SIEMENS

SIMATIC

S7-1500、ET 200MP、ET 200SP、ET 200AL、ET 200pro, ET 200eco PN 通信

功能手册

11/2024
A5E03735819-AM

转下页

续

S7-1500、ET 200MP、ET 200SP、ET 200AL、ET 200pro, ET 200eco PN 通信

功能手册

法律资讯

警告提示系统

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

⚠ 危险

表示如果不采取相应的小心措施，**将会**导致死亡或者严重的人身伤害。

⚠ 警告

表示如果不采取相应的小心措施，**可能**导致死亡或者严重的人身伤害。

⚠ 小心

表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

注意

表示如果不采取相应的小心措施，可能导致财产损失。

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

合格的专业人员

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

按规定使用 Siemens 产品

请注意下列说明：

⚠ 警告

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

商标

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

责任免除

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

Siemens Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
德国

A5E03735819-AM
Ⓟ 10/2024 本公司保留更改的权利

Copyright © Siemens 2013 - 2024.
保留所有权利

目录

简介

1

本文档的用途

在本功能手册中，简要介绍了
SIMATIC S7-1500、ET 200MP、ET 200SP、ET 200AL、ET 200pro 和 SIMATIC Drive Controller 系统中的通信选件、CPU、通信模块、处理器和 PC 系统。本功能手册主要介绍了基于连接的异步通信。

本文档中包含以下内容：

- 通信服务概述
- 通信服务的属性
- 设置通信服务的用户操作概述

所需基本知识

要理解本功能手册中的内容，需要具备以下知识：

- 自动化技术的基本知识
- 工业自动化系统 SIMATIC 的知识
- 有关如何使用 STEP 7 (TIA Portal) 的基本知识

本文档的适用范围

本文档是 SIMATIC S7-1500、ET 200MP、ET 200SP、ET 200AL 和 ET 200pro 系统中所有产品的基础性文档。产品文档基于本文档。

《通信功能手册》版本 11/2024 与版本 11/2023 相比的新增内容

新增内容	客户收益	信息出处
以太网通信接口模块中使用的通信协议和端口号表	有关接口模块使用的协议和端口的信息。可即刻了解应用的默认设置。用户可以有针对性地调整与应用相关的设置。	"以太网通信的通信协议和端口号 (页 33)"部分
OPC UA：对于自固件版本 V4.0 起的以下 S7-1500 CPU，增加了组态限制 • CPU 151x(F)-3 PN • CPU 151xT(F)-3 PN	自固件版本 V4.0 起的 S7-1500 CPU 151x(F)-3 PN 和 151xT(F)-3 PN 允许创建大量服务器接口节点以及大量服务器方法。 此外，读取、写入和订阅的性能也得到了改进。	相应 CPU 的设备手册

新增内容	客户收益	信息出处
OPC UA：同时接收多种语言的消息	例如，对于 S7-1500 CPU，自固件版本 V4.0 起的 OPC UA 客户端可以选择一种可用的消息语言。 此外，一个 OPC UA 客户端可同时请求全部三种激活项目语言的消息文本。例如，在一个中央服务器上收集消息，以便掌握不同语言的人员可对这些消息进行评估。	“同时接收多种语言的报警 (页 298)”部分
OPC UA：自固件版本 V4.0 起 S7-1500 CPU 的基于角色的安全性	通过自固件版本 V4.0 起的 S7-1500 CPU 实施基于角色的概念，OPC UA 服务器可以详细管理特定 OPC UA 客户端对地址空间的访问。	“在 OPC UA 中实现基于角色的安全性 (页 197)”部分
OPC UA：S7-1500 的“传送订阅”功能的支持	对于自版本 V3.1.4 起的 S7-1500 CPU，提供“传送订阅”功能，以支持在不同 OPC UA 客户端上分配负载的应用程序。通过这种方式， 订阅可传送到另一个 OPC UA 客户端。无需组态功能。	“使用 TransferSubscription 服务 (页 230)”部分

与 11/2022 版相比，《通信功能手册》版本 11/2023 中新增的内容

新增内容	客户收益	信息出处
“工业网络安全”部分	由于机器和工厂的数字化和网络化程度在不断提高，网络攻击的风险也在不断增加。因此，采取适当的保护措施势在必行，重要的基础设施尤为如此。 此部分包含以下信息： • 工业网络安全主题的基本信息 • 保护个别组件和整个系统免受篡改和不必要访问的建议措施。	工业网络安全 (页 24)
修订了以太网通信中使用的通信协议和端口号表	更新了有关使用的协议和端口的信息。可即刻了解应用的默认设置。用户可以有针对性地调整与应用相关的设置。	以太网通信的通信协议和端口号 (页 33)
更新了有关 CPU 和 HMI 连接资源的信息。	更新了有关以下连接资源的信息： • 某些 CPU 型号所支持的连接资源最大数量 • 不同 HMI 设备占用的连接资源的最大数	连接资源 (页 392)
使用通信处理器扩展冗余系统	自 STEP 7 V19 起，可使用 CP 1543-1 通信处理器扩展 S7-1500R/H 冗余系统（自固件版本 V3.1 起）。	通信处理器的系统 IP 地址 (页 413)。
与冗余系统进行开放式用户安全通信	* 自 STEP 7 V19 起，S7-1500R/H 冗余系统（固件版本 V3.1 及更高版本）还支持开放式用户安全通信。	与冗余系统 S7-1500R/H 进行开放式用户通信 (页 423)

与 05/2021 版相比，《通信功能手册》版本 11/2022 中新增的内容

新增内容	客户收益	信息出处
修订了以太网通信中使用的通信协议和端口号表	更新了有关使用的协议和端口的信息。可即刻了解应用的默认设置。用户可以有针对性地调整与应用相关的设置。	以太网通信的通信协议和端口号 (页 33)
激活/取消激活 SNMP	根据 S7-1500 CPU 的固件版本，在默认设置中激活或禁用 SNMP。可根据需要更改默认设置。	SNMP (页 108)
修订了基于 IP 的应用的虚拟接口	对于固件版本为 V3.0 或更高版本的 CP 1543-1，可使用内部 CP 防火墙。防火墙用于确保通过虚拟接口传输的数据流量的安全。	基于 IP 的应用程序的虚拟接口 (页 388)
OPC UA server:读取自身地址空间的诊断状态	通过使用 OPC UA 指令进行读取（“OPC_UA_ReadList”），可访问 OPC UA 服务器的自有命名空间。这样便可读取自带 OPC UA 服务器的状态，还可以读取 OPC UA 客户端连接、会话以及订阅的状态，并可在用户程序中对其进行响应。例如，这能够快速检测连接问题，并提高工厂可用性。	在程序中运行 OPC UA 服务器诊断 (页 308)
OPC UA server:节点源时间的时间戳	通过使用 OPC UA 指令进行写入（“OPC_UA_WriteList”），可更改“SourceTimestamp”以及 OPC UA 变量（节点）的状态代码。自 V18 起，可通过这种方式区分“源”和“服务器”时间。	OPC UA 服务器的客户端访问和本地访问 (页 209)
OPC UA GDS 机制：现在也可用于 Web 服务器证书	现在，HTTPS 通信的 Web 服务器证书也可通过 OPC UA GDS 机制进行管理，无需单独下载硬件配置。	证书管理的必备知识 (页 56) 通过 GDS 实现自动化证书管理 (页 181)

与 11/2019 版相比，《通信功能手册》版本 05/2021 中新增的内容

新增内容	客户收益	信息出处
改进了 SIMATIC PG/HMI 通信的安全性	• 允许根据单独证书唯一标识每个 PLC • 通过加密通信提供额外的保密性 • 通过单独密码保护组态数据	PG/HMI 间安全通信 (页 97)
新 PLC 安全机制的安全向导	• 一次操作即可快速、轻松地组态 PLC 的新安全机制 • 支持信息可为自己的应用选择合适的设置	保护机密的组态数据 (页 68)
通过 OPC UA 进行证书管理 全球发现服务器 (GDS)	• 运行期间进行证书更新 • 支持 CRL • 证书管理的访问保护	通过全球发现服务器 (GDS) 实现证书管理 (页 181)
将 CPU 报警传送到 OPC UA 客户端	• 利用订阅功能，客户端可订阅来自 CPU 的 OPC UA 服务器的 CPU 报警作为“报警和条件”。 • 包含相关值的程序消息由 OPC UA 服务器提供 • 待确认的报警可通过 OPC UA 客户端进行确认（可禁用） • 报警突发显示为“过载”，可通过刷新方法重新加载客户端	提供 OPC UA 服务器报警 (页 284)

新增内容	客户收益	信息出处
通过 DHCP 动态分配网络组态	使用以下功能在 IT 管理型网络中部署 CPU： • 无需额外手动组态网络接口即可将 CPU 连接到现有网络 • 根据 RFC 2131 向 DHCPv4 服务器请求 CPU 网络参数（IP 地址和子网掩码、默认 IP 路由器地址和其它可选网络参数，如 DNS 和 NTP 服务器地址）	通过 DHCP 寻址 (页 352)
使用 DNS 进行基于名称的寻址	• 可通过 DHCP 从 CPU 获取 DNS 服务器地址 • 对于通过 OPC UA 或（安全）OUC 实现的应用，CPU 可从 DHCP 服务器获取主机名和域名 • CPU 可将已组态的主机名或域名传送到与 DNS 服务器关联的 DHCP 服务器，以实现动态匹配（动态 DNS） • CPU 的 NTP 客户端可通过名称对 NTP 服务器进行寻址 • 可通过新增的“CommConfig”指令写入 IP 地址参数、DNS 服务器、主机名和域名等网络参数	DHCP 与 DNS (页 360)

与 10/2018 版相比，《通信功能手册》版本 11/2019 中新增的内容

新增内容	客户收益	信息出处
IP 转发	轻松实现控制级到现场级的访问，以便对设备进行组态和参数分配，例如通过 PDM 或 Web 浏览器。	IP 转发 (页 380)
OPC UA 服务器扩展	对于 V2.8 及更高固件版本的 S7-1500 CPU 以及 TIA Portal 版本 V16，通过对应的运行系统许可证可以有效利用集成的 OPC UA 服务器的以下扩展功能： • 改进的诊断功能：OPC UA 服务器通过诊断缓冲区中的报警、TIA Portal“在线和诊断”(Online & Diagnostics) 区域中的 OPC UA 类别以及改进的连接资源显示接收关于 OPC UA 服务器状态的信息。 • 下载特性：在 RUN 模式下，仅当新下载的数据会影响 OPC UA 服务器数据管理的情况下，OPC UA 服务器才会在下载期间从 TIA Portal 执行重启。 • 服务器接口建模：现在可以在 TIA Portal 中为服务器接口建模或导入 OPC UA 配套规范，并将其映射到 PLC 数据管理。	“OPC UA 通信 (页 155)”部分

与 12/2017 版相比，《通信功能手册》版本 10/2018 中新增的内容

新增内容	客户收益	信息出处
与 S7-1500R/H 冗余系统的通信说明	收到有关与冗余系统 S7-1500R/H 进行通信的信息	“与冗余系统 S7-1500R/H 进行通信 (页 406)”部分
在功能手册中扩展了 S7-1500R/H 冗余系统的内容	S7-1500R/H 冗余系统包含 SIMATIC S7-1500R 自动化系统中的常用功能。	《S7-1500R/H 冗余系统》系统手册 (https://support.industry.siemens.com/cs/ww/zh/view/109754833)

与 09/2016 版相比，《通信功能手册》版本 12/2017 中新增的内容

新增内容	客户收益	信息出处
OPC UA 配套规范	通过 OPC UA 配套规范，可独立于制造商统一指定方法。使用这些指定的方法，可轻松地将不同制造商的设备集成到工厂和生产过程中。	“OPC UA 服务器接口组态 (页 246)”部分
通过 CPU 接口建立与邮件服务器的安全连接	可直接与邮件服务器建立安全连接，而无需额外安装硬件装置。	“通过电子邮件实现 OUC (页 94)”部分
通过 Modbus TCP 进行安全通信	在 Modbus TCP 客户端与 Modbus TCP 服务器之间，可建立 TCP 安全连接。	“通过 Modbus TCP 进行 OUC 安全连接 (页 92)”部分

与 12/2014 版相比，《通信功能手册》版本 09/2016 中新增的内容

新增内容	客户收益	信息出处
OPC UA 服务器	OPC UA 是一种独立于操作系统的统一数据通信标准。 OPC UA 可在各种自动化系统中集成各种安全防护机制，例如在应用层中，进行合法用户的数据交换。 OPC UA 服务器可提供大量数据： • 客户端可访问的 PLC 变量值 • 这些 PLC 变量的数据类型 • OPC UA 服务器及 CPU 信息 这样，客户端可了解变量管理的概览信息而且可对这些值进行读写。	“OPC UA 通信 (页 155)”部分
开放式用户安全通信	与其它设备进行数据安全交换。	“开放式用户安全通信 (页 80)”部分
STEP 7 中的证书处理	在 STEP 7 中，可管理以下应用的证书： • OPC UA 服务器 • 开放式用户安全通信 • CPU 的 Web 服务器	“使用 TIA Portal 进行证书管理 (页 57)”部分
取消激活 CPU 的 SNMP 功能	可以取消激活 CPU 的 SNMP 功能。在某些情况下需要执行该功能。例如，网络中的安全规则不允许使用 SNMP 时。	“SNMP (页 108)”部分

约定

STEP 7：在本文档中，“STEP 7”是指组态与编程软件“STEP 7 V12 (TIA Portal) 及以上版本”。

由于控制器功能的连续性，术语“S7-1500 CPU”通常指 S7-1500F、S7-1500T、S7-1500TF、S7-1500C、S7-1500R/H、S7-1500pro 系列 CPU、ET200SP、S7-1500 软件控制器以及 SIMATIC Drive Controller。由于高级控制器、分布式控制器和软件控制器之间的设计和应用不同，会出现差异。

本文档中包含所述设备的相关图片。这些图可能与所提供的设备略有不同。

请特别关注以下注意事项：

说明

这些注意事项中包含有关产品、产品操作和文档中应特别关注部分的重要信息。

工业商城

工业商城为西门子公司推出的全集成自动化 (TIA) 和全集成能源管理 (TIP) 自动化与驱动解决方案产品目录和订购系统。

Internet (https://mall.industry.siemens.com) 提供自动化和驱动器领域的所有产品目录。

1.1 功能手册文档指南

1.1.1 信息类“功能手册”

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

用户可根据需要快速访问所需内容。

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

基本信息

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

STEP 7 在线帮助为用户提供了组态和编程方面的支持。

示例：

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

设备信息

设备手册中包含模块特定信息的简要介绍，如特性、接线图、功能特性和技术规范。

示例：

- CPU 设备手册
- “接口模块”设备手册
- “数字量模块”设备手册
- “模拟量模块”设备手册
- “通信模块”设备手册
- “工艺模块”设备手册
- “电源模块”设备手册
- BaseUnit 设备手册

常规信息

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

示例：

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

产品信息

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

有关产品信息的最新版本，敬请访问 Internet：

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

手册集

手册集中包含系统的完整文档，这些文档收集在一个文件中。

可以在 Internet 上找到手册集：

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

1.1.2 基本工具

工具

下面介绍的工具在所有步骤中都会为您提供支持：从规划到调试，再到系统分析。

TIA Selection Tool

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

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

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

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

SIMATIC Automation Tool

通过 SIMATIC Automation Tool，可对各个 SIMATIC S7 站进行调试和维护操作（作为批量操作），而无需打开 TIA Portal。

SIMATIC Automation Tool 可提供各种功能：

- 扫描 PROFINET/Ethernet 系统网络，识别所有连接的 CPU
- 为 CPU 分配地址（IP、子网、Gateway）和设备名称（PROFINET 设备）
- 将日期和已转换为 UTC 时间的编程设备/PC 时间传送到模块中
- 将程序下载到 CPU 中
- RUN/STOP 模式切换
- 通过 LED 闪烁进行 CPU 本地化
- 读取 CPU 错误信息
- 读取 CPU 诊断缓冲区
- 复位为出厂设置
- 更新 CPU 和所连接模块的固件

SIMATIC Automation Tool 可从 Internet 上下载。
(https://support.industry.siemens.com/cs/cn/zh/view/98161300/en)

PRONETA

SIEMENS PRONETA（PROFINET 网络分析）是一款调试和诊断工具，用于 PROFINET 网络。PRONETA Basic 有两个核心功能：

- 在网络分析中，您可以概览 PROFINET 拓扑。将真实组态与参考安装进行比较或进行简单的参数更改，例如设备的名称和 IP 地址。
- 通过 IO 测试，可简单、快速完成工厂接线和模块组态测试，其中包括测试结果的记录。

有关 SIEMENS PRONETA Basic，敬请访问 Internet。
(https://support.industry.siemens.com/cs/cn/zh/view/67460624)

SIEMENS PRONETA Professional 是为用户提供附加功能的许可产品。它提供在 PROFINET 网络中轻松管理资产的能力，还通过各种功能为自动化系统的操作员自动收集/获取所用组件的数据提供支持：

- 用户界面 (API) 提供自动化单元的访问点，以使用 MQTT 或命令行自动执行扫描功能。
- 借助 PROFIenergy 诊断，可以快速检测支持 PROFIenergy 的设备的当前暂停模式或运行准备情况，并根据需要进行更改。
- 数据记录向导可支持 PROFINET 开发人员在无需 PLC 和工程组态的情况下快速轻松地读取和写入非循环 PROFINET 数据记录。

可从 Internet 上下载 SIEMENS PRONETA Professional。 (https://www.siemens.com/proneta-professional)

SINETPLAN

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

优势概览：

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

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

1.1.3 S7 端口组态工具 (S7-PCT)

SIMATIC S7-PCT

Port Configuration Tool (PCT) 是一款基于 PC 的软件，用于为 Siemens IO-Link Master 模块和来自其它制造商的 IO-Link 设备分配参数。

可以使用从相应设备制造商处获得的标准化设备描述"IODD"集成 IO-Link- 设备。S7-PCT 支持 IODD 的 V1.0 和 V1.1 版本。

S7-PCT 通过来自 STEP 7 的 IO-Link Master 硬件配置进行调用。STEP 7 未使用或者 IO-Link Master 未工作在 SIMATIC 控制器上时，也可进行 "standalone"-操作。

有关 IO-Link 的更多信息，敬请访问 Internet (https://new.siemens.com/global/en/products/automation/industrial-communication/io-link.html)。

1.1.4 S7 Failsafe Configuration Tool (S7-FCT)

SIMATIC S7-FCT

Failsafe Configuration Tool (FCT) 使您能够通过 GSD 在第三方工程组态系统中组态以下设备：

- 选定的功能故障安全 SIMATIC I/O 设备
- 功能故障安全 SIRIUS ACT PROFINET 接口

为此，工程组态系统必须满足以下要求：

- 支持符合 "PROFIsafe - Profile for Safety Technology on PROFIBUS DP and PROFINET IO" 的 CPD 系统集成
- 符合 Conformance Class C3 的 TCI 实现

可以在 Internet 上找到有关 S7-FCT 的更多信息 (https://support.industry.siemens.com/cs/cn/zh/view/109762827)。

1.1.5 多现场总线组态工具 (MFCT)

MultiFieldbus Configuration Tool

MultiFieldbus Configuration Tool (MFCT) 是一款基于 PC 的软件，支持组态 MultiFieldbus- 和 DALI- 设备。此外，MFCT 还为支持 MultiFieldbus- 的 ET 200 设备的大规模固件更新以及读取许多其它西门子设备的服务数据提供了方便的选项。

MFCT 的功能范围

- MultiFieldbus 组态：
  MultiFieldbus- 设备的工程组态、组态和诊断，提供所需的项目文件（项目、UDT-、CSV- 和 EDS- 文件），将文件传输/导出到设备和/或数据存储器。
- DALI 组态：
  DALI 设备的设备选型和在线组态。
- TM FAST：
  生成和下载 FPGA-UPD- 和 FPGA-DB-文件。
- 维护：
  Ethernet 网络的拓扑扫描、读取服务数据、参数分配和固件更新。
- 设置：
  德语/英语语言切换，网络扫描仪速度，网络适配器设置，安装 GSDML- 和 EDS- 文件。

MFCT 的系统/安装要求

MFCT 在 Microsoft Windows 环境下运行，不需要安装或管理员权限。

对于 MFCT，还必须安装以下软件：

- Microsoft .NET Framework 4.8（您可以在 Internet 上找到离线安装程序。(https://support.microsoft.com/en-us/topic/microsoft-net-framework-4-8-offline-installer-for-windows-9d23f658-3b97-68ab-d013-aa3c3e7495e0)）
- "Misc" 目录中的 NPcap
- "Misc" 目录中的 PG/PC interface
- 适用于 x86- 系统的 Microsoft C++ Redistributable（您可以在 Internet 上找到可下载的安装数据。(https://aka.ms/vs/15/release/vc_redist.x86.exe)）

可以在 Internet 上找到该工具的下载以及有关 MFCT 各个功能的更多信息和文档。(https://support.industry.siemens.com/cs/de/en/view/109773881)

1.1.6 SIMATIC 技术文档

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

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

SIMATIC 技术文档概述

可以在此处找到西门子工业在线技术支持中可用的 SIMATIC 文档的概述：

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

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

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

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

保留文档

保留本文档供以后使用。

对于以数字形式提供的文档：

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

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

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

更新文档

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

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

我的技术支持

通过“我的技术支持”，可以最大程度善用您的工业在线支持服务。

注册	要使用“我的技术支持”中的所有功能，必须先进行注册。注册后，可以在个人工作区中创建过滤器、收藏夹和选项卡。
支持申请	支持申请页面还支持用户资料自动填写，用户可随时查看当前的所申请的支持请求。
文档	在“文档”(Documentation) 区域中，可以构建您的个人库。
收藏夹	可使用“添加到我的技术支持收藏夹”(Add to mySupport favorites) 来标记特别感兴趣或经常需要的内容。在“收藏夹”(Favorites) 下，会显示所标记条目的列表。
最近查看的文章	“我的技术支持”中最近查看的页面位于“最近查看的文章”(Recently viewed articles) 下。
CAx 数据	借助 CAx 数据区域，可以访问 CAx 或 CAe 系统的最新产品数据。仅需单击几次，用户即可组态自己的下载包： • 产品图片、二维码、3D 模型、内部电路图、EPLAN 宏文件 • 手册、功能特性、操作手册、证书 • 产品主数据

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

应用示例

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

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

安全信息

2

2.1 一般安全信息

请注意相应系统手册中提供的安全信息。

有关网络安全的信息，请参见“工业网络安全 (页 24)”部分。

工业网络安全 3

由于机器和工厂的数字化和网络化程度在不断提高，网络攻击的风险也在不断增加。因此，采取适当的保护措施势在必行，重要的基础设施尤为如此。

有关工业网络安全的一般信息和措施，请参见系统手册和 SIMATIC HMI 设备的安全指南 (https://support.industry.siemens.com/cs/cn/zh/view/109481300)。

本节将概括介绍与 SIMATIC 系统通信相关的安全相关信息。

说明

有关软件或设备的安全相关更改，请参见"简介 (页 10)"部分。

3.1 网络安全信息

西门子为其产品及解决方案提供了工业网络安全功能，以支持工厂、系统、机器和网络的安全运行。

为了防止工厂、系统、机器和网络受到网络攻击，需要实施并持续维护先进且全面的工业网络安全保护机制。西门子的产品和解决方案构成此类概念的其中一个要素。

客户负责防止其工厂、系统、机器和网络受到未经授权的访问。只有在有必要连接时并仅在采取适当安全措施（例如，防火墙和/或网络分段）的情况下，才能将该等系统、机器和组件连接到企业网络或 互联网。关于可采取的工业网络安全措施的更多信息，请访问 https://www.siemens.com/cybersecurity-industry 。

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

要及时了解有关产品更新的信息，请订阅西门子工业网络安全 RSS 源，网址为 https://www.siemens.com/cert 。

3.2 本手册中的安全相关信息

请留意本通信手册中与各个主题相关的所有安全相关注意事项。

有关以下项的安全注意事项	部分
接口	产品概述 (页 26)
端口和协议	通信选件概述 (页 31) 用于以太网通信的通信协议和端口号 (页 33)
安全通信	安全通信 (页 47)
激活/取消激活服务	激活和取消激活 SNMP (页 108) 通过 DHCP 寻址 (页 352) IP 转发 (页 380)
OPC UA 通信的安全功能（身份验证、证书、用户创建和角色、安全消息传输）	OPC UA 通信 (页 155)

有关以下项的安全注意事项	部分
激活通信模块中的安全功能	基于 IP 的应用程序的虚拟接口 (页 388)
冗余系统 S7-1500R/H 的通信	冗余系统 S7-1500R/H 的通信 (页 406)
使用 CP 1543-1 实现工业以太网安全保护	使用 CP 1543-1 确保工业以太网安全 (页 444)
安全的 PG/HMI 通信	安全的 PG/HMI 通信 (页 97)
保护机密数据	保护机密组态数据 (页 68)

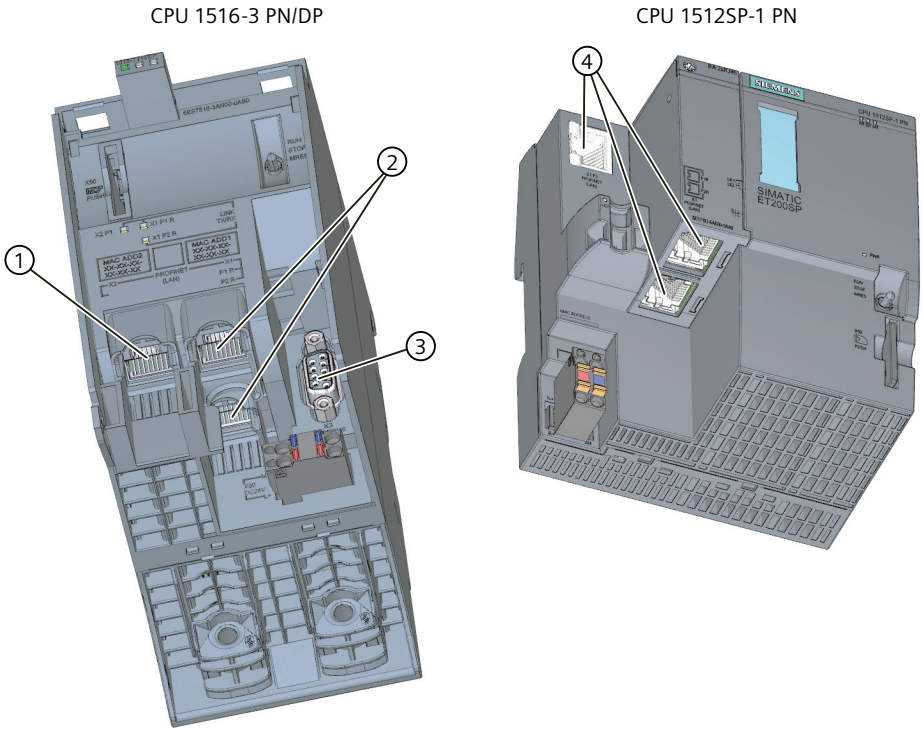
产品概述

4

S7-1500、ET 200MP、ET 200SP、ET 200pro 和 ET 200AL 系统的 CPU、通信模块和处理器以及 PC 系统提供相应接口，支持通过 PROFINET、PROFIBUS 和点到点连接进行通信。

CPU、通信模块和通信处理器

PROFINET 和 PROFIBUS DP 接口集成在 S7-1500 CPU 中。例如，CPU 1516-3 PN/DP 上带有 2 个 PROFINET 接口和 1 个 PROFIBUS DP 接口。使用通信模块 (CM) 和通信处理器 (CP) 时，可支持其它 PROFINET 和 PROFIBUS DP 接口。

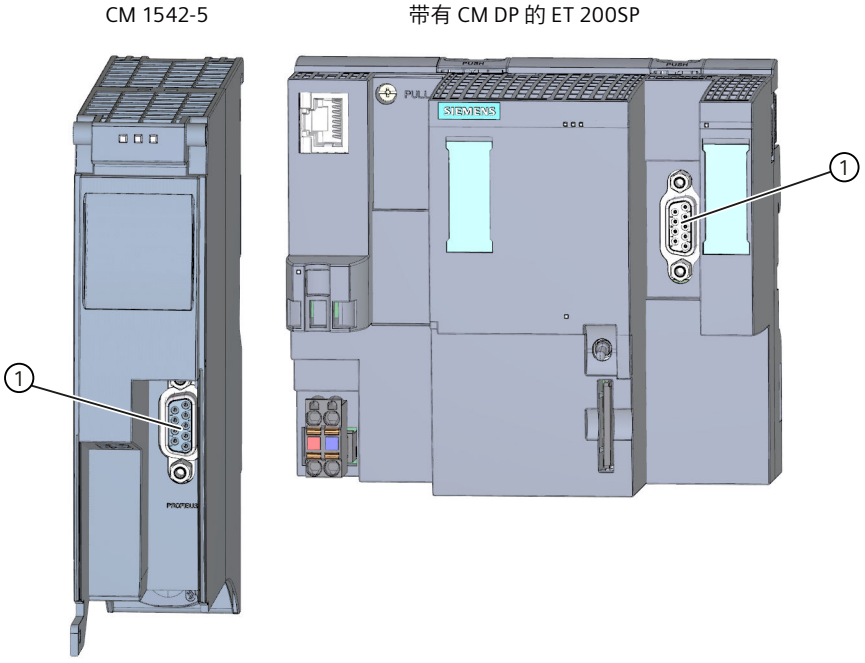


① PROFINET 接口 (X2)，带一个端口
② PROFINET 接口 (X1)，带双端口交换机
③ PROFIBUS DP 接口 (X3)
④ PROFINET 接口 (X1)，带 3 端口交换机

图 4-1 CPU 1516-3 PN/DP 和 CPU 1512SP-1 PN 的接口

通信模块的接口

通信模块 (CM) 接口对 CPU 的接口进行了扩展（例如，使用通信模块 CM 1542-5 时，S7-1500 自动化系统将增加一个 PROFIBUS 接口）。

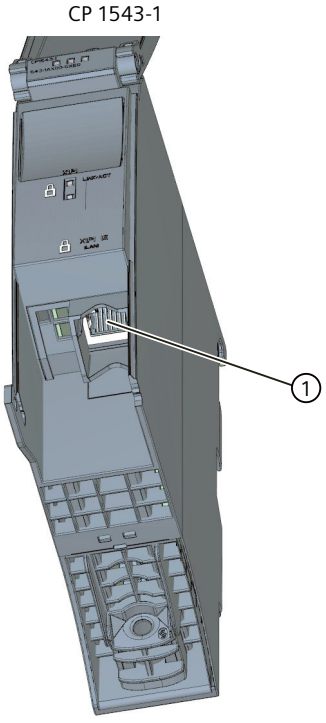


① PROFIBUS DP 接口

图 4-2 CM 1542-5 和 CM DP 的 PROFIBUS DP 接口（连接至 ET 200SP CPU）

通信处理器的接口

通信处理器 (CP) 接口为 CPU 中集成接口所提供的功能提供了附加功能。CP 支持一些特殊功能。例如，CP 1543-1 支持安全功能，可通过工业以太网接口确保工业以太网安全。



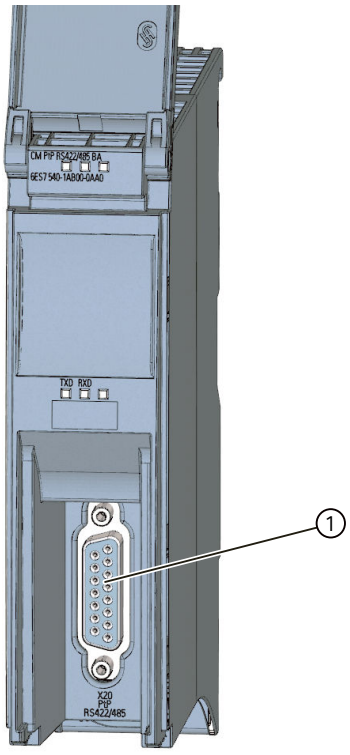
① 工业以太网接口

图 4-3 CP 1543-1 的工业以太网接口

进行点到点连接的通信模块接口

点到点连接通信模块可通过 RS 232-、RS 422- 和 RS 485 接口进行通信，如 Freeport 或 Modbus 通信。

CM PtP RS422/485 BA

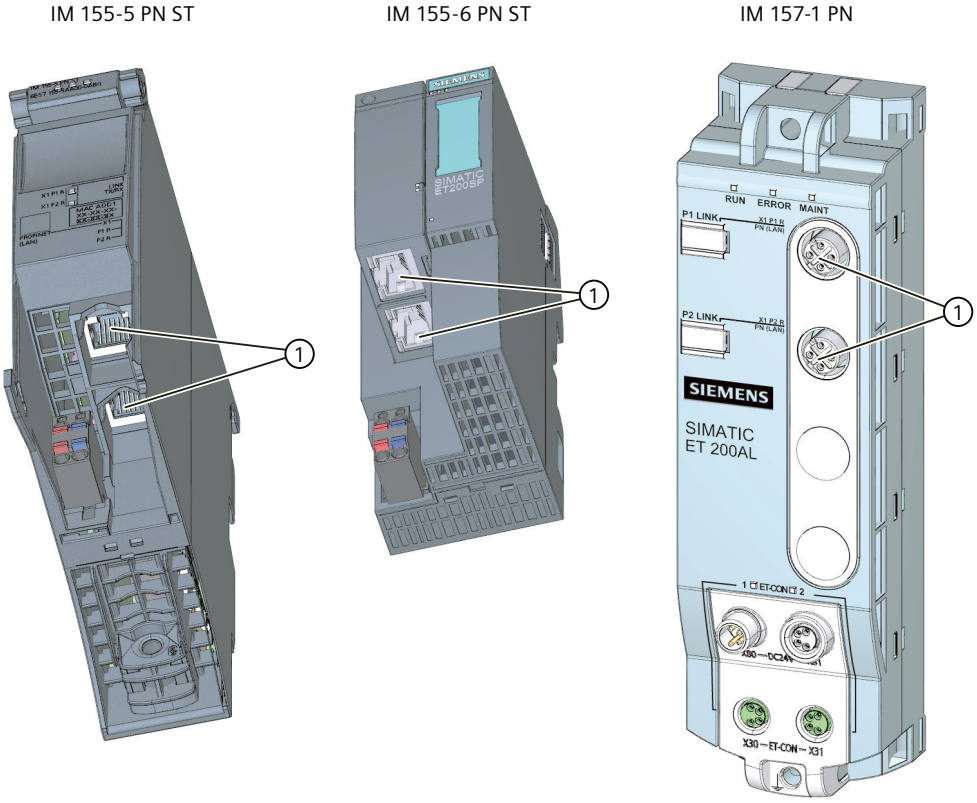


① 点到点连接接口

图 4-4 CM PtP RS422/485 BA 中点到点连接的接口示例。

接口模块上的接口

通过 ET 200MP、ET 200SP 和 ET 200AL 接口模块 (IM) 上的 PROFINET 和 PROFIBUS DP 接口，可将分布式 I/O ET 200MP、ET 200SP 和 ET 200AL 连接到上位 I/O 控制器或 DP 主站的 PROFINET 或 PROFIBUS 中。



① PROFINET 接口，带双端口交换机

图 4-5 PROFINET 接口 IM 155-5 PN ST (ET 200MP)、IM 155-6 PN ST (ET 200SP) 和 IM 157-1 PN (ET 200AL)

取消激活 PROFINET 接口的端口

激活的端口代表着未经授权访问网络和程序的潜在风险。
为了最大限度地降低风险，请取消激活所有未使用的端口。
操作步骤：

1. 在 STEP 7 中的 CPU/IM 的设备视图中，选择 PROFINET 接口的端口。
2. 在巡视窗口中，浏览到"高级选项 > 端口 [x] > 端口选项 > 激活"(Advanced options > Port [x]> Port options > Activate)。
3. 取消选中"激活"(Activate) 复选框。

CPU/IM 的至少一个端口保持激活状态。

通信服务

下文中所介绍的通信服务，将使用系统中 CPU、通信模块和处理器的接口和通信机制进行通信。

通信服务

5

5.1 通信方式概述

通信方式概述

在执行自动化任务时，可使用以下通信方式。

表格 5-1 通信方式

通信方式	功能	接口：		
		PN/IE[1]	DP	串行
PG 通信[2]	调试、测试、诊断	✓	✓	-
HMI 通信[2]	操作员监控	✓	✓	-
通过 TCP/IP 协议实现开放式通信[2]	基于 TCP/IP 协议，通过 PROFINET/工业以太网进行数据交换 指令： • TSEND_C/TRCV_C • TSEND/TRCV • TCON • T_DISCON	✓	-	-
基于 ISO-on-TCP 进行开放式通信[2]	基于 ISO-on-TCP 协议，通过 PROFINET/工业以太网进行数据交换 指令： • TSEND_C/TRCV_C • TSEND/TRCV • TCON • T_DISCON	✓	-	-
通过 UDP 实现开放式通信[2]	基于 UDP 协议，通过 PROFINET/工业以太网进行数据交换 指令： • TSEND_C/TRCV_C • TUSEND/TURCV • TCON • T_DISCON	✓	-	-
基于 ISO 进行开放式通信（仅适用于带有 PROFINET/工业以太网接口的 CP）	基于 ISO 协议，通过 PROFINET/工业以太网进行数据交换 指令： • TSEND_C/TRCV_C • TSEND/TRCV • TCON • T_DISCON	✓	-	-

1 IE - 工业以太网

2 遵守 S7-1500R/H 的特殊特性

3 仅通过 CPU 的内部 PROFINET 接口和激活"通过通信模块访问 PLC"功能的以太网接口 CP 1543 1。

通信方式	功能	接口：		
		PN/IE[1]	DP	串行
基于 FDL 进行开放式通信（仅适用于 CM 1542-5 固件版本 V2.0 及以上版本）	基于 FDL 协议，通过 PROFIBUS 进行数据交换 指令： • TSEND_C/TRCV_C • TSEND/TRCV • TUSEND/TURCV • TCON • T_DISCON	-	✓	-
OPC UA 服务器[3]	通过 OPC UA 客户端进行数据交换	✓	-	-
基于 Modbus TCP 进行通信	基于 Modbus TCP 协议，通过 PROFINET 进行数据交换 指令： • MB_CLIENT • MB_SERVER	✓	-	-
电子邮件	通过电子邮件发送过程报警 指令： • TMAIL_C	✓	-	-
FTP（仅适于带有 PROFINET/工业以太网接口的 CP）	基于 FTP（文件传输协议）进行文件管理和文件访问时，CP 既可以作为 FTP 客户端也可以作为 FTP 服务器 指令： • FTP_CMD	✓	-	-
Fetch/Write（仅适于带有 PROFINET/工业以太网接口的 CP）	通过 TCP/IP、ISO-on-TCP 和 ISO 执行服务器服务 通过 Fetch/Write 的特殊指令	✓	-	-
S7 通信	通过 PROFINET/PROFIBUS，使用 S7 协议进行数据交换。 指令： • PUT/GET • BSEND/BRCV • USEND/URCV	✓	✓	-
点到点串行连接	基于 Freeport、3964(R)、USS 或 Modbus 协议，进行点到点数据交换 通过 PtP、USS 或 Modbus RTU 的特定指令	-	-	✓
Web 服务器	通过 HTTP(S) 进行数据交换，如诊断	✓	-	-
SNMP（简单网络管理协议）	基于标准 SNMP 协议，通过对 IP 网络组件进行参数设置，可对 IP 网络进行监控和故障识别	✓	-	-
时间同步	通过 PN/IE 接口：CPU 作为 NTP 客户端（网络时间协议）	✓	-	-
	通过 DP 接口：CPU/CM/CP 作为时间主站或时间从站	-	✓	-

[1] IE - 工业以太网

[2] 遵守 S7-1500R/H 的特殊特性

[3] 仅通过 CPU 的内部 PROFINET 接口和激活“通过通信模块访问 PLC”功能的以太网接口 CP 1543 1。

有关 S7-1500R/H 的信息

有关与 S7-1500R/H 冗余系统通信可能性的信息，请参见“与冗余系统 S7-1500R/H 进行通信 (页 406)”部分。

更多信息

- 有关基于 TLS 的 PG/HMI 通信的组态以及 CPU 机密组态数据保护的应用示例，请参见本应用示例 (https://support.industry.siemens.com/cs/ww/zh/view/109798583)。
- 有关与 SIMATIC 控制器之间的 CPU-CPU 通信（概要）的常规应用示例，请参见本应用示例 (https://support.industry.siemens.com/cs/cn/zh/view/20982954)。
- TIA 库“LOpcUa”提供用于为 SIMATIC S7-1500 实现 OPC UA PubSub 的函数块，有关该库的信息，请参见本应用示例 (https://support.industry.siemens.com/cs/ww/zh/view/109782455)。
- 在该常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/102420020)中，介绍了如何通过 S7-1500 中的 CP1543-1 组态获取/写入通信。
- 有关获取/写入服务的更多信息，请参见 STEP 7 在线帮助。
- 有关 PtP 连接的更多信息，请参见功能手册《CM PtP - 点到点连接组态 (https://support.industry.siemens.com/cs/cn/zh/view/59057093)》。
- 有关 Web 服务器的功能介绍，请参见功能手册《Web 服务器 (https://support.industry.siemens.com/cs/cn/zh/view/59193560)》。
- 有关 SNMP 标准协议的常规信息，敬请访问 Internet (https://support.industry.siemens.com/cs/cn/zh/view/15166742) 中的服务与支持页面。有关哪些 SNMP 请求支持 S7-1500 CPU 和 S7-1200 CPU 这一问题的答案，请参见“常见问题与解答 (https://support.industry.siemens.com/cs/at/zh/view/79993228)”。
- 有关时间同步的信息，请参见“常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/86535497)”。

5.2 以太网通信的通信协议和端口号

在本章节中，简要介绍了通过 PN/IE 接口进行通信时支持的协议和端口号。在各种协议中，分别指定了地址参数、相应的通信层以及通信角色和通信方向。

基于这些信息，可将自动化系统所有的安全保护措施与相应的协议进行匹配（如，防火墙）。由于安全措施仅限于以太网或 PROFINET 网络，因此表格中不包含任何 PROFIBUS 协议。

说明

使用的端口号

指定的端口号为 S7-1500 CPU、通信模块和 ET 200 接口模块使用的标准端口号。由于支持各种不同的通信协议和通信连接，因此也可使用其它端口号。

下表列出了 S7-1500 CPU、S7-1500 通信模块和 ET 200 接口模块中使用的不同层和协议。

S7-1500 CPU 和软件控制器的通信层和协议（通过 CPU 的 PROFINET 接口）

下表列出了 S7-1500 CPU、ET 200SP CPU 和 1513/1516pro-2 PN CPU 支持的协议。S7-1500 软件控制器也支持下表中所列协议，以太网接口将基于这些协议分配给相应的软件控制器。

表格 5-2 S7-1500 CPU 和软件控制器的通信层和协议（通过 CPU 的 PROFINET 接口）

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	默认设置/说明
PROFINET 协议				
DCP	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET Discovery and Basic Configuration Protocol. DCP 决定 PROFINET 设备并启用基本设置。	默认值：固件版本 V3.0 及以下版本启用，固件版本 V3.1 及更高版本启用写保护。 在有效的通信关系期间，DCP 不允许在写保护模式下从外部发出 DCP Set 命令。 可通过 CPU 属性中接口的 Boundary"可访问节点检测结束"(End of detection of accessible nodes) 取消激活此功能。
DHCP 客户端	68	(4) UDP	Dynamic Host Configuration Protocol. IP 地址套件是在 PROFINET 接口启动期间从 DHCP 服务器获取的。	默认值：取消激活。 可以在 CPU 属性中更改（自固件版本 2.9 起）。
LLDP	不相关	(2) Ethertype 0x88CC (LLDP)	PROFINET Link Layer Discovery Protocol. LLDP 决定和管理 PROFINET 设备间的相邻关系。	默认值：激活。 可通过 CPU 属性中的 Boundary"拓扑发现结束"(End of topology discovery) 取消激活发送功能；仍处于准备接收状态。 LLDP 使用特定的多播 MAC 地址 01-80-C2-00-00-0E.
MRP	不相关	(2) Ethertype 0x88E3 (IEC 62493-2-2010)	Media Redundancy Protocol. MRP 采用环形拓扑结构对冗余传输路径进控制。	默认值："管理器（自动）"。 可以在 CPU 属性中更改。如果组态 CPU 并将 PN 接口与子网连接，则 TIA Portal 中的默认设置为"非环中的设备"。 MRP 使用标准的多播 MAC 地址。
PROFINET IO 数据	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET Cyclic IO Data Transfer. 通过 PROFINET IO 报文，基于以太网在 PROFINET IO 控制器与 IO 设备之间循环传输 IO 数据。	默认值：取消激活。 仅为 PROFINET IO 数据流量激活此协议。
PROFINET 上下文管理器	34964	(4) UDP	PROFINET 通信。 管理 IO 控制器与 IO 设备之间的应用和通信关系。	默认值：激活（UDP 端口打开）。 不能取消激活此功能。
PTCP	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET Precision Transparent Clock Protocol，基于 IEEE 1588。 PTCP 提供 RJ45 端口之间的延时测量，并随后发送时钟和时间同步。	默认值：取消激活。 可通过以下组态激活： • 采用同步域的 IRT。 • 通过指定长度的电缆进行端口互连。 可通过 CPU 属性中接口的"同步域结束"(End of sync domain) Boundary 取消激活此功能。 PTCP 使用标准的多播 MAC 地址。

[1] 注：OUC（开放式通信）可直接访问 UDP 和 TCP 协议。必须考虑 IANA (Internet Assigned Numbers Authority) 端口限制和定义。

[2] 请勿将其它协议已用端口分配给 OUC。

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	默认设置/说明
面向连接的通信协议				
HTTP 服务器	80	(4) TCP	Hypertext Transfer Protocol. HTTP 用于与 CPU 内部 Web 服务器通信。	默认值：取消激活。 可以在 CPU 属性中启用。 要求：在 CPU 的属性中启用 Web 服务器。
HTTPS 服务器	443	(4) TCP	Hypertext Transfer Protocol Secure. HTTPS 用于通过安全套接层 (SSL) 与 CPU 内部的 Web 服务器通信。	默认值：取消激活。 可以在 CPU 属性中启用。 要求：在 CPU 的属性中启用 Web 服务器。
IGMPv2	不相关	(3) 网络层	Internet Group Management Protocol. IGMPv2 是用于组织多播组的网络协议（仅限 UDP 多播）。	IGMPv2 是 IP 堆栈的功能。此系统功能通过多播功能激活。
ISO-on-TCP 服务器	102	(4) TCP	ISO-on-TCP 协议（基于 RFC 1006）。 S7 协议使用 ISO-on-TCP（基于 RFC 1006）与工程组态系统进行 PG/HMI 通信 (TIA Portal)。	默认值：激活。 不能取消激活此功能。
MODBUS TCP 服务器/客户端	502	(4) TCP	MODBUS Transmission Control Protocol. MODBUS/TCP 由用户程序中的 MB_CLIENT/MB_SERVER 指令使用。	默认值：取消激活。 可在用户程序中通过 Modbus 指令激活。
NTP 客户端	123	(4) UDP	Network Time Protocol. NTP 用于同步 CPU 系统时间与 NTP 服务器时间。	默认值：取消激活。 可以在 CPU 属性中启用。
OPC UA 服务器/客户端	4840	(4) TCP	Open Platform Communications Unified Architecture（基于 TCP/IP 协议）。 从企业级到现场级的通信标准。	默认值：取消激活。 可以在 CPU 属性中启用服务器和客户端功能。 可在用户程序中组态客户端访问。
OUC[1] OUC 安全连接 服务器/客户端	1 到 1999 使用范围有限[2] 2000 到 5000 （建议） 自固件版本 V3.0 起，以下要求适用于程序设定连接和已组态连接： 5001 ... 65535 使用范围有限[2]	(4) TCP (4) UDP (4) ISO-on-TCP（端口：102）	Open User Communication (TCP/UDP)。 Secure Open User Communication (TLS)。 OUC 指令可通过用户程序建立连接、终止连接和传输数据。	默认值：取消激活。 在用户程序中通过相应的 Open User Communication 指令或通过网络视图中的连接组态激活各个协议。 以下要求适用固件版本低于 V3.0 的情况： • 程序设定的连接：5001 ... 49152 • 通过组态建立连接：5001 ... 65535

[1] 注：OUC（开放式通信）可直接访问 UDP 和 TCP 协议。必须考虑 IANA (Internet Assigned Numbers Authority) 端口限制和定义。

[2] 请勿将其它协议已用端口分配给 OUC。

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	默认设置/说明
SMTP 客户端	25	(4) TCP	Simple Mail Transfer Protocol. SMTP 用于发送电子邮件。	默认值：取消激活。 可在用户程序中通过 TMAIL_C 指令启用。
SMTPS（通过 TLS 进行 SMTP 连接） 客户端	465	(4) TCP	Simple Mail Transfer Protocol Secure. SMTP 用于通过安全连接发送电子邮件。	默认值：取消激活。 可在用户程序中通过 TMAIL_C 指令启用。
使用 STARTTLS 进行 SMTP 连接 客户端	25 587	(4) TCP	Simple Mail Transfer Protocol 使用 SMTP 命令 "STARTTLS". SMTP 用于发送电子邮件。	默认值：取消激活。 可在用户程序中通过 TMAIL_C 指令启用。
SNMP 代理	161 162（陷阱）	(4) UDP	Simple Network Management Protocol. SNMP 管理器使用 SNMP 读取和设置网络管理数据（SNMP 管理的对象）。	默认值：固件版本 V2.9 及以下版本激活，固件版本 V3.0 及更高版本取消激活。 可在用户程序中通过数据记录启用。 可在 CPU 属性中启用（自固件版本 V3.0 起）。 自固件版本 V3.1 起，可在 CPU 属性中额外启用写保护。
Syslog（系统日志）	6514 514	(4) TCP (4) UDP	Syslog 属于 IETF 标准协议 (RFC 5424)，用于传输 CPU 检测到的事件。	默认值：取消激活。 可以在 CPU 属性中启用。 可在 CPU 属性中组态为，将 Syslog 消息转发到 Syslog 服务器。自固件版本 V3.1 起，无法禁用 CPU 内系统日志事件的收集。
预留	49152 到 65535	(4) TCP (4) UDP	如果应用程序未寻址到本地端口，CPU 会为活动连接点使用该端口范围。	-

[1] 注：OUC（开放式通信）可直接访问 UDP 和 TCP 协议。必须考虑 IANA (Internet Assigned Numbers Authority) 端口限制和定义。

[2] 请勿将其它协议已用端口分配给 OUC。

S7-1500 软件控制器的通信层和日志（通过 Windows 端的以太网接口）

下表列出了 S7-1500 软件控制器支持的协议，通过以太网接口分配给 Windows 系统。

表格 5-3 S7-1500 软件控制器的通信层和日志（通过 Windows 端的以太网接口）

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	注意/默认设置
PROFINET 协议				
DCP	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET Discovery and Basic Configuration Protocol. DCP 决定 PROFINET 设备并启用基本设置。	默认值：版本 V30.0 及以下版本启用。自版本 V30.1 起启用写保护。 在有效的通信关系期间，DCP 不允许在写保护模式下从外部发出 DCP Set 命令。 可通过 CPU 属性中的 Boundary"可访问节点检测结束"(End of detection of accessible nodes) 禁用此功能。
DHCP 客户端	68	(4) UDP	Dynamic Host Configuration Protocol. IP 地址套件是在 PROFINET 接口启动期间从 DHCP 服务器获取的。	默认值：取消激活。 可以在 CPU 属性中更改（自固件版本 2.9 起）。
面向连接的通信协议				
HTTP 服务器	可调节 [1]	(4) TCP	Hypertext Transfer Protocol。 HTTP 用于与 CPU 内部 Web 服务器通信。	默认值：取消激活。 可以在 CPU 属性中更改。 为了避免与 Windows 系统中其它 Web 服务器冲突，可调整端口号。如果使用 S7-1500 软件控制器的 Web 服务器访问，必须在 Windows 防火墙中启用分配的端口。
IGMPv2	不相关	(3) 网络层	Internet Group Management Protocol. IGMPv2 是用于组织多播组的网络协议（仅限 UDP 多播）。	IGMPv2 是 IP 堆栈的功能。此系统功能通过多播功能激活。
ISO-on-TCP 服务器	102	(4) TCP	ISO-on-TCP 协议（基于 RFC 1006）。 S7 协议使用 ISO-on-TCP（基于 RFC 1006）与工程组态系统进行 PG/HMI 通信 (TIA Portal)。	默认值：取消激活。
OUC[2] 和 OUC 安全连接	1 到 1999 使用范围有限[3, 4]	(4) TCP (4) UDP (4) ISO-on-TCP（端口：102）	Open User Communication (TCP/UDP)。 Secure Open User Communication (TLS)。 OUC 指令可基于套阶层建立连接、终止连接和进行传输数据。	默认值：取消激活。 可在用户程序中通过数据记录启用。 如果要使用 OUC，则必须在 Windows 防火墙中激活该端口。
	2000 到 5000（建议）[4]			
	5001 到 49151 使用范围有限[3, 4]			

[1] Windows 分配的接口的默认设置：81

[2] 注：OUC（开放式通信）可直接访问 UDP 和 TCP 协议。必须考虑 IANA (Internet Assigned Numbers Authority) 端口限制和定义。

[3] 请勿将其它协议已用端口分配给 OUC。

[4] 请勿将其它 Windows 应用已用端口分配给 OUC。

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	注意/默认设置
SMTP 客户端	25	(4) TCP	Simple Mail Transfer Protocol. SMTP 用于发送电子邮件。	默认值：取消激活。 可通过在用户程序中调用块激活，或通过 CPU 设置激活（从版本 V3.0 开始）。
Syslog（系统日志）	6514 514	(4) TCP (4) UDP	Syslog 属于 IETF 标准协议 (RFC 5424)，用于传输 CPU 检测到的事件。	默认值：取消激活。 可以在 CPU 属性中启用。 可在 CPU 属性中组态为，将 Syslog 消息转发到 Syslog 服务器。自固件版本 V3.1 起，无法禁用 CPU 内系统日志事件的收集。
预留	49152 到 65535	(4) TCP (4) UDP	如果应用程序未指定本地端口号，将为活动的连接端点使用该动态端口范围。	如果要使用该连接，则必须在 Windows 防火墙中激活这些端口。

1 Windows 分配的接口的默认设置：81

2 注：OUC（开放式通信）可直接访问 UDP 和 TCP 协议。必须考虑 IANA (Internet Assigned Numbers Authority) 端口限制和定义。

3 请勿将其它协议已用端口分配给 OUC。

4 请勿将其它 Windows 应用已用端口分配给 OUC。

S7-1500 通信模块的层和协议

有关 S7-1500 通信模块（例如 CP 1543-1）协议的文档，请参见此处 (https://support.industry.siemens.com/cs/cn/zh/view/67700710)。

接口模块的层和协议

下表列出了 ET 200 接口模块所支持的协议。要了解 ET 200 接口模块所支持的协议，请参见相关设备手册中的技术规范。

表格 5-4 ET 200 接口模块的层和协议

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	默认设置/说明
PROFINET 协议				
DCP	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET Discovery and Basic Configuration Protocol. DCP 决定 PROFINET 设备并启用基本设置。	如果上位 IO 控制器也支持 DCP，则支持的接口模块的默认设置为： • 启用写保护 在有效的通信关系期间，DCP 不允许在写保护模式下从外部发出 DCP Set 命令。 可在 PROFINET 接口的属性中禁用该功能。

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	默认设置/说明
LLDP	不相关	(2) Ethertype 0x88CC (LLDP)	PROFINET Link Layer Discovery Protocol. LLDP 决定和管理 PROFINET 设备间的相邻关系。	默认值：激活。 可通过 PROFINET 接口属性中的 Boundary"拓扑发现结束"(End of topology discovery) 取消激活发送功能；仍处于准备接收状态。 LLDP 使用特定的多播 MAC 地址 01-80-C2-00-00-0E.
MRP	不相关	(2) Ethertype 0x88E3 (IEC 62493-2-2010)	Media Redundancy Protocol. MRP 采用环形拓扑结构对冗余传输路径进控制。	默认值："非环网中的设备"。 可在 PROFINET 接口属性中更改。如果要组态接口模块并激活 MRP，请为接口模块分配"客户端"介质冗余角色。 MRP 使用标准的多播 MAC 地址。
PROFINET IO 数据	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET Cyclic IO Data Transfer. 通过 PROFINET IO 报文，基于以太网在 PROFINET IO 控制器与 IO 设备之间循环传输 IO 数据。	默认值：取消激活。 一旦接口模块接收到有效组态，就会间接激活。
PROFINET 上下文管理器	34964	(4) UDP	PROFINET 通信。 管理 IO 控制器与 IO 设备之间的应用和通信关系。	默认值：激活（UDP 端口打开）。 不能取消激活此功能。
PTCP	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET Precision Transparent Clock Protocol，基于 IEEE 1588。 PTCP 支持以下功能： • 两个端口之间的时间延迟测量 • 发送时钟与时间同步	时间延迟测量的默认设置：已为支持的接口模块激活。要进行时间延迟测量，不能禁用 PTCP。 发送时钟和时间同步的默认设置：取消激活。可根据组态由上位 IO 控制器激活。 PTCP 使用标准的多播 MAC 地址。
面向连接的通信协议				
IGMPv2	不相关	(3) 网络层	Internet Group Management Protocol. IGMPv2 是用于组织多播组的网络协议（仅限 UDP 多播）。	IGMPv2 是 IP 堆栈的功能。此系统功能通过多播功能激活。
MODBUS TCP 从站	502	(4) TCP	MODBUS Transmission Control Protocol.	默认值：取消激活。 可通过多现场总线组态工具 (MFCT) 激活。
SNMP 代理	161 162（陷阱）	(4) UDP	Simple Network Management Protocol. SNMP 管理器使用 SNMP 读取和设置网络管理数据（SNMP 管理的对象）。	支持接口模块的默认设置： • 与 IO 控制器中的默认值同步 • 取消激活
EtherNet/IP CIP	44818 (TCP) 2222 (UDP)	(4) TCP (4) UDP	EtherNet/IP 协议和通用工业协议 (CIP) 协议可实现接口模块和 CPU 之间的通信。 EtherNet/IP 使用多现场总线设备进行数据交换。 CIP 隐式消息传送用于在时间关键型应用中连续传输实时数据。 CIP 显式消息传送用于有针对性地检索或传输组态和诊断数据。	默认值：取消激活。 可通过多现场总线组态工具 (MFCT) 激活。
TCP/IP	不相关	(4) TCP	Transmission Control Protocol. TCP 可确保 2 个通信伙伴之间的可靠连接。	默认值：激活。 不能取消激活此功能。

协议/角色	端口号	(2) 链路层 (4) 传输层	说明/功能	默认设置/说明
UDP	不相关	(4) UDP	User Datagram Protocol. UDP 基于不安全的互联网协议 (IP)，无需连接即可工作。	默认值：激活。 不能取消激活此功能。
ARP	不相关	(2) Ethertype 0x0806	Adress Resolution Protocol. 将 IP 地址分配给 MAC 地址。	默认值：激活。 不能取消激活此功能。
IPv4	不相关	(3) 网络层	Internet Protocol Version 4. 通过 IP 地址识别设备。IPv4 用于将数据包从其来源发送到其目标。	默认值：激活。 不能取消激活此功能。
预留	49152 到 65535	(4) TCP (4) UDP	如果应用程序未寻址到本地端口，CPU 会为活动连接点使用该端口范围。	-

5.3 连接资源概览

连接资源

某些通信服务需要进行连接。连接需要占用所用 CPU、CP 和 CM 中的资源（例如，CPU 操作系统中的存储区域）。大多数情况下，每个 CPU/CP/CM 都将占用一个连接。在 HMI 通信中，每个 HMI 连接最多需要 3 个连接资源。

具体可用的连接资源，取决于所用的 CPU、CP 和 CM，且不得超出自动化系统中定义的上限。

站中的可用连接资源

站中最大的可用资源数量取决于 CPU。

每个 CPU 都会为 PG、HMI 和 Web 服务器通信预留一定数量的连接资源。此外，还会为 SNMP、电子邮件连接、HMI 和 S7 通信以及开放式通信等其它通信服务提供资源。

何时占用连接资源？

连接资源的占用时间，取决于连接建立、自动连接、编程或组态的方式（参见"建立连接 (页 41)"部分）。

更多信息

有关连接资源占用以及连接资源在 STEP 7 中显示的更多信息，请参见"连接资源 (页 392)"部分。

5.4 建立连接

自动连接

如果将 PG/PC 接口物理连接到 CPU 的接口，并通过 STEP 7 中的“转至在线”(Go online) 对话框进行了接口分配，则 STEP 7 将自动建立连接（例如，PG 或 HMI 连接）。

建立编程连接

在 STEP 7 的程序编辑器中，可根据所选 CPU 使用相应的通信指令（如 TSEND_C）建立编程连接。

指定连接参数（在巡视窗口、指令属性中）时，通过用户界面使得操作更为方便快捷。

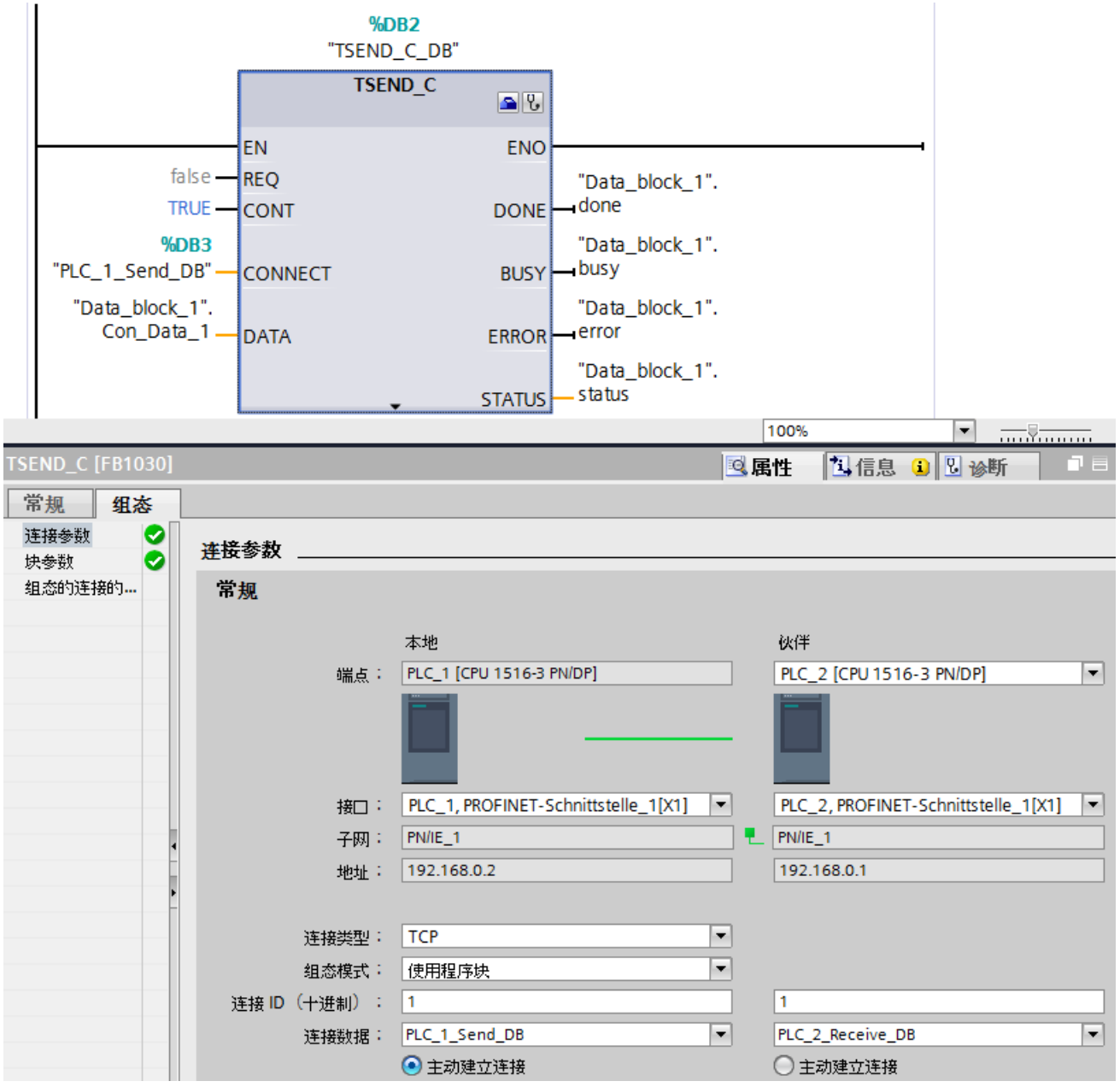


图 5-1 通过编程建立连接

建立组态连接

根据所选的 CPU 或软件控制器，可在 STEP 7 的"设备与网络"(Devices & networks) 编辑器中的网络视图内建立组态的连接。

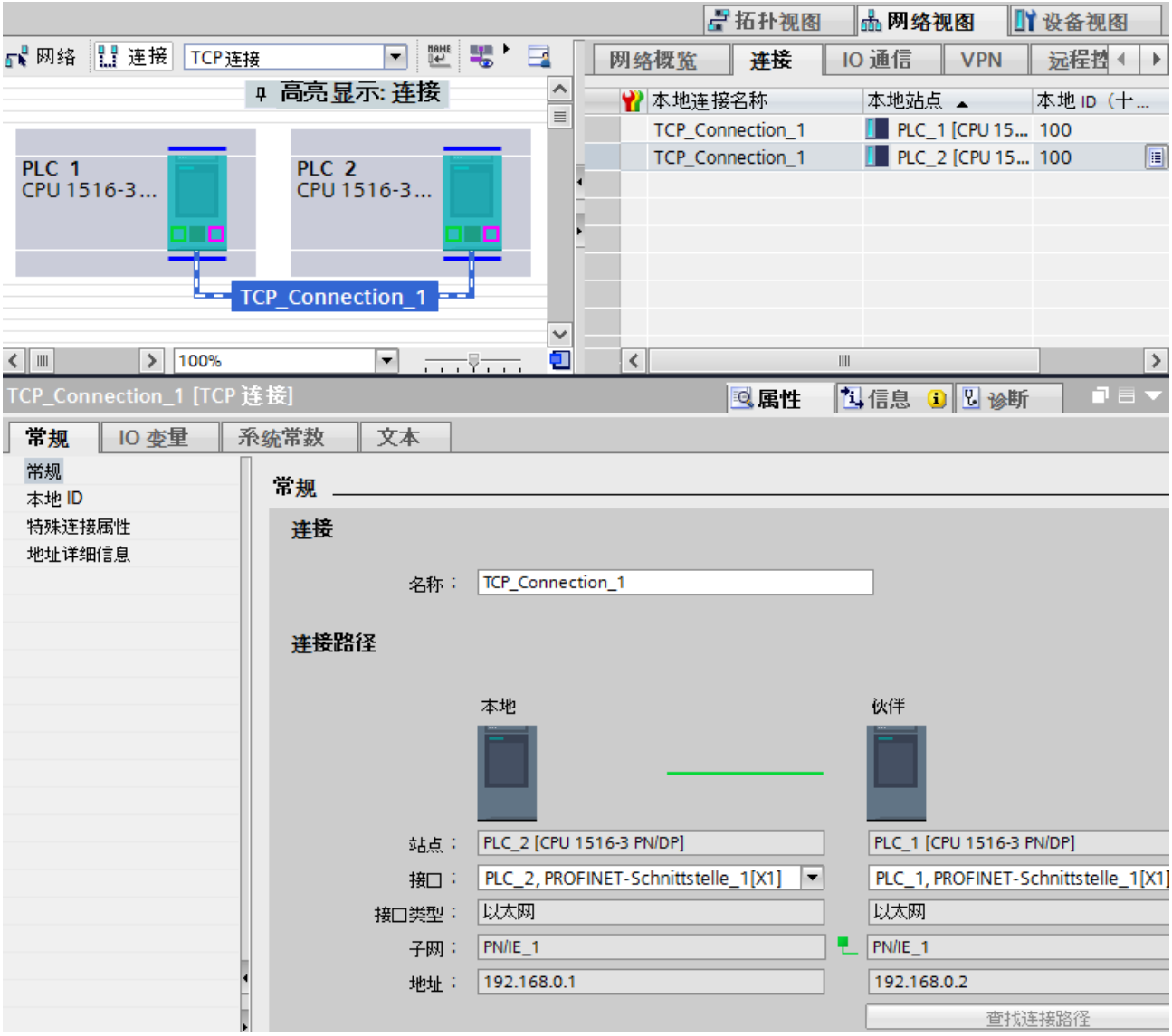


图 5-2 通过组态建立连接

对 CPU 连接资源的影响

通常，可以选择通过组态建立连接或者通过编程建立连接。如果选择通过编程建立连接，则将在数据传输结束后释放连接资源。与路由连接类似，编程的连接仍无法保证，也就是说，仅当资源可用时才会建立这类连接。建立组态的连接时，下载组态后资源处于可用状态，直至组态再次更改。因此，相应资源将预留，通过所组态的连接进行连接建立。在 CPU 巡视窗口中的"连接资源"(Connection resources) 表格中，简要列示了已使用的连接资源和仍然可使用的连接资源。

如何建立连接？

表格 5-5 建立连接

连接	自动连接	通过编程建立连接	通过组态建立连接
编程设备连接	√	-	-
HMI 连接	√	-	√
Web 通信	√	-	-
OPC UA 服务器通信	√	-	-
OPC UA 客户端通信	-	√	-
通过 TCP/IP 连接实现开放式通信	-	√	√
通过 ISO-on-TCP 连接实现开放式通信	-	√	√
通过 UDP 连接实现开放式通信	-	√	√
通过 ISO 连接实现开放式通信	-	√	√
通过 FDL 连接实现开放式通信	-	√	√
通过 Modbus TCP 连接进行通信	-	√	-
电子邮件连接	-	√	-
FTP 连接	-	√	-
S7 连接*	-	-	√

*请注意，对于 S7-1500 CPU，必须在 CPU 的属性中启用 PUT/GET 通信。有关该主题的更多信息，请参见 STEP 7 在线帮助。

更多信息

有关连接资源占用以及连接资源在 STEP 7 中显示的更多信息，请参见“连接资源 (页 392)”部分。

5.5 数据的一致性

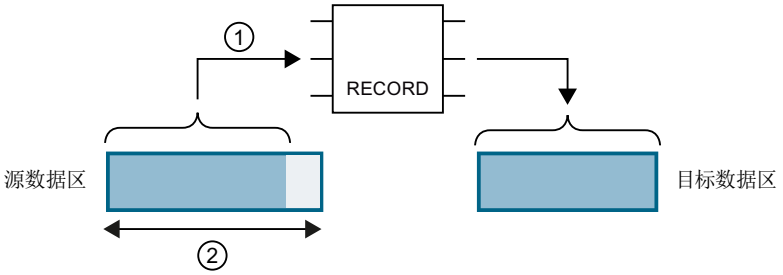
定义

在数据传输中，数据一致性至关重要。因此在组态通信任务时，必需注意。否则，可能导致故障发生。

同步运行中无法修改的数据区又称为一致性数据区。即，在超出一致性数据区所允许最大空间的连续数据区中，可同时包含新数据和旧数据。

一个通信指令中断时（如，由高优先级的硬件中断 OB 进行中断），将导致不一致现象。这会导致数据区域传输中断。如果 OB 中的用户程序对通信指令尚未处理的数据进行了更改，则每次传输的数据将不同：

下图显示的数据区空间小于一致性数据区允许的最大空间。此时，可确保进行数据访问时，用户程序不会中断数据区域的传输，从而有效避免了数据变更。

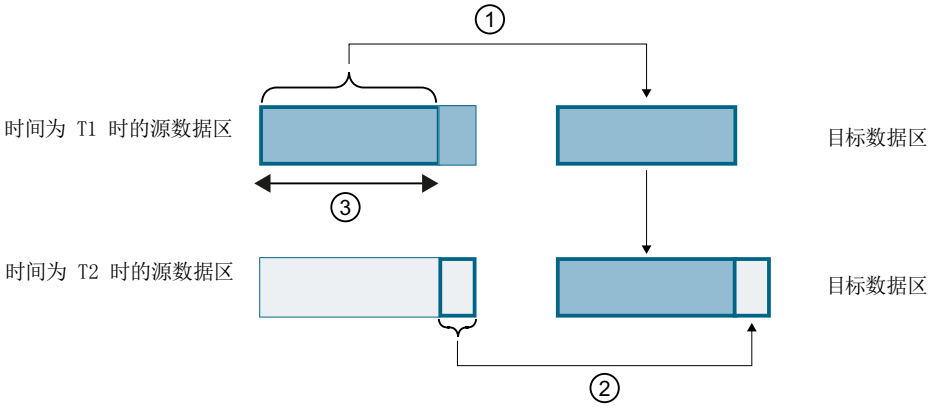


① 源数据区小于一致性数据区允许的最大空间 (②)。该指令将数据一同传输到目标数据区中。

② 一致性数据区的最大空间

图 5-3 数据的一致性传输

下图显示的数据区空间大于一致性数据区允许的最大空间。在这种情况下，数据会因传输中断而发生更改。将该数据区传输到多个地方时，也可能会发生传输中断。如果因传输中断而导致数据更改，则每次传输的数据将不同。



① 源数据区大于一致性数据区允许的最大空间 (③)。在 T1 时刻，该指令仅将源数据区中大小与一致性数据区相同的数据量传送到目标数据区中。

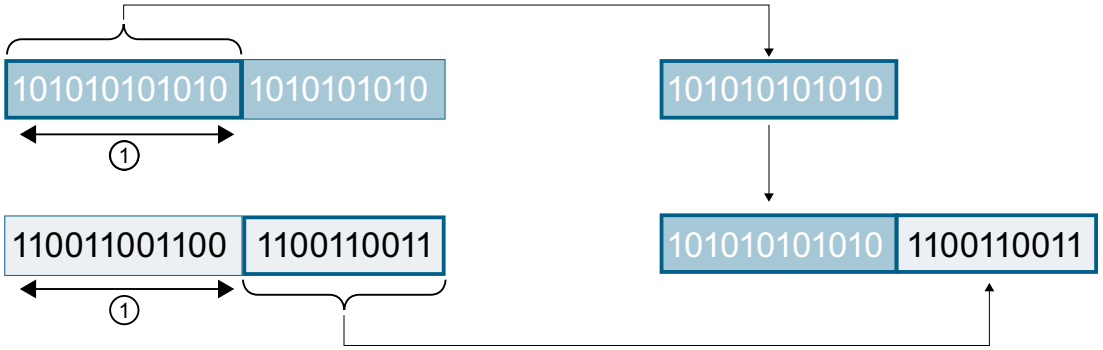
② 在 T2 时刻，该指令将源数据区中其余的数据传输到目标数据区中。传输完成时，目标数据区中包含不同时刻传输的数据。此时，如果源数据区中的数据发生变更，则会导致数据不一致。

③ 一致性数据区的最大空间

图 5-4 传输的数据量超出了一致性数据的最大数量

数据不一致的示例

下图举例说明了数据过程中数据的变更。目标数据区中包含不同时刻传输的数据。



① 一致性数据区的最大空间

图 5-5 示例：在数据传输过程中，数据发生变更

S7-1500 中，系统特定的一致性数据的最大数量：

如果遵循系统中所指定的一致性数据的最大数量，则不会产生不一致现象。在程序循环过程中，S7-1500 最多可将块中 512 个字节的通信数据一致性地复制到或传出用户存储器。超出该数据区时，将无法确保数据的一致性。如果要定义确保数据的一致性，则 CPU 内用户程序中的通信数据长度不能超过 512 个字节。之后，即可在 HMI 设备上通过 Read/Write 变量对这些数据进行一致性访问。

如果需一致性传输的数据量超出了系统指定的数据最大量，则需在应用程序中使用特殊措施确保数据的一致性。

确保数据一致性

通过指令访问公共数据：

如果在用户程序中通过一些通信指令访问公共数据（如 TSEND/TRCV），则可使用诸如“DONE”等参数对该数据区进行访问。因此，在用户程序中使用指令进行数据传输，可确保通信过程中数据区中数据的一致性。

说明

用户程序中采取的具体措施

要确保数据一致性，可将待传输数据复制到一个单独的数据区（如，全局数据块）中。用户程序继续传输源数据时，可通过通信指令将一致性地传输单独数据区中存储的数据。

在复制过程中，系统将使用相应的不可中断型指令，如 UMOVE_BLK 或 UFILL_BLK。这些指令可确保高达 16 KB 的数据一致性。

使用 PUT/GET 指令或通过 HMI 通信进行 Write/Read 操作：

使用 PUT/GET 指令进行 S7 通信或通过 HMI 通信进行 Write/Read 操作时，编程或组态中需考虑一致性数据区的大小。将 S7-1500 用作服务器时，用户程序没有可用于数据传输的指令。在用户程序运行过程中，可通过 PUT/GET 指令进行数据交换，对 S7-1500 进行更新。但在循环执行用户程序时，不支持对数据进行一致性传输。待传送数据区的长度应小于 512 个字节。

更多信息

- 有关通信模块所支持的一致性数据最大数量，请参见设备手册中的相应技术规范。
- 有关数据一致性的更多信息，请参见 STEP 7 在线帮助中的指令说明。

5.6 安全通信

5.6.1 安全通信的基础知识

5.6.1.1 有关安全通信的实用信息

在 STEP 7 (TIA Portal) V14 及更高版本和固件版本 V2.0 及更高版本的 S7-1500 CPU 中，设计了大量的安全通信选项。

"S7-1500 CPU"是指 S7-1500F、S7-1500T、S7-1500C 系列 CPU 和 S7-1500pro CPU 和 ET200SP CPU。

在后续版本中，其它组件也将支持安全通信（如 OUC 安全通信），详见下一部分。

在 S7-1200 CPU 固件版本 V4.4 及以上版本中，还支持安全通信。

要求

- 支持带有 DT TCON_IP_V4_SEC 或 SDT TCON_QDN_SEC 结构的连接描述 DB 的 CPU，包括以下 CPU：
  - S7-1200（固件版本 V4.4 及以上版本）
  - S7-1500（固件版本 V2.0 及以上版本）
- 也可通过以下 CP：
  - CP 1243-1（固件版本 V3.2 及以上版本）
  - CP 1243-8 IRC（固件版本 V3.2 及以上版本）
  - CP 1543-1（固件版本 V2.0 及以上版本）
  - CP 1545-1
  - CP 1543SP-1

  CP 1242-7 GPRS V2 不支持安全通信。

公钥基础结构 (PKI)

"安全"(secure) 属性用于识别以 Public Key Infrastructure (PKI) 为基础的通信机制（例如，RFC 5280，用于Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile）。Public Key Infrastructure (PKI) 是一个可签发、发布和检查数字证书的系统。PKI 通过签发的数字证书确保计算机通信安全。如果 PKI 采用非对称密钥加密机制，则可对网络中的消息进行数字签名和加密。

在 STEP 7 (TIA Portal) 中组态用于安全通信的组件，将使用一个非对称密钥加密机制，使用一个公钥 (Public Key) 和一个私钥 (Private Key) 进行加密。并使用 TLS (Transport Layer Security) 作为加密协议。TLS 是 SSL (Secure Sockets Layer) 协议的后继协议。

安全通信的目的

安全通信可用于实现以下目标：

- 机密性
  即，数据安全/窃听者无法读取。
- 完整性
  即，接收方接收到的消息与发送方发送的消息完全相同，未经更改。消息在传送过程中未经更改。
- 端点认证
  即，端点通信伙伴确实是声称为参与通信的本人。对伙伴方的身份进行检查。

在过去，这些目标通常仅与 IT 和计算机网络相关。但如今，包含有敏感数据的工业设备和控制系统也开始面临相同的信息安全高风险。这是因为，这些设备它们同样实现了网络互联，因而必须满足严格的数据交换安全要求。

在过去，往往会采用单元保护机制，通过防火墙或 VPN 连接保护自动化单元安全（如，使用安全模块），而如今同样如此。

但是，通过企业内部网或公共网络以加密形式将数据传送到外部计算机变得越来越重要。

安全通信的通用原则

无论采用何种机制，安全通信都基于 Public Key Infrastructure (PKI) 理念，包含以下组成部分：

- 非对称加密机制：
  - 使用公钥或私钥对消息进行加密/解密。
  - 验证消息和证书中的签名。

    发送方/认证机构通过自己的私钥对消息/证书进行签名。接收方/验证者使用发送方/认证机构的公钥对签名进行验证。
- 使用 X.509 证书传送和保存公钥。
  - X.509 证书是一种数字化签名数据，根据绑定的身份对公钥进行认证。
  - X.509 证书中还包含有公钥使用的详细说明或使用限制。例如，证书中公钥的生效日期和过期日期。
  - X.509 证书中还包含证书颁发方的安全相关信息。

在后续的章节中，将简要介绍在 STEP 7 (TIA Portal) 中管理证书和编写 secure Open User Communication (sOUC) 通信指令等所需的基本知识。

使用 STEP 7 进行安全通信：

在 STEP 7 V14 及其更高版本中，提供了安全通信的组态和操作所需的 PKI。

示例：

- 基于 TLS (Transport Layer Security) 协议，将 Hypertext Transfer Protokoll (HTTP) 转换成 Hypertext Transfer Protokoll Secure (HTTPS)。由于 HTTPS 中集成了 HTTP 和 TLS 协议，因此在相应的 RFC 中，又称为“HTTP over TLS”。在该浏览器中，可清楚地查看到所用的协议为 HTTPS：浏览器地址栏中 URL 为“https://”，而非“http://”。在大多数浏览器中，这类的安全连接将突出显示。
- 将 Open User Communication 转换为 secure Open User Communication。这种通信方式的底层协议同样为 TLS。
- 电子邮件服务提供商同样支持基于“Secure SMTP over TLS”协议进行访问，从而提高电子邮件通信的安全性。

下图显示了通信层中的 TLS 协议。

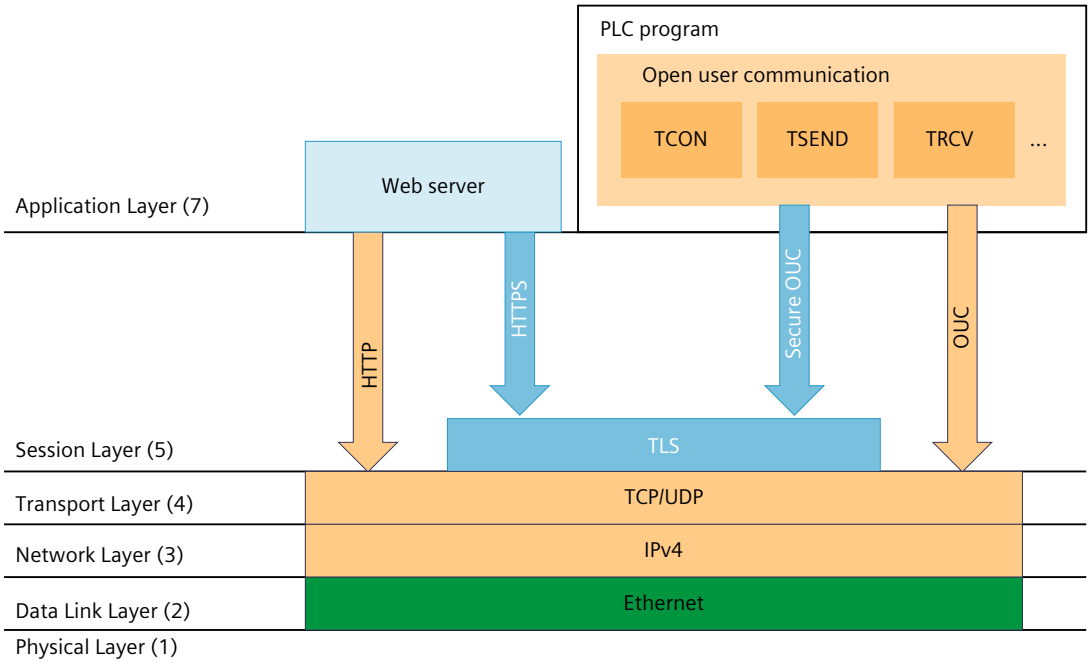


图 5-6 通信层中的 TLS 协议。

采用 OPC UA 的安全通信

固件版本 V2.0 及更高版本的 S7-1500 CPU 中，具有 OPC UA 服务器功能。OPC UA Security 中也涉及使用 X.509 数字证书进行认证、加密以及数据完整性检查，并且同样采用 Public Key Infrastructure (PKI)。根据应用的具体要求，端点安全可选择不同安全等级。我们将在一个单独章节中对 OPC UA 服务器功能进行介绍。

PG/HMI 间安全通信

在 V17 及以上版本中集成有最新型控制器和最新型 HMI 设备，TIA Portal、STEP 7 和 WinCC 的主要组件可实现创新型 PG/PC 和 HMI 标准安全通信（简称为 PG/HMI 通信）。

更多信息

有关 OPC UA 的更多信息，请参见"将 S7-1500 用作 OPC UA 服务器 (页 200)"部分。

有关安全编程设备/HMI 通信的更多信息，请参见"PG/HMI 间安全通信 (页 97)"部分。

5.6.1.2 设备相关的安全功能

传输层安全 (TLS) 是一种广泛使用的安全协议，可提高传输数据的安全性。对于自动化系统 S7-1500，TLS 用于以下基于证书的应用的安全通信：

- Web 服务器（HTTPS 协议框架）
- 安全的开放式用户通信 (OUC)，包括安全电子邮件（TMAIL_C 指令）
- PG/HMI 间安全通信

TLS 负责对所列应用的客户端和服务器之间的通信进行身份验证和加密并保证完整性，例如 CPU 的 Web 服务器和显示 CPU 的诊断网页的 Web 浏览器之间的通信。

OPC UA 服务器和 OPC UA 客户端应用实际上并不直接使用 TLS，但使用的加密过程类似。

TLS 不断发展进步，产生了各种 TLS 版本，这些版本在支持的密码套件（标准化加密方法集）和性能方面存在区别。

互联网工程任务组 (IETF) 负责 TLS 协议的描述。以下相关性适用：

- TLS 1.3 对应于 RFC 8446
- TLS 1.2 对应于 RFC 5246

此外，并非每个设备都支持 RFC 中定义的所有加密方法。因此，建立连接后，客户端和服务器协商一个双方都支持的方法 (Handshake) 以及要使用的参数。

支持的 TLS 版本 (S7-1500)

下表显示了给定 CPU 固件版本中支持的 TLS 版本。

CPU 固件版本	支持的 TLS 版本
V3.0	TLS 1.2、TLS 1.3
V2.9	TLS 1.2、TLS 1.3
V2.8 ... V2.0	TLS 1.2

创建证书时支持的加密方法和参数

要为新证书生成公钥，请在 TIA Portal 中设置加密方法和加密参数。这些证书参数与具体设备和所使用的应用程序相关

一种可能性：在 CPU 属性中，转到“保护和安全 > 证书管理器”(Protection & Security > Certificate manager) 并生成新的设备证书。可以在“生成证书”(Generate Certificates) 对话框的“证书参数”(Certificate Parameters) 下找到加密方法和加密参数的设置。

示例：RSA 2048 代表加密密钥长度为 2048 位的非对称 RSA 加密方法。

下表根据 CPU 应用或服务列出了支持的加密方法和加密参数。

加密方式/参数 S7-1500（固件 V3.0）	Web 服务器 (HTTPS) PG/HMI 间安全通信 OUC 安全连接	OPC UA
EC prime256v1	✓	-
EC secp384r1	✓	-
EC secp256k1	-	-
RSA 1024	✓	✓
RSA 2048	✓	✓
RSA 4096	✓	✓
RSA 8192	-	-

5.6.1.3 通过加密确保数据机密

消息加密是数据安全的一项重要措施。在通信过程中，即使加密的消息被第三方截获，这些潜在的侦听者也无法访问所获取的信息。

在进行消息加密时，采用了大量的数学处理机制（算法）。

所有算法都通过一个“密钥”参数，对消息进行加密和解密。

- 算法 + 密钥 + 消息 => 密文
- 密文 + 密钥 + 算法 =>（明文）消息

对称加密

对称加密的关键在于，两个通信伙伴都采用相同的密钥对消息进行加密和解密，如下图所示：Bob 使用的加密密钥与 Alice 使用的解密密钥相同。即，我们常说的双方共享一个密钥，可通过该密钥对消息进行加密和解密。

① Bob 采用对称密钥对消息进行加密

② Alice 采用对称密钥对加密后的消息进行解密

图 5-7 对称加密

该过程类似于一个公文箱，发送方和接收方使用同一把钥匙打开或锁上该公文箱。

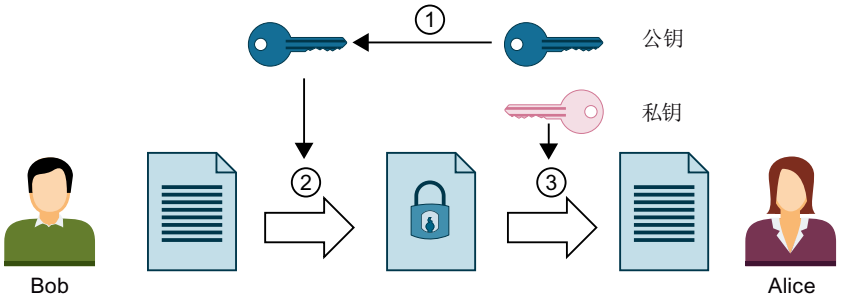
- 优势：对称加密算法（如，AES、Advanced Encryption Algorithm）的速度较快。
- 缺点：如何将密钥发送给接收方，而不会落到其他人手中？此为密钥分发问题。如果截获的消息数量足够大，则可推算出所用的密钥，因此必须定期更换。

如果通信伙伴比较多，则需分发的密钥数量巨大。

非对称加密

在非对称加密技术中使用一对密钥：一个公钥和一个私钥。与 PKI 一同使用时，又称为公钥加密系统，简称 PKI 加密系统。通信伙伴（下图中的 Alice）拥有一个私钥和一个公钥。公钥对所有人公开。即，任何通信伙伴都可以获得该公钥。拥有公钥的通信伙伴可对发送给 Alice 的消息进行加密。即下图中的 Bob。

Alice 的私钥为她自己所有而不公开，用于对发送给她的密文进行解密。



① Alice 将其公钥提供给 Bob。无需采取防范措施即可实现该过程：只要确定采用的是 Alice 的公钥，所有人都可以发消息给 Alice。

② Bob 使用 Alice 的公钥对消息进行加密。

③ Alice 使用私钥对 Bob 发送的密文进行解密。由于仅 Alice 拥有私有且未公开，因此只有她才能对该消息进行解密。通过私钥，Alice 可以对使用她所提供的公钥加密的消息进行解密，而不仅仅只是 Bob 的消息。

图 5-8 非对称加密

该系统与邮箱类似，所有人都可以向邮箱发送消息，但只有拥有密钥的人才能删除这些消息。

- 优势：使用公钥加密的消息，仅私钥拥有者才能进行解密。由于在解密时需要使用另一密钥（私钥），而且加密的消息数量庞大，因此很难推算出解密密钥。这意味着，公钥无需保持机密性，而这与对称密钥不同。

  另一大优点在于，公钥的发布更为方便快捷。在非对称密钥系统中，接收方将公钥发送到发送方（消息加密方）时无需建立专用的安全通道。与对称加密过程相比，密钥管理工作量相对较少。
- 缺点：算法复杂（如，RSA，以三位数学家 Rivest、Shamir 和 Adleman 的名字的首字母命名），因此性能低于对称加密机制。

实际通信中的加密过程

在实际通信过程中（如，与 CPU Web 服务器通信和开发式用户安全通信），通常在相关的应用层之后使用 TLS 协议。例如，应用层采用的协议为 HTTP 或 SMTP，详细信息见前文所述。

例如，TLS (Transport Layer Security) 混合采用非对称加密和对称加密（混合加密）机制确保数据通过 Internet 进行安全传输，并支持以下子协议：

- TLS Handshake Protocol，对通信伙伴进行身份验证，并在非对称加密的基础上对数据传输所需的算法和密钥进行协商
- TLS Record Protocol 采用对称加密机制对用户数据加密以及进行数据交换。

无论是非对称加密还是对称加密，这两种数据安全加密机制在安全性方面没有明显差异。数据安全等级取决于设置的参数，如所选密钥的长度等等。

加密使用不当

通过位串，无法指定公钥的身份。欺瞒者可使用他们自己的公钥声明为其他人。如果第三方使用该公钥将其认作是指定的通信伙伴，则将导致机密信息被窃取。之后，欺瞒者再使用自己的密钥对这些本消息进行解密，虽然这些消息本不应发送给他们。最终，导致敏感信息泄露，落入他人之手。

为了有效预防此类错误的发生，该通信伙伴必须确信与正确的通信伙伴进行数据通信。此类信任关系是通过 PKI 中的数字证书建立的。

5.6.1.4 通过签名确保数据的真实性和完整性

由能够截获服务器与客户端之间的通信并将自身伪装成客户端或服务器的程序实施的攻击称为中间人攻击。如果未能检测到这些程序的真实身份，则将造成诸如 S7 程序、CPU 中设定值等重要信息泄漏，进而导致设备或工厂遭受攻击。可使用数字证书避免此类攻击。

在安全通信过程中，所用的数字证书符合 International Telecommunication Union (ITU) 的 X.509 标准。该证书用于检查（认证）程序、计算机或组织机构的身份。

如何通过证书建立信任关系

X.509 证书主要用于将带有证书的数据身份（如，电子邮件地址或计算机名称）与公钥中的身份绑定在一起。身份可以是个人、计算机，也可以是机器设备。

证书由证书颁发机构（Certificate Authority，CA）或证书主体签发。而 PKI 系统则指定了用户信任证书颁发机构及其所签发证书的规则。

证书认证过程：

1. 要获取一份证书，需要向与证书颁发机构相关联的注册机构提交一份证书申请。
2. 证书颁发机构将基于既定标准对该申请和申请人进行评估。
3. 如果可以清晰识别申请人的身份，则证书颁发机构将签发一份已签名的证书进行确认。申请人现成为证书主体。

在下图中，对这一过程进行了简要说明。但不涉及 Alice 对该数字签名的检查过程。

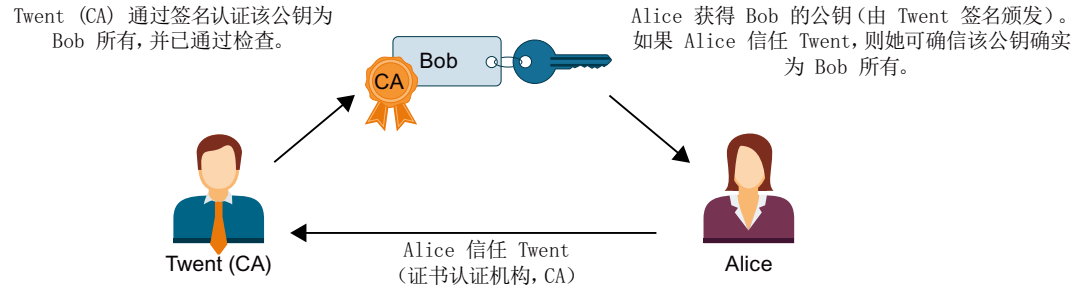


图 5-9 由证书颁发机构对证书进行签名

自签名证书

自签名证书指，由证书主体而非独立的证书颁发机构签名的证书。

示例：

- 用户也可以自己创建证书并签名，对发送给通信伙伴的消息进行加密。在上述示例中，Bob（而非 Twent）可以使用私钥对自己的证书进行签名。之后，Alice 将使用 Bob 的公钥检查该签名是否与 Bob 的公钥相匹配。该过程可用于简单的工厂内部数据加密通信。
- 例如，根证书是一种由证书颁发机构 (CA) 签署的自签名证书，其中包含证书颁发机构的公钥。

自签名证书的特性

自签名证书的证书主体“CN”(Common Name of Subject) 和“Issuer”属性相同：用户已完成对证书的签名。字段“CA” (Certificate Autority) 需设置为“False”；自签名证书不得用于对其它证书进行签名。

自签名证书未包含在 PKI 系统中。

证书内容

符合 X.509 V3 标准（同样用于 STEP 7 和 S7-1500 CPU）要求的证书通常包含以下元素：

- 公钥
- 证书主体（即，密钥持有者）的详细信息。如，Common Name (CN) of Subject 。
- 各种属性，如序列号和有效期等等
- 证书颁发机构 (CA) 的数字签名，用于证实信息的正确性。

除此之外，还包含以下扩展详细：

- 指定公钥的使用范围(Key Usage)，如签名或密钥加密。
  在开放式用户安全通信中，使用 STEP 7 创建一个新证书时，可从用途列表中选择相应的条目，如“TLS”。
- 指定 Subject Alternative Name (SAN)，用于与 Web 服务器进行安全通信 (HTTP over TLS)，以确保 Web 浏览器地址栏中的证书同样属于该 URL 所指定的 Web 服务器。

如何生成并验证签名

非对称密钥可用于证书的验证：在“MyCert”证书示例中，介绍了具体的“签名”与“验证签名”过程。

生成签名：

1. “MyCert”证书的签发者使用一个特定的哈希函数（例如，SHA-1，Secure Hash Algorithm），根据证书数据生成一个哈希值。

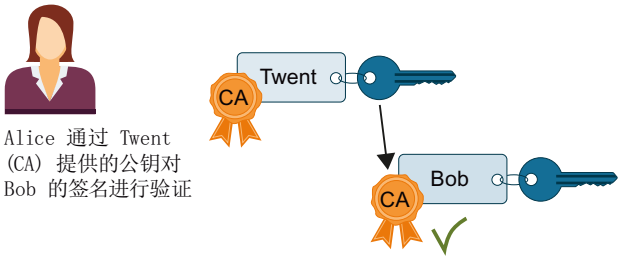
   该 HASH 值是一个长度固定的位串。HASH 值长度固定的优势在于，签名的时间始终相同。

2. 之后，证书的签发者再使用由这种方式生成的 HASH 值和私钥，生成一个数字签名。通常采用 RSA 签名机制。
3. 数字签名将保存在证书中。此时，证书已签名。

验证一个签名：

1. “MyCert”证书的认证方将获得签发者签发的证书和公钥。
2. 使用签名时所用的哈希算法（例如，SHA-1），根据证书数据生成一个新的哈希值。
3. 最后，再将由证书签发者公钥确定的 HASH 值与签名算法进行比较，对签名进行检查。
4. 如果签名通过检查，则表示证书主体的身份以及完整性（即，证书内容的可靠性和真实性）均通过验证。拥有该公钥（即，证书颁发机构的证书）的任何人均可对该签名进行检查，并确认该证书确实由该证书颁发机构签发。

下图显示了 Alice 如何采用 Twent（代表证书颁发机构，CA）证书中的公钥，验证 Bob 的公钥签名。因此，在验证时仅需检查证书颁发机构所颁发证书的可用性。验证会在 TLS 会话中自动执行。



Alice 通过 Twent (CA) 提供的公钥对 Bob 的签名进行验证

图 5-10 使用证书颁发机构的证书公钥对证书进行验证

签名消息

上文中介绍的证书签名与验证机制，同样使用 TLS 会话对消息进行签名和验证：

如果发送方基于一条消息生成一个 HASH 值并使用私钥进行加密，之后在添加到原消息中，则消息接收方即可对消息的完整性进行检查。接收方使用发送方的公钥对该 HASH 值进行解密，并将其与所收到消息中的 HASH 进行比较。如果这两个值不同，则表示该消息或加密的 HASH 值在传送过程中被篡改。

Root 证书的证书链

PKI 证书通常按层级进行组织：层级顶部由根证书构成。Root 证书并非由上一级证书颁发机构签名。Root 证书的证书主体与证书的签发者相同。根证书享受绝对信任。它们构成了信任"点"，因此可作为接收方的可信证书。此类证书存储在专门存储受信证书的区域。

基于该 PKI，Root 证书可用于对下级证书颁发机构颁发的证书（即，所谓的中间证书）进行签名。从而实现从 Root 根证书到中间证书信任关系的传递。由于中间证书可对诸如 Root 证书之类的证书进行签名，因此这两种证书均称为"CA 证书"

这种证书签名层级可通过多个中间证书进行延伸，直至最底层的实体证书。最终实体证书即为待识别用户的证书。

验证过程则反向贯穿整个层级结构：综上所述，先通过签发者的公钥确定证书签发者并对其签名进行检查，之后再沿着整条信任链确定上一级证书签发者的证书，直至到达根证书。

结论：无论组态何种安全通信类型，每台设备中都必需包含一条到 Root 证书的中间证书链（即证书路径），对通信伙伴的最低层实体证书进行验证。

5.6.2 管理证书

5.6.2.1 证书管理的必备知识

本节介绍了根据所使用的服务（CPU 应用程序）以及 TIA Portal/CPU 固件的版本提供的相应 S7-1500 CPU 证书管理选项。

证书管理选项概述

对于 TIA Portal V14 和 CPU 固件版本 V2.0 及更高版本，可以在 TIA Portal 中管理 S7-1500 CPU 的不同服务之间实现安全通信的证书并将其下载到 CPU 中。

对于 TIA Portal V17 以及 S7-1500 CPU 固件版本 V2.9 及更高版本，支持另一种证书管理方式：使用 GDS 推送方法，可以在 CPU 运行期间传输或更新证书，而无需重新下载 CPU。

采用同一方式，自 TIA Portal V18 起，还可以传输 S7-1500 CPU（自固件 V3.0 起）的 Web 服务器证书。

下表概述了与所使用的服务以及 TIA Portal 固件版本或 CPU 固件版本相关的证书管理选项。

服务	使用 TIA Portal 进行证书管理（TIA Portal 版本/S7-1500 CPU 固件版本）	使用 OPC UA GDS 推送方法进行证书管理（TIA Portal 版本/S7-1500 CPU 固件版本）
Web 服务器	自 V14 起/自 V2.0 起	自 V18 起/自 V3.0 起
安全 OUC 通信	自 V14 起/自 V2.0 起	-
OPC UA 服务器	自 V14 起/自 V2.0 起	自 V17 起/自 V2.9 起
OPC UA 客户端	自 V15.1 起/自 V2.6 起	-
安全 PG/HMI 通信	自 V17 起/自 V2.9 起	-

更多信息

单击此处了解使用 GDS 推送方法进行证书管理的说明：通过全球发现服务器 (GDS) 实现证书管理 (页 181)。

如果不需要加载新证书，但需要更新证书（例如因为有效期已过期），则在 RUN 下也可以执行此操作，但需遵守以下条件：请参见“提示：在 RUN 模式中更新下载的证书 (页 67)”。

5.6.2.2 使用 TIA Portal 进行证书管理

STEP 7 V14 及更高版本与 S7-1500-CPU 固件版本 V2.0 及更高版本一同使用时，支持 Internet PKI (RFC 5280)。因此，S7-1500 CPU 可与同样支持 Internet PKI 的设备进行数据通信。

如，可使用 X.509 证书验证上文中所介绍的证书。

STEP 7 V14 及更高版本采用的 PKI 与 Internet PKI 类似。例如，证书吊销列表 (CRL) 不受支持。

在 TIA Portal 中创建或分配证书

对于具有安全特性的设备（如，S7-1500 CPU 固件版本 V2.0 及以上版本），可在 STEP 7 中根据不同应用创建特定的证书。

在 CPU 巡视窗口的以下区域内，可创建新的证书或选择现有的证书：

- “Web 服务器 > 安全”(Web server > Security) - 用于生成和分配 Web 服务器证书。
- “保护和安全 > 连接机制”(Protection & Security > Connection mechanisms) - 用于生成或分配 PLC 通信证书（TIA Portal V17 及以上版本的 PG/HMI 间安全通信）。
- “保护和安全 > 证书管理器”(Protection & Security > Certificate manager)- 用于生成和分配所有类型的证书。生成证书时，将预设开放式用户安全通信的 TLS 证书。
- “OPC UA > 服务器 > 安全”(OPC UA > Server > Security) - 用于生成或分配 OPC UA 服务器证书。

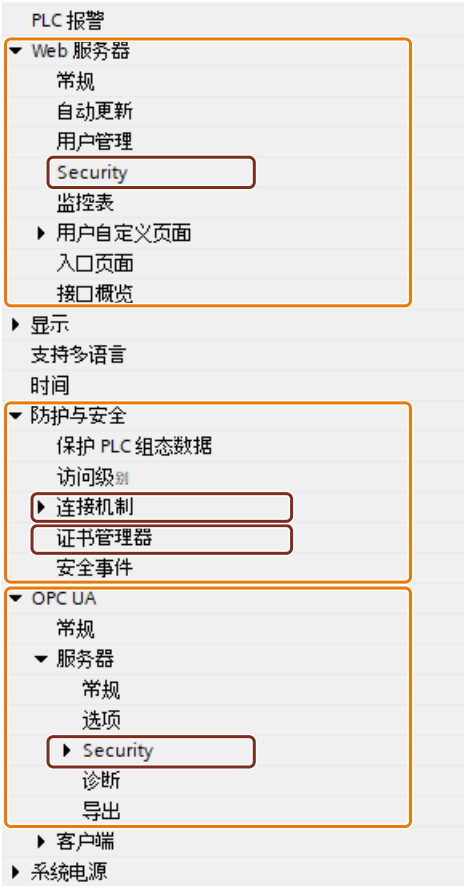


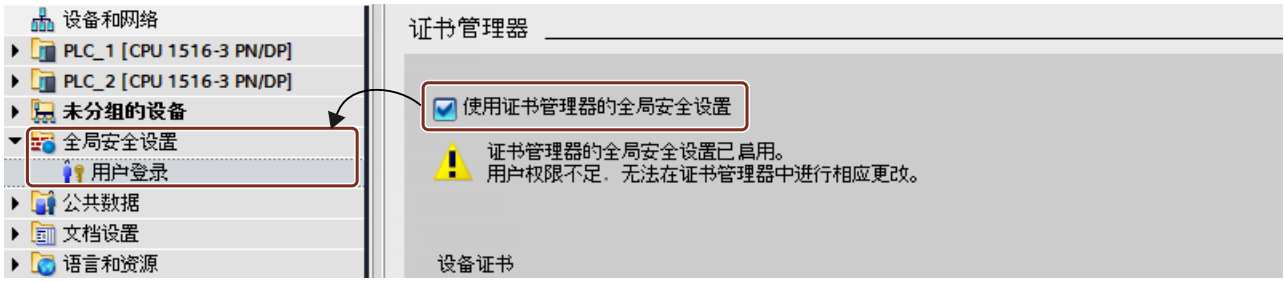
图 5-11 STEP 7 中 S7-1500 CPU 的安全设置

“保护与安全 > 证书管理器”区域的特性

在巡视窗口中，只有该区域内才能进行全局（即，项目级）和局部（即，设备特定）证书管理器切换（选项“使用证书管理器的全局安全设置”(Use global security settings for certificate manager)）。该选项确定了您是否有权访问项目中的所有证书。

- 如果在全局安全设置中未使用证书管理器，则只能访问 CPU 的局部证书存储器。例如，无法访问所导入的证书或 Root 证书。如果没有这些证书，则可用功能将受到限制。例如，只能生成自签名证书。
- 如果在全局安全设置中使用证书管理器并以管理员身份登录，则有权访问全局（项目级）证书存储器。例如，可为 CPU 分配所导入的证书，也可创建由项目 CA（项目的证书颁发机构）签发与签名的证书。

下图显示了在 CPU 的巡视窗中激活“使用证书管理器的全局安全设置”(Use global security settings for the certificate manager) 选项后，项目树中的“全局安全设置”(Global security settings) 显示。



双击项目树中全局安全设置下的“用户登录”(User login) 并进行登录时，则将显示“证书管理器”(Certificate manager) 行。

双击“证书管理器”(Certificate manager) 行，则可访问项目中的所有证书。这些证书分别位于选项卡“CA”（证书颁发机构）、“设备证书”(Device certificates) 和“可信证书与 Root 证书颁发机构”(Trusted certificates and root certificate authorities) 内。

私钥

生成设备证书和服务器证书（最终实体证书）时，STEP 7 将生成私钥。私钥的加密存储的位置，取决于证书管理器中是否使用全局安全设置：

- 如果使用全局安全设置，则私钥将以加密形式存储在全局（项目级）证书存储器中。
- 如果未使用全局安全设置，则私钥将以加密形式在局部（CPU 特定的）证书存储器中。

解密数据时所需的私钥将显示在全局安全设置中证书管理器中“设备证书”(Device certificates) 选项卡的“私钥”(Private key) 列中。

下载硬件配置时，同时会将设备证书、公钥和私钥下载到 CPU 中。

注意
启用“使用证书管理器的全局安全设置”(Use global security settings for the certificate manager) 选项 - 后果 “使用证书管理器的全局安全设置”(Use global security settings for certificate manager) 选项会影响之前所用的私钥：如果创建证书时未使用证书管理器中的全局安全设置，而且更改了使用该证书管理器的选项，则将导致私钥丢失且证书 ID 发生变更。系统会发出警告，提示您注意这种情况。因此，在开始组态项目时，需指定证书管理器选项。

5.6.2.3 证书管理示例。

如前文所述，每种类型的安全通信都需要使用证书。在以下章节中，将举例说明如何通过 STEP 7 进行证书管理，以满足开放式用户安全通信的要求。

不同通信伙伴所用的设备往往不同。为各个通信伙伴提供所需证书的相应操作步骤也各不相同。通常需使用 S7-1500 CPU 或 S7-1500 软件控制器，固件版本 V2.0 及以上版本。

基本规则为：

建立安全连接（“握手”）时，通信伙伴通常仅传送最终实体证书（设备证书）。

因此，验证已传送设备证书所需的 CA 证书必须位于相应通信伙伴的证书存储器中。

说明

在 CPU 中，需设置当前的日期/时间。

使用安全通信（如，HTTPS、安全 OUC、OPC UA）时，需确保相应模块为当前时间和当前日期。否则，模块会将所用的证书评估为无效，且无法进行安全通信。

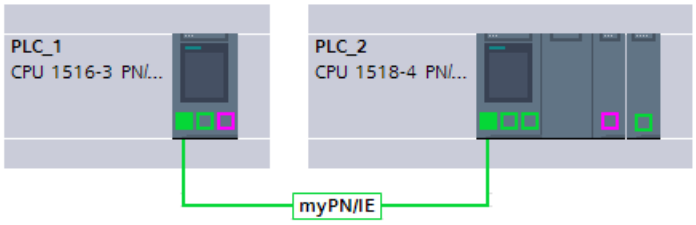
两个 S7-1500 CPU 之间的开放式用户安全通信

两个 S7-1500 CPU（PLC_1 和 PLC_2）之间通过开放式用户安全通信进行数据交换。

使用 STEP 7 生成所需的设备证书，然后将其分配给 CPU，如下所述。

STEP 7 项目证书颁发机构（项目的 CA）用于对设备证书进行签名。

在用户程序中根据证书 ID 对证书进行引用（TCON 通信指令组合相关的系统数据类型，例如 TCON_IPV4_SEC）。在生成或创建证书时，STEP 7 将自动分配证书 ID。



操作步骤

STEP 7 自动将所需的 CA 证书与硬件配置一同加载到通信伙伴的 CPU 中，确保两个 CPU 中满足证书验证需求。因此，用户只需生成相应 CPU 的设备证书，其余操作将由 STEP 7 完成。

1. 在“保护和安全”(Protection & Security) 区域中，标记 PLC_1 并激活“使用证书管理器的全局安全设置”(Use global security settings for certificate manager) 选项。
2. 在项目树的“全局安全设置”(Global security settings) 区域中，以 user 身份进行登录。对于新项目，首次登录时的身份为“Administrator”。
3. 返回“保护与安全”(Protection & Security) 区域的 PLC-1 中。在“设备证书”(Device certificates) 表格中，单击“证书主体”(Certificate subject) 列的一个空行，添加新的证书。
4. 在下拉列表中，选择一个证书并单击“添加”(Add) 按钮。

   “创建证书”(Create Certificate) 对话框随即打开。
5. 保留该对话框中的默认设置。这些设置专用于开放式用户安全通信（用途：TLS）。

   提示：补充证书主体的默认名称（此时，为 CPU 名称。为了便于区分，需管理大量设备证书时，建议保留系统默认的 CPU 名称。

   示例：PLC_1/TLS 变为 PLC_1-SecOUC-Chassis17FactoryState。
6. 编译组态。

   设备证书和 CA 证书是组态的一部分。
7. 对于 PLC_2，重复以上操作步骤。

在下一个操作步骤中，需创建用户程序进行数据交换，并加载组态和该程序。

使用自签名证书而非 CA 证书

创建设备证书时，可选择“自签名”(Self-signed) 选项。即使在未登录，也可创建自签名证书进行全局安全设置。但不建议执行该操作。这是因为，采用这种方式创建的证书不会保存在全局证书存储器中，也无法直接分配给伙伴 CPU。

如上文所述，选择证书的主体名称时需小心谨慎，以确保为设备指定的证书正确无误。

对于自签名证书，无法通过 STEP 7 项目的 CA 证书进行验证。要确保自签名证书可通过验证，需要将通信伙伴的自签名证书加入每个 CPU 的可信伙伴设备列表中。为此，必须激活选项“使用证书管理器的全局安全设置”(Use global security settings for certificate manager)，并以 user 身份登录全局安全设置。

要将通信伙伴的自签名证书添加到 CPU 中，请按以下步骤操作：

1. 选择 PLC_1，并导航到“保护与安全”(Protection & Security) 区域中的“伙伴设备证书”(Certificates of partner devices) 表格处。
2. 在“设备证书”(Device certificates) 表格中，单击“证书主体”(Certificate subject) 列的一个空行，添加新的证书。
3. 在下拉列表中选择该通信伙伴的自签名证书，并进行确认。

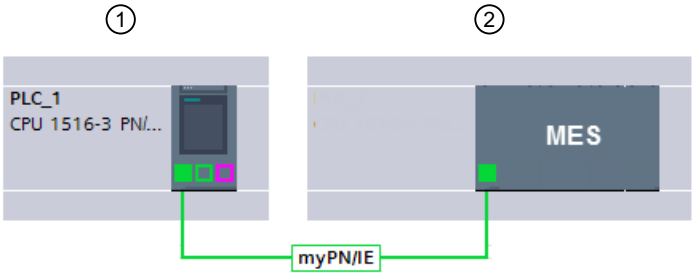
在下一个操作步骤中，需创建用户程序进行数据交换，并加载组态和该程序。

S7-1500 CPU（作为 TLS 客户端）与外部设备（作为 TLS 服务器）之间的开放式用户安全通信

两个设备将通过 TLS 连接或 TLS 会话进行数据交换（如，配方、生产数据或质量数据）：

- S7-1500-CPU (PLC_1) 作为 TLS 客户端；该 CPU 采用开放式用户安全通信
- 外部设备（如，制造执行系统 (MES)）作为 TLS 服务器

S7-1500 CPU 作为 TLS 客户端，与 MES 系统建立 TLS 连接/会话。



① TLS 客户端

② TLS 服务器

验证 TLS 服务器时，S7-1500 CPU 需要具有 MES 系统的 CA 证书：用于验证证书路径的 Root 证书和中间证书（如果适用）。

需要将这些证书导入 S7-1500 CPU 的全局证书存储器中。

要导入通信伙伴的证书，请按照以下步骤进行操作：

1. 打开项目树中全局安全设置下的证书管理器。
2. 选择待导入证书的相应表格（可信证书和 Root 证书颁发机构）。
3. 右键单击该表，打开快捷菜单。单击“导入”(Import)，导入所需证书或所需 CA 证书。
   导入证书时，系统将为该证书指定一个证书 ID，并在下一步操作中将其指定给一个模块。
4. 选择 PLC_1，并导航到“保护与安全”(Protection & Security) 区域中的“伙伴设备证书”(Certificates of partner devices) 表格处。
5. 单击“证书主体”(Certificate subject) 列中的空行，添加所导入的证书。
6. 在下拉列表中选择该通信伙伴所需的 CA 证书，并进行确认。

MES 系统还需要提供 CPU 的设备证书，用于对该 CPU 进行验证（即，TLS 客户端）。此时，MES 系统中应包含该 CPU 的 CA 证书。如果要将证书导入 MES 系统，则需先从 CPU 的 STEP 7 项目中导出该 CA 证书。请按以下步骤操作：

1. 打开项目树中全局安全设置下的证书管理器。
2. 选择待导出证书的匹配表（CA 证书）。
3. 右键单击所选择的证书，打开快捷菜单。
4. 单击“导出”(Export)。
5. 选择证书的导出格式。

在下一个操作步骤中，需创建用户程序进行数据交换，并加载组态和该程序。

S7-1500 CPU（作为 TLS 服务器）和外部设备（作为 TLS 客户端）之间安全的开放式用户通信

如果将 S7-1500 CPU 用作 TLS 服务器，并且外部设备（如，ERP 系统（企业资源规划系统））建立了 TLS 连接/会话，则需要具有以下证书：

- 对于 S7-1500 CPU，需使用私钥生成一个设备证书（服务器证书），并随硬件配置一同下载到 S7-1500 CPU 中。生成服务器证书时，需使用选项“由证书颁发机构签名”(Signed by certificate authority)。
  密钥交换需要使用私钥，如示例“基于 TLS 的 HTTP”的图所示。
- 对于 ERP 系统，需先导出 STEP 7 项目中的 CA 证书，然后再将其导入/加载到 ERP 系统中。基于 CA 证书，ERP 系统在建立 TLS 连接/会话时将检查从 CPU 传送到 ERP 系统的 S7-1500 服务器证书。

① TLS 服务器
② TLS 客户端

图 5-12 S7-1500 CPU 与 ERP 系统间的 OUC 安全通信

相关操作步骤，请参见上文介绍。

与邮件服务器进行开放式用户安全通信 (SMTP over TLS)

S7-1500 CPU 可使用通信指令 TMAIL-C 与邮件服务器建立安全连接。

系统数据类型 TMail_V4_SEC 和 TMail_QDN_SEC 可确定电子邮件服务器的伙伴端口，并通过“SMTP over TLS”协议访问电子邮件服务器。

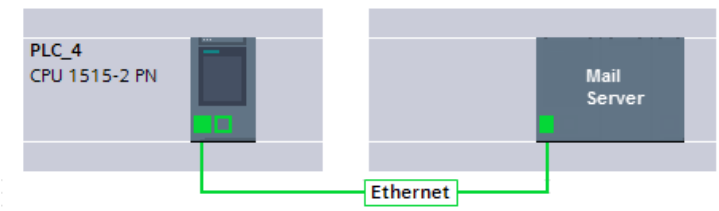


图 5-13 S7-1500 CPU 与邮件服务器间的 OUC 安全通信

要建立安全的邮件连接，则需将电子邮件服务器（提供方）的根证书和中间证书导入 S7-1500 CPU 的全局证书存储器中。基于这些证书，CPU 在建立 TLS 连接 / 会话时将检查由邮件服务器发送的服务器证书。

要导入邮件服务器的证书，请按以下步骤操作：

1. 打开项目树中全局安全设置下的证书管理器。
2. 选择待导入证书的相应表格（可信证书和 Root 证书颁发机构）。
3. 右键单击该表，打开快捷菜单。单击“导入”(Import)，导入所需证书或所需 CA 证书。
   导入证书后，系统将为该证书指定一个证书 ID，并在下一步操作中将其指定给一个模块。
4. 选择 PLC_1，并导航到“保护与安全”(Protection & Security) 区域中的“伙伴设备证书”(Certificates of partner devices) 表格处。
5. 单击“证书主体”(Certificate subject) 列中的空行，添加所导入的证书。
6. 在下拉列表中选择该通信伙伴所需的 CA 证书，并进行确认。

在下一个操作步骤中，需创建该 CPU 中电子邮件客户端功能的用户程序，并加载组态与该程序。

5.6.2.4 证书通信原理：基于 TLS 的 HTTP

下图显示了如何使用后以下机制在 S7-1500 CPU 的 Web 浏览器和 Web 服务器之间建立安全通信。

首先需要在 STEP 7 中更改“仅允许 HTTPS 访问”(Permit access only through HTTPS) 选项。在 STEP 7 V14 及以上版本中，可能会影响 S7-1500 CPU（固件版本 V2.0 及以上版本）中 Web 服务器的服务器证书：服务器证书将在 STEP 7 的以上版本及更改版本中生成。

此外，在该示例中还显示了 PC 的 Web 浏览器端如何基于加密的 HTTPS 连接所调用 CPU 的 Web 服务器网站。

S7-1500 CPU（固件版本 V2.0 及以上版本）中 Web 服务器证书的应用

对于固件版本 V2.0 及以下版本的 S7-1500 CPU，设置 Web 服务器属性时，如果无特殊要求，需设置为“只允许通过 HTTPS 访问”(Permit access only with HTTPS)。

对于此类 CPU，无需进行证书处理；CPU 将自动为 Web 服务器生成所需证书。

对于固件版本 V2.0 及更高版本的 S7-1500 CPU，STEP 7 会为 CPU 生成服务器证书（最终实体证书）。在 CPU 的属性中为 Web 服务器分配服务器证书（“Web 服务器 > 安全”(Web server > Security)）。

由于服务器证书名称通常为系统预设，因此无需任何更改即可轻松完成 Web 服务器的组态：激活 Web 服务器。默认启用“仅允许 HTTPS 访问”(Permit access only with HTTPS) 选项，STEP 7 将在编译过程中使用默认名称生成服务器证书。

无论您是否在全局安全设置中使用证书管理器：STEP 7 中包含生成服务器证书所需的全部信息。

此外，还需确定服务器证书的相关特性。如，名称或有效期等。

说明

在 CPU 中，需设置当前的日期/时间。

使用安全通信（如，HTTPS、安全 OUC、OPC UA）时，需确保相应模块为当前时间和当前日期。否则，模块会将所用的证书评估为无效，且无法进行安全通信。

加载 Web 服务器证书

加载硬件配置时，系统将自动加载 STEP 7 生成的服务器证书。

- 如果在全局安全设置中使用证书管理器，则项目的证书颁发机构（CA 证书）对 Web 服务器的服务器证书进行签名。在加载过程中，项目的 CA 证书也将自动加载。
- 如果未在全局安全设置中使用证书管理器，则 STEP 7 会生成服务器证书作为自签名证书。

通过 CPU 的 IP 地址对 CPU 的 Web 服务器进行寻址时，每次 CPU 中以太网接口的 IP 地址发生更改时，都必须生成新的服务器证书并加载（最终实体证书）。这是由于 CPU 的身份随 IP 地址一同更改。根据 PKI 规则，该身份必须进行签名。

如果使用域名（如，“myconveyer-cpu.room13.myfactory.com”）而非 IP 地址对 CPU 进行寻址，则可避免这一问题。为此，需通过 DNS 服务器对该 CPU 的域名进行管理。

为 Web 浏览器提供一份 Web 服务器的 CA 证书

在 Web 浏览器中，通过 HTTPS 访问 CPU 网站时，需安装该 CPU 的 CA 证书。如果未安装证书，则将显示一条警告消息，不建议访问该页面。要查看该页面，需显式“添加例外情况”。

有效的 Root 证书，可从 CPU Web 服务器"简介"(Intro) Web 页面的"下载证书"(Download certificate) 中下载。

在 STEP 7 中，可采用另一种方式：使用证书管理器，将项目的 CA 证书导出到 STEP 7 中的全局安全设置中。之后，再将 CA 证书导入浏览器中。

安全通信的过程

下图简要说明了通信的建立方式（"握手"），并着重介绍了通过 HTTP over TLS 进行数据交换时所用的密钥协商过程。

该过程可适用于基于 TLS 的所有通信方式。即，也可适用于开放式用户安全通信（请参见"安全通信的基本知识"）。

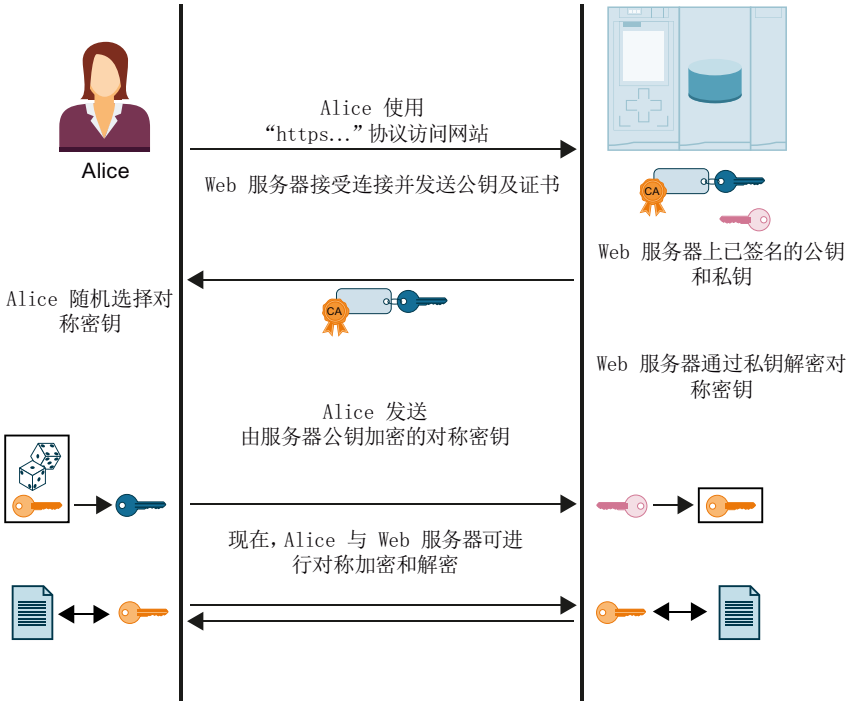


图 5-14 基于 HTTPS 的通信握手

在本示例图中并不涉及 Alice 端（浏览器端）对 Web 服务器所发送证书的验证措施。Alice 是否信任收到的 Web 服务器证书、信任该 Web 服务器的身份并接受数据交换，具体取决于验证结果。

验证 Web 服务器可靠性的操作步骤如下所示：

1. Alice 必须获得所有相关颁发机构的公钥。即，必须拥有整个证书链，才能对该 Web 服务器证书（即，Web 服务器的最终实体证书）进行验证。

   Alice 的证书存储器中通常包含所需的根证书。安装 Web 浏览器时，将自动安装所有可信的 Root 证书。如果 Alice 没有 Root 证书，则必须从证书颁发机构下载并安装到浏览器的证书颁发机构中。证书颁发机构还可以是该 Web 服务器所处的设备。

可通过以下几种方式获得中间证书：

- 服务器以消息签名方式将所需的中间证书连同最低层实体证书一并发送给 Alice。这样，Alice 即可对证书链的完整性进行验证。
- 在这些证书中，通常包含证书签发者的 URL。Alice 可通过这些 URL 加载所需的中间证书。

在 STEP 7 中进行证书处理时，通常假设已将所需的中间证书和 Root 证书导入项目中，并已分配给模块。

2. Alice 使用这些证书的公钥，对证书链中的签名进行验证。
3. 对称密钥需已经生成并传送到 Web 服务器中。
4. 如果采用域名寻址 Web 服务器，则 Alice 还必须根据 RFC 2818 中定义的 Internet PKI 规范验证该 Web 服务器的身份。由于该 Web 服务器的 URL（此时，为"Fully Qualified Domain Name"(FQDN)）将保存到 Web 服务器的最终实体证书中，因此 Alice 可对该 Web 服务器的身份进行验证。如果字段"Subject Alternative Name"中的证书项与浏览器地址栏中的一致，则通过验证。

之后，即可通过对称密钥进行数据交换，如上图所示。

5.6.2.5 提示：在 RUN 模式中更新下载的证书

从 TIA Portal V19 和 S7-1500 CPU（包括 R/H-CPU）的固件版本 V3.1 开始，可以在运行期间（即在 CPU 的 RUN 模式下）更新证书。

这些是通过 TIA Portal 组态并下载的证书。

这允许在不中断正在执行的过程的情况下更新或替换证书。

要求

- 项目保护已激活。
- 项目管理员 (Engineering-Administrator) 以相应的安全权限登录。
- 硬件配置没有进一步的更改。这些更改需要 CPU 转入 STOP 才能下载。

规则

待更新证书的现有 ID 不得改变。

操作步骤

1. 在 TIA Portal（项目导航）中，导航到证书管理器（安全设置 > 安全功能）。
2. 根据要更新的证书，选择“证书颁发机构 (CA)”(Certificate authority (CA)) 选项卡或“设备证书”(Device certificates) 选项卡。
3. 选择需要更新的证书。
4. 在快捷菜单中，选择“更新”(Renew) 或“替换”(Replace) 命令，具体取决于要用文件系统中的证书替换现有证书还是仅更新现有证书的有效性数据。

   对于 CA 证书，所有派生的设备证书将使用更新后的 CA 证书自动进行数字签名。
5. 将更改下载到 CPU（下载到设备 > 硬件配置）。

5.6.3 安全通信要求

5.6.3.1 保护机密的组态数据

基于证书的协议需要私钥才能正常发挥作用，并且私钥必须获得妥善保护，如有关安全通信的基本信息中所述。

在 STEP 7 V17 及以上版本中，可通过密码保护这些密钥和其它需保护的数据：保护机密 PLC 组态数据的密码。

如果已采取相应措施保护 TIA Portal 项目和 CPU 组态防止未经授权的访问，则可以不使用密码。

无论是否分配密码：TIA Portal 都会生成用于保护机密 PLC 组态数据的密钥信息。此密码对安全通信过程没有影响。但是，用于保护机密 PLC 组态数据的密码的复杂度决定了私钥受到保护的程度。

提供密钥信息是进行安全通信（例如，基于 TLS 的 PG/HMI 间安全通信）的先决条件：只有此密钥信息可用时，CPU 才能处理安全通信所需的证书。

下图显示了描述的上下文。

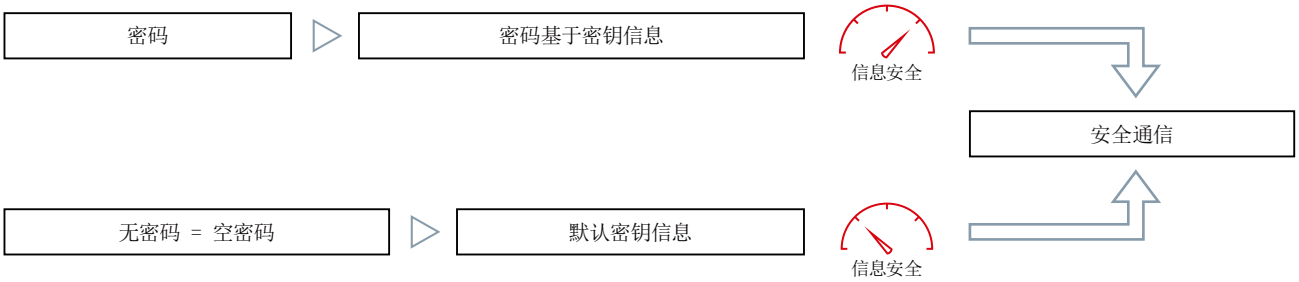


图 5-15 修改机密组态数据的上下文

安全设置向导

将硬件目录中的 CPU 添加到 TIA Portal 中支持 PG/HMI 安全通信的项目时，该 CPU 的安全设置向导随即启动。

该向导将引导您逐步完成以下 CPU 设置：

- 保护机密 PLC 组态数据的密码
- PG/PC 和 HMI 通信模式
- 访问等级

在向导中，将这些设置逐一进行详细说明。最后，在总览中再次统一显示所有设置。

在 TIA Portal 的网络视图中更换模块，该向导也将启动。与替换下来的 CPU 不同，新 CPU 支持 PG/HMI 安全通信。

向导中的所有设置都将应用到巡视窗口（CPU 属性）中。

通过 CPU 属性中"保护与安全"(Protection & Security) 区域内的"开始"(Start) 按钮，可随随时启动该向导。

要求

- TIA Portal 版本 V17 及以上版本
- CPU 支持 PG/HMI 间安全通信（S7-1500 CPU 固件版本 V2.9 及以上版本）
- CPU 尚未下载，或使用选项"删除用于保护机密 PLC 组态数据的密码"(Delete password for protection of confidential PLC configuration data) 将 CPU 复位为出厂设置

操作步骤

1. 在网络视图或设备视图中打开 CPU 属性。
2. 导航至区域"保护与安全 > 保护 PLC 组态数据"(Protection & Security > Protection of the PLC configuration data)。

   结果：首先启用"保护机密 PLC 组态数据"(Protect confidential PLC configuration data) 选项，然后用于输入密码的空白字段以红色突出显示。
3. 通过"设置"(Set) 按钮组态密码（推荐）或禁用"保护机密 PLC 组态数据"(Protect confidential PLC configuration data) 选项。
4. 完成组态并创建用户程序。
5. 下载到 CPU。
   下载硬件配置时，系统将要求用户重新输入一次密码。
   背景：在 TIA Portal 中使用已组态的密码来生成密钥信息，以保护机密组态数据。出于安全原因，密码和密钥信息均未保存在项目中。为了将密钥信息传送到 CPU，在下载硬件配置时会重新生成密钥信息，因此，此时必须重新输入一次密码。

也可在 PG/HMI 与 CPU 之间建立基于证书的通信

由于 TIA Portal 版本 V17 及以上版本和 CPU 固件版本 V2.9 (S7-1500) 或 V4.5 (S7-1200) 中的 PG/HMI 通信同样基于证书，因此在调试过程中，系统将提示用户接受服务器证书。

密码管理的提示和规则

- 在密码管理器中管理密码。
- 要检查新输入的密码合规性，防止密码简单等问题，请使用 TIA Portal 的密码策略验证设置。
  - 在项目树中，导航至区域“<项目名称> > 安全设置 > 设置”(<Project name> > Security settings > Settings)，然后选择“密码策略”(Password policies) 区域。
  - 例如，指定密码必须包含的最少字符数或最少特殊字符数。
- 无需为系统或计算机中的每个 CPU 分配不同的密码。如果满足要求，可以为一组 CPU 定义相同的密码。在更换部件方案中，该策略也具有优势：如果将组密码分配给更换的 CPU，则可减少更换 CPU 的工作量。

  请注意，如果其中一个 CPU 的密码发生泄露，则采用相同密码的所有 CPU 都面临风险。
- 由于除组态之外，还需将保护机密 PLC 组态数据的密码也传送到新（替换）CPU 中，因此密码的定义也会影响部件的更换方案（参见“更换部件方案的规则 (页 79)”）。
- 使用 **S7-1500R/H CPU** 时，加载过程中仅将机密 PLC 组态数据的密码加载到其中一个 CPU 中。为确保同步过程正常运行且伙伴 CPU 正常运行，同步前需通过“在线与诊断”(Online and Diagnostics) 编辑器将该密码传送到伙伴 CPU 中：
  - 在“在线和诊断”(Online and diagnostics) 视图中，可指定区域“保护机密 PLC 组态数据的密码”(Password to protect confidential PLC configuration data)。
  - 输入所需密码，并单击“设置”(Set) 按钮。

    如果输入的密码正确，则伙伴 CPU 可使用受保护的 PLC 组态数据并启动同步过程。

5.6.3.2 有关保护机密 PLC 组态数据的实用信息

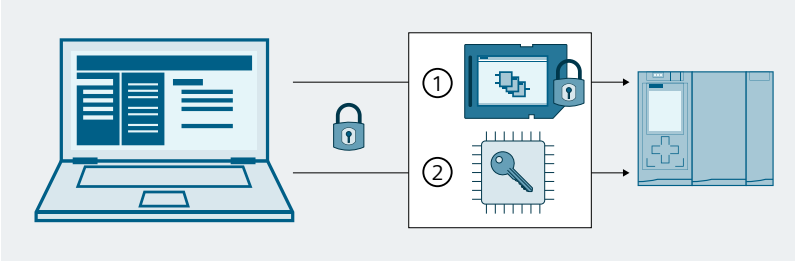
受安全标准保护的安全通信概念包括以下组成部分：

- 基于密码的密钥信息，用于保护机密组态数据（例如，证书、密码的私钥）。
- 保障参与者（例如，编程设备和 CPU）之间通信的标准化日志 (TLS)。

“保护机密组态数据”原则

下图简要显示了如何保护标准 S7-1500 CPU 等设备的机密组态数据：首次下载时，两个组件项目和密钥信息将放在不同的存储区中。该项目位于装载存储器（存储卡）中，密钥信息位于 CPU 的存储区中。

对于具有不同存储概念的其它目标系统（例如 S7-1200 CPU、软件控制器），实现方式取决于相应的存储概念，但存储原理相同。



① 具有密码保护的机密组态数据的项目（此处：在装载存储器中，即存储卡中）

② 使用受保护机密组态数据的密钥信息（通过密码生成）（此处：在 CPU 的存储区中）

图 5-16 保护机密组态数据的原理

两个存储区可提高安全性

涉及的组件像两个匹配的拼图一样彼此相关：项目绑定到下载的密钥信息，下载的密钥信息绑定到组态期间分配的密码。

项目和密钥信息必须匹配，否则 CPU 将无法启动。

两个独立存储区的原理也适用于不带存储卡的 S7-1200 CPU 和 S7-1500 CPU 版本，例如软件控制器或 PLCSim/PLCSim Advanced。在不带存储卡的版本中，使用两个单独的分区，以便可以独立管理两个信息项。

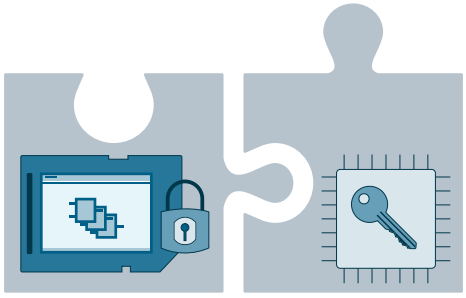


图 5-17 两个独立存储区的原理

5.6.3.3 更改密码

具体操作步骤取决于是否已下载 CPU。如果 CPU 已下载，则包含密钥信息，可以通过该密钥信息使用受密码保护的 PLC 组态数据。

更改密码 - 尚未下载组态

只要尚未将组态下载到 CPU 中，就可以直接更改输入的密码或取消激活密码保护。

要求

- CPU 尚未下载。

操作步骤

1. 在网络视图或设备视图中打开 CPU 属性。
2. 导航至区域“保护与安全 > 保护 PLC 组态数据”(Protection & Security > Protection of the PLC configuration data)。
3. 单击“更改”(Change) 按钮或禁用选项“保护机密 PLC 组态数据”(Protect confidential PLC configuration data)。
4. 在对话框中输入之前的有效密码。如果要更改密码，还需输入新密码并确认新密码。

只要尚未将组态下载到 CPU 中，CPU 便处于配置阶段（参见“从下载到运行就绪的 CPU 行为 (页 101)”），可以使用组态的密码下载任何有效的组态。

更改密码 - 组态已下载

如果 CPU 已经下载组态，并且该组态受到机密 PLC 组态数据所用密码的保护，则必须首先将 CPU 复位为出厂设置，并删除 CPU 中机密 PLC 组态数据的密码，或直接在线删除密码，然后进行设置。

要求

- 具有对 CPU 的写访问权限
- CPU 必须处于 STOP 模式。

操作步骤

1. 在网络视图中选择 CPU。
2. 在快捷菜单中，选择“在线和诊断”(Online & Diagnostics) 命令。

3. 如果还更改存储卡上的项目，即重新下载组态：
   - 在打开的在线和诊断视图中选择"复位为出厂设置"(Reset to factory settings) 区域。
   - 激活选项"删除保护机密 PLC 组态数据的密码"(Delete password to protect confidential PLC configuration data)。为了避免 CPU 重复启动，还需选择"格式化存储卡"(Format memory card) 选项。
   - 然后使用更改后的组态和所需的密码下载项目。
4. 如果无需更改存储卡上的项目，即仅设置密码：
   - 在"在线和诊断"(Online and diagnostics) 视图中，指定区域"保护机密 PLC 组态数据的密码"(Password for the protection of confidential PLC configuration data)。
   - 单击"删除"(Delete) 按钮。如果"删除"(Delete) 按钮不可用，则表示尚未在 CPU 中设置密码。
   - 输入所需密码，然后单击"设置"(Set) 按钮。

   如果输入了正确的密码，则 CPU 可以使用受保护的 PLC 组态数据。

没有对 CPU 的写访问权限

如果没有对装载存储器的写访问权限（读访问级别），请先从 CPU 上移除存储卡或从外部（例如在计算机中）删除存储卡，然后再使用选项"删除用于保护机密 PLC 组态数据的密码"(Delete password to protect confidential PLC configuration data) 复位为出厂设置。

说明

通过模式开关/模式选择键将 CPU 恢复为出厂设置

通过模式开关/模式选择键将 CPU 恢复为出厂设置时，还会删除 CPU 的 IP 地址，但不会删除用于保护机密 PLC 组态数据的密码。

说明

通过显示屏将 CPU 恢复为出厂设置

通过显示屏将 CPU 恢复为出厂设置时，将删除用于保护机密 PLC 组态数据的密码。

更多信息

有关如何在采用备件的情况下继续操作的信息，请参见"更换部件方案的规则 (页 79)"部分。

5.6.3.4 重置密码

可以重置机密 PLC 组态数据的保护。例如，若希望更改密码，但不再记得当前密码，则必须使用此操作。

密码丢失 - 尚未下载组态

首次通过 TIA Portal 下载 CPU 时必须输入密码，否则无法使用该 CPU 的 CPU 组态。要在 CPU 属性中更改密码，还必须输入先前有效的密码。如果忘记密码，请执行以下操作：

要求

- CPU 尚未下载。

操作步骤

1. 在网络视图或设备视图中打开 CPU 属性。
2. 导航至区域"保护与安全 > 保护 PLC 组态数据"(Protection & Security > Protection of the PLC configuration data)。
3. 单击"复位"(Reset)。

   请注意，CPU 的证书（例如 Web 服务器、OPC UA 服务器、PG/PC 通信和 HMI 通信的证书）在复位后无法再继续使用，必须重新创建和分配。

   – 如果证书管理器中使用全局安全设置，则必须通过证书管理器重新分配证书。
   – 如果证书管理器中未使用全局安全设置，则必须重新创建和分配证书。
4. 确认重置密码。

保护机密 PLC 组态数据的选项仍处于激活状态。

删除密码 – 组态已下载

如果 CPU 已经下载组态，并且该组态受到机密 PLC 组态数据所用密码的保护，则为了下载新项目，请在线删除机密 PLC 组态数据的密码，然后指定新密码。

要求

- 具有对 CPU 的写访问权限
- CPU 必须处于 STOP 模式。

操作步骤

1. 在网络视图中选择 CPU。
2. 在快捷菜单中，选择"在线和诊断"(Online & Diagnostics) 命令。
3. 在区域"保护机密 PLC 组态数据的密码"(Password to protect confidential PLC configuration data) 中，单击"删除"(delete) 按钮。
   如果"删除"(Delete) 按钮不可用，则表示尚未在 CPU 中设置密码。

注意
删除机密组态数据的密码 如果删除密码，而下载的项目需要相应的密码，则该项目在没有密码的情况下无法继续工作。

4. 如需要，可通过“设置”(Set) 按钮输入新密码。
5. 重启 CPU。

说明

CPU 恢复出厂设置后，用于保护机密组态数据的密码

- **通过模式开关/模式选择键（未插入 SIMATIC 存储卡）：**用于保护机密组态数据的密码保留。
- **使用显示屏：**用于保护机密组态数据的密码删除。
- **使用 STEP 7：**用于保护机密组态数据的密码保留。仅当设置了“删除用于保护机密 PLC 组态数据的密码”(Delete password for protection of confidential PLC configuration data) 选项时，才会删除该密码。

更多信息

有关更改密码的信息，请参见“更改密码 (页 72)”部分。

5.6.3.5 通过 SIMATIC 存储卡分配密码

如果要在不使用 TIA Portal 的情况下将用于保护机密 PLC 组态数据的密码传送到 CPU，可以使用 SIMATIC 存储卡来实现此功能。

SIMATIC 存储卡适用于以下用途：

- 准备一个新的 CPU

  如果再次设置 CPU，则组态时应设置用于保护机密 PLC 组态数据的密码。完成此组态后，可以使用包含所需项目的另一个 SIMATIC 存储卡。

  （S7-1200 CPU：具有传送作业的“传送”卡也可用于在 CPU 上安装程序）。
- CPU 具有用于保护机密 PLC 组态数据的密码，但该密码与项目不匹配

  如果密码不相同，则可使用 CPU 中的存储卡设置正确的密码。

  （S7-1200 CPU：配有 SIMATIC“传送”卡或 SIMATIC“程序”卡）。
- 在 CPU 中重置用于保护机密 PLC 组态数据的密码

  准备处置旧 CPU 或为 CPU 准备新项目。

要求

- TIA Portal 版本 V17 及以上版本

基本操作步骤

1. 创建具有“设置密码”作业的 SIMATIC 存储卡

   该操作按照特殊模式创建文件夹和文件结构，并将用于保护机密 PLC 组态数据的密码以纯文本形式写入到 SIMATIC 存储卡的特殊文件中。参见以下描述。

2. 将准备好的 SIMATIC 存储卡插入 CPU 中并接通 CPU 电源。

   PLC 读取密码并对其进行处理，然后将结果存储在内部存储器中。任何现有数据都将被覆盖。

3. 移除 SIMATIC 存储卡并重启 CPU。

   结果 (S7-1500)：CPU 读取 SIMATIC 存储卡时，LED 指示灯的闪烁方式与固件更新时相同。CPU 设置密码时，RUN/STOP LED 指示灯闪烁。该过程成功完成后，RUN/STOP LED 指示灯呈黄色亮起且 MAINT LED 指示灯呈黄色闪烁。

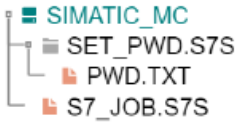
操作结果以成功或错误消息的形式显示在诊断缓冲区中。如果无法设置密码，则错误 LED 指示灯将与其它 LED 指示灯一起闪烁。

创建具有“设置密码”作业的 SIMATIC 存储卡

1. 在根目录中创建一个名为“SET_PWD.S7S”的文件夹。
2. 在该文件夹中创建一个名为“PWD.TXT”的文本文件，其中仅包含文本形式的密码。
3. 在存储卡的根目录中创建一个名为“S7_JOB.S7S”的文本文件，其中包含内容“SET_PWD”。

   此文件作为“作业文件”，用于分配保护 PLC 的机密 PLC 组态数据的密码。

4. SIMATIC 存储卡上的文件结构显示如下：



说明

SIMATIC 存储卡的安全存储

将 SIMATIC 存储卡存储在只有授权人员才能访问的安全位置。

规则和建议

- 设置密码必须在安全的环境中进行。
- 文本文件“PWD.TXT”的内容定义用于保护机密 PLC 组态数据的密码。该密码必须与 CPU 组态中分配的密码匹配。
- 要重置 PLC 的现有密码，文本文件“PWD.TXT”必须为空，即文件大小为 0 字节。
- 使用任意文本编辑器来创建文本文件。推荐的文本格式为“UTF-8”。
- 文件夹名称和文件名不区分大小写。但是，密码本身区分大小写。
- 不要在末尾处输入 CR/LF 字符（PWD.TXT 或 S7_JOB.S7S）。

5.6.3.6 备份和恢复 CPU 时的特殊功能

在 TIA Portal 中，可备份 CPU 的功能组态以便后期访问。即，之后可恢复最初备份的状态。备份后用户便可以下载修改后的组态，例如，测试产品增强功能，更改程序以在系统中进行故障排除，或者可以在测试的基础上更换组件。然后，可恢复该 CPU 最初备份的组态。

备份组态。

在备份 CPU（TIA Portal 中的"在线"(Online) 菜单，"从在线设备下载备份"(Load backup from online device)）时，也会备份用于保护机密 PLC 组态数据的密码。

恢复备份

恢复 CPU 的备份时（TIA Portal 中的菜单"在线"(Online)，对标记的备份执行命令"下载到设备"(Download to device)），只有满足以下条件，CPU 才能与 PG/PC 或 HMI 进行通信：

- 恢复保护机密 PLC 组态数据时使用密码保护的组态后，该 CPU 中必需包含此密码。
  否则，CPU 无法访问该组态数据，因此无法启动。

补救措施

如果发生上述错误（即保护机密 PLC 组态数据的密码与备份不匹配），则必须删除保护 CPU 中机密 PLC 组态数据的密码，然后设置正确的密码。重新启动 CPU 后，备份功能恢复正常。

5.6.3.7 有关避免错误和错误处理的提示

以下说明列出了一些可能导致 CPU 错误消息的用例。

诊断缓冲区提供信息

用于保护机密组态数据的密码与下载的组态不匹配时，CPU 会检测到该问题。诊断缓冲区中的消息指示可能的原因和补救措施，通常可以作为问题的解决方案。

典型的"陷阱"

为了避免或纠正错误，请注意以下情况：

- 组态已下载？
  无论是否使用密码保护机密组态数据：如果没有下载的组态，CPU 便不会退出配置阶段。
- 正在尝试下载包含组态密码的 CPU，而 CPU 已经收到另一个密码。
  例如：CPU 已更换为库存中的另一个 CPU。更换的 CPU 件并未完全复位（通过选项"删除用于保护机密 PLC 组态数据的密码"(Delete password for protection of confidential PLC configuration data) 复位为出厂设置）。
  补救措施：
  - 准备更换 CPU 时，始终使用适当的设置（密码已删除）。
  - 对于要下载的组态，使用已下载的组态中所使用的密码。
  - 也可能下载了错误的项目/CPU 组态。检查正确的 CPU 组态是否可用。
  - 使用在线功能"设置用于保护机密 PLC 组态数据的密码"(Set password to protect confidential PLC configuration data) 删除密码或设置为与 CPU 组态相同的密码。然后重启设备。
- 如果 CPU 组态不使用密码，而已下载的组态需要用户自定义密码，则仍会发生错误。
  补救措施：
  - 使用在线功能"设置用于保护机密 PLC 组态数据的密码"(Set password to protect confidential PLC configuration data) 删除密码或设置为与 CPU 组态相同的密码。然后重启设备。

5.6.3.8 更换部件方案的规则

分配用于保护机密 PLC 组态数据的密码也会对更换部件方案产生影响。

更换部件方案的规则

请遵守以下更换部件方案的规则：

通过 TIA Portal 组态更换的 CPU

- 更换 CPU 不应具有组态或用于保护机密 PLC 组态数据的密码。
  优势：无论是否组态了密码，都可以将项目下载到更换 CPU 中，而无需进行任何其它准备工作。
- 如果已组态更换 CPU，则必须将 CPU 复位为出厂设置，同时设置以下选项：
  - “删除保护机密 PLC 组态数据的密码”(Delete password for protection of confidential PLC configuration data)
  - “格式化存储卡”

通过存储卡向更换 CPU 提供组态数据

- 如果未向项目中的 CPU 分配用于保护机密 PLC 组态数据的密码，则可以将旧 CPU 的存储卡插入到全新未使用的 CPU 中，而无需采取任何其它操作。
  如果更换 CPU 组态了保护机密 PLC 组态数据的密码，则必须首先使用“删除保护机密 PLC 组态数据的密码”(Delete password for protection of confidential PLC configuration data) 选项将该 CPU 复位为出厂设置。
- 如果已为一组 CPU 分配了相同的密码，则还可以通过 TIA Portal 或适用的存储卡将组密码分配给更换 CPU（参见“保护机密的组态数据 (页 68)”）。
  此时，可将包含有当前项目的存储卡插入 CPU 中并直接进行处理，而无需任何密码相关操作。
- 如果为项目中的每个 CPU 分配不同的密码，则在使用更换 CPU 时，需要先使用在线和诊断编辑器为各 CPU 设置有效密码（“设置保护机密 PLC 组态数据的密码”(Set password for protection of confidential PLC configuration data) 区域，具体参见“更改密码 (页 72)”）。

更多信息

在“通过 SIMATIC 存储卡分配密码 (页 75)”部分，可了解如何使用 SIMATIC 存储卡分配密码，以保护机密 PLC 组态数据。

5.6.4 开放式用户安全通信

5.6.4.1 S7-1500 CPU（作为 TLS 客户端）与外部 PLC（TLS 服务器）之间的安全 OUC

在以下章节中，将介绍如何通过 TCP 建立 S7-1500 CPU（作为 TLS 客户端）与 TLS 服务器之间的开放式用户通信。

建立 S7-1500 CPU（作为 TLS 客户端）与 TLS 服务器之间的安全 TCP 连接

S7-1500 CPU 固件版本 V2.0 及以上版本支持通过域名系统 (DNS) 进行寻址的安全通信。

要通过域名进行 TCP 安全通信，则需手动创建一个 TCON_QDN_SEC 系统数据类型的数据块，并分配参数，之后在 TSEND_C、TRCV_C 或 TCON 指令中直接调用该数据块。

要求：

- 在 CPU 中，设置当前的日期和时间。
- 网络中包含至少一台 DNS 服务器。
- 已为 S7-1500 CPU 组态至少一台 DNS 服务器。
- TLS 客户端和 TLS 服务器具有所需的全部证书。

要建立与 TLS 服务器的 TCP 安全连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个 TCON_QDN_SEC 数据类型的变量。

   在以下示例中，显示了一个全局数据块“Data_block_1”，其中，定义了数据类型 TCON_QDN_SEC 的变量“DNS ConnectionSEC”。

Data_block_1

	Name	Data type	Start value	Comment
1	Static			
2	DNS Connection SEC	TCON_QDN_SEC		
3	ConnPara	TCON_QDN		parameter of the TCP connection
4	InterfaceId	HW_ANY	0	not relevant
5	ID	CONN_OUC	5	connection reference / identifier
6	ConnectionType	Byte	11	type of connection: 16#0B=11=TCP/IP, 16#13
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	RemoteQDN	String[254]	'plc_1.factory127.'	fully or partially qualified domain name of rem
9	RemotePort	UInt	4000	remote UDP / TCP port number
10	LocalPort	UInt	0	local UDP / TCP port number
11	ActivateSecureConn	Bool	true	activate the security functionality of that conn
12	TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a
13	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 a
14	TLSServerCertRef	UDInt	7	for Server side: Reference to own X.509 V3 se
15	TLSClientCertRef	UDInt	0	for Client side: add id of own X.509 V3 client c

图 5-18 数据类型 TCON_QDN_SEC

3. 在“起始值”(Start value) 列中，设置 TCP 连接的连接参数。例如，在 "RemoteQDN"中输入 TLS 服务器全限定的域名 (FQDN)。

4. 在“起始值”(Start value) 列中，设置安全通信的参数。
   - “ActivateSecureConn”：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可建立非安全的 TCP 或 UDP 连接。
   - “ExtTLSCapabilities”：如果输入值 1，则客户端将通过验证服务器端 X.509-V3 证书中的 subjectAlternateName，验证该服务器的身份。验证过程将由该指令执行。
   - “TLSServerCertRef”：X.509-V3 证书（通常为 CA 证书）的 ID，TLS 客户端使用该 ID 验证 TLS 服务器的身份。如果该参数为 0，则 TLS 客户端将使用客户端证书中心当前加载的所有 (CA) 证书对服务器的身份进行验证。

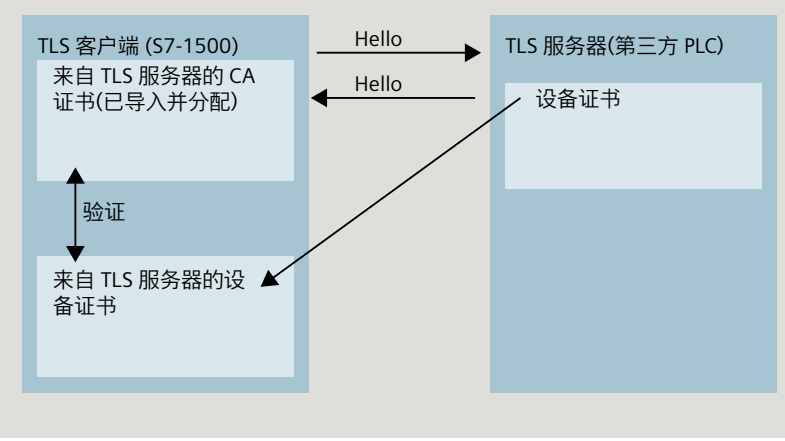


图 5-19 从作为 TLS 客户端的 S7-1500 的角度处理证书

   - “TLSClientCertRef”：自身 X.509-V3 证书的 ID。
5. 在程序编辑器中，创建一个 TSEND_C、TRCV_C 或 TCON 指令。
6. 将 TSEND_C、TRCV_C 或 TCON 指令的 CONNECT 参数与 TCON_QDN_SEC 数据类型的变量进行互连。

   在以下示例中，TCON 指令的 CONNECT 参数已与变量“DNS connectionSEC”（数据类型 TCON_QDN_SEC）互连。

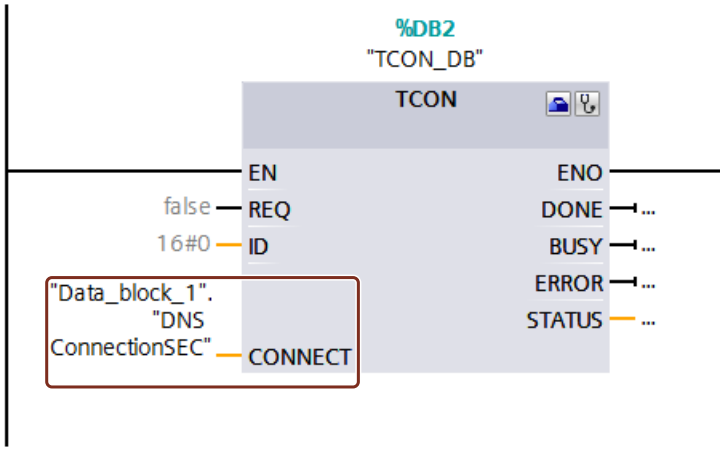


图 5-20 TCON 指令

更多信息

有关 TCON_QDN_SEC 系统数据类型的更多信息，请参见 STEP 7 在线帮助。

有关安全通信的更多信息，请参见“安全通信 (页 47)”部分。

5.6.4.2 S7-1500 CPU（作为 TLS 服务器）与外部 PLC（TLS 客户端）之间的安全 OUC

在以下章节中，将介绍如何通过 TCP 建立 S7-1500 CPU（作为 TLS 服务器）与 TLS 客户端之间的开放式用户通信。

通过通信伙伴的域名建立 TCP 安全连接。

S7-1500 CPU 固件版本 V2.0 及以上版本支持通过域名系统 (DNS) 进行寻址的安全通信。

要通过域名进行 TCP 安全通信，则需手动创建一个 TCON_QDN_SEC 系统数据类型的数据块，并分配参数，之后在 TSEND_C、TRCV_C 或 TCON 指令中直接调用该数据块。

要求：

- 在 CPU 中，设置当前的日期和时间。
- 网络中包含至少一台 DNS 服务器。
- 已为 S7-1500 CPU 组态至少一台 DNS 服务器。
- TLS 客户端和 TLS 服务器具有所需的全部证书。

要建立与 TLS 客户端的安全 TCP 连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个 TCON_QDN_SEC 数据类型的变量。

   在以下示例中，显示了一个全局数据块“Data_block_1”，其中，定义了数据类型 TCON_QDN_SEC 的变量“DNS ConnectionSEC”。

Data_block_1

		Name	Data type	Start value	Comment
1		Static			
2		DNS Connection SEC2	TCON_QDN_SEC		
3		ConnPara	TCON_QDN		parameter of the TCP connection
4		InterfaceId	HW_ANY	0	not relevant
5		ID	CONN_OUC	8	connection reference / identifier
6		ConnectionType	Byte	11	type of connection: 16#0B=11=TCP/IP, 16#13
7		ActiveEstablished	Bool	false	active/passive connection establishment
8		RemoteQDN	String[254]	''	fully or partially qualified domain name of rem
9		RemotePort	UInt	0	remote UDP / TCP port number
10		LocalPort	UInt	2010	local UDP / TCP port number
11		ActivateSecureConn	Bool	true	activate the security functionality of that conn
12		TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a
13		ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 a
14		TLSServerCertRef	UDInt	5	for Server side: Reference to own X.509 V3 se
15		TLSClientCertRef	UDInt	0	for Client side: add id of own X.509 V3 client c

图 5-21 TCON_QDN_SEC_Server

3. 在“起始值”(Start value) 列中，设置 TCP 连接的连接参数。例如，在“ID”中输入 TCP 连接的本地 ID。

4. 在“起始值”(Start value) 列中，设置安全通信的参数。
   - “ActivateSecureConn”：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可建立非安全的 TCP 或 UDP 连接。
   - “TLSServerReqClientCert”：TLS 客户端需具有 X.509-V3 证书。
   - “TLSServerCertRef”：自身 X.509-V3 证书的 ID。

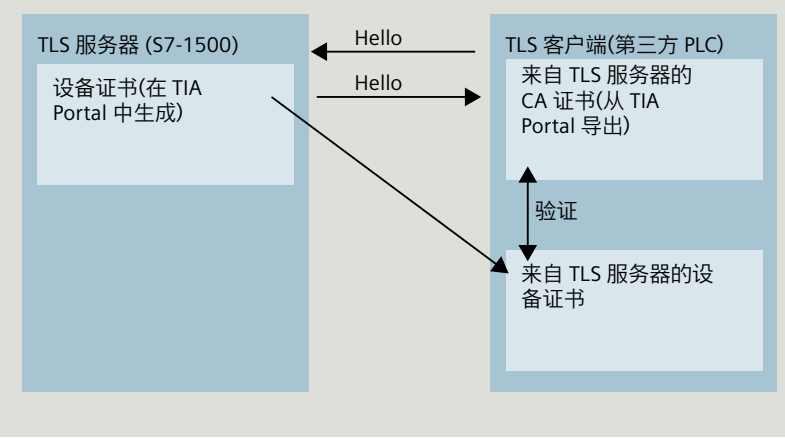


图 5-22 从作为 TLS 服务器的 S7-1500 的角度处理证书

   - “TLSClientCertRef”：X.509-V3 证书（或 X.509-V3 证书组）的 ID，TLS 服务器使用该 ID 验证 TLS 客户端的身份。如果该参数为 0，则 TLS 服务器将使用服务器证书中心当前加载的所有 (CA) 证书对客户端的身份进行验证。

5. 在程序编辑器中，创建一个 TSEND_C、TRCV_C 或 TCON 指令。
6. 将 TSEND_C、TRCV_C 或 TCON 指令的 CONNECT 参数与 TCON_QDN_SEC 数据类型的变量进行互连。

   在以下示例中，TCON 指令的 CONNECT 参数已与变量“DNS connectionSEC”（数据类型 TCON_QDN_SEC）互连。

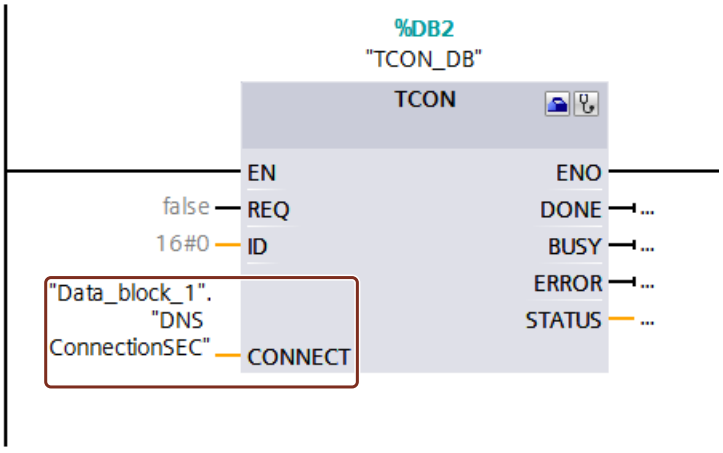


图 5-23 TCON 指令

更多信息

有关 TCON_QDN_SEC 系统数据类型的更多信息，请参见 STEP 7 在线帮助。

有关安全通信的更多信息，请参见"安全通信 (页 47)"部分。

5.6.4.3 两个 S7-1500 CPU 之间的安全 OUC

在以下章节中，介绍如何通过 TCP 在两个 S7-1500 CPU 之间建立开放式用户安全通信。在此过程中，一个 S7-1500 CPU 用作 TLS 客户端（主动建立连接）而另一个 S7-1500 CPU 则用作 TLS 服务器（被动建立连接）。

建立两个 S7-1500 CPU 之间的安全 TCP 连接

要在两个 S7-1500 CPU 之间建立 TCP 安全通信，则需为每个 CPU 手动创建 TCON_IP_V4_SEC 系统数据类型的数据块，并分配相应参数，之后在 TSEND_C、TRCV_C 或 TCON 指令中直接调用该数据块。

要求：

- 在 CPU 中，设置当前的日期和时间。
- 两个 S7-1500 CPU 的固件版本为 V2.0 及以上版本
- TLS 客户端和 TLS 服务器具有所需的全部证书。

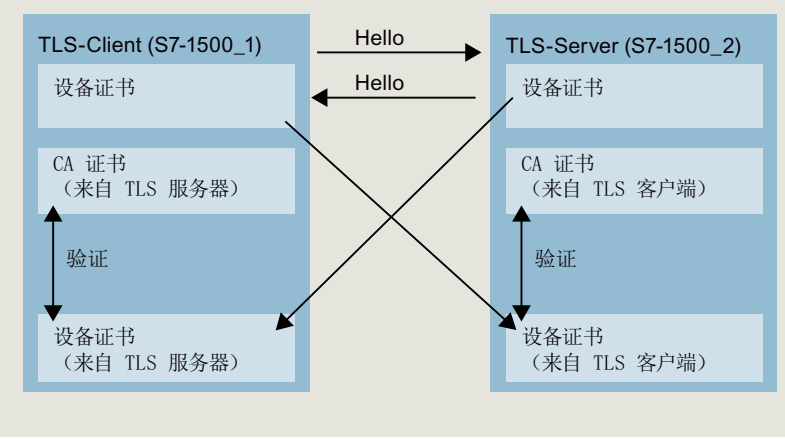


图 5-24 两个 S7-1500 CPU 之间安全 OUC 的证书处理过程

TLS 客户端的设置

要在 TLS 客户端中建立安全的 TCP 连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个数据类型为 TCON_IP_V4_SEC 的变量。

   以下示例中显示了全局数据块“Data_block_1”，其中，定义了数据类型为 TCON_IP_V4_SEC 的变量“SEC 连接 1 TLS 客户端”(SEC connection 1 TLS-Client)。

Data_block_1

	Name	Data type	Start value	Comment
1	Static			
2	SEC connection 1 TLS-Client	TCON_IP_V4_SEC		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	InterfaceId	HW_ANY	72	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connetion: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	100	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against th
19	TLSServerCertRef	UDInt	1	for Server side: Reference to own X.509 V3 server certificate; fc
20	TLSClientCertRef	UDInt	5	for Client side: add id of own X.509 V3 client certificate; for Sen

图 5-25 IP_V4_SEC_Client

3. 在“起始值”(Start value) 列中，设置 TCP 连接的连接参数。例如，在“RemoteAddress”中输入 TLS 服务器的 IPv4 地址。

说明

连接参数接口 ID

请注意，可为数据类型为 TCON_IP_V4_SEC 的接口 ID 输入值“**0**”。在这种情况下，CPU 会自行搜索合适的本地 CPU 接口。

---

4. 在“起始值”(Start value) 列中，设置安全通信的参数。
   - “ActivateSecureConn”：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可建立非安全的 TCP 或 UDP 连接。
   - “TLSServerCertRef”：输入值“2”（引用 TIA Portal 项目 (SHA256) 的 CA 证书），或输入值“1”（引用 TIA Portal 项目 (SHA1) 的 CA 证书）。如果使用不同的 CA 证书，则需在证书管理器的全局安全设置中输入相应的 ID。
   - “TLSClientCertRef”：自身 X.509-V3 证书的 ID。
5. 在程序编辑器中，创建一个 TSEND_C、TRCV_C 或 TCON 指令。
6. 将 TSEND_C、TRCV_C 或 TCON 指令的 CONNECT 参数与 TCON_IP_V4_SEC 数据类型的变量进行互连。

TLS 服务器的设置

要在 TLS 服务器中建立安全的 TCP 连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个数据类型为 TCON_IP_4_SEC 的变量。

   以下示例中显示了全局数据块“Data_block_1”，其中，定义了数据类型为 TCON_IP_V4_SEC 的变量“SEC 连接 1 TLS 服务器”(SEC connection 1 TLS-Server)。

Data_block_1

	Name	Data type	Start value	Comment
1	Static			
2	SEC connection 1 TLS-Server	TCON_IP_V4_SEC		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	InterfaceId	HW_ANY	120	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connetion: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	false	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	10	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	true	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against the
19	TLSServerCertRef	UDInt	6	for Server side: Reference to own X.509 V3 server certificate; fo
20	TLSClientCertRef	UDInt	1	for Client side: add id of own X.509 V3 client certificate; for Serv

图 5-26 IP_V4_SEC_Server

3. 在“起始值”(Start value) 列中，设置 TCP 连接的连接参数。例如，在“RemoteAddress”中输入 TLS 客户端的 IPv4 地址。
4. 在“起始值”(Start value) 列中，设置安全通信的参数。
   - “ActivateSecureConn”：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可建立非安全的 TCP 或 UDP 连接。
   - “TLSServerReqClientCert”：TLS 客户端需具有 X.509-V3 证书。输入值“true”。
   - “TLSServerCertRef”：自身 X.509-V3 证书的 ID。
   - “TLSClientCertRef”：输入值“2”（引用 TIA Portal 项目 (SHA256) 的 CA 证书），或输入值“1”（引用 TIA Portal 项目 (SHA1) 的 CA 证书）。如果使用不同的 CA 证书，则需在证书管理器的全局安全设置中输入相应的 ID。
5. 在程序编辑器中，创建一个 TSEND_C、TRCV_C 或 TCON 指令。

6. 将 TSEND_C、TRCV_C 或 TCON 指令的 CONNECT 参数与 TCON_IP_V4_SEC 数据类型的变量进行互连。

   在以下示例中，TSEND_C 指令的 CONNECT 参数将与变量"SEC connection 1 TLS client"（数据类型 TCON_IP_4_SEC）进行互连。

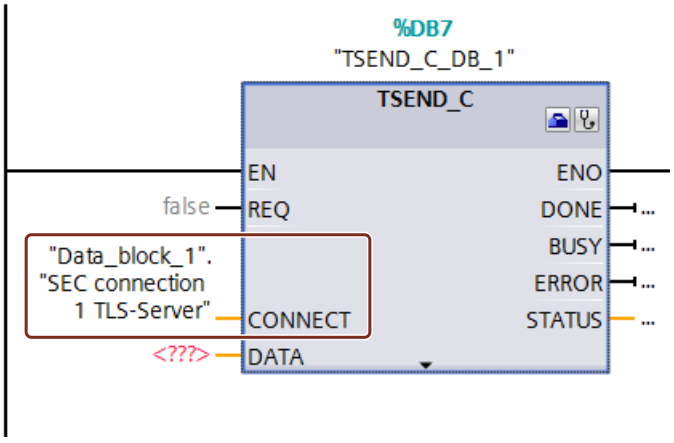


图 5-27 TSEND_C

更多信息

有关 TCON_IP_4_SEC 系统数据类型的更多信息，请参见 STEP 7 在线帮助。

有关安全通信的更多信息，请参见"安全通信 (页 47)"部分。

5.6.4.4 通过 CP 接口进行安全 OUC 连接

在以下章节中，将介绍通过 CP 接口进行开放式用户安全通信时应注意的特殊事项。至少一个站为 S7-1500 站，并包含以下模块：

- S7-1500 CPU（固件版本 V2.0 及更高版本）（S7-1500 软件控制器除外）
- CP
  - CP 1543-1（固件版本 V2.0 及更高版本）
  - CP 1545-1（固件版本 V1.0 及更高版本）
  - CP 1543SP-1（固件版本 V1.0 及更高版本）

该 CP 在 S7-1500 站中将作为 TLS 客户端（主动建立连接）或 TLS 服务器（被动建立连接）。

通过 CP 接口进行安全通信的基本操作步骤与概念，与通过 S7-1500 CPU 接口进行安全通信的类似。在此，必须将证书分配给作为 TLS 服务器或 TLS 客户端的 CPU，而非其它 CPU。因此，也可使用其他角色和操作步骤。在下文中，将对此进行详细介绍。

管理 CP 的证书

以下规则普遍适用：在入全局安全设置中，需登录证书管理器。生成自签名的证书时，需登录全局安全设置。需要具有足够的用户权限（管理员权限，或具有“安全组态”权限的“标准”用户）。

在 CP 中，可在“安全 > 安全属性”(Security > Security properties) 部分生成或分配证书。在此部分中，可登录全局安全设置。

操作步骤：

1. 在 STEP 7 的网络视图中，选中该 CP 并在巡视窗口中选择“安全 > 安全属性”(Security > Security properties) 部分。
2. 单击“用户登录”(User logon) 按钮。
3. 使用用户名和密码进行登录。
4. 启用“激活安全功能”(Activate security functions) 选项。

   系统将初始化相应的安全属性。
5. 单击“设备证书”(Device certificates) 表格的第一行，生成一个新的证书或选择现有的设备证书。
6. 如果通信伙伴也是一个 S7-1500 站，则需按照上述操作，使用 STEP 7 为通信伙伴或该 S7-1500 CPU 指定一个设备证书。

示例：通过 CP 接口，在两个 S7-1500 CPU 之间建立 TCP 安全连接

为了在两个 S7-1500 CP 之间建立安全的 TCP 通信，必须在各个 CPU 中创建系统数据类型为 TCON_IP_V4_SEC 的数据块，执行参数分配以及直接以指令进行调用。

要求：

- 两个 S7 1500 CPU 均具有上述指定固件版本之一。
- CP 具有上述指定固件版本之一。
- TLS 客户端和 TLS 服务器具有所需的全部证书。
  - 必须为该 CP 生成设备证书（最终实体证书）并存储在该 CP 的证书存储器中。如果通信伙伴是一个外部设备（如，MES 或 ERP 系统），则需确保该设备上包含有设备证书。
  - 对通信伙伴设备证书进行签名的 root 证书（CA 证书）也必须位于该 CP 的证书存储器中，或位于外部设备的证书存储器中。如果使用中间证书，则必须确保所验证设备中的证书路径完整。设备将通过这些证书验证通信伙伴的设备证书。
- 这些通信伙伴需通过 IPv4 地址进行寻址，而不能通过域名进行寻址。

下图显示了在两个通信伙伴通过一个 CP 进行通信的场景中，设备中的不同证书。此外，在该图中还显示了建立连接时设备证书的传输（“Hello”）。

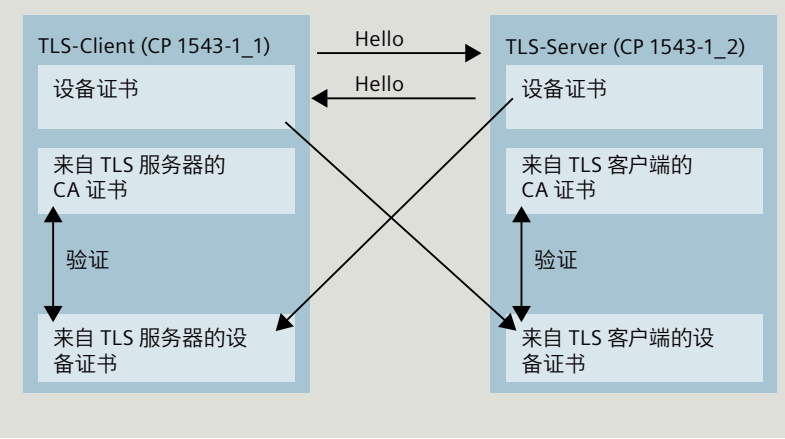


图 5-28 通过 CP 接口，在两个 S7-1500 CPU 之间进行 OUC 安全连接的证书处理操作。

TLS 客户端的设置

要在 TLS 客户端中建立安全的 TCP 连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个数据类型为 TCON_IP_4_SEC 的变量。为此，需在“数据类型”(Data type) 字段中输入字符串“TCON_IP_V4_SEC”。

   以下示例中显示了全局数据块“Data_block_1”，其中，定义了数据类型为 TCON_IP_V4_SEC 的变量“SEC 连接 1 TLS 客户端”(SEC connection 1 TLS-Client)。

   该接口 ID 的值为本地 CP（TLS 客户端）中 IE 接口的硬件标识符。

Data_block_1

	Name	Data type	Start value	Comment
1	Static			
2	SEC connection 1 TLS-Client	TCON_IP_V4_SEC		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	InterfaceId	HW_ANY	258	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connetion: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	100	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against th
19	TLSServerCertRef	UDInt	1	for Server side: Reference to own X.509 V3 server certificate; fo
20	TLSClientCertRef	UDInt	5	for Client side: add id of own X.509 V3 client certificate; for Serv

图 5-29 IP_V4_SEC_Client

3. 在“起始值”(Start value) 列中，设置 TCP 连接的连接参数。例如，在“RemoteAddress”中输入 TLS 服务器的 IPv4 地址。
4. 在“起始值”(Start value) 列中，设置安全通信的参数。
   - “ActivateSecureConn”：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可建立非安全的 TCP 或 UDP 连接。
   - “TLSServerCertRef”：输入值“2”（引用 TIA Portal 项目 (SHA256) 的 CA 证书），或输入值“1”（引用 TIA Portal 项目 (SHA1) 的 CA 证书）。
   - “TLSClientCertRef”：自身 X.509-V3 证书的 ID。
5. 在程序编辑器中，创建一个 TCON 指令。
6. 将 TCON 指令的 CONNECT 参数与 TCON_IP_V4_SEC 数据类型的变量进行连接。

TLS 服务器的设置

要在 TLS 服务器中建立安全的 TCP 连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个数据类型为 TCON_IP_4_SEC 的变量。

   以下示例中显示了全局数据块"Data_block_1"，其中，定义了数据类型为 TCON_IP_V4_SEC 的变量"SEC 连接 1 TLS 服务器"(SEC connection 1 TLS-Server)。

   该接口 ID 的值为本地 CP（TLS 服务器）中 IE 接口的硬件标识符。

Data_block_1

	Name	Data type	Start value	Comment
1	Static			
2	SEC connection 1 TLS-Server	TCON_IP_V4_SEC		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	InterfaceId	HW_ANY	260	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connetion: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	false	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	10	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	true	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against the
19	TLSServerCertRef	UDInt	6	for Server side: Reference to own X.509 V3 server certificate; fo
20	TLSClientCertRef	UDInt	1	for Client side: add id of own X.509 V3 client certificate; for Serv

图 5-30 IP_V4_SEC_Server

3. 在"起始值"(Start value) 列中，设置 TCP 连接的连接参数。例如，在"RemoteAddress"中输入 TLS 客户端的 IPv4 地址。
4. 在"起始值"(Start value) 列中，设置安全通信的参数。
   - "ActivateSecureConn"：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可建立非安全的 TCP 或 UDP 连接。
   - "TLSServerReqClientCert"：TLS 客户端需具有 X.509-V3 证书。输入值"true"。
   - "TLSServerCertRef"：自身 X.509-V3 证书的 ID。
   - "TLSClientCertRef"：输入值"2"（引用 TIA Portal 项目 (SHA256) 的 CA 证书），或输入值"1"（引用 TIA Portal 项目 (SHA1) 的 CA 证书）。
5. 在程序编辑器中，创建一个 TCON 指令。
6. 将 TCON 指令的 CONNECT 参数与 TCON_IP_V4_SEC 数据类型的变量进行连接。

上传设备作为新站

在将带有证书的组态进而组态的开放式用户安全通信作为新站上传到 STEP 7 项目中时，与 CPU 的证书不同，CP 的证书不会上传。在将设备加载为新站后，在 CP 的设备证书表格中不会包含更多证书。

上传后，需再次对证书进行组态。否则，重新加载组态将导致 CP 之前存在的证书删除，无法进行安全通信。

通过 CPU 和 CP 接口的 OUC 安全连接（操作相似）

- 连接资源：
  OUC 和安全 OUC 之间无差别。编程的 OUC 安全连接将使用诸如 OUC 连接之类的连接资源，而不考虑与该站通信的 IE/PROFINET 接口。
- 连接诊断：
  OUC 和 OUC 安全连接诊断之间无差别。
- 将带有 OUC 安全连接的项目加载到 CPU 中：
  如果还需加载证书，则只能在 CPU 处于 STOP 模式下时进行。有关在 RUN 模式下更新已加载证书的要求，请参见"提示：在 RUN 模式中更新下载的证书 (页 67)"。
  建议：加载到设备 > 硬件和软件 (Load to device > Hardware and software)。原因：确保程序与安全 OUC、硬件配置和证书之间的一致性。
  仅当所需的证书位于模块中时，才能在 RUN 模式下重载使用其它 OUC 安全连接的块。

5.6.4.5 通过 Modbus TCP 进行 OUC 安全连接

要进行 Modbus TCP 安全连接，需手动创建一个 TCON_IP_V4_SEC 或 TCON_QDN_SEC 系统数据类型的数据块，分配相应参数并在 MB_Server 或 MB_CLIENT 指令中直接调用该数据块。

要求：

- S7-1500 CPU 固件版本 V2.5 或更高版本
- Modbus 客户端（TLS 客户端）可通过网络中的 IP 通信访问 Modbus 服务器（TLS 服务器）。
- TLS 客户端和 TLS 服务器具有所需的全部证书。

与 Modbus TCP 服务器建立 Modbus TCP 安全连接的示例

在下文章节中，介绍如何通过 Modbus TCP 在 Modbus TCP 客户端与 Modbus TCP 服务器之间建立开放式用户安全通信。

要在 Modbus TCP 客户端（TLS 客户端）与 Modbus TCP 服务器（TLS 服务器）之间建立安全连接并设置邮件服务器的 Ipv4 地址，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个 TCON_IP_V4 SEC 数据类型的变量。

Data_block_1

	Name	Data type	Start value	Comment
1	Static			
2	SEC_ModbusTCP_1	TCON_IP_V4...		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	InterfaceId	HW_ANY	64	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	15	connection reference / identifier
6	ConnectionType	Byte	11	type of connection: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of B...		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	10	IPv4 address
13	ADDR[4]	Byte	100	IPv4 address
14	RemotePort	UInt	502	remote UDP/TCP port number
15	LocalPort	UInt	502	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	0	Bit 0: Just for client side: validate given IPv4 address against the subjectAlt
19	TLSServerCertRef	UDInt	2	for Server side: Reference to own X.509 V3 server certificate; for Client side:
20	TLSClientCertRef	UDInt	7	for Client side: add id of own X.509 V3 client certificate; for Server side: add

图 5-31 TCON_IP_V4_SEC

3. 在“起始值”(Start value) 列中，设置 TCP 连接的连接参数。例如，在“MailServerAddress”中输入邮件服务器的 IPv4 地址。
4. 在“起始值”(Start value) 列中，设置安全通信的参数。例如，在“TLSServerCertRef”中输入通信伙伴的 CA 证书的证书 ID。
   - “ActivateSecureConn”：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可设置一个 Modbus TCP 非安全连接。
   - “TLSServerCertRef”：对 Modbus TCP 服务器中 X.509 V3 (CA) 证书的引用，TLS 客户端使用该信息对 Modbus TCP 服务器进行身份验证。
5. 在程序编辑器中，创建一个 MB_CLIENT 指令。
6. 将 MB_Client 指令的 CONNECT 参数与 TCON_IP_4_SECC 数据类型的变量进行互连。

5.6.4.6 通过电子邮件实现 OUC

通过 CPU 接口建立与邮件服务器的安全连接

要建立与邮件服务器的安全通信，需手动创建一个 TMAIL_V4_SEC 或 TMAIL_QDN_SEC 系统数据类型的数据块，分配参数并在 TMAIL_C 指令中直接调用该数据块。

要求：

- TMAIL_C 指令，版本 V5.0 或更高版本
- STEP 7 V15 及更高版本
- S7-1500 CPU V2.5 及更高版本
- 已经为 CPU（TLS 客户端）分配了邮件服务器（TLS 服务器）的所有 CA 证书，且组态已下载到 CPU 中。
- 在 CPU 中，设置当前的日期和时间。

与邮件服务器建立安全连接的操作过程

与邮件服务器建立安全连接时，可选择以下两种操作过程：

- SMTPS：客户端尝试与邮件服务器立即建立 TLS 连接（"握手"过程）如果邮件服务器不支持 TLS，则不建立连接。
- STARTTLS：客户端与邮件服务器建立 TCP 连接。客户端将发送一个请求，"更新"当前通过 TCP 连接与 TLC 安全连接的连接。如果邮件服务器支持 TLS，则客户端将发生该命令建立安全连接。为此，邮件服务器将使用 SMTP 命令"STARTTLS"。之后，客户端将建立与邮件服务器的安全连接。优势：如果邮件服务器不支持 TLS，则客户端和邮件服务器之间可进行非安全通信。

在块参数"MAIL_ADDR_PARAM"的数据类型中设置"远程端口"(Remote Port)，可定义进行通信的进行。

表格 5-6 SMTPS 和 STARTTLS 进程的端口号

进程	端口
SMTPS：	465[1]
STARTTLS：	任意端口 (≠465)[2]

[1] 指令 TMAIL_C 仅在 465 端口采用 SMTPS 通信协议。其它所有端口将使用 STARTTLS 通信协议。

[2] 根据 RFC，邮件服务器使用端口 25，而 STARTTLS 安全连接则使用端口 587。RFC 不建议 SMTP 使用其它端口号，否则无法确保与邮件服务器的通信成功。

示例：通过 IPv4 与邮件服务器建立安全连接

在以下章节中，将介绍如何使用 TMAIL_C 通信指令与 IPv4 邮件服务器建立安全连接。

要通过邮件服务器的 IP4 地址建立安全连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个 TMAIL_V4_SEC 数据类型的变量。

   在以下示例中，显示了一个全局数据块“MailConnDB”，其中，定义了数据类型为 TMAIL_V4_SEC 的变量“MailConnectionSEC”。

MailConnDB

Name	Datentyp	Startwert	Kommentar
Static			
MailConnectionSEC	TMail_V4_SEC		
InterfaceId	HW_ANY	"Local~CP_1543-1_1~Ethernet-Schnittstelle_1"	Use HW-identifier of the IE-interface to specify th
ID	CONN_OUC	100	connection reference / identifier
ConnectionType	Byte	16#20	type of connection 16#20=32=TMail_V4 or TMai
ActiveEstablished	Bool	true	active / passive connection establishment
WatchDogTime	Time	T#5000ms	watchdog time to monitor SMTP server associat
MailServerAddress	IP_V4		IPv4 address of mail server
ADDR	Array[1..4] of Byte		IPv4 address
ADDR[1]	Byte	144	IPv4 address
ADDR[2]	Byte	145	IPv4 address
ADDR[3]	Byte	2	IPv4 address
ADDR[4]	Byte	20	IPv4 address
UserName	String[254]	'myName'	user name which is necessary to login into the
PassWord	String[254]	'myPW'	user password which is necessary to login into
From	EMAIL_ADDR		source mail address
LocalPartPlusAt...	String[64]	'Mustermann@'	local part of e-mail address plus "@" sign
FullQualifiedD...	String[254]	'siemens.com'	full qualified domain name part of e-mail addre
RemotePort	UInt	587	remote TCP port number
ActivateSecureConn	Bool	TRUE	activate the security functionality of that conne
ExtTLSCapabilities	Byte	16#0	for further capability extensions of the TLS hand
TLSServerCertRef	UDInt	7	Reference to the X.509 V3 (CA-) certificate of the

图 5-32 数据类型 TMAIL_V4_SEC

3. 在“起始值”(Start value) 列中，设置 TCP 连接的连接参数。例如，在“MailServerAddress”中输入邮件服务器的 IPv4 地址。

   说明

   连接参数接口 ID

   请注意，在数据类型 TMAIL_V4_SEC 中，如果指令 TMAIL_C 的指令版本为 V5.0 或更高版本，则需在接口 ID 中输入值“**0**”。此时，CPU 将自行搜索适用的本地 CPU 接口。

4. 在“起始值”(Start value) 列中，设置安全通信的参数。例如，在“TLSServerCertRef”中输入通信伙伴的 CA 证书的证书 ID。
   - “ActivateSecureConn”：激活该连接的安全通信。如果该参数的值为 FALSE，则忽略后面的安全参数。此时，可建立非安全的 TCP 或 UDP 连接。
   - “TLSServerCertRef”：对电子邮件服务器中 X.509 V3 (CA) 证书的引用，供 TLS 客户端用来对邮件服务器进行身份验证。
5. 在程序编辑器中，创建一个 TMAIL_C 指令。

6. 将 TMAIL_C 指令的 MAIL_ADDR_PARAM 参数与 TMAIL_V4_SEC 数据类型的变量进行互连。
   在以下示例中，TMAIL_C 指令的 MAIL_ADDR_PARAM 参数已与“MailConnectionSEC”变量（TMAIL_V4_SEC 数据类型）进行互连。

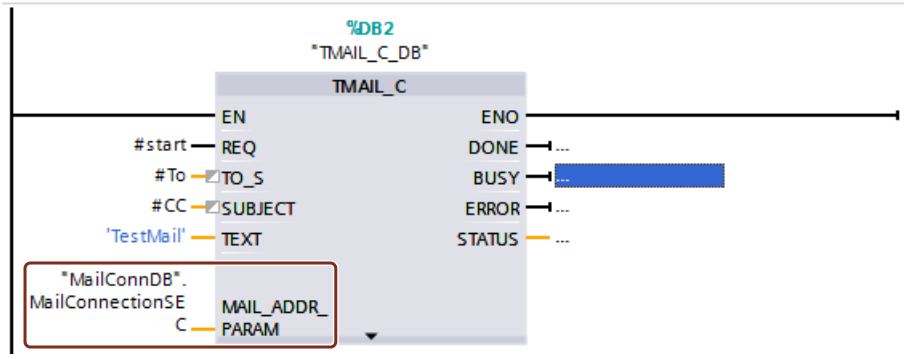


图 5-33 TMAIL_C 指令

通过通信模块接口与邮件服务器建立安全连接

要通过一个通信模块与邮件服务器建立安全通信，则需手动创建一个系统类型为 TMAIL_V4_SEC、TMAIL_QDN_SEC 或 TMAIL_V6_SEC 的数据块，分配参数并在 TMAIL_C 指令中直接调用该数据块。

要求：

- TMAIL_C 指令，版本 **V4.0**
- S7-1500 CPU 固件版本 V2.0 及以上版本，通信模块 CP 1543-1 固件版本 V2.0 及以上版本
- ET 200SP CPU 固件版本 V2.0 及以上版本，通信模块 CP 1542SP-1 (IRC) 固件版本 V1.0 及以上版本
- 已将邮件服务器的所有 CA 证书分配给 CP（TLS 客户端），而且已将组态下载到 CPU 中。
- 在 CPU 中，设置当前的日期和时间。

有关如何通过通信模块的接口与邮件服务器建立安全连接的信息，请参见 STEP 7 在线帮助。

应用示例

通过本应用示例 (https://support.industry.siemens.com/cs/cn/zh/view/46817803)，介绍了如何使用 S7-1500 或 S7-1200 站的 CP 与电子邮件服务器建立安全连接，以及通过默认应用程序“TMAIL_C”从 S7 CPU 发送电子邮件。

更多信息

有关系统数据类型 TMail_V4_SEC 和 TMAIL_QDN_SEC 的更多信息，请参见 STEP 7 在线帮助。

有关安全通信的更多信息，请参见“安全通信 (页 47)”部分。

5.6.5 PG/HMI 间安全通信

5.6.5.1 基于标准化安全机制的 PG/HMI 通信

在 V17 及以上版本中集成有最新型控制器和最新型 HMI 设备，TIA Portal、STEP 7 和 WinCC 的主要组件可实现创新型 PG/PC 和 HMI 标准安全通信（简称为 PG/HMI 通信）。

具体涉及以下 CPU 系列：

- S7-1500 控制器系列固件版本 V2.9 及以上版本
- S7-1200 控制器系列固件版本 V4.5 及以上版本
- 软件控制器固件版本 V21.9 及以上版本
- SIMATIC 启动控制器固件版本 V2.9 及以上版本
- PLCSim 和 PLCSim Advanced 版本 V4.0

HMI 组件经更新以支持 PG/HMI 间安全通信：

- 使用 WinCC 精简版、精智版和高级版组态的面板或 PC
- 安装有 WinCC 专业版运行系统 的 PC
- WinCC Unified PC 和精智面板

还更新了 V6.1 及以上版本的 SINAMICS RT SW 和 V17 及以上版本的 STARTDRIVE。

PG/HMI 通信的特性

PG 通信和 HMI 通信最显著的一个特点是简单：在安装 TIA Portal 的编程设备和 CPU 之间建立在线连接（例如，以下载程序）只需几步简单的操作。此在线连接基于公认的 SIMATIC 通信标准，可满足机密性和完整性等方面的要求。

在将机器和系统集成到开放 IT 环境过程中，必须确保编程设备/HMI 设备与 CPU 之间的通信不仅要有效保护敏感数据的完整性和机密性，同时还要确保其符合公认的安全标准，从而能够应对未来的挑战。

在 TIA Portal 版本 V14 中，基于用户程序的"开放式用户通信"过程已扩展为"**安全的**开放式用户通信"机制。同时还建立了其它基于证书的通信机制（HTTPS、Secure SMTP over TLS 或 OPC UA）。在 TIA Portal 版本 V17 即以上版本中，还对 PG/HMI 通信进行了升级：在此，TLS（传输层安全）协议用于采用标准化安全机制的 PG/HMI 间安全通信。

更改的内容

用于提高安全性的附加可选密码

上述设备的组态形式中最显著的变化是能够分配密码以保护相应 CPU 的敏感组态数据。敏感组态数据包括诸如私钥等数据，基于证书的协议正常运行（安全通信）需要私钥，对于 TIA Portal V17 及以上版本，PG/HMI 通信也需要私钥。在 TIA Portal 中输入密码时，可以使用策略设置来检查已分配的密码。这样，可确保企业遵循既定的密码策略。

如果计算机或系统已采用其它类似保护，而无需实施基于西门子工业深度防御机制的保护措施，则无需进行密码分配。如果已采取相应措施保护 TIA Portal 项目和 CPU 组态防止未经授权的访问，则可以不使用密码。

⚠警告

如果不使用密码，则私钥仅获得弱保护

请注意，如果未使用密码来保护受信任的组态数据，则安全通信所需证书的私钥仅获得弱保护。

PG/HMI 与 CPU 之间基于证书的通信

由于 PG/HMI 通信基于证书，因此调试过程中要求接受服务器证书。

通过其它参数分配选项，可以确定 CPU 运行期间的行为：例如，可以指定 CPU 允许连接到不支持 PG/HMI 间安全通信的设备。

维护/更换部件方案

为了在更换部件方案中更换 CPU 时不发生故障，必须遵守特定的规则（参见"更换部件方案的规则 (页 79)"）。

更多信息

有关如何保护机密组态数据的概览，请参见"保护机密的组态数据 (页 68)"部分。

5.6.5.2 PG/HMI 间安全通信的其它设置

除了分配用于保护机密 PLC 组态数据的密码外，还提供其它设置选项以确定 CPU 运行期间的行为。

PG/PC 和 HMI 通信模式

可以设置 CPU 与编程设备和 HMI 设备的通信方式：

- 仅通过 PG/HMI 间安全通信
- 通过 PG/HMI 间安全通信和先前使用的 PG/HMI 通信（简称为"传统的 PG/HMI 通信"）。

操作步骤

1. 在 CPU 属性中，导航至区域"保护与安全 > 连接机制"(Protection & Security > Connection mechanisms)。
2. 选择要使用的选项。

选择证书或生成新证书

如果选择用于 PG/HMI 通信的连接机制，则可以选择符合条件的 PLC 通信证书来保护连接，或者由 TIA Portal 生成证书。如果已分配密码或已取消激活保护机密 PLC 组态数据的选项（即未设置密码），则"保护与安全 > 连接机制"(Protection & Security > Connection mechanisms) 中已预先设置了具有适当设置和有效默认名称的证书。

操作步骤

如果要通过 TIA Portal 生成新证书，或者要选择其它现有证书：

1. 在"PLC 通信证书"(PLC communication certificate) 字段中，单击三个点以展开该字段。
2. 选择所需证书，或单击"添加"(Add) 按钮。
3. 添加证书时，将显示一个包含证书设置选项的对话框。
   用于设置"TLS 服务器"，可以更改其它参数（例如名称、哈希算法）。

应用证书管理的通用规则。例如，如果要生成 CA 证书，则必须选择"证书管理器的全局设置"(Global settings for the certificate manager) 选项。此外，也可选择生成自签名 PLC 证书。

更多信息

有关证书管理主题的说明，请参见"使用 TIA Portal 进行证书管理 (页 57)"部分。

5.6.5.3 PG 与 CPU 之间基于证书的通信的提示

基于证书的 PG/PC 通信（PG/PC 间安全通信）意味着 CPU 的通信伙伴（安装了 TIA Portal 的编程设备）必须信任 CPU 的设备证书，才能下载连接。

简而言之，从 TIA Portal 的角度来说，可使用以下方式信任 CPU 的证书：

- 安装了 TIA Portal 的编程设备已具有 CPU 的设备证书，例如，已在项目中创建或导入证书。此时，将系统自动运行证书检查，而无任何提示。
- 安装了 TIA Portal 的编程设备不具有 CPU 的设备证书，例如，CPU 通过"可访问站"(Accessible stations) 确定，而在项目中不可用。此时，TIA Portal 将询问 TIA Portal 用户该证书是否可信。只有通过大量的工作才能做出判断，因为 CPU 不在眼前，因此无法立即鉴定真伪。
- 安装了 TIA Portal 的编程设备具有 CA 证书（证书颁发机构），并且 TIA Portal 可通过网络访问的所有 CPU 都具有该 CA 证书颁发的设备证书。

  该解决方案的优势：即使通信伙伴的设备证书在 TIA Portal 中不可用，TIA Portal 仍可以自动检查设备证书。

下文将详细介绍 CA 证书（证书颁发机构）解决方案。

要求

可以使用 TIA Portal 的证书颁发机构创建 CPU 的设备证书，并使用现有 CA 证书为设备证书签名。还可以在 TIA Portal 中导入并使用另一个证书颁发机构。

必须启用证书管理器的全局安全策略。只有完成此设置，才能生成 CA 签名的证书。

另请参见"使用 TIA Portal 进行证书管理 (页 57)"

导出编程设备的 CA 证书

要在创建和分配证书后导出相应的 CA 证书，请按照以下步骤进行操作：

1. 打开项目树中全局安全设置下的证书管理器。
2. 针对要导出的证书，选择“CA 证书”(CA certificates) 表。
3. 单击右键，打开所选证书的快捷菜单。
4. 单击“导出”(Export)。
5. 选择证书的导出格式和存储位置。

在 TIA Portal 中存储 CA 证书

为确保安装了 TIA Portal 的编程设备能够识别导出的证书从而启用自动证书检查，请按照以下步骤进行操作：

1. 将上一步骤中导出的 CA 证书复制到以下目录：

   C:\ProgramData\Siemens\Automation\Certstore\Trusted

2. 启动 TIA Portal。

   在巡视窗口的“信息”(Info) 选项卡中，每个 CA 证书对应显示一条消息，说明该 CA 证书是否可以成功传输到 TIA Portal 的 CA 存储区。

   如果出错，并不输出详细原因。

向 TIA Portal 证书吊销列表 (CRL) 添加设备证书

如果出现关联的密钥不再安全等情况，可以选择将设备证书单独添加到证书吊销列表 (CRL)。

当 TIA Portal 与设备证书位于证书吊销列表中的 CPU 建立连接时，TIA Portal 中将出现一个对话框，询问是否仍要信任该证书。如果拒绝，将无法建立连接。

要向证书吊销列表中添加设备证书，请按照以下步骤操作：

1. 将设备证书复制到以下目录：

   C:\ProgramData\Siemens\Automation\Certstore\CRL

2. 启动 TIA Portal。

   在巡视窗口的“信息”(Info) 选项卡中，每个证书对应显示一条消息，说明该证书是否可以成功传输到 TIA Portal 的 CRL 存储区。

   如果出错，并不输出详细原因。

5.6.5.4 从下载到运行就绪的 CPU 行为

为确保 CPU 与编程设备或 HMI 设备之间的通信安全，必须首先具有证书。用于生产运行的证书仅在项目下载到 CPU 之后发布。

为了保障初始下载过程的安全，CPU 首先创建一个自签名证书。下文中介绍了建立连接的不同阶段。

关于初始建立连接并进而下载到 CPU 的要求

- CPU 中未设置保护机密 PLC 组态数据的密码。

  如果该 CPU 已设置并因此设置有一个保护机密 PLC 组态数据的密码，则该密码必需与待加载项目的密码相匹配。
- 具有 CPU 组态（包括机密 PLC 组态数据的密码）和用户程序的项目可供使用。
- CPU 处于 STOP 模式。
- 编程设备和 CPU 直接互连并且位于受保护的环境中；即，可以识别要下载的 CPU，并控制 CPU 与编程设备之间的连接。

首次与 CPU 建立连接 - 配置阶段

用于下载 CPU 而建立的第一个连接采用 PG/HMI 间安全通信并由 TLS 程序提供安全保障。

但 CPU 可使用制造商的设备证书（如果有），或使用自签名的证书建立连接。在该阶段中，此 CPU 仅能有限范围内使用。在此阶段中，CPU 将等待基于密码的密钥信息。即，保护机密 PLC 组态数据的密码。此阶段下称配置阶段。诊断缓冲区中的消息指示 CPU 处于配置阶段。

项目下载到 CPU 中后，CPU 会接收项目数据：

- 硬件配置，包括用于安全通信（OPC UA、HTTPS、安全 OUC、PG/HMI 间安全通信）的已组态证书
- 用户程序

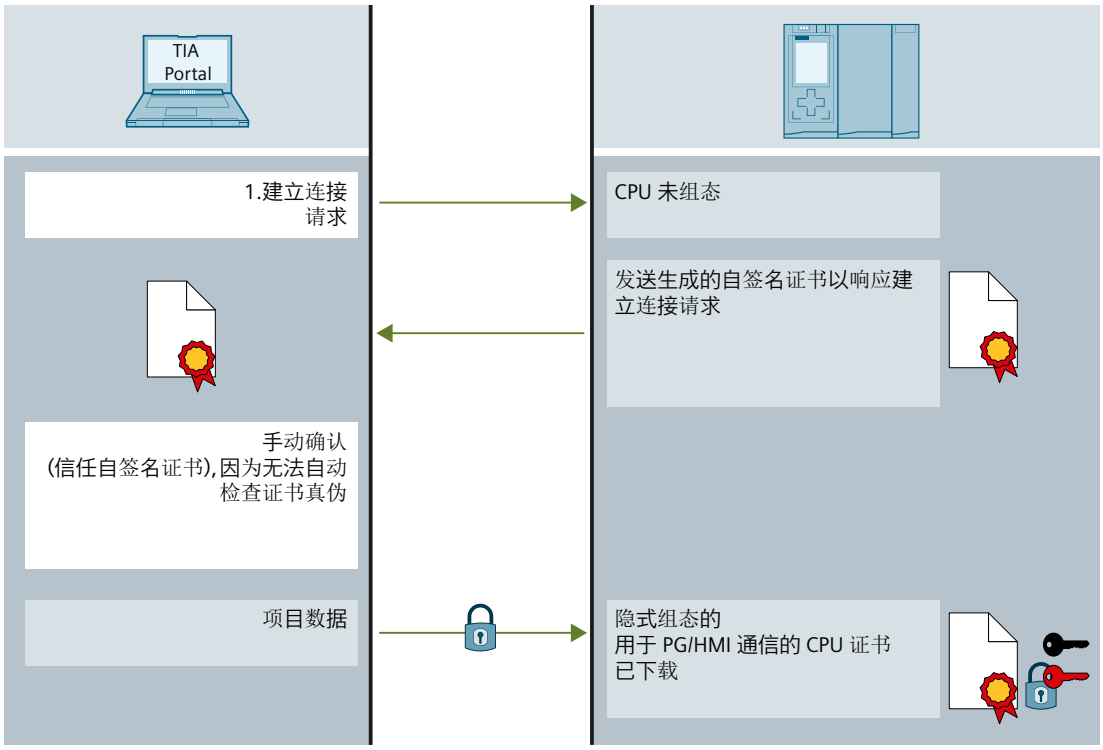


图 5-34 连接建立、配置阶段

> ⚠ 警告
>
> **调试期间可能存在的安全风险**
>
> 在调试过程中，CPU 提供制造商的设备证书（如果有）或自签名证书，必须信任该证书才能建立连接。仅当编程设备与 CPU 处于受保护网络、并彼此直接相连时，才会信任此证书。在不受保护的环境中，这些证书可能被操纵，允许攻击者访问编程设备/HMI 与 CPU 之间的通信（例如通过中间人攻击）。

配置阶段结束

TIA Portal 不会在项目中存储机密 PLC 组态数据的密码本身或通过密码生成的密钥信息。

因此，首次下载项目或下载新项目时，会在对话框中请求输入密码，并将该密码作为密钥信息传送到 CPU。只有在执行此步骤之后，CPU 才能使用受保护的 PLC 组态数据 - 这样便可完成配置阶段，CPU 才能开始运行。

如果未使用密码保护机密 PLC 组态数据，则首次下载 CPU 时无需输入密码。此时，对 PG/HMI 数据通信无影响；但需注意，机密的 PLC 组态数据（如，私钥）几乎无任何保护，无法防止未经授权的访问。

PG/HMI 通信启动

当 CPU 已下载并收到用于 PG/HMI 间安全通信的 CPU 证书后，编程设备将再次连接 - 此时基于下载的 CA 证书。

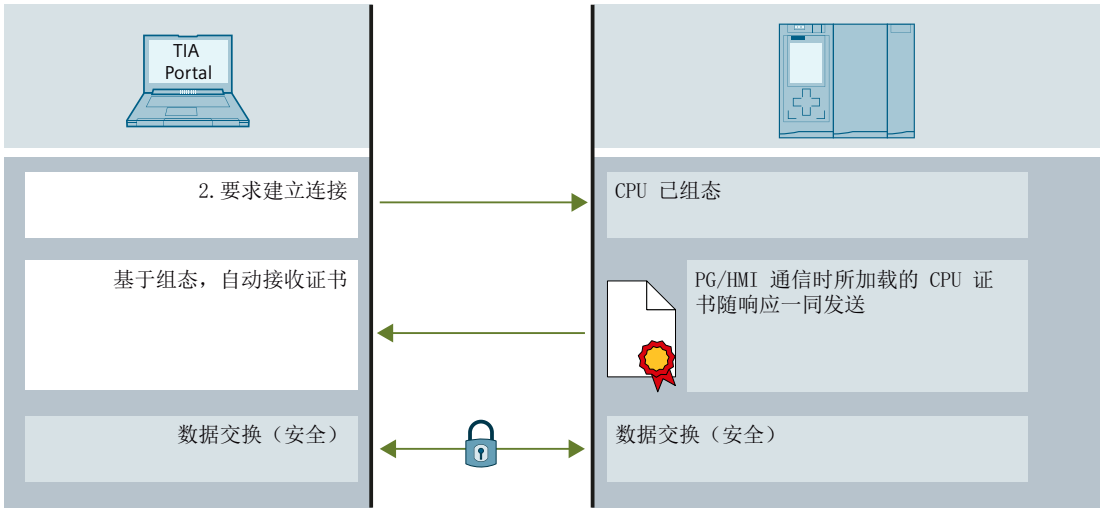


图 5-35 PG/HMI 通信启动

5.6.5.5 使用 HMI 安全通信

在 TIA Portal V17 及以上版本中，如果 CPU 和 HMI 设备均满足 HMI 安全通信要求，则可使用这种通信方式。

要使用 HMI 安全通信，HMI 设备可在建立通信连接时通过 CPU 发送的 PLC 通信证书对该 CPU 进行身份验证，确定该 CPU"可信"。仅当满足以上条件时，才能进行 HMI 安全通信。

在本章节中，将介绍各 HMI 设备将 PLC 通信证书手动标记为"可信"的具体措施。

要求

- CPU 和 HMI 设备支持 HMI 安全通信。
- 当前项目位于 CPU 中（TIA Portal V17 及更高版本）。

组态 HMI 安全通信

1. 组态 HMI 设备的报警视图。

   说明

   如果报警视图缺失，则无法设备连接错误。

2. 组态 CPU 中所需的安全设置。选择 PLC 通信证书，保护 HMI 连接安全；或通过 TIA Portal 生成一个 PLC 通信证书。
3. 组态 CPU 与 HMI 设备间的 HMI 连接。
4. 将项目下载到 CPU 和 HMI 设备中。在项目传送过程中，系统将 PLC 通信证书传送到 CPU 和 HMI 设备中。必要时，还将传输 CA（证书颁发机构）证书。

   说明

   更新和加载 CPU 组态需要重载到 HMI 设备中

   如果更改 CPU 组态并且此更改导致 PLC 通信证书更改（例如，更改接口的 IP 地址时），则还必须重载所连接 HMI 设备的 PLC 通信证书。

   否则，HMI 设备和 PLC 之间无法进行通信。

将 PLC 通信证书设置为可信

连接建立时，CPU 将该 PLC 通信证书传送到 HMI 设备中。

- 如果该 PLC 通信证书在 HMI 设备中的状态已标记为“可信”，则 CPU 与 HMI 设备间将自动建立一条 HMI 安全通信。
- 如果该 PLC 通信证书在 HMI 设备中的未标记为“可信”，则在 HMI 设备的报警视图中将显示一条消息指示该 CPU 不可信，并提供一个错误代码。

  此时，需在 HMI 设备上将该 PLC 通信证书标记为“可信”。

根据 HMI 设备类型，执行以下操作步骤。

第二代精简面板

1. 在 Start Center 中，选择“Settings > Internet Settings > Certificate store”。
2. 在“Available certificates in Device”列表中，选择该 CPU 的 PLC 通信证书。
3. 按下“Trust”。
4. 重新启动 HMI 运行系统软件。

Unified 系列精智面板

1. 打开“控制面板”(Control Panel)。
2. 选择“Security > Certificates”。
3. 在“Certificate store”选择列表中，选择条目“Other Certificates”。
4. 在“Other certificates”列表中，选择该 CPU 的 PLC 通信证书。
5. 按下“Trust”。
6. 重新启动 HMI 运行系统软件。

精智面板，第二代移动面板

1. 通过 Windows CE 桌面图标“My Device”，打开文件管理器。
2. 浏览到目录“\flash\simatic\SystemRoot\OMS\Untrusted”。该 CPU 的 PLC 通信证书位于该目录中。
3. 将该 CPU 的 PLC 通信证书复制到目录“\flash\simatic\SystemRoot\OMS\Trusted”中。
4. 重新启动 HMI 运行系统软件。

如果该 PLC 通信证书在 HMI 设备中的状态已标记为“可信”，则可建立 HMI 安全通信。更多信息，请参见 HMI 设备的操作说明。

5.6.5.6 在 TIA Portal 中使用传统的 PG/PC 通信

在 TIA Portal V17 及以上版本中，TIA Portal 支持与 S7-1200/S7-1500 CPU 固件版本 V4.5/V2.9 及以上版本自动进行“安全”通信。即，连接伙伴自动将各自的连接机制设置为所支持的最高安全连接方式。

仅在特定条件下（参见“兼容性相关信息 (页 107)”），才会回退为原 PG/PC 通信方式，即“传统的 PG/PC 通信”。

如果 CPU 的通信性能较差，而高安全性会影响该 CPU 传输速率。此时可能无需采用较高安全性。

要求

- CPU 间未建立在线连接。
- 如果对 CPU 进行在线访问，则需禁用“仅支持 PG/PC 和 HMI 安全通信”(Only permit secure PG/PC and HMI communication) 选项（“连接机制”(Connection mechanisms) 区域中的 CPU 参数）。
- 通信伙伴位于受保护环境中，如调试阶段。

设置传统的 PG/PC 通信

1. 在“在线”(Online) 菜单中，选择命令“仅使用传统的 PG/PC 通信”(Use only Legacy PG/PC communication)。
2. 选择该菜单命令前的复选框。

结果：TIA Portal V17 以下版本均建立在线连接。

在会话期间，该设置始终有效。项目打开时，“仅使用传统的 PG/PC 通信”(Use only·legacy PG/PC communication) 选项未设置。

启用“仅使用传统的 PG/PC 通信”(Use only·Legacy PG/PC communication) 选项时的特性

- CPU 中保护机密 PLC 组态数据的密码无法在线指定、修改或删除。需要禁用“仅使用传统的 PG/PC 通信”(Use only·Legacy PG/PC communication) 选项才能使用上述功能。
- 设置为仅支持 PG/PC 和 HMI 安全通信的 CPU 无法在线访问。

5.6.5.7 兼容性相关信息

下文中介绍了不同 TIA Portal 版本与不同 CPU 固件版本间的相互关系以及对 PG/HMI 连接类型的影响。

使用 TIA Portal V17 以下版本创建的项目

例如，如果使用 TIA Portal V16 创建适用于 S7-1500 CPU（例如，版本 V2.8）的项目，也可以将使用 TIA Portal V17 实现的相应组态下载到 S7-1500 CPU V2.9 中，例如，在备件方案中 - 与 S7-1500 CPU V2.8 上的组态具有相同的行为。

对于使用 TIA Portal V17 以下版本创建并传送到存储卡的项目，在 S7-1500 CPU V2.9 中也可以正常运行。

但是，使用 TIA Portal V17 及以上版本打开项目，通过更换设备来更新 CPU 的固件版本，并借此将其保存为固件版本为 V2.9 及以上版本的 CPU 后，就会立即应用保护机密 PLC 组态数据的概念（参见"有关保护机密 PLC 组态数据的实用信息 (页 71)"）。不可再使用低于 TIA Portal V17 的版本编辑该项目。

PG/HMI 和 CPU 的连接方式不同

如前几节所述，在 V17 及以上版本中，PG/HMI 设备与 CPU（最新版本）之间的安全 PG/HMI 连接的优势在于采用标准化通信程序 TLS（传输层安全）。

可以选择将 V2.9 CPU 连接到装有 TIA Portal V17 或更高版本的最新编程设备，此外，还可以连接到装有早期运行系统版本的 HMI 设备：设备会相应地自动调整其连接机制。为了能够更好地区分这两种连接机制，我们将先前的程序为"传统方式"（基于 S7 通信的升级版）。

概括地说（此处"PG"代表装有 TIA Portal 的编程设备）：

- PG/HMI 和 CPU 随 V17（或后续版本）提供：使用 TLS 程序。
- PG/HMI 的版本为旧版本 (< V17)：使用传统方式 - 前提是已取消激活 CPU 属性中的选项"仅允许 PG/PC 和 HMI 间安全通信"(Only allow secure PG/PC and HMI communication)。
- CPU 随 V17（或更高版本）提供，连接的多个 PG/HMI 来自 V17（或更高版本）和以前的版本：使用 TLS + 传统方式 - 前提是已取消激活 CPU 属性中的选项"仅允许 PG/PC 和 HMI 间安全通信"(Only allow secure PG/PC and HMI communication)。

当 CPU 状态改变时

如果 CPU 状态因 PG/HMI 间安全通信相关事件而发生改变，则诊断缓冲区会向用户提供相关信息。

示例：

- 成功下载包含已组态密码的组态后，诊断缓冲区将报告 CPU 正在从配置阶段切换为安全模式（TLS 程序）。
- 已将装有 TIA Portal V17 的 PG 连接到 CPU V2.9。若"在线"(Online) 菜单中禁用"仅使用传统的 PG/PC 通信"，将自动建立 PG/HMI 间安全通信（TLS 程序）。

更多信息

有关设备或固件特性（如使用的 TLS 版本）的信息，请参见"设备相关的安全功能 (页 50)"部分。

5.7 SNMP

5.7.1 激活和取消激活 SNMP

网络管理协议 SNMP (Simple Network Management Protocol) 用于对网络拓扑进行监视和诊断。SNMP 采用传输协议 UDP 并具有两个角色：SNMP 管理器（客户端）和 SNMP 代理（服务器）。

- SNMP 管理器用于对网络节点进行监视：
- SNMP 代理则收集各个网络节点的各种网络特定信息，并以结构化的形式存储在 MIB (Management Information Base) 中。多种服务和工具（作为 SNMP 管理器）以这些数据为基础执行详细的网络诊断。

SNMP 还适用于 PROFINET IO 系统，用于管理网络基础设施以及 IO 控制器/IO 设备。

说明

如果取消激活设备的 SNMP 功能，则无法使用各种网络拓扑诊断选项（例如，使用 PRONETA 工具）。

示例：对于在线-离线拓扑比较，TIA Portal 确定实际连接的端口并将 SNMP 用于此功能。

默认设置取决于固件版本

S7-1500 CPU 已集成 SNMP 代理。SNMP 采用不同的默认设置（SNMP 激活或取消激活），与具体的固件版本有关。

对于固件版本低于 V3.0 的 S7-1500 CPU，SNMP 代理默认情况下**激活**，仅可在用户程序中通过数据记录取消激活。

在某些特定条件下，可能需要取消激活 SNMP。示例：

- 网络中的安全规则不允许使用 SNMP。
- 用户可使用自己的通信指令，定制相应的 SNMP 解决方案。

对于固件版本为 V3.0 的 S7-1500 CPU，SNMP 代理默认情况下**取消激活**。如果未下载组态或未插入存储卡，将采用默认设置"已取消激活"。对于固件版本为 V3.0 或更高版本的 S7-1500 CPU，STEP 7 V18 可通过以下方式更改 SNMP 设置：

- 在 TIA Portal 的 CPU 属性中组态 SNMP。
- 通过向 PROFINET 接口传送数据记录在用户程序中激活/禁用 SNMP。

说明

更换部件方案

出于兼容性原因，固件版本为 V3.0 及以上版本的 S7-1500 CPU，如果下载了低版本项目（CPU 固件 < V3.0），其行为与低版本项目中的 CPU 类似：

SNMP 已激活，"public"和"private"社区字符串生效。

组态 SNMP

自 CPU 固件版本 V3.0 以及 TIA Portal V18 起，可在 CPU 属性中更改以下 SNMP 设置：

- 激活 SNMP（默认设置：已取消激活）
- 只读团体字符串（默认值："public"）
- 读写团体字符串（默认值："private"）

相关设置，请参见"高级组态 > SNMP"(Advanced configuration > SNMP) 区域。

从 CPU 固件版本 V3.1 和 TIA Portal 版本 V19 开始，还可以在激活 SNMP 时启用 SNMP 的写保护访问。

团体字符串的含义和属性

SNMP 社区字符串（也称为社区名称）类似于 ID 或密码，用于访问设备（例如路由器）的信息/统计信息。
为了提高访问的安全性，请更改 CPU 属性中的默认社区字符串。SNMP 管理器在收到 SNMP 代理发出的验证请求时通过传输团体字符串来验证自己的身份。

团体字符串作为纯文本传输。

- SNMP 只读操作 (GET) 的默认团体字符串是"public"。
- SNMP 读写操作 (SET) 的默认团体字符串是"private"。

团体字符串的字符数：1-240。

团体字符串支持以下字符：

- a-z
- A-Z
- 0-9
- -
- .

在用户程序中激活/取消激活 SNMP

除了 CPU 属性中的组态，还可在用户程序中激活或禁用 SNMP。为此，将数据记录 0xB071 传送到 CPU 的 PROFINET 接口。在该记录中，包含 SNMP 是否激活/取消激活的代码。无论将数据记录传送到哪个 PROFINET 接口，数据记录都适用于 CPU 的所有接口。

传送 0xB071 数据记录的一种方法：在数据块中定义数据集结构，并在程序循环 OB（例如 OB1）中通过指令"WRREC"（写入数据记录）向 CPU 的 PROFINET 接口传送数据。

为此，请执行以下操作步骤：

1. 在 STEP 7 中，创建一个包含数据记录 0xB071 结构的数据块。
   下表列出了数据记录 0xB071 的结构：

字节	元素	代码	说明
0 到 1	BlockID	0xF003	标头 该数据记录的长度从字节 4"版本"开始计算。
2 到 3	BlockLength	8	
4	版本	0x01	

字节	元素	代码	说明
5	子版本	0x00	
6 到 7	预留	-	-
8 到 11	SNMP 控制器	0.1	0：取消激活 SNMP。 1：激活 SNMP。

2. 例如，在程序循环 OB (OB1) 中通过"WRREC"指令（写入数据记录）将数据记录 0xB071 传送到 CPU 中。将 CPU 中集成的 PROFINET 接口作为硬件 ID。

SNMP 组态和用户程序的相互作用

- SNMP 设置通过用户程序"激活/禁用"不会永久存储在 CPU 中。例如，每次电源关闭/电源接通转换、下载新硬件配置或复位为出厂设置后，组态的设置会再次生效。
- 从 CPU 下载组态（"上传设备作为新站"）时，将采用组态的 SNMP 设置（已激活/已取消激活）。先前由用户程序中的数据记录设置的 SNMP 设置将被忽略。
- 团体字符串只能在组态中更改；团体字符串不能通过用户程序中的数据记录来设置。
  但是，可以通过数据记录激活组态的团体字符串。
  例如：
  已在 S7-1500 CPU 的组态中将 SNMP 设置为取消激活。更改 CPU 属性中的默认团体字符串，然后将组态下载到 CPU 中。
  然后，通过数据记录传输激活 SNMP。
  结果：将采用更改后的团体字符串。
- 对于固件版本低于 V3.0 的 S7-1500 CPU，当激活 SNMP 时，预设的社区字符串（"public"和"private"）始终有效。

5.7.2 通过数据记录传送激活/取消激活 SNMP：CPU 1516-3 PN/DP 的示例

简介

要使用 SNMP 管理网络基础设施、CPU 和 IO 设备，请为 CPU 1516-3 PN/DP 激活 SNMP。以下示例显示了为此需要传送到 PROFINET 接口的 0xB071 数据记录。

要求

- CPU 1516-3 PN/DP 固件版本 V2.0 及更高版本
- STEP 7 版本 V14 或更高版本

解决方法

将数据记录 0xB071 传送到 CPU 的 PROFINET 接口。因此，SNMP 在 CPU 的所有 PROFINET 接口中启用。

以下示例说明如何在全局数据块中创建数据记录并在程序循环 OB（例如 OB1）中将其传送到 PROFINET 接口 (Local~PROFINET_interface_1)。

要在 CPU 1516-3 PN/DP 的已寻址 PROFINET 接口中激活 SNMP，请按以下步骤操作：

1. 创建一个全局数据块。
2. 指定一个名称，例如“ActivateSnmp”。
3. 在“Static”下，创建 0xB071 数据记录的结构（图中：“snmpRecord”）和其它用于传送数据记录的变量。下图显示了数据块结构“ActivateSnmp”。

ActivateSnmp

		Name	Data type	Start value	Re...	Ac...	Wr...	Vis...	Comment
1		Static							
2		snmpWrite	Bool	TRUE		✓	✓	✓	Start writing Data record 16#B071
3		snmpRecord	Struct			✓	✓	✓	Data record 16#B071
4		blockID	UInt	16#F003		✓	✓	✓	Data record ID
5		blockLength	UInt	8		✓	✓	✓	Length of block
6		version	USInt	1		✓	✓	✓	Byte 1 of blockversion
7		subversion	USInt	0		✓	✓	✓	Byte 2 of blockversion
8		reserved	UInt	0		✓	✓	✓	Reserverd for future usage
9		snmpControl	UDInt	1		✓	✓	✓	0: deactivate SNMP, 1: activate SNMP
10		snmpWrDone	Bool	false		✓	✓	✓	writing done
11		snmpWrError	Bool	false		✓	✓	✓	error while writing
12		snmpWrStatus	DWord	16#0		✓	✓	✓	status of writing

图 5-36 全局数据块“ActivateSnmp”的结构

4. 使用“WRREC”（写入数据记录）指令在 OB 程序循环（例如，OB1）中将 0xB071 数据记录传送到 CPU 1516-3 PN/DP。相关示例程序，请参见下一部分。

OB1 中数据记录传送的编程示例

数据记录 0xB071 在以下程序代码中传送：

```
//-----------------------------------------
// Start writing SNMP settings
//-----------------------------------------
IF "ActivateSnmp".snmpWrite THEN
 IF (NOT "ActivateSnmp".snmpWriteDone)
 AND (NOT "ActivateSnmp".snmpWriteError) THEN
 "instWrrec_1"(REQ := "ActivateSnmp".snmpWrite,
 ID := "Local~PROFINET-Schnittstelle_1",
 INDEX := 16#B071,
 DONE => "ActivateSnmp".snmpWriteDone,
 ERROR => "ActivateSnmp".snmpWriteError,
 STATUS => "ActivateSnmp".snmpWriteStatus,
 RECORD := "ActivateSnmp".snmpRecord);
 END_IF;
 IF "ActivateSnmp".snmpWriteError THEN
 ; // add error handling
 END_IF;
 IF "ActivateSnmp".snmpWriteDone THEN
 "ActivateSnmp".snmpWrite := FALSE;
```

```
 END_IF;
END_IF;
```

再次取消激活 SNMP

以上程序代码稍作改动，即可取消激活 SNMP。在用户程序中，为变量""ActivateSnmp".snmpRecord.snmpControl"赋值"0"：

```
"ActivateSnmp".snmpRecord.snmpControl := 0;
```

下次调用"WRREC"指令时，将再次取消激活 SNMP。

5.7.3 使用 S7-1500R/H CPU 激活/取消激活通过 SNMP 进行数据记录传送

在 S7-1500R/H 系统中，可像使用标准 CPU 一样在用户程序中激活/禁用 SNMP。但是，两个 CPU 的 PROFINET 接口（X1、X2、...）具有不同的硬件 ID。例如，左侧 CPU 的 PROFINET 接口 X1 与右侧 CPU 的 PROFINET 接口 X1 具有不同的硬件 ID。S7-1500R/H 系统不会自动同步两个 CPU 的 SNMP 状态（激活/禁用）。通过"WRREC"指令设置的 SNMP 状态（激活/禁用）仅在 PROFINET 接口通过"WRREC"指令进行寻址的 CPU 中生效。

示例：

S7-1500R/H 系统处于"RUN Redundant"系统状态。例如，如果左侧 CPU 的 PROFINET 接口使用"WRREC"指令寻址，则左侧 CPU 的 SNMP 状态将发生变化。右侧 CPU 的 SNMP 状态保持不变。如果左侧 CPU 发生故障或被更换，则 SYNCUP 之后 SNMP 状态不发生变化。

解决方法：

调用"WRREC"指令 2 次。第一次调用"WRREC"指令时，寻址左侧 CPU 的 PROFINET 接口的硬件 ID。再次调用"WRREC"指令。此时，寻址右侧 CPU 的 PROFINET 接口的硬件 ID。

PROFINET 接口 X1 的硬件 ID：

- 左侧 CPU 的 PROFINET 接口 X1 的硬件 ID 为 65164（默认名称：Local1~PROFINET-interface_1）。
- 右侧 CPU 的 PROFINET 接口 X1 的硬件 ID 为 65364（默认名称：Local2~PROFINET-interface_1）。

通过 PROFINET 接口 X1 的相应硬件 ID 进行寻址的方法，也在以下示例中用于调用两个 R/H CPU 的"WRREC"指令。

说明

将数据记录传送到备用 CPU

只有在 S7-1500R/H 系统达到"Run REDUNDANT"系统状态后，才可将数据记录传送到备用 CPU 的已寻址 PROFINET 接口。否则，数据记录无法传送到备用 CPU 的寻址 PROFINET 接口。

S7-1500R/H 系统达到系统状态"Run REDUNDANT"时，将启动 CPU 冗余错误 OB (OB72)。OB72 的"Fault_ID"变量包含错误代码"B#16#03"或"B#16#06"。

示例：两个 R/H CPU 的 WRREC 调用

要通过传送数据记录在两个 CPU 的已寻址 PROFINET 接口中激活/取消激活 SNMP，请按以下步骤操作：

1. 创建一个全局数据块。
2. 指定一个名称，例如“ActivateSnmp”。
3. 在“Static”下，创建 0xB071 数据记录的结构（图中：“snmpRecord”）和其它用于传送数据记录的变量。下图显示了数据块“ActivateSnmp”的结构。

ActivateSnmp

	Name	Data type	Start value	Re...	Ac...	Wr..	Vis...	Comment
1	Static							
2	snmpWrite	Bool	TRUE		☑	☑	☑	Start writing Data record 16#B071
3	snmpRecord	Struct			☑	☑	☑	Data record 16#B071
4	blockID	UInt	16#F003		☑	☑	☑	Data record ID
5	blockLength	UInt	8		☑	☑	☑	Length of block
6	version	USInt	1		☑	☑	☑	Byte 1 of blockversion
7	subversion	USInt	0		☑	☑	☑	Byte 2 of blockversion
8	reserved	UInt	0		☑	☑	☑	Reserverd for future usage
9	snmpControl	UDInt	1		☑	☑	☑	0: deactivate SNMP, 1: activate SNMP
10	plcLeft	Struct			☑	☑	☑	Writing status of left plc
11	snmpWrDone	Bool	false		☑	☑	☑	writing done
12	snmpWrError	Bool	false		☑	☑	☑	error while writing
13	snmpWrStatus	DWord	16#0		☑	☑	☑	status of writing
14	plcRight	Struct			☑	☑	☑	Writing status of right plc
15	snmpWrDone	Bool	false		☑	☑	☑	writing done
16	snmpWrError	Bool	false		☑	☑	☑	error while writing
17	snmpWrStatus	DWord	16#0		☑	☑	☑	status of writing

图 5-37 全局数据块“ActivateSnmp”的结构

4. 将组织块“CPU 冗余错误”(OB72) 添加到用户程序中。相关 OB72 示例程序，请参见下一部分。
5. 打开程序循环 OB (OB1)。
6. 在 OB1 2 中，执行“WRREC”指令以将数据记录传送到两个 CPU 分别寻址的 PROFINET 接口。相关 OB1 示例程序，请参见下一部分。

   结果：0xB071 数据记录被分别传送到寻址的两个 CPU 的 PROFINET 接口。

OB72 和 OB1 组织块的编程示例

打开已添加的 OB72。使用以下程序代码，判断 R/H 系统是否已进入“Run REDUNDANT”状态，并设置“WRREC”指令的启动命令：

```
//-----------------------------------------------
// Check redundancy state and set "snmpWrite"
//-----------------------------------------------
IF #Fault_ID = B#16#03 OR #Fault_ID = B#16#06 THEN
 "ActivateSnmp".snmpWrite := TRUE;
END_IF;
```

打开程序循环 OB (OB1)。使用以下程序代码，可运行 2 条“WRREC”指令以将数据记录传送到两个 CPU 分别寻址的 PROFINET 接口：

```
//----------------------------------------
// Start writing SNMP settings
//----------------------------------------
IF "ActivateSnmp".snmpWrite THEN
 IF (NOT "ActivateSnmp".plcLeft.snmpWrDone)
 AND (NOT "ActivateSnmp".plcLeft.snmpWrError) THEN
 // write SNMP settings for the left PLC
 "instWrrec_1"(REQ := "ActivateSnmp".snmpWrite,
 ID := "Local1~PROFINET_interface_1",
 INDEX := 16#B071,
 DONE => "ActivateSnmp".plcLeft.snmpWrDone,
 ERROR => "ActivateSnmp".plcLeft.snmpWrError,
 STATUS => "ActivateSnmp".plcLeft.snmpWrStatus,
 RECORD := "ActivateSnmp".snmpRecord);
 END_IF;
 IF "ActivateSnmp".plcLeft.snmpWrError THEN
 ; // add error handling for left plc
 END_IF;
 IF (NOT "ActivateSnmp".plcRight.snmpWrDone)
 AND (NOT "ActivateSnmp".plcRight.snmpWrError) THEN
 // write SNMP settings for the right PLC
 "instWrrec_2"(REQ := "ActivateSnmp".snmpWrite,
 ID := "Local2~PROFINET_interface_1",
 INDEX := 16#B071,
 DONE => "ActivateSnmp".plcRight.snmpWrDone,
 ERROR => "ActivateSnmp".plcRight.snmpWrError,
 STATUS =>
 "ActivateSnmp".plcRight.snmpWrStatus,
 RECORD := "ActivateSnmp".snmpRecord);
 END_IF;
 IF "ActivateSnmp".plcRight.snmpWrError THEN
 ; // add error handling for right plc
 END_IF;
 IF "ActivateSnmp".plcLeft.snmpWrDone
 AND "ActivateSnmp".plcRight.snmpWrDone THEN
 "ActivateSnmp".snmpWrite := FALSE;
 END_IF;
END_IF;
```

再次取消激活 SNMP

以上程序代码稍作改动，即可取消激活 SNMP。在用户程序中，为变量""ActivateSnmp".snmpRecord.snmpControl"赋值"0"：

```
"ActivateSnmp".snmpRecord.snmpControl := 0;
```

下次调用"WRREC"指令时，将再次禁用 SNMP。

PG 通信

6

特性

使用 PG 进行通信时，CPU 或其它具备通信功能的模块可在工程师站进行数据交换（例如，PG、PC）。可以通过 PROFIBUS 和 PROFINET 子网进行数据交换。此外，还支持 S7 子网之间的网关。

PG 通信具有装载程序和组态数据、运行测试以及评估诊断信息所需的功能。这些功能集成在具有通信功能的模块的操作系统中。

说明

自 TIA Portal 版本 V17 起，支持将 TLS（传输层安全）协议用于编程设备/HMI 间通信，以确保采用标准化安全机制的编程设备/PC 与 CPU 之间数据交换的安全性。

有关详细信息，请参见以下章节：

- 安全通信要求 (页 68)
- PG/HMI 间安全通信 (页 97)

要求

- 编程设备/PC 与具有通信功能的模块进行物理连接。
- 如果需要通过 S7 路由来访问具有通信功能的模块，则必须在参与的站（S7 路由器和端点）中装载硬件组态。

在线连接步骤

若要实现编程设备通信，必须建立与 CPU 的在线连接：

1. 在 STEP 7 的项目树中选择 CPU。
2. 选择“在线 > 转至在线”(Online > Go online) 菜单命令。
3. 在“转至在线”(Go online) 对话框中，针对在线连接进行以下设置：
   - 在“编程设备/PC 接口类型”(Type of PG/PC interface) 下拉列表中，选择接口类型（如 PN/IE）。
   - 在“PG/PC 接口”(PG/PC interface) 下拉列表中，选择待建立在线连接的 PG/PC 接口（如，工业以太网卡）。
   - 从“连接到接口/子网”(Connection to interface/subnet) 下拉列表，选择用于将编程设备/PC 物理连接的接口或 S7 子网。
   - 如果可以通过 S7 路由器（网关）访问具有通信功能的模块，请从“第一网关”(1st gateway) 下拉列表选择用于连接相关子网的 S7 路由器。

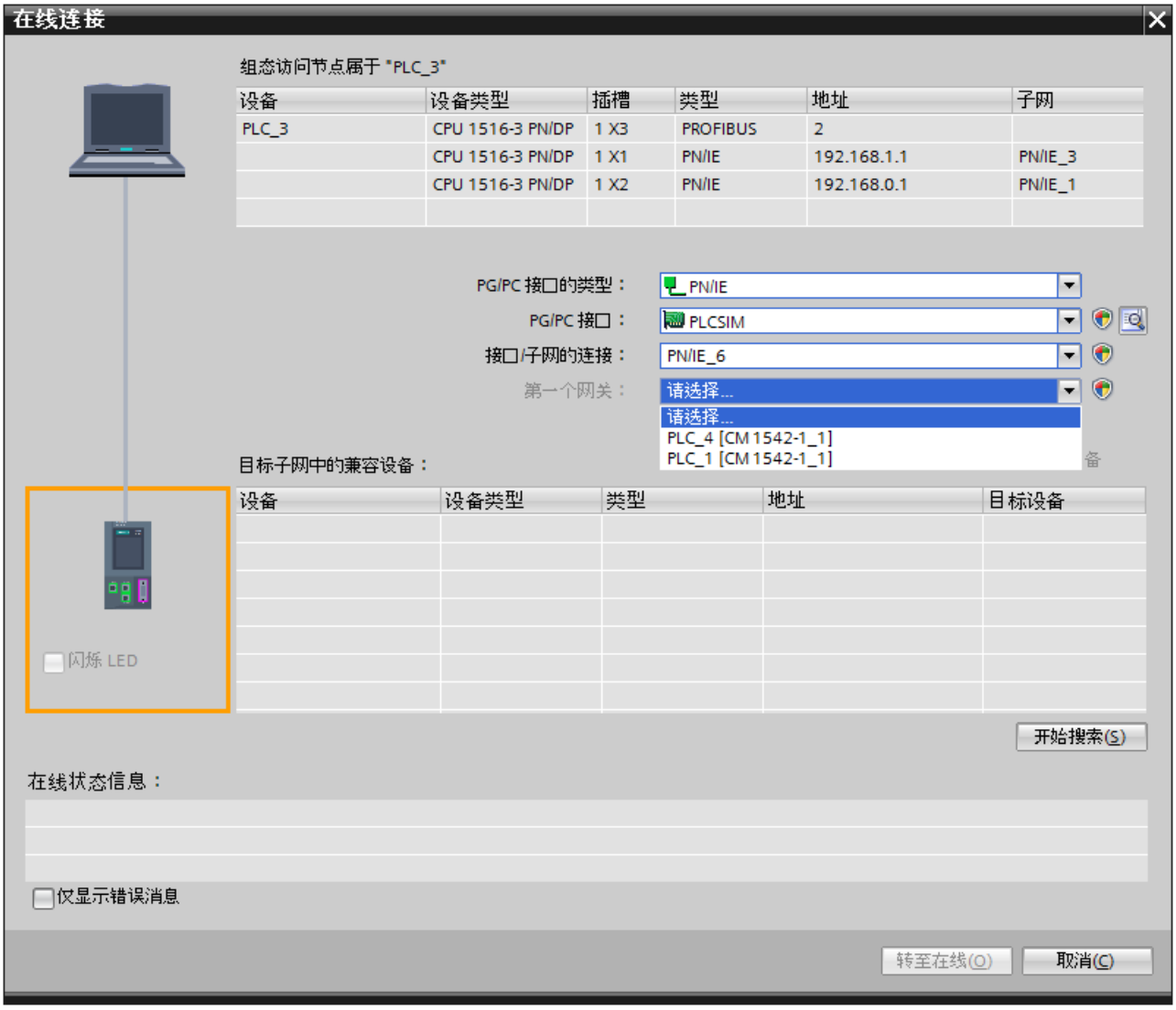


图 6-1 建立 PG 通信

4. 单击"开始搜索"(Start search)。
   PG 通信可寻址的所有设备随后都将显示在表格"目标子网中的兼容设备"(Compatible devices in target subnet) 中。
5. 在表格"目标子网中的兼容设备"(Compatible devices in the target subnet) 中，选择相应的 CPU，并通过"转至在线"(Go online) 进行确认。

更多信息

有关"转至在线"(Go online) 的信息，请参见 STEP 7 的在线帮助。

HMI 通信

7

特性

基于 HMI 通信，CPU 可通过 PROFINET 或 PROFIBUS DP 接口与一个或多个 HMI 设备（如，HMI 精简面板/精智面板/移动面板）进行数据交换，进行操作员监控。通过 HMI 连接进行数据交换。

如果要设置与 CPU 之间的多个 HMI 连接，可使用：

- CPU 的 PROFINET 和 PROFIBUS DP 接口
- 带相关接口的 CP 和 CM

说明

自 TIA Portal 版本 V17 起，支持将 TLS（传输层安全）协议用于编程设备/HMI 间通信，以确保采用标准化安全机制的编程设备/PC 与 CPU 之间数据交换的安全性。

有关详细信息，请参见以下章节：

- 安全通信要求 (页 68)
- PG/HMI 间安全通信 (页 97)

建立 HMI 通信的操作步骤

拖放标签时，例如，将标签从全局数据块中拖入 HMI 画面或 HMI 标签表时，STEP 7 会自动建立 HMI 连接。此外，也可手动建立 HMI 连接。

要建立 HMI 连接，请按以下步骤操作：

1. 在 STEP 7"设备与网络"(Devices & networks) 编辑器的网络视图中，可以在 CPU 的当前组态中组态 HMI 设备。
2. 选择"连接"(Connections) 按钮，并从下拉列表中选择"HMI 连接"(HMI connection)。
3. 在连接的断点（HMI 设备和 CPU）之间拖出一条线。端点将使用颜色突出显示。如果所需的 S7 子网尚不存在，则系统将自动创建。

4. 在“连接”(Connections) 选项卡中，选择 HMI 连接所在的行。

   在“属性”(Properties) 选项卡的“常规”(General) 区域中，将显示 HMI 连接的属性，其中一些属性可以更改。

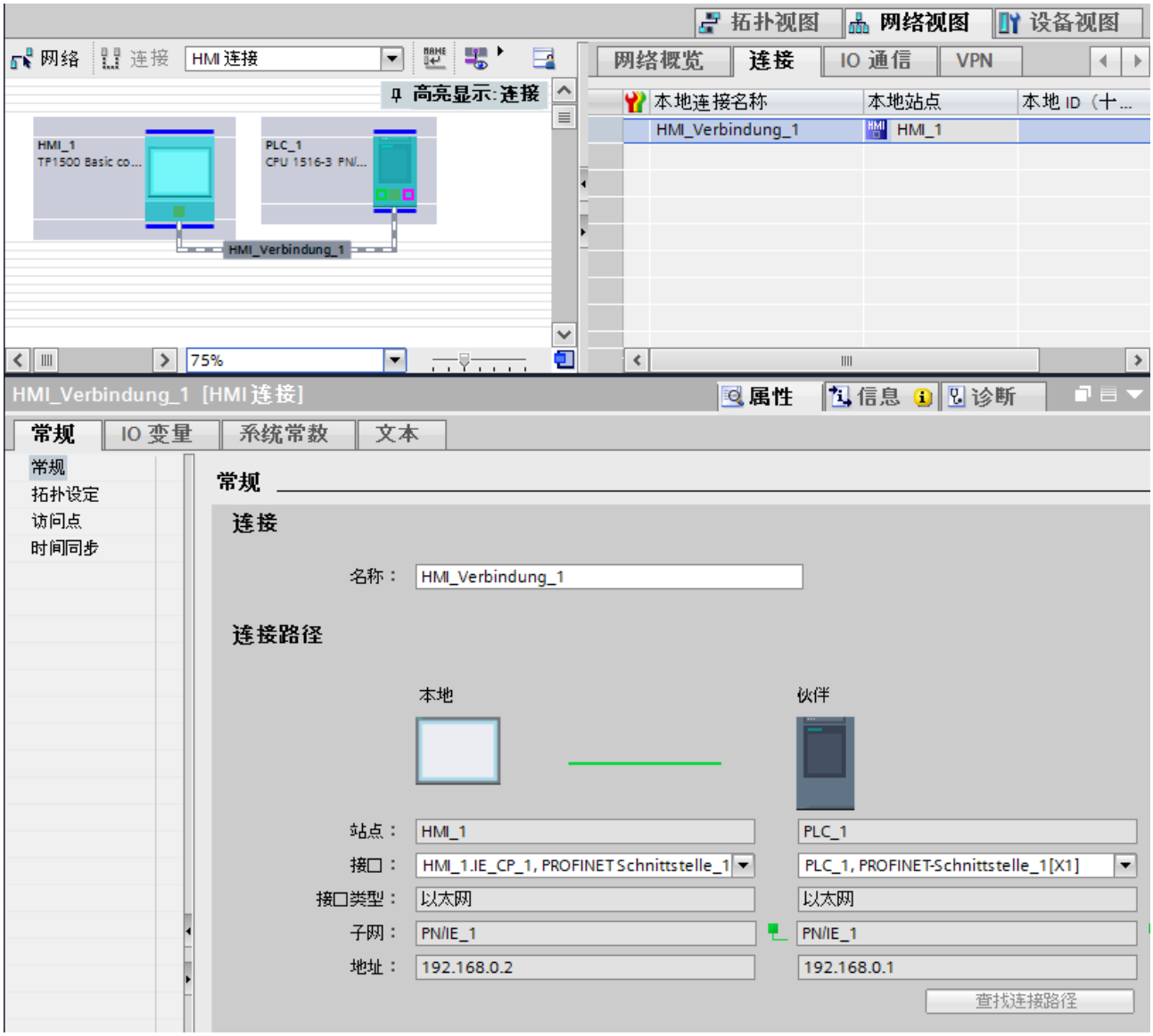


图 7-1 建立 HMI 通信

5. 将硬件配置下载到 CPU。
6. 将硬件配置下载到 HMI 设备。

更多信息

有关 HMI 的 S7 路由的信息，请参见“S7 路由 (页 376)”部分。

有关设置 HMI 连接的信息，请参见 STEP 7 的在线帮助。

开放式用户通信

8

8.1 开放式用户通信概述

开放式用户通信的特性

通过开放式用户通信（也称为"开放式通信"），CPU 可以与具有通信功能的其它模块进行数据交换。开放式用户通信具有以下功能及特性：

- 开放式标准（通信伙伴可以是两个 SIMATIC CPU，也可以是 SIMATIC CPU 和适当的第三方设备）。
- 通过各种协议（在 STEP 7 中称为"连接类型"）进行通信
- 可传输的数据结构上具有高度灵活性。因此，通信设备只要支持这些连接类型，都可以进行开放式数据交换。
- 安全通信：要保护自动化系统的安全，可通过"开放式用户通信"进行安全数据交换。使用"开放式用户安全通信"时，将对发送的数据进行签名并加密，另请参见"开放式用户安全通信 (页 80)"。
- 开放式用户通信适用于各种自动化系统中，具体参见相应手册中的技术规范。
  示例：
  - CPUCPU 的以太网接口（S7-1500、ET 200SP CPU、S7-1500 软件控制器、CPU 1513/1516pro 2 PN）
  - 通信模块通信模块的以太网接口（例如 CP 1543-1、CM 1542-1、CP 1543SP-1）

有关"开放式用户安全通信"的信息，请参见"安全通信 (页 47)"部分。

有关 S7-1500R/H 的信息

有关与 S7-1500R/H 冗余系统进行开放式用户通信的信息，请参见"与冗余系统 S7-1500R/H 进行通信 (页 406)"部分。

8.2 开放式用户通信协议

开放式用户通信协议

以下协议适用于开放式通信：

表格 8-1 开放式通信的传输协议

传输协议	所用接口
TCP，符合 RFC 793 标准	PROFINET/工业以太网
ISO-on-TCP，符合 RFC 1006 (Class 4) 标准	PROFINET/工业以太网
ISO，符合 ISO/IEC 8073 标准	工业以太网（仅 CP 1543-1）
UDP，符合 RFC 768 标准	PROFINET/工业以太网
FDL	PROFIBUS

表格 8-2 开放式通信的应用协议

应用协议	所用传输协议
Modbus TCP	**TCP**，符合 RFC 793 标准
电子邮件	**TCP**，符合 RFC 793 标准
FTP	**TCP**，符合 RFC 793 标准

TCP、ISO-on-TCP、ISO、UDP

在进行数据传输之前，这些协议（UDP 除外）首先会建立与通信伙伴的传输连接。如需防止数据丢失，则可使用面向连接的协议。

采用 UDP 协议时，可以：

- 通过 CPU 的 PROFINET 接口或 CP 1543-1 的工业以太网接口，向 PROFINET 上的一个设备进行单播或向所有设备进行广播。
- 通过 CPU 的 PROFINET 接口或 CP 1543-1 的 PROFINET/工业以太网接口向多播组的所有接收方进行多播

支持的最大多播组数和最大用户数据长度：参见相应设备手册的技术规范。

通过 PROFIBUS 进行通信的协议：FDL

通过 FDL 连接（现场总线数据链路）的数据传输适用于将相关数据块传送到 PROFIBUS 通信伙伴。这些通信伙伴基于符合 EN 50170 标准（第 2 卷）的 FDL 服务 SDA（需要确认的数据发送）对数据进行发送及接收。两个伙伴具有同样的权限；即，每个伙伴都可进行基于事件的发送和接收操作。

基于符合 EN 50170（第 2 卷）标准的 FDL 服务 SDN（无需确认的数据发送）时，可通过 FDL 执行以下操作：

- 通过 CM 1542-5 的 PROFIBUS 接口，向 PROFIBUS 上的所有设备进行广播
- 通过 CM 1542-5 的 PROFIBUS 接口，向一个多播组中的所有接收方进行多播

Modbus TCP

Modbus 协议是是一种基于主站/从站架构的通信协议，采用线形拓扑结构。在 Modbus TCP（传输控制协议）中，数据作为 TCP/IP 数据包进行传输。

只有用户程序中的相关指令才能对通信进行控制。

电子邮件和 FTP

例如，可使用邮件来发送数据块内容的附件（如过程数据）。

可以使用 FTP 连接（FTP = 文件传输协议）与 S7 设备之间双向传输文件。

通信由客户端用户程序中的指令控制。

应用示例：SIMATIC S7-1500 CPU 的 MQTT 发布方

"消息队列遥测传输" (MQTT) 是一种 TCP/IP 层级的简单通信协议。该协议适用于在功能较少的设备间进行消息交换，以及通过非可靠网络进行数据传输。

在本应用示例中，通过一个函数块在 SIMATIC S7-1500 中实施 MQTT 协议。

有关该应用示例，敬请访问 Internet (https://support.industry.siemens.com/cs/cn/zh/view/109772284)。

参见

Syslog (https://support.industry.siemens.com/cs/cn/zh/view/51929235)

8.3 开放式用户通信的指令

简介

通过以下方式，可基于相应的连接（如，TCP 连接）建立开放式用户通信：

- 通过编程通信伙伴的用户程序
- 通过在 STEP 7 的硬件和网络编辑器中组态连接

无论是通过编程建立连接还是通过组态建立连接，都需要在通信双方的用户程序中使用相应的指令发送和接收数据。

通过用户程序建立连接

如果通过编程建立连接，则需在用户程序中使用相应的指令建立和终止连接。

在某些应用领域中，与通过硬件配置建立通信连接相比，通过用户程序静态建立通信连接的优势更为明显。必要时，只需一个特定的应用程序指令即可建立连接。如果选择通过编程建立连接，则将在数据传输结束后还将释放连接资源。

每个通信连接中都需要一个数据结构，用于保存建立连接的参数（例如，TCP 中的系统数据类型“TCON_IP_v4”）。

系统数据类型 (SDT) 由系统提供，这种数据类型预定义的结构不能更改。

各个协议都有自己的数据结构（见下表）。这些参数将保存在系统数据类型（如，TCON_IP_v4）的数据块中（“连接描述 DB”）。

可通过以下两种方式创建带该数据结构的数据块：

- 建议：在对 TSEND_C、TRCV_C 和 TCON 指令的连接进行参数分配期间，在程序编辑器中的属性中自动创建数据块。
- 手动创建这种数据块，进行参数分配并直接写入指令中
  进行以下连接时所需：
  - OUC 安全连接
  - 通过 DNS 进行连接
  - 电子邮件
  - FTP

可以在“连接描述 DB”中修改连接的参数。

该常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/58875807)介绍了如何编程 TCON 指令来建立连接，实现两个 S7-1500 CPU 之间的开放式用户通信。

通过编程建立连接时的协议、系统数据类型和可用指令

下表列出了开放式用户通信的通信协议以及相对应的系统数据类型和指令。

表格 8-3 通过编程建立连接的指令

协议	系统数据类型	指令
TCP	• TCON_QDN • TCON_IP_v4	建立连接并通过以下指令收/发数据： • TSEND_C/TRCV_C 或 • TCON，TSEND/TRCV 或 • TCON，TUSEND/TURCV （可通过 TDISCON 终止连接）
ISO-on-TCP	• TCON_IP_RFC	
ISO，符合 ISO/IEC 8073 (Class 4) 标准	• TCON_ISOnative[1] • TCON_Configured	
UDP	• TCON_IP_v4 • TADDR_Param • TADDR_SEND_QDN • TADDR_RCV_IP	建立连接并通过以下指令收/发数据： • TSEND_C/TRCV_C • TUSEND/TURCV/TRCV （可通过 TDISCON 终止连接）
FDL[1]	• TCON_FDL	建立连接并通过以下指令收/发数据： • TSEND_C/TRCV_C 或 • TCON，TSEND/TRCV 或 • TCON，TUSEND/TURCV （可通过 TDISCON 终止连接）
Modbus TCP	• TCON_IP_v4 • TCON_QDN • TCON_Configured	• MB_CLIENT • MB_SERVER
电子邮件	• TMAIL_v4 • TMAIL_v6 • TMAIL_FQDN	• TMAIL_C
FTP[2]	• FTP_CONNECT_IPV4[3] • FTP_CONNECT_IPV6[3] • FTP_CONNECT_NAME[3]	• FTP_CMD

1 此协议仅适用于 CM 1542-5

2 此协议仅适用于 CP 1543-1

3 用户自定义数据类型

下表列出了开放式用户安全通信的各种不同连接方式以及相对应的系统数据类型和指令。

OUC 安全连接	系统数据类型	指令
S7-1500 CPU 作为 TLS 客户端时，与第三方 PLC（TLS 服务器）进行 TCP 安全连接 S7-1500 CPU 作为 TLS 服务器时，与第三方 PLC（TLS 客户端）进行 TCP 安全连接	• TCON_QDN_SEC	• TSEND_C/TRCV_C • TCON、TSEND/TRCV
在两个 S7-1500 站之建立 TCP 安全连接	• TCON_IP_V4_SEC[1]	
与邮件服务器建立安全连接[2]	• TMAIL_V4_SEC • TMAIL_QDN_SEC	• TMAIL_C（V5.0 或更高版本）
建立 Modbus TCP 安全连接	• TCON_IP_V4_SEC[1]	• MB_Client • MB_Server
	• TCON_QDN_SEC	

1 同样适用于 CP 1543-1

2 CP1543-1 也可使用 TMAIL_C (V4.0) 与邮件服务器建立安全连接

通过连接组态建立连接

通过连接组态建立连接时，需要在 STEP 7 的硬件和网络编辑器中指定连接的地址参数。数据发送和接收指令与通过编程建立连接的相同：

表格 8-4 通过组态建立连接的发送/接收指令

协议	通过组态建立连接的数据发送/接收
支持的指令：	
TCP	通过以下指令发送/接收数据： • TSEND_C/TRCV_C 或 • TSEND/TRCV 或 • TUSEND/TURCV
ISO-on-TCP	
ISO，符合 ISO/IEC 8073 (Class 4) 标准	
UDP	通过以下指令发送/接收数据： • TSEND_C/TRCV_C 或 • TUSEND/TURCV
FDL	通过以下指令发送/接收数据： • TSEND_C/TRCV_C 或 • TSEND/TRCV 或 • TUSEND/TURCV
Modbus TCP	不支持
电子邮件	不支持
FTP	不支持

开放式通信的其它指令

通过用户程序中建立的连接以及通过组态建立的连接，可使用以下指令：

- T_RESET：终止和建立连接
- T_DIAG：检查连接

开放式用户通信的基本示例

有关快速处理开放式用户通信指令的各种函数块 (FB)，敬请访问西门子在线支持。相关函数块及其示例，敬请访问 Internet (https://support.industry.siemens.com/cs/cn/zh/view/109747710)。

更多信息

STEP 7 在线帮助中介绍了：

- 用户数据类型和系统数据类型
- 开放式通信的指令
- 连接参数

有关连接资源的分配和释放的信息，请参见“连接资源的分配 (页 395)”部分。

有关开放式用户安全通信的信息，请参见“开放式用户安全通信 (页 80)”部分。

8.4 通过域名进行寻址的开放式用户通信

自固件版本 V2.0 起，S7-1500 CPU、ET 200SP CPU 和 CPU 1513/1516pro-2 PN 支持通过域名系统 (DNS) 寻址的开放式用户通信。CPU 中集成有 DNS 客户端。在通过 DNS 进行通信的情况下，可使用域名作为 IP 地址的别名来对通信伙伴进行寻址。对于通过 TCP 和 UDP 进行的开放式通信，可通过域名对通信伙伴进行寻址。

通过 DNS 进行通信时，要求网络中必须存在至少一台 DNS 服务器。

对于分配给 S7-1500 软件控制器的所有接口，该款软件控制器支持通过 DNS 进行通信。

通过 DNS 建立通信

CPU 的 DNS 客户端需至少确定一个 DNS 服务器的 IPv4 地址，才能确保 CPU 可通过其域名与通信伙伴建立连接。CPU 最多支持 4 个 不同的 DNS 服务器。

要通过域名建立 S7-1500 CPU 通信，请按以下步骤操作：

1. 在 STEP 7 的网络视图中选择 CPU。
2. 在巡视窗口中，导航至“属性 > 常规 > 高级组态 > DNS 组态”(Properties > General > Advanced configuration > DNS configuration)。
3. 在表格“服务器列表”(Server list) 的“DNS 服务器地址”(DNS server addresses) 列中，输入 DNS 服务器的 IPv4 地址。
   最多可输入 4 个 DNS 服务器的 IPv4 地址。

图 8-1 输入 DNS 服务器地址（以 CPU 1516-3 PN/DP 为例）

通过通信伙伴的域名建立 TCP 连接。

要通过域名进行 TCP 通信，需要手动创建 TCON_QDN 系统数据类型的数据块，然后分配相应参数并在指令中直接调用该数据块。TCON、TSEND_C 和 TRCV_C 指令支持系统数据类型 TCON QDN：

要通过通信伙伴的域名建立 TCP 连接，请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个 TCON_QDN 数据类型的变量。

   在以下示例中，显示了一个全局数据块“Data_block_1”。其中，定义了数据类型 TCON_QDN 的变量“DNS Connection1”。

Data_block_1

	Name	Datentyp	Startwert	Kommentar
1	Static			
2	DNS Connection1	TCON_QDN		
3	InterfaceId	HW_ANY	0	not relevant
4	ID	CONN_OUC	16#0	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connection: 16#0B=11=TCP/IP, 16#13=19=UDP
6	ActiveEstablished	Bool	false	active/passive connection establishment
7	RemoteQDN	String[254]	''	fully or partially qualified domain name of remote partner
8	RemotePort	UInt	0	remote UDP / TCP port number
9	LocalPort	UInt	0	local UDP / TCP port number

图 8-2 数据类型 TCON_QDN

3. 在数据类型为 TCON_QDN 的变量中，编程 TCP 连接（如，全限定的域名 (FQDN)）的参数。
4. 在程序编辑器中，创建一个 TCON 指令。
5. 将 TCON 指令的 CONNECT 参数与 TCON_QDN 数据类型的变量进行互连。

   在以下示例中，TCON 指令的 CONNECT 参数已与变量“DNS connection1”（数据类型 TCON_QDN）互连。

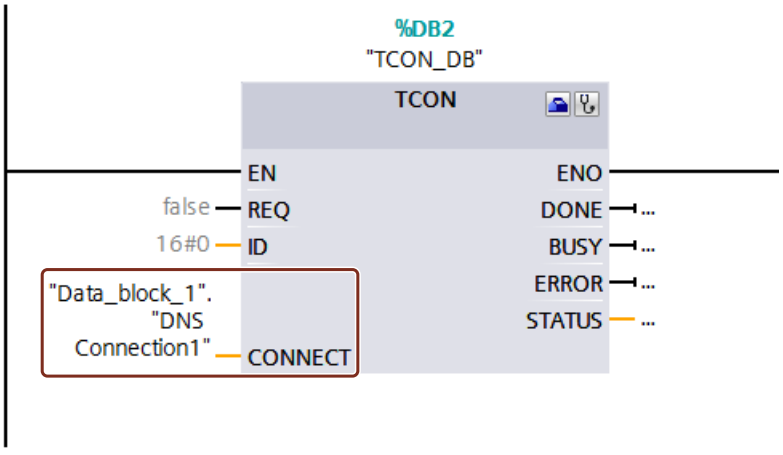


图 8-3 TCON 指令

基于通信伙伴的域名对 UDP 连接进行寻址

对于 S7-1500 CPU 固件版本 V2.0 及以上版本，通过 UDP 发送数据时，可通过全限定域名 (FQDN) 进行寻址。使用参数 ADDR 中的指令 TUSEND 时，可引用 TADDR_SEND_QDN 类型的结构。

接收方可返回 IPv4 地址或 IPv6 地址。使用参数 ADDR 中的指令 TURCV 时，可引用 TADDR_RCV_IP 类型的结构。只有这种结构才能包含两种 IP 地址类型。

说明

网路负载

与 TCP 协议不同，UDP 通信协议不是面向连接的。在块参数 REQ 的每个跳变沿，TUSEND 或 TURCV 命令都会执行一次 DNS 服务器查询。这将导致网络负载或 DNS 服务器上的负载过高。

更多信息

有关系统数据类型 TCON_QDN、TADDR_SEND_QDN 和 TADDR_RCV_IP 的更多信息，请参见 STEP 7 在线帮助。

有关基于通信伙伴的域名建立 TCP 安全连接的信息，请参见"开放式用户安全通信 (页 80)"部分。

8.5 通过 TCP、ISO-on-TCP、UDP 和 ISO 建立开放式用户通信

组态 TSEND_C、TRCV_C 或 TCON 指令的连接

要求：已在程序编辑器中，创建了 TSEND_C、TRCV_C 或 TCON 指令。

1. 在程序编辑器中，选择开放式用户通信的 TCON、TSEND_C 或 TRCV_C 块。
2. 在巡视窗口中，打开"属性 > 组态"(Properties > Configuration) 选项卡。
3. 选择"连接参数"(Connection parameters) 组。在选择连接伙伴之前，只显示伙伴端点的空下拉列表。其它所有输入选项均禁用。

   同时显示一些已知的连接参数：

   – 本地端点的名称
   – 本地端点的接口

– 本地端点的 IPv4 地址

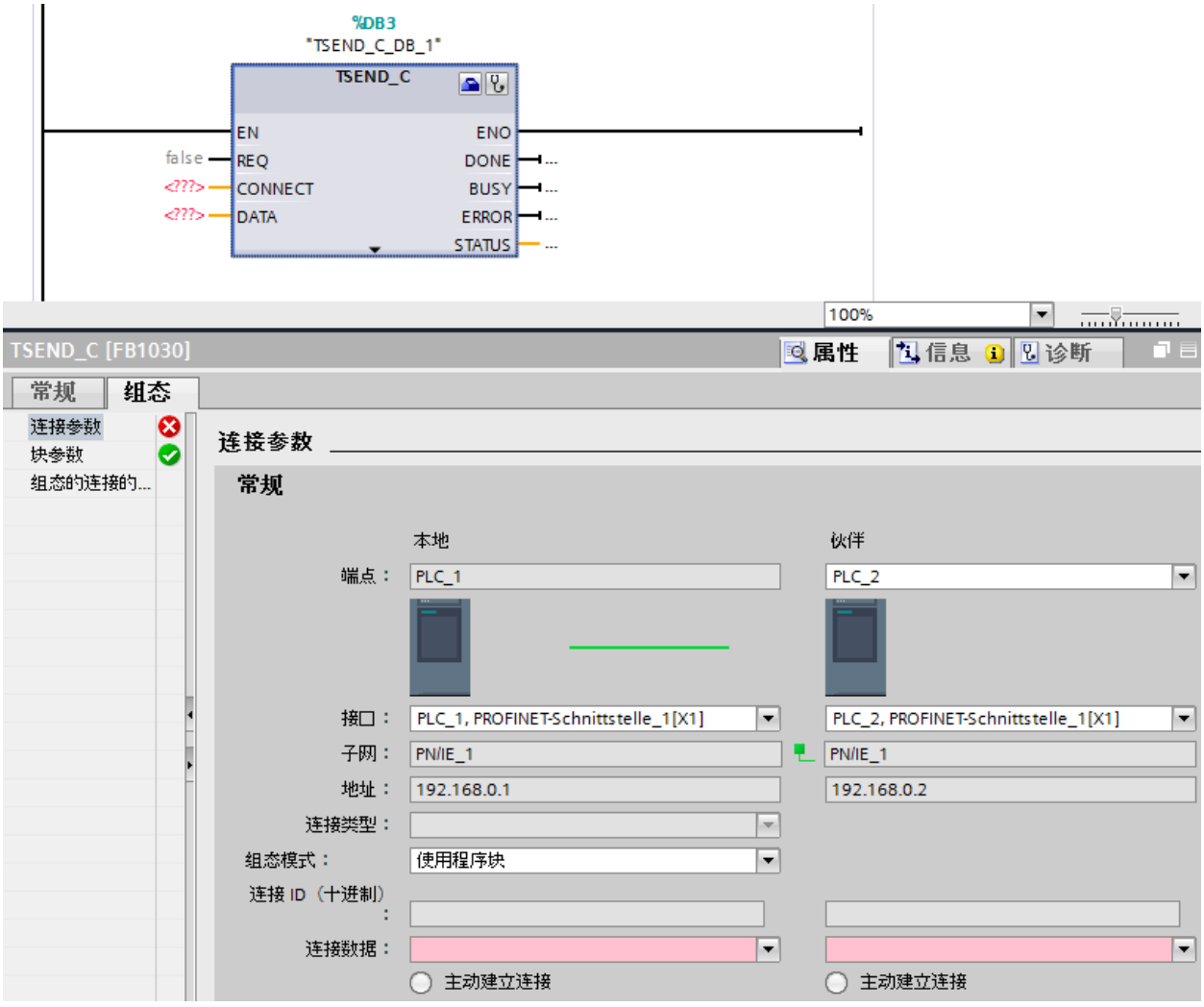


图 8-4 TSEND_C 的连接参数

4. 从伙伴端点的下拉列表框中，选择一个连接伙伴。可以选择项目中未指定的设备或 CPU 作为通信伙伴。之后，系统将自动输入一些特定的连接参数。

   用户需要设置以下参数：

   – 伙伴端点的名称
   – 伙伴端点的接口
   – 伙伴端点的 IPv4 地址

   如果连接伙伴已联网，则显示子网名称。

5. 在"组态类型"(Configuration type) 下拉列表中，选择使用程序块或使用组态的连接。

6. 在“连接数据”(Connection data) 下拉列表中选择现有连接描述 DB，或者对于已组态的连接，在“连接名称”(Connection name) 下选择现有连接。也可以新建连接描述 DB 或已组态的连接。之后，仍可以选择其它连接描述 DB 或已组态的连接，或者更改连接描述 DB 的名称，以创建新的数据块：
   - 也可以在所选 TCON、TSEND_C 或 TRCV_C 指令的 CONNECT 输入参数互连中查看所选的数据块。
   - 如果已使用 TCON、TSEND_C 或 TRCV_C 指令的 CONNECT 参数为连接伙伴指定了连接描述 DB，则可使用此 DB 或创建一个新 DB。
   - 如果编辑下拉列表中所显示的数据块的名称，则会生成一个新数据块用于该连接，新数据块使用更改的名称，但结构和内容不变。
   - 更改的数据块名称在通信伙伴系统中必须唯一。
   - 连接描述 DB 必须具有结构 TCON_Param、TCON_IP_v4 或 TCON_IP_RFC，具体取决于 CPU 类型和连接。
   - 无法为未指定的伙伴选择数据块。

   在选择或创建连接描述 DB 或已组态的连接后确定并输入其它值。

   以下适用于指定的连接伙伴：
   - ISO-on-TCP 连接类型
   - 默认值为 1 的连接 ID
   - 由本地伙伴方创建的主动连接建立
   - TSAP ID
     对于 S7-1200/1500：E0.01.49.53.4F.6F.6E.54.43.50.2D.31

   以下适用于未指定的连接伙伴：
   - TCP 连接类型
   - 伙伴端口 2000

   以下适用于带有指定连接伙伴的已组态连接：
   - TCP 连接类型
   - 默认值为 257 的连接 ID
   - 由本地伙伴方创建的主动连接建立
   - 伙伴端口 2000

   以下适用于带有未指定连接伙伴的已组态连接：
   - TCP 连接类型
   - 本地端口 2000
7. 输入连接伙伴所需的连接 ID。不能为未指定的伙伴分配任何连接 ID。

说明

必须为已知连接伙伴的连接 ID 输入一个唯一值。连接参数的设置不会检查连接 ID 的唯一性。因此，在创建新连接时，不会输入连接 ID 的默认值。

8. 从相关的下拉列表中选择所需的连接类型。根据连接类型设定详细地址信息的默认值。可选择以下通信协议：
   - TCP
   - ISO-on-TCP
   - UDP
   - ISO（仅适用于组态模式"使用已组态的连接"(Use configured connection)）

   可以编辑地址详细信息中的输入框。根据所选的协议，可以编辑端口（TCP 和 UDP）或 TSAP（ISO-on-TCP 和 ISO）。
9. 使用"主动连接建立"(Active connection establishment) 复选框来设置 TCP、ISO 和 ISO-on-TCP 的连接建立特性。用户可以决定主动建立连接的通信伙伴。

连接组态将立即检查更改后的值是否存在输入错误，然后将值输入连接描述数据块中。

说明

只有在将伙伴端点的程序段下载到硬件后，两个通信伙伴之间才能进行开放式用户通信。要实现功能完整的通信，应确保在设备上不仅装载了本地 CPU 的连接描述，而且还装载了伙伴 CPU 的连接描述。

组态 TSEND/TRCV 的连接

如果要在开放式通信中使用 TSEND/TRCV 指令，则需先组态一个连接（如，TCP 连接）。

要组态 TCP 连接，请按以下步骤操作：

1. 在 STEP 7 的"设备与网络"(Devices & networks) 编辑器的网络视图中，组态通信伙伴。
2. 单击"连接"(Connections) 按钮，然后从下拉列表中选择"TCP 连接"(TCP connection) 连接类型。
3. 使用拖放操作，互连通信伙伴（通过接口或本地端点）。如果所需的 S7 子网尚不存在，则系统将自动创建。

   还可以设置与未指定伙伴的连接。
4. 从网络视图中选择已创建的连接。
5. 在"属性"(Properties) 选项卡的"常规" (General) 区域中，设置连接的属性（例如，连接名称和将使用的通信伙伴接口）。

   如果要连接一个未指定的伙伴，则需设置该伙伴的地址。
   本地 ID（用户程序中的连接参考）位于"本地 ID"(Local ID) 区域中。
6. 在项目树中，选择用于 1 个 CPU 的"程序块"(Program blocks) 文件夹。双击文件夹，打开文件夹中的 OB1。将打开程序编辑器。
7. 从"指令"(Instructions) 任务卡中"通信"(Communication) 区域内的"开放式通信"(Open communication) 中，选择所需的指令（如 TSEND）并拖放到 OB1 中的程序段中。
8. 通过该指令的 ID 参数，指定要用于数据传输的已组态连接的本地 ID。
9. 互连 TSEND 指令的"DATA"参数和数据块中的用户数据。
10. 将硬件配置和用户程序下载到 CPU。

按照以上步骤，通过接收指令 TRCV 建立与伙伴 CPU 的连接，并将下载到该 CPU 上。

使用 CP 1543-1 进行 ISO 连接时的注意事项

使用"ISO 连接"(ISO connection) 连接类型时，如果要通过 MAC 地址进行寻址，则需在 CP 的属性中选中复选框"使用 ISO 协议"(Use ISO protocol)。

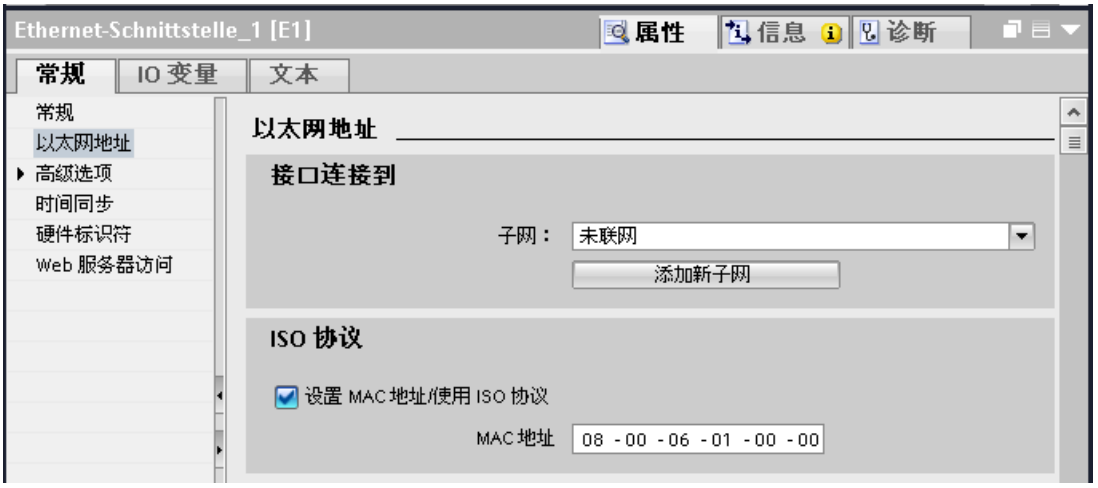


图 8-5 选择 CP 1543-1 ISO 协议

更多信息

STEP 7 在线帮助介绍了：

- 开放式通信的指令
- 连接参数

该常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/109479564)介绍了指令 TSEND_C 和 TRCV_C 在 S7-1500 中的行为。

8.6 建立 FDL 通信

要求

- 组态软件：STEP 7 Professional V14
- 连接的端点：CPU S7-1500 固件版本 V2.0 或更高版本，带有通信模块 CM 1542-5 固件版本 V2.0

建立组态的 FDL 连接

要在 STEP 7 中建立组态的 FDL 连接，请按以下步骤操作：

1. 在程序编辑器中，创建一个 TSEND_C 指令。
2. 在巡视窗口中，选择该 TSEND_C 指令并转至"属性 > 常规 > 连接参数"(Properties > General > Connection parameters)。
3. 在端点下，选择伙伴端点。选择以下两个伙伴端点中的一个：
   - CPU S7-1500，带有 CM 1542-5
   - 未指定
4. 在"组态类型"(Configuration type) 中，选择"使用组态的连接"(Use configured connection)。
5. 在"连接类型"(Connection type) 中，选择"FDL"。
6. 在"接口"(Interface) 中，选择以下接口：
   - "本地"(Local)：CM 1542-5 的 PROFIBUS 接口
   - "指定的伙伴"(Specified partner)：CM 1542-5 的 PROFIBUS 接口
7. 在"连接数据"(Connection data) 中，选择"设置<新>"(setting <new>)。

下图显示了 STEP 7 中 FDL 连接的完整组态。

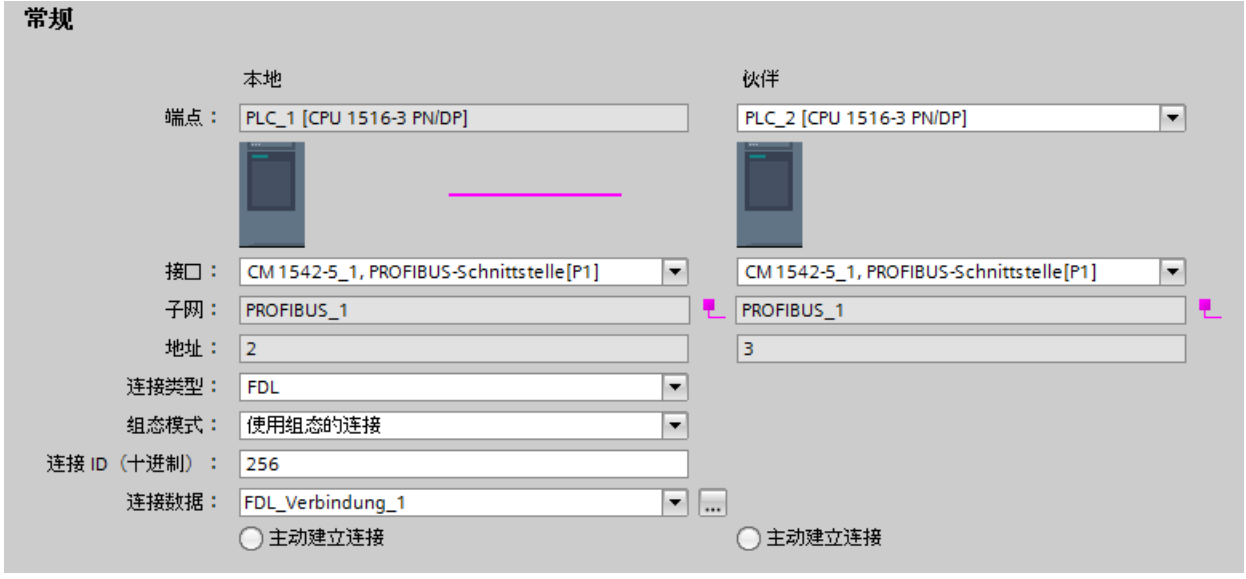


图 8-6 组态 FDL 连接

在用户程序中建立 FDL 连接

要通过 FDL 进行数据通信，在任何情况下都需要手动创建 TCON_FDL 系统数据类型的数据块、分配相应参数，并在指令中直接调用该数据块。请按以下步骤操作：

1. 在项目树中，创建一个全局数据块。
2. 在该全局数据块中，定义一个 TCON_FDL 数据类型的变量。

   在以下示例中，显示了一个全局数据块“FDL_connection”。其中，变量“FDL_connection”的数据类型为 TCON_FDL。

FDL_connection

	名称	数据类型	启动值	...	...	...	...	...	监控	注释
1	Static									
2	FDL_connection	TCON_FDL			☑	☑	☑			
3	InterfaceId	HW_ANY	0		☑	☑	☑			HW identifier of PB interface submodule
4	ID	CONN_OUC	16#0		☑	☑	☑			connection reference / identifier
5	ConnectionType	Byte	16#15		☑	☑	☑			type of connection: 21= FDL connection
6	ActiveEstablished	Bool	false		☑	☑	☑			active/passive connection establishment
7	ServiceId	Byte	16#0		☑	☑	☑			service id: 0 – default, 1 – SDA, 2 – SDN
8	RemotePBAddress	Byte	16#0		☑	☑	☑			remote ProfiBus partner address
9	LocalPBAddress	Byte	16#0		☑	☑	☑			local ProfiBus partner address
10	RemoteLSAP	Byte	16#0		☑	☑	☑			remote PB link-layer service access point
11	LocalLSAP	Byte	16#0		☑	☑	☑			local PB link-layer service access point

图 8-7 编程 FDL 连接

3. 在数据类型为 TCON_FDL 的变量中，编程 FDL 连接的参数（如，PROFIBUS 地址）。
4. 在程序编辑器中，创建一个 TCON 指令。
5. 将 TCON 指令的 CONNECT 参数与 TCON_FDL 数据类型的变量进行互连。

   在以下示例中，TCON 指令的 CONNECT 参数已互连到变量“FDL_Connection”（数据类型 TCON_FDL）。

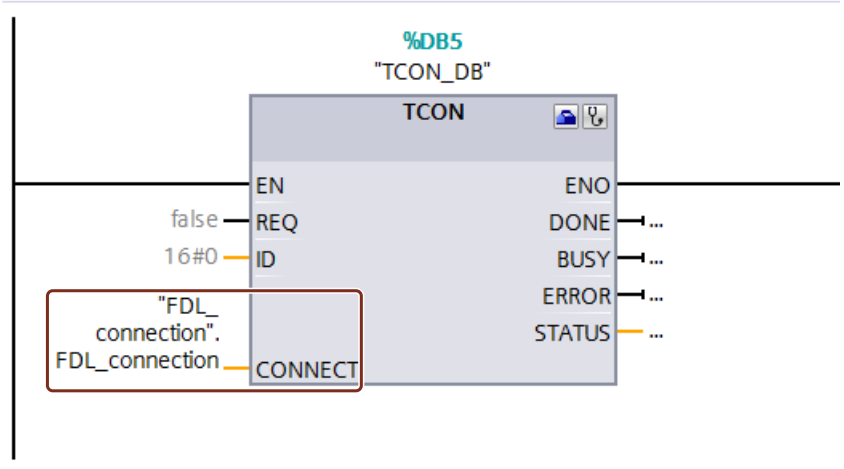


图 8-8 示例：FDL 连接的 TCON 指令

8.7 建立与 Modbus TCP 的通信

通过用户程序建立 Modbus TCP 连接

使用指令 MB_CLIENT 或 MB_SERVER，可以在程序编辑器中分配参数。

通过 Modbus TCP 建立通信的操作步骤

MB_CLIENT 指令作为 Modbus TCP 客户端通过 TCP 连接进行通信。通过该指令，可以在客户端和服务器之间建立连接、向服务器发送 Modbus 请求并接收相应的 Modbus 响应。通过该指令，还可控制 TCP 连接的设置。

MB_SERVER 指令作为 Modbus TCP 服务器通过 TCP 连接进行通信。该指令将处理 Modbus 客户端的连接请求、接收并处理 Modbus 请求并发送响应。也可用于控制 TCP 连接的设置。

要求：客户端可通过网络中的 IP 通信访问服务器。

1. 在 STEP 7 的“设备与网络”(Devices & Networks) 编辑器的网络视图中，组态带有 CPU 的 S7-1500 自动化系统。
2. 在项目树中，选择“程序块”(Program blocks) 文件夹。双击该文件夹，打开文件夹中的 OB1。将打开程序编辑器。
3. 从“指令”(Instructions) 任务卡中“通信”(Communication) 区域内的“其它”(Other) 中的“MODBUS TCP”，选择所需的指令（如 MB_CLIENT）并拖放到 OB1 的程序段中。
4. 分配 MB_CLIENT 或 MB_SERVER 指令的参数。请遵守以下规则：

   必须为每个 MB_CLIENT 连接指定 IPv4 服务器地址。

   每个 MB_CLIENT 或 MB_SERVER 连接都必须使用一个数据结构为 TCON_IP_v4、TCON_QDN 或 TCON_Configured 的唯一背景数据块。

   每个连接都需要一个唯一的连接ID。而且该连接 ID 与背景数据块组合成对，对于每个连接而言均唯一。

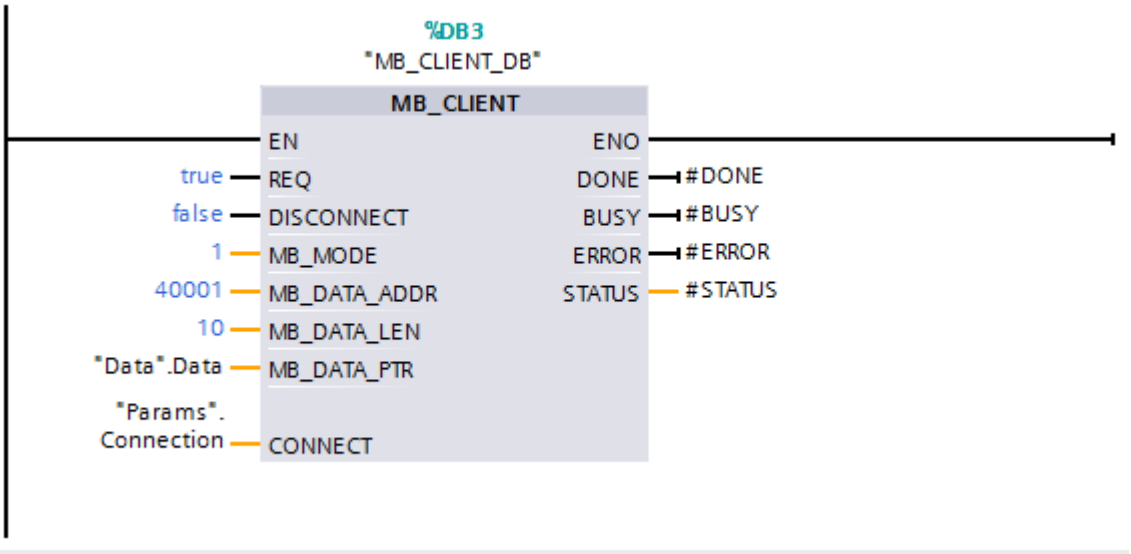


图 8-9 MB_CLIENT

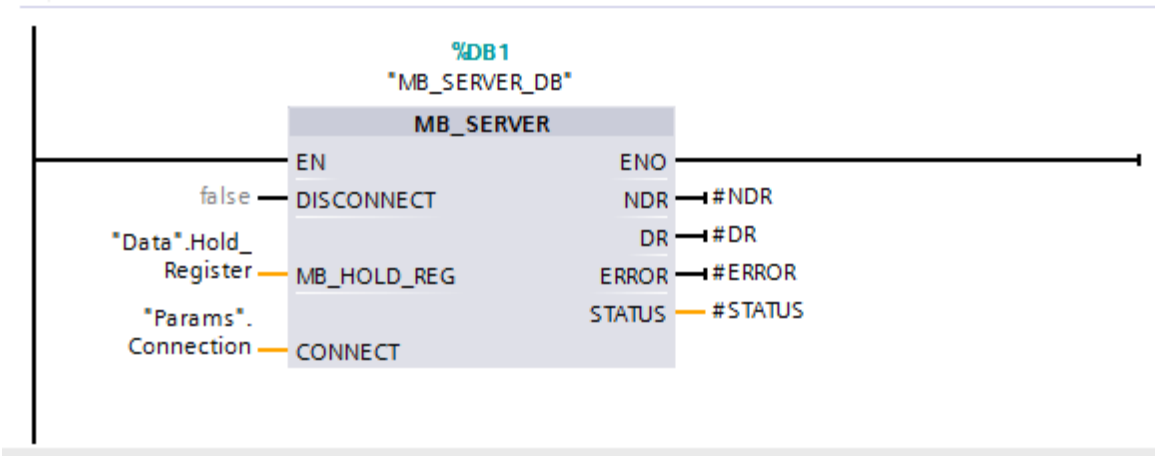


图 8-10 MB_SERVER

5. 将硬件配置和用户程序下载到 CPU。

通过 Modbus TCP 进行冗余通信

使用 MB_RED_CLIENT 或 MB_RED_SERVER 指令基于 Modbus TCP 为冗余通信分配参数：

指令 MB_RED_CLIENT：可使用指令"MB_RED_CLIENT"在客户端和服务器之间建立冗余连接、发送 Modbus 请求、接收响应并控制 Modbus TCP 客户端的连接终止。

指令 MB_RED_SERVER："MB_RED_SERVER"指令将处理 Modbus TCP 客户端的连接请求、接收并处理 Modbus 请求并发送响应。CPU 可以用于：

- 处理多个服务器连接并
- 在同一个服务器端口同时接受多个来自不同的客户端的多个连接。

更多关于 MB_RED_CLIENT 或 MB_RED_SERVER 的信息，请参见 STEP 7 在线帮助。

Modbus TCP 服务器作为连接到 Modbus RTU 的网关

如果将 Modbus TCP 服务器用作连接 Modbus RTU 协议的网关，则使用静态参数 MB_UNIT_ID 对串行网络中的从站设备进行寻址。MB_UNIT_ID 参数与 Modbus RTU 协议中的从站地址字段相对应。在此情况下，MB_UNIT_ID 参数会将请求转发到正确的 Modbus RTU 从站地址。

用户无需编程网关功能。

MB_UNIT_ID 参数位于与 MB_CLIENT 指令相关的背景数据块中。

有关 MB_UNIT_ID 参数的更多信息，请参见 STEP 7 在线帮助。

参考

- 本常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/94766380)介绍了如何对两个 S7-1500 CPU 之间的 Modbus TCP 通信进行编程和组态。
- 本常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/102020340)介绍了如何对 S7-1500 CPU 和 S7-1200 CPU 之间的 Modbus TCP 通信进行编程和组态。

8.8 通过电子邮件建立通信

通过用户程序建立电子邮件连接

在通过电子邮件进行通信时，需要手动创建相关系统数据类型的数据块并分配参数和直接调用指令。在下文中，将介绍具体的操作步骤。

建立通过电子邮件进行通信的步骤

CPU 可以发送电子邮件。通过 TMAIL_C 指令，从 CPU 的用户程序发送电子邮件。

要求：可通过 IPv4 网络访问 SMTP 服务器。

1. 在 STEP 7 的"设备与网络"(Devices & Networks) 编辑器的网络视图中，组态带有 CPU 的 S7-1500 自动化系统。
2. 为 TMAIL_C 指令分配参数。如，在"主题"(Subject) 中输入电子邮件的主题。
3. 在一个全局数据块中，创建类型为 TMAIL_v4、TMAIL_v6（仅 CP 1543-1）或 TMAIL_FQDN（仅 CP 1543-1）的变量。
4. 在该变量的"起始值"(Start value) 列中，设置 TCP 连接的连接参数。
   在"MailServerAddress"中，输入邮件服务器的 IPv4 地址（TMAIL_v4）

   说明
   连接参数接口 ID
   请注意，在指令版本 V5.0 或更高版本的指令 TMAIL_C 中，可为数据类型为 TMAIL_V4_SEC 的接口 ID 输入值"**0**"。此时，CPU 将自行搜索适用的本地 CPU 接口。

   将该变量连接到 TMAIL_C 指令的 MAIL_ADDR_PARAM 参数。
5. 将硬件配置和用户程序下载到 CPU。

更多信息

STEP 7 在线帮助中介绍了：

- 系统数据类型
- 开放式通信的指令
- 连接参数

8.9 通过 FTP 建立通信

通过用户程序建立 FTP 连接

在通过 FTP 进行通信时，必须手动创建相关系统数据类型的数据块，并分配参数和直接调用指令。在下文中，将介绍具体的操作步骤。

FTP 客户端和服务器的功能

CPU 可以将文件发送到 FTP 服务器，也可以从 FTP 服务器接收文件。S7-1500 中只能通过 CP 1543-1 进行 FTP 通信。该 CP 既可以作为 FTP 服务器，也可以作为 FTP 客户端，或者可以同时作为服务器和客户端。FTP 客户端可以是第三方系统/PC。

在 STEP 7 中对 CP 进行相应的组态后，FTP 服务器才能正常运行。

使用 FTP 的客户端功能，可以建立和终止 FTP 连接、传输以及删除服务器上的文件。若要使用 FTP 客户端功能，请使用 FTP_CMD 指令。

设置 FTP 服务器功能的步骤

要求：可通过 IPv4 网络访问 FTP 服务器。

1. 在 STEP 7 的“设备与网络”(Devices & Networks) 编辑器的设备视图中，组态带有 CPU 和 CP 1543-1 的 S7-1500 自动化系统。

   同时，需要在“连接机制”(Connection mechanisms) 部分的“保护”(Protection) 区域导航下 S7-1500 CPU 的硬件配置内选择选项“允许借助 PUT/GET 通信从远程伙伴（PLC、HMI、OPC...）访问”(Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...))。

2. 在“FTP 组态”(FTP configuration) 中的 CP 属性内，进行以下设置：
   - 选择复选框“使用 FTP 服务器传送 S7 CPU 数据”(Use FTP server for S7 CPU data)。
   - 指定要存储 FTP DB 的 CPU、数据块和文件名称。

图 8-11 设置 FTP 组态

3. 将硬件配置下载到 CPU。

设置 FTP 客户端功能的步骤

要求：可通过 IPv4 网络访问 FTP 服务器。

1. 在 STEP 7 的"设备与网络"(Devices & Networks) 编辑器的设备视图中，组态带有 CPU 和 CP 1543-1 的 S7-1500 自动化系统。

   同时，需要在"连接机制"(Connection mechanisms) 部分的"保护"(Protection) 区域导航下 S7-1500 CPU 的硬件配置内选中复选框"允许借助 PUT/GET 通信从远程伙伴（PLC、HMI、OPC...）访问"(Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...))。
2. 在 CPU 的用户程序中调用 FTP_CMD 指令。
3. 在指令 FTP_CMD 中设置 FTP 服务器的连接参数。
4. 创建一个全局数据块，并在此数据块内创建一个 FTP_CONNECT_IPV4、FTP_CONNECT_IPV6 或 FTP_CONNECT_NAME 类型的变量。
5. 将数据块内的变量与 FTP_CMD 指令互连。
6. 要连接 FTP 服务器，需要在 DB 中指定以下参数：
   - 进行 FTP 访问的相关数据类型（FTP_CONNECT_IPV4、FTP_CONNECT_IPV6 或 FTP_CONNECT_NAME）的用户名、密码和 IP 地址
7. 将硬件配置和用户程序下载到 CPU。

应用示例

- 应用示例：与 S7-1500 和 CP 1543-1 进行 FTP 通信
  有关应用示例，敬请访问 Internet (https://support.industry.siemens.com/cs/cn/zh/view/103550797)。
- 应用示例：与 S7-1200/1500 进行的 FTP 客户端通信
  有关的应用示例，敬请访问 Internet (https://support.industry.siemens.com/cs/cn/zh/view/81367009)。

更多信息

STEP 7 在线帮助中介绍了：

- 系统数据类型
- 开放式通信的指令
- 连接参数

8.10 建立和终止通信关系

建立和终止通信

下表显示了如何建立和终止开放式通信中的通信。

表格 8-5 建立和终止通信

设置连接	建立通信	终止通信
通过用户程序	将用户程序下载到 CPU 之后： 通信被动方将通过调用 TSEND_C/TRCV_C 或 TCON 建立本地通信访问。 通信主动方则通过调用 TSEND_C/TRCV_C 或 TCON 开始建立连接。 如果无法建立连接，则将向用户程序中的指令发送一条正反馈。 在终止与 T_RESET 指令的连接后，将会重新建立连接。 如果连接中止，则通信主动方将尝试重新建立连接。 仅当事先与 TCON 成功建立了连接时，才会这样。	• 使用 TSEND_C/TRCV_C、TDISCON 和 T_RESET 指令 • 将 CPU 从 RUN 模块切换到 STOP 模式时 • CPU 断电/通电
通过组态建立连接时	将连接组态和用户程序下载到 CPU 之后。	删除 STEP 7 中的连接组态并将更改后的组态下载到 CPU 中。

S7 通信

9

S7 通信的特点

S7 通信作为 SIMATIC 的同构通信，属于 SIMATIC CPU 之间进行供应商相关的通信（非开放式标准）。在移植和连接现有系统（S7-300、S7-400）进行通信时通常使用 S7 通信。

对于两个 S7-1500 自动化系统之间的数据传输，建议使用开放式通信（请参见"开放式用户通信 (页 119)"部分）。

S7 通信的属性

通过 S7 通信，CPU 可与另一个 CPU 交换数据。一旦用户在接收端接收到数据，就将自动向发送端 CPU 确认已接收到数据。

通过所组态的 S7 连接进行数据交换。S7 连接可以在一端或者同时在两端进行组态。

S7 通信可通过以下方式进行：

- CPU 的集成 PROFINET 或 PROFIBUS DP 接口
- CP/CM 的接口

在一端组态 S7 连接

对于在一端组态的 S7 连接，仅在一个通信伙伴中组态此连接并且仅下载到此伙伴。

可以组态一个连接到一台 CPU 的单向 S7 连接，该 CPU 仅作为 S7 连接的服务器（例如，CPU 315-2 DP）。该 CPU 已组态，因此其地址参数和接口也是已知的。

另外，还可以组态一个连接到伙伴的单向 S7 连接，该伙伴不在项目中，其地址参数和接口都未知。因此，需要输入地址；STEP 7 不对其进行检查。开始时未指定伙伴（创建 S7 连接时未注册伙伴地址）。输入地址后，该地址是"未知"的（即虽然已命名，但项目是未知的）。

这样便可在项目之外使用 S7 连接。本地项目将无法识别该通信伙伴（未指定），将在另一个 STEP 7 或第三方项目中进行组态。

在两端组态 S7 连接

在两端同时组态 S7 连接时，将同时在两个通信伙伴中组态和下载所组态的 S7 连接参数。

S7 通信的指令

与 S7-1500 进行 S7 通信时，可以使用以下指令：

- PUT/GET

  可使用指令“PUT”，将数据写入一个远程 CPU。使用指令“GET”从远程 CPU 读取数据。PUT 和 GET 指令是单向指令，也就是说，只需在一个通信伙伴中有该指令即可。通过连接组态，可方便地设置 PUT 和 GET 指令。

  说明

  PUT/GET 指令的数据块

  使用 PUT/GET 指令时，只能使用进行绝对寻址的数据块。不能使用进行符号寻址的数据块。

  用户还必须在“保护”(Protection) 区域启用此服务以保护 CPU 组态。

  本常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/82212115)提供了有关如何组态和编程 S7 指令以及 GET 和 PUT 通信指令，从而在两个 S7-1500 CPU 之间进行数据交换的信息。

- BSEND/BRCV

  指令“BSEND”可将数据发送到类型为“BRCV”的远程伙伴指令。指令“BRCV”从类型为“BSEND”的远程伙伴指令接收数据。 可通过指令对 BSEND/BRCV 进行 S7 通信，以实现安全数据传输。

- USEND/URCV

  指令“USEND”可将数据发送到类型为“URCV”的远程伙伴指令。指令“URCV”从类型为“USEND”的远程伙伴指令接收数据。无论通信伙伴的处理时间如何，用户都可通过 USEND/URCV 指令对进行 S7 通信，以实现快速的非安全数据传输，例如，传输操作和维护消息。

在从站模式下，通过 PROFIBUS DP 接口进行 S7 通信

在 STEP 7 中的通信模块（如 CM 1542-5）PROFIBUS DP 接口属性中，有一个"测试、调试和路由"(Test, commissioning, routing) 复选框。通过该复选框，可以确定将 DP 从站上的 PROFIBUS DP 作为 PROFIBUS 上的主动设备或被动设备。

- 选中复选框：该从站将作为 PROFIBUS 上的主动设备。
- 禁用复选框：DP 从站将作为 PROFIBUS 上的被动设备。只能为该 DP 从站建立一端组态的 S7 连接。

图 9-1 "测试、调试和路由"(Test, commissioning, routing) 复选框

组态 PUT/GET 指令的 S7 连接

可以创建 S7 连接并在指定 PUT/GET 指令的连接参数时分配这些连接的参数。分配连接参数时，会立即检查更改的值有无输入错误。

要求：PUT 或 GET 指令是在程序编辑器中创建的。

要使用 PUT/GET 组态 S7 连接，请按以下步骤操作：

1. 在程序编辑器中，选择调用 PUT 或 GET 指令。
2. 在巡视窗口中，打开"属性 > 组态"(Properties > Configuration) 选项卡。
3. 选择"连接参数"(Connection parameters) 组。在选择连接伙伴之前，只显示伙伴端点的空下拉列表。其它所有输入选项均禁用。

   同时显示一些已知的连接参数：

   – 本地端点的名称
   – 本地端点的接口

– 本地端点的 IPv4 地址

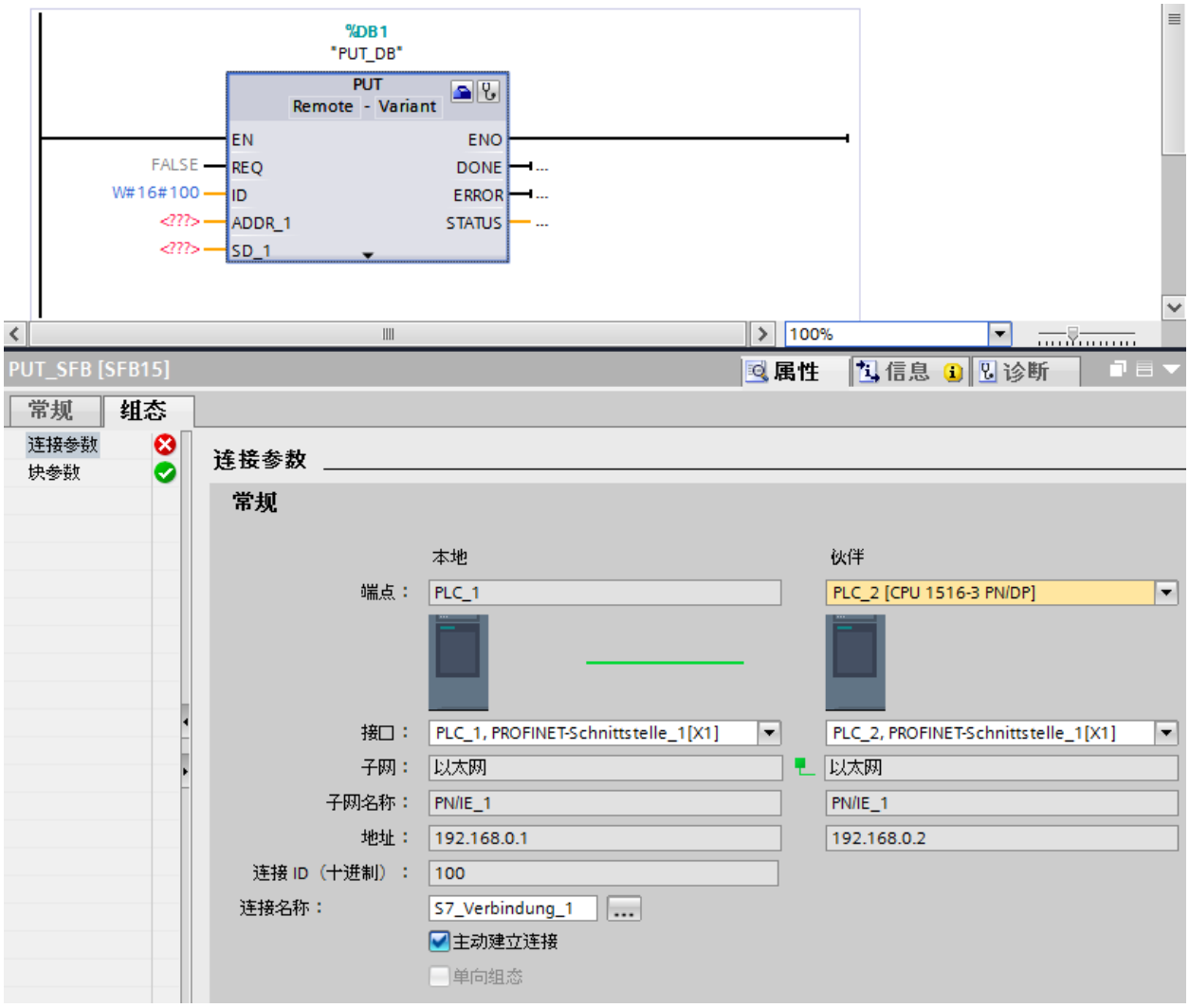


图 9-2 PUT 指令的连接组态

4. 从伙伴端点的下拉列表框中，选择一个连接伙伴。可以选择项目中未指定的设备或 CPU 作为通信伙伴。

   选择连接伙伴后，会自动输入以下参数：

   – 伙伴端点的名称
   – 伙伴端点的接口。如果有多个接口，则可根据需要更改接口。
   – 伙伴端点的接口类型
   – 两个端点的子网名称
   – 伙伴端点的 IPv4 地址
   – 用于通信的连接的名称。

5. 如果需要，请在“连接名称”(Connection name) 输入框中更改连接名称。如果要创建新的连接或编辑现有连接，则可单击连接名称输入框右侧的“选择连接”(Select connection) 按钮。

说明

仅当已将伙伴端点的硬件配置和程序部分加载到硬件中后，两个通信伙伴之间的 PUT 和 GET 指令才能运行。要实现功能完整的通信，应确保在设备上不仅装载了本地 CPU 的连接描述，而且还装载了伙伴 CPU 的连接描述。

组态 BSEND/BRCV 的 S7 连接

例如，如果要使用 BSEND/BRCV 指令进行 S7 通信，首先需要组态 S7 连接。

要组态 S7 连接，请按以下步骤操作：

1. 在 STEP 7 的“设备与网络”(Devices & networks) 编辑器的网络视图中，组态通信伙伴。
2. 选择“连接”(Connections) 按钮，并从下拉列表中选择“S7 连接”(S7 connection) 条目。
3. 使用拖放操作，互连通信伙伴（通过接口或本地端点）。如果所需的 S7 子网尚不存在，则系统将自动创建。

   还可以设置与未指定伙伴的连接。
4. 在选项卡“连接”(Connections) 中，选择 S7 连接所在的行。
5. 在“属性”(Properties) 选项卡的“常规” (General) 区域中，设置 S7 连接的属性（例如，连接名称和将使用的通信伙伴接口）。

   若要建立与未指定的伙伴间的 S7 连接，请设置该伙伴的地址。
   可在“本地 ID”(Local ID) 区域中找到本地 ID（用户程序中的 S7 连接参考）。
6. 在项目树中，选择用于 1 个 CPU 的“程序块”(Program blocks) 文件夹。双击文件夹，打开文件夹中的 OB1。将打开程序编辑器。
7. 在程序编辑器中，如果在一端组态 S7 连接，则在通信伙伴的用户程序中调用相关的指令进行 S7 通信；如果在两端组态，则在通信伙伴的用户程序中调用。例如，从“指令”(Instructions) 任务卡中的“通信” (Communication) 区域内，选择 BSEND 和 BRCV 指令，并将其拖放到 OB1 的一个程序段中。
8. 通过该指令的 ID 参数，指定要用于数据传输的已组态连接的本地 ID。
9. 指定指令的参数，以标识待读取/写入的数据以及数据的来源和目的地。
10. 将硬件配置和用户程序下载到 CPU。

通过 CP 1543-1 进行 S7 通信

如果通过 CP 1543-1 的工业以太网接口建立 S7 通信，则可以在 S7 连接属性的“常规”(General) 区域中选择数据传输的传输协议：

- 选中“TCP/IP”复选框（默认选中）：ISO-on-TCP (RFC 1006)：在 S7-1500 CPU 间进行 S7 通信
- 禁用“TCP/IP”复选框：ISO 协议 (ISO/IEC 8073)：使用 MAC 地址进行寻址

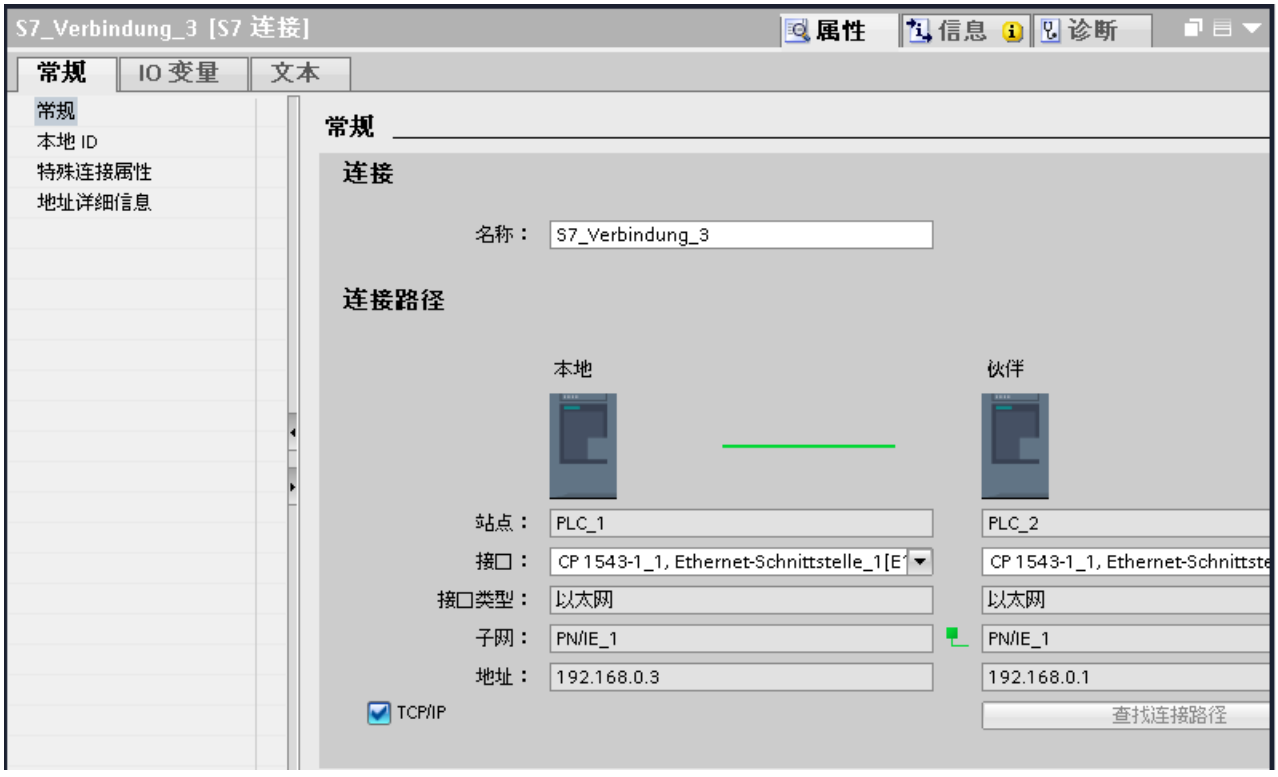


图 9-3 选择 CP 1543-1 的传输协议

建立不同 S7 子网间 S7 连接的步骤

可通过各种 S7 子网（PROFIBUS、PROFINET/工业以太网）建立 S7 连接（S7 (页 376)路由）。

1. 在 STEP 7 的"设备与网络"(Devices & networks) 编辑器的网络视图中，组态通信伙伴。
2. 选择"网络"(Network) 按钮。
3. 通过拖放操作，连接 S7 子网（PROFIBUS、PROFINET/工业以太网）中的相应接口。
4. 选择"连接"(Connections) 按钮，并从下拉列表中选择"S7 连接"(S7 connection) 条目。
5. 在本例中，通过拖放操作将左侧 S7 子网 (PROFIBUS) 中的 PLC_1 连接到右侧 S7 子网 (PROFINET) 中的 PLC_3。

   已组态 CPU 1 和 CPU 3 之间的 S7 连接。

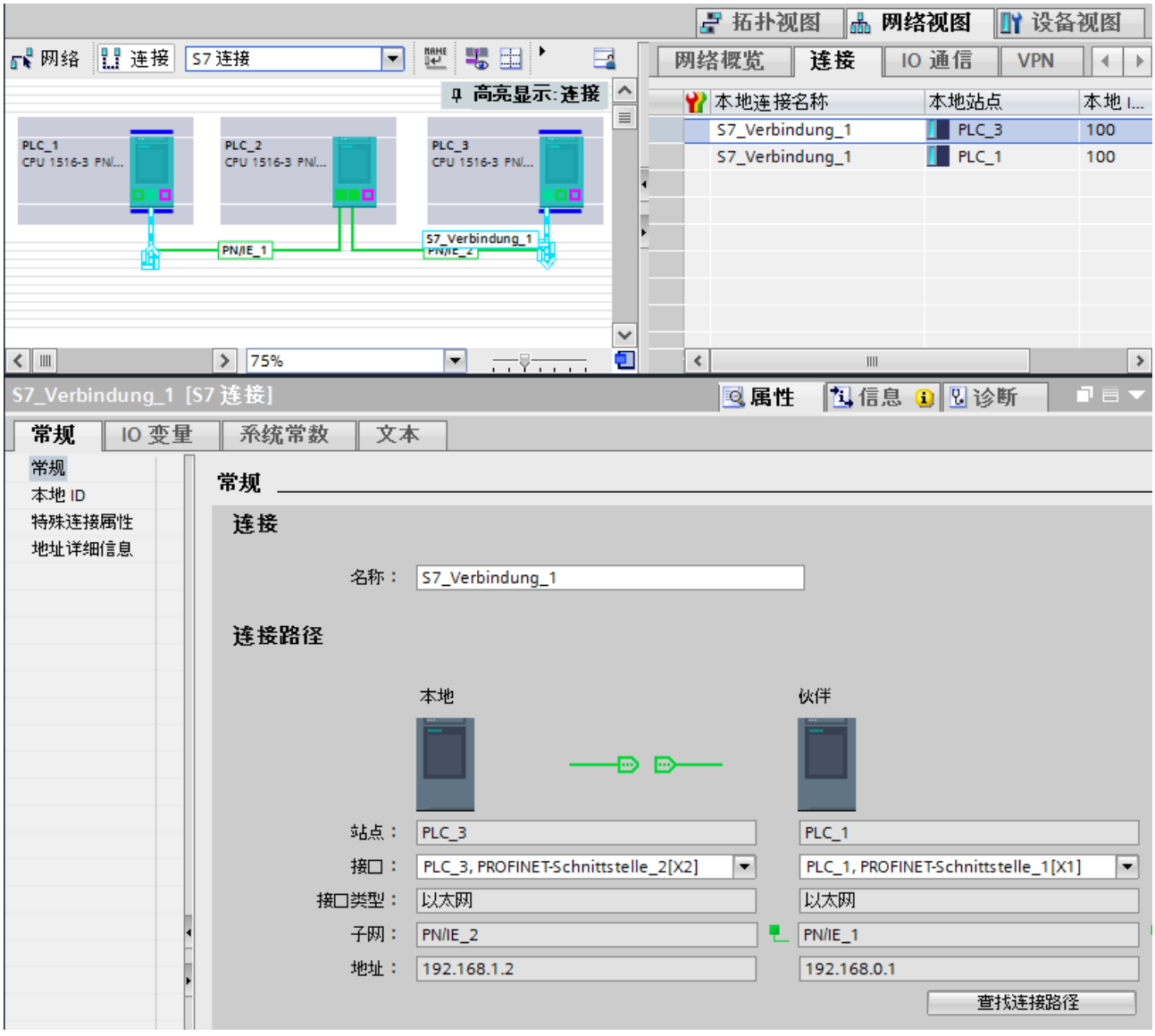


图 9-4 通过不同子网进行 S7 连接

ET 200SP 开放式控制器作为 S7 连接的路由器

如果将 "PROFINET onboard [X2]" 接口分配给 SIMATIC PC 站的 CPU 1515SP PC (F)，CPU 1515SP PC (F) 便可用作 S7 连接的路由器。如果 CP 接口设置为“无，或其它 Windows 设置”(None, or a different Windows setting)，则开放式控制器无法用作路由的 S7 连接的路由器。

如果 CPU 1515SP PC (F) 分配的接口从“SIMATIC PC 站”(SIMATIC PC station) 更改为“无，或其它 Windows 设置”(None, or a different Windows setting)，则 CPU 1515SP PC (F) 路由的现有 S7 连接将失效。由于 PLC 现在不再处理此连接的路由功能，因此在编译 CPU 1515SP PC (F) 时，将不会显示与无效连接相关的消息。在编译连接的端点时，将仅显示路由的无效 S7 连接。

路由的 S7 连接所需的接口必须在 CPU 1515SP PC (F) 上明确指定。可以在“PROFINET 内置 [X2] > 接口分配”(PROFINET onboard [X2] > Interface assignment) 下的属性中编辑 CPU 1515SP PC (F) 的接口分配。

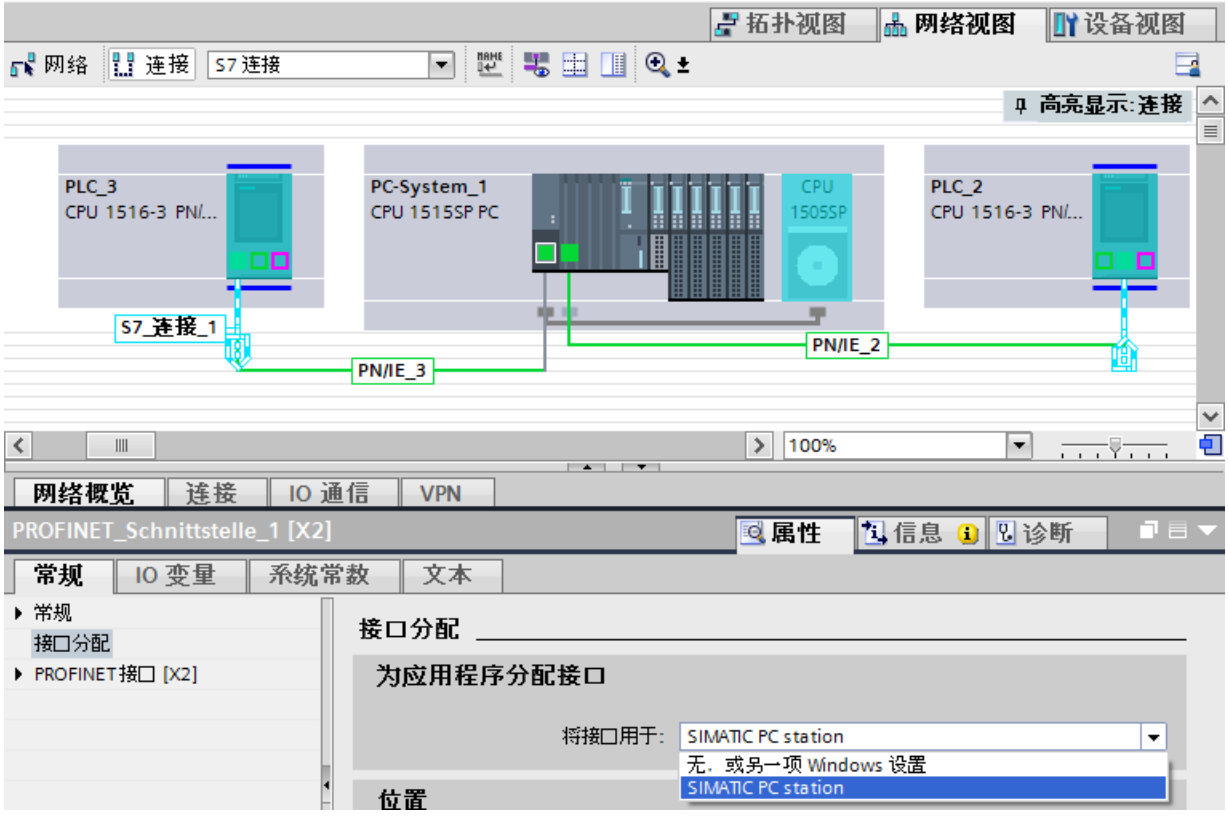


图 9-5 S7 路由 PC 站

更多信息

有关 S7 连接组态和如何在用户程序中使用 S7 通信指令的详细信息，请参见 STEP 7 在线帮助。

点到点连接

10

功能

通过带有串口（RS232、RS422 或 RS485）的通信模块 (CM)，可建立 S7-1500、ET 200MP 和 ET 200SP 的点到点连接。

- S7-1500/ET 200MP：
  - CM PtP RS232 BA
  - CM PtP RS422/485 BA
  - CM PtP RS232 HF
  - CM PtP RS422/485 HF
- ET 200SP：
  - CM PtP

通过点到点连接，通信模块与具有通信能力的第三方系统或设备之间可以进行双向数据交换。进行点到点通信时，需要至少两个通信伙伴。通过 RS422 和 RS485，可以在两个以上通信伙伴间进行通信。

点到点连接的通信协议

- Freeport 协议（也称为 ASCII 协议）
- 3964(R) 程序
- RTU 格式的 Modbus 协议（RTU：远程终端设备）
- USS 协议（通用串行接口协议）

根据 ISO/OSI 参考模型，这些协议将使用不同层：

- Freeport：使用第 1 层（物理层）
- 3964 (R)、USS 和 Modbus:使用第 1 层和第 2 层（物理层和数据链路层。因此，与 Freeport 相比，传输的可靠性更高）。USS 和 Modbus 还另外使用第 4 层。

Freeport 协议的特性

- 通过一个可选择的结束条件（例如，超出字符延时时间、收到结束字符、收到一定数量的数据），接收方可以识别出数据传输是否结束。
- 但发送方无法识别接收方所接收到的发送数据有无错误。

3964 (R) 程序的特性

- 发送数据时，将添加一些控制字符（起始、结束和块校验字符）。请确保这些控制字符不作为数据包含在帧中。
- 可通过这些控制字符建立和终止连接。
- 如果发生传输错误，则将自动重新传输数据。

通过 Freeport 或 3964 (R) 通信进行数据交换

待发送的数据将存储在相应 CPU 数据块的用户程序（发送缓冲区）中。通信模块上的接收缓冲区将用于存储接收数据。检查接收缓冲区的属性，必要时进行调整。必须创建用于在 CPU 中接收的数据块。

在 CPU 的用户程序中，可通过“Send_P2P”和“Receive_P2P”指令在 CPU 和 CM 间进行数据传输。

建立 Freeport 或 3964 (R) 通信的步骤

1. 在 STEP 7 的硬件和网络编辑器的设备视图中，组态一个带有 CPU 和 CM 的 S7-1500 组态。
2. 在 STEP 7 的设备视图中，选择 CM 的接口。
3. 在 STEP 7 巡视窗口的“属性 > 常规”(Properties > General) 下，分配接口的参数（例如连接通信、消息发送组态）。
4. 在“通信 > 通信处理器”(Communication >Communications processor)下的“指令”(Instructions) 任务卡中，选择“Send_P2P”或“Receive_P2P”指令，并将指令拖放到用户程序中（例如拖入 FB）。
5. 根据组态，指定这些指令的参数。
6. 将硬件配置和用户程序下载到 CPU。

或者：通信模块的动态参数分配

在某些类型的应用中，动态建立通信连接更有优势。即，通过一个特定的应用中的程序建立通信。

这种应用的典型示例有串行计算机生产商。为了用户更为便捷地使用用户接口，这些制造商可以按照特定的操作要求对通信服务进行调整。

Freeport 的通信指令

一共有 3 条指令可在用户程序中动态组态 Freeport 通信。以下情况适用于所有 3 条指令：之前有效的组态数据将被覆盖，但不会在目标系统中永久保存。

- “Port_Config”指令可用于对通信模块的相关端口进行程序控制的组态。
- “Send_Config”指令可用于对相关的端口进行动态组态。例如，传输的时间间隔和中断（串行传输参数）。
- “Receive_Config”指令可用于对相关的端口进行动态组态。例如，消息传输的开始和结束条件（串行接收参数）

3964 (R) 通信的指令

一共有 2 种指令可在用户程序中动态组态 3964 (R) 通信。以下情况适用于这些指令：之前有效的组态数据将被覆盖，但不会在目标系统中永久保存。

- “Port_Config”指令可用于对通信模块的相关端口进行程序控制的组态。
- “P3964_Config”指令可用于对协议的参数进行动态组态。

USS 协议的特性

- 一种简单的串行数据传输协议，采用半双工模式并通过循环帧进行数据传输，为驱动技术量身定制。
- 根据主站/从站模式进行数据传输。
  - 主站可以访问驱动器的功能，并可以控制驱动器，读取状态值，对驱动器参数进行读/写操作。

通过 USS 通信进行数据交换

该通信模块将作为主站。主站可以向最多 16 个驱动器连续发送帧（任务帧），并将收到每个所寻址驱动器的响应帧。

在以下条件下，驱动器发送一个响应帧：

- 接收到的帧无错误
- 驱动器在此帧中寻址

如果不满足以上这些条件，或者在广播帧中对驱动器进行寻址，则驱动器将不发送响应帧。

如果主站在一定的处理时间（响应延时时间）之后从驱动器接收到一条响应帧，则主站与相应的驱动器之间存在连接。

建立 USS 通信的步骤

1. 在 STEP 7 的硬件和网络编辑器的设备视图中，组态一个带有 CPU 和 CM 的 S7-1500 组态。
2. 在项目树中，选择“程序块”(Program blocks) 文件夹。双击该文件夹，打开文件夹中的 OB1。将打开程序编辑器。
3. 从“指令”(Instructions) 任务卡的“通信”(Communication) 区域中的“通信处理器”(Communications processor) 文件夹，根据当前的任务选择 USS 通信的指令，并将其拖放到 OB1 的一个程序段中：
   - 使用“USS_Port_Scan”指令，可通过 USS 程序段进行通信。
   - “USS_Drive_Control”指令为驱动器准备发送数据并评估驱动器的响应数据。
   - “USS_Read_Param”指令可用于读出驱动器的参数。
   - “USS_Write_Param”指令可用于更改驱动器的参数。
4. 根据组态，指定这些指令的参数。
5. 将硬件配置和用户程序下载到 CPU。

Modbus 协议 (RTU) 的特性

- 采用串行、异步传输的通信方式，传输速率高达 115.2 kbps，半双工。
- 根据主站/从站模式进行数据传输。
- Modbus 主站可发送向 Modbus 从站进行读写操作的作业：
  - 读取输入、定时器、计数器、输出、存储位、数据块
  - 写入输出、存储位、数据块
- 还可以向所有从站进行广播。

通过 Modbus 通信 (RTU) 进行数据交换

通信模块可以作为 Modbus 主站，也可以作为 Modbus 从站。Modbus 主站可与一个或多个 Modbus 从站进行通信（具体数量取决于物理接口）。只允许 Modbus 主站通过对 Modbus 从站进行显式寻址，向 Modbus 主站返回数据。从站将检测数据传输是否终止，并进行确认。如果发生错误，将向主站发送一个错误代码。

建立 Modbus 通信 (RTU) 的步骤

1. 在 STEP 7 的硬件和网络编辑器的设备视图中，组态一个带有 CPU 和 CM 的 S7-1500 组态。
2. 在项目树中，选择"程序块"(Program blocks) 文件夹。双击该文件夹，打开文件夹中的 OB1。将打开程序编辑器。
3. 从"指令"(Instructions) 任务卡的"通信"(Communication) 区域中的"通信处理器"(Communications processor) 文件夹，根据当前的任务选择 Modbus 通信的指令，并将其拖放到 OB1 的一个程序段中：
   - "Modbus_Comm_Load"指令将对 Modbus 通信的 CM 端口进行组态。
   - "Modbus_Master"指令可用于调用 Modbus 主站的功能。
   - "Modbus_Slave"指令可用于调用 Modbus 从站的功能。
4. 根据组态，指定这些指令的参数。
5. 将硬件配置和用户程序下载到 CPU。

更多信息

- 有关通过点到点连接进行通信的更多详细信息以及串行数据传输的基本知识，请参见功能手册《CM PtP 通信模块 - 点到点连接的组态 (https://support.industry.siemens.com/cs/cn/zh/view/59057093)》。
- 有关如何在用户程序中使用点到点连接指令的说明，请参见 STEP 7 在线帮助。
- 有关带有串行接口的通信模块的信息，请参见特定的通信模块手册。

OPC UA 通信

11

11.1 需了解的 OPC UA 知识

11.1.1 OPC UA 和工业 4.0

信息与数据交换的统一标准

工业 4.0 是指在企业层级对 IT 系统中的大量生产数据进行统一应用、评估和分析。借助工业 4.0，生产与企业层级间的数据交换正在迅速增长。但为确保成功执行，信息与数据交换应采用统一的标准。

标准 OPC 仅支持 Windows 操作系统。为了应对这一限制条件，OPC Foundation 研发出了 OPC UA（OPC 统一架构）标准。

由于 OPC UA 标准独立于特定的操作系统，并采用安全传送机制和数据语义描述，因此尤其适合于跨层级的数据交换。机器数据（受控变量，测量值或参数）也可采用这种方式传输。

这一概念比较重要的一点是允许同时进行 OPC UA 通信和实时通信，从而实现对时间要求严格的机器级数据传送。

OPC UA 具有极高的可扩展性，因此可以在传感器、控制器和 MES 或 ERP 系统之间实现一致的信息交换。

OPC UA 不仅可进行数据传递，而且还可传递与数据有关的信息（数据类型），因此可对该数据进行机器解析访问。

OPC UA 主题页

有关 OPC UA 最重要的文章和链接概览，请访问西门子工业在线支持网站。

OPC UA 主题页 (https://support.industry.siemens.com/cs/cn/zh/view/109770435)

11.1.2 OPC UA 的常规特性

OPC UA 和 PROFINET

可以同时使用 OPC UA 和 PROFINET。这两种协议使用相同的网络基础设施。

独立于操作系统报警

OPC UA 标准并不特定于某个平台，并且针对高性能应用使用优化的基于 TCP 的二进制协议。

OPC UA 支持诸如 Window、Linux、Apple OS X、实时操作系统或移动操作系统（Android 或 iOS）。

独立于特定的传输层

OPC UA 目前支持以下传输机制和协议：

- 通过 TCP/IP，将消息作为二进制流直接传输
- 通过 TCP/IP 和 HTTP 采用 XML 形式传送消息。由于这种传输机制仅支持慢速传输，因此极少使用。S7-1500 CPU 不支持该传输机制。

所有 OPC UA 应用均支持二进制数据交换（基于 OPC UA 技术规范）。

简单的客户端/服务器机制

OPC UA 服务器可在网络中提供大量信息，如有关 CPU、OPC UA 服务器、数据和数据类型的信息。OPC UA 客户端访问这些信息。

支持多种编程语言

OPC 基金会已推出了不同编程语言版本的 OPC UA 标准：虽然已停止对 ANSI C 和 Java 的堆栈进行维护，但仍可以使用 .NET、ANSI C 和 Java 的堆栈。

OPC 基金会提供了 .NET 协议栈，并以开源软件的形式提供了示例程序。请参见"Github (https://github.com/opcfoundation)"。

许多公司提供 Software Development Kits (SDK)。这类开发软件包内含有 OPC Foundation 的协议栈以及其它有助于简化解决方案开发过程的功能。

使用 SDK 的优点：

- 供应商支持
- 经测试的软件
- 详细的文档
- 明确的许可证条件（对于销售解决方案很重要）

易于扩展

OPC UA 可用于不同性能等级的设备：

- 传感器
- 嵌入式系统
- 控制器
- PC 系统
- 智能手机
- 运行 MES 或 ERP 应用程序的服务器。

设备的性能等级因配置文件而异。利用不同的 OPC UA 配置文件，可以针对超小型简单设备以及极高性能的设备调整 OPC UA。

OPC UA 行规描述的是服务器和客户端必须支持的功能和服务。此外，可以选择提供行规中未要求的其它功能/服务。

OPC UA 配置文件与 PROFINET 配置文件不同；后者从供应商中立的软件接口意义上为安装和系统定义附加的跨供应商属性和行为。

Nano Embedded Device 2017 Server Profile

对于功能极为有限的超小型设备，可以采用 OPC 基金会的"Nano Embedded Device 2017 Server Profile"。其作用相当于核心服务器，并定义了 OPC UA TCP 二进制协议作为所需的传输行规。通过该行规无需 UA 安全性即可建立连接，但不支持订阅或方法调用。该配置文件可根据需要支持诊断对象和变量。

其它行规基于"Nano Embedded Device 2017 Server Profile"进行创建，需要使用更多资源，可提供更多功能。

Micro Embedded Device 2017 Server Profile

此行规提供的功能有限；且需要至少两个并行连接。此外，该文件支持订阅/数据监视功能，但不支持 UA 安全性和方法调用。

- S7-1200 基本控制器支持"Micro Embedded Device 2017 Server Profile"。S7-1200 还支持 UA 安全性。

Embedded 2017 UA Server Profile

该配置文件专为搭载 50 MB 以上 RAM 和更高性能处理器的设备而开发。它基于 Micro Embedded Device Server 配置文件。此外，它还需要 UA 安全性和方法调用。

此外，服务器必须使其使用的类型模型（数据类型、引用类型、变量类型等）可用。

- S7-1500 高级控制器支持"Embedded 2017 UA Server Profile"。

标准和全局发现配置文件

"OPC UA Specification Part 7"定义附加配置文件：

- "Standard 2017 UA Server Profile"，适用于基于 PC 的 OPC UA 服务器
- 2个全局配置文件，"Global Discovery Server 2017 Profile"和"Global Discovery and Certificate Management 2017 Server Profile"，涵盖了全局发现服务器所需的服务和信息模型

类型-实例概念

OPC UA 为命名空间提供了一个完全互连的（全网状网络）面向对象的信息模型，包括对象描述的元数据。可以通过相互之间引用实例及其类型来生成任何对象结构。由于服务器会公开其实例和类型系统，因此客户端可以浏览此网络并获取所需的全部信息。无论是实例还是类型定义，都在运行过程中使用。

关于如何处理对类型的引用的过程或概念会随着时间的推移而得到优化。这些优化会体现在 OPC UA 规范的新版本中（例如 V1.03 => V1.04）。

PLC 变量映射

OPC UA 服务器中的信息（如，PLC 变量）可建模为节点，通过引用相互连接。服务器会在地址空间显示语义，也可以通过客户端获取（在导航时）。这样，即可通过 OPC UA 客户端从一个节点浏览另一个节点，查找可读取、监视或写入的内容。

集成信息安全机制

OPC UA 可在不同层级应用息安全机制：

- 仅当 OPC UA 客户端和 OPC UA 服务器均通过 X.509-v3 证书进行注册并接受对方的证书时，服务器与客户端之间才能建立安全连接（应用层的信息安全）。可以使用多种安全策略，包括服务器和客户端之间的非安全连接（安全策略："不安全"）。
- 服务器可以随时向用户请求以下信息，以便进行授权访问（身份验证）：
  - 用户证书（不可在 STEP 7 中组态）
  - 用户名和密码
  – 无用户认证

信息安全机制为可选项且可以组态。

更多信息

有关更多信息，请访问 OPC 基金会 (https://opcfoundation.org)网站。

11.1.3 S7-1200/S7-1500 CPU 的 OPC UA

在 OPC UA 中，一个系统作为服务器运行，并为其它系统（客户端）提供数据和已有信息。

举例来说，OPC UA 客户端可对 OPC UA 服务器上的数据进行读写访问。OPC UA 客户端可调用 OPC UA 服务器中的方法。

可通过客户端在线访问此数据，包括关于性能和诊断的信息。在 OPC UA 术语中，此功能称为"Browsen"。使用"Subscription"功能无需对变量进行定期读取；通过此功能，服务器可通知客户端值的更改情况。

系统可同时为客户端和服务器。

S7-1500 CPU 的 OPC UA 服务器

自固件版本 2.0 起，S7-1500 CPU 配备 OPC UA 服务器。

以下章节将介绍如何组态 S7-1500 CPU 的 OPC UA 服务器才能使数据和方法可用于 OPC UA 客户端，以便客户端可对 CPU 上的 PLC 变量进行读访问和写访问以及可以调用服务器方法。

以下章节还将介绍如何将配套规范集成到 OPC UA 服务器的地址空间中。

S7-1200 CPU 的 OPC UA 服务器

自固件 V4.4 起，S7-1200 CPU 配备 OPC UA 服务器。

OPC UA 服务器组态通常与在 S7-1500 CPU 中的组态一样；功能范围和数量限值受所支持"Micro Embedded Device 2017 Server Profile"的限制。与 S7-1500 CPU 不同的是，"Registered Read"和"Registered Write"功能不可用。

自固件版本 V4.5 起，S7-1200 CPU 支持服务器方法以及结构化数据类型（结构和数组）。

更多信息，请参见 STEP 7 在线帮助。

S7-1500 CPU 的 OPC UA 客户端

自固件版本 V2.6 起，S7-1500 CPU 额外配备 OPC UA 客户端。

以下部分将介绍如何使用标准化指令（PLCopen 函数块）创建用户程序，该程序在 OPC UA 客户端中提供以下功能：

- 从 OPC UA 服务器读取数据
- 向 OPC UA 服务器写入数据
- 调用 OPC UA 服务器的方法

STEP 7 (TIA Portal) 提供客户端接口编辑器并为 OPC UA 连接分配参数，以帮助用户创建用户程序。

指令（“指令 > 通信 > OPC UA”(Instructions > Communication > OPC UA)）的帮助中详细介绍了作为客户端的 S7-1500 CPU 的 OPC UA 指令。

用于测试用途的 OPC UA 客户端

以下说明使用了几种不同的 OPC UA 客户端来说明 OPC UA 客户端的使用情况：

- Unified Automation 的“UaExpert”。可免费使用的功能丰富的客户端：
  下载 UaExpert 的链接 (https://www.unified-automation.com/downloads/opc-ua-clients.html)
- OPC Foundation 的“UA Sample Client”。在 OPC Foundation 注册的用户可免费使用该客户端：
  下载 OPC Foundation 示例客户端的链接 (https://opcfoundation.org)

工业在线支持中的应用示例

西门子工业在线支持提供了免费的应用示例，其中包含用于各种应用的客户端 API。用户可使用此接口的函数创建与其应用相匹配的自有 OPC UA 客户端。为了简化对 API 的处理，我们提供了高级 .NET helper 类。

客户端 API 基于 OPC 基金会的 .NET OPC UA 协议栈。

该应用程序示例说明了如何建立服务器与客户端之间的连接等。其中还介绍了对 PLC 变量的读取和写入。

下载链接：SIMATIC S7-1500 OPC UA 服务器的 OPC UA .NET 客户端
(https://support.industry.siemens.com/cs/cn/zh/view/109737901)

11.1.4 访问 OPC UA 应用程序

下文介绍了通过同一站中的 CP 对包含 OPC UA 应用程序的 S7-1500 CPU（客户端或服务器）进行访问的可能情况。此外，还介绍了将这些访问方式与"IP 转发"功能相结合，以通过 S7-1500 站访问另一个 IP 子网的设备。

可以在巡视窗口的 CPU 属性"高级组态"(Advanced configuration) 区域中找到所有相关设置。

需满足以下要求才能通过 CP 接口访问 CPU 中的 OPC UA 应用程序：

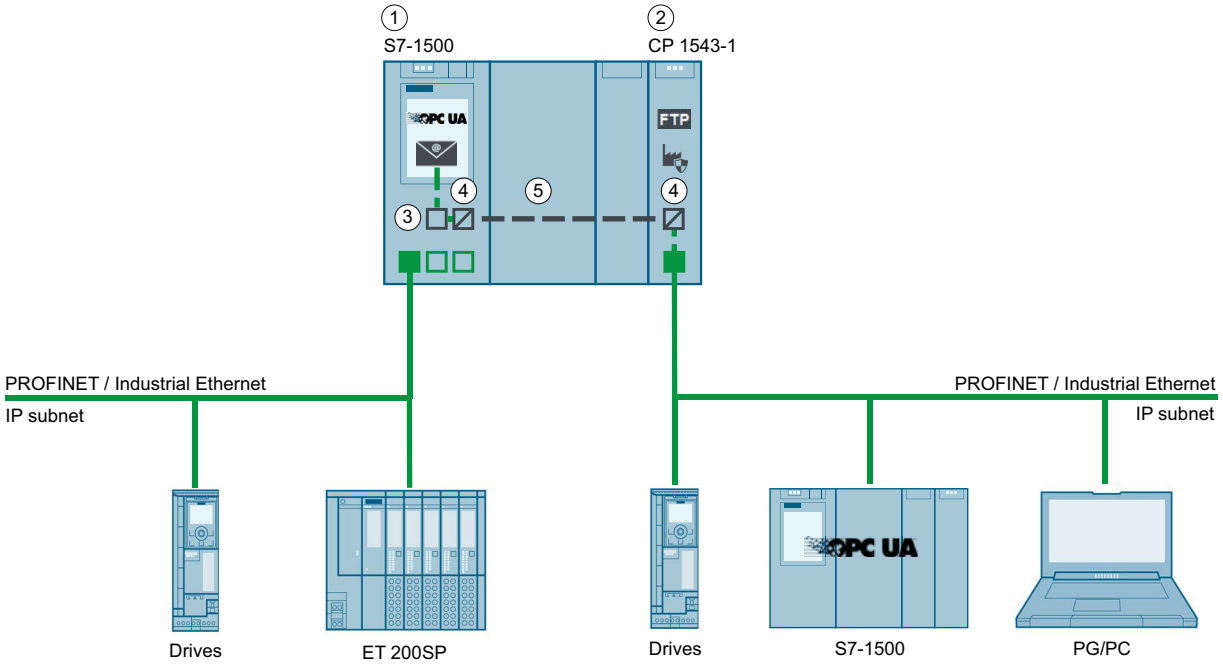
- S7-1500 CPU（高级控制器）固件版本为 V2.8 或更高版本，S7-1500R/H CPU 固件版本为 V3.1 或更高版本
- CP 1543-1 固件版本 V2.2 或更高版本

建议：使用固件版本为 V3.0 或更高版本的 CP 1543-1。自该版本起，还为虚拟接口 (W1) 提供安全功能（防火墙），且不需要在站与非安全网络之间安装额外的防火墙。

原理：用于通过通信模块进行访问的接口

对于 CPU 应用程序（如 OPC UA），必须组态虚拟接口 (W1) 才能通过 CP 接口对其进行访问。之后可以通过此虚拟接口的 IP 地址参数访问基于 IP 的应用程序。

原理图如下所示。



① CPU S7-1500 固件 V2.8 或更高版本（例如 CPU 1515-2 PN）
② CP 1543-1（固件 V2.2 或更高版本）
③ 虚拟接口 (W1)
④ 背板总线上的 PROFINET/工业以太网协议转换，或 PROFINET/工业以太网上的背板总线
⑤ 背板总线

图 11-1 原理：用于通过通信模块进行访问的接口

示例：CPU 中 OPC UA 客户端对 OPC UA 服务器的访问

CPU 中 OPC UA 客户端对 OPC UA 服务器访问时，以下 S7-1500 站接口可用：

- S7-1500 CPU 的本地 PROFINET 接口
- CP 1543-1 的以太网接口（固件版本 V2.2 及更高版本）

下图显示了可能的组态示例：该 CPU 可能具有 OPC UA 客户端角色，CP 子网中的设备可能具有 OPC UA 服务器角色。

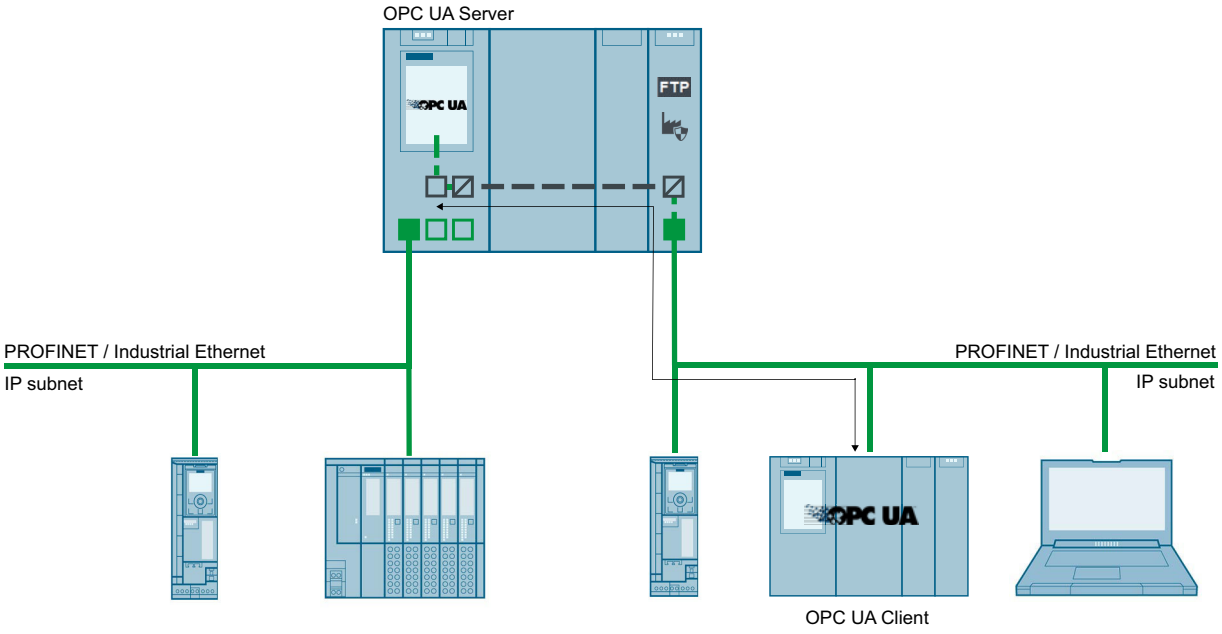


图 11-2 示例：CPU 中 OPC UA 客户端对 OPC UA 服务器的访问

示例：激活 IP 转发功能的 S7-1500 CPU 中 OPC UA 客户端对 OPC UA 服务器的访问

OPC UA 客户端和 OPC UA 服务器也可以通过 S7-1500 CPU 互连，在这种情况下，S7-1500 CPU 用作 IP 转发器。此组态方式可以对现有系统进行灵活扩展。

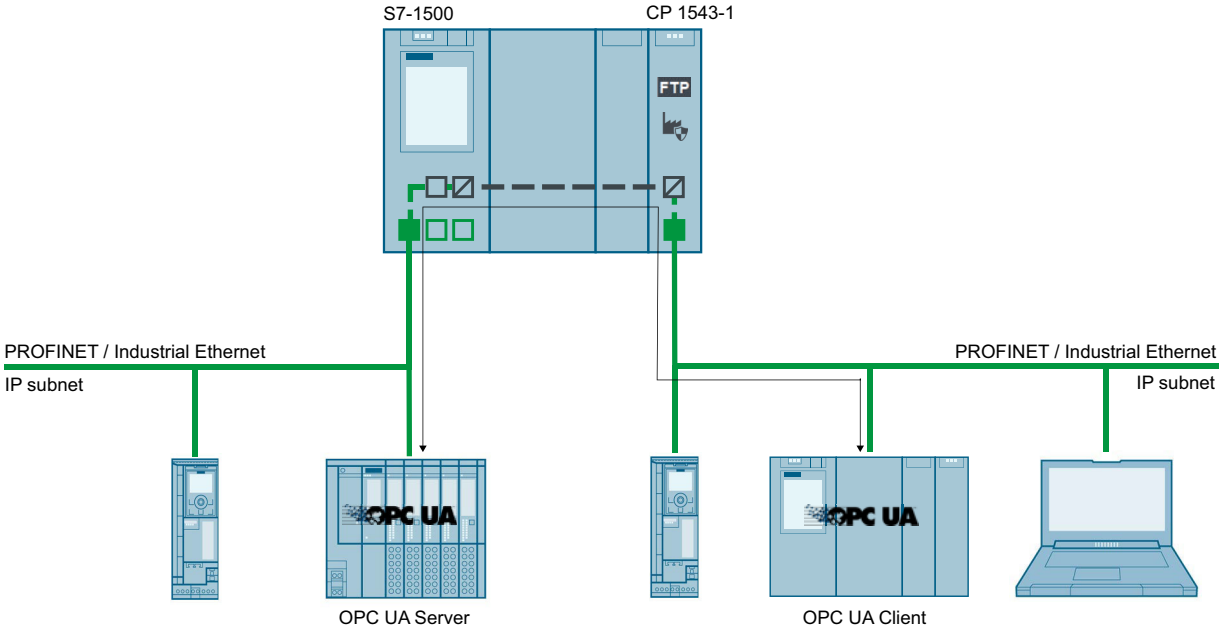


图 11-3 示例：激活 IP 转发功能的 S7-1500 CPU 中 OPC UA 客户端对 OPC UA 服务器的访问

更多信息

有关采用 IP 转发功能时通过虚拟接口的访问方式信息，请参见后续部分：

- IP 转发 (页 380)
- 基于 IP 的应用程序的虚拟接口 (页 388)

11.1.5 节点寻址

节点是 OPC UA 的基本元素，它们相当于面向对象编程中的对象。举例来说，可为用户数据（变量）或其它元数据使用节点。节点用于建立同样包含类型模型和类型定义的 OPC UA 地址空间的模型。

节点 ID (NodeId)

OPC UA 地址空间内的节点由一个 NodeId（节点标识符）进行唯一标识。

NodeId 由一个标识符、标识符类型和一个命名空间索引构成。使用命名空间可避免命名时发生冲突。

OPC 基金会定义了大量节点，用于提供指定 OPC UA 服务器的有关信息。这些节点可以在 OPC Foundation 的命名空间中找到且索引为 0。

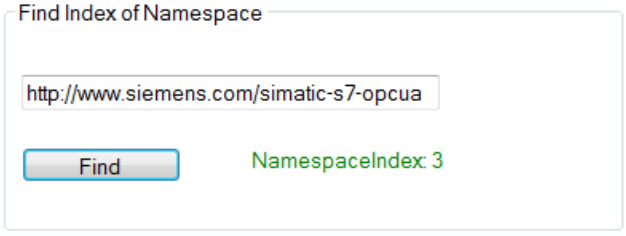
OPC Foundation 还定义有数据类型和变量类型。

命名空间 (Namespace)

除了上述 OPC 基金会命名空间之外，还要关注用于访问 CPU 数据的命名空间：S7-1500 OPC UA 服务器的所有变量或方法都包括在标准服务器接口的命名空间 (Namespace)“http://www.siemens.com/simatic-s7-opcua”中。

系统默认，该命名空间的索引为 3。如果在服务器中插入其它命名空间或删除现有的某个命名空间，则索引将随之更改。因此 OPC UA 客户端需要在读取或写入其数值之前向服务器请求命名空间（例如“http://www.siemens.com/simatic-s7-opcua”）的当前索引。

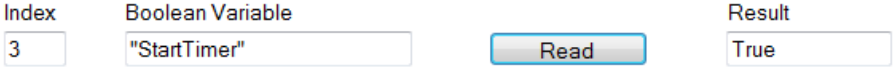
下图举例说明了此类请求的结果。



Identifier

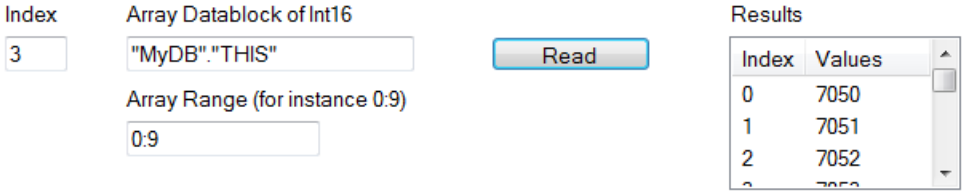
Identifier 对应于引号内的 PLC 变量名称。在 STEP 7 中，引号是唯一不能用作名称的符号。引号可避免发生命名冲突。

在以下示例中介绍了如何读取“StartTimer”变量的值：



Identifier 可包含多个组成部分。各个组成部分之间以句点进行分隔。

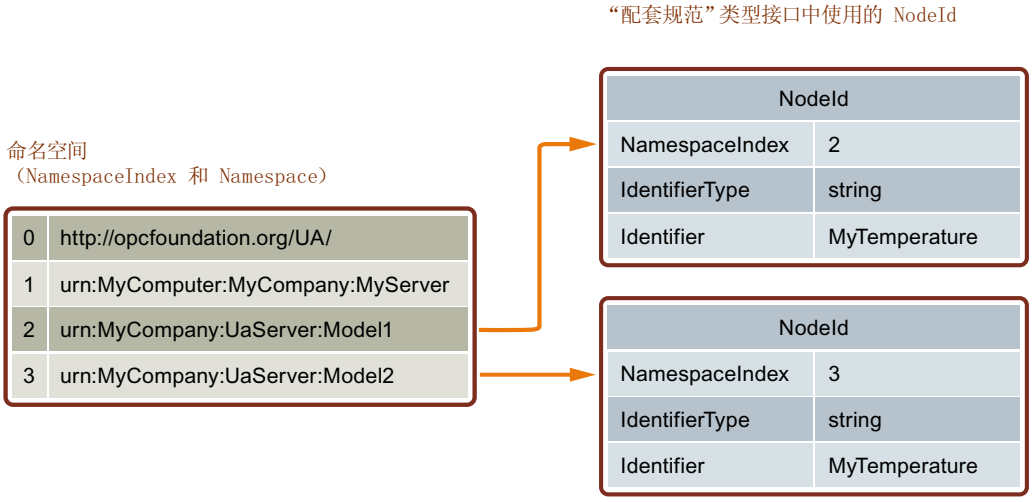
下图举例说明了"MyDB"数组数据块的完整读取。该数据块包含带十个整数值的数组。全部十个值应一次性读取。因此，需在数组范围中输入"0:9"。



NodeId、标识符和命名空间示例

下图说明了 NodeId、标识符和命名空间之间的相互关系：两个节点使用相同标识符但属于不同命名空间时不会出现问题。

STEP 7 (TIA Portal) 可通过服务器接口轻松导入命名空间。



OPC UA 服务器地址空间中的 PLC 变量

下图所示为示例程序的 PLC 变量在 OPC UA 服务器地址空间中所处的位置（摘自 UA 客户端）：

“MyDB”数据块是一个全局数据块。因此，该数据块位于节点“DataBlocksGlobal”下。“StartTimer”是一个内存变量，存储在“Memory”节点下。

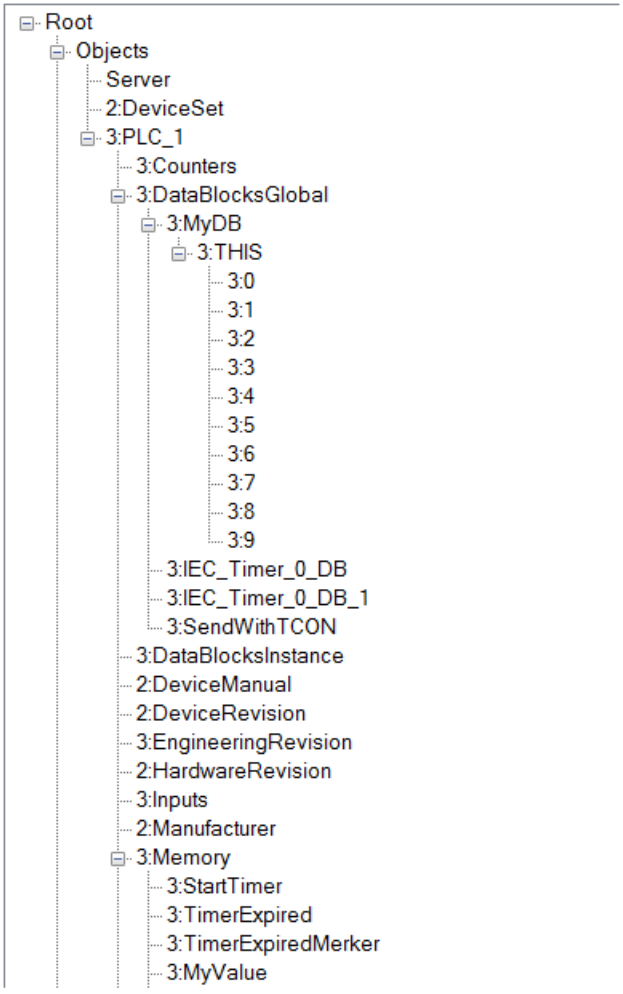


图 11-4 OPC UA 服务器地址空间中的 PLC 变量

OPC UA 服务器地址空间中的方法

如果通过用户程序实现某个方法，则在 OPC UA 服务器的地址空间中将采用以下形式（请参见在 OPC UA 服务器上提供方法 (页 279)）：

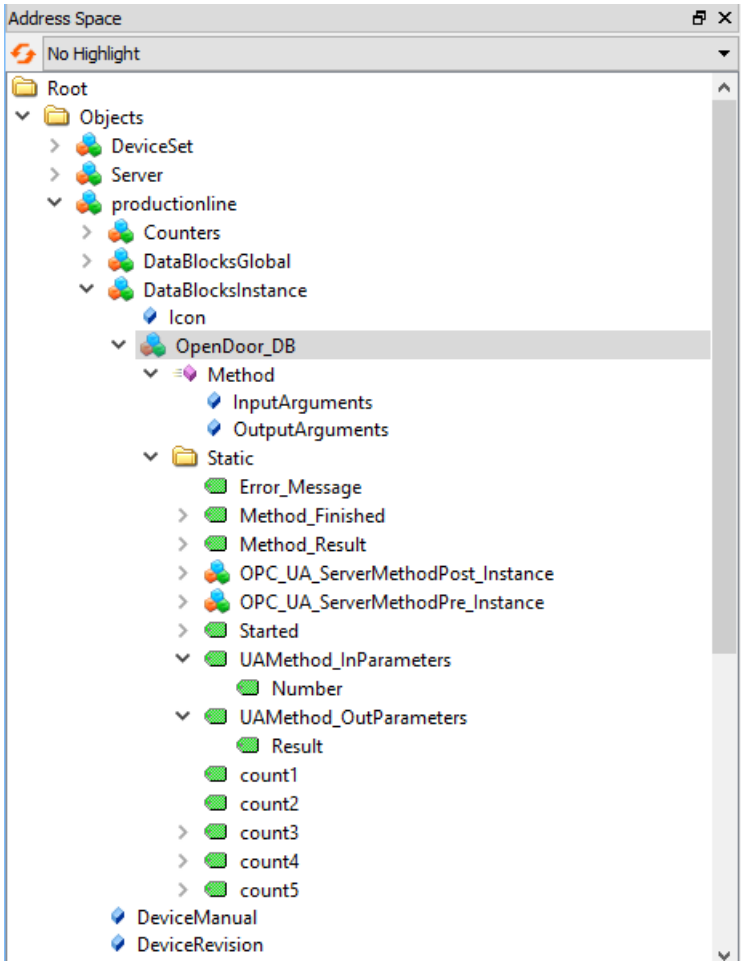


图 11-5 OPC UA 服务器地址空间中的方法

11.1.6 S7-1200/1500 CPU 的 OPC UA 服务器的命名空间概述

正如“节点寻址”主题所述，命名空间索引是节点 ID 的一部分。为了确保节点 ID 在地址空间中始终保持唯一，OPC UA 中使用了名称空间；BrowseName 作为识别节点的唯一方法可能产生歧义。

命名空间由开发 OPC UA 信息模型的不同“命名机构”在 OPC UA 中指定，例如由工作组、OPC 基金会或开发标准信息模型的组织指定。

命名空间通过命名空间 URI 来标识；命名空间 URI 标识命名机构。

命名空间索引用于优化对服务器节点的访问，无需命名空间 URI。命名空间索引是指向服务器管理的命名空间数组的指针。客户端从服务器读取命名空间索引后，就可以直接使用整数（无需 URI 字符串）访问服务器的节点。

下表包含了 S7-1500 和 S7-1200 CPU 的命名空间 URI 和命名空间索引之间的分配。

命名空间 URI 和命名空间索引之间的分配

命名空间索引 0 到 3 在 S7-1200 和 S7-1500 CPU 中具有固定的分配方式。当前未指定其它命名空间索引。

命名空间 URI	命名空间索引	描述
http://opcfoundation.org/UA/	0	OPC UA 规范 (OPC 10000-3) 的节点 ID 和 BrowseName 的命名空间。 示例 0:EngineeringUnits
本地服务器 URI（“urn:” + TIA Portal 中组态的应用程序名称（CPU 属性的“OPC UA > 常规”(OPC UA > General) 区域））	1	本地服务器 (OPC 10000-5) 中定义的节点的命名空间。 示例 urn:SIMATIC.S7-1500.OPC-UA.Application:PLC_1
http://opcfoundation.org/UA/DI/	2	设备 OPC UA 规范 (OPC 10000-100) 的节点 ID 和 BrowseName 的命名空间。 示例 2:DeviceRevision
http://www.siemens.com/simatic-s7-opcua	3	节点 ID 和 BrowseName（专门为 S7-1200/S7-1500 CPU 定义的产品）的命名空间。在此命名空间中，定义了实例（CPU、变量、DB 等）及其类型（UDT、LDT、FB 等）。 示例 3:DataBlocksGlobal
其它实例和类型的附加命名空间的 URI 示例 http://machinesupplier.org/implementedPackML http://opcfoundation.org/UA/PackML	> 3	未指定。例如，可使用 SiOME 定义。 示例 4:MyPackMLMachineInstance 5:PackMLBaseObjectType

11.1.7 需了解的 OPC UA 客户端知识

OPC UA 客户端的基本知识

OPC UA 客户端程序可用于执行以下操作：

- 从 OPC UA 服务器进行信息访问（如 S7-1500 CPU）：读/浏览访问、写访问、订阅
- 通过 OPC UA 服务器执行方法

但是，OPC US 客户端仅可访问为此目的启用的数据（请参见“管理读写权限 (页 213)”）。

要建立与 OPC UA 服务器的连接，需通过服务器的端点（请参见“OPC UA 服务器的端点 (页 202)”）。

从 OPC UA 服务器读取信息

如果存在与服务器端点的连接，则可使用客户端的导航功能：从既定的起始点（“根”节点）开始，浏览服务器的地址空间。

该过程提供了以下信息：

- 启用的 PLC 变量、数据块和数据块元素
- 这些 PLC 变量、数据块和 DB 元素的命名空间索引及标识符
- PLC 变量和 DB 元素的数据类型
- 数组中的元素数量（读取和写入数组时需要）

此外，还可读取有关 OPC UA 服务器自身的信息，以及基于 OPC Foundation 中“OPC UA for Devices”标准的 S7-1500 信息（如，序列号和固件版本）

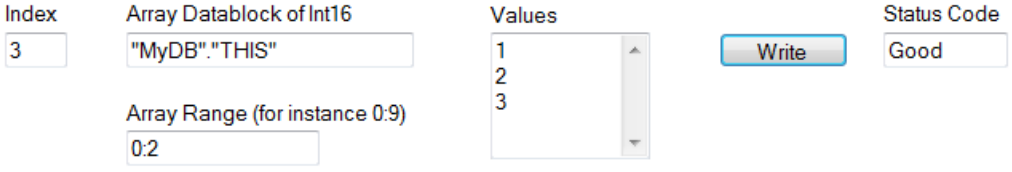
从服务器中读取数据和写入服务器中的数据

现在，您已明确 PLC 变量的命名空间、标识符和数据类型。这表示，用户现在可专门读取各个 PLC 变量和 DB 元素，以及整个数组和结构。

有关读取布尔变量和数组数据块的示例，请参见“寻址节点 (页 163)”部分。

有关访问结构的规则，请单击此处 (页 335)。

基于浏览服务器地址空间时所读取的信息（索引、标识符和数据类型），还可通过 OPC UA 客户端将这些值传输到 S7-1500 中。在以下示例中，介绍了如何覆盖数组数据块“MyDB”中的前三个值。



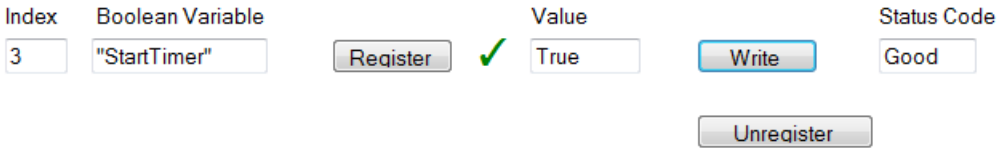
对于“Array Range”，可指定待覆盖的数组元素。状态代码“Good”用于指示数据传输已成功。不过，您只能向 S7-1500 写入值，而不能写入这些值的时间戳。时间戳为只读。

通过注册提高访问速度

Registered Read/Write 有助于对数据进行重复的优化访问 – 具有最高性能。注册变量节点时，OPC UA 服务器会创建一个直接引用所注册节点的数字 Identifier（数字 NodeId）。对于客户端对此数字 Identifier 的读取或写入作业，服务器不必将任何字符串解析为 Identifier，并且可以通过优化的方式访问所请求的变量。

该 Identifier 仅适用于当前会话。会话连接中断/丢失时，需重新查询。

在以下示例中，首先在服务器上注册一个"StartTimer"变量。之后，将使用快速功能"RegisteredWrite"对该值进行设置。



在相同模式中，也可使用函数"RegisteredRead"。在重复读出数据时，该函数优势彰显。但在具体应用中，则建议使用 Subscription 进行代替。

建议：由于注册需要等待一段时间，因此建议在将注册信息保存在 OPC UA 客户端的启动程序中。

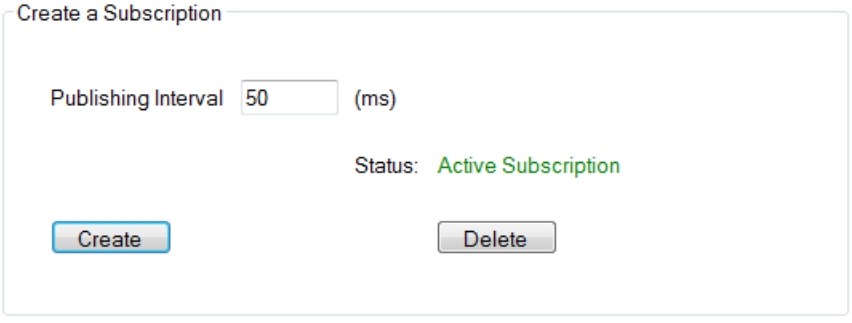
请注意 S7-1500 CPU 属性中可设置的注册节点最大数目，同时客户端需也需符合该数目的要求。具体信息，请参见"OPC UA 服务器的常规设置 (页 227)"。

订阅

术语"Subscription"时一个函数，该函数仅传输 OPC UA 服务器上已注册 OPC UA 客户端中的变量。数值发生变更后，OPC UA 服务器仅向 OPC UA 客户端发送一条有关已注册变量的消息 (monitored Items)。通过对这些变量进行监视，OPC UA 客户端无需再进行固定采样 (Polling)，这有助于降低网络负荷。

要使用该功能，需创建一个 Subscription。为此，需在 UA 客户端中指定"发布间隔"(Publishing Interval)，并单击"创建"(Create) 按钮。发布时间间隔是服务器在通知 (data change notification) 中向客户端发送新值的时间间隔。

在下面的示例中，已创建了一个订阅：客户端将每隔 50 ms 接收一条包含新值的消息（发布间隔为 50 ms）。



防止服务器过载

可通过"最小发布时间间隔"(Minimum publishing interval) 设置 S7-1500 CPU 的 OPC UA 服务器，确保不会提供客户端请求的极短发送时间间隔。请参见"服务器的订阅设置 (页 228)"。

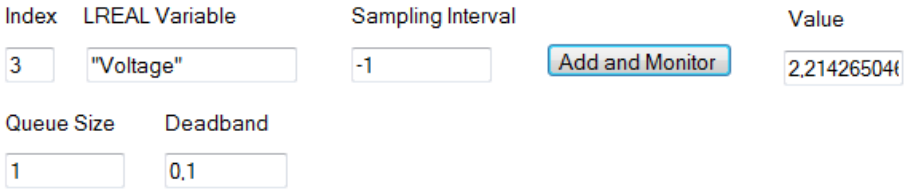
示例：如上所述，客户端想要以 50 ms 的发布时间间隔进行操作。但是，这样短的发布时间间隔会导致网络负荷和服务器负荷较高。因此，应将服务器的“最短发布时间间隔”(Minimum publishing interval) 设置为 1000 ms。并将那些订阅需要较短发布时间间隔的客户端“减速”为 1000 ms，从而防止服务器过载。

订阅范围内的采样和传输 (Sampling & Publishing) 属于通信过程，与其它通信过程（TCP/UDP/Web 服务器通信...）一样，均由 CPU 按优先级 15 进行处理。优先级较高的 OB 会中断通信。如果设置的采样和传输时间间隔过短，该设置会导致通信负荷过高。因此，在满足应用需求的前提下，应尽可能选择较大的时间间隔。

有关变量一致性的信息，请参见“CPU 变量的一致性 (页 218)”。

监视 PLC 变量

Subscription 创建后，系统将通知服务器该功能待监视的变量。在以下示例中，将“Voltage”变量添加到订阅中。



在“Voltage”变量中，包含 S7-1500 CPU 检测的电压值。

采样间隔（“Sampling Interval”）包含一个负值 (-1)。该值决定了将为采样间隔使用 OPC UA 服务器的默认设置。默认设置由订阅的传输间隔（“Publishing Interval”）定义。如果要设置最小的可行采样间隔，请选择数值“0”。

在本示例中，该队列的长度设置为“1”：每隔 50 ms，从 CPU 中仅读取一个值，并在该值发生变更后发送到 OPC UA 客户端。

在本示例中，“死区”(Deadband) 参数设置为“0.1”：值的更改值需要达到 0.1 V；只有这样，发送方才会向客户端发送新值。若值的更改幅度小于该值，则服务器不会发送。例如，可使用该参数禁用信号噪声：没有实际意义的过程变量的轻微变化。

11.2 OPC UA 的信息安全

11.2.1 安全设置

寻址风险

OPC UA 支持过程和生产层级中的不同系统之间以及这些系统与控制与企业层级中的系统之间的数据交换。

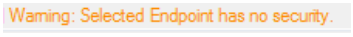
这同样将导致信息安全风险。因此，OPC UA 提供了一系列安全防护机制：

- OPC UA 服务器和客户端的身份验证。
- 检查用户的身份。
- 在 OPC UA 服务器和客户端间，对已签名/加密的数据进行交换。

仅在绝对有必要的情况下，才应绕过这些安全策略：

- 调试过程中
- 在没有外部以太网连接的独立项目中

例如，如果 OPC Foundation 的“UA Sample Client”端点选择了“无”(None)，则程序将发出一条明确的警告消息：



STEP 7 编译项目时，还会检查用户是否考虑保护设置选项，并会警告用户可能存在的风险。还包括采用“不安全”(no security) 设置的 OPC UA 安全策略，该设置对应于端点“无”(None)。

说明

禁用不需要的安全策略

如果在 S7-1500 OPC UA 服务器的安全通道设置中启用了所有安全策略，即采用端点“无”(None)（不安全），则服务器和客户端之间还可能存在非安全数据通信（既未签名也未加密）。S7-1500 CPU 的 OPC UA 服务器还会向设置为“无”(None)（不安全）的客户端发送公用证书。某些客户端会检查该证书。但不会强制客户端向服务器发送证书。客户端的身份可能仍保持未知。无论后续为哪种安全设置，每个 OPC UA 客户端随后都可以连接到服务器。

组态 OPC UA 服务器时，请确保只选择与您的设备或工厂的安全概念兼容的安全策略。应禁用所有其它安全策略。

建议：使用“Basic256Sha256 - 签名和加密”(Basic256Sha256 - Sign and Encrypt) 设置，说明服务器只接受 Sha256 证书。安全策略“Basic128Rsa15”和“Basic256”默认取消激活，不能用作端点。请选择安全策略较高的端点。

附加安全规则

- 仅在特殊情况下，使用端点“无”(None)。
- 仅在特殊情况下，使用“访客身份验证”。
- 如果确实有必要，则仅允许通过 OPC UA 访问 PLC 变量和 DB 元素。
- 在 S7-1500 OPC UA 客户端的设置中使用可信客户端列表，以仅允许对特定客户端进行访问。

11.2.2 ITU X.509 证书

OPC UA 的多个层级中，都集成有安全机制。其中，数字证书至关重要。仅当 OPC UA 服务器接受 OPC UA 客户端的数字证书并将其归类为可信时，客户端才能与服务器建立安全连接。

请参见"处理客户端和服务器证书 (页 232)"部分。

与此同时，客户端还必须检查并信任服务器的证书。服务器和客户端必须显示自己的身份，并证明该身份与声明的相同：即，服务器和客户端必须证明自己的身份。例如，客户端和服务器的相互验证可有效防止中间人攻击。

"中间人"攻击

"中间人"可能会出现在服务器和客户端之间。中间人是一种程序，会截获服务器与客户端之间的通信并将自身伪装为客户端或服务器，以获取 S7 程序的相关信息或设置 CPU 的值，进而对设备或工厂进行攻击。

OPC UA 使用的数字证书符合国际电信联盟 (ITU) 的 X.509 标准，

可识别（认证）一个程序、计算机或机构的身份。

X.509 证书

X.509 证书包含以下信息：

- 证书的版本号
- 证书的序列号
- 证书颁发机构对证书进行签名的算法。
- 证书颁发机构的名称
- 证书有效期的起始和结束时间
- 由证书颁发机构签名证书的程序、个人或机构名称。
- 程序、个人或机构的公钥。

因此，X509 证书将身份（程序、个人或机构的名称）与该程序、个人或机构的公钥关联在一起。

在连接建立期间检查

客户端与服务器建立连接时，设备将基于证书检查全部所需信息以确保其完整性，如签名、有效期、应用程序名称 (URN)，对于固件版本 V2.5，还会检查客户端证书中客户端的 IP 地址。

说明

此外，还会检查证书中存储的有效期。因此必须设置 CPU 时钟，且日期/时间必须在有效期内，否则将无法进行通信。

签名和加密

要检查证书是否篡改，则需对证书进行签名。

可通过以下几种方式进行操作：

- 在 TIA Portal 中，可生成证书并为证书签名。如果您已对项目进行保护，并以具有可进行安全设置的功能权限的用户身份登录，则可以使用全局安全设置。通过全局安全设置可访问证书管理器，由此也可访问 TIA Portal 的证书颁发机构 (CA)。
- 还可通过其它选项创建证书并为证书签名。在 TIA Portal 中，可将证书导入到全局证书管理器中。
  - 联系一家证书颁发机构 (CA) 并对证书进行签名。

    此时，认证颁发机构将核实您的身份，并通过该证书颁发机构的私钥对您的证书进行签名。为此，需向证书颁发机构发送一个 CSR（证书签名请求）。
  - 自行创建证书并对其进行签名。

    例如，为实现上述过程，您应使用 OPC 基金会的"Opc.Ua.CertificateGenerator"程序。还可使用 OpenSSL。
    有关更多信息，请参见"用户自己生成 PKI 密钥对和证书 (页 176)"。

有用信息：证书类型

- 自签名证书

  每个设备都可生成并签署自己的证书。应用示例：通信节点数量有限的静态组态。

  不能从自签名证书派生新的证书。但是，需要将所有自签名证书从伙伴设备加载到 CPU（需要在 STOP 模式下执行）。
- CA 证书：

  所有证书都由证书颁发机构生成和进行签名。应用示例：动态添加设备。

  只需将证书从证书颁发机构下载到 CPU。证书颁发机构可以生成新的证书（添加伙伴设备无需在 CPU STOP 模式下）。

签名

如下所述，通过该签名，可验证消息的完整性和来源。

首先，发送方根据纯文本信息（纯文本消息）生成 HASH 值。之后，再通过私钥对该 HASH 值进行加密，并将该纯文本消息连同加密后的 HASH 值一同发送到接收方。验证签名时，接收方需要一个发送方的公钥（包含在发送方的 X509 证书中）。接收方基于发送方的公钥，对接收到的 HASH 值进行解密。然后，接收方再根据接收到的纯文本消息生成自己的 HASH 值（HASH 过程包含在发送方的证书中）。接收方对这两个 HASH 值进行比较：

- 如果两个 HASH 值相同，则表示从发送方接收到的纯文本消息未经更改并未被篡改。
- 如果两个 HASH 不匹配，则表示到达接收方的的纯文本消息发生了更改。纯文本消息在传送过程中被篡改或受损。

加密

加密数据可防止非经授权的读取。X509 证书不加密；这些证书为公开证书，任何人均可查看。

在加密过程中，发送方将使用接收方的公钥对纯文本消息进行加密。为此，发送方需要接收方的 X509 证书。这是因为，该证书中包含接收方的公钥。接收方使用自己的私钥对消息进行解密。只有接收方才能对该消息进行解密：只有他们才拥有相应的私钥。因此，任何时候私钥都不得泄露。

安全通道

OPC UA 使用客户端与服务器的私钥和公钥建立安全连接，即安全通道。建立安全连接后，客户端和服务器将生成一个只有它们才了解的内部密钥，它们使用此密钥对消息进行签名和加密。较非对称加密过程（私钥和公钥）过程，对称加密过程（共享密钥）的运行速度要快得多。

更多信息

有关通过 TIA Portal 使用证书的应用示例，请参见此处：通过 TIA Portal使用证书 (https://support.industry.siemens.com/cs/ww/zh/view/109769068)。

11.2.3 OPC UA 证书

使用 OPC UA 的 X509 证书

OPC UA 可使用各种类型的 X.509 证书在客户端与服务器之间建立连接：

- OPC UA 应用程序证书

  这类 X.509 证书用于标识软件实例、客户端或服务器软件的安装。在“机构名称”(Organization name) 属性中，可输入该软件使用方的名称。

  说明

  即使安全设置为“无”(None)（不安全），S7-1500 的 OPC UA 服务器也会使用应用程序证书。这可保证与 OPC UA V1.1 及更早版本的兼容性。

- OPC UA 软件证书

  X-509 证书用于标识客户端或服务器软件的特定版本。这些证书中包含有关属性，用于说明通过 OPC 基金会（或认可的测试实验室）认证时的软件版本。在“机构名称”(Organization name) 属性中，可输入该软件的研发或销售方名称。

  说明

  STEP 7 不支持软件证书。

- OPC UA 用户证书

  该 X.509 证书用于标识特定用户，例如从 OPC UA 服务器检索过程数据的用户。如果用户可通过密码自行认证或组态为匿名访问，则无需使用该证书。

  说明

  STEP 7 不支持用户证书。

所述证书属于最底层实体证书：这些证书用于识别个人、机构、公司或软件实例（安装）等信息。

11.2.4 创建自签名证书

使用客户端的证书生成器

很多 OPC UA 客户端应用程序或 SDK 都集成到示例应用程序中，允许用户通过此应用程序为客户端生成证书。

通常可在介绍 OPC UA 客户端应用程序的上下文中找到证书生成的说明。

在线支持的示例客户端
SIMATIC S7-1500 OPC UA 服务器的 OPC UA .NET 客户端 (https://support.industry.siemens.com/cs/ww/zh/view/109737901)会在程序首次启动过程中在 Windows 证书商店中创建客户端应用程序的自签名软件证书。本示例的文档介绍了处理这些证书的步骤。

使用 TIA Portal 的证书生成器

如果使用的 OPC UA 客户端未生成客户端证书，可通过 STEP 7 创建自签名证书。

为此，请执行以下操作步骤：

1. 在 CPU 特性中，双击“保护和安全 > 证书管理器”(Protection & Security > Certificate manager) 下的“<新增>”(<Add new>)，
2. 单击“添加”(Add)。
3. 在“创建新证书”(Create a new certificate) 对话框中，为“使用”(Usage) 选择“OPC UA 客户端”(OPC UA client) 选项。
4. 单击“确定”(OK)。

在“主题备用名称”(Subject Alternative Name) 字段中，STEP 7 将自动输入所生成证书的 URI。在使用 OPC 基金会的 .NET 堆栈生成程序特定的证书时，将调用该字段（如“ApplicationUri”）。在其它证书生成工具中，该基金会的名称可能不同。

更多信息

有关处理客户端证书的更多信息，请参见“S7-1500 CPU 的客户端证书处理 (页 340)”部分。

11.2.5 用户自己生成 PKI 密钥对和证书

只有在使用无法自行创建 PKI 密钥对和客户端证书的 OPC UA 客户端时，才会涉及此部分内容。此时，可通过 OpenSSL 生成一个私钥和一个公钥，生成一个 X.509 证书，并对该证书进行签名。

使用 OpenSSL

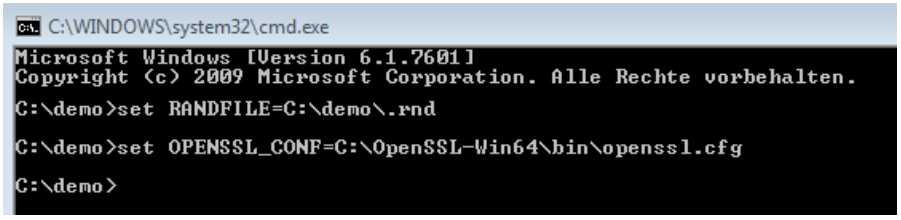
OpenSSL 属于传输层安全工具，可用来创建证书。您还可以使用其它工具，例如 XCA，一款密钥管理软件，该软件具有图形用户界面，改进了已颁发证书的总览功能。

要在 Windows 系统中使用 OpenSSL，请按以下步骤操作：

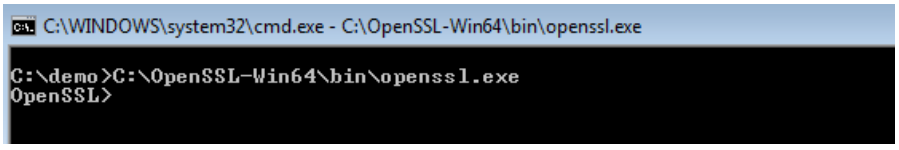
1. 在 OpenSSL 系统中，安装 Windows。如果操作系统为 64 位，则 OpenSSL 将安装在“C:\OpenSSL-Win64”目录中。OpenSSL-Win64 作为开源软件，可从不同的软件提供商处下载。
2. 创建一个目录，如“C:\demo”。
3. 打开命令提示符。为此，单击“Start”，并在搜索栏中输入“cmd”或“command prompt”。右键单击结果列表中的“cmd.exe”，并以管理员身份运行该程序。Windows 将打开命令提示符。
4. 切换到“C:\demo”目录。为此，可输入以下命令：“cd C:\demo”。

5. 设置以下网络变量：
   - set RANDFILE=c:\demo\.rnd
   - set OPENSSL_CONF=C:\OpenSSL-Win64\bin\openssl.cfg

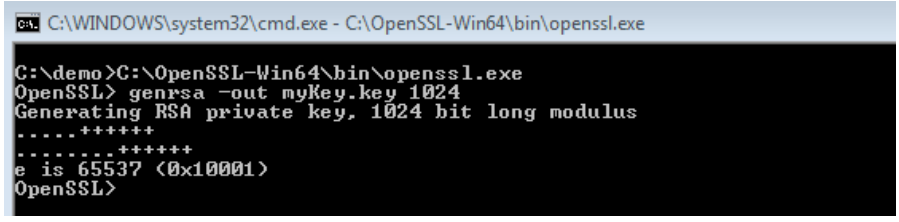
   下图显示了包含以下命令的命令行窗口：



6. 现在，启动 OpenSSL。如果 OpenSSL 已安装在 C:\OpenSSL-Win64 目录中，则可输入：C:\OpenSSL-Win64\bin\openssl.exe。下图显示的命令行窗口中包含以下命令：



7. 生成私钥。将密钥保存到“myKey.key”文件。在本示例中，密钥的长度为 1024 位；为了实现更高的 RSA 安全性，实际长度采用 2048 位。输入以下命令：“genrsa -out myKey.key 2048”（在本示例中为“genrsa -out myKey.key 1024”）。下图显示了包含该命令的命令行以及 OpenSSL 输出结果：



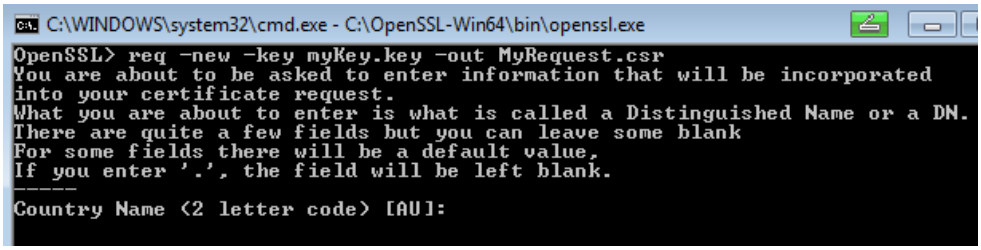
8. 生成一个 CSR (Certificate Signing Request)。为此，可输入以下命令：“req -new -key myKey.key -out myRequest.csr”。在该命令的执行过程中，OpenSSL 将查询有关证书的信息：
   - 国家/地区名称：如，“DE”为德国，“FR”为法国
   - 州或省名称：例如“Bavaria”。
   - 位置名称：如，“Augsburg”
   - 机构名称：输入公司的名称。
   - 机构单位名称：如，“IT”
   - 公共名称：如，“OPC UA client of machine A”
   - 电子邮件地址：

说明

针对固件版本为 V2.5、作为服务器的 S7-1500 CPU 的注意事项

客户端程序的 IP 地址需存储在 S7-1500 CPU 版本 V2.5（仅针对此版本）所创建证书的“主题备用名称”(Subject Alternative Name) 字段中；否则 CPU 将不接受该证书。

输入的信息将添加到证书中。下图显示了包含该命令的命令行以及 OpenSSL 输出结果：



该命令将在包含有 Certificate Signing Request (CSR) 的 C:\demo 目录中创建一个文件；在本示例中，为“myRequest.csr”。

使用 CSR

可通过以下两种方式使用 CSR：

- 将 CSR 发送到证书颁发机构 (CA)：读取特定证书颁发机构的信息。证书颁发机构 (CA) 将检查用户的身份和信息（认证），并使用该证书颁发机构的私钥对该证书进行签名。如，接收已签名的 X.509 证书，并将该证书用于 OPC UA、HTTPS 或 Secure OUC (secure open user communication) 中。通信伙伴将使用该证书颁发机构的公钥检查该证书是否确实由 CA 机构颁发（即，该证书颁发机构已确定您的信息）。
- 用户对 CSR 进行自签名：使用用户的私钥。该选项将在下一个操作步骤中介绍。

自签名证书

输入以下命令，生成一个证书并对自签名（自签名证书）：“x509 -req -days 365 -in myRequest.csr -signkey myKey.key -out myCertificate.crt”。

下图显示了包含以下命令和 OpenSSL 的命令行窗口：

该命令将生成一个 X.509 证书，其中包含通过 CSR 传送的属性信息（在本示例中，为“myRequest.csr”），例如有效期为一年（-days 365）。该命令还将使用私钥对证书进行签名（在本示例中为“myKey.key”）。通信伙伴可使用公钥（包含在证书中）检查您是否拥有属于该公钥的私钥。这样还可以防止公钥被攻击者滥用。

通过自签名证书，用户可确定自己证书中的信息是否正确。此时，无需依靠任何机构即可检查信息是否正确。

更多信息

有关处理 S7-1500 CPU 客户端证书的信息，请参见“S7-1500 CPU 的客户端证书处理 (页 340)”部分。

11.2.6 消息的安全传送

使用 OPC UA 建立安全连接

OPC UA 将在客户端与服务器之间建立安全连接。OPC UA 将检查通信伙伴的身份。OPC UA 使用基于 ITU（国际电信联盟）X.509-V3 标准的证书对客户端和服务器进行认证。例外：使用安全策略“不安全”(No security) 时，将不建立安全连接。

消息的安全模式

OPC UA 使用以下安全策略确保消息安全：

- 不安全

  所有消息均不安全。要使用该安全策略，则需与服务器建立安全策略为“无”(None) 的端点连接。

- 签名

  所有消息均已签名。系统将对所接收消息的完整性进行检查。检测篡改行为。要使用该安全策略，则需与端点安全策略为“签名”(Sign) 的服务器立连接。

- 签名和加密

  对所有消息进行签名并加密。系统将对所接收消息的完整性进行检查。检测篡改行为。而且，攻击者无法读取消息内容（保护机密）。要使用该安全策略，则需与端点安全策略为“签名并加密”(SignAndEncrypt) 的服务器建立连接。

安全策略还可根据所使用的算法命名。示例：“Basic256Sha256 -签名和加密”表示：端点进行安全连接，支持一系列 256 位哈希和 256 位加密算法。

所需层级

下图显示了建立连接时通常所需的三个层：传输层、安全通道和会话。

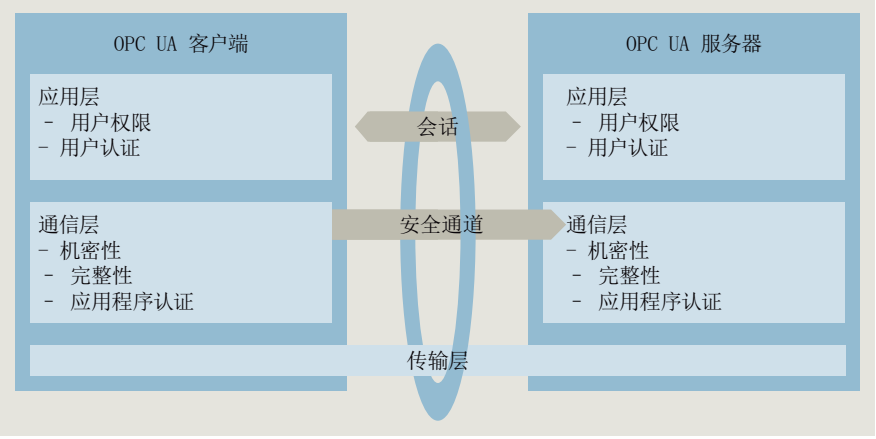


图 11-6 所需层级：传输层、安全通道和会话

- 传输层：

  该层用于发送和接收消息。OPC UA 在此使用优化的基于 TCP 的二进制协议。传输层是后续安全通道的基础。

- 安全通道

  安全层从传输层接收数据，再转发到会话层中。安全通道将待发送的会话数据转发到传输层中。

  在“签名”(Sign) 安全模式中，安全通道将对待发送的数据（消息）进行签名。接收消息时，安全通道将检查签名以检测是否存在篡改的情况。

  采用安全策略“签名并加密”(SignAndEncrypt) 时，安全通道将对待发送数据进行签名并加密。安全通道将对接收到的数据进行解密，并检查签名。

  采用安全策略“不安全”(No security) 时，安全通道将直接传送该消息包而不进行任何更改（消息将以纯文本形式接收和发送）。

- 会话

  会话将安全通道的消息转发给应用程序，或接收应用程序中待发送的消息。此时，应用程序即可使用这些过程值或提供这些值。

建立安全通道

建立安全通道，如下所示：

1. 服务器接收到客户端发送的请求时，开始建立安全通道。该请求将签名或签名并加密，甚至以纯文本形式发送，具体取决于所选服务器端的安全模式。在“签名”(Sign) 和“签名并加密”(Sign & Encrypt) 安全模式中，客户端将随该请求一同发送一个机密数（随机数）。
2. 服务器将验证客户端的证书（包含在请求中，未加密）并检验该客户端的身份。如果服务器信任此客户端证书，
   - 则会对消息进行解密并检查签名（“签名并加密”(Sign & Encrypt)），
   - 仅检查签名（“签名”(Sign)），
   - 或不对消息进行任何更改（“不安全”(No security)）
3. 之后，服务器会向客户端发送一个响应（与请求的安全等级相同）。响应中还包含服务器机密。客户端和服务器根据客户端和服务器的机密数计算对称密钥。此时，安全通道已成功建立。

对称密钥（而非客户端与服务器私钥和公钥）可用于对消息进行签名和加密。

建立会话

执行会话，如下所示：

1. 客户端将 CreateSessionRequest 发送到服务器后，开始建立会话。该消息中包含一个仅能使用一次的随机数 Nonce。服务器必须对该随机数 (Nonce) 进行签名，证明自己为该私钥的所有者。此私钥属于该服务器建立安全通道时所用证书。该消息（及所有后续消息）将基于所选服务器端点的安全策略（所选的安全策略）进行加密。
2. 服务器将发送一个 CreateSession Response 响应。该消息中包含有服务器的公钥和已签名的 Nonce。客户端将检查已签名的 Nonce。
3. 如果服务器通过测试，则客户端将向该服务器发送一个 SessionActivateRequest。该消息中包含用户认证时所需的信息：
   - 用户名和密码，或
   - 用户的 X.509 证书（STEP 7 不支持），或
   - 无数据（如果组态为匿名访问）。
4. 如果用户具有相应的权限，则服务器将返回客户端一条消息 (ActivateSessionResponse)。激活会话。

OPC UA 客户端与服务器已成功建立安全连接。

建立与 PLCopen 函数块的连接。

PLCopen 规范针对 OPC UA 客户端定义了一系列 IEC 61131 函数块。指令 UA_Connect 可根据上述模式启动安全通道和会话。

11.2.7 通过全球发现服务器 (GDS) 实现证书管理

11.2.7.1 通过 GDS 实现自动化证书管理

在 TIA Portal V17 及以上版本和 S7-1500 CPU 固件版本 V2.9 及以上版本中，OPC UA 服务器的证书管理服务可用于在运行期间传送 OPC UA 服务器证书。

通过 GDS 推送管理功能，S7-1500 CPU 的 OPC UA 服务器上的 OPC UA 证书、信任列表和证书吊销列表 (CRL) 可自动进行更新。证书管理自动化意味着，当证书到期后以及对 CPU 执行全新下载操作后，无需再手动重新组态 CPU。此外，使用 GDS 推送管理功能还可以在 CPU 处于 STOP 和 RUN 操作状态时传送更新后的证书和列表。

证书管理信息模型在 OPC UA 第 12 部分（OPC 10000-12：OPC 统一架构，第 12 部分：发现和全球服务）中指定。

自 TIA Portal V18 起以及 S7-1500 CPU 固件版本 V3.0 起，GDS 推送管理功能可用于 Web 服务器证书。通过 GDS 推送管理功能更新证书的顺序理论上与通过 OPC UA 服务器证书功能更新证书的顺序相同。与 OPC UA 服务器证书功能的不同之处是，还可以在运行期间或操作期间将 Web 服务器证书传送到 CPU。下文的相应部分介绍了两者之间的区别或局限性。

以下章节概括介绍了全球发现服务以及 TIA Portal V17/CPU 固件版本 V2.9 及更高版本支持的自动化证书更新功能。

发现服务器

要连接到 OPC UA 服务器，OPC UA 客户端需要其端点的相关信息，如端点 URL 和安全策略。如果网络中提供大量可用服务器，则发现服务器可负责处理对该服务器信息的搜索和管理。

- OPC UA 服务器注册使用发现服务器。
- OPC UA 客户端向发现服务器请求获取可访问的服务器列表，然后连接到所需 OPC UA 服务器。

全球发现服务器 (GDS)

OPC UA GDS 理念一方面可组态跨子网发现服务，另一方面为证书集中管理提供接口。

全球发现服务器 (GDS) 提供的机制可实现对以下组件的集中管理：

- CA 签名证书和自签名证书
- 受信任列表和证书吊销列表 (CRL)

因此，GDS 提供中央证书管理的接入点，并接管 OPC UA 网络中安全服务器的任务。

GDS 主要用于通过相应的 CRL 来管理 CA 签名证书：

- 例如，为 OPC UA 服务器或 Web 服务器初次创建 OPC UA 应用程序证书。
- 定期更新受信任列表和 CRL
- 更新应用程序证书

证书管理

证书管理的任务是自动管理和分发不同服务器或 UA 应用程序的证书和信任列表。

在该上下文中，有以下两种不同的角色：

- 证书管理器 - 提供证书管理功能的 OPC UA 应用
- 证书接收方 - 从证书管理器接收证书、信任列表和 CRL 的 OPC UA 应用程序。

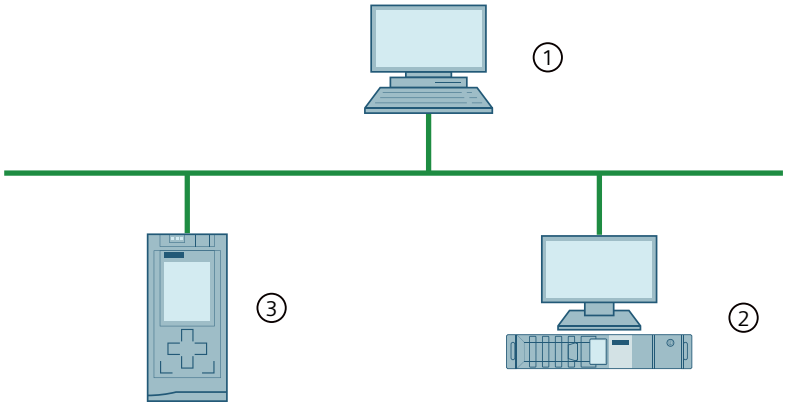
证书管理分为以下两种模式：拉取管理和推送管理。

- 采用拉取管理模式时，OPC UA 应用作为 GDS 服务器的客户端运行，并使用证书管理方法来请求获取证书更新和信任列表更新。
- 采用推送管理模式时，OPC UA 应用作为服务器运行，并提供将 OPC UA GDS 用作 OPC UA 客户端的方法。充当证书管理器的 GDS 用此等方法传送（“推送”）证书和受信列表更新，有关概念说明，请参见下文中的自动证书更新。

目前，仅 S7-1500 CPU 固件版本 V2.9 及以上版本的 OPC UA 才支持推送管理。

使用 GDS 的系统组态

下图显示了与提供证书管理功能的 GDS 相关的各个设备的任务示例。



① 根 CA - 为系统颁发证书的设备（此等证书也可通过其它方式传送，例如通过电子邮件方式）

② 安装有证书管理器的 OPC UA GDS，可创建或签名设备证书、管理信任列表和证书吊销列表 (CRL)，以及将证书和列表写入设备中（推送功能）。对于推送功能，此设备需要 OPC UA 客户端功能。

③ 装有 OPC UA 应用的设备，接收"推送"的证书和列表

STEP 7 版本 V17 及更高版本的自动证书更新概念

GDS 和证书管理器通常合并到一个应用中，但下图中以两个独立的组件显示。

"普通的"OPC UA 客户端之类的设备也可以用作证书管理器，但它们需要支持 Bytestring 数据类型才能传送证书，例如，固件版本为 V2.9 以及更高版本的 S7-1500 CPU 作为 OPC UA 客户端或者具有 GDS 插件的 UA Expert 工具 (Unified Automation)。

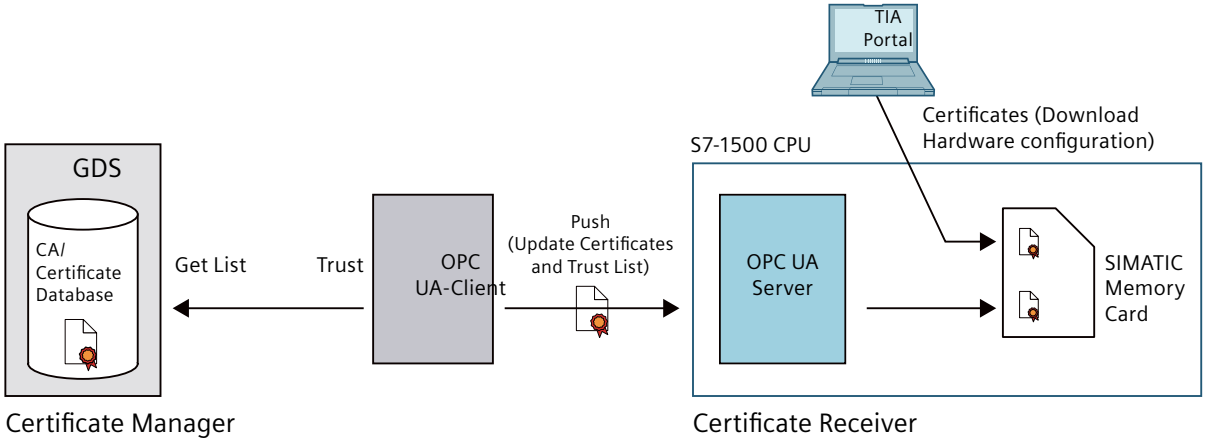
S7-1500 CPU 的 OPC UA 服务器作为证书接收方，可提供 OPC UA 客户端证书读取和写入信任列表和 CRL 时所需的标准方法与属性。

S7-1500 CPU 的 OPC UA 服务器上下文的侧重点是介绍如何使用推送功能为 CPU 提供证书，并与常规方法（通过下载硬件配置）进行了比较。

下图显示了 S7-1500 CPU 固件版本 V2.9 或更高版本中 OPC UA 证书与列表的传输方式：

- 通过加载硬件配置来传输（如果适用，在 CPU 处于 STOP 模式下；证书是硬件配置的组成部分，只能在特定条件下在 RUN 模式下加载；参见"提示：在 RUN 模式中更新下载的证书 (页 67)"）
- 或是在 CPU 处于 RUN 或 STOP 模式时，通过 GDS 推送方法来更新。

两种方法不能同时使用。如果选择在运行系统中通过 GDS 推送功能传送 OPC UA 服务器证书，则必须通过此途径将其它所有证书类型传送到 CPU。



更多信息

有关 OPC UA 证书的更多信息，请参见“OPC UA 证书 (页 175)”部分。

11.2.7.2 推送功能的组态限制

推送功能的证书数量

在 S7-1500 CPU 固件版本 V2.9 及以上版本中，无论何种类型，OPC UA 推送功能的组态限值均为 62 个信任列表条目。

- 每个激活的基于证书的服务（CPU 应用程序）“消耗”一个证书条目和一个私钥条目。
- 证书吊销列表条目 (CRL) 的计数与受信任证书列表条目的计数方式一样。
- 由不同服务（CPU 应用程序）使用的证书计为一个信任列表条目。

推送功能的元素大小（例如证书）

最多 4096 个字节

示例

希望授予最多 62 个 OPC UA 客户端对 OPC UA 服务器的访问权限，并相应填写受信任列表。

在受信任列表中添加“证书吊销列表”条目时，最多只能信任 61 个客户端证书。

不能通过将硬件配置下载到 CPU 来传输更多的 OPC UA 证书。

提示

为了尽可能减少所需证书的数量，建议您通过同一个 CA 对 OPC UA 客户端证书进行签名。

在这种情况下，作为 OPC UA 服务器的 CPU 仅需要相应的 CA 证书和 CRL。通过这些元素，OPC UA 服务器随后可以验证由 CA 签名的所有客户端证书。即，无需将每个客户端证书逐一添加到受信任列表中。

11.2.7.3 设置和下载 GDS 参数

下文介绍了证书更新的所需设置。

要求

- 不同应用程序证书需要使用对应的 STEP 7/TIA Portal 版本和 S7-1500 CPU 固件版本。

  另请参见“证书管理的必备知识 (页 56)”
  - OPC UA 服务器证书需要使用 TIA Portal V17 及更高版本、CPU 固件版本 V2.9
  - Web 服务器证书需要使用 TIA Portal V18 及更高版本、CPU 固件版本 V3.0
- 已设置 CPU 的时间/日期（通常应用于基于证书的通信）
- 已启用 OPC UA 服务器。
- 必须启用 GDS 推送管理使用的服务。例如，必须启用 Web 服务器才能传送 Web 服务器证书。
- 必须至少为 OPC UA 服务器组态一个采用“签名并加密”安全策略的端点。伙伴必须使用此端点。
- 已组态经过身份验证且具备足够功能权限的用户
  用户必须拥有具备**“管理证书”功能权限**的角色。

  该功能权限具有以下要求：
  - 必须在项目树中启用**项目保护**：项目树：“安全设置 > 设置 > 项目保护”(Security settings > Settings > Project protection)。
  - 在项目导航“安全设置 > 用户和角色”(Security Settings > Users and Roles) 中组态的用户拥有具备“管理证书”功能权限的角色。
    TIA Portal < V19：在 CPU 设置的“CPU UA > 常规”(OPC UA > General) 区域中，必须启用以下常规用户管理设置：“通过项目安全设置启用其它用户管理”(Enable additional user management via project security settings)

“具有 OPC UA 功能权限的用户和角色 (页 242)”部分介绍了如何设置功能权限。

激活 GDS

满足上述要求后，仍必须启用 GDS：

1. 在巡视窗口（CPU 参数）中，转到“OPC UA > 服务器 > 常规”(OPC UA > Server > General) 区域。
2. 启用“启用全球发现服务（推送）”(Enable Global Discovery Services (Push)) 选项。

确定使用的证书存储区

使用 GDS 进行管理的证书与通过 TIA Portal (STEP 7) 下载的证书不在同一存储区中。

启用 GDS 推送证书管理后，CPU 的服务（应用程序）同样使用该证书存储区中的证书，这些证书可以在运行期间进行管理。

1. 在 CPU 设置中，导航至"保护与安全 > 证书管理"(Protection & Security > Certificate management) 区域。
2. 选择"运行期间使用证书管理器提供的证书"(Use certificates provided by the certificate management at runtime) 选项。

   另一个选项（使用通过 TIA Portal 组态和下载的证书）则使用从 TIA Portal 下载到 CPU 中的证书（包含组态或组态更改）。在此证书存储区中，运行时无法更新任何证书或信任列表，并且证书只能在特定限制下更新（请参见"提示：在 RUN 模式中更新下载的证书 (页 67)"）。

启用证书失效诊断

如果希望提前收到证书失效通知，请在"保护与安全 > 证书管理"(Protection & Security > Certificate management) 区域中选择"启用证书失效系统诊断事件"(Enable system diagnostics event for the certificate lapsing) 选项。

在输入字段"显示剩余证书有效期的事件："(Show event at remaining certificate validity period of:) 输入百分比值。

这些设置的作用：

- 证书达到该值时，将出现相应的系统诊断消息，该消息在证书失效或刷新后才会消失。
- 如果证书已到期，CPU 将生成相应的系统诊断消息，并在诊断缓冲区中生成一个条目，且维护 LED 指示灯亮起。

示例：
在 2022 年 6 月 1 日通过 GDS 传送的证书的有效期为 2022 年 6 月 1 日至 2022 年 6 月 30 日（30 天）。已在诊断事件中输入百分数值 10。2022 年 6 月 27 日，90% 的有效期将到期。此时，将显示一条消息，指示所传送的证书将于 2022 年 6 月 30 日到期。
无论组态的百分数值是多少，证书的有效期到期后，都将显示一条相应的消息并在诊断缓冲区中输入一个条目，同时维护 LED 指示灯亮起。

下载到 CPU

将组态下载到 CPU 之前，可删除由 GDS 管理的证书。确认删除后，下载完成时将进入配置阶段（参见调试部分）。

下载 CPU 之外的存储卡（读卡器）时，始终会删除该证书存储区。

如果激活全球发现服务（推送）但未推送任何证书，则 OPC UA 服务器上没有任何证书、信任列表或 CRL。

11.2.7.4 GDS 调试

OPC UA 规范第 12 部分对证书管理期间的配置阶段和运行阶段进行了区分定义。

在配置阶段，GDS 或 OPC UA 客户端为 OPC UA 服务器的客户端提供初始信任列表和 CRL。在此阶段中，CPU 的 OPC UA 服务器接受提供的所有客户端证书和列表；与 OPC UA 服务器的"受信任的客户端"设置类似，在运行期间接受所有客户端证书。服务器只能通过这种方式与未知客户端建立连接。例如，客户端无法通过现有证书或信任列表进行身份验证，而只能在接收相应的客户端证书或相应的信任列表后餐呢个进行验证。

配置阶段有信息安全水平低的特点；因此，配置阶段将通过点亮维护 LED 以及在相应的诊断缓冲区中记录条目（需要维护）的方式加以指示。

在运行阶段中，现有的 CRL 将进行更新（举例而言），并且证书和信任列表也将更新。通信在此阶段中是安全的。

要求

在配置阶段，只有具备足够功能权限的授权用户才能建立连接。用户必须拥有具备"管理证书"功能权限的角色。

另请参见"设置和下载 GDS 参数 (页 185)"。

配置阶段的规则

在配置阶段，CPU 的 OPC UA 无法对发起连接建立动作的 OPC UA 客户端进行身份验证。因此，必须遵循以下规则：

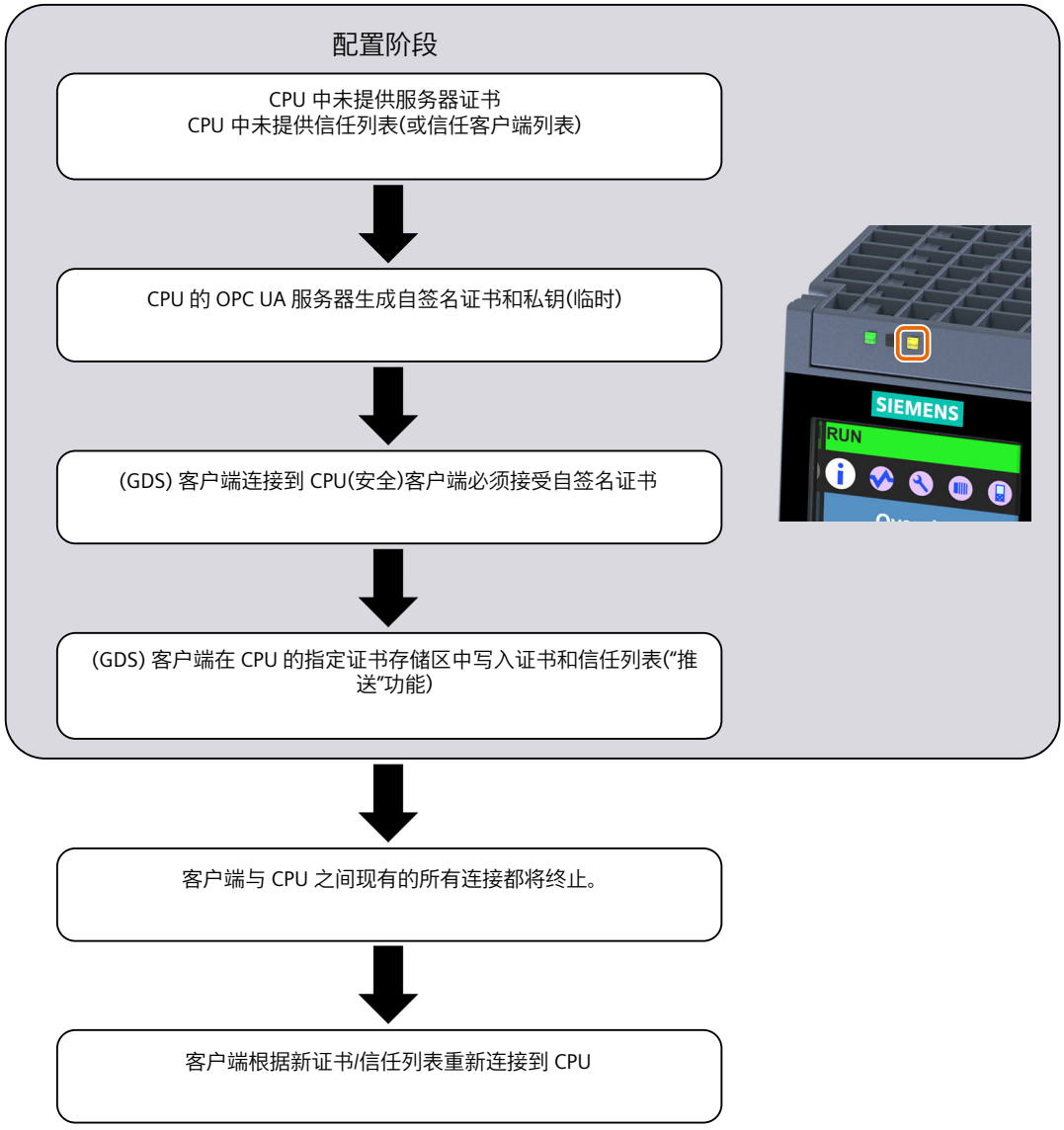
- 提供安全环境，例如仅限调试人员访问 CPU。检查彼此通信的设备是否为正确的设备。
- 限制此阶段的时间。

CPU 通过点亮维护 LED 以及在相应的诊断缓冲区中记录条目（需要维护）的方式指示其处于配置阶段。

配置阶段的顺序

下文中简要介绍了 OPC UA 服务器证书和信任列表配置阶段的相应过程。

Web 服务器证书配置阶段的过程与此类似。与 OPC UA 不同，GDS 客户端仅推送 Web 服务器证书，但不会将信任列表推送到相应的证书存储区中。



进入配置阶段

OPC UA 服务器启动后，CPU 会在满足下面其中一个条件时自动进入配置阶段：

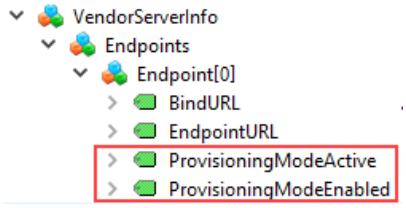
- OPC UA 服务器证书是 CPU 生成的初始自签名证书，尚未替换为有效的服务器证书。
- 信任列表（可信任客户端列表）为空。

CPU 生成的 OPC UA 服务器证书包含 OPC UA 服务器最重要的参数，并且除非已存在有效的服务器证书，否则将在每次接通电源后启动 OPC UA 服务器时重新生成（包括私钥在内）。出于此原因，OPC UA 服务器可能在接通电源后需要更长的启动时间。

在硬件配置已下载后，可在运行时更新的证书，其证书存储区将在下载时删除，或者证书将得到保留，具体取决于设置。这意味着，如果 GDS 处于激活状态并且证书存储区已删除，CPU 将在硬件配置已下载后进入配置阶段。

配置阶段诊断

除了维护 LED 点亮之外，GDS 地址模型还有两个节点可提供信息，指示 CPU 的 OPC UA 服务器是否处于配置阶段：



只有在 GDS 的要求得到满足后（端点安全已签署并加密，另外也已具备管理员功能权限），用户才能出于诊断目的使用图中标记的两个节点。

ProvisioningModeEnabled：表示支持配置阶段

ProvisioningModeActive：表示 CPU 的 OPC UA 服务器处于配置阶段。

配置阶段结束

在满足以下条件时，CPU 将自动结束配置阶段：

- CPU 在配置阶段生成并自签名的证书已由有效的服务器证书覆盖。该有效的服务器证书既可以是自签名证书，也可以是 CA 签名证书。
- CPU 中的信任列表不为空，即，存在用于检查客户端证书的 CA 证书，或者存在可信赖的 OPC UA 客户端的客户端证书。

如果 OPC UA 客户端传送 CA 签名证书并且另外也将 CA 证书添加到信任列表中，则 CPU 的 OPC UA 服务器可自动接受 OPC UA 客户端发送的由同一 CA 签名的所有其它证书。

请求有效服务器证书

自 TIA Portal 版本 V18/S7-1500 CPU 版本 V3.0 起，除了 OPC UA 服务器证书之外，也可将其它服务的证书传送到 CPU 中，例如用于 Web 服务器。

相应的服务（例如 CPU 的 OPC UA 服务器）通过以下步骤接收有效的证书：

1. GDS 客户端（OPC UA 客户端）调用"CreateSigningRequest"方法请求服务器证书：通过证书签名请求 (CSR)。
2. 此 CSR 必须由证书颁发机构 (CA) 签署。
3. 签名的 CSR 必须再传送回 CPU 的 OPC UA 服务器并用作证书。

在客户端具有所需的"管理证书"功能权限时，CPU 的 OPC UA 服务器会支持此方法。

"CreateSigningRequest"方法允许用于以下变体：

- 更新证书，但不创建新的密钥对（使用已有的内部 CPU 密钥）
- 更新证书，并创建新的密钥对（CPU 内部）

此外，也可以使用外部创建的密钥对来生成证书。

注意
关于生成证书的推荐过程 应避免传送私钥；私钥不得离开设备。 因此，我们建议在生成证书时不创建新密钥对，或在 CPU 内创建密钥对。

创建证书但不创建密钥对

- "CreateSigningRequest"方法返回证书签名请求 (CSR)，即包含服务器或服务的特定信息（例如应用程序名称和 URL）的文件 (*.csr)。
- 在 CPU 外部，必须对该 CSR 进行验证并由证书颁发机构 (CA) 进行签名，最后必须返回证书。
- 随后，必须使用"UpdateCertificate"方法将证书传送到 CPU（"推送"）。

在这种情况下，密钥不会离开 CPU。

使用内部创建的密钥对创建证书

此过程与上一节介绍的方法类似，唯一的区别是除了生成 CSR 之外，还会生成一个密钥对。在"CreateSigningRequest"方法的参数中指定将生成密钥对。

在此过程中，私钥不能离开 CPU。

生成新的密钥对会给 CPU 带来很大负载。CPU 会在通信负载的预留区内以更低的优先级处理此请求，且需要的时间较长。此时间的长短取决于 CPU 的性能。

由于在密钥生成过程的较长时间内将完全利用所设通信负载的空间部分，在设置"通信用循环负载"(Cycle load due to communication) 空间部分时，应确保不会超出最大循环时间并且预留空间充足。为此，使用 CPU 的 Web 服务器页面"诊断 > 运行时间信息"(Diagnostics > Runtime information)。此页面显示当前程序/通信负载和用户程序循环时间的信息。就更改后的通信负载对循环时间的影响，用户可通过控制器获得帮助。

使用外部创建的密钥对创建证书

借助诸如可以生成其它密钥的工具来生成证书。

证书和密钥通过"UpdateCertificate"方法传送到 CPU。

由于安全性低，此过程不推荐。

注意
为不同的目标系统使用不同的密钥 对于生产系统，始终使用新生成的密钥。如对项目进行仿真和测试（例如，通过 PC 上的 PLCSIM Advanced 进行），在任何情况下都不得将作为仿真用途的密钥用于生产系统。 应通过设置相应的权限来限制对 PC 式控制器的访问。

11.2.7.5 推送证书管理的地址模型

OPC UA 规范第 12 部分 (OPC 10000-12: Discovery, Global Services) 定义了 OPC UA 服务器的方法和属性，例如允许 GDS 或 OPC UA 客户端更新服务器上的证书和信任列表（“推送证书管理”）。这些方法和属性也包含在 OPC UA 服务器的地址模型中。

下文介绍了 S7-1500 CPU 的 OPC UA 服务器的地址模型中的相关部分。

要求

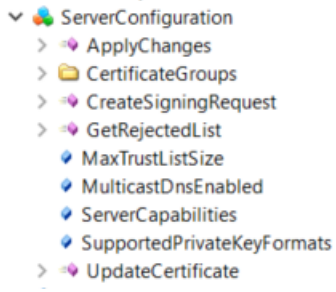
为了使相关方法和属性对 GDS 推送功能可见，必须满足以下要求：

- GDS 已激活。
- 设定的安全策略支持通过签名和加密确保数据的完整性和机密性。
- 使用运行系统功能权限“管理证书”进行访问

GDS 推送功能的地址模型

GDS 推送功能的地址模型相当于 OPC UA 规范的“Information Model for Push Certificate Management”OPC 10000-12: Discovery, Global Services。

“ServerConfiguration”节点下方的结构如下所示：



用于访问地址模型的方法和属性

下文简要介绍了这些方法和属性，并介绍了 S7-1500 CPU 特定地址模型的特殊功能和限制。上文列出的 OPC UA 规范包含一般说明。

此概述表下方给出了有关各个方法的详细说明。

方法/属性（变量）	说明
CreateSigningRequest	此方法用于生成以服务（例如 OPC UA 服务器）私钥进行签名的 PKCS#10 编码证书请求。
UpdateCertificate	此方法用于为 OPC UA 服务器更新服务器证书等。
ApplyChanges	此方法用于，在已设置“ApplyChangesRequired”属性的情况下，在执行之前执行过的方法时，应用安全相关更改。 注 如果证书因“ApplyChanges”而更改，CPU 将中断此证书所担保的连接/会话。 背景：作为担保连接基础的证书不再有效。
GetRejectedList	此方法会返回被 OPC UA 服务器拒绝的证书列表。 S7-1500 CPU 的 OPC UA 服务器目前不会存储被拒绝的证书。此方法返回一个空数组 RejectedList。
ServerCapabilities	S7-1500 CPU 的 OPC UA 服务器不支持该变量。
SupportedPrivateKeyFormats	此变量用于指定允许使用的私钥格式。对于 S7-1500 CPU，仅允许使用“PEM”（字符串数组）
MaxTrustListSize	此变量用于指定信任列表的最大大小。
MulticastDnsEnabled	此变量用于指定是否支持组播 DNS。对于 S7-1500 CPU，该值为“False”。
CertificateGroups	该对象（文件夹）用于组织 OPC UA 服务器所支持的所有证书组。这些证书组包含运行期间可动态更新的对象。例如，每个证书组包含一个信任列表，还包含一个或多个分配给服务（例如 OPC UA 应用程序）的证书。 有关 CertificateGroups 对象的结构以及此对象中提供的方法和属性的详细信息，请参见下一节。

CreateSigningRequest

该方法具有以下参数：

参数	数据类型	说明
[in] certificateGroupId	NodeId	CertificateGroup 对象的 NodeId。
[in] certificateTypeId	NodeId	请求的证书类型。 允许使用的证书类型列表由证书组的“CertificateTypes”变量指定。 例如，OPC UA 服务器使用的证书类型是“RsaSha256ApplicationCertificateType”，Web 服务器使用的证书类型是“HttpsCertificateType”
[in] subjectName	String	证书请求中请求的主题名称。如果未指定，则使用证书的当前主题名称。

参数	数据类型	说明
[in] regeneratePrivateKey	Boolean	True：服务器生成新私钥。该密钥会一直保存到调用签名证书匹配的 UpdateCertificate 方法为止。 False：服务器使用既有私钥。
[in] nonce	ByteString	用于生成新私钥的额外随机数（参见 regeneratePrivateKey）。长度必须至少为 32 字节。
[out] certificateRequest	ByteString	PKCS #10 - DER 编码证书请求。

方法结果代码

结果代码	说明
Bad_InvalidArgument	certificateTypeId、certificateGroupId 或 subjectName 无效。
Bad_UserAccessDenied	当前用户不具备所需的功能权限。

UpdateCertificate

应用：

- 通过 CreateSigningRequest 生成证书。未提供私钥。
- 新私钥和新证书在服务器之外生成。两者均通过 UpdateCertificate 进行更新。
- 证书通过现有证书的私钥生成并签名。未提供私钥。

参数	数据类型	说明
[in] certificateGroupId	NodeId	CertificateGroup 对象的 NodeId。
[in] certificateTypeId	NodeId	请求的证书类型。 允许使用的证书类型列表由证书组的“CertificateTypes”变量指定。
[in] certificate	ByteString	替换现有证书的 DER 编码证书。
[in] issuerCertificates	ByteString	颁发机构证书
[in] privateKeyFormat	String	私钥格式。目前仅支持 PEM。如果未指定 privateKey：零或空字符串。
[in] privateKey	ByteString	按 privateKeyFormat 中指定的格式进行编码的私钥。
[out] applyChangesRequired	Boolean	指示使用新证书前必须调用“ApplyChanges”方法。

方法结果代码

结果代码	说明
Bad_InvalidArgument	certificateTypeId 或 certificateGroupId 无效。
Bad_CertificateInvalid	证书无效或格式不受支持。

结果代码	说明
Bad_NotSupported	私钥无效或格式不受支持。
Bad_UserAccessDenied	当前用户不具备所需的功能权限。
Bad_SecurityChecksFailed	验证证书完整性时出错。

应用更改

该方法没有参数。

方法结果代码

结果代码	说明
Bad_UserAccessDenied	当前用户不具备所需的功能权限。

GetRejectedList

该方法具有以下参数：

参数	数据类型	说明
[out] certificates	ByteStrings	被拒绝的 DER 编码证书列表。 由于不会存储被拒绝的证书，此方法目前返回一个空列表（空数组）。

方法结果代码

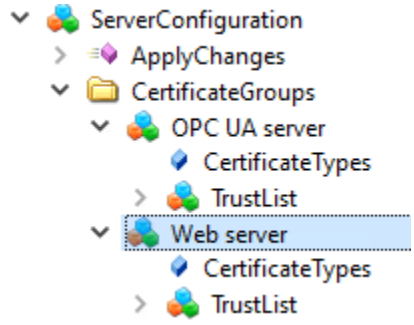
结果代码	说明
Bad_UserAccessDenied	当前用户不具备所需的功能权限。

11.2.7.6 地址模型中的 CertificateGroups

运行过程中，CPU（OPC UA 服务器）中可更新服务或应用程序的证书和信任列表位于地址模型中的“CertificateGroups”对象中，该对象是 S7-1500 CPU 中所有服务都有的一个证书组。对于 OPC UA 服务器证书，证书组的名称为“OPC UA server”。

地址模型中的 CertificateGroup

下图显示了“ServerConfiguration”节点下方“CertificateGroups”对象的结构。



在 STEP 7 (TIA Portal) 中，可更改 CertificateGroups 的 Display Name（例如，“OPC UA server”的显示名称）：

1. 在巡视窗口（CPU 属性）中，导航至“保护与安全 > 证书管理”(Protection & Security > Certificate management) 区域。
2. 启用“运行期间使用证书管理提供的证书”(Use certificates provided by certificate management during runtime) 选项。
3. 更改下表中证书组的组名称 (DisplayName)。允许 7 位 ASCII 格式的 1-64 字符。
   第一列包含已激活的可在运行时传送证书的服务，“ID”列包含用于在 CPU 内部引用证书的固定数字标识符。

以下是“证书管理”(Certificate management) 区域中的显示示例：

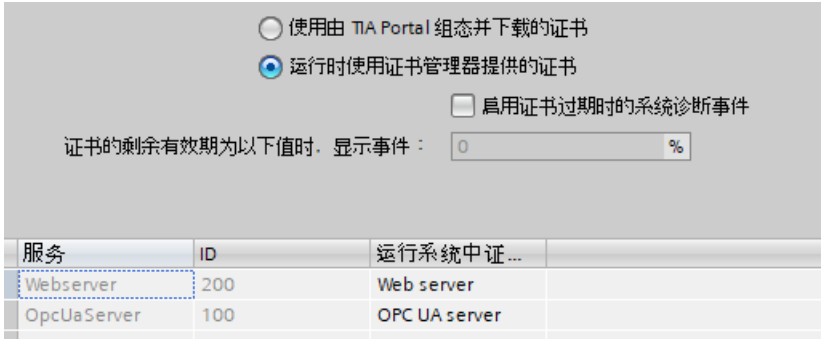


图 11-7 证书管理设置

“CertificateTypes”节点

“CertificateTypes”变量指定分配给服务器应用程序的证书类型的 NodeId。

例如，OPC UA 服务器服务支持“RsaSha256ApplicationCertificateType”CertificateType，Web 服务器支持“HttpsCertificateTypeCertificateType”。

“TrustList”节点

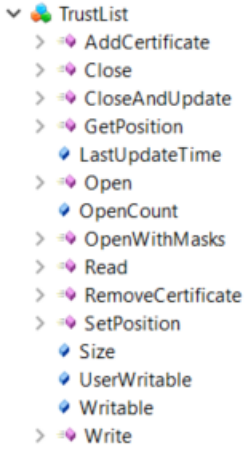
信任列表对象的节点（TrustList 文件）定义了 OPC UA 文件类型（二进制编码流），其中包含有关证书和 CRL 的信息。此信息可在存储卡的“pki store\trusted\issuer”目录中读取和更新。该节点提供用于读取和更新的方法和属性。

节点是 OPC UA 数据类型“TrustListDataType”的实例，其结构如下：

参数	数据类型	说明
specifiedLists	TrustListsMasks	该位掩码用于显示包含信息的列表。
trustedCertificates	ByteStrings	可信任的应用程序证书和 CA 证书列表。
trustedCrls	ByteStrings	“trustedCertificates”列表中证书的 CRL。
issuerCertificates	ByteStrings	验证 CA 签名证书所需的 CA 证书列表。
issuerCrls	ByteStrings	“issuerCertificates”列表中 CA 证书的 CRL。

“TrustList”节点的结构

“TrustList”节点的结构如下所示：



“TrustList”节点的方法和属性

下方是“TrustList”下的各节点的描述，此等节点是对 Object Type "FileType" 方法的补充。TrustList Type 由 FileType 派生而来（参见 OPC 10000-5：OPC 统一架构，第 5 部分：信息模型）。

方法/属性（变量）	说明
LastUpdateTime	此变量用于显示上次更新时间。
OpenWithMasks	此方法允许客户端仅读取部分 TrustList。
CloseAndUpdate	此方法用于关闭 TrustList 文件并应用更改。
AddCertificate	此方法用于将单个证书添加到 TrustList。
RemoveCertificate	此方法用于从 TrustList 中移除单个证书。

方法说明

OPC UA 规范第 12 部分"发现和全局服务"中介绍了上述方法及其结果代码、属性和 TrustList 对象类型。

11.2.8 在 OPC UA 中实现基于角色的安全性

11.2.8.1 关于基于角色的安全性的一些事实

在 OPC UA 中，基于角色的安全性概念在 OPC UA 规范第 18 部分（OPC 10000-18：UA 第 18 部分：基于角色的安全性）中定义。OPC UA 规范的第 3 部分和第 5 部分也与角色定义的个别方面相关，例如，服务器应支持哪些具有权限的角色，以及在地址空间中定义的对象及其属性或特性。

通过固件版本 V4.0 及以上版本的 S7-1500 CPU 实施基于角色的概念，OPC UA 服务器可以精细管理客户端对地址空间的访问。

地址空间中节点的单独 OPC UA 访问权限

借助 OPC UA 基于角色的安全性，用户可组态角色和相关权限，以确定使用 OPC UA 服务器接口/配套规范访问现有命名空间中的节点的可行性。

服务器接口中的角色和权限的组态选项与 TIA Portal 中的用户管理（用户和角色）相结合，可实现有效的访问管理。

OPC UA 的角色/权限与用户和角色的角色/功能权限之间的关系

对于在项目安全设置中作为本地或集中用户管理的一部分定义的用户（用户和角色），考虑将其用于 OPC UA 服务器的访问控制，如下所述：只有拥有"OPC UA 服务器访问"功能权限的用户才允许访问 OPC UA 服务器。

- CPU 的本地或集中用户管理负责对访问 OPC UA 服务器的用户（OPC UA 客户端）进行身份验证。
- CPU 的 OPC UA 服务器负责授权用户（OPC UA 客户端），即，检查用户是否被允许执行所请求的操作（例如，读取或写入节点）。

  在相应的 OPC UA 服务器接口中组态 OPC UA 服务器的访问控制。OPC UA 服务器接口编辑器中已添加"访问控制"(Access control) 区域。可在此区域中进行以下设置：

  - 将现有角色（从项目的"安全设置"(Security settings) >"用户和角色"(Users and roles)）映射到 OPC UA 角色。
  - 为服务器接口的命名空间分配默认权限。此外，还可定义命名空间的访问限制。例如，可指定仅在某些约束条件下向角色授予某项操作的权限，如对消息进行签名和加密。
  - 自定义服务器接口/配套规范各个节点的默认权限：在服务器接口编辑器的单独"访问管理"(Access management) 选项卡中，可自定义节点级别的每个角色的权限（从 CPU 视图：变量）来调整权限。

在操作期间，这会导致为登录用户的会话分配一个或多个角色。分配的角色可确保仅当此节点角色的 OPC UA 权限和组态的访问限制允许（例如读取或写入）时，才可访问地址空间中的节点。

OPC UA 服务器上的用户角色

服务器接口视图可用于带有用户管理数据的所导入配套规范，除了显示来自用户管理的角色（用户和角色）之外，还显示导入的 OPC UA 角色。在此视图中可实现 OPC UA 角色与项目中的角色之间的映射。对于每个角色，还显示在其中定义了角色的命名空间。

对于命名空间 0 中的一些预定义 OPC UA 角色，“用户和角色”(Users and roles) 中没有直接等效角色，例如“AuthenticatedUser”角色。使用用户名/密码成功验证身份的用户将获得分配给“AuthenticatedUser”角色的访问权限。

OPC UA 的角色和权限

在用户和角色编辑器中，可以创建角色，然后可用于服务器接口的进一步组态。

可为服务器接口的元素（变量、方法）和完整的命名空间（即命名空间中的所有节点）分配角色和 OPC UA 权限。

可为角色分配以下 OPC UA 权限：

- Browse - 具有此权限的客户端可以查看对此节点的引用，即，也可以查看除 Value 和 RolePermissions 属性之外的其它属性。
- Read - 具有此权限的客户端可读取 Value 属性。
- Write - 具有此权限的客户端可写入 Value 属性。
- Call - 具有此权限的客户端可以调用相应的方法；对对象、对象类型或方法有效。
- Receive events - 具有此权限的客户端可接收事件（报警和条件）。
- Read Role Permissions - 具有此权限的客户端可读取 RolePermissions 属性。

还可以激活命名空间和单个节点的额外访问限制：

- Signing Required - 仅当使用对消息进行数字签名的安全通道时，客户端才能访问节点。另请参见“应用浏览限制”的解释。
- Encryption Required - 只有在使用加密消息的安全通道时，客户端才能访问节点。另请参见“应用浏览限制”的解释。
- Apply Restrictions to Browse - 如果激活此访问限制，则“需要签名”和“需要加密”描述的要求也适用于浏览权限。

要求

- 固件版本为 V4.0 或更高的 S7-1500 CPU
- TIA Portal V20 及以上版本

规则

- 要为用户或用户组分配角色，请使用用户和角色编辑器（项目树中的安全设置 > 用户和角色）。

  这些用户和用户组也由 OPC UA 服务器使用。CPU 或所连接 UMC 服务器在登录时执行身份验证服务。
- 登录的用户必须具备"OPC UA 服务器访问"功能权限。
- 不支持在运行时添加其它角色。地址空间的所有角色和相关 OPC UA 权限必须在调试和下载到 CPU 之前进行组态。

  如果使用 UMC 服务器进行集中用户管理并将用户组合成组，则可实现更大的用户灵活性。
- OPC UA 提供了授权服务器访问的各种标准；在 OPC UA 术语中，这些是"映射规则"，例如：
  - 用户标识：访问取决于用户身份
  - 应用程序标识：访问取决于客户端证书中指定的 ApplicationUri
  - 端点：访问取决于用于访问服务器的 URL。

  S7-1500 CPU 的 OPC UA 服务器仅通过用户身份验证权限。
- S7-1500 CPU 的 OPC UA 服务器支持"已知角色"，这些角色在 OPC 10000-3 中定义：UA 第 3 部分：地址空间模型。"SecurityKeyServer..."角色在 OPC 10000-14 中定义：UA 第 14 部分 PubSub，第 8.8 节。

已知角色

OPC 基金会在命名空间 0 中预定义了角色 (http://opcfoundation.org/UA/)；对于支持基于角色的安全性的每个 OPC UA 服务器，这些角色无需单独的角色定义即可使用。

"已知角色"示例：

匿名、AuthenticatedUser、观察者、操作员、工程师、主管、ConfigureAdmin、SecurityAdmin 和安全密钥服务角色。

可轻松地将这些角色作为标准 OPC UA 角色添加到 TIA Portal 中。

更多信息

有关为 S7-1500 CPU 组态基于角色的安全性的最新信息，请参见产品信息 (https://support.industry.siemens.com/cs/cn/zh/view/68052815)。

11.3 将 S7-1500 用作 OPC UA 服务器

11.3.1 关于 S7-1500 CPU 的 OPC UA 服务器的有效信息

11.3.1.1 S7-1500 CPU 的 OPC UA 服务器

固件版本 V2.0 及以上版本的 S7-1500 CPU 均可作为 OPC UA 服务器。除了标准 S7-1500 CPU，此特性同样适用于 S7-1500F、S7-1500T、S7-1500C、S7-1500pro CPU、ET 200SP CPU、SIMATIC S7-1500 软件控制器和 PLCSIM Advanced。

约定："S7-1500 CPU"同样包括上述的 CPU 类型。

S7-1500 CPU OPC UA 服务器的基本知识

S7-1500 CPU 上所有集成的以太网接口，均可用于访问该 CPU 的 OPC UA 服务器。

在以下条件中，不能借助 CP 通过自动化系统的背板总线直接访问 CPU 的 OPC UA 服务器：

- 通过 TIA Portal V16 或更高版本进行组态
- S7-1500 CPU 固件版本 2.8 或更高版本以及 CP 1543-1 固件版本 V2.2 或更高版本

有关组态的信息，请参见"访问 OPC UA 应用程序 (页 160)"。

不能借助 CM 通过自动化系统的背板总线直接访问 CPU 的 OPC UA 服务器。

通过客户端进行访问时，服务器将以节点形式保存启用的 PLC 变量和其它信息（请参见"访问 OPC UA 服务器数据 (页 209)"）。这些节点相互连接并形成一个网络。OPC UA 将定义该网络的接入点（已知节点），可导航到下级节点。

通过 OPC UA 客户端，可以读取、监视或写入 PLC 程序中的变量值，并调用服务器中可用的方法。在固件版本 V2.5 及以上版本中，可实现这些方法。具体信息，请参见 关于服务器方法的有用信息 (页 279)。

节点类别

OPC UA 服务器将基于节点提供相应的信息。节点可以是一个对象、变量、方法或属性。

在以下示例中，显示了 S7-1500 CPU 中 OPC UA 服务器的地址空间（摘自 Unified Automation 的 OPC UA 客户端"UaExpert"）。

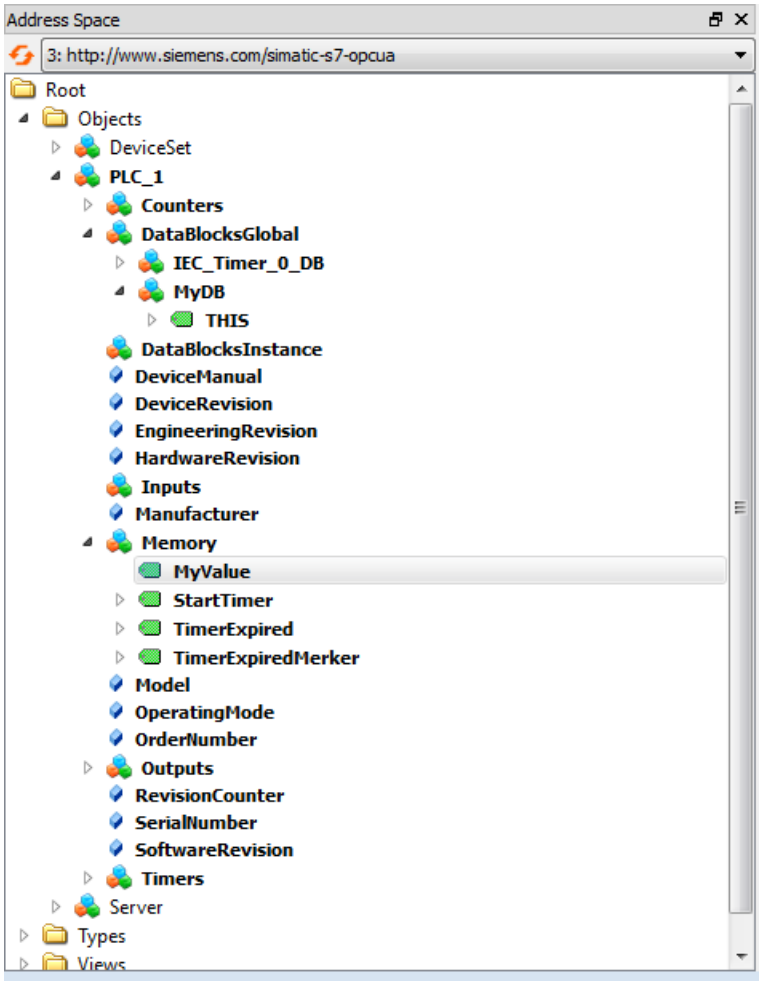


图 11-8 S7-1500 CPU 的 OPC UA 服务器地址空间示例

在上图中，已选择"MyValue"变量（以灰色突出显示）。

此变量位于节点类别为"Object"的"Memory"节点下。

"Memory"位于""PLC_1"节点下（也是一个 Object）。

地址空间

节点通过引用进行连接（如，引用"HasComponent"）。即，节点与子节点之间为层级关系。通过引用，这些节点将构成一个网络。该网络可以为树形结构等。

因此，节点网络也可称为地址空间。可从根节点开始，访问地址空间中的所有节点。

11.3.1.2 OPC UA 服务器的端点

在 OPC UA 服务器的端点，将定义连接的安全级别。基于所用或期望的安全级别，在端点处需执行相应的连接设置。

不同的安全设置

建立安全连接之前，OPC UA 客户端会询问服务器采用哪些安全设置进行连接。服务器将返回服务器提供的所有安全设置（端点）的列表。

端点结构

端点由以下几部分组成：

- OPC 的标识符：“opc.tcp”
- IP 地址：192.168.178.151（在本示例中）
- OPC UA 的端口号：4840（标准端口）

  端口号可组态。
- 消息的安全设置（消息安全模式）：“无”(None)、“签名”(Sign)、SignAndEncrypt。
- 加密和 HASH 程序 (Security Policy)：无，Basic128Rsa15、Basic256、Basic256Sha256（在本示例中）。

下图显示了 OPC Foundation 的“UA Sample Client”。

客户端已与 S7-1500 CPU 中 OPC UA 服务器的端点“opc.tcp://192.168.178.151:4840 - [SignAndEncrypt: Basic256Sha256:Binary]”建立了安全连接：该端点的安全设置为“SignAndEncrypt:Basic256Sha256”。

说明

为服务器选择安全策略最严格的端点

- 仅在 OPC UA 服务器上激活客户端仍支持的最安全的端点。
- 在 OPC UA 服务器上取消激活安全性较低的安全策略。

与服务器建立连接（客户端）

- 与服务器建立连接时，需为应用程序选择合适的安全策略。
- CPU 的 OPC UA 服务器已激活的端点需要相应证书（例如，Basic256Sha256 需要 Sha256 证书）。

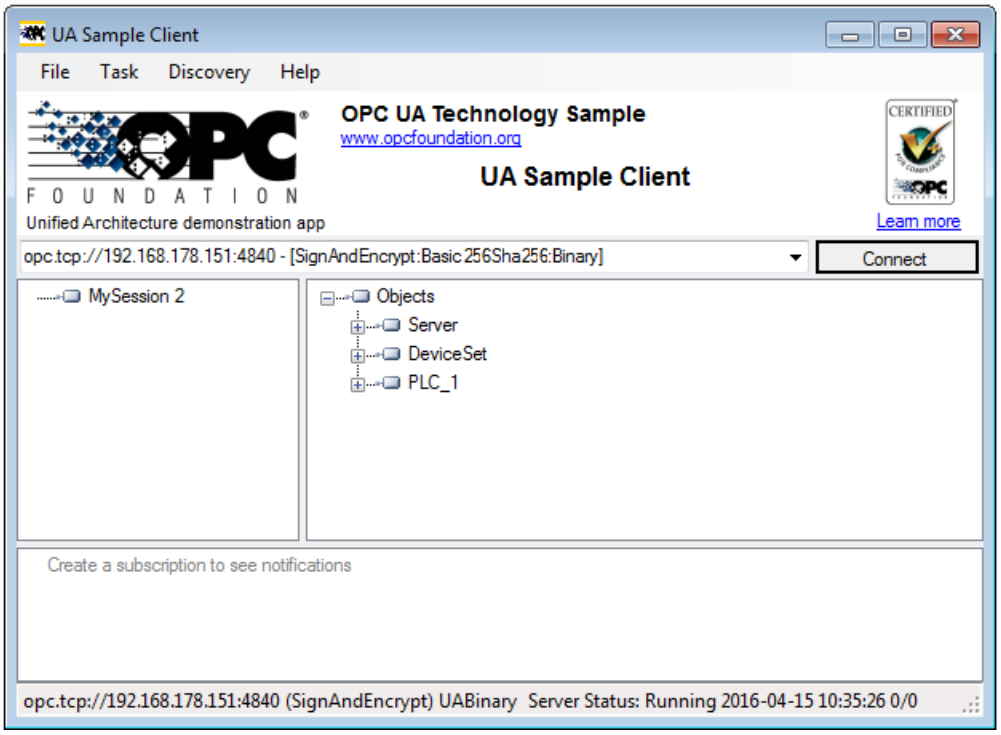


图 11-9 OPC 基金会的"UA Sample Client"程序

仅当 OPC UA 客户端符合服务器端点的安全策略时，才能与服务器端点建立连接。

OPC UA 服务器提供的信息

OPC UA 服务器可提供大量信息：

- 客户端可能访问的 DB 元素以及 PLC 变量的值。
- 这些 PLC 变量和 DB 元素的数据类型。
- 有关 OPC UA 服务器和 CPU 的信息。

因此，客户端可了解并读取相应的特定信息，无需具备之前的 PLC 程序和 CPU 数据。读取 PLC 变量时，无需询问 PLC 程序的研发人员。所有相关信息均存储在服务器中（如，PLC 变量的数据类型）。

OPC UA 服务器信息的显示

可通过以下几种方式：

- 在线：在 OPC UA 服务器运行期间显示所有可用信息。为此，需导航（浏览）服务器的地址空间。
- 离线：可导出基于 OPC 基金会的 XML 架构的 XML 文件。

  在 STEP 7 V15.1 及以上版本中，不导出用户创建的服务器方法（可通过 OPC UA 客户端调用的 FB 实例），请参见"在 OPC UA 服务器上提供方法 (页 279)"。
- 离线并使用 Openness API：在程序中，可通过 TIA Portal 的 API（应用程序编程接口）访问导出 OPC UA 可读取的所有 PLC 变量的功能。需要安装有 .NET Framework 4.0；请参见 TIA Portal Openness，使用脚本实现 SIMATIC 项目自动化 (https://support.industry.siemens.com/cs/ww/zh/view/109477163)。
- 如果您熟知相关语法和 PLC 编程，则可直接访问 OPC UA 服务器，而无需先了解相关信息。

11.3.1.3 数据类型映射

SIMATIC 和 OPC UA 数据类型

SIMATIC 数据类型通常与 OPC UA 数据类型不对应。

S7-1500 CPU 将 SIMATIC 变量（SIMATIC 数据类型）提供给其自己的 OPC UA 服务器作为 OPC UA 数据类型。随后，OPC UA 客户端可以通过服务器接口访问这些 OPC UA 数据类型的变量。

客户端可以从这样的变量中读取属性"DataType"，并在 SIMATIC 中重建原始数据类型。

示例

一个变量的 SIMATIC 数据类型为"COUNTER"。在表中可读取 COUNTER → UInt16。现在了解到不需要进行转换；COUNTER 值以 UInt16 数据类型通过该线路发送。

客户端将通过属性"DataType"检测该变量实际上是否为 SIMATIC 数据类型"COUNTER"，并基于此信息，重新构建该据类型。

表格 11-1 SIMATIC 和 OPC UA 数据类型

SIMATIC 数据类型	OPC UA 数据类型性
BOOL	Boolean
BYTE	BYTE → Byte
WORD	WORD → UInt16
DWORD	DWORD → UInt32
LWORD	LWORD → UInt64
SINT	SByte
INT	Int16
DINT	Int32
LINT	Int64

SIMATIC 数据类型	**OPC UA 数据类型性**
USINT	Byte
UINT	UInt16
UDINT	UInt32
ULINT	UInt64
REAL	Float
LREAL	Double
S5TIME	S5TIME → UInt16
TIME	TIME → Int32
LTIME	LTIME → Int64
DATE	DATE → UInt16
TIME_OF_DAY (TOD)	TOD → UInt32
LTIME_OF_DAY (LTOD)	LTOD → UInt64
DATE_AND_TIME (DT)	DT → Byte[8]
LDT	DateTime
DTL 特殊说明：只能使用 OPC UA 客户端完整描述该结构。该结构中的各元素仅支持只读访问（如"YEAR"）	映射为结构
CHAR	CHAR → Byte
WCHAR	WCHAR → UInt16
STRING （代码页或 1252 或 Windows-1252）	STRING → String
WSTRING （UCS-2；通用编码字符集）	String
TIMER	TIMER → UInt16
COUNTER	COUNTER → UInt16

数组

OPC UA 通常采用数组访问方式进行读写操作问，即带有下标和长度。一个单变量实际上就是一各特殊的数组（下标为 0，长度为 1）。只是在该线路上重复发送此数据类型。对于变量，"DataType"属性指示基本数据类型。属性"ValueRank"和"ArrayDimensions"用于显示当前是否使用数组进行处理以及该数组的大小。

基于数组的数据类型

一些 SIMATIC 数据类型的 OPC UA 值映射到字节数组中。这些数据类型的数组随后会映射为二维数组。

示例：SIMATIC 数据类型 DATE_AND_TIME (DT) 在 OPC UA 侧映射到 8 字节数组 (Byte[8])，见上表。定义 SIMATIC 数据类型 DATE_AND_TIME (DT) 的数组时，会将其视为二维数组。

这会影响 OPC_UA_NodeAdditionalInfo 和 OPC_UA_NodeAdditionalInfoExt 系统数据类型的使用，例如：

对于上述数据类型，必须为多维数组使用系统数据类型 OPC_UA_NodeAdditionalInfoExt，而不是 OPC_UA_NodeAdditionalInfo。

结构

结构作为 ExtensionObject 进行传送。S7-1500 服务器使用二进制表示来在线路上传输 ExtensionObjects；各结构元素相继出现。在前面的是数据类型的 NodeId；客户端使用其来建立结构。

对于 OPC UA 规范 V1.03 及以下版本，要实现该目的，客户端需读取、解码和解析完整的 DataTypeDictionary（除非已通过 XML 导入功能离线学习此库）。

自 OPC UA V1.04 起，可使用 DataTypeDefinition 属性，更为轻松便捷地进行读取和解析。客户端仅在第一次访问期间或之前一次性确定结构设置，随后会在会话期间使用此信息。

特殊 SIMATIC 数据类型

上表中不存在以及无法定义为结构或 PLC 数据类型元素的 SIMATIC 数据类型不受 OPC UA 客户端支持。

举例来说，此类数据类型有“ANY”或“POINTER”指针、函数块“Block_FB”、函数“Block_FC”或硬件数据类型“REMOTE”。

如果选择不受支持的数据类型，则将生成一条错误消息。

更多信息

有关基本数据类型、数组和结构映射的更多详细信息，请参见 OPC UA 规范第 6 部分“映射”（参见 OPC UA BINARY）。

对于 SIMATIC S7-1500 OPC UA 服务器中的数组与数据类型 DTL 和 LDT，必须考虑哪些因素？常见问题解答 (https://support.industry.siemens.com/cs/cn/zh/view/109766726)

11.3.1.4 OPC UA 服务器运行期间的行为

运行过程中的 OPC UA 服务器

激活服务器并将项目下载到 CPU 后，S7-1500 CPU 的 OPC UA 服务器会启动。

此处介绍了如何激活 OPC UA 服务器。

CPU STOP 操作状态的行为

即使 CPU 切换到"STOP"模式，已激活的 OPC UA 服务器仍然保持运行状态。OPC UA 服务器会继续响应来自 OPC UA 客户端的请求。

服务器响应的详细信息：

- 如果用户请求 PLC 变量的值，则会获得 CPU 切换到或被设置为"STOP"模式之前的最新值。
- 如果用户向 OPC UA 服务器写入值，则 OPC UA 服务器将接受这些值。

  但是，由于用户程序不是在"STOP"模式下执行的，所以 CPU 不会处理这些值。

  尽管如此，OPC UA 客户端仍可从 CPU 的 OPC UA 服务器读取 STOP 模式下所写入的值。

  在重新启动过程中，CPU 将在开始执行 PLC 变量时覆盖 STOP 模式下所写入的值。
- 调用某个服务器方法时，系统将因为服务器方法（用户程序）当前未运行而输出错误消息 16#00AF_0000 (BadInvalidState)。
- 操作模式转换 (STOP > RUN or RUN > STOP) 时，与 OPC UA 服务器的连接保持激活。例外：加载 OPC UA 相关数据，具体请参见下一章节。

下载到 CPU 可能会影响 OPC UA 服务器

如果在 OPC UA 服务器运行时加载 CPU，则可能需要根据加载的对象停止并重新启动服务器。在这种情况下，活动连接会中断，必须在服务器重新启动后重新建立连接。

重新启动的持续时间主要取决于以下参数：

- 数据结构的范围
- OPC UA 地址空间中可见的变量数
- 关于根据 OPC UA 规范 (<= V1.03) 向下兼容数据类型定义的设置（启用 TypeDictionary）
- 有关通信负载和最短循环时间设置的更多信息，请单击此处 (页 348)

对于 V2.8 以下的 CPU 固件版本，每次下载到 CPU 时 OPC UA 服务器都会停止，之后再重新启动。

自固件版本 V2.8 起，OPC UA 服务器的行为已得到如下优化：

- 在 CPU 的 STOP 操作状态下载对象时，OPC UA 服务器仍始终停止，之后再重新启动。在这种情况下，STEP 7 不会显示警告。
- 在 CPU 的 RUN 操作状态下载对象时，OPC UA 服务器仅在加载的对象与 OPC UA 相关或者可能与之相关的情况下才会停止。由于 OPC UA 数据发生修改，OPC UA 服务器会在重新初始化后再重新启动。

  在将 OPC-UA 相关对象加载到 CPU 并停止 OPC UA 服务器之前，STEP 7 会在加载预览对话框中显示警告。随后，用户可以决定是在完成下载操作后重新启动服务器，还是取消下载操作。这类警告仅在 OPC UA 服务器运行时显示。如果 OPC UA 服务器未启用，修改后的 OPC UA 数据对下载过程没有影响。

示例

- 只需要向程序中添加其他代码模块。
  数据块以及输入、输出、标记、时间或计数器均不受影响。
  加载期间的反应：正在运行的 OPC UA 服务器不中断。
- 需要加载新数据模块并将数据模块标记为非 OPC-UA 相关：
  加载期间的反应：正在运行的 OPC UA 服务器不中断。
- 需要覆盖数据模块。
  加载期间的反应：显示警告，提示您服务器即将重启。
  背景：STEP 7 无法确定更改是否与 OPC-UA 数据相关。

通过 OPC UA 服务器读取 CPU 的操作模式

通过 OPC UA 服务器可读出 CPU 模式，如下图所示：

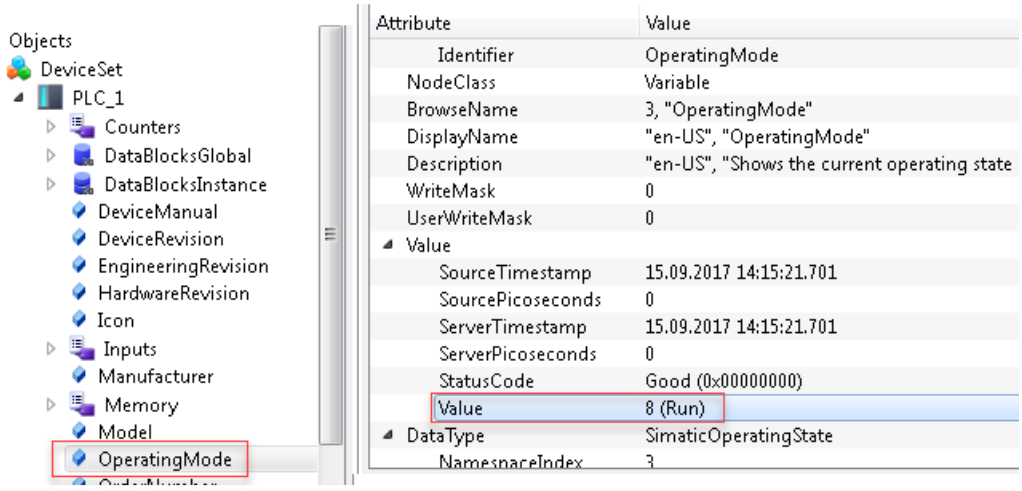


图 11-10 通过 OPC UA 服务器读取 CPU 的操作模式

除了 CPU 的操作模式，还可读取手册 (DeviceManual) 或固件版本 (HardwareRevision) 中的信息。

11.3.2 访问 OPC UA 服务器数据

11.3.2.1 OPC UA 服务器的客户端访问和本地访问

OPC UA 服务器为网络中的 OPC UA 客户端提供大量信息。以下部分介绍了在 OPC UA 服务器的地址空间中提供 CPU 变量（PLC 变量和 DB 元素）的几种方式。

在 OPC UA 地址空间中通过服务器接口提供 CPU 变量

将 CPU 变量自动传输到 OPC UA 服务器地址空间最便捷的方式：

- 在 CPU 的 OPC UA 属性中，激活标准 SIMATIC 服务器接口。
  为 OPC UA 发布的所有 CPU 变量随后也会自动在 CPU 名称下的 OPC UA 地址空间中可用。

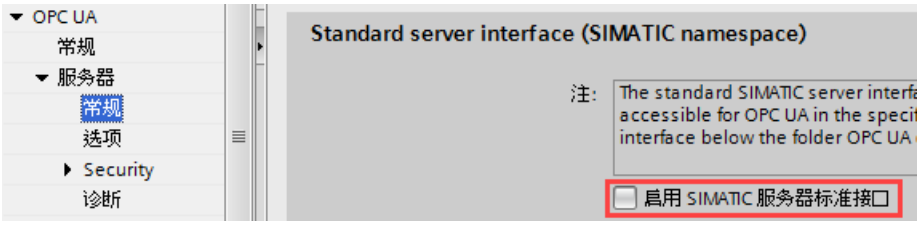


图 11-11 OPC UA 服务器的标准 SIMATIC 服务器接口

OPC UA 服务器接口的使用方式更灵活、程序结构更清晰，因为只需在项目树中组态服务器接口（在 CPU 下方的"OPC UA 通信"文件夹）。用户自定义 OPC UA 服务器接口可以轻松映射 OPC UA 变量和 CPU 变量（本地数据）。

OPC UA 服务器接口

Browse name	节点类型	访问级别	本地数据
myServerInterface	Interface	---	
myOPC_UA_Variable	BOOL	RD	"myDataBlock"."myVarBool"
myOPC_UA_Variable2	Word	RD/WR	"myDataBlock"."myVarWord"

图 11-12 创建映射了 CPU 变量的用户自定义服务器接口

下文以两个 S7-1500 CPU 为例清楚地说明了 OPC UA 客户端和 OPC UA 服务器之间的数据交换。

此处，作为客户端的 S7-1500 CPU 将值写入 OPC UA 服务器的 OPC UA 变量。CPU 变量和 OPC UA 变量之间的映射看起来就像 OPC UA 客户端直接将值写入 CPU 变量一样。对于 S7-1500 客户端 CPU，将“OPC_UA_WriteList”指令与数据交换所需的附加指令结合使用。

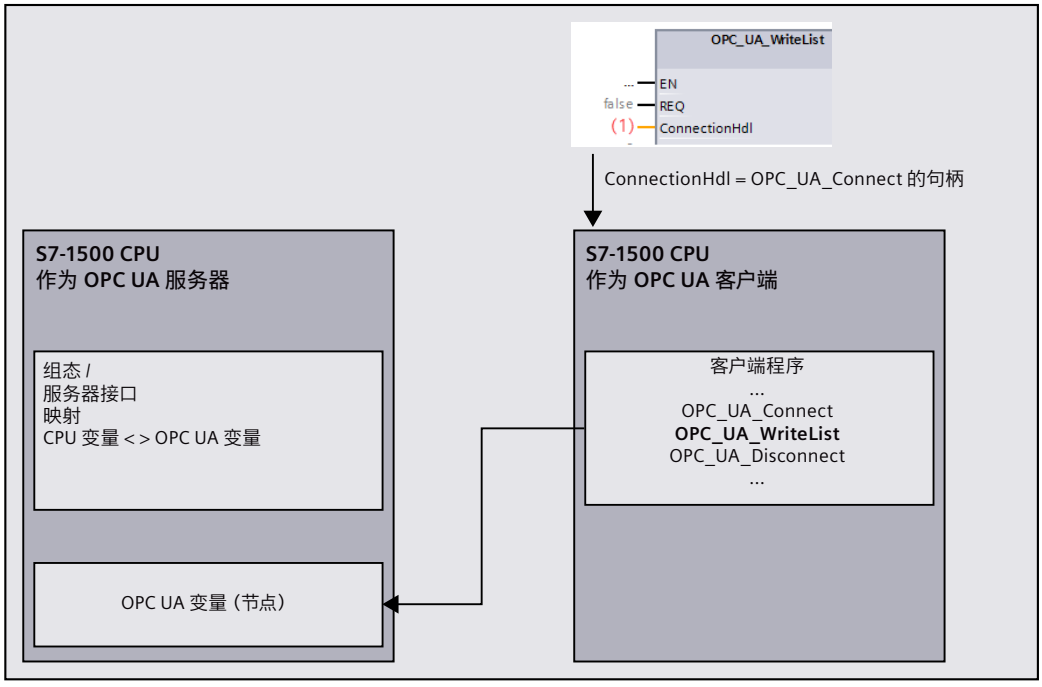


图 11-13 客户端访问服务器的 OPC UA 变量

将 CPU 的变量值直接写入 OPC UA 变量中（设置 OPC UA DataValue）

自固件版本 V3.0 起，S7-1500 CPU 不仅可映射变量，还可以通过“OPC_UA_WriteList”指令将值直接写入服务器的本地 OPC UA 变量节点。通常，CPU 客户端程序中的“OPC_UA_WriteList”指令用于将值写入远程 OPC UA 服务器的 OPC UA 变量中。

在服务器中使用“OPC_UA_WriteList”的优势：除了该值之外，还可以为 OPC UA 变量节点提供以下附加信息：

- SourceTimestamp
- StatusCode

OPC UA 内置有一个“DataValue”数据类型。DataValue 是一个结构，用于记录 Value 以及 SourceTimestamp 和 StatusCode 作为该值的附加信息。DataValue 结构仅供 OPC UA 服务使用，不能直接在 CPU 程序中写入该结构的元素。只有通过使用“OPC_UA_WriteList”指令才能进行写访问。

应用选项

CPU 变量无法记录指示最后一次将值写入 CPU 变量的时间戳。如果通过服务器接口映射 CPU 变量和 OPC UA 变量，则 OPC UA 服务器不会将 SourceTimestamp 设置为 CPU 变量发生变化的时间，而是设置为服务器中“采集”值的时间；例如，通过读取服务或在订阅环境中采样。

例如，如果使用“OPC_UA_WriteList”将 DataValue 直接写入 OPC UA 变量节点，则可以提供在程序中确定的时间戳作为值的 SourceTimestamp。

用于设置 DataValues 的“OPC_UA_WriteList”指令的原理功能

例如，DataValue 结构建模为 UDT，并且此数据类型的变量被传送到“OPC_UA_WriteList”指令。然后，该指令将变量的元素持续传输到 OPC UA 变量节点。

“ConnectionHdl”指令参数的值决定了“OPC_UA_WriteList”指令的功能：“正常”客户端指令或写入本地 OPC UA 变量节点的指令。在后一种情况下，OPC UA 客户端可以读取带有附加信息的值并相应地对其进行评估。

该原理如下图所示，一种情况是使用任意客户端，另一种情况是使用 S7-1500 CPU 作为 OPC UA 客户端。使用 S7-1500 CPU 客户端时，显示了将 DataValue 元素分配给“OPC_UA_ReadList”指令的相应指令参数的情况。可以完全访问 DataValue 结构的所有元素。

“OPC_UA_WriteList”指令的“Read”(-42) 值会使服务器写入本地 OPC UA 变量节点。

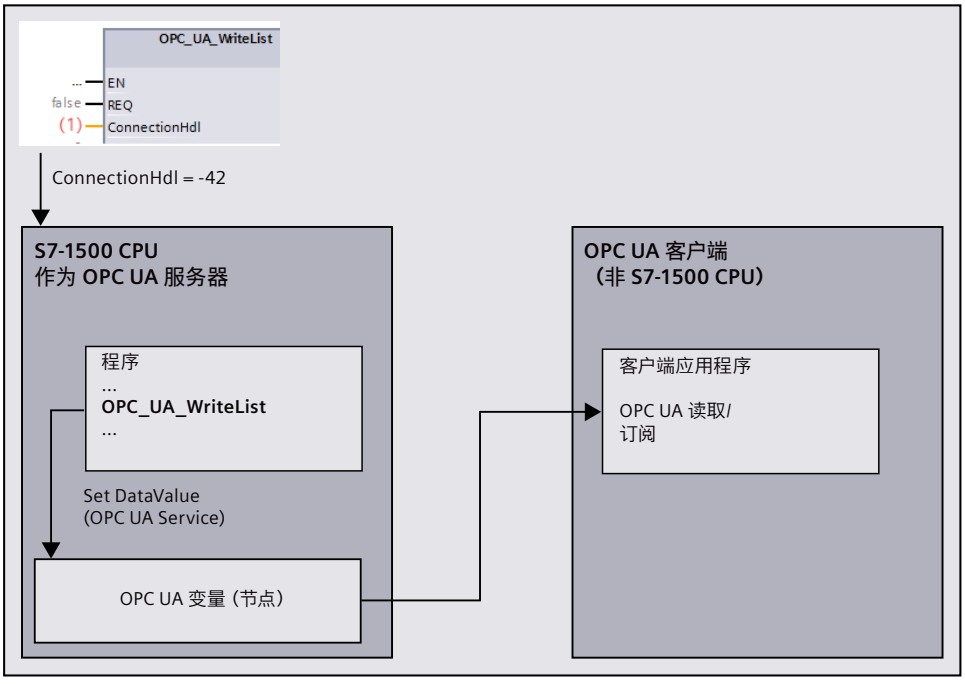


图 11-14 设置服务器本地 OPC UA 变量的数据值

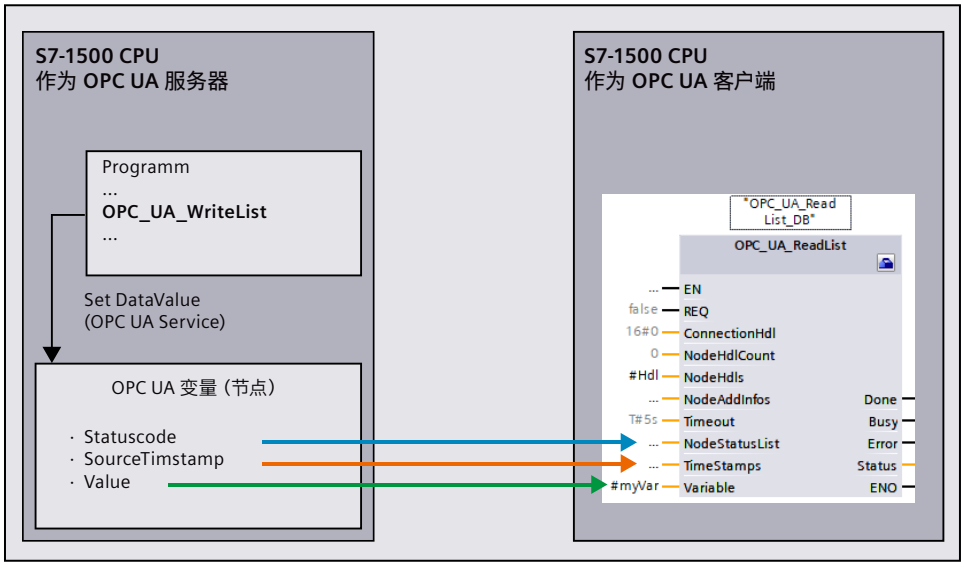


图 11-15 客户端读取数据值（S7-1500 CPU 的 OPC UA 变量）

其它应用选项

如果 OPC UA 客户端在订阅环境中向 S7-1500 CPU 注册值更改（受监视的项目），并且为相应的 DataValue 提供了上述值和附加信息，则对附加信息的更改也可以触发通知。

示例：二进制值变化非常快，在采样间隔内就可能恢复其原始值（快速变化 TRUE > FALSE > TRUE）。未检测到值的变化。但是检测到时间戳的变化。同样，当 StatusCode 发生变化时，即使值没有变化，也可以触发通知。

限制

- OPC UA 客户端只允许读取 OPC UA 变量；必须为 OPC UA 变量相应地设置读/写权限的“AccessLevel”属性。
- 只能在本地设置用户自定义的服务器接口的 OPC UA 变量。
- 在用户自定义的服务器接口中，直接写入的 OPC UA 变量不得映射到 CPU 变量。

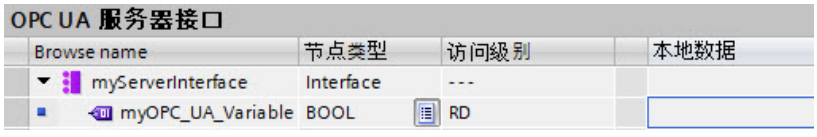


图 11-16 用户自定义服务器接口

有关“设置 OPC UA DataValue”时“OPC_UA_WriteList”指令用法的详细信息，请参见通信指令帮助的相应部分。

设置类型为数组和结构的 OPC UA DataValue 属性

设置 OPC UA DataValue 属性时，如果使用"OPC_UA_WriteList"设置类型为结构或数组的 OPC UA 变量，则系统将填充该结构或数组的所有元素。

不应将类型为结构或数组的单个元素构建为较低级的 OPC UA 变量。
原因：在该服务器地址空间中，如果将类型为数组或结构的各元素构建为较低层级的各个节点，则系统不会自动填充这些节点。对于 OPC UA 服务器，由于这些单独的节点没有 CPU 变量进行映射，因此与结构或数组类型的上一级 OPC UA 变量无关。

要填充这些单独建模的节点，则需在程序中创建单独的元素作为各自的 DataValue 结构。

提示：为确保 OPC UA 客户端能够同时了解相关节点发生的变更，可在同一个"OPC_UA_WriteList"调用中设置所有相关 OPC UA 变量的值。

更多信息

借助应用示例 (https://support.industry.siemens.com/cs/cn/zh/view/109820694)深入了解"设置 OPC UA DataValue 属性"主题。

有关如何协调 CPU 变量读写权限的信息，请参见"协调 CPU 变量的读写权限 (页 216)"部分。

有关如何创建用户自定义服务器接口的信息，请参见"创建用户自定义服务器接口 (页 258)"部分。

11.3.2.2 管理读写权限

启用 OPC UA 的 PLC 变量和 DB 变量

如果 OPC UA 启用了 PLC 变量（默认设置），则 OPC UA 客户端对 PLC 变量和 DB 变量具有读写权限。对于已启用的变量，已选中复选框"可从 HMI/OPC UA 访问"(Accessible from HMI/OPC UA)。

可在 TIA Portal 的设置中更改默认设置："选项"(Options) 菜单中的命令"设置 > PLC 编程 > 常规"(Settings > PLC programming > General)。"块接口/数据块元素"(Block interface/data block elements) 区域中包含相应选项。

以下为数组数据块的示例：

MyDB

	名称	数据类型	可从 HMI/OPC UA 访问	从 HMI/OPC UA 可写	在 HMI 工程组态中可见
	MyDB	Arr...	☑	☑	☑
	MyDB[0]	Int	☑	☑	☑
	MyDB[1]	Int	☑	☑	☑
	MyDB[2]	Int	☑	☑	☑
	MyDB[3]	Int	☑	☑	☑
	MyDB[4]	Int	☑	☑	☑
	MyDB[5]	Int	☑	☑	☑
	MyDB[6]	Int	☑	☑	☑
	MyDB[7]	Int	☑	☑	☑
	MyDB[8]	Int	☑	☑	☑
	MyDB[9]	Int	☑	☑	☑

图 11-17 启用 OPC UA 变量的 PLC 变量和 DB 变量

OPC UA 客户端可一次性完整读取该数组（请参见"节点寻址 (页 163)"）。同时，该数组中的所有元素都将激活"从 HMI/OPC UA 可访问"(Accessible from HMI/OPC UA) 和"从 HMI/OPC UA 可写入"(Writable from HMI/OPC UA) 复选框。

结果：OPC UA 客户端既可以对这些元素进行读操作，也可以对其进行写操作。

撤消写入权限

如果要对一个变量进行写保护，则可取消选中该变量的"从 HMI/OPC UA 可写"(Writable from HMI/OPC UA) 选项。这将取消 OPC UA 客户端和 HMI 设备的写入权限。

结果：OPC UA 客户端和 HMI 设备仅具备读权限。OPC UA 客户端将无法为该变量赋值，因此也无法影响 S7 程序的执行。

撤消读写权限

要对变量进行读写保护，可禁用该变量的"从 HMI/OPC UA 可访问"(Accessible from HMI/OPC UA) 选项（不选中该复选框）。这样，OPC UA 服务器将从地址空间中删除该变量。OPC UA 客户端无法再访问该 CPU 变量。

结果：OPC UA 客户端和 HMI 设备无法对该变量进行读取和写入。

结构的读写权限

如果移除某结构组件的读写权限，则无法将该结构或数据块作为一个整体进行写入或读取。

如果移除某个 PLC 数据类型 (UDT) 中各组件的读写权限，则将同时移除该数据类型的所有数据块的相应权限。

在 HMI 工程组态中可见

"在 HMI 工程组态中可见"(Visible in HMI Engineering) 选项将影响西门子的工程组态工具。如果禁用选项"在 HMI 工程组态中可见"(Visible in HMI Engineering)（未勾选），则无法在 WinCC (TIA Portal) 对该变量进行组态。

该选项不会对 OPC UA 产生任何影响。

规则

- 如果与其它系统（控制器、嵌入式系统或 MES）进行通信时需要，则只能在 STEP 7 中对 PLC 变量和数据块变量进行读取访问。

  而不应启用其它 PLC 变量。
- 如特定的 PLC 变量和数据块变量确实需要写入权限，则只允许通过 OPC UA 进行写入访问。
- 如果为数据块的所有元素复位"可通过 HMI/OPC UA 访问"(Accessible from HMI/OPC UA) 选项，则 OPC UA 客户端的数据块不再显示在 S7-1500 CPU 的 OPC UA 服务器地址空间中。
- 还可以阻止集中访问整个数据块（请参见管理整个 DB 的读写权限 (页 215)）。此设置会"否决"DB 编辑器中组件的设置。

更多信息

有关如何协调 CPU 变量的读写权限的信息，请参见“协调 CPU 变量的读写权限 (页 216)”部分。

11.3.2.3 管理整个 DB 的读写权限

隐藏 OPC UA 客户端的 DB 或 DB 内容

可通过 OPC UA 客户端轻松阻止对整个数据块的访问。

利用此选项，相应 DB 的数据（包括函数块的示例 DB）对 OPC UA 客户端保持隐藏。

在模式设置中，数据块可通过 OPC UA 客户端进行读写。可在 TIA Portal 的设置中更改此默认设置：“选项”(Options) 菜单中的命令“设置 > PLC 编程 > 常规”(Settings > PLC programming > General)。“新块的默认设置”(Default settings for new blocks) 区域中包含相应选项。

操作步骤

要对 OPC UA 客户端完全隐藏某一数据块或避免通过 OPC UA 客户端对数据块进行写访问，请按以下步骤操作：

1. 在项目树中选择要保护的数据块。
2. 选择“特性”(Properties) 快捷菜单。
3. 选择“属性”(Attributes) 区域。
4. 根据需要选中/清除“DB 可从 OPC UA 访问”(DB Accessible from OPC UA) 复选框。

图 11-18 隐藏 OPC UA 客户端的 DB 或 DB 内容

说明

对 DB 编辑器中设置的影响

如果使用此处描述的 DB 属性隐藏 DB，则 DB 编辑器中组件的设置将不再相关；不能再访问或写入单个组件。

提示：使用所有程序块的总览图

如果使用多个数据块，则可以使用“程序块”(Program blocks) 文件夹的详细总览图有选择的激活或禁用 OPC UA 可访问性。

请按以下步骤操作：

1. 在项目树中选择“程序块”(Program blocks) 文件夹。
2. 在“视图”(View) 菜单中，选择“总览图”(Overview) 命令。
3. 选择“详细信息”(Details) 选项卡。
   将显示块及其属性的总览图。
4. 确保选中“可通过 OPC UA 访问的数据块”(Data block accessible via OPC UA) 列。
5. 仅选择要通过 OPC UA 访问的数据块。

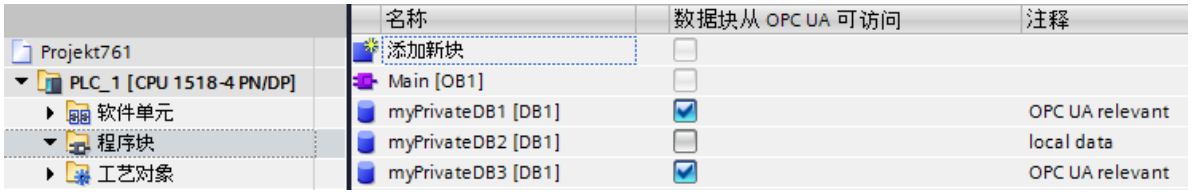


图 11-19 程序块概述

11.3.2.4 协调 CPU 变量的读写权限

信息模型 (OPC UA XML) 中读写权限的定义

在 OPC UA 信息模型中，属性“AccessLevel”调节对变量的访问权限。

AccessLevel 按位定义：

位 0 = CurrentRead，位 1 = CurrentWrite。位组合的含义如下：

- AccessLevel = 0：无访问权
- AccessLevel = 1：只读
- AccessLevel = 2：只写
- AccessLevel = 3：读+写

读写权限（读+写）的分配示例

```
<UAVariable NodeId="ns=3;s=&quot;Data_block_2&quot;.&quot;Static_1&quot;"
BrowseName="3:Static_1"
ParentNodeId="ns=3;s=&quot;Data_block_2&quot;"
DataType="INT"
AccessLevel="3">
    <DisplayName>Static_1</DisplayName>
```

STEP 7 中读写权限的定义

定义变量时，使用“从 HMI/OPC UA 可访问”(Accessible from HMI/OPC UA) 和“从 HMI/OPC UA 可写”(Writable from HMI/OPC UA) 特性来指定访问权限。

读写权限的分配示例

Name	Data type	Accessible from HMI/OPC UA	Writable from HMI/OPC UA
Static_1	Int	☑	☑
<Add new>		☑	☑

图 11-20 读写权限的分配示例

读写权限之间的交互

如果已导入 OPC UA 服务器接口，并且在此 OPC UA XML 文件中设置了 AccessLevel 属性，则通过以下规则定义读写权限：各个设置的最不广泛访问权限适用。

示例

- OPC UA 服务器接口中的 AccessLevel = 1（只读）
- 在 PLC 变量表中选择了“从 HMI/OPC UA 可访问”(Accessible from HMI/OPC UA) 和“从 HMI/OPC UA 可写”(Writable from HMI/OPC UA)。

结果：该变量为只读。

规则

如果需要写权限：

- AccessLevel = 2 或 3
- 启用“从 HMI/OPC UA 可写”(Writable from HMI/OPC UA)

如果需要读权限：

- AccessLevel = 1（AccessLevel 3 也可以，但是具有误导性。该设置表示 OPC UA 客户端具有读写权限）
- 启用“从 HMI/OPC UA 可访问”(Accessible from HMI/OPC UA)，禁用“从 HMI/OPC UA 可写”(Writable from HMI/OPC UA)

如果不授予读写权限（无访问权限）：

- AccessLevel = 0
- 禁用“从 HMI/OPC UA 可访问”(Accessible from HMI/OPC UA)

  要阻止所有访问权限，需满足两个条件之一。在这种情况下，请检查 OPC UA 服务器接口中的变量实际上是否完全需要。

访问表

如果要通过 OPC UA 进行访问，必须设置“从 HMI/OPC UA 可访问”(Accessible from HMI/OPC UA)。如果要允许 OPC UA 客户端写入变量/DB 元素，必须设置“从 HMI/OPC UA 可写”(Writable from HMI/OPC UA)。

请参见下表了解实现的访问权限。

表格 11-2 访问表

OPC UA XML	**STEP 7 (TIA Portal)，例如变量表**		
AccessLevel	从 **HMI/OPC UA** 可访问	从 **HMI/OPC UA** 可写	实现的访问权限
0	x	x	无访问权
x	0	x	无访问权
1	启用	x	只读
2	启用	禁用	无访问权
3	启用	禁用	只读
2	启用	启用	只写
3	启用	启用	读+写

（x = 无关）

11.3.2.5 CPU 变量的一致性

“AccessLevelEx”属性会扩展访问特性

自固件版本 V2.6 起，S7-1500 CPU 的 OPC UA 服务器不仅支持“AccessLevel”属性（参见“协调 CPU 变量的读写权限 (页 216)”），还支持“AccessLevelEx”属性，该属性除了提供已介绍的用于读取权限和写入权限的位之外，还提供关于 OPC UA 变量一致性的信息。新属性自 OPC UA 规范的版本 V1.04 起引入（第 3 部分，地址空间模型）。

读取一致性特性

在 OPC UA 服务器的 OPC UA 信息模型中，属性“AccessLevel”定义访问权限。

AccessLevel 按位定义；此时，相关位为：

- 位 0 = CurrentRead
- 位 1 = CurrentWrite
- 位 2 到 7 与 S7-1500 CPU 的 OPC UA 服务器无关

关于读取和写入权限的部分中介绍了位组合的含义：

还添加了下列用于表示一致性的位：

- 位 8 = NonatomicRead；如果不能一致地读取变量，此位会置位。对于变量的读取一致性，位 8 = 0。
- 位 9 = NonatomicWrite；如果不能一致地写入变量，此位会置位。对于变量的写入一致性，如果未批准写入权限的情况，位 9 = 0。

示例

OPC UA 变量（结构体）可读取且可写入，但读取和访问权限不一致。

因此：位 0、1、8 和 9 会置位：AccessLevelEx =“771”(1+2+256+512)。

另一结构体为只读。

因此：位 0 和 8 会置 1，位 1 和位 9 不会置位：AccessLevelEx =“257”(1+0+256+0)。

服务器中属性的处理

“AccessLevelEx”属性仅可用于 OPC UA 服务器。该属性不存在于节点集文件（XML 导出文件）中。

但导出的属性“AccessLevel”包含“AccessLevelEx”中的信息，请参见下一部分。

导出

对标准 SIMATIC 服务器接口执行 XML 导出时，服务器会将“AccessLevel”属性（与 V1.03 相比，V1.04 中将该属性扩展为 32 位）设为“AccessLevelEx”属性的值。

导入

导入节点集文件时（例如来自服务器接口导出），S7-1500 CPU 会按照其自身对已导入数据类型一致性的估算来设置属性“AccessLevelEx”，请参见下一部分。会忽略导入的值。

服务器接口中数据类型的一致性

对于以下数据类型，会在服务器接口节点处确保 S7-1500 CPU 程序循环中变量的一致性（OPC UA 语言使用中的“原子性”）：

- BOOL、BYTE、WORD、DWORD、LWORD
- SINT、INT、LINT、DINT、USINT、UINT、ULINT、UDINT
- REAL、LREAL
- DATE、LDT、TIME、LTIME、TIME_OF_DAY、LTIME_OF_DAY、S5TIME
- CHAR、WCHAR
- 基于上述数据类型的系统数据类型和硬件数据类型也保持一致。

  示例：HW_ANY，源自 UINT (UInt16)。

提示：如果浏览 S7-1500 CPU 的地址空间（例如使用 OPC UA 客户端 UaExpert），可在“类型 BaseDataType > 枚举/数字/字符串”(Types > BaseDataType > Enumeration/Number/String) 下找到一致的数据类型。

以下数据类型的变量不一致（OPC UA 的语言使用中为“nonatomic”）：

- SIMATIC 结构体通常不一致。这意味着所有变量（例如包含未知结构或 UDT 数据类型）均不一致。
- DTL、IEC_Counter、IEC_TIMER 等系统数据类型是源自结构体的数据类型。
- 字符串（CHAR 型数组）不一致。

提示：如果浏览 S7-1500 CPU 的地址空间（例如使用 OPC UA 客户端 UaExpert），可在“类型 BaseDataType > 结构体”(Types > BaseDataType > Structure) 下找到基于结构体的数据类型。

11.3.2.6 对 S7-1500 Motion Control 中的 OPC UA 变量的写访问。

CPU 除了检查数据类型的一致性之外，还检查工艺对象的变量的合理性和有效性。

如果 OPC UA 客户端将无效的值或不合理的值写入变量，则工艺对象的变量中仍保留原始值。

虽然写入访问没有成功，仍将输出“良好”(Good) 状态。

示例 1

循环凸轮的插补类型

变量 "Cam_1".InterpolationSettings.InterpolationMode 的类型是 INT，但仅接受值 1...2。

如果使用 OPC UA 将变量更改为无效值（例如 3），虽然输出状态代码“Good”，但变量并不会改变。

示例 2

在定位轴上定位软限位开关

正向硬限位开关的位置必须大于负向软限位开关的位置。

"PosAxis_1".PositionLimits_SW.MaxPosition > "PosAxis_1".PositionLimits_SW.MinPosition

如果使用 OPC UA 将变量更改为不满足此条件的值，虽然输出状态代码“Good”，但变量并不会改变。

有关适用于工艺对象变量的有效值，请参见工艺对象文档 (https://support.industry.siemens.com/cs/cn/zh/view/109751049)。

11.3.2.7 访问 OPC UA 服务器数据

符合应用程序的高性能

OPC UA 设计用于在较短的时间内传送大量数据。如果将数组和结构作为一个整体进行读写访问，而非对单个 PLC 变量进行访问，则可显著提高系统性能。

这是最快的访问数组的方式。因此，需将 OPC UA 客户端数据组合到数组中。

关于通过 OPC UA 客户端访问 OPC UA 服务器的建议

- 对于一次性或不频繁的数据访问，请使用标准的读/写访问。
- 对于少量数据的循环访问（循环间隔最长约为 5 秒），请使用订阅。
  优化 OPC UA 服务器中的最短发布时间间隔设置和最小采样时间间隔设置。
- 如果定期访问某些特定变量（重复访问），则可使用函数“RegisteredRead”和“RegisteredWrite”。

通过增加通信循环负载值，可增大 PLC 上的通信负载。确保更改设置后应用程序仍能正常工作。

创建数组 DB 的操作步骤

在全局数据块中或某个函数块的背景数据块中，可创建数组或创建为一个数组 DB。以下章节中，将介绍如何创建一个数组 DB。

要创建带数组的数据块（数组数据块），请按照以下步骤进行操作：

1. 在项目树中选择带 OPC UA 服务器的 CPU。
2. 双击"程序块"(Program blocks)。
3. 双击"添加新块"(Add new block)。
4. 单击"数据块"(Data block)。
5. 为数据块选择一个唯一名称，并接受已输入的名称。
6. 从"类型"(Type) 下拉列表中选择"数组 DB"(Array DB) 条目。
7. 从"数组数据类型"(Array data type) 下拉列表中选择数组各个元素的数据类型。
8. 在"数组限值"(Array limit) 中，输入数组的上限。
9. 单击"确定"(OK)。

11.3.2.8 MinimumSamplingInterval 属性

变量的 MinimumSamplingInterval 属性

除了"Value"、"DataType"和"AccessLevel"之外，在表示服务器地址空间的 XML 文件中还可为变量设置"MinimumSamplingInterval"属性。

该属性用于指定服务器采样变量值的速度。

S7-1500 CPU 的 OPC UA 服务器按以下方式处理 MinimumSamplingInterval 的值：

- 负值和大于 4294967 的值会设为 -1；这表示：最低采样率无法确定。服务器未指定可以对变量值进行采样的速度。
- 小数会舍入到小数点后三位。

11.3.2.9 将 OPC UA 导出为 XML 文件

生成 OPC UA 导出文件

OPC 基金会已经指定了一种基于 XML 的标准格式来描述信息模型。这种格式支持预先将 OPC UA 服务器的信息模型提供给客户端，或者可将信息模型下载到 OPC UA 服务器中。由于这种格式的文件中将信息模型描述为一组节点，因此称为节点集文件。

可通过 STEP 7 (TIA Portal) 轻松将作为服务器的 S7-1500 CPU 的标准 SIMATIC 信息模型导出到 OPC UA XML 文件（节点集文件）；包括为 OPC UA 启用的以下元素：

- CPU 变量（PLC 变量和 DB 元素）
- 函数块及其输入/输出

导出后，OPC UA XML 文件中不包括 CPU 中包含但程序中未使用的元素。此类未使用元素的示例有：

- 未映射到数据块的 UDT
- 具有输入/输出但未将输入/输出分配给 CPU 变量的函数块

可使用 OPC UA XML 文件对 OPC UA 客户端进行离线组态；其结构符合 OPC UA 规范规定，并用作标准 SIMATIC 服务器接口。

要创建和导出 OPC UA XML 文件，请按以下步骤操作：

1. 选择 CPU。单击该 CPU 符号（如，在网络视图中）。
2. 单击 CPU 属性中的"常规 > OPC UA > 服务器 > 导出"(General > OPC UA > Server > Export)。
3. 单击"导出 OPC UA XML 文件"(Export OPC UA XML file)。
4. 选择导出文件的保存目录。
5. 为该文件设置一个新名称，或保留之前输入的原名称。
6. 单击"保存"(Save)。

说明

自 STEP 7 (TIA Portal) V15.1 起，服务器方法与其输入和输出参数共同包含在 OPC UA 导出文件（节点集）中。

单独导出所有数组元素

如果在"OPC UA > 服务器 > 导出"(OPC UA > Server > Export) 下的 CPU 属性中选择了"将所有数组元素作为单独节点导出"(Export all array elements as separate nodes) 选项，则 OPC UA XML 文件包含数组的所有元素，每个元素都作为单独的 XML 元素。此外，数组本身也会在 XML 文件的 XML 元素中分别进行说明。

如果数组包含的数组元素很多，则 XML 文件包含的信息非常多。

提示

在以下常见问题与解答中介绍了一种转换器，可将导出文件转换为 CSV 格式。然后，可获取可通过 OPC UA 访问的 CPU 变量列表。

可在 Internet (https://support.industry.siemens.com/cs/ww/zh/view/109742903) 上找到常见问题与解答。

11.3.3 组态 OPC UA 服务器

11.3.3.1 启用 OPC UA 服务器

要求

- 如果使用安全通信**证书**（如 HTTPS、Secure OUC、OPC UA），请确保相关模块采用**当前时钟和当前日期**。否则，模块将所用的证书评估为无效，且无法进行安全通信。
- 已获得操作 OPC UA 功能的运行系统许可证，请参见“OPC UA 的许可证 (页 246)”。
- 如果启用了访问控制并且正在使用用户管理（本地或中央）：用户必须具备“OPC UA 服务器访问”功能权限。
  有关“OPC UA 服务器访问”功能权限分配的信息，请参见西门子工业在线支持中的以下条目 ID：SIOS 条目 ID 109954947
  (https://support.industry.siemens.com/cs/cn/zh/view/109954947)

调试 OPC UA 服务器

出于安全方面的考虑，默认情况下未启用 CPU 的 OPC UA 服务器：OPC UA 客户端不具备 S7-1500 CPU 的读写访问权限。

要激活 CPU 的 OPC UA 服务器，请按照以下步骤进行操作：

1. 选择 CPU。单击该 CPU 符号（如，在网络视图中）。
2. 单击 CPU 属性中的“OPC UA > 服务器”(OPC UA > Server)。
3. 激活 CPU 的 OPC UA 服务器。
4. 确认安全说明。
5. 转至 CPU 属性，选择“运行系统许可证”(Runtime licenses)，并设置所获得的 OPC UA 服务器的运行系统许可证。
6. 编译项目。
7. 将项目下载到 CPU。

   CPU 的 OPC UA 服务器现在启动。

设置始终存储

如果服务器已启用且进行了相应设置，则再禁用服务器时，设置不会丢失。依旧会保存这些设置，并在再次启用服务器时提供这些设置。

应用程序名称

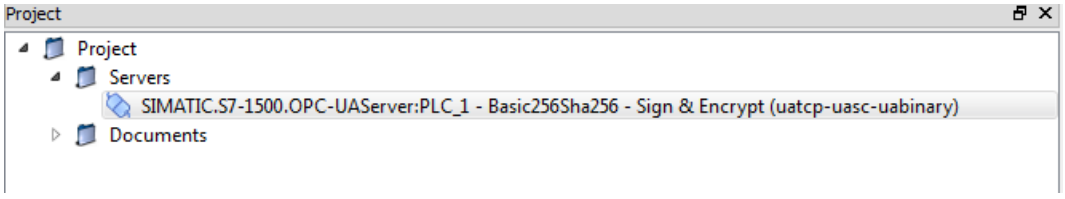
应用程序名称即为 OPC UA 应用程序的名称，会应用于服务器及其客户端。该名称显示在“OPC UA > 常规”(OPC UA > General) 下：

- 应用程序名称的默认设置为：“SIMATIC.S7-1500.OPC-UA.Application:PLC_1”。
- 默认名称由“SIMATIC.S7-1500.OPC-UA.Application:”以及“常规 > 产品信息 >名称”(General > Product information > Name) 中选择的 CPU 名称组成（本示例中为“PLC_1”）。
- OPC UA 服务器将使用该应用程序名称向通信伙伴（OPC UA 客户端）标识自己的身份。例如，OPC UA 客户端使用发现服务检测可访问的服务器时。
- 显示的应用程序名称在连接到 OPC UA 服务器时使用 CPU 的 OPC UA 客户端。这意味着，CPU 自动输入此应用程序名称，作为指令“OPC_UA_Connect”的“ApplicationName”（指令“OPC_UA_Connect”的参数“SessionConnectInfo”中类型为“OPC_UA_SessionConnectInfo”的变量）。

  在编程“OPC_UA_Connect""指令时，需为“ApplicationName”指定一个空字符串。例如，诊断时，可使用该应用名称标识客户端及其会话 (SessionNames)。

如果已激活服务器，则还可使用在项目中有意义的其它名称以及满足项目要求的其它名称（例如，满足全球唯一性要求的名称）。

以下示例源自 UaExpert：



更改应用程序名称

要更改应用程序名称，请按以下步骤操作：

1. 选择 CPU。单击该 CPU 符号（如，在网络视图中）。
2. 单击 CPU 属性中的“OPC UA > 常规”(OPC UA > General)。
3. 输入一个有意义的名称。

请注意，还要在证书上输入应用程序名称（主题备用名称），并且更改应用程序名称后可能需要再次生成现有证书。

11.3.3.2 访问 OPC UA 服务器

服务器地址

可通过 CPU（固件 V2.0 及更高版本）上所有集成的 PROFINET 接口访问 S7-1500 CPU 的 OPC UA 服务器。

在以下条件中，不能借助 CP 通过自动化系统的背板总线直接访问 CPU 的 OPC UA 服务器：

- 使用 TIA Portal 版本 V16 或更高版本、S7-1500 CPU 固件版本 2.8 或更高版本以及 CP 1543-1 固件版本 V2.2 或更高版本进行组态。

  有关组态的信息，请参见"访问 OPC UA 应用程序 (页 160)"。

不能借助 CM 通过自动化系统的背板总线直接访问 CPU 的 OPC UA 服务器。

使用 SIMATIC S7 1500 软件控制器时，可通过分配给软件 PLC 的 PROFINET 接口对 OPC UA 服务器进行访问。

以下应用示例介绍了软件控制器的其它访问选项：通过软件控制器 V2.5 或更高版本的虚拟以太网接口建立的内部和外部 OPC UA 连接 (https://support.industry.siemens.com/cs/ww/zh/view/109760541)。

可用于与 CPU 的 OPC UA 服务器建立连接的 URL (Uniform Resource Locator) 示例：

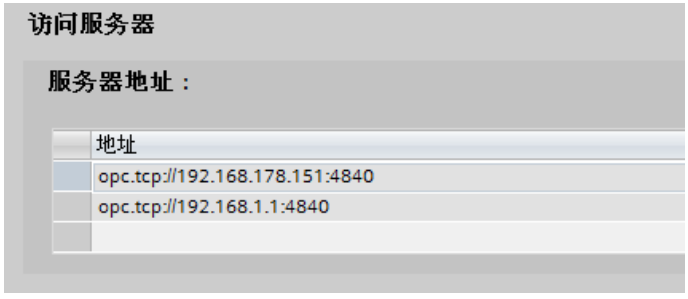


图 11-21 服务器地址的显示

URL 的结构如下所示：

- 协议标识符"opc.tcp://"
- IP 地址
  - 192.168.178.151

    用于从以太网子网 192.168.178 访问 OPC UA 服务器的 IP 地址。
  - 192.168.1.1

    用于从以太网子网 192.168.1 访问 OPC UA 服务器的 IP 地址。
- TCP 端口号
  - 默认值：4840（标准端口）

    可以在"OPC > UA > 服务器 > 端口"(OPC > UA > Server > Port) 下更改端口号。

动态 IP 地址

在以下示例中，未指定 PROFINET 接口 [X2] 的 IP 地址。

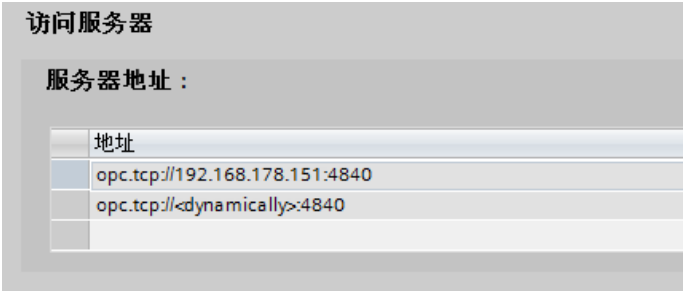


图 11-22 使用动态 IP 地址显示服务器地址

在表格中，将显示占位符“<dynamically>”。

之后，可通过 CPU 显示屏等在设备上设置该 PROFINET 接口的 IP 地址。

激活 SIMATIC 服务器标准接口

如果选择了“启用 SIMATIC 服务器标准接口”(Enable standard SIMATIC server interface) 选项，则 CPU 的 OPC UA 服务器将基于西门子在自定义命名空间中的规定为客户端提供已启用的 PLC 变量和服务器方法。

默认设置中会选择此选项。

保留该选项为选中状态，以便 OPC UA 客户端可自动连接该 CPU 的 OPC UA 服务器并进行数据交换。

如果未选择该选项，则需通过在项目树中输入“OPC UA 通信”(OPC UA communication) 条目，添加服务器接口。之后，该接口将用作 OPC UA 服务器接口，请参见“OPC UA 服务器接口组态 (页 246)”。

说明

即使 SIMATIC 服务器标准接口取消激活，设备常规信息仍可读取

即使禁用 SIMATIC 服务器标准接口，OPC UA 客户端仍可读取该 CPU 中 OPC UA 服务器的常规设备信息。

相关设备信息示例：DeviceManual、DeviceRevision、OrderNumber。但此时，该应用程序的所有对象对客户端均不可见。

如果要保护该设备信息不可见，则需禁用该 CPU 的 OPC UA 服务器。

11.3.3.3 OPC UA 服务器的常规设置

OPC UA 的 TCP 端口

OPC UA 默认使用 TCP 端口 4840。但用户可选用其它端口，此时，可选择 1024 到 49151 的所有端口。此时，需确保与其它应用程序不冲突。OPC UA 客户端在建立连接时必须使用选定的端口。

在以下示例中，选择端口 48400：

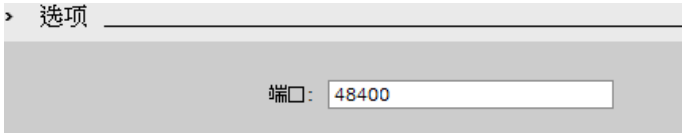


图 11-23 OPC UA 的 TCP 端口

有关 S7-1500 CPU 支持的协议和使用的端口号概述，请参见"以太网通信的通信协议和端口号 (页 33)"部分。

会话设置

- 会话最大超时

  在该字段中指定在不进行数据交换的情况下 OPC UA 服务器关闭会话之前的最大时长。

  允许值在 1 到 600000 秒之间。

- 最大 OPC UA 会话数

  在该字段中指定 OPC UA 服务器启动并同时操作的最大会话数。

  最大会话数取决于 CPU 的性能。每个会话都会占用资源。

最大注册节点数

在该字段中指定 OPC UA 服务器注册的最大节点数。

最大注册节点数取决于 CPU 的容量，并会在组态字段内容时显示（将光标放在字段中）。每次注册都会占用资源。

说明

即使尝试注册的节点数超过所组态的最大注册节点数，也不会出现错误消息

即使客户端在运行期间尝试注册的节点数超过所组态的最大注册节点数，S7-1500 CPU 的服务器也只会注册所组态的最大数量的节点。从所组态的最大可注册节点数开始，服务器会向客户端返回未更改的常规字符串节点 ID，由此这些节点会失去通过注册所获得的速度优势。客户端不会接收到错误消息。

组态时，应考虑可注册的最大节点数（例如，使用 CPU 的技术数据），以确保预留足够的节点。

更多信息

有关进行 TCP 和 UDP 数据传输时各服务所用端口，以及使用路由器和防火墙时的需注意的各事项详细信息，请参见“常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/8970169)”。

根据 OPC UA 规范（V1.03 及以下版本）定义向下兼容数据类型

通过 OPC UA 规范 (<= V1.03) 中定义的相关机制，可通过 TypeDictionaries 从服务器中读取用户自定义结构 (UDT) 的数据类型定义。

在 CPU 的 OPC UA 服务器特性中，可设置 CPU 是否会根据 OPC UA 规范（V1.03 及以下版本）为标准 SIMATIC 服务器接口生成这些向下兼容的数据类型定义。

由于 TypeDictionaries 比较复杂，而且会生成大量需要在客户端进行解译的 OPC UA XML 文件（服务器接口），因此可使用 OPC UA 规范 V1.04（“DataTypeDefinition”属性）中的一个较为简单的解决方案。如果客户端支持 OPC UA 规范（V1.04 或更高版本），请禁用此选项。

根据 OPC UA 规范（V1.04 及更高版本）定义数据类型的优势：

- 服务器启动更快
- 内存利用率更高
- “浏览”(Browse) 功能的速度更快

11.3.3.4 服务器的订阅设置

使用订阅替代循环查询

通过对 PLC 变量进行值监视，也可实现循环查询（轮询）。使用 Subscription：如果 PLC 变量的值发生变化，服务器将通知客户端。

一台服务器通常监控大量的 PLC 值。因此，服务器定期向客户端发送包含 PLC 变量新值的通知。

订阅的优势：

- 服务器启动更快
- 内存利用率更高

服务器发送通知的频率

创建 Subscription 时，OPC UA 客户端可设定变量值发生变化时，新值发送的时间间隔。要限制 OPC UA 的通信载荷，可设置消息的最短时间间隔。为此，可使用最短发布时间间隔参数和最短采样时间间隔的参数。

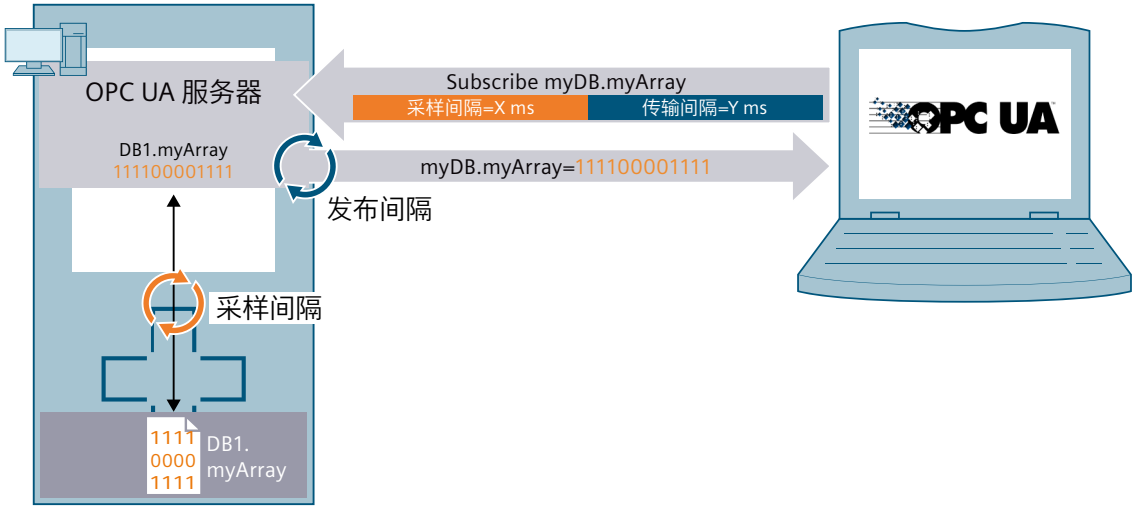


图 11-24 订阅原理

最短发布时间间隔

在“最短发布时间间隔”(Minimum publishing interval) 中，可设置变量值发生改变时服务器通过新值向客户端发送消息的时间间隔。

在下图中“最短采样时间间隔”为 250 ms。输入 200 ms 作为“最短发布时间间隔”。

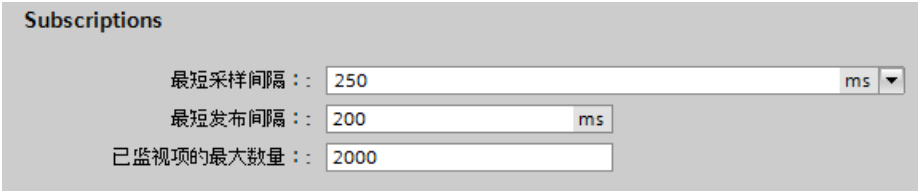


图 11-25 订阅设置

在本示例中，数值更改后，如果 OPC UA 客户端请求更新，则 OPC UA 服务器将按照 200 ms 的时间间隔发送新消息。

如果 OPC UA 客户端要求的更新频率为 1000 ms，则 OPC UA 服务器每隔 1000 ms（1 秒）仅发送一条带有新值的消息。

如果 OPC UA 客户端要求的更新频率为 100 ms，则服务器每隔 200 ms 也只发送一条消息（最短发布时间间隔）。

最短采样时间间隔

在“最短采样时间间隔”(Minimum sampling interval) 中，可设置 OPC UA 服务器记录 CPU 变量值并与以前值相比较检查是否发生变更的时间间隔。

如果所选择的采样时间间隔小于发布时间间隔，且 OPC UA 客户端请求对特定 PLC 变量进行高速采样，则在每个发布时间间隔内将测量两个或更多变量值。

此时，OPC UA 服务器将值变更写入队列中，并在发布间隔时间结束后，将所有值更改发送到客户端。在发布间隔时间内，如果发生的值更改次数过多且超出队列容量，则 OPC UA 服务器将覆盖最旧的值（具体取决于订阅数据的客户端的设置“Discard Policy”，此时，需激活选项“Discard Oldest”）。最新值将发送到客户端。

所监视元素（所监视条目）的最大数量

在该字段中，指定该 CPU 的 OPC UA 服务器可同时监视值更改的最大元素数量。

监视会占用资源。可监视元素的最大数量取决于所用的 CPU。

更多信息

有关 S7-1500 CPU（固件 V2.0 和 V2.1）中 OPC UA 服务器有关订阅、采样间隔和发布间隔等的系统限值信息，请参见“常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/109755846)”。

使用订阅时，可通过某些错误状态代码确定该错误的具体信息。有关 OPC UA 客户端各状态代码的原因及补救措施等信息，请参见 STEP 7 (TIA Portal) 在线帮助中的错误代码列表或以下“常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/109755860)”。

订阅规则参见“订阅规则 (页 348)”部分。

有关订阅诊断的信息，请参见“订阅诊断 (页 313)”部分。

11.3.3.5 使用 TransferSubscription 服务

有关转移订阅的有用信息

OPC UA 规范规定，OPC UA 客户端可以将订阅及其监控项从一个会话转移到另一个会话（请参见 OPC 10000-4：UA 第 4 部分：服务 – 特别是“TransferSubscriptions”部分）。

S7-1500 CPU 的 OPC UA 服务器从固件版本 V3.1.4 开始支持订阅转移。

应用示例：

- 活动会话中断/超时：例如由于通信中断：订阅从中断的会话转移到活动会话以避免数据丢失。
- 一个客户端即将关闭；另一个客户端接管订阅。
- 负荷分配：订阅从一个活动会话转移到另一个活动会话或一个新会话。

订阅转移的好处：

- 订阅的转移可以防止数据丢失（在客户端发生故障、断开连接或过载的情况下）。

  转移订阅服务可以转移订阅的数据。利用此服务，可以按照下列方式解决出现的连接终止问题：

  如果接管订阅的客户端与服务器之间已经建立连接，则无需重新创建包含所有监控项的订阅。订阅转移成功后，可以重新发布累积的（即未发送的）通知消息。另请参见 OPC 10000-4：UA 第 4 部分：服务，“重新建立连接”部分。
- 通过订阅转移，OPC UA 客户端避免花费大量时间和资源来创建订阅。

  当创建包含大量监控项的订阅时，会产生较高的 CPU 负载。

要求

不需要组态 S7-1500 OPC UA 服务器；“TransferSubscription”服务是根据 OPC UA 客户端的请求提供的。

自固件版本 V3.1.4 起的所有 S7-1500 CPU 均支持订阅转移。

其它要求：

- 用户（客户端）必须具备授予其Read permission的角色。
- 如果要转移已关闭会话的订阅，则必须在订阅超时之前完成此过程。
- 订阅可以在经过身份验证的用户的会话之间转移，也可以在匿名用户的会话之间转移。
  - 对于经过身份验证的用户，两个会话必须分配给同一个用户。
  - 对于匿名用户，连接还必须加密，并且会话必须由具有相同应用程序 URI 的客户端创建。

11.3.3.6 处理客户端和服务器证书

仅当 OPC UA 服务器可向 OPC UA 客户端证明身份时，才能建立服务器与客户端之间的安全连接。服务器证书可用于证实身份。

OPC UA 服务器的证书

激活 OPC UA 服务器并确认安全提示后，STEP 7 会自动为服务器生成证书，并将其保存在 CPU 的局部证书目录中。可以使用 CPU 的局部证书管理器查看并管理此目录（导出或删除证书）。

下图所示为包含 OPC UA 服务器自动生成的证书的 CPU 局部证书管理器：

图 11-26 CPU 的本地证书管理器

或者，您还可以自行生成服务器证书。

在建立连接时，服务器证书将从服务器传送到客户端。客户端将检查该证书。

客户端用户将确定是否信任该服务器证书。

此时，客户端用户需确定是否信任该服务器证书。如果信任该服务器证书，则客户端将服务器证书存储在包含可信服务器证书的目录中。

在以下示例中，显示了客户端 "UA Sample Client" 的对话框。如果用户单击“是”(Yes) 按钮，则客户端将信任此服务器证书：

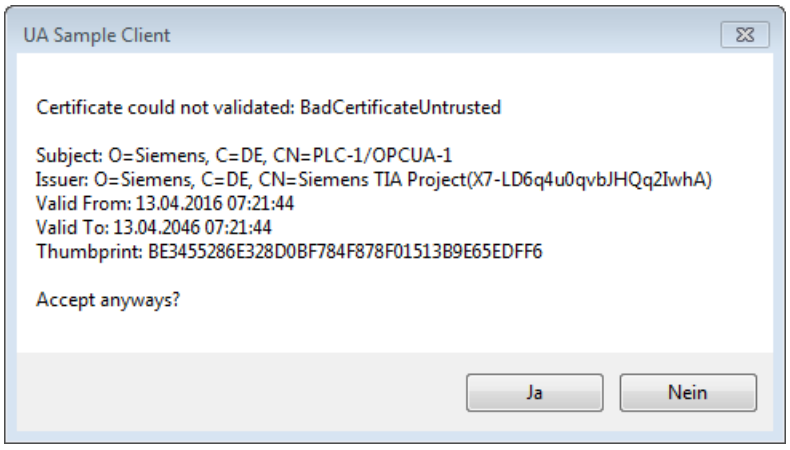


图 11-27 “UA Sample Client”客户端的对话框

客户端证书来自哪里？

S7-1500 的客户端

如果使用 S7-1500 CPU 的 OPC UA 客户端（已启用 OPC UA 客户端），可使用 STEP 7 V15 及更高版本为这些客户端创建证书。

1. 在项目树中，选择将用作客户端的 CPU。
2. 双击“设备组态”(Device configuration)。
3. 在该 CPU 的属性中，单击“保护和安全 > 证书管理器”(Protection & Security > Certificate manager)。
4. 在“设备证书”(Device certificates) 表格中，双击“<新增>”(<Add new>)。

   在 STEP 7 中，将打开一个对话框。
5. 单击“添加”(Add) 按钮。
6. 从“使用”(Usage) 列表选择“OPC UA 客户端”(OPC UA client) 条目。

   注：

   必须在“主题备用名称 (SAN)”(Subject Alternative Name (SAN)) 中输入用于访问系统中 CPU 的 IP 地址。

   因此，在生成客户端证书之前，需要对 CPU 的 IP 接口进行组态。
7. 单击“确定”(OK)。

   此时，STEP 7 将在“设备证书”(Device certificates) 表格中显示该客户端证书。
8. 右键单击该行，并在快捷菜单中选择“导出证书”(Export certificate) 条目。
9. 选择该客户端证书的目标存储目录。

其它制造商的客户端

如果使用来自制造商或 OPC 基金会的 UA 客户端，则会在安装期间或在首次调用程序时自动生成客户端证书。需要在 STEP 7 中通过全局证书管理器导入这些证书，并将其用于相应的 CPU（如前文所示）。

用户自己编程 OPC UA 客户端时，可生成相应的证书；请参见"客户端的实例证书"部分。也可通过工具生成证书（如，使用 OpenSSL 或 OPC 基金会的证书生成器）：

- 使用 OpenSSL 时的操作步骤："用户自己生成 PKI 密钥对和证书"。
- 使用 OPC 基金会的证书生成器时："创建自签名的证书"。

向服务器宣布客户端证书

您需要向服务器发送客户端证书，以允许建立安全连接。

为此，请执行以下操作步骤：

1. 在服务器的本地证书管理器中，选择"使用证书管理器的全局安全设置"(Use global security settings for certificate manager) 选项。这将激活全局证书管理器。

   可以在用作服务器的 CPU 的特性"保护和安全 > 证书管理器"(Protection & Security > Certificate manager) 下找到此选项。

   如果项目未受保护，请在 STEP 7 的项目树中选择"安全设置 > 设置"(Security settings > Settings)，然后单击"保护此项目"(Protect this project) 按钮并登录。

   "全局安全设置"(Global security settings) 菜单项随即显示在 STEP 7 项目树的"安全设置"(Security setting) 下。
2. 双击"全局安全设置"(Global security settings)。
3. 双击"证书管理器"(Certificate manager)。

   STEP 7 将打开全局证书管理器。
4. 单击"受信任证书"(Trusted certificates) 选项卡。
5. 在此选项卡的空白区域（而非证书上）中，右键单击鼠标。
6. 选择快捷菜单中的"导入"(Import) 命令。

   将显示用于导入证书的对话框。
7. 选择服务器信任的客户端证书。
8. 单击"打开"(Open)，导入证书。

   客户端证书现已包含在全局证书管理器中。

   请留意刚刚导入的客户端证书 ID。
9. 单击用作服务器的 CPU 的特性中的"常规"(General) 选项卡。
10. 单击"OPC UA > 服务器 > 安全 > 安全通道"(OPC UA > Server > Security > Secure Channel)。
11. 在"安全通道"(Secure Channel) 对话框中向下滚动至"受信客户端"(Trusted clients) 部分。
12. 双击表中空行的"<新增>"(<add new>)。随即会在该行中显示浏览按钮。
13. 单击该按钮。
14. 选择已导入的客户端证书。

15. 单击带有绿色复选标记的按钮。

16. 编译项目。

17. 将组态加载到 S7-1500 CPU。

结果：

服务器现已信任此客户端。如果还将服务器证书视为受信证书，则服务器和客户端之间可建立安全连接。

自动接受客户端证书

如果选择选项"运行时自动接受所有客户端证书"(Automatically accept all client certificates during runtime)（位于"受信客户端"(Trusted clients) 列表下），则服务器会自动接受所有客户端证书。

注意
调试后的设置 为了避免安全风险，在调试后，需再次取消选中"运行过程中自动接受客户端证书"(Automatically accept client certificates during runtime) 选项。

组态服务器的安全设置

下图显示了适合对消息进行签名和加密的服务器安全设置。

图 11-28 组态服务器的安全设置

默认情况下，服务器证书创建时使用 SHA256 签名。并启用以下安全策略：

- 无
  不安全端点

  说明

  禁用不需要的安全策略

  如果在 S7-1500 OPC UA 服务器的安全通道设置中启用了所有安全策略（默认设置），即采用端点"无"(None)（不安全），则服务器和客户端之间还可能存在非安全数据通信（既未签名也未加密）。由于选择"不安全"(No security)，客户端的身份仍然未知。无论后续为哪种安全设置，每个 OPC UA 客户端随后都可以连接到服务器。

  组态 OPC UA 服务器时，请确保只选择与您的设备或工厂的安全概念兼容的安全策略。应禁用所有其它安全策略。

  建议：如果可能，请使用"Basic256Sha256"设置。

- Basic128Rsa15 - 签名
  不安全端点，支持一系列使用哈希算法 RSA15 和 128 位加密的算法。
  该端点通过签名确保数据的完整性。
- Basic128Rsa15 - 签名和加密
  安全端点，支持一系列使用哈希算法 RSA15 和 128 位加密的算法。
  该端点通过签名和加密确保数据的完整性。
- Basic256Rsa15 - 签名
  安全端点，支持一系列使用哈希算法 RSA15 和 256 位加密的算法。
  该端点通过签名确保数据的完整性。
- Basic256Rsa15 - 签名和加密
  安全端点，支持一系列使用哈希算法 RSA15 和 256 位加密的算法。
  该端点通过签名和加密确保数据的完整性。
- Basic256Sha256 - 签名
  端点进行安全连接，支持一系列 256 位哈希和 256 位加密算法。
  该端点通过签名确保数据的完整性。
- Basic256Sha256 - 签名和加密
  安全端点，支持一系列 256 位哈希和 256 位加密算法。
  该端点将通过签名与加密机制确保数据的完整性和保密性。
- Aes256_Sha256_RsaPss - 签名

  端点进行安全连接，支持一系列 256 位加密和 256 位哈希算法。所有证书必须至少使用 Sha256 签名。该端点通过签名来保护数据的完整性。
  对于较高的安全性要求。需要 PKI 基础结构。
- Aes256_Sha256_RsaPss - 签名和加密

  端点进行安全连接，支持一系列 256 位加密和 256 位哈希算法。所有证书必须至少使用 Sha256 签名。该端点通过签名和加密来保护数据的完整性和机密性。
  对于较高的安全性要求。需要 PKI 基础结构。

要启用安全设置，请单击相关行的复选框。

说明

如果设置为“Basic256Sha256 - 签名”(Basic256Sha256 -Sign) 和“Basic256Sha256 - 签名并加密”(Basic256Sha256 -Sign & Encrypt)，则 OPC UA 服务器和 OPC UA 客户端必须使用“SHA256”签名的证书。

对于“Basic256Sha256-签名”(Basic256Sha256 -Sign) 和“Basic256Sha256-签名并加密”(Basic256Sha256 -Sign & Encrypt) 设置，STEP 7 中的证书颁发机构将使用“SHA256”自动对证书进行签名。

“不安全”安全策略和通过用户名和密码进行身份验证

可执行以下组合设置：

“不安全”安全策略和通过用户名和密码进行身份验证

- S7-1500 的 OPC UA 服务器支持该组合设置。OPC UA 客户端可连接并加密认证数据，反之亦然。
- S7-1500 CPU 的 OPC UA 客户端也支持该组合设置：但在运行时，仅当通过电缆发送加密的认证数据时才能连接！

11.3.3.7 使用 STEP 7 生成服务器证书

在下文中，将介绍使用 STEP 7
生成新证书的操作过程，以及各种证书的不同应用方式。STEP 7 将基于启动以下对话框时的 CPU 属性区域，设置应用目标。在本示例中，为“OPC UA 客户端和服务器”(OPC UA Client & Server)。

建议：要使用 OPC UA 服务器的所有安全功能，则需使用全局安全设置。

在 CPU 特性的“保护和安全 > 证书管理器”(Protection & Security > Certificate manager) 下启用全局安全设置。

用户自定义的服务器证书

如果您激活 S7-1500 的 OPC UA 服务器，则 STEP 7 会自动为该服务器生成证书（请参见"激活 OPC UA 服务器 (页 223)"）。在该过程中，STEP 7 使用证书参数的默认值。如果要更改参数，请按照以下步骤进行操作：

1. 单击 CPU 属性中"常规 > OPC UA > 服务器 > 安全 > 安全通道 > 服务器证书"(General > OPC UA > Server > Security > Secure channel > Server certificate) 下的"浏览"(Browse) 按钮。随即会显示一个对话框，用于显示局部可用的证书。
2. 单击"添加"(Add) 按钮。
3. 将显示用于生成新证书的对话框（如下图所示）。已输入示例的值：

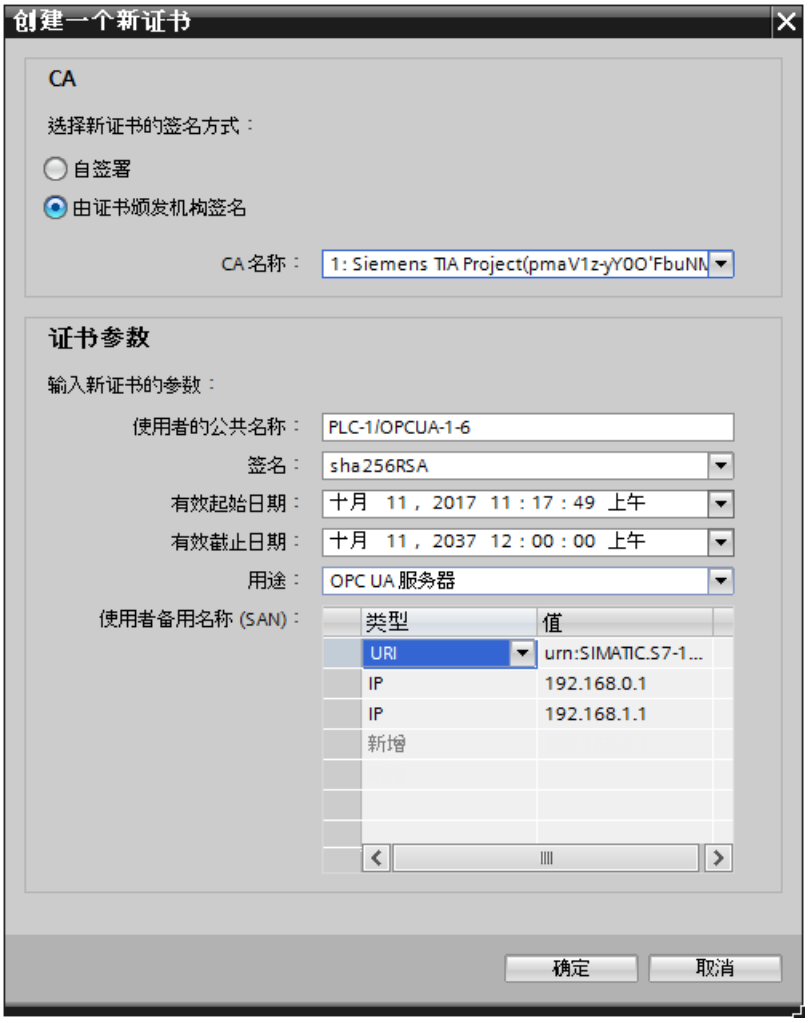


图 11-29 用户自定义的服务器证书

4. 必要时，可根据公司或客户的安全规范使用其它参数。

用于生成证书的字段的说明

- CA

  选择证书是自签名，还是由 TIA Portal 的一个 CA 证书进行签名。有关这些证书，请参见“带 OPC UA 的证书”部分。如果要生成由 TIA-Portal 的一个 CA 证书签名的证书，项目必须受保护，而且您必须以具有全部所需功能权限的用户身份登录。更多相关信息，请参见“TIA Portal 中用户管理的基本知识”。

- 证书持有者

  在默认设置中，通常包括项目名称和“\OPCUA-1”。在本示例中，项目名称为“PLC1”。在 CPU 属性的“常规 > 项目信息 > 名称”("General > Project information > Name) 下设置项目名称。保留默认设置，或者在“证书持有者”(Certificate holder) 下为 OPC-UA 服务器输入其它更有意义的名称。

- 签名

  在此处选择对服务器证书进行签名时要使用的哈希和加密过程。下列条目可用：

  – “sha1RSA”、

  – “sha256RSA”。

- 生效日期

  在此处输入服务器证书开始生效的日期和时间。

- 截止日期

  在此处输入服务器证书有效终止的日期和时间。确保证书的有效期不仅为一年或几年。在本示例中，证书的有效期为 30 年。不过，出于安全方面的考虑，应该以更短的时间间隔更新证书。如果有效期较长，您便有机会决定何时为对系统执行保养等作业的合适时机。

- 用途

  默认设置为“OPC UA 客户端和服务器”(OPC UA client & server)。保留 OPC UA 服务器的默认设置。在 STEP 7 中，可从多个位置调用“创建新证书”(Create a new certificate) 对话框。例如，如果在 CPU 的 Web 服务器中调用此对话框，则需在“使用”(Usage) 下输入“Web 服务器”(Web server)。“用途”(Usage) 下拉列表中包含以下条目：

  – “OPC UA 客户端”(OPC UA client)

  – “OPC UA 客户端和服务器”(OPC UA client & server)

  – “OPC UA 服务器”(OPC UA server)

  – “TLS”

  – “Web 服务器”(Web server)

- 主题备用名称 (SAN)

  在上述示例中输入以下内容：“URI:urn:SIMATIC.S7-1500.OPC-UAServer:PLC1,IP:192.168.178.151,IP:192.168.1.1”。必须正确输入此 URI，因为将根据所传达的应用程序描述对其进行检查。

  以下条目也将有效：“IP：192.168.178.151，IP：192.168.1.1”。注意，在此处输入可用于访问 CPU 的 OPC UA 服务器的 IP 地址。

  请参见“访问 OPC UA 服务器 (页 225)”。

  借此，OPC UA 客户端可验证是否要与 S7-1500 的 OPC UA 服务器真正建立连接，或验证实际上是否攻击者在尝试将另一台 PC 的篡改值发送至 OPC UA 客户端。

11.3.3.8 用户认证

用户认证方式

对于 S7-1500 的 OPC UA 服务器，可设置 OPC UA 客户端中用户访问服务器时需通过的认证。

可通过以下几种方式：

- **访客认证**

  用户无需证明其身份（匿名访问）。OPC UA 服务器不会检查客户端用户的授权。

  对于固件版本不超过 V3.0 的 CPU：如需使用这种认证方式，则可在"OPC UA > 服务器 > 安全 > 用户认证"(OPC UA > Server > Security > User authentication) 中选择"启用访客认证"(Enable guest authentication) 选项。

  对于固件版本为 V3.1 及以上版本的 CPU：使用本地用户管理通过"匿名"用户执行访客身份验证。

  说明

  为增加安全性，应只允许访问支持用户认证的 OPC UA 服务器。

- **用户名和密码认证**

  用户必须证明其身份（非匿名访问）。OPC UA 服务器将检查客户端用户是否具备访问服务器的权限。通过用户名和正确的密码进行身份验证。

  对于固件版本不超过 V3.0 的 CPU：如果要使用此类型的用户身份验证，请按以下步骤进行操作：

  – 在"OPC UA > 服务器 > 安全 > 用户身份验证"(OPC UA > Server > Security > User authentication) 中选择"启用用户名和密码身份验证"(Enable user name and password authentication) 选项。
  – 取消激活访客认证。
  – 在"用户管理"(User management) 表中输入用户。

    此时，可单击条目"<新增用户>"(<Add new user>)。系统将会创建一个新的用户并自动命名。用户可对该用户名进行编辑并输入密码。最多可添加 21 个用户。

- **通过项目的安全设置进行额外的用户管理**

  对于固件版本高达 V3.0 的 CPU，"通过项目安全设置，启用附加用户管理"(Enable additional user management via project security settings) 选项可用。可以在以下常规 OPC UA 设置下找到此设置：（CPU 属性："OPC UA > 常规"(OPC UA > General)）下。如果选择此选项，打开项目的用户管理也会用于对 OPC UA 服务器进行用户认证：随后，当前项目中的相同用户名和密码同样在 OPC UA 中生效。

  要激活项目的用户管理，请按以下步骤操作：

  – 在项目树中单击"安全设置 > 设置"(Security settings > Settings)。
  – 单击"保护此项目"(Protect this project) 按钮。
  – 输入用户名和密码。
  – 在"安全设置 > 用户和角色"(Security settings > Users and roles) 下输入其它用户。

  如果组态项目中的其它 OPC UA 服务器，还应选择"通过项目的安全设置启用额外用户管理"(Enable additional user administration via the security settings of the project) 选项。随后不需要重复输入用户名和密码。

通过安全通道传送用户身份验证数据

在 OPC UA 中，用户身份验证的标识数据（如，用户名和密码）使用单独的安全策略进行传输。该安全策略称为"UserTokenPolicy"策略。

如果需要进行用户身份验证，则 OPC UA 客户端将在连接的建立过程中选择一个合适的"UserTokenPolicy"，而与安全通道所组态的安全策略无关。此 UserTokenPolicy 可确保 UserIdentityToken（如，用户名和密码）始终以适当的安全设置进行传输。

如果安全通道根据所组态的安全策略采用"无安全设置"，则 OPC UA 客户端之后可通过加密方式传送用户名和密码。有关 OPC UA 中建立安全连接的过程信息，请参见"消息的安全传送 (页 179)"。

11.3.3.9 具有 OPC UA 功能权限的用户和角色

用户认证的以下选项使用集中项目设置（针对项目用户）：

- 针对服务器：

  用于组态 CPU 特性（"OPC UA > 服务器 > 安全 > 用户认证"(OPC UA > Server > Security > User authentication)）。选项："通过项目的安全设置启用额外用户管理"(Enable additional user administration via the security settings of the project)

- 针对客户端：

  用于组态客户端接口（"安全"(Security) 下的"组态"(Configuration) 选项卡）。选项："用户（TIA Portal - 安全设置）"(User (TIA Portal - security settings))

要求

在编辑安全设置之前，项目必须受保护，且您必须以具有足够权限的身份（例如作为管理员）登录。

项目树中"安全设置"下的设置

在项目树的"安全设置"(Security setting) 下访问受保护项目中的集中用户设置和角色。在这里集中定义包含用户名、密码和功能权限的用户。可以在其它位置直接使用这些设置。

图 11-30 设置用户和角色

重用集中安全设置

在其它位置进行重用的示例：

- 针对 OPC UA 服务器用户管理的用户选择

  借助此设置，可以通知服务器具有哪个用户名和哪个密码的哪个客户端（用户）可以访问服务器。

- 针对 OPC UA 客户端认证的用户选择

  借助此设置，可以通知客户端用于服务器客户端认证的用户名和密码。

客户端和服务器的设置必须对应：客户端登录所使用的用户名和密码必须已在服务器上设置，并被分配所需权限。

服务器和客户端的功能权限

还必须为 S7-1500 CPU 上客户端功能的用户和服务器功能的用户启用相应的客户端或服务器功能权限。仅集中保存用户名和密码远远不够。

以下为此类型权限使用的说明示例：

1. 例如，在"安全设置 > 用户和角色"(Security settings > Users and roles) 下的"角色"(Roles) 选项卡中定义新角色（名称为"PLC-opcua-role-all-inclusive"）。
   提示：选项卡可能被信息窗口"尚未检查当前状态..."(The current status has not yet been checked...) 覆盖。在这种情况下，请先关闭信息窗口。
2. 在"功能权限类别"(Function rights categories) 部分中，导航到运行系统权限，然后导航到 CPU 功能权限，选择要设置其功能权限的 CPU。
3. 可在"功能权限"(Function rights) 部分中找到以下功能权限：
   - **OPC UA 服务器访问**

     此功能权限适用于 S7-1500 CPU 的 OPC UA 服务器。只有选择此选项时，角色为"PLC-opcua-role-all-inclusive"的用户才可在运行时将证书、CRL 或受信任列表传送到 CPU（推送功能）。自动化证书处理需要用到此功能权限，例如在 GDS（全球发现服务）上下文中。
   - **管理证书**

     此功能权限适用于 S7-1500 CPU 的 OPC UA 服务器。只有启用此选项时，角色为"PLC-opcua-role-all-inclusive"的用户才可在运行时将证书、CRL 或受信任列表传送到 CPU（推送功能）。自动化证书处理需要用到此功能权限，例如在 GDS（全球发现服务）上下文中。

– **OPC UA 客户端的用户认证**

此功能权限适用于 S7-1500 CPU 的 OPC UA 客户端（具有客户端指令）。只有选择此选项时，角色为“PLC-opcua-role-all-inclusive”的用户才能使用用户名和密码进行身份验证，以与服务器建立会话。

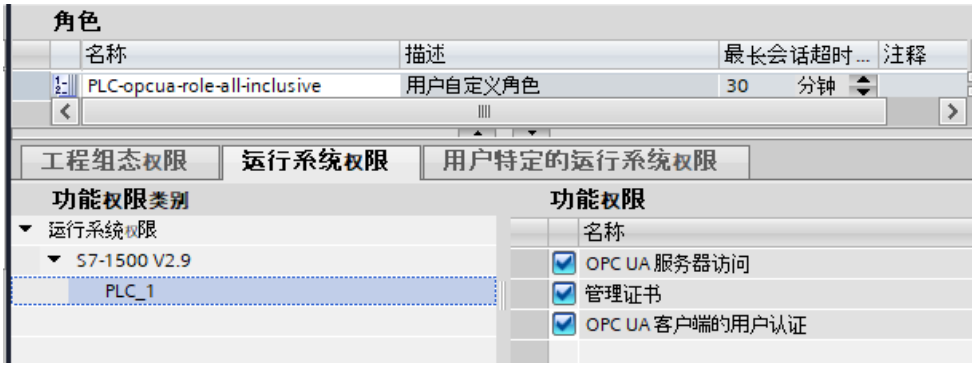


图 11-31 设置功能权限

4. 仍需将“PLC-opcua-role-all-inclusive”角色分配给相关用户（项目树中“安全设置”(Security settings) 下的“用户”(Users) 选项卡）。

说明

具有 OPC UA 功能权限用户的“运行系统超时时间”

用户组态表中，“运行系统超时时间”(Runtime timeout) 列中的值（会话最长持续时间）不对 CPU 的 OPC UA 运行系统权限进行评估。

因此，用户不会在特定时间过后就自动注销。为此使用 OPC UA 特有的机制，例如参数“最大会话超时”(Max. session timeout)（“OPC UA > 服务器 > 设置”(OPC UA > Server > Settings) 区域）。

11.3.3.10 服务器的诊断设置

诊断

可以在 CPU 设置中指定 OPC UA 服务器的诊断范围。

要更改诊断范围，请导航至“OPC UA > 服务器 > 诊断”(OPC UA > Server > Diagnostics) 区域。

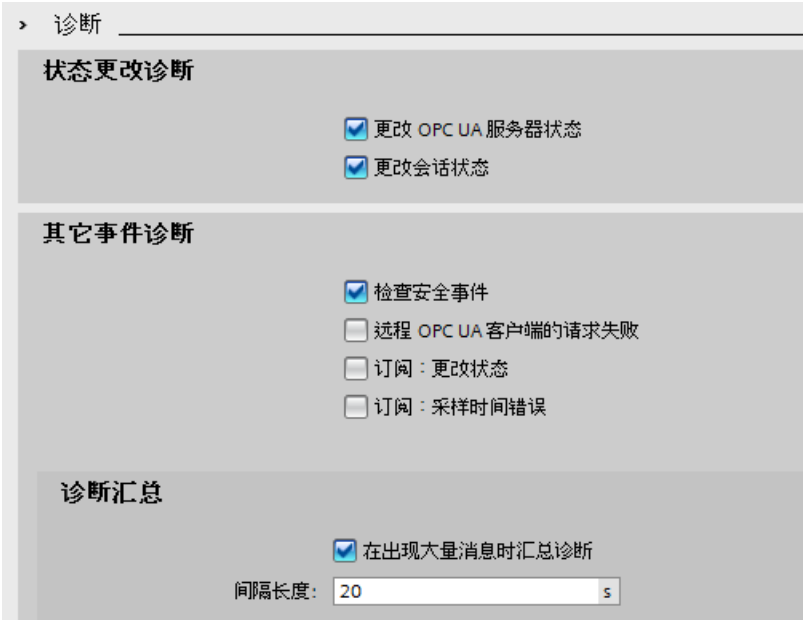


图 11-32 OPC UA 服务器的诊断设置

默认设置

默认设置是一种诊断行为，这种行为支持最重要的诊断，而不会明显增加通信负载。

当 OPC UA 服务器也使用订阅时（仅在调试阶段有必要使用）启用对订阅的诊断。

原因：大量诊断活动会在 CPU 中产生较高的通信负载，并可能抑制其它重要消息。或者，诊断量过大可能导致重要消息在大量消息中消失或被忽略。

更多信息

有关上述设置的含义和作用的更多信息，请参见此处 (页 307)。

11.3.3.11 OPC UA 的许可证

运行系统许可证

运行 S7-1500 CPU 的 OPC UA 服务器需要使用许可证。所需的许可证类型取决于相应 CPU 的性能。将许可证类型分为以下几类：

- SIMATIC OPC UA S7-1500 小型（CPU 1511、CPU 1512、CPU 1513、ET 200SP CPU、CPU 1515SP PC 需要使用这种类型）
- SIMATIC OPC UA S7-1500 中型（适用于 CPU 1515、CPU 1516、软件控制器 CPU 1507、CPU 1516pro-2PN）
- SIMATIC OPC UA S7-1500 大型（CPU 1517、CPU 1518 需要使用这种类型）

所需许可证类型显示在"属性 > 常规 > 运行许可证 > OPC-UA > 所需许可证类型"(Properties > General > Runtime licenses > OPC-UA > Type of required license) 下：

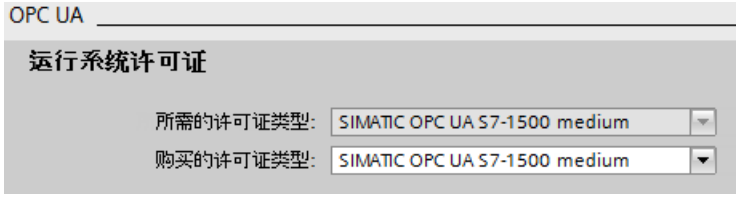


图 11-33 OPC UA 服务器运行系统许可证

若要确认购买所需许可证，请按照以下步骤进行操作：

1. 单击 CPU 属性中的"运行许可证 > OPC UA"(Runtime licenses > OPC UA)。
2. 在"购买的许可证类型"(Type of purchased license) 下拉列表中，选择所需的许可证。

11.3.4 OPC UA 服务器接口组态

11.3.4.1 什么是服务器接口？

定义

服务器接口将 CPU 的 OPC UA 地址空间的节点合并到一个单元中，以便为 OPC UA 客户端提供该 CPU 的特定视图。

每个服务器接口都会在 CPU 的 OPC UA 服务器中定义一个或多个命名空间。

STEP 7 (TIA Portal) 根据以下服务器接口类型加以区分：

- 配套规范

  例如，对于此类服务器接口，可以使用工作组创建的配套规范。

  工作组通常由 OPC 基金会成员以及其它共同制定特定用途（例如与 RFID 设备或注塑机进行数据交换）的 OPC UA 信息模型的工业组织组成。

  该信息模型在 OPC UA 非的地址空间中以 OPC UA 节点的形式实现。OPC UA 客户端可访问这些 OPC UA 节点。

  例如，还可以使用服务器接口类型"配套规范"在 SiOME 中下载公司内部信息模型。

  如果在项目中实施某一配套规范，则会将该配套规范的具体规范作为服务器接口应用到项目中。

对于"配套规范"类型的服务器接口，可以导入配套规范使用的多个命名空间。

有关配套规范的更多信息，请参见此处 (页 248)。

有关 SiOME 的更多信息，请参见此处 (https://support.industry.siemens.com/cs/cn/zh/view/109755133)。

- 当配套规范引用从属规范中的类型定义时，将引用命名空间用于此目的。导入引用命名空间，如同实际的配套规范一样。

  请参见"为配套规范创建服务器接口 (页 254)"。

- 自 TIA Portal V17 起，如果希望 OPC UA 客户端可以访问该 CPU 中 FB 或 UDT 内的实例数据，可自动分配这些实例数据。用户只需将 FB 类型或 UDT 映射到已导入的引用命名空间的适用 OPC UA 数据类型。为了实现此映射，请在对话框中启用选项"基于本地数据映射生成 OPC UA 节点"(Generate OPC UA nodes based on the local data mapping)，以创建配套规范/引用命名空间类型的 OPC UA 服务器接口。

  请参见"基于 FB 类型和 UDT 的本地数据映射生成 OPC UA 节点 (页 276)"

- 关于用户自定义服务器接口：

  对于这种类型的服务器接口，会将 OPC UA 服务器的 OPC UA 节点合并到一个单元中。

  为此，请使用项目规范、机器或设备要求作为基础。

  有关用户自定义服务器接口的更多信息，请参见此处 (页 258)。

以注塑机作为配套规范的示例

本例中，服务器接口包含以下元素：

- OPC UA 节点，可通过 OPC UA 客户端读取该元素，以接收关于该注塑机的信息（可读 PLC 变量中）
- OPC UA 节点，可通过 OPC UA 客户端写入该元素，以将数值传送到注塑机（可写 PLC 变量中）
- OPC UA 节点，可通过 OPC UA 客户端调用该元素，以启动注塑机功能（通过服务器方法）。

该服务器接口会启用可用于控制注塑机的 CPU 默认视图。

对于注塑机，配套规范"OPC UA specifications for plastics and rubber machines"（之前为"Euromap"）定义了可用作服务器接口中的整个系列的 OPC UA 节点。

CPU 的其它 OPC UA 节点不包含在此服务器接口中。这样可以更好地提供概览。

用户自定义服务器接口示例

CPU 应控制工件的生产。当生产作业从更高级的控制系统到达时，会开始生产。

生产作业通过服务器方法传送：控制系统通过调用 CPU 中的服务器方法将信息传送到工件上。该服务器方法也会启动生产。

控制系统（即连接的 OPC UA 客户端）应当只能看到这一种服务器方法。因此，应在 CPU 中创建一个用户自定义服务器接口并将服务器方法分配给该服务器接口。仅可为 OPC UA 客户端启用该服务器接口，因此 CPU 视图仅限于这一种功能。

11.3.4.2 使用 OPC UA 配套规范

简介

OPC UA 普遍适用：例如，标准本身不指定 PLC 变量的命名方式。由个人用户（应用程序开发人员）编写和命名可通过 OPC UA 调用的服务器方法。

针对设备和部门的信息建模和标准化

对于同类应用，应使用"OPC UA 工具包"来标准化设备或机器接口。

许多不同的机构和工作组已经推动标准化，并制定了一系列配套规范。

这些规范定义了：

- 用于描述典型设备或机器的对象、方法和变量。
- 用于指定对象的命名空间。

机器通常由功能或技术单元构成，然后对这些单元进行标准化。

配套规范为机器和工厂操作员提供了标准化接口的优势。例如，符合 AutoID 规范的所有 RFID 阅读器均可采用相同的方式集成。这意味着，无论制造商如何，符合 AutoID 规范的所有 RFID 阅读器均可由 OPC UA 客户端以相同方式寻址。

配套规范的另一个示例是，注塑机械部门的 Euromap 77 配套规范。

以下部分以 Euromap 77 为例详细介绍了如何在 STEP 7 (TIA Portal) 中应用配套规范，以及创建必要的 PLC 变量。

说明

EUROMAP 和 OPC 基金会成立联合工作组"OPC UA 塑料和橡胶机械"。

既有 EUROMAP 推荐标准 EUROMAP 77 (data exchange between injection moulding machines and MES)、82.1 (temperature control devices) 和 83 (general definitions) 等同于中立机构 OPC 基金会发布的标准 OPC 40077、40082-1 和 40083。

其中一个重大更改为，对命名空间进行了更改。例如，EUROMAP 77 的新命名空间为：最新为："http://opcfoundation.org/UA/PlasticsRubber/IMM2MES/"。

在以下示例中，仍使用之前的有效标识和引用。

Euromap 77 示例（新：OPC 40077）

Euromap 77 或后续标准 OPC 40077 对注塑机与上位 MES（制造执行系统）之间的数据交换进行标准化。这样，MES 便能以相同的方式连接所有下一级注塑机。

标准化数据接口有助于将注塑机整合到工厂中。

使用配套规范：概述

在 OPC UA XML 文件"Opc_Ua.EUROMAP77.NodeSet2.xml"中对 Euromap 77 进行了介绍。

说明

Euromap 77、Euromap 83 和 OPC UA for Devices (DI)

对于候选版本 2，一些 Euromap 定义已经从 Euromap 77 转移到 Euromap 83（最新为 OPC 40083）。因此，还需要导入 Euromap 83 的 OPC UA 服务器接口。

"OPC UA for Devices"是普遍适用的信息模型，用于组态硬件和软件组件。此信息模型还是其它配套标准的基础，因此也要导入。

以下部分提供了 OPC UA XML 文件：

Euromap77 (https://www.euromap.org/euromap77)

Euromap83 (https://www.euromap.org/euromap83)

OPC UA for Devices (https://opcfoundation.org/UA/schemas/DI/)

这些 XML 文件定义了符合 Euromap 77 的注塑机的 OPC UA 接口。

使用 Euromap 77：概述

要使用 Euromap 77，请按以下步骤操作：

1. 使用 SiOME 程序创建"IMM_MES_InterfaceType"类型的实例，生成 XML 文件。

   有关如何继续操作的信息，请参见下文中的"步骤 1：在 SiOME 中创建实例"。

2. 在 STEP 7 (TIA Portal) 中，创建对应于"IMM_MES_InterfaceType"类型实例的 PLC 变量和服务器方法（在步骤 1 中创建）。

   有关如何继续操作的信息，请参见下文中的"步骤 2：在 STEP 7 中创建 PLC 变量"。

   有关 OPC UA 节点以及相应 PLC 变量的示例，请参见"为配套规范创建服务器接口 (页 254)"。

3. 在 STEP 7 (TIA Portal) 中，添加配套规范类型的新服务器接口，并导入在步骤 1 中创建的 XML 文件。

   "为配套规范创建服务器接口 (页 254)"部分介绍了如何继续操作。

4. 将新服务器接口的 OPC UA 节点分配给在步骤 2 中创建的相应的 PLC 变量。

   "为配套规范创建服务器接口 (页 254)"部分介绍了如何继续操作。

步骤 1：在 SiOME 中创建实例

以下部分介绍了如何使用免费程序"SiOME"（"西门子 OPC UA 建模编辑器"）。

利用 SiOME，可创建描述服务器接口的 OPC UA XML 文件（信息模型）。

有关 SiOME 的下载链接和相关说明，敬请访问此处的链接 (https://support.industry.siemens.com/cs/cn/zh/view/109755133)。

STEP 7 中的操作步骤

要使用新的服务器接口，请将该服务器接口导入到 STEP 7 项目，请参见"为配套规范创建服务器接口 (页 254)"。

项目加载到 CPU 中后，新的服务器接口可供 OPC UA 客户端使用。

SiOME 1.7.3 中的操作步骤

说明

以下说明介绍了 SiOME 1.7.3 中的操作步骤。

SiOME 的后续版本更易于在用户程序中创建相应的数据块、结构、变量或方法。使用拖放操作，可以将数据从 SiOME 传输到 TIA Portal（用户程序）。在这种情况下，变量等已经正确映射，对于方法，相应的 FB 元素已经在用户程序中正确生成。

使用上面列出的下载链接下载最新的 SiOME 版本，并按照下载中随附文档的说明进行操作。

以下说明介绍了 SiOME 1.7.3 中的操作步骤。

要使用 Euromap 77，请创建包含“IMM_MES_InterfaceType”实例的 XML 文件。

对象类型必须实例化，以便在 OPC UA 服务器的地址空间中显示特定机器的信息模型。

对象类型“IMM_MES_InterfaceType”是 Euromap 77 的根对象类型。“IMM”代表“Injection Moulding Machine”。

请按以下步骤操作：

1. 从 Euromap 网站下载文件“Opc_Ua.EUROMAP77.NodeSet2.xml”和“Opc_Ua_EUROMAP83_NodeSet2.xml”（见上文）。
2. 从 OPC 基金会的网站上下载文件“Opc.Ua.Di.NodeSet2.xml”。

   “Opc.Ua.Di.NodeSet2.xml”文件包含 Euromap 77 使用的类型定义。
3. 启动 SiOME。
4. 首先导入命名空间“http://opcfoundation.org/UA/DI/”。

   为此，请单击“Information model”区域中的“"Import XML”按钮。

   图 11-34 SiOME 中的“导入 XML”(Import XML) 按钮

   SiOME 会为打开的文件显示对话框。
5. 要导入文件，请选择文件“Opc.Ua.Di.NodeSet2.xml”，然后单击“打开”(Open)。
   结果：SiOME 会导入 XML 文件，并在“Namespaces”区域中显示命名空间“http://opcfoundation.org/UA/DI/”。

   标准命名空间“http://opfoundation.org/UA/”始终可在 SiOME 中使用，不需要导入。
6. 现在导入命名空间“http://www.euromap.org/euromap83/”

   为此，请再次单击“Information model”区域中的"Import XML" 按钮。

   选择文件“Opc_Ua.EUROMAP83.NodeSet2.xml”。

   结果：SiOME 会导入 XML 文件，并在“Namespaces”区域中显示命名空间“http://www.euromap.org/euromap83/”。
7. 现在导入命名空间“http://www.euromap.org/euromap77/”

   为此，请再次单击“Information model”区域中的"Import XML" 按钮。

   选择文件“Opc_Ua.EUROMAP77.NodeSet2.xml”。

8. 为项目创建自己的命名空间。

   为此，请右键单击“Namespaces”区域中的“OPC UA Modelling Editor Project”或“Namespaces”，并选择“Add Namespace”。

   SiOME 打开“Add Namespace”对话框。

9. 输入新命名空间的名称。

   本示例中使用的是命名空间“YourCompany.org”。

   SiOME 现在还会显示新的命名空间：

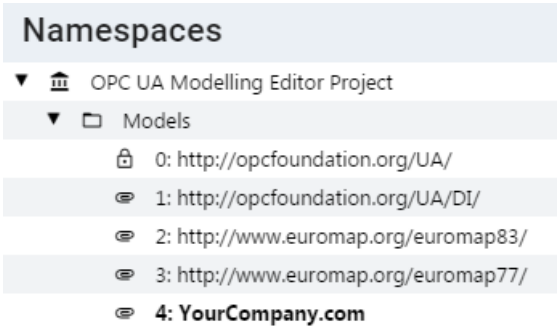


   图 11-35 SiOME 中命名空间的显示

10. 通过配套规范 Euromap 77 的根对象类型 IMM_MES_InterfaceType 创建实例。
    为此，在“Information model”区域中，右键单击“DeviceSet”目录并选择“Add Instance”。

    SiOME 会显示“Add Instance”对话框。

11. 对于“Name”，请为实例输入一个有意义的名称。

    在本例中，请输入“IMM_Manufacturer_01234”。

    对于“TypeDefinition”，请选择“IMM_MES_InterfaceType”。

    该对象类型是 Euromap 77 的根对象类型：如果生成该对象类型的实例，则在 OPC UA 服务器的地址空间中使用一次 Euromap 77。

12. 单击“确定”(OK)。

    SiOME 会在“Information model”区域的“DeviceSet”下显示新实例“IMM_Manufacturer_01234”：

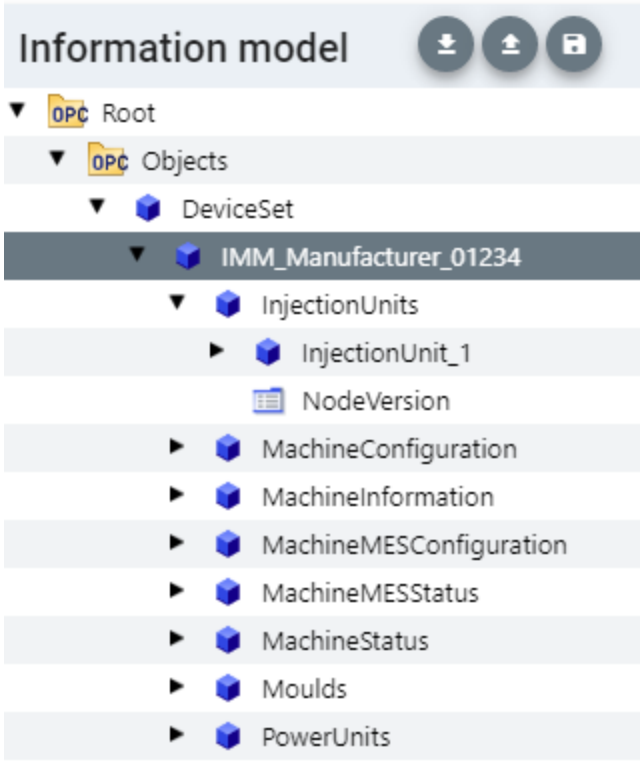


图 11-36 显示信息模型

13. 创建“InjectionUnitType”数据类型的实例。

    为此，请右键单击“Information model”区域中的“InjectionUnits”目录，并选择“Add Instance”。

    SiOME 会显示“Add Instance”对话框。

    对于“Name”，请为实例输入一个有意义的名称。

    在本例中输入“InjectionUnit_1”。

    对于“TypeDefinition”，请选择“InjectionUnitType”。

    单击“确定”(OK)。

14. 在“Moulds”目录中创建“MouldType”对象类型的新实例“Mould_1”。

15. 在“PowerUnits”目录中创建“PowerUnitType”对象类型的新实例“PowerUnit_1”。

16. 保存该 XML 文件。

    为此，请单击“Information model”区域中的“Quick save”按钮。

图 11-37 SiOME 中的“快速保存”(Quick save) 按钮

17. 导出该 XML 文件。

    为此，请单击“Information model”区域中的“"Export XML”按钮。

图 11-38 SiOME 中的“导出 XML”(Export XML) 按钮

SiOME 会显示“导出 XML”(Export XML) 对话框.

18. 将所有命名空间保持激活状态并单击“确定”(OK)。

    SiOME显示“另存为”(Save as) 对话框。

19. 选择一个有意义的名称并保存导出的文件。

    本例中，将 XML 文件命名为“IMM_Manufacturer_01234”。

结果：

现已创建使用一次配套规范“Euromap 77”（包含一个实例）的 XML 文件。

步骤 2：在 STEP 7 中为 Euromap 77 实例创建 PLC 变量。

对于 Euromap 77，必须在用户程序中提供 PLC 变量和服务器方法，并分配“IMM_MES_InterfaceType”类型的实例。

要为“IMM_MES_InterfaceType”类型的实例创建 PLC 变量，请按以下步骤操作：

1. 创建用户自定义数据类型 (UDT)

   下图以用户自定义数据类型“InjectionUnit”的开头为例。

   该数据类型的结构与“IMM_MES_InterfaceType”类型中的“InjectionUnit”相同。

   请务必使用与 OPC UA 数据类型兼容的 SIMATIC 数据类型（参见下文的“数据类型映射”）。

InjectionUnit

	Name	Data type	Default value	Accessible from HMI/OPC UA	Writable from HMI/OPC UA
	BarrelId	String	''	☑	☑
	Index	UDInt	0	☑	☑
	InProduction	Bool	false	☑	☑
	IsPresent	Bool	false	☑	☑

图 11-39 在 STEP 7 中创建 UDT

2. 将新的全局数据块添加到 STEP 7 项目中。

   在本例中，将数据块命名为“IMM_Manufacturer_01234”，以指代相应制造商和序列号的注塑机。

3. 在该数据块中创建一个新元素。

   在本例中，将该元素命名为“InjectionUnit_1”

4. 为该元素分配新的用户自定义数据类型“InjectionUnit”。

结果

在 STEP 7 项目中，已为“IMM_Manufacturer_01234”数据块中的 Euromap 77 创建一个变量。

11.3.4.3 为配套规范创建服务器接口

有关配套规范的基本信息，请参见“使用 OPC UA 配套规范 (页 248)”。此部分还详细探讨了提供注塑机模型的 Euromap 77 配套规范的优点。

举例来说，利用这一配套标准，S7-1500 CPU 可控制注塑机，并为 OPC UA 客户端（比如上一级 MES 系统）提供接口来访问注塑机的功能和变量。

“配套标准”类型的 OPC UA 服务器接口会将客户端的访问限制为上一级系统（MES 系统）等必需的几个功能和变量。

以下说明介绍了如何在 STEP 7 (TIA Portal) 中创建仅包含 Euromap 77 配套规范的服务器接口。

如果要使 OPC UA 客户端可访问注塑机管理所需变量或方法以外的其它变量或方法，只需创建另一 OPC UA 服务器接口即可。通过这种方式，可清晰地排列作为 OPC UA 服务器的 CPU 的功能。

为配套规范创建服务器接口

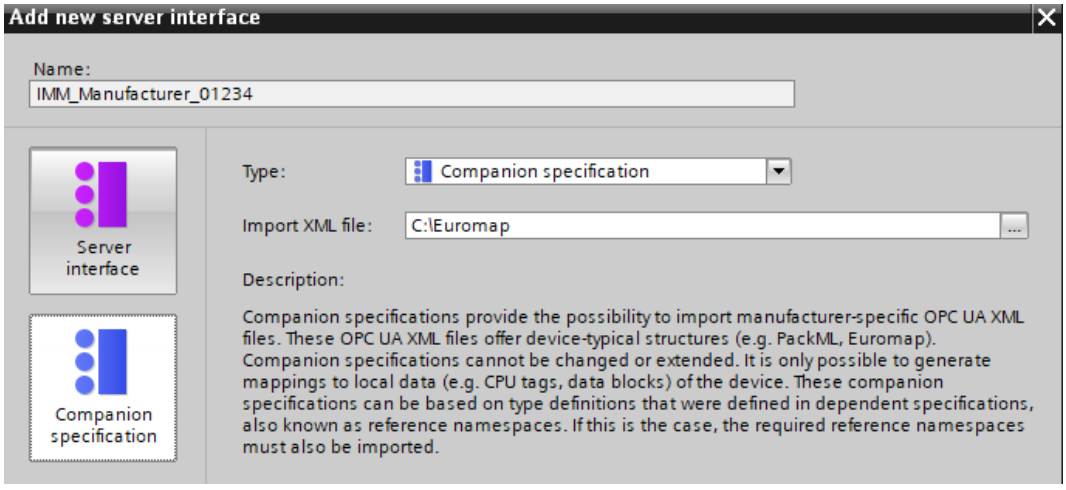
要通过 STEP 7 (TIA Portal) 为配套规范创建服务器接口，请执行以下操作：

1. 选择要作为 OPC UA 服务器使用的 CPU。
2. 在项目树中，单击“OPC UA 通信 > 服务器接口”(OPC UA communication > Server interfaces)。
3. 双击“添加新服务器接口”(Add new server interface)。
4. 要选择此类型的服务器接口，请单击“配套规范”(Companion specification)。

   新服务器接口的一般名称会输入到对话框中，例如“Server_Interface_1”。
5. 更改新服务器接口的名称，使其在项目中具有说明性含义。

   按照 Euromap 77 规定，名称应采用以下结构：“IMM_<Manufacturer>_<Serial number>”。

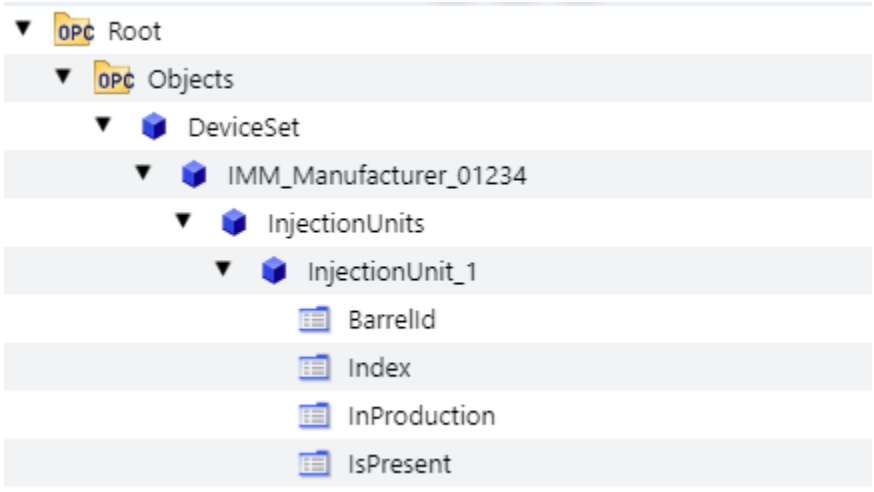
   示例使用的名称为“IMM_Manufacturer_01234”。



6. 在“导入 XML 文件”(Import XML file) 字段中，选择描述信息模型的 XML 文件。

   “使用 OPC UA 配套规范 (页 248)”部分介绍了如何使用 SiOME 工具创建此类 XML 文件。

   下图显示了信息模型的一部分：“IMM_MANUFACTURER_0123456”是由 Euromap 77 定义的“IMM_MES_InterfaceType”类型的实例（应用）。“InjectionUnit_1”是 Euromap 77 的“InjectionUnitType”类型的实例。



7. 单击“确定”(OK)。

   STEP 7 (TIA Portal) 将导入所选 XML 文件中描述的信息模型。

   如果在已导入 XML 文件中使用 STEP 7 (TIA Portal) 中尚不存在、且不包含在已导入 XML 文件中的类型定义，则会出错。

   本例中导入的 XML 文件使用在以下命名空间 (Namespaces) 中定义的类型定义：

   – http://opcfoundation.org/UA/DI/
   – http://www.euromap.org/euromap83/
   – http://www.euromap.org/euromap77/

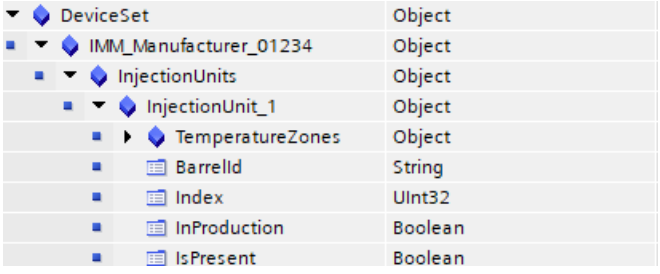
   提示：STEP 7 在 OPC UA 接口编辑器的下方区域（“属性”(Properties) 选项卡）中显示缺少的命名空间。

   为此，在项目树中选择服务器接口（这里选择的是IMM_Manufacturer_01234），并在巡视窗口中选择“命名空间”(Namespaces) 区域。选择缺少的命名空间。

   如果 STEP 7 项目中缺少一个或多个命名空间，可为每个命名空间创建“引用命名空间”类型的新服务器接口。

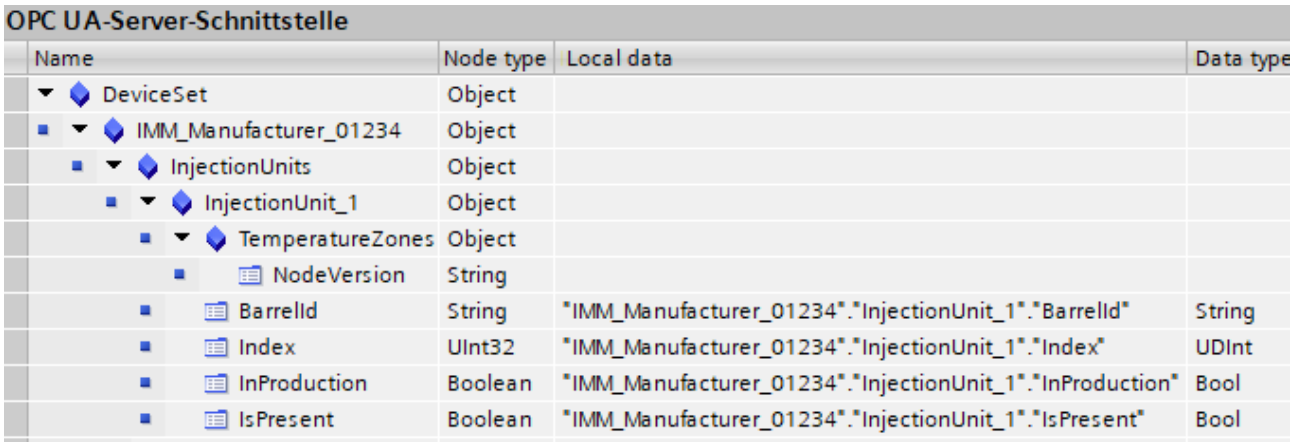
   “为引用命名空间创建服务器接口 (页 273)”部分介绍了相应的操作步骤。

   如果所有引用命名空间均可用，STEP 7 显示的表格不含任何错误：



8. 将 OPC UA 元素从表格的右侧区域（OPC UA 元素）拖放到表格的左侧部分（OPC UA 服务器接口），从而将相应的 OPC UA 元素（本地 PLC 变量）分配给 Euromap 77 的相应 OPC UA 节点。

   下图显示了将本地数据（PLC 变量）分配给 Euromap 77 的 OPC UA 节点的部分：



注意
在 OPC UA 服务器接口的节点上检查 CPU 本地数据的映射 当服务器接口中存在无效的分配（映射）时，它们可造成错误的读取和写入操作。检查分配并运行一致性检查。

关于服务器接口的信息

用于组态 OPC UA 服务器接口的编辑器采用表格结构，可提供以下信息：

- **名称**

  本例中，顶级节点（根节点）名为“IMM_Manufacturer_01234”。如果客户端在服务器的地址空间中进行浏览，该节点是所有下级节点的容器。该节点的 BrowseName 和 DisplayName 取决于为服务器接口分配的名称。

  举例来说，在这种情况下，该名称代表作为整体的注塑机。这是此处使用的 Euromap 77 配套规范的实例名称。按照配套规范，实例名称应以“IMM”开头，后接注塑机制造商名称；机器序列号添加到结尾处。这样便可唯一地标识机器。

  其它所有（下级）节点的名称均由规范定义（上例中由 Euromap 77 定义）。不得更改这些节点名称。这样可确保所有注塑机的统一视图符合规范的规定。

- **节点类型**

  OPC UA 节点的类型。类型由所用配套规范指定。

  以下情况下，STEP 7 会将表格中的一个节点类型标为彩色：

  – 导入的 XML 文件中不包含该节点类型的定义，或者

  – 定义类型的命名空间在 STEP 7 中不可用。

  在这种情况下，为缺失的命名空间或所缺失命名空间中的每一个创建类型为“引用命名空间”的服务器接口。

  缺少的命名空间位于服务器接口属性的“命名空间”(Namespaces) 下。

- **本地数据**

  STEP 7 会显示分配给 OPC UA 节点的数据块：CPU 从该数据块中读取 OPC UA 节点的值。

  如果数据块以彩色突出显示（例如，在一致性检查之后），则指定的数据块在 CPU 中不可用。

  在这种情况下，需要创建 CPU 中缺失的（用户程序）数据块，并为其提供值。

- **数据类型**

  CPU 中 PLC 变量（例如数据块的元素）的 SIMATIC 数据类型，会通过该数据类型读取 OPC UA 节点的值（UAVariable 类型）或向该数据类型分配值。

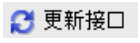
更新界面

可更新配套规范的服务器接口。

示例：在导入配套规范后，然后导入配套规范的相关参考命名空间时，参考命名空间的类型定义不会立即生效。

单击“更新接口”(Update interface) 按钮后，将修复配套规范中缺失的类型定义。

“更新接口”(Update interface) 按钮：



生成本地数据

如果服务器接口的节点尚未分配（“映射”）CPU 的本地数据，则可选择为所有节点或者选定的节点生成本地数据。系统将自动映射新创建的本地数据。

对于未映射的所有节点，可单击“生成本地数据”(Generate local data) 按钮；对于单个节点，可选择相应节点并单击“生成本地数据”(Generate local data) 快捷菜单，自动生成本地数据。

“生成本地数据”(Generate local data) 按钮：

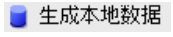


图 11-40 “生成本地数据”(Generate local data) 按钮：

生成的节点只能映射本地数据。即，无对象、无文件夹、无方法或方法无输入/输出参数。

单击该按钮或选择快捷菜单后，必须在后续对话框中选择在新数据块中或现有数据块中创建本地数据。

一致性检查

可选择检查服务器接口。

STEP 7 (TIA Portal) 会检查是否已为服务器接口 PLC 变量（数据块）的 OPC UA 节点分配兼容的 SIMATIC 数据类型。

要检查服务器接口的一致性，请单击 OPC UA 服务器接口编辑器工具栏中的以下图标：

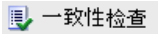


图 11-41 “一致性检查”(Consistency check) 按钮

导出接口

可选择以 XML 文件格式导出 OPC UA 服务器接口。该 XML 文件包含服务器接口引用的所有数据类型定义。

要导出 OPC UA 服务器接口，请单击 OPC UA 服务器接口编辑器工具栏中的以下图标：

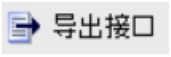


图 11-42 “导出接口”(Export interface) 按钮

11.3.4.4 创建用户自定义服务器接口

简介

说明基于以下示例：

生产单元“Cell_1”周围的防护围栏。围栏配有门“Gate_1”。

S7-1500 CPU 控制整个生产单元，还控制通过 Gate_1 进入生产单元的权限。

机器人将药物装入生产单元的盒子中，然后将盒子堆放在货盘上。

用于自动化物料运输的自驾车辆将货盘移动到中央仓库，从而通过 Gate_1。

CPU 发布一个服务器接口，无人驾驶运输系统可通过该接口安排 Gate_1 打开。

该服务器接口包含用于打开门的服务器方法“smOpenGate”和用于指示门状态（打开或关闭）的变量“Gate_1_State”。

创建用户自定义服务器接口

要创建服务器接口，请按以下步骤操作：

1. 选择已使用并组态为 OPC UA 服务器的 CPU。
2. 单击“OPC UA 通信 > 服务器接口”(OPC UA communication > Server interfaces)。
3. 双击“添加新服务器接口”(Add new server interface)。
   STEP 7 会显示以下对话框。

图 11-43 添加服务器接口

4. 更改新服务器接口的名称，使其在项目中具有说明性含义。
   本示例中，将 STEP 7 建议的名称“Server-interface_1”改为“Cell_1”。
5. 单击“服务器接口”(Server interface)，然后单击“确定”(OK)。
6. 单击“OPC UA 元素”(OPC UA elements) 区域中“程序块”(Program blocks) 前面的三角形。
   STEP 7 会显示以下表格供用户编辑：

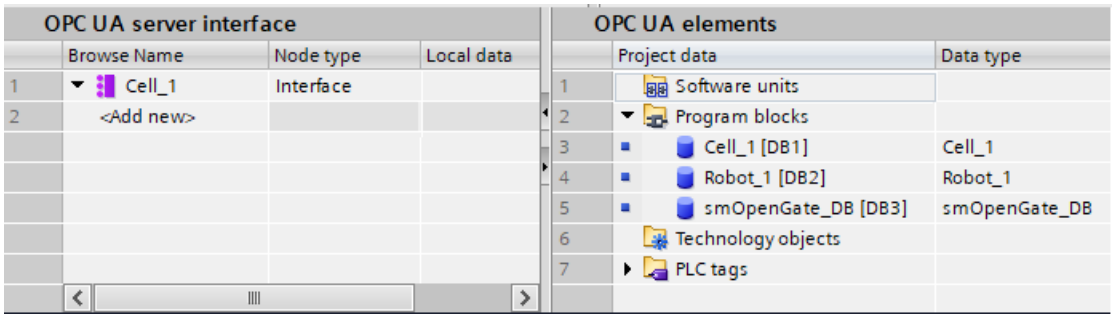


图 11-44 编辑服务器接口

该编辑器分为两个区域。

- **OPC UA 服务器接口**

  左侧是服务器接口“Cell_1”的根节点。

  该接口目前仍为空。尚未向该服务器接口添加任何 OPC UA 元素。

- **OPC UA 元素**

  右侧为 OPC UA 元素。

  OPC UA 元素是到目前为止在 STEP 7 项目中创建的对象，具有属性“可从 HMI/OPC UA 访问”(Accessible from HMI/OPC UA)。

  可将这些 OPC UA 元素添加到新的服务器接口“Cell_1”。

7. 将 OPC UA 元素拖放到新服务器接口的“<新增>”(<Add new>) 行中。

说明

以下规则普遍适用：如果将数据块或工艺对象存储在表格的左侧区域，则 STEP 7 (TIA Portal) 会在服务器接口中创建一个对象。数据块的元素作为单独的节点排列在该对象下方。

如果将结构存储在表格的左侧区域，STEP 7 会为结构整体创建一个节点，并会为结构的各个元素创建节点。

这一点同样适用于数组：同样，STEP 7 会为数组整体创建一个节点，并会为数组的各个元素创建节点。

将方法放在表格的左侧区域时，STEP 7 会创建一个节点；将显示已插入方法的参数以供参考。

在本示例中，将“Gate_1_State”变量从右侧区域拖到左侧区域的“<新增>”(<Add new>) 位置。

然后，将服务器方法拖到左侧区域。

服务器方法位于右侧区域的“smOpenGate_DB [DB3]”数据块中。

STEP 7 (TIA Portal) 将显示如下对话框：

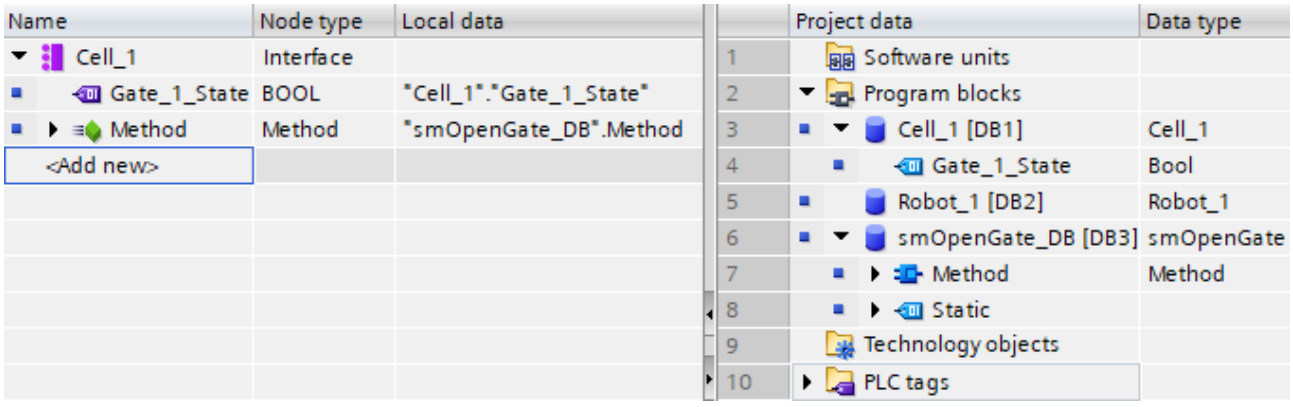


图 11-45 向服务器接口添加 OPC UA 元素

注意
在 OPC UA 服务器接口的节点上检查 CPU 本地数据的映射 当服务器接口中存在无效的分配（映射）时，它们可造成错误的读取和写入操作。检查分配并运行一致性检查。 在 TIA Portal 中，由于无效分配仅生成警告而不生成错误，因此可按以下步骤进行逐步操作： 例如，在第一步中，可修改程序/本地数据，确保程序运行无任何错误。在下一步中，可修改 OPC UA 服务器的接口并消除不一致错误。 在 TIA Portal 生成警告时，OPC UA 服务器接口在运行期间失效。OPC UA 服务器生成运行系统错误。

限制 OPC UA 服务器的视图

通过选择 OPC UA 元素可限制 OPC UA 服务器的视图以及 OPC UA 客户端的选项。

在该示例的服务器接口中，缺少“Robot_1”数据块，因为工业卡车不需要访问机器人的服务器方法和变量。

在这种情况下，最好在 S7-1500 CPU 的 OPC UA 属性中禁用标准服务器接口（SIMATIC 命名空间），以便无法通过其它任何方式访问过滤的节点。

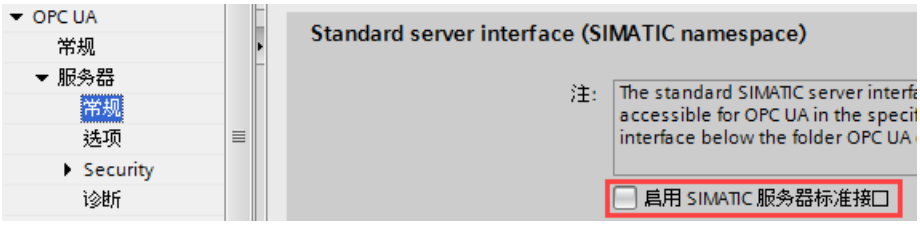


图 11-46 禁用标准服务器接口

还可在服务器接口的属性中禁用每个已组态 OPC UA 服务器接口的可见性，从而避免客户端在接口运行期间使用该服务器接口。

- 为此，请选择服务器接口并右键单击“属性”(Properties) 命令。

举例来说，可通过此选项集中定义多个服务器接口，并且仅启用和下载所需服务器接口。

定义了服务器接口后，可将其拖动到项目树的其它 CPU 中。

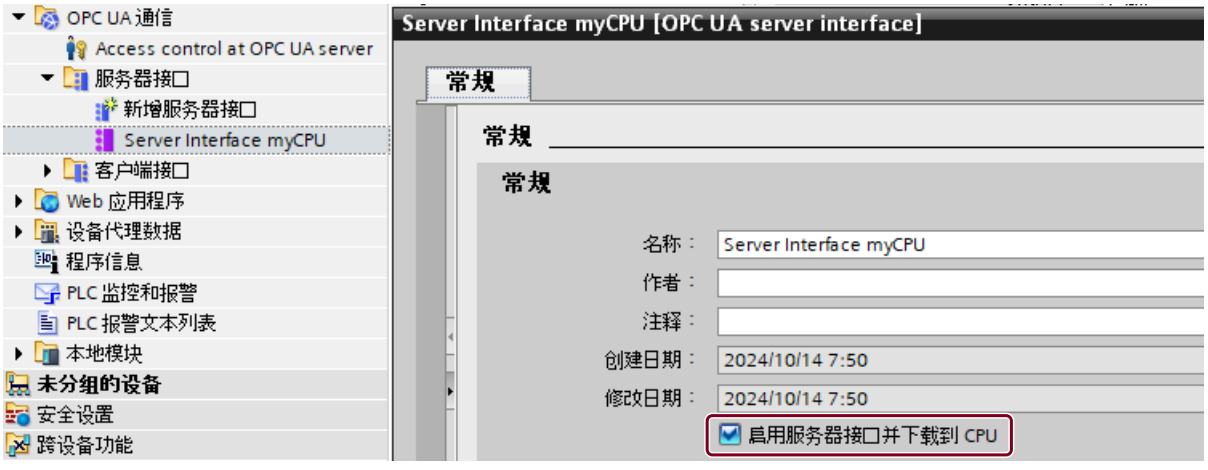


图 11-47 禁用服务器接口可见性

关于服务器接口的信息

"OPC UA 服务器接口"(OPC UA Server Interface) 编辑器采用表格结构，提供以下信息：

请注意，最初并不会显示所有列。可右键单击表格的标题行确定显示的列。

现在该行时，可在巡视窗口（"OPC UA 属性"区域）中显示该节点的 OPC UA 属性，如节点 ID、节点类别、节点类型及描述。

- **BrowseName**

  用户自定义服务器接口的语言中立名称位于最顶端 (BrowseName)。可任意选择该名称。

  已添加到服务器接口的各个 OPC UA 节点的名称 (BrowseNames) 位于接口名称下方。

  不能在该对话框中更改 OPC UA 节点的名称。名称来自 STEP 7 项目。

  可将 OPC UA 节点从表中删除。这意味着该节点不再属于服务器接口，并且不再对 OPC UA 客户端可见。

- **DisplayName**

  与 BrowseName 类似。但名称可进行翻译并以相应的语言显示（若可用）。

- **节点 ID**

  OPC UA 节点的 NodeId，例如 http://Server-Node_1；i=1

- **节点类型**

  OPC UA 节点的可指定为 BOOL、BYTE、INT 等。

  这些节点类型是由 Siemens 定义的，而不是 OPC 基金会定义的。例如，OPC 基金会为 BOOL 使用布尔型节点类型。BOOL 直接由布尔型派生而来。

  不能在此对话框中更改指定的节点类型：如果要使用其它节点类型，必须在 STEP 7 项目中更改相应 PLC 变量的类型。

- **数据类型**

  指定 STEP 7 项目中使用的 SIMATIC 数据类型，例如布尔型、字节型、整型等。

- **访问等级**

  – 如果 OPC UA 节点为变量（UAVariable 类型），则节点只能是可读 (RD) 或可读写 (RD/WR) 节点。

  – 如果 OPC UA 节点为方法（UAMethod 类型），则该节点始终可调用。

- **本地数据**

  CPU 中 SIMATIC 数据类型的数据块，会通过该数据块读取 OPC UA 节点的值（UAVariable 类型）或向该数据块写入值。

生成本地数据

如果服务器接口的节点尚未分配（"映射"）CPU 的本地数据，则可选择为所有节点或者选定的节点生成本地数据。系统将自动映射新创建的本地数据。

对于未映射的所有节点，可单击"生成本地数据"(Generate local data) 按钮；对于单个节点，可选择相应节点并单击"生成本地数据"(Generate local data) 快捷菜单，自动生成本地数据。

"生成本地数据"(Generate local data) 按钮：

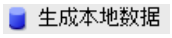


图 11-48 "生成本地数据"(Generate local data) 按钮：

生成的节点只能映射本地数据。即，无对象、无文件夹、无方法或方法无输入/输出参数。

单击该按钮或选择快捷菜单后，必须在后续对话框中选择在新数据块中或现有数据块中创建本地数据。

一致性检查

可选择检查服务器接口的一致性。

在一致性检查过程中，STEP 7 会检查服务器接口的 OPC UA 节点是否分别分配给合适的 OPC UA 元素（相同数据类型），或者使用的元素是否仍存在于 CPU 中。

要检查服务器接口的一致性，请单击 OPC UA 服务器接口编辑器工具栏中的以下图标：

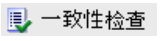


图 11-49 "一致性检查"(Consistency check) 按钮

导出接口

可选择以 XML 文件格式导出 OPC UA 服务器接口。该 XML 文件包含服务器接口引用的所有数据类型定义。

要导出 OPC UA 服务器接口，请单击 OPC UA 服务器接口编辑器工具栏中的以下图标：

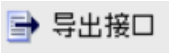


图 11-50 "导出接口"(Export interface) 按钮

更多信息

有关 OPC UA 通信模板副本的信息，请参见"OPC UA 通信的模板副本 (页 350)"部分。

参见

OPC UA 服务器的客户端访问和本地访问 (页 209)

11.3.4.5 配套规范的数据类型

数据类型的映射

下表显示了各个 OPC UA 数据类型的兼容 SIMATIC 数据类型。

根据下图所示，指定数据类型（SIMATIC 数据类型 - OPC UA 数据类型）。系统不支持其它分配方式。STEP 7 不会检查是否遵循该规则，因此也不会预防分配错误。用户需确保所做的选择和数据类型分配符合规则。

例如，所列出的数据类型，也可用作自定义服务器方法中输入和输出参数结构 / UDT 的元素（UAMethod_InParameters 和 UAMethod_OutParameters）。

表格 11-3 数据类型的映射

SIMATIC 数据类型	**OPC UA 数据类型性**
BOOL	Boolean
SINT	SByte
INT	Int16
DINT	Int32
LINT	Int64
USINT	Byte
UINT	UInt16
UDINT	UInt32
ULINT	UInt64
REAL	Float
LREAL	Double
LDT	DateTime
WSTRING	String
DINT	Enumeration (Encoding Int32) 和所有派生的数据类型
所需的用户自定义数据类型（UDT，user-defined data type） 用户创建的自定义数据类型必须以“Union_”为前缀，如“Union_MyDatatype”。请参见表格下方的示例。 在 UDT 中，第一个元素 (Selector) 的数据类型必须为“UDINT”。	UNION 和所有派生的数据类型
请参见“LocalizedText 和 ByteString 数据类型 (页 265)”	LocalizedText ByteString

所需的 UNION 用户自定义数据类型

下图显示了数据类型为“Union_MyDatatype”的变量“MyVariable”。

此 SIMATIC 数据类型对应于数据类型为 UNION 的 OPC UA 变量。

下图显示了声明示例：当 Selector = 1 时，Union 取 ByteArray；当 Selector = 2 时，Union 取 WString。

名称	数据类型
Static	
Selector	UDInt
ByteArray	Array[0..1] of Byte
WString	WString[42]

11.3.4.6 LocalizedText 和 ByteString 数据类型

在 TIA Portal 版本 V17 和 S7-1500 CPU 固件版本 V2.9 及以上版本中，可通过两个内置的 OPC UA 数据类型"LocalizedText"和"ByteString"对相应的 SIMATIC 数据结构进行映射。有关这些 OPC UA 数据类型的定义，另请参见"OPC 10000-3 数据类型"的定义。

这些数据类型用于配套规范中，用户程序可通过 OPC UA 接口编辑器进行轻松处理编辑。

LocalizedText

一种结构，包含有一个带有区域设置标识符（如，en-US）字符串。

该结构中包含三个按既定顺序排列的元素和以下 SIMATIC 结构：

- **编码**（数据类型 OPC_UA_LocalizedTextEncodingMask）：位 0 指示"区域设置"(Locale) 字段是否包含内容；位 1 指示"文本"(Text) 字段是否包含内容。这两个字段中均应包含内容。因此，建议将 SIMATIC 的"编码"值设置为 2#00000011。
- **本地**（WString 数据类型）：区域设置，如"en-US"。
- **文本**（WString 数据类型）：文本框，如"Text"。

ByteString

一个八位字节序列。

该结构的构成如下所示：

- **ActualLength**（数据类型"OPC_UA_ByteStringActualLength"）：ByteString 数组的长度
- **ByteString**（"Array of Byte"数据类型）：字节数组

要求

已创建了一个 OPC UA 服务器接口。

应用

导入一个包含"LocalizedText"或"ByteString"类型定义的配套规范或参考命名空间。

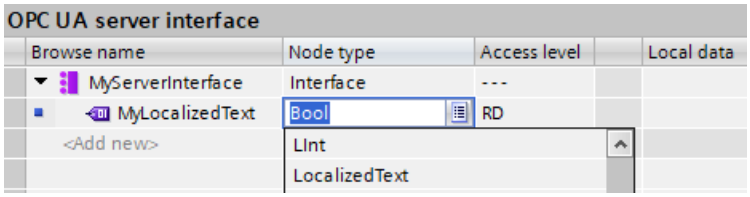
除此之外，还可自行创建一个服务器接口并定义带有数据类型"LocalizedText"或"ByteString"的地址模型。相关的操作过程，请参见下一章节。

操作步骤

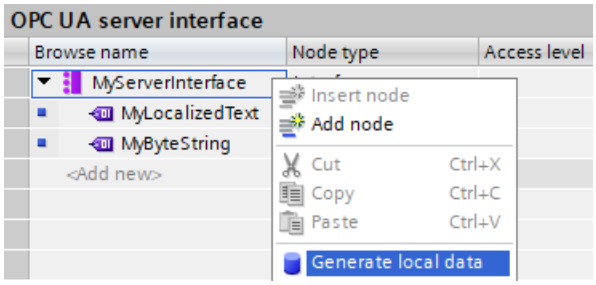
在以下章节中，将介绍如何使用接口编辑器创建一个类型为“LocalizedText”或“ByteString”的节点，并为该节点自动创建一个 SIMATIC 数据结构。

要在服务器接口中定义类型为“LocalizedText”/“ByteString”的 OPC UA 节点，请执行以下操作步骤：

1. 在“OPC UA 服务器接口”(OPC UA server interface) 区域中，创建类型为“LocalizedText”或“ByteString”的节点。这些节点类型包含在可选节点类型列表中。



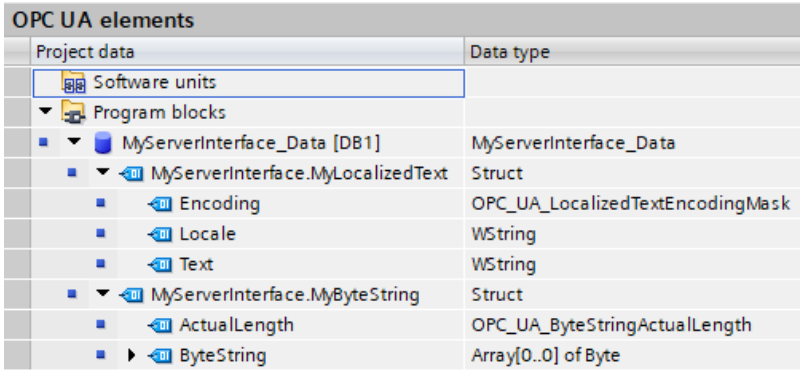
2. 在快捷菜单中，选择“生成本地数据”(Generate local data) 命令。要生成本地数据，可选择一个数据块。例如，名为“MyServerInterface_Data”的新数据块。



结果：STEP 7 生成映射所需的结构，但仍需对“LocalizedText”所需的文本长度 (Text) 和所需的区域设置 (Locale) 进行调整。

此时，还需对“ByteString”的长度和数组进行调整。

在一致性检查后生成的一条警告消息中，指示需执行的相应调整。



规则

- 节点“LocalizedText”或“ByteString”也可按照上文所述创建 UDT 结构，用于各种 DB 元素。
- 节点类型“LocalizedText”或“ByteString”也可用于其它结构（嵌套）中。
- SIMATIC 结构“LocalizedText”或“ByteString”只能整体使用；不能单独使用其中一种数据类型，如“OPC_UA_LocalizedTextEncodingMask”。
- 各种方法的输入和输出参数也可为数据类型/节点类型“LocalizedText”或“ByteString”。

11.3.4.7 将其它 OPC UA 数据类型用于配套规范

除了“映射数据类型”部分列出的 OPC UA 数据类型以及 SIMATIC 端对应的数据类型之外，还可使用以下 OPC UA 基本数据类型：

- OpcUa_NodeId
- OpcUa_QualifiedName
- OpcUa_Guid
- OpcUa_XmlElement
- OpcUa_ByteString (页 265)
- OpcUa_LocalizedText (页 265)

在应用程序中使用上文中所示基本数据类型的变量时，应满足以下要求：基本数据类型需用作复杂数据类型，且结构与相应的 OPC UA 基本数据类型完全相同。

- OpcUa_NodeId 和 OpcUa_QualifiedName 可用作系统数据类型；因此，这些数据类型不仅可用于单个变量，也可用作结构中的元素。
- 对于基本数据类型或内置数据类型 GUID 和 XmlElement，需根据 OPC UA 规范创建一个 PLC 数据类型，之后将其用作某个结构中的元素，以便对这些元素的数据类型进行解析。在下文中，介绍了每个基本数据类型所对应的 PLC 数据类型。
- 对于 OpcUa_ByteString 和 OpcUa_LocalizedText，要求已在 TIA Portal V17 中创建，以便在“配套规范”类型的服务器接口中使用这些数据类型：
  - 在服务器接口中创建相应的节点类型（例如，OpcUa_LocalizedText）
  - 单击“生成本地数据”(Generate local data)

  然后，STEP 7 在 DB 中自动生成适当的数据结构。
- 对于 OpcUa_Guid，这些要求已在 TIA Portal V19 中得到满足。按照上一节所述进行操作。

系统数据类型“OPC_UA_NodeId”

在下表中，列出了 OPC UA 基本数据类型“OpcUa_NodeId”各个参数的含义。OPC_UA_NodeId 用于识标 OPC UA 服务器中的节点。

参数	S7 数据类型	含义
NamespaceIndex	UINT	OPC UA 服务器中，节点的命名空间索引。 例如，节点可以是一个变量。
Identifier	WSTRING[254]	节点（对象或变量）的名称取决于标识符类型： • 数字标识符：节点使用一个数字进行标记，如“12345678”。 • 字符串标识符：节点使用一个名称进行标记，如“MyTag”。不区分大小写。
IdentifierType	UDINT	标识符的类型 • 0：数字标识符 • 1： 字符串标识符 • 2： GUID • 3：Opaque

系统数据类型“OPC_UA_QualifiedName”

请参见下表中系统数据类型“OPC_UA_QualifiedName”的结构：

名称	S7 数据类型	含义
NamespaceIndex	UINT	名称的命名空间索引。
Name	WSTRING[64]	节点或变量的名称。

系统数据类型“GUID”

自 TIA Portal V19 起，“Guid”数据类型可用作节点类型。有关该 OPC UA 数据类型的定义，另请参见 OPC 10000-6 映射。

此图显示了服务器接口中变量节点的“Guid”数据类型的分配。

下图显示了使用“常规本地数据”自动创建的具有 GUID 元素的数据块。

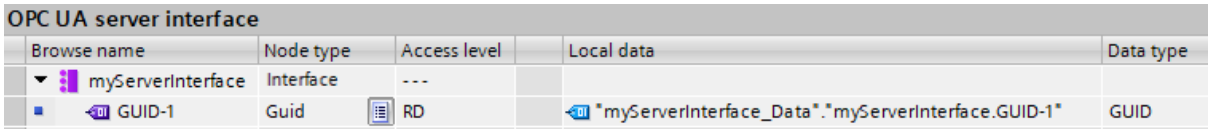


图 11-51 服务器界面中的 GUID 节点

下图显示了使用"常规本地数据"自动创建的具有 GUID 元素的数据块。

Name	Data type	Start value
Static		
myServerInterface.GUID-1	GUID	
Data1	UDInt	16#C4965784
Data2	UInt	16#0DFE
Data3	UInt	16#4B8F
Data4	Array[0..7] of Byte	
Data4[0]	Byte	16#87
Data4[1]	Byte	16#0A
Data4[2]	Byte	16#74
Data4[3]	Byte	16#52
Data4[4]	Byte	16#38
Data4[5]	Byte	16#C6
Data4[6]	Byte	16#AE
Data4[7]	Byte	16#AE

图 11-52 具有 SDT GUID 的数据块

基本数据类型 XmlElement 的 UDT

XmlElement 是一种序列化的 XML 段（UTF 8 字符串）。

为基本数据类型"XmlElement"创建以下 PLC 数据类型：

XmlElement

	Name	Data type	Default value
1	XmlElement	WString	WSTRING#''

11.3.4.8 动态数组

有关动态数组的实用信息

在 TIA Portal V19 和 S7-1500 CPU 固件版本 V3.1 及以上版本中，OPC UA 服务器和 OPC UA 客户端各接口中支持"动态数组"的使用。下文中解释了服务器接口的原理和功能。动态数组的结构/UDT 结构与客户端和服务器应用程序的相同。

通过动态数组，可定义服务器接口端的数据结构，这些数据结构的元素数量在运行过程中会发生变更。在 CPU 端，动态数组对应一个大小固定的结构。

示例：生产中用于质量保证的过程数据将传输到更高级别的管理系统。过程数据元素的数量不同，具体取决于产品类型。如果产品类型在运行期间更改，则所需过程数据元素的数量也会更改。

在 STEP 7 (TIA Portal) 中，数组就是可分配给变量的数据类型。数组表示一定数量的数据类型相同的元素组成的数据结构。这些元素支持除 ARRAY 之外的所有数据类型。

相比之下，OPC UA 不将数组定义为数据类型。任何变量值 (Value) 都可以是数组。OPC UA 中通过以下 Variable NodeClass 属性和特征，定义变量作为数组时的"几何"结构（https://reference.opcfoundation.org/Core/Part3/v104/docs/5.6 除外）：

属性	数据类型	可能值及其含义
ValueRank (Mandatory)	Int32	• n > 1：值是一个 n 维数组。 • 1：值是一个一维数组 • 0：值是一个一维或多维数组 • -1：值是标量（而非数组） • -2：值是标量或任意维度的数组 • -3：值是标量或一维数组
ArrayDimensions (Optional)	UInt32[]	每个维度支持的最大长度。 0 或空：未知长度。
MaxArrayLength (Optional)	UInt32	对于 ValueRank 属性未设置为标量的 DataVariables： 数组的最大长度。 示例：2x3x10 长度的三维数组的长度为 60。

此处使用"动态数组"在 SIMATIC 数据类型端为 OPC UA 中定义的变量几何结构动态元素建立对应元素，以便映射数组数据。运行期间可以使用 STEP 7 程序更改所映射数组的几何结构。在 OPC UA 端上的 OPC UA 服务器接口中定义一个节点，客户端可通过该接口访问映射的 CPU 数据（动态数组）。

原理

为了将 OPC UA 变量映射到数组类型的 CPU 变量，请使用具有新系统数据类型"OPC_UA_ArrayBoundaries"的结构或 UDT。

结构或 UDT 具有以下结构元素，具体结构如下：

- 第一个结构元素（数组大小）本身是一个数组，用于定义动态数组的当前大小。系统数据类型"OPC_UA_ArrayBoundaries"为每个维度定义服务器接口中可用的下标范围（每种情况的下限和上限元素）。
- 第二个结构元素（数组数据）包含 OPC UA 允许的可选数据类型的所有数组元素。

例如，STEP 7 (TIA Portal) 自动确保 SIMATIC 端的负下标转换为 OPC UA 端（服务器接口）的非负下标（从"0"开始）。

一维数组示例

以下示例列出了一维简单数组的结构（最多包含 10 个元素）。数组的当前大小设置为 3 个元素 [0..2]。

名称	数据类型	可能的值 / 说明
1-Dim-Struct	Struct (UDT)	
- **ArraySizeAct**	OPC_UA_ArrayBoundaries 类型的 Array [0..0]	当前数组大小
- - ArraySizeAct[0]	OPC_UA_Boundaries	一维
- - - Lower	DInt	0（下标下限）
- - - Upper	DInt	2（可能的下标上限值：0..9）

名称	数据类型	可能的值 / 说明
- **MyDataArray**	Byte 类型的 Array[**0**..9]	**数组数据（所有数组元素）**
- - MyDataArray[0]	Byte	第一个元素
...	...	...
- - MyDataArray[9]	Byte	最后一个元素

二维数组示例

以下示例列出了一个二维数组的结构（包含有 10x6 = 60 个元素）。此时，两个维度的下标下限均不为“0”。数组的当前大小设置为 2x6 = 12 个元素。

名称	数据类型	可能的值 / 说明
2-Dim-Struct	Struct (UDT)	
- **ArraySizeAct**	OPC_UA_ArrayBoundaries 类型的 Array [0..1]	**当前数组大小**
- - ArraySizeAct[0]	OPC_UA_Boundaries	第一个维度
- - - Lower	DInt	**-9**（下标下限）
- - - Upper	DInt	-8（可能的下标上限值：-9..0）
- - ArraySizeAct[1]	OPC_UA_Boundaries	第二个维度
- - - Lower	DInt	**5**（下标下限）
- - - Upper	DInt	10（可能的下标上限值：5..10）
- **MyDataArray**	Byte 类型的 Array[-**9**..0, **5**..10]	**数组数据（所有数组元素）**
- - MyDataArray[-9,5]	Byte	第一个元素
- - MyDataArray[-9,6]	Byte	第二个元素
...	...	...
- - MyDataArray[0,9]	Byte	第二个到最后一个元素
- - MyDataArray[0,10]	Byte	最后一个元素

对服务器接口的影响

服务器接口的显示内容：如果将动态数组（例如 2-Dim-Struct）拖放到服务器接口编辑器中的服务器接口中，STEP 7 (TIA Portal) 将创建不包含子元素的相同名称的节点。该节点在“节点类型”(Node type) 列中指示类型“ARRAY”以及数组的最大可能长度。

浏览：动态数组可用作 OPC UA 地址空间中的节点。与静态数组不同，该数组的子元素作为子节点不可见。

读取：可使用读取服务读取动态数组。客户端可指定要读取整个数组还是只读取数组的一部分 (IndexRange)。只有位于当前数组范围边界内的读取部分才能返回状态“良好”。如果每个维度的数组大小都是“0”，则将返回一个状态“良好”的空数组。
如果维度中读取的数组大小为“0”，则返回一个状态为“良好”的空数组。

写入：如果待写入的范围小于等于最大数组大小，则写入所有元素（状态为“良好”）。如果待写入的范围大于最大数组大小，则不写入任何内容并返回状态 8074_0000

(OpcUa_BadTypeMismatch)。如果写入部分数组 (IndexRange)，则特性完全相同：待写入数组范围的所有维度必需完全位于最大数组大小的范围内。只有这样，才会写入数据并返回状态“良好”。
如果大小为“0”的数组至少写入一个维度，则该值在 CPU 数据中正确设置。对应 OPC_UA_ArrayBoundaries 结构中的元素“Lower”和“Upper”将进行相应调整（“Upper”=“Lower-1”）。

规则和特性

- 在以下方法中，输入参数和输出参数也可使用动态数组。
- 客户端访问数组时，避免因更改数组的当前大小引起不一致。在这种情况下，访问结果未定义。
  使用不间断指令 UMOVE_BLK 更改当前数组大小。
- 有别于 SIMATIC 数据类型 Array，OPC UA 中可以有“空”数组。使用 OPC_UA_ArrayBoundaries 将最高下标值 (Upper) 设置为小于最低下标值 (Lower)（例如 Upper = Lower-1），以此定义某个维度上元素数量为 0 的数组。

  如果在服务器接口（或客户端接口）中定义一个多维数组，且至少一个维度为“0”（空），则根据定义，该数组是一个空数组（所有维度均为 0）。
- 当前数组大小的值和数组数据的值必须一致。否则，将输出错误代码 803A_0000 (OpcUa_BadNotReadable)。
  - 一个维度中为当前数组大小设置的“Lower”值，还需设置为相应维度中数组数据范围中的下标下限值。
  - 为一个维度的当前数组大小设置的“Upper”值还必须处于该维度的数组数据范围内。
- 如果 OPC UA 客户端将动态数组写入服务器，则必须写入相同维度的数组 (ValueRank)。否则，将返回状态 8074_0000 (OpcUa_BadTypeMismatch)。

11.3.4.9 OPC UA XML 文件的规则

将已导出 OPC UA XML 文件导入到 S7-1500 CPU

导入来自 S7-1500 的 OPC UA XML 导出的服务器接口时，请注意以下信息。

说明

对于命名空间“http://www.siemens.com/simatic-s7-opcua”，导入被阻止

不能将命名空间为“http://www.siemens.com/simatic-s7-opcua”的服务器接口导入到 S7-1500 CPU，因为该命名空间为 S7-1500 CPU（标准 SIMATIC 服务器接口）预留，不可导入。

如果要导入命名空间为“http://www.siemens.com/simatic-s7-opcua”的服务器接口，请打开要导入的服务器接口（OPC UA XML 文件）并在相关位置更改命名空间。然后可以导入更改后的文件。

OPC UA XML 文件的完整性

OPC UA XML 文件用于说明服务器的地址空间。例如，在调整应用程序后，基于 OPC UA Companion 规范导入为服务器接口的这些文件，将加载到 S7-1500 CPU 中进行测试。

⚠警告

系统不会对导入的 OPC UA XML 文件进行检查

由于 STEP 7 不会检查这些文件的完整性，因此需确保这些 OPC UA XML 文件防止未经授权的篡改。

建议

对服务器的地址空间进行控制或调整时，为了将风险降至最低，可执行以下操作步骤：

1. 保护项目（项目导航：“安全设置 > 设置”(Security settings > Settings)）。
2. 对地址空间进行扩展或调整前，导出相应的服务器接口。
3. 修改该 OPC UA XML 文件。
4. 再次将该文件导入为服务器接口。

11.3.4.10 为引用命名空间创建服务器接口

配套规范和引用的命名空间

配套规范中定义了一系列 OPC UA 对象类型（以及其它定义）。这些对象类型是分别在命名空间中定义的，以确保对象类型名称（类型定义）的唯一性。

要在项目中使用配套规范，请创建该配套规范对象类型的实例。

为此，对象定义必须在 STEP 7 项目中可用。如果不可用，则必须导入对象定义。要导入命名空间的所有定义，请在 STEP 7 中为每个命名空间创建“引用命名空间”类型的服务器接口。

说明

EUROMAP 和 OPC 基金会成立联合工作组“OPC UA 塑料和橡胶机械”。

既有 EUROMAP 推荐标准 EUROMAP 77（注塑机和 MES 之间的数据交换）、82.1（温度控制设备）和 83（通用类型定义）等同于中立机构 OPC 基金会发布的标准 OPC 40077、40082-1 和 40083。不过，下面列出的示例仍使用之前的有效标识和引用。

示例 Euromap 77 (最新为 OPC 40077)

已为配套规范 Euromap 77 （最新为 OPC 40077）添加一个服务器接口。

该服务器接口使用 OPC UA DI 以及 Euromap 83 和 Euromap 77 在其相应命名空间中定义的对象类型。

因此，除了"配套规范"类型的服务器接口 Euromap 77 之外，还应在 STEP 7 中为以下命名空间分别创建"引用命名空间"类型的附加服务器接口。

- http://opcfoundation.org/UA/DI/
- http://www.euromap.org/euromap83/
- http://www.euromap.org/euromap77/

以下说明介绍了具体操作步骤。

为引用命名空间创建服务器接口

要为引用命名空间创建服务器接口，请执行以下操作：

1. 选择要作为 OPC UA 服务器使用的 CPU。
2. 单击"OPC UA 通信 > 服务器接口"(OPC UA communication > Server interfaces)。
3. 双击"添加新服务器接口"(Add new server interface)。

   STEP 7 (TIA) 现在会显示"添加新服务器接口"(Add new server interface) 对话框。

   新服务器接口的一般名称会输入到对话框中，例如"Server_Interface_1"。
4. 为新的服务器接口分配一个描述性名称。

   在本例中，选择名称"OPC.Ua.Di"或明确引用命名空间"http://opcfoundation.org/UA/DI/"的类似名称。

   必须先导入该命名空间。其中包含基本定义（例如 UAObjectType"DeviceType"）。
5. 单击"配套规范"(Companion specification) 按钮并选择"引用命名空间"(Reference namespace) 类型。
6. 单击"导入 XML 文件"(Import XML file) 字段旁边的三个点。
7. 选择包含"http://opcfoundation.org/UA/DI/"命名空间定义的 XML 文件。

   本例中选择"Opc.Ua.Di.NodeSet2.xml"文件。要下载该文件，请访问此处：

   Opc.Ua.Di.NodeSet2.xml (https://opcfoundation.org/UA/schemas/DI/)

   注：如果此文件无法导入，这可能是因为 TIA Portal 中支持的 OPC UA（"CORE"）模型版本不匹配设备 OPC UA（"DI"）模型版本。在这种情况下，选择其它 DI 模型版本（例如，之前的版本）。

下图显示了对话框及条目：

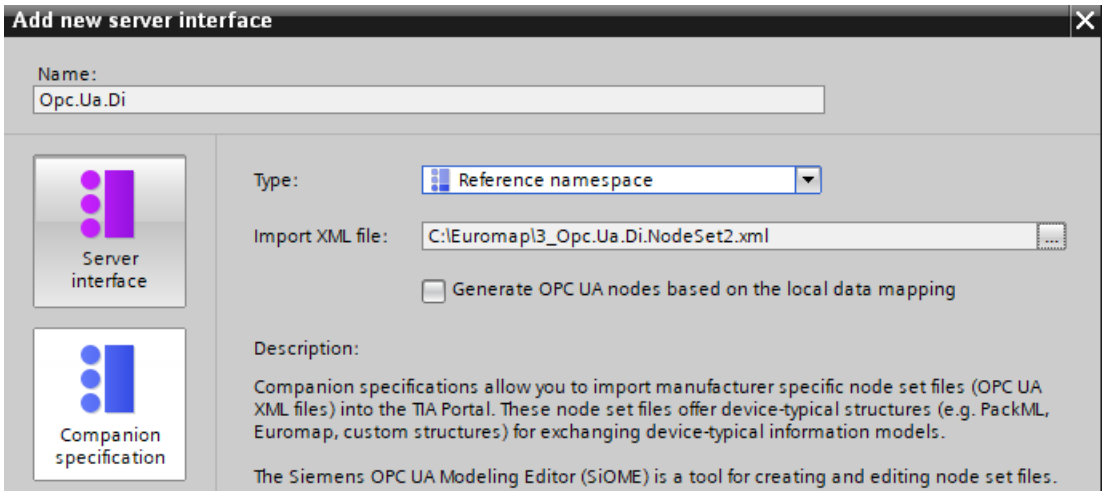


图 11-53 添加参考命名空间

8. 单击“确定”(OK)。

   STEP 7 (TIA) 现在会生成新的服务器接口。

服务器接口位于 STEP 7 (TIA Portal) 项目树的“OPC UA 通信 > 服务器接口 > 引用命名空间”(OPC UA Communication > Server interfaces > Reference namespace) 下方。

如果配套规范使用其它命名空间，则为每个命名空间添加新的服务器接口。

为 Euromap77 添加其它服务器接口

对于 Euromap 77，仍需要以下命名空间：

- http://www.euromap.org/euromap83/
- http://www.euromap.org/euromap77/

先为命名空间“http://www.euromap.org/euromap83/”添加一个服务器接口。

该命名空间包含 Euromap 77 的基本定义，因此需要先在此处使用。该命名空间的所有定义均包含在 XML 文件“Opc_Ua.EUROMAP83NodeSet2.xml”中，可从 Euromap 网站 (https://www.euromap.org/en/euromap83)下载此文件。

然后为命名空间“http://www.euromap.org/euromap77”添加一个服务器接口。该命名空间的所有定义均包含在 XML 文件“Opc_Ua.EUROMAP77.NodeSet2.xml”中，同样可从 Euromap 网站 (https://www.euromap.org/en/euromap77)下载此文件。

11.3.4.11 基于 FB 类型和 UDT 的本地数据映射生成 OPC UA 节点

自 TIA Portal V17 起，如果希望 OPC UA 客户端可以访问该 CPU 中 FB 或 UDT 内的实例数据，可自动分配这些实例数据。

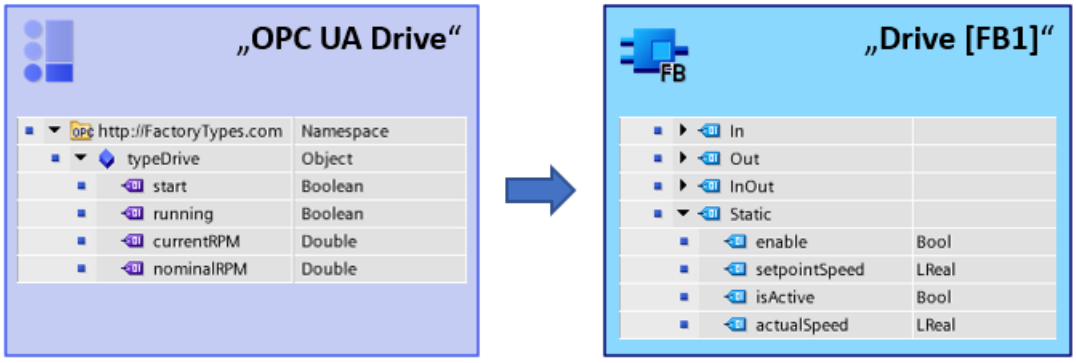
用户只需将 FB 类型或 UDT 映射到已导入的引用命名空间的适当 OPC UA 数据类型即可。基于 STEP 7 (TIA Portal) 中创建的这些映射，编译时在服务器接口中为每个 FB 实例或为每个 UDT 用途生成所需的节点。

如果用户扩展程序并添加更多 FB 实例或 UDT 用途，或者如果添加既有实例或予以删除，都无需为服务器接口的调整工作担忧：STEP 7 将在编译程序时自动调整服务器接口。

示例

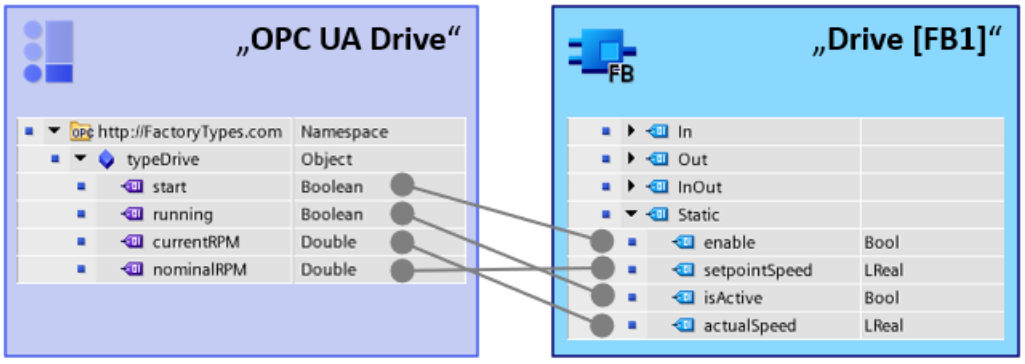
- 用户在 CPU 的用户程序中创建函数块 (FB)，并在 FB 接口的"静态"区域中定义构成此 FB"存储器"的参数。此参数的实例（值）将可由 OPC UA 客户端访问。
- 用户创建 OPC UA 数据类型（例如，通过 SiOME 创建）并采用与 FB 接口静态区域中参数的数据类型相对应的元素。元素的顺序无关紧要。之后，将引用节点集文件（引用命名空间）导入为一个引用命名空间。

下图显示的是元素的分配情况，其中比较了引用命名空间视图（服务器接口）和 OPC UA 元素视图（程序）。



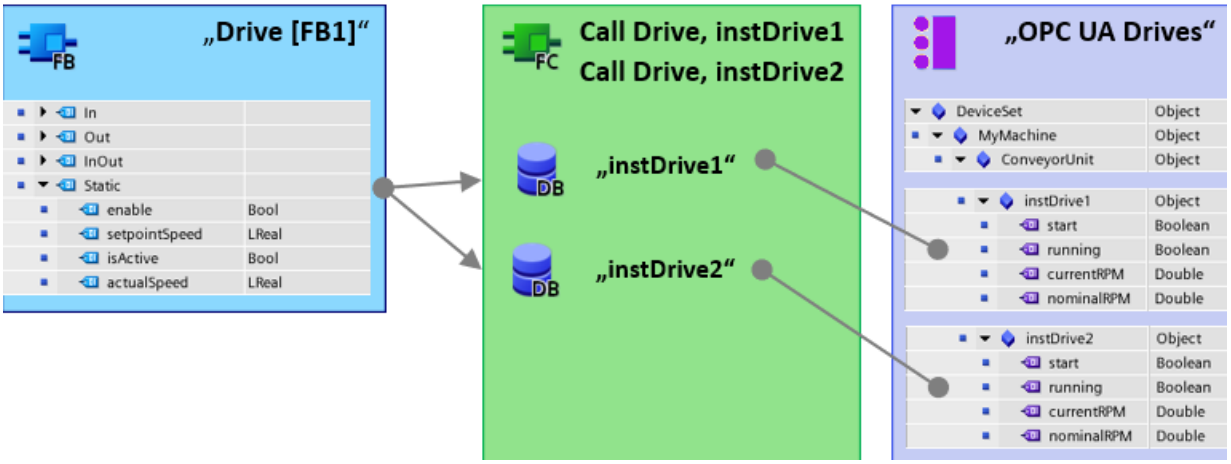
数据类型的映射（FB 接口 - OPC UA 接口）：原理

下图显示了 CPU 用户程序的元素分配与 OPC UA 服务器接口的元素分配。元素的顺序无需互相匹配。

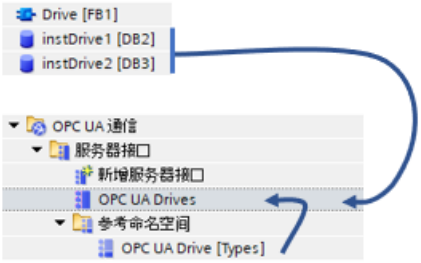


在服务器接口中自动生成 OPC UA 服务器实例：原理

下图显示了项目的编译。用户程序的示例也将在服务器接口中生成。



通过在 FB 类型信息/UDT 类型信息和 OPC UA 类型信息之间建立映射，STEP 7 能够在服务器接口中以节点形式创建程序中存在的所有实例。



规则

- 只有 FB 接口“静态”区域中的 FB 元素可映射到 OPC UA 类型描述。
- 在映射数据类型时，对象通常需选择同一个 FB 接口中或同一 UDT 中的 OPC UA 元素。不允许从不同的 FB 或 UDT 映射对象。

要求

- 所用的 FB 类型（在 FB 的“静态”区域中定义）必须组态为“OPC UA 可访问”(Accessible for OPC UA)。
- 所用的 UDT 必须组态为“OPC UA 可访问”(Accessible for OPC UA)。
- 节点集文件（XML 文件）中包含有 OPC UA 数据类型定义，而且该数据类型定义与用户程序中所定义的 FB 类型或 UDT 相匹配。
  使用“SiOME”工具创建节点集文件（西门子工业在线支持）。
- 用户程序及 FB 实例和 UDT 用途可用。

操作步骤

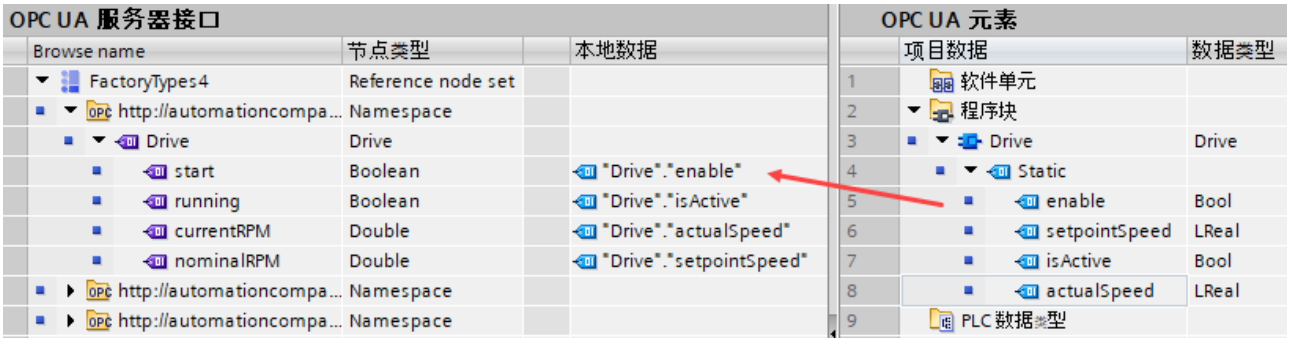
要将数据类型从引用命名空间映射到 FB 类型或 UDT 数据类型，按以下步骤操作：

1. 选择要作为 OPC UA 服务器使用的 CPU。
2. 将预先准备好且包含类型定义的节点集文件（XML 文件）导入为引用命名空间。
   - 在"添加新服务器接口"(Add new server interface) 对话框中，启用选项"基于本地数据映射生成 OPC UA 节点"(Generate OPC UA nodes based on the local data mapping)。

     只有在启用此选项后，才能通过将 FB 类型或 UDT 拖动到 OPC UA 类型描述来映射它们。
3. 双击刚生成的"引用命名空间"类型的服务器接口的图标。

   用于在 OPC UA 服务器接口和 OPC UA 元素之间进行映射的编辑器将打开。在编辑器的属性区域的"本地数据映射"(Mapping of local data) 区域中，已启用选项"基于本地数据映射生成OPC UA节点"(Generate OPC UA nodes based on the local data mapping)。否则，请立即启用该选项。

   在"接口名称"(Interface name) 字段中，调整要创建的服务器接口的名称。编译期间，会创建一个使用该名称的"配套规范"类型的新服务器接口。
4. 将既有 FB 类型或 UDT 分配到服务器接口的节点（引用命名空间），方法为：将 OPC UA 元素（编辑器右侧）拖到服务器接口的相应节点上（引用命名空间，"本地数据"(Local data) 列）。



5. 编译项目。

   编译后，新生成的实例节点位于生成的"配套规范"类型的服务器接口中。STEP 7 为每个背景数据块都创建一个对象。生成的元素将位于每个此等对象之下。

   同样地，STEP 7 也会为在实例化 UDT 时所创建的每个全局数据块创建一个对象。

   所生成的配套规范接口无法更改（不再支持：生成本地数据，导入配套规范）。该措施可对数据进行保护，防止手动扩展服务器接口并重新编译时，因覆盖而导致数据丢失。仅生成的配套规范接口的名称可修改。

   此外，在生成的配套规范接口中，TIA 项目的项目树中所创建的用于整理数据的组文件夹可用作"文件夹"（节点类型）。

创建用户程序及 FB 类型或 UDT

关于如何创建 FB 和 UDT 在此将不再赘述；就此目的，请参见有关创建用户程序的说明，举例而言，可声明块接口和声明 PLC 数据类型 (UDT)。

一致性检查

一致性检查（编辑器的"一致性检查"(Consistency check) 按钮）还将检查数据类型的映射并更新编辑器相应列中数据类型的显示。

11.3.4.12 使用服务器接口时组态限制的注意事项

使用 OPC UA 服务器接口时，必须遵循依据 S7-1500 CPU 性能等级的以下对象的限制：

- 服务器接口数
- OPC UA 节点数
- 加载对象数据量
- 如果方法已执行：服务器方法或服务器方法实例的数量

OPC UA 服务器接口和方法的组态限制

有关 OPC UA 服务器接口和方法的组态限制，请参见相应 CPU 的设备手册的技术规范。有关 CPU 的最新技术规范，敬请访问 Internet (https://support.industry.siemens.com/cs/ww/zh/ps/td)。

违反组态限制会导致出现错误消息。

11.3.5 在 OPC UA 服务器上提供方法

11.3.5.1 关于服务器方法的有用信息

提供用于服务器方法的用户程序

在 S7-1500 CPU（自固件版本 V2.5 起）的 OPC UA 服务器中，可以选择通过用户程序提供方法。例如，OPC UA 客户端可使用这些方法，通过 S7-1500 CPU 的方法调用启动生产作业。

OPC UA 方法是"远程过程调用"的实现，为不同通信节点之间的交互提供了有效机制。该机制提供作业确认和反馈值，因此用户无需再编程握手机制。

例如，使用 OPC UA 方法，可以持续传输数据，而无需触发位/握手或触发控制器上的特定操作。

OPC UA 方法的工作原理

通常，OPC UA 方法的工作原理与运行系统中由外部 OPC UA 客户端调用的受专有技术保护函数块的原理类似。

OPC UA 客户端仅"监视"已定义的输入和输出。函数块、方法或算法的内容对外部 OPC UA 客户端保持隐藏。OPC UA 客户端接收成功执行的反馈以及函数块（方法）返回的值，或者，如果执行不成功，则会收到错误消息。

作为程序员，对 OPC UA 方法运行的程序环境负有责任并具有完全控制权。

编程方法和运行行为的规则

- 确保 OPC UA 方法返回的值与 OPC UA 客户端提供的输入值一致。
- 遵守分配参数名称和结构的规则以及允许的数据类型（请参见 OPC UA 服务器指令的说明）。
- 运行期间的行为：对于每个实例，OPC UA 服务器均接受一次调用。在调用已由用户程序处理或已超时之前，方法实例不可用于其它 OPC UA 客户端。

实现用户程序（作为服务器方法）的基本步骤如下。

服务器方法的实现

用于实现服务器方法的程序（函数块）的结构如下：

1. **使用 OPC_UA_ServerMethodPre 查询服务器方法调用**

   先在用户程序中（即服务器方法中）调用"OPC_UA_ServerMethodPre"指令。

   该指令将执行以下任务：

   – 通过该指令询问 CPU 的 OPC UA 服务器是否已通过 OPC UA 客户端调用服务器方法。
   – 如果已调用方法，并且服务器方法具有输入参数，服务器方法现在会接收到输入参数。
     服务器方法的输入参数来自调用 OPC UA 客户端。

2. **编辑服务器方法**

   在这部分服务器方法中，用户提供实际用户程序。

   选项与其它任何用户程序中的选项相同（例如访问其它函数块或全局数据块）。

   如果服务器方法使用输入参数，则可使用这些参数。

   仅当 OPC UA 客户端已调用服务器方法时，才可执行服务器方法的这一部分。

   成功执行方法后，如果方法具有输出参数，需要设置服务器方法的输出参数。

3. **使用 OPC_UA_ServerMethodPost 响应服务器方法**

   要完成服务器方法，应调用"OPC_UA_ServerMethodPost"指令。

   使用参数通知"OPC_UA_ServerMethodPost"指令是否已处理用户程序。

   如果用户程序已成功执行，则会通过相关参数通知 OPC UA 服务器。OPC UA 服务器随后会将服务器方法的输出参数发送到 OPC UA 客户端。

无论用户程序是由"OPC_UA_ServerMethodPre"和"OPC_UA_ServerMethodPost"指令处理还是在下一个周期继续执行，始终以成对的形式调用这两个指令。

有关实现服务器方法的示例，请参见 STEP 7 在线帮助。

集成服务器方法

下图显示了 OPC UA 客户端 (A) 如何调用服务器方法“Cool”：

CPU 在循环用户程序 ⑥ 中执行服务器方法“Cool”的实例“Cool1”。

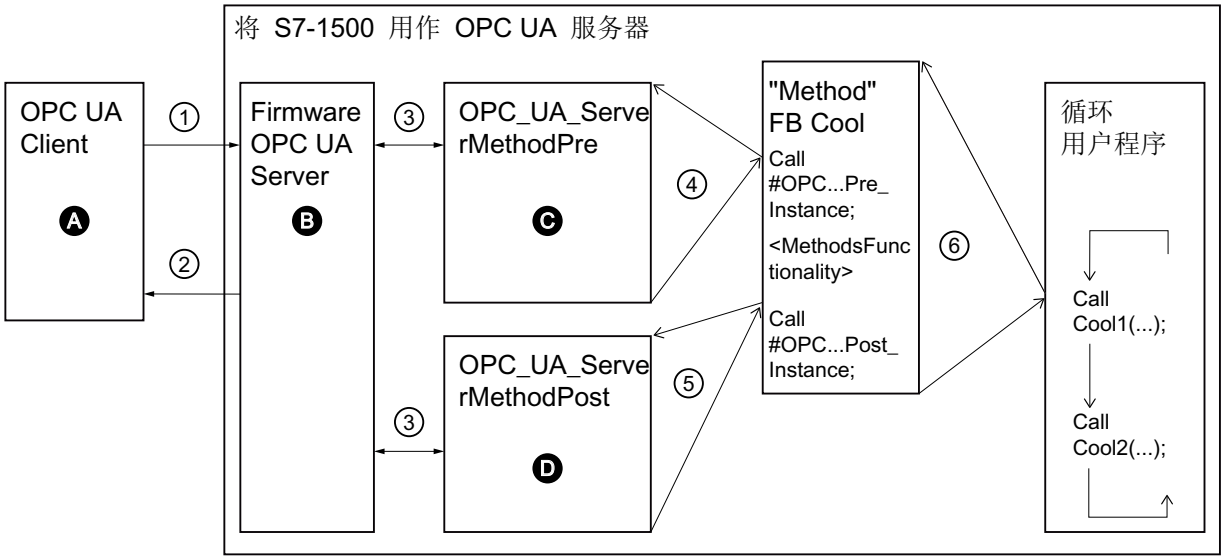
CPU 首先使用指令“OPC_UA_ServerMethodPre”查询 ④ OPC UA 客户端是否调用了服务器方法“Cool”①。

- 如果尚未调用服务器方法，则程序执行将通过 ④ 和 ⑥ 直接返回至循环用户程序。
  在“Cool1”之后，CPU 恢复循环用户程序。
- 如果已调用服务器方法，则该信息将通过 ④ 返回至服务器方法“Cool”。实际功能现在会在 Cool 服务器方法中执行，请参见图中的“<方法功能>”(<Method Functionality>)。

  然后服务器方法使用指令“OPC_UA_ServerMethodPost”⑤ 通知固件 (B) 该指令已执行 ③。

  固件通过 ② 将该信息返回至调用 OPC UA 客户端 (A)。

  在“Cool1”之后，CPU 恢复循环用户程序。



A 服务器方法的调用以及“完成”(Done) 信息（方法已完成）的管理

① 服务器方法的异步调用

② 调用方法的异步“完成”(Done) 信息（方法已完成）

B 等待 OPC UA 客户端调用，管理队列中的调用，将“完成”(Done) 信息从循环用户程序转发到 OPC UA 客户端

③ 在 OPC UA 服务器与用户程序的方法实例之间的数据传输

C 检查方法是否已调用。
如果已调用，将输入数据从 OPC UA 服务器转发到用户程序的方法实例，并为方法实例反馈方法已调用（“已调用”）

④ 同步调用指令 OPC_UA_ServerMethodPre（作为多重实例），说明来自 OPC UA 服务器的输入数据的存储区域。
返回值指示 OPC UA 客户端是否调用了方法。

⑤ 检查方法是已完成还是仍处于活动状态（“忙”）。

D 检查方法是否已完成。
如果已完成，则会将方法实例的输出数据转发到 OPC UA 服务器，并通知方法实例方法已完成。通知 OPC UA 服务器。

⑥ 使用所需实例和过程参数调用方法 FB（在本例中为：FB Cool）

图 11-54 示例：调用“Cool”服务器方法

关于服务器指令的信息

“指令 > 通信 > OPC UA 服务器”(Instructions > Communication > OPC UA > OPC UA server) 的帮助中详细介绍了“OPC_UA_ServerMethodPre”和“OPC_UA_ServerMethodPost”。

11.3.5.2 使用服务器方法的边界条件

支持的数据类型

提供服务器方法时，请遵循以下规则：

- 根据下图所示，指定数据类型（SIMATIC 数据类型 - OPC UA 数据类型）。系统不支持其它分配方式。

STEP 7 不会检查是否遵循该规则，因此也不会预防分配错误。用户需确保所做的选择和数据类型分配符合规则。

例如，所列出的数据类型，也可用作自定义服务器方法（UAMethod_InParameters 和 UAMethod_OutParameters）中输入和输出参数结构/数组/UDT 的元素。

SIMATIC 数据类型	OPC UA 数据类型性
BOOL	Boolean
SINT	SByte
INT	Int16
DINT	Int32
LINT	Int64
USINT	Byte
UINT	UInt16
UDINT	UInt32
ULINT	UInt64
REAL	Float
LREAL	Double
LDT	DateTime
WSTRING	String
DINT	Enumeration (Encoding Int32) 和所有派生的数据类型
所需的用户自定义数据类型（UDT，user-defined data type） 用户创建的自定义数据类型必须以“Union_”为前缀，如“Union_MyDatatype”。 在 UDT 中，第一个元素 (Selector) 的数据类型必须为“UDINT”。	UNION 和所有派生的数据类型
请参见“LocalizedText 和 ByteString 数据类型 (页 265)”	LocalizedText ByteString
GUID 请参见“将其它 OPC UA 数据类型用于配套规范 (页 267)”	GUID

支持的服务器方法数和参数数量

在用户程序中执行服务器方法时，可使用的方法数量取决于 CPU 类型。相关信息，请参见相应 CPU 的设备手册或 Internet (https://support.industry.siemens.com/cs/ww/zh/ps/td) 上的 CPU 最新技术规范。

超出时的错误消息

如果超出服务器方法的最大数量，则指令 OPC_UA_ServerMethodPre 或 OPC_UA_ServerMethodPost 将报告错误代码 0xB080_B000 (TooManyMethods)。

可使用服务器方法传输到 S7-1500 CPU 的数据量

最大 2097152 字节（= 2.1 兆字节 MaxMessageSize）。

使用带有嵌套数组的结构化数据类型

如果结构化数据类型 (Struct/UDT) 中包含一个数组，则 OPC UA 服务器无法提供该数组的长度信息。

如果将该结构用作服务器方法的输入或输出参数，则需确保调用该方式时使用的嵌套数组长度正确。

如果长度错误，则该方法调用失败且错误代码为“BadInvalidArgument”。

11.3.6 提供 OPC UA 服务器报警

11.3.6.1 有关报警的实用信息

通过报警可以快速检测自动化系统中的过程控制错误，并准确定位和清除这些错误。这有助于大幅缩短工厂停机时间。OPC UA 信息模型“Alarms & Conditions”提供不限平台的标准化消息处理方式。

自固件版本 V2.9 起，S7-1500 CPU 的 OPC UA 支持 OPC UA 信息模型“报警和条件”。通过这种方式，OPC UA 服务器可支持访问控制报警。

下面部分将介绍 SIMATIC 中可用的哪些报警类型是 OPC UA 服务器的 OPC UA 接口所支持的。

下文中还介绍了 S7-1500 CPU 中 OPC UA 服务器的报警和条件组态；OPC UA 与报警和条件模型的架构；以及与 CPU 报警系统的 SIMATIC 控制器报警相比，使用 OPC UA 服务器地址空间的报警时，需考虑的特殊事项。

将报警转换到 OPC UA Alarms & Conditions 的基础

Alarms and Conditions 信息模型在“OPC 10000-9: OPC Unified Architecture Part 9: Alarms & Conditions”规范中指定。

SIMATIC 中的控制器报警

S7-1500 CPU 的 OPC UA 服务器支持下列控制器报警，这些报警均可供 S7-1500 CPU 使用。用户可按常规方式对这些报警进行组态和编程，无需额外考虑在 OPC UA 客户端上使用这些报警的规则。

OPC UA Alarms and Conditions 所带来的额外优势是，这些报警类型不仅可通过 HMI 设备、Web 浏览器、CPU 显示屏或 TIA Portal 显示，而且也可在支持 OPC UA 报警和条件的所有 OPC UA 客户端中显示。

- ProDiag 的 PLC 监控报警
  只需执行几个组态步骤，即可在程序中快速集成监控功能，且无需更改程序代码。由于仅监控单个的操作数且无需额外编程，因此监控的组态与 TIA Portal 的编程语言无关。
- 系统诊断报警
  与配置相关的模块事件以 CPU 硬件配置的方式提供，并可通过连接的显示设备加以评估。这些事件只能在报警编辑器中查看，不能编辑。
- 程序报警（Program_Alarm 指令）
  为报告程序同步事件，程序报警一次分配到一个块中。这些报警在程序编辑器中创建，在报警编辑器 (TIA Portal) 中编辑。
- GRAPH 报警
  对于 GRAPH 函数块，用户还可启用报警；例如用于互锁、监控和 GRAPH 警告（步时间监视）。

有关报警类型的重要信息

在报警行为的差异上，下列特性有重要意义：

- 报警是否有状态（例如，报警当前是处于进入还是离开状态，是否有相应的时间戳）？
- 报警是否需要确认？

如果这些特性均不适用，也就是说报警没有任何状态且无需确认，那么报警的作用仅是就已经发生的事件提供相关信息（“发后即忘”）。具体是将报警缓存起来以备后用，还是仅用于显示目的，这取决于接收报警的设备。

报警类别决定确认行为

本部分将介绍程序报警的设置选项。用户还可为系统诊断报警和 PLC 监控报警（例如，ProDiag 监控设置）设置报警行为 - 有关详细信息，请参见链接中的更多信息。

有关程序报警的设置选项，可在报警编辑器中找到（在项目树中双击“PLC 监控和报警”(PLC supervisions and alarms)，并选择“报警”(Alarms) 选项卡）。

对于 S7-1500 CPU，用户可在此通过报警类别设置是否需要确认报警。除了确认行为之外，在创建新报警类别时，还可定义此报警类别下报警的默认优先级。

报警是否有状态可在报警类型或其它位置通过“仅供参考”(Information only) 选项设置；选择此选项后，将按照“发后即忘”的方式处理报警。

这里就报警编辑器中的设置提供了一个示例，其中包括不同的报警类别（项目树中的"PLC 监控和报警"(PLC supervisions and alarms)）：

- 第一行"Program_Alarm"：不需要确认，仅供参考（"发后即忘"）。
- 第二行"Program_Alarm_1"：需要确认并有状态，也就是说，其中包含信息，可指示报警是处于进入还是离开状态。
- 第三行"Program_Alarm_2"：不需要确认但有状态，也就是说，其中包含信息，可指示报警是处于进入还是离开状态。

报警类型

名称	类型	ID	位置	报警文本	信息文本	报警类别	确认	仅供参考
Program_Alarm	PLC 报警		AlarmType	myText	myINFO	No Acknowledgement	☐	☑
Program_Alarm_1	PLC 报警		AlarmType			Acknowledgement	☑	☐
Program_Alarm_2	PLC 报警		AlarmType			No Acknowledgement	☐	☐

报警在 TIA Portal 中的显示方式

在运行时，用户可选择在 TIA Portal 中显示报警：报警画面的位置就在报警编辑器之下（"诊断"(Diagnostics) 选项卡 >"报警画面"(Alarm display) 选项卡）。

以下内容适用于状态和确认行为：

- 在单击"当前报警"(Current alarms) 按钮时，将显示最近进入、离开或确认的报警。这里仅显示带有状态且需要确认的报警。用户也可在此视图中对需要确认的报警（蓝色字体）进行确认，确认时可使用快捷菜单，也可使用"确认"(Acknowledge) 按钮。
- 如要了解时间顺序（例如，报警进入，得到确认，然后离开），则需要单击"报警归档"(Alarm archive) 按钮。属于此报警的三个事件将逐个列出，但仅限于此视图。有关当前状态的信息，只能通过"当前报警"(Current alarm) 视图查看。
- 信息报告（具有"仅供参考"(Information only) 特性的报警）仅会显示在"报警归档"(Alarm archive) 视图中。由于这些报警仅会触发一次并且不会加以缓存，因此它们不会出现在"当前报警"(Current alarms) 视图中。
- PLC 监控也会显示在报警画面中。
- 系统报警通常都属于"无需确认"(No Acknowledgement) 报警类别，且会选中"仅供参考"(Information only) 选项。这些报警会记录在 CPU 的诊断缓冲区中，允许就一个有限时间段进行系统报警序列的分析。相比之下，诊断缓冲区中另外也记录的操作状态变化则具有状态，也就是说，会反映 CPU 是否或者何时进入 STOP 状态，以及是否或者何时再退出此状态（例如，进入 RUN 状态）。此信息通过状态"进入/离开"(incoming/outgoing) 显示。

由 OPC UA 服务器提供控制器报警

当 OPC UA 客户端需要接收 S7-1500 CPU 的报警时，此客户端需要订阅 OPC UA 事件 (MonitoredEventItems)。

就此目的，S7-1500 CPU 的 OPC UA 服务器地址空间包含相应的节点，这些节点会通知事件的发生（"事件通知者"(Event-Notifiers)），并会创建订阅，以便 OPC UA 客户端能够接收报警。

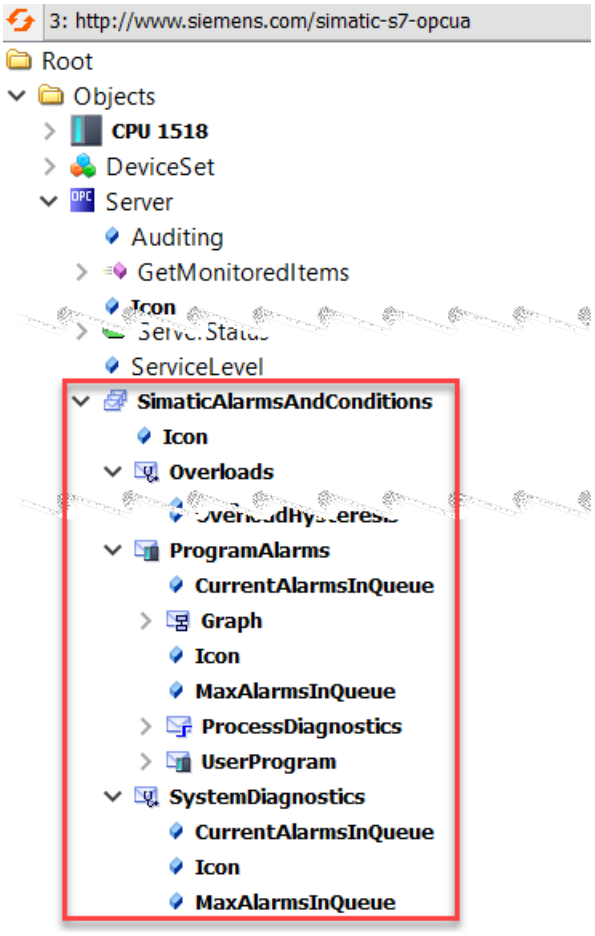
为了完整起见，这里也提及服务器地址空间中所含的同样用于此目的的其它类型定义，即，"类型"(Types) 下的节点。借助"BaseEventType"和"ConditionType"下的类型定义，可确保 SIMATIC 报警所用的字段也将在 OPC UA 服务器中提供。

在激活 OPC UA 报警和条件（硬件配置中的 CPU 特性）之后，S7-1500 CPU 的 OPC UA 地址空间也会进而反映出不同报警类型（控制器报警），如上文所述：

- ProcessDiagnostics
  对应于 ProDiag 的 PLC 监控报警
- SystemDiagnostics
  对应于系统诊断报警
- UserProgram
  对应于程序报警
- Graph
  对应于 GRAPH 报警

通过为订阅选择节点，用户就决定了 OPC UA 客户端将接收的报警类型。例如，"服务器"(Server) 节点支持接收所有报警，而"UserProgram"节点仅可接收程序报警。

有关 OPC UA 模型"报警和条件"的详情将在下一选择中提供，而具体上与"过载"(Overloads) 节点相关的信息，请参见：处理 OPC UA 报警和条件的存储器限制 (页 304)。



有关报警类型的更多信息

这里将不会更多地介绍控制器报警的概念和组态选项。有关报警组态、报警显示和相关指令（如"Progam_Alarm"）的信息，请参见 STEP 7 在线帮助。

11.3.6.2 OPC UA 事件

这里将就 OPC UA 中报警处理的基本概念进行展开说明，其中也将介绍“事件”的概念。这里将沿用 OPC UA 规范的各个部分中所使用的术语。

事件的特性

在 OPC UA 服务器的地址模型中，自 CPU 固件版本 V2.9 起，用户不仅可选择通过节点访问 PLC 变量（读、写）以及选择使用不同的方法，同时还可通过节点接收事件和报警。按 OPC UA 术语，这些都称为“事件”。

事件包含事件文本（消息）、时间戳（时间）和事件源（源节点）。

服务器事件所提供的具体信息取决于事件的类型。OPC UA 在其规范的第 5 部分定义了 BaseEventType (Information Model)。

其它提供不同报警行为的事件类型均由 BaseEventType 派生而来。不同事件类型的类型信息在 OPC UA 服务器的地址空间中可见（“类型”(Types) 文件夹）。其适用场合的示例包括，“Conditions”和“Alarms”的事件类型，这些将在下一部分中介绍。

OPC UA 规范定义了就 BaseEventType 和派生的 EventTypes 而言，事件的哪些特性（字段）是强制的，哪些是可选的。

下图显示了 BaseEventType 的层级结构。

下面部分显示了专用 EventTypes 是如何从作为派生层级根源的 BaseEventType 而得来的。借助 SIMATIC 特定的派生可带来的优势包括，对于在 SIMATIC 中通过报警形式提供的和在 HMI 设备上显示的信息，也可由 OPC UA 客户端在 OPC UA 服务器的地址空间中进行订阅。

事件本身不以地址空间中的节点形式提供。事件的触发只能源自于那些会就事件的发生进行通知的节点或对象（即，具有“事件通知者”(Event-Notifiers) 特性的节点或对象）。这些节点通常也称为事件信号传送对象。只有具备此特性的节点可指定为一个订阅中的 EventMonitoredItem，进而支持在客户端中接收相应事件。

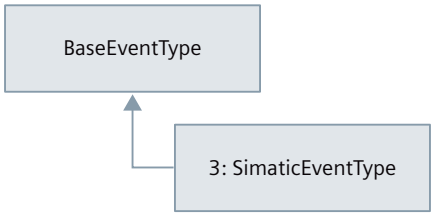
可触发 S7-1500 CPU 事件的节点示例包括：“服务器”(Server)、服务器下的“SimaticAlarmsAndConditions”对象，以及再下方的三个对象，即 ProcessDiagnostics、SystemDiagnostics 和 UserProgram。对于这些对象，将在 CPU 的 OPC UA 服务器的地址空间中设置“EventNotifier”属性。

SimaticEventType 的定义

下图显示了直接由 BaseEventType 派生而来的类型“SimaticEventType”。

BaseEventType 是 OPC UA 事件的基本类型定义。

可基于 BaseEventType 直接或间接定义 OPC UA 的所有事件类型。



“SimaticEventType”类型在 SIMATIC 命名空间中定义 (http://www.siemens.com/simatic-s7-opcua)。

SimaticEventType 具有 BaseEventType 的所有特性，同时也具有作为 SIMATIC 报警现场结构映像的特殊特性。

SimaticEventType 事件字段描述

对于类型为"仅供参考"(Information only) 的报警，下表包含 SimaticEventType 字段的信息。对 OPC UA 而言为可选且 CPU 的 OPC UA 服务器也不使用的字段将予以忽略。有关各字段的总体描述，另请参见规范 OPC 10000-5：OPC 统一架构，第 5 部分：信息模型（版本 1.04）。

BrowsePath	DataType	说明
EventId	ByteString	事件的唯一事件 ID
EventType	NodeId	事件类型的节点 ID
Time	UtcTime	事件的时间戳（事件发生）
ReceiveTime	UtcTime	OPC UA 事件生成时的时间戳。
Message	LocalizedText	事件文本
Severity	UInt16	报警的优先级从 SIMATIC (0..16)，一直到对于 OPC UA 的范围 1..1000，请参见下表。 优先级表示的是，就事件而言需要收到响应的紧急程度。
3:AdditionalText_01	LocalizedText	其它可选文本 1
...	...	...
3:AdditionalText_09	LocalizedText	其它可选文本 9
3:AssociatedValue_01	3:SimaticAssociatedAlarmValue	可选关联值 1（不适用于系统诊断）
...	...	...
3:AssociatedValue_10	3:SimaticAssociatedAlarmValue	可选关联值 10（不适用于系统诊断）
3:InfoText	LocalizedText	信息文本
3:ID	UInt16	报警编号 - 由系统分配的在 CPU 中唯一的编号 (ID)，可以识别报警。
3:DisplayClass	UInt16	显示类别（供 HMI 设备使用。决定特定 HMI 设备上显示的事件。
3:GroupID	UInt8	供统一确认用的报警确认组。

优先级分配 (SIMATIC) - 严重程度 (OPC UA)

下表显示的是，在 SIMATIC 环境中可为报警分配的 17 个优先级与在 S7-1500 CPU 中 OPC UA 服务器的 1000 级 Severity 之间的映射关系。

具体分配取决于制造商。其它设备可能使用不用的分配方式。

OPC 范围	优先级 0..16 (SIMATIC)	严重程度 1..1000 (OPC UA)
高 (667 – 1 000)	16	1000
	15	930
	14	860
	13	790
	12	720
中 (334 – 666)	11	650

OPC 范围	优先级 0..16 (SIMATIC)	严重程度 1..1000 (OPC UA)
	10	600
	9	550
	8	500
	7	450
	6	400
	5	350
低 (1 – 333)	4	300
	3	225
	2	150
	1	75
	0	1

11.3.6.3 OPC UA 条件和 OPC UA 报警

在前面部分就事件所做说明的基础上，下文将更进一步，介绍 OPC UA Conditions 和 OPC UA Alarms 的基本概念。同样，这里也将沿用 OPC UA 规范的各个部分中所使用的术语。

Conditions 的特性

理解的先决条件是 OPC UA 中"Events"的概念。

在 OPC UA 中，如果事件报警对象在能够发出 Events之外还可提供状态信息，那么就涉及到 Conditions。Conditions 代表的是系统或系统组件的一种状态。基本状态为"enabled"和"disabled"，同时也可以定义其它状态。

反过来，相关的 OPC UA 客户端也会通过事件 (Condition Events) 这一途径获得状态变更通知。

Condition 的一个示例是状态信息，例如，设备需要维护。

Alarms 的特性

但 ConditionType 的特性不足以完全映射 OPC UA 服务器中 SIMATIC 报警的特征。

在从 BaseEventType 派生而来的 ConditionType 基础上，OPC UA 可定义进一步派生的事件类型，例如 AcknowledgeableConditionType 和 AlarmConditionType。

AcknowledgeableConditionType 可为 ConditionType 特性补充"是否可确认"的特征 (AckedState)。

而 AlarmConditionType 又进一步在 ConditionType 和 AcknowledgeableConditionType 特性基础上增添了"ActiveState"特征。按 SIMATIC 中的表述方法，这是一个进入的报警。ActiveState 传递信号，指示 Condition 所反映的情形当前已存在或已发生。

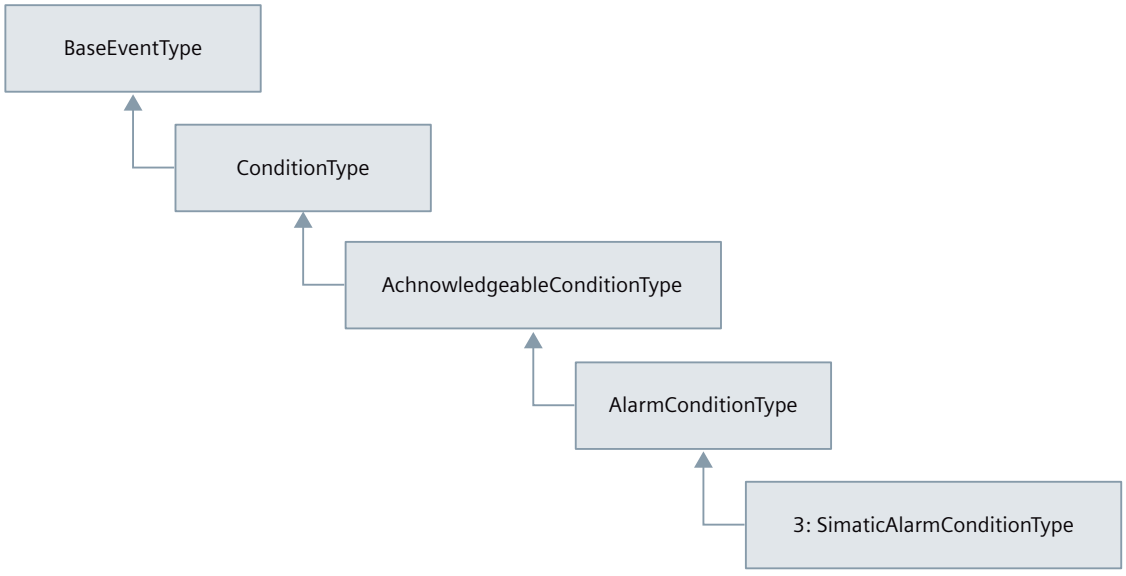
示例：温度已超出限值。如果未设置"ActiveState"，则反映此条件的情形将不再存在，而这通常称为"常态"。按 SIMATIC 中的表述方法，这对应的是离开的报警。

OPC UA 中也会定义其它状态，例如 SilenceState 和 ShelvingState，但这些状态不涉及与 SIMATIC 报警系统的映射，因此这里将不再做更多说明。

SimaticAlarmConditionType 从 AlarmConditionType 衍生而来，其中包含用于映射 SIMATIC 消息的状态和确认情况的所有事件字段。

SimaticAlarmConditionType 的定义

下图显示了“SimaticAlarmConditionType”类型的事件是如何在 OPC UA 的“BaseEventType”基础上一步步扩展而来的。



SimaticAlarmConditionType 事件字段说明

下表就带有状态且支持确认的报警提供了有关 SimaticAlarmConditionType 的各个字段的信息，这些将补充诸如 SimaticEventType 等的事件字段。对 OPC UA 而言为可选且 CPU 的 OPC UA 服务器也不使用的字段将予以忽略。有关各字段的说明，另请参见规范 OPC 10000-9：OPC 统一架构，第 9 部分：报警和条件（版本 1.04）。

BrowsePath	DataType	说明
ConditionClassId	NodeId	可以是 SystemConditionClassType、ProcessConditionClassType 或 BaseConditionClassType 的节点 ID
ConditionClassName	LocalizedText	ConditionClassId 的显示名称
Retain	布尔型	指示对于 OPC UA 客户端而言报警仍然相关（在报警仍处于未决状态并且尚未确认时设置）。
Comment	LocalizedText	• 通过“AddComment”或“Acknowledge”方法输入的最新注释。 • 在服务器重启之后和在未输入任何注释的情况下，为 ZERO。
Comment.SourceTimestamp	UtcTime	注释字段上次更改的时间戳
AckedState	LocalizedText	“已确认”(Acknowledged) 或“未确认”(Unacknowledged)

BrowsePath	DataType	说明
AckedState.Id	布尔型	在已确认时设置
AckedState.TransitionTime	UtcTime	报警得到确认的时间。 如果未确认或不可确认，则为 ZERO。
ActiveState	LocalizedText	“激活”(Active) 或“未激活”(Inactive)
ActiveState.Id	布尔型	在“激活”(Active) 时设置

11.3.6.4 激活报警和条件

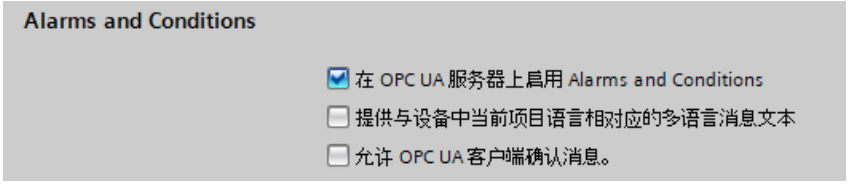
要求

- S7-1500 CPU 固件版本 V2.9 或更高版本。
- 根据许可证规范与 CPU 属性中的设置，购买了 OPC UA 运行系统许可证
- 已启用“PLC 中的中央报警管理”参数（CPU 属性中的“PLC 消息”区域）。

操作步骤

要通过 OPC UA 报警和条件激活报警，按以下步骤操作：

1. 在 CPU 属性中，转到“OPC UA > 服务器 > 常规”(OPC UA > Server > General) 区域。
2. 选择“为 OPC UA 服务器启用报警和条件”(Enable alarms and conditions on the OPC UA server) 选项。
   可触发事件的相应类型和对象只有在此选项已激活时，才会在地址空间显示出来。
3. 必要时，还需激活选项“允许 OPC UA 客户端确认消息”(Allow message acknowledgment by OPC UA client)。
   此时，任何连接的 OPC UA 客户端都可通过“确认”(Acknowledge) 方法确认需要确认的报警。



建议：激活诊断“远程 OPC UA 客户端请求失败”

如果 OPC UA 服务器的存储空间不足，则无法生成 OPC UA 报警；OPC UA 客户端可能发生消息丢失的情况。

因此，应激活诊断“远程 OPC UA 客户端请求失败”(Requests of a remote OPC UA client failed)，对该状态进行诊断：CPU 属性 > OPC UA > 服务器 > 诊断 (OPC UA > Server > Diagnostics)。

除此之外，还应激活选项“消息量较大时汇总诊断”(Summarize diagnostics in case of high message volume)

一旦存储空间充足，OPC UA 客户端应调用 ConditionRefresh 条件，接收该报警系统的当前状态。

更多信息

有关 OPC UA 报警和条件支持的方法的信息，请参见“OPC UA 报警和条件支持的方法 (页 300)”部分。

有关远程客户端请求失败的信息，请参见“远程客户端请求失败 (页 312)”部分。

11.3.6.5 订阅 OPC UA 服务器的事件

通过“服务器”节点订阅所有事件

OPC UA 服务器通过“Server”节点及其下级节点提供事件。当 OPC UA 客户端订阅“Server”节点时，这些客户端将接收此 OPC UA 服务器的所有事件和报警。

“Server”节点位于“对象”(Objects) 文件夹的“Root”之下。

OPC UA 服务器会将 OPC UA 客户端所使用的事件类型通知给这些客户端（在地址空间中的“Root > Types > EventTypes”之下）。

事件过滤选项

OPC UA 客户端可选择并仅订阅“服务器”(Server) 节点下的特定节点，进而仅订阅特定的事件类型，例如，仅订阅“UserProgram”节点。借此可减少从 OPC UA 服务器到程序报警的事件数量。

另一种过滤方式是选择事件字段，即 OPC UA 术语中所谓的“Select 语句”。

这意味着在订阅过程中，OPC UA 客户端除了事件报警对象之外还对事件字段进行选择（例如，“UserProgram”节点）。用户可通过浏览相应字段名称的方式选择事件字段。

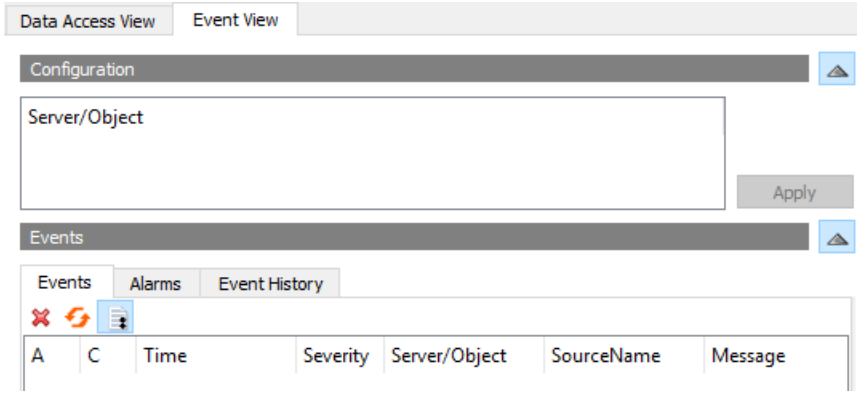
OPC UA 另外也定义所谓的“Where 语句”。事件过滤器中的 Where 语句用于进一步限定 OPC UA 服务器为所选对象提供的事件数量，例如，可按严重程度范围进行过滤。

示例客户端 UaExpert

UaExpert OPC UA 客户端示例显示了如何能通过订阅方式接收 OPC UA 服务器的事件。以下是有关所示事件/报警的最重要的信息：

- 事件视图是在数据访问视图之外的一个单独的事件视图。
- “Configuration”区域包含所选的事件信号传送对象，以及 Select 语句的字段。目前，在 UaExpert 中不支持组态 Where 语句。
- 在“Events”区域中，“Events”选项卡：对应于“报警归档”(Alarm archive) 按钮已激活的 TIA 报警视图；其中也将显示离开的报警和“仅供参考”(Information only) 类别的事件，因为 UaExpert 会在后台对其进行缓冲并支持进行显示。这些事件在“报警”(Alarms) 选项卡中不可见。
- 在“Events”字段中，“Alarms”选项卡：对应于“当前报警”(Current alarms) 按钮已激活的 TIA 报警画面；其中将显示报警及其状态，例如，“激活”(active)（对应于“进入”(incoming)），并且这些报警也可通过快捷菜单进行确认。离开的报警将不会再在此视图中显示。

在事件区域的各个列中提供一系列事件字段，例如，事件文本 (Message) 以及报警是否已确认 (A=Acknowledged)。



CPU 的 OPC UA 服务器针对报警显示提供的特殊功能

下面再一次汇总了 OPC UA 报警和条件的报警画面在反映当前状态上所提供的特殊功能。

主题	说明
注释	通过 OPC UA，用户可通过"AddComment"途径或"Acknowledge"方法为报警添加注释。此注释在服务器重启后将不复存在。
而未决报警在服务器重启后将不会丢失	OPC UA 服务器支持"ConditionRefresh"方法，借此可在下载新数据块之后（需要重启服务器并重新建立连接）或在其它此类情况下，将系统当前状态提供给 OPC UA 客户端。

11.3.6.6 报警相关值的处理方式

用户可指定 SIMATIC 报警占位符。通过这些占位符，可将最多 10 个相关值（SD_1 到 SD_10）集成到报警文本中。占位符也可以是特定的文本列表条目。

使用带有占位符的报警时，需遵循以下规则：

- 仅在系统诊断报警或安全事件报警中，才会在报警中自动插入代表相应值的占位符。对于其它类别的报警（如，程序报警），系统不会对这些值的占位符进行解析。OPC UA 客户端必需对这些报警进行解析。
- 引用文本列表的占位符由 CPU 进行解析（格式示例：%t#<文本列表的名称>）。

通过 UaExpert 分配值和占位符的示例

1. 请确保 UaExpert 组态中所需的所有字段均已选中。

   请注意，所有不需要的字段都会产生通信负载。因此，应避免以下示例中所示的全部选择。

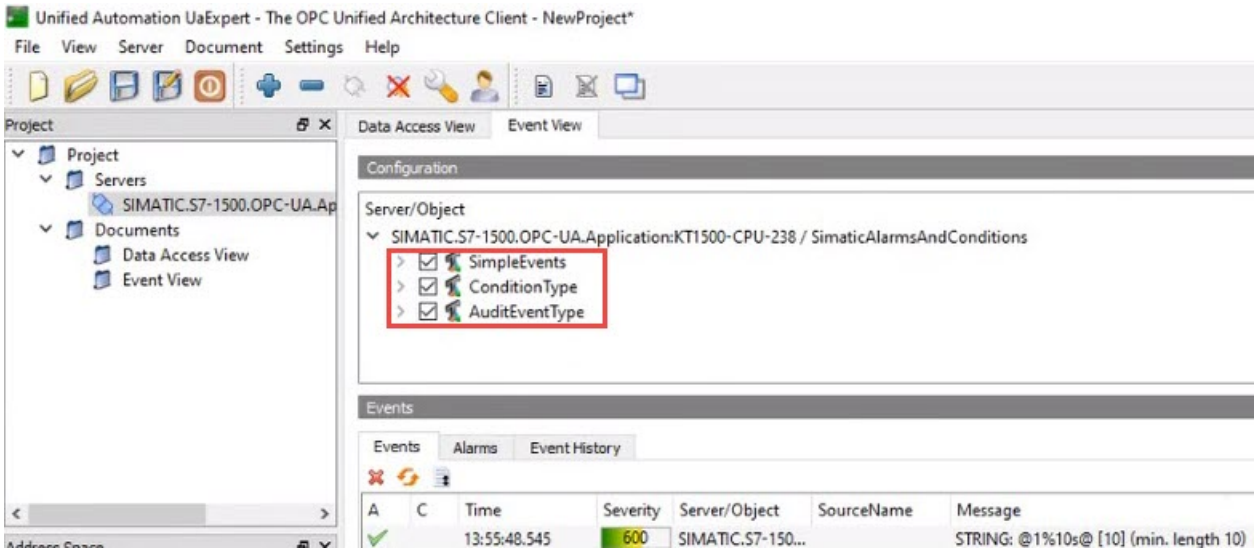


图 11-55 ServerObject-Configuration_neutral120

2. 在 UaExpert 的“Events”选项卡中，选择集成有相关值的报警。
   这些待集成到报警中的值，将显示在报警的“Details”区域中。
   示例：“AssociatedValue_01”分配给 SD_1（格式：@1% ...@）。
   有关相关值的格式说明，请参见 TIA Portal 信息系统（例如，搜索“相关值示例”）。

支持简单数据类型作为关联值

字段类型"AssociatedValue_01"到 ..._10 为 Union 类型且限制为一些简单类型。OPC UA 数据类型为 SimaticAssociatedAlarmValue。支持以下简单数据类型：

3:AssociatedValue_01	SimaticAssociatedAlarmValue
Switch Field	2
Boolean	True
3:AssociatedValue_02	SimaticAssociatedAlarmValue
Switch Field	6
SByte	123
3:AssociatedValue_03	SimaticAssociatedAlarmValue
Switch Field	3
Int16	1234
3:AssociatedValue_04	SimaticAssociatedAlarmValue
Switch Field	4
Int32	12345
3:AssociatedValue_05	SimaticAssociatedAlarmValue
Switch Field	7
Byte	123
3:AssociatedValue_06	SimaticAssociatedAlarmValue
Switch Field	8
UInt16	1234
3:AssociatedValue_07	SimaticAssociatedAlarmValue
Switch Field	11
Float	1
3:AssociatedValue_08	SimaticAssociatedAlarmValue
Switch Field	12
Double	2
3:AssociatedValue_09	SimaticAssociatedAlarmValue
Switch Field	9
UInt32	12345
3:AssociatedValue_10	SimaticAssociatedAlarmValue
Switch Field	13
String	HelloWorld

图 11-56 UnionDataTypes_neutral120

映射 SIMATIC 数据类型

SIMATIC 数据类型 => OPC UA 数据类型的映射方式如下所示：

SD_1 到 SD_10 支持的数据类型	**映射到 OPC UA 的数据类型**
BOOL	Boolean
BBOOL	Boolean
BYTE	Byte
CHAR	Byte
SINT	SByte
USINT	Byte
WORD	UInt16
WChar	UInt16
INT	Int16
UINT	UInt16
DWORD	Uint32

SD_1 到 SD_10 支持的数据类型	**映射到 OPC UA 的数据类型**
DINT	Int32
UDINT	Uint32
REAL	Float
LREAL	Double
String	String
WString	String

11.3.6.7 同时接收多种语言的报警

报警以默认语言或参考语言从 OPC UA 服务器发送到 OPC UA 客户端。在 CPU 参数的"多语言支持"(Multilingual support) 区域中，可设置要用于传输报警的已下载项目语言。

自 S7-1500 CPU 版本 4.0 起，可选择连接的 OPC UA 客户端的语言：

- 建立会话时，客户端可使用参数"LocaleIds []"请求一种或多种激活项目语言的消息文本；请参见 OPC 10000-4：服务，ActivateSession 服务参数。

  通过这种方式，只能从 CPU 的 OPC UA 服务器请求消息文本；该参数对地址空间中的其它本地化文本没有影响。

  要求：已启用"提供与设备中激活项目语言相对应的多语言消息文本"(Provide multilingual message texts corresponding to the active project languages in the device) 选项。
- 例如，一个客户端可同时请求所有三种激活项目语言的消息文本，以在中央服务器上收集报警，以便掌握不同语言的人员可对其进行评估。

要求

- S7-1500 CPU 固件版本 4.0 及更高版本
- 对于多语言文本，OPC UA 客户端支持 LocaleId"mul"和"qst"。
- 已启用"提供与设备中激活项目语言相对应的多语言消息文本"(Provide multilingual message texts corresponding to the active project languages in the device) 选项（CPU 属性的"Alarms and Conditions"区域）。

LocaleId"mul"和"qst"的作用

OPC UA 规范（OPC 10000-3：UA 第 3 部分：地址空间模型）描述了内置的数据类型"LocalizedText"，此类型用于不同语言的文本。

数据类型 LocalizedText 的结构元素定义如下：

名称	类型	描述
LocalizedText	Structure	
Locale	LocaleId	语言标识符（语言代码）<language>-<Country/Region>，例如"en-US" 由 IETF RFC 5646 指定
文本	String	本地化文本（特定语言的翻译）

LocaleId ="mul"

当 OPC UA 客户端请求带有 LocaleId“mul”的 LocalizedText 时，会在 CPU 上接收所有已下载项目语言的消息文本作为 JSON 文本元素，其中包括区域设置/文本对的数组。

在以下示例中，文本以两种语言指定。空格和断点不会被传输；其被添加到此处的目的是为了方便阅读：

```
{
 "t": [
 ["de-DE","mein Text"],
 ["en-US","my text"]
 ]
}
```

得到的文本：

```
de-DE “mein Text”
en-US “my text”
```

LocaleId =“qst”

“qst”不是一个缩写，而仅仅是 LocaleId 的定义字符串，除了内容之外，其还包含替换文本，就像“mul”的情况一样。

当 OPC UA 客户端请求带有 LocaleId“qst”的 LocalizedText 时，会在 CPU 上接收所有已下载的项目语言的消息文本。除了 LocaleId 数组之外，JSON 文本元素还有一个包含替换文本定义的部分（替换部分）。客户端必须执行消息文本中提供的替换。

两种语言的示例文本；“t”代表“text”，“r”代表“replacement”：

```
{
 "t": [
 ["de-DE","mein Text @1@ /2/"],
 ["en-US","my text @1@ /2/"]
 ],
 "r": [
["@1@", "myCompany"]
["/2/", 1.2345]
 ]
}
```

得到的文本：

```
de-DE “mein Text myCompany 1,2345”
en-US “my text myCompany 1.2345”
```

因为替换部分中的十进制数 1.2345 未格式化为字符串，因此其可用不同的语言正确书写（在 en-US 中使用小数点，在 de-DE 中使用小数逗号）- 前提是此功能已在客户端实现。

规则

报警文本由服务端按照如下规则组合，并提供给客户端：

- 为简化多语言消息文本的处理，S7-1500 的服务器只允许使用 LocaleId“qst”。

  如果请求带有“qst”的多语言消息文本，并且文本中有占位符（关联值），则会在替换部分提供占位符。如果文本中没有占位符，则文本将作为 LocaleId“mul”返回，不包含替换部分。

  如果仅返回一种语言（例如，因为只有一种激活的语言或与所请求的语言对齐导致只有一种语言）并且文本中没有占位符，则消息文本不会以 JSON 形式返回，而是直接以适当语言的本地化文本形式返回。
- 客户端可检索服务器提供的所有语言的报警，或者仅检索其中某些语言的报警。客户端通过 LocaleId 数组定义选择，该数组遵循 LocaleId“qst”。如果没有其它 LocaleId 遵循 LocaleId“qst”，则服务器将提供所有可用的语言。
- 替换部分不会返回多语言替换文本。可使用文本列表实现多语言替换测试，请参见“AUTOHOTSPOT”。

读取支持的语言

OPC UA 服务器在节点“服务器功能 > LocaleIdArray”(Server Capabilities > LocaleIdArray) 下发布支持的语言。

11.3.6.8 OPC UA 报警和条件支持的方法

举例而言，OPC UA 规范第 9 部分 (OPC 10000-9: Alarms & Conditions) 定义了如何借助 OPC UA 服务器让 OPC UA 客户端能够对状态变化做出响应的方法。

下文将介绍 S7-1500 CPU 的 OPC UA 服务器支持的这些方法及其特殊功能。

要求

欲使用报警和条件功能的相应方法，需要满足以下各项：

- 报警和条件已激活
- 对于“Acknowledge”方法，必须在服务器一侧允许由 OPC UA 客户端确认报警。

OPC UA 报警和条件支持的方法

下文将简要介绍各个方法，以及因实施 S7-1500 CPU 的 OPC UA 服务器而带来的特殊功能和限制。

各方法在类型空间中可见。

上文列出的 OPC UA 规范包含一般说明。

此概述表下方给出了有关各个方法的详细说明。

方法	说明
Acknowledge	此方法用于确认以 EventId 唯一标识的报警对象。
ConditionRefresh	此方法用于请求对所有报警对象进行更新（用 SIMATIC 语言表述：更新所有未决报警）。订阅之下所有受监视的项都将更新。 对于 CPU 中 OPC UA 服务器的未决报警对象，其同步情况将在诸如以下之类的情形下加以指示： • 首次连接或恢复连接（通信中断后） • HMI 设备操作员屏幕的屏幕切换
AddComment	此方法用于为报警对象添加注释。

调用“Acknowledge”和“AddComment”方法

在 OPC UA 中调用方法时将用到 MethodId 和 ObjectId。

对于报警对象，ObjectId 是报警对象实例的节点 ID。

由于 Simatic 报警和条件的地址模型不提供报警对象的实例，OPC UA 规范在此情况下规定，OPC UA 客户端使用 ConditionId 来作为 ObjectId。

有关如何通过事件过滤器的 SelectClause 中的 SimpleAttributeOperands 来确定 ConditionId 的更多信息，另请参见 OPC UA 规范第 9 部分 (OPC 10000-9: Alarms & Conditions)：

Name	Type	Description
SimpleAttributeOperand		
typeId	NodeId	*NodeId* of the *ConditionType Node*
browsePath[]	QualifiedName	empty
attributeId	IntegerId	Id of the *NodeId Attribute*

Acknowledge

Acknowledge 方法 (MethodId:i=9111) 有以下参数：

参数	数据类型	说明
[in] EventId	ByteString	EventId 用于标识特定的事件通知。 只有 AckedState.Id 字段具有值"False"的事件可通过"Acknowledge"方法确认。
[in] comment	LocalizedText	操作员或其他人员就确认给出的注释文本。 另请参见"AddComment"方法的补充说明。

方法结果代码

结果代码	说明
Good	方法已成功执行。
BadNotSupported	方法无法调用，因为由 OPC UA 客户端确认报警和条件的选项已在 OPC UA 的 CPU 特性中遭到禁用。
BadConditionBranchAlreadyAcked	确认已经完成。
BadNodeIdUnknown	方法调用时使用的 ConditionId 有误（参见 ObjectId 说明）。
BadEventIdUnknown	方法调用时使用的 EventId 有误。

ConditionRefresh

ConditionRefresh 方法 (MethodId:i=3875) 有以下参数：

参数	数据类型	说明
[in] SubscriptionId	Uint32	有待更新的订阅的 SubscriptionId。

方法结果代码

结果代码	说明
Bad_SubscriptionIdInvalid	SubscriptinId 无效。
Bad_RefreshInProgress	"ConditionRefresh"当前正在运行。
Bad_UserAccessDenied	"ConditionRefresh"方法运行所在的会话背景有误。这意味着此订阅属于另一个会话。

说明

ConditionRefresh2 方法

ConditionRefresh2 方法可在订阅中专门同步一个受监视项 (MonitoredItem)，而 S7-1500 CPU 的 OPC UA 不支持此方法。在这种情况下，OPC UA 服务器将返回结果代码"Bad_MethodInvalid"。转而使用方法"ConditionRefresh"。

AddComment

用户可以为 SimaticAlarmConditionType 类型的 Alarms- 对象添加注释，因为 OPC UA Alarms and Conditions 强制要求支持注释。

注释保存在“Comment”事件字段。

以下时间戳事件字段属于注释：

- “Comment.SourceTimestamp”，注释传送到 CPU 的时间
- “Time”，修改 Alarms 对象的时间

在调用“AddComment”方法时，“Time”和“Comment.SourceTimestamp”相同。

CPU 的 OPC UA 服务器针对报警和条件注释提供的特殊功能

AddComment 方法 (MethodId:i=9029) 有以下参数：

参数	数据类型	说明
[in] EventId	ByteString	EventId 用于标识做状态报告之用的事件通知。
[in] comment	LocalizedText	用于注释指定 Alarms 对象的文本。

方法结果代码

结果代码	说明
Good	方法已成功执行。
BadNodeIdUnknown	方法调用时使用的 ConditionId 有误（参见“Acknowlege”和“AddComment”方法调用说明）。
BadEventIdUnknown	方法调用时使用的 EventId 有误。

CPU 的 OPC UA 服务器针对报警和条件注释提供的特殊功能

用户可以通过 AddComment 方法为"SimaticAlarmConditionType"类型的报警对象添加注释。在调用 Acknowledge 方法时也将设置注释。"AddComment"方法可多次调用。

- 注释保存在"Comment"事件字段。"Comment.SourceTimestamp"指示**注释**上一次设置的时间。
- "Time"时间戳标记的是，**报警对象上一次的修改时间**。

在调用"AddComment"方法时，"Time"和"Comment.SourceTimestamp"相同。

在调用"Acknowledge"方法时，两个时间戳可能不同，因为确认不是同步进行的。

支持注释是 OPC UA 报警和条件的强制要求。SIMATIC 报警系统没有相应报警注释的信息。因此，一些特殊功能必须加以考虑：

- 只有一个注释：
  某报警对象只有一个注释，因此在有多个方法连续进行调用时，既有注释始终都会受到覆盖。
- 使用寿命和时间戳：
  注释仅存储在当前报警对象中。如果报警对象不复存在（例如，在服务器重启之后），相应的注释也将同样消失。相应的"Comment"和"Comment.SourceTimestamp"事件字段将受到复位（归零）。
  "Time"事件字段也将设置，就像是方法调用"AddComment"从未存在过一般。示例：如果对未确认的 Alarms 报警对象添加注释，"Time"事件字段将收到此注释变更的时间。在服务器重启后，"Time"事件字段不会显示注释设置的时间，而是会显示相应 Event 到达的时间。

11.3.6.9 处理 OPC UA 报警和条件的存储器限制

S7-1500 CPU 的 OPC UA 服务器根据产品的不同而对"报警和条件"功能有各异的有限存储器容量（参见 CPU 规范）。

供有两个存储器池，分别存储不同类别的报警：

- 仅适用于 ProgramAlarms 的存储器池（对应于与程序相关的报警源（生产者），例如基于 Program_Alarm、ProDiag、Graph 的程序报警）
- 仅适用于 SystemDiagnostics 的存储器池（对应于系统诊断报警）

在不利的条件下（例如，报警激增），CPU 无法将所有来自 SIMATIC 报警区域的所有未决报警（ProgramAlarms 或 SystemDiagnostics）提供给 OPC UA 报警和条件系统。但此时报警将不会丢失。

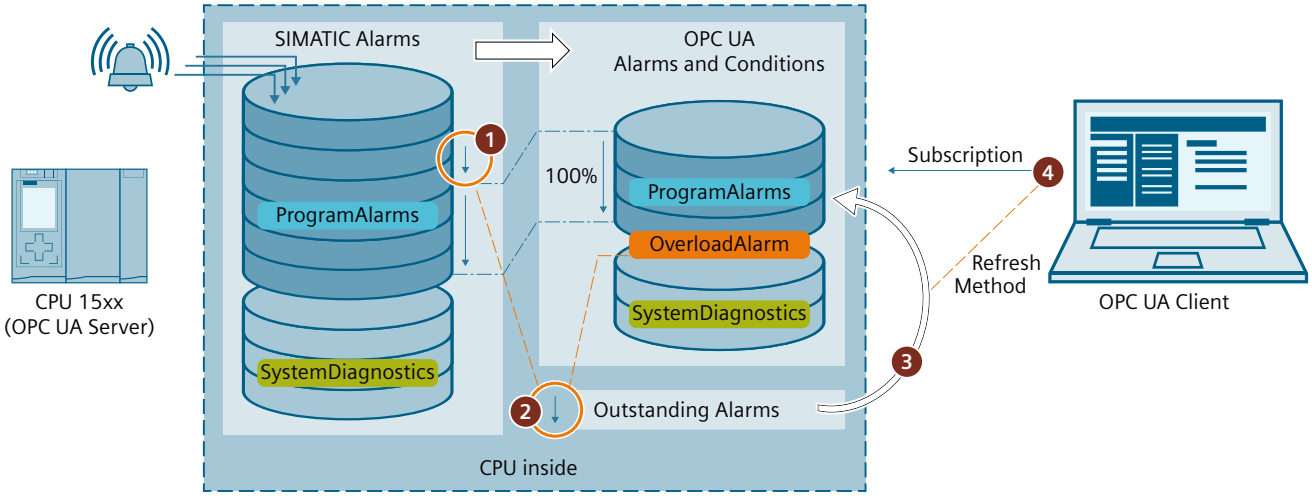
用户可以在用户程序中就此过载事件做出响应。根据具体应用，用户可使用"ConditionRefresh"方法来将"未能进入 OPC UA 报警和条件系统"的报警再提供给 OPC UA 报警和条件系统。

要求

- 报警和条件已激活
- 事件订阅已在 OPC UA 客户端中设置

原理

下图显示了一个简化的过程，即，会将 ProgramAlarms 临时存储下来，并另寻时间来再次提供给 OPC UA 报警和条件系统。说明中提到的节点在以下地址模型图片中可见。



① 活动报警的数量过多，无法通过 OPC UA 报警和条件访问全部报警

② 过载报警 (Overloads) 已触发。过载报警在发生以下情况之前保持激活：

- 对于 OPC UA 报警和条件系统，没有更多报警处于未决状态 (OutstandingProgramAlarms = 0)；
- OPC UA 报警和条件系统的报警数量 < 已清除滞后的 OPC UA 报警数量最大值 (= MaxAlarmsInQueue - OverloadHysteresis)

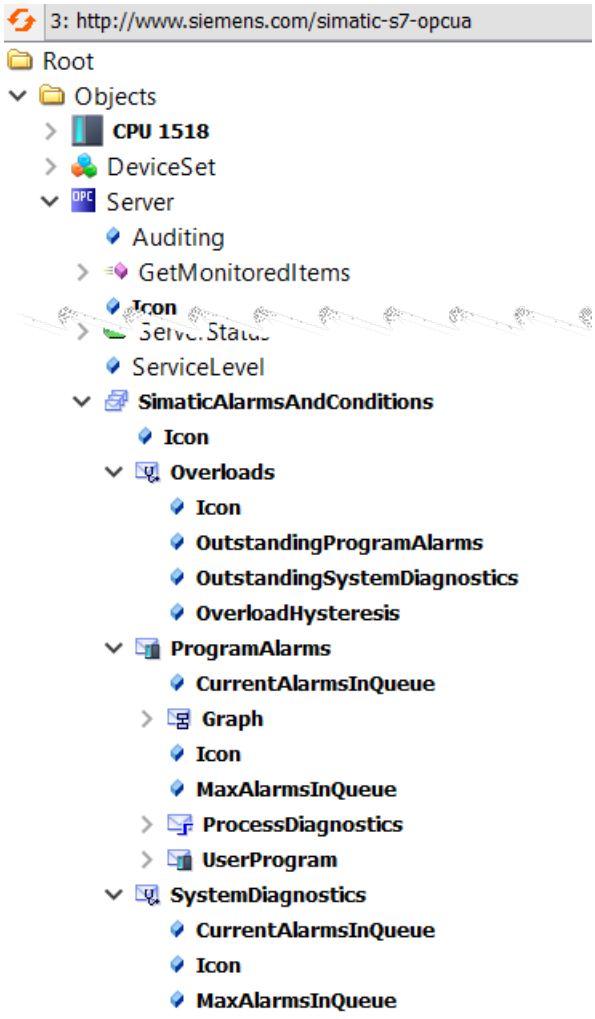
因过载情况而在 OPC UA 报警和条件系统中不可用的报警由 CPU 作为“OutstandingAlarms”进行缓冲。

③ 在 OPC UA 客户端执行 ConditionRefresh 方法时，不仅相关订阅的所有报警对象都将同步，而且 OPC UA 报警和条件的未确认报警 (OutstandingAlarms) 也将传送到报警和条件存储区中（但前提是未达到报警的最大数量）。“最早”的报警将最先传送。在此之后，这些报警的每个订阅（不仅限于调用 ConditionRefresh 方法的 OPC UA 客户端）都将收到已传送的报警。

④ OPC UA 客户端通过“过载”(Overloads) 节点的信息控制未决报警的处理。

报警和条件的地址模型

下图显示了 OPC UA 报警和条件地址模型的节点。



特殊功能

- 在未决报警转出或得到确认后，将不再经由 ConditionRefresh 方法进入 OCP UA 报警和条件系统区域。于是，它们将对 OPC UA 报警和条件“不可见”，进而也无法由所连的 OPC UA 客户端获取。这会影响报警进行过程的统计评估以及其它类似方面。
- 为避免在报警数量围绕最大值上下波动时致使过载报警出现较高的报警频率，触发报警的限值要高于取消报警的限值：此差值显示在“OverloadHysteresis”节点中。
  示例：最大报警数量：200，OverloadHysteresis：3。
  过载报警的数量在达到 200 时就开始触发，但只有在下降到 197 以下时才会取消。如果报警数量再次增加，仍需超过 200 才会触发报警。

11.3.7 使用诊断选项

11.3.7.1 OPC UA 服务器诊断

OPC UA 服务器在线诊断

S7-1500 CPU OPC UA 服务器可通过标准 OPC UA 客户端（如 UaExpert）进行在线诊断。

诊断信息分为以下几部分：

- 服务器诊断
- 会话诊断：
- 订阅诊断

举例来说，在服务器的地址空间中，以下节点提供诊断信息：

- **ServerDiagnosticsSummary**：服务器诊断汇总
  - CurrentSessionCount：活动会话数量
  - SecurityRejectedSessionCount：因客户端与服务器之间的端点安全设置不匹配而被拒绝的会话数
- **SessionsDiagnosticsSummary**：会话诊断汇总
  - ActualSessionTimeout：设置会话在连接断开等情况下的持续时间。
- **SubscriptionsDiagnosticsArray**：为每个会话的每个订阅包含一个元素的数组

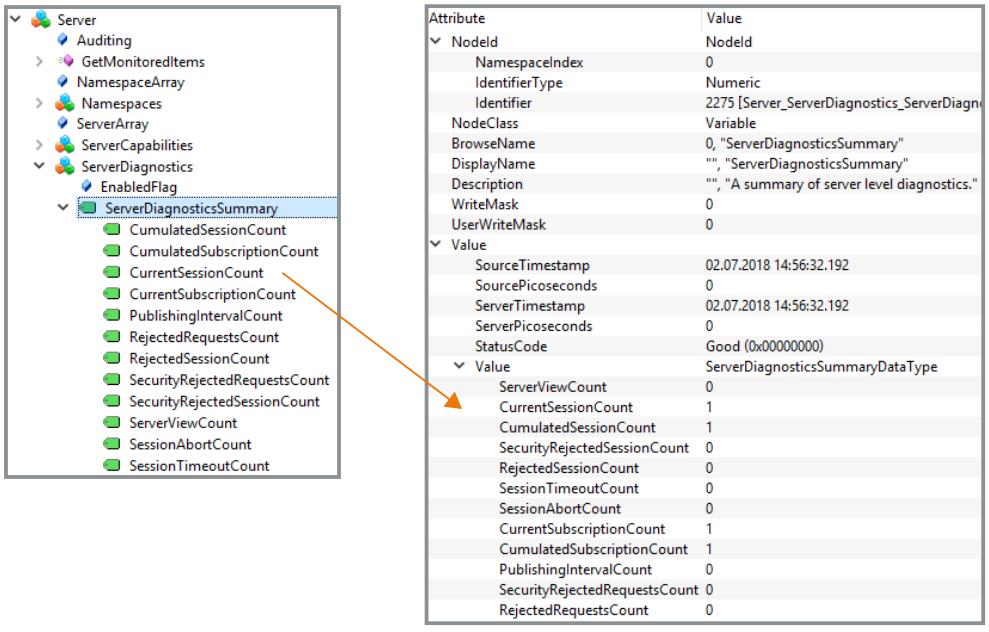


图 11-57 服务器诊断

SessionsDiagnosticsSummary 节点还显示在会话中访问服务器的客户端应用程序的特性。

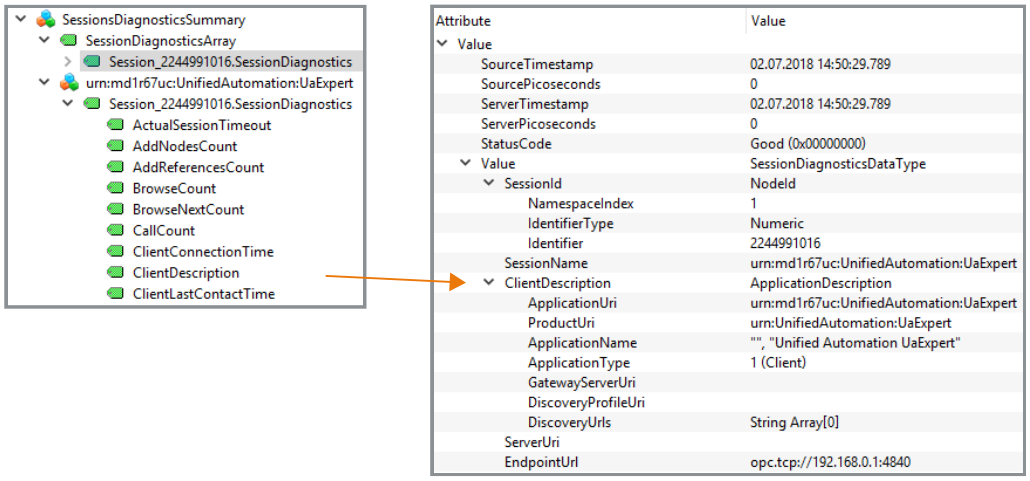


图 11-58 客户端应用程序属性会话诊断

客户端与服务器间的连接诊断

要诊断客户端中程序运行期间的连接状态，请使用以下指令：

OPC_UA_ConnectionGetStatus：读取连接状态。

11.3.7.2 在程序中运行 OPC UA 服务器诊断

在 STEP 7 (TIA Portal) V18 及以上版本中，可通过访问 S7-1500 CPU（固件版本 V3.0 及以上版本）内 OPC UA 地址空间中的节点，评估程序待诊断的内容。

工作原理

在 CPU 的本地地址空间中，包含很多 CPU 的 OPC UA 服务器用于存储数据和状态的节点。通过“OPC_UA_ReadList”指令，可访问相关信息并在用户程序中 进行评估。

示例：“ServerState”是 CPU 中的一个地址空间，其中包含有服务器的状态值或状态转换值（运行、关闭、失败等等）。

该指令并不是一个客户端指令，而是一个读取本地 OPC UA 地址空间节点的指令。此时，需使用特殊的规则和要求。

更多信息

有关调用“OPC_UA_Readlist”指令进行诊断的更多信息，请参见 TIA Portal 帮助中的“通过 OPC_UA_Readlist 诊断 OPC UA 服务器”主题。

11.3.7.3 服务器状态转换诊断

关于服务器状态的信息

OPC UA 服务器的状态发生变化时，S7-1500 CPU 固件版本 V2.8 及以上版本会在诊断缓冲区中创建一个条目。

诊断缓冲区显示新的状态。

同时显示状态变化的原因，例如下载到 CPU、POWER OFF - POWER ON 转换，来自伙伴（客户端）的用户程序指令或服务请求。

要求

在 CPU 的 OPC UA 属性中，选择“OPC UA 服务器状态改变”(Change of OPC UA server status) 选项。

说明

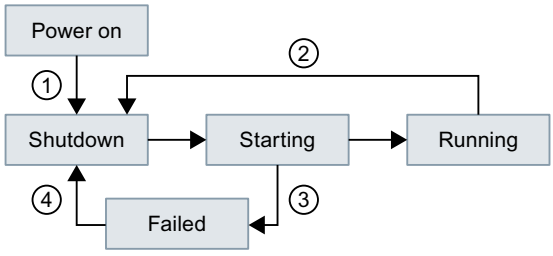
选中此选项后，CPU 也会在启动后使设置为最低优先级的安全策略进入诊断缓冲区。

示例

如果 CPU 的 OPC UA 服务器因为下载过程而关闭，然后使用有效的新组态启动，则诊断缓冲区显示新的服务器状态，例如“关闭 => 启动 => 运行”。

如果 OPC UA 服务器因为下载过程而关闭，并且服务器因为类型字典过大而无法启动，则诊断缓冲区最后显示状态“已失败”（“关闭 => 启动 => 已失败”）。

服务器状态和状态转换



①、④ 如果 OPC UA 相关数据可能受到影响，则上电或加载到 RUN 状态。

② OPC UA 服务器停用时加载硬件配置。服务器仍然关闭。
OPC UA 服务器激活且 OPC UA 数据错误时加载硬件配置（例如因结构过多导致类型字典变得过大）。在这种情况下，服务器无法启动（参见 ③）。

③ OPC UA 服务器因组态故障等问题无法启动。

图 11-59 服务器状态和状态转换

服务器状态说明

下面介绍了 OPC UA 服务器可呈现的各个状态。

服务器状态	说明
关闭	初始状态 • 上电后 • OPC UA 服务器激活或停用时加载硬件配置后。 • 加载 OPC UA 相关数据后
启动	服务器中的 OPC UA 地址空间已初始化。
正在运行	OPC UA 服务器运行（OPC UA 服务器的正常生产状态）。
已失败	错误状态。OPC UA 服务器因组态故障等问题无法启动。

11.3.7.4 会话状态转换诊断

关于会话状态的信息

OPC UA 会话的状态发生变化时，S7-1500 CPU 固件版本 V2.8 及以上版本会在诊断缓冲区中创建一个条目。

诊断缓冲区显示新的状态。也将显示相应的会话 ID。

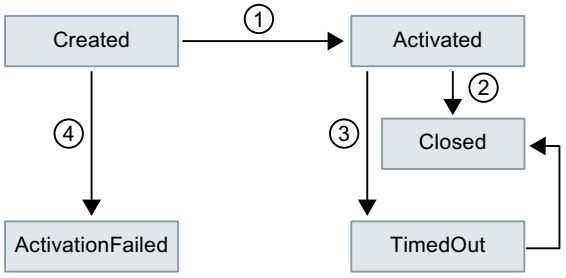
要求

已在 CPU 的 OPC UA 属性中选择"会话状态改变"(Change of session states) 选项（OPC UA > 服务器 > 诊断）。

示例

连接建立时客户端传输的认证数据不正确（例如密码不正确）。"ActivationFailed"会话的新状态以及相应的会话 ID 会进入诊断缓冲区。

订阅状态和状态转换



① 客户端连接到服务器、使用正确的认证数据登录（正确的凭证）。
② 客户端正确关闭连接。
③ 客户端不再发送消息；会话以超时结束。
④ 客户端连接到服务器、使用不正确的认证数据登录。

图 11-60 会话状态和状态转换

11.3.7.5 检查安全事件

如果 CPU 诊断在 OPC UA 通信期间检测到安全事件，可使该事件进入诊断缓冲区。

要求

- 固件版本为 V2.8 及以上版本的 S7-1500 CPU
- “检查安全事件”(Check for security events) 选项已激活（CPU 属性中的“OPC UA > 服务器 > 诊断”(OPC UA > Server > Diagnostics)）。

诊断中检测到安全事件

S7-1500 CPU 对以下 OPC UA 相关的安全事件执行诊断：

- 客户端证书无效（例如语法或语义错误、签名错误、当前日期不在有效期内）
- 用户名/密码登录失败（数据已停用或不正确）
- 客户端要使用特定的安全策略或特定的消息安全模式；服务器不支持该安全策略或请求的安全模式。
- 客户端未按照规范（OPC UA 规范）建立连接（例如未预期的 SecureChannelID/SessionID/客户端 Nonce）

示例

如果试图破坏通信（例如通过会话拦截、中间人攻击等），服务器会通过分析检测到此情况。

11.3.7.6 远程客户端请求失败

固件版本为 V2.8 及以上版本的 S7-1500 CPU 会在诊断缓冲区中为以下事件创建条目：

- 不良客户端请求（不正确使用）
- 出现服务错误
- 超出 OPC UA 服务器的 CPU 特定上限

错误客户端请求示例

例如，当客户端寻址一个不存在的节点（变量）或请求不存在的资源时，则会发生请求错误。

此时，导致错误的相应服务以及相应会话 ID 都会进入诊断缓冲区。

服务故障

如果服务自身发生故障，服务器会返回 ServiceFault。此时，状态代码（不良...）以及相应会话 ID 都会进入诊断缓冲区。

超出限值示例

如果服务请求超出 CPU 特定的限值，例如会话数、监视项数目、订阅数等，该诊断会进入诊断缓冲区，与消息共同指示所超出的限值。

例外：如果汇总诊断时消息频繁出现，则引发该错误的限值不会进入诊断缓冲区。您会收到已超出支持的组态限值的常规信息。

导致错误的服务的可能条目

根据使用的客户端应用程序，从客户的角度来看，可通过不同方式触发对服务器的请求，例如，可通过具有图形用户界面的在线工具触发，也可以通过客户端程序中的指令触发。

OPC UA 采用面向服务的架构，遵循请求-响应范例，因此相应的客户端应用程序会将请求转换为 OPC UA 中定义的服务请求。

这些服务的名称按照其用途来定义和分组，另请参见 opcfoundation.org。

如果未正确使用，则作为导致错误的服务，可在诊断缓冲区中准确找到这些服务的名称及相应的会话 ID。

下表列出了 OPC UA 提供的服务。

发现服务集

FindServers

GetEndpoints

会话服务集

CreateSession

ActivateSession

CloseSession

Cancel

视图服务集
Browse
BrowseNext
TranslateBrowsePathsToNodeIds
RegisterNodes
UnregisterNodes

属性服务集
Write
Read

方法服务集
调用

监视项服集集
CreateMonitoredItems
ModifyMonitoredItems
DeleteMonitoredItems
SetMonitoringMode
SetTriggering

订阅服务集
CreateSubscription
ModifySubscription
DeleteSubscriptions
转移订阅
Publish
Republish
SetPublishingMode

11.3.7.7 订阅诊断

有关订阅的信息

订阅状态发生变化时，固件版本为 V2.8 及以上版本的 S7-1500 CPU 可在在诊断缓冲区中创建一个条目。

诊断缓冲区会显示新状态；但以下状态除外："KeepAlive"。

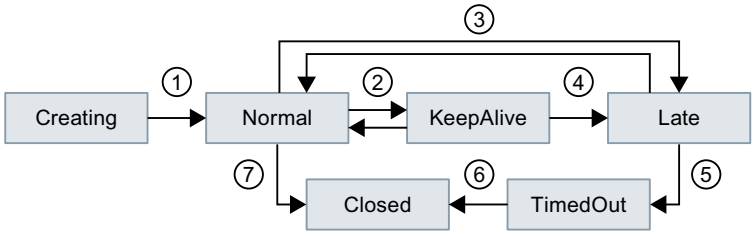
要求

在 CPU 的 OPC UA 属性中，已选择"订阅：状态改变"(Subscriptions: Change of status) 选项（OPC UA > 服务器 > 诊断）。

示例

OPC UA 客户端已连接作为 OPC UA 服务器的 S7-1500 CPU，并在服务器中生成订阅。
在 CPU 的 OPC UA 属性中，已选择订阅对应的选项。
“Creating”和“Normal”状态以及相应的订阅 ID 会依次进入诊断缓冲区。

订阅状态和状态转换



① 订阅已生成，随后变为活动状态。
② 由于进入诊断缓冲区的条目可能过多（具体视数据量而定），状态更改不会进入诊断缓冲区中。
③ 参见表中对“Late”的说明；例如，没有要从客户端发送的请求。
④ 已达到最大 KeepAlive 值。
⑤ 参见表中对“TimedOut”的说明。
⑥ 已达到最大订阅使用期。
⑦ 客户端已删除订阅。

图 11-61 订阅状态和状态转换

订阅状态说明

OPC UA 服务器中的订阅可能有以下状态：

状态	含义
Creating	客户端已请求在服务器中订阅；服务器创建订阅。
Normal	在服务器中创建了订阅，且订阅处于活动状态。
Closed	客户端已删除订阅。
KeepAlive	受监视项的状态长时间未更改。这些状态转换不会进入诊断缓冲区。
Late	客户端已生成具有最小采样和发布间隔的订阅。受监视项的数量在这段时间内未传送到客户端。 客户端不再传送要发送的请求（由于故障等原因）。
TimedOut	客户端已请求订阅。 仅当客户端的发送请求（发布请求）数量足够多时，服务器才会允许订阅（发送发布响应）。 如果客户端停止发送订阅请求，订阅会在特定时间后进入“TimedOut”状态。

订阅：采样时间存在错误

对于固件版本为 V2.5 及以上版本的 SIMATIC S7-1500 CPU，如果在对项目进行采样时发生 CPU 过载，则在使用订阅时，OPC UA 服务器可传送状态代码"GoodOverload"。

对于固件版本为 V2.8 及以上版本的 SIMATIC S7-1500 CPU，OPC UA 服务器还会使该事件进入诊断缓冲区。

要求

在 CPU 的 OPC UA 属性中，已选择"订阅：采样时间存在错误"(Subscription: Sampling time errors) 选项（OPC UA > 服务器 > 诊断）。

无错订阅

如果 OPC UA 订阅多个元素（比如变量），SIMATIC S7-1500 的 OPC UA 服务器必须以指定间隔（采样间隔）检查元素的值是否更改。这种检查称为"采样"，需要一定的时间，具体时长取决于项目数量和数据类型。采样完成并接收到发布请求后，服务器会向客户端发送元素。

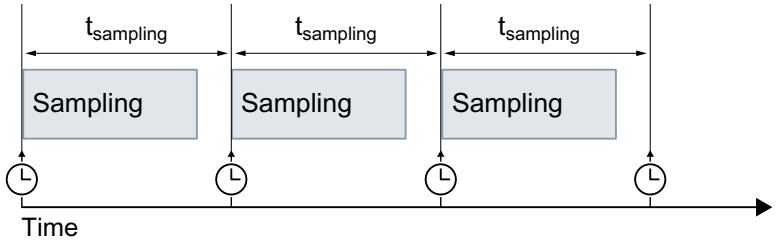
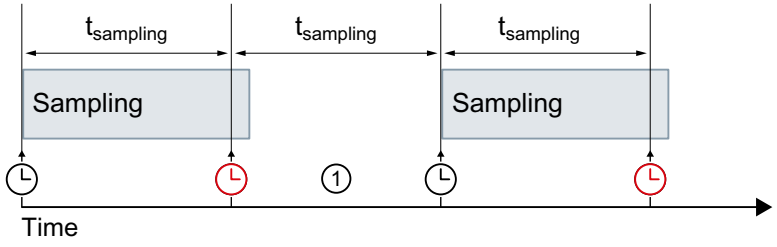


图 11-62 无错订阅

订阅存在错误

如果队列中的元素过多，可能会出现通信堆栈过载的情况。CPU 无法以给定的采样间隔检查所有的元素，因此必须跳过下一采样作业。

在这种情况下，CPU 会为每个元素发送状态码"GoodOverload"(0x002F0000)，即使未对元素进行检查时也是如此。状态码符合 IEC 61131-3 要求，其含义如下："由于资源限制，采样速度减慢"。



① 跳过采样作业

图 11-63 订阅存在错误

另请参见 FAQ 109763090
(https://support.industry.siemens.com/cs/ww/zh/view/109763090)。

更多信息

有关订阅服务器设置的信息，请参见“服务器的订阅设置 (页 228)”部分。

11.3.7.8 汇总诊断

为防止诊断缓冲区被大量相同的 OPC UA 诊断“淹没”，自 STEP 7 V16 服务包 1 开始，可设置相应参数，使这些诊断作为组报警进入到诊断缓冲区中。在每个间隔（监视时间）内，CPU 仅为每个 OPC UA 诊断生成一个组报警。

以下部分介绍了 CPU 对诊断的分组标准以及消息量较大时过程的运行方式。

要求

在 CPU 的 OPC UA 属性中，激活“消息量较大时汇总诊断”(Summarize diagnostics in case of high message volume) 选项（“OPC UA > 服务器 > 诊断”(OPC UA > Server > Diagnostics)，“汇总诊断”(Summarize diagnostics) 区域）。

示例

OPC UA 客户端使用服务器无法处理的采样率（过载）使作为 OPC UA 服务器的 S7-1500 CPU 重复“过载”。

激活“消息量较大时汇总诊断”(Summarize diagnostics in case of high message volume) 设置。

一条消息会出现在该诊断选项的诊断缓冲区中。该消息会提示无法达到该采样率；后接组态间隔内此类事件的数量。

可概括的 OPC UA 诊断

下列诊断各自形成自己的组（类型）。来自同一组的诊断事件通过“消息量较大时汇总诊断”(Summarize diagnostics in case of high message volume) 设置合并在一起：

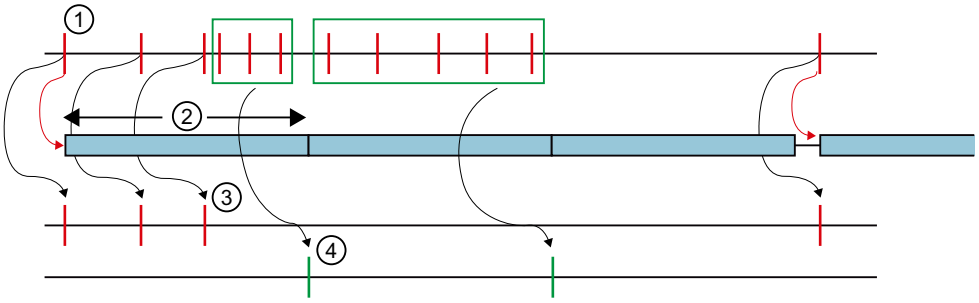
- OPC UA 服务使用错误
- OPC UA 服务错误
- 订阅状态已更改
- 无法达到采样率（订阅、过载）
- OPC UA 安全检查失败
- 超出 OPC UA 服务器的组态限值

工作原理

CPU 在诊断缓冲区内输入一种事件类型的前三个事件。随后会忽略该组的所有后续诊断。

在监视时间（间隔）结束时，CPU 生成组报警，在该组中输入过去的时间间隔内的诊断和该诊断的频率。如果这些诊断在随后的时间间隔内也有出现，CPU 将仅为每个后续的时间间隔生成一个组报警。

诊断激增会在诊断缓冲区中留下以下模式：三个单独的消息，后跟一系列组报警。此系列可以包含两个、三个或更多的组报警，具体取决于选定的监视时间和诊断激增的持续时间。



① 一组（一种类型）的诊断结果，例如“无法达到采样率”。

② 间隔（监视时间）：在诊断事件首次发生（或重复发生）时，监视时间开始（或重新开始）计时。

③ 单个报警：来自同一组的前三个诊断事件会立即进入诊断缓冲区。从第四个诊断事件开始，CPU 仅会生成组报警。如果该组的一个诊断事件在至少暂停一个间隔后发生，CPU 将在诊断缓冲区中输入单个报警并对监视时间重新计时。

④ 组报警：在三个诊断事件后，CPU 仅生成一个组报警作为此间隔内所有附加诊断事件的汇总。如果这些诊断事件在随后的时间间隔内也有出现，CPU 将仅为每个后续的间隔生成一个组报警。

图 11-64 诊断摘要

11.4 将 S7-1500 CPU 用作 OPC UA 客户端

11.4.1 概述和要求

利用 STEP 7 (TIA Portal) 版本 V15.1 及更高版本，可为可读取 OPC UA 服务器中 PLC 变量的 OPC UA 客户端分配参数并进行编程。此外，还可以将 PLC 变量的新值传送到 OPC UA 服务器。另外还可以在用户程序中调用 OPC UA 服务器提供的方法。为此，在用户程序中使用 OPC UA 客户端的指令。

OPC UA 客户端的指令基于“符合 IEC61131-3 规范的 PLCopen OPC UA 客户端”。

PLCopen 规范

可利用这些标准化指令在用户程序中开发 OPC UA 客户端函数，该函数可在 S7-1500 CPU 中执行。

此外，只需稍作调整便可在其它制造商生产的控制器中运行该用户程序（如果这些制造商也实施了 OPC UA 规范“符合 IEC61131-3 规范的 PLCopen OPC UA 客户端”）。

STEP 7 中便捷的编辑器

为了对 OPC UA 客户端的指令进行参数分配，TIA Portal 中提供了便捷的编辑器 连接参数分配 (页 223)。

自版本 V15.1 起，STEP 7 还增加了用于客户端接口的编辑器 (页 323)。

本节将介绍这些编辑器的操作方法。

首先会介绍如何使用接口编辑器创建和组态新接口，因为需要使用此类型的接口进行后续的连接参数分配。

我们通过举例的方式让说明更易于理解，请参见"示例说明 (页 322)"。

要求

- 必须具有 OPC UA 的运行系统许可，并且已在 STEP 7 中组态该许可"CPU 属性 > 运行系统许可证"(CPU Properties > Runtime Licenses)。
- S7-1500 CPU 的客户端已激活。

要使用 S7-1500 CPU 的客户端，必须启用该客户端：

1. 在 CPU 特性中选择"OPC UA > 客户端"(OPC UA > Client)。
2. 选择"启用 OPC UA 客户端"(Enable OPC UA client) 选项。

如果未启用客户端，则不会建立连接。收到指令（例如"OPC_UA_Connect"）的相应错误消息。

有关同样应用于服务器和客户端的应用程序名称的信息，请参见此处 (页 223)。

概述

要使用编辑器和连接参数分配，请执行以下步骤：

1. 首先指定一个客户端接口为该客户端接口添加要访问的 PLC 变量和 PLC 方法接口（"第一步 (页 323)"）。
2. 接下来组态与 OPC UA 服务器的连接（第二步 (页 336)）。
3. 最后使用为 OPC UA 客户端指令组态的连接（第三步 (页 343)）。

11.4.2 有关客户端指令的重要信息

利用标准化 OPC UA 客户端指令，用户能够控制以下任务与作为 OPC UA 客户端的 S7-1500 CPU 的通信。

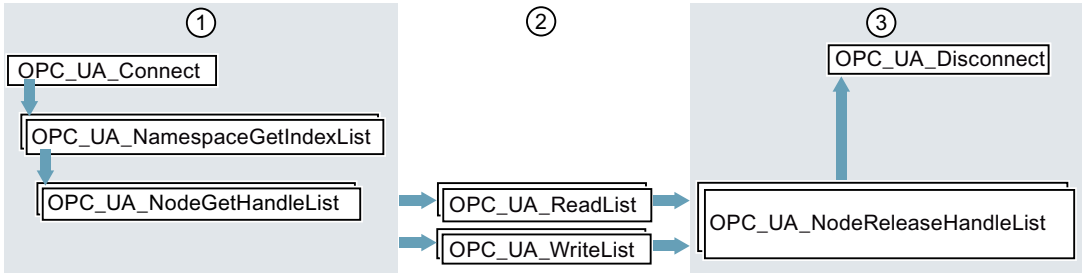
- 读取/写入 OPC UA 服务器的变量
- 调用 OPC UA 服务器中的方法

使用可选指令可确定以下信息：

- OPC UA 客户端与 OPC UA 服务器之间连接的状态
- 地址空间层级已知的节点的节点 ID

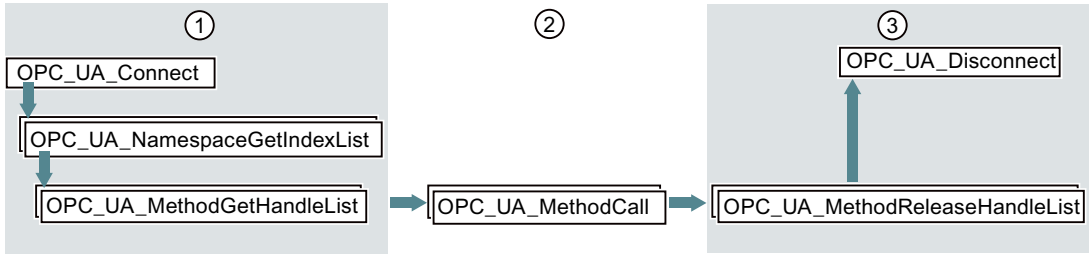
OPC UA 通信的标准化顺序

通信顺序以及指令顺序按照下图所示的模式进行。



① 用于准备读取和写入操作的指令
② 读取和写入指令
③ 完成读取或写入操作后用于进行"清理"的指令

图 11-65 读取或写入操作的运行顺序

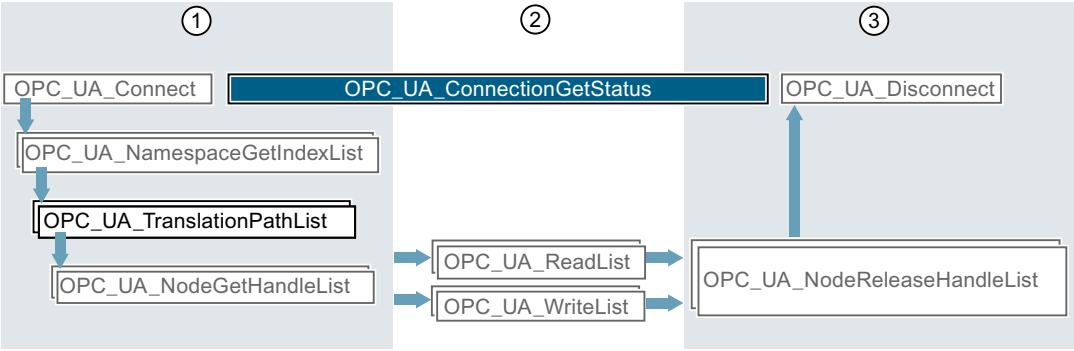


① 用于准备方法调用的指令
② 方法调用
③ 完成方法调用后用于进行"清理"的指令

图 11-66 OPC UA 服务器中方法调用的运行顺序

可选指令（读取连接状态/读取地址空间层级已知的节点的节点 ID）

- OPC_UA_ConnectionGetStatus
- OPC_UA_TranslatePathList



① 用于准备通过插入的指令执行读写操作的指令，可用于请求 OPC UA 服务器节点的 NodeID 等信息。

② 可与其它指令共同确定连接建立与终止之间的连接状态。

③ 用于进行"清理"的指令

图 11-67 可选指令的运行顺序

STEP 7 中便捷的编辑器

参考部分（STEP 7 信息系统）详细介绍了 OPC UA 客户端指令为了对指令进行参数分配，TIA Portal 中提供了便捷的编辑器 – 连接参数分配 (页 336)。

建议先为第一个程序草稿进行连接参数分配，根据需要使用附加指令并手动优化程序。

有关客户端指令的信息

"指令 > 通信 > OPC UA 客户端"(Instructions > Communication > OPC UA > OPC UA client) 的帮助中详细介绍了客户端指令。

在线支持中的应用示例

此应用示例 (https://support.industry.siemens.com/cs/ww/zh/view/109762770)为用户提供 S7 用户块"OpcUaClient"，该块汇总了 OPC UA 指令的最重要功能，加快项目实现并简化编程。示例中的 OPC UA 服务器是一个 S7-1500 控制器，带有简单的过程值仿真程序。

S7 用户块执行以下操作：

- 建立和终止与服务器的连接
- 诊断连接以及在连接终止后自动重新连接
- 注册读取
- 注册写入
- 注册方法调用

11.4.3 可同时使用的客户端指令数

OPC UA 客户端指令的 SIMATIC 错误代码

同时使用 OPC UA 客户端指令时，将应用以下限值（有关 CPU 的最新技术规范，敬请访问 Internet (https://support.industry.siemens.com/cs/ww/zh/ps/td)）：

表格 11-4 OPC UA 客户端指令的结构数量

OPC UA 指令	**最大数量 CPU 1510SP (F) CPU 1511 (C/F/T/TF) CPU 1512C CPU 1512SP (F) CPU 1513 (F)**	**最大数量 CPU 1505 (S/SP/SP F/SP T/SP TF) CPU 1515 (F/T/TF) CPU 1515 SP PC (F/T/TF) CPU 1516 (F/T/TF)**	**最大数量 CPU 1507S (F) CPU 1517 (F/T/TF) CPU 1518 (F)**
OPC_UA_Connect	4	10	40
OPC_UA_NamespaceGetIndexList	4*	10*	40*
OPC_UA_NodeGetHandleList	4*	10*	40*
OPC_UA_MethodGetHandleList	4*	10*	40*
OPC_UA_TranslatePathList	4*	10*	40*
OPC_UA_ReadList	总计 20 个（每个连接最多 5 个；参见 OPC_UA_Connect）	总计 50 个（每个连接最多 5 个；参见 OPC_UA_Connect）	总计 200 个（每个连接最多 5 个；参见 OPC_UA_Connect）
OPC_UA_WriteList	总计 20 个（每个连接最多 5 个；参见 OPC_UA_Connect）	总计 50 个（每个连接最多 5 个；参见 OPC_UA_Connect）	总计 200 个（每个连接最多 5 个；参见 OPC_UA_Connect）
OPC_UA_MethodCall	总计 20 个（每个连接最多 5 个；参见 OPC_UA_Connect）	总计 50 个（每个连接最多 5 个；参见 OPC_UA_Connect）	总计 200 个（每个连接最多 5 个；参见 OPC_UA_Connect）
OPC_UA_NodeReleaseHandleList	4*	10*	40*
OPC_UA_MethodReleaseHandleList	4*	10*	40*
OPC_UA_Disconnect	4*	10*	40*
OPC_UA_ConnectionGetStatus	4*	10*	40*

* 每个连接最多 1 个

可用的 OPC UA 客户端接口最大数量

如果通过连接参数分配创建 OPC UA 客户端接口，则客户端接口的最大数量将限制为 40 个。

如果在项目树“OPC UA 通信”(OPC UA communication) 区域内，通过双击“新增客户端接口”(Add new client interface) 符号，创建 OPC UA 客户端接口，

则 OPC UA 客户端接口的最大数量与是否将该 CPU 用作 OPC UA 服务器无关。

11.4.4 OPC UA 示例组态

以下部分介绍了如何使用客户端接口编辑器和连接参数分配。

说明基于特定示例：两个 S7-1500 CPU 在系统中运行：一个 CPU 用作 OPC UA 客户端，另一个用作 OPC UA 服务器。

当然，其它制造商生产的控制器、传感器和 IT 系统也可用作 OPC UA 客户端或服务器。特别值得一提的是，在不同系统之间进行数据交换（互操作性）是 OPC UA 的主要优点。

使用示例说明连接参数分配：

工厂在生产线中生产坯件。

会使用以下控制器：

1. S7-1511 CPU 用作生产线的控制器。

   在本示例中，该控制器名为“**Productionline**”。

   控制器的 OPC UA 服务器已启用。

   在本示例中，该 CPU 的 IP 地址为 192.168.1.1。

   该 CPU 通过 OPC UA 服务器发布以下变量的值：

   - **NewProduct**

     变量的数据类型为“BOOL”。

     该 PLC 变量的值为 TRUE 时，生产线已加工一个坯件。

     坯件准备好被拾取。

   - **ProductNumber**

     该变量包含坯件的标识号。

     变量的数据类型为“Int”。

   - **Temperature**

     该变量包含在生产坯件过程中记录的温度值。

     变量为包含“Real”数据类型的元素的数组。

   此外，该 CPU 提供以下可写变量：

   - **ProductionEnabled**

     变量通过 OPC UA 客户端进行设置。

     变量的数据类型为“BOOL”。

     如果数值设为 TRUE，说明生产线已释放，可生产坯件。

   此外，该 CPU 还通过 OPC UA 服务器提供以下方法：

   - **OpenDoor**

     通过此方法，OPC UA 客户端可安排打开生产线检修门。

2. S7-1516 CPU 控制着与其它生产线的交互。

   在本示例中，该 CPU 的名称为“**Supervisor**”。

   该 CPU 的 OPC UA 客户端已启用。

   利用 OPC UA，该 CPU 可读取 NewProduct 和 ProductNumber 变量、设置 ProductionEnabled 变量，并可调用 OpenDoor 方法。

   在本示例中，该 CPU 的 IP 地址为 192.168.1.2。

下图显示了 TIA Portal 网络视图中的示例：

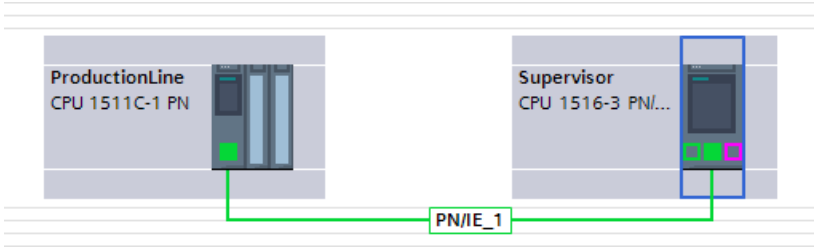


图 11-68 在网络视图中分配连接参数的示例

11.4.5 创建客户端接口

自版本 V15.1 起，TIA Portal 增加了用于客户端接口的编辑器。

将要从 OPC UA 服务器读取或写入的所有 PLC 变量分组到客户端接口中。

此外，客户端接口包含 OPC UA 服务器提供的以及要通过用户程序（作为 OPC UA 客户端）调用的所有方法。

如果创建客户端接口，STEP 7 还会创建数据块，用于对与 OPC UA 服务器进行数据读写操作要使用的连接进行参数分配。

客户端接口的最大数量

最多可以创建 40 个客户端接口。

用户客户端接口的编辑器

要创建客户端接口，请按以下步骤操作：

1. 在 TIA Portal 中选择项目视图。
2. 在“设备”(Devices) 区域，选择要作为 OPC UA 客户端使用的 CPU。
3. 单击“OPC UA 通信 > 客户端接口”(OPC UA communication > Client interfaces)。
4. 双击“添加新客户端接口”(Add new client interface)。

   STEP 7 会创建新客户端接口并在编辑器中显示。

图 11-69 添加 OPC UA 客户端接口

STEP 7 将新接口命名为“Client interface_1”。如果“Client interface_1”已存在，新接口会接收名称“Client interface_2”，以此类推。

此外，STEP 7 还会创建下列数据块：

– Client_Interface_1_Configuration

数据块已包含 OPC UA 客户端指令需要的所有系统数据类型。

组态与 OPC UA 服务器的连接时，会填充该数据块。

在客户端接口的特性中组态连接，参见“OPC UA 示例组态 (页 322)”。

– Client_Interface_1_Data

要从 OPC UA 服务器读取或写入的 PLC 变量以及要在 OPC UA 服务器中调用的方法对应的数据块。

在用户程序中使用该数据块。

该数据块当前仍为空。

5. 为新客户端接口选择一个描述性名称。

在本示例中，选择“Productionline”。

此操作还会将关联数据块的名称改为：

– Productionline_Data

– Productionline_Configuration

6. 要导入 OPC UA 服务器接口，请单击编辑器右上方的“导入接口”(Import interface) 按钮。

这样便可导入描述 OPC UA 服务器接口的 XML 文件。

或者：要在线确定已连接 OPC UA 服务器的服务器接口，请参见“在线确定服务器接口 (页 330)”。

7. STEP 7 会显示一个对话框，可通过该对话框选择 XML 文件。

该 XML 文件描述 OPC UA 服务器的地址空间。

OPC UA 服务器的地址空间包含由 OPC UA 服务器发布的所有 PLC 变量和服务器方法。

OPC UA 客户端可访问该地址空空间：

- 读取 PLC 变量

- 写入 PLC 变量

- 调用服务器方法

OPC UA 服务器的地址空间可分为一个或多个服务器接口。

要创建服务器接口，请参见：为配套规范创建服务器接口 (页 254)。

8. 在该客户端接口创建一个**读取列表**。

为此，请执行以下操作步骤：

– 单击编辑器左侧部分的“添加新读取列表”(Add new read list)。

STEP 7 将添加一个名为“ReadList_1”的新列表。

本示例中，将该名称更改为“ReadListProduct”。

– 现在将要从该 OPC UA 服务器读取的 PLC 变量添加到新读取列表中。

在本示例中，将“NewProduct”和“ProductNumber”变量添加到“ReadListProduct”读取列表中。

在编辑器右侧区域选择“NewProduct”变量（“OPC UA 服务器接口”）。
将“NewProduct”变量拖动到编辑器中间部分的“ReadProduct”读取列表中。
对“ProductNumber”变量采用相同的操作步骤。

下图显示了编辑器的右侧部分。

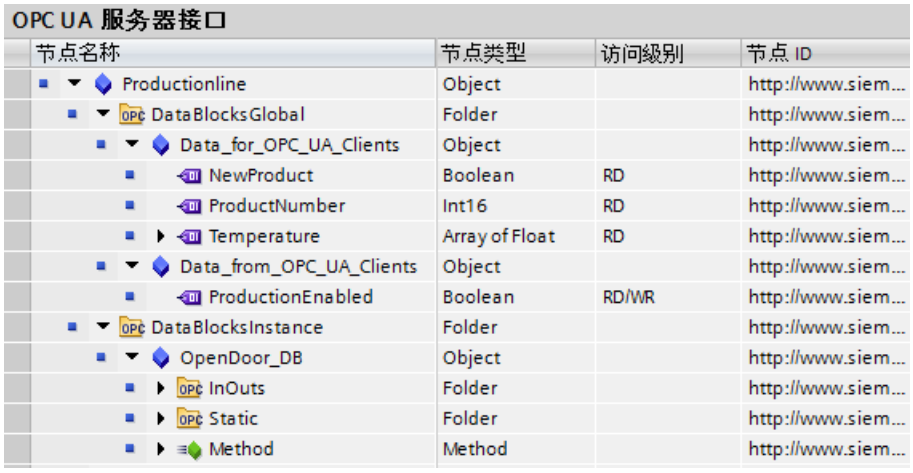


图 11-70 OPC UA 服务器接口中的读取列表

或者：

选择新读取列表时，还可将编辑器的右侧部分（“OPC UA 服务器接口”）拖动到类型为 Object 或 Folder 的节点处，然后再将其拖动到编辑器左侧部分的“添加新读取列表”(Add new read list) 中。新读取列表随即包含已移动节点的所有 PLC 变量。

在本示例中，选择包含“NewProduct”和“ProductNumber”变量的对象“Data_for_OPC_UA_Clients”。STEP 7 生成新的读取列表“Data_for_OPC_UA_Clients”。此外，对象还包含“Temperature”变量。将“Temperature”变量从读取列表中删除。因此本例中不应读取这些变量。

在“ReadListProduct”中更改读取列表的名称。

下图显示了读取列表的内容：

图 11-71 读取列表

说明

读取和写入列表并不支持所有节点类型。

S7-1500 CPU 的 OPC UA 客户端不支持可通过 OPC UA 服务器接口实现的所有 OPC UA 数据类型（节点类型）。举例来说，如果将不受支持的节点类型放在读取列表或写入列表中，则会出现相应的错误信号。在这种情况下，不能将相应节点包含在读取或写入列表中。

有关支持的类型，请参见“数据类型映射 (页 204)”

---

9. 如果要将新值分配给 PLC 变量，则在该客户端接口创建一个**写入列表**。

   为此，请执行以下操作步骤：

   – 单击编辑器左侧部分中的“添加新写入列表”(Add new write list)。

     STEP 7 将添加一个名为“ReadList_1”的新列表。

     在本示例中，将该名称更改为“WriteListStatus”。

   – 现在添加新写入列表，其中包含要为其分配新值的所有 OPC UA 服务器变量。

     在本示例中，将“WriteListStatus”变量添加到写入列表“ProductionEnabled”中。

     选择编辑器右侧区域（“OPC UA 服务器接口”）的变量。将变量拖动到编辑器中间部分的写入列表中。

   或者：

   创建新写入列表时，还可在编辑器的右侧部分（“OPC UA 服务器接口”）选择类型为 Object 或 Folder 的节点，然后再将其拖动到编辑器左侧部分的“添加新写入列表”(Add new write list) 中。

   新写入列表随即包含相关节点的所有变量。

   在本示例中，选择包含“ProductionEnabled”变量的对象“Data_from_OPC_UA_Clients”。STEP 7 会生成新的写入列表“Data_from_OPC_UA_Clients”。在“WriteListStatus”中更改名称。

   下图显示了写入列表的内容：

WriteListStatus

节点名称	节点类型	访问级别	节点 ID
ProductionEnabled	BOOL	RD/WR	http://www.siemen...

图 11-72 写入列表

10. 如果要调用该 OPC UA 服务器的方法，应生成新方法列表。

    为此，请执行以下操作步骤：

    – 在编辑器左侧部分中，单击“添加新方法列表”(Add new method list)。

      STEP 7 将添加一个名为“Method List_1”的新列表。

      在本示例中，将该名称更改为“MethodListOpenDoor”。

    – 现在将 OPC UA 服务器的方法添加到新方法列表中。

      在本示例中，将方法“OpenDoor”添加到方法列表“MethodListOpenDoor”中。

      选择编辑器右侧区域（“OPC UA 服务器接口”）的方法。将方法拖动到编辑器中间部分的方法列表中。

    或者：

    生成新方法列表时，还可在编辑器的右侧部分（OPC UA 服务器接口）选择方法（类型为 Object 的节点），然后再将其拖动到编辑器左侧部分的“添加新方法列表”(Add new method list) 中。新方法列表随即包含相关节点的方法。

    下图显示了方法列表的内容：

MethodListOpenDoor

节点名称	节点类型	访问级别	节点 ID
Method			http://www.siemen...

图 11-73 方法列表

另请参见“关于服务器方法的有用信息 (页 279)”。

11. 编译项目。

    为此，请选择项目并单击工具栏中的以下按钮：

STEP 7 会编译项目并更新属于“Productionline”客户端接口的数据块。

说明

编译过程中，STEP 7 会覆盖属于客户端接口的数据块中的所有数据。因此，不应手动向这些数据块添加内容，也不能进行更正。

说明

重命名节点 (DisplayNames)

在读取列表、写入列表和方法列表中，可通过快捷菜单重命名节点。该名称为 OPC UA 语言用例中的“DisplayName”。

如果重命名方法列表节点，且该节点已用于方法调用“OPC_UA_MethodCall”的已编程块中，项目编译会出现一致性错误：编译过程中，会生成方法的 UDT 以及已更改的名称。对程序中所用方法的引用随后不再正确。

要更正一致性错误，可在客户端接口中撤消对方法名称的更改，也可以浏览至方法调用并再次在“特性 > 块参数”(Properties > Block parameters)（“组态”(Configuration) 选项卡）下分配相关参数。

客户端接口的数据块

以下数据块属于“Productionline”客户端接口：

- **Productionline_Configuration**

  用于组态的数据块。

  在本示例中，该数据块名为“Productionline_Configuration”。

  数据块已包含 OPC UA 客户端指令需要的所有系统数据类型。

  此外，数据块还包含与 OPC UA 服务器的连接的参数分配常规默认值。

  如果要进行连接参数分配，该将该数据块填入数值。

- **ProductionLine_Data**

  在客户端接口编辑器中输入的用于 PLC 变量的数据块。

  在本示例中，该数据块名为“Productionline_Data”。

  下图显示了数据块。

Static	
ReadListProduct	Struct
Variable	"Productionline.ReadListProduct"
NewProduct	Bool
ProductNumber	Int
NodeStatusList	Array[0..1] of DWord
NodeStatusList[0]	DWord
NodeStatusList[1]	DWord
TimeStamps	Array[0..1] of LDT
TimeStamps[0]	LDT
TimeStamps[1]	LDT
WriteListStatus	Struct
Variable	"Productionline.WriteListStatus"
ProductionEnabled	Bool
NodeStatusList	Array[0..0] of DWord
NodeStatusList[0]	DWord
MethodListOpenDoor	Struct
MethodStatusList	Array[0..0] of DWord
MethodStatusList[0]	DWord
MethodResultList	Array[0..0] of DWord
MethodResultList[0]	DWord
Method	Struct
Inputs	"Productionline.MethodListOpendoor.Method.Inputs"
Number	Int
Outputs	"Productionline.MethodListOpendoor.Method.Outputs"
Result	Int

图 11-74 “Productionline_Dat”数据块

在用户程序中使用“Productionline_Data”数据块并访问“NewProduct”和“ProductNumber”PLC 变量的读取值。下一章节将通过示例对此进行说明。

读取和写入客户端接口的 PLC 变量

示例：读取“ProductNumber”值

例如，在 SCL 程序中写入：

```
#MyLocalVariable :=
"Productionline_Data".ReadListProduct.Variable.ProductNumber;
```

举例来说，可使用该语句将生产线中刚生产出的坯件编号分配给局部变量“#MyLocalVariable”。

要求：

- 存在与控制着生产线的 CPU 的 OPC UA 服务器的连接。
- OPC UA 客户端已读取当前值。

为此，应检查读取值是否有效：

- 检查 "Productionline_Data".ReadListProduct.NodeStatusList[1] 中的值是否等于 0。
- 可选：检查从 OPC UA 服务器发送该值的时间。该值位于 "Productionline_Data".Product.TimeStamps[1] 中。如果未请求时间戳，通信负荷会降低。

示例：写入"ProductEnabled"值

使用数据块将 PLC 变量（本示例中为"ProductionEnabled"）的新值传送到 OPC UA 服务器。

进行下列分配后，可启用示例工厂中的生产线：

```
"Productionline_Data".WriteListStatus.Variable.ProductionEnabled := TRUE;
```

但只有满足以下要求时才能成功：

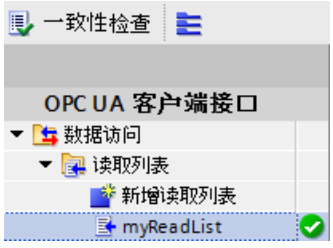
- 存在与控制着生产线的 CPU 的 OPC UA 服务器的连接。
- 当前值将通过 OPC UA 客户端写入

一致性检查

最后，检查读取/写入列表或方法列表的一致性。

1. 选择要检查的列表。
2. 单击"OPC UA 客户端接口"(OPC UA client interface) 区域上方的"一致性检查"(Consistency check) 按钮。

绿色复选标记指示将变量或方法分配给服务器接口的相应元素时不存在错误。



可假定客户端与服务器间的数据交换以及方法调用在运行时未出错。

一旦出错，将在巡视窗口中显示一个列表。通过该列表，可跳转到相应的错误处。

一致性检查期间，STEP 7 会检查：

- 在相应列表中使用的所有元素是否同样存在于服务器中。
- 所用的数据类型是否匹配？
- 对于方法：方法变量的数量、名称、顺序和数据类型是否匹配？

11.4.6 在线确定服务器接口

可通过 STEP 7 (TIA Portal) 在线确定 OPC UA 服务器的接口。这样便可提供可通过 OPC UA 客户端读取或设置（写入）已连接 OPC UA 服务器的哪些变量的相关信息，还可提供 OPC UA 服务器的哪些服务器方法可用于 OPC UA 客户端的相关信息。

如果离线操作，可通过 OPC UA XML 文件创建 OPC UA 服务器的接口。服务器的地址空间在 OPC UA XML 文件中进行描述，请参见"将 OPC UA 导出为 XML 文件 (页 221)"。

确定在线服务器接口

要在线确定服务器接口，请按以下步骤操作：

1. 在 STEP 7 项目树中，选择组态为 OPC UA 客户端（本例为 Supervisor）的 CPU。
2. 选择客户端接口（本例中为"OPC UA 通信 > 客户端接口 > Productionline"(OPC UA communication > Client interfaces > Productionline)）。

   如果尚未创建客户端接口，请双击"添加新客户端接口"(Add new client interface)。
3. 双击所选客户端接口。

   会显示客户端接口的编辑器。

图 11-75 客户端接口编辑器

4. 在编辑器左侧部分，单击"添加新读取列表"(Add new read list)、"添加新写入列表"(Add new write list) 或"添加新方法列表"(Add new method list)。
5. 在编辑器的右侧部分，选择"在线 []"(Online[]) 作为"服务器数据源"(Source of server data) 的数据源：

6. 单击“在线访问”(Online Access) 按钮。

   STEP 7 会显示“连接到 OPC UA 服务器”(Connect to OPC UA server) 对话框。

图 11-76 “连接到 OPC UA 服务器”(Connect to OPC UA server) 对话框

提示：首次与 OPC UA 服务器建立在线连接时，可使用“在线访问”(Online access) 按钮。断开后重新连接时，可选择“在线”(Online) 选择框旁的“连接到在线服务器”(Connect To Online Server) 按钮。

在右上方输入要在线确定其服务器接口的 OPC UA 服务器的 IP 地址。

7. 单击“查找已选服务器”(Find selected server)。

   STEP 7 会与 OPC UA 服务器建立连接，并会确定服务器保持在就绪状态的所有安全设置（服务器端点）。

   STEP 7 会以列表形式显示端点：

图 11-77 发现包含所有服务器端点的 OPC UA 服务器

8. 单击将 STEP 7 连接到 OPC UA 服务器时要使用的端点。

9. 是否要使用安全连接？
   - 如果选择了一个安全端点，则为“证书位置”(Certificate location) 选择条目“TIA Portal”。

     在“证书（客户端）”(Certificate (Client)) 下，为当前运行 STEP 7 (TIA Portal) 的 PC 选择客户端证书。

     如果不存在用于此 PC 的客户端证书，可在 TIA Portal 中生成客户端证书。

     要为 PC 生成证书，请按以下步骤操作：

     - 单击“证书（客户端）”(Certificate (Client)) 输入字段中的按钮。

     - 单击“添加”(Add)。

     - 对于“证书所有者”(Certificate owners)，输入“STEP 7 (TIA Portal)”。

     - 在“使用”(Usage) 处选择“OPC UA 客户端”(OPC UA client) 条目。

     - 对于“主题备用名称 (SAN)”(Subject Alternative Name (SAN))，在“值”(Value) 下输入当前运行 STEP 7 (TIA Portal) 的 PC 的 IP 地址。覆盖已输入的 IP 地址。

     - 如果您的 PC 使用其它 IP 地址，也请输入该地址。如果 PC 未使用其它 IP 地址，请删除已输入的另一 IP 地址。

     - 单击“确定”(OK)。
   - 如果尚未选择安全端点，则请保留默认值（“无”(None)）。

10. 希望以何种身份登录？
    - 如果要以访客身份登录 OPC UA 服务器，则为“用户认证”(User authentication) 应用默认设置。
    - 如果要使用用户名和密码登录，请选择“用户名和密码”(User name and password)。

      使用组态 OPC UA 服务器期间在 CPU 特性的“常规 > OPC UA > 服务器 > 安全 >用户认证 > 用户管理”(General > OPC UA > Server > Security > User authentication > User management) 下存储的用户名和密码。

11. 单击“转至在线”(Go online) 按钮。

    建立安全连接时，会显示一条消息，提示必须接受服务器证书才能建立安全连接。在消息窗口中，可通过链接显示关于服务器证书的其它详细信息。

    标准 Windows 窗口仅提供关于服务器证书的信息。如果单击按钮来安装服务器证书，则服务器证书不会保存在 TIA Portal 的证书存储器中，也就是说，下一次尝试建立连接时，系统会再次提示用户接受服务器证书。

    STEP 7 随即会与 OPC UA 服务器建立连接，并会再次显示客户端接口编辑器。

    在编辑器的右侧部分中，STEP 7 会显示 OPC UA 服务器的最上级地址空间：

12. 单击“Objects”旁的黑色小三角形。

    STEP 7 现在还会显示 Objects 以下的等级。

13. 单击“Productionline”旁的黑色小三角形。

    STEP 7 现在还会显示 Productionline 以下的等级。

14. 现在打开其它等级较低的文件夹：

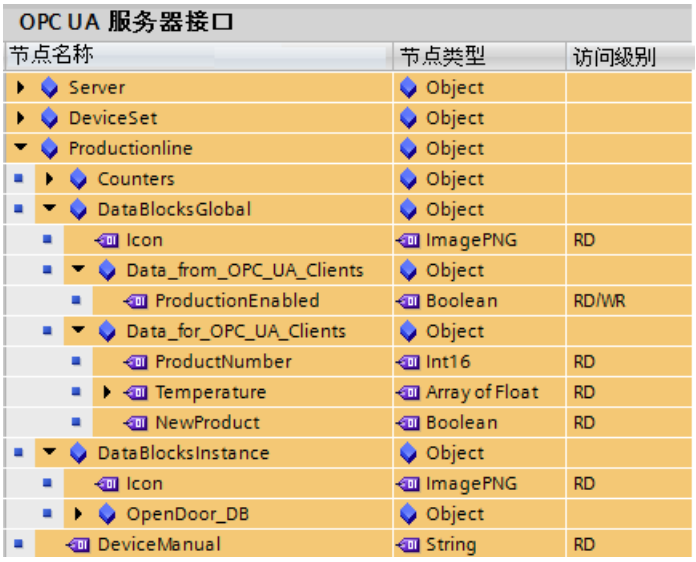


图 11-78 OPC UA 服务器接口在线视图

更多信息

有关数据类型映射的信息，请参见“数据类型映射 (页 204)”部分。

有关创建客户端接口的信息，请参见“创建客户端接口 (页 323)”部分。

11.4.7 使用多语言文本

在客户端接口编辑器中，还要导入可在 OPC UA XML 文件（信息模型）中以不同语言显示的文本。多语言显示为可选功能，可针对各节点提供的语言进行不同定义。

在 XML 文件中，可为不同语言准备以下字段：

- 显示名称
- 说明

OPC UA XML 文件中的多语言文本示例

举例来说，在下方的 XML 文件中，会使用“默认”文本和多个可本地化文本输入显示语言和描述。

- 默认文本是不含本地化信息的第一个条目。
- 本地化文本是“Locale=”后的文本加语言代码，例如“it-IT”代表意大利语。

```
<UAVariable NodeId="ns=3;i=6070" BrowseName="3:EngineeringRevision" ParentNodeId="ns=3;i=1002"
DataType="String">
    <DisplayName>EngineeringRevision</DisplayName>
    <DisplayName Locale="en-US">EngineeringRevision</DisplayName>
    <DisplayName Locale="de-DE">Revisionsstand</DisplayName>
    <Description>Revision Level of the engineering environment.</Description>
    <Description Locale="en-US">Revision Level of the engineering environment.</Description>
    <Description Locale="de-DE">Revisionsstand der Engineeringumgebung.</Description>
    <Description Locale="fr-FR">Niveau de révision de l'environnement d'ingénierie.</Description>
    <Description Locale="it-IT">Livello di revisione dell'ambiente di ingegneria.</Description>
<References>
    <Reference ReferenceType="HasTypeDefinition">i=68</Reference>
    <Reference ReferenceType="HasModellingRule">i=78</Reference>
</References>
</UAVariable>
```

图 11-79 OPC UA XML 文件中的多语言文本示例

多语言文本显示

导入服务器接口时，可用的多语言文本会保存在内部，并会与项目一起下载到 CPU。

客户端编辑器会在“节点名称”(Name of the node) 列（对应于“DisplayName”）和“说明”(Description) 列（对应于“Description”）中显示 OPC UA XML 文件中的文本。

以下级联规则可确定显示节点所用的语言：

- 如果节点包含使用当前所用编辑语言的文本，则文本还会以该编辑语言显示。

  （设置编辑语言：在项目树中，选择“语言 & 资源 > 项目语言”(Languages & resources > Project language) 区域。
- 如果节点不包含采用编辑语言的文本，但定义了默认文本（无语言代码），则会显示默认文本。
- “节点名称”(Name of the node) 列：如果未定义默认文本，但存在采用其它任何语言的文本，则会以第一种可用语言显示 DisplayName 文本。此规则不适用于说明文本。
- 如果上述条件无一满足，则不会显示文本。

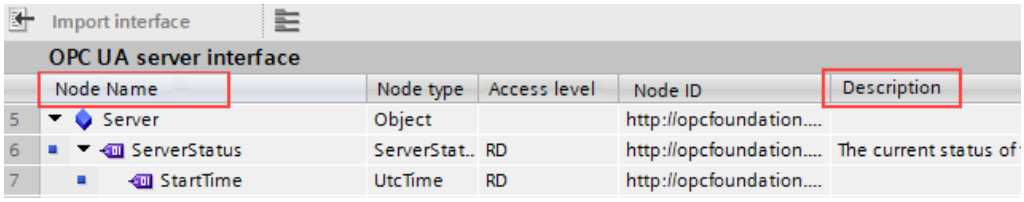


图 11-80 多语言文本的显示

更改编辑语言时，已导入接口中的多语言文本也会按照上述规则更改。

随后可通过拖放操作应用相应列表（读取列表、写入列表、方法列表）中的节点。

不能更改列表中的语言（读取列表、写入列表、方法列表）。

以 PLC 数据类型中的注释形式应用显示的说明文本

编译程序时，STEP 7 会自动为每个读取列表、写入列表以及每个方法的输入或输出创建 PLC 数据类型 (UDT)。这些 UDT 均为各节点包含一个元素。

UDT 会根据上述规则以注释形式应用说明文本。STEP 7 仅会以一种语言创建注释，正如 OPC UA 服务器接口中的文本仅会以一种语言显示。

11.4.8 结构的访问规则

在下文中，将详细介绍访问结构时的相应规则。读取和写入 OPC UA 服务器中整个结构的值时，需遵循这些规则。

S7-1500 CPU 的客户端如何访问结构

S7-1500 CPU 的 OPC UA 客户端并不使用 TypeDictionaries 和 DataTypeDefinition 属性（服务器通过这些属性对结构进行解析）进行结构访问。

在运行系统中，OPC UA 客户端用于检查结构化元素的这些选项使用受限。

结构的访问规则

如果使用客户端接口组态读取和写入列表（连接参数设置），并将 PLC 数据类型分配给该服务器导入的或在线选定的地址模型，则在运行系统中可正常对结构进行读写访问。

通过客户端接口进行的组态可自动确保客户端和服务器端结构元素的顺序和数据类型相匹配。

建议：将 S7-1500 CPU（作为服务器）更新为最新固件版本（例如 V2.0 > V2.5.2 或更高版本）。

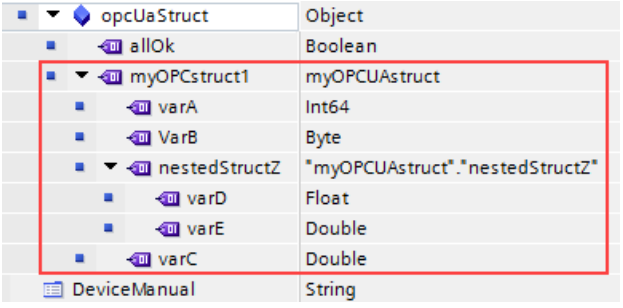
在运行系统中，OPC UA 客户端仅检查传输值的总长度，而不会进行更为详细的检查。

结构中还允许使用字符串（WSTRING、STRING 和 OPC UA ByteString）。字符串的长度虽然可变，但 OPC UA 通过以下措施限制长度变化：传送时，在每个字符串前面附加一个长度字段，对字符串长度进行编码。因此，作为 OPC UA 客户端的 S7-1500 CPU 可检查字符串长度，并确定该字符串是否“适合”分配的 CPU 变量。通过这种方式，CPU 还可以检查结构的总长度。

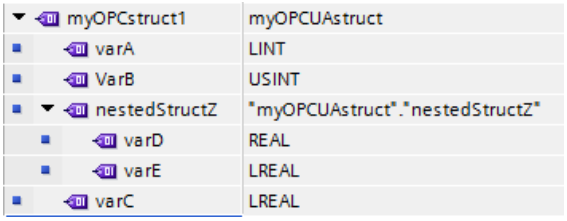
将 OPC UA 结构分配给 PLC 变量或 DB 变量时，需遵循映射规则（参见“数据类型映射 (页 204)”）。

正确分配结构元素的示例

在所导入的节点集文件（XML 导出）中，结构定义如下所示：



该结构与读取列表中的顺序、分配的数据类型，节点集文件中相应节点相匹配。



如果在服务器上更改该结构（如，交换变量 A 和变量 B），而客户端的读取列表保持不变，则将发生分配错误：

- 数据的总长度保持不变（仅顺序更改）
- 客户端和服务器的结构组态不同！

> ⚠警告
>
> **客户端和服务器的结构组态不同时，不显示任何错误消息**
>
> 如果客户端的结构与服务器的不匹配，则在编译过程中该错误可能不会生成任何错误，在运行时也不会出错。
>
> 请确保不在运行时中更改所组态的结构分配。必要时，可在读取和写入列表中对分配进行重新组态！

11.4.9 使用连接参数分配

11.4.9.1 创建和组态连接

利用 OPC UA 客户端的指令，可创建与 OPC UA 服务器交换数据的用户程序。为此，需要使用一系列系统数据类型。

为了简化对这些系统数据类型的操作，从 STEP 7 (TIA Portal) 版本 V15.1 开始，可对 OPC UA 客户端进行连接参数分配。

可以选择是否使用连接参数分配，并不强制要求使用。还可以手动创建所需系统数据类型。

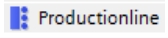
我们通过举例的方式让说明更易于理解，请参见“示例说明 (页 322)”。

打开连接参数分配

要组态与 OPC UA 服务器的连接，请按以下步骤操作：

1. 在“OPC UA 通信”(OPC UA communication) 区域，双击要在项目树中为其分配参数的客户端接口。

   对于示例组态：双击“ProductionLine”客户端接口。

   “创建客户端接口 (页 323)”部分介绍了如何创建客户端接口。
2. 如果选项卡尚未显示，请单击“特性”(Properties) 选项卡（巡视窗口）。

   STEP 7 现在显示 OPC UA 客户端指令的连接参数分配。

   “常规”(General) 选项卡会打开。
3. 单击“组态”(Configuration) 选项卡并设置与 OPC UA 服务器的连接。

设置连接参数

1. 为会话选择一个描述性名称。在本示例中，将选择名称“OPC UA Connection to ProductionLine”。
2. 在“地址”(Address) 字段中，输入用户程序（作为 OPC UA 客户端运行）要与之建立连接的 OPC UA 服务器的 IP 地址。在示例组态中，控制生产线的 CPU 的 IP 地址为“192.168.1.1”。将与该 CPU 的 OPC UA 服务器建立连接。为此，需要在“地址”(Address) 字段中输入 IP 地址。在这种情况下，OPC UA 服务器会使用默认端口 4840。
   或者，也可以在“地址”(Address)字段中输入有效的 DNS 名称。DNS 名称的长度限制为 242 个字符。
   如果地址无效，则会显示错误消息：“输入有效地址”(Enter a valid address)。
   如果“地址”(Address)、“端口”(Port) 和“路径”(Path) 字段的字符串长度超过 254 个字符，也会显示错误消息。
3. 在 OPC UA 服务器中输入路径，可限制对该路径的访问。该信息可选。但如果指定了服务器路径，则某些服务器仅建立一条连接。

   指定某个路径时，系统将在客户端接口内组态 DB 的“ServerEndpointUrl”条目中输入该路径。该条目由组件“OPC 示意前缀”(OPC Schematic Prefix)、“IP 地址”(IP address)、“端口号”(Port number)”和“服务器路径”(Server path) 组成，例如：“opc.tcp://192.168.0.10:4840/example/path”。

   下图显示了 OPC UA 服务器的 IP 地址条目：

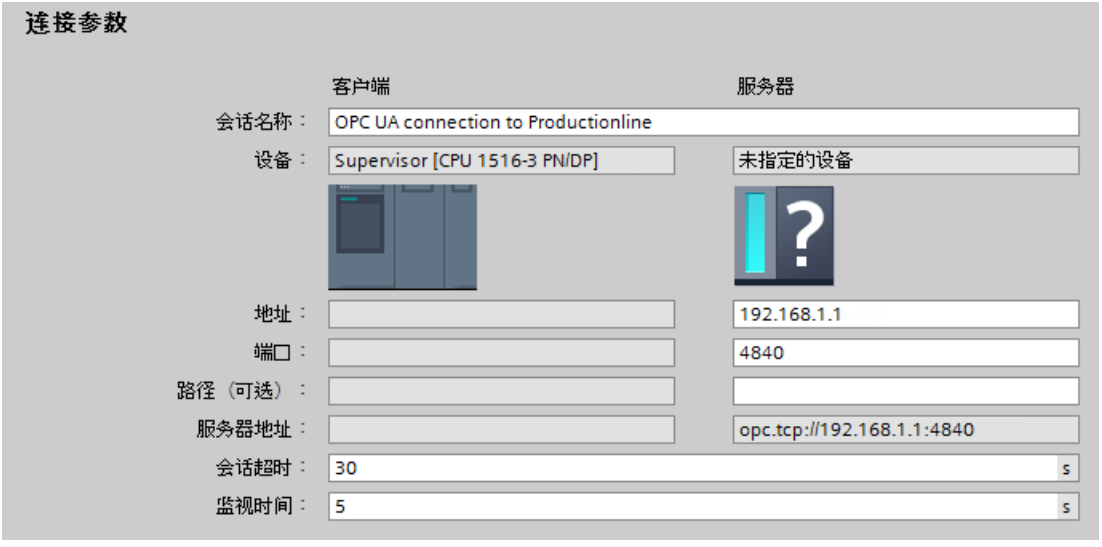


图 11-81 连接参数

4. 如果 OPC UA 服务器不使用标准端口 4840，则必须在此插入端口号。

   例如，如果要与之建立连接的 OPC UA 服务器使用该端口号，则在字段中输入数字 65535。
5. 此外，还接受会话超时（30 秒）和监视时间（5 秒）的默认设置。

设置安全参数

1. 单击“组态”(Configuration) 选项卡中的“安全”(Security) 区域。
   此区域包含与 OPC UA 服务器的连接的所有安全设置。

可进行以下设置，例如：

“常规”(General) 区域

安全模式：
从下拉列表中选择与 OPC UA 服务器的连接必须达到的安全模式。
如果服务器不满足所选模式的要求，将不建立连接。
可进行以下设置：

- 不安全：无安全连接！
- 签名：OPC UA 服务器和 OPC UA 客户端对数据传输进行签名（所有消息）：因此可检测到修改。
- 签名并加密：OPC UA 服务器和 OPC UA 客户端对数据传输进行签名和加密（所有消息）：

安全策略：
设置将为消息签名和加密使用的加密技术。
可进行以下设置：

- 不安全
- Basic128Rsa15
- Basic256
- Basic256Sha256

要组态安全连接，必须注意以下事项：

- 需要为客户端使用证书才能建立安全连接。
- 需要让服务器知晓该客户端证书。

相关操作步骤，请参见“处理客户端和服务器证书 (页 232)”部分“OPC UA 客户端的证书”下的内容。

“证书”(Certificates) 区域

客户端证书：
证书确认 OPC UA 客户端的真实性。
要选择证书，请单击以下符号：

STEP 7 会显示证书列表。
选择已让服务器知晓的证书。
单击带有绿色复选标记的符号。

或者创建新证书。此时，可单击“添加”(Add) 符号。
如果创建新证书，必须让服务器知晓该证书。

“用户认证”(User authentication) 区域

可为用户身份认证进行以下设置：

- 访客
- 用户名和密码
- 用户（TIA Portal - 安全设置）

更多信息，请参见“具有 OPC UA 功能权限的用户和角色 (页 242)”。

设置语言

String 类型的 UA 变量可通过 OPC UA 进行本地化，也就是说，文本（UA 变量的值）能够以不同的语言形式提供给服务器。例如，本地化文本可用于 DisplayName（节点名称）和 Description（描述）。

例如，在“组态”(Configuration) 选项卡的“语言”(Languages) 区域，可通过以下操作改变服务器返回文本的语言：

在“语言”(Languages) 区域中，输入连接建立期间服务器传送到客户端的语言数。

在第一行中输入的语言或与之关联的本地 ID（“语言代码”）是客户端的首选语言。

- 如果服务器能够以请求的语言提供 UA 变量，则会将该变量传送到客户端。
- 如果服务器不能以请求的语言提供 UA 变量，则会检查能否以在第二行中输入的语言（第一替代语言）提供 UA 变量。
- 服务器会逐个检查列表中的各条目，如果服务器既不能提供请求的语言，也不能提供替代语言，则将提供默认语言。

更多信息

与 OPC UA 服务器的连接发生故障的原因。常见问题解答 (https://support.industry.siemens.com/cs/cn/zh/view/109766709)

11.4.9.2 S7-1500 CPU 的客户端证书处理

客户端证书来自何处？

如果使用 S7-1500 CPU 的 OPC UA 客户端（OPC UA 客户端已启用），则可按照以下章节中的详细介绍，使用 STEP 7 V15.1 及更高版本为这些客户端创建证书。

如果使用来自制造商或 OPC 基金会的 UA 客户端，则会在安装期间或在首次调用程序时自动生成客户端证书。在 STEP 7 中，需要通过全局证书管理器导入这些证书，并在相应的 CPU 中使用。

如果自行编写 OPC UA 客户端程序，则可以通过程序生成证书。也可通过工具生成证书（如，使用 OpenSSL 或 OPC 基金会的证书生成器）：

- 有关使用 OpenSSL 的操作步骤，请参见此处："用户自己生成 PKI 密钥对和证书 (页 176)"。
- 有关使用 OPC 基金会的证书生成器的步骤，请参见此处："创建自签名证书 (页 175)"。

S7-1500 CPU 的 OPC UA 客户端证书

仅当 OPC UA 服务器将 OPC UA 客户端证书归类为可信任证书时，服务器与客户端之间才能建立安全连接。

因此，需要让服务器知晓该客户端证书。

在以下章节中，将介绍最初如何为 S7-1500 CPU 的 OPC UA 客户端生成证书，并提供给服务器。

1.生成并导出客户端证书

要进行安全连接，需生成一个客户端证书，如果服务器和客户端位于不同项目中，还需要导出该证书。

如果客户端和服务器位于相同项目中，则无需导出客户端以及进行后续导入。

要求

CPU 的 IP 接口已组态，IP 地址可用。

背景：在"主题备用名称 (SAN)"(Subject Alternative Name (SAN)) 中，输入用于访问系统中该 CPU 的 IP 地址。

创建 OPC UA 客户端接口

为 S7-1500 CPU 生成客户端证书的最简单方法是组态一个客户端接口。

为选择或生成客户端证书而提供的客户端接口的组态，参见"创建和组态连接 (页 336)"。

或者可按以下方法生成客户端证书：

1. 在项目树中，选择将用作客户端的 CPU。
2. 双击"设备组态"(Device configuration)。
3. 在该 CPU 的属性中，单击"保护和安全 > 证书管理器"(Protection & Security > Certificate manager)。
4. 在"设备证书"(Device certificates) 表格中，双击"<新增>"(<Add new>)。

   在 STEP 7 中，将打开一个对话框。

5. 单击"添加"(Add) 按钮。
6. 从"使用"(Usage) 列表选择"OPC UA 客户端"(OPC UA client) 条目。
7. 单击"确定"(OK)。

   此时，STEP 7 将在"设备证书"(Device certificates) 表格中显示该客户端证书。
8. 如果服务器位于另一项目中：右键单击此行，并从快捷菜单中选择"导出证书"(Export certificate)。
9. 选择该客户端证书的目标存储目录。

2.向服务器通告该客户端证书

用户需要将该客户端证书发送至服务器，以便允许建立安全连接。

为此，请执行以下操作步骤：

1. 如果客户端是在另一项目中组态的，并且已在该项目中创建并导出客户端证书：
   - 在服务器的本地证书管理器中，选择"使用证书管理器的全局安全设置"(Use global security settings for certificate manager) 选项。这将激活全局证书管理器。

     可以在用作服务器的 CPU 的特性"保护和安全 > 证书管理器"(Protection & Security > Certificate manager) 下找到此选项。
   - 如果项目未受保护，请在 STEP 7 的项目树中选择"安全设置 > 设置"(Security settings > Settings)，然后单击"保护此项目"(Protect this project) 按钮并登录。

     "全局安全设置"(Global security settings) 菜单项随即显示在 STEP 7 项目树的"安全设置"(Security setting) 下。
   - 双击"全局安全设置"(Global security settings)。
   - 双击"证书管理器"(Certificate manager)。

     STEP 7 将打开全局证书管理器。
   - 单击"设备证书"(Device certificates) 选项卡。
   - 在此选项卡的空白区域（而非证书上）中，右键单击鼠标。
   - 选择"导入"(Import) 快捷菜单。

     将显示用于导入证书的对话框。
   - 选择服务器信任的客户端证书。
   - 单击"打开"(Open)，导入证书。

     客户端证书现已包含在全局证书管理器中。请留意刚刚导入的客户端证书 ID。
2. 单击用作服务器的 CPU 的特性中的"常规"(General) 选项卡。
3. 单击"OPC UA > 服务器 > 安全 > 安全通道"(OPC UA > Server > Security > Secure Channel)。
4. 在"安全通道"(Secure Channel) 对话框中向下滚动至"受信客户端"(Trusted clients) 部分。
5. 双击表中空行的"<新增>"(<add new>)。随即会在该行中显示浏览按钮。
6. 单击该按钮。
7. 选择准备好的客户端证书。
8. 单击带有绿色复选标记的按钮。

9. 编译项目。
10. 将组态加载到 S7-1500 CPU（服务器）。

结果

服务器现已信任此客户端。如果还将服务器证书视为受信证书，则服务器和客户端之间可建立安全连接。

11.4.9.3 用户认证

对于 S7-1500 的 OPC UA 客户端接口，可设置 OPC UA 客户端中用户访问服务器时需通过的认证。为此，必须在所请求的 S7-1500 CPU 项目树的"OPC UA 通信 > 客户端接口"(OPC UA communication > Client interfaces) 中选择相应的客户端接口，然后在巡视窗口的"属性 > 组态 > 信息安全"(Properties > Configuration > Security) 中选择用户认证方式。

用户认证方式

可通过以下几种方式进行用户认证：

- **访客**

  此类用户无需进行身份验证（匿名访问）。CPU 将为该用户创建一个匿名会话，同时 OPC UA 服务器也不会对该客户端用户进行身份验证。

- **用户名和密码**

  此类用户需证明身份验证（非匿名访问）。OPC UA 服务器将检查客户端用户是否具备访问服务器的权限。通过用户名和正确的密码进行身份验证。客户端接口无法检查这些输入，即所有值都将接受为有效值。

  说明

  STEP 7 会将用户名和密码以未加密形式存储在数据块/背景数据块中。建议：对"用户（TIA Portal - 安全设置）"(User (TIA Portal - security settings)) TIA 项目使用用户认证。

- **用户（TIA Portal - 安全设置）**

  通过在项目中所输入的用户名列表中输入一个用户名进行验证。在项目树中的用户管理中，通过"安全设置 > 用户和角色"(Security Settings > Users and roles) 查看当前项目中已注册的用户名称。此外，也可输入其它用户名。

  用户可输入该项目用户管理中未列出的名称，或将该字段保留为空。仅当运行过程中相应的用户名出处不同（如，通过 HMI 或来自不同的 OPC UA 客户端）时，才需执行该操作。

“不安全”安全策略和通过用户名和密码进行身份验证

可执行以下组合设置：

“不安全”安全策略和通过用户名和密码进行身份验证

- S7-1500 的 OPC UA 服务器支持该组合设置。OPC UA 客户端可连接并加密认证数据，反之亦然。
- S7-1500 CPU 的 OPC UA 客户端也支持该组合设置：但在运行时，仅当通过电缆发送加密的认证数据时才能连接！

结果：使用以下组态，无法在运行时中建立连接。

- S7-1500 用作 OPC UA 客户端
- 当安全策略设置为“不安全”(="none") 时，不支持认证数据加密的 OPC UA 服务器。

更多信息

有关具有 OPC UA 功能权限的用户和角色的信息，请参见“具有 OPC UA 功能权限的用户和角色 (页 242)”部分。

11.4.9.4 使用组态连接

简介

本节介绍了如何为 OPC UA 指令使用组态连接（第三步）。

要求

- 已创建客户端接口，并已向该接口添加 PLC 变量和 PLC 方法，参见（“第一步 (页 323)”）。
- 已组态与 OPC UA 服务器的连接（第二步 (页 336)）。

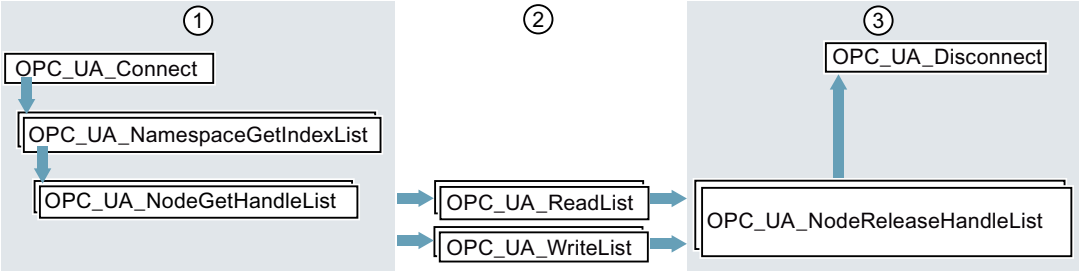
概述

要从 OPC UA 服务器读取数据或向 OPC UA 服务器写入数据，请使用以下指令：

- OPC_UA_Connect
- OPC_UA_NamespaceGetIndexList
- OPC_UA_NodeGetHandleList
- OPC_UA_ReadList or OPC_UA_WriteList
- OPC_UA_NodeReleaseHandleList
- OPC_UA_Disconnect

OPC UA 指令的顺序

下图显示了使用 OPC UA 指令读取或写入 PLC 变量时这些指令在用户程序中的调用顺序：



① 用于准备读取和写入操作的指令

② 读取和写入指令

③ 完成读取或写入操作后用于进行“清理”的指令
如果随后立即调用“OPC_UA_Disconnect”，则可省略“OPC_UA_NodeReleaseHandleList”指令。

图 11-82 读取和写入操作的调用顺序

如果使用的是客户端接口以及与 OPC UA 服务器的已组态连接，则 STEP 7 (TIA Portal) 会自动提供这些指令的参数。

下一节中介绍了具体操作步骤。

使用客户端接口和已组态连接

要使用已组态 OPC UA 连接，请按以下步骤操作：

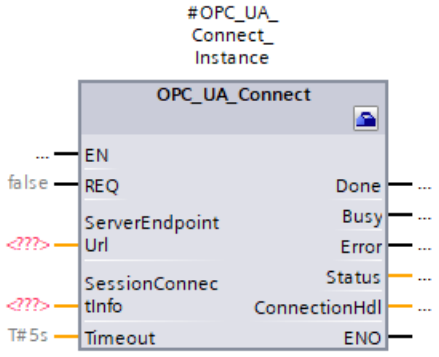
1. 在 TIA Portal 中打开用户程序。
2. 通过拖放的方式将“**OPC_UA_Connect**”指令移入程序编辑器。

   该指令将出现在 TIA Portal 中的“指令 > 通信 > OPC UA”(Instructions > Communication > OPC UA) 下方。
3. 选择指令的调用选项。

   示例使用多重实例。

   STEP 7 会在程序编辑器中显示指令。

   函数块图 (FBD) 编程语言编辑器使用以下显示：



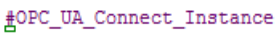
梯形逻辑 (LAD) 编程语言编辑器采用相似的方式显示指令。

4. 单击 FBD 或 LAD 编辑器中的工具箱符号。

   该符号位于指令标题中。

   如果使用 STL 或 SCL 编辑器：点击实例名称第一个字符下方的小绿方块：

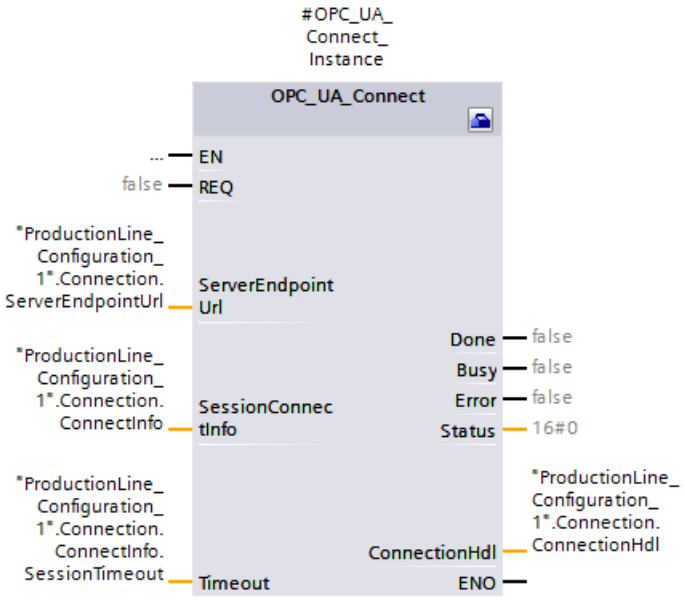
   示例 (页 322)使用“#OPC_UA_Connect_Instance”作为实例名称。

   STEP 7 会在单独的对话框中显示特性。

5. 对于“客户端接口”(Client interface)，选择要为指令使用的客户端接口。

   在本示例中，选择“ProductionLine”客户端接口。

   STEP 7 现在通过 OPC_UA_Connect 指令的参数与“ProductionLine”客户端接口互连。

   在 OPC UA 客户端示例 (页 322)中，使用“ProductionLine”作为接口与 OPC UA 服务器“ProductionLine”进行数据交换。

6. 通过拖放的方式将“**OPC_UA_NamespaceGetIndexList**”指令移入程序编辑器。

   该指令将出现在 TIA Portal 中的“指令 > 通信 > OPC UA”(Instructions > Communication > OPC UA) 下方。

   选择“多重实例”(Multi-instance) 调用选项。

   如果编辑器尚未打开，请单击工具箱符号（LAD 和 FBD）或实例名称下方的小绿框（STL 和 SCL）。

   选择要使用的客户端接口（示例中为“ProductionLine”）。

   STEP 7 现在自动与“OPC_UA_NamespaceGetIndexList”指令的所有参数互连：

7. 通过拖放的方式将“**OPC_UA_NodeGetHandleList**”指令移入程序编辑器。

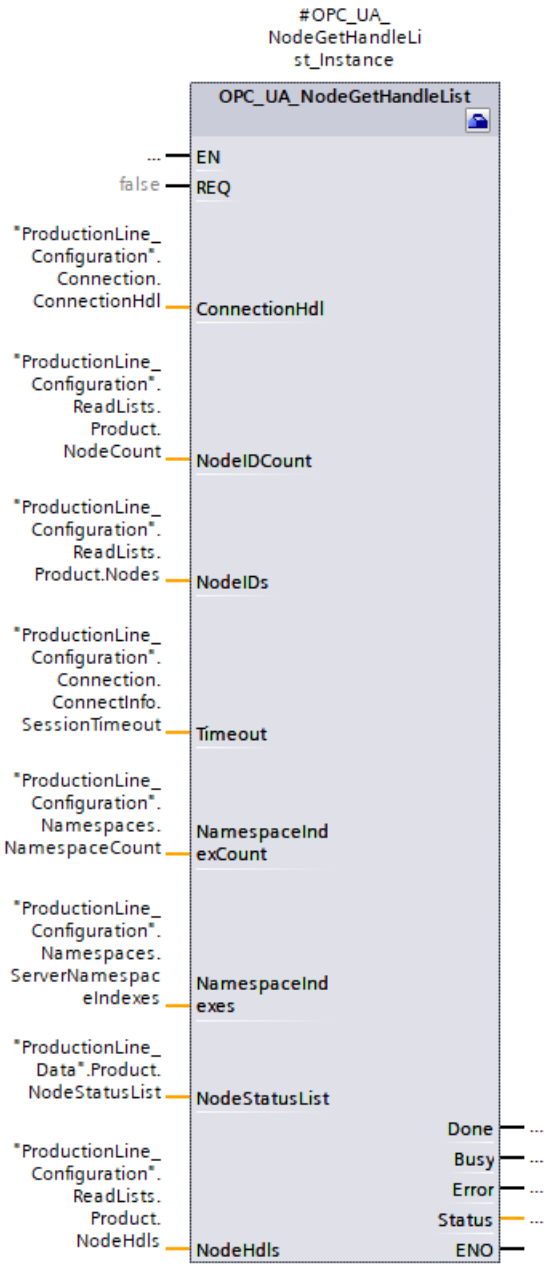
   选择“多重实例”(Multi-instance) 调用选项。

   如果编辑器尚未打开，请单击工具箱符号（LAD 和 FBD）或实例名称下方的小绿框（STL 和 SCL）。

选择要使用的客户端接口。在本示例中，使用“ProductionLine”客户端接口。

在“数据访问 > 读取/写入列表”(Data access > Read/Writelist) 中，选择需使用的读写列表（本示例中为读取列表“Product”）。

STEP 7 现在自动与“OPC_UA_NodeGetHandleList”指令的所有参数互连：



如果要将数据写入到 OPC UA 服务器，请在“数据访问 > 写入列表”(Data access > Write list) 下选择要使用的写入列表（示例中为“ProductionStatus”写入列表）。

8. 通过拖放的方式将“**OPC_UA_ReadList**”指令移入程序编辑器。

   选择“多重实例”(Multi-instance) 调用选项。

   如果编辑器尚未打开，请单击工具箱符号（LAD 和 FBD）或实例名称下方的小绿框（STL 和 SCL）。

选择要使用的客户端接口。示例使用"ProductionLine"客户端接口。

在"数据访问 > 读取列表"(Data access > Read list) 中，选择需使用的读取列表（本示例中为"Product"读取列表）。

STEP 7 现在自动与"OPC_UA_ReadList"指令的所有参数互连。

如果要将数据写入到 OPC UA 服务器，请使用"**OPC_UA_WriteList**"指令，并在"数据访问 > 写入列表"(Data access > Write list) 下选择要发送到服务器的变量列表（示例中为"ProductionStatus"写入列表）。

9. 如果要将其它读取列表或写入列表用作用户程序中受程序控制的列表，请通过拖放操作将"**OPC_UA_NodeReleaseHandleList**"指令移入程序编辑器。

   选择要使用的客户端接口。

   现在选择要发布的读取列表或写入列表。由于重新注册比较耗时，请仅释放很少使用的读取或写入列表。

   然后，使用"OPC_UA_NodeGetHandleList"指令重复执行第 7 步开始的步骤。

10. 通过拖放的方式将"**OPC_UA_Disconnect**"指令移入程序编辑器。

    选择"多重实例"(Multi-instance) 调用选项。

    如果编辑器尚未打开，请单击工具箱符号（LAD 和 FBD）或实例名称下方的小绿框（STL 和 SCL）。

    选择要使用的客户端接口。在本示例中，使用"ProductionLine"客户端接口。

    STEP 7 现在自动与"OPC_UA_Disconnect"指令的所有参数互连。

支持的指令

对于下列指令，如果使用的是客户端接口以及与 OPC UA 服务器的已组态连接，则 STEP 7 会自动提供参数。

- OPC_UA_Connect
- OPC_UA_NamespaceGetIndexList
- OPC_UA_NodeGetHandleList
- OPC_UA_MethodGetHandleList
- OPC_UA_MethodReleaseHandleList
- OPC_UA_ReadList
- OPC_UA_WriteList
- OPC_UA_MethodCall
- OPC_UA_NodeReleaseHandleList
- OPC_UA_Disconnect

紧凑指令

自 TIA Portal V17 起，OPC UA 提供紧凑指令，这些指令总结了写入作业/读取作业/方法调用和建立连接的信息：

- OPC_UA_ReadList_C 用于生成连接和读取变量
- OPC_UA_WriteList_C 用于生成连接和写入变量
- OPC_UA_MethodCall_C 用于生成连接和调用方法

有关紧凑指令服务的信息，请参见 TIA Portal 在线帮助。

11.5 提示和建议

11.5.1 订阅规则

以下规则适用于订阅部分：

- 根据不同的采样和发布时间间隔对订阅分组，并将被监视的元素（变量）分配到这些组中。

  示例：创建一个发布时间间隔较长（如 5 秒）的订阅和一个发布时间间隔较短（如 0.1 秒）的订阅。
- 禁用不需要的订阅。

  原因："已禁用"订阅模式可以降低资源消耗。
- 为进一步优化资源利用率，请缩短客户端的订阅超时。订阅超时不能直接指定，此时间由服务器确认的订阅设置"PublishingInterval"和"LifetimeCount"决定。
  背景：客户端创建了订阅并且会话终止时，订阅仍保留在服务器中并占用存储器资源。OPC UA 服务器仅在因订阅超时而结束订阅周期时释放所需资源。
- 需注意相应 S7-1500 CPU 可监视的订阅项目的最大数量。

  在相应 CPU 的技术规范中，可以找到该信息。此信息基于 1 秒的采样/发布时间间隔。

  有关更多信息，请参见常见问题解答 109755846 (https://support.industry.siemens.com/cs/cn/zh/view/109755846)。
- 针对 OPC UA 客户端和 OPC UA 服务器，选择相同的采样和发布时间间隔。
- 避免将数组和结构作为订阅的元素（如果过程允许）。

  原因：即使数组/结构中有一个值发生变化，也需要传送整个结构，从而产生不必要的通信负载。
- 偶尔发生与所需的采样率不兼容的情况，S7-1500 CPU 的 OPC UA 服务器根据 OPC UA 规范使用"GoodOverload"错误代码进行确认，另请参见 TIA Portal 帮助。不同的 OPC UA 客户端以不同的方式处理不等于"0"的"Good"错误代码。请注意此行为，必要时根据上述措施降低通信负载。

更多信息

有关设置订阅服务器的信息，请参见"服务器的订阅设置 (页 228)"部分。

11.5.2 面向用户程序的规则

OPC UA 的用户程序

以下规则适用于用户程序：

- 如果您的应用程序允许，并且通信负载过高，应该设置周期性 OB 的最小时间。

  优势：

  – 周期时间多数情况下是不变的

  – 整个过程中 CPU 可以分配更多的时间处理通信任务

  提示：使用指令“Runtime_Info”；模式 21 或模式 25（参见 TIA Portal 帮助）分析 CPU 利用率（例如通信）。

- 减少可以通过 OPC UA/HMI 访问的变量或数据块的数量。默认情况下，创建变量/DB/IDB 时，来自 OPC UA/HMI 的所有变量都可以访问。在运行状态下加载时，此措施可以改进性能。

  提示：通过在 TIA Portal 中使用详细对象显示，可以轻松将非 OPC-UA 相关数据块标记为“无法从 OPC UA 访问”(not accessible from OPC UA)。

- 只有通过 OPC UA 方法才能实现一致的数据传输，不受简单数据类型的限制。如果使用其他 OPC UA 功能（订阅、读写），必须确保应用中的数据一致性。

- OPC UA 提供“RegisterNodes”服务对相同的变量进行重复读写。服务器可使用该服务准备对变量的优化访问。作为 OPC UA 客户端的 S7-1500 的指令“OPC_UA_NodeGetHandleList”可隐式调用该服务，使服务器准备好进行优化访问（在 OPC UA 用法“注册的读写”中）。

在 TIA Portal 中调用详细的对象显示

要调用详细的对象显示，请执行以下步骤：

1. 在门户视图中切换到“PLC 编程”(PLC Programming) 门户。
2. 选择“显示所有对象”(Show all objects)：
3. 在选择窗口中切换到“详细信息”(Details) 选项卡。
4. 在“DB 从 OPC UA 可访问”(DB accessible from OPC UA) 列中，禁用各个对象的 OPC UA 可访问性。

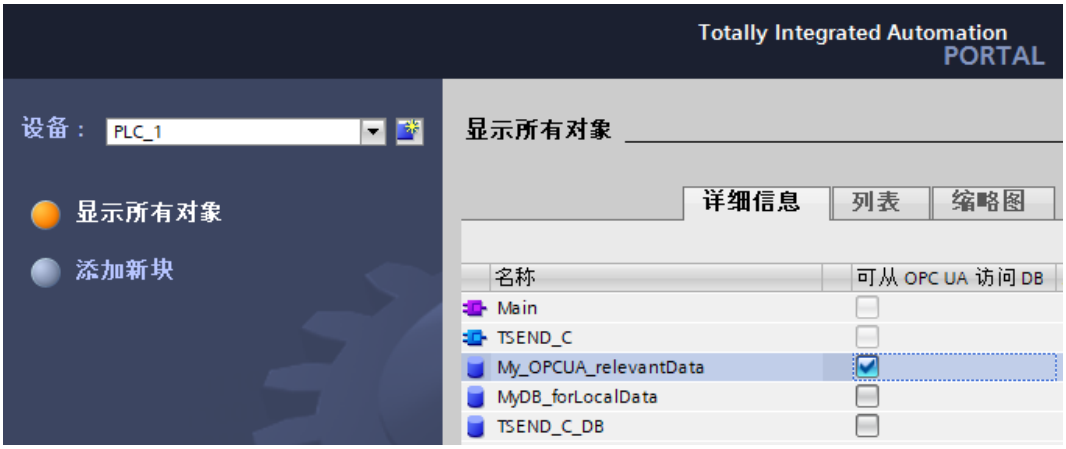


图 11-83 在 TIA Portal 中调用详细的对象显示

11.5.3 OPC UA 通信的模板副本

OPC UA 接口的模板副本

要多次使用的 OPC UA 服务器和 OPC UA 客户端的接口可存储在项目库或全局库中。项目库中的模板副本只能在项目中使用。在全局库中创建模板副本时，模板副本可用于不同的项目中。

支持 OPC UA 的 CPU 根据 OPC UA 服务器的 3 种接口类型加以区分：

- 标准 OPC UA 服务器接口
- 配套规范接口
- 命名空间引用

将 OPC UA 接口添加到项目树的"OPC UA 通信"(OPC UA Communication) 下方时，每个接口类型都会获得自己的符号。模板副本会使用相同符号。

创建单个模板副本或包含多个接口的模板副本。

基于选择创建多个模板副本

选择一个或多个元素并使用它们来创建各个模板副本

1. 在"库"(Libraries) 任务卡中打开库。
2. 选择所需的元素。
3. 使用拖放操作，将这些元素移到"模板副本"(Master copies) 文件夹或"模板副本"(Master copies) 的任意子文件夹中。

基于选择创建模板副本

选择多个元素并创建包含所有选中元素的单个模板副本。

1. 将要创建为模板副本的元素复制到剪贴板中。
2. 右键单击“模板副本”(Master copies) 文件夹或库中的任意一个子文件夹。
3. 在快捷菜单中，选择“作为单个模板副本粘贴”(Paste as a single master copy) 命令。

如果多个接口从 OPC UA 服务器或 OPC UA 客户端添加到模板副本，库中的标签和符号也会相应地更改。

会显示带“+”的符号，而不是简单的符号。

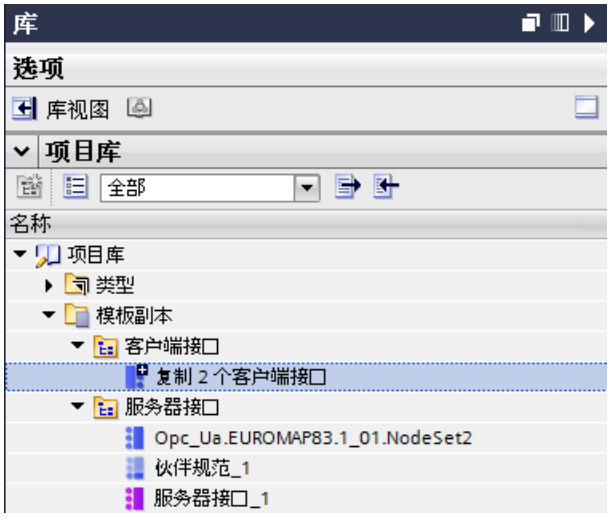


图 11-84 在 STEP 7 中创建复制模板

更多信息

有关创建用户自定义服务器接口的信息，请参见“创建用户自定义服务器接口 (页 258)”部分。

通过 DHCP 寻址

12

为了提供面向未来的高效灵活的自动化解决方案，制造领域中越来越多的组件开始支持 IT 标准。凭借全球以太网标准、集成通信和多功能性，具有 IT 支持的自动化解决方案成为可满足用户需要的经济型解决方案。借助 S7-1500 CPU 通信选项的功能扩展，可以在使用系统或机器时获得更高的自由度。通过使用 IT 技术提高自动化的效率。对于固件版本为 V2.9 及以上版本的 S7-1500 CPU，通过 DHCP 的引入以及 DNS 的扩展，可以在设计自动化解决方案时实现更高的灵活性。

对于 S7-1500 CPU 的接口，可以设置为通过 DHCPv4 服务器（以下称为 DHCP 服务器）获取地址参数，例如 IP 地址以及子网掩码。

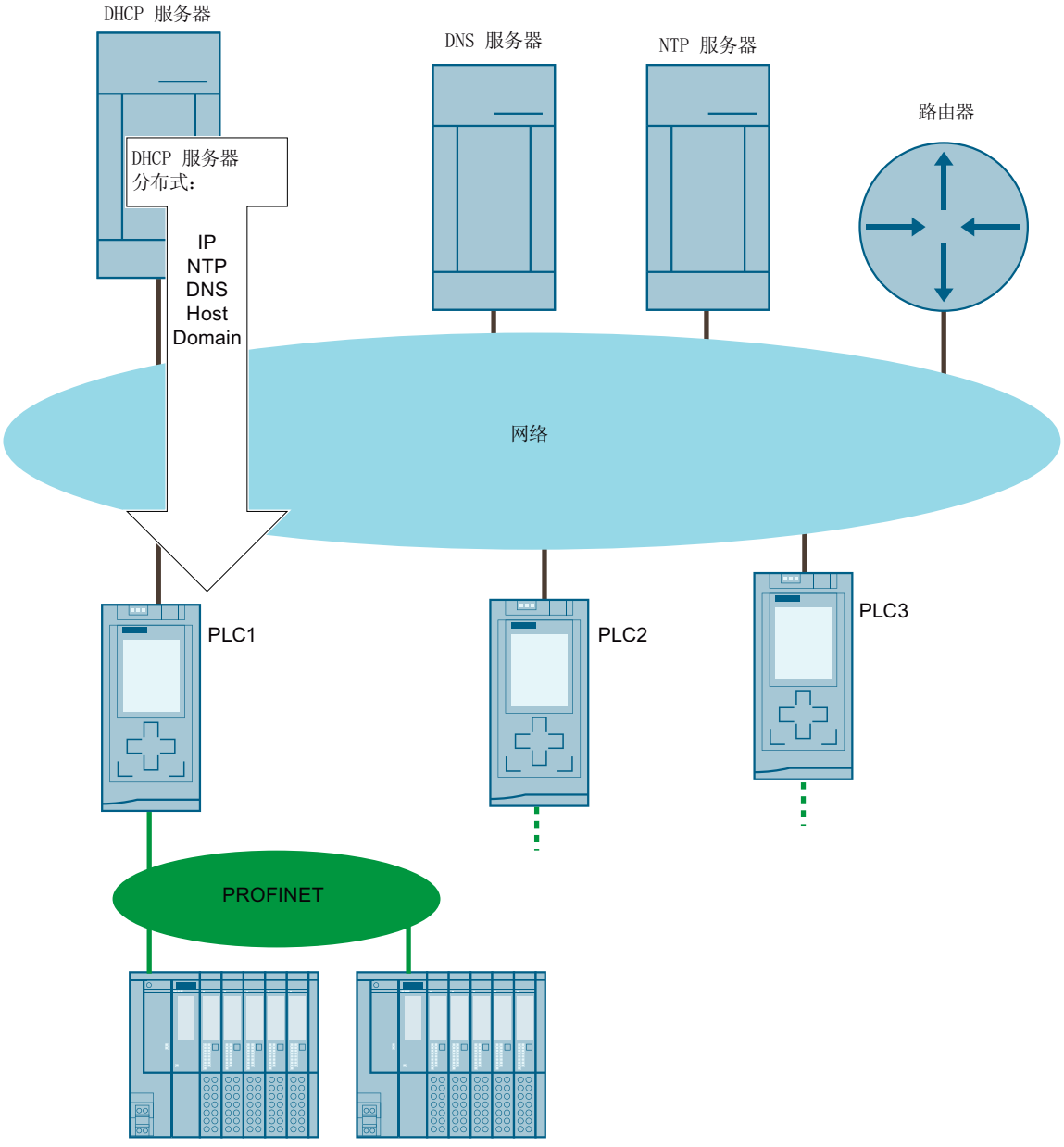


图 12-1 DHCP 概述

应用范围

- 在管理型 IT 环境中使用 S7-1500 CPU
- 在模块化制造结构中添加新设备

12.1 DHCP 的地址分配原则

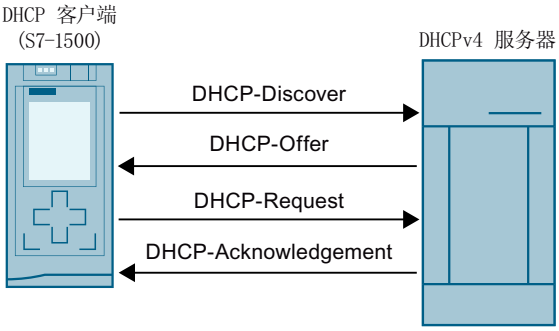
组态要求

必须满足以下要求，S7-1500 CPU 的 PROFINET 接口才可以通过 DHCP 服务器获取 IP 地址参数：

- 已组态通过 DHCP 服务器进行地址分配。
  激活 DHCP (页 371)
- 没有为该接口组态 PROFINET IO 通信。

DHCP 地址分配原则

将项目下载到 CPU 中，此过程将立即开始，或者已组态 DHCP 地址分配功能的 CPU 接通并启动后，DHCP 分配过程便会开始：



DHCP Discover	DHCP 客户端通过广播搜索合适的 DHCP 服务器。DHCP 客户端使用组态的客户端 ID 或 MAC 地址作为 DHCP 服务器上的身份标识。
DHCP Offer	DHCP 服务器向 DHCP 客户端提供 IP 地址参数（IPv4 地址、子网掩码、可选的默认路由器），并在必要时提供其它数据（选项）。
DHCP Request	DHCP 客户端请求 DHCP offer 中提供的 IP 地址参数和选项。 S7-1500 CPU 的 DHCP 客户端始终接受满足要求的 DHCP 服务器的第一个 DHCP offer（IP 地址以及子网掩码）。
DHCP Acknowledgment	DHCP 服务器确认并传输 DHCP offer 中提供的 IP 地址参数和选项。 DHCP 服务器还向 DHCP 客户端发出通知，指明可以使用地址参数的时间（租用时间）。

图 12-2 DHCP 的地址分配原则

IP 地址参数和选项存储在 CPU 的装载存储器中。CPU 常规复位或重新启动后，将通过 DHCP 重新获得 IP 地址参数和选项。

DHCP 地址分配选项

对于 S7-1500 CPU，可以组态通过 DHCP 服务器获得以下选项：

- 最多四个 DNS 服务器的地址

  通过 DHCP 获取 DNS 服务器的地址 (页 373)
- 最多四个 NTP 服务器的地址

  通过 DHCP 获取 NTP 服务器的地址 (页 373)
- 主机和域名

  通过 DHCP 获取主机和域名 (页 374)

如有必要，DHCP 服务器还提供路由器地址（默认网关）。

S7-1500 CPU 可以使用 DHCP 地址参数的时间

除地址参数外，DHCP 服务器还会向 S7-1500 CPU（DHCP 客户端）发出通知，指明租用时间。租用时间定义 CPU 可以使用地址参数的时间。

租用时间到期后，CPU 返还分配的地址参数。CPU 通过内部时间监视功能来监视租用时间。

在租期到期前的特定时间点，CPU 可以选择延长租用时间：

- 续租：租用时间过半时：CPU 联系原始 DHCP 服务器，并要求延长租用时间。原始 DHCP 服务器可以确认现有租用时间或分配新的租用时间。使用新的租用时间，重置 CPU 中的时间监视。
- 重新绑定：租用时间已消耗 7/8：CPU 通过广播联系所有可用的 DHCP 服务器，并要求延长租用时间。DHCP 服务器可以确认现有租用时间或分配新的租用时间。使用新的租用时间，重置 CPU 中的时间监视。

  如果在重新绑定过程中 DHCP 服务器发出否定响应，或没有 DHCP 服务器进行响应，则 CPU 将在租用时间耗尽后返还地址参数。

如果在租用时间到期后 CPU 已返回地址参数，则 CPU 将使用新的 DHCP Discover 来启动新的 DHCP 寻址周期。

更多信息

有关组态客户端 ID 的信息，请参见"组态客户端 ID (页 371)"部分。

12.2 DHCP 与 DNS

自 STEP 7 V17 起，S7-1500 CPU 支持在基于名称的通信 (DNS) 中使用主机名和域地址参数。

对于特定的通信服务，通过主机名和域组成的完整名称执行基于名称的寻址具有实用价值：

- 可使用完整名称对 CPU 进行寻址，例如，通过 OPC UA 的开放式用户通信。在通过 DHCP 服务器动态分配 IP 地址时，始终可以通过 DNS 名称进行唯一寻址。
- S7-1500 CPU 的证书可能包含完整名称，例如，用于 OPC UA 通信、Web 服务器、安全通信。
  - 只有在 STEP 7 中为 S7-1500 CPU 组态主机名和域时，才能在项目的设备证书中输入完整名称作为使用者替代名称 (SAN)。
  - 通过 DHCP 获得主机名和/或域后，或者通过用户程序分配主机名和/或域后，完整名称将不会存储在项目的设备证书中。

为 CPU 设置 DNS 组态的方式取决于在网络中分配主机名和域的方式。

- 主机名和域的集中分配

  例如，通过组态的 DNS 服务器在网络中集中分配主机名和域。在 STEP 7 中组态 CPU 通过 DHCP 获取主机名和域。

在以下组态中，仅在 S7-1500 CPU 中组态了客户端 ID。在分配 DHCP 地址时，DHCP 服务器将主机名和域选项返回给 CPU。

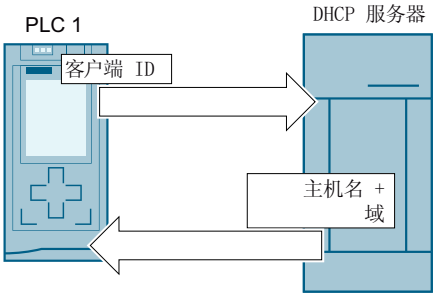


图 12-3 通过 DHCP 获取主机和域名

对于此组态，必须首先在 STEP 7 中激活主机名和域组态。然后，组态通过 DHCP 获取主机和域名。

通过 DHCP 获取主机和域名 (页 374)

- 主机名和域的本地分配

  可以在 STEP 7 中组态主机名和域，或者在用户程序中分配主机名和域。

说明

通过 DHCP 获取的数据的有效性

如果在用户程序中更改主机名和/或域，则通过 DHCP 获取的所有数据（IP 套件、主机名、域、NTP 服务器和 DNS 服务器）都会失效，并且会从 DHCP 服务器再次获取。因此，仅应在紧急情况下而不是运行期间更改主机名/域。

如果接口的 IP 地址发生更改，则所有连接都会中断。

在以下组态中，除了客户端 ID 外，还在 S7-1500 CPU 中组态了主机名和域。在分配 DHCP 地址时，CPU 将客户端 ID 以及主机名和域提供给 DHCP 服务器。DHCP 服务器接收要更新的信息，例如带有 CPU 地址数据的 DNS 服务器。

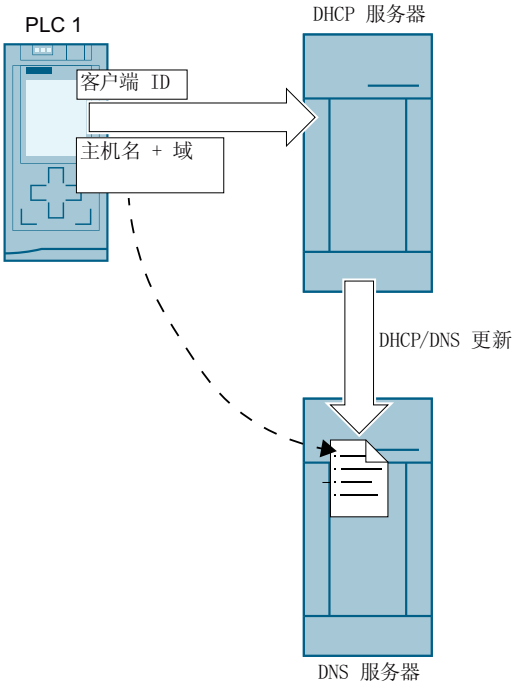


图 12-4 组态主机和域名

对于此组态，必须首先在 STEP 7 中激活主机名和域组态。然后在 STEP 7 中组态主机名和域。

- 域的集中分配和主机名的本地分配。
  - 在 STEP 7 中组态 CPU 通过 DHCP 获取域。
  - 可以在 STEP 7 中组态主机名，或者通过用户程序分配主机名。

在以下组态中，除了客户端 ID 外，还在 S7-1500 CPU 中组态了主机名。在分配 DHCP 地址时，CPU 将客户端 ID 以及主机名提供给 DHCPv4 服务器。DHCP 服务器将域选项提供给 CPU。

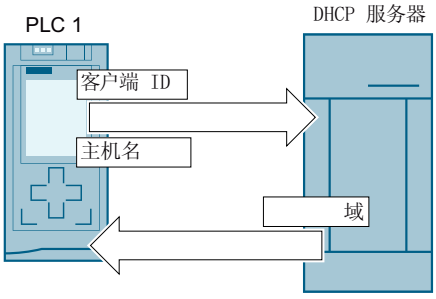


图 12-5 组态主机名，通过 DHCP 获取域名

对于此组态，必须首先在 STEP 7 中激活主机名和域组态。然后在 STEP 7 中组态主机名，并组态通过 DHCP 获取域。

要求

- 已经在 S7-1500 CPU 的至少一个接口中激活了 DHCP 地址分配。

组态主机和域名

要在 STEP 7 中激活主机名和域组态，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > 主机和域名 > 主机和域名组态”(Advanced configuration > Host and domain name > Host and domain name configuration)。
3. 选中“启用主机名和域”(Enable host name and domain) 复选框。

在 STEP 7 中组态主机名。

要在 STEP 7 中组态主机名，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > 主机和域名 > 主机和域名组态 > 主机名”(Advanced configuration > Host and domain name > Host and domain name configuration > Host name)。
3. 对于“主机名组态：”(Host name configuration:)，从下拉列表中选择“在项目中设置主机名”(Set host name in the project)。
4. 在“主机名：”(Host name:) 下输入主机名。
   - 输入所需的主机名。
   - 如果选中“主机名与设备名称相同”(Host name identical to device name) 复选框，则 STEP 7 会自动将设备名称分配为主机名。

只有在 STEP 7 中组态主机和域名后，“全名：”(Full name:) 下才会显示全名。

在用户程序中分配主机名

要在用户程序中分配主机名，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > 主机和域名 > 主机和域名组态 > 主机名”(Advanced configuration > Host and domain name > Host and domain name configuration > Host name)。
3. 在“主机名组态：”(Host name configuration:) 下，从下拉列表中选择“在设备上直接设置主机名（例如，PLC 程序、显示屏）”(Set hostname directly on the device (e.g. PLC program, display))。
4. 在用户程序中调用指令“CommConfig”。DATA 参数必须指向用于定义主机名的 UDT“Conf_Hostname”。

有关“CommConfig”指令和 UDT“Conf-Hostname”的更多信息，请参见 STEP 7 在线帮助。

在 STEP 7 中组态域

要在 STEP 7 中组态域，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > 主机和域名 > 主机和域名组态 > 域”(Advanced configuration > Host and domain name > Host and domain name configuration > Domain)。
3. 对于“域组态：”(Domain configuration:)，从下拉列表中选择“在项目中设置域”(Set domain in the project)。
4. 在“域：”(Domain:) 下输入所需的域。

只有在 STEP 7 中组态主机和域名后，“全名：”(Full name:) 下才会显示全名。

在用户程序中分配域

要在用户程序中分配域，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > 主机和域名 > 主机和域名组态 > 域”(Advanced configuration > Host and domain name > Host and domain name configuration > Domain)。
3. 在“主机名组态：”(Hostname configuration:) 下，从下拉列表中选择“在设备上直接设置域（例如，PLC 程序、显示屏）”(Set domain directly on the device (for example PLC program, display))。
4. 在用户程序中调用指令“CommConfig”。DATA 参数必须指向用于指定域名的 UDT“Conf_Domainname”。

有关“CommConfig”指令和 UDT“Conf-Domainname”的更多信息，请参见 STEP 7 在线帮助。

主机名、域和客户端 ID 的最大长度规则

请注意以下最大长度（以字节为单位）。一个字节对应一个字符：

- 主机名：最多 63 个字节
- 域：最多 252 个字节
- 客户端 ID：最多 254 个字节
- 主机 + 域名：最多 254 个字节
- 主机 + 域名 + 客户 ID：最多 260 个字节

  仅在必须将主机名和域名发送到 DHCP 服务器时适用。

12.3 激活 DHCP

要求

- S7-1500 CPU 固件版本 V2.9 或更高版本

操作步骤

要为 S7-1500 CPU 的 PROFINET 接口激活 DHCP，请按以下步骤操作：

1. 在 STEP 7 中，选择 S7-1500 CPU 的 PROFINET 接口。
2. 在接口属性中，导航至“以太网地址 > Internet 协议版本 4 (IPv4)”(Ethernet addresses > Internet Protocol Version 4 (IPv4))。
3. 选择选项“DHCP 服务器的 IP 地址”(IP address of DHCP server)。

结果

接口设置完毕，该接口现在可以通过 DHCP 服务器获取 IP 地址。

在 S7-1500 CPU 上，将“使用 MAC 地址作为客户端 ID”(Use MAC address as client ID) 设置为 DHCP 的操作模式。有关如何调整客户端 ID 的信息，请参见“组态客户端 ID (页 371)”。

12.4 组态客户端 ID

客户端 ID

S7-1500 CPU 始终使用客户端 ID（DHCP 选项 61）向 DHCP 服务器标识自己的身份。客户端 ID 具体取决于接口。

对于客户端 ID，S7-1500 CPU 支持以下两种操作模式：

- 使用 MAC 地址作为客户端 ID：使用 CPU 的 MAC 地址作为 DHCP 客户端的客户端 ID。注意，如果在此操作模式下执行 CPU 的设备更换，则 MAC 地址以及客户端 ID 会更改。
- 用户自定义客户端 ID：使用此选项，可以在 STEP 7 的组态中指定客户端 ID。此外，还可以选择在运行期间修改客户端 ID，例如，在用户程序中使用“CommConfig”指令执行修改。如果在此操作模式下执行 CPU 的设备更换，则会为新 CPU 分配已组态的客户端 ID。

要求

- 接口已激活 DHCP 地址分配。

组态客户端 ID

要在 STEP 7 中组态客户端 ID，请按以下步骤操作：

1. 在 STEP 7 中，选择 S7-1500 CPU 的 PROFINET 接口。
2. 在接口属性中，导航至“以太网地址 > Internet 协议版本 4 (IPv4) > DHCP 服务器的 IP 地址”(Ethernet addresses > Internet Protocol Version 4 (IPv4) > IP address of DHCP server)。
3. 对于“操作模式：”(Operating mode:)，从下拉列表中选择所需的操作模式：
   - 使用 MAC 地址作为客户端 ID（默认设置）
   - 用户自定义客户端 ID

   如果选择了选项“使用 MAC 地址作为客户端 ID”(Use MAC address as client ID)，则操作步骤已完成。对于“用户自定义客户端 ID”(User-defined client ID)，继续执行步骤 4。
4. 为“客户端 ID”(Client ID) 输入有效的客户端 ID。
   - 在此区域中允许使用 7 位 ASCII 字符串（0x21 到 0x7e）。
   - 某些 DHCP 服务器需要加一个前导“0”（如，某些 SCALANCE 设备）。这时，需在客户端 ID 前输入“\ 0”。
   - 也可以将字段留空。在这种情况下，必须选中“可以在运行系统中更改客户端 ID”(Client ID can be changed at runtime) 复选框。
5. 为了在运行系统中对用户自定义客户端 ID 进行修改，需选中“可以在运行系统中更改客户端 ID”(Client ID can be changed at runtime) 复选框。

在运行系统中更改客户端 ID

可以使用“CommConfig”指令通过用户程序更改客户端 ID。调用该指令。DATA 参数必须指向 UDT“Conf_ClientId”或 UDT“Conf_ClientId_Opaque”。必须在 UDT 中指定客户端 ID。

如果在 STEP 7 的组态中将用户自定义客户端 ID 留空，则 CPU 将使用 MAC 地址作为客户端 ID，直到第一次修改此客户端 ID。

说明

通过 DHCP 获取的数据的有效性

如果使用“CommConfig”更改 ClientId，则通过 DHCP 获取的所有数据都将失效：IP 套件、域名、NTP 服务器和 DNS 服务器。因此，仅应在紧急情况下而不是运行期间更改客户端 ID。

有关“CommConfig”指令及 UDT“Conf_ClientId”和“Conf_ClientId_Opaque”的更多信息，请参见 STEP 7 在线帮助。

12.5 通过 DHCP 获取 DNS 服务器的地址

要求

- 已经在 S7-1500 CPU 的至少一个接口中激活了 DHCP 地址分配。

通过 DHCP 从 DNS 服务器获取地址

要通过 DHCP 获取最多 4 个 DNS 服务器的地址，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > DNS 组态 > 服务器列表”(Advanced configuration > DNS configuration > Server list)。
3. 对于“通过 DNS 解析名称：”(Name resolution via DNS:)，从下拉列表中选择“远程设置 DNS 服务器（例如 DHCP）”(Set DNS server remotely (e.g. DHCP))。

结果：如果 DHCP 服务器提供来自 DNS 服务器的地址作为选项，则 CPU 最多使用 4 个地址。

12.6 通过 DHCP 获取 NTP 服务器的地址

要求

- 已经在 S7-1500 CPU 的至少一个接口中激活了 DHCP 地址分配。

通过 DHCP 从 NTP 服务器获取地址

要通过 DHCP 获取最多四个 NTP 服务器的地址，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航到“时钟 > 时间同步 > NTP 模式”(Time of day > Time synchronization > NTP mode)。
3. 对于“时间同步：”(Time synchronization:)，从下拉列表中选择“远程设置 NTP 服务器（例如 DHCP）”(Set NTP server remotely (e.g. DHCP))。

结果：如果 DHCP 服务器提供来自 NTP 服务器的地址作为选项，则 CPU 最多使用 4 个地址。

12.7 通过 DHCP 获取主机和域名

要求

- 已经在 S7-1500 CPU 的至少一个接口中激活了 DHCP 地址分配。
- 已在 STEP 7 中激活了主机名和域组态。

通过 DHCP 获取主机名

要通过 DHCP 获取主机名，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > 主机和域名 > 主机和域名组态 > 主机名”(Advanced configuration > Host and domain name > Host and domain name configuration > Host name)。
3. 对于“主机名组态：”(Host name configuration:)，从下拉列表中选择“远程设置主机名（例如 DHCP）”(Set host name remotely (e.g. DHCP))。

结果：如果 DHCP 服务器提供主机名作为选项，则 CPU 使用该主机名。

通过 DHCP 获取域

要通过 DHCP 获取域，请按以下步骤操作：

1. 在 STEP 7 中选择 S7-1500 CPU。
2. 在 CPU 的属性中，导航至“高级组态 > 主机和域名 > 主机和域名组态 > 域”(Advanced configuration > Host and domain name > Host and domain name configuration > Domain)。
3. 对于“域组态：”(Domain configuration:)，从下拉列表中选择“远程设置域（例如 DHCP）”(Set domain remotely (e.g. DHCP))。

结果：如果 DHCP 服务器提供域作为选项，则 CPU 使用该域。

路由

13

13.1 S7-1500 CPU 的路由机制概述

下表列出了 S7-1500 CPU 路由机制的概要信息。

路由机制	说明	应用	部分
S7 路由	S7 路由就是跨 S7 子网传输数据。可以跨越几个 S7 子网将信息从发送方传送到接收方。	下载用户程序 加载硬件配置 执行测试和诊断函数	S7 路由 (页 376)
IP 转发	IP 转发是一种在两个已连接 IP 子网之间转发 IP 数据包的设备功能。	轻松实现控制级到现场级的访问，以便对设备进行组态和参数分配，例如通过 PDM 或 Web 浏览器。 简化远程访问设备的集成，例如，在远程维护或固件更新期间进行诊断。	IP 转发 (页 380)
数据记录路由	可以通过 PROFINET，从工程师站将数据发送到多个网络中的现场设备。由于工程师站使用标准化的记录对现场设备寻址且这些记录通过 S7 设备来路由，因此使用“数据记录路由”这个术语来表示这种路由。	例如，在使用不同厂商的现场设备时，可使用数据记录路由。为进行组态和诊断，将使用标准数据记录 (PROFINET) 来寻址现场设备。	数据记录路由 (页 386)

13.2 S7 路由

S7 路由的定义

S7 路由就是跨 S7 子网传输数据。可以跨越几个 S7 子网将信息从发送方传送到接收方。S7 路由器提供从一个 S7 子网到一个或多个其它子网的网关。S7 路由器具有连接至相应 S7 子网的接口。S7 路由可通过各种 S7 子网（PROFINET/工业以太网和/或 PROFIBUS）实现。

S7 路由的要求

- 在 STEP 7 的项目中已对网络中可访问的所有设备进行了组态和下载。
- S7 路由中涉及的所有设备必须接收有关可通过特定 S7 路由器访问的 S7 子网的信息。由于 CPU 扮演着 S7 路由器的角色，这些设备通过将硬件配置下载到 CPU 来获取路由信息。

  在具有多个连续 S7 子网的拓扑中，必须按照以下顺序进行下载：首先，将硬件配置下载到同一 S7 子网中作为 PG/PC 的 CPU；然后，按照 S7 子网自近到远的顺序，逐一下载到 S7 子网的 CPU。
- 必须将用于通过 S7 路由器建立连接的 PG/PC 分配给与其物理连接的 S7 子网。可以根据菜单命令“在线诊断 > 在线访问 > 连接到接口/子网(Online & Diagnostics > Online accesses > Connection to interface/subnet)，将该 PG/PC 指定为 STEP 7 中的 PG/PC。
- 对于类型为 PROFIBUS 的 S7 子网：CPU 必须组态为 DP 主站。如果要组态为 DP 从站，则必须选中 DP 从站上 DP 接口属性内的“测试、调试、路由”(Test, commissioning, routing) 复选框。
- 从 STEP 7 V13 SP1 起，支持 HMI 连接的 S7 路由。

说明

防火墙和 S7 路由

如果发送方位于与防火墙相邻的 S7 子网之外，则防火墙在 S7 路由过程中无法识别该发送方的 IP 地址。

有关支持“S7 路由”功能的设备概览，请参见本常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/584459)。

用于在线连接的 S7 路由

通过 PG/PC，可访问 S7 子网以外的设备。如，可执行以下操作：

- 下载用户程序
- 下载硬件配置
- 执行测试和诊断功能

在下图中，CPU 1 为 S7 子网 1 和 S7 子网 2 间的 S7 路由器。

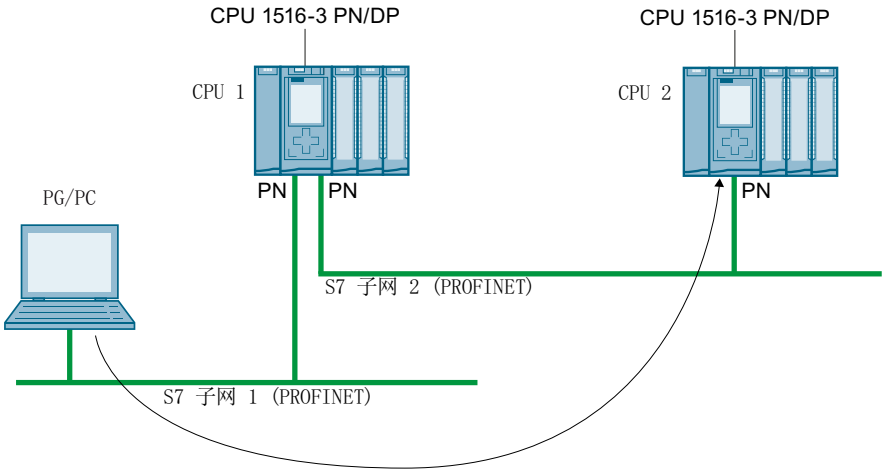


图 13-1 S7 路由：PROFINET - PROFINET

下图举例说明了从 PG 通过 PROFINET 访问 PROFIBUS 的过程。CPU 1 是 S7 子网 1 和 S7 子网 2 间的 S7 路由器；CPU 2 是 S7 子网 2 和 S7 子网 3 间的 S7 路由器。

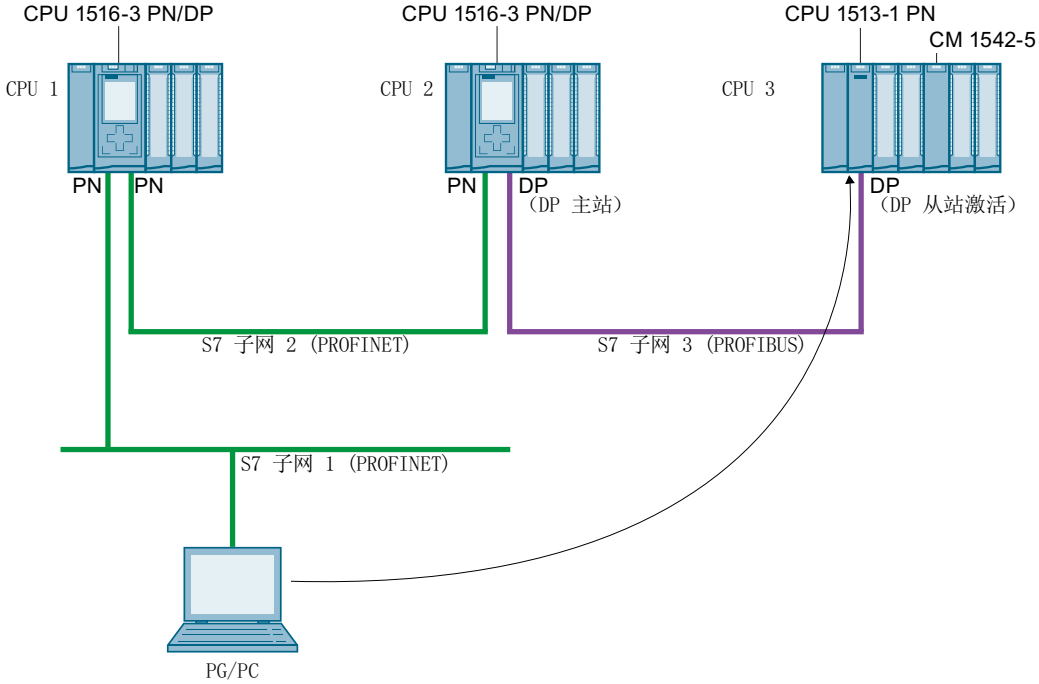


图 13-2 S7 路由：PROFINET - PROFIBUS

用于 HMI 连接的 S7 路由

可通过不同的子网（PROFIBUS、PROFINET 或工业以太网），在 HMI 与 CPU 间建立 S7 连接。在下图中，CPU 1 为 S7 子网 1 和 S7 子网 2 间的 S7 路由器。

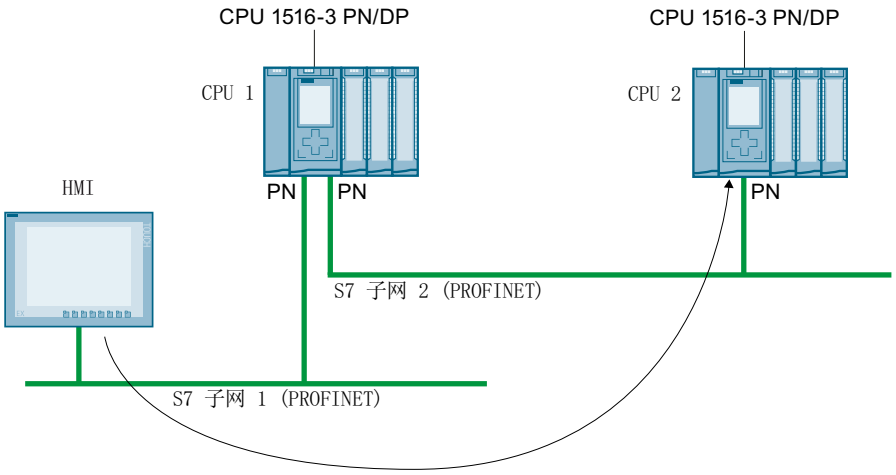


图 13-3 通过 HMI 连接实现的 S7 路由

用于 CPU-CPU 通信的 S7 路由

可通过不同的子网（PROFIBUS、PROFINET 或工业以太网），在 CPU 间建立 S7 连接。有关连接建立的具体操作步骤，请参见“S7 通信 (页 142)”部分中的示例。

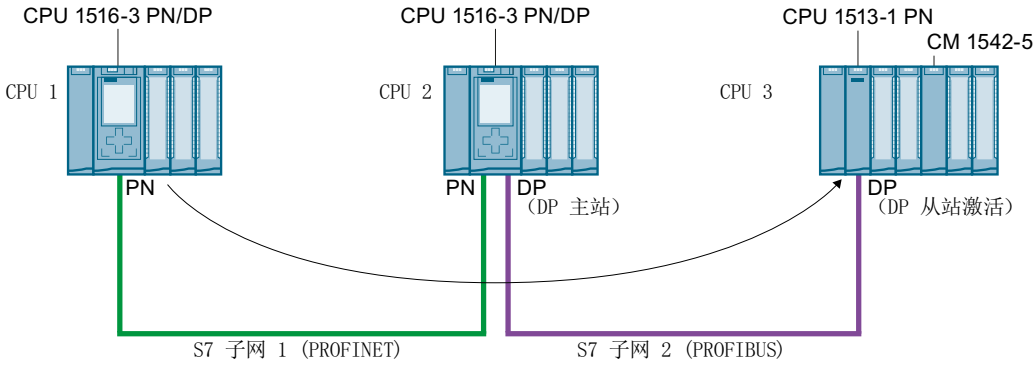


图 13-4 通过 CPU-CPU 通信实现的 S7 路由

使用 S7 路由

对于 CPU，在 STEP 7 的“转至在线”(Go online) 对话框中，选择 PG/PC 接口和 S7 子网。S7 路由将自动执行。

S7 路由的连接数量

有关 S7 路由器（CPU、CM 或 CP）上 S7 路由的连接数量，请参见相关 CPU/CM/CP 手册中的技术规范。

S7 路由：应用示例

下图举例说明了如何使用 PG 对系统进行远程维护。这里，两个 S7 子网之间通过调制解调器进行连接。

可以在 STEP 7 的“在线访问”(Online access) 或“转至在线”(Go online) 中组态一个通过 TeleService 的远程连接。

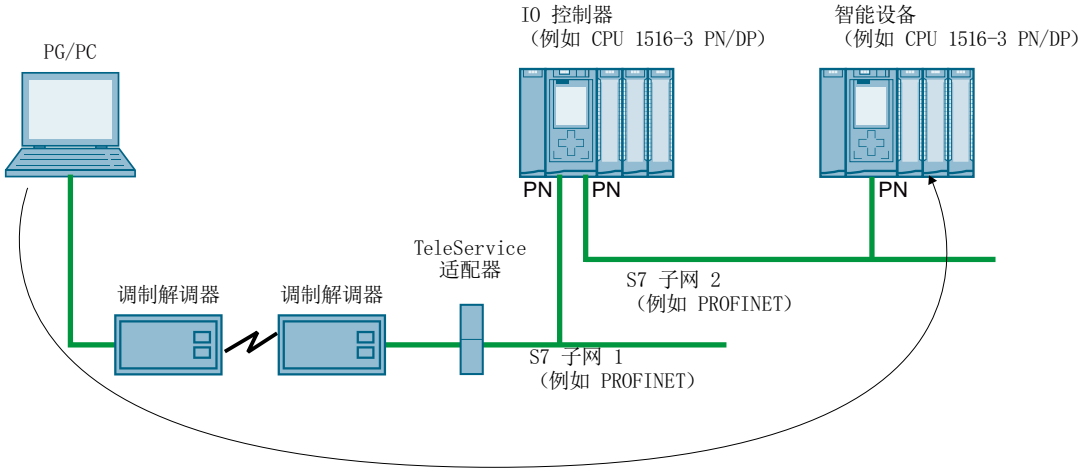


图 13-5 使用 TeleService 对设备进行远程维护

更多信息

- 有关 S7 路由的连接资源分配，请参见“连接资源的分配 (页 395)”部分。
- 有关设置 TeleService 连接的信息，请参见 STEP 7 的在线帮助。
- 有关 HMI 通信的信息，请参见“HMI 通信 (页 117)”部分。
- 有关 S7 路由和 TeleService 适配器的更多信息，可通过以下链接访问 Internet。
  - 设备手册《工业软件工程组态工具 TS Adapter 的 IE 基础知识 (https://support.industry.siemens.com/cs/cn/zh/view/51311100)》
  - TS Adapter (https://support.industry.siemens.com/cs/cn/zh/ps/16006/dl) 的下载内容

13.3 IP 转发

通过 IP 转发功能转发 IP 数据包

IP 转发是一种在两个已连接 IP 子网之间转发 IP 数据包的设备功能。

启用/禁用 STEP 7 中的 IP 转发功能。如果启用 IP 转发，则 S7-1500 CPU 会将已接收但未发送到 CPU 的 IP 数据包转发到本地连接的 IP 子网或已组态的路由器。

下图显示了编程设备访问 HMI 设备中数据的方式：编程设备和 HMI 设备位于不同的 IP 子网中。IP 子网与 CPU 的两个接口 X1 和 X2 相连。

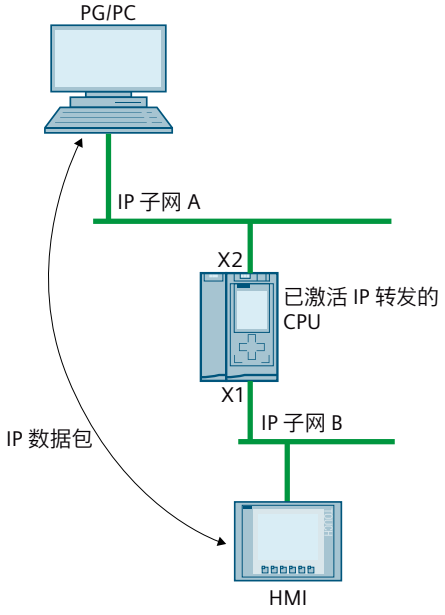


图 13-6 编程设备通过 IP 转发访问 HMI

应用范围

- 从控制级轻松访问现场级，以便对现场设备进行组态和参数分配，例如通过 PDM 或 Web 浏览器
- 简化远程访问设备的集成，例如，在远程维护或固件更新期间进行诊断

使用 IP 转发的要求

- S7-1500 CPU 固件版本 V2.8 及更高版本
- 以太网接口的数量：
  - CPU 至少具有两个以太网接口。
  - 或者 CPU 具有一个以太网接口，而由固件版本 V2.2 及更高版本的 CP 1543-1 提供另一个以太网接口。在这种情况下，必须在 CPU 中为 CP 启用"通过通信模块访问 PLC"(Access to PLC via communication module) 功能。
- IP 转发已启用。
- 在每个参与设备中沿 IP 数据包的传出和返回路径组态适当的标准网关/路由。

IP 路由表

如果启用 IP 转发，则 CPU 会对已接收但未发送到其自身的 IP 数据包进行转发。CPU 转发 IP 数据包的方式在其内部 IP 路由表中定义。

CPU 会通过已下载硬件配置的以下信息自动创建 IP 路由表。

- 以太网接口的 IP 组态
- 已组态的路由器

带有 IP 转发的组态示例

下图显示了带有所需 IP 地址设置和路由器设置的组态示例。

- IP 子网 192.168.4.0 上的 PC 与 IP 子网 192.168.2.0 上的 HMI 设备进行通信。
- 在 CPU 的以太网接口 X3 上组态路由器的 IP 地址（"标准网关"）；在下图中，它是指定为"IP 路由器"的设备。
  在 STEP 7 中，在"以太网地址 > IP 协议"(Ethernet Addresses > IP Protocol) 下的接口属性中组态路由器。

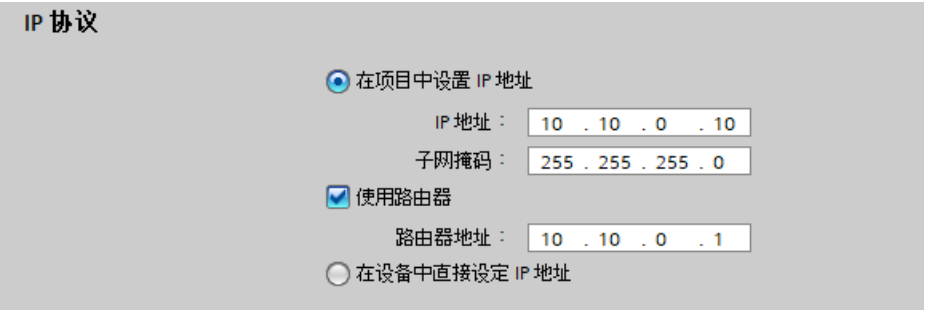


图 13-7 组态路由器

- 对于 PC、IP 路由器、IO 设备和 HMI 设备，还需输入标准网关的 IP 地址或相应的路由。

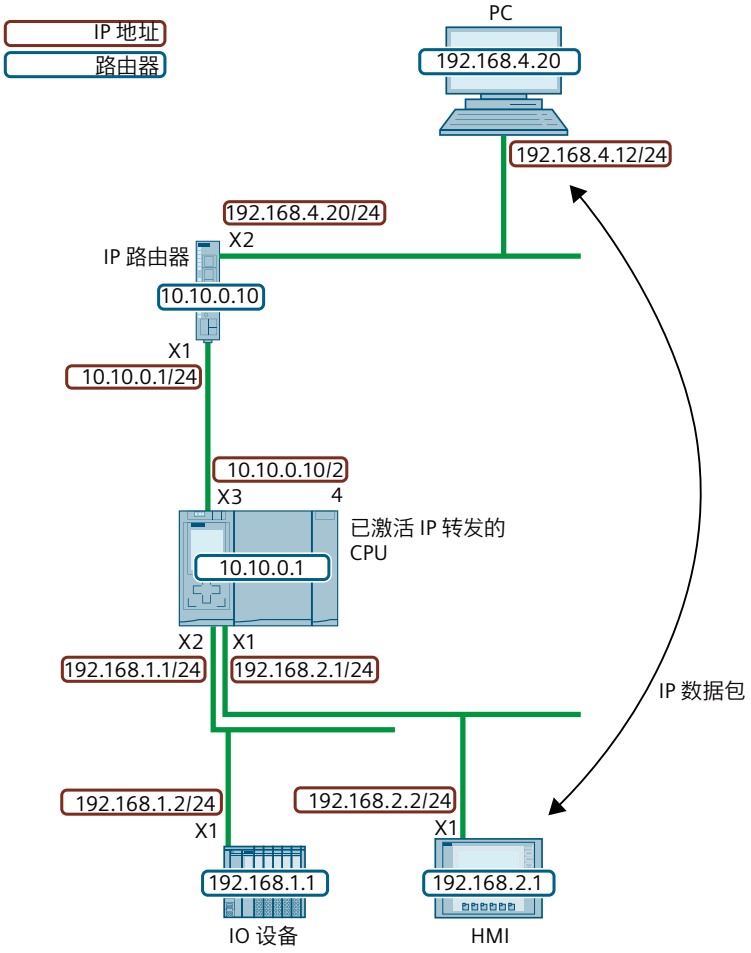


图 13-8 示例组态

此示例组态为 CPU 生成以下 IP 路由表。

表格 13-1 CPU 的 IP 路由表

网络目标	接口	网关
0.0.0.0/0	10.10.0.10	10.10.0.1
192.168.1.0/24	192.168.1.1	-
192.168.2.0/24	192.168.2.1	-
10.10.0.0/24	10.10.0.10	-

对于 PG/PC 和 HMI 设备之间的 IP 通信，需要同时在 PC 和 IP 路由器中设置到 HMI 设备 IP 子网的附加 IP 路由。在 HMI 设备中，将 CPU 接口 X1 的 IP 地址组态为标准网关。

例如，在 Windows 计算机中，通过命令提示符使用命令"route add <目标 IP 子网> mask <子网掩码> <网关>"设置附加 IP 路由。但是，需要特定权限才能完成此操作。针对本示例，输入如下提示：

- "route add 192.168.2.0 mask 255.255.255.0 192.168.4.20"

在 IP 路由器中，可以设置附加路由，例如，通过 Web 接口。针对本示例，设置如下路由：

- 目标 IP 子网：192.168.2.0
- 子网掩码：255.255.255.0
- 网关：10.10.0.10

限制

对于 S7-1500 CPU，您无法为其组态路由器（“标准网关”）以外的任何其它 IP 路由。网络目标是连接的 IP 子网，或者只能通过一个可组态的路由器访问网络目标。由于 S7-1500 CPU 不支持附加 IP 路由，因此，无法构建双向 IP 路由器级联。

在以下组态中，您可以在
CPU 中组态“路由器 1”或“路由器 2”。以组态“路由器 1”为例。在这种情况下，无法组态“路由器 2”。PC 和 HMI 设备之间的 IP 通信无法实现，因为两个方向上的路由不连续。

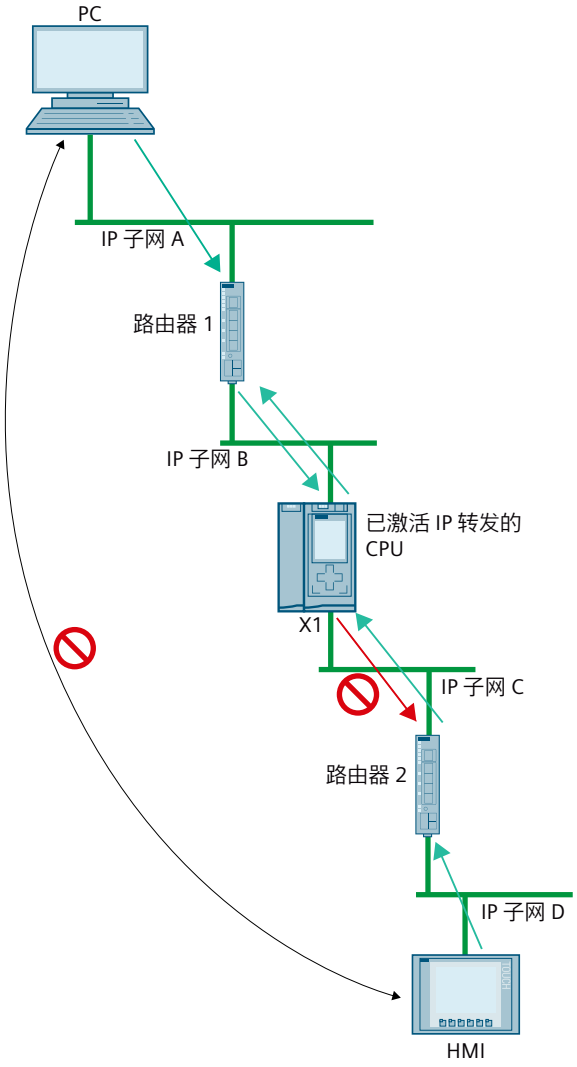


图 13-9 不支持的 IP 路由器级联

通过 CP 接口进行的 IP 转发

通过 CP 接口也可以进行 IP 转发。在这种情况下，必须在 CPU 中为 CP 激活"通过通讯模块访问 PLC"(Access to PLC via communication module) 功能。

STEP 7 的在线帮助中介绍了如何启用"通过通讯模块访问 PLC"(Access to PLC via communication module) 功能。

ET 200SP CPU 的 CP (IE) 不支持通过 CP 的接口进行 IP 转发。

通过 X1 或 X2 接口访问 CPU 1518 4 PN/DP MFP 的 C/C++ Runtime

如果为 CPU 1518 4 PN/DP 激活 PN/DP MFP IP 转发，则不仅可以通过 X1 和 X2 接口访问 X3 接口 IP 子网中的设备，还可以访问 C/C++ Runtime。通过 CPU 1518 4 PN/DP MFP 的 C/C++ Runtime，可以访问接口 X1、X2 和 X3 的 IP 子网中的所有设备。

条件：

- 已针对 CPU 1518 4 PN/DP MFP 启用了 IP 转发。
- C/C++ Runtime 的 IP 地址和 X3 接口的 IP 地址位于同一 IP 子网中。
- 在 C/C++ Runtime 中，输入到 X1 和 X2 接口 IP 子网的路由。
  在 C/C++ Runtime 中使用以下命令输入路由："Route add-net <目标 IP 子网> mask <子网掩码> gw <网关>

下图显示了 PC 通过接口 X2 访问 CPU 1518-4 PN/DP MFP 的 C/C++ Runtime 的组态。

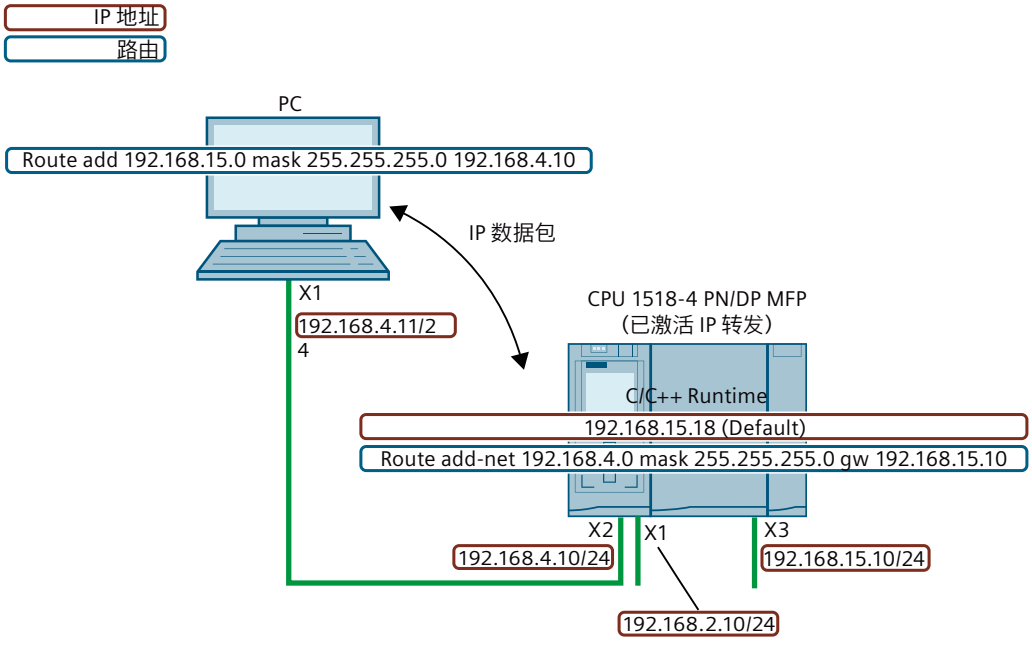


图 13-10 通过接口 X2 访问 C/C++ Runtime

进行 IP 转发时考虑网络安全

如果激活 CPU 的 IP 转发，则可以对实际只能由 CPU 访问和控制的设备启用"外部"访问。因此，这些设备通常无法防止攻击。

下图显示了如何保护自动化系统以防止未授权的访问。

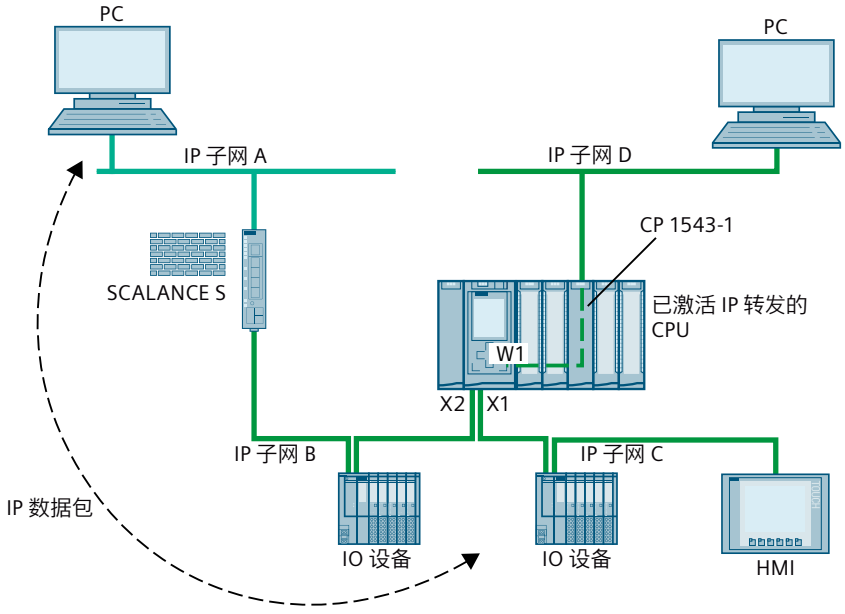


图 13-11 IP 转发的网络安全

- CPU 通过接口 X1 和 X2 直接靠近 CPU 的深绿色 IP 子网 B 和 C 内的所有设备。
- 已在 CPU 中组态 SCALANCE S 路由器。CPU 通过路由器访问远程浅绿色 IP 子网 A 中的设备。
- 已在 CPU 中为 CP 1543 启用"通过通信模块访问 PLC"(Access to PLC via communication module) 功能。CPU 通过 W1 接口访问 IP 子网 D 内的所有设备。

如果在 CPU 中激活 IP 转发，则 IP 子网 A 中的设备可以访问邻近 CPU 的 IP 子网 B、C 和 D 中的任何设备。

保护自动化系统和连接的设备以防止来自外部的未授权访问。

使用防火墙分隔 CPU 相关的 IP 子网和远程 IP 子网。例如，使用集成了防火墙的 SCALANCE S 安全模块。
此应用示例 (https://support.industry.siemens.com/cs/cn/zh/view/22376747)将介绍如何使用 SCALANCE S602 V3 和 SCALANCE S623 安全模块保护带防火墙的自动化单元。

启用/禁用 IP 转发

要启用 IP 转发，请执行以下操作：

1. 在 STEP 7 (TIA Portal) 的网络视图中选择 CPU。
2. 在巡视窗口的 CPU 属性中，浏览至“常规 > 高级组态 > IP 转发”(General > Advanced Configuration > IP forwarding)。
3. 在“组态 IPv4 转发”(Configuration IPv4 Forwarding) 区域中，选中“为此 PLC 的接口激活 IPv4”(Activate IPv4 for interfaces of this PLC) 复选框。

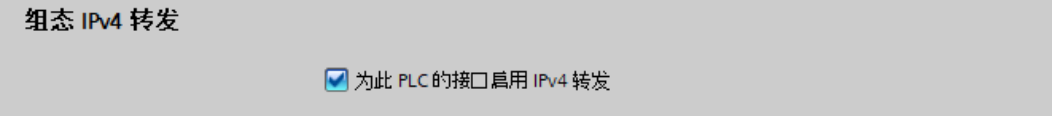


图 13-12 启用 IP 转发

结果：已针对 S7-1500 CPU 的所有接口启用 IP 转发。

通过清除“为此 PLC 的接口启用 IPv4 转发”(Enable IPv4 forwarding for interfaces of this PLC) 复选框来禁用 IP 转发。

13.4 数据记录路由

数据记录路由的定义

可以通过 PROFINET，从工程师站将数据发送到多个网络中的现场设备。由于工程师站使用标准化的记录对现场设备寻址且这些记录通过 S7 设备来路由，因此使用“数据记录路由”这个术语来表示这种路由。

通过数据记录路由发送的数据包括使用的现场设备（从站）的参数分配，以及设备的特定信息（如，设定值、限值等）。

例如，在使用不同厂商的现场设备时，可使用数据记录路由。为进行组态和诊断，将使用标准数据记录 (PROFINET) 来寻址现场设备。

使用 STEP 7 实现数据记录路由

通过借助于 TCI 接口（工具调用接口）调用设备工具（如 PCT) 并传递调用参数，可使用 STEP 7 执行数据路由。设备工具使用的通信路径也可供 STEP 7 用来与现场设备通信。

除了在 STEP 7 PC 上安装 TCI 工具外，这种路由不需要其它组态。

示例：通过 Port Configuration Tool (PCT) 实现的数据记录路由

可以使用端口组态工具 (PCT) 来组态 ET200 的 IO Link 主站，并向连接的 IO Link 设备分配参数。子网通过数据记录路由器来连接。例如，数据记录路由器可以是 CPU、CP、IM 或 IO Link 主站。

有关 PCT 支持的数据记录路由器的配置，请参见本常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/87611392)。

下图显示了通过 PCT 实现数据记录路由的组态示例。

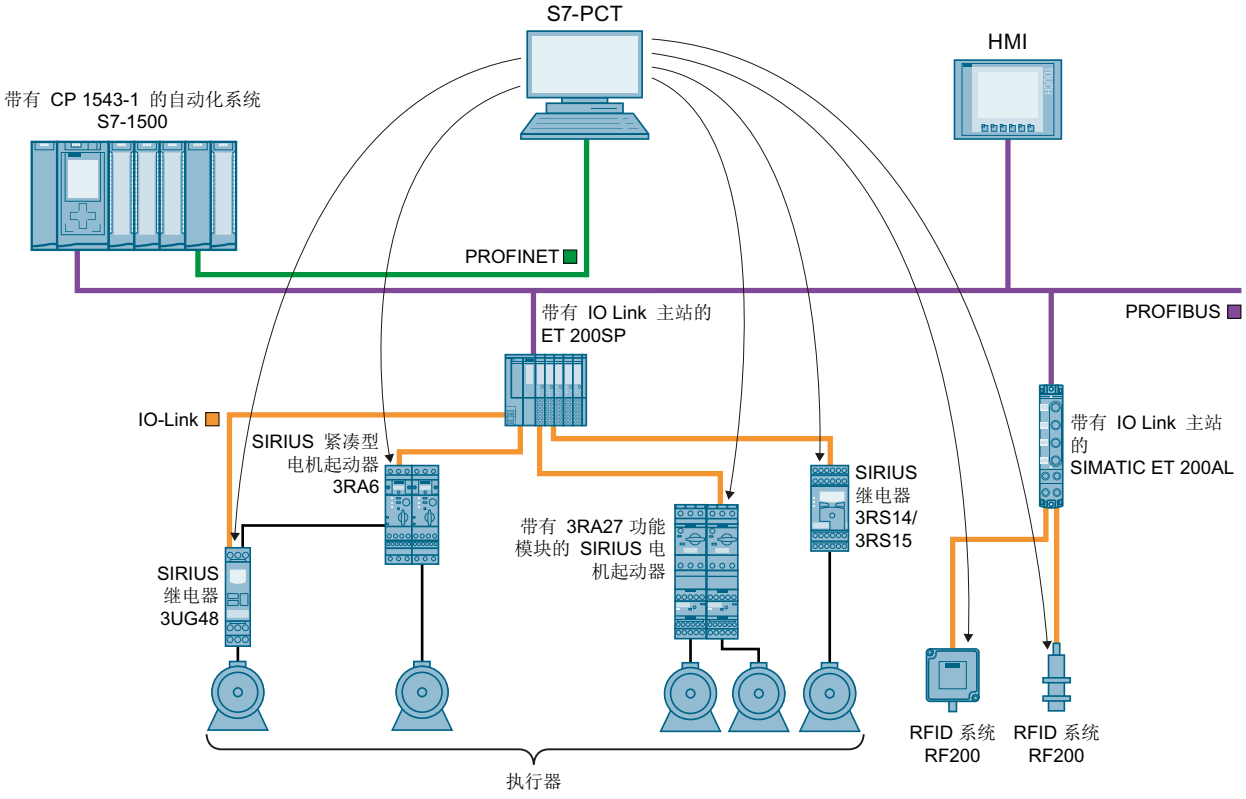


图 13-13 通过 PCT 实现数据记录路由的组态示例

更多信息

- 在本常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/7000978)中，介绍了“正常”路由与数据记录路由之间的差异。
- 有关 CPU、CP 或 CM 是否支持数据记录路由，请参见相关手册。
- 有关数据记录路由的连接资源分配的信息，请参见“连接资源分配 (页 395)”部分。
- 有关使用 STEP 7 进行组态的更多信息，请参见 STEP 7 在线帮助。

13.5 基于 IP 的应用程序的虚拟接口

在 S7-1500 CPU 固件版本 V2.8 及以上版本中，可选择通过其本地 (PN) 接口或通过同一站中通信处理器的接口访问其基于 IP 的应用程序（如 OPC UA）。有关支持的通信处理器，请参见要求。通信伙伴可通过虚拟接口访问基于 IP 的应用程序，该接口可在 TIA Portal V16 及以上版本中组态。虚拟接口称为 W1（根据 IEC 81346-2）。

虚拟接口的特性

虚拟接口不是具有传统接口常用属性的完全可诊断接口。由于通过背板总线实现的内部连接不代表 S7 子网，并且没有任何端口，因此虚拟接口不会在诊断视图中显示。因此，无法建立通过网络电缆实现的物理连接。

将显示虚拟接口的 IP 地址（在 TIA Portal 中的 CPU 显示中），并且可以对该地址进行组态。

可以通过虚拟接口 W1 使用以下通信选项：

- OPC UA（客户端和服务器）
- 编程的 OUC 连接
- 编程设备/HMI 通信
- 通过 PUT/GET 指令从 S7 CPU 进行伙伴访问

激活的接口可在组态了基于 IP 的连接的对话框中使用。

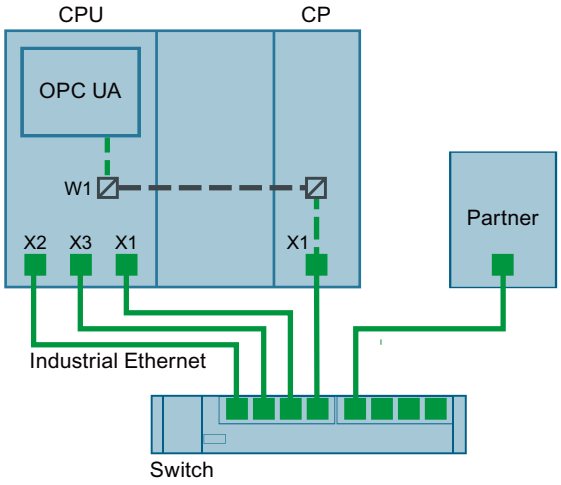


图 13-14 虚拟接口的原理

与传统接口相比，虚拟接口有以下限制：

- 无法通过虚拟接口访问 Web 服务器。
- 无法通过带有 TIA Portal 的已连接编程设备进行在线备份。
- 如果 CPU 和通信伙伴通过虚拟接口连接，则无法通过 LLDP（链路层发现协议）交换数据。
- S7 路由服务不使用虚拟接口 W1。

要求

要通过 CP 的以太网接口访问 CPU 服务，必须满足以下要求：

- S7-1500 CPU 固件版本 V2.8 及更高版本
- CP 1543-1 固件版本 V2.2 及更高版本

建议：使用固件版本不低于 V3.0 的
CP 1543-1。自该版本起，还为虚拟接口提供安全功能（防火墙），且不需要在站与非安全网络之间安装额外的防火墙。

虚拟接口 W1 的组态

在固件版本不低于 V2.8 的 S7-1500 CPU 的属性中，在“高级组态 > 通过通信模块访问 PLC”(Advanced configuration > Access to PLC via communication module) 下，可将插入的通信模块分配给 W1 虚拟接口。之后，即可使用该接口对 CPU 进行外部访问。如果未插入 CP，或者插入的 CP 不支持对 CPU 进行访问，则选项保持为空。

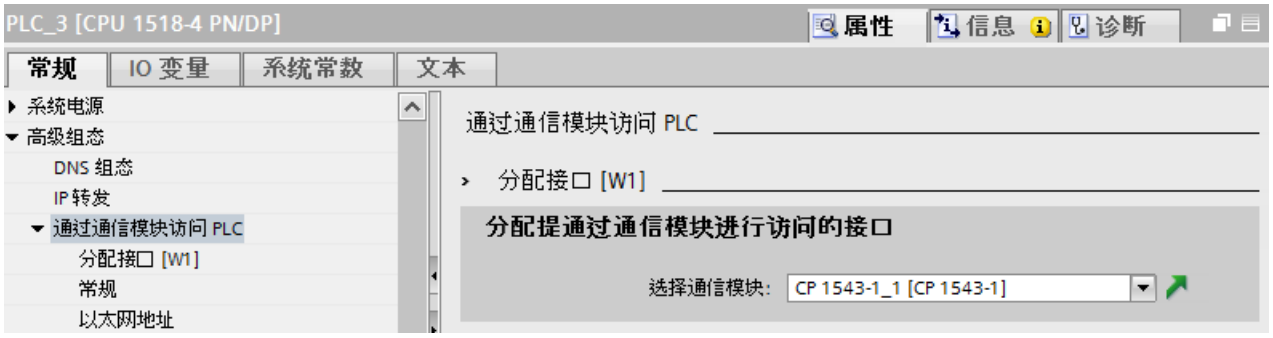


图 13-15 在 CPU 属性中选择 CP

选择 CP 后，将显示虚拟接口的规格和参数。可在此处编辑 IP 协议和 PROFINET 参数的设置。

- 与 CP 一样，IP 子网可以自由选择。通过虚拟接口的子网掩码和 IP 地址可以输入 IP 子网。
- 输入虚拟接口的 IP 子网时，请注意，不要使用与 CPU 本地接口所用子网相同的 IP 子网。

输入 IP 地址后，该地址将显示在服务器地址列表中 OPC UA 服务器的属性对话框中。这些设置为 CPU 配备新的 W1 虚拟接口，通过该接口，可通过通信模块访问上述 CPU 服务，如 OPC UA 服务器。通过该接口可进行相应的连接和 S7 通信（例如 HMI 和 BSEND、BRCV）。OPC UA 服务器不允许选择特定接口（通过 IP 地址选择），只能全部选择或全部不选。

说明

虚拟接口的 IP 地址不会作为 W1 列于设备显示中当前显示的本地接口 (Xn) 下，但会在“设置”(Settings) 部分中的“地址”(Addresses) 下可用。未插入 CP 或者未激活虚拟接口时，也会显示虚拟接口。如果没有可用的 IP 套件，则 IP 地址和子网掩码为 0.0.0.0。

如果通过显示画面、T_CONFIG 指令或在线的方式更改虚拟接口的已组态和加载的 IP 地址参数，则在 CPU 重新启动后，已加载的组态将再次激活。

CP 上的组态更改

更改分配的通信模块可能会影响虚拟接口的组态：

- 在 CPU 的属性中：
  - 分配不同 CP：该组态用于新 CP。
  - 取消选择已分配的 CP：取消激活虚拟接口 W1 且组态丢失。
    如果再次分配 CP，则必须再次执行组态。
- 在设备上：
  - 移动 CP：如果 CP 只是移动到设备的其它插槽，则组态仍有效。
  - 拆卸 CP：如果 CP 已删除或移动到其它设备，则会保留组态。在 CPU 的下拉列表中，CP 显示为缺失，编译组态时会出现错误。可以取消选择缺失的 CP 或将其分配给另一个 CP。

在诊断和系统常量中显示

虚拟接口 W1 显示在诊断视图中的"在线和诊断"(Online & Diagnostics) 下。虚拟接口的硬件 ID 显示在 CPU 属性的系统常量中。

通信模块中的设置（CP 1543-1 固件版本 V3.0 及更高版本）

自固件版本 V3.0 起，可使用 CP 的内部防火墙确保通过虚拟接口传输的数据流量的安全。要在通信模块中激活防火墙，可在受保护的项目中进行如下操作：

1. 在 STEP 7 工作区中，选择该通信模块。
2. 在巡视窗口中，转至"属性 > 安全"(Properties> Security)。
3. 激活"启用安全功能"(Enable security functions) 选项。
   可组态的安全功能现在会出现在巡视窗口中。
4. 激活"启用防护墙"(Enable firewall) 选项。
   在巡视窗口中，现在可允许使用虚拟接口 W1。

对于 OPC UA 服务，仅可确保通过 CPU 的 W1 虚拟接口传输的数据流量的安全。

说明

检查手动组态

如果启用防火墙，则需要手动检查防火墙是否允许所需的服务。仅为待通过该 CP 接口进行访问的 IP 和 MAC 过滤器启用这些服务。请参见 TIA Portal 信息系统中关于 S7-1500 CP 安全设置和防火墙规则的说明。

通信模块中的设置（CP 1543-1，固件版本 V2.2，低于 V3.0）

固件版本低于 V3.0 的 CP 1543-1 的安全功能无法确保通过虚拟接口的数据传输安全。虽然可在 TIA Portal 中激活安全功能，但不能编译此类组态。

注意
连接到非安全网络 如果将 CP 连接到非安全网络，则务必在该 CP 与非安全网络间连接一个附加防火墙。例如，可通过集成防火墙连接安全模块 SCALANCE S602 V3 和 SCALANCE S623。

连接资源

14

14.1 站中的连接资源

简介

某些通信服务需要进行连接。将占用自动化系统（站）中的连接资源。CPU、通信处理器 (CP) 和通信模块 (CM) 可为站提供所需的连接资源。

站中的连接资源

可用连接资源取决于所使用的 CPU、CP 和 CM，且不得超过单个站的最大可用连接资源数量。

站中最大的可用资源数量取决于 CPU。

预留的连接资源

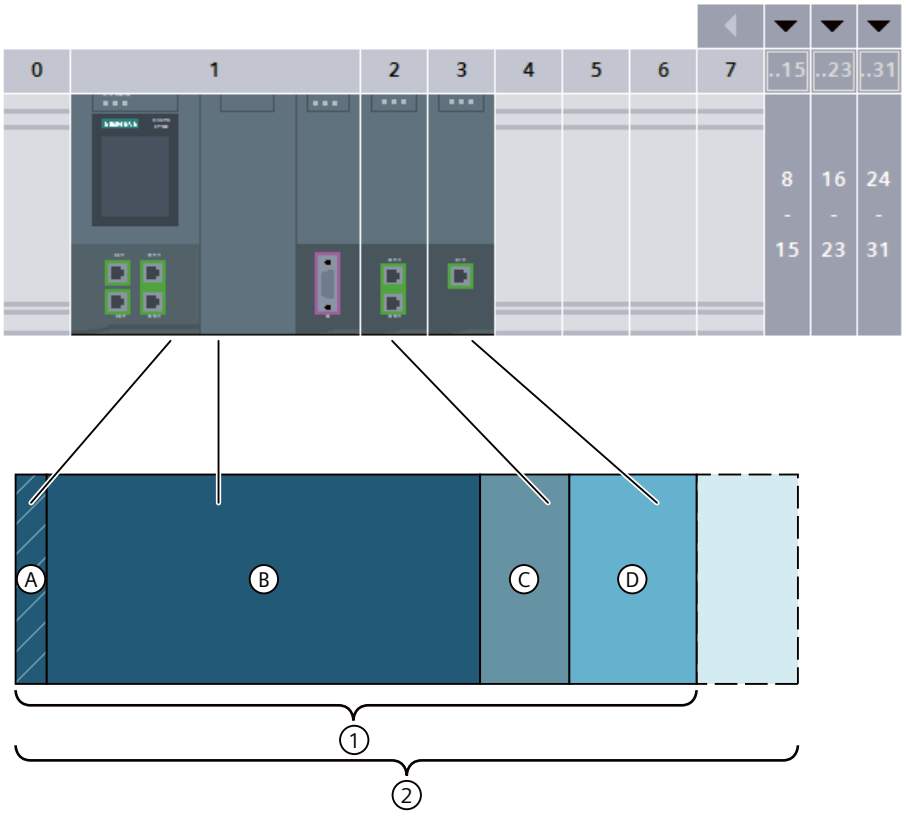
每个 CPU 都会为 PG、HMI 和 Web 服务器通信预留一定数量的连接资源。这样可确保无论多少个通信服务使用连接资源时，PG 与 CPU 间始终保留至少一条在线连接。

动态连接资源

此外，存在动态资源。最大连接资源数量减去预留的连接资源数量，即为动态连接资源的最大数量。

将使用 PG 通信、HMI 通信、S7 通信、开放式用户通信、Web 通信、OPC UA 客户端/服务器通信和动态连接资源池中的其它通信等通信服务。

下图举例说明了各个组件如何使连接资源用于 S7-1500 站。



① 站的可用连接资源的分配

A 站中预留的连接资源

A + B CPU 1518 的连接资源

C 通信模块 CM 1542-1 的连接资源

D 通信处理器 CP 1543-1 的连接资源

② 在采用 CPU 1518、CM 1542-1 和 CP 1543-1 的组态示例中，该站通信资源的最大数量

图 14-1 站中的连接资源

站中连接资源的数量

表格 14-1 某些 CPU 型号所支持的连接资源最大数量

站中的连接资源		1511C	1511 1512C 1513	1515 1516	1517	1518
站中连接资源的最大数		96	128	256	320	384
	预留	10				
	动态	86	118	246	310	374
CPU 中的连接资源		64	88	128	288	320
插入 CM/CP 时，可额外使用的连接资源最大数		32	40	128	32	64

站中的连接资源	1511C	1511 1512C 1513	1515 1516	1517	1518
CM 1542-1 可额外使用的连接资源数量	64				
CP 1543-1 可额外使用的连接资源	118				
CM 1542-5 可额外使用的连接资源数量	48				
CP 1542-5 可额外使用的连接资源	16				

CPU 或通信模块所支持的连接资源数将在设备手册的技术规范部分进行介绍。

示例

在 CPU 1516-3 PN/DP 组态中，包含一个 CM 1542-1 通信模块和一个 CP 1542-5 通信处理器。

- 站中连接资源的最大数：**256**
- 可用的连接资源：
  - CPU 1516-3 PN/DP：128
  - CM 1542-1：64
  - CP 1542-5:16
  - 总计：**208**

在该设置中，可使用 208 个连接资源。添加其它通信模块后，该站最多可额外支持 48 个连接资源。

预留的连接资源

为带有 S7-1500 CPU 的站、ET 200SP CPU 和基于 S7-1500 的 ET 200pro CPU，预留了 10 个连接资源

- 4 个连接资源，用于 STEP 7 所需的 PG 通信，如进行测试和诊断或将数据下载到 CPU 中
- 4 个连接资源，用于 STEP 7 中所组态的第一个 HMI 连接的 HMI 通信
- 2 个连接资源，用于与 Web 服务器进行通信

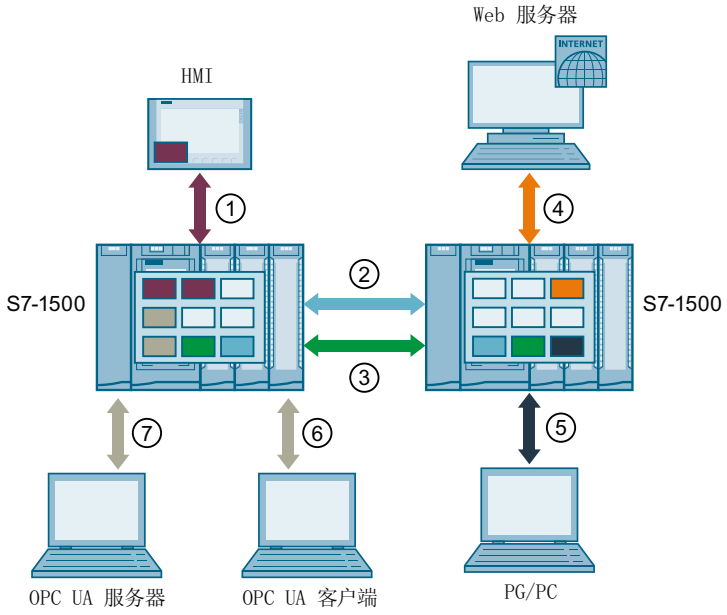
更多信息

有关 S7-1500R/H 冗余系统的连接资源的信息，请参见"冗余系统 S7-1500R/H 的连接资源 (页 419)"部分。

14.2 连接资源的分配

概览 - 连接资源的分配

下图描述了不同连接方式中 S7-1500 资源的分配。



① HMI 通信：见下文。

② 开放式用户通信：开放式用户通信连接会占用每个端点的一个连接资源。

③ S7 通信：S7 通信连接会占用每个端点的一个连接资源。

④ Web 通信：Web 服务器连接会占用站中至少一个连接资源。占用的连接数取决于具体的浏览器。

⑤ PG 通信：PG 通信连接会占用站中一个连接资源。

⑥ OPC UA 客户端/服务器通信：服务器连接资源分配，见下文

⑦ OPC UA 客户端/服务器通信：客户端连接资源分配，见下文

HMI 通信的连接资源

开放式用户通信的连接资源

S7 通信的连接资源

Web 通信的连接资源

PG 通信的连接资源

用于 OPC UA 服务器通信的连接资源

图 14-2 连接资源的分配

HMI 通信的连接资源

进行 HMI 通信时，站中所占用的连接资源数量取决于所使用的 HMI 设备。

表格 14-2 不同 HMI 设备占用的连接资源的最大数

HMI 设备	各 HMI 连接占用的站中连接资源的最大数
精简面板	1
Unified 基本面板	3
精智面板	2[1]
Unified 精智面板	3
移动面板	2[1]
RT Advanced	2[1]
RT Professional	3
Unified PC	3

[1] 如果未使用系统诊断或报警组态，每个 HMI 连接仅占用该站的一个连接资源。

示例：已组态 CPU 1516-3 PN/DP 的以下 HMI 连接：

- 至 HMI TP700 Comfort 的两个 HMI 连接。（每个连接占用 2 个连接资源）
- 至 HMI KTP1000 Basic 的一个 HMI 连接。（1 个连接资源）

CPU 中的 HMI 通信总共占用 5 个连接资源。

用于 OPC UA 客户端通信的连接资源

CPU 的 OPC UA 客户端与 OPC UA 服务器建立的每条连接都将占用站中的一个连接资源。

建立和关闭 OPC UA 连接时，OPC UA 客户端会临时占用额外的连接资源。根据 RFC 793，此连接资源在等待大约 60 秒后再次释放。

说明

因存在临时连接资源而导致资源不足

在下列情况下将缺少连接资源：

- CPU 的 OPC UA 客户端同时建立或关闭几个连接。
- 对于 OPC UA 客户端通信的永久和临时连接资源，站中可用连接资源的数量不足。

请始终确保站中的可用连接资源充足，以便建立和终止 OPC UA 连接。

措施：

- 计划针对 OPC UA 客户端连接预留足够的可用资源。
- 如有必要，依次建立或关闭 OPC UA 连接。

路由的连接资源

如果要在 S7 子网中进行数据传输（“S7 路由”），则需在两个 CPU 之间建立一条 S7 连接。S7 子网将通过网关（即，S7 路由器）进行连接。在 S7-1500 中，CPU、CM 和 CP 可作为 S7 路由器。

以下信息说明了 S7 连接中的路由数据情况：

- 在两个端点上，路由的连接各占用一个连接资源。STEP 7 在“连接资源”(Connection resources) 表中显示这些连接资源。
- 在 S7 路由器中，S7 路由将占用两个特定的连接资源。STEP 7 不会在“连接资源”(Connection resources) 表中显示 S7 路由的专用连接资源。S7 路由时所需的连接资源数量取决于 CPU。有关 S7 路由所需的连接资源数量，请参见 CPU 技术规范中的“S7 路由的连接数量”。

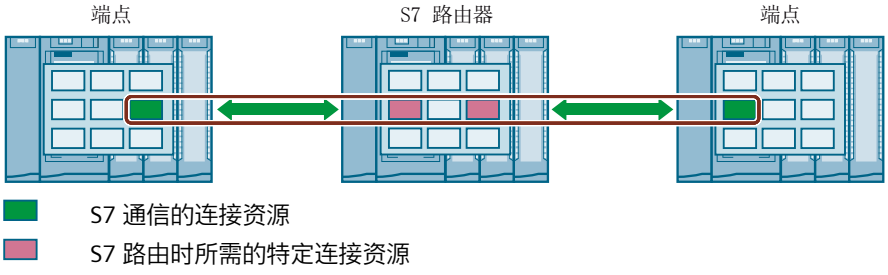


图 14-3 S7 路由的连接资源

数据记录路由还可以跨 S7 子网进行数据传输，通过 PROFIBUS 从连接到 PROFINET 的工程师站将数据传输到各种现场设备。

数据记录路由功能与 S7 路由功能类似，每一个数据记录路由器也会占用两个用于 S7 路由的特定连接资源。

说明

数据记录路由的连接资源

进行数据记录路由时，数据记录路由器会占用两个用于 S7 路由的特定连接资源。数据记录连接和分配的连接资源都不会显示在连接资源表中。

何时占用连接资源？

连接资源的占用时间取决于连接的建立方式（参见“建立连接 (页 41)”部分）。

- **通过编程设置连接**：
  在用户程序中调用连接建立指令（TSEND_C/TRCV_C 或 TCON）时，将立即占用连接资源。

  通过对 TSEND_C/TRCV_C 指令的 CONT 参数进行相应设置或调用 TDISCON 指令，可在数据传输后终止连接并释放连接资源。连接终止时，CPU/CP/CM 中的连接资源将再次可用。
- **已组态连接**（如 S7 连接）：
  如果在 STEP 7 中组态了一个连接，则在硬件配置下载到 CPU 时将立即占用连接资源。

  通过组态的连接完成数据传输后，连接不会终止。该连接资源为永久性占用。要再次释放该连接资源，则需在 STEP 7 中删除所组态的这一连接，并将修改后组态下载到 CPU 中。
- **PG 连接**：
  在 STEP 7 中，在线连接 PG 与 CPU 时，将会立即占用连接资源。

- **Web 服务器**：
  只要在浏览器中打开 CPU 的 Web 服务器，就占用 CPU 中的连接资源。
- **OPC UA 服务器**

  与该 CPU 的 OPC UA 服务器建立的每一个连接都将占用站中的一个连接资源。连接终止时，该连接资源立即释放。
- **OPC UA 客户端**

  CPU 的 OPC UA 客户端与 OPC UA 服务器建立的每条连接都将占用站中的一个连接资源。OPC UA 连接建立时，OPC UA 客户端将临时占用一个额外的连接资源。根据 RFC 793，OPC UA 连接终止后，该连接资源将等待约 60 秒钟时间才释放。

监视连接资源的最大数

离线

在组态连接时，STEP 7 将监视连接资源的占用情况。如果超出了连接资源的最大数量，则 STEP 7 将发出一条相应的警告消息。

在线

CPU 将监视自动化系统中连接资源的使用情况。如果用户程序中创建的连接数量超出了自动化系统可提供的数量，则 CPU 将确认该指令建立连接并显示相应错误。

S7-1500 和 S7-300 比较

有关 S7-1500 和 S7-300 通信资源管理方法的比较，请参见本“常见问题解答 (https://support.industry.siemens.com/cs/cn/zh/view/109747092)”。

14.3 连接资源的显示

在 STEP 7 中显示连接资源（离线视图）

在硬件配置中，可显示自动化系统的连接资源。这些连接资源将显示在 CPU 属性中的巡视窗口内。

连接资源

	① 站资源			② 模块资源	模块资源	模块资源
	预留		动态	PLC_1 [CPU 15...	CM 1542-1_1 [..	CP 1543-1_1 [...
最大资源数：		10	246	128	64	118
	最大	已组态	已组态	已组态	已组态	已组态
PG 通信：	4	-	-	-	-	-
HMI 通信：	4	4	6	6	0	4
S7 通信：	0	-	7	4	3	0
开放式用户通信：	0	-	13	8	5	0
Web 通信：	2	-	-	-	-	-
OPC UA 客户端/服务器通信：	0	-	-	-	-	-
其它通信：	-	-	0	0	0	0
使用的总资源：		4	26	18	8	4
可用资源：		6	220	110	56	114

图 14-4 示例：预留和可用的连接资源（离线视图）

① 站特定连接资源

在"站特定的连接资源"列中，显示有关站中已用和可用的连接资源信息。

在本示例中，自动化系统中最多可使用 256 个站特定的连接资源。

- 10 个预留的连接资源中，4 个已使用，其余 6 个可用。
  已使用的资源分别为：
  - 4 个连接资源，用于 HMI 通信
- 246 个动态连接资源。其中，26 个已使用，其余 220 个可用。
  已使用的资源划分为：
  - 6 个连接资源，用于 HMI 通信
  - 7 个连接资源，用于 S7 通信
  - 13 个连接资源，用于开放式用户通信

由于 CPU、CP 和 CM 中可用连接资源的最大数（= 310 个连接资源）超出了站的限值 256 个，将在动态站资源列中显示一个三角形警告标志。

说明

超出可用的连接资源数

STEP 7 通过一个警告表示超出站特定的连接资源数量。要确保 CPU、CP 和 CM 中连接资源的充分利用，可使用具有较大站特定连接资源数量的 CPU，也可减少通信连接的数量。

---

② 模块特定的连接资源

在模块特定的连接资源列中，显示有关自动化系统中的 CPU、CP 和 CM 的使用资源的信息：

在此，将按照模块而非接口分别显示。

在本示例中，CPU 支持最多 128 个连接资源；其中 18 个已使用，其余 110 个仍可用。已使用的资源将划分为：

- 6 个连接资源，用于 HMI 通信
- 4 个连接资源，用于 S7 通信
- 8 个连接资源，用于开放式用户通信

在 STEP 7 中显示连接资源（在线视图）

如果已在线连接 CPU，在“连接信息”(Connection information) 下还可查看当前正在使用的资源数量。

	站资源					模块资源	
	预留			动态		CPU 1516-3 PN/DP (R0/S1)	
最大资源数：		10	10	194	194	86	86
	最大	已组态	已用	已组态	已用	已组态	已用
PG 通信：	4	-	4	-	0	-	0
HMI 通信：	4	4	4	4	4	8	8
S7 通信：	0	-	0	72	68	34	34
开放式用户通信：	0	-	0	118	118	45	45
Web 通信：	2	-	0	-	0	-	0
OPC UA 客户端/服务器通信：	0	-	0	-	0	-	0
其它通信：	-	-	0	0	0	0	0
使用的总资源：		4	8	194	190	82	82
可用资源：		6	0	0	4	4	4

图 14-5 连接资源 - 在线

除离线视图外，“连接资源”(Connection resources) 表格的在线视图中也包含有正在使用的连接资源列。因此，在线视图中会显示自动化系统中**所有**已使用的连接资源，而不考虑采用的连接方式。

在“其它通信”(Other communication) 行中，将为与外部设备通信而分配的连接资源。该表格将自动进行更新。

说明

如果路由的 S7 连接经过某一 CPU，则该 CPU 所需的连接资源不出现在连接资源表中。

显示 HMI 的连接资源

有关 HMI 连接中可用的连接资源和具体分配，请参见巡视窗口中离线视图（HMI 设备）内的“连接资源”(Connection resources) 属性。

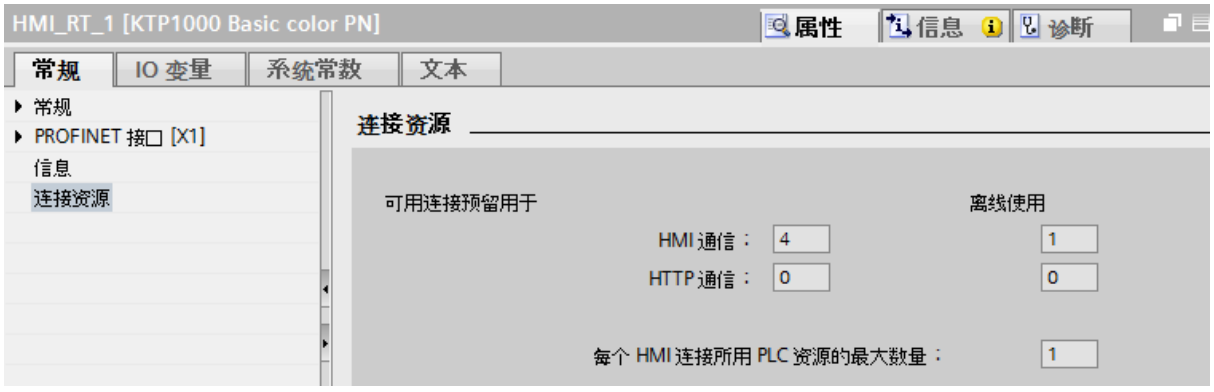


图 14-6 连接资源 - HMI 通信

在连接资源区域中，将显示以下信息：

- 为 HMI 通信和 HTTP 通信预留的 HMI 连接资源数量
- HMI 离线时，进行 HMI 离线通信和 HTTP 通信可使用的连接资源数量

  如果超出 HMI 设备可用的最大连接资源数量，则 STEP 7 将输出一条相关消息。
- “每个 HMI 连接所用的 PLC 资源最大数量”。该参数是一个系数，将乘与离线使用的 HMI 连接数量。乘积结构为 CPU 中已占用的 HMI 资源数量。

在 Web 服务器中显示连接资源

除了可在 STEP 7 中显示连接资源，也可在浏览器中显示 Web 服务器的相关资源页面。

有关在 Web 服务器中显示连接资源的信息，请参功能手册《Web 服务器 (https://support.industry.siemens.com/cs/cn/zh/view/59193560)》。

15 诊断和故障排除

15.1 连接诊断

在线视图中的连接表

在 STEP 7 的“设备与网络”(Devices & networks) 编辑器中选择了一个 CPU 后，将在连接表的在线视图中显示连接的状态。

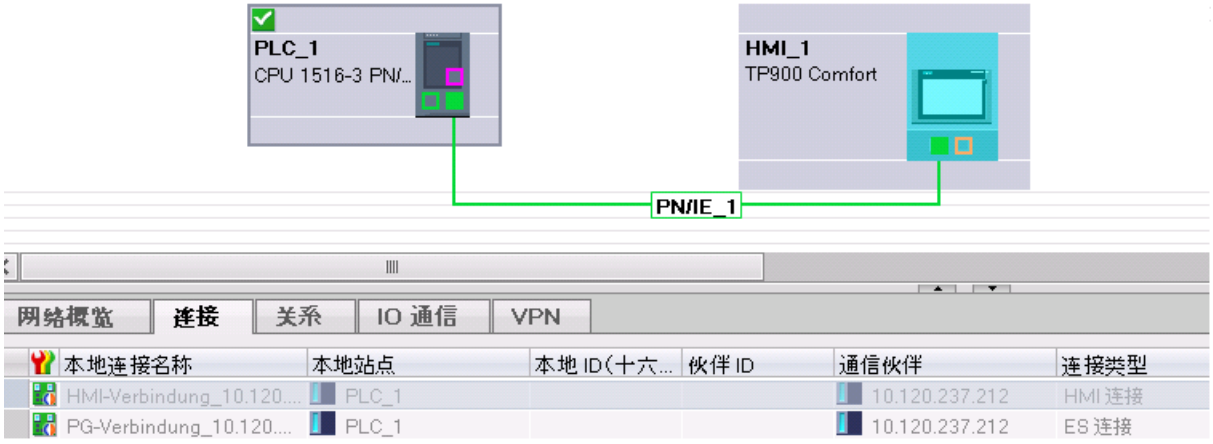


图 15-1 连接表的在线视图

在连接表中选择连接之后，可在“连接信息”(Connection information) 选项卡中查看详细诊断信息。

“连接信息”(Connection information) 选项卡：连接的详细信息

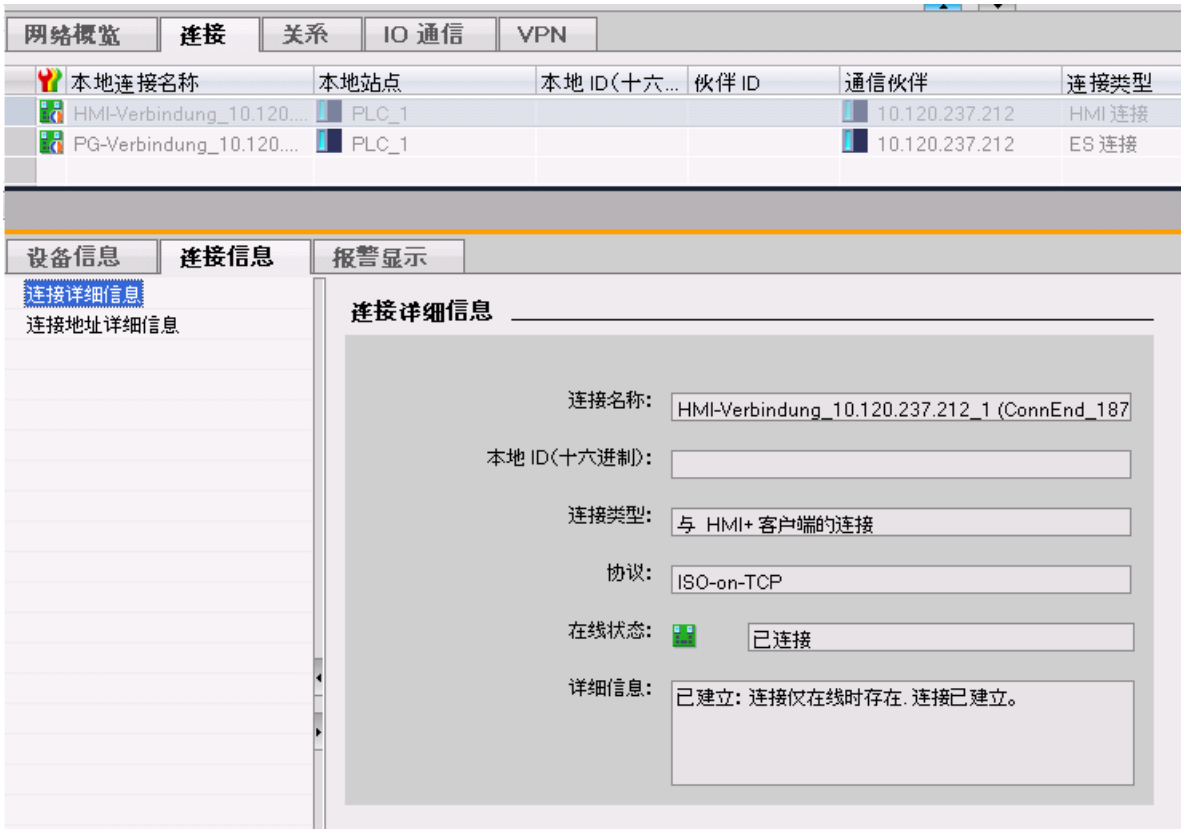


图 15-2 连接诊断 - 连接的详细信息

“连接信息”(Connection information) 选项卡：地址详细信息

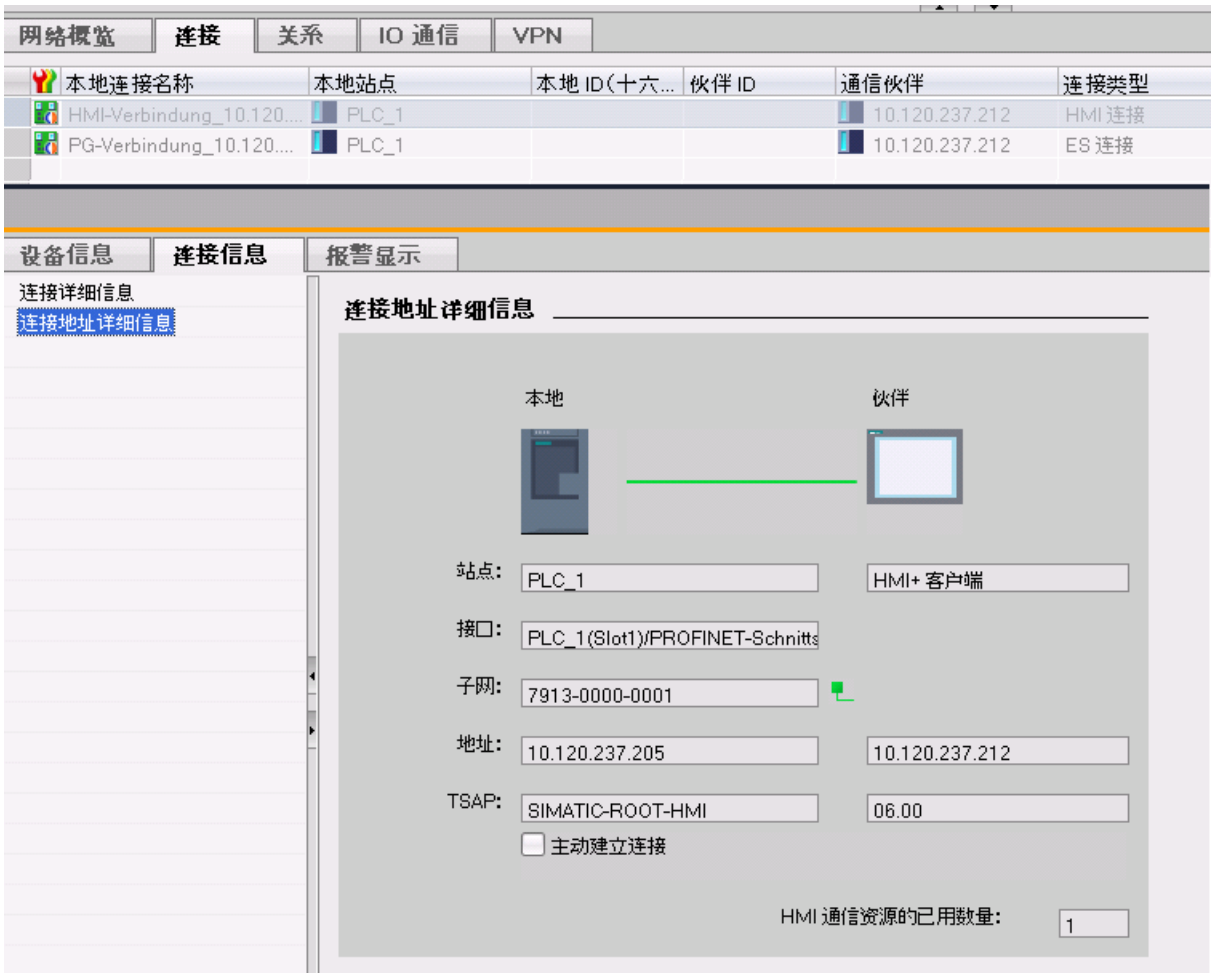


图 15-3 连接诊断 - 地址的详细信息

通过 Web 服务器进行诊断

通过 CPU 的集成 Web 服务器并使用 Web 浏览器，可以从 CPU 分析诊断信息。

在“通信”(Communication) 网页中，可在各个选项卡中查看通过 PROFINET 进行通信的以下信息：

- 有关 CPU 的 PROFINET 接口的信息（例如，地址、子网、物理属性）。
- 有关数据传输质量的信息（例如，正确发送/接收的数据包数量）。
- 有关分配的/可用的连接资源的信息。
- “连接状态”(Connections status) 页面类似于 STEP 7 中的视图，还通过详细视图提供所有连接的概览。

用户程序中的诊断

在编程 T_DIAG 指令时，可使用用户程序来评估有关 CPU 的已组态和已编程连接的诊断信息。

更多信息

有关 Web 服务器的功能介绍，请参见功能手册《Web 服务器 (https://support.industry.siemens.com/cs/cn/zh/view/59193560)》。

15.2 紧急地址

如果无法通过 IP 地址访问 CPU，可以为 CPU 设置临时紧急地址（紧急 IP）。可以通过此紧急地址重新建立与 CPU 的连接，以便加载具有有效 IP 地址的设备组态。

可以设置紧急地址，而不受 CPU 的保护等级限制。

何时需要紧急地址？

下列情况下无法访问 CPU：

- PROFINET 接口的 IP 地址已被分配两次。
- 子网掩码设置错误。

要求

- 在 STEP 7 的设备组态中，已为 IP 协议选择"在项目中设置 IP 地址"(Set IP address in the project)。
- 未加载启用了 DCP 写保护的组态。
- CPU 处于 STOP 模式。

使用紧急地址恢复有效的设备组态

1. 使用 DCP 工具设置 CPU 接口的紧急地址。例如，SIMATIC Automation Tool 有一个 DCP 命令"定义 IP 地址"(Define IP address)。
   CPU 的维护 LED 灯亮起。诊断缓冲区还显示以太网接口的紧急地址已激活。
2. 将具有有效 IP 地址的 STEP 7 项目加载到 CPU 中。
3. 关闭 CPU 并再次打开。
   将重置紧急地址。

结果

CPU 以有效的 IP 地址启动。

与冗余系统 S7-1500R/H 进行通信

16

简介

S7-1500R/H 冗余系统的基本通信功能与 S7-1500 标准系统的相同。

在本章节中，将介绍与 S7-1500R/H 冗余系统进行通信时的特殊功能与限制条件。

S7-1500R/H 冗余系统的通信方式

- 通过 TCP/IP、UDP、ISO on-TCP 和 Modbus/TCP 建立开放式用户通信
- 开放式安全用户通信，“通过电子邮件建立安全 OUC”功能除外（另请参见“开放式用户安全通信 (页 80)”）
- S7 在通信中作为服务器
- HMI 通信和 PG 通信
- 通过 OPC UA 作为服务器进行数据交换
- PG/HMI 安全通信（另请参见PG/HMI 间安全通信 (页 97)）
- SNMP
- 通过 NTP 进行时间同步
- Web 服务器（仅通过 Web API）
- 支持 CP 1543-1 通信处理器作为中央插入模块（另请参见《S7-1500R/H 系统手册 (https://support.industry.siemens.com/cs/cn/zh/view/109754833)》）

S7-1500R/H 冗余系统通信的限制条件

- 开放式用户通信：
  - 不支持所组态的连接
  - 电子邮件：S7-1500R/H CPU 支持版本低于 V5.0 的“TMAIL_C”指令。不支持自 V5.0 起的版本。
  - 不支持“TCON_Param”的连接描述
- 不支持在 S7 通信作为客户端
- PG 通信：不能同时访问两个 CPU。可访问主 CPU 或备份 CPU。

16.1 R/H CPU 的系统 IP 地址

简介

除了各 CPU 的设备 IP 地址之外，S7-1500R/H 冗余系统还支持以下系统 IP 地址：

- 两个 CPU 上 PROFINET 接口 X1 的系统 IP 地址（系统 IP 地址 X1）
- 两个 CPU 上 PROFINET 接口 X2 的系统 IP 地址（系统 IP 地址 X2）
- 两个 CPU 上 PROFINET 接口 X3 的系统 IP 地址（系统 IP 地址 X3）

通过系统 IP 地址，可与其它设备（例如，HMI 设备、CPU 和 PC）通信。这些设备通常基于系统 IP 地址与冗余系统的主 CPU 进行数据通信。这样，可确保在冗余操作中原来的主 CPU 发生故障后，通信伙伴可在 RUN-Solo 系统状态下与新的主 CPU（之前的备用 CPU）进行数据通信。

每个系统 IP 地址都有一个虚拟 MAC 地址

用户可在 STEP 7 中启用该系统 IP 地址。

与设备 IP 地址相比，系统 IP 地址的优势

- 通信伙伴专与主 CPU 进行通信。
- 即使主 CPU 发生故障，S7-1500R/H 冗余系统仍可继续通过系统 IP 地址进行通信。

应用

系统 IP 地址适用于以下应用中：

- 与 S7-1500R/H 冗余系统进行 HMI 通信可以使用 HMI 设备控制或监视冗余 S7 1500R/H 系统上的过程。
- 与 S7-1500R/H 冗余系统进行开放式用户通信：
  - 另一个 CPU 或某 PC 中应用程序访问 S7-1500R/H 冗余系统的数据。
  - S7-1500R/H 冗余系统访问一个不同的设备

  可建立 TCP、UDP 和 ISO-on-TCP 连接。
- IP 转发：如果使用系统 IP 地址作为通过 S7-1500R/H 冗余系统进行 IP 路由的网关/默认路由，则即使其中一个 CPU 出现故障，也会转发 IP 数据包。

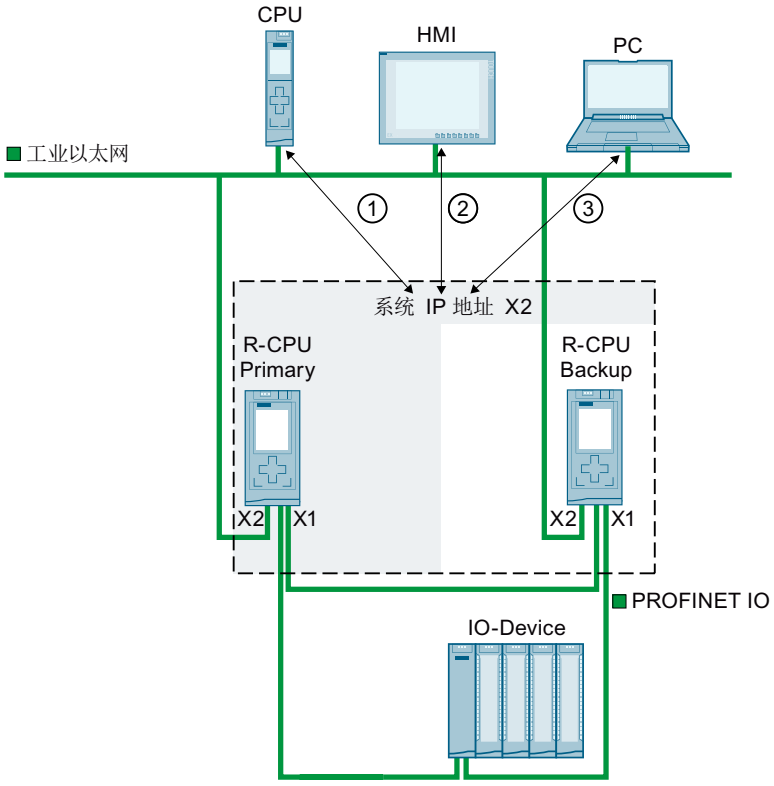
要求

- 通信伙伴接口通过同一个接口（如 X2）连接两个 CPU。
- S7-1500R/H 系统接口的系统 IP 地址已启用。

通过系统 IP 地址 X2 和 X3 进行通信

如果 S7-1500R/H 冗余系统的 CPU 上配有两个或三个 PROFINET 接口，则适合使用 PROFINET 接口 X2 或 X3 与其它设备进行数据通信。

下图显示的组态中，通信伙伴通过 S7-1500R/H 冗余系统 CPU 的相应 PROFINET 接口 X2 连接。



① 另一 CPU 与 S7-1500R 冗余系统之间的开放式用户通信

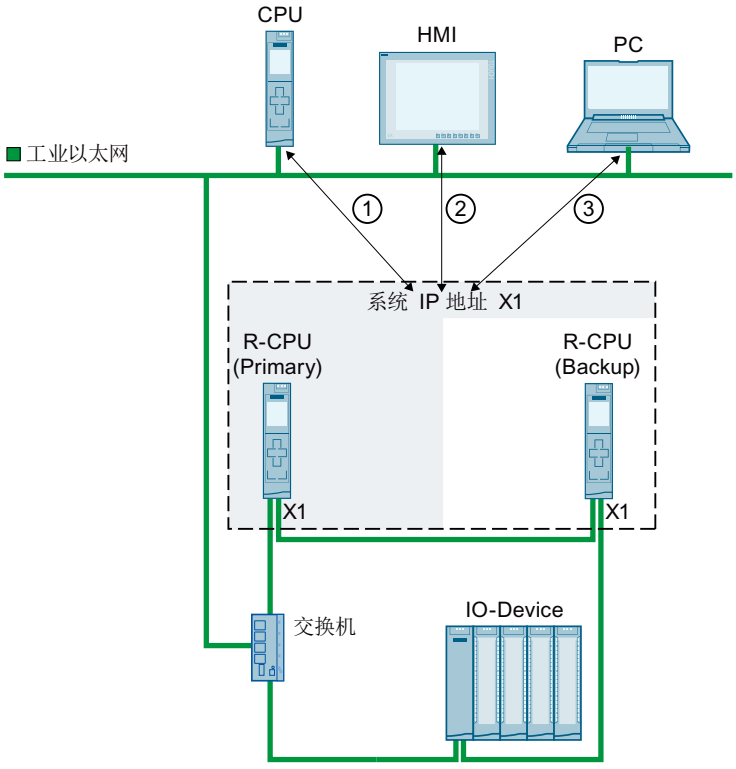
② 与 S7-1500R/H 冗余系统进行 HMI 通信

③ S7-1500R 冗余系统与某个 PC 间的开放式用户通信

图 16-1 示例：通过系统 IP 地址 X2 与 S7-1515R 冗余系统进行通信

通过系统 IP 地址 X1 进行通信

在下图显示的组态中，通信伙伴与 S7-1500R/H 冗余系统中 PROFINET 环网的交换机相连接。PROFINET 环网通过两个 CPU 上相应的 PROFINET 接口 X1 与通信伙伴相连接。
由于 CPU 1513R-1 PN 只有一个 PROFINET 接口，因此，PROFINET 环网连接只能通过系统 IP 地址 X1 进行数据通信。

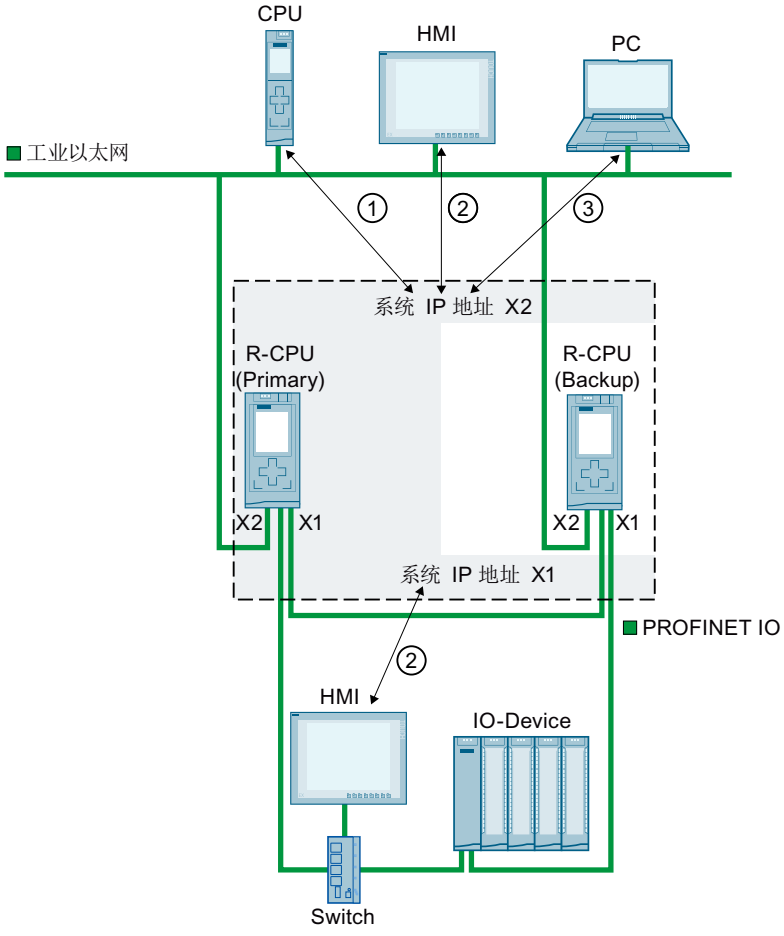


① S7-1500R 冗余系统与另一 CPU 间的开放式用户通信
② 与 S7-1500R/H 冗余系统进行 HMI 通信
③ S7-1500R 冗余系统与某个 PC 间的开放式用户通信

图 16-2 示例：通过系统 IP 地址 X1 与 S7-1513R 冗余系统进行通信

通过系统 IP 地址 X1、X2 和 X3 进行通信

可以为 S7-1500R/H 冗余系统的每个 PROFINET 接口使用一个系统 IP 地址。与 CPU 上 X1 接口相连的 PROFINET 设备通过系统 IP 地址 X1 进行通信。与 CPU 上 X2 接口相连的 PROFINET 设备通过系统 IP 地址 X2 进行通信。与 CPU 上 X3 接口相连的 PROFINET 设备通过系统 IP 地址 X3 进行通信。



① S7-1500R 冗余系统与另一 CPU 间的开放式用户通信

② 与 S7-1500R/H 冗余系统进行 HMI 通信

③ S7-1500R 冗余系统与某个 PC 间的开放式用户通信

图 16-3 示例：通过系统 IP 地址 X1 和 X2 与 S7-1515R 冗余系统进行通信

对于 S7-1500H(F) 系统，也可选择将系统分为几个 PROFINET 环网。

在这种情况下，必须在 Y 交换机后的单独 PROFINET 环网中连接所需 S1/S2 设备。

建议：为提高 S1/S2 设备的可用性，需要两个具有 DNA 冗余的 Y 型交换机 (SCALANCE XF204-2BA DNA)。一个 Y 型交换机承担 MRP 管理器和 DNA 管理器的角色。另一个 Y 型交换机承担 MRP 客户端和 DNA 客户端的角色。DNA 冗余只能通过连接的 PROFINET 环网实现。

更多关于使用 Y 型交换机的组态场景的信息，请参见《S7-1500R/H 冗余系统 (https://support.industry.siemens.com/cs/ww/zh/view/109754833) 系统手册》。

通过系统 IP 地址进行 IP 转发

如果使用系统 IP 地址作为通过 S7-1500R/H 冗余系统进行 IP 路由的网关/默认路由，则即使其中一个 CPU 出现故障，也会转发 IP 数据包。

在下图中，PC 连接到 S7-1500R CPU 的两个 X2 接口。在 PC 中输入系统 IP 地址 X2 作为网关，以获取到 HMI 设备的路径。HMI 设备通过交换机连接到 S7-1500 冗余系统的 PROFINET 环网。在 HMI 设备中，系统 IP 地址 X1 组态为路由器。

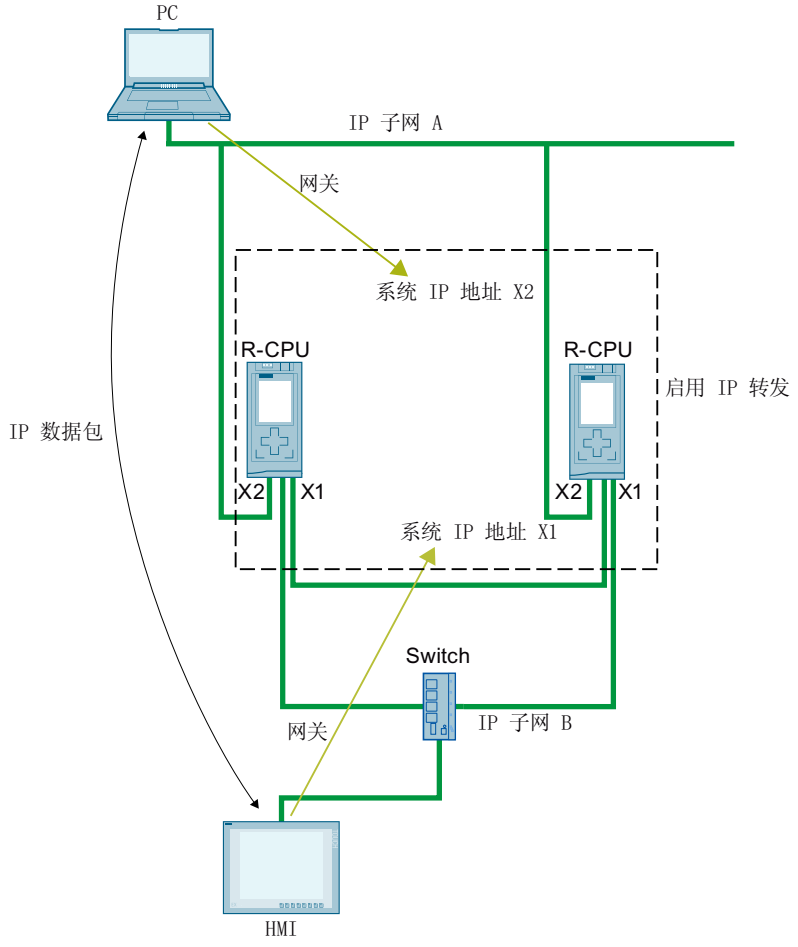


图 16-4 示例：通过系统 IP 地址进行 IP 转发

启用系统 IP 地址

要求：

- STEP 7 V15.1 及更高版本
- S7-1500R/H 冗余系统，带有两个 CPU，如两个 1513R-1 PN CPU

如果 S7-1500R/H 冗余系统的 CPU 配有两个 PROFINET 接口（X1 和 X2），则这两个 PROFINET 接口可使用一个系统 IP 地址。下文介绍了如何为 X1 接口启用系统 IP 地址。

要为 S7-1500R/H 冗余系统激活系统 IP 地址，请按以下步骤操作：

1. 在 STEP 7 的网络视图中，选择两个 CPU 中一个 CPU 的 X1 接口。
2. 在巡视窗口中，在“切换通信的系统 IP 地址”(System IP address for switched communication) 区域中选择“属性 > 常规 > 以太网地址”(Properties > General > Ethernet addresses)。
3. 选中“启用切换通信的系统 IP 地址”(Enable the system IP address for switched communication) 复选框。

   STEP 7 将自动创建一个系统 IP 地址。

图 16-5 组态系统 IP 地址

4. 根据需要调整系统 IP 地址。
5. 如有需要，可更改虚拟 MAC 地址。为此，应为“虚拟 MAC 地址”中的最后一个字节分配一个项目内唯一的值（值范围 01_H 到 FF_H）。

说明

虚拟 MAC 地址的唯一性

S7-1500R/H 冗余系统为每个系统 IP 地址使用地址范围 00-00-5E-00-01-00 到 00-00-5E-00-01-00 中的 MAC 地址。该地址范围也用于 VRRP（虚拟冗余协议）。

如果使用支持 VRRP 的设备（如交换机），则需确保 MAC 地址在以太网广播域中的唯一性。

结果：两个 CPU 上 X1 PROFINET 接口的 X1 系统 IP 地址已启用。

16.2 通信处理器的系统 IP 地址

简介

自 STEP 7 V19 起，可选择使用 CP 1543-1 通信处理器（自固件版本 V3.0 起）对 S7-1500R/H 冗余系统（固件版本 V3.1 起）进行扩展。使用 CP 1543-1 通信处理器进行扩展后，R/H-CPU 支持为 W1 虚拟接口组态设备和系统 IP 地址。连接到 CP 的通信伙伴通过这些系统 IP 地址与 R/H-CPU 通信。

可在 STEP 7 中启用 W1 的系统 IP 地址。

每个系统 IP 地址都有一个虚拟 MAC 地址

通信处理器故障

如果 CP 1543-1 通信处理器发生故障，S7-1500R/H 冗余系统的行为如下：

- S7-1500R：
  CP 发生故障时，相应的 CPU 会切换为 STOP 模式。S7-1500R 系统切换为 RUN-Solo 系统状态。
  位于主 CPU 上的 CP 发生故障时，S7-1500R 冗余系统会执行主-备用切换。分配的系统 IP 地址更改为新的主 CPU。随后会通过冗余 CP 建立新的通信连接。
  如果备用 CPU 上的 CP 发生故障，该故障不会影响现有的通过系统 IP 地址进行的通信。
- 带有源背板总线的 S7-1500H：
  CP 发生故障时，S7-1500H 系统会保持 RUN-Redundant 系统状态。
  如果主 CPU 上的 CP 发生故障，S7-1500H 冗余系统不会执行主-备用切换。系统 IP 地址 W1 通过主 CPU 分配给发生故障的 CP。要将系统 IP 地址 W1 切换到备用 CPU，可在用户程序中使用 mode=15 调用"RH_CTRL"指令。系统 IP 地址 W1 便会分配给备用 CPU。随后会通过备用 CPU 的冗余 CP 建立新的通信连接。

  如果主 CPU 上的另一个 CP 发生故障，仅可通过分配的设备 IP 地址在冗余 CP 上建立新的通信连接。

说明

如果通过设备 IP 地址与冗余系统通信，则最好使用主 CPU 的设备 IP 地址。使用主 CPU 的设备 IP 地址时，传输速率更高，通信负载更低。

更多关于"RH_CTRL"指令的信息，请参见 STEP 7 在线帮助。

有关组态 W1 虚拟接口的系统 IP 地址和虚拟 MAC 地址的信息，请参见《冗余系统 S7-1500R/H (https://support.industry.siemens.com/cs/cn/zh/view/109754833)》系统手册。

使用 CP 1543-1 通信处理器进行扩展的优势

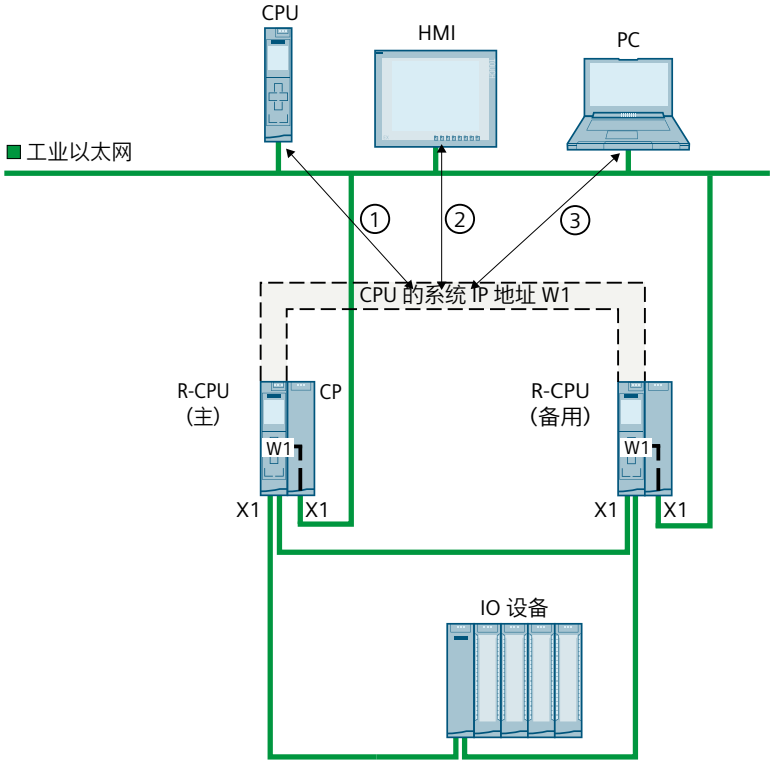
如果使用 CP 1543-1 通信处理器对冗余系统进行扩展，会提供额外的通信接口。对于只有一个 PROFINET 接口的 CPU 1513R-1 PN，可利用此优势实现网络隔离等操作。

使用 W1 系统 IP 地址的要求

- 已为 W1 虚拟接口分配 CP 1543-1。
- W1 虚拟接口的系统 IP 地址已激活。
- 已为 W1 虚拟接口分配系统 IP 地址。
- 通信伙伴可通过分配的系统 IP 地址访问两个 CP。

通过 W1 系统 IP 地址进行通信

下图显示了通过 CP 进行了扩展的 S7-1513R 冗余系统的组态。通信伙伴通过分配的 W1 系统 IP 地址与冗余系统进行通信。



① S7-1500R 冗余系统与另一 CPU 间的通信
② 与 S7-1500R/H 冗余系统进行 HMI 通信
③ S7-1500R 冗余系统与某个 PC 间的通信

图 16-6 示例：通过分配的 W1 系统 IP 地址与 S7-1513R 冗余系统进行通信

通过系统 IP 地址进行 IP 转发

如果使用系统 IP 地址作为通过 S7-1500R 冗余系统进行 IP 路由的网关/默认路由，则即使其中一个 CP 出现故障，也会转发 IP 数据包。

说明

带有源背板总线的 S7-1500H 系统中的主-备用切换

如果带有源背板总线的 S7-1500H 系统中的 CP 发生故障，则不会进行主-备用切换。在这种情况下，可根据需要在用户程序中设置主-备用切换。

在下图中，PC 连接到 CP 1543-1 的两个 X1 接口。对于到 HMI 设备的路由，在 PC 中输入分配的 W1 系统 IP 地址作为路由器。HMI 设备通过交换机连接到 S7-1500R 冗余系统的 PROFINET 环网。在 HMI 设备中，输入 CPU 的 X1 系统 IP 地址作为路由器。

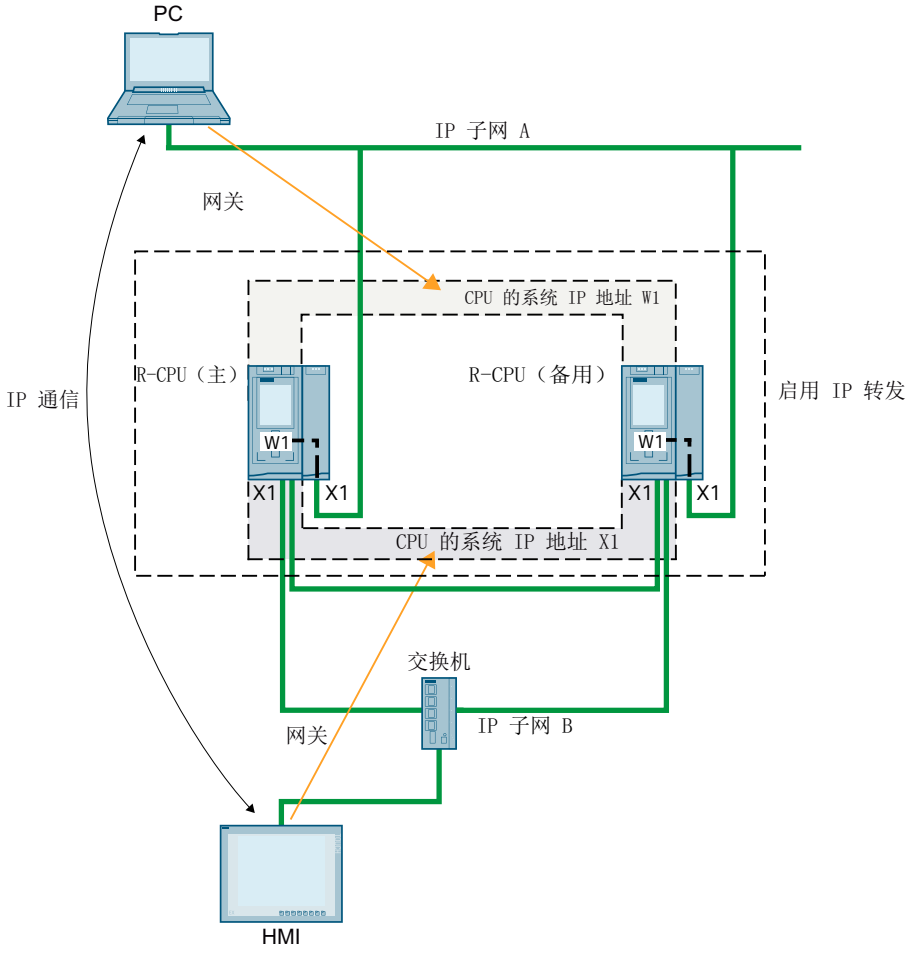


图 16-7 示例：通过 CPU 和 CP 的系统 IP 地址进行 IP 转发

为 R/H CPU 的 W1 虚拟接口分配系统 IP 地址。

自 STEP 7 V19 起，可以为 R/H CPU 的 W1 虚拟接口分配系统 IP 地址。已连接到 CP 的通信伙伴通过分配的系统 IP 地址与 R/H 系统进行通信。

要更改 R/H CPU 的 W1 虚拟接口的系统 IP 地址，请按以下步骤操作：

1. 在设备视图中选择 rail_0 上的 R/H CPU。
2. 在巡视窗口中，浏览到"高级组态 > 通过通信模块访问 PLC"(Advanced configuration > Access to PLC via communications module)。
3. 在"分配接口 [W1]"(Assign interface [W1]) 部分，从"选择通信模块"(Select communication module) 选择列表中选择所需 CP。
   选择后，会出现一条警告消息，告知用户此设置可能伴随的风险。如果确认警告消息，则会显示 W1 虚拟接口的组态选项。
4. 浏览到"Internet 协议版本 4 (IPv4)"(Internet protocol version 4 (IPv4)) 区域。
5. 接受建议或分配新的 IP 地址。
6. 浏览到"切换通信的系统 IP 地址"(System IP address for switched communication) 区域。
7. 选择"启用切换通信的系统 IP 地址"(Enable the system IP address for switched communication) 选项。
8. 接受建议或分配新的 IP 地址（以及虚拟 MAC 地址）。

9. rail_1 上的 R/H CPU 会自动采用相应的设置。

图 16-8 组态 W1 的系统 IP 地址

结果：已为 W1 虚拟接口分配 CP 1543-1，并已组态设备/系统 IP 地址。

16.3 通过切换系统 IP 地址来提高可访问性

CPU 无法访问时切换系统 IP 地址

默认情况下，系统 IP 地址分配给主 CPU。

从 TIA Portal V20（CPU 固件版本 V4.0 或更高版本）开始，可以使用其它模式将备用 CPU 的系统 IP 地址分配给"RH_CTRL"指令，以实现相应目的，例如，在主 CPU 无法访问时仍能保持通信。

更多信息

有关通过程序控制方式切换系统 IP 地址的说明，请参见《S7-1500R/H 系统手册》(https://support.industry.siemens.com/cs/cn/zh/view/109754833)中的 S7-1500R/H 冗余系统的特殊说明。

16.4 对 Snycup 状态的响应

SYNCUP 系统状态下通过系统 IP 地址的通信连接的响应

- HMI、PG 连接和 S7 连接临时关闭。在 SYNCUP 组态下，短时内无法与 S7-1500R/H 冗余系统建立连接。
- 所有现有的开放式用户通信连接均中断：
  - 在 SYNCUP 后，冗余系统中的 CPU 将作为主动连接伙伴重新建立连接。
  - 在 SYNCUP 后，S7-1500R/H 冗余系统将重新建立连接端点，从而建立被动连接。
- 系统将停止 TSEND 和 TRCV 指令中正在运行的实例处理过程。块参数 STATUS 将返回 $80C4_H$（暂时性通信错误）。

16.5 主/备份 CPU 切换响应

主 CPU 与备用 CPU 切换过程中，通过系统 IP 地址的通信连接响应

- 系统将停止 TSEND 和 TRCV 指令当前正在运行的实例并返回状态 $80C4_H$（暂时性通信错误）。
- 新的主 CPU 将重新建立之前与 S7-1500R/H 冗余系统的成功连接。
- 新的主 CPU 将重新建立连接端点，从而建立被动连接。

说明

延长了连接中断的持续时间

如果远程系统在主 - 备用 CPU 切换后未主动传送数据，可能需要由远程系统执行连接监视（例如 TCP-Keep-Alive 或应用程序），直至可以重新建立连接为止。

16.6 冗余系统 S7-1500R/H 的连接资源

S7-1500R/H 冗余系统的最大连接资源数

S7-1500R/H 冗余系统支持最大数量的连接资源。

所用 CPU 将确定冗余系统的最大资源数量：

- CPU 1513R：最多 88 个连接资源
- CPU 1515R：最多 128 个连接资源（V3.0 及更高版本），最多 108 个连接资源（V2.9.x 及以下版本）
- CPU 1517H：最多 288 个连接资源
- CPU 1518HF：最多 320 个连接资源

连接资源的分配

通信连接会占用 S7-1500R/H 冗余系统中的通信资源。

冗余系统 S7 1500R/H 的每条通信连接都会占用 S7 1500R/H 站中的连接资源。S7-1500R/H 站中包含 S7-1500R/H 冗余系统的两个 CPU 的硬件设置。

根据所使用的 IP 地址，通信连接还将占用 S7-1500R/H 冗余系统中一个或两个 CPU 的连接资源。S7-1500R/H 站也可用于建立通信连接。

下表根据所使用的 IP 地址列出了各 CPU 中通信连接占用的连接资源。

连接方式...	站的连接资源	冗余 ID 为 1 的 CPU 的连接资源	冗余 ID 为 2 的 CPU 的连接资源
系统 IP 地址	√	√	√
冗余 ID 为 1 的 CPU 的设备 IP 地址	√	√	-
冗余 ID 为 2 的 CPU 的设备 IP 地址	√	-	√

在 STEP 7 中显示已占用的连接资源

要求：在线连接到 S7-1500R/H 冗余系统

有关在线显示连接资源的信息，请参见巡视窗口中的“诊断 > 连接信息”(Diagnostics" > "Connection information)。STEP 7 通常会显示所选 CPU 与 S7-1500R/H 站的连接资源。

连接资源

	站资源					模块资源	
	预留			动态		CPU 1517H-3 PN (R0/S1)	
最大资源数：		10	10	150	150	160	160
	最大	已组态	已用	已组态	已用	已组态	已用
PG 通信：	4	-	2	-	0	-	2
HMI 通信：	4	0	0	0	0	0	0
S7 通信：	0	-	0	0	0	0	0
开放式用户通信：	0	-	0	0	0	0	0
Web 通信：	2	-	0	-	0	-	0
其它通信:	-	-	0	0	0	0	0
使用的总资源：		0	2	0	0	0	2
可用资源：		10	8	150	150	160	158

图 16-9 在 STEP 7 中显示 S7-1500R/H 冗余系统的连接资源

16.7 与冗余系统 S7-1500R/H 进行 HMI 通信

16.7.1 通过系统 IP 地址进行 HMI 连接

要求

- S7-1500R/H 冗余系统，如 CPU 1513R-1PN
- 系统 IP 地址已启用
- 带有 PROFINETI 接口的 HMI 设备

操作步骤

要与 S7-1500R/H 冗余系统建立 HMI 连接，请按以下步骤操作：

1. 在 STEP 7 的网络视图中，选择 HMI 设备的 PROFINET 接口。
2. 使用拖放操作，在 HMI 设备的 PROFINET 接口与 S7-1500R/H 冗余系统的 PROFINET 接口之间绘制一条线。
   HMI 设备与 S7-1500R/H 冗余系统将连接在一起。

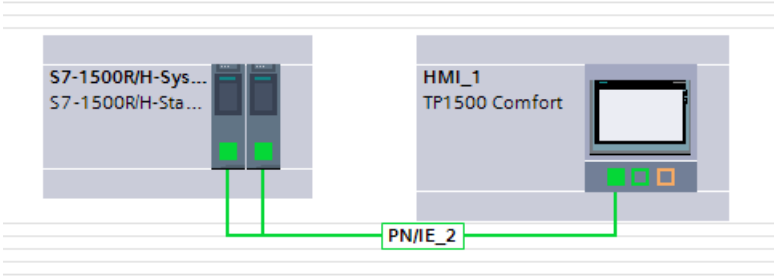


图 16-10 联网 HMI 设备与 S7-1500R/H 冗余系统

3. 在功能表中，单击“连接”(Connections) 图标。该操作将激活连接模式。
4. 使用拖放操作，在 HMI 设备的 PROFINET 接口与 S7-1500R/H 冗余系统的 CPU 间画一条线。
   “连接伙伴”(Connection partners) 列表随即打开。

图 16-11 与 S7-1500R/H 冗余系统建立 HMI 连接

5. 在“连接伙伴”(Connection partners) 列表中，选择 S7-1500R/H 冗余系统。

结果：在 HMI 设备和 S7-1500R/H 冗余系统间建立了一条 HMI 连接。HMI 连接将使用该系统 IP 地址。HMI 设备始终与主 CPU 相连。

将 HMI 连接更改为设备 IP 地址

要将 HMI 连接永久地更改为所选择的 CPU，则需取消选择 HMI 连接属性中的"使用切换通信的系统 IP 地址"(Use system IP address for switched communication) 复选框。HMI 连接之后将使用该 PROFINET 接口的设备 IP 地址。如果该 CPU 发生故障，则与该 CPU 的 HMI 连接将永久失效。

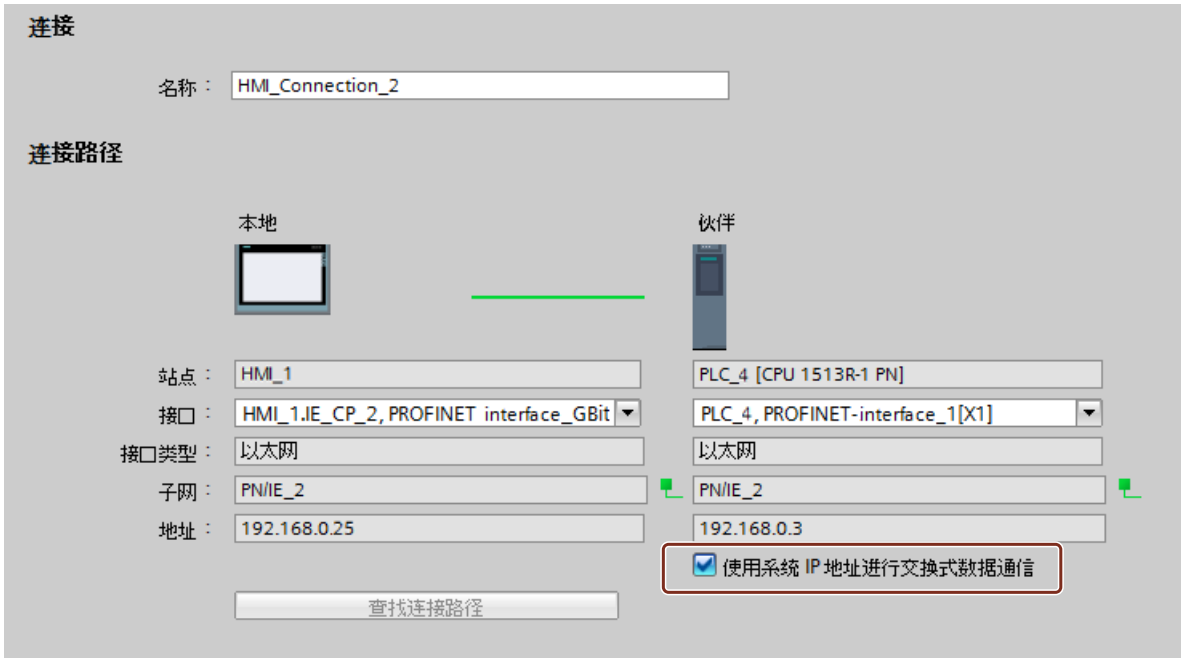


图 16-12 HMI 连接的属性

说明

自动建立 HMI 连接

将一个变量从 S7-1500R/H 冗余系统拖放到 HMI 画面或 HMI 变量表中时，STEP 7 将自动建立一条 HMI 连接。系统默认，HMI 设备的 PROFINET 接口与冗余 ID 为 1 的 CPU 的 PROFINET 接口 X1 间存在这样一条 HMI 连接。该连接使用 PROFINET 接口 X1 的设备 IP 地址。

在 HMI 连接属性中，可将 HMI 连接更改为一个系统 IP 地址。

更多信息

可通过设备 IP 地址建立与 S7-1500R/H 冗余系统的 HMI 连接。借助 HMI 组态中的脚本，故障 CPU 的连接会自动切换到仍在运行的 CPU。有关步骤的说明，请参见以下常见问题解答 (https://support.industry.siemens.com/cs/cn/zh/view/109781687)。

16.8 与冗余系统 S7-1500R/H 进行开放式用户通信

简介

* 自固件版本 V3.1 起，S7-1500R/H 系统还支持开放式用户安全通信（安全 OUC）。

如果使用 CP 1543-1 通信处理器对 S7-1500R/H 系统（固件版本 V3.1 起）进行扩展，还可通过这些连接的 CP 进行安全 OUC。

要求：

- STEP 7 V19 及更高版本
- CP 1543-1 固件版本 V3.0 及更高版本

S7-1500R/H 冗余系统的开放式用户安全通信协议

下表列出了 S7-1500R/H 冗余系统中可使用的开放式用户通信协议以及相应的系统数据类型和指令。

表格 16-1 可在与 S7-1500R/H 冗余系统进行开放式用户通信时使用的协议、系统数据类型和指令

协议	系统数据类型	指令
TCP	• TCON_QDN • TCON_QDN_SEC • TCON_IP_v4 • TCON_IP_V4_SEC	建立连接并通过以下指令收/发数据： • TSEND_C/TRCV_C 或 • TCON，TSEND/TRCV 或 • TCON，TUSEND/TURCV （可通过 TDISCON 终止连接）
TLS over TCP		
ISO-on-TCP	• TCON_IP_RFC	
UDP	• TCON_IP_v4 • TADDR_Param • TADDR_SEND_QDN • TADDR_RCV_IP	建立连接并通过以下指令收/发数据： • TSEND_C/TRCV_C • TUSEND/TURCV/TRCV （可通过 TDISCON 终止连接）
Modbus TCP	• TCON_IP_v4 • TCON_IP_V4_SEC • TCON_QDN • TCON_QDN_SEC	• MB_CLIENT • MB_RED_CLIENT • MB_SERVER • MB_RED_SERVER

16.8.1 与冗余系统 S7-1500R/H 建立开放式用户通信连接

通过 CPU 的集成 PROFINET 接口进行开放式用户通信

S7-1500R/H 冗余系统可通过开放式用户通信与其它设备进行通信。

在用户程序中可通过"TSEND_C"之类的指令建立连接。S7-1500R/H 冗余系统不支持所组态的连接。

可通过集成 PROFINET 接口的以下 IP 地址建立连接：

- 设备 IP 地址：
  在冗余模式下，冗余系统可建立或终止连接，并通过冗余系统的所有设备 IP 地址发送或接收数据。
  如果通过设备 IP 地址建立连接，则将通过关联的 CPU 进行通信路由。如果 CPU 出现故障，通过该 CPU 的设备 IP 地址进行的所有通信都会失败。
- 系统 IP 地址：
  如果通过系统 IP 地址建立连接，则始终会通过主 CPU 进行通信路由。

根据用例确定希望采用哪种方式建立连接。

通过 CP 1543-1 通信处理器进行开放式用户通信

如果通过 CP 1543-1 通信处理器对 S7-1500R/H 冗余系统进行扩展，则还可以使用以下通信选项：

- CP 1543-1 的本地 X1 接口
- 每个 R/H CPU 的 W1 虚拟接口的自有设备 IP 地址
- R/H 系统的 W1 虚拟接口的公共系统 IP 地址

有关如何为 S7-1500R/H 系统使用 CP 1543-1 的信息，请参见"通信处理器的系统 IP 地址 (页 413)"部分。

通过系统 IP 地址建立连接

下文介绍了如何通过 S7 1500R/H 冗余系统的集成 PROFINET 接口的系统 IP 地址与伙伴 CPU 建立连接。

在 S7-1500R/H 冗余系统的用户程序中，可通过 TSEND_C 等指令建立连接。在伙伴 CPU 的用户程序中，创建相应的 TRCV_C 指令。

在此，我们将以 S7-1500R/H 冗余系统与 CPU 1516-3PN/DP 间的 TCP 连接为例，进行详细说明。

要求

- S7-1500R/H 冗余系统作为 TCP 客户端，例如 2 个 CPU 1513-1PN
- PROFINET 接口 X1 的系统 IP 地址已启用
- 接口伙伴作为 TCP 服务器，例如 CPU 1516-3 PN/DP
- 冗余 CPU 1513R 的 PROFINET 接口 X1 与 CPU 1516-3PN/DP 的 PROFINET 接口 X2 位于同一子网中。

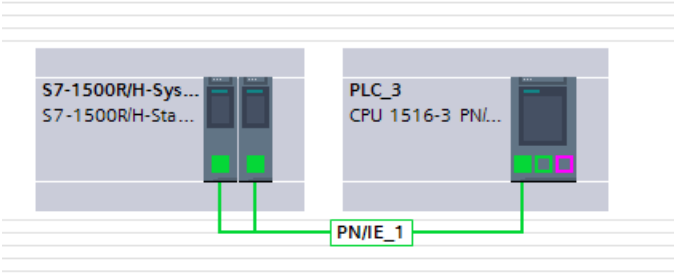


图 16-13 TCP连接的组态示例

S7-1500R/H 冗余系统内用户程序中的 TSEND_C 指令

要与其它 CPU 建立 TCP-连接，请按以下步骤操作：

1. 在用户程序中，创建一个“TSEND_C”指令。

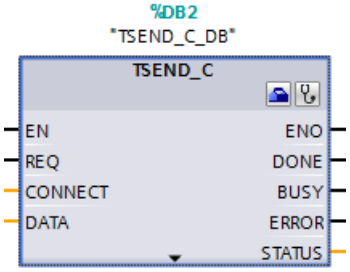


图 16-14 S7-1500R/H：“TSEND_C”指令

2. 选择“TSEND_C”指令。
3. 在巡视窗口中，浏览到“属性 > 组态 > 连接参数”(Properties > Configuration > Connection parameters)。
   在左侧，S7-1500R/H 冗余系统为该连接的本地端点：
   - “接口”(Interface)：X1 为当前接口。
   - “子网：”(Subnet:)：如果接口 X1 分配给 S7 子网，则 STEP 7 中会显示该 S7 子网的名称。
   - “使用系统 IP 地址”(Use system IP address) 复选框已启用。S7-1500R/H 冗余系统的系统 IP 地址位于“地址”(Address) 中。

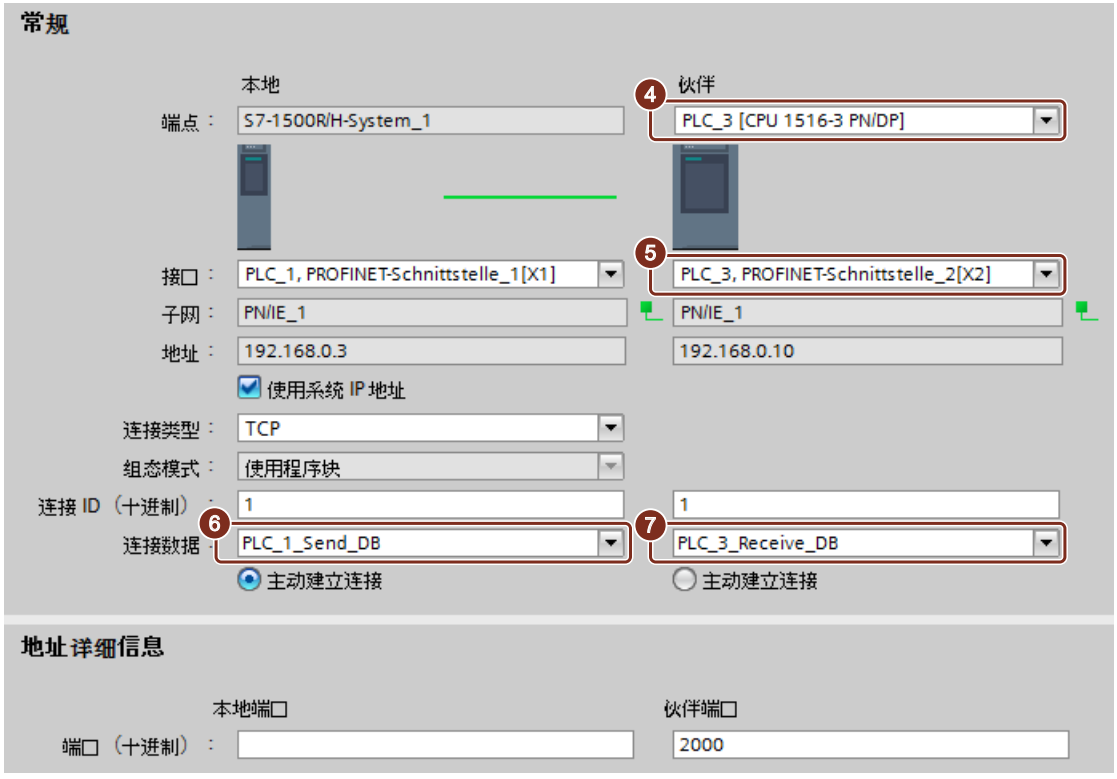


图 16-15 S7-1500R/H：在 STEP 7 中为指令“TSEND_C”分配参数：

4. 在“端点：”(End point:) 下方的“伙伴”(Partner) 中，选择 CPU 1516-3 PN/DP 作为通信伙伴。
5. 在“接口：”(Interface:) 下方的“伙伴”(Partner) 中，选择 CPU 1516-3 PN/DP 的 PROFINET 接口 X2。
6. 在“连接数据”(Connection data) 下的“本地”(Local) 中，选择设置“<新建>”(<new>)。
   STEP 7 将在 S7-1500R/H 冗余系统的用户程序中为连接数据创建一个数据块，例如“PLC_1_Send_DB”。
   “TCP”默认设置为该连接类型。
7. 在“连接类型”(Connection type) 下的“伙伴”(Partners) 中，选择设置“新建”(NEW)。
   STEP 7 将在其它 CPU 的用户程序中为连接数据创建一个数据块，例如“PLC_3_Receive_DB”。

CPU 1516 3PN/DP 用户程序中的 TRCV_C 指令

在 CPU 1516-3PN/DP 的用户程序中创建一个 TRCV_C 指令并按照以下方式分配参数：

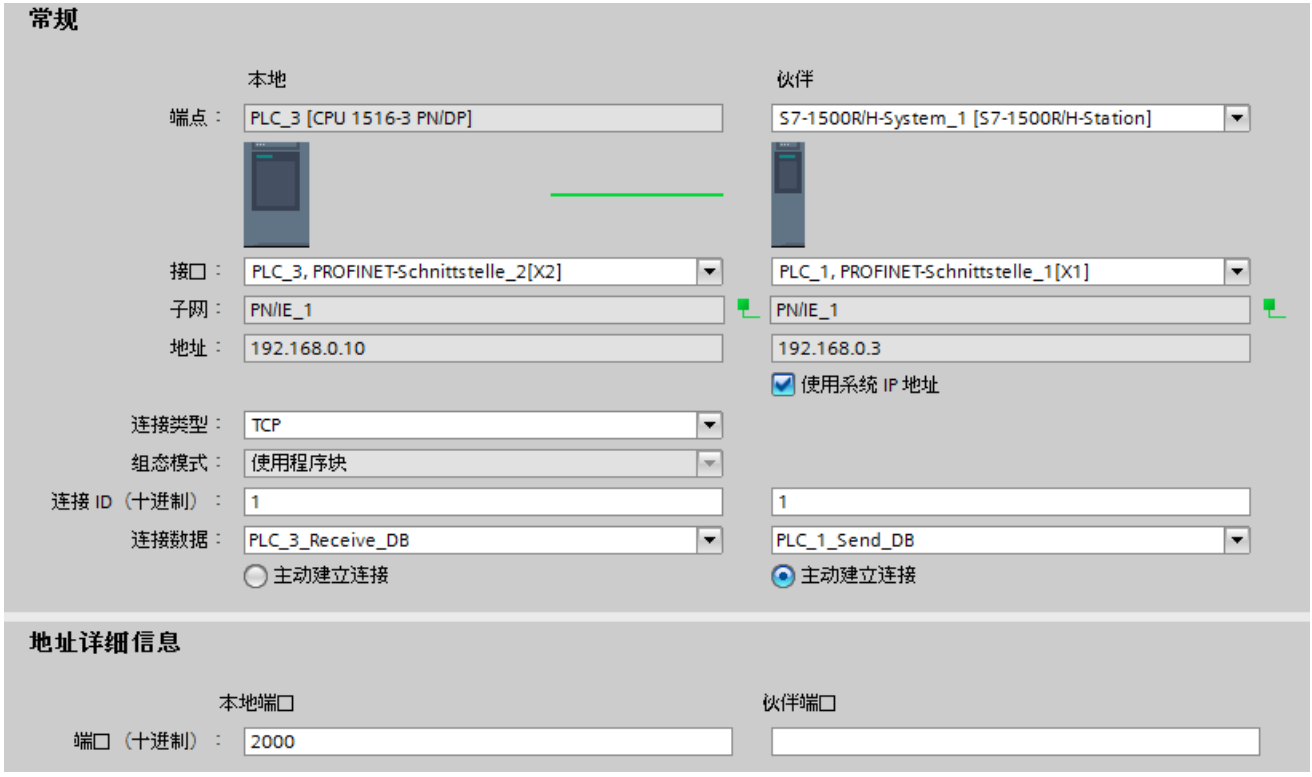


图 16-16 S7-1500-3PN/DP：在 STEP 7 中为指令“TRCV_C”分配参数：

通过设备 IP 地址建立连接

要通过两个 CPU 中一个 CPU 的设备 IP 地址建立 OUC 连接：

- 选择 S7-1500R/H 冗余系统中所需的 PROFINET 接口。
- 取消选中“使用系统 IP 地址”(Use system IP address) 复选框。

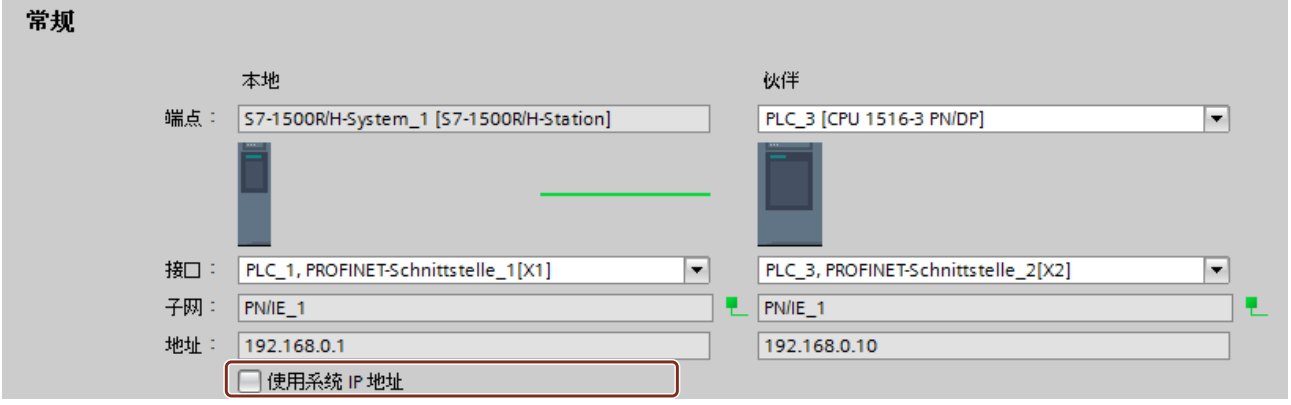


图 16-17 通过设备 IP 地址建立 OUC 连接

更多信息

有关系统状态的更多信息，请参见系统手册《S7-1500R/H (https://support.industry.siemens.com/cs/ww/zh/view/109754833)》。

有关 PROFINET IO 系统的组态和参数分配的更多信息，请参见《PROFINET 功能手册 (https://support.industry.siemens.com/cs/cn/zh/view/49948856)》。

16.8.2 与 CP 1543-1 进行开放式用户通信

简介

通过 CP 1543-1 通信处理器，可同时使用开放式用户通信 (OUC) 以及安全 OUC。

如果要使用安全 OUC，必须满足以下附加要求，以便处理设备和 CA 证书：

- 项目保护已激活
- 已在每个要用于安全 OUC 的 CP 1543-1 中启用安全功能。
- 同一插槽中主 CPU 和备用 CPU 上的每个 CP 的安全设置相同
- 为两个 CP 组态相同的 CA 证书
- CP 的设备证书分别引用两个 CP，例如，在两个设备证书中组态使用者可选名称 (SAN)。

说明

不会自动同步安全设置

在 STEP 7 中，不会在 CP 之间自动同步安全设置。因此，对主 CPU 和备用 CPU 的同一插槽中的 CP 进行相同组态。

通过 CP 进行开放式用户通信的连接选项

对于以下组态，始终会通过本地 CP 接口与伙伴建立 OUC 连接，而不考虑使用的是 OUC 还是安全 OUC。建立通信连接时，需要根据使用的组态调整连接参数。使用以下选项之一在连接参数中选择接口的相应硬件标识符：

- 使用分配了本地 IP 地址的 CP 接口：
  对于“InterfaceId”连接参数，指定 CP 本地以太网接口的硬件标识符，例如“Local1~CP_1543-1_1~Ethernet_interface_1”。在 RUN-Redundant 系统状态下使用主 CPU 和备用 CPU 的 CP 接口。
- 使用组态了设备 IP 地址的 CPU 的 W1 虚拟接口：
  对于“InterfaceId”连接参数，指定 CPU 虚拟接口的硬件标识符，例如“Local1~Virtual_communication_interface”。在 RUN-Redundant 系统状态下使用主 CPU 和备用 CPU 的 W1 虚拟接口
- 使用组态了系统 IP 地址的 CPU 的 W1 虚拟接口：
  对于“InterfaceId”连接参数，使用字符串“...HSystemIPRef...”和默认名称中的“Virtual”标识符组态对象的硬件标识符，例如“Local1~Virtual_communication_interface~HsystemIPRef_1”。该对象随即会引用 CPU 的 W1 虚拟接口的系统 IP 地址。

通过 CP 创建连接

可为 S7-1500R/H 冗余系统的集成 PROFINET 接口创建连接。但需要对连接参数进行相应调整。在“接口”(Interface) 选择列表中选择要使用的 CP 1543-1 通信处理器。

如果要使用“使用系统 IP 地址”(Use system IP address) 选项，则必须为 W1 虚拟接口组态系统 IP 地址。

为通过 CP 进行的开放式用户安全通信选择证书

根据对配有 CP 1543-1 的 S7-1500R/H 站的组态方式，以下证书规则适用于与连接伙伴进行的通信：

- 通过使用本地 IP 地址的 CP 进行通信：
  使用 CP 的设备证书进行身份验证。为了使连接伙伴能够验证设备证书，“伙伴设备证书”(Certificates of the partner devices) 下必须提供 CP 的 CA 证书。
- 通过组态了系统 IP 地址的 CPU 的 W1 虚拟接口进行通信：
  使用 CPU 的设备证书进行身份验证。为了使连接伙伴能够验证设备证书，“伙伴设备证书”(Certificates of the partner devices) 下必须提供 CPU 的证书颁发机构。

更多信息

有关如何创建或分配证书的信息，请参见“管理证书 (页 56)”部分。

有关通过 CP 接口进行开放式用户安全通信的信息，请参见“通过 CP 接口进行安全 OUC 连接 (页 87)”部分。

16.9 在 S7-1500R/H 系统中使用 OPC UA 服务器

16.9.1 S7-1500R/H 系统中 OPC UA 服务器的实用信息

自固件版本 V3.1 起，S7-1500R/H CPU 支持 OPC UA 服务器功能。不支持 OPC UA 客户端。

S7-1500R/H 系统的 OPC UA 服务器的扩展符合 OPC 10000-4 规范：服务（版本 1.04），考虑到下述限制。

OPC UA 服务器可通过 CPU 的所有集成接口以及 CP 1543-1 进行访问。为此，必须通过 CPU 的虚拟接口 (W1) 连接 CP（CPU 属性中的“通过通信模块访问 PLC”(Access to PLC via communication module) 区域）。

通过 OPC UA 服务器可实现对 CPU 的开放式和标准化访问。在 R/H 系统中，OPC UA 服务器分别在两个 CPU 上运行。主 CPU 和备用 CPU 的 OPC UA 服务器通过冗余机制进行同步。在冗余操作中，从 OPC UA 客户端的角度来看，它是一个服务器应用程序。

与标准 CPU 相比，S7-1500R/H CPU 提供高级信息模型。通过这种高级信息模型，客户可以考虑冗余系统的具体功能。

从 OPC UA 客户端的角度来看，使用与访问标准 S7-1500 CPU 相同的机制来访问 S7-1500R/H 系统（例如发现服务）。

OPC UA 服务器的限制

OPC UA 服务器用于 S7-1500R/H 系统时，相对于标准 S7-1500 CPU 的功能范围，存在以下限制：

- 仅通过服务器接口进行数据访问。标准 SIMATIC 服务器接口不受支持。因此，也不支持节点集导出（标准 SIMATIC 服务器接口的 OPC UA XML 文件）。
- 不支持 GDS（运行时认证管理）。
- 不支持报警和条件。
- 无法通过 OPC_UA_ReadList 和 OPC_UA_WriteList 指令访问 CPU 自身 OPC UA 服务器的地址空间。

  即，无法实现 TIA Portal 信息系统以下部分中描述的功能：
  - 使用 OPC_UA_ReadList 诊断 OPC UA 服务器
  - 使用 OPC_UA_WriteList 设置 OPC UA-DataValue
- 服务器提供的时间戳分辨率为 1 ms（标准 CPU：1 ns）。

有关 S7-1500R/H 冗余系统与 S7-1500 自动化系统相比的一般限制的概述，请参见《S7-1500R/H 冗余系统》(https://support.industry.siemens.com/cs/cn/zh/view/109754833)系统手册中的"应用规划 > 限制"部分。

有关如何组态 OPC UA 服务器（例如创建服务器接口）的说明，请参见"OPC UA 通信 (页 155)"部分或 TIA Portal 的信息系统。

对周期和响应时间的影响

如果启用 S7-1500R/H 系统的 OPC UA 服务器，则将影响周期和响应时间。

必须考虑以下影响：

- 通过激活 OPC UA 服务器，可以延长处于 RUN-Redundant 系统状态的 S7-1500R/H 系统的循环时间。
- 例如，较高优先级报警中断循环 OB 的准确性会降低。示例：循环中断的"抖动"增加。
- 否则，以下文档中的基本规则适用：《周期和响应时间》功能手册 (https://support.industry.siemens.com/cs/cn/zh/view/59193558)。

⚠ 小心
在调试阶段全面测试最大循环时间 在调试阶段，检查 S7-1500R/H 系统即使在最差情况下（报警、来自最大客户端数量的通信负载（包括连接/断开））是否能够在设定的最大循环时间内可靠工作。

要求

- 已安装硬件支持包 (HSP)"HSP_V19_0445_001_S71500_RH_3.1"的 TIA Portal V19。

  S7-1500R/H CPU 随此 HSP 一起安装。

OPC UA 中的冗余

由两台或更多服务器组成的冗余系统在 OPC UA 中称为"冗余服务器集"。符合 OPC UA 规范的服务器冗余可识别以下工作模式 (Modes of Redundancy)：

- Transparent Mode (transparent Redundancy)

  在透明工作模式下，服务器单独负责发生错误时的故障转移 (Failover)。OPC UA 客户端不需要对冗余有任何了解即可继续数据交换。要在 Transparent Mode 下工作，客户端必须连接到 CPU 接口的系统 IP 地址。如果使用 CP 1543-1，请使用具有已组态系统 IP 地址的 CPU 的虚拟接口 W1。

- Non-transparent Mode (non-transparent Redundancy)

  在非透明模式下，客户端负责在发生错误时从一台服务器切换到另一台服务器 (Failover)。客户端必须启动所需的操作才能从冗余系统中受益。要在 non-transparent Mode 下工作，客户端必须连接到 R/H CPU 对应的设备 IP 地址。如果使用 CP 1543-1，请使用已组态设备 IP 地址的 CPU 的虚拟接口 W1。

S7-1500R/H 系统支持这两种工作模式。

Redundant Server Set 支持通过"ServiceLevel"变量（一个以数字方式反映服务器"健康状态"的字节）访问关于可用性的信息。

ServiceLevel 允许得出有关冗余状态的结论，例如 CPU 是主 CPU 还是备用CPU，或者系统是否处于 RUN-Redundant 系统状态。有关 ServiceLevel 的信息，请参见"非透明模式 (页 434)"部分。

更多信息

可以在以下条目中找到有关将 OPC UA 服务器与 S7-1500R/H 系统结合使用的其它最新信息：常见问题解答：如何在 S7-1500 R/H 系统中使用 OPC UA 服务器？
(https://support.industry.siemens.com/cs/ww/zh/view/109822965)

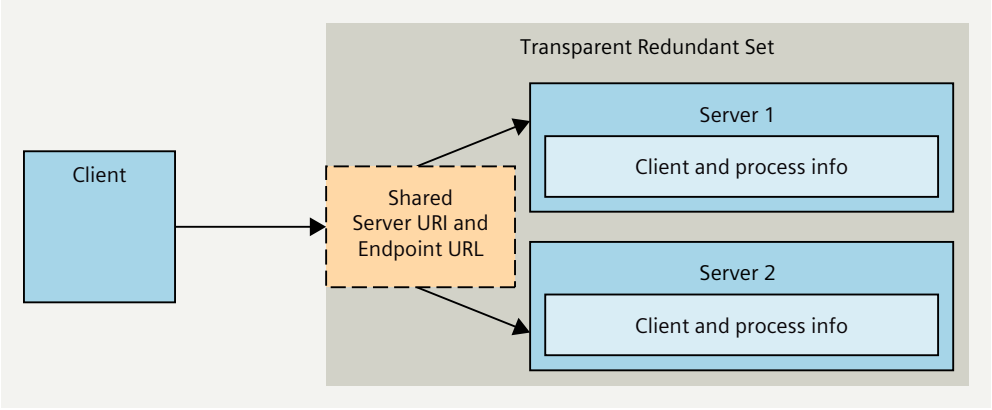
16.9.2 Transparent Mode (transparent Redundancy)

下面解释了有关透明冗余的详细信息，即它是一种允许客户端像单个系统或标准 CPU 的 OPC UA 服务器一样处理冗余系统的工作模式。

主 CPU 始终可通过系统 IP 地址访问

对于 S7-1500R/H 站，使用系统 IP 地址进行透明冗余。通过系统 IP 地址，客户端访问主 CPU，即当前托管该 IP 地址的 CPU。

下图给出了根据 OPC UA 10000-4 使用透明冗余的示例。



主 CPU 丢失时的响应 (RUN-Redundant > RUN-Solo)

如果发生错误（故障转移），客户端与服务器之间的通信会短暂中断。所有已建立的会话和订阅都将保留，并可以使用新的主 CPU 恢复。在这种情况下，从客户端的角度来看，行为对应于会话也会恢复的临时通信错误情况下的行为。

故障转移后，一旦使用系统 IP 地址访问新的主 CPU，客户端便会恢复通信（重新激活会话）。

恢复冗余时的行为 (SYNCUP > RUN-Redundant)

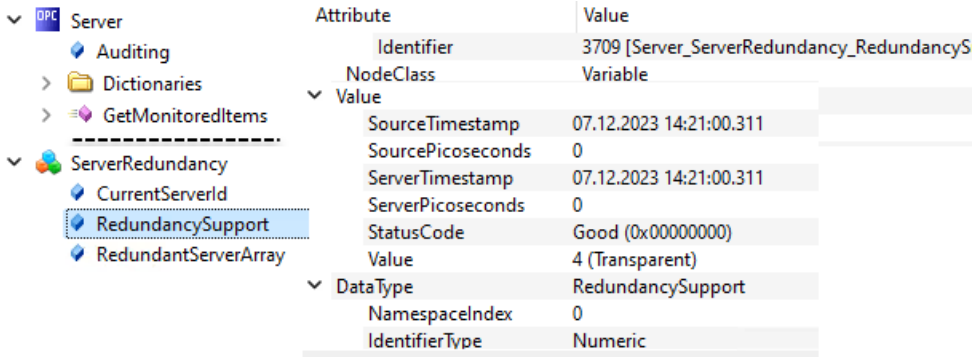
如果在故障转移后（例如更换有故障的 CPU 后）恢复冗余系统的初始状态，系统将执行 SYNCUP。

在 SYNCUP 期间，OPC UA 服务器在两个 CPU 中重新启动。所有现有会话和订阅都将被删除。

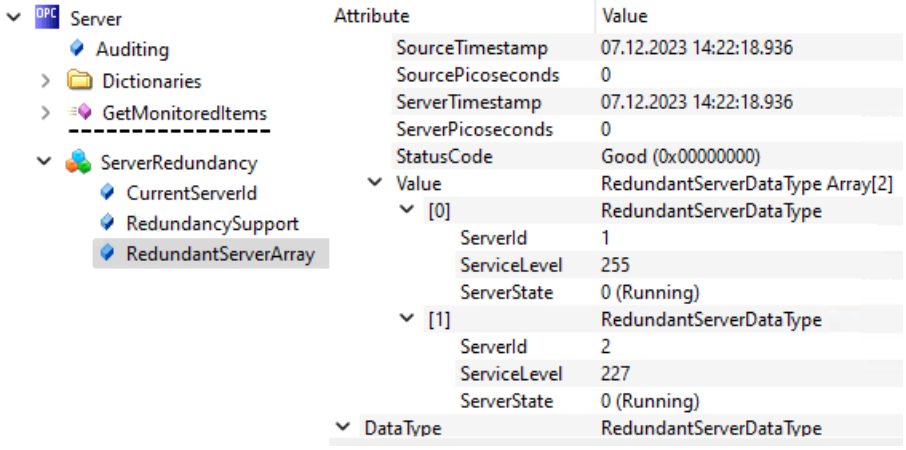
一旦主 CPU 的服务器再次可访问，客户端就必须重新创建会话和订阅。

信息模型中用于透明冗余的相关节点

可以在信息模型中的“ServerRedundancy”对象（命名空间索引“0”或命名空间“http://opcfoundation.org/UA/”）下找到有关 R/H 站透明模式的信息。



可使用 RedundantServerArray 确定 ServerState 和 ServiceLevel 的冗余服务器：



有关 ServiceLevel 的更多信息，请参见下一节。

16.9.3 Non-transparent Mode (non-transparent Redundancy)

下面会详细说明非透明冗余。在此工作模式下，OPC UA 为客户端提供以下信息：

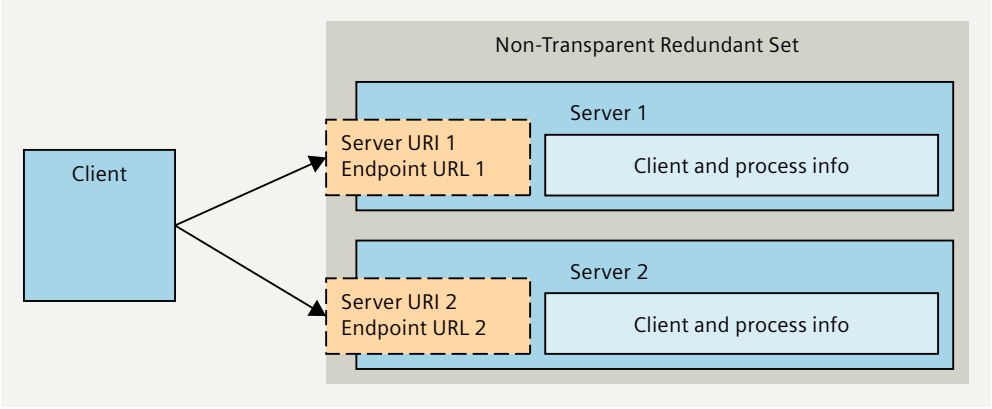
- 冗余服务器集中有哪些服务器可用
- 支持可用服务器之间哪些类型的故障转移

此信息允许客户端确定需要执行哪些操作来实现切换。客户不仅必须能够解释 ServerRedundancy 属性，还需要了解冗余服务器的扩展信息模型。

R/H 系统中冗余 OPC UA 服务器的地址参数

与透明冗余相反，R/H 系统中的服务器通过其唯一的 IP 地址（如果是已组态的 CP 1543-1，则为设备 IP 地址或虚拟 IP 地址）进行寻址。

下图给出了根据 OPC UA 10000-4 使用非透明冗余的示例。



主 CPU 丢失时的响应：故障转移模式

根据 OPC 1000-4: Services，服务器可以支持不同的故障转移模式：Cold、Warm、Hot 和 HotAndMirrored。

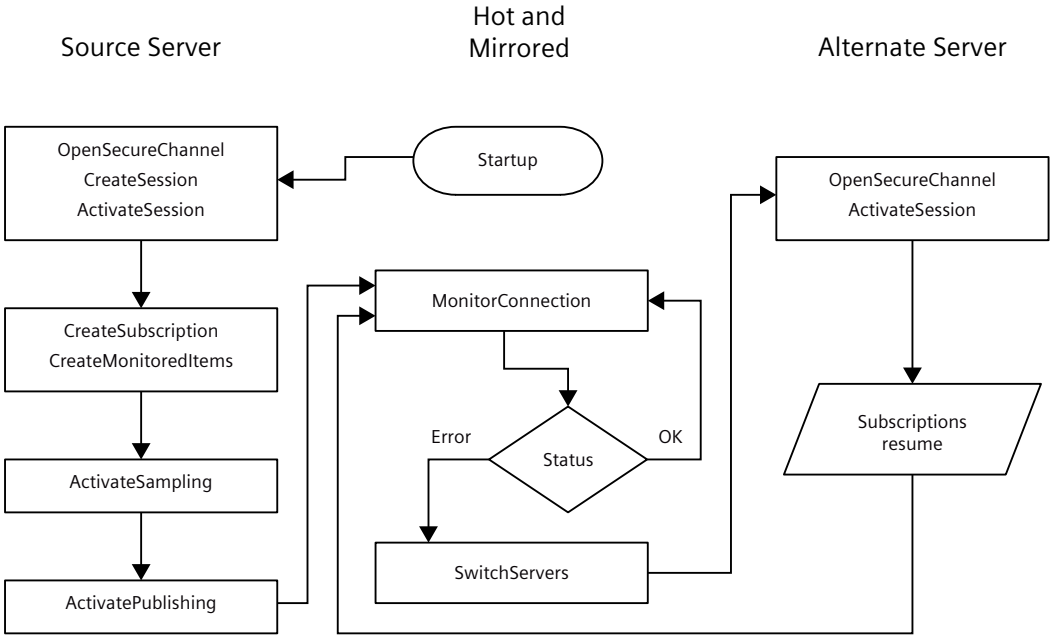
S7-1500R/H CPU 的 OPC UA 服务器支持“HotAndMirrored”故障转移模式。

客户端必须了解 OPC UA 服务器为冗余服务器，并且由于支持的 Failover modes，必须以某种方式执行故障转移过程。

OPC 10000-4 规范：OPC 统一架构中的服务 -“客户端故障转移行为 - HotAndMirrored”部分。客户端负责跟踪服务器的状态，即定期查询 ServiceLevel 的状态，并在发生错误时切换到备用服务器（对于 S7-1500R/H 系统，这意味着备用 CPU）。这种类型的冗余控制具有以下优点：

- 客户端可以完全控制替代服务器的选择。
- 客户端可以通过单独的网络访问 OPC UA 服务器。

下图来自 OPC 10000-4 规范：服务，它显示了客户端收到由于"源服务器"的 ServiceLevel 值较低而切换到"备用服务器"的信号时所经历的步骤。



ServiceLevel

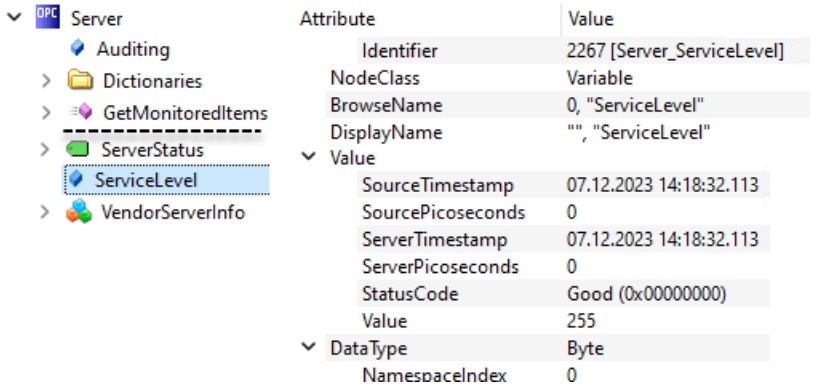
ServiceLevel 是一个变量，它被建模为服务器地址空间中 ServerType 对象的属性。例如，可以使用 ServiceLevel 来找出所寻址的服务器是否仍在提供数据。ServiceLevel 是服务器"健康状态"的数值，因此可为客户端提供切换服务器的触发器。

OPC 10000-5: Information Model 规范的"ServerType"部分介绍了 ServerType 对象的结构。

对于 S7-1500R/H，以下 ServiceLevel 值适用于相应的 OPC UA 服务器：

- **RUN-Redundant**：主 CPU ServiceLevel = 255（CPU 处于 RUN 状态），备用 CPU ServiceLevel = 227（CPU 处于 RUN 状态）。
- **CPU 处于 STOP 状态**：故障 CPU 的 ServiceLevel（RUN STOP 转换）= 1 (NoData)。这是客户端进行故障转移的触发器。

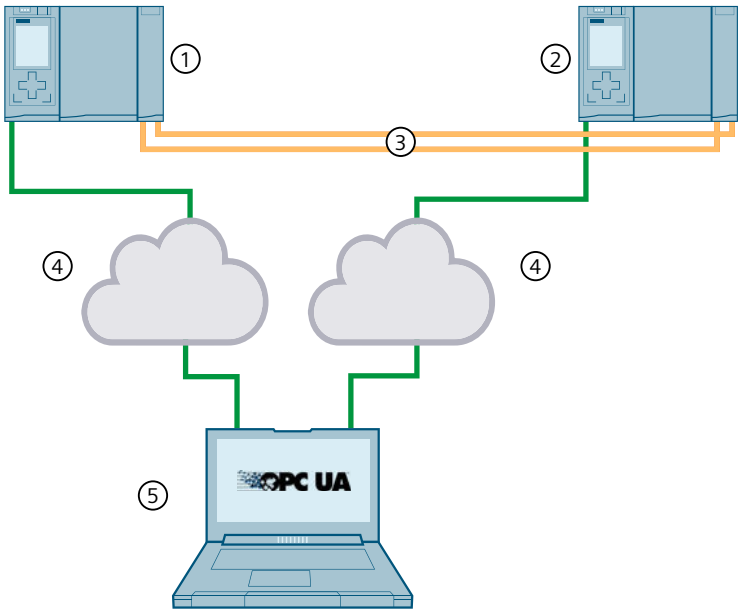
主 CPU 处于 RUN 状态的示例：



非透明冗余的应用

如果客户端通过不同的（即独立）子网访问 OPC UA 服务器，则无法使用系统 IP 地址实现透明冗余。

下图显示了通过两个子网连接到客户端的 H 系统的组态。

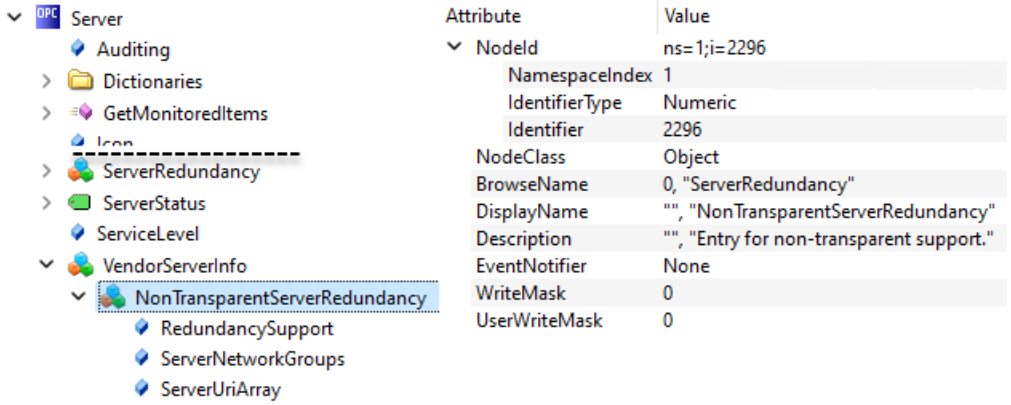


客户端可通过两个独立的网络连接到 H 系统。两个 H CPU 未通过其中一个网络连接。这样，网络的故障是可以接受的。

非透明模式不需要独立子网。还可以使用非透明模式对子网进行组态。

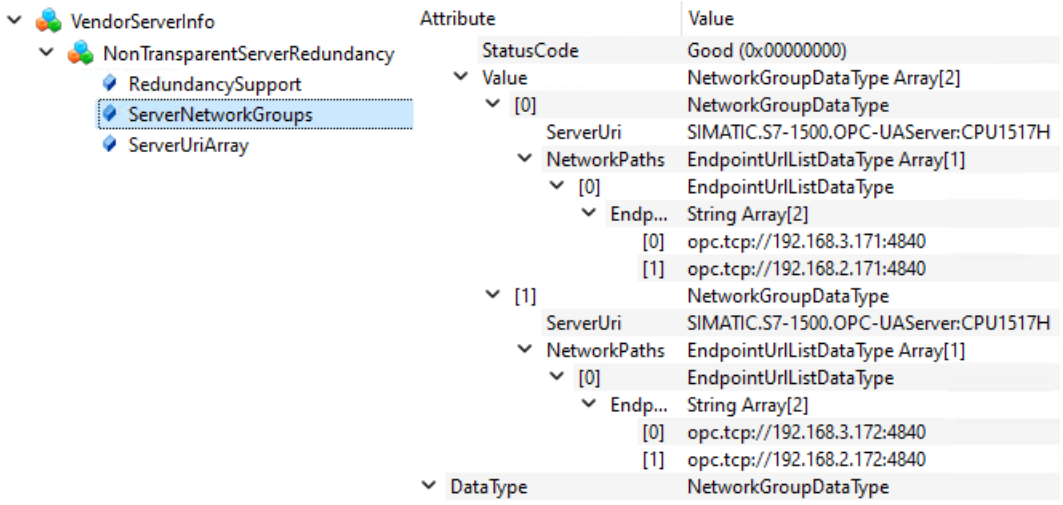
信息模型中用于非透明冗余的相关节点

R/H 站信息模型中与非透明模式相关的信息位于"VendorServerInfo"对象下，该对象是一个占位符对象，用于存储有关 OPC UA 服务器的供应商特定信息。此供应商特定的节点及其下面的节点位于本地服务器的命名空间中
（例如 urn: SIMATIC.S7-1500.OPC-UA.Application:PLC_1，命名空间索引为 1）。



在下面的 ServerRedundancy 对象中，客户端可以找到连接到 R/H 系统所需的所有信息。

ServerUriArray-Property 允许客户端选择合适的服务器。可通过对象类型 NonTranparentNetworkRedundancyType （NonTransparentRedundancyType 的一个子类型）找到有关可以使用哪些网络路径访问服务器的信息。对象类型 NonTranparentNetworkRedundancyType 引用了 ServerNetworkGroups 变量，其中包含服务器数组 (EndpointUrlList) 以及每个服务器存在哪些冗余网络路径 (NetworkPaths) 的信息。



可以在 OPC UA 规范 (OPC 10000-5: Information Model) 中的 VendorServerInfo 节点下方找到相关节点及其引用的详细说明。

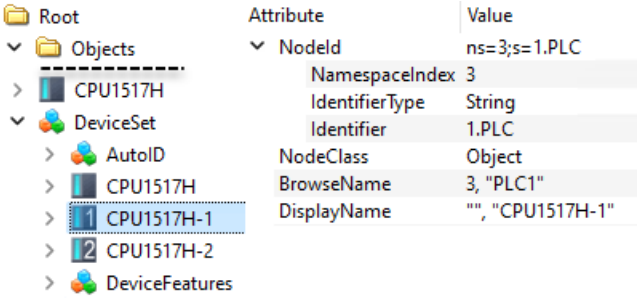
对于 S7-1500R/H 系统，EndpointUrlList 包含每个设备 IP 地址或虚拟设备 IP 地址（如果与 CP 1543-1 一起组态）的条目，可通过该条目访问 S7-1500R/H 系统的服务器。发生故障时，客户端可以决定是否通过不同的路径连接到同一服务器或是否选择冗余服务器。

16.9.4 信息模型详细信息

以下部分详细介绍了 S7-1500R/H 系统的信息模型结构。该信息模型适用于像标准 CPU 一样访问 R/H 系统的客户端和希望对 R/H 系统具有完全控制权的客户端。

冗余系统的高级信息模型

下图举例说明了如何在 DeviceSet 节点中将 R/H 系统建模为各个 R/H CPU 旁边的单独对象。



R/H 系统作为信息模型中的单独节点

在 DeviceSet 节点下，除了每个 R/H CPU 的节点外，还有一个代表 R/H 系统的节点。

OPC UA 服务器在 SYNCUP 系统状态（主 CPU 的 RUN-Syncup 模式）下暂时不可用。服务器在此状态下重新启动。

节点代表以下信息：

节点 <组态名称>	内容	RUN-Redundant 系统状态下的可访问数据	RUN-Solo 系统状态下的可访问数据
<R/H 系统>	有关 R/H 系统的信息	服务器接口数据（两个 CPU 相同） 主 CPU 的 CPU 特定信息	客户端连接到： • 主 CPU：主 CPU 的服务器接口数据。 • 备用 CPU：服务器接口数据（如果 ServiceLevel = NoData：服务器接口数据无效或不可用）。
<PLC 1>	关于冗余 ID 为 1 的 R/H CPU 的信息	服务器接口数据（两个 CPU 相同） 关于冗余 ID 为 1 的 CPU 的 CPU 特定信息	客户端连接到： • PLC 1（= 主 CPU）：主 CPU 的服务器接口数据。 • PLC 1（= 备用 CPU）：服务器接口数据（ServiceLevel = NoData 表示数据无效或不可用）。 • PLC 2（= 备用 CPU）：无数据（访问错误 BadInvalidState / (BadResourceUnavailable)）
<PLC 2>	关于冗余 ID 为 2 的 R/H CPU 的信息	服务器接口数据（两个 CPU 相同） 关于冗余 ID 为 2 的 CPU 的 CPU 特定信息	客户端连接到： • PLC 2（= 主 CPU）：主 CPU 的服务器接口数据。 • PLC 2（= 备用 CPU）：服务器接口数据（ServiceLevel = NoData 表示数据无效或不可用）。 • PLC 1（= 备用 CPU）：无数据（访问错误 BadInvalidState / (BadResourceUnavailable)）

设备节点下方的节点

设备节点为 DeviceSet 节点下面的节点（R/H 系统、PLC 1、PLC 2）。

特别关注“OperatingMode”和“RedundancyMode”属性。

工作模式

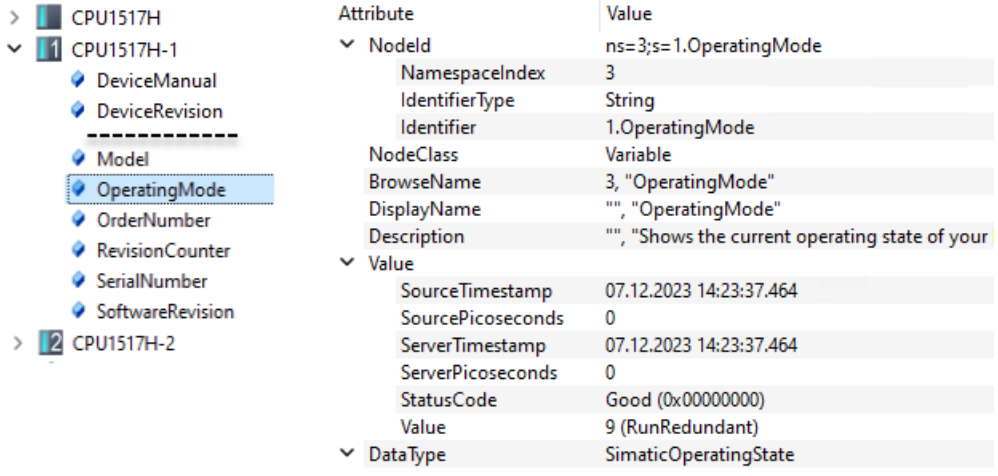
R/H 系统节点下方的 OperatingMode-Property 显示 R/H 系统节点的主 CPU 的操作状态。

对于 R/H 系统的各个 CPU 以及 R/H 系统本身，定义了以下值：

OperatingMode	R/H 系统	R/H CPU
1（STOP - 固件更新）	✓	✓
3（STOP - 初始化）	✓	✓
4 (STOP)	✓	✓
6（启动）	✓	✓

OperatingMode	R/H 系统	R/H CPU
8 (RUN)	✓	✓
9（冗余运行）	✓	✓
10 (HOLD)	✓	✓
13（故障）	✓	✓
14（故障排除）	✓	✓
15（无电源）	✓	✓
17（STOP - 无断电/输出替代值开关）	✓	✓
18（RUN - 断电/输出替代值开关）	✓	✓
19（程序测试）	✓	✓
20（运行程序测试）	8 (RUN)	x（主；测试模式下为备用）
21 (RUN Syncup)	8 (RUN)	✓
22 (Syncup)	8 (RUN)	✓
31（远程未知）	✓	-

OperatingMode 节点属性示例：



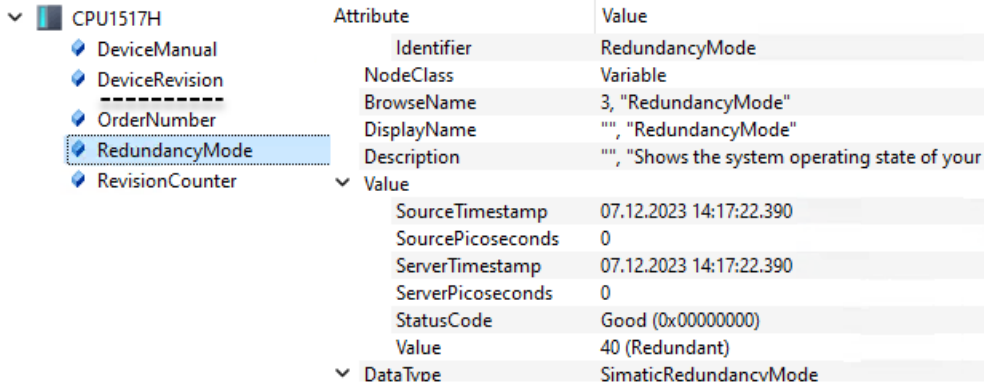
RedundancyMode

RedundancyMode-Property 显示 R/H 系统的系统状态，并且仅在 R/H 系统节点中可用。

RedundancyMode	R/H 系统	R/H CPU
32（无电源）	✓	-
33 (STOP)	✓	-
34 (HOLD)	✓	-
35（启动）	✓	-
37（单独运行）	✓	-

RedundancyMode	R/H 系统	R/H CPU
38 (Syncup)	✓	-
39（程序测试）	✓	-
40（冗余）	✓	-

RedundancyMode 节点属性示例：



16.9.5 更新了服务器指令的说明

要在 S7-1500 CPU 中实现服务器方法，请使用“OPC_UA_ServerMethodPre”和“OPC_UA_ServerMethodPost”指令。这两个指令的新版本 V1.1 的详细信息解释如下。

- 对于自固件版本 V3.1 起的 S7-1500R/H CPU，必须使用服务器指令版本 V1.1 - 不允许使用版本 V1.0。
- 该建议适用于 S7-1500 标准 CPU：对于自 TIA Portal V19 和 CPU 固件版本 V3.1 起的新项目，请始终使用新服务器指令版本 V1.1 或更高版本。
- 在项目中，可以同时使用版本 V1.0 的指令和版本 V1.1 的指令。一个 CPU 用户程序中只能使用一个版本。

要在用户程序中实现服务器方法，请使用此处发布的示例程序，而不是 TIA Portal 帮助系统中的示例。

在用户程序中实现服务器方法

首先简要介绍通过用户程序向 S7-1500 CPU 的 OPC UA 服务器提供 OPC UA 方法的步骤。任何 OPC UA 客户端都可以调用此方法。

- 在用户程序中，功能块 (FB) 代表服务器方法。
- 必须循环调用此 FB（例如，通过 OB 1）。
- 在 FB 中，必须首先调用“OPC_UA_ServerMethodPre”指令，以便将客户端提供给调用的方法的输入参数映射到相应的 CPU 变量（输入数据传输）。
- 调用“OPC_UA_ServerMethodPre”指令后，将处理编程的算法（方法本身，包括处理输入参数和准备输出参数）。

- 一旦运行了编程的算法并且计算了输出参数，就必须调用"OPC_UA_ServerMethodPost"指令。"OPC_UA_ServerMethodPost"指令确保方法的有效输出数据可供客户端使用。
- 创建 OPC UA 方法时请遵守命名约定：
  - 调用服务器方法时，函数必须创建为多实例；多实例的名称必须是"OPC_UA_ServerMethodPre_Instance"/"OPC_UA_ServerMethodPost_Instance"，以便在地址空间中创建方法。
  - 在块的静态区域中定义 OPC UA 方法的输入参数时，变量的名称必须是"UAMethod_InParameters"，以便显示方法的输入参数。
  - 在块的静态区域中定义 OPC UA 方法的输出参数时，变量的名称必须是"UAMethod_OutParameters"，以便显示方法的输出参数。

用户程序的顺序

"OPC_UA_ServerMethodPre"和"OPC_UA_ServerMethodPost"两个指令是异步指令，没有可触发相应指令启动的"REQ 输入"。

"OPC_UA_ServerMethodPre"指令由 OPC UA 客户端调用 OPC UA 方法隐式触发（边沿触发）。这意味着，如果客户端在某个循环中未调用方法，而是在下一个循环中调用，则会发生以下情况：

- STATUS 从 0x7000（无调用）变为 0x7001（第一次调用）。BUSY 输出参数设置为"1"。
- 如果数据传输时间超过一个周期，则状态在第一个周期后变为 0x7002（中间调用）。
- 当输入数据传输完成时：
  - DONE 变为"1"。
  - UAMethodCalled 变为"1"。
  - BUSY 复位为"0"。
- "OPC_UA_ServerMethodPre"和"OPC_UA_ServerMethodPost"：
  - 当 OPC_UA_ServerMethodPost 指令的输出数据传输完成后，DONE 变为"1"。这是复位输入参数"UAMethod_Finished"和"UMethod_Result"的条件（请参见示例程序）。
  - 又经过一个周期后，所有输出参数均复位（对应于 REQ 触发指令的下降沿）。
  - 然后再次发布指令的相应实例（最多可同时激活 20 个实例）。

指令版本 V1.0 和 V1.1 之间的兼容性

V1.0 版本与 V1.1 版本在服务器指令方面存在以下差异：

- 复位服务器指令“OPC_UA_ServerMethodPre”的输出参数“UAMethod_Called”：
  - V1.0：如果“OPC_UA_ServerMethodPre”指令的“DONE”和“OPC_UA_ServerMethodPost”指令的“DONE”均复位，则“UAMethod_Called”会自动复位。因此，仅当“OPC_UA_ServerMethodPost”完成时才会复位“UAMethod_Called”。
  - V1.1：当“OPC_UA_ServerMethodPre”指令的输出参数“DONE”复位时，“UAMethod_Called”会自动复位。
- “OPC_UA_ServerMethodPre”和“OPC_UA_ServerMethodPost”实例的发布：
  - V1.0：无需循环调用服务器指令。
  - V1.1：强制循环调用所需的服务器指令。

说明

当 OPC UA 服务器方法从版本 V1.0 切换为版本 V1.1 时，请检查用户程序是否考虑了所述规则和调整后的功能。如有必要，请调整程序。

示例程序

```
FUNCTION_BLOCK "mySERVER_METHOD"
{ S7_Optimized_Access := 'TRUE' }
VERSION :0.1
 VAR_DB_SPECIFIC
 UAMethod_InParameters :Struct
 IN_BOOL :Bool;
 IN_INT :Int;
 END_STRUCT;
 UAMethod_OutParameters :Struct
 OUT_BOOL :Bool;
 OUT_INT :Int;
 END_STRUCT;
 END_VAR
 VAR
OPC_UA_ServerMethodPre_Instance {InstructionName :=
'OPC_UA_ServerMethodPre'; LibVersion := '1.1'}
:OPC_UA_ServerMethodPre;
 DONE_PRE { S7_SetPoint := 'True'} :Bool;
 BUSY_PRE :Bool;
 ERROR_PRE :Bool;
 STATUS_PRE :DWord;
 UAMethod_Called :Bool;
 OPC_UA_ServerMethodPost_Instance {InstructionName :=
'OPC_UA_ServerMethodPost'; LibVersion := '1.1'}
:OPC_UA_ServerMethodPost;
 UAMethod_Result { S7_SetPoint := 'True'} :DWord;
 UAMethod_Finished :Bool;
 DONE_POST { S7_SetPoint := 'True'} :Bool;
 BUSY_POST :Bool;
 ERROR_POST :Bool;
 STATUS_POST :DWord;
 END_VAR
BEGIN
```

```
 #OPC_UA_ServerMethodPre_Instance(Done => #DONE_PRE,
 Busy => #BUSY_PRE,
 Error => #ERROR_PRE,
 Status => #STATUS_PRE,
 UAMethod_Called => #UAMethod_Called,
 UAMethod_InParameters := #UAMethod_InParameters);

//Method is called
IF #UAMethod_Called AND NOT #ERROR_PRE THEN

(* Functionality:
 InParameters are valid
and copied to OutParameters *)

#UAMethod_OutParameters.OUT_BOOL := #UAMethod_InParameters.IN_BOOL;

#UAMethod_OutParameters.OUT_INT := #UAMethod_InParameters.IN_INT;


(* If Method is finished without errors
prepare output parameters of OPC_UA_ServerMethodPost accordingly *)

#UAMethod_Result := 0;
 #UAMethod_Finished := TRUE;
 END_IF;



#OPC_UA_ServerMethodPost_Instance(UAMethod_Result :=
#UAMethod_Result,
 UAMethod_Finished := #UAMethod_Finished,
 Done => #DONE_POST,
 Busy => #BUSY_POST,
 Error => #ERROR_POST,
 Status => #STATUS_POST,
 UAMethod_OutParameters := #UAMethod_OutParameters);

//Reset Input Parameters after OPC_UA_ServerMethodPost is done
IF #DONE_POST OR #ERROR_POST THEN

#UAMethod_Finished := FALSE;
 #UAMethod_Result := 0;

END_IF;



END_FUNCTION_BLOCK
```

使用 CP 1543-1 确保工业以太网安全

17

全方位保护 - 工业以太网安全的任务

通过工业以太网安全，可以对以太网中的单个设备、自动化单元或网段进行保护。此外，还可以通过组合其它不同的安全措施，对数据传输提供如下保护：

- 数据侦听
- 数据操纵
- 未经授权的访问

安全措施

- 防火墙
  - 全状态数据包检测型 IP 防火墙 （第 3 层和第 4 层）
  - 符合 IEEE 802.3 标准的以太网"非 IP"帧的防火墙（第 2 层）
  - 带宽限制
  - 全局防火墙规则

  防火墙将保护 CP 1543-1 内网段中的所有网络节点。例外：如果使用"通过通信模块访问 PLC"功能通过 CP 的接口访问 CPU，则防火墙不会保护此连接。
- 日志记录

  在监视过程中，事件将存储在日志文件中，可通过组态工具进行读取或者自动发送到 Syslog 服务器中。
- HTTPS

  对网站传输进行加密，例如在过程控制期间。
- FTPS（显式模式）

  对文件传输进行加密。
- 安全 NTP

  对时间同步和传输进行保护
- SNMPv3

  对网络分析信息的传输进行安全保护，以防窃听。
- VPN 组
  通过组态，可将 CP 1543-1 及其它安全模块整合到 VPN 组中。在 VPN 组 (VPN) 的所有安全模块之间建立 IPsec 隧道。这些安全模块的所有内部节点可通过此隧道互相进行安全通信。
- 对设备和网段进行保护

  防火墙与 VPN 组的保护功能可应用于单个设备、多个设备或整个网段的操作。

更多信息

有关工业安全的最重要内容的链接，请参见该常见问题与解答 (https://support.industry.siemens.com/cs/cn/zh/view/92651441)。

17.1 防火墙

防火墙的任务

防火墙的目的是保护网络和站点免受外部的影响和干扰。 这意味着，只能与之前指定的通信伙伴进行通信。

可通过 IPv4 地址、IPv4 子网、端口号或 MAC 地址等信息对为数据流进行过滤。

可以为以下协议层组态防火墙功能：

- 全状态数据包检测型 IP 防火墙 （第 3 层和第 4 层）
- 符合 IEEE 802.3 标准的以太网"非 IP"帧的防火墙（第 2 层）

防火墙的规则

在防火墙规则中将介绍允许或禁止传输的数据包以及传输的方向。

17.2 日志记录

功能

安全模块可以通过诊断和日志功能进行测试和监视。

- 诊断功能

  包括各种可在线使用的系统和状态功能。
- 记录功能

  记录系统和安全事件。 根据事件类型的不同，记录的信息将包含在 CP 1543-1 的易失性或非易失性的本地缓冲区中。 此外，也可以存储在网络服务器中。

  只能通过网络连接对这些功能进行参数分配和评估。

通过日志功能记录事件

通过日志设置，指定待记录的事件。 在此，可组态以下记录方式：

- 本地日志记录

  通过这种记录方式，可以将事件记录在 CP 1543-1 的本地缓冲区中。 并通过安全组态工具的在线对话框访问和显示这些记录，并在服务站中进行归档。
- 网络 Syslog

  使用网络 Syslog，可以记录到网络中的 Syslog 服务器上。 使用这种方式时，将根据日志设置中的组态信息对事件进行记录。

17.3 NTP 客户端

功能

要检查证书时间的有效性以及日志条目的时间戳，则需在 CPU 上对 CP 1543-1 中的日期和时间进行维护。 时间可以与 NTP 同步。 CP 1543-1 通过自动化系统的背板总线将同步的时间转发到 CPU。 这样 CPU 还可以在执行程序时接收时间事件的同步时间。

可通过安全型或非安全型 NTP 服务器对时间进行自动设置和定期同步。 最多可为 CP 1543-1 分配 4 个 NTP 服务器。 但不能混合使用非安全型和安全型 NTP 服务器的组态。

17.4 SNMP

功能

与 CPU 类似，CP 1543-1 也可基于简单网络管理协议 (SNMP) 传输管理信息。为此，需在 CP/CPU 上安装一个“SNMP 代理”，用于接收和响应 SNMP 查询。有关具有 SNMP 功能设备的属性信息保存在 MIB（管理信息库）文件中，需要具有相应权限才能访问。

在使用 SNMPv1 安全措施时，还将发送“社区字符串”。“社区字符串”类似于一个密码，与 SNMP 查询一起发送。在“社区字符串”正确时，发送请求的信息。在此字符串不正确时，丢弃该请求。

在使用 SNMPv3 安全措施时，将对数据进行加密传输。为此，需要选择一种认证方法（例如 SHA）或者一种认证和加密方法（例如 AES）。

用户可以激活和取消激活 CP/CPU 的 SNMP 应用。如果网络的安全准则不允许使用 SNMP 或需使用用户自己的 SNMP 解决方案时，则可取消激活 SNMP。

有关如何激活和取消激活 CPU 中 SNMP 功能的信息，请参见“SNMP (页 108)”部分。

17.5 VPN

功能

对于保护内部网络的安全模块，可借助 VPN（虚拟专用网络）隧道，通过非安全外部网络实现安全的数据连接。

该模块采用 IPsec 协议（IPsec 隧道模式）建立隧道。

在 STEP 7 中，可向安全模块分配 VPN 组。在 VPN 组的所有模块之间会自动建立 VPN 隧道。在该过程中，一个项目中的某个模块可能同时属于多个不同的 VPN 组。

词汇表

CA 根证书

→ 另请参见"根证书"

CM

→ 通信模块

CP

→ 通信处理器

CPU

中央处理单元 (**CPU**) - S7 自动化系统的核心模块，带有控制和算术逻辑运算单元、存储器、操作系统以及编程设备的接口。

DP 从站

PROFIBUS 上分布式 I/O 中的从站，采用 PROFIBUS DP 协议且符合 EN 50170 标准的第 3 部分。

→ 另请参见"*DP* 主站"

DP 主站

在 PROFIBUS DP 中，分布式 I/O 中的主站符合 EN 50170 标准的第 3 部分。

→ 另请参见"*DP* 从站"

FETCH/WRITE

使用 TCP/IP、ISO-on-TCP 和 ISO 协议的服务器服务，用于访问 S7 CPU 的系统存储区。可以从 SIMATIC S5 或第三方设备/PC 进行访问（客户端功能）。FETCH：直接读取数据；WRITE：直接写入数据。

Freeport 协议

可任意编程的 ASCII 协议；使用该协议可通过点到点连接进行数据传输。

FTP

文件传输协议 (**FTP**) 是一种网络协议，用于通过 IP 网络进行文件传输。FTP 用于在服务器与客户端间进行文件的上传或下载。FTP 目录可以创建并读取，也可以重命名或删除。

HMI

人机界面 (**HMI**)，用于显示和控制自动化过程的设备。

IE

→ 工业以太网

IM

→ 接口模块

IO 控制器、PROFINET IO 控制器

PROFINET 系统中的中央设备，通常为典型的可编程逻辑控制器或 PC。IO 控制器将建立与 IO 设备的连接，与这些设备进行数据交换，并对系统进行监控。

IO 设备、PROFINET IO 设备

PROFINET 系统中分布式 I/O 内的设备，通过 IO 控制器（例如，分布式 I/O、阀岛、变频器和交换机）进行监控。

IP 地址

用作采用 Internet 协议 (IP) 、PC 网络中唯一地址的二进制数。根据该二进制数，可对这些设备进行唯一寻址和单独访问。使用可分离网络部分或主机部分结构的子网掩码来分析 IPv4 地址。例如，一个 IPv4 地址的文本表示由 4 个十进制数字组成，值范围为 0 到 255。这些十进制数使用句点进行分隔。

IPv4 子网掩码

二进制掩码，用于将 IPv4 地址（二进制数）划分为"网络部分"和"主机部分"。

ISO 协议

以太网中对消息或数据包进行数据传输的通信协议。该协议面向硬件、速度快、支持动态数据长度，ISO 协议适用于大中型数据传输。

ISO-on-TCP 协议

支持 S7 路由功能的通信协议，用于在以太网中对数据包进行数据传输，支持网络寻址，ISO-on-TCP 协议适用于大中型数据传输，并支持动态数据长度。

MAC 地址

所有以太网设备在全球范围内都唯一的设备标识码。MAC 地址由制造商分配，其中 3 字节为供应商 ID，另外 3 字节（以连续数字表示）为设备 ID。

Modbus RTU

远程终端单元 (**R**emote **T**erminal **U**nit)；基于主站/从站架构的开放式串行接口通信协议。

Modbus TCP

传输控制协议 (**T**ransmission **C**ontrol **P**rotocol)；基于主站/从站架构的开放式以太网通信协议。数据以 TCP/IP 数据包的形式传输。

NTP

网络时间协议 (**N**etwork **T**ime **P**rotocol, NTP) 规定了由工业以太网建立的自动化系统中同步时钟的标准。NTP 采用适用于 Internet 的 UDP 传输协议。

OPC UA

OPC Unified **Automation** 协议由 OPC 基金会制定，用于在机器间进行数据通信。

PG

→ 编程设备

PNO

→ PROFIBUS 用户组织

PROFIBUS

过程现场总线 (**Pro**cess **Fi**eld **Bus**) - 欧洲现场总线标准。

PROFIBUS DP

支持 DP 协议且符合 EN 50170 的 PROFIBUS。DP 即为分布式 I/O，可进行快捷实时的周期性数据交换。从用户程序的角度来看，分布式 I/O 与集中式 I/O 的寻址方式完全相同。

PROFIBUS 地址

连接到 PROFIBUS 上的设备的唯一标识符。PROFIBUS 地址通过帧形式发送，用于寻址一个设备。

PROFIBUS 设备

该设备上至少有 1 个 PROFIBUS 接口为电气接口（如 RS-485）或光纤接口（如聚合物光纤）。

PROFIBUS 用户组织

该技术委员会致力于 PROFIBUS 和 PROFINET 标准的定义和开发。

PROFINET

基于组件的开放式工业通信系统，以分布式自动化系统的以太网为基础。这种通信技术由 PROFIBUS 用户组织推出。

PROFINET 接口

模块的 PROFINET 接口具备通信功能（如 CPU、CP），带有 1 个或多个端口。出厂前已为该接口分配有 MAC 地址。接口地址与 IP 地址和设备名称（来自各个组态）一起使用，可确保在网络中唯一识别 PROFINET 设备。该接口可以是电气接口、光学接口或者是无线接口。

PROFINET 设备

始终带有一个 PROFINET 接口（电气、光学或无线）的设备。

PROFINET IO

IO 即为输入/输出，分布式 I/O 可进行快速实时的周期性数据交换。从用户程序的角度来看，分布式 I/O 与集中式 I/O 的寻址方式完全相同。

PROFINET IO 作为 PROFIBUS 和 PROFINET International 基于以太网的自动化标准，它定义了跨厂商的通信、自动化系统和工程组态模式。

借助 PROFINET IO，实现一种允许所有设备随时访问网络的交换技术。因此，通过多个设备的并行数据传输，可以更为高效地使用网络。数据的并行发送和接收通过交换式以太网的全双工操作来实现。

PROFINET IO 基于交换式以太网的全双工操作，其带宽为 100 Mbps。

PtP

点到点 (**PtP**)，是两个（只能是两个）通信伙伴间进行双向数据交换的接口和/或传输协议。

RS232、RS422 和 RS485

串行接口标准。

RTU

Modbus RTU（RTU：远程终端设备 (**RTU**)，用于传输二进制格式的数据，具有较高的数据吞吐率。在对数据进行评估之前，必须将其转换为一种可读取的格式。

S7 路由

通过作为 S7 路由器的一个或多个网络节点，可以在不同 S7 子网中的 S7 自动化系统、S7 应用或 PC 站之间进行通信。

SDA 服务

发送数据，需要进行确认。SDA 是一种基本服务。通过这种该服务，发起方（例如，DP 主站）可以向其它设备发送一条消息，并在之后接收从接收方发回的确认消息。

SDN 服务

发送数据，无需进行确认。该服务主要用于向多个站发送数据，但无需进行确认。该服务适用于发送同步任务和状态消息。

SNMP

简单网络管理协议 (**SNMP**)，使用无线 UDP 传输协议。SNMP 的工作模式与客户端/服务器的非常类似。SNMP 管理器对网络节点进行监视。SNMP 代理收集各网络节点中的各种网络特定信息，并以一种以结构化的形式将这种信息存储在管理信息库 (**M**anagement **I**nformation **B**ase) 中。网络管理系统可以使用该信息进行详细的网络诊断。

TCP/IP

传输控制协议/因特网协议 (**TCP** / **IP**)，一种面向连接的网络协议，通常作为异构网络中数据传输的标准。

UDP

用户数据报协议 (**UDP**)，适用于快速简单数据传输的通信协议，无需进行确认。TCP/IP 中未定义错误检查机制。

USS

通用串行接口协议 (**USS**)，根据主站/从站原理定义了一种通过一根串行总线进行通信的访问方式。

Web 服务器

通过因特网进行数据交换的软件/通信服务。Web 服务器通过标准传输协议（HTTP、HTTPS）将文档传输到 Web 浏览器。文档可以是静态文档，也可以是由 Web 服务器根据 Web 浏览器的请求从不同的数据源动态生成的文档。

备用 CPU

如果 R/H 系统为 RUN-Redundant 系统状态，则主 CPU 将对过程进行控制。备用 CPU 将同步处理用户程序，并在主 CPU 发生故障时接管过程控制。

编程设备

编程设备实质上是一种适用于工业应用的紧凑型便携式 PC。它们通过用于可编程逻辑控制器的特定硬件和软件组态进行识别。

操作系统

使用和操作计算机的软件。操作系统将对诸如内存、输入和输出设备等资源进行管理，并控制程序的执行。

操作状态

操作状态是指在特定时间点某个单 CPU 的操作特性。

SIMATIC 标准系统的 CPU 具有三种操作状态：STOP、STARTUP 和 RUN。

S7-1500R/H 冗余系统的主 CPU 则具有 STOP、STARTUP、RUN、RUN-Syncup 和 RUN-Redundant 五种操作状态。备用 CPU 具有 STOP、SYNCUP 和 RUN-Redundant 三种操作状态。

从站

现场总线系统中的分布式设备，只有在主站提出请求后才能与主站进行数据交换。

→ 另请参见*"DP 从站"*

点到点连接

通过通信模块上的串行接口，在两个通信伙伴（只能是两个）间进行双向数据交换。

端口

用于将设备连接到 PROFINET 的物理连接器。PROFINET 接口具有一个或多个端口。

服务器

可提供某些特定服务的设备或（广义上的）对象，并根据客户端的请求执行这些服务。

根证书

该证书为一个证书颁发机构的证书：它使用其私钥对最终实体证书和中间 CA 证书进行签名。

这种证书的“主体”(Subject) 与“颁发者”(Issuer) 属性必须相同。该证书颁发机构对自己的证书进行签名。

字段“CA”必须设置为“True”。

TIA Portal V14 包含一个这样的 CA 根证书：

如果要在 TIA Portal 中组态 S7-1500 的 OPC UA 服务器，则 TIA Portal 将为该 OPC UA 服务器生成一个最终实体证书并使用私钥对该证书进行签名。

该最终实体证书的签名可通过 TIA Portal 的公钥进行验证。该公钥位于 TIA Portal 的 CA 根证书中。

工业以太网

在工业环境中以太网络的构建指南。它与标准以太网的最大区别在于各组件的机械性能的鲁棒性和抗干扰性。

过程映像 (I/O)

CPU 将输入和输出模块中的值传送到该存储区域内。循环程序开始时，CPU 将过程映像输出作为信号状态传送到输出模块中。CPU 随后将输入模块的信号状态读取到过程映像输入中。随后 CPU 执行用户程序。

环形拓扑

网络中的所有设备彼此连接形成一个环。

交换机

用于连接局域网 (LAN) 中多个终端或网段的网路组件。

接口模块

分布式 I/O 系统中的模块。接口模块通过现场总线将分布式 I/O 系统连接到 CPU（IO 控制器/DP 主站）并提供用于 I/O 模块的数据。

客户端

一种网络设备，需要网络中的其它设备（服务器）将为其提供服务。

路由器

具有唯一标识符（名称和地址）的网络节点，用于将子网连接在一起，以便将数据传输到网络中唯一标识的网络通信节点。

切换通信

除了各 CPU 的设备 IP 地址之外，S7-1500R/H 冗余系统还支持以下系统 IP 地址：

- 两个 CPU 上 PROFINET 接口 X1 的系统 IP 地址（系统 IP 地址 X1）
- 两个 CPU 上 PROFINET 接口 X2 的系统 IP 地址（系统 IP 地址 X2）

通过系统 IP 地址，可与其它设备（如，HMI 设备、CPU、PG/PC）进行通信。这些设备通常基于系统 IP 地址与冗余系统的主 CPU 进行数据通信。这样，可确保在冗余操作中原来的主 CPU 发生故障后，通信伙伴可在 RUN-Solo 系统状态下与新的主 CPU（之前的备用 CPU）进行数据通信。

冗余系统

冗余系统具有多个（冗余）重要自动化组件实例。如果冗余组件发生故障，过程控制仍将保持。

设备

通用术语，适用于：

- 自动化系统（例如 PLC、PC）
- 分布式 I/O 系统
- 现场设备（例如 PLC、PC、液压设备、气动设备）以及
- 有源网络组件（例如交换机、路由器）
- PROFIBUS 的网关、AS Interface 或其它现场总线系统

设备证书

此类证书由证书颁发机构 (CA) 签名。

最终实体证书的签名则使用证书颁发机构的公钥进行验证。

“主体”(Subject) 属性不得与“颁发者”(Issuer) 属性相同。

例如，“主体”(Subject) 包含 OPC UA 应用程序证书等的程序名称。

“颁发者”(Issuer) 则是对证书进行签名的证书颁发机构。

字段“CA”必须设置为“False”。

时间同步

可以将一个单源标准系统时间传送给系统中的所有设备，以便这些设备可根据该标准时间设置自己的时钟。

树形拓扑结构

具有分支结构的一种网络拓扑结构：每个总线节点上连接有 2 或更多个总线节点。

双工

数据传输系统，分为全双工和半双工。

半双工：使用一个通道交替地进行数据交换（可交替地发送或接收数据，但不能同时进行）。

全双工：使用 2 个通道同时双向地进行数据交换（可双向同时发送和接收数据）。

双绞线

使用双绞线电缆连接的快速以太网基于 IEEE 802.3u 标准 (100 Base-TX)。传输介质是阻抗为 100 欧姆的 2x2 屏蔽双绞线电缆 (22 AWG)。这种电缆的传输特性必须符合类别 5 的要求。

终端设备与网络组件之间的最大连接长度不得超过 100 m。带有 RJ-45 连接插头的连接器基于 100Base-TX 标准而设计。

通信处理器

执行其它通信任务的模块，可实现诸如区域安全之类的特殊应用。

通信模块

自动化系统中执行通信任务的模块，作为 CPU（例如 PROFIBUS）的接口扩展并具有附加通信功能 (PtP)。

网络

网络由 1 或多个相互连接的子网组成，可以包含任意数量的设备。各个网络可以彼此独立共存。

系统状态

S7-1500R/H 冗余系统的系统状态取决于主 CPU 和备用 CPU 的操作状态。术语"系统状态"用于快速标识两个 CPU 上同时出现操作状态。S7-1500R/H 冗余系统具有 STOP、STARTUP、RUN-Solo、SYNCUP 和 RUN-Redundant 五种系统状态。

现场设备

→ 设备

线性总线形拓扑结构

一种网络拓扑结构，各种设备都连接到一个总线上。

协议

有关在两个或更多通信伙伴之间进行通信所遵循的规则的协议。

信息安全

为防止以下各项丢失而采取的所有措施的统称

- 对数据进行未经授权的访问而导致机密性缺失
- 因数据操作而导致的完整性缺失
- 因数据破坏而导致的可用性缺失

一致性数据

这些数据属于一个整体，在传输时不能分开。

以太网

基于帧的局域网 (LAN)，采用国际标准技术。其中对电缆类型、物理层的信号发送、数据包的格式以及介质访问控制的协议进行定义。

以太网适配器

电子线路器件，用于将计算机接入以太网。以便在以太网中进行数据交换/通信。

用户程序

在 SIMATIC 中，对 CPU 操作系统和用户程序做了区分。用户程序中包含用于控制一个系统或过程的所有指令、声明和数据。用户程序可分配给可编程模块（例如，CPU 和 FM），并可由更小的单元构成。

指令

用户程序中最小的独立单元，并根据结构、功能或目的分为单个的用户程序部分。一条指令可代表处理器重的一个操作过程。

中间 CA 证书

此为证书颁发机构的证书，使用根证书颁发机构的私钥进行签名。

中间证书颁发机构使用自己的私钥对最终实体证书进行签名。

最终实体证书的签名则使用中间证书颁发机构的公钥进行验证。

中间 CA 证书的"主体"(Subject) 与"颁发者"(Issuer) 属性不得相同：该证书颁发机构毕竟不对自己的证书进行签名。

字段"CA"必须设置为"True"。

主 CPU

如果 R/H 系统为 RUN-Redundant 系统状态，则主 CPU 将对过程进行控制。备用 CPU 将同步处理用户程序，并在主 CPU 发生故障时接管过程控制。

主站

通信/PROFIBUS 子网的上一层主动参与者。主站有权访问总线（令牌），并能请求和发送数据。

→ 另请参见“*DP* 主站”

子网

网络的一部分，子网的参数必须与设备（例如，PROFINET 中的）相匹配。子网中包含总线组件和所有连接的站。通过网关或路由器，子网可以进行互连从而构成一个网络。

自动化系统

可编程逻辑控制器，用于对过程工程组态行业和制造技术的过程链进行开环和闭环控制。自动化系统可包含各种组件和集成系统功能，具体取决于自动化任务。

自签名证书

对于这些证书，用户使用自己的私钥对其进行签名，并作为最终实体证书。

最终实体证书的签名通过用户的公钥进行验证。

自签名证书的“主体”(Subject) 与“颁发者”(Issuer) 属性必须相同：用户已完成对自己证书的签名。

字段“CA”必须设置为“False”。

例如，用户可将自签名证书用作 OPC UA 客户端的应用程序证书。

有关基于 OPC 基金会证书生成器生成自签名证书的具体步骤，请参见“这里 (页 %getreference)”。

总线

一种传输介质，用于连接多个设备。可通过电缆或光缆以串行或并行方式进行数据传输。

最终实体证书

→ 另请参见“设备证书”

索引

P

R

S

T

U

W

X

安

导

点

通

系

写

信

证

中

自

最