

# 西门子断路器SIEMENS授权代理商

产品名称	西门子断路器SIEMENS授权代理商
公司名称	浔之漫智控技术（上海）有限公司-西门子模组
价格	.00/件
规格参数	西门子:全国代理
公司地址	213室
联系电话	13817547326

## 产品详情

### 西门子断路器SIEMENS授权代理商

西门子中国有限公司授权合作伙伴——浔之漫智控技术(上海)有限公司，作为西门子中国有限公司授权合作伙伴，浔之漫智控技术（上海）有限公司代理经销西门子产品供应全国，西门子工控设备包括S7-200SMART、S7-200CN、S7-300、S7-400、S7-1200、S7-1500、S7-ET200SP等各类工业自动化产品。公司国际化工业自动化科技产品供应商，是专业从事工业自动化控制系统、机电一体化装备和信息化软件系统

集成和硬件维护服务的综合性企业。西部科技园，东边是松江大学城，西边和全球\*\*芯片制造商台积电毗邻，作为西门子授权代理商，西门子模块代理商，西门子一级代理商，西门子PLC代理商，西门子PLC模块代理商，

，建立现代化仓

储基地、积累充足的产品储备、引入万余款各式工业自动化科技产品与此同时，我们

向北5公里是佘山国家旅游度假区。轨道交通9号线、沪杭高速公路、同三国道、松闵路等

交通主干道将松江工业区与上海市内外连接，交通十分便利。

建立现代化仓

储基地、积累充足的产品储备、引入万余款各式工业自动化科技产品，我们以持续的卓越与服务，取得了年销

售额10亿元的佳绩，凭高满意的服务赢得了社会各界的好评及青睐。与西门子品牌合作，只为能给中国的客户提供值得信赖的服务体系，我们

的业务范围涉及工业自动化科技产品的设计开发、技术服务、安装调试、销售及配套服务领域。

目前，将产品布局于中、高端自动化科技产品领域，主要销售西门子PLC模块，西门子交换机，西门子变频器，西门子触摸屏，西门子电机，西门子数控软件，西门子电线电缆，西门子低压产品等等

西门子建议您在执行编程任务时，不要查看安装文件夹中提供的 SDK 解决方案。而应按照本教程的指导，了解如何使用 API。在开发工作应用程序之后，您可以查看提供的解决方案以进行比较并“检查工作”。为了\*大限度地减少要学习的代码量，提供的解决方案对每个 API 操作实施了\*少的结果

检查。在实际操作中，应使用良好的防御性编程技巧并经常检查操作结果。对应用程序进行编程，使其在操作成功时继续执行，并适当地处理错误。示例程序和 API 方法调用使用 Result 对象返回操作状态。在程序逻辑中调用 API 方法后，检查 Result 对象的值。此外，为您的用户添加可见反馈，例如长时间操作的等待光标或进度指示条。

请注意，示例应用程序的用户界面仅使用英语。另请注意，示例应用程序仅运行全套 API 功能 (页 49)的一部分。在您熟悉使用 API 编程后，您可以使用创建其它应用程序所需的任何 API 接口、类别和方法。为简单起见，示例程序对受密码保护的 CPU

不执行任何密码操作。在实际应用中，根据需要需要使用 SetPassword 方法 (页 183)。

目标系统 SIMATIC S7-300 ( CPU 313 或更高版本 ) SIMATIC S7-400SIMATIC C7  
所需的软件包 STEP 7 V5.6 或更高版本  
PG/PC 16 MB  
的主内存配置  
PG/PC 中要求 1.85 MB  
的磁盘空间

1) 取决于 CPU

2) 5 个控制回路

## 更多信息订货信息

Standard PID Control 包括一个参数化工具和若干功能块。

每个编程设备一个授权/参数化工具始终需要 PC

可以根据需要随时将函数块复制到编程设备/PC 上。但是，使用这些函数块的每个 CPU 总是需要一份许可证。

根据不同应用场合必须订购不同的产品：

初始安装：

工具：Standard PID Control，浮动许可证6ES7 830-2AA22-0YX0

功能块：Standard PID Control 标准函数块，单一许可证6ES7 860-2AA21-0YX0

如果已有工具和功能块，则可以重复使用：

功能块：标准功能块 Standard PID Control，单一授权，不带软件或文档学完本章后，您将可以使用 API 提供以下功能：用户用于选择网络接口的选择器 (页 15) 按 IP 地址插入 CPU 的方法 (页 16) 显示和信任 TLS 证书的方法 (页 20) 输入的 CPU 的设备信息 (页 22) 用于更改 CPU 操作模式的按钮 (页 23) 先决条件

要开发您的应用程序，必须满足以下先决条件。运行您的应用程序的用户也必须满足这些先决条件：您通过网络接口将 CPU 连接到您的编程设备。您知道 CPU 的 IP 地址。您可以对网络上的 CPU 执行 ping 操作并验证它是否响应。

### 2.2.1 创建使用 API 的项目

使用 Microsoft Visual Studio 以及 SIMATIC Automation Tool API (页 49) 开发应用程序。您的应用程序和 SIMATIC Automation Tool API.dll 文件必须位于同一文件夹中创建使用 API 的项目。该 API 由西门子使用 Microsoft Visual Studio 2017 和 .NET Framework 4.8 开发。使用这些版本以及该 API 开发 64 位 (x64)

应用程序。本文档中的所有代码示例均采用 C# 编程语言。要创建可以使用 SIMATIC Automation Tool API 的项目，请按照下列步骤操作：

1. 在 Visual Studio 中，创建一个新项目，即 Visual C# Windows 窗体应用程序。将其命名为“我的应用程序” (My Application)。
2. 从“解决方案平台” (Solutions Platforms) 下拉列表中，选择“组态管理器” (Configuration Manager)。
3. 在“组态管理器” (Configuration Manager) 中，单击“平台” (Platform) 下拉列表并创建一个新平台。
4. 从“新建项目平台” (New Project Platform) 对话框，为新平台选择“x64”。单击“确定” (OK)，关闭“组态管理器” (Configuration Manager)。
5. 在项目属性中，将目标框架设置为 .NET Framework 4.8。说明 SIMATIC Automation Tool API 使用 .NET Framework 4.8。如果您的编程设备上没有此框架，则安装程序会安装此框架。

设置项目以使用 API 要设置项目以使用 API，请按照下列步骤操作：

1. 运行 SAT SDK File Extractor，将 API 文件和文件夹复制到默认位置。您可以从“开始” (Start) 菜单中访问 Siemens Automation 组中的 SAT SDK File Extractor。
2. 打开解压缩文件的位置，并将“Bin”文件夹的内容复制到 \bin\x64\Debug 文件夹中。
3. 在项目中添加对 bin\x64\Debug\SIMATICAutomationToolAPI.dll 的引用。
4. 在文件 Form1.cs 中，添加以下语句：`using Siemens.Automation.AutomationTool.API;` 应用程序可通过此语句使用 API dll。您将通过 API 类别和接口在程序中创建对象。
5. 在窗体中添加“关闭” (Close)

按钮，以使用户可以关闭应用程序。

6. 构建您的解决方案并验证它是否成功构建。为用户提供一种选择网络接口的方法

要读取和显示网络接口，并使用户能够选择接口，请按照以下步骤操作：

1. 使用文本标签（如“网络接口”）向窗体添加 ComboBox。

2. 将 ComboBox 命名为“SelectNetworkInterface”。

3. 向 Form1.cs 中的 public partial class Form1 类别输入网络的全局变量：`namespace MyApplication { public partial class Form1 : Form { public Network Net = new Network();`

请注意，“Network (页 87)”是 API 提供的类别。

4. 在 Form1 类别中为全局辅助方法设置一个区域：`#region 辅助方法 #endregion`

5. 在辅助方法的 #region 中添加一个全局空方法：`private void ClearCPUStatus() { }` 稍后您将向此方法添加代码。每当用户选择网络接口时，应用程序将清除任何现有的 CPU

状态信息。双击窗体上的 ComboBox 控件以添加允许用户选择网络接口的方法。在方法中插入如下

所示的代码：`private void SelectNetworkInterface_SelectedIndexChanged(object sender, EventArgs e) { Result res = Net.SetCurrentNetworkInterface((string)SelectNetworkInterface.SelectedItem); }`

此时，编译您的项目并验证它是否已成功编译。测试下拉框以确保它显示可用的网络接

口。确认可以选择每个网络接口。根据需要添加其它错误检查。

如果您对自己的工作感到满意，请转到通过 IP 地址连接到网络上的 CPU (页 16)。

### 2.2.3 通过 IP 地址连接到网络上的 CPU

如果网络接口使用 S7-1200 或 S7-1500 CPU

连接到网络，则下一步是让用户能够连接到网络上的 CPU。随后，该 CPU

将成为设备指令的连接设备。如果您根据提供的示例程序实现您的应用程序，则必须执行以下任务：

1. 接受并验证 IP 地址的输入内容。

2. 将 IP 地址的设备插入设备集合。在此示例中，设备集合将是单个设备的集合。

3. 从与输入的地址对应的网络设备中读取数据。

4. 确认 IP 地址处的设备是 S7-1200 或 S7-1500 CPU。

5. 显示 CPU 的设备信息。本章节将指导您完成每项任务的窗体设计和编程。

说明 此任务序列假定之前的每项任务都已成功。在您的应用程序中，执行验证和错误检查。只

有前一项任务成功后，才能继续执行下一任务。提供输入 IP 地址的文本框 用户在文本框中输入 CPU 的

IP 地址。要为您的用户提供此文本框，请按照以下步骤操作：

1. 在您的窗体上放置一个带有相应标签的文本框控件，例如“IP 地址：” (IP Address:).

2. 将文本框命名为“InsertDeviceIPAddress”。

3. 确保文本框足够宽，能够容纳 IPv4 地址。

4. 将以下全局声明添加到 Form1.cs 中的 public partial class Form1 类别。于在

接受来自可用网络接口的选定接口 (页 15) 中添加的公共网络声明后添加这些声明：`public IProfinetDeviceCollection Devices; public ICPUCurrentCPU;` 请注意，“IProfinetDeviceCollection (页 96)”是

API 提供的类别，“ICPU (页 146)”是接口。本教程应用程序在所有示例中均使用 Devices 集合和

CurrentCPU 对象。5. 在辅助方法区域中，添加名为 ParseIP 的 IP 地址解析器方法。ParseIP 会将基于文本的 IP 地址转换为无符号整数实现处理输入的 IP 地址的按钮 您需要为用户提供输入 IP 地址后可单击的按钮。该按钮尝试通过 IP 地址连接到网络上的 CPU。要实现此按钮，请按以下步骤操作：1. 在窗体上放置一个按钮并将其命名为“InsertDevice”。2. 为按钮编写单击事件：`public void InsertDevice_Click(object sender, EventArgs e)` 3. 在辅助方法的 #region (页 15) 中，添加一个全局空方法：`private void UpdateCPUStatus() {}` 稍后您将向此方法添加代码。4. 在 InsertDevice\_Click 方法内，添加代码以清除任何现有设备数据并清除 CPU 状态：`CurrentCPU = null; ClearCPUStatus();` 5. 调用您的 ParseIP 方法，将输入的 IP 地址从文本转换为 32 位大端无符号整数：`uint ipAddress = ParseIP(InsertDeviceIPAddress.Text);`

在整个应用程序中，根据需要添加错误检查。为简单起见，本教程省略了典型的错误处理代码。使用 IP 地址以连接至 CPU 现在您的窗体上有一个带标签的文本框和相应的按钮，下面编写代码以查找网络上的 CPU。找到 CPU 后，您的应用程序可以从 CPU 读取设备信息。API 提供了设备集合类别，IProfinetDeviceCollection (页 96)。您将把 CPU 插入集合 (Devices)，以便可以在 CPU 上执行后续设备操作。Network (页 87) 类别的 ScanNetworkDevices (页 90) 方法用于将 CPU 输入集合。将设备转换为 CPU (ICPU (页 146) 类别) 并保存引用。应用程序通过 CurrentCPU 在 CPU 上执行所有设备操作：`CurrentCPU = Devices[0] as ICPU;` 5. 更新设备的状态：`UpdateCPUStatus();` 6. 同样，添加对返回的操作结果的适当错误检查。7. 构建解决方案并验证代码是否可编译。

在设备集合中成功插入 S7-1200 或 S7-1500 CPU 后，您的下一个任务是显示 CPU 信息 (页 22)。显示和信任 TLS 证书 如果您的应用程序将支持可通过 TLS V1.3 及更高版本的证书实现安全 PG/PC 通信的 CPU，则必须创建相应的机制，使用户能够显示并设置 TLS 证书的信任等级。ICPU 证书提供显示证书的方法：`ICPU.CertificateStore.ShowDialog()` 和设置证书信任等级的方法：`ICPU.SetTrustCertificateStore()`。选择证书信任等级 提供一个方框，用户可通过此框选择证书信任等级。此框允许客户选择“从不”(Never)、“始终”(Always) 或“需要选择”(Selection needed)。\*后一项为默认设置，将通过组合框完成此选项的设置。要实现设置，请按以下步骤操作：1. 在窗体上放置一个组合框并将其命名为“TrustTLSCertificate”。2. 向项目集合添加三个项目：“从不”(Never)、“始终”(Always) 和“需要选择”(Selection needed)。3. 双击组合框并将以下代码添加到 SelectedIndexChanged 方法显示 CPU 信息

此时在应用程序中，您已实现连接到网络上 CPU 的控件。既然您的应用程序已成功建立此连接，接下来添加控件和逻辑以显示有关设备的信息：添加 CPU 设备信息的标签 将标签控件添加到您的窗体中，并对每个标签进行描述：窗体上的文本 标签控件名称 设备： DeviceName 订货号： ArticleNumber 操作模式： OperatingMode 将每个标签控件的 AutoSize 设置为 FALSE，并将宽度设置得足够大，以容纳每个字段的

\*大预期文本在之前的任务中，您创建了两个空帮助方法：`ClearCPUStatus` (页 15) 和 `UpdateCPUStatus` (页 16)。您的下一个任务是为这些方法添加代码以获取 CPU 设备信息：1. 将以下代码插入 `ClearCPUStatus()`：`DeviceName.Text = string.Empty; ArticleNumber.Text = string.Empty; OperatingMode.Text = string.Empty;` 请注意，当用户尚未连接到 CPU 时，每个字段\*初显示“...”。2. 在 `UpdateCPUStatus()` 中，插入以下代码：`DeviceName.Text = CurrentCPU.Name; ArticleNumber.Text = CurrentCPU.ArticleNumber; OperatingMode.Text = CurrentCPU.OperatingMode == OperatingState.Run ? "RUN" : "STOP";` 回想一下，CurrentCPU 包含您在通过 IP 地址连接到网络上的 CPU (页 16) 中从设备读取的设备信息。当用户单击“插入设备”(Insert Device) 按钮时，InsertDevice\_Click 方法将调用 `ClearCPUStatus` 方法和 `UpdateCPUStatus` 方法。3. 在继续下一步之前编译并测试您的程序：更改 CPU 的操作模式 (页 23) 2.2.6 更改 CPU 的操作模式 到目前为止，您已完成以下任务：接受来自可用网络接口的选定接口 (页 15) 通过 IP 地址连接到网络上的 CPU (页 16) 显示 CPU 信息 (页 22) 此示例程序的下一个任务是添加按钮控件和逻辑以更改操作模式