

这是一个在工业自动化领域非常常见和重要的话题。OPC UA 与西门子 TIA Portal (博图) 的通讯, 是实现西门子 PLC 数据与上层信息系统 (如 MES、SCADA、ERP 或云平台) 进行标准化、安全交互的核心技术。

下面我将为您详细讲解其原理、配置步骤和注意事项。

核心概念理解

1. TIA Portal (博图):

- 西门子的全集成自动化软件, 用于对西门子 PLC (如 S7-1200, S7-1500, S7-300/400 等)、HMI 等进行编程、组态和调试。
- 它本身不直接提供 OPC UA 服务, 而是在 PLC 中**激活并配置** OPC UA 服务器功能。

2. OPC UA:

- 一个独立于平台、面向服务的工业通讯架构。它提供了一个统一的“信息模型”, 让不同厂商的设备都能以相同的方式“说话”。
- 服务器 (Server):** 提供数据的设备。在这里, 就是**开启了 OPC UA 服务器功能的西门子 PLC**。
- 客户端 (Client):** 访问和使用数据的软件。例如: 上位机 SCADA 系统 (如 WinCC、Ignition)、MES 系统、自定义开发的应用程序、或者一个测试工具 (如 UaExpert)。

通讯原理

简单来说, 整个过程就是:

在 TIA Portal 中配置 PLC -> 激活并设置其 OPC UA 服务器 -> 编译下载到硬件 -> OPC UA 客户端通过网络连接到 PLC 的 OPC UA 端口, 访问其数据。

配置步骤 (以 S7-1500 为例)

以下是在 TIA Portal V17/V18 中配置一个 S7-1500 PLC 作为 OPC UA 服务器的典型流程。

步骤 1: 在 TIA Portal 中激活 OPC UA 服务器

- 打开项目与设备:** 在 TIA Portal 中打开你的项目, 并进入 PLC 的设备视图。
- 进入属性设置:** 在左侧“设备组态”中, 选中你的 CPU (如 CPU 1511-1 PN), 然后在右侧的属性窗口中, 找到“**OPC UA**”选项。
- 激活服务器:**
 - 勾选“**激活 OPC UA 服务器**”。
 - 系统会自动分配一个端口号, 通常是 **4840**。你可以根据需要修改, 但要确保不与其它服务冲突。
 - 注意:** 对于 S7-1200/1500, 通常需要**固件版本 V2.0 或更高**才支持 OPC UA 服务器功能。

步骤 2: 配置服务器端点和安全策略

- 服务器端点:** 在“服务器端点”下, 你可以看到默认的端点 `opc.tcp://<PLC_IP_Address>:4840`。这是客户端需要连接的地址。
- 安全设置:**
 - 无安全策略 (None):** 仅用于测试环境, 数据传输不加密也不签名。
 - 基本 128Rsa15 (Basic128Rsa15):** 较低级别的安全策略。
 - 基本 256Sha256 (Basic256Sha256):** **目前推荐和常用的安全策略**。它提供签名和加密。
 - 根据你的安全需求, 至少选择“**Basic256Sha256**”并勾选“**签名和加密**”。

步骤 3: 管理用户身份认证 (可选但推荐)

为了提高安全性, 可以为 OPC UA 连接设置用户名和密码。

- 在“用户管理”中, 你可以添加新的用户, 并为其分配角色 (如“匿名”、“游客”、“操作员”、“管理员”等)。
- 在“服务器设置”中, 你可以禁用“匿名登录”, 强制客户端使用用户名/密码登录。

步骤 4: 创建和发布地址空间

OPC UA 客户端能看到 PLC 中的哪些变量, 取决于你“发布”了什么。

- 创建变量:** 在你的 PLC 程序中 (例如, 在全局数据块“DB”中) 创建需要的变量。

```
pascal
```

```
// 例如, 在全局数据块 "MainData" 中创建变量
```

```
Temperature : Real;
```

```
Pressure : Int;
```

```
Motor_Start : Bool;
```

- 发布变量到 OPC UA 服务器:**

- 在项目树中, 找到你的 PLC, 展开它, 找到“**OPC UA**”->“**服务器接口**”。

- 右键点击“服务器接口”，选择“**添加新的 OPC UA 文件夹**”来组织你的变量结构。
- 然后，你可以直接从左侧的“PLC 变量”表或“程序块”中的数据块中，**拖拽变量**到右侧的“服务器接口”窗口中。TIA Portal 会自动为这些变量创建 OPC UA 节点。

步骤 5: 编译与下载

1. **编译项目**: 确保没有错误。
2. **下载到 PLC**: 将硬件配置和程序下载到 PLC 中。**下载后, OPC UA 服务器就已经在 PLC 上运行了。**

客户端连接测试 (使用 UaExpert)

UaExpert 是一款免费的通用 OPC UA 客户端测试工具。

1. **打开 UaExpert**, 创建一个新项目。
2. **添加服务器**:
 - 双击“Servers” -> “Double-Click to Add...”。
 - 在“Endpoint”中输入你在 TIA Portal 中看到的 URL: `opc.tcp://192.168.0.1:4840` (请替换为你的 PLC 实际 IP)。
3. **连接与安全**:
 - UaExpert 会显示 PLC 提供的可用端点。选择你配置的安全策略 (如 Basic256Sha256)。
 - 如果设置了用户名/密码, 在下一步中输入。
4. **浏览与读写数据**:
 - 连接成功后, UaExpert 会显示 PLC 的整个地址空间。你可以找到你发布的文件夹和变量。
 - 将变量拖拽到“Data Access View”中, 即可实时监控其值。你也可以尝试修改变量值 (对于可写变量)。

优势与总结

通过 OPC UA 与博图通讯, 带来了以下巨大优势:

- **标准化**: 打破了传统驱动 (如西门子专属的 S7 协议) 的束缚, 任何支持 OPC UA 的客户端都可以访问。
- **跨平台**: 客户端可以运行在 Windows、Linux、macOS 等系统上。
- **信息丰富**: 不仅传输数据值, 还传输数据类型、描述、工程单位等丰富的元数据。
- **安全性**: 内置了工业级的安全机制, 包括身份验证、授权、签名和加密。
- **可靠性**: 基于 TCP, 提供了稳定的通信机制和错误处理。

注意事项:

- **性能**: OPC UA 通信会占用 PLC 的 CPU 负载和网络带宽, 需合理规划。
- **固件版本**: 务必检查你的 CPU 固件版本是否支持 OPC UA 服务器功能。
- **防火墙**: 确保客户端和服务端之间的 OPC UA 端口 (默认 4840) 在防火墙上开放的。

希望这个详细的解释能帮助您成功实现 OPC UA 与博图的通讯!

好的, 为您提供一份关于 **MQTT (通常指 MQTT 协议) 与西门子 S7-1513R-1 PN 在 TIA Portal (博图) 中的通讯配置指南**。

与标准的 OPC UA 不同, MQTT 是一种基于发布/订阅模式的轻量级物联网协议。S7-1500 系列 (包括 1513R-1 PN) 从固件版本 V2.8 起, 开始原生集成 **MQTT 客户端** 功能, 这意味着 PLC 可以直接向 MQTT 代理 (服务器) 发布消息或订阅主题, 而无需额外的网关。

架构原理

在开始配置前, 理解其通信架构至关重要:

- **S7-1513R-1 PN 作为 MQTT 客户端**。
- 一个 **MQTT 代理**, 例如 **Mosquitto, EMQX, HiveMQ** 或云服务 (如 AWS IoT Core, 阿里云物联网平台)。它负责消息的路由。
- **TIA Portal** 是进行 PLC 编程和通信配置的工程软件。

通信流程: PLC 将特定数据 (如 DB 块中的数据) 作为消息**发布**到代理的某个主题, 或者从代理的特定主题**订阅**消息并写入 PLC 数据区。

配置步骤

以下是在 TIA Portal V17 或更新版本中配置 S7-1513R-1 PN 作为 MQTT 客户端的详细步骤。

第一步: 准备工作与硬件组态

1. **确认固件版本**:
 - 确保您的 CPU 1513R-1 PN 的固件版本**至少为 V2.8** (强烈建议使用最新版本)。这是支持原生 MQTT 客户端功能的最低要求。
 - 在 TIA Portal 中创建项目时, 务必选择正确版本的 CPU。
2. **添加硬件并配置网络**:
 - 在 TIA Portal 中创建新项目, 并添加设备“SIMATIC S7-1500” -> “CPU 1513R-1 PN”。

- 进入“设备视图”，为 CPU 分配正确的 IP 地址（例如 192.168.0.1）。

3. 安装 MQTT 服务器证书 (如果服务器使用 TLS/SSL):

- 如果您的 MQTT 代理使用了自签名证书，为了建立安全的 TLS 连接，您需要先将代理的 CA 证书上传到 PLC。
- 在项目树中展开您的 PLC，找到“证书管理”。
- 将您的 MQTT 代理的 CA 证书 (.crt 或 .der 格式) 拖拽到“受信任的根证书”文件夹中。

第二步：配置 MQTT 连接参数 (“连接管理”)

1. 打开连接管理:

- 在项目树中，进入 PLC 的“设备组态”。
- 在属性窗口中，导航到“MQTT” -> “连接管理”。

2. 创建新连接:

- 点击“添加”按钮，创建一个新的 MQTT 连接。
- 您可以为连接命名，例如 Connection_to_MQTT_Broker。

3. 配置连接详情:

- **常规:**
 - **客户端 ID:** 为您的 PLC 设置一个在该 MQTT 代理中唯一的 ID，例如 S7-1513R-1。
 - **代理地址:** 输入 MQTT 代理的 IP 地址或域名，例如 192.168.0.100。
 - **端口:** 默认为 1883 (非加密) 或 8883 (TLS 加密)。
- **连接机制:**
 - **保持连接时间:** 默认为 60 秒。客户端会在此时间间隔内发送心跳包以保持连接。
 - **清除会话:** 通常勾选。断开重连后不保留之前的会话状态。
 - **最后意愿:** 可选项。配置当 PLC 异常断开时，代理需要发布的最后一条消息。

4. 配置安全设置:

- 如果您的代理需要身份验证，在“用户身份验证”部分输入**用户名和密码**。
- 如果使用 TLS 加密 (强烈推荐用于生产环境):
 - 在“传输层安全”中，选择“**使用 TLS**”。
 - 在“TLS 配置”中，选择您之前导入的受信任根证书。

第三步：定义 MQTT 通信任务 (“发布/订阅”)

这是核心步骤，定义了 PLC 要发送和接收什么数据。

1. 创建数据块:

- 创建一个全局数据块 (例如 MQTT_Data)，并在其中定义您要通过 MQTT 传输的变量结构。
- 例如:

```
pascal
"MQTT_Data" (DB100)
{
    Temperature : Real;
    Pressure : Int;
    Setpoint : Real;
    Motor_Status : Bool;
}
```

2. 配置“发布”任务 (PLC -> 代理):

- 在“连接管理”中，选中您创建的连接，然后在下方的“发布”表中点击“添加”。
- **主题:** 输入发布主题，例如 factory1/line1/sensor_data。
- **有效负载:** 点击“...”按钮，选择数据源。这里您可以选择整个数据块 MQTT_Data，或其中的某个特定变量。
- **内容类型:** 可以选择 JSON，这样 PLC 会将数据块内容自动转换为 JSON 格式发布，便于其他系统解析。
- **发布间隔:** 设置定时发布的周期，例如每 5 秒发布一次。

3. 配置“订阅”任务 (代理 -> PLC):

- 在“订阅”表中点击“添加”。
- **主题:** 输入要订阅的主题，例如 factory1/line1/commands。
- **有效负载:** 点击“...”按钮，选择一个 PLC 中的数据区 (如 MQTT_Data.Setpoint) 来接收来自该主题的消息。
- 当任何客户端向 factory1/line1/commands 主题发布消息时，消息内容会被自动写入到 PLC

的 Setpoint 变量中。

第四步：编程与下载

1. 编译项目：

- 确保所有配置无误，编译整个项目。

2. 下载到 PLC：

- 将硬件配置和程序下载到 S7-1513R-1 PN 中。
- **重要：** 下载完成后，需要将 PLC 从“STOP”模式切换到“RUN”模式，MQTT 客户端连接才会开始尝试建立。

测试与诊断

1. 使用 MQTT 客户端工具测试：

- 在您的电脑上使用一个 MQTT 客户端工具（如 **MQTT Explorer**, **MQTT.fx**）。
- 连接到同一个 MQTT 代理。
- **订阅** PLC 发布的主题（如 factory1/line1/sensor_data），查看是否能收到 PLC 定时发送的 JSON 数据。
- **向** PLC 订阅的主题（如 factory1/line1/commands）发布一个消息（例如 {"Setpoint": 75.5}），查看 PLC 中的 Setpoint 变量是否更新。

2. PLC 诊断：

- 在线连接到 PLC 后，在“连接管理”中可以看到 MQTT 连接的状态（例如“已建立”）。
- 在“诊断缓冲区”中可以查看连接建立失败等错误信息。

优势总结

通过这种原生集成的方式，S7-1513R-1 PN 可以：

- **直接上云/上平台：** 无需额外硬件网关，直接与云端的 IoT Hub 或第三方 SCADA/MES 系统通信。
- **降低系统复杂度：** 减少了中间转换环节，提高了系统可靠性和实时性。
- **标准化数据交换：** 使用 JSON 等通用格式，便于与 IT 系统集成。

这份指南涵盖了从硬件配置到通信测试的全过程。实际应用中，请根据您的网络环境和 MQTT 代理的具体要求调整安全设置和主题结构。