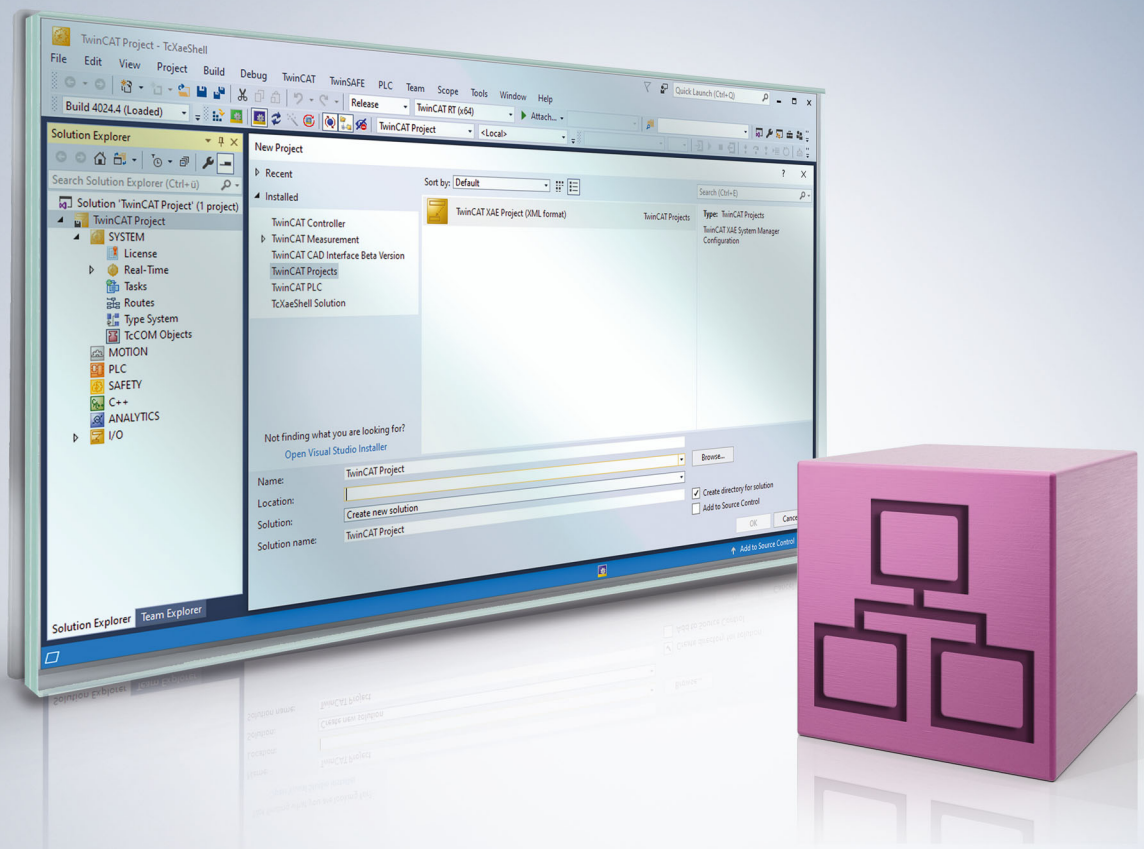
BECKHOFF New Automation Technology

Handbuch | DE

TF6620

TwinCAT 3 | S7 Communication



13.08.2025 | Version: 1.5.0

Inhaltsverzeichnis

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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® 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 Kennzeichnungen führen.

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

Copyright

Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: https://www.beckhoff.com/trademarks.

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.

1.4 Ausgabestände der Dokumentation

Version	Änderung
1.5.x	**Neu:** AddReadByteArray [▶ 42] AddWriteByteArray [▶ 46]

2 Übersicht

TwinCAT S7 Communication ermöglicht einen Datenaustausch zwischen dem TwinCAT System und einer Siemens S7 Steuerung. Das Produkt steht sowohl in Form eines einfach zu konfigurierenden TwinCAT I/O Geräts als auch einer SPS Bibliothek zur Verfügung. Die zugrunde liegende Protokollimplementierung basiert auf dem TwinCAT TCP/UDP RT Treiber und erlaubt das Absetzen von Lese/Schreib-Befehlen auf absolut adressierte Variablen einer S7 Steuerung, wobei verschiedene S7 Systeme und Funktionalitäten [▶ 27] unterstützt werden.

3 Installation

3.1 Systemvoraussetzungen

Technische Daten	Beschreibung
Betriebssystem	Windows 10, TwinCAT/BSD
Zielplattform	PC-Architektur (x86, x64)
Minimale TwinCAT-Version	TwinCAT 3.1 Build 4024.11 und höher (Windows) TwinCAT 3.1 Build 4024.12 und höher (TwinCAT/BSD)
Erforderliches TwinCAT-Setup-Level	TwinCAT 3 XAE, XAR
Erforderliche TwinCAT-Lizenz	TF6620 TC3 S7 Communication

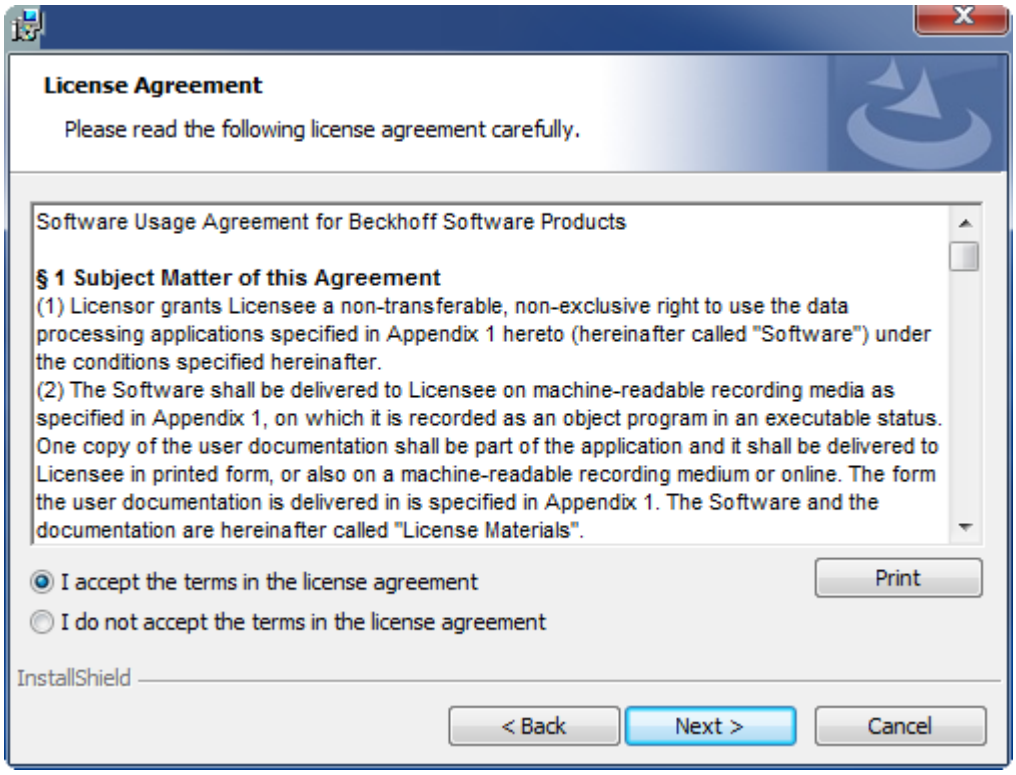
3.2 Installation

Setup-Installation (TwinCAT 3.1 Build 4024)

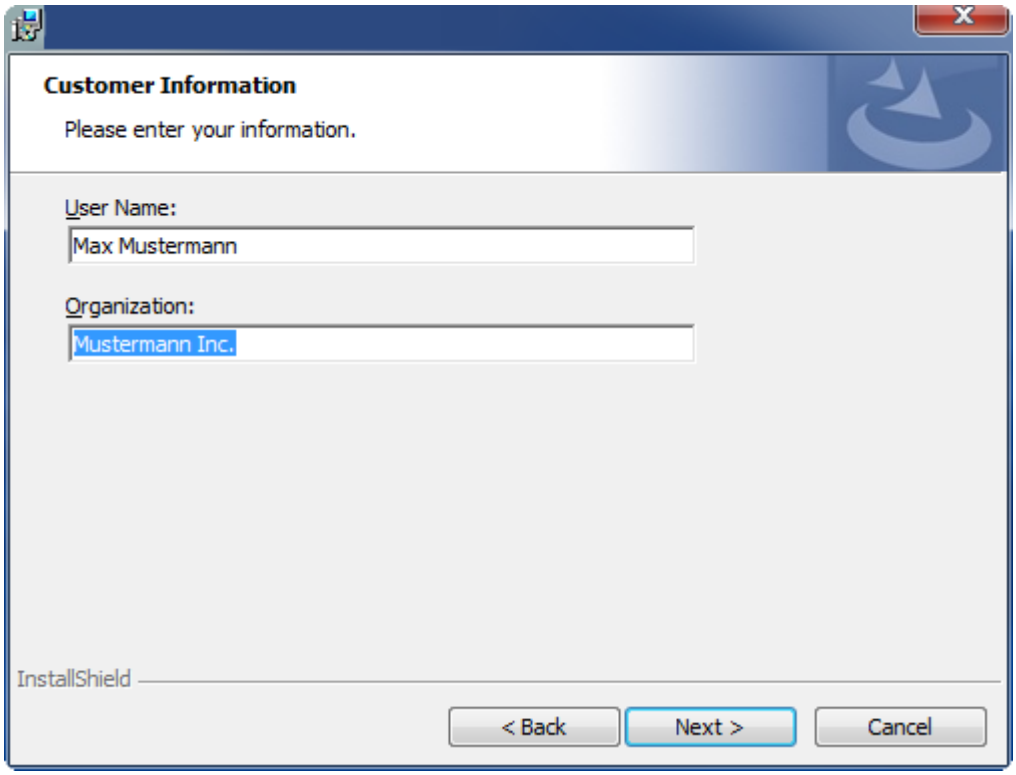
Nachfolgend wird beschrieben, wie die TwinCAT 3 Function für Windows-basierte Betriebssysteme installiert wird.

✓ Die Setup-Datei der TwinCAT 3 Function wurde von der Beckhoff-Homepage heruntergeladen.

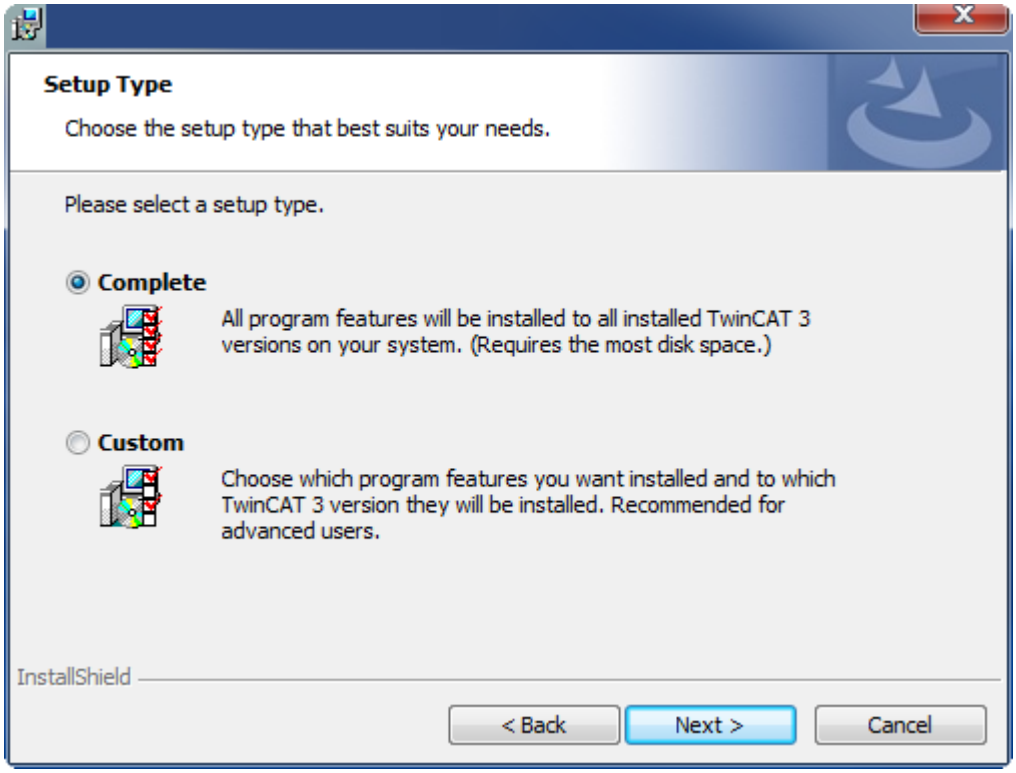
1. Führen Sie die Setup-Datei als Administrator aus. Wählen Sie dazu im Kontextmenü der Datei den Befehl **Als Administrator ausführen**.
   ⇨ Der Installationsdialog öffnet sich.
2. Akzeptieren Sie die Endbenutzerbedingungen und klicken Sie auf **Next**.



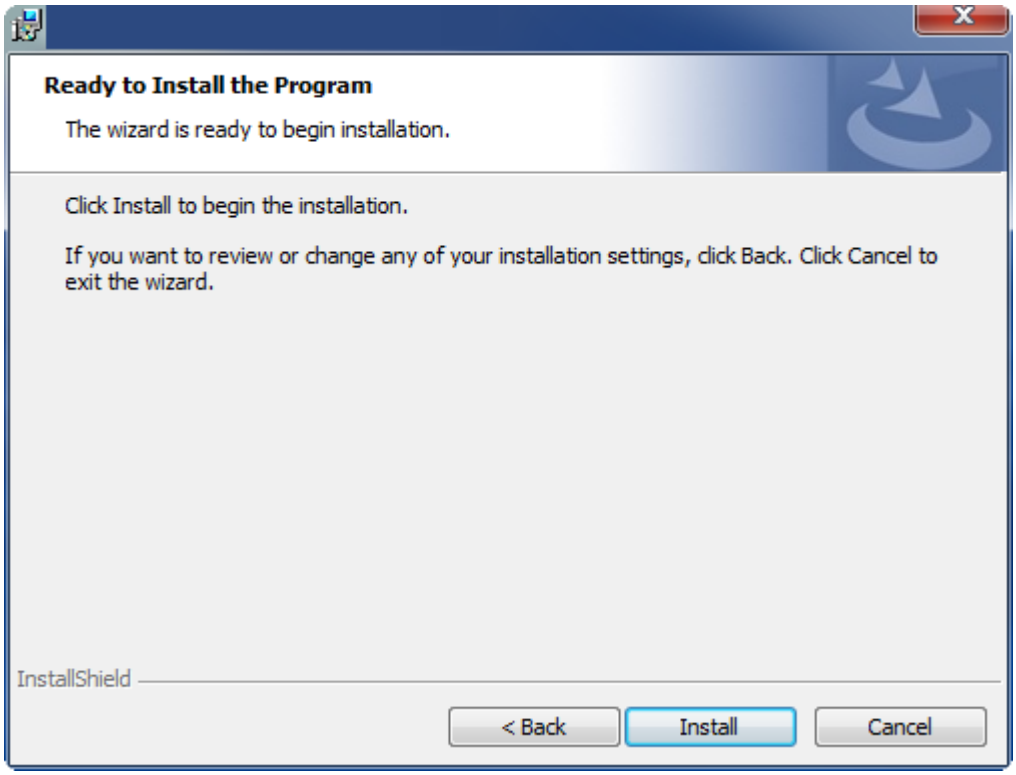
3. Geben Sie Ihre Benutzerdaten ein.



4. Wenn Sie die TwinCAT 3 Function vollständig installieren möchten, wählen Sie **Complete** als Installationstyp. Wenn Sie die Komponenten der TwinCAT 3 Function separat installieren möchten, wählen Sie **Custom**.

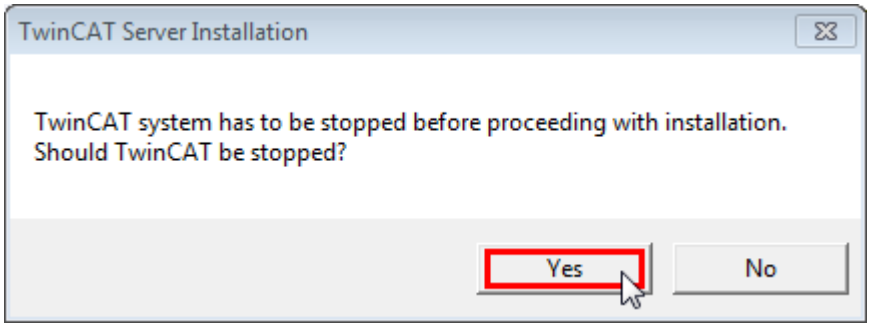


5. Wählen Sie **Next** und anschließend **Install**, um die Installation zu beginnen.

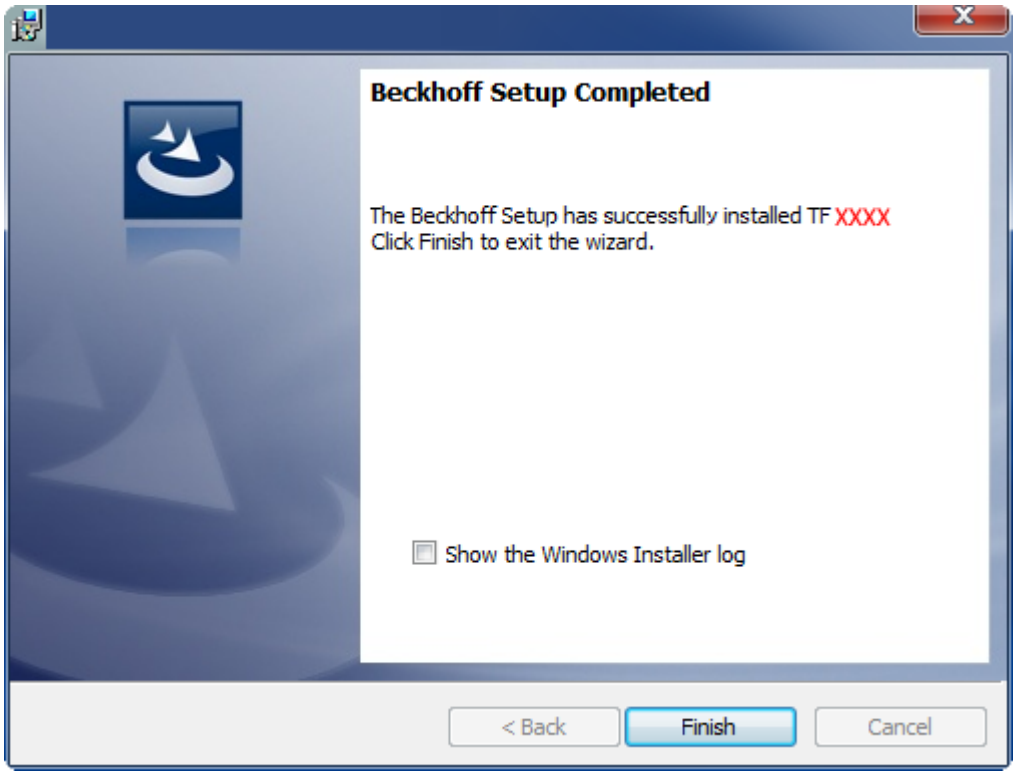


⇨ Ein Dialog weist Sie darauf hin, dass das TwinCAT-System für die weitere Installation gestoppt werden muss.

6. Bestätigen Sie den Dialog mit **Yes.**



7. Wählen Sie **Finish**, um das Setup zu beenden.



⇨ Die TwinCAT 3 Function wurde erfolgreich installiert.

3.3 Installation ab TwinCAT 4026

TwinCAT Package Manager

Wenn Sie TwinCAT 3.1 Build 4026 (und höher) auf dem Betriebssystem Microsoft Windows verwenden, können Sie diese Function über den TwinCAT Package Manager installieren, siehe Dokumentation zur Installation.

Normalerweise installieren Sie die Function über den entsprechenden Workload; dennoch können Sie die im Workload enthaltenen Pakete auch einzeln installieren. Diese Dokumentation beschreibt im Folgenden kurz den Installationsvorgang über den Workload.

Kommandozeilenprogramm TcPkg

Über das TcPkg **C**ommand **L**ine **I**nterface (CLI) können Sie sich die verfügbaren Workloads auf dem System anzeigen lassen:

```
tcpkg list TF6620
```

Über das folgende Kommando können Sie den Workload der TF6620 TC3 S7 Communication-Function installieren.

```
tcpkg install TF6620.S7Communication.XAE
tcpkg install TF6620.S7Communication.XAR
```

TwinCAT Package Manager UI

Über das **U**ser **I**nterface (UI) können Sie sich alle verfügbaren Workloads anzeigen lassen und diese bei Bedarf installieren.
Folgen Sie hierzu den entsprechenden Anweisungen in der Oberfläche.

3.4 Lizenzierung

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

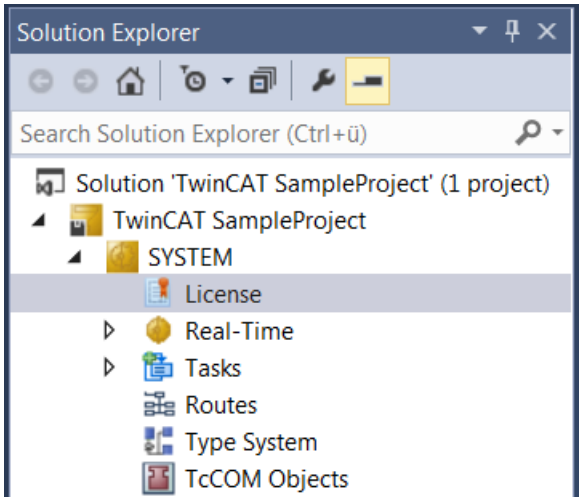
Lizenzierung der Vollversion einer TwinCAT 3 Function

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

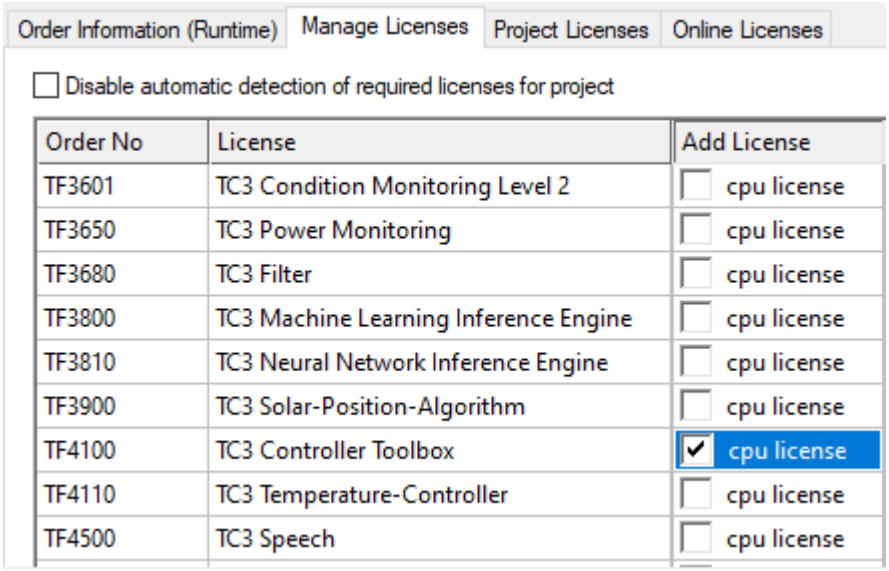
Eine 7-Tage-Testversion kann nicht für einen TwinCAT-3-Lizenz-Dongle freigeschaltet werden.

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.**

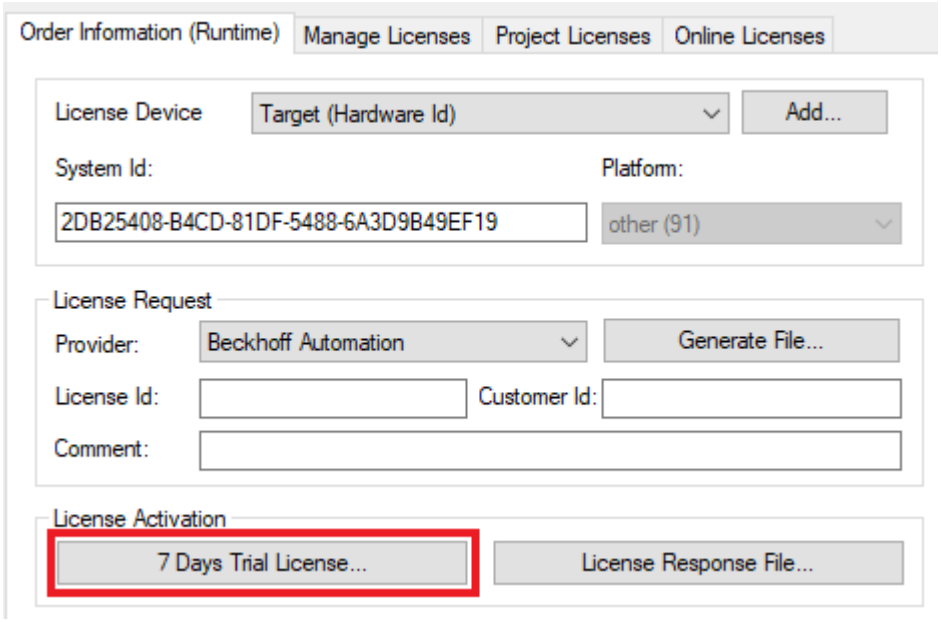


⇨ Der TwinCAT-3-Lizenzmanager öffnet sich.

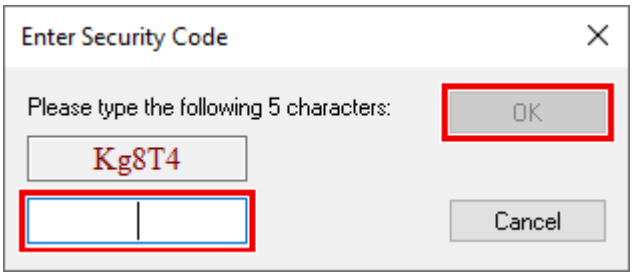
5. Öffnen Sie die Registerkarte **Manage Licenses.** Aktivieren Sie in der Spalte **Add License** das Auswahlkästchen für die Lizenz, die Sie Ihrem Projekt hinzufügen möchten (z. B. „TF4100 TC3 Controller Toolbox“).



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



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



8. Geben Sie den Code genauso ein, wie er angezeigt wird, und bestätigen Sie ihn.
9. 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.

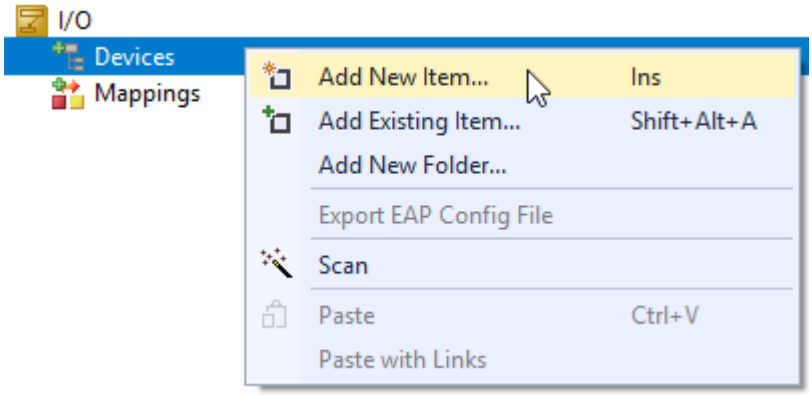
4 Technische Einführung

4.1 Getting Started

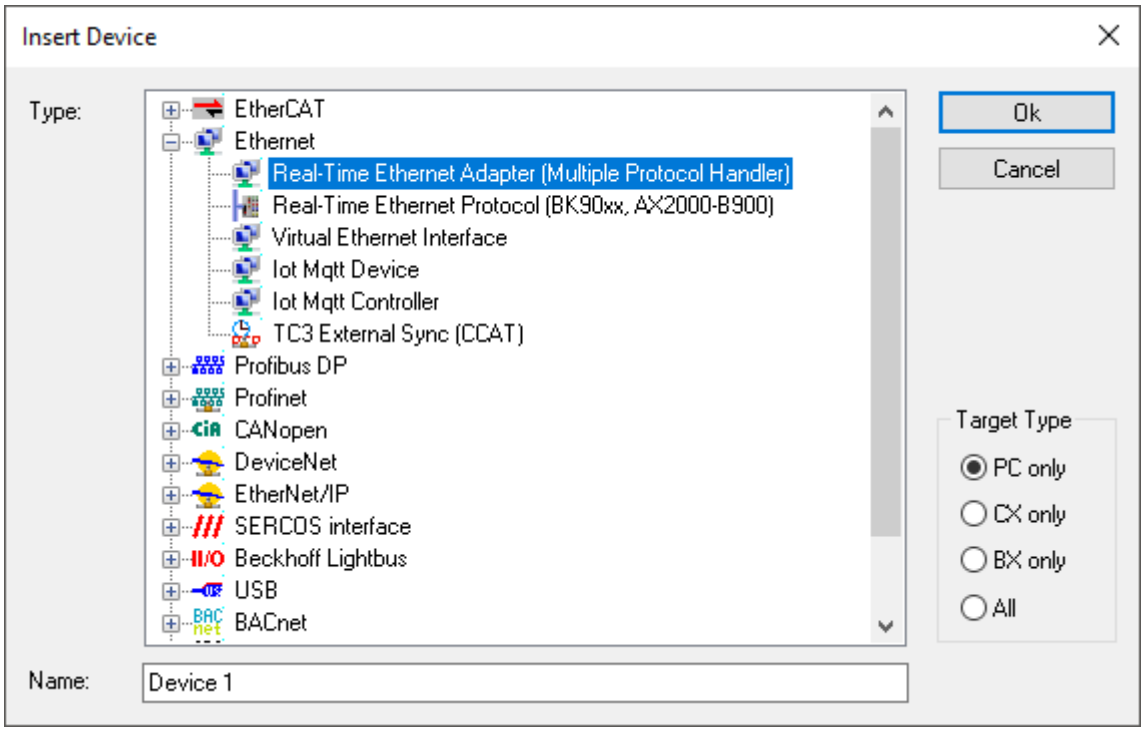
Dieser Dokumentationsartikel soll Ihnen einen ersten, schnellen Start in die Verwendung dieses Produkts ermöglichen. Nach der erfolgreichen Installation [▶ 9] und Lizenzierung [▶ 13] führen Sie die folgenden Schritte aus, um eine Verbindung zu einer S7 Steuerung herzustellen und Variablen für den Lese-/Schreibzugriff zu konfigurieren.

Hinzufügen eines S7 Communication I/O Geräts

1. Da das Produkt TwinCAT S7 Communication auf dem Realtime Ethernet Adapter basiert, fügen Sie zunächst einen Real-Time Ethernet Adapter (Multi Protocol Handler) als I/O Gerät zu Ihrer TwinCAT Konfiguration hinzu. Wählen Sie dazu **Add New Item..**

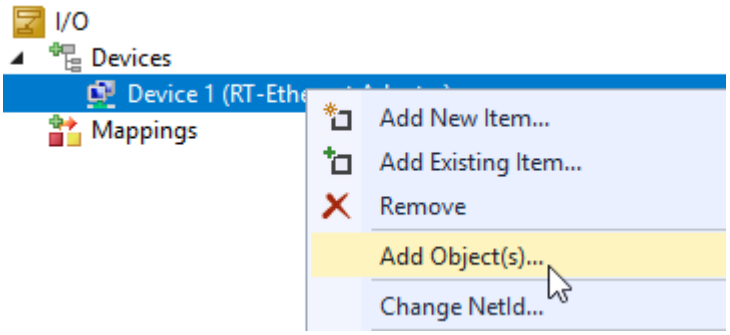


2. Im Dialog Insert Device bestätigen Sie die Auswahl **Real-Time Ethernet Adapter (Multi Protocol Handler)** mit **OK**

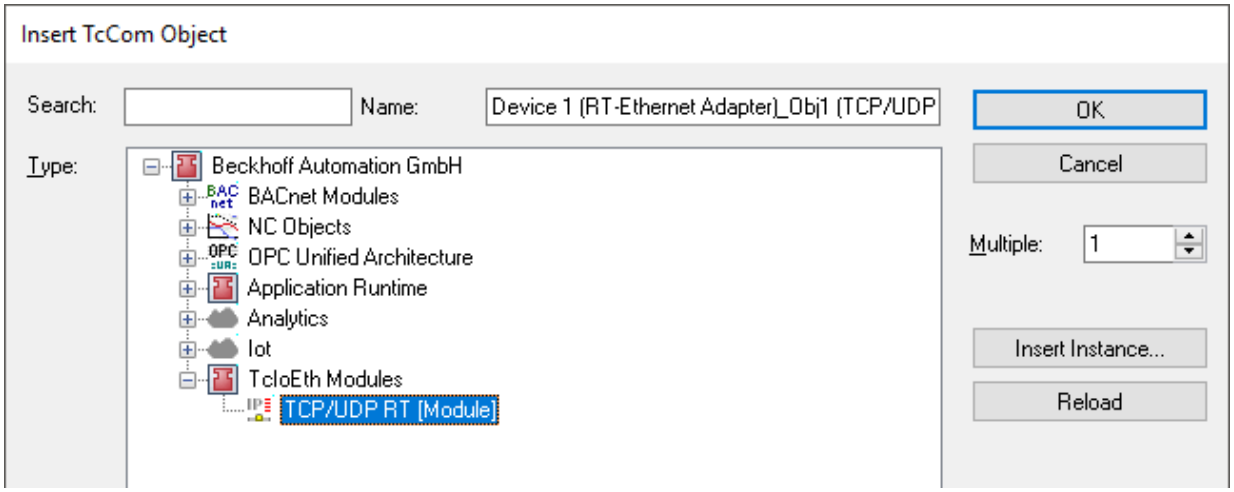


3. Anschließend verknüpfen Sie diesen Adapter mit der entsprechend dafür konfigurierten Netzwerkkarte.

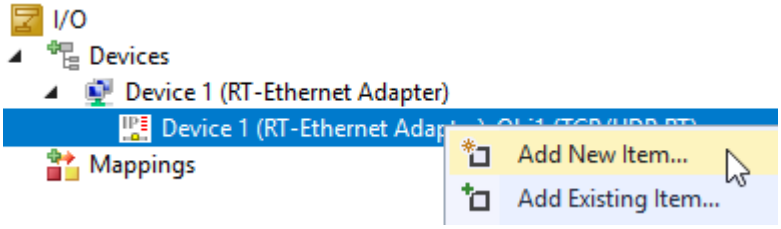
4. Im nächsten Schritt fügen Sie ein TCP/UDP RT Modul unterhalb des Real-Time Ethernet Adapters hinzu. Wählen Sie dazu **Add Object(s)..**



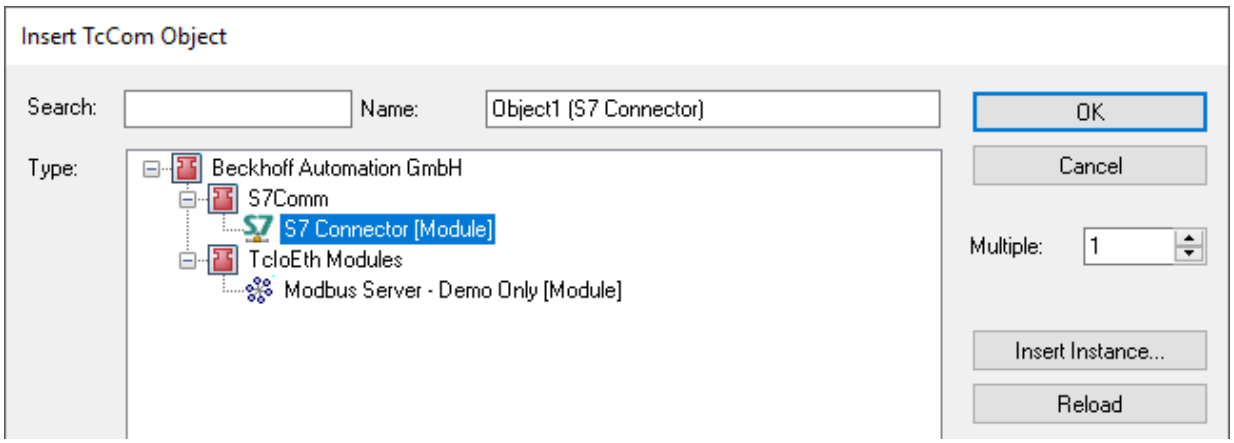
5. Bestätigen Sie die Auswahl **TCP/UDP RT Modul** mit **OK.**



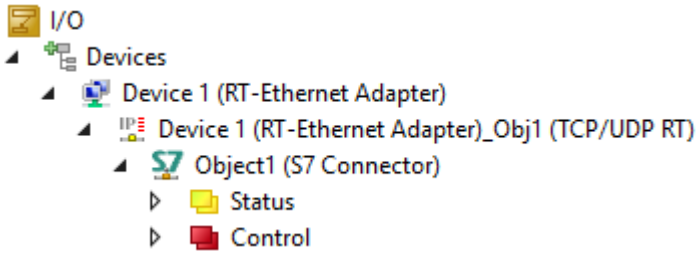
6. Danach fügen Sie einen S7 Connector zum TCP/UDP RT Modul hinzu. Es können mehrere S7 Connectoren hinzugefügt werden, beachten Sie hierzu auch die Hinweise zu eventuell vorhandenen Technischen Einschränkungen [▶ 27].
Wählen Sie hierzu wieder **Add New Item…**



7. In dem sich öffnenden Dialog fügen Sie mit **OK** das **S7 Connector (Module)** hinzu.



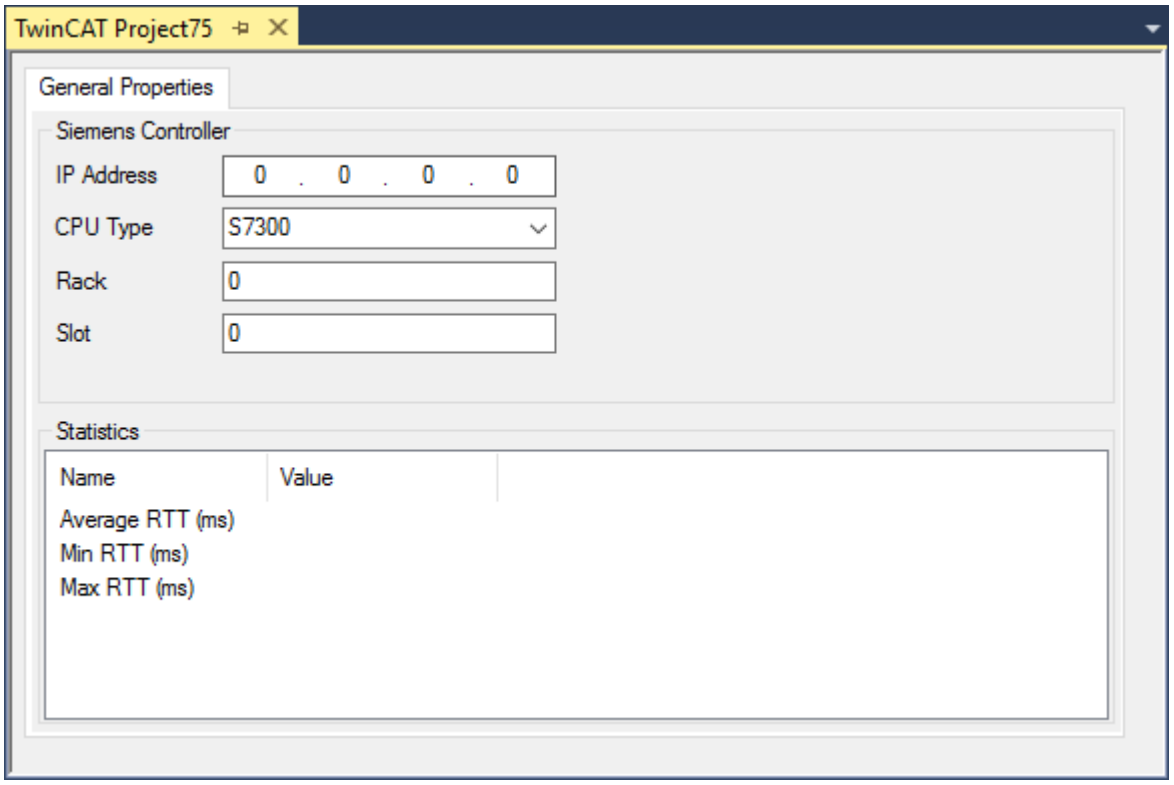
⇨ Die fertige I/O Konfiguration sollte dann wie folgt aussehen:



Konfigurieren der Verbindungsparameter

Nachdem Sie das I/O Gerät hinzugefügt haben, können Sie die Verbindungsparameter zur Siemens S7 Steuerung an dem S7 Connector definieren.

1. Führen Sie hierzu einen Doppelklick auf dem S7 Connector aus.

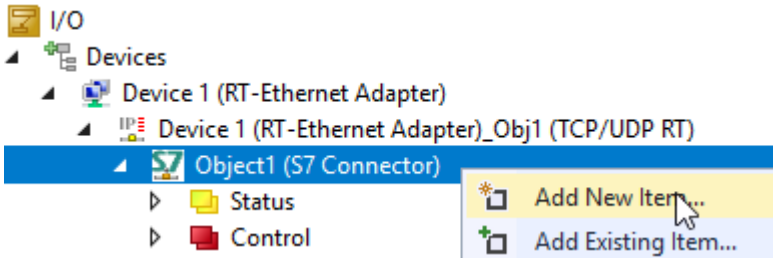


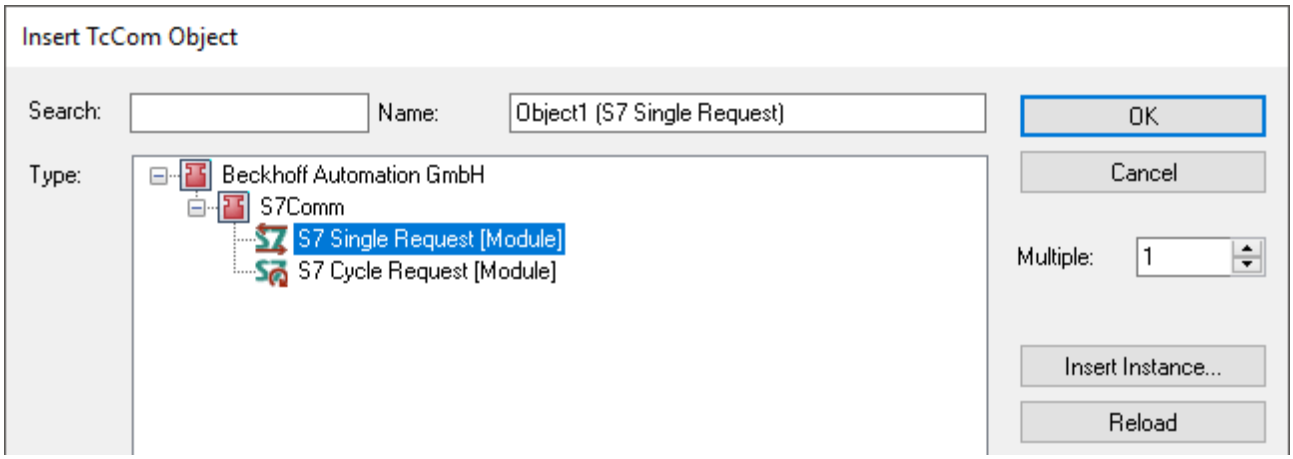
⇨ Die folgenden Verbindungsparameter zur Siemens S7 Steuerung müssen konfiguriert werden:

Parameter	Beschreibung
IP Address	IP-Adresse der Siemens S7 Steuerung
CPU Type	Art der Siemens S7 Steuerung
Rack	Rack ID, siehe S7 Geräteansicht
Slot	Slot ID, siehe S7 Geräteansicht

Zugriff auf Datenpunkte über das Prozessabbild

Im Normalfall erfolgt ein Zugriff auf Datenpunkte der S7 Steuerung über das Prozessabbild, d.h. die Datenpunkte sollen als Variablen im Prozessabbild mit anderen Variablen, z.B. aus der SPS, verknüpfbar sein. Hierzu können am S7 Connector zwei verschiedene Zugriffsarten konfiguriert werden: SingleRequest und CyclicRequest.



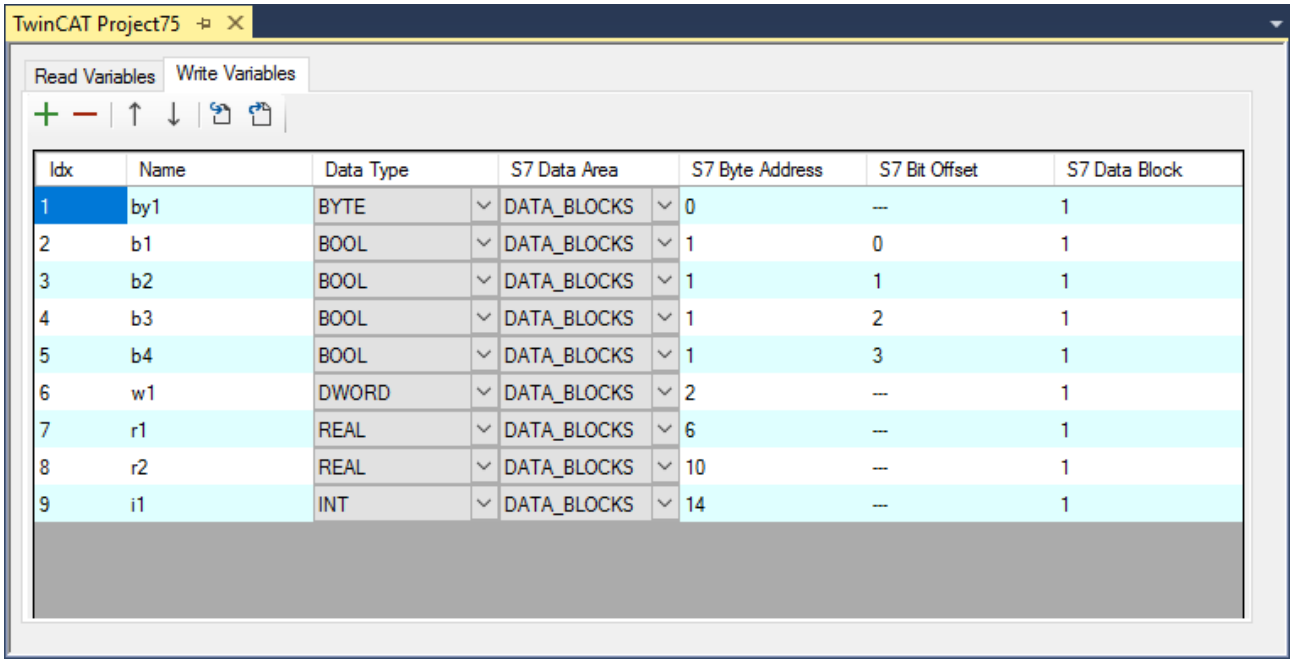


Zugriffsarten

Beim SingleRequest werden die konfigurierten Datenpunkte nur "On Demand" gelesen bzw. geschrieben. Hierfür stehen entsprechende Triggervariablen im Prozessabbild zur Verfügung. Beim CyclicRequest erfolgt ein zyklisches Lesen/Schreiben der entsprechenden Datenpunkte bei einer konfigurierbaren Zykluszeit. Beide Zugriffsarten werden in einem separaten Dokumentartikel zum Thema SingleRequest vs. CyclicRequest [▶ 22] noch einmal ausführlich beschrieben.

Konfiguration der Datenpunkte

Nach der Auswahl einer Zugriffsart können die Datenpunkte konfiguriert werden. Dies erfolgt über die entsprechenden Registerkarten **Read Variables** bzw. **Write Variables** des S7 Request Objekts.

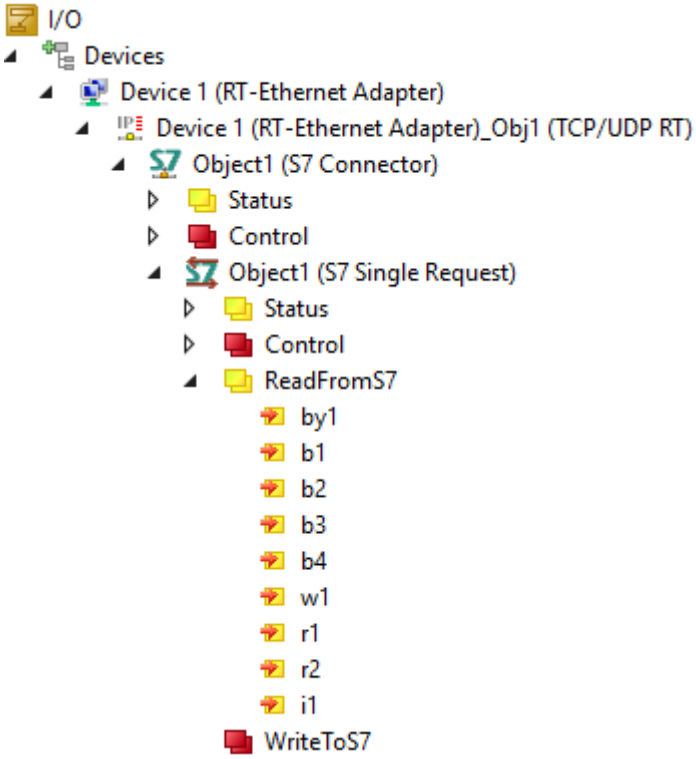


Idx	Name	Data Type	S7 Data Area	S7 Byte Address	S7 Bit Offset	S7 Data Block
1	by1	BYTE	DATA_BLOCKS	0	---	1
2	b1	BOOL	DATA_BLOCKS	1	0	1
3	b2	BOOL	DATA_BLOCKS	1	1	1
4	b3	BOOL	DATA_BLOCKS	1	2	1
5	b4	BOOL	DATA_BLOCKS	1	3	1
6	w1	DWORD	DATA_BLOCKS	2	---	1
7	r1	REAL	DATA_BLOCKS	6	---	1
8	r2	REAL	DATA_BLOCKS	10	---	1
9	i1	INT	DATA_BLOCKS	14	---	1

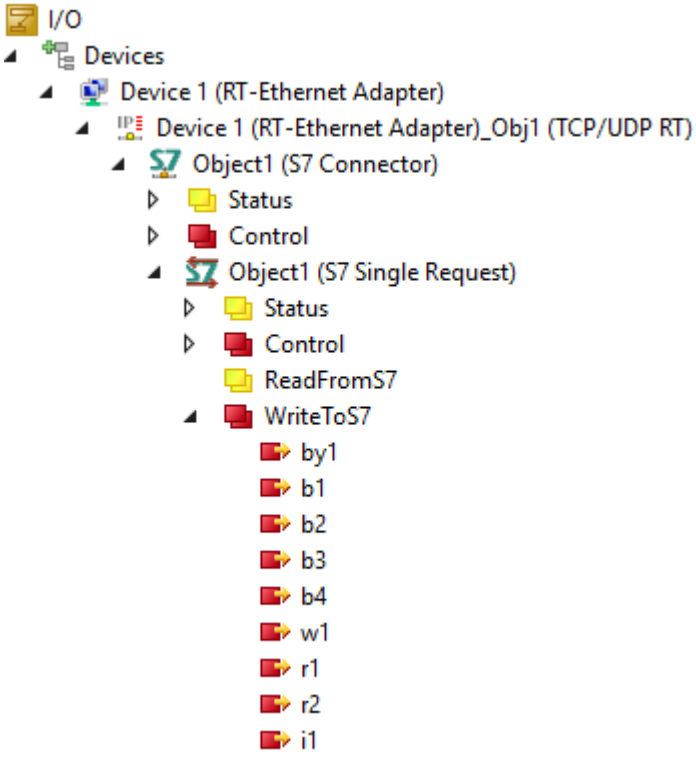
In dieser tabellarischen Übersicht lassen sich die Adressinformationen eines Datenpunkts auf der S7 Steuerung konfigurieren. Hierzu gehören: Name der Variablen (nur zur Anzeige im Prozessabbild), Datentyp, S7 Data Area, S7 Byte Address, S7 Bit Offset, S7 Data Block. Diese Informationen werden von der Siemens S7 Steuerung bereitgestellt.

Sie können die Datenpunkte auch aus einer Datei importieren oder bereits konfigurierte Datenpunkte exportieren. Dies erleichtert ggf. den Austausch dieser Informationen mit anderen Tools. Weitere Informationen finden Sie im Dokumentationsartikel Datenpunkte importieren und exportieren [▶ 27].

Die konfigurierten Datenpunkte unterhalb der Registerkarte **Read Variables** werden im Prozessabbild zum Knoten **ReadFromS7** als Eingangsvariablen hinzugefügt und lassen sich nun von dort mit anderen Variablen verknüpfen.

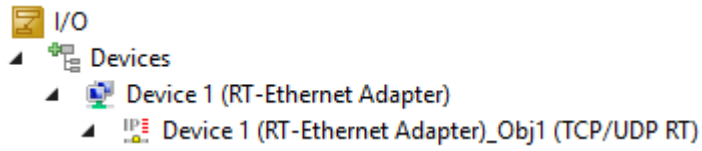


Die konfigurierten Datenpunkte unterhalb der Registerkarte **Write Variables** werden im Prozessabbild zum Knoten **WriteToS7** als Ausgangsvariablen hinzugefügt und lassen sich nun von dort mit anderen Variablen verknüpfen.



Zugriff auf Datenpunkte über die SPS

Alternativ können die Datenpunkte auch aus dem SPS-Programm heraus konfiguriert und über einen Funktionsbaustein ausgelesen bzw. geschrieben werden. Hierfür steht die SPS-Bibliothek Tc3_S7Comm [▶ 36] zur Verfügung. Im Gegensatz zur Konfiguration der Datenpunkte über das Prozessabbild muss bei dieser Variante keine Zugriffsart spezifiziert werden, da der Zugriff direkt aus der SPS Logik heraus erfolgt. Auch die Konfiguration der Verbindungsparameter erfolgt aus der SPS heraus. Daher müssen Sie für diese Variante keinen S7 Connector hinzufügen.



Über die oben bereits angesprochene SPS-Bibliothek und die darin enthaltenen Funktionsbausteine haben Sie dann die Möglichkeit diese Informationen zu konfigurieren.

Beispiel:

```
fbConnection: FB_S7CommConnection(16#01010050);
fbRequestRead: FB_S7CommSingleRequest;

fbConnection.sIpAddr := '10.3.32.101';
fbConnection.eCpuType := E_S7COMM_CPUTYPE.S71500;
fbConnection.nRack := 0;
fbConnection.nSlot := 0;

fbRequestRead.AddReadVar(ADR(data_byte), SIZEOF(data_byte), 0, E_S7COMM_DATAAREA.DATA_BLOCKS, 1);
fbRequestRead.AddReadVar(ADR(data_dword), SIZEOF(data_dword), 2, E_S7COMM_DATAAREA.DATA_BLOCKS, 1);
```

Weitere Informationen zu den beiden Kommunikationsarten finden Sie im Dokumentartikel Mapping vs. SPS Bibliothek [▶ 21], sowie in den Samples [▶ 62].

HINWEIS

Konstruktorparameter beim FB_S7CommConnection

Bitte achten Sie darauf, dass der Konstruktorparameter des Funktionsbausteins "FB_S7CommConnection" mit der Object-ID Ihres TCP/UDP Stacks konfiguriert wird. Diese finden Sie auf der Registerkarte "Objects" des TCP/UDP RT Adapters in Ihren I/O Einstellungen.

FB_S7CommConnection

Die Initialisierung des Funktionsbausteins FB_S7CommConnection erfolgt mit der ID des TCP/UDP RT Moduls. Diese kann entweder, wie in obigem Code Snippet gezeigt, statisch eingetragen oder über die Symbol Initialization in den Eigenschaften der SPS Projektinstanz konfiguriert werden. Letzteres wird in den Samples gezeigt.

4.2 Mapping vs. SPS Bibliothek

Die Kommunikation mit einer S7 Steuerung kann auf zwei Arten erfolgen. Klassischerweise kann der Zugriff auf die Steuerung und die auszulesenden bzw. zu schreibenden Datenpunkte über ein I/O Gerät konfiguriert werden. Für Betriebsumgebungen in denen ein etwas dynamischerer Zugriff erfolgen soll, steht die SPS-Bibliothek Tc3_S7Comm [▶ 36] für den Verbindungsaufbau und das Auslesen/Schreiben von Datenpunkten zur Verfügung. Die folgende Tabelle stellt beide Kommunikationsmodi gegenüber.

	Mapping	SPS
Verbindungsparameter dynamisch ändern	nein	ja
S7 Datenpunkte dynamisch hinzufügen/entfernen	nein	ja

Im Kapitel Samples [▶ 62] finden Sie Beispiele für beide Kommunikationsarten.

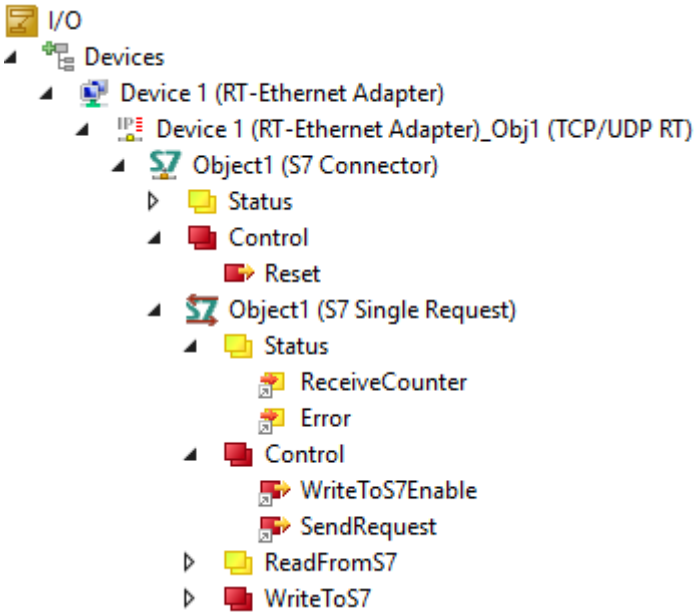
4.3 SingleRequest vs. CyclicRequest

Beim Zugriff auf Datenpunkte einer S7 Steuerung über ein I/O Mapping (vgl. Kapitel zu Mapping vs. SPS Bibliothek [▶ 21]) lassen sich verschiedene Zugriffsarten konfigurieren. Deren Funktionsweise soll im Folgenden näher beschrieben werden. Die folgende Tabelle stellt beide Zugriffsarten zunächst gegenüber.

Request	Trigger
Single	über SendRequest Variable
Cyclic	über konfigurierbare CycleTime

SingleRequest

Beim SingleRequest werden die Datenpunkte konfiguriert und zum Prozessabbild hinzugefügt. Der Lese- bzw. Schreibzugriff erfolgt jedoch "On Demand", d.h. bei Eintreten einer bestimmten Bedingung. Diese Bedingung lässt sich über die Status und Control Variablen realisieren. Ein Request wird zum Beispiel genau dann ausgeführt, wenn SendRequest um eins größer ist als ReceiveCounter. Hierbei gilt es auch den Überlauf der Variablen beim Datentyp BYTE zu beachten, d.h. 0 > 255.

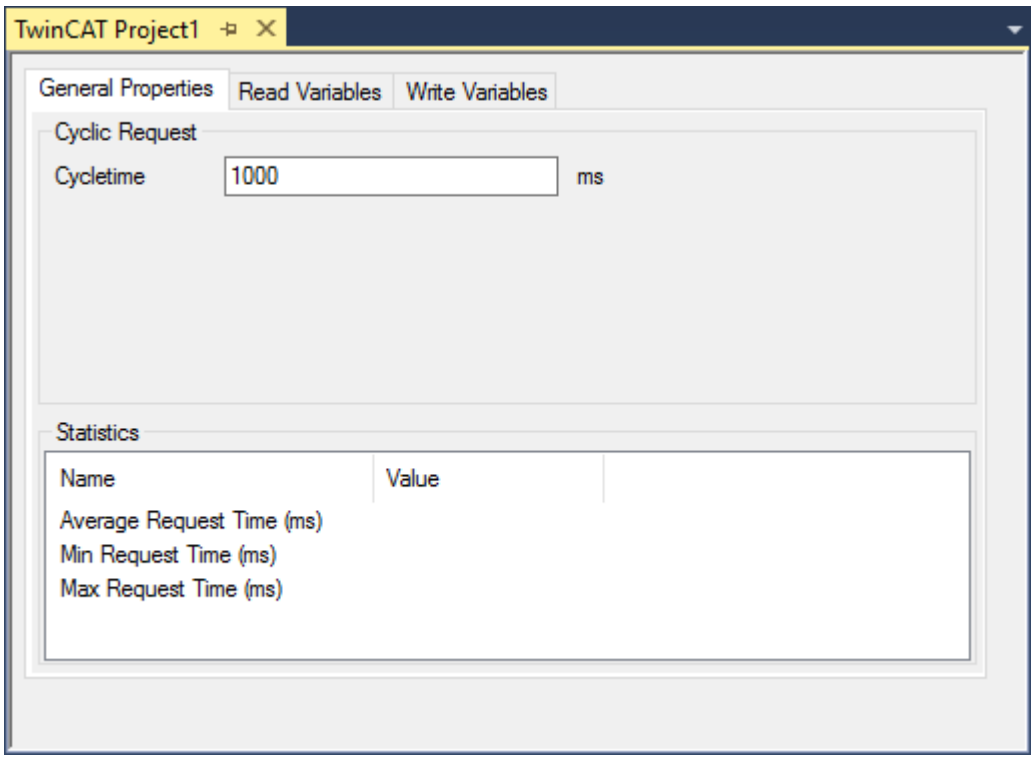


Die folgende Tabelle gibt hierzu einen Überblick:

Variable	Datentyp	Beschreibung
Control.SendRequest	BYTE	Durch Inkrementieren dieser Variablen aus dem Anwendungscode heraus wird ein Lese-/ Schreibkommando (Request) ausgelöst. Sobald die zugehörige Antwort von der S7 Steuerung empfangen wurde, wird die Eingangsvariable Status.ReceiveCounter entsprechend ebenfalls um 1 inkrementiert. So weiß die Anwendung dass die Lese/ Schreib Operation erfolgreich war.
Control.WriteToS7Enable	BIT	Schreib-Kommandos werden nur dann ausgeführt, wenn diese Variable auf TRUE gesetzt wurde.
Status.ReceiveCounter	BYTE	siehe oben Control.SendRequest
Status.Error	WORD	Bei Auftreten eines Fehlers während der Abarbeitung eines Kommandos wird hier der zugehörige Fehlercode angezeigt. Eine Beschreibung der Fehlercodes ist im Kommentarfeld hinterlegt.

CyclicRequest

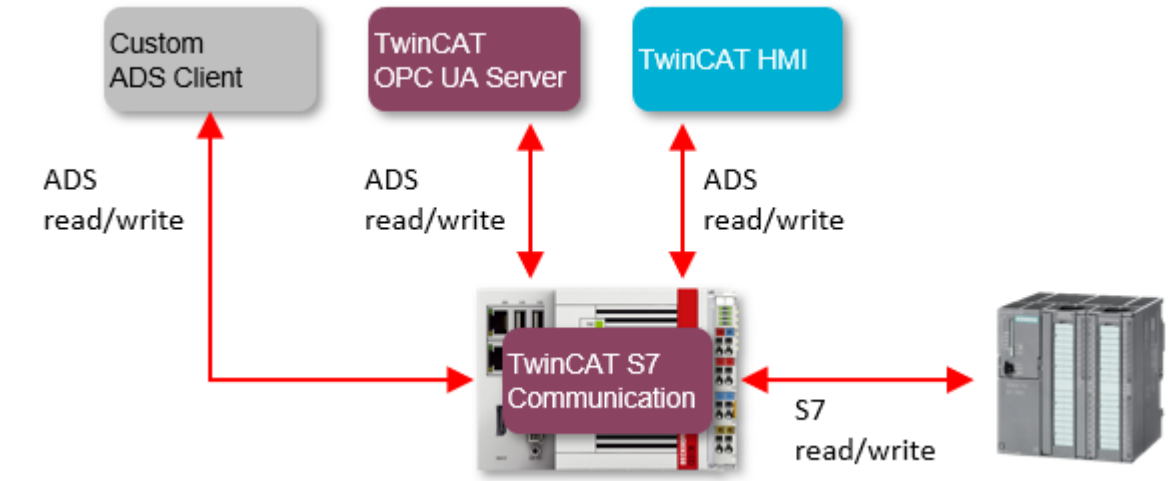
Beim CyclicRequest erfolgt eine zyklische Abarbeitung des Lese- bzw. Schreibkommandos. Die Zykluszeit lässt sich über einen Parameter konfigurieren. Wird die Zykluszeit schneller als die Echtzeit-Task des Systems eingestellt, dann läuft der Request so schnell wie möglich ab, d.h. ein neuer Request wird abgeschickt sobald der vorherige vom Remotesystem beantwortet wurde.



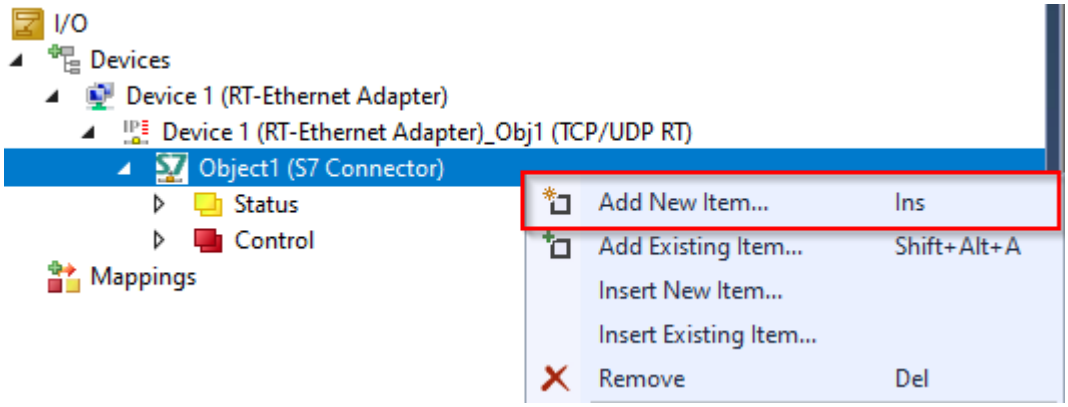
4.4 Symbolserver Schnittstelle

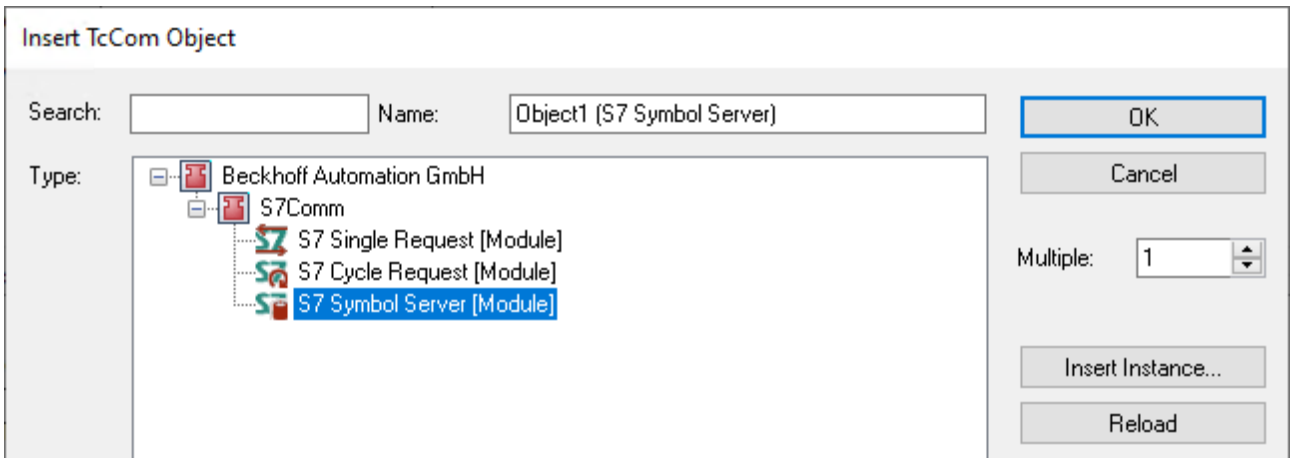
Seit der Produktversion 1.1.4 enthält die Implementierung des S7-Protokoll-Treibers eine ADS-Symbolserver-Schnittstelle, die ADS den Lese-/Schreibzugriff auf konfigurierte S7-Variablen ermöglicht. Es gibt viele verschiedene Anwendungsfälle für diese Art des Zugriffs. Solche Anwendungsfälle sind möglich, jedoch nicht beschränkt auf folgende:

- Bereitstellung des Lese-/Schreibzugriffs auf S7-Variablen für die TwinCAT HMI
- Bereitstellung des Lese-/Schreibzugriffs auf S7-Variablen für den TwinCAT OPC UA Server
- Bereitstellung des Lese-/Schreibzugriffs auf S7-Variablen für kundenspezifische ADS-Client-Anwendungen
- Durchsuchen der konfigurierten S7-Variablen mit Tools wie dem TwinCAT Target Browser
- ...

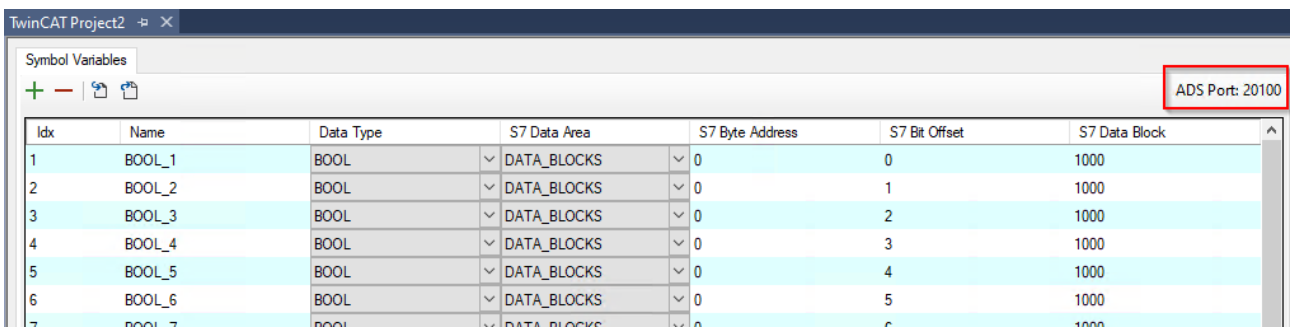


Der Symbolserver ist als separates TcCOM-Objekt verfügbar, das zu einem S7-Connector-Gerät hinzugefügt wird.



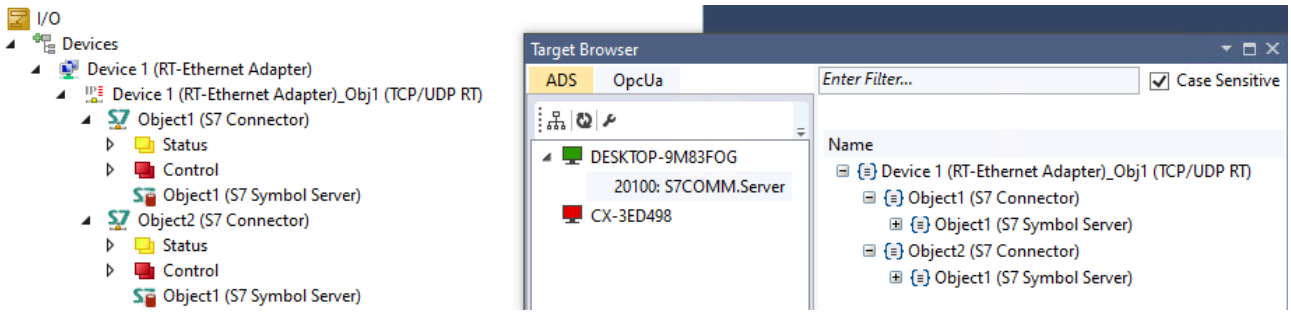


Das Symbolserver-Objekt definiert kein Prozessabbild. Stattdessen werden alle S7-Variablen im Fenster **Symbolvariablen** konfiguriert. In diesem Fenster finden Sie den Server Port des ADS-Symbolservers in der oberen rechten Ecke.



Mehrere Symbolserver

Bitte beachten Sie, dass Sie mehr als einen Symbolserver konfigurieren können, z. B., wenn Sie auf mehrere S7-Steuerungen zugreifen wollen. In diesem Fall teilen sich alle Symbolserver denselben ADS-Serverport. Die einzelnen Symbolserver werden dann unter diesem ADS-Serverport einsortiert. Der folgende Screenshot zeigt ein Beispiel für eine solche Konfiguration – zwei S7-Connectors mit jeweils einem eigenen Symbolserver-Objekt. Der TwinCAT Target Browser verbindet sich mit dem ADS-Serverport und zeigt den Namensraum des Symbolservers an.



In der folgenden Tabelle sind alle derzeit an der Symbolserver-Schnittstelle verfügbaren ADS-Befehle aufgeführt.

Befehl	Beschreibung
AdsGetHandle	Beziehen eines ADS-Handles über den Symbolnamen der Variablen.
AdsReleaseHandle	Freigeben des ADS-Handles.
AdsReadByHandle	Leseoperationen auf dem bezogenen Handle. Unterstützt auch Summenkommandos.
AdsWriteByHandle	Schreiboperationen auf dem bezogenen Handle. Unterstützt auch Summenkommandos.
AdsRead	Leseoperationen über direkte Kommunikation mit IndexGroup/IndexOffset. Unterstützt auch Summenkommandos.
AdsWrite	Schreiboperationen über direkte Kommunikation mit IndexGroup/ IndexOffset. Unterstützt auch Summenkommandos.

Bitte beachten Sie, dass ADS Notifications vom Symbolserver derzeit nicht unterstützt werden.

Beispiel: Verbindung zum Symbolserver mit TwinCAT OPC UA Server

Im vorigen Kapitel wurde das Aktivieren der Symbolserver-Schnittstelle auf einem S7-Connector-Gerät und das Browsen mit dem TwinCAT Target Browser durch dessen Namensraum erläutert. Als weiteres Beispiel wollen wir nun den TwinCAT OPC UA Server so konfigurieren, dass er auf die Symbolserver-Schnittstelle zugreift und die konfigurierten S7-Variablen über seinen OPC-UA-Server-Adressraum zur Verfügung stellt.

Nachdem der TwinCAT OPC UA Server installiert wurde, öffnen Sie seine Data-Access-Konfiguration (TcUaDaConfig.xml), um die Verbindungsdetails des Symbolservers zu konfigurieren. Sie können die Data-Access-Konfiguration entweder mit dem TwinCAT OPC UA Configurator oder mit einem beliebigen Texteditor bearbeiten.

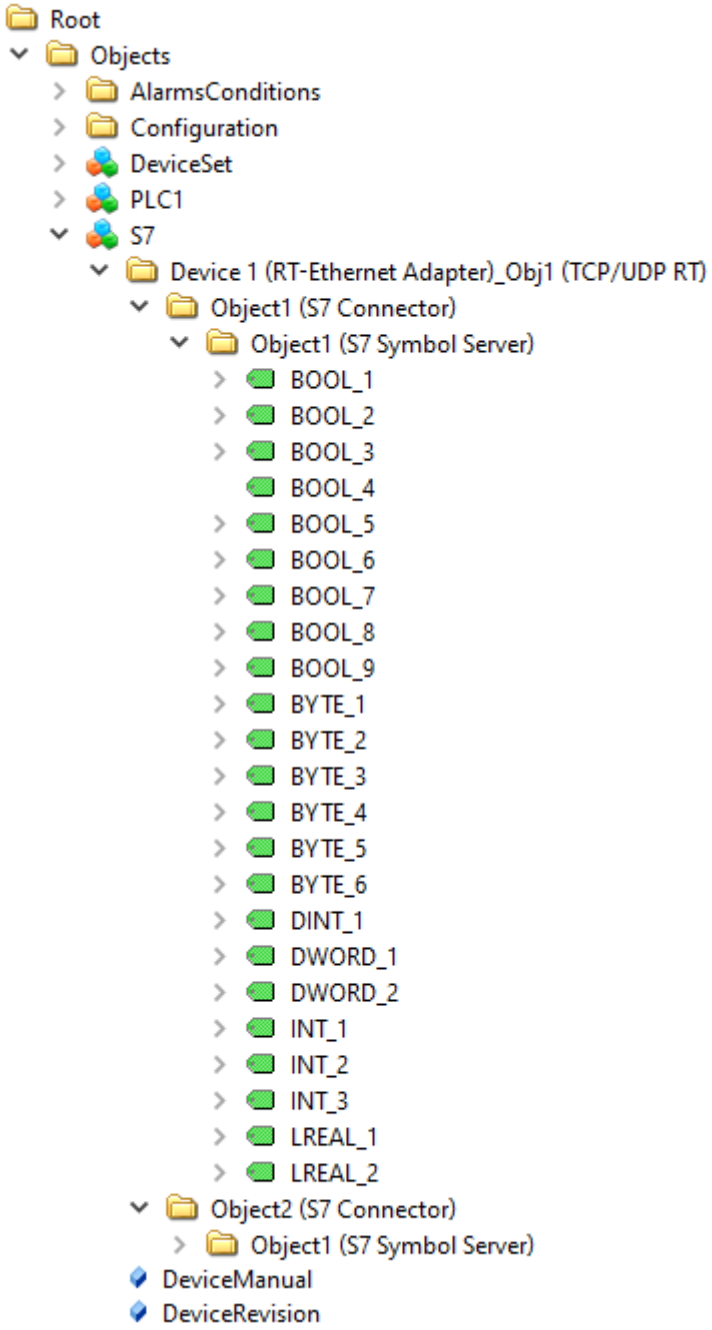
Fügen Sie ein neues Datenzugriffgerät mit den folgenden erforderlichen Parametern hinzu:

Parameter	Beschreibung
Name	Eindeutiger Name für das Gerät. Wird als Einstiegspunkt in den Adressraum des OPC UA Server verwendet.
AdsPort	ADS-Serverport des Symbolservers, z. B. 20100.
AdsNetId	NetID des Systems, auf dem das Produkt TwinCAT S7 Communication läuft.
AutoCfg	Wert 5: Schaltet den TwinCAT OPC UA Server in den Modus „Symbol Server Access“.

Der folgende Auszug aus TcUaDaConfig.xml zeigt ein Beispiel für eine solche Konfiguration.

```
<UaNodeManager>
  <Name>S7</Name>
  <AdsPort>20100</AdsPort>
  <AdsNetId>127.0.0.1.1.1</AdsNetId>
  <AdsTimeout>2000</AdsTimeout>
  <AdsTimeSuspend>20000</AdsTimeSuspend>
  <AutoCfg>5</AutoCfg>
  <AutoCfgSymFile></AutoCfgSymFile>
  <IoMode>1</IoMode>
  <MaxGetHandle>100</MaxGetHandle>
  <ReleaseAdsVarHandles>1</ReleaseAdsVarHandles>
  <Disabled>0</Disabled>
</UaNodeManager>
```

Sobald diese Konfiguration aktiviert ist, verbindet sich der TwinCAT OPC UA Server mit dem Symbolserver und importiert dessen Namensraum. Ein OPC UA Client kann sich dann mit dem Server verbinden und auf die Variablen zugreifen.

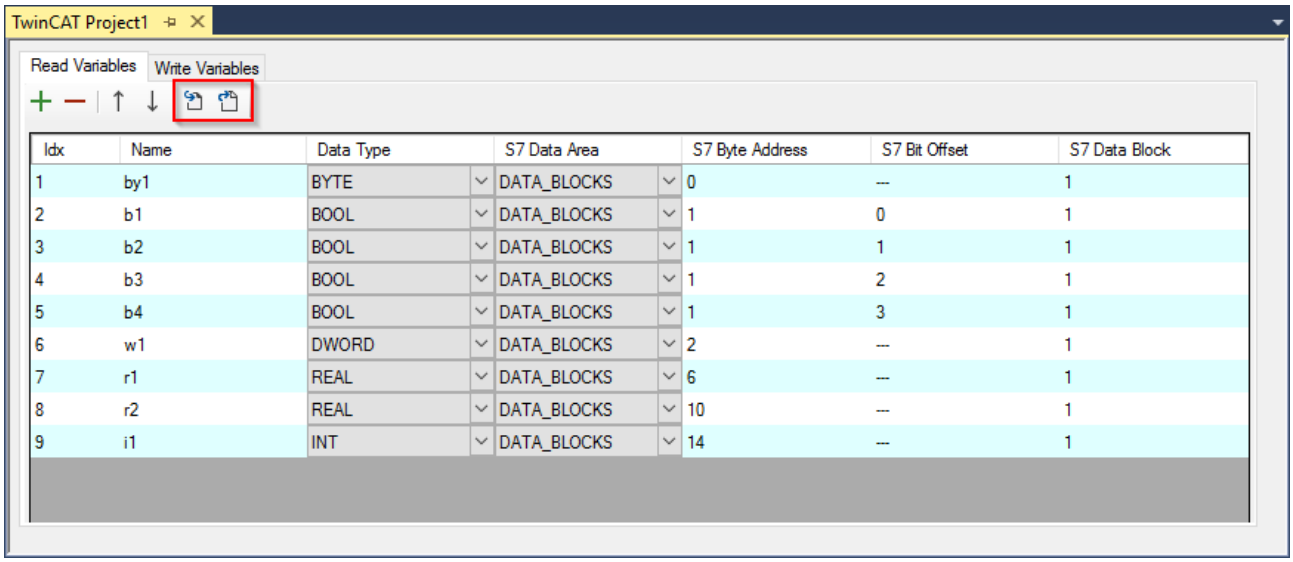


Beispiel: Verbindung zum Symbolserver mit benutzerdefiniertem ADS-Client

Unser Repository mit Beispielen [▶ 62] auf GitHub enthält ein .NET Core Projekt, das die verschiedenen Möglichkeiten für den Zugriff des ADS-Clients auf den Symbolserver zeigt.

4.5 Datenpunkte importieren und exportieren

Datenpunkte aus einer S7 Steuerung lassen sich über einen Import/Export Mechanismus mit anderen Systemen austauschen bzw. von dort importieren. Die entsprechende Funktion steht in den Registerkarten **Read/Write Variables** vom S7 Request Objekt im Prozessabbild zur Verfügung.



Idx	Name	Data Type	S7 Data Area	S7 Byte Address	S7 Bit Offset	S7 Data Block
1	by1	BYTE	DATA_BLOCKS	0	---	1
2	b1	BOOL	DATA_BLOCKS	1	0	1
3	b2	BOOL	DATA_BLOCKS	1	1	1
4	b3	BOOL	DATA_BLOCKS	1	2	1
5	b4	BOOL	DATA_BLOCKS	1	3	1
6	w1	DWORD	DATA_BLOCKS	2	---	1
7	r1	REAL	DATA_BLOCKS	6	---	1
8	r2	REAL	DATA_BLOCKS	10	---	1
9	i1	INT	DATA_BLOCKS	14	---	1

Als Datenaustauschformat wird eine Semikolon-separierte Liste verwendet. Nachfolgend ein Beispiel, passend zu obigem Screenshot:

```
by1;BYTE;DATA_BLOCKS;0;0;1
b1;BOOL;DATA_BLOCKS;1;0;1
b2;BOOL;DATA_BLOCKS;1;1;1
b3;BOOL;DATA_BLOCKS;1;2;1
b4;BOOL;DATA_BLOCKS;1;3;1
w1;DWORD;DATA_BLOCKS;2;0;1
r1;REAL;DATA_BLOCKS;6;0;1
r2;REAL;DATA_BLOCKS;10;0;1
i1;INT;DATA_BLOCKS;14;0;1
```

4.6 Unterstützte Systeme und Funktionalitäten

Die folgenden Siemens S7 Steuerungen wurden getestet und werden unterstützt:

- Siemens S7-300
- Siemens S7-400
- Siemens S7-1200
- Siemens S7-1500

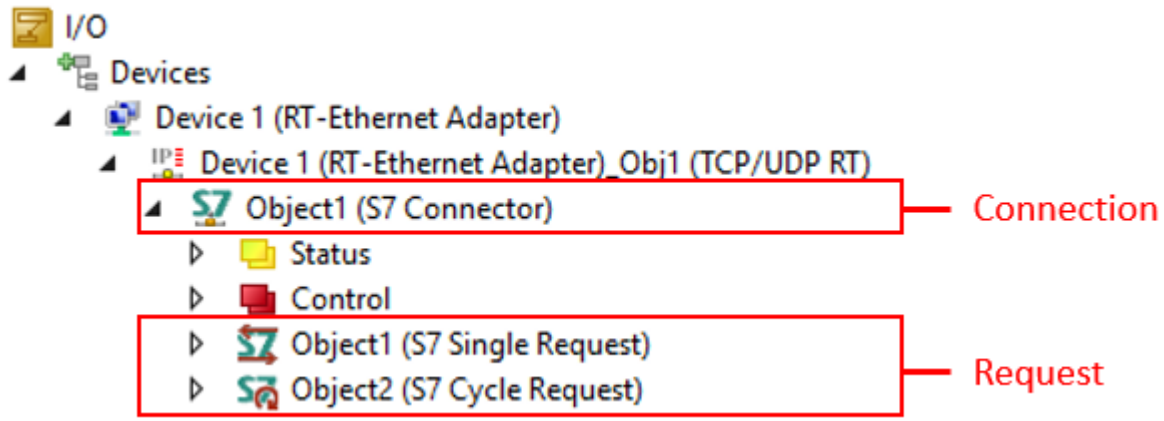
Bei der Kommunikation mit diesen Steuerungssystemen werden die folgenden Siemens Data Areas unterstützt:

- INPUT
- OUTPUT
- DATA_BLOCKS
- MERKER ("FLAGS")

Bitte beachten Sie, dass Siemens Kommunikationsprozessoren („Communication Processors“) nicht explizit getestet wurden.

4.7 Technische Einschränkungen

Bei der Kommunikation mit einer der unterstützten Siemens S7 Steuerungen [▶ 27] gelten die folgenden technischen Einschränkungen:

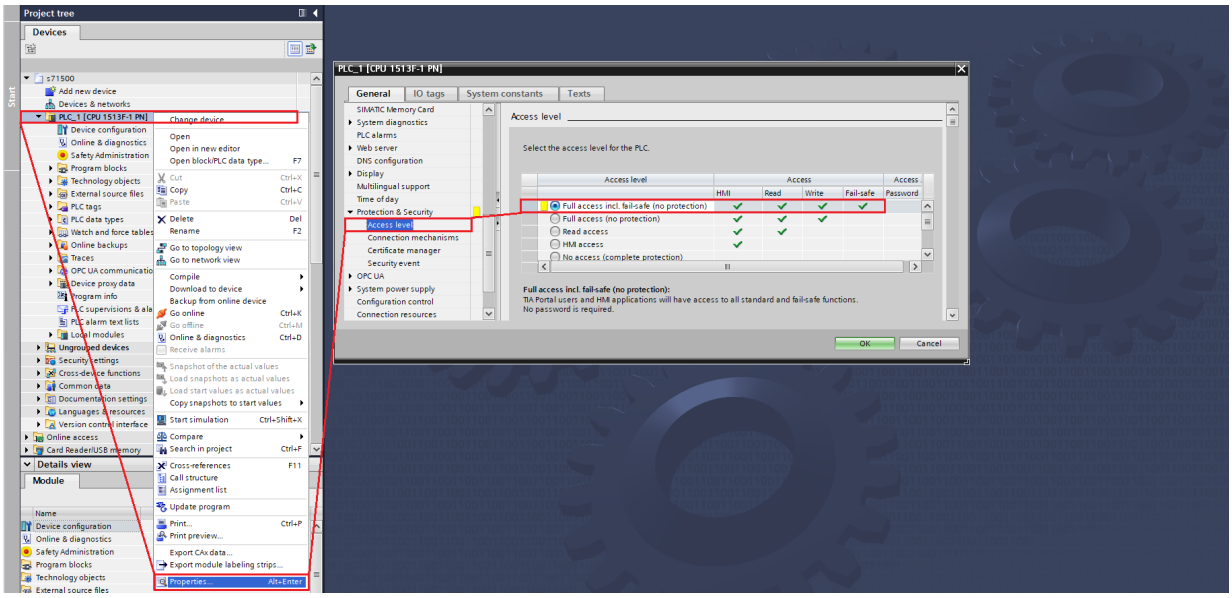


Einschränkung	Beschreibung
Aktivierung/Freischaltung der Kommunikation	Bei neueren Siemens S7 Steuerungen muss die Kommunikation explizit aktiviert bzw. freigeschaltet werden. Bitte konsultieren Sie das Handbuch zu Ihrer Siemens S7 Steuerung für weitere Informationen zur Freischaltung der TCP/IP Kommunikation mit der Steuerung. Das Kapitel Aktivierung des S7 Protokollzugriffs [▶ 28] zeigt Ihnen einige beispielhafte Screenshots aus dem TIA Portal.
Nur Kommunikation mit absolut adressierten Variablen	Bei der Kommunikation mit Variablen aus der jeweiligen Siemens Data Area können nur absolut adressierte Variablen verwendet werden. Ein symbolischer Zugriff ist nicht möglich. Daher ist Vorsicht geboten, falls sich das Siemens Steuerungsprogramm ändert und sich hierdurch ggf. Speicheradressen verändern.
Maximale Framegröße pro Request	Siemens S7 Steuerungen haben eine maximale Framegröße pro Request. Diese wird beim Hochfahren der Kommunikation von der Steuerung gemeldet und beträgt bei den meisten S7-Typen circa 960 Bytes. Falls mehr als 960 Bytes im Prozessabbild angelegt werden, splittet der TwinCAT S7 Kommmunikationstreiber die Requests automatisch auf mehrere Anfragen auf.
Maximale Verbindungsanzahl	Die Function TF6620 basiert auf dem TwinCAT TCP/UDP RT Stack, welcher standardmäßig maximal 32 Verbindungen (pro Stack) erlaubt. Zusätzlich können auch Limitierungen auf Seiten der Siemens S7 Steuerung bzgl. der gleichzeitigen Anzahl an TCP/IP Verbindungen existieren.
Maximale Anzahl Requests pro Connection	Es können maximal 64 Requests pro Connection konfiguriert werden. Durch zusätzliche Connections lässt sich diese Anzahl jedoch erhöhen.
Maximale Anzahl Variablen pro Request	Es können maximal 255 Variablen pro Request verwendet werden. Durch zusätzliche Requests lässt sich die Anzahl jedoch erhöhen.

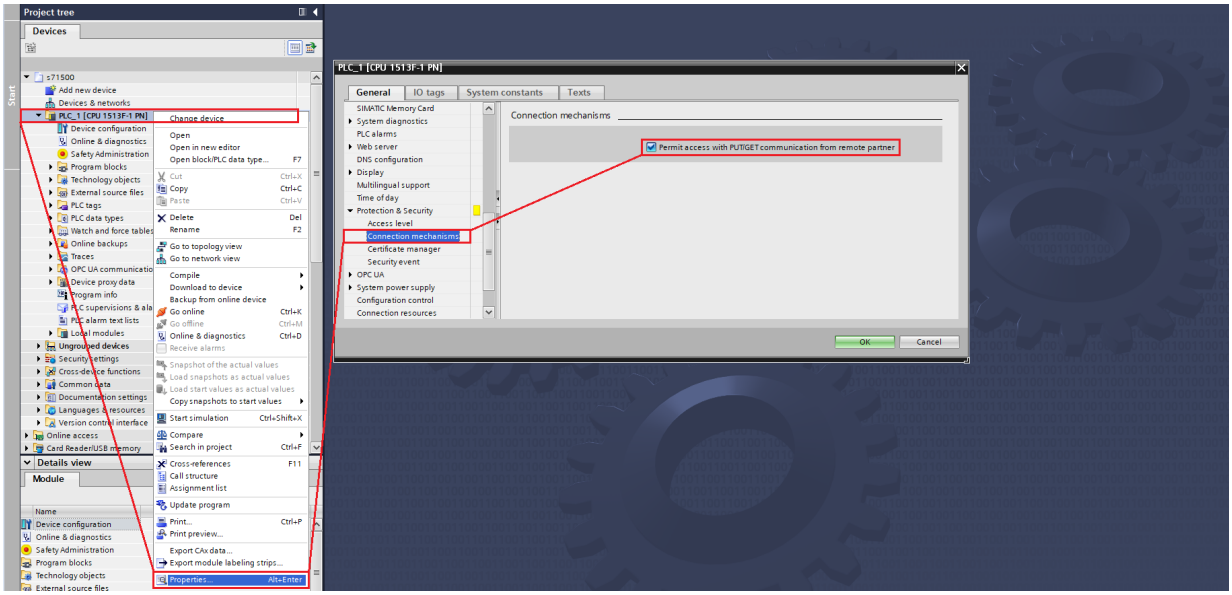
4.8 Aktivierung des S7 Protokollzugriffs

Die folgenden Screenshots zeigen eine beispielhafte Aktivierung der S7 Protokollfunktionen im TIA-Portal, welche üblicherweise nur bei S7-1200 und S7-1500 Steuerungen notwendig ist. Bitte beachten Sie, dass die Screenshots von Ihrer Betriebsumgebung abweichen und sich je nach TIA Version unterschiedlich darstellen können.

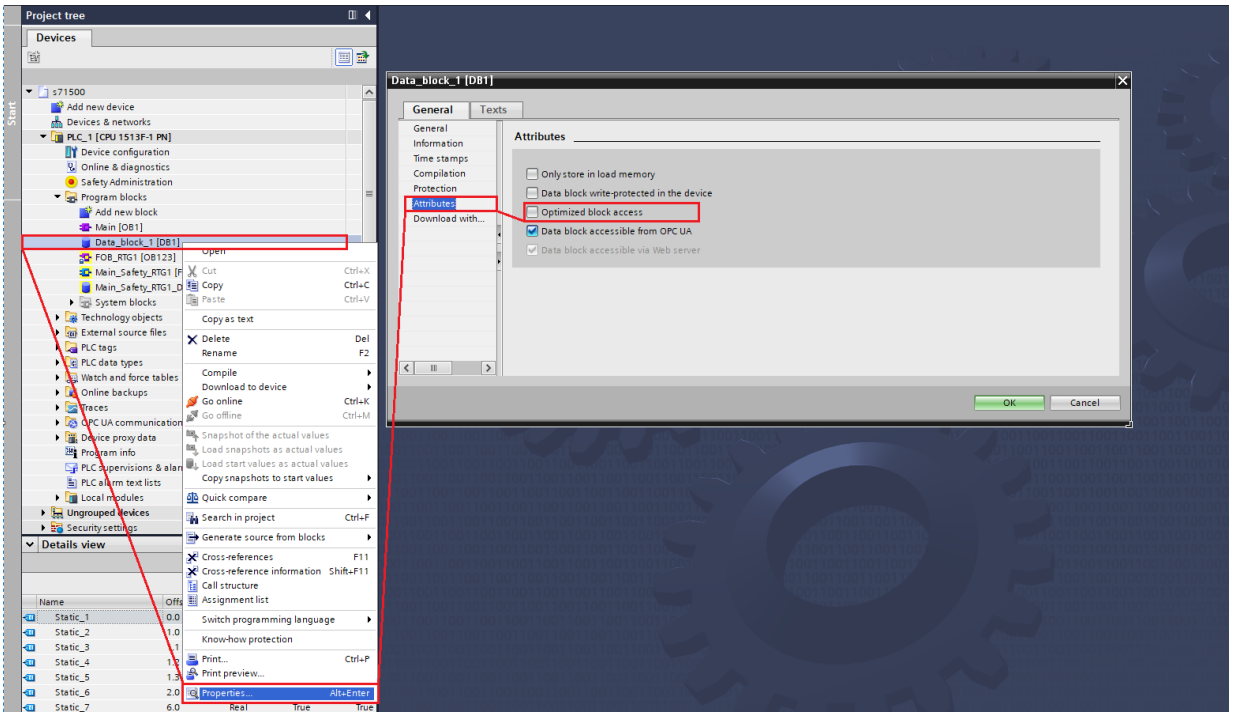
1. Aktivieren Sie zunächst den Zugriff über die Access Level.



2. Danach aktivieren Sie den COTP PUT/GET Zugriff.



3. Deaktivieren Sie den optimierten Blockzugriff.



HINWEIS

Hardware Download

Bitte beachten Sie, dass Sie nach einer Änderung der Zugriffsparameter einen Hardware-Download durchführen müssen damit diese Änderungen aktiv werden.

4.9 Optimierungsmöglichkeiten

Manchmal ist der offensichtliche Weg nicht immer auch der effizienteste Weg. Der folgende Artikel zeigt einige Möglichkeiten auf, wie Sie die S7 Datenkommunikation optimieren können.

Beispiel 1: Auslesen von vielen BOOL-/BIT-Variablen

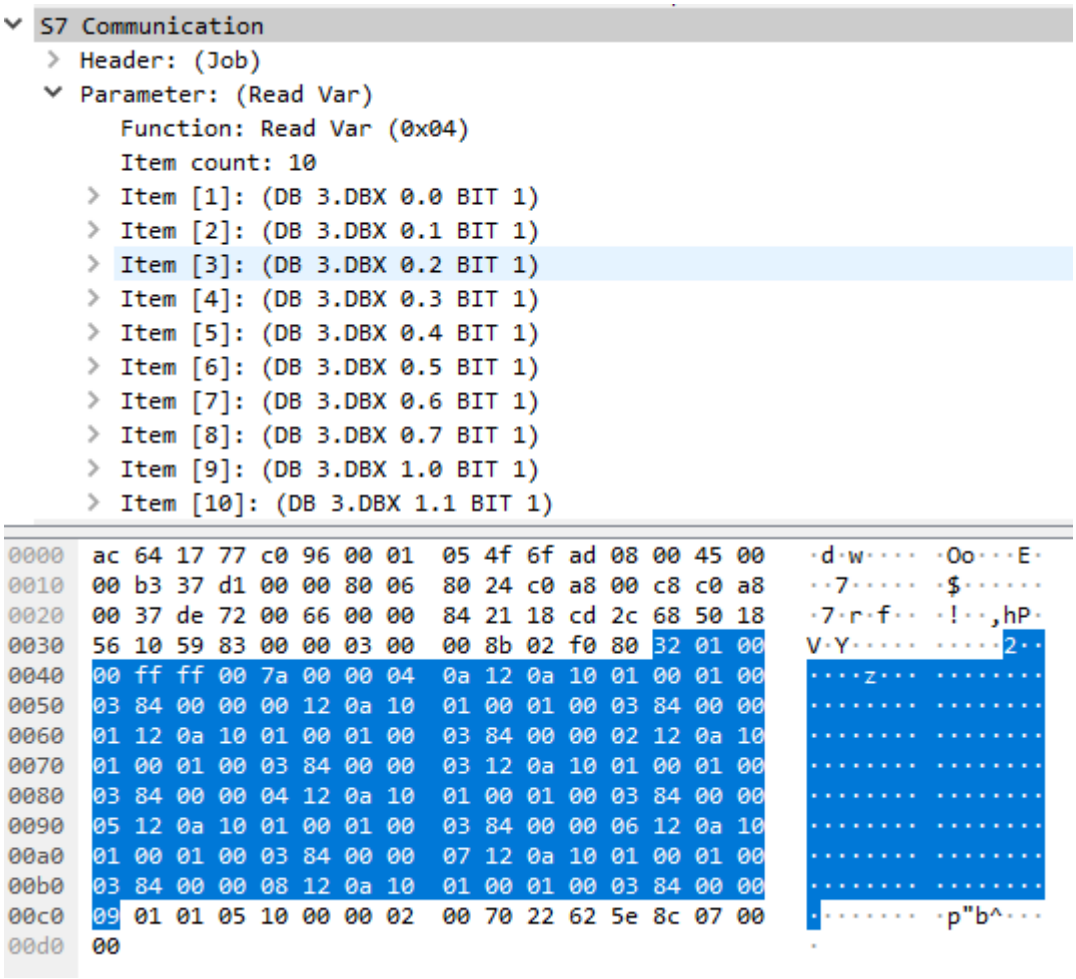
Die Konfiguration im TIA-Projekt enthält mehrere (in diesem Beispiel 10) BOOL-/BIT-Variablen, welche sich im Speicher direkt „nacheinander“ befinden und ausgelesen werden sollen:

	Name	Data type	Offset	Start value	Re
1	Static				
2	bool1	Bool	0.0	1	
3	bool2	Bool	0.1	false	
4	bool3	Bool	0.2	1	
5	bool4	Bool	0.3	false	
6	bool5	Bool	0.4	1	
7	bool6	Bool	0.5	false	
8	bool7	Bool	0.6	1	
9	bool8	Bool	0.7	1	
10	bool9	Bool	1.0	false	
11	bool10	Bool	1.1	1	

Die offensichtlichste Implementierung wäre es, die 10 Variablen vom Typ BOOL an einen Request zu hängen.

```
fbReq.AddReadBit(ADR(bBool_1 ), 0, 0, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_2 ), 0, 1, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_3 ), 0, 2, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_4 ), 0, 3, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_5 ), 0, 4, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_6 ), 0, 5, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_7 ), 0, 6, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_8 ), 0, 7, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_9 ), 1, 0, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
fbReq.AddReadBit(ADR(bBool_10), 1, 1, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
```

Auf dem Draht würde diese Implementierung dann wie folgt aussehen:



Die verschiedenen Variablen werden hierbei einzeln angefragt, wobei jeder „Item“-Eintrag 12 Bytes lang ist. Der gesamte Payload des S7 Request ist dadurch 132 Bytes lang. Eine entsprechende Response würde dann wie folgt aussehen:

```
S7 Communication
  Header: (Ack_Data)
  Parameter: (Read Var)
      Function: Read Var (0x04)
      Item count: 10
  Data
    Item [1]: (Success)
    Item [2]: (Success)
    Item [3]: (Success)
    Item [4]: (Success)
    Item [5]: (Success)
    Item [6]: (Success)
    Item [7]: (Success)
    Item [8]: (Success)
    Item [9]: (Success)
    Item [10]: (Success)
EtherCAT Switch Link
```

```
0000  00 01 05 4f 6f ad ac 64  17 77 c0 96 08 00 45 00   ···Oo··d ·w····E·
0010  00 78 8e 6e 40 00 40 06  29 c2 c0 a8 00 37 c0 a8   ·x·n@·@· )····7··
0020  00 c8 00 66 de 72 18 cd  2c 68 00 00 84 ac 50 18   ···f·r·· ,h····P·
0030  20 00 61 ff 00 00 03 00  00 50 02 f0 80 32 03 00    ·a····· ·P···2··
0040  00 ff ff 00 02 00 3b 00  00 04 0a ff 03 00 01 01   ·····;· ········
0050  00 ff 03 00 01 00 00 ff  03 00 01 01 00 ff 03 00   ········ ········
0060  01 00 00 ff 03 00 01 01  00 ff 03 00 01 00 00 ff   ········ ········
0070  03 00 01 01 00 ff 03 00  01 01 00 ff 03 00 01 00   ········ ········
0080  00 ff 03 00 01 01 01 01  05 10 00 00 01 00 d8 aa   ········ ········
0090  71 5e 8c 07 00 00                                 q^····
```

Jedes „Item“ ist hierbei 5 Bytes lang und der gesamte S7 Response 73 Bytes.

Eine (auf dem Draht) effizientere Implementierung wäre es ein Byte-Array zu lesen, in diesem Fall 2 Bytes.

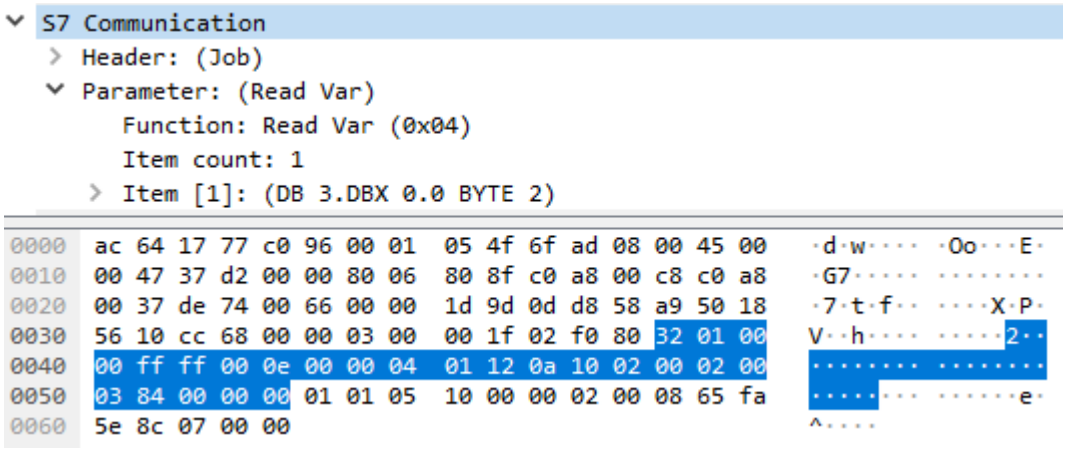
```
aByteArray : ARRAY[0..1] OF BYTE;
```

```
fbReq.AddReadByteArray(ADR(aByteArray), 2, 0, E_S7COMM_DATAAREA.DATA_BLOCKS, 3);
```

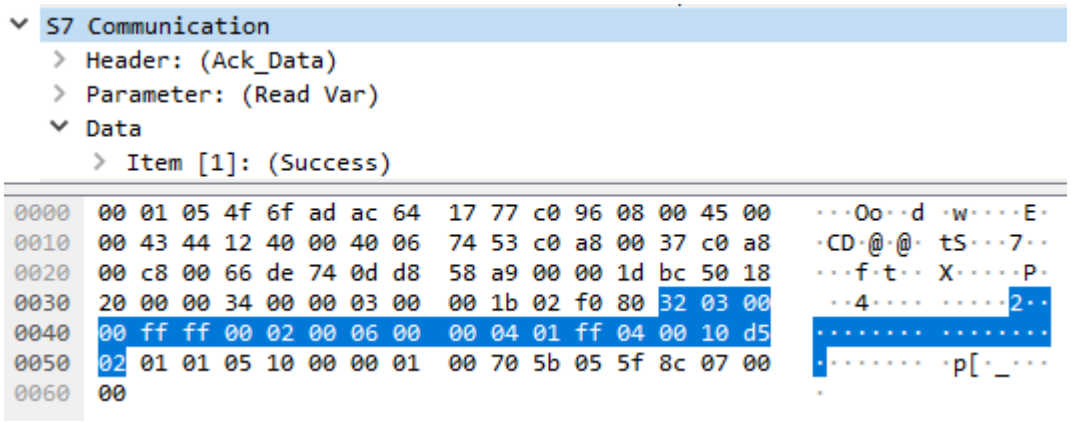
Das Byte-Array würde dann auf die BOOL-Variablen „gemapped“.

```
bBool_1  := aByteArray[0].0;
bBool_2  := aByteArray[0].1;
bBool_3  := aByteArray[0].2;
bBool_4  := aByteArray[0].3;
bBool_5  := aByteArray[0].4;
bBool_6  := aByteArray[0].5;
bBool_7  := aByteArray[0].6;
bBool_8  := aByteArray[0].7;
bBool_9  := aByteArray[1].0;
bBool_10 := aByteArray[1].1;
```

Der Request würde in diesem Fall wie folgt aussehen:



Der Payload ist in diesem Fall nur noch 24 Bytes lang (also 108 Bytes weniger). Die entsprechende Response wäre dann:



Der Payload der Response ist in diesem Fall nur noch 20 Bytes lang (vorher 73 Bytes).

Request:

- Direkte Implementierung: 132 Bytes
- Optimierte Implementierung: 24 Bytes
- Ersparnis: 108 Bytes

Response:

- Direkte Implementierung: 73 Bytes
- Optimierte Implementierung: 20 Bytes
- Ersparnis: 53 Bytes

Weitere Informationen

Wird ein Request-Frame länger als die S7-Steuerung verarbeiten kann, so muss der TwinCAT 3 S7 Kommunikationstreiber den Request in mehrere Requests aufteilen. Dadurch kann sich die Antwortzeit um ein Vielfaches erhöhen.

Beispiel 2: Auslesen von vielen WORD-Variablen

Die Konfiguration im TIA Projekt enthält mehrere (in diesem Beispiel 5) WORD-Variablen, welche sich im Speicher direkt „nacheinander“ befinden und ausgelesen werden sollen:

	Name	Data type	Offset	Start value
1	Static			
2	Word1	Word	0.0	16#0123
3	Word2	Word	2.0	16#4567
4	Word3	Word	4.0	16#89AB
5	Word4	Word	6.0	16#CDEF
6	Word5	Word	8.0	16#1122

Hierbei gilt dasselbe Prinzip wie im Beispiel 1: jeder Eintrag erhöht die Payload-Größe des Requests und der zugehörigen Response. Zur Optimierung kann derselbe Ansatz gewählt werden (bitte beachten Sie die „Endianess“). Ein optimiertes SPS-Programm könnte dann wie folgt aussehen:

```
VAR_OUTPUT
    aWordArray : ARRAY[0..4] OF WORD;
END_VAR
VAR
    aByteArray : ARRAY[0..9] OF BYTE;
    pWord      : POINTER TO WORD;
    nIdx       : INT;
END_VAR
```

```
fbReq.AddReadByteArray(ADR(aByteArray), 10, 0, E_S7COMM_DATAAREA.DATA_BLOCKS, 5);
```

Die Einträge müssen dann entsprechend „gemapped“ und richtig „gedreht“ werden.

```
FOR nIdx := 0 TO 4 BY 1 DO
    pWord := ADR(aByteArray[nIdx *2]);
    aWordArray[nIdx] := HOST_TO_BE16(pWord^);
END_FOR
```

Die Ersparnis im Vergleich zur direkten (offensichtlichen) Implementierung wäre dann wie folgt:

Request:

- Direkte Implementierung: 72 Bytes
- Optimierte Implementierung: 24 Bytes
- Ersparnis: 48 Bytes

Response:

- Direkte Implementierung: 44 Bytes
- Optimierte Implementierung: 28 Bytes
- Ersparnis: 16 Bytes

4.10 String-Länge

Sie können die Standardlänge von STRING-Variablen ändern, indem Sie die Länge im entsprechenden DataType-Feld bearbeiten. Beispiel:

Idx	Name	Data Type	S7 Data Area	S7 Byte Address
1	S7CommVar	STRING(80)	INPUT	0

Idx	Name	Data Type	S7 Data Area	S7 Byte Address
1	S7CommVar	STRING(80)	INPUT	0

Idx	Name	Data Type	S7 Data Area	S7 Byte Address
1	S7CommVar	STRING(95)	INPUT	0

Idx	Name	Data Type	S7 Data Area	S7 Byte Address
1	S7CommVar	STRING(95)	INPUT	0

5 SPS API

5.1 Tc3_S7Comm

5.1.1 Funktionsbausteine

5.1.1.1 FB_S7CommConnection

Mit dem Funktionsbaustein FB_S7CommConnection kann eine TCP/IP basierte Verbindung zu einer Siemens S7 Steuerung hergestellt werden. Über den Ausgang bIsConnected kann festgestellt werden, ob die Verbindung erfolgreich hergestellt wurde. Fehler beim Verbindungsdaufbau werden über den Ausgang bError und sErrorTxt angezeigt.

Syntax

Definition:

```
VAR_INPUT
    bExecute      : BOOL;
    sIpAddr       : STRING(15);
    eCpuType      : E_S7COMM_CPUTYPE;
    nRack         : UINT;
    nSlot         : UINT;
END_VAR
VAR_OUTPUT
    bError        : BOOL;
    sErrorTxt     : STRING;
    bBusy         : BOOL;
    bIsConnected  : BOOL;
END_VAR
```

Eingänge

Name	Datentyp	Beschreibung
bExecute	BOOL	Bei einer steigenden Flanke wird die Verbindung zur S7 Steuerung aufgebaut. Bei einer fallenden Flanke wird die Verbindung getrennt.
sIpAddr	STRING(15)	IP Adresse der Siemens S7 Steuerung.
eCpuType	E_S7COMM_CPUTYPE [▶ 61]	Typ der Siemens S7 Steuerung
nRack	UINT	Rack-ID der S7 Steuerung. Diese kann aus dem S7 Gerätemanager bezogen werden.
nSlot	UINT	Slot-ID der S7 Steuerung. Diese kann aus dem S7 Gerätemanager bezogen werden.

Ausgänge

Name	Datentyp	Beschreibung
bError	BOOL	Schaltet auf TRUE, wenn bei der Ausführung ein Fehler aufgetreten ist.
sErrorTxt	STRING	Enthält im Fehlerfall den Fehlertext.
bBusy	BOOL	TRUE, bis der Baustein einen Befehl ausgeführt hat. Solange bBusy = TRUE ist, akzeptiert der Baustein keine neuen Befehle.
bIsConnected	BOOL	TRUE, wenn die Verbindung zur Siemens S7 Steuerung hergestellt wurde.

Methoden

Name	Definitionsort	Beschreibung
AddRequest [▶ 37]	Lokal	Fügt der Verbindung einen Single- oder CyclicRequest hinzu.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.1.1 AddRequest

Fügt einen Single/Cyclic Request (FB_S7CommSingleRequest [▶ 49], FB_S7CommCyclicRequest [▶ 38]) zu einer S7 Kommunikationsverbindung (FB_S7CommConnection [▶ 36]) hinzu. An der jeweiligen Request-Art des hinzugefügten Objekts lassen sich Lese/Schreib Kommandos definieren.

Syntax

```
METHOD AddRequest : BOOL
VAR_INPUT
    ipRequest : ITF_S7CommRequest;
END_VAR
```

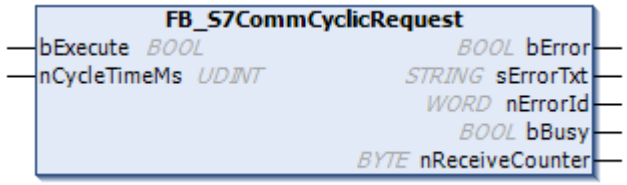
Rückgabewert

Name	Typ	Beschreibung
AddRequest	BOOL	TRUE wenn die Methode korrekt ausgeführt wurde. Im Fehlerfall liefern die Error-Ausgänge des Funktionsbausteins weitere Informationen zur Fehlerursache.

Eingänge

Name	Typ	Beschreibung
ipRequest	ITF_S7CommRequest	Single/Cyclic Request Objekt

5.1.1.2 FB_S7CommCyclicRequest



Mit dem Funktionsbaustein FB_S7CommCyclicRequest kann eine zyklische Abarbeitung (read/write) von Datenpunkten einer S7 Kommunikationsverbindung konfiguriert werden. Über den Ausgang bError kann festgestellt werden, ob der Request erfolgreich durchgeführt wurde. Eventuell auftretende Fehler beim Request werden über den Ausgang sErrorTxt und nErrorId angezeigt.

Syntax

Definition:

```
VAR_INPUT
    bExecute        : BOOL;
    nCycleTimeMs    : UDINT;
END_VAR
VAR_OUTPUT
    bError          : BOOL;
    sErrorTxt       : STRING;
    nErrorId        : WORD;
    bBusy           : BOOL;
    nReceiveCounter : BYTE;
END_VAR
```

Eingänge

Name	Datentyp	Beschreibung
bExecute	BOOL	Der Funktionsbaustein wird durch eine steigende Flanke an diesem Eingang ausgeführt.
nCycleTimeMs	UDINT	Zu verwendende Zykluszeit in [ms].

Ausgänge

Name	Datentyp	Beschreibung
bError	BOOL	Schaltet auf TRUE, wenn bei der Ausführung ein Fehler aufgetreten ist.
sErrorTxt	STRING	Enthält im Fehlerfall den Fehlertext.
nErrorId	WORD	Gibt im Fehlerfall den Fehlercode aus.
bBusy	BOOL	TRUE, bis der Baustein einen Befehl ausgeführt hat. Solange bBusy = TRUE ist, akzeptiert der Baustein keine neuen Befehle.
nReceiveCounter	BYTE	Counter für von der S7 Steuerung empfangene Antworten. Damit lässt sich überprüfen, ob eine Response in der zyklischen Abarbeitung empfangen wurde.

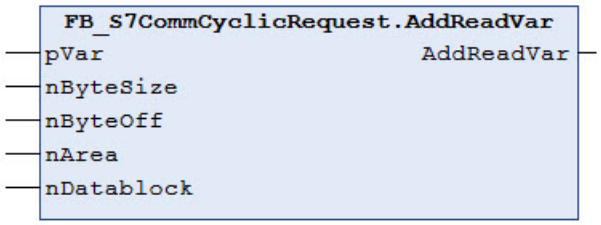
Methoden

Name	Definitionsort	Beschreibung
AddReadVar [▶ 39]	Lokal	Fügt dem Request ein Lese-Kommando für einen bestimmten S7 Datenpunkt hinzu.
AddReadBit [▶ 40]	Lokal	Fügt dem Request ein Lese-Kommando für einen bestimmten S7 Datenpunkt vom Typ BIT hinzu.
AddReadString [▶ 41]	Lokal	Fügt ein Lese-Kommando auf einen S7 Datenpunkt vom Typ STRING zu einem Request hinzu.
AddReadByteArray [▶ 42]	Lokal	Fügt dem Request ein Lese-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 nacheinander liegende Datenpunkte ausgelesen werden sollen.
AddWriteVar [▶ 43]	Lokal	Fügt dem Request ein Schreib-Kommando für einen bestimmten S7 Datenpunkt hinzu.
AddWriteBit [▶ 44]	Lokal	Fügt dem Request ein Schreib-Kommando für einen bestimmten S7 Datenpunkt vom Typ BIT hinzu.
AddWriteString [▶ 45]	Lokal	Fügt ein Schreib-Kommando auf einen S7 Datenpunkt vom Typ STRING zu einem Request hinzu.
AddWriteByteArray [▶ 46]	Lokal	Fügt dem Request ein Lese-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 nacheinander liegende Datenpunkte geschrieben werden sollen.
RemoveRead [▶ 47]	Lokal	Entfernt eine Variable aus einem Read-Request.
RemoveWrite [▶ 48]	Lokal	Entfernt eine Variable aus einem Write-Request.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.1 AddReadVar



Fügt ein Lese-Kommando auf einen S7 Datenpunkt zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert.

Syntax

```
METHOD AddReadVar   : HRESULT
VAR_INPUT
    pVar            : PVOID;
    nByteSize       : WORD;
    nByteOff        : WORD;
    nArea           : E_S7COMM_DATAAREA;
    nDatablock      : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

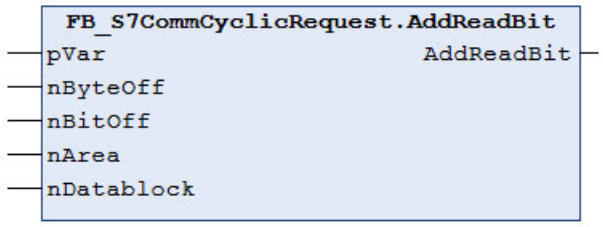
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.2 AddReadBit



Fügt ein Lese-Kommando auf einen S7 Datenpunkt vom Typ BIT zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die übergebene Adresse der Zielvariablen muss vom Datentyp BOOL sein (kein BIT).

Syntax

```
METHOD AddReadBit    : HRESULT
VAR_INPUT
    pVar             : PVOID;
    nByteOff         : WORD;
    nBitOff          : BYTE;
    nArea            : E_S7COMM_DATAAREA;
    nDatablock       : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadBit	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

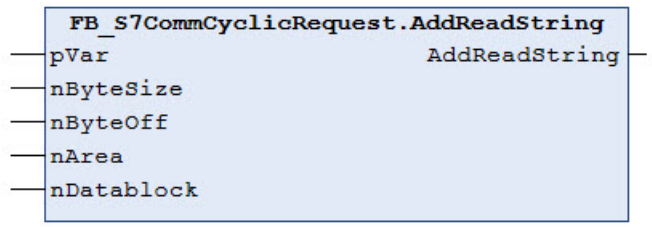
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteOff	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nBitOff	BYTE	Bit Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.3 AddReadString



Fügt ein Lese-Kommando auf einen S7 Datenpunkt vom Typ STRING zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die Länge des Strings darf 254 Zeichen nicht überschreiten.

Syntax

```
METHOD AddReadString    : HRESULT
VAR_INPUT
    pVar                : PVOID;
    nByteSize           : WORD;
    nByteOff            : WORD;
    nArea               : E_S7COMM_DATAAREA;
    nDatablock          : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadString	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.4 AddReadByteArray

Fügt dem Request ein Lese-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 Steuerung nacheinander liegende Datenpunkte gelesen werden sollen.

Syntax

```
METHOD AddReadVar    : HRESULT
VAR_INPUT
    pVar             : PVOID;
    nByteSize        : WORD;
    nByteOff         : WORD;
    nArea            : E_S7COMM_DATAAREA;
    nDatablock       : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert.

Name	Datentyp	Beschreibung
		E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

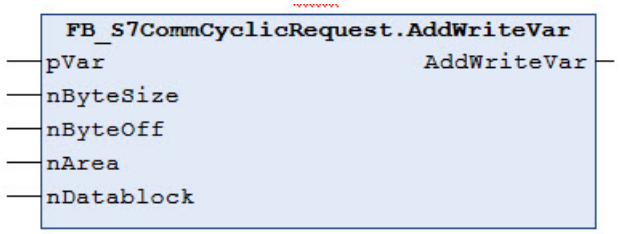
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.5 AddWriteVar



Fügt ein Schreib-Kommando auf einen S7 Datenpunkt zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert.

Syntax

```
METHOD AddWriteVar    : HRESULT
VAR_INPUT
    pVar              : PVOID;
    nByteSize         : WORD;
    nByteOff          : WORD;
    nArea             : E_S7COMM_DATAAREA;
    nDatablock        : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddWriteVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

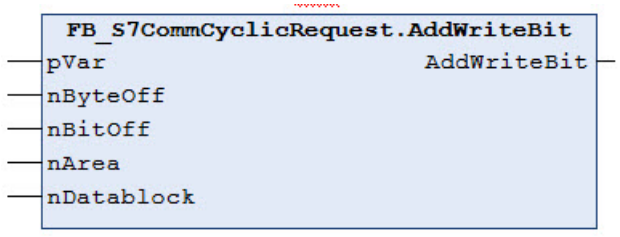
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Quellvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu schreibenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.6 AddWriteBit



Fügt ein Lese-Kommando auf einen S7 Datenpunkt vom Typ BIT zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die übergebene Adresse der Quellvariablen muss vom Datentyp BOOL sein (kein BIT).

Syntax

```
METHOD AddWriteBit    : HRESULT
VAR_INPUT
    pVar              : PVOID;
    nByteOff          : WORD;
    nBitOff           : BYTE;
    nArea             : E_S7COMM_DATAAREA;
    nDatablock        : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddWriteBit	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

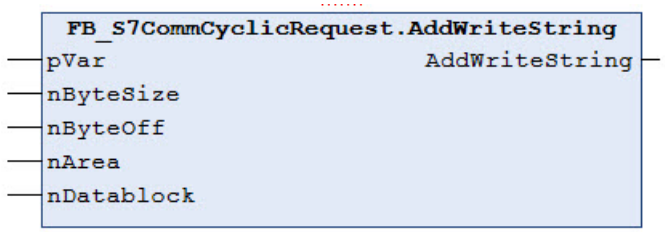
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteOff	WORD	Bytelänge des zu schreibenden Datentyps aus der S7 Steuerung
nBitOff	BYTE	Bit Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.7 AddWriteString



Fügt ein Schreib-Kommando auf einen S7 Datenpunkt vom Typ STRING zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die maximale Länge des Strings darf 254 Zeichen nicht überschreiten.

Syntax

```
METHOD AddWriteString    : HRESULT
VAR_INPUT
    pVar                 : PVOID;
    nByteSize            : WORD;
    nByteOff             : WORD;
    nArea                : E_S7COMM_DATAAREA;
    nDatablock           : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddWriteString	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Quellvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu schreibenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.8 AddWriteByteArray

Fügt dem Request ein Schreib-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 Steuerung nacheinander liegende Datenpunkte geschrieben werden sollen.

Syntax

```
METHOD AddReadVar    : HRESULT
VAR_INPUT
    pVar             : PVOID;
    nByteSize        : WORD;
    nByteOff         : WORD;
    nArea            : E_S7COMM_DATAAREA;
    nDatablock       : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert.

Name	Datentyp	Beschreibung
		E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

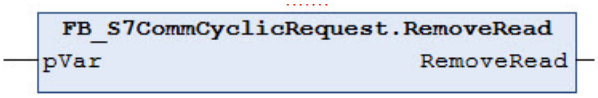
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.9 RemoveRead



Entfernt eine Variable aus einem Read-Request. Die Variable wird über ihre Adresse in der TwinCAT SPS spezifiziert. Es darf kein Read-Request zu dem Zeitpunkt anstehen, an dem die Methode ausgeführt wird, d.h. der Ausgang bBusy des Funktionsbausteins muss FALSE sein.

Syntax

```
METHOD RemoveRead    : HRESULT
VAR_INPUT
    pVar             : PVOID;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
RemoveRead	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden

Name	Datentyp	Beschreibung
		E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

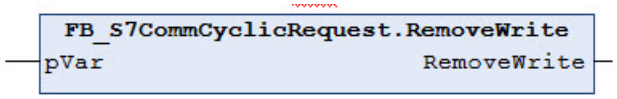
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Variablen in der TwinCAT SPS

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.2.10 RemoveWrite



Entfernt eine Variable aus einem Write-Request. Die Variable wird über ihre Adresse in der TwinCAT SPS spezifiziert. Es darf kein Write-Request zu dem Zeitpunkt anstehen, an dem die Methode ausgeführt wird, d.h. der Ausgang bBusy des Funktionsbausteins muss FALSE sein.

Syntax

```
METHOD RemoveWrite    : HRESULT
VAR_INPUT
    pVar              : PVOID;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
RemoveWrite	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Variablen in der TwinCAT SPS

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3 FB_S7CommSingleRequest

Mit dem Funktionsbaustein FB_S7CommSingleRequest kann ein read/write Request auf einen Datenpunkt einer S7 Kommunikationsverbindung durchgeführt werden. Über den Ausgang bError kann festgestellt werden, ob der Request erfolgreich durchgeführt wurde. Eventuell auftretende Fehler beim Request werden über den Ausgang sErrorTxt und nErrorId angezeigt.

Syntax

Definition:

```
VAR_INPUT
    bExecute      : BOOL;
END_VAR
VAR_OUTPUT
    bError        : BOOL;
    sErrorTxt     : STRING;
    nErrorId      : WORD;
    bBusy         : BOOL;
END_VAR
```

Eingänge

Name	Datentyp	Beschreibung
bExecute	BOOL	Der Funktionsbaustein wird durch eine steigende Flanke an diesem Eingang ausgeführt.

Ausgänge

Name	Datentyp	Beschreibung
bError	BOOL	Schaltet auf TRUE, wenn bei der Ausführung ein Fehler aufgetreten ist.
sErrorTxt	STRING	Enthält im Fehlerfall den Fehlertext.
nErrorId	WORD	Gibt im Fehlerfall den Fehlercode aus.
bBusy	BOOL	TRUE, bis der Baustein einen Befehl ausgeführt hat. Solange bBusy = TRUE ist, akzeptiert der Baustein keine neuen Befehle.

Methoden

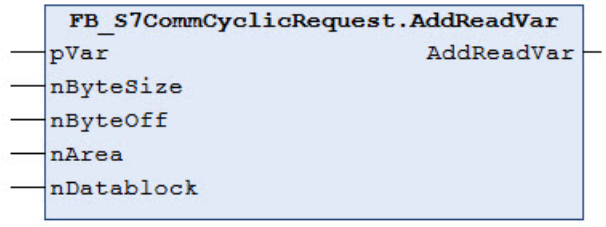
Name	Definitionsort	Beschreibung
AddReadVar [▶ 39]	Lokal	Fügt dem Request ein Lese-Kommando für einen bestimmten S7 Datenpunkt hinzu.
AddReadBit [▶ 51]	Lokal	Fügt dem Request ein Lese-Kommando für einen bestimmten S7 Datenpunkt vom Typ BIT hinzu.

Name	Definitionsort	Beschreibung
AddReadString [▶ 52]	Lokal	Fügt dem Request ein Lese-Kommando für einen bestimmten S7 Datenpunkt vom Typ STRING hinzu.
AddReadByteArray [▶ 53]	Lokal	Fügt dem Request ein Lese-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 nacheinander liegende Datenpunkte ausgelesen werden sollen.
AddWriteVar [▶ 43]	Lokal	Fügt dem Request ein Schreib-Kommando für einen bestimmten S7 Datenpunkt hinzu.
AddWriteBit [▶ 55]	Lokal	Fügt dem Request ein Schreib-Kommando für einen bestimmten S7 Datenpunkt vom Typ BIT hinzu.
AddWriteString [▶ 56]	Lokal	Für dem Request ein Schreib-Kommando für einen bestimmten S7 Datenpunkt vom Typ STRING hinzu.
AddWriteByteArray [▶ 57]	Lokal	Fügt dem Request ein Lese-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 nacheinander liegende Datenpunkte geschrieben werden sollen.
RemoveRead [▶ 58]	Lokal	Entfernt eine Variable aus einem Read-Request.
RemoveWrite [▶ 59]	Lokal	Entfernt eine Variable aus einem Write-Request.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.1 AddReadVar



Fügt ein Lese-Kommando auf einen S7 Datenpunkt zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert.

Syntax

```
METHOD AddReadVar   : HRESULT
VAR_INPUT
    pVar            : PVOID;
    nByteSize       : WORD;
    nByteOff        : WORD;
    nArea           : E_S7COMM_DATAAREA;
    nDatablock      : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

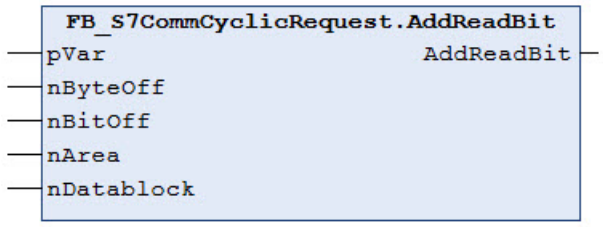
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.2 AddReadBit



Fügt ein Lese-Kommando auf einen S7 Datenpunkt vom Typ BIT zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die übergebene Adresse der Zielvariablen muss vom Datentyp BOOL sein (kein BIT).

Syntax

```
METHOD AddReadBit    : HRESULT
VAR_INPUT
    pVar             : PVOID;
    nByteOff         : WORD;
    nBitOff          : BYTE;
    nArea            : E_S7COMM_DATAAREA;
    nDatablock       : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadBit	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

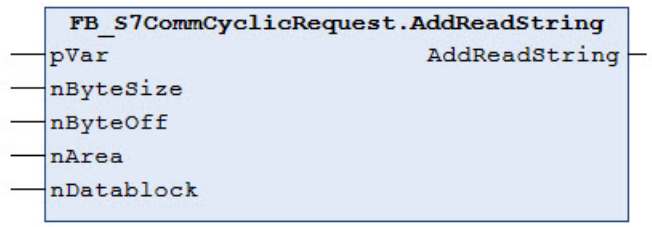
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteOff	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nBitOff	BYTE	Bit Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.3 AddReadString



Fügt ein Lese-Kommando auf einen S7 Datenpunkt vom Typ STRING zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die Länge des Strings darf 254 Zeichen nicht überschreiten.

Syntax

```
METHOD AddReadString    : HRESULT
VAR_INPUT
    pVar                : PVOID;
    nByteSize           : WORD;
    nByteOff            : WORD;
    nArea               : E_S7COMM_DATAAREA;
    nDatablock          : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadString	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.4 AddReadByteArray

Fügt dem Request ein Lese-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 Steuerung nacheinander liegende Datenpunkte gelesen werden sollen.

Syntax

```
METHOD AddReadVar    : HRESULT
VAR_INPUT
    pVar             : PVOID;
    nByteSize        : WORD;
    nByteOff         : WORD;
    nArea            : E_S7COMM_DATAAREA;
    nDatablock       : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert.

Name	Datentyp	Beschreibung
		E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

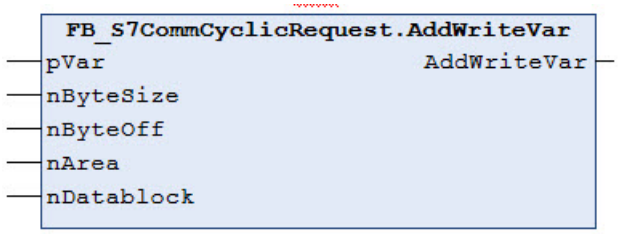
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.5 AddWriteVar



Fügt ein Schreib-Kommando auf einen S7 Datenpunkt zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert.

Syntax

```
METHOD AddWriteVar    : HRESULT
VAR_INPUT
    pVar              : PVOID;
    nByteSize         : WORD;
    nByteOff          : WORD;
    nArea             : E_S7COMM_DATAAREA;
    nDatablock        : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddWriteVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

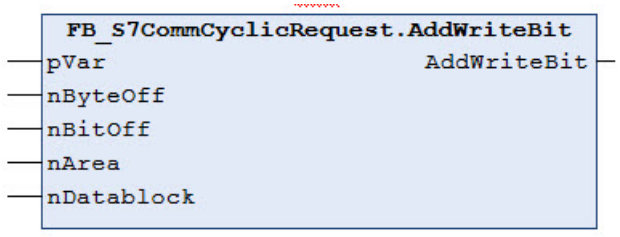
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Quellvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu schreibenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.6 AddWriteBit



Fügt ein Lese-Kommando auf einen S7 Datenpunkt vom Typ BIT zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die übergebene Adresse der Quellvariablen muss vom Datentyp BOOL sein (kein BIT).

Syntax

```
METHOD AddWriteBit    : HRESULT
VAR_INPUT
    pVar              : PVOID;
    nByteOff          : WORD;
    nBitOff           : BYTE;
    nArea             : E_S7COMM_DATAAREA;
    nDatablock        : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddWriteBit	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

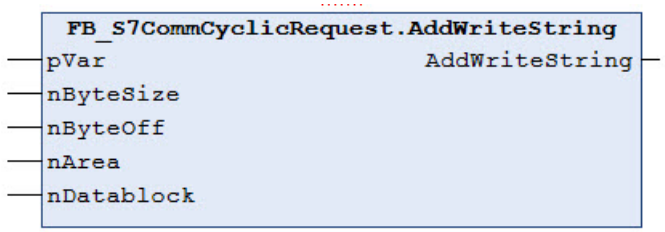
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteOff	WORD	Bytelänge des zu schreibenden Datentyps aus der S7 Steuerung
nBitOff	BYTE	Bit Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.7 AddWriteString



Fügt ein Schreib-Kommando auf einen S7 Datenpunkt vom Typ STRING zu einem Request hinzu. Der Datenpunkt wird über seine absolute Adresse in der S7 Steuerung spezifiziert. Die maximale Länge des Strings darf 254 Zeichen nicht überschreiten.

Syntax

```
METHOD AddWriteString    : HRESULT
VAR_INPUT
    pVar                 : PVOID;
    nByteSize            : WORD;
    nByteOff             : WORD;
    nArea                : E_S7COMM_DATAAREA;
    nDatablock           : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddWriteString	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Quellvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu schreibenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.8 AddWriteByteArray

Fügt dem Request ein Schreib-Kommando in optimierter [▶ 30] Weise hinzu, wenn mehrere im Speicher der S7 Steuerung nacheinander liegende Datenpunkte geschrieben werden sollen.

Syntax

```
METHOD AddReadVar    : HRESULT
VAR_INPUT
    pVar             : PVOID;
    nByteSize        : WORD;
    nByteOff         : WORD;
    nArea            : E_S7COMM_DATAAREA;
    nDatablock       : WORD;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
AddReadVar	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert.

Name	Datentyp	Beschreibung
		E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

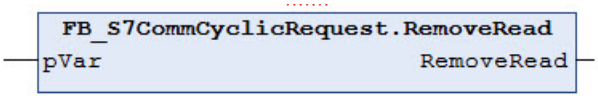
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Zielvariablen in der TwinCAT SPS
nByteSize	WORD	Bytelänge des zu lesenden Datentyps aus der S7 Steuerung
nByteOff	WORD	Byte Offset in der S7 Steuerung
nArea	E_S7COMM_DATAAREA [▶ 61]	S7 Data Area
nDatablock	WORD	ID des Datenblocks. Wird nur versendet, wenn als Data Area E_S7COMM_DATAAREA.DATA_BLOCKS verwendet wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.9 RemoveRead



Entfernt eine Variable aus einem Read-Request. Die Variable wird über ihre Adresse in der TwinCAT SPS spezifiziert. Es darf kein Read-Request zu dem Zeitpunkt anstehen, an dem die Methode ausgeführt wird, d.h. der Ausgang bBusy des Funktionsbausteins muss FALSE sein.

Syntax

```
METHOD RemoveRead    : HRESULT
VAR_INPUT
    pVar             : PVOID;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
RemoveRead	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden

Name	Datentyp	Beschreibung
		E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

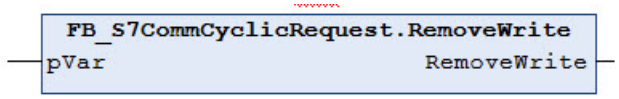
Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Variablen in der TwinCAT SPS

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.1.3.10 RemoveWrite



Entfernt eine Variable aus einem Write-Request. Die Variable wird über ihre Adresse in der TwinCAT SPS spezifiziert. Es darf kein Write-Request zu dem Zeitpunkt anstehen, an dem die Methode ausgeführt wird, d.h. der Ausgang bBusy des Funktionsbausteins muss FALSE sein.

Syntax

```
METHOD RemoveWrite    : HRESULT
VAR_INPUT
    pVar              : PVOID;
END_VAR
```

Rückgabewert

Name	Datentyp	Beschreibung
RemoveWrite	HRESULT	E_HRESULTAdsErr.NOTINIT = Funktionsbaustein ist nicht richtig initialisiert. E_HRESULTAdsErr.BUSY = Request ist aktiv E_HRESULTAdsErr.INVALIDDATA = Ein Übergabeparameter ist falsch definiert worden E_HRESULTAdsErr.INVALIDSIZE = Die Framelänge ist größer als die PDU Länge

Eingänge

Name	Datentyp	Beschreibung
pVar	PVOID	Adresse der Variablen in der TwinCAT SPS

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.2 Datentypen

5.1.2.1 E_S7COMM_CONNECT_STATE

E_S7COMM_CONNECT_STATE gibt den Status der Kommunikationsverbindung mit der S7 Steuerung an.

```
TYPE E_S7COMM_CONNECT_STATE:
(
    IDLE,
    START,
    TCP_SETUP,
    TCP_WAIT,
    COTP_SETUP,
    COTP_WAIT,
    S7_SETUP,
    S7_WAIT,
    CONNECTED,
    TCP_ERROR,
    COTP_ERROR,
    S7_ERROR,
    RESET,
    TCP_SETUP_ERROR,
    TCP_TIMEOUT_ERROR
);
END_TYPE
```

Parameter

Wert	Beschreibung
IDLE	
START	
TCP_SETUP	
TCP_WAIT	
COTP_SETUP	
COTP_WAIT	
S7_SETUP	
S7_WAIT	
CONNECTED	
TCP_ERROR	
COTP_ERROR	
S7_ERROR	
RESET	
TCP_SETUP_ERROR	
TCP_TIMEOUT_ERROR	

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.2.2 E_S7COMM_CPUTYPE

E_S7COMM_CPUTYPE gibt beim Verbindungsaufbau mit der S7 Steuerung an, um welchen S7 Steuerungstyp es sich handelt.

```
TYPE E_S7COMM_CPUTYPE:
(
    S7300,
    S7400,
    S71200,
    S71500
);
END_TYPE
```

Parameter

Wert	Beschreibung
S7300	
S7400	
S71200	
S71500	

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

5.1.2.3 E_S7COMM_DATAAREA

E_S7COMM_DATAAREA gibt bei der Adressierung von Datenpunkten an, aus welcher S7 Data Area diese stammen.

```
TYPE E_S7COMM_DATAAREA:
(
    INPUT,
    OUTPUT,
    FLAGS,
    DATA_BLOCKS
);
END_TYPE
```

Parameter

Wert	Beschreibung
INPUT	
OUTPUT	
FLAGS	
DATA_BLOCKS	

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc3_S7Comm (Communication)

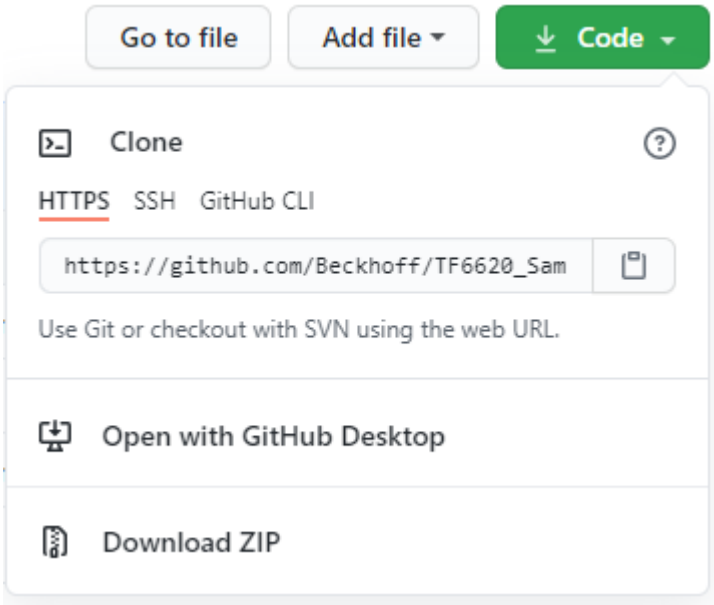
6 Beispiele

Für die beiden Kommunikationswege (Mapping vs. SPS Bibliothek [▶ 21]) mit einer Siemens S7 Steuerung steht jeweils ein eigenes Beispiel zum Download zur Verfügung.

Beispiel	Beschreibung
TF6620_S7CommunicationSample_PLC	Dieses Beispiel zeigt wie sich die SPS Bibliothek Tc3_S7Comm [▶ 36] verwenden lässt, um eine Kommunikationsverbindung mit einer S7 Steuerung herzustellen und Datenpunkte auszulesen bzw. zu schreiben.
TF6620_S7CommunicationSample_SysMan	Dieses Beispiel zeigt wie sich die S7 Kommunikationsverbindung als TwinCAT I/O Gerät konfigurieren lässt, um Datenpunkte von einer S7 Steuerung auszulesen bzw. zu schreiben.

Download

Beispielcode und -konfigurationen für dieses Produkt können über das entsprechende Repository auf GitHub bezogen werden: https://github.com/Beckhoff/TF6620_Samples . Sie haben dort die Möglichkeit das Repository zu clonen oder ein ZIP File mit dem Sample herunterzuladen.



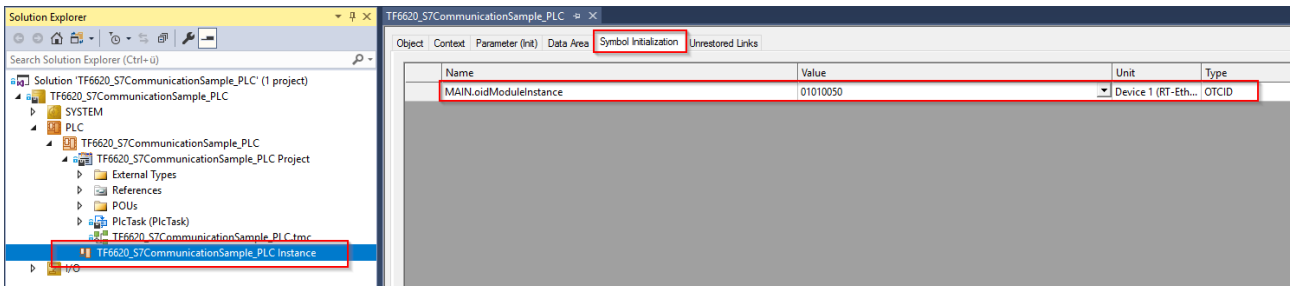
Weitere Informationen

Die Beispiele gehen von einer imaginären S7 Steuerung (S7-1500) aus, welche im lokalen Netzwerk unter der IP-Adresse 192.168.179.1 erreichbar ist. Diese Steuerung wurde für den S7 Protokollzugriff konfiguriert und stellt in der Data Area DATA_BLOCKS folgende Variablen bereit:

Idx	Name	Data Type	S7 Data Area	S7 Byte Address	S7 Bit Offset	S7 Data Block
1	by1	BYTE	DATA_BLOCKS	0	---	1
2	b1	BOOL	DATA_BLOCKS	1	0	1
3	b2	BOOL	DATA_BLOCKS	1	1	1
4	b3	BOOL	DATA_BLOCKS	1	2	1
5	b4	BOOL	DATA_BLOCKS	1	3	1
6	w1	DWORD	DATA_BLOCKS	2	---	1
7	r1	REAL	DATA_BLOCKS	6	---	1
8	r2	REAL	DATA_BLOCKS	10	---	1
9	i1	INT	DATA_BLOCKS	14	---	1

Bitte adaptieren Sie die Beispiele passend zu Ihrer Betriebsumgebung.

In dem SPS-Beispiel wird die Funktionsbausteininstanz vom Typ FB_S7CommConnection mit dem TCP/UDP RT Modul vorinitialisiert. Dies geschieht über die Registerkarte **Symbol Initialization** in den Eigenschaften der SPS Projektinstanz.


Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'TF6620_S7CommunicationSample_PLC' (1 project)
TF6620_S7CommunicationSample_PLC
SYSTEM
PLC
TF6620_S7CommunicationSample_PLC
TF6620_S7CommunicationSample_PLC Project
External Types
References
POUs
PlcTask (PlcTask)
TF6620_S7CommunicationSample_PLC.tmc
TF6620_S7CommunicationSample_PLC Instance
I/O
TF6620_S7CommunicationSample_PLC
Object
Context
Parameter (Init)
Data Area
Symbol Initialization
Unrestored Links
Name
Value
Unit
Type
MAIN.oidModuleInstance
01010050
Device 1 (RT-Eth...
OTCID

7 Anhang

7.1 Troubleshooting

Verhalten	Kategorie	Beschreibung
Es kommt keine Verbindung zustande. Der Connector State geht in den Zustand 0xF3 TCP Timeout.	Connector	Es kann keine TCP-Verbindung zu der S7 Steuerung aufgebaut werden. Bitte überprüfen Sie, ob die Steuerung von Ihrem Zielsystem aus erreichbar ist, z.B. über einen Ping Befehl.
Es kommt keine Verbindung zustande. Der Connector State geht in den Zustand 0xF3 TCP Timeout. Die IP-Adresse der S7 Steuerung ist über einen Ping Befehl erreichbar.	Connector	Bitte überprüfen Sie, ob sich hinter der IP-Adresse die gewünschte S7 Steuerung befindet.
Es kommt keine Verbindung zustande. Der Connector State geht in den Zustand 0xF4 COTP Setup Error.	Connector	Es konnte eine TCP-Verbindung zur S7 Steuerung aufgebaut werden, jedoch schlug die Verbindung zum S7 Communication Service fehl. Bitte überprüfen Sie, ob die Verbindungsparameter "CPU Type", "Rack" und "Slot" zu der gewünschten S7 Steuerung passen.
Die TCP Verbindung zur S7 Steuerung kann nicht hergestellt werden.	Connector	Bitte stellen Sie sicher dass sich zwischen dem TwinCAT Gerät und der S7 Steuerung keine Firewalls befinden, welche die Datenverbindung blockieren könnten oder der entsprechende Kommunikationsport in der Firewall zugelassen wird. Siemens S7 Steuerungen verwenden für eingehende Verbindung den TCP-Port 102.
Es kommt keine Verbindung zustande. Der Connector State geht in den Zustand 0xF2 TCP Setup Error. Der Logger zeigt die folgende Meldung an: S7Connection: src ip address is invalid - maybe ethernet device is not supported?	Connector	Bitte stellen Sie sicher, dass der ausgewählte Netzwerkadapter am Realtime Ethernet Device eine gültige IP-Adresse erhalten hat. Falls die IP-Adresse am Adapter 0.0.0.0 zeigt, überprüfen Sie bitte die korrekte Funktionsweise Ihres DHCP-Servers oder vergeben Sie manuell eine IP-Adresse für den Adapter - entweder in den Windows Netzwerkeinstellungen oder im Dialog "Parameter (Init)" vom TCP/UDP RT Gerät.
Nach einem Write Befehl erscheinen die folgenden Meldungen im Logger: S7Connection: S7 error, errorClass 0x81, ErrorCode 0x04	Request	Generell bedeutet dieser Fehler dass eine bestimmte Funktion vom S7 Gerät nicht unterstützt wird. Üblicherweise tritt dieser Fehler bei S7-1200 und S7-1500 Steuerungen nur dann auf wenn der Remotezugriff nicht aktiviert wurde.

Verhalten	Kategorie	Beschreibung
S7Connection: Please check if remote access is enabled at siemens controller S7Connection State = 0xF5		Manche S7 Steuerungen jedoch (z.B. S7-412-1) erlauben für einen Schreibvorgang nur 2-Byte Datentypen, z.B. WORD, wodurch dieser Fehlercode auch bei einem Schreibvorgang auftreten kann.
Nach einem Read/Write Befehl befindet sich die Verbindung im State 0xF5 S7 Error. Im Logger findet sich der Eintrag "CS7Connector::ReceiveS7Comm( )<<< S7 error, errorClass 81, ErrorCode 04"	Request	Der Remotezugriff auf der S7 Steuerung ist nicht freigegeben. Siehe dazu auch Kapitel Aktivierung des S7 Protokollzugriffs [▶ 28].
Nach einem Read/Write Befehl steht der Request in einem Error Zustand. Das zweite Nibble der Error-Variablen steht auf dem Wert 3 (Address out of range).	Request	Option 1: Der Konfigurationsparameter "S7 Byte Address" ist größer als der Datenbereich der angefragten "Data Area". Option 2: Bitte überprüfen Sie die Einstellung "Optimized Block Access" beim angefragten Data Block der S7 Steuerung. Weitere Informationen hierzu finden Sie im Kapitel Aktivierung des S7 Protokollzugriffs [▶ 28].
Nach einem Read/Write Befehl steht der Request in einem Error Zustand. Das zweite Nibble der Error-Variablen steht auf dem Wert 6 (Object does not exist).	Request	Das angefragte Objekt befindet sich nicht auf der S7 Steuerung. Bitte überprüfen Sie, ob der Konfigurationsparameter "S7 Data Block" richtig gesetzt wurde.
Nach einem Read Befehl werden "falsche Werte" angezeigt.	Request	Bitte überprüfen Sie, ob die Adressdaten des S7 Datenpunkts richtig gewählt wurden oder ob sich diese eventuell geändert haben, z.B. durch eine Änderung im S7 Steuerungsprogramm.
Ich kann keine Input Dara Area zu einem Write-Request hinzufügen	Request	Das Schreiben von Input-Variablen ist nicht zulässig.
Ich habe den Symbolserver zum Target Browser hinzugefügt, sehe aber keine Variablen.	Symbolserver	Bitte stellen Sie sicher dass Sie mindestens TwinCAT 3.1 Build 4024.14 auf dem System verwenden auf dem Sie das Projekt aktiviert haben. Die Symbolserver Schnittstelle ist erst ab dieser TwinCAT Version verfügbar.

7.2 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

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Mehr Informationen:
www.beckhoff.de/tf6620

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