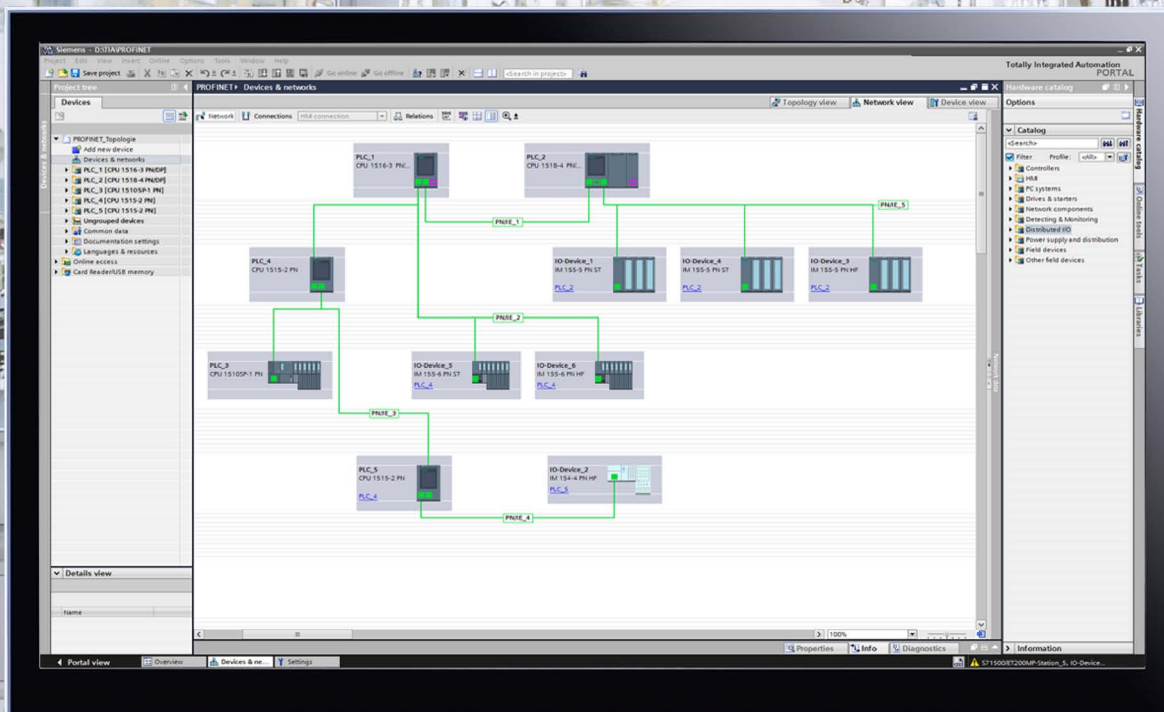


SIEMENS



SIMATIC

S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro

通信

功能手册

版本

09/2016

siemens.com

SIMATIC

S7-1500、ET 200MP、 ET 200SP、ET 200AL、ET 200pro 通信

功能手册

前言

文档指南

1

产品概述

2

通信服务

3

PG 通信

4

HMI 通信

5

开放式用户通信

6

S7 通信

7

点到点连接

8

OPC UA 服务器

9

路由

10

连接资源

11

连接诊断

12

工业以太网安全

13

法律资讯

警告提示系统

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

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

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

合格的专业人员

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

按规定使用 Siemens 产品

请注意下列说明：

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

商标

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

责任免除

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

前言

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 和 ET 200pro 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
- 通信服务的特性
- 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。

与之前版本相比，该功能手册（12/2014 版）的新增内容

新增内容		客户收益	信息出处
新增内容	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 服务器 (页 131)”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 74)”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 STEP 7 管理证书 (页 84)”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。	“禁用 SNMP (页 37)”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”。

“S7-1500 CPU”是指 CPU 系列 S7-1500F、S7-1500T、S7-1500C 及 S7-1500pro CPU、ET200SP CPU 和 SIMATIC S7-1500 软件控制器。

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。

参见

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

目录 (<http://mall.industry.siemens.com>)

安全信息

Siemens 为其产品及解决方案提供了工业安全功能，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安全保护机制。Siemens 的产品和解决方案仅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。只有在必要时并采取适当安全措施（例如，使用防火墙和网络分段）的情况下，才能将系统、机器和组件连接到企业网络或 Internet。

此外，应考虑遵循 Siemens 有关相应安全措施的指南。更多有关工业安全的信息，请访问 (<http://www.siemens.com/industrialsecurity>)。

Siemens 不断对产品和解决方案进行开发和完善以提高安全性。Siemens 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 Siemens 工业安全 RSS 源，网址为 (<http://www.siemens.com/industrialsecurity>)。

Siemens 工业在线支持

在此处可轻松快速地获取以下主题的最新信息：

- **产品支持**

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

- **应用示例**

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

- **服务**

介绍了行业服务、现场服务、技术支持、备件和培训提供情况的相关信息。

- **论坛**

提供了自动化技术相关的答疑和解决方案。

- **我的技术支持**

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

由 Internet (<http://www.siemens.com/automation/service&support>) 上的西门子工业在线支持提供这部分信息。

网上商城

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

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

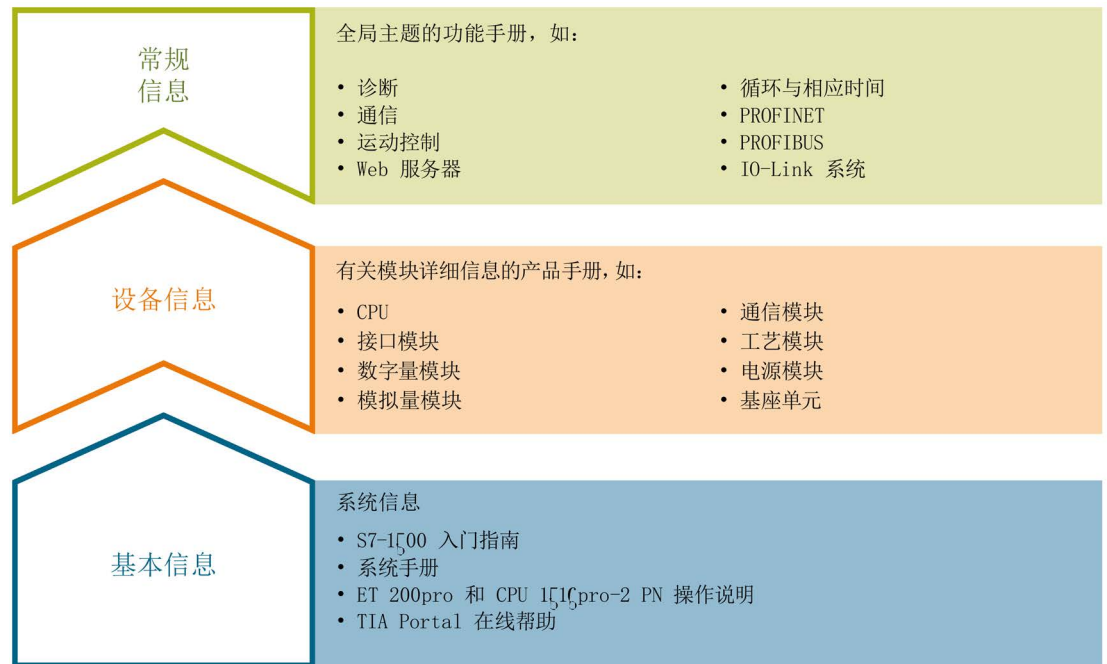
目录

- 前言4
- 1 文档指南 11
- 2 产品概述 15
- 3 通信服务 20
 - 3.1 通信方式概述 20
 - 3.2 连接资源概览 24
 - 3.3 建立连接..... 25
 - 3.4 数据的一致性 28
 - 3.5 以太网通信的通信协议和端口号 31
 - 3.6 SNMP 37
 - 3.6.1 禁用 SNMP 37
 - 3.6.2 示例：禁用 CPU 1516-3 PN/DP 的 SNMP 功能 38
- 4 PG 通信 41
- 5 HMI 通信 44
- 6 开放式用户通信 46
 - 6.1 开放式用户通信概述 46
 - 6.2 开放式用户通信协议 47
 - 6.3 开放式用户通信的指令 49
 - 6.4 通过域名进行寻址的开放式用户通信 53
 - 6.5 通过 TCP、ISO-on-TCP、UDP 和 ISO 建立开放式用户通信 56
 - 6.6 建立 FDL 通信 63
 - 6.7 建立与 Modbus TCP 的通信 66
 - 6.8 通过电子邮件建立通信 69
 - 6.9 通过 FTP 建立通信 70
 - 6.10 建立和终止通信关系 73
 - 6.11 开放式用户安全通信 74
 - 6.11.1 基本知识..... 74
 - 6.11.1.1 安全通信的基础知识 74
 - 6.11.1.2 通过加密确保数据机密 77
 - 6.11.1.3 通过签名确保数据的真实性和完整性 80

6.11.1.4	使用 STEP 7 管理证书.....	84
6.11.1.5	证书管理示例。.....	87
6.11.1.6	示例：基于 TLS 的 HTTP.....	93
6.11.2	S7-1500 CPU（作为 TLS 客户端）与外部 PLC（TLS 服务器）之间的安全 OUC.....	97
6.11.3	S7-1500 CPU（作为 TLS 服务器）与外部 PLC（TLS 客户端）之间的安全 OUC.....	100
6.11.4	两个 S7-1500 CPU 之间的安全 OUC.....	103
6.11.5	通过 CP 接口进行安全 OUC 连接.....	107
6.11.6	通过电子邮件实现 OUC.....	113
7	S7 通信.....	116
8	点到点连接.....	126
9	OPC UA 服务器.....	131
9.1	需了解的 OPC UA 知识.....	131
9.1.1	OPC UA 和工业 4.0.....	131
9.1.2	说明的结构.....	132
9.1.3	OPC UA 的特性.....	133
9.1.4	从标准 OPC 接口到 OPC UA.....	134
9.1.5	CPU S7-1500 的 OPC UA 服务器.....	135
9.1.6	OPC UA 服务器的地址空间.....	136
9.1.7	OPC UA 服务器的端点.....	137
9.1.8	节点寻址.....	140
9.1.9	OPC UA 客户端.....	143
9.2	OPC UA 的安全性.....	146
9.2.1	信息安全.....	146
9.2.2	请勿禁用安全策略.....	147
9.2.3	ITU X.509 证书.....	148
9.2.4	OPC UA 证书.....	151
9.2.5	创建自签名证书.....	152
9.2.6	用户自己生成 PKI 密钥对和证书.....	154
9.2.7	安全地传输消息.....	158
9.3	组态对 PLC 变量的访问.....	161
9.3.1	管理读写权限.....	161
9.3.2	更快速地访问数组和结构.....	163
9.3.3	导出带有已启用 PLC 变量的 XML 文件.....	164
9.4	组态 S7-1500 的 OPC UA 服务器.....	165
9.4.1	启用 OPC UA 服务器.....	165
9.4.2	访问 OPC UA 服务器.....	167
9.4.3	OPC UA 服务器的设置.....	168
9.4.4	通过 STEP 7 生成服务器证书.....	171
9.4.5	处理服务器和客户端证书.....	175
9.4.6	用户认证.....	181
9.4.7	OPC UA 服务器的许可证.....	182

9.5	对 OPC UA 客户端进行编程	183
9.5.1	采用 C# 编写的 OPC UA 客户端	183
9.5.2	示例程序的要求	183
9.5.3	示例程序的结构	184
9.5.4	建立与 OPC UA 服务器的连接	184
9.5.4.1	本章节概述	184
9.5.4.2	以不同的安全设置访问服务器	185
9.5.4.3	建立与服务器端点的连接	186
9.5.5	使用与服务器端点的现有连接	188
9.5.5.1	读取 PLC 变量和 DB 变量	188
9.5.6	断开与 OPC UA 服务器的连接	191
9.5.6.1	断开与服务器的连接	191
9.5.7	生成客户端证书	192
9.5.7.1	客户端实例证书	192
9.5.8	CPU 用户程序	193
9.5.8.1	用户程序的结构 - Main [OB1]	193
9.5.8.2	启用 PLC 变量以通过 OPC UA 进行访问	194
10	路由	195
10.1	S7 路由	195
10.2	数据记录路由	200
11	连接资源	202
11.1	站中的连接资源	202
11.2	连接资源分配	206
11.3	连接资源的显示	210
12	连接诊断	214
13	工业以太网安全	218
13.1	防火墙	219
13.2	日志记录	220
13.3	NTP 客户端	220
13.4	SNMP	221
13.5	VPN	221
	术语表	222
	索引	235

SIMATIC S7-1500 自动化系统、基于 SIMATIC S7-1500 的 CPU 1516pro-2 PN 和分布式 I/O 系统 SIMATIC ET 200MP、ET 200SP 与 ET 200AL 的文档分为 3 个部分。这样，用户可以根据具体需求快速访问自己所需的特定信息。



基本信息

在系统手册和入门指南中，对 SIMATIC S7-1500、ET 200MP、ET 200SP 和 ET 200AL 系统的组态、安装、接线和调试进行了详细介绍。对于 CPU 1516pro-2 PN，可参见相应的操作说明。STEP 7 在线帮助则为用户提供有关组态和编程方面的技术支持。

设备信息

产品手册中包含模块特定信息的简洁描述，如特性、端子图、功能特性、技术数据。

常规信息

功能手册中包含有关常规主题的详细介绍，如诊断、通信、运动控制、Web 服务器、OPC UA 等等。

相关文档，可从 Internet (<http://w3.siemens.com/mcms/industrial-automation-systems-simatic/en/manual-overview/Pages/Default.aspx>) 免费下载。

产品信息数据表中记录了对这些手册的更改和补充。

有关产品信息，敬请访问 Internet:

- S7-1500/ET 200MP (<https://support.industry.siemens.com/cs/cn/zh/view/68052815>)
- ET 200SP (<https://support.industry.siemens.com/cs/cn/zh/view/73021864>)
- ET 200AL (<https://support.industry.siemens.com/cs/cn/zh/view/99494757>)

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 (<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>)

“我的技术支持”

通过“我的技术支持”（我的个人工作区），“工业在线技术支持”的应用将更为方便快捷。

在“我的技术支持”中，用户可以保存过滤器、收藏夹和标签，请求 CAx 数据以及编译“文档”区内的个人数据库。此外，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。

要使用“我的技术支持”中的所有功能，必须先进行注册。

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

“我的技术支持” - 文档

在“我的技术支持”中的“文档”区域，用户可以使用整个手册或部分手册生成自己的手册。也可以将手册导出为 PDF 文件或后期可编辑的其它格式。

有关“我的技术支持” - 文档，敬请访问 Internet
(<http://support.industry.siemens.com/My/ww/zh/documentation>)。

“我的技术支持” - CAx 数据

在“我的技术支持”中的 CAx 数据区域，可以访问 CAx 或 CAe 系统的最新产品数据。

仅需轻击几次，用户即可组态自己的下载包。

在此，用户可选择：

- 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” - CAx 数据，敬请访问 Internet

(<http://support.industry.siemens.com/my/ww/zh/CAxOnline>)。

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/sc/ww/zh/sc/2054>)。

TIA Selection Tool

通过 TIA Selection Tool，用户可选择、组态和订购全集成自动化 (TIA) 中所需设备。

该工具作为 SIMATIC Selection Tool 的新一代产品，在一个工具中完美集成了自动化技术的各种组态程序。

通过 TIA Selection Tool，用户可以根据产品选择或产品组态生成一个完整的订购列表。

TIA Selection Tool 可从 Internet (<http://w3.siemens.com/mcmts/topics/en/simatic/tia-selection-tool>) 上下载。

SIMATIC Automation Tool

通过 SIMATIC Automation Tool，可同时对不同的 SIMATIC S7 站进行系统调试和维护操作，而无需打开 TIA Portal 系统。

SIMATIC Automation Tool 支持以下各种功能：

- 扫描 PROFINET/以太网工厂网络，识别所有连接的 CPU
- 为 CPU 分配地址（IP、子网、网关）和站名称（PROFINET 设备）
- 将日期和已转换为 UTC 时间的 PG/PC 时间传送到模块中
- 将程序下载到 CPU 中
- 切换操作模式 RUN/STOP
- 通过 LED 指示灯闪烁确定 CPU 状态
- 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 和所连模块的固件版本

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

PRONETA

SIEMENS PRONETA（PROFINET 网络分析服务）用于在调试过程中快速分析工厂网络的具体状况。PRONETA 具有以下两个核心功能：

- 拓扑总览功能，分别扫描 PROFINET 和连接的所有组件。
- IO 检查，快速测试工厂接线和模块组态。

SIEMENS PRONETA 可从 Internet (<https://support.industry.siemens.com/cs/cn/zh/view/67460624>) 上下载。

产品概述

基于 S7-1500, , ET 200MPET 200SPET 200pro 和 ET 200AL 系统的 CPU、通信模块和处理器以及 PC 系统, 可通过 PROFINET, PROFIBUS 和点到点的连接进行通信连接。

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。

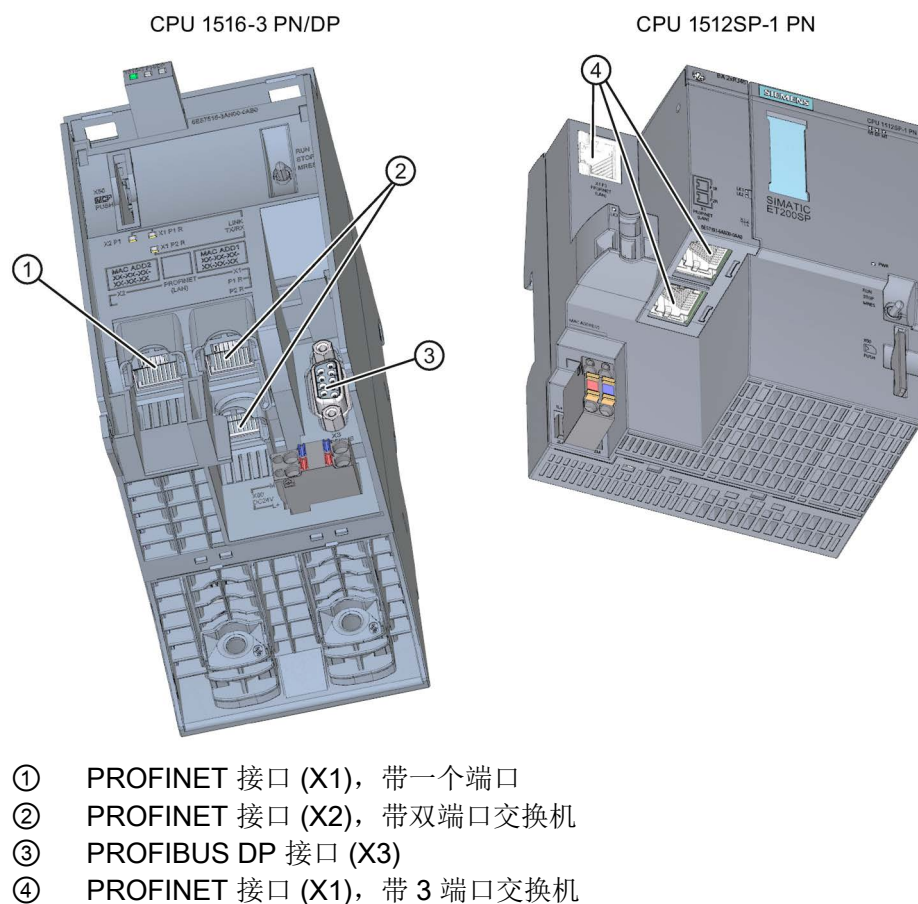


图 2-1 CPU 1516-3 PN/DP 和 CPU 1512SP-1 PN 的接口

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）。

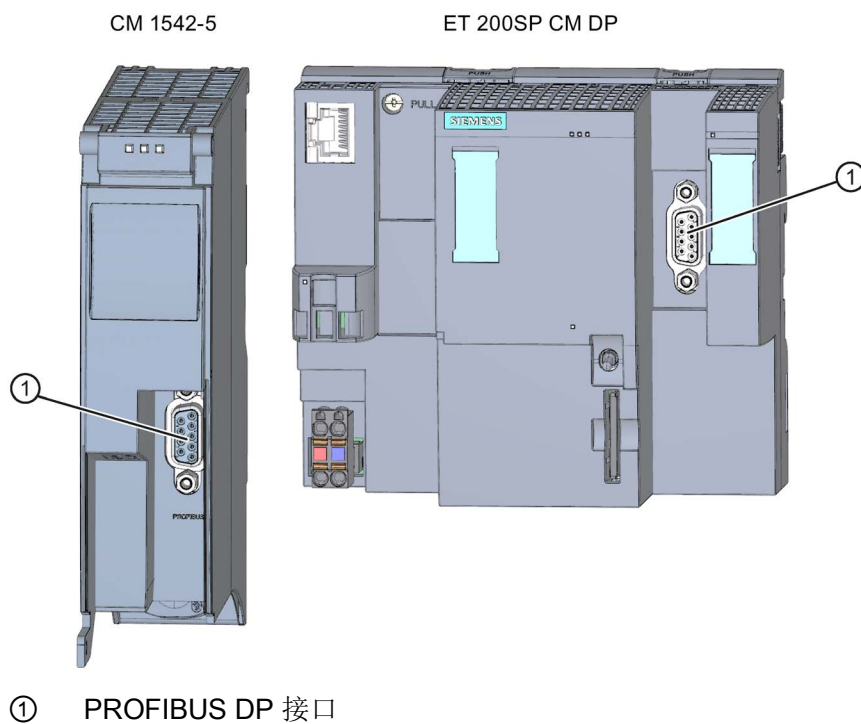
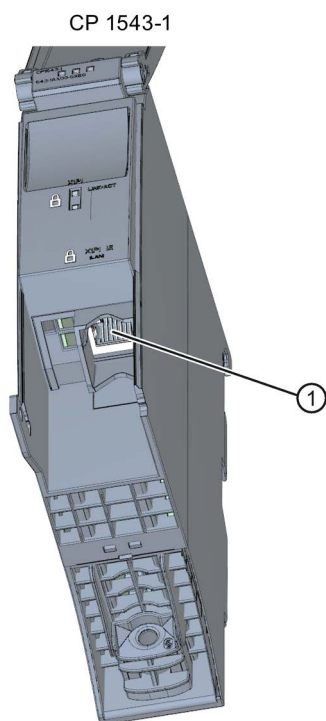


图 2-2 CM 1542-5 和 CM DP 的 PROFIBUS DP 接口

通信处理器的接口

与 CPU 上集成的接口相比，通信处理器 (CP) 接口的功能更多。CP 支持一些特殊功能。例如，CP 1543-1 支持工业以太网安全功能，可通过其工业以太网接口确保工业以太网安全。



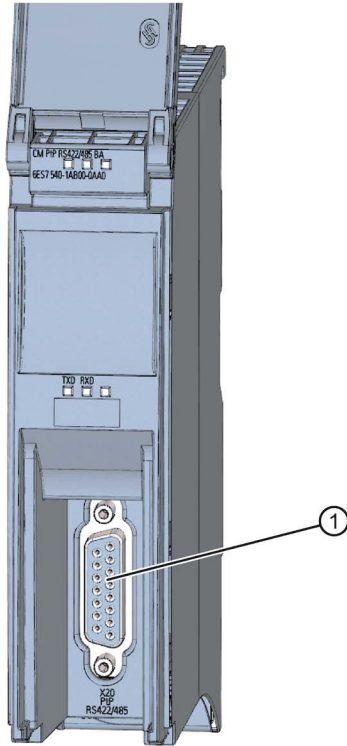
① 工业以太网接口

图 2-3 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

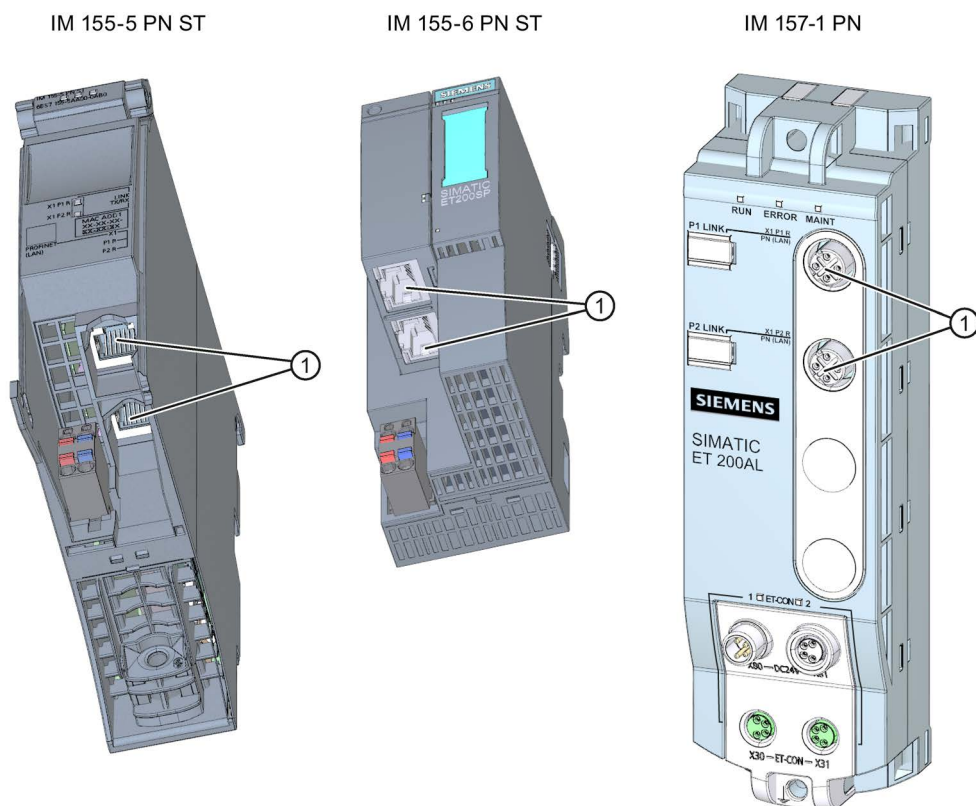


① 点到点连接接口

图 2-4 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 2-5 PROFINET 接口 IM 155-5 PN ST (ET 200MP)、IM 155-6 PN ST (ET 200SP) 和 IM 157-1 PN (ET 200AL)

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

3.1 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 3-1 通信方式

通信方式	功能	接口：		
		PN/IE 1	DP	串行
PG 通信	调试、测试、诊断	√	√	-
HMI 通信	操作员监控	√	√	-
基于 TCP/IP 进行开放式通信	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	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 • TUSEND/TURCV • TCON • T_DISCON	√	-	-

通信方式	功能	接口:		
		PN/IE 1	DP	串行
基于 UDP 进行开放式通信	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/TRCV • 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	√	-	-
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（仅通过 CPU 的内部 PROFINET 接口进行通信）	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	√	-	-

3.1 通信方式概述

通信方式	功能	接口:		
		PN/IE 1	DP	串行
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	客户端/服务器或客户端/客户端的数据交换 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	-	√	-

* IE - 工业以太网

更多信息

- 应用示例：与 SIMATIC 控制器进行 CPU-CPU 通信（概要）
有关该应用示例，敬请访问 Internet
(<https://support.industry.siemens.com/cs/cn/zh/view/20982954>)。
- 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 (<http://support.automation.siemens.com/WW/view/de/59057093/0/zh>)》。
- 有关 Web 服务器的功能介绍，请参见功能手册《Web 服务器 (<http://support.automation.siemens.com/WW/view/zh/59193560>)》。
- 有关 SNMP 标准协议的信息，敬请访问 Internet
(<http://support.automation.siemens.com/WW/view/zh/15166742>) 中的服务与支持页面。
- 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>)”。

3.2 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、S7 通信以及开放式通信提供相应数量的资源。

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 25)”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 202)”部分。

3.3 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。

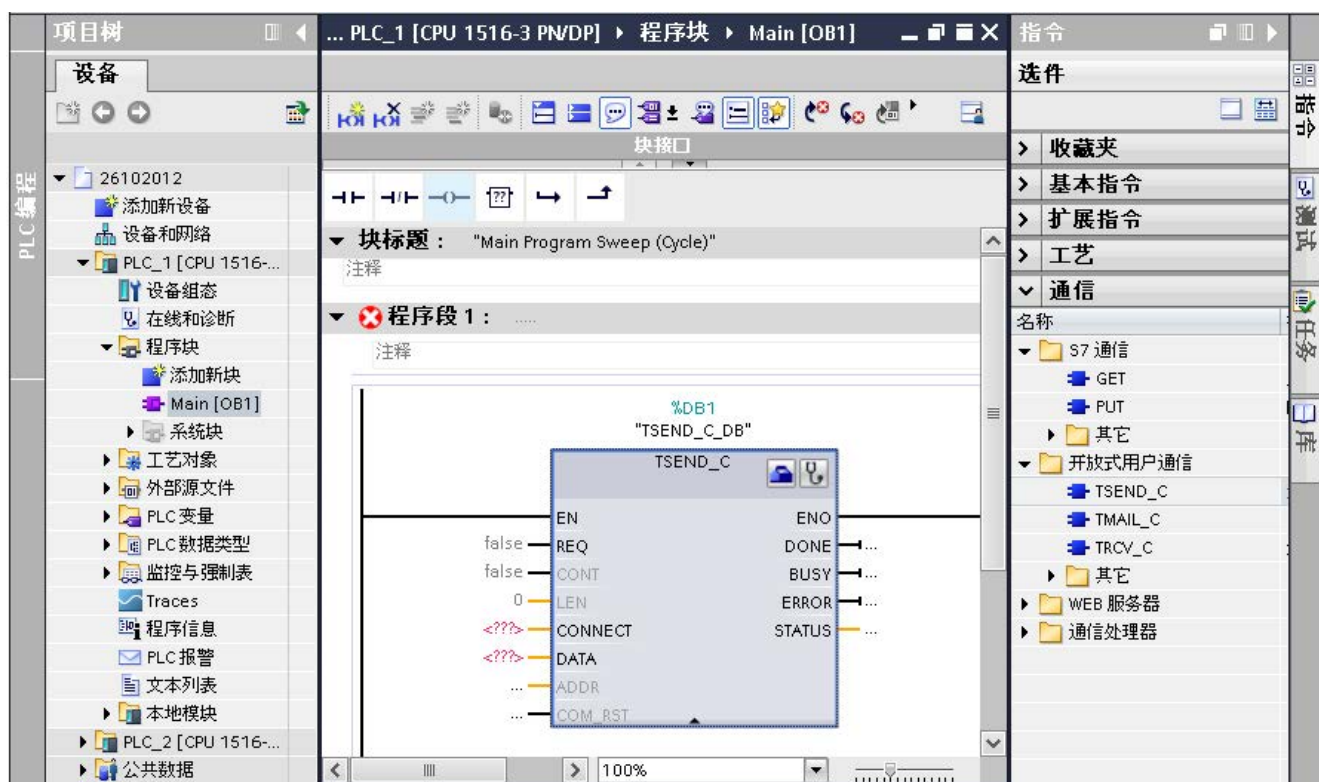


图 3-1 通过编程建立连接

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。

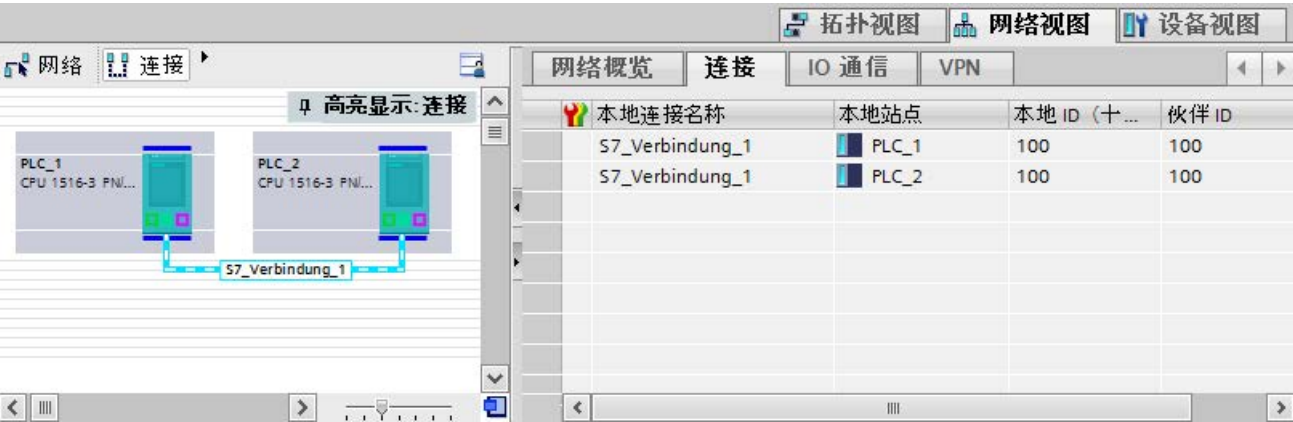


图 3-2 通过组态建立连接

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 3-2 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	√	-	√
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	-	-	√

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 202)”部分。

3.4 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。

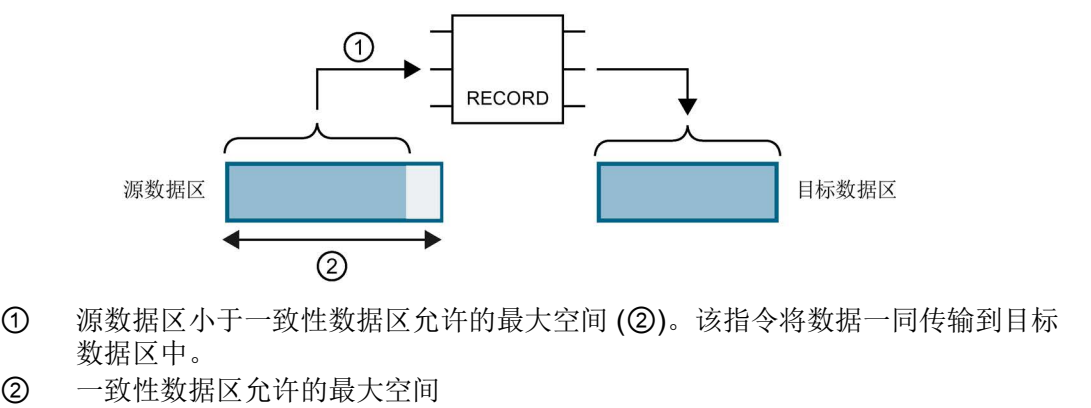
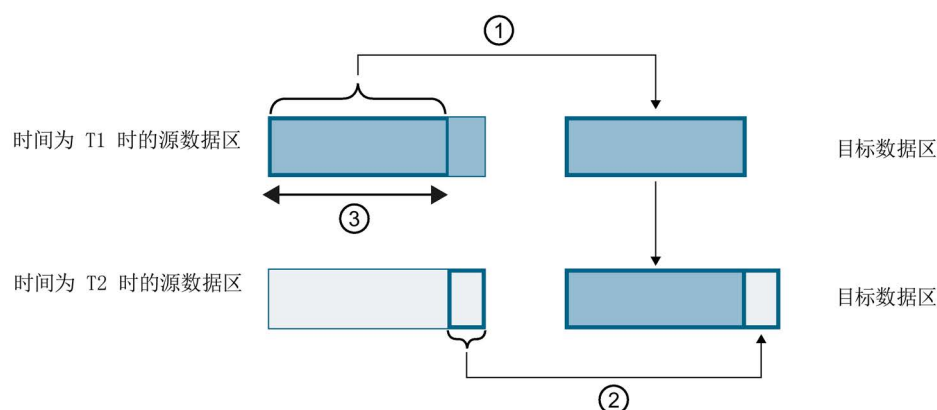


图 3-3 数据的一致性传输

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 一致性数据区允许的最大空间

图 3-4 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。

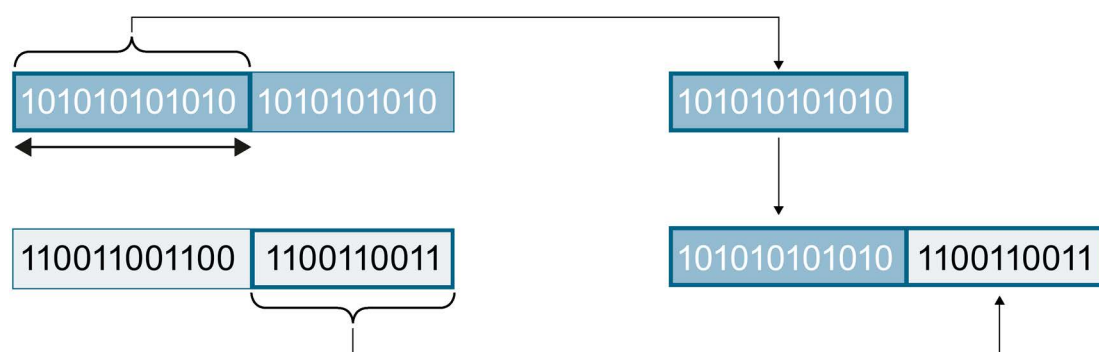


图 3-5 示例：在数据传输过程中，数据发生变更

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。

3.5 以太网通信的通信协议和端口号

在本章节中，将简要介绍通过 PN/IE 接口进行通信时所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，因此下表中不包含任何 PROFIBUS 协议。

在下表中，列示了不同的通信层级以及所用的协议。

下表列出了 S7-1500 CPU、ET 200SP CPU 和 CPU 1516pro-2 PN 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 3-3 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	端口号	(2) 链路层 (4) 传输层	功能	说明
PROFINET 协议				
DCP 发现和组态 协议	不相关	(2) Ethertype 0x8892 (PROFINET)	可访问的设备 PROFINET 发现 和组态	PROFINET 通过 DCP 发现 PROFINET 设备并提供基本设置。
LLDP 链路层发现 协议	不相关	(2) Ethertype 0x88CC (LLDP)	PROFINET 链路 层发现协议	PROFINET 通过 LLDP 发现和管理 PROFINET 设备间的相邻关系。 LLDP 使用特定的多播 MAC 地址：01-80-C2-00-00-0E
MRP 介质冗余协 议	不相关	(2) Ethertype 0x88E3 (IEC 62493-2-2010)	PROFINET 介质 冗余	MRP 通过环形拓扑结构对冗余传输路径 进控制。 MRP 将基于该标准使用多播 MAC 地址
PTCP 精确、透明 的时钟协议	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET 基于 IEEE 1588 进行时钟发送和时 间同步	PTC 提供 RJ45 端口之间的延时测量，并 随后发送时钟同步和时间同步。 PTCP 将基于该标准使用多播 MAC 地址
PROFINET IO 数据	不相关	(2) Ethertype 0x8892 (PROFINET)	PROFINET IO 数 据循环传输	PROFINET IO 帧用于基于以太网在 PROFINET IO 控制器和 IO 设备之间对 IO 数据进行循环传输。

3.5 以太网通信的通信协议和端口号

协议	端口号	(2) 链路层 (4) 传输层	功能	说明
PROFINET 上下文管理器	34964	(4) UDP	不带 RPC 的 PROFINET 连接	PROFINET 上下文管理器可作为一个端点映射器，用于建立应用关联 (PROFINET AR)。
面向连接的通信协议				
SMTP 简单邮件传输协议	25	(4) TCP	简单邮件传输协议	SMTP 用于发送电子邮件和短消息（具体取决于提供商）。
HTTP 超文本传输协议	80	(4) TCP	超文本传输协议	HTTP 用于与 CPU 内部 Web 服务器通信。
ISO on TCP （符合 RFC 1006 标准）	102	(4) TCP	ISO-on-TCP 协议	ISO on TCP（遵循 RFC 1006 标准）适用于与远程 CPU 或软件控制器进行面向消息的数据交换。 S7 与 ES、HMI、OPC Server 等通信。
NTP 网络时间协议	123	(4) UDP	网络时间协议	NTP 用于同步 CPU 系统时间与 NTP 服务器时间。
SNMP 简单网络管理协议	161 162（陷阱）	(4) UDP	简单网络管理协议	SNMP 由 SNMP 管理器用于读取和设置网络管理数据（SNMP 管理的对象）。
HTTPS 安全超文本传输协议	443	(4) TCP	安全超文本传输协议	HTTPS 用于通过安全套接层 (SSL) 与 CPU 内部的 Web 服务器通信。
Modbus TCP Modbus 传输控制协议	502	(4) TCP	Modbus/TCP 协议	Modbus/TCP 由用户程序中的 MB_CLIENT/MB_SERVER 指令使用。
OPC UA 开放式平台 通信 统一架构	4840	(4) TCP	基于 TCP/IP 协议	从企业级到现场级的通信标准。

协议	端口号	(2) 链路层 (4) 传输层	功能	说明
OUC ¹ 开放式用户通信 和 安全 OUC	1 到 1999 使用范围有限 ²	(4) TCP (4) UDP	开放式用户通信 (TCP/UDP) 安全的开放式用户通信 (TLS)	OUC 指令可用于建立连接、终止连接并基于套接层传输数据。
	2000 到 5000 (建议)			
	5001 到 49151 使用范围有限 ²			
IGMPv2 Internet 组管理协议	不相关	(3) 网络层	Internet 组管理协议	多播组的组织网络协议
预留	49152 到 65535	(4) TCP (4) UDP	-	如果应用程序未确定本地端口号，将使用动态端口区域激活连接端点。

¹ 注：基于开放式通信，可直接访问 UDP/TCP。用户必须遵守 IANA (Internet Assigned Numbers Authority, 互联网数字分配机构) 的端口限制/定义。

² 请勿将其它协议已用端口分配给 OUC。

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。

3.5 以太网通信的通信协议和端口号

表格 3-4 S7-1500 软件控制器的通信层和协议（通过 Windows 端的以太网接口）

协议	端口号	(2) 链路层 (4) 传输层	功能	说明
PROFINET 协议				
DCP 发现和组态 协议	不相关	(2) Ethertype 0x8892 (PROFINET)	可访问的设备 PROFINET 发现 和组态	PROFINET 通过 DCP 发现 PROFINET 设备并提供基本设置。
面向连接的通信协议				
SMTP 简单邮件传 输协议	25	(4) TCP	简单邮件传输协议	SMTP 用于发送电子邮件和短消息（具体取决于提供商）。
HTTP 超文本传输 协议	可调节 ¹	(4) TCP	超文本传输协议	HTTP 用于与 CPU 内部 Web 服务器通信。为了避免与 Windows 系统中其它 Web 服务器冲突，可更改端口号。 如果要使用 Web 服务器访问，必须在 Windows 防火墙中激活该端口。
ISO on TCP （符合 RFC 1006 标准）	102	(4) TCP	ISO-on-TCP 协议	ISO on TCP 用于与 PG/PC 或 HMI 进行 S7 通信（符合 RFC 1006 标准）。
OUC ² 开放式用户 通信 和 安全 OUC	1 到 1999 使用范围 有限 ^{3,4}	(4) TCP (4) UDP	开放式用户通信 (TCP/UDP) 安全的开放式用户 通信 (TLS)	OUC 指令可用于建立连接、终止连接并基于套接层传输数据。 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}			

协议	端口号	(2) 链路层 (4) 传输层	功能	说明
IGMPv2 Internet 组管理协议	不相关	(3) 网络层	Internet 组管理协议	多播组的组织网络协议
预留	49152 到 65535	(4) TCP (4) UDP	-	如果应用程序无法确定本地端口号，则可使用动态端口范围主动连接端点。 如果需使用此类通信，则需在 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：基于开放式用户通信，可直接访问 UDP/TCP。用户必须遵守 IANA (Internet Assigned Numbers Authority, 互联网数字分配机构) 的端口限制/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。

除 S7-1500 通信模块（如 CP 1543-1）表格已列示的协议之外，下表中列出了支持的更多协议。

3.5 以太网通信的通信协议和端口号

表格 3- 5 S7-1500 通信模块的层和协议

协议	端口号	(2) 链路层 (4) 传输层	功能	说明
PROFINET/工业以太网协议				
面向连接的通信协议				
FTP 文件传输协议	20（数据） 21（控制）	(4) TCP	文件传输协议	FTP 用于传输文件（仅与 CP 1543-1 连接）。
SecureFTP 文件传输协议	20（数据） 21（控制）	(4) TCP	文件传输协议	SecureFTP 用于通过 TLS 连接传输文件（仅与 CP 1543-1 连接）。
DHCP 主机动态配置协议	68	(4) UDP	主机动态配置协议	DHCP 用于在启动 IE 接口时从 DHCP 服务器检索 IP 地址套件。
安全 NTPv3 网络时间协议	123	(4) UDP	网络时间协议	安全 NTP 用于同步 CP 1543-1 内部系统时钟与 NTP 服务器。
SNMP 简单网络管理协议	161 162（陷阱）	(4) UDP	简单网络管理协议	SNMPv3 允许 CP 1543-1 通过认证从 SNMPv3 代理读取网络管理数据 (MIB)。

3.6 SNMP

3.6.1 禁用 SNMP

网络管理协议 **SNMP**（简单网络管理协议），采用多种服务和工具对网络拓扑机构进行检测和诊断。

在 **SNMP** 中，采用传输协议 **UDP**。**SNMP** 可识别两种网络组件：**SNMP** 管理器和 **SNMP** 客户端。**SNMP** 管理器用于对网络节点进行监视：**SNMP** 客户端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**（管理信息库）中。各种服务和工具将基于这些数据进行详细的网络诊断。

在某些特定条件下，需要禁用 **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。

如果禁用设备的 **SNMP** 功能，则无法使用不同方式对网络拓扑进行诊断（如，使用 **PRONETA** 工具或 **CPU** 中的 **Web** 服务器）。

禁用 SNMP

要禁用 S7-1500 CPU 集成接口的 SNMP 功能，请按以下步骤操作：

- 1. 在 STEP 7 中，创建一个包含数据记录 B071_H 结构的数据块。
 - 下表列出了数据记录 B071_H 的结构：

字节	元素	代码	说明
0 到 1	块 ID	F003 _H	标头 该数据记录的长度从字节 4“版本”开始计算。
2 到 3	块长度	8	
4	版本	01 _H	
5	子版本	00 _H	
6 到 7	预留	-	-
8 到 11	SNMP 控制器	禁用/启用 SNMP	如果要禁用 SNMP，请输入值 0。 如果要启用 SNMP，请输入值 1。

- 2. 通过 WRREC 指令，可将启动 OB (OB100) 中的数据记录 B071_H 传送到 CPU 中。
并将 CPU 中集成的接口作为硬件 ID。

3.6.2 示例：禁用 CPU 1516-3 PN/DP 的 SNMP 功能

任务

如果网络中的安全规则不允许使用 SNMP，则需禁用 CPU 1516-3 PN/DP 的 SNMP 功能。

要求

- CPU 1516-3 PN/DP，固件版本 V2.0
- STEP 7 V14

解决方法

首先，创建一个包含数据记录 B071_H 结构的数据块。下图显示了数据块“Deactivate SNMP”。数据块“Deactivate SNMP”中不仅包含数据记录 B071_H，还包含用于传输该数据记录的其它变量。变量“snmp_deactivate”用于触发 WRREC 作业。该变量应放置在保持性存储区中，以便启动 OB (OB100) 也可使用该值。

Deactivate SNMP								
	Name	Data type	Start value	R...	A...	W...	V...	Comment
1	Static			☐	☐	☐	☐	
2	snmp_deactivate	Bool	TRUE	☒	☒	☐	☒	Tag for deactivation
3	snmp_record	Struct		☒	☒	☐	☒	Data record 16#B071
4	BlockID	UInt	16#F003	☒	☒	☐	☒	
5	BlockLenght	UInt	8	☒	☒	☐	☒	
6	Version	USInt	1	☒	☒	☐	☒	
7	Subversion	USInt	0	☒	☒	☐	☒	
8	Reserved	UInt	0	☒	☒	☐	☒	
9	SNMPControl	UDInt	0	☒	☒	☐	☒	
10	snmp_done	Bool	false	☐	☒	☐	☒	
11	snmp_error	Bool	false	☐	☒	☐	☒	
12	snmp_status	DWord	16#0	☐	☒	☐	☒	

图 3-6 示例：禁用 SNMP 的数据块

通过 WRREC 指令（写数据记录），可将启动 OB (OB100) 中的数据记录 B071_H 传送到 CPU 1516-3 PN/DP 中。

在以下程序代码中，在 REPEAT UNTIL 循环中使用 WRREC 指令传输数据记录 B071_H。

```

ORGANIZATION_BLOCK "Startup"
TITLE = "Complete Restart"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
BEGIN
  REPEAT
    "WRREC_DB_1"
    (REQ := "Deactivate SNMP".snmp_deactivate,
      //Transfer data record
      INDEX:=16#B071,
      //Data record number for SNMP deactivation
      ID:="Local~PROFINET_interface_1",
      //any integrated PROFINET Interface
      DONE => "Deactivate SNMP".snmp_done,
      ERROR => "Deactivate SNMP".snmp_error,
      STATUS => "Deactivate SNMP".snmp_status,
      RECORD := "Deactivate SNMP".snmp_record)
      //Data record
  UNTIL "Deactivate SNMP".snmp_done OR "Deactivate
  SNMP".snmp_error
  END_REPEAT;
END_ORGANIZATION_BLOCK

```

使用程序代码

有关完整的程序代码，请访问[这里](#)。

要在项目中用于该程序代码，请按以下步骤操作：

1. 使用快捷键 **Ctrl+A** 和 **Ctrl+C**，将完整的程序代码复制到剪贴板中。
2. 打开文本编辑器（如，“**Editor**”）。
3. 使用快捷键 **Ctrl+V**，将剪贴板中的内容粘贴到文本编辑器中。
4. 将文档另存为 **scl** 文件，如 **SNMP_DEACT.scl**。
5. 在 **STEP 7** 中打开项目。
6. 将该 **scl** 文件作为外部资源导入。
有关导入外部源文件的更多信息，请参见 **STEP 7** 在线帮助。
7. 生成启动 **OB** 和数据块。（右键单击该 **scl** 文件，选择快捷菜单：“从源生成块”(Generate blocks from source)）

重新启用 SNMP

只需对以上程序代码进行少量改动，即可启用 **SNMP**。

在用户程序中，将变量 **"Deactivate SNMP".snmp_record.SNMPControl** 的值赋值为“1”：
`"Deactivate SNMP".snmp_record.SNMPControl := 1;`

SNMP 将在 **CPU** 下一次启动时重新启用。

PG 通信

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。

PG/PC 可以在线连接到一个 CPU。PG/PC 同时可操作最多 4 个在线连接（例如，连接 4 个 CPU）。

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“编程设备/PC 接口”(PG/PC interface) 下拉列表中，选择要用于建立在线连接的编程设备/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。

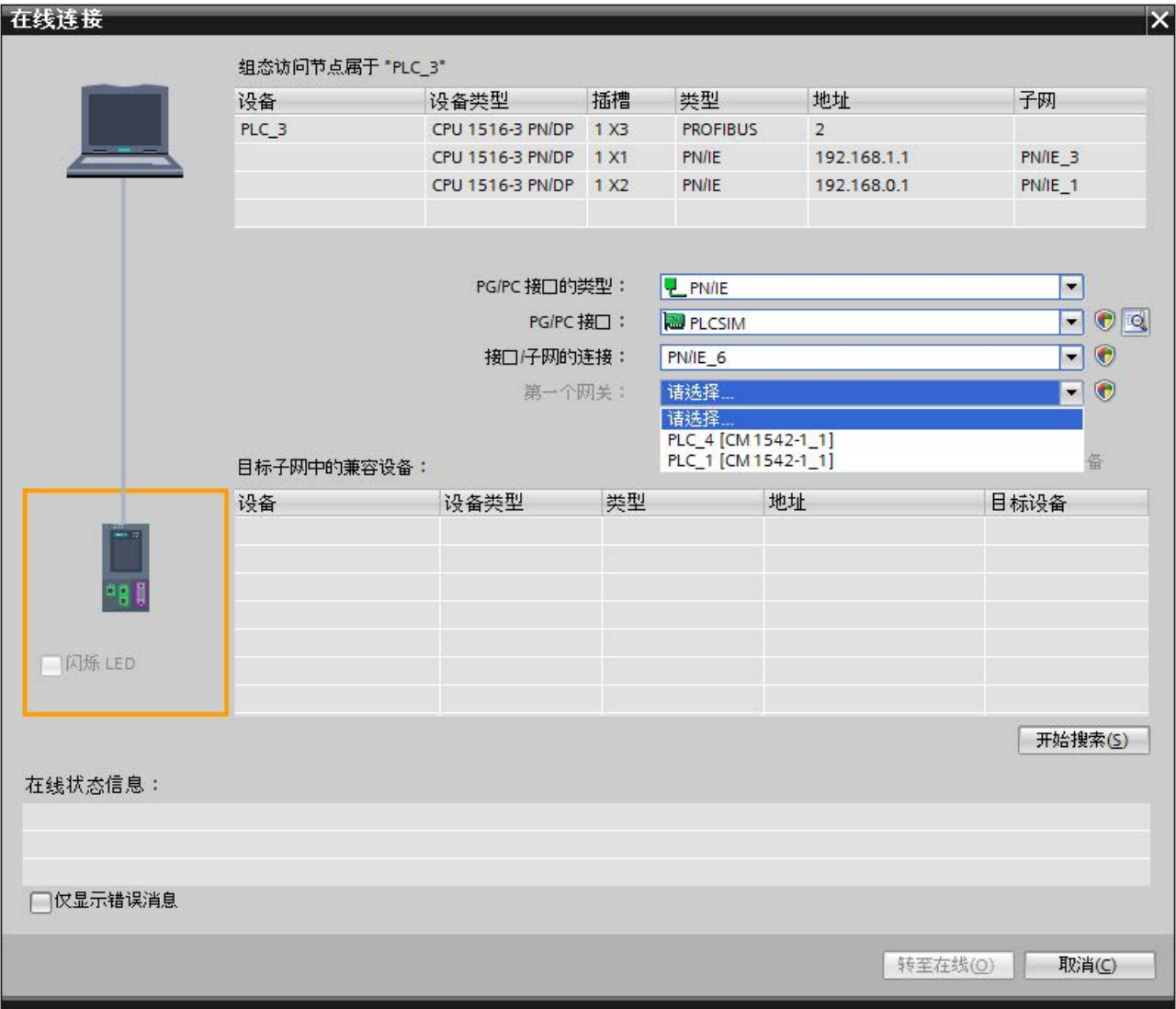


图 4-1 建立 PG 通信

4. 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。

附加信息

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

HMI 通信

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

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。

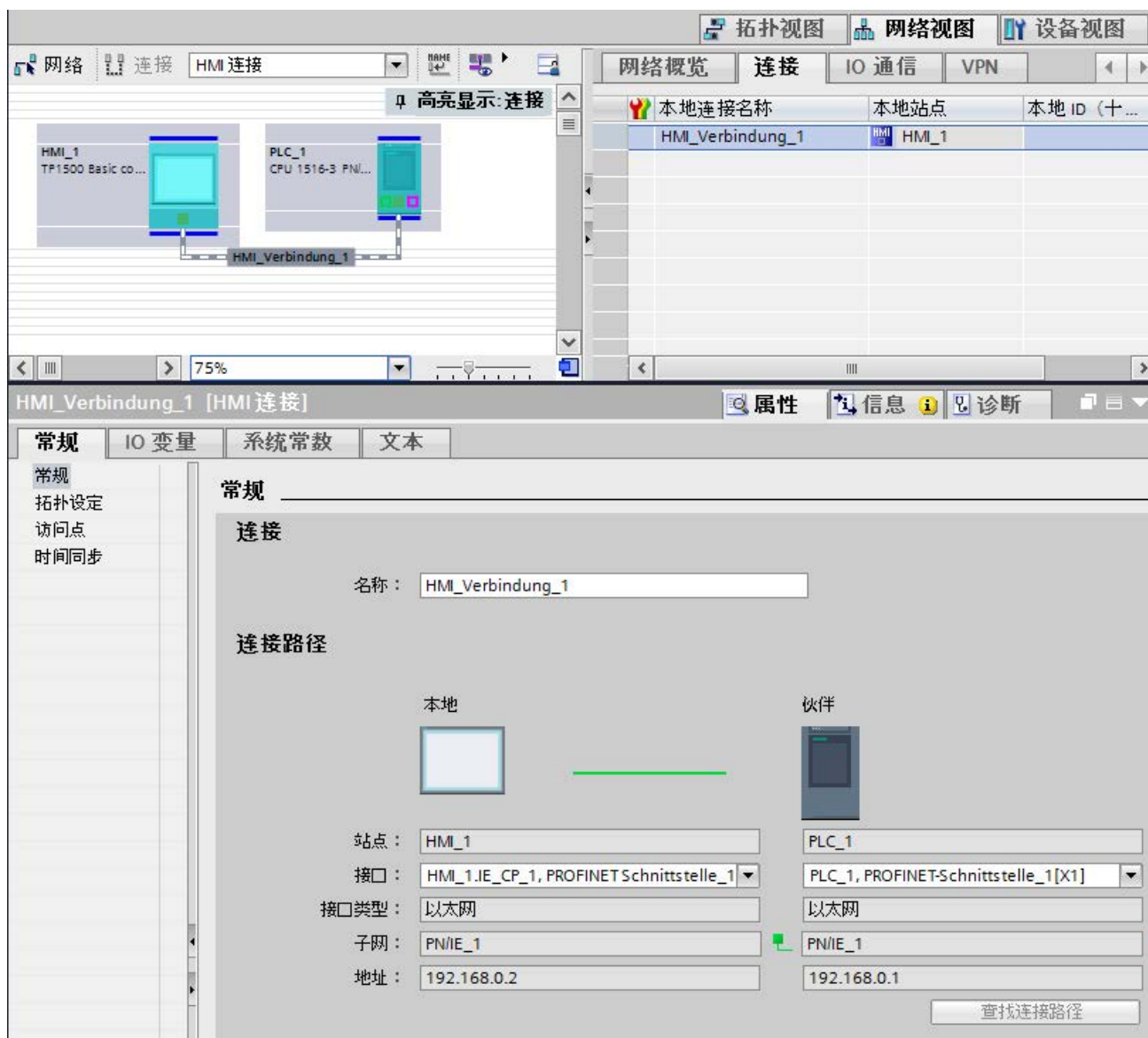


图 5-1 建立 HMI 通信

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 195)”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

6.1 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。
- 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 1516pro-2 PN）
- 通信模块的 PROFINET/工业以太网接口（如，CP 1543-1, CM 1542-1）

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 74)”部分。

6.2 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 6-1 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 6-2 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。

* CPU 固件版本 V2.0 及以上版本，PROFINET 接口最多支持 5 个多播组

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。

6.3 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，通过用户程序建立连接而非通过硬件配置中的组态静态通信连接，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：

- 手动创建这种数据块，并为其分配参数，然后再将该数据块直接写入指令中（建立电子邮件和 FTP 连接时，需要使用）
- 建议：在组态 TSEND_C、TRCV_C 和 TCON 指令的连接期间，在程序编辑器中的属性中自动创建数据块。

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。

6.3 开放式用户通信的指令

通过编程建立连接时的协议、系统数据类型和可用指令

表格 6-3 通过编程建立连接的指令

协议	系统数据类型	用户程序指令
TCP	• TCON_QDN • TCON_IP_v4	建立连接并通过以下指令收/发数据：
ISO-on-TCP	• TCON_IP_RFC	• TSEND_C/TRCV_C 或 • TCON, TSEND/TRCV 或 • TCON, TUSEND/TURCV (可通过 TDISCON 终止连接)
ISO, 符合 ISO/IEC 8073 (Class 4) 标准	• TCON_ISOnative¹ • 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：
FDL ¹	• TCON_FDL	建立连接并通过以下指令收/发数据：
Modbus TCP	• TCON_IP_v4	• TSEND_C/TRCV_C 或 • TCON, TSEND/TRCV 或 • TCON, TUSEND/TURCV (可通过 TDISCON 终止连接)
		• MB_CLIENT • MB_SERVER

协议	系统数据类型	用户程序指令
电子邮件	• TMAIL_v4 • TMAIL_v6 • TMAIL_FQDN	• TMAIL_C
FTP ²	• FTP_CONNECT_IPV4³ • FTP_CONNECT_IPV6³ • FTP_CONNECT_NAME³	• 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

6.3 开放式用户通信的指令

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 6-4 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

用于开放式用户安全通信的指令和系统数据类型

有关开放式用户安全通信的指令和系统数据类型的信息，请参见“开放式用户安全通信 (页 74)”部分。

更多信息

STEP 7 在线帮助介绍了：

- 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“连接资源分配 (页 206)”部分。

6.4 通过域名进行寻址的开放式用户通信

S7-1500 CPU、ET 200SP CPU 和 CPU 1516pro-2 PN 固件版本 V2.0 及以上版本，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 的通信，请按以下步骤操作：

1. 在 STEP 7 的网络视图选择 CPU。
2. 在巡视窗口中，选择“属性 > 常规 > DNS 配置”（Properties > General > 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。

6.4 通过域名进行寻址的开放式用户通信



图 6-1 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 6-2 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。

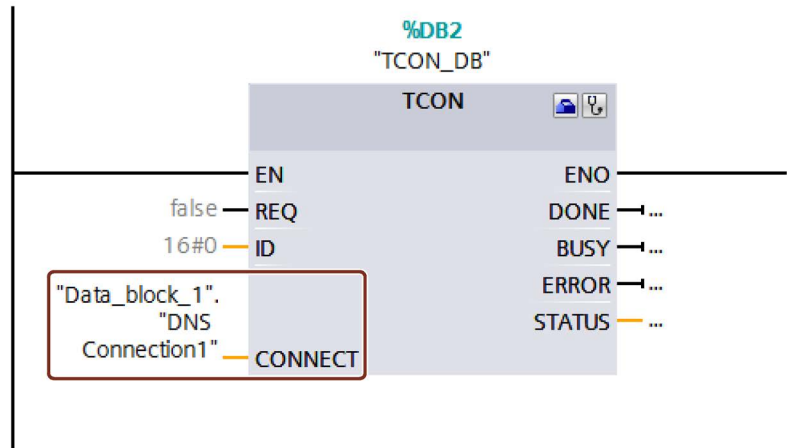


图 6-3 TCON 指令

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。

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 74)”部分。

6.5 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

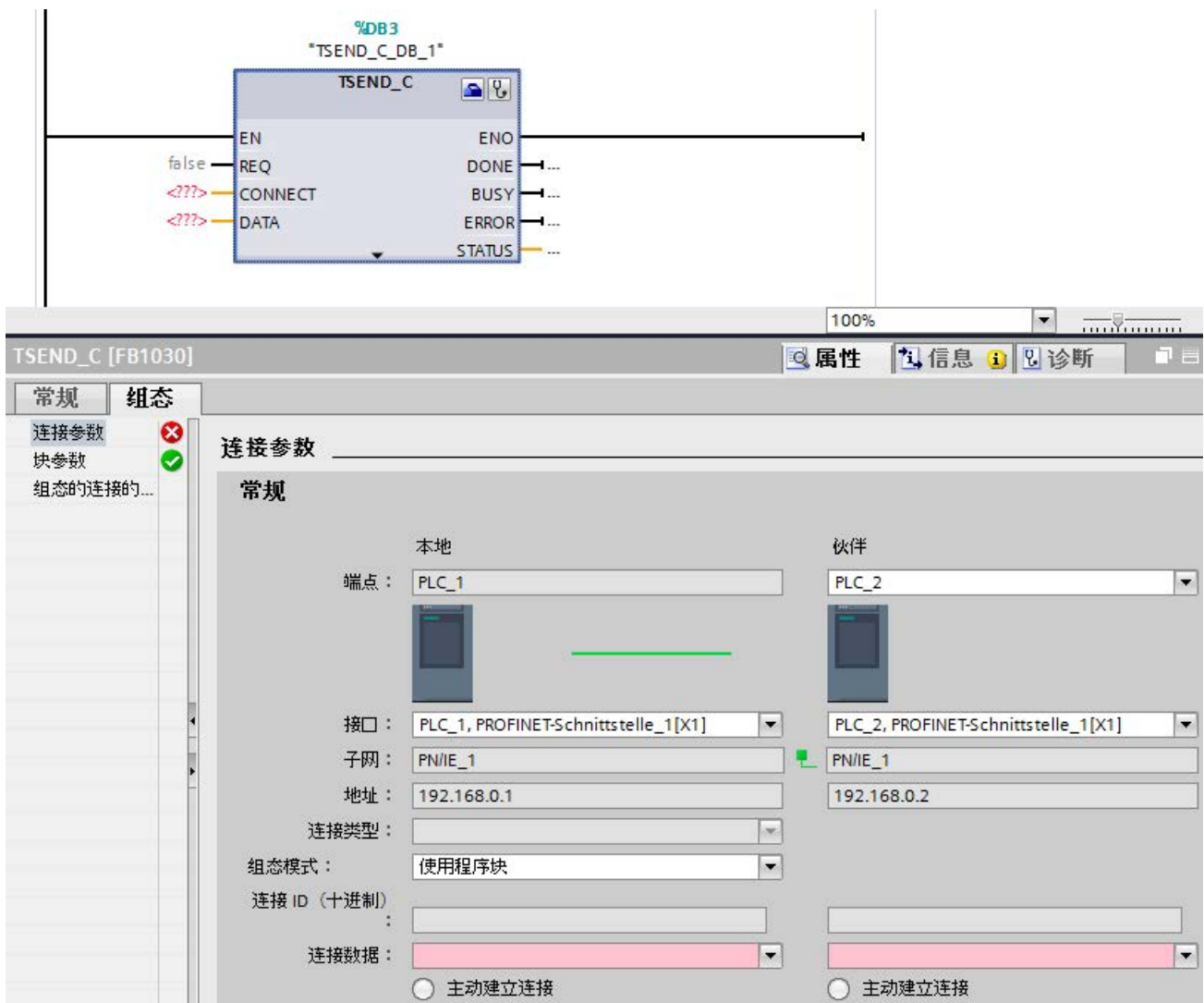


图 6-4 TSEND_C 的连接参数

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。

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

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 6-5 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。

6.6 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。

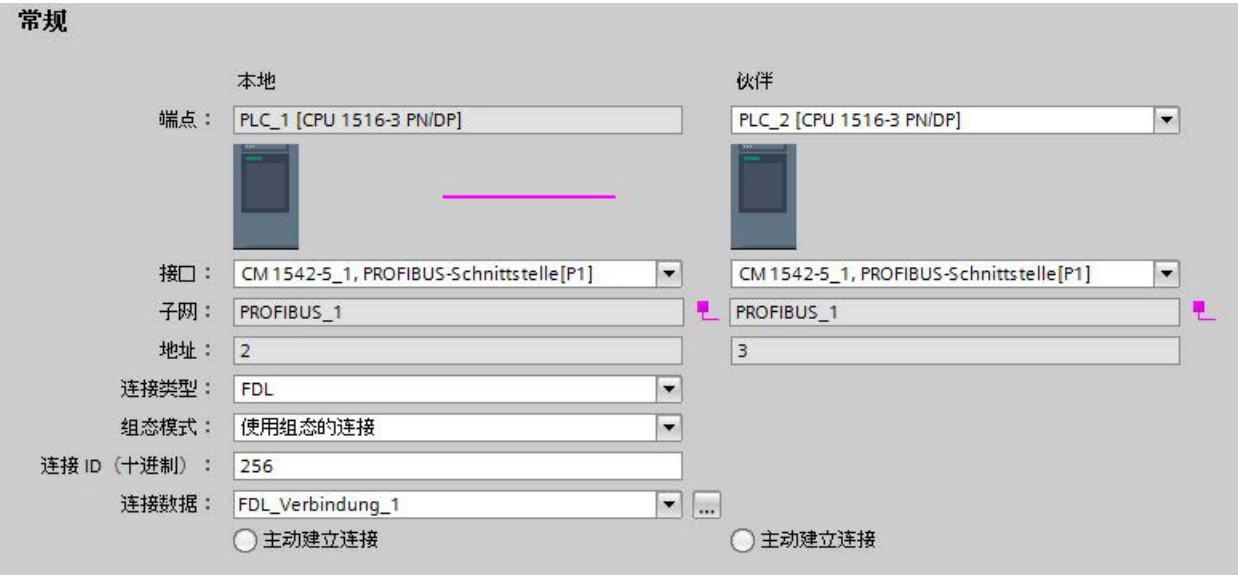


图 6-6 组态 FDL 连接

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	...	...	...	...	...	...	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 6-7 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）。

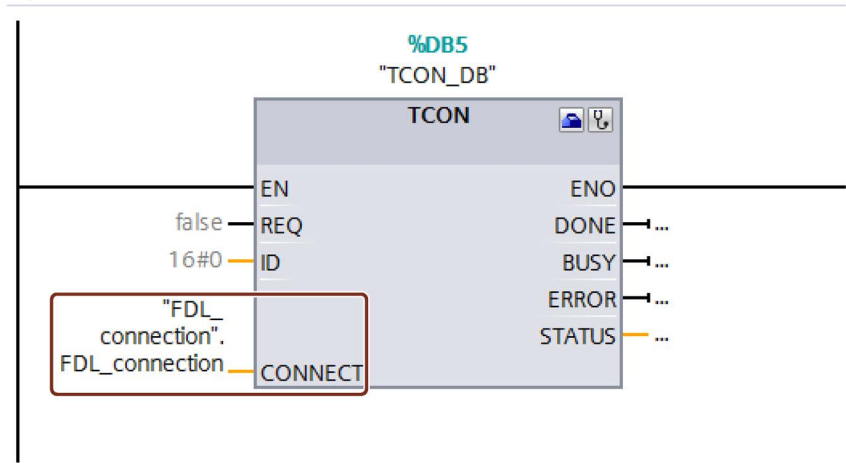


图 6-8 示例：FDL 连接的 TCON 指令

6.7 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 连接都必须使用具有数据结构 TCON_IP_v4 的唯一背景数据块。

每个连接都需要一个唯一的连接 ID。连接 ID 与背景数据块组合成对，对于每个连接，组合对都必须唯一。

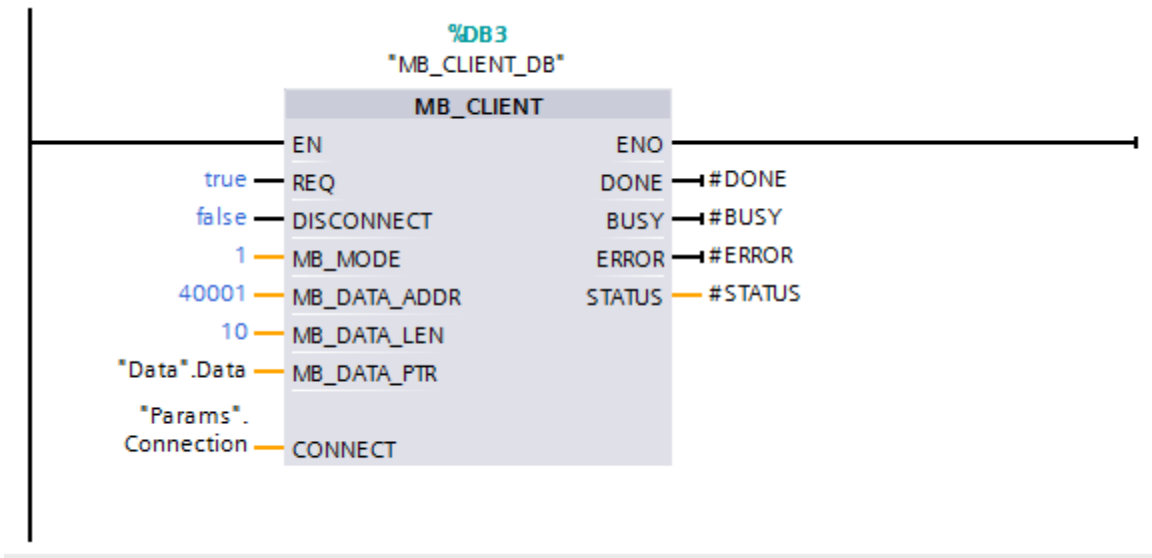


图 6-9 MB_CLIENT

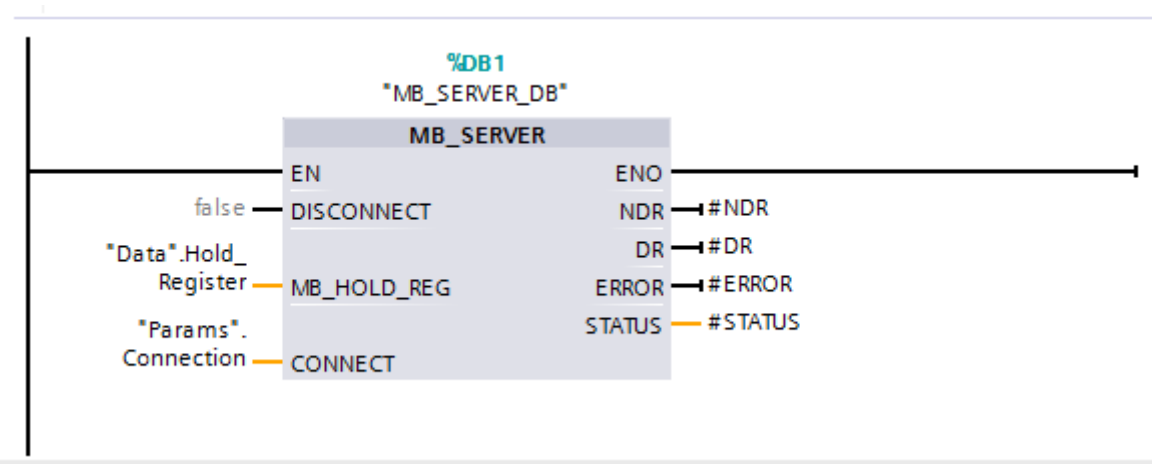


图 6-10 MB_SERVER

5. 将硬件配置和用户程序下载到 CPU。

Modbus TCP 服务器作为连接到 Modbus RTU 的网关

如果使用 Modbus TCP 服务器作为连接到 Modbus RTU 协议的网关，请使用静态参数 MB_UNIT_ID 对串行网络中的从站设备进行编址。MB_UNIT_ID 参数对应于 Modbus RTU 协议的从站地址域。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。

两个 S7-1500 CPU 之间的 Modbus TCP 连接

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 通信进行编程和参数化。

6.8 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. 在指令 TMAIL_C 中设置 SMTP 服务器的地址参数。（建议：指定附录名称。）
3. 创建一个全局数据块，并在此数据块内创建类型为 TMAIL_v4、TMAIL_v6（仅限 CP 1543-1）或 TMAIL_FQDN（仅限 CP 1543-1）的变量。
4. 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 在线帮助介绍了：

- 系统数据类型
- 开放式通信的指令
- 连接参数

6.9 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。FTP 的客户端功能将调用 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“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。

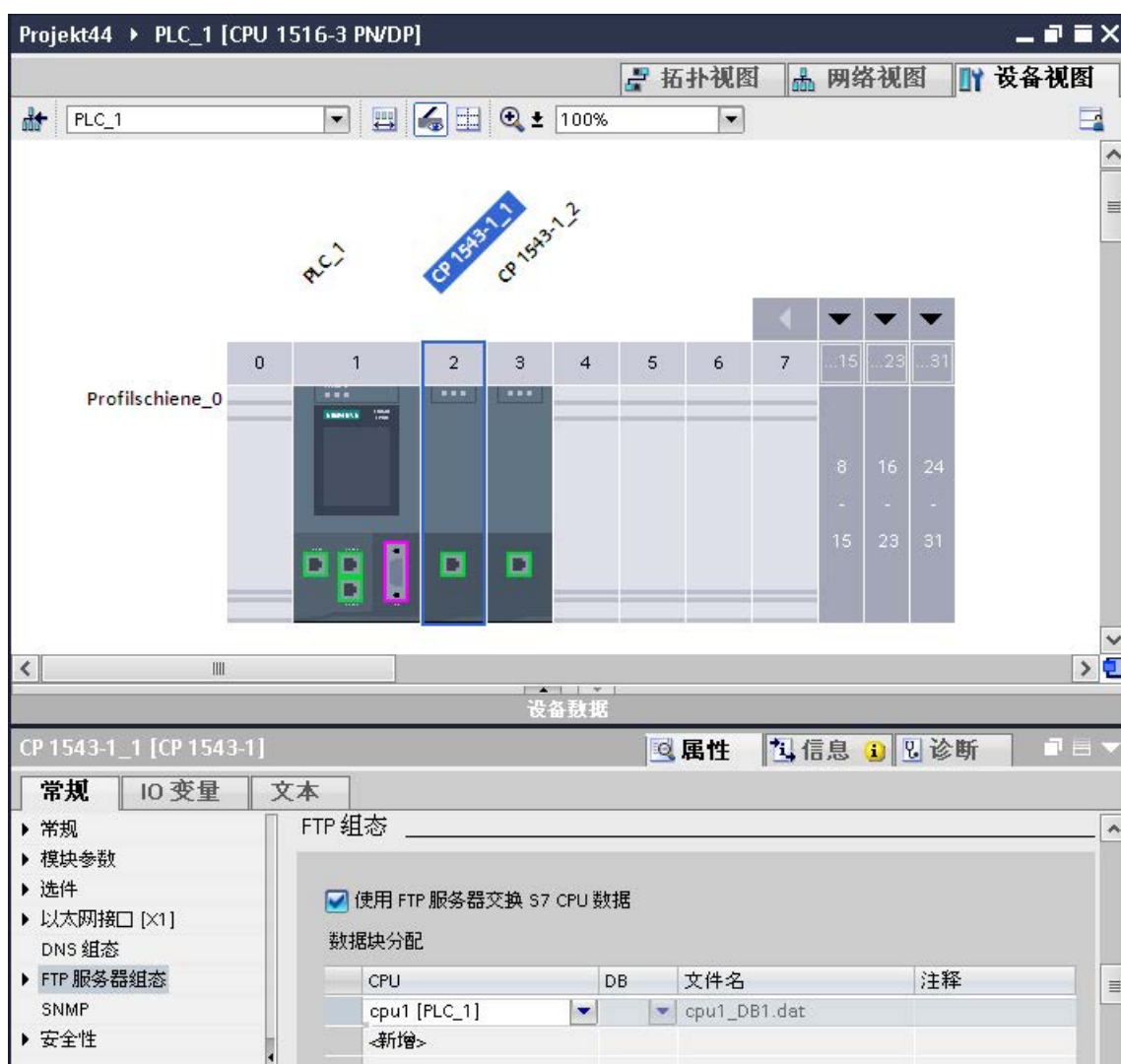


图 6-11 设置 FTP 组态

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：评估与 S7-1200/1500 进行的 FTP 客户端通信
有关应用示例，敬请访问 Internet
(<https://support.industry.siemens.com/cs/cn/zh/view/81367009>)。

更多信息

STEP 7 在线帮助介绍了：

- 系统数据类型
- 开放式通信的指令
- 连接参数

6.10 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 6-5 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。

6.11 开放式用户安全通信

6.11.1 基本知识

6.11.1.1 安全通信的基础知识

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 以及 S7-1500pro CPU、ET200SP CPU 和 SIMATIC S7-1500 软件控制器。

简介

“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。

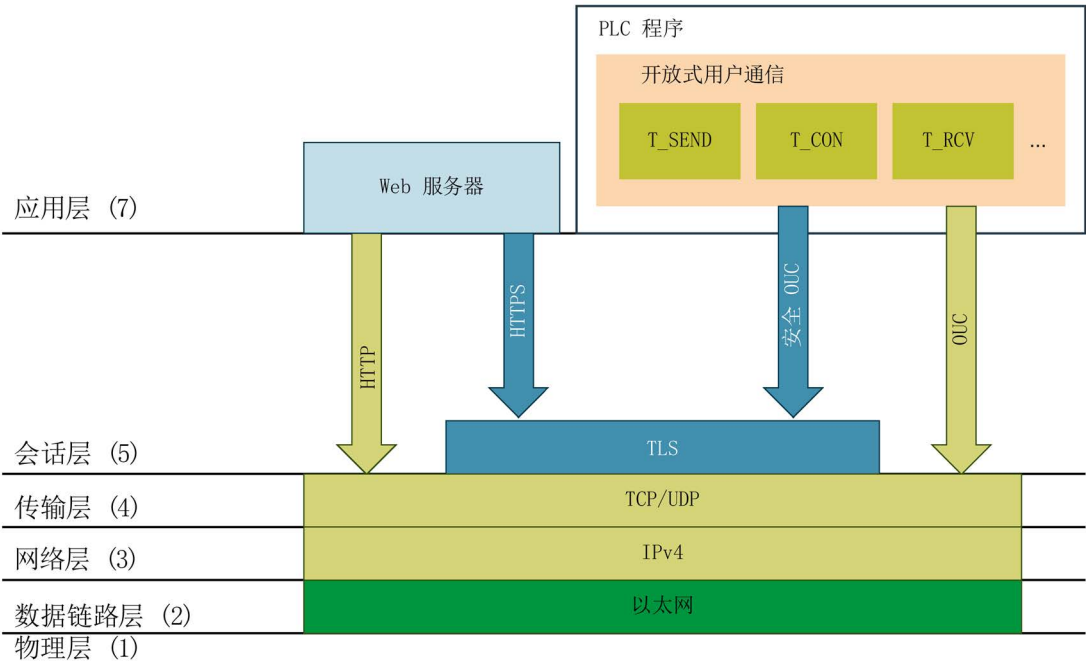


图 6-12 通信层中的 TLS 协议。

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。有关 OPC UA 服务器的功能说明，请参见“OPC UA 服务器 (页 131)”部分。

6.11.1.2 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。

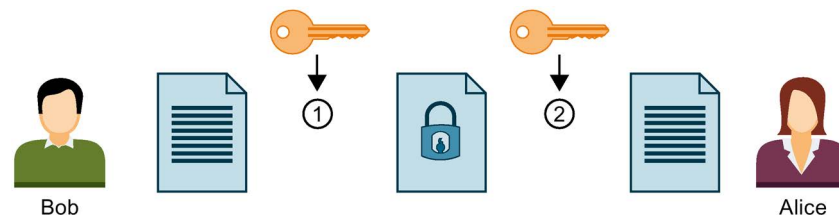
在进行消息加密时，采用了大量的数学处理机制（算法）。

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 6-13 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。

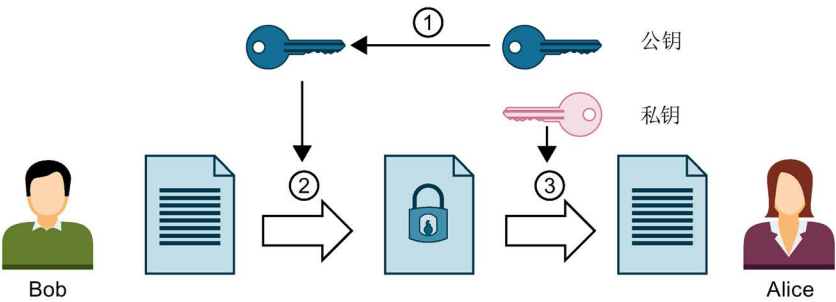
- 优势：对称加密算法（如，AES、Advanced Encryption Algorithm）的速度较快。
- 缺点：如何将密钥发送给接收方，而不会落到其他入手中？此为密钥分发问题。如果截获的消息数量足够大，则可推算出所用的密钥，因此必须定期更换。

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，而不仅仅是 Bob 的消息。

图 6-14 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 服务器通信和开放式用户安全通信），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。

6.11.1.3 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，则会造成诸如 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。

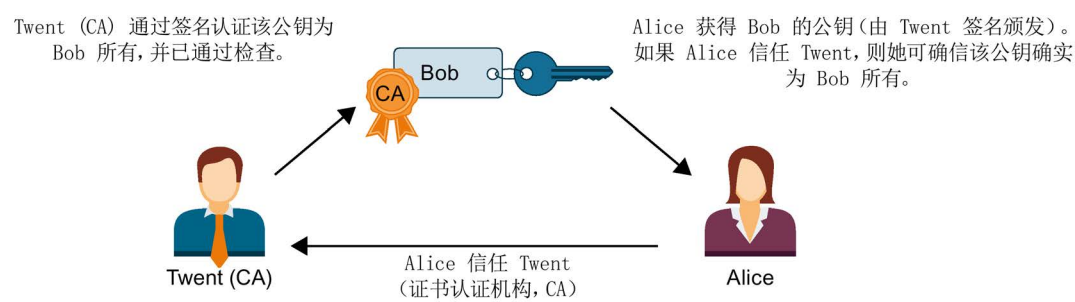


图 6-15 由证书颁发机构对证书进行签名

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 Authority) 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。
如在安全的开放式用户通信中，使用 STEP 7 创建一个新的证书，则可从可能的用途列表中选择相应的条目，例如“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。

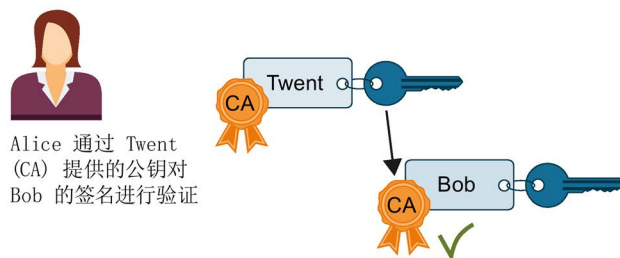


图 6-16 使用证书颁发机构的证书公钥对证书进行验证

签名消息

在上文中介绍的签名与验证机制，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哈希值并使用私钥对该哈希值进行签名，之后将其添加到原始消息中，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。

6.11.1.4 使用 STEP 7 管理证书

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 及以上版本使用的 PKI 与 Internet PKI 相似。例如，证书吊销列表 (CRL) 不受支持。

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：

- “保护和安全性 > 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。
- “Web 服务器 > 服务器安全”(Web server > Server security) - 用于生成和分配 Web 服务器证书。
- “OPC UA > 服务器 > 安全”(OPC UA > Server > Security)- 用于生成和分配 OPC UA 证书。



图 6-17 STEP 7 中 S7-1500 CPU 的安全设置

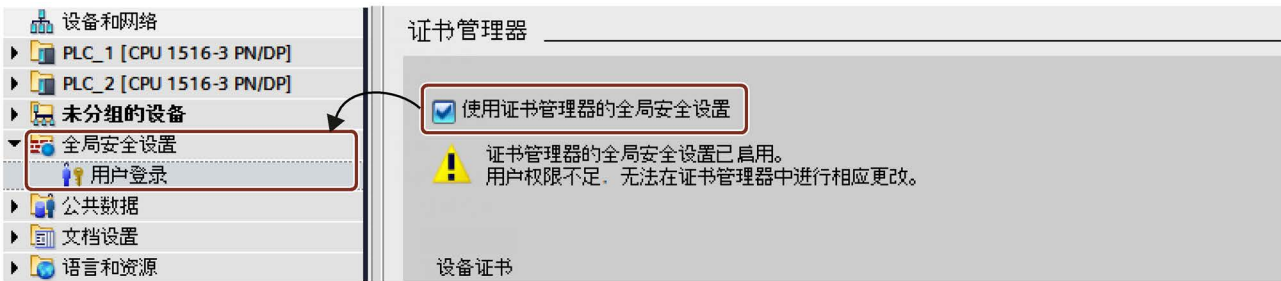
“保护与安全 > 证书管理器”(Protection & Security > Certificate manager) 区域的特性

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。

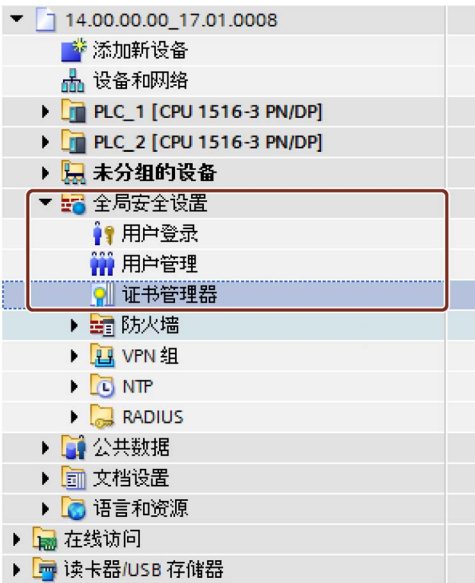
6.11 开放式用户安全通信

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
“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。

6.11.1.5 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。

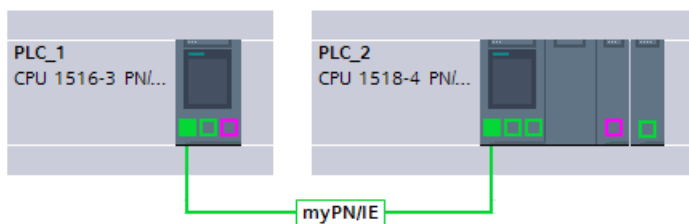
两个 S7-1500 CPU 之间安全的开放式用户通信

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“保护和安全”(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. 标记 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。

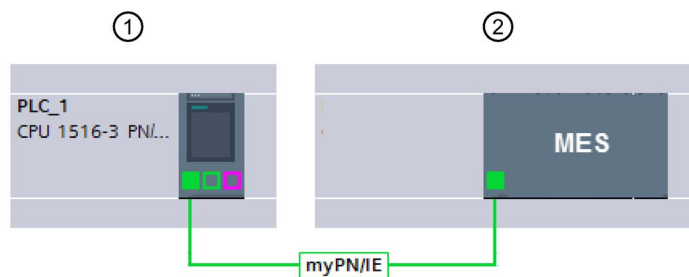
在下一个操作步骤中，需创建用户程序进行数据交换，并加载组态和该程序。

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. 标记 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。

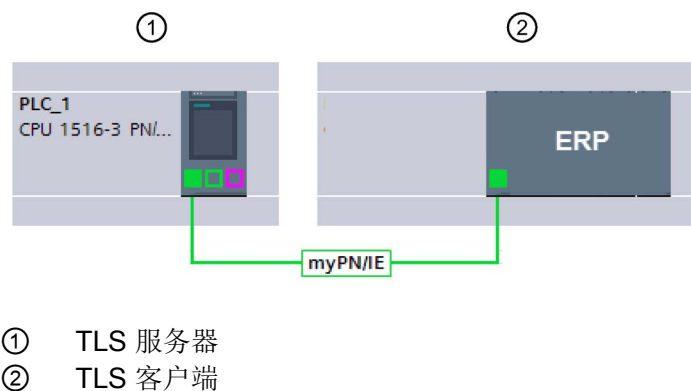


图 6-18 S7-1500 CPU 与 ERP 系统间的 OUC 安全通信

相关操作步骤，请参见上文介绍。

通过 S7-1543-1 CP（作为电子邮件客户端）实现的安全的开放式用户通信（基于 TLS 的 SMTP）

S7-1500 CPU 可使用通信指令 TMAIL-C 通过 CP 1543-1 与电子邮件服务器建立安全连接。

系统数据类型 TMail_V4_SEC、TMail_V6_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。

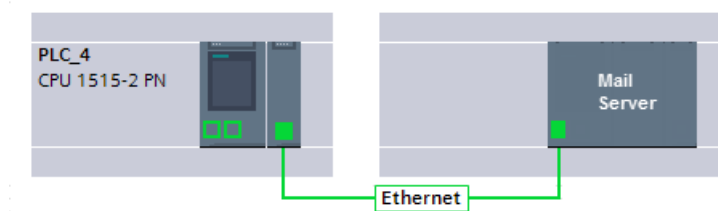


图 6-19 CP1543-1 与邮件服务器间的 OUC 安全通信

要建立电子邮件安全连接，则需将电子邮件服务器（提供方）的 Root 证书和中间证书导入 STEP 7 项目的全局证书存储器中。如果这些证书已分配给 CP，则 CP 将在建立 TLS 连接/会话时通过这些证书验证电子邮件服务器发送的服务器证书。

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（Root 证书颁发机构的可信证书）。
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，系统将其指定一个证书 ID，并在下一步操作中将其分配给该 CP。

4. 标记 CP 1543-1，并导航到“安全”(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. 在下拉列表中选择电子邮件服务器所需的 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。

6.11.1.6 示例：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 > 服务器安全”(Web server > 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：如果激活 Web 服务器和“仅允许 HTTPS 访问”(Permit access only with HTTPS) 选项，则 STEP 7 将在编译过程中使用默认名称生成一个服务器证书。

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。

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（证书颁发机构），可从 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”）。

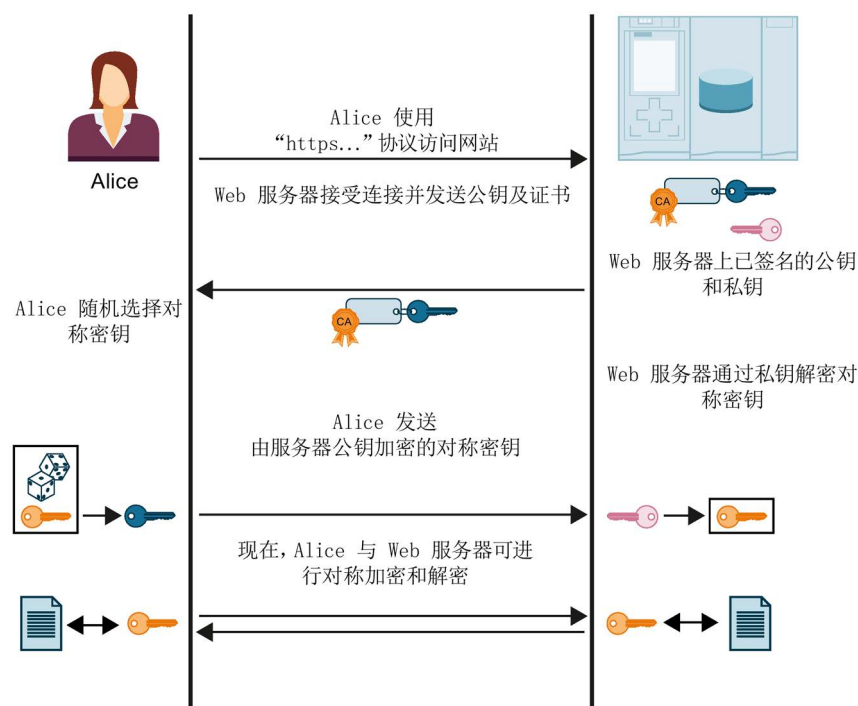


图 6-20 基于 HTTPS 的通信握手

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 的证书中心通常包含有所需的 Root 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，则需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。

6.11.2 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 TCON_QDN_SEC 系统数据类型的数据块，分配相应参数并在 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 服务器具备所需的全部证书。

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	Interfaceld	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 ad
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 6-21 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，则表示客户端将验证服务器端 X.509-V3 证书中的 subjectAlternateName，以验证该服务器的身份。该验证过程由 TCON 指令完成。
 - “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。

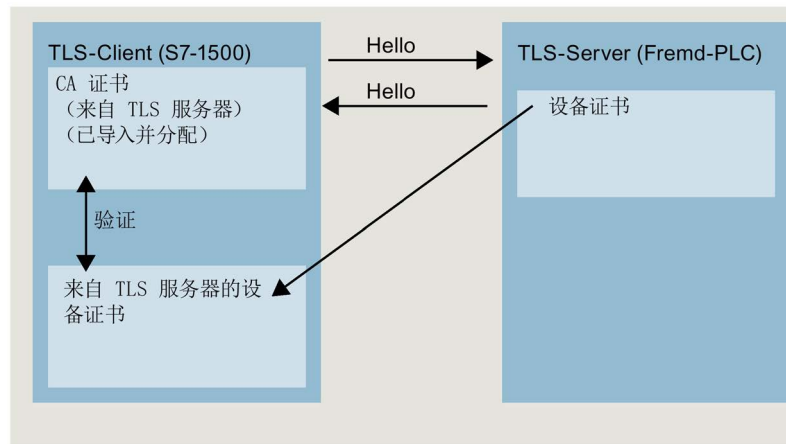


图 6-22 从作为 TLS 客户端的 S7-1500 的角度处理证书

- “TLSClientCertRef”：自身 X.509-V3 证书的 ID。

5. 在程序编辑器中，创建一个 TCON 指令。
6. 将 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。

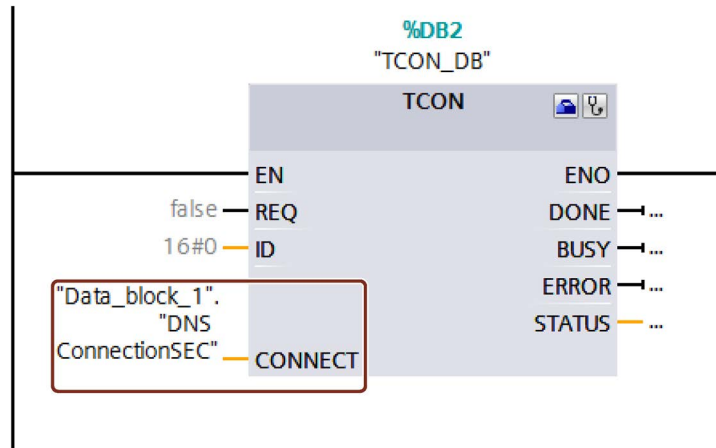


图 6-23 TCON 指令

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“基本知识 (页 74)”部分。

6.11.3 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 TCON_QDN_SEC 系统数据类型的数据块，分配相应参数并在 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 服务器具备所需的全部证书。

要建立与 TLS 客户端的安全 TCP 连接，请按以下步骤操作：

- 在项目树中，创建一个全局数据块。
- 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，定义了数据类型为 TCON_FDL_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	Interfaceld	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 ad...
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 6-24 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。

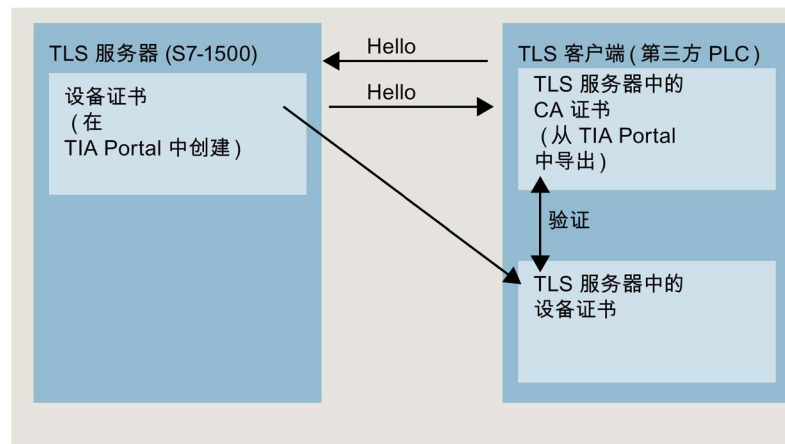


图 6-25 从作为 TLS 服务器的 S7-1500 的角度处理证书

- “TLSCertRef”：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 TCON 指令。
- 6. 将 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。

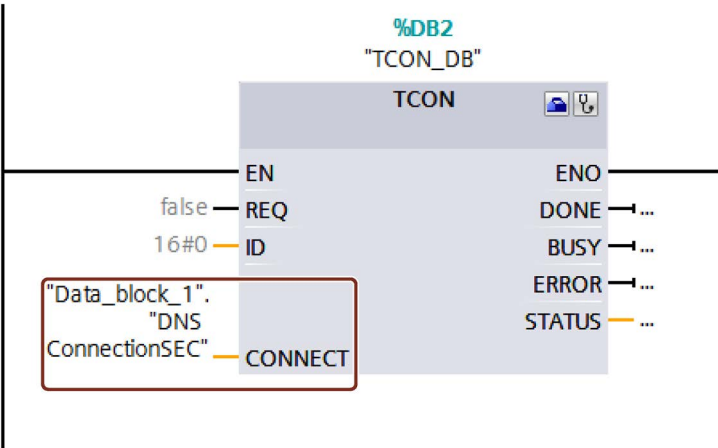


图 6-26 TCON 指令

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“基本知识 (页 74)”部分。

6.11.4 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 充当 TLS 客户端（主动建立连接）而另一个 S7-1500 CPU 充当 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

要建立两个 S7-1500 CPU 之间的安全 TCP 通信，需要为每个 CPU 手动创建 TCON_IPv4_SEC 系统数据类型的数据块，然后分配相应参数并在 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 服务器具备所需的全部证书。

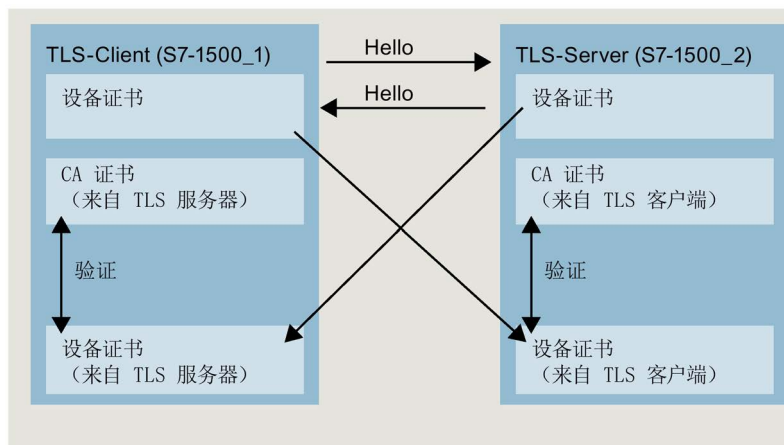


图 6-27 两个 S7-1500 CPU 之间安全 OUC 的证书处理过程

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_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 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 connction: 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	TLSCClientCertRef	UDInt	5	for Client side: add id of own X.509 V3 client certificate; for Sen

图 6-28 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）。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。
 - “TLSCClientCertRef”：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_4_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	Interfaceld	HW_ANY	120	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connction: 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; for
20	TLSClientCertRef	UDInt	1	for Client side: add id of own X.509 V3 client certificate; for Serv

图 6-29 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。

6.11 开放式用户安全通信

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 TCON 指令。
6. 将 TCON 指令的 CONNECT 参数与 TCON_IP_4_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“SEC connection 1 TLS client”（数据类型 TCON_IP_4_SEC）互连。

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“基本知识 (页 74)”部分。

6.11.5 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 及以上版本（S7-1500 软件控制器除外）
- CP 1543-1 固件版本 V2.0 及以上版本，或 CP 1543SP-1 V1.0 及以上版本

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

要在两个 S7-1500 CP 之间建立 TCP 安全通信，则需在各个 CPU 中创建系统数据类型为 TCON_IPv4_SEC 的数据块、进行参数分配并在指令中直接调用该数据块。

要求：

- 这两个 S7 1500 CPU 的固件版本为 V2.0 及以上版本如果使用 CP 1543SP-1：固件 V1.0 及以上版本。
- 这两个 CP（如 CP 1543-1）的固件版本必须 V2.0 及以上版本
- TLS 客户端和 TLS 服务器具备所需的全部证书。
 - 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。

下图显示了两个通信伙伴通过 CP 1543-1 进行通信时，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”）。

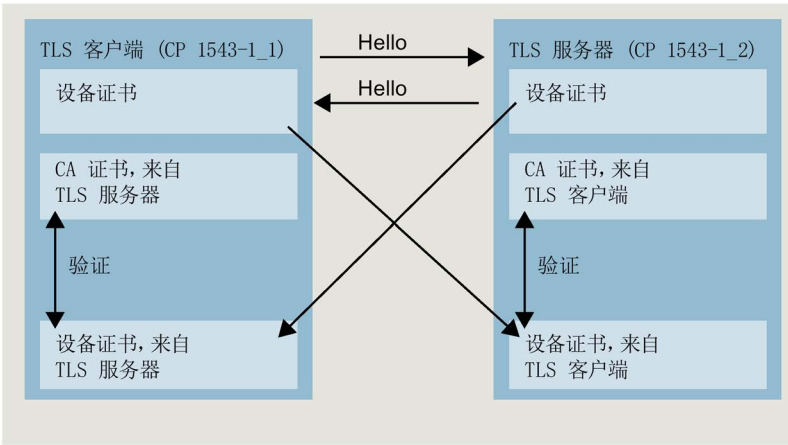


图 6-30 通过 CP 接口，在两个 S7-1500 CPU 之间进行 OUC 安全连接的证书处理操作。

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	Interfaceld	HW_ANY	258	HW-identifier of IE-interface submodule	
5	ID	CONN_OUC	10	connection reference / identifier	
6	ConnectionType	Byte	11	type of connction: 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	TLSCClientCertRef	UDInt	5	for Client side: add id of own X.509 V3 client certificate; for Sen	

图 6-31 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）。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 TCON 指令。
6. 将 TCON 指令的 CONNECT 参数与 TCON_IP_4_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)。

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	Interfaceld	HW_ANY	260	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connction: 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; for
20	TLSCClientCertRef	UDInt	1	for Client side: add id of own X.509 V3 client certificate; for Serv

图 6-32 IP_V4_SEC_Server

- 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。
- 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。
 - “TLSCClientCertRef”：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。
- 在程序编辑器中，创建一个 TCON 指令。
- 将 TCON 指令的 CONNECT 参数与 TCON_IP_4_SEC 数据类型的变量进行互连。

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 **CPU STOP** 模式下进行。
建议：加载到设备 > 硬件和软件 (**Load to device > Hardware and software**)。原因：
需确保带有安全 OUC 的程序、硬件配置和证书一致。
证书随硬件配置一同加载。因此，加载过程中需要停止 **CPU** 的运行。仅当所需的证书位于模块中时，才能在 **RUN** 模式下重新加载使用其它 OUC 安全连接的块。

6.11.6 通过电子邮件实现 OUC

在以下章节中，将介绍如何利用 TMAIL_C 通信指令与邮件服务器建立安全连接（基于 TLS 的 SNMP）。无法通过 CPU 的集成接口建立与邮件服务器的安全连接。仅可通过 CP 的接口建立此类安全连接。

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、TMAIL_V6_SEC 或 TMAIL_QDN_SEC 系统数据类型的数据块，分配相应参数并在 TMAIL_C 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。
- 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
- 已经为 CP（TLS 客户端）分配了邮件服务器（TLS 服务器）的所有 CA 证书，并已将其组态加载到 CPU 中。

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 HWIdentifier of the IE-interface to specify the connection reference / identifier
	ID	CONN_OUC	100	
	ConnectionType	Byte	16#20	type of connection 16#20=32=TMail_V4 or TMail_V4_SECURE
	ActiveEstablished	Bool	true	active / passive connection establishment
	WatchDogTime	Time	T#5000ms	watchdog time to monitor SMTP server association
	▼ 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 mail server
	PassWord	String[254]	'myPW'	user password which is necessary to login into the mail server
	▼ 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 address
	RemotePort	UInt	587	remote TCP port number
	ActivateSecureConn	Bool	TRUE	activate the security functionality of that connection
	ExtTLSCapabilities	Byte	16#0	for further capability extensions of the TLS handshake
	TLSServerCertRef	UDInt	7	Reference to the X.509 V3 (CA-) certificate of the mail server

图 6-33 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。
- 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。

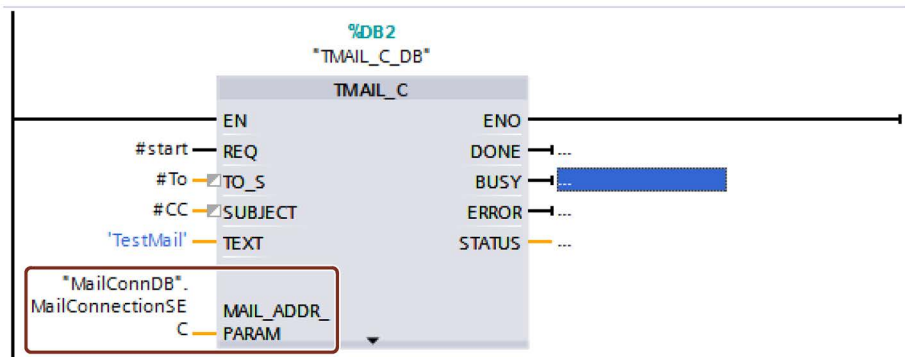


图 6-34 TMAIL_C 指令

更多信息

有关 TMail_V4_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“基本知识 (页 74)”部分。

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 7-1 “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。

- 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

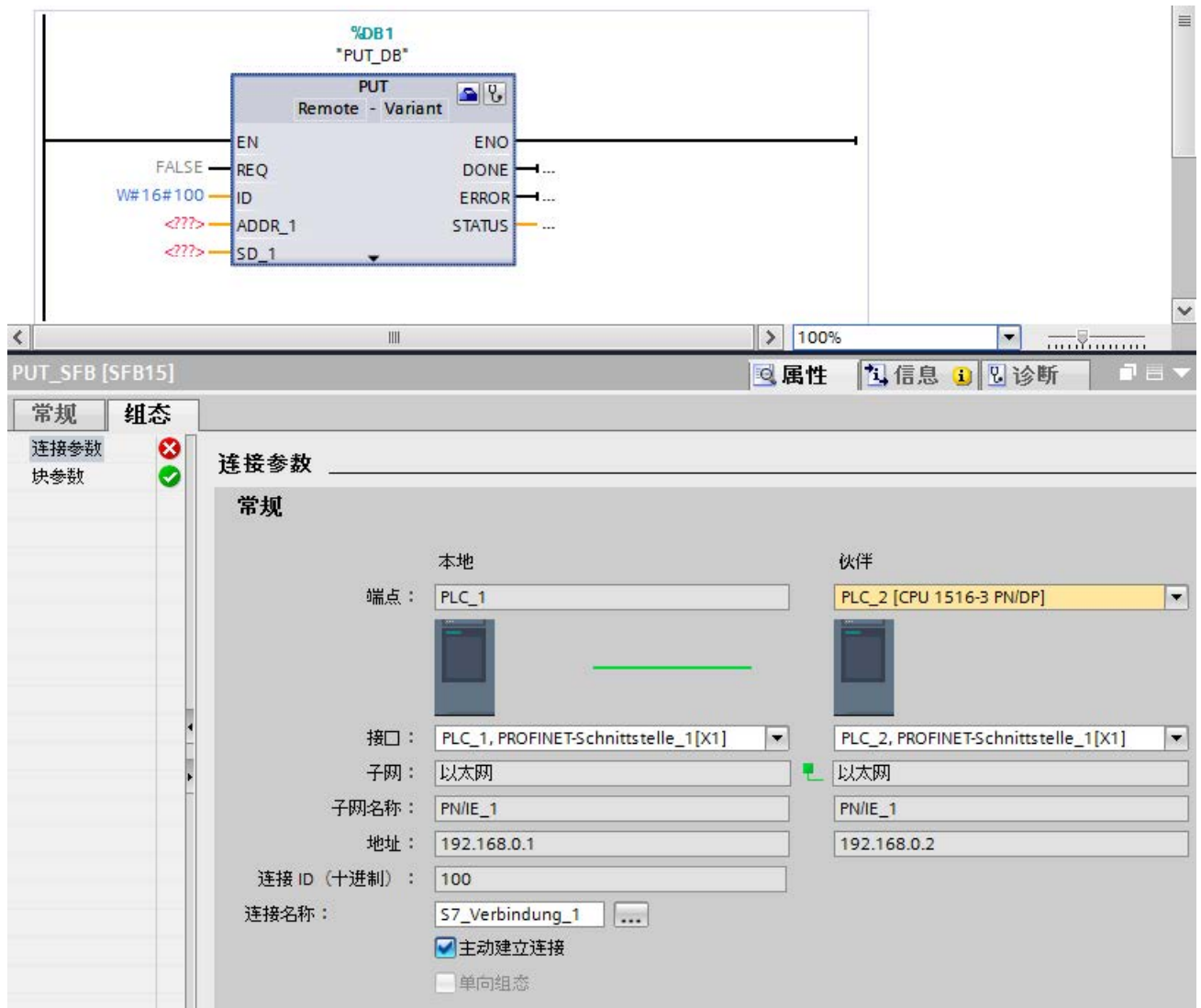


图 7-2 PUT 指令的连接组态

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 (RFC1006)：在 S7-1500 CPU 间进行 S7 通信
- 禁用“TCP/IP”复选框：ISO 协议 (IEC8073)：使用 MAC 地址进行寻址

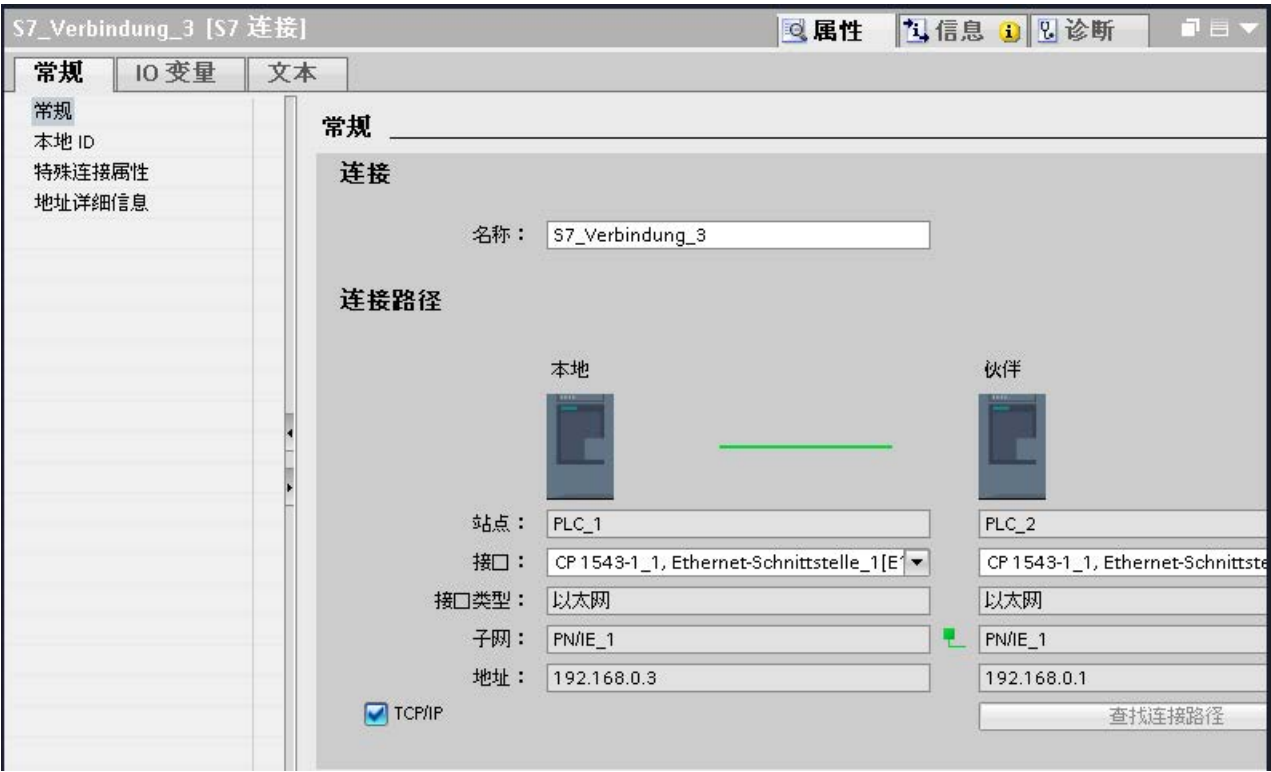


图 7-3 选择 CP 1543-1 的传输协议

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 195）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。

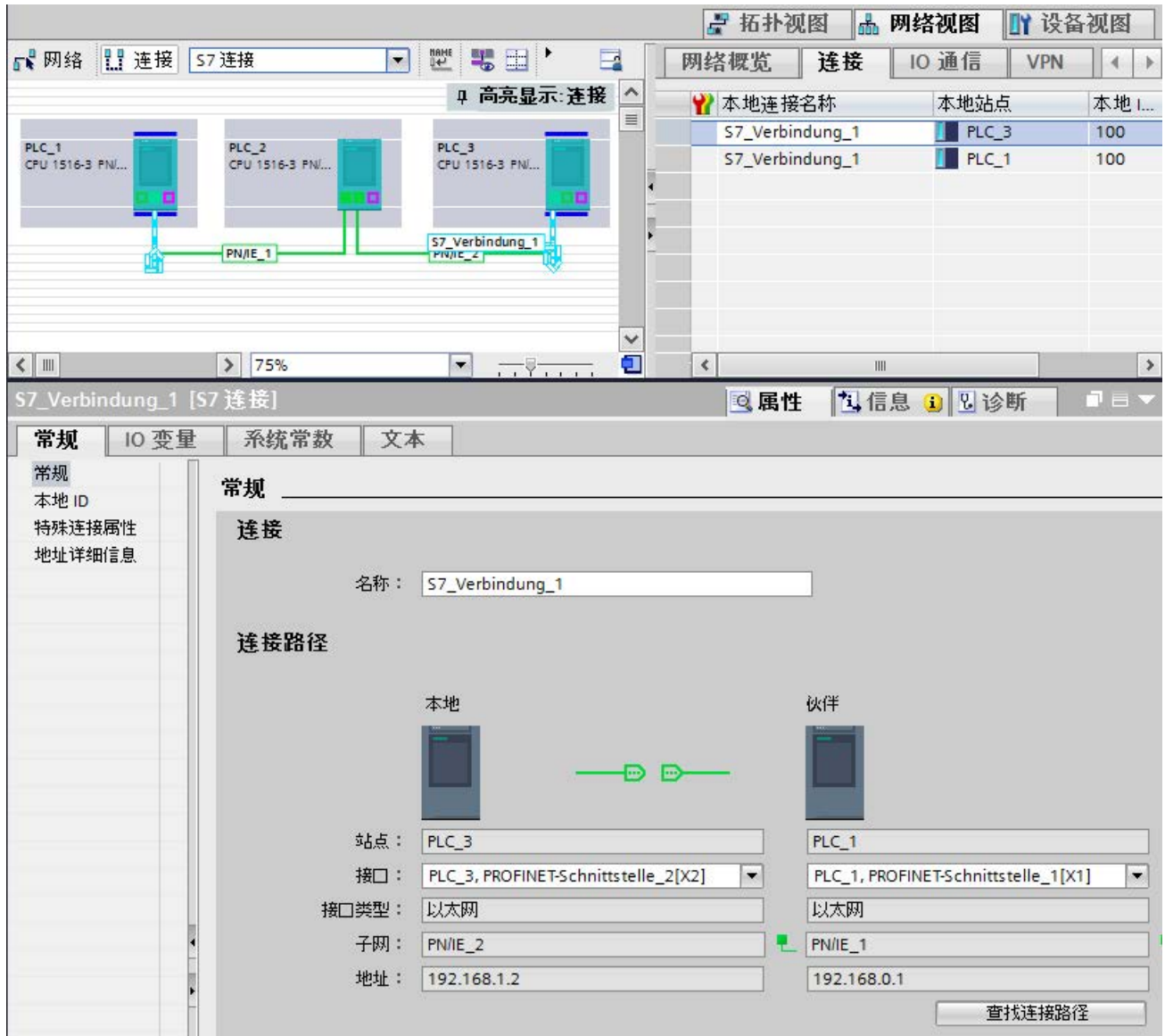


图 7-4 通过不同子网进行 S7 连接

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。

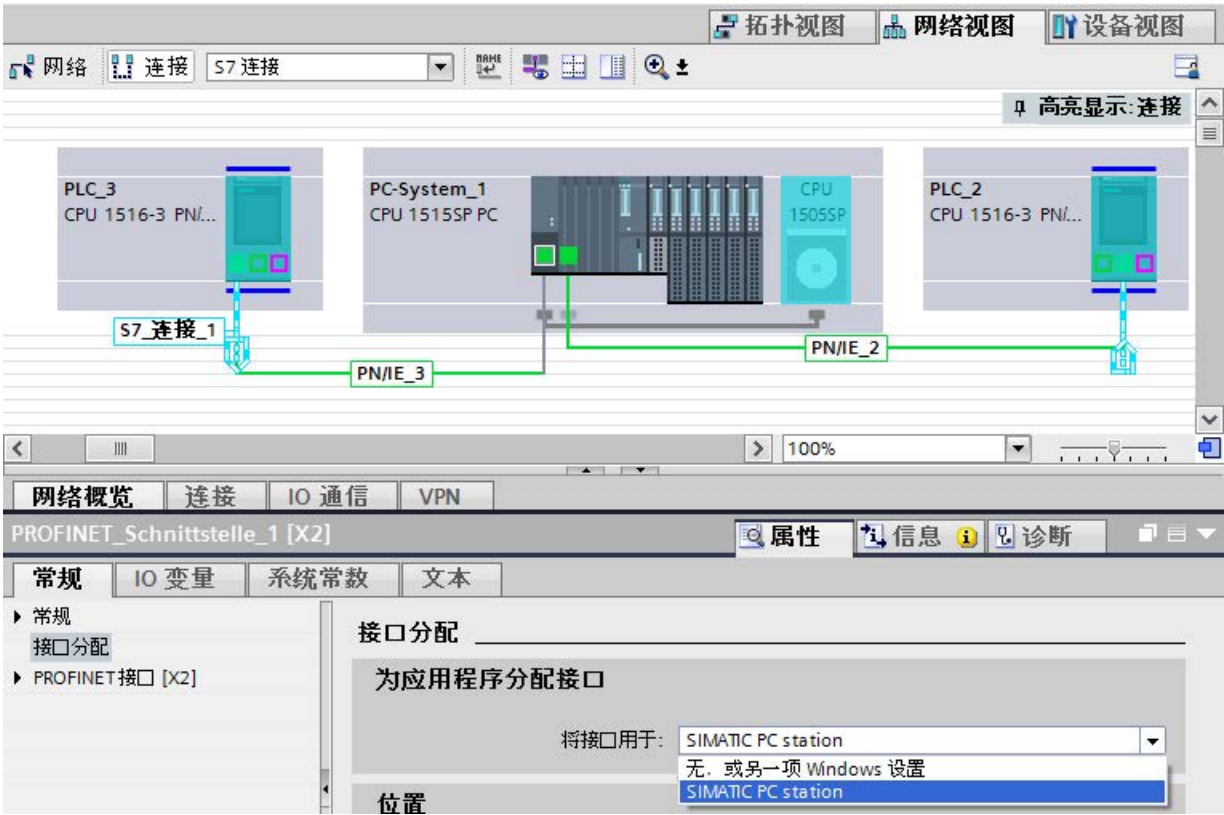


图 7-5 S7 路由 PC 站

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

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 (<http://support.automation.siemens.com/WW/view/zh/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 服务器

9.1 需了解的 OPC UA 知识

9.1.1 OPC UA 和工业 4.0

数据交换的统一标准

工业 4.0 表示在企业级 IT 系统中对生产数据进行集中利用、评估和分析。

在 PLC 程序中，如今已记录大量的生产和过程数据（压力值、温度和计数器读数）并使其可用于企业级系统，例如，用于提高产品质量。

借助工业 4.0，生产与企业级之间的数据交换将在未来实现更快速的增长。

然而，成功实现工业 4.0 的前提条件是采用统一的数据交换标准。

OPC UA（统一架构）标准尤其适合于跨不同层级的数据交换，这是因为其独立于特定的操作系统，采用安全的传递程序以及数据的语义描述。

OPC UA 不仅提供数据，而且提供有关数据（数据类型）的信息，因而可对数据进行机器解析。

9.1.2 说明的结构

说明应用

本说明适用于 S7-1500 CPU 的 OPC UA 服务器用户。

由一个 OPC UA 系统（服务器）提供数据，其它系统（客户端）可使用这些数据。

因此，必须使用 OPC UA 客户端来访问 S7-1500 的数据。

在演示 OPC UA 客户端的操作过程中，将使用以下 OPC UA 客户端：

- Unified Automation 的“UaExpert”。测试客户端，可免费使用。

Unified Automation (<https://www.unified-automation.com>)。

- OPC 基金会的“UA Sample Client”。在 OPC 基金会注册的用户可免费使用此客户端。

OPC 基金会 (<https://opcfoundation.org>)。

- SIEMENS 示例程序“UaClient”。

该应用程序免费提供，接收了如何建立服务器与客户端之间的连接以及如何读写 PLC 变量等，详情请参见“对 OPC UA 客户端进行编程 (页 183)”部分。在后续章节中，也将使用该程序说明 OPC UA 客户端/服务器的操作模式/理念。

有关该示例程序的源代码，请参见本手册的附录部分。

要先组态 S7-1500 的服务器，客户端才能访问 S7-1500 的数据。具体操作步骤，请参见“组态 S7-1500 的 OPC UA 服务器”部分。

由于安全性是 OPC UA 的决定性因素，因此在该说明中用大量篇幅介绍了安全性相关的主题。

在下一章中，将以关键点形式列示 OPC UA 的重要特性。

应用示例

有关应用示例和客户端 API，敬请访问“Siemens 工业在线支持”。基于该接口的功能，.NET 程序员可访问 S7-1500 的 OPC UA 服务器。客户端 API 基于 OPC Foundation 的 .NET OPC UA 栈。该应用示例和 API 可免费使用，可在此处 (<http://support.automation.siemens.com/WW/view/zh/109737901>) 下载。

9.1.3 OPC UA 的特性

显著特性

该新标准具有以下特性：

- OPC UA 与特定的操作系统平台无关。

OPC UA 支持 Window、Linux、Apple OS X、实时操作系统或移动操作系统（如 Android）。

- OPC UA 支持多种编程语言。

OPC 基金会已经推出了多种编程语言的 OPC UA 标准：支持 ANSI C、.NET 和 Java 的协议栈。

- OPC 基金会以开源软件的形式提供了 Java 和 .Net 协议栈以及示例程序。请参见“Github (<https://github.com/opcfoundation>)”。

- 多家公司提供了包含 OPC 基金会协议栈及附加功能的软件开发工具包 (SDK)。

这些开发包可大幅减少 OPC UA 应用的开发工作量。

- 可扩展性：

OPC UA 可应用于传感器、嵌入式系统、控制器、PC 系统与智能电话，以及运行 MES 和 ERP 应用程序的服务器中。

- OPC UA 支持客户端/服务器机制：

OPC UA 服务器提供网络信息。

OPC UA 客户端访问这些信息。

- 集成信息安全机制

OPC UA 可在各种不同层级中使用信息安全机制：

- 仅当 OPC UA 客户端和 OPC UA 服务器均通过证书 (X.509 V3) 进行身份验证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）。

另外，服务器与客户端之间也可建立非安全连接（“无”），具体取决于组态设置。

- 用户访问信息时，服务器可能要求用户验证。认证方式可以为证书（在 STEP 7 中不可组态）、用户名称/密码进行，也可能无需认证（guest 访问），具体取决于组态设置。

这些安全机制是可选的并且可以组态。

- OPC UA 与特定的传输层无关。

OPC UA 当前支持以下传输机制：

- 基于 TCP/IP，直接将消息作为二进制流传送
- 采用 XML（通过 TCP/IP 和 HTTP；仅可实现慢速传输，因此很少使用；无法在 STEP 7 中组态）传输消息

所有 OPC UA 应用程序均支持基于 TCP/IP 直接进行高速数据交换（OPC UA 技术规范要求）。

- PLC 变量映射

OPC UA 服务器中的信息（如 PLC 变量）可建模为节点，通过引用进行连接。这样，即可通过 OPC UA 客户端从一个节点浏览到另一个节点，查找可读取、监视或写入的内容。

- 元数据

OPC UA 服务器可提供大量信息。如，CPU 和 OPC UA 服务器信息、数据和数据类型等等。OPC UA 基于类型实例机制。无论是实例还是其类型定义，都在运行过程中使用。

9.1.4 从标准 OPC 接口到 OPC UA

标准接口

传统 OPC 仅支持 Windows 操作系统。

为了打破这一限制，OPC 基金会制定了 OPC UA 标准。

本标准无平台限制，并采用应用广泛的标准协议 (TCP/IP)。

因此，支持不同系统间的数据交换，例如：

- 控制器与 MES / ERP 系统之间
- 西门子控制器与其它制造商的控制器之间
- 智能电话与控制器之间
- 嵌入式系统与控制器之间
- 智能传感器与控制器之间

9.1.5 CPU S7-1500 的 OPC UA 服务器

CPU 1500 的 OPC UA 服务器

固件版本为 V2.0 及更高版本的 S7-1500 CPU 均配备 OPC UA 服务器。

“S7-1500 CPU”是指 CPU 变体 S7-1500F、S7-1500T、S7-1500C 及 S7-1500pro CPU、ET 200SP CPU 以及 SIMATIC S7-1500 软件控制器。

通过 CPU 1500 上所有集成的 PROFINET 接口，可访问该 CPU 中的 OPC UA 服务器。
例外：使用 SIMATIC S7-1500 软件控制器时，只能通过软件 PLC 所用的接口对 OPC UA 服务器进行访问。

固件版本为 V2.0 时，不能通过 CP 或 CM 访问。

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“组态对 PLC 变量的访问 (页 161)”）。

这些节点将连接成一个网络。

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，可读取、监视和写入 PLC 程序中的变量值。

9.1.6 OPC UA 服务器的地址空间

不同节点类别

OPC UA 服务器可采用节点形式提供信息。

节点可以是一个对象、变量，甚至方法。

以下示例显示了 S7-1500 CPU 的 OPC UA 服务器的地址空间（摘自 Unified Automation 的“UaExpert”）。

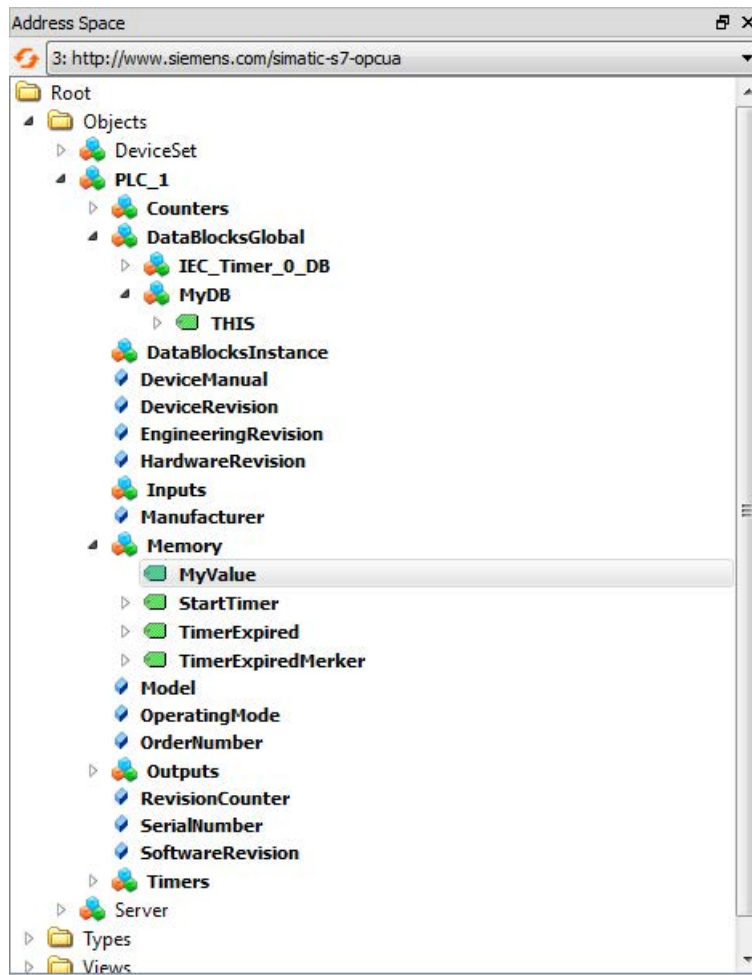


图 9-1 S7-1500 CPU 的 OPC UA 服务器的地址空间

上述示例中的下列节点属于“Object”节点类：“Objects”、“DeviceSet”、“PLC_1”。

而下列节点属于“Tag”节点类：“MyDB”、“MyValue”、“StartTimer”下的“THIS”。

节点中包含大量属性。如，“节点 ID”(NodeIdD)、“显示名称”(BrowseName) 和“说明”(Description)。

变量中还包含有属性“值”(Value)，用于保存 PLC 变量的值。

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。该网络可以是树形结构但并非必需。

节点网络也可称为地址空间。

可从根节点开始访问 S7-1500 CPU 的 OPC UA 服务器地址空间中的所有节点。

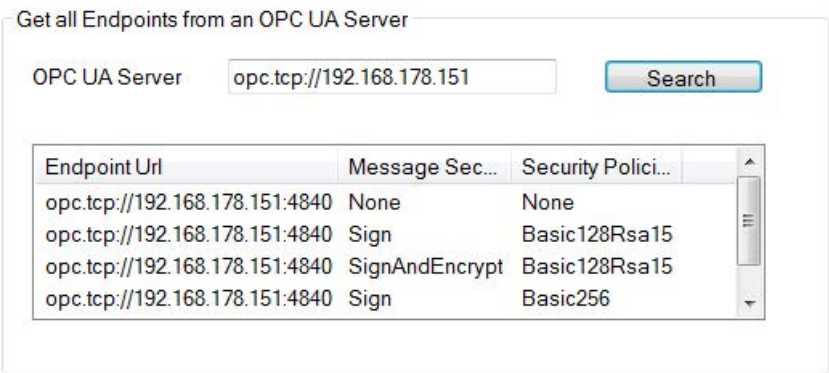
9.1.7 OPC UA 服务器的端点

各种安全策略

建立安全连接之前，OPC UA 客户端将询问服务器采用何种安全策略建立连接。

服务器则返回所有的端点以及服务器可提供的相关安全策略列表。

在以下示例中，显示了 S7-1500 CPU 中的 OPC UA 服务器端（摘自西门子示例程序“UaClient”；参见“对 OPC UA 客户端进行编程 (页 183)”部分）：



Get all Endpoints from an OPC UA Server		
OPC UA Server	opc.tcp://192.168.178.151	Search
Endpoint Url	Message Sec...	Security Polici...
opc.tcp://192.168.178.151:4840	None	None
opc.tcp://192.168.178.151:4840	Sign	Basic128Rsa15
opc.tcp://192.168.178.151:4840	SignAndEncrypt	Basic128Rsa15
opc.tcp://192.168.178.151:4840	Sign	Basic256

图 9-2 SIEMENS 示例程序“UaClient”中 S7-1500 CPU 的 OPC UA 服务器端点

有关组态 S7-1500 端点的信息，请参见“处理服务器和客户端证书 (页 175)”部分。

端点结构

端点由以下元素组成：

- 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（默认端口）

端口号可以配置，详情请参见“OPC UA 服务器的设置 (页 168)”部分。

- 消息的安全设置（消息安全模式）：无、签名、SignAndEncrypt。
- 加密和哈希程序（安全策略）：无、Basic128Rsa15、Basic256、Basic256Sha256（本示例中）。

下图所示为 OPC 基金会的程序“UA Sample Client”。

客户端已建立与 S7-1500 CPU 的 OPC UA 服务器端点的安全连接

“opc.tcp://192.168.178.151:4840 - [SignAndEncrypt:Basic 128Rsa 15:二进制”。

安全策略“SignAndEncrypt”：Basic 128Rsa 15”包含在端点中。

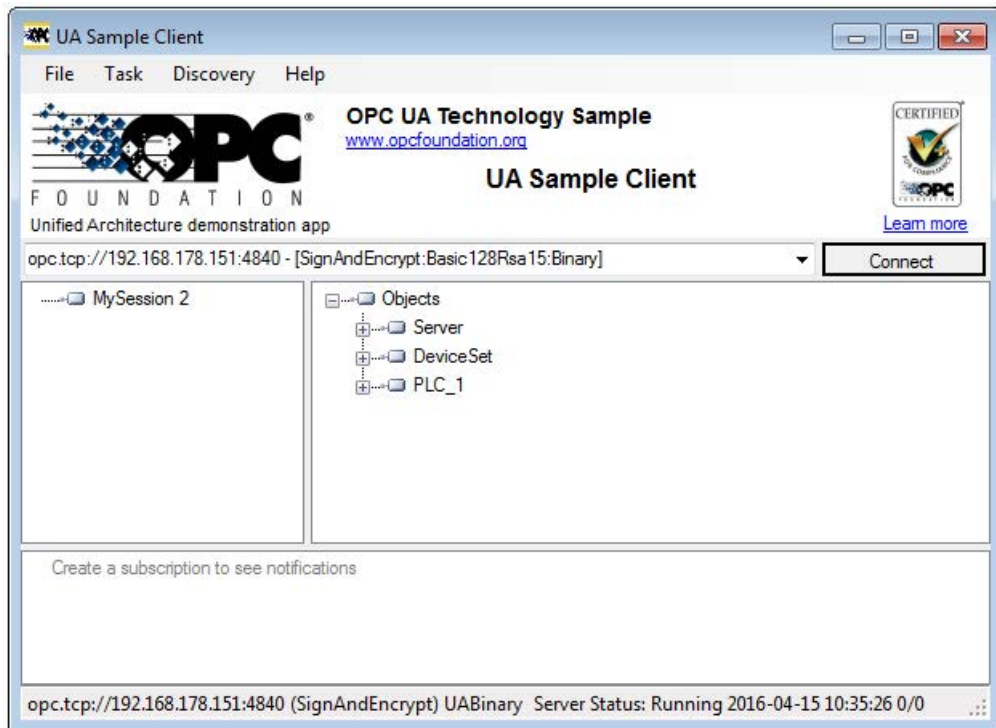


图 9-3 摘自示例程序“UA Sample Client”

仅当 OPC UA 客户端满足服务器端的安全政策时，才建立与该服务器端的连接。

OPC UA 服务器提供哪些信息？

OPC UA 服务器可提供大量数据：

- 客户端可访问的 PLC/DB 变量值。
- 这些 PLC/DB 变量的数据类型。
- 有关 OPC UA 服务器本身和 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。

例如，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）。

如何访问这些信息？

可通过以下方式进行设置：

- 在线：在 OPC UA 服务器运行时，可以显示所有可用的信息。为此，需导航（浏览）该服务器的地址空间。
- 离线：可从 STEP 7 V14 中导出基于 OPC 基金会的 XML 架构的 XML 文件。
- 离线并使用 STEP 7 V14 的开放性 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 的开放性”）。
- 如果用户熟知相关语法和 PLC 编程，则可直接访问 OPC UA 服务器，而无需先了解相关信息。

9.1.8 节点寻址

Namespace 和 NodeId

NodeId 可唯一地标识 OPC UA 地址空间中的节点。

NodeId 由一个标识符和一个命名空间索引构成。

命名空间用于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。

此外，OPC 基金会已定义有基本数据和变量类型。

这些节点位于 OPC 基金会的命名空间中且索引为 0。

西门子命名空间和标识符

命名空间 "http://www.siemens.com/simatic-s7-opcua" 已分配给 SIMATIC S7-1500。

SIEMENS 命名空间的当前索引为 3。

然而，如果向服务器插入其它命名空间或删除现有的命名空间，索引会发生变化。

因此，读取或写入值之前，需要从服务器上查询名称空间的当前索引。

下图显示了此类查询的结果：此处以 SIEMENS 的“UaClient”程序为例。

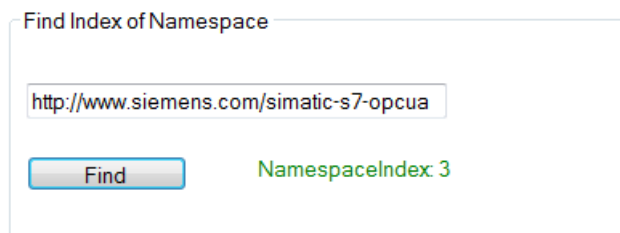


图 9-4 示例程序“UaClient”中的命名空间索引查询

标识符

标识符对应于引号内的 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。

下列示例展示了如何读取“StartTimer”变量的值：

Index	Boolean Variable		Result
3	"StartTimer"	Read	True

图 9-5 示例：读取布尔变量“StartTimer”

标识符可能包含多个组成部分。各组成部分以句点分隔。

下列示例展示了如何读取数组数据块“MyDB”的内容。

此数据块包含带有十个整数值的数组。

可同时读取所有十个值。因此，在数组范围内输入“0:9”id。

Index	Array Datablock of Int16		Results										
3	"MyDB"."THIS"	Read	<table><tr><th>Index</th><th>Values</th></tr><tr><td>0</td><td>7050</td></tr><tr><td>1</td><td>7051</td></tr><tr><td>2</td><td>7052</td></tr><tr><td>3</td><td>7053</td></tr></table>	Index	Values	0	7050	1	7051	2	7052	3	7053
Index	Values												
0	7050												
1	7051												
2	7052												
3	7053												
	Array Range (for instance 0:9)												
	0:9												

图 9-6 示例：读取数组数据块“MyDB”

位于 OPC UA 服务器的地址范围中的 PLC 变量

下图显示了示例中的 PLC 变量在 OPC UA 服务器的地址空间中所处的位置（摘自 UaClient）：

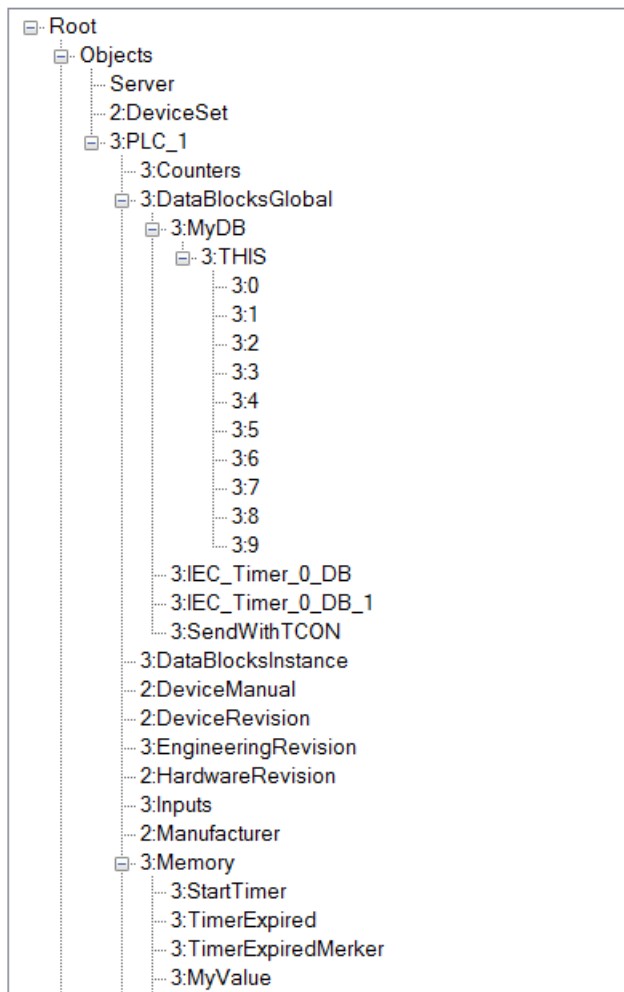


图 9-7 示例：OPC UA 服务器的地址空间中的 PLC 变量

数据块“MyDB”是一个全局数据块。因此，该数据块位于“DataBlocksGlobal”节点下。

“StartTimer”是一个内存变量，因此存储在“Memory”节点下。

9.1.9 OPC UA 客户端

从 S7-1500 中读取数据

OPC UA 客户端是一种程序，用于从 S7-1500 的 OPC UA 服务器中读取数据并将数据写入该服务器。

然而，该程序仅可读写已针对此目的启用的数据（请参见“管理读写权限 (页 161)”）。

要建立与 OPC UA 服务器之间的连接，需要使用服务器的端点（请参见“OPC UA 服务器的端点 (页 137)”部分）。

服务器上存在哪些数据？

如果存在与服务器端点的连接，则可以使用客户端的导航功能：可以从定义的起始点开始浏览（从“根”节点）服务器的地址空间。

除其它信息外，还提供了以下信息：

- 启用的 PLC 变量、数据块和数据块变量
- 上述 PLC 变量、数据块及数据块变量的命名空间索引和标识符
- PLC 变量和 DB 变量的数据类型
- 数组中的元素数（数组读写操作需要此信息）
- 有关 OPC UA 服务器自身的信息以及基于 OPC UA 的 S7-1500 的有关 OPC 基金会设备标准的信息（例如，序列号、固件版本）

从服务器中读取数据

现在了解了 PLC 变量的命名空间索引、标识符和数据类型。这样就能够专门读取各个 PLC 变量和 DB 变量以及完整的数组数据块和结构。

有关从 S7-1500 CPU 的 OPC UA 服务器中读取布尔变量和数组数据块的示例，请参见“节点寻址 (页 140)”部分。

将数据写入服务器

借助于浏览服务器的地址空间时获取的信息（索引、标识符和数据类型），还可以通过 OPC UA 客户端将值传输到 S7-1500。

以下示例将展示如何覆盖数组数据块“MyDB”（摘自 UaClient）的前三个值：

Index	Array Datablock of Int16	Values	Write	Status Code
3	"MyDB"."THIS"	1 2 3		
	Array Range (for instance 0:9) 0:2			

图 9-8 示例：向数组数据块“MyDB”写入值

可以在“数组范围”(Array range) 中指定要覆盖的的数组元素。

状态代码“良好”(Good) 表示已成功传输数值。

然而，仅可向 S7-1500 写入值，而不能写入这些值的时间戳。时间戳仅可读取。

通过注册加快访问速度

到目前为止，示例使用字符串作为标识符，例如，“MyBD2”、“THIS”。

然而，如果使用内部地址而不是字符串作为标识符，则访问速度要快得多。

如果定期访问特定的变量，则应使用函数“RegisteredRead”和“RegisteredWrite”：

在此过程中，客户端首先要在服务器中注册 PLC 变量。服务器将返回客户端在实际访问中所用的标识符。

该标识符仅对当前会话有效，因此会话断开/丢失后，必须再次发出请求。

在以下示例中，首先注册了服务器的“StartTimer”变量。随后，使用快速函数“RegisteredWrite”设置值。

Index	Boolean Variable	Register	Value	Write	Status Code
3	"StartTimer"	✓	True		
					Unregister

图 9-9 示例：注册后写入布尔变量“StartTimer”

“RegisteredRead”函数可按照相同的模式使用，在反复读出数据时特别实用。然而，考虑到具体应用，可能最好使用订阅来代替。

订阅

术语“订阅”一词用于仅传输 OPC UA 客户端在 OPC UA 服务器中已注册的那些变量的函数。变量值更改后，OPC UA 服务器仅针对这些已注册变量（订阅）向 OPC UA 客户端发送一条消息。监控这些变量可使 OPC UA 客户端的固定采样变得多余；因此会降低网络负荷。

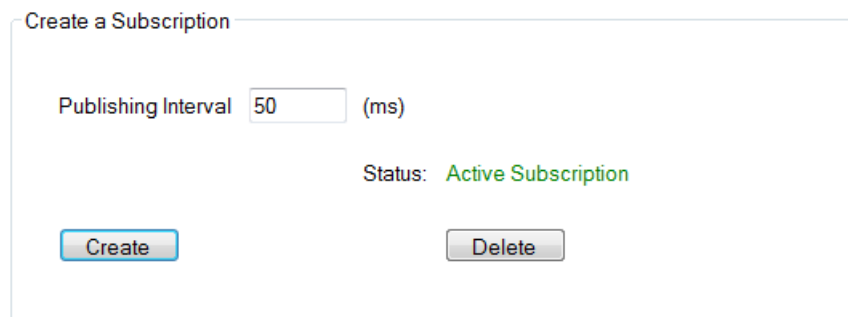
要使用此功能，必须首先创建一个订阅。

为此，要指定 UaClient 的“发布时间间隔”(Publishing Interval)（发送时间间隔），然后单击“创建”(Create) 按钮。

“发布时间间隔”(Publishing Interval) 是指将新值发送到客户端的时间间隔。

为以下示例创建一个订阅：值发生更改后，客户端每 50 毫秒接收一条包含新值（发布时间间隔为 50 毫秒）的消息。

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，确保不提供客户端要求的极短发送时间间隔。具体消息，请参见“OPC UA 服务器的设置 (页 168)”。



The image shows a software interface titled "Create a Subscription". It contains a text label "Publishing Interval" followed by a text input field containing the number "50" and the unit "(ms)". Below this, the status is displayed as "Status: Active Subscription" in green text. At the bottom, there are two buttons: "Create" and "Delete".

图 9-10 创建订阅

监控哪些 PLC 变量？

创建订阅后，可指定要监控的服务器变量。

在以下示例中，将“Voltage”变量添加到订阅中。

Index	LREAL Variable	Sampling Interval		Value
3	"Voltage"	-1	Add and Monitor	2.21426504
Queue Size	Deadband			
1	0.1			

图 9-11 示例：“Voltage”变量已添加到订阅中

“Voltage”变量包含由 S7-1500 CPU 检测到的电压变量（通过 2 线模拟量输入模块 AI 4xU/I）。

采样时间间隔包含负值 (-1)。这表示 OPC UA 服务器将以最小的时间间隔检查值的变更。

在本示例中，队列长度将设置为值“1”：始终以 50 毫秒的时间间隔从 CPU 上只读取一个值，并且如果值发生更改，随后会将该值发送到 OPC UA 客户端。

在本例中，“死区”参数相当于“0.1”：值变化必须至少等于 0.1 伏；只有这样，服务器才会将新值发送到客户端。更小的值变化则不会发送。

例如，可以使用此参数抑制信号噪声：过程变量中的微小变化不具备任何实质意义。

9.2 OPC UA 的安全性

9.2.1 信息安全

在本章节中，介绍了 OPC UA 服务器与客户端之间，数字证书与加密连接的安全问题。

指定项目中，需要哪种安全等级？

什么是数字证书？

建立安全连接时，证书有何作用？

9.2.2 请勿禁用安全策略

寻址风险

OPC UA 支持过程和生产层级的不同系统之间、以及这些系统与控制和企业层级的系统之间进行数据交换。

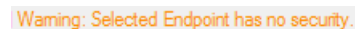
这种功能同时也带来了安全风险。正因为此，OPC UA 采用了一系列安全防护机制：

- OPC UA 服务器和客户端的身份验证。
- 对用户身份进行验证。
- 在 OPC UA 服务器与客户端之间进行签名/加密的数据交换。

仅当必需时，才应在特定状况下禁用这些安全策略：

- 调试过程中
- 未连接外部以太网的单机项目。

例如，对于 OPC 基金会的“UA Sample Client”，如果为端点选择“无”(None)，则程序会发出明确警告：



Waming: Selected Endpoint has no security.

图 9-12 示例程序“UA Sample Client”发出的安全警告

与采用“无”(None) 的端点之间的连接是不安全的

S7-1500 CPU 的 OPC UA 服务器将其公共证书发送至也采用“无”(None) 的客户端。有些客户会验证此证书。但客户端并不向服务器发送证书。由于选择“无”(None)，客户端的身份仍然未知。

服务器与客户端之间的数据通信并不安全（未签名也未加密）。对于端点“无”(None) 的连接，可以通过使用网络分析工具 Wireshark（用于记录并分析网络中的数据通信）记录通信流量。

规则

- 仅将采用“无”(None) 的端点视为例外。
- 仅将用户的“访客身份验证”视为例外。
- 如果确实有必要，则仅允许通过 OPC UA 访问 PLC 变量和 DB 变量。

9.2.3 ITU X.509 证书

数字证书

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。

只有 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“组态 S7-1500 的 OPC UA 服务器 (页 165)”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。

客户端和服务器的相互身份验证可有效防止中间人攻击。

中间人攻击

“中间人”可能处于服务器和客户端间的连接中。中间人是一种应用程序，用于侦听服务器与客户端的通信并声称自己为客户端或服务器，以获取 S7 程序相关信息或在 CPU 中设置值，进而对设备或工厂实施攻击。

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。

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。

证书签名

要检查证书是否篡改，则需对证书进行签名。

可通过以下两种操作步骤：

- 联系一家证书颁发机构 (CA) 并对证书进行签名。

此时，证书颁发机构将验证您的身份，并通过该证书颁发机构的私钥对您的证书进行签名。

为此，向证书颁发机构发送 CSR（证书签名请求）。

利用 OpenSSL 工具自行生成 CSR 的步骤在“此处 (页 154)”介绍。

- 自行创建证书并签名。

为此，可使用 OPC 基金会的“Opc.Ua.CertificateGenerator”程序。该步骤在此处 (页 152) 进行介绍。

或使用 OpenSSL：具体信息，请参见“用户自己生成 PKI 密钥对和证书 (页 154)”。

签名

用户不仅可对证书进行签名，还可对 Word 文件、PDF 文件以及源代码等进行签名。

首先，发送方根据纯文本信息（纯文本消息）生成 HASH 值。

之后，再使用发送方的私钥对该 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：传送的证书、Word 文件或 PDF 文件未篡改。
- 如果两个 HASH 值不匹配，则表示从发送方接收到的纯文本消息已发生更改：传送的证书、Word 文件或 PDF 文件在传送过程中被篡改或受损。

通过该签名，可验证消息的完整性。

加密

用户可以对 Word、PDF 文件和源代码进行加密，防止非经授权的读取。

X 509 证书无法加密：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：仅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仅他们知晓的内部密钥，用于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。

9.2.4 OPC UA 证书

在本章节中，将介绍 OPC UA 如何使用 X509 证书。

OPC UA 的 X509 证书

OPC UA 可通过三不同类型的 X509 证书创建客户端到服务器的连接：

- OPC UA 应用程序证书

这类 X509 证书用于标识软件实例，即客户端或服务器的安装实例。

在“机构名称”(Organization name) 属性中，可输入该软件使用方的名称。

注：

对于安全设置“无”(None)，S7-1500 的 OPC UA 服务器也使用应用程序证书。

这确保能够与 OPC UA V1.1 及之前的版本兼容。

- OPC UA 软件证书

这类 X509 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（销售）方名称。

注：软件证书不得与 STEP 7 V14 一同使用。

- OPC UA 用户证书

此类 X509 证书用于标识特定的用户，如从 S7-1500 的 OPC UA 服务器访问过程数据的用户。

如果用户通过密码进行认证或组态为匿名访问，则无需使用该证书。

注：用户证书不得与 STEP 7 V14 一同使用。

此类证书属于最底层实体证书：它们用于识别人、组织、公司，软件的实例（安装）等。

参见

创建自签名证书 (页 152)

9.2.5 创建自签名证书

仅当使用本身无法创建客户端证书的 OPC UA 客户端时，才涉及本章内容。

在本节中，了解如何使用 OPC 基金会的“OPC.UA.CertificateGenerator”工具来生成客户端的自签名应用程序证书。

在以下章节中，将介绍如何使用“OPENSSL”工具实现此目的。

OPC 基金会的证书生成器

例如，可使用 OPC.UA.CertificateGenerator 生成自签名证书。

为此，请按以下步骤操作：

1. 从 OPC 基金会的网站上下载该工具：

例如，在 OPC 基金会网站 (<https://opcfoundation.org/developer-tools/developer-kits-unified-architecture>)“示例应用程序”(Sample Applications) 的“资源 > 示例/代码 > 统一架构”(Resources > Samples / Code > Unified Architecture) 下，提供了该程序。

2. 在计算机上安装 OPC 基金会的示例应用程序。
3. 使用 Windows 资源管理器中调用安装目录：目录位于“C:\Program Files (x86)\OPC Foundation\UA 1.02\Sample Applications”下。
4. 按住 [Shift] 键并在目录中单击鼠标右键，以便显示快捷菜单。
5. 选择“在此处打开命令提示符”(Open command prompt here)。
6. 在命令提示窗口的提示符后输入以下命令：

```
"Opc.Ua.CertificateGenerator -cmd issue -sp . -an MyClient"
```

7. 按下 Enter 键。

8. 该程序为“MyClient”生成以下内容：

- 在“证书”(certs) 子文件夹中，生成包含客户端公钥的“MyClient”证书
- 在“私钥”(private) 子文件夹中，生成客户端私钥。

“MyClient”仅仅是个示例。

让我们假定 OPC UA 客户端名为“SuperClient”。然后，在命令提示窗口中输入以下字符串：

“Opc.Ua.CertificateGenerator -cmd issue -sp . -an SuperClient”

下图显示了在命令提示窗口中输入的内容：

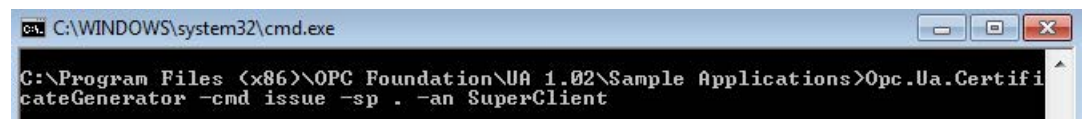


图 9-13 使用 OPC.Ua.CertificateGenerator 生成自签名客户端证书。

该描述基于 OPC 基金会 2015 年 6 月 25 日的“Opc.Ua.CertificateGenerator”。

针对不同的生成器版本，可能需要输入不同的内容。

要获取有关所需输入的信息，请按以下步骤操作：

1. 在命令提示窗口中，输入以下命令：“Opc.Ua.CertificateGenerator -?”

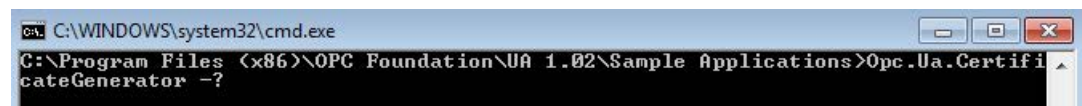


图 9-14 调用帮助功能

显示帮助。

2. 使用在“Create a self-signed Application Certificate”中输入的选项。

9.2.6 用户自己生成 PKI 密钥对和证书

仅当使用本身无法创建 PKI 密钥对和客户端证书的 OPC UA 客户端时，才涉及本章内容。

在本章节中，将介绍如何使用 OpenSSL 生成私钥、公钥和 X509 证书，并对该证书进行自签名。

使用 OpenSSL

要在 Windows 下使用 OpenSSL，请按以下步骤操作：

1. 在 Windows 系统中安装 OpenSSL。

如果操作系统为 64，则安装“C:\OpenSSL-Win64”目录中的 OpenSSL。

可以从开源软件的各种提供方下载 OpenSSL-Win64。

2. 创建一个目录，如“C:\demo”。

3. 打开命令行工具 (cmd.exe)。

单击“开始”(Start)，并在搜索字段中输入“cmd”。

在结果列表中，右键单击“cmd.exe”。以管理员身份执行程序。

Windows 的命令行窗口随即打开（DOS 提示符）。

4. 切换到“C:\demo”目录下。

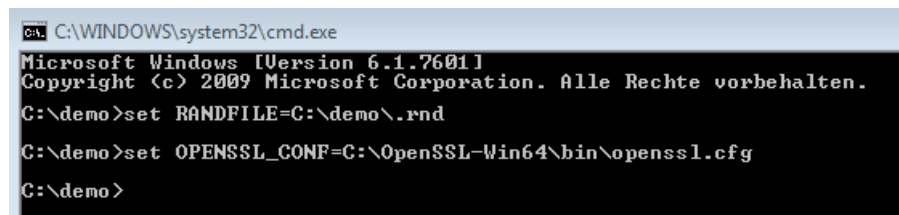
此时，可输入命令“cd C:\demo”。

5. 设置以下网络变量：

设置 RANDFILE=c:\demo\.rnd

设置 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：



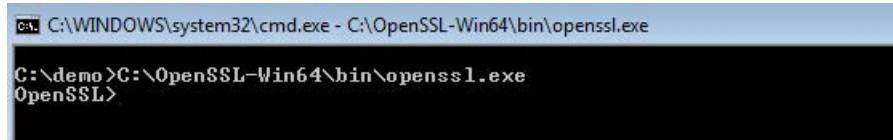
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.
C:\demo>set RANDFILE=C:\demo\.rnd
C:\demo>set OPENSSL_CONF=C:\OpenSSL-Win64\bin\openssl.cfg
C:\demo>
```

图 9-15 用户自己生成 PKI 密钥对和证书 (1)

6. 现在，可启动 OpenSSL。

如果 OpenSSL 安装在 C:\OpenSSL-Win64 目录中，则输入：C:\OpenSSL-Win64\bin\openssl.exe

下图显示了包含以下命令的命令行窗口：



```
C:\WINDOWS\system32\cmd.exe - C:\OpenSSL-Win64\bin\openssl.exe
C:\demo>C:\OpenSSL-Win64\bin\openssl.exe
OpenSSL>
```

图 9-16 用户自己生成 PKI 密钥对和证书 (2)

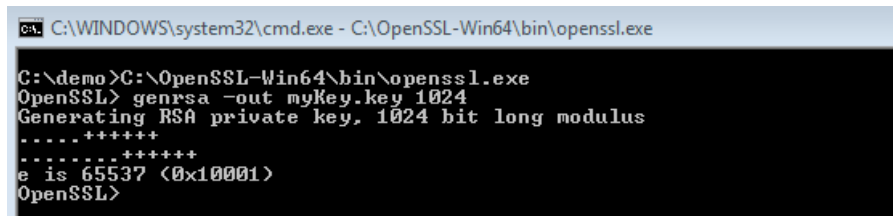
7. 生成一个私钥（该私钥也包含相应的公钥）。

该密钥应保存在“myKey.key”文件中。

密钥长度为 1024 位

输入命令：“genrsa -out myKey.key 1024”

下图显示了包含以下命令和 OpenSSL 的命令行窗口：



```
C:\WINDOWS\system32\cmd.exe - C:\OpenSSL-Win64\bin\openssl.exe
C:\demo>C:\OpenSSL-Win64\bin\openssl.exe
OpenSSL> genrsa -out myKey.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
OpenSSL>
```

图 9-17 用户自己生成 PKI 密钥对和证书 (3)

8. 生成一个 CSR（证书签名请求）。

为此，可输入命令：“req -new -key myKey.key -out myRequest.csr”。

在该命令的执行过程中，OpenSSL 将查询证书中的相关信息：

- 国家/地区名称：如，“DE”为德国，“FR”为法国
- 州或省名称：如，“Bavaria”。
- 位置名称：如，“Augsburg”
- 机构名称：在此，可输入公司的名称。
- 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：

输入的信息将添加到证书中。

下图显示了包含以下命令和 OpenSSL 的命令行窗口：

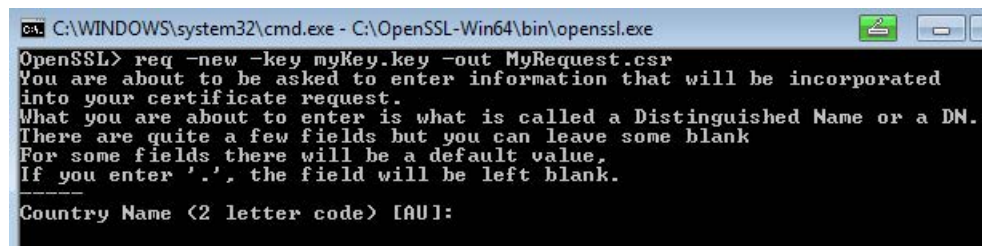


图 9-18 用户自己生成 PKI 密钥对和证书 (4)

该命令将在 C:\demo 目录中创建一个包含证书签名请求 (CSR) 的文件。在本示例中，为“myRequest.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)：

请遵循证书颁发机构的操作说明。

证书颁发机构 (CA) 将检查用户信息和身份，并使用该证书颁发机构的私钥对该证书进行签名。

在收到签名后的 X509 证书时，即可在诸如 OPC UA、HTTPS 或 SOUC（开放式用户安全通信）中使用该证书。

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）。

- 用户自签名 CSR

使用用户的私钥。

该选项将在下一个操作步骤中介绍。

9. 生成并签名自己的证书（自签名证书）

为此，可输入命令：“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：



```
C:\WINDOWS\system32\cmd.exe - C:\OpenSSL-Win64\bin\openssl.exe
OpenSSL> x509 -req -days 365 -in myRequest.csr -signkey myKey.key -out myCertificate.crt
Signature ok
subject=/C=DE/ST=Bayern/L=Augsburg/O=MyCompany/OU=IT/CN=MyName/emailAddress=MyEmail@me.de
Getting Private key
OpenSSL>
```

图 9-19 用户自己生成 PKI 密钥对和证书 (5)

该命令将生成一个 X509 证书，包含通过 CSR 传送的属性信息。在本示例中，为“myRequest.csr”。该命令还将使用私钥（在本示例中为“myKey.key”）对证书进行签名。

通信伙伴则使用公钥（包含在证书中）检查该证书是否真实有效。这样，可防止证书被攻击者篡改。

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。

9.2.7 安全地传输消息

使用 OPC UA 建立安全连接

OPC UA 在客户端与服务器间建立安全连接。OPC UA 检查通信伙伴的身份。

OPC UA 使用基于 ITU X509-V3 标准的证书对客户端和服务端进行认证。

例外情况：尚未建立采用“无”(None) 安全模式的安全连接。

消息安全模式

OPC UA 采用以下模式保护消息：

- 无
- 所有消息都不安全。
- 若要使用该安全模式，要建立与服务端“无”(None) 端点的连接。
- 签名
- 所有消息都将签名。系统将对所接收消息的完整性进行检查。检测篡改行为。
- 若要使用该安全模式，要建立与服务端的“签名”(Sign) 端点的连接。
- SignAndEncrypt
- 所有消息都将签名并加密。系统将对所接收消息的完整性进行检查。系统将检测篡改行为，攻击者无法读取消息内容（保护消息机密）。
- 若要使用该安全模式，要建立与服务端的 SignAndEncrypt 端点的连接。

所需层级

下图显示了建立安全连接时通常涉及三个层级：传输层、安全通道和会话。

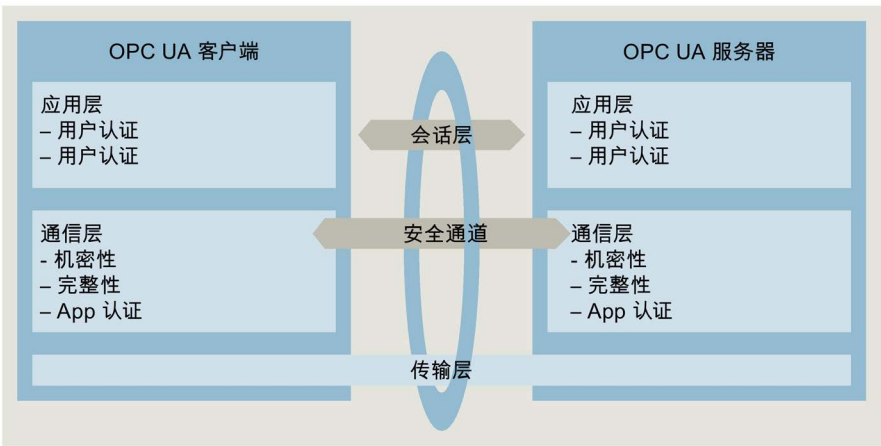


图 9-20 OPC UA 连接的层模型

- 传输层：

该层发送和接收消息。为此，OPC UA 使用标准以太网协议 TCP/IP 的扩展 (TCP UA)。

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至传输层。

在“签名”(Sign) 安全模式中，安全通道对发送的数据（消息）进行签名。接收消息时，安全通道将检查签名判断是否篡改。

在“签名并加密”(SignAndEncrypt) 安全模式中，安全通道对发送的数据进行签名并加密。安全通道对接收的数据进行解密并检查签名。

在“无”(None) 安全模式中，消息包将不做任何更改直接通过安全通道（消息将以纯文本形式接收和发送）。

- 会话层

会话层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

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”(SignAndEncrypt) 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）

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。

如果服务器信任该客户端的证书，则对消息进行解密并验证其签名（“签名并加密”）、仅检查签名（“签名”）或不对消息进行任何更改（“无”）。

之后，服务器向客户端发送一个响应（与请求的安全等级相同）。在响应中，还包含有服务器的机密信息。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

客户端向服务器发送 **CreateSessionRequest** 请求后，将启动会话建立过程。该消息中包含一个临时的随机数，仅一次性使用。服务器需要对这一临时的随机数进行签名，证明该随机数确实为该服务器建立安全通道时所用证书的私钥主体。

该消息（及所有后续消息）也将基于所选服务器端的安全策略（安全模式）进行加密。

之后，服务器将发送一个 **CreateSession** 响应。该消息中包含服务器的软件证书和已签名的临时随机数。

客户端将对签名的临时随机数进行检查。

如果服务器通过检查，则客户端将向服务器发送一个 **SessionActivate** 请求。该消息中包含有用户认证时所需的信息：

- 用户名和密码
- 或用户的 X509 证书（在 STEP 7 V14 中不使用）
- 或无数据（如果已组态匿名访问）。

如果用户具有相应权限，服务器将返回客户端一条消息（**ActivateSession** 响应），激活会话。

至此，OPC UA 客户端与服务器已成功建立安全连接。

9.3 组态对 PLC 变量的访问

9.3.1 管理读写权限

为 OPC UA 启用 PLC 变量和 DB 变量

如果为 OPC UA 启用了变量（默认设置），则 OPC UA 客户端可对 PLC 变量和 DB 变量执行读写访问。

如果启用了变量，则将激活“可从 HMI/OPC UA 访问”(Can be accessed from HMI/OPC UA) 复选框。

以下示例显示了数组数据块：

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 9-21 示例：为 OPC UA 启用的数组数据块

此数据块可通过 OPC UA 客户端在一次传递中完整读取（请参见“节点寻址 (页 140)”）。将激活数组所有元素的“可从 HMI/OPC UA 访问”(Accessible from HMI/OPC UA) 复选框。

此外，由于已选中“可从 HMI/OPC UA 写入”(Writable from HMI/OPC UA) 复选框，客户端可以为数组的所有元素指定新值。

在本例中，OPC UA 客户端具有对数组所有元素的读写权限。

结果：OPC UA 客户端可以对这些元素执行读写操作。

撤消写权限

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 客户端的写入权限。

结果：OPC UA 客户端只能进行读取访问。OPC UA 客户端无法为这些变量赋值，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 客户端无法对该变量进行读取和写入。

如果移除对结构中某元素的读或写权限，则该结构作为一个整体将无法再进行读写操作。

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“在 HMI 工程组态中可见”(Visible in HMI Engineering) 选项（未勾选），则无法在 TIA Portal WinCC 对该变量进行组态。该选项不会影响 OPC UA。

规则

- 如果需要与其它系统（控制器、嵌入式系统或 MES）进行通信，则只能在 STEP 7 中对 PLC 变量和数据块变量进行读访问。
而不应启用其它 PLC 变量。
- 如果确实需要访问某些特定的 PLC 变量和数据块变量，则只能通过 OPC UA 进行写访问。

9.3.2 更快速地访问数组和结构

高性能

OPC UA 设计用于在短时间内传输多个数据。

尤其在不访问单个 PLC 变量而是作为整体对数组和结构进行读写访问时，可以显著提高性能。

访问数组是最快的。因此，应将 OPC UA 客户端数据组合到数组中。

如果无法实现此操作，则应将数据放在结构中。

操作步骤

若要利用数组创建数据块（数组数据块），请按以下步骤操作：

1. 在项目树中选择 CPU 的 OPC UA 服务器。
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. 为数据块选择唯一名称。或接受已输入的名称。
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“数组限制”(Array limit) 中输入数组上限。
9. 单击“确定”(OK)。

9.3.3 导出带有已启用 PLC 变量的 XML 文件

生成 OPC UA 导出文件

对于在组态中为 OPC UA 启用的 PLC 变量和 DB 变量，可以采用 XML 文件的形式导出，以便于对 OPC UA 客户端进行离线工程组态。XML 文件基于 OPC 基金会的 XML 架构。

请按以下步骤操作：

1. 选择 CPU。

为此，单击 CPU 图标（例如，网络视图中的图标）。

2. 在 CPU 属性中，单击“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) 按钮：



图 9-22 在 STEP 7 中导出 OPC UA 文件

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)。

提示：在本手册随附的 ZIP 文件中，包含可将 XML 格式的变量和节点 ID 转换成 CSV 格式的文件转换器。然后，可获取可通过 OPC UA 访问的 CPU 变量列表。

9.4 组态 S7-1500 的 OPC UA 服务器

9.4.1 启用 OPC UA 服务器

要求

- 已获得操作 OPC UA 服务器的运行许可证，请参见“OPC UA 服务器的许可证 (页 182)”。
- 使用安全通信（如，HTTPS、安全 OUC、OPC UA）的证书时，需确保相应模块为当前时间和当前日期。否则，模块将所用的证书评估为无效，同时不进行安全通信。

调试 OPC UA 服务器

在基本设置中，基于安全原因，不会启用 CPU 的 OPC UA 服务器：OPC UA 客户端不具备对 S7-1500 执行读写访问的权限。

请按以下步骤操作，启用 CPU 的 OPC UA 服务器。

1. 选择 CPU。

为此，单击 CPU 图标（例如，网络视图中的图标）。

2. 在 CPU 属性中，单击“OPC UA > 服务器>”(OPC UA > Server)。

3. 激活 CPU 的 OPC UA 服务器。

4. 确认此安全信息。

保留设置

如果服务器已启用并进行了相应设置，此后再禁用服务器时，设置不会丢失。重新启用服务器后，这些设置会继续保存并可用。

应用程序名称

应用程序名称即 CPU 的 OPC UA 服务器的名称。该名称会显示在“OPC UA > 常规”(OPC UA > General) 下：

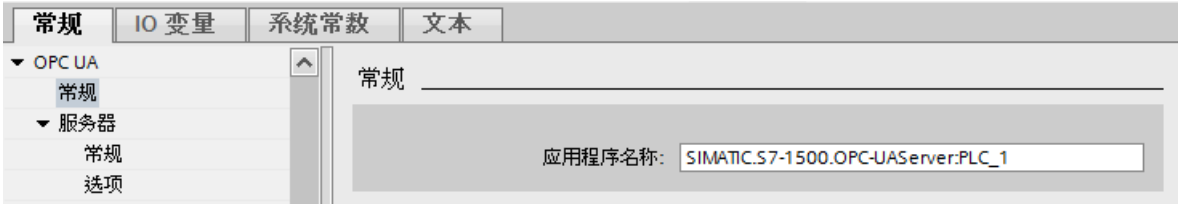


图 9-23 STEP 7 中 OPC UA 服务器的应用程序名称

服务器名称的默认设置为：“SIMATIC.S7-1500.OPC-UAServer:PLC1”。

默认设置包括“SIMATIC.S7-1500.OPC-UAServer:”和“常规 > 产品信息 > 名称”(General > Product information > Name) 中所选择的 CPU 名称（在本示例中，为“PLC_1”）。

客户端可将应用程序名称用作有意义的服务器名称。

以下示例来自 UaExpert:

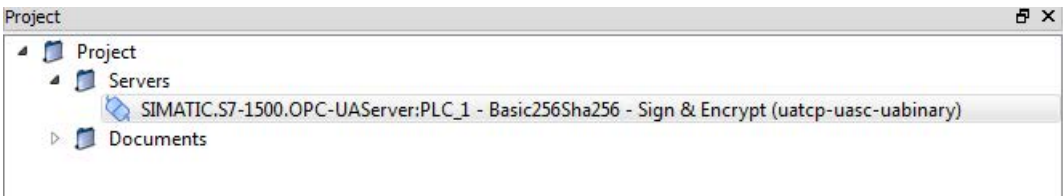


图 9-24 在 UaExpert 中显示的应用程序名称

如果已激活服务器，还可以使用在项目中有意义的其它名称。

更改应用程序名称

若要更改 OPC UA 服务器名称，请按下列步骤操作：

1. 选择 CPU。
2. 在 CPU 属性中，单击“OPC UA > 常规”(OPC UA > General)。
3. 输入一个有意义的名称。

9.4.2 访问 OPC UA 服务器

S7-1500 的 OPC UA 服务器通常可通过该 CPU 上的所有内部 PROFINET 接口进行访问（但不支持 CP/CM 的 PROFINET 接口）。

在本示例中，通过 URL（统一资源标识符）建立与 CPU 上 OPC UA 服务器的连接：

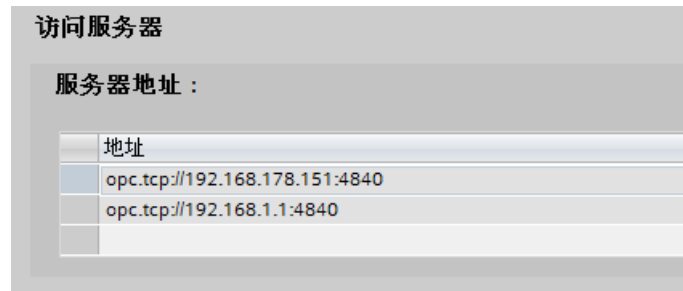


图 9-25 示例：STEP 7 中 OPC UA 服务器的 URL

URL 的结构如下所示：

- 协议标识符“opc.tcp://”
- IP 地址
 - 192.168.178.151
从 192.168.178 以太网子网可访问的 OPC UA 服务器的 IP 地址。
 - 192.168.1.1
从 192.168.1 以太网子网可访问的 OPC UA 服务器的 IP 地址。
- TCP 端口号
 - 默认值：4840（默认端口）
端口号可在“OPC UA 服务器 > 设置 > 端口”(OPC UA server > Settings > Port) 中更改。

动态 IP 地址

在以下示例中，未指定 PROFINET 接口 [X2] 的 IP 地址。

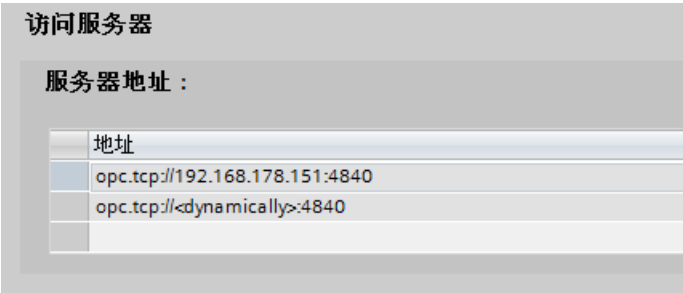


图 9-26 示例：采用动态 IP 地址的 URL

该表中显示有“<dynamically>”通配符。

之后，可在设备上设置该 PROFINET 接口的 IP 地址。

例如，通过 OPC UA 客户端搜索可用的 OPC UA 服务器时，查看已分配给该 PROFINET 接口 [X2] 的 IP 地址（TIA Portal 系统外）。此时，用户将接收到一个列表，显示可用的服务器及其 IP 地址。

9.4.3 OPC UA 服务器的设置

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。

创建连接是，OPC UA 客户端必须使用选定的端口（仅默认端口 4840 可忽略）。

在以下示例中，选择端口 48400：

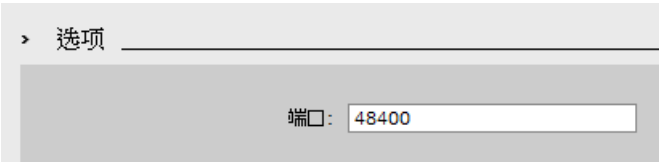


图 9-27 STEP 7 中 OPC UA 服务器的端口设置

使用订阅替代循环查询

对 PLC 变量的循环查询（轮询）可替换为对该值进行监控。这可以使用订阅来实现：PLC 变量的值发生变化时，服务器将通知客户端。请参见“OPC UA 客户端 (页 143)”部分。

一台服务器通常监控大量的 PLC 值。值发生变更时，服务器将向客户端发送包含 PLC 变量新值的消息（通知）。

服务器发送通知的频率

创建订阅时，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。

为了限制 OPC UA 的通信载荷，可设置需遵循的最短通知时间间隔。

最短发布时间间隔

在“最短发布时间间隔”(Minimum publishing interval) 中，可设置变量值发生变更时，服务器向客户端发送通知（含新值）的时间间隔。

在下图中，“最短发布时间间隔”(Minimum publishing interval) 设置为 200 ms。“最短采样时间间隔”为 100 ms。



图 9-28 STEP 7 中的“最短发布时间间隔”(Minimum publishing interval) 和“最短采样时间间隔”(Minimum sampling interval) 设置

在本示例中，创建订阅时，如果 OPC UA 客户端要求这样一个较快更新速度，则 OPC UA 服务器将每隔 200 ms 发送一条新的通知。

如果 OPC UA 客户端要求更新频率为 1000 ms，则 OPC UA 服务器将每隔一秒最多发送一个带有新值的通知。

如果客户端要求的更新频率为 100 ms，则服务器仍按照 200 ms 的频率发送消息（最短发布时间间隔）。

示例：

连接操作面板与 CPU。由于仅观察过程值而不会无规律地更改，因此，200 ms 的发布时间间隔足以满足画面刷新需求。

所用 CPU 支持的最短发布时间间隔为 10 ms。

操作面板的 OPC UA 客户端可设置为请求更快的更新速度（如，1 ms）。

此时，只能通过设置参数“最短发布时间间隔”(Minimum publishing interval) = 200 ms，降低通信载荷。

否则，会在发生值变更时，向 OPC UA 客户端发送大量不必要的通知消息。

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。

在以上示例中，设置的时间间隔为 100 ms。

与发布时间间隔时间类似，采样时间间隔由服务器和客户端协商。

在本示例中，OPC UA 客户端请求对特定 CPU 变量的采样速度为每隔 1000 ms，服务器将以 1000 ms 的时间间隔记录该变量的值。

而另一方面，本示例中 OPC UA 客户端请求对特定 CPU 变量的采样速度为每隔 10 ms 时，OPC UA 服务器将以 100 ms 的时间间隔记录该变量的值。在本示例中，100 ms 为最短采样时间间隔。

如果输入一个负值（如，-1）作为 OPC UA 客户请求的采样速度，OPC UA 服务器将使用设置为最短采样时间间隔的时间值（在本示例中，为 100 ms）。

采样时间间隔小于发布时间间隔

如果选定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值变更。

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。如果在发布间隔时间内发生的值变更次数过多且超过队列容量，则 OPC UA 服务器将覆盖最旧的值。并将最新的变更值发送到客户端。

9.4.4 通过 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 客户端和服务端”(OPC UA Client & Server)。

建议：如果要使用 OPC UA 服务器的所有安全功能，则需使用全局安全设置。

自定义服务器证书

激活 S7-1500 的 OPC UA 服务器时，STEP 7 将自动生成该服务器的证书（请参见“启用 OPC UA 服务器 (页 165)”）。

在此过程中，STEP 7 使用证书参数的默认值。

若要修改参数，请按如下步骤操作：

1. 单击 CPU 属性中的“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（三个点的按钮）。

将显示一个对话框，其中列出本地可用的证书。

2. 单击“添加”(Add)。

3. 将显示生成新证书的对话框（如下图所示）。

已输入值的情况如下：

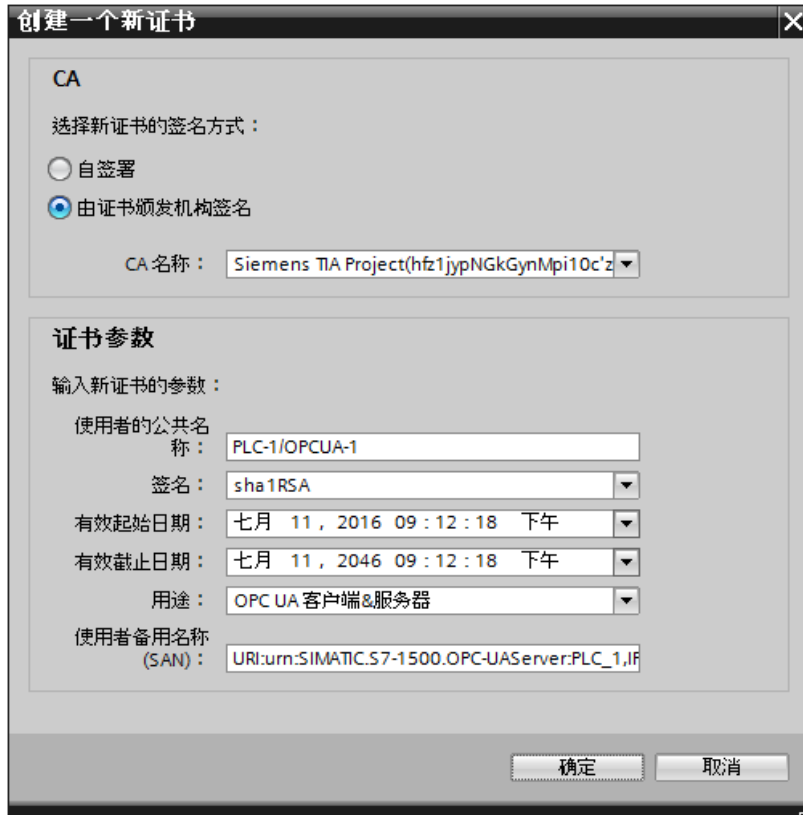


图 9-29 示例：在 STEP 7 中生成 OPC UA 服务器证书

4. 如果您所在公司或客户的安全规范要求使用其它参数，则使用其它参数。

- 请注意以下事项：

- CA

选择是要对证书进行自签名还是通过 TIA Portal 的 CA 证书之一进行签名。

有关这些证书的说明信息，请参见“OPC UA 证书 (页 151)”部分。

如果希望生成的证书通过 TIA Portal 的 CA 证书之一进行签名，则必须登录 STEP 7。

若要登录，单击“项目树”中的项目。然后，单击“全局安全设置”(Global security settings) 下的“用户管理”(User management) 并登录。

- 证书主体

默认设置始终由项目名称和“\OPCUA-1”构成。

在本例中，项目名称为“HallA-Works1-Heating”。

项目名称在 CPU 属性的“常规 > 项目信息 > 名称”(General > Project information > Name) 下设置。

可保留默认设置，也可以在项目的“证书使用者”(Certificate subject) 中为 OPC UA 服务器输入更有意义的其它名称。

- 签名

在此选择对服务器证书进行签名时要使用的哈希和加密程序。

下列条目可用：

“sha1RSA”（HASH 安全算法 V1）。缩写 RSA 表示签名算法（以算法开发者 Rivest、Shamir 和 Adleman 的首字母进行命名）。

“sha256RSA”（HASH 安全算法 V2；数值 256 为 HASH 值的长度）。

- 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期截止的日期和时间。

确保证书的有效期不只一年或仅仅几年。

在本例中，该证书的有效期为 30 年。

然而，为安全起见，应以尽量短的时间间隔更新证书。由于有效期较长，用户可自行决定何时更新证书，如需要对系统进行维护时更新证书。

– 应用程序

默认设置为“OPC UA 客户端和服务端”(OPC UA client & server)。

保留 OPC UA 服务器的该默认设置。

在 STEP 7 中，可从多个位置调用“生成新证书”(Generate new certificate) 对话框。

例如，若要为 CPU 的 Web 服务器调用此对话框，则在“应用程序”(Application) 下输入“Web 服务器”(Web server)。

在应用程序下拉列表中，提供有下列条目：

“OPC UA 客户端”(OPC UA client)、

“OPC UA 客户端和服务端”(OPC UA client & server)、

“OPC UA 客户端 - 服务端”(OPC UA client-server)、

“TLS”、

“Web 服务器”(Web server)。

– 证书使用者的备选名称

在上述示例中输入以下内容：“URI:urn:SIMATIC.S7-1500.OPC-UA-Server:WorksA-Hall1-Heating,IP:192.168.178.151,IP:192.168.1.1”

下列内容同样有效：“IP:192.168.178.151,IP:192.168.1.1”。

在此，需输入访问 CPU 中 OPC UA 服务器的 IP 地址（请参见“访问 OPC UA 服务器 (页 167)”）。

这意味着，OPC UA 客户端可检查是否真正建立与 S7-1500 的 OPC UA 服务器的连接，或攻击者是否可能尝试从另一台计算机向 OPC UA 客户端发送篡改后的值。

参见

处理服务器和客户端证书 (页 175)

9.4.5 处理服务器和客户端证书

OPC UA 服务器的证书

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间的安全连接。

服务器证书可用于证实身份。

服务器证书会自动生成

如果已激活 OPC UA 服务器并确认了安全信息，STEP 7 会自动生成服务器证书并将其存储在 CPU 的本地证书目录中。

可利用 CPU 的本地证书管理器来查看并管理此目录（导出或删除证书）。

下图显示了包含自动生成的 OPC UA 服务器证书的 CPU 本地证书管理器：



图 9-30 CPU 本地证书管理器

或者，也可以自行生成服务器证书。请阅读“通过 STEP 7 生成服务器证书 (页 171)”部分的信息。

在建立连接的过程中，服务器证书将从服务器传送到客户端，请参见“安全地传输消息 (页 158)”部分。

客户端将检查该证书。

客户端用户决定是否要信任服务器证书。

客户端用户现在必须决定是否要信任服务器证书。

如果用户信任该证书，则客户端会将服务器证书存储在包含其信任的服务器证书的目录中。

以下示例显示了“UA Sample Client”客户端的对话框。

如果用户单击“是”(Yes)，则客户端会信任服务器证书：

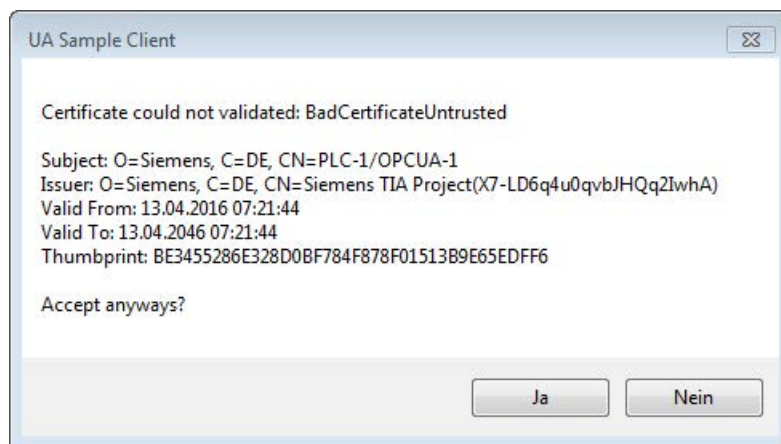


图 9-31 在“UA Sample Client”中接受证书

OPC UA 客户端的证书

仅当 OPC UA 服务器也将 OPC UA 客户端证书归类为受信证书时，才能建立服务器与客户端之间的安全连接。

用户自行决定是否信任客户端证书。

若要信任客户端证书，请按以下步骤操作：

1. 激活本地证书管理器中的“使用证书管理器的全局安全设置”(Use global security settings for certificate manager) 选项。
选中后，会激活全局证书管理器。
“全局安全设置”(Global security settings) 菜单项随即显示在 STEP 7 项目树中。
2. 双击“全局安全设置”(Global security settings)。
3. 双击“用户登录”(User login)。

4. 登录。

“证书管理器”(Certificate manager) 菜单项现在显示在 STEP 7 项目树的“全局安全设置”(Global security settings) 下。

5. 双击“证书管理器”(Certificate manager)。

STEP 7 会打开全局证书管理器。

6. 单击“设备证书”(Device certificates) 选项卡。

7. 在此选项卡的空白区域（而不是证书上）单击右键。

8. 从快捷菜单中，选择“导入”(Import) 选项。

将显示导入证书的对话框。

9. 选择客户端证书。

10. 单击“打开”(Open) 按钮，导入证书。

客户端证书现在包含在全局证书管理器中。

注意刚刚导入的客户端证书 ID。

11. 在 CPU 属性中，单击“常规”(General) 选项卡。

12. 单击“OPC UA > 安全”(OPC UA > Security) 区域。

13. 在“安全通道”(Secure channel) 对话框中，向下滚动到“受信任的客户端”(Trusted clients) 部分。

14. 双击表中带有“<新增>”(<add new>) 按钮的空行。

该行中会显示带有三个点的按钮（浏览按钮）。

15. 单击此按钮。

16. 选择已导入的客户端证书。

17. 单击带有绿色复选标记的按钮。

18. 编译项目。

19. 将组态加载到 S7-1500 CPU 中。

结果

服务器现在信任该客户端。

如果服务器证书也可信，服务器与客户端现在即可建立安全连接。

自动接受客户端证书

如果激活“运行期间自动接受客户端证书”(Automatically accept client certificates during runtime) 选项（位于“受信任的客户端”(Trusted clients) 列表下方），则服务器将接受所有客户端证书。

小心：安全风险。调试后停用此选项。

注意
调试后的设置 为避免安全风险，调试后，请再次停用“运行期间自动接受客户端证书”(Automatically accept client certificates during runtime) 选项。

客户端证书来自哪里？

- 如果使用制造商或 OPC 基金会的 UA 客户端，则在安装过程中或首次调用程序的过程中，会自动生成客户端证书。
- 在 STEP 7 中，需要利用全局证书管理器导入这些证书，并将其应用于各自的 CPU（如前文所示）。
- 如果自行编写 OPC UA 客户端，则可以通过程序生成证书，详情请参见“客户端实例证书 (页 192)”部分。
- 或者使用工具生成证书，例如，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“用户自己生成 PKI 密钥对和证书 (页 154)”部分。
 - 有关使用 OPC 基金会证书生成器的信息，请参见创建自签名证书 (页 152)部分。

组态服务器的安全策略

下图显示了可用于对消息进行签名和加密的服务器安全策略。



图 9-32 STEP 7 中 OPC UA 服务器的安全设置

默认情况下, 会生成采用 SHA1 进行签名的服务器证书, 同时会启用以下安全策略:

- 无 (None)
非安全端点
- Basic128Rsa15 - 签名 (Basic128Rsa15 -Sign)
安全端点, 支持一系列采用哈希算法 RSA1.5 和 128 位加密的算法。
此端点通过签名确保数据的完整性。
- Basic128Rsa15 - 签名及加密 (Basic128Rsa15 -Sign & Encrypt)
安全端点, 支持一系列采用哈希算法 RSA1.5 和 128 位加密的算法。
此端点通过签名及加密确保数据的完整性和保密性。
- Basic256Rsa15 - 签名 (Basic256Rsa15 -Sign)
安全端点, 支持一系列采用哈希算法 RSA1.5 和 256 位加密的算法。
此端点通过签名确保数据的完整性。
- Basic256Rsa15 - 签名并加密 (Basic256Rsa15 -Sign & Encrypt)
端点进行安全连接, 支持一系列使用 HASH 算法 RSA1.5 和 256 位加密的算法。
该端点基于签名和加密, 确保数据的完整性和保密性。

为实现更高的安全性，还可启用以下安全策略：

- **Basic256Sha256 - 签名 (Basic256Sha256 - Sign)**
安全端点，支持一系列 256 位哈希和 256 位加密算法。
此端点通过签名确保数据的完整性。
- **Basic256Sha256 - 签名及加密 (Basic256Sha256 - Sign & Encrypt)**
安全端点，支持一系列 256 位哈希和 256 位加密算法。
此端点通过签名及加密确保数据的完整性和保密性。

要启用某个安全策略，可选择相应行的复选框。

说明

如果使用安全策略“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。

对于安全策略“Basic256Sha256-签名”(Basic256Sha256 -Sign) 和“Basic256Sha256-签名并加密”(Basic256Sha256 -Sign & Encrypt)，STEP 7 中的证书颁发机构将使用“SHA256”自动对证书进行签名。

9.4.6 用户认证

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。

可通过以下方式进行设置：

- 访客认证

用户无需进行身份验证（匿名访问）。

如果 OPC UA 服务器不验证客户端用户的身份，则激活“激活访客身份验证”(Activate guest authentication) 选项。

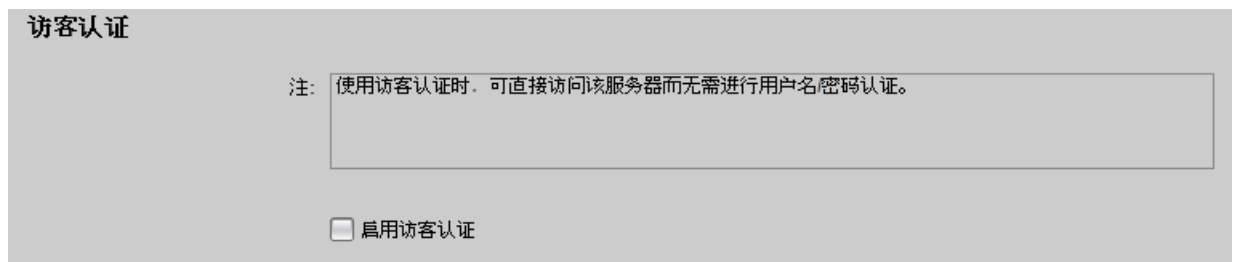


图 9-33 在 STEP 7 中激活访客身份验证

- 通过用户名和密码进行验证

用户必须进行身份验证（而非匿名访问）。

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。

例如，用户名是否持有正确的密码。

如果要使用这种类型的用户身份验证，请设置“OPC UA > 服务器 > 安全 > 用户身份验证 > 通过用户名和密码进行身份验证”(OPC UA > Server > Security > User authentication > Authentication by means of user name and password) 下的“通过用户名和密码激活身份验证”(Activate authentication by means of user name and password) 选项。

取消激活 guest 认证。

在“用户管理”(User management) 表中输入用户名。

要完成此操作，可单击“<添加新用户>”(<Add new user>) 选项并输入用户名及相应的密码。最多可添加 21 个用户。

9.4.7 OPC UA 服务器的许可证

运行时许可证

S7-1500 的 OPC UA 服务器运行所需的许可证。

所需的许可证类型取决于相应 CPU 的性能。

分为以下许可证类型：

- SIMATIC OPC UA S7-1500 小型（CPU 1511、CPU 1512、CPU 1513、ET 200SP CPU、CPU 1515SP PC 需要）
- 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）

所需的许可证类型在“属性 > 常规 > 运行时许可证 > OPC-UA > 所需的许可证类型”(Properties > General > Runtime licenses > OPC-UA > Type of license required) 下显示：



图 9-34 在 STEP 7 中显示运行时许可证

请执行以下步骤，确认购买所需的许可证：

1. 在 CPU 属性中，单击“运行时许可证 > OPC UA”(Runtime licenses > OPC UA)。
2. 从“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。

9.5 对 OPC UA 客户端进行编程

9.5.1 采用 C# 编写的 OPC UA 客户端

本章节介绍示例程序“UaClient”。示例程序位于本手册随附的 ZIP 文件中。

该程序显示了如何自行对 OPC UA 客户端进行编程。

9.5.2 示例程序的要求

要求

该示例假设满足以下要求：

- 采用 Microsoft Visual Studio 2013 作为开发环境
- C# 编程语言
- .NET Framework 4.5
- 用于 Unified Automation 的 OPC UA 客户端的软件开发工具包 (SDK) (基于 .NET 的 OPC UA 客户端/服务器 SDK 软件包)。

可以免费下载 SDK，并使用该工具包对客户端进行开发和测试。

生产操作需要 Unified Automation 的许可证。

9.5.3 示例程序的结构

程序的基本结构

该程序主要由以下模块组成：

1. 建立与 OPC UA 服务器的连接
2. 建立连接后，可执行以下操作：
 - 从服务器中读取 PLC 变量和 DB 变量
 - 将 PLC 变量和 DB 变量写入服务器中
 - 在服务器的地址空间内进行导航
 - 读写注册的 PLC 变量（Registered Read 和 Registered Write）
 - 建立订阅并为该订阅分配条目
3. 断开与服务器的连接
4. 生成客户端证书

9.5.4 建立与 OPC UA 服务器的连接

9.5.4.1 本章节概述

S7-1500 的 OPC UA 服务器可通过不同的安全设置（端点）来访问。

以下说明展示了如何定位这些端点，然后建立与某个端点的连接。

9.5.4.2 以不同的安全设置访问服务器

查询安全要求

建立与 OPC UA 服务器之间的连接之前，应首先确定连接服务器所需的安全设置。

各种安全要求称为 Endpoints。

“FindEndpointsBtn_Click”方法展示了如何查询服务器的 Endpoints。单击“连接到服务器”(Connect to Server) 选项卡中的“搜索”(Search) 按钮，会调用该方法：

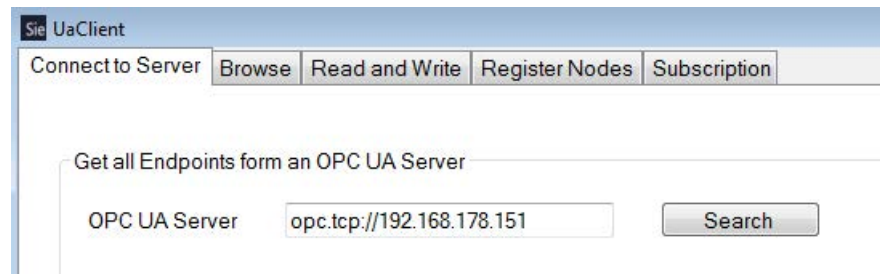


图 9-35 在“UaClient”中搜索端点

“FindEndpointsBtn_Click”方法完全包含在随附的 Visual Studio 项目的“Form1.cs”文件中。

可通过以下步骤运行该方法：

1. `Discovery discovery = new Discovery();`

该方法会创建 `Discovery` 类的对象。

2. `m_endpoints = discovery.GetEndpoints(ServerUrlTB.Text);`

搜索时，会调用 `GetEndpoints` 方法。

从“ServerUrlTB”文本框的文本属性中读取该参数。

`GetEndpoints` 建立与 OPC UA 服务器的非安全连接并查询可用的安全要求。

在保持 `EndpointDescription` 类型对象的列表“`m_endpoints`”中，服务器响应将保存示例程序。

```
3. foreach (EndpointDescription endpoint in m_endpoints)
{
    ListViewItem item = new ListViewItem();

    item.Text = endpoint.EndpointUrl;

    item.SubItems.Add(endpoint.SecurityMode.ToString());

    item.SubItems.Add(SecurityProfiles.GetDisplayName(endpoint.SecurityPolicyUri));

    item.Tag = endpoint;

    EndpointsLV.Items.Add(item);
}
```

在此 **Foreach** 循环中，该方法将准备所接收的服务器 **Endpoints**，以便在 **ListView** 中显示：

结果

将显示可用的 OPC UA 服务器的端点：

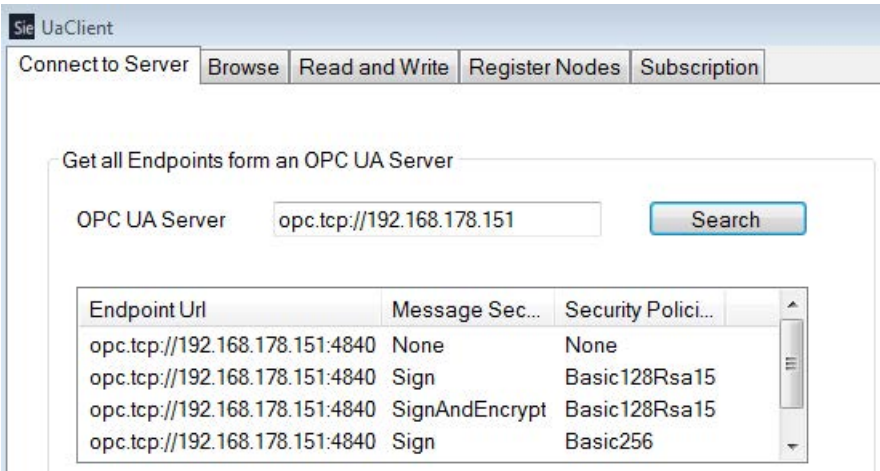


图 9-36 在“UaClient”中搜索端点的结果

9.5.4.3 建立与服务器端点的连接

要求

现已确定 **OPC UA** 服务器具有哪些安全要求（端点）（请参见“以不同的安全设置访问服务器（页 185）”）。

建立连接

现在可以建立与服务器某个特定端点的连接并打开会话。

该过程显示在“ConnectBtn_Click”方法中。

在“连接到服务器”(Connect to Server) 选项卡中选中某个特定端点，然后单击“连接”(Connect) 按钮，则将调用该方法。

将执行以下程序段（完整源代码保存在“Form1.cs”文件中）：

```
1. m_session = new Session(m_application);
```

该方法会创建新的会话对象。

该构造函数接收客户端当前实例的引用作为参数。

m_session 变量包含对新会话对象的引用。

```
2. m_endpoint = (EndpointDescription)m_endpoints[m_selectedEndpointIndex];
```

该方法将读取在 **m_endpoints** 变量中选中的端点并将其分配给 **m_endpoint** 变量。

```
3. m_session.Connect(m_endpoint, m_session.DefaultRequestSettings);
```

与选定的服务器端点的会话结构：

- 将调用 **Session** 类的 **Connect** 方法。
- 该 **Connect** 方法接收所选端点和默认设置作为参数。

结果

已建立与 OPC UA 服务器的连接。

可建立会话。

m_session 对象表示与选定的服务器端点的连接和会话。

UaClient 在“连接到服务器”(Connect to Server) 选项卡中指示已建立连接：

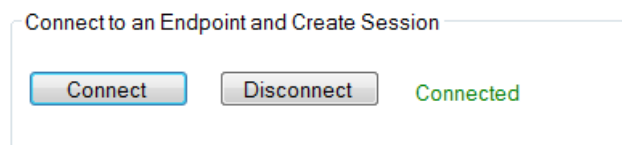


图 9-37 UaClient 中显示与端点的连接

9.5.5 使用与服务器端点的现有连接

9.5.5.1 读取 PLC 变量和 DB 变量

本章节概述

本章介绍了如何利用示例程序 UaClient 读取 S7-1500 的 OPC UA 服务器的 PLC 变量值。

查询命名空间索引

要求

已建立与 OPC UA 服务器的会话（请参见“建立与服务器端点的连接 (页 186)”）。

查询命名空间索引

OPC UA 服务器使用命名空间来避免命名冲突。

在以下情况下，可更改命名空间编号：服务器添加其它命名空间或删除现有命名空间。

因此，实际执行读取或写入访问之前，需要查询命名空间索引。

该过程显示在“FindNamespaceIndexBtn_Click”方法中。

单击“连接到服务器”(Connect to Server) 选项卡中的“查找”(Find) 按钮，会调用该方法：

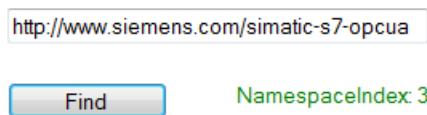


图 9-38 在 UaClient 中查询命名空间索引

对于 UaClient，可输入 Siemens 命名空间作为默认值。在本例中，Siemens 命名空间的索引为 3。

访问 PLC 变量和 DB 变量时，需使用命名空间索引。

读取 PLC 变量

要求

已建立与 OPC UA 服务器的连接并打开了一个会话（请参见“建立与服务器端点的连接 (页 186)”），同时查询了命名空间索引。

寻址 PLC 变量

在服务器端，可读取通过 OPC UA 可访问的 PLC 变量和 DB 变量。

具体过程显示在“ReadBooleanBtn_Click”方法中。

单击“读取和写入”(Read and write) 选项卡“从 PLC 读取”(Read from PLC) 部分中的“读取”(Read) 按钮，会调用该方法：

Index	Boolean Variable		Result
3	<input starttimer\""="" type="text" value="\"/>	Read	True

图 9-39 在 UaClient 中寻址 PLC 变量

在上述示例中输入以下信息：

- 索引

Siemens 命名空间的当前索引。

- 布尔变量

在此输入“StartTimer”标识符。

“StartTimer”是 PLC 程序中使用的存储位的符号名称，为 UaClient 提供布尔值。

StartTimer 在 PLC 程序中定义，因此加上引号。

有关 PLC 程序的信息，请参见“CPU 用户程序 (页 193)”部分。

读取值的源代码

读取 PLC 变量的程序示例

“ReadBooleanBtn_Click”方法将执行以下程序段（完整源代码位于随附的 Visual Studio 项目的 “Form1.cs”文件中）：

```
1. List<ReadValueId> nodesToRead = new List<ReadValueId>();
```

程序会生成要读取值的所有变量的列表。

此列表仅包含 ReadValueId 类型的对象。

```
2. NodeId nodeId1 = new NodeId(ReadPLCVariable_1_TB.Text,
    Convert.ToUInt16(NamespaceIndexTB1.Text));
```

程序将生成 NodeId。

标识符取自 ReadPLCVariable_1_TB 文本框的文本属性（在本例中，为 “StartTimer”）。

命名空间索引位于 NamespaceIndexTB1 文本框的文本属性中。该程序将索引转换为一个整数。

```
3. nodesToRead.Add(new ReadValueId() { NodeId = nodeId1, AttributeId =
    Attributes.Value });
```

示例程序将新的 ReadValueId 对象添加到 nodesToRead 列表中。

要读取的变量的值（位于变量的值属性中）。

```
4. List<DataValue> results = m_session.Read(nodesToRead, 0, TimestampsToReturn.Both,
    new RequestSettings() { OperationTimeout = 10000 });
```

在 m_session 对象中，将调用 Session 类的 Read 方法。

该方法包含待读取的节点的列表（PLC 变量，本例中，为 “StartTimer”）。

第二个参数决定值的时间，以毫秒为单位。如果值为 “0”，则 OPC UA 服务器将从数据源再次请求该值。

第三个参数可决定 OPC UA 服务器返回两个时间戳：

- 在数据源中检测到变量值的时刻。
- 服务器将值发送到客户端的时刻。

第四个参数将读操作的超时值设为 10 秒。

results 列表包含已读取的值（在本例中，读取 PLC 变量 “StartTimer” 的值）。

```
5. ReadBooleanResultTB.Text = results[0].Value.ToString();
```

将结果列表 **results** 中第 0 项的值转换为字符串，并将其分配给 **ReadBooleanResultTB** 文本框中的文本属性。

9.5.6 断开与 OPC UA 服务器的连接

9.5.6.1 断开与服务器的连接

断开连接

“DisconnectBtn_Click”方法显示如何断开与 OPC UA 服务器的连接。

单击“连接到服务器”(Connect to Server) 选项卡中的“Disconnect”按钮，会调用该方法：

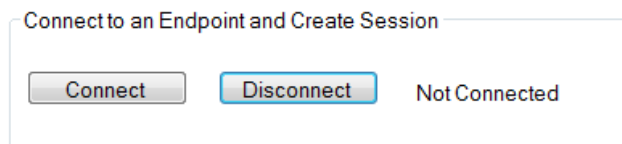


图 9-40 在 UaClient 中断开与端点的连接

将执行以下程序段（完整源代码位于随附的 Visual Studio 项目的“Form1.cs”文件中）：

```
1. m_session.Disconnect();
```

该方法将关闭现有会话并终止与 OPC UA 服务器的连接。

```
2. m_session.Dispose();
```

释放连接资源。

9.5.7 生成客户端证书

9.5.7.1 客户端实例证书

每个实例都需要证书

在计算机上首次启动 UaClient 时，尚不存在该 UaClient 实例的证书。

因此，“Application_MissingApplicationCertificate”方法（“Program.cs”文件）将调用以下对话框以生成证书（来源：Unified Automation）：

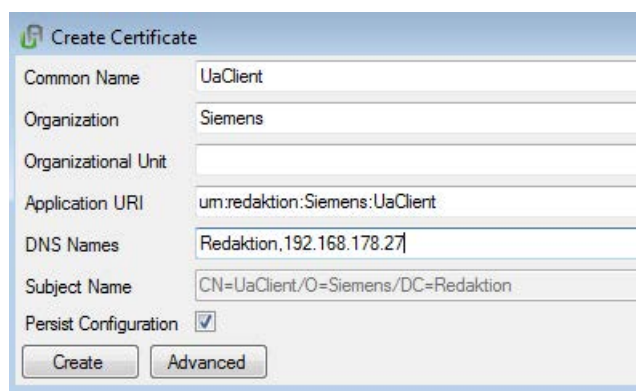


图 9-41 生成 UaClient 实例的证书

例如，在上述对话框中，将“Redaktion”输入到计算机 DNS 名称中。此外，这台计算机的 IP 地址也插入到“DNS Names”中。

单击“创建”(Create) 按钮，生成证书。

该证书已插入到目录“C:\ProgramData\Siemens\Automation\UaClient\pkiclient\own\certs”中。

除证书外，还会生成私有 PKI 密钥并存储在以下目录中：

“C:\ProgramData\Siemens\Automation\UaClient\pkiclient\own\private”。

目录在示例项目的“App.config”文件中指定。

为建立与 S7-1500 的 OPC UA 服务器的安全连接，需要通过 STEP 7 V14 (TIA Portal) 将生成的客户端证书传送到服务器。

9.5.8 CPU 用户程序

9.5.8.1 用户程序的结构 - Main [OB1]

在 PLC 程序中，通过客户端从 S7-1500 服务器中读取的变量值有自己的初始值。

本章节将介绍使 PLC 变量值可用的 PLC 程序。程序仅用于模拟 PLC 变量的值；并不能执行任何实际任务（不控制任何机器和系统）。

作为示例，该程序使以下 PLC 变量值可用：

- StartTimer

布尔变量，其值在一个周期内以五秒的间隔从“1”变为“0”。在 PLC 程序的下一个周期中，该变量值重新返回“1”。

- MyValue

整型变量 (Int16)，当五秒的时间间隔过期时，其值增加 1。

用户程序使值可用

Network 1

下图所示为 PLC 程序的 Network 1：

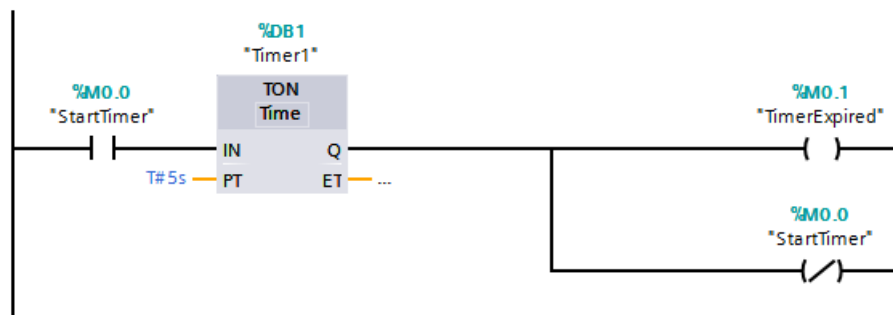


图 9-42 用户程序 Network 1

在输入“IN”的上升沿（从“0”转换为“1”），接通延时开始（输入“PT”中为 T#5s）。

五秒钟后，计数器输出“Q”切换为“1”。这将使 PLC 变量“StartTimer”设为“0”，“TimerExpired”设为“1”。

“TimerExpired”的信号电平为“1”时，表示设定的时间已到期。

Network 2

下图所示为 PLC 程序的 Network 2:

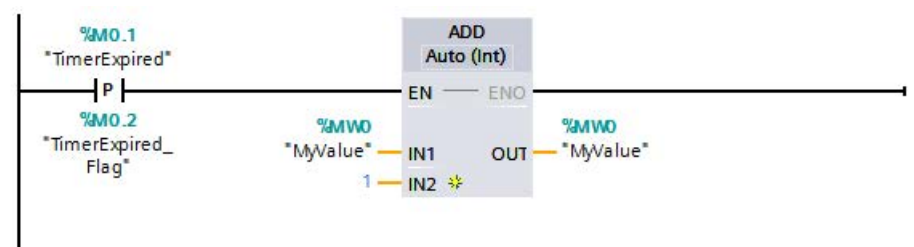


图 9-43 用户程序 Network 2

从“0”切换为“1”可将 ADD 指令的输入“EN”设为 1，从而激活 ADD 指令。
ADD 指令读取“MyValue”的当前值，将该值加 1，然后将新值写入“MyValue”变量。

Network 3

下图所示为 PLC 程序的 Network 3:

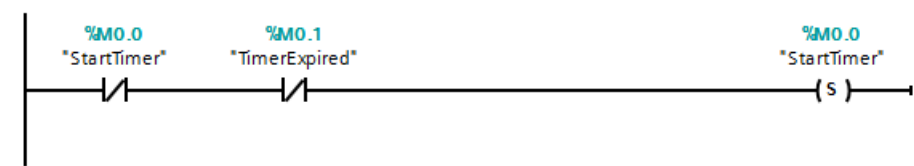


图 9-44 用户程序 Network 3

如果“StartTimer”和“TimerExpired”的值均为“0”，则将“StartTimer”设为“1”，从而启动 Timer1。

9.5.8.2 启用 PLC 变量以通过 OPC UA 进行访问

在 STEP 7 中启用 PLC 变量

下图所示为可针对“UaClient”生成值的 PLC 程序的 PLC 变量组态。

PLC 变量					
	名称	变量表	数据类型	可从 HMI/OPC UA 访问	从 HMI/OPC UA 可写
	StartTimer	Standard...	Bool	☒	☒
	TimerExpired	Stan...	Bool	☒	☒
	TimerExpired_Flag	Standard...	Bool	☒	☒
	MyValue	Standard...	Int	☒	☒

图 9-45 在 STEP 7 中启用 PLC 变量

组态本示例中的所有 PLC 变量，以便通过 OPC UA 进行访问。
此外，OPC UA 客户端可将新值写入变量。

路由

10.1 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。

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 是 S7 子网 1 和 S7 子网 2 间的 S7 路由器。

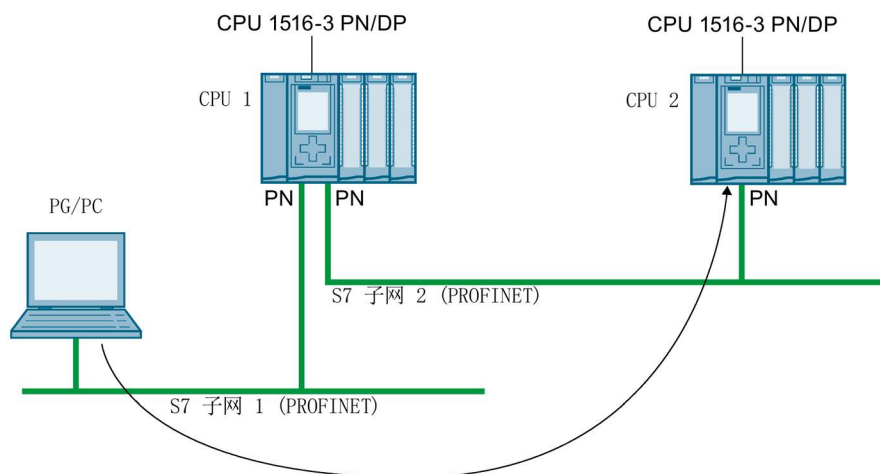


图 10-1 S7 路由：PROFINET - PROFINET

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。

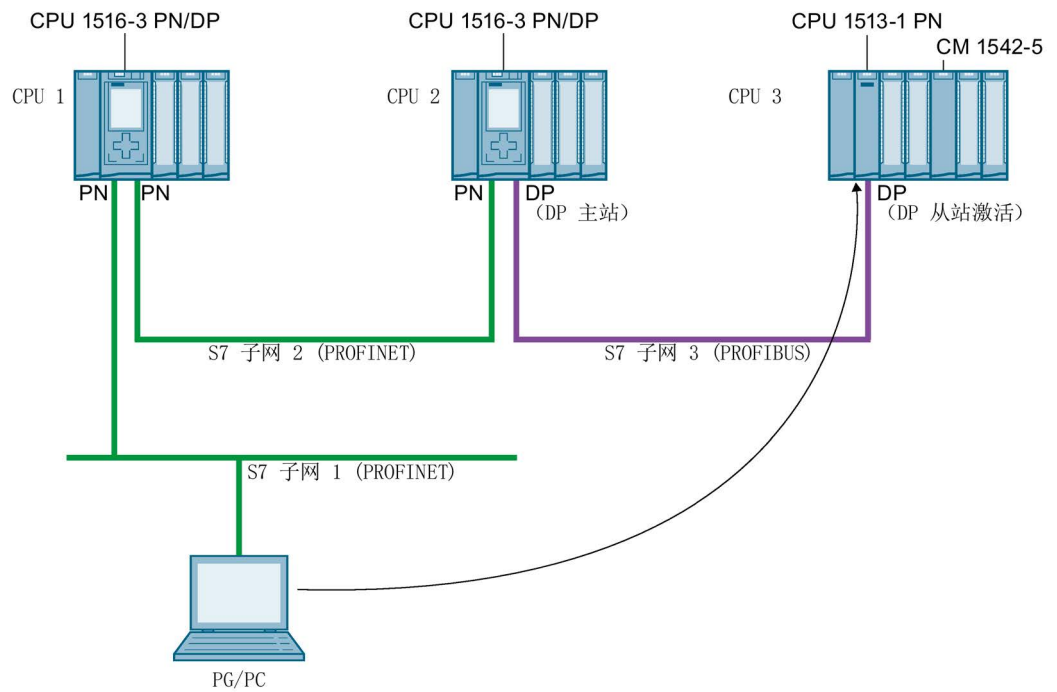


图 10-2 S7 路由: PROFINET - PROFIBUS

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 是 S7 子网 1 和 S7 子网 2 间的 S7 路由器。

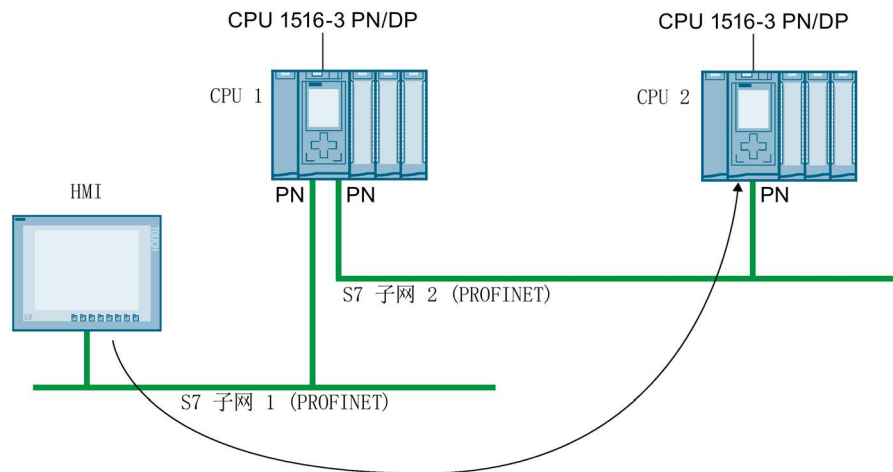


图 10-3 通过 HMI 连接实现的 S7 路由

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 116)”部分中的示例。

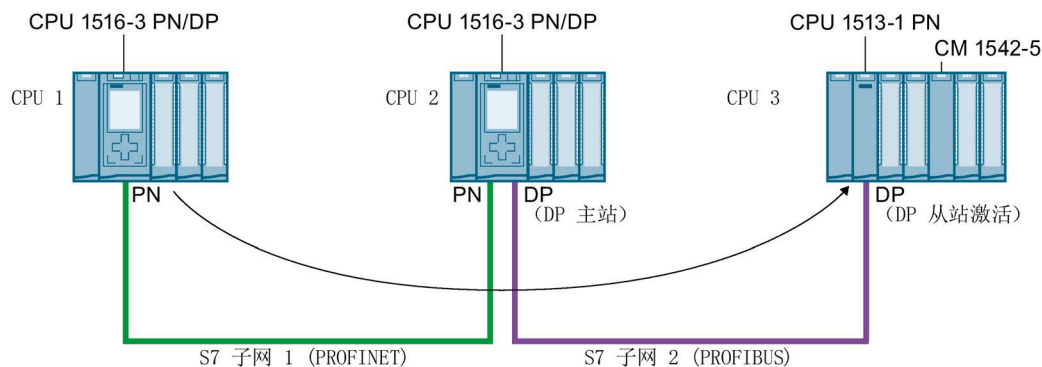


图 10-4 通过 CPU-CPU 通信实现的 S7 路由

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。

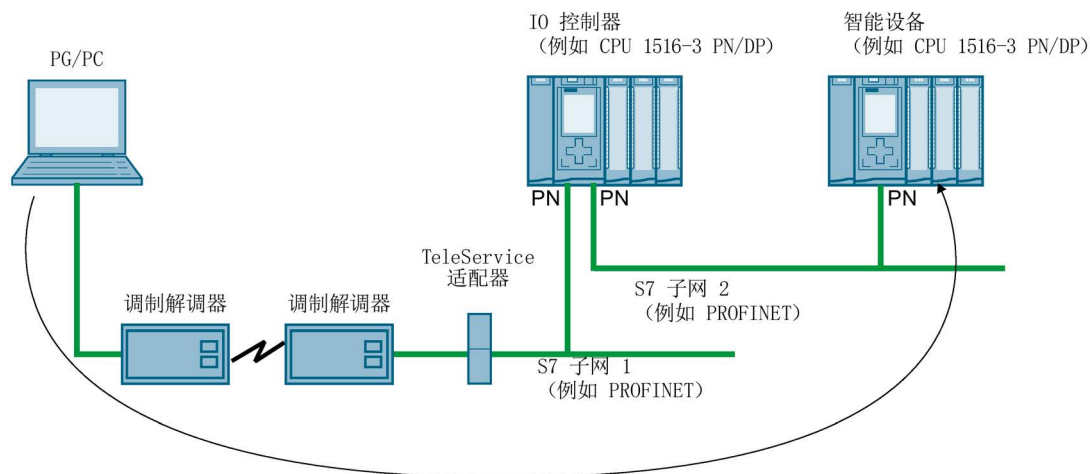


图 10-5 使用 TeleService 对设备进行远程维护

更多信息

- 有关 S7 路由的连接资源分配，请参见“连接资源分配 (页 206)”部分。
- 有关设置 TeleService 连接的信息，请参见 STEP 7 的在线帮助。
- 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 (<http://support.automation.siemens.com/WW/view/ch/51311100>)》
 - TS Adapter (<http://support.automation.siemens.com/WW/view/zh/10805406/133100>) 的下载内容

参见

HMI 通信 (页 44)

10.2 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 (<http://support.automation.siemens.com/WW/view/zh/87611392>)。

下图显示了通过 **PCT** 实现数据记录路由的组态示例。

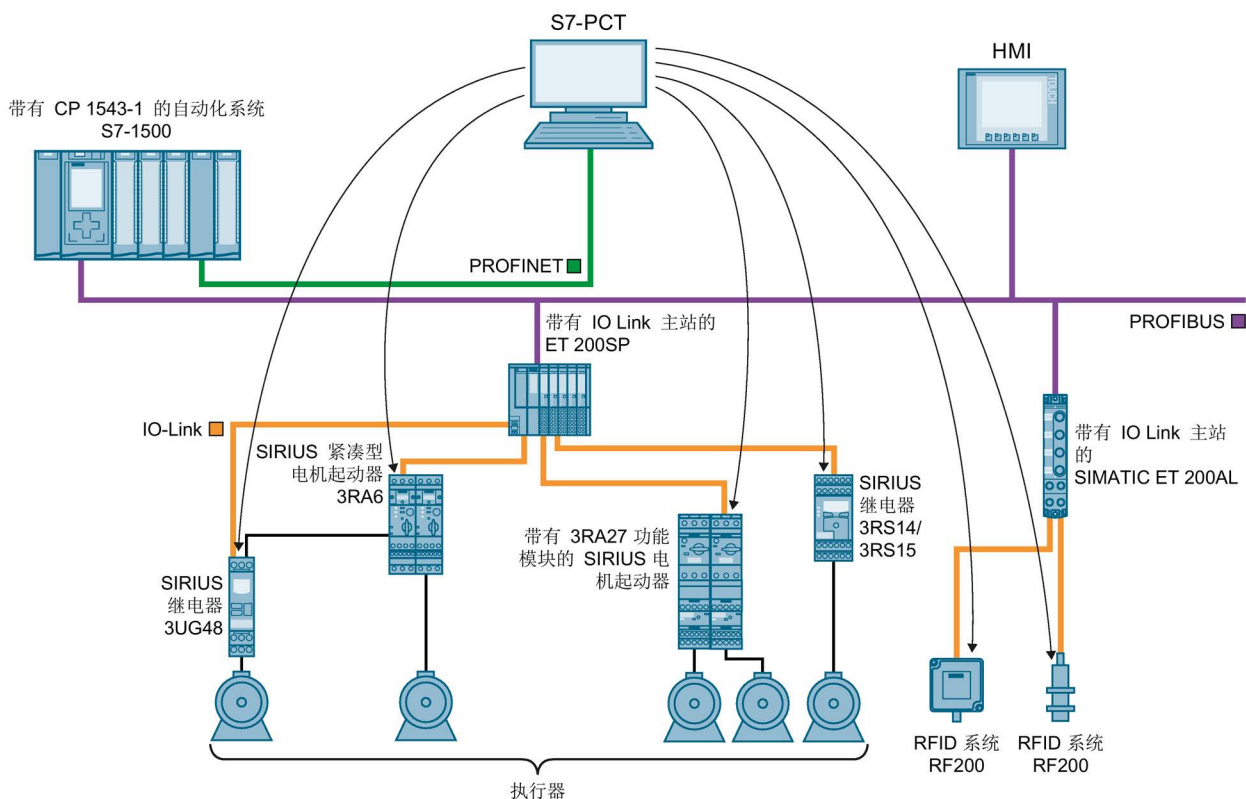


图 10-6 通过 PCT 实现数据记录路由的组态示例

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 206)”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。

连接资源

11.1 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

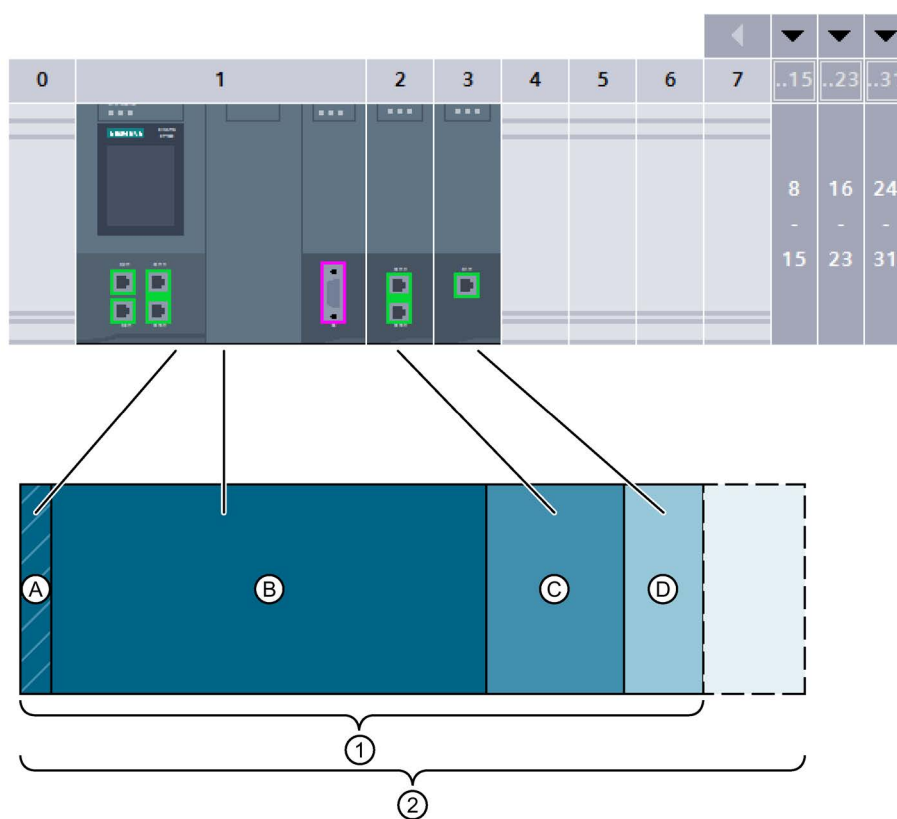
可用连接资源取决于所使用的 **CPU**、**CP** 和 **CM**，且不得超过单个站的最大可用连接资源数量。

站中最大的可用资源数量取决于 **CPU**。

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。

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。**PG** 通信、**HMI** 通信、**S7** 通信、开放式用户通信、**Web** 通信和其它通信（例如 **OPC UA**）等通信服务会使用资源池中的动态连接资源。

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 11-1 站中的连接资源

11.1 站中的连接资源

站中连接资源的数量

表格 11- 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		1511 1511C	1512C 1513	1515	1516	1517	1518
站中连接资源的最大数		96	128	192	256	320	384
	预留	10					
	动态	86	118	182	246	310	374
CPU 中的连接资源		64	88	108	128	160	192
插入 CM/CP 时，可额外使用的连接资源最大数		32	40	84	128	160	192
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		40					
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在设备手册的“技术参数”中指定。

示例

在 CPU 1518-4PN/DP 组态中，包含一个通信模块 CM 1542-1 和一个通信处理器 CP 1542-5。

- 站中连接资源的最大数: **384**
- 可用的连接资源:
 - CPU 1518-4 PN/DP: 192
 - CM 1542-1: 64
 - CP 1542-5: 16
 - 总计: **272**

在该组态中，可使用 272 个连接资源。通过增加其它通信模块，该站最多可额外支持 112 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

11.2 连接资源分配

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。

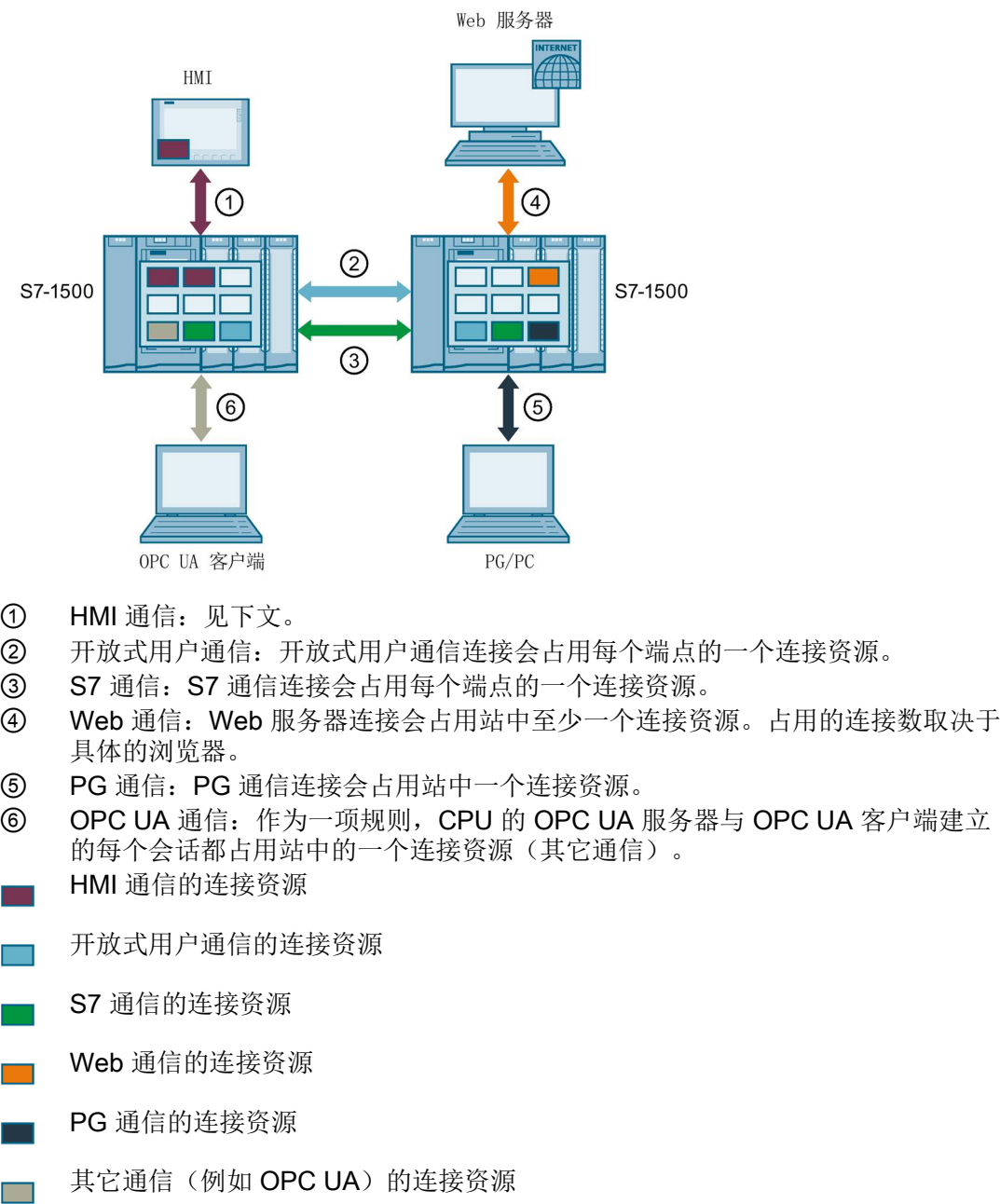


图 11-2 连接资源分配

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 11-2 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
精智面板	2 ¹
RT Advanced	2 ¹
RT Professional	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。

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”。

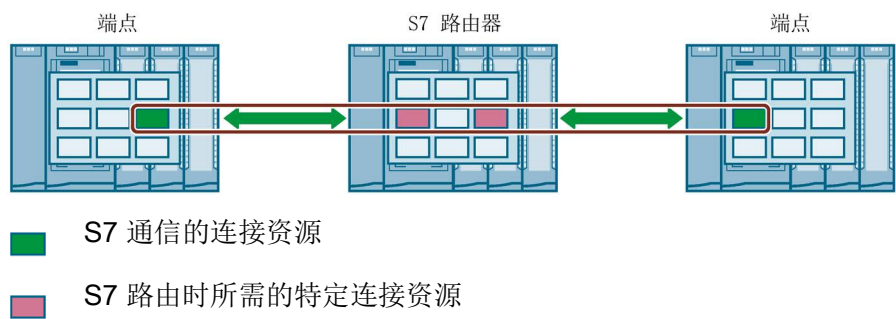


图 11-3 S7 路由的连接资源

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 25)”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。如果连接终止，则 CPU/CP/CM 中的连接资源将再次可用。

- **已组态连接（如 HMI 连接）：**

如果在 STEP 7 中组态了一条连接，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：**

只要 CPU 的 OPC UA 服务器与 OPC UA 客户端之间存在会话，就会占用 CPU 中的连接资源。

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。

11.3 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				
		预留		动态 ⚠	PLC_1 [CPU ...	CP 1543-1_...	模块资源		
最大资源数：		10		246	128	118	CM 1542-5_...		
		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	4	0		
S7 通信：		0	-	23	2	0	21		
开放式用户通信：		0	-	52	39	13	0		
Web 通信：		2	-	-	-	-	-		
其它通信：		-	-	0	0	0	0		
使用的总资源：		4		81	47	17	21		
可用资源：		6		165	81	101	27		

图 11-4 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，81 个已使用，其余 165 个可用。
已使用的资源分别为：
 - 6 个连接资源，用于 HMI 通信
 - 23 个连接资源，用于 S7 通信
 - 52 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（294 个连接资源）超出了站的限值 256 个，因此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 中共有 128 个可用的连接资源。其中，47 个已使用，其余 81 个可用。

已使用的资源分别为：

- 6 个连接资源，用于 HMI 通信
- 2 个连接资源，用于 S7 通信
- 39 连接个资源，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。

PLC_2 [CPU 1516-3 PN/DP] 属性 信息 诊断

设备信息 连接信息 报警显示

连接资源

	站资源				模块资源	
	预留		动态		PLC_2 [CPU 1516-3 PN/DP]	
最大资源数：	10	10	194	194	86	86
	最大	已组态	已用	已组态	已组态	已用
PG 通信：	4	-	4	-	0	0
HMI 通信：	4	4	4	4	4	4
S7 通信：	0	0	0	72	68	34
开放式用户通信：	0	0	0	118	118	45
Web 通信：	2	-	2	-	0	0
其它通信：	-	-	0	-	0	0
使用的总资源：	4	10	194	190	82	82
可用资源：	6	0	0	4	4	4

图 11-5 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 11-6 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 (<http://support.automation.siemens.com/WW/view/zh/59193560>)》。

连接诊断

在线视图中的连接表

在 STEP 7 的“设备与网络”(Devices & networks) 编辑器中选择一个 CPU 后，将在连接表的在线视图中显示连接的状态。



图 12-1 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

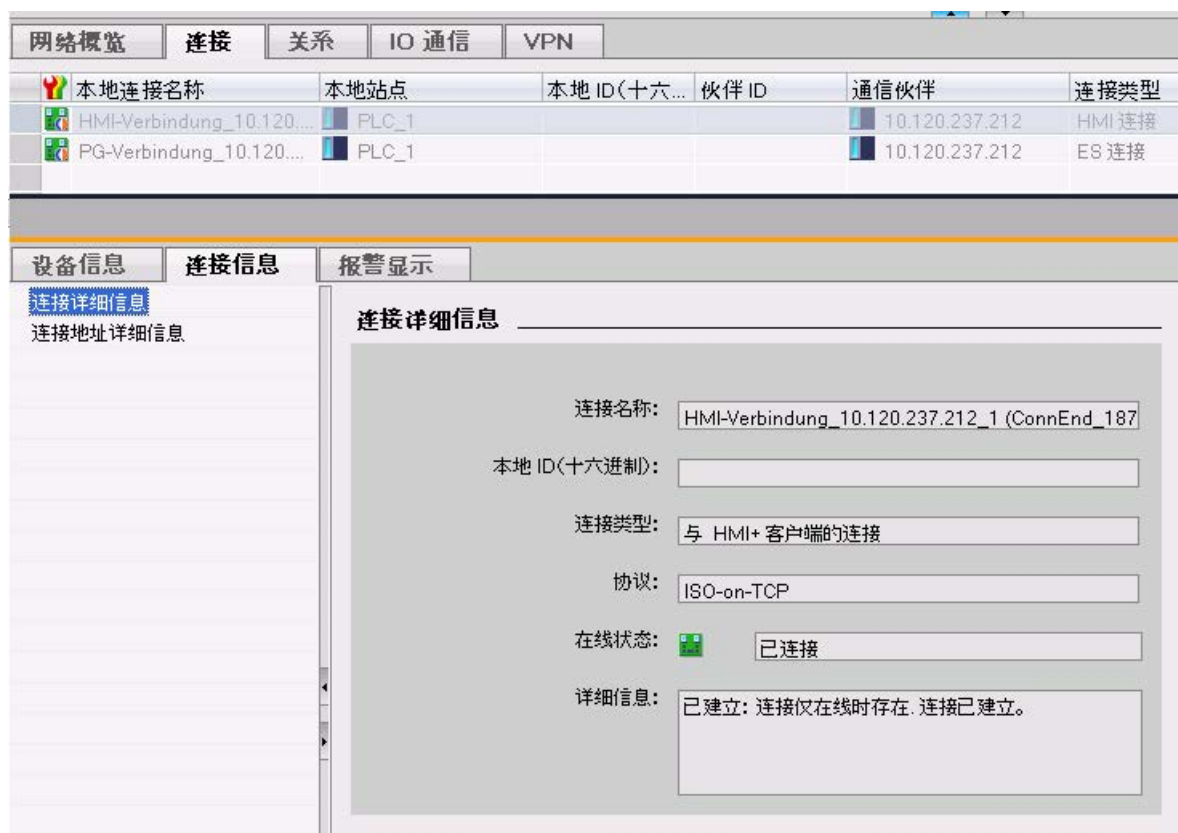


图 12-2 连接诊断 - 连接的详细信息

“连接信息”(Connection information) 选项卡： 地址详细信息

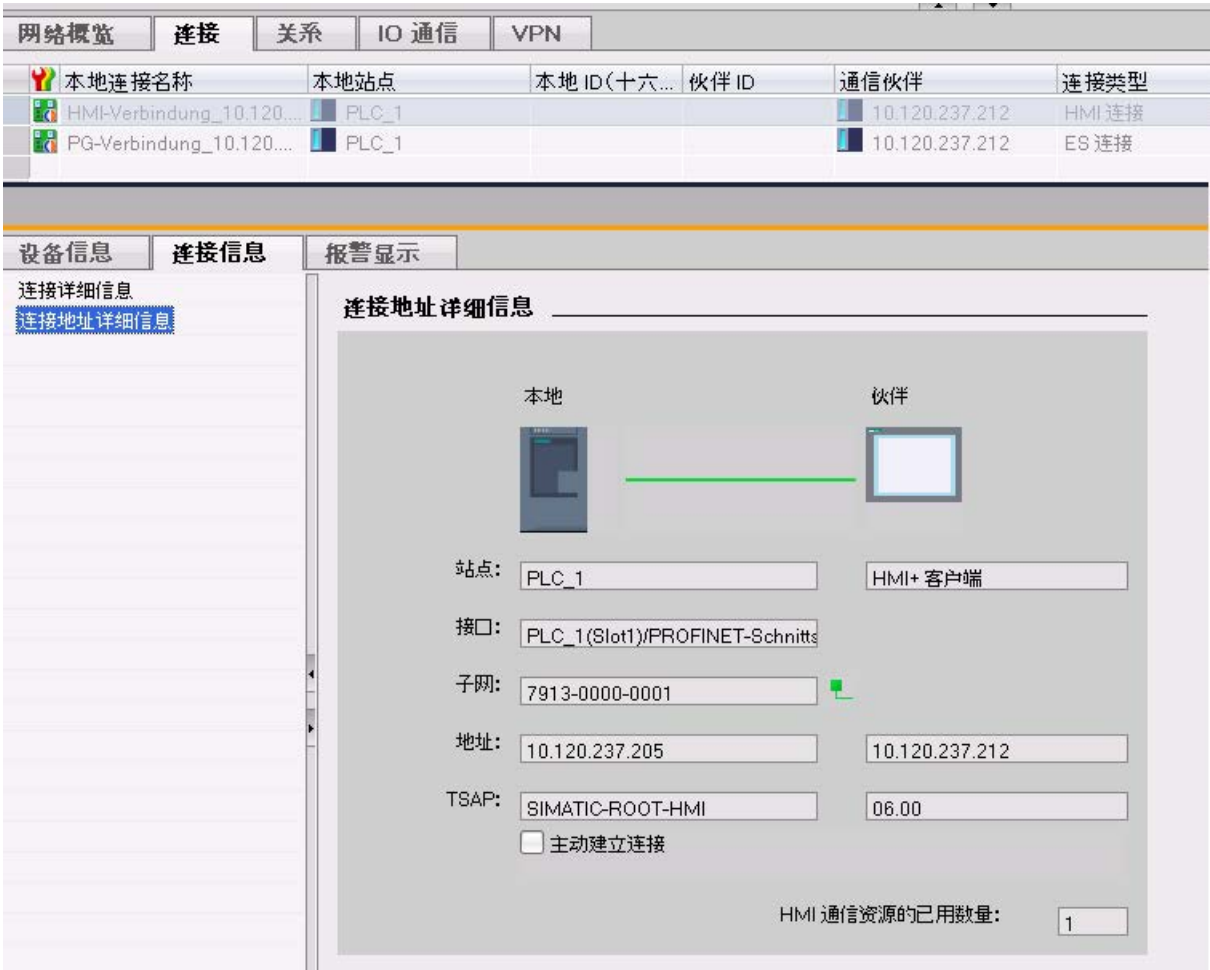


图 12-3 连接诊断 - 地址的详细信息

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”(Connections) 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。

附加信息

有关 Web 服务器的功能介绍，请参见功能手册“Web 服务器 (<http://support.automation.siemens.com/WW/view/zh/59193560>)”。

工业以太网安全

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。

- 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>)。

13.1 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。

13.2 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。

13.3 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。

13.4 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或者一种认证和加密方法。

可选择：

- 认证算法：无、MD5、SHA-1
- 加密算法：无、AES-128、DES

用户也可取消激活 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。

有关取消激活 CPU 中 SNMP 功能的信息，请参见“禁用 SNMP (页 37)”部分。

13.5 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。

术语表

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** 用于服务器与客户端间的文件上传或下载。可创建并读取 **FTP** 目录，也可对目录和文件进行重命名或删除。

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 and 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 (Remote Terminal Unit)；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 (Transmission Control Protocol)；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 (Network Time Protocol, 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 基金会制定的用于在机器之间进行通信的协议。

PG

→ 编程设备

PNO

→ *PROFIBUS 用户组织*

PROFIBUS

过程现场总线 (**Process Field 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 (**Management Information Base**) 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。

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。

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

过程映像是可编程逻辑控制器 (PLC) 的地址区域，用于以数字形式存储所连接模块上输入的信号状态和输出的逻辑状态。

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。

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。

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）、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)”。

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

3

3964(R) 程序, 126

A

Advanced Encryption Algorithm, 77

AES, 77

B

BRCV, 117

BSEND, 117

C

CM, 15

CP, 15

F

FDL, 48

Freeport 协议, 126

FTP, 20, 48, 69, 70

G

GET, 117

H

Handshake Protocol, 79

HMI 通信, 20, 44

I

IM, 19

ISO, 20, 47

ISO-on-TCP, 47, 56

M

Modbus TCP, 48

Modbus 协议 (RTU), 126

N

NTP, 20, 220

P

PCT, 200

PG 通信, 20, 41

Private Key, 75

Public Key, 75

PUT, 117

R

Record Protocol, 79

RFC 5280, 75

S

S7 通信, 20, 116, 207

S7 路由, 195

连接资源, 207

Secure Socket Layer, 79

SNMP, 20, 221

SSL, 79

Syslog, 220

T

TCON, 49

TCP, 20, 47, 56

TDISCON, 49

TLS, 79

Transport Layer Security, 79

TRCV, 49

TRCV_C, 49

TSEND, 49

TSEND_C, 49

U

UDP, 20, 47, 56

URCV, 117

USEND, 117

USS 协议, 126

W

Web 服务器, 20

X

X.509, 75

G

工业以太网安全, 218

K

开放式用户通信

协议, 47

指令, 49

特性, 46

开放式用户通信协议, 47

开放式通信

连接组态, 56

建立 FTP, 70

建立 TCP、ISO-on-TCP、UDP 连接, 56

建立电子邮件, 69

R

日志记录, 220

Z H

中间人攻击, 80

S H

申请方, 80

D

电子邮件, 20, 48, 69

X

写入, 20

D

对称加密, 77

Z

自签名证书, 81

A

安全通信, 75

安全措施, 218

 NTP, 220

 SNMP, 221

 日志记录, 220

 防火墙, 219

F

防火墙, 219

L

连接

 开放式用户通信的指令, 49

 诊断, 214

连接诊断, 214

连接资源

 HMI 通信, 207

 S7 路由, 207

 占用, 209

 在 STEP 7 中显示, 210

 在 Web 服务器中显示, 213

 站特定的, 210

 概述, 24, 202

 数据记录路由, 207

 模块特定, 211

连接资源的占用, 209

S H

时间同步, 20

通信

功能手册, 09/2016, A5E03735819-AE

X

系统数据类型, 50

Z H

证书主体, 80

证书颁发机构, 80

F

非对称加密, 78

J

建立连接, 25

 CP 1543-1 的 ISO 连接, 62

 通过组态, 61

建立和终止通信, 73

D

点到点连接, 20, 126

X

信息安全, 218

H

获取, 20

G

根证书, 83

T

通过 PUT/GET 指令进行通信

 创建和组态连接, 118

通信

 HMI 通信, 44

 PG 通信, 41

 S7 通信, 116

 S7 路由, 195

 开放式用户通信, 46

 开放式通信, 46

 建立和终止, 73

 点到点连接, 126

 通信协议, 47

 数据记录路由, 200

通信方式

 概述, 20

通信处理器, 15

通信处理器的接口, 17

通信服务

 连接资源, 24

通信接口, 16

通信模块, 15

通信模块的接口

 点到点连接, 18

J

接口模块, 19

Z

最终实体证书, 83

Q

签名, 82

S H

数字证书, 80

数据一致性, 28

数据记录路由, 200

数据的一致性, 28