

How to...

## echochange

Setting up data exchange between a Siemens S7-300 and an Allen-Bradley ControlLogix, using Siemens SEND/RECEIVE functions and Allen-Bradley CIP Generic messages



Version: E-032014-01

## **Disclaimer of liability**

The information contained in these instructions corresponds to the technical status at the time of printing of it and is passed on with the best of our knowledge. The information in these instructions is in no event a basis for warranty claims or contractual agreements concerning the described products, and may especially not be deemed as warranty concerning the quality and durability pursuant to Sec. 443 German Civil Code. We reserve the right to make any alterations or improvements to these instructions without prior notice. The actual design of products may deviate from the information contained in the instructions if technical alterations and product improvements so require.

It may not, in part or in its entirety, be reproduced, copied, or transferred into electronic media.

### **Softing Industrial Automation GmbH**

Richard-Reitzner-Allee 6  
85540 Haar / Germany  
Tel: + 49 89 4 56 56-0  
Fax: + 49 89 4 56 56-488  
Internet: <http://industrial.softing.com>  
Email: [info.automation@softing.com](mailto:info.automation@softing.com)  
Support: [support.automation@softing.com](mailto:support.automation@softing.com)

The latest version of this manual is available in the Softing download area at: <http://industrial.softing.com>.

# Table of Contents

<b>Chapter 1</b>	<b>Introduction.....</b>	<b>5</b>
<b>Chapter 2</b>	<b>RSLogix 5000- create tags to send and receive data.....</b>	<b>6</b>
<b>Chapter 3</b>	<b>NetCon echo – echochange network settings .....</b>	<b>8</b>
<b>Chapter 4</b>	<b>Create data blocks to send and receive data .....</b>	<b>9</b>
<b>Chapter 5</b>	<b>RSLogix 5000 – send data to echochange.....</b>	<b>12</b>
<b>Chapter 6</b>	<b>STEP 7 – configure send and receive connections.....</b>	<b>18</b>
<b>Chapter 7</b>	<b>STEP 7 – program communications.....</b>	<b>28</b>
<b>Chapter 8</b>	<b>NetCon echo - establish the connections.....</b>	<b>45</b>

This page is intentionally left blank.

# 1 Introduction

This document provides step by step instructions on how to establish data exchange between a Siemens S7-300 PLC and an Allen-Bradley ControlLogix PLC, using a Softing echochange protocol converter.

## General description of the presented example

- A ControlLogix 1756-ENBT or 1756-EN2T communications processor is used
- The S7-300 communicates raw data using the TCON, TSEND and TRCV functions
- The ControlLogix uses CIP Generic messages to send data
- The S7-300 DB2 data block (**Data\_to\_CLX**, 1 INT) is transferred onto the **Data\_from\_S7\_raw** ControlLogix tag
- The **Data\_to\_S7** ControlLogix tag is transferred onto the S7-300 DB1 data block (**Data\_from\_CLX**, integer array of 50 elements)

## Software used

- Siemens SIMATIC Step 7 V5.4 SP5 with Open Communication Wizard V2.3 (<http://support.automation.siemens.com/WW/view/en/25209116>)
- Allen-Bradley RSLogix 5000 V20.01.00 (CPR 9 SR 5)
- Allen-Bradley RSLinx Classic Lite V2.59.02 (CPR 9 SR 5)
- Softing NetCon echo V4.33

## Conventions

The following conventions are used throughout Softing customer documentation:

Keys, buttons, menu items, commands and other elements involving user interaction are set in bold font and menu sequences are separated by an arrow

Open **Start** → **Control Panel** → **Programs**

Buttons from the user interface are enclosed in brackets and set to bold typeface

Press **[Start]** to start the application

Coding samples, file extracts and screen output is set in Courier font type

MaxDlsapAddressSupported=23

Filenames and directories are written in italic

Device description files are located in C:  
*\StarterKit\delivery\software\Device  
Description files*

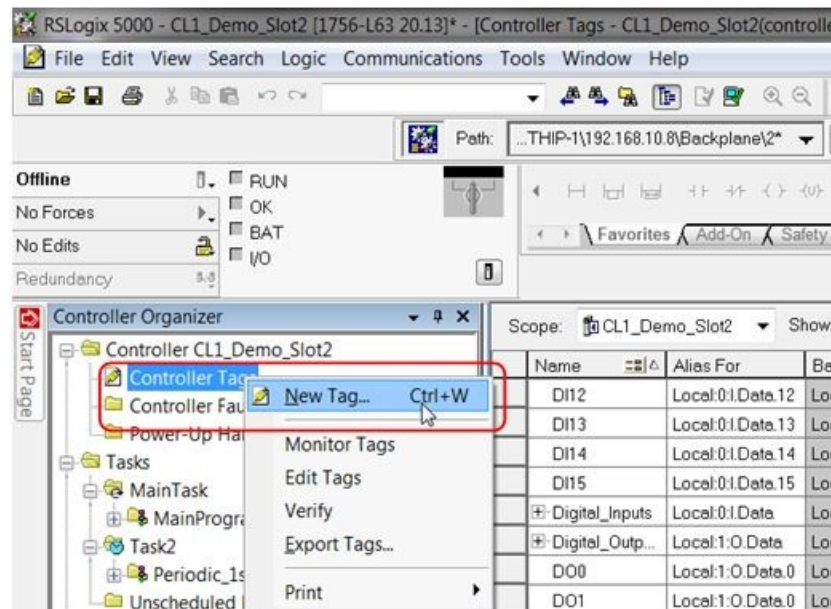


### Note

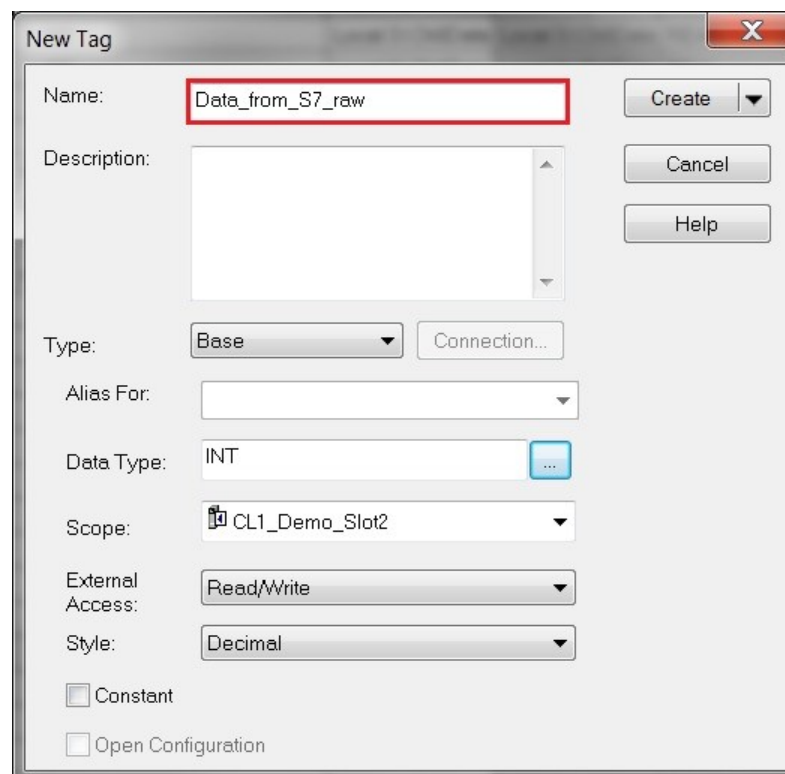
This symbol is used to call attention to notable information that should be followed during installation, use, or servicing of this device.

## 2 RSLogix 5000- create tags to send and receive data

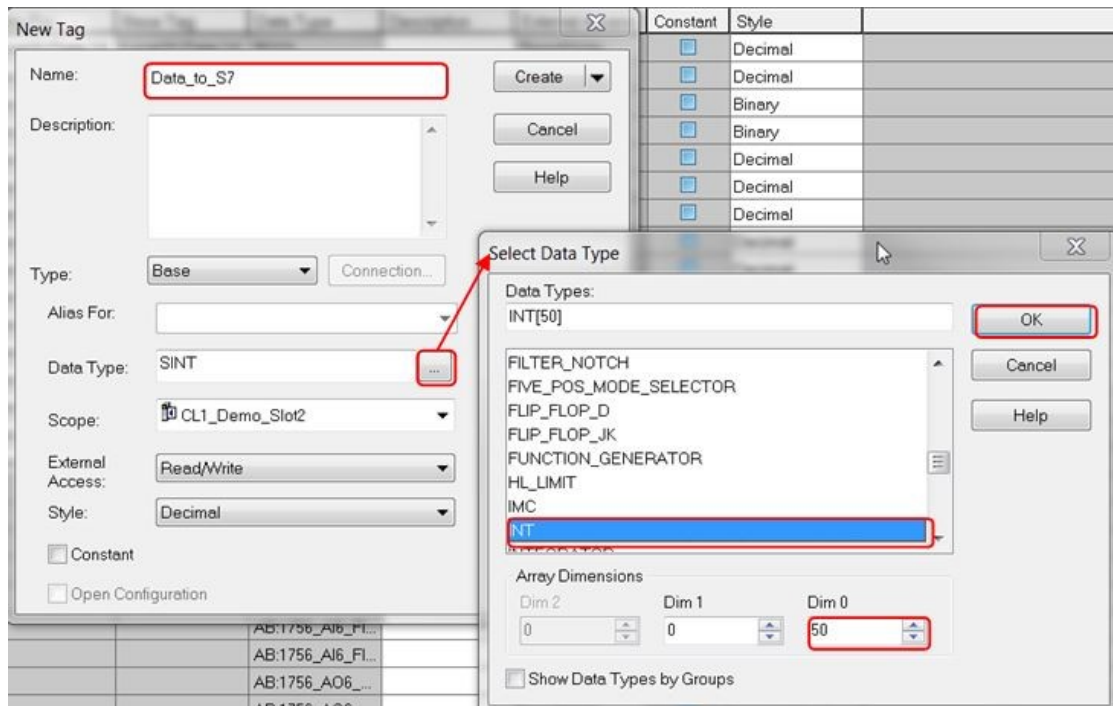
1. Create a tag to receive data from the S7-300 PLC. In this example we will create a SINT tag (short integer, 1 byte).
  - a. In the **Controller Organizer** section, right-click on **Controller Tags** and then click on **New Tag...**



- b. Enter a tag name, select data type **SINT** and click **Create**.



- Use the previous method to create a tag for the S7-300 to read. We will create an INT (integer) array with 50 elements.



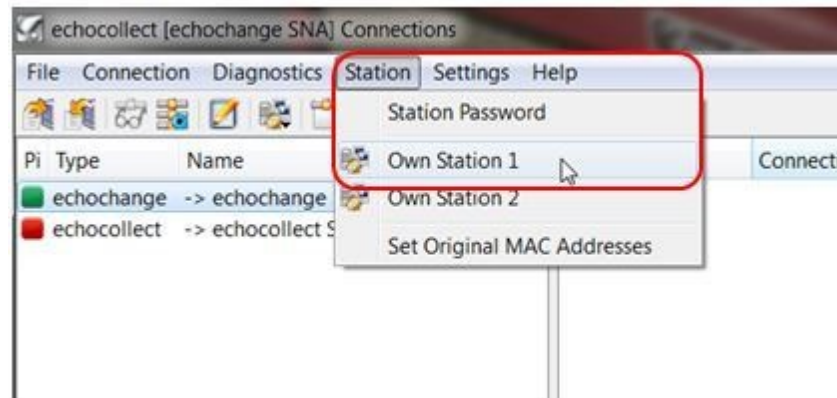
- This is how the new tags should look in the **Controller Tags** table:

Scope: CL1_Demo_Slot2 Show: All Tags									
Name	Alias For	Base Tag	Data Type	Description	External Access	Constant	Style		
Data_from_S7_raw			INT		Read/Write	<input type="checkbox"/>	Decimal		
Data_to_S7			INT[50]		Read/Write	<input type="checkbox"/>	Decimal		
DI0	Local:0:1.Data.0	Local:0:1.Data.0	BOOL		Read/Write	<input type="checkbox"/>	Decimal		
DI1	Local:0:1.Data.1	Local:0:1.Data.1	BOOL		Read/Write	<input type="checkbox"/>	Decimal		
DI2	Local:0:1.Data.2	Local:0:1.Data.2	BOOL		Read/Write	<input type="checkbox"/>	Decimal		
DI3	Local:0:1.Data.3	Local:0:1.Data.3	BOOL		Read/Write	<input type="checkbox"/>	Decimal		

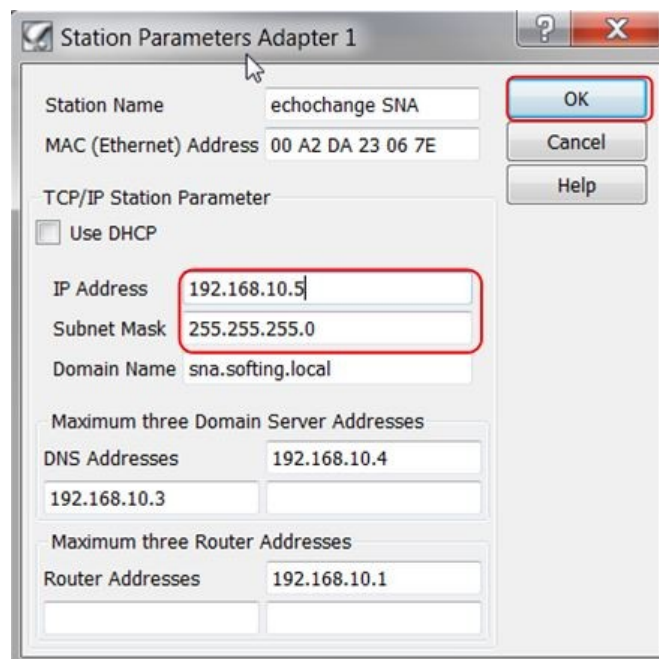
- Download the project to the ControlLogix device.

### 3 NetCon echo – echochange network settings

1. Configure the network settings of the echochange port to be used. We will use port 1 (Eth 1).
  - a. Click on **Station** → **Own Station 1**.



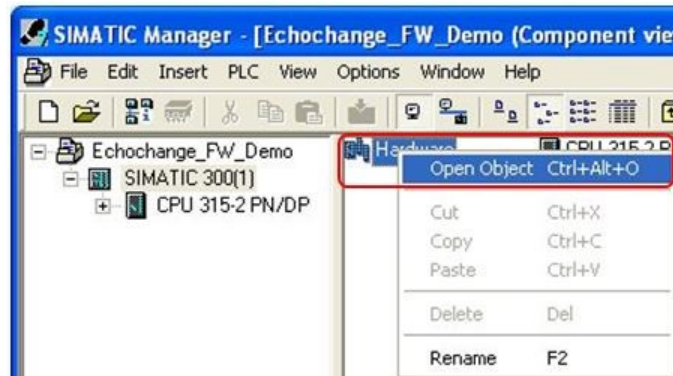
- b. As a minimum, the IP address and subnet mask are required. DNS and Router addresses may be necessary in some cases.



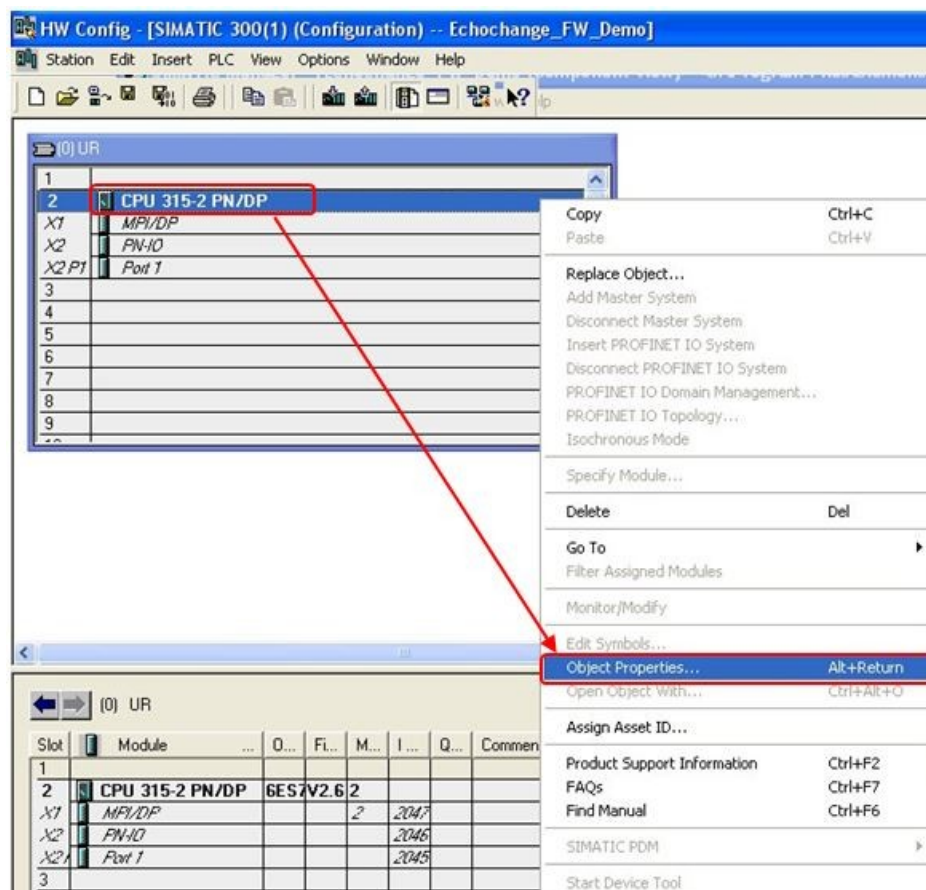


## 4 Create data blocks to send and receive data

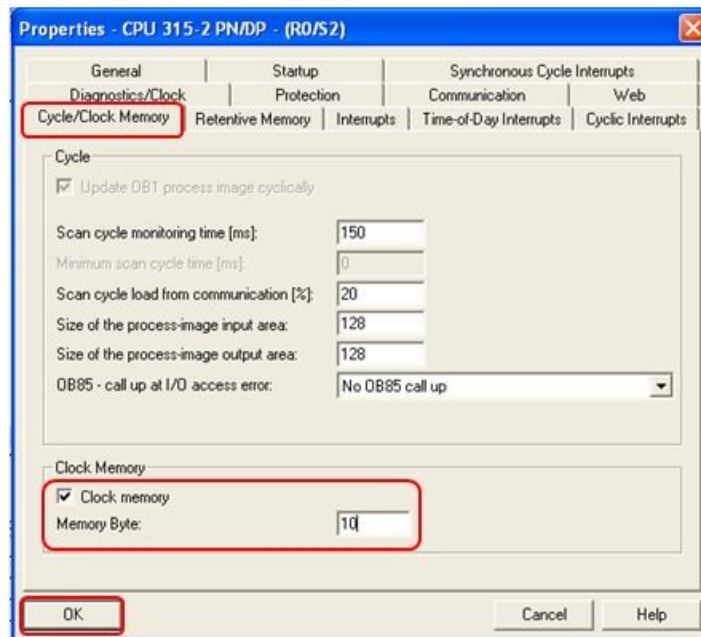
1. Open or create your project.
2. Enable the clock memory byte, as we will need a cyclic trigger.
  - a. Open the hardware configuration.



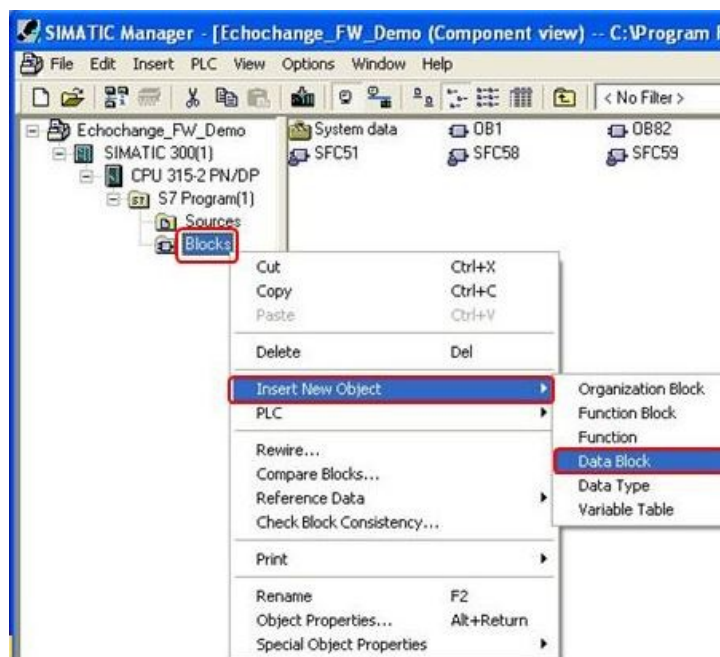
- b. Right-click on the CPU and select **Object Properties....**



- c. Select the **Cycle/Clock Memory** tab to make sure **Clock memory** is checked, and choose an unused memory byte. We use Memory Byte **10** in this example.



3. Create a data block for the data to be sent to the ControlLogix PLC.
  - a. Go back to the project window and add a new data block.



- b. Enter the **Name and type** and the **Symbolic Name**.

**Properties - Data Block**

General - Part 1 | General - Part 2 | Calls | Attributes

Name and type: DB2 Shared DB

Symbolic Name: Data\_to\_CLX

Symbol Comment:

Created in language: DB

Project path:

Storage location of project: C:\Program Files\Siemens\Step7\s7proj\Echochan

Code Interface

Date created: 10/08/2013 11:44:12 AM

Last modified: 10/08/2013 11:44:12 AM

Comment:

OK Cancel Help

- c. After clicking **[OK]**, open the new block and verify that the data type is an **INT** (integer).

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	DB_VAR	INT	0
+2.0		END_STRUCT	

- d. Close the data block configuration window and save the block.
4. Create a data block to receive data from the ControlLogix PLC, following the same method described previously. This will be an integer array of 50 elements.

**Properties - Data Block**

General - Part 1 | General - Part 2 | Calls | Attributes

Name and type: DB1 Shared DB

Symbolic Name: Data\_from\_CLX

Symbol Comment:

Created in Language: DB

Project path:

Storage location of project: C:\Program Files\Siemens\Step7\s7proj\Echochan

Date created: 10/08/2013 11:41:17 AM

Last modified: 10/08/2013 11:41:17 AM

Comment:

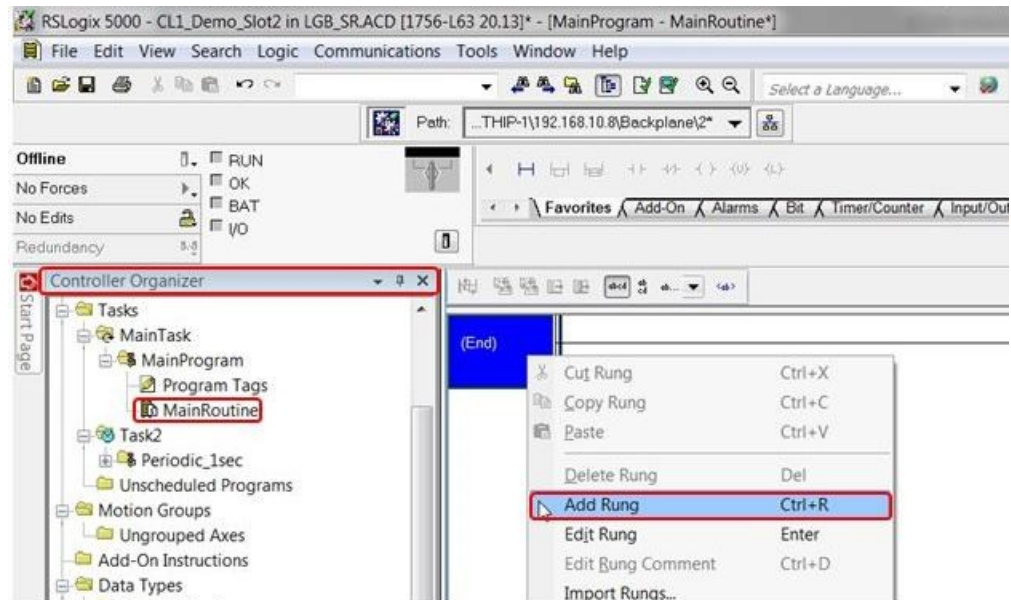
OK Cancel Help

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	ARRAY[0..50]		
+2.0		INT		
=102.0		END_STRUCT		

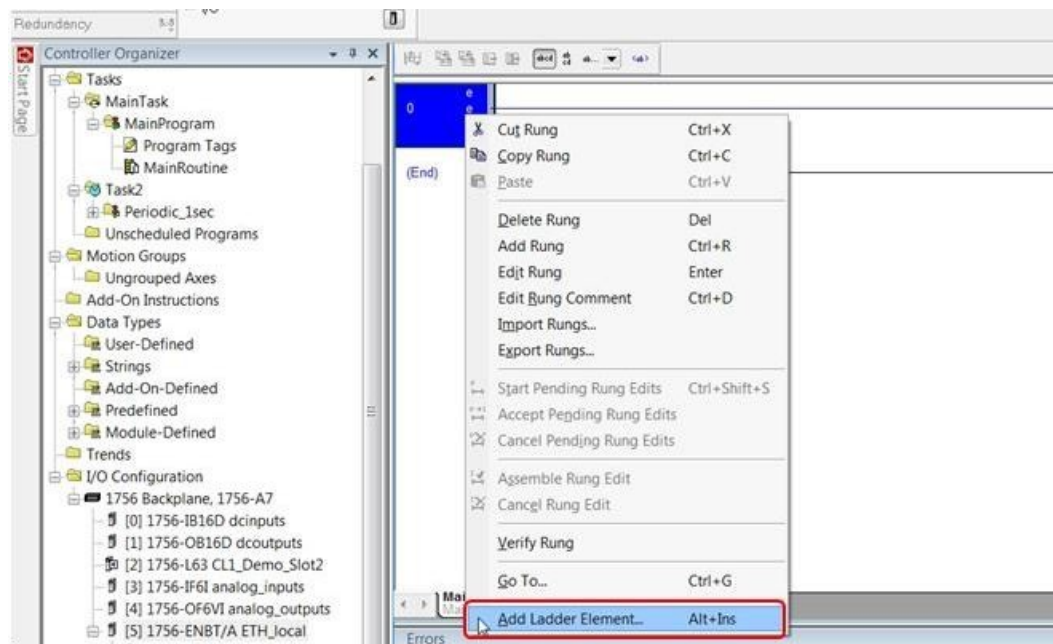
- Close the data block configuration window and save the block

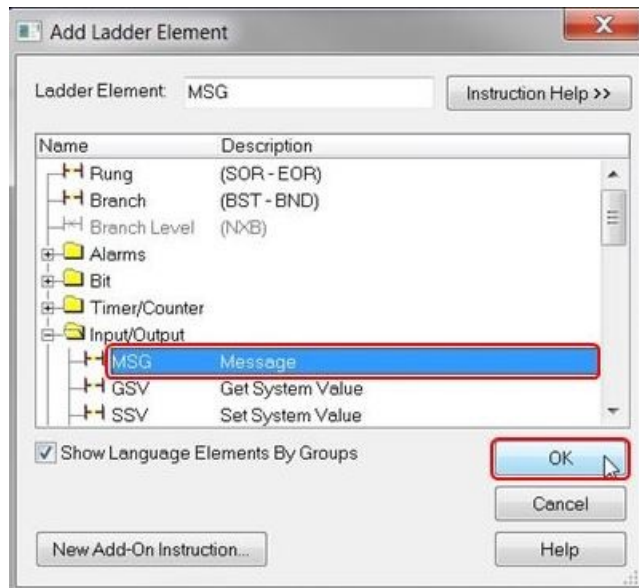
## 5 RSLogix 5000 – send data to echochange

1. Configure the send instruction to transmit data from the ControlLogix to the S7 through the echochange.
  - a. Open the **MainRoutine** from the **Controller Organizer** and add a rung to your program.

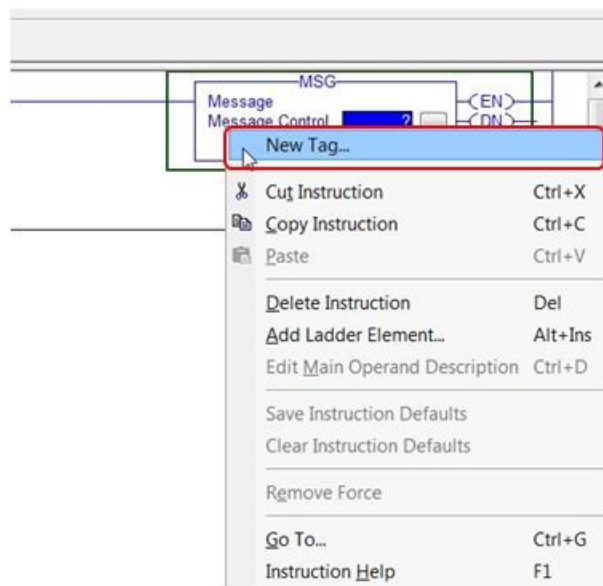


- b. Add a MSG instruction on the newly created rung.

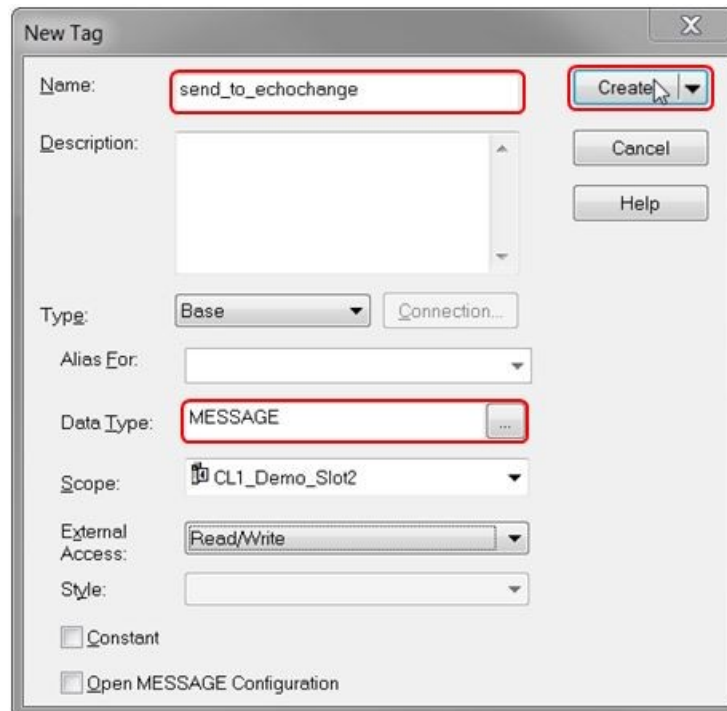




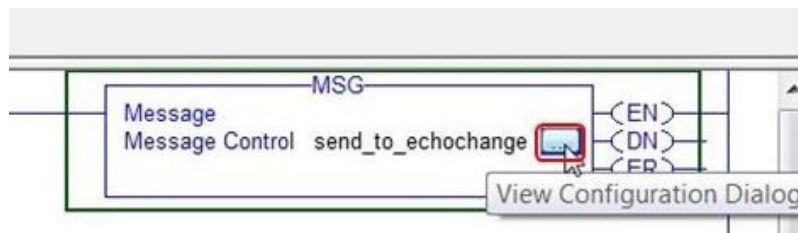
- c. On the **MSG** function block, right-click on **Message Control** to create a new tag.



- d. Name the new tag and verify its Data Type is set to **MESSAGE**.



- e. Open the **Message Configuration** dialog box by clicking on [...], as shown in the figure below:

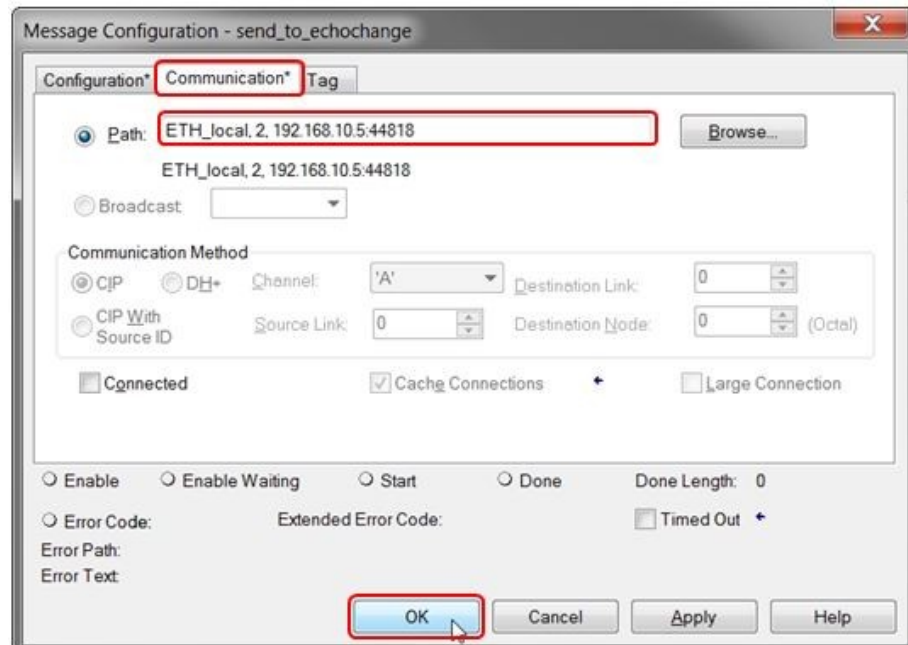




- f. In the **Configuration** tab, verify that the message type is set to **CIP Generic** and the service type to **Set Attribute Single**.
- g. Choose the tag created in chapter 2, step 2 (**Data\_to\_S7**) as the **Source Element** and specify how many bytes will be transferred. In this example we will send the complete array consisting of 50 integers, so we chose a **Source Length** of 100 bytes.
- h. In our example, we enter **1** for **Class**, **Instance** and **Attribute**. **Service Code** is set to **10** by default.

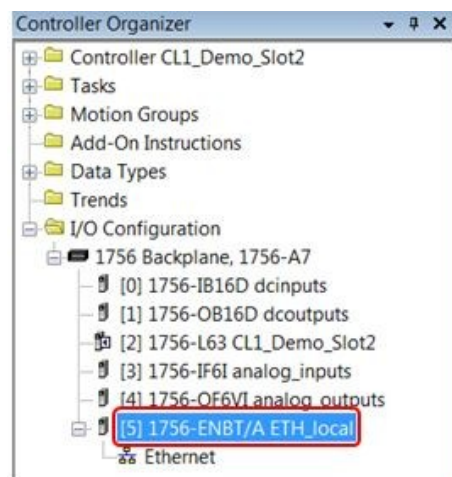
The screenshot shows the 'Message Configuration - send\_to\_echocchange' dialog box. The 'Configuration' tab is selected. The 'Message Type' is set to 'CIP Generic'. The 'Service Type' is set to 'Set Attribute Single'. The 'Source Element' is set to 'Data\_to\_S7'. The 'Source Length' is set to 100 (Bytes). The 'Service Code' is 10 (Hex), 'Class' is 1 (Hex), 'Instance' is 1, and 'Attribute' is 1 (Hex). The 'Destination Element' is empty. There is a 'New Tag...' button. At the bottom, there are radio buttons for 'Enable', 'Enable Waiting', 'Start', and 'Done'. The 'Done Length' is 0. There are also checkboxes for 'Error Code', 'Extended Error Code', and 'Timed Out'. The 'Error Path' and 'Error Text' fields are empty. The 'OK', 'Cancel', 'Apply', and 'Help' buttons are at the bottom right.

- i. Switch to the **Communication** tab and enter the path to establish the connection with the echochange. The path consists of the symbolic name of the **local module**, the **local port** (port from which the message exits), **target IP address** and **target port number**.



ETH\_local, 2, 192.168.10.5:44818

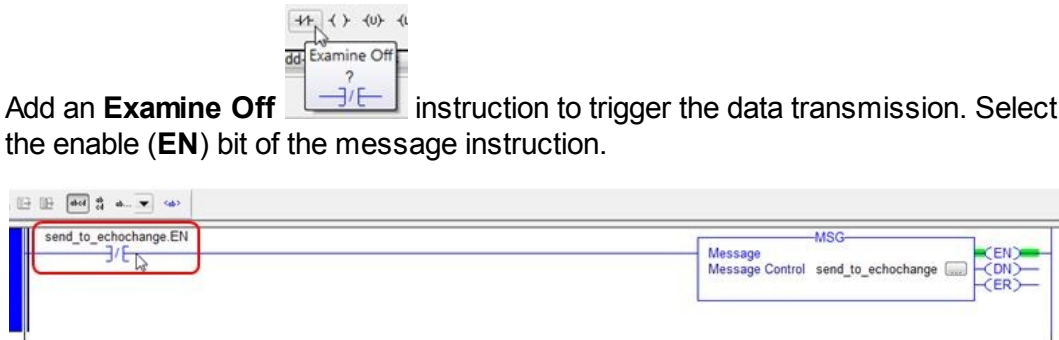
- **ETH\_local**: symbolic name of the Ethernet communications processor



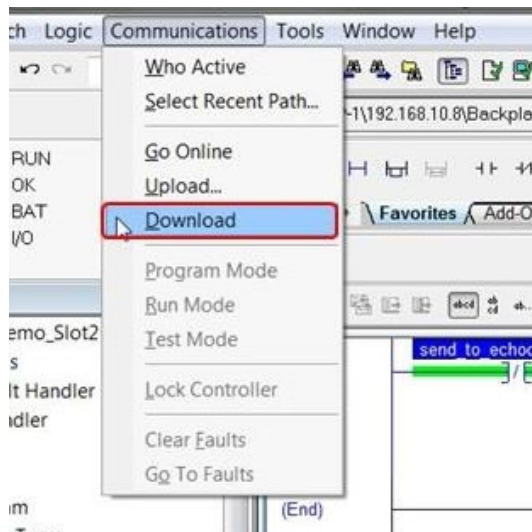
- **2**: Ethernet port from a 1756-ENET module
- **192.168.10.5**: echochange IP address
- **44818**: echochange TCP port



- j. Add an **Examine Off** instruction to trigger the data transmission. Select the enable (**EN**) bit of the message instruction.

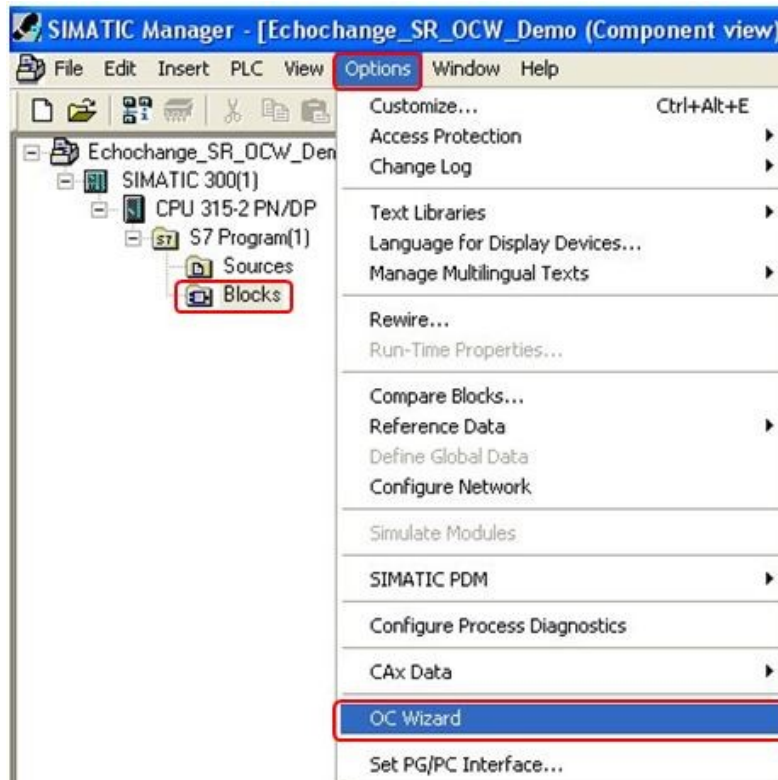


- k. Download the project to the controller.

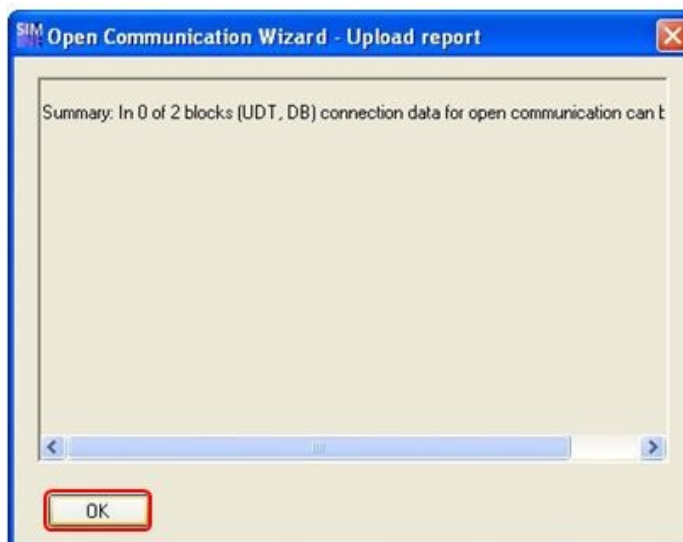


## 6 STEP 7 – configure send and receive connections

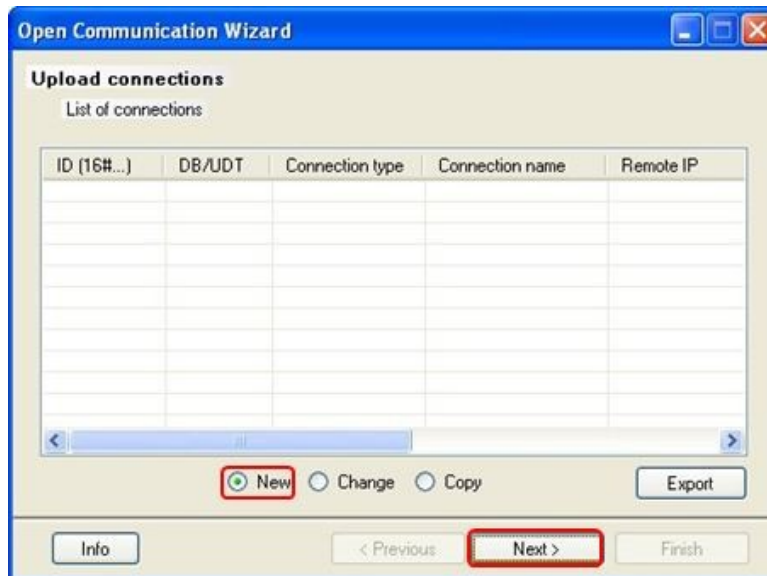
1. Click on the **Blocks** folder and start the **Open Communication Wizard**.



2. Click **[OK]** when this dialog box comes up.



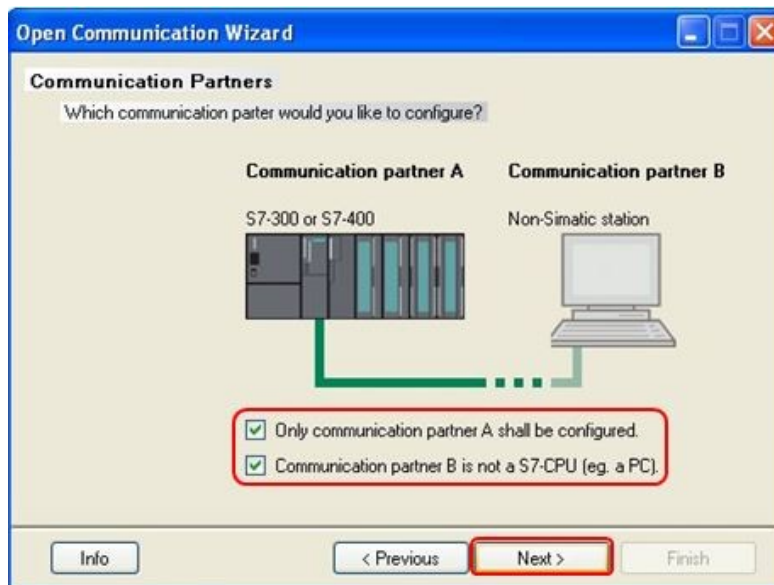
3. Make sure **New** is selected in the following dialog box and click **[Next]**:



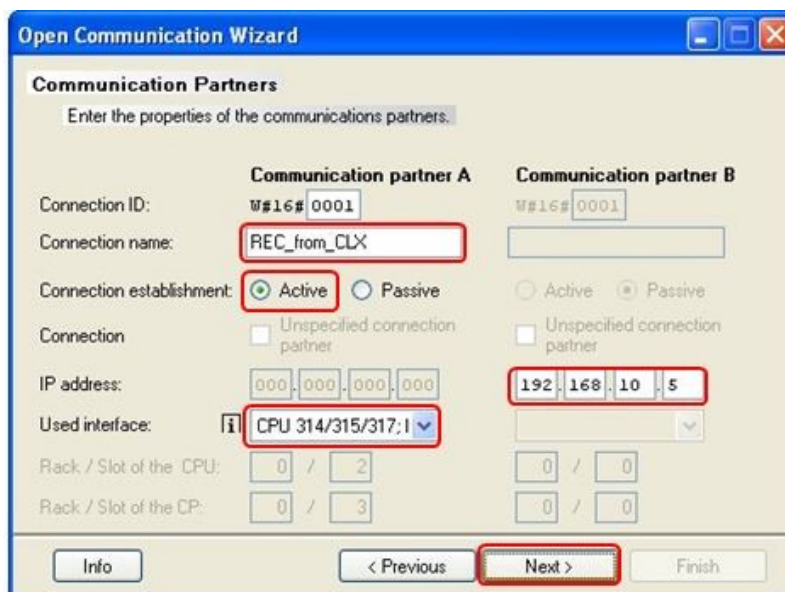
4. Select **ISO on TCP** as the protocol layer and click **[Next]**.



5. Make sure both options below are checked and click the **[Next]** button.



6. The **Connection ID** is automatically set by the wizard with the first free value. Name the connection and verify the connection establishment is set to **Active**. Select the **Used interface** (CPU type) from the dropdown menu. Under the **Communication partner B** section, enter the IP address of the echochange and click **[Next]**.



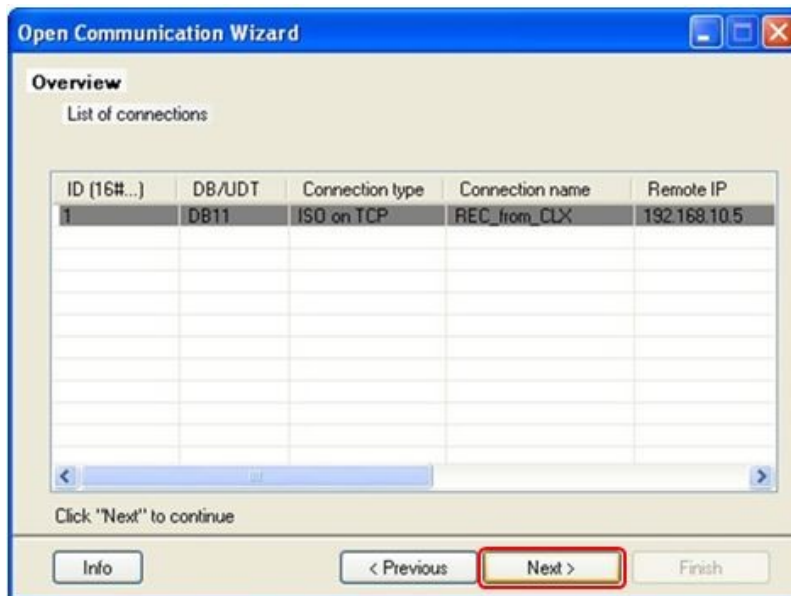
7. Enter TSAP extensions for **Communication Partner A** and **Communication Partner B**. Use the same on both for simplicity. Names can be up to 6 characters long.
8. Take note of the TSAP-IDs in the bottom half of the window. These are generated from the TSAP extensions you entered. Click **[Next]**.

The screenshot shows the 'Open Communication Wizard' dialog box, specifically the 'Connection parameters' step. The title bar reads 'Open Communication Wizard'. Below the title bar, the text 'Connection parameters' is displayed, followed by the instruction 'Enter the parameters for the connection.' The dialog is divided into two main sections: 'Communication partner A' and 'Communication partner B'. Under 'Communication partner A', the 'TSAP extension' is set to 'ASCII' (selected with a radio button) and 'Specify TSAP ID' is checked. The 'TCP-1' text is entered in the adjacent field. Under 'Communication partner B', the 'TSAP extension' is also set to 'ASCII' and 'Specify TSAP ID' is checked, with 'TCP-1' entered in the adjacent field. Below these, the 'Local TSAP-ID' section shows two checked options: 'CPU 3xx (FW < V2.7), CPU 4xx (FW < V5.2)'. The 'Communication partner A' field displays the generated TSAP-ID '80025443502D31' with a '(7)' suffix. The 'Communication partner B' field displays the generated TSAP-ID '5443502D31' with a '(5)' suffix. At the bottom, there are buttons for 'Info', '< Previous', 'Next >', and 'Finish'. The 'Next >' button is highlighted with a red rectangle.

9. Choose a free data block (DB11 in our example) where the wizard will store the communication parameters and click **[Next]**.

The screenshot shows the 'Open Communication Wizard' dialog box, specifically the 'Choose destination project' step. The title bar reads 'Open Communication Wizard'. Below the title bar, the text 'Choose destination project' is displayed, followed by the instruction 'Select STEP 7 project, block folder and block.' The 'STEP 7 project' field contains the path 'C:\Program Files\Siemens\Step7\proj\Echoch\_2'. Below this, the 'Communication partner A' section has 'Name' set to 'DB11' (highlighted with a red rectangle) and 'Block folder' set to 'Echochange\_SR\_OCW\_Demo\SIMATIC 300(1)\CPU 315-2 PN/DP...'. The 'Communication partner B' section has empty fields for 'Name' and 'Block folder'. At the bottom, there are buttons for 'Info', '< Previous', 'Next >', and 'Finish'. The 'Next >' button is highlighted with a red rectangle.

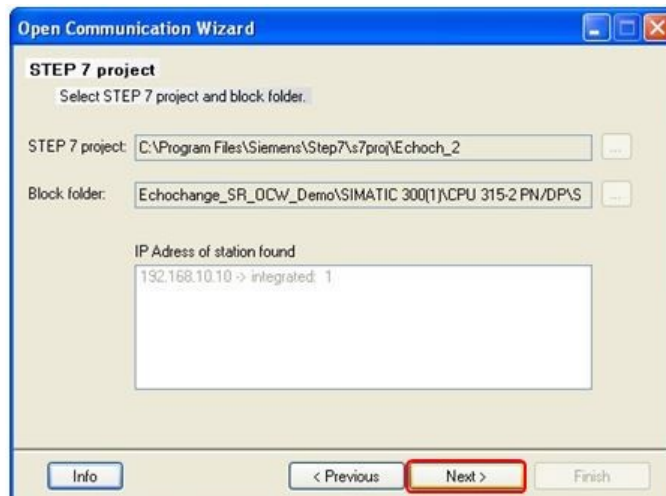
10. Click **[Next]** in the following window:



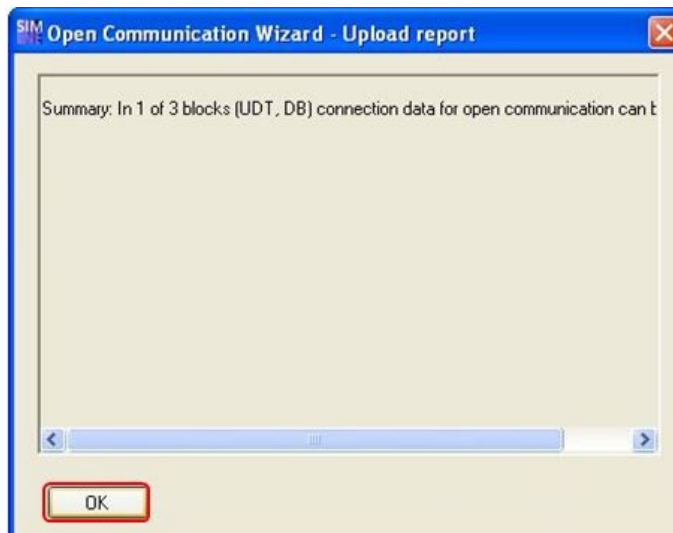
11. The wizard will compile the connection parameters. Make sure that everything finishes without errors or warnings. Since we want to configure a second connection for sending data, check **Configure another connection**, and then click **[Finish]**.



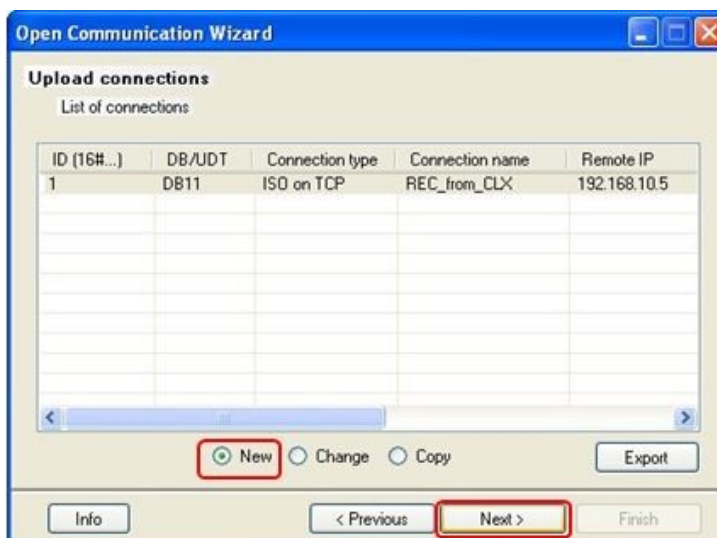
12. Configuration of the send connection starts here. Click **[Next]**.



13. Click **[OK]** to continue.



14. Make sure **New** is selected in the following dialog box and click **[Next]**.

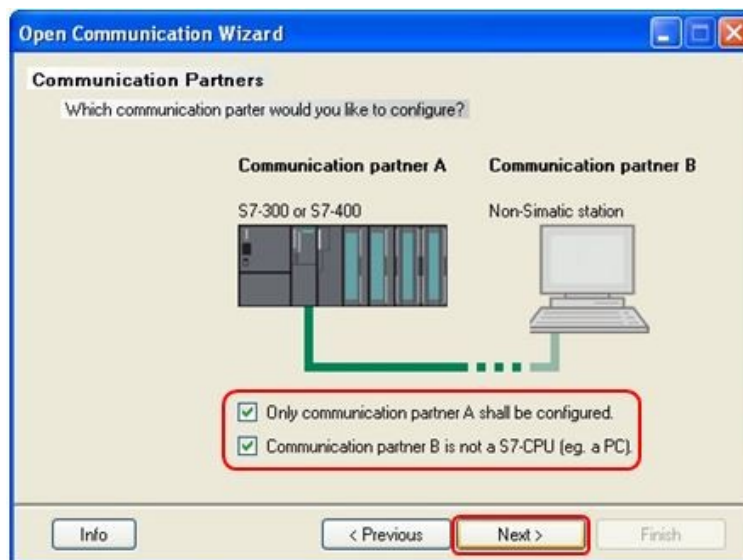




15. Select **ISO on TCP** and click **[Next]**.



16. Make sure both options below are checked and click **[Next]**.





17. The **Connection ID** is automatically set by the wizard with the first free value. Name the connection and verify the connection establishment is set to **Active**. Select the **Used interface** (CPU type) from the dropdown menu. Under the **Communication partner B** section, enter the IP address of the echochange and click **[Next]**.

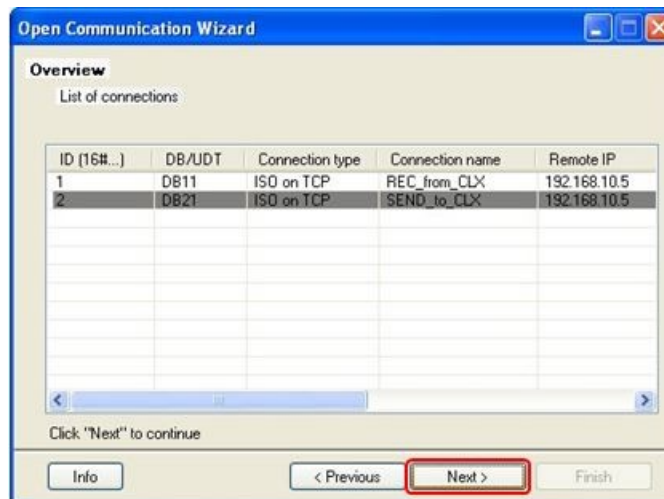
18. Enter TSAP extensions for **Communication Partner A** and **Communication Partner B**. Use the same on both for simplicity. Names can be up to 6 characters long.
19. Take note of the TSAP-IDs in the bottom half of the window. These are generated from the TSAP extensions you entered. Click **[Next]**.

20. Choose a free data block (DB21 in our example) where the wizard will store the communication parameters and click **[Next]**.



The screenshot shows the 'Open Communication Wizard' dialog box. The 'Choose destination project' section is active, with the 'STEP 7 project' field set to 'C:\Program Files\Siemens\Step7\proj\Echoch\_2'. Below this, the 'Communication partner A' section is visible, with the 'Name' field set to 'DB21' (highlighted with a red box) and the 'Block folder' field set to 'Echochange\_SR\_OCW\_Demo\SIMATIC 300(1)\CPU 315-2 PN\DP\...'. The 'Communication partner B' section is also visible but empty. At the bottom, the 'Next >' button is highlighted with a red box.

21. Click **[Next]**.



The screenshot shows the 'Open Communication Wizard' dialog box in the 'Overview' section. It displays a table titled 'List of connections' with the following data:

ID (16#...)	DB/UDT	Connection type	Connection name	Remote IP
1	DB11	ISO on TCP	REC_from_CLX	192.168.10.5
2	DB21	ISO on TCP	SEND_to_CLX	192.168.10.5

Below the table, there is a scroll bar and the text 'Click "Next" to continue'. At the bottom, the 'Next >' button is highlighted with a red box.

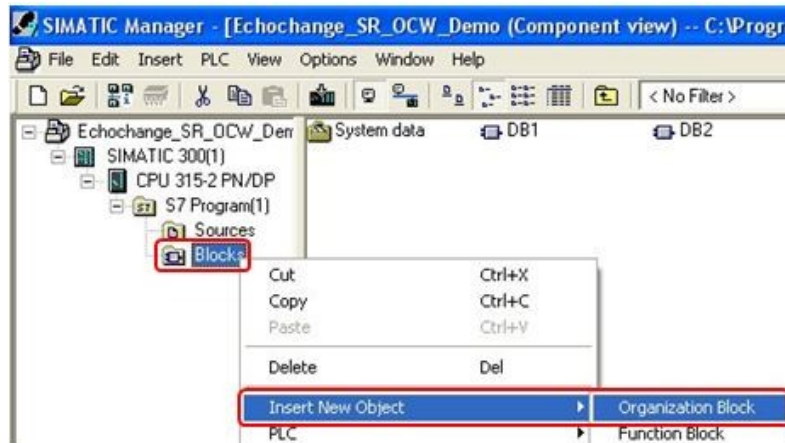
22. The wizard will compile the connection parameters. Make sure that everything finishes without errors or warnings and click **[Next]**.



23. Click **[Finish]** to close the wizard.

## 7 STEP 7 – program communications

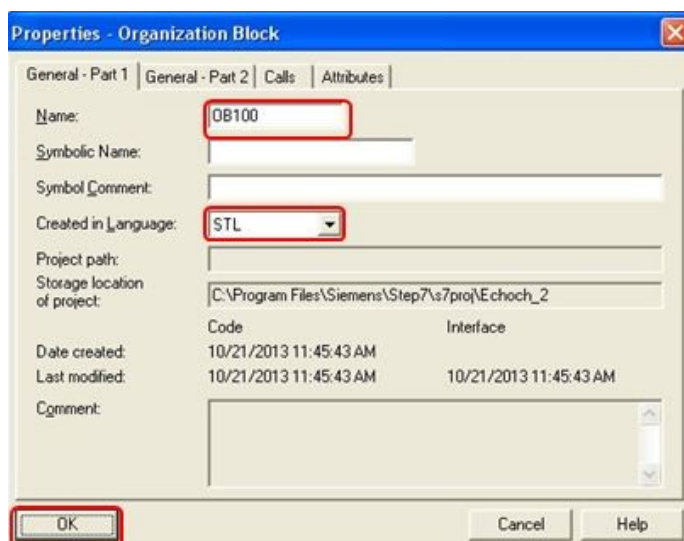
1. If you don't already have it, create OB100 (this is a special start-up OB that is executed before OB1).



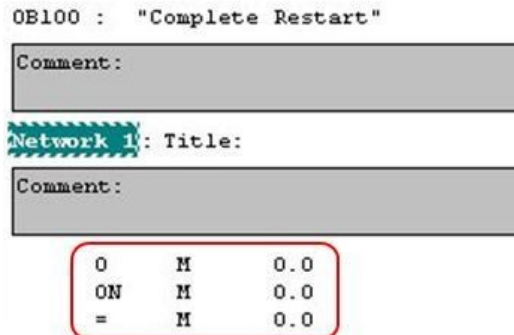
2. Type in **OB100** for the name and click **[OK]**.



Note that in our example we use STL language, but the same programming can be done in any language.



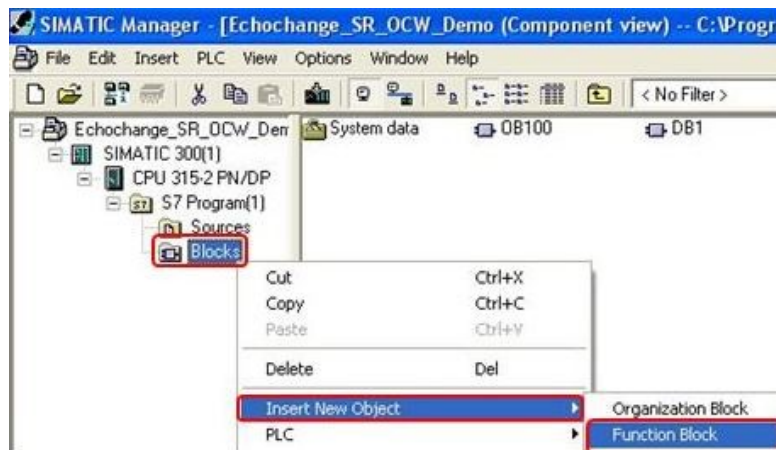
3. We want to initialize the TCON function at startup (warm restart). Open OB100 and pick a free memory bit (M0.0 in our example) to be used to trigger the connection establishment. We want to set the bit to 1 in OB100:



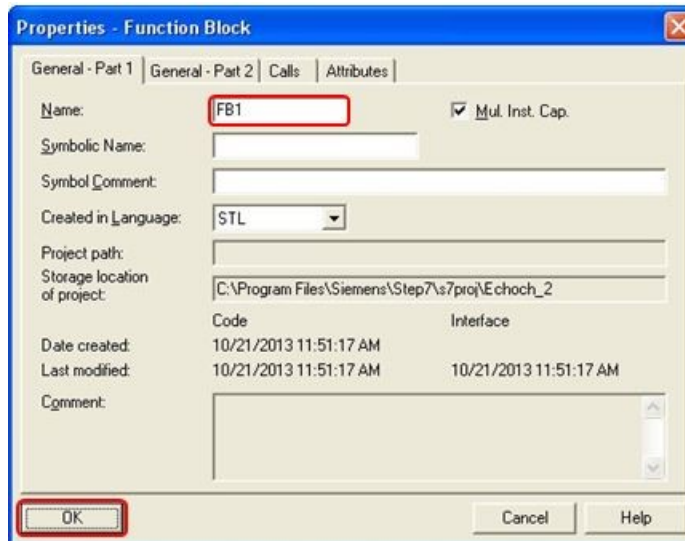
4. This is the LAD equivalent:



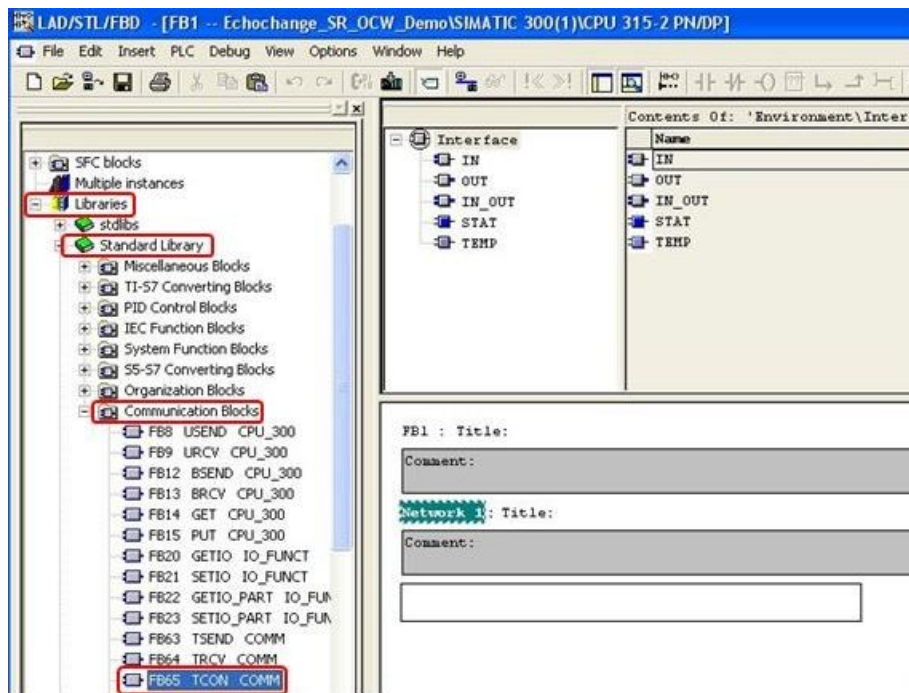
5. Save and close OB100.
6. Create a new function block for the receive connection.



7. Name the function block (**FB1** in this example) and click **[OK]**.



8. Open the newly created function block to program it. Add a call to **FB65 TCON**, found under **Libraries → Standard Library → Communication Blocks**.



9. After adding FB65 to your program, assign an unused DB number (**DB13** in this example).

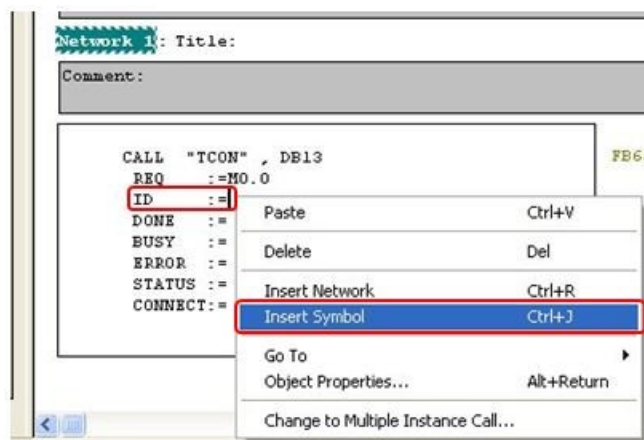


10. After hitting **Enter**, click **[Yes]** on the following dialog box:

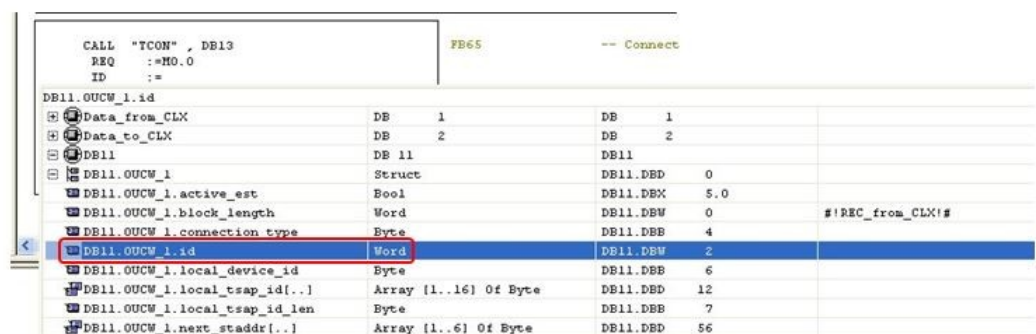


11. Set the parameters for the TCON function block.

- a. Right-click on **ID** and select **Insert Symbol**.

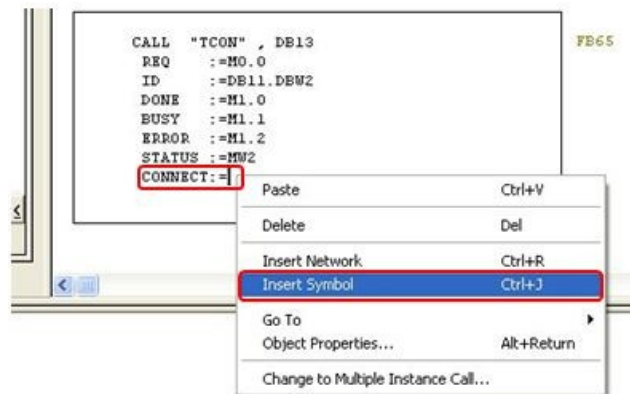


- b. Browse to the id symbol of the DB structure chosen in chapter 6 step 8. In our example it is **DB11.OUCW\_1.id**. Double-click on it.

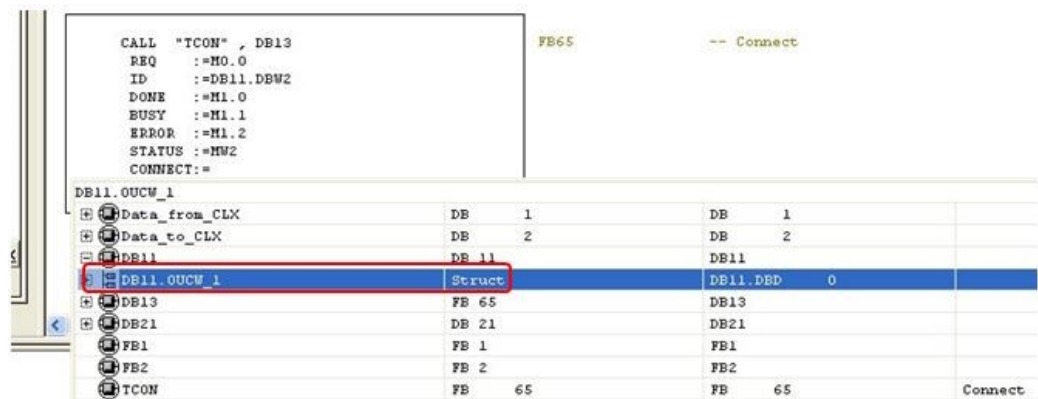




- c. Right-click on **CONNECT** and select **Insert Symbol**.



- d. Select the whole structure that was created in chapter 6 step 8 by double-clicking on it. In our example this is **DB11.OCW\_1**.



- e. For the REQ parameter, enter the memory bit we set to **1** in chapter 6 step 3 (**M0.0**). Choose any unused bit or word respectively for the output parameters DONE, BUSY, ERROR and STATUS.

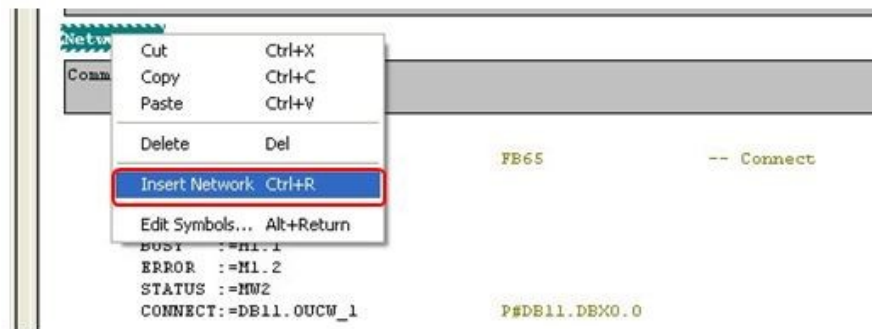


```

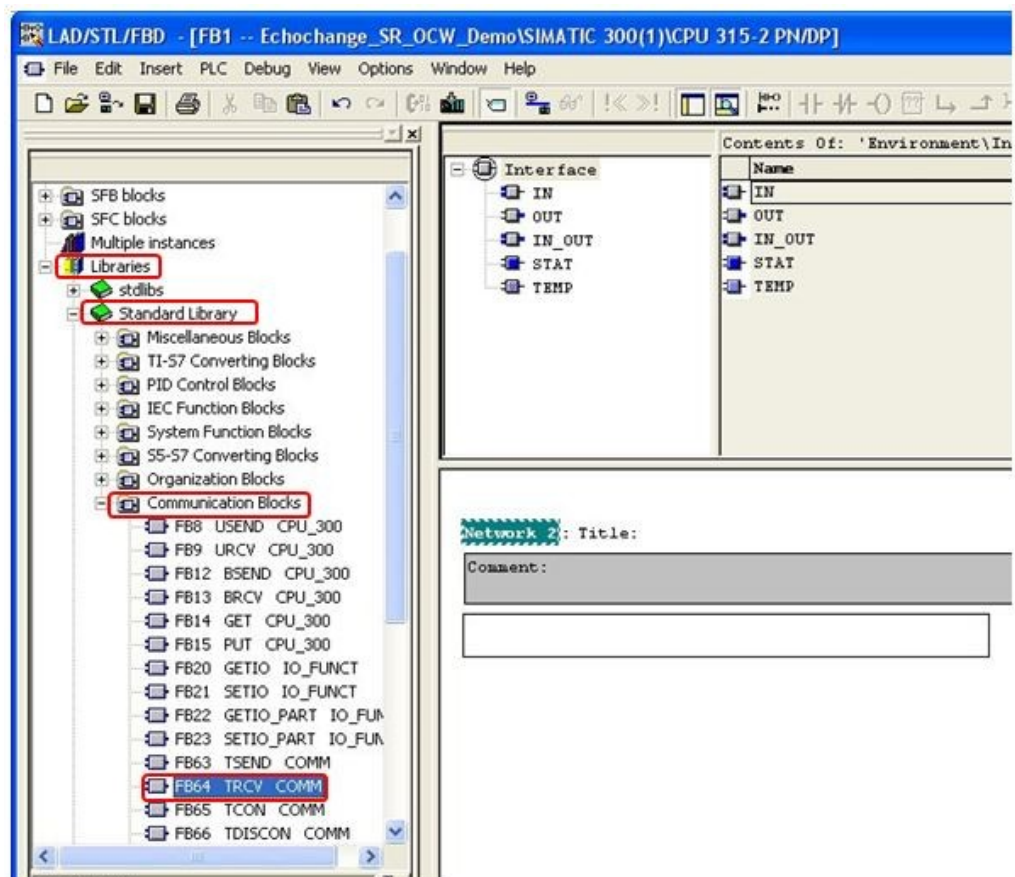
CALL "TCON", DB13
REQ      :=M0.0
ID       :=DB11.DBW2
DONE     :=M1.0
BUSY     :=M1.1
ERROR    :=M1.2
STATUS   :=MW2
CONNECT:=DB11.OCW_1
  
```



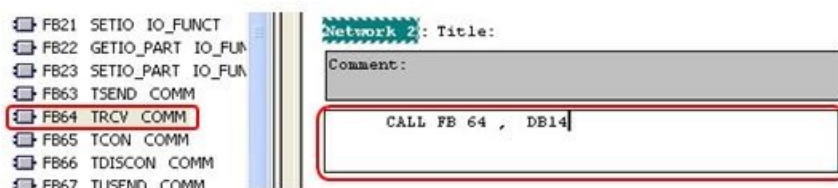
12. Call the **TRCV** function.
  - a. Insert a new network into function block FB1.



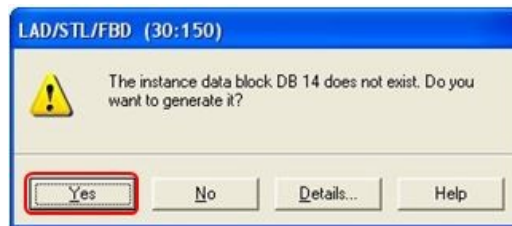
- b. Add a call to FB64 TRCV, found under **Libraries** → **Standard Library** → **Communication Blocks**.



- c. After adding FB64 to Network 2, assign an unused DB number (**DB14** in this example).

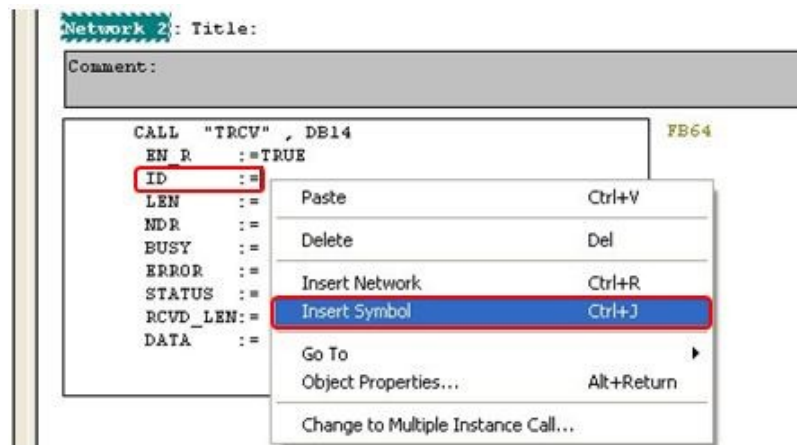


- d. After hitting **Enter**, click **[Yes]** on the following dialog box:

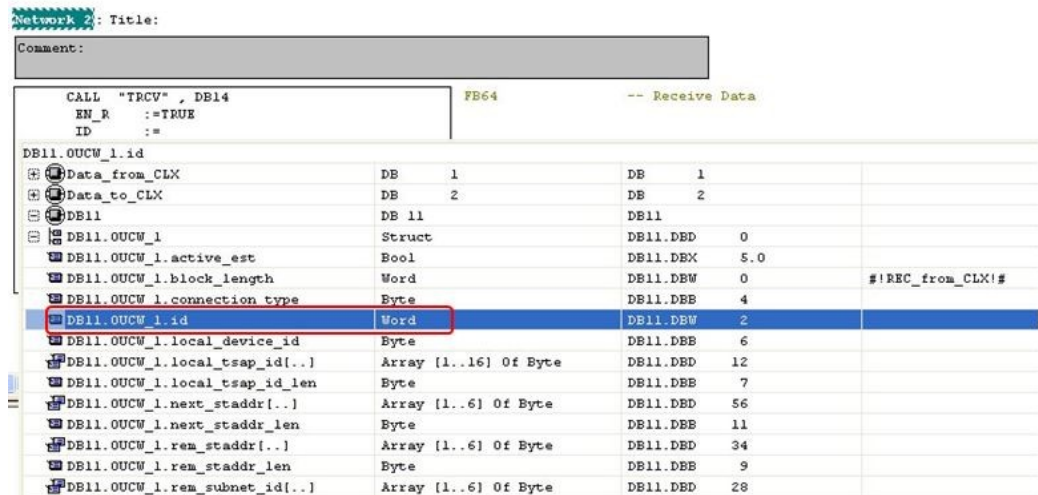


13. Set the parameters for the TRCV function block.

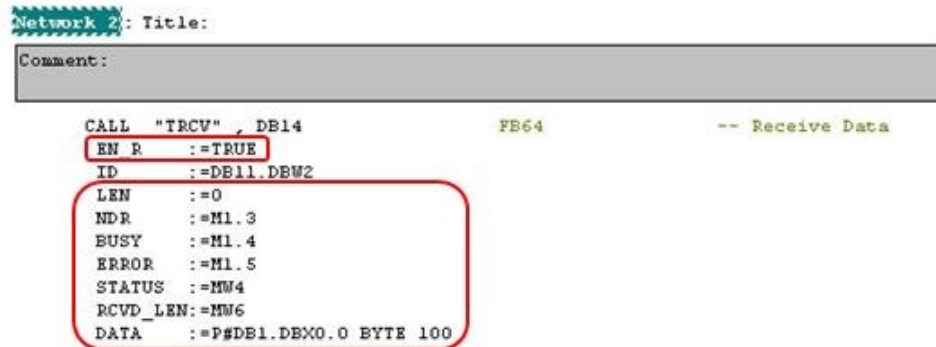
- a. Right-click on **ID** and select **Insert Symbol**.



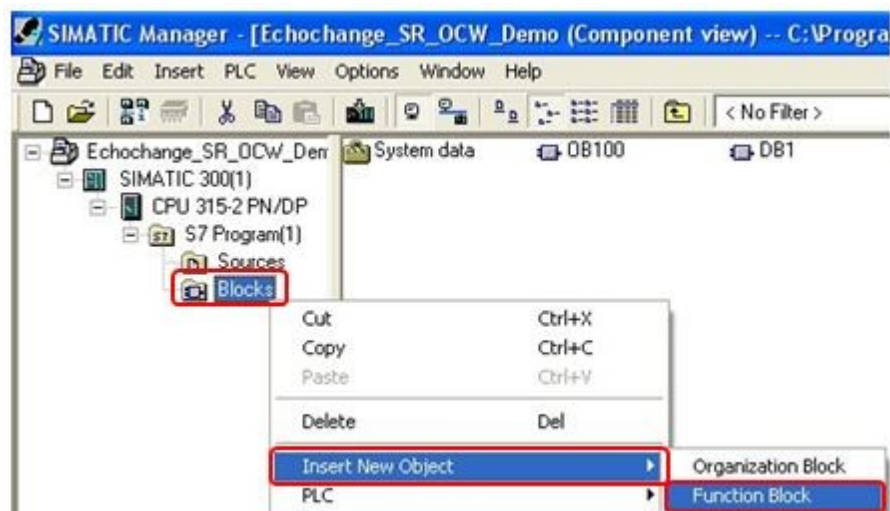
- b. Browse to the id symbol of the DB structure chosen in chapter 6 step 8. In our example it is **DB11.OCW\_1.id**. Double-click on it.



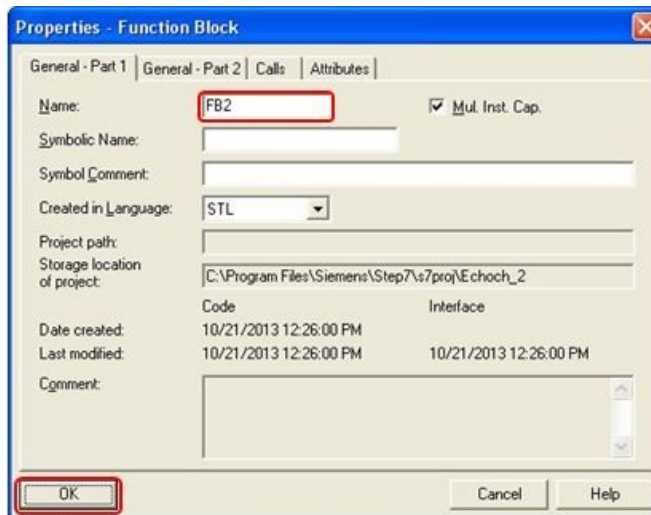
- c. Set EN\_R to **TRUE**.
- d. Set LEN to **0**.
- e. Set DATA with a pointer to the area where the received data will be stored. In chapter 4 step 4 we created DB1 for this purpose and the pointer to this area is P#DB1.DBX0.0 BYTE 100.
- f. Choose any unused bit or word respectively for the remaining output parameters NDR, BUSY, ERROR, STATUS and RCVD\_LEN.



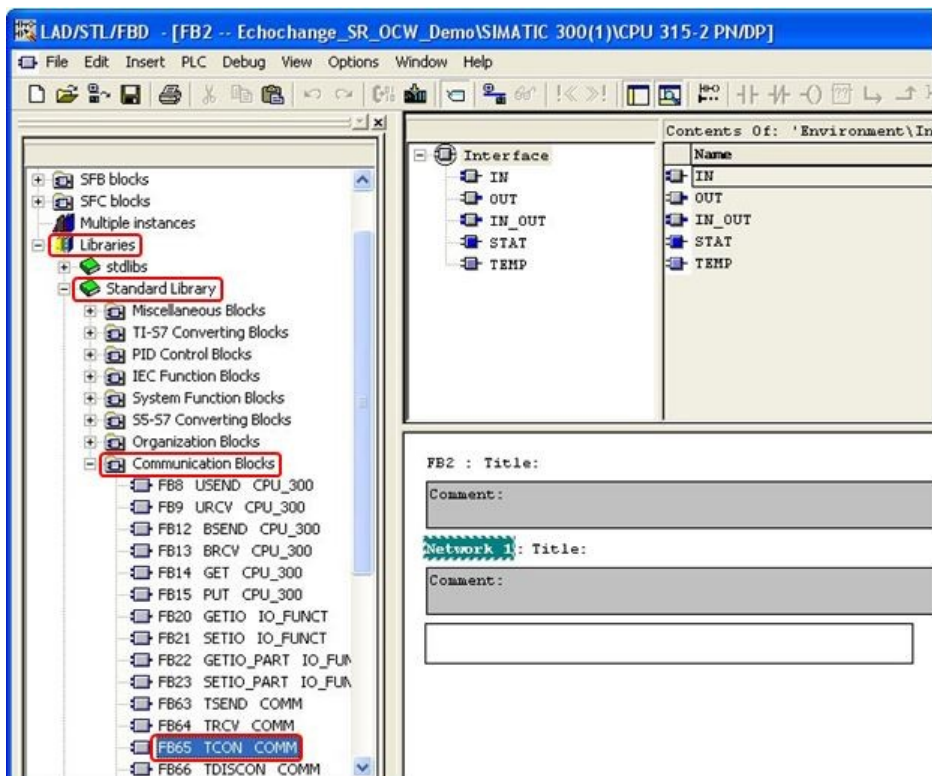
- g. Save and close FB1.
14. Create a new function block for the send connection.



15. Name the function block (**FB2** in this example) and click **[OK]**.



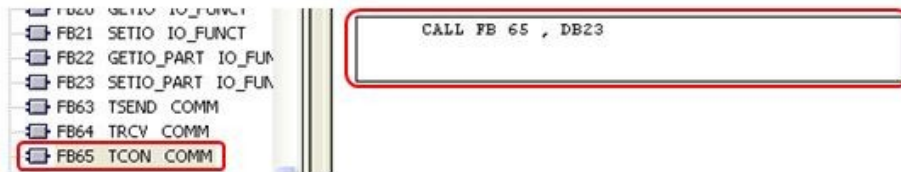
16. Open the newly created function block to program it. Add a call to FB65 TCON, found under **Libraries** → **Standard Library** → **Communication Blocks**.



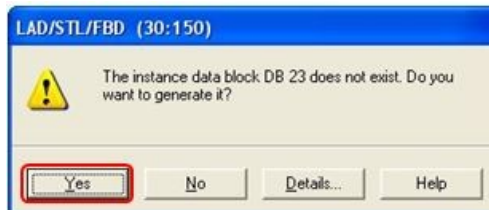
17. After adding FB65 to your program, click **[No]** on the following dialog box:



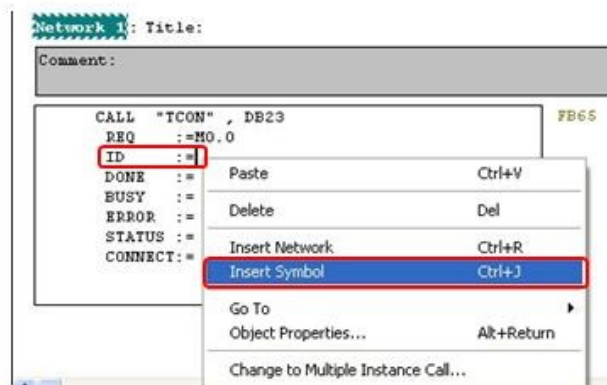
18. Assign an unused DB number (**DB23** in this example).



19. After hitting **Enter**, click **[Yes]** on the following dialog box:



20. Set the parameters for the TCON function block.
- Right-click on **ID** and select **Insert Symbol**.



- b. Browse to the id symbol of the DB structure chosen in chapter 6 step 18. In our example it is **DB21.OUCW\_1.id**. Double-click on it.

The screenshot shows the HW Config window for a station. The 'DB21.OUCW\_1' structure is expanded, and the 'DB21.OUCW\_1.id' symbol is selected. The table below shows the structure's components:

Symbol	Address	Length	Symbol	Address	Length
Data_from_CLX	DB 1	1	DB 1	1	
Data_to_CLX	DB 2	2	DB 2	2	
DB11	DB 11		DB11		
DB13	FB 65		DB13		
DB14	FB 64		DB14		
DB21	DB 21		DB21		
DB21.OUCW_1	Struct		DB21.DBD	0	
DB21.OUCW_1.active_est	Bool		DB21.DBX	5.0	
DB21.OUCW_1.block_length	Word		DB21.DBW	0	#!SEND_to_CLX!
DB21.OUCW_1.connection_type	Byte		DB21.DBB	4	
DB21.OUCW_1.id	Word		DB21.DBW	2	
DB21.OUCW_1.local_device_id	Byte		DB21.DBB	6	
DB21.OUCW_1.local_tsap_id[...]	Array [1..16] Of Byte		DB21.DBD	12	
DB21.OUCW_1.local_tsap_id_len	Byte		DB21.DBB	7	
DB21.OUCW_1.next_staddr[...]	Array [1..6] Of Byte		DB21.DBD	56	
DB21.OUCW_1.next_staddr_len	Byte		DB21.DBB	11	
DB21.OUCW_1.rem_staddr[...]	Array [1..6] Of Byte		DB21.DBD	34	
DB21.OUCW_1.rem_staddr_len	Byte		DB21.DBB	9	
DB21.OUCW_1.rem_subnet_id[...]	Array [1..6] Of Byte		DB21.DBD	28	

- c. Right-click on **CONNECT** and select **Insert Symbol**.

The screenshot shows the HW Config window with the 'CONNECT' symbol selected. A right-click context menu is open, and the 'Insert Symbol' option is highlighted.

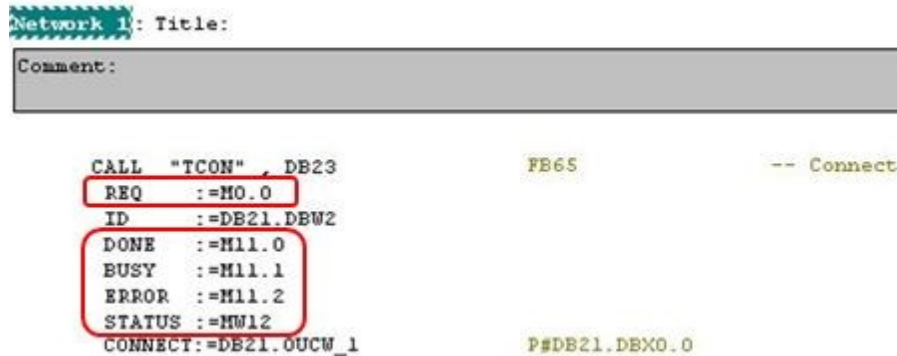
- d. Select the whole structure created in chapter 6 step 18 by double-clicking on it. In our example this is **DB21.OUCW\_1**.

The screenshot shows the HW Config window with the 'DB21.OUCW\_1' structure selected. The table below shows the structure's components:

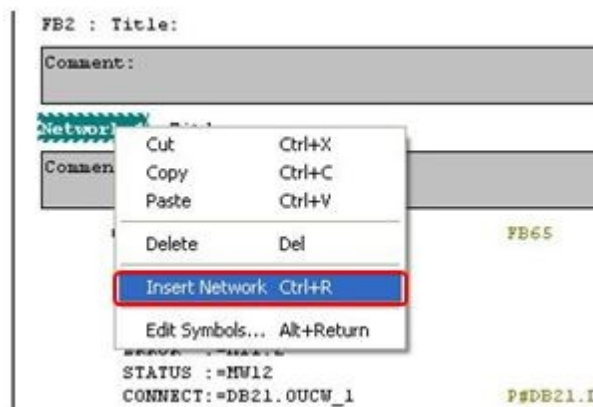
Symbol	Address	Length	Symbol	Address	Length
Data_from_CLX	DB 1	1	DB 1	1	
Data_to_CLX	DB 2	2	DB 2	2	
DB11	DB 11		DB11		
DB13	FB 65		DB13		
DB14	FB 64		DB14		
DB21	DB 21		DB21		
DB21.OUCW_1	Struct		DB21.DBD	0	
DB23	FB 65		DB23		
FB1	FB 1		FB1		
FB2	FB 2		FB2		
TCON	FB 65		FB 65		Connect
TRCV	FB 64		FB 64		Receive Data



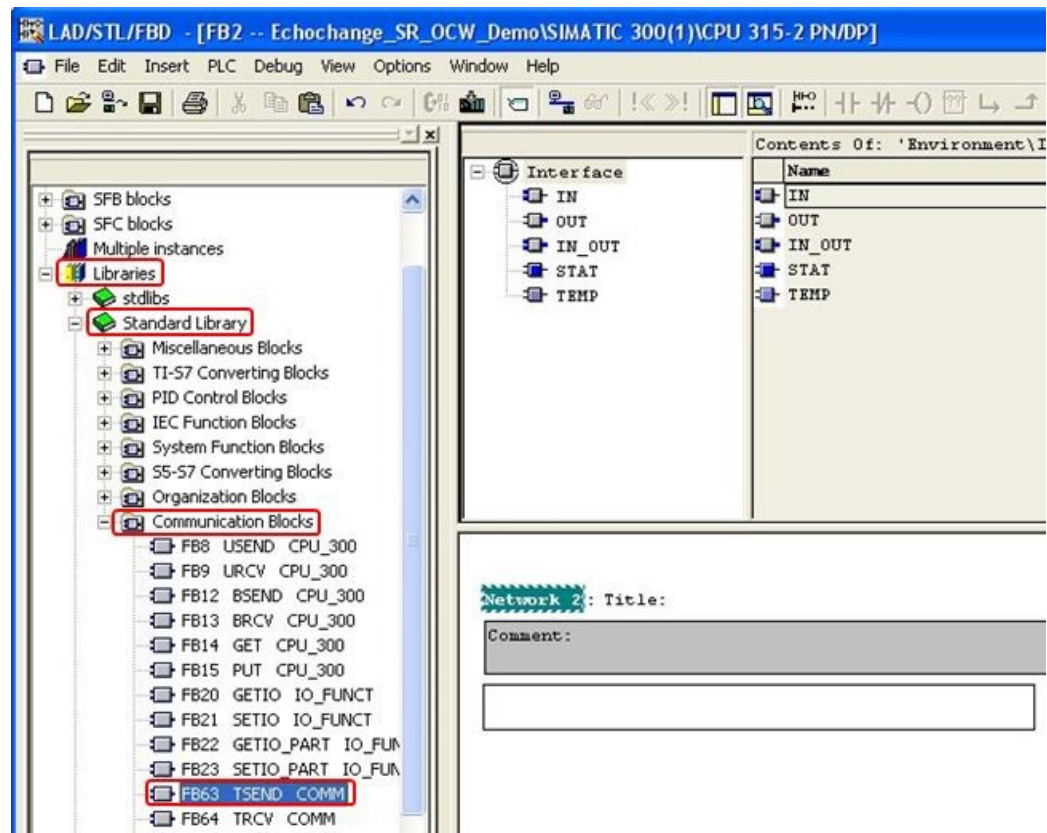
21. For the REQ parameter, enter the memory bit we set to 1 in chapter 6 step 3 (**M0.0**). Choose any unused bit or word respectively for the output parameters DONE, BUSY, ERROR and STATUS.



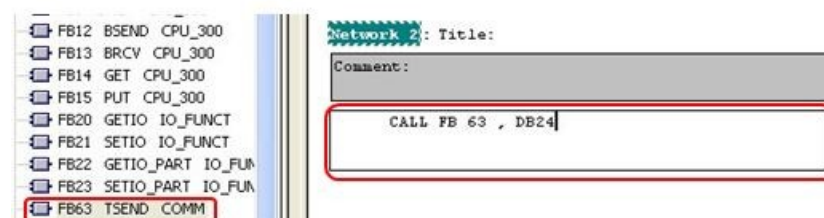
22. Call the **TSEND** function
- a. Insert a new network into function block FB2.



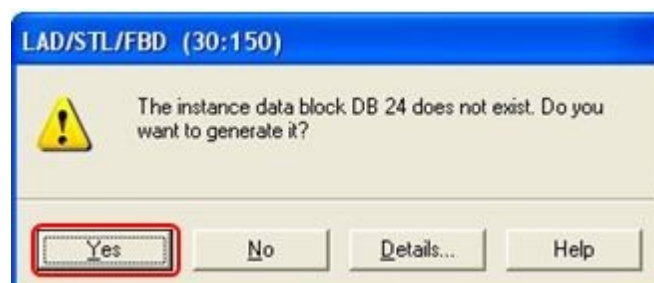
- b. Add a call to FB63 TSEND, found under **Libraries** → **Standard Library** → **Communication Blocks**.



- c. After adding FB63 to Network 2, assign an unused DB number (**DB24** in this example).

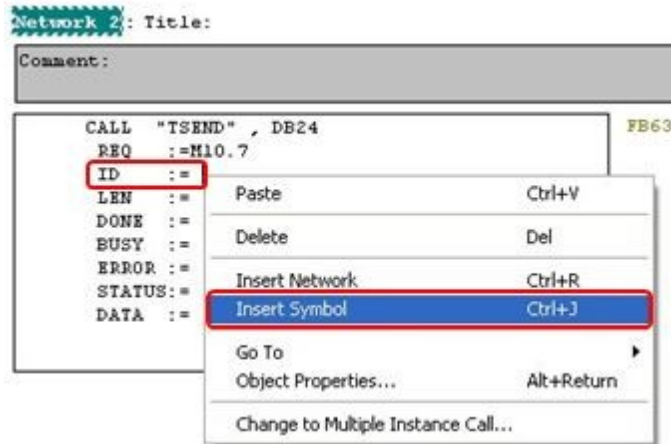


- d. After hitting **Enter**, click **[Yes]** on the following dialog box:

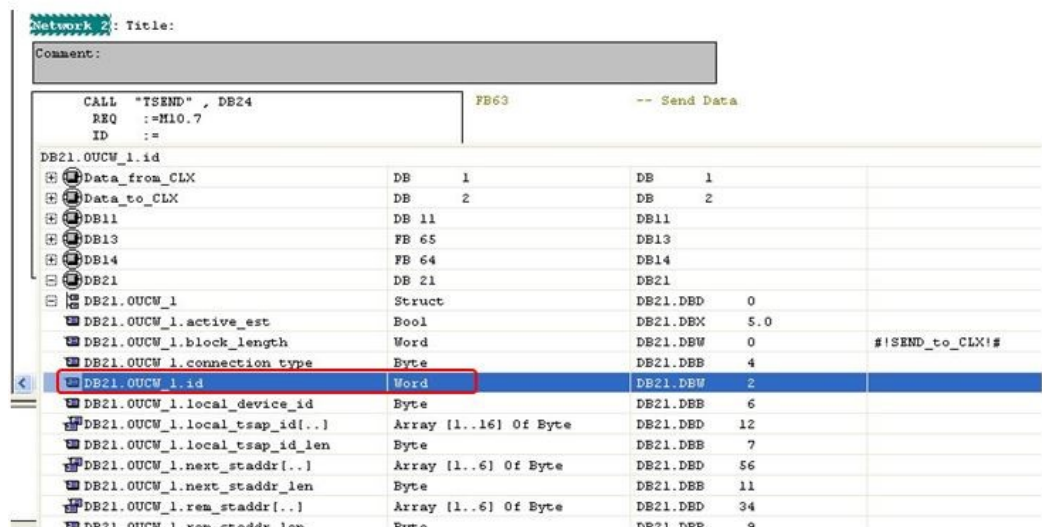




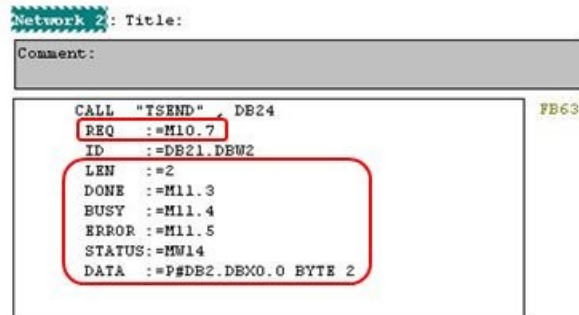
23. Set the parameters for the TSEND function block.
  - a. Right-click on **ID** and select **Insert Symbol**.



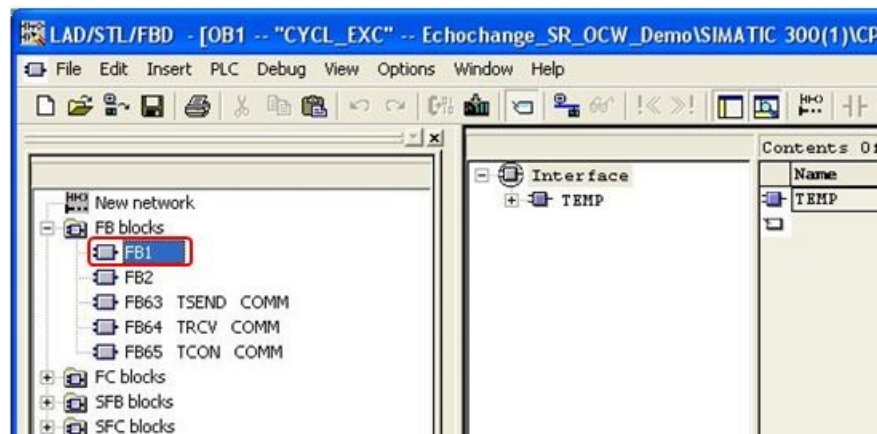
- b. Browse to the id symbol of the DB structure chosen in chapter 6 step 18. In our example it is **DB21.OUCW\_1.id**. Double-click on it.



- c. Set the REQ parameter to the clock memory bit enabled in chapter 4 step 2. We use **M10.7** in this example, which has a frequency of 0.5Hz.
- d. Set LEN to **2**.
- e. Set DATA with a pointer to the data area that will be sent. In chapter 4 step 3 we created DB2 for this purpose and the pointer to this area is P#DB2.DBX0.0 BYTE 2.
- f. Choose any unused bit or word respectively for the remaining output parameters DONE, BUSY, ERROR and STATUS.



- g. Save and close FB2.
24. Program OB1 to call FB1 (receive) and FB2 (send).
- a. Locate FB1 within the FB blocks.



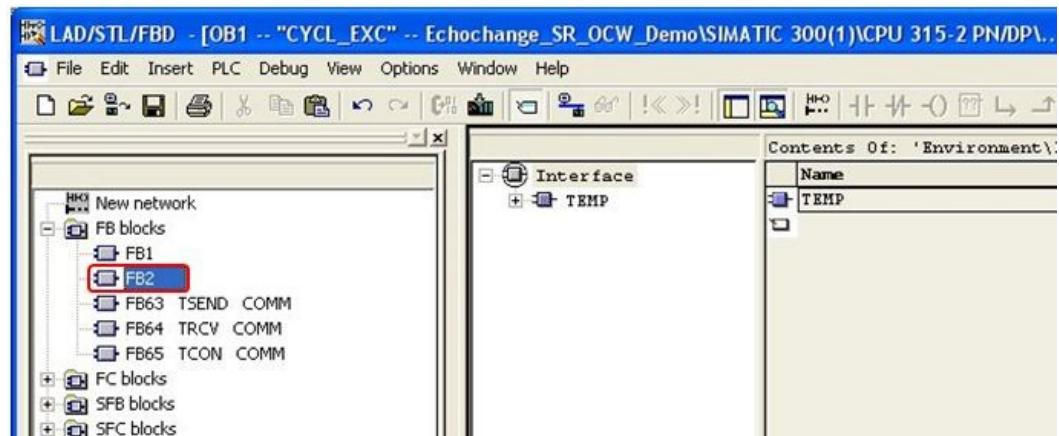
- b. After adding FB1 to your program, assign an unused DB number (**DB12** in this example).



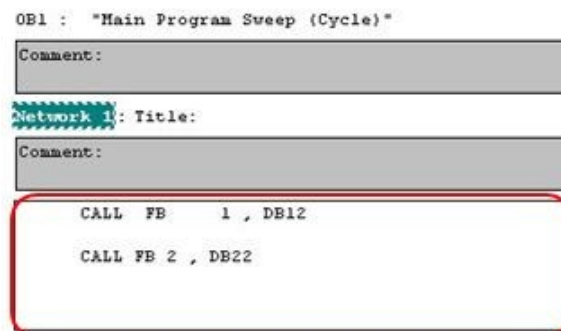
- c. After hitting **Enter**, click **[Yes]** on the following dialog box:



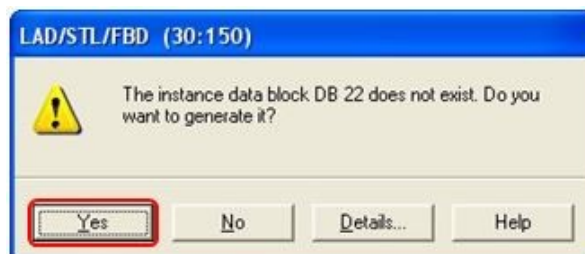
- d. Locate FB2 within the **FB blocks**.



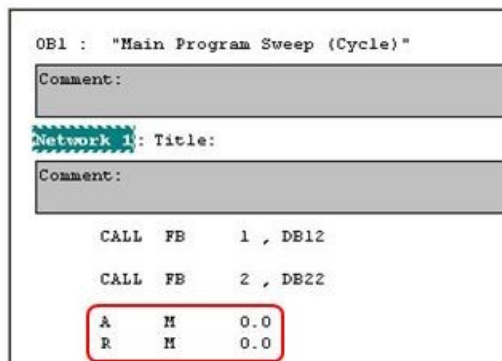
- e. After adding FB2 to your program, assign an unused DB number (**DB22** in this example).



- f. After hitting **Enter**, click **[Yes]** on the following dialog box:



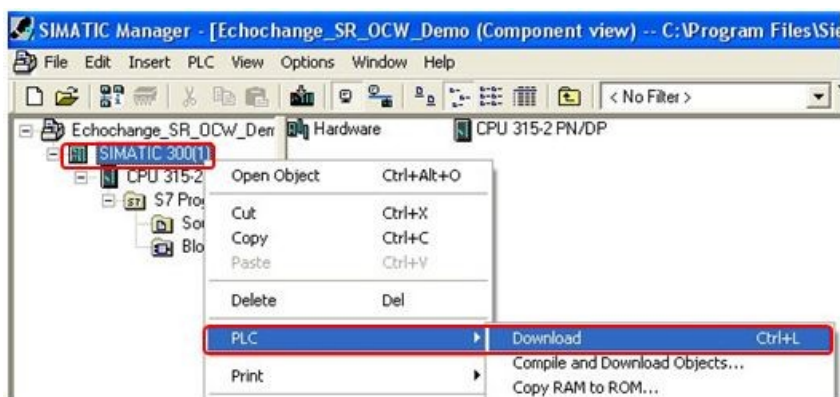
25. Reset the memory bit M0.0 which was used to indicate the start-up of the CPU and trigger the connection establishment of both connections.



26. This is the LAD equivalent:



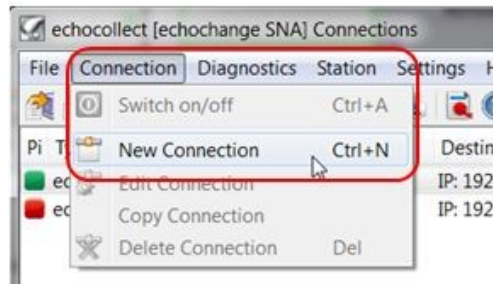
27. Download the project to the S7-300 device.



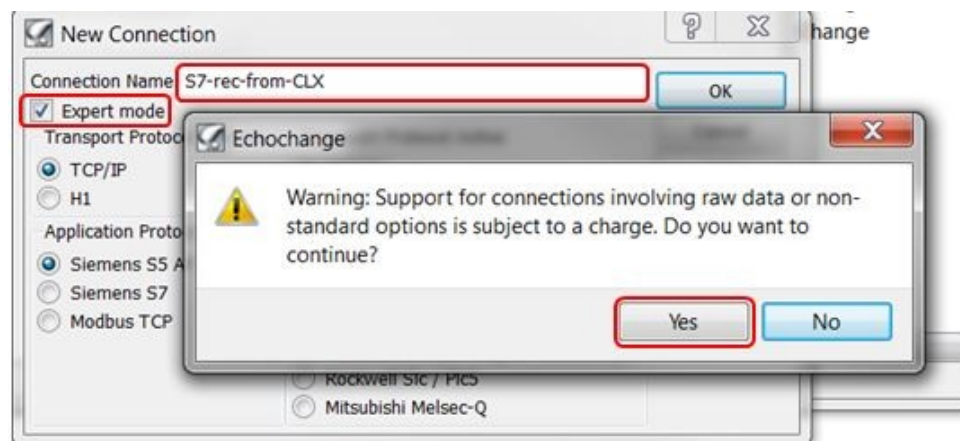
## 8 NetCon echo - establish the connections

1. Create a new connection for the communication direction **CLX → Echochange → Siemens**.

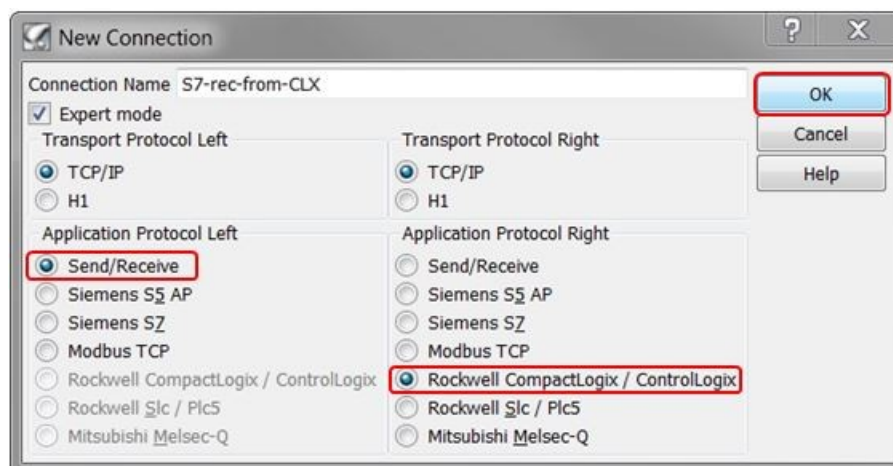
- a. Click on **Connection → New Connection**.



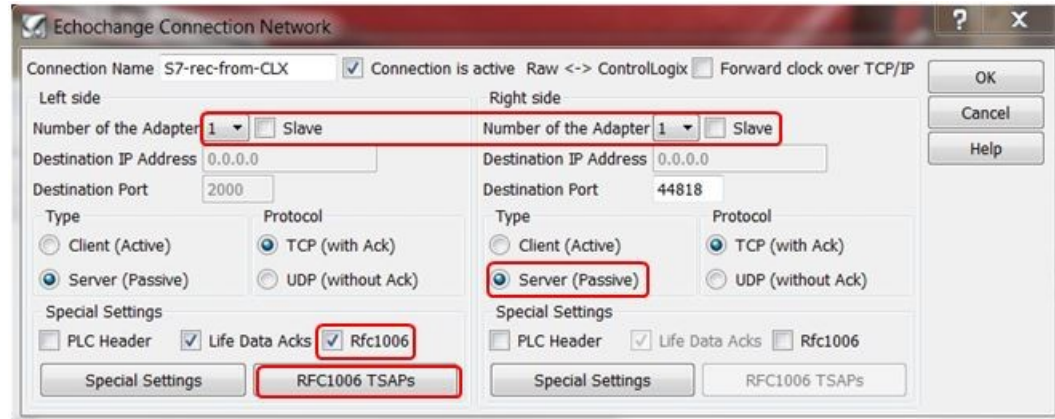
- b. Enter a connection name and enable **Expert mode**.
- c. After **Expert mode** is checked, a warning message will pop up. Click **[Yes]**.



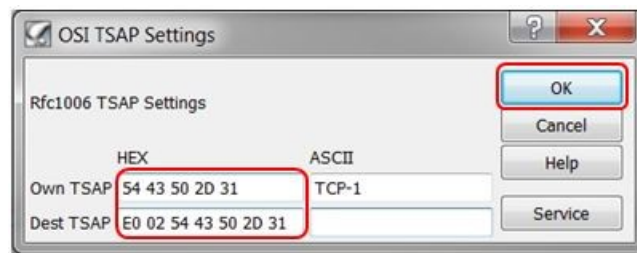
- d. Select **Send/Receive** under the **Application Protocol Left** section and **Rockwell CompactLogix/ControlLogix** under the **Application Protocol Right** section.



- e. Since we are only using port 1 (Eth 1), we will leave the **Number of Adapter** at **1**. On the **Right side** (ControlLogix), uncheck **Slave** and select **Server (Passive)** type. On the **Left side** under **Special Settings**, check **Rfc1006**.
- f. Click on the **[RFC1006 TSAPs]** button.



- g. Enter the HEX values of the TSAPs created by the **Open Communication Wizard** in chapter 6 step 7.

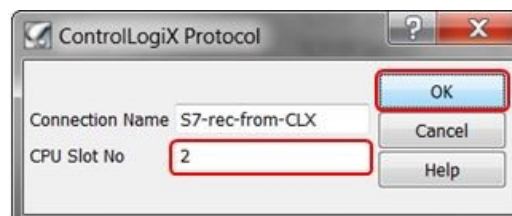


### Note

Own TSAP = Partner B TSAP

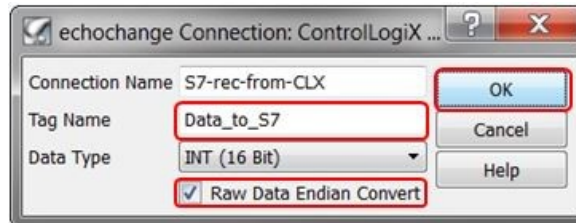
Dest TSAP = Partner A TSAP

- h. After clicking **[OK]** in the **Echochange Connection Network** window, enter the ControlLogix CPU slot. In our case it's 2.



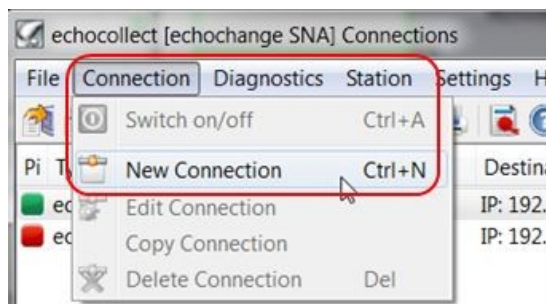


- i. Enter the ControlLogix tag name created in chapter 2 step 2 (used to send data to the S7-300) and enable **Raw Data Endian Convert**.

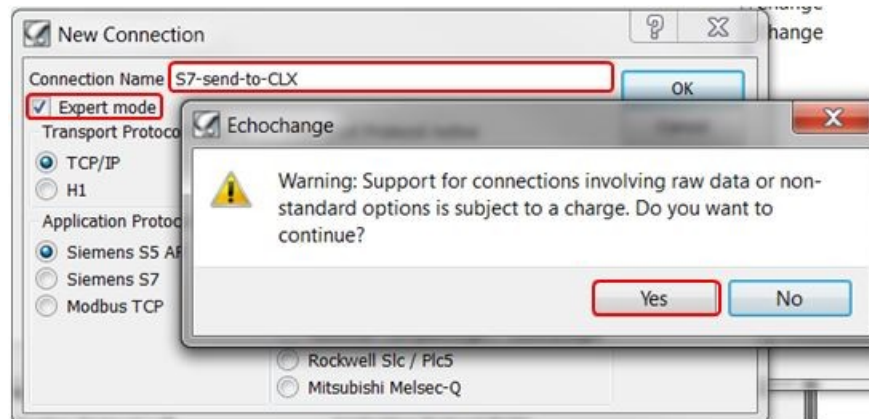


2. Create a new connection for the communication direction **Siemens → Echochange → CLX**.

- a. Click on **Connection → New Connection**:

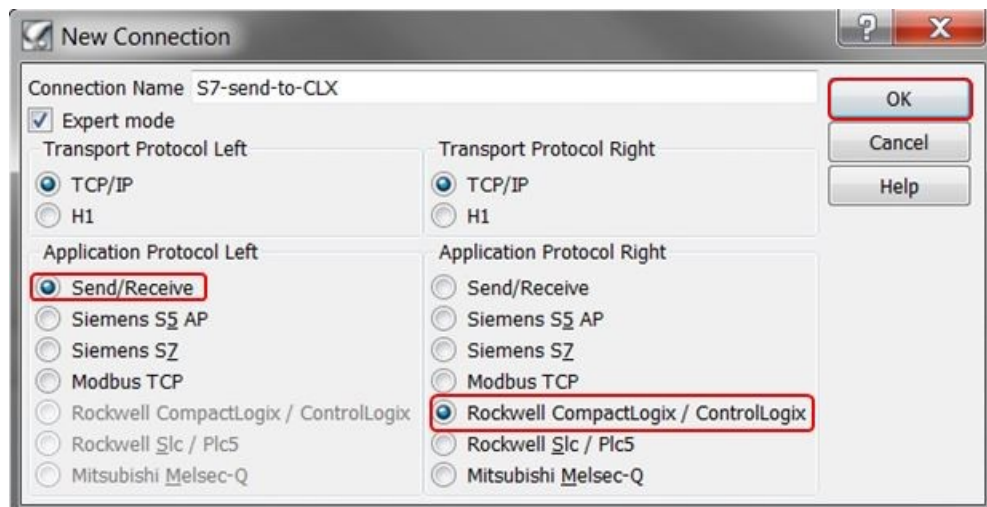


- b. Enter a connection name and enable **Expert mode**. After **Expert mode** is checked, a warning will pop up. Click **[Yes]**.

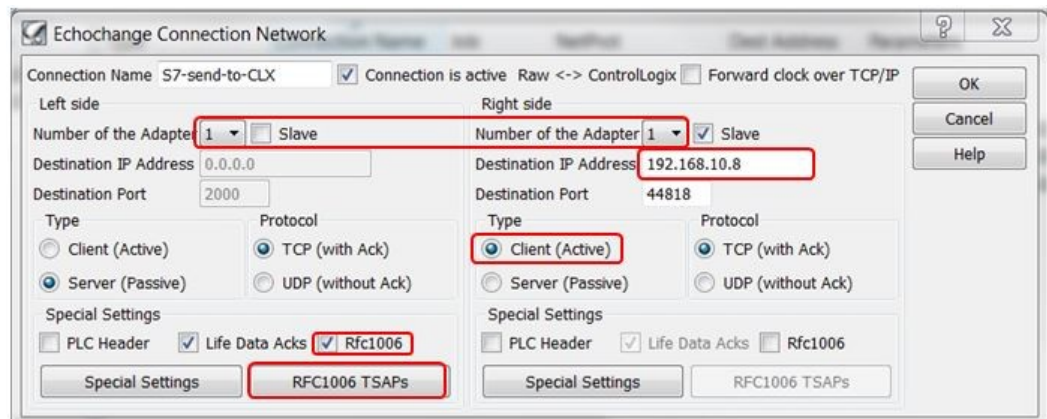




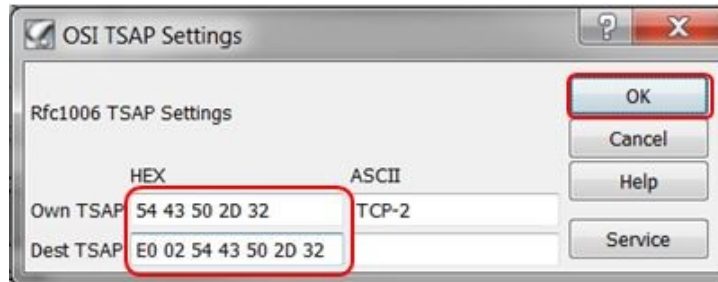
- c. Select **Send/Receive** under the **Application Protocol Left** section and **Rockwell CompactLogix/ControlLogix** under the **Application Protocol Right** section. Then click [OK].



- d. Since we are only using port 1 (Eth 1), we will leave the **Number of Adapter** at 1.
- e. On the **Right side** (ControlLogix), select **Client (Active)** type and enter the IP address of the ControlLogix PLC.
- f. On the **Left side** under **Special Settings**, check **Rfc1006**.
- g. Click on the **[RFC1006 TSAPs]** button.



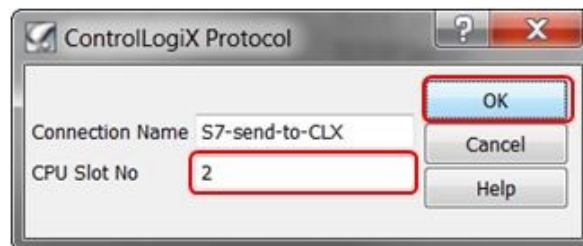
- h. Enter the HEX values of the TSAPs created by the Open Communication Wizard in chapter 6 step 17.

**Note**

Own TSAP = Partner B TSAP

Dest TSAP = Partner A TSAP

- i. After clicking **[OK]** in the **Echochange Connection Network** window, enter the ControlLogix CPU slot. In our case it's 2.



- j. Enter the ControlLogix tag name created in chapter 2 step 1 (used to receive data from the S7-300) and enable **Raw Data Endian Convert**.



Setup is now complete.



**Softing Industrial Automation GmbH**

Richard-Reitzner-Allee 6

85540 Haar / Germany

Tel: + 49 89 4 56 56-0

Fax: + 49 89 4 56 56-488

Internet: <http://industrial.softing.com>

Email: [info.automation@softing.com](mailto:info.automation@softing.com)

Support: [support.automation@softing.com](mailto:support.automation@softing.com)