SIEMENS

SIMATIC NET

Step by Step: Excel OPC-Automation Client for access to S7-200 with CP243-1

Handbuch
Manual

Ausgabe 02/2003

Classification of Safety-Related Notices

This document contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:

Danger

indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning

indicates that death or severe personal injury **can** result if proper precautions are not taken.

Caution

with warning triangle indicates that minor personal injury can result if proper precautions are not taken.

Caution

without warning triangle indicates that damage to property can result if proper precautions are not taken.

Notice

indicates that an undesirable result or status can result if the relevant notice is ignored.

Note

highlights important information on the product, using the product, or part of the documentation that is of particular importance and that will be of benefit to the user.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 48 48, D-90327 Nürnberg

Disclaimer

We have checked the contents of this manual for agreement with the hardware described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Trademarks

SIMATIC®, SIMATIC NET®, SINEC® and SIMATIC NET Networking for Industry® are registered trademarks of Siemens AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Safety Instructions Regarding your Product

Before you use the product described here, read the safety instructions below thoroughly.

Qualified Personnel

Only qualified personnel should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage of Hardware Products

Please note the following instructions regarding the correct usage of hardware products:

Caution

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Before you use the supplied sample programs or programs you have written yourself, make certain that no injury to persons nor damage to equipment can result in your plant or process.

EU Directive: Do not start up until you have established that the machine on which you intend to run this component complies with the directive 89/392/EEC

Correct Usage of Software Products

Please note the following instructions regarding the correct usage of software products:

Caution

This software may only be used for the applications described in the catalog or the technical description, and only in connection with software products, devices, or components from other manufacturers which have been approved or recommended by Siemens.

Before you use the supplied sample programs or programs you have written yourself, make certain that no injury to persons nor damage to equipment can result in your plant or process.

Prior to Startup

Prior to startup, please note the following warning:

Caution

Prior to startup you must observe the notes in the relevant documentation. For ordering data of the documentation please refer to catalogs or contact your local SIEMENS representative.

Preface

Purpose of the Manual

This manual supports you when creating user programs and configurations in the OPC environment. The activities involved in creating a program and the required configuration work are presented in the form of a sequential series of steps. The configurations used are created with the basic software SIMATIC STEP 7 or NCM PC.

This manual serves as a reference work for configuration and communication with OPC components.

Aims

This manual should help you to expand the components described and to integrate them in your program.

We assume that you are thoroughly familiar with your programs and development environment.

The Package

This sample project consists of the following parts:

- Documentation commissioning, documentation Excel
- Microsoft Excel file incl. VBA implementation as example
- MicroWIN project file for download
- SimaticNet OPC project file for observing variables

Validity of this Manual

This manual applies the following software versions:

- SIMATIC MicroWin V**3.2 + SP1 or higher**
- SIMATIC NET S7 OPC Server is part of the SOFTNET S7/Windows 6.0 package of Industrial Ethernet

Baugruppen	MLFB
SimaticNet S7-OPC Server	6GK1704-1CW60-3AA0
Simatic S7-224 (DC-Version) Simatic S7-224 (Relais-Version)	6ES7 214-1AD22-0XB0 6ES7 214-1BD22-0XB0
Simatic CP243-1	6GK7 243-1EX00-0XE0

Required Documentation

The following documentation contains additional information on the STEP 7 basic software of the SIMATIC programmable controller and can be obtained from your local Siemens office.

Topic	Document	Order Number
S7-200 Dokumentation	Automation System S7-200 • System Manual	6ES7 298-8FA22-8BH0
Basic information on the OPC interface and installation and commissioning of the SIMATIC NET OPC Server.	Industrial Communication with SIMATIC NET • User Manual	6GK1 971-1GA00-0AA1

Orientation in the General Documentation Landscape

To set up the S7 controller and to prepare for operation, you require the following documentation:

- Automation System S7-200, System Manual

Structure of the Documentation

The SIMATIC NET Documentation includes the following:

- Manual
- Product information
- Installation instructions
- Readme.txt files on the data medium of the software product

Finding Your Way Through the Manual

To help you to locate specific information quickly, the manual includes the following tools:

- At the start of the manual, you will find a full table of contents and a list of figures and tables contained in the manual.
- In the chapters, you will see a brief overview of the contents of the sections in the left margin.
- After the appendix, there is a glossary that defines the most important technical terms used in the manual.
- At the back of the manual, there is a detailed index that allows you to locate information quickly.

Documentation on Programming

The appendix includes a list of the documentation you require to program and commission the S7 controller. You will also find a list of technical books on the topic of programmable controllers.

CD-ROM

You can also order the entire SIMATIC S7 documentation as a collection on CD-ROM.

Personnel Qualification Requirements

Only qualified personnel should be allowed to install and work on this equipment. Qualified personnel as referred to in the operating instructions or in the warning notes are defined as persons who are familiar with the installation, assembly, startup and operation of this product and who possess the relevant qualifications for their work, for example:

- Training in or authorization for connecting up, grounding or labeling circuits and devices or systems in accordance with current standards in safety technology;
- Training in or authorization for the maintenance and use of suitable safety equipment in accordance with current standards in safety technology;
- First aid qualification.

Further Support

If you have further questions on SIMATIC products, please contact your local Siemens office or representative. You will find the addresses in the catalogs, on the Internet and in CompuServe (go autforum).

Who to Contact

If you have technical questions about using the software and your problem is not dealt with in the documentation or in the integrated help system, please contact your Siemens representative or dealer.

You will find the addresses:

- In the "Readme.rtf" file in the main folder of the SIMATIC NET CD
- Internet http://www.siemens.de/simatic-net
- In Catalog IK PI

License

Note that you can only use the samples described on this CD if you have valid licenses for the software required.

Note

You can obtain demonstration versions of the required software products for test or demonstration purposes and to familiarize yourself with the functions.

Certification

The products and systems listed in this document are manufactured and marketed using a quality management system complying with DIN ISO 9001 and certified by DQS (certificate registration no. 2613). The DQS certificate is recognized in all IQNet countries (Reg. No.: 2613).

Standards and Approvals

The S7 controllers meet the requirements and criteria of IEC 1131, Part 2. The S7 controllers meet the requirements for the CE Mark. CSA, UL and FM approvals have been obtained for the S7 controllers.

You will find more detailed information on the approvals and standards in the appendix ....

Contents

Overview and Environment 1

In this chapter the sample plant and the functional principles of the environment are described. Additional the required hard- and software will be listed.

1.1 Schematic of the Environment

Block Diagram

The following figure shows the components of the sample plant in principle:

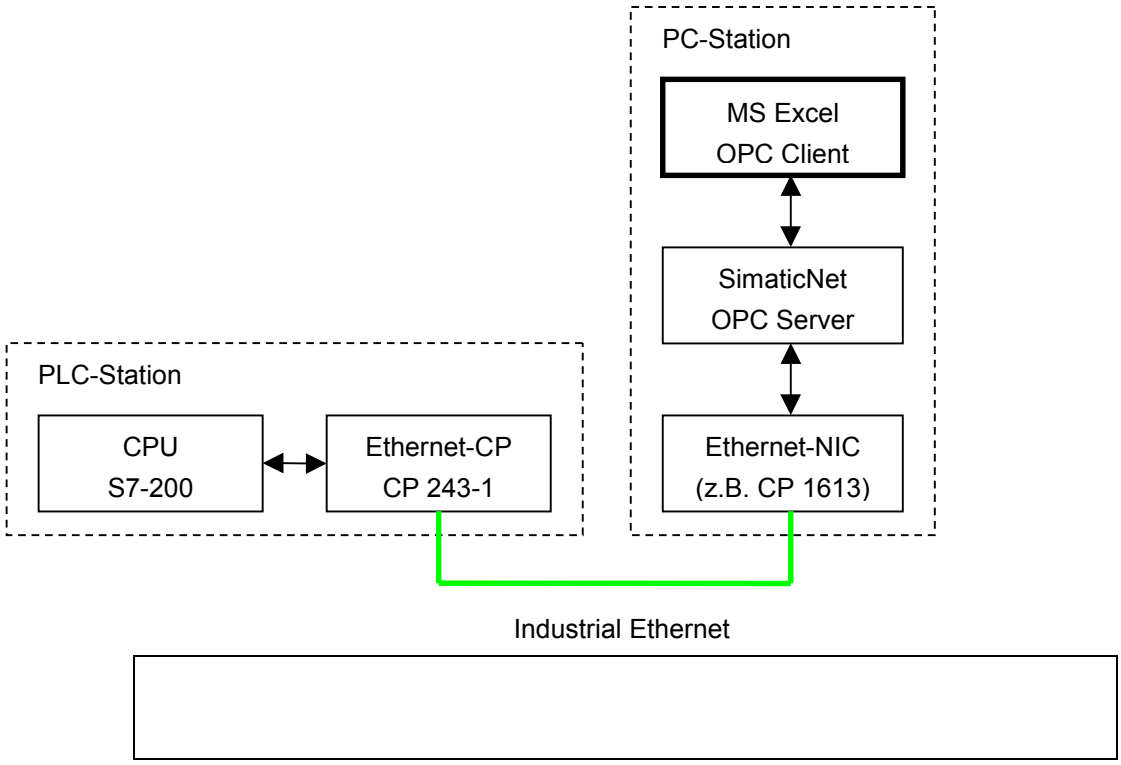

Figure 1-1 Block Diagram

General Description

The PLC Station consists of S7-200 with Ethernet Communication Processor (CP243-1), and id connected through Ethernet with SimaticNet OPC Servers. The access to the OPC Server is done using Microsoft Excel.

Note

Using a direct connection between two Ethernet communication devices, a so-called Cross Cable is required.

1.2 General Description

PLC Station

The S7-200 Controller including Communication Processor (CP243-1) is connected through Ethernet to the PC Station. The required configuration and programming is described in an additional document. A fault free configuration and programming is assumed regarding this example.

PC Station

On the PC the SimaticNet OPC Server is installed and configured. The required steps of configuration are described in an additional document. A fault free installation and configuration is assumed regarding this example.

Custom / Automation Interface

The OPC Server provides the data of the S7-200 on the PC side as server interface. The date is offered at the so called Custom Interface. C++ Clients can directly access this interface via Microsoft's COM/DCOM. Visual Basic Clients (e.g. Excel, Visual Basic for Application) must use the so called Automation Interface. The Automation Interface operates as translator (Wrapper) and implemented as DLL (Dynamic Link Library). This DLL is shipped with SimaticNet PC Software be referenced by the Client.

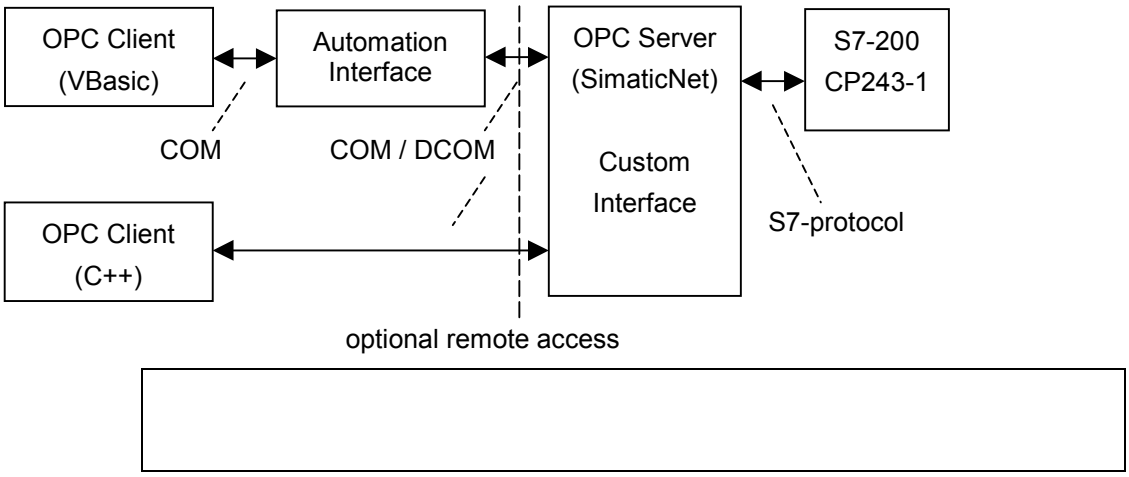

Figure 1-2 Custom / Automation Interface

COM / DCOM

Microsoft specifies the basic service for communication on windows based platforms. This mechanism is called COM (Component Object Model) respectively DCOM (Distributed COM). The specification governs communication between software objects. DCOM and its security settings comes into play when software components communicate over peripheral borders. With respect to DCOM a remote access operates the OPC Client on one machine accessing an OPC Server running on another computer.

Note

Correct DCOM security settings are required for successful remote access on OPC Servers. Only Administrators are allowed to change DCOM settings. A detailed description can be found in the online help of SimaticNet PC Software.

Important

Faulty settings of DCOM security can require reinstallation of the operating system. Additional information can be found in the Microsoft Knowledgebase.

1.3 Required Software

To run this example the following software packages are required.

Operating System

- Microsoft Windows 2000 Professional SP3 or Windows XP Professional SP1

SIMATIC

- SimaticNet CD 7/02, PC-Software V6.0 for Windows 2000 or SimaticNet CD 7/02, PC-Software V6.1 for Windows XP

This software package includes drivers for the communication processor CP1613 and other Ethernet cards (NIC), das S7 Protocol and the S7 OPC Server including the OPC Scout.

- SIMATIC NCM S7 IE V5.2 or SIMATIC STEP 7 V5.2

This software package includes configuration software HWKonfig and NetPro with the required Hardware Catalog.

Microsoft

- Microsoft Excel 2000 or Excel XP

This software package includes the spread sheets and Visual Basic for Applications.

1.4 Required Hardware

PC Station

IBM PC including Ethernet Card (e.g. Communication Processor CP1613)

PLC Station

SIMATIC S7-224 Controller including Ethernet Communication Processor CP243-1

Miscellaneous

Ethernet Hub and 2x Ethernet Cable with RJ45 connector, so-called patch cable or 1x Cross Cable for direct connection devices.

1.5 Suitability for a Different Hardware Configuration

Important

The samples relate to specific hardware configurations. These must exist to ensure problem-free operation.
If you want to use a different configuration, adaptations will be necessary. (See also the notes below)

Note

Using a different CPU (e.g.: CPU 222)

If you want to use a different CPU, you only need to run the Ethernet-Assistant as part of Step7-MicroWin again and the Module-Command-Byte must be changed.

Using a different Ethernet card in the PC

If you want to use a different PCI adapter from the CP1613, you only need to replace the adapter in the hardware configuration (HWKonfig). When using third party NIC the CP „IE General“ must be selected.

Programming the Client

2

This chapter describes the required steps for programming the OPC Client in Microsoft Excel.

2.1 Overview

General

The installation of Microsoft Excel and its components will not be described. The installation of the Ethernet adapter, the installation of SimaticNet Software including the required License is described in an additional document. Refer to the notes in the product manual and related product information including the read me files.

Note

Regarding this documentation fault free installation of SimaticNet PC Software and NCM PC Software and Microsoft Excel is assumed. Additionally a correctly configured OPC Server and S7-200 controller is assumed.

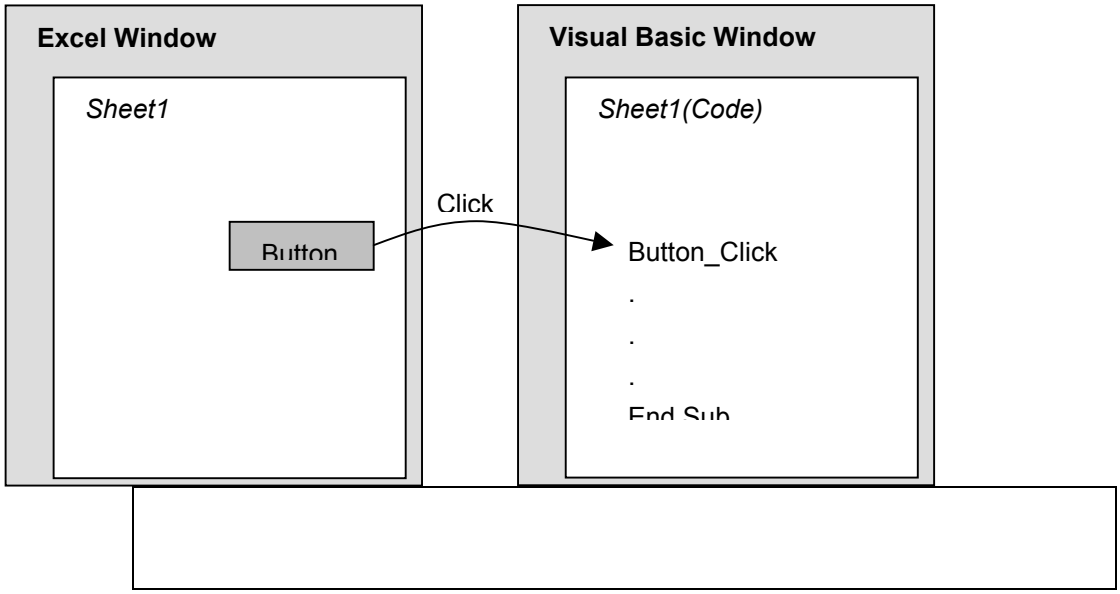

Figure 2-1 Excel and Visual Basic

Excel

Excel represents the user interface of the OPC Client. Acting as container for displaying- and monitoring components the sheet hosts e.g. the buttons. A message, coming from the embedded object (e.g. when clicking a button), results in executing the dedicated event method in code window of its container.

Visual Basic for Application

Visual Basic for Application is a reduced Version of Microsoft's develop studio and programming language Visual Basic. For each spreadsheet, but also for the work book, a separate code window exists within Visual Basic Editor. With respect to the functionality the code window of a spreadsheet is equal to a class module. The programmer must do the implementation of the event method.

Design of User Interface

In this example the user interface of the OPC Client is created inside the spreadsheet. From the menu bar (**View → Symbol Bar → Visual Basic**) the symbolic buttons of Visual Basic are activated. In Design Mode the buttons and Check Boxes can be placed on the sheet.

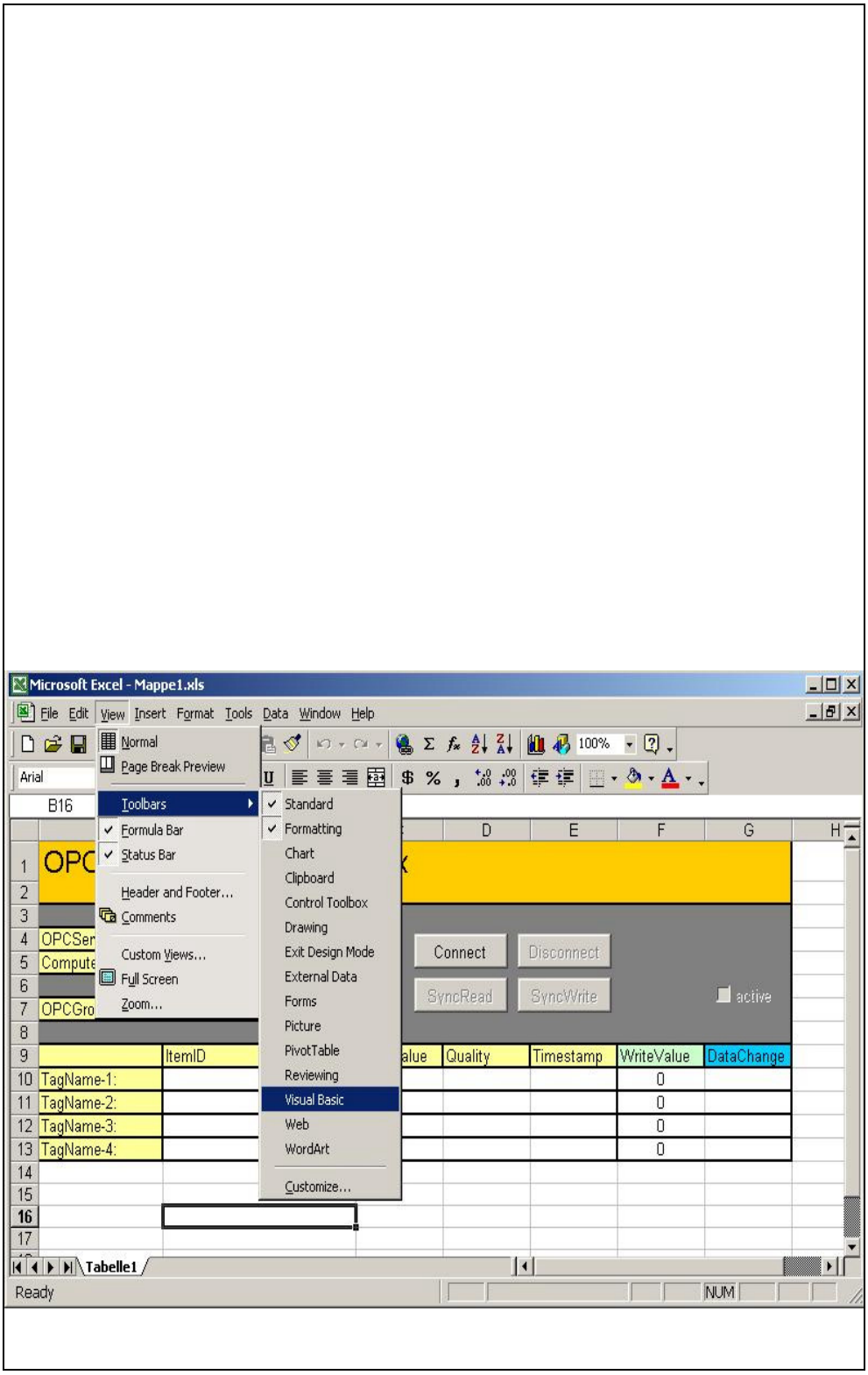

Figure 2-2 User Interface

The properties of the individual controls (right click on the control to pop up the properties) are defined regarding the following tables. Names are given regarding the desired function.

Connect OPC Server

Type: Button	Property
Name	cmdConnect
Enabled	True
Caption	Connect

Disconnect from OPC Server

Type: Button	Property
Name	cmdDisconnect
Enabled	False
Caption	Disconnect

Write into the Server

Type: Button	Property
Name	cmdSyncWrite
Enabled	False
Caption	SyncWrite

Read from the Server

Type: Button	Property
Name	cmdSyncRead
Enabled	False
Caption	SyncRead

Activate the Group

Type: Checkbox	Property
Name	chkActive
Enabled	True
Caption	active
Value	False

Note

Naming the individual controls is done for clarity. The object can be named different. However, the name of an object is part of the name of the dedicated event method.

2.2 Implementation

General

The Visual Basic Editor mainly consists of three windows. The Project Browser in the upper left for navigation within the project. The property window sowing the properties of the selected object down left. The main window showing the code and implementation is situated on the right side.

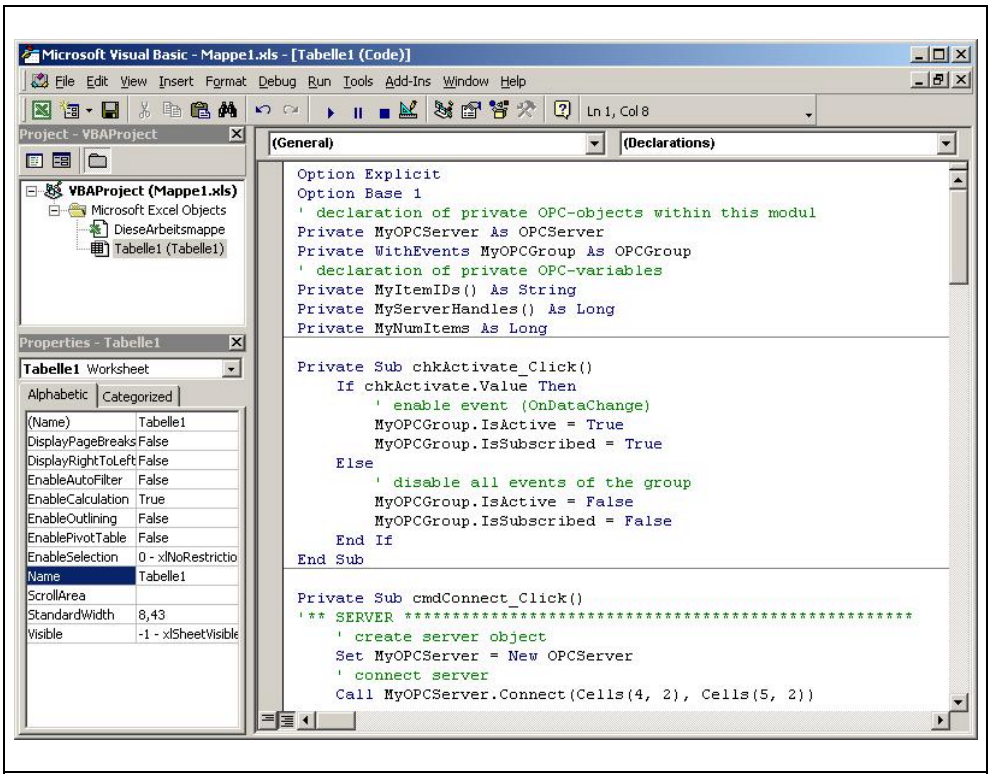

Figure 2-3 Visual Basic Editor

Automation Interface

To use the object and methods of OPC Automation Interface the Automation DLL (Dynamic Link Library) must be referenced. Therefore the DLL must be selected in the Visual Basic Editor using **Tools → References...**.

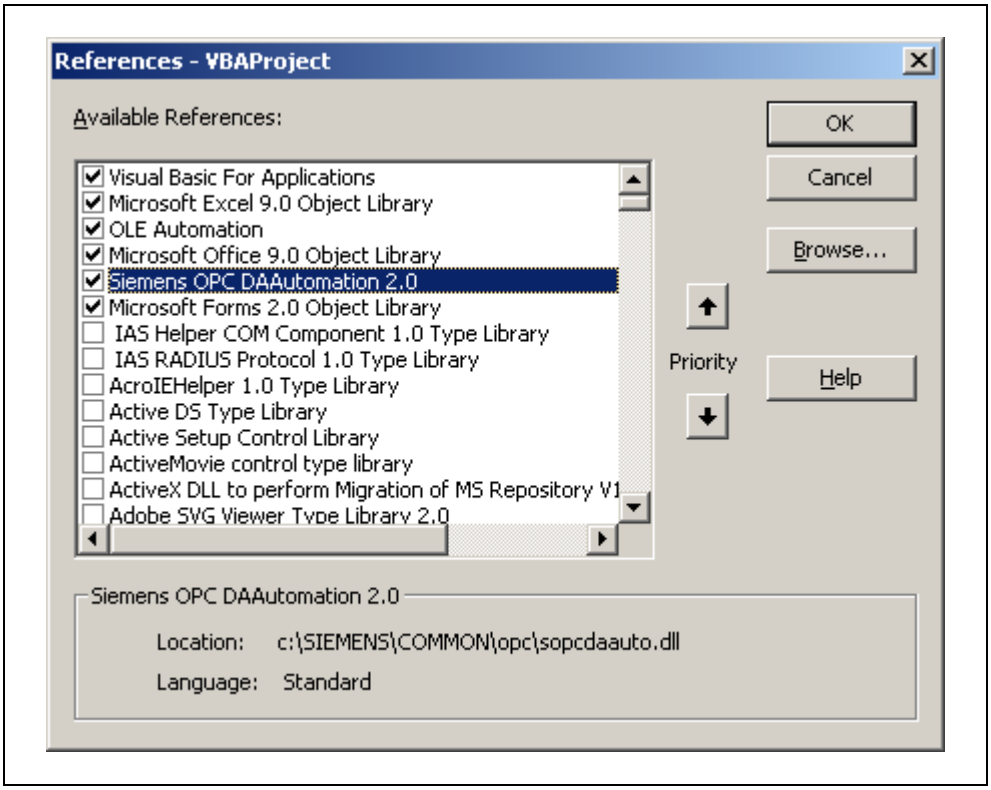

Figure 2-4 Automation Interface

Note

The Automation DLL must always be on the computer where the client application is executed. Especially when the Client runs on a machine where no OPC Server was installed before these DLL will be missing.

Object Catalog

As soon as the Automation DLL is referenced, the objects and methods of the OPC interface can be used. With **View → Object Catalog** the catalog is displayed. The Object Catalog shows, amongst other things, the OPC Functions and their parameters.

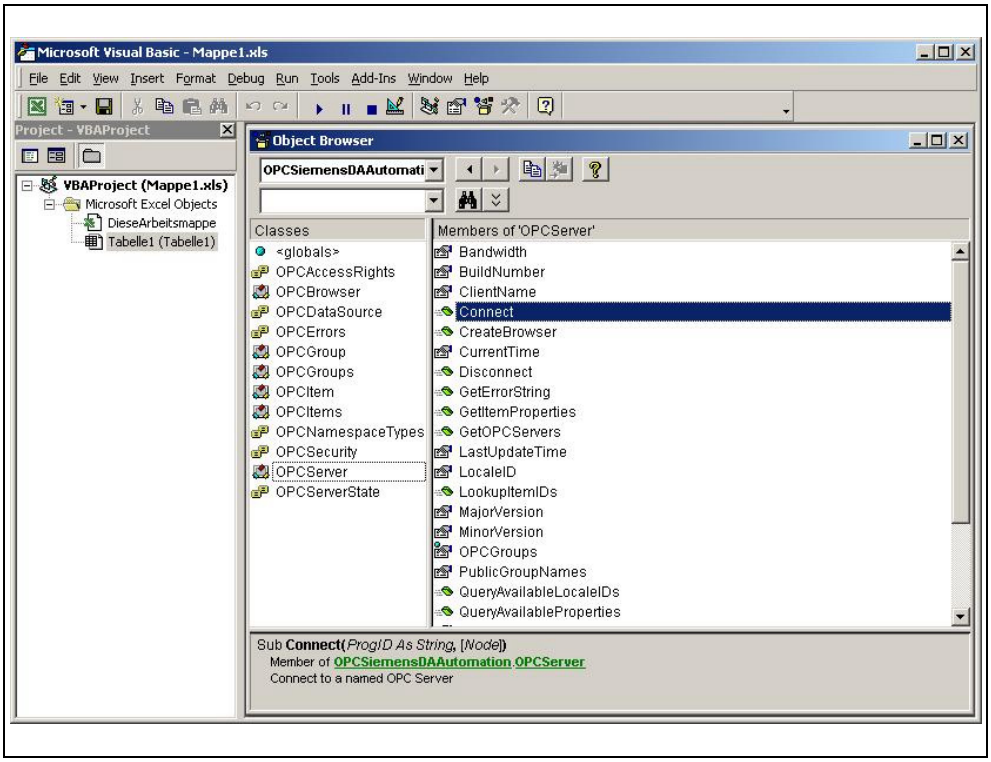

Figure 2-5 Object Catalog

2.2.1 Declarations

Compiler Commands

In the general declaration section of the code window the compliler is assigned to check all variables on explicit declaration. Additionally the definition of arrays is assigned to start from element 1 and not with 0.

Declaration of Global Variables

Objects of type „OPCServer“ and „OPCGroup“ are declared. For the object „OPCGroup“ the key word „WithEvents“ enables this object to receive events, thus a callback is implemented. E.g. the event “OnDataChange” from the OPC Server can be received.

```
Option Explicit
Option Base 1
' declaration of private OPC-objects within this modul
Private MyOPCServer As OPCServer
Private WithEvents MyOPCGroup As OPCGroup
' declaration of private OPC-variables
Private MyItemIDs() As String
Private MyServerHandles() As Long
Private MyNumItems As Long
```

2.2.2 Connect Server

General

The following code is implemented within the Click-Event of the Connect-Button and is executed whenever the button is clicked.

Connect to the OPC Server

With the “Set“ command the OPCServer object named “MyOPCServer“ is instantiated. If this object exists the Conect-Method is called and filled with the parameters “ProgID“ and „Node“. The “ProgID“ is the unique name of the OPC Servers (OPC.SimaticNet), “Node“ gives the computer on that the OPC Server can be found (computer name or IP-Address). The parameters are read from the spreadsheet.

Add Group

Before creating the Group the default update rate is set to 0. This forces the Server to provide the fastest update rate possible. Using the “Set“ command the return value of the “Add” method, which created the OPCGroup object, is directly assigned to “MyOPCGroup”. Than all events of this new group are deactivated.

Add Items

Before adding the tags the parameters are prepared in a loop. The ItemIDs (names if the variables) are read from the spreadsheet and the Clienthandles (index numbers within the client) are assigned. The “AddItems” method itself than adds the tags and returns Serverhandles (index numbers of the server) and error codes. The client should not manipulate the Serverhandles. They are needed in other function calls and therefore stored in a global array. The array of error codes gives an individual error for every single tag to indicate the success of adding the item.

Note

To simplify the code enhanced error handling is abandoned. However it is strongly recommended to catch faulty function calls and implement an error handling.

Implementation

```
Private Sub cmdConnect_Click()
'** SERVER ***********************************************
On Error GoTo errorconnect
    ' create server object
    Set MyOPCServer = New OPCServer
    ' connect server
    Call MyOPCServer.Connect(Cells(4, 2), Cells(5, 2))
 '** GROUP ************************************************
On Error GoTo errorgroup
    ' set fastest update rate for all groups
    MyOPCServer.OPCGroups.DefaultGroupUpdateRate = 0
    ' create group
    Set MyOPCGroup = MyOPCServer.OPCGroups.Add(Cells(7, 2))
    ' disable all events of the group
    MyOPCGroup.IsActive = False
    MyOPCGroup.IsSubscribed = False
'** ITEMS ************************************************
On Error GoTo erroritems
    MyNumItems = 4
    ReDim MyItemIDs(MyNumItems)
    ReDim MyClientHandles(MyNumItems) As Long
    Dim i As Long
    Dim Errors() As Long
    ' get the ItemIDs
    For i = 1 To MyNumItems
        MyItemIDs(i) = Cells(9 + i, 2)
        MyClientHandles(i) = i
    Next
    ' add items to the group
    Call MyOPCGroup.OPCItems.AddItems(MyNumItems,_
         MyItemIDs, MyClientHandles, MyServerHandles,_
         Errors)
    For i = 1 To MyNumItems
        If Errors(i) <> 0 Then
            Call MsgBox(MyItemIDs(i) & Chr(13) &_
                 MyOPCServer.GetErrorString(Errors(i)),_
                 vbCritical)
        End If
    Next
```

Implementation (Continuing)

```
'** SETTINGS
************************************************
    ' setting the buttons
    cmdConnect.Enabled = False
    cmdDisconnect.Enabled = True
    cmdSyncRead.Enabled = True
    cmdSyncWrite.Enabled = True
    chkActivate.Enabled = True
Exit Sub
errorconnect:
    Call MsgBox("Error Connect:" & Chr(13) &_
         Err.Description, vbCritical)
    Exit Sub
errorgroup:
    Call MsgBox("Error AddGroup:" & Chr(13) &_
         Err.Description, vbCritical)
    Exit Sub
erroritems:
    Call MsgBox("Error AddItems:" & Chr(13) &_
         Err.Description, vbCritical)
End Sub
```

2.2.3 Disconnect Server

General

The following code is implemented within the Click-Event of the Disconnect -Button and is executed whenever the button is clicked.

Remove Items

To remove tags the “Remove“ method from the OPCItems collection is used. This function is called with the amount of items and the array of Serverhandles (Index number within OPC Server). The returned array of errors indicates the success of the removal.

Remove Group

The “Remove“ method from the groups collection removes the group. Only the Name of the group, given from the spreadsheet is needed. The “MyOPCGroup“ than is overwritten with “nothing” to free the memory.

Disconnect Server

To disconnect from the server the “Disconnect“ method is called directly from the object “MyOPCServer”. Than the object is overwritten with “nothing to free the memory.

Note

To simplify the code enhanced error handling is abandoned. However it is strongly recommended to catch faulty function calls and implement an error handling.

Implementation

```
Private Sub cmdDisconnect_Click()
On Error GoTo errorhandler
    ' remove the items
    Dim Errors() As Long
    Call MyOPCGroup.OPCItems.Remove(MyNumItems,_
         MyServerHandles, Errors)
    Erase Errors()

    chkActivate.Value = 0
    ' remove the group
    Call MyOPCServer.OPCGroups.RemoveAll
    ' free the object
    Set MyOPCGroup = Nothing

    ' disconnect from server
    Call MyOPCServer.Disconnect
    ' free the object
    Set MyOPCServer = Nothing

    ' setting the buttons
    cmdConnect.Enabled = True
    cmdDisconnect.Enabled = False
    cmdSyncRead.Enabled = False
    cmdSyncWrite.Enabled = False
    chkActivate.Enabled = False
Exit Sub
errorhandler:
    Call MsgBox(Err.Description, vbCritical)
End Sub
```

2.2.4 Synchronous Read

General

The following code is implemented within the Click-Event of the SyncRead-Button and is executed whenever the button is clicked.

Synchronous Read

The function call for synchronous read is filled with the required input parameters. “OPCCache” instructs the server to get the values from it’s internal cache. In contrast to “OPCDevice”, where the server is forced to get the value directly from the device, reading from cache is considerably faster. The number of Items and the Serverhandles identify the tags to read.

Filling the Spreadsheet

The return values of “SyncRead“ method are filled into the cells of the spreadsheet. The quality code 0x0C (192) means “good” and indicates that the read value is valid. The OPC Server fills the array of errors. If an error occurs for an Item the returned value should not be used.

Note

To simplify the code enhanced error handling is abandoned. However it is strongly recommended to catch faulty function calls and implement an error handling.

Implementation

```
Private Sub cmdSyncRead_Click()
On Error GoTo errorhandler
    Dim Values() As Variant
    Dim Errors() As Long
    Dim Qualities() As Integer
    Dim TimeStamps() As Date
    Dim i As Long

    ' read the values
    Call MyOPCGroup.SyncRead(OPCDevice, MyNumItems,_
         MyServerHandles, Values, Errors, Qualities,_
         TimeStamps)

    ' fill values into the cells
    For i = 1 To MyNumItems
      If Errors(i) = 0 Then
        Cells(9 + i, 3) = Values(i)
        Cells(9 + i, 4) = Qualities(i)
        Cells(9 + i, 5) = TimeStamps(i)
      End If
    Next

    'free server allocated memory
    Erase Values()
    Erase Errors()
    Erase Qualities()
    Erase TimeStamps()
Exit Sub
errorhandler:
        Call MsgBox(Err.Description, vbCritical)
End Sub
```

2.2.5 Synchronous Write

General

The following code is implemented within the Click-Event of the SyncWrite -Button and is executed whenever the button is clicked.

Preparing Values

In a loop cycling all variables the cells in the spreadsheet are checked for new data. If a value was found it is assigned to the value array and the dedicated Serverhandle is copied. Thus only cells filled with data are passed to the Server.

Synchronous Write

The prepared input parameters are passed to the “SyncWrite“ method. The server returns an array of error codes to indicate the success of the write request.

Note

To simplify the code enhanced error handling is abandoned. However it is strongly recommended to catch faulty function calls and implement an error handling.

Implementation

```
Private Sub cmdSyncWrite_Click()
On Error GoTo errorhandler
    Dim Values() As Variant
    Dim HServer() As Long
    Dim NumWriteItems As Long
    Dim Errors() As Long
    Dim i As Long

    NumWriteItems = 0
    ' fill values and serverhandles
    For i = 1 To MyNumItems
        ' check for valid entry
        If Cells(9 + i, 6) <> "" Then
            ReDim Preserve Values(NumWriteItems + 1)
            ReDim Preserve HServer(NumWriteItems + 1)
            HServer(NumWriteItems + 1) = MyServerHandles(i)
            Values(NumWriteItems + 1) = Cells(9 + i, 6)
            NumWriteItems = NumWriteItems + 1
        End If
    Next

    ' write only where valid values found
    Call MyOPCGroup.SyncWrite(NumWriteItems, HServer,_
         Values, Errors)

    'free server allocated memory
    Erase Errors()
Exit Sub
errorhandler:
        Call MsgBox(Err.Description, vbCritical)
End Sub
```

2.2.6 Activate Group (DataChange)

General

The following code is implemented within the Click-Event of the Activate-Checkbox and is executed whenever the Checkbox is selected respectively unselected.

Activate Group

Depending on the Value of the checkbox (checked or not) the group is activated and the callback is advised. Only if the group is active and the object is subscribed callbacks of type “OnDataChange” can be received.

Note

The active state of a group has no effect on the other callback of this object (OnReadComplete and OnWriteComplete), however activating the OnDataChange callback.

Implementation

```
Private Sub chkActivate_Click()
    If chkActivate.Value Then
        ' enable event (OnDataChange)
        MyOPCGroup.IsActive = True
        MyOPCGroup.IsSubscribed = True
    Else
        ' disable all events of the group
        MyOPCGroup.IsActive = False
        MyOPCGroup.IsSubscribed = False
    End If
End Sub
```

DataChange Event

The body of the function is available as soon as the key word “WithEvents” is declared for the object. Whenever the OPC server detects value changes within the requested update rate a message is send to the client calling exactly this event method. The serve only send values that actually changed. These values (“ItemValues”) are filled into the cells of the spreadsheet.

Note

Within the DataChange events of the group the array of “Qualities“ should be evaluated. During runtime this is the only possibility to check the validity of the values. Here this was abandoned to simplify the code.

Implementation

```
Private Sub MyOPCGroup_DataChange(ByVal TransactionID As _
Long, ByVal NumItems As Long, ClientHandles() As Long, _
ItemValues() As Variant, Qualities() As Long, _
TimeStamps() As Date)
    Dim i As Integer

    ' fill the values in the correct cells
    For i = 1 To NumItems
        Cells(9 + ClientHandles(i), 7) = ItemValues(i)
    Next
End Sub
```

Syntax of ItemIDs

3

In this chapter the syntax of ItemIDs is described to access data in S7-200 controllers using SimaticNet.

3.1 Overview

General

An ItemID names a process variable. The syntax depends on the OPC Server. The SimaticNet OPC Server uses protocol identifier, connection name and variable name, to uniquely identify a process value.

Note

A detailed description of all parts of an ItemID following OPC Servers syntax can be found in the online help of the product SimaticNet.

SimaticNet

SimaticNet OPC Server can operate several protocols in parallel. Therefore a protocol identifier is needed within the ItemID.

S7:

SimaticNet OPC Server can connect several PLC in parallel. Therefore a connection name is needed (masking e.g. the IP-Address)

[S7-Verbindung_1]

SimaticNet OPC Server can access different data areas inside the PLC. A direct address of this data area is used.

MB1

The complete ItemID reads as follows:

S7:[S7-Verbindung_1]MB1

3.2 ItemIDs of S7-200

Overview

The following table exemplifies possible ItemIDs of S7-200 controllers.

Data Area	ItemID
MB0	S7:[S7-Verbindung_1]MB0
QB1	S7:[S7-Verbindung_1]AB1
IB1	S7:[S7-Verbindung_1]EB1
QX0.0	S7:[S7-Verbindung_1]QX0.0

Example

An OPC Item can be seen as interpretation of a process variable. Thus the client can ask for different data types for the same memory area. The Client can add one Item of type DWORD (double word) and also ask for four Items of type BYTE representing the same memory area.

Glossary and Index

4

Client	Requesting services and data
COM Port	Serial interface of PC
COM	Component Object Model
DCOM	Distributed COM
Event	Event caused by a message
Event-Method	Function executed as result of an event
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PC	Personal Computer (IBM PC)
PLC	Programmable Logic Controller
PPI	Point to Point Interconnection
Server	Providing services and data
SIMATIC	Siemens Automation Systems
SimaticNet	Network components for Siemens Automation Systems