

Handbuch | DE

# Real-Time Ethernet



TwinCAT 2 | Connectivity





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort.....</b>	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Sicherheitshinweise .....	6
1.3	Hinweise zur Informationssicherheit .....	7
<b>2</b>	<b>Übersicht.....</b>	<b>8</b>
<b>3</b>	<b>Einführung von TwinCAT-Echtzeit-Ethernet .....</b>	<b>9</b>
<b>4</b>	<b>TwinCAT Ethernet Treiber - Installation.....</b>	<b>14</b>
<b>5</b>	<b>Konfiguration.....</b>	<b>21</b>
5.1	Beispiel.....	23
<b>6</b>	<b>Beispiel .....</b>	<b>25</b>



# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

## EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

## 1.2 Sicherheitshinweise

### Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!  
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

### Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

### Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

### Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

#### **GEFAHR**

##### **Akute Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

#### **WARNUNG**

##### **Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

#### **VORSICHT**

##### **Schädigung von Personen!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

#### **HINWEIS**

##### **Schädigung von Umwelt oder Geräten**

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

#### **Tipp oder Fingerzeig**



Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

## 1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

## 2 Übersicht

Mit ADS over Real-Time Ethernet ist es möglich eine echtzeitfähige Kommunikation über einen UDP-Kanal und der Ethernet-Physik aufzubauen.

### Systemvoraussetzungen

- W2K, XP, Windows Embedded Standard;
- TwinCAT Installation Level: TwinCAT PLC oder höher;
- TwinCAT System Version 2.10.0 Build >= 1328;

### Installation

Voraussetzung für Real-Time Ethernet ist, dass alle Kommunikationsteilnehmer den TwinCAT Ethernet Protokolltreiber [[▶ 14](#)] installiert haben.

### Weiterführende Dokumentation

- Einführung von TwinCAT-Echtzeit-Ethernet [[▶ 9](#)]
- Netzwerk Variablen (Publisher/Subscriber)

## 3 Einführung von TwinCAT-Echtzeit-Ethernet

### Ethernet Echtzeit-Kommunikation

Ethernet meistert die nächste Hürde als „Feldbus“ in der Beckhoff TwinCAT-Steuerung. Neben der Erfüllung hoher Echtzeit-Anforderungen ermöglicht es auch den Einsatz von Standardkomponenten „in derselben Leitung“. Der BK9000 Ethernet-Buskoppler und der AX2000-B900 Servoverstärker besitzen als erste Feldbuskomponenten Echtzeitfähigkeit. Neue Netzwerkvariablen beschleunigen den Echtzeit-Datenaustausch zwischen den einzelnen Controllern; eine Implementierung ist so einfach wie der Anschluss eines weiteren digitalen Eingangs.

Beckhoff Ethernet-Produkte sind seit vielen Jahre bewährt und erfreuen sich immer größerer Beliebtheit. Die Vorteile des Einsatzes alltäglicher Ethernet Standardlösungen in Industrieanlagen liegen auf der Hand:

- **Einsatz von Standard-Hardwarekomponenten (z.B. marktüblicher Standard-)Switch**
- **Standardprotokolle** können eingesetzt werden
- **Die Datenübertragungsraten sind im Vergleich zu anderen Netzwerken sehr hoch**
- **Das Netzwerk kann einfach über das Internet mit dem Rest der Welt verbunden werden**
- **Fernwartung und –diagnose**

**Kommunikation über Ethernet** ist in der industriellen Automation bereits weitgehend akzeptiert; viele Gruppen und Ausschüsse beschäftigen sich mit diesem Thema. Durch die fehlende Echtzeitfähigkeit war der Einsatz von Ethernet im klassischen Feldbusbereich jedoch eingeschränkt. Es gibt einige Techniken, die eine gewisse Echtzeitfähigkeit ermöglichen. Diese basieren aber auf selbst entwickelten Systemen, die nicht parallel auf Standardkomponenten und Protokolle zugreifen können.

- Der Begriff „Echtzeitfähigkeit“ wird in der Steuerungstheorie unterschiedlich interpretiert.
- Was „Echtzeit“ ist, hängt in hohem Maße von den Anforderungen der jeweiligen Anwendung ab, ebenso wie von den Regelkreisen, in denen die Automatisierungskomponenten eingesetzt werden. Auf Grundlage der Automatisierungstechnik und vor dem Hintergrund dessen, was Feldbusspezialist Beckhoff anbietet, kann man eine grobe Einteilung vornehmen:
- **Die höchsten Anforderungen** betreffen Zykluszeiten von ca. 50  $\mu$ s und einen zulässigen Jitter (Abweichungen von der gewünschten Zykluszeit) von ca. 10  $\mu$ s. Noch höhere Anforderungen werden derzeitig noch eher mit spezieller Hardware realisiert als direkt mit Feldbussen. Die Anforderungen an Zykluszeiten für positionsgesteuerte Antriebe bewegen sich im Allgemeinen im Millisekundenbereich (1-4 ms), die Jitterzeiten sollten unter 20  $\mu$ s liegen. Reine SPS-Anwendungen erfordern oft Zykluszeiten von mindestens 10 ms; die Jitterzeiten können hier also auch entsprechend länger sein und sich im Millisekundenbereich bewegen. Der Datenaustausch zwischen Controller und Überwachungssystem kann oft mit Jitterzeiten im Sekundenbereich zufriedenstellend realisiert werden. Er muss hier nicht unbedingt zyklisch konfiguriert werden, sondern kann ereignisgesteuert sein.

Ein weiterer wichtiger Punkt ist die **Fernwartung und –diagnose**. Hier sind die Zyklus- und Jitterzeiten weniger wichtig als die Reaktionszeiten und die generelle Möglichkeit, über Grenzen eines Netzwerks hinaus zu kommunizieren. TwinCAT-Automatisierungssoftware mit Echtzeit-Ethernet heißt, dass alle oben erwähnten Kommunikationsanforderungen mit derselben Technologie erfüllt werden, sowohl hinsichtlich der eingesetzten Geräte, als auch der genutzten Protokolle.

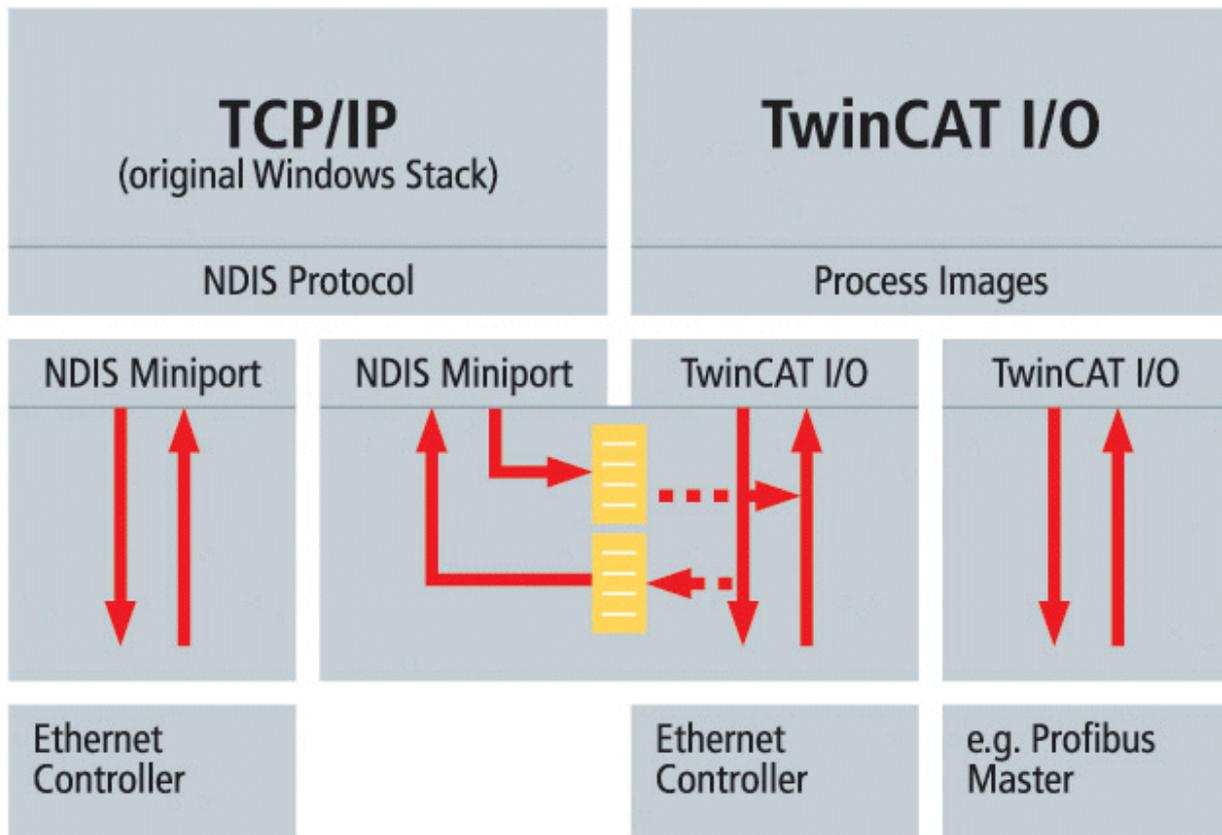
### Funktionsprinzip

Der Treiber der TwinCAT-Netzwerkkarte ist so in das System eingebunden, dass er als Betriebssystem konformer Netzwerktreiber und zusätzlich als TwinCAT Feldbuskarte erscheint. Auf Sendeseite wird über interne Priorisierung und Puffer sichergestellt, dass anstehende Ethernet-Frames des Echtzeitsystems immer eine freie Sendeleitung vorfinden. **Ethernet-Frames des Betriebssystems werden erst später verschickt, in den „Lücken“, wenn genügend Zeit ist.**

Auf Empfangsseite werden alle empfangenen Ethernet-Frames vom TwinCAT I/O System überprüft und die Echtzeit relevanten herausgefiltert. Alle anderen Frames werden nach Überprüfung an das Betriebssystem übergeben, ohne Kontext zum Echtzeitsystem.

Durch handelsübliche Switches, die alle einen Vollduplexbetrieb bei 100 Mbaud unterstützen, werden die übertragenen Frames mit konstanter Verzögerung an den Empfänger weitergeleitet. Ein Switch stellt sicher, dass Kollisionen vermieden werden und nur Verzögerungen auftreten. Daher muss in einem zyklischen Steuerungssystem nur gewährleistet sein, dass alle wichtigen Eingangsinformationen angekommen sind,

bevor der nächste Zyklus startet. Wann oder in welcher Reihenfolge sie ankommen spielt dabei keine Rolle. Voraussetzung für eine Echtzeit-Ethernetkommunikation ist, dass die Anzahl von Teilnehmern oder die Framerate entsprechend der erforderlichen Zykluszeit beschränkt ist.



Das Bild oben zeigt das Prinzip, wie der original Windows-Protokollstack (TCP/IP und UDP/IP) parallel zum TwinCAT-Echtzeitethernet benutzt wird. Dieses Prinzip bezeichnen wir als „Y-Treiber“-Architektur.

### Betriebsarten/Protokolle

Im Gegensatz zu den weit verbreiteten TCP/IP- und UDP/IP-Protokollen, die weltweit für die Zustellung individueller Ethernet-Frames verantwortlich sind, geht Echtzeitkommunikation nie über das lokale Subnetz hinaus. Der Overhead aus TCP/IP und UDP/IP ist unnötig, Geräte können direkt über die Hardware-Adressen (MAC-ID) der Netzwerkkarten adressiert werden. Die Struktur der Ethernet Frames stellt sicher, dass eine Koexistenz mit anderen Protokollen immer möglich ist; auch „Echtzeit-Frames“ können mit TCP oder UDP übertragen werden, wenn sie ihr eigenes Subnetz verlassen müssen.

Für den Einsatz in der Steuerungstechnik wurde eine Reihe von Betriebsarten definiert. Sie haben unterschiedliche Kommunikationsaufgaben und können natürlich auch simultan eingesetzt werden.

#### 1.) Master - Slave Prozessdatenkommunikation

Zyklische oder ereignisgesteuerte Übertragung von I/O-Daten – die typische Anwendung moderner Feldbusse

#### 2.) Publisher - Subscriber Prozessdatenkommunikation

Prozessdaten nach dem Publisher-Subscriber-Modell (auch Netzwerkvariablen genannt) werden für die reguläre Kommunikation zwischen Controllern verwendet, wenn ein feststehendes Master-Slave-Verhältnis unangemessen wäre. Der Publisher sendet Informationen wahllos ohne Zielangabe aus. Die Kommunikation wird nur vom Subscriber überwacht. Gegenseitige Verhältnisse zwischen Publishern und Subscribern erlauben bidirektionale und multidirektionale Kommunikation. Der Publisher kann so konfiguriert werden,

dass Daten per Broadcast, Multicast oder Unicast gesendet werden. Multicast verringert die Belastung der Eingangswarteschlangen der Netzwerkgeräte, da die Meldungen sofort bei Ankunft am Ethernetcontroller bewertet werden. Nur bei Unicast kann der Switch (ohne umfangreiche Konfiguration) parallele Kommunikationspfade öffnen und die nutzbare Bandbreite erhöhen.

### 3.) Datenkommunikation nach Bedarf

Diese Art der Kommunikation wird im TwinCAT-System durch ADS-Kommunikation ermöglicht. Sie sendet Kommunikationsstrings „bei Bedarf“ von einem Gerät zum anderen. So werden Dienste ausgeführt und Parameter ausgetauscht.

**Dank der gewählten Protokollstruktur können andere Betriebsarten oder Kommunikationsprofile zukünftig leicht integriert werden und problemlos mit den vorhandenen Betriebsarten koexistieren.**

### Kompatible Komponenten

Die ersten Beckhoff-Komponenten, die für die Echtzeit-Ethernetanwendung erweitert wurden, sind der BK9000 -Ethernet Buskoppler und der AX2000-B900-Servoverstärker. Mit diesen beiden Komponenten kann fast der ganze Bereich an Industriesignalen und –anwendungen abgedeckt werden.

Alle TwinCAT-Controller (ab Version 2.9) sind kompatibel und können sowohl als „Feldbusmaster“ als auch bei Kommunikationen mit Netzwerkvariablen eingesetzt werden. TwinCAT Echtzeit-Ethernet unterstützt alle Controller (Netzwerkadapter) der Intel 8255x-Familie. Das sind die meistverbreiteten Ethernetcontroller. Sie sind Komponenten des aktuellen Intel Chip-Sets, das bereits eine kompatible Netzwerkverbindung enthält. Angesichts der großen Popularität der Intel-Familie und der Tatsache, dass sie sogar mit Gigabit Ethernet kompatibel ist (Intel 8254x-Familie), kann eine Unterstützung anderer Ethernetcontroller zukünftig zwar in Betracht gezogen werden, ist jedoch nicht unbedingt notwendig. Der neue Embedded-PC CX1000 Kleinststeuerung wird natürlich immer mit einem entsprechenden Ethernetcontroller kombiniert.

### Leistung

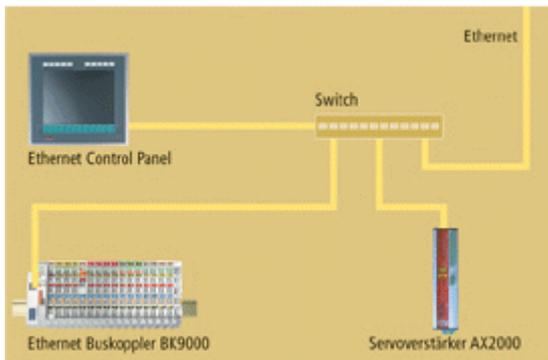
Eine Beurteilung der Leistung eines Feldbussystems ausschließlich auf Basis der Baudrate wäre zweifellos zu einseitig, da andere wichtige Kommunikationsparameter wie z.B. Protokolleffizienz, Reaktionszeit, Jitter, minimale Telegrammlänge, Pausen usw. einfach ignoriert würden. Trotzdem erlaubt 100 MBaud eine bedeutend schnellere Datenübertragung als derzeit in Feldbusumgebungen üblich. Zudem bieten die eingesetzten Protokolle, insbesondere bei kurzen Datentelegrammen, eine deutlich erhöhte Effizienz gegenüber der TCP- oder UDP-Kommunikation .

Ein Faktor, der bei der Betrachtung der Kapazität eines Feldbussystems oft ignoriert wird, ist die Datenübertragung zwischen Controller-CPU und Kommunikationschip oder –prozessor, d.h. zwischen Hostsystemspeicher und Subsystemspeicher. Bei PC-basierten Controllern werden die Daten meist via ISA- oder PCI-Bus in den DPRAM einer Feldbus-Masterkarte kopiert und umgekehrt. Da PC-Prozessoren heute Geschwindigkeiten von bis zu 3 GHz erreichen, ist selbst der „schnelle“ PCI-Bus zu einem Engpass geworden; 20-30 % der CPU-Leistung gehen allein bei PCI-Übertragung verloren.

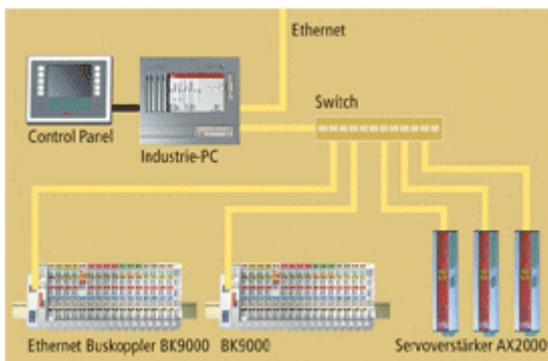
Moderne Netzwerkcontroller arbeiten im sogenannten „Busmaster DMA-Modus“, greifen also direkt auf den Speicher des Hostsystems zu. Das erfolgt parallel zu anderen CPU-Tasks, wodurch die CPU-Auslastung deutlich verringert werden kann.

### Anwendungen/Anbindung an TwinCAT

Die hohe Datenübertragungskapazität, die fundamentale Echtzeitfähigkeit und die eingesetzten Protokolle decken alle Kommunikationsanforderungen einer schnellen Maschinensteuerung ab. Betrachtet man all diese Eigenschaften, liegt die Schlussfolgerung nahe, dass herkömmliche Feldbusse überflüssig sind. Allerdings muss sich die Ethernet-Technologie erst noch auf Feldebene bewähren und die dortigen Anforderungen erfüllen: z.B. einfache Installation und Konfiguration, gegenseitige Kompatibilität, EMV-Festigkeit und nicht zuletzt Effizienz und Gerätekosten, die für einen industriellen Einsatz relevant sind. Standard-Feldbusse sind weitverbreitet und es gibt viele Geräte von vielen Anbietern (u.a. Beckhoff). Deshalb werden sie am Markt weiterhin einen großen Stellenwert haben. Auch hier bietet das flexible I/O-System von TwinCAT eine Lösung: Mit ihm können mehrere Feldbusse völlig transparent gegenüber der Anwendung parallel arbeiten – natürlich auch herkömmliche Feldbusse parallel zu Echtzeit-Ethernet.

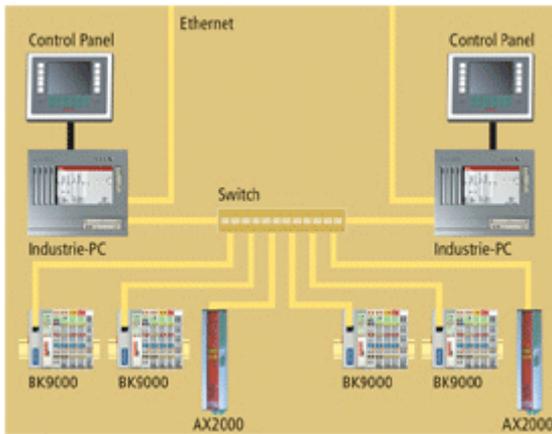
**Anwendungsbeispiel 1:**

Für relativ einfache Anwendungen ist nur eine Ethernetverbindung erforderlich, um sowohl die Echtzeitkommunikation zur I/O-Ebene als auch die übergeordnete Kommunikation für Verwaltung und Ferndiagnose zu realisieren. Die verwendete Priorisierung stellt sicher, dass die Echtzeitkommunikation problemlos erfolgt.

**Anwendungsbeispiel 2:**

Bei größeren Anwendungen wird eine zweite Ethernetverbindung eingerichtet, die die Echtzeitkommunikation und die übergeordnete Kommunikation zwischen zwei Netzwerken voneinander trennt. Das Routing, das beispielsweise für eine Ferndiagnose an einem mit einem Echtzeitnetzwerk verbundenen Treiber benötigt wird, wird vom IP-Stack des Betriebssystems automatisch ausgeführt und erfordert keine spezielle Konvertierung in ein anderes Protokoll.

**Anwendungsbeispiel 3:**

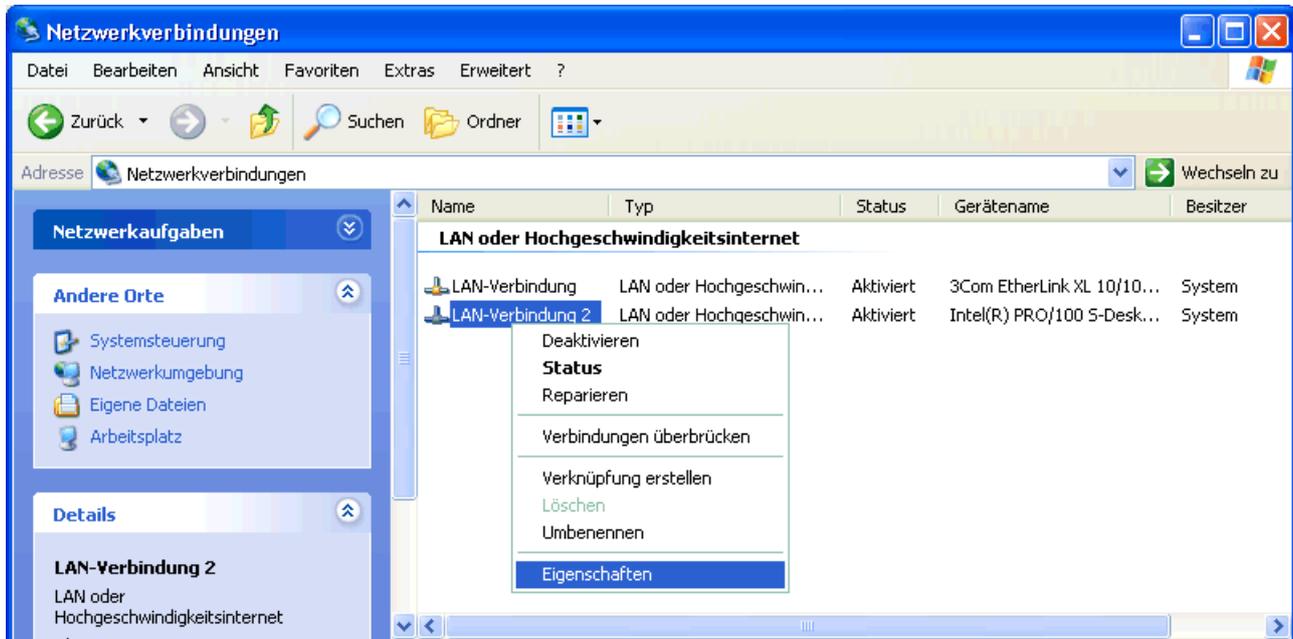


Bei sehr großen Anwendungen, für die auch die Rechenleistung eines 3 GHz-Systems nicht ausreicht, können die Steuerungsaufgaben zwischen mehreren PC-Controllern aufgeteilt werden. So können mit echtzeitfähigen Netzwerkvariablen selbst große Datenmengen quasi zeitgleich mit den Zyklen ausgetauscht werden (selbst bei Geschwindigkeiten unter einer Millisekunde).

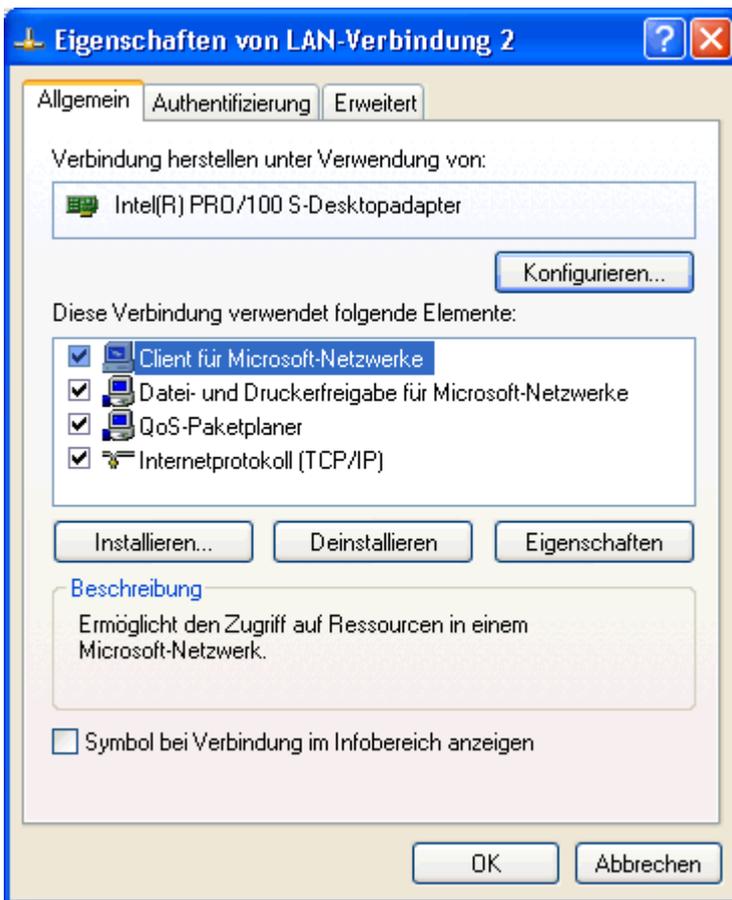
## 4 TwinCAT Ethernet Treiber - Installation

Der auf Intel 8255x basierende Netzwerkadapter (NIC) sollte nach dem Einbau automatisch von Windows XP / 2000 gefunden werden. Nach der automatischen Erkennung, muss der Microsoft-spezifische Treiber für dieses Gerät durch einen Beckhoff-eigenen ausgetauscht werden. Dazu muss unter 'Netzwerkverbindungen' (z.B. über die Systemsteuerung) der Dialog 'Eigenschaften' der Netzwerkverbindung geöffnet werden. Falls im System mehr als ein Netzwerkadapter installiert ist, ist der Adapter auszuwählen, welcher für Echtzeit-Ethernet vorgesehen ist.

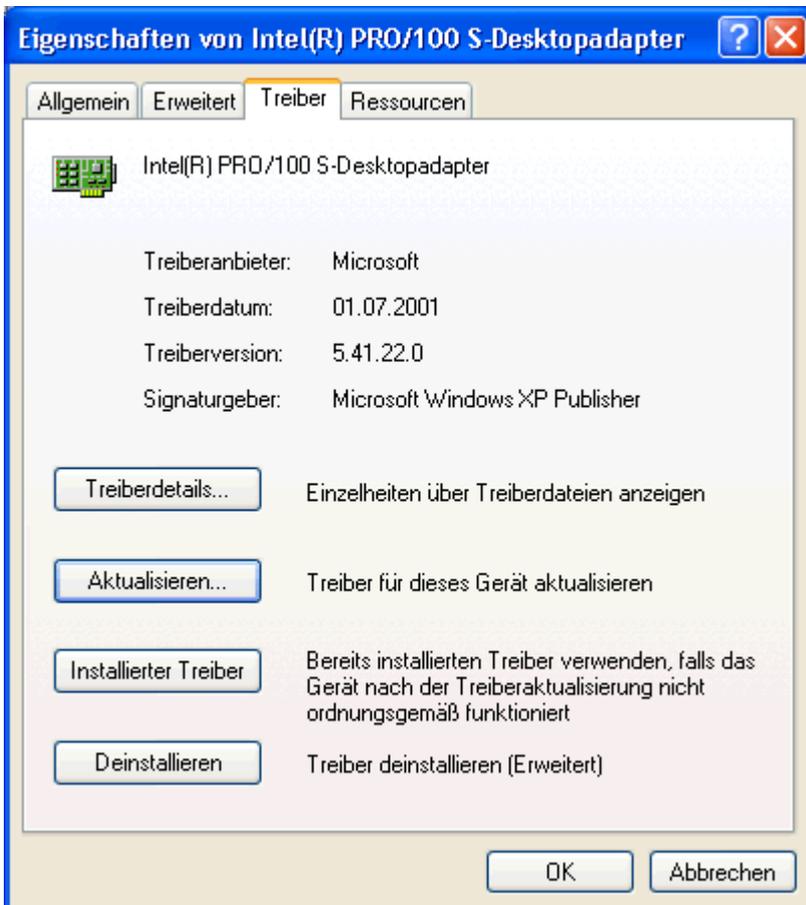
Folgende Dialoge zeigen die notwendigen Schritte zur Installation unter Windows XP Professional.



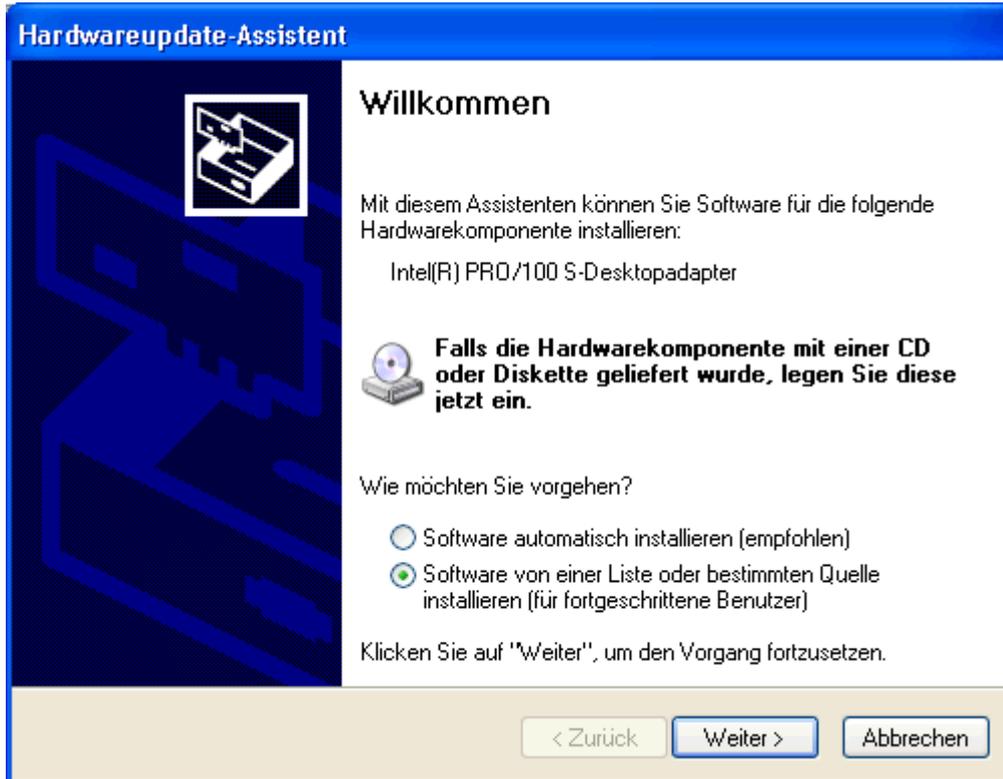
Der Dialog 'Eigenschaften' erscheint. Bitte betätigen Sie hier die Schaltfläche 'Konfigurieren...!'.



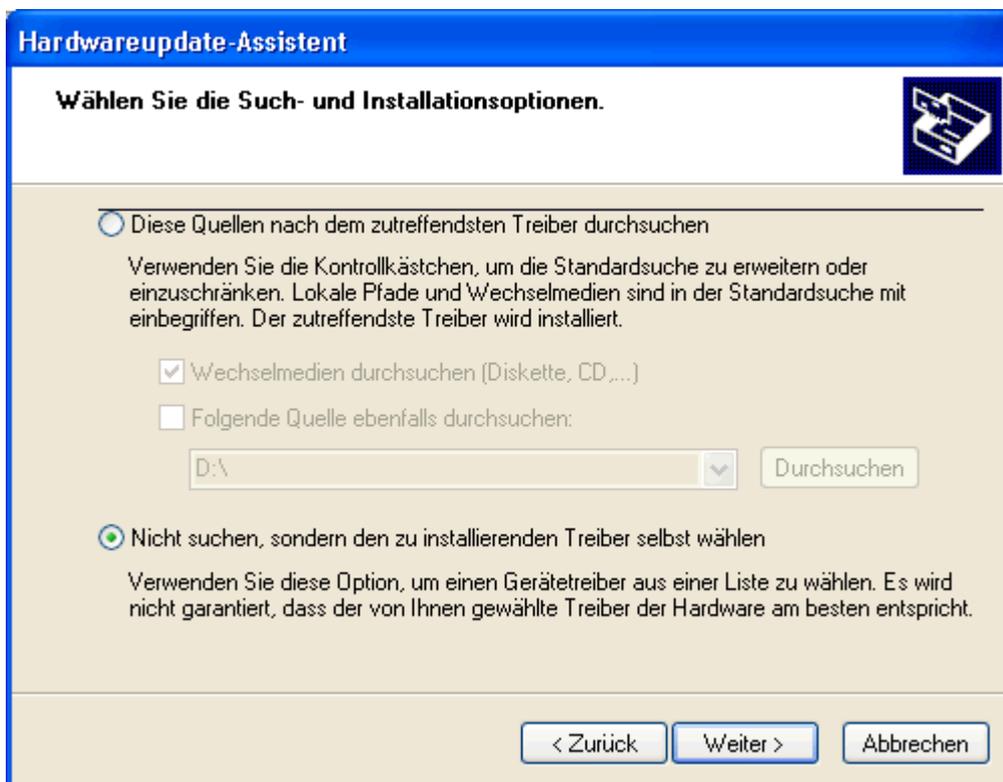
Bitte wählen Sie den Karteireiter 'Treiber' aus, und betätigen Sie dort die 'Aktualisieren...' Schaltfläche.



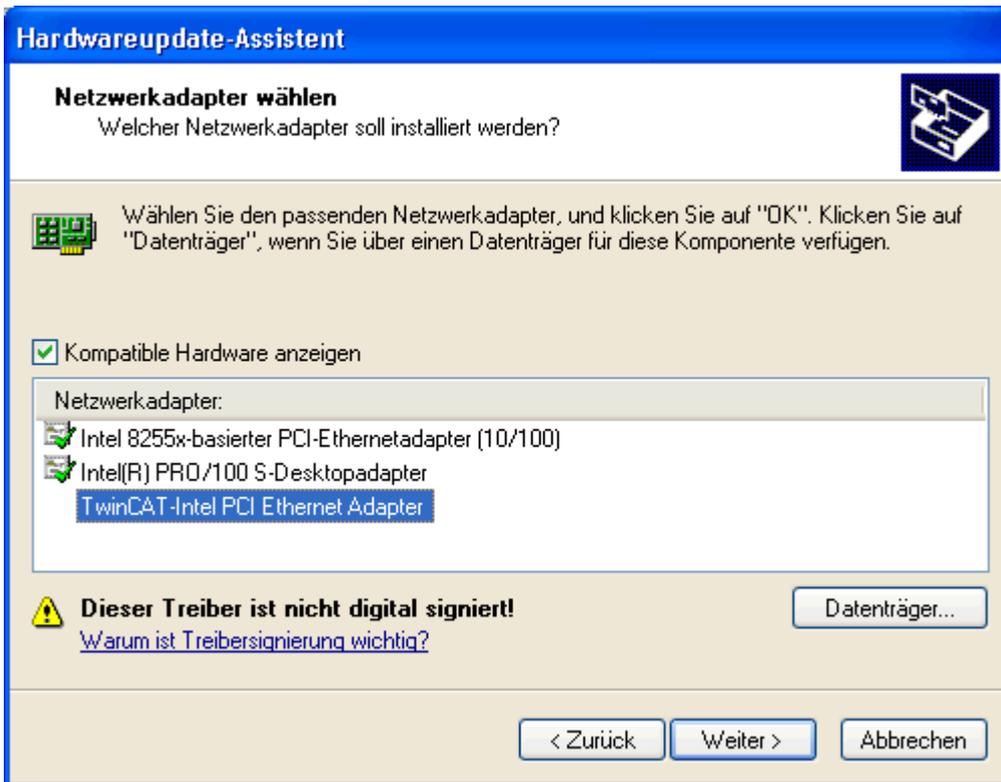
Der 'Hardwareupdate-Assistent' erscheint. Hier bitte 'Software von einer Liste oder bestimmte Quelle installieren' selektieren und auf 'Weiter >' klicken...



Bitte 'Nicht suchen, sondern den zu installierenden Treiber selbst wählen' selektieren und auf 'Weiter >' klicken.



Wählen sie den 'TwinCAT-Intel PCI Ethernet Adapter' und klicken auf 'Weiter >'.



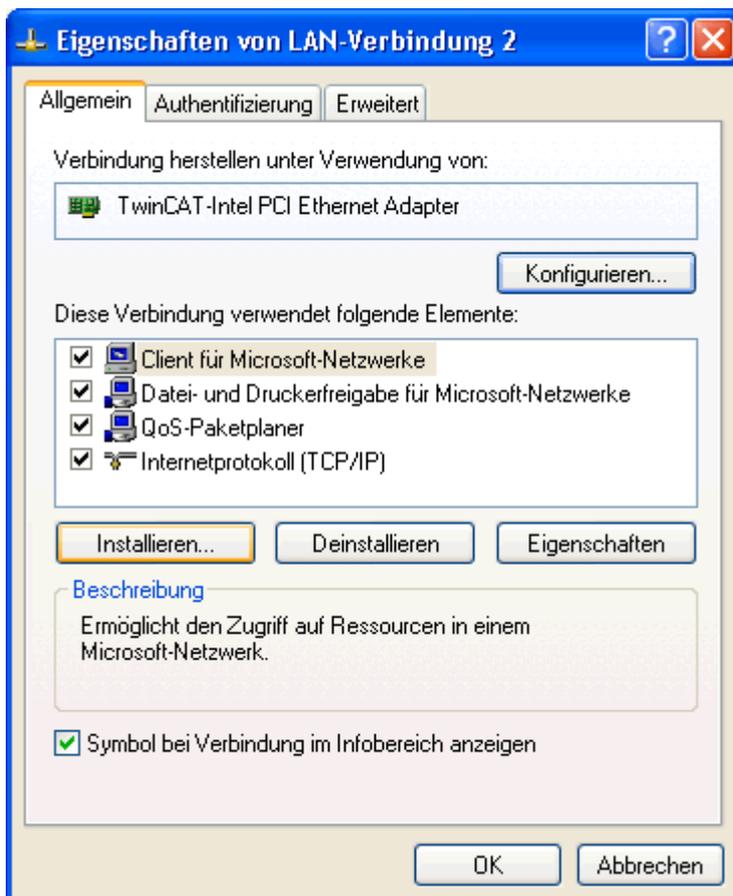
Nun erscheint eine Warnung, dass der ausgewählte Treiber nicht von Microsoft "digital signiert" ist. Quittieren sie die Abfrage mit 'Installation fortsetzen'.



Die Treiberinstallation ist nun erledigt, nur der Protokolltreiber muss noch installiert werden. Bitte die Schaltfläche 'Fertig stellen' betätigen.



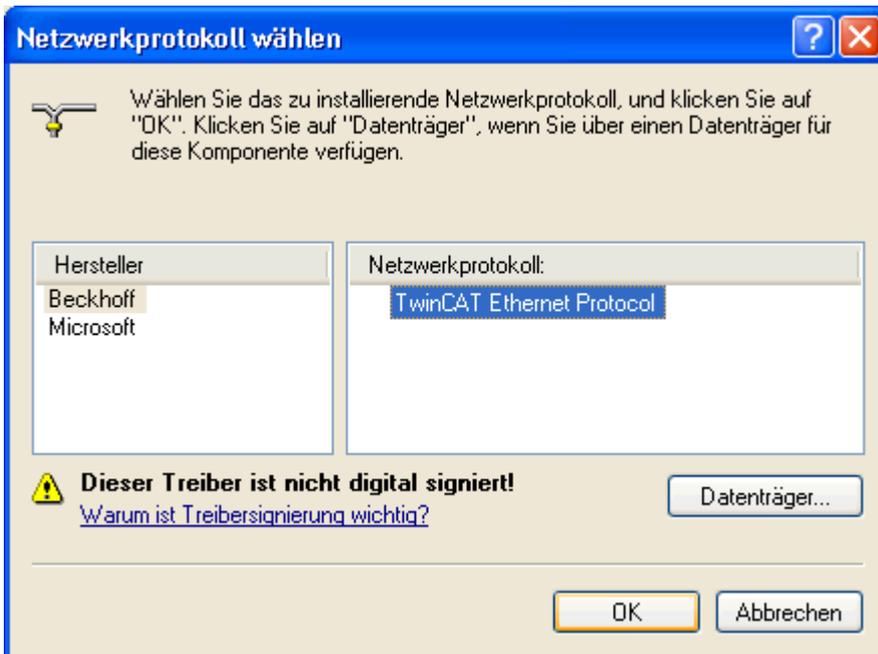
Bitte öffnen sie erneut den Dialog 'Eigenschaften' der Echtzeit-Ethernet Netzwerkverbindung. Der neue Netzwerkadapter sollte nun hier zur Auswahl angeboten werden, wie im nachfolgenden Dialog gezeigt.



Bitte im oberen Dialog die Schaltfläche 'Installieren...' betätigen und im dann erscheinenden und nachfolgend gezeigten Dialog 'Protokoll' auswählen und 'Hinzufügen...' betätigen.

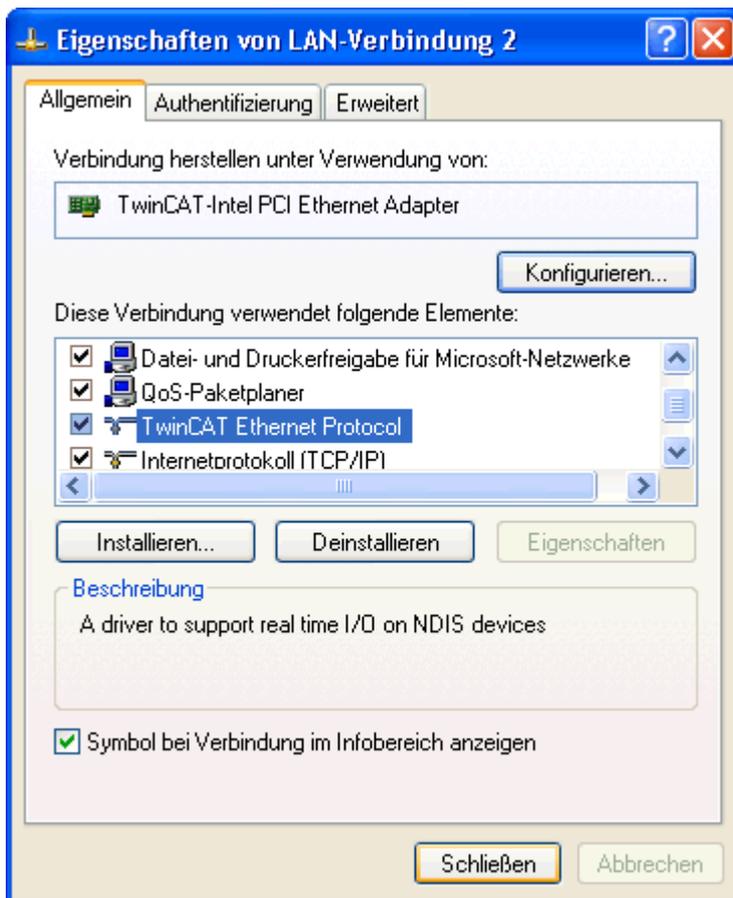


Hier bitte 'Beckhoff' und 'TwinCAT Ethernet Protocol' auswählen. Erscheint es nicht in der Liste, betätigen sie die Schaltfläche "Datenträger..." und wechseln in das \Windows\Inf (Windows XP) bzw. \WinNt\Inf (Windows 2000) Verzeichnis. Sollte das Verzeichnis nicht sichtbar sein, muss im Explorer unter Extras/Ordneroptionen/Ansicht "Alle Dateien und Ordner anzeigen" angewählt werden. Nach Auswahl des Verzeichnisses (egal welche Datei darin selektiert ist) sollte das Beckhoff / TwinCAT Ethernet Protocol nun aufgeführt sein.



Das neue Protokoll sollte nun in der Liste der verwendeten Elemente erscheinen.

Der "Quality of Service" (QoS) Paket Abwicklerfilter kann deaktiviert werden, denn er hat keine Funktion in Verbindung mit dem TwinCAT Treiber.



Nun sind alle erforderlichen Schritte durchgeführt worden. Nur das "TwinCAT Ethernet Protocol" ist beim ersten Mal nicht automatisch gestartet. Man hat hier nun die Möglichkeit, den PC zu rebooten oder das Protokoll manuell zu starten (nur notwendig beim allerersten Mal). Um das Protokoll manuell zu starten (z.B. per 'Ausführen...' von der Kommandozeile nach der Eingabe von 'cmd'), ist im Erscheinenden Bildschirm "net start TcEther" wie unten gezeigt einzugeben und mit 'Enter' zu übernehmen.



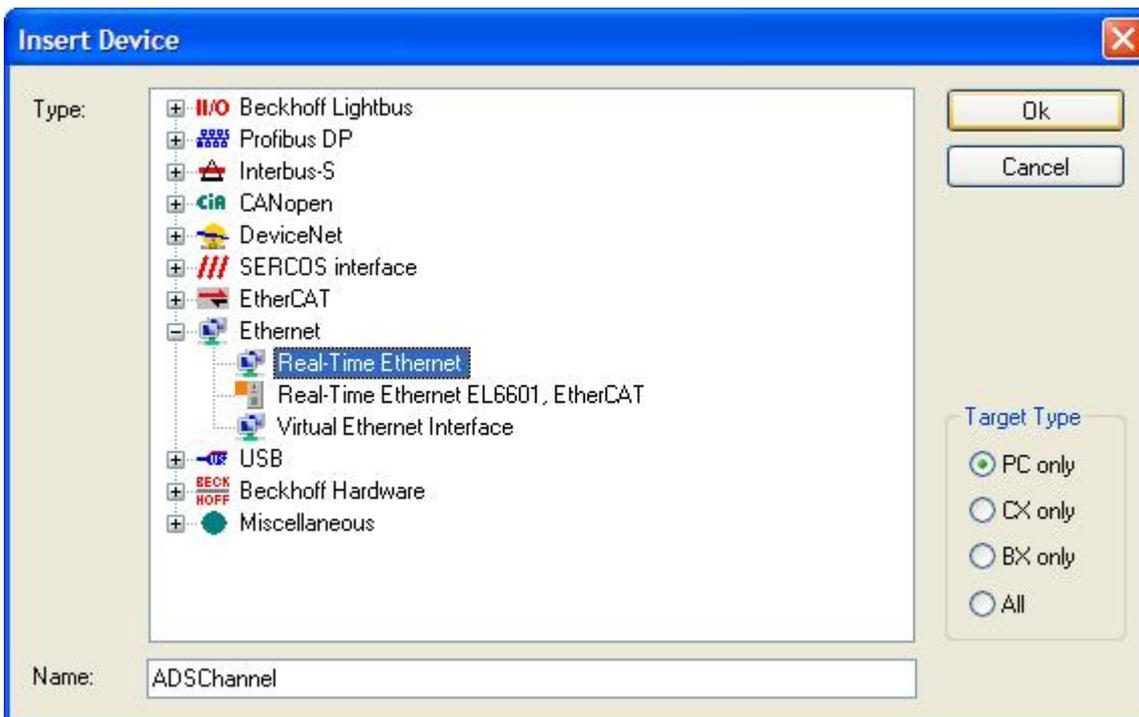
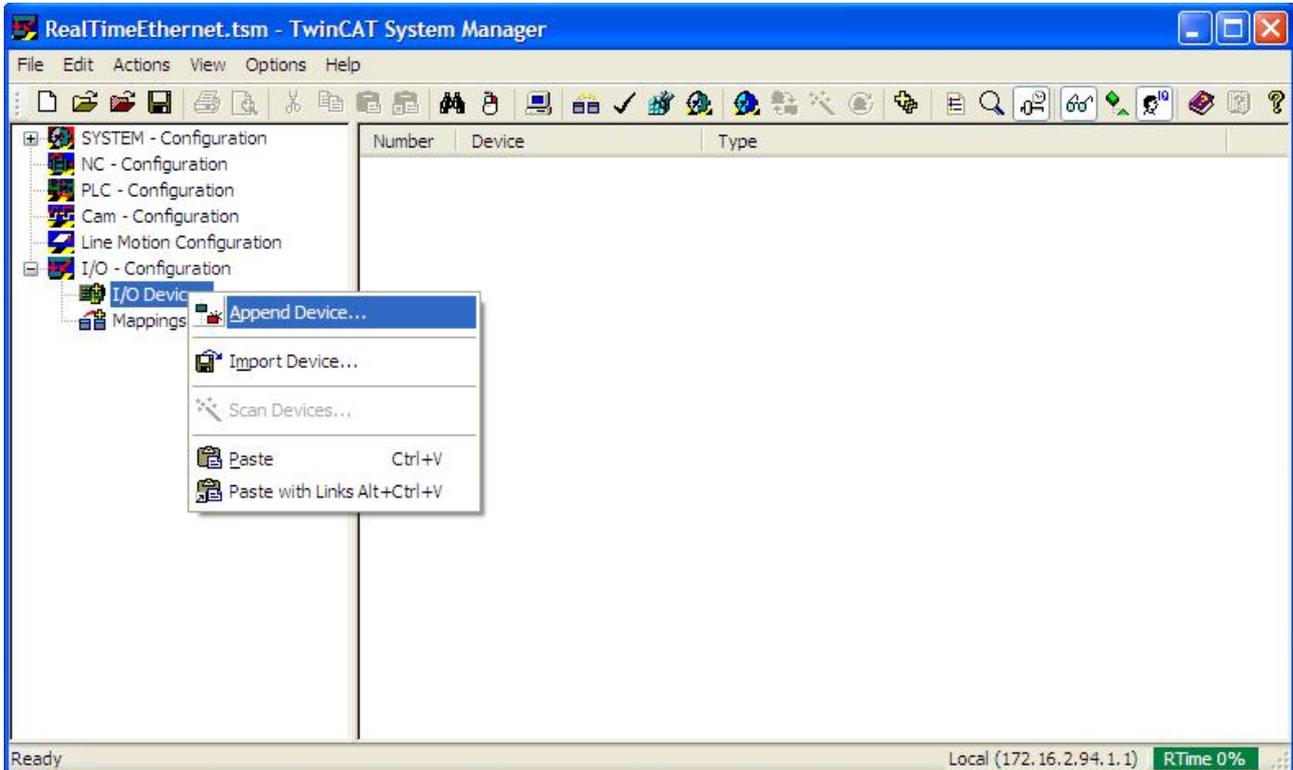
Nach diesen Schritten sollte der TwinCAT System Manager den Echtzeit-Ethernet Adapter als "E/A - Gerät" finden.

# 5 Konfiguration

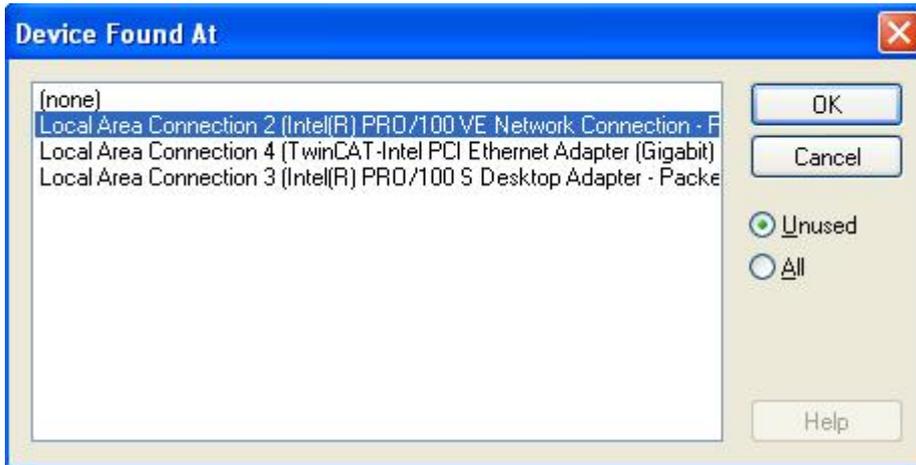
Konfiguration am Beispiel einer ADS Server/Client Kommunikation.

## Server

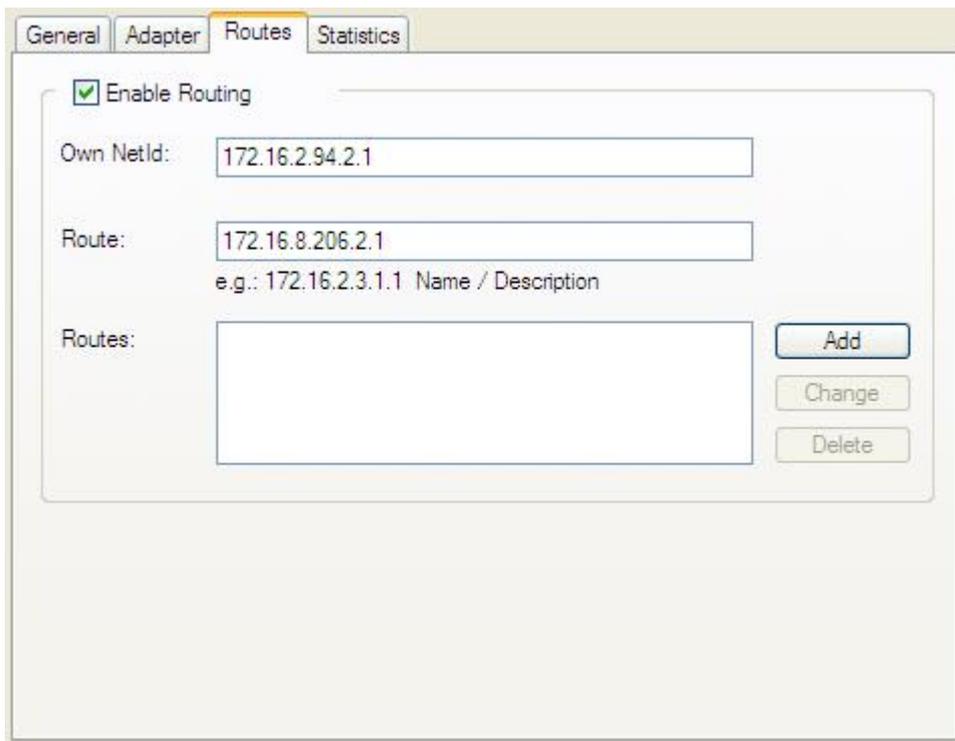
Wählen Sie im TwinCAT System Manager das lokale Zielsystem für den Server aus. Unter "I/O-Configuration" können Sie mit einem rechten Mausklick auf "I/O Devices" ein neues Gerät einfügen. In dem sich öffnenden Fenster haben Sie nun die Möglichkeit unter dem Punkt Ethernet das "Real-Time Ethernet" Device auszuwählen.



Bevor Sie Ihre Auswahl durch den Ok-Button bestätigen, können Sie dem Gerät unten im Fenster noch einen neuen Namen zuweisen. Nach der Bestätigung müssen Sie die entsprechende Netzwerkkarte auswählen, wie es in der nächsten Abbildung dargestellt ist.



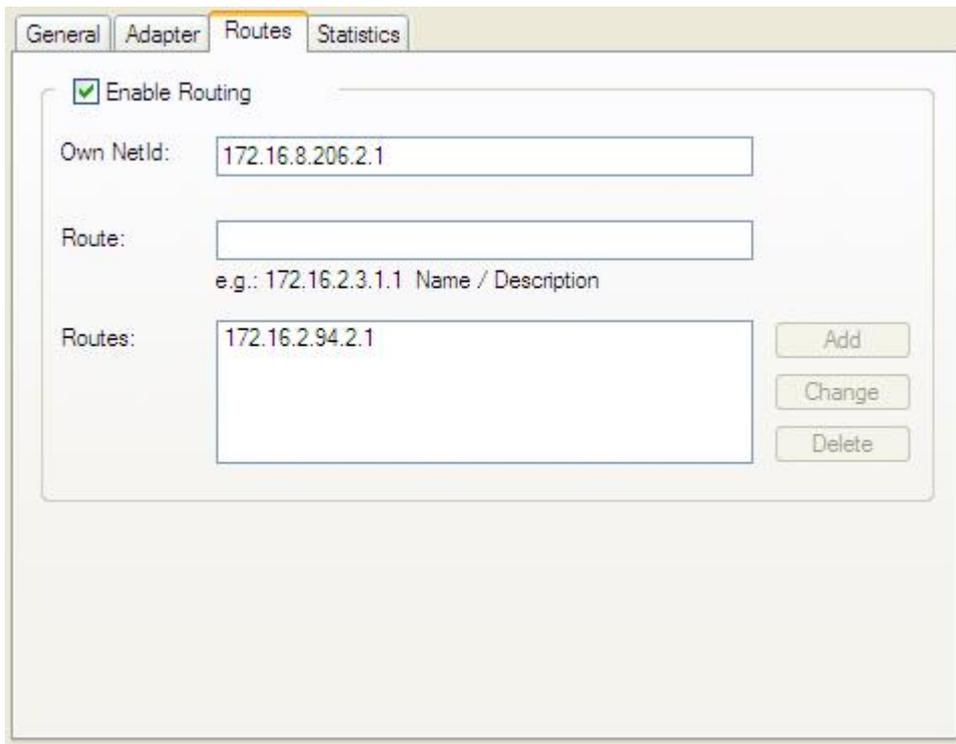
Im Karteireiter "Routes" des neuen Geräts müssen Sie zunächst das Kontrollkästchen "Enable Routing" aktivieren. Durch die Aktivierung wird automatisch die NetID des eingefügten RT-Ethernet Gerätes im entsprechenden Feld eingetragen.



Im nächsten Schritt können Sie die NetID eines Clients unter Route eingetragen und mit der Bestätigung durch "Add" zu möglichen weiteren NetIDs, mit denen der Server dann kommunizieren kann, hinzufügen.

#### Client:

Bei den Clients sind die gleichen Einstellungen wie beim Server vorzunehmen, mit dem Unterschied, dass Sie unter "Routes" des Real-Time Ethernet Gerätes die NetID des Servers angeben müssen.



## 5.1 Beispiel

Nach der zuvor beschriebenen Konfiguration, können Sie nun einen Test mit dem folgenden SPS-Projekt durchführen. Für dieses Beispiel holt sich der Server per ADS ein Handle für eine Variable des Clients und beschreibt diese mit dem Wert 500. Im Zusammenhang mit ADS kann hier eine sehr sinnvolle Anwendung gezeigt werden.

### 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

```



- Auf der Client-Seite reicht das Anlegen einer UDINT Variablen.
- Wenn Sie den Quellcode für einen Test benutzen wollen, müssen die NetIDs auf Ihr System entsprechend angepasst werden.

## 6 Beispiel

Nach der zuvor beschriebenen Konfiguration, können Sie nun einen Test mit dem folgenden SPS-Projekt durchführen. Für dieses Beispiel holt sich der Server per ADS ein Handle für eine Variable des Clients und beschreibt diese mit dem Wert 500. Im Zusammenhang mit ADS kann hier eine sehr sinnvolle Anwendung gezeigt werden.

### 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
  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
```



- Auf der Client-Seite reicht das Anlegen einer UDINT Variablen.
  - Wenn Sie den Quellcode für einen Test benutzen wollen, müssen die NetIDs auf Ihr System entsprechend angepasst werden.
-



Mehr Informationen:  
**[www.beckhoff.de/automation](http://www.beckhoff.de/automation)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.de](mailto:info@beckhoff.de)  
[www.beckhoff.de](http://www.beckhoff.de)

