

SIEMENS

SIMATIC

ET 200MP/ET 200SP 用于 S7-300/400 分布式 I/O 的 PtP 通信指令

编程手册

前言

文档指南

1

简介

2

编程 - 使用指令进行通信

3

错误消息

4




服务与支持

A

法律资讯

警告提示系统

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

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


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

合格的专业人员

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

按规定使用 Siemens 产品

请注意下列说明：

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

商标

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

责任免除

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

前言

本文档用途

本文档提供了有关在 S7-300/400 系统中使用为 TIA-Portal V12 开发的指令对 ET 200MP 和 ET 200SP 的点对点通信模块进行编程的重要信息（分布式使用）。

所需基本知识

理解本文档中的内容，需要具备以下知识：

- 自动化技术的基本知识
- SIMATIC 工业自动化系统知识
- 基于 Windows 的计算机使用知识
- 熟练掌握 STEP 7

文档的有效性

本文档适用于以下点对点通信模块

- CM PtP RS232 BA (6ES7540-1AD00-0AA0) 通信模块
- CM PtP RS422/485 BA (6ES7540-1AB00-0AA0) 通信模块
- CM PtP RS232 HF (6ES7541-1AD00-0AB0) 通信模块
- CM PtP RS422/485 HF (6ES7541-1AB00-0AB0) 通信模块
- CM PtP (6ES7137-6AA00-0BA0) 通信模块

约定

本手册中使用的术语“CPU”既可指代 S7-300/400 的 CPU，也可指代分布式 I/O 系统的接口模块。

请同时遵循以下所标注的注意事项：

说明

这些注意事项包含有关本文档所述的产品、使用该产品或应特别关注的文档部分的重要信息。

回收和处置

这些产品是低污染设备，所以可回收。为了使旧设备的回收和处置符合环保要求，请联系一家经认证的电子废料处理服务公司。

其它帮助

- 有关技术支持 (<http://www.siemens.com/automation/support-request>)的信息，请参见本文档中的附录部分。
- 有关各种 SIMATIC 产品和系统的技术文档信息，请访问 SIMATIC Portal (<http://www.siemens.com/simatic-tech-doku-portal>)。
- 目录 (<http://mall.automation.siemens.com>) 中还提供了在线目录和在线订购系统。

有关 IT 安全的提示

西门子为其自动化和驱动产品系列提供 IT 安全机制，从而支持设备/机器的安全运行。因此，建议您持续关注这些产品 IT 安全准则的更新信息。可上网 (<http://support.automation.siemens.com>)查找相关信息。

您可以在该网站中进行注册，定制指定产品的实时信息。

不过，为了确保设备/机器的安全运行，还需将该自动化组件集成到整个设备/机器的全面的 IT 安全方案中。可上网 (<http://www.siemens.com/industrialsecurity>)查找相关提示。

在此，还可以查看其它厂商所使用的产品。

目录

前言	3
1 文档指南	7
2 简介	9
2.1 约定	9
2.2 指令概述	9
3 编程 - 使用指令进行通信	13
3.1 点对点	13
3.1.1 PtP 通信概述	13
3.1.2 使用指令	16
3.1.3 点对点操作的共享参数。	17
3.1.4 Port_Config: 组态 PtP 通信端口	20
3.1.5 Send_Config: 组态 PtP 发送方	24
3.1.6 Receive_Config: 组态 PtP 接收方	26
3.1.7 P3964_Config: 组态 3964 (R) 协议	33
3.1.8 Send_P2P: 发送数据	35
3.1.9 在通信操作中使用 LENGH 和 BUFFER 参数	37
3.1.10 Receive_P2P: 接收数据	38
3.1.11 Receive_Reset: 清除接收缓冲区	40
3.1.12 Signal_Get: 读取状态	41
3.1.13 Signal_Set: 设置伴随信号	42
3.1.14 Get_Features: 获取扩展功能	44
3.1.15 Set_Features: 设置扩展功能	45
3.2 Modbus	46
3.2.1 Modbus RTU 通信概述	46
3.2.2 Modbus_Comm_Load: 对 Modbus 的通信模块进行组态	48
3.2.3 Modbus_Master: 作为 Modbus 主站进行通信	52
3.2.4 Modbus_Slave: 作为 Modbus 从站进行通信	61
3.2.5 帧结构	66
3.3 USS	74
3.3.1 USS 通信概述	74
3.3.2 USS 协议使用要求	76
3.3.3 USS_Port_Scan: 通过 USS 网络进行通信	80
3.3.4 USS_Drive_Control: 准备并显示变频器数据	83
3.3.5 USS_Read_Param: 从变频器读取数据	87
3.3.6 USS_Write_Param: 在变频器中更改数据	89
3.3.7 关于变频器设置的常规信息	91

4	错误消息.....	95
A	服务与支持	115
	词汇表	119
	索引	123

文档指南

简介

本 SIMATIC 产品模块化文档涉及有关您的自动化系统的诸多主题。

S7-300/400 和 ET 200MP/SP 系统的完整文档包含相关操作说明、手册和产品信息。

有关点对点通信主题的文档概述

下表中包含修订本说明的附加文档。

主题	文档	重要内容
系统	系统手册ET 200SP 分布式 I/O 系统 http://support.automation.siemens.com/WW/view/zh/58649293 系统手册ET 200MP 分布式 I/O 系统 http://support.automation.siemens.com/WW/view/zh/59193214	<ul style="list-style-type: none"> • 应用规划 • 安装 • 连接 • 寻址 • 调试 • 维护

主题	文档	重要内容
通信	功能手册通信 (http://support.automation.siemens.com/WW/view/zh/59192925)	<ul style="list-style-type: none"> • 串行通信的基本知识 • 数据传输功能 • 诊断功能
	设备手册CM PtP RS232 BA 通信模块 (http://support.automation.siemens.com/WW/view/zh/59057152)	<ul style="list-style-type: none"> • 产品总览 • 连接 • 分配参数
	设备手册CM PtP RS232 HF 通信模块 (http://support.automation.siemens.com/WW/view/zh/59057160)	<ul style="list-style-type: none"> • 编程 • 诊断
	设备手册CM PtP RS422/485 BA 通信模块 (http://support.automation.siemens.com/WW/view/zh/59057390)	
	设备手册CM PtP RS422/485 HF 通信模块 (http://support.automation.siemens.com/WW/view/zh/59061372)	
	设备手册CM PtP 通信模块 (http://support.automation.siemens.com/WW/view/zh/59061378)	
系统诊断	功能手册系统诊断 (http://support.automation.siemens.com/WW/view/zh/59192926)	对于 S7-1500、ET 200MP、ET 200SP: <ul style="list-style-type: none"> • 诊断评估 • 硬件/软件
设置	功能手册EMC/EMI 兼容控制系统的安装 (http://support.automation.siemens.com/WW/view/zh/59193566)	<ul style="list-style-type: none"> • 基本信息 • 电磁兼容性 • 避雷

SIMATIC 手册

SIMATIC 产品的最新手册已在 Internet (<http://www.siemens.com/automation/service&support>) 上发布，可免费下载。

简介

2.1 约定

TIA Portal 术语

说明

本文档中使用 TIA Portal 术语。

- 指令
对于点对点、Modbus 和 USS 的块说明，这意味着您在 S7-300/400 环境中熟悉的 FB 和 SFB 之间的区别以及 FC 和 SFC 之间的区别不再相关，现在仅对指令进行引用。所有指令均不再取决于所使用的 CPU，而是既可用于 S7-300/400（分布在 ET 200MP/SP 中），又可用于 S7-1500（中央和分布式）。
 - 通信模块
对于设备的说明，这意味着您在 S7-300/400 环境所熟悉的术语“通信处理器”被“通信模块”取代。
-

2.2 指令概述

说明

指令的使用

为了能够在 S7-300/400-Umfeld（分布在 ET 200MP/SP 中）中使用下面介绍的指令，需要通过 HSP (Hardware Support Package) 在 STEP 7 (SIMATIC Manager) 中集成所需的库。

以下库可用：

- Punkt-zu-Punkt: PtP Communication, PtP_Comm
 - Modbus (RTU): MODBUS (RTU), MODBUS_(RTU)
 - USS: USS Communication, USS_Comm
-

指令概述

通信协议在通信模块上执行。协议用于使通信模块的接口适应通信伙伴的接口。

CPU、通信模块和通信伙伴之间通过特殊指令和支持相应通信模块的协议进行通信。

指令构成 CPU 与通信模块之间的软件接口。必须从用户程序中循环调用这些指令。数据传输将在多个周期内进行。

表格 2-1 PtP 的指令

指令	含义
Port_Config (页 20)	指令 Port_Config 允许动态分配基本接口参数。
Send_Config (页 24)	指令 Send_Config (发送组态) 允许动态分配端口的发送参数。
Receive_Config (页 26)	指令 Receive_Config (接收组态) 允许动态分配端口的接收参数。
P3964_Config (页 33)	指令 P3964_Config (协议组态) 允许动态分配 3964(R) 程序的参数。
Send_P2P (页 35)	指令 Send_P2P 允许将数据发送给通信伙伴。
Receive_P2P (页 38)	指令 Recveive_P2P 允许接收来自通信伙伴的数据。
Receive_Reset (页 40)	指令 Receive_Reset 允许清除通信模块的接收缓冲区。
Signal_Get (页 41)	指令 Signal_Get 允许读取 RS232 伴随信号。
Signal_Set (页 42)	指令 Signal_Set 允许设置 RS232 伴随信号。
Get_Features (页 44)	指令 Get_Features 允许读取通信模块支持的扩展功能。
Set_Features (页 45)	指令 Set_Features 允许激活通信模块支持的扩展功能。

表格 2-2 USS 的指令

指令	含义
USS_Port_Scan (页 80)	USS_Port_Scan 指令允许通过 USS 网络通信。
USS_Drive_Control (页 83)	USS_Drive_Control 指令允许与变频器交换数据。
USS_Read_Param (页 87)	USS_Read_Param 指令允许从变频器中读取参数。
USS_Write_Param (页 89)	USS_Write_Param 指令允许更改变频器中的参数。

表格 2-3 Modbus 的指令

指令	含义
Modbus_Comm_Load (页 48)	指令 Modbus_Comm_Load 允许组态 Modbus RTU 的通信模块端口。
Modbus_Master (页 52)	Modbus_Master 指令允许通过 PtP 端口作为 Modbus 主站进行通信。
Modbus_Slave (页 61)	Modbus_Slave 指令允许通过 PtP 端口作为 Modbus 从站进行通信。

编程 - 使用指令进行通信

3.1 点对点

3.1.1 PtP 通信概述

STEP 7 扩展了用户程序可以使用的指令，通过在用户程序中指定的协议实现点对点通信。这些指令可分为两类：

- 组态指令
- 通信指令

说明

CPU 数量结构

点对点指令通过读取或写入数据记录与通信模块进行通信。

因此在使用这种指令时，应检查 CPU 用来读取和写入数据记录的数量结构。

如果 CPU 上同时有多项指令需要读取或写入数据记录，那么用户程序发出的每项指令间都需要一定的间隙。

3.1 点对点

组态指令

在用户程序开始 PtP 通信之前，必须组态通信接口和收发数据的参数。

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 20)
- Send_Config (页 24)
- Receive_Config (页 26)
- P3964_Config (页 33)

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

用户程序使用这些指令实现点对点通信，向通信接口发送数据，并从中接收数据。CMs 向通信站发送数据，并从中接收数据。协议位于缓冲区中，通过特定的通信接口进行接收或发送。

- Send_P2P (页 35)
- Receive_P2P (页 38)

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 参数设置了 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，那么只有当 NDR = true 时，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 40)
- Signal_Get (页 41)
- Signal_Set (页 42)

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 44)
- **Set_Features** (页 45)

所有的 PtP 指令都不同步，也就是说，用户程序只有在指令完成后才会停止。因此，必须在输出参数 **DONE** 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。在到达模块特定的最大缓冲区大小前，通信模块会根据需要缓冲发送和接收的数据。

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。尽管如此，**CM** 可测量的最大时间段为 **8 秒**。

3.1 点对点

3.1.2 使用指令

查询已接收数据时必须循环调用点对点指令。发送过程的查询会向用户程序发出传输何时结束的信号。

根据数据量，数据传输可能需要多次调用（程序循环）才可完成。

以 `DONE = TRUE` 或 `NDR = TRUE` 处理的命令已无错执行。

说明

STATUS 备份

参数 `DONE`, `NDR`, `ERROR` 和 `STATUS` 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` 指令触发对 `CM` 的传输。
在 `REQ` 输入的上升沿开始传输数据。
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 步并重复循环。

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，则 `Receive_P2P` 指令将表示该新数据可用，并将该请求复制给 `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. 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 步并重复循环。

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。如果设置的 **OB** 周期时间能够确保两次运行在主站超时设置过程中发生，则用户程序可接收所有传输且不受损失。

3.1.3 点对点操作的共享参数。

表格 3-1 PtP 指令的通用输入参数

参数	说明
REQ	在 REQ 输入的上升沿开始传输数据。只有在命令执行完毕后（DONE 或 ERROR），才可在 REQ 中创建另一个沿。数据传输可能会进行多次调用（程序循环），具体取决于数据量。 向程序添加 PtP 指令时，STEP 7 将提示指定背景数据块（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	组态通信模块期间将分配端口地址。指令可通过 PORT 参数得到特定通信模块的分配信息。 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。

PtP 指令的输出参数 DONE、NDR、ERROR 和 STATUS 指示 PtP 功能的执行状态。

表格 3-2 输出参数 DONE、NDR、ERROR 和 STATUS

参数	数据类型	标准	说明
DONE	Bool	FALSE	一次循环中上一个完成的请求有错误就设置为 TRUE；否则设置为 FALSE。
NDR	Bool	FALSE	一次循环中接收到新数据就设置为 TRUE；否则设置为 FALSE。

3.1 点对点

参数	数据类型	标准	说明
ERROR	Bool	FALSE	一次循环中上一个完成的请求有错误就设置为 TRUE，可在 STATUS 中找到相应的错误代码；反之，否则设置为 FALSE。
STATUS	Word	0 或 0x7000	<p>结果状态：</p> <ul style="list-style-type: none"> • 如果位 DONE 或 NDR 置位，则 STATUS 将设置为 0/0x700 或一个特定的错误代码。 • 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。 <p>再次调用（使用同一个端口地址调用）该指令之前，STATUS 中的值始终是有效的。</p>

表格 3- 3 COM_RST 输入/输出参数

参数	数据类型	标准	说明
COM_RST	Bool	FALSE	<p>指令的初始化</p> <p>指令在为 TRUE 时执行。COM_RST 随后复位为 FALSE。</p> <p>注意： 必须在启动期间将 COM_RST 设置为 TRUE，随后不应更改此参数（即，调用指令时不要进行赋值）。初始化背景数据块后，指令将复位 COM_RST。</p>

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 3-4 共享错误代码

错误代码	说明
0x0000	无错误
0x7000	功能未激活
0x7001	请求后启动了初始调用。
0x7002	请求后启动了后续调用。
0x8x3A	参数 x 中的指针无效

表格 3-5 STATUS 参数的共享错误类别

类别说明	错误类别	说明
端口组态	0x81Ax	针对接口组态中常见错误的说明
发送组态	0x81Bx	针对发送组态中错误的说明
接收组态	0x81Cx	针对接收组态中错误的说明
发送	0x81Dx	针对发送期间运行时错误的说明
接收	0x81Ex	针对接收期间运行时错误的说明
RS232 伴随信号	0x81Fx	针对信号处理相关错误的说明

3.1 点对点

3.1.4 Port_Config: 组态 PtP 通信端口

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。
 Port_Config 组态更改将保存在 CM 中，而不是 CPU 中。设备组态中保存的参数将在 CPU 再次上电时恢复。

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	在此输入的上升沿激活组态更改。
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500)（输入地址； S7-300/400）的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
PROTOCOL	IN	UInt	Word	0	协议 <ul style="list-style-type: none"> • 0 = 点对点通信协议（自由端口） • 1 = 3964(R) 协议 • 2-n = 保留

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
BAUD	IN	UInt	Word	6	端口数据传输速率： <ul style="list-style-type: none"> • 1 = 300 Bit/s • 2 = 600 Bit/s • 3 = 1200 Bit/s • 4 = 2400 Bit/s • 5 = 4800 Bit/s • 6 = 9600 Bit/s • 7 = 19200 Bit/s • 8 = 38400 Bit/s • 9 = 57600 Bit/s • 10 = 76800 Bit/s • 11 = 115200 Bit/s
PARITY	IN	UInt	Word	1	端口的奇偶校验： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 1 = 8 个数据位 • 2 = 7 个数据位
STOPBITS	IN	UInt	Word	1	停止位： <ul style="list-style-type: none"> • 1 = 1 个停止位 • 2 = 2 个停止位

3.1 点对点

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
FLOWCTRL	IN	UInt	Word	1	流控制： <ul style="list-style-type: none"> • 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
XONCHAR	IN	Char		11H	指定用作 XON 字符的字符。通常为 DC1 字符 (11H)。仅当流控制处于活动状态时才评估此参数。
XOFFCHAR	IN	Char		13H	指定用作 XOFF 字符的字符。通常为 DC3 字符 (13H)。仅当流控制处于活动状态时才评估此参数。
WAITIME	IN	UInt	Word	2000	指定接收到 XOFF 字符后等待 XON 字符所需的时间或接收到 CTS = OFF (0 与 65535 ms 之间) 后等待 CTS = ON 所需的时间。仅当流控制处于活动状态时才评估此参数。
MODE	IN	UInt	Byte	0	工作模式 有效的工作模式包括： <ul style="list-style-type: none"> • 0 = 全双工 (RS232) • 1 = 全双工 (RS422) 四线制模式 (点对点) • 2 = 全双工 (RS422) 四线制模式 (多点主站) • 3 = 全双工 (RS422) 四线制模式 (多点从站) • 4 = 半双工 (RS485) 双线制模式

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
LINE_PRE	IN	USInt	Byte	0	接收线路初始状态 有效的初始状态是： <ul style="list-style-type: none"> • 0 = “无”初始状态 • 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。
CABLE_BRK	IN	USInt	Byte	0	断路检测 下列设置有效： <ul style="list-style-type: none"> • 0 = 断路检测已禁用 • 1 = 断路检测已激活
COM_RST	IN/OUT	---	Bool	FALSE	Port_Config 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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	如果上一个请求完成但存在错误，则为 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 95)）

3.1 点对点

3.1.5 Send_Config: 组态 PtP 发送方

说明

Send_Config 指令（发送组态）允许在运行期间更改串行传输参数（使用程序）。CM 中未决的任何数据将在执行 Send_Config 时删除。

Send_Config 组态更改将保存在 CM 中，而不是 CPU 中。设备组态中保存的参数将在 CPU 再次上电时恢复。

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，激活组态更改。
PORT	IN	PORT (UInt)	Word	0	组态完 CM 后，便可在设备组态 (S7-1200/1500)（输入地址；S7-300/400）的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。在 S7 300/400/WinAC 系统中，必须将 HWCN 中分配的输入地址分配给 CM 端口。
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。
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 BREAK。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。

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
USR_END	IN	STRING[2]		0	输入结束符。 最多可组态 2 个结束符。 发送包括结束符在内的所有数据，与所组态的帧长度无关。
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	Send_Config 指令的初始化 指令在 TRUE 时执行。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	如果上一个请求完成但存在错误，则为 TRUE 并保持一个周期
STATUS	OUT	Word		16#700 0	错误代码（请参见 错误消息 (页 95)）

3.1 点对点

3.1.6 Receive_Config: 组态 PtP 接收方

说明

Receive_Config 指令（接收组态）允许在运行期间更改串行接收参数（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。

Port_Config 组态更改将保存在 CM 中，而不是 CPU 中。设备组态中保存的参数将在 CPU 再次上电时恢复。

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，激活组态更改。
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
Receive_Conditions	IN	Variant	Any	-	Receive_Conditions 的数据结构指定用于识别帧的开始和结束的开始和结束条件。
COM_RST	IN/OUT	---	Bool	FALSE	Receive_Konfig 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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	如果上一个请求完成但存在错误，则为 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 95)）

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 指令确定点对点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。如果指定多个开始条件，则必须满足所有这些条件数据传输才能开始。

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

表格 3-6 开始条件的 Receive_Conditions 的结构

参数	声明	数据类型	标准	说明
START.STARTCOND	IN	Word	02H	指定开始条件 <ul style="list-style-type: none"> • 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	28H	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“线路空闲”(Line in idle) 条件关联时。 0 到 FFFF
START.STARTCHAR	IN	Byte	02H	条件“起始字符”(Start character) 的起始字符。（默认值：B#16#2）

3.1 点对点

参数	声明	数据类型	标准	说明
START.SEQ[1].CTL	IN	Byte	0	<p>开始序列 1，禁用/激活每个字符的比较：（默认值：B#16#0）</p> <p>这些是起始字符串的每个字符的激活位。</p> <ul style="list-style-type: none"> • 01H - 字符 1 • 02H - 字符 2 • 04H - 字符 3 • 08H - 字符 4 • 10H - 字符 5 <p>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）。</p>
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，忽略/比较每个字符的控制。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，忽略/比较每个字符的控制。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，忽略/比较每个字符的控制。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。

表格 3-7 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	68H	xx	xx	68H	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。

用户可以在设备组态的通信接口属性中组态结束条件，或者使用 Receive_Config 指令组态结束条件。每次 CPU 上电时，接收参数（开始和结束条件）将重置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。

3.1 点对点

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

表格 3-8 结束条件的 Receive_Conditions 的结构

参数	声明	数据类型	标准	说明
END.ENDCOND	IN	Word	0	此参数指定帧结束的条件： <ul style="list-style-type: none"> • 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“固定长度”(Fixed length) 时使用。 1 到 4000 个字节（取决于模块）
END.MAXLEN	IN	Word	1	最大帧长度：仅当选择结束条件“最大长度”(Maximum length) 时使用。 1 到 4000 个字节（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（取决于模块）
END.LENGTHSIZE	IN	Word	0	字节字段的大小（1、2 或 4 字节）。仅与结束条件 N+LEN+M 一起使用。
END.LENGTHM	IN	Word	0	输入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	指定接收到的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。此信息仅与条件“响应时间”(Response time) 结合使用。（0 到 65535 ms）。 注意：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	标准	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。只有选择“帧时间限制用完后”条件时，才会使用该参数。（0 到 65535 ms）
END.CHARGAP	IN	Word	12	输入字符间的位时间数。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“字符间隔”(Character interval) 结合使用。（0 到 65535 位时间）
END.SEQ.CTL	IN	Byte	0	字符串 1，忽略/比较每个字符的控制： 这些是结束字符串的每个字符的激活位。字符 1 为位 0，字符 2 为位 1，...，字符 5 为位 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。
END.SEQ.STR[1] .. END.SEQ.STR[5]	IN	Char[5]	0	字符串 1，起始字符（5 个字符）

3.1 点对点

表格 3-9 Receive_P2P 指令的通用参数

参数	声明	数据类型	标准	说明
GENERAL.MBUF_SIZE	IN	Byte	255	指定要在 CM 接收缓冲区中缓冲的帧数。 超出要保存的帧数之后接收到的任何帧都将被丢弃。(1 到 255 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。 <ul style="list-style-type: none"> • 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。 <ul style="list-style-type: none"> • 0 - 未激活 • 1 - 已激活

3.1.7 P3964_Config: 组态 3964 (R) 协议

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 组态更改将保存在 CM 中，而不是 CPU 中。设备组态中保存的参数将在 CPU 再次上电时恢复。

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，激活组态更改。
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500)（输入地址；S7-300/400）的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
BCC	IN	SInt	Byte	1	启用/禁用块检查 <ul style="list-style-type: none"> • 0 = 不带块检查 • 1 = 带块检查
Priority	IN	SInt	Byte	1	选择优先级 <ul style="list-style-type: none"> • 0 = 优先级低 • 1 = 优先级高
CharacterDelayTime	IN	UInt	Word	DCH	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 到 65535 ms
AcknDelayTime	IN	UInt	Word	7D0H	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 到 65535 ms

3.1 点对点

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
BuildupAttempts	IN	SInt	Byte	6H	设置连接尝试次数 1 到 255
RepetitionAttempts	IN	SInt	Byte	6H	设置传输尝试次数 1 到 255
COM_RST	IN/OUT	---	Bool	FALSE	P3964_Config 指令的初始化指令在 TRUE 时执行。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	如果上一个请求完成但存在错误，则为 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 95)）

3.1.8 Send_P2P: 发送数据

说明

Send_P2P 指令（发送点对点数据）启动数据传输并向通信模块传输分配的缓冲区中的内容。在 CM 以指定数据传输速率发送数据的同时，CPU 程序会继续运行。任何时刻，每个通信模块都只能有一条发送指令处于待命状态。当 CM 已经开始发送帧时，如果执行第二个 Send_P2P 指令，则 CM 将输出错误。

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	在此使能输入的上升沿使能请求的传输。这会将缓冲区中的内容传送到点对点通信接口。
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500) (S7-300/400; 输入地址) 的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
BUFFER	IN	Variant	Any	0	此参数指向发送缓冲区的起始地址。 注意： 不支持布尔数据和布尔字段。
LENGTH	IN	UInt	Word	0	要传输的数据长度（字节）
COM_RST	IN/OUT	---	Bool	FALSE	Send_P2P 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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	如果上一个请求完成但存在错误，则为 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 95)）

3.1 点对点

参数

如果发送指令正在执行，那么输出状态 DONE 和 ERROR 将为 FALSE。发送指令结束时，DONE 或 ERROR 输出会设为 TRUE 以表示发送指令的状态。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，指令将输出状态 16#7001。如果 CM 仍在发送，随后执行的 Send_P2P 将输出值 16#7002。发送指令结束时，CM 将输出发送指令的状态 16#0000（如果未发生错误）。随后执行的 Send_P2P REQ = 0 将输出状态 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）。

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

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 参数如何只在一个周期内有效。

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

下表列出了发生错误时，参数 DONE、ERROR 和 STATUS 之间的关系。

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

仅在同一背景数据块再次执行 Send_P2P 前，DONE、ERROR 和 STATUS 值才有效。

3.1.9 在通信操作中使用 LENGH 和 BUFFER 参数

Send_P2P 的 LENGTH 与 BUFFER 参数交互

Send_P2P 指令发送的最小数据大小为 1 字节。BUFFER 参数指定要发送的数据大小。无法对 BUFFER 参数使用 Bool 数据类型或 Bool 类型的数组。

表格 3- 10 LENGTH 参数

LENGTH	说明
> 0	传送已组态的字节数。
= 0	传送使用 BUFFER 寻址的发送缓冲区中的全部内容。 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。

表格 3- 11 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	对于优化存储器：BUFFER 可能的最大长度为 1024 Byte。 对于发送：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。
Array	对于优化存储器：如果数组数据类型不等于 Byte、Word 或 DWord，则可能的最大缓冲区长度为 1024 Byte。 对于发送：LENGTH 值必须包括小于数组完整字节长度的字节大小；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。如果已指定 LENGTH，则以 LENGTH 字节为单位传输数组元素的数目。例如，如果 BUFFER 包含一个具有 15 个 DWord 元素（总计 60 Bytes）的数组并指定 LENGTH = 20，则传输数组前 5 个 DWord 元素。
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。

3.1 点对点

3.1.10 Receive_P2P: 接收数据

说明

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

参数

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500) (S7-300/400; 输入地址) 的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
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。 注意： 不支持布尔数据或布尔字段。
COM_RST	IN/OUT	---	Bool	FALSE	Receive_P2P 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
LENGTH	OUT	UInt	Word	0	接收到的帧的长度（以字节为单位）
NDR	OUT	Bool		FALSE	如果新数据可用且指令无错完成，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 并保持一个周期。
STATUS	OUT	Word		16#700 0	错误代码（请参见 错误消息 (页 95)）

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 CM 中终止接收操作的原因。它通常是正值，表示接收操作成功且已检测到帧标准。

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 **PtP** 通信接口最多可缓冲模块特定的最大字节数。如果 **CM** 中存在多个可用帧，则 **Receive_P2P** 指令将输出最早的可用帧。一旦获取到最早的帧，则第二早的帧将变为最早的帧。

3.1 点对点

3.1.11 Receive_Reset: 清除接收缓冲区

说明

Receive_Reset 指令（复位接收器）会清除 CM 中的接收缓冲区。

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	通过此使能输入的上升沿激活接收缓冲区清除功能
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500) (S7-300/400; 输入地址) 的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
COM_RST	IN/OUT	---	Bool	FALSE	Receive_Reset 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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 意味着上次请求完成但存在错误。如果此输出为 TRUE，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 95)）

3.1.12 Signal_Get: 读取状态

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。

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	以此输入的上升沿获取 RS232 信号状态
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500) (S7-300/400; 输入地址) 的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
COM_RST	IN/OUT	---	Bool	FALSE	Signal_Get 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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 并保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见 错误消息 (页 95)）
DTR	OUT	Bool		FALSE	数据终端就绪，模块就绪（输出）
DSR	OUT	Bool		FALSE	数据记录准备就绪，通信站准备就绪（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	检测到数据存储介质，收到信号电平（总为 False，不支持）
RING	OUT	Bool		FALSE	呼叫显示，呼入消息（总为 False，不支持）

3.1 点对点

3.1.13 Signal_Set: 设置伴随信号

说明

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

此功能只对 RS232 CM 有效。

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	用于设置 RS232 信号的指令以此输入的上升沿开始。
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500) (S7-300/400; 输入地址) 的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
SIGNAL	IN	Byte		0	选择要设置的信号（多种可能）： <ul style="list-style-type: none"> • 01H = RTS • 02H = DTR • 04H = DSR
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	数据记录准备就绪（仅适用于 DCE 接口类型），未使用。

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
COM_RST	IN/OUT	---	Bool	FALSE	Signal_Set 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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	如果上一个请求完成但存在错误，则为 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 95)）

3.1 点对点

3.1.14 Get_Features: 获取扩展功能

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		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
REQ	IN	Bool		FALSE	当此输入存在上升沿时，用于获取扩展功能的指令开始。
PORT	IN	PORT (UInt)	Word	0	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500) (S7-300/400; 输入地址) 的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
COM_RST	IN/OUT	---	Bool	FALSE	Get_Features 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
NDR	OUT	Bool		FALSE	如果新数据可用且指令无错完成，则为 TRUE 并保持一个周期
MODBUS_CRC	OUT	Bool		FALSE	Modbus CRC 支持
DIAG_ALARM	OUT	Bool		FALSE	生成诊断消息
ERROR	OUT	Bool		FALSE	如果指令完成但出现错误，则为 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 95)）

3.1.15 Set_Features: 设置扩展功能

说明

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

参数

参数	声明	数据类型		标准	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	安装并组态完 CM 后，便可在设备组态 (S7-1200/1500) (S7-300/400; 输入地址) 的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
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
COM_RST	IN/OUT	---	Bool	FALSE	Set_Features 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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 并保持一个周期
STATUS	OUT	Word		16#7000	错误代码（请参见 错误消息 (页 95)）

3.2 Modbus

3.2.1 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 3-12 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）的功能 - 标准寻址
01	读取输出位：每个请求 1 到 1992 位
02	读取输入位：每个请求 1 到 1992 位
03	读取保持寄存器：每个请求 1 到 124 个字
04	读取输入字：每个请求 1 到 124 个字

表格 3-13 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	用于写入从站（服务器）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 到 1960 位
16	写入一个或多个保持寄存器：每个请求 1 到 122 个字

- Modbus 功能代码 08 和 11 提供从站设备的通信诊断信息。
- 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 3- 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，与一个 Modbus 主站设备进行通信。

3.2 Modbus

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

说明

Modbus_Comm_Load 指令通过 Modbus-RTU 协议对用于通信的通信模块进行组态。当在程序中添加 Modbus_Comm_Load 指令时，将自动分配背景数据块。

参数

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
REQ	IN	Bool		FALSE	指令起始于上升沿（0 到 1）。
PORT	IN	Port	Laddr	0	安装并组态完 CM 后，便可在设备组态的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
BAUD	IN	UDInt	DWord	6	选择数据传输速率： 1 = 300, 2 = 600, 3 = 1200, 4 = 2400, 5 = 4800, 6 = 9600, 7 = 19200, 8 = 38400, 9 = 57600, 10 = 76800, 11 = 115200 bit/s。所有其它值无效。
PARITY	IN	UInt	Word	1	选择奇偶校验： <ul style="list-style-type: none"> • 0 – 无 • 1 – 奇校验 • 2 – 偶校验
FLOW_CTRL	IN	UInt	Word	0	选择流控制： <ul style="list-style-type: none"> • 0 – （默认）无流控制 • 1 – 硬件流控制，RTS 始终开启（不适用于 RS485-CM） • 2 – 带 RTS 切换的硬件流控制

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： <ul style="list-style-type: none"> • 0 –（默认）从“RTS 激活”一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 RS485 CM）。无论是否选择 FLOW_CTRL，都必须使用 RTS 延迟。
RTS_OFF_DLY	IN	UInt	Word	0	RTS 关断延迟选择： <ul style="list-style-type: none"> • 0 - 从传送上一个字符一直到“RTS 未激活”之前无延迟 • 1 到 65535 - 从传送上一个字符一直到“RTS 未激活”之前的延迟（以毫秒表示）（不适用于 RS485 端口）。无论是否选择 FLOW_CTRL，都必须使用 RTS 延迟。
RESP_TO	IN	UInt	Word	1000	响应超时： <p>5 ms 到 65535 ms - Modbus_Master 等待从站响应的的时间（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）。</p>
MB_DB	IN/OUT	MB_BASE		-	对 Modbus_Master 或 Modbus_Slave 指令的背景数据块的引用。 <p>必须以 Modbus_Master 或 Modbus_Slave 指令的 MB_DB 参数替换 MB_DB 参数（静态，因此在指令中不可见）。</p>
COM_RST	IN/OUT	---	Bool	FALSE	Modbus_Comm_Load 指令的初始化 <p>该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。</p>

3.2 Modbus

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
DONE	OUT	Bool		FALSE	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个周期时间。
ERROR	OUT	Bool		FALSE	上一请求已完成且出错后，ERROR 位将保持为 TRUE 一个周期时间。STATUS 参数的错误代码只有在 ERROR = TRUE 的循环中有效。
STATUS	OUT	Word		16#7000	错误代码（请参见 错误消息 (页 95)）

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，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。需要将指令 `Modbus_Comm_Load` 的 `MB_DB` 参数连接到指令 `Modbus_Master` 或指令 `Modbus_Slave` 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 3- 15 背景数据块中的静态变量

变量	数据类型	标准	说明
ICHAR_GAP	Word	0	字符之间字符间距的延迟。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_DIAG_ALARM	Word	0	激活诊断中断： <ul style="list-style-type: none"> • 0 - 未激活 • 1 - 已激活
MODE	USInt	0	工作模式 有效的工作模式包括： <ul style="list-style-type: none"> • 0 = 全双工 (RS232) • 1 = 全双工 (RS422) 四线制模式（点对点） • 2 = 全双工 (RS422) 四线制模式（多点主站） • 3 = 全双工 (RS422) 四线制模式（多点从站） • 4 = 半双工 (RS485) 双线制模式
LINE_PRE	USInt	0	接收线路初始状态 有效的初始状态是： <ul style="list-style-type: none"> • 0 = “无”初始状态 • 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。

3.2 Modbus

3.2.3 Modbus_Master: 作为 Modbus 主站进行通信

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。

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 247，0 用于 Broadcast） 扩展地址范围（1 到 65535，0 用于 Broadcast） 值 0 为将帧广播到所有 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 功能表中提供了其它信息。
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。

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
COM_RST	IN/OUT	---	Bool	FALSE	Modbus_Master 指令的初始化指令在 TRUE 时执行。随后会将 COM_RST 复位为 FALSE。
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。
DONE	OUT	Bool		FALSE	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个周期时间。
BUSY	OUT	Bool		-	<ul style="list-style-type: none"> • FALSE – Modbus_Master 无激活命令 • TRUE – Modbus_Master 命令执行中
ERROR	OUT	Bool		FALSE	上一请求已完成且出错后，ERROR 位将保持为 TRUE 一个周期时间。STATUS 参数的错误代码只有在 ERROR = TRUE 的循环中有效。
STATUS	OUT	Word		0	错误代码（请参见 错误消息 (页 95)）

3.2 Modbus

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 3- 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	0.5	<p>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。</p> <p>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”。</p>
Extended_Addresssing	Bool	FALSE	<p>将从站地址组态为单字节或双字节。</p> <ul style="list-style-type: none"> • 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	<p>Modbus 的 CP 341 和 CP 441-2 以及驱动器 Modbus RTU 和 ET 200SP 1SI 兼容模式。默认值为 0。</p> <ul style="list-style-type: none"> • FALSE = 根据 Modbus 规范，不兼容 • TRUE = 兼容 <ul style="list-style-type: none"> - 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。

变量	数据类型	标准	说明
MB_DB	MB_BA SE	-	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。要保留 Modbus 的标准 CP 341, CP 441-2 和 ET 200SP 1SI 响应, 使用“Compatibility_Mode”参数。

程序可以向 Blocked_Proc_Timeout 和 Extended_Adressing 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_Master¹⁾ 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/时间控制的执行级别中执行。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 指令相同的互连, 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

当 MODE=0 且 DATA_ADDR=10 时调用 Modbus_Master

此命令在以 DONE=1 或 ERROR=1 完成, 或在超出 Blocked_Proc_Timeout 参数中组态的看门狗时间前, 将一直处于激活状态。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。

3.2 Modbus

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，将会输出错误消息 **0x8200**（接口正忙于处理当前请求）。

如果无法完整地完调用，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

在此使能输入的上升沿使能请求的传输。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 3-17 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
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

3.2 Modbus

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	2	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	在此功能中忽略了 Modbus_Master 的 DATA_ADDR 和 DATA_LEN 两个操作数。						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				-

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 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 允许用户使用 Modbus 功能 15 和 16 写入一个或更多的输出位和保持寄存器。
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。

DATA_PTR 参数

DATA_PTR 参数指向可在其中写入或从中读取的数据块或标记地址。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。

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。

3.2 Modbus

DATA_PTR 参数的数据块结构

- 这些数据类型适用于在 Modbus 地址范围 (DATA_PTR) 30001 到 39999、40001 到 49999 和 400001 到 465535 内**读**字，以及在 Modbus 地址范围 (DATA_PTR 参数) 40001 到 49999 和 400001 到 465535 内**写**字。
 - 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。
- 针对在 Modbus 地址范围 (DATA_PTR 参数) 00001 到 09999 内**读**取并写入位，以及从 10001 到 19999 内**读**取位。
 - 布尔数据类型的标准字段。
 - 来自唯一指定的布尔变量的指定布尔结构。
- 给每个 Modbus_Master 指令分配各自的独立存储区并不是必须进行的操作，但我们建议进行此操作。原因是，如果多个 Modbus 指令都在同一个存储区读取和写入，那么数据损坏的可能性将大大提高。
- 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。

参见

Modbus_Slave: 作为 Modbus 从站进行通信 (页 61)

3.2.4 Modbus_Slave: 作为 Modbus 从站进行通信

说明

Modbus_Slave 指令允许您的程序通过 CM 的 PtP 端口（RS422/485 或 RS232）作为 Modbus 从站进行通信。STEP 7 在您添加此指令时自动生成一个背景数据块。

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。

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 从站的站地址： 标准寻址范围（1 到 247） 扩展寻址范围（0 到 65535） 注意：0 是广播地址
COM_RST	IN/OUT	---	Bool	FALSE	Modbus_Slave 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
MB_HOLD_REG	IN/OUT	Variant	Any	-	指向 Modbus 保持寄存器 DB 的指针： Modbus 保持寄存器可能为标志或数据块的存储区。
NDR	OUT	Bool		FALSE	可用的新数据： <ul style="list-style-type: none"> • 0 - 无新数据 • 1 - 表示新数据已由 Modbus 主站写入上一请求已完成且没有出错后，NDR 位将保持为 TRUE 一个周期时间。
DR	OUT	Bool		FALSE	读取数据： <ul style="list-style-type: none"> • 0 - 未读取数据 • 1 - 表示数据已由 Modbus 主站读取上一请求已完成且没有出错后，DR 位将保持为 TRUE 一个周期时间。

3.2 Modbus

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
ERROR	OUT	Bool		FALSE	上一请求已完成且出错后，ERROR 位将保持为 TRUE 一个周期时间。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 95)）

Modbus 通信的功能代码（1、2、4、5 和 15）可直接在 CPU 的输入过程映像和输出过程映像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 3- 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	输出的过程映像	O0.0	到	O1023.7
02	读取位	输入	0	到	8191	输入的过程映像	I0.0	到	I1023.7
04	读取字	输入	0	到	511	输入的过程映像	IW0	到	IW1022
05	写入位	输出	0	到	8191	输出的过程映像	O0.0	到	O1023.7
15	写入位	输出	0	到	8191	输出的过程映像	O0.0	到	O1023.7

表格 3- 19 将 Modbus 地址分配给过程映像

Modbus 功能					S7-1500 / S7-300 / S7-400				
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	输出的过程映像	O0.0	到	O1248.6
02	读取位	输入	0	到	9998	输入的过程映像	I0.0	到	I1248.6
04	读取字	输入	0	到	9998	输入的过程映像	IW0	到	IW19996
05	写入位	输出	0	到	9998	输出的过程映像	O0.0	到	O1248.6
15	写入位	输出	0	到	9998	输出的过程映像	O0.0	到	O1248.6

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，此寄存器是标志或数据块的存储区中的一个地址区。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 - 数据块的类型 MB_HOLD_REG**

具有 Modbus 保持寄存器的数据块必须允许直接（绝对）寻址和符号寻址。

表格 3-20 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	回应测试的输出请求数据： 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。它也会随导致通信错误的请求而递增（例如，奇偶校验或 CRC 错误）。

Modbus_Slave 指令支持来自 Modbus 主站的广播写入请求，只要请求包括到有效地址的访问。Modbus_Slave 指令将为广播功能不支持的功能代码生成错误代码 0x8188。

3.2 Modbus

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 3-21 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）
Extended_Addressing	Bool	FALSE	扩展寻址，将从站寻址组态为单字节或双字节。 （0 = 单字节地址，1 = 双字节地址，默认 = 0）
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	需要返回例外的 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。

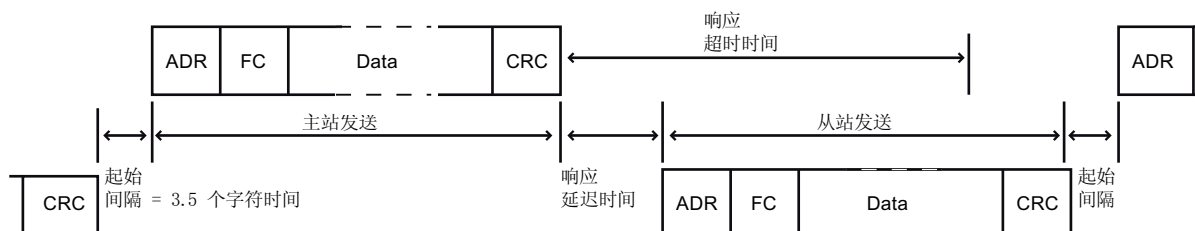
程序可以向 HR_Start_Offset 和 Extended_Addressing 变量中写入值来控制 Modbus 从站的操作。可读取其它变量以监视 Modbus 状态。

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 从站开始响应的的时间。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。但是，您可以组态 `HR_Start_Offset` 变量，为 Modbus 保持寄存器设置除 40001 或 400001 以外的起始地址。

例如，可以组态从 MW100 开始、长度为 100 字的保持寄存器。通过偏移量 20，您可以输入起始地址为 40021 的保持寄存器，而不是 40001。40021 下方和 400119 上方的每个地址都会导致寻址错误。

表格 3-22 示例：当 `DATA_PTR` 为 MW100 的指针时寻址 Modbus 保持寄存器

HR_Start_Offset	地址	最小	最大
0	Modbus 地址（字）	40001	40099
	S7-1500 地址	MW100	MW298
20	Modbus 地址（字）	40021	40119
	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。

3.2 Modbus

使用标准 DB 名称输入 Modbus 从站变量：

1. 将光标置于参数字段 OUT1 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”。

3.2.5 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。

可以组态单字节（Modbus 标准）或双字节来寻址 Modbus 从站。扩展寻址用于在单个网络中寻址超过 247 个设备。如果决定使用扩展寻址，则可以寻址最多 65535 个地址。以下示例显示了 Modbus 功能 1 的帧。

表格 3-23 大小为一个字节的从站地址（字节 0）

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

表格 3-24 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		E 代码	起始地址		数据	
有效响应	从站地址		E 代码	长度	数据...		
错误消息	从站地址		0xxx	E 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 CRC 校验码。

功能代码 1 - 此功能允许读取各个输出位

表格 3- 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	功能代码	起始地址		输出数目	
有效响应	从站地址	功能代码	长度 ¹	输出数据 ³⁾		
错误消息	从站地址	0x81	E 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

	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		功能代码	起始地址		输出数目	
有效响应	从站地址		功能代码	长度 ¹	输出数据		
错误消息	从站地址		0x81	E 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

3.2 Modbus

功能代码 2 - 此功能允许读取各个输入位

表格 3-26 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	功能代码	起始地址		输入数目	
有效响应	从站地址	功能代码	长度 ¹	输入数据		
错误消息	从站地址	0x82	E 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		功能代码	起始地址		输入数目	
有效响应	从站地址		功能代码	长度 ¹	输入数据		
错误消息	从站地址		0x82	E 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 3-27 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	功能代码	起始地址		寄存器数	
有效响应	从站地址	功能代码	长度 ¹	寄存器数据		
错误消息	从站地址	0x83	E 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		功能代码	起始地址		寄存器数	
有效响应	从站地址		功能代码	长度 ¹	寄存器数据		
错误消息	从站地址		0x83	E 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 3- 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	功能代码	起始地址		输入字数目：	
有效响应	从站地址	功能代码	长度 ¹	输入数据		
错误消息	从站地址	0x84	E 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		功能代码	起始地址		输入字数目：	
有效响应	从站地址		功能代码	长度 ¹	输入数据		
错误消息	从站地址		0x84	E 代码 ²		---	

1 长度：2 * 输入字数目

2 E 代码：01 或 02 或 03 或 04

3.2 Modbus

功能代码 5 - 此功能可以设置或删除各个位

表格 3-29 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	功能代码	起始地址		值	
有效响应	从站地址	功能代码	长度	值		
错误消息	从站地址	0x85	E 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		功能代码	起始地址		值	
有效响应	从站地址		功能代码	长度	值		
错误消息	从站地址		0x85	E 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 3-30 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	功能代码	地址		标签	
有效响应	从站地址	功能代码	地址		标签	
错误消息	从站地址	0x86	E 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		功能代码	地址		标签	
有效响应	从站地址		功能代码	地址		标签	
错误消息	从站地址		0x86	E 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 3-31 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	功能代码	诊断代码		测试值	
有效响应	从站地址	功能代码	诊断代码		测试值	
错误消息	从站地址	0x88	E 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		功能代码	诊断代码		测试值	
有效响应	从站地址		功能代码	诊断代码		测试值	
错误消息	从站地址		0x88	E 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 3-32 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	功能代码	---			
有效响应	从站地址	功能代码	状态		事件计数器	
错误消息	从站地址	0x8B	E 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		功能代码	---			
有效响应	从站地址		功能代码	状态		事件计数器	
错误消息	从站地址		0x8B	E 代码 ¹	---		

¹ E 代码: 01 或 04

3.2 Modbus

功能代码 15 - 此功能允许写入多个位

表格 3- 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	功能代码	起始地址		输出字数目		字节计数器 ¹	值	
有效响应	从站地址	功能代码	起始地址		输出字数目		---		
错误消息	从站地址	0x8F	E 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	功能代码	起始地址		输出字数目		字节计数器 ¹	值		
有效响应	从站地址	功能代码	起始地址		输出字数目		---			
错误消息	从站地址	0x8F	E 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 3-34 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	功能代码	起始地址		寄存器数		字节计数器 ¹	值	
有效响应	从站地址	功能代码	起始地址		寄存器数		---		
错误消息	从站地址	0x90	E 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	功能代码	起始地址		寄存器数		字节计数器 ¹	值		
有效响应	从站地址	功能代码	起始地址		寄存器数		---			
错误消息	从站地址	0x90	E 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

3.3 USS

3.3 USS

3.3.1 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。每个 RS485 端口最多可运行 16 个变频器。

USS 协议使用主站-从站网络进行基于串行总线的通信。主站使用地址参数向所选从站发送数据。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。

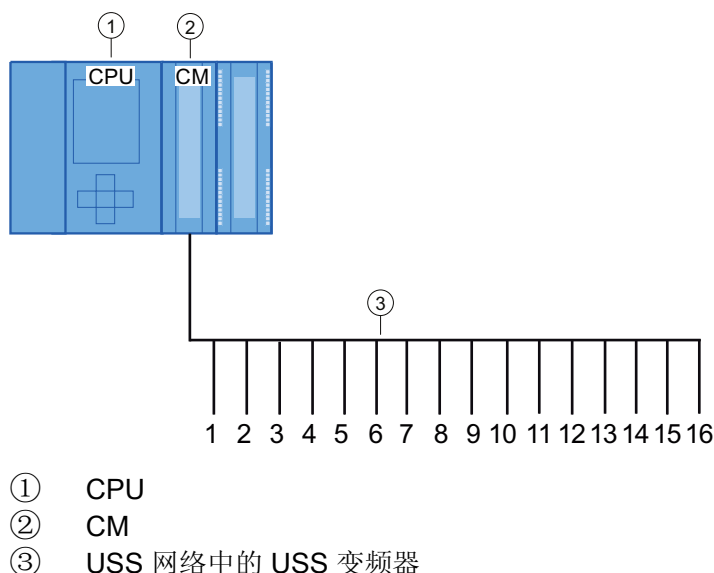


图 3-1 S7-1500 通信模块接线示例

说明

使用 RS232 与变频器通信

原则上，CM PtP RS232 BA 和 CM PtP RS232 HF 也可用于与变频器通信。但只可将一台变频器连接到 RS232 端口。

程序中的 USS 指令

- **USS_Port_Scan** 指令允许在 USS 网络中通过通信模块与最多 16 个变频器进行通信（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** 中为变频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上一请求中从 **USS_Port_Scan** 收到的数据。

- **USS_Read_Param**: **USS_Read_Param** 指令允许从变频器中读取参数。
- **USS_Write_Param**: **USS_Write_Param** 指令允许用户更改变频器中的参数。

3.3 USS

3.3.2 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_Control 的所有调用。

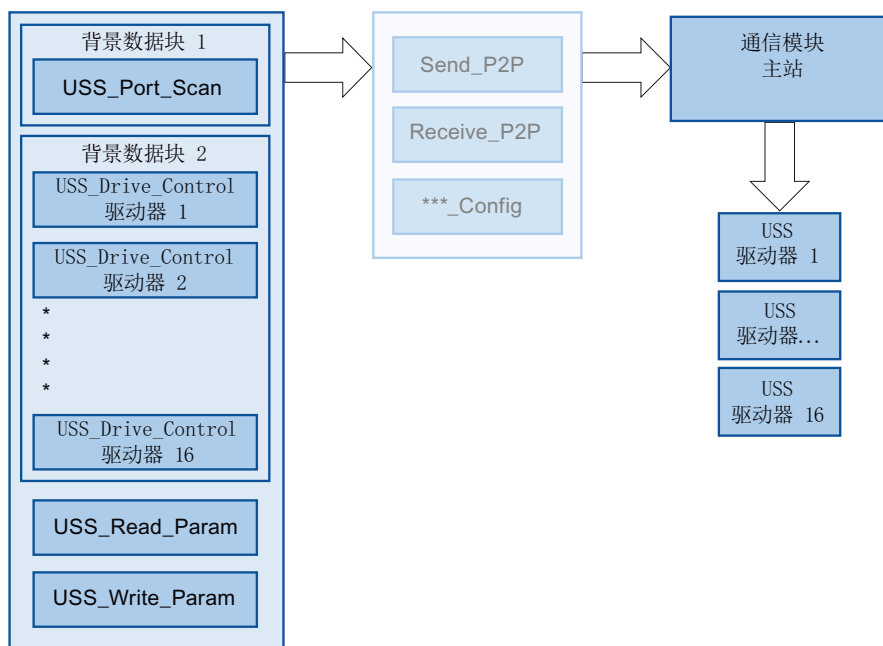


图 3-2 USS 程序顺序

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_Control，属于 USS 网络的所有指令都必须共享此背景数据块。对于此函数，USS_Port_Scan, USS_Read_Param 和 USS_Write_Param 具有参数 USS_DB。此参数必须连接到 USS_Drive_Control 指令背景数据块的（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_Port_Scan` 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 (PtP) 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。只能从主程序的循环 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。

3.3 USS

计算与变频器的通信时间

与变频器进行的通信与 S7-1500 的周期不同步。与变频器的通信完成前，S7-1500 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。

两次发送帧的时间间隔按如下公式计算：

$$(N * 5 * \text{周期时间} + 2 * \text{帧运行时间}) * \text{重试次数} + (\text{已接收帧的超时}) * (\text{重试次数} - 1)$$

N	该网络中的变频器数量
因数 5	发送和接收帧需要 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。
2 * 帧时间	每次发送和接收时进行一次
帧运行时间	帧运行时间 = (每帧的字符数) * (11 Bit/每字符) / (以 Bit/s 为单位的传输速率)
重试次数	3

下列时间适用于“已接收帧的超时”

波特率 = 115200:

```

Receive_Conditions.END.RCVTIME := 25;
Receive_Conditions.END.MSGTIME := 25;
ELSIF BAUD = 57600 THEN
Receive_Conditions.END.RCVTIME := 29;
Receive_Conditions.END.MSGTIME := 29;
ELSIF BAUD = 38400 THEN
Receive_Conditions.END.RCVTIME := 33;
Receive_Conditions.END.MSGTIME := 33;
ELSIF BAUD = 19200 THEN
Receive_Conditions.END.RCVTIME := 56;
Receive_Conditions.END.MSGTIME := 56;
ELSIF BAUD = 9600 THEN
Receive_Conditions.END.RCVTIME := 72;
Receive_Conditions.END.MSGTIME := 72;
ELSIF BAUD = 4800 THEN

```

```
Receive_Conditions.END.RCVTIME := 100;  
Receive_Conditions.END.MSGTIME := 124;  
ELSIF BAUD = 2400 THEN  
Receive_Conditions.END.RCVTIME := 100;  
Receive_Conditions.END.MSGTIME := 240;  
ELSIF BAUD = 1200 THEN  
Receive_Conditions.END.RCVTIME := 100;
```

示例:

5 个变频器

数据传输速率 = 9600 bps

每帧 28 个字符

周期时间 = 0.020 sec

时间间隔 = $(5 * 5 * 0.02 \text{ sec} + 2 * 28 * 11 / 1200 \text{ sec}) * 3 = 1.69 \text{ sec}$

这种情况下，变频器的看门狗时间需要设置为约 2 秒。

3.3 USS

3.3.3 USS_Port_Scan: 通过 USS 网络进行通信

说明

USS_Port_Scan 指令通过 USS 网络处理通信。STEP 7 在添加指令时会自动生成数据块。

说明

更改通信模块的使用方式

如果将通信模块用于 Modbus 后还想用于 USS 协议时，首先要关闭电源，然后再打开（复位 Modbus_Comm_Load 设置）。

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	安装并组态完 CM 后，便可在设备组态的“硬件 ID”(Hardware ID) 属性中找到 CM 端口值。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
BAUD	IN	DInt		9600	USS 通信的数据传输速率 以下内容有效： <ul style="list-style-type: none"> • 1200 Bit/s • 2400 Bit/s • 4800 Bit/s • 9600 Bit/s • 19200 Bit/s • 38400 Bit/s • 57600 Bit/s • 115200 Bit/s
USS_DB	INOUT	USS_BASE		-	USS_DB 参数必须连接到背景数据块的（静态）参数 USS_DB，这是在向程序中添加 USS_Drive_Control 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		
COM_RST	INOUT	---	Bool	FALSE	USS_Port_Scan 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
ERROR	OUT	Bool		FALSE	如果为 TRUE，则此输出表示发生错误且 STATUS 输出适用。
STATUS	OUT	Word		0	请求的状态值显示循环或初始化的结果。可在“USS_Extended_Error”变量中找到有关某些状态的更多信息（请参见 错误消息 (页 95)）。

程序中每个 PtP 通信端口只有一条 USS_Port_Scan 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 指令，以防止变频器超时（请参见 USS 协议使用要求 (页 76)“计算与变频器通信的时间”）。

通常从循环中断 OB 调用 USS_Port_Scan 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。

3.3 USS

USS_Port_Scan 数据块变量

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

表格 3- 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	<p>工作模式</p> <p>有效的工作模式包括：</p> <ul style="list-style-type: none"> • 0 = 全双工 (RS232) • 1 = 全双工 (RS422) 四线制模式（点对点） • 2 = 全双工 (RS422) 四线制模式（多点主站） • 3 = 全双工 (RS422) 四线制模式（多点从站） • 4 = 半双工 (RS485) 双线制模式
LINE_PRE	USInt	2	<p>接收线路初始状态</p> <p>有效的初始状态是：</p> <ul style="list-style-type: none"> • 0 = “无”初始状态 • 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。
RETRIES_MAX	SInt/Byte	2	<p>发生通信错误时的重试次数。</p> <p>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。</p>

3.3.4 USS_Drive_Control: 准备并显示变频器数据

说明

指令 USS_Drive_Control 准备变频器的发送数据并评估变频器的响应数据。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 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 自动创建数据块。

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，则电机将滑行至静止。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，则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，则此位将通过制动变频器产生快速停止。
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）。
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。默认值为 2。

3.3 USS

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
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 指令的初始化 该指令以 1 进行初始化。该指令随后将 COM_RST 复位为 0。
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 输出适用。出错时所有其它输出都置零。只在 USS_Port_Scan 指令的 ERROR 和 STATUS 输出发出通信错误信号。
STATUS	OUT	Word		0	请求的状态值显示循环的结果。这不是变频器输出的状态字（请参见 错误消息 (页 95)）。
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。 <ul style="list-style-type: none"> • 0 – 正向 • 1 – 反向

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
INHIBIT	OUT	Bool		FALSE	变频器已禁止：此位表示变频器的禁止位状态。 <ul style="list-style-type: none"> • 0 – 未禁止 • 1 – 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	变频器的状态字 1 此值包括变频器的固定状态位。
STATUS3	OUT	Word		0	变频器的状态字 3 此值包括变频器的用户可定义状态字。
STATUS4	OUT	Word		0	变频器的状态字 4 此值包括变频器的用户可定义状态字。
STATUS5	OUT	Word		0	变频器的状态字 5 此值包括变频器的用户可定义状态字。
STATUS6	OUT	Word		0	变频器的状态字 6 此值包括变频器的用户可定义状态字。
STATUS7	OUT	Word		0	变频器的状态字 7 此值包括变频器的用户可定义状态字。
STATUS8	OUT	Word		0	变频器的状态字 8 此值包括变频器的用户可定义状态字。

最初执行 USS_Drive_Control 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 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 切换到 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 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。

3.3 USS

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 3-36 SPEED_SP 与 DIR 参数交互

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

3.3.5 USS_Read_Param: 从变频器读取数据

说明

USS_Read_Param 指令从变频器读取参数。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 指令的背景数据块。必须从主程序的循环 OB 调用 USS_Read_Param。

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 指定哪个变频器参数已写入。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于访问大于 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 指定要写入哪个变频器参数索引。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)参数 USS_DB, 这是在向程序中添加 USS_Drive_Control 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 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 时此位复位。

3.3 USS

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
ERROR	OUT	Bool		FALSE	发生错误：如果 ERROR 为 TRUE，则表示发生错误且 STATUS 适用。出错时所有其它输出都置零。只在 USS_Port_Scan 指令的 ERROR 和 STATUS 输出发出通信错误信号。
STATUS	OUT	Word		0	STATUS 指定读取请求的结果。可在“USS_Extended_Error”变量中找到有关某些状态码的更多信息（请参见 错误消息 (页 95)）。
VALUE	OUT	Variant (Word, Int, UInt, DWord, DInt, UInt, Real)	Any (Word, Int, UInt, DWord, DInt, UInt, 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 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 或 USS_Write_Param 会导致错误 0x818A。

3.3.6 USS_Write_Param: 在变频器中更改数据

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 指令更改变频器中的参数。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 的背景数据块。

必须从主程序循环的 OB 调用 USS_Write_Param。

参数

表格 3-37 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 指定哪个变频器参数已写入。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 指定要写入哪个变频器参数索引。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。

3.3 USS

参数	声明	数据类型		标准	说明
		S7-1200/1500	S7-300/400/WinAC		
EEPROM	IN	Bool		-	保存在变频器的 EEPROM 中：如果为 TRUE，则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，则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, UInt, Real)	Any (Word, Int, UInt, DWord, DInt, UInt, 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）参数 USS_DB，这是在向程序中添加 USS_Drive_Control 指令时生成并初始化的。
DONE ¹	OUT	Bool		FALSE	如果为 TRUE，DONE 表示输入 VALUE 已写入变频器。USS_Drive_Control d 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 时此位复位。
ERROR	OUT	Bool		FALSE	如果 ERROR 为 TRUE，则表示发生错误且 STATUS 适用。出错时所有其它输出都置零。只在 USS_Port_Scan 指令的 ERROR 和 STATUS 输出发出通信错误信号。
STATUS	OUT	Word		0	STATUS 指定写入请求的结果。可在“USS_Extended_Error”变量中找到有关某些状态码的更多信息（请参见 错误消息 (页 95)）。

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 或 USS_Write_Param 函数会导致错误 0x818A。

3.3.7 关于变频器设置的常规信息

变频器设置的常规要求

- 需要对变频器的 4 个 PIV 字 (ParameterIDValue) 的用途进行组态。
- 变频器可组态 2 个、4 个、6 个或 8 个 PZD 字 (process data area)。
- 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。
- 所有变频器的数据传输速率都必须对应于 USS_Port_Scan 指令的 bps 输入。
- 必须为远程控制设置变频器。
- 必须为变频器的 COM 连接处的频率设定值指定 USS。
- 必须指定 1 到 16 个变频器地址。地址必须对应于变频器的 USS_Drive_Control 块的 DRIVE 输入。
- 必须设置为使用变频器设定值的极性，以便对变频器进行方向控制。
- 必须已正确终止 RS485 网络。

连接 MicroMaster 变频器

关于 SIEMENS MicroMaster 变频器的信息可用作示例。其它变频器的设置指令包含在变频器手册中。

如果要连接 4 系列的 MicroMaster 变频器 (MM4)，请将 RS485 电缆的末端插入两个无螺纹压力钳中，以进行 USS 操作。可使用常规 PROFIBUS 电缆和插头连接器将 S7-1200/1500 连接到 MicroMaster 变频器。

注意

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。屏蔽必须接地或连接到 9 针插头连接器的引脚 1。MicroMaster 变频器的接线终端 2-0V 应接地。

将 RS485 电缆另一端的两根电线插入 MM4 变频器的端子块中。要在 MM4 变频器中创建电缆连接，拆下变频器盖以便使用端子块。MM4 变频器系统手册中包含关于如何在指定的变频器上拆下盖的详细信息。

终端块上连接采用连续的编号方式。在 S7-1200/1500 端，使用 PROFIBUS 连接插头将电缆的端子 A 连接到变频器的端子 15（对于 MM420 变频器）或变频器的端子 30（对于

3.3 USS

MM440 变频器)。将连接器的端子 B (P) A (N) 连接到端子 14 (对于 MM420 变频器) 或端子 29 (对于 MM440 变频器)。

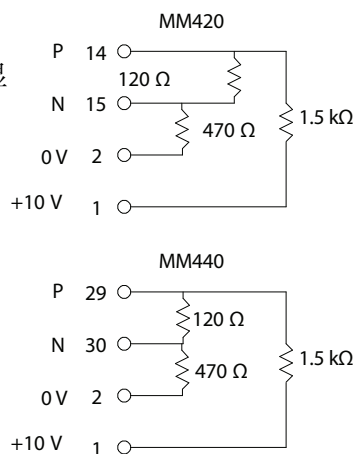
如果 S7-1200/1500 是网络中的终端站或采用点对点连接, 则必须使用插头连接器的端子 A1 和 B1 (不是 A2 和 B2), 因为这些端子提供了终止设置 (例如, DP 插头连接器 6ES7972-0BA40-0X40)。

注意

安装变频器盖

连接设备的电源之前, 仔细确认再次安装了变频器盖。

如果变频器已组态为网络中的终端站, 则终止电阻必须与对应的端子相连。此图显示了用于终止 MM4 变频器的连接的示例。



设置 MM4 变频器

将变频器连接到 S7-1200/1500 前，确保变频器具有下列系统参数。使用变频器键盘来设置参数：

1.	将变频器复位为出厂设置（可选）。	P0010 = 30 P0970 = 1
	如果跳过步骤 1，确保这些参数已设置为指定值：	USS PZD 长度 = P2012 索引 0 = (2、4、6 或 8) USS PKW 长度 = P2013 索引 0 = 4
2.	激活所有参数的读取/写入访问（专家模式）。	P0003 = 3
3.	检查变频器的电机设置。设置取决于相应的电机。 将参数 P010 设置为 1 后，才可设置参数 P304, P305, P307, P310 或 P311（快速调试模式）。设置完参数后，将参数 P010 设置为 0。参数 P304, P305, P307, P310 和 P311 只能在快速调试模式中更改。	P0304 = 电机的额定电压 (V) P0305 = 电机的额定电流 (A) P0307 = 电机的额定功率 (W) P0310 = 电机的额定频率 (Hz) P0311 = 电机的额定转速
4.	设置本地/远程模式。	P0700 索引 0 = 5
5.	设置到 USS 的 COM 连接的频率设定值。	P1000 索引 0 = 5
6.	加速时间（可选） 电机加速到最大频率需要的时间（以秒为单位）。	P1120 = (0 到 650.00)
7.	斜坡时间（可选） 电机减速到完全停止需要的时间（以秒为单位）。	P1121 = (0 到 650.00)
8.	设置串行连接的参考频率：	P2000 = (1 到 650 Hz)
9.	设置 USS 正常化：	P2009 索引 0 = 0

3.3 USS

10.	设置 RS-485 串行接口的 bps:	P2010 索引 0 = 4 (2400 Bit/s) 5 (4800 Bit/s) 6 (9600 Bit/s) 7 (19200 Bit/s) 8 (38400 Bit/s) 9 (57600 Bit/s) 12 (115200 Bit/s)
11.	输入从站地址。 每个变频器（最多 31 个）都可通过总线进行操作。	P2011 索引 0 = (0 到 31)
12.	指定串行连接的超时值。 这是两个进入数据帧之间允许的最大间隔。此功能将关闭逆变器以避免通信故障。接收到有效帧后即开始时间测量。如果在特定时间内未收到其它数据帧，逆变器将关闭并显示错误代码 F0070。如果将值设置为零，则控制会关闭。	P2014 索引 0 = (0 到 65535 ms) 0 = 超时，取消激活
13.	将数据从 RAM 传输到 EEPROM:	P0971 = 1 (开始传输)。将更改保存到 EEPROM 中的参数设置。

错误消息

错误消息概述 - PtP

错误消息在指令的 STATUS 输出处输出，并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	补救措施
0x0000	无错误	-
接收状态和错误代码		
0x0094	基于“接收固定/最大帧长度”(Receipt of fixed/maximum frame length) 识别的帧结束	-
0x0095	基于“消息超时”(Message timeou) 识别的帧结束	-
0x0096	基于“字符延迟时间”(Character delay time) 的结束而识别的帧结束	-
0x0097	因为达到最大响应时间而中止了帧。	-
0x0098	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	-
0x0099	基于“结束序列”(End sequence) 的接收而识别的帧结束	-

错误代码	说明	补救措施
发送状态和错误代码		
0x7000	块空闲	-
0x7001	新帧的初始调用：数据传输已启动	-
0x7002	中间调用：数据传输运行	-
0x8085	无效长度	选择合适的帧长度。 以下内容有效（取决于模块）： 1-1024/2048/4096 (Byte)
0x8088	指定长度超过了接收缓冲区中设置的范围。 注意：如果将 BUFFER 参数指定为 STRING 数据类型，那么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。 以下内容有效（取决于模块）： 1-1024/2048/4096 (Byte)
接收状态和错误代码		
0x8088	接收的字符数超过 BUFFER 参数指定的数量。	选择合适的帧长度。 以下内容有效（取决于模块）： 1-1024/2048/4096 (Byte)
特殊功能的错误消息代码		
0x818F	错误的参数编号设置（仅限 USS）	选择合适的参数编号 (PARAM)。 下列编号有效：0-2047
0x8190	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。
0x8191	诊断错误中断的错误设置	为“诊断错误中断” (Diagnostic error interrupt) 选择合适的值。 以下内容有效：诊断错误中断禁用或诊断错误中断激活。 检查所寻址的模块是否支持生成诊断中断。

错误代码	说明	补救措施
“端口组态”的错误消息代码		
0x81A0	该模块不支持此协议。	为模块 (PROTOCOL) 选择一个有效的协议。
0x81A1	该模块不支持此数据传输速率。	为模块 (BAUD) 选择有效的数据传输速率。
0x81A2	该模块不支持此奇偶校验设置。	为“奇偶校验” (Parity) (PARITY) 选择合适的值。 以下内容有效： <ul style="list-style-type: none"> • 无 (1) • 偶校验 (2) • 奇校验 (3) • 标记校验 (4) • 间隔校验 (5) • 任意 (6)
0x81A3	该模块不支持此数据位数。	为“数据位数” (Number of data bits) (DATABITS) 选择合适的值。 以下内容有效： <ul style="list-style-type: none"> • 7 (2) • 8 (1)
0x81A4	该模块不支持此停止位数。	为“停止位数” (Number of stop bits) (STOPBITS) 选择合适的值。 以下内容有效： <ul style="list-style-type: none"> • 1 (1) • 2 (2)
0x81A5	该模块不支持此数据流控制类型。	为模块 (FLOWCTRL) 选择有效的数据流控制。
0x81A7	无效的 XON 或 XOFF 值	为 XON (XONCHAR) 和 XOFF (XOFFCHAR) 选择合适的值。 数值的有效范围： 0...255

错误代码	说明	补救措施
0x81AA	无效的工作模式	<p>有效的工作模式包括：</p> <ul style="list-style-type: none"> • 全双工 (RS232) (0) • 全双工 (RS422) 四线制模式（点对点）(1)/(CM PtP (ET 200SP)) • 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) • 半双工 (RS485) 双线制模式 (4)
0x81AB	无效接收线路初始状态	<p>有效的初始状态是：</p> <ul style="list-style-type: none"> • “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）。
0x81AC	“断路检测” (Break detection) 的值无效	<p>为“断路检测” (Break detection) 选择合适的值。以下内容有效：</p> <ul style="list-style-type: none"> • 断路检测已禁用 (0) • 断路检测已激活 (1)。
0x81AF	该模块不支持此协议。	为该模块选择一个有效的协议。

错误代码	说明	补救措施
“发送组态”的错误代码		
0x81B5	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： <ul style="list-style-type: none"> • 取消激活 (0)， • 1 个 (1) 或 2 个 (2) 结束符 或 <ul style="list-style-type: none"> • 取消激活 (0)， • 结束序列的 1 个 (1) 至最多 5 个 (5) 字符。
0x81B6	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		
0x81C0	开始条件无效	选择合适的启动条件。 以下内容有效： <ul style="list-style-type: none"> • 在帧开始前发送中断 • 发送“空闲线路”(Idle Line)。
0x81C1	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 Auto-Hotspot）。
0x81C3	“最大消息长度”(Maximum message length) 的值无效	为“最大信息长度”(Maximum message length) (MAXLEN) 选择合适的值。 值的有效范围（取决于模块）： 1-1024/2048/4096 (Byte)
0x81C4	“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）： 1-1024/2048/4096 (Byte)

错误代码	说明	补救措施
0x81C5	“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) (LENGTHSIZE) 选择合适的值。 值的有效范围（以字节表示）： <ul style="list-style-type: none"> • 1 (1) • 2 (2) • 4 (4)
0x81C6	“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) (LENGTHM) 选择合适的值。 数值的有效范围： 0 到 255（字节）
0x81C7	“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： <ul style="list-style-type: none"> • 消息中的偏移量（取决于模块）： 0-1024/2048/4096 (Byte) • “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）
0x81C8	“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： 1-65535 (ms)
0x81C9	“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： 1-65535（位时间）
0x81CB	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。
0x81CC	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	补救措施
0x81CD	“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： <ul style="list-style-type: none"> • 取消激活防止覆盖操作 (0) 或 • 激活防止覆盖操作 (1)
0x81CE	“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： <ul style="list-style-type: none"> • 取消激活启动时的清除接收缓冲区 (0) • 激活启动时的清除接收缓冲区 (1)
发送状态和错误代码		
0x81D0	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。
0x81D1	超出 XON 或 CTS = ON 的等待时间。	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。
0x81D2	“硬件 RTS 始终 ON”：由于 DSR = ON 变为 OFF，已中止发送命令	检查通信伙伴。确保 DSR 在整个传输持续期间始终为 ON。
0x81D3	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）： 1-1024/2048/4096（字节）
0x81D5	传输因参数更改、检测到线路断路或 CPU 处于 STOP 状态而中止	检查参数分配、线路断路和 CPU 状态。
0x81D6	传输因未接收到结束标识符而取消	检查结束字符的参数分配和通信伙伴的帧。
0x81D7	用户程序和模块间的通信错误	检查通信（例如，匹配序列号）。
0x81D8	尝试传输因未组态模块而被拒绝	组态模块。

错误代码	说明	补救措施
接收组态的错误代码		
0x81E0	帧已中止： 发送缓冲区上溢/发送帧太长	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。
0x81E1	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。
0x81E2	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。
0x81E3	帧已中止： 字符上溢错误	固件出错： 请联系客户支持。
0x81E4	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”(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。
0x81E5	帧已中止： 中断	连接到伙伴的接收线路中断。 重新连接或接通伙伴电源。
0x81E6	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。
0x81E8	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。
0x81E9	Modbus CRC 错误（仅适用于支持 Modbus 的通信模块）	Modbus 帧的校验和错误。检查通信伙伴。
0x81EA	Modbus 帧过短（仅适用于支持 Modbus 的通信模块）	不符合 Modbus 帧的最短长度。检查通信伙伴。
0x81EB	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 (Byte) 检查帧检测结束参数。

错误代码	说明	补救措施
错误代码 V24 伴随信号		
0x81F0	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 RS232 模块或已设置 RS232 模式 (ET 200SP)。
0x81F1	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。
接收组态的错误代码		
0x8201 ¹⁾	BUFFER 是指向无效数据类型的指针	输入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, STRING
0x8225	BUFFER 指向大于 1 KB 的优化存储区 或 BUFFER 指向优化存储区并且接收长度大于 BUFFER 访问的区域。	输入一个指针，其指向区域的最大长度需满足： <ul style="list-style-type: none"> • 优化存储区： 1 kByte • 非优化区域： 4 kByte 注意： 如果指针指向优化存储区，发送的数据不要超过 1 kByte。
0x8229 ¹⁾	BUFFER 是指向 BOOL 的指针，其位数不等于 $n * 8$	如果正使用指向 BOOL 的指针，位数必须是 8 的倍数。
错误代码，一般		
0x8280	读取模块时进行否定确认	检查 PORT 参数上的输入 在 RDREC.STATUS 静态参数和 SFB RDREC 的说明中会找到有关错误原因的更多详细信息。
0x8281	写入模块时进行否定确认	检查 PORT 参数上的输入 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。
0x8282	模块不可用	检查 PORT 参数上的输入并确保模块可以访问。

错误代码	说明	补救措施
接收组态的错误代码		
0x82C1	“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： 1-255
0x82C2	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，请确保未发送接收组态。
0x8301 1)	BUFFER 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, STRING
0x8322	读取参数时发生范围长度错误	检查 BUFFER 参数上的输入
0x8324	读取参数时发生范围错误	检查 BUFFER 参数上的输入
0x8328	读取参数时发生设置错误	检查 BUFFER 参数上的输入
发送组态的错误代码		
0x8328 1)	BUFFER 是指向 BOOL 的指针，其位数不等于 $n * 8$	如果正使用指向 BOOL 的指针，位数必须是 8 的倍数。
接收组态的错误代码		
0x8332	Receive_Conditions 参数有无效数据块	检查 Receive_Conditions 参数上的输入
0x833A	BUFFER 参数上的数据块标志表示的是未下载的数据块。	检查 BUFFER 参数上的输入
0x8351	数据类型无效	检查 BUFFER 参数上的输入
0x8352 1)	Receive_Conditions 未指向数据块	检查指向 Receive_Conditions 的指针
0x8353 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	说明	补救措施
错误代码 3964(R) 协议		
0x8380	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)
0x8381	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)
0x8382	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： <ul style="list-style-type: none"> • 高 (1) • 低 (0)
0x8383	参数分配错误：“块检查”(Block check) 值无效	为“块检查”(Block check) (BCC) 选择合适的值。 以下内容有效： <ul style="list-style-type: none"> • 带块检查 (1) • 不带块检查 (0)
0x8384	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： 1-255
0x8385	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： 1-255
0x8386	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。
0x8387	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。
0x8388	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。

错误代码	说明	补救措施
0x8389	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。
0x838A	运行错误： 接收时发生逻辑错误。 收到 DLE 后接收到另一个字符（不是 DLE, ETX）。	检查帧头和数据字符中的伙伴 DLE 是否总是重复两次或者是否通过 DLE ETX 断开连接。 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。
0x838B	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。
0x838C	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。
0x838D	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。
0x838E	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。
0x838F	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
0x8391	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。

1) 仅限 S7-300/400 CPU 的指令

错误消息概述 - Modbus

错误代码	说明	补救措施
0x0000	无错误	-
接口组态错误 - Modbus_Comm_Load		
0x8181	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。
0x8182	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： <ul style="list-style-type: none"> • 无 (1) • 偶校验 (2) • 奇校验 (3) • 标记校验 (4) • 间隔校验 (5) • 任意 (6)
0x8183	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。
0x8184	“响应超时” (Response timeout) 值无效。	在 RESP_TO 参数上为“响应超时”(Response timeout) 选择有效的值。 数值的有效范围： 1-65535 (ms)
0x8280	读取模块时进行否定确认	检查 PORT. 参数上的输入 在静态参数 Send_Config.RDREC.STATUS、Receive_Config.RDREC.STATUS 或 RDREC.STATUS，以及 SFB RDREC 的说明中会找到有关错误原因的更多详细信息。
0x8281	写入模块时进行否定确认	检查 PORT. 参数上的输入 在静态参数 Send_Config.WRREC.STATUS、Receive_Config.WRREC.STATUS 或 WRREC.STATUS，以及 SFB WRREC 的说明中会找到有关错误原因的更多详细信息。
0x8282	模块不可用	检查 PORT 参数上的输入并确保模块可以访问。

错误代码	说明	补救措施
组态错误 - Modbus_Slave		
0x8186	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：标准地址范围 1-247； 扩展地址区域 1-65535 (0 预留给广播)
0x8187	参数 MB_HOLD_REG 的值无效	在 MB_HOLD_REG 参数上为保持寄存器选择合适的值。
0x8188	无效的操作模式或广播 (MB_ADDR = 0) 并且参数 MODE ≠ 1	在广播操作模式下为 MODE 选择值 1，或选择不同的操作模式。
0x818C	指向 MB_HOLD_REG 的指针必须为数据块或标记区。	为指向 MB_HOLD_REG 区的指针选择有效的值。
0x8280	读取模块时进行否定确认	检查 PORT. 参数上的输入 在静态参数 Send_P2P.RDREC.STATUS 或 Receive_P2P.RDREC.STATUS，以及 SFB RDREC 的说明中会找到有关错误原因的更多详细信息。
0x8281	写入模块时进行否定确认	检查 PORT. 参数上的输入 在静态参数 Send_P2P.WRREC.STATUS 或 Receive_P2P.WRREC.STATUS，以及 SFB WRREC 的说明中会找到有关错误原因的更多详细信息。
0x8452 ¹⁾	MB_HOLD_REG 不是指向数据块或标记区的指针	检查指针 MB_HOLD_REG
0x8453 ¹⁾	MB_HOLD_REG 不是 BOOL 类型或 WORD 类型的指针	检查指针 MB_HOLD_REG
0x8454 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。	检查指针 MB_HOLD_REG
0x8455 ¹⁾	MB_HOLD_REG 指向具有写保护的数据块	检查指针 MB_HOLD_REG
0x8456 ¹⁾	指令执行期间出错。错误原因显示在静态参数 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	补救措施
组态错误 - Modbus_Master		
0x8180	无效 MB_DB 参数值	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。
0x8186	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: 标准地址范围 1-247; 扩展地址区域 1-65535 (0 预留给广播)
0x8188	无效的操作模式或广播 (MB_ADDR = 0) 并且参数 MODE ≠ 1	在广播操作模式下为 MODE 选择值 1, 或选择不同的操作模式。
0x8189	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 52) 说明
0x818A	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 52) 说明
0x818B	无效的 DATA_PTR 值	在 DATA_PTR 参数 (标记或数据块地址) 上为数据指针选择合适的值。 请参见信息系统中的 Modbus_Master (页 52) 说明
0x818C	DATA_PTR 参数的互连错误	检查指令的互连。
0x818D	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。	检查指针 DATA_PTR
0x8280	读取模块时进行否定确认	检查 PORT. 参数上的输入 在静态参数 Send_P2P.RDREC.STATUS 或 Receive_P2P.RDREC.STATUS, 以及 SFB RDREC 的说明中会找到有关错误原因的更多详细信息。

错误代码	说明	补救措施
0x8281	写入模块时进行否定确认	检查 PORT. 参数上的输入 在静态参数 Send_P2P.WRREC.STATUS、Receive_P2P.WRREC.STATUS 或 Receive_Reset, 以及 SFB WRREC 的说明中会找到有关错误原因的更多详细信息。
通信错误 - Modbus_Master 和 Modbus_Slave		
0x80 D1	超出 XON 或 CTS = ON 的等待时间。	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。
0x80D2	“硬件 RTS 始终 ON”: 由于 DSR = ON 变为 OFF, 已中止发送命令	检查通信伙伴。确保 DSR 在整个传输持续期间始终为 ON。
0x80E0	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。
0x80E1	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。
0x80E2	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。
0x80E3	帧已中止: 字符上溢错误	检查通信伙伴的帧中的数据个数。
0x80E4	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		
0x80C8	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。
0x8200	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。

错误代码	说明	补救措施
协议错误 - Modbus_Slave (仅适用于支持 Modbus 的通信模块)		
0x8380	CRC 错误	Modbus 帧的校验和错误。检查通信伙伴。
0x8381	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。
0x8382	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。
0x8383	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。
0x8384	请求帧中的数据值无效	检查 Modbus 主站的请求帧中的数据值
0x8385	Modbus 从站不支持诊断值 (功能代码 08)	Modbus 从站仅支持诊断值 0x0000 和 0x000A。
协议错误 - Modbus_Master (仅适用于支持 Modbus 的通信模块)		
0x8380	CRC 错误	Modbus 帧的校验和错误。检查通信伙伴。
0x8381	来自 Modbus Slave 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。
0x8382	来自 Modbus Slave 的响应帧有下列错误消息: 无效长度	选择合适的数据长度。
0x8383	来自 Modbus Slave 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。
0x8384	来自 Modbus Slave 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。
0x8385	来自 Modbus Slave 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 0x0000 和 0x000A。
0x8386	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。
0x8387	从站未发出请求的应答	检查从站的响应帧。检查从站的地址设置。
0x8388	从站对写入请求的响应出现错误。	检查从站的响应帧。
0x8828 ¹⁾	DATA_PTR 指向的位地址不等于 $n * 8$	检查指针 DATA_PTR
0x8852 ¹⁾	DATA_PTR 不是指向数据块或标记区的指针	检查指针 DATA_PTR
0x8853 ¹⁾	DATA_PTR 不是 BOOL 类型或 WORD 类型的指针	检查指针 DATA_PTR
0x8855 ¹⁾	DATA_PTR 指向具有写保护的数据块	检查指针 DATA_PTR
0x8856 ¹⁾	调用 SFC51 时出错	再次调用 Modbus_Master 指令

错误代码	说明	补救措施
错误 - Modbus_Slave (仅适用于支持 Modbus 的通信模块)		
0x8428 ¹⁾	MB_HOLD_REG 指向的位地址不等于 $n * 8$	检查指针 MB_HOLD_REG
0x8452 ¹⁾	MB_HOLD_REG 不是指向数据块或标记区的指针	检查指针 MB_HOLD_REG
0x8453 ¹⁾	MB_HOLD_REG 不是 BOOL 类型或 WORD 类型的指针	检查指针 MB_HOLD_REG
0x8454 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。	检查指针 MB_HOLD_REG
0x8455 ¹⁾	MB_HOLD_REG 指向具有写保护的 DB	检查指针 MB_HOLD_REG
0x8456 ¹⁾	调用 SFC51 时出错	再次调用 Modbus_Slave 指令

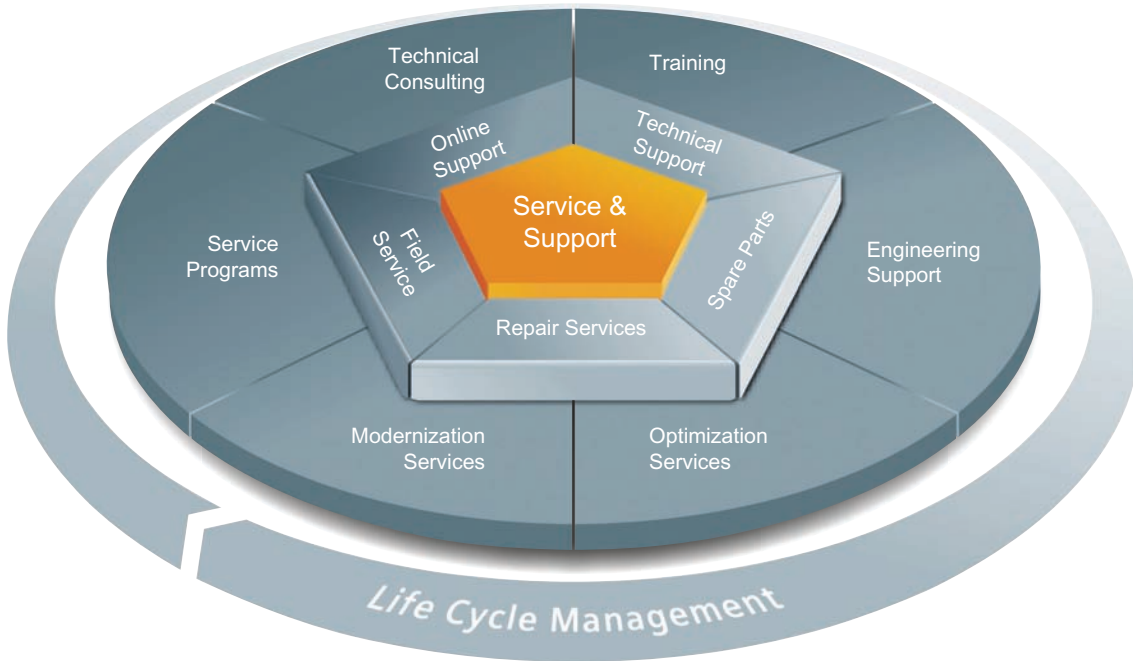
1) 仅限 S7-300/400CPU 的指令

错误消息概述 - USS

错误代码	说明	补救措施
0x0000	无错误	-
0x8180	变频器响应的长度错误	检查变频器的响应帧。
0x8181	数据类型错误	选择合适的数据类型。 以下内容有效： <ul style="list-style-type: none"> • 实数 • 字 • 双字
0x8182	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。
0x8183	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。
0x8184	变频器响应的校验和错误	检查变频器和通信连接。
0x8185	寻址错误	有效的变频器地址范围： 1 到 16
0x8186	设定值错误	有效的设定值范围： -200 % 到 +200 %
0x8187	返回的变频器编号错误	检查变频器的响应帧。
0x8188	无效 PZD 长度	允许的 PZD 长度： 2、4、6 个字
0x8189	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。
0x818A	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。
0x818B	变频器未响应。	检查变频器。
0x818C	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。
0x818D	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。
0x818E	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	补救措施
0x8280	读取模块时进行否定确认	检查 PORT. 参数上的输入 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 的说明中会找到有关错误原因的更多详细信息。
0x8281	写入模块时进行否定确认	检查 PORT. 参数上的输入 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 的说明中会找到有关错误原因的更多详细信息。

1) 仅限 S7-300/400 CPU 的指令



整个生命周期内的全面非凡服务

对于设备制造商、解决方案供应商以及工厂操作员而言：西门子工业自动化与驱动技术集团将为制造和加工行业内所有领域中的各种不同用户提供全面服务。

为了配合我们的产品和系统，我们提供有集成的结构化服务，以便在您设备或工厂生命周期的每个阶段都提供有高质量的服务和支持：从规划和实施到调试，以及维护和现代化改造，一应俱全。

我们的服务和支持时刻伴在您的左右，为您解决所有的西门子自动化和驱动技术问题。我们在 100 多个国家为设备和工厂生命周期的所有阶段都提供有现场支持。

在您的身边，将有一支由经验丰富的专家所组成的团队，为您提供积极的支持和专业技术。即使您与我们横跨多个大陆，我们的员工也将定期为您开展各种培训课程并与您保持密切的联系，以确保在各种领域为您提供可靠的服务。

在线支持

全面的在线信息平台，可以随时随地为您提供全面的服务与支持。

您可以在以下 Internet (<http://www.siemens.com/automation/service&support>) 地址上找到在线支持。

技术咨询

全面地为您的项目进行规划和设计：我们的规划和设计内涵盖了实际状态的详细分析、目标定义、产品和系统问题咨询，以及自动化解决方案的创建，无所不及。

技术支持

除了为客户提供有关技术问题的专家建议，我们还提供大量针对我们产品和系统的按需服务。

您可以在以下 Internet (<http://www.siemens.com/automation/support-request>) 地址上找到技术支持。

培训

我们为您提供的各种实践专业知识，助您在激烈的竞争中处于不败之地。

您可以在以下 Internet (<http://www.siemens.com/sitrain>) 地址上找到培训课程。

工程组态支持

在项目工程组态和开发阶段，我们将专门针对您的要求进行量身定制的服务支持，涵盖了从自动化项目组态到实施的所有阶段。

现场服务

我们的现场服务为您提供调试和维护服务，以确保您的设备和工厂始终处于运行状态。

备件

在全世界的每个行业中，持久的可靠性是工厂和系统在运作时的必要条件。我们通过遍布全球的网络和最优秀的物流链，从一开始就为您提供所需的支持，使工厂和系统运行通畅。

维修

停机会在工厂中导致各种问题的产生并由此引发不必要的成本。我们通过遍布全球的维修设施，可以帮助您将这两者的成本降至最低。

优化

在设备和工厂的服务寿命期间，通常有很大的空间来提高生产力或降低成本。为了帮助您实现这一终极目标，我们提供了全面的优化服务。

现代化改造

在需要现代化改造时，您也将得到我们的支持，我们将提供有从规划阶段直到调试完成的全面服务。

服务计划

我们的服务计划是针对自动化和驱动系统或产品组特选的各种服务包。各个服务之间相互协调以确保全面覆盖整个生命周期并对产品和服务的使用进行优化。

服务计划中的服务可以随时灵活更改并单独使用。

服务计划示例：

- 服务合同
- 工厂 IT 安全服务
- 驱动工程生命周期服务
- SIMATIC PCS 7 生命周期服务
- SINUMERIK 机床增效及制造信息化
- SIMATIC 远程支持服务

优势一览：

- 减少停机时间，提高生产力
- 量身定制各种服务，降低了维护成本
- 可预先计算并规划的成本
- 响应时间和备件交付时间有保障，服务十分可靠
- 客户服务人员将为额外任务提供支持以及解决方案
- 一站式全面服务，更少的联络，更多的专业技术

联系方式

在全球范围内就近为您提供各种服务：针对工业自动化和驱动技术集团提供的所有产品，我们都为您提供咨询、销售、培训、服务、支持、备件等服务。

有关人员联系方式，请访问 Internet (<http://www.siemens.com/automation/partner>) 上的联系方式数据库。

词汇表

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

清除以发送。通信伙伴可随时接收数据。

RTS

请求发送。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。

点对点通讯

在点对点通信中，通信处理器构成了可编程逻辑控制器与通讯伙伴之间的接口。

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

通信模块可用于点对点连接和总线链接。

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 and 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

B

BUFFER 参数, Send_P2P, 37

G

Get_Features, 10

L

LENGTH 参数, Send_P2P, 37

M

Modbus

Modbus_Comm_Load, 48

Modbus_Slave, 52, 61

Modbus_Comm_Load, 11, 48

Modbus_Master, 11

Modbus_Slave, 11, 52, 61

P

P3964_Config, 10

P3964_Config (协议组态), 33

Port_Config, 10

Port_Config (端口组态), 20

PtP 指令返回值, 17

PtP 通信

编程, 13

PtP 错误类别, 19

R

Receive_Config, 10

Receive_Config (接收组态), 26

Receive_P2P, 10

Receive_P2P (接收点对点数据), 38

Receive_Reset, 10

Receive_Reset (复位接受器), 40

S

Send_Config, 10

Send_Config (发送组态), 24

Send_P2P, 10

Send_P2P (发送点对点数据), 35
LENGTH 和 BUFFER 参数, 37

Set_Features, 10

Signal_Get, 10

Signal_Get (获取 RS232 信号), 41

Signal_Set, 10

Signal_Set (设置 RS232 信号), 42

U

USS 协议库

USS_Drive_Control, 83

USS_Port_Scan, 80

USS_Read_Param, 87

USS_Write_Param, 89

关于变频器设置的常规信息, 91

使用要求, 76

概述, 74

USS_Drive_Control, 11, 75, 83

USS_Port_Scan, 11, 75, 80

USS_Read_Param, 11, 75, 87

USS_Write_Param, 11, 75, 89

六划

全局库

USS 协议概述, 74

共享 PtP 参数错误, 19

七划

返回值

PtP 指令, 17

返回值接收运行时间, 38

八划

参数组态

Send_P2P 的 LENGH 和 BUFFER, 37

九划

帧组态

指令, 14

指令

P3964_Config (协议组态), 33

Port_Config (端口组态), 20

Receive_Config (接收组态), 26

Receive_P2P (接收点对点数据), 38

Receive_Reset (复位接受器), 40

Send_Config (发送组态), 24

Send_P2P (发送点对点数据), 35

Signal_Get (获取 RS232 信号), 41

Signal_Set (设置 RS232 信号), 42

USS_Drive_Control, 83

USS_Port_Scan, 80

USS_Read_Param, 87

USS_Write_Param, 89

查询架构, 16

查询架构从站, 16

查询架构主站, 16

点对点编程, 13

十划

通信

查询架构, 16

通信接口

编程, 13

通信模块 (CM)

编程, 13

数据接收, 38

十一划

接口组态

指令, 14

十二划

编程

PtP 指令, 13

十三划

数据传输, 触发, 35