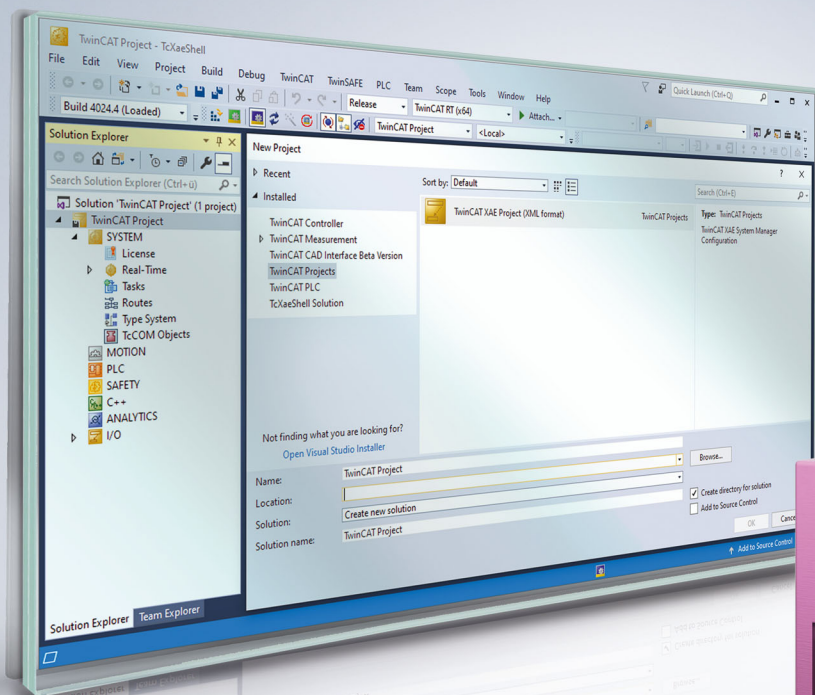


BECKHOFF New Automation Technology

Handbuch | DE

TF6280

TwinCAT 3 | EtherNet/IP Adapter



Inhaltsverzeichnis

1 Vorwort.....	5
1.1 Hinweise zur Dokumentation	5
1.2 Zu Ihrer Sicherheit.....	6
1.3 Hinweise zur Informationssicherheit	7
2 Übersicht.....	8
3 Voraussetzungen	9
3.1 Hardware-Voraussetzungen prüfen	9
4 Lizenzierung	12
5 Konfiguration.....	15
5.1 Anlegen eines EtherNet/IP Adapters	15
5.1.1 Firewall Einstellung	18
5.1.2 IP Routing	18
5.2 Einstellen der Zykluszeit	18
5.3 EtherNet/IP-Einstellungen verändern.....	19
5.3.1 Objektbeschreibung	21
5.3.2 ADS-Write Kommando.....	21
5.3.3 ADS-Read Kommando.....	22
5.3.4 Beispiel.....	23
5.4 Anlegen des EtherNet/IP Adapters in anderen EtherNet/IP Scannern	23
5.4.1 Beispiel für Rockwell CPUs.....	25
5.5 Azyklische Kommunikation via Explicit Messaging	26
5.5.1 Common Industrial Protocol (CIP)	26
5.5.2 Forward Message to AMS Port via Explicit Messaging.....	27
6 Eigenschaften.....	35
6.1 Virtueller Adapter	35
6.2 TF6280 - Konfigurations-Parameter.....	36
6.2.1 Index 0x8000 Adapter Settings	36
6.2.2 Index 0x8001 IO Assembly Settings	37
6.2.3 Index 0x9000 Adapter Info	37
6.2.4 Index 0x9001 IO Assembly Info	37
7 Diagnose History.....	38
7.1 Error Codes TF6280	39
8 Anhang.....	41
8.1 Support und Service.....	41

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, stets die aktuell gültige 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 weiterentwickelt.

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® 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 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.

Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

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.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

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 der Function TF6280 TwinCAT EtherNet/IP Adapter können Sie in Kombination mit einem netzwerkfähigen Beckhoff-PC mit einer physikalischen Schnittstelle bis zu 8 EtherNet/IP Adapter parametrieren. Hierbei wird eine virtuelle MAC-Adresse gebildet, über die Sie bis zu 8 EtherNet/IP Adapter an einem PC über eine Ethernet-Schnittstelle betreiben können.

Technische Daten	TF6280							
Benötigt	TC1200 ab Build 4020							
Zielsystem	Windows XP, Windows 7/8, Windows CE							
Leistungsklasse (pp)	20	30	40	50	60	70	80	90
	–	–	X	X	X	X	X	X

Bestellangaben	
TF6280-00pp	TC3 EtherNet/IP Adapter

Die Function TF6280 TwinCAT EtherNet/IP Adapter ermöglicht den Datenaustausch mit einem EtherNet/IP-Scanner, es wird sowohl Multicast als auch Unicast unterstützt. Die Function TF6280 TwinCAT EtherNet/IP Adapter kann sich außerdem wie acht EtherNet/IP-Adapter verhalten.

Z. B. ist es möglich:

- einen Scanner mit acht Adaptern zu verbinden
- bis zu acht Scanner mit 8 Adaptern zu verbinden

Auf diese Weise können mehr Daten transportiert oder die Scanner mit unterschiedlichen Zykluszeiten betrieben werden.

In einem EtherNet/IP-Netzwerk verhält sich die TF6280 als Adapter Teilnehmer und es ist keine weitere Konfiguration über einen EtherNet/IP-Scanner notwendig. Die Konfiguration erfolgt über den Konfigurator in TwinCAT 3.1, indem Sie z.B. die IP-Einstellungen sowie die Anzahl der Daten festlegen. Lediglich die Daten selbst müssen im EtherNet/IP Scanner genauso eingestellt werden, damit eine Verbindung aufgebaut werden kann.

EtherNet/IP

EtherNet/IP (Ethernet Industrial Protocol, EIP) ist ein Echtzeit Ethernet Protokoll was durch die ODVA (Open DeviceNet Vendor Association) offengelegt und standardisiert wurde. Dieses Protokoll basiert auf TCP, UDP und IPv4.

Weitere Informationen finden Sie auf www.odva.org oder <https://en.wikipedia.org/wiki/EtherNet/IP>.

3 Voraussetzungen

Software

Die TF6280 ist in der **TwinCAT**-Version **3.1** Build **4020.28** enthalten. Es ist keine weitere Installation notwendig.

● Ältere Produktversionen

i Ältere Versionen sind Beta-Versionen. Löschen Sie die älteren EtherNet/IP-Geräte-Konfigurationen und erstellen Sie eine neue Konfiguration.

Hardware

Für die Nutzung der TF6280 ist es notwendig, dass auf dem Zielsystem ein Intel®-Netzwerkchipsatz vorhanden ist (siehe: [Hardware-Voraussetzungen prüfen](#) [► 9]).

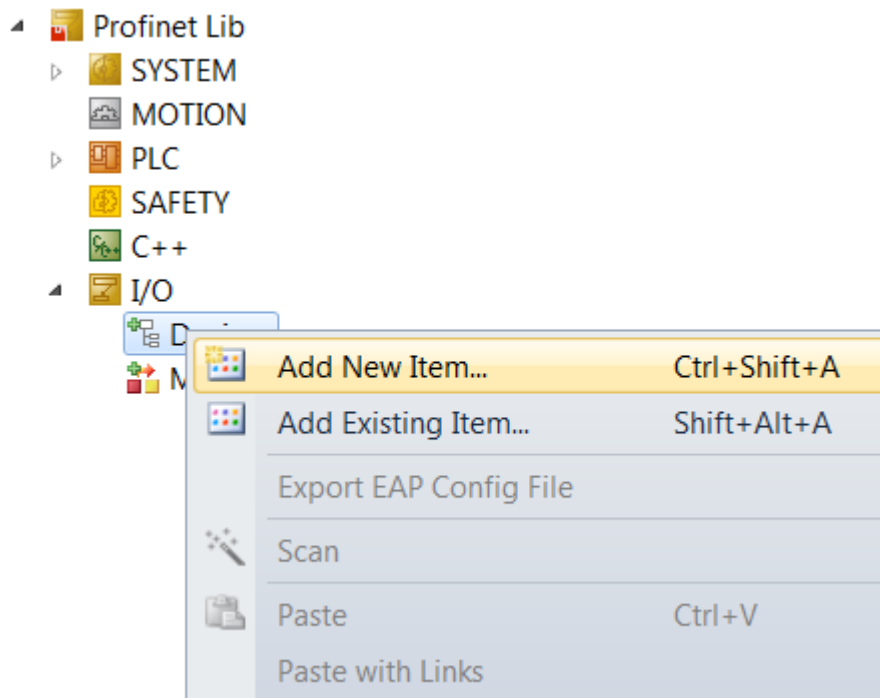
● Beckhoff-PC

i Beckhoff PC-Systeme sind in der Regel für den Betrieb von EtherNet/IP-Geräten vorkonfiguriert.

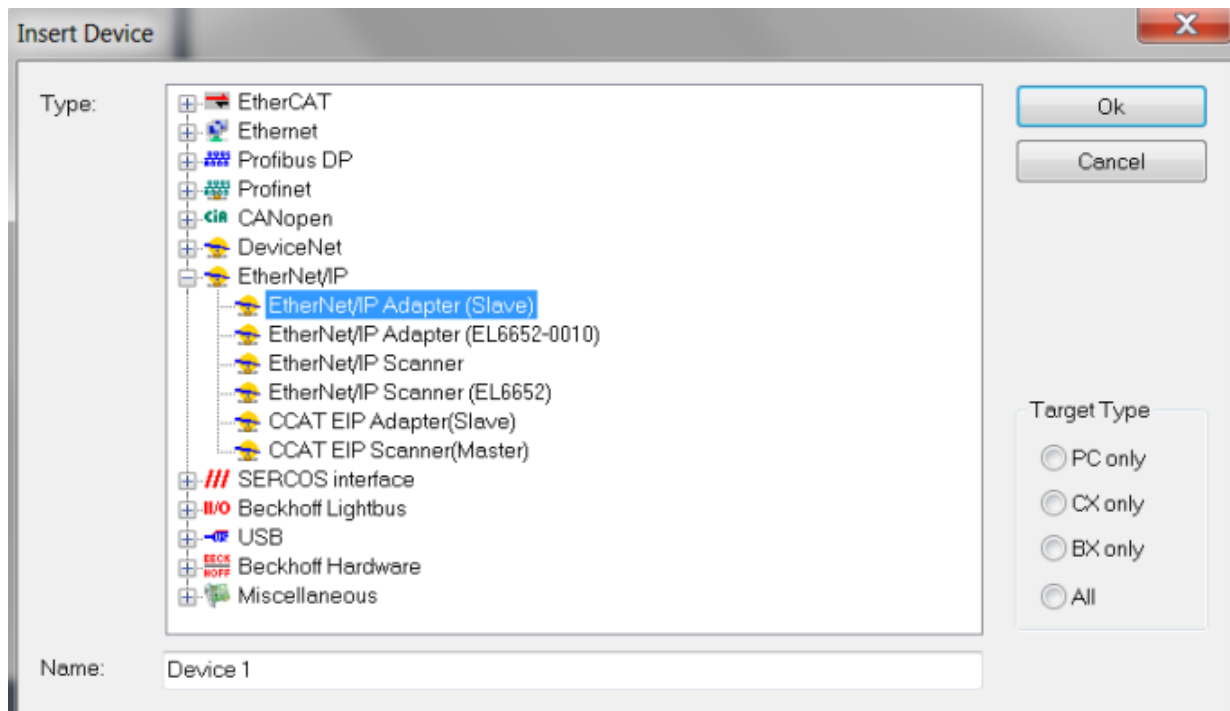
3.1 Hardware-Voraussetzungen prüfen

Prüfen, ob die Netzwerkschnittstelle geeignet ist

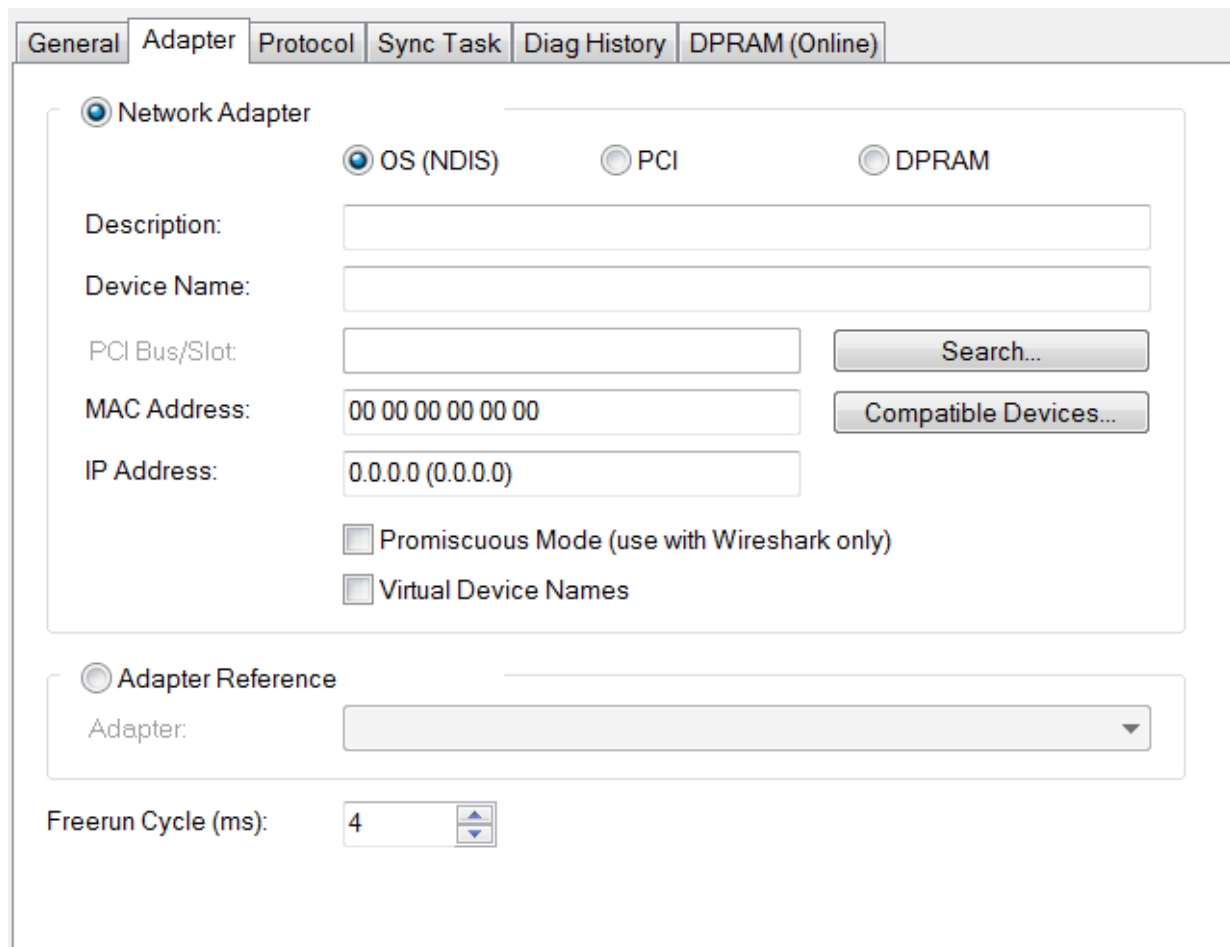
1. Legen Sie einen EtherNet/IP Adapter an. Klicken Sie dazu mit der rechten Maustaste auf **Devices** und fügen Sie mit **Add New Item...** ein neues Gerät an.



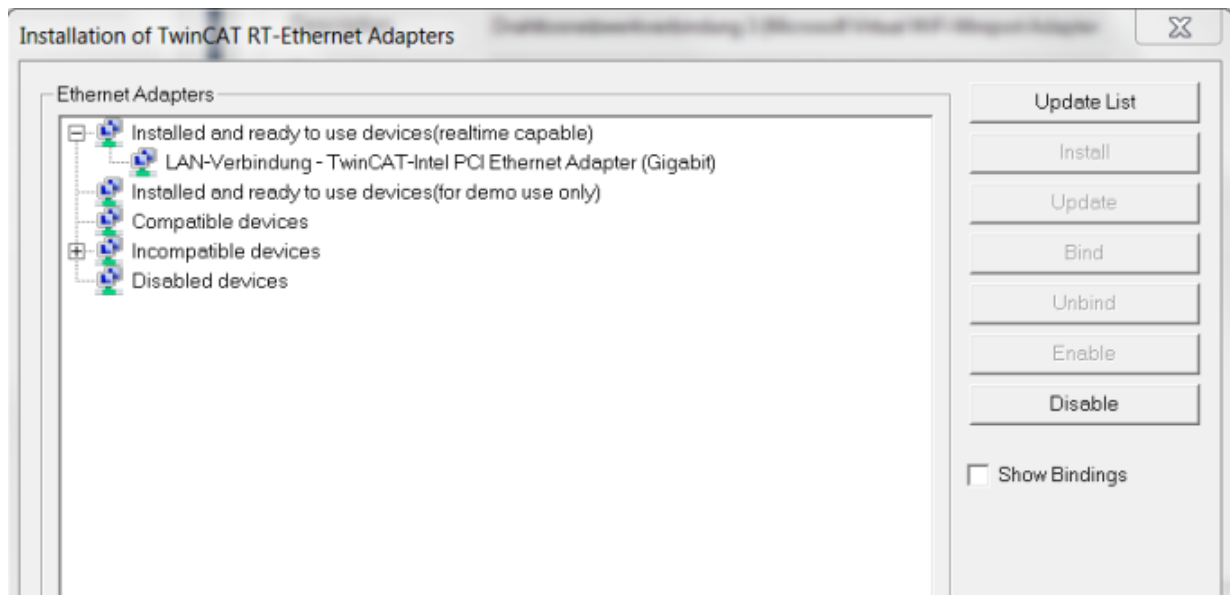
2. Wählen Sie **EtherNet/IP Adapter (Slave)** aus.



3. Wählen Sie nun den Adapter aus und suchen (**Search...**) Sie die passende Ethernet Schnittstelle aus.



4. Wählen Sie eine „realtime capable“-Schnittstelle unter **Compatible devices** aus.



⇒ Sie können den Echtzeittreiber installieren.

i Keine realtime capable-Netzwerkschnittstelle vorhanden

Wenn die Liste keine **Compatible devices**-Netzwerkschnittstellen enthält, ist die Nutzung der TF6280 Function auf der genutzten Hardware ist nicht möglich.

4 Lizenzierung

Die TwinCAT 3 Function ist als Vollversion oder als 7-Tage-Testversion freischaltbar. Beide Lizenztypen sind über die TwinCAT-3-Entwicklungsumgebung (XAE) aktivierbar.

Lizenzierung der Vollversion einer TwinCAT 3 Function

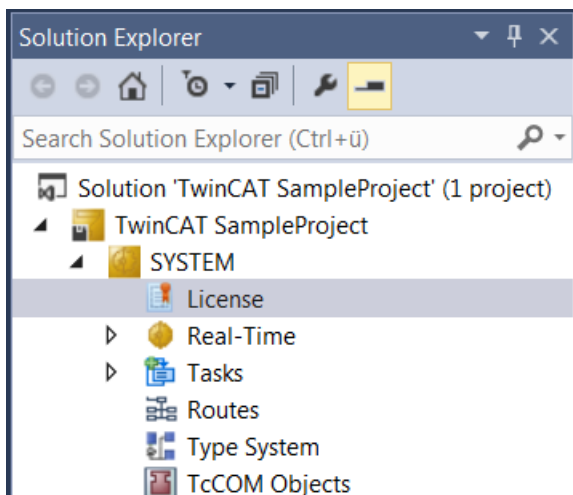
Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT-3-Lizenzierung](#)“.

Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function



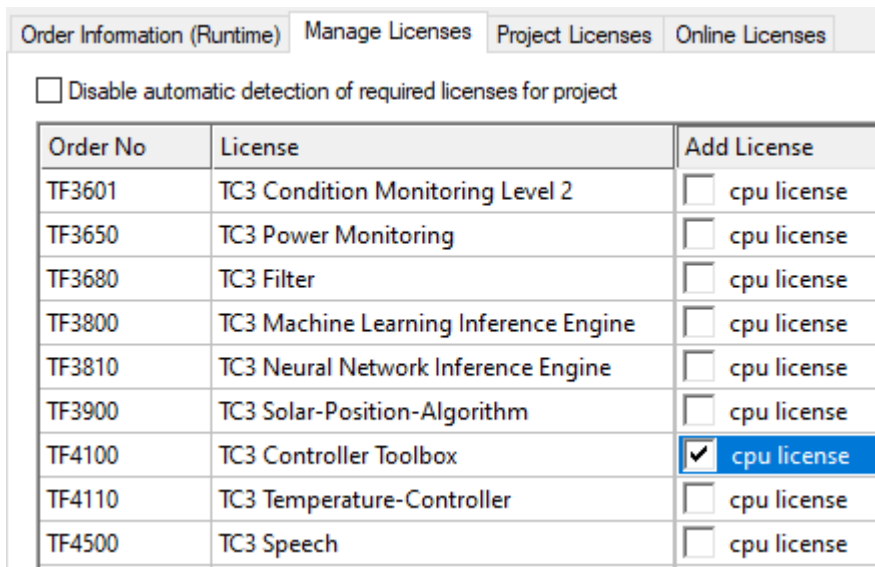
Eine 7-Tage-Testversion kann nicht für einen [TwinCAT-3-Lizenz-Dongle](#) freigeschaltet werden.

1. Starten Sie die TwinCAT-3-Entwicklungsumgebung (XAE).
2. Öffnen Sie ein bestehendes TwinCAT-3-Projekt oder legen Sie ein neues Projekt an.
3. Wenn Sie die Lizenz für ein Remote-Gerät aktivieren wollen, stellen Sie das gewünschte Zielsystem ein. Wählen Sie dazu in der Symbolleiste in der Drop-down-Liste **Choose Target System** das Zielsystem aus.
 - ⇒ Die Lizenzierungseinstellungen beziehen sich immer auf das eingestellte Zielsystem. Mit der Aktivierung des Projekts auf dem Zielsystem werden automatisch auch die zugehörigen TwinCAT-3-Lizenzen auf dieses System kopiert.
4. Klicken Sie im **Solution Explorer** im Teilbaum **SYSTEM** doppelt auf **License**.



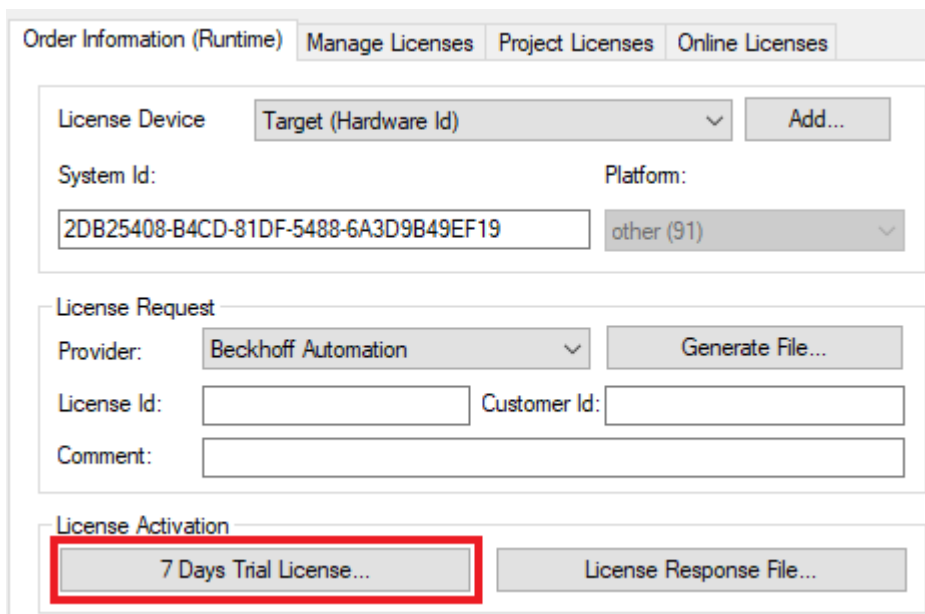
⇒ Der TwinCAT-3-Lizenzmanager öffnet sich.

- Öffnen Sie die Registerkarte **Manage Licenses**. Aktivieren Sie in der Spalte **Add License** das Auswahlkästchen für die Lizenz, die Sie Ihrem Projekt hinzufügen möchten (z. B. „TF4100 TC3 Controller Toolbox“).



Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	☐ cpu license
TF3650	TC3 Power Monitoring	☐ cpu license
TF3680	TC3 Filter	☐ cpu license
TF3800	TC3 Machine Learning Inference Engine	☐ cpu license
TF3810	TC3 Neural Network Inference Engine	☐ cpu license
TF3900	TC3 Solar-Position-Algorithm	☐ cpu license
TF4100	TC3 Controller Toolbox	☒ cpu license
TF4110	TC3 Temperature-Controller	☐ cpu license
TF4500	TC3 Speech	☐ cpu license

- Öffnen Sie die Registerkarte **Order Information (Runtime)**.
⇒ In der tabellarischen Übersicht der Lizenzen wird die zuvor ausgewählte Lizenz mit dem Status „missing“ angezeigt.
- Klicken Sie auf **7 Days Trial License...**, um die 7-Tage-Testlizenz zu aktivieren.



Order Information (Runtime) Manage Licenses Project Licenses Online Licenses

License Device: Target (Hardware Id) Add...

System Id: 2DB25408-B4CD-81DF-5488-6A3D9B49EF19 Platform: other (91)

License Request

Provider: Beckhoff Automation Generate File...

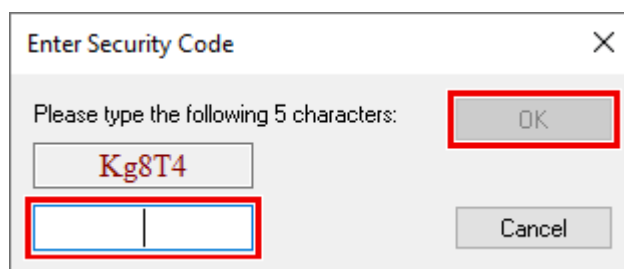
License Id: Customer Id:

Comment:

License Activation

7 Days Trial License... License Response File...

- ⇒ Es öffnet sich ein Dialog, der Sie auffordert, den im Dialog angezeigten Sicherheitscode einzugeben.



Enter Security Code

Please type the following 5 characters: Kg8T4

OK Cancel

- Geben Sie den Code genauso ein, wie er angezeigt wird, und bestätigen Sie ihn.
- Bestätigen Sie den nachfolgenden Dialog, der Sie auf die erfolgreiche Aktivierung hinweist.

- ⇒ In der tabellarischen Übersicht der Lizenzen gibt der Lizenzstatus nun das Ablaufdatum der Lizenz an.

10. Starten Sie das TwinCAT-System neu.

- ⇒ Die 7-Tage-Testversion ist freigeschaltet.

5 Konfiguration

Die wichtigsten Einstellungen, um eine Verbindung mit einem EtherNet/IP-Scanner herzustellen sind:

- die IP-Adresse
- die Assembly-Instance-Nummern und damit die Länge der Daten
- und die richtige Zykluszeit.

IP-Adresse:

Die IP-Adresse kann frei vergeben werden, sollte allerdings in der gleichen Netzwerkklassse wie der Scanner liegen, sonst muss ein Gateway eingetragen sein, um das Protokoll entsprechend zu routen.

Assembly-Instance-Nummern:

Die Assembly-Instance-Nummern sind fest vorgegeben und müssen im Scanner richtig eingestellt werden. Dazu gehört auch immer die Anzahl der Daten bzw. die Größe des Prozessabbildes.

Zykluszeit:

Die Task-Zykluszeit darf in der TF6280 höchstens genau so groß sein, wie auf der Scanner-Seite, sie kann aber auch einen Bruchteil davon betragen. Auf der Scanner-Seite kann beispielsweise eine Zykluszeit beim EtherNet/IP von 10 ms eingestellt werden und auf der Adapter-Seite kann dann eine Task-Zykluszeit von 10 ms, 5 ms, 2 ms oder 1 ms verwendet werden.



Empfohlene Zykluszeit

Bei EtherNet/IP sind Zykluszeiten von 1 ms oder höher möglich. Sie können daher die Task immer mit 1 ms betreiben, solange die Systemauslastung Ihres [Systems \[► 8\]](#) dies zulässt.

5.1 Anlegen eines EtherNet/IP Adapters

Sobald Sie einen EtherNet/IP-Adapter angefügt haben, wird Ihrer Konfiguration automatisch ein Adapter hinzugefügt.

1. Stellen Sie die IP-Adresse des Adapters ein. (Die IP-Adresse muss nicht gleich der IP-Adresse des Betriebssystems sein.) Klicken Sie auf die Box und wechseln Sie zum Karteireiter **Settings**. Hier stellen Sie die **IP-Adresse**, die **Netzwerkmaske** und die **Gateway-Adresse** ein.

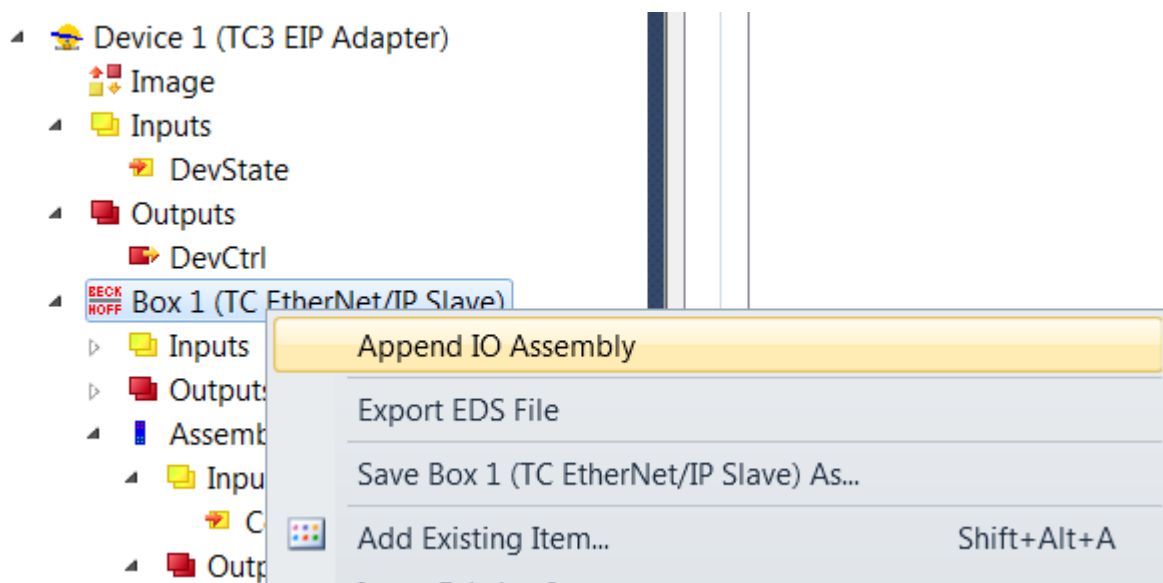
General Settings					
Slave Settings					
Index	Name	Flags	Value	Unit	
8000:0	Slave Settings (Box 1)	M RO	> 43 <		
8000:01	Slave Number	M RO	0x0001 (1)		
8000:03	Product Name	M RW	Box 1 (TC EtherNet/IP Slav...		
8000:04	Device Type	M RO	0x000C (12)		
8000:05	Vendor ID	M RO	0x006C (108)		
8000:06	Product Code	M RO	0x1888 (6280)		
8000:07	Revision	M RO	3.1		
8000:08	Serial Number	M RO	0x00000000 (0)		
8000:20	MAC Address	M RO	EE 00 01 1F 7E 88		
8000:21	IP Address	M RW	0.0.0.0		
8000:22	Network Mask	M RW	0.0.0.0		
8000:23	Gateway Address	M RW	0.0.0.0		
8000:24	DHCP Max Retries	M RW	0		
8000:25	TCP/IP TTL	M RW	128		
8000:26	TCP/IP UDP Checksum	M RW	TRUE		
8000:27	TCP/IP TCP Timeout	M RW	300 Seconds		
8000:28	MultiCast TTL	M RW	1		
8000:29	MultiCast UDP Checksum	M RW	FALSE		

1 a) Wenn die IP-Adresse von einem DHCP-Server in Ihrem Netzwerk vergeben werden soll, tragen Sie in das Feld **IP Address** den Wert 0.0.0.0 ein.

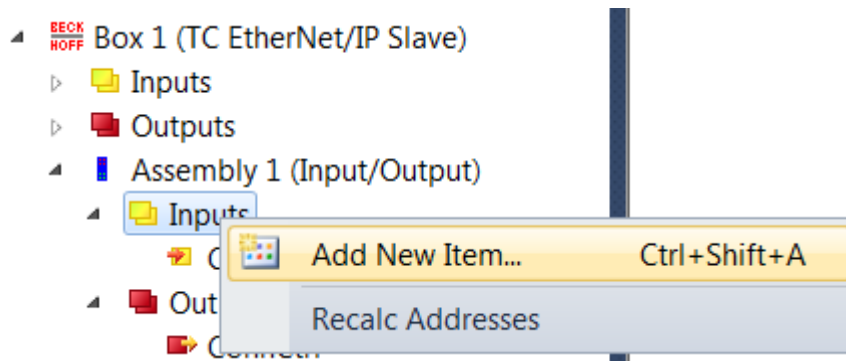
1 b) Wenn die IP-Adresse des Betriebssystems verwendet werden soll, tragen Sie in das Feld **IP Address** den Wert 255.255.255.255 ein. Die Subnetzmaske und die Gateway-Adresse können unverändert übernommen werden. Der EtherNet/IP-Treiber verwendet dann beim Starten von TwinCAT die IP-Adresse des Systems.

Beachten Sie die [Firewall Einstellungen](#) [► 18].

2. Klicken Sie auf die Box und wählen Sie **Append IO Assembly**.



3. Legen Sie unter Inputs Daten an, rechte Maustaste **Add New Item...**



4. Wählen Sie jetzt das Datenformat aus und die Anzahl an Daten, die übertragen werden sollen. Wichtig ist später die Anzahl an Bytes, diese können im Objektbaum ausgelesen werden. Z. B.: fügen Sie 4 Worte ein, also 8 Byte Prozessdaten:

Insert Variable

General

Name: Var 7 Multiple: 4

Start Address: Byte: 6 Bit: 0

☐ Show All

Data Type	>Size	Name Space
ST_AX5000_S_0085	2	AX5000
ST_AX5000_S_0169	2	AX5000
TcEventArgumentType	2	
TcEventConfirmationState	2	
TcEventSeverity	2	
UINT	2	
WORD	2	
BOOL32	4	
BX_KBUS_STATE	4	IO
DATE	4	

Dazu kommen noch 4 Byte für das ConnState. Das ConnState ist aktuell ohne Funktion, kann aber in Zukunft für weitere Infos verwendet werden.

5. Es müssten also 12 Byte Prozessdaten angelegt sein. Gehen Sie dafür auf die Box und wählen Sie den Karteireiter **Settings**.

General Settings				
Slave Settings				
Index	Name	Flags	Value	Unit
8000:0	Slave Settings (Box 1)	M RO	> 43 <	
8001:0	IO Assembly 1 Settings	M RO	> 12 <	
8001:01	Assembly Number	M RO	0x0001 (1)	
8001:02	Configuration Instance	M RO	128	
8001:03	Configuration Size	M RO	0 Byte	
8001:04	Input Instance (T->O)	M RO	129	
8001:05	Input Size (T->O)	M RO	4 Byte	
8001:06	Output Instance (O->T)	M RO	130	
8001:07	Output Size (O->T)	M RO	12 Byte	
8001:08	Heartbeat Instance (Listen Onl...	M RO	136	
8001:09	Heartbeat Size (Listen Only)	M RO	0 Byte	
8001:...	Heartbeat Instance (Input Only)	M RO	137	
8001:...	Heartbeat Size (Input Only)	M RO	0 Byte	
8001:...	Advanced Assembly Options	M RW	0x0000 (0)	
9000:0	Slave Info (Box 1)	RO	> 43 <	
9001:0	IO Assembly 1 Info	RO	> 12 <	

⇒ Im Index-Feld 0x8001:07 finden Sie die Länge. Die Darstellung der Länge erfolgt aus Scanner-Sicht. Was in TwinCAT Eingänge sind, sind im Scanner die Ausgänge, also hier die Output Size.

6. Verfahren Sie ebenso mit den Ausgängen des EtherNet/IP-Adapters.

⇒ Damit ist das Anlegen der Daten beendet. Verlinken Sie die Daten nun mit der SPS.

5.1.1 Firewall Einstellung

Falls die EtherNet/IP Adresse gleich der IP Adresse des Betriebssystems (OS) sein soll, muss die Firewall eingeschaltet sein. Weicht die IP Adresse des EtherNet/IP Scanners von der IP Einstellung des Betriebssystems ab, wird empfohlen, die Firewall einzuschalten.

5.1.2 IP Routing

Wird das IP Routing verwendet, dann muss sich die IP Adresse des OS in einem anderen SubNetz als die IP Adresse vom Ethernet/IPAdapter/Scanner befinden.

Der Regkey kann abhängig vom Betriebssystem und der Version unterschiedlich sein, hier nur als Beispiel, Default ist „0“.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters "IPEnableRouter"

5.2 Einstellen der Zykluszeit

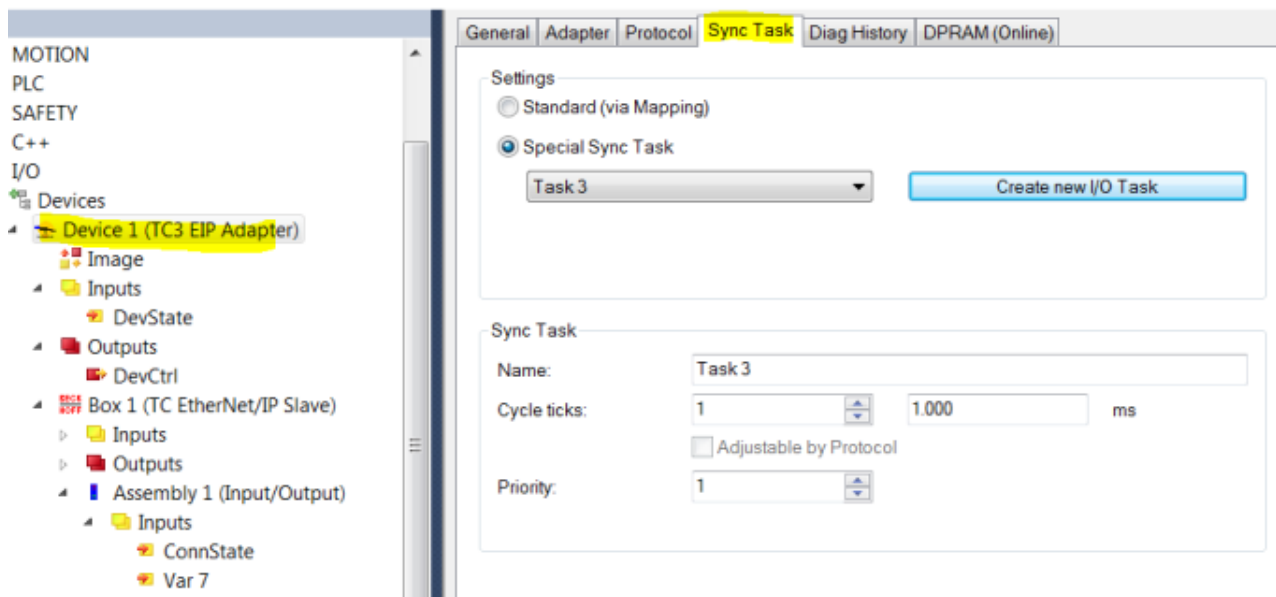
Die Zykluszeit des EtherNet/IP Adapters wird vom Scanner vorgegeben, allerdings muss die Task auf dem TwinCAT System mindestens in der gleichen Geschwindigkeit arbeiten.



Empfohlene Zykluszeit

Bei EtherNet/IP sind Zykluszeiten von 1 ms oder höher möglich. Sie können daher die Task immer mit 1 ms betreiben, solange die Systemauslastung Ihres Systems [► 8] dies zulässt.

Zum Einstellen der Task-Zykluszeit gehen Sie auf **Device EIP Adapter**. Im Karteireiter **Sync Task** stellen Sie die Zeit ein.



i Dedizierte Sync-Task verwenden

Verwenden Sie eine eigene Sync Task, da ein Mapping über die SPS, bei einem Breakpoint z. B. dazu führt, dass die Task angehalten wird und damit die EtherNet/IP Verbindung unterbrochen wird.

5.3 EtherNet/IP-Einstellungen verändern

Für die Einstellung muss in der TwinCAT Systemkonfiguration die *Store Category* [► 21] festgelegt werden. Diese wird bei allen EtherNet/IP-Geräten im Objekt F8000:2B "Advanced Options" eingetragen. Ist das entsprechende Bit gesetzt, wird die IP-Adresse aus dem Speicher verwendet, ist dort nichts eingetragen, wird das Bit ignoriert und es wird mit den Parametern des TwinCAT Systems gearbeitet.

Im folgenden Beispiel ist das Bit 8 (0x0100) gesetzt was bedeutet, dass die Store Category 1 angewählt ist, die wiederum die IP-Einstellung verändert (Index 0x8000: 21...23).

Slave Settings				
Index	Name	Flags	Value	
8000:0	Slave Settings (Box 2)	M RO	> 43 <	
8000:01	Slave Number	M RO	0x0002 (2)	
8000:03	Product Name	M RW	Box 2 (TC EtherNet/IP Slave)	
8000:04	Device Type	M RO	0x000C (12)	
8000:05	Vendor ID	M RO	0x006C (108)	
8000:06	Product Code	M RO	0x1888 (6280)	
8000:07	Revision	M RO	3.1	
8000:08	Serial Number	M RO	0x00000000 (0)	
8000:20	MAC Address	M RO	02 00 02 12 47 D6	
8000:21	IP Address	M RW	10.1.1.2	
8000:22	Network Mask	M RW	255.0.0.0	
8000:23	Gateway Address	M RW	0.0.0.0	
8000:24	DHCP Max Retries	M RW	0	
8000:25	TCP/IP TTL	M RW	128	
8000:26	TCP/IP UDP Checksum	M RW	TRUE	
8000:27	TCP/IP TCP Timeout	M RW	300 Seconds	
8000:28	MultiCast TTL	M RW	1	
8000:29	MultiCast UDP Checksum	M RW	FALSE	
8000:2A	Forward Class3 to PLC	M RW	FALSE	
8000:2B	Advanced Slave Options	M RW	0x0100 (256)	
8001:0	IO Assembly 5 Settings	M RO	> 12 <	
9000:0	Slave Info (Box 2)	RO	> 43 <	
9001:0	IO Assembly 5 Info	RO	> 12 <	

Möchten Sie Store Category 1 und 2 nutzen, ist ein 0x0300 im Objekt 8000:2B einzutragen. Es sind nur die Bits 8 und 9 zu verwenden, alle anderen Bits sind reserviert und dürfen nicht verwendet werden.

Um die Einstellungen aus der SPS heraus zu lesen oder zu schreiben, werden ADS Bausteine verwendet.

5.3.1 Objektbeschreibung

Offset	Name	Data Type	SubIndex	Store Category	
				1	2
0x00..0x01	ID	UINT16	1		
0x02..0x03	Reserved	UINT16	-		
0x04..0x23	Product Name	BYTE[32], STRING(31)	3		X
0x24..0x27	Device Type	UINT32	4		
0x28..0x2B	Vendor ID	UINT32	5		
0x2C..0x2F	Product Code	UINT32	6		X
0x30..0x33	Revision	UINT32	7		
0x34..0x37	Serial Number	UINT32	8		
0x38..0x7D	Reserved	BYTE[70]	-		
0x7E..0x83	MAC Address	BYTE[6]	32		
0x84..0x87	IP Address	UINT32	33	X	
0x88..0x8B	Network Mask	UINT32	34	X	
0x8C..0x8F	Gateway Address	UINT32	35	X	
0x90..0x91	DHCP Max Retries	UINT16	36		
0x92..0x93	TCP/IP TTL	UINT16	37		
0x94..0x95	TCP/IP UDP Checksum	UINT16	38		
0x96..0x97	TCP/IP TCP Timeout	UINT16	39		
0x98..0x99	Multicast TTL	UINT16	40		
0x9A..0x9B	Multicast Checksum	UINT16	41		
0x9C..0x9D	Forward Class3 to PLC	UINT16	42		
0x9E..0x9F	Flags	UINT16	43		
0xA0..0xFF	Reserved	Byte[96]	-		

Store Category

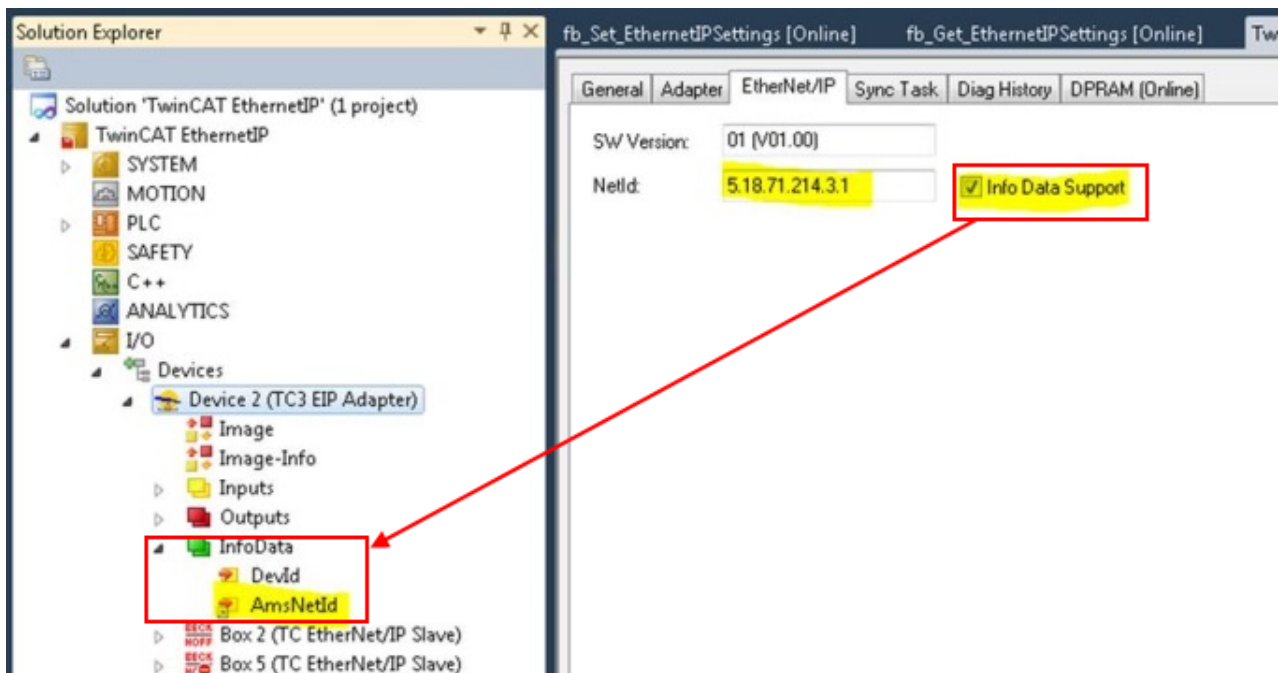
Die "Store Category" legt fest, welche Einstellungen mit den Werten aus dem remanenten Speicher überschrieben werden sollen. Dazu müssen im Projekt unter „Flags“ die Bits Bit9 - Bit8 entsprechend gesetzt werden. Um beides zu ändern, müssen beide Bits gesetzt sein.
(Bit9=Cat2, Bit8=Cat1)

5.3.2 ADS-Write Kommando

AmsNetId

Die AMSNetId finden Sie unter dem Karteireiter **EtherNet/IP**, im Feld **NetId**. Wenn Sie die Option **Info Data Support** anwählen, wird sie direkt verknüpft.

Vorteil einer direkten Verknüpfung ist, dass diese sich auch bei verschiedenen Steuerungen, die eine unterschiedliche AMSNETID verwenden, durch die Verknüpfung immer die aktuelle AMSNETID des Systems holt. Ein manuelles Auslesen der AMSNETID des EtherNet/IP Adapters entfällt somit.



ADS Port-Nummer

Bei der Function „EtherNet/IP Adapter“ stellen Sie die ADS Port-Nummer fest auf 0xFFFF ein.

Adapter

IDXGRP: 0x0001F480

IDXOFFS: 0x00000000

Einstellung zum Setzen (4 Byte + Objektgröße (256 Byte))

Byte Offset 0: 0x45

Byte Offset 1: 0x23

Byte Offset 2: ObjIndex LoByte (z. Bsp. 0x8000 für **Adapter 1** und 0x8010 für **Adapter 2**)

Byte Offset 3: ObjIndex HiByte

Byte Offset 4-260: Daten des Objects (siehe Objektbeschreibung unten)

Einstellung zum Zurücksetzen (4 Byte)

Byte Offset 0: 0x00

Byte Offset 1: 0x00

Byte Offset 2: ObjIndex LoByte (z. Bsp. 0x8000 für **Adapter 1** und 0x8010 für **Adapter 2**)

Byte Offset 3: ObjIndex HiByte

• Änderungen übernehmen

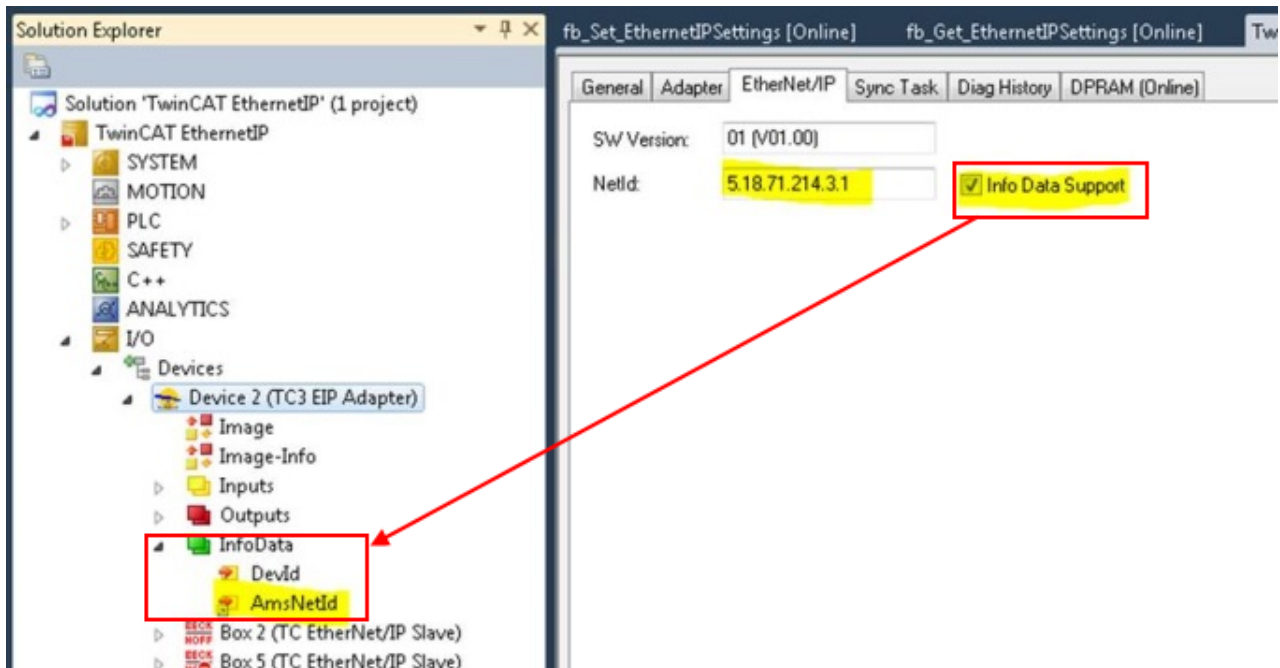
i Nach dem Einstellen der Eigenschaften ist bei der TF6280 ein TwinCAT-Restart durchzuführen, dann werden die neuen Settings übernommen und sind gültig. Die Einstellungen bleiben gespeichert und müssen nur dann erneut runtergespielt werden, wenn diese sich wieder geändert haben.

5.3.3 ADS-Read Kommando

AmsNetId

Die AMSNetId finden Sie unter dem Karteireiter **EtherNet/IP**, im Feld **NetId**. Wenn Sie die Option **Info Data Support** anwählen, wird sie direkt verknüpft.

Vorteil einer direkten Verknüpfung ist, dass diese sich auch bei verschiedenen Steuerungen, die eine unterschiedliche AMSNETID verwenden, durch die Verknüpfung immer die aktuelle AMSNETID des Systems holt. Ein manuelles Auslesen der AMSNETID des EtherNet/IP Adapters entfällt somit.



ADS Port-Nummer

Bei der Function „EtherNet/IP Adapter“ stellen Sie die ADS Port-Nummer fest auf 0xFFFF ein.

Adapter

IDXGRP: 0x1F480

IDXOFFS: 0x8000 für den **ersten Adapter**

IDXOFFS: 0x8010 für den **zweiten Adapter**

IDXOFFS: 0x8020 für den **dritten Adapter**

...

IDXOFFS: 0x8070 für den **achten Adapter**

LEN: 256

Die Daten werden wie oben beschrieben in dem Datenarray abgelegt -> siehe Objektbeschreibung [► 21].

5.3.4 Beispiel

Ein Beispielprogramm kann heruntergeladen werden: https://infosys.beckhoff.com/content/1031/TF6280_Tc3_EtherNetIPSlave/Resources/3105211403.tzip

5.4 Anlegen des EtherNet/IP Adapters in anderen EtherNet/IP Scannern

Alle wichtigen Informationen erhalten Sie im Karteireiter **Settings**:

General

Settings

Slave Settings

Index	Name	Flags	Value	Unit
8000:0	Slave Settings (Box 1)	MRO	> 43 <	
8001:0	IO Assembly 1 Settings	MRO	> 12 <	
8001:01	Assembly Number	MRO	0x0001 (1)	
8001:02	Configuration Instance	MRO	128	
8001:03	Configuration Size	MRO	0 Byte	
8001:04	Input Instance (T->O)	MRO	129	
8001:05	Input Size (T->O)	MRO	12 Byte	
8001:06	Output Instance (O->T)	MRO	130	
8001:07	Output Size (O->T)	MRO	12 Byte	
8001:08	Heartbeat Instance (Listen Onl...	MRO	136	
8001:09	Heartbeat Size (Listen Only)	MRO	0 Byte	
8001:...	Heartbeat Instance (Input Only)	MRO	137	
8001:...	Heartbeat Size (Input Only)	MRO	0 Byte	
8001:...	Advanced Assembly Options	MRW	0x0000 (0)	
9000:0	Slave Info (Box 1)	RO	> 43 <	
9001:0	IO Assembly 1 Info	RO	> 12 <	

Sie benötigen

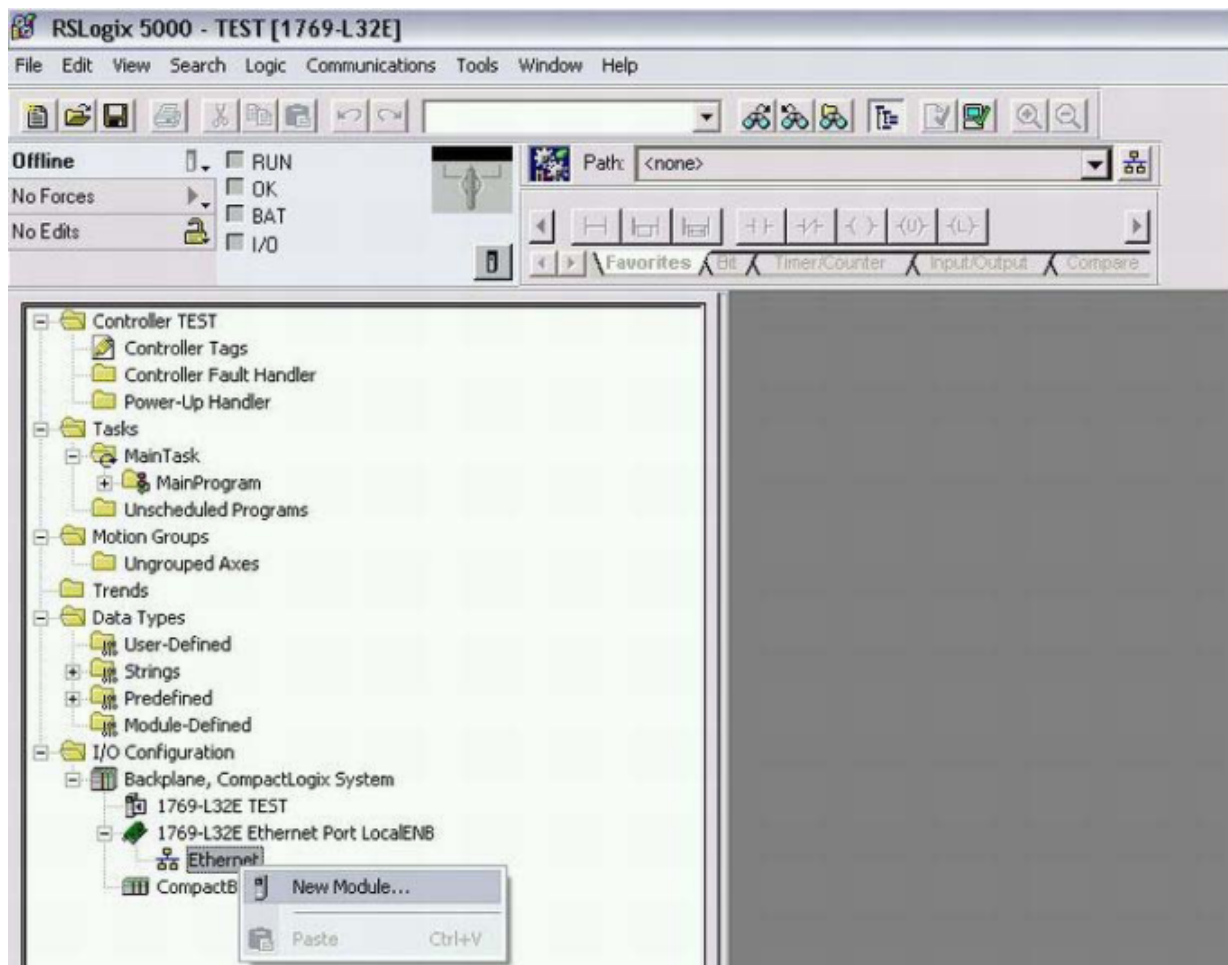
- die IP-Adresse des Adapters (siehe [Anlegen eines EtherNet/IP Adapters](#) [► 15])
- die „Assamby Instance“-Nummern (siehe Karteireiter Settings)
- die Anzahl an Daten (siehe Karteireiter Settings)
- die „Configuration Instance“-Nummer 128 Länge 0
- die „Input Instance“-Nummer 129 Länge 12
- die „Output Instance“-Nummer 130 Länge 12

Die Instance-Nummern sind immer gleich, das Exportieren der EDS Datei beinhaltet nur die Instance-Nummern, die Anzahl der Daten müssen noch eingetragen werden.

Das Einbinden des EtherNet/IP Adapters kann über eine „Generic Node“-Struktur oder mittels der EDS-Datei erfolgen.

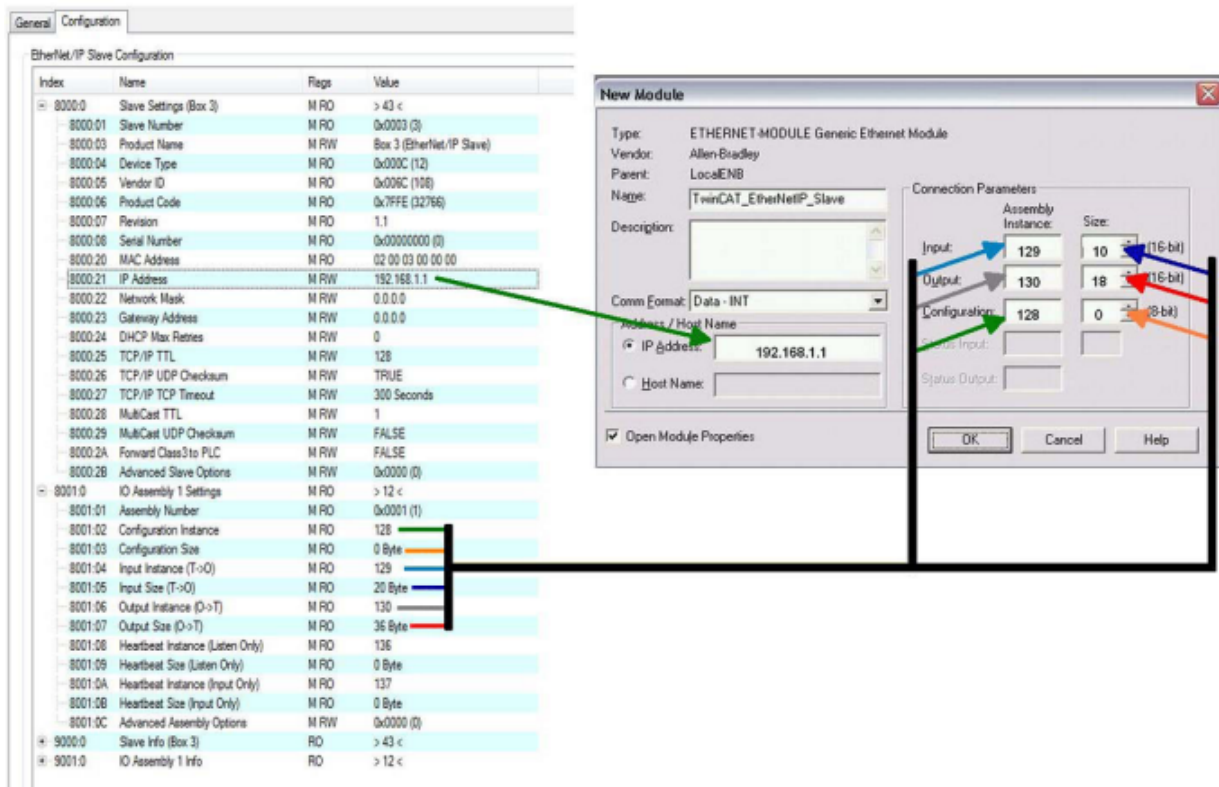
5.4.1 Beispiel für Rockwell CPUs

1. Wählen Sie unter **Ethernet, New Module...** und hier das ETHERNET-MODULE **Generic Ethernet Module** an.



2. Tragen Sie die IP-Adresse aus Objekt $0 \times 8000 : 21$ ein.
3. Tragen Sie bei Input Instance 129_{dez} ein.
4. Tragen Sie bei Output Instance 130_{dez} und
5. bei Config Instance 128_{dez} ein.

⇒ Die Datenlänge steht in Abhängigkeit des Comm Formats.



Eigenschaften des gewählten Comm-Formates berücksichtigen

Im Beispiel oben wurde das Comm Format *INT* gewählt, also müssen die Anzahl der Daten aus Objekt 0x8001:05 und 0x8001:07 noch mal durch 2 geteilt werden, da diese in TwinCAT in Byte angegeben werden und auf der RSLogix in Word-Länge (INT).

Wird eine ungerade Anzahl der Bytes angelegt, muss aufgerundet werden. Das gilt auch, wenn das Comm-Format auf DINT gestellt wird, dann muss auf die nächste ganzzahlige Zahl aufgerundet werden.



Systemlimitierungen

Achten Sie bei Multicast auf die dadurch entstehende hohe Netzwerklast besonders bei Systemen mit vielen oder kurzen Zykluszeiten. Eine hohe Netzwerklast kann unter Umständen die Kommunikation beeinträchtigen.

5.5 Azyklische Kommunikation via Explicit Messaging

5.5.1 Common Industrial Protocol (CIP)

Das Common Industrial Protocol (CIP) ist ein objektorientiertes Peer-to-Peer-Protokoll, das Verbindungen zwischen industriellen Geräten (Sensoren, Aktoren) und übergeordneten Geräten (Steuerungen) ermöglicht. CIP ist unabhängig von physikalischen Medien und der Datenübertragungsschicht. Dabei hat CIP zwei Hauptzwecke: Zum einen den Transport von Steuerungsorientierten Daten, welche mit E/A-Geräten verbunden sind und zum anderen den Transport von Informationen, die sich auf das zu steuernde System beziehen, wie zum Beispiel Konfigurationsparameter oder Diagnose.

CIP nutzt abstrakte Objekte, um einen Teilnehmer zu beschreiben. Ein CIP-Teilnehmer besteht aus einer Gruppe von Objekten. Objekte beschreiben die verfügbaren Kommunikationsdienste, das nach außen sichtbare Verhalten des Teilnehmers und einen Weg, wie Informationen abgerufen und ausgetauscht werden können. CIP-Objekte sind in Klassen, Instanzen und Attribute aufgeteilt. Eine Klasse ist eine Menge von Objekten, die alle die gleiche Komponente repräsentieren. Eine Instanz ist die aktuelle Darstellung eines

bestimmten Objekts. Jede Instanz hat die gleichen Attribute, jedoch mit möglicherweise unterschiedlichen Attributwerten. Adressiert werden die einzelnen Objekte über eine Node-Adresse, bei EtherNet/IP die IP-Adresse, sowie Class, Instance und Attribute.

- Objekt
 - Eine abstrakte Darstellung einer bestimmten Komponente innerhalb eines Produkts.
- Klasse
 - Eine Menge von Objekten, die alle die gleiche Art von Systemkomponente darstellen. Eine Klasse ist eine Verallgemeinerung eines Objekts. Alle Objekte in einer Klasse sind in Form und Verhalten identisch, können aber unterschiedliche Attributwerte enthalten.
- Instanz
 - Ein spezifisches und reales Exemplar eines Objekts.
Zum Beispiel: Berlin ist eine Instanz der Objektklasse Hauptstadt.
- Attribut
 - Eine Beschreibung einer Eigenschaft oder eines Merkmals eines Objekts. Typischerweise liefern Attribute Statusinformationen oder regeln den Betrieb eines Objekts.

(Quelle: The CIP Networks Library Volume 1: Common Industrial Protocol, Edition 3.22)

Folgende Objekte werden von Beckhoff intern verwendet und sind somit reserviert:

1. Identity Object → Class 0x1
2. Message Router Object → Class 0x2
3. Assembly Object → Class 0x4
4. Connection Manager Object → Class 0x6
5. TCP/IP Interface Object → Class 0xF5
6. Ethernet Link Object → Class 0xF6

5.5.2 Forward Message to AMS Port via Explicit Messaging

„Explicit Messaging“ wird für die Übermittlung von Informationen und Daten verwendet, die keine kontinuierlichen Aktualisierungen erfordern. Mit „Explicit Messaging“ können Sie die Parameter eines Slave-Geräts im Ethernet/IP-Netzwerk konfigurieren und überwachen.

Mit dem Feature “FwdMsgToAmsPort” besteht die Möglichkeit, azyklische Anfragen von Ethernet/IP-Scannern via Explicit Messaging zu verarbeiten.

Das folgende Beispiel zeigt die Realisierung einer azyklischen Kommunikation zwischen einer TwinCAT 3 und einer RS Logix-Steuerung.

Realisierung TwinCAT 3:

- ✓ Voraussetzung: Ethernet/IP-Treiber-Version, min. V1.23
- 1. Zum Aktivieren des Features “FwdMsgToAmsPort” tragen Sie bei den Slave-/Master-Settings (0x8000:2A/0xF800:2A) in TwinCAT der AmsPort der PLC (im Beispiel 851) der SPS ein.

The top screenshot shows the 'Project Settings' dialog for 'EtherNetIP_Example_FwdMsgToAmsPort'. The 'Port' field is set to 851. The bottom screenshot shows the 'Slave Settings' table, where the 'Forward Class3 to AmsPort' setting (Index 8000:2A) is also set to 851. A red arrow points from the 'Port' field in the top screenshot to the 'Forward Class3 to AmsPort' field in the bottom screenshot.

Project Settings

Project Name:	EtherNetIP_Example_FwdMsgToAmsPort	ID:	1
Project Path:	EtherNetIP_Example_FwdMsgToAmsPort		
Project Type:	Plc Project	Port:	851
Project Guid:	{397F6E45-5F82-432A-AB9B-1DC79C8C4094}		
Encryption:	No boot project encryption (default)		
☒ Autostart Boot Project ☒ Symbolic Mapping ☐ Force Multi Instance			
Comment:			
Compiler Defines	Manual:		

Slave Settings

Index	Name	Flags	Value
8000:0	Slave Settings (Box 1)	M RO	> 43 <
8000:01	Slave Number	M RO	0x0001 (1)
8000:03	Product Name	M RW	Box 1 (TC EtherNet/IP Slave)
8000:04	Device Type	M RO	0x000C (12)
8000:05	Vendor ID	M RO	0x006C (108)
8000:06	Product Code	M RO	0x1888 (6280)
8000:07	Revision	M RO	3.1
8000:08	Serial Number	M RO	0x00000000 (0)
8000:20	MAC Address	M RO	02 00 01 17 EA 3C
8000:21	IP Address	M RW	192.168.1.213
8000:22	Network Mask	M RW	255.255.255.0
8000:23	Gateway Address	M RW	0.0.0.0
8000:24	DHCP Max Retries	M RW	0
8000:25	TCP/IP TTL	M RW	128
8000:26	TCP/IP UDP Checksum	M RW	TRUE
8000:27	TCP/IP TCP Timeout	M RW	30 Seconds
8000:28	MultiCast TTL	M RW	1
8000:29	MultiCast UDP Checksum	M RW	FALSE
8000:2A	Forward Class3 to AmsPort	M RW	851
8000:2B	Advanced Slave Options	M RW	0x0000 (0)
8001:0	IO Assembly 1 Settings	M RO	> 12 <
9000:0	Slave Info (Box 1)	RO	> 43 <
9001:0	IO Assembly 1 Info	RO	> 12 <

2. Es werden ADSRDWRT-Anfragen (ADSRDWRT-Requests) von dem Ethernet/IP-Treiber (IDGRP: 0x848180E9 IOFFS: SlaveId (Adapter)) an die SPS-Task als Indications registriert und dies erlaubt deren Bearbeitung. Dazu wird der ADSRDWRTIND-Funktionsbaustein verwendet.

⇒ In der vom Ethernet/IP-Treiber registrierten Indication befindet sich als erstes ein 32 Byte (8xULONG) Header:

```

TYPE DUT_MsgToAmsPortHeader:
STRUCT
  nServiceCode:UDINT;
  nClassId:UDINT;
  nInstanceId:UDINT;
  nAttributeId:UDINT;
  nReservedId:UDINT;
  nGeneralStatus:UDINT;
  nAdditionalStatus:UDINT;
  nDataLen:UDINT;
END_STRUCT
END_TYPE

```

```

TYPE DUT_IncomingMsgRequest:
STRUCT
    reqHdr:DUT_MsgToAmsPortHeader;
    reqData:ARRAY [0...991] OF BYTE;
END_STRUCT
END_TYPE

```

```

TYPE DUT_OutgoingMsgResponse:
STRUCT
    resHdr:DUT_magToAmsPortHeader;
    resData:ARRAY [0...991] OF BYTE;
End_Struct
END_TYPE

```

Derselbe Header wird auch für das Response verwendet.

- Die eigentlichen Schreib-/Lesedaten folgen direkt nach dem Header (nDataLen <> 0 ist entsprechend der Datenlänge zu setzen). Die maximal unterstützte Datenlänge beträgt 992 Byte (+ 32 Byte Header = 1024 Byte). Mögliche Klassen/Instanzen/Attribut Werte

	Min	Max
Klasse	1	0xFFFF
Instanz	1	0xFFFF
Attribut	1	0xFFFF

- Nachdem eine Indication bearbeitet wurde, muss eine Antwort über den ADSRDWRTRES-Funktionsbaustein an das Quellgerät gesendet werden.

```

PROGRAM MAIN
VAR
    i
    IdxGroup      : UDINT;           //Ethernet/IP-Treiber -> 16#848180E9
    IdxOffset     : UDINT;           //SlaveId (Adapter) bzw. 0xFFFF (Scanner)
    fbADSRDWRINDEX : ADSRDWRINDEX;
    fbAdsRdWrRes  : ADSRDWRTRES;
    request       : DUT_IncomingMsgRequest;
    response      : DUT_OutgoingMsgResponse;
    nResponseLen  : UINT;
    nAdsResult    : UDINT:=0;
    nAdsResponsesSent : UDINT;
    Attributes    : ARRAY [1..4] OF STRING :=['TestReadOnlyAttribute1','TestReadOnlyAttrib
ute2','TestReadOnlyAttribute3','TestReadWriteAttribute4'];
END_VAR

```

```

CASE i OF
0: //check for ADSReadWrite-Requests
    fbADSRDWRINDEX (
    CLEAR:=FALSE ,
    MINIDXGRP:= 16#84000000,
    VALID=> ,
    NETID=> ,
    PORT=> ,
    INVOKEID=> ,
    IDXGRP=> ,
    IDXOFFS=> ,
    );

    IF fbADSRDWRINDEX.VALID THEN
        IdxGroup:= fbADSRDWRINDEX.IDXGRP;
        IdxOffset:= fbADSRDWRINDEX.IDXOFFS ;
        MEMSET(ADR(request), 0, SIZEOF(request));
        MEMSET(ADR(response), 0, SIZEOF(response));
        nResponseLen:=0;
        //check for Indication Request = Ethernet/IP-driver -> 16#848180E9
        IF IdxGroup = 16#848180E9 THEN
            //check for Indication.datalength >= DUT_MsgToAmsPortHeader
            IF fbADSRDWRINDEX.WRTLENGTH >= SIZEOF(request.reqHdr) THEN
                MEMCPY(ADR(request.reqHdr), fbADSRDWRINDEX.DATAADDR, SIZEOF(request.reqHdr));
            END_IF
            //check for Indication.datalength > DUT_MsgToAmsPortHeader >>> save additional data
            IF fbADSRDWRINDEX.WRTLENGTH > SIZEOF(request.reqHdr) THEN
                MEMCPY(ADR(request.reqData), fbADSRDWRINDEX.DATAADDR+SIZEOF(request.reqHdr), fbADSRDW
RTINDEX.WRTLENGTH-SIZEOF(request.reqHdr));
            END_IF
            i:=10;
        ELSE
            i:=20;
        END_IF
    END_IF

```

```

        END_IF
    END_IF

10:    //new Ind from EthIp-Drv received
    response.resHdr.nServiceCode := request.reqHdr.nServiceCode OR CONST.CN_SC_REPLY_MASK;
    response.resHdr.nGeneralStatus := 0;
    response.resHdr.nAdditionalStatus := 0;
    response.resHdr.nDataLen := 0;
    IF request.reqHdr.nServiceCode = CONST.CN_SC_GET_ATTR_SINGLE OR request.reqHdr.nServiceCode = CONS
T.CN_SC_SET_ATTR_SINGLE THEN
        i:=11;
    ELSE
        response.resHdr.nGeneralStatus := CONST.CN_GRC_BAD_SERVICE;
        nResponseLen := SIZEOF(response.resHdr);
        i:=20;
    END_IF

11:    //case decision for request
    CASE request.reqHdr.nClassId OF
        16#1000:    //erlaubte Beispiel Class 0x10000
            CASE request.reqHdr.nInstanceId OF
                16#1:    //erlaubte Beispiel Instance 0x1
                    CASE request.reqHdr.nAttributeId OF    //
Attributes 1-4 erlaubt; only attr 4 is settable
                        1,2,3:    IF request.reqHdr.nServiceCode = CONST.CN_SC_SET_ATTR_SINGLE THEN
                                response.resHdr.nGeneralStatus := CONST.CN_GRC_ATTR_NOT_SETTABLE;
                                nResponseLen := SIZEOF(response.resHdr);
                                i:=20;
                                ELSE
                                    i:=12;
                                END_IF
                        4:    IF request.reqHdr.nServiceCode = CONST.CN_SC_SET_ATTR_SINGLE THEN
                                i:=14;
                                ELSE
                                    i:=12;
                                END_IF
                        ELSE
                                response.resHdr.nGeneralStatus := CONST.CN_GRC_UNDEFINED_ATTR;
                                nResponseLen := SIZEOF(response.resHdr);
                                i:=20;
                                END_CASE
                        ELSE
                                response.resHdr.nGeneralStatus := CONST.CN_GRC_BAD_CLASS_INSTANCE;
                                nResponseLen := SIZEOF(response.resHdr);
                                i:=20;
                                END_CASE
                        ELSE
                                response.resHdr.nGeneralStatus := CONST.CN_GRC_BAD_CLASS_INSTANCE;
                                nResponseLen := SIZEOF(response.resHdr);
                                i:=20;
                                END_CASE
                        END_CASE

12:    //GetAttribute
    response.resHdr.nGeneralStatus := CONST.CN_GRC_SUCCESS;
    MEMCPY(ADR(response.resData), ADR(Attributes[request.reqHdr.nAttributeId]), SIZEOF(Attributes[requ
est.reqHdr.nAttributeId]));
    response.resHdr.nDataLen := INT_TO_UINT(LEN(Attributes[request.reqHdr.nAttributeId]));
    nResponseLen := UDINT_TO_UINT(response.resHdr.nDataLen) + SIZEOF(response.resHdr);
    i:=20;

14:    //SetAttribute
    response.resHdr.nGeneralStatus := CONST.CN_GRC_SUCCESS;
    IF request.reqHdr.nDataLen <= SIZEOF(STRING)-1 THEN
        MEMCPY(ADR(Attributes[request.reqHdr.nAttributeId]), ADR(request.reqData), request.reqHdr.nDat
aLen);
    ELSE
        response.resHdr.nGeneralStatus := CONST.CN_GRC_BAD_DATA;
    END_IF
    nResponseLen := SIZEOF(response.resHdr);
    i:=20;

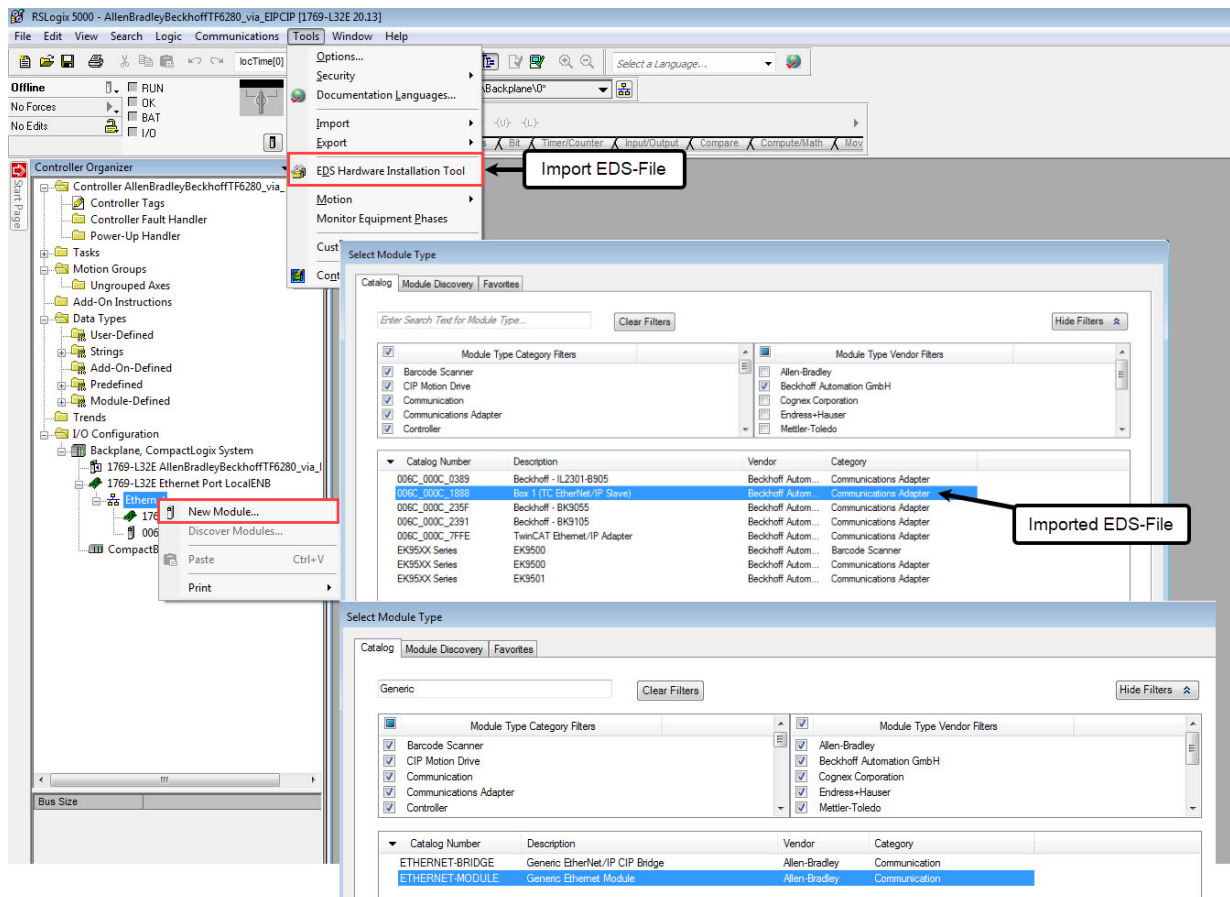
20:    //response to Ethernet/IP-driver
    fbAdsRdWrRes(
    NETID:= fbADSRDWRINDEX.NETID ,
    PORT:= fbADSRDWRINDEX.PORT ,
    INVOKEID:= fbADSRDWRINDEX.INVOKEID ,
    RESULT:=nAdsResult ,
    LEN:=nResponseLen,
    DATAADDR:=ADR(Response) ,
    RESPOND:=TRUE );

```

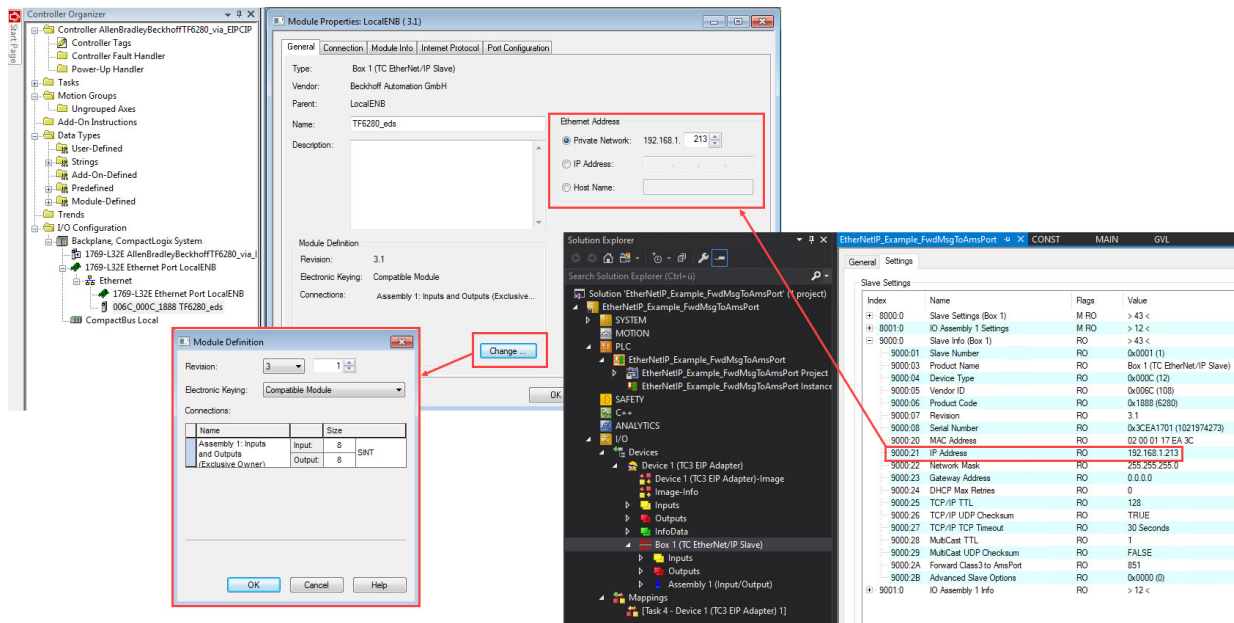
```
i:=21;
nAdsResponsesSent:=nAdsResponsesSent+1;
fbADSRDWRINDEX (CLEAR:=TRUE);
21: i:=0;
fbAdsRdWrRes (RESPOND:=FALSE);
END_CASE
```

Realisierung RS Logix 5000:

1. Zu Beginn müssen Sie ein neues Modul anlegen, entweder ein „Generic Ethernet Module“ oder ein aus TwinCAT exportiertes EDS-File.
Der Vorteil der importierten EDS-File ist der, dass dieses bereits die Größe der Prozessdaten, welche in der TwinCAT-Konfiguration angelegt worden sind, beinhaltet.



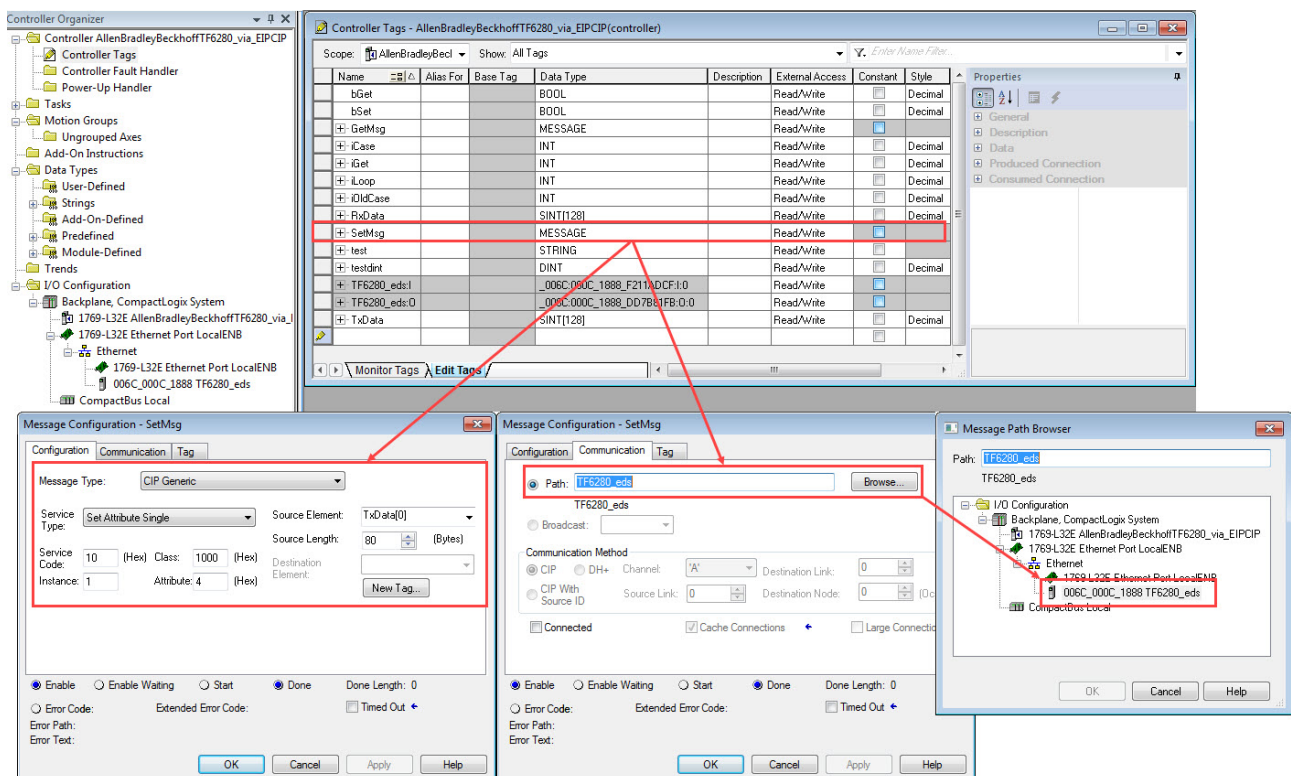
2. In den Einstellungen des angefügten Moduls müssen Sie gegebenenfalls die IP- und die Prozessdateneinstellungen anpassen.



3. Um azyklisch Nachrichten zu senden und empfangen zu können, sind Strukturen vom Typ „Messages“ notwendig.

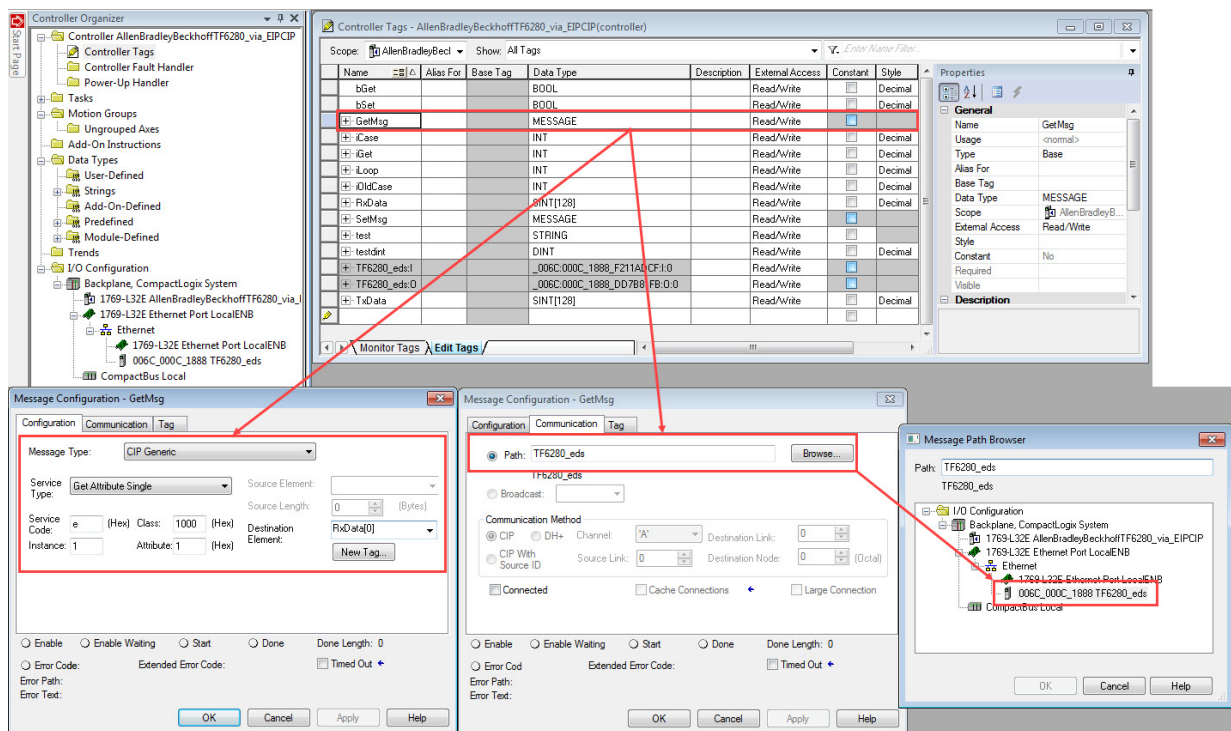
Im Beispiel dient eine Struktur zum Senden und eine zum Empfangen. Beide Strukturen müssen Sie entsprechend für das Senden bzw. für das Empfangen konfigurieren.

4. Mit einem Klick der rechten Maustaste auf den Tag **SetMsg-Configure SetMsg** öffnen Sie die Konfigurationseinstellungen. Diese sind wie im Screenshot angegeben zu übernehmen. Die Angaben **Class**, **Instance** und **Attribute** sind frei wählbar. Bei **Service Type** stellen Sie **Set Attribute Single** ein. Legen Sie unter **Source Element** ein Array an, dessen Inhalt versendet werden soll. Die **Source Length** wählen Sie so, dass diese nicht die Länge der in TwinCAT angelegten Ziel-Variable überschreitet.



5. Mit einem Klick der rechten Maustaste auf den Tag **GetMsg - Configure GetMsg** öffnen Sie die Konfigurationseinstellungen. Diese sind wie im Screenshot angegeben zu übernehmen. Die Angaben **Class**, **Instance** und **Attribute** sind frei wählbar. Bei **Service Type** stellen Sie **Get**

Attribute Single ein. Legen Sie unter **Destination Element** ein Array an, welches die azyklischen Nachrichten empfängt. Die Größe des Arrays ist entsprechend den empfangenden Nachrichten zu wählen.



⇒ Der folgende Beispielcode sendet Anfragen an den Ethernet/IP-Treiber der TF6280, welcher diese an die TwinCAT-PLC zur weiteren Verarbeitung weiterleitet.

Mit einer positiven Flanke an „bGet“ wird ein einzelner Attributwert aus der TwinCAT PLC gelesen. In diesem Beispiel können die Werte „TestReadOnlyAttribute1, TestReadOnlyAttribute2 und TestReadOnlyAttribute3“ gelesen werden.

Mit einer positiven Flanke an „bSet“ wird ein einzelner Attributwert in der TwinCAT PLC geschrieben. In diesem Beispiel kann das vierte Attribut in der TwinCAT PLC mit dem Inhalt „123Beckhoff567“ und „HelloBeckhoff“ beschrieben werden.

```
//GetAttribute
IF bGet THEN
    bGet:=0;
    iCase:=20+iGet;
END_IF;

//SetAttribute
IF bSet AND iOldCase=5 THEN
    bSet:=0;
    iCase:=6;
ELSIF bSet AND iOldCase=6 THEN
    bSet:=0;
    iCase:=5;
END_IF;

CASE iCase OF
5:    //HelloBeckhoff --> (ASCII)
    iOldCase:=iCase;

    TxData[0]:=72;           //H
    TxData[1]:=101;          //e
    TxData[2]:=108;          //l
    TxData[3]:=108;          //l
    TxData[4]:=111;          //o
    TxData[5]:=66;           //B
    TxData[6]:=101;          //e
    TxData[7]:=99;           //c
    TxData[8]:=107;          //k
    TxData[9]:=104;          //h
    TxData[10]:=111;         //o
    TxData[11]:=102;         //f
    TxData[12]:=102;         //f
    iCase:=10;
```

```
6:    //123Beckhoff567 --> (ASCII)
    iOldCase:=iCase;

    TxData[0]:=49;           //1
    TxData[1]:=50;           //2
    TxData[2]:=51;           //3
    TxData[3]:=66;           //B
    TxData[4]:=101;          //e
    TxData[5]:=99;           //c
    TxData[6]:=107;          //k
    TxData[7]:=104;          //h
    TxData[8]:=111;          //o
    TxData[9]:=102;          //f
    TxData[10]:=102;         //f
    TxData[11]:=52;          //4
    TxData[12]:=53;          //5
    TxData[13]:=54;          //6
    iCase:=10;

10:    //SetAttribute
    msg(SetMsg);
    IF SetMsg.DN OR SetMsg.ER THEN
        FOR iLoop:=0 TO 80 DO
            TxData[iLoop]:=0;
        end_FOR;
        iCase:=0;
    END_IF;

20:    //TestReadOnlyAttribute1
    GetMsg.Class:=16#1000;
    GetMsg.Instance:=16#01;
    GetMsg.Attribute:=16#01;
    iCase:=30;

21:    //TestReadOnlyAttribute2
    GetMsg.Class:=16#1000;
    GetMsg.Instance:=16#01;
    GetMsg.Attribute:=16#02;
    iCase:=30;

22:    //TestReadOnlyAttribute3
    GetMsg.Class:=16#1000;
    GetMsg.Instance:=16#01;
    GetMsg.Attribute:=16#03;
    iCase:=30;

30:    //GetAttribute
    msg(GetMsg);
    IF GetMsg.DN OR GetMsg.ER then
        iGet:=iGet+1;
        IF iGet >= 3 THEN
            iGet:=0;
        END_IF;
        iCase:=0;
    END_IF;

END_CASE;
```

Hier finden Sie das dokumentierte Beispiel als TwinCAT Projekt: https://infosys.beckhoff.com/content/1031/TF6280_Tc3_EthernetIPSlave/Resources/14758092427.zip.

6 Eigenschaften

6.1 Virtueller Adapter

Mit der TF6280 können Sie mit einer physikalischen Schnittstelle bis zu acht Adapter parametrieren. Hierbei wird jeweils eine virtuelle MAC-Adresse für ein virtuelles Adapter-Gerät gebildet, um damit bis zu acht EtherNet/IP-Adapter an einem PC über eine Ethernet-Schnittstelle betreiben zu können.

Dies hat den Vorteil, dass acht EtherNet/IP-Steuerungen einfach verbunden werden können bzw. dass Limitierungen bei der Buskommunikation mit dem Adapter umgangen werden, ohne eine zusätzliche Hardware zu verwenden

Dieses Feature kann zum Beispiel genutzt werden, um mit einem EtherNet/IP-Scanner eine größere Menge Daten auszutauschen oder sich mit mehreren EtherNet/IP-Scannern in verschiedenen Subnetzen zu verbinden.

Legen Sie in der TwinCAT Systemkonfiguration eine weitere Box an und verfahren Sie genauso, wie bei der Konfiguration eines realen Adapters.



Eindeutige MAC-Adresse

Achten Sie bei der manuellen Vergabe der virtuellen MAC-Adresse darauf, dass diese in Ihrem Netzwerk eindeutig ist.

6.2 TF6280 - Konfigurations-Parameter

6.2.1 Index 0x8000 Adapter Settings

Index	Name	Bedeutung
8000:0	Adapter Settings	
8000:1	Adapter Number	Adapter Box ID
8000:3	Product Name	Name des Gerätes
8000:4	Device Type	Geräte Typ
8000:5	Vendor ID	Hersteller-Nummer
8000:6	Product Code	Produkt Code
8000:7	Revision	Version
8000:8	Serial Number	Serien-Nummer (siehe Objekt 0x9000)
8000:20	MAC Address	MAC Adresse (siehe Objekt 0x9000)
8000:21	IP Address	IP-Adresse • 0.0.0.0: Wird dynamisch vom DHCP-Dienst vergeben • 255.255.255.255: Verwendung der Betriebssystemadresse Andernfalls: statisch vergebene IP-Adresse
8000:22	Network Mask	Subnetzmaske • 0.0.0.0: Wird dynamisch vom DHCP-Dienst vergeben Andernfalls: statisch vergebene Subnetzmaske
8000:23	Gateway Adress	Gateway-Adresse • 0.0.0.0: Wird dynamisch vom DHCP-Dienst vergeben Andernfalls: statisch vergebene Gateway-Adresse
8000:24	DHCP Max Retries	0: Kontinuierliche Wiederholung der DHCP-Adressierungsversuche. (Aktuell ist nur dieser Modus implementiert, Stand: 10-2016)
8000:25	TCP/IP TTL	„Time to live“ – Wert für Unicast TCP/UDP-Kommunikation
8000:26	TCP/IP UDP Checksum	Prüfsummenfunktion (Unicast): • 0: UDP-Prüfsumme deaktiviert. • 1: UDP-Prüfsumme aktiviert
8000:27	TCP/IP TCP Timeout	Zeitschalter für inaktive TCP-Verbindung in Sekunden • 0: Zeitschalter deaktiviert
8000:28	MultiCast TTL	„Time to live“-Wert für Multicast-UDP-Kommunikation
8000:29	MultiCast UDP Checksum	Prüfsummenfunktion (Multicast): • 0: UDP-Prüfsumme deaktiviert • 1: UDP-Prüfsumme aktiviert
8000:2A	Forward Class3 to PLC	Nachrichtenweiterleitung an die SPS (Derzeit nicht implementiert, Stand: 10-2016)
8000:2B	Advanced Adapter Options	„Store Category“-Parameter • Bit9=Cat2, • Bit8=Cat1 siehe IP-Adresse aus der SPS beschreiben [► 19]

6.2.2 Index 0x8001 IO Assembly Settings

Index	Name	Bedeutung
8001:0	IO Assembly Settings	
8001:1	Assembly Number	Assembly Id
8001:1	Configuration Instance	Konfigurationsinstanz
8001:3	Configuration Size	Konfigurationsgröße (immer 0)
8001:4	Input Instance (T->O)	Verknüpfungspunkt für Eingangswerte (T->O: Target->Originator)
8001:5	Input Size (T->O)	Größe der Eingangswerte (in Bytes)
8001:6	Output Instance (O->T)	Verknüpfungspunkt für Ausgangswerte (O->T, Originator->Target)
8001:7	Output Size (O-T)	Größe der Ausgangswerte (in Bytes)
8001:8	Heartbeat Instance (Listen Only)*	Heartbeat-Verknüpfungspunkt (nur für Mithören-Verbindungen)
8001:9	Heartbeat Size (Listen Only)*	Immer 0
8001:A	Heartbeat Instance (Input Only)**	Heartbeat-Verknüpfungspunkt (nur für Eingangs-Verbindungen)
8001:B	Heartbeat Size (Input Only)**	Immer 0
8001:C	Advanced Assembly Options	Bit 14: 0x4000 hex 0 = Standardeinstellung 1 = Deaktiviert die Verknüpfung zwischen "ConnCtrl" und „ConnState“ zur EtherNet/IP-IO-Verbindung Die anderen Bits sind immer mit 0 eingestellt (reserviert)

* Heartbeat Instance (Listen Only): Erlaubt das Mithören der Eingangsdaten (beim TF6280 sind das die Ausgangsdaten) wenn eine Verbindung besteht. Die "Listen Only" Verbindung wird mit Beenden der normalen Verbindung auch beendet.

** Heartbeat Instance (Input Only): Erlaubt das Mitlesen der Eingangsdaten (beim TF6280 sind das die Ausgangsdaten). Diese Verbindung ist unabhängig von der eigentlichen Kommunikation.

Der Heartbeat ist für beide Verbindungstypen (Listen-Only und Input-Only) zur Überwachung der Verbindung notwendig.

6.2.3 Index 0x9000 Adapter Info

Hier werden die aktuell gültigen Einstellungen angezeigt, diese können vom Objekt 0x8000 abweichen. Das Objekt 0x9000 zeigt die aktiven Parameter an.

6.2.4 Index 0x9001 IO Assembly Info

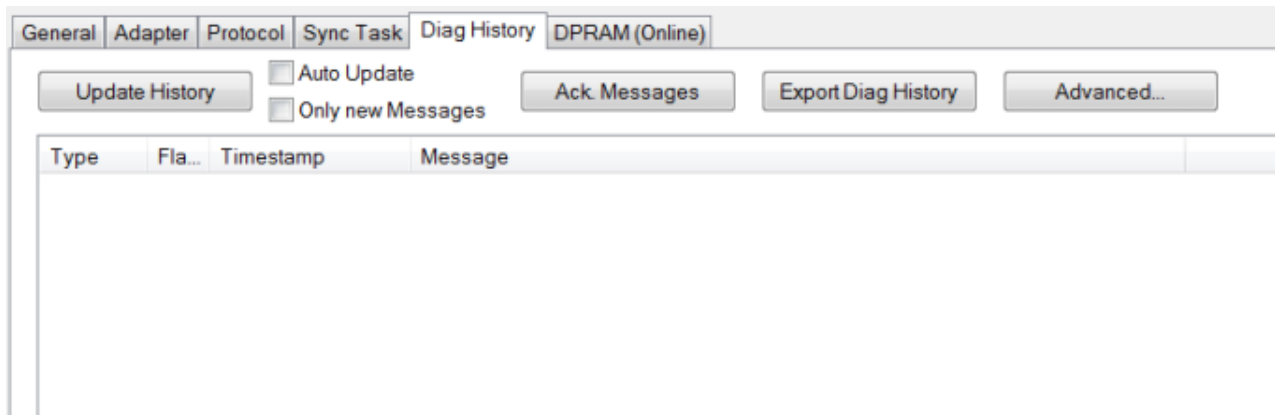
Hier werden die aktuell gültigen Assembly-Einstellungen angezeigt, diese können vom Objekt 0x8001 abweichen. Das Objekt 0x9001 zeigt die aktiven Parameter an.

7 Diagnose History

Die Diagnose History ist ein Hilfsmittel, um den Status des EtherNet/IP Interface zu überwachen und im Klartext die Diagnose-Meldungen mit Zeitstempel anzuzeigen.

Des Weiteren werden Informationen / Fehler, die in der Vergangenheit aufgetreten sind, mitgeloggt, um auch später noch eine genaue Fehlerursache festzustellen zu können. Dies betrifft auch Fehler, die nur so kurz anstanden, dass sie als Meldung nicht sichtbar waren.

Die Diagnose History ist ein Bestandteil des TwinCAT Systems und ist dort unter dem Baueintrag Geräte EtherNet/IP im Karteireiter **Diag History** zu finden.



7.1 Error Codes TF6280

Error	Code hex / (decimal)	Beschreibung	Abhilfe/Bedeutung
CN_ORC_ALREADY_USED	0x100 / (256)	Connection already in use	Die Verbindung besteht bereits, verwenden Sie eine andere Verbindung oder schließen Sie diese.
CN_ORC_BAD_TRANSPORT	0x103 / (259)	Transport type not supported	Der Transport Typ wird nicht unterstützt
CN_ORC_OWNER_CONFLICT	0x106 / (262)	More than one guy configuring	Es besteht bereits eine Verbindung, eine weitere kann nicht aufgebaut werden
CN_ORC_BAD_CONNECTION	0x107 / (263)	Trying to close inactive conn	Fehlerhafte Connection
CN_ORC_BAD_CONN_TYPE	0x108 / (264)	Unsupported connection type	Der Connection Typ wird nicht unterstützt, kontrollieren Sie ihre Einstellung.
CN_ORC_BAD_CONN_SIZE	0x109 / (265)	Connection size mismatch	Die Connection Größe passt nicht, kontrollieren Sie ihre Einstellung.
CN_ORC_CONN_UNCONFIGURED	0x110 / (272)	Connection unconfigured	Verbindung wurde nicht konfiguriert
CN_ORC_BAD_RPI	0x111 / (273)	Unsupportable RPI	In der Regel passt die Task Zeit nicht, achten Sie darauf, dass die EL6652 mit 1 ms intern arbeitet und Sie dies mit dem Cycle Time Multiplier einstellen können. Ansonsten passen Sie die Task Zeit an.
CN_ORC_NO_CM_RESOURCES	0x113 / (275)	Conn Mgr out of connections	Es stehen keine weiteren Recourcen zur Verfügung
CN_ORC_BAD_VENDOR_PRODUCT	0x114 / (276)	Mismatch in electronic key	Fehlerhafte Vendor Nummer
CN_ORC_BAD_DEVICE_TYPE	0x115 / (277)	Mismatch in electronic key	Fehlerhafter Device Type
CN_ORC_BAD_REVISION	0x116 / (278)	Mismatch in electronic key	Fehlerhafte Revision Number
CN_ORC_BAD_CONN_POINT	0x117 / (279)	Nonexistant instance number	Fehlerhafte Connection Number
CN_ORC_BAD_CONFIGURATION	0x118 / (280)	Bad config instance number	Fehlerhafte Konfiguration
CN_ORC_CONN_REQ_FAILS	0x119 / (281)	No controlling connection open	Connection konnte nicht aufgebaut werden
CN_ORC_NO_APP_RESOURCES	0x11A / (282)	App out of connections	Keine freien Connections mehr zur Verfügung.

Sollten Sie die Fehler nicht selbstständig lösen können, benötigt der Support folgende Informationen:

- TwinCAT Versions- und Build-Nummer und eine
- Wireshark Aufzeichnung

Wireshark-Aufzeichnung vorbereiten

Die Wireshark-Aufzeichnung kann mit einem Netzwerk-Hub, einem Netzwerk-Switch mit Port-Spiegelung, zum Beispiel mit dem Beckhoff ET2000, oder mit dem **Promiscuous Mode** des TwinCAT Systems erstellt werden.

General	Adapter	Protocol	Sync Task	Diag History	DPRAM (Online)
☒ Network Adapter ☒ OS (NDIS) ☐ PCI ☐ DPRAM Description: LAN-Verbindung (Intel(R) Ethernet Connection I218-LM - VirtualBox Br Device Name: \\DEVICE\\{C706CD25-DCCF-42A7-B4B7-81D7E66BD979} PCI Bus/Slot: Search... MAC Address: ec f4 bb 1f 7e 88 Compatible Devices... IP Address: 169.254.254.51 (255.255.0.0) ☒ Promiscuous Mode (use with Wireshark only) ☐ Virtual Device Names					

☐ Adapter Reference

Adapter:

Freerun Cycle (ms):

4

8 Anhang

8.1 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser Downloadfinder beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157

E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460

E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0
E-Mail: info@beckhoff.com
Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.com/tf6280

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

