

Communication/Networks > SIMATIC NET Industrial Communication > Industrial Ethernet > System interfacing > PG/PC > OPC Server for Industrial Ethernet > FAQ

# ▲ OPC Server -- Configuring and programming communication -- Programming and connecting OPC clients

- How do you create an OPC client with Microsoft Excel?
- How do you incorporate the S7 controls in Microsoft Excel?



How do you create an OPC client with Microsoft Excel?

## **Instructions:**

This instructions are for creating an OPC client with VBA in Microsoft Excel and how to read data from the PLC and write data to the PLC via the OPC interface.

Microsoft Excel as OPC client initiates the communication via the OPC interface and sends read/write requests to the OPC server. The OPC server executes these read/write requests.

Proceed as follows to configure Microsoft Excel as OPC client.

No.	Procedure
1	<p>Create an interface in Excel and store the separate control elements with statements.</p>  <p>( 47 KB )</p> <p>Fig. 01</p>
2	<p>Incorporate the "Siemens OPC DA Automation 2.0" components in the Visual Basic editor under "Tools-&gt;References..." so that Visual Basic knows the OPC objects.</p>
3	<p>Now all the necessary objects are created.</p> <p>Example: the "MyOPCGroup" object:</p> <p>Dim: Create a variable</p> <p>WithEvents: The object should be able to provide events (e.g. DataChange)</p> <p>MyOPCGroup: Object name</p> <p>As OPCGroup: Type of variable</p>  <p>( 17 KB )</p> <p>Fig. 02</p>
4	<p>Then the "MyOPCServer" object is provided with memory:</p> <ul style="list-style-type: none"> <li>• Set MyOPCServer = New ...</li> </ul> <p>and Microsoft Excel sets up the connection to the OPC server:</p>

- Call MyOPCServer.Connect(ServerName)

Using the function

- Cells(line, column)

you can access the data of a cell in Microsoft Excel. In this example the name of the OPC server is taken from the cell (line 4, column B).

```
Set MyOPCServer = New OPCServer
Call MyOPCServer.Connect(Cells(4, 2))
```

 ( 4 KB )

Fig. 03

- 5 Once Microsoft Excel has set up the connection to the OPC server the OPC groups are created. This is done via the Collection object.

- MyOPCServer.OPCGroups.Add(group name)

In this example the newly created group is addressed via the "MyOPCGroup" variable.

In order for the OPC group to provide events like DataChange, you must set

- MyOPCGroup.IsSubscribed = True

```
'add new OPC Group
Set MyOPCGroup = MyOPCServer.OPCGroups.Add("Scrupel")

'apply the DataChange
MyOPCGroup.IsSubscribed = True
MyOPCGroup.UpdateRate = 500
```

 ( 11 KB )

Fig. 04

- 6 Then the items are created. This is done via the Collection object.

- MyOPCItems(i) = MyOPCGroup.OPCItems.AddItem(ItemID)

In this example, the ItemIDs are stored in the following cells (line 9-12, column B). Via the function

- Cells(line, column)

you determine the ItemIDs and transfer them to the collection object "MyOPCGroup.OPCItems.AddItem".

```
'add ItemID
Set MyOPCItem = MyOPCGroup.OPCItems.AddItem(Cells(9, 2).Value)

For i = 1 To 4
  Set MyOPCItem(i) = MyOPCGroup.OPCItems.AddItem(Cells(i+8, 2).Value)
Next i
```

 ( 9 KB )

Fig. 05

- 7 With the function

- MyOPCGroup.SyncRead()

the values are read out of the PLC.

The server handles of the OPC items are transferred to this function as parameters.

```

'enum ServerHandles
Handles(1) = MyOPCItem(1).ServerHandle
Handles(2) = MyOPCItem(2).ServerHandle

Call MyOPCGroup.SyncRead(OPCIndex, 2, Handles, Values, Errors, Qual, TS)
  
```

( 9 KB )

Fig. 06

- 8 In this example the values read out of the PLC are displayed in the cells (line 9-10, column E). The quality and time stamp of the values read are shown in the cells (line 9-10, column H) and (line 9-10, column I). The function "MyOPCGroup.SyncRead()" provides this data as return parameters.

```

'fill the cells with the read values
For i = 1 To 2
  Cells(CR + 1, 5) = Values(i) 'column "read"
  Cells(CR + 1, 8) = Qual(i) 'column "quality"
  Cells(CR + 1, 9) = TS(i) 'column "timestamp"
Next i
  
```

( 11 KB )

Fig. 07

- 9 For the DataChange events of the OPC server to be processed in Microsoft Excel an event handling routine is called for the OPC group: DataChange(). This supplies a number of parameters like ItemValue (value of the OPC Item), client handle etc.

Via the DataChange event you receive the data when the value of the OPC item changes. I

The current value of the OPC item is shown in the cells (line 9-12, column D).

```

Private Sub MyOPCGroup_DataChange(ByVal TransactionID As Long, _
    ByVal Handle As Long, _
    ByVal ItemValue As Long, _
    ByVal Qual As Long, _
    ByVal TS As Long, _
    ByVal ItemName As String)
    'fill the cells with the read values
    For i = 1 To Handles
        Cells(CR + 1, 4) = ItemValue(i)
    Next i
End Sub
  
```

( 19 KB )

Fig. 08

- 10 If you enter a value in the cell (line 9-13, column F) in Excel, the OPC item should be written. A value is written to the controller via

- MyOPCGroup.SyncWrite()

The values written to the PLC and the server handle are transferred to this function as parameters.

```

'enum ServerHandles
Handles(1) = MyOPCItem(1).ServerHandle
Handles(2) = MyOPCItem(2).ServerHandle

'fill the values which are written into the following array (same index, same PL)
'also values should be written into PLC

For i = 1 To 2
  Values(i) = Cells(CR + 1, 6)
  Qual(i) = 0
Next i

Call MyOPCGroup.SyncWrite(Handles, Values, Errors)
  
```

( 23 KB )

Fig. 09

- 11 The function GetQualityText supplies an error message as string for a predefined error code.

Private Function GetQualityText(Quality) As String

Select Case Quality

Case 0: GetQualityText = "BAD"

```

Case 64: GetQualityText = "UNCERTAIN"
Case 192: GetQualityText = "GOOD"
Case 8:  GetQualityText = "NOT_CONNECTED"
Case 13: GetQualityText = "DEVICE_FAILURE"
Case 16: GetQualityText = "SENSOR_FAILURE"
Case 20: GetQualityText = "LAST_KNOWN"
Case 24: GetQualityText = "COMM_FAILURE"
Case 28: GetQualityText = "OUT_OF_SERVICE"
Case 132: GetQualityText = "LAST_USABLE"
Case 144: GetQualityText = "SENSOR_CAL"
Case 148: GetQualityText = "EGU_EXCEEDED"
Case 152: GetQualityText = "SUB_NORMAL"
Case 216: GetQualityText = "LOCAL_OVERRIDE"
Case Else: GetQualityText = "UNKNOWN ERROR"
End Select

```

End Function

12 Finally, the connection to the OPC server is cleared down.

```

'disconnect the OPC Server
Call MyOPCServer.Disconnect
Set MyOPCServer = Nothing

```


 ( 5 KB )

Fig. 10

Here you can download the relevant Visual Basic program for OPC client programming in Microsoft Excel.



OPCClient.zip ( 17 KB )

#### Important:

The sample program is freeware. Any user can use, copy and forward this program FREE OF CHARGE. The authors and owners of this program take no responsibility whatsoever for the functionality and compatibility of this software. Use of the software is at the user's own risk. Since this software is free of charge, there is no warranty nor entitlement to error correction and support.

Furthermore, Entry ID 23829402 includes an application that shows a very simple procedure for creating individual visualization interfaces using Microsoft Excel and VBA.

#### Part number:

6GK1706-0HB63-3AA0 IE PN CBA OPC-Server Version 2005

6GK1706-0XW63-3AA0 IE DX OPC-SERVER Version 2005

**Entry ID:**24299099 **Date:**12/27/2006