP 通信库的指令：用于接收数据的 Receive_P2P
2. 将指令的输入参数和输出参数互联，例如：
   - PORT 输入处系统变量的 HWID
   - BUFFER 输入处用于存储已接收数据的数据结构

   注意：运行期间 NDR 输出处的高电平表示新数据已接收并存储在指定的数据区。必须调用该块，直到 NDR = TRUE。然后，可以分析接收的数据并且可以再次调用 RECEIVE_P2P。

可选附件

- 可以选择使用以 _Config 结尾的指令以在用户程序运行期间更改硬件组态的参数。不会在硬件组态中保存这些更改。下次重启时会将其覆盖。
- 如果自动操作不是一个合适的选择，指令 Signal_Set 和 Signal_Get 可用于单独地控制 RS232 随附信号。

4.4.2 使用 3964(R) 程序的数据传输

简介

3964(R) 程序可控制通信模块与一个通信伙伴之间的点对点数据交换，并包含物理层（第 1 层）和链路层（第 2 层）。

可将指令用于与通信伙伴之间的通信（请参见 PtP 编程概述）。

4.4.3 控制字符

简介

数据传输期间，程序 3964 (R) 将控制字符添加到原始数据（数据链路层）。通信伙伴可使用这些控制字符检查其是否已完整地接收到所有数据并且未出现任何错误。

3964(R) 程序的控制字符

3964(R) 程序可对下列控制字符进行评估：

STX	Start of Text	要传输的字符串的开始部分	02H
DLE	Data Link Escape	数据传输切换	10H
ETX	End of Text	要传输的字符串的结束部分	03H
NAK	Negative Acknowledge	否定确认	15H
BCC	Block Check Character（仅限 3964R）	块检查字符	

BCC 在通信模块中自动形成并受监视。块检查字符并不是作为帧内容传送到 CPU。

说明

如果将 DLE 字符在帧中作为信息字符传输，则在连接建立和终止期间会发送该字符两次（DLE 副本），以区分于 DLE 控制字符。接收器将恢复 DLE 副本。

优先级

在 3964(R) 程序中，必须为一个通信伙伴分配较高的优先级，为另一个伙伴分配较低的优先级。如果两个伙伴同时开始建立连接，则低优先级的伙伴将取消其发送作业。

4.4.4 块检查字符

块检查字符

使用 3964R 传输协议时，可通过发送附加的块检查字符（BCC = 块检查字符）来增强数据安全性。

块检查字符是已发送或已接收块的偶纵向奇偶校验（所有数据字节的 EXOR 逻辑操作）。其计算开始于连接建立后的第一个用户数据字节（帧的第一个字节），在连接终止时的 DLE ETX 字符后结束。

说明

通过 DLE 副本，DLE 字符被包括在 BBC 计算中两次。

4.4.5 使用 3964(R) 发送数据

为发送建立连接

3964(R) 程序发送 STX 控制字符以建立连接。 如果通信伙伴在 acknowledgment delay time 结束前以 DLE 字符进行响应，则程序将切换至发送模式。

如果通信伙伴以 NAK 或任何其它字符（DLE 或 STX 除外）进行应答，或 acknowledgment delay time 无响应结束，则程序将再次尝试建立连接。 尝试建立连接失败的次数达到组态的次数后，程序将取消连接建立，并将 NAK 字符发送给通信伙伴。 通信模块会输出一条相应的错误消息。

发送数据

如果成功建立了连接，则会将通信模块的输出缓冲区中所包含的用户数据连同所选择的传输参数一起发送给通信伙伴（发送作业期间，用户数据中识别到的 DLE 将被发送两次）。 通信伙伴会监视引入字符间的时间间隔。 两个字符的间隔时间不得超过字符延时时间。 在连接建立后立即开始监视字符延时时间。

如果通信伙伴在激活的发送操作期间发送 NAK 字符，则程序将取消该块，并按上述步骤从建立连接开始重复此块。 如果发送了其它字符，则程序将首先等待字符延时时间结束，然后发送 NAK 字符以将通信伙伴设置为空闲状态。 然后，程序通过 STX 建立连接以重新开始发送数据。

发送期间连接终止

一旦发送了缓冲区中的内容，程序将添加 DLE 和 ETX 字符以及块校验和 BCC（仅限 3964R）作为结束标识符，然后等待确认代码。 如果通信伙伴在 acknowledgment delay time 内发送 DLE 字符，则说明已无错接收数据块。 如果通信伙伴以 NAK、任何其它字符（DLE 除外）或损坏的字符码进行响应，或 acknowledgment delay time 无响应结束，则程序将通过 STX 建立连接以重新开始发送数据。

尝试发送的次数达到组态的次数后，程序将停止该过程，并将 NAK 发送给通信伙伴。 通信模块会输出一条相应的错误消息。

4.4.6 使用 3964(R) 接收数据

为接收建立连接

在空闲状态下，如果没有要处理的发送作业，则程序将等待通信伙伴建立连接。

如果在通过 STX 建立连接期间没有可用的空闲接收缓冲区，则等待时间开始（等待时间 = acknowledgment delay time - 10 ms，但最多为 400 ms）。 如果此时间结束后没有可用的空闲接收缓冲区，则会生成一条错误消息。 此程序将发送 NAK 字符并返回空闲状态。 否则，程序将发送 DLE 并按上述步骤接收数据。

应为两个通信伙伴设置相同的 acknowledgment delay time 值。

如果程序在空闲状态下接收了除 STX 或 NAK 以外的任何字符，则它将等待字符延时时间 (CDT) 结束，然后发送 NAK 字符。 通信模块会输出一条相应的错误消息。

接收数据

成功建立连接后，引入的接收字符将保存在接收缓冲区中。 如果接收到两个连续的 DLE 字符，则只有其中一个保存在接收缓冲区中。

在建立连接以及每个接收字符后，程序会在字符延时时间期间等待下一个字符。 如果字符延时时间结束后还没收到另一个字符，则将 NAK 发送给通信伙伴。 通信模块会输出一条相应的错误消息。 然后将重试。

如果接收过程中发生传输错误（帧错误、奇偶校验错误等），程序将继续接收数据直到连接终止，然后将 NAK 发送给通信伙伴。 然后将重试。 如果尝试传输的次数达到指定的次数后仍无法在不出现错误的情况下接收块，或者通信伙伴没有在 4 秒的块等待时间内开始重试，则程序将取消接收操作。 通信模块将报告第一个受损的传输和最终的取消。

为接收建立连接

如果 3964 程序检测到一个 DLE ETX 字符串，则它将终止接收操作并通过向通信伙伴发送 DLE 来确认已成功接收到块。 接收出错时，会将 NAK 发送给通信伙伴。 然后将重试。

3964R 程序在检测到 DLE ETX BCC 字符串后会终止接收操作。 它将接收到的块检查字符 BCC 与内部计算的纵向奇偶校验加以比较。 如果 BCC 正确并且没有发生其它接收错误，则 3964R 程序将发送 DLE 然后返回空闲状态。 通信模块通知控制系统有新接收数据。

如果 BCC 有故障或发生其它接收错误，则会将 NAK 发送给通信伙伴。 然后将重试。

4.5 通过 Modbus RTU 通信

4.5.1 与 Modbus RTU 建立串行连接的程序

要求

- 已设置硬件并且存在到链路伙伴的电气连接。
- 已在 STEP 7 (TIA Portal) 中创建项目并且 CPU 已插入到硬件组态中。

步骤 - 硬件组态

1. 将 CM PtP 通信模块插入硬件配置中。
2. 选择自由端口/Modbus 协议。

   注：对于 Modbus RTU，CPU 启动期间使用 Modbus_Comm_Load 指令设置大多数通信参数。
3. 根据报文长度，确定是否要激活"针对多短帧情况进行性能优化"(Performance optimized for many short frames) 参数。

步骤 - 编程

1. 创建数据结构，该结构包括要传送的数据。
2. 将 Modbus_Comm_Load 指令集成到通信模块参数分配的循环序列中。
3. 在 PORT 输入处互连系统变量的 HWID。
4. 调用指令，直到在 DONE 输出处显示成功执行。之后不要再次调用该指令，除非您想更改通信参数。

作为 Modbus 主站的操作：

1. 插入 MODBUS (RTU) 库的 Modbus_Master 指令：
2. 将数据结构与要在 DATA_PTR 输入处发送的数据互连。
3. 在 Modbus_Comm_Load 的 MB_DB 输入处互连 Modbus_Master 指令的背景数据块。

   注：运行期间，REQ 输入处的每个上升沿将处理指定的作业一次。必须调用该块，直到 DONE 指示已将数据传送到模块。

   发生错误时，如果设置 ERROR 一次并在 STATUS 中显示相应的信息，则表示没有传送数据。

作为 Modbus 从站的操作：

1. 插入 MODBUS (RTU) 库的 Modbus_Slave 指令。
2. 互连包含 Modbus 保持寄存器的数据结构。
3. 在 MB_ADDR 参数处输入 Modbus 从站地址。
4. 在 Modbus_Comm_Load 的 MB_DB 输入处互连 Modbus_Slave 指令的背景数据块。

   注：运行期间 NDR 输出处的高电平表示新数据已接收并存储在指定的数据区。

4.5.2 modbus 通信概述

Modbus RTU 通信

Modbus RTU（远程终端设备）是用于网络中通信的标准协议，使用电气 RS232 或 RS422/485 连接在网络中的 Modbus 设备间进行串行数据传输。

Modbus RTU 使用主/从站网络，其中整个通信仅由一个主站设备触发，而从站只能响应主站的请求。主站将请求发送到从站地址并且只有该从站地址响应该命令（例外情况：发送给从站地址 0、未被从站确认的广播帧）。

使用的程序是明码、异步半双工的程序。数据传输无须握手。

系统环境中的位置

以下 Modbus 描述与以下通信模块的使用相关：

- CM PtP RS232 HF
- CM PtP RS422/485 HF
- CM PtP (ET 200SP)

耦合功能

利用相应的通信模块和相关指令，可建立远程 Modbus 控制系统与 SIMATIC S7 之间的通信连接。

使用 RTU 格式的 MODBUS 协议进行传输。

功能代码 01、02、03、04、05、06、08、15 和 16 用于作为 Modbus 从站运行的通信模块与主站系统之间的通信（见“功能代码 (页 70)”）。

如果 SIMATIC S7 通信模块作为 Modbus 主站运行，则功能代码 11 和 12 也可用。

SIMATIC S7 用作 Modbus 从站

主站拥有传输的主动权，通信模块用作从站。

无法进行从从站到从站的帧通信。

指令 Modbus_Slave 根据映射规范使数据在 SIMATIC 数据区可用，或者存储这些数据。

SIMATIC S7 用作 Modbus 主站

作为主站，通信模块将初始化传输，随后输出请求帧，然后开始等待用于从站响应帧的组态响应监视时间。如果从站没有响应，主站将根据组态在输出错误消息之前重复此请求。

帧结构

"主站-从站"和/或"从站-主站"数据交换以**从站地址** 开始，然后是**功能代码**。随后传输数据。数据字段的结构取决于使用的功能代码。帧的最后传送的是 CRC 校验码。

ADDRESS	FUNCTION	DATA	CRC-CHECK
字节/字	字节	n 个字节	2 个字节

ADDRESS	Modbus 从站地址 • 标准地址：1 到 247（字节） • 扩展的站地址：1 至 65535（字）
FUNCTION	Modbus 功能代码 (页 70)
DATA	帧数据：与功能代码相关的管理数据和净数据
CRC-CHECK	帧校验和

从站地址

从站地址范围可介于 1 到 247（字节）或 1 到 65535（字）。该地址用于对总线中所定义的从站进行寻址。

广播消息

主站使用从站地址 0 对总线上的所有从站进行寻址。

广播消息仅允许与写功能代码 05、06、15 和 16 相结合。

从站不会对广播消息发出响应帧。

数据域 DATA

数据域 DATA 用于传送功能代码特定数据，例如：

- 字节数、线圈起始地址、寄存器起始地址、线圈数量和寄存器数量等等

有关详细信息，请参见“功能代码 (页 70)”。

CRC 校验

帧的最后是由 2 个字节组成的 CRC 16 校验和。校验和是按如下多项式计算的：
$x^{16} + x^{15} + x^2 + 1$。

先传输低位字节，然后传输高位字节。

帧结束

当在传输 3.5 个字符所需的时间段内（字符延迟时间的 3.5 倍）不传输任何数据时，将识别为帧结束（请参见《Modbus 协议参考指南》）。

因此，此帧结束 TIME_OUT 取决于数据传输速率，并以位时间指示（内部固定编码为 35 位时间；可在指令中额外组态其它位时间）。

收到帧结束 TIME_OUT 后，将对从连接伙伴接收到的 Modbus 消息帧进行评估和正式检查。

异常响应

当在主站的请求帧中检测到错误时，例如：寄存器地址非法，从站将设置响应帧的功能代码的最高值位。

之后将传输一个字节异常代码，说明错误原因。

有关上述参数含义的详细说明，请参见“GOULD MODICON Modbus 协议”（不属于本文档部分）。

异常代码帧

从站中的异常代码帧具有如下形式：

- 例如，从站地址 5，功能代码 5，异常代码 2

设备 EXCEPTION_CODE_xx 的响应帧：

05H	从站地址
85H	功能代码
02H	异常代码 (1...7)
xxH	CRC 校验和"低字节"
xxH	CRC 校验和"高字节"

驱动程序接收到异常代码帧后，当前作业将由于错误而结束。

根据 Modbus 规范定义了下列错误代码：

错误代码	符合 **Modbus** 规范的含义	原因—短描述*
1	Illegal function	功能代码非法
2	Illegal data address	从站具有非法的数据地址
3	Illegal data value	从站具有非法的数据值
4	Failure in associated device	从站出现内部错误
5	Acknowledge	函数已执行
6	Busy, Rejected message	从站尚未准备好接收消息
7	Negative acknowledgement	该函数不能执行。
* 检查从站获取更多详细信息。		

RS232 模式

以下通信模块支持 RS232 模式：

- CM PtP RS232 HF
- CM PtP (ET 200SP)

有关 RS232 模式的详细信息，请参见 RS232 模式 (页 28)一章。

有关硬件数据流控制和伴随信号的自动运行的信息，请参见握手程序 (页 41)一章。

RS422/485 模式

以下通信模块支持 RS422/485 模式：

- CM PtP RS422/485 HF
- CM PtP (ET 200SP)

有关 RS422/485 模式的详细信息，请参见 RS422 模式 (页 33)和 RS485 模式 (页 38)章节。

常见问题解答

有关详细信息，请参见西门子工业在线支持中的以下常见问题解答。

- 条目 ID 68202723 (https://support.industry.siemens.com/cs/ww/en/view/68202723)：使用 Modbus RTU 协议通过 CM PtP 进行主从通信
- 条目 ID 58386780 (https://support.industry.siemens.com/cs/ww/en/view/58386780)：采用 MODBUS RTU 协议实现 SIMATIC S7 站与第三方设备之间的通信需要哪些硬件和软件组件？

4.5.3 功能代码

未使用性能优化选项的功能代码

功能代码定义了消息帧的含义。同样它也定义了消息帧的结构。

通信模块支持以下功能代码：

功能代码	符合 **MODBUS** 规范的功能	范围
01	Read Coil Status	1 到 2000 位/请求
02	Read Input Status	1 到 2000 位/请求
03	Read Holding Registers	1 到 124/125 字/请求（采用扩展站地址时为 124）
04	Read Input Registers	1 到 124/125 字/请求（采用扩展站地址时为 124）
05	Force Single Coil	1 位/请求
06	Preset Single Register	1 字/请求
08 *	Loop Back Test	读取从站状态或复位从站中的事件计数器
11 *	Fetch Communications Event Counter （仅限主站）	—
15	Force Multiple Coils	1 到 1968 位/请求
16	Preset Multiple Registers	1 到 123 字/请求

* 从站通信的诊断信息

MODBUS 功能代码 00 向所有从站发送广播消息（无从站响应）。

使用性能优化选项的功能代码

激活性能优化选项 (页 47)后，所传输数据的组态限值存在以下限制：

功能代码	符合 MODBUS 规范的功能	CM PtP 为 Modbus 主站	CM PtP 为 Modbus 从站
01	Read Coil Status	1 到 168/160 位/请求（采用扩展站地址时为 160）	1 到 216/208 位/请求（采用扩展站地址时为 208）
02	Read Input Status	1 到 168/160 位/请求（采用扩展站地址时为 160）	1 到 216/208 位/请求（采用扩展站地址时为 208）
03	Read Holding Registers	1 到 10 字/请求	1 到 13 字/请求
04	Read Input Registers	1 到 10 字/请求	1 到 13 字/请求
05	Force Single Coil	1 位/请求	1 位/请求
06	Preset Single Register	1 字/请求	1 字/请求
15	Force Multiple Coils	1 到 184/176 位/请求（采用扩展站地址时为 176）	1 到 136/128 位/请求（采用扩展站地址时为 128）
16	Preset Multiple Registers	1 到 11 字/请求	1 到 8 字/请求

MODBUS 功能代码 00 向所有从站发送广播消息（无从站响应）。

将 Modbus 地址分配给 SIMATIC 地址

下表显示了 Modbus 地址到 SIMATIC 地址的分配。

Modbus				S7-1500	
FC [1]	功能	声明	地址区	声明	CPU 地址
01	读取位	输出	1 - 9999	输出的过程映像	A0.0 - A1249.6
02	读取位	输入	10001 - 19999	输入的过程映像	E0.0 - E1249.6
03 [2]	读取字	保持寄存器	40001 - 49999 or 400001 - 465535	DW0 - DW19998 或 DW0 - DW131068	M 地址区取决于 CPU
04	读取字	输入	30001 - 39999	输入的过程映像	EW0 - EW19996
05 [2]	写入位	输出	1 - 9999	输出的过程映像	A0.0 - A1248.7

Modbus				S7-1500	
FC [1]	**功能**	**声明**	**地址区**	**声明**	**CPU 地址**
06	写入字	保持寄存器	40001 - 49999 or 400001 - 465535	DW0 - DW19998 或 DW0 - DW131068	M 地址区取决于 CPU
15	写入位	输出	1 - 9999	输出的过程映像	Q0.0 - Q1249.6
16 [2]	写入字	保持寄存器	40001 - 49999 or 400001 - 465535	DW0 - DW19998 或 DW0 - DW131068	M 地址区取决于 CPU

1) FC = 功能代码

2) HR_Start_Offset 的值决定了数据区或位存储器地址区是否可通过 SIMATIC CPU 中的 FC 03、05 和 16 进行寻址。

4.6 使用 USS 通信

4.6.1 与 USS 建立串行连接的操作过程

要求

- 已设置硬件并且存在到链路伙伴的电气连接。
- 已在 STEP 7 (TIA Portal) 中创建项目并且 CPU 已插入到硬件组态中。

步骤 - 硬件组态

1. 将 CM PtP 通信模块插入硬件组态中。
2. 选择自由口协议并设置通信参数。

   注意：通过指令实现 USS 功能。
3. 根据报文长度，确定是否要激活"针对多短帧情况进行性能优化"(Performance optimized for many short frames) 参数。

步骤 - 编程

1. 插入来自 USS 通信库的 USS_Port_Scan 指令。
2. 在 PORT 输入处互连系统变量的 HWID。
3. 插入来自 USS 通信库的 USS_Drive_Control 指令。
4. 将 USS_Drive_Control 指令的背景数据块中的 USS_DB 数据结构互连到 USS_Port_Scan 指令的 USS_DB。数据结构包含要传送的所有驱动器的数据。
5. 针对要通过 USS 接口连接的每个附加轴，插入 USS_Drive_Control 指令的附加调用。

   每次都使用相同背景数据块。借助在 USS_Drive_Control 指令的 DRIVE 输入处指定的 USS 地址，会产生区别。这意味着您可在每个驱动器的相应调用的参数处访问控制和反馈数据。

4.6.2 USS 通信概述

系统环境中的位置

下列 USS 描述指的是相应通信模块的使用。

- CM PtP RS232 BA
- CM PtP RS422/485 BA
- CM PtP RS232 HF
- CM PtP RS422/485 HF
- CM PtP (ET 200SP)

简介

USS® 协议（通用串行接口协议）是一种简单的串行数据传输协议，旨在满足变频器技术的要求。

USS 协议定义了一种访问方法，该方法基于主站-从站原理，通过串行总线进行通信。总线可以连接一个主站和最多 16 个变频器（从站）。主站使用消息帧中的地址字符来选择各个变频器。只有通过主站启动的变频器才能发送消息。因此，各个变频器之间无法直接传输数据。以半双工模式进行通信。无法传输此主站功能。

变频器技术需要具体的响应时间以进行任务控制和严格的循环帧通信：

主站连续向变频器发送帧（作业帧），并等待接收来自每个寻址的变频器的响应帧。

如果变频器收到帧而没有出错，

- 并且该帧中寻址的就是此变频器，
- 则变频器必须发送响应帧。

如果不满足上述条件或是在广播中寻址变频器，则变频器不会发送响应帧。

如果在指定的处理时间（响应延迟时间）内收到变频器发送的响应帧，则主站会与相应的这个变频器连接。

帧结构

每个帧均以一个起始字符 (STX) 开头，后面依次为长度规范 (LGE)和地址字节 (ADR)。然后是数据域。帧以块校验字符 (BCC) 结束。帧长度包括用户数据（数量 n）、地址字节 (ADR) 和数据校验符 (BCC)。

STX	LGE	ADR	1	2	...	N	BCC

对于单字（16 位）数据，首先发送高字节，然后发送低字节。相应地，对于双字数据，采取同样的发送方式。帧长度以字节为单位。

数据加密

此数据按如下方式进行加密：

- STX：1 个字节，文本开头，02H
- LGE：1 个字节，包含以二进制数形式表示的帧长度
- ADR：1 个字节，包含以二进制代码形式表示的从站地址和帧类型
- 数据域：每个域一个字节，内容取决于作业
- BCC：1 个字节，块校验字符

数据传输步骤

该主站可确保在帧中进行循环数据传输。该主站使用作业帧对所有从站设备逐个进行寻址。被寻址的节点通过一个响应帧进行响应。接收作业帧之后，从站必须按照主站-从站过程向主站发送响应帧。只有这样，主站才能对下一个从站进行寻址。

帧中的数据域

数据域分为两个区域：参数区 (PKW) 和过程数据区 (PZD)。

STX	LGE	ADR	参数 (PKW)	过程数据 (PZD)	BCC

- **参数区** (PKW)

  PKW 区域处理两个通信伙伴（例如控制器和变频器）之间的参数传输。例如，这包括读取和写入参数值以及读取参数说明和关联的文本。PKW 接口通常包含操作和显示、维护和诊断作业。

- **过程数据区** (PZD)

  PZD 区包含自动化操作所需的信号：

  – 控制字和设定值（从主站到从站）

  – 状态字和实际值（从从站到主站）

  参数区和过程数据区的内容由从站变频器进行定义。

  有关这方面的其它信息，请参考变频器文档。

4.6.3 功能概述

传输顺序

此类指令为最多 16 个变频器从站循环处理数据传输。一次只能为一个变频器激活一个作业。

性能特性：

- 根据总线组态为通信创建数据存储区
- 执行和监视 PKW 作业
- 监视整个系统和故障排除
- 与 CPU 进行通信
- 访问变频器功能
- 读取变频器参数
- 写入变频器参数

编程 - 使用指令进行通信

5

5.1 点对点编程概述

使用自由口或 3964(R) 通信进行数据交换

必须在相关 CPU 的用户程序中以数据块的形式提供传输的数据。通信模块中有一个接收缓冲区用于接收数据。在数据块中设置相应数据块。

在 CPU 用户程序中，以下指令在 CPU 和通信模块之间传输数据：

- Send_P2P
- Receive_P2P

接收缓冲区可以通过 Receive_Reset 指令删除。

通过用户程序的动态组态

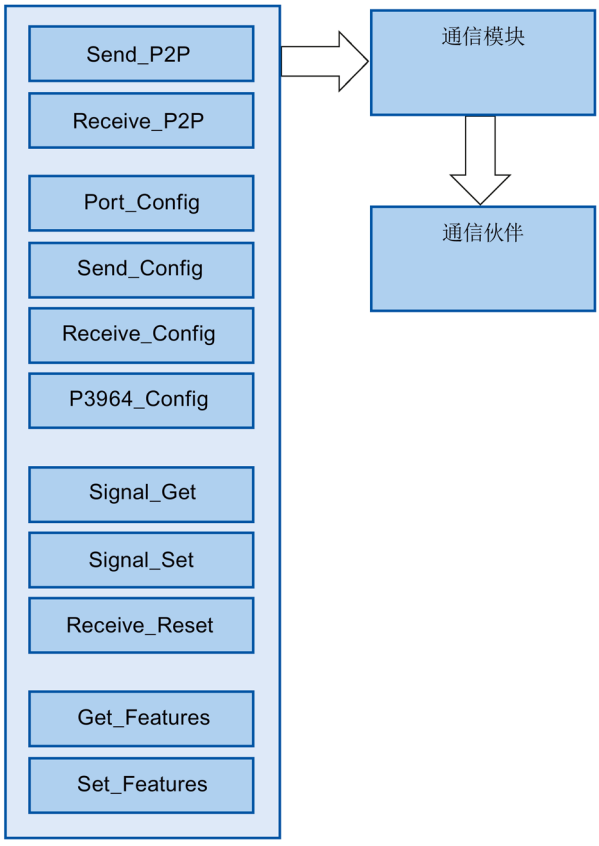
作为“通信模块的组态/参数分配 (页 46)”部分中描述的通信模块接口参数分配的替代或补充，在某些应用领域中，可能建议动态建立通信，即由特定应用程序进行程序控制。

还可使用下列指令之一在运行期间更改在通信模块的属性对话框中执行的所有参数分配：

- Port_Config
- Send_Config
- Receive_Config
- P3964_Config

点对点通信的程序调用 - 顺序

下图显示了用于用户程序与通信伙伴之间通信的点对点指令的功能。



PtP 指令

应用	指令	说明
CPU、通信模块与通信伙伴之间的数据交换（通信）	Send_P2P (页 110)	指令 Send_P2P（发送点对点数据）可用于向通信伙伴发送数据。 调用指令 Send_P2P 以通过自由口协议发送数据。在指令的输出参数中接收到相应确认前，您必须循环调用该指令。 注：在 XON/XOFF 数据流控制的参数分配期间，用户数据不可包含任何已组态的 XON 或 XOFF 字符。默认设置为 DC1 = 11H (XON) 和 DC3 = 13H (XOFF)。
	Receive_P2P (页 114)	指令 Receive_P2P（接收点对点数据）可用于获取来自通信伙伴的通信模块中接收到的消息。 循环调用 Receive_P2P 指令以通过自由口协议接收数据。如果新接收的数据可用，指令将在 NDR 参数中加以指示。 为了表示消息传输的开始和结束，需要在识别消息的开始和结束的自由口协议中定义标准。
检测接收缓冲区	Receive_Reset (页 117)	指令 Receive_Reset（删除接收缓冲区）允许清除通信模块的接收缓冲区。
接口或端口的动态参数分配（可选）	Port_Config (页 94)	您可以使用 Port_Config 指令（端口组态）来组态基本接口参数，如数据传输率、奇偶校验和数据流控制（通过用户程序动态执行）。
	Send_Config (页 98)	根据指令 Send_Config（发送组态），可为点对点通信接口动态组态串行发送参数，例如 RTS ON 延迟/RTS OFF 延迟。
	Receive _Config (页 100)	指令 Receive_Config（接收参数分配）允许将串行接收参数动态分配给通信模块。 该指令可参数化表示接收消息开始和结束的条件。
	P3964_Config (页 107)	指令 P3964_Config（组态协议）可用于动态组态程序 3964(R) 的协议参数，例如字符延迟时间、优先级和块检查（使用程序）。
操作 RS232 伴随信号	Signal_Get (页 118)	通过 Signal_Get 指令（获取 RS232 信号），您可以读取 RS232 信号的当前状态。
	Signal_Set (页 119)	通过 Signal_Set 指令（获取 RS232 信号），可以设置 RS232 信号 DTR 和 RTS 的状态。

应用	指令	说明
启用 Modbus CRC 支持和诊断中断	Get_Features (页 122)	可使用 Get_Features 指令（获取扩展功能）获取有关 Modbus 支持和有关生成诊断报警的信息。
	Set_Features (页 124)	根据 Set_Features 指令（设置扩展功能），如果模块支持，则可激活诊断中断的生成。

自由口或 3964(R) 通信的设置步骤

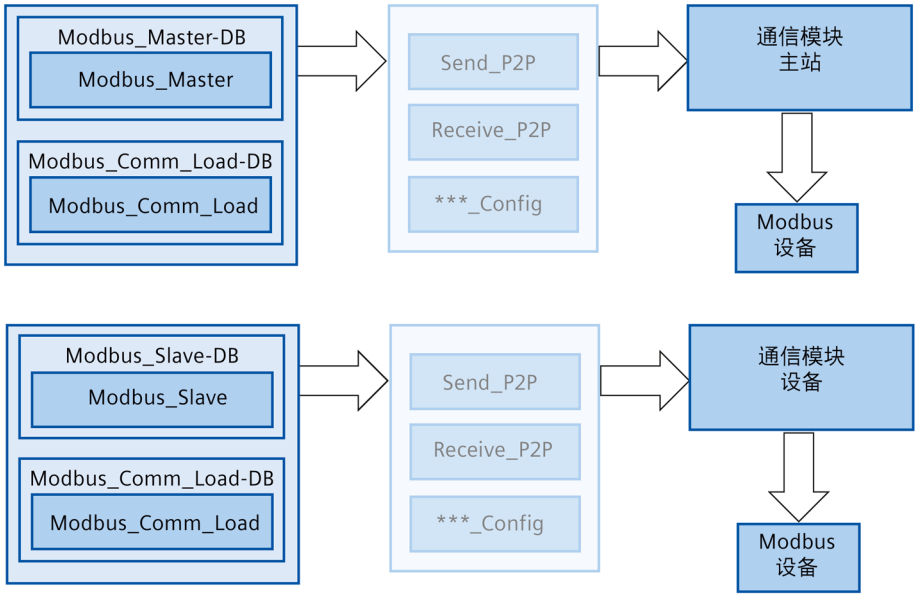
要求：在设备视图和通信模块的属性对话框中，CPU 和通信模块的组态和参数分配已完成。

1. 在 CPU 的项目导航中，选择文件夹"程序块"(Program blocks)，然后双击打开文件夹中的 Main (OB1)。程序编辑器随即打开。
2. 从"指令"(Instructions) 任务卡的"通信"(Communication) 区域中选择指令 Send_P2P 和 Receive_P2P 并将它们拖放到 Main (OB1) 的网络中。
3. 按照规范组态指令。
4. 将硬件组态和用户程序下载到 CPU 中。

5.2 Modbus 编程概述

Modbus 通信的程序调用 - 顺序

下图所示是用户程序和 Modbus 设备之间通信的 Modbus 指令的功能。（下游使用 Send_P2P、Receive_P2P 和 Config 指令）。



Modbus 指令

应用	指令	说明
在用户程序和 Modbus 设备之间进行数据交换（通信）	Modbus_Master (页 147)	Modbus_Master 指令允许通过 PtP 端口作为 Modbus 主站进行通信。 利用 Modbus_Master 指令，CPU 可用作 Modbus RTU 主站设备，与一个或多个 Modbus 从站设备进行通信。
	Modbus_Slave (页 155)	Modbus_Slave 指令允许通过 PtP 端口作为 Modbus 从站进行通信。 利用 the Modbus_Slave 指令，CPU 可用作 Modbus RTU 从站设备，与一个 Modbus 主站设备进行通信。

应用	指令	说明
接口和协议的参数分配（可选）	Modbus_Comm_Load (页 141)	指令 Modbus_Comm_Load 允许组态 Modbus RTU 的通信模块端口。 必须运行 Modbus_Comm_Load 来设置 PtP 端口参数，例如：数据传输率、奇偶校验和流控制。为 Modbus RTU 协议组态完接口后，它只能由 Modbus_Master 或 Modbus_Slave 指令使用。

说明

交替使用 Modbus_Slave 和 Modbus_Master

通信模块既可充当主站也可充当从站。

设置 Modbus 通信的步骤

要求：通信模块的设备视图以及属性对话框中 CPU 和通信模块的组态和参数分配均已完成。

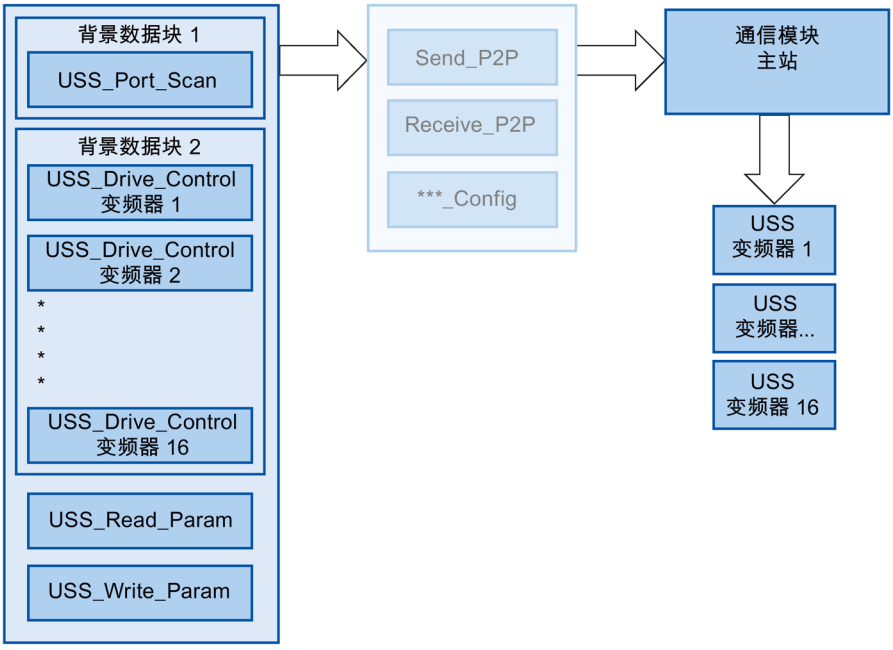
1. 在 CPU 的项目导航中，选择文件夹"程序块"(Program blocks)，然后双击打开文件夹中的 Main (OB1)。程序编辑器随即打开。
2. 根据您的任务，从"指令"(Instructions) 任务卡的"通信"(Communication) 区域中为 Modbus 通信选择相应指令并将它们拖放到 Main (OB1) 的网络中：
   - 指令 Modbus_Comm_Load 可为 Modbus 通信组态通信模块的端口。

     报告 DONE（或 ERROR）前，必须在 Main (OB1) 中调用 Modbus_Comm_Load。
   - Modbus_Master 指令用于 Modbus 主站功能。
   - Modbus_Slave 指令用于 Modbus 从站功能。
3. 按照规范组态指令。
4. 将硬件组态和用户程序下载到 CPU 中。

5.3 USS 编程概述

程序要求 USS 通信 - 顺序

下图显示用户程序和 USS 变频器之间通信的 USS 指令的功能。（下游需要使用指令 Send_P2P、Receive_P2P 和 Config 指令）。



USS 指令

应用	指令	说明
CPU、通信模块和 USS 驱动器之间的数据通信	USS_Port_Scan (页 186)	USS_Port_Scan 指令允许使用 USS 程序段（必须循环调用）通过通信模块与最多 16 个驱动器进行通信。 USS_Port_Scan 指令通过 PtP 通信端口控制 CPU 和变频器之间的通信。每次调用此功能时，将进行与变频器之间的通信。需要执行一次指令 USS_Port_Scan： 由于大多数驱动器具有可组态的内部功能，可根据超时监视通信的完整性，因此应从时间控制的 OB 调用 USS_Port_Scan 指令。
与 USS 变频器进行数据交换	USS_Drive _Control (页 190)	USS_Drive_Control 指令允许为变频器准备发送数据并显示接收数据。 指令的输入和输出与变频器的状态和操作功能相对应。每个变频器必须调用一次 USS_Drive_Control 指令。对于面向一个 USS 程序段的所有 USS_Drive_Control 指令调用，只需要 USS_Port_Scan 的公共背景 DB。针对 USS 网络，将指令 USS_Drive_Control 的所有调用与同一背景数据块互联。 应从主程序的循环 Main (OB1) 中调用 USS_Drive_Control 指令。
读取或修改 USS 驱动器中的参数	USS_Read_Param (页 195)	USS_Read_Param 指令允许从变频器中读取参数。 使用 USS_Read_Param 指令读取控制变频器内部功能的变频器操作参数。 应从主程序的循环 Main (OB1) 中调用 USS_Read_Param 指令。
	USS_Write_Param (页 197)	USS_Write_Param 指令允许更改变频器中的参数。 USS_Write_Param 指令应从主程序的循环主程序 (OB1) 中调用。

设置 USS 通信的步骤

要求：在设备视图和通信模块的属性对话框中，CPU 和通信模块的组态和参数分配已完成。

1. 在 CPU 的项目树中，选择"程序块"(Program blocks) 文件夹，然后双击打开所需的时间控制的 OB。程序编辑器随即打开。
2. 从"指令"(Instructions) 任务卡的"通信"(Communication) 区域中选择指令 USS_Port_Scan 并将其拖放到时间控制 OB 的网络中。

   USS_Port_Scan 指令允许通过 USS 网络通信。
3. 在 CPU 的项目导航中，选择文件夹"程序块"(Program blocks)，然后双击打开文件夹中的 Main (OB1)。程序编辑器随即打开。
4. 根据您的任务，从"指令"(Instructions) 任务卡的"通信"(Communication) 区域中为 USS 通信选择相应指令并将它们拖放到 Main (OB1) 的网络中：
   - USS_Drive_Control 指令用于与变频器进行数据交换。
   - USS_Read_Param 指令用于从变频器中读取参数。
   - USS_Write_Param 指令用于更改变频器中的参数。
5. 按照规范组态指令。
6. 将硬件组态和用户程序下载到 CPU 中。

5.4 指令

5.4.1 点对点

5.4.1.1 自由口通信概述

STEP 7 提供扩展指令，可用于通过用户程序中指定的协议进行自由口通信。这些指令可分为两类：

- 组态指令
- 通信指令

数据通信

可在硬件配置中使用模块的"针对许多短报文进行性能优化"参数来定义数据交换的类型。只要不超过传入/传出报文的最大长度，建议使用性能优化选项。

CPU 与通信模块之间的数据交换有两种类型：

- 异步数据交换

  自由端口指令通过读取或写入数据集与通信模块异步通信（相对于应用周期）。数据传输需要经过多个应用周期。最大报文长度需符合模块技术规范。

- 同步数据交换（性能优化选项 (页 47)）

  自由端口指令通过通信模块的 IO 数据基于应用周期与通信模块同步进行通信。

  传入报文的最大长度为 24 个字节，传出报文的最大长度为 30 个字节。通过基于应用周期同步使用数据，响应时间得到优化，尤其是在并行使用多个 CM PtP 时。

说明

性能优化选项适用于 V4.0 及更高版本的 PtP Communication 指令库。

组态指令

在用户程序启动自由端口通信之前，必须组态通信接口以及用于发送和接收数据的参数。

可以通过用户程序中的以下指令或在设备组态中为每个 CM 设置接口组态和数据组态：

- Port_Config (页 94)
- Send_Config (页 98)
- Receive_Config (页 100)
- P3964_Config (页 107)

注意
设备组态 <-> 组态指令 CPU 每次 Power On（恢复电压）时，都会向 CM 传输设备组态参数。 根据用户程序中的定义，向 CM 传输组态指令参数。 设备组态参数与组态指令参数不同步，也就是说，组态指令参数不适用于 CPU 设备组态。 在用户程序中可确定 CM 何时采用哪些参数。

通信指令

用户程序使用自由端口通信指令向通信接口发送数据和从通信接口接收数据。CM 向通信站发送数据，并从中接收数据。

- Send_P2P (页 110)
- Receive_P2P (页 114)

说明

数据一致性

- 如果要保持发送数据的一致性，那么只有在 Send_P2P 指令设置了 DONE 后，才可在 REQ 参数的上升沿对其进行更改。
- 如果要保持读取数据的一致性，那么只有在 NDR = TRUE 的循环中才可对其赋值。

可用附加指令重置接收缓冲区，可查询并设置特殊 RS232 信号。

- Receive_Reset (页 117)
- Signal_Get (页 118)
- Signal_Set (页 119)

只要模块支持，就可使用以下指令读取或写入扩展功能。

- Get_Features (页 122)
- Set_Features (页 124)

所有自由口指令异步工作。因此，必须在 DONE 或 NDR 输出参数表示执行已完成后，才可调用该指令。

用户程序能够通过查询架构决定发送和接收状态，Send_P2P 和 Receive_P2P 能够同时运行。通信模块根据需要缓冲发送和接收数据，直至达到模块特定的最大缓冲区大小。

说明

位时间的精度

以组态的数据传输速率为不同的参数指定位时间数。以位时间指定参数后，参数将与数据传输速率无关。可使用最大为 65535 的数值指定所有以位时间为单位的参数。

5.4.1.2 使用指令

必须循环调用自由口指令来查询发送过程中接收到的数据或传送的结束。

根据数据量以及是否激活性能优化选项，数据传输可通过多次调用（应用周期）进行。如果作业执行时 DONE = TRUE 或 NDR = TRUE，则作业执行无错误。

说明

备份 STATUS

DONE、NDR、ERROR 和 STATUS 参数只可用于一个块循环。要显示 STATUS，应将 STATUS 复制到空闲数据区。

主站

主站的典型轮询顺序：

1. Send_P2P 指令触发到 CM 的传输。
   数据传输将通过 REQ 输入的上升沿触发。
2. Send_P2P 指令在后续周期中执行，以查询传输过程的状态。
3. 当 Send_P2P 指令发出传输在 DONE 输出处结束的信号时，用户代码可准备接收应答。
4. 反复运行 Receive_P2P 指令以查询应答。如果 CM 获得了响应数据，则 Receive_P2P 指令会将此响应复制给 CPU 并表示 NDR 输出已收到新数据。
5. 用户程序可处理响应。
6. 回到第 1 步并重复上述顺序。

从站

从站的典型轮询顺序：

1. 用户程序在每个周期中运行 Receive_P2P 指令。
2. 如果 CM 已收到请求，Receive_P2P 将发出该新数据在 NDR 输出中可用的信号，并将该请求复制到 CPU。
3. 用户程序处理请求并创建响应。
4. 响应通过 Send_P2P 指令返回至主站。
5. 必须反复运行 Send_P2P 指令以确保发送过程实际正在进行。
6. 回到第 1 步并重复上述顺序。

从站必须确保 Receive_P2P 充足的调用频率，这样主站才能在因等待响应超时需取消进程前接收传输。为此，可从循环 OB 中调用用户程序 Receive_P2P，该 OB 需有足够短的周期时间，以便主站在超时设置结束前可以接收传输。

5.4.1.3 用于自由口操作的通用参数

表格 5- 1 自由口指令的通用输入参数

参数	说明
REQ	数据传输将通过 REQ 输入的上升沿触发。只有在命令执行完毕（DONE 或 ERROR）后，才可能在 REQ 上再生成一个上升沿。数据传输可能会进行多次调用（程序周期），具体取决于数据量。 向程序添加自由口指令时，STEP 7 将提示用户指定背景数据块（或令 STEP 7 创建相应的背景数据块）。对每个 PtP 指令调用使用唯一的 DB。
PORT	在组态通信模块期间分配端口地址。PORT 参数将特定通信模块的分配传达给指令。 组态后可以为标准端口选择一个符号名称。已分配的 CM 端口值为 S7-1200/1500 中设备组态以及 S7-300/400 中"输入地址"(Input address) 的"硬件 ID"(Hardware ID) 属性。符号端口名称在符号表中指定。

自由口指令的输出参数 DONE、NDR、ERROR 和 STATUS 指示自由口功能的执行状态。

表格 5- 2 输出参数 DONE、NDR、ERROR 和 STATUS

参数	数据类型	标准	说明
DONE	Bool	FALSE	设置为 TRUE 并保持一个周期，表明上一请求已经完成且有错误；否则为 FALSE。
UNIVERSAL [1]	Bool	FALSE	在 CPU 和通过 PORT 指定的 CM 之间进行数据通信的类型： FALSE：性能优化选项（同步）（页 47） • 接收帧最多 24 个字节 • 发送帧最多 30 个字节 TRUE Universal（异步） • 根据 CM 将帧长度限制为 1、2 或 4 KB
NDR	Bool	FALSE	设置为 TRUE 并保持一个周期，表示已接收到新数据；否则为 FALSE。

参数	数据类型	标准	说明
ERROR	Bool	FALSE	设置为 TRUE 并保持一个周期，表示上一请求已完成但有错误；可在 STATUS 中找到相应的错误代码；否则为 FALSE。
STATUS	Word	16#0000 或 16#7000	结果状态： • 如果位 DONE 或 NDR 置位，则 STATUS 将设置为 0/16#7000 或一个特定的状态代码。 • 如果位 ERROR 置位，则 STATUS 将显示一个错误代码。 • 如果未设置上述任何位，该指令将返回描述功能当前状态的状态结果。 再次调用（使用同一个端口地址调用）该指令之前，STATUS 中的值始终有效。

[1] 自库版本 V4.0 起可用

表格 5- 3 输入/输出参数 COM_RST

参数	数据类型	标准	说明
COM_RST	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。COM_RST 然后设置回 FALSE。 注： COM_RST 必须在 CPU 启动时设置为 TRUE，之后不能再触发。初始化背景数据块后，指令将复位 COM_RST。

说明

请注意，参数 DONE、NDR、ERROR 和 STATUS 的设置仅对一个周期有效。

表格 5- 4　共享错误代码

错误代码	说明
16#0000	无错误
16#7000	功能未激活
16#7001	请求后启动了初始调用。
16#7002	请求后启动了后续调用。
16#8x3A	参数 x 中的指针无效

表格 5- 5　STATUS 参数的共享错误类别

类别说明	错误类别	说明
端口组态	16#81Ax	针对接口组态中常见错误的说明
发送组态	16#81Bx	针对发送组态中错误的说明
接收组态	16#81Cx	针对接收组态中错误的说明
发送	16#81Dx	针对发送期间运行时错误的说明
接收	16#81Ex	针对接收期间运行时错误的说明
RS232 伴随信号	16#81Fx	针对信号处理相关错误的说明

5.4.1.4 Port_Config：组态 PtP 通信端口

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

通过 Port_Config 指令（端口组态），可使用程序更改运行期间数据传输速率等参数。CM 中未决的数据将在执行 Port_Config 时删除。

Port_Config 的组态更改将保存在 CM 中，而不是 CPU 中。恢复电压时，将使用保存在设备配置中的数据对 CM 进行组态。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	在此输入的上升沿开始向 CM 传输数据。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
PRO-TOCOL	IN	UInt	Word	0	协议 • 0 = 自由口协议 • 1 = 协议 3964(R)

参数	声明	数据类型		默认值	说明
BAUD	IN	UInt	Word	6	端口数据传输速率： • 1 = 300 bps • 2 = 600 bps • 3 = 1200 bps • 4 = 2400 bps • 5 = 4800 bps • 6 = 9600 bps • 7 = 19200 bps • 8 = 38400 bps • 9 = 57600 bps • 10 = 76800 bps • 11 = 115200 bps • 12 = 250000 位/s
PARITY	IN	UInt	Word	1	端口的奇偶校验： • 1 = 无奇偶校验 • 2 = 偶校验 • 3 = 奇校验 • 4 = 传号校验 • 5 = 空号校验 • 6 = 任意
DATABITS	IN	UInt	Word	1	每个字符的位数： • 1 = 8 个数据位 • 2 = 7 个数据位
STOPBITS	IN	UInt	Word	1	停止位： • 1 = 1 个停止位 • 2 = 2 个停止位

参数	声明	数据类型		默认值	说明
FLOW-CTRL	IN	UInt	Word	1	流控制： • 1 = 无流控制 • 2 = XON/XOFF • 3 = 硬件 RTS 始终开启 • 4 = 硬件 RTS 已开启 • 5 = 硬件 RTS 始终开启，忽略 DTR/DSR
XON-CHAR	IN	Char		16#0011	指定用作 XON 字符的字符。通常为 DC1 字符 (11H)。仅当软件流控制处于活动状态时才评估此参数。
XOFF-CHAR	IN	Char		16#0013	指定用作 XOFF 字符的字符。通常为 DC3 字符 (13H)。仅当软件流控制处于活动状态时才评估此参数。
WAITIME	IN	UInt	Word	2000	指定 XON 字符在收到 XOFF 字符后等待的时间，或者 CTS = OFF 后 CTS = ON 信号等待的时间（0 到 65535 ms）。仅当流控制处于活动状态时才评估此参数。
MODE	IN	USInt	Byte	0	工作模式 有效的工作模式包括： • 0 = 全双工 (RS232) • 1 = 全双工 (RS422) 四线制模式（点对点） • 2 = 全双工 (RS 422) 四线制模式（多点主站；CM PtP (ET 200SP)) • 3 = 全双工 (RS 422) 四线制模式（多点从站；CM PtP (ET 200SP)) • 4 = 半双工 (RS485) 二线制模式 [1)]

参数	声明	数据类型		默认值	说明
LINE_PRE	IN	USInt	Byte	0	接收线路初始状态 有效的初始状态是： • 0 =“无”初始状态 [1)] • 1 = 信号 R(A)=5 V，信号 R(B)=0 V（断路检测）： 在此初始状态下，可进行断路检测。 仅可以选择以下项：“全双工 (RS422) 四线制模式（点对点连接）”和“全双工 (RS422) 四线制模式（多点从站）”。 • 2 = 信号 R(A)=0 V，信号 R(B)=5 V： 此默认设置对应于空闲状态（无激活的发送操作）。在此初始状态下，无法进行断路检测。
BRK_DET	IN	USInt	Byte	0	断路检测 允许以下设置： • 0 = 断路检测已禁用 • 1 = 断路检测已激活
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
DONE	OUT	Bool		FALSE	如果上一个请求无错完成，将变为 TRUE 并保持一个周期
ERROR	OUT	Bool		FALSE	如果上一个请求有错完成，将变为 TRUE 并保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

[1)] 使用 PROFIBUS 电缆连接 CM 1241 的 RS485 时所需的设置

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

5.4.1.5 Send_Config： 组态 PtP 发送方

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Send_Config 指令（发送组态）允许用户使用程序在运行时更改发送参数（确定待发送数据开始和结束的条件）。CM 中未决的任何数据将在执行 Send_Config 时删除。

Send_Config 的组态更改将保存在 CM 中，而不是 CPU 中。CPU 或通信模块的电压恢复后，设备组态中保存的参数会立即恢复。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	激活在此输入上出现上升沿时更改组态。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
RTSONDLY	IN	UInt	Word	0	从激活 RTS 后到开始传输发送数据之前等待的毫秒数。仅当硬件流控制处于活动状态时此参数才有效。有效范围是 0 至 65535 ms。值为 0 将取消激活该功能。

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
RTSOFFDLY	IN	UInt	Word	0	传输发送数据之后到 RTS 取消激活之前所等待的毫秒数：仅当硬件流控制处于活动状态时此参数才有效。有效范围是 0 至 65535 ms。值为 0 将取消激活该功能。
BREAK	IN	UInt	Word	0	此参数指定，在每帧开始时，将在指定数量的位时间内发送中断。最大值为 65535 位时间。值为 0 将取消激活该功能。
IDLELINE	IN	UInt	Word	0	该参数指定，在每帧开始前，线路将在指定数量的位时间内保持空闲状态。最大值为 65535 位时间。值为 0 将取消激活该功能。
USR_END	IN	STRING[2]		0	输入结束符。 最多可组态 2 个结束符。 发送时包括文本结束字符。
APP_END	IN	STRING[5]		0	输入要添加的字符。 最多可添加 5 个字符。
COM_RST	IN /OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
DONE	OUT	Bool		FALSE	如果上一个请求无错完成，将变为 TRUE 并保持一个周期
ERROR	OUT	Bool		FALSE	如果上一个请求有错完成，将变为 TRUE 并保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

5.4.1.6 Receive_Config： 组态 PtP 接收方

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Receive_Config 指令（接收组态）允许在运行期间更改接收参数（使用程序）。该指令可组态标记所接收数据的开始和结束的条件。CM 中未决的任何数据将在执行 Receive_Config 时删除。

Receive_Config 的组态更改以**非保持性**的方式保存到 CM 上。CPU 或通信模块的电压恢复后，设备组态中保存的参数会立即恢复。因此，CPU 或通信模块的电压恢复后，必须从用户程序中再次调用 Receive_Config 指令，以便覆盖存储在设备组态中的参数。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	激活在此输入上出现上升沿时更改组态。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
RECEIVE_CON-DITIONS	IN	Variant	Any	-	Receive_Conditions 的数据结构指定用于识别帧的开始和结束条件。

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
DONE	OUT	Bool		FALSE	如果上一个请求无错完成，将变为 TRUE 并保持一个周期
ERROR	OUT	Bool		FALSE	如果上一个请求有错完成，将变为 TRUE 并保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

Receive_P2P 指令的开始条件

Receive_P2P 指令使用设备组态中指定的组态或通过 Receive_Config 指令确定自由口通信帧的开始和结束。帧的开始由开始条件定义。可以使用一个或多个开始条件确定帧开始。如果中断和空闲线路均已激活，则必须先满足中断的条件，然后满足空闲线路的条件。之后，满足其它条件之一（开始字符或开始序列）便足以开始数据传输。
开始条件“任意字符”(Any character) 不能与其它开始条件结合使用。

Receive_Conditions 参数的数据类型结构，第 1 部分（开始条件）

表格 5- 6　开始条件的 Receive_Conditions 的结构

参数	声明	数据类型	默认值	说明
START .STARTCOND	IN	Word	16#0002	指定开始条件 • 01H - 开始字符的检测 • 02H - 任意字符 • 04H - 换行符的检测 • 08H - 空闲线路的检测 • 10H - 开始序列 1 的检测 • 20H - 开始序列 2 的检测 • 40H - 开始序列 3 的检测 • 80H - 开始序列 4 的检测 将这些值相加即可组合开始条件。
START .IDLETIME	IN	Word	16#0028	空闲状态下检测到新的帧开始所需的位时间数量（默认值：W#16#28）。仅当与"检测空闲线路"(Detection of an idle line) 条件关联时。 0 到 FFFF
START .STARTCHAR	IN	Byte	16#0002	条件"起始字符"(Start character) 的起始字符。（默认值：B#16#2）
START .SEQ[1].CTL	IN	Byte	0	开始序列 1，禁用/激活每个字符的比较：（默认值：B#16#0） 这些是起始字符串的每个字符的激活位。 • 01H - 字符 1 • 02H - 字符 2 • 04H - 字符 3 • 08H - 字符 4 • 10H - 字符 5 当禁用特定字符的某个位时，这意味着字符串中该位置的每个字符都表示有效的开始字符串（例如，1FH = 已解释的全部 5 个字符）。

参数	声明	数据类型	默认值	说明
START.SEQ[1].STR[1] .. START.SEQ[1].STR.[5]	IN	Char[5]	0	开始序列 1，开始字符（5 个字符）
START.SEQ[2].CTL	IN	Byte	0	开始序列 2，禁用/激活每个字符的比较。默认值：B#16#0）
START.SEQ[2].STR[1] .. START.SEQ[2].STR.[5]	IN	Char[5]	0	开始序列 2，开始字符（5 个字符）
START.SEQ[3].CTL	IN	Byte	0	开始序列 3，禁用/激活每个字符的比较。默认值：B#16#0
START.SEQ[3].STR[1] .. START.SEQ[3].STR.[5]	IN	Char[5]	0	开始序列 3，开始字符（5 个字符）
START.SEQ[4].CTL	IN	Byte	0	开始序列 4，禁用/激活每个字符的比较。默认值：B#16#0
START.SEQ[4].STR[1] .. START.SEQ[4].STR.[5]	IN	Char[5]	0	开始序列 4，开始字符（5 个字符）

示例

以十六进制编码格式查看下列接收到的数据：“**68** 10 aa **68** bb 10 aa 16”。下表中提供了组态的起始字符串。成功收到第一个字符 68H 后，即会评估起始字符串。成功收到第四个字符（第二个 68H）后，即满足开始条件 1。一旦满足开始条件，即开始评估结束条件。

起始字符串的处理可能因字符之间的校验、成帧或时间间隔中存在不同错误而取消。由于不满足开始条件，因此这些错误将阻止接收数据（输出错误消息）。

表格 5- 7　开始条件：

开始条件	第一个字符	第一个字符 +1	第一个字符 +2	第一个字符 +3	第一个字符 +4
1	**68**H	xx	xx	**68**H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Receive_P2P 指令的结束条件

帧的结束由第一次出现的一个或多个已组态结束条件来定义。

可以在设备组态的通信接口的属性中组态结束条件，或者通过 Receive_Config 指令组态结束条件。每次 CPU 或通信模块的电压恢复后，接收参数（开始和结束条件）都会复位为设备组态中的设置。当 STEP 7 用户程序执行 Receive_Config 时，设置将变为 Receive_Config 的参数。

Receive_Conditions 参数的数据类型结构，第 2 部分（结束条件）

表格 5- 8　结束条件的 Receive_Conditions 的结构

参数	声明	数据类型	默认值	说明
END .ENDCOND	IN	Word	0	此参数指定帧结束的条件： • 01H - 响应超时 • 02H - 消息超时 • 04 - 字符延迟时间 • 08H - 最大帧长度 • 10 - 读取消息中的消息长度 (N+LEN+M) • 20H - 结束序列 • 40H - 固定帧长度
END.FIXLEN	IN	Word	1	固定帧长度：仅当选择结束条件"固定帧长度"(Fixed frame length) 时使用。 1 到 4000 个字节（最大为 4 KB，取决于模块）
END.MAXLEN	IN	Word	1	最大帧长度：仅当选择结束条件"最大帧长度"(Maximum frame length) 时使用。 1 到 4000 个字节（最大为 4 KB，取决于模块）
END.N	IN	Word	0	帧中长度字段的字节位置。仅与结束条件 N+LEN+M 一起使用。 1 到 4000 个字节（最大为 4 KB，取决于模块）
END .LENGTHSIZE	IN	Word	0	长度字段的大小（1、2 或 4 字节）。仅与结束条件 N+LEN+M 一起使用。
END .LENGTHM	IN	Word	0	长度字段后面未包含在长度字段值中的字符数量。此条目仅与结束条件 N+LEN+M 一起使用。0 到 255 字节
END.RCVTIME	IN	Word	200	指定发送帧后接收到第一个字符的等待时间。如果在指定时间内未收到字符，接收指令将终止并生成错误消息。此信息仅与条件"响应超时"(Response timeout) 结合使用。（0 到 65535 ms）。 注：该参数不可独立作为结束标准，至少须同一个其它结束条件配合使用。

参数	声明	数据类型	默认值	说明
END.MSGTIME	IN	Word	200	指定收到第一个字符后等待接收完整帧的时间。仅当选择了条件"消息超时"(Message timeout) 时才使用此参数。（0 到 65535 ms）
END .CHARGAP	IN	Word	12	输入字符间的最大位时间数。如果字符间的位时间数超出指定值，则满足结束条件。此信息仅与条件"字符延迟时间"(Response delay time) 结合使用。（0 到 65535 位时间） 注： 要实现较高的数据传输速度，建议采用至少 100 个位时间的值。
END.SEQ.CTL	IN	Byte	0	结束分隔符序列 1，禁用/激活每个字符的比较： 这些是结束字符串的每个字符的激活位。字符 1 是位 0，字符 2 是位 1，... ，字符 5 是位 4。如果取消激活特定字符的某个位，则表明字符串的这一位置的每个字符都是一致的。 **注：** 检查字符时注意顺序： 如果要使用一个结束分隔符，则必须在字符 5 (END.SEQ.STR[5]) 中输入条目，并且只有该字符必须在 END.SEQ.CTL 中激活。如果要使用两个结束分隔符，则必须在 END.SEQ.STR[5] 和 END.SEQ.STR[4] 中输入条目，并且只有这些字符必须在 END.SEQ.CTL 中激活。使用其它字符时也是如此。
END.SEQ .STR[1] .. END.SEQ .STR[5]	IN	Char[5]	0	结束分隔符 1，起始字符（5 个字符）

表格 5- 9　Receive_P2P 指令的通用参数

参数	声明	数据类型	默认值	说明
GENERAL .MBUF_SIZE	IN	Byte	255	输入要在 CM 的接收缓冲区中缓冲的帧数。 如果没有激活影响接收缓冲区响应（防止超时、数据流控制）的其它条件，那么一旦达到限制后，其它帧均会被丢弃。（1 至 255 个帧）
GENERAL .OW_PROT	IN	Byte	0	激活以下功能：在 CM 接收到新的帧且尚未读取 CM 的接收缓冲区时，不对已缓冲的帧进行覆盖。此设置可避免已缓冲的接收帧丢失。 • 0 - 未激活 • 1 - 已激活
GENERAL .CLR_MBUF	IN	Byte	0	当 CPU 启动时，激活接收缓冲区的删除功能。 当 CPU 从 STOP 切换为 RUN 时，自动删除接收缓冲区。接收缓冲区只包含 CPU 启动后收到的帧。 • 0 - 未激活 • 1 - 已激活

5.4.1.7　P3964_Config： 组态 3964 (R) 协议

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

P3964_Config 指令（协议组态）允许在运行期间更改 3964(R) 的协议参数，如字符延迟时间、优先级和块检查（使用程序）。

P3964_Config 的组态更改将保存在 CM 中，而不是 CPU 中。CPU 或通信模块的电压恢复后，设备组态中保存的参数会立即恢复。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	当此输入出现上升沿时，启动该指令。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的“硬件标识符”。 符号端口名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的“输入地址”。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
BCC	IN	USInt	Byte	1	启用/禁用块检查 • 0 = 不带块检查 • 1 = 带块检查
Priority	IN	USInt	Byte	1	选择优先级 • 0 = 优先级低 • 1 = 优先级高
Character-DelayTime	IN	UInt	Word	16#00DC	设置字符延迟时间（取决于设置的数据传输速率）（默认值：220 ms） 1 ms 到 65535 ms
AcknDelay-Time	IN	UInt	Word	16#07D0	设置确认延迟时间（取决于设置数据传输速率）（默认值：2000 ms） 1 ms 到 65535 ms
Buildup-Attempts	IN	USInt	Byte	16#0006	设置连接尝试次数（默认值：6 次连接尝试） 1 至 255
Repetition-Attempts	IN	USInt	Byte	16#0006	设置传输尝试次数（默认值：6 次连接尝试） 1 至 255

参数	声明	数据类型		默认值	说明
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
DONE	OUT	Bool		FALSE	如果上一个请求无错完成，将变为 TRUE 并保持一个周期
ERROR	OUT	Bool		FALSE	如果上一个请求有错完成，将变为 TRUE 并保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

5.4.1.8 Send_P2P：发送数据

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Send_P2P 指令（发送点对点数据）启动数据的传输，并将所分配缓冲区的内容传输到通信模块。当 CM 以指定的数据传输速率发送数据时，CPU 程序仍然执行。每个通信模块在任何时候只能有一个发送指令处于未决状态。当 CM 已经在发送帧时，如果执行第二条 Send_P2P 指令，则 CM 会指示错误。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	在此输入的上升沿开始向 CM 传输数据。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的“硬件标识符”。 符号端口名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的“输入地址”。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。

参数	声明	数据类型		默认值	说明
BUFFER	IN	Variant	Any	0	此参数指向发送缓冲区的存储区。 **注意：** • 不支持布尔数据和布尔字段。 • 如果发送缓冲区在优化存储区中，则发送数据的最大允许长度为 1024 字节。 例外：支持的字节数组、字或双字的长度最大为 4096 字节。 • 如果发送缓冲区是字符串或宽字符串，则不使用当前长度和最大长度传送字符串内容。 更多信息，请参见"使用 BUFFER 和 LENGTH 参数进行通信操作 (页 113)"
LENGTH	IN	UInt	Word	0	要传输的数据长度（字节）。 在 BUFFER 参数中被寻址的存储区完全通过 LENGTH = 0 传输。 更多信息，请参见"使用 BUFFER 和 LENGTH 参数进行通信操作 (页 113)"
COM_RST	IN/OUT	---	Bool	FALSE	初始化 Send_P2P 指令 将使用 1 对指令进行初始化。随后会将 COM_RST 复位为 0。 注： 该参数仅适用于 S7-300/400 指令。
UNI-VERSAL [1]	OUT	Bool	---	FALSE	在 CPU 和通过 PORT 指定的 CM 之间进行数据通信的类型： FALSE：性能优化选项（同步）(页 47) • 接收帧最多 24 个字节 • 发送帧最多 30 个字节 TRUE Universal（异步） • 根据 CM 将帧长度限制为 1、2 或 4 KB
DONE	OUT	Bool		FALSE	如果上一个请求无错完成，将变为 TRUE 并保持一个周期

参数	声明	数据类型	默认值	说明
ERROR	OUT	Bool	FALSE	如果上一个请求有错完成，将变为 TRUE 并保持一个周期
STATUS	OUT	Word	16#7000	错误代码（请参见错误消息 (页 126)）

[1] 自库版本 V4.0 起可用

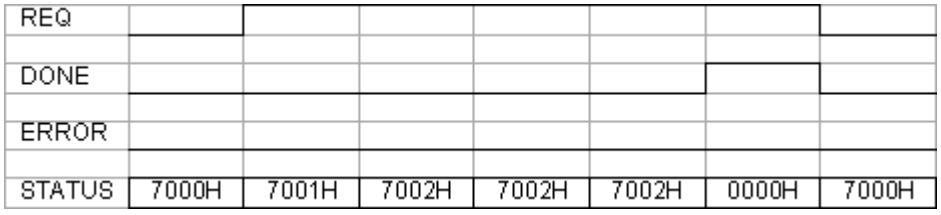
有关常规参数的更多信息，请参见"用于自由口操作的通用参数 (页 91)"。

参数

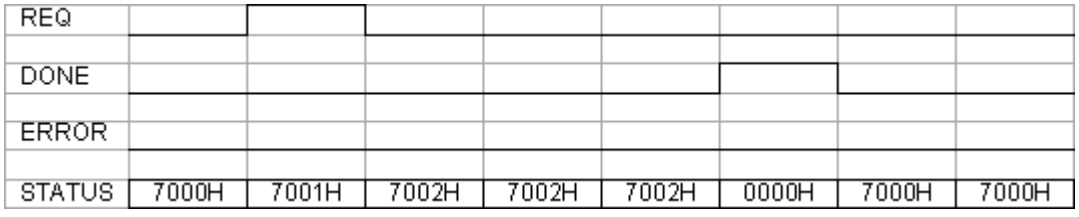
正在处理发送指令时，DONE 和 ERROR 输出处于 FALSE 状态。发送指令结束时，DONE 或 ERROR 输出中会有一个设为 TRUE 并保持一个周期，以指示发送指令的状态。当 ERROR 的状态为 TRUE 时，可以评估 STATUS 输出中的错误代码。

通信接口接受发送数据时，指令输出状态 16#7001。如果 CM 仍在发送，随后执行的 Send_P2P 输出值 16#7002。发送指令结束时，CM 输出发送指令状态 16#0000（如果未发生错误）。随后执行的 Send_P2P (REQ = 0) 输出状态 16#7000（空闲）。

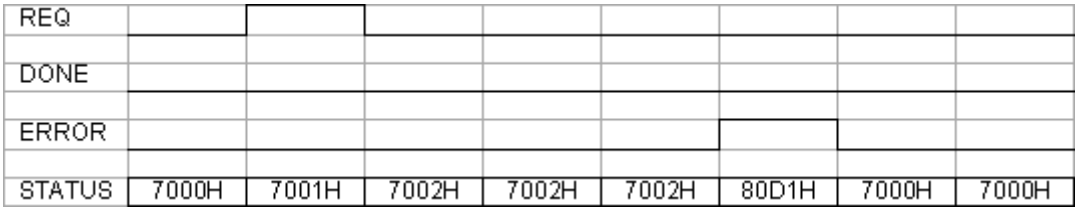
下图显示输出值与 REQ 之间的关系。它的假设条件是周期性调用指令检查发送过程的状态（由 STATUS 值指示）。



下图显示，如果脉冲在 REQ 电平线上处于待定状态（持续一个周期）以触发发送指令，DONE 和 STATUS 参数为何只对一个周期有效。



下图显示出错时 DONE、ERROR 和 STATUS 参数之间的关系。



DONE、ERROR 和 STATUS 值只在以相同背景数据块再次执行 Send_P2P 之前有效。

5.4.1.9 使用 BUFFER 和 LENGTH 参数进行通信操作

为 Send_P2P 交互 BUFFER 和 LENGTH 参数

Send_P2P 指令发送的最小数据大小为 1 字节。

调用期间，当 LENGTH 参数中传递"0"时，BUFFER 参数会指定要发送数据的大小。对此，变量的规格足够。

无法对 BUFFER 参数使用 Bool 数据类型或 Bool 类型的数组。如果要传输大量数据，我们建议对阵列或结构数据类型进行映射。

表格 5- 10 BUFFER 参数

BUFFER	**说明**
基本数据类型	发送时：LENGTH 值必须包括此数据类型的字节大小。 示例：对于 Word 值，LENGTH 必须为 2。对于 DWord 值或 Real 值，LENGTH 必须为 4。
结构	如果未激活性能优化选项： • 对于优化存储器：允许的最大 BUFFER 长度为 1024 Byte；否则，根据模块的不同，允许的最大长度为 4 KB。 • 传输时，如下要求适用：LENGTH 值可以包括小于结构完整字节长度的字节大小；这种情况下，只发送 BUFFER 的第一个 LENGTH 结构字节。 如果激活了性能优化选项： • BUFFER 允许的最大长度为 30 字节。

BUFFER	说明
Array	对于优化存储器：如果数组数据类型不等于 Byte、Word 或 DWord，则允许的最大缓冲区长度为 1024 字节。如果存储器未经过优化，则根据数据结构的不同，传输的最大缓冲区长度可达 4 KB，与数据结构无关。 对于发送：LENGTH 值可包括小于数组完整字节长度的字节大小，其中，此字节大小是数据元素字节大小的倍数。示例：Word 类型的数组的 LENGTH 参数必须是 2 的倍数，而对 Real 类型的数组来说必须是 4 的倍数。 例如，如果 BUFFER 包括一个具有 15 个 DWord 元素（总计 60 字节）的数组并指定 LENGTH = 20，则传输数组前 5 个 DWord 元素。如果 LENGTH 未指定或值为 0，则传输整个数组。
String	LENGTH 参数包括要发送的数字或字符。只传输 String 的字符。不发送具有最大和实际 String 长度的字节。

表格 5- 11　LENGTH 参数

LENGTH	说明
= 0	传送 BUFFER 指定的存储区的完整内容。 如果 BUFFER 指向字符串，则除包含最大长度和实际长度的字节外，将传送全部字符串内容。
> 0	传送 BUFFER 指定的存储区的长达组态长度的内容。

5.4.1.10　Receive_P2P：接收数据

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Receive_P2P 指令（使用点对点通信接收数据）检查 CM 中接收到的帧。如果有可用的帧，则将其从 CM 传输到 CPU。在 STATUS 参数中指示接收错误。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	S7-300/400/WinAC		
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
BUFFER	IN	Variant	Any	0	此参数指向接收缓冲区的起始地址。此缓冲区必须足够大，以便接收最大帧长度。 **注：** • 不支持布尔数据或布尔字段。 • 如果接收缓冲区在优化存储区中，则接收数据的最大允许长度为 1024 字节。 例外：支持的字节数组、字或双字的长度最大为 4096 字节。 • 如果接收缓冲区是字符串或宽字符串，则接收数据将写入字符串的内容中，并据此设置字符串当前长度。 更多信息，请参见"使用 BUFFER 和 LENGTH 参数进行通信操作 (页 113)"
UNIVERSAL [1)]	OUT	Bool	---	FALSE	在 CPU 和通过 PORT 指定的 CM 之间进行数据通信的类型： FALSE：性能优化选项（同步）(页 47) • 接收帧最多 24 个字节 • 发送帧最多 30 个字节 TRUE Universal（异步） • 根据 CM 将帧长度限制为 1、2 或 4 KB

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
NDR	OUT	Bool		FALSE	如果新数据可用且指令无错完成，则为 TRUE 且保持一个周期。
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
ERROR	OUT	Bool		FALSE	如果指令完成但出现错误，则为 TRUE 且保持一个周期。
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）
LENGTH	OUT	UInt	Word	0	接收到的帧的长度（以字节为单位） 更多信息，请参见“使用 BUFFER 和 LENGTH 参数进行通信操作 (页 113)”。

[1] 自库版本 V4.0 开始提供

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

当 ERROR 的状态为 TRUE 时，可以评估 STATUS 输出中的错误代码。STATUS 值提供了终止 CM 中的接收操作的原因。
这通常是一个正值，表示接收操作已成功并且已检测到帧标准。
如果 STATUS 值为负（十六进制值的最高有效位置位），则接收操作因出错而终止，例如奇偶效验、帧或溢出错误。

每个通信模块均可缓冲一个模块特定的帧号。如果 CM 中存在多个帧，则 Receive_P2P 指令输出最早存在的帧 (FIFO)。

5.4.1.11 Receive_Reset：清除接收缓冲区

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Receive_Reset 指令（复位接收器）清除 CM 中的接收缓冲区。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	在此输入的上升沿开始向 CM 传输数据。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
DONE	OUT	Bool		FALSE	TRUE 保持一个周期意味着上次请求无错完成。
ERROR	OUT	Bool		FALSE	TRUE 意味着上次请求有错完成。如果此输出为 TRUE，则 STATUS 输出将包含相应的错误代码。
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

有关常规参数的更多信息，请参见"用于自由口操作的通用参数 (页 91)"。

5.4.1.12 Signal_Get： 读取状态

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Signal_Get 指令（获取 RS232 信号）会读取 RS232 伴随信号的当前状态并在相应指令输出中显示这些状态。

说明

限制

- 此指令仅适用于 CM RS232 BA 和 RS232 HF。
- 如果为操作模式设置了 RS232C，该指令也可适用于 CM PtP (ET200SP)。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	在此输入的上升沿开始向 CM 传输数据。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
NDR	OUT	Bool		FALSE	如果已经读取 RS232 伴随信号并且指令无错完成，则为 TRUE 并保持一个周期。
ERROR	OUT	Bool		FALSE	如果指令完成但出现错误，则为 TRUE 且保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）
DTR	OUT	Bool		FALSE	数据设备就绪，模块就绪（输出）
DSR	OUT	Bool		FALSE	数据设备就绪，通信站就绪（输入）
RTS	OUT	Bool		FALSE	发送请求，模块发送准备就绪（输出）
CTS	OUT	Bool		FALSE	发送准备就绪，通信站可以接收数据（输入）
DCD	OUT	Bool		FALSE	检测到数据载体信号，收到信号电平
RING	OUT	Bool		FALSE	呼叫显示，指示呼入

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

5.4.1.13 Signal_Set：设置伴随信号

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Signal_Set 指令（设置 RS232 信号）允许设置 RS232 通信信号。

说明

限制条件

- 此指令仅适用于 CM RS232 BA 和 RS232 HF。
- 如果为操作模式设置了 RS232C，该指令也可适用于 CM PtP (ET200SP)。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	当此输入出现上升沿时，启动该指令。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
SIGNAL	IN	Byte		0	选择要设置的信号（多种可能）： • 01H = RTS • 02H = DTR • 04H = DSR（仅适用于 DCE 接口类型）
RTS	IN	Bool		FALSE	发送请求，模块已准备好发送 在输出中设置该值（TRUE 或 FALSE），默认值：FALSE
DTR	IN	Bool		FALSE	数据终端就绪，模块就绪 在输出中设置该值（TRUE 或 FALSE），默认值：FALSE
DSR	IN	Bool		FALSE	数据终端就绪（仅适用于 DCE 接口类型），未使用。
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
DONE	OUT	Bool		FALSE	如果上一个请求无错完成，将变为 TRUE 并保持一个周期
ERROR	OUT	Bool		FALSE	如果上一个请求有错完成，将变为 TRUE 并保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

5.4.1.14 Get_Features： 获取扩展功能

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

如果模块支持，可使用 Get_Features 指令（获取扩展功能）获取模块支持 CRC 和生成诊断消息的功能信息。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	当此输入出现上升沿时，启动该指令。
PORT	IN	PORT	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
NDR	OUT	Bool		FALSE	如果新数据可用且指令无错完成，则为 TRUE 且保持一个周期
MODBUS_CRC	OUT	Bool		FALSE	Modbus CRC 支持
DIAG_ALARM	OUT	Bool		FALSE	生成诊断消息

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
SUPPLY_VOLT	OUT	Bool		FALSE	对电源电压 L+ 缺失的诊断可用：
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
ERROR	OUT	Bool		FALSE	如果指令完成但出现错误，则为 TRUE 且保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

5.4.1.15 Set_Features： 设置扩展功能

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

如果模块支持，可使用 Set_Features 指令（选择扩展功能）激活 CRC 支持和诊断消息生成。

参数

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
REQ	IN	Bool		FALSE	当此输入存在上升沿时，用于设置扩展功能的指令开始。
PORT	IN	PORT (UInt)	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的“硬件标识符”。 符号端口名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的“输入地址”。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
EN_MODBUS_CRC	IN	Bool		FALSE	激活 Modbus CRC 支持
EN_DIAG_ALARM	IN	Bool		FALSE	激活诊断消息生成

参数	声明	数据类型		默认值	说明
		S7-1200/1500	**S7-300/400/WinAC**		
EN_SUPPLY_VOLT	IN	Bool		FALSE	启用对电源电压 L+ 缺失的诊断 注： S7-1500 / ET 200MP 通信模块不支持此诊断。即使该参数可与 MODBUS_CRC 等一起设置，也同样不支持该支持。
COM_RST	IN/OUT	---	Bool	FALSE	指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。
DONE	OUT	Bool		FALSE	如果上一个请求完成并且没有错误，则为 TRUE 并保持一个周期
ERROR	OUT	Bool		FALSE	如果指令完成但出现错误，则为 TRUE 且保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 126)）

有关常规参数的更多信息，请参见“用于自由口操作的通用参数 (页 91)”。

5.4.1.16 错误消息

PtP 错误消息概述

错误消息在指令的 STATUS 输出中提供，并且可在该处进行评估或在用户程序中进行处理。

错误代码	说明	解决方案
16#0000	无错误	-
接收状态和错误代码		
16#0094	基于“接收固定/最大帧长度”(Receipt of fixed/maximum frame length) 识别的帧结束	-
16#0095	基于“消息超时”(Message timeou) 识别的帧结束	-
16#0096	基于“字符延迟时间”(Character delay time) 的结束而识别的帧结束	-
16#0097	因为达到最大响应时间而中止了帧。	-
16#0098	基于“读取消息中的消息长度”(Read message length from message) 条件的实现而识别的帧结束	-
16#0099	基于“结束序列”(End sequence) 的接收而识别的帧结束	-
发送状态和错误代码		
16#7000	块空闲	-
16#7001	新帧的初始调用：数据传输已启动	-
16#7002	中间调用：数据传输运行	-
16#8085	无效长度	选择合适的帧长度。 • UNIVERSAL = 1（通过数据集进行数据通信）：根据模块，允许：1 到 1024/2048/4096 字节 • UNIVERSAL = 0（通过 IO 数据进行数据通信；针对多短帧情况进行性能优化 (页 47)）：最大长度为 30 个字节（Send_P2P 指令）。

错误代码	说明	解决方案
16#8087	UNIVERSAL = 0（性能优化选项）时，CM PTP 模块接收的字符数超过支持的数量。	选择合适的帧长度或使用 UNIVERSAL = 1（通过数据集进行数据通信）。 如果 UNIVERSAL = 0（性能优化选项 (页 47)），则最大长度为 24 字节（Receive_P2P 指令）。
16#8088	指定长度超过了接收缓冲区中设置的范围。 注：如果将 BUFFER 参数指定为 STRING 数据类型，则在当前字符串长度小于 LENGTH 参数指定的长度时，同样会出现此错误代码。	改变接收缓冲区的范围或选择与接收缓冲区中设置的范围相对应的帧长度。 根据模块，允许：1 到 1024/2048/4096 字节
16#8090	组态错误：WString 字节数为奇数	选择一个偶数字节数。
16#8091	UNIVERSAL = 0（性能优化选项）时，数据集 48、49 和 50 不受支持。	禁用"针对多短帧情况进行性能优化"(Performance optimized for many short frames) 参数。 或 通过 IO 数据访问接收和传输的数据。 或 至少使用 V4.0 版本的指令库 PtP Communication。
接收状态和错误代码		
16#7001	新帧的初始调用：数据传输已启动	-
16#7002	中间调用：数据传输运行	-
16#8088	接收的字符数超过 BUFFER 参数指定的字符数。	选择合适的帧长度。 根据模块，允许：1 到 1024/2048/4096 字节
16#8090	组态错误：WString 字节数为奇数	选择一个偶数字节数。

错误代码	说明	解决方案
特殊功能的错误消息代码		
16#818F	错误的参数编号设置（仅限 USS）	选择合适的参数编号 (PARAM)。 允许下列编号：0 到 2047
16#8190	CRC 计算的错误设置	为 CRC 计算选择合适的值。 以下内容有效：禁用或激活。 检查所寻址的模块是否支持 CRC 计算。
16#8191	诊断错误中断的错误设置	为“诊断中断”(Diagnostics interrupt) 选择合适的值。 以下内容有效：取消激活诊断中断或激活诊断中断。 检查所寻址的模块是否支持生成诊断中断。
16#8193	该模块不支持电源电压诊断 L+。	为“诊断中断”(Diagnostics interrupt) 选择合适的值。 以下内容有效：取消激活诊断中断或激活诊断中断。 检查所寻址的模块是否支持生成诊断中断。
“端口组态”的错误消息代码		
16#81A0	该模块不支持此协议。	为模块选择一个有效的协议 (PROTOCOL)。
16#81A1	该模块不支持此数据传输速率。	为模块选择有效的数据传输速率 (BAUD)。
16#81A2	该模块不支持此奇偶校验设置。	为“奇偶校验”(Parity) 选择合适的值 (PARITY)。 以下内容有效： • 无 (1) • 偶校验 (2) • 奇校验 (3) • 标记校验 (4) • 间隔校验 (5) • 任意 (6)

错误代码	说明	解决方案
16#81A3	该模块不支持此数据位数。	为"数据位数"(Number of data bits) 选择合适的值 (DATABITS)。 以下内容有效： • 7 (2) • 8 (1)
16#81A4	该模块不支持此停止位数。	为"停止位数"(Number of stop bits) 选择合适的值 (STOPBITS)。 以下内容有效： • 1 (1) • 2 (2)
16#81A5	该模块不支持此数据流控制类型。	为模块选择有效的数据流控制 (FLOWCTRL)。
16#81A7	XON 或 XOFF 的值无效	为 XON (XONCHAR) 和 XOFF (XOFFCHAR) 选择合适的值。 数值的有效范围：0 到 127
16#81AA	无效的工作模式	有效的工作模式包括： • 全双工 (RS232) (0) • 全双工 (RS422) 四线制模式（点对点）(1) • 全双工 (RS422) 四线制模式（多点主站）(2)/(CM PtP (ET 200SP)) • 全双工 (RS422) 四线制模式（多点从站）(3)/(CM PtP (ET 200SP)) • 半双工 (RS485) 两线制操作。(4)
16#81AB	无效接收线路初始状态	有效的初始状态是： • "无"默认设置 (0) • 信号 R(A)=5 V、信号 R(B)=0 V（断路检测）(1)： 仅可以选择以下项："全双工 (RS422) 四线制模式（点对点连接）"和"全双工 (RS422) 四线制模式（多点从站）"。 • 信号 R(A)=0 V、信号 R(B)=5 V (2)：此默认设置对应于空闲状态（无激活的发送操作）。

错误代码	说明	解决方案
16#81AC	“断路检测”(Break detection) 的值无效	为“断路检测”(Break detection) 选择合适的值。以下内容有效： • 断路检测已禁用 (0) • 断路检测已激活 (1)。
16#81AF	该模块不支持此协议。	为该模块选择一个有效的协议。
“发送组态”的错误代码		
16#81B5	两个以上的结束符，或 结束序列 > 5 个字符	为“结束符”(End delimiter) 和“结束序列”(End sequence) 选择合适的值。 以下内容有效： • 取消激活 (0)， • 1 个 (1) 或 2 个 (2) 结束符 或 • 取消激活 (0)， • 结束序列的 1 个 (1) 至最多 5 个 (5) 字符。
16#81B6	因选择了 3964(R) 协议而拒绝了发送组态	如果设置 3964(R) 协议，则确保未传输发送组态。
“接收组态”的错误代码		
16#81C0	开始条件无效	选择合适的启动条件。 以下内容有效： • 在帧开始前发送中断 • 发送“空闲线路”(Idle Line)。
16#81C1	结束条件无效或未选择结束条件	选择合适的结束条件（请参见使用自由口发送数据 (页 50)）。
16#81C3	“最大消息长度”(Maximum message length) 的值无效	为“最大信息长度”(Maximum message length) 选择合适的值 (MAXLEN)。 值的有效范围（取决于模块）：1 到 1024/2048/4096（字节）
16#81C4	“消息中长度规范的偏移量”(Offset of the length specification in the message) 的值无效	为“消息中长度规格的偏移量”(Offset of the length specification in the message) 选择合适的值。 值的有效范围（取决于模块）：1 到 1024/2048/4096（字节）

错误代码	说明	解决方案
16#81C5	“长度字段的大小”(Size of length field) 的值无效	为“长度字段的大小”(Size of length field) 选择合适的值 (LENGTHSIZE)。 值的有效范围（以字节表示）： • 1 (1) • 2 (2) • 4 (4)
16#81C6	“长度规范中未计字符数”(Number of characters not counted in length specification) 的值无效	为“长度规范中未计字符数”(Number of characters not counted in length specification) 选择合适的值 (LENGTHM)。 数值的有效范围：0 到 255（字节）
16#81C7	“消息偏移量 + 长度字段大小 + 未计字符数”的总和大于最大帧长度	为“消息偏移量”(Offset in message)、“长度字段大小”(Size of length field) 和“未计字符数”(Number of characters not counted) 选择合适的值。 数值的有效范围： • 消息偏移量（取决于模块）： 0 ... 1024/2048/4096（字节） • “长度字段的大小”(Size of length field)： 1、2 或 4（字节） • “未计字符数”(Number of characters not counted)：0 到 255（字节）
16#81C8	“响应超时”(Response timeout) 的值无效	为“响应超时”(Response timeout) 选择合适的值。 数值的有效范围：1 到 65535 (ms)
16#81C9	“字符延迟时间”(Character delay time) 的值无效	为“字符延迟时间”(Character delay time) 选择合适的值。 数值的有效范围：1 至 65535（位时间）
16#81CB	激活了帧结束序列，但没有为检查激活字符	为检查激活一个或多个字符。
16#81CC	激活了帧开始序列，但没有为检查激活字符	为检查激活一个或多个字符。

错误代码	说明	解决方案
16#81CD	“禁止覆盖”(Prevent overwriting) 的值无效	为“禁止覆盖”(Prevent overwriting) 选择合适的值。 以下内容有效： • 取消激活防止覆盖操作 (0) 或 • 激活防止覆盖操作 (1)
16#81CE	“启动时清空接收缓冲区”(Clear receive buffer on startup) 的值无效	为“启动时清空接收缓冲区”(Clear receive buffer on startup) 选择合适的值。 以下内容有效： • 取消激活启动时清除接收缓冲区 (0) • 激活启动时清除接收缓冲区 (1)
发送状态和错误代码		
16#81D0	在发送命令运行期间接收发送请求	确保未在发送命令运行期间接收到附加发送请求。
16#81D1	XON 或 CTS = ON 的等待时间已结束。	通信伙伴有故障、太慢或已离线。检查通信伙伴，或在需要时更改参数。
16#81D2	“硬件 RTS 始终开启”(Hardware RTS always ON)：发送作业因从 DSR = ON 更改为 DSR = OFF 而取消	检查通信伙伴。确保 DSR 在整个传输持续期间内均保持为 ON。
16#81D3	发送缓冲区上溢/发送帧太长	选择较短的帧长度。 以下内容有效（取决于模块）：1 到 1024/2048/4096（字节）
16#81D5	传输因参数更改、检测到线路断路或 CPU 处于 STOP 状态而取消	检查参数分配、线路断路和 CPU 状态。
16#81D6	传输因未接收到结束标识符而取消	检查结束符的参数分配和通信伙伴的帧。
16#81D7	用户程序和模块间的通信错误	检查通信（例如，匹配序列号）。
16#81D8	尝试传输因未组态模块而被拒绝	组态模块。
16#81DF	模块因为下列其中一个原因复位了 FB 的接口： • 模块重启 • 模块参数重新分配 • CPU STOP	—

错误代码	说明	解决方案
接收组态的错误代码		
16#81E0	帧已中止：接收缓冲区上溢/接收帧太大	增加用户程序中对接收功能的调用率，或组态带有数据流控制的通信。
16#81E1	帧已中止：奇偶校验错误	检查通信伙伴的连接线路，或确认两台设备是否针对相同的数据传输速率、奇偶校验和结束位数进行了组态。
16#81E2	帧已中止：字符帧错误	检查起始位、数据位、奇偶校验位、数据传输速率和结束位的设置。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。
16#81E3	帧已中止：字符上溢错误	固件出错：请联系客户支持。
16#81E4	帧已中止："消息偏移量 + 长度字段大小 + 未计字符数"(Offset in the message + size of the length field + number of characters not counted) 的总长度大于接收缓冲区	为消息偏移量、长度字段大小和未计字符数选择合适的值。
16#81E5	帧已中止：中断	到伙伴的接收线路断路。 重新连接或接通伙伴电源。
16#81E6	超出"缓冲的接收帧数"(Buffered receive frames) 最大值	在用户程序中更频繁调用指令、利用数据流控制组态通信或者增加已缓冲的帧数。
16#81E7	模块和 Receive_P2P 同步出错	确保 Receive_P2P 的多个实例未访问相同的模块。
16#81E8	帧已中止：字符延时时间在检测到消息结束标准前结束	伙伴设备有故障或太慢。根据需要，使用传输线路中互联的接口测试设备对此进行检查。
16#81E9	Modbus CRC 错误（仅限支持 Modbus 的通信模块）	Modbus 帧的校验和错误。检查通信伙伴。
16#81EA	Modbus 帧过短（仅限支持 Modbus 的通信模块）	不符合 Modbus 帧的最短长度。检查通信伙伴。
16#81EB	帧已中止：达到最大帧长度	在通信伙伴上选择较短的帧长度。 以下内容有效（取决于模块）：1 到 1024/2048/4096（字节） 检查帧检测结束参数。

错误代码	说明	解决方案
错误代码 V24 伴随信号		
16#81F0	模块不支持 V24 伴随信号	您已尝试不支持 V24 伴随信号的模块设置伴随信号。确保此为 RS 232 模块或者已设置 RS232 模块 (ET 200SP)。
16#81F1	无 V24 伴随信号操作	如果激活了硬件数据流控制，则无法手动操作 V24 伴随信号。
16#81F2	由于该模块的类型为 DTE，因此无法设置 DSR 信号。	检查模块的组态类型。 模块类型必须为 DCE（数据通信设备）。
16#81F3	由于该模块的类型为 DCE 类型，因此无法设置 DTR 信号。	检查模块的组态类型。 模块类型必须为 DTE（数据终端设备）。
16#81F4	块头错误（例如，块类型不正确或块长度不正确）	检查背景数据块和块头。
接收组态的错误代码		
16#8201 [1]	Receive_Conditions 是指向无效数据类型的指针	输入一个指向以下数据类型的指针： DB、BOOL、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TIME_OF_DAY、TIME、S5TIME、DATE_AND_TIME 和 STRING
16#8225	Receive_Conditions 指向大于 1 kB 的优化存储区 或 Receive_Conditions 指向优化存储区并且接收长度大于 Receive_Conditions 访问的区域。	输入一个指针，其指向区域的最大长度需满足： • 优化存储区：1 KB • 非优化存储区：4 KB 注：如果指针指向优化存储区，发送的数据不要超过 1 KB。
16#8229 [1]	Receive_Conditions 是指向 BOOL 的指针，其位数不等于 n * 8	如果使用指向 BOOL 的指针，位数必须是 8 的倍数。
错误代码，一般		

错误代码	说明	解决方案
16#8280	读取模块时进行否定确认	有关错误原因的更多详细信息，请参见 RDREC.STATUS 静态参数和 SFB RDREC 的说明。 • 检查 PORT 参数中的输入 • 首次调用前设置 COM_RST 参数。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。
16#8281	写入模块时进行否定确认	检查 PORT 参数中的输入 在 WRREC.STATUS 静态参数和 SFB WRREC 的说明中会找到有关错误原因的更多详细信息。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。
16#8282	模块不可用	检查 PORT 参数中的输入并确保模块可以访问。
接收组态的错误代码		
16#82C1	“缓冲的接收帧数”(Buffered receive frames) 的值无效。	为“缓冲的接收帧数”(Buffered receive frames) 选择合适的值。 数值的有效范围：1 至 255
16#82C2	因选择了 3964(R) 协议而拒绝了接收组态	如果设置了 3964(R) 协议，则请确保未发送接收组态。
16#8301 [1]	Receive_Conditions 是指向无效数据类型的指针	选择有效的数据类型。 以下内容有效：DB、BOOL、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TIME_OF_DAY、TIME、S5TIME、DATE_AND_TIME 和 STRING
16#8322	读取参数时发生范围长度错误	检查 Receive_Conditions 参数上的输入
16#8324	读取参数时发生范围错误	检查 Receive_Conditions 参数上的输入
16#8328	读取参数时发生设置错误	检查 Receive_Conditions 参数上的输入

错误代码	说明	解决方案
发送状态和错误代码		
16#8328 [1]	BUFFER 是指向 BOOL 的指针，其位数不等于 n * 8	如果使用指向 BOOL 的指针，位数必须是 8 的倍数。
接收组态的错误代码		
16#8332	Receive_Conditions 参数有无效数据块	检查 Receive_Conditions 参数上的输入
16#833A	Receive_Conditions 参数上的数据块标志表示的是未下载的数据块。	检查 Receive_Conditions 参数上的输入
16#8351	数据类型无效	检查 Receive_Conditions 参数上的输入
16#8352 [1]	Receive_Conditions 未指向数据块	检查指向 Receive_Conditions 的指针
16#8353 [1]	Receive_Conditions 未指向 Receive_Conditions 类型的结构	检查指向 Receive_Conditions 的指针
错误代码 3964(R) 协议		
16#8380	参数分配错误："字符延迟时间"(Character delay time) 值无效。	为"字符延迟时间"(Character delay time) (CharacterDelayTime) 选择合适的值。 数值的有效范围：1 ... 65535 (ms)
16#8381	参数分配错误："响应超时"(Response timeout) 值无效。	为"响应超时"(Response timeout) (AcknDelayTime) 选择合适的值。 数值的有效范围：1 ... 65535 (ms)
16#8382	参数分配错误："优先级"(Priority) 值无效。	为"优先级"(Priority) (Priority) 选择合适的值。 以下内容有效： • 高 (1) • 低 (0)
16#8383	参数分配错误："块检查"(Block check) 值无效	为"块检查"(Block check) 选择合适的值 (BCC)。 以下内容有效： • 带块检查 (1) • 不带块检查 (0)
16#8384	参数分配错误："连接尝试次数"(Connection attempts) 值无效。	为"连接尝试次数"(Connection attempts) (BuildupAttempts) 选择合适的值。 数值的有效范围：1 至 255

错误代码	说明	解决方案
16#8385	参数分配错误：“传输尝试次数”(Transmission attempts) 值无效。	为“传输尝试次数”(Transmission attempts) (RepetitionAttempts) 选择合适的值。 数值的有效范围：1 至 255
16#8386	运行错误：超出连接尝试次数	检查接口电缆和传输参数。 还要检查是否在伙伴设备上正确组态了接收功能。
16#8387	运行错误：超出传输尝试次数	检查接口电缆、传输参数和通信伙伴的组态。
16#8388	运行错误：“块检查字符”(Block check character) 错误 内部计算的块检查字符值与伙伴在连接结束时收到的块检查字符不一致。	检查连接是否被严重破坏；此时也可以不时地查看错误代码。可以使用切换到传输线路的接口测试设备检查伙伴设备上的正确功能。
16#8389	运行错误：等待空闲接收缓冲区时接收到的无效字符	接收缓冲区为空时，通信伙伴的发送请求（STX，02H）仅使用 DLE 应答。之前不可能接收到其它字符（再次收到 STX 除外）。 可以使用切换到传输线路的接口测试设备检查伙伴设备上的正确功能。
16#838A	运行错误：接收时发生逻辑错误。 收到 DLE 后，又收到一个随机字符（DLE 或 ETX 除外）。	检查伙伴是否总是复制帧报头和数据字符串中的 DLE，或连接是否用 DLE ETX 终止。可以使用切换到传输线路的接口测试设备检查伙伴设备上的正确功能。
16#838B	运行错误：超过字符延时时间	伙伴设备过慢或发生故障。 根据需要，用切换到传输线路上的接口测试设备进行验证。
16#838C	运行错误：空闲接收缓冲区的等待时间已开始	在用户程序中更频繁调用指令或者利用数据流控制组态通信。
16#838D	运行错误：未在 NAK 4 秒后开始帧重复	检查通信伙伴。伙伴必须在 4 秒内重复所接收到的可能受损的帧。
16#838E	运行错误：在空闲模式下，收到了一个或多个字符（NAK 或 STX 除外）。	可以使用切换到传输线路的接口测试设备检查伙伴设备的正确功能。
16#838F	运行错误：初始化冲突 - 两个伙伴均具有高优先级	在其中一个伙伴上设置“低”(Low) 优先级
16#8391	参数分配错误：因设置了自由口而拒绝了 3964 组态数据	如果已设置自由口协议，确保未发送任何 3964 参数分配数据。

错误代码	说明	解决方案
错误代码，一般		
16#8FFF	模块因复位而暂时未准备就绪。	重复请求。

[1] 仅限 S7-300/400 CPU 的指令

5.4.2 Modbus (RTU)

5.4.2.1 库版本间的依赖性

必须按照下列一一对应的组合关系来使用“MODBUS (RTU)”和“点对点”指令库：

“MODBUS (RTU)”库版本	“点对点”库版本
V1.1	V1.1
V2.1	V2.4
V3.1	V2.4
V4.4	V3.2
V5.0	V4.0
V5.1	V4.1

5.4.2.2 Modbus RTU 通信概述

Modbus RTU 通信

Modbus RTU (Remote Terminal Unit) 是用于网络中通信的标准协议，使用 RS232 或 RS422/485 连接在网络中的 Modbus 设备之间进行串行数据传输。

Modbus RTU 使用主/从站网络，其中整个通信仅由一个主站设备触发，而从站只能响应主站的请求。主站将请求发送到一个从站地址，并且只有该地址上的从站做出响应。

例外：Modbus 从站地址为 0 时会向所有从站发送广播帧（从站均不响应）。

Modbus 功能代码

- 作为 Modbus RTU 主站运行的 CPU 能够在 Modbus RTU 从站中通过通信连接读取和写入数据和 I/O 状态。
- 作为 Modbus RTU 从站运行的 CPU 允许利用通信连接进行连接的 Modbus RTU 主站在其自身的 CPU 中读取并写入数据和 I/O 状态。

表格 5- 12　用于读取数据的功能：读取分布式 I/O 和程序数据

Modbus 功能代码	用于读取从站（服务器）数据的功能 - 标准寻址
01	读取输出位：每个请求 1 至 2000/1992[1)] 位
02	读取输入位：每个请求 1 至 2000/1992[1)] 位
03	读取保持寄存器：每个请求 1 至 125/124[1)] 字
04	读取输入字：每个请求 1 至 125/124[1)] 字

1) 用于扩展寻址

表格 5- 13　用于写入数据的功能：更改分布式 I/O 和程序数据

Modbus 功能代码	用于向从站（服务器）写入数据的功能 - 标准寻址
05	写入一个输出位：每个请求 1 位
06	写入一个保持寄存器：每个请求 1 个字
15	写入一个或多个输出位：每个请求 1 至 1960 位
16	写入一个或多个保持寄存器：每个请求 1 至 122 个字

- Modbus 功能代码 08 和 11 提供从站设备的通信诊断选项。
- Modbus 从站地址为 0 时会将广播帧发送给所有从站（无从站响应；针对功能代码 5、6、15、16）。

表格 5- 14　Modbus 网络中的站地址

站		地址
RTU 站	标准站地址	1 到 247 ， 0 用于广播
	扩展站地址	1 到 65535，0 用于广播

Modbus 存储器地址

可用的 Modbus 存储器地址（输入/输出地址）的实际数量取决于 CPU 版本和可用的工作存储器。

程序中的 Modbus RTU 指令

- Modbus_Comm_Load：需要运行 Modbus_Comm_Load 来设置 PtP 参数，例如数据传输速率、奇偶校验和数据流控制。为 Modbus RTU 协议组态完通信模块后，它只能由 Modbus_Master 指令或 Modbus_Slave 指令使用。
- Modbus_Master：利用 Modbus 主站指令，CPU 可用作 Modbus RTU 主站设备，与一个或更多的 Modbus 从站设备进行通信。
- Modbus_Slave：利用 Modbus 从站指令，CPU 可用作 Modbus RTU 从站设备，与 Modbus 主站设备进行通信。

5.4.2.3 Modbus_Comm_Load： 对 Modbus 的通信模块进行组态

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Modbus_Comm_Load 指令组态通信模块，以用于通过 Modbus RTU 协议进行通信。当在程序中添加 Modbus_Comm_Load 指令时，将自动分配背景数据块。

Modbus_Comm_Load 的组态更改将保存在 CM 中，而不是 CPU 中。恢复电压和插拔时，将使用保存在设备配置中的数据组态 CM。必须在这些情况下调用 Modbus_Comm_Load 指令。

参数

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
REQ	IN	Bool		FALSE	当此输入出现上升沿时，启动该指令。
PORT	IN	Port	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。
BAUD	IN	UDInt	DInt	9600	选择数据传输速率 有效值为：300、600、1200、2400、4800、9600、19200、38400、57600、76800、115200 bps。对于订货号为 6ES7541-1AB01-0AB0 的模块，也允许使用值 250000 bps。
PARITY	IN	UInt	Word	0	选择奇偶校验： • 0 – 无 • 1 – 奇校验 • 2 – 偶校验
FLOW _CTRL	IN	UInt	Word	0	选择流控制： • 0 – （默认）无流控制 • 1 – 硬件流控制，RTS 始终开启（不适用于 RS422/485 CM） • 2 – 硬件流控制，RTS 切换（不适用于 RS422/485 CM）

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
RTS_ON_DLY	IN	UInt	Word	0	RTS 接通延迟选择： • 0 – 从"RTS 激活"直到发送帧的第一个字符之前无延迟。 • 1 到 65535 - 从"RTS 激活"一直到发送帧的第一个字符之前的延迟（以毫秒表示）（不适用于 RS422/485 CM）。不论选择 FLOW_CTRL 为何，都会使用 RTS 延迟。
RTS_OFF_DLY	IN	UInt	Word	0	RTS 关断延迟选择： • 0 - 从传送上一个字符一直到"RTS 未激活"之前无延迟 • 1 到 65535 - 从传送上一个字符直到"RTS 未激活"之前的延迟（以毫秒表示）（不适用于 RS422/485 端口）。不论选择 FLOW_CTRL 为何，都会使用 RTS 延迟。
RESP_TO	IN	UInt	Word	1000	响应超时： 5 ms 到 65535 ms - Modbus_Master 等待从站响应的时间（以毫秒为单位）。如果从站在此时间段内未响应，Modbus_Master 将重复请求，或者在指定数量的重试请求后取消请求并提示错误（请参见下文，RETRIES 参数）。
MB_DB	IN/OUT	MB_BASE		-	对 Modbus_Master 或 Modbus_Slave 指令的背景数据块的引用。 MB_DB 参数必须与 Modbus_Master 或 Modbus_Slave 指令的（静态，因此在指令中不可见）MB_DB 参数相连。
COM_RST	IN/OUT	---	Bool	FALSE	初始化 Modbus_Comm_Load 指令 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。 注： 该参数仅适用于 S7-300/400 指令。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
DONE	OUT	Bool		FALSE	如果上一个请求完成并且没有错误，DONE 位将变为 TRUE 并保持一个周期。
ERROR	OUT	Bool		FALSE	如果上一个请求完成出错，则 ERROR 位将变为 TRUE 并保持一个周期。STATUS 参数中的错误代码仅在 ERROR = TRUE 的周期内有效。
STATUS	OUT	Word		16#7000	错误代码（请参见错误消息 (页 173)）

执行 Modbus_Comm_Load 以对 Modbus RTU 协议的端口进行组态。为 Modbus RTU 协议组态完端口后，它只能由 Modbus_Master 指令或 Modbus_Slave 指令使用。

必须运行 Modbus_Comm_Load 来完成将用于 Modbus 通信的每个通信端口的组态。必须为使用的每个端口分配唯一的 Modbus_Comm_Load 背景数据块。如果需要更改数据传输速率或奇偶校验等通信参数，或者网络已经恢复，只需再次运行 Modbus_Comm_Load。

例如，当在程序中添加 Modbus_Master 或 Modbus_Slave 指令时，将自动为指令分配背景数据块。需要将 Modbus_Comm_Load 指令的 MB_DB 参数连接到 Modbus_Master 或 Modbus_Slave 指令的 MB_DB 参数。

Modbus_Comm_Load 数据块变量

下表显示了可在程序中使用的 Modbus_Comm_Load 背景数据块中的公共静态变量。

表格 5- 15　背景数据块中的静态变量

变量	数据类型		标准	说明
	S7-1200 /1500	**S7-300/400/ WinAC**		
ICHAR_GAP	Word		0	字符间的最长字符延迟时间。此参数以毫秒为单位指定，并且增加了所接收字符之间的预期周期。将此参数的相应位时间数添加到 Modbus 默认值 35 位时间（3.5 字符时间）。
RETRIES	Word		2	返回"无响应"错误代码 0x80C8 之前主站执行的重复尝试次数。
EN_SUPPLY _VOLT	Bool		0	启用对电源电压 L+ 缺失的诊断
MODE	USInt	字节	0	工作模式 有效的工作模式包括： • 0 = 全双工 (RS232) • 1 = 全双工 (RS422) 四线制模式（点对点） • 2 = 全双工 (RS 422) 四线制模式（多点主站，CM PtP (ET 200SP)） • 3 = 全双工 (RS 422) 四线制模式（多点从站，CM PtP (ET 200SP)） • 4 = 半双工 (RS485) 二线制模式 [1)]
LINE_PRE	USInt	字节	0	接收线路初始状态 有效的初始状态是： • 0 ="无"初始状态 [1)] • 1 = 信号 R(A)=5 V，信号 R(B)=0 V（断路检测）： 在此初始状态下，可进行断路检测。 仅可以选择以下项："全双工 (RS422) 四线制模式（点对点连接）"和"全双工 (RS422) 四线制模式（多点从站）"。 • 2 = 信号 R(A)=0 V，信号 R(B)=5 V： 此默认设置对应于空闲状态（无激活的发送操作）。在此初始状态下，无法进行断路检测。

变量	数据类型		标准	说明
	S7-1200 /1500	**S7-300/400/ WinAC**		
BRK_DET	USInt	字节	0	断路检测 以下内容有效： • 0 = 断路检测已禁用 • 1 = 断路检测已激活
EN_DIAG _ALARM	Bool		0	激活诊断中断： • 0 - 未激活 • 1 - 已激活
STOP_BITS	USINT	字节	1	停止位个数； • 1 = 1 个停止位, • 2 = 2 个停止位, • 0、3 到 255 = 保留

[1] 使用 PROFIBUS 电缆连接 CM 1241 的 RS485 时所需的设置

指令版本

版本 3.1 的功能与版本 3.0 完全相同，本次版本升级仅仅体现在内部措施方面。

5.4.2.4 Modbus_Master： 作为 Modbus 主站进行通信

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

Modbus_Master 指令可通过由 Modbus_Comm_Load 指令组态的端口作为 Modbus 主站进行通信。当在程序中添加 Modbus_Master 指令时，将自动分配背景数据块。Modbus_Comm_Load 指令的 MB_DB 参数必须连接到 Modbus_Master 指令的（静态）MB_DB 参数。

说明

无法为 Modbus_Master 指令的背景数据块激活保持 (Retain)。

参数

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
REQ	IN	Bool		FALSE	FALSE = 无请求 TRUE = 请求向 Modbus 从站发送数据
MB _ADDR	IN	UInt	Word	-	Modbus RTU 站地址： 标准地址范围（1 到 247 以及 0 用于 Broadcast） 扩展地址范围（1 到 65535 以及 0，用于 Broadcast） 值 0 为将报文广播到所有 Modbus 从站预留。广播仅支持 Modbus 功能代码 05、06、15 和 16。
MODE	IN	USInt	Byte	0	模式选择：指定请求类型（读取、写入或诊断）。下面的 Modbus 功能表中提供了更多信息。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
DATA_ADDR	IN	UDInt	DWord	0	从站中的起始地址：指定在 Modbus 从站中访问的数据的起始地址。下面的 Modbus 功能表中列出了有效地址。
DATA_LEN	IN	UInt	Word	0	数据长度：指定此指令将访问的位或字的个数。下面的 Modbus 功能表中列出了有效长度。
COM_RST	IN/OUT	---	Bool	FALSE	Modbus_Master 指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。 注： 该参数仅适用于 S7-300/400 指令。
DATA_PTR	IN/OUT	Variant	Any	-	数据指针：指向要进行数据写入或数据读取的标记或数据块地址。 自指令版本 V3.0 起： 该参数可指向优化存储区。在优化存储区中，允许使用以下数据类型的单个元素或数组：Bool, Byte, Char, Word, Int, DWord, DInt, Real, USInt, UInt, UDInt, SInt, WChar。所有其它数据类型都会导致出现错误消息 16#818C。
DONE	OUT	Bool		FALSE	如果上一个请求完成并且没有错误，DONE 位将变为 TRUE 并保持一个周期。
BUSY	OUT	Bool		-	• FALSE – Modbus_Master 无激活命令 • TRUE – Modbus_Master 命令执行中
ERROR	OUT	Bool		FALSE	如果上一个请求完成出错，则 ERROR 位将变为 TRUE 并保持一个周期。STATUS 参数中的错误代码仅在 ERROR = TRUE 的周期内有效。
STATUS	OUT	Word		0	错误代码（请参见错误消息 (页 173)）

Modbus 主站数据块中的变量

下表显示了可在程序中使用的 Modbus_Master 背景数据块中的公共静态变量。

表格 5- 16　背景数据块中的静态变量

变量	数据类型	标准	说明
Blocked_Proc _Timeout	Real	3.0	在"激活"状态移除此实例前，等待受阻的 Modbus 主站实例的持续时间（以秒为单位）。例如，如果输出主站请求，随后在其完全结束请求之前，程序停止并调用主站功能，则可能发生这种情况。时间值必须大于 0 秒而小于 55 秒，以避免发生错误。 另请参见"Modbus-Master 通信规则"和"使用不同的参数设置调用 Modbus_Master 指令"。
Extended_Addressing	Bool	FALSE	将从站地址组态为单字节或双字节。 • FALSE = 1 字节地址；0 到 247 • TRUE = 2 字节地址（对应于扩展地址）；0 到 65535
Compatibility_Mode [1]	Bool	FALSE	对于 Modbus，使用 Modbus RTU 驱动程序 的 CP 341、CP 441-2 和 ET 200S 1SI，以及 ET 200S 1SI 的兼容模式默认值为 0。 • FALSE = 根据 Modbus 规范，不兼容 • TRUE = 兼容 – 对于 FC1 和 FC2：从帧中读取的数据逐字写入访问的 CPU 存储器内，并逐字节替换。如果要传输的位数不是 16 的倍数，那么不相关的位将在最后一个字中设置为空值。 – 对于 FC15：要传输的字从访问的存储器中逐字读取并逐字节写入发送帧。如果要传输的位数不是 8 的倍数，那么最后一个字节中不相关的位将从访问的存储器中读取，并输入到发送帧中。
MB_DB	MB _BASE	-	Modbus_Comm_Load 指令的 MB_DB 参数必须连接到 Modbus_Master 指令的此 MB_DB 参数。

[1] 点对点通信模块会根据 Modbus 规范中的定义进行响应。对于 Modbus，要保留与 CP 341, CP 441-2 和 ET 200SP 1SI 一样的响应，使用"Compatibility_Mode"参数。

程序可以向 Blocked_Proc_Timeout 和 Extended_Addressing 变量中写入值来控制 Modbus 主站的操作。

Modbus-Master 通信规则

- 必须运行 Modbus_Comm_Load 来组态端口，以便 Modbus_Master 指令可以使用该端口进行通信。
- 要用来作为 Modbus 主站的端口不可作为 Modbus_Slave 使用。对于该端口，可以使用一个或多个 Modbus_Master [1] 的实例。但是，所有版本的 Modbus_Master 都必须为该端口使用相同的背景数据块。
- Modbus 指令不会使用通信报警事件来控制通信过程。程序必须查询 Modbus_Master 指令来获得完整的命令（DONE、ERROR）。
- 我们建议为来自程序周期 OB 的特定端口调用 Modbus_Master 的所有执行。Modbus 主站指令只能在一个程序周期或一个周期/时间控制的处理级别中执行。它们无法在不同的处理级别中进行处理。由具有较高优先级的处理级别中的 Modbus 主站指令引起的 Modbus 主站指令的优先级中断将导致操作不正确。Modbus 主站指令无法在启动、诊断或时间错误级别中处理。

[1] 此处的"Modbus 主站的实例"意味着，调用具有与 Modbus_Comm_Load 指令相同的互联，并具有与 MB_ADDR、MODE、DATA_ADDR 和 DATA_LEN 参数相同的设置的 Modbus_Master 指令。

示例

MODE = 0 且 DATA_ADDR = 10 时会调用 Modbus_Master

此作业将一直处于激活状态，直到通过参数 DONE=1 或 ERROR=1 完成，或者 Blocked_Proc_Timeout 参数中组态的监视时间到期。如果在看门狗时间用完且之前的命令还未完成时启动了新命令，那么之前的命令将会中止而不会有错误消息。

如果在此命令正在运行期间使用相同的实例数据以不同的 MODE 和 DATA_ADDR 参数设置再次调用，则第二次的调用将以 ERROR = 1 和 STATUS = 8200 终止。

使用不同的参数设置调用 Modbus_Master 指令

如果程序中含有使用不同 MB_ADDR、MODE、DATA_ADDR 或 DATA_LEN 设置的多个 Modbus_Master 指令调用，必须确保在任意给定时间，只有一个调用处于激活状态。否则，将输出错误消息 16#8200（接口正忙于处理当前请求）。

如果无法完整地完成调用，那么看门狗会由 Blocked_Proc_Timeout 参数激活，并终止当前命令。

REQ 参数

FALSE = 无请求；TRUE = 请求向 Modbus 从站发送数据

启用请求的传输。这会将缓冲区中的内容传送到点对点通信接口。

可以使用 DATA_ADDR 和 MODE 参数来选择 Modbus 功能代码。

DATA_ADDR（从站中的 Modbus 起始地址）：指定在 Modbus 从站中访问的数据的起始地址。

Modbus_Master 指令使用 MODE 输入，不使用功能代码输入。MODE 和 DATA_ADDR 结合使用可指定在实际 Modbus 帧中使用的功能代码。下表显示了 MODE 参数、Modbus 功能代码和 DATA_ADDR 中 Modbus 地址范围之间的关系。

表格 5- 17　Modbus 功能

MODE	DATA_ADDR （Modbus 地址）			DATA_LEN （数据长度）			Modbus 功能代码	运行和数据		
0				每个请求的位数			01	读取输出位：		
	1	到	9999	1	到	2000/1992 [1]		0	到	9998
0				每个请求的位数			02	读取输入位：		
	10001	到	19999	1	到	2000/1992 [1]		0	到	9998
0				每个请求的字数			03	读取保持寄存器：		
	40001	到	49999	1	到	125/124 [1]		0	到	9998
	400001	到	465535	1	到	125/124 [1]		0	到	65534
0				每个请求的字数			04	读取输入字：		
	30001	到	39999	1	到	125/124 [1]		0	到	9998
1				每个请求的位数			05	写入一个输出位：		
	1	到	9999	1				0	到	9998
1				每个请求 1 个字			06	写入一个保持寄存器：		
	40001	到	49999	1				0	到	9998
	400001	到	465535	1				0	到	65524

MODE	DATA_ADDR （Modbus 地址）			DATA_LEN （数据长度）			Modbus 功能代码	运行和数据		
1				每个请求的位数			15	写入多个输出位：		
	1	到	9999	2	到	1968/1960 [1]		0	到	9998
1				每个请求的字数			16	写入多个保持寄存器：		
	40001	到	49999	2	到	123/122		0	到	9998
	400001	到	465534	2	到	123/122 [1]		0	到	65534
2 [2]				每个请求的位数			15	写入一个或多个输出位：		
	1	到	9999	1	到	1968/1960 [1]		0	到	9998
2 [2]				每个请求的字数			16	写入一个或多个保持寄存器：		
	40001	到	49999	1	到	123		0	到	9998
	400001	到	465535	1	到	122 [1]		0	到	65534
11	此功能将忽略 Modbus_Master 的 DATA_ADDR 和 DATA_LEN 操作数。						11	读取从站通信的状态字和事件计数器。状态字表示"忙"（0 - 不忙，0xFFFF - 忙）。事件计数器随着帧的每次成功处理而递增。		
80				每个请求 1 个字			08	使用数据诊断代码 0x0000 检查从站状态（回送测试 - 从站返回请求的回应）		
	-			1				-		
81				每个请求 1 个字			08	利用数据诊断代码 0x000A 重新设置从站事件计数器		
	-			1				-		
104 [3]				每个请求的字数			04	读取输入字		
	0	到	65535	1	到	125/124 [1]		0	到	65535

MODE	DATA_ADDR （Modbus 地址）			DATA_LEN （数据长度）			Modbus 功能代码	运行和数据
3 到 10, 12 到 79, 82 到 103, 105 到 255	-			-				保留

[1] 在扩展寻址中（请参见 Extended_Adressing 参数），最大数据长度根据功能的数据类型而缩减 1 字节或 1 个字。

[2] MODE 2 允许使用 Modbus 功能 15 和 16 写入 1 个或多个输出位和 1 个或多个保持寄存器。
MODE 1 使用 Modbus 功能 5 和 6 写入 1 个输出位和 1 个保持寄存器，使用 Modbus 功能 15 和 16 写入多个输出位和多个保持寄存器。

[3] 以下情况适用于 S7-300/400/WinAC：不支持。

DATA_PTR 参数

DATA_PTR 参数指向在其中执行读取或写入的数据块或位存储器地址。如果使用数据块，则必须创建全局数据块，以便为 Modbus 从站上的读取或写入过程提供数据存储器。

说明

S7-1200/1500 - 使用 DATA_PTR 的访问的数据块必须支持直接寻址

数据块必须允许直接（绝对）寻址和符号寻址。

说明

使用功能代码 5

功能代码 5 用于设置或删除各个位。

设置位时，必须在通过 DATA_PTR 寻址到的 DB 或位存储区的首个字中指定值 “16#FF00”。

- 对于 S7-1200，也可以指定值“16#0100”以设置位。
- 为复位某个位，必须在通过 DATA_PTR 寻址到 DB 或位存储区的首个字中指定值 “16#0000”。

所有其它值通过 ERROR = TRUE 和 STATUS = 16#8384 拒绝。

DATA_PTR 参数的数据块结构

- 这些数据类型对读取 Modbus 地址范围 (DATA_PTR) 30001 到 39999、40001 到 49999 和 400001 到 465535 中的字有效，以及对写入 Modbus 地址范围（DATA_PTR 参数）40001 到 49999 和 400001 到 465535 中的字有效。
  - 数据类型 WORD、UINT 或 INT 的标准数组
  - WORD、UINT 或 INT 类型的指定结构，其中每个元素都有唯一的名称和一个 16 位的数据类型。
  - 指定的复杂结构，其中每个元素都有唯一的名称和一个 16 位或 32 位的数据类型。
- 用于读/写 Modbus 地址范围（DATA_PTR 参数）00001 到 09999 中的位和用于读取 10001 到 19999 中的位。
  - 布尔数据类型的标准字段。
  - 来自明确指定的布尔变量的指定布尔结构。
- 为了降低数据损坏的风险，推荐为每个 Modbus_Master 指令保留其自身的独立存储区。
- DATA_PTR 的数据区不必位于相同的全局数据块中。可以为 Modbus 读取过程创建具有多个区域的数据块，为 Modbus 写入过程创建数据块或为每个从站创建数据块。

指令版本

版本 3.0 的功能与版本 2.4 完全相同，本次版本升级仅仅体现在内部措施方面。

5.4.2.5 Modbus_Slave

Modbus_Slave: 作为 Modbus 从站进行通信

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

程序可利用 CM（RS422/485 或 RS232）端口，使用 Modbus_Slave 指令来作为 Modbus 从站进行通信。添加指令时，STEP 7 将自动创建背景数据块。Modbus_Comm_Load 指令的 MB_DB 参数必须连接到 Modbus_Slave 指令的（静态）MB_DB 参数。

说明

无法为 Modbus_Slave 指令的背景数据块激活保持 (Retain)。

参数

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
MB_ADDR	IN	UInt	Word	-	Modbus 从站的标准寻址： 标准寻址范围（1 到 247） 扩展寻址范围（0 到 65535） 注：0 是广播地址
COM_RST	IN/OUT	---	Bool	FALSE	Modbus_Slave 指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。 注： 该参数仅适用于 S7-300/400 指令。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
MB_HOLD_REG	IN/OUT	Variant	Any	-	Modbus 保持寄存器 DB 的指针：Modbus 保持寄存器可能为标志的存储区或者数据块。 自指令版本 V4.0 起： 该参数必须指向长度为 16 位以上的存储区，否则会导致出现错误消息 16#8187。这一要求对于单个元素、数组、STRUCT 和 UDT 均适用。例如，如果 Single Bool 或数组中包含的布尔元素个数小于 16，则会导致出现错误消息。 如果长度不是 16 位的倍数，则存储区末端的剩余位无法通过 Modbus_Slave 指令读取或写入。 该参数可指向优化存储区。在优化存储区中，允许使用以下数据类型的单个元素或数组：Bool, Byte, Char, Word, Int, DWord, DInt, Real, USInt, UInt, UDInt, SInt, WChar。所有其它数据类型都会导致出现错误消息 16#818C。
NDR	OUT	Bool		FALSE	可用的新数据： • FALSE - 无新数据 • TRUE – 表示新数据已由 Modbus 主站写入 如果上一个请求完成并且没有错误，NDR 位将变为 TRUE 并保持一个周期。
DR	OUT	Bool		FALSE	读取数据： • FALSE - 未读取数据 • TRUE - 表示该指令已将 Modbus 主站接收到的数据存储在目标区域中。 如果上一个请求完成并且没有错误，DR 位将变为 TRUE 并保持一个周期。
ERROR	OUT	Bool		FALSE	如果上一个请求完成出错，则 ERROR 位将变为 TRUE 并保持一个周期。如果执行因错误而终止，则 STATUS 参数中的错误代码仅在 ERROR = TRUE 的周期内有效。
STATUS	OUT	Word		0	错误代码（请参见错误消息 (页 173)）

Modbus 通信的功能代码（1、2、4、5 和 15）可直接在 CPU 的过程映像输入和过程映像输出中读取或写入位和字。对于这些功能代码，必须将 MB_HOLD_REG 参数定义为大于一个字节的数据类型。下表显示了将 Modbus 地址分配给 CPU 中过程映像的示例。

表格 5- 18　将 Modbus 地址分配给过程映像

Modbus 功能						S7-1200			
代码	功能	数据区	地址区			数据区	CPU 地址		
01	读取位	输出	0	到	8191	过程映像输出	O0.0	到	O1023.7
02	读取位	输入	0	到	8191	过程映像输入	I0.0	到	I1023.7
04	读取字	输入	0	到	511	过程映像输入	IW0	到	IW1022
05	写入位	输出	0	到	8191	过程映像输出	O0.0	到	O1023.7
15	写入位	输出	0	到	8191	过程映像输出	O0.0	到	O1023.7

表格 5- 19　将 Modbus 地址分配给过程映像

Modbus 功能						S7-1500/S7-300/S7-400			
功能代码	功能	数据区	地址区			数据区	CPU 地址		
01	读取位	输出	0	到	9998	过程映像输出	O0.0	到	A1249.6
02	读取位	输入	0	到	9998	过程映像输入	I0.0	到	I1249.6
04	读取字	输入	0	到	9998	过程映像输入	IW0	到	IW19996
05	写入位	输出	0	到	9998	过程映像输出	O0.0	到	A1249.6
15	写入位	输出	0	到	9998	过程映像输出	O0.0	到	A1249.6

说明

可用的地址区可能更小，具体取决于 CPU 的存储器组态。

Modbus 通信的功能代码（3、6 和 16）使用 Modbus 保持寄存器，此寄存器是标志的存储区或者数据块中的一个地址区。保持寄存器的类型由 Modbus_Slave 指令的 MB_HOLD_REG 参数指定。

说明

S7-1200/1500 - MB_HOLD_REG 数据块的类型

具有 Modbus 保持存器的数据块必须允许直接（绝对）寻址和符号寻址。

表格 5- 20　诊断功能

S7-1200 Modbus_Slave 的 Modbus 诊断功能		
功能代码	**子功能**	**说明**
08	0000H	输出回应测试的请求数据：Modbus_Slave 指令会将所接收数据字的回应返回到 Modbus 主站。
08	000AH	清除通信事件计数器：Modbus_Slave 指令将清除用于 Modbus 功能 11 的通信事件计数器。
11		调用通信事件计数器：Modbus_Slave 指令使用内部通信事件计数器来检测将发送到 Modbus 从站的成功的 Modbus 读取和 Modbus 写入数量。该计数器不随功能 8、功能 11 和广播请求而递增。它也不会随导致通信错误（例如，奇偶校验或 CRC 错误）的请求而递增。

Modbus_Slave 指令支持来自 Modbus 主站的广播写入请求，只要该请求包括到有效地址的访问即可。针对广播功能不支持的功能代码，Modbus_Slave 指令将生成错误代码 16#8188。

指令版本 V3.0 中 Modbus 从站的变量

下表显示了可在程序中使用的 Modbus_Slave 背景数据块中的公共静态变量。

表格 5- 21　Modbus 从站的变量

变量	数据类型	标准	说明
HR_Start_Offset	Word	0	为 Modbus 保持寄存器指定起始地址（默认 = 0）
QB_Start	Word	0	输出的有效可写入寻址范围起始地址（字节 0 到 65535） 注： 该变量不适用于 S7-300、S7-400 和 WinAC。

变量	数据类型	标准	说明
QB_Count	Word	0xFFFF	可由 Modbus 主站写入的输出字节数。 注： 该变量不适用于 S7-300、S7-400 和 WinAC。
Extended_Addressing	Bool	FALSE	扩展寻址，将从站寻址组态为单字节或双字节。 （FALSE = 单字节地址，TRUE = 双字节地址）
Request_Count	Word	0	该从站接收的所有请求的数量
Slave_Message_Count	Word	0	该特定从站接收的所有请求的数量
Bad_CRC_Count	Word	0	存在 CRC 错误的已接收请求的数量
Broadcast_Count	Word	0	已接收的广播请求的数量
Exception_Count	Word	0	使用主站的例外进行确认的 Modbus 特定错误
Success_Count	Word	0	该特定从站接收的无协议错误的请求数量
MB_DB	MB_BASE	-	Modbus_Comm_Load 指令的 MB_DB 参数必须连接到 Modbus_Master 指令的此 MB_DB 参数。

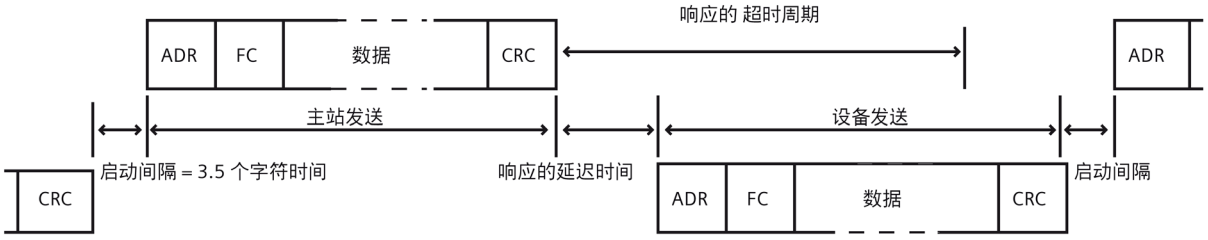
程序可以向 HR_Start_Offset 和 Extended_Addressing 变量中写入值来控制 Modbus 从站的操作。可读取其它变量以监视 Modbus 状态。

Modbus 从站通信的规则

- 必须运行 Modbus_Comm_Load 以组态端口，以便 Modbus_Slave 指令可以通过该端口进行通信。
- 如果端口作为从站响应 Modbus 主站，则不能使用 Modbus_Master 指令对该端口进行编程。
- 只有 Modbus_Slave 的一个实例可与特定端口一起使用；否则可能遇到意外行为。
- Modbus 指令不会使用通信报警事件来控制通信过程。为实现完整的发送和接收过程，程序必须通过查询 Modbus_Slave 指令来控制通信过程。
- 必须以允许及时响应 Modbus 主站进入请求的频率，定期执行 Modbus_Slave 指令。建议在每个来自程序周期 OB 的周期内执行 Modbus_Slave 。Modbus_Slave 可在周期性中断 OB 中执行，但不建议这样做，因为中断程序中过长的时间延迟会临时地阻碍其它中断程序的执行。

Modbus 信号的时间控制

必须定期执行 Modbus_Slave 来接收 Modbus 主站的每个请求并进行相应响应。执行 Modbus_Slave 的频率取决于由 Modbus 主站指定的响应超时值。下图中显示了这点。



(RESP_TO) 响应的超时周期为 Modbus 主站等待 Modbus 从站开始回答的持续时间。此周期不是由 Modbus 协议定义，而是由 Modbus_Comm_Load 指令的参数定义。由于发送和接收帧都需要多次调用 Modbus_Slave 指令（至少三次），因此应在超时周期内为 Modbus 主站的响应执行至少十二次 Modbus_Slave 指令，以便 Modbus 从站能按超时周期的规定执行两次接收和发送数据操作。

HR_Start_Offset

Modbus 保持寄存器的地址从 40001 或 400001 开始。这些地址与目标系统存储器中保持寄存器的起始地址相对应。但可以组态 HR_Start_Offset 变量来为 Modbus 保持寄存器组态不同于 40001 或 400001 的起始地址。

接收帧中的地址 0 与目标系统存储器中保持寄存器的起始地址相对应。使用 HR_Start_Offset 变量为 Modbus 保持寄存器组态 0 之外的起始地址。

例如，可以组态从 MW100 开始、长度为 100 字的保持寄存器。如果 HR_Start_Offset = 20，接收帧中的地址 20 与目标存储器 (MW100) 中保持寄存器的起始地址相对应。接收帧中低于 20 和高于 119 的各个地址将导致寻址错误。

表格 5- 22　当 DATA_PTR 为 MW100 的指针时（长度为 100 字）对 Modbus 保持寄存器寻址的示例

HR_Start_Offset	地址	最小	最大
0	Modbus 地址（字）	0	99
	S7-1500 地址	MW100	MW298
20	Modbus 地址（字）	20	119
	S7-1500 地址	MW100	MW298

HR_Start_Offset 为字的值，用于指定 Modbus 保持寄存器的起始地址，保存在 Modbus_Slave 背景数据块中。向程序中添加 Modbus_Slave 指令后，即可通过参数下拉列表选择公共静态变量。

例如，如果已经向 LAD 程序段中添加 Modbus_Slave 指令，则可以使用移动命令转至先前的程序段并分配值 HR_Start_Offset。必须在执行 Modbus_Slave 之前分配该值。

使用标准 DB 名称输入 Modbus 从站变量：

1. 将光标置于 OUT1 参数字段中并输入字符 m。
2. 从下拉列表中选择 Modbus_Slave 指令所需的背景数据块。
3. 将光标置于 DB 名称右侧（引号后面），并输入一个点。
4. 在下拉列表中选择"Modbus_Slave_DB.HR_Start_Offset"。

指令版本

版本 4.0 的功能与版本 3.0 完全相同，本次版本升级仅仅体现在内部措施方面。

自版本 V4.0 起访问 DB 中的数据区域而不直接访问 MODBUS 地址

自 Modbus_Slave 版本 V4.0 以及固件版本 V2.5 (S7-1500 CPU) 或 V4.2 (S7-1200 CPU) 起，可以访问 DB 中的数据区域而不直接访问过程映像和保持寄存器。为此，必须禁用 DB 的"优化块访问"(Optimized block access) 属性，并确保 DB 不会单独存在于加载存储区中。

如果 MODBUS 请求到达时尚未定义相应功能代码的 MODBUS 数据类型的数据区域，请求会按之前的指令版本处理，即直接访问过程映像和保持性寄存器。

但是，如果已定义功能代码的 MODBUS 数据类型的数据区域，则 Modbus_Slave 指令会对此数据区域执行读写操作。具体是读操作还是写操作取决于作业类型。

单个 MODBUS 请求只能对一个数据区域进行读写操作。如果要读取覆盖多个数据区域的保持性寄存器，则需要多个 MODBUS 请求。

数据区域定义规则

用户最多可在不同数据块中定义八个数据区域，每个数据块只能包含一个数据区域。单个 MODBUS 请求只能对恰好一个数据区域进行读写操作。每个数据区域对应于一个 MODBUS 地址区域。数据区域用背景数据块的静态变量 Data_Area_Array 定义；Data_Area_Array 是一个包含八个元素的字段。

如果要使用的数据区域不到八个，则所需数据区域必须紧密相连，没有间隙。在处理过程中，数据区域中的第一个空白条目会终止数据区域搜索。如果已定义字段元素 1、2、4 和 5，由于字段元素 3 留空，则只会识别字段元素 1 和 2。

Data_Area_Array 字段包含 8 个元素：Data_Area_Array[1] 到 Data_Area_Array[8]

每个字段元素 Data_Area_Array[x]（其中 1 <= x <= 8）都是 MB_DataArea 类型的 UDT，其结构如下：

参数	数据类型	含义
Data_type	UInt	映射到此数据区域的 MODBUS 数据类型的标识符： • 0：空字段元素或未使用数据区域的标识符。此时，db、start 和 length 的值不相关。 • 1：过程映像输出（与功能代码 1、5 和 15 一起使用） • 2：过程映像输入（与功能代码 2 一起使用） • 3：保持性寄存器（与功能代码 3、6 和 16 一起使用） • 4：输入寄存器（与功能代码 4 一起使用） 注：如果已定义 MODBUS 数据类型的数据区域，则指令 MB_SERVER 不能再直接访问此 MODBUS 数据类型。如果该数据类型的 MODBUS 请求的地址与定义的数据区域不对应，则 STATUS 中会返回一个值 W#16#8383。
db	UInt	后续定义的 MODBUS 寄存器或位所映射的目标数据块的编号。 数据块编号在数据区域中必须是唯一的。不得在多个数据区域中定义相同的数据块编号。 数据块必须支持标准访问，并且不得单独存在于加载存储器中。 数据区域也是从数据块的字节地址 0 开始。 允许值：1 到 60999
start	UInt	映射到数据块中的首个 MODBUS 地址（从地址 0.0 开始）。 允许值：0 到 65535
length	UInt	位数（对于 data_type 的值 1 和 2）或寄存器数量（对于 data_type 的值 3 和 4）。 相同 MODBUS 数据类型的 MODBUS 地址区域不得重叠。 允许值：1 到 65535

数据区域定义示例

- 第一个示例：data_type = 3，db = 1，start = 10，length = 6

  保持性寄存器 (data_type = 3) 映射在数据块 1 (db = 1)。Modbus 地址 10 (start = 10) 位于数据字 0。最后有效的 Modbus 地址 15 (length = 6) 位于数据字 5。

- 第二个示例：data_type = 2，db = 15，start = 1700，length = 112

  输入 (data_type = 2) 映射在数据块 15 (db = 15)。Modbus 地址 1700 (start = 1700) 位于数据字 0。最后有效的 Modbus 地址 1811 (length = 112) 位于数据字 111。

自版本 V4.0 起的过程映像读访问限制

过程映像读访问限制

自 Modbus_Slave 的指令版本 V4.0 起，可以在输入的过程映像和输出的过程映像中分别定义一个远程 MODBUS 设备有权读取的区域。随后，远程 MODBUS 设备便不能对超出这些过程映像区域的地址进行读访问。

说明

过程映像写访问限制

自指令版本 V3.0 起，可以选择将对输出的过程映像的写访问限制到一个特定区域。

过程映像中读取区域的定义

背景数据块的以下静态变量定义了过程映像中的读取区域：

- QB_Read_Start：可由远程 MODBUS 设备读取的过程映像输出中的第一个字节的地址（应用于功能代码 1）
- QB_Read_Count：可由远程 MODBUS 设备读取的过程映像输出中的字节数（应用于功能代码 1）
- IB_Read_Start：可由远程 MODBUS 设备读取的过程映像输入中的第一个字节的地址（应用于功能代码 2 和 4）
- IB_Read_Count：可由远程 MODBUS 设备读取的过程映像输入中的字节数（应用于功能代码 2 和 4）

背景数据块中用于定义过程映像中的读写区域的静态变量

下表说明了在上述 Modbus_Slave 指令的实例 DB 中列出的静态变量，使用这些变量可以定义过程映像的读取区域。

为了保持完整性，自版本 V3.0 起用于定义过程映像（QB_Start 和 QB_Count）的写入区域的静态变量也有相关说明。

变量	数据类型	起始值
QB_Start	UInt	0
QB_Count	UInt	65535
QB_Read_Start	UInt	0
QB_Read_Count	UInt	65535
IB_Read_Start	UInt	0
IB_Read_Count	UInt	65535

5.4.2.6 帧结构

Extended_Addressing

按照关于 HR_Start_Offset 参考的说明访问 Extended_Addressing 变量，Extended_Addressing 变量为布尔值时除外。

如果 Extended_Adressing = FALSE，可组态单字节（Modbus 标准）或两个字节（Extended_Adressing = TRUE）来寻址 Modbus 从站。扩展寻址用于在单个网络中寻址超过 247 个设备。如果 Extended_Adressing = TRUE，最多可寻址 65535 个地址。以下示例显示了 Modbus 帧。

表格 5- 23　大小为一个字节的从站地址（字节 0）

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	
请求	从站地址	功能代码	起始地址		数据		
有效响应	从站地址	功能代码	长度	数据...			
错误消息	从站地址	0xxx	异常代码				

表格 5- 24　大小为两个字节的从站地址（字节 0 和字节 1）

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**
请求	从站地址		功能代码	起始地址		数据	
有效响应	从站地址		功能代码	长度	数据...		
错误消息	从站地址		0xxx	异常代码			

帧说明

主站和从站/从站和主站之间的数据通信从从站地址开始，接下来是功能代码。随后传输数据。数据字段的结构取决于使用的功能代码。帧的最后传送的是校验和 (CRC)。

有性能优化时的功能代码

激活性能优化选项后，所传输数据的组态限值存在限制。有关限制的更多信息，请参见“功能代码 (页 70)”部分。

功能代码 1 - 此功能允许读取各个输出位

表格 5- 25　FC 1 - 读取输出位

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5
查询	从站地址	函数代码 1	起始地址		输出数目	
有效响应	从站地址	函数代码 1	长度 1)	输出数据 3)		
错误消息	从站地址	0x81	异常代码 2)	---		

1) 长度：如果将输出数目除以 8 后产生余数，则字节数必须加 1。

2) E 代码：01 或 02 或 03 或 04

3) 输出数据可包含多个字节

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6
查询	从站地址		函数代码 1	起始地址		输出数目	
有效响应	从站地址		函数代码 1	长度 1	输出数据		
错误消息	从站地址		0x81	异常代码 2	---		

1 长度：如果将输出数目除以 8 后产生余数，则字节数必须加 1。

2 E 代码：01 或 02 或 03 或 04

3 输出数据可由多个字节组成

功能代码 2 - 此功能允许读取各个输入位

表格 5- 26　FC 2 - 读取输入位

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**
查询	从站地址	函数代码 2	起始地址		输入数目	
有效响应	从站地址	函数代码 2	长度 [1]	输入数据		
错误消息	从站地址	0x82	异常代码 [2]	---		

[1] 长度：如果将输入数目除以 8 后产生余数，则字节数必须加 1。

[2] E 代码：01 或 02 或 03 或 04

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**
查询	从站地址		函数代码 2	起始地址		输入数目	
有效响应	从站地址		函数代码 2	长度 [1]	输入数据		
错误消息	从站地址		0x82	异常代码 [2]	---		

[1] 长度：如果将输入数目除以 8 后产生余数，则字节数必须加 1。

[2] E 代码：01 或 02 或 03 或 04

功能代码 3 - 此功能允许读取各个寄存器

表格 5- 27　FC 3 - 读取保持寄存器

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**
查询	从站地址	函数代码 3	起始地址		寄存器数	
有效响应	从站地址	函数代码 3	长度 [1]	寄存器数据		
错误消息	从站地址	0x83	异常代码 [2]	---		

[1] 长度：字节数

[2] E 代码：01 或 02 或 03 或 04

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**
查询	从站地址		函数代码 3	起始地址		寄存器数	
有效响应	从站地址		函数代码 3	长度 [1]	寄存器数据		
错误消息	从站地址		0x83	异常代码 [2]	---		

[1] 长度：字节数

[2] E 代码：01 或 02 或 03 或 04

功能代码 4 - 此功能允许读取各个寄存器

表格 5- 28　FC 4 - 读取输入字

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**
查询	从站地址	函数代码 4	起始地址		输入字数目：	
有效响应	从站地址	函数代码 4	长度 [1]	输入数据		
错误消息	从站地址	0x84	异常代码 [2]	---		

[1] 长度：2 * 输入字数目

[2] E 代码：01 或 02 或 03 或 04

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**
查询	从站地址		函数代码 4	起始地址		输入字数目：	
有效响应	从站地址		函数代码 4	长度 [1]	输入数据		
错误消息	从站地址		0x84	异常代码 [2]	---		

[1] 长度：2 * 输入字数目

[2] E 代码：01 或 02 或 03 或 04

功能代码 5 - 此功能可以设置或删除各个位

表格 5- 29　FC 5 - 写入输出位

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5
查询	从站地址	函数代码 5	起始地址		值	
有效响应	从站地址	函数代码 5	长度	值		
错误消息	从站地址	0x85	异常代码 [1]	---		

[1] E 代码：01 或 02 或 03 或 04

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6
查询	从站地址		函数代码 5	起始地址		值	
有效响应	从站地址		函数代码 5	长度	值		
错误消息	从站地址		0x85	异常代码 [1]	---		

[1] E 代码：01 或 02 或 03 或 04

功能代码 6 - 此功能允许写入各个寄存器

表格 5- 30　FC 6 - 写入保持寄存器

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5
查询	从站地址	函数代码 6	地址		寄存器	
有效响应	从站地址	函数代码 6	地址		寄存器	
错误消息	从站地址	0x86	异常代码 [1]	---		

[1] E 代码：01 或 02 或 03 或 04

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6
查询	从站地址		函数代码 6	地址		寄存器	
有效响应	从站地址		函数代码 6	地址		寄存器	
错误消息	从站地址		0x86	异常代码 [1]	---		

[1] E 代码：01 或 02 或 03 或 04

功能代码 8 - 此功能用于检查通信连接

表格 5- 31　FC 8 - 从站状态

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5
查询	从站地址	函数代码 8	诊断代码		测试值	
有效响应	从站地址	函数代码 8	诊断代码		测试值	
错误消息	从站地址	0x88	异常代码 [1]	---		

[1] E 代码：01 或 03 或 04

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6
查询	从站地址		函数代码 8	诊断代码		测试值	
有效响应	从站地址		函数代码 8	诊断代码		测试值	
错误消息	从站地址		0x88	异常代码 [1]	---		

[1] E 代码：01 或 03 或 04

功能代码 11 - 此功能可以读取 2 个字节的"状态字"和 2 个字节的"事件计数器"

表格 5- 32　FC 11 - 从站通信用事件计数器

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5
查询	从站地址	函数代码 11	---			
有效响应	从站地址	函数代码 11	状态		事件计数器	
错误消息	从站地址	0x8B	异常代码 [1]	---		

[1] E 代码：01 或 04

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6
查询	从站地址		函数代码 11	---			
有效响应	从站地址		函数代码 11	状态		事件计数器	
错误消息	从站地址		0x8B	异常代码 [1]	---		

[1] E 代码：01 或 04

功能代码 15 - 此功能允许写入多个位

表格 5- 33　FC 15 - 写入一个/多个输出位

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**	**字节 7**	**字节 n**
查询	从站地址	函数代码 15	起始地址		输出字数目		字节计数器 [1]	值	
有效响应	从站地址	函数代码 15	起始地址		输出字数目		---		
错误消息	从站地址	0x8F	异常代码 [2]	---					

[1] 字节计数器：如果将字节数除以 8 后产生余数，则字节数必须加 1。

[2] E 代码：01 或 02 或 03 或 04

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**	**字节 7**	**字节 8**	**字节 n**
查询	从站地址		函数代码 15	起始地址		输出字数目		字节计数器 [1]	值	
有效响应	从站地址		函数代码 15	起始地址		输出字数目		---		
错误消息	从站地址		0x8F	异常代码 [2]		---				

[1] 字节计数器：如果将字节数除以 8 后产生余数，则字节数必须加 1。

[2] E 代码：01 或 02 或 03 或 04

功能代码 16 - 此功能允许写入单个寄存器或多个寄存器

表格 5- 34　FC 16 - 写入一个/多个保持寄存器

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**	**字节 7**	**字节 n**
查询	从站地址	函数代码 16	起始地址		寄存器数		字节计数器 [1]	值	
有效响应	从站地址	函数代码 16	起始地址		寄存器数		---		
错误消息	从站地址	0x90	异常代码 [2]	---					

[1] 字节计数器：寄存器数 * 2

[2] E 代码：01 或 02 或 03 或 04

	字节 0	**字节 1**	**字节 2**	**字节 3**	**字节 4**	**字节 5**	**字节 6**	**字节 7**	**字节 8**	**字节 n**
查询	从站地址		函数代码 16	起始地址		寄存器数		字节计数器 [1]	值	
有效响应	从站地址		函数代码 16	起始地址		寄存器数		---		
错误消息	从站地址		0x90	异常代码 [2]		---				

[1] 字节计数器：寄存器数 * 2

[2] E 代码：01 或 02 或 03 或 04

5.4.2.7 错误消息

Modbus 错误消息概述

错误代码	说明	解决方案
16#0000	无错误	-
接口组态错误 - Modbus_Comm_Load		
16#8181	该模块不支持此数据传输速率。	在 BAUD 参数上为该模块选择有效的数据传输速率。
16#8182	该模块不支持此奇偶校验设置。	在 PARITY 参数上为“奇偶校验”(Parity) 选择合适的值。 以下内容有效： • 无 (1) • 偶校验 (2) • 奇校验 (3) • 标记校验 (4) • 间隔校验 (5) • 任意 (6)
16#8183	该模块不支持此数据流控制类型。	在 FLOW_CTRL 参数上为该模块选择有效的数据流控制。
16#8184	“响应超时”(Response timeout) 值无效。	在 RESP_TO 参数上为“响应超时”(Response timeout) 选择合适的值。 数值的有效范围：1 到 65535 (ms)
16#8280	读取模块时进行否定确认	检查 PORT 参数中的输入。 在 Send_Config.RDREC.STATUS 或 Receive_Config.RDREC.STATUS 静态参数、RDREC.STATUS 和 SFB RDREC 的说明中会找到有关错误原因的更多详细信息。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。

错误代码	说明	解决方案
16#8281	写入模块时进行否定确认	检查 PORT 参数中的输入。 在 Send_Config.WRREC.STATUS 或 Receive_Config.WRREC.STATUS 静态参数、WRREC.STATUS 和 SFB WRREC 的说明中会找到有关错误原因的更多详细信息。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。
16#8282	模块不可用	检查 PORT 参数中的输入并确保模块可以访问。
组态错误 - Modbus_Slave		
16#8186	从站地址无效	在 MB_ADDR 参数上选择合适的从站地址。 以下内容有效：标准地址区上的 1-247； 扩展地址区上的 1-65535 （为广播保留 0）
16#8187	MB_HOLD_REG 参数上的值无效	在 MB_HOLD_REG 参数上为保持寄存器选择合适的值。
16#8188	无效操作模式或广播 (MB_ADDR = 0) 和 MODE 参数模式 ≠ 1	在广播模式下为 MODE 选择值 1 或选择不同的操作模式。
16#818C	指向 MB_HOLD_REG 区的指针必须是数据块或位存储器地址区。	为指向 MB_HOLD_REG 区的指针选择合适的值。
16#8280	读取模块时进行否定确认	检查 PORT 参数中的输入。 在 Send_P2P.RDREC.STATUS 或 Receive_P2P.RDREC.STATUS 静态参数和 SFB RDREC 的说明中可找到有关错误原因的更多详细信息。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。

错误代码	说明	解决方案
16#8281	写入模块时进行否定确认	检查 PORT 参数中的输入。 在 Send_P2P.WRREC.STATUS 或 Receive_P2P.WRREC.STATUS 静态参数和 SFB WRREC 的说明中会找到有关错误原因的更多详细信息。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。
16#8389	数据区定义无效： • data_type 值非法 • 数据块编号不被允许或不可用： – 数据块的值无效 – 数据块编号不存在 – 数据块编号已被另一个数据区占用 – 优化了访问权限的数据块 – 数据块不在工作存储器中 • length 值非法 • 属于同一种 MODBUS 数据类型的多个 MODBUS 地址区域重叠	检查数据区的定义。 请参见“自版本 V4.0 起访问 DB 中的数据区域而不直接访问 MODBUS 地址 (页 161)”部分。
16#8452 1)	MB_HOLD_REG 不是指向数据块或位存储区的指针	检查 MB_HOLD_REG 指针
16#8453 1)	MB_HOLD_REG 并非 BOOL 或 WORD 类型的指针	检查 MB_HOLD_REG 指针
16#8454 1)	MB_HOLD_REG 访问的区域长度大于数据块，或者所访问的区域对于要读取或写入的数据字节来说太小。	检查 MB_HOLD_REG 指针
16#8455 1)	MB_HOLD_REG 指向具有写保护的数据块	检查 MB_HOLD_REG 指针
16#8456 1)	指令执行期间出错。错误原因显示在参数 STATUS 中。	确定 SFCSTATUS 参数的值。在参数 SFC51 和 STATUS 的说明中检查其含义。

错误代码	说明	解决方案
组态错误 - Modbus_Master		
16#8180	MB_DB 参数的值无效	在 Modbus_Comm_Load 指令上为 MB_DB（背景数据块）组态的值无效。 检查 Modbus_Comm_Load 指令和其错误消息的互连情况。
16#8186	无效站地址	在 MB_ADDR 参数上选择合适的站地址。 以下内容有效：标准地址区上的 1-247； 扩展地址区上的 1-65535 （为广播保留 0）
16#8188	无效操作模式或广播 (MB_ADDR = 0) 和 MODE 参数模式 ≠ 1	在广播模式下为 MODE 选择值 1 或选择不同的操作模式。
16#8189	无效数据地址	在 DATA_ADDR 参数上为数据地址选择合适的值。 请参见信息系统中的 Modbus_Master (页 147) 说明
16#818A	无效长度	在 DATA_LEN 参数上选择合适的数据长度。 请参见信息系统中的 Modbus_Master (页 147) 说明
16#818B	DATA_PTR 的值无效	在 DATA_PTR 参数（M 或 DB 地址）上为数据指针选择合适的值。 请参见信息系统中的 Modbus_Master (页 147) 说明
16#818C	DATA_PTR 参数的互连错误	检查指令的互连。
16#818D	DATA_PTR 访问的区域长度大于数据块，或者所访问的区域对于要读取或写入的数据字节来说太小。	检查 DATA_PTR 指针

错误代码	说明	解决方案
16#8280	读取模块时进行否定确认	检查 PORT 参数中的输入。 在 Send_P2P.RDREC.STATUS 或 Receive_P2P.RDREC.STATUS 静态参数和 SFB RDREC 的说明中可找到有关错误原因的更多详细信息。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。
16#8281	写入模块时进行否定确认	检查 PORT 参数中的输入. 在静态参数 Send_P2P.WRREC.STATUS、Receive_P2P.WRREC.STATUS 或 Receive_Reset 和 SFB WRREC 的说明中会找到有关错误原因的更多详细信息。 有关错误代码的更多信息，请参见 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109815286) 中的 FAQ，条目 ID 为 109815286。
通信错误 - Modbus_Master 和 Modbus_Slave		
16#80D1	XON 或 CTS = ON 的等待时间已结束。	通信伙伴有故障、太慢或已离线。检查通信伙伴，或在需要时更改参数。
16#80D2	“硬件 RTS 始终开启”(Hardware RTS always ON)：发送作业因从 DSR = ON 更改为 DSR = OFF 而取消	检查通信伙伴。确保 DSR 在整个传输持续期间内均保持为 ON。
16#80E0	帧已中止：发送缓冲区上溢/发送帧太长	在用户程序中更频繁调用指令或者利用数据流控制组态通信。
16#80E1	帧已中止：奇偶校验错误	检查通信伙伴的连接线路，或确认两台设备是否针对相同的数据传输速率、奇偶校验和结束位数进行了组态。
16#80E2	帧已中止：字符帧错误	检查起始位、数据位、奇偶校验位、数据传输速率和结束位的设置。
16#80E3	帧已中止：字符上溢错误	检查通信伙伴的帧中的数据个数。

错误代码	说明	解决方案
16#80E4	帧已中止：达到最大帧长度	在通信伙伴上选择较短的帧长度。 以下内容有效（取决于模块）：1-1024/2048/4096（字节）
通信错误 - Modbus_Master		
16#80C8	从站在设置时间内未响应	检查数据传输率、奇偶校验和从站的接线情况。
16#80C9	从站未在通过 Blocked_Proc_Timeout 设置的时间内做出响应。	检查 Blocked_Proc_Timeout 的设置。 检查是否已使用 Modbus_Comm_Load 指令组态模块。插拔后或恢复电压后，可能需要使用 Modbus_Comm_Load 重新组态模块。
16#8200	接口处于连续请求中。	稍后重复该命令。开始新的命令前，确保没有正在运行中的命令。
协议错误 - Modbus_Slave（仅限支持 Modbus 的通信模块）		
16#8380	CRC 错误	Modbus 帧的校验和错误。检查通信伙伴。
16#8381	不支持功能代码或广播中不支持功能代码。	检查通信伙伴，确保有效功能代码已发送。
16#8382	请求帧中的长度信息无效	在 DATA_LEN 参数上选择合适的数据长度。
16#8383	请求帧中的数据地址无效	在 DATA_ADDR 参数上为数据地址选择合适的值。
16#8384	请求帧中的数据值无效	检查 Modbus 主站的请求帧中的数据值
16#8385	Modbus 从站不支持诊断值 （功能代码 08）	Modbus 从站仅支持诊断值 16#0000 和 16#000A。
协议错误 - Modbus_Master（仅限支持 Modbus 的通信模块）		
16#8380	CRC 错误	Modbus 帧的校验和错误。检查通信伙伴。
16#8381	来自 Modbus 从站的响应帧有下列错误消息：不支持功能代码。	检查通信伙伴，确保有效功能代码已发送。
16#8382	来自 Modbus 从站的响应帧有下列错误消息：无效长度	选择合适的数据长度。
16#8383	来自 Modbus 从站的响应帧有下列错误消息：请求帧中的数据地址无效	在 DATA_ADDR 参数上为数据地址选择合适的值。
16#8384	来自 Modbus 从站的响应帧有下列错误消息：数据值错误	检查发送到 Modbus 从站的请求帧。

错误代码	说明	解决方案
16#8385	来自 Modbus 从站的响应帧有下列错误消息：Modbus 从站不支持诊断值	Modbus 从站仅支持诊断值 16#0000 和 16#000A。
16#8386	返回的功能代码与请求的功能代码不匹配。	检查从站的响应帧和地址。
16#8387	从站未发出请求的应答	检查设备的响应帧。检查从站的地址设置。
16#8388	从站对写入请求的响应出现错误。	检查从站的响应帧。
16#8828 [1)]	DATA_PTR 指向的位地址不等于 n * 8	检查 DATA_PTR 指针
16#8852 [1)]	DATA_PTR 不是指向数据块或位存储区的指针	检查 DATA_PTR 指针
16#8853 [1)]	DATA_PTR 并非 BOOL 或 WORD 类型的指针	检查 DATA_PTR 指针
16#8855 [1)]	DATA_PTR 指向具有写保护的数据块	检查 DATA_PTR 指针
16#8856 [1)]	调用 SFC51 时出错	再次调用 Modbus_Master 指令
错误 - Modbus_Slave（仅限支持 Modbus 的通信模块）		
16#8428 [1)]	MB_HOLD_REG 指向的位地址不等于 n * 8	检查 MB_HOLD_REG 指针
16#8452 [1)]	MB_HOLD_REG 不是指向数据块或位存储区的指针	检查 MB_HOLD_REG 指针
16#8453 [1)]	MB_HOLD_REG 并非 BOOL 或 WORD 类型的指针	检查 MB_HOLD_REG 指针
16#8454 [1)]	MB_HOLD_REG 访问的区域长度大于数据块，或者所访问的区域对于要读取或写入的数据字节来说太小。	检查 MB_HOLD_REG 指针
16#8455 [1)]	MB_HOLD_REG 指向具有写保护的数据块	检查 MB_HOLD_REG 指针
16#8456 [1)]	调用 SFC51 时出错	再次调用 Modbus_Slave 指令
错误代码，一般		
16#8FFF	模块因复位而暂时未准备就绪。	重复请求。

[1)] 仅限 S7-300/400 CPU 的指令

5.4.3 USS

5.4.3.1 库版本间的依赖性

必须按照下列一一对应的组合关系来使用"USS"和"点对点"指令库：

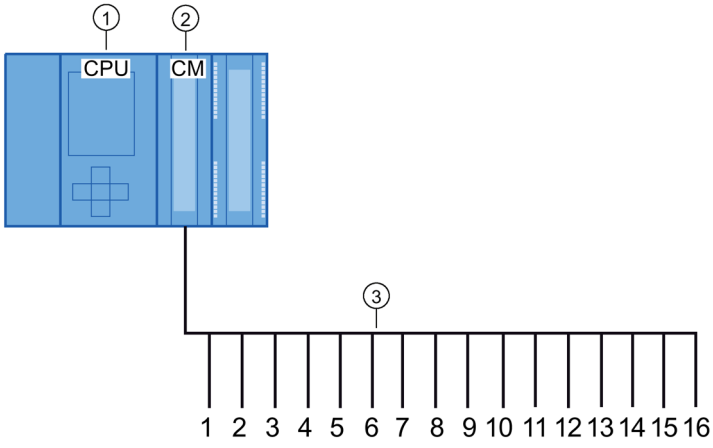
"USS"库版本	**"点对点"库版本**
V1.3	V1.1
V2.4	V2.4
V3.1	V2.4
V4.3	V3.2
V5.0	V4.0
V5.1	V4.1

5.4.3.2 USS 通信概述

USS 通信

USS 指令控制支持通用串行接口协议 (USS) 的变频器运行。可通过 PtP 通信模块的 RS485 连接和 USS 指令与多个变频器通信。通常，每个 RS 485 端口最多可运行 16 个驱动器。一些通信模块甚至最多可运行 31 个驱动器。

USS 协议使用主从网络通过串行总线进行通信。主站使用地址参数将数据发送到所选从站。未先收到发送请求时从站不能发送。各从站之间无法通信。USS 通信在半双工模式下进行。下图显示具有 16 台变频器的示例应用网络图。



① CPU
② CM
③ USS 网络中的 USS 变频器

图 5-1 S7-1500 通信模块接线示例

说明

通过 RS232 与变频器通信

基本上还可以使用 CM PtP RS232 BA 和 CM PtP RS232 HF 与变频器通信。但只可将一台变频器连接到 RS232 端口。

通过 RS422 与变频器通信

基本上还可以使用 CM PtP RS422/485 BA 和 CM PtP RS422/485 HF 的 RS422 接口与变频器通信。但只可将一台变频器连接到 RS422 端口。

程序中的 USS 指令

- USS_Port_Scan：USS_Port_Scan 指令允许在 USS 网络中通过一个通信模块与最多 16 个驱动器进行通信（必须循环调用）。

  程序中每个 PtP 通信端口只有一条 USS_Port_Scan 指令，并且该指令控制发往所有变频器的传输。

- USS_Drive_Control：USS_Drive_Control 指令允许从 USS_Port_Scan 中为驱动器准备发送数据并显示其接收数据。

  USS_Drive_Control 组态要发送的数据并评估在前一请求中从 USS_Port_Scan 收到的数据。

- USS_Read_Param：USS_Read_Param 指令允许从驱动器中读取参数。
- USS_Write_Param：USS_Write_Param 指令允许用户更改驱动器中的参数。

5.4.3.3 USS 协议使用要求

四条 USS 指令使用 2 个 FB 和 2 个 FC 支持 USS 协议。对于每个 USS 网络，一个背景数据块 (DB) 用于 USS_Port_Scan，一个背景数据块用于 USS_Drive_Conrol 的所有调用。

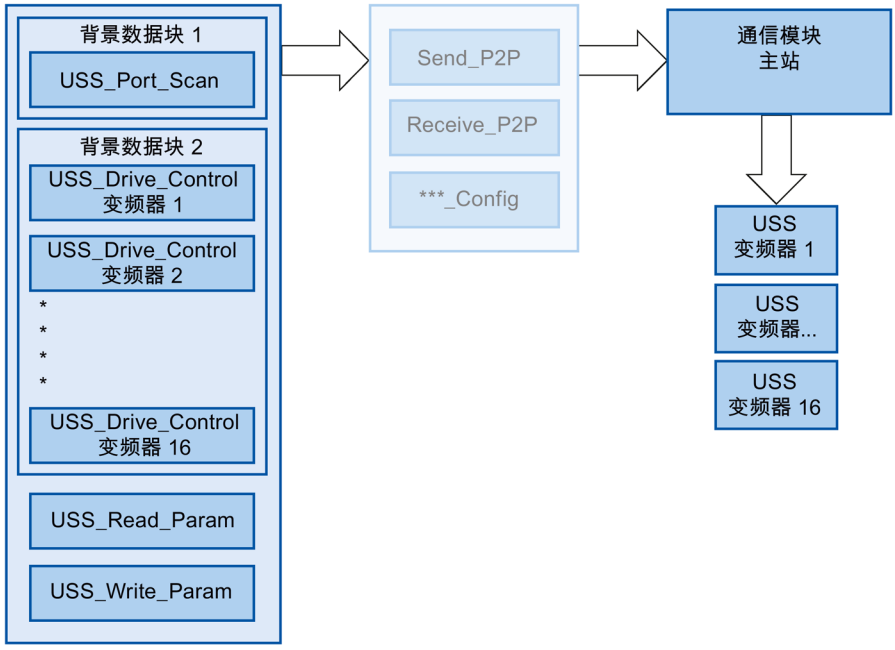


图 5-2 USS 程序顺序

连接到一个 RS485 端口的所有变频器（最多 16 个）都属于同一 USS 网络。连接到另一个 RS485 端口的所有变频器都属于其它 USS 网络。对于所有 USS_Drive_Control 指令，每个 USS 网络都通过唯一的背景数据块进行管理，对于 USS_Port_Scan 指令，则加上另一个背景数据模块。对于 USS_Drive_Conrol，属于 USS 网络的所有指令都必须共享此背景数据块。USS_Port_Scan, USS_Read_Param 和 USS_Write_Param 指令具有该函数的 USS_DB 参数。该参数必须连接到 USS_Drive_Control 指令的背景数据块的（静态）USS_DB 参数。

- 指令 USS_Drive_Control 和 USS_Port_Scan 是函数块 (FB)。如果向程序编辑器中添加 USS_Drive_Control 或 USS_Port_Scan 指令，“调用选项”(Call options) 对话框将提示您为此 FB 分配 DB。如果它是此程序中此 USS 网络的第一条 USS_Drive_Control 指令，则可应用 DB 标准分配（必要时也可更改名称），并会为您创建新 DB。但如它不是此变频器的第一条 USS_Drive_Control 指令，则必须在“调用选项”(Call options) 对话框的下拉菜单中选择已分配给此 USS 网络的 DB。
- 指令 USS_Write_Param 和 USS_Read_Param 是函数 (FC)。在编辑器中添加这些 FC 时不会分配 DB。如果在编辑器中添加这些 FC 或 USS_Port_Scan 指令，则需要将相应 USS_Drive_Control 背景数据块的 USS_DB 参数分配给这些指令的 USS_DB 输入。双击参数字段，然后单击符号显示可用 DB。输入一个句点“.”并从下拉列表中选择 USS_DB 参数。
- USS_Port_Scan 函数通过点对点 RS485 通信端口控制 CPU 与变频器之间的通信。每次调用此功能时，将进行与变频器之间的通信。程序必须快速调用此函数，以使变频器不发出超时信号。为确保帧通信的响应时间恒定，应在循环中断 OB 中调用该指令。
- USS_Drive_Control 指令使程序能够访问 USS 网络中的指定变频器。其输入和输出对应于变频器的状态和运行功能。如果网络中有 16 台变频器，在程序中必须至少调用 USS_Drive_Control 16 次，即每次一台变频器。

  只能从循环 OB 中调用 USS_Drive_Control 指令。
- USS_Read_Param 和 USS_Write_Param 函数用于读写变频器的操作参数。这些参数控制变频器内部运行。有关这些参数的定义，请参见变频器手册。程序可能包括任意多个此类函数，但在任意时刻一台变频器都只能激活一个读取或写入请求。只可从主程序的循环 OB 调用 USS_Read_Param 和 USS_Write_Param 函数。

注意
USS 指令调用 只从主程序的循环 OB 调用 USS_Drive_Control, USS_Read_Param 和 USS_Write_Param 。可从任何 OB 调用 USS_Port_Scan 指令函数，但通常从循环中断 OB 调用。 不要在优先级比 USS_Port_Scan 指令所在 OB 的优先级高的 OB 中使用 USS_Drive_Control, USS_Read_Param 或 USS_Write_Param 指令。例如，不要向主程序中添加 USS_Port_Scan 或向循环中断 OB 中添加 USS_Read_Param 。如果其它指令中断了 USS_Port_Scan 的执行，可能会发生意外错误。

说明

参数 ID 值

用户需要对变频器的 4 个 PIV 字 (ParameterIDValue) 的用途进行组态。

计算与变频器的通信时间

与变频器进行的通信与 CPU 的周期不同步。CPU 与变频器的通信完成前，通常会运行几个周期。

为确保不触发变频器的看门狗设置，必须在看门狗时间内向变频器发送帧。如果通信发生错误，用户必须允许多次重试来完成这一事务。默认情况下，使用 USS 协议时每个事务最多进行 2 次重试。
两次发送帧的最长时间间隔按如下公式计算：

N * (5 * 周期时间 + 帧运行时间 + 接收帧的最长超时) * (传送尝试次数)

N	该网络中的变频器数量
因数 5	发送和接收帧通常需要 5 个周期。
周期时间	调用 USS_Port_Scan 指令的循环中断 OB 的最大周期时间。
帧运行时间	帧运行时间 =（每帧的字符数）*（11 Bit/每字符）/（以 Bit/s 为单位的数据传输速率）
传输尝试次数	重试次数 + 1
接收帧的超时	RCVTIME（如果未收到驱动器的任何响应）
接收帧的最长超时	RCVTIME + MSGTIME（如果在 RCVTIME 快结束前收到不完整回复且 MSGTIME 的监视已过期，**或者**如果在 RCVTIME 过期后仍在处理响应，则超时将延迟 MSGTIME）

下列时间适用于"已接收帧的超时"(ms)：

比特/秒	**115200**	**57600**	**38400**	**19200**	**9600**	**4800**	**2400**	**1200**
Receive_Conditions .END.RCVTIME	25	29	33	56	72	100	100	100
Receive_Conditions .END.MSGTIME	25	29	33	56	72	124	240	460

接收帧的最长超时 = (Receive_Conditions.END.RCVTIME (0.072 s) + Receive_Conditions.END.MSGTIME (0.072 s))

示例：

5 个驱动器
数据传输速率 = 9600 bps
每帧 28 个字符
周期 = 0.020 s
重试次数 = 2

时间间隔 = 5 * ((5*0.02) + ((1*28*11)/9600) + 0.072 + 0.072) * (2+1) = 4.14（秒）

这种情况下，驱动器的监视时间必须设置为大约 4 秒。

说明

性能优化选项

性能优化选项激活时，两个发送报文之间的最大时间间隔由 OB 周期决定（Send_P2P 和 Receive_P2P），报文长度和波特率起决定性作用。

5.4.3.4 USS_Port_Scan / USS_Port_Scan_31：通过 USS 网络进行通信

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

使用 USS_Port_Scan_31 指令

只能在 S7-1500 CPU 上使用 USS_Port_Scan_31 指令。

说明

USS_Port_Scan 指令通过 USS 网络为最多 16 个变频器处理通信。

USS_Port_Scan_31 指令通过 USS 网络为最多 31 个变频器处理通信。

添加指令时 STEP 7 自动创建背景数据块。

参数

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
PORT	IN	Port	Word	0	指定用于通信的通信模块： • 对于 S7-1500/S7-1200：来自设备组态的"硬件标识符"。 符号端口名称在 PLC 变量表的"系统常量"(System constants) 选项卡中分配，可从此处进行应用。 • 对于 S7-300/S7-400：来自设备组态的"输入地址"。 在 S7-300/400/WinAC 系统中，硬件配置中分配的输入地址被分配给 PORT 参数。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
BAUD	IN	DInt		9600	USS 通信的数据传输速率 以下内容有效： • 1200 bps • 2400 bps • 4800 bps • 9600 bps • 19200 bps • 38400 bps • 57600 bps • 115200 bps
USS_DB	INOUT	USS_BASE		–	USS_DB 参数必须连接到背景数据块的（静态）USS_DB 参数，该参数是在向程序中添加 USS_Drive_Control / USS_Drive_Control_31 指令时生成并初始化的。
COM_RST	INOUT	---	Bool	FALSE	初始化 USS_Port_Scan / USS_Port_Scan_31 指令将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。 注： 该参数仅适用于 S7-300/400 指令。
ERROR	OUT	Bool		FALSE	如果为 TRUE，此输出表示发生错误且 STATUS 输出有效。 可能需要检查 USS_Drive_Control / USS_Drive_Control_31 指令的背景数据块中静态变量 USS_DB. w_USSExtendedError 的值。
STATUS	OUT	Word		0	错误代码（请参见错误消息 (页 204)）。

程序中每个 PtP 通信端口只有一条 USS_Port_Scan / USS_Port_Scan_31 指令，并且此指令的每个调用都控制往返于此网络中所有变频器的传输。分配给一个 USS 网络和一个 PtP 通信端口的所有 USS 函数都必须使用相同的背景数据块。

程序必须足够频繁地执行 USS_Port_Scan / USS_Port_Scan_31 指令，以防止变频器超时（请参见 USS 协议使用要求 (页 182)"计算与变频器通信的时间"）。

通常从循环中断 OB 调用 USS_Port_Scan / USS_Port_Scan_31 指令，以防变频器超时并使上次 USS 数据更新可用于调用 USS_Drive_Control / USS_Drive_Control_31 。

USS_Port_Scan / USS_Port_Scan_31 数据块变量

下表显示了可在程序中使用的 USS_Port_Scan / USS_Port_Scan_31 背景数据块中的公共静态变量。

表格 5- 35　背景数据块中的静态变量

变量	数据类型	标准	说明
MODE	USInt	4	工作模式 有效的工作模式包括： • 0 = 全双工 (RS232) • 1 = 全双工 (RS422) 四线制模式（点对点） • 2 = 全双工 (RS 422) 四线制模式（多点主站；CM PtP (ET 200SP)) • 3 = 全双工 (RS 422) 四线制模式（多点从站；CM PtP (ET 200SP)) • 4 = 半双工 (RS485) 二线制模式 [1)]
LINE_PRE	USInt	2	接收线路初始状态 有效的初始状态是： • 0 =”无”初始状态 [1)] • 1 = 信号 R(A)=5 V，信号 R(B)=0 V（断路检测）： 在此初始状态下，可进行断路检测。 仅可以选择以下项：”全双工 (RS422) 四线制模式（点对点连接）”和”全双工 (RS422) 四线制模式（多点从站）”。 • 2 = 信号 R(A)=0 V，信号 R(B)=5 V： 此默认设置对应于空闲状态（无激活的发送操作）。在此初始状态下，无法进行断路检测。
BRK_DET	USInt	0	激活诊断中断： • 0 - 未激活 • 1 - 已激活
RETRIES_MAX	SInt/Byte	2	发生通信错误时的重试次数。 在设定时间内未收到响应帧时，可使用此参数设置请求帧的重试次数。

变量	数据类型	标准	说明
EN_DIAG _ALARM	Bool	0	激活诊断中断： • 0 - 未激活 • 1 - 已激活
EN_SUPPLY _VOLT	Bool	0	启用对电源电压 L+ 缺失的诊断 • 0 - 未激活 • 1 - 已激活

[1] 使用 PROFIBUS 电缆连接 CM 1241 的 RS485 时所需的设置

版本 2.5 的功能与版本 2.4 完全相同，本次版本升级仅仅体现在内部措施方面。

指令版本

USS_Port_Scan:

版本 2.5 的功能与版本 2.4 完全相同，本次版本升级仅仅体现在内部措施方面。

USS_Port_Scan_31:

版本 1.2 的功能与版本 1.1 完全相同，本次版本升级仅仅体现在内部措施方面。

5.4.3.5 USS_Drive_Control / USS_Drive_Control_31：准备并显示变频器数据

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

使用 USS_Drive_Control_31 指令

只能在 S7-1500 CPU 上使用 USS_Drive_Control_31 指令。

说明

USS_Drive_Control 指令为最多 16 个驱动器准备发送数据并评估驱动器的响应数据。

USS_Drive_Control_31 指令为最多 31 个驱动器准备发送数据并评估驱动器的响应数据。

需要对每台变频器使用单独的指令实例，并且分配给一个 USS 网络和一个 PtP 通信端口的所有 USS 函数都必须使用同一背景数据块。在添加第一条 USS_Drive_Control / USS_Drive_Control_31 指令时必须输入 DB 名称。之后引用这个在添加第一条指令时创建的 DB。

添加指令时 STEP 7 自动创建 DB。

参数

参数	声明	数据类型		标准	说明
		S7-1200 /1500	S7-300/400/ WinAC		
RUN	IN	Bool		FALSE	变频器的起始位：如果此参数为 TRUE，则输入允许以预设速度运行变频器。如果在变频器运行期间 RUN 变为 FALSE，则电机滑行至静止。此行为不同于断开电源 (OFF2) 和电机制动 (OFF3)。
OFF2	IN	Bool		FALSE	“滑行至静止”位：如果此参数为 FALSE，此位会使变频器滑行至静止而不制动。
OFF3	IN	Bool		FALSE	快速停止位：如果此参数为 FALSE，此位通过制动变频器产生快速停止。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
F_ACK	IN	Bool		FALSE	错误确认位：此位复位变频器的错误位。清除错误后此位置位，变频器以此方式检测前一错误不必报告。
DIR	IN	Bool		FALSE	变频器方向控制：如果变频器正向运行，则此位置位（如果 SPEED_SP 为正值；请参见表"SPEED_SP 与 DIR 参数的交互"）。
DRIVE	IN	USInt	Byte	1	变频器地址：此输入是 USS 变频器的地址。有效范围是变频器 1 与变频器 16 之间。
PZD_LEN	IN	USInt	Byte	2	字长度：这是 PZD 数据字数。有效值为 2、4、6 或 8 个字。
SPEED_SP	IN	Real		0.0	速度设定值：这是组态频率百分比形式的变频器速度。正值表示变频器正向运行（如果 DIR 为 True）。有效值范围是 200.00 至 -200.00。
CTRL3	IN	Word		0	控制字 3：写入变频器用户定义参数的值。需要在变频器中对其进行组态（可选参数）。
CTRL4	IN	Word		0	控制字 4：写入变频器用户定义参数的值。需要在变频器中对其进行组态（可选参数）。
CTRL5	IN	Word		0	控制字 5：写入变频器用户定义参数的值。需要在变频器中对其进行组态（可选参数）。
CTRL6	IN	Word		0	控制字 6：写入变频器用户定义参数的值。需要在变频器中对其进行组态（可选参数）。
CTRL7	IN	Word		0	控制字 7：写入变频器用户定义参数的值。需要在变频器中对其进行组态（可选参数）。
CTRL8	IN	Word		0	控制字 8：写入变频器用户定义参数的值。需要在变频器中对其进行组态（可选参数）。
COM_RST	IN/OUT	---	Bool	FALSE	USS_Drive_Control / USS_Drive_Control_31 指令的初始化 将使用 TRUE 对指令进行初始化。随后会将 COM_RST 复位为 FALSE。 注： 该参数仅适用于 S7-300/400 指令。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
NDR	OUT	Bool		FALSE	可用的新数据：如果此参数为 TRUE，该位表示新通信请求数据可用于输出。
ERROR	OUT	Bool		FALSE	发生错误：如果为 TRUE，则表示发生错误且 STATUS 输出有效。出错时所有其它输出都置零。只在 USS_Port_Scan / USS_Port_Scan_31 指令的 ERROR 和 STATUS 输出发出通信错误信号。
STATUS	OUT	Word		0	错误代码（请参见错误消息 (页 204)）。
RUN_EN	OUT	Bool		FALSE	运行已启用：此位表示变频器是否在运行。
D_DIR	OUT	Bool		FALSE	变频器方向：此位表示变频器是否在正向运行。 • FALSE – 正向 • TRUE – 反向
INHIBIT	OUT	Bool		FALSE	变频器已禁止：此位表示变频器的禁止位状态。 • FALSE – 未禁止 • TRUE – 已禁止
FAULT	OUT	Bool		FALSE	变频器错误：此位表示变频器出现错误。必须修复错误并将 F_ACK 置位以将此位清零。
SPEED	OUT	Real		0.0	实际值变频器速度（变频器状态字 2 的换算值）：这是组态速度百分比形式的变频器速度。
STATUS1	OUT	Word		0	变频器状态字 1 此值包括变频器的固定状态位。
STATUS3	OUT	Word		0	变频器状态字 3 此值包括变频器的用户可定义状态字。
STATUS4	OUT	Word		0	变频器状态字 4 此值包括变频器的用户可定义状态字。
STATUS5	OUT	Word		0	变频器状态字 5 此值包括变频器的用户可定义状态字。
STATUS6	OUT	Word		0	变频器状态字 6 此值包括变频器的用户可定义状态字。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
STATUS7	OUT	Word		0	变频器状态字 7 此值包括变频器的用户可定义状态字。
STATUS8	OUT	Word		0	变频器状态字 8 此值包括变频器的用户可定义状态字。

最初执行 USS_Drive_Control / USS_Drive_Control_31 时，在背景数据块中初始化 USS 地址（DRIVE 参数）指定的变频器。初始化过后，USS_Port_Scan / USS_Port_Scan_31 指令可按此变频器编号开始与变频器进行通信。

如果更改变频器编号，必须先将 CPU 置于 STOP 模式然后再回到 RUN 模式才能初始化背景数据块。在 USS 发送缓冲区中组态输入参数，从"上一个"有效的响应缓冲区读取任意输出。USS_Drive_Control / USS_Drive_Control_31 只会组态要发送的数据并会评估上一个请求中收到的数据。

可使用 D_IR 输入 (Bool) 或对 SPEED_SP 输入 (Real) 使用符号（正或负）来控制变频器旋转方向。下表解释这些输入如何共同确定变频器旋转方向（假设电机正向旋转）。

表格 5- 36　SPEED_SP 与 DIR 参数的交互

SPEED_SP	**DIR**	**变频器旋转方向**
值 > 0	0	反向
值 > 0	1	正向
值 < 0	0	正向
值 < 0	1	反向

USS_Drive_Control / USS_Drive_Control_31 数据块变量

下表显示了可在程序中使用的 USS_Drive_Control / USS_Drive_Control_31 背景数据块中的公共静态变量。

表格 5- 37　背景数据块中的静态变量

变量	数据类型	标准	说明
USS_DB. W _USSExtendedError	Word	16#0	USS 驱动器的扩展错误代码 - 特定于驱动器的值 错误消息的含义取决于第一个报告错误的指令 (ERROR = TRUE)。区分为以下情况： • USS_Write_Param / USS_Write_Param_31：错误代码的含义可在变频器说明中找到。 • USS_Read_Param / USS_Read_Param_31：错误代码的含义可在变频器说明中找到。 • USS_Port_Scan / USS_Port_Scan_31：受错误消息影响的变频器的编号。

指令版本

USS_Drive_Control:

版本 2.0 的功能与版本 1.2 完全相同，本次版本升级仅仅体现在内部措施方面。

USS_Drive_Control_31:

版本 2.0 的功能与版本 1.0 完全相同，本次版本升级仅仅体现在内部措施方面。

5.4.3.6 USS_Read_Param / USS_Read_Param_31：从变频器读取数据

说明

使用 CM1241

自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

使用 USS_Read_Param_31 指令

只能在 S7-1500 CPU 上使用 USS_Read_Param_31 指令。

说明

USS_Read_Param 指令从最多 16 个变频器之一读取参数。

USS_Read_Param_31 指令从最多 31 个变频器之一读取参数。

分配给一个 USS 网络和一个 PtP 通信端口的所有 USS 函数都必须使用 USS_Drive_Control / USS_Drive_Control_31 指令的背景数据块。必须从主程序的循环 OB 调用 USS_Read_Param / USS_Read_Param_31。

参数

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
REQ	IN	Bool		–	在 REQ 的上升沿创建新的读取请求。
DRIVE	IN	USInt	Byte	–	变频器地址：DRIVE 是 USS 变频器的地址。有效范围是变频器 1 与变频器 16 之间。
PARAM	IN	UInt		–	参数编号：PARAM 指定要写入的变频器参数。此参数的范围是 0 到 2047 之间。对某些变频器来说，INDEX 参数的最高有效字节可用来访问大于 2047 的参数值。变频器手册中有关于访问扩展范围的更多信息。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
INDEX	IN	UInt		–	参数索引：INDEX 指定要写入的变频器参数索引。它是 16 位值，其中最低有效位是介于 0 至 255 的实际索引值。变频器也可使用特定于变频器的最高有效字节。有关详细信息，请参见变频器手册。
USS_DB	INOUT	USS_BASE		–	USS_DB 参数必须连接到背景数据块的（静态）USS_DB 参数，该参数是在向程序中添加 USS_Drive_Control / USS_Drive_Control_31 指令时生成并初始化的。
DONE[1]	OUT	Bool		FALSE	如果此参数为 TRUE，则读取参数的先前请求值可用于 VALUE 输出。USS_Drive_Control / USS_Drive_Control_31 指令识别出变频器的读取响应时此位置位。下次调用 USS_Read_Param / USS_Read_Param_31 时此位复位。
ERROR	OUT	Bool		FALSE	ERROR = TRUE：出现错误且 STATUS 输出有效。出错时所有其它输出都置零。只在 USS_Port_Scan / USS_Port_Scan_31 指令的 ERROR 和 STATUS 输出发出通信错误信号。 可能需要检查 USS_Drive_Control / USS_Drive_Control_31 指令的背景数据块中静态变量 USS_DB. w_USSExtendedError 的值。
STATUS	OUT	UInt		0	错误代码（请参见错误消息 (页 204)）。
VALUE	OUT	Variant (Word, Int, UInt, DWord, DInt, UDInt, Real)	Any (Word, Int, DWord, DInt, Real)	–	这是读取的参数值，此值只有在 DONE 位为 True 时才有效。

[1] DONE 位表示已从引用的电机变频器读出有效数据并将其传到 CPU。这并不表示该指令能够立即读出其它参数。在相应变频器释放参数通道以供使用之前，必须将空的读取请求发送到电机变频器，并且必须由指令确认。直接调用特定电机变频器的 USS_Read_Param / USS_Read_Param_31 或 USS_Write_Param / USS_Write_Param_31 会导致错误 16#818A。

指令版本

USS_Read_Param:

版本 1.5 的功能与版本 1.4 完全相同，本次版本升级仅仅体现在内部措施方面。

USS_Read_Param_31:

版本 1.1 的功能与版本 1.0 完全相同，本次版本升级仅仅体现在内部措施方面。

5.4.3.7 USS_Write_Param / USS_Write_Param_31：在变频器中更改数据

说明

使用 CM1241
自模块的固件版本 V2.1 起，才能通过 CM1241 使用该指令。

说明

使用 USS_Write_Param_31 指令
只能在 S7-1500 CPU 上使用 USS_Write_Param_31 指令。

说明

对于 EEPROM 写入指令（USS 变频器中的 EEPROM）：
尽可能减少 EEPROM 写入操作的次数以最大化地延长 EEPROM 的使用寿命。

说明

USS_Write_Param 指令更改 16 个变频器之一的参数。

USS_Write_Param_31 指令更改 31 个变频器之一的参数。

分配给一个 USS 网络和一个 PtP 通信端口的所有 USS 函数都必须使用 USS_Drive_Control / USS_Drive_Control_31 的背景数据块。

必须从主程序循环的 OB 调用 USS_Write_Param / USS_Write_Param_31。

参数

表格 5- 38　参数的数据类型

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
REQ	IN	Bool		–	在 REQ 的上升沿创建新的写入请求。
DRIVE	IN	USInt	Byte	–	变频器地址：DRIVE 是 USS 变频器的地址。有效范围是变频器 1 与变频器 16 之间。
PARAM	IN	UInt		–	参数编号：PARAM 指定要写入的变频器参数。此参数的范围是 0 到 2047 之间。对某些变频器来说，INDEX 参数的最高有效字节可用来访问大于 2047 的参数值。变频器手册中有关于访问扩展范围的更多信息。
INDEX	IN	UInt		–	参数索引：INDEX 指定要写入的变频器参数索引。它是 16 位值，其中最低有效位是介于 0 至 255 的实际索引值。变频器也可使用特定于变频器的最高有效字节。有关详细信息，请参见变频器手册。
EEPROM	IN	Bool		–	保存在变频器的 EEPROM 中：如果为 TRUE，用于写入变频器的参数事务保存在变频器的 EEPROM 中。如果为 FALSE，写入的值仅临时保存，下次启动变频器时将丢失。
VALUE	IN	Variant (Word, Int, UInt, DWord, DInt, UDInt, Real)	Any (Word, Int, DWord, DInt, Real)	–	要写入的参数值。它在 REQ 上升沿时必须有效。
USS_DB	INOUT	USS_BASE		–	USS_DB 参数必须连接到背景数据块的（静态）USS_DB 参数，该参数是在向程序中添加 USS_Drive_Control / USS_Drive_Control_31 指令时生成并初始化的。

参数	声明	数据类型		标准	说明
		S7-1200 /1500	**S7-300/400/ WinAC**		
DONE[1]	OUT	Bool		FALSE	如果为 TRUE，则表示 VALUE 输入已写入变频器。USS_Drive_Control / USS_Drive_Control_31 指令识别出变频器的写入响应时此位置位。下次调用 USS_Write_Param / USS_Write_Param_31 时此位复位。
ERROR	OUT	Bool		FALSE	如果为 TRUE，则表示出现错误且 STATUS 输出有效。出错时所有其它输出都置零。只在 USS_Port_Scan / USS_Port_Scan_31 指令的 ERROR 和 STATUS 输出发出通信错误信号。 可能需要检查 USS_Drive_Control / USS_Drive_Control_31 指令的背景数据块中静态变量 USS_DB. w_USSExtendedError 的值。
STATUS	OUT	UInt		0	错误代码（请参见错误消息 (页 204)）。

[1] DONE 位表示已从引用的电机变频器读出有效数据并将其传到 CPU。这并不表示 USS 库可立即读出其它参数。在相应变频器释放参数通道以供使用之前，必须将空的写入请求发送到电机变频器，并且必须由指令确认。直接调用特定电机变频器的 USS_Read_Param / USS_Read_Param_31 或 USS_Write_Param / USS_Write_Param_31 函数会导致错误 0x818A。

指令版本

USS_Write_Param:

版本 1.6 的功能与版本 1.5 完全相同，本次版本升级仅仅体现在内部措施方面。

USS_Write_Param_31:

版本 1.1 的功能与版本 1.0 完全相同，本次版本升级仅仅体现在内部措施方面。

5.4.3.8 关于变频器设置的常规信息

变频器设置的要求

- 用户需要对变频器的 4 个 PIV 字 (**P**arameter**I**D**V**alue) 的用途进行组态。
- 变频器可组态 2 个、4 个、6 个或 8 个 PZD 字（过程数据区）。
- 变频器中 PZD 字的数量必须对应于变频器的 USS_Drive_Control 指令的 PZD_LEN 输入。
- 确保所有变频器的数据传输速率都对应于 USS_Port_Scan 指令的 BAUD 输入。
- 确保为 USS 通信设置变频器。
- 确保在变频器中指定由 USS 接口提供频率设定值。
- 确保指定了变频器地址（区域：1-16）。
  此地址必须对应于变频器的 USS_Drive_Control 块的 DRIVE 输入。
- 确保正确终止 RS485 网络。

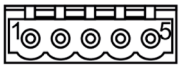
SINAMICS V20 变频器的连接与设置

有关在 S7-1200 中运行 SINAMICS V20 的应用实例，请访问 Internet (http://support.automation.siemens.com/CN/view/zh/63696870)。

连接 SINAMICS V20 变频器

将 SIEMENS G120(C) 变频器连接到 USS 网络的示例。 有关其它变频器的连接示例，请参见相应的变频器手册。

通过插入式连接实现 SINAMICS G120(C) 变频器到 USS 网络的连接。 连接具有短路保护和绝缘功能。



1 0 V 参考电位
2 RS485N，接收和发送 (-)
3 RS485N，接收和发送 (+)
4 电缆屏蔽
5 未使用

图 5-3 USS 连接

注意
不同参考电压 如果连接没有相同参考电压的设备，则可能在连接电缆中产生意外电流。 这些意外电流可能导致通信错误或设备损坏。 确保使用通信电缆连接的所有设备在电路中具有相同的参考导线，或者已在电气上断开以避免产生意外电流。 确保屏蔽接地或连接到变频器总线连接器的引脚 1。 确保 G120(C) 的接线终端 2 (GND) 接地。

如果 RS485 主站（例如，带 CM1241 通信模块的 S7-1200 CPU）通过 PROFIBUS 连接器连接，则按如下操作连接总线电缆：

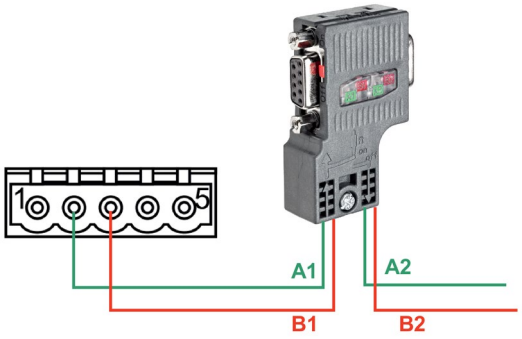


图 5-4　　通信模块的连接

如果 RS485 主站是网络中的终端站或采用点对点连接，则必须使用 PROFIBUS 连接器的端子 A1 和 B1（不是 A2 和 B2），因为这些端子提供了终止设置（例如，DP 插头连接器 6ES7972-0BB52-0XA0）。

如果 G120(C) 已组态为网络中的终端站，则必须将总线终端电阻器开关设置为"接通"。

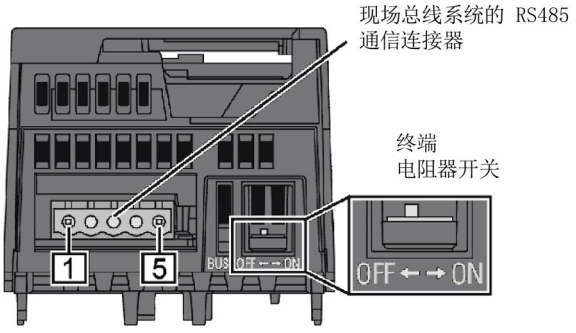


图 5-5　　终端站的连接

G120(C) 变频器设置

将变频器连接到 S7-1500 或 ET 200SP 前，确保变频器具有下列系统参数。

步骤	指令	操作说明	
		G120[1)]	**G120C [2)]**
1	通过操作员面板 BOP-2 执行变频器的基本调试。 变频器为输入、输出和现场总线接口提供了不同的默认值（宏指令）。 在基本调试的第九步 (MAC PAR p15)，为 USS 通信选择宏指令 21。 以决定以下参数的默认值： • 数据传输速率 (p2020)： 38400 bps • PZD 数量 (p2022)： 2 • PIV 数量 (p2023)： 变量 注意： 也可通过 STARTER 调试软件或 SINAMICS Startdrive 进行基本调试。	章节 4.4.3	章节 6.4.1

步骤	指令	操作说明	
		G120[1)]	G120C [2)]
2	通过 G120 或 G120(C) 控制单元上的地址开关指定变频器的 USS 地址。 • 有效地址范围：1...30 注意： 也可通过参数 p2021、STARTER 或 SINAMICS Startdrive 设置 USS 地址。	章节 6.2.2.1	章节 8.4.2.1
	通过以下步骤，可使用 BOP-2 输入参数编号和修改参数值，从而直接访问参数。	章节 4.4.2	章节 6.4.2
3	在您的应用中，调整以下与通信相关的变频器参数： • 数据传输速率 (p2020)，如果 ≠ 38400 bps （确保设置与 USS_Port_Scan 通信指令的 BAUD 参数完全相同。） • PZD 数量 (p2022)，如果 ≠ 2 （确保设置与 USS_Drive_Control 通信指令的 PZD_LEN 参数完全相同。） • PIV 数量 (p2023) = 4 （将通过宏指令 21 设置为“变量”(127) 的值更改为默认值“4”（指令 USS_Read_Param 和 USS_Write_Param 所需）。） • 现场总线 SS 监视时间 [ms] (p2040)	章节 6.2.2.2	章节 8.4.2.1
4	指定速度设定值源。 • n_set Eval (p1000[0]) = 6 （速度设定值由 USS 总线提供。）		
5	设置速度和频率参考值。 • n_reference f_reference (p2000) =（6,00 min^{-1} 到 210000,00 min^{-1}） （所有相关速度或频率均参考此参考值。 参考值对应于 100%、4000_{hex}（字）或 4000 0000_{hex}（双字）。如下要求适用： $f_{reference\_value}$（以 Hz 为单位）= $n_{reference\_value}$（以（(min^{-1})/60) x 极对数）为单位）		
6	将参数传送到非易失性存储器中。 • 保存 par (p0971) = 1		

1) G120 (http://support.automation.siemens.com/CN/view/zh/62089662)

2) G120(C) (http://support.automation.siemens.com/WW/view/zh/61462568)

5.4.3.9 错误消息

USS 错误消息概述

错误代码	说明	解决办法
16#0000	无错误	-
16#8180	变频器响应的长度错误	检查变频器的响应帧。
16#8181	数据类型错误	检查参数 VALUE。
	参数编号错误	PARAM 参数允许的值范围：0 到 2047
16#8182	数据类型错误：针对"字"请求不得返回"双字"或"实数"。	检查变频器的响应帧。
16#8183	数据类型错误：针对"双字"或"实数"请求不得返回"字"。	检查变频器的响应帧。
16#8184	变频器响应的校验和错误	检查变频器和通信连接。
16#8185	寻址错误	有效的变频器地址范围：1 到 16
16#8186	设定值错误	有效的设定值范围：-200% 至 +200%
16#8187	返回的变频器编号错误	检查变频器的响应帧。
16#8188	无效 PZD 长度	允许的 PZD 长度：2、4、6 或 8 个字
16#8189	该模块不支持此数据传输速率。	为该模块选择有效的数据传输速率。
16#818A	此变频器的另一个请求当前处于激活状态。	稍后重复参数读取或写入命令。
16#818B	变频器未响应。	检查变频器。
16#818C	变频器对参数请求响应错误消息。	检查变频器的响应帧。 检查参数请求。 检查指令 USS_Read_Param、USS_Write_Param 或 USS_Port_Scan 是否已报告错误。如果已报告错误，请检查 USS_Drive_Control 指令的静态变量 USS_DB.w_USSExtendedError 的值。
16#818D	变频器对参数请求响应访问错误消息。	检查变频器的响应帧。 检查参数请求。
16#818E	变频器未初始化。	检查用户程序，确保向此变频器调用 USS_Drive_Control 指令。

错误代码	说明	解决办法
16#8280	读取模块时进行否定确认	检查 PORT 参数中的输入。 在静态参数 Port_Config.RDREC.STATUS、Send_Config.RDREC.STATUS 、Receive_Config.RDREC.STATUS 、Send_P2P.RDREC.STATUS 或 Receive_P2P.RDREC.STATUS，以及 SFB RDREC 的说明中可找到有关错误原因的更多详细信息。
16#8281	写入模块时进行否定确认	检查 PORT 参数中的输入. 在静态参数 Port_Config.WRREC.STATUS、Send_Config.WRREC.STATUS 、Receive_Config.WRREC.STATUS 、Send_P2P.RDREC.STATUS 或 Receive_P2P.RDREC.STATUS，以及 SFB WRREC 的说明中可找到有关错误原因的更多详细信息。
错误代码，一般		
16#8FFF	模块因复位而暂时未准备就绪。	重复请求。

[1] 仅限 S7-300/400 CPU 的指令

6 启动和诊断

6.1 启动特性

工作模式转换

通信模块启动后，CPU 和通信模块之间的所有数据均通过指令进行交换。

CPU STOP	在通信模块与 CPU 之间运行数据传输期间，发送与接收作业均中止。 将继续接收帧。但是，只有在组态为不清除接收缓冲区的情况下，在 STOP-RUN 切换后才会将相关信息转发给 CPU。
CPU RUN	CPU 处于 RUN 状态下时，可确保发送和接收操作。 通过通信模块的属性对话框中的相应组态，您可以在 CPU 启动期间自动清除通信模块上的接收缓冲区。

从通信模块的角度来看，没有任何其它工作状态/工作状态转换。

6.2 诊断功能

通信模块的诊断功能可快速定位发生的错误。您可以选择以下诊断选项：

通过通信模块的显示元件进行诊断	指示器提供有关通信模块的操作模式或可能的错误状态的信息。指示器提供所有内部或外部错误以及接口特定错误的初步概述。有关详细信息，请参见相应通信模块的设备手册。
通过指令的 STATUS 输出的诊断	指令具有用于错误诊断的 STATUS 输出；其提供有关通信模块和 CPU 之间的通信错误的信息。可以在用户程序中评估 STATUS 参数 (页 126)（该参数仅存在一个周期）。
诊断错误中断	通信模块可在分配给它的 CPU 上触发诊断错误中断。通信模块提供诊断信息。可通过用户程序或通过读取 CPU 诊断缓冲区对该信息进行分析。
通过 EventTracePtP 数据记录进行诊断 (页 207)	通过 EventTracePtP 数据记录，可记录和读出最近 1000 个通信事件以及通信模块的参数分配。

6.3 诊断中断

诊断在 STEP 7 (TIA Portal) 的在线和诊断视图中以纯文本形式显示。可通过用户程序评估错误代码。

可能指示以下诊断信息：

- 错误 (9_H)
- 参数分配错误 (10_H)
- 断线（S7-1500/ET 200MP：109_H；ET 200SP：6_H）

6.4 通过 EventTracePtP 数据记录进行诊断

可在模块的环形缓冲区（循环存储器）中记录最近的 1000 个通信事件以及通信模块的参数分配。如有必要，可读取并评估记录的数据，例如接收和传输的数据或错误。该功能通过 EventTracePtP 数据记录（数据记录 62）进行控制。

自固件版本 V2.0 起，可使用以下通信模块的事件诊断：

- ET 200MP CM PtP RS232 BA
- ET 200MP CM PtP RS232 HF
- ET 200MP CM PtP RS422/485 BA
- ET 200MP CM PtP RS422/485 HF
- ET 200SP CM PtP（即将上市）

记录事件和数据

在事件诊断期间，模块记录以下事件和数据：

- 启动模块
- 模块的参数分配：
  - 端口组态
  - 发送组态
  - 接收组态

  - RS232 伴随信号组态
  - 3964 协议组态
  - 诊断中断组态
  - 错误代码
- 接收的数据
- 发送的数据
- RS232 伴随信号
- 报文开始和结束条件
- 检测到发送和接收错误
- 与 CPU 的数据交换

读取选项

可通过以下方式读取记录的事件诊断：

- 编程和调试支持：
  MFCT 已安装到 PC/PG 上。如果需要，可创建读取数据的副本。有关 MFCT 的信息，敬请访问 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109773881)。
- 偶发错误分析：
  发生错误时，LPTP_GetEventTrace 功能块自动将读取的数据副本存储在 CPU 的存储卡上。稍后可对该副本进行更详细的分析。有关功能模块的信息，请参见 FAQ 中的条目 ID 109973142 (https://support.industry.siemens.com/cs/ww/zh/view/109973142)。

  说明

  功能块 LPTP_GetEventTrace 应已包含在用户程序中，以便可在需要时进行激活（例如，在线激活）。
- 与其他制造商的系统搭配使用：借助此部分中的信息，
  还可将事件诊断与其他制造商的系统搭配使用。

参见

数据记录 EventTracePtP（页 209）

数据记录 EventTracePtP

A

A.1 EventTracePtP 的使用和结构（数据记录 62）

通过 EventTracePtP 数据记录，可激活和读出最近 1000 个通信事件的记录以及相应通信模块的参数分配。记录在环形缓冲区（循环存储器）中进行。控制命令通过数据记录 62 发送到模块（例如使用 WRREC 指令），或者从模块读取状态和事件诊断数据（例如使用 RDREC 指令）。

数据记录 62 的结构 (EventTracePtP)：写入

下表列出了写入数据记录 62 时的结构。

表格 A- 1　写入数据记录 62

位 → / 字节 ↓	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0...7	**PtpVersionHeader:** 数据记录的长度							
0...1	块类型：$003E_H$							
2...3	块长度：0006_H							
4	高字节块版本：01_H							
5	低字节块版本：00_H							
6...7	预留 [1]							
8	**EventTrace_Control:** 控制 EventTracePtP							
	预留 [1]						模式：	事件诊断：
							0_B: Logging Mode[3]	0_B：禁用 [2]
							1_B: Reading Mode[4]	1_B：激活（默认）
9	预留 [1]							

1 预留的位必须置为 0。

2 未记录其它事件。模块中相应的缓冲区被删除。

3 已记录新事件。

4 新事件的记录被中断。

数据记录 62 的结构 (EventTracePtP)：读取

下表列出了读取数据记录 62 时的结构。

表格 A- 2　读取数据记录 62

位 → / 字节 ↓	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0...5	**EventTrace_ReadRecordHeader**							
0...1	EventTrace_DataReadAddress: EventTrace_Data 的起始地址							
2...3	EventTrace_DataTotalSize: EventTrace_Data 的长度							
4	EventTrace_Status: 事件诊断的状态信息							
	预留 [1]				Reading Mode 已完成：	Reading Mode 已启动：	模式状态：	事件诊断状态：
					0_B：否	0_B：否	0_B: Logging Mode	0_B：已禁用
					1_B：是	1_B：是	1_B: Reading Mode	1_B：已激活（默认）
5	预留 [1]							
6 ... 5 +n*6	**EventTrace_Data:** 来自参数分配和事件诊断的数据（事件 1 ... n）							

[1] 预留位置为 0。

设置 EventTrace_Data

下表列出了 EventTrace_Data 的结构。

字节 ↓	含义
0...7	**PtpVersionHeader:** 数据记录的长度
0...1	保留（不评估该值）
2...3	保留（不评估该值）
4	高字节块版本
5	低字节块版本
6...7	预留 [1]
8...15	**General_Info**
8	固件版本：Letter (CHAR)
9	固件版本：Major (BYTE)
10	固件版本：Minor (BYTE)
11	固件版本：Patch (BYTE)
12...15	模块 ID
16...43	**Port_Config_Str:** 数据记录 57（端口组态：0039H）
44...75	**Send_Config_Str:** 数据记录 59（传输组态：003BH）
76...143	**Receive_Config_Str:** 数据记录 60（接收组态：003CH）
144...159	**P3964_Config_Str:** 数据记录 61（3964 协议组态：003DH）
160...171	**Set_Features_Str:** 数据记录 58（激活特殊功能：003AH）
172...183	**Signal_Set:** 数据记录 53（设置 RS232 伴随信号：0035H）
184...203	**PtP RD_ESTAT:** 数据记录 55（读取错误代码：0037H）

字节 ↓	含义
204...205	**Number_Trace_Events:** 事件数 n
206... 205+n*6	**EventTraceEntry[n]:** n 个事件的条目（n 乘以 6 个字节）
206...211	EventTraceEntry[1]: 事件 1 的条目
206...209	EventTraceEntry[1]: UDINT: TimeStamp: 事件时间戳，分辨率为 1 µs，32 位计数器
210	EventTraceEntry[1]: EventType: 事件编号（请参见下表）
211	EventTraceEntry[1]: EventInfoByte: 事件详细信息（请参见下表）
212...217	EventTraceEntry[2]: 事件 2 的条目
...	...

[1] 预留位置为 0。

说明

数据集 53 至 61

数据集 53 至 61 的结构可在 Internet (https://support.industry.siemens.com/cs/ww/zh/view/59062563) 的《在没有 SIMATIC 系统指令的情况下，运行 CM PtP》手册中找到。

Event-Type	EventInfoByte	含义
组态事件：		
0	1	模块启动
1	• 0：模块已重新组态 • 1：组态已删除	已收到新组态
2	• 57D: Port_Config • 58D: Set_Features • 59D: Send_Config • 60D: Receive_Config • 61D: P3964_Config	已收到新参数化
接收事件：		
50	0...FFH	接收数据的新接收有效字节
51	0...FFH	接收数据的新接收字节，有奇偶校验错误
52	0...FFH	接收数据的新接收字节，有字符帧错误
53	0...FFH	无法读取接收数据的新接收字节 (Overrun)。因此，可读取最新接收的字节的条目。
54	0...FFH	接收中断。因此，可读取最新接收的字节。

Event-Type	EventInfoByte	含义
55	• 1_H：START_CHAR（以起始字符开头） • 2_H：START_ANY_CHAR（以任意字符开头） • 4_H：START_LINE_BREAK（在换行符后开头） • 8_H：START_IDLE_LINE（在空闲线后开头） • 10_H：START_SEQ1（以初始序列 1 开头） • 20_H：START_SEQ2（以初始序列 2 开头） • 40_H：START_SEQ3（以初始序列 3 开头） • 80_H：START_SEQ4（以初始序列 4 开头）	检测到报文开始

Event-Type	EventInfoByte	含义
56	• 94H：RX_COMPLETE_LENGTH（因过长而导致报文结束） • 95H：RX_COMPLETE_MESSAGE_TO（由于消息超时而导致报文结束） • 96H：RX_COMPLETE_INTERCHAR_TO（由于字符延迟时间到期而导致报文结束） • 97H：RX_COMPLETE_RESPONSE_TO（由于响应超时而导致报文结束） • 98H：RX_COMPLETE_CALC_LENGTH（由于从消息中读取消息长度而导致报文结束） • 99H：RX_COMPLETE_END_SEQUENCE（因结束序列而导致报文结束） • E0H：RX_COMPLETE_BUFFER_FULL（接收缓冲区已满） • E1H：RX_COMPLETE_PARITY（由于奇偶校验错误而导致消息错误） • E2H：RX_COMPLETE_FRAMING（由于组帧错误而导致消息错误） • E3H：RX_COMPLETE_OVERRUN（由于内部超限而导致消息错误）	检测到报文结束

Event-Type	EventInfoByte	含义
56	• $E4_H$：RX_COMPLETE_CALC_LENGTH_ERROR（由于从消息中读取消息长度时出错而导致消息错误） • $E5_H$：RX_COMPLETE_BREAK_DETECT（检测到断线消息错误） • $E8_H$：RX_FIXLENGTH_TIMEOUT_ERROR（由于固定报文长度的超时错误而导致消息错误） • EB_H：RX_MAX_MSG_SIZE_EXCEEDED（由于超出最大消息长度而导致消息错误）	检测到报文结束
57	$0...FF_H$	接收到 XOFF 字符
58	$0...FF_H$	接收到 XON 字符
59	• 0: CTS = OFF • 1: CTS = ON	CTS 已更改
60	• 0: DSR = OFF • 1: DSR = ON	DSR 已更改
61		接收缓冲区已满
62		超出报文长度
63	$0...FF_H$	接收缓冲区上溢：最旧的消息被覆盖
64		CRC 错误

Event-Type	EventInfoByte	含义
发送数据事件：		
100	0...FF$_H$	已发送一个字节
101	• 00$_H$：由于其它原因 • 01$_H$：由于 DSR • 02$_H$：由于 CTS • 03$_H$：由于 DTR • 04$_H$：由于 RTS • 05$_H$：由于 XOFF	无法发送已传输数据的新字节...
102	0...FF$_H$	XOFF 字符已发送
103	0...FF$_H$	XON 字符已发送
104	• 0: RTS = OFF • 1: RTS = ON	RTS 已更改
105	• 0: DTR = OFF • 1: DTR = ON	DTR 已更改
106	• D1$_H$：WAIT_TIMEOUT（超出 CTS 的等待时间） • D2$_H$：DSR_INACTIVE（由于 DTR 被收回，DSR 处于非活动状态） • D5$_H$：CANCELED（CPU 停止或断线）	发送中断
107		开始发送报文
108		报文发送完成
109	低字节报文长度	从用户程序向模块传输新的发送报文

Event-Type	EventInfoByte	含义
CPU 通信事件：		
149	• 00H：无错误 • A2H：模块错误 • B0H：索引无效 • B1H：数据记录长度不正确 • B5H：无效状态 • E1H：参数无效	针对数据记录 49 执行了 RDREC（例如通过 S7-1500 指令 Receive_P2P）
150	• 00H：无错误 • A2H：模块错误 • B0H：索引无效 • B1H：数据记录长度不正确 • B5H：无效状态 • E1H：参数无效	针对数据记录 50 执行了 RDREC（例如通过 S7-1500 指令 Receive_P2P）
151	• 00H：无错误 • B0H：索引无效 • B1H：数据记录长度不正确 • B5H：无效状态 • E1H：参数无效	针对数据记录 48 执行了 WRREC（例如通过 Send_P2P 指令）
152	• 00H：无错误 • B1H：数据记录长度不正确 • E1H：参数无效	针对数据记录 62 执行了 WRREC
153	• 00H：无错误 • B1H：数据记录长度不正确 • E1H：参数无效	针对数据记录 53 执行了 WRREC
154	• 00H：无错误 • B1H：数据记录长度不正确 • B5H：无效状态 • E1H：参数无效	针对数据记录 54 执行了 WRREC（例如通过 Receive_Reset 指令）

Event-Type	EventInfoByte	含义
155	0：在输入数据中输入了新报文	选择性能优化选项时： 更改了输入数据
156	0：CPU 接收到新报文	选择性能优化选项时： 更改了初始数据
其它事件：		
200		已开始记录新事件，或在中断后继续记录新事件
201		模式已更改为“Reading Mode”
202		模式已更改为“Logging Mode”
205		时间戳计数器溢出
206		内部接收缓冲区已删除
207	字节数（低字节）	接收为 CPU 准备的报文
208	• 0：修复断线 • 1：存在断线	内部断线信号
209	• 0：修复断线 • 1：存在断线	断线诊断（发送）
210	• 0：修复参数化错误 • 1：存在参数化错误	参数化错误诊断（发送）

事件诊断

默认情况下，事件诊断在 EventTracePtP 数据记录中激活。

可通过 EventTracePtP 控制以下两种模式：

Logging Mode	新事件和数据记录在模块中。
Reading Mode	从模块中读取事件和数据。新事件和数据的记录被中断。

两种模式的步骤如下所述。

激活事件诊断 (Logging Mode)

要使用数据记录 62 激活事件诊断，请按以下步骤操作：

1. 将"事件诊断"位设为 1。
2. 将"模式"位设为 0。
3. 将数据记录发送至模块。
   "Logging Mode"已激活。新事件已保存。

读取事件诊断 (Reading Mode)

要使用数据记录 62 从事件诊断中读取一致数据，请按以下步骤操作：

1. 将"事件诊断"位设为 1。
2. 将"模式"位设为 1。
3. 将数据记录发送至模块。
   "Reading Mode"已激活。在此期间无法记录新事件。
4. 从模块中读取地址 0 (EventTrace_DataReadAddress) 处的数据记录。
   由于数据量高达 6212 字节 (EventTrace_DataTotalSize)，可能需要多个读取周期。
   - 检查"事件诊断状态"和"模式状态"位是否设为 1。
   - 检查"读取模式已开始"(EventTrace_DataReadAddress = 0) 或"读取模式已结束"位。
     当"读取模式已结束"位设为 0 时，需要对数据记录进行额外的读取周期。
5. 将 EventTrace_Data 中的新数据添加到缓冲区中的相应位置，该位置必须具有 EventTrace_DataTotalSize 的最小大小。
6. 重复读取数据记录并保存数据，直到"Reading Mode 已完成"位设为 1，或者达到 EventTrace_DataTotalSize 值。
   然后参数分配和事件诊断数据被完全读出。

词汇表

"性能"选项

自通信模块的固件版本 V2.0 起，可使用性能优化选项。如果仅通过几个通信模块发送和接收短帧，则此选项适用。对于接收帧，将帧长度限制为 24 个字节；对于发送帧，则限制为 30 个字节。优化了响应时间，尤其是并行使用多个 CM PtP 时。

从 V4.0 开始的 Send_P2P 和 Receive_P2P PTP 指令以及从 V5.0 开始的 Modbus (RTU) 和 USS 通信指令库支持性能选项。

CPU

中央处理单元 = 包含控制和计算单元、存储器、系统程序和 I/O 模块接口的自动化系统的中央模块。

CTS

Clear to send. 通信伙伴可随时接收数据。

DCD

Data carrier detect. 通信伙伴指明其已识别到进入数据。

DSR

Data set ready. 通信伙伴就绪。

DTR

Data terminal ready. 通信模块就绪。

RI

Ring indicator. 用于连接调制解调器的呼入。

RTS

Request to send. 通信模块做好发送准备。

USS

USS® 协议（通用串行接口协议）定义了一种基于主站-从站原理通过串行总线进行通信的访问方法。其中，点对点连接是该协议的一个子集。

XON/XOFF

使用 XON/XOFF 进行软件数据流控制。可为 XON 和 XOFF 组态字符（任何 ASCII 字符）。用户数据可能不包含这些字符。

参数

参数是可以分配的值。有两种不同类型的参数：块参数和模块参数。

参数分配

参数分配是指模块特性的设置。

程序

程序是指根据特定协议进行数据传输的过程。

点对点连接

在点对点连接中，通信模块构成可编程逻辑控制器与通信伙伴之间的端口。

接收线路初始状态

RS422 和 RS485 模式接收线路的初始状态：

- 实现断路检测（断线）
- 确保未发送时接收线路上的已定义电平。

模块参数

模块参数是可以用来设置模块行为的值。

缺省设置

缺省设置是一种合理的基本设置，只要未指定其它值就可以使用缺省设置。

软件

软件是计算系统中使用的所有程序的总称。操作系统和用户程序都属于软件。

通信模块

通信模块是用于点对点连接和总线连接的模块。

位时间

“位时间”始终被指定为位数。

以位数设置的“时间”取决于自动纳入考虑范围内的选定数据传输速率。

示例：

在两个字符间隙之后检测到帧结束。

设置的数据传输速率为 9600 位/秒。

设置的字符帧为 10 位。

10 x 2 = 20 位时间

这对应于时间：

20 x 1/9600 ≈ 0,0021 s

协议

数据传输涉及的所有通信伙伴必须遵守一套固定的规则来处理和实现数据通信。这些规则称为协议。

循环程序处理

在循环程序处理中，用户程序以固定时间间隔重复执行的程序循环(或称为“周期”)运行。

硬件

硬件是自动化系统的全部物理和技术设备。

用户程序

用户程序包含处理用于控制系统或过程的信号的所有指令和声明。在 SIMATIC S7 中，将用户程序结构化，并以块为单位划分为较小的单元。

在线/离线

在线时，自动化系统和编程设备之间存在数据连接；离线时，二者之间无数据连接。

诊断功能

诊断功能涉及整个系统诊断，并包括自动化系统中对错误的识别、解释及报告。

诊断缓冲区

根据诊断事件的发生顺序，在其中输入有关所有诊断事件的详细信息的存储区。

诊断事件

举例而言，诊断事件是 CPU 中的模块错误或系统错误，这些错误可能由程序错误引起。

周期时间

周期时间是 CPU 处理用户程序一次所需要的时间。

自动化系统

自动化系统是一个可编程逻辑控制器，至少由一个 CPU、各种输入和输出模块以及操作和监视设备组成。

组态

组态是指组态表中自动化系统的各个模块的组态。

索引

R

S

U

S H