Application for Communication

applications & TOOLS

Simple Plant visualization with OCX Data Control in VBA for Excel

Application

SIEMENS

Note The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are correctly used. These application examples do not relieve you of the responsibility to use sound practices in application, installation, operation and maintenance. In using these Application Examples, you recognize that Siemens cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

Warranty, Liability and Support

We accept no liability for information contained in this document.

Any claims against us - based on whatever legal reason - resulting from the use of the examples, information, programs, engineering and performance data etc., described in this application example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (Produkthaftungsgesetz), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (Wesentliche Vertragspflichten). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health. The above provisions do not imply a change in the burden of proof to your detriment.

Copyright© 2006 Siemens A&D. It is not permissible to transfer or copy these Application Examples or excerpts of them without first having prior authorization from Siemens A&D in writing.

For questions about this document please use the following e-mail address:

mailto:csweb@ad.siemens.de

Foreword

Objective of this application

The international OPC standard is an optimized interface for accessing the process data of a SIMATIC S7 station from a Windows application.

The application on hand shows a very simple and cost-effective process for generating individual visualization user interfaces with Microsoft Excel and VBA. In this application, a simple production plant is simulated with a SIMATIC S7 controller. The required process information are exchanged between the Excel application (OPC client) and the S7-CPU.

Main contents of this application

The following main points are discussed in this application:

- Visualization / programming
  - Generating a Microsoft Excel/ VBA application and the OPC ActiveX Controls of SIMATIC NET
  - Handling the OPC standard interface for OPC Data Access via OCX Data Control
  - Interconnecting the SIMATIC NET ActiveX Controls with the OCX Data Control
  - Interconnecting the Excel Standard Controls with S7 variables
- Configuration
  - Configuration of a PC station with OPC server and a connection via the MPI interface with the S7 CPU

Delimitation

This application does not contain

- Basics for Microsoft Excel
- Basics on the programming language Visual Basic for Application (VBA)
- Basics on LAD/FBD/STL

Previous knowledge in this field is assumed.

Structure of the document

The documentation of this application is divided into the following main parts.

Part	Description
Application Description	Provides a general overview of the contents. You will learn about the components used (standard hardware and software components and the specially created software).
Function Principles and Program Structures	This part describes the detailed function processes of the involved hardware and software components, the solution structures and – where useful – the specific implementation of this application. This part is necessary if you want to learn about the interaction of the solution components, for example in order to use them as the basis for own development.
Structure, Configuration and Operation of the Application	This part leads you step by step through the structure, important configuration steps, commissioning and operation of the application.
Appendix	This section of the documentation includes further information, e.g. literature, glossary etc.

Reference to Automation and Drives Service & Support

This entry originates from the internet application portal of the A&D Service and Support. Clicking the link below directly displays the download page of this document.

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

Table of Contents

Application Description

Content

Here an overview of this application is given. You will learn about the components used (standard hardware and software components and the specially created software).

1 Automation Task

Here you find information on ...

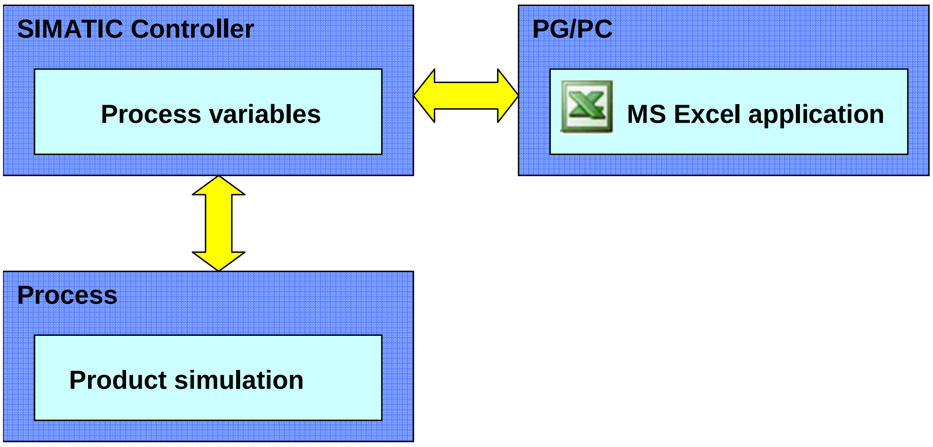
the automation task discussed in the documentation on hand.

1.1 Overview

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



Description of the Automation Task

A cheap connection to an S7 CPU is to be used to realize a simple individual visualisation user interface for a small production plant with MS Office means for the lower performance range.

Using an OPC Client programmed in VBA this application shows, how a production process can be visualized. Pressing elements on the user interface affects the behavior of the user program on the S7 CPU which simulates this process. On the other hand it is possible to read larger data volumes from the controller and to display them on visualization elements.

1.2 Requirements in detail

Requirements for the automation task

- The application has to be cost-efficient.
- The entire production process is to be visualized with a VBA user interface in Excel.
- The connection with the controller is performed via the MPI interface of the S7 CPU.
- The application is to show how the SIMATIC NET OPC server can be employed here by means of the OCX Data Control in VBA.
- Realizing a simple
  - operator user interface for controlling the plant
  - Recipe management
  - Display of temporal processes

Requirement for the visualization user interface

- The design of the operator user interface should be simple.
- Process variables in the S7 CPU are to be controlled and visualized via a graphical user interface.
- It is to be illustrated how S7 process data can be switched and displayed using
  - SIMATIC NET ActiveX Controls (without programming skills) and
  - Excel Standard Controls (individual programming)
- It is to be demonstrated how larger data volumes are written into the S7 CPU or read from it
- It is to be demonstrated how the most common S7 variable types are converted in VBA and displayed

Controller requirements

- Simulation of a technical process
- The communication is to occur without special communication blocks (variable services of the S7 communication)

2 Automation Solution

Here you find information on …

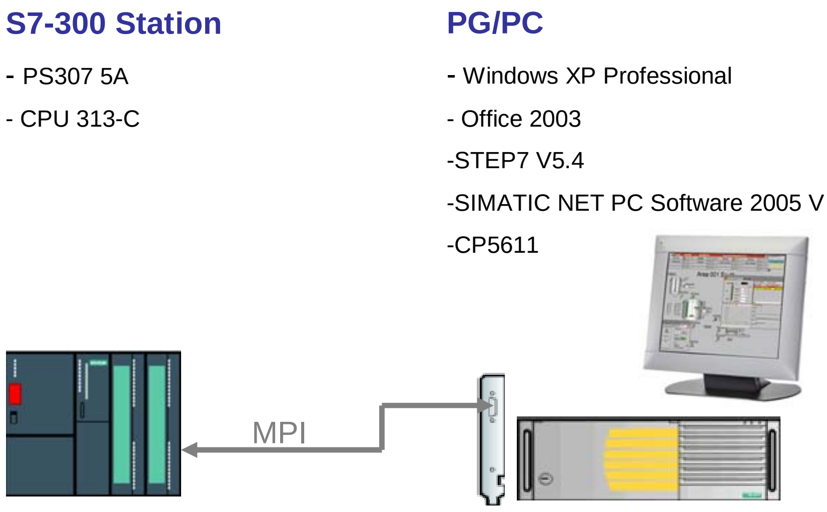
the solution selected for the automation task.

2.1 Overview of the overall solution

Display

The following figure displays the most important components of the solution:

Figure 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

Setup

The hardware of the automation solution consists of an S7 300 station and a PG/PC, which are inter-connected via the MPI:

- the PG/PC via the CP5611 A2
- the CPU via the integrated MPI interface

2.2 Description of the core functionality

Function scope of the example

The Excel project folder contains three worksheets which as an example simulate a simplified ice production.

The following table shows you how which worksheet fulfills which main functions.

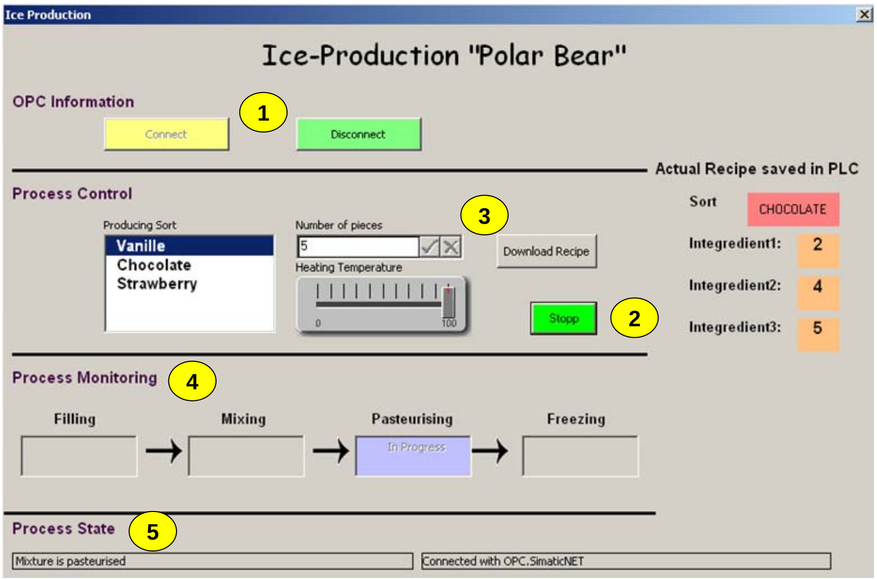
Table 2-1

Worksheet name	Function
Process	• UserForm with controls for the production • Status display of a sequence chart
Recipe	• Recipe management for the production
Archive	• Diagram records of a process sequence • Archived production data

Operator mask “Process“

The “Ice Production“ dialog is a UserForm included in VBA and is activated when activating the “Process“ Excel sheet. The OCX Data Control is integrated here.

Figure 2-2



Number of pieces: 5

Sort: CHOCOLATE

Integredient1: 2

Integredient2: 4

Integredient3: 5

Mixture is pasteurised

Connected with OPC.SimaticNET

Functionality of the user interface

The user interface of this dialog fulfills the following functionalities:

Table 2-2

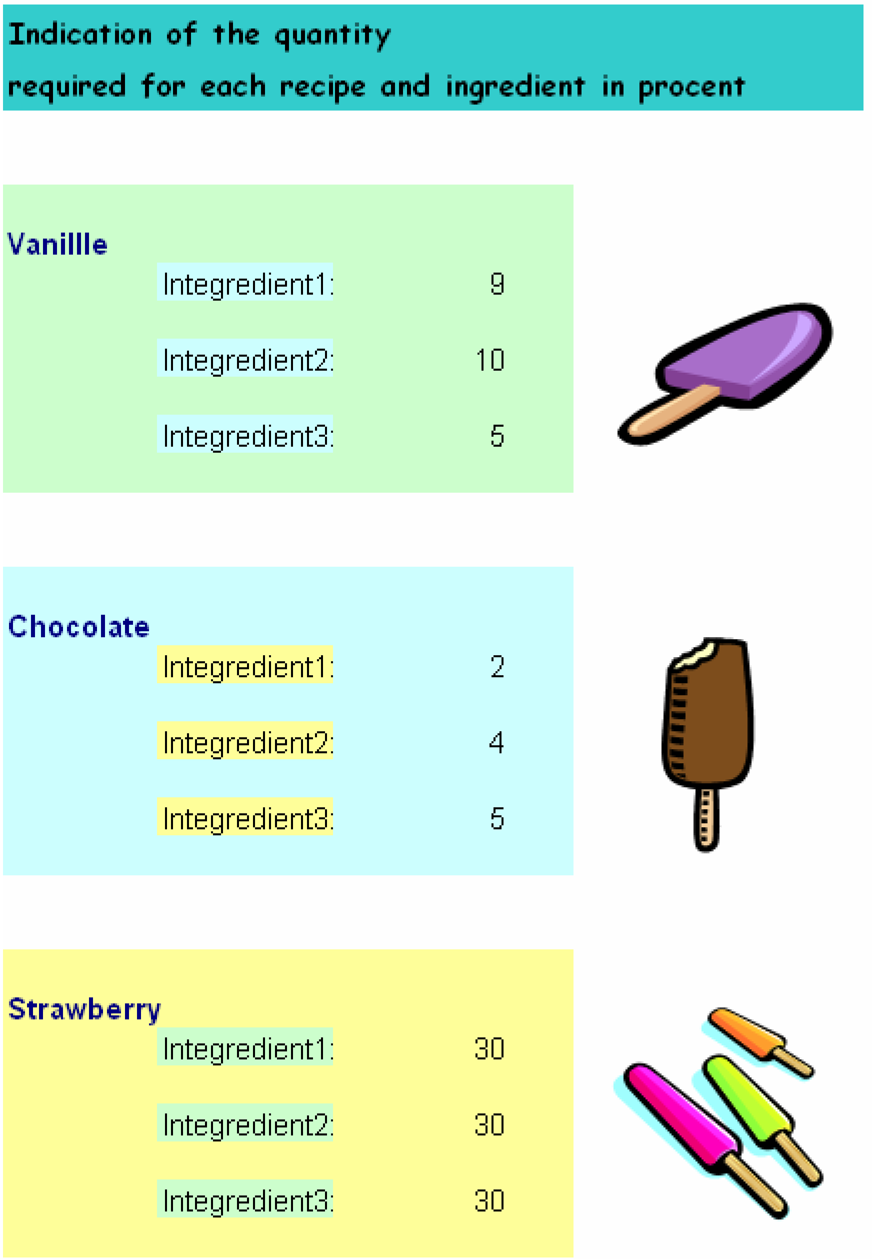
Operating element	Function
1.	Connecting and disconnecting the OPC server.
2.	Starting and shutdown of the controller.
3.	Selecting production type, item number and temperature and loading recipe data into the CPU
4.	Process monitoring
5.	Process status display

Operator mask “Recipe“

The user interface of this Excel worksheet displays the setpoint values for the ingredients of the individual production types. These values can be changed and adopted in the CPU when activating the Download Recipe button in the dialog.

The following figure shows the Excel worksheet recipe.

Figure 2-3



Indication of the quantity
required for each recipe and ingredient in procent

Vanillle

Integredient1:	9
Integredient2:	10
Integredient3:	5

Chocolate

Integredient1:	2
Integredient2:	4
Integredient3:	5

Strawberry

Integredient1:	30
Integredient2:	30
Integredient3:	30

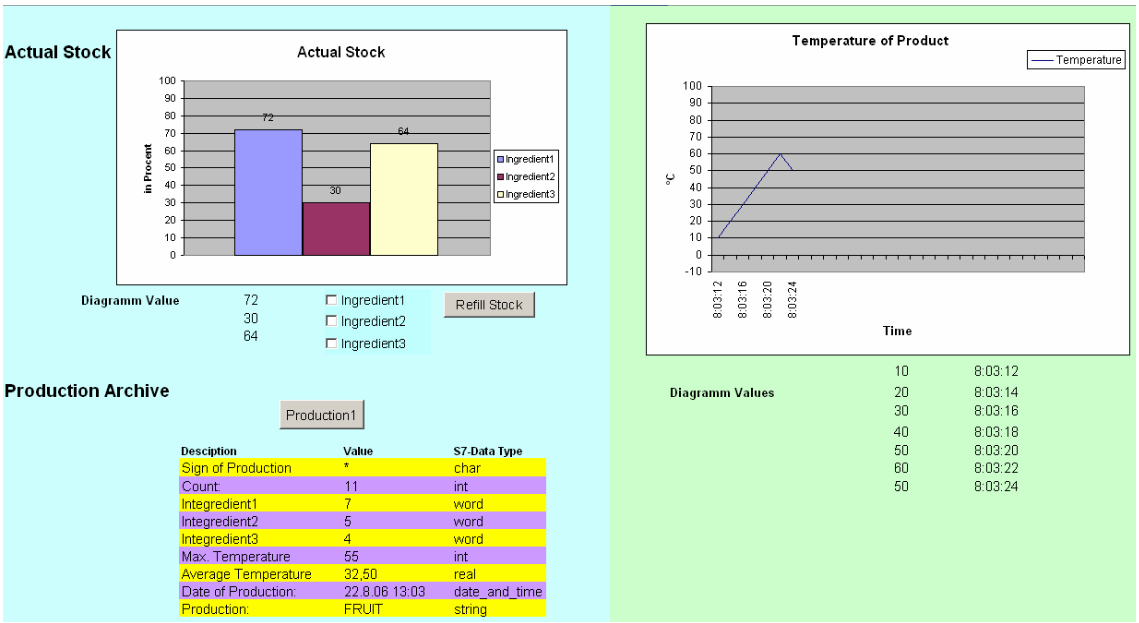
Operator mask “Archive“

The user interface of this worksheet fulfills three distinctive functionalities:

- **Actual Stock**: diagram display with current stock.
- **Production Archive**: recipe of last production
- **Recording of Temperature**: recording the temperature change during a production process in diagram format

The following screenshot shows the “Archive“ worksheet.

Figure 2-4



Actual Stock

Diagramm Value 72 30 64

□ Ingredient1
□ Ingredient2
□ Ingredient3

Refill Stock

Production Archive

Production1

Desciption	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:	22.8.06 13:03	date_and_time
Production:	FRUIT	string

Diagramm Values

10	8:03:12
20	8:03:14
30	8:03:16
40	8:03:18
50	8:03:20
60	8:03:22
50	8:03:24

Advantages of the application solution

- Cost effective MPI interface exists as a standard on every CPU; therefore no expensive CPU is required
- Minimal hardware setup
- No additional development environment necessary as the VBA editor is an integral part in Microsoft Office.

Advantages of OPC application

The application of the SIMATIC NET OPC server for plant visualization has the following advantages:

- Cost-effective, as OPC server contained in the delivery scope of the SIMATIC NET software.
- Simple configuration of the OPC server (as usual in STEP 7).
- Efficient data exchange from one process variable to an application for further processing.
- Maintenance and distribution by the OPC foundation.
- Data packages up to 65Kbytes.
- Using a simple script language for OPC programming.

2.3 Required hardware and software components

Hardware components

Table 2-3

Component	MLFB / Order number	Note
PS305 5A	6ES7 307-1EA00-0AA0	Or similar PS
CPU 313C	6ES7 313-5BE01-0AB0	As an alternative, a comparable CPU may also be used for this purpose.
MICRO MEMORY CARD 8 Mbyte	6ES7953-8LP11-0AA0	Or similar
Field PG or similar PC with PB/ MPI card	6ES7 711-xxxxx-xxxx	Configuration computer Link to PG configurator
CP5611 A2	6GK1561-1AA01	Optional: MPI card for connecting with a PC
MPI cable	6ES7 901-0BF00-0AA0	

Standard software components

Table 2-4

Component	MLFB / Order number	Note
SIMATIC S7 STEP7 V 5.4	6ES7810-4CC08-0YA5	
SIMATIC NET SOFTWARE EDITION 2005	6GK1704-0AA07-3AA0	OPC-Server, OCX Data Control
SIMATIC NET PB SOFTNET-S7/2005	6GK1704-5CW63-3AA0	Software + license
Microsoft Office 2003		Available at Microsoft

Example files and projects

The following list contains all files and projects used in this example.

Table 2-5

Component	Note
23829402_OPC_DATCON_EXCEL_CODE_v10.zip	This zip file contains the STEP 7 project and the EXCEL spreadsheet.
23829402_OPC_DATCON_EXCEL_DOKU_v10_d.pdf	This document

2.4 Alternative solutions

2.4.1 Alternative for OCX Data Control in Excel

Programming an OPC client using the automation interface of the SIMATIC NET OPC server.

Advantages:

- Easy programming by using the script languages VBA
- Fast generation of user interfaces directly in the Excel spreadsheet or as UserForm
- Integration of ActiveX components

Disadvantages:

- Limited performance due to script language
- Only average number of variables possible

2.4.2 Programming an OPC client in an other high-level language

C/C++/C# using the Custom interface

Advantages:

- High performance due to high-level language programming.
- High expressiveness and flexibility
- Large number of variables possible
- Parallel execution of several functionalities possible
- Web-application programmable with .NET language C#

Disadvantages:

- Long familiarization period required
- Higher expenses for generating complex user interfaces
- for C# additional Runtime Callable Wrapper required as intermediate layer for OPC server access.

Visual Basic V 6.0 using the automation interface

Advantages:

- Integration of ActiveX components
- Development of an application within short time possible due to short familiarization period.

Disadvantages:

- Unsatisfactory performance due Visual Basic Runtime and Wrapper as additional software layer
- Parallel execution of several functionalities not possible

2.4.3 Using a standard HMI system

Using WinCC or WinCC flexible as OPC client.

Advantages:

- Integrated functions for operator control & monitoring, reporting, acknowledging and archiving
- No programming necessary
- Quick generation of user interfaces
- Simultaneous access to different OPC servers possible

Disadvantages:

- High purchasing costs for WinCC and WinCC flexible software

Function Principles and Program Structures

Content

This part discusses the detailed function processes of the involved hardware and software components, the solution structures, and where sensible the concrete implementation of this application.

You only need this part if you want to learn about the interaction of the solution components.

3 General Function Mechanisms

Here you find information on …

the general functional mechanisms which apply for OPC.

3.1 Basics on the topic of OPC

What is OPC?

OPC is a software interface independent of manufacturer, based on COM/DCOM, which enables data exchange between hardware and software also from different manufacturers. The OPC Foundation, an interest group of well-known manufacturers, created these four OPC specifications.

Table 3-1

Specification	Application
OPC Data Access (DA)	Access to process data
OPC Alarm& Events (A&E)	Interface for event-based information including acknowledgement
OPC Historical Data Access (HDA)	Function for archived data
OPC Data eXchange (DX)	Server to server cross-communication.

This example uses exclusively the “OPC Data Access” specification.

OPC is a client/server architecture. Manufacturers for modules providing process data provide an OPC server with their modules which interfaces to the respective data source. An OPC client contacts the server and reads or writes the data.

OPC DA interface

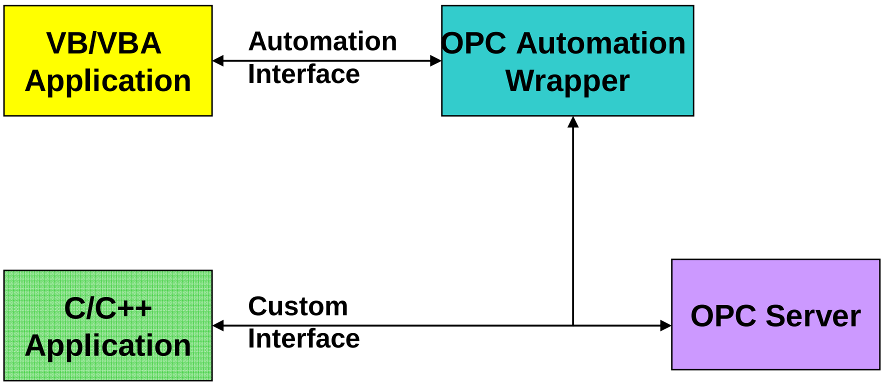
The OPC DA specification was the first OPC standard. It is used to exchange process data between controller and HMI devices or other clients. The read data contains data type, time stamp and quality, the statements on the quality of the value. The clients can read process data as well as write commands to the OPC server. The server then forwards the control data to the controller.

Automation and custom interface

OPC provides two interfaces for Data Access. The custom and automation interface enables the client to call the server functions. The interfaces are available for accessing process variables (Data Access) as well as for processing events and alarms.

The figure below illustrates the relation:

Figure 3-1

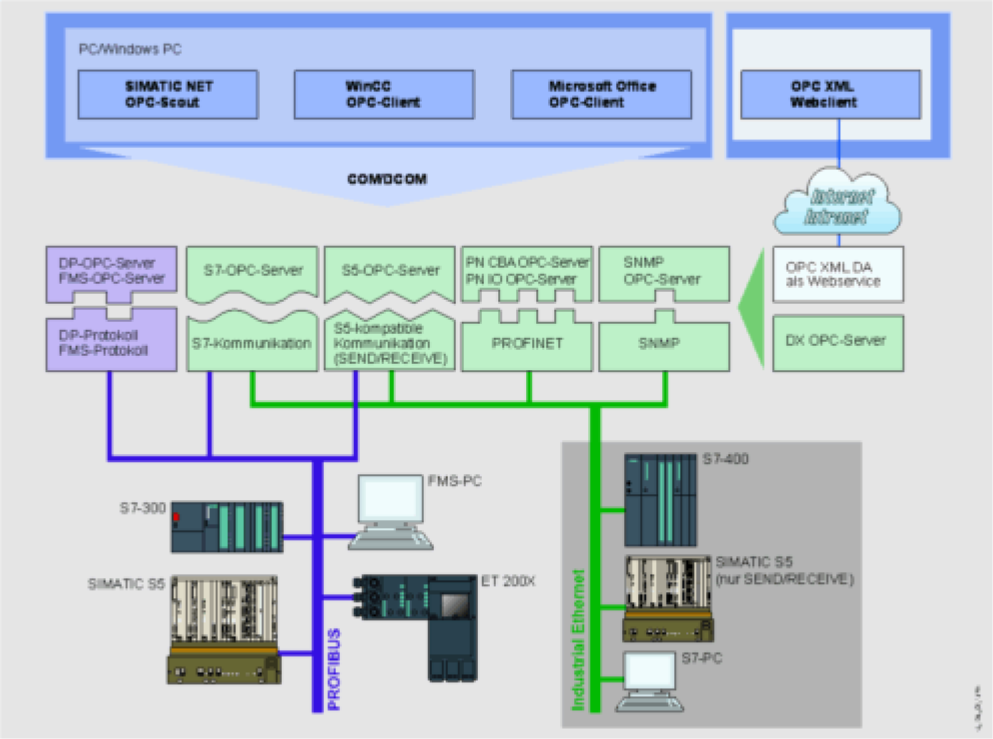


SIMATIC NET OPC server

The following graphic illustrates a structure with different SIMATIC NET OPC servers and the respective protocol drivers.

For this application the S7 OPC server was used. An S7 communication is established with an S7-300 via MPI.

Figure 3-2



The SIMATIC NET OPC server provides the following accesses:

- Industrial Ethernet
  - S7 communication
  - S5-compatible communication
  - PROFINET
  - SNMP
- PROFIBUS
  - DP protocol
  - FMS protocol
  - S7 communication
  - S5-compatible communication

3.2 Multi Point Interface

Each SIMATIC S7 CPU has an MPI (MultiPoint Interface). It enables setting up a subnet in the automation system where HMI and programming devices can exchange data.

Table 3-2

Criterion	Technical data
Number of stations	Max. 32
Transmission Rates	19.2 Kbit/s 187.5 Kbit/s 12 Mbit/s
Network expansion	Segment length 50 m
Transmission medium	Shielded two-wire line

4 Function Mechanisms of this Application

Here you find information on …

the functionalities offered by the OPC client, how OPC server and OPC client function mechanisms work together, and how this has been realized in a VBA project.

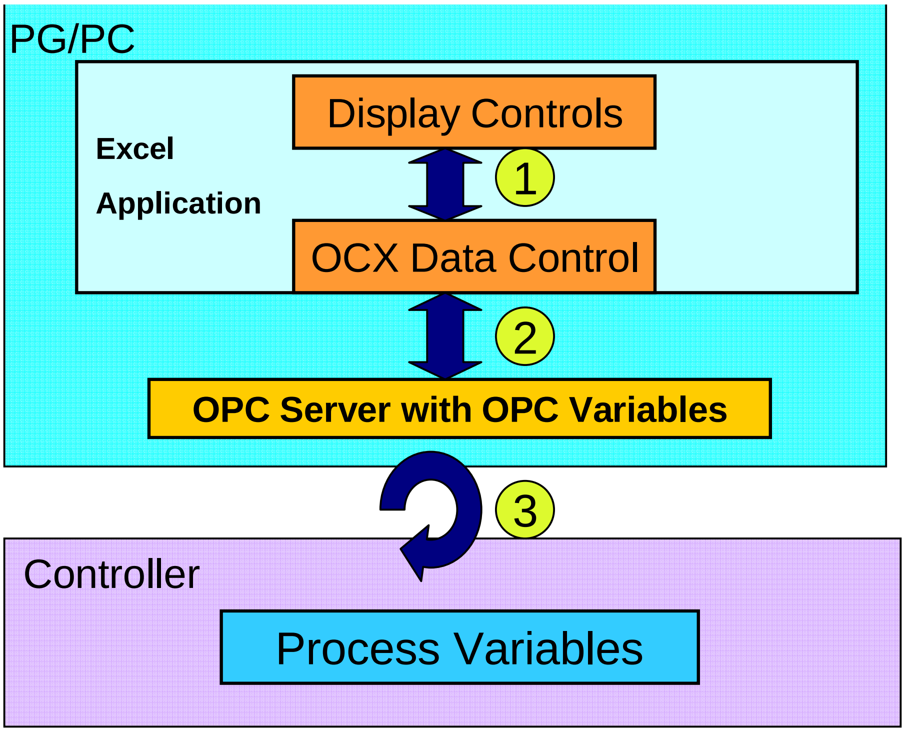
4.1 Function principles of OCX Data Control

The central component is the SIMATIC NET OCX Data Control. No other SIMATIC control can access process data without this data control. OCX Data Control establishes the connection with the SIMATIC NET OPC server and accesses process data. During program run this control is not visible.

Display

The following figure illustrates the connection between display controls and OPC process variables

Figure 4-1



The numbering has the following meaning:

Table 4-1

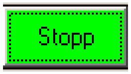
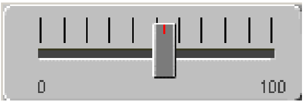
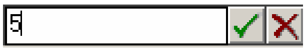
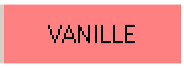
No.	Instructions
1.	The OCX Data Control provides the controls with data. Otherwise the controls give write commands to the OCX Data Control.
2.	The OPC server supplies the OCX Data Control with process values via the Data Access Interface.
3.	The OPC server continuously monitors the assigned process variables for changes and processes write jobs to process variables.

Available display controls of SIMATIC NET

The display controls are elements for visualizing process data. They receive their data via the SIMATIC NET OCX Data Control and not directly by accessing OPC or another interface. These controls enable generating an HMI user interface without programming.

The following table shows the existing display controls:

Table 4-2

Control	Display	Brief description
Button		Accesses individual bits stored in the OPC server. The button control only knows 2 values 0 = Start 1 = Stop When pressing the button, the respective bit also changes.
Slider		Accesses process data in byte, Word and double word format via variables. If the slider control in the value changes, the value of the assigned variable also changes
Number		Accesses process data in byte, Word and double word format via variables. If a new value is entered in Number control, the value of the assigned variable also changes
Label		Displays the value of a variable. The control label can only display values of process data

Note The display controls must be entered into a UserForm of Visual Basic.

Methods of OCX Data Control

The OCX Data Control has some methods to process information from the OPC server or process jobs. The most important methods are listed below.

Connect/ Disconnect

The Connect / Disconnect method enables manually connecting with the OPC server or establishing the connection. This method is required if the automatic connection with the OPC server is deactivated.

ValueChanged

The ValueChanged method is called if one or several events have been triggered by a value or quality change of the assigned variables. The event name and the new value or quality are transferred in the method call as array.

The Syntax of the ValueChanged method is:

```
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)
```

The parameters have the following meaning:

Table 4-3

Parameter	Meaning
Count	Number of triggered events
UserIDs	Event names
ItemIDs	ItemIDs of the assigned variables
Values	Changed values of the variables
Qualities	Qualities of the variables
TimeStamps	Time of the event

ReadVariable

The ReadVariable method reads the value of a process variable specified by the ItemID.

The Syntax of the method is:

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

The parameters have the following meaning:

Table 4-4

Parameter	Meaning
ItemID	ItemID of the variable to be read
Value	Values of the variables
Quality	Quality of the variables
TimeStamp	Time of read job

ReadMultiVariable

The ReadMultiVariable method reads the values from several process variables specified by the ItemID.

The Syntax of the method is:

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

The parameters have the following meaning:

Table 4-5

Parameter	Meaning
ItemIDs	String array with the ItemIDs of the variables to be read
Values	Variant array for the values
Errors	Variant array for the error status
Quality	Long array with the read qualities
TimeStamp	Date array with the read qualities

WriteVariable

The WriteVariable method writes a certain value to a process variable specified by the ItemID.

The Syntax of the method is:

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

The parameters have the following meaning:

Table 4-6

Parameter	Meaning
ItemID	ItemID of the variables to be described
Value	New value

WriteMultiVariable

The WriteMultiVariable method writes certain values into several process variables specified by the ItemID.

The Syntax of the method is:

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

The parameters have the following meaning:

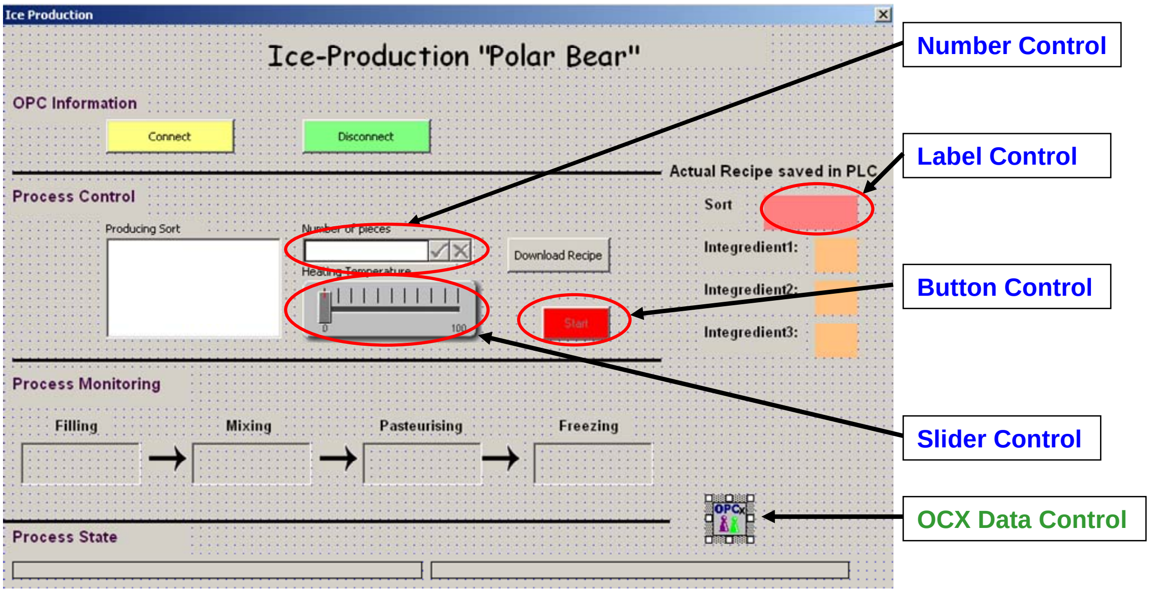
Table 4-7

Parameter	Meaning
ItemIDs	String array with the ItemIDs of the variables to be described
Values	Variant array with the new values
Errors	Variant array for the error status

The ActiveX Controls used in the user interface

The following figure shows the user interface of the dialog in the development mode. It contains all display controls of SIMATIC NET as well as the data control, presented in color.

Figure 4-2



The OPC process variables

The process variables are represented in the OPC server by an OPC class. The so-called ItemID is a string sequence and identifies the process variable uniquely. It informs the server of which process variable is allocated to the OPC item. The process variable can then be accessed via the OPC item. The ItemID can thereby identify the process variable absolutely or symbolically.

Absolute addressing

For the ItemID with an absolute addressing there are three options:

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

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

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

The meaning of the individual address areas is listed below:

Table 4-8

Address areas	Meaning	Note
S7	S7 protocol for accessing the process variables	
<Connection name>	Name of the connection.	Configurable in NetPro
DB	Data block; identification for a variable in a DB	
DI	Instance data block; identification for a variable in a DI	
<Number>	Number of the DB or DI	
<Type>	S7 data type	An S7 data type is in the OPC server transformed into the corresponding OLE (COM) data type
<Object>	S7 data type	e.g. memory bit M, input I, output O
<Address>	Address of the first variable to be addressed.	-Byte-Offset -Byte-Offset.Bit (only data type X) -Byte-Offset.String length (only data type string, string length 1 byte to 254 bytes)
<Quantity>	Number of variables of a type to be addressed from the offset on which is specified in the *Adresse* parameter (value range 0...65535).	For data type **X**, input of the number for the write access is only possible in multiples of 8. The bid address must then be zero.

Symbolic addressing

If the process variables are declared as symbols in the controller, then the ItemID can also be symbolic.

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

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

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

The meaning of the address areas is listed below:

Table 4-9

Address areas	Meaning	Note
<Station Name>	Name of the SIMATIC Station	Configurable in STEP7
<CPU Name>	Name of the CPU	Configurable in STEP7
<DB Name>	Symbolic name of the data block	Configurable in STEP7
<DI Name>	Symbolic name of the instance data block	Configurable in STEP7
<Variable Name>	Symbolic name for the variables	Configurable in STEP7

Note For the OPC client to be able to transmit symbolic variable accesses via the OPC server, the symbols configured in STEP 7 must be accepted in the system data for the OPC server. (See Chapter.6)

ItemIDs of this application

The following table lists the ItemIDs symbolically used in this application.

Table 4-10

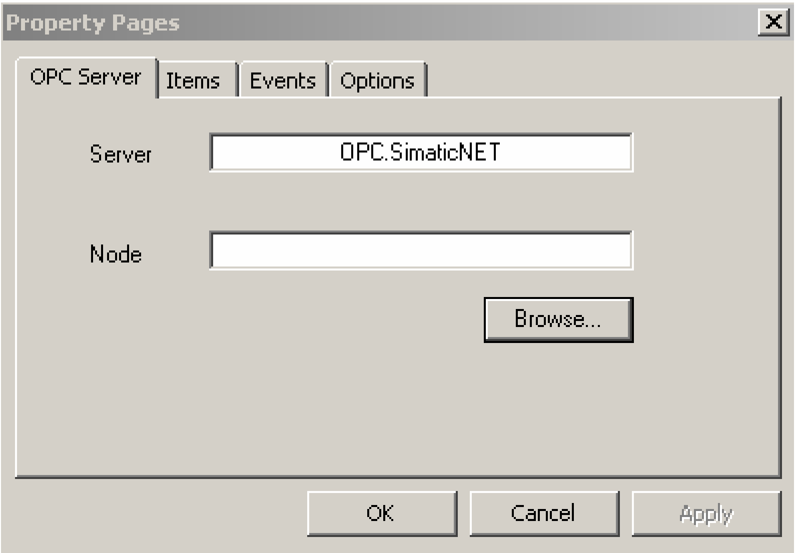
ItemID symbolic	ItemID absolute
SIMATIC 300.CPU 313C.PROD_RECIPE.Ingredient1	S7:[OPC]DB6,INT0
SIMATIC 300.CPU 313C.PROD_RECIPE.Ingredient2	S7:[OPC]DB6,INT2
SIMATIC 300.CPU 313C.PROD_RECIPE.Ingredient3	S7:[OPC]DB6,INT4
SIMATIC 300.CPU 313C.DB_ACT_PROD.PRO_SORT	S7:[OPC]DB1,STRING0,30
SIMATIC 300.CPU 313C.DB_ACT_PROD.PRO_COUNT	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

Note More information on the OPC process variables are available in the “Industrial communication with PG/PC Bd2 interface“ manual under SIMATIC -> Documentation.

4.2 Exemplary configuration of the OCX Data Control

The OCX Data Control is the most important component of the OPC client in this application. For the configuration, the OCX Data Control has a Properties window with several tabs.

Figure 4-3



Server: OPC.SimaticNET

Node:

This section gives an example of how to interconnect a variable with an event.

Table 4-11

Step	Action	Dialog tab
1.	Select OPC server	OPC server
2.	Connect display controls with process variables	Items
3.	Assign process variables with events	Events
4.	Make settings	Options

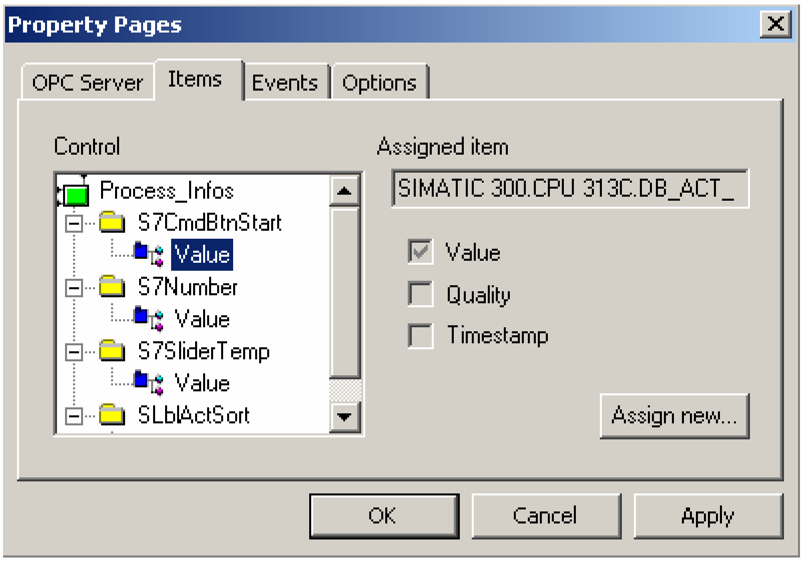
Select OPC server

As an OPC client, the SIMATIC NET OCX Data Control can use any OPC server as an OPC client. The SIMATIC NET OPC server is called **OPC.SimaticNET.** The OPC server is configured by OCX Data Control is such a way that all assigned variables can be continuously monitored and for a changed value of a process variable the new value is transferred to the data control and from there to the respective display control.

Connecting the display controls with the process variables

The process variables are accessed via the data control. Via the OCX Data Control the display controls are connected with a property (Value, Quality, Time Stamp) of a process variable of the OPC server.

Figure 4-4



Assigned item: SIMATIC 300.CPU 313C.DB_ACT_

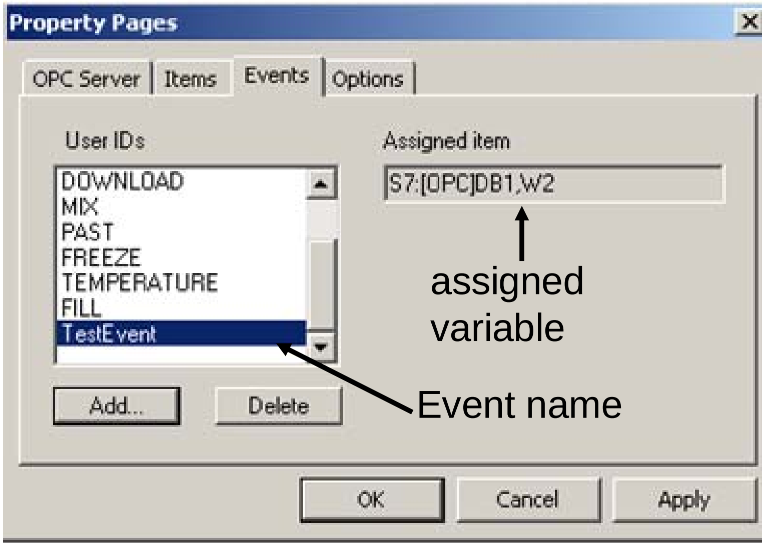
☑ Value
□ Quality
□ Timestamp

Connecting process variables with events

Frequently, the display controls are insufficient for generating a more individual and complex user interface. Apart from the display controls there is the option of connecting process variables with the OCX Data Control in form of events. If the status or value of the variable changes, the respective event is triggered and the VB program can react accordingly.

The following figure shows an event with the assigned variables. As soon as the value of the logically connected variable changes, an even named “TestEvent“ is triggered.

Figure 4-5



User IDs:
- DOWNLOAD
- MIX
- PAST
- FREEZE
- TEMPERATURE
- FILL
- TestEvent

Assigned item: S7:[OPC]DB1,W2

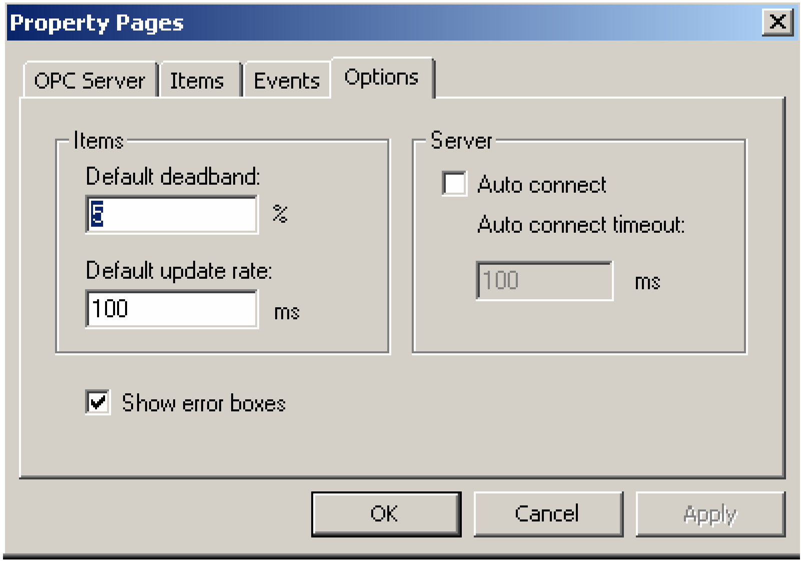
assigned variable

Event name

Make settings

In the Options tab default values can be set for the OPC connection. This includes update time of the variables, dead time of the connection but also switching off the automatic connection with the OPC server when calling the data control.

Figure 4-6



Items

Default deadband: 5 %

Default update rate: 100 ms

Server

☐ Auto connect

Auto connect timeout: 100 ms

☑ Show error boxes

4.3 STEP7 example program

The following table gives an overview of the STEP7 blocks for simulating the ice production.

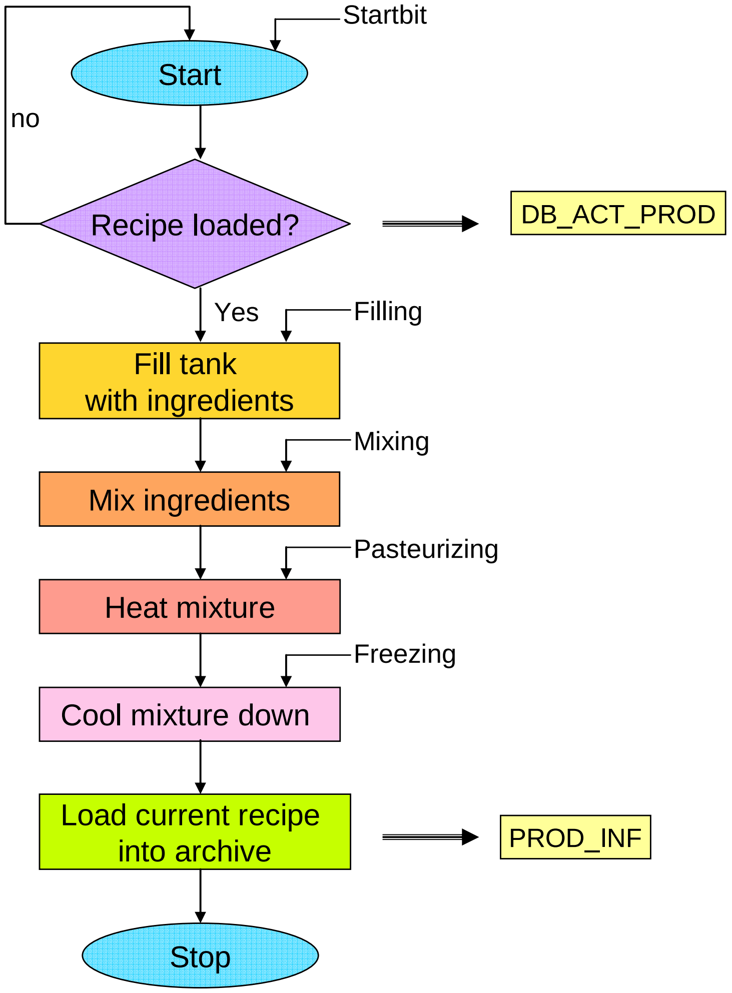
Table 4-12

Block	Function
FC 1	The “PRO_CONTROL“ function simulates the ice production in a step chain.
DB 1	The “DB_ACT_PROD“ data block contains the current recipe data and control variables
DB 2	The data block “PRO_INF“ contains the production information of the production process
DB 6	The data block “PROD_STOCK“ contains the actual values of the stock

Simulation program run

The following graphic shows the flow chart of the simulation program. The simulation of the production is stored in the FC1 “PRO_CONTROL“, which is cyclically called by OB1. FC1 is realized as step chain.

Figure 4-7



5 Explanations for the Example Program

Here you find information on …

Details from the code of some core program parts of the VBA program, how the error routine was realized in the OPC client and information on running the STEP7 program.

5.1 Error handling in the OPC client

This section shows how the VBA program reacts to certain errors. The following errors are evaluated:

- during connecting and disconnecting
- during reading of process variables from the controller
- during writing of values to the controller

Error handling in the user interface:

- The user interface contains an output window for error messages.
- An error during the connecting process with the OPC server calls the ConnectionError method.
- The returned values of all OPC calls are checked for errors.
- In case of an error a message is output in the output window.
- Many functions have an ErrorHandler, which in case of an error outputs a message.

5.2 Explanations on the VBA example program

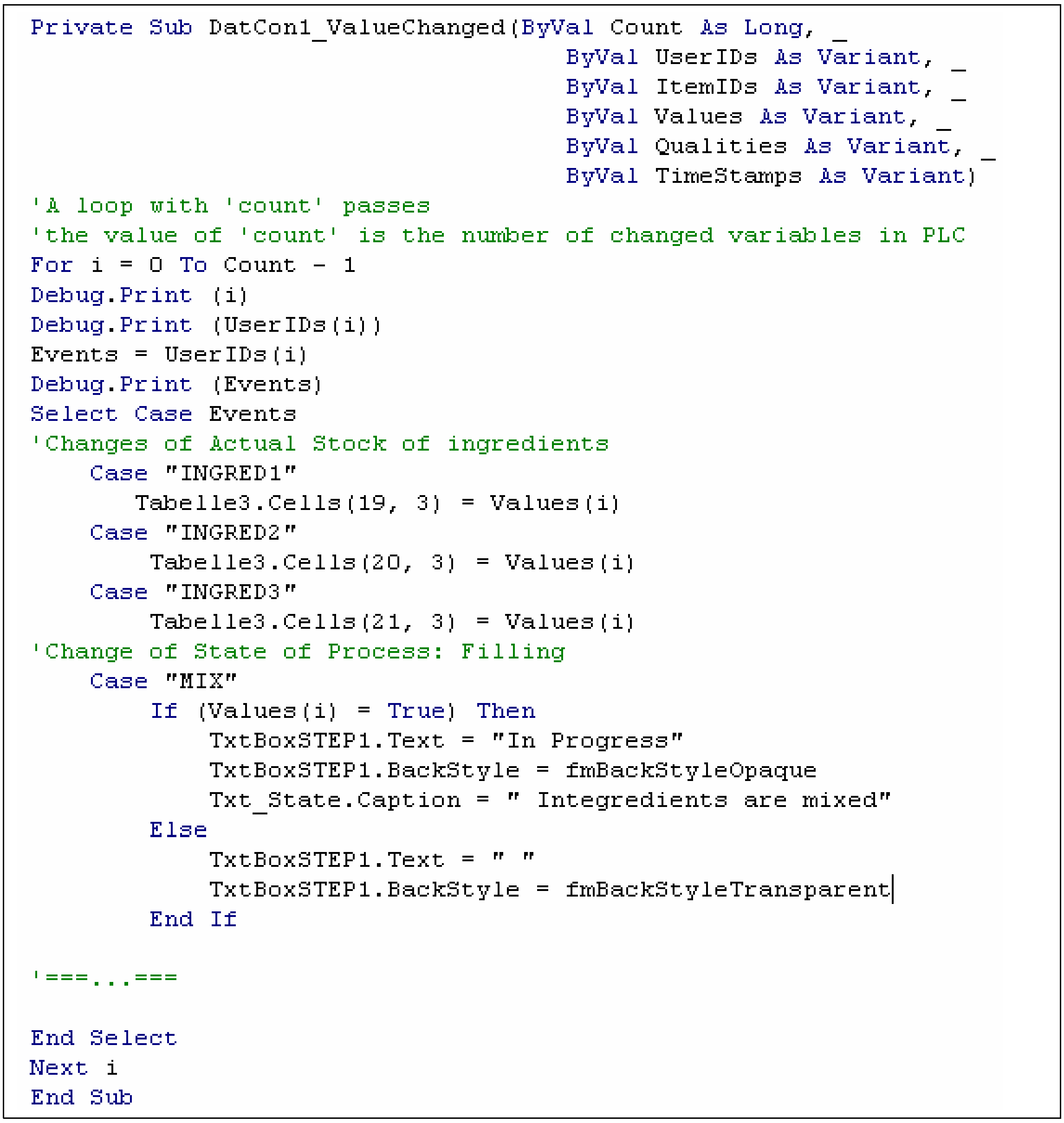
This chapter gives you further details on the Visual Basic Code. Using code extracts, the following core program parts are explained in greater detail:

- Event analysis with the OCX Data Control
- Reading block data
- Converting S7 data types into VBA types

ValueChanged method

The ValueChanged method of the OCX Data Controls is called if one or several events were triggered. The method memorizes the event names so that the VB program can react to the respective events. The following graphic shows an extract of the ValueChange method of this application.

Figure 5-1

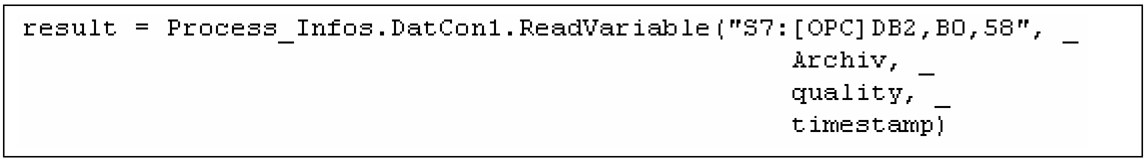


```
Private Sub DatCon1_ValueChanged(ByVal Count As Long, _
                                    ByVal UserIDs As Variant, _
                                    ByVal ItemIDs As Variant, _
                                    ByVal Values As Variant, _
                                    ByVal Qualities 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
            TxtBoxSTEP1.Text = "In Progress"
            TxtBoxSTEP1.BackStyle = fmBackStyleOpaque
            Txt_State.Caption = " Integredients are mixed"
        Else
            TxtBoxSTEP1.Text = " "
            TxtBoxSTEP1.BackStyle = fmBackStyleTransparent
        End If

'===...===

End Select
Next i
End Sub
```

Reading of larger data volumes

For the load reduction of the OPC server there is the option of reading data with different types en block from the CPU. The ReadVariable method enables reading data in a buffer using a fixed format. The following graphic shows the call for reading a data block of bytes.

Figure 5-2



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

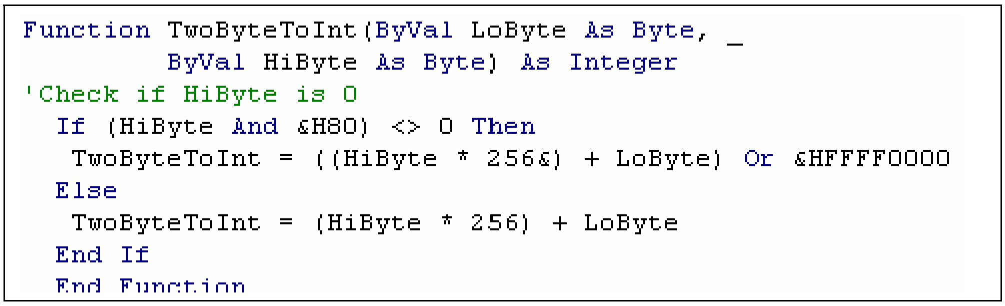
Data conversion routines in VBA

In this section we show you how to split the byte data block, which was read en bloc, into the original types. The data conversion regulations from S7 data types into the respective VBA types must be observed.

Bytes ←→VBA Integers

The following graphic shows the conversion into a VBA Integer. The read bytes represent an integer in the CPU.

Figure 5-3

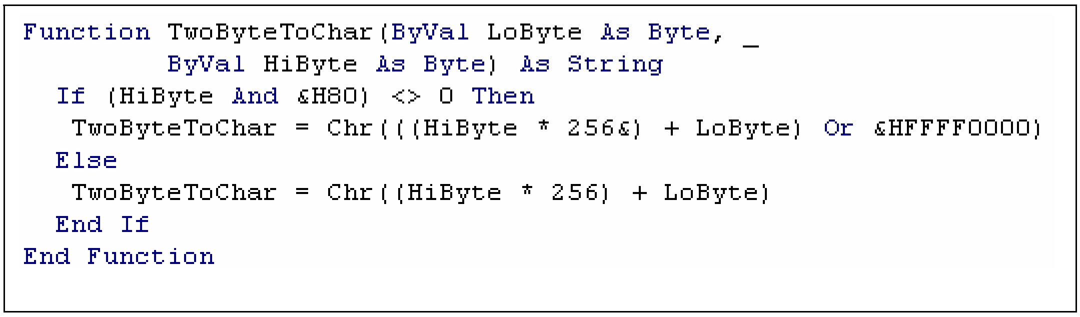


```
Function TwoByteToInt(ByVal LoByte As Byte, _
         ByVal HiByte As Byte) As Integer
'Check if HiByte is 0
  If (HiByte And &H80) <> 0 Then
   TwoByteToInt = ((HiByte * 256&) + LoByte) Or &HFFFF0000
  Else
   TwoByteToInt = (HiByte * 256) + LoByte
  End If
  End Function
```

Bytes ←→VBA characters

The following graphic shows the conversion into a VBA character. The read bytes represent a Char in the CPU.

Figure 5-4

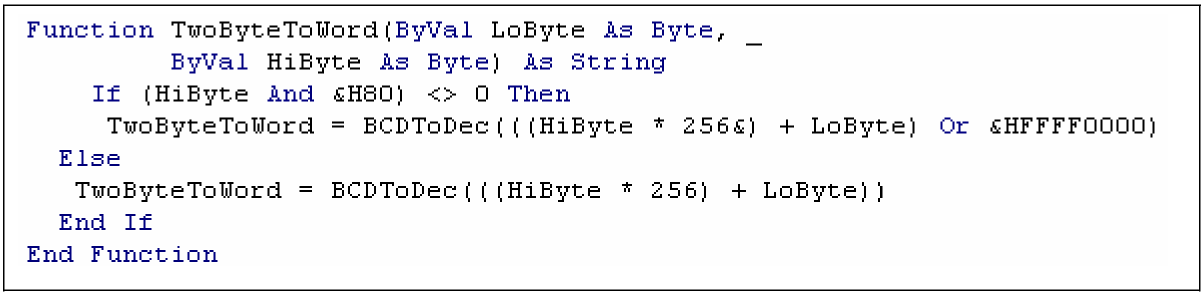


```
Function TwoByteToChar(ByVal LoByte As Byte, _
         ByVal HiByte As Byte) As String
  If (HiByte And &H80) <> 0 Then
   TwoByteToChar = Chr(((HiByte * 256&) + LoByte) Or &HFFFF0000)
  Else
   TwoByteToChar = Chr((HiByte * 256) + LoByte)
  End If
End Function
```

Bytes ←→VBA Word

The following graphic shows the conversion into a VBA word. The read bytes represent a word in the CPU.

Figure 5-5

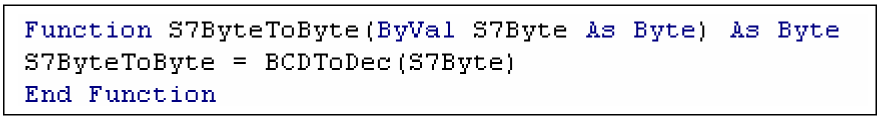


```
Function TwoByteToWord(ByVal LoByte As Byte, _
         ByVal HiByte As Byte) As String
    If (HiByte And &H80) <> 0 Then
     TwoByteToWord = BCDToDec(((HiByte * 256&) + LoByte) Or &HFFFF0000)
  Else
   TwoByteToWord = BCDToDec(((HiByte * 256) + LoByte))
  End If
End Function
```

Byte ←→VBA byte

The following graphic shows the conversion into a VBA byte. The read bytes represent a byte in the CPU.

Figure 5-6

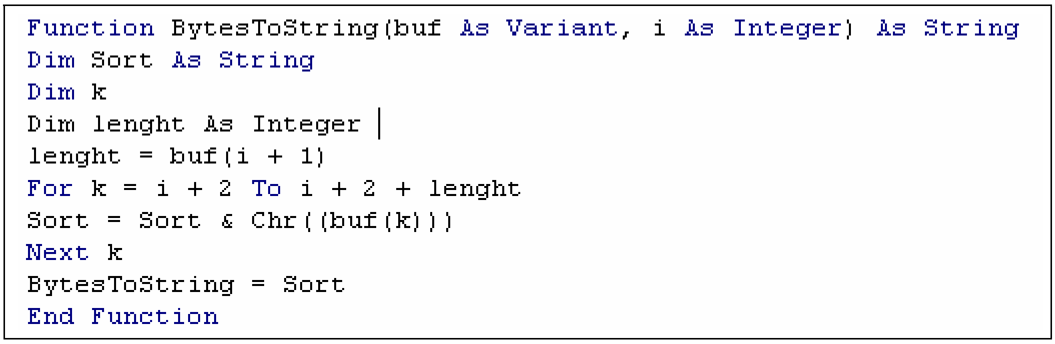


```
Function S7ByteToByte(ByVal S7Byte As Byte) As Byte
S7ByteToByte = BCDToDec(S7Byte)
End Function
```

Bytes ←→VBA string

The following graphic shows the conversion into a VBA string. The read bytes represent a string in the CPU.

Figure 5-7

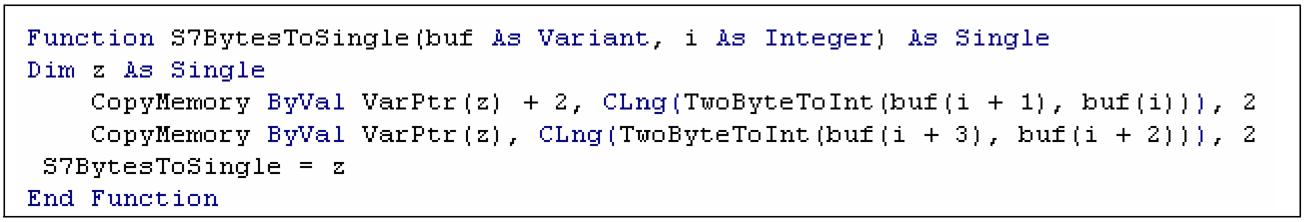


```
Function BytesToString(buf As Variant, i As Integer) As String
Dim Sort As String
Dim k
Dim lenght As Integer
lenght = buf(i + 1)
For k = i + 2 To i + 2 + lenght
Sort = Sort & Chr((buf(k)))
Next k
BytesToString = Sort
End Function
```

Bytes ←→VBA single

The following graphic shows the conversion into a VBA single. The read bytes represent a Real in the CPU.

Figure 5-8

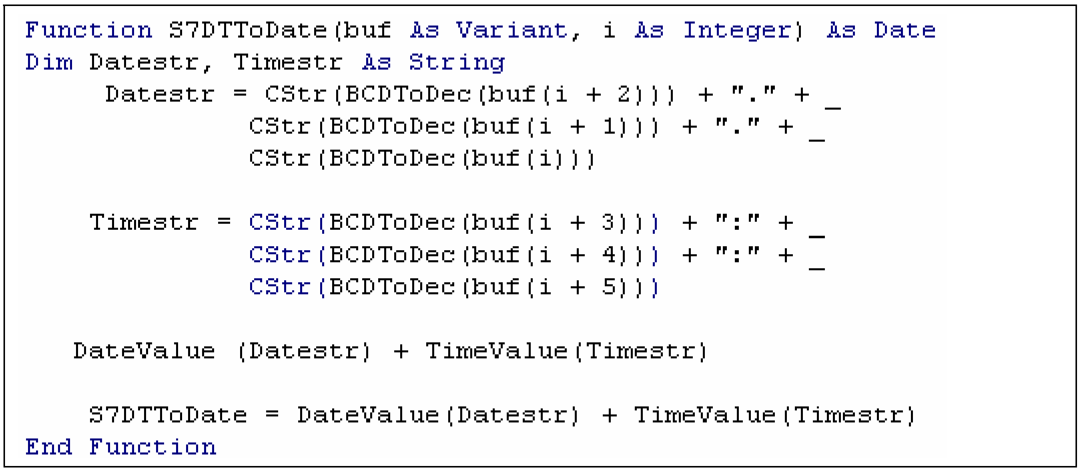


```
Function S7BytesToSingle(buf As Variant, i As Integer) As Single
Dim z As Single
    CopyMemory ByVal VarPtr(z) + 2, CLng(TwoByteToInt(buf(i + 1), buf(i))), 2
    CopyMemory ByVal VarPtr(z), CLng(TwoByteToInt(buf(i + 3), buf(i + 2))), 2
 S7BytesToSingle = z
End Function
```

Bytes ←→VBA date

The following graphic shows the conversion into a VBA date. The read bytes represent a Date_and_Time in the CPU.

Figure 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 Explanations on the STEP7 program

This chapter gives you further details on the STEP7 program:

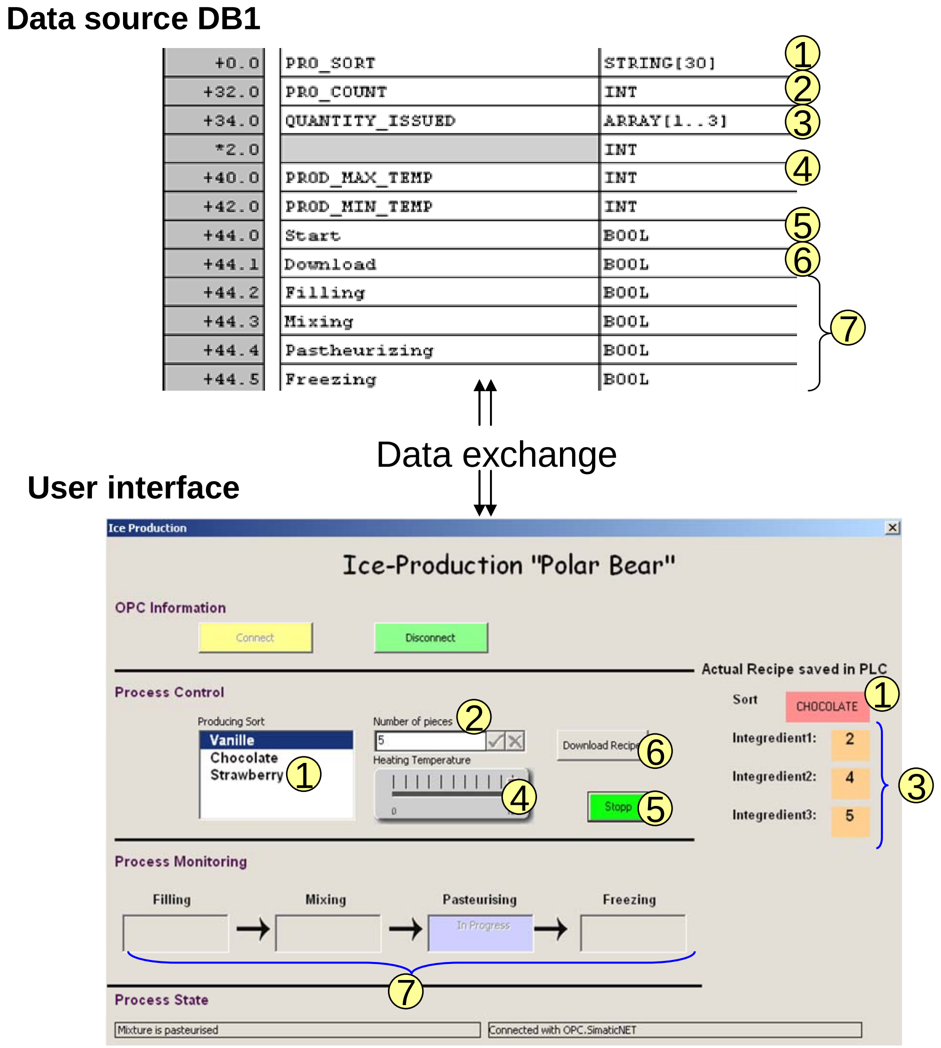
- Visualization interface
- Archive data block
- Stock

Visualization interface

DB1 “DB_ACT_PROD“ works as data source for the operator user interface. It summarizes all variables required for operator control & monitoring.

The following table illustrates the relationship between the operator user interface and the data block.

Figure 5-10



Data source DB1

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

Data exchange

User interface

Ice Production

Ice-Production "Polar Bear"

OPC Information

Process Control

Producing Sort: Vanille, Chocolate, Strawberry

Number of pieces: 5

Heating Temperature

Actual Recipe saved in PLC

Sort: CHOCOLATE

Integredient1: 2

Integredient2: 4

Integredient3: 5

Process Monitoring

Filling → Mixing → Pasteurising (In Progress) → Freezing

Process State

Mixture is pasteurised

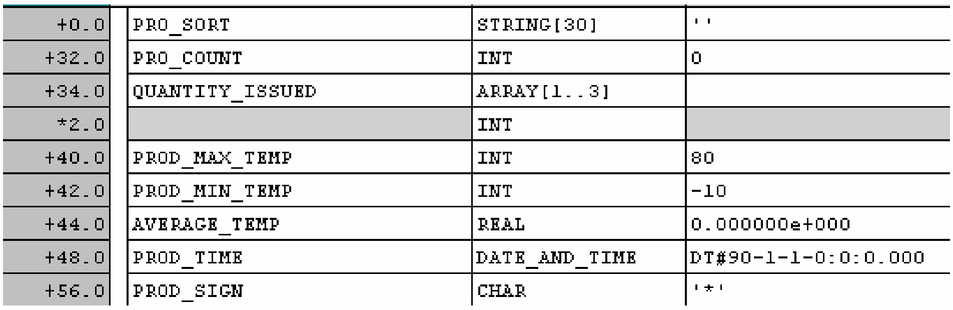
Connected with OPC.SimaticNET

Archive data block DB 2

Data block “PROD_INF“ contains information on the completed production process such as recipe data, production time and average temperature. These data can be loaded and read in the archive section of the “Archive“ Excel spreadsheet.

The following table shows the structure of the DB 2.

Figure 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	'*'

Modification options

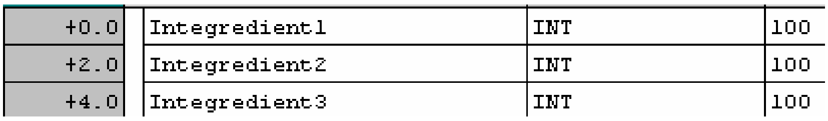
In this application, the archive data in the data block are overwritten with the next production. If the information on the production processes are saved over a longer period of time, the STEP7 project must first be adjusted accordingly. This includes the option of programming the data block as ring buffer for saving the production information.

Stock

The “PROD_STOCK“ data block manages the stock for the individual ingredients.

The following table shows the structure of the DB 6.

Figure 5-12



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

Structure, Configuration and Operation of the Application

Content

This part leads you step by step through the structure, important configuration steps, commissioning and operation of the application.

6 Installation and Commissioning

Here you find information on …

the hardware and software to be installed, and the steps necessary for commissioning the example.

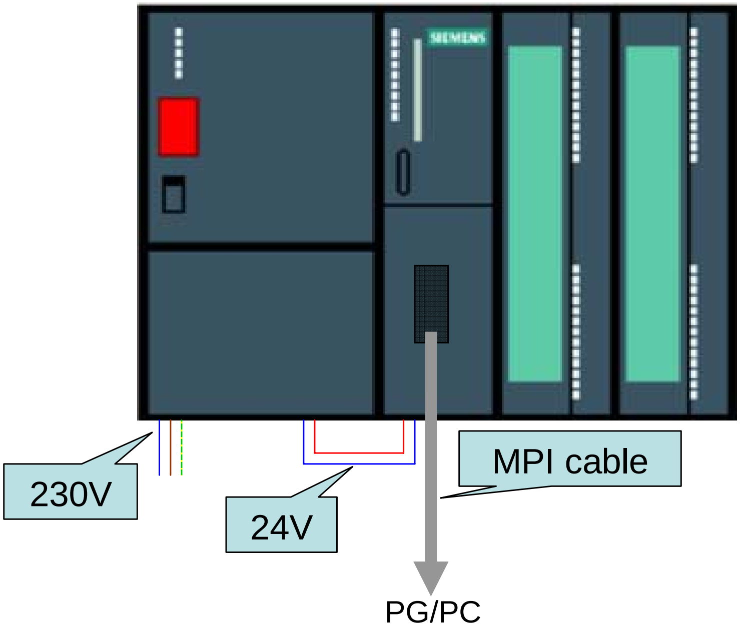
6.1 Installation of Hardware and Software

This chapter describes the hardware and software components to be installed. The descriptions and manuals as well as delivery information included in the delivery of the respective products should be observed in any case.

Installation of the hardware

Set up the SIMATIC hardware components according to the following figure and the item list in chapter 2.3.

Figure 6-1



Note The setup guidelines for installing the individual components must generally be considered.

Installation of the standard software

Table 6-1

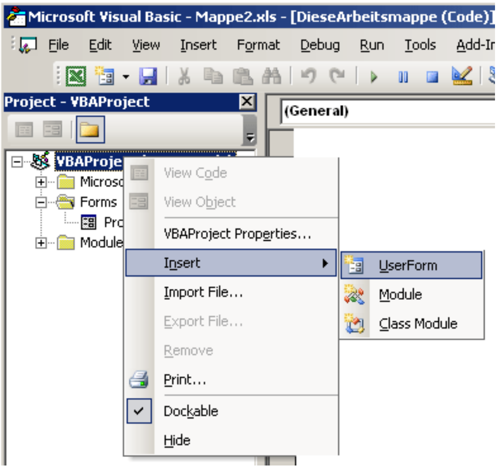
No.	Instructions	Note
5.	Installation of STEP 7 V5.4	Follow the instructions of the installation program
6.	Install the SIMATIC NET SOFTWARE EDITION 2005	Follow the instructions of the installation program

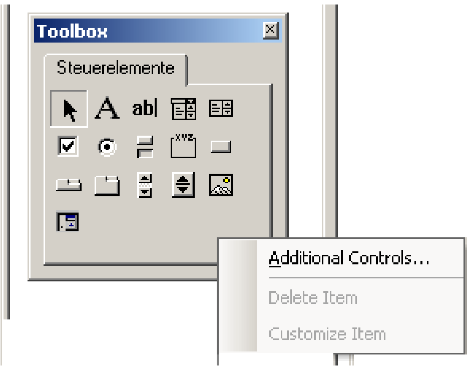
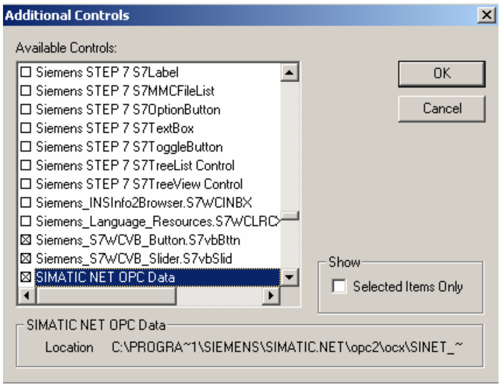
6.2 Configuration of the OPC client

The following instruction only provides a better understanding of integrating the S7 controls in VBA and configuring the OCX Data Control. The sample application already includes these configurations.

Integrating the S7Controls into the VBA

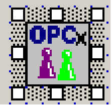
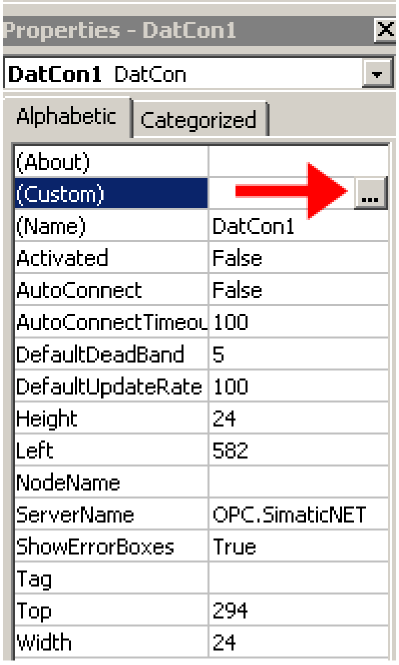
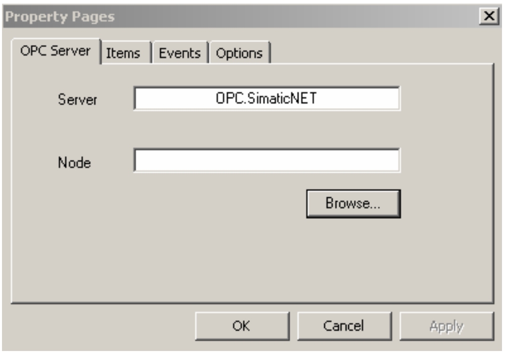
Table 6-2

No.	Procedure	Note
7.	Open the Excel folder of the example application	
8.	Open the VBA-Editor in the folder.	At **`Tools -> Macros -> Visual Basic Editor`**
9.	Select the VBA project. Via the right mouse button you enter a UserForm with **`Insert -> UserForm`**.	 Microsoft Visual Basic - Mappe2.xls - [DieseArbeitsmappe (Code)] File Edit View Insert Format Debug Run Tools Add-Ir Project - VBAProject (General) VBAProje Microsc Forms Prc Module View Code View Object VBAProject Properties... Insert Import File... Export File... Remove Print... Dockable Hide UserForm Module Class Module

No.	Procedure	Note
10.	Click Toolbox with the right mouse button and select **`Additional Controls…`** A new dialog box appears.	 Toolbox Steuerelemente Additional Controls... Delete Item Customize Item
11.	Select the display controls • Siemens S7Label Control • Siemens S7Number Control • Siemens_S7WCVB_Button.S7vbBttn • Siemens_S7WCVB_Slider.S7vbSlid and the data control SIMATIC NET OPC DATA Acknowledge the dialog with OK. The controls have now been accepted in the toolbox.	Additional Controls Available Controls: ☐ Siemens STEP 7 S7Label ☐ Siemens STEP 7 S7MMCFileList ☐ Siemens STEP 7 S7OptionButton ☐ Siemens STEP 7 S7TextBox ☐ Siemens STEP 7 S7ToggleButton ☐ Siemens STEP 7 S7TreeList Control ☐ Siemens STEP 7 S7TreeView Control ☐ Siemens_INSInfo2Browser.S7WCINBX ☐ Siemens_Language_Resources.S7WCLRC ☒ Siemens_S7WCVB_Button.S7vbBttn ☒ Siemens_S7WCVB_Slider.S7vbSlid ☒ SIMATIC NET OPC Data OK Cancel Show ☐ Selected Items Only SIMATIC NET OPC Data Location C:\PROGRA~1\SIEMENS\SIMATIC.NET\opc2\ocx\SINET_~

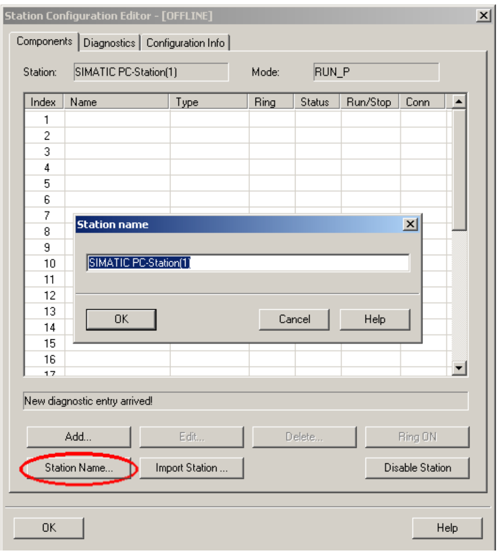
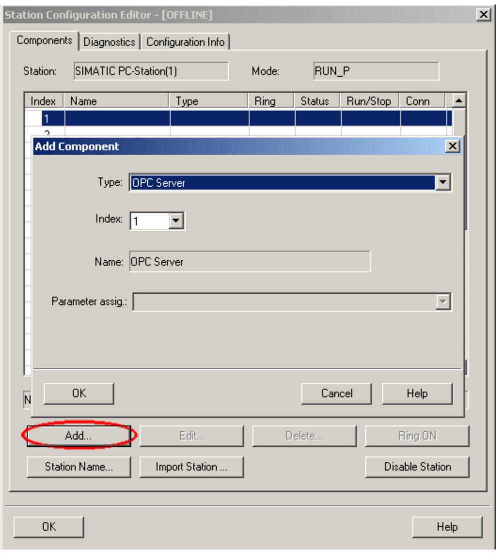
Configuration of the OCX Data Control

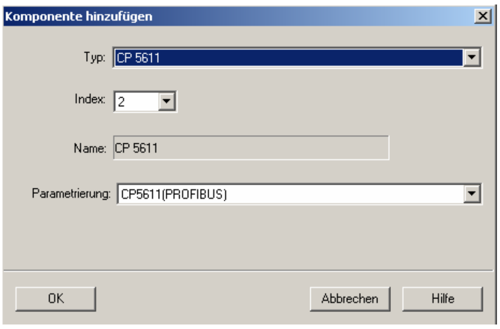
Table 6-3

No.	Procedure	Note
1.	Select the symbol for the OCX Data Control in your UserForm. Right-click the Properties option.	
2.	Select the **`Custom`** field. The "**...**" button appears. Press the button.	Properties - DatCon1 DatCon1 DatCon Alphabetic | Categorized (About) (Custom) ... (Name) DatCon1 Activated False AutoConnect False AutoConnectTimeou 100 DefaultDeadBand 5 DefaultUpdateRate 100 Height 24 Left 582 NodeName ServerName OPC.SimaticNET ShowErrorBoxes True Tag Top 294 Width 24
3.	The properties page of the OCX Data Control appears. Here you can configure the OPC server, link items with display controls, and create events.	Property Pages OPC Server | Items | Events | Options Server: OPC.SimaticNET Node: Browse... OK | Cancel | Apply

6.3 Configuration of the component configurator

Table 6-4

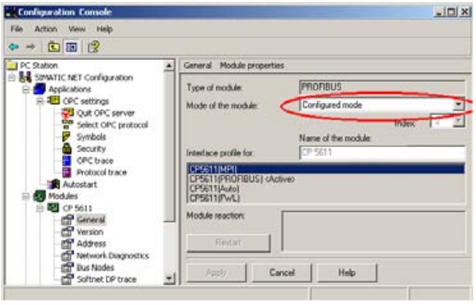
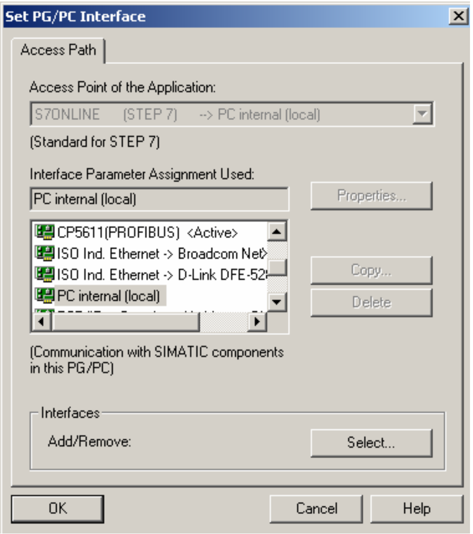
Step	Procedure	Note
1.	Open the **Component Configurator**	**`Start -> Station Configurator`**
2.	In the PC station, under **`Station Name…`** , you enter the same name as in the STEP7 program. In this example: **SIMATIC PC-Station(1)**	 Station Configuration Editor - [OFFLINE] Station: SIMATIC PC-Station(1) Mode: RUN_P Station name: SIMATIC PC-Station(1) New diagnostic entry arrived! Components Diagnostics Configuration Info Index Name Type Ring Status Run/Stop Conn OK Cancel Help Add... Edit... Delete... Ring ON Station Name... Import Station ... Disable Station
3.	Now add the new OPC connection by clicking the first line and pressing “**Add…**“. The **Add Component** dialog appears. Select the **OPC Server** as type and acknowledge this and the following dialog with OK.	 Station Configuration Editor - [OFFLINE] Station: SIMATIC PC-Station(1) Mode: RUN_P Add Component Type: OPC Server Index: 1 Name: OPC Server Parameter assig.: Components Diagnostics Configuration Info Index Name Type Ring Status Run/Stop Conn OK Cancel Help Add... Edit... Delete... Ring ON Station Name... Import Station ... Disable Station
4.	The OPC server is now integrated in the component configurator.	

Step	Procedure	Note
5.	Now add a CP5611. To do this you select the second line and press **Add…** The **Add Component** dialog appears. Select the **CP5611** as type and acknowledge this and the following dialog with OK.	 Komponente hinzufügen Typ: CP 5611 Index: 2 Name: CP 5611 Parametrierung: CP5611(PROFIBUS)
6.	Close the “Station Configuration Editor” dialog box.	

6.4 Configuration of the engineering PG/PC

Set PG/PC interface

Table 6-5

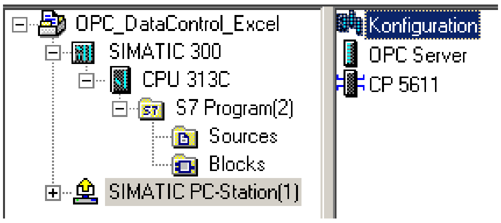
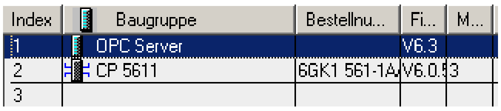
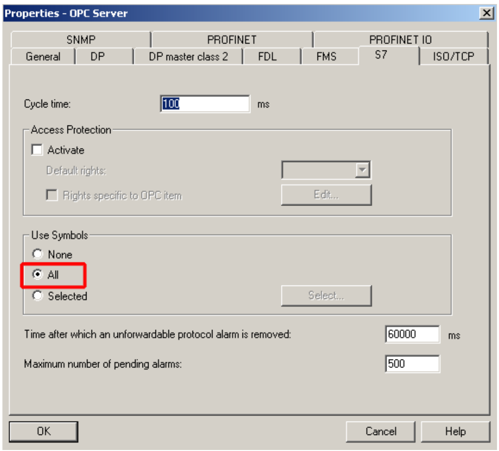
No.	Instructions	Note / Explanation
1.	With **`Start -> SIMATIC NET`** you enter the **Configuration Console**. In the **Modules** submenu you select the installed MPI card of your PGs/PCs (here CP5611). In **General** you select **Configuration Mode** as mode type**.**	 Configuration Console Type of module: PROFIBUS Mode of the module: Configured mode Name of the module: CP 5611 Interface profile for: CP5611(MPI) CP5611(PROFIBUS) <Active> CP5611(Auto) CP5611(FWL) File Action View Help PC Station SIMATIC NET Configuration Applications OPC settings Quit OPC server Select OPC protocol Symbols Security OPC trace Protocol trace Autostart Modules CP 5611 General Version Address Network Diagnostics Bus Nodes Softnet DP trace General Module properties Index Module reaction: Restart Apply Cancel Help
2.	Call up the **Set PG/PC interface** program via **`START -> Settings -> Control Panel`**. There you adjust the S7ONLINE access point to the PC_internal interface of your PC/PG.	 Set PG/PC Interface Access Path Access Point of the Application: S7ONLINE (STEP 7) --> PC internal (local) (Standard for STEP 7) Interface Parameter Assignment Used: PC internal (local) CP5611(PROFIBUS) <Active> ISO Ind. Ethernet -> Broadcom Net ISO Ind. Ethernet -> D-Link DFE-52 PC internal (local) (Communication with SIMATIC components in this PG/PC) Interfaces Add/Remove: Properties... Copy... Delete Select... OK Cancel Help

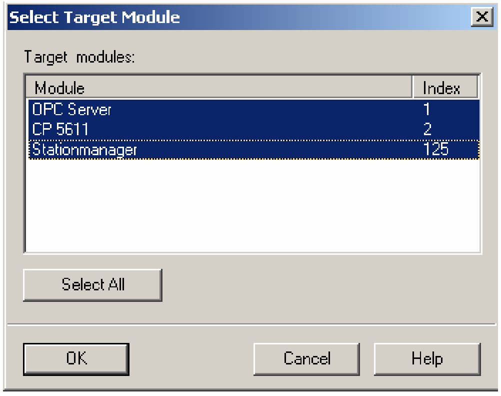
6.5 Commissioning the SIMATIC PC station

Unzip the file 23829402_OPC_DATCON_EXCEL_CODE_v10.zip into any directory.
This file contains

- the STEP7 project **opc_data.zip**
- the Excel file **IceProduktion.xls**

Table 6-6

Step	Procedure	Note
1.	Start the SIMATIC Manager	
2.	Retrieve the **opc_data.zip** project	With **`File -> Retrieve`**
3.	Change to the configuration of the SIMATIC PC station	OPC_DataControl_Excel SIMATIC 300 CPU 313C S7 Program(2) Sources Blocks SIMATIC PC-Station(1) Konfiguration OPC Server CP 5611
4.	Select the OPC server and double-click to open the Properties.	Index | Baugruppe | Bestellnu... | Fi... | M... 1 | OPC Server | | V6.3 | 2 | CP 5611 | 6GK1 561-1A | V6.0.53 | 3
5.	Switch to the **S7** tab and activate **All** in User Symbols. Acknowledge the dialogue with OK. This adopts all the symbols into the system data of the OPC server.	Properties - OPC Server SNMP | PROFINET | PROFINET IO General | DP | DP master class 2 | FDL | FMS | S7 | ISO/TCP Cycle time: 100 ms Access Protection □ Activate Default rights: □ Rights specific to OPC item Edit... Use Symbols ○ None ● All ○ Selected Select... Time after which an unforwardable protocol alarm is removed: 60000 ms Maximum number of pending alarms: 500 OK | Cancel | Help
6.	Compile your configuration and load it onto your PC/PG.	

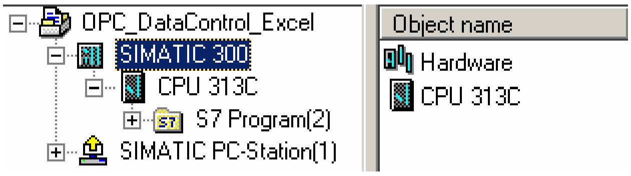
Step	Procedure	Note
7.	In the **Select Target Module** dialog you select the OPC server, the CP and the station manager and acknowledge with OK.	 Select Target Module Target modules: Module | Index OPC Server | 1 CP 5611 | 2 Stationmanager | 125
8.	The configuration is now saved onto your PC/PG.	

6.6 Commissioning the SIMATIC S7 station

This chapter describes the commissioning process for the SIMATIC S7 station.

Note The STEP 7 project delivered with this example contains the entirely configured SIMATIC S7 station with control program. This project can only be used without adjustment if the hardware is identical with the configuration.

Table 6-7

No.	Instructions	Note
1.	Select the **SIMATIC 300** station and load the station into the controller.	 OPC_DataControl_Excel SIMATIC 300 CPU 313C S7 Program(2) SIMATIC PC-Station(1)

7 Configuration Process

Here you find information on …

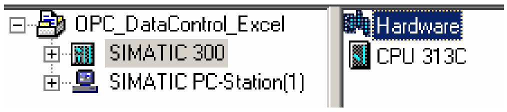
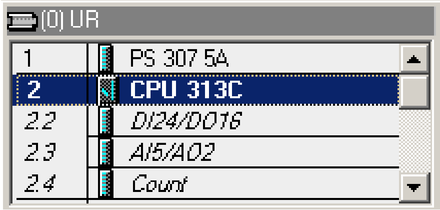
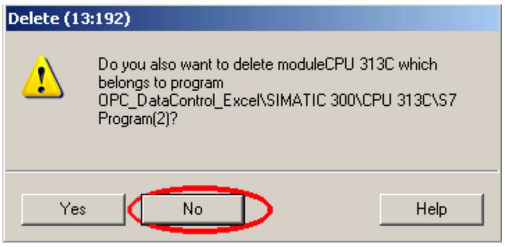
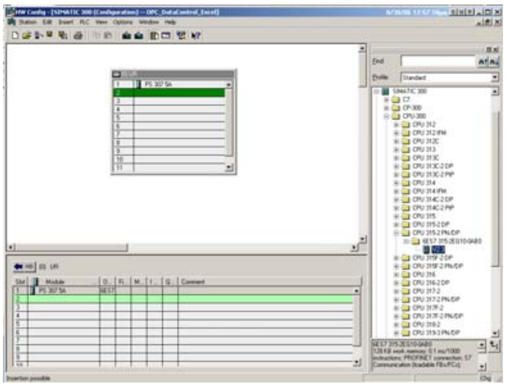
the configuration steps necessary if the hardware deviates from the hardware used in the example.

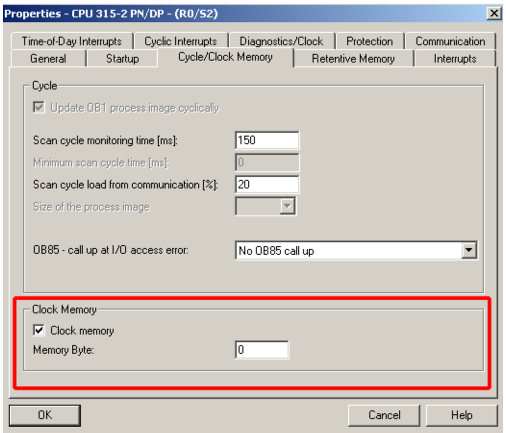
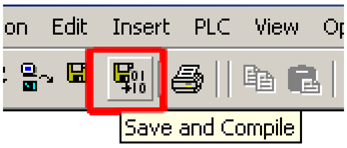
7.1 Configuration of a new CPU in the SIMATIC S7 station

The configuration of the SIMATIC S7 station is performed with STEP 7 and described step by step using the example of a CPU315-2PN.

Integrating another CPU

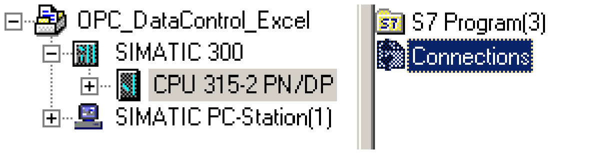
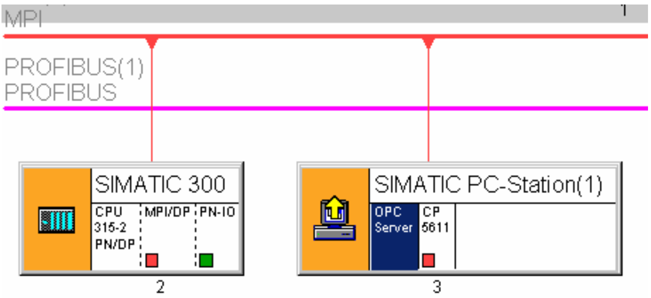
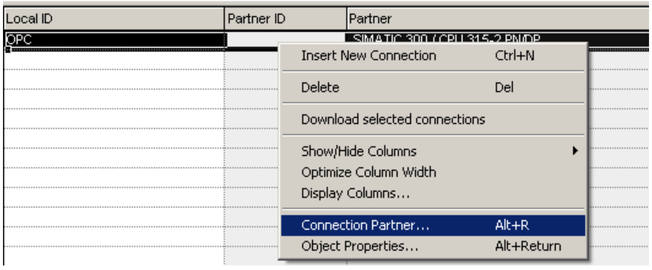
Table 7-1

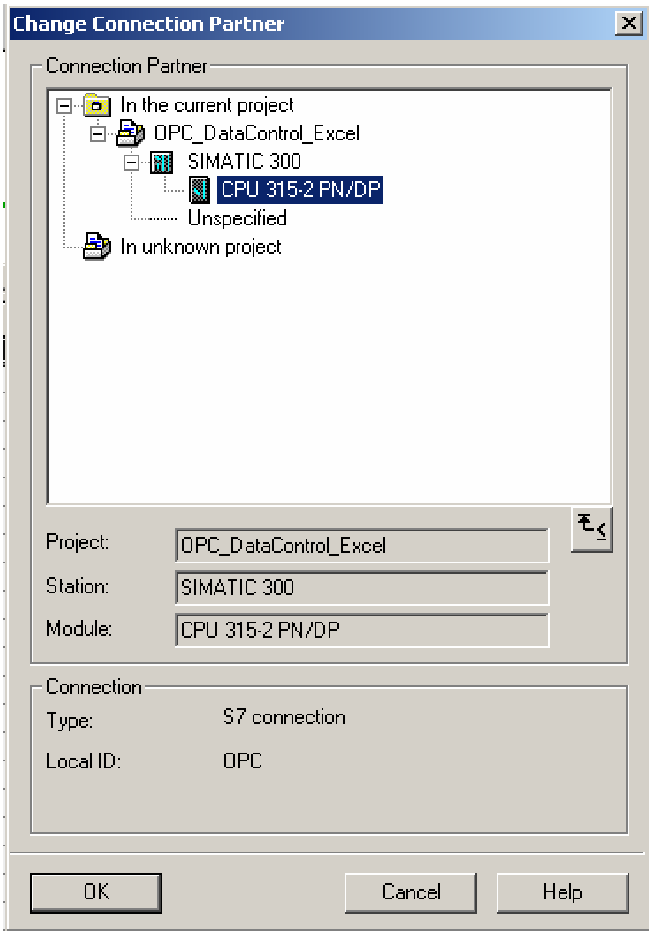
Step	Procedure	Note
1.	Start the SIMATIC Manager	
2.	Retrieve the **opc_data.zip** project	With **File -> Retrieve**
3.	Open the hardware configuration by double-clicking **Hardware** in the **SIMATIC** 300 folder	 OPC_DataControl_Excel SIMATIC 300 SIMATIC PC-Station(1) Hardware CPU 313C
4.	In the hardware configuration you select CPU313C and delete it from the project. Confirm the delete process in the following dialog with **Yes.**	 (0) UR 1 PS 307 5A 2 CPU 313C 2.2 DI24/DO16 2.3 AI5/AO2 2.4 Count
5.	Do **not** delete the attached S7 program! Quit the dialog with **No**.	 Delete (13:192) Do you also want to delete moduleCPU 313C which belongs to program OPC_DataControl_Excel\SIMATIC 300\CPU 313C\S7 Program(2)?
6.	Select your new CPU from the hardware catalog and draw it to slot 2 via drag&drop.	

Step	Procedure	Note
7.	Activate the MPI interface of your CPU.	
8.	Open the properties of the CPU and switch to the **`Cycle/Clock Memory`** tab. Activate **Memory bit 0**. Close dialog box with "OK".	 Properties - CPU 315-2 PN/DP - (R0/S2) Time-of-Day Interrupts Cyclic Interrupts Diagnostics/Clock Protection Communication General Startup Cycle/Clock Memory Retentive Memory Interrupts Cycle ☑ Update OB1 process image cyclically Scan cycle monitoring time [ms]: 150 Minimum scan cycle time [ms]: 0 Scan cycle load from communication [%]: 20 Size of the process image OB85 - call up at I/O access error: No OB85 call up Clock Memory ☑ Clock memory Memory Byte: 0
9.	Save and compile the hardware configuration. Close the hardware configuration.	 on Edit Insert PLC View Op Save and Compile

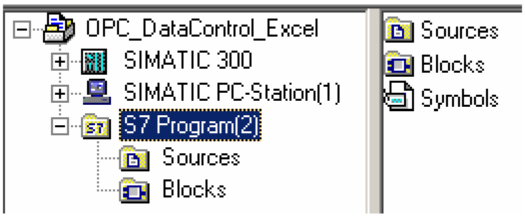
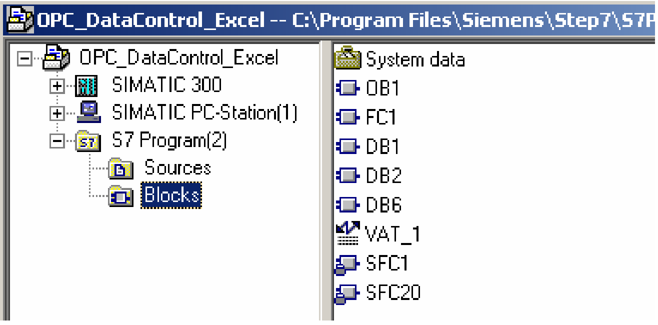
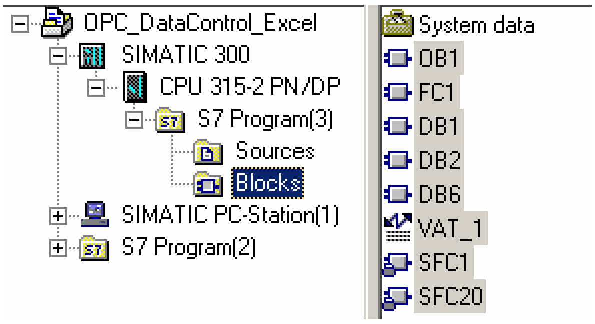
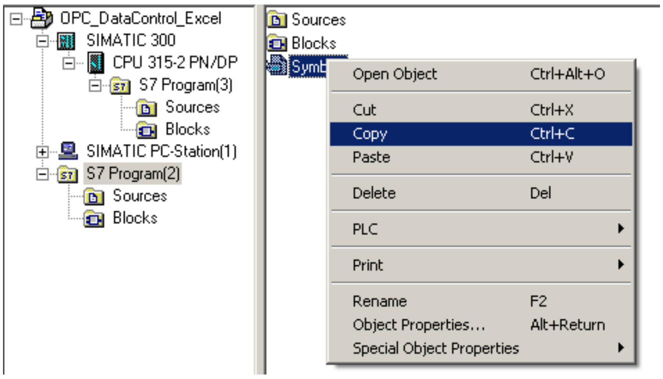
Connection configuration

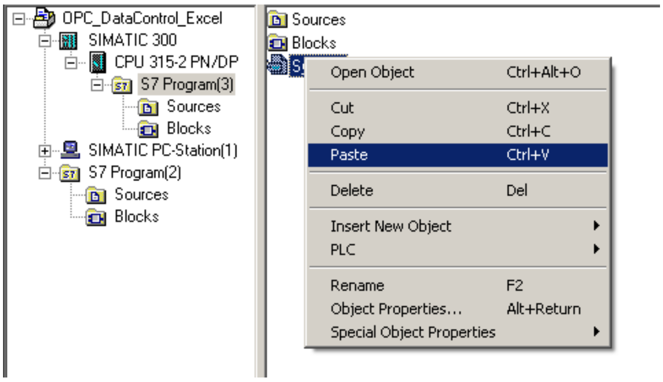
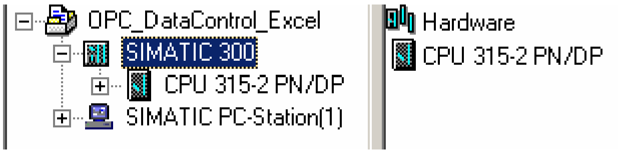
Table 7-2

No.	Procedure	Note
1.	Open the connection configuration by double-clicking **`Connections`** in the CPU folder.	 OPC_DataControl_Excel SIMATIC 300 CPU 315-2 PN/DP SIMATIC PC-Station(1) S7 Program(3) Connections
2.	NetPro opens. There you select the **OPC server.**	 MPI 1 PROFIBUS(1) PROFIBUS SIMATIC 300 CPU 315-2 PN/DP MPI/DP PN-IO 2 SIMATIC PC-Station(1) OPC Server CP 5611 3
3.	Select the **OPC connection.** Right-click the connection and select **`Connection Partner….`**	 Local ID Partner ID Partner OPC SIMATIC 300 / CPU 315-2 PN/DP Insert New Connection Ctrl+N Delete Del Download selected connections Show/Hide Columns Optimize Column Width Display Columns… Connection Partner… Alt+R Object Properties… Alt+Return

No.	Procedure	Note
3.	Select your CPU and acknowledge the dialog with **OK**. The OPC connection contains your added CPU as a connection partner.	 Change Connection Partner Connection Partner In the current project OPC_DataControl_Excel SIMATIC 300 CPU 315-2 PN/DP Unspecified In unknown project Project: OPC_DataControl_Excel Station: SIMATIC 300 Module: CPU 315-2 PN/DP Connection Type: S7 connection Local ID: OPC
4.	Save and compile the NetPro configuration. Close NetPro.	

Integrating the blocks into the new CPU

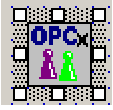
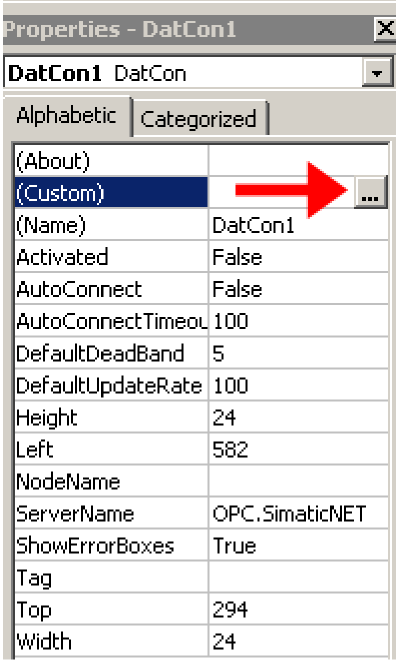
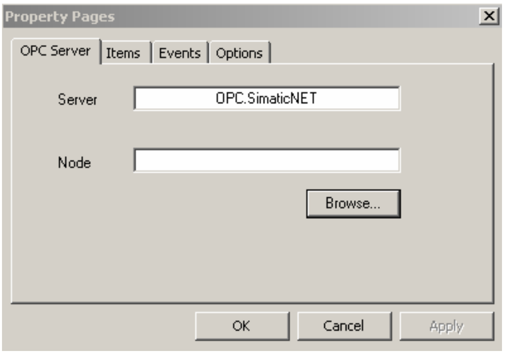
No.	Instructions	Note
1.	Blocks and symbols of the previous CPU were filed in a separate folder.	 OPC_DataControl_Excel SIMATIC 300 SIMATIC PC-Station(1) S7 Program(2) Sources Blocks Sources Blocks Symbols
2.	Open the **Blocks** folder.	 OPC_DataControl_Excel -- C:\Program Files\Siemens\Step7\S7P OPC_DataControl_Excel SIMATIC 300 SIMATIC PC-Station(1) S7 Program(2) Sources Blocks System data OB1 FC1 DB1 DB2 DB6 VAT_1 SFC1 SFC20
	Select all blocks (apart from the system data) via right mouse button -> **Copy.** Switch to the block folder of your new CPU and add the previously copied blocks via right mouse button -> **Paste** .	 OPC_DataControl_Excel SIMATIC 300 CPU 315-2 PN/DP S7 Program(3) Sources Blocks SIMATIC PC-Station(1) S7 Program(2) System data OB1 FC1 DB1 DB2 DB6 VAT_1 SFC1 SFC20
3.	Switch back to the separate program folder. Select the symbols file and copy it via right mouse button -> **Copy**.	 OPC_DataControl_Excel SIMATIC 300 CPU 315-2 PN/DP S7 Program(3) Sources Blocks SIMATIC PC-Station(1) S7 Program(2) Sources Blocks Sources Blocks SymB Open Object Ctrl+Alt+O Cut Ctrl+X Copy Ctrl+C Paste Ctrl+V Delete Del PLC Print Rename F2 Object Properties... Alt+Return Special Object Properties

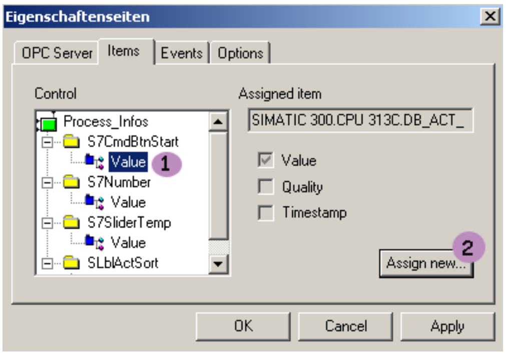
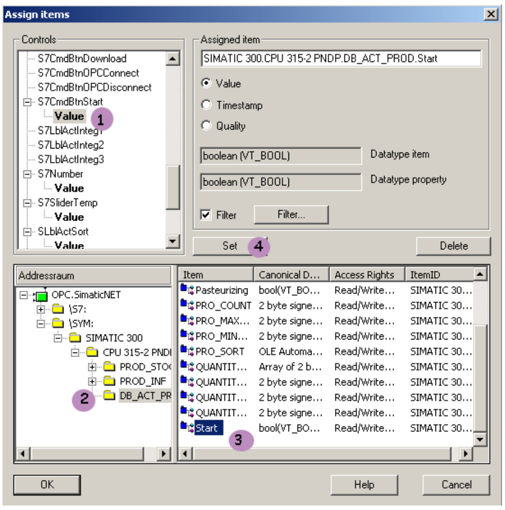
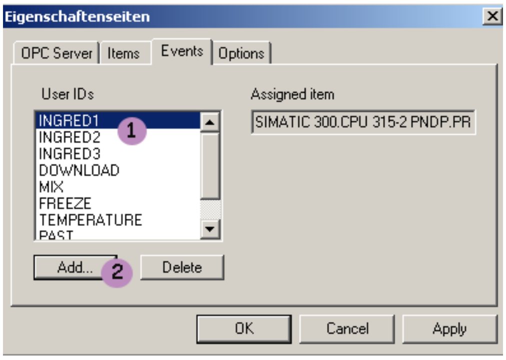
4.	Add the symbols file into the program folder of your new CPU via right mouse button -> **`Paste`**.	 OPC_DataControl_Excel SIMATIC 300 CPU 315-2 PN/DP S7 Program(3) Sources Blocks SIMATIC PC-Station(1) S7 Program(2) Sources Blocks Sources Blocks S Open Object Ctrl+Alt+O Cut Ctrl+X Copy Ctrl+C Paste Ctrl+V Delete Del Insert New Object PLC Rename F2 Object Properties... Alt+Return Special Object Properties
5.	Delete the separate program folder of the previous CPU.	
6.	Select the **SIMATIC 300** station and load the station into the controller.	 OPC_DataControl_Excel SIMATIC 300 CPU 315-2 PN/DP SIMATIC PC-Station(1) Hardware CPU 315-2 PN/DP
7.	Select the **SIMATIC PC station** and load it to your PC.	

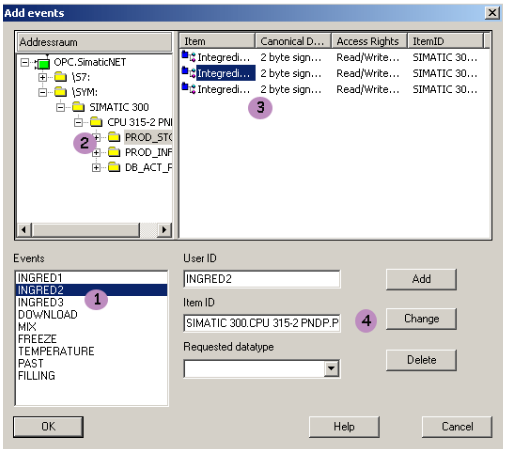
7.2 Change the ItemIDs in the VBA project

The symbolic addressing of the ItemID receives the CPU name as address area. If the CPU name is changed, the ItemIDs must be adjusted accordingly. Changes are made in the VBA editor.

Table 7-3

Step	Procedure	Note
1.	Open the Excel folder of the example application	
2.	Open the VBA-Editor in the folder.	At **`Tools -> Macros -> Visual Basic Editor`**
3.	Select the symbol for the OCX Data Control in your UserForm. Right-click the Properties option.	
4.	Select the **`Custom`** field (**`user defined`**). The **[...]** button appears. Press the button.	 Properties - DatCon1 DatCon1 DatCon Alphabetic | Categorized (About) (Custom) ... (Name) DatCon1 Activated False AutoConnect False AutoConnectTimeou 100 DefaultDeadBand 5 DefaultUpdateRate 100 Height 24 Left 582 NodeName ServerName OPC.SimaticNET ShowErrorBoxes True Tag Top 294 Width 24
5.	The properties page of the OCX Data Control appears. Switch to the **`Items`** tab.	 Property Pages OPC Server | Items | Events | Options Server OPC.SimaticNET Node Browse... OK Cancel Apply

Step	Procedure	Note
6.	In the **Items** tab the ItemIDs for the display controls are defined. Select a **Value** (1) and press the **`Assign new...`**(2) button. A new dialog box appears.	 Eigenschaftenseiten OPC Server | Items | Events | Options Control: Process_Infos, S7CmdBtnStart, Value (1), S7Number, Value, S7SliderTemp, Value, SLblActSort Assigned item: SIMATIC 300.CPU 313C.DB_ACT_ ☑ Value ☐ Quality ☐ Timestamp Assign new... (2) OK | Cancel | Apply
7.	Select a value (1) of a display control and in the address space navigate to the symbol of the suitable process variable (2). Select the suitable variable (3) and replace the symbolic ItemID with **`Set`** (4) This changes the symbolic ItemIDs of all values.	 Assign items Controls: S7CmdBtnDownload, S7CmdBtnOPCConnect, S7CmdBtnOPCDisconnect, S7CmdBtnStart, **Value** (1), S7LblActInteg1, S7LblActInteg2, S7LblActInteg3, S7Number, **Value**, S7SliderTemp, **Value**, SLblActSort, **Value** Assigned item: SIMATIC 300.CPU 315-2 PNDP.DB_ACT_PROD.Start ● Value ○ Timestamp ○ Quality boolean (VT_BOOL) Datatype item boolean (VT_BOOL) Datatype property ☑ Filter | Filter... Set (4) | Delete Addressraum: OPC.SimaticNET, \S7:, \SYM:, SIMATIC 300, CPU 315-2 PNDI, PROD_STO(, PROD_INF, DB_ACT_PR (2) Item | Canonical D... | Access Rights | ItemID Pasteurizing | bool(VT_BO... | Read/Write... | SIMATIC 30... PRO_COUNT | 2 byte signe... | Read/Write... | SIMATIC 30... PRO_MAX... | 2 byte signe... | Read/Write... | SIMATIC 30... PRO_MIN... | 2 byte signe... | Read/Write... | SIMATIC 30... PRO_SORT | OLE Automa... | Read/Write... | SIMATIC 30... QUANTIT... | Array of 2 b... | Read/Write... | SIMATIC 30... QUANTIT... | 2 byte signe... | Read/Write... | SIMATIC 30... QUANTIT... | 2 byte signe... | Read/Write... | SIMATIC 30... QUANTIT... | 2 byte signe... | Read/Write... | SIMATIC 30... Start (3) | bool(VT_BO... | Read/Write... | SIMATIC 30... OK | Help | Cancel
8.	If all display controls are assigned to the new symbolic addressing, close the dialog with OK.	
9.	Switch to the **`Events`** tab. Select an event (1). The **Add...** (2) takes you to a new dialog box.	 Eigenschaftenseiten OPC Server | Items | Events | Options User IDs: INGRED1 (1), INGRED2, INGRED3, DOWNLOAD, MIX, FREEZE, TEMPERATURE, PAST Assigned item: SIMATIC 300.CPU 315-2 PNDP.PR Add... (2) | Delete OK | Cancel | Apply

Step	Procedure	Note
10.	Select an event name (1) and in the address space navigate to the symbol of the suitable process variable (2). Select the variable (3) and replace the symbolic ItemID with **Change** (4) This changes the symbolic ItemIDs of all events.	 Add events Addressraum OPC.SimaticNET \S7: \SYM: SIMATIC 300 CPU 315-2 PNI PROD_STC PROD_INF DB_ACT_F 1 2 3 4 Add Change Delete OK Help Cancel Item / Canonical D… / Access Rights / ItemID Integredi… / 2 byte sign… / Read/Write… / SIMATIC 30… Integredi… / 2 byte sign… / Read/Write… / SIMATIC 30… Integredi… / 2 byte sign… / Read/Write… / SIMATIC 30… Events: INGRED1, INGRED2, INGRED3, DOWNLOAD, MIX, FREEZE, TEMPERATURE, PAST, FILLING User ID: INGRED2 Item ID: SIMATIC 300.CPU 315-2 PNDP.P Requested datatype:
11.	If all events are assigned to the new symbolic addressing, close the dialog with OK.	
12.	Close the properties window of the OCX Data Control.	All symbolic ItemIDs were adjusted to the new CPU.
13.	Save the VBA project.	

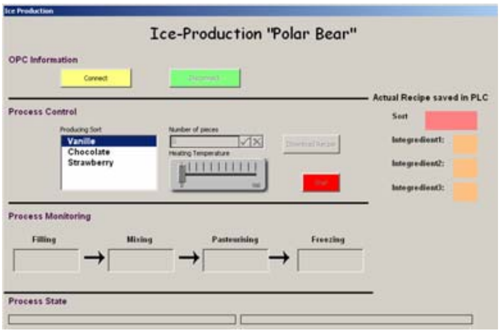
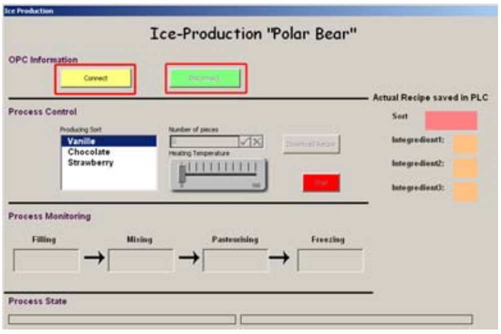
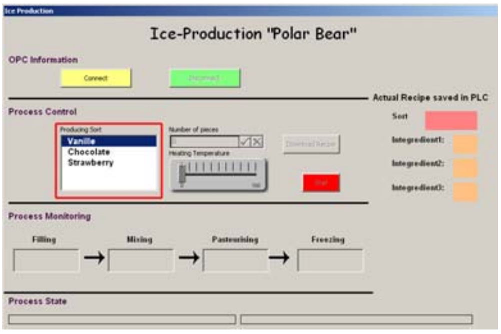
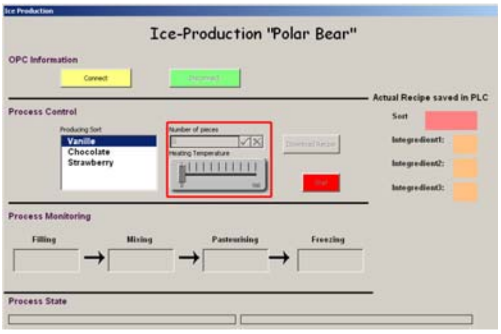
8 Operating the Application

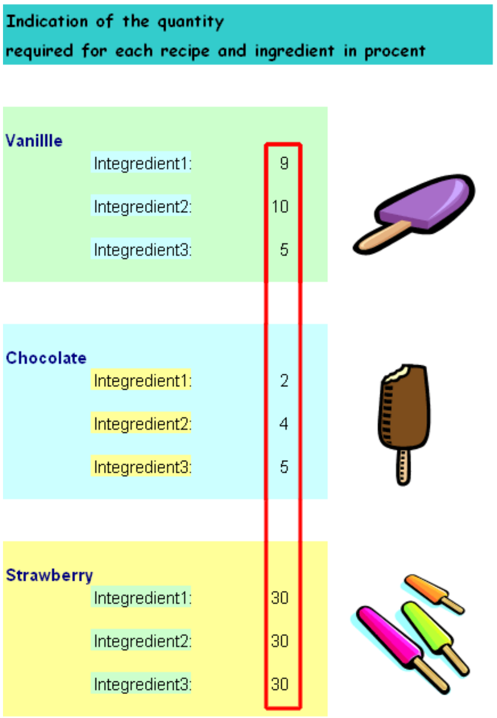
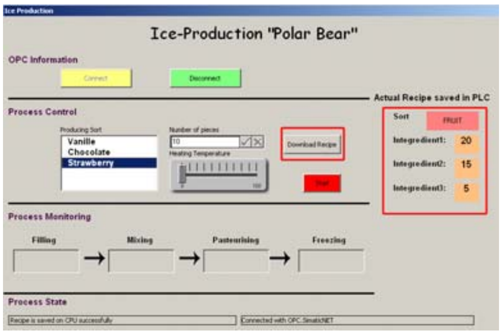
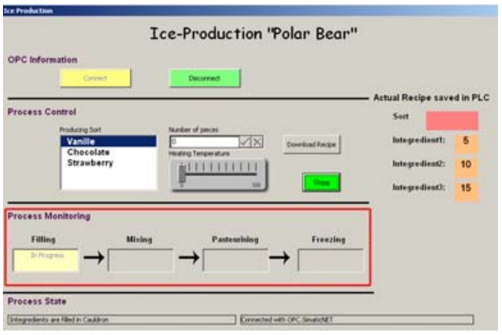
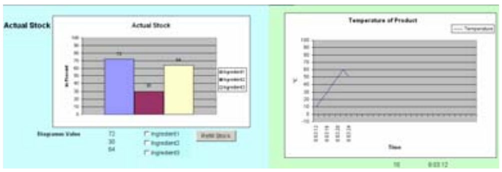
Here you will find information on …

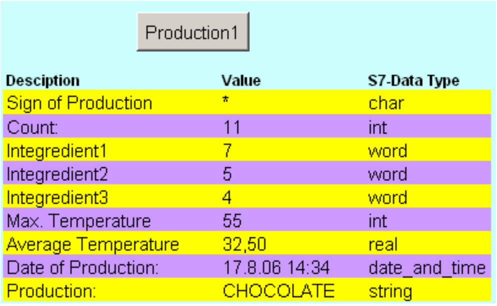
how to operated all functions of this application.

Operation of the ice production

Table 8-1

No.	Instructions	Note
1.	Open the Excel file **IceProduktion.xls.** The Ice Production dialog appears.	 Ice Production Ice-Production "Polar Bear" OPC Information Connect Disconnect Actual Recipe saved in PLC Process Control Producing Sort Vanille Chocolate Strawberry Number of pieces Heating Temperature Download Recipe Start Sort Integredient1: Integredient2: Integredient3: Process Monitoring Filling Mixing Pasteurising Freezing Process State
1.	The **Connect** button connects you with the OPC server. You disconnect by pressing the **Disconnect** button.	 Ice Production Ice-Production "Polar Bear" OPC Information Connect Disconnect Actual Recipe saved in PLC Process Control Producing Sort Vanille Chocolate Strawberry Number of pieces Heating Temperature Download Recipe Start Sort Integredient1: Integredient2: Integredient3: Process Monitoring Filling Mixing Pasteurising Freezing Process State
2.	In the selection field you select a type to be produced.	 Ice Production Ice-Production "Polar Bear" OPC Information Connect Disconnect Actual Recipe saved in PLC Process Control Producing Sort Vanille Chocolate Strawberry Number of pieces Heating Temperature Download Recipe Start Sort Integredient1: Integredient2: Integredient3: Process Monitoring Filling Mixing Pasteurising Freezing Process State
3.	You enter an item number and change the maximum production temperature by shifting the slider.	 Ice Production Ice-Production "Polar Bear" OPC Information Connect Disconnect Actual Recipe saved in PLC Process Control Producing Sort Vanille Chocolate Strawberry Number of pieces Heating Temperature Download Recipe Start Sort Integredient1: Integredient2: Integredient3: Process Monitoring Filling Mixing Pasteurising Freezing Process State

No.	Instructions	Note
4.	In the **Recipe** worksheet you can shift the setpoint values of the individual productions. Write the new values into the lines.	 Indication of the quantity required for each recipe and ingredient in procent Vanillle Integredient1: 9 Integredient2: 10 Integredient3: 5 Chocolate Integredient1: 2 Integredient2: 4 Integredient3: 5 Strawberry Integredient1: 30 Integredient2: 30 Integredient3: 30
5.	Adopt your personal production information into the CPU by pressing **Download Recipe**. The value are immediately displayed as current recipe.	 Ice Production Ice-Production "Polar Bear" OPC Information Process Control Producing Sort: Vanille, Chocolate, Strawberry (selected) Number of pieces: 10 Heating Temperature Actual Recipe saved in PLC Sort: FRUIT Integredient1: 20 Integredient2: 15 Integredient3: 5 Process Monitoring Process State Recipe is saved on CPU successfully Connected with OPC.SimaticNET
6.	Start the production with the **Start** button which is located in the dialog of the **Process** worksheet. In **Process Monitoring** you can monitor the process.	 Ice Production Ice-Production "Polar Bear" OPC Information Process Control Producing Sort: Vanille (selected), Chocolate, Strawberry Number of pieces: 0 Heating Temperature Actual Recipe saved in PLC Sort Integredient1: 5 Integredient2: 10 Integredient3: 15 Process Monitoring Filling: In Progress Process State Integredients are filled in Cauldron Connected with OPC.SimaticNET
7.	Switch to the Archive worksheet. Here you see the current stock displayed as column diagram and the temperature curve of the production.	 Actual Stock Diagramm Value: 72, 30, 64

No.	Instructions	Note
8.	In the production archive you can download the production recipe by pressing the button.	

Desciption	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

Appendix and List of Further Literature

9 Literature

9.1 Bibliographic References

This list is by no means exhaustive and only gives a selection of appropriate sources.

Figure 9-1

No.	Topic	Title
/1/	STEP7	Automatisieren mit STEP7 in AWL und SCL *[Automation with STEP7 in STL and SCL]* Hans Berger Publicis MCD Verlag ISBN 3-89578-113-4

9.2 Internet links

This list is by no means exhaustive and only gives a selection of appropriate sources.

No.	Topic	Title
1	Reference to this documentation	http://support.automation.siemens.com/WW/view/en/23829402
2	Siemens A&D Customer Support	http://www.ad.siemens.de/support
3	OPC Custom Interface, Interface description On the documentation CD-Rom by SIMATICNET Or the OPC foundation website for the download www.opcfoundation.org	OPC Data Access 3.0 specification
4	Description or information on: • General Information on OPC • Using OPC server Installed by SIMATIC NET, see: Start → SIMATIC → Documentation → [Language] In product support under the entry ID: 2044387	SIMATIC NET – Industrial Communication with PG/PC
5	Manual for generating the PC station Installed by SIMATIC NET, see: Start → SIMATIC → Documentation → [Language] In product support under the entry ID:13542666	SIMATIC NET Commissioning PC Stations - Manual / Quick Start
6	Modified installation instruction for CP 5411, CP 5511 and CP 5611 In product support under the entry ID: 284796	Modified installation instruction for CP 5411, CP 5511 and CP 5611
7	Manual / Operating instruction CP5611 In product support under the entry ID: 13654902	SIMATIC NET CP 5611 Installation instruction / product information

9.3 References to further OPC applications

No.	Topic	Description
1	Application: OPC Communication via the SEND/RECEIVE Protocol with a Visual Basic .NET OPC Client In product support under the entry ID: 21523291.	OPC Communication via the SEND/RECEIVE Protocol with a Visual Basic .NET OPC Client
2	Application: Data Exchange between different Bus Systems with OPC Data eXchange (OPC DX) In product support under the entry ID: 21523569).	Data Exchange between different Bus Systems with OPC Data eXchange (OPC DX)
3	Application: PC-based automation: Connection of databases via open interfaces by using OPC-XML, programmed in C# .net In product support under the entry ID: 21576581).	PC-based automation: Connection of databases via open interfaces by using OPC-XML, programmed in C# .net
4	Application: Using the XML-DA interface of the SIMATIC NET OPC server with Visual Basic .NET In product support under the entry ID: .	Using the XML-DA interface of the SIMATIC NET OPC server with Visual Basic .NET
5	Application: External Block Management with S7 Block services of the SIMATIC NET OPC server In product support under the entry ID: 21495347 .	External Block Management with S7 Block services of the SIMATIC NET OPC server
6	Application: Mass Data Acquisition with an OPC Client in C# based on .NET In product support under the entry ID: 21447513.	Mass Data Acquisition with an OPC Client in C# based on .NET
7	Application: Individual Visualization with OPC based on Microsoft® .NET and the Programming Language C# In product support under the entry ID: 21043779 .	Individual Visualization with OPC based on Microsoft® .NET and the Programming Language C#
8	Application: Controlling with the PC - The SIMATIC NET OPC Inproc Server as DP Master In product support under the entry ID: 21045282.	Controlling with the PC - The SIMATIC NET OPC Inproc Server as DP Master
9	Application: Decentralizing with the PC - The SIMATIC NET OPC Server as DP slave In product support under the entry ID: 21040390 .	Decentralizing with the PC - The SIMATIC NET OPC Server as DP slave