

# applications & TOOLS

通过Excel VBA中的OCX Data Control实现简单的  
生产可视化

**SIEMENS**

应用范例

**注意事项**

这些应用实例不是固定不变的，也不表示完全考虑了所示电路、装备和任何可能的偶发事件。这些应用实例也不代表客户特定的解决方案。这些应用实例并未免除用户要安全、专业地使用、安装、操作和维修设备方面的责任。使用这些实例时，用户已经认可西门子不为超出其所述责任条款外的任何损害或赔偿要求负责。我们保留在没有预先通知情况下，在任何时间修改这些应用实例的权力。如这些应用实例提供的建议与其它西门子出版物，如：目录，有任何不同的话，则以其它文档的内容为准。

## 担保、责任和支持

我们对本文档中包含的信息不承担任何责任。

由于使用本应用实例中描述的实例、信息、程序、工程设计和性能数据等而导致的任何索赔要求，不论是基于何种法律缘由，我们都会拒绝。该责任免除不适用于下列一些强制性责任，如根据德国产品责任法令

（“**Produkthaftungsgesetz**”），故意触犯、重大过失，或对生命、身体或健康造成伤害，对产品质量的担保，对缺陷的欺骗性隐瞒或影响协约确立的违规（“**wesentliche Vertragspflichten**”）。然而，除了有意触犯或者重大过失或基于强制责任的生命、身体或健康伤害方面，影响协约确立的违规的索赔必须限定在合约里的预计损失内。上述限制性条款并不意味着改动用户提供所受损失的证据的责任。

**版权所有© 2006 西门子自动化与驱动。未经西门子自动化与驱动部门的书面授权，禁止转让或拷贝这些应用实例或摘录实例内容。**

如果您对本文档有任何建议，请通过以下 e-mail 地址将其发给我们：

<mailto:csweb@ad.siemens.de>

## 前言

### 本应用范例的目标

国际标准 OPC 是从 Windows 应用程序访问 SIMATIC S7 站点中的过程数据的理想接口。

本文中的应用范例演示了 Microsoft Excel 和 VBA 的使用方法，通过非常简便和经济的操作过程创建单个可视化用户界面。在本应用实例中，通过一台 SIMATIC S7 控制器模拟一个简单的生产设备。在 Excel 应用程序(OPC 客户端)和 S7-CPU 之间互相交换所需的过程信息。

### 本应用范例的主要内容

本应用范例中主要讨论了下列问题：

- 可视化/编程
  - 创建一个 Microsoft Excel/ VBA 应用程序以及 SIMATIC NET 的 OPC ActiveX 控件
  - 通过 OCX Data Control 管理用于 OPC 数据访问的 OPC 标准接口。
  - 将 SIMATIC NET ActiveX 控件与 OCX Data Control 互联
  - 将 Excel 标准控件与 S7 变量互联
- 组态
  - 为 PC 站点组态一个 OPC 服务器，并通过 MPI 接口与 S7 CPU 相连接。

### 不包括的内容

本应用范例中不包含：

- Microsoft Excel 基础知识
- VBA 编程基础知识
- LAD/FBD/STL 基础知识

我们假设用户已经具备上述知识。

## 本文档的结构

本应用范例的文档分为下列几个主要部分。

部分	说明
应用范例概述	对整个内容做了简要概述。用户可了解所用到的组件(标准硬件和软件以及专门创建的软件)。
功能原理和程序结构	本部分详细描述了所用硬件和软件组件的工作过程、解决方案的结构,以及该应用程序的特定实现方式。如果您想要了解方案中所用组件之间的互动关系,以便以后在此基础上进行自行开发,那么您就需要阅读本部分。
应用程序的结构、组态和操作	本部分逐步向您讲解了该应用范例的结构、重要组态步骤、调试以及程序的操作。
附录	本部分文档包含了补充信息,如文献、术语表等

## 参考自动化和驱动部的服务和支持网站

本条目来源于 A&D 服务和支持网站。点击下面的链接可以直达本文档的下载页面。

<http://support.automation.siemens.com/WW/view/en/23829402>

## 目录

目录 .....	5
应用范例概述 .....	7
<b>1 自动化任务 .....</b>	<b>7</b>
1.1 概述 .....	7
1.2 详细需求 .....	8
<b>2 自动化解决方案.....</b>	<b>9</b>
2.1 解决方案总述 .....	9
2.2 核心功能描述 .....	9
2.3 所需硬件和软件组件.....	14
2.4 替代解决方案 .....	15
2.4.1 替代 Excel 中的 OCX Data Control .....	15
2.4.2 用其它高级语言编写一个 OPC 客户端程序.....	15
2.4.3 使用标准 HMI 系统.....	16
工作原理和程序结构 .....	17
<b>3 通用工作机理.....</b>	<b>17</b>
3.1 OPC 基本知识.....	17
3.2 多点接口 .....	20
<b>4 本应用程序的工作机理.....</b>	<b>21</b>
4.1 OCX 数据控制的工作机理.....	21
4.2 OCX 数据控制的组态示例.....	29
4.3 STEP7 范例程序 .....	31
<b>5 范例程序说明.....</b>	<b>33</b>
5.1 OPC 客户端的故障处理 .....	33
5.2 VBA 范例程序说明 .....	33
5.3 STEP7 程序说明 .....	38
应用程序的结构、组态和操作.....	41
<b>6 安装和调试 .....</b>	<b>41</b>
6.1 安装硬件和软件.....	41
6.2 组态 OPC 客户端 .....	42
6.3 组态组件组态器 .....	45
6.4 组态工程 PG/PC .....	46
6.5 调试 SIMATIC PC 站点 .....	47

6.6	调试 SIMATIC S7 站点.....	48
<b>7</b>	<b>组态过程.....</b>	<b>49</b>
7.1	在 SIMATIC S7 站点中组态一个新 CPU .....	49
7.2	更改 VBA 项目中的 ItemID.....	54
<b>8</b>	<b>操作应用程序.....</b>	<b>57</b>
附录和参考文献列表 .....		<b>60</b>
<b>9</b>	<b>文献.....</b>	<b>60</b>
9.1	参考文献 .....	60
9.2	Internet 链接.....	60
9.3	其它 OPC 应用参考书目.....	61

## 范例应用程序概述

### 内容

本部分概要介绍了本应用程序。您将了解所用到的组件(标准硬件和软件以及专门创建的软件)。

## 1 自动化任务

### 这里您将了解...

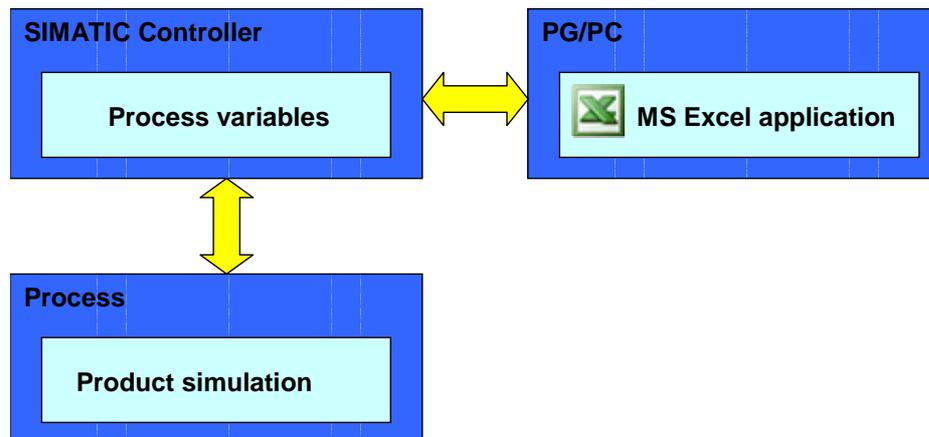
本文档中所讨论的自动化任务。

### 1.1 概述

#### 自动化任务概述

下图对自动化任务进行了概括描述。

图 1-1



#### 自动化任务描述

在低性能应用领域，通过 MS Office 方式，为 S7 CPU 建立一条经济的连接，实现小型生产设备单独的可视化用户界面。

通过用 VBA 编程的 OPC 客户端程序，本范例应用程序向您展示了如何让生产过程可视化。点击用户接口上的元件将影响 S7 CPU 中用户程序的行为，S7 CPU 上的用户程序对此过程进行模拟。另一方面，还可以从控制器读取大量数据并在可视化元件上进行显示。

## 1.2 详细需求

### 自动化任务的详细需求

- 应用项目必须具有成本效益。
- 整个生产过程必须在 Excel 的 VBA 用户接口中可视。
- 通过 S7 CPU 的 MPI 接口连接控制器。
- 该应用范例将演示如何在 VBA 中通过 OCX 数据控制使用 SIMATIC NET OPC 服务器。
- 实现一个简单的
  - 操作员用户接口用于控制设备
  - 配方管理
  - 显示临时过程

### 对可视化用户接口的要求

- 操作员用户接口的设计应当简单。
- 通过图形化的用户接口控制和显示 S7 CPU 中的过程变量。
- 将演示如何切换和显示 S7 过程数据，使用：
  - SIMATIC NET ActiveX 控件(无需编程技术)以及
  - Excel 标准控件(单独编程)
- 将演示如何向 S7 CPU 写入或者从中读取大量数据。
- 将演示如何在 VBA 中将最普通的 S7 变量类型进行转换并显示。

### 控制器要求

- 模拟一个工艺过程
- 不使用专门的通信功能块而实现通信(S7 通信的变量服务)

## 2 自动化解决方案

这里您将了解如下信息...

为自动化任务选择的解决方案。

### 2.1 解决方案总述

显示

下图显示了解决方案中最重要的组件：

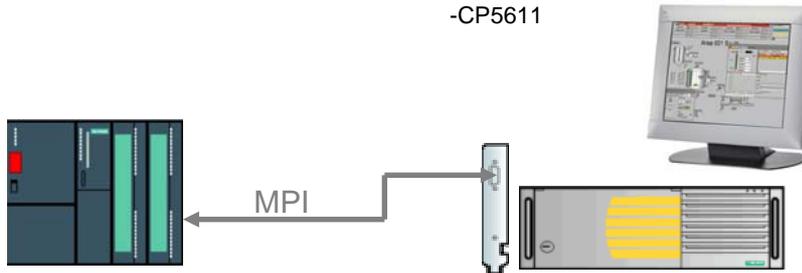
图 2-1

#### S7-300 Station

- PS307 5A
- CPU 313-C

#### PG/PC

- Windows XP Professional
- Office 2003
- STEP7 V5.4
- SIMATIC NET PC Software 2005 V 6.3
- CP5611



配置

自动化解决方案的硬件由一个 S7 300 站点和一个 PG/PC 组成，两者通过 MPI 实现内部连接：

- PG/PC 通过 CP5611 A2
- CPU 通过集成 MPI 接口

### 2.2 核心功能描述

范例程序的功能范围

Excel 项目文件夹包含三个工作表，作为模拟简单制冰的例子。

下表告诉您哪个工作表如何实现何种主要功能。

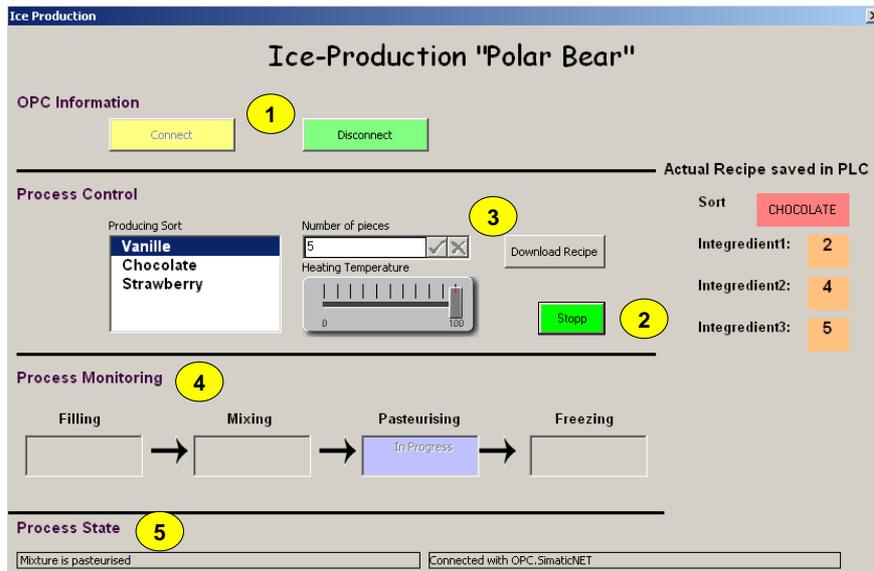
表 2-1

工作表名	功能
Process(过程)	<ul style="list-style-type: none"> <li>带有控件的 UserForm, 用于生产</li> <li>顺序图的状态显示</li> </ul>
Recipe(配方)	<ul style="list-style-type: none"> <li>生产的配方管理</li> </ul>
Archive(归档)	<ul style="list-style-type: none"> <li>用图形记录过程序列</li> <li>归档的生产数据</li> </ul>

## 操作界面 “Process”

“Ice Production(制冰)” 对话框是 VBA 中的一个 UserForm, 激活 “Process(过程)” Excel 表时, 该对话框激活。此对话框中集成了 OCX Data Control。

图 2-2



## 用户界面的功能

对话框的用户界面实现下列功能:

表 2-2

操作元件	功能
1.	连接和断开 OPC 服务器。
2.	启动和停止控制器。
3.	选择产品类型、生产量和温度并将配方数据装入 CPU。
4.	过程监视。
5.	过程状态显示。

## 操作界面 “Recipe”

Excel 工作表的用户界面显示每种类型产品成分的设置值。在对话框中激活下载配方(Download Recipe)按钮时，可以在 CPU 中改变和修订这些值。

下面的数据显示了 Excel 工作表配方。

图 2-3

Indication of the quantity required for each recipe and ingredient in procent	
<b>Vanille</b>	
Integredient1:	9
Integredient2:	10
Integredient3:	5
	
<b>Chocolate</b>	
Integredient1:	2
Integredient2:	4
Integredient3:	5
	
<b>Strawberry</b>	
Integredient1:	30
Integredient2:	30
Integredient3:	30
	

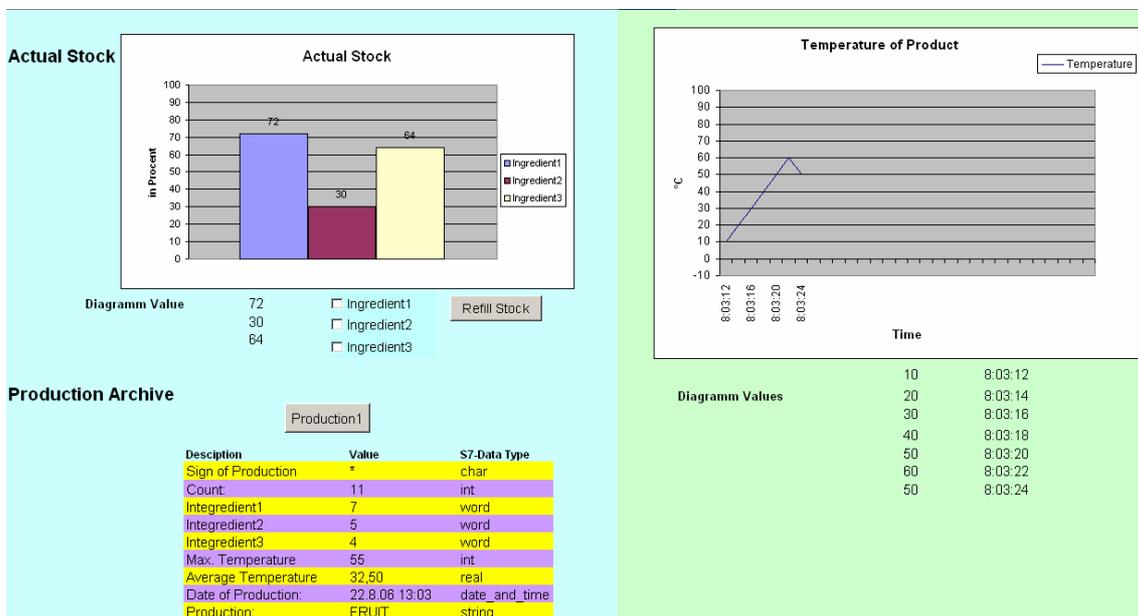
## 操作界面 “Archive”

此工作表的用户界面实现了三个不同的功能：

- **实际库存：**以图形显示当前库存。
- **生产档案：**上次生产的配方。
- **温度记录：**以图形形式记录生产过程中的温度变化。

下列屏幕截图展示了 “Archive” 工作表。

图 2-4



## 此应用程序解决方案的优势

- 具有成本效益的 **MPI** 接口作为标准配置存在于任何 **CPU** 上；因此无需配置昂贵的 **CPU**。
- 硬件配置最低。
- 因为 **VBA** 编辑器是 **Microsoft Office** 的集成部分，所以无需额外的开发环境。

## OPC 应用程序的优势

将 **SIMATIC NET OPC** 服务器应用于生产可视化有下列优势：

- 具有成本效益，因为 **OPC** 服务器属于 **SIMATIC NET** 软件的供货范围。
- 服务器组态简单(如同在 **STEP 7** 中一样)。
- 高效的数据交换，从一个过程变量传送到一个应用程序以便做进一步处理。
- 由 **OPC** 基金会维护和发布。
- 数据包最大可达 **65Kbytes**。
- 使用简单的脚本语言进行 **OPC** 编程。

## 2.3 所需的硬件和软件组件

### 硬件组件

表 2-3

组件	MLFB /订货号	备注
PS305 5A	6ES7 307-1EA00-0AA0	或类似的 PS
CPU 313C	6ES7 313-5BE01-0AB0	功能相当的 CPU 也可以在此应用中作为替代品
微存储卡 8 Mbyte	6ES7953-8LP11-0AA0	或类似的产品
现场 PG 或类似的 PC 且带有 PB/ MPI 卡	6ES7 711-xxxxx-xxxx	组态计算机 <a href="#">链接至 PG 组态器</a>
CP5611 A2	6GK1561-1AA01	可选择：MPI 卡用来与 PC 相连接
MPI 电缆	6ES7 901-0BF00-0AA0	

### 标准软件组件

表 2-4

组件	MLFB /订货号	备注
SIMATIC S7 STEP7 V 5.4	6ES7810-4CC08-0YA5	
SIMATIC NET 软件 2005 版	6GK1704-0AA07-3AA0	OPC-Server, OCX Data Control
SIMATIC NET PB SOFTNET-S7/2005	6GK1704-5CW63-3AA0	软件+许可证
Microsoft Office 2003		可从 Microsoft 获取

### 示例文件和项目

下表包含本例中用到的所有文件和项目。

表 2-5

组件	备注
23829402_OPC_DATCON_EXCEL_CODE_v10.zip	该 ZIP 文件包含 STEP 7 项目和 EXCEL 电子表格
23829402_OPC_DATCON_EXCEL_DOKU_v10_d.pdf	本文档

## 2.4 替代解决方案

### 2.4.1 替代 Excel 中的 OCX Data Control

使用 SIMATIC NET OPC 服务器的自动化接口编程一个 OPC 客户端。

#### 优势:

- 通过脚本语言 VBA 轻松编程
- 直接在 Excel 电子表单或者 UserForm 上创建用户界面
- 集成 ActiveX 组件

#### 不利因素:

- 因使用脚本语言而使性能受到限制
- 只允许中等数目的变量

### 2.4.2 使用其它高级语言对 OPC 客户端进行编程

#### C/C++/C# 使用用户接口

#### 优势:

- 因使用高级语言，从而可以获得高性能。
- 更强的表现力和灵活性。
- 允许更多数目的变量。
- 允许同时执行多个功能。
- 使用 .NET 语言 C# 可进行 Web 编程。

#### 不利因素:

- 需要较长时间才能熟练运用。
- 创建复杂用户界面所需费用较高。
- 使用 C# 需要额外的 Runtime Callable Wrapper(运行时可调用包装器)作为访问 OPC 服务器的中间层。

#### Visual Basic V 6.0 使用自动化接口

#### 优势:

- 集成了 ActiveX 组件。
- 因为容易入门，所以开发应用程序仅需较少时间。

#### 不利因素:

- 由于使用了 Visual Basic Runtime 和 Wrapper 作为额外的软件层，因此性能不甚令人满意。
- 不能同时执行多个功能。

### 2.4.3 使用标准 HMI 系统

用 WinCC 或 WinCC flexible 作为 OPC 客户端。

#### 优势:

- 集成了操作控制和监视、报表、应答和归档功能
- 无需编程
- 快速创建用户接口
- 可同时访问不同的 OPC 服务器

#### 不利因素:

- WinCC 和 WinCC flexible 软件的采购费用很高

## 工作原理和程序结构

### 内容

本部分讨论所用硬件和软件组件的详细机理、解决方案结构，以及该应用程序的具体实现过程。

如果不想了解解决方案中所用组件之间的互动关系，则不必参考本部分。

## 3 通用工作机理

### 此处您可了解...

适用于 OPC 的通用工作机理。

### 3.1 OPC 基础知识

#### 什么是 OPC?

OPC 是一个独立于制造商的软件接口，它基于 COM/DCOM 技术，该技术允许来自不同制造商的硬件和软件之间进行数据交换。OPC Foundation(OPC 基金会)是多家知名厂商的一个利益组织，创建了下面四个 OPC 技术规范。

表 3-1

技术规范	应用
OPC Data Access (DA)	访问过程数据
OPC Alarm& Events (A&E)	基于事件的信息接口(包括应答功能)
OPC Historical Data Access (HDA)	数据归档功能
OPC Data eXchange (DX)	服务器到服务器的交叉通信。

本例单独使用“OPC Data Access”规范。

OPC 是客户机/服务器架构。过程数据供应模块的制造商为他们的模块提供 OPC 服务器，以连接不同的数据源。OPC 客户端与服务器通信，进行数据读写。

## OPC DA 接口

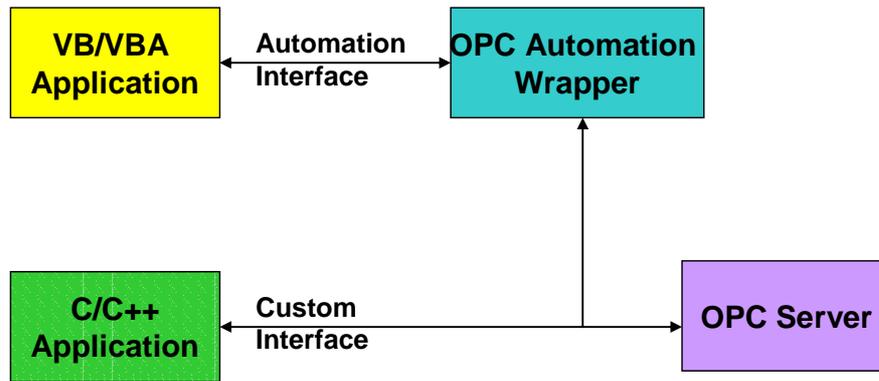
OPC DA 规范是第一个 OPC 标准。用于在控制器和 HMI 设备或其它客户端之间交换过程数据。读取的数据包含数据类型、时间戳和数量，以及关于数值数量的注释。客户端可以读取数据以及向 OPC 服务器写入命令。然后服务器将控制数据传送到控制器。

## 自动化和客户接口

OPC 向 Data Access(数据访问)提供两种接口：客户接口和自动化接口，允许客户端通过这些接口调用服务器功能。接口可用于访问过程变量(数据存取)以及处理事件和报警。

下图说明了其中的关系：

图 3-1

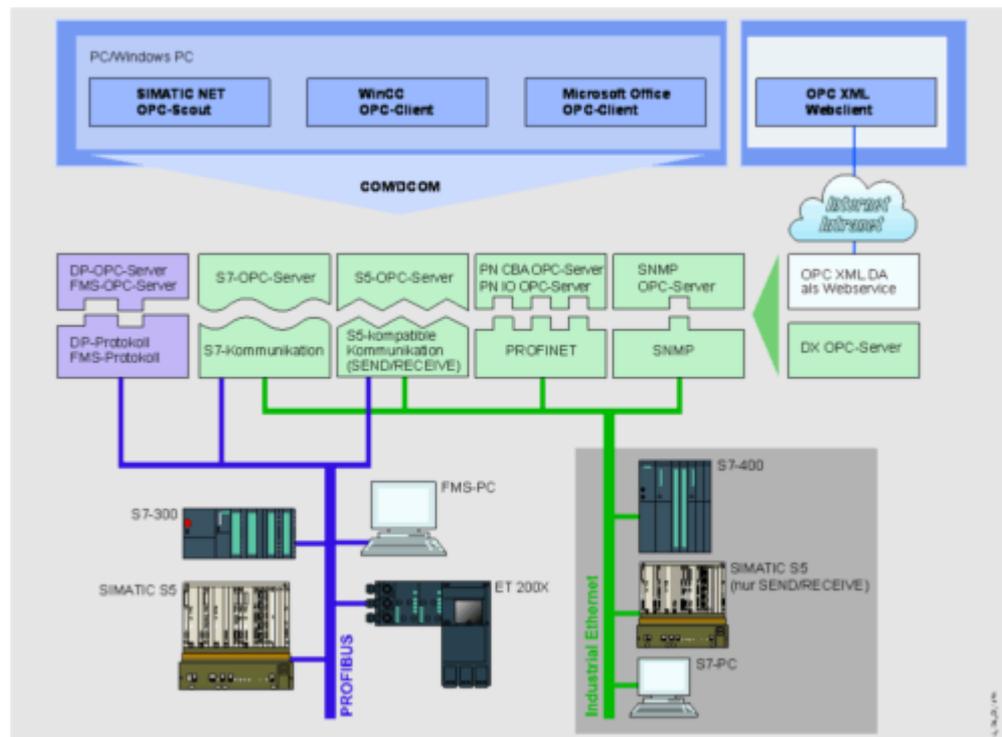


## SIMATIC NET OPC 服务器

下图展示的结构中包含不同的 SIMATIC NET OPC 服务器和不同的协议驱动。

在本应用范例中，使用了 S7 OPC 服务器，通过 MPI 建立了与 S7-300 之间的通信。

图 3-2



SIMATIC NET OPC 服务器提供了下列访问方式：

- 工业以太网
  - S7 通信
  - S5-兼容通信
  - PROFINET
  - SNMP
- PROFIBUS
  - DP 协议
  - FMS 协议
  - S7 通信
  - S5 兼容通信

## 3.2 多点接口

每个 SIMATIC S7 CPU 都有一个 MPI(多点接口)。通过该接口可以在自动化系统中建立一个子网，HMI 和编程设备通过该子网交换数据。

表 3-2

参数	技术数据
站点数量	最多 32 个
传输速率	19.2 Kbit/s 187.5 Kbit/s 12 Mbit/s
网络扩展	段长度 50 m
传输介质	屏蔽双股线

## 4 本应用程序的工作机理

这里您可以了解...

OPC 客户端提供的功能，OPC 服务器和 OPC 客户端的工作机制如何协调配合，以及这些功能如何在一个 VBA 项目中实现。

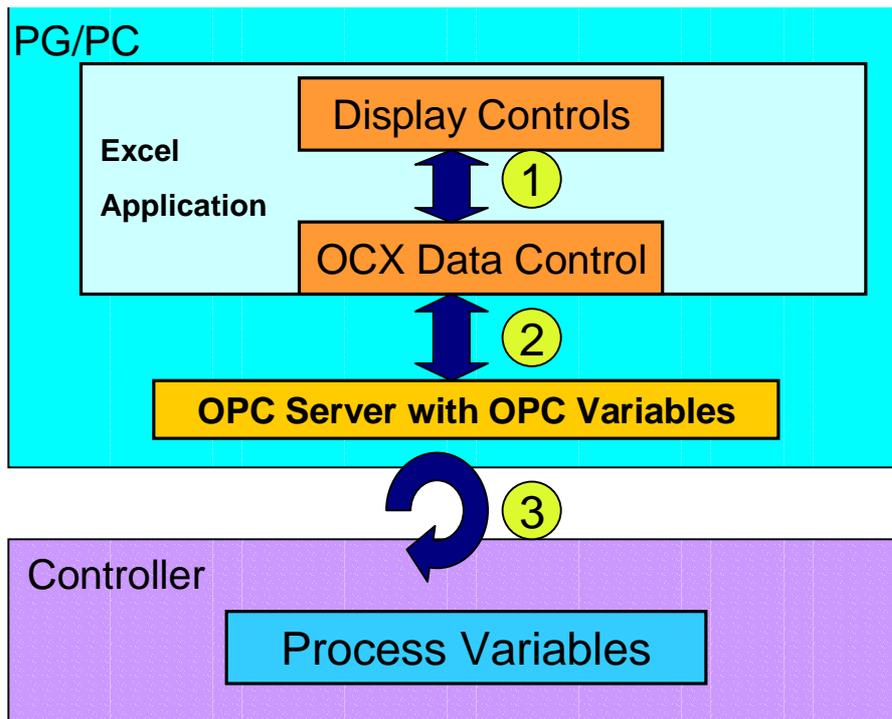
### 4.1 OCX Data Control 的工作机理

核心组件为 SIMATIC NET OCX Data Control 。没有该组件，其它任何 SIMATIC 控件均不能访问过程数据。OCX Data Control 与 SIMATIC NET OPC 建立连接并访问过程数据。程序运行期间该控件不可见。

显示

下图展示了显示控件和 OPC 过程变量之间的连接

图 4-1



其中的编号含义如下:

表 4-1

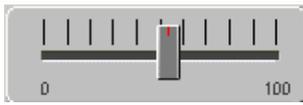
编号	说明
1.	OCX Data Control 向控件提供数据。 否则, 控件向 OCX Data Control 发出写命令。
2.	OPC 服务器通过 Data Access Interface(数据访问接口)向 OCX Data Control 提供过程值。
3.	OPC 服务器持续监视指定过程变量的变化并处理对过程变量的写操作。

## SIMATIC NET 可用的显示控件

显示控件是实现过程数据可视化的元件。它们通过 SIMATIC NET OCX Data Control 接收数据, 而不是直接通过访问 OPC 或其它接口获取数据。这些控件允许创建 HMI 用户接口而无需编程。

下表显示了现有的显示控件:

表 4-2

控件	外观	简介
按钮 (Button)		存取 OPC 服务器中存储的单个位。按钮控件只能识别 2 个值: 0 = 启动 1 = 停止 按下按钮时, 相应的位也随之改变。
滑块 (Slider)		通过变量按字节、字和双字格式存取数据。如果滑块控件的值发生了变化, 则其所分配变量的值也随之改变。
数字框 (Number)		通过变量按字节、字和双字格式存取数据。如果数字框控件中输入了一个新值, 则指定变量的值也随之改变。
标签 (Label)		显示变量的值。标签控件只能显示过程数据的值。

**注意事项** 显示控件必须放入 Visual Basic 的 UserForm 窗体中。

## OCX Data Control 的方法

OCX Data Control 具有一些处理来自 OPC 服务器或过程作业的方法。最重要的方法罗列如下。

### Connect/ Disconnect

**Connect / Disconnect** 方法允许手动连接 OPC 服务器或者建立连接。如果 OPC 服务器自动连接被禁用，则需要使用此方法。

### ValueChanged

如果因为指定的变量中发生了数值或数量变化从而触发了一个或者多个事件，则调用 **ValueChanged** 方法。事件名和新值或数量在该方法的调用中以数组格式传送。

**ValueChanged** 方法的语法为：

```
DatCon1_ValueChanged(ByVal Count As Long,  
                    ByVal UserIDs As Variant,  
                    ByVal ItemIDs As Variant,  
                    ByVal Values As Variant,  
                    ByVal Qualities As Variant,  
                    ByVal TimeStamps As Variant)
```

其中的参数含义如下：

表 4-3

参数	含义
Count	被触发的事件数目
UserIDs	事件名
ItemIDs	指定变量的 ItemID
Values	变量变化了的数值
Qualities	变量的数量
TimeStamps	事件发生的时间

## ReadVariable

ReadVariable 方法读取 ItemID 所指定过程变量的值。

该方法的语法为：

```
result = DatCon1.ReadVariable (ItemID as String,  
                                Value as Variant,  
                                Quality as Long,  
                                Timestamp as Date)
```

其中的参数含义如下：

表 4-4

参数	含义
ItemID	被读变量的 ItemID
Value	变量的值
Quality	变量的数量
TimeStamp	读作业发生的时间

## ReadMultiVariable

ReadMultiVariable 方法从 ItemID 指定的多个过程变量中读取数值。

该方法的语法为：

```
result = DatCon1.ReadMultiVariable (ItemIDs as String,  
                                       Values as Variant,  
                                       Errors as Variant  
                                       Qualities as Long,  
                                       Timestamps as Date)
```

其中的参数含义如下：

表 4-5

参数	含义
ItemIDs	含有将要被读取的变量 ItemID 的 String 数组
Values	数值的 Variant 数组
Errors	错误状态的 Variant 数组
Quality	被读取数量的 Long 型数组
TimeStamp	被读取数量的 Date 型数组

## WriteVariable

WriteVariable 方法向 ItemID 指定的过程变量中写入某个值。

该方法的语法为：

```
result = DatCon1.WriteVariable (ItemID as String,  
                                   Value as Variant)
```

其中的参数含义为：

表 4-6

参数	含义
ItemID	所述变量的 ItemID
Value	新值

## WriteMultiVariable

WriteMultiVariable 方法向 ItemID 指定的多个过程变量中写入某些值。

该方法的语法为：

```
result = DatCon1.WriteVariable (ItemIDs as String,  
                                   Values as Variant,  
                                   Errors as Variant)
```

其中的参数含义为：

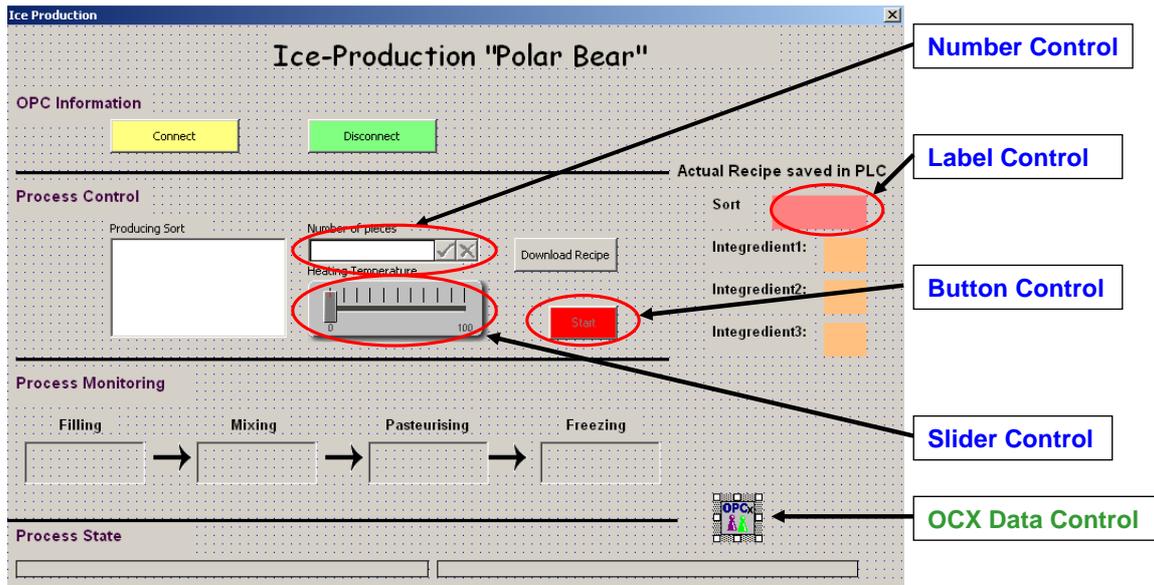
表 4-7

参数	含义
ItemIDs	包含所述变量的 ItemID 的 String 数组
Values	包含新值的 Variant 数组
Errors	包含错误状态的 Variant 数组

## 用户接口中使用的 ActiveX 控件

下图显示了开发模式下对话框中的用户接口。其中包含所有的 SIMATIC NET 显示控件以及数据控件，以彩色显示。

图 4-2



## OPC 过程变量

在 OPC server 中，过程变量由一个 OPC 类代表。所谓的 ItemID 是一个字符串序列，它是过程变量的唯一性标识。它告知服务器哪个过程变量被分配给了 OPC Item。从而，可通过 OPC Item 访问过程变量。因此 ItemID 可以通过绝对地址或符号地址标识过程变量。

### 绝对地址

对于带有绝对地址的 ItemID，有三个选项：

S7:[<Connection Name>]DB <Number>, {<Type>} <Address> {,<Quantity>}

S7:[<Connection Name>]DI <Number>, {<Type>} <Address> {,<Quantity>}

S7:[<Connection Name>]<Object>, {<Type>} <Address> {,<Quantity>}

各个地址区域的含义列表如下：

表 4-8

地址区域	含义	注释
S7	用于访问过程变量的 S7 协议	
<Connection name>	连接的名称	在 NetPro 中组态
DB	数据块；标识该 DB 中的一个变量	
DI	背景数据块；标识该 DI 中的一个变量	
<Number>	DB 或 DI 的编号	
<Type>	S7 数据类型	在 OPC 服务器中，S7 数据类型被转换为对应的 OLE(COM)数据类型
<Object>	S7 数据类型	例如存储器位 M、输入 I、输出 Q
<Address>	将被访问的第一个变量的地址	-字节偏移量 -字节偏移量.位(仅适用 X 数据类型) -字节偏移量.字符串长度(仅适用 String 数据类型，字符串长度为 1 至 254 字节)
<Quantity>	从 Adresse 参数(数值范围 0...65535)指定的偏移量起，将要被寻址的某个类型的变量的数目。	对于 X 数据类型，写访问时输入的数值只能为 8 的倍数。因而位地址必须为零。

## 符号寻址

如果过程变量在控制器中被声明为符号，则 ItemID 也可以是符号。

<Station Name>.<CPU Name>.<DB Name>.<Variable Name>

<Station Name>.<CPU Name>.<DI Name>.<Variable Name>

<Station Name>.<CPU Name>.<Variable Name>

地址区域的含义列表如下：

表 4-9

地址区域	含义	备注
<Station Name>	SIMATIC 站名	可在 STEP7 中组态
<CPU Name>	CPU 名称	可在 STEP7 中组态
<DB Name>	数据块的符号名	可在 STEP7 中组态
<DI Name>	背景数据块的符号名	可在 STEP7 中组态
<Variable Name>	变量的符号名	可在 STEP7 中组态

**注意事项** 为了让 OPC 客户端能够通过 OPC 服务器传递符号变量存取，在 STEP 7 中组态的符号必须被 OPC 服务器接受为系统数据。(参见第 6 章)

## 本应用示例中的 ItemID

下表列出了本应用示例中使用的符号化 ItemID。

表 4-10

符号地址的 ItemID	绝对地址的 ItemID
SIMATIC 300.CPU 313C.PROD_RECIPES.Ingredient1	S7:[OPC]DB6,INT0
SIMATIC 300.CPU 313C.PROD_RECIPES.Ingredient2	S7:[OPC]DB6,INT2
SIMATIC 300.CPU 313C.PROD_RECIPES.Ingredient3	S7:[OPC]DB6,INT4
SIMATIC 300.CPU 313C.DB_ACT_PROD.PRO_SORT	S7:[OPC]DB1,STRING0,30
SIMATIC 300.CPU 313C.DB_ACT_PROD.PRO_COUNT	S7:[OPC]DB1,INT32
SIMATIC 300.CPU 313C.DB_ACT_PROD.PRO_MAX_TEMP	S7:[OPC]DB1,INT40
SIMATIC 300.CPU 313C.DB_ACT_PROD.Start	S7:[OPC]DB1,X44.0
SIMATIC 300.CPU 313C.DB_ACT_PROD.Download	S7:[OPC]DB1,X44.1
SIMATIC 300.CPU 313C.DB_ACT_PROD.Filling	S7:[OPC]DB1,X44.2
SIMATIC 300.CPU 313C.DB_ACT_PROD.Mixing	S7:[OPC]DB1,X44.3
SIMATIC 300.CPU 313C.DB_ACT_PROD.Pasteurizing	S7:[OPC]DB1,X44.4
SIMATIC 300.CPU 313C.DB_ACT_PROD.Freezing	S7:[OPC]DB1,X44.5

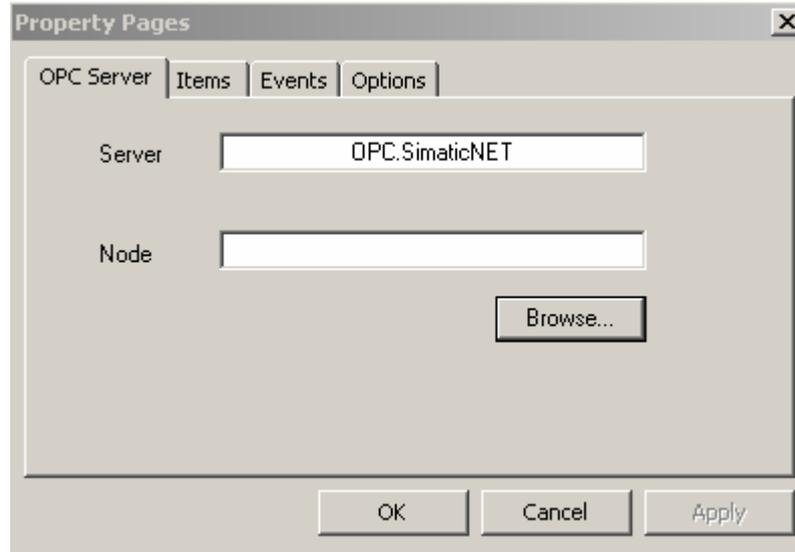
### 注意事项

更多关于 OPC 过程变量的信息，请参见 SIMATIC -> Documentation 下的“Industrial communication with PG/PC Bd2 interface(使用 PG/PC Bd2 接口进行工业通信)”

## 4.2 OCX Data Control 组态示例

在本应用范例中，OCX Data Control 是 OPC 客户端最重要的组件。关于组态，OCX Data Control 有一个属性窗口，其中还包含几个选项卡。

图 4-3



本节举例说明如何将变量与事件进行互联。

表 4-11

步骤	操作	选项卡
1.	选择 OPC server	OPC server
2.	将显示的控件与过程变量相连接	Items
3.	为过程变量分配事件	Events
4.	进行设置	Options

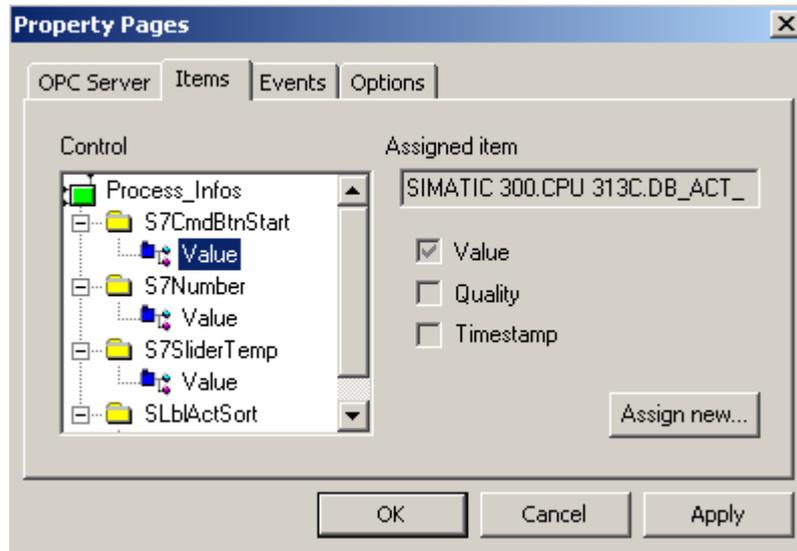
### 选择 OPC 服务器

作为 OPC 客户端，SIMATIC NET OCX Data Control 可以将任何 OPC 服务器作为 OPC 客户端。SIMATIC NET OPC 服务器称为 **OPC.SimaticNET**。通过 OCX Data Control 组态 OPC 服务器便可以让所有分配的变量可被持续监视，如果过程变量的值发生变化，新的值被传送到数据控制模块，然后从那里再到各个显示控件。

## 将显示控件与过程变量相连接

通过数据控制模块访问过程变量。通过 **OCX Data Control** 将显示控件与 OPC 服务器的过程变量的属性(值、数量、时间戳)相连接。

图 4-4

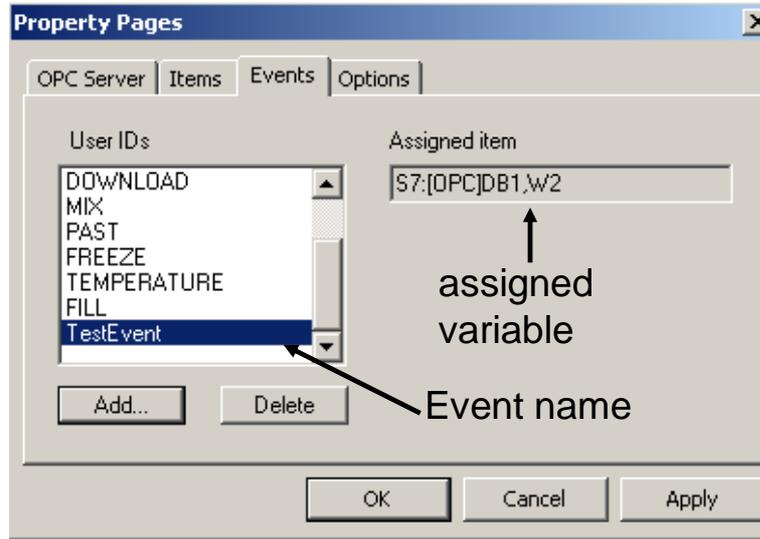


## 将过程变量与事件相连接

显示控件常常不足以用来创建一个更加个性化和复杂的用户接口，除了显示控件外，还可以选择以事件的形式将过程变量与 **OCX Data Control** 相连接。如果变量的状态或值发生了变化，则触发相关的事件，同时 VB 程序可做出相应的反应。

下图显示了一个分配了变量的事件。一旦逻辑上相连的变量的值发生变化，即触发名为“TestEvent”事件。

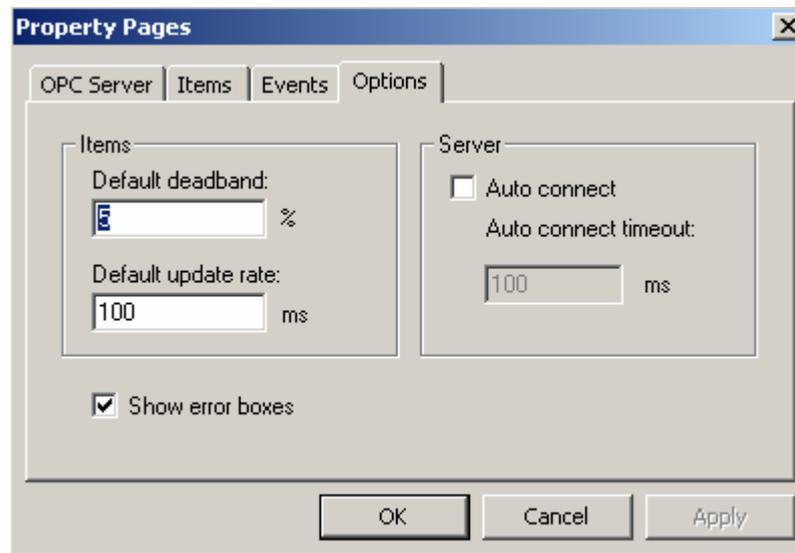
图 4-5



## 进行设置

在 **Options** 选项卡中，可以为 OPC 连接设置缺省值。包括变量的更新时间、连接的断开时间，当调用数据控制模块时，还要自动断开与 OPC 服务器的自动连接。

图 4-6



## 4.3 STEP7 示例程序

下表给出了模拟制冰所用到的 STEP7 模块。

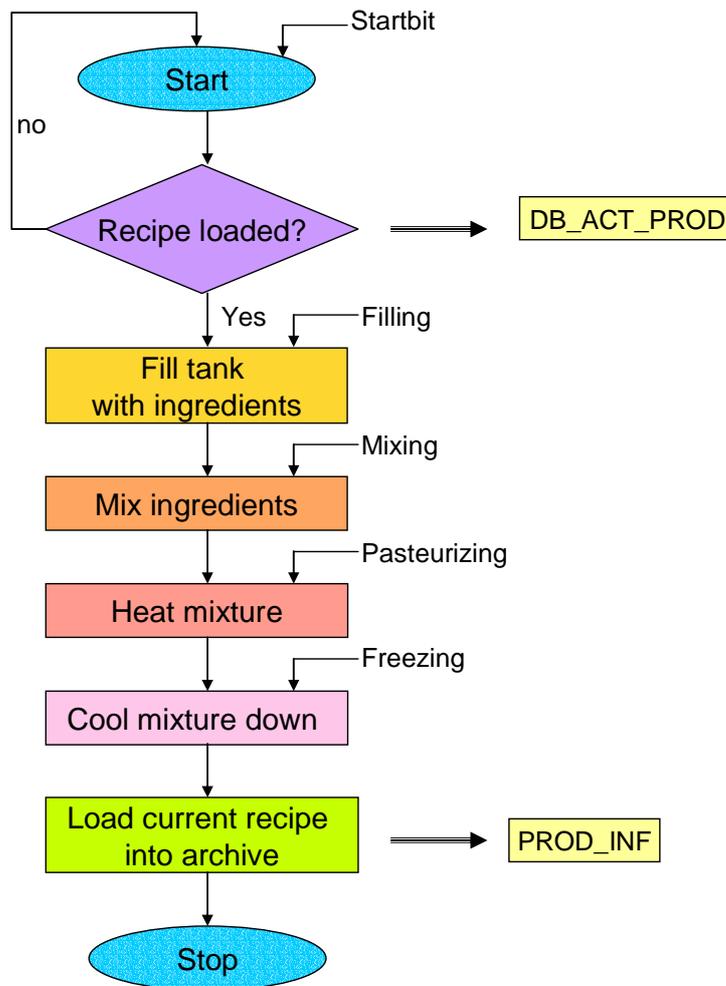
表 4-12

模块	功能
FC 1	“PRO_CONTROL” 功能在一个步序列中模拟制冰。
DB 1	“DB_ACT_PROD” 数据块包含当前的配方数据和控制变量
DB 2	数据块 “PRO_INF” 包含生产过程的生产信息
DB 6	数据块 “PROD_STOCK” 包含库存的实际值

## 模拟程序运行

下图显示为模拟程序的流程图。模拟生产指令存储在 FC1 “PRO\_CONTROL” 中，该功能由 OB1 循环调用。以步序列的方式实现 FC1。

图 4-7



## 5 范例程序的说明

这里您可以了解...

VBA 程序核心程序段程序代码的详细情况，在 OPC 客户端如何实现错误处理例程，以及关于运行 STEP7 程序的信息。

### 5.1 在 OPC 客户端进行错误处理

本节介绍 VBA 程序如何对特定错误进行处理。下列错误将受到处理：

- 发生在连接或者断开连接期间的错误
- 发生在从控制器读取过程变量期间的错误
- 发生在向控制器写入数值期间的错误

在用户界面中处理错误：

- 用户界面中包含一个错误信息输出窗口。
- 与 OPC 服务器建立连接期间发生错误将调用 `ConnectionError` 方法。
- 检查所有 OPC 调用的返回值，查看是否有出错。
- 发生错误时，在输出窗口输出一条消息。
- 许多功能都有一个 `ErrorHandler`，它将在发生错误时输出一条消息。

### 5.2 VBA 范例程序说明

本节将更加详细地介绍 Visual Basic 代码。通过摘录的代码，详细讲解下列核心程序部分：

- 使用 `OCX Data Control` 进行事件分析
- 读取某一块数据
- 将 S7 数据类型转换为 VBA 类型

## ValueChanged 方法

如果触发了一个或多个事件，则调用 OCX Data Control 的 ValueChanged 方法。该方法记录事件名称，以便 VB 程序可以对相应的事件做出响应。下图中摘录了 ValueChanged 方法的程序代码。

图 5-1

```
Private Sub DatCon1_ValueChanged(ByVal Count As Long, _
                                ByVal UserIDs As Variant, _
                                ByVal ItemIDs As Variant, _
                                ByVal Values As Variant, _
                                ByVal Qualities As Variant, _
                                ByVal TimeStamps As Variant)

    'A loop with 'count' passes
    'the value of 'count' is the number of changed variables in PLC
    For i = 0 To Count - 1
        Debug.Print (i)
        Debug.Print (UserIDs(i))
        Events = UserIDs(i)
        Debug.Print (Events)
        Select Case Events
            'Changes of Actual Stock of ingredients
            Case "INGRED1"
                Tabelle3.Cells(19, 3) = Values(i)
            Case "INGRED2"
                Tabelle3.Cells(20, 3) = Values(i)
            Case "INGRED3"
                Tabelle3.Cells(21, 3) = Values(i)
            'Change of State of Process: Filling
            Case "MIX"
                If (Values(i) = True) Then
                    TextBoxSTEP1.Text = "In Progress"
                    TextBoxSTEP1.BackStyle = fmBackStyleOpaque
                    Txt_State.Caption = " Integredients are mixed"
                Else
                    TextBoxSTEP1.Text = " "
                    TextBoxSTEP1.BackStyle = fmBackStyleTransparent
                End If
        End Select
    Next i
End Sub
```

## 读取大量数据

为了减轻 OPC 服务器的负担，可以选择从 CPU 读取包含各种块的数据。  
`ReadVariable` 允许以固定格式从一个缓冲区读取数据。下图为读取一个字节  
型数据块的调用语句。

图 5-2

```
result = Process_Infos.DatCon1.ReadVariable("S7:[OPC]DB2,B0,58", _  
                                           Archiv, _  
                                           quality, _  
                                           timestamp)
```

## VBA 中的数据转换例程

本节中我们向您介绍如何将刚才读取的数据块拆分为原始字节。将 S7 数据  
类型转换为相关的 VBA 类型必须遵守相应的转换规则。

### Byte ↔ VBA Integer

下图为转换为 VBA Integer 类型的程序代码。读取的字节代表 CPU 中的一个  
integer(整数)。

图 5-3

```
Function TwoByteToInt (ByVal LoByte As Byte, _  
                       ByVal HiByte As Byte) As Integer  
'Check if HiByte is 0  
If (HiByte And &H80) <> 0 Then  
    TwoByteToInt = ((HiByte * 256&) + LoByte) Or &HFFFFFF0000  
Else  
    TwoByteToInt = (HiByte * 256) + LoByte  
End If  
End Function
```

## Byte ↔ VBA 字符

下图为转换为一个 VBA 字符的程序代码。读取的字节代表 CPU 中的一个 Char。

图 5-4

```
Function TwoByteToChar(ByVal LoByte As Byte, _
    ByVal HiByte As Byte) As String
    If (HiByte And &H80) <> 0 Then
        TwoByteToChar = Chr(((HiByte * 256&) + LoByte) Or &HFFFF0000)
    Else
        TwoByteToChar = Chr((HiByte * 256) + LoByte)
    End If
End Function
```

## Byte ↔ VBA Word

下图为转换为一个 VBA word 的程序代码。读取的字节代表 CPU 中的一个 word。

图 5-5

```
Function TwoByteToWord(ByVal LoByte As Byte, _
    ByVal HiByte As Byte) As String
    If (HiByte And &H80) <> 0 Then
        TwoByteToWord = BCDToDec(((HiByte * 256&) + LoByte) Or &HFFFF0000)
    Else
        TwoByteToWord = BCDToDec((HiByte * 256) + LoByte)
    End If
End Function
```

## Byte ↔ VBA byte

下图为转换为一个 VBA byte 的程序代码。读取的字节代表 CPU 中的一个 byte。

图 5-6

```
Function S7ByteToByte(ByVal S7Byte As Byte) As Byte
    S7ByteToByte = BCDToDec(S7Byte)
End Function
```

## Byte ↔ VBA string

下图为转换为一个 VBA string 的程序代码。读取的字节代表 CPU 中的一个 string。

图 5-7

```
Function BytesToString(buf As Variant, i As Integer) As String
Dim Sort As String
Dim k
Dim lenght As Integer |
lenght = buf(i + 1)
For k = i + 2 To i + 2 + lenght
Sort = Sort & Chr((buf(k)))
Next k
BytesToString = Sort
End Function
```

## Byte ↔ VBA single

下图为转换为一个 VBA single 的程序代码。读取的字节代表 CPU 中的一个 single。

图 5-8

```
Function S7BytesToSingle(buf As Variant, i As Integer) As Single
Dim z As Single
CopyMemory ByVal VarPtr(z) + 2, CLng(TwoByteToInt(buf(i + 1), buf(i))), 2
CopyMemory ByVal VarPtr(z), CLng(TwoByteToInt(buf(i + 3), buf(i + 2))), 2
S7BytesToSingle = z
End Function
```

## Byte ↔ VBA date

下图为转换为一个 VBA date 的程序代码。读取的字节代表 CPU 中的一个 Date\_and\_Time。

图 5-9

```
Function S7DTToDate(buf As Variant, i As Integer) As Date
Dim Datestr, Timestr As String
    Datestr = CStr(BCDToDec(buf(i + 2))) + "." + _
              CStr(BCDToDec(buf(i + 1))) + "." + _
              CStr(BCDToDec(buf(i)))

    Timestr = CStr(BCDToDec(buf(i + 3))) + ":" + _
              CStr(BCDToDec(buf(i + 4))) + ":" + _
              CStr(BCDToDec(buf(i + 5)))

    DateValue (Datestr) + TimeValue(Timestr)

    S7DTToDate = DateValue(Datestr) + TimeValue(Timestr)
End Function
```

## 5.3 STEP7 程序说明

本节进一步向您详细介绍 STEP7 程序:

- 可视化界面
- 归档数据块
- 库存

### 可视化界面

DB1 “DB\_ACT\_PROD” 是操作员用户界面的数据源。它涵盖了操作员控制 & 监视所需的所有变量。

下图显示了操作员用户界面与数据块之间的关系。

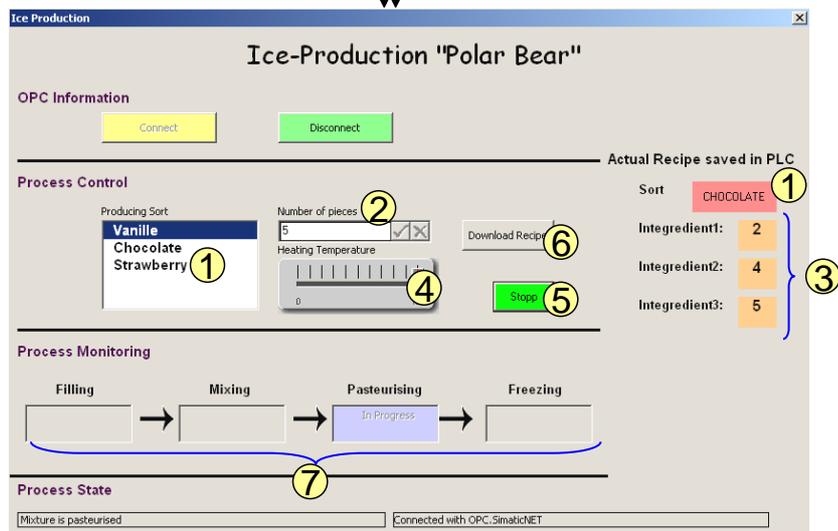
图 5-10

## Data source DB1

+0.0	PRO_SORT	STRING[30]	①
+32.0	PRO_COUNT	INT	②
+34.0	QUANTITY_ISSUED	ARRAY[1..3]	③
*2.0		INT	
+40.0	PROD_MAX_TEMP	INT	④
+42.0	PROD_MIN_TEMP	INT	
+44.0	Start	BOOL	⑤
+44.1	Download	BOOL	⑥
+44.2	Filling	BOOL	⑦
+44.3	Mixing	BOOL	
+44.4	Pasteurizing	BOOL	
+44.5	Freezing	BOOL	

Data exchange

User interface



## 归档数据块 DB 2

数据块“PROD\_INF”包含整个生产过程的信息，例如配方数据、生产时间和平均温度。在 Excel 电子表单“Archive”中可以装载并读取这些数据。

下表为 DB 2 的结构：

图 5-11

+0.0	PRO_SORT	STRING[30]	' '
+32.0	PRO_COUNT	INT	0
+34.0	QUANTITY_ISSUED	ARRAY[1..3]	
*2.0		INT	
+40.0	PROD_MAX_TEMP	INT	80
+42.0	PROD_MIN_TEMP	INT	-10
+44.0	AVERAGE_TEMP	REAL	0.000000e+000
+48.0	PROD_TIME	DATE_AND_TIME	DT#90-1-1-0:0:0.000
+56.0	PROD_SIGN	CHAR	'*'

## 修改选项

在该应用范例中，数据块中的归档数据被下次的生产数据所覆盖。如果想要将生产过程信息保留较长的时间，必须首先相应地调整 STEP7 项目。包括选择将数据块编程为环形缓冲区用来保存生产信息。

## 库存

数据块“PROD\_STOCK”管理各种原料成分的库存。

下表显示了 DB 6 的结构。

图 5-12

+0.0	Integredient1	INT	100
+2.0	Integredient2	INT	100
+4.0	Integredient3	INT	100

## 应用范例的结构、组态和操作

### 内容

本部分带领您一步步地了解本范例应用程序的结构设计、重要组态步骤和操作运行。

## 6 安装和调试

这里您可以了解关于...

即将安装的硬件和软件，以及调试该范例所必须的步骤。

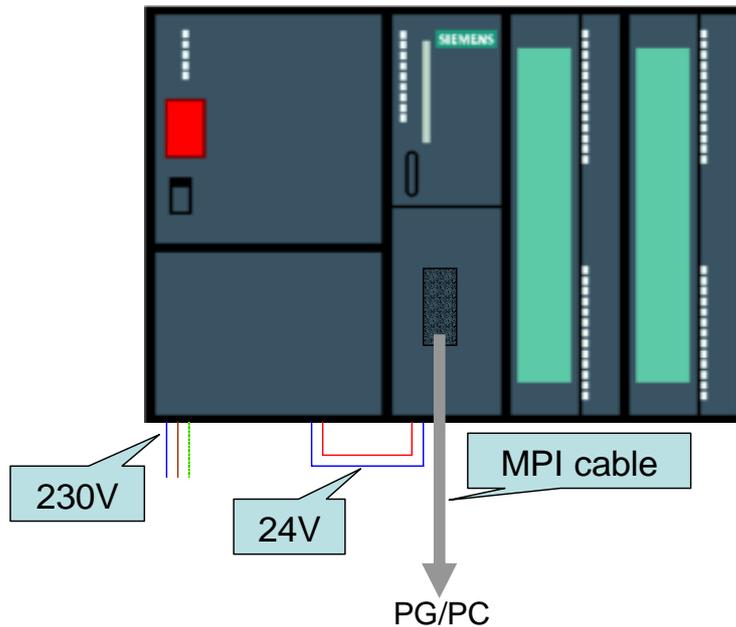
### 6.1 安装硬件和软件

本章讲述将要安装的硬件和软件。任何情况下都应当遵照手册和指南以及相关产品供货信息中的要求进行操作。

#### 安装硬件

按照下图以及第 2.3 节所列的配置表安装 SIMATIC 硬件部件。

图 6-1



**注意事项** 安装各个组件时均须遵守安装指南。

## 安装标准软件

表 6-1

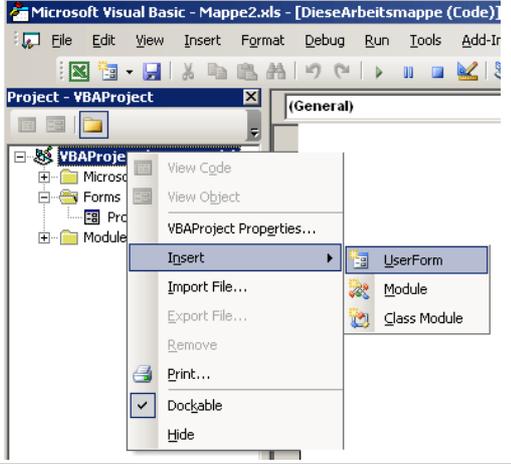
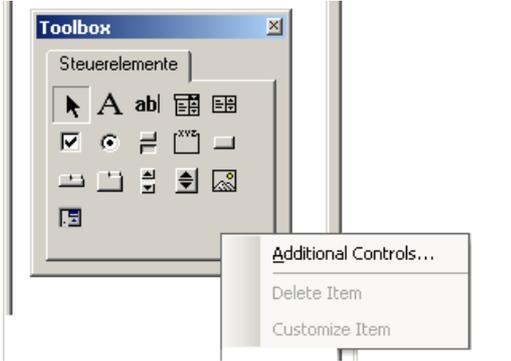
序号	操作	备注
5.	安装 STEP 7 V5.4	按照安装程序的指示进行操作
6.	安装 SIMATIC NET SOFTWARE EDITION 2005	遵循安装程序的指示

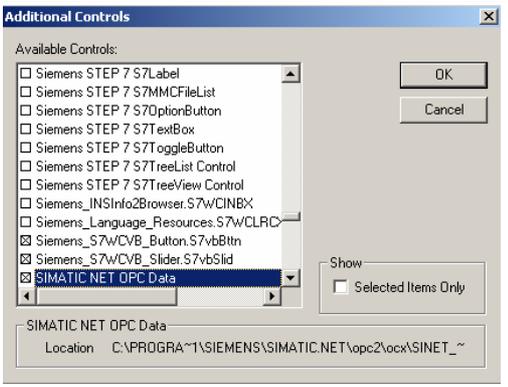
## 6.2 组态 OPC 客户端

下面的指南只是让您更好地理解如何将 S7 控件集成到 VBA，以及如何组态 OCX Data Control。范例应用程序中已经包含了这些组态。

### 将 S7 控件集成到 VBA 中

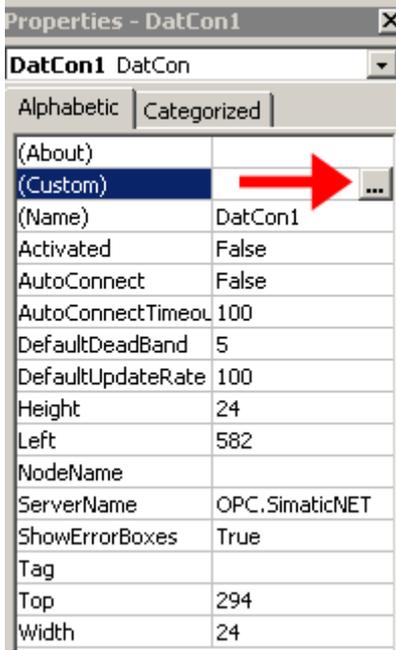
表 6-2

序号	操作	备注
7.	打开范例程序的 Excel 文件夹	
8.	打开文件夹中的 VBA 编辑器	位于 <b>Tools -&gt; Macros -&gt; Visual Basic Editor</b> 下
9.	选择 VBA 项目，通过鼠标右键菜单选择 <b>Insert -&gt; UserForm</b> 打开一个 UserForm。	
10.	右键点击 Toolbox 并选择 <b>Additional Controls...</b> 出现一个新的对话框	

序号	操作	备注
11.	<p>选择显示控件</p> <ul style="list-style-type: none"> <li>Siemens S7Label Control</li> <li>Siemens S7Number Control</li> <li>Siemens_S7WCVB_Button.S7vbBttn</li> <li>Siemens_S7WCVB_Slider.S7vbSlid</li> </ul> <p>及数据控件 SIMATIC NET OPC DATA</p> <p>点击 OK 确认, 关闭对话框。这些控件现在已添加到了 toolbox(工具箱)。</p>	

## 组态 OCX Data Control

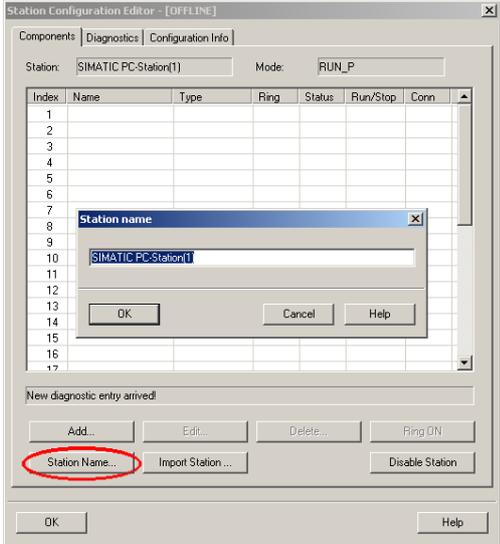
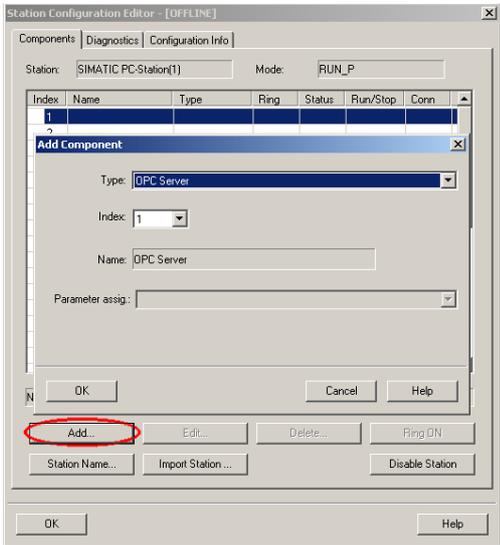
表 6-3

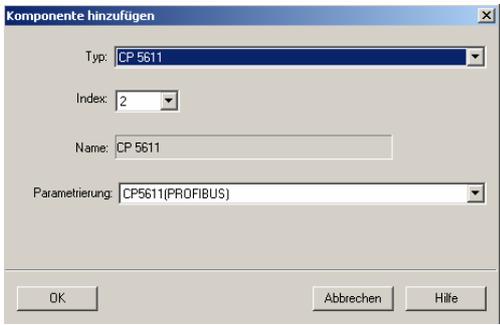
序号	操作	注释
1.	<p>在您的 UserForm 中选择 OCX Data Control 的图标。</p> <p>右键点击 Properties 选项。</p>	
2.	<p>选择 <b>Custom</b> 域, 出现 “...” 按钮。</p> <p>点击该按钮。</p>	

序号	操作	注释
3.	弹出 OCX Data Control 属性页面。 此处，您可以组态 OPC 服务器，将 items 链接到显示控件，并创建事件。	

## 6.3 组态组件组态器

表 6-4

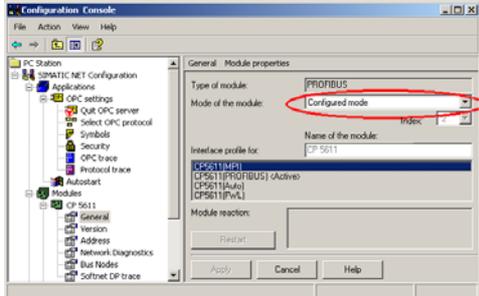
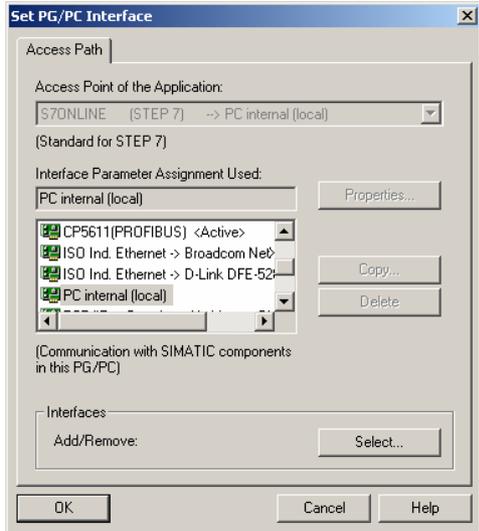
步骤	操作	备注
1.	打开组件组态器	<b>Start -&gt; Station Configurator</b>
2.	在 PC 站点的 <b>Station Name...</b> 下，输入与 STEP7 程序中同样的名字。 在本例中为： <b>SIMATIC PC-Station(1)</b>	
3.	现在，点击第一行并点击“Add...”添加新的 OPC 连接。 弹出 <b>Add Component</b> 对话框。选择 Type 为 <b>OPC Server</b> ，并点击 OK 确认此对话框和后面的对话框。	
4.	现在，OPC 服务器已集成到了组件组态器中。	

步骤	操作	备注
5.	现在添加一个 CP5611, 为此, 选择第二行并点击 <b>Add...</b> 弹出 <b>Add Component</b> 对话框, 选择 Type 为 <b>CP5611</b> 并点击 OK 确认此对话框和后面的对话框。	
6.	关闭 “Station Configuration Editor” 对话框。	

## 6.4 组态工程 PG/PC

### 设置 PG/PC 接口

表 6-5

序号	操作	提示/注释
1.	通过 <b>Start -&gt; SIMATIC -&gt; SIMATIC NET</b> 进入 <b>Configuration Console</b> 。 在 <b>Modules</b> 子菜单中选择您的 PG/PC 所安装的 MPI 卡(本例为 CP5611)。 在 <b>General</b> 选项卡上选择模式类型为 <b>Configuration Mode</b> 。	
2.	通过 <b>START -&gt; Settings -&gt; Control Panel</b> 调用 <b>Set PG/PC interface</b> 过程。在这里您可以将 S7ONLINE 访问点调整为您的 PC/PG 中的 PC internal 接口。	

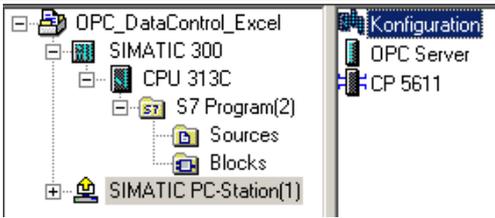
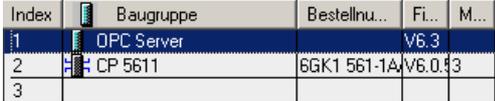
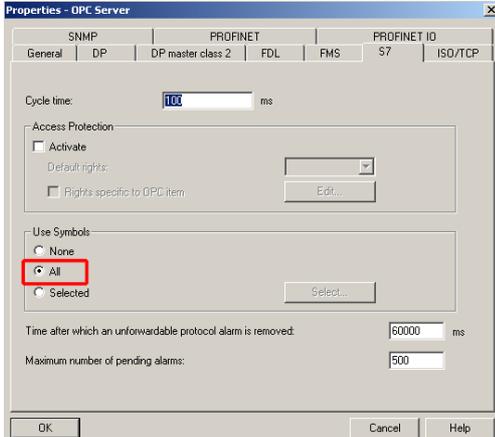
## 6.5 调试 SIMATIC PC 站点

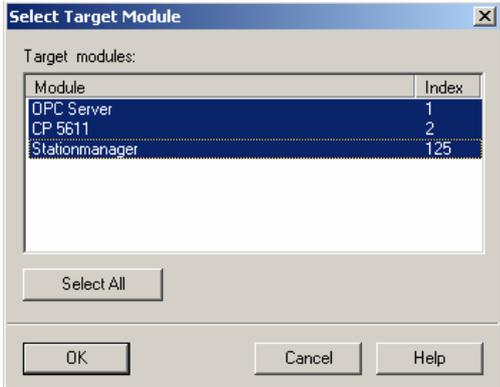
将文件 23829402 OPC\_DATCON\_EXCEL\_CODE\_v10.zip 解压缩到任意文件夹中。

该文件中包含：

- STEP7 项目 **opc\_data.zip**
- Excel 文件 **IceProduktion.xls**

表 6-6

步骤	操作	说明
1.	启动 SIMATIC 管理器	
2.	解压缩 <b>opc_data.zip</b> 项目	通过 <b>File -&gt; Retrieve</b>
3.	改变 SIMATIC PC 站点的组态	
4.	选择 OPC 服务器并双击打开 Properties 对话框。	
5.	切换到 <b>S7</b> 选项卡并在 User Symbols 框中选择 <b>All</b> 。点击 <b>OK</b> 关闭对话框。此操作将所有符号包含到 OPC 服务器的系统数据中。	
6.	编译您的组态并将其装载到您的 PC/PG。	

步骤	操作	说明
7.	在 <b>Select Target Module</b> 对话框中，选择 OPCServer、CP 和 Station manager 并点击 OK 确认。	
8.	此时，组态被保留在您的 PC/PG 上。	

## 6.6 调试 SIMATIC S7 站点

本节讲述 SIMATIC S7 站点的调试过程。

### 注意事项

与本范例程序一起提供的 STEP 7 项目中包含所有已组态的 SIMATIC S7 站点，并且站点中含有控制程序。如果所用硬件与组态完全相同，则使用项目时不要对其做任何更改。

表 6-7

序号.	步骤	备注
1.	选择 <b>SIMATIC 300</b> 站点并将站点装载到控制器。	

## 7 组态过程

此处您可了解关于...

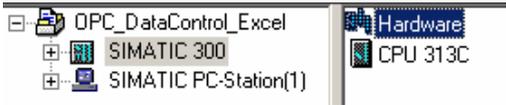
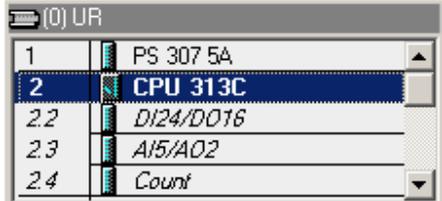
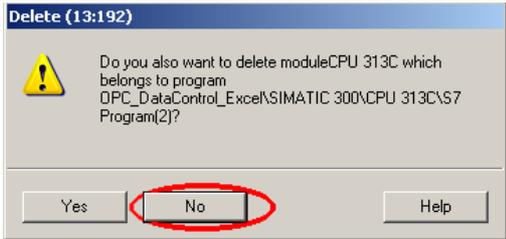
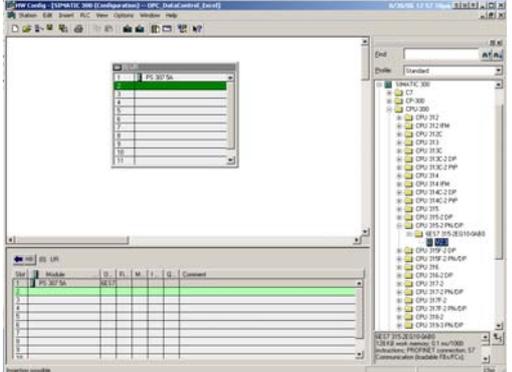
如果硬件与范例中所用的硬件有所不同时所必须的组态步骤。

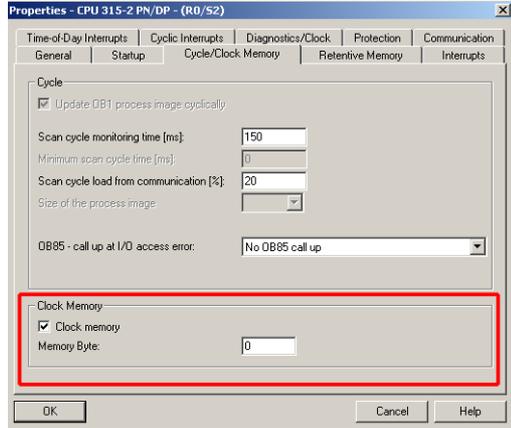
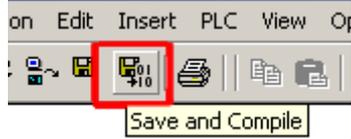
### 7.1 在 SIMATIC S7 站点中组态新的 CPU

通过 STEP 7 组态 SIMATIC S7 站点，以 CPU315-2PN 为例逐步说明组态过程。

#### 集成另一个 CPU

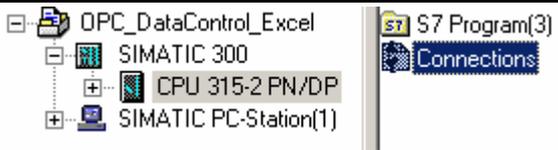
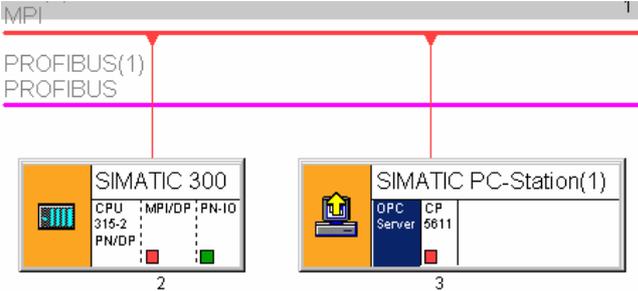
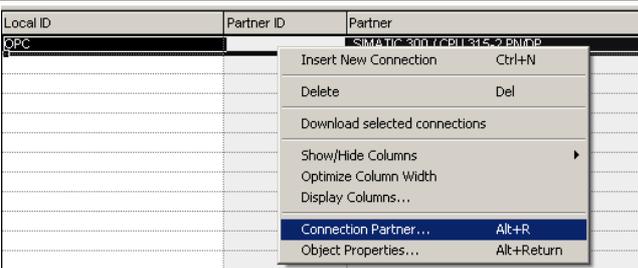
表 7-1

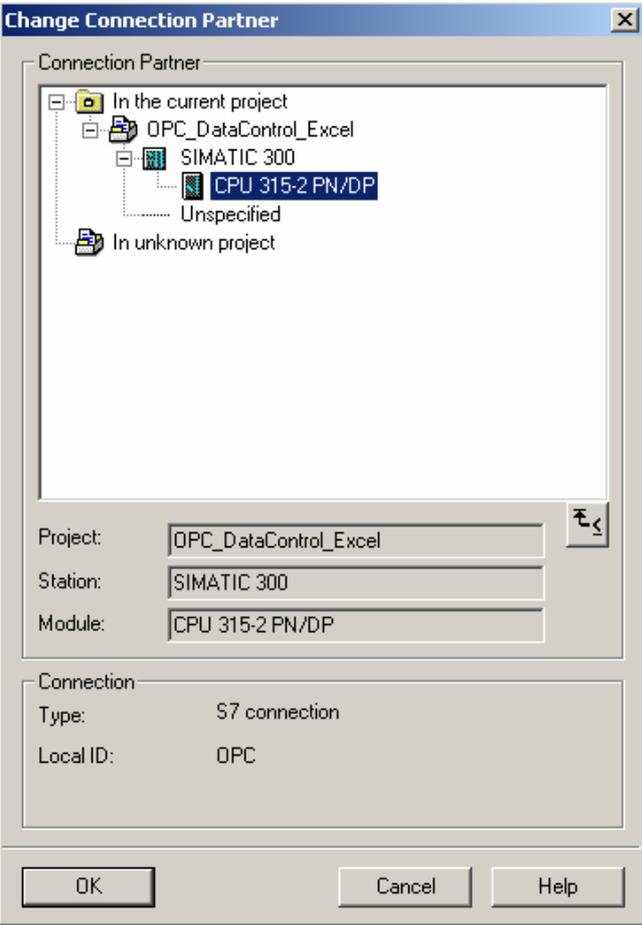
步骤	操作	备注
1.	启动 SIMATIC 管理器	
2.	解压缩 <b>opc_data.zip</b> 项目	通过 <b>File -&gt; Retrieve</b>
3.	在 <b>SIMATIC 300</b> 文件夹中双击 <b>Hardware</b> 打开硬件组态	
4.	在硬件组态中选择 <b>CPU313C</b> 并将其从项目中删除。 在随后的对话框中点击 <b>Yes</b> 确认删除操作。	
5.	<b>不要删除所附带的 S7 程序。点击 No</b> 退出对话框。	
6.	从硬件目录中选择您的 CPU 并通过拖放操作将其拖到 2 号插槽。	

步骤	操作	备注
7.	激活您 CPU 的 MPI 接口。	
8.	打开 CPU 的属性对话框并切换到 <b>Cycle/Clock Memory</b> 选项卡。 激活 <b>Memory Byte 0</b> 。 点击 OK 关闭对话框。	
9.	保存并编译(Save and compile)硬件组态。 关闭硬件组态。	

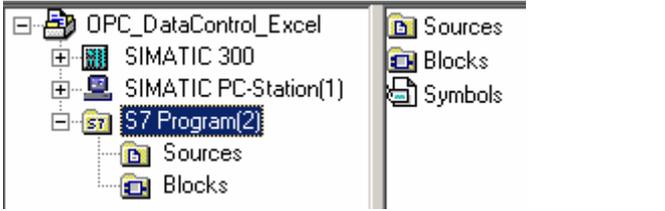
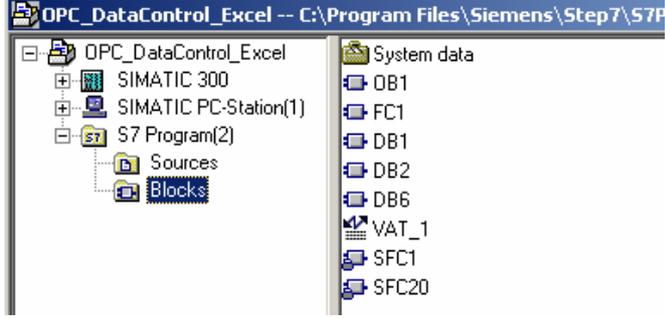
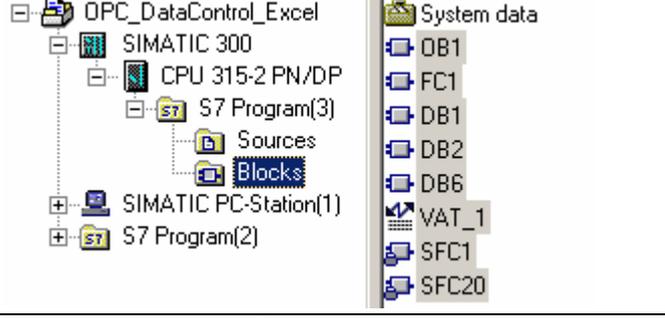
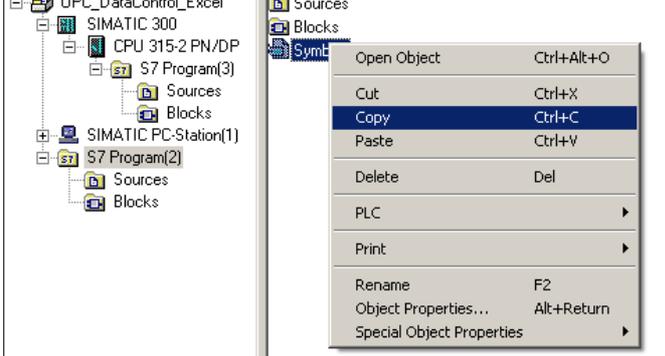
## 连接组态

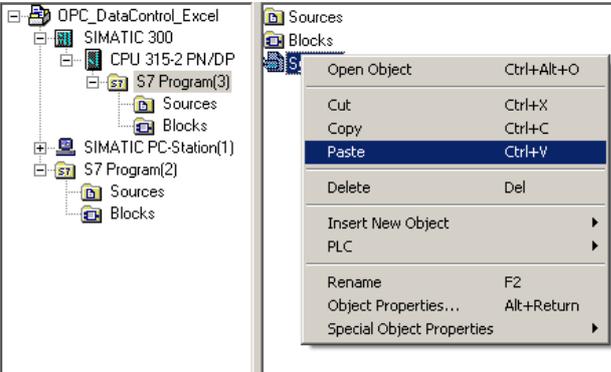
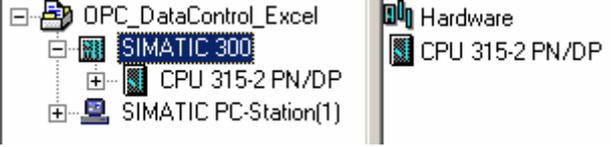
表 7-2

序号	操作	备注
1.	在 CPU 文件夹中双击 <b>Connections</b> 打开连接组态。	
2.	NetPro 打开, 在此选择 <b>OPC server</b> 。	
3.	选择 <b>OPC connection</b> , 右击连接并选择 <b>Connection Partner...</b> 。	

序号	操作	备注
3.	<p>选择您的 CPU 并点击 <b>OK</b> 确认对话框。</p> <p>OPC 连接中包含您添加的 CPU，将其作为一个连接伙伴。</p>	
4.	<p>保存并编译 NetPro 组态。 关闭 NetPro。</p>	

## 将各种块集成到新增 CPU 中

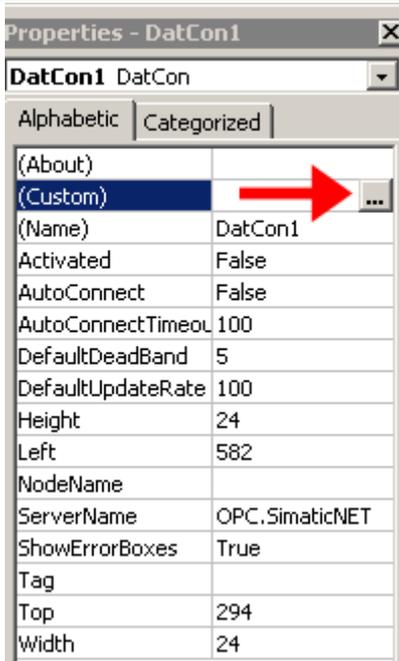
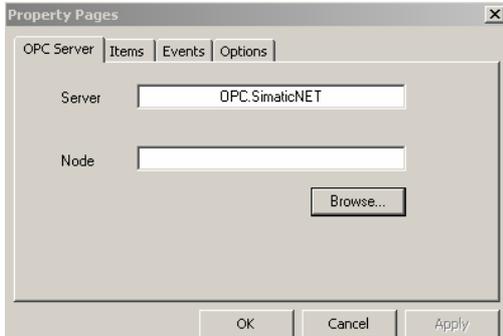
序号	操作	备注
1.	先前的 CPU 的各种块和符号保存在一个独立的文件夹中。	
2.	打开 <b>Blocks</b> 文件夹。	
	通过右击鼠标-> <b>Copy</b> , 选择所有的块(除系统数据之外) 切换到您新增 CPU 的块文件夹, 并通过右击鼠标-> <b>Paste</b> 将先前复制的块加入其中。	
3.	返回独立的程序文件夹, 选择符号文件并通过鼠标右击-> <b>Copy</b> 复制该文件。	

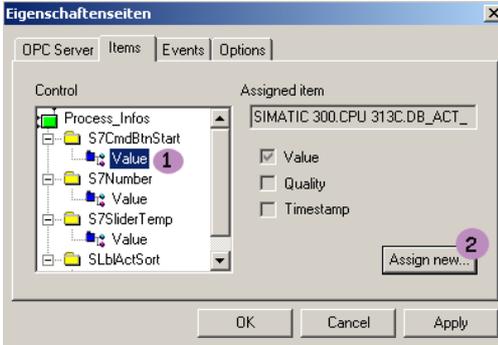
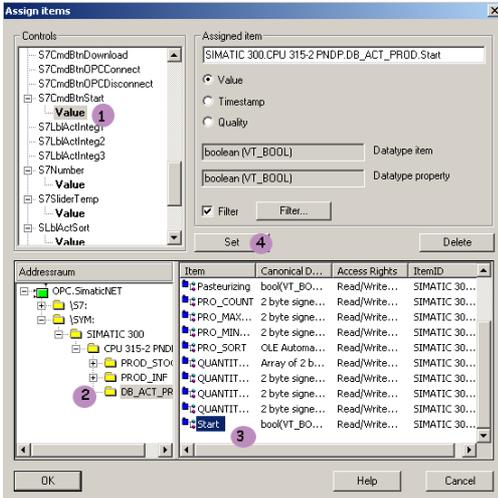
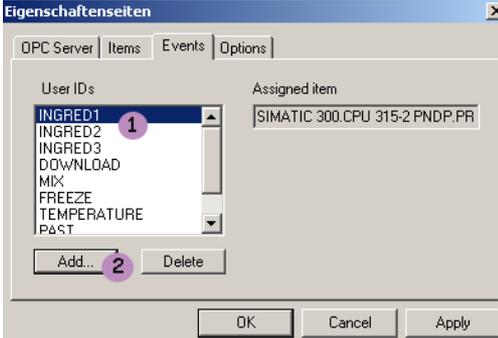
<p>4. 通过右击鼠标-&gt; <b>Paste</b> 将符号文件添加到您新增 CPU 的程序文件夹中。</p>	
<p>5. 删除从前 CPU 的程序文件夹。</p>	
<p>6. 选择 <b>SIMATIC 300</b> 站点并将该站点装入控制器。</p>	
<p>7. 选择 <b>SIMATIC PC</b> 站点并将其装入您的 PC。</p>	

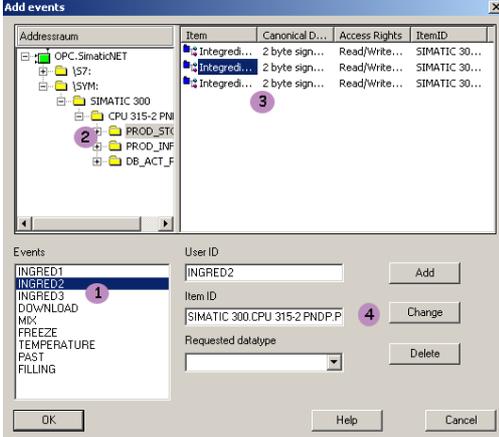
## 7.2 在 VBA 项目中更改 ItemIDS

对 ItemID 进行符号寻址时，CPU 名称将被用在地址区域中。如果 CPU 名称有变化，则必须相应地调整 ItemIDs。在 VBA 编辑器中进行这些更改操作。

表 7-3

步骤	操作	备注
1.	打开范例程序的 Excel 文件夹	
2.	在文件夹中打开 VBA 编辑器。	位于 <b>Tools -&gt; Macros -&gt; Visual Basic Editor</b>
3.	在 UserForm 中选择 OCX Data Control 图标。 右击 Properties 选项。	
4.	选择 <b>Custom</b> (用户定义)域，出现[...]按钮。 点击此按钮。	
5.	出现 OCX Data Control 的属性页面。 切换到 <b>Items</b> 选项卡。	

步骤	操作	备注
6.	在 <b>Items</b> 选项卡中定义显示控件的 ItemIDs。选择一个 <b>Value</b> (1) 并点击 <b>Assign new...</b> (2) 按钮。出现一个新的对话框。	
7.	选择显示控件的一个值(1)并在地址区域找到匹配过程变量的符号(2)。选择匹配变量(3)并点击 <b>Set</b> (4) 替换其符号 ItemID。这样就更改了所有值的符号 ItemID。	
8.	如果所有显示控件都分配到了新的符号地址，点击 <b>OK</b> 关闭对话框。	
9.	切换到 <b>Events</b> 选项卡。选择一个事件(1)。点击 <b>Add...</b> (2) 按钮打开一个新的对话框。	

步骤	操作	备注
10.	<p>选择一个事件名(1)并且在地址区域找到匹配的过程变量符号(2)。选择该变量(3)并点击 <b>Change</b> 按钮替换符号 ItemID。这样就改变了所有事件的符号 ItemID。</p>	
11.	<p>如果所有事件都分配到了新的符号地址，点击 <b>OK</b> 关闭对话框。</p>	
12.	<p>关闭 <b>OCX Data Control</b> 的属性对话框。</p>	<p>所有的符号 <b>ItemID</b> 都按照新的 CPU 做了调整。</p>
13.	<p>保存 VBA 项目。</p>	

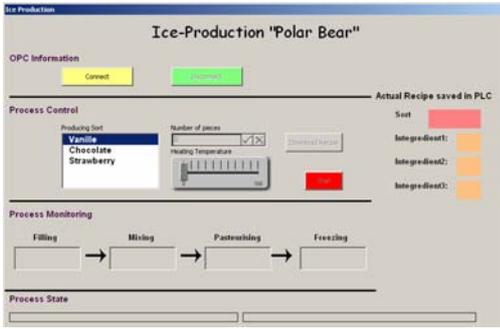
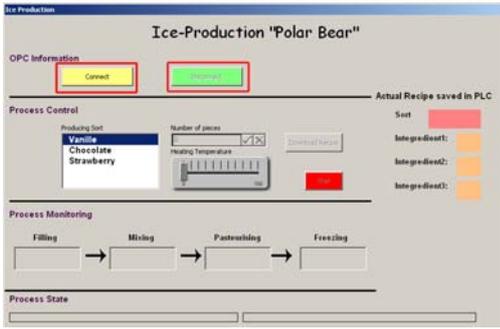
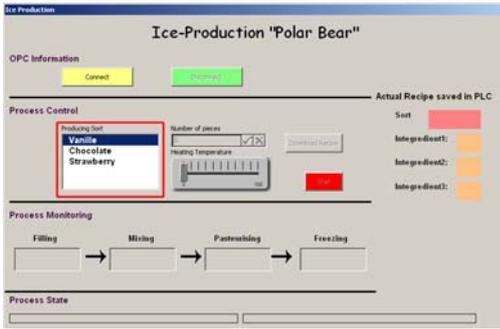
## 8 操作应用程序

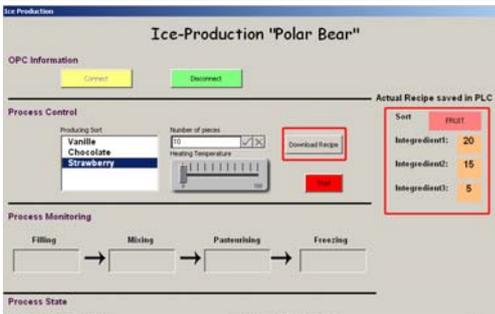
这里您将了解关于...

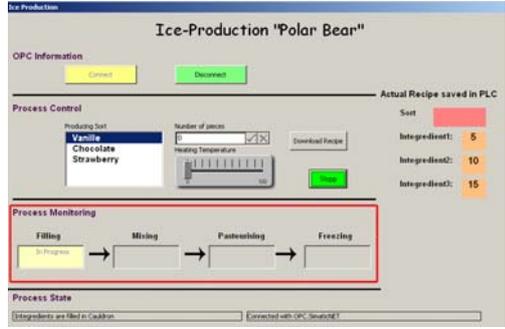
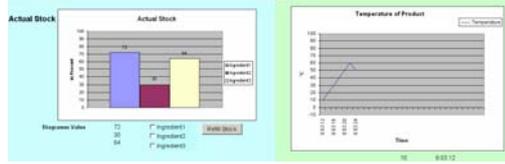
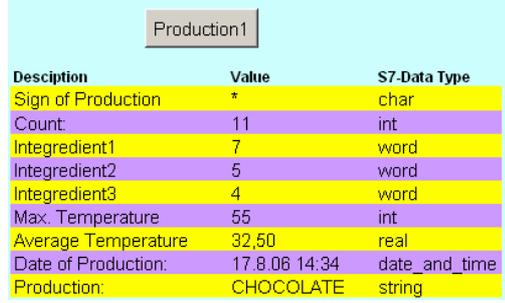
如何操作本范例应用程序中的所有功能。

制冰过程的操作

表 8-1

序号	步骤	备注
1.	打开 Excel 文件 <b>IceProduktion.xls</b> 。出现 Production 对话框。	
1.	<b>Connect</b> 按钮将您连接到 OPC 服务器。 点击 <b>Disconnect</b> 按钮断开连接。	
2.	在选择框内选择一种产品。	

序号	步骤	备注																														
3.	输入生产量并移动滑块改变最高生产温度。																															
4.	在 <b>Recipe</b> 工作表中，您可以改变每种产品的设定值。在这些行内输入新的数值。	<p><b>Indication of the quantity required for each recipe and ingredient in percent</b></p> <table border="1" data-bbox="858 853 1353 1458"> <tbody> <tr> <td colspan="2"><b>Vanille</b></td> <td></td> </tr> <tr> <td>Integredient1:</td> <td>9</td> <td rowspan="3"></td> </tr> <tr> <td>Integredient2:</td> <td>10</td> </tr> <tr> <td>Integredient3:</td> <td>5</td> </tr> <tr> <td colspan="2"><b>Chocolate</b></td> <td></td> </tr> <tr> <td>Integredient1:</td> <td>2</td> <td rowspan="3"></td> </tr> <tr> <td>Integredient2:</td> <td>4</td> </tr> <tr> <td>Integredient3:</td> <td>5</td> </tr> <tr> <td colspan="2"><b>Strawberry</b></td> <td></td> </tr> <tr> <td>Integredient1:</td> <td>30</td> <td rowspan="3"></td> </tr> <tr> <td>Integredient2:</td> <td>30</td> </tr> <tr> <td>Integredient3:</td> <td>30</td> </tr> </tbody> </table>	<b>Vanille</b>			Integredient1:	9		Integredient2:	10	Integredient3:	5	<b>Chocolate</b>			Integredient1:	2		Integredient2:	4	Integredient3:	5	<b>Strawberry</b>			Integredient1:	30		Integredient2:	30	Integredient3:	30
<b>Vanille</b>																																
Integredient1:	9																															
Integredient2:	10																															
Integredient3:	5																															
<b>Chocolate</b>																																
Integredient1:	2																															
Integredient2:	4																															
Integredient3:	5																															
<b>Strawberry</b>																																
Integredient1:	30																															
Integredient2:	30																															
Integredient3:	30																															
5.	<p>点击 <b>Download Recipe</b> 按钮将您的个性化生产信息装入 CPU。</p> <p>以上数值立即被显示为当前配方数值。</p>																															

序号	步骤	备注																														
6.	<p>点击 <b>Start</b> 按钮开始生产过程，此按钮位于 <b>Process</b> 工作表窗口中。</p> <p>在 <b>Process Monitoring</b> 中可监视生产过程。</p>																															
7.	<p>切换到 <b>Archive</b> 工作表。</p> <p>在这里您可看到以柱状图显示的当前库存以及生产温度曲线。</p>																															
8.	<p>在生产档案中您可通过点击按钮下载配方。</p>	 <table border="1"> <thead> <tr> <th>Description</th> <th>Value</th> <th>S7-Data Type</th> </tr> </thead> <tbody> <tr> <td>Sign of Production</td> <td>*</td> <td>char</td> </tr> <tr> <td>Count</td> <td>11</td> <td>int</td> </tr> <tr> <td>Integredient1</td> <td>7</td> <td>word</td> </tr> <tr> <td>Integredient2</td> <td>5</td> <td>word</td> </tr> <tr> <td>Integredient3</td> <td>4</td> <td>word</td> </tr> <tr> <td>Max. Temperature</td> <td>55</td> <td>int</td> </tr> <tr> <td>Average Temperature</td> <td>32,50</td> <td>real</td> </tr> <tr> <td>Date of Production:</td> <td>17.8.06 14:34</td> <td>date_and_time</td> </tr> <tr> <td>Production:</td> <td>CHOCOLATE</td> <td>string</td> </tr> </tbody> </table>	Description	Value	S7-Data Type	Sign of Production	*	char	Count	11	int	Integredient1	7	word	Integredient2	5	word	Integredient3	4	word	Max. Temperature	55	int	Average Temperature	32,50	real	Date of Production:	17.8.06 14:34	date_and_time	Production:	CHOCOLATE	string
Description	Value	S7-Data Type																														
Sign of Production	*	char																														
Count	11	int																														
Integredient1	7	word																														
Integredient2	5	word																														
Integredient3	4	word																														
Max. Temperature	55	int																														
Average Temperature	32,50	real																														
Date of Production:	17.8.06 14:34	date_and_time																														
Production:	CHOCOLATE	string																														

## 附录和参考文献列表

### 9 文献

#### 9.1 参考文献

本列表并未涵盖所有相关文献，而只是适当选择了一些资源。

图 9-1

序号	主题	文献名
/1/	STEP7	Automatisieren mit STEP7 in AWL und SCL [使用 STL 和 SCL 通过 STEP7 实现自动化] Hans Berger Publicis MCD Verlag ISBN 3-89578-113-4

#### 9.2 Internet 链接

本列表并未涵盖所有相关文献，而只是适当选择了一些资源。

序号	主题	链接/文献名
1	指向该文档的链接	<a href="http://support.automation.siemens.com/WW/view/en/23829402">http://support.automation.siemens.com/WW/view/en/23829402</a>
2	Siemens A&D 客户支持网站	<a href="http://www.ad.siemens.de/support">http://www.ad.siemens.de/support</a>
3	SIMATICNET 文档 CD-Rom 上的 OPC 客户接口、接口描述文档 以及 OPC 基金会网站的下载服务网页 <a href="http://www.opcfoundation.org">www.opcfoundation.org</a>	OPC Data Access 3.0 技术规范
4	有关下列内容的描述和信息： <ul style="list-style-type: none"> <li>• OPC 的一般信息</li> <li>• 使用 OPC server</li> </ul> 随 SIMATIC NET 安装已经提供，参见 Start → SIMATIC → Documentation → [Language] 产品支持网站上的条目 ID: <a href="#">2044387</a>	SIMATIC NET – 用 PG/PC 进行工业通信
5	关于创建 PC 站点的手册 随 SIMATIC NET 安装已经提供，参见 Start → SIMATIC → Documentation → [Language] 产品支持网站上的条目 ID: <a href="#">13542666</a>	SIMATIC NET 调试 PC 站点 – 手册/快速入门
6	关于 CP 5411、CP 5511 和 CP 5611 的修订后的安装指南 产品支持网站上的条目 ID: <a href="#">284796</a>	关于 CP 5411、CP 5511 和 CP 5611 的修订后的安装指南

7	CP 5611 手册/操作指南 产品支持网站上的条目 ID: <a href="#">13654902</a>	SIMATIC NET CP 5611 安装指南/产品信息
---	--	-------------------------------

### 9.3 其它 OPC 应用参考文献

序号	主题	描述
1	应用: 使用 Visual Basic .NET OPC 客户端通过 SEND/RECEIVE 协议进行 OPC 通信。 产品支持网站上的条目 ID: <a href="#">21523291</a> 。	使用 Visual Basic .NET OPC 客户端通过 SEND/RECEIVE 协议进行 OPC 通信。
2	应用: 使用 OPC Data eXchange (OPC DX)在不同的总线系统间交换数据。 产品支持网站上的条目 ID: <a href="#">21523569</a> 。	使用 OPC Data eXchange (OPC DX)在不同的总线系统间交换数据。
3	应用: 基于 PC 的自动化: 使用 OPC-XML 通过开放的接口连接数据库, 用 C# .net 编程。 产品支持网站上的条目 ID: <a href="#">21576581</a> 。	基于 PC 的自动化: 使用 OPC-XML 通过开放的接口连接数据库, 用 C# .net 编程。
4	应用: 通过 Visual Basic .NET 使用 SIMATIC NET OPC 服务器的 XML-DA 接口。 产品支持网站上的条目 ID:	通过 Visual Basic .NET 使用 SIMATIC NET OPC 服务器的 XML-DA 接口。
5	应用: 使用 SIMATIC NET OPC 服务器的 S7 块服务管理外部块。 产品支持网站上的条目 ID: <a href="#">21495347</a> 。	使用 SIMATIC NET OPC 服务器的 S7 块服务管理外部块。
6	应用: 通过基于 .NET 的 C#编程的 OPC 客户端获取批量数据。 产品支持网站上的条目 ID: <a href="#">21447513</a> 。	通过基于 .NET 的 C#编程的 OPC 客户端获取批量数据。
7	应用: 通过基于 Microsoft®.NET 和 C#编程语言的 OPC 实现单独可视化。 产品支持网站上的条目 ID: <a href="#">21043779</a> 。	通过基于 Microsoft®.NET 和 C#编程语言的 OPC 实现单独可视化。
8	应用: 通过 PC 进行控制—SIMATIC NET OPC 服务器作为 DP 主站。 产品支持网站上的条目 ID: <a href="#">21045282</a> 。	通过 PC 进行控制—SIMATIC NET OPC 服务器作为 DP 主站。
9	应用: 通过 PC 进行分散控制—SIMATIC NET OPC 服务器作为 DP 从站。 产品支持网站上的条目 ID: <a href="#">21040390</a> 。	通过 PC 进行分散控制—SIMATIC NET OPC 服务器作为 DP 从站。