

# 深入浅出 **WinAC**

## 目 录

前言

第一章 **WinAC** 产品概述

第二章 **WinAC** 站组态和硬件组态

第三章 **WinAC Computing** 和 **NET OPC**

第四章 **WinCC** 访问 **WinAC**

第五章 **WinAC ODK**

第六章 **WinAC MP**

附录 **WinAC** 技术参数

## 前言

在工业和制造业的各个领域向规模化方向发展的过程中，自动化程度的需求越来越广泛，对自控系统的要求也越来越高。每年有大量的 DCS 系统和 PLC 系统装备于各种设备以及自动化系统中。由于工业和制造业对电控、仪控一体化及管控一体化的需求，由于信息技术的发展和高集成度控制系统的需求，以及 PC 技术的完善使其性能价格比日益提高，为一种全新的自控系统解决方案——基于 PC（PC-based）控制的快速发展提供了有利的条件和广阔的前景，也为众多用户提供了除 DCS 和 PLC 以外的另一种选择。

PC-based 控制技术是一种基于 PC 机，配合操作系统(例如 Microsoft 的 Windows2000/XP) 以及一套由不同厂家开发的基于 PC 的控制软件，加上一些必要的 I/O 硬件设备组成的控制系统。该控制系统的开发运行环境放在 PC 机内，其中可包括集成的组态软件、HMI(人机接口)、程序编译运行环境以及必要的调试环境和驱动程序库等。

基于 PC 控制具有以下优点：

(1) 系统开放性能好。软件方面能够支持各种开放性协议，例如:OPC、DDE、COM/DCOM、OLE 等等。硬件方面不仅能够支持各种运动控制产品，而且能够支持各种现场总线系统，例如: DeviceNet, PROFIBUS 等，还可通过 EtherNet 方便地进行网络扩充。

(2) 具有实时控制能力。基于 PC 的控制技术的实时性能主要有三种。一种称为 SoftPLC, 即软件 PLC, 它是利用其内部软件来确定扫描周期，达到实时控制目的。这种性能特点决定了运行 PC 除可完成控制任务外，还可处理大量的 PC 任务。这比较适合于对实时性和响应时间要求不高的系统。另一种是带有实时扩展的实时性能，一般采用 Windows 提供的标准实时扩展解决方案——VenturCOM, 这是一种针对确定性能的控制任务的解决方案。第三种被称为硬实时引擎(Hard Real-Time Engine)，它的控制始终处于最高的优先级，其它的 Windows 任务是中断控制任务的，而这就提供了与 PLC 相同的执行特性和扫描特性。

(3) 具有形式多样的编程语言。

(4) 与人机接口 (HMI) 集成。这样使得上位监控软件与逻辑控制软件结合起来，而且也提高了经济效益。

(5) 数据处理能力强。交互方便，快速可靠，而且随着 PC 机的进一步发展，这种能力的优势将会越来越突出。

# 第一章 WinAC 产品概述

## 1.1 WinAC 产品分类

SIMATIC WinAC 是西门子公司开发的基于 PC 控制的核心组件，它的出现扩展了 SIMATIC S7 的控制范围。WinAC 是一个名副其实的控制中心，它将 PLC 控制、数据处理、通讯、可视化及工艺集成于一台 PC 机上。

SIMATIC WinAC 产品包括软件型和插槽型两大类，包括如下 5 种产品：

### 1. WinAC Basis (WinAC 基本型)

WinAC Basis 是低成本解决方案，用于对控制无精确时间要求，有大量、快速的数据处理与控制任务(控制任务指 PLC 的控制功能)相结合或其它 PC 任务的控制场合。

### 2. WinAC PN

第一个支持 PROFINET 通讯标准的 SIMATIC CPU，性能与 WinAC Basis 相似。WinAC PN 支持基于组件的自动化(CBA)和 PROFINET 通讯标准。基于组件的自动化和 PROFINET 提供了一个开放的标准，用于在复杂任务中机械和系统单元之间的数据交换。数据交换通过 SIMATIC iMap 工具来进行配置。WinAC PN 适应于以下任务：

- 来自整个复杂系统的机械和车间区域之间的协调和连接
- 控制系统单元或机械可被集成到一个全范围的复杂系统

WinAC Basis 4.1 有一个选件 WinAC PN，带有 WinAC PN 选件的 WinAC Basis 4.1 支持 PROFINET 和 CBA。

### 3. WinAC RTX(WinAC 实时型)

WinAC RTX 提供了 Windows 2000/XP 的实时子系统，具有“硬实时”和“抗死机”特性。适应于具有高速和精确时间要求的控制任务的场合，如运动控制、闭环控制等。

### 4. WinAC MP

WinAC MP 基于 WinCE 操作系统和 SIMATIC MP370(一种多功能面板)硬件平台。MP370 为无硬盘、无风扇设计，WinCE 具有实时特性，可实现严格的确定性动作。WinAC MP 用在恶劣工业环境和有大量数据要处理的场合。

### 5. WinAC Slot 412/416

以板卡的形式插入在 PC 中，在板卡上已经集成了用于控制任务的 CPU、存储器等元件。它可独立于 PC 进行控制操作。板卡上集成一个 MPI/DP 接口和一个 DP 口。WinAC Slot 适用于对安全性和稳定性要求较高的场合。WinAC Slot 412/416 在性能上与 S7-412/416 相近。

## 1.2 WinAC 特性和适应场合

由于其功能强大的开放和灵活的接口，SIMATIC WinAC 成为基于 PC 自动化解决方案的理想平台。

- WinAC 运行在基于 Windows 2000/XP 操作系统下的标准 PC 平台中
  - 与 SIMATIC S7 有一致的代码兼容性，统一的编程环境，生成的程序也可被 SIMATIC S7 所使用
  - 提供了 OPC、ActiveX 控件、DCOM 接口等方法，与其它 PC 应用程序交换数据
  - 为具有特殊工艺硬件软件相结合的控制任务提供了开放的接口
- WinAC 适用于以下任务：
- 由于空间和性能的原因，数据处理、通讯、可视化和控制任务必须集成在同一台 PC 平台上
  - 需要有高性能数据处理的任务
  - 在协调控制功能中必须有被执行的特殊过程任务
  - 存在特殊硬件软件模块集成的任务

### 1.3 WinAC Computing

WinAC Computing 是所有 WinAC 产品系列中的一个组件，并对数据处理提供了两个重要的接口：

- WinAC OPC 服务器开放了 WinAC 对过程数据的访问，用于数据处理和实现可视化
- ActiveX 组件支持从生产过程到特殊软件应用或办公室应用的功能强大的连接

WinAC Basis/RTX 4.1 和 WinAC Slot 3.4 版本支持 SIMATIC S7 控制器统一的 SIMATIC NET OPC 服务器。

### 1.4 WinAC 开放工具 ODK

WinAC ODK 4.1 包括三个组件：

#### 1. CCX（定制代码扩展）：

将 C/C++ 程序直接嵌入到 WinAC Basis/RTX 的 Step7 程序中，可用于连接第三方软件和硬件，由此可提供 PLC 任务所需的所有 PC 上的操作系统功能和系统资源。

#### 2. SMX（共享内存扩展）：

支持与 WinAC 进行高速（速度是 OPC 交换速度的百倍以上）数据交换的应用程序的开发，确保与技术功能应用的紧密协同。

#### 3. CMI（控制管理接口）：

使用 CMI 可使应用程序监控和改变 WinAC 的特性。

## 第二章 WinAC 站组态和硬件组态

本章主要介绍 WinAC Basis 和 WinAC Slot 的站组态和硬件组态。WinAC RTX 的组态与 WinAC Basis 相类似，因此不做专门介绍。

### 2.1 WinAC 站组态

在安装好 SIMATIC NET、WinAC 之后就可以进行 WinAC 站组态。如果 WinAC Basis 需要连接外部 I/O，至少需要一块 CP5611 或 CP5613 通讯卡。下面的例子用到了如下的软件和硬件：

1. WinAC Basis V4.1
2. SIMATIC NET 6.2 (随 WinAC Basis 4.1 一起供货)
3. CP5613
4. Step 7 V5.3

Step7 可以与 WinAC 装在同一台 PC 机上，也可以装在不同的 PC 机上。如果 WinAC 与 Step7 安装在同一台 PC 机上，则除了 Step7 外不需要其它的编程软件和硬件。如果 Step7 安装在另外的 PC 机上，则装有 Step7 的 PC 上还需要另外的设备，如 CP5511。下面步骤描述的是 WinAC 与 Step7 安装在不同的 PC 机上，两台 PC 机都有以太网卡（且两网卡已物理连接好），硬件和程序的下载通过以太网进行。

进行 WinAC 站组态的步骤如下：

- 一. 在“Station Configuration Editor”中添加“WinLC”

使用 WinAC “Station Configuration Editor”和“Configuration Console”来配置 WinAC 的 PC 站。

WinAC Basis 安装好后，在“Station Configuration Editor”中自动把“WinLC”加入到索引号为 2 的 PC 站中，如图 2.1 所示。（如果没有自动添加可从弹出式菜单中选择“Add”，并从随后出现的“Add Component”对话框中选择“WinLC”。）索引是 PC 站虚拟机架中一个带编号的虚拟插槽（它与 PC 主板的实际 PCI 插槽顺序无关）。PC 站为基于 PC 的自动化解决方案中的 SIMATIC 组件提供插槽。其中不仅包括 WinLC 和 CP 卡，而且包括 SIMATIC HMI 和 SIMATIC NET OPC。

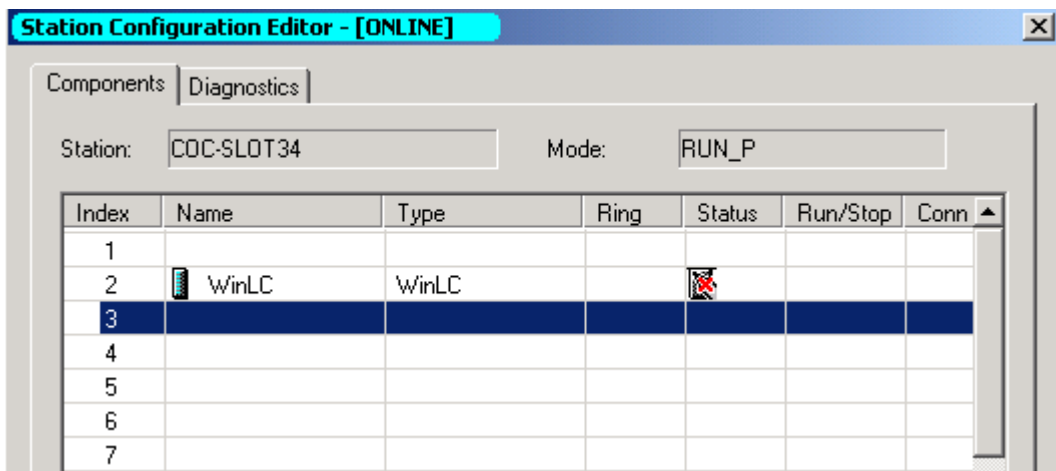


图 2.1 添加了“WinLC”后的站组态编辑器

## 二. 配置 CP5613 为组态方式

按如下步骤来将 CP5613 卡配置为组态方式

1. 鼠标单击“开始”->“SIMATIC”->“SIMATIC NET”->“Settings”->“Configuration Console”，打开组态控制台。

2. 在“Configuration Console”窗口中依次单击“Modules”和“CP5613/CP5614”前的符号“+”后选择“General”，在右边数据窗口中将“Mode of the module”的列表框选为“Configured mode”，列表框“index”选为 4(可以为其它值)，如图 2.2 所示。

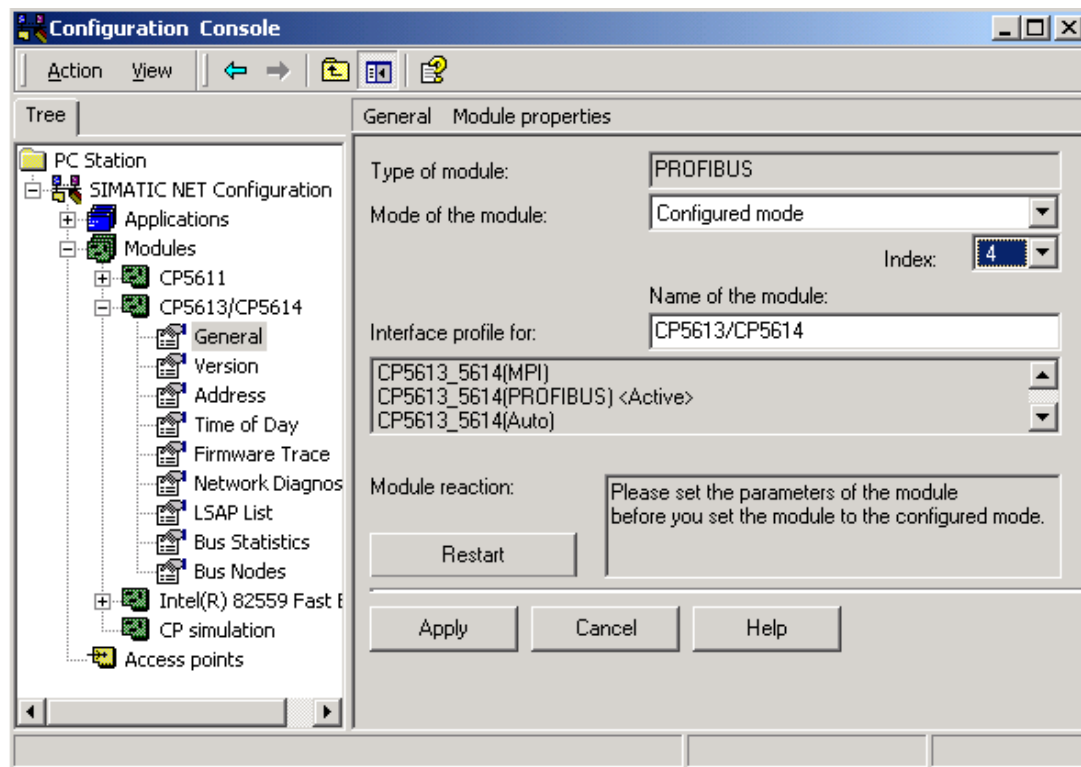


图 2.2 配置 CP5613 为组态方式

单击按钮“Apply”并关闭“Configuration Console”程序。

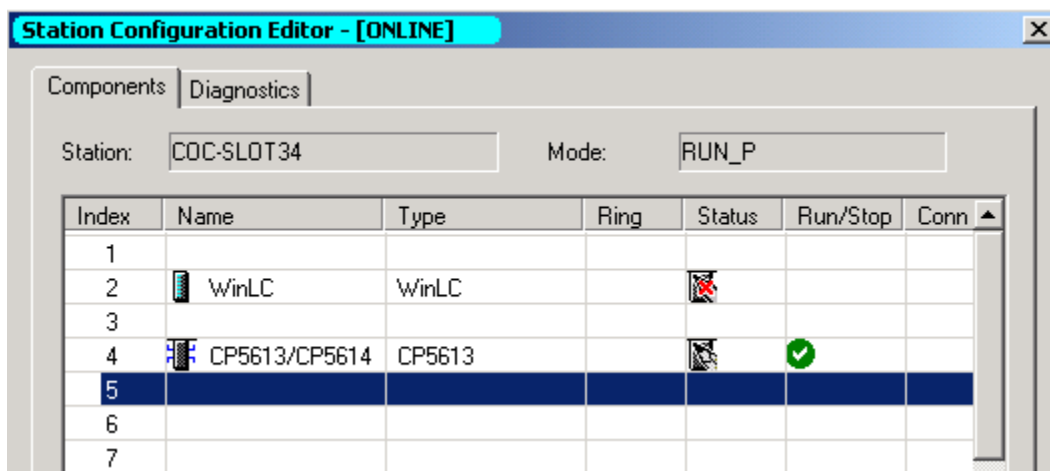


图 2.3 CP5613 设置了组态方式后的站组态编辑器

重新打开“Station Configuration Editor”，如图 2.3 所示，我们看到 CP5613 位于索引号为 4 的 PC 站虚拟插槽中。

### 三. 将已设置为组态方式的 CP5613 设为 WinLC 的子模块

要使 WinLC (WinAC Basis 的控制器) 可以带各种 DP 从站，将 CP5613 设置为组态方式后，还要将其设为 WinAC 的子模块。每个 WinLC 最多有 4 个子模块。

在如图 2.3 所示的站组态编辑器中，右键单击索引号为 2 的行 (即 WinLC 行)，从快捷菜单中选择“Edit”，打开“Edit Component”对话框，在此对话框中选择“Properties...”，打开“WinLC Properties”对话框，在此对话框中右键单击索引号为 IF1 的行，并从快捷菜单中选择“Add”，如图 2.4 所示。

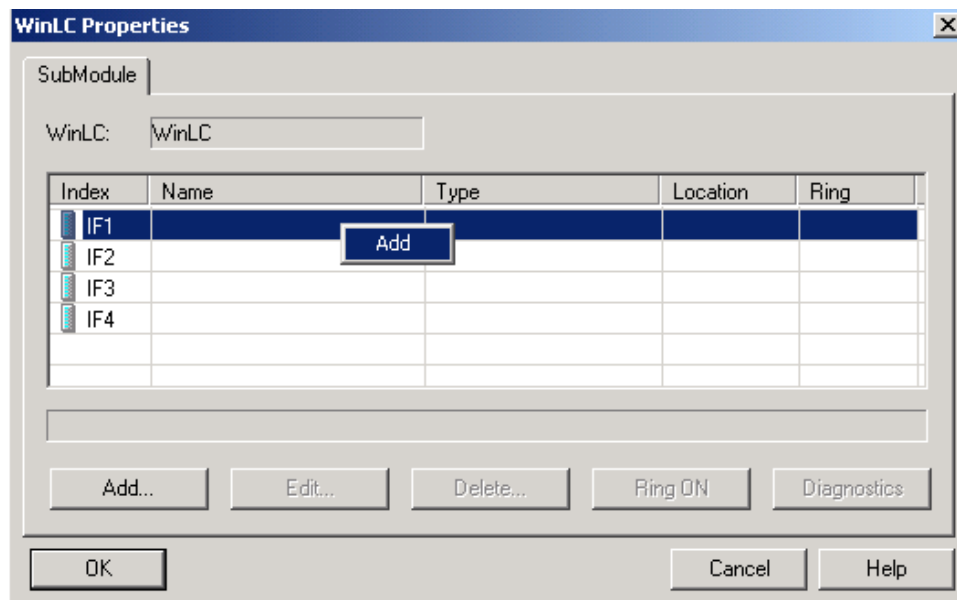


图 2.4 添加 WinLC 的子模块

在打开的“Add CP Module to WinLC Interface IF1”对话框中选择 CP5613。如图 2.5 所示，单击“OK”，关闭“Add CP module To WinLC Interface IF1”对话框，回到“WinLC Properties”窗口，我们看到 CP5613/CP5614 位于 IF1 索引号上，如图 2.6 所示，单击“OK”后。至此基本的 WinAC PC 站组态完毕。当再次打开“Station Configuration Editor”后，便发现 CP5613 已不在索引号为 4 的插槽上，这是因为 CP5613 已被作为 WinLC 的子模块插入到 WinLC 的“插槽”中。

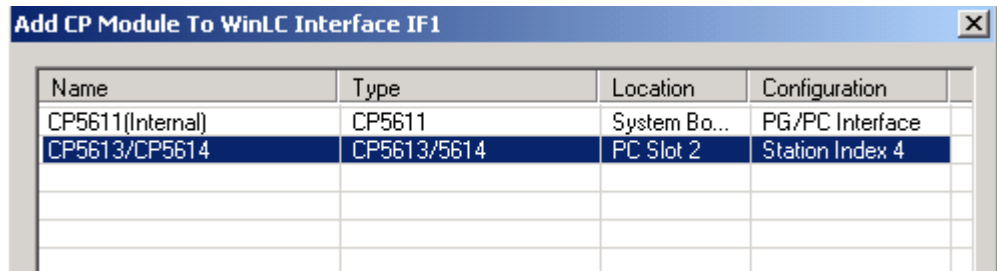


图 2.5 选择 CP5613 作为 WinLC 的子模块

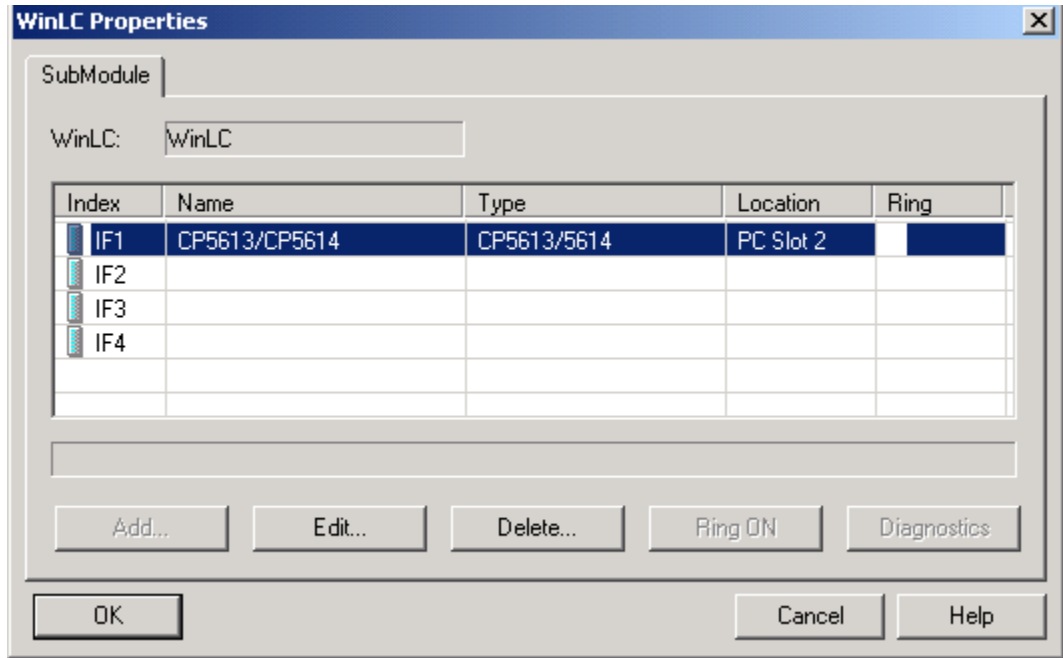


图 2.6 添加子模块后的 WinLC 属性

#### 四. 在 PC 站中添加实现其它功能的模块

上述的 PC 站只组态了 WinAC，如果要使 WinAC 支持 SIMATIC NET OPC 还应在 PC 站的插槽中插入“OPC Server”。再次打开“Station Configuration Editor”，右键单击索引号为 1 的行从快捷菜单中选择“Add”，打开“Add Component”对话框，添加“OPC Server”，如图 2.7 所示。如果不使用 SIMATIC NET OPC 服务器，而只使用 WinAC OPC 服务器则无需添加此项。



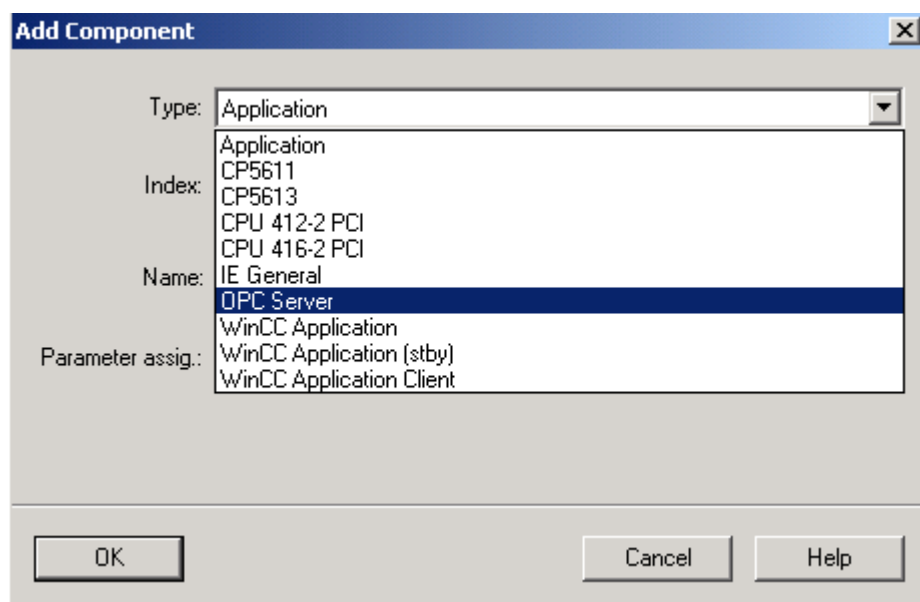


图 2.7 添加 SIMATIC OPC 服务器

如果需要使用以太网与其它 S7 控制进行通讯或使用以太网来下载 Step7 程序，在 PC 站的插槽中还应加入“IE General”（使用此项功能需要一个 Softnet 的授权）。按照同样的方法把“IE General”添加在索引号为 5 的虚拟插槽中。最后的 PC 站组态如图 2.8 所示。

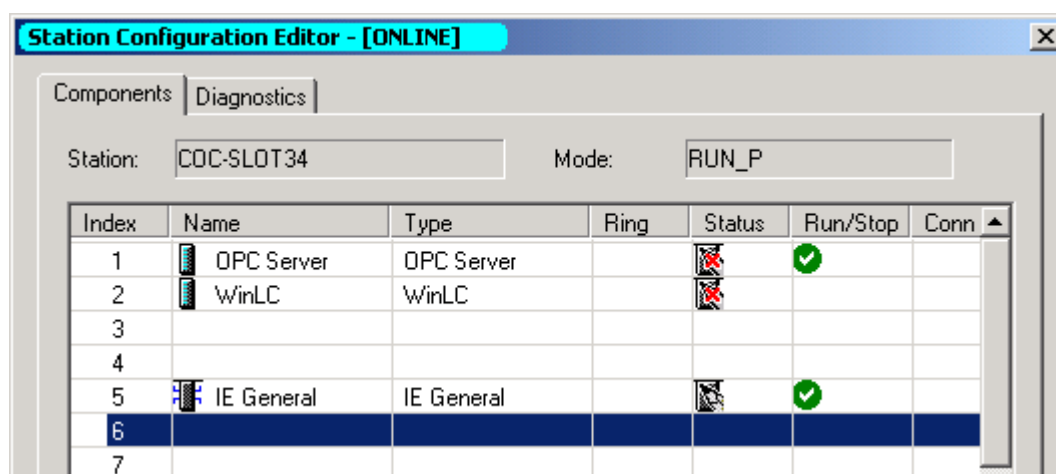


图 2.8 添加 SIMATIC OPC 服务器和 IE General 后的 PC 站

至此，我们组态了一个 PC 站，它包括一个 SIMATIC NET OPC 服务器、一个带有 CP5613 子模块的 WinLC、一个可以与其它 S7 控制器或编程器进行通讯的以太网卡。上面步骤说明的如何组态 WinAC Basis。由于 WinAC Slot 不需要另外的 CP 卡，因此组态 WinAC Slot 时不需要上述步骤的第二步、第三步。另外默认时 WinAC Slot 位于索引号为 3 的插槽上。按照同样的方法组态 WinAC Slot412 如图 2.9 所示

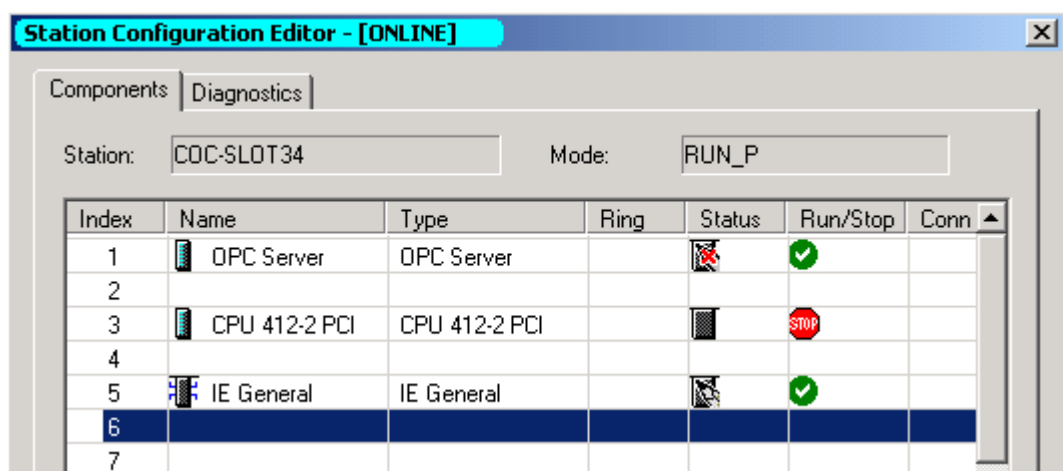


图 2.9 WinAC Slot 的 PC 站组态

## 2.2 用 Step7 进行 WinAC 的硬件组态

对于 WinAC, 不能从 PC 站上上载正确组态信息到 Step7 中, 必须手工在 Step7 中完成硬件组态。在 STEP 7 创建的 WinAC 硬件组态, 它应和 PC 站组态编辑器中的组态相吻合。

### 一. 插入 SIMATIC PC Station

启动 SIMATIC Manager, 创建一个新的工程, 并在此工程下插入一个“SIMATIC PC Station”, 如图 2.10 所示。

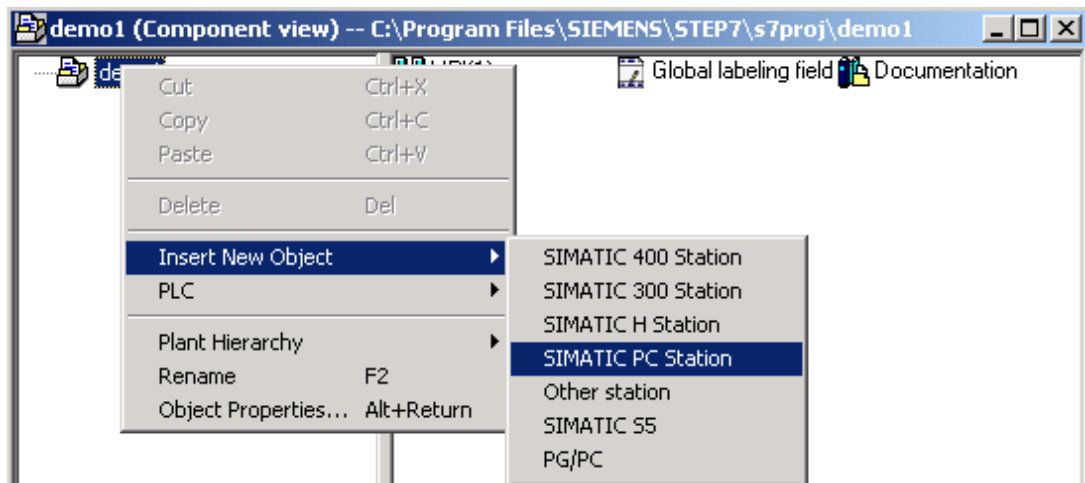


图 2.10 插入 SIMATIC PC 站

将默认名“SIMATIC PC Station(1)”改成图 2.8 中的站名“COC-SLOT34”。在 SIMATIC Manager 左边浏览窗口中选择站点“COC-SLOT34”, 双击右边数据窗口的“Configuration”, 打开硬件组态编辑器。

### 二. 硬件组态

1. 在硬件组态窗口中选择右边的硬件目录, 并从中选择 WinLC 4.1 并把它拖放到左边的第二个插槽上(插槽号对应 PC 站组态编辑器中的索引号), 如图 2.11 所示。

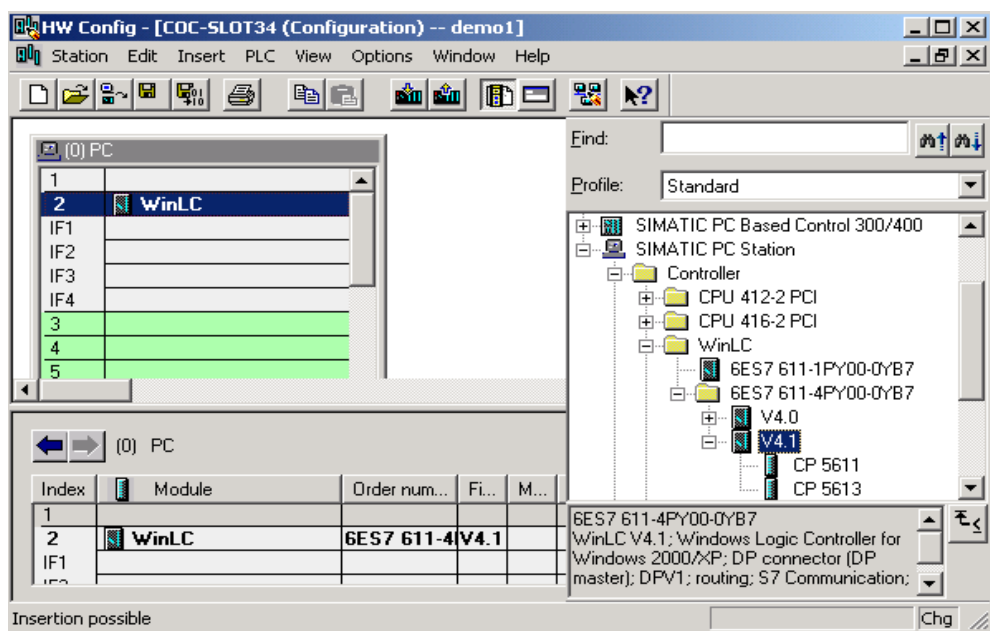


图 2.11 添加 WinLC 4.1

2. 在图 2.11 所示的硬件目录中选择 V4.1 下的 CP5613，并将它拖放到左边 WinLC 下的 IF1 插槽上。接着自动弹出“Properties-PROFIBUS CP5613”对话框，为该 CP5613 卡选一个地址，并单击此对话框上的“new”按钮新建一 PROFIBUS 总线，选择此 PROFIBUS，单击按钮“Properties...”可设置总线的参数如通讯速率、协议类型、最大站点地址等。我们将上面三个参数分别设置成 1.5M, DP, 10。

3. 在右边硬件目录中选择“OPC Server”（在“SIMATIC PC Station”->“User Application”->“OPC Server”下），并将它拖放到左边的第一个插槽中。

4. 打开 WinAC 所在 PC 机上的“Configuration Console”，如图 2.12 所示，记下这台 PC 机的以太网卡的地址，图上所示的以太网的 MAC 地址为“08-00-06-24-C4-E3”。

5. 回到 Step7 所在的 PC，在硬件组态窗口中选择硬件目录中的“IE General”并将它拖放到第五个插槽中，弹出“Properties-Ethernet interface IE General”对话框，单击按钮“new”添加一以太网网络，选中“Set MAC address/use ISO protocol”复选按钮，输入如图 2.12 所示的 MAC 地址，取消“IP protocol is being used”复选按钮，结果如图 2.13 所示，单击“OK”关闭配置设置以太网卡属性对话框。

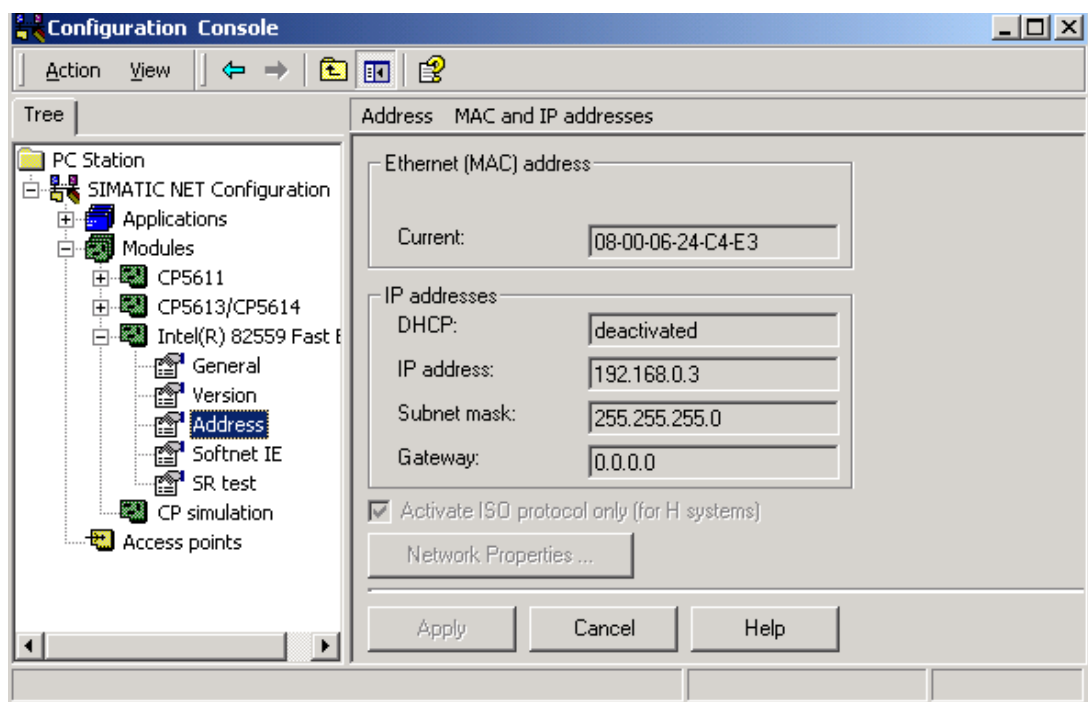


图 2.12 WinAC 所在 PC 机网卡的 MAC 地址

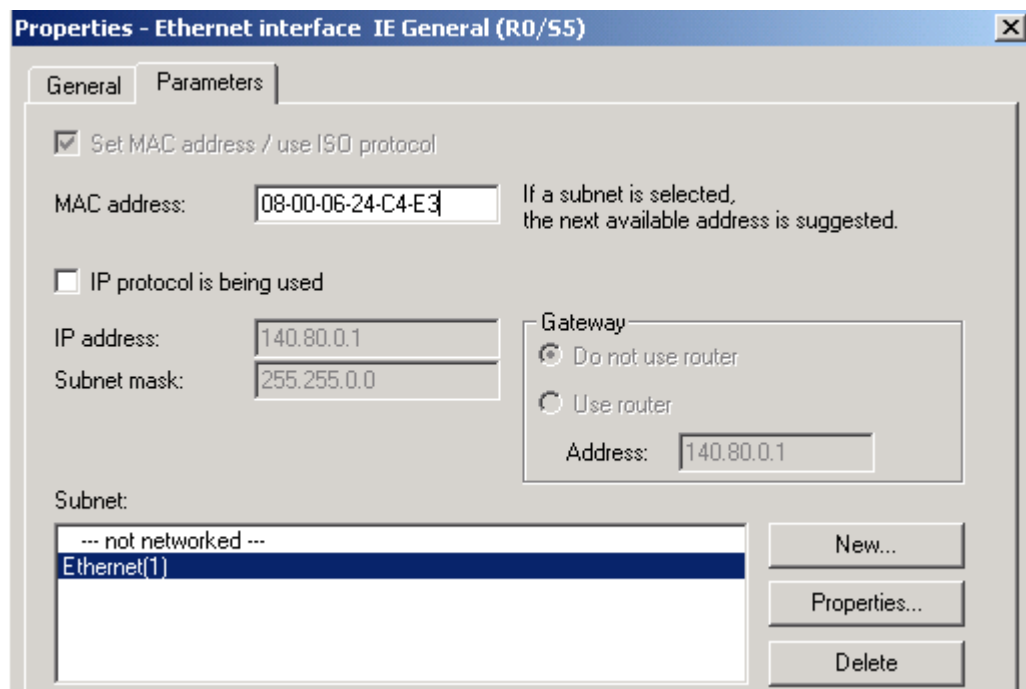


图 2.13 设置以太网卡属性

整个硬件组态结果如图 2.14 所示

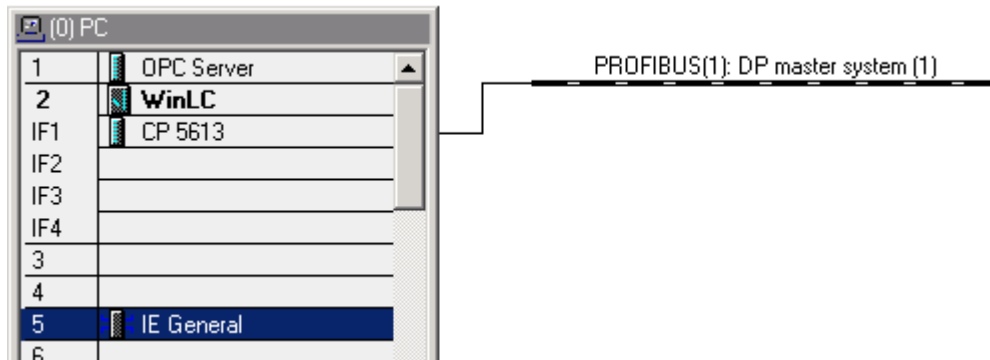



图 2.14 硬件组态图

6. 根据实际硬件情况，选择图 2.14 中的 PROFIBUS(1) 总线，添加各种 DP 从站。单击硬件组态窗口中的按钮  (或选择菜单 “Station->Save and Compile”) 保存并编译组态信息。

7. 在 WinAC 所在的 PC 机上启动 WinLC。

8. 在 SIMATIC Manager 程序中，选择菜单 “Options” -> “Set PG/PC interface” 打开 Set PG/PC interface” 对话框，将 “S7ONLINE (Step7)” 的访问点设置成 “ISO Ind. Ethernet”，如图 2.15 所示，如果 WinAC 与 Step7 装在同一台 PC 机，也可将访问点设置成 “PC internal(local)”，单击 “OK” 关闭此对话框。

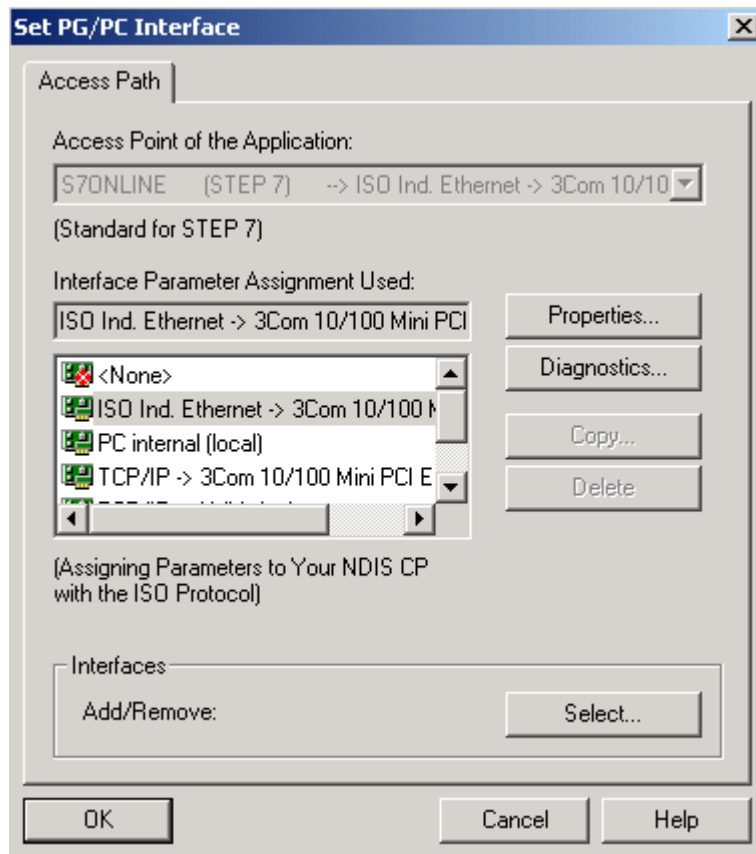



图 2.15 设置 S7ONLINE 访问点

10. 单击硬件组态程序工具栏上的按钮“”将硬件组态信息下载到 WinAC 中。


当把组态信息正确下载到 WinAC 站后，便可如 s7-300/400 一样进行编程，可以添加各种 OB 块、FB 块、FC 块、DB 块等，可以使用 WinAC 支持的各个 SFB/SFC，可使用 SIMATIC Manager 中的各种编程语言。

### 2.3. 网络组态

如果需要 WinAC 支持 SIMATIC NET OPC 服务器或是 WinAC 需要与 S7-300/400 控制、其它 WinAC 等进行通讯，则还应进行网络组态，Step7 使用软件 NetPro 网络组态。

下面的步骤描述了如何组态 SIMATIC NET OPC 服务器到 WinAC 的连接，以及 WinAC 与 S7-300 建立连接。假定我们已在同一个工程下组态了一个 S7-300 站，此站上使用了一个 CPU315-2 DP 和一个通讯处理器 CP343-5。

#### 一. 组态 SIMATIC NET OPC Server 与 WinAC 的连接

单击 SIMATIC Manager 工具栏上的按钮，打开网络组态程序“NetPro”。在网络窗口中选择 PC-Station 站(此处是“COC-SLOT34”)上的 OPC Server 后将鼠标指向窗口下部的连接区，从快捷菜单中选择“Insert New Connection”项，弹出“Insert New Connection”对话框，如图 2.16 所示，选择“WinLC”作为通讯伙伴，选择连接类型为“S7

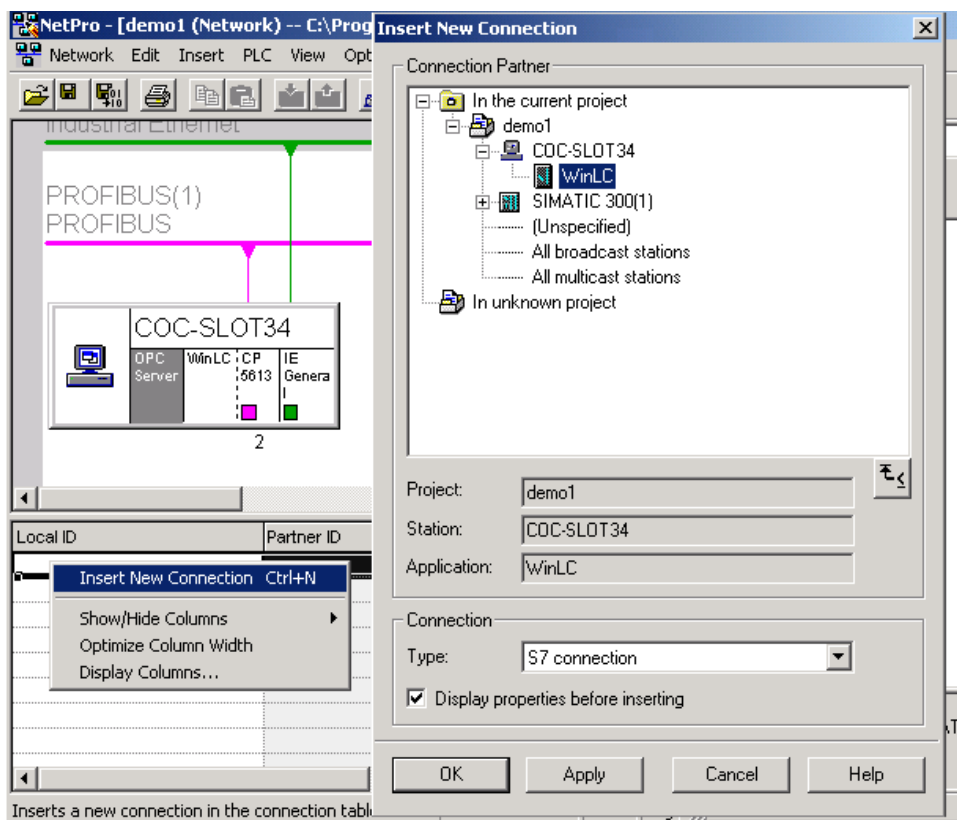


图 2.16 建立 SIMATIC NET OPC Server 到 WinAC 的连接

connection”，单击“OK”，在随后程序的“Properties S7 Connection”窗口中直接单击“OK”退出连接窗口。

通过上面的方法就已建立了 OPC Server 和 WinAC 的连接，可以使用 SIMATIC NET 带的程序 OPC Scout 查看连接对象，可以编写 OPC 客户端程序访问 OPC Server 服务器。

## 二. 组态 WinAC 到 S7-300 的连接

在网络组态的窗口中选择 CPU “WinLC”，单击窗口下部的网络连接区的空白处，从弹出菜单中选择“Insert New Connection”，在随后出现的窗口中选择通讯伙伴为“CPU 315-2 DP”，连接类型为“S7 connection”，单击“OK”，接着弹出“Properties s7-connection”对话框，如图 2.17 所示。

在 WinAC 与 S7-315 的连接中，WinAC 上使用的通讯处理器为 CP5613，此 CP5613 即作为 WinLC 的子模块来连接 WinAC 的 DP 从站，又作为一个主站实现与 S7-300 的主-主通讯。而在 S7-300 上实现此次连接的通讯处理器为“CP 343-5”。在连接属性中还有一个参数称为“Block Parameters”，图中为 2，在编写通讯程序时需要用到这个数值。点击“OK”关闭此对话框。

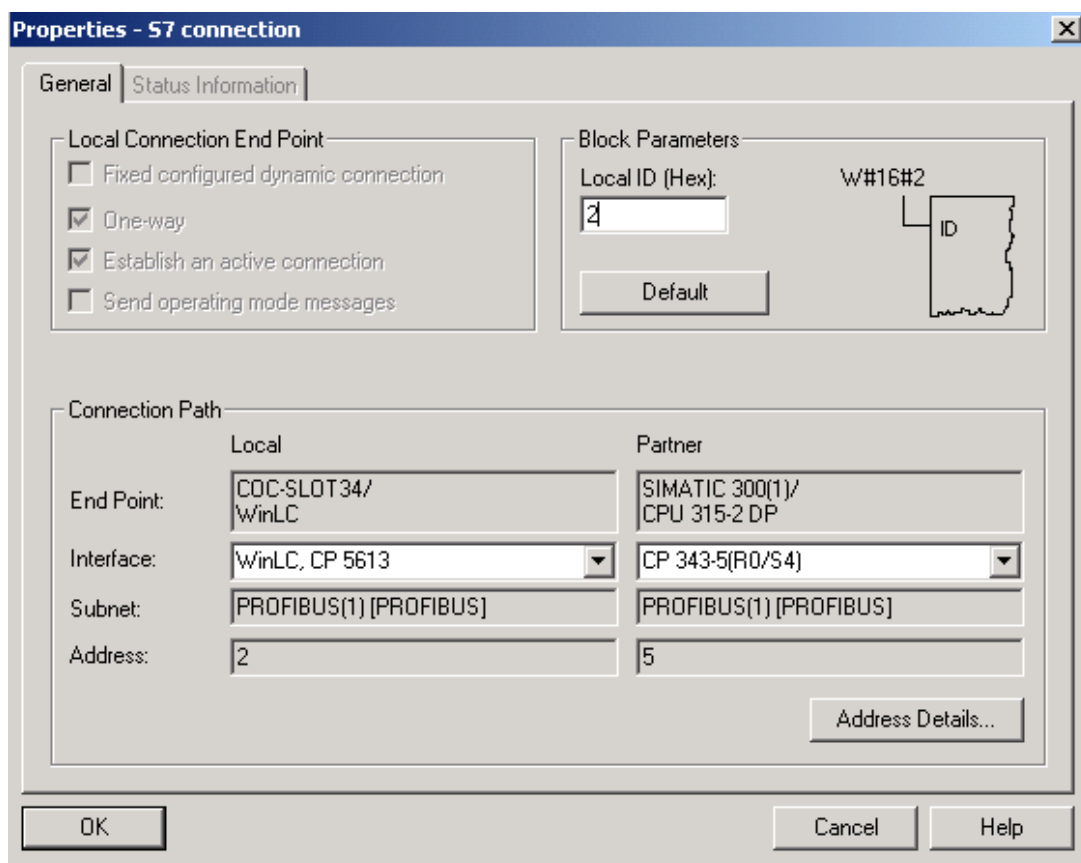


图 2.17 WinLC 与 CPU315-2 DP 的连接属性

至此，我们已在 OPC Server 与 WinAC 之间建立了一个连接，在 WinAC 与 S7-315 之间也建立一个连接。连接后状态如图 2.18 所示。

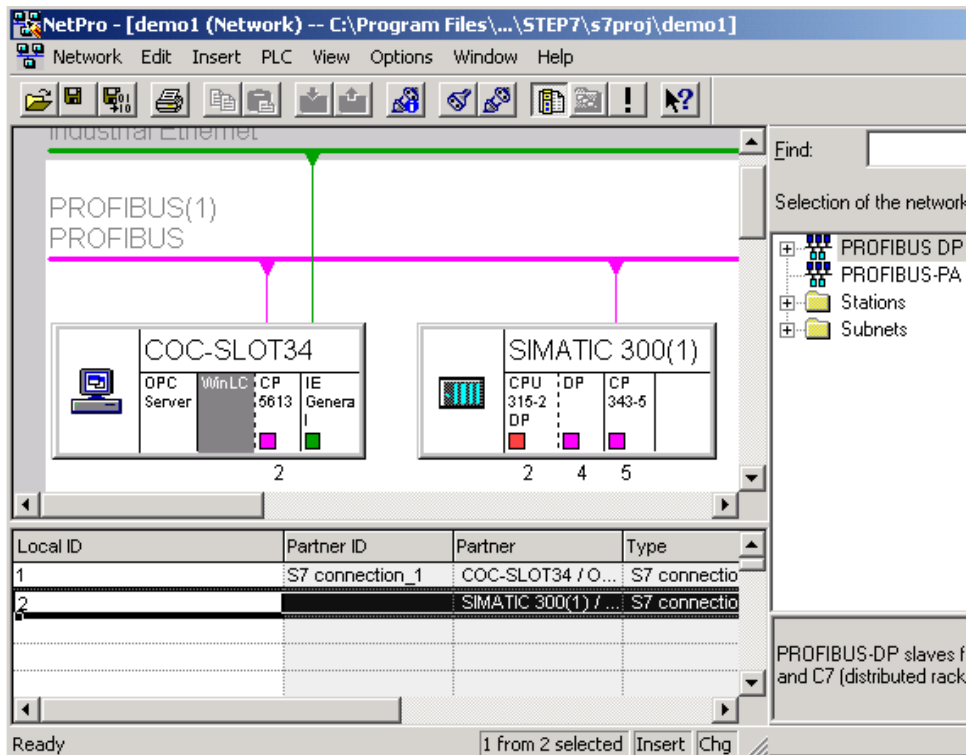


图 2.18 WinLC 与其它通讯伙伴的连接

网络组态好后，应编译和存盘。要将所组态的连接信息下载到站点，可点击菜单“PLC→Download to current project→Connections and Gateways”来完成。

### 三. 通讯编程

建立连接后，便可在程序中调用通讯类型的系统功能块来实现 WinAC 与通讯伙伴间的数据交换。可使用 BSEND(SFB 12)、BRCV(SFB 13)、USEND(SFB 8)、URCV(SFB 9)、GET(SFB 14)、PUT(SFB 15)等系统功能块。下表为这些通讯类型的简要描述。

通讯类型	SFBs	描述	最大数据
BSEND BRCV	SFB 12 SFB 13	带应答的数据传送到远程方。发送方必须执行 BSEND，接收方必须执行 BRCV。	64K
USEND URCV	SFB 8 SFB 9	不带应答的数据传送到远程方。发送方必须执行 USEND，接收方必须执行 URCV。	400 bytes
GET	SFB 14	从远程 CPU 读数据，不需要在远程 CPU 调用另外的通讯程序。	400 bytes
PUT	SFB 15	将数据写到远程 CPU，不需要在远程 CPU 调用另外的通讯程序。	400 bytes
CONTROL	SFC 62	S7 通讯连接的查询状态	N/A

如何调用和使用这些系统功能块，请参看相应的手册。



## 2.4 WinAC 操作面板

无论是软件型 WinAC 还是插槽型 WinAC，安装完成 WinAC 后，都出现一个 WinAC 操作面板，在此面板上包含菜单、按钮、指示灯等。通过此面板，可监控 WinAC 上的总线状态、有无内部和外部错误，以及显示诊断缓冲区。图 2.19 即为 WinAC 的操作面板，左半部分为 WinAC Basis 的操作面板，右半部分为 WinAC Slot 的操作面板。

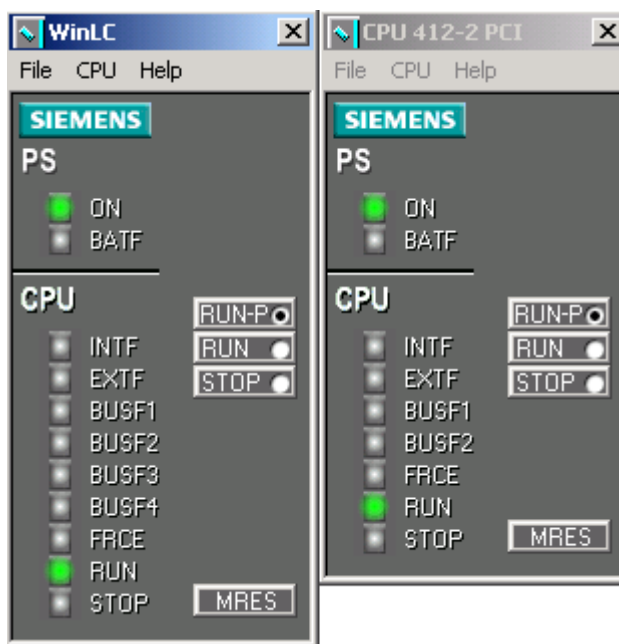


图 2.19 WinAC 的操作面板

### 1. 按钮

面板上有四个按钮：

RUN-P：将 WinAC 设置成运行状态，并可在线编程。

RUN：设置 WinAC 为运行状态，不可编程。

STOP：将 WinAC 设置成停止运行。

MRES：清除控制器中的所有内容，包括硬件组态信息、程序部分、数据部分。

### 2. 指示灯

ON：表示控制器已上电。

BATF：指示 WinAC Slot 的外部电池状态，当电池失效时，显示红色。对于 WinAC Basis，BATF 一直都是不亮。

INTF：控制器内部错误，如程序错误、定时器或计数器错误等。

EXTF：控制器外部故障，如硬件故障、参数分配错误、通讯故障、I/O 故障等。

BUSF1-BUSF4：表示 WinAC Basis 子模块的通讯总线或 WinAC Slot 集成通讯口的通讯总线上是否有总线故障，例如总线站点地址是否冲突，或不存在配置的站点地址等。WinAC Basis 可带四个子模块，而 WinAC Slot 只有两个集成的 DP 口。

FRCE: 当在 Step7 上对变量进行强制(Force)时, 此指示灯亮, 呈黄色, 对于 WinAC Basis 此指示灯无效。

RUN: 常亮时表示 WinAC 已在运行状态。

STOP: 常亮时表示 WinAC 已处于停机状态。当 STOP 常亮, 而 RUN 指示灯在闪烁时, 表示程序运行到某一断点处。

### 3. 菜单

#### (1) File:

Archive: 存储当前控制程序, 当前系统配置, 以及当前 DB 块的数值到 .wld 文件。

Restore: 重新装载以前归档文件到控制器中, 包括系统配置、控制程序以及当前 DB 块的数据。

注意: 需将控制器设置成 Stop 工作方式时才能操作这两个菜单项。

Minimize: 将窗口最小化。

Exit: 退出。点击 Exit 按钮时, WinAC Basis 和 WinAC Slot 表现不同。

- WinAC Basis: 如果 WinAC 已上电, 单击此菜单项将使窗口放置在 Windows 的托盘中, 双击托盘上的图标可重新打开窗口。如果 WinAC 已关机(已执行“Shut Down Controller”), 将退出程序和关闭窗口。

- WinAC Slot: 关闭窗口, 不影响控制器的控制。

#### (2) CPU:

RUN-P、RUN、STOP 与 MRES 同面板按钮功能相同。

Tuning Panel: 打开/关闭 CPU 运行参数调节面板, 可设置 WinAC 优先级、最小睡眠数据等。显示 WinAC 占用的 CPU 时间、WinAC 执行时间和循环时间等。WinAC Slot 无此项。

Diagnostic Buffer: 打开/关闭诊断缓冲区窗口。

Start Controller: 将控制器上电。WinAC Slot 的对应菜单指令为 Power On。

Shut Down Controller: 将控制器断电。WinAC Slot 的对应菜单指令为 Power Off。

Register Controller for Start at PC Boot: 设置成在系统重启时自动使控制器上电。

Unregister Controller for Start at PC Boot: PC 重启时将不会自动启动 WinAC。

以上两项只有在控制器断电时才可操作, WinAC Slot 无此两项。

Option: 定制 WinAC 操作和设置操作 WinAC 的密码。

## 第三章 WinAC Computing 和 NET OPC

WinAC 提供了一套组件：WinAC Computing。WinAC Computing 提供了访问 WinAC 控制器的接口，它包含如下的部分：

- 一组 ActiveX 控件。它提供了对 WinAC 控制引擎中数据的访问。
- 一个 OPC 服务器。它允许 OPC 客户端程序与 WinAC 控制器进行数据交换。即 WinAC 支持两种 OPC 服务器：WinAC Computing OPC 服务器和 SIMATIC NET OPC 服务器。
- 一个变量表编辑器 TagFile Configurator。借助 TagFile，访问 WinAC 控制器数据时可以使用符号名。
- 一个设置程序。它用于对远程访问数据进行设置。
- 一个 OLE 容器 Computing SoftContainer。

尽管 WinAC Computing 使用更简单，但由于 WinAC 新版本支持 SIMATIC NET OPC，而且 NET OPC 具有更好的性能和更强的功能，我们推荐使用 NET OPC 来访问 WinAC 控制器的数据。安装 WinAC Basis 4.1 和 WinAC Slot 3.4 时，并不会自动安装 WinAC Computing，如果需要安装 WinAC Computing，可打开 WinAC 安装盘上的“WinCP”文件夹，运行此文件夹下的“Setup.exe”进行安装。

### 3.1 WinAC Computing 设置

#### 1. 设置 Computing 的访问接口

使用 WinAC Computing 前，应对 Computing 的访问点进行设置。启动 SIMATIC NET 下的设置程序“Configuration Console”，选择“Access points”，双击右边数据窗口的“Computing”行，从打开的对话框中选择“PC internal(local)”，单击“OK”关闭对话框，结果如图 3.1 所示。

WinAC 控制器和 WinAC Computing 可以安装在不同的 PC 机上，以支持远程访问 WinAC 控制器的数据。不论 WinAC 控制器与 WinAC Computing 是否安装在同一 PC 机上，都要将 Computing 的访问接口设成“PC internal(local)”。

#### 2. Computing Configuration

使用 WinAC Computing OPC Server 前应设置 OPC 服务器。

点击“start”->“Simatic”->“PC Based Control”->“Computing Configuration”打开 Computing 的设置程序，在“OPC”选项卡上有一连接选择项，选择是通过变量文件连接还是直接连接，如图 3.2 所示。如果通过标签文件连接，在这之前还应使用“Computing TagFile Configurator”创建标签文件。选择直接连接有两项内容要求输入：

- Computer name：此处输入的是被访问的 WinAC 控制器的计算机名，也可以是计算机的 IP 地址，如果访问本机 WinAC 控制器的数据，“Computer name”的值也可以是“<Local>”。

- Control Engine: WinAC 控制器的类型。  
WinAC Basis: 输入 “WinLC”。
- WinAC RTX: 输入 “wcS7=2”。
- WinAC Slot: 输入 “wcS7=3”。

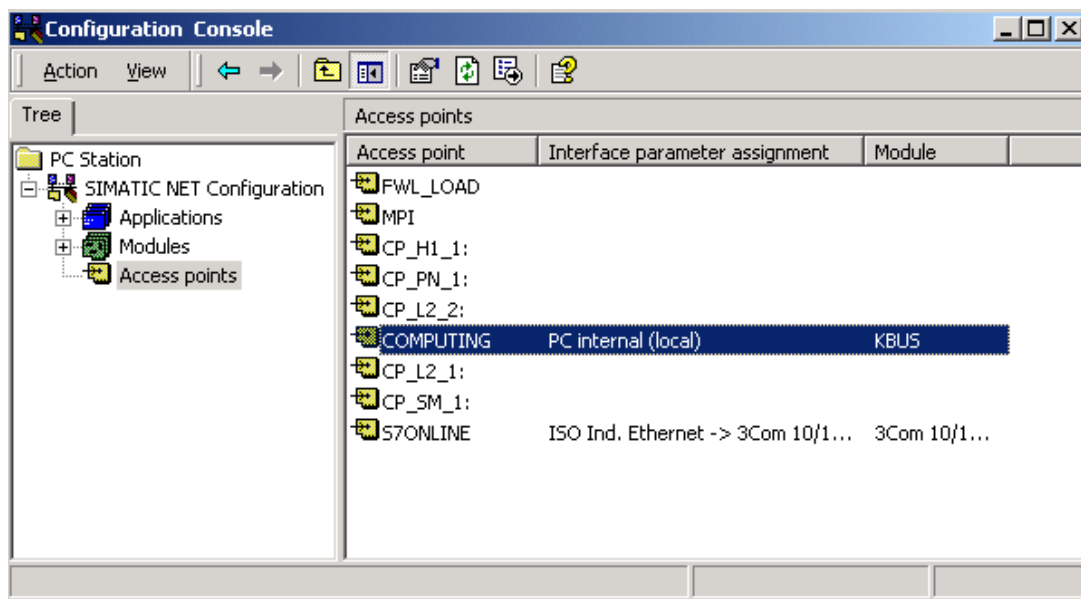


图 3.1 设置 WinAC Computing 的访问接口

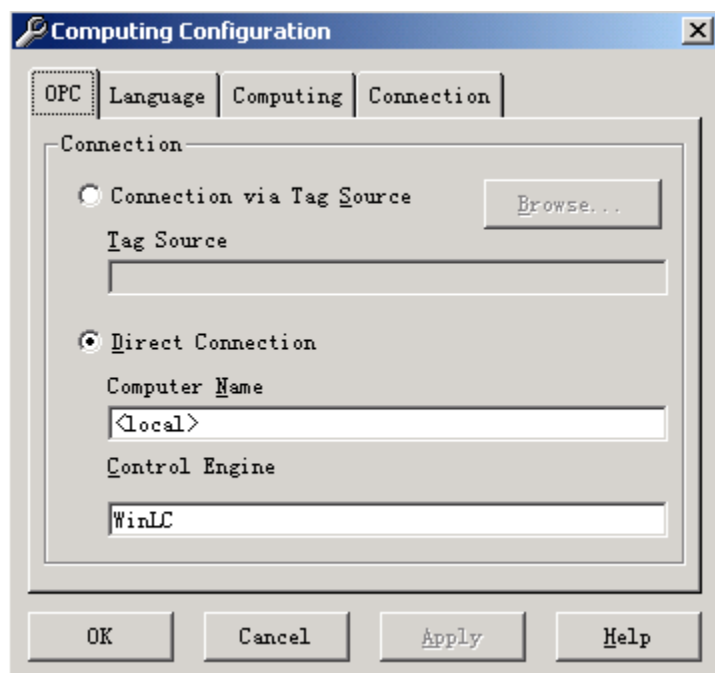


图 3.2 设置 OPC 的连接选项


图 3.2 的设置值表示：通过 Computing OPC 方式访问 WinAC 数据时，它访问的是本机上的 WinAC Basis 控制器中的数据。

“Computing Configuration” 程序的选项卡 “Computing” 在新的版本中已不使用。

3.2 组态标签文件

使用标签文件不但可以采用符号的方式访问 WinAC 控制器中的数据，还可以使用 OPC 客户端程序访问多个 WinAC 控制器中的数据。我们举一个例子，首先在 Step7 中定义了如图 3.3 所示的符号表。

图 3.3 Step7 的符号表

然后，在 WinAC 中建立组态标签。单击“PC Based Control”下的“Computing TagFile Configurator”，打开 WinAC 的符号表编辑器，程序自动新建一个标签文件。右键单击此标签文件视图窗口的左边，从弹出菜单中选择“Insert Program”，打开选择 step7 源程序的对话框，从中选择要进行 WinAC 符号标定的程序，单击按钮  将文件选入右边窗口。结果如图 3.4 的底层窗口所示，从中可以看到 Step7 符号表中的变量已包含其中。右键单击窗口左边的文件“COC-SLOT34\_WinLC”，从菜单中选择“Edit”，打开“control Engine Configuration”对话框，按照 3.1 节描述的方法设置计算机名和控制器，如图 3.4 中的顶层窗口所示的。

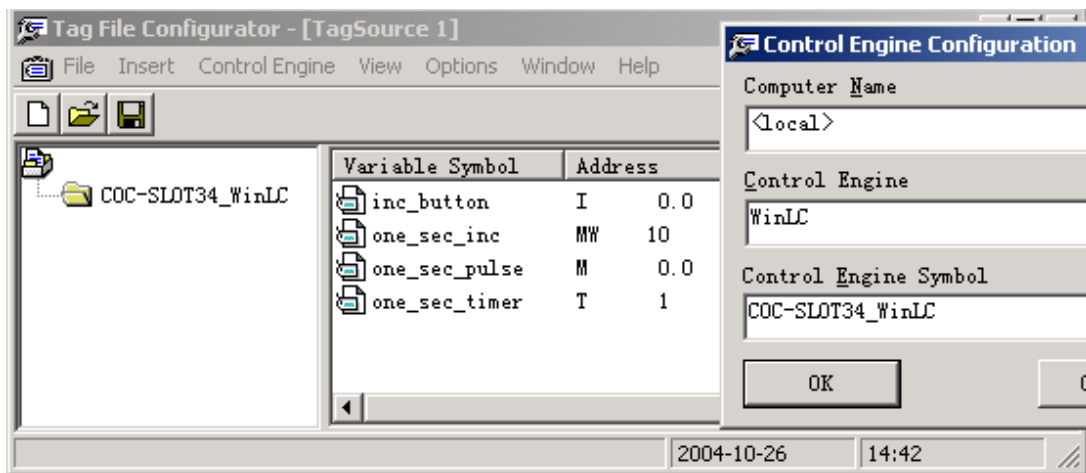


图 3.4 组态标签文件

如有需要可以添加新的文件(在同一标签文件下)并进行设置，完成后存盘退出。标签文件的扩展名为.tsd。标签文件创建后便可与图 3.2 中的标签源文件连接，之后可以通过符号的方式访问 WinAC 控制器中的数据。

为了说明本章后面部分的内容，我们举一个例子，其中系统组态同第二章。在 Step7 中编写一段代码，其完成的功能是：当 I0.0 为 1(true)时，每一秒钟将 mw10 的值加 1。当 mw10=10000 时，将 mw100 清零。

OB1 循环中的代码：

```

A      I      0.0
AN     M      0.0
L      S5T#1S
SD     T      1
A      T      1
=      M      0.0

AN     M      0.0
JC     m001
L      1
L      MW     10
+I
T      MW     10
L      10000
<I
JC     m001
L      0
T      MW     10
m001: NOP  0

```

### 3.3 Computing SoftContainer

Computing SoftContainer 为一 OLE 容器，在它的窗体上只能放置 WinAC Computing 提供的 ActiveX 控件。使用 Computing SoftContainer 可以生成简单实用的 HMI 界面窗口。

运行程序组“PC Based Control”下的程序 Computing SoftContainer。在 Computing SoftContainer 的工具栏上除了常规的按钮以外还有 WinAC Computing 所带的 ActiveX 控件图标，这些 ActiveX 控件还可用在可作为 OLE 容器的程序上，如 Visual Basic、WinCC 等。

- S7WCPanel: WinAC 操作面板控件。
- S7Data: WinAC 控制器数据访问控件。
- Dbuffer: 诊断缓冲区。

此外还有用于显示和操作数据的控件。下面的步骤描述如何在窗体上显示上节的 i0.0 和 mw10 两个数据。

- (1) 在窗体上添加一个 S7Data，一个 Button 和一个 Edit。
- (2) 双击窗体上的 S7Data1 对象，从弹出的窗口中选择“Engine”选项卡。
- (3) 选中通过标签文件进行连接，并选择上节保存的文件名作为标签文件。如图 3.5 所示。

(4) 选中 Connections 选项卡，分别设置对象 Button1 和 Edit1 的 value 属性值。这里我们将 Button1 的 value 设成“COC-SLOT34\_WinLC.inc\_button (即 I0.0)，将 Edit1 的 value 值设成“COC-SLOT34\_WinLC.one\_sec\_inc (即 MW10)。结果如图 3.6 所示。单击“OK”退出设置窗口。

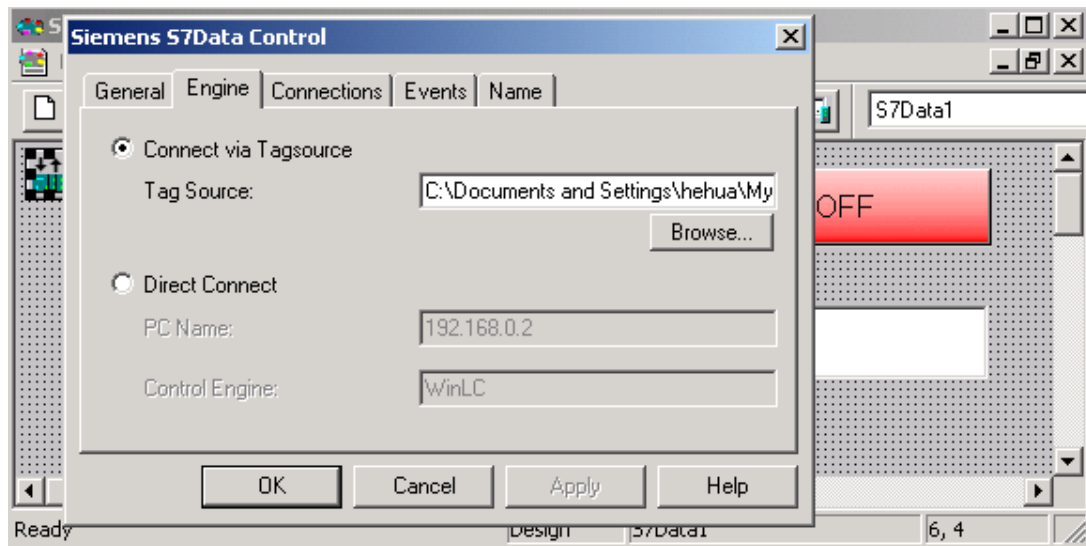


图 3.5 选择连接方式为标签连接

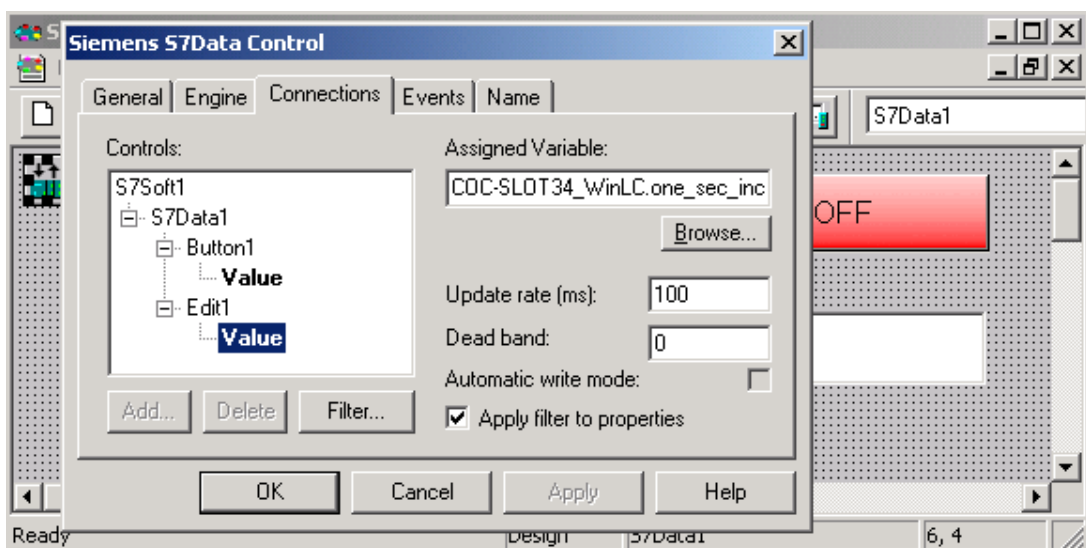


图 3.6 设置对象的 value 属性值

(5) 鼠标单击工具栏上的按钮 ，运行程序，结果如图 3.7 所示。

在 Edit1 对象中显示 MW10 的计数值，Button1 对象表示 I0.0 的状态，还可以用鼠标操作改变 I0.0 的状态，用键盘输入更改 MW10 的值。

其它控件的使用方法可参考 WinAC Computing 手册。

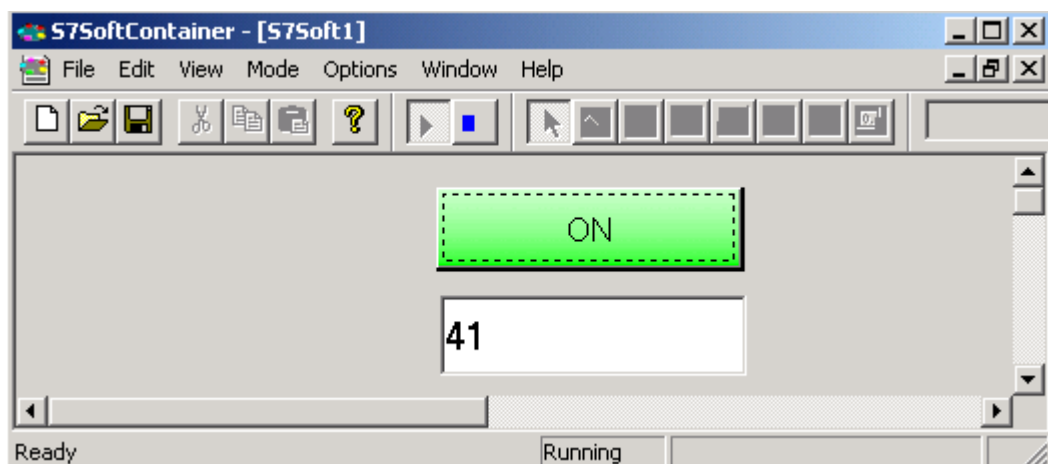


图 3.7 Computing SoftContainer 的运行效果

### 3.4 通过 OPC 方式访问 WinAC 中的数据

OPC 即基于过程控制的 OLE。从 2.0 版本开始，OLE 基于微软的 COM(组件对象模型)技术，所以 OPC 也基于 COM 技术。WinAC 支持 WinAC Computing OPC 和 SIMATIC NET OPC。编写同一功能不同 OPC 服务器的 OPC 客户端程序大体相同。SIMATIC NET OPC 提供 DA(数据访问)、A & E(报警和事件)服务；Computing OPC 提供 DA 服务。SIMATIC 的这两个 OPC 服务器



都支持定制接口和自动化接口，定制接口只支持用 C/C++ 编写的客户应用程序，自动化接口则支持更上层的应用，如 Visual Basic 以及 Excel 等支持 VBA 的所有应用程序。两种接口的数据访问服务都支持用同步和异步的方法读取数据。

在下面的三节中，我们将分别介绍如何使用 WinAC Computing 控件，WinAC Net OPC 控件，以及通过编写 WinAC Computing 和 WinAC Net OPC 客户端程序代码（不调用控件）访问 WinAC 的数据。

### 3.5 在 Visual Basic 中使用 WinAC Computing 的控件

WinAC Computing 的 ActiveX 有许多方法和属性，在高级语言中可很容易的使用这些控件。

本节描述如何在 Visual Basic 6.0 中使用 WinAC 的数据访问控件。

打开 Visual Basic 6.0，右键单击控件工具栏的空白处，从弹出菜单中选择“Component”，打开“Components”对话框，选择如图 3.8 所示的 Computing 控件。

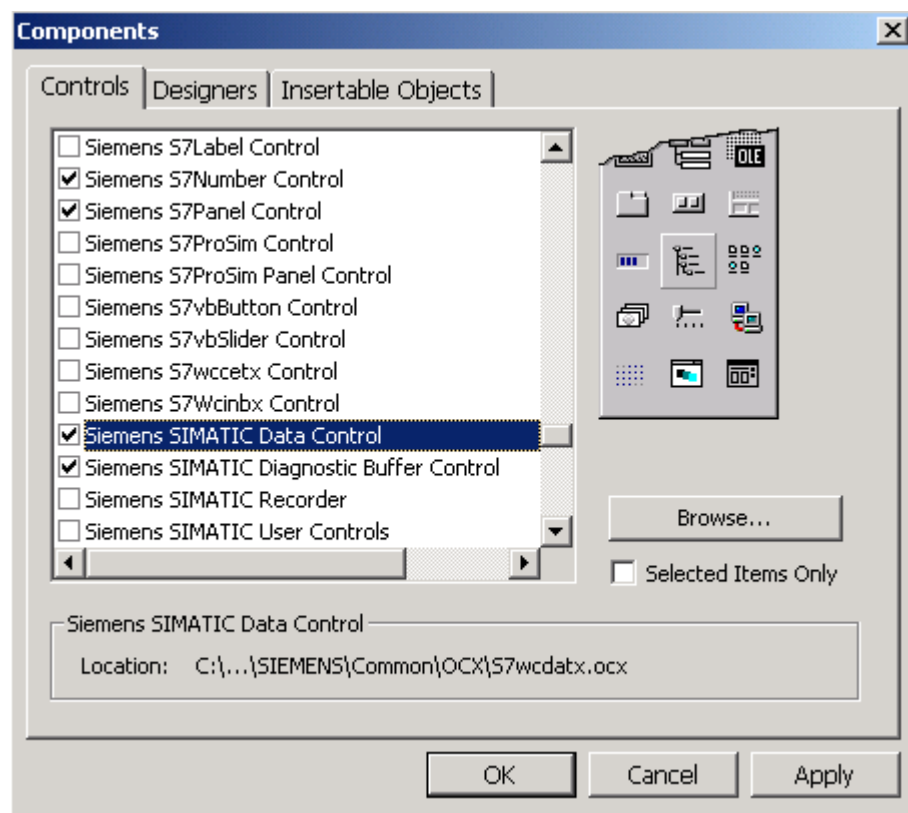


图 3.8 在 VB 中安装 ActiveX 控件

单击“OK”关闭对话框后在控件栏上将添加刚才所选的控件。在窗体上添加一个 S7Data

对象，以及一些按钮、文本框，和一个列表框来实现下列例子程序，设计窗体见图 3.9。在编写代码前，还必须设置 S7Data1 对象的属性，在图 3.5 所示的“Engine”选项卡上，我们这一次选择直接连接。PC Name 设为“<Local>”，Control Engine 设为“WinLC”。

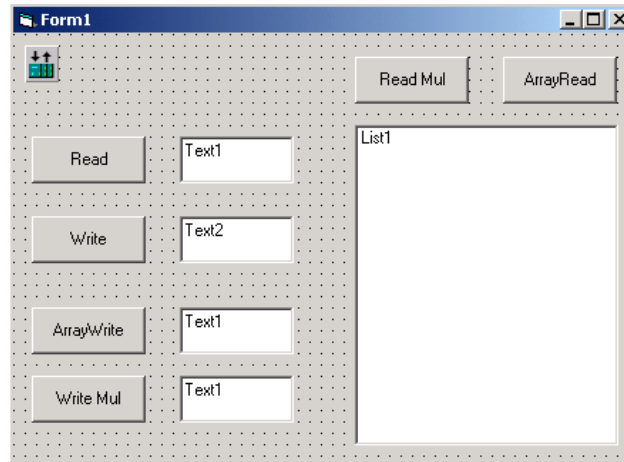


图 3.9 VB 的设计窗体

#### 例 1. 读变量的操作

```
Private Sub btnRead_Click()
    Dim r1, statel As Long
    ' 下面的语句为读变量，第一个参数为变量的地址，第二个参数存放结果
    r1 = S7Data1.ReadVariable("mw10:int", v1, statel, 0)
    ' txtRead 为 TextBox 的对象
    txtRead.Text = v1
End Sub
```

#### 例 2. 写变量的操作

```
Private Sub btnWrite_Click()
    Dim r1 As Long
    v1 = txtWrite.Text
    ' 第一参数为变量的地址，第二个参数存放结果
    r1 = S7Data1.WriteVariable("mw10:int", v1, 0)
End Sub
```

#### 例 3. 读取多个在 step7 中有连续存储位置的操作

```
Private Sub btnArrayRead_Click()
    Dim r1, statel As Long
    ' 读 3 个实数
    r1 = S7Data1.ReadVariable("md100:real[3]", v1, statel, 0)
    List1.Clear
    For i = 0 To 2
        List1.AddItem ("md" & 100 + i * 4 & "= " & v1(i))
    Next i
End Sub
```

#### 例 4. 读取多个在 step7 中有连续存储位置的操作

```
Private Sub btnArrayWrite_Click()
    Dim r1 As Long
    Dim v1(3) As Variant
    For i = 0 To 2
```

```

        v1(i) = txtArrayWrite.Text
    Next i
    ' 写 3 个实数
    r1 = S7Data1.WriteVariable("md100:real[3]", v1, 0)
End Sub

```

例 5. 一次读多个不连续的变量

```

Private Sub btnReadMul_Click()
    Dim name1(3) As String
    name1(0) = "m0.0:bool"
    name1(1) = "mw10:int"
    name1(2) = "md100:real"
    ' 一次读多个变量函数
    r1 = S7Data1.ReadMultiVariables(name1, v1, statel)
    List1.Clear
    For i = 0 To 2
        List1.AddItem (name1(i) & "=" & v1(i))
    Next I

```

例 6. 一次写多个不连续的变量

```

Private Sub btnWriteMul_Click()
    Dim r1 As Long
    Dim v1(3) As Variant
    Dim name1(3) As String
    name1(0) = "i0.0:bool"
    name1(1) = "mw10:int"
    name1(2) = "md100:real"
    v1(0) = False
    v1(1) = 100
    v1(2) = 9.999
    r1 = S7Data1.WriteMultiVariables(name1, v1, statel)
End Sub

```

例 7. 变量有变化时作出反映

步骤 1. 右键单击窗体上的 S7Data1 对象，从弹出菜单中选择“Properties”。打开 S7Data1 对象的属性设置对话框，选择“Event”选项卡上的“S7Data1”，单击按钮“Add”，添加一变量的别名，回到“Event”选项卡，选择刚创建的变量别名，在窗口右边的“Assigned Variable”编辑框中输入要监控变量的地址，如“mw12”，如有需要可修改刷新时间和死区。按同样的方法添加另一监控的变量，结果如图 3.10 所示。单击“OK”关闭对话框。

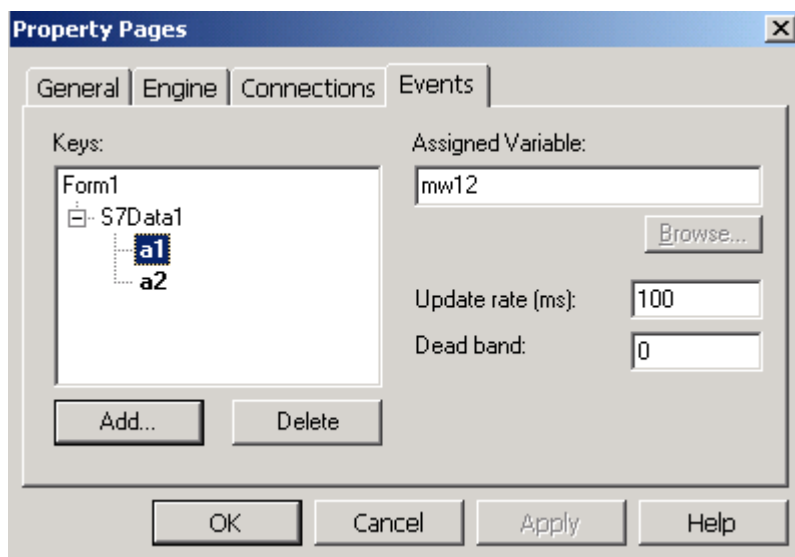


图 3.10 监控变量值变化事件

步骤 2. 在代码窗口的对象组合框中选择“S7Data1”，事件组合框中选择“ValueChanged”，添加事件处理程序代码如下。

```
Private Sub S7Data1_ValueChanged(ByVal Property As String, ByVal VarName As String,
                                ByVal Value As Variant, ByVal Quality As Integer)
    If VarName = "mw12" Then
        MsgBox "mw12 changed"
    ElseIf VarName = "mw14" Then
        MsgBox "mw14 changed"
    End If
End Sub
```

### 3.6 SIMATIC NET OPC 控件

可以使用相同的接口来访问 S7-300/400 和 WinAC 控制器，即 SIMATIC NET OPC。通过 NET OPC 来访问 WinAC 的数据不需要另外的授权和硬件。NET OPC 除了提供定制和自动化接口来访问控制器的数据外，还提供一个 ActiveX 控件方便使用。

SIMATIC NET 提供一个工具 OPC Scout，它是一个简单的 SIMATIC OPC 客户端工具软件，可浏览存在的 SIMATIC OPC 服务器名称、添加 OPC 组、添加项、监控项值等功能。

在图 3.8 所示的添加控件窗口中添加名称为“SIMATIC NET OPC DATA”控件。

在窗体上添加一个 DatCon 对象 DatCon1 以及若干按钮、文本编辑框对象、列表框。打开 DatCon1 的属性设置窗口，选择“OPC Server”选项卡，Server 编辑框中输入 OPC 服务器名称“OPC.SimaticNet”（“OPC.SimaticNet”是 SIMATIC NET 作为 OPC SERVER 的服务器主件的名称），在 Node 编辑框中输入计算名“COC-SLOT34”，如图 3.11 所示。

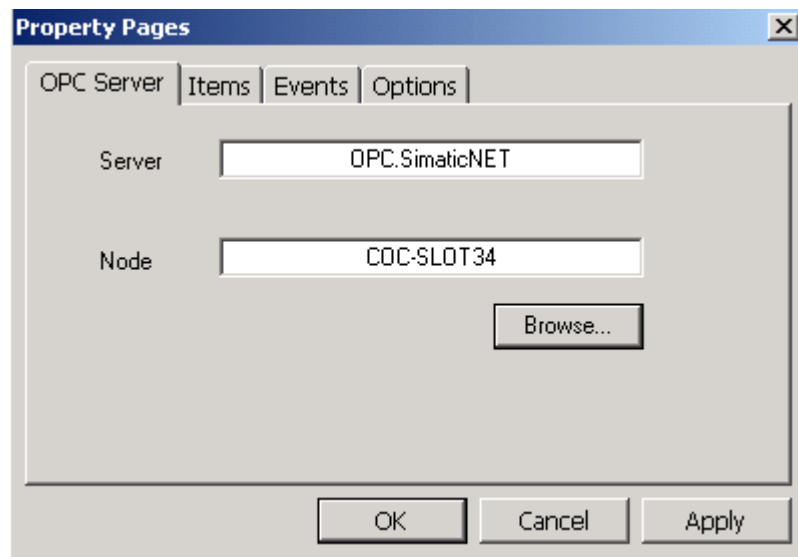


图 3.11 设置 NET OPC DATA 控件的服务器和计算机名

#### 例 1. 读数据

```
Private Sub btnRead_Click()
    Dim r1, q1 As Long
    Dim strname(2) As String
    Dim time1 As Date
    '读单个变量
    r1 = DatCon1.ReadVariable("S7:[S7 connection_1]MINT10,1", v1, q1, time1)
    Text1.Text = v1
    '读多个变量
    strname(0) = "S7:[S7 connection_1]MINT10,3"
    strname(1) = "S7:[S7 connection_1]db1,REAL0,3"
    r1 = DatCon1.ReadMultiVariables(strname, v2, e2, q2, time1)
    for i=0 to 2
        for j=0 to 2
            list1.AddItem(v2(i)(j))
        Next j
    Next i
End Sub
```

读取变量的格式为 “S7:[连接名]数据区 数据类型 起始地址, 数量”

连接名在网络组态时已确定, 在图 2.18 中的连接名为 “S7 connection\_1”。

“S7:[S7 connection\_1]MINT10,3” 表示读取 mw10 开始的 3 个整数值。

“S7:[S7 connection\_1]db1,REAL0,3” 表示读取 db1.dbd0 开始的 3 个浮点数值。

#### 例 2. 写数据

```
Private Sub Command2_Click()
    Dim r1 As Long
    Dim e2, q2 As Variant
```

```

Dim strname(2) As String
Dim v2(2) As Variant
strname(0) = "S7:[S7 connection_1]MINT10,1"
strname(1) = "S7:[S7 connection_1]DB1,REAL0,1"
v2(0) = 100
v2(1) = 2.2
r1 = DatCon1.WriteMultiVariables(strname, v2, e2)
End Sub

```

### 3.7 编写 WinAC Computing OPC 和 SIMATIC NET OPC 客户端程序

要编写 OPC 客户端程序，先引用 OPC 的 DLL 文件。单击 Visual Basic 6.0 菜单“Project”->“Reference”，打开引用对话框，将“Siemens OPC DAAutomation 2.0”选中。如图 3.12 所示。如果这个组件没有在列表框出现，可能的原因是组件没有注册，可以使用 Windows 附带的程序 regsvr32.exe 进行注册，SIMENS OPC 自动化组件包含在 SopcDAAuto.DLL 库中。

编写这两种 OPC 的客户端程序只有两项不同，一个是服务器名称，另一个是项名称(可以认为是变量名)。

Computing OPC 的服务器名称为“OPCServer.WinAC”。

Simatic NET OPC 的服务器名称为“OPC.SimaticNET”。

Simatic NET OPC 的项名称和在 3.6 节中描述的类似，形如：

```
Set ItemObj1 = GroupObj.OPCItems.AddItem("S7:[S7 connection_1]MINT10", 1)
```

```
Set ItemObj2 = GroupObj.OPCItems.AddItem("S7:[S7 connection_1]DB1,REAL0", 2)
```

Computing OPC 的项名称在 3.5 节有介绍，形如：

```
Set ItemObj1 = GroupObj.OPCItems.AddItem("MW10:INT", 1)
```

```
Set ItemObj2 = GroupObj.OPCItems.AddItem("DB1.DBDO:REAL", 2)
```

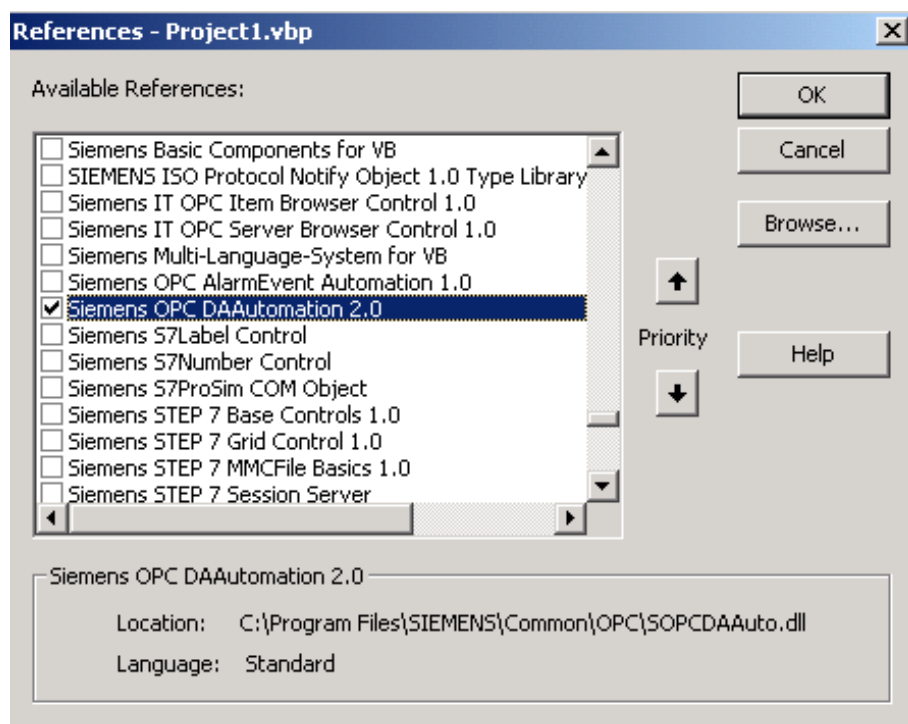


图 3.12 引用 SIEMENS OPC 数据访问自动化组件

在安装 Simatic NET 的目录下(如 c:\programfiles\siemens\simatic.net\opc2\samples)下有何编写 OPC 客户端程序的例子程序。

## 第四章 WinCC 访问 WinAC

### 4.1 WinCC 与 WinAC 安装在相同的 PC 机上

WinCC 提供了专门的通道单元来访问 WinAC 控制器中的数据。如果 WinAC 与 WinCC 安装在同一台 PC 机上，使用“SIMATIC S7 PROTOCOL SUITE”下的通道单元“Soft PLC”和“Slot PLC”，WinAC Basis 和 WinAC RTX 使用 Soft PLC 通道，WinAC Slot 412/416 使用 Slot PLC 通道。

在 WinCC 建立与 WinAC 连接时，应该注意其属性设置，由于 WinCC 与 WinAC 安装在同一 PC 机上，它们是通过内部通道进行通讯的，因此设置连接参数时与 WinAC 的站地址无关。在通道单元 Soft PLC 和 Slot PLC 连接属性中的站地址和插槽号应相同，且都为 WinAC 在 PC 站中的虚拟插槽号(即 Station Configuration Editor 中的索引号)。默认时，WinAC Basis/RTX 插在虚拟插槽 2 中，WinAC Slot 插在虚拟插槽 3 中。WinCC 在默认情况下，也已正确设置了站地址和插槽号。

如图 4.1 即为 WinCC 连接 WinAC Basis/RTX 的设置，站地址与插槽号都设置成 2，即 WinAC 在 PC 站中的虚拟插槽号 2。

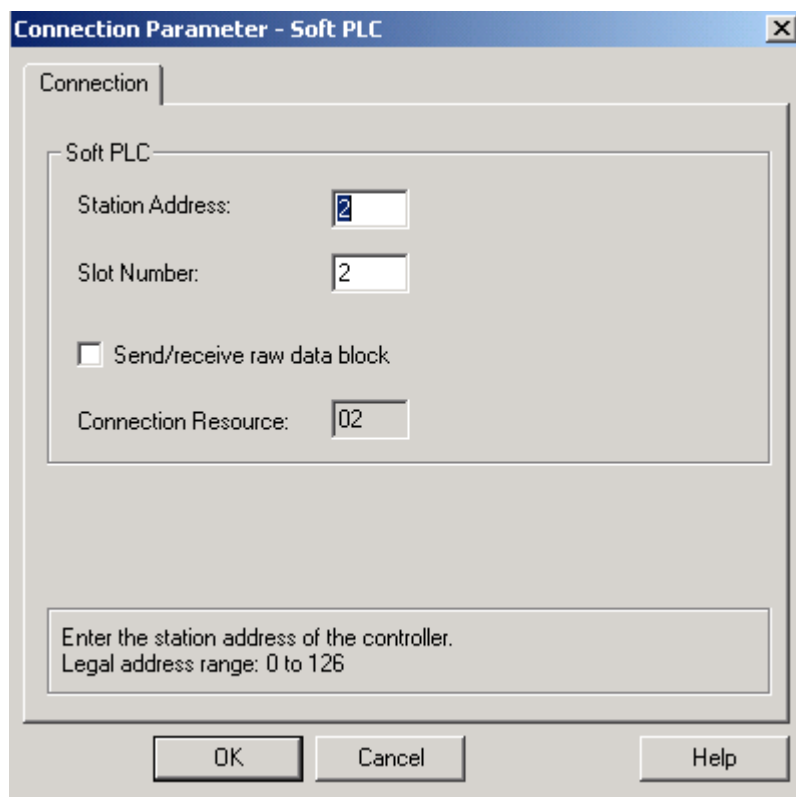


图 4.1 Soft PLC 通道连接属性参数



图 4.2 为 WinCC 连接 WinAC Slot 时的参数属性，站地址与插槽号都为 3，即 WinAC Slot 在 PC 站中的虚拟插槽号。

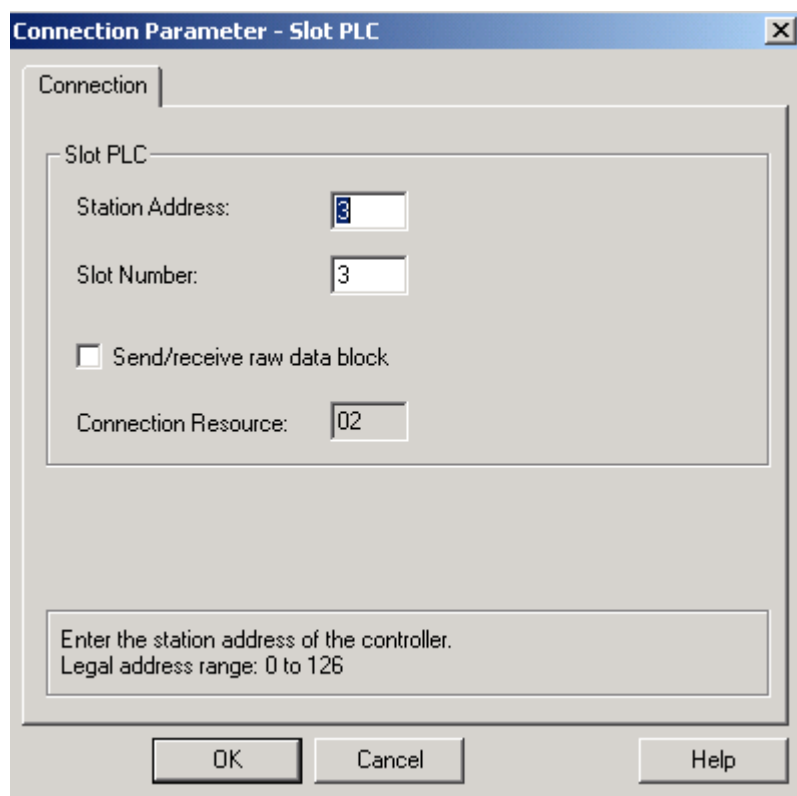


图 4.2 Slot PLC 通道连接属性参数

## 4.2 WinCC 与 WinAC 安装在不同的 PC 机上

WinCC 与 WinAC 安装在不同的 PC 机上时，根据这两台 PC 机的不同连接选择不同的通道单元，这时可把 WinAC 当作一台 PLC 站来处理。连接属性中的插槽号为 PC 站的虚拟插槽号（即 WinAC Basis/RTX 为 2，WinAC Slot 为 3）。站地址（对于 PROFIBUS 和 MPI 而言）为 WinAC 站上与 WinCC 相连的 CP 卡的站地址。以太网地址或 IP 地址（对于以太网连接而言）为 WinAC 站上与 WinCC 相连的网卡的 MAC 地址或 IP 地址。

WinAC 与第三方 HMI 软件的通讯方式可用那些适用于 SIMATIC S7 控制器的通道，也可使用 OPC 通道。

## 第五章 WinAC ODK

WinAC ODK 4.1 是一个专用于 WinAC 的开发工具，使用这个工具可以将 C/C++ 程序经编译连接后嵌入到 WinAC Basis/WinAC RTX 的 PLC 程序中，还可以编写应用程序与 WinAC 进行快速的数据交换、监控 WinAC 状态以及控制 WinAC 操作面板。WinAC ODK 4.1 包括三个组件：

- 定制代码扩展 (CCX)
- 共享内存交换 (SMX)
- 控制器管理接口 (CMI)

### 5.1 定制代码扩展 (CCX: Custom Code Extension)

WinAC ODK 提供了一个向导，使用该向导可生成 Visual C++ 6.0 或 Visual .NET 的工程，在 Visual Studio 集成开发环境下编译连接后，生成动态链接库。在 WinAC 的 Step7 程序中利用 2 个系统功能块 SFB65001 和 SFB65002 便可调用动态链接库中的函数。该功能可用于软件型 WinAC (WinAC Basis 和 WinAC RTX)，但不能用于插槽型 WinAC。下面将以例子来说明如何生成和调用。

将 Step7 中的数据 mw20 和 mw22 作为被加数和加数传给动态链接库，由动态链接库中的函数将这两个数相加后送回 Step7 的 mw24 中。

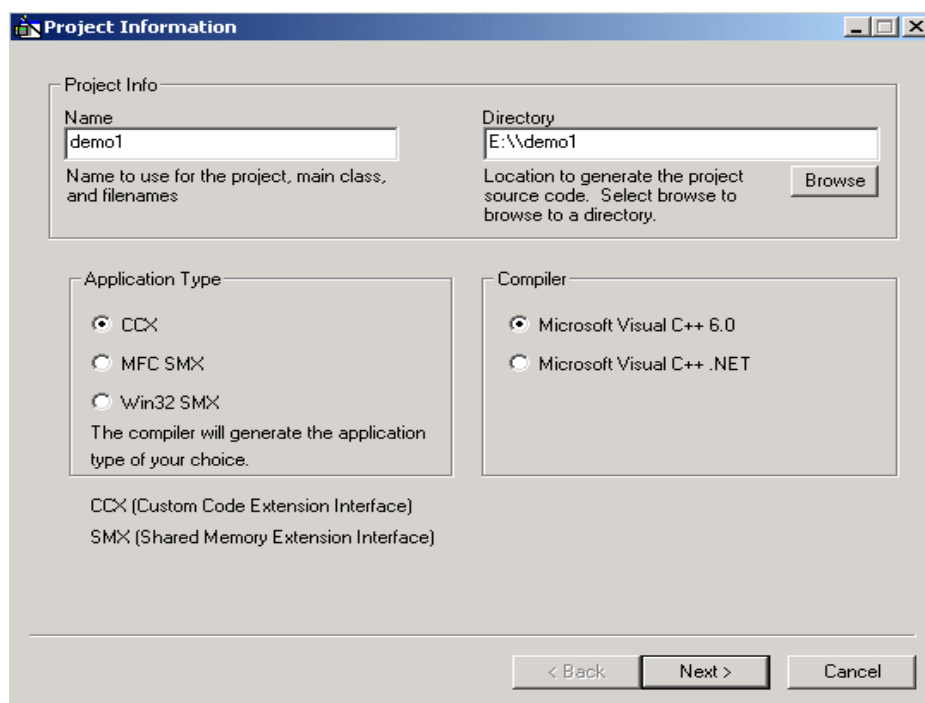


图 5.1 应用程序类型和开发工具类型

1. 运行“PC Based Control”程序组下的“WinAC ODK Appwizard”，打开 ODK 动态链

接库生成向导，单击“Next”，出现如图 5.1 所示的项目信息窗口，在“Name”编辑框中输入项目的工程名称，“Directory”框中选择工程存放目录，应用程序类型选择“CCX”，编译器类型选择“Visual C++ 6.0”，如图 5.1 所作的选择，单击“Next”进入一步。

2. 在“CCX Subcommands”窗口中，可创建函数给 Step7 调用，默认时只有一个函数，根据实际需求可创建多个，创建时可以给函数名称取一个有意义的名字。每一个函数都有函数索引号，Step7 调用这些函数时必须输入函数的索引号。在图 5.2 上，我们建立名为“AddInt”的函数，它的函数索引号为 0。

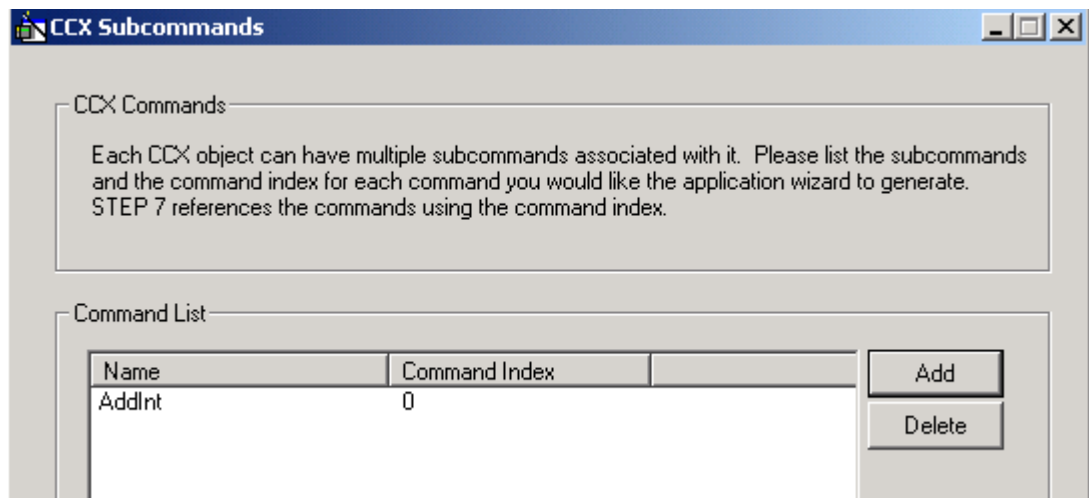


图 5.2 添加函数

3. 单击“Next”进入“Asynchronous Processing”窗口，在此窗口中指明动态链接库是否包含异步处理器，异步处理器是指在后台运行的一个 Windows 线程，这个线程可以处理多个事件，多个事件在队列中等待异步处理器的处理。当有一个事件产生时，如果异步处理器正在处理其它的事件，则当前产生的事件进入到事件队列等待处理。某一个事件处理完，异步处理器就将该事件从队列中删除，并从队列中取出下一个事件继续处理。如果程序中有包含异步处理器，可指定异步处理器线程的优先级，以及创建多少个异步事件。比较耗时的非经常性操作，如磁盘操作、数据库查询等放在异步处理器中执行。本例不使用异步处理器。直接单击“Next”进入下一步。

4. 在“Asynchronous Monitoring”窗口中，可指明动态链接库程序中是否要包含异步监控，可以同时生成多个异步监控，每个异步监控都是在后台运行的一个 Windows 线程，这些线程都是在后台循环运行，它不影响 OB1 块（或其它组织块）的执行时间。比较耗时的且要求循环执行的操作放在异步监控中执行，如串口通讯、Socket 通讯等。本例不使用异步处理器。直接单击“Next”进入下一个窗口。

5. “Vendor Information”窗口中，我们不输入任何内容单击“Next”后，在下一窗口中单击“Finish”完成向导的各个步骤。

6. 在 Visual C++ 6.0 中打开刚才创建的工程，找到在第二步中添加的“AddInt”函数，如果工程名为 demo1，“AddInt”在 demo1func.cpp 文件中。在此函数体中输入如图 5.3 所示的代码。

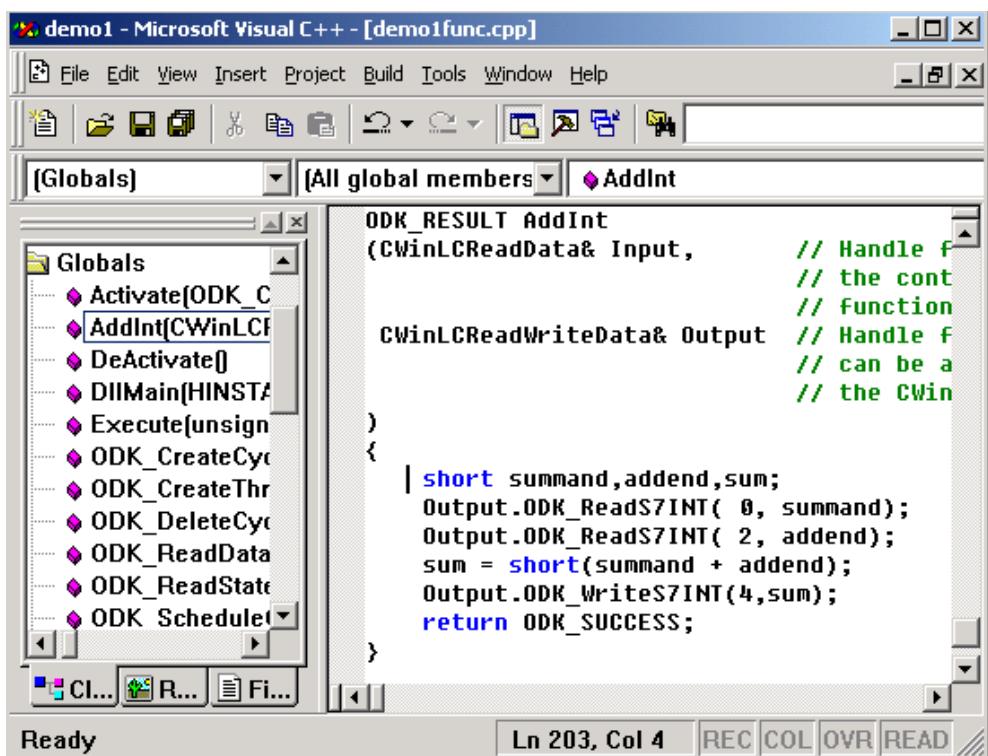


图 5.3 在 Visual C++ 中编写的相加函数

AddInt 函数中的参数 Input, Output 是与 PLC 程序进行数据交换时的对象变量, Input 参数数据只能从 Step7 传给 C++, 而 Output 参数数据既可以从 Step7 传给 C++, 也可以从 C++ 传给 Step7。图中代码的含义是: 先定义三个短整数的变量, 被加数从输出缓冲区偏移地址为 0 处读取, 加数从输出缓冲区偏移地址为 2 处读取, 输入输出缓冲区的偏移地址以字节为单位。将两数相加后的结果写入到输出缓冲区偏移地址为 4 处。

单击 Visual C++ 开发运行环境菜单 “Build” -> “Set Active Configuration”, 打开 “Set Active Project Configuration” 对话框, 如果是 WinAC RTX 且装有 RTX SDK, 则可选择 “Win32 RTSS” 的调试版或发布版, 否则为 “Win32” 的调试版或发布版。这里我们选择 “Win32 Release”。

点击菜单上的 “Build” -> “Build demo11.dll”, 编译并连接生成 demo1.dll 动态库。

7. 用 Step7 打开一个已组态好的 WinAC 工程。添加组织块 OB100, 在 OB100 中调用 SFB65001。SFB65001 和 SFB65002 为 WinAC ODK 提供的系统功能块, 功能相当于 WinAC 调用动态链接库的接口块。SFB65001 为打开动态库函数, SFB65002 为调用动态库中的函数。只有调用这两个系统功能块, Step7 语句才能调用动态库中的函数。

如果在梯形图编辑器的库中没有出现 “WinAC ODK library” (即这两个系统功能块), 可单击 “SIMATIC Manager” 的菜单项 “File” -> “Retrieve”, 打开 “Retrieving Select an Archive” 对话框, 在对话框中打开目标文件 “ODKlib.zip”, 此文件位于 WinAC ODK 的安装目录下。按照提示完成操作后, 梯形图编辑器中的库下便增加了 WinAC ODK 库。ODK 库下包含 SFB65001 和 SFB65002。

在 OB100 编写程序如图 5.4 所示：

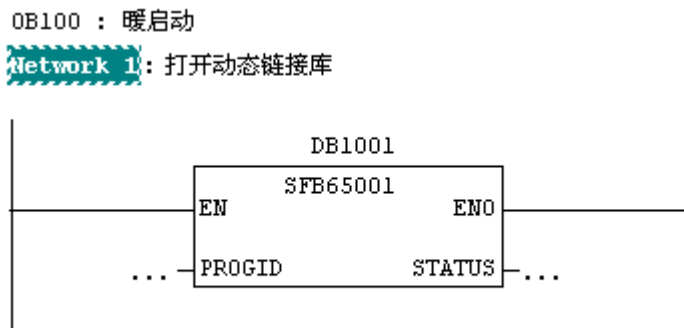


图 5.4 打开动态链接库

8. 编辑 SFB65001 的背景数据块 DB1001，内容如图 5.5 所示

	Address	Decl	Name	Type	Initial value	Actual value
1	0.0	in	PROGID	STRING [ 254 ]	"	*DLL:e:\demo1\release\demo1.dll
2	256.0	out	STATUS	WORD	W#16#0	W#16#0

图 5.5 SFB65001 的调用参数

数据块起始地址 0 的数值为一字符串，此字符串指明动态链接库的路径和文件名。如果动态链接库的扩展名为 dll，则路径名前加“\*DLL:”，如果动态链接库的扩展名为 RTDLL，则路径名前加“\*RTSS:”。第二个参数是调用动态库是否成功的返回标志。

9. 编辑组织块 OB1，增加如图 5.6 所示的一条语句，调用 SFB65002。

Command 参数为在第 3 步建立函数时的函数索引号，在本例中为 0。

Input 和 Output 为与 C 代码进行数据交换的输入/输出缓冲区，这里的 Input 即对应图 5.3 中的 Input，Output 对应图 5.3 中的 Output。Input 和 Output 为 Step7 的“ANY”数据类型，指明数据区的起始地址和长度。在图 5.6 中 Output 指明为从 m20.0 开始，占用十个字节长度，即输出缓冲区为 mw20–mw28，输入缓冲区没有指定。

10. 将程序编译存盘并下载到 WinAC 控制器。新建一个变量表进行测试，可以看到 mw24 为 mw20 与 mw22 之和。

除了在 Step7 中调用 C 编译后的代码外，WinAC ODK 中还提供了其它的功能和函数，如在 C++中也可以设置条件使 PLC 程序中的组织块被调用，以及当 WinAC 停机或启动时应调用的 C 代码、异步处理器和异步监控等。可参考 WinAC ODK 安装目录下的例子程序。

## Network 2: 调用函数

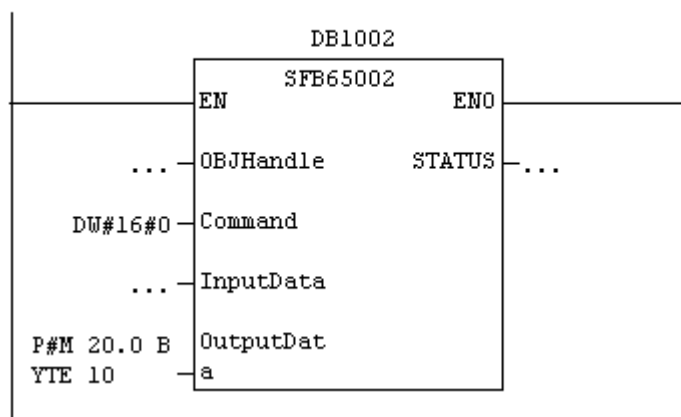


图 5.6 调用动态库中的函数

## 5.2 共享内存交换(SMX:Shared Memory Exchange)

SMX 可用于软件型 WinAC 和插槽式 WinAC。SMX 其实是一组动态链接库，Win32 应用程序可调用库中的函数来与 WinAC 中的 PI、PQ 区进行数据通讯。软件型的 WinAC 使用共享内存的方式，插槽式的 WinAC 采用双端口存储器(dual-port RAM)的方式来交换数据，即通讯要么直接在内存中完成，要么通过 PC 的系统总线来完成，因此数据交换速度极快。但无论使用两种方式中的哪一种，在软件实现上都没有区别。

输入输出区各占用 4K 字节。WinAC Basis、WinAC RTX 和 WinAC Slot 416 的输入区为 PIW16384-PIW20478，输出区为 PQW16384-PQW20478，WinAC Slot 412 的输入区为 PIW4096-PIW8190，输出区为 PQW4096-PQW8190。

可以使用向导来生成 SMX 程序，在如图 5.1 中选择应用程序类型为“MFC SMX 或 Win32 SMX”，则生成的程序是 SMX 程序。用向导生成的 MFC 程序不一定能满足实际要求，如果这种情况发生，将不能使用 SMX 向导。下面以例子来说明不采用向导，如何使用 SMX。

1. 使用 Visual C++向导(非 ODK 向导)生成一个对话框程序。
2. 打开 Visual C++菜单下的 Project->Settings。在打开的项目设置对话框中，选择 C/C++选项卡，在 Category 列表框中选择“Preprocessor”，在 Addition Include directories 编辑框中输入“c:\program files\siemens\winac\odk\include”（根据本机 WinAC ODK 实际安装目录来设置，下同）。
3. 将“c:\program files\siemens\winac\odk\lib 目录下的库文件 s7smxx.lib 拷贝到当前项目目录下。
4. 在对话框头文件中加上“include <s7smxx.h>”语句。
5. 在对话框资源中添加两个按钮：一个读数据按钮，一个写数据按钮。两个编辑框：一个接收读取的数，一个设置要写的数，用 Visual C++的类向导创建这两个编辑框关联的无符号整型变量，m\_readvalue 和 m\_writevalue。

6. 在对话框类中增加两个私有变量

```
long *hSmx;  
S7SMX_ERROR smxError;
```

7. 在对话框窗口创建后的初始化对话框内的控件函数(OnInitDialog)的最后加上如下语句:

```
smxError=S7SMX_Open(S7SMX_OPEN_MODE_WRITE, "WinLC", (void **)&hSmx);
```

这条语句的含义为打开 SMX 连接, 其中的第二个参数为 WinAC 控制器类型。

8. 编写读数据和写数据的按钮数据如下

```
void CDemo2Dlg::OnRead()  
{  
    ReadS7WORD(hSmx, 0, (unsigned short *)&m_readvalue);  
    UpdateData(false);  
}  
//第二个参数为偏移地址, 相对于 PQB 16384 时的偏移地址, 为 0 即指 PQW16384。  
//即读取 WinAC PQW 16384 的值, 并存放在 m_readvalue 中。  
void CDemo2Dlg::OnWrite()  
{  
    UpdateData();  
    WriteS7WORD(hSmx, 6, (unsigned short)m_writevalue);  
}  
//将 m_writevalue 值写入到 WinAC PIW 16390(16384 + 偏移量 6)中。
```

9. 编译连接生成可执行程序

10. 在 Step7 的 OB1 增加如下语句并将程序下载。

```
L    MW    30  
T    PQW16384
```

```
L    PIW16390  
T    MW    32
```

11. 打开变量表, 监控 mw30 和 mw32。运行第 9 步生成的程序, 点击按钮测试结果。

上面简要介绍了如何使用 SMX, WinAC ODK 提供了详细的文档和例子程序来说明如何使用 SMX。

### 5.3 控制器管理接口(CMI Controller Management Interface)

控制器管理接口可使用户程序集成 WinAC 操作面板的全部功能, 比如在客户程序上可以完成如下功能:

- 显示 WinAC 控制器运行状态
- 归档和恢复 PLC 程序
- 本地或远程启停 WinAC
- 更改 WinAC 操作面板的运行参数

如何使用 CMI 可阅读参考手册和例子程序。

## 第六章 WinAC MP

### 6.1 WinAC MP—SIMATIC 嵌入式控制

WinAC MP 是软件型 PLC，应用于多功能面板 MP370 12” 和 MP370 15” 中，运行在 Windows CE 操作系统上。因为 WinAC MP 的运行平台 SIMATIC MP370 是无风扇、无硬盘的高可靠硬件平台，因此 WinAC MP 适用于恶劣环境下现场直接安装的场合，以及用于联网现场的单元级控制应用。WinAC MP、ProTool 和 MP370 的完美组合增加了任何运行环境下的操作可靠性。Windows CE 3.0 操作系统保证 WinAC MP 的实时操作性。MP370 的高运算性能允许 PLC 用户程序的高处理速率，同时允许可视化画面高的图像刷新率。

### 6.2 将 WinAC MP 移植到多功能面板中

#### 一. 概述

为了将 WinAC MP 从组态计算机移植到多功能面板中，需要在组态计算机和多功能面板中执行以下操作：

- 在组态计算机和多功能面板中设置通讯接口参数
- 连接多功能面板和组态计算机，并设置多功能面板到“Transfer”模式（参见多功能面板的硬件手册）
- 使用 ProSave 从组态计算机下载 WinAC MP 运行文件到多功能面板
- 使用 ProSave 从组态计算机下载 WinAC MP 授权到多功能面板
- 使用 ProTool CS 从组态计算机下载 ProTool 项目到多功能面板

当你完成这些任务之后，你可以组态硬件设备，在 STEP7 中编写程序逻辑，将 STEP7 的项目下载到多功能面板 WinAC MP 的控制引擎中。

#### 二. 设置通讯参数

组态计算机和多功能面板之间的通讯支持四种接口：以太网、串口、MPI、USB。为了设备之间的成功连接，你必须在组态计算机适当的软件中（例如：ProSave 或 ProTool）设置参数，并在多功能面板中设置控制面板应用。举例说明，当你下载 WinAC MP 运行文件或 WinAC MP 授权时，你使用 ProSave，所以你必须多功能面板和 ProSave 中组态通讯参数。同样的，当你下载 ProTool 项目时，你必须确保通讯参数与 ProTool 和多功能面板中的参数匹配。（对于 ProSave 和 ProTool，没有必要使用相同的通讯协议。）

对于描述的这四种通讯选择，你必须激活多功能面板和组态计算机之间的直接连接。步骤如下：

- 在多功能面板中的控制面板，打开“通讯”应用并选择“PC Connection”页
- 确认单选框“Enable Direct Connection To the Desktop Computer”被选中



### 三. MPI 通讯参数

在四种通讯方式中，我们只介绍 MPI 通讯方法，其余方法参见使用手册。

为了设置 MPI 通讯，你必须在组态计算机中安装卡（对于台式机选用 CP5611，对于笔记本电脑选用 CP5511）和驱动程序。在组态计算机中通讯卡的 9 针口和多功能面板的 DP/MP/PPI 端口之间连接 MPI/PROFIBUS 电缆，并配置如下参数。

在你的组态计算机中	在你的多功能面板中
<b>用 ProSave 连接</b> 1. 在 ProSave 的 General 页中为你的多功能面板选择正确的设备类型（MP370 Touch 或 MP370 Key） 2. 在 Connection 列表框中选择 MPI/PROFIBUS DP 3. 在连接参数中，为已经分配地址的多功能面板配置 OP 地址（OP 地址和 S7-TRANSFER 中的地址相同）	<b>编辑 S7-TRANSFER 设置</b> 1. 在控制面板中，打开 S7-TRANSFER 设置 2. 在列表框中选择 MPI 3. 单击属性按钮 4. 编辑地址，为多功能面板分配合适的站地址（地址是被组态计算机使用的 OP 地址） <b>编辑 TRANSFER 设置</b> 1. 在控制面板中，打开 TRANSFER 应用 2. 在通道 2，从列表框中选择 MPI 3. 在单选框中选择 Enable Channel
<b>用 ProTool 连接</b> 1. 打开 Download Preferences（File->Download->Preferences）设置下载对话框 2. 在列表框中选择 MPI/PROFIBUS DP 3. 为已经分配地址的多功能面板配置 OP 地址（OP 地址和 S7-TRANSFER 中的地址相同） 4. （可选）当你需要从多功能面板上载 ProTool 项目到组态计算机中，选择允许上载框	<b>保存设置</b> 1. 在控制面板中，打开 OP 应用 2. 单击 Save Registry 按钮以确保保存你的组态。否则，重新上电设置将被复位到缺省值

## 6.3 下载 WinAC MP 运行文件

你在下载带有 WinAC MP 组件的 ProTool 项目到多功能面板之前，你需要下载 WinAC MP 运行文件。WinAC MP 运行文件包括 WinAC MP 控制引擎。为了将 WinAC MP 运行文件从组态计算机下载到多功能面板，你需要使用 ProSave。在 Windows 启动菜单中，单击 SIMATIC->ProSave V6.0->ProSave 打开 ProSave。然后执行以下步骤：

- 单击 General 页，从设备类型列表框中选择正确的设备，参见图 6.1 操作

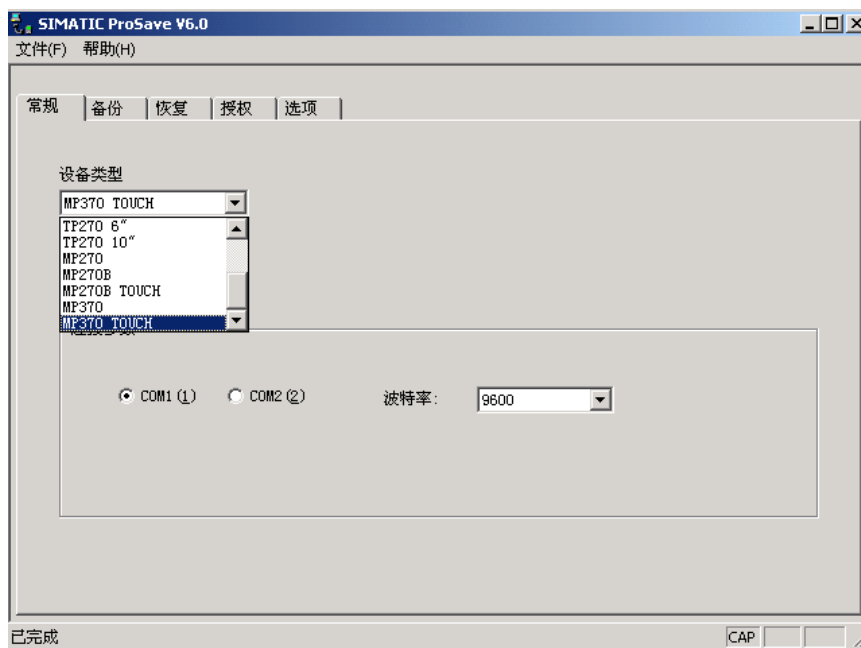


图 6.1 SIMATIC PROSAVE GENERAL 设置

- 在 Connection 列表框中，选择你在组态计算机和多功能面板之间使用的连接类型（参见手册中通讯部分的介绍）
- 填写 Connection 参数
- 单击 Option 页，单击 Device Status 按钮确认连接
- 在左边的可选型中选择 WinAC MP，然后单击>>按钮将 WinAC MP 添加到右边的栏中。这个操作将引起从组态计算机下载 WinAC MP 运行文件到多功能面板。参见图 6.2

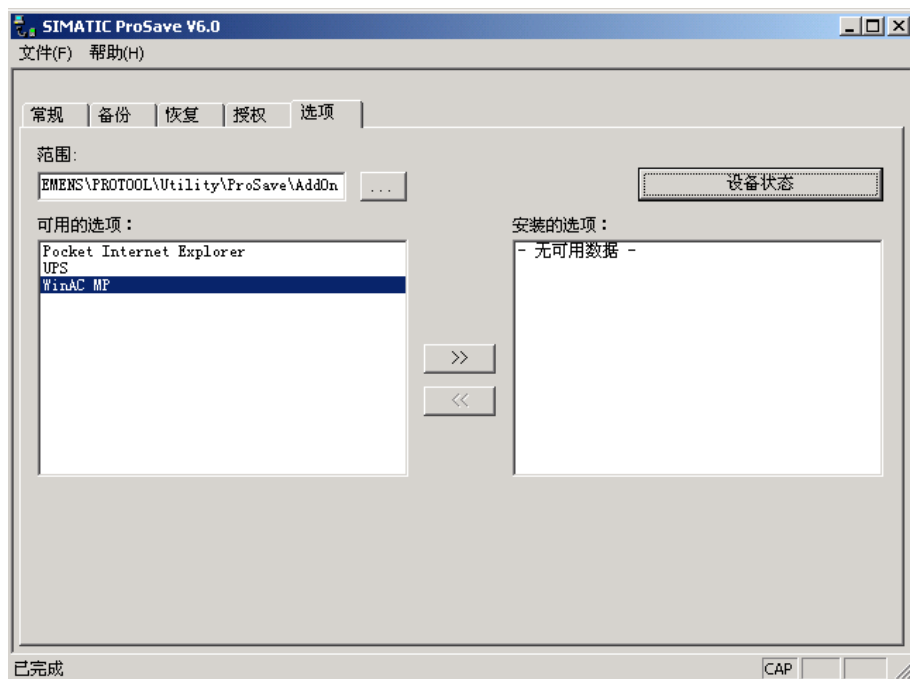


图 6.2 SIMATIC PROSAVE OPTION 设置

几分钟之后（时间依赖于通讯方法），组态计算机将文件移植到多功能面板中。多功能面板接收到文件，不压缩直接保存到 RAM/flash 内存中。移植过程中，组态计算机显示移植进程。移植结束，信息框提示移植成功。下载完成，提示重新启动多功能面板。WinAC MP 控制引擎已经安装在多功能面板中（直到下载了带有 WinAC MP 的 ProTool 项目，你才有到控制引擎的人机接口）。不要关闭 ProSave，你还需要下载 WinAC MP 的授权。

## 6.4 下载 WinAC MP 授权

你可以使用 ProSave 将 WinAC MP 的授权从授权盘移植到多功能面板。如果 ProSave 没有打开，在 Windows 启动菜单中单击 SIMATIC->ProSave V6.0->ProSave 打开 ProSave。

为了使用 ProSave 将 WinAC MP 的授权移植到多功能面板，采用以下步骤：

- 在 ProSave 中单击 Authorize 页
- 选择插入授权盘的适当驱动器。在 Selection 框中显示授权。对于 WinAC MP 授权，认证计数器将显示数值 001，该数值表明认证在磁盘中。（为了判断授权是否已经安装，你可以使用 Device Status 按钮读多功能面板）
- 为了将授权从磁盘移植到多功能面板，在 Selection 框中选择产品授权，并使用>>按钮将授权添加到 Installed 选项中

授权移植完毕，你可以退出 ProSave。

当你备份多功能面板中的存储器时，你不能备份 WinAC MP 的授权。因此你必须将 WinAC MP 授权移植到磁盘中，再重新安装授权。具体步骤如下：

在 ProSave 中的 Authorize 页，使用<<按钮将授权认证从多功能面板（右框，Installed 选项）移植到磁盘中（左框，Selection）。

## 附录 WinAC 技术参数

### 软 PLC 和插槽 PLC

特性	软 PLC		插槽 PLC	
	WINAC Basis V4.1	WINAC RTX V4.1	WINAC Slot 412 V3.4	WINAC Slot 416 V3.4
基本特点： 主存储区（集成）  I/O 数 位存储区 S7 计数器/ S7 定时器 外插存储卡	可用的 PC 内存  16/16K 2K 512/512 无	可用的 PC 内存 （无寻址内存） 16/16K 2K 512/512 无	128K 代码+128K 数据 4/4K 4K 256/256 有	6M 代码+1.6M 数 据 10/10K 16K 512/512 有
数据保持： 没有 UPS 和电源扩展 有 UPS 有电源扩展和电池	不能保存 所有数据 不能保存		不能保存 不能保存 所有数据	
功能块数	受 PC 可用内存的限制		256FB 256FC 512DB	2048FB 2048FC 4095DB
指令执行时间： 位操作 整数运算 浮点数运算 参考平台 确定性	0.013μs 0.025μs 0.025μs Pentium 2.4G 不支持	0.013μs 0.025μs 0.025μs Pentium 2.4G 支持	0.2μs 0.2μs 0.6μs 独立于 PC 支持	0.08μs 0.08μs 0.48μs 独立于 PC 支持
多处理器模式	1 个软 PLC 和最多 3 个插槽 PLC		最多 4 个插槽 PLC	
DP 连接数： 集成 CP5611/集成到 SIMATIC PC 接口 (最大) CP5613 (最大)	4 无 1  4	4 无 无  4	2 2 无  无	
通讯功能： PG/OP 通讯 S7 通讯，做服务器 S7 通讯，做客户端 PROFINET	支持 支持 支持 支持(WINAC PN)	支持 支持 支持 不支持		
过程数据存取： OPC	支持			

通过 SIMATIC 接口的 HMI: <b>PROTOOL/PRO WINCC/WINCC flexible</b>	支持 支持
工艺技术: <b>SIMATIC FM</b>  简单运动控制 C/C++接口	FM 350(计数)/351(快速行程和爬行回馈定位)/352(电子凸轮)/353(步进电机定位)/354(伺服电机定位)/355(闭环控制) 支持 支持,带有 ODK
<b>PLC 编程软件</b>	STEP7 V5.2 以上版本, 工程工具 (可选)
操作系统: <b>WINDOWS NT 4.0 WS</b> <b>WINDOWS2000 专业版</b> <b>WINDOWS XP 专业版</b>	不支持 支持, SP3 版本以上  支持, SP1 版本以上

#### **WINAC ODK**

<b>性能</b>	<b>WINAC ODK V4.1</b>
<b>WINAC 产品支持</b> <b>WINAC Basis V4.1</b> <b>WINAC RTX V4.1</b> <b>WINAC Slot V3.4</b>	支持 支持 支持
<b>系统要求:</b> <b>硬件</b> 最小主存储区 硬盘 <b>操作系统</b> <b>WINDOWS NT 4.0 WS</b> <b>WINDOWS2000 专业版</b> <b>WINDOWS XP 专业版</b> <b>PC 编程软件</b> <b>MS Visual Development Studio.V6.0</b> <b>VenturCom SDK</b> <b>附属设备</b>	Intel Pentium 800M 256M RAM 10M 空白区  不支持 支持, SP3 版本以上 支持, SP1 版本以上  支持, SP3 版本以上 只适用于 WINAC RTX 显示器、键盘、鼠标

#### **WINAC MP**

<b>性能</b>	<b>WINAC MP V3.1</b>
<b>用户内存:</b> 闪存 (集成) 工作内存 (集成) 装载内存 (集成) 位存储区 计数器	5M 1M 1M 2K 512 512

定时器 数据保存	带有 UPS 的可支持
块数量: <b>FB/FC/OB/DB/SDB</b>	最大 2500
<b>I/O:</b> I/O 地址区 I/O 数 I/O 连接 DP 从站数	每个 16KI/O 每个 1KI/O PROFIBUS 最大 12Mbits/s (MP370 集成) 32
执行时间: 位操作 算术运算	0.2μs 0.15μs
工艺技术: <b>SIMATIC FM</b> 简单的运动控制	FM350/351/352 支持
系统要求: 硬件 操作系统 PLC 编程软件 可视化组态软件 工业以太网通讯软件（需要编程设备）	MP370 12” TOUCH, 12”KEYS, 15”TOUCH WINDOWS CE 3.0(预装在 MP370 中)  STEP7 5.2 版本或以上 PROTOOL 6.0 版本 SP2 或以上 用于 IE 的 SOFTNET PG
通讯功能: 总连接 以太网（最多） <b>PROFIBUS（最多）</b> 保持的 OS 连接 保持的 PG 连接 <b>PG/OP 通讯</b> 全局数据通讯 <b>S7 基本通讯</b> <b>S7 通讯</b> 做服务器 做客户端	24 22 4 1 1 支持 不支持 不支持  支持 支持