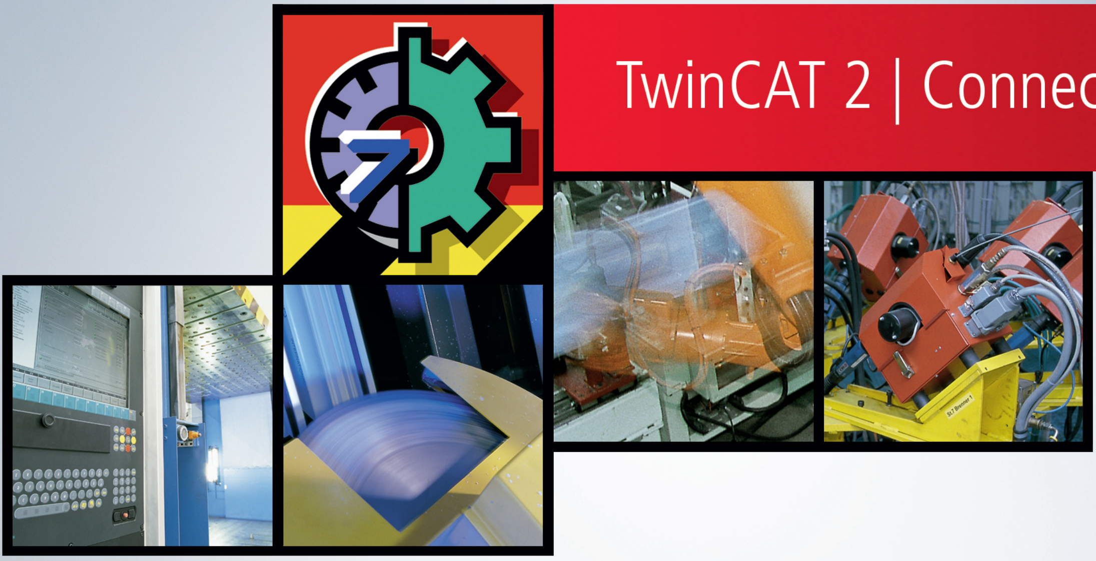
BECKHOFF New Automation Technology

Manual | EN

Real-Time Ethernet



TwinCAT 2 | Connectivity

2022-09-28 | Version: 1.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

Trademarks

Patent Pending

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

ADS over Real-time Ethernet enables real-time capable communication via Ethernet with UDP.

System requirements

- W2K, XP, Windows Embedded standard;
- TwinCAT installation level: TwinCAT PLC or higher;
- TwinCAT system version 2.10.0 build >= 1328;

Installation

A prerequisite for real-time Ethernet is that all communication devices have the TwinCAT Ethernet protocol driver [▶ 14] installed.

Further documentation

- Introduction in TwinCAT real-time Ethernet [▶ 9]
- Network variables (publisher/subscriber)

3 Introduction of TwinCAT Real-Time Ethernet

Ethernet communication in real-time

Ethernet takes the next step as a “Fieldbus” in Beckhoff’s TwinCAT control system. In addition to tough real-time requirements, it permits the use of standard components “on the same wire”. The BK9000 Ethernet Bus Coupler and the AX2000-B900 servo amplifier are the first Fieldbus components to benefit from the real-time capability. New network variables accelerate the real-time data exchange
between controllers, making it as easy to implement as connecting another digital input.

Beckhoff’s range of Ethernet products has been in use for years, and is becoming increasingly popular. The advantages of using the office Ethernet standard communication method in industrial settings are clear:
- **Uses standard hardware components (e.g. a standard of-the-shelf)***Switch*
- **Standard protocols** can be employed
- **Data transmission rates are very high compared to other networks**
- **Linking the network to the rest of the world over the Internet is straightforward**
- **Remote maintenance and diagnosis**

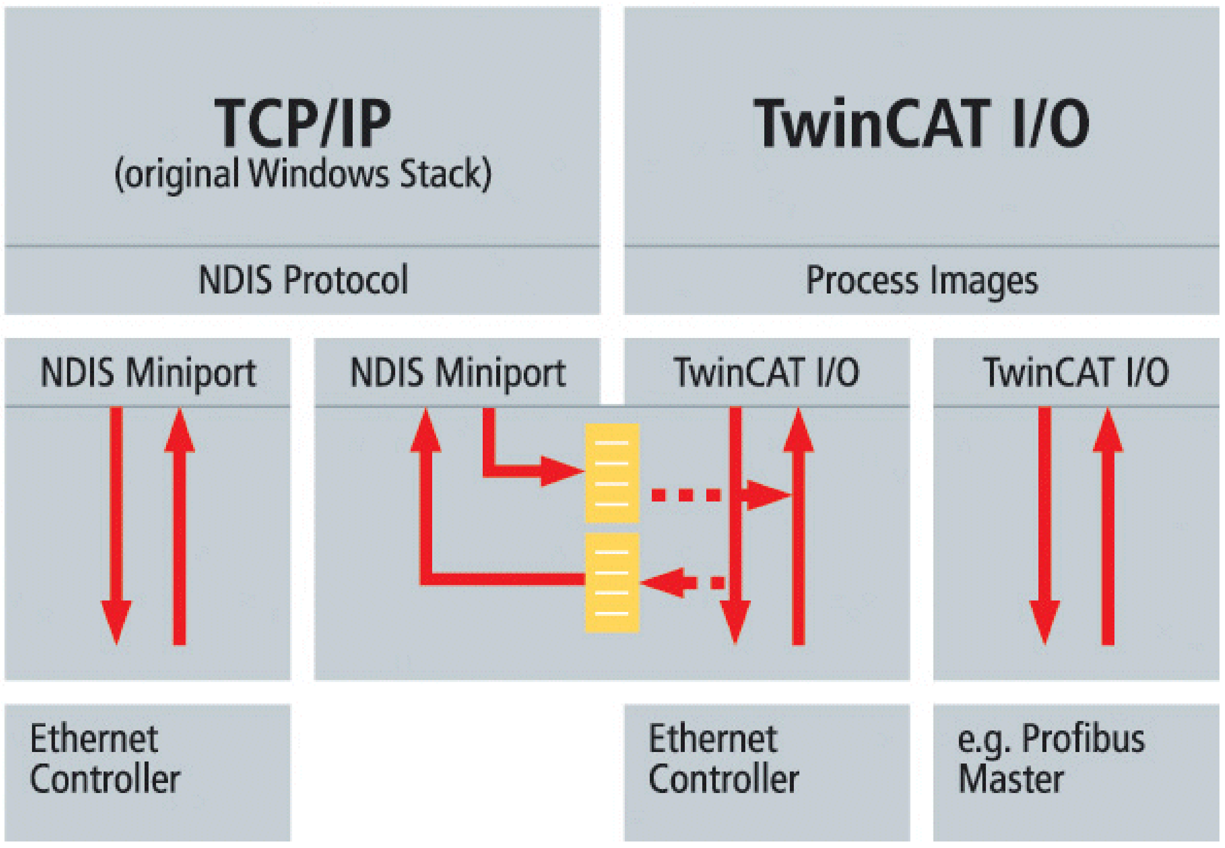
Communication over Ethernet has now become accepted in industrial automation and many groups and committees are concerning themselves with this topic. The absence of effective real-time capability, however, is a problem, which has limited Ethernet’s use in the classical fieldbus area. Some techniques do allow a certain degree of real-time capability, but are based on proprietary systems, and do not allow standard components and protocols to be used at the same time.
- "Real-time capability" is a somewhat flexible term when used in control theory.
- "Real-time" depends heavily on the requirements of the particular application and the control loops in which the automation components are used. However, from the point of view of automation engineering, and against the background that fieldbus specialist Beckhoff have to offer, a rough division can be drawn:
- **The toughest requirements** involve cycle times of around 50 µs and permissible jitter (deviations from the desired cycle time) of around 10 µs. Requirements tighter than this are currently still handled with the aid of special hardware rather than directly over fieldbusses. Typical cycle time requirements with position-controlled drives are in the range of milliseconds (1-4 ms), in which case jitter times should be less than 20 µs. Pure PLC applications often require cycle times no shorter than 10 ms; correspondingly, jitter times may here be significantly longer, and lie in the millisecond range. Data communication between the controller and the supervisory system can often satisfactorily be handled with cycle times in the range of seconds. In fact it may not be configured cyclically, but may be event driven.

Remote servicing and diagnosis should also be mentioned. Cycle and jitter times here are less relevant than reaction times and the general possibility of being able to communicate across the boundaries of networks. TwinCAT automation software now offered with Real-time Ethernet means that all the communication requirements that have just been mentioned can be satisfied using one and the same technology, both from the point of view of the devices used and of the protocols employed.

Principle of operation

The TwinCAT network card driver is linked into the system in such a way that it appears as a network driver, compatible with the operating system, and additionally as a TwinCAT fieldbus card. An internal prioritization system and buffer is provided at the transmitter end which always finds a free transmission channel for Ethernet frames from the real-time system that may be queuing. **The operating system’s Ethernet frames are only later transmitted in the “gaps” if sufficient time is available**.
At the receiving end, all the Ethernet frames received are examined by the TwinCAT I/O system, and those with real-time relevance are filtered out. All other frames are passed on to the operating system after examination, outside the context of the real-time system.

Using commercially available switches, all of which support full duplex operation at 100 Mbaud, the transmitted frames are passed on to the receiver with a constant delay. A switch ensures that collisions are avoided and only delays occur. In a cyclic control system it is therefore only necessary to ensure that all the relevant input information has arrived before the next cycle starts. When, or in what sequence, they have arrived is not significant.
If the number of participating devices or the frame rate is restricted in accordance with the required cycle time, the preconditions for Ethernet communication with real-time capability are satisfied.



The above picture shows the principle of how the original Windows operating system protocol stack (TCP/IP and UDP/IP) is used besides TwinCAT Real-Time Ethernet. We call this principle "Y-driver" architecture.

Operating modes/protocols

In contrast to the widely used TCP/IP and UDP/IP protocols, which are responsible for the provision of individual Ethernet frames around the world, real-time communication does not leave the local subnet. The overhead involved in TCP/IP, and even that of UDP/IP, can be omitted, and the devices can be addressed directly by means of the hardware addresses (MAC-ID) of the network cards. The structure of Ethernet frames ensures that it is always possible to co-exist with other protocols; even the “real-time frames” can, if necessary, be transmitted with TCP or UDP, if they have to leave their own subnet.
A number of different operating modes have been defined for use in control engineering. They serve different communication tasks, and can, of course, be employed simultaneously.

1.) Master-slave process data communication

Cyclic or event-driven transmission of I/O data – the typical use of modern fieldbusses

2.) Publisher-subscriber process data communication

Process data according to the publisher-subscriber model (also referred to as network variables) is used for regular communication between controllers when a fixed master-slave relationship would not be appropriate. The Publisher sends out its information without concerning itself with where it is going. The communication is only monitored in the Subscriber. Mutual publisher-subscriber relationships permit bi-directional and multi-directional communication. The publisher can be configured to send the data by broadcast, multicast or unicast. Multicasts reduce the loads at the network devices’ receive queues, because the messages are evaluated as soon as they reach the Ethernet controller. Only if unicasts are used the switch (without extensive configuration) can open parallel communication paths and increase the useful bandwidth.

3.) Data communication as required

This is a type of communication made possible in the TwinCAT system through ADS communication, and which sends communication strings from one device to another “as required”. Services are executed and parameters exchanged in this way.

The chosen protocol structure means that other operating modes or communication profiles can easily be integrated in the future, and can co-exist with the existing modes without difficulty.

Compatible components

The first components from the Beckhoff product range to have been extended for real-time Ethernet application are the BK9000 Ethernet Bus Coupler and the AX2000-B900 drive. These two components open up almost the entire range of industrial signals and applications.
All TwinCAT controllers (from Version 2.9 upwards) are compatible, and can participate both as “fieldbus masters” and in communication using network variables. TwinCAT Real-Time Ethernet supports all controllers (network adapters) of the Intel 8255x family. This is one of the most widespread Ethernet controllers, a component in the latest Intel chip set, which already include a compatible network connection. Support for further Ethernet controllers in the future may be considered – in the light of the wide popularity of the Intel family and its compatibility even with Gigabit Ethernet (Intel 8254x family) – but not absolutely essential. The new Embedded-PC CX1000 small controller is, of course, always fitted with an appropriate Ethernet controller.

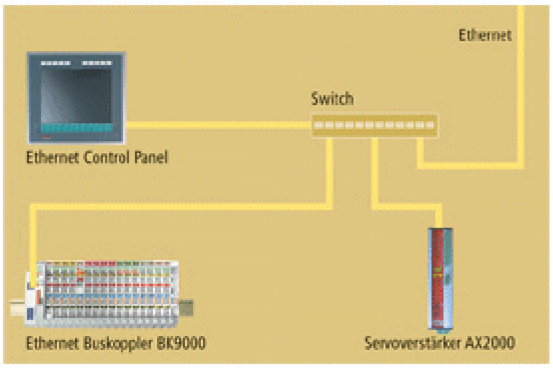
Performance

Judging the performance of a fieldbus system exclusively on the basis of the baudrate is, without doubt, too simple, ignoring as it does other significant communication parameters such as the efficiency of the protocols, reaction time, jitter, minimum telegram length, pause times and so forth. All the same, 100 Mbaud permit significantly faster data transfer than is usual at present in fieldbus environments. In addition to this, the protocols used provide a significantly improved efficiency in contrast with TCP or UDP communication, in particular when data telegrams are short.
One factor which is often ignored when considering the capacity of a fieldbus system is the transfer of data between the controller CPU to the communication chip or processor; that is to say, between the host system memory and the subsystem memory. In PC based controllers, the data is usually copied over the ISA or PCI bus into the DPRAM of a fieldbus master card, and vice-versa. Whereas PC processors have reached speeds of around 3 GHz, even the supposedly fast PCI bus has become a bottleneck, so that 20-30 % of the CPU performance is lost for PCI transfers.
Modern network controllers operate in what is known as the “bus master DMA mode”, accessing the host system’s memory directly. This occurs in parallel with other CPU tasks, and therefore significantly reduces the CPU load.

Applications/Linking to TwinCAT

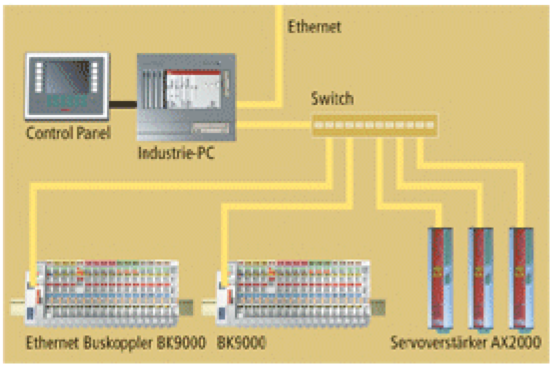
The high data transfer capacity, the fundamental real-time capability and the protocols that are employed cover all the communication demands made by a fast machine controller. When all these features are considered we are soon led to the conclusion that classical fieldbusses are no longer needed.
First of all, however, Ethernet technology must still be proven at the field level, and must meet all requirements such as those for simplicity of installation and configuration, mutual compatibility, EMC immunity and, not least, efficiency and device costs relevant to the industrial environment. Standard fieldbusses are widely accepted, and there are many devices offered by many vendors including Beckhoff. Therefore, they will continue to have great significance to the market. Here once again, TwinCAT’s flexible I/O system offers a way forward: It permits multiple fieldbusses to be operated in parallel, including, of course, the parallel operation of classical fieldbusses with real-time Ethernet – and all of this is entirely transparent to the application.

Application Example 1:



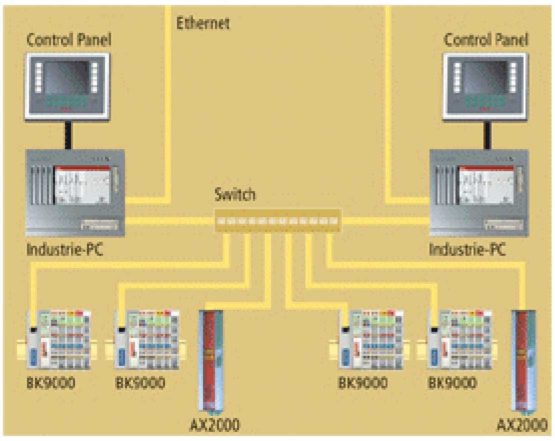
Relatively simple applications can manage with a single Ethernet connection, both for the real-time communication to the I/O level and for the higher-level communication for the purposes of administration and remote diagnosis. The prioritization that is used ensures that the real-time communication takes place without difficulty.

Application Example 2:



Larger applications make use of a second Ethernet connection, and divide the real-time communication and the higher level communication between two networks. The routing that is required in order, for instance, to remotely diagnose a drive that is connected to a real-time network is performed automatically by the operating system's IP stack, and does not require any proprietary conversion to a different protocol.

Application Example 3:

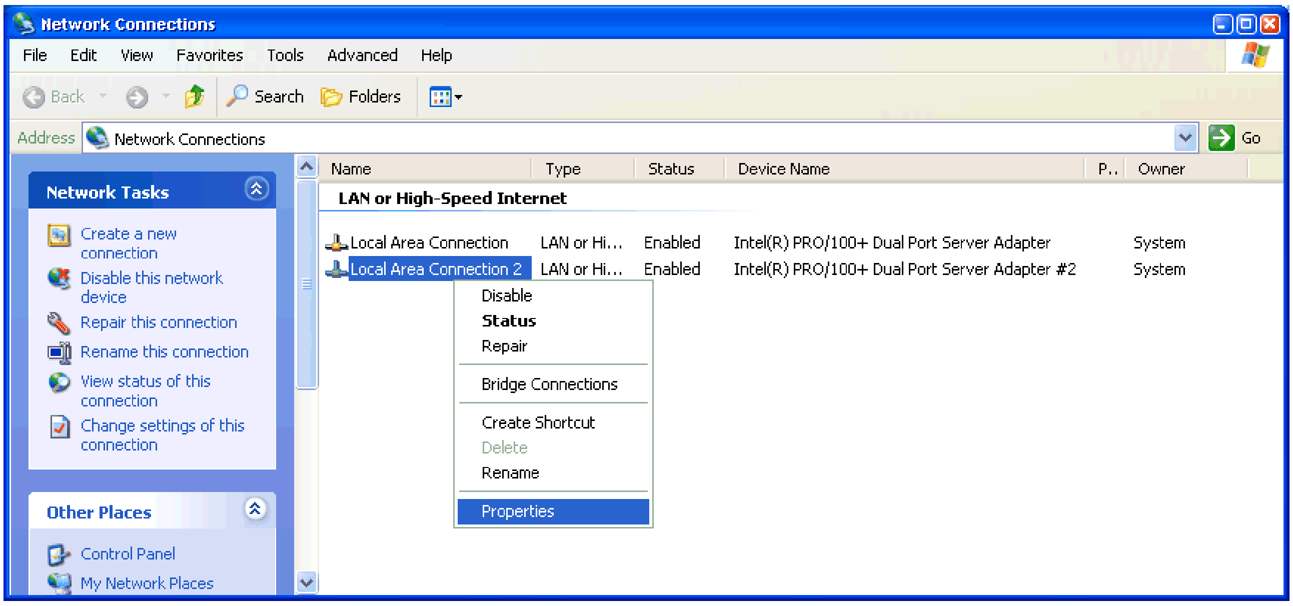


Very large applications, for which the computing power even of a 3 GHz system is insufficient, can distribute the control tasks between a number of PC controllers, and can exchange even large amounts of data quasi-synchronously with the cycles (even at sub-millisecond speeds), using the real-time capable network variables.

4 TwinCAT Ethernet Driver - Installation

The Intel 8255x based network interface card (NIC) should be plugged in and - automatically - detected and installed from Windows XP / 2000. After that, the driver of the device must be changed. Open the 'Network Connections' (via Control Panel or 'Properties' of 'My Network Places') and open the 'Properties' of the LAN connection. If more than one NIC is installed, choose the one the Real-Time Ethernet shall be connected to.

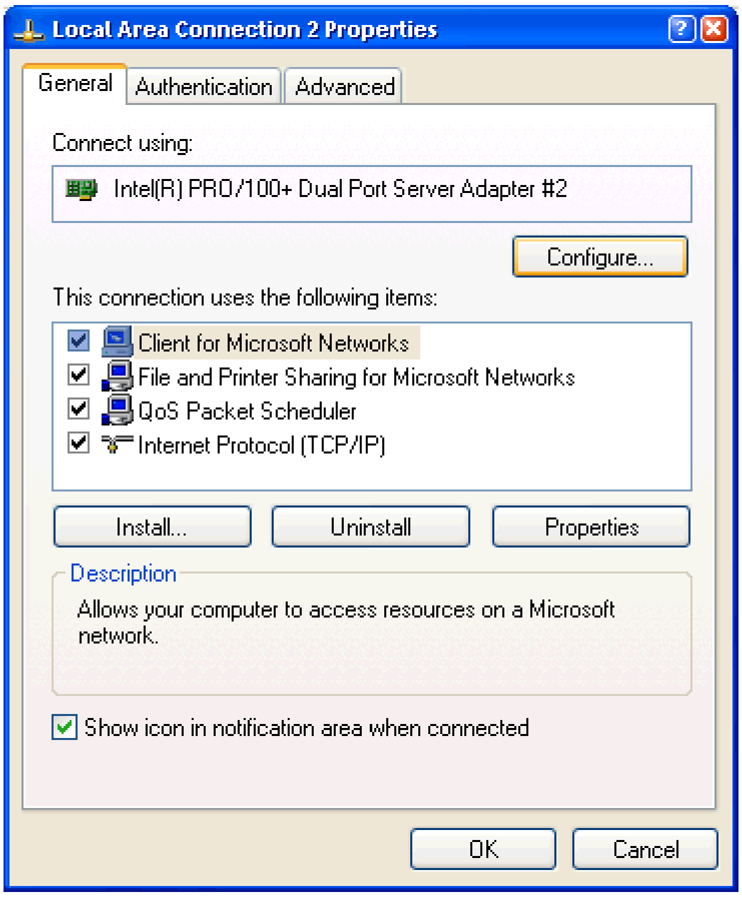
The following screenshots are showing the necessary steps for the driver setup under Windows XP Professional.



LAN or High-Speed Internet

Name	Type	Status	Device Name	P..	Owner
Local Area Connection	LAN or Hi...	Enabled	Intel(R) PRO/100+ Dual Port Server Adapter		System
Local Area Connection 2	LAN or Hi...	Enabled	Intel(R) PRO/100+ Dual Port Server Adapter #2		System

The 'Properties' page appears. Please push the 'Configure...' button.



Connect using:

Intel(R) PRO/100+ Dual Port Server Adapter #2

This connection uses the following items:

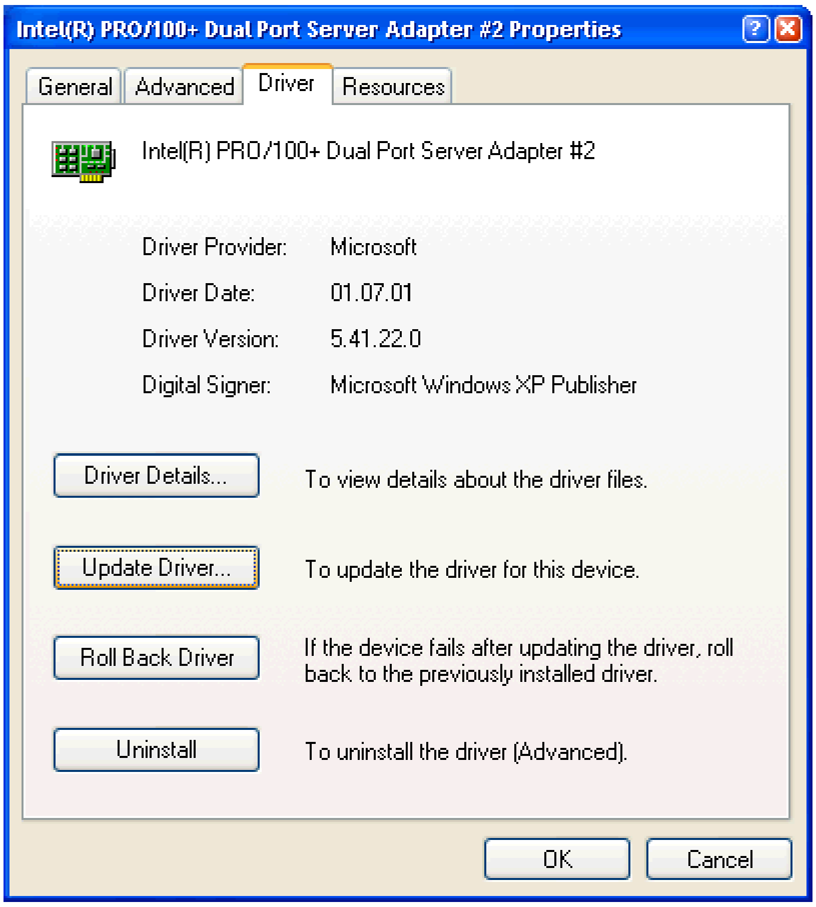
- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol (TCP/IP)

Description

Allows your computer to access resources on a Microsoft network.

☑ Show icon in notification area when connected

Select the 'Driver' Page and press the 'Update Driver...' button.



Intel(R) PRO/100+ Dual Port Server Adapter #2

Driver Provider:	Microsoft
Driver Date:	01.07.01
Driver Version:	5.41.22.0
Digital Signer:	Microsoft Windows XP Publisher

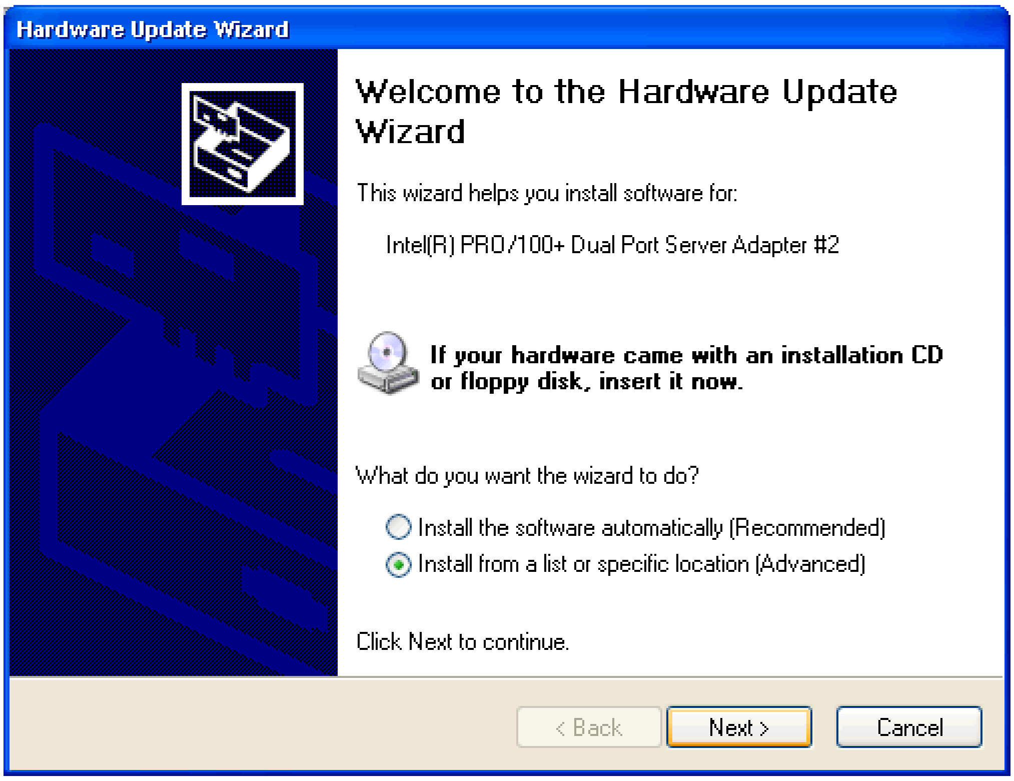
To view details about the driver files.

To update the driver for this device.

If the device fails after updating the driver, roll back to the previously installed driver.

To uninstall the driver (Advanced).

The 'Hardware Update Wizard' appears. Please select 'Install from a list or specific location (Advanced)' radio button and push 'Next'.



Welcome to the Hardware Update Wizard

This wizard helps you install software for:

Intel(R) PRO/100+ Dual Port Server Adapter #2

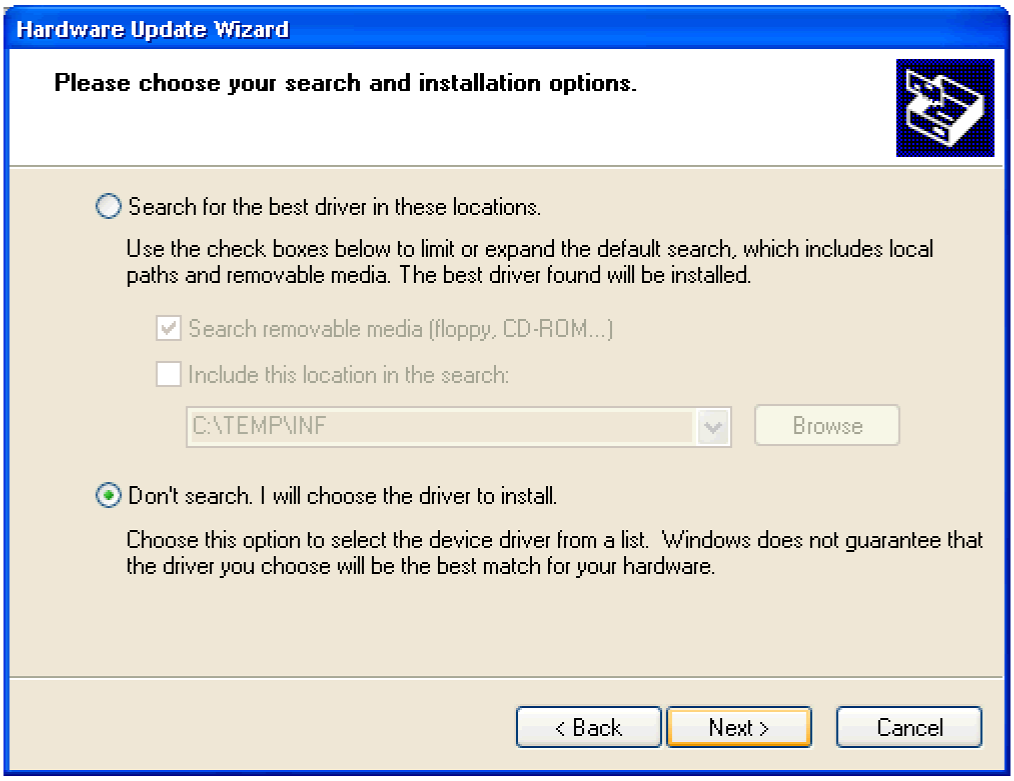
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

- ○ Install the software automatically (Recommended)
- [✓ Install from a list or specific location (Advanced)]

Click Next to continue.

On the following dialog select the 'Don't search' option and press 'Next'.


Hardware Update Wizard

Please choose your search and installation options.

○ Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

[✓] Search removable media (floppy, CD-ROM...)

[ ] Include this location in the search:

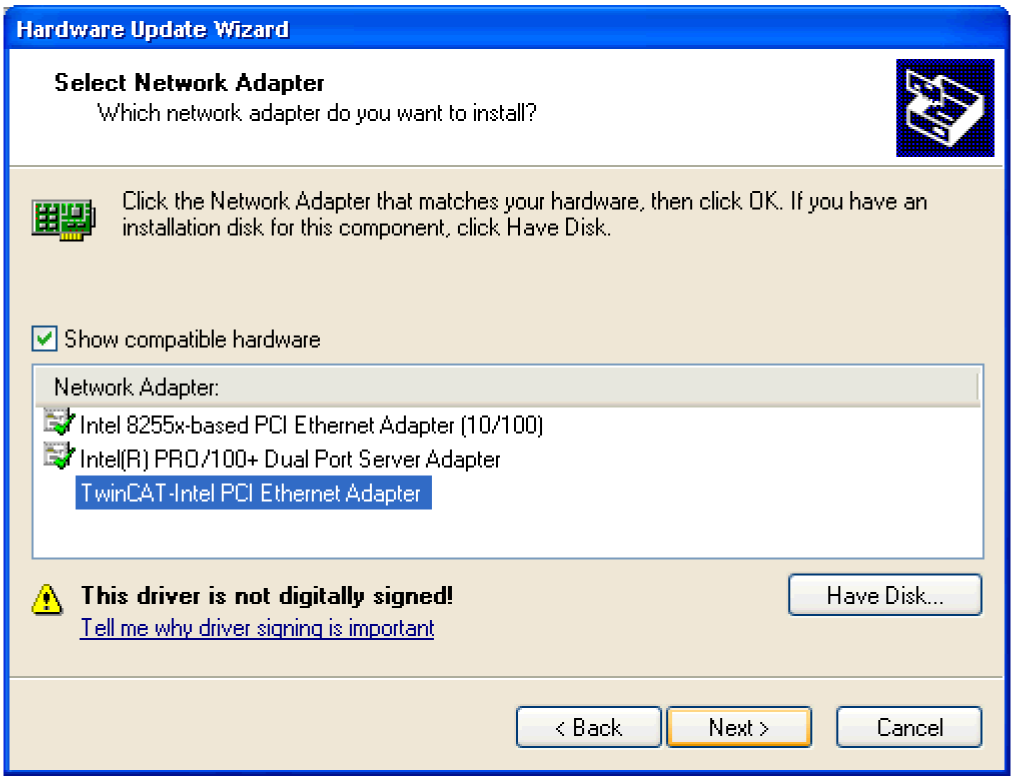
C:\TEMP\INF — Browse

● Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

< Back | Next > | Cancel

Select the 'TwinCAT-Intel PCI Ethernet Adapter' and push 'Next'


Hardware Update Wizard

Select Network Adapter

Which network adapter do you want to install?

Click the Network Adapter that matches your hardware, then click OK. If you have an installation disk for this component, click Have Disk.

[✓] Show compatible hardware

Network Adapter:
- Intel 8255x-based PCI Ethernet Adapter (10/100)
- Intel(R) PRO/100+ Dual Port Server Adapter
- TwinCAT-Intel PCI Ethernet Adapter

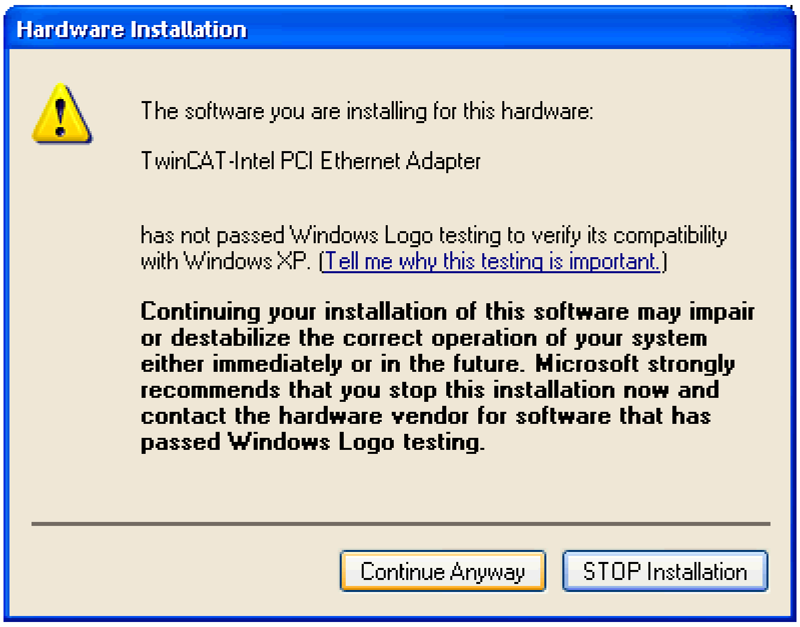
This driver is not digitally signed!

Tell me why driver signing is important

Have Disk...

< Back | Next > | Cancel

A warning will appear because the TwinCAT driver is not digitally signed by Microsoft - just trust Beckhoff...


Hardware Installation

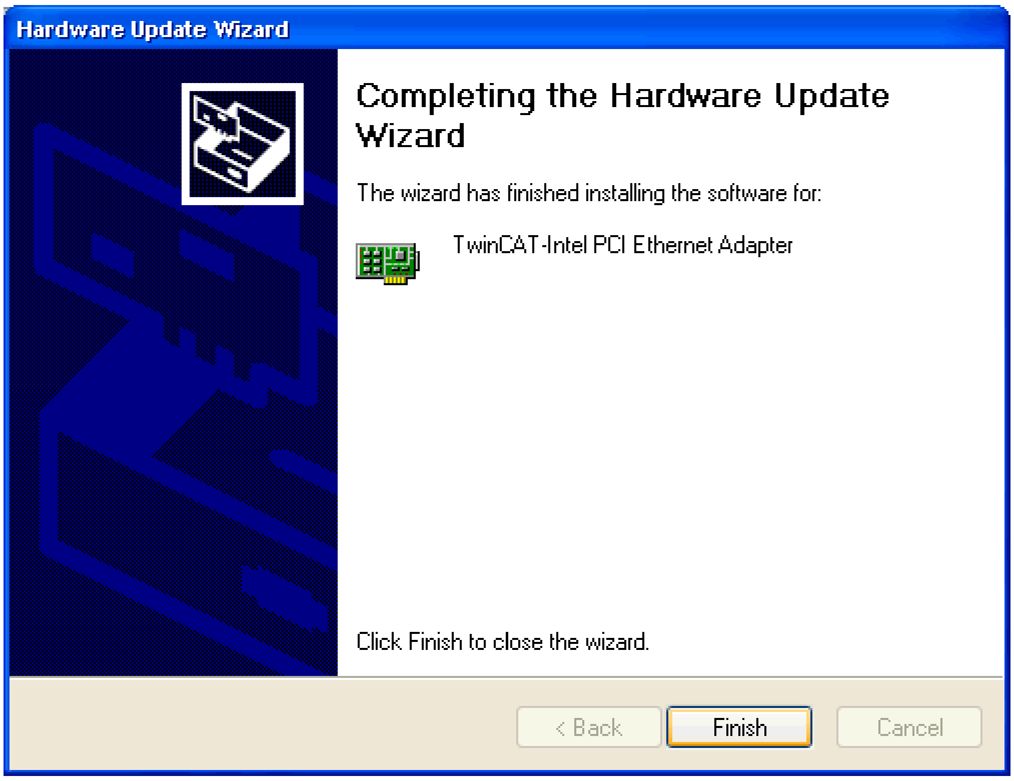
The software you are installing for this hardware:

TwinCAT-Intel PCI Ethernet Adapter

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

The installation of the driver is now finished, but the TwinCAT protocol driver has to be installed still.


Hardware Update Wizard

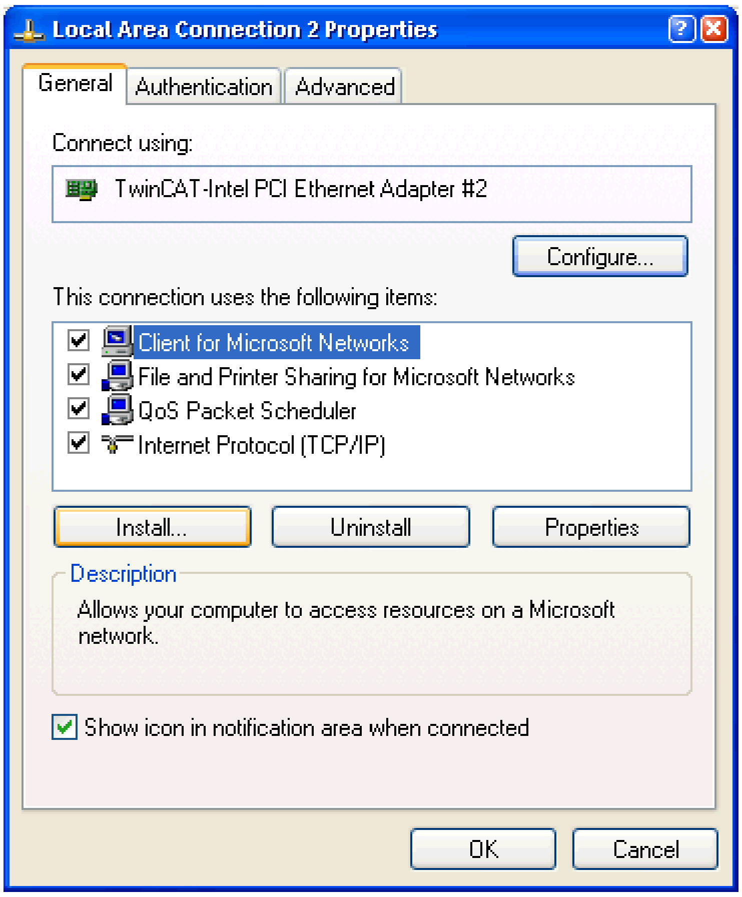
Completing the Hardware Update Wizard

The wizard has finished installing the software for:

TwinCAT-Intel PCI Ethernet Adapter

Click Finish to close the wizard.

Please re-open the 'Properties' of the LAN connection - the new adapter driver should appear.


Local Area Connection 2 Properties

General | Authentication | Advanced

Connect using:

TwinCAT-Intel PCI Ethernet Adapter #2

Configure...

This connection uses the following items:

- [✓] Client for Microsoft Networks
- [✓] File and Printer Sharing for Microsoft Networks
- [✓] QoS Packet Scheduler
- [✓] Internet Protocol (TCP/IP)

Install... | Uninstall | Properties

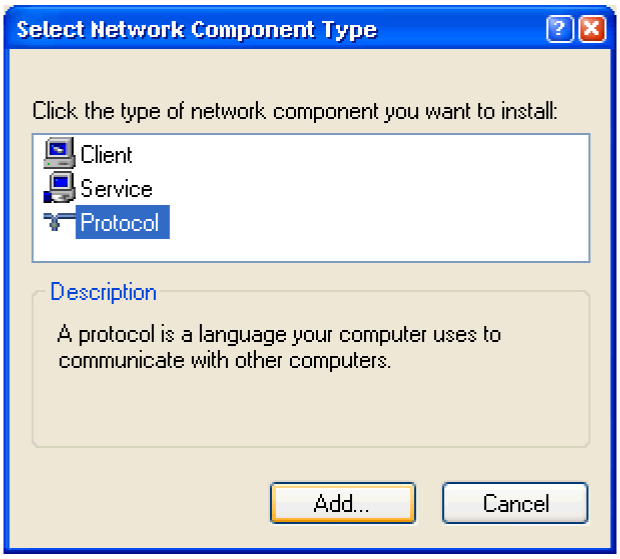
Description

Allows your computer to access resources on a Microsoft network.

[✓] Show icon in notification area when connected

OK | Cancel

Push the 'Install...' button and select the 'Protocol' choice. Press 'Add...' after this selection.


Select Network Component Type

Click the type of network component you want to install:

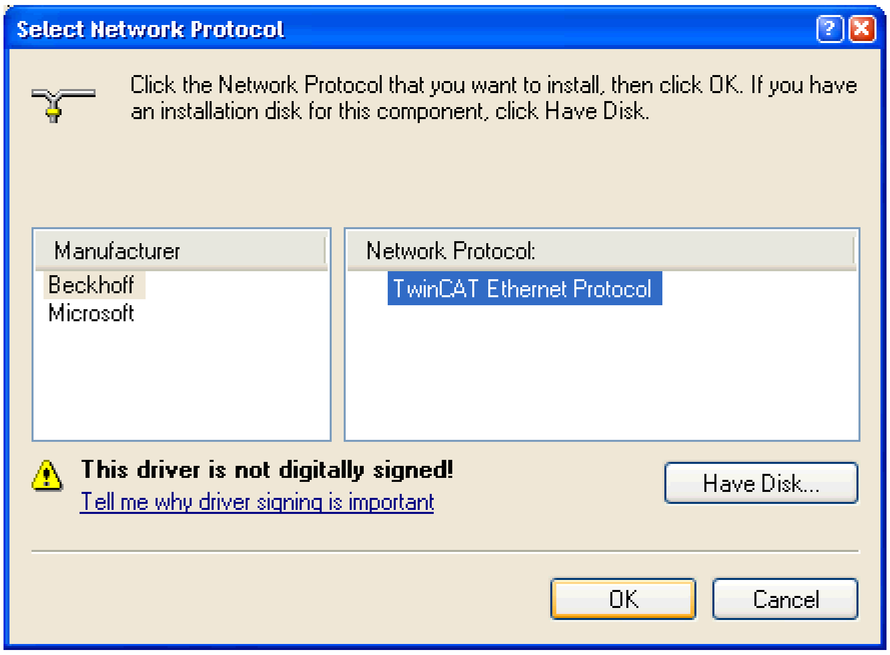
- Client
- Service
- Protocol

Description

A protocol is a language your computer uses to communicate with other computers.

Add... | Cancel

Select the 'TwinCAT Ethernet Protocol' from manufacturer 'Beckhoff'. If the protocol does not appear push the "Have Disk..." button and select the \Windows\Inf (Windows XP) or \WinNt\Inf (Windows 2000) directory. If the directory is not visible check the "Show hidden files or folder" radio button in the Explorer Tools/Folder Options/View dialog. After selecting the folder (doesn't matter which file is selected) the Beckhoff TwinCAT Ethernet Protocol should appear.

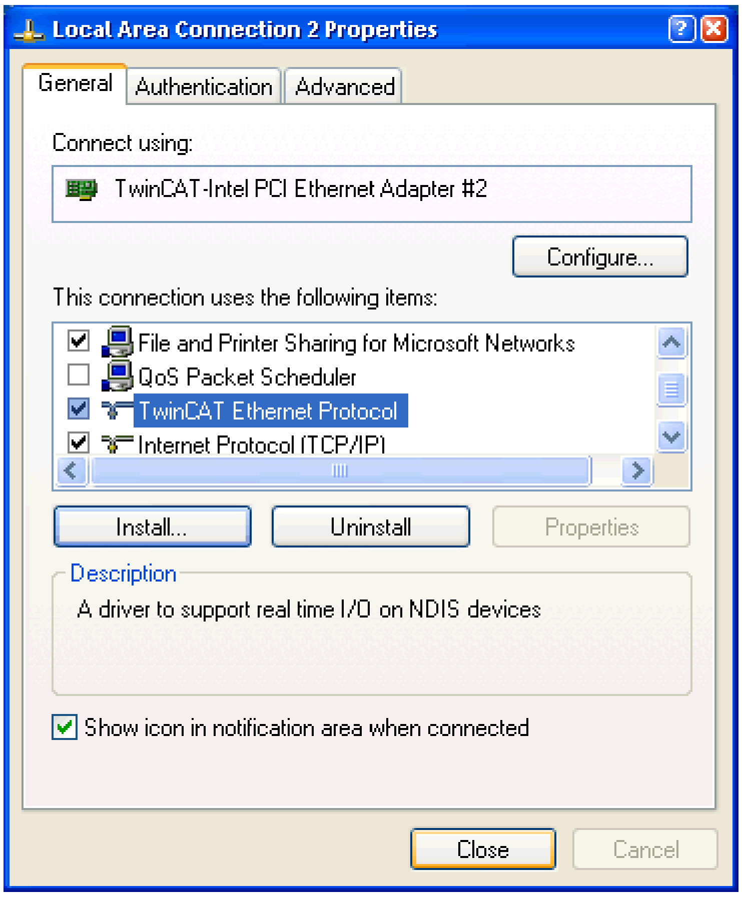

Select Network Protocol

Click the Network Protocol that you want to install, then click OK. If you have an installation disk for this component, click Have Disk.

Manufacturer	Network Protocol:
Beckhoff	TwinCAT Ethernet Protocol
Microsoft	

This driver is not digitally signed!
Tell me why driver signing is important

The new protocol should appear. The quality of service (QoS) packet scheduler filter driver can be disabled, because it has no function in reference to the TwinCAT driver.


Local Area Connection 2 Properties

General | Authentication | Advanced

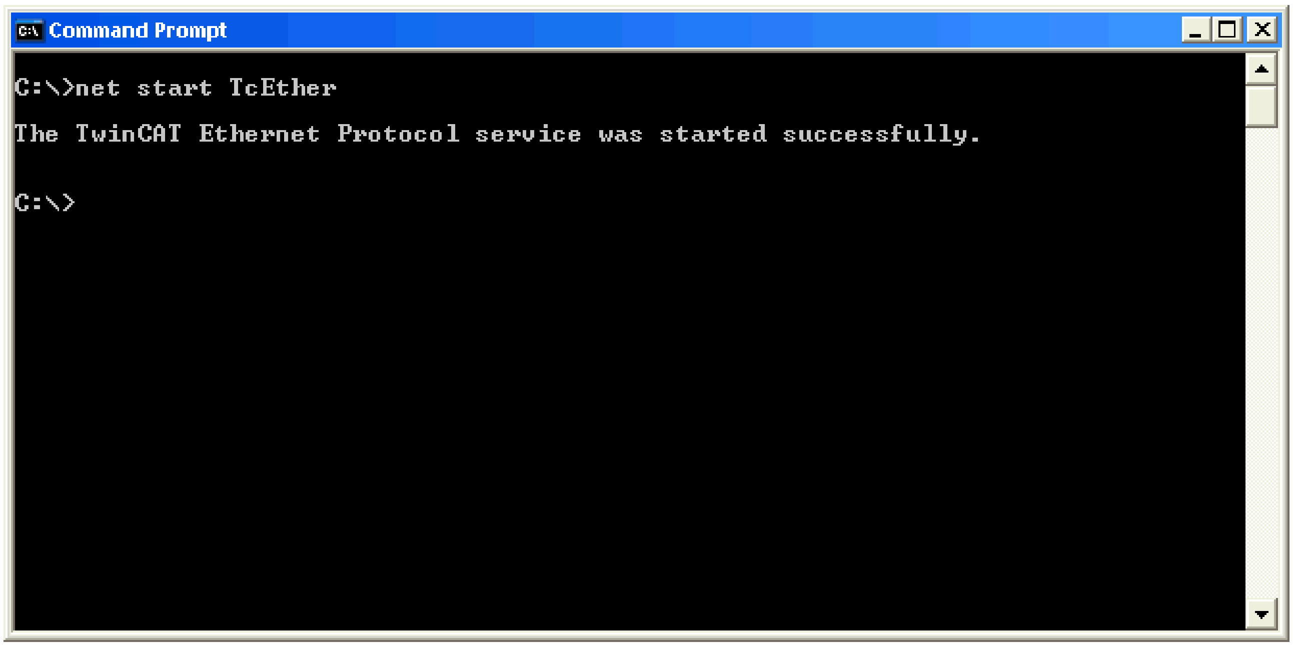
Connect using:
TwinCAT-Intel PCI Ethernet Adapter #2

This connection uses the following items:
- [✓] File and Printer Sharing for Microsoft Networks
- [ ] QoS Packet Scheduler
- [✓] TwinCAT Ethernet Protocol
- [✓] Internet Protocol (TCP/IP)

Description
A driver to support real time I/O on NDIS devices

[✓] Show icon in notification area when connected

Now the required steps are made. Only the 'TwinCAT Ethernet Protocol' did not start automatically for the first time. You have the choice to reboot the system or to start the protocol manually (only the very first time). To start the protocol manually, please open the command window (e.g. via the 'Run...' button on the start menu and enter 'cmd'). Enter in the command window the string 'net start TcEther' and press enter.



```
C:\>net start TcEther
The TwinCAT Ethernet Protocol service was started successfully.

C:\>
```

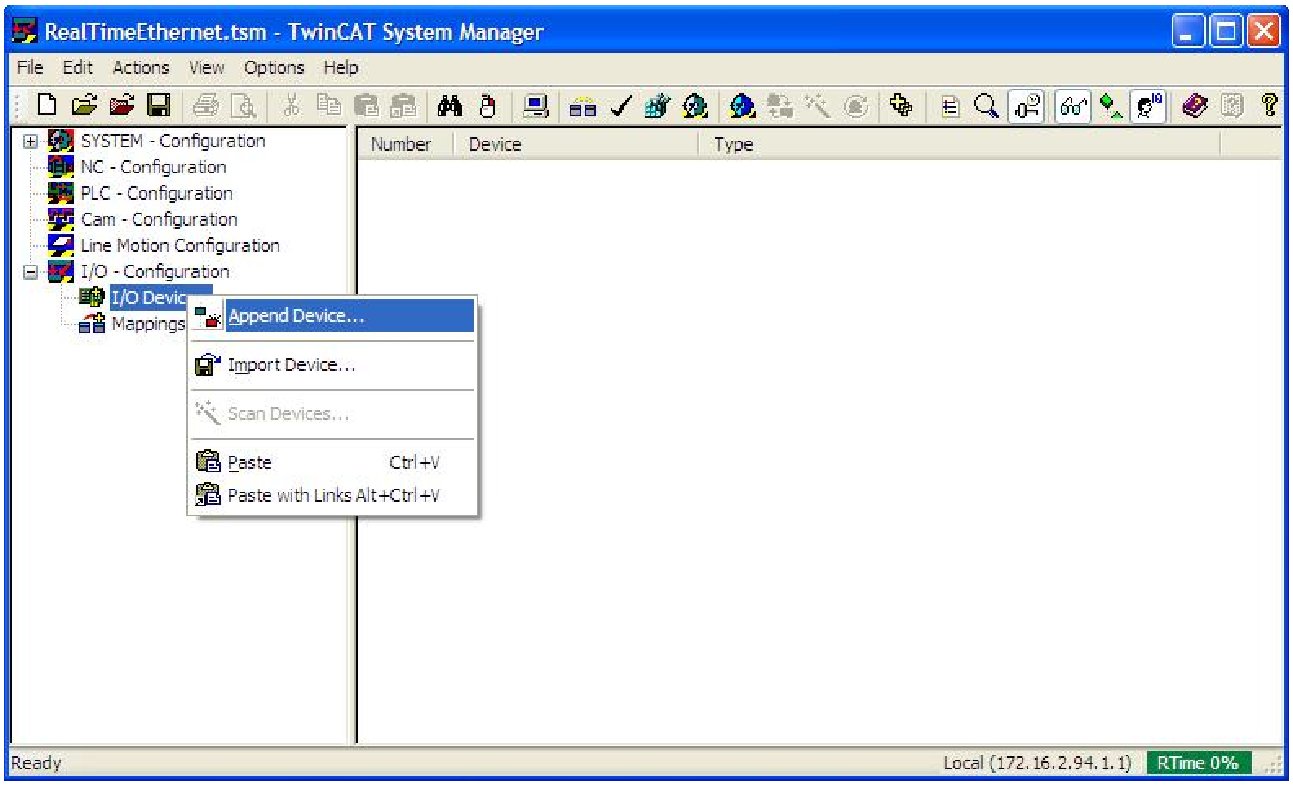
After that, the TwinCAT System Manager should find this new adapter underneath "I/O Devices".

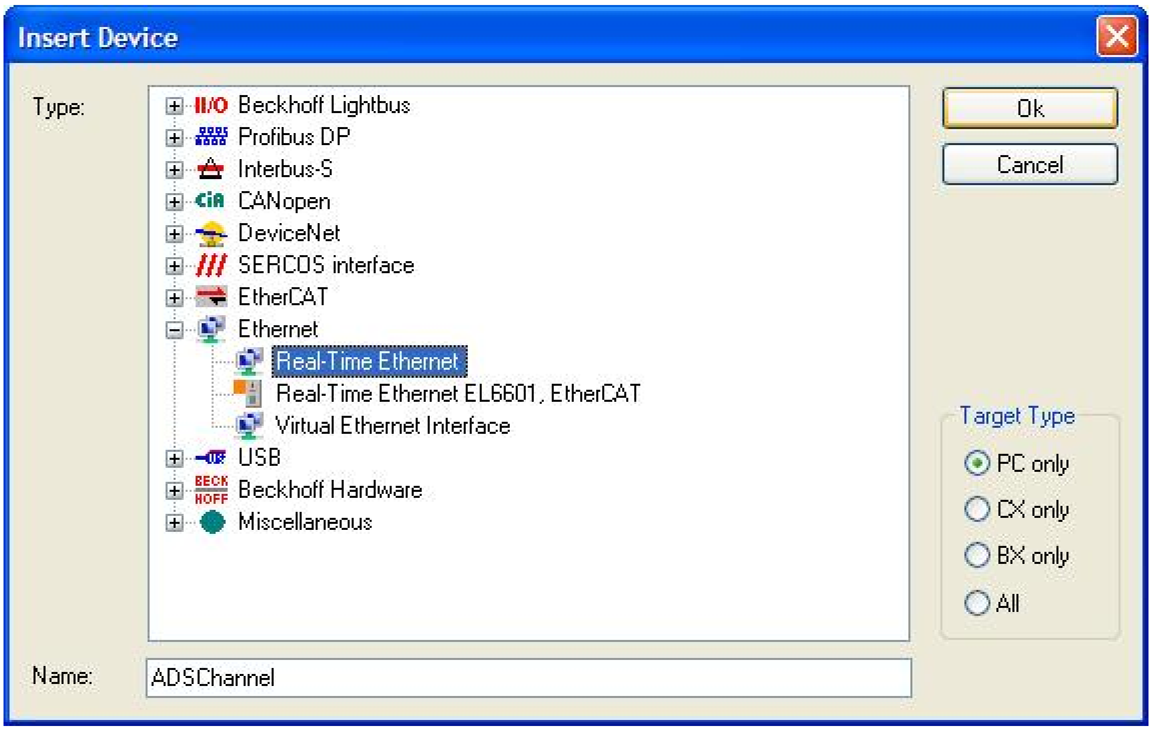
5 Configuration

Configuration example for ADS server/client communication.

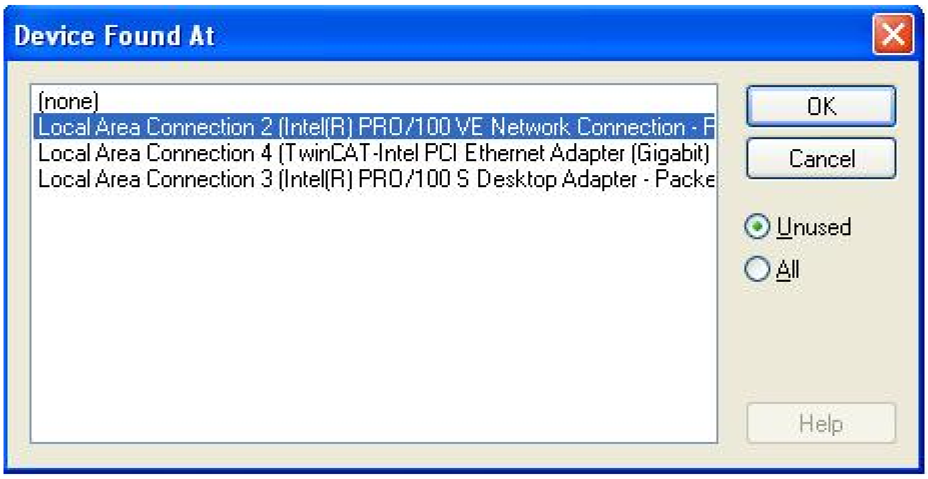
Server:

In the TwinCAT System Manager select the local target system for the server. Right-click on "I/O devices" under "I/O configuration" to insert a new device. In the window that opens you can select "Real-Time Ethernet" device under Ethernet.





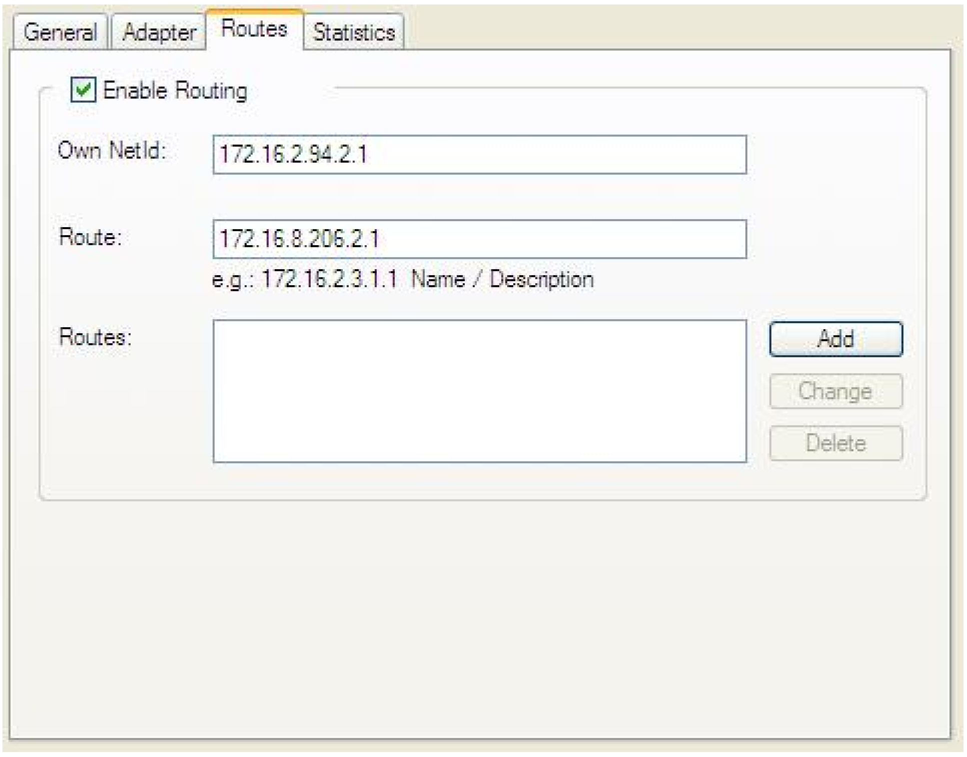
Before confirming your selection with OK, you can allocate a new name to the device at the bottom of the window. After confirmation select the appropriate network card, as shown in the next image.



(none)
Local Area Connection 2 (Intel(R) PRO/100 VE Network Connection - F
Local Area Connection 4 (TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
Local Area Connection 3 (Intel(R) PRO/100 S Desktop Adapter - Packe

[✓ Unused] All

Activate the "Enable Routing" checkbox in the "Routes" tab of new device. The NetID of the added RT Ethernet device is automatically entered in the associated field.



☑ Enable Routing

Own NetId: 172.16.2.94.2.1

Route: 172.16.8.206.2.1

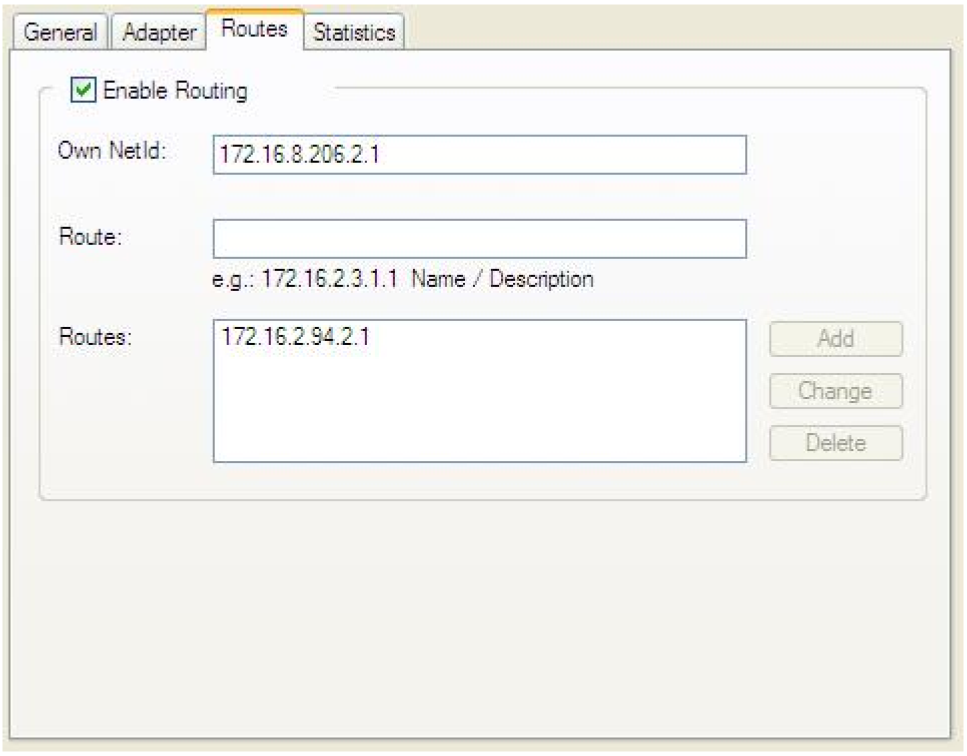
e.g.: 172.16.2.3.1.1 Name / Description

Routes:

In the next step you can enter the NetID of a client under Route, and under "Add" you can add the NetID to other NetIDs with which the server can then communicate.

Client:

The same settings apply to the clients as for the server, except that the NetID of the server must be specified under "Routes" in the real-time Ethernet device.



☑ Enable Routing

Own NetId: 172.16.8.206.2.1

Route:

e.g.: 172.16.2.3.1.1 Name / Description

Routes: 172.16.2.94.2.1

5.1 Example

Based on the configuration described above you can now carry out a test with the following PLC project. In this example the server fetches a handle via ADS for a client variable and writes the value 500 to it. A useful application is shown in conjunction with ADS.

Server:

```
PROGRAM Server
VAR
    state        : UINT;
    trigger      : R_TRIG;
    bExecute     : BOOL;
    bBusy        : BOOL;
    bError       : BOOL;
    uErrorID     : UDINT;

    fbWRI        : ADSWRITE;
    fbRW         : ADSRDWRT;

    uTransmit    : UDINT := 500;
    uHandle      : UDINT;
    sTest        : STRING := 'MAIN.uReceive';
END_VAR

trigger(CLK := bExecute);
IF trigger.Q THEN
    state := 1;
    bBusy := TRUE;
END_IF
CASE state OF

    0: (* Wait state *)
    bBusy := FALSE;

    1:
    fbRW(    WRTRD    := FALSE);
    fbRW(    NETID    := '172.16.2.94.2.1',
           PORT       := 801,
           IDXGRP     := 16#F003,
           IDXOFFS    := 16#0000,
           WRITELEN   := LEN(sTest)+1,
           READLEN    := SIZEOF(uHandle),
           SRCADDR    := ADR(sTest),
           DESTADDR  := ADR(uHandle),
           WRTRD     := TRUE,
           TMOUT     := t#5s,
           BUSY       => ,
           ERR        => bError,
           ERRID      => uErrorID);
           state     := 2;

    2:
    fbRW(     WRTRD     := FALSE);
    IF NOT fbRW.BUSY THEN
        IF NOT fbRW.ERR THEN
            state := 3;
        ELSE
            state := 100;
        END_IF
    END_IF

    3:
    fbWRI(    WRITE      := FALSE);
    fbWRI(    NETID      := '172.16.2.94.2.1',
              PORT       := 801,
              IDXGRP     := 16#F005,
              IDXOFFS    := uHandle,
              LEN        := SIZEOF(uTransmit),
              SRCADDR    := ADR(uTransmit),
              WRITE      := TRUE,
              TMOUT      := t#5s,
              BUSY        => ,
              ERR         => bError,
              ERRID       => uErrorID);
              state      := 4;

    4:
    fbWRI(WRITE := FALSE);
    IF NOT fbWRI.BUSY ANDNOT fbWRI.ERR THEN
        state := 0;
    ELSIF fbWRI.ERR THEN
        state := 100;
    END_IF

    100:
    ; (* Error *)
END_CASE
```

Client:

```
PROGRAM Client
VAR
    uReceive : UDINT;
END_VAR
```

- On the client-side creation of a UDINT variable is sufficient.
- To use the source code for a test you must adapt the NetIDs to your system.

6 Example

Based on the configuration described above you can now carry out a test with the following PLC project. In this example the server fetches a handle via ADS for a client variable and writes the value 500 to it. A useful application is shown in conjunction with ADS.

Server:

```
PROGRAM Server
VAR
    state         : UINT;
    trigger       : R_TRIG;
    bExecute      : BOOL;
    bBusy         : BOOL;
    bError        : BOOL;
    uErrorID      : UDINT;

    fbWRI         : ADSWRITE;
    fbRW          : ADSRDWRT;

    uTransmit     : UDINT := 500;
    uHandle       : UDINT;
    sTest         : STRING := 'MAIN.uReceive';
END_VAR

trigger(CLK := bExecute);
IF trigger.Q THEN
    state := 1;
    bBusy := TRUE;
END_IF
CASE state OF

    0: (* Wait state *)
    bBusy := FALSE;

    1:
    fbRW(    WRTRD    := FALSE);
    fbRW(    NETID    := '172.16.2.94.2.1',
           PORT       := 801,
           IDXGRP     := 16#F003,
           IDXOFFS    := 16#0000,
           WRITELEN   := LEN(sTest)+1,
           READLEN    := SIZEOF(uHandle),
           SRCADDR    := ADR(sTest),
           DESTADDR   := ADR(uHandle),
           WRTRD      := TRUE,
           TMOUT      := t#5s,
           BUSY        => ,
           ERR         => bError,
           ERRID       => uErrorID);
           state      := 2;

    2:
    fbRW(     WRTRD     := FALSE);
    IF NOT fbRW.BUSY THEN
        IF NOT fbRW.ERR THEN
            state := 3;
        ELSE
            state := 100;
        END_IF
    END_IF

    3:
    fbWRI(    WRITE       := FALSE);
    fbWRI(    NETID       := '172.16.2.94.2.1',
              PORT        := 801,
              IDXGRP      := 16#F005,
              IDXOFFS     := uHandle,
              LEN         := SIZEOF(uTransmit),
              SRCADDR     := ADR(uTransmit),
              WRITE       := TRUE,
              TMOUT       := t#5s,
              BUSY         => ,
              ERR          => bError,
              ERRID        => uErrorID);
              state       := 4;

    4:
```

```
    fbWRI(WRITE := FALSE);
    IF NOT fbWRI.BUSY ANDNOT fbWRI.ERR THEN
        state := 0;
    ELSIF fbWRI.ERR THEN
        state := 100;
    END_IF

    100:
    ; (* Error *)
END_CASE
```

Client:

```
PROGRAM Client
VAR
    uReceive : UDINT;
END_VAR
```

- On the client-side creation of a UDINT variable is sufficient.
- To use the source code for a test you must adapt the NetIDs to your system.

More Information:
www.beckhoff.com/automation

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com