

WEINTEK LABS., INC.

# TCP/UDP Socket

TCP/UDP Data Transmission  
achieved by Network Library

Demo Project

## Contents

1. Overview.....	1
2. Network Library.....	2
3. Operation .....	9

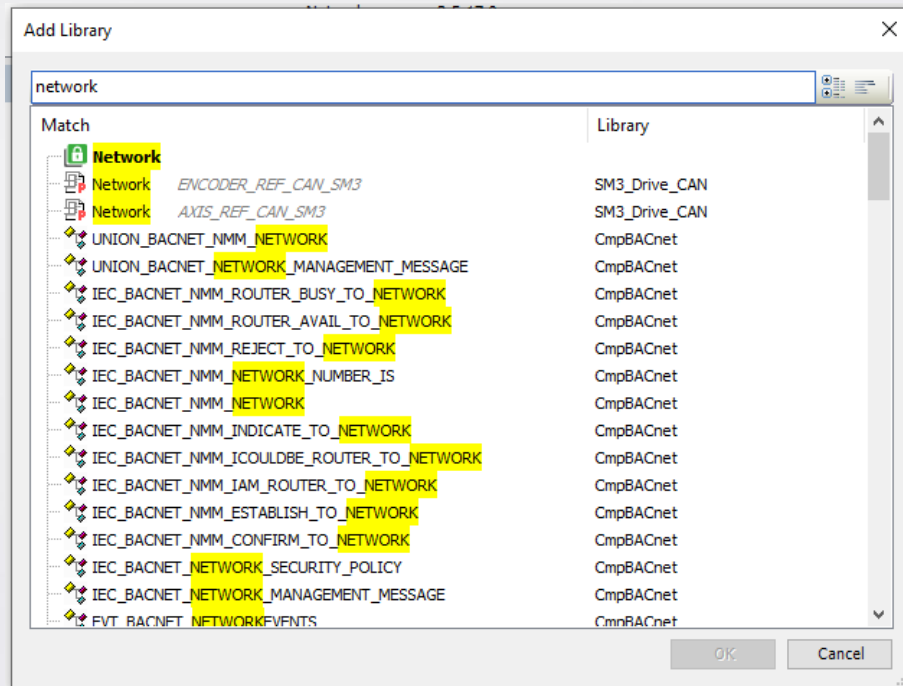
### 1. Overview

TCP and UDP are communication protocols widely used in the industrial field for communication between production devices or between production devices and internal management system. With Network Library supported in CODESYS, applications using TCP/UDP Socket can be achieved with TCP\_Server / TCP\_Client / UDP\_Peer function blocks.

This demo project explains how to perform TCP & UDP data transmission using Weintek Built-in CODESYS and SocketTest software that runs on PC. Other TCP & UDP socket test software can be used to replace SocketTest on PC.

## 2. Network Library

**Step 1.** In CODESYS software add Network Library.



※ In old versions of CODESYS, adding CAA Net Base Service & CAA Type External Library is necessary because Network Library is not supported.

### Function Block

#### TCP\_Client:

NBS.TCP_Client	
EN	ENO
xEnable	xDone
udiTimeOut	xBusy
ipAddr	xError
uiPort	eError
	xActive
	hConnection

Function: Establishes connection with TCP Server.

Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
udiTimeOut	UDINT	Defines the time ( $\mu$ s) after which the connection setup aborts with an error message. After the timer has expired, xError is set. With udiTimeOut=0 the timeout is deactivated.
ipAddr	NBS.IP_ADDR	The IP address to establish connection.
uiPort	UINT	The port number to establish connection.
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>
xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID
xActive	BOOL	TRUE : The handle of the connection is valid. FALSE : The handle of the connection is invalid.
hConnection	CAA.HANDLE	Handle of Connection

## TCP\_Server:

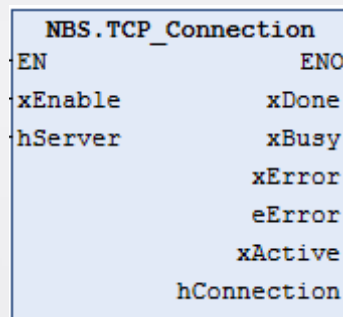
NBS.TCP_Server	
EN	ENO
xEnable	xDone
ipAddr	xBusy
uiPort	xError
	eError
	hServer

Function: Sets up a TCP Server.

Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
udiTimeOut	UDINT	Defines the time ( $\mu$ s) after which the connection setup aborts with an error message. After the timer has expired, xError is set. With udiTimeOut=0 the timeout is deactivated.
ipAddr	NBS.IP_ADDR	The IP address for sending and receiving data.
uiPort	UINT	The port number for sending and receiving

		data.
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>
xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID
hServer	CAA.HANDLE	Handle of Server

### TCP\_Connection



Function: Allows connection when TCP Server is active.

Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
hServer	CAA.HANDLE	Handle of Server
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>
xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID
xActive	BOOL	TRUE : The handle of the connection is valid. FALSE : The handle of the connection is invalid.
hConnection	CAA.HANDLE	Handle of Connection

## TCP\_Read

NBS.TCP_Read	
EN	ENO
xEnable	xDone
hConnection	xBusy
szSize	xError
pData	eError
	xReady
	szCount

Function: Reads data from the previously established connection.

Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
hConnection	CAA.HANDLE	Handle of Connection
szSize	CAA.SIZE	Maximum number of bytes to be read; can be retrieved via operator SIZEOF.
pData	CAA.PVOID	Target address for the first byte to be read; can be retrieved via operator ADR.
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>
xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID
xReady	BOOL	TRUE : Data received FALSE : No data received
szCount	CAA.SIZE	Size of the received data.

## TCP\_Write

NBS.TCP_Write	
EN	ENO
xExecute	xDone
udiTimeOut	xBusy
hConnection	xError
szSize	eError
pData	

Function: Writes data to the previously established connection.

Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
udiTimeOut	UDINT	Defines the time ( $\mu$ s) after which the connection setup aborts with an error message. After the timer has expired, xError is set. With udiTimeOut=0 the timeout is deactivated.
hConnection	CAA.HANDLE	Handle of Connection
szSize	CAA.SIZE	Maximum number of bytes to be written; can be retrieved via operator SIZEOF.
pData	CAA.PVOID	Target address for the first byte to be written; can be retrieved via operator ADR.
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>
xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID

## UDP\_Peer

NBS.UDP_Peer	
EN	ENO
xEnable	xDone
ipAddr	xBusy
uiPort	xError
ipMultiCast	eError
	xActive
	hPeer

Function: Activates UDP.

Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
ipAddr	NBS.IP_ADDR	The IP address to establish connection.
uiPort	UINT	The port number to establish connection.
ipMultiCast	IP_ADDR	Multicast address. '255.255.255.255' => INADDR_NONE
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>



xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID
xActive	BOOL	TRUE : Handle of peer is valid. FALSE : Handle of peer is invalid.
hPeer	CAA.HANDLE	Handle of Peer

## UDP\_Send

NBS.UDP_Send	
EN	ENO
xExecute	xDone
udiTimeOut	xBusy
hPeer	xError
ipAddr	eError
uiPort	
szSize	
pData	

Function: Sends data when UDP is activated.

Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
hPeer	CAA.HANDLE	Handle of Peer
ipAddr	NBS.IP_ADDR	The IP address for sending and receiving data.
uiPort	UINT	The port number for sending and receiving data.
szSize	CAA.SIZE	Maximum number of bytes to be written; can be retrieved via operator SIZEOF.
pData	CAA.PVOID	Target address for the first byte to be written; can be retrieved via operator ADR.
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>
xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID

## UDP\_Receive

NBS.UDP_Receive	
EN	ENO
xEnable	xDone
hPeer	xBusy
szSize	xError
pData	eError
	xReady
	ipFrom
	uiPortFrom
	szCount

Function: Receives data when UDP is activated.

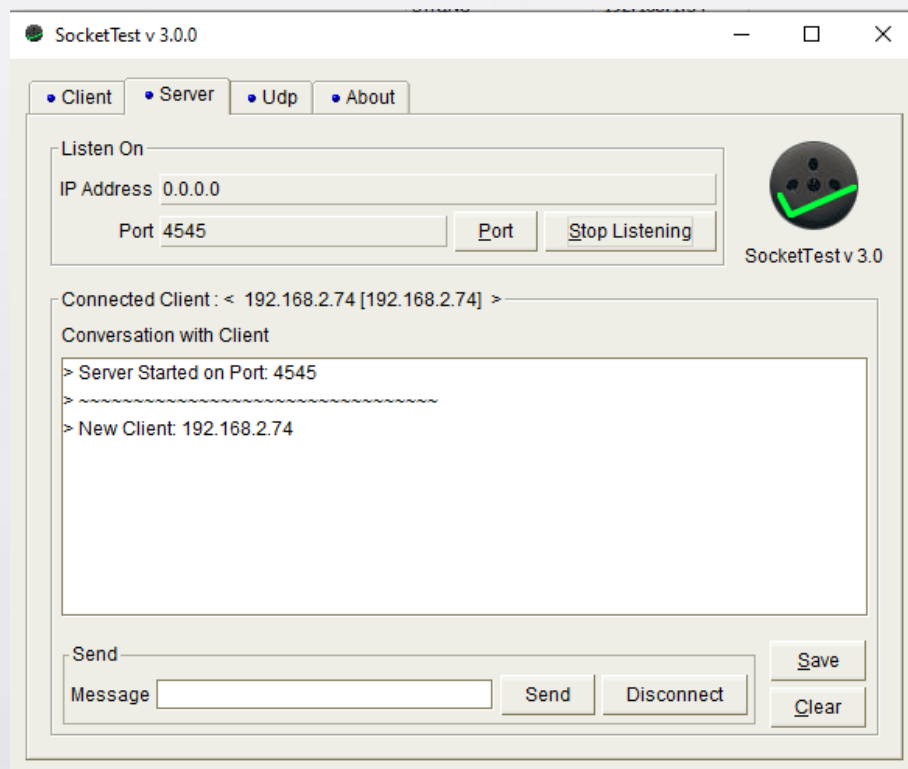
Input/Output	Type	Description
xEnable	BOOL	See: <a href="#">Library Guidelines</a>
hPeer	CAA.HANDLE	Handle of Peer
szSize	CAA.SIZE	Maximum number of bytes to be read; can be retrieved via operator SIZEOF.
pData	CAA.PVOID	Target address for the first byte to be read; can be retrieved via operator ADR.
xDone	BOOL	See: <a href="#">Library Guidelines</a>
xBusy	BOOL	See: <a href="#">Library Guidelines</a>
xError	BOOL	See: <a href="#">Library Guidelines</a>
eError	NBS.ERROR	Error ID
ipFrom	NBS.IP_ADDR	The IP address for receiving data.
uiPortFrom	UINT	The port number for receiving data.
xReady	BOOL	TRUE : Data received FALSE : No data received
szCount	CAA.SIZE	Size of the received data.

### 3. Operation

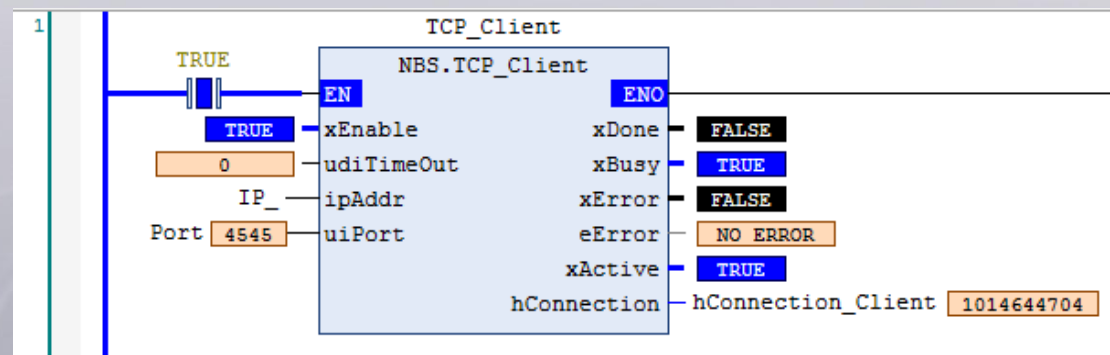
Run the demo project: CODESYS\_TCP\_Socket.

#### TCP\_Client:

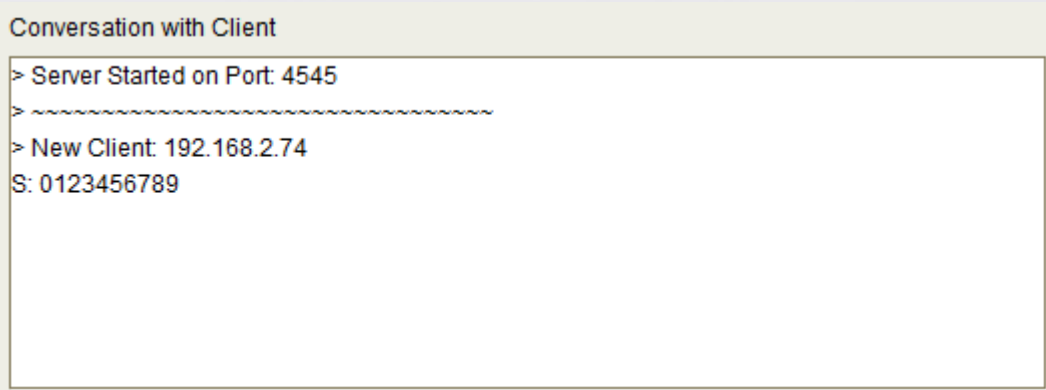
**Step 1.** In SocketTest software open the Server tab and then click [Start Listening].



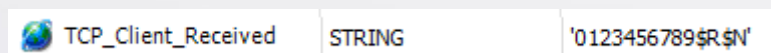
**Step 2.** Execute Program TCP\_Client. IP address = PC's address, Port = the port number found in SocketTest's Server tab, TCP\_Client.xEnable=TRUE.



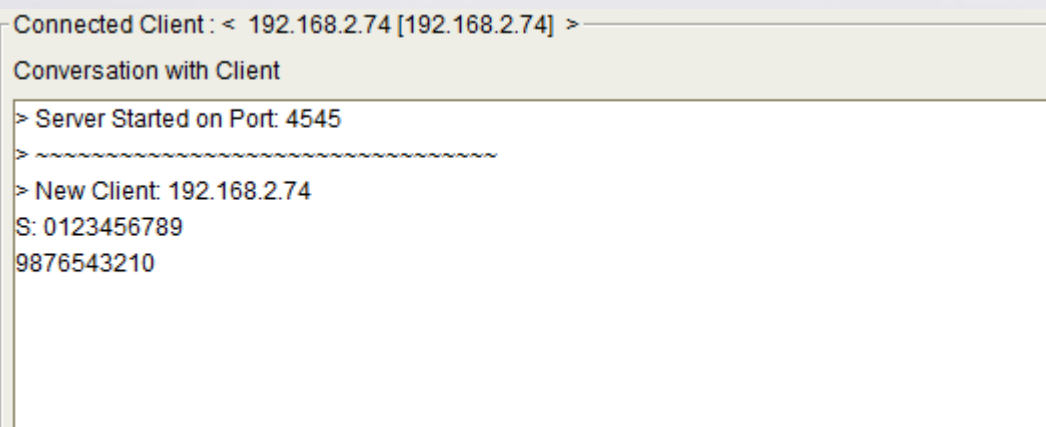
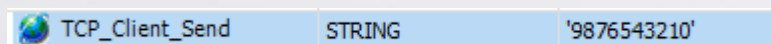
**Step 3.** Enter a string in the Message field in SocketTest and then click [Send].



**Step 4.** When TCP\_Read.xEnable = “True”, the string sent from SocketTest is received in GVL.TCP\_Client\_Received.

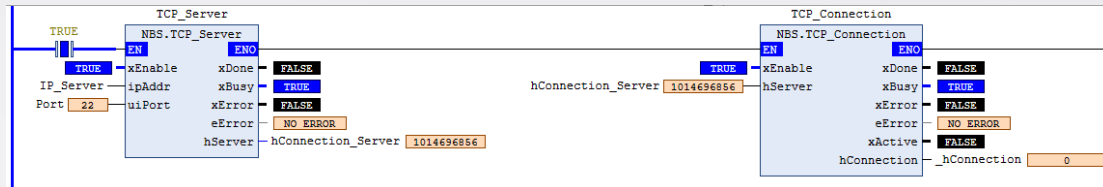


**Step 5.** In GVL.TCP\_Client\_Send enter a string, and then the string is sent to SocketTest when TCP\_Write.xEnable = “True”. The data in GVL.TCP\_Client\_Send can then be viewed in the Message field of SocketTest.

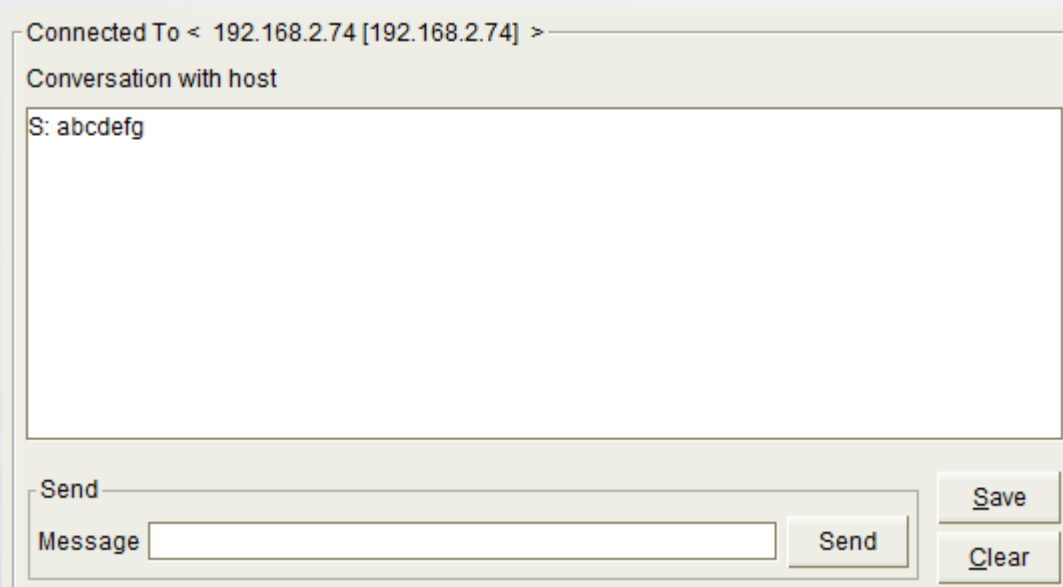


## TCP Server:

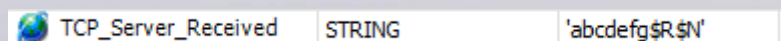
**Step 1.** Execute Program TCP\_Client. When TCP\_Server.xEnable & TCP\_Connection.xEnable = "TRUE", the TCP server is activated and ready for connection.



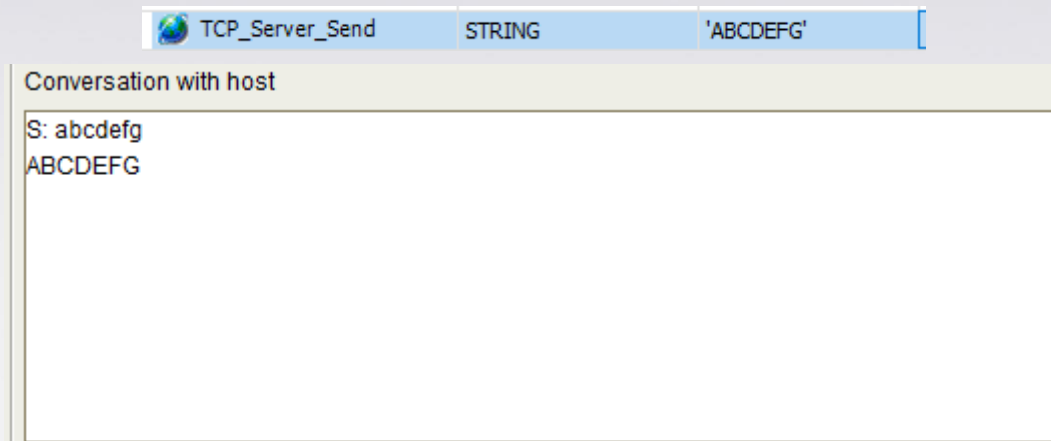
**Step 2.** In the Client tab in SocketTest, IP address = CODESYS address, Port = TCP\_Server's uiPort. Click [Connect], enter a string in Message filed and then click [Send].



**Step 3.** When TCP\_Read.xEnable = "TRUE", the string sent from SocketTest is received in GVL.TCP\_Server\_Received.

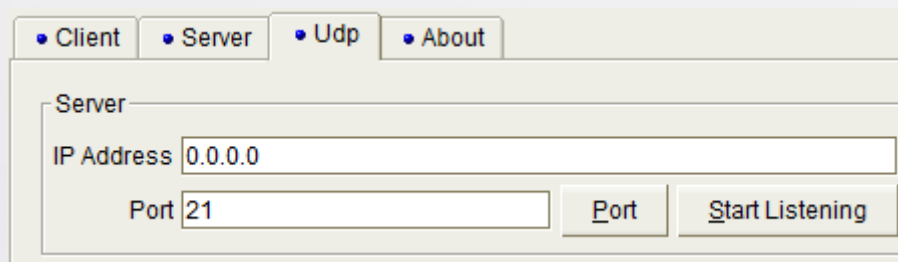


**Step 4.** In GVL.TCP\_Client\_Send enter a string, and then the string is sent to SocketTest when TCP\_Write.xEnable = "True". The data in GVL.TCP\_Client\_Send can then be viewed in the Message filed of SockektTest.

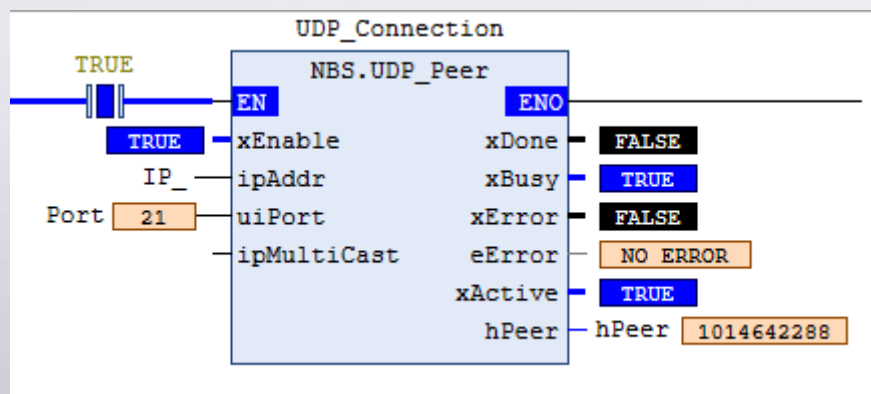


## UDP

**Step 1.** In the UDP tab in SocketTest click [Start Listening].



**Step 2.** Execute Program UDP. IP address = PC's address, Port = the port number found in SocketTest's UDP tab, UDP\_Peer.xEnable = "TRUE".



**Step 3.** Enter a string in the Message field in SocketTest and then click [Send].  
When UDP\_Read.xEnable = "TRUE", the string sent from SocketTest is received in GVL.TCP\_Client\_Received.

Conversation

> Server Started on Port : 21

> ~~~~~

S[192.168.2.74:21]: UDP Client send

Client

IP Address

192.168.2.74

Port

21


Port

Message


Send

Save

Clear

	UDP_Received	STRING	'UDP Client send'
-----------------------------------------------------------------------------------	--------------	--------	-------------------

**Step 4.** In GVL\_UDP\_Send enter a string, and then the string is sent to SocketTest when TCP\_Write.xEnable = “True”. The data in GVL\_UDP\_Send can then be viewed in the Message filed of SocektTest.

	UDP_Send	STRING	'CODESYS UDP send'
-------------------------------------------------------------------------------------	----------	--------	--------------------

Conversation

> Server Started on Port : 21

> ~~~~~

S[192.168.2.74:21]: UDP Client send

R[192.168.2.74:21]: CODESYS UDP send

CODESYS® is a trademark of CODESYS GmbH.

Other company names, product names, or trademarks in this document are the trademarks or registered trademarks of their respective companies.

This document is subject to change without prior notice.

Copyright© 2020 Weintek Lab., Inc. All rights reserved.