

SIEMENS

SIMATIC

S7-300/S7-400

用于点对点 CP 的可加载驱动程序：
MODBUS 协议，RTU 格式，S7 为
从站

操作说明

前言	1
产品说明	2
安装	3
工作模式	4
调试驱动程序	5
调试通讯 FB	6
CPU – CP 接口	7
传输协议	8
功能代码	9
驱动程序的诊断	10
通讯 FB 的诊断	11
技术数据	A
多点接线图	B
参考资料	C

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
带有警告三角，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。

小心
不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。

注意
表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有权利的目的由第三方使用而特别标示的。

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。

目录

1	前言	7
2	产品说明	11
2.1	应用程序选件	11
2.2	硬件和软件需求	13
2.3	GOULD-MODBUS 协议概述	14
3	安装	17
3.1	使用软件狗	17
3.2	接口连接	18
4	工作模式	19
4.1	SIMATIC / MODBUS 从站数据链接的组件	19
4.2	任务分配	21
4.3	使用的 MODBUS 功能代码	21
4.4	SIMATIC CPU 中的数据区	22
4.5	使用面向位的功能代码进行访问	23
4.6	使用面向寄存器的功能代码进行访问	25
4.7	启用/禁用写访问	27
5	调试驱动程序	29
5.1	在 STEP 7 编程设备/PC 中安装驱动程序	29
5.2	卸载驱动程序	30
5.3	组态数据链接	31
5.3.1	使用 CP 341 组态数据链接	31
5.3.2	使用 CP 441-2 组态数据链接	32
5.4	为 CP 分配参数	33
5.4.1	为 CP 分配参数	33
5.4.2	为 CP 341 分配参数	33
5.4.3	为 CP 441-2 分配参数	34
5.5	数据链接的组态	36

5.6	为可加载驱动分配参数	37
5.6.1	MODBUS 从站协议	38
5.6.2	位功能的 MODBUS 地址转换	41
5.6.3	转换寄存器功能的 MODBUS 地址	46
5.6.4	写功能的限制	49
5.6.5	RS422/485 (X27) 接口	51
5.7	装载 CP 341 的组态和参数分配数据	53
5.8	装载驱动程序到 CP 341	54
5.9	装载 CP 441-2 的组态和参数分配数据	55
5.10	CP 的启动特性	56
5.11	“CPU 启动”的参数分配	57
6	调试通讯 FB	59
6.1	安装 FB	59
6.2	STEP 7 项目	60
6.3	FB 80 参数 (CP 341)	62
6.4	FB 180 参数 (CP 441-2)	63
6.5	程序调用	64
6.6	周期性操作	67
7	CPU – CP 接口	69
7.1	CP 341 的 CPU-CP 接口	69
7.1.1	常规信息	69
7.1.2	数据传送长度 CPU <-> CP	70
7.1.3	数据一致性	70
7.2	用于 CP 441-2 的 CPU-CP 接口	71
7.2.1	常规信息	71
7.2.2	数据传送长度 CPU <-> CP	72
7.2.3	数据一致性	73
8	传输协议	75

9	功能代码	83
9.1	功能代码 01 — 读取线圈（输出）状态	85
9.2	功能代码 02 — 读输入状态	90
9.3	功能代码 03 — 读输出寄存器	94
9.4	功能代码 04 — 读输入寄存器	99
9.5	功能代码 05 — 写单个线圈	104
9.6	功能代码 06 — 预设单个寄存器	108
9.7	功能代码 08 — 回送诊断测试	111
9.8	功能代码 15 — 写多个线圈	113
9.9	功能代码 16 — 预设多个寄存器	118
10	驱动程序的诊断	123
10.1	CP 341 上的诊断工具	124
10.1.1	通过 CP 341 的显示元件进行诊断	124
10.1.2	CP 341 功能块的诊断消息	125
10.2	CP 441-2 上的诊断工具	126
10.2.1	通过 CP 441-2 的显示元件进行诊断	126
10.2.2	CP 441-2 的系统功能块的诊断消息	128
10.2.3	读取 CP 441-2 的错误消息区域 SYSTAT	130
10.2.4	通过 CP 441-2 的错误消息区 SYSTAT 进行诊断	131
10.3	错误/事件表	133
10.3.1	SYSTAT 中的“CPU 作业错误”错误代码	134
10.3.2	SYSTAT 中的“接收错误”错误代码	134
10.3.3	SYSTAT 中常规处理错误的错误代码	136
11	通讯 FB 的诊断	141
11.1	通过参数 ERROR_NR 和 ERROR_INFO 进行诊断	142
11.1.1	初始化时出错	143
11.1.2	在处理功能代码的过程中出错	144
11.1.3	其他错误	146

A 技术数据..... 147

 A.1 技术数据..... 147

B 多点接线图 155

C 参考资料..... 157

 C.1 参考资料..... 157

 词汇表 159

 索引..... 167

前言

本手册的用途

手册的信息可帮助您建立“Modbus 功能”的从站 CP 和 MODBUS 主站控制系统之间的数据链接，并调试该数据链接。

所需的基本知识

使用本手册需要具备自动化工程的基本知识。

此外，还应该了解如何使用装有 Microsoft® Windows® 操作系统的计算机或具有相似功能的设备（例如，编程设备），并且应该对 STEP 7 编程有一些了解。

本手册内容

本手册适用于以下软件：

产品	订货号	起始版本
用于点对点 CP 的可加载驱动程序	6ES7870-1AB01-0YA0	3.0

说明

手册包含了驱动程序和功能块的描述，自出版发行起生效。

指南

手册描述了可加载驱动程序的功能及其与 CP 341 和 CP 441-2 通信处理器的软件和硬件的集成。

本手册包括以下主题：

- 产品描述/安装
- 数据链接的工作模式
- 调试驱动程序/安装/参数分配
- 调试功能块/参数分配/STEP 7 应用程序实例
- CPU-CP 接口
- 传输协议/功能代码
- 驱动程序的诊断
- 功能块的诊断

约定

本手册中使用通用术语 CP，或 CP 341 和 CP 441-2。

特别说明

本手册中描述的驱动程序可用作 CP 的可加载协议，用它来替换 3964R、RK512、ASCII 和打印机标准协议。

说明

使用此驱动程序可以对 CP 和 CPU 之间的通信顺序进行修改或扩展。

特别是可以对诊断的现有事件类别和事件编号进行修改和扩展。

另外注意，本手册仅描述了相对于标准功能的修改和扩充。可以在所使用的 CP 的手册中找到所有基本信息。

为了确保该驱动程序的安全使用，需要具备关于 CP 运行的详细知识。

技术支持

可以使用技术支持请求的 Web 表单获得所有工业自动化产品的技术支持，Web 表单的网址为：

<http://www.siemens.com/automation/support-request>。

有关技术支持的更多信息，请访问 Internet，网址为：

<http://www.siemens.com/automation/service>。

因特网上的“服务与支持”

除文档外，我们还提供了在线技术知识，网址为：

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

在此可以找到：

- 不断为您提供产品最新信息的新闻快递。
- 所需文档，可以使用产品支持搜索功能找到。
- 论坛，世界各地的用户和专家在此交流想法。
- 当地的工业自动化合作伙伴。
- 有关维修、备件和咨询的信息。

更多支持

如果您对使用本手册中介绍的产品还有疑问，且在手册中未找到正确的答案，请联系当地西门子代表。

可在以下网站找到有关联系人的信息：

<http://www.siemens.com/automation/partner>

我们为各 SIMATIC 产品和系统提供的技术文档的向导位于：

<http://www.siemens.com/simatic-tech-doku-portal>

在线目录和订购系统位于：

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

培训中心

我们提供了一系列课程，来帮助您熟悉 SIMATIC S7 自动化系统。请联系当地的培训中心或位于德国纽伦堡的培训中心总部，以获取详细资料，网址为：

<http://www.sitrain.com>

产品说明

2.1 应用程序选件

在系统环境中的位置

此处描述的驱动程序是用于通信处理器 CP 341 (S7-300) 和 CP 441-2 (S7-400) 的软件产品。

CP 341 和 CP 441-2 可用于 S7 自动化系统，用于建立与伙伴系统之间的串行通信链接。

驱动程序的功能

驱动程序和适当的功能块可以帮助您在 MODBUS 主站控制系统（例如：Modicon 控制器或 Honeywell TDC 3000）和具有 Modbus 功能的从站系统 CP 341 或 CP 441-2 通信模块之间建立通信链接。

使用的传输协议为 RTU 格式的 GOULD - MODBUS 协议。数据传输符合主-从原则。

传输过程中主站具有主动性，CP/S7-CPU 作为从站运行。

功能代码 01, 02, 03, 04, 05, 06, 08, 15 和 16 可以用于 CP 和主机系统之间的通信。

MODBUS 地址的解释

从主站发出的请求消息帧中 MODBUS 地址由驱动程序以“S7-链接”的方式解释。

这意味着可以实现以下操作：

- 读写存储器位、输出、数据块，
- 读取 S7-CPU 中的存储器位、输入、定时器和计数器。

以下章节将阐明对 MODBUS 地址的解释。

可用的接口和协议

CP 441-2 的两个串行接口可以相互独立地使用不同的标准协议或者可加载驱动程序协议进行操作。

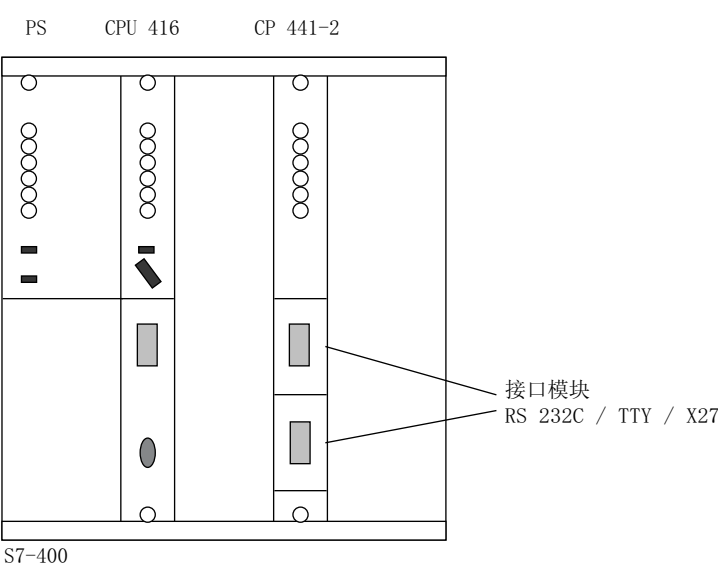
CP 可用的物理接口类型包括 RS 232、TTY 或者 RS 422/485 (X27)。

当使用两线操作的 RS422/485 (X27) 接口时，一台主站最多可以连接 32 个从站，这样便建立了多点连接（网络）。

使用其它类型的物理接口创建多点连接需要附加的外部硬件（例如：Honeywell DHP）。

允许的系统组态

下图显示了一种允许的系统组态的原理示意图。



数据一致性

S7 CPU 和 CP 之间的数据交换通过集成的系统功能逐块进行。

更多相关信息，请参见本手册中的“数据一致性 (页 70)”和“数据一致性 (页 73)”部分。

2.2 硬件和软件需求

可用的模块

驱动程序在部件编号为 MLFB 6ES7441-2AA02-0AE0 或更高的 CP 341 和 CP 441-2 上运行。

部件编号为 MLFB 6ES7441-1AA0x-0AE0 的 CP 441-1 和部件编号为 MLFB 6ES7441-2AA00-0AE0 或 6ES7441-2AA01-0AE0 的 CP 441-2 不能与可加载驱动程序一起使用。

软件狗

要使用具有可加载驱动程序的 CP，需要有软件狗。软件狗和驱动程序一起提供。

CPU 的装载存储器（存储卡）

使用 CP441-2 时，可加载驱动程序基于其参数分配装载到 CPU 的装载存储器，并在 CPU 启动时传送到 CP 存储器。

因此，CPU 必须有足够的装载存储空间。因此，需要 RAM 或 FLASH 存储卡（部件编号 MLFB 6ES7952-...）。

分配了可加载驱动程序的每个 CP 接口大约需要 25 KB 的 CPU 装载存储器。

当使用 CP 341 时，将可加载驱动程序直接装载到 CP 341。因此 S7-300 CPU 上的装载存储器不是必需的。但是注意，在没有编程设备的情况下将不能更换模块。

软件要求

已安装 STEP 7 Basis V5.3 或更高版本。

已安装为点对点连接分配参数的可选软件包 CP PtP Param V5.1 或更高版本。

数据结构

在组态 S7 数据接口之前，应该确保该数据结构与 MODBUS 主站系统的用户程序相兼容。明确要使用哪些功能代码和哪些 MODBUS 地址。

2.3 GOULD-MODBUS 协议概述

功能代码

MODBUS 系统之间的数据交换类型是由功能代码 (FC) 控制的。

数据交换

下列的 FC 可用于执行 **逐位** 的数据交换：

- FC 01 读线圈（输出）状态，
- FC 02 读输入状态，
- FC 05 写单个线圈，
- FC 15 写多个线圈。

下列的 FC 可用于执行 **逐个寄存器** 的数据交换：

- FC 03 读保持寄存器，
- FC 04 读输入寄存器，
- FC 06 预置单个寄存器，
- FC 16 预置多个寄存器。

数据区域

作为规则，各 FC 的操作将依照下表进行：

功能代码	数据	数据类型		访问类型
01, 05, 15	线圈（输出）状态	位	输出	读/写
02	输入状态	位	输入	只读
03, 06, 16	保持寄存器	寄存器 (16 位)	输出寄存器	读/写
04	输入寄存器	寄存器 (16 位)	输入寄存器	只读

地址表示

和划分成读/写区和只读区相似，用户层级的数据可用下表显示的方法表示：

功能代码	数据类型	用户层级的地址表示（十进制）
01, 05, 15	输出位	0xxxx
02	输入位	1xxxx
04	输入寄存器	3xxxx
03, 06, 16	保持寄存器	4xxxx

在串行传输线上的**传输消息帧**中，MODBUS 用户系统使用的地址以 **0** 为基准。

在 **MODBUS 用户系统**里，这些地址从 **1** 开始计算！

示例

- 用户系统的第一个保持寄存器表示为寄存器 **40001**。在传输消息帧中，当使用 FC 03、06 或 16 时，十六进制值 0000 作为寄存器地址传送。
- 用户系统中的第 127 个线圈表示为线圈 **00127**，并且在传输消息帧中为该线圈分配的地址为十六进制的 **007E**。

2.3 GOULD-MODBUS 协议概述

安装

3.1 使用软件狗

简介

要使用具有可加载驱动程序的 CP，需要有软件狗。当插入软件狗时，可加载驱动程序。对于 CP 441-2，可以为两个接口下载驱动程序。

插入软件狗

要插入软件狗，需要将 CP 从机架上移除。插入软件狗的模块插槽位于 CP 的背面，在背板总线可插拔连接器的上方。

3.2 接口连接

RS 232C / TTY

可以与主站系统创建点对点连接。
关于接口连接的进一步信息，参见手册“CP：点对点通信”。

X27/RS485（2 线）

可以与主站系统创建多点连接（网络），一个主站最多可连接 32 个从站。
CP 的驱动程序将会使接收的 2 线制线路在发送和接收之间切换。

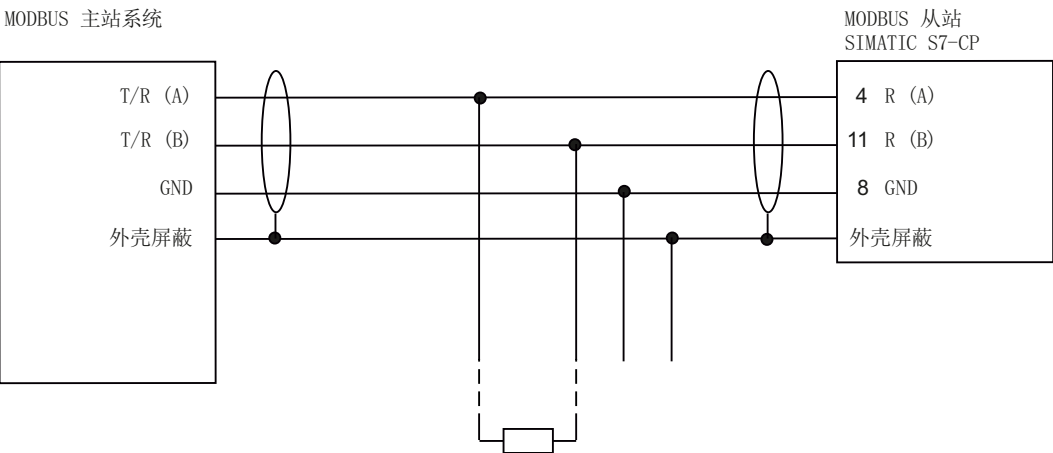


图 3-1 连接示意图：1 个主站系统，1 个从站系统在总线上

关于接口连接的进一步信息，参见手册“CP：点对点通信”。

X27/RS485（4 线）

可以与主站系统创建点对点连接。
不可以直接创建多点连接（网络），因为 RS422/RS485 (X27) 接口硬件不支持这种操作。
有关接口连接的更多信息，请参见“多点接线图 (页 155)”部分以及手册《CP：点对点通信》。

工作模式

常规信息

所提供的数据库链接将 MODBUS 协议的数据访问转换到 SIMATIC S7 CPU 上的特定存储区。

4.1 SIMATIC / MODBUS 从站数据链接的组件

MODBUS 从站数据链接

CP 的 MODBUS 从站数据链接包含两个部分：

- CP 的可加载驱动程序
- SIMATIC S7 CPU 的 MODBUS 通信功能块

MODBUS 从站通信 FB

除了可加载的 MODBUS 从站驱动程序，SIMATIC MODBUS 从站数据链接还需要 S7 CPU 中的特定通信 FB。

安装 MODBUS 从站 CD 后，可以在 **STEP 7 库 Modbus** 中获得所需的通信 FB。该库包括下列各块：

- MODBUS 通信功能块 FB 180，适合 S7-400 CPU 的应用
- MODBUS 通信功能块 FB 80，适合 S7-300 CPU 的应用

因为 FB 80 调用功能块 FB 7 和 FB 8，所以必须在控制器中装载 FB 7 和 FB 8。

在 **STEP 7 工程文件 示例/Modbus/** 中的示例 OB 中显示了 FB 的调用。CP 441-2 的调用储存在 Station Modbus Slave 400 中，CP 341 的调用储存在 Station Modbus Slave 300 中。

4.1 SIMATIC / MODBUS 从站数据链接的组件

Modbus 通信 FB 处理数据链接所需的所有功能。

提供的 MODBUS 从站通信功能块 FB180/80 必须在用户程序的循环程序中调用。

MODBUS 通信 FB 使用背景数据块作为工作区域。

说明

如果对所提供的功能块做任何修改，担保都将失效。所导致的损坏将无法得到赔偿。

MODBUS 从站驱动程序

可加载驱动程序实现了 MODBUS 协议，并在 SIMATIC 存储区中转换 MODBUS 地址。

使用参数分配工具 *CP*，将可加载驱动程序装载到 *SIMATIC*：点对点连接的参数分配，这样驱动程序便自动传输到 *CP* 中。

参数

以下列出的参数和工作模式必须使用参数分配工具为可加载驱动程序设置。

- 传输率、奇偶校验
- CP 的从站地址
- 工作模式（正常、干扰抑制）
- 字符延迟时间的倍增因子
- FC 01、05、15 的地址区域。
- FC 02 的地址区域
- FC 03、06、16 的基本 DB 号
- FC 04 的基本 DB 号
- 只写访问的限制

4.2 任务分配

CP 341 的任务分配

使用 CP 341 时，必须为所有功能代码调用通信 FB FB 80。

CP 441 的任务分配

CP 441 独立处理功能代码 01、02、03、04、06 和 16。

对于功能代码 05 和 15，通信 FB FB 180 将数据逐位输入 SIMATIC 存储区。

4.3 使用的 MODBUS 功能代码

使用的功能代码

驱动程序支持下列 MODBUS 功能代码：

功能代码	符合 MODBUS 规范的功能	常规描述
01	读线圈状态	读位
02	读输入状态	读位
03	读保持寄存器	读寄存器（字）
04	读输入寄存器	读寄存器（字）
05	写单个线圈	写 1 位
06	预设单个寄存器	写 1 个寄存器（字）
08	环路测试	线路检查
15	写多个线圈	写多个位
16	预设多个寄存器	写多个寄存器

4.4 SIMATIC CPU 中的数据区

数据区

各个 FC 访问 PLC 中的下列 SIMATIC 数据区域：

功能代码	MODBUS 数据类型	SIMATIC 数据类型	访问类型
01	线圈状态	存储器位	逐位读取
		输出	逐位读取
		定时器	逐位读取 (16 位间隔)
		计数器	逐位读取 (16 位间隔)
02	输入状态	存储器位	逐位读取
		输入	逐位读取
03	保持寄存器	数据块	逐字读取
04	输入寄存器	数据块	逐字读取
05	单个线圈	存储器位	逐位写入
		输出	逐位写入
06	单个（保持）寄存器	数据块	逐字写入
08	-	-	镜像接收
15	多个线圈	存储器位	逐位写入
		输出	逐位写入
16	多个（保持）寄存器	数据块	逐字写入

地址转换

消息帧中的 MODBUS 地址由驱动程序以“S7 方式”进行解释，并在 SIMATIC 存储区中进行转换。

对各个 SIMATIC 存储区的访问可以由用户使用**参数分配工具 CP：点对点连接的参数分配**进行指定。

4.5 使用面向位的功能代码进行访问

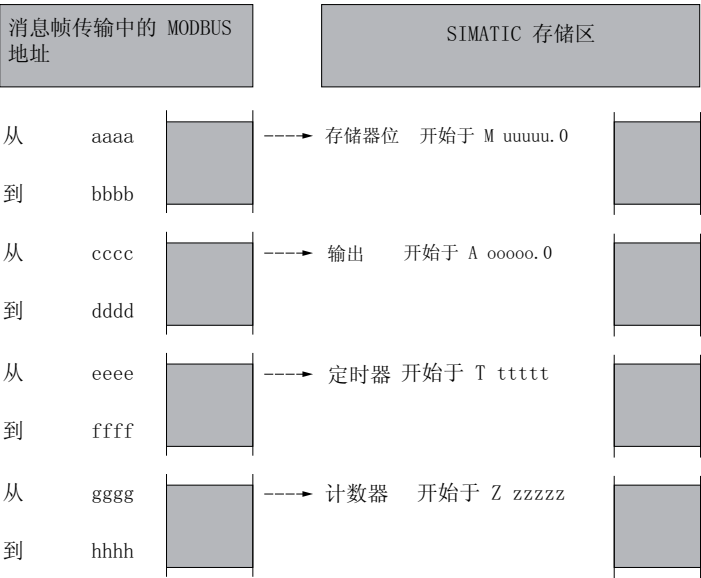
功能代码 01、05 和 15

面向位的功能代码 **01**、**05** 和 **15** 允许对**存储位**、**输出**、**定时器**和**计数器**的 SIMATIC 存储区进行读写访问；

对于 **FC 01**，定时器和计数器为只读。

可以使用参数分配工具指定对存储位、输出、定时器和计数器进行读写访问的 MODBUS 地址。

也可以参数化 SIMATIC 存储区中的数据元素，访问应该从该存储区开始。



4.5 使用面向位的功能代码进行访问

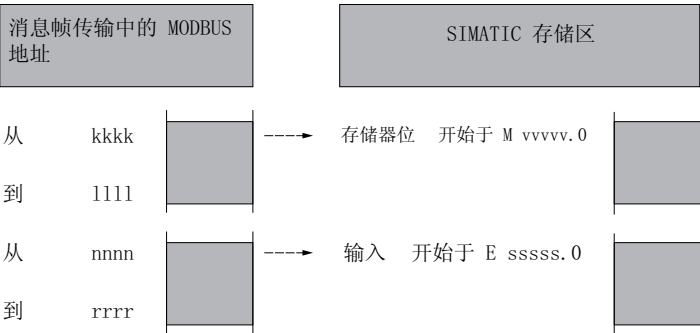
功能代码 02

面向位的功能代码 **02** 可以对 **SIMATIC 存储区存储器位和输入**进行只读访问。

可以使用参数分配工具指定对存储器位和输入进行读写访问的 **MODBUS 地址**。

也可以参数化 **SIMATIC 存储区**中的数据元素，访问应该从该存储区开始。

FC 02 的 MODBUS 地址区和 SIMATIC 存储区的选择独立于 FC 01、05 和 15 的地址区和存储区。

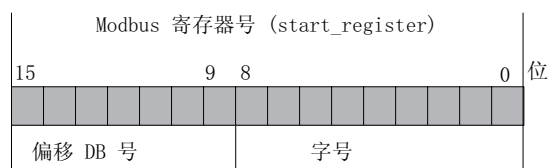


4.6 使用面向寄存器的功能代码进行访问

功能代码 03、06 和 16

面向寄存器的功能代码 **03**、**06** 和 **16** 允许对 **SIMATIC 数据块** 存储区进行读写访问。
所需的数据块编号 **分两步** 计算。

1. 参数化接口可用于指定 **基本 DB 号**。这个基本 DB 为可以访问的 **第一个 DB**。
2. 消息帧中传递的 MODBUS 地址 **Start_Register**（寄存器号）按以下方式解释：



生成的 DB 号

那时访问的生成 DB 号，它由以下公式计算得出：
基本 DB 号 + 偏移 DB 号。

DB 号可以用于在整个可寻址的数据块区域中访问包含 **128** 个连续数据块的数据块区，整个可寻址数据区域包含 **65,535** 个 DB。

DB 中的字编号

使用字编号可以在各数据块中对 **DBW 0** 到 **DBW 1022** 的区域进行寻址。

DB 的基本结构由字节组成，这样可由驱动程序逐字解释，方法如下：

字编号	访问	
0	DBW 0	(= DBB 0/1)
1	DBW 2	(= DBB 2/3)
2	DBW 4	(= DBB 4/5)
3	DBW 6	(= DBB 6/7)
:	:	:
511	DBW 1022	(= DBB 1022/1023)

4.6 使用面向寄存器的功能代码进行访问

功能代码 04

面向寄存器的功能代码 **04** 可以对 SIMATIC 存储区**数据块**进行只读访问。

这种访问的工作方式和功能代码 03、06 和 16 相同。

功能代码 **04** 具有自己的基本 DB 号，可以使用参数分配工具进行自由参数分配。 这样可以选第二个包含 128 个 BD 的**独立** 区域。

然而，对这些 DB 只能进行只读访问；不能对其写入。

4.7 启用/禁用写访问

功能代码 05、06、15 和 16

对于**写**功能代码 **05、06、15 和 16**，可以 **禁用**或限制访问相关 **SIMATIC** 存储区。

可以使用参数分配工具在 **MODBUS** 主站系统上指定**启用写** 访问的区域。

如果主站试图访问该启用区域之外的 **SIMATIC** 存储区，则访问会被拒绝，同时输出一条错误消息帧（例外）。

启用写访问



4.7 启用/禁用写访问

调试驱动程序

常规信息

STEP 7 V5.3 中提供了下述与 STEP 7 相关的任务。

序列、名称和目录信息可能不同于后续的版本。

5.1 在 STEP 7 编程设备/PC 中安装驱动程序

简介

要安装由驱动代码和驱动程序特定屏蔽文件组成的驱动程序，请按下列步骤执行：

1. CD-ROM 驱动器中插入 MODBUS 从站 CD。
2. 在 Windows 操作系统下，双击“控制面板”中的“添加/删除程序”图标，打开安装软件的对话框。
3. 在对话框中，选择 CD-ROM 驱动器然后选择 **setup.exe** 文件，启动安装过程。
4. 按照安装程序显示的说明逐步操作。

结果：驱动程序和参数分配屏幕窗体将安装到以下目录：**Step7\S7ftp\S7Driver**。

该目录包含了除此之外的下列文件：

- S7wfpb1a.dll
- S7wfpb1x.cod
- S7wfpb2x.cod

5.2 卸载驱动程序

简介

可以在 Windows 环境下从 STEP 7 程序包中卸载驱动程序，在“控制面板”，“添加/删除程序”中选择“卸载”。

随后，您可以检查 Step7\S7ftp\S7Driver 目录下名称为 S7wfpb1??.*、S7wfpb2??.*、S7wfpb3??.* 的文件是否全部被删除。

说明

在卸载软件包“参数分配工具 CP：为点对点连接分配参数” (Parameter Assignment Tool CP: Assigning Parameters to Point-To-Point Connections) 之前，需要先卸载所有的可加载驱动程序。

5.3 组态数据链接

简介

数据链接的组态需要在组态表中包含使用 HW config 进行硬件放置。可以使用 STEP 7 软件进行组态。

5.3.1 使用 CP 341 组态数据链接

S7 项目

在进行组态之前，必须使用 STEP 7 创建一个 S7 项目。

项目组件

使用 SIMATIC 管理器将所需的项目组件插入到打开的项目中：

- SIMATIC 300 站。

在每次插入组件之前，必须单击选择所需的项目。

插入 > 站 > SIMATIC 300 站 (Insert > Station > SIMATIC 300 station)

硬件配置 (Hardware Configuration)

硬件配置包含定义硬件组件本身及其属性。

要开始配置硬件，请选择 SIMATIC 300 站并双击“硬件”或选择菜单命令“编辑 > 打开对象”。

使用菜单命令“插入 > 硬件组件”(Insert > Hardware Components) 插入下列组件：

- 从 SIMATIC 300 中插入一个 RACK-300、一个 PS-300 和一个 CPU-300
- 从 CP-300 中插入具有适当订货号的 CP PtP。

STEP 7 的用户手册中详细介绍了如何组态 S7-300 模块。

5.3.2 使用 CP 441-2 组态数据链接

简介

对于点对点数据链接，必须先配置 SIMATIC 400 站、连接伙伴站、PtP 节点和 PtP 网络。

S7 项目

在进行组态之前，必须使用 STEP 7 创建一个 S7 项目。

项目组件

使用 SIMATIC 管理器将所需的项目组件插入到打开的项目中：

- SIMATIC 站，其它站，PtP 网络。

在每次插入组件之前，必须单击选择所需的项目。

- 使用 **插入 > 站 > SIMATIC 400 站 (Insert > Station > SIMATIC 400 Station)**
创建自己的 S7 程序（机架、电源、CPU、CP 441-2, ...），
- 使用 **插入 > 站 > 其它站 (Insert > Station > Other Station)**
创建数据连接伙伴，
- 使用 **插入 > 子网 > PtP (Insert > Subnet > PtP)**
创建一个 SIMATIC 400 站和数据连接伙伴之间的 PtP 网络。

硬件配置 (Hardware Configuration)

硬件配置包含定义硬件组件本身及其属性。

要开始配置硬件，请选择 SIMATIC 400 站并双击“硬件”或选择菜单命令“编辑 > 打开对象”。

使用菜单命令“插入 > 硬件组件”(Insert > Hardware Components) 插入下列组件：

- 从 SIMATIC 400 中插入一个 RACK-400、一个 PS-400 和一个 CPU-400
- 从 CP-400 中插入具有适当订货号的 CP PtP。

关于如何组态 S7 400 模块的详细描述，请参见 STEP 7 的用户手册。

5.4 为 CP 分配参数

5.4.1 为 CP 分配参数

为 CP 分配参数

使用”硬件配置“在机架上放置好模块后，必须为其分配参数。

要启动参数分配工具，在“硬件配置”(Hardware Configuration) 中双击 CP，或者单击 CP 并选择菜单命令 **编辑 > 对象属性 (Edit > Object Properties)**。

5.4.2 为 CP 341 分配参数

步骤如下：

1. 属性 — CP > 基本参数 (Properties - CP > Basic Parameters)

单击“**参数**”(Parameters) 按钮（单击）打开协议选择接口“**为点对点连接分配参数**”(Assigning Parameters to Point-To-Point Connections)。这里可以选择所需的传输协议。

选择协议之后，可以进行**驱动的参数分配**（通过双击字母符号开始）。

关于如何选择协议以及如何在对话框中为可加载驱动程序分配参数，详情请参见“驱动的参数分配 (页 37)”一节。

参数分配完成后，返回到“**属性 - CP**”对话框。

2. 属性 — CP > 地址 (Properties - CP > Addresses)

在“属性 - CP”(Properties - CP) 对话框中的“**地址**”(Addresses) 选项卡中**无需**任何设置。

3. 属性 — CP > 常规 (Properties - CP > General)

在“属性 - CP”(Properties - CP) 对话框中的“**常规**”(General) 选项卡中**无需**任何设置。

点击“属性 - CP”对话框中的“**确认**”，完成 CP 的参数分配。然后返回到“硬件配置”对话框。

保存参数设置并关闭“硬件配置”对话框。返回到 STEP 7 项目的主菜单。

5.4.3 为 CP 441-2 分配参数

步骤如下：

1. 属性 — CP 441-2 > 基本参数 (Properties - CP 441-2 > Basic Parameters)

在“基本参数”(Basic Parameters) 选项卡中指定 CP 441 模块所需的“接口”(interface) (1=上面的接口, 2=下面的接口)。选择已插入的接口子模块作为“模块”(Module)。

单击“参数”(Parameters) 按钮（单击）打开协议选择接口“为点对点连接分配参数”(Assigning Parameters to Point-To-Point Connections)。

这里可以选择所需的传输协议。

选择协议之后，可以进行驱动程序的参数分配。双击字母符号启动。

关于如何为可加载驱动程序，选择协议和为对话框分配参数的详细描述可从“为可加载驱动程序分配参数 (页 37)”部分中找到。

参数分配完成后，返回到“属性 - CP441-2”对话框。

2. 属性 - CP 441-2 > 地址

在“属性 - CP 441-2”(Properties - CP 441-2) 对话框中的“地址”(Addresses) 选项卡中无需任何设置。

3. 属性 - CP 441-2 > 常规

在“属性 - CP 441-2”(Properties - CP 441-2) 对话框中的“常规”(General) 选项卡中，可以指定 CP 的接口连接哪个 PtP 网络 (PtP network)。

PtP(1) 对应 CP 上面的接口，PtP(2) 对应 CP 下面的接口。

单击 PtP(1) 或 PtP(2) 按钮打开用于组态子网的对话框。

选择所需的子网 (subnet) 并且激活复选框“伙伴已连接到所选的网络”(Partner is connected to the selected network)。

所选的子网表示 CP 接口和连接伙伴接口之间的连接。

点击“确认”返回到“属性 - CP 441-2”对话框。在此点击“确认”完成 CP 的参数分配后，返回到“硬件配置”对话框。

保存参数设置并关闭“硬件配置”对话框。返回到 STEP 7 项目的主菜单。

为连接伙伴分配参数

在 **STEP 7** 项目中插入连接伙伴站之后，按照在“项目组件：插入 > 其它站”下的描述，需要为该伙伴站指定对象属性。

从打开的 **STEP 7** 项目开始，单击选择连接伙伴站（其它站）。

选择菜单命令 **编辑 > 对象属性** (Edit > Object Properties)。

打开“属性 - 其它站”对话框。

1. 属性 - 其它站 > 节点列表

在“**节点列表**” (Node List) 选项卡中选择“新建”(New) 按钮。

在“选择类型”项，选择“**PTP 节点**”，然后单击“确认”。

显示“**网络连接**” (Network Connection) 对话框。

选择所需的 **子网 (subnet)**，该子网表示 CP 接口和连接伙伴接口之间的连接，同时激活复选框“节点已经连接到所选网络”(Node is connected to selected network)。

点击“确认”返回到“节点列表”选项卡。

2. 属性 - 其它站 > 常规

在“**常规**” (General) 选项卡中无须进行任何设置。

单击“**确定**”返回到 **STEP 7** 项目的主菜单。

伙伴站也可以有几个接口（=PtP 节点），因此也可以连接到不同的点对点网络。

5.5 数据链接的组态

简介

本章节仅介绍 CP 441-2，如果您使用的是 CP 341，本章可以跳过。

通讯链接

CP 代表了通过点对点数据链接建立的 S7 CPU 和通讯伙伴/总线之间的数据链接。必须为每个要连接到连接伙伴/总线的串行接口组态数据链接。

数据链接的组态

在 STEP 7 项目中打开的 S7-400 站上选择 CPU，并且通过双击“**连接**”(Connections) 打开数据链接组态。

对话框“进行连接的项目组态”弹出。

选择菜单命令**插入 > 连接** (Insert > Connection) 打开“新建连接”(New Connection) 对话框。这里可以选择用于新数据链接的连接伙伴（其它站），并且选择“点对点连接”(Point-to-Point Connection) 作为连接类型。

单击“确定”确认。现在“连接属性”(Connection Properties) 对话框将打开。

连接属性

可以根据需要修改给定的 ID。

选择“连接方向 3：本地 <-> 伙伴”。

将显示参数化的路由。

两个 CPU 编号的指示与本驱动程序的操作无关。

点击“确定”接受所有设置。

保存“数据链接的项目组态”，并关闭对话框。

请注意 当在用户程序中调用 MODBUS 通讯 FB 时，必须再次使用连接 ID（Local ID）。

5.6 为可加载驱动分配参数

打开参数分配工具 CP-PtP

选择 SIMATIC 站然后双击“硬件”(Hardware) 或者选择“编辑 > 打开对象”(Edit > Open object) 启动“配置硬件”(Configure hardware)。

选择 CP，然后选择**编辑 > 对象属性** (Edit > Object Properties)。

选择接口（仅限 CP 441-2）和接口模块（仅限 CP 441-2），然后选择“**参数**”(Parameters) 按钮打开协议选择对话框。

协议选择

除了标准协议之外，选择对话框还将显示所有已安装的可加载驱动程序。为该驱动程序选择“**MODBUS 从站**”。

双击邮箱图标，以设置传输协议。将显示用于设置协议特定参数的对话框。

驱动程序特定参数

以下描述的用于本驱动程序的参数可以在各对话框中设置。

5.6.1 MODBUS 从站协议

传输参数概述

表格 5- 1 速率，字符帧

参数	说明	值范围	默认值
波特率	数据传输速度（位/秒）	300 600 1200 2400 4800 9600 19200 38400 76800	9600
具有其它波特率的 CP 341 的订货号如下： • 6ES7341-1xH01-0AE0 • 6ES7341-1xH02-0AE0		57600	
具有其它波特率的 CP 441-2 的订货号如下： • 6ES7441-2AA03-0AE0 • 6ES7441-2AA04-0AE0		57600 115200	
数据位	每个字符的位数	8	8
停止位	停止位个数	1 2	1
奇偶校验	没有传送奇偶校验位。	无	偶校验
	数据位数量将补充成奇数。	奇校验	
	数据位数量将补充成偶数。	偶校验	

传输速率

传输速率为数据传输的速度，单位：位/秒（波特）。

注意 CP 441-2 的最大总传输速率。总传输速率是两个接口的已参数化的传输速率的总和。

TTY 接口的最大传输速率为 19200 波特。

数据位

数据位数说明了要传输的一个字符映射的位数。

停止位

停止位数定义了要传输的两个字符之间的最小允许时间间隔。

奇偶校验

奇偶校验位用于数据安全。 根据参数分配，将要传输的数据位数补充成偶数或奇数。
如果奇偶校验已设置为“无校验”(none)，则不传输奇偶校验位。这会降低传输的完整性。

协议参数概述

表格 5-2 协议参数

参数	说明	值范围	默认值
从站地址	CP 自身的从站地址	1 到 255	222
激活 RS485 的操作模式选择	启用在“半双工”(RS485)两线制模式下的“常规操作”选择。	是 否	否
操作模式	“常规”操作模式 “干扰抑制”(Interference suppression)	常规操作 干扰抑制	- 在“全双工 (RS422) 四线模式”下：常规操作 - 在“半双工 (RS485) 两线模式”下：干扰抑制
倍增字符延时	用于传输速率的倍增因子—取决于字符延迟时间	1 到 10	1

5.6 为可加载驱动分配参数

从站地址

在此，可以对需要应答的 CP 指定单独的 MODBUS 从站地址。当接收到的从站地址和参数化的从站地址一致时，CP 才应答消息帧。不检查其它从站的消息帧，并且不接收响应。

激活 RS485 的操作模式选择

在激活“RS485 的操作模式选择”后，如果您已经选择了“半双工 (RS485) 两线制模式”，您还可以在“接口”选项卡下将操作模式切换为“常规操作”。

常规操作

这种操作模式下，在接收来自连接伙伴的消息帧前后所有检测到传输错误或“中断”将生成一条相应的错误消息。

帧的第一个字符必须是一个有效的从站地址。

仅当字符延迟时间结束时，才识别为消息帧结束。

干扰抑制

如果在开始接收消息帧时接收线检测到断路 (BREAK) 信号，或者 CP 接口块检测到传输错误，将不会在用户程序中生成错误消息。

如果传输错误或 BREAK 是在接收消息帧 (CRC 代码) 结束后出现的，则也会被忽略。

正确接收到的从站地址可以检测到来自连接伙伴的接收消息帧的起始位置。

倍增字符延时

如果一个连接伙伴无法满足 MODBUS 规范的时间需求，允许使用倍增字符延时 ZVZ，通过倍增因子 f_{MUL} 实现。只有当要求的时间无法满足时，才可以调节字符延时。

产生的字符延迟时间 t_{ZVZ} 计算方法如下：

- $t_{ZVZ} = t_{ZVZ_TAB} \times f_{MUL}$
- t_{ZVZ_TAB} : ZVZ 的查表值（参见“传输协议”章节）
- f_{MUL} : 倍增因子

5.6.2 位功能的 MODBUS 地址转换

FC 01、05、15 概述

表格 5-3 为 FC 01、05 和 15 进行 MODBUS 寻址转换

参数		输入	含义
SIMATIC 区存储器位			
消息帧传输的 MODBUS 地址 (位编号)	从	0 .. 65535 (十进制)	从此 MODBUS 地址开始
	到	0 .. 65535 (十进制)	包含此 MODBUS 地址
SIMATIC 存储区 存储器位 (存储器字节编号)	开始于	0 .. 65535 (十进制)	从此存储器字节开始
SIMATIC 区输出			
消息帧传输的 MODBUS 地址 (位编号)	从	0 .. 65535 (十进制)	从此 MODBUS 地址开始
	到	0 .. 65535 (十进制)	包含此 MODBUS 地址
SIMATIC 存储区 输出 (输出字节编号)	开始于	0 .. 65535 (十进制)	从此存储器字节开始
SIMATIC 区定时器			
消息帧传输的 MODBUS 地址 (位编号)	从	0 .. 65535 (十进制)	从此 MODBUS 地址开始
	到	0 .. 65535 (十进制)	包含此 MODBUS 地址
SIMATIC 存储区 定时器 (定时器编号)	开始于	0 .. 65535 (十进制)	从此定时器开始 (= 16 位字)

5.6 为可加载驱动分配参数

参数		输入	含义
SIMATIC 区计数器			
传输消息帧中的 <i>MODBUS 地址</i> (位编号)	从	0 .. 65535 (十进制)	从此 MODBUS 地址开始
	到	0 .. 65535 (十进制)	包含此 MODBUS 地址
SIMATIC 存储区计数器 (计数器编号)	开始于	0 .. 65535 (十进制)	从此计数器开始 (= 16 位字)

MODBUS 地址的“起”/“止”

可以使用“起始”地址为 MODBUS 地址分配参数，该 MODBUS 地址为某专用区域的开始位置；例如：存储器位、输出等 (= 区域的第一个位编号)。

可以使用“截止”地址为 MODBUS 地址分配参数，该 MODBUS 地址为某专用区域的末尾位置；例如：存储器位、输出等 (= 区域的最后一个位编号)。

“起”/“止”地址指的是功能代码 FC 01、05 和 15 的传输消息帧 (开始于 0 的位数) 中的 MODBUS 地址。

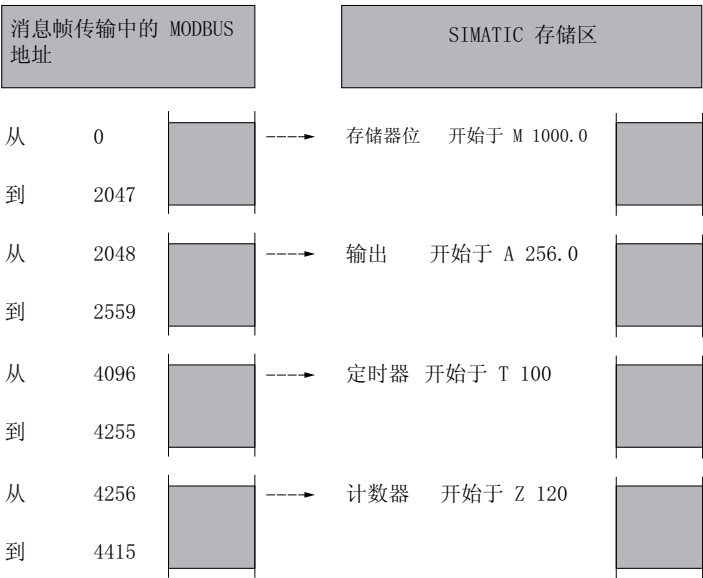
各个“起/止”区域不能重叠。

允许各个“起/止”区域之间有间隔。

“开始于” SIMATIC 存储区

可以使用“开始于”指定 SIMATIC 存储区的起始位置，用于映射 MODBUS 区的“起/止” (= SIMATIC 区的第一个存储器字节/输出字节/定时器/计数器的编号)。

示例



MODBUS 地址（0 到 2047）访问 SIMATIC 存储器位，从存储器位 M 1000.0 开始；例如区域长度 = 2048 个位 = 256 个字节，这意味着最后一个存储器位 = M 1255.7。

MODBUS 地址（2048 到 2559）访问 SIMATIC 输出，从输出 Q 256.0 开始；例如区域长度 = 512 个位 = 64 字节，这意味着最后一个输出位 = Q 319.7。

MODBUS 地址（4096 到 4255）访问 SIMATIC 定时器，从定时器 T 100 开始；例如区域长度 = 160 个位 = 10 字节，这意味着最后一个定时器 = T 109。

MODBUS 地址（4256 到 4415）访问 SIMATIC 计数器，从计数器 C 120 开始；例如区域长度 = 160 个位 = 10 字节，这意味着最后一个计数器 = C 129。

FC 02 概述

表格 5- 4 为 FC 02 进行寻址转换

参数		输入	含义
SIMATIC 区存储器位			
传输消息帧中的 MODBUS 地址 (位编号)	从	0 .. 65535 (十进制)	从此 MODBUS 地址开始
	到	0 .. 65535 (十进制)	包含此 MODBUS 地址
SIMATIC 存储区存储器位 (存储器字节编号)	开始于	0 .. 65535 (十进制)	从此存储器字节开始
SIMATIC 区输入			
消息帧传输的 MODBUS 地址 (位编号)	从	0 .. 65535 (十进制)	从此 MODBUS 地址开始
	到	0 .. 65535 (十进制)	包含此 MODBUS 地址
SIMATIC 存储区输入 (输入字节编号)	开始于	0 .. 65535 (十进制)	从此存储器字节开始

MODBUS 地址的“起”/“止”

可以使用“起始”地址为 MODBUS 地址分配参数，该 MODBUS 地址为某专用区域的开始位置；例如：存储器位、输出 (= 区域的第一个位编号)。

可以使用“截止”地址为 MODBUS 地址分配参数，该 MODBUS 地址为某专用区域的末尾位置 (= 区域的最后一个位编号)。

“起/止”地址指的是功能代码 FC 02 的传输消息帧中的 MODBUS 地址 (位编号从 0 开始)。

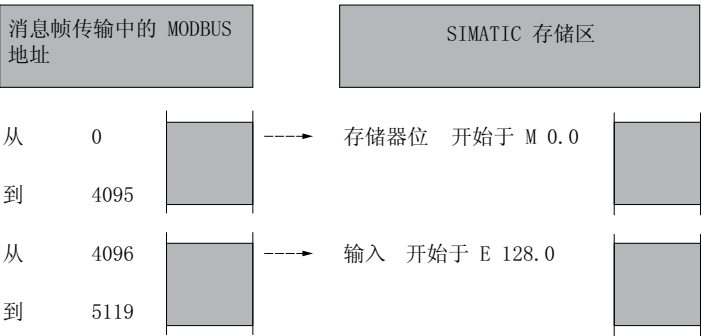
各个“起/止”区域不能重叠。

允许各个“起/止”区域之间有间隔。

“开始于” SIMATIC 存储区

可以使用“开始于”指定 SIMATIC 存储区的起始位置，用于映射 MODBUS 区的“起/止”（= SIMATIC 区的第一个存储器字节/输入字节的编号）。

示例



MODBUS 地址（0 到 4095）访问 SIMATIC 存储器位，从存储器位 M 0.0 开始；例如区域长度 = 4096 个位 = 512 个字节，这意味着最后一个存储器位 = M 511.7。

MODBUS 地址（4096 到 5119）访问 SIMATIC 输出，从输入 I 128.0 开始；例如区域长度 = 1024 个位 = 128 字节，这意味着最后一个输入位 = I 255.7。

说明

“开始存储器位”的输入值完全独立于功能代码 01、05 和 15 的“开始存储器位”的输入值。

这意味对于 FC 02 可以使用第二个 SIMATIC 存储器位区域（只读），完全独立于第一个存储器位区域。

对于 FC 01 和 FC 02 定义同步访问的存储器字节没有意义！

5.6.3 转换寄存器功能的 MODBUS 地址

FC 03、06 和 16 的概述

表格 5- 5 功能代码 FC 03、06 和 16 的 MODBUS 寻址转换

参数		输入	含义
SIMATIC 区 数据块			
传输消息帧中的 MODBUS 地址 = 0 (寄存器编号) 意味着访问:			
SIMATIC 存储区 数据块	开始于 DB	1 .. 65535 (十进制)	开始于此数据块 开始于 DBW 0 (= 基本 DB 号)

“开始于 DB”

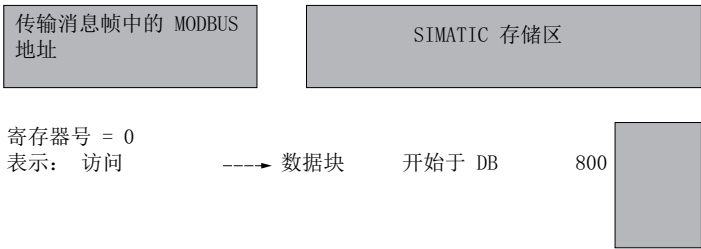
“开始于 DB”可以用于指定要访问的 SIMATIC 区域的第一个数据块 (=基本 DB 号)。

MODBUS 消息帧的寄存器编号的值是 0 时，从数据字 DBW 0 开始访问此 DB。

更高的 MODBUS 寄存器编号访问此后的数据字/数据块。

最多可以对 127 个连续的 DB 进行寻址。驱动程序解释 MODBUS 寄存器号的位 9 - 15，以访问各个连续 DB（请注意“功能代码 (页 83)”部分中的信息）。

示例



MODBUS 寄存器地址 0 可以用于访问 SIMATIC 系统中从 DBW 0 开始的数据块 800。

更高的 MODBUS 寄存器地址 (≥ 512 等) 访问此后的 DB，如 DB 801 等等。

FC 04 概述

表格 5-6 转换 FC 04 的 MODBUS 寻址

参数		输入	含义
SIMATIC 区 数据块			
传输消息帧 中的 MODBUS 地址 = 0（寄存器编号） 意味着访问：			
SIMATIC 存储区 数据块	开始于 DB	1 .. 65535 （十进制）	开始于此数据块 开始于 DBW 0 （= 基本 DB 号）

“开始于 DB”

“开始于 DB”可以用于指定要访问的 SIMATIC 区域的**第一个**数据块（=基本 DB 号）。

MODBUS 消息帧的寄存器编号的值是 0 时，从数据字 DBW 0 开始访问此 DB。

更高的 MODBUS 寄存器编号访问此后的数据字/数据块。

最多可以对 127 个连续的 DB 进行寻址。驱动程序解释 MODBUS 寄存器号的位 9 - 15，以访问各个连续 DB（请注意“功能代码 (页 83)”部分中的信息）。

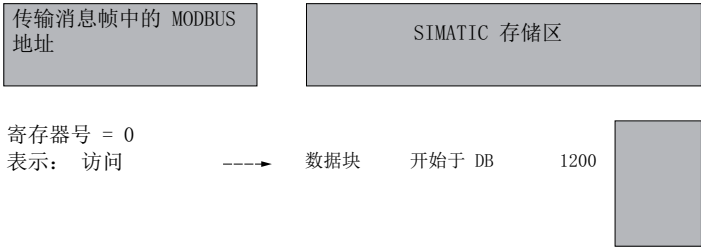
说明

“开始于 DB”的输入值与功能代码 03、06 和 16 的“开始于 DB”的输入值完全无关。

这意味对于 FC 04 可以使用第二个 SIMATIC 数据块区域（只读），完全独立于第一个数据块区域。

5.6 为可加载驱动分配参数

示例



MODBUS 寄存器地址 0 可以用于访问 SIMATIC 系统中从 DBW 0 开始的数据块 1200。更高的 MODBUS 寄存器地址 (≥ 512 , 1024 等) 访问此后的 DB, 如 DB 1201, 1202 等等。

5.6.4 写功能的限制

FC 05、06、15 和 16 的概述

表格 5- 7 写访问的 SIMATIC 限制 (FC 05、06、15 和 16)

参数		输入	含义
数据块 DB: 生成的 DB 编号	MIN	1 到 65535	第一个启用的 DB
	MAX	1 到 65535	最后一个启用的 DB MAX= 0 禁用所有 DB
存储器位 M (存储器字节编号)	MIN	0 到 65535	第一个启用的存储器字节
	MAX	1 到 65535	最后一个启用的存储器字节 MAX= 0 禁用所有存储器位
输出 Q (输出字节编号)	MIN	0 到 65535	第一个启用的输出字节
	MAX	1 到 65535	最后一个启用的输出字节 MAX= 0 禁用所有输出

“MIN”/“MAX”SIMATIC 存储区

对于写功能代码，可以指定访问上下限 (MIN/MAX)。只可以在启用区域进行写访问。

如果上限值为 0，则禁用整个区域。

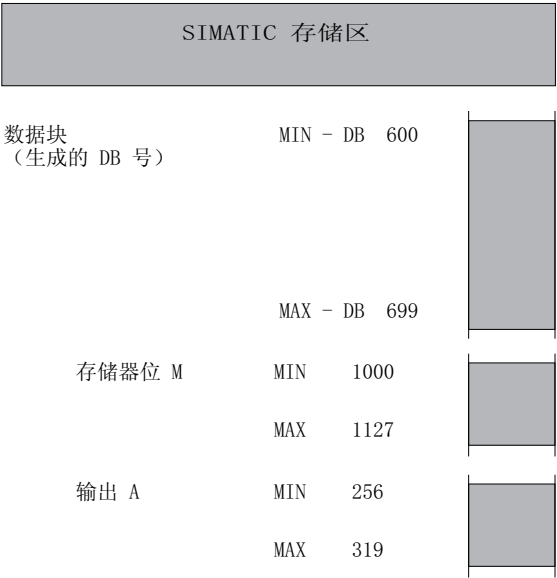
在选择启用区域大小时，注意所指的是哪个 CPU。

如果主站试图对超出上/下限的区域进行写访问，则会被 CP 拒绝，并产生一条错误消息帧。

数据块区的 MIN/MAX 值必须指定为生成的 DB 号。

5.6 为可加载驱动分配参数

示例



可以使用写功能代码 FC 06 和 FC 16 访问 SIMATIC 数据块 DB 600 到 DB 699。

可以使用写功能代码 FC 05 和 FC 15 访问 SIMATIC 存储器字节 MB 1000 到 MB 1127。

可以使用写功能代码 FC 05 和 FC 15 访问 SIMATIC 输出字节 QB 256 到 QB 319。

5.6.5 RS422/485 (X27) 接口

概述

表格 5- 8 RS422/485 (X27) 接口

参数	说明	值范围	默认值
操作模式	指定 RS 422/485 (X27) 接口是在全双工 (RS 422) 模式下运行还是在半双工 (RS 485) 模式下运行。	- 全双工 (RS 422) 四线制模式 - 半双工 (RS 485) 两线制模式	全双工 (RS 422) 四线制模式
接收线路的初始状态	无 (None): 两线模式下线 R(A) 和 R(B) 没有初始化。此时将由连接伙伴进行初始化。	- 无 - 信号 R(A) 5V / 信号 R(B) 0V（断路检测）¹ - 信号 R(A) 0V/ 信号 R(B) 5V	- 在“全双工 (RS422) 四线模式”下： 信号 R(A) 5V/ 信号 R(B) 0V（断路检测）¹ - 在“半双工 (RS485) 两线制模式”下： 信号 R(A) 0V/ 信号 R(B) 5V
	信号 R(A) 5V/ 信号 R(B) 0V（断路检测）： 该缺省设置支持“全双工 (RS422) 4 线制模式”中的断点检测。不能针对“半双工 (RS485) 两线模式”进行选择		
	信号 R(A) 0 V/ 信号 R(B) 5V： 该初始状态对应于“半全工 (RS 485) 两线制模式”下的空闲状态（无激活的发送器）。在此初始状态下，不能进行断点检测。		

1 仅在“全双工 (RS 422) 四线制模式”下。

5.6 为可加载驱动分配参数

“全双工 (RS422) 四线操作”

在该操作模式下，发送是通过发送线 T(A)- 和 T(B)+，而接收是通过接收线 R(A)- 和 R(B)+。

根据“驱动工作模式”参数的功能设置进行错误处理（常规操作或干扰抑制）。

“半双工 (RS485) 两线操作”

在此操作模式下，该驱动程序将在发送和接收操作之间切换接口的 2 线制接收线 R(A)-、R(B)+。

在此操作模式的初始设置下，所有在接收消息帧之前和之后识别的传输错误和/或断路 (BREAK) 都将被忽略。

在消息帧暂停期间的断路 (BREAK) 电平也将忽略。

正确接收到的从站地址可以检测到来自连接伙伴的接收消息帧的起始位置。

推荐“信号 R(A) 0V, R(B) 5V”作为节点的接收线的初始状态设置。所有其他节点的“接受线初始状态”应设置为“无”。

接收线路初始状态

“无”

两线模式下线 R(A)- 和 R(B)+ 没有初始化。此时将由连接伙伴进行初始化。

初始化 “R(A) 5V, R(B) 0V” (断路)

两线模式下的线 R(A)- 和 R(B)+ 将通过 CP 按如下方法初始化：

R(A) --> +5V, R(B) --> 0V ($V_A - V_B \geq +0.3V$)。

这就意味着在断线情况下 CP 将产生断路 (BREAK) 电平。

此选项仅在“全双工 (RS 422) 四线制模式”下可以选择。

初始化 “R(A) 0V, R(B) 5V”

两线模式下的线 R(A)- 和 R(B)+ 将通过 CP 按如下方法初始化：

R(A) --> 0V, R(B) --> +5V ($V_A - V_B \leq 0.3V$)。

这就意味着在线上没有传输时，在线缆断开和/或空闲状态情况下将产生高电平。无法检测断路线路状态。

参数选择

选择您的连接所必需的设置，并单击“确定”(OK) 退出单个屏幕窗口。

5.7 装载 CP 341 的组态和参数分配数据

数据存储

当您关闭“硬件配置”(hardware configuration)时，数据将自动保存到 STEP 7 项目中。

装载组态和参数

现在可以将组态和参数分配数据从编程设备在线装载到 CPU 中。选择 **目标系统 > 装载 (Target system > Load)** 菜单命令将数据传送到 CPU 中。

当 CPU 启动时，并且在 CPU 从 STOP（停止）切换到 RUN（运行）或者反过来切换时，只要通过 S7300 背板总线可以访问到 CP，CP 的模块参数将自动从 CPU 传送到 CP 中。

驱动程序代码没有储存在 CPU 中，而是通过参数分配接口直接存在 CP 341 上的保持性存储器中。然而，请注意因此没有编程设备就不能替换模块。

更多信息

STEP 7 用户手册提供了以下操作的详细说明：

- 保存组态和参数
- 将组态和参数装载到 CPU
- 读取、修改、复制和打印组态和参数。

5.8 装载驱动程序到 CP 341

要求

可以在线连接到 CPU。

装载驱动程序

1. 在"分配点到点连接参数"(Assigning Parameters to Point-to-Point Connection) 窗口内的"协议"(Protocol) 下拉框选择所需的可加载驱动程序。
2. 点击“装载驱动程序” (Load drivers) 图标。

在“装载驱动程序到 CP341”(Load drivers to CP341) 窗口内，您可以看到在线装载到模块的驱动程序版本，以及您在编程设备上离线选择的驱动程序版本。

3. 点击“装载驱动程序” (Load drivers) 按钮并用“是”(Yes) 确认。

装载驱动程序到 CP 341。

装载完成后，“模块上的在线驱动程序版本”(Driver version online on the module) 信息将会更新。

如果装载的驱动程序已经存在于 CP 341，装载操作将取消并显示消息“驱动程序已存在”(Driver already exists)。这种情况下，点击“确定”(OK) 确认并关闭“下载驱动程序到 CP341”(Download Drivers to CP341) 窗口。

如果驱动程序文件不存在或者不正确，将会收到错误信息“模块拒绝驱动程序下载”(Module rejected driver download)。在这种情况下，您必须重新安装驱动程序。

5.9 装载 CP 441-2 的组态和参数分配数据

数据存储

当关闭“硬件配置”和/或“连接组态”时，会自动将数据（包括模块参数和驱动程序代码）存储在 STEP 7 项目中。

装载组态和参数

现在可以将组态和参数分配数据从编程设备在线装载到 CPU 中。选择 **目标系统 > 装载 (Target system > Load)** 菜单命令将数据传送到 CPU 中。

当 CPU 启动时，只要通过 S7400 背板总线可以访问到 CP，CP 的模块参数将自动从 CPU 传送到 CP 中。

更多信息

STEP 7 用户手册提供了以下操作的详细说明：

- 保存组态和参数
- 将组态和参数装载到 CPU
- 读取、修改、复制和打印组态和参数。

5.10 CP 的启动特性

简介

CP 的启动分为两个阶段：

- 上电初始化 CP
- 参数分配

初始化

只要 CP 一上电，在执行完硬件测试程序后，CP 的固件即可运行。

参数分配

在参数分配过程中，CP 将接收分配到当前插槽的模块参数。

CP 现在即可运行。

5.11 “CPU 启动”的参数分配

简介

本章节仅介绍 CP 441-2，如果您使用的是 CP 341，本章可以跳过。

硬件配置 (Hardware Configuration)

要避免在 CPU-CP 启动时出现问题，在使用“**硬件配置**”(Hardware configuration) 对 **CPU** 执行 **参数分配** (parameter assignment) 时，应进行以下设置。

通过双击 CPU 或者单击 CPU 选择菜单命令 **编辑 > 对象属性** (Edit > Object properties) 启动参数分配后，将显示“**属性 — CPU**”(Properties - CPU) 页面。

在“**启动**”(Startup) 选项卡中，将“**监视时间**”(Monitoring Time for) 下的“**到模块的参数传送时间 (100ms)**”(Transfer of Parameters to Modules (100ms)) 设置为最小值 **1000 (= 100s)**。

原因：在为具有可加载驱动程序的 CP 441-2 接口分配参数时，除了已分配的参数外，还会将驱动程序代码传送到 CP。在上述时间对整个装载的过程进行监控（时间必须充足）。

5.11 “CPU 启动”的参数分配

调试通讯 FB

6.1 安装 FB

随附的光盘

MODBUS 从站通信 FB 是 STEP 7 项目的一部分，在库 *Modbus* 中安装驱动程序时，该项目存储在 STEP 7 软件的 EXAMPLES 目录下，名称为“Modsl”（对于 CP 441-2 或 CP 341）。

应该确保没有同名的项目存在。

传送

1. 项目文件 *Modsl* 包含一个可加载示例形式的完整 STEP 7 项目。
2. 如果要在自己的用户项目中继续工作，请将 MODBUS 通信 **FB 180**（对于 CP 441-2）或 **FB 80**（对于 CP 341）传送到您的用户项目。
3. 如果需要，请将启动 OB（如 **OB 100** 和 **OB 101**）、循环 **OB 1** 以及 **DB 180** 或 **DB 80** 传送到您的用户程序。

通过此操作，您将可以访问通信 FB 的 **调用示例** 以及 FB 的完整背景数据块。

说明

OB 1 和 OB 100/OB 101 也可以自行生成。

如果没有传送背景数据块，则必须在调用 OB 1/OB 100/OB 101 中的 FB 180 或 FB 80 时生成。

6.2 STEP 7 项目

STEP 7 项目

STEP 7 项目文件 *Modsl* 包含一个可加载示例形式的完整 STEP 7 项目，由以下内容组成：

- 采用 UR1、PS、CPU 和 CP 的硬件配置
- CP 参数分配
- 采用 OB 和 MODBUS 通讯 FB 的 STEP 7 程序

程序文件中的块仅用作示例，可由用户根据其需求进行更改。

如果有必要，可以根据要求重命名 MODBUS 通讯 FB。

CP 341 的 Modsl 内容

项目文件示例包含以下内容：

FB 80	功能块	CP 341 的 MODBUS 从站通讯 FB
DB 80	背景数据块	MODBUS FB 的背景数据块和工作区域
OB 1	示例 OB	循环程序
OB 100	示例 OB	冷启动
FB 7	P_RCV_RK	接收数据
FB 8	P_SND_RK	发送数据
SFC 36	MSK_FLT	屏蔽同步错误事件
SFC 37	DMSK_FLT	取消屏蔽同步错误事件
SFC 38	READ_ERR	读取事件状态寄存器
SFC 41	DIS_AIRT	延迟中断
SFC 42	EN_AIRT	启用中断
SFC 51	RDSYSST	读取 CPU 的系统区域 (SZL)

SFC 集成在 CPU 中，已添加了仅用于诊断的变量表。

CP 441-2 的 Modsl 内容

项目文件示例包含以下内容：

FB 180	功能块	CP 441-2 的 MODBUS 从站通讯 FB
DB 180	背景数据块	MODBUS FB 的背景数据块和工作区域
OB 1	示例 OB	循环程序
OB 100	示例 OB	冷启动
OB 101	示例 OB	热启动
SFB 12	BSEND	发送数据块
SFB 22	STATUS	读取错误消息区域 SYSTAT
SFC 36	MSK_FLT	屏蔽同步错误事件
SFC 37	DMSK_FLT	取消屏蔽同步错误事件
SFC 38	READ_ERR	读取事件状态寄存器
SFC 41	DIS_AIRT	延迟中断
SFC 42	EN_AIRT	启用
SFC 51	RDSYSST	读取 CPU 的系统区域 (SZL)

SFB 和 SFC 集成在 CPU 中，已添加了仅用于诊断的变量表。

6.3 FB 80 参数 (CP 341)

概述

名称	类型	数据类型	含义	允许的分配
LADDR	I	Int	CP 的基址	使用 HW Config 配置
START_TIMER	I	定时器	定时器超时初始化	
START_TIME	I	S5Time	时间值超时初始化	
OB_MASK	I	BOOL	屏蔽 I/O 访问错误，延迟中断	FALSE: I/O 访问错误未屏蔽。 TRUE: I/O 访问错误已屏蔽，且中断已延迟。
CP_START	I	BOOL	启动 FB 初始化	
CP_START_FM	I	BOOL	初始化已使用 CP_START 的上升沿激活。	
CP_START_NDR	O	BOOL	信息：从 CP 写入作业	
CP_START_OK	O	BOOL	启动已完成且无错	TRUE: 在监视时间结束前，初始化作业可以完成，而不会产生错误。
CP_START_ERROR	O	BOOL	启动已完成但有错	TRUE: 即使在监视时间结束后，初始化作业也无法无错完成。
ERROR_NR	O	字	错误号	分配，请参阅“诊断”一章
ERROR_INFO	O	字	其他错误信息	分配，请参阅“诊断”一章

6.4 FB 180 参数 (CP 441-2)

概述

名称	类型	数据类型	含义	允许的分配
ID	I	字	连接 ID	源自数据链接组态的连接 ID
START_TIMER	I	定时器	定时器超时初始化	
START_TIME	I	S5Time	时间值超时初始化	
STATUS_TIMER	I	定时器	SYSTAT 的读取定时器	
STATUS_TIME	I	S5Time	SYSTAT 的读取时间间隔	
OB_MASK	I	BOOL	屏蔽 I/O 访问错误，延迟中断	FALSE: I/O 访问错误未屏蔽。 TRUE: I/O 访问错误已屏蔽，且中断已延迟。
CP_START	I	BOOL	启动 FB 初始化	
CP_START_FM	I	BOOL	初始化已使用 CP_START 的上升沿激活。	
CP_START_NDR	O	BOOL	信息：从 CP 写入作业	
CP_START_OK	O	BOOL	启动已完成且无错	TRUE: 在监视时间结束前，初始化作业可以完成，而不会产生错误。
CP_START_ERROR	O	BOOL	启动已完成但有错	TRUE: 即使在监视时间结束后，初始化作业也无法无错完成。
ERROR_NR	O	字	错误号	分配，请参阅“诊断”一章
ERROR_INFO	O	字	其他错误信息	分配，请参阅“诊断”一章

6.5 程序调用

常规信息

可加载 MODBUS 从站驱动程序的 MODBUS 通信 FB 必须在 SIMATIC S7 CPU 的循环部分调用。

通信 FB 将初始化 CP 并执行那些驱动程序本身无法执行的 MODBUS 功能。尽管 MODBUS 主站系统不使用这些功能代码，但 MODBUS 从站通信 FB 必须在用户程序中调用。

CP 与 FB 之间的通信通过 CPU 操作系统功能和系统功能块 SFB BSEND (CP 441-2) 或从 FB 中调用的 P_SND_RK 和 P_RCV_RK (CP 341) 进行。

此外，MODBUS FB（仅适用于 CP 441-2 (FB 180)）将使用 SFB STATUS 读取 CP 的错误消息区域 SYSTA。

启动，初始化

每次 CPU 冷启动或暖启动之后，必须执行 MODBUS 通信 FB 的初始化。

通过 CP_START 输入处的上升沿激活初始化。

首先，FB 将删除背景数据块，从带有 SFC51 SZL_READ 的 CPU 读取操作数区域 I、Q、M、T 和 C，然后将其存储在背景数据块中。通过此操作，您可以检查 MODBUS 主站系统的写需求，以查看是否有区域溢出。

背景数据块的数量和完成的初始化序列将通过 SEND 作业传送到 CP。

SEND 作业无错完成后，将设置输出 CP_START_OK，且 FB 初始化也将完成。

如果 SEND 作业完成时发生错误，则将复位 CP_START，并设置 CP_START_ERROR。

如果初始化完成时发生错误，则不能进行 MODBUS 通信。所有源自 MODBUS 主站系统的请求都使用“异常代码”消息帧应答。

背景数据块

所有与 MODBUS FB 相关的数据都位于背景数据块中。此数据块也是所用 FB 和 SFB 的背景数据块（多重背景）以及 MODBUS 通信 FB 的工作区域。不需要其他的数据区。

MODBUS FB 仅使用背景数据块和本地数据。

只允许对背景数据块进行只读访问。

超时初始化

接通电源后，CP 需要几秒的时间来进行硬件和内存检查，直到准备好运行为止。
MODBUS FB 在这段时间内的初始化尝试由于错误而结束。因此，MODBUS FB 将在此次超时过程中重复几次初始化作业。

如果初始化可以在定时器 START-TIMER 的参数化时间 START-TIME 内完成，且不发生错误，则将置位 CP_START_OK。如果监视时间结束后初始化仍然无法无错完成，则将置位 CP_START_ERROR。

读取 SYSTAT 的时间值（仅适用于 CP 441-2）

由于在每个 PLC 循环或每两个 PLC 循环时对 SYSTAT 区域的循环读取必然会导致 CP 和 K 总线的过载和数据吞吐量的减少，因此，可以设置 SYSTAT 区域的读取时间间隔（仅与 CP 441-2 (FB180) 相关）。

STATUS_TIMER 中的时间间隔 STATUS_TIME 结束后，MODBUS FB 将激活 SFB STATUS，以读取 SYSTAT 区域。

I/O 访问错误，延迟中断

可以使用输入参数 OB_MASK 引导 MODBUS FB 屏蔽 I/O 访问错误。在对不存在的 I/O 进行写访问时，CPU 不会停止，也不会调用错误 OB。

FB 检测到访问错误后，会终止该功能，并向 CP 发送一条错误消息。仅当参数 OB_MASK 为 = TRUE 时，才会屏蔽发出写命令时的 I/O 访问错误。

在屏蔽访问错误前，所有优先级较高的中断都会延迟 (SFC 14)，并且会在对 FB 进行写访问和取消屏蔽访问错误后重新启用这些中断 (SFC 42)。

这确保了在屏蔽和取消屏蔽的间隙中断 FB 时，更高优先级的程序（如时间中断或过程中断）可识别访问错误。

示例 OB100/101

程序段 1	
AN M 180.0	// 设置 CP_START
S M 180.0	// !
A M 180.1	// 重置 CP_START_FM
R M 180.1	// !

CP 341 的示例 OB1

程序段 1		
CALL FB 80 , DB80		// MODBUS 从站 CP341 FB
LADDR	:=256	// CP 的基址
START_TIMER	:=T120	// 定时器超时初始化
START_TIME	:=S5T#5S	// 时间值超时
OB_MASK	:=TRUE	// 屏蔽错误
CP_START	:=M180.0	// 开始初始化
CP_START_FM	:=M180.1	// 沿触发存储器位
CP_NDR	:=M180.2	// 新建 CP 的写入作业
CP_START_OK	:=M180.3	// 初始化 CP-FB，无错
CP_START_ERROR	:=M180.4	// 初始化 CP-FB，有错
CP_ERROR_NR	:=MW182	// 错误号
CP_ERROR_INFO	:=MW184	// 其他错误信息

CP 441-2 的示例 OB1

程序段 1		
CALL FB 180 , DB180		// MODBUS 从站 CP441 FB
ID	:=W#16#1000	// 连接 ID
START_TIMER	:=T180	// 定时器超时初始化
START_TIME	:=S5T#5S	// 时间值超时
STATUS_TIMER	:=T181	// 定时器读取 SYSTAT
STATUS_TIME	:=S5T#2S	// 时间值读取 SYSTAT
OB_MASK	:=TRUE	// 屏蔽错误
CP_START	:=M180.0	// 开始初始化
CP_START_FM	:=M180.1	// 沿触发存储器位
CP_NDR	:=M180.2	// 新建 CP 的写入作业
CP_START_OK	:=M180.3	// 初始化 CP-FB，无错
CP_START_ERROR	:=M180.4	// 初始化 CP-FB，有错
CP_ERROR_NR	:=MW182	// 错误号
CP_ERROR_INFO	:=MW184	// 其他错误信息

6.6 周期性操作

通讯 FB

MODBUS 通讯 FB 执行所有必需的 SFB 调用并处理那些 CP 无法自行运行的功能代码，如使用 FC05 或 FC15 逐位写入到 SIMATIC 区域存储器位和输出。

SYSTAT 区域（仅适用于 CP 441-2）

CP 441-2 的 MODBUS FB 180 在参数化时间间隔内从 CP 的 SYSTAT 区域读取，并将数据存储在开始于 DBW 40 的背景数据块中。

您必需在用户程序中编写用户特定错误处理。

如果已在 SYSTAT 中置位错误位（位 2.0），则 SYSTAT 的第 4 到 15 字节（事件 1 到 6）必须始终存储在开始于 DBW 44 的背景数据块中。

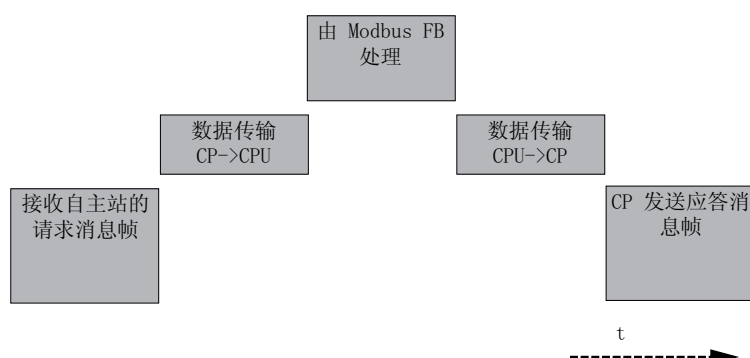
有关 SYSTAT 结构的详细说明，请参阅本手册的“诊断 (页 123)”一章和 CP 441-2 手册。

响应时间

需要 1 个 FB 序列、1 个 PLC 循环和 CP--->CPU 和 CPU--->CP 的数据传送时间，才能处理写功能代码 FC 05 和 FC 15。

由 CP 直接处理的其他功能仅需要 CP--->CPU 或 CPU--->CP 的数据传送时间。

在 CPU--->CP 的数据传送之前，CP 不会向主站系统发送应答消息帧。在此实例中，可以达到 2 秒的标准应答监视时间。



响应时间取决于 CPU 程序 (MODBUS FB) 的循环时间和 CPU 类型（数据传送 CPU<-->CP）。

6.6 周期性操作

CPU – CP 接口

7.1 CP 341 的 CPU-CP 接口

7.1.1 常规信息

MODBUS 通信 FB

CP 和 CPU 之间的数据传送由功能块 **P_SND_RK** 和 **P_RCV_RK** 执行。

提供的 MODBUS 通信 FB 将调用该 FB。

不需要在 SIMATIC 用户程序中编写任何其他的 FB 调用。

在用户程序中集成作业

要完全处理一份独立的 MODBUS 从站作业，必须多次打开 MODBUS 通信 FB FB 80。所需的读/写数据量决定需要打开该功能块的次数。如果在 CPU 的 OB1 循环中打开 MODBUS 通信 FB，则可加快数据交换。如果在“慢”循环中断中打开 MODBUS 通信 FB，则数据交换将需要更长的时间。

模块地址

剩下的任务即为在 MODBUS 通信 FB 处指定模块地址 LADDR。

7.1.2 数据传送长度 CPU <-> CP

数据传送长度

CP <-> CPU 之间的数据传送由功能块 P_SND_RK 和 P_RCV_RK 执行。

数据传送长度取决于功能代码（见“功能代码 (页 83)”一章）。

这表示，在 MODBUS 读访问或写访问操作超出以上指定的数据长度时，MODBUS 从站驱动程序会对带有 **Exception_Code** 的错误消息帧作出应答。

7.1.3 数据一致性

块大小

CPU 和 CP 之间的数据传送由功能块 P_SND_RK 和 P_RCV_RK 执行。

为了确保对 S7 自动化系统的系统中断作出稳定的响应处理，将逐块执行 CPU 与 CP 之间的数据传送。

访问 SIMATIC 存储区（数据块、存储器位等）的块大小为 32 字节。

数据一致性

仅给出以上列出的 32 字节块大小的数据传输期间的**数据一致性**。

对于较大的数据量，将在所列出的块大小中传送数据，并且每个块之间都有一个时间延迟。

无法保证个别块之间的数据一致性，因为用户程序可能会同时处理数据。

用户程序运行时，只要传递 P_RCV_PK，就会访问 CPU 内存。

MODBUS 从站

对于驱动程序 MODBUS 从站，这表示：

如果在读/写寄存器或位时需要**数据一致性**，则必须将单个消息帧传送的数据量限制到上述的块大小：例如，对于 FC 03、FC 04 和 FC 16，最大为 16 个寄存器，或者对于 FC 01、FC 02 和 FC 15，最大为 256 位。

如果需要，可以通过用户程序级适当的协调机制来确保相关数据区处理的一致性。

7.2 用于 CP 441-2 的 CPU-CP 接口

7.2.1 常规信息

MODBUS 通讯 FB

CP 与 CPU 之间的数据传送由集成系统功能 **PUT** 和 **GET** 以及 **SFB BSEND** 和 **STATUS** 执行。

提供的 MODBUS 通讯 FB 将调用 **SFB BSEND** 和 **STATUS**。

PUT 和 **GET** 是在 CPU 的后台程序中独立于用户程序运行的操作系统功能。

不需要在 SIMATIC 用户程序中编程任何其他的 **SFB** 调用。

通讯链接

剩下的任务即为在 MODBUS 通讯 FB 处指定链接配置的连接 ID。

参数 ID 描述了到通讯伙伴的唯一通讯链接。

7.2.2 数据传送长度 CPU <-> CP

数据传送长度

CP <-> CPU 之间的数据传送由系统功能 PUT 和 GET 执行。

数据传送长度取决于 CPU 类型：（另请参见参考手册“系统功能和标准功能”）

系统功能	S7-400
PUT/GET	450 字节

这表示，在 MODBUS 读访问或写访问操作超出以上指定的数据长度时，MODBUS 从站驱动程序会对带有 **Exception_Code** 的错误消息帧作出应答。

下表所示为最大数据量。

	S7-400	
MODBUS 功能代码	最大寄存器数	最大位数
01, 02		2040
03, 04, 16	127	
15		2040

7.2.3 数据一致性

块大小

CP 和 CPU 之间的数据传送由集成的系统功能 PUT 和 GET 执行。

为了确保对 S7 自动化系统的系统中断作出稳定的响应处理，将逐块执行 CPU 与 CP 之间的数据传送。

访问 SIMATIC 存储区（数据块、存储器位等）的块大小为 32 字节。

数据一致性

由于实用程序 PUT 和 GET 对 CPU 存储区域的访问与对 STEP 7 应用程序的访问异步执行，因此，仅提供上述块大小的数据传输期间**数据一致性**。

对于较大的数据量，将在所列出的块大小中传送数据，并且每个块之间都有一个时间延迟。

无法保证个别块之间的数据一致性，因为用户程序可能会同时处理数据。

另请注意，因为在 CPU 的后台程序中运行集成系统功能 PUT 和 GET，所以在用户程序运行时可随时访问 CPU 内存。

此特定于 S7 CPU 的系统响应在参考手册“系统功能和标准功能”的第 17 和 18 章（数据一致性）进行了介绍（另请参见本手册前言中的“其他信息源”）。

MODBUS 从站

对于驱动程序 MODBUS 从站，这表示：

如果在读/写寄存器或位时需要**数据一致性**，则必须将单个消息帧传送的数据量限制到上述的块大小：例如，对于 S7-400：对于 FC 03、FC 04 和 FC 16，最大为 16 个寄存器，或者对于 FC 01、FC 02 和 FC 15，最大为 256 位。

还应注意，MODBUS 功能代码的读和写访问任何时间都可以在用户程序内执行。

如果需要，可以通过用户程序级适当的协调机制来确保相关数据区处理的一致性。

传输协议

常规信息

使用的程序是明码、异步半双工的程序。

数据传输无须握手。

主站-从站关系

MODBUS 主站系统将启动传输，输出请求消息帧后，将等待从站的应答消息帧。从站与从站之间不能进行消息帧交换。

消息帧结构

“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**。然后传输数据。数据域的结构取决于使用的功能代码。消息帧结束时传送的是 **CRC 校验码**。

地址	功能	数据	CRC 校验
字节	字节	n 个字节	2 个字节

地址	MODBUS 从站地址
功能	MODBUS 功能代码
数据	消息帧数据：字节数、线圈编号和数据
CRC 校验	消息帧校验和

从站地址

从站地址设置范围为 1 到 255。地址可分配给总线上已定义的从站。

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。

广播消息不需要从站的应答消息帧。

功能代码

功能代码定义了消息帧的含义。 还定义了消息帧的结构。

CP 支持以下这些功能代码：

功能代码	符合 MODBUS 规范的功能
01	读线圈状态
02	读输入状态
03	读保持寄存器
04	读输入寄存器
05	写单个线圈
06	预设单个寄存器
08	环路测试
15	写多个线圈
16	预设多个寄存器

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、
- 寄存器数量等等

有关信息，请参见 “功能代码 (页 83)”部分。

CRC 校验

消息帧的最后是由 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。

消息帧的结尾

在 3.5 个字符传输时间段（字符延迟时间的 3.5 倍）内没有任何数据传输时，驱动程序将认为消息帧传输已经结束（见《MODBUS 协议参考指南》）。

因此消息帧结束的超时 (TIME_OUT) 取决于传输速率。

波特率	超时 (TIME_OUT)
76800 波特	0.5 毫秒
38400 波特	1 毫秒
19200 波特	2 毫秒
9600 波特	4 毫秒
4800 波特	8 毫秒
2400 波特	16 毫秒
1200 波特	32 毫秒
600 波特	64 毫秒
300 波特	128 毫秒

在“常规操作”期间，通过连接伙伴接收的 Modbus 消息帧将在收到与 TIME_OUT 对应的帧结束信息之后进行评估和格式校验。

在“干扰抑制”期间，使用正确的 CRC 码通过格式正确的接收帧识别帧的结束信息。

异常响应

当检测到主站的请求消息有错误时，比如：寄存器地址非法，从站将置位应答消息帧的功能代码的最高位。

其后，将传输一个字节的错误代码（异常代码），该代码描述了错误原因。

异常代码消息帧

从站的错误代码应答消息帧具有以下结构：

例如，从站地址 5，功能代码 5，异常代码 02

从站的响应消息帧 EXCEPTION_CODE_xx:

05H	从站地址
85H	功能代码
02H	异常代码 (1..4)
xxH	CRC 校验码“低字节”
xxH	CRC 校验码“高字节”

以下错误代码由 CP 发送：

异常代码	符合 MODBUS 规范的含义	原因
01	非法功能	接收到的功能代码非法。
02	非法数据地址	不支持对 SIMATIC 区域的访问（见参数分配—区域，限制）
03	非法数据值	长度大于 2040 位或 127 个寄存器， 对于 FC05，数据域不是 FF00 或 0000， 对于 FC08，诊断子代码 <>0000。
04	关联设备发生故障	尚未通过 MODBUS 通信 FB 执行初始化，或 FB 报告故障， CP <-> CPU 数据传输期间发生错误 （例如，数据块不存在，最大可传输数据长度超出（块大小 CPU <-> CP））。

RS 232C 辅助信号

若使用的是 RS 232C 接口子模块，则 CP 中存在下列的 RS 232C 辅助信号：

DCD	数据载体检测	(输入)
	检测到的数据载体	
DTR	数据终端就绪	(输出)
	CP 就绪	
DSR	数据集准备就绪	(输入)
	通信伙伴就绪	
RTS	请求发送	(输出)
	CP 发送准备就绪	
CTS	清除发送	(输入)
	通信伙伴可以从 CP 接收数据。(响应 CP 的 RTS = ON)	
RI	响铃指示	(输入)
	指示呼叫进入	

打开 CP 441 时，输出信号处于 OFF 状态（不活动）。

如果使用了 DTR/DSR 和 RTS/CTS 控制信号，则可以使用参数化接口**为点对点连接分配参数 (Assigning Parameters to Point-To-Point Connections)** 设置路径参数，或者通过用户程序中调用功能 (FB) 来控制它们。

使用 RS 232C 辅助信号

RS 232C 辅助信号可在以下情况下使用：

- 参数设置成自动使用所有 RS 232C 二次信号。
- 通过 FB V24_STAT 和 FB V24_SET 功能

说明

当参数设置成自动使用所有 RS 232C 辅助信号时，通过 V24_SET FB 实现的 RTS/CTS 数据流控制以及 RTS 和 DTR 控制都不允许使用！

另一方面，总是允许通过 FB V24_STAT 功能读取所有的 RS 232C 辅助信号。

下面一部分将描述如何控制和评估 RS 232C 辅助信号。

二次信号的自动使用

CP 中 RS 232C 辅助信号的自动使用是按如下步骤执行的：

- 只要 CP 通过参数设置切换成自动使用 RS 232C 辅助信号操作模式，在上电时它将 RTS 设置成 OFF，同时将 DTR 设置成 ON（CP 就绪）。
- 在 DTR 线路设置为 ON 之前，无法发送和接收消息帧。只要 DTR 仍设置为 OFF，便不能通过 RS 232C 接口接收任何数据。如果提出发送请求，请求将被中止，并生成相应的错误消息。
- 当发送请求生成时，RTS 将设置成 ON 并且已设置的数据输出等待时间将启动。当数据输出等待时间到并且 CTS = ON 时，数据将通过 RS 232C 接口发送。
- 如果 CTS 线路在数据输出时间内未设置为 ON 以便可以发送数据，或者 CTS 在传输过程中更改为 OFF，发送请求会被中止，并生成错误消息。
- 数据发送后，RTS 线路在 RTS OFF 设置的时间结束后设置为 OFF。CP340 不会等待 CTS 更改为 OFF。
- 一旦 DSR 线路设置为 ON，即可通过 RS 232C 接口接收数据。如果 CP 的接收缓冲区预警将要溢出，则 CP 将不响应。
- 如果 DSR 从 ON 更改为 OFF，则激活的发送请求或数据接收将被中止，并生成错误消息。消息“DSR = OFF（自动使用 V24 信号）”将输入到 CP 的诊断缓冲区中。

说明

RS 232C 辅助信号的自动使用仅在半双工模式下可以实现。如果设置了自动使用 RS 232C 辅助信号，则无论是 RTS/CTS 数据流控制还是 RTS 和 DTR 控制，都无法通过 FB 24_SET 实现。

时序图

下图说明了发送请求的时间顺序。

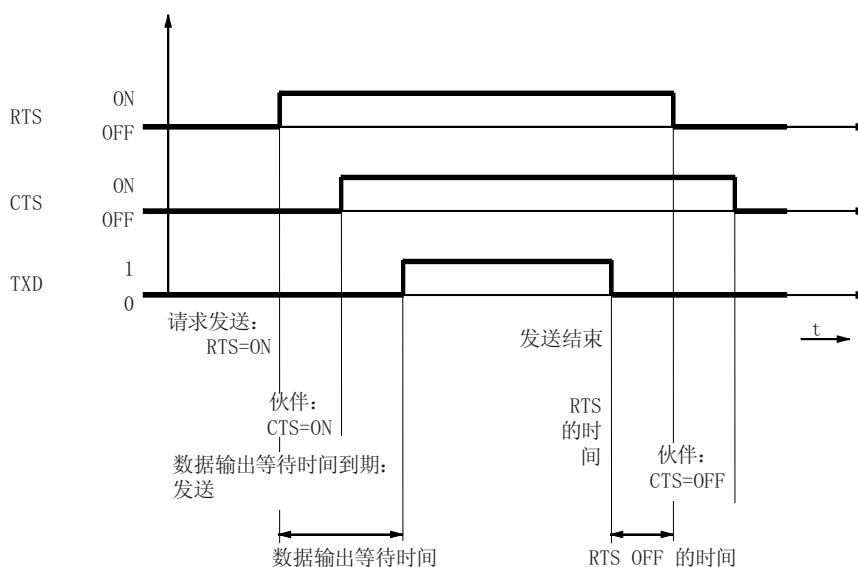


图 8-1 自动使用 RS 232C 辅助信号的时序图

功能代码

所用的功能代码

驱动程序支持下列 MODBUS 功能代码：

功能代码	符合 MODBUS 规范的功能	SIMATIC S7 中的功能	
01	读取线圈状态	逐位读取	存储器位 M
		逐位读取	输出 Q
		逐位读取 (16 位间隔)	定时器 T
		逐位读取 (16 位间隔)	计数器 C
02	读取输入状态	逐位读取	存储器位 M
		逐位读取	输入 I
03	读取保持寄存器	逐字读取	数据块 DB
04	读取输入寄存器	逐字读取	数据块 DB
05	写单个线圈	逐位写入	存储器位 M
		逐位写入	输出 Q
06	预设单个寄存器	逐字写入	数据块 DB
08	回送测试	-	-
15	写多个线圈	逐位写入 (1 至 2040 位)	存储器位 M
		逐位写入 (1 至 2040 位)	输出 Q
16	预设多个 (保持) 寄存器	逐字写入 (1 至 127 个寄存器)	数据块 DB

说明

下列所有 MODBUS 地址都指向 MODBUS 主站系统中的传输消息帧级别，而不是用户级别。

这意味着传输消息帧中的 MODBUS 地址以 0000 十六进制开头。

说明

MODBUS 从站驱动程序在所有访问 SIMATIC 范围“数据块 DB”（FC 03、FC 04、FC 06 和 FC 16）的功能代码中支持**最大 512 数据字的数据块长度**。

将地址从 MODBUS 转换为 SIMATIC 地址时，不能将 MODBUS 主站请求消息帧中的 1 个 DB 号“直接转换”为下一个 DB 号。MODBUS 从站将以异常代码消息帧对此请求消息帧作出响应，并将“0E 39”（访问 SIMATIC 范围“数据块”时出错）输出到诊断缓冲区。

例如，无法输出长度大于 2 的 DBX 字地址 510，因为寄存器编号大于 2（例如，寄存器 3）的地址在 DBX+1 中将自动表示为字地址 0。

9.1 功能代码 01 — 读取线圈（输出）状态

功能

该功能使 MODBUS 主站系统可以从下列 SIMATIC 存储区读取各个位。

请求消息帧

ADDR	FUNC	start_address	bit_number	CRC
------	------	---------------	------------	-----

应答消息帧

ADDR	FUNC	Byte_count n	n 个字节的 DATA	CRC
------	------	--------------	----------------	-----

start_address

MODBUS 位地址“start_address”由驱动程序解释为如下内容：

驱动程序将检查 “start_address”是否位于参数分配过程中在对话框“**FC 01、05、15 的 MODBUS 寻址转换**”(Conversion of MODBUS Addressing for FC 01, 05, 15) 中所指定的其中一个区域（从/到：存储器位、输出、定时器、计数器）。

如果 MODBUS 位地址 start_address 位于其中一个区域中，	则将访问此 SIMATIC 存储区	
从 aaaaa 到 bbbbbb	开始于存储器位	M uuuuu.0
从 ccccc 到 dddddd	开始于输出	Q ooooo.0
从 eeeee 到 fffff	开始于定时器	T ttttt
从 ggggg 到 hhhhh	开始于计数器	C zzzzz

9.1 功能代码 01 — 读取线圈（输出）状态

访问地址（地址转换）的计算如下：

访问开始处（使用 转换公式
SIMATIC）

存储器字节 = ((start_address - aaaaa) / 8) + uuuuu
输出字节 = ((start_address - ccccc) / 8) + ooooo
定时器 = ((start_address - eeeee) / 16) + ttttt
计数器 = ((start_address - ggggg) / 16) + zzzzz

访问 “存储器位” 和 “输出”

当访问 SIMATIC“存储器位”和“输出”区域时，会计算剩余的 Bit_Number 并将其用于对第一个/最后一个存储器或输出字节中的相应位进行寻址。

访问 “定时器” 和 “计数器”

通过地址计算，结果必须能够被 16 整除：

- (start_address - eeeee)
- (start_address - ggggg)

仅能从字限值开始逐字访问。

bit_number

1 到 2040 之间的值都可以作为 bit_number，线圈数。读取该位数。

访问 “定时器” 和 “计数器”

访问 SIMATIC“定时器”和“计数器”区域时，“bit_number”必须可以被 16 整除（仅适用于逐字访问）。

请注意，如果使用的是 CP 341，则最多可以读取 16 个“定时器”和/或“计数器”。

说明

请注意 “CPU-CP 接口 (页 69)”一章中描述的 CPU 特定限制。

应用示例

参数分配示例：

表格 9-1 功能代码 FC 01、05 和 15 的 Modbus 寻址的转换

消息帧传输中的 MODBUS 地址	SIMATIC 存储区	
从 0 到 2047	开始于存储器位	M 1000.0
从 2048 到 2559	开始于输出	Q256.0
从 4096 到 4607	开始于定时器	T 100
从 4608 到 5119	开始于计数器	C 200

请求消息帧 FUNCTION 01：

05H	从站地址 ADDR
01H	功能代码 FUNC
00H	start_address“高字节”
40H	start_address“低字节”
00H	bit_number“高字节”
20H	bit_number“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验码“高字节”

9.1 功能代码 01 — 读取线圈（输出）状态

应答消息帧 FUNCTION 01:

05H	从站地址 ADDR
01H	功能代码 FUNC
04H	Byte_count
01H	<DATA 1> M 1008.0 - M 1008.7
17H	<DATA 2> M 1009.0 - M 1009.7
02H	<DATA 3> M 1010.0 - M 1010.7
18H	<DATA 4> M 1011.0 - M 1011.7
xxH	CRC 校验码“低字节”
xxH	CRC 校验码“高字节”

地址计算:

MODBUS 地址“start_address”0040 十六进制（64 十进制）位于“存储器位”区域中:

存储器字节 = ((start_address - aaaaa) / 8) + uuuuu

 = ((64 - 0) / 8) + 1000

 = 1008 ;

剩余的其他 bit_number 的结果如下:

其他 bit_no. = ((start_address - aaaaa) % 8) [模数为 8]

 = ((64 - 0) % 8)

 = 0 ;

从位 M 1008.0 开始访问，一直访问到 M 1011.7（含）

位数:

MODBUS 位“bit_number”0020 十六进制（32 十进制）的值表示要读取 32 位（4 个字节）。

更多示例

下表列出了一些其它的访问示例。

所有示例都基于上述的区域规范。

start_address		在 SIMATIC 中的访问开始处					->	有
十六进制	十进制	(十进制)						
0000	0	存储器位	((0	- 0)	/ 8)	+ 1000	->	M 1000.0
0021	33	存储器位	((33	- 0)	/ 8)	+ 1000	->	M 1004.1
0400	1024	存储器位	((1024	- 0)	/ 8)	+ 1000	->	M 1128.0
0606	1542	存储器位	((1542	- 0)	/ 8)	+ 1000	->	M 1192.6
0840	2112	输出	((2112	- 2048)	/ 8)	+ 256	->	Q 264.0
09E4	2532	输出	((2532	- 2048)	/ 8)	+ 256	->	Q 316.4
1010	4112	定时器	((4112	- 4096)	/ 16)	+ 100	->	T 101
10C0	4288	定时器	((4288	- 4096)	/ 16)	+ 100	->	T 112 开始
1200	4608	计数器	((4608	- 4608)	/ 16)	+ 200	->	C 200
13E0	5088	计数器	((5088	- 4608)	/ 16)	+ 200	->	C 230

9.2 功能代码 02 — 读输入状态

功能

该功能使 Modbus 主站系统可以从下列 SIMATIC 存储区读取各个位。

请求消息帧

ADDR	FUNC	start_address	bit_number	CRC
------	------	---------------	------------	-----

应答消息帧

ADDR	FUNC	Byte_count n	n 个字节的 DATA	CRC
------	------	--------------	-------------	-----

start_address

MODBUS 位地址“start_address”由驱动程序解释为如下内容：

驱动程序将检查 “start_address”是否位于参数分配过程中在对话框“FC 02 的 MODBUS 寻址转换”(Conversion of MODBUS Addressing for FC 02) 中 输入的其中一个区域内（从 /至：存储器位，输入）。

如果 MODBUS 位地址位于区域 start_address 中，	则将访问 SIMATIC 存储区	
从 kkkkk 到 lllll	开始于存储器位	M vvvvv.0
从 nnnnn 到 rrrrr	从输入	I sssss.0 开始

访问地址（地址转换）的计算如下：

访问开始处（使用 转换公式
SIMATIC）

存储器字节 = ((start_address – kkkkk) / 8) + vvvvv

输入字节 = ((start_address – nnnnn) / 8) + sssss

访问“存储器位”和“输入”

当访问 SIMATIC“存储器位”和“输入”区域时，会计算剩余的 bit_number 并将其用于对第一个/最后一个存储器或输入字节中的相应位进行寻址。

bit_number

1 到 2040 之间的任何值都可以作为 bit_number，线圈数。读取位数。

说明

请注意“CPU-CP 接口 (页 69)”一章中描述的 CPU 特定限制。

应用示例

参数分配示例:

表格 9- 2 功能代码 FC 02 的 Modbus 寻址转换

消息帧传输中的 MODBUS 地址	SIMATIC 存储区	
从 0 到 4095	开始于存储器位	M 2000.0
从 4096 到 5119	开始于输入	I 128.0

请求消息帧 FUNCTION 02:

05H	从站地址 ADDR
02H	功能代码 FUNC
10H	start_address“高字节”
30H	start_address“低字节”
00H	bit_number“高字节”
18H	bit_number“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

应答消息帧 FUNCTION 02:

05H	从站地址 ADDR
02H	功能代码 FUNC
03H	Byte_count
12H	<DATA 1> I 134.0 - I 134.7
34H	<DATA 2> I 135.0 - I 135.7
56H	<DATA 3> I 136.0 - I 136.7
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

地址计算:

MODBUS 地址“start_address”1030 十六进制（4144 十进制）位于“输入”区域中:

输入字节 = ((start_address - nnnnn) / 8) + sssss
 = ((4144 - 4096) / 8) + 128
 = 134 ;

剩余的其他 bit_number 的结果如下:

其他 bit_no. = ((start_address - nnnnn) % 8) [以 8 为模]
 = ((4144 - 4096) % 8)
 = 0 ;

访问输入 I 134.0 到 I 136.7（含）。

位数:

MODBUS 位“bit_number”0018 十六进制（24 十进制）的值表示要读取 24 位（3 个字节）。

更多示例

下表列出了一些其它的访问示例。

所有示例都基于上述的区域规范。

start_address		在 SIMATIC 中的访问开始处					->	有
十六进制	十进制	(十进制)						
0000	0	存储器位	((0	- 0)	/ 8)	+ 2000	->	M 2000.0
0071	113	存储器位	((113	- 0)	/ 8)	+ 2000	->	M 2014.1
0800	2048	存储器位	((2048	- 0)	/ 8)	+ 2000	->	M 2256.0
0D05	3333	存储器位	((3333	- 0)	/ 8)	+ 2000	->	M 2416.5
1000	4096	输入	((4096	- 4096)	/ 8)	+ 128	->	I 128.0
10A4	4260	输入	((4260	- 4096)	/ 8)	+ 128	->	I 148.4

9.3 功能代码 03 – 读输出寄存器

功能

此功能使 MODBUS 主站系统可以从数据块中读取数据字

请求消息帧

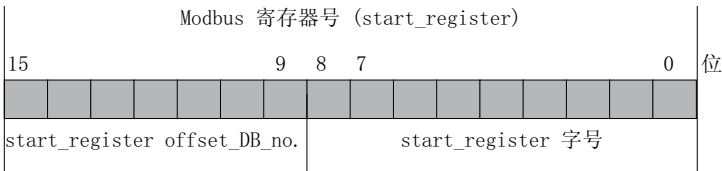
ADDR	FUNC	start_register	register_number	CRC
------	------	----------------	-----------------	-----

应答消息帧

ADDR	FUNC	Byte_count n	n/2 寄存器 DATA（高字节、低字节）	CRC
------	------	--------------	-----------------------	-----

start_register

MODBUS 寄存器地址“start_address”由驱动程序解释为如下内容：



为了进一步生成地址，驱动程序将使用参数分配过程中在对话框 “FC 03、06、16 的 MODBUS 寻址转换”(Conversion of MODBUS Addressing for FC 03, 06, 16) 中输入的 “基本 DB 号”（开始于 DB xxxxx）。

访问地址（地址转换）的计算如下：

访问 SIMATIC

数据块 DB

（生成的 DB）

数据字 DBW

转换公式

= （基本 DB 号 xxxxx + start_register offset_DB_No.）

= (start_register word_No. * 2)

start_register 的计算公式

如果已知要读取所生成的 DB，则可以根据以下公式计算主站系统中所需的 MODBUS 地址 **start_register**:

$$\text{start_register} = ((\text{生成的 DB} - \text{基本 DB 号}) * 512) + (\text{data word_DBW}/2)$$

这仅基于偶数的数据字数。

register_number

1 到 127 之间的任何值都可以作为 **register_number**（寄存器数）。读取该寄存器数。请遵守以下规则：

$$(\text{register_number})_{\text{max}} = 512 - \text{start_register}$$

说明

请注意“CPU-CP 接口 (页 69)”一章中描述的 CPU 特定限制。

应用示例

参数分配示例：

表格 9-3 功能代码 FC 03、06 和 16 的 Modbus 寻址的转换

传输消息帧中的 MODBUS 地址	SIMATIC 存储区	
0	开始于数据块 (基本 DB 号)	DB 800

9.3 功能代码 03 – 读输出寄存器

请求消息帧 FUNCTION 03:

05H	从站地址 ADDR
03H	功能代码 FUNC
00H	start_register“高字节”
50H	start_register“低字节”
00H	register_number“高字节”
02H	register_number“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

应答消息帧 FUNCTION 03:

05H	从站地址 ADDR
03H	功能代码 FUNC
04H	Byte_count
87H	<DATA 1> DBW 160“高字节”
65H	<DATA 2> DBW 160“低字节”
43H	<DATA 3> DBW 161“高字节”
21H	<DATA 4> DBW 161“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

地址计算:

MODBUS 地址“start_register”0050 十六进制（80 十进制）解释如下所示:



将访问 DB 800，数据字 DBW 160。

寄存器数:

MODBUS 寄存器数“register_number”0002 十六进制（2 十进制）表示要读取 2 个寄存器（2 个数据字）。

更多示例

下表列出了一些其它的访问示例。

		start_register					
start_register		基本 DB 号	偏移 DB 号	字编号		生成的 DB	DBW
十六进制	十进制	十进制	十进制	十六进制	十进制	十进制	十进制
0000	0	800	0	000	0	800	0
01F4	500	800	0	1F4	500	800	1000
0200	512	800	1	000	0	801	0
02FF	767	800	1	0FF	255	801	510
0300	768	800	1	100	256	801	512
03FF	1023	800	1	1FF	511	801	1022
0400	1024	800	2	000	0	802	0

9.4 功能代码 04 – 读输入寄存器

功能

此功能使 MODBUS 主站系统可以从数据块中读取数据字。

请求消息帧

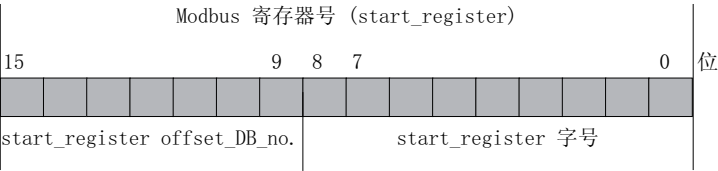
ADDR	FUNC	start_register	register_number	CRC
------	------	----------------	-----------------	-----

应答消息帧

ADDR	FUNC	Byte_count n	n/2 寄存器 DATA（高字节、低字节）	CRC
------	------	--------------	-----------------------	-----

start_register

MODBUS 寄存器地址“start_address”由驱动程序解释为如下内容：



为了进一步生成地址，驱动程序将使用在对话框 “FC4 的 MODBUS 寻址转换”(Conversion of MODBUS addressing for FC4) 中指定的“基本 DB 号”（开始于 DB xxxxx）。

访问地址（地址转换）的计算分如下 2 步：

访问	转换公式
SIMATIC	
数据块 DB	= （基本 DB 号 xxxxx + start_register-offset_DB_no.）
（生成的 DB）	
数据字 DBW	= (start_register word_No.* 2)

start_register 的计算公式

如果已知要读取所生成的 DB，则可以根据以下公式计算主站系统中所需的 MODBUS 地址 **start_register**:

$$\text{start_register} = ((\text{生成的 DB} - \text{基本 DB 号}) * 512) + (\text{data word_DBW}/2)$$

仅允许偶数的数据字数。

register_number

1 到 127 之间的任何值都可以作为 **register_number**（寄存器数）。读取该寄存器数。请遵守以下规则：

$$(\text{register_number})_{\text{max}} = 512 - \text{start_register}$$

说明

请注意“CPU-CP 接口 (页 69)”一章中描述的 CPU 特定限制。

应用示例

参数分配示例：

表格 9-4 功能代码 FC 04 的 MODBUS 寻址的转换

消息帧传输中的 MODBUS 地址	SIMATIC 存储区	
0	开始于数据块 (基本 DB 号)	DB 900

请求消息帧 FUNCTION 04:

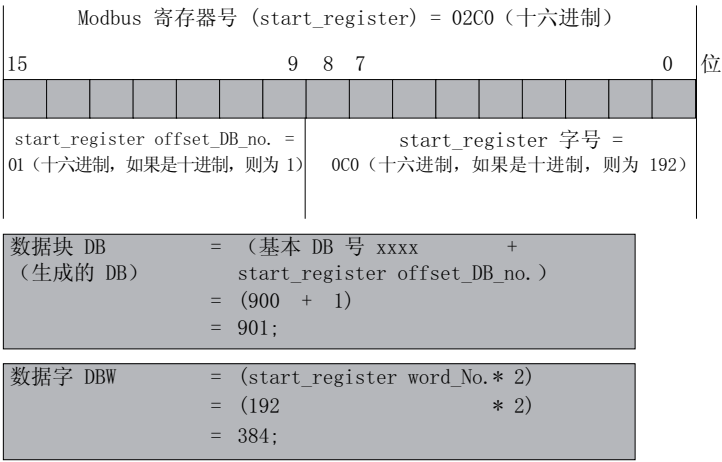
05H	从站地址 ADDR
04H	功能代码 FUNC
02H	start_register“高字节”
C0H	start_register“低字节”
00H	register_number“高字节”
03H	register_number“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

应答消息帧 FUNCTION 04:

05H	从站地址 ADDR
04H	功能代码 FUNC
06H	Byte_count
A1H	<DATA 1> DBW 384“高字节”
A2H	<DATA 2> DBW 384“低字节”
A3H	<DATA 3> DBW 385“高字节”
A4H	<DATA 4> DBW 385“低字节”
A5H	<DATA 5> DBW 386“高字节”
A6H	<DATA 6> DBW 386“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

地址计算:

MODBUS 地址“start_register”02C0 十六进制（704 十进制）的解释如下:



将访问 DB 901，数据字 DBW 384。

寄存器数:

MODBUS 寄存器数“register_number”0003 十六进制（3 十进制）表示要读取 3 个寄存器（3 个数据字）。

更多示例

下表列出了一些其它的访问示例。

		start_register					
start_register		基本 DB 号	偏移 DB 号	字编号		生成的 DB	DBW
十六进制	十进制	十进制	十进制	十六进制	十进制	十进制	十进制
0000	0	900	0	000	0	900	0
0064	100	900	0	064	100	900	200
00C8	200	900	0	0C8	200	900	400
0190	400	900	0	190	400	900	800
1400	5120	900	10	000	0	910	0
1464	5220	900	10	064	100	910	200
14C8	5320	900	10	0C8	200	910	400

9.5 功能代码 05 — 写单个线圈

功能

该功能使 MODBUS 主站系统可以向下面列出的 CPU 的 SIMATIC 存储器写入一个位。

请求消息帧

ADDR	FUNC	coil_address	DATA 开/关	CRC
------	------	--------------	----------	-----

应答消息帧

ADDR	FUNC	coil_address	DATA 开/关	CRC
------	------	--------------	----------	-----

coil_address

MODBUS 位地址“coil_address”由驱动程序解释为如下内容：

驱动程序将检查 “coil_address”是否位于参数分配过程中在对话框“FC 01、05、15 的 MODBUS 寻址的转换”(Conversion of MODBUS Addressing for FC 01, 05, 15) 中 输入的 其中一个区域（从/至：存储器位，输出，定时器，计数器）。

如果 MODBUS 位地址 coil_address 位于该区域	则将访问 SIMATIC 存储区	
从 aaaaa 到 bbbbb	开始于存储器位	M uuuuu.0
从 ccccc 到 ddddd	开始于输出	Q ooooo.0

访问地址（地址转换）的计算如下：

访问开始处（使用 SIMATIC）	转换公式
存储器字节	= ((coil_address - aaaaa) / 8) + uuuuu
输出字节	= ((coil_address - ccccc) / 8) + ooooo

访问存储器位和输出

当访问 SIMATIC“存储器位”区和“输出”区时，会计算剩余的 bit_number 并将其用于寻址存储器字节或输出字节内的相关位。

访问“定时器”和“计数器”

使用功能代码 FC 05 不允许访问 SIMATIC 定时器区和计数器区，此类访问将被驱动程序拒绝，并输出错误消息帧。

DATA 开/关

在 DATA 开/关时允许以下两个值：

- FF00H -> 设置位。
- 0000H -> 删除位。

应用示例

参数分配示例：

表格 9- 5 功能代码 FC 01、05、15 的 MODBUS 寻址的转换

传输消息帧中的 MODBUS 地址	SIMATIC 存储区	
从 0 到 2047	开始于存储器位	M 1000.0
从 2048 到 2559	开始于输出	Q 256.0

请求消息帧 FUNCTION 05:

05H	从站地址 ADDR
05H	功能代码 FUNC
08H	coil_address“高字节”
09H	coil_address“低字节”A257.1
FFH	DATA 开/关“高字节”
00H	DATA 开/关“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

应答消息帧 FUNCTION 05:

05H	从站地址 ADDR
05H	功能代码 FUNC
08H	coil_address“高字节”
09H	coil_address“低字节”A257.1
FFH	DATA 开/关“高字节”
00H	DATA 开/关“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

地址计算：

MODBUS 地址“coil_address”0809 十六进制（2057 十进制）位于“输出”区中：

$$\begin{aligned}\text{输出字节} &= ((\text{coil_address} - \text{cccc}) / 8) + 00000 \\ &= ((2057 - 2048) / 8) + 256 \\ &= 257 ;\end{aligned}$$

剩余的其他 bit_number 的结果如下：

$$\begin{aligned}\text{其他 bit_no.} &= ((\text{coil_address} - \text{cccc}) \% 8) \quad [\text{模数为 } 8] \\ &= ((2057 - 2048) \% 8) \\ &= 1 ;\end{aligned}$$

访问输出 Q 257.1。

更多示例

有关访问存储器位和输出的更多示例，请参考 FC 01。

9.6 功能代码 06 — 预设单个寄存器

功能

该功能使 MODBUS 主站系统能够在 CPU 的数据块中写入一个数据字。

请求消息帧

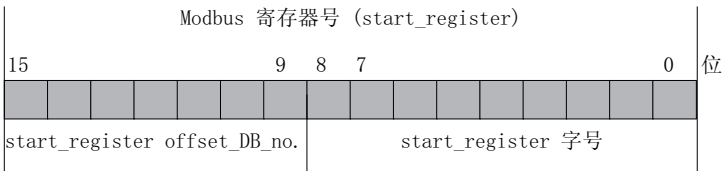
ADDR	FUNC	start_register	DATA 值（高字节、低字节）	CRC
------	------	----------------	-----------------	-----

应答消息帧

ADDR	FUNC	start_register	DATA 值（高字节、低字节）	CRC
------	------	----------------	-----------------	-----

start_register

MODBUS 寄存器地址“start_register”由驱动程序解释为如下内容：



为了进一步生成地址，驱动程序将使用参数分配过程中在对话框“FC 03、06、16 的 MODBUS 寻址的转换” (Conversion of MODBUS Addressing for FC 03, 06, 16) 中输入的“基本 DB 号”（开始于 DB xxxxx）。

访问地址（地址转换）的计算分如下 2 步：

访问	转换公式
SIMATIC	
数据块 DB	= （基本 DB 号 xxxxx + start_register-offset_DB_no.）
（生成的 DB）	
数据字 DBW	= (start_register word_No. * 2)

start_register 的计算公式

如果已知要写入所生成的 DB，则可以根据以下公式计算主站系统中所需的 MODBUS 地址 **start_register**：

$$\text{start_register} = ((\text{生成的 DB} - \text{基本 DB 号}) * 512) + (\text{data word_DBW}/2)$$

仅允许偶数的数据字数。

DATA 值

任何值都可用作 **DATA 值**（寄存器值）。

应用示例**参数分配示例：**

表格 9-6 功能代码 FC 03、06 和 16 的 Modbus 寻址的转换

消息帧传输中的 MODBUS 地址	SIMATIC 存储区	
0	开始于数据块 (基本 DB 号)	DB 800

请求消息帧 FUNCTION 06：

05H	从站地址 ADDR
06H	功能代码 FUNC
01H	start_register“高字节”
80H	start_register“低字节”DBW 768
2BH	DATA 值“高字节”
1AH	DATA 值“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码“高字节”

应答消息帧 FUNCTION 06:

05H	从站地址 ADDR
06H	功能代码 FUNC
01H	start_register“高字节”
80H	start_register“低字节”DBW 768
2BH	DATA 值“高字节”
1AH	DATA 值“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

地址计算:

MODBUS 地址“start_register”0180 十六进制（384 十进制）的解释如下所示:

Modbus 寄存器号 (start_register) = 0180 (十六进制)																位
15							9	8	7							
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
start_register offset_DB_no. = 00 (十六进制, 如果是十进制, 则为 0)								start_register 字号 = 180 (十六进制, 如果是十进制, 则为 384)								
数据块 DB = (基本 DB 号 xxxx + start_register offset_DB_no.) = (800 + 0) = 800;																
数据字 DBW = (start_register word_No.* 2) = (384 * 2)= 768;																

将访问 DB 800，数据字 DBW 768。

更多示例

有关更多访问示例，请参考 FC 03。

9.7 功能代码 08 — 回送诊断测试

功能

该功能用于检查通讯连接。

它对 S7 CPU、用户程序或用户数据没有影响。接收的消息帧通过驱动程序独立返回到主站系统中。

请求消息帧

ADDR	FUNC	诊断代码（高字节、低字节）	测试数据	CRC
------	------	---------------	------	-----

应答消息帧

ADDR	FUNC	诊断代码（高字节、低字节）	测试数据	CRC
------	------	---------------	------	-----

诊断代码

仅支持诊断代码 0000！

测试数据

任何 16 位值。

应用示例

请求消息帧 FUNCTION 08:

05H	从站地址 ADDR
08H	功能代码 FUNC
00H	诊断代码“高字节”
00H	诊断代码“低字节”
A5H	测试值“高字节”
C3H	测试值“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码“高字节”

应答消息帧 FUNCTION 08:

05H	从站地址 ADDR
08H	功能代码 FUNC
00H	诊断代码“高字节”
00H	诊断代码“低字节”
A5H	测试值“高字节”
C3H	测试值“低字节”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码“高字节”

9.8 功能代码 15 — 写多个线圈

功能

该功能使 MODBUS 主站系统可以向下列 SIMATIC 存储区写入多个位。

请求消息帧

ADDR	FUNC	start_address	数量	Byte_count n	n DATA	CRC
------	------	---------------	----	--------------	--------	-----

应答消息帧

ADDR	FUNC	start_address	数量	CRC
------	------	---------------	----	-----

start_address

MODBUS 位地址“**start_address**”由驱动程序解释为如下内容：

驱动程序将检查 “**start_address**”是否位于参数分配过程中在对话框“**FC 01、05、15 的 MODBUS 寻址转换**”(Conversion of MODBUS Addressing for FC 01, 05, 15) 中所输入的其中一个区域（从/到：存储器位、输出、定时器、计数器）。

如果 MODBUS 位地址位于区域 start_address 中，	则将访问 SIMATIC 存储区	
从 aaaaa 到 bbbbbb	开始于存储器位	M uuuuu.0
从 ccccc 到 ddddd	开始于输出	Q ooooo.0

访问地址（地址转换）的计算如下：

访问开始处（使用 SIMATIC）	转换公式
存储器字节	= ((start_address - aaaaa) / 8) + uuuuu
输出字节	= ((start_address - ccccc) / 8) + ooooo

访问“存储器位”和“输出”

当访问 SIMATIC“存储器位”区和“输出”区时，会计算剩余的 bit_number 并将其用于寻址存储器字节或输出字节内的相关位。

访问“定时器”和“计数器”

功能代码 FC 15 不允许访问 SIMATIC 定时器区和计数器区，此类访问将被驱动程序拒绝，并输出错误消息帧。

数量

1 到 2040 之间的任何值都可以作为数量（位数）。

说明
请注意“CPU-CP 接口 (页 69)”一章中描述的 CPU 特定限制。

数据

包含在数据域中的任何值都可作为位状态。

应用示例

参数分配示例：

表格 9-7 功能代码 FC 01、05 和 15 的 Modbus 寻址的转换

消息帧传输中的 MODBUS 地址	SIMATIC 存储区	
0 到 2047	开始于存储器位	M 1000.0
从 2048 到 2559	开始于输出	Q 256.0

操作：

MODBUS 主站系统要将以下位状态写入存储器位 M 1144.0 ... M 1144.7 和 M 1145.0 ... M 1145.3:

存储器位 M 1144	7	6	5	4	3	2	1	0	位
	亮	亮	不亮	不亮	亮	亮	不亮	亮	
存储器位 M 1145	7	6	5	4	3	2	1	0	位
	-	-	-	-	亮	不亮	不亮	亮	

请求消息帧 FUNCTION 15:

05H	从站地址 ADDR
0FH	功能代码 FUNC
04H	start_address“高字节”
80H	start_address“低字节”(M 1144.0 ...
00H	数量 “高”
0CH	数量 “低” （12 位）
02H	bytecount
CDH	状态线圈 (M 1144.0 ... M 1144.7)
09H	状态线圈 (M 1145.0 ... M 1145.3)
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

应答消息帧 FUNCTION 15:

05H	从站地址 ADDR
0FH	功能代码 FUNC
04H	start_address“高字节”
80H	start_address“低字节”
00H	数量 “高”
0CH	数量 “低”
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

地址计算：

MODBUS 地址“coil_address”0480 十六进制（1152 十进制）位于“存储器位”区中：

$$\begin{aligned}\text{存储器字节} &= ((\text{start_address} - \text{aaaaa}) / 8) + \text{uuuuu} \\ &= ((1152 - 0) / 8) + 1000 \\ &= 1144 ;\end{aligned}$$

剩余的其他 bit_number 的结果如下：

$$\begin{aligned}\text{其他 bit_no.} &= ((\text{start_address} - \text{aaaaa}) \% 8) \text{ [模数为 8]} \\ &= ((1152 - 0) \% 8) \\ &= 0 ;\end{aligned}$$

将访问开始于 M 1144.0 的存储器位。

更多示例

有关访问存储器位和输出的更多示例，请参考 FC 01。

9.9 功能代码 16 — 预设多个寄存器

功能

该功能代码使 MODBUS 主站系统能够在 SIMATIC CPU 的数据块中写入多个数据字。

请求消息帧

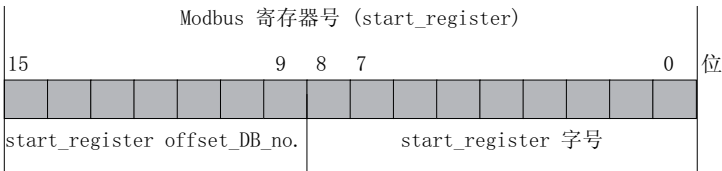
ADDR	FUNC	start_register	数量	Byte_count n	n-DATA（高字节、低字节）	CRC
------	------	----------------	----	--------------	-----------------	-----

应答消息帧

ADDR	FUNC	start_register	数量	CRC
------	------	----------------	----	-----

start_register

MODBUS 寄存器地址“start_register”由驱动程序解释为如下内容：



为了进一步生成地址，驱动程序将使用参数分配过程中在对话框“FC 03、06、16 的 MODBUS 寻址的转换”(Conversion of MODBUS Addressing for FC 03, 06, 16) 中输入的“基本 DB 号”(开始于 DB xxxxx)。

访问地址（地址转换）的计算分如下 2 步：

访问 SIMATIC	转换公式
数据块 DB (生成的 DB)	= (基本 DB 号 xxxxx + start_register-offset_DB_no.)
数据字 DBW	= (start_register word_No. * 2)

start_register 的计算公式

如果已知要写入所生成的 DB，则可以根据以下公式计算主站系统中所需的 MODBUS 地址 **start_register**:

start_register = ((生成的 DB — 基本 DB 号) * 512) + (data word_DBW/2)

仅允许偶数的数据字数。

数量

1 到 127 之间的任何值都可以作为**数量**（寄存器数）。请遵守以下规则:

(数量)_{最大值} = 512 - start_register

说明

请注意“CPU-CP 接口 (页 69)”一章中描述的 CPU 特定限制。

DATA（高字节、低字节）

任何值都可以用作 **DATA DATA（高字节、低字节）**（寄存器值）。

应用示例

参数分配示例:

表格 9-8 功能代码 FC 03、06、16 的 MODBUS 寻址的转换

传输消息帧中的 MODBUS 地址	SIMATIC 存储区	
0	开始于数据块 (基本 DB 号)	DB 800

操作：

MODBUS 主站系统要将值 CD09 十六进制、DE1A 十六进制和 EF2B 十六进制写入 DB 800 的数据字 DBW 100、DBW 102 和 DBW 104。

请求消息帧 FUNCTION 16:

05H	从站地址 ADDR
10H	功能代码 FUNC
00H	start_register“高字节”
32H	start_register“低字节”DBW 100
00H	数量 “高”
03H	数量 “低” （3 个寄存器）
06H	bytecount
CDH	寄存器值 — 高 (DBW100)
09H	寄存器值 — 低
DEH	寄存器值 — 高 (DBW102)
1AH	寄存器值 — 低
EFH	寄存器值 — 高 (DBW104)
2BH	寄存器值 — 低
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

应答消息帧 FUNCTION 16:

05H	从站地址 ADDR
10H	功能代码 FUNC
00H	start_register“高字节”
32H	start_register“低字节”
00H	数量 “高”
03H	数量 “低” （3 个寄存器）
xxH	CRC 校验码“低字节”
xxH	CRC 校验代码 “高字节”

地址计算:

MODBUS 地址“start_register”0032 十六进制（50 十进制）的解释如下所示:

Modbus 寄存器号 (start_register) = 0032 (十六进制)																位	
15							9	8	7								0
	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	start_register 字号 = 32 (十六进制, 如果是十进制, 则为 50)
start_register offset_DB no. = 00 (十六进制, 如果是十进制, 则为 0)																	
数据块 DB = (基本 DB 号 xxxx + (生成的 DB) start_register offset_DB_no.) = (800 + 0)= 800;																	
数据字 DBW = (start_register word_No.* 2) = (50 * 2) = 100;																	

访问 DB 800, 数据字 DBW 100。

更多示例

有关更多访问示例, 请参考 FC 03。

驱动程序的诊断

诊断功能

CP 的诊断功能使您可以快速地定位发生的错误。可使用下列诊断功能：

- 通过 CP 的显示元件进行诊断
- 通过功能块的 **STATUS** 输出进行诊断
- 通过 **SYSTAT** 错误消息区域进行诊断（仅适用于 CP 441-2）
- CP 的诊断缓冲区

指示灯 (LED)

显示元件提供有关 CP 的操作模式和/或可能的错误状态的信息，并使您对所有的内部或外部错误以及接口特定错误有一个初步了解。

FB/SFB 的 STATUS 输出

每个功能块和系统功能块都有一个用于错误诊断的 **STATUS** 输出。读取 **STATUS** 输出可以获得关于在通信期间发生的错误的信息。可以在用户程序中评估 **STATUS** 参数。

错误消息区 SYSTAT（仅适用于 CP 441-2）

错误消息区 **SYSTAT** 是 CP 441-2 上的存储区，CP 识别到的所有错误和事件都详细输入到此区域。可以通过在用户程序中对系统功能块 **STATUS** 进行编程来读取 **SYSTAT** 区域。

CP 的诊断缓冲区

在“错误/事件表 (页 133)”一章中描述的所有错误和事件也会输入到 CP 的诊断缓冲区内。CP 的手册说明了如何读取诊断缓冲区。

10.1

CP 341 上的诊断工具

10.1.1

通过 CP 341 的显示元件进行诊断

显示元件

CP 341 的显示元件提供了有关 CP 341 的信息。可使用以下显示功能：

组错误显示	
• SF（红色）	发生的错误或分配的新参数
特殊显示	
• TXD（绿色）	发送 (Send) 激活；在 CP 341 通过接口发送用户数据时亮起。
• RXD（绿色）	接收 (Receive) 激活；在 CP 341 通过接口接收用户数据时亮起。

组错误显示 SF

组错误显示 SF 始终会在通电后亮起，在初始化后熄灭。 如果已经为 CP 341 生成了参数分配数据，则当分配了新参数时 SF LED 再次短暂亮起。

当发生下列错误时，组错误显示 SF 亮起：

- 硬件错误
- 固件错误
- 参数分配错误
- 断路 (BREAK)（CP 341 和通讯伙伴之间的接收线路中断。）

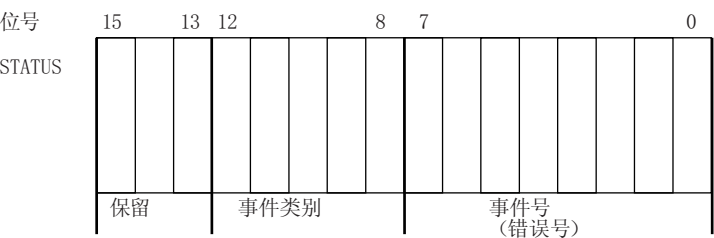
10.1.2 CP 341 功能块的诊断消息

简介

每个功能块都有一个用于错误诊断的 STATUS 参数。无论使用哪个功能块，每个 STATUS 消息编号始终具有相同的含义。

事件类别/事件编号的编号方案

下图说明了 STATUS 参数的结构。



个别错误/事件将在“错误/事件表 (页 133)”一章中列出。

10.2 CP 441-2 上的诊断工具

10.2.1 通过 CP 441-2 的显示元件进行诊断

显示功能

CP 441-2 的显示元件提供了有关 CP 441-2 的信息。可使用以下显示功能：

组错误显示	
INTF	内部错误
EXTF	外部错误
特殊显示	
TXD	发送激活；指示灯在 CP 441-2 通过接口发送用户数据时亮起。
RXD	接收激活；指示灯在 CP 441-2 通过接口接收用户数据时亮起。
接口错误显示	
FAULT	接口错误

显示元件的错误消息

下表说明了显示元件的错误消息。

故障显示	错误说明	解决方法
INTF 亮起	CP 441-2 报告内部错误；例如，硬件或软件错误。	对 SFB STATUS 进行编程，以获取详细信息。
EXTF 亮起	CP 441-2 报告外部错误；例如，接收线路断路。	对 SFB STATUS 进行编程，以获取详细信息。
FAULT 熄灭	接口准备好开始工作，或接口子模块未插入。	-
FAULT 缓慢闪烁	接口已初始化并已准备好开始工作，但是无法通过 S7-400 背板总线进行通信。	检查常规组态和数据链接组态。

故障显示	错误说明	解决方法
FAULT 快速闪烁	参数不正确或错误和/或插入了有故障的接口子模块。（子模块和接口参数不匹配）。	检查参数分配接口和/或接口子模块中的参数设置。
FAULT 亮起	没有接口参数可用，或子模块（硬件）中出现严重故障。	使用参数分配工具进行参数分配和/或检查接口子模块。

10.2.2 CP 441-2 的系统功能块的诊断消息

简介

每个系统功能块都有一个用于错误诊断的 **STATUS** 参数。无论使用哪个系统功能块，每个 **STATUS** 消息编号都始终具有相同的含义。下表说明了对 **CP** 最重要的一些 **STATUS** 消息。您会在参考手册《S7-300/400 的系统软件、系统功能和标准功能》中找到 **STATUS** 消息最新的完整描述。

SFB 的 STATUS 输出处的消息

STATUS	错误说明
0	无错误
1	CP 和 CPU 之间的通讯问题
2	否定应答，不能执行功能；例如，链接伙伴无响应，或发送了否定应答。
3	此通讯链接中的 R-ID 未知，设备不可用。
4	数据区数量或各个数据类型不匹配。
5	接收到复位请求
6	远程块处于禁用状态
7	远程伙伴处于不正确的状态。
8	访问远程对象被拒绝，服务器中发生访问错误 (GET/PUT)。
9	过速警告 (ERROR=0)：接收数据被较新的数据覆盖。
10	无法访问本地用户存储器（例如 DB 已删除）。
11	警告 (ERROR=0)：新作业未激活，因为前一作业尚未完成。
12	实例与系统调用不兼容；未调用实例，但是调用了常规 DB。
13	格式描述中出错
14	引用数据链接（应用程序相关）不存在，ID 未知。必须指定来自数据链接组态的本地 ID。
15	生成通过 ID 引用的数据链接。
16	由于资源不足，无法生成数据链接。

显示和评估 STATUS 输出

可以使用 STEP 7 变量表来显示和评估系统功能块的 STATUS 输出。

说明

使用 STATUS 作业读取 SYSTAT 区域可以提供错误和事件的详细信息；这些错误和事件是在 CP、相关 CPU 和所连接的链接伙伴之间进行通讯期间发生的。

10.2.3 读取 CP 441-2 的错误消息区域 SYSTAT

通过通讯 FB 读取

错误消息区域 SYSTAT 已由 **MODBUS 通讯 FB** 读取。

FB 将循环读取 SYSTAT 区域并将数据存储在开始于 DBW 40 的背景数据块中（另请参见“调试通讯 FB (页 59)”一章）。

说明

由于 STATUS 请求与链接上运行的其余请求是异步执行的，因此无法将带有特定 R_ID 的 SFB 分配给错误消息。这意味着，尽管 SYSTAT 可以显示数据链接上发生的错误，但是无法显示触发了错误的 SFB 调用。

10.2.4 通过 CP 441-2 的错误消息区 SYSTAT 进行诊断

简介

错误消息区 SYSTAT 是 CP 441-2 上的数据区域，CP 识别到的所有错误和事件都详细输入到此区域。SYSTAT 区由每个接口的六个事件，以及 CP 的工作状态相关的信息和 SYSTAT 区的状态组成。

SYSTAT 区的结构

CP 识别到的前 6 个错误/事件输入到 SYSTAT 区。只有删除了 SYSTAT 区的内容之后才能输入更多错误/事件。

错误/事件按下述方式输入到参数 LOCAL 中：

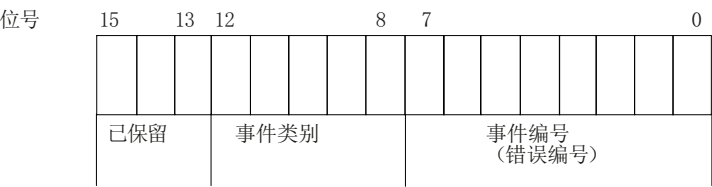
字节 0	CP 的操作状态（RUN 为 02H，故障为 05H）	
字节 1	保留	
字节 2	位 0 - F	输入到 SYSTAT 中的错误
	位 1 - U	错误溢出
	位 2 - B	断路 (BREAK)
字节 3	保留	
字节 4/5	事件 1	
字节 6/7	事件 2	
字节 8/9	事件 3	
字节 10/11	事件 4	
字节 12/13	事件 5	
字节 14/15	事件 6	

正在删除 SYSTAT 区

使用 SFB STATUS 读取 SYSTAT 区域后，将自动删除所有 SYSTAT 消息。

编号方案

SYSTAT 区中事件的编号方案结构如下：



可以在下列章节和手册“CP 441-2：点到点通信”中找到事件类别和事件编号的详细分类。

说明

与标准驱动程序相反，为了与可加载的驱动程序一起使用，对 SYSTAT 区的事件类别/事件编号作了部分修改。

下列章节描述了所有**修改的事件级别/事件号**的驱动程序特定的含义。

除非在本手册中提到的事件，否则都可以假设该事件对应标准数据链接，并在 CP 441-2 手册中进行说明。

10.3 错误/事件表

事件类别

定义了下列事件类别：

事件类别	描述	描述位置
1	CP 上的硬件错误	CP 手册
2	初始化时出错	CP 手册
3	PBK 的参数分配期间出错	CP 手册
4	CP 检测到 CP - CPU 数据传输中出错	CP 手册
5	处理 CPU 作业期间出错	CP 手册，驱动程序手册
6	处理伙伴作业期间出错	CP 手册
7	发送错误	CP 手册
8	接收错误	驱动程序手册
9	从链接伙伴接收到的错误消息帧	未使用
10	CP 在来自伙伴的应答消息帧中检测到的错误	未使用
14	可加载驱动程序的常规处理错误	驱动程序手册

10.3.1 SYSTAT 中的“CPU 作业错误”错误代码

描述

事件类别 5 (05H) “CPU 作业错误”			
事件类别/ 编号（十 六进制）	事件编号 （十进 制）	事件文本	补救措施
05 18H	24	传输期间的传输长度太大（> 4 KB），或 SEND 的传输长度太小。	检查通讯 FB，可能的话，应重新装载。

10.3.2 SYSTAT 中的“接收错误”错误代码

描述

事件类别 8 (08H) “接收错误”			
事件类别/ 编号（十 六进制）	事件编号 （十进 制）	事件文本	补救措施
08 06H	6	超过字符延迟时间 (ZVZ)	消除伙伴设备中的故障或对传输线路的干扰。
08 0CH	12	在字符中识别到传输错误（奇偶校验错误、溢出错误、停止位错误（帧））。	检查干扰是否会影响传输线路。如果需要，请更改系统结构和/或电缆铺设。 检查在 CP 和链接伙伴上是否设置了相同的协议参数，如传输速率、数据位数、奇偶校验、停止位数等。

事件类别 8 (08H) “接收错误”			
事件类别/ 编号 (十 六进制)	事件编号 (十进 制)	事件文本	补救措施
08 0DH	13	断路 (BREAK) 连接到伙伴设备的接收线路中断。	在设备之间建立连接或打开伙伴设备。 对于与 TTY 操作一起使用, 检查空闲状态的线路电流。 对于与 RS422/485 (X27) 连接一起使用, 检查并在需要时更改 2 线接收线路 R(A) 和 R(B) 的连接器引脚分配。
08 30H	48	该功能代码不允许使用广播。	对于能够使用广播功能的功能代码, 仅允许 MODBUS 主站系统使用广播功能。
08 31H	49	不允许接收功能代码。	该功能代码不能用于此驱动程序。
08 32H	50	超出最大位数或寄存器数或者访问 SIMATIC 存储区定时器或计数器时位数无法被 16 整除。	将最大位数和最大寄存器数分别限制为 2040 和 127。仅以 16 位间隔访问 SIMATIC 定时器/计数器。
08 33H	51	功能代码 FC 15/16 的位数或寄存器数与消息元素 byte_count 不匹配。	纠正位数/寄存器数或 byte_count。
08 34H	52	识别出“设置位/重设位”的位代码非法。	仅将编码 0000 十六进制或 FF00 十六进制用于 FC 05。
08 35H	53	识别出功能代码 FC 08“环路测试”的诊断子代码 (≠ 0000 十六进制) 非法。	仅将子代码 0000 十六进制用于 FC 08。
08 36H	54	CRC 16 校验和的内部产生的值与接收到的 CRC 校验和不匹配。	检查 Modbus 主站系统生成的 CRC 校验和。
08 37H	55	消息序列错误: MODBUS 主站系统会在驱动程序传送上一应答消息前发送新的请求消息。	增加 MODBUS 主站系统的从站应答消息的超时值。

10.3.3 SYSTAT 中常规处理错误的错误代码

事件类别 14 (0EH) “可加载的驱动程序 — 常规处理错误”			
事件类别/编号 (十六进制)	事件编号 (十进制)	事件文本	解决方法
0E 01H	1	在初始化驱动程序特定的 SCC 进程期间出错。	重新分配驱动程序的参数，然后重新装载。
0E 02H	2	启动驱动程序过程中出错： 激活了错误的 SCC 过程 (SCC 驱动程序)。 驱动程序无法与此 SCC 驱动程序一起运行。	重新分配驱动程序的参数，然后重新装载。
0E 03H	3	启动驱动程序过程中出错： 激活了错误的数据传送过程 (连接到 SFB)。 驱动程序无法与此数据传送过程一起运行。	重新分配驱动程序的参数，然后重新装载。
0E 04H	4	启动驱动程序过程中出错： 接口子模块非法。 驱动程序无法与参数化的接口子模块一起运行。	检查并更正参数分配。
0E 05H	5	驱动程序软件狗出错： 未插入软件狗，或插入的软件狗有故障。 驱动程序未就绪。	检查是否在 CP 中插入了驱动程序软件狗。 如果插入的软件狗有故障，请使用正确的软件狗替换。
0E 06H	6	驱动程序软件狗出错： 软件狗无有效内容。 驱动程序未就绪。	从为您提供此驱动程序的西门子办事处获取正确的软件狗。
:	:		
0E 10H	16	内部错误程序： 在自动发送设备中跳转失效 (default branch)。	重新启动 CP (Power_On)
0E 11H	17	内部错误程序： 在自动接收设备中跳转失效 (default branch)。	重新启动 CP (Power_On)

事件类别 14 (0EH) “可加载的驱动程序 — 常规处理错误”			
事件类别/编号 (十六进制)	事件编号 (十进制)	事件文本	解决方法
0E 12H	18	主动自动化设备发生内部错误: 跳转失效 (default branch)	重新启动 CP (Power_On)
0E 13H	19	被动自动化设备发生内部错误: 跳转失效 (default branch)	重新启动 CP (Power_On)

事件类别 14 (0EH) “可加载的驱动程序 - 常规处理错误 <参数分配>”			
事件类别/编号 (十六进制)	事件编号 (十进制)	事件文本	解决方法
0E 20H	32	对于此数据链接, 数据位数必须设置为 8。 驱动程序未就绪。	更正驱动程序的参数分配。
0E 21H	33	字符延迟时间的倍增因子设置不在 1 到 10 的范围内。 驱动程序使用标准设置 1 工作。	更正驱动程序的参数分配。
0E 22H	34	驱动程序的操作模式设置非法。 必须指定是“常规”还是“干扰抑制”。 驱动程序未就绪。	更正驱动程序的参数分配。
0E 23H	35	为从站地址设置了非法值。 不允许从站地址为 0。 驱动程序未就绪。	更正驱动程序的参数分配。
0E 24H	36	为写访问设置了非法限制。 驱动程序未就绪。	更正驱动程序的参数分配。
0E 25H	37	为“FC 01、05、15 的 Modbus 寻址转换”区域的输入所设置的“从/到”组合非法。 (相关区域包括: 存储器位、输出、定时器、计数器)。 驱动程序未就绪。	更正驱动程序的参数分配。

事件类别 14 (0EH) “可加载的驱动程序 - 常规处理错误 <参数分配>”			
事件类别/编号 (十六进制)	事件编号 (十进制)	事件文本	解决方法
0E 26H	38	为 “FC 02 的 Modbus 寻址转换”区域的输入所设置的“从/到”组合非法。 (相关区域包括: 存储器位、输入)。 驱动程序未就绪。	更正驱动程序的参数分配。
0E 27H	39	为 “FC 01、05、15 的 Modbus 寻址转换”区域的输入所设置的“从/到”组合出现重叠。 (相关区域包括: 存储器位、输出、定时器、计数器)。 驱动程序未就绪。	更正驱动程序的参数分配。
0E 28H	40	为 “FC 02 的 Modbus 寻址转换”区域的输入所设置的“从/到”组合出现重叠。 (相关区域包括: 存储器位、输入)。 驱动程序未就绪。	更正驱动程序的参数分配。
0E 2EH	46	读取接口参数文件时发生错误。 驱动程序未就绪。	重新启动 CP (Power_On)。

事件类别 14 (0EH) “可加载的驱动程序 - 常规处理错误 <CPU-CP>”			
事件类别/编号 (十六进制)	事件编号 (十进制)	事件文本	解决方法
0E 30H	48	数据传送到 CPU 期间发生内部错误： 意外的否定应答。	如果只是间断性地发生，可以忽略。
0E 31H	49	数据传送到 CPU 的过程中超时：	检查 CP-CPU 接口。
0E 32H	50	使用 RCV 将数据传送到 CPU 的过程中出错： 准确的故障原因（详细错误）在 SYSTAT 中此条目之前。	检查 CP-CPU 接口。
0E 33H	51	数据传送到 CPU 的过程中发生内部错误： 自动化设备的状态非法	检查 CP-CPU 接口。
:	:		
0E 38H	56	使用功能代码 FC 01 或 FC 02 访问 SIMATIC“存储器位、输出、定时器、计数器、输入”区域之一时发生错误 例如，输入不存在，或读取尝试超出了范围限制。	检查所寻址的 SIMATIC 区域是否存在，以及所做的访问尝试是否超出范围限制。
0E 39H	57	使用功能代码 FC 03、04、06、16 访问 SIMATIC“数据块”区域时发生错误： 数据块不存在或太短。	检查所寻址的数据块是否存在，以及该数据块是否足够长。
0E 3AH	58	使用功能代码 FC 05、15 执行写入作业时发生错误： MODBUS FB 的背景数据块不存在或太短。	检查在 MODBUS 通信 FB 上参数化的背景数据块是否存在，及其是否足够长。
0E 3BH	59	Modbus 通信 FB 在执行写入作业的过程中超时。	检查数据链接的组态和 CP-CPU 接口 (SFB SEND)： 如果可能，请重新装载 MODBUS 通信 FB。
0E 3CH	60	此驱动程序的作业非法。	仅允许 SFB SEND 和 STATUS（仅适用于 CP 441-2）。

事件类别 14 (0EH) “可加载的驱动程序 - 常规处理错误 <接收评估>”			
事件类别/编号 (十六进制)	事件编号 (十进制)	事件文本	解决方法
0E 50H	80	对于面向世界的 SIMATIC 定时器/计数器区， Modbus 地址产生的其他位数 $\neq 0$ 。	仅使用可产生有效位数的 MODBUS 地址。
0E 51H	81	接收到的 Modbus 地址超出了参数化的“从/到”区域范围。（请参见“为可加载驱动分配参数 (页 37)”部分）	仅将先前在参数分配过程中定义的请求消息帧中的地址用作地址规范。
0E 52H	82	- 在 Modbus 主站系统进行的访问尝试过程中，超出了 SIMATIC 范围限制： 产生的 DB 号 < 1 - 或对尚未启用的区域进行写访问（参数分配）， - 或对通信 FB 的背景数据块进行写访问。	限制对有效 SIMATIC 存储区的访问范围。
0E 53H	83	在 Modbus 主站系统进行的访问尝试过程中，超出了 SIMATIC 范围限制 例如，生成产生的 DB 号 (> 65535) 时溢出。	限制对有效 SIMATIC 存储区的访问范围。
0E 54H	84	访问超出了参数化的范围限制，或超出了 SIMATIC 范围限制。	限制对有效 SIMATIC 存储区的访问范围。
0E 55H	85	不允许对此 SIMATIC 存储区进行写访问。	仅对 SIMATIC 数据区存储器位和输出区执行写访问。
0E 56H	86	不能进行数据链接操作，因为通信 FB 没有运行。	在 STEP 7 用户程序中对 MODBUS 通信 FB 进行循环调用。 如果需要，重新初始化通信 FB 。
0E 57H	87	在处理 Modbus 功能代码的过程中通信 FB 发生错误。	如“通讯 FB 的诊断 (页 141)”部分所述分析确切原因。

通讯 FB 的诊断

诊断功能

MODBUS 通讯 FB 具有以下两个**输出参数**，可以指示发生的错误：

- 参数 “ERROR_NR”
- 参数 “ERROR_INFO”

ERROR_NR, ERROR_INFO

在 **ERROR_NR** 输出中指示发生的错误。

有关 ERROR_NR 中错误的更多详细信息显示在输出 **ERROR_INFO** 中。

删除错误

在 CP_START 的某个上升沿上删除错误。

如果需要，用户可随时删除错误显示。

11.1 通过参数 *ERROR_NR* 和 *ERROR_INFO* 进行诊断

ERROR_No 1...9

初始化 FB 和 CP 时出错

错误号 1 到 9 指示初始化出错。参数 *CP_START_ERROR* 为 1。

MODBUS 无法与主站系统进行通讯。

ERROR_No 10...19

处理功能代码期间出错

错误号 10 到 19 指示处理功能代码期间出现的某个错误。CP 向通讯 FB 发送了一个非法的处理作业。

系统还将该错误报告给了驱动程序。

对后续处理作业仍将继续进行处理。

ERROR_No 90...99

其它错误

发生了一个处理错误。

系统未将该错误报告给驱动程序。

对后续处理作业仍将继续进行处理。

11.1.1 初始化时出错

ERROR_No (十进制)	CPx41	ERROR_INFO	错误文本	解决方法
0	CPx41	0	无错误	
1	CPx41	SFC 51- >RET_VAL	使用 SFC 51 读取 SZL 时 出错。	分析 ERROR_INFO 中的 RET_VAL；消除出错原因。
2	CPx41		初始化 CP 时超时，或 初始化 CP 时出错	检查协议“MODBUS 从站”是否 在 CP 接口上分配了参数。 分析 ERROR_INFO。
	CP 341	FB 8->STATUS	P_SND_RK 作业中有错 误。	检查通信 FB 中的 LADDR 模块地 址。
	CP 441	SFB 12- >STATUS	BSEND 作业中有错误。	检查通信 FB 中的连接 ID。

11.1.2 在处理功能代码的过程中出错

ERROR_No (十进制)	ERROR_INFO	错误文本	补救措施
10	处理代码	驱动程序传送到通讯 FB 的处理功能非法。	重新启动 CP (Power_On)
11	起始地址	驱动程序传送到通讯 FB 的起始地址非法。	检查 MODBUS 主站系统的 MODBUS 地址。
12	寄存器数	驱动程序传送到通讯 FB 的寄存器数非法： 寄存器数 = 0。	检查 MODBUS 主站系统的寄存器数，如果需要，请重新启动 CP (Power_On)。
13	寄存器数	驱动程序传送到通讯 FB 的寄存器数非法： 寄存器数 = 128。	检查 MODBUS 主站系统的寄存器数，如果需要，请重新启动 CP (Power_On)。
14	存储器位 M — 结束地址	对 SIMATIC 存储器位存储区的尝试访问次数超出范围限制。 注意： SIMATIC CPU 的范围长度取决于 CPU 类型。	减少主站系统中 MODBUS 起始地址长度和/或访问长度。
15	输出 Q — 结束地址	对 SIMATIC 输出存储区的尝试访问次数超出范围限制。 注意： SIMATIC CPU 的范围长度取决于 CPU 类型。	减少主站系统中 MODBUS 起始地址长度和/或访问长度。
16	定时器 T — 结束地址	对 SIMATIC 定时器存储区的尝试访问次数超出范围限制。 注意： SIMATIC CPU 的范围长度取决于 CPU 类型。	减少主站系统中 MODBUS 起始地址长度和/或访问长度。
17	计数器 C — 结束地址	对 SIMATIC 计数器存储区的尝试访问次数超出范围限制。 注意： SIMATIC CPU 的范围长度取决于 CPU 类型。	减少主站系统中 MODBUS 起始地址长度和/或访问长度。

11.1 通过参数 *ERROR_NR* 和 *ERROR_INFO* 进行诊断

ERROR_No (十进制)	ERROR_INFO	错误文本	补救措施
18	0	驱动程序向通讯 FB 传送的 SIMATIC 存储区非法。	如果需要，重新启动 CP (Power_On)。
19		访问 SIMATIC I/O 的过程中出错。	检查所需的 I/O 是否存在并且无错误。

11.1.3 其他错误

ERROR_No (十进制)	CPx41	ERROR_INFO	错误文本	解决方法
90	CP 341	FB 8->STATUS	使用 FB 8 P_SND_RK 向驱动程序发送确认消息时出错。	分析 STATUS 信息。
	CP 441	SFB 12 "STATUS"	使用 SFB 12 BSEND 向驱动程序发送确认消息时出错。	
91	CPx41	SFB 22->STATUS	使用 SFB 22 (STATUS) 读取 SYSTAT 时出错。	分析 STATUS 信息。
92	CPx41	FB 7->STATUS	使用 FB7 (RCV_RK) 执行 RECEIVE/FETCH 调用时出错	分析 FB7-STATUS

技术数据

A.1 技术数据

传输时间

下表包含了测量到的不同功能块的传输时间。

使用 S7-300 可编程控制器（带有 CPU 315-2 DP [6ES7315-2AF01-0AB0]、CP 341）以及作为伙伴设备的 S7-400 可编程控制器（带有 CPU 414 [6ES7414- 1XG01-0AB0]、CP 441-2）测量的时间。测量到下列时间：

- 处理时间（从在用户程序中初始化作业开始算起），包括主站上的处理时间在内，
- 作业通过串口传送到伙伴设备所需的时间
- 在从站上进行处理使用的时间，
- 在串口上传送确认所需要的时间。

必须将四个时间加在一起，才能计算出整个传输所使用的时间。

如果使用其他主站或从站（作为伙伴设备），则必须使用所使用的主站或从站的相应时间，而不是表格中的时间。作业和确认的时间不变；它们只取决于使用的传输速率。

从站为 CP 341

功能代码 1（读取）— 读取线圈（输出）状态（单位为毫秒的时间）

传输速率（波特）	300			
用户数据	主站 CP 441-2	作业	从站 CP 341	确认
1 个字节	229	257	179	184
10 个字节	229	257	179	514
20 个字节	229	257	180	882
50 个字节	232	257	182	1986
100 个字节	236	257	192	3824
200 个字节	243	257	208	7501
255 个字节	251	257	214	9487

传输速率（波特）	9600			
用户数据	主站 CP 441-2	作业	从站 CP 341	确认
1 个字节	74	8	18	6
10 个字节	75	8	19	16
20 个字节	77	8	19	27
50 个字节	83	8	62	62
100 个字节	90	8	119	119
200 个字节	92	8	235	235
255 个字节	95	8	296	296

传输速率（波特）	76800			
	主站 CP 441-2	作业	从站 CP 341	确认
用户数据				
1 个字节	73	1	13	1
10 个字节	74	1	13	2
20 个字节	76	1	13	3
50 个字节	86	1	20	8
100 个字节	93	1	29	15
200 个字节	95	1	45	29
255 个字节	97	1	50	37

从站为 CP 341

功能代码 15（写入）— 写多个线圈（单位为毫秒的时间）

传输速率（波特）	300			
	主站 CP 441-2	作业	从站 CP 341	确认
用户数据				
1 个字节	205	331	199	257
10 个字节	206	662	200	257
20 个字节	206	1028	201	257
50 个字节	208	2132	212	257
100 个字节	211	3971	223	257
200 个字节	217	7648	238	257
255 个字节	221	9634	243	257

传输速率（波特）	9600			
用户数据	主站 CP 441-2	作业	从站 CP 341	确认
1 个字节	48	10	41	8
10 个字节	48	20	41	8
20 个字节	50	32	43	8
50 个字节	52	67	48	8
100 个字节	55	124	56	8
200 个字节	63	239	74	8
255 个字节	67	301	88	8

传输速率（波特）	76800			
用户数据	主站 CP 441-2	作业	从站 CP 341	确认
1 个字节	58	1	40	1
10 个字节	61	3	43	1
20 个字节	62	4	43	1
50 个字节	63	8	44	1
100 个字节	64	15	50	1
200 个字节	66	30	69	1
255 个字节	68	38	85	1

从站为 CP 441-2

功能代码 1（读取）— 读取线圈（输出）状态（单位为毫秒的时间）

传输速率（波特）	300			
用户数据	主站 CP 341	作业	从站 CP 441-2	确认
1 个字节	236	257	188	184
10 个字节	236	257	190	515
20 个字节	238	257	190	882
50 个字节	244	257	193	1986
100 个字节	280	257	199	3824
200 个字节	286	257	207	7502
255 个字节	288	257	216	9487

传输速率（波特）	9600			
用户数据	主站 CP 341	作业	从站 CP 441-2	确认
1 个字节	33	8	40	6
10 个字节	33	8	43	16
20 个字节	35	8	44	28
50 个字节	42	8	45	62
100 个字节	56	8	56	120
200 个字节	75	8	64	235
255 个字节	82	8	77	296

传输速率（波特）	76800			
用户数据	主站 CP 341	作业	从站 CP 441-2	确认
1 个字节	35	1	23	1
10 个字节	36	1	25	2
20 个字节	37	1	26	3
50 个字节	46	1	27	8
100 个字节	61	1	30	15
200 个字节	82	1	39	29
255 个字节	92	1	48	37

从站为 CP 441

功能代码 15（写入）— 写多个线圈（单位为毫秒的时间）

传输速率（波特）	300			
用户数据	主站 CP 341	作业	从站 CP 441-2	确认
1 个字节	225	331	223	257
10 个字节	227	662	224	257
20 个字节	227	1030	228	257
50 个字节	227	2132	232	257
100 个字节	229	3971	236	257
200 个字节	230	7648	243	257
255 个字节	237	9634	255	257

传输速率（波特）	9600			
用户数据	主站 CP 341	作业	从站 CP 441-2	确认
1 个字节	64	11	62	8
10 个字节	64	21	63	8
20 个字节	69	32	64	8
50 个字节	69	67	68	8
100 个字节	72	124	70	8
200 个字节	75	239	76	8
255 个字节	75	301	86	8

传输速率（波特）	76800			
用户数据	主站 CP 341	作业	从站 CP 441-2	确认
1 个字节	60	1	56	1
10 个字节	60	3	58	1
20 个字节	62	4	58	1
50 个字节	64	9	60	1
100 个字节	67	16	67	1
200 个字节	72	30	77	1
255 个字节	77	38	84	1

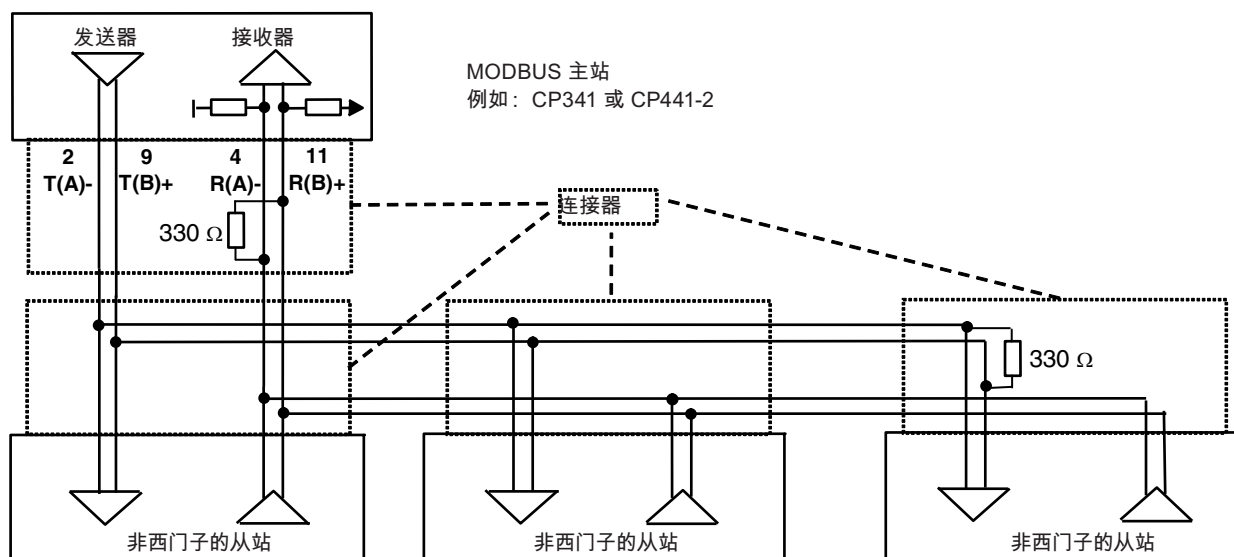
存储器要求

下表显示了 CP 341 (FB 80) 和 CP 441 (FB 180) 的功能块的存储器要求（单位为字节）。FB 7 和 8 的存储器要求可以在 CP 341 手册中找到。

块	名称	版本	装载存储器	主存储器	本地数据
FB 80	MODB_341	1.0	2770	2034	104
FB 180	MODB_441	1.0	2982	2360	102

多点接线图

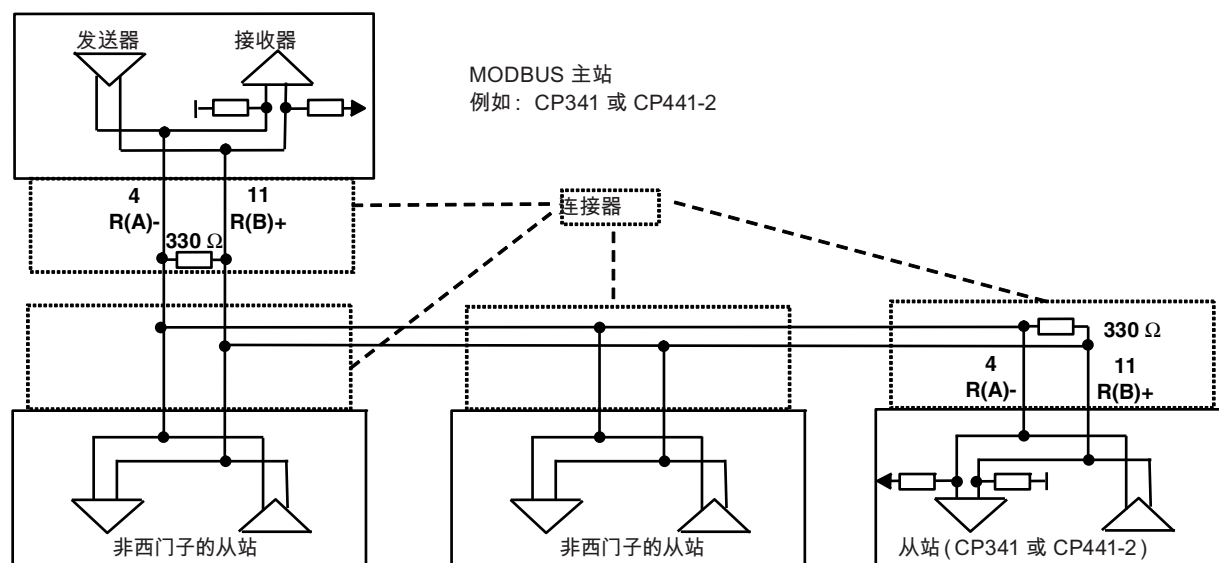
RS422 多点接线图（MODBUS 多点）



注意

在 **RS422** 模式下，CP 341 和 CP 441-2 只能用作“主站”，原因是它们无法将发送线路切换成“三态”。

RS485 多点接线图 (MODBUS 多点)



以下内容对于两个模块都适用：

- 两边的 GND (CP341 / CP441-2 的第 8 脚) 都必须连接。
- 无论在什么位置都必须安装外壳屏蔽。
- 节点序列的最后一个接收器的连接器要焊接一个大约为 330 Ω 的终端电阻。
- 推荐的电缆类型：LIYCY 3 x 2 x 0.14；R(A)/R(B) 或 T(A)/T(B) 类型的双绞线
- 不允许使用“短接线”进行接线。

参考资料

C.1 参考资料

MODBUS 协议

- /1/ Gould Modbus 协议
参考指导
PI-MBUS-300 Rev B
GOULD 电子设备

词汇表

CPU

S7 可编程控制器的中央处理单元具有控制和算法单元、存储器、系统程序以及连接 I/O 模块的接口。

CPU 的操作系统

CPU 操作系统用于组织 CPU 的所有功能和操作，与特定控制任务无关。

CRC

循环冗余校验 = 保证能准确识别错误的校验和。

STEP 7

STEP 7 是 SIMATIC S7 的编程软件。

上传

将装载对象（例如代码块）从中央处理单元的装载存储器上传到编程设备中。

下载

将装载对象（例如代码块）从编程设备下载到中央处理单元 (CPU) 的装载存储器中。

中断

中断是一个术语，通过外部报警指示可编程控制器的处理器中的程序处理中断

主存储器

主存储器是 CPU 中的 RAM 存储器，在运行用户程序时处理器将访问主存储器。

功能块 (FB)

功能块是用户程序的组成部分，并且是符合 IEC 标准的“带存储器的块”。功能块的存储器是已分配好的数据块，即“背景数据块”。可以为功能块分配参数，也就是说可以带或不带参数使用。

协议

数据交换涉及的通信伙伴必须遵守用于处理和执行数据交换的固定规则。这些规则称为协议。

参数

参数是

- STEP 7 代码块的变量，
- 该变量用于指定模块的动作，
交货时，每个模块均具有合适的缺省设置，并且该缺省设置可以通过硬件配置改变。

有两种类型的参数：静态参数和动态参数

参数分配

参数分配用于设置模块的行为。

参数分配工具 CP：为点对点连接分配参数

CP：为点对点连接分配参数 工具用于为通信处理器的接口子模块分配参数，并且用于设置驱动程序特定参数。

对于每个可加载驱动程序，都将扩展其标准范围。

变量

变量是一个内容可变的数据元素，可以在 STEP 7 用户程序中使用。变量由操作数（例如 M 3.1）和数据类型（例如 BOOL）组成，并且可使用符号（例如 BELT_ON）表示。

可编程控制器

可编程控制器是一种电子控制设备，它由至少一个 CPU、多种输入和输出模块以及操作员监控设备组成。

启动

从 **STOP** 模式转换到 **RUN** 模式时，将执行 **RESTART** 模式。这可以由下列事件触发：

- 通过激活操作模式开关
- 在电源接通后
- 通过编程设备中的操作员输入

可区分为冷重新启动、重新启动和热重新启动。

在线/离线

当处于在线状态时，**PLC** 和编程设备之间有数据连接；当处于离线状态时，它们之间将没有数据连接。

在线帮助

在使用编程软件的过程中，**STEP 7** 允许您在屏幕上显示上下文相关的帮助文本。

块

块是用户程序的组成元素，可根据其功能、结构或者用途进行定义。**STEP 7** 有如下述块：

- 代码块 (**FB**、**FC**、**OB**、**SFB**、**SFC**)
- 数据块 (**DB**、**SDB**)
- 用户自定义数据类型 (**UDT**)。

块参数

块参数用于在多用块内占位，在调用相关块时这些参数会得到更新值。

块调用

当程序处理跳转到调用块时，发生了块调用。

工具

工具是用于组态和编程的软件工具。

循环时间

循环时间是 CPU 处理一次用户程序所需要的时间。

循环程序处理

在循环程序处理中，用户程序运行在一个程序循环中，或者不断重复的循环中。

接口子模块

CP 441-2 接口模块对信号进行物理转换。通过改变插入式接口模块，可以使通信处理器与通信伙伴兼容。

操作数

操作数是 STEP 7 指令的组成部分，用来指示处理器要执行的操作。操作数可以使用绝对寻址和符号寻址。

操作模式

SIMATIC S7 可编程控制器有三种不同的操作模式：

STOP、RESTART 和 RUN。在不同的操作模式下，CPU 的功能不同。

数据块 (DB)

数据块是包含用户程序所用数据和参数的块。与其它块不同的是，数据块不包含任何指令。它们可细分为全局数据块和背景数据块。

数据块中包含的数据可以通过绝对地址或符号地址进行访问。复杂数据可以通过结构化的形式存储。

数据类型

数据类型允许用户定义如何使用用户程序中变量或者常量的值。SIMATIC S7 允许使用两种符合 IEC 1131-3 的数据类型：

- 基本数据类型
- 结构化数据类型

数据链接的组态（仅限 CP 441-2）

数据链接组态指的是系统功能块中 `connection_ID` 的规范。连接 ID 使得系统功能块可以在两个通信终端之间进行通信。

机架

机架是包含模块插槽的导轨。

模块

模块是用于自动化系统的可插拔 PCB。

模块参数

模块参数是可用于设置模块功能的值。可区分为静态模块参数和动态模块参数。

点对点连接

在点对点通信中，通信处理器建立了可编程逻辑控制器和对等通信伙伴之间的接口。

用户程序

用户程序包含用于处理信号（控制系统或过程所使用的信号）的所有指令和声明。在 SIMATIC S7 中，将用户程序结构化，并划分为较小的单元（块）。

硬件

硬件是 PLC 的所有物理和技术设备的通称。

程序

程序指根据特定协议进行数据传输。

系统功能 (SFC)

系统功能 (SFC) 是不带存储器的功能，它集成在 S7-CPU 的操作系统中，并且需要时用户程序可以像调用功能 (FC) 一样调用它。

系统功能块 (SFB)

系统功能块 (SFB) 是带有存储器的功能块，它集成在 S7-CPU 的操作系统中，并且需要时用户程序可以像调用功能块 (FB) 一样调用它。

系统块

系统块和其它块的不同之处在于它们已经集成到 S7-300/S7-400 系统中了，并且可以作为已定义的系统功能使用。系统块可以再细分为系统数据块、系统功能和系统功能块。

组态

组态是指组合各模块以形成一个 PLC。

缺省设置

缺省设置是一个基本设置，在没有指定其它值之前它将一直使用。

背景数据块

背景数据块存储功能块的形式参数和静态数据。背景数据块可以分配给 FB 调用或者分配给功能块的调用层级。

诊断事件

诊断事件将在 CPU 的诊断缓冲区中触发一个条目。诊断事件可区分为以下几种：

- 模块错误
- 过程接线错误
- CPU 的系统错误
- CPU 操作模式转换
- 用户程序错误
- 用户自定义诊断事件

诊断功能

诊断功能覆盖整个系统诊断，包括 PLC 中错误的探测、解释以及报告。

诊断缓冲区

诊断缓冲区是 CPU 中带电池备份的存储区，举个例子，该缓冲区具有环形缓冲区的结构。诊断事件按发生顺序进行存储。

软件

软件是计算机系统中使用的所有程序的总称。这包括操作系统和用户程序。

过程映像

过程映像是可编程控制器中的一个特殊存储区。循环程序开始时，将输入模块的信号状态传输到过程输入映像。循环程序结束时，将过程输出映像作为信号状态传送到输出模块。

通信处理器

通信处理器是用于通信任务（例如，网络或点对点连接）的可编程模块。

索引

C

CRC, 77

S

SYSTAT, 123, 130, 131

事件类别, 132

事件编号, 132

三划

广播, 76

四划

为 CPU 分配参数, 57

从站地址, 39, 75

计数器, 22

五划

功能代码, 11, 22, 76

FC 01, 41

FC 03, 46

FC 04, 47

FC 05, 41

FC 06, 46

FC 15, 41

FC -15, 14, 49

FC -15, 14, 49

FC 16, 46

FC -16, 14, 49

FC -16, 14, 49

FC-01, 14

FC-02, 14

FC-03, 14

FC-04, 14

FC-05, 14, 49

FC-06, 14, 49

FC-08, 111

功能块, 11, 19, 21, 60, 61, 130

六划

传输时间, 147

传输速率, 38

传输错误, 40, 52

地址表示, 15

多点连接, 12

字符延迟时间 CDT, 40, 77

存储器位, 22

异常代码, 78

七划

启动特性, 56

装载过程, 57

诊断, 123, 141

连接 ID, 36, 71

八划

参数分配, 33, 34
奇偶校验, 38
定时器, 22
组态数据链接, 36
软件狗, 13, 17

九划

卸载, 30
响应时间, 67
指示灯 (LED), 123
背景数据块, 20, 59, 64

十划

消息帧结构, 75

十一划

接口模块
 RS 232C, 12
 TTY, 12
 X27, 12

十二划

装载存储器, 13

十三划

数据一致性, 70, 73
数据块, 22
输入, 22
输出, 22

十六划

操作模式, 39