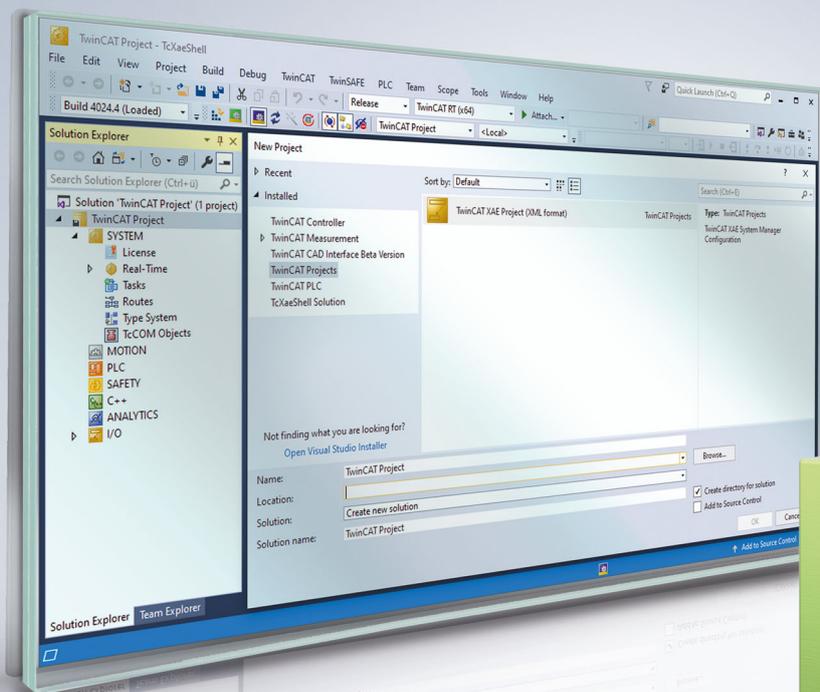


BECKHOFF New Automation Technology

Handbuch | DE

TE1410

TwinCAT 3 | Interface for MATLAB®/Simulink®



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Bis Version 1.1.xxxx.x	8
2.1	Übersicht	8
2.2	Installation	9
2.3	Lizenzen	10
2.4	TE1410 Simulink Bibliothek	10
2.4.1	Asynchrone Blöcke	11
2.4.2	Synchrone Blöcke	15
2.4.3	Utilities.....	23
3	Ab Version 2.x.xxxx.x	24
3.1	Übersicht	24
3.2	Installation	24
3.3	Lizenzen	25
3.4	Kurzeinführung: TwinCAT ADS.....	25
3.5	ADS in MATLAB®.....	27
3.5.1	API-Dokumentation	27
3.5.2	Grundkonzepte.....	30
3.6	ADS in Simulink®.....	30
3.6.1	ADS-Blöcke.....	30
3.7	Beispiele.....	40

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

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

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

Disclaimer

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

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

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

Marken

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

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

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Erklärung der Symbole

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

GEFAHR

Akute Verletzungsgefahr!

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

WARNUNG

Verletzungsgefahr!

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

VORSICHT

Schädigung von Personen!

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

HINWEIS

Schädigung von Umwelt oder Geräten

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

Tipp oder Fingerzeig

i Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

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

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

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

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

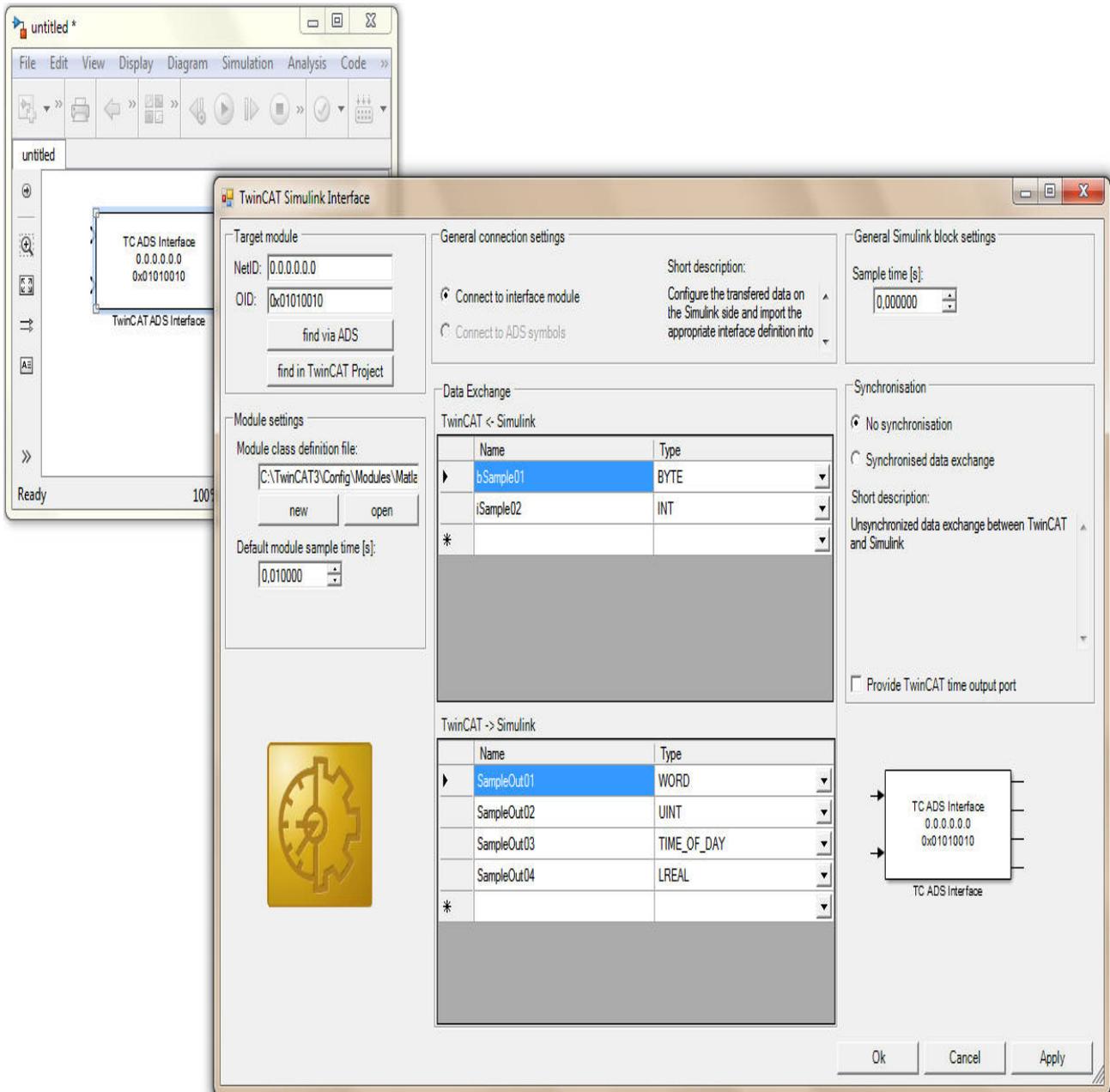
2 Bis Version 1.1.xxxx.x

2.1 Übersicht

TE1410 Interface für MATLAB®/Simulink®

TE1410 Interface für MATLAB®/Simulink® ist eine Schnittstelle für den Datenaustausch zwischen TwinCAT3 und MATLAB®/ Simulink®.

Der Datenaustausch erfolgt über ADS mit Hilfe von Simulink-Blöcken, die in einer Simulink-Bibliothek bereitgestellt werden.



Webinare zu TE1400 und TE1410

Termin	Thema	Referent
01.10.2013	TwinCAT 3 Matlab®/Simulink® -Integration: Einführung, Anwendungsbeispiele, TC3 Interface for Matlab®/Simulink®	Dr. Knut Güttel

Eine Übersicht der aktuellen Beckhoff Webinare finden Sie auf der Beckhoff Homepage: <http://www.beckhoff.com/default.asp?support/webinars.htm>

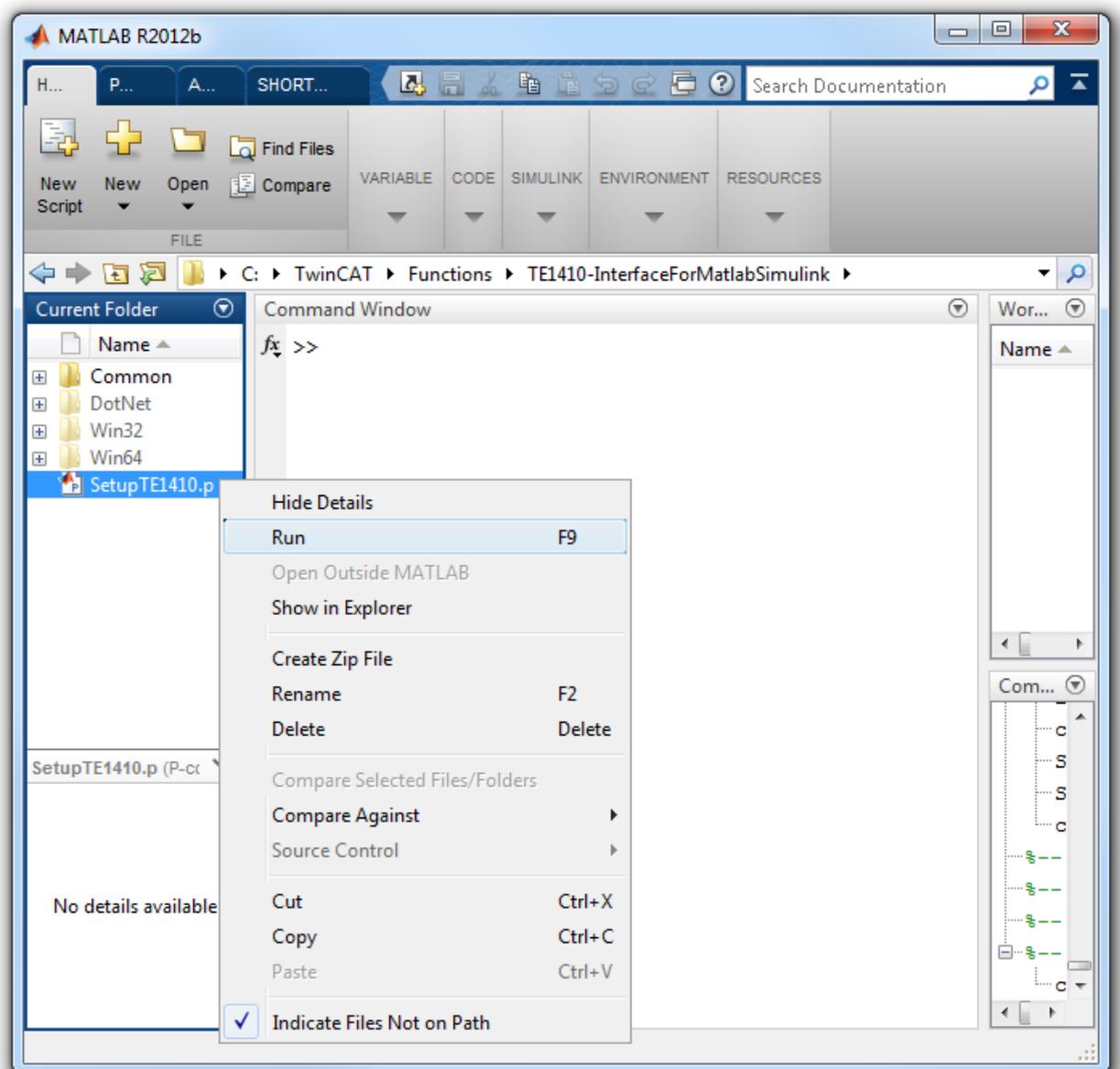
2.2 Installation

Systemvoraussetzungen

- MATLAB®/Simulink® R2010a oder neuere Version.
- TwinCAT 3.0 oder neuere Version.

Installationsanleitung

1. TwinCAT 3-Setup ausführen
2. Setup "TE1410-InterfaceForMatlabSimulink" ausführen
3. Starten Sie MATLAB als Administrator und führen Sie "%TwinCAT3Dir%..\Functions\TE1410-InterfaceForMatlabSimulink\SetupTE1410.p" in MATLAB aus.



HINWEIS**Dateien werden nicht gefunden**

Wird MATLAB in einem System mit aktivierter User Account Control (UAC) ohne Administratorbefugnis ausgeführt, kann der MATLAB-Pfad nicht dauerhaft gespeichert werden. In diesem Fall muss nach jedem Start von MATLAB "SetupTE1410.p" ausgeführt werden, da sonst einige Dateien für die Generierung von TwinCAT-Modulen nicht gefunden werden.

2.3 Lizenzen

Um die Funktionen des TE1410 Interface für MATLAB®/Simulink® nutzen zu können wird eine TE1410 Lizenz auf dem System benötigt, auf dem MATLAB®/Simulink® mit den TE1410-Simulink-Blöcken betrieben wird. Außerdem ist auf diesem Rechner zumindest eine TwinCAT ADS Installation notwendig, um den ADS-Router zur Verfügung zu stellen.

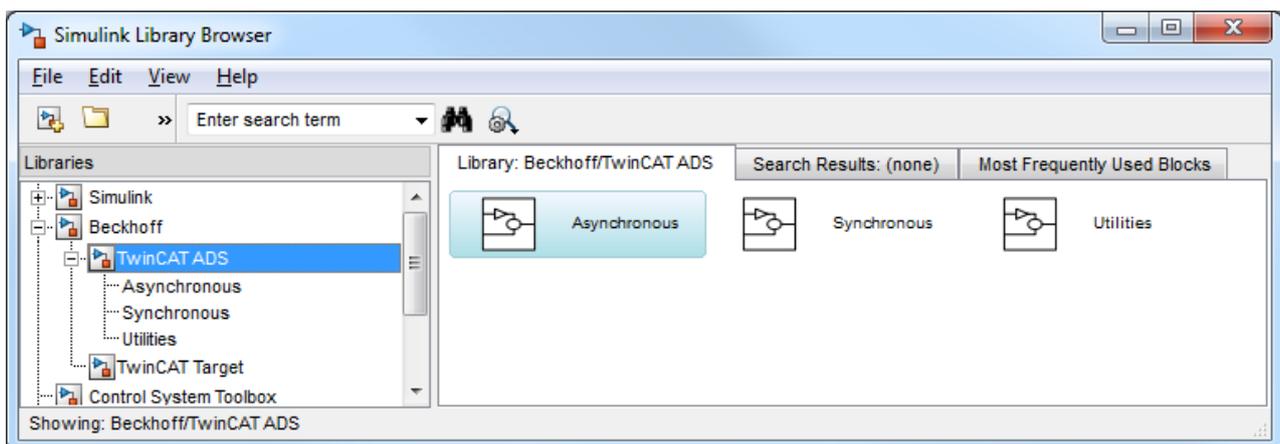
Von diesem System ausgehend, können Sie sich dann auf verschiedene Zielplattformen verbinden. Für die Zielplattformen müssen Sie hinsichtlich der Nutzung mit dem TE1410 keine zusätzlichen Laufzeit-Lizenzen erwerben.

Demoversion

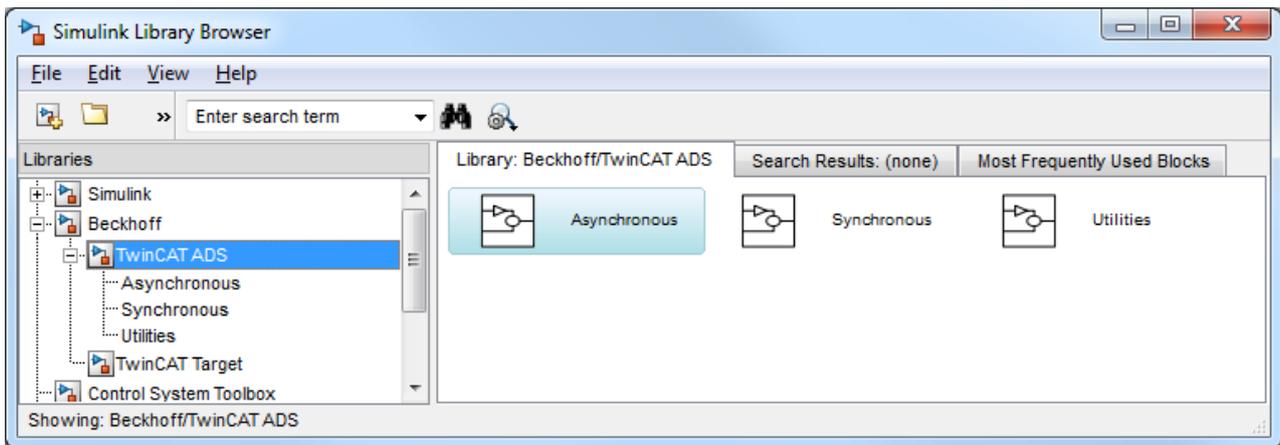
Für das TE1410 existiert keine 7-Tage-Testlizenz, da es sich um ein Engineering-Produkt handelt. Sie können das Produkt jedoch mit einem limitierten Funktionsumfang testen (Demoversion). Die Limitierungen der Demoversion werden automatisch aktiviert, wenn auf dem System keine gültige TE1410-Lizenz gefunden wurde. Dabei wird der Funktionsumfang auf insgesamt 5 ADS-Variablen und 5 ADS-Blöcken pro Simulink-Modell begrenzt.

2.4 TE1410 Simulink Bibliothek

Nach erfolgreicher Installation des „TE1410-InterfaceForMatlabSimulink“ enthält der „Simulink Library Browser“ den Eintrag „Beckhoff / TwinCAT ADS“:

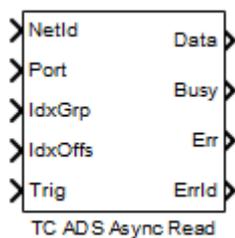


2.4.1 Asynchrone Blöcke



2.4.1.1 TC ADS Async Read

Der Baustein "TC ADS Async Read" erlaubt asynchronen Lesezugriff auf TwinCAT ADS Variablen über deren ADS IndexGroup und ADS IndexOffset. Sie finden diesen Baustein in der Bibliothek "Beckhoff/TwinCAT ADS/Asynchronous" im "Simulink Library Browser".



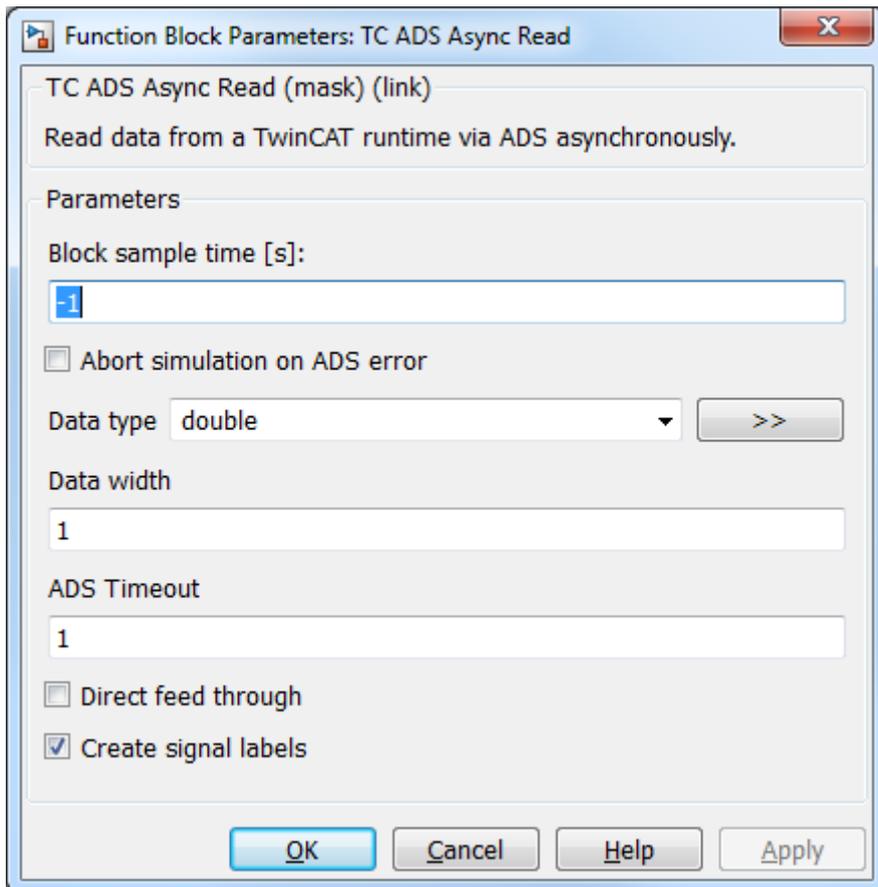
Eingangsports des Blocks

NetId	AMS NetId
Port	ADS Port
IdxGrp	ADS Index Group
IdxOffs	ADS Index Offset
Trig	Positive Flanke löst den Lesebefehl aus

Ausgangsports des Blocks

Daten	Gelesener Wert der ADS-Variable
Busy	TRUE, wenn der Baustein auf eine Antwort des ADS-Servers wartet
Err	TRUE, wenn ein ADS-Fehler vorliegt
ErrId	ADS-Fehlercode

Block-Parameter



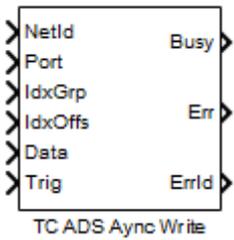
Block sample time	Die Abtastzeit des Simulink-Blocks
Abort simulation on ADS error	Simulation bei dem ersten ADS Fehler stoppen
Data type	Der Datentyp des <i>Daten</i> -Ausgangssignals. Er muss dem Datentyp der übermittelten ADS Variablen entsprechen
Data width	Anzahl der Elemente bei der Übertragung von Arrays; 1 bei skalaren Werten.
ADS Timeout	Simulink wartet auf Antwort der letzten ADS Anforderung bis diese Zeit überschritten wird
Direkt feed through	Eingangswerte des Blocks werden nicht in der Update- sondern der Output-Methode ausgewertet. Dadurch wird sichergestellt, dass die Werte an den Ausgangsports im aktuellen Zeitschritt zu den ADS Parametern an den Eingangsports desselben Zeitschritts gehören. Ansonsten gehören die Ausgangswerte zu den Eingangs-ADS-Parametern des vorherigen Zeitschritts. Die Simulation kann erheblich verlangsamt werden, wenn diese Option aktiviert wird.
Create signal labels	Es werden Signalbezeichnungen für die an den Ausgangsports angeschlossenen Signalleitungen erzeugt

Codegenerierung

Die Codegenerierung aus diesem Block mit Hilfe des „Simulink Coder“ ist nur mit dem TwinCAT Target (TE1400) möglich.

2.4.1.2 TC ADS Async Write

Der Baustein "TC ADS Async Write" erlaubt asynchronen Schreibzugriff auf TwinCAT ADS Variablen über deren ADS IndexGroup und ADS IndexOffset. Sie finden diesen Baustein in der Bibliothek "Beckhoff/TwinCAT ADS/Asynchronous" im "Simulink Library Browser".



Eingangsports des Bausteins

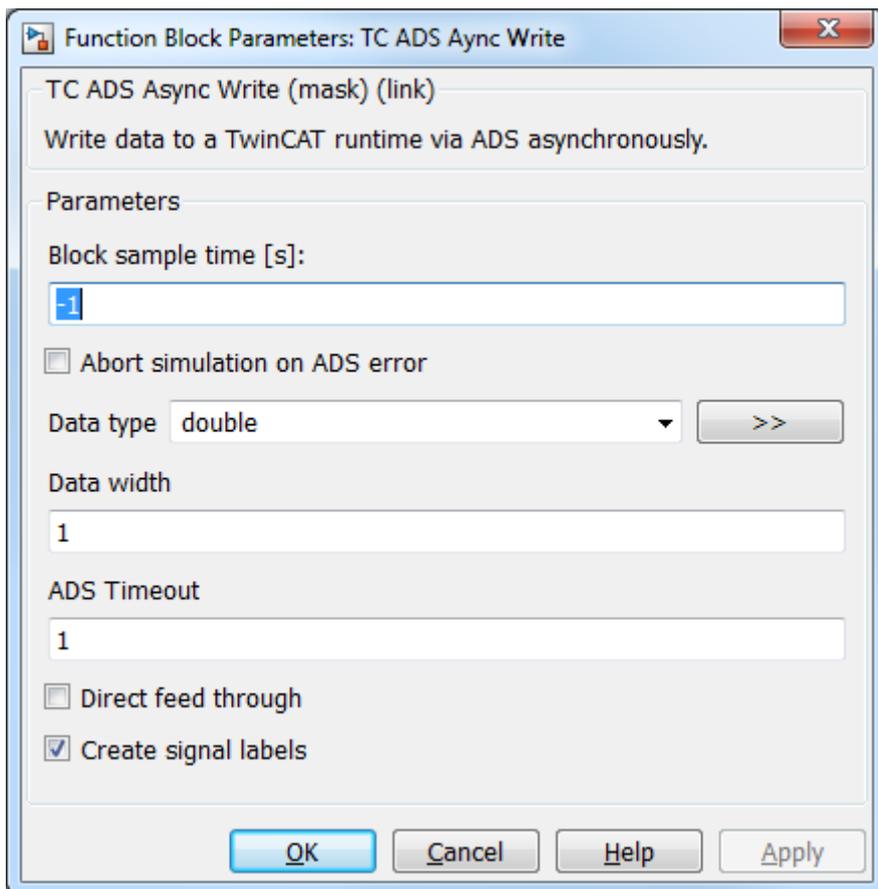
NetId	AMS NetId
Port	ADS Port
IdxGrp	ADS Index Group
IdxOffs	ADS Index Offset
Data	Zu schreibender Wert der ADS-Variable
Trig	Positive Flanke löst den Lesebefehl aus

Ausgangsports des Bausteins

Ausgangsports des Blocks

Busy	TRUE, wenn der Baustein auf eine Antwort des ADS-Servers wartet
Err	TRUE, wenn ein ADS-Fehler vorliegt
ErrId	ADS-Fehlercode

Block-Parameter

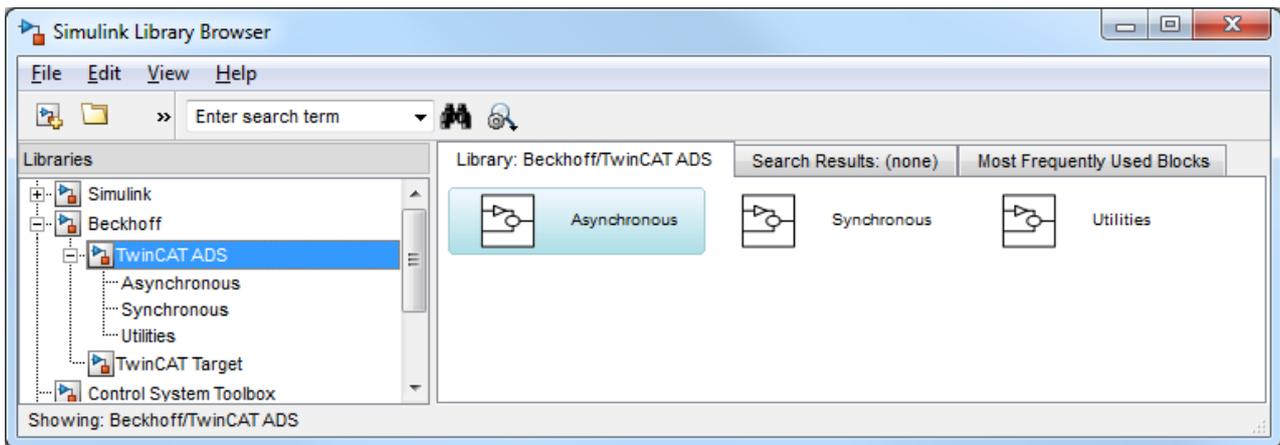


Block sample time	Die Abtastzeit des Simulink-Blocks
Abort simulation on ADS error	Simulation bei dem ersten ADS Fehler stoppen
Data type	Der Datentyp des <i>Daten</i> -Ausgangssignals. Er muss dem Datentyp der übermittelten ADS Variablen entsprechen
Data width	Anzahl der Elemente bei der Übertragung von Arrays; 1 bei skalaren Werten.
ADS Timeout	Simulink wartet auf Antwort der letzten ADS Anforderung bis diese Zeit überschritten wird
Direkt feed through	Eingangswerte des Blocks werden nicht in der Update- sondern der Output-Methode ausgewertet. Dadurch wird sichergestellt, dass die Werte an den Ausgangsports im aktuellen Zeitschritt zu den ADS Parametern an den Eingangsports desselben Zeitschritts gehören. Ansonsten gehören die Ausgangswerte zu den Eingangs-ADS-Parametern des vorherigen Zeitschritts. Die Simulation kann erheblich verlangsamt werden, wenn diese Option aktiviert wird.
Create signal labels	Es werden Signalbezeichnungen für die an den Ausgangsports angeschlossenen Signalleitungen erzeugt

Codegenerierung

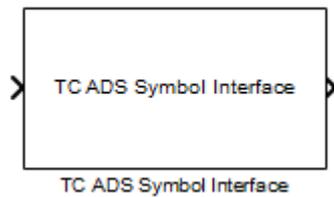
Die Codegenerierung aus diesem Block mit Hilfe des „Simulink Coder“ ist nur mit dem TwinCAT Target (TE1400) möglich.

2.4.2 Synchrone Blöcke



2.4.2.1 TC ADS Symbol Interface

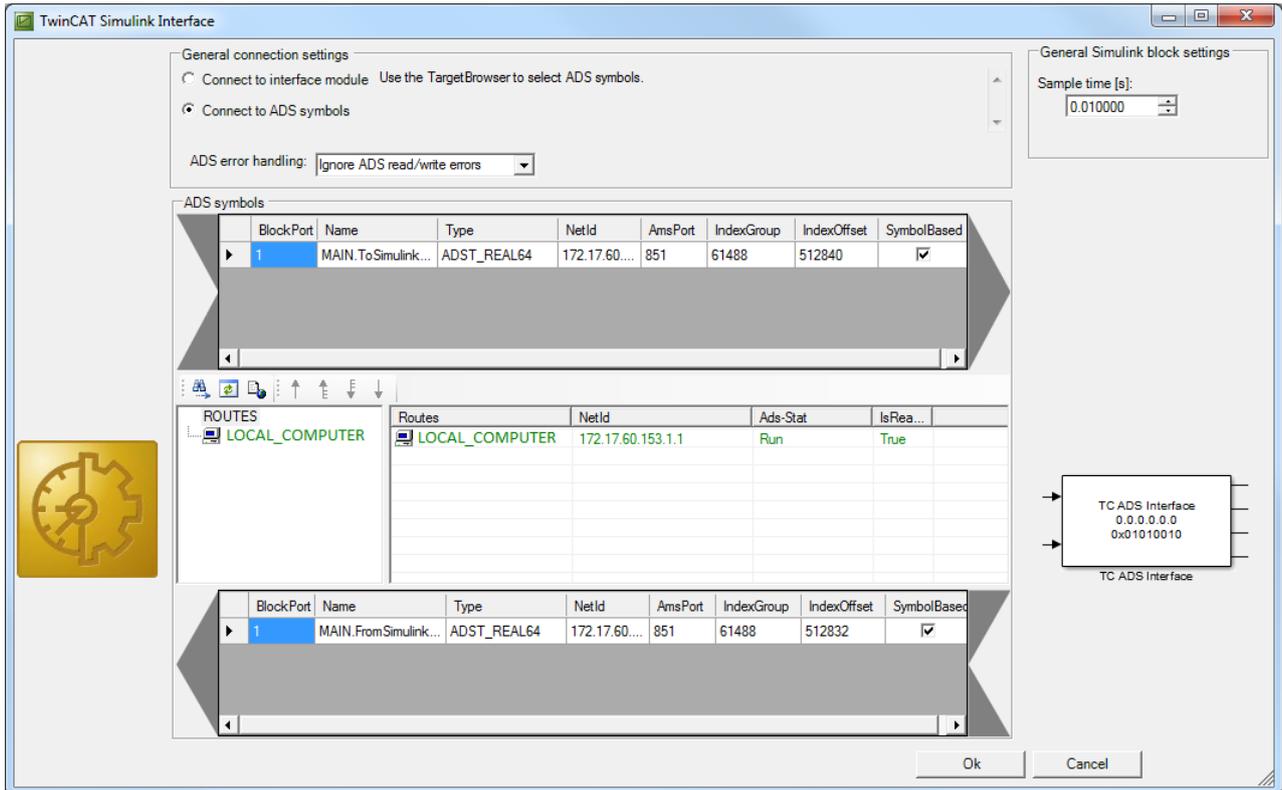
Der Baustein "TC ADS Symbol Interface" erlaubt synchronen Lese- und Schreibzugriff auf TwinCAT ADS Variablen über deren Symbolnamen oder alternativ per ADS IndexGroup und ADS IndexOffset. Der Block ist Bestandteil der Bibliothek "Beckhoff/TwinCAT ADS/Synchronous" im "Simulink Library Browser".



Ein- und Ausgangsports des Blocks

Die Portanzahl und die Datentypen der Ports hängen von der Blockkonfiguration ab. Der Block besitzt keine festen Ein- oder Ausgangsports.

Block-Parameter



ADS Symbole
 Hier kann die Konfiguration der Ein- und Ausgangsports des Simulink®-Blocks vorgenommen werden. Mit Hilfe des Target Browsers können ADS-Symbole gefunden und der Symbolliste der gewünschten Übertragungsrichtung zugewiesen werden. Dies kann entweder über die entsprechenden Schaltflächen des Target Browsers oder mittels Drag&Drop erfolgen.

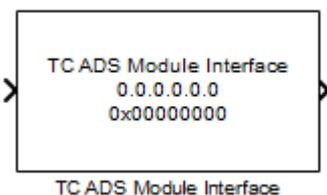
Allgemeine Block-Parameter	
Sample time	Die Abtastzeit des Simulink®-Bausteins. Sie definiert den zeitlichen Abstand der ADS-Anfragen bezogen auf die Simulink®-Zeit (nicht die Echtzeit)

Codegenerierung

Die Codegenerierung aus diesem Block mit Hilfe des „Simulink Coder“ ist nur mit dem TwinCAT Target (TE1400) möglich. Das erzeugte TwinCAT Modul erhält zusätzliche Datenbereiche (Prozessabbilder), welche die Ein- und Ausgangsvariablen des Simulink®-Bausteins enthalten.

2.4.2.2 TC ADS Module Interface

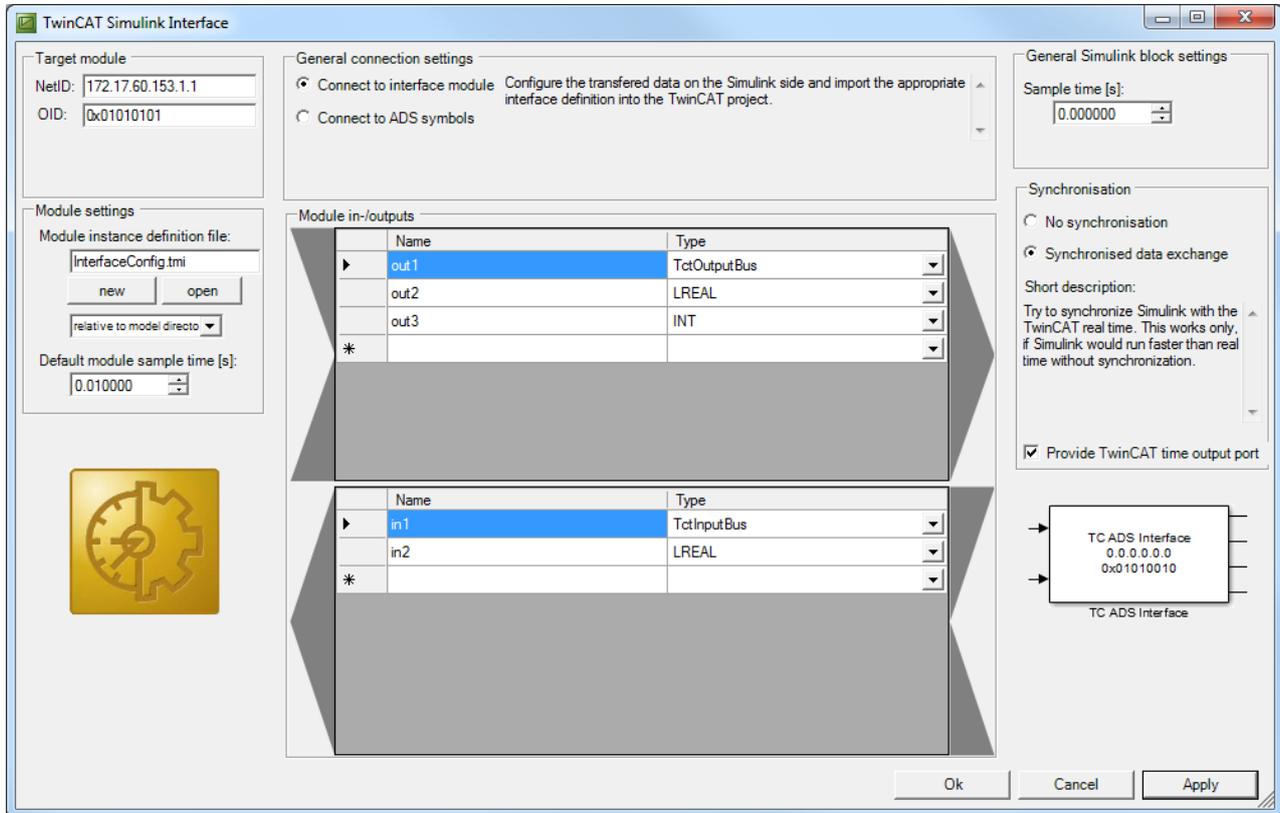
Der Baustein "TC ADS Module Interface" erlaubt synchronen Lese- und Schreibzugriff auf ein spezielles TwinCAT Modul. Der Block ist Bestandteil der Bibliothek "Beckhoff/TwinCAT ADS/Synchronous" im "Simulink Library Browser".



Ein- und Ausgangsports des Blocks

Die Portanzahl und die Datentypen der Ports hängen von der Blockkonfiguration ab. Der Block besitzt keine festen Ein- oder Ausgangsports.

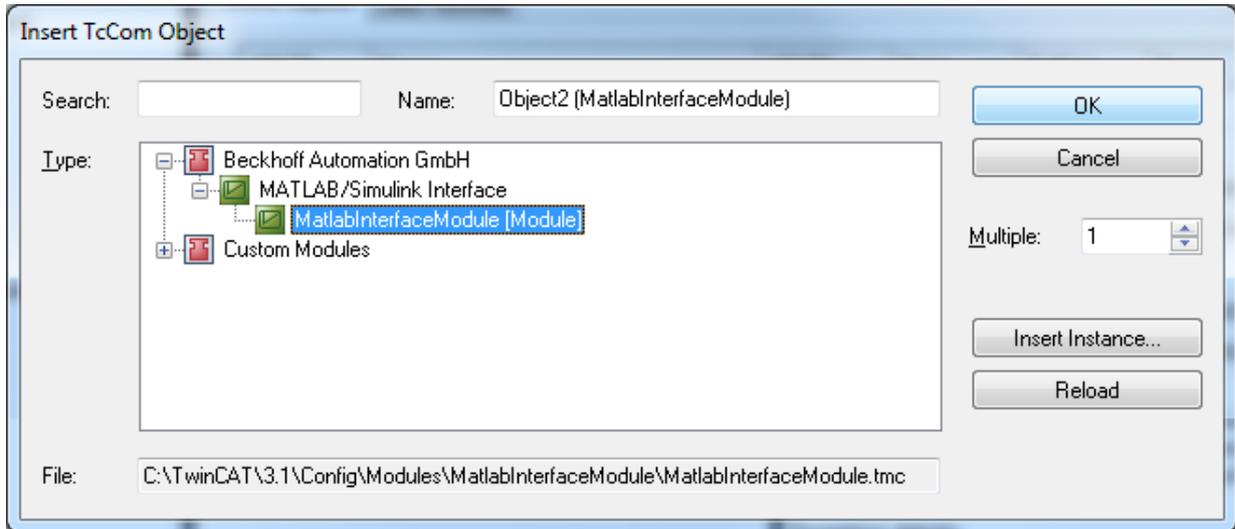
Block-Parameter



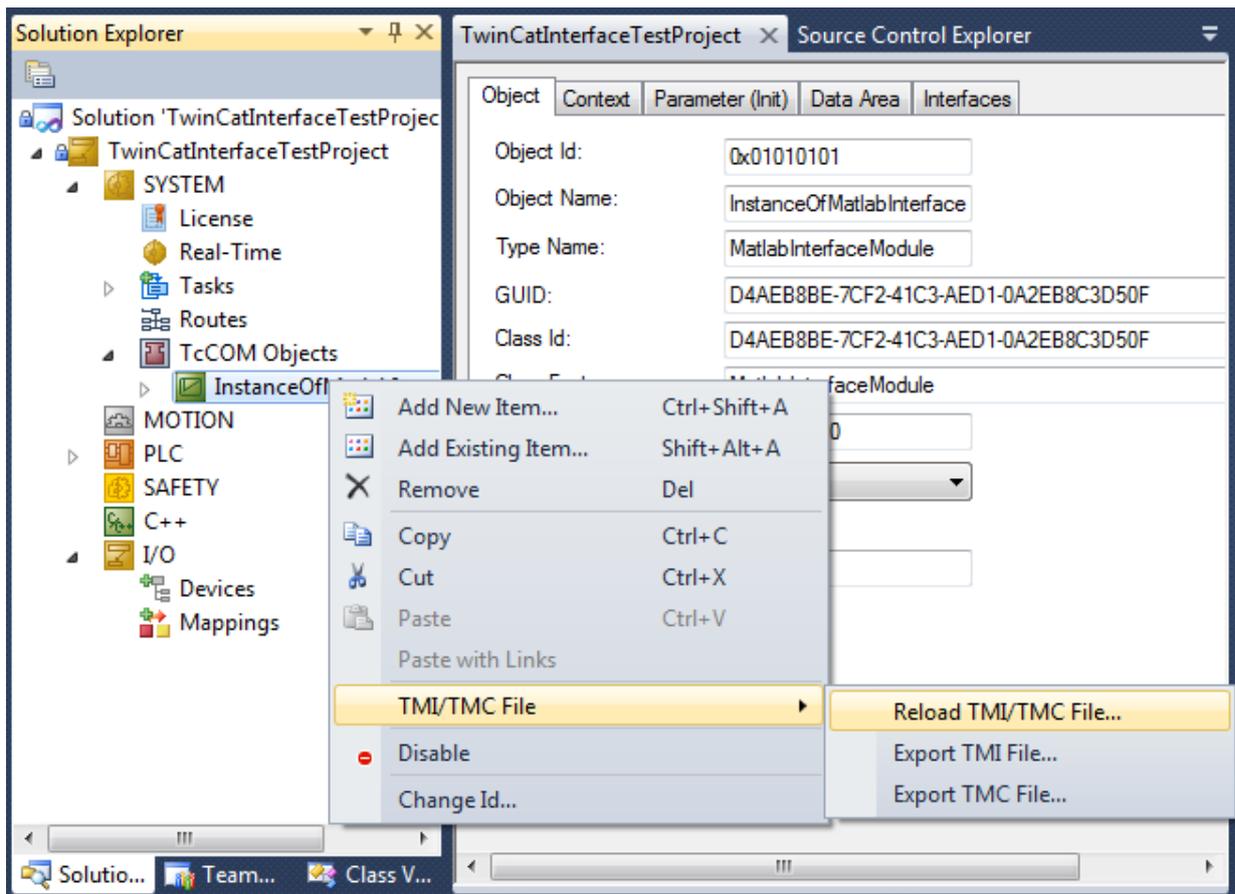
Zielmodul	
NetId	Die NetId des Zielsystems, auf dem das Schnittstellenobjekt (Instanz des Schnittstellenmoduls) läuft.
OID	Die Objekt-ID des Schnittstellenobjektes
Moduleinstellungen	
Module instance definition file	TMI-Datei (TwinCAT Modulinstanz), welche die Konfiguration des Schnittstellenobjektes enthält, erzeugen oder öffnen. Diese Datei kann in das TwinCAT Projekt importiert werden.
Default module sample time	Die standardmäßige Abtastzeit des Schnittstellenmoduls.
Modul-Ein-/Ausgänge	
Ein- und Ausgangsports des Simulink®-Bausteins definieren. Diese Einstellungen werden in die ausgewählte TMI- Datei (TwinCAT Modulinstanz) gespeichert. Die Signale können jedem Basistyp entsprechen oder ein Simulink® Busobjekt sein, das im MATLAB Arbeitsbereich definiert ist.	
Allgemeine Block-Parameter	
Sample time	Die Abtastzeit des Simulink®-Bausteins. Sie definiert den zeitlichen Abstand der ADS-Anfragen bezogen auf die Simulink®-Zeit (nicht die Echtzeit)
Synchronisation	
No Synchronisation / Synchronised data exchange	Die Zeit seit dem Start der Simulation wird vom Schnittstellenobjekt an Simulink® übermittelt und wird bei der Einstellung „Synchronised data exchange“ zur Synchronisation der Simulationszeit auf die Echtzeit verwendet.
Provide TwinCAT time output port	Stellt die Zeit seit dem Start der Simulation als Ausgangsport zur Verfügung.

Schnittstellenmodulkonfiguration

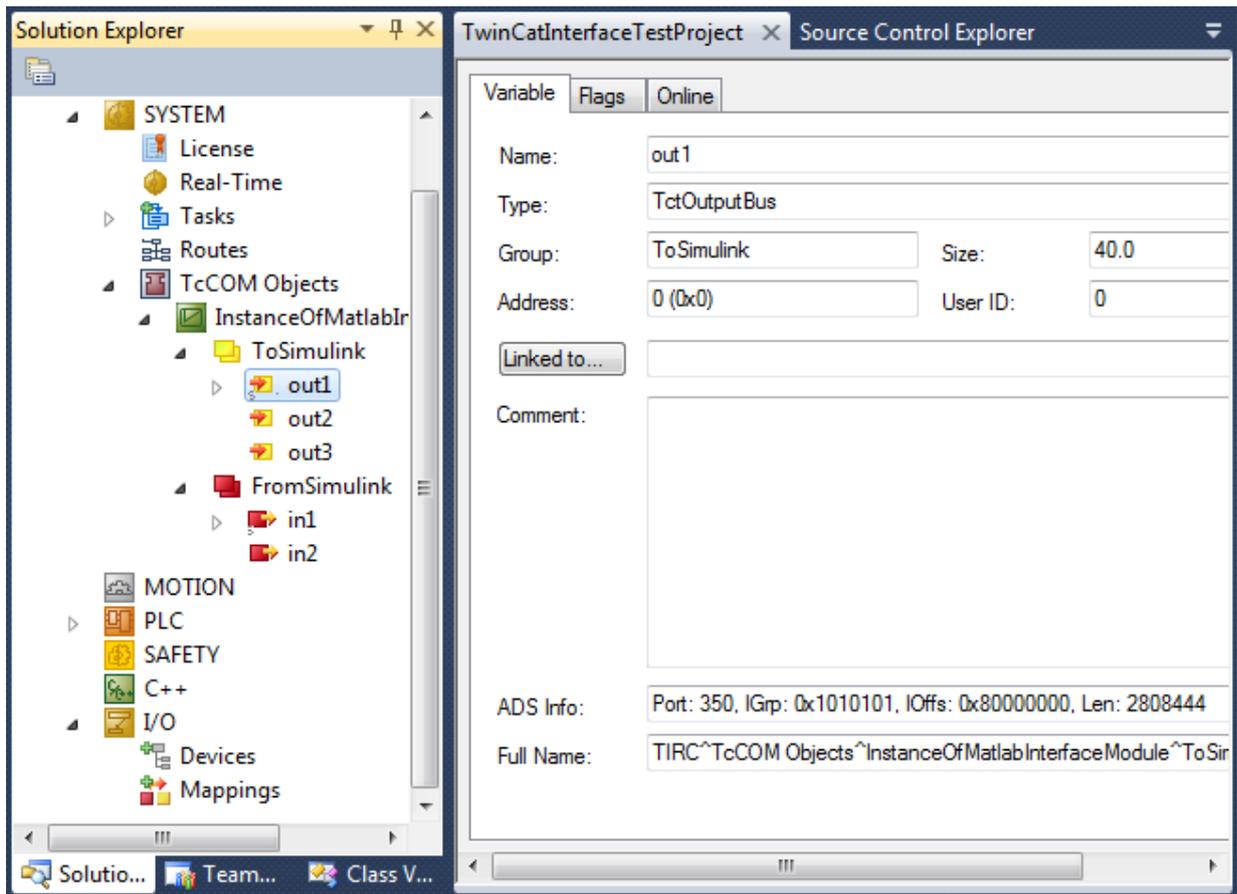
1. Öffnen Sie das TwinCAT Projekt, mit dem die Verbindung hergestellt werden soll
2. Erzeugen Sie eine neue Instanz des **MatlabInterfaceModule** am TwinCAT Projektknoten "**System / TcCom Objects**" über dessen Kontextmenüpunkt "**Neues Element hinzufügen...**"



3. Öffnen Sie im Kontextmenü des neu erzeugten Objektknotens den Eintrag **TMI-Datei neu laden**. Öffnen Sie die TMI-Datei, die über die Blockkonfiguration in Simulink® erstellt wurde.



4. Verbinden Sie die Prozessabbildvariablen des Schnittstellenobjektes mit den gewünschten Prozessabbildern der anderen TwinCAT Module (PLC, C++, MATLAB®/ Simulink®, IOs). Der Eingangsbereich der Objekte enthält die Ausgänge des Simulink-Blocks und umgekehrt.

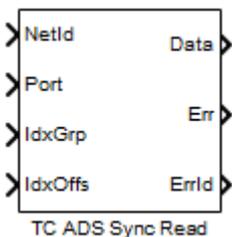


Codegenerierung

Die Codegenerierung aus diesem Block mit Hilfe des „Simulink Coder“ ist nur mit dem TwinCAT Target (TE1400) möglich. Das erzeugte TwinCAT Modul erhält zusätzliche Datenbereiche (Prozessabbilder), welche die Ein- und Ausgangsvariablen des Simulink®-Bausteins enthalten.

2.4.2.3 TC ADS Sync Read

Der Block "TC ADS Sync Read" erlaubt synchronen Lesezugriff auf TwinCAT ADS Variablen über deren ADS IndexGroup und ADS IndexOffset. Sie finden diesen Baustein in der Bibliothek "Beckhoff/TwinCAT ADS/Synchronous" im "Simulink Library Browser".



Eingangsports des Blocks

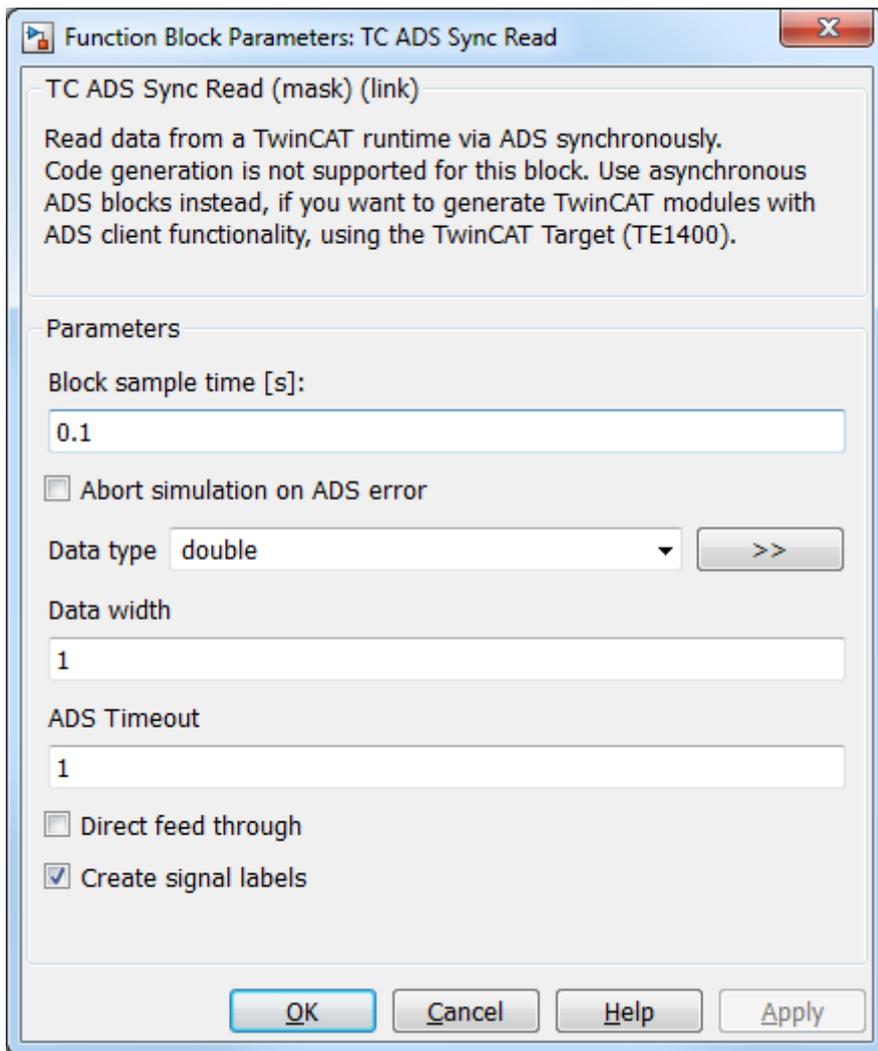
NetId	AMS NetId
Port	ADS Port
IdxGrp	ADS Index Group

IdxOffs	ADS Index Offset
----------------	------------------

Ausgangsports des Blocks

Daten	Gelesener Wert der ADS-Variable
Err	TRUE, wenn ein ADS-Fehler vorliegt
ErrId	ADS-Fehlercode

Block-Parameter



Block sample time	Die Abtastzeit des Simulink-Blocks
Abort simulation on ADS error	Simulation bei dem ersten ADS Fehler stoppen
Data type	Der Datentyp des <i>Daten</i> -Ausgangssignals. Er muss dem Datentyp der übermittelten ADS Variablen entsprechen
Data width	Anzahl der Elemente bei der Übertragung von Arrays; 1 bei skalaren Werten.
ADS Timeout	Simulink wartet auf Antwort der letzten ADS Anforderung bis diese Zeit überschritten wird
Direkt feed through	Eingangswerte des Blocks werden nicht in der Update- sondern der Output-Methode ausgewertet. Dadurch wird sichergestellt, dass die Werte an den Ausgangsports im aktuellen Zeitschritt zu den ADS Parametern an den Eingangsports desselben Zeitschritts gehören.

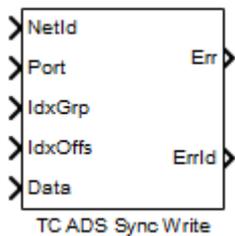
	Ansonsten gehören die Ausgangswerte zu den Eingangs-ADS-Parametern des vorherigen Zeitschritts. Die Simulation kann erheblich verlangsamt werden, wenn diese Option aktiviert wird.
Create signal labels	Es werden Signalbezeichnungen für die an den Ausgangsports angeschlossenen Signalleitungen erzeugt

Codegenerierung

Die Codegenerierung aus diesem Block mit Hilfe des „Simulink Coder“ wird derzeit nicht unterstützt. In einem Modell, aus dem mit Hilfe des TE1400 ein TwinCAT-Modul erzeugt werden soll, sollten stattdessen die asynchronen ADS-Blöcke verwendet werden.

2.4.2.4 TC ADS Sync Write

Der Block "TC ADS Sync Write" erlaubt synchronen Lesezugriff auf TwinCAT ADS Variablen über deren ADS IndexGroup und ADS IndexOffset. Sie finden diesen Baustein in der Bibliothek "Beckhoff/TwinCAT ADS/Synchronous" im "Simulink Library Browser".



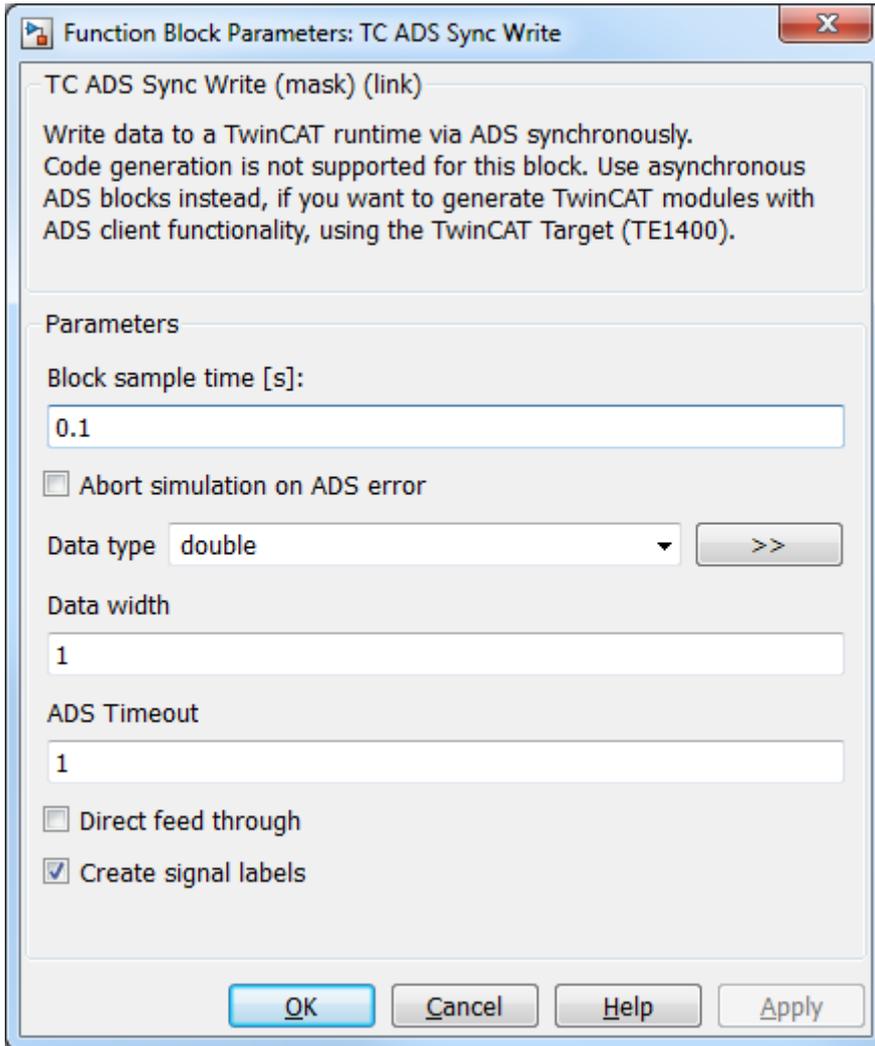
Eingangsports des Blocks

NetId	AMS NetId
Port	ADS Port
IdxGrp	ADS Index Group
IdxOffs	ADS Index Offset
Daten	Zu schreibender Wert der ADS-Variable

Ausgangsports des Blocks

Err	TRUE, wenn ein ADS-Fehler vorliegt
ErrId	ADS-Fehlercode

Block-Parameter



Block sample time	Die Abtastzeit des Simulink-Blocks
Abort simulation on ADS error	Simulation bei dem ersten ADS Fehler stoppen
Data type	Der Datentyp des <i>Daten</i> -Ausgangssignals. Er muss dem Datentyp der übermittelten ADS Variablen entsprechen
Data width	Anzahl der Elemente bei der Übertragung von Arrays; 1 bei skalaren Werten.
ADS Timeout	Simulink wartet auf Antwort der letzten ADS Anforderung bis diese Zeit überschritten wird
Direkt feed through	Eingangswerte des Blocks werden nicht in der Update- sondern der Output-Methode ausgewertet. Dadurch wird sichergestellt, dass die Werte an den Ausgangsports im aktuellen Zeitschritt zu den ADS Parametern an den Eingangsports desselben Zeitschritts gehören. Ansonsten gehören die Ausgangswerte zu den Eingangs-ADS-Parametern des vorherigen Zeitschritts. Die Simulation kann erheblich verlangsamt werden, wenn diese Option aktiviert wird.
Create signal labels	Es werden Signalbezeichnungen für die an den Ausgangsports angeschlossenen Signalleitungen erzeugt

Codegenerierung

Die Codegenerierung aus diesem Block mit Hilfe des „Simulink Coder“ wird derzeit nicht unterstützt. In einem Modell, aus dem mit Hilfe des TE1400 ein TwinCAT-Modul erzeugt werden soll, sollten stattdessen die asynchronen ADS-Blöcke verwendet werden.

2.4.3 Utilities

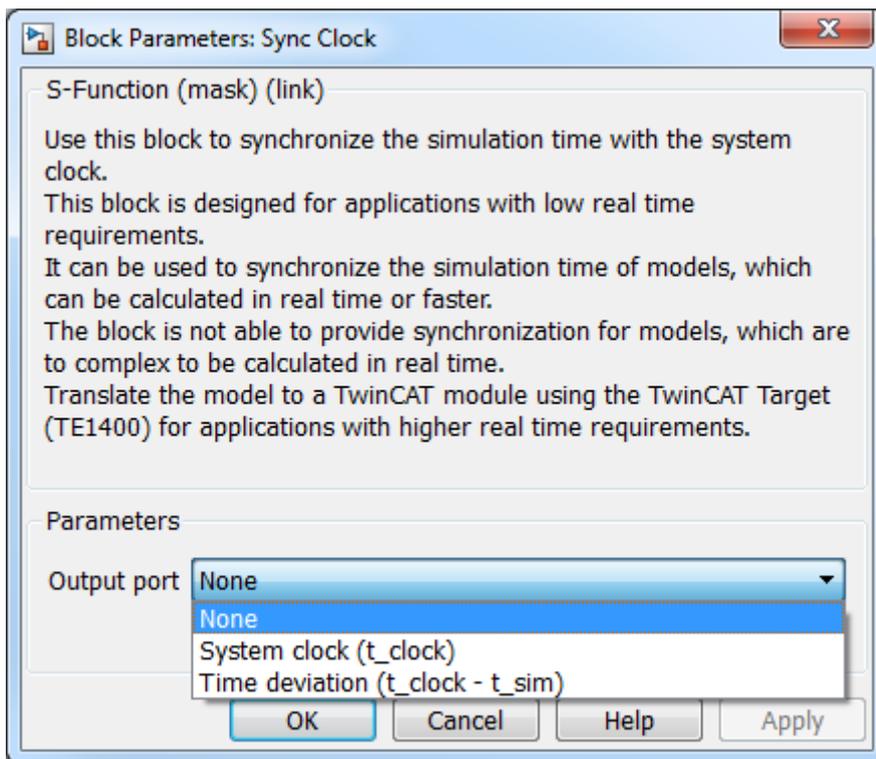
2.4.3.1 Sync Clock

Der Baustein "Sync Clock" kann dazu verwendet werden, die Simulationszeit auf die Systemzeit zu synchronisieren. Sie finden diesen Baustein in der Bibliothek "Beckhoff/TwinCAT ADS/Utilities" im "Simulink Library Browser".



Sync Clock

Block-Parameter



Output port	Aktiviert einen optionalen Baustein-Ausgabeport, der für die Überwachung der Abweichung zwischen der Simulations- und der Systemzeit verwendet werden kann.
--------------------	---

Codegenerierung

Codegenerierung wird vom TwinCAT Target unterstützt. Es wird aber kein Code für diesen Baustein generiert, weil Echtzeitmodule keine Systemzeitsynchronisation benötigen.

3 Ab Version 2.x.xxxx.x

- TE1410 Interface for MATLAB®/Simulink® Versionen geringer als 1.2.xxxx.x unterstützen MATLAB R2010b bis MATLAB R2019a.
- TE1410 Interface for MATLAB®/Simulink® Versionen höher als 2.x.xxxx.x unterstützen MATLAB R2019a und höher.
- Die Installationen für beide Versionen können parallel auf einem Engineering System verwendet werden.
- Kompatibilität: Die Versionen sind nicht kompatibel zueinander. Bei Einrichtung der Version 2.x für eine MATLAB®-Version werden die Funktionen der Version 1.2.x aus dem MATLAB®- Pfad entfernt.

3.1 Übersicht

TE1410 TwinCAT Interface for MATLAB®/Simulink®

Das TwinCAT 3 Interface for MATLAB®/Simulink® ermöglicht den Datenaustausch zwischen MATLAB® und der TwinCAT Runtime sowie zwischen Simulink® und der TwinCAT Runtime.

Für Simulink® werden ADS-Client-Blöcke in der Simulink®-Library zur Verfügung gestellt, welche einfach in Simulink®-Projekten eingebunden und konfiguriert werden können. So ist es möglich, von der Simulink®-Umgebung ausgehend, in einer TwinCAT Runtime Daten zu schreiben oder zu lesen. Ein Anwendungsbeispiel für die Nutzung von ADS-Blöcken in Simulink® ist die Software-in-the-Loop-Simulation, bei der ein in der TwinCAT Runtime ausgeführter Regel- oder Steueralgorithmus mit einem in Simulink® ausgeführten Modell verbunden wird, siehe Beschreibung [ADS in Simulink®](#) [► 30].

Für MATLAB® wird ein ADS-Port-Objekt bereitgestellt, welches in der Skriptumgebung genutzt werden kann. Das Objekt bietet diverse Methoden zum Datenaustausch zwischen MATLAB® und der TwinCAT Runtime an. Neben einem ADS-Client in MATLAB® – d. h. die MATLAB®-Umgebung initiiert den Datenaustausch – ist auch die Nutzung eines ADS-Servers in MATLAB® möglich. Letzteres bietet die Möglichkeit, MATLAB®-Funktionen innerhalb der MATLAB®-Umgebung zu schreiben und diese über ein ADS-Kommando aus TwinCAT aufzurufen. Ein Anwendungsbeispiel für die Nutzung des ADS-Clients in MATLAB® ist die Erstellung einer graphischen Bedien- und Monitoringoberfläche, z. B. auf Basis einer [MATLAB®-App](#). Der ADS-Server kann z. B. genutzt werden, um nicht-echtzeitrelevante Aufgaben, wie Komponenten eines prädiktiven Wartungssystems oder eine Parameteroptimierung, auszuführen. Der [MATLAB® Compiler™](#) kann genutzt werden, um die erstellten Programme als eigenständige Applikationen im Feld, z. B. auf einem Edge Device oder direkt auf dem Industrie-PC, auszuführen, siehe Beschreibung [ADS in MATLAB®](#) [► 27].

Weitere Informationen

- <https://www.beckhoff.com/TE1410>
- <https://www.beckhoff.com/matlab>

3.2 Installation

Systemvoraussetzungen

Es wird unterschieden zwischen dem System, auf dem MATLAB®, Simulink® oder die MATLAB® Compiler Runtime (im Folgenden MATLAB®-PC) ausgeführt wird, und dem System, auf dem die TwinCAT Runtime ausgeführt wird (Laufzeit-PC).

MATLAB-PC

- Windows 10 oder äquivalent
- MATLAB R2019a oder höher
- TwinCAT 3 ADS oder TwinCAT 3 XAE Setup, mindestens Version 3.1.4024.7

Laufzeit-PC

- TwinCAT 3 XAR, mindestens Version 3.0

- Betriebssystem des Laufzeit-PC ist nicht relevant

Installation

Die Installation muss nur auf dem MATLAB®-PC ausgeführt werden.

- Falls kein TwinCAT auf dem System vorhanden ist, installieren Sie TwinCAT 3 ADS oder TwinCAT 3 XAE.
- Starten Sie TwinCAT Tools for MATLAB® and Simulink® Setup zur Installation des TE1410.
- Die Installation des TE1410 erfolgt innerhalb der TwinCAT-Ordnerstruktur, d. h. sie ist losgelöst von der MATLAB®-Installation.
- Falls kein MATLAB® auf dem System vorhanden ist, installieren Sie spätestens jetzt MATLAB®.
- Starten Sie MATLAB® als Administrator und führen Sie `%TwinCAT3Dir%..\Functions\TE14xx-ToolsForMatlabAndSimulink\SetupTE14xx.p` in MATLAB® aus, um die Funktionen aus TE1410 mit dieser MATLAB®-Version zu verknüpfen.

3.3 Lizenzen

Um die Funktionen des TE1410 Interface für MATLAB®/Simulink® nutzen zu können, wird eine **TE1410 Lizenz** auf dem **MATLAB®-PC** benötigt.

Vom MATLAB®-PC ausgehend, können Sie sich mit uneingeschränkt vielen Zielplattformen verbinden. Für die Zielplattformen müssen Sie hinsichtlich der Nutzung mit dem TE1410 keine zusätzlichen Laufzeit-Lizenzen erwerben.

Sollten Sie keine TE1410-Lizenz besitzen, können Sie die Software als Demoversion nutzen.



Keine 7-Tage-Testlizenz

Für dieses Produkt ist keine 7-Tage-Testlizenz mit vollständigem Funktionsumfang verfügbar.

Einschränkungen der Demoversion

Bei Nutzung der Demoversion werden dem Nutzer entsprechende Warnungen angezeigt.

Für [ADS in Simulink®](#) [► 30] gilt: Nach 30 Minuten endet die Simulation mit einer Fehlermeldung. Die Simulation muss neu gestartet werden. Zusätzlich ist die Anzahl der ADS-Symbole zum Lesen oder Schreiben auf 4 limitiert.

Für [ADS in MATLAB®](#) [► 27] gilt: Nach 30 Minuten Laufzeit geht das erstellte Port-Objekt in einen Fehlerzustand. MATLAB® muss neu gestartet werden, um wieder 30 Minuten testen zu können. Eine Limitierung hinsichtlich der ADS-Symbole gibt es in diesem Fall nicht.

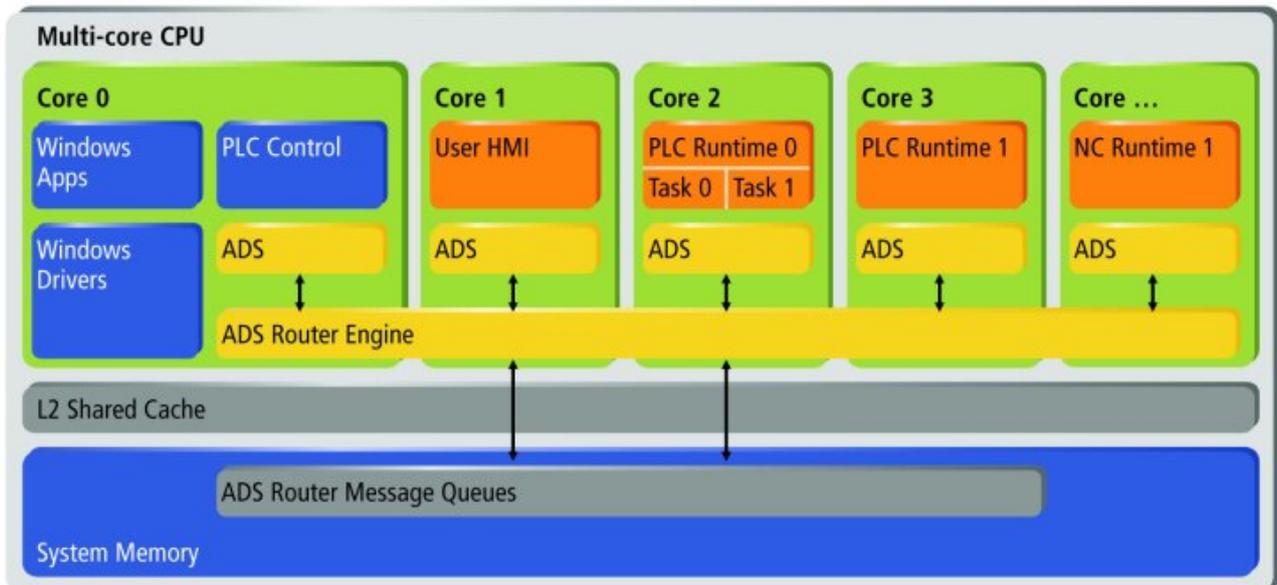
3.4 Kurzeinführung: TwinCAT ADS

Grundstruktur von ADS-Geräten und ADS-Symbolen

Automation Device Specification (ADS) bildet die Grundlage für das Interface for MATLAB®/Simulink®. ADS beschreibt eine geräte- und feldbusunabhängige Schnittstelle und ermöglicht die Kommunikation zwischen ADS-Geräten.

Nachfolgend wird das ADS-Gerätekonzept sowie die Identifikation eines ADS-Geräts erläutert.

Die modulare Systemarchitektur von TwinCAT erlaubt es, die einzelnen Teile der Software (z. B. TwinCAT PLC, TwinCAT NC ...) als eigenständige Geräte zu betrachten: Für jede einzelne Aufgabe gibt es ein Softwaremodul. Die Server im System sind die ausführenden Geräte, welche bestimmte Dienste anbieten. Die Clients sind Programme, welche die Dienste der Server anfordern. Ein Client baut entsprechend initial eine Verbindung zum Server auf und fragt einen Dienst an: Zum Beispiel fragt er lesend den Wert einer Variablen an oder er fragt das Schreiben einer Variablen an.



Das Interface for MATLAB®/Simulink® stellt für Simulink® und MATLAB® eine ADS-Client-Schnittstelle zur Verfügung, mit welcher ein Datenaustausch (schreibend und lesend) mit TwinCAT-Laufzeiten ermöglicht wird. Die TwinCAT-Laufzeit, bzw. ihre ADS-Geräte, stellen damit als ADS-Server ihre Dienste zur Verfügung und können aus LabVIEW™ heraus genutzt werden.

Darüber hinaus stellt das Interface for MATLAB®/Simulink® einen ADS-Server für MATLAB® bereit, sodass eine MATLAB®-Umgebung Dienste für ADS-Clients bereitstellen kann.

Der ADS-Datenaustausch zwischen ADS-Geräten findet über den ADS-Router statt. Wie im obigen Schaubild aufgezeigt, erfolgt der Datenaustausch von ADS-Geräten, welche auf demselben System realisiert sind, über den Systemspeicher. Sind zwei ADS-Geräte, z. B. MATLAB® und die TwinCAT-Laufzeit auf unterschiedlichen Systemen, kann eine Route zwischen zwei ADS-Routern erzeugt werden. Beim Erzeugen der ADS-Route kann der Transporttyp (i.d.R. TCP/IP) für die Kommunikation zwischen den beiden ADS-Routern definiert werden. Entsprechend identifiziert sich ein ADS-Gerät durch die AMS-NetId des ADS-Routers und durch eine Portnummer, welche dann das ADS-Gerät auf dem System spezifiziert. Beispielsweise ist der Port 851 der Default-Port für die erste PLC-Instanz in der TwinCAT-Laufzeit. ADS-Dienste eines ADS-Geräts werden dann durch zwei Parameter, Index Group und Index Offset, spezifiziert. Zum Beispiel ist eine SPS-Variable zum Lesen oder Schreiben unter einem bestimmten Index Group und Index Offset erreichbar.

Zusammenfassung

- AMS NetId: Identifiziert den ADS-Router, also das System.
- ADS-Route: Spezifiziert die Verbindung zwischen zwei ADS-Routern.
- Port: Identifiziert ein ADS-Gerät.
- Index Group/Offset: Spezifiziert den ADS-Systemdienst, z. B. eine Variable zum Lesen und Schreiben.

Um die Adressierung von Variablen in einer TwinCAT-Laufzeit für den Anwender komfortabler zu gestalten, erstellt TwinCAT ADS-Symbole, welche z. B. mit dem Target Browser durchsucht werden können. Der Target Browser ist auch im Interface for MATLAB®/Simulink® im TC Symbol Interface integriert, sodass ADS-Symbole einfach und schnell selektiert werden können. Ein ADS-Symbol für eine Variable in TwinCAT enthält dann die oben genannten Informationen: AMS NetId, Port, Index Group und Index Offset und darüber hinaus die Bit Size sowie einen Symbolnamen und den Datentyp der Variablen.

Weitere Informationen zu ADS finden Sie unter folgenden Links:

- AMS NetId und Port: [ADS device identification](#)
- Index Group und Index Offset: [Specification for ADS devices](#)
- ADS Routen: [System Node "Routes"](#) und [Add Route](#)
- Connecting Devices with same AMS NetId: [AmsNAT](#)
- Using MQTT and Message Broker with ADS: [ADS-over-MQTT](#)

- TLS encrypted ADS: [Secure ADS](#)
- Target Browser: [Target Browser](#)
- ADS Kommunikation aufzeichnen: [ADS Monitor](#)

Grundlegende ADS-Datenkommunikation

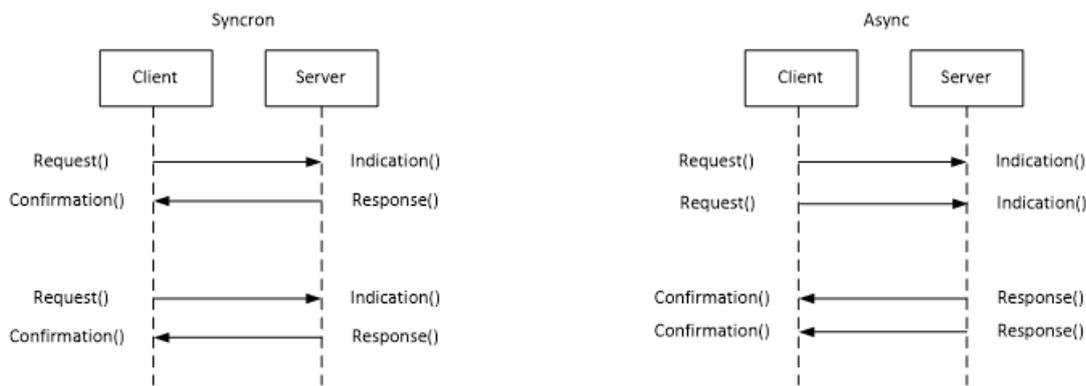
Grundlegend gibt es bei ADS drei unterschiedliche Möglichkeiten/Modi, Daten zu kommunizieren:

Synchrones Lesen oder Schreiben

- Der ADS-Client wartet auf Antwort des ADS-Servers, bevor mit der Ausführung des Codes fortgeschritten wird.

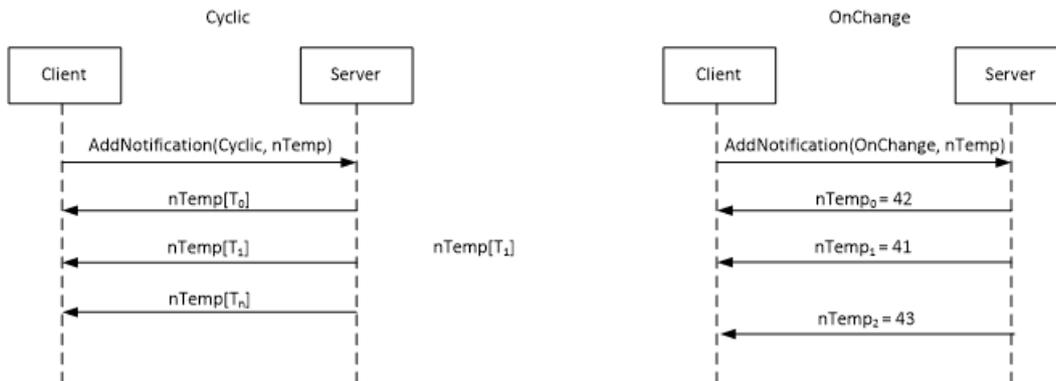
Asynchrones Lesen oder Schreiben

- Der ADS-Client sendet eine Lese- oder Schreibfrage an den ADS-Server, wartet aber nicht auf Antwort, sondern führt weitere Programmteile (z. B. Anfragen an andere Variablen) weiter aus.



Eventbasierte Kommunikation (nur Lesen)

- Es wird nur einmalig eine sogenannte ADS-Notification am Server angemeldet. Die Notification kann „on change“ oder „cyclic“ angemeldet werden. Nach Anmeldung einer Notification am Server sendet dieser ohne weitere Anfragen des Clients an den Server.



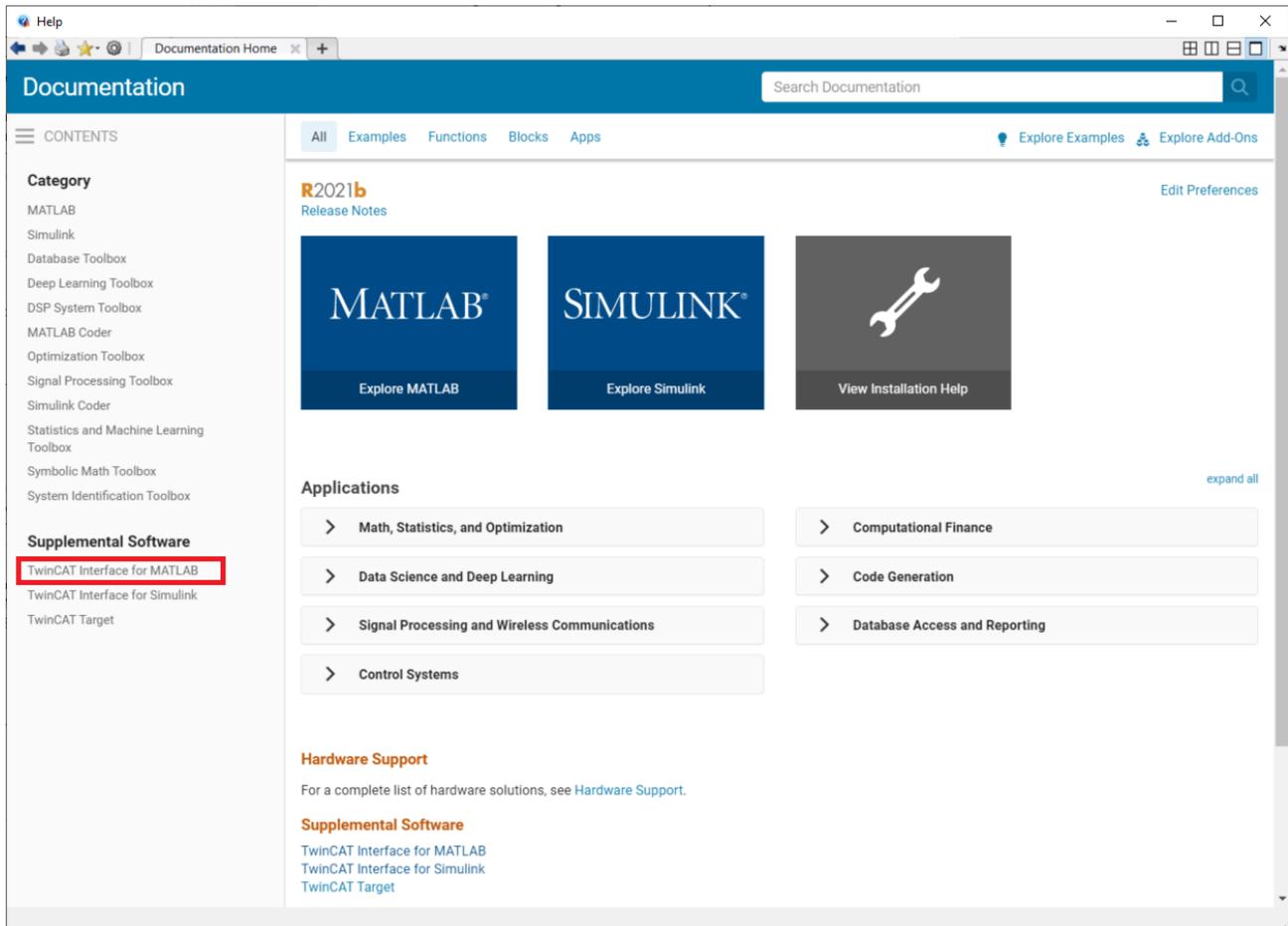
3.5 ADS in MATLAB®

3.5.1 API-Dokumentation

Dokumentation in der MATLAB®-Hilfe

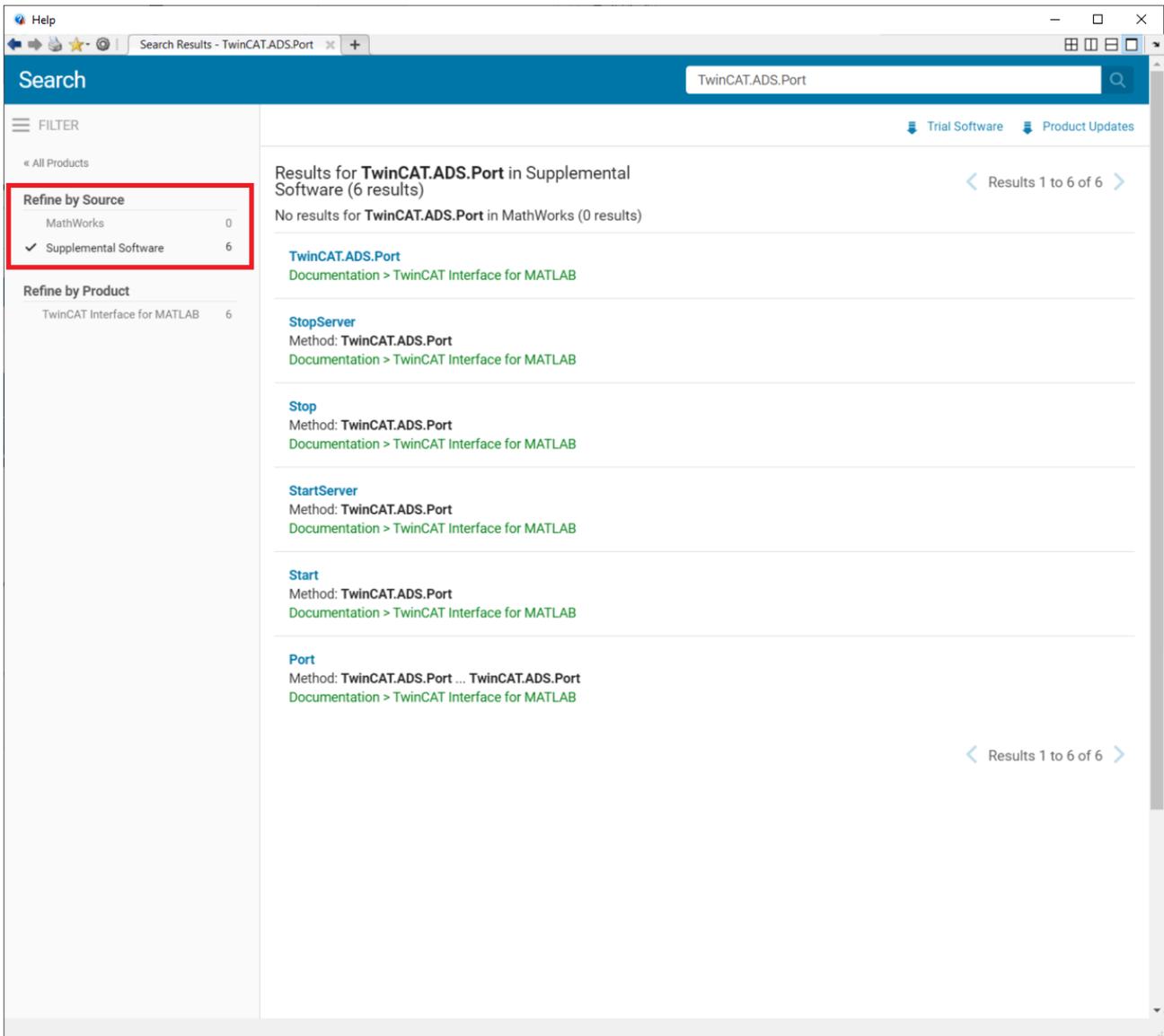
- ✓ Die API ist vollständig in der MATLAB®-Hilfe dokumentiert.

1. Rufen Sie die MATLAB®-Hilfe auf und wählen Sie unter **Supplemental Software** den Eintrag **TwinCAT Interface for MATLAB** aus.



The screenshot shows the MATLAB documentation website. The left sidebar contains a 'CONTENTS' menu with a 'Supplemental Software' section. Under this section, 'TwinCAT Interface for MATLAB' is highlighted with a red box. The main content area features a search bar, navigation tabs for 'All', 'Examples', 'Functions', 'Blocks', and 'Apps', and a 'R2021b Release Notes' section. Below this, there are three large buttons: 'Explore MATLAB', 'Explore Simulink', and 'View Installation Help'. The 'Applications' section lists various categories like 'Math, Statistics, and Optimization', 'Computational Finance', 'Data Science and Deep Learning', 'Code Generation', 'Signal Processing and Wireless Communications', 'Database Access and Reporting', and 'Control Systems'. The 'Hardware Support' and 'Supplemental Software' sections are also visible at the bottom.

⇒ Die TE1410-Dokumentation ist auch über die Suchfunktion von MATLAB® erreichbar.



Intellisense in MATLAB®

Während der Programmierung in MATLAB® werden Ihnen Hilfestellungen angezeigt, um die Übergabeparameter direkt zu erfassen und nicht jedes Mal in der Hilfe suchen zu müssen.

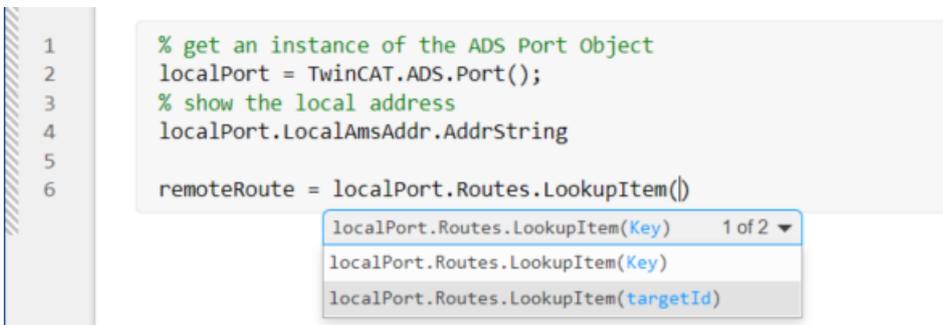


Abb. 1:

```

1  % get an instance of the ADS Port Object
2  localPort = TwinCAT.ADS.Port();
3  % show the local address
4  localPort.LocalAmsAddr.AddrString
5
6
7
8
9  localPort.Write(
    localPort.Write(vAdsVarAddress,vWriteData,timeout_ms)
    e.g. {'Local' 851 'MAIN.counter'}, 'Local:851/MAIN.counter', {'Loc...
    e.g. {'Local' 851 'MAIN.counter'}, 'Local:851/MAIN.counter', {'Local' 851 hex2dec('80000000') 456}

```

Diese Funktionalität ist für alle unterstützten MATLAB® Releases bei Nutzung von MATLAB® Live Scripts verfügbar. Bei Nutzung von MATLAB® Scripts wird mindestens eine MATLAB Version R2021b vorausgesetzt.

3.5.2 Grundkonzepte

Die Grundkonzepte zur Verwendung des ADS-Port-Objekts in MATLAB® werden in unterschiedlichen Beispielen [\[► 40\]](#) aufgegriffen.

Eine Einführung in die Nutzung eines ADS-Clients in MATLAB® erhalten Sie durch folgendes Beispiel:

```
TwinCAT.ADS.Samples.Start('ADS Client Basics')
```

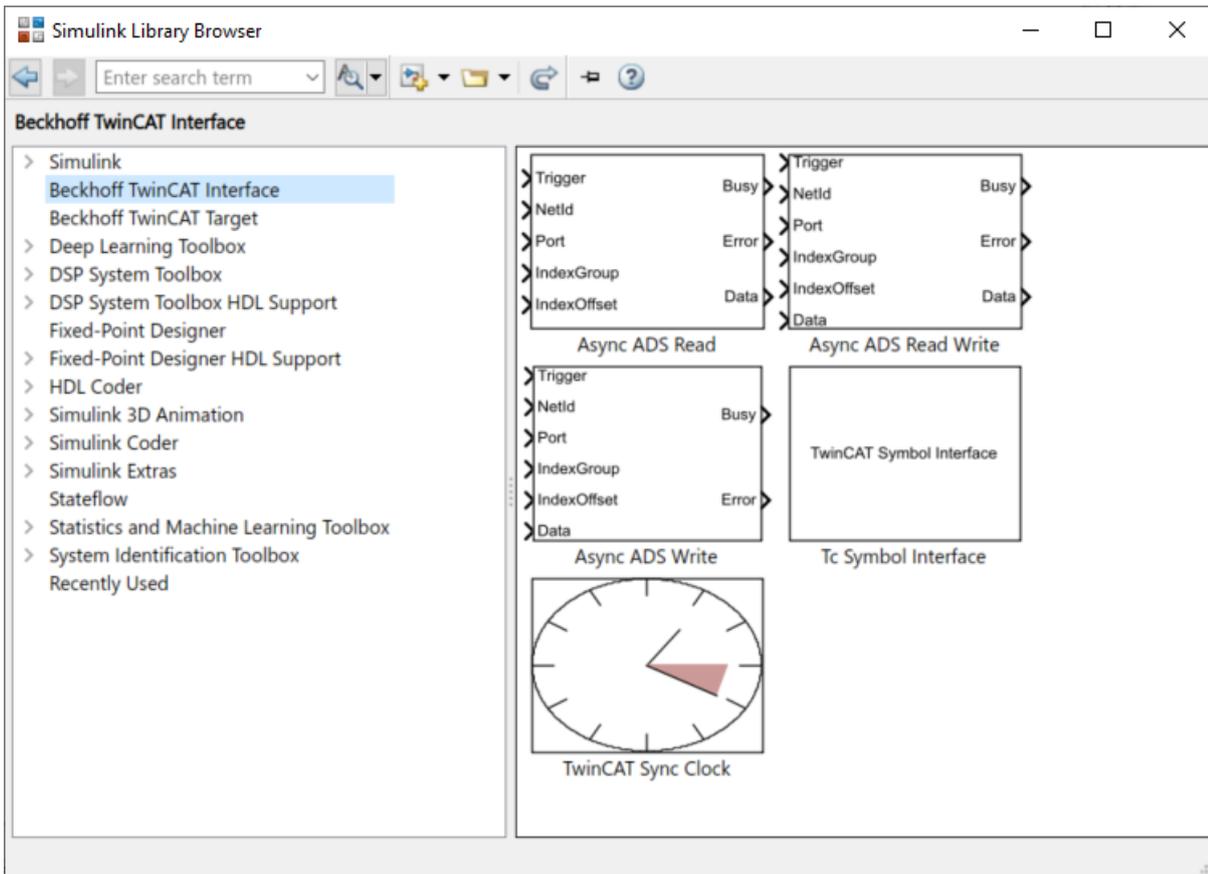
Eine Einführung in die Nutzung eines ADS-Servers in MATLAB® erhalten Sie durch folgendes Beispiel:

```
TwinCAT.ADS.Samples.Start('ADS Server Basics')
```

3.6 ADS in Simulink®

3.6.1 ADS-Blöcke

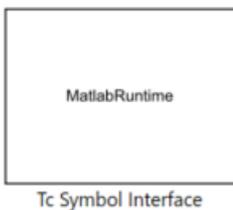
Die ADS-Blöcke in Simulink® können wie gewohnt über den Library Browser eingefügt werden.



Synchrone ADS-Kommunikation zwischen Simulink® und TwinCAT ist auch möglich. Siehe folgendes Beispiel: `TwinCAT.ADS.Samples.Start('ADS Client Simulink Simulation')`

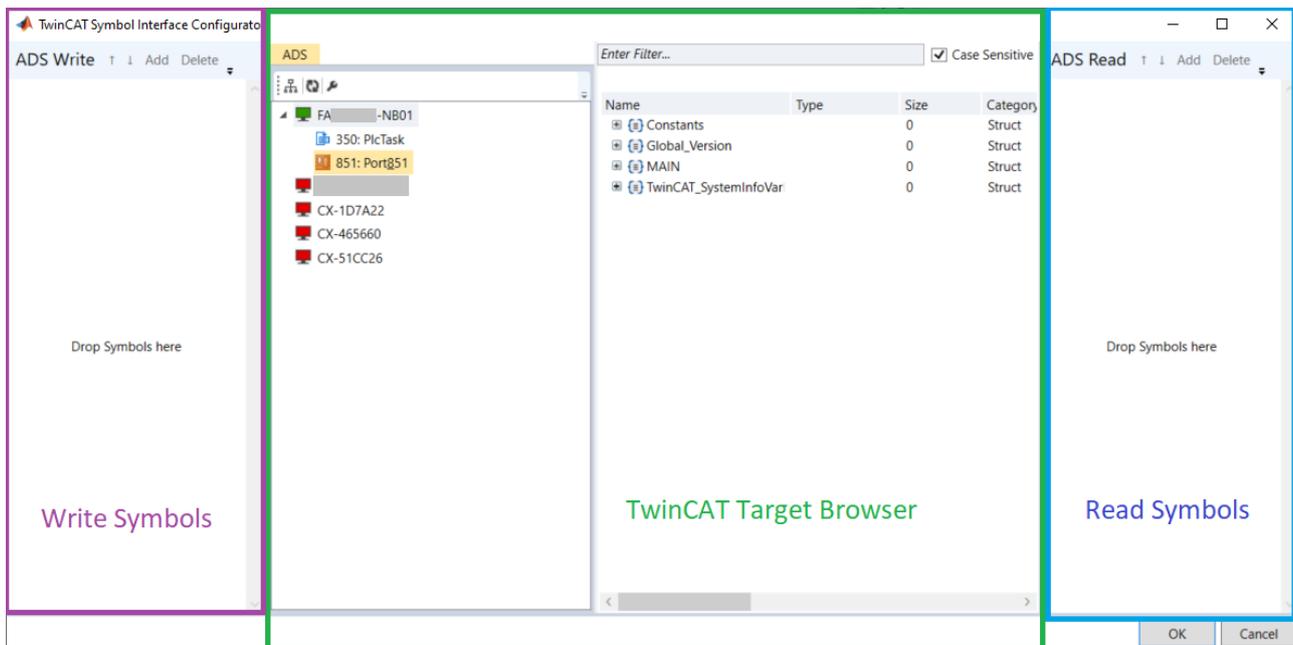
3.6.1.1 Tc Symbol Interface

Tc Symbol Interface ist der zentrale Baustein zur Kommunikation zwischen Simulink® und TwinCAT via ADS. Die Konfiguration der zu lesenden und zu schreibenden ADS-Symbole erfolgt über den integrierten Target Browser.



TwinCAT Symbol Interface Configurator

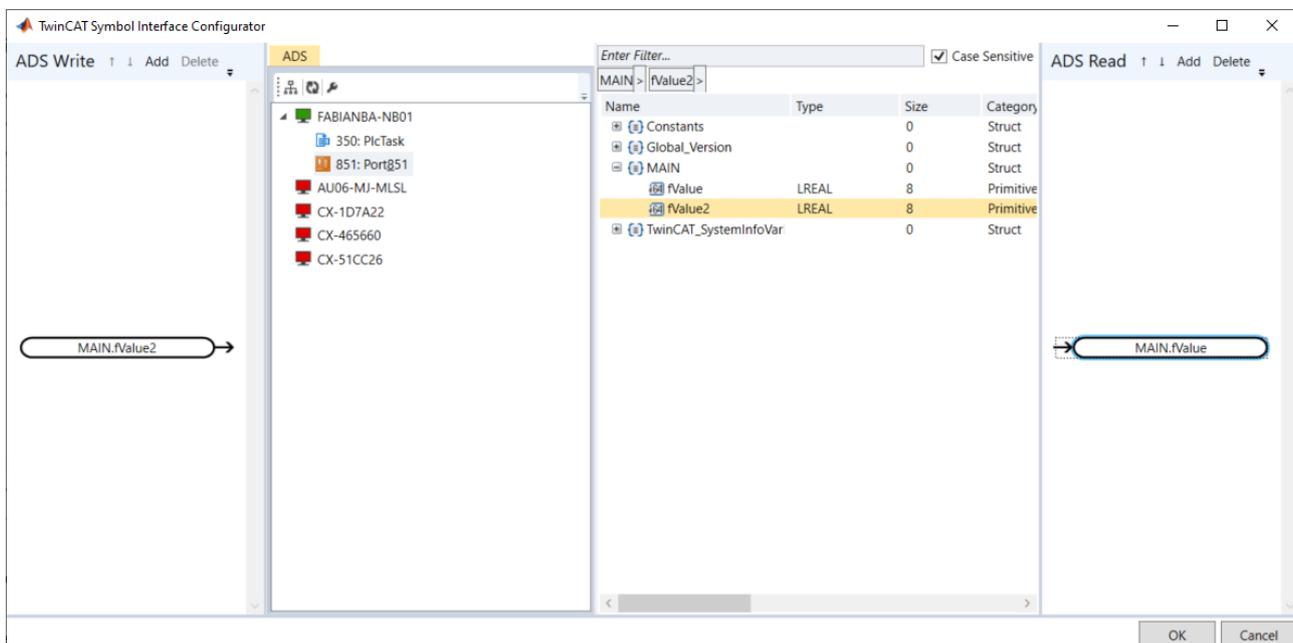
Ohne Konfiguration existieren am Block keine Eingangs- oder Ausgangsports. Diese werden über den TwinCAT Symbol Interface Configurator zunächst vom Nutzer konfiguriert. Der Configurator öffnet sich durch Doppelklick auf den Block.



Im mittleren Teil wird der TwinCAT Target Browser dargestellt. Zu dessen Verwendung siehe [Dokumentation Scope View \(Extension ADS\)](#).

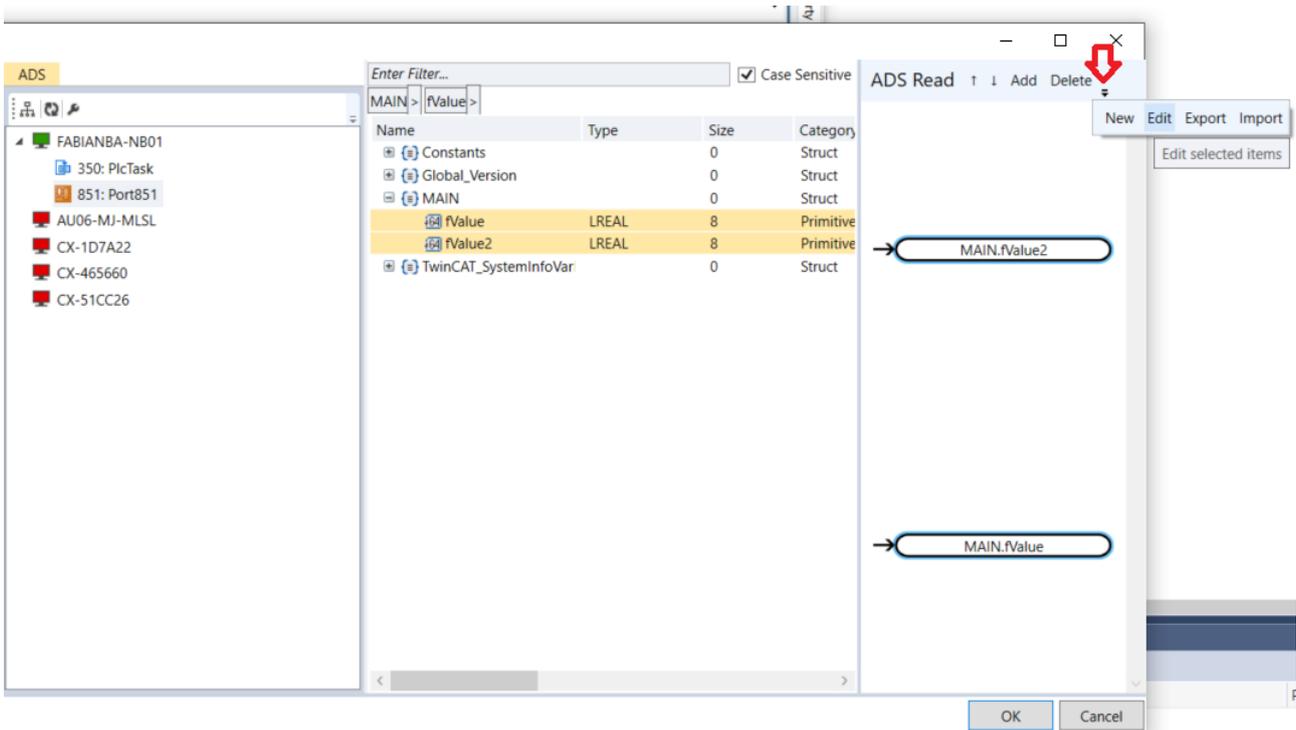
Mit dem TwinCAT Target Browser können Sie in die mit Routes verbundenen Zielsysteme hineinbrowsen und die ADS-Symbole durchsuchen.

Um Symbole zum Lesen oder zum Schreiben zu selektieren (Multi-Select ist möglich), müssen Sie lediglich die Symbole per Drag-and-Drop nach rechts zum Lesen (blauer Kasten) und nach Links zum Schreiben (lila Lasten) ziehen.



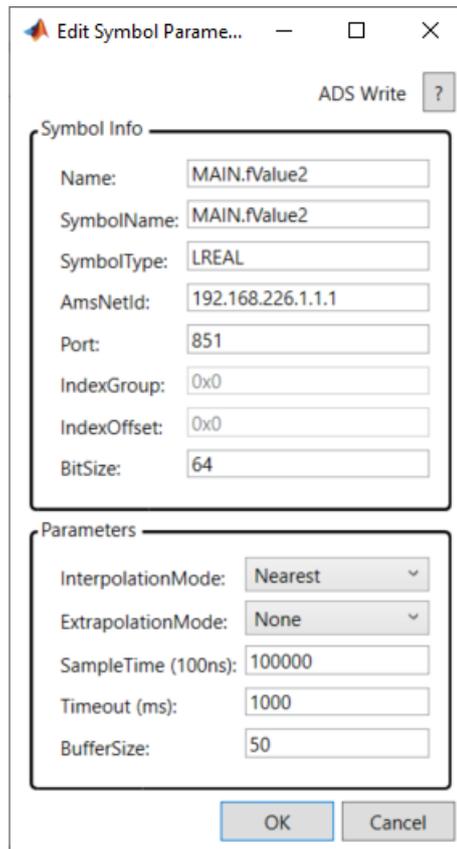
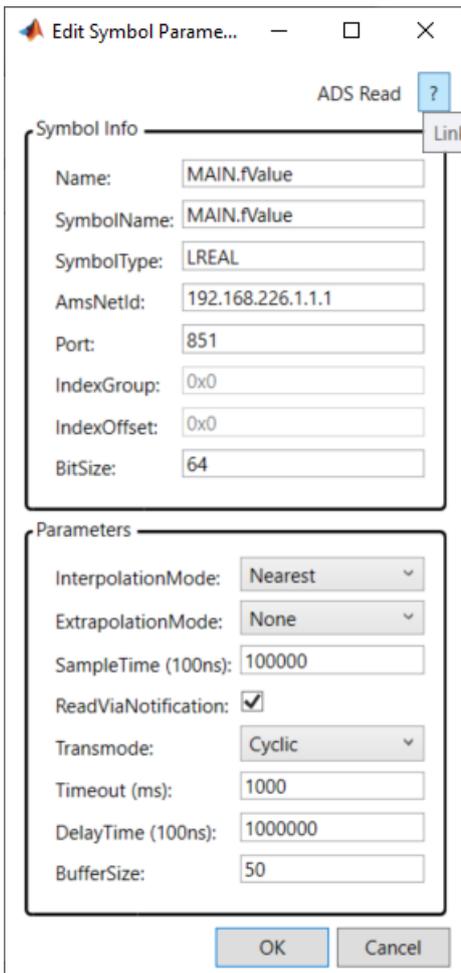
Für jedes ADS-Symbol erscheint dann im linken und rechten Bereich ein Knoten. Durch Doppelklick auf einen Knoten können Sie die Lese- bzw. Schreibigenschaften des Knotens präzisieren.

Sie können auch mehrere Knoten markieren (Ctrl+Click) und über die Schaltfläche **Edit** alle selektierten Knoten gleichzeitig bzgl. der Lese- oder Schreibigenschaften editieren.



Ebenso können Sie die Konfiguration hier exportieren und wieder importieren.

Durch Doppelklick auf einen Knoten oder durch Anwählen der Schaltfläche **Edit** öffnet sich ein neues Fenster.



Im oberen Bereich sind die Symbolinformationen eingetragen. Diese sind für die Art des Datentransports nicht relevant. Bei Multi-Select ist dieser Bereich ausgegraut, da er nur einem Symbol und nicht mehreren gleichzeitig zugeordnet werden kann. Im unteren Bereich können Sie die Eigenschaft des Lesens und Schreibens präzisieren.

Symbole Lesen

Parameter	Default-Wert	Beschreibung
Sample Time (100ns)	TwinCAT-Task Zykluszeit	Periodendauer, mit der das ADS-Symbol gelesen werden soll.
Read Via Notification	Ja	Ja: ADS Notification wird angemeldet. Nein: Lesen per Polling mit SampleTime
Transmode	Cyclic	Nur für Notifications. Cyclic: Wert wird mit SampleTime an Simulink® übertragen. OnChange: Wert wird an Simulink® übertragen, wenn der Wert sich ändert.
Timeout (ms)	1000	ADS Timeout-Zeit
DelayTime (100ns)	10*SampleTime	Buffergröße auf TwinCAT-XAR-Seite. Default sind 10 Werte des ADS-Symbols.
BufferSize	50	Buffergröße in gelesenen Samples auf Simulink®-Seite.
Interpolation Mode	Nearest	None: Keine Interpolation Previous: Nutzt den vorherigen Wert. Next: Nutzt den nächsten bekannten Wert. Nearest: Nutzt den vorherigen oder nächsten Wert, je nachdem, welcher näher ist [*].
Extrapolation Mode	None	None: Keine Extrapolation Constant: Neuer Wert wird zum zuletzt bekannten Wert gesetzt.

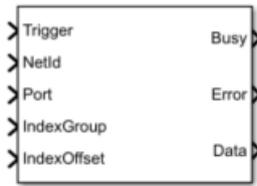
[*] Wenn die Simulink® SampleTime nicht mit der TwinCAT-Task Zykluszeit eines ADS-Symbols übereinstimmt, kann es sein, dass Werte für einen bestimmten Zeitstempel nicht verfügbar sind. Weiter kann es passieren, dass bei Simulation mit mehreren SampleTimes ein Wert mit Zeitstempel zwischen vorhandenen Werten von Simulink® erwartet wird. Dementsprechend kann man, wenn genügend Daten im Buffer (siehe BufferSize) vorhanden sind, zwischen zwei Werten interpolieren oder nicht interpolieren.

Symbole Schreiben

Parameter	Default-Wert	Beschreibung
Sample Time (100ns)	TwinCAT-Task Zykluszeit	Periodendauer, mit der das ADS-Symbol geschrieben werden soll.
Timeout (ms)	1000	ADS Timeout-Zeit
BufferSize	50	Buffergröße in Samples auf Simulink®-Seite.
Interpolation Mode	Nearest	s.o.
Extrapolation Mode	None	s.o.

3.6.1.2 Async ADS Read

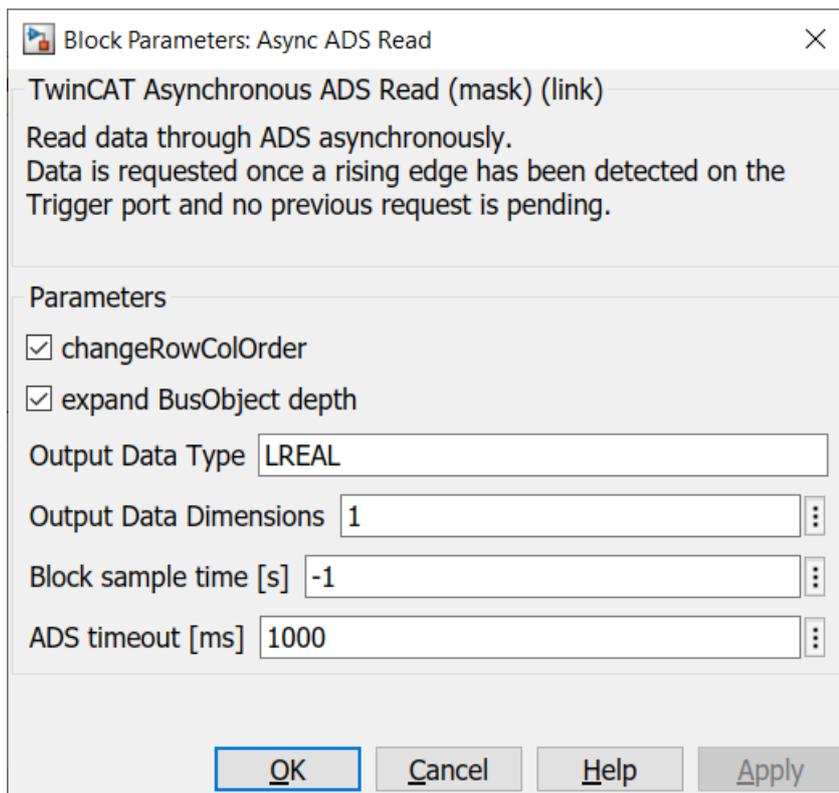
Asynchrones Lesen eines ADS-Symbols, beschrieben am Eingang über AMS Net Id, Port, Index Group und Index Offset [► 25]. Der Lesebefehl wird aus Simulink® mit positiver Flanke am Trigger-Port abgesendet.



Portbeschreibung

Port Name	Datentyp	Beschreibung
Trigger	boolean	Block sendet Lesebefehl bei positiver Flanke.
NetId	uint8	AMS Net Id des Zielsystems
Port	uint16	Port des ADS-Geräts
IndexGroup	uint32	Index Group
IndexOffset	uint32	Index Offset
Busy	boolean	TRUE von Absenden der Anfrage bis Empfang der Antwort des Servers
Error	int32	Errorcode. Wichtig: <i>Pending</i> wird als Errorcode angezeigt.
Data	<i>konfigurierbar</i>	Siehe Output Datatype und Dimension in Block Parameter

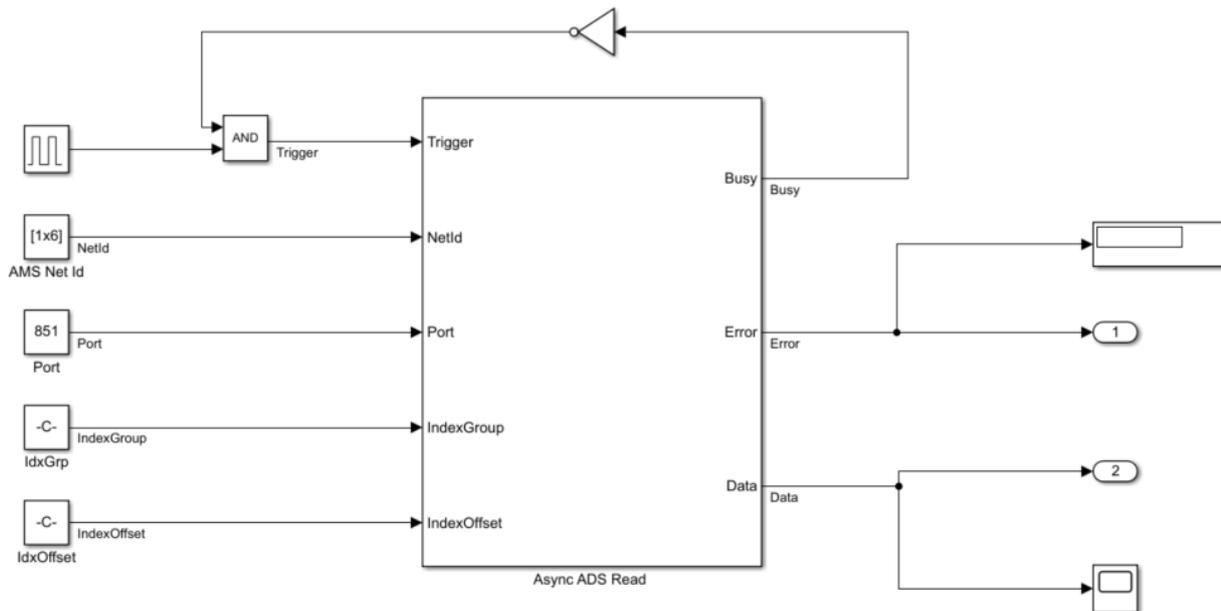
Block Parameter



Parameter	Beschreibung
“Change Row Col Order” oder “Change row major to column major matrix element order”	Transponiert die Daten in die gewöhnliche Darstellung in Simulink®.
Output Data Type	TwinCAT-Primärdatentyp, z. B. UINT, LINT, REAL, ...
Output Data Dimensions	Dimension, z. B. 5, [2,3], ...
Block sample time [s]	Sample Time des Blocks (Default -1 = Simulink Sample Time)
ADS timeout [ms]	Timeout für ADS-Aufruf

Beispiel

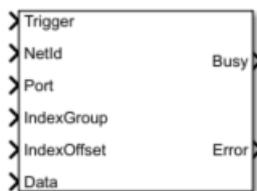
Ein Beispielmodell kann wie folgt aufgebaut werden:



- Die Const-Blöcke tragen hier die ADS-Adressinformationen.
- Die AMS Net Id wird als Array übergeben, z. B. [192,168,226,1,1,1].
- Bei Index Group und Index Offset ist zu beachten, dass i.d.R. in TwinCAT (zum Beispiel im Target Browser), die Informationen Hexadezimal angegeben werden. Es bietet sich an, den hex2dec Befehl zu nutzen, z. B. hex2dec('5DFF0').
- Über die Rückkopplung des Busy Ausgangs wird verhindert, dass während einer Anfrage eine nächste Anfrage gestellt wird.

3.6.1.3 Async ADS Write

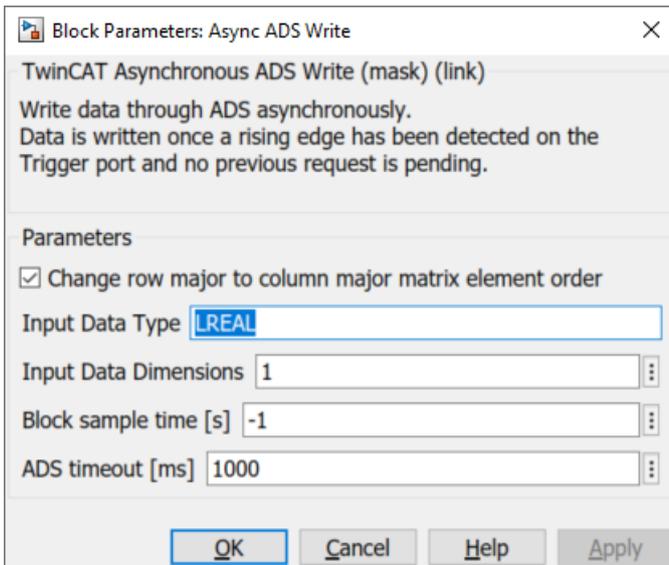
Asynchrones Schreiben eines ADS-Symbols, beschrieben am Eingang über AMS Net Id, Port, Index Group und Index Offset [► 25]. Der Lesebefehl wird aus Simulink® mit positiver Flanke am Trigger-Port abgesendet.



Portbeschreibung

Port Name	Datentyp	Beschreibung
Trigger	boolean	Block sendet Lesebefehl bei positiver Flanke.
NetId	uint8	AMS Net Id des Zielsystems
Port	uint16	Port des ADS-Geräts
IndexGroup	uint32	Index Group
IndexOffset	uint32	Index Offset
Data	<i>konfigurierbar</i>	Siehe Input Datatype und Dimension in Block Parameter
Busy	boolean	TRUE von Absenden der Anfrage bis Empfang der Antwort des Servers
Error	int32	Errorcode

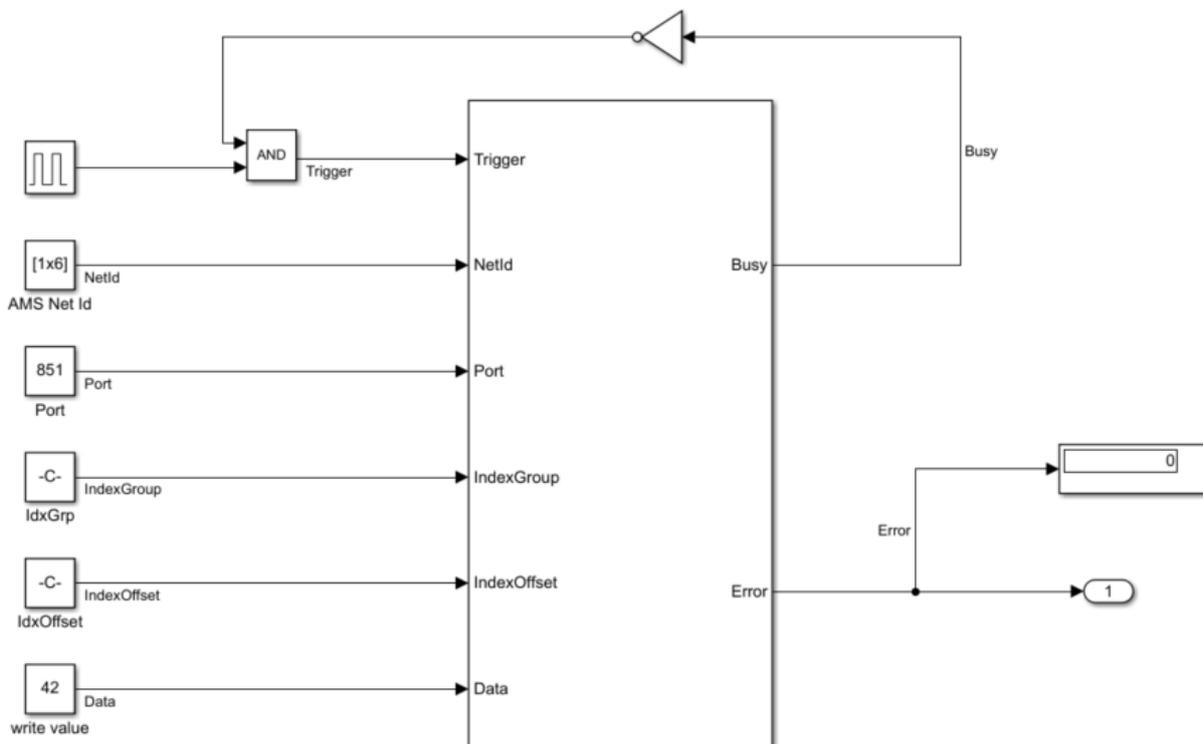
Block Parameter



Parameter	Beschreibung
Input Data Type	TwinCAT-Primärdatentyp, z. B. UINT, LINT, REAL, ...
Input Data Dimensions	Dimension, z. B. 5, [2,3], ...
Block sample time [s]	Sample Time des Blocks (Default -1 = Simulink Sample Time)
ADS timeout [ms]	Timeout für ADS-Aufruf

Beispiel

Ein Beispielmmodell kann wie folgt aufgebaut werden:

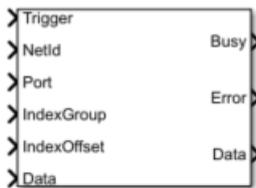


- Die Const-Blöcke tragen hier die ADS-Adressinformationen und den Wert 42, um ihn an TwinCAT zu übertragen.
- Die AMS Net Id wird als Array übergeben, z. B. [192,168,226,1,1,1].

- Bei Index Group und Index Offset ist zu beachten, dass i.d.R. in TwinCAT (zum Beispiel im Target Browser), die Informationen Hexadezimal angegeben werden. Es bietet sich an, den hex2dec Befehl zu nutzen, z. B. hex2dec('5DFF0').
- Über die Rückkopplung des Busy Ausgangs wird verhindert, dass während einer Anfrage eine nächste Anfrage gestellt wird.

3.6.1.4 Async ADS ReadWrite

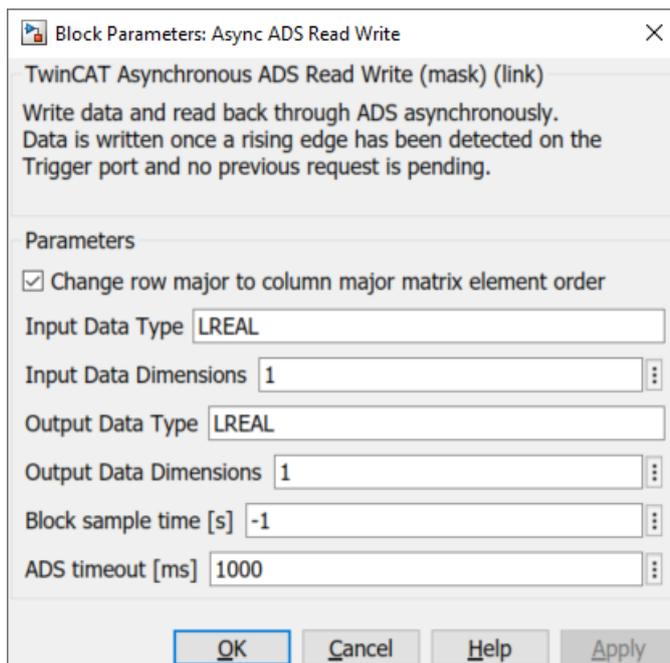
Asynchrones ReadWrite eines ADS-Symbols, beschrieben am Eingang über AMS Net Id, Port, Index Group und Index Offset [▶ 25]. Der Lesebefehl wird aus Simulink® mit positiver Flanke am Trigger-Port abgesendet.



Portbeschreibung

Port Name	Datentyp	Beschreibung
Trigger	boolean	Block sendet Lesebefehl bei positiver Flanke.
NetId	uint8	AMS Net Id des Zielsystems
Port	uint16	Port des ADS-Geräts
IndexGroup	uint32	Index Group
IndexOffset	uint32	Index Offset
Data	<i>konfigurierbar</i>	Siehe Input Datatype und Dimension in Block Parameter
Busy	boolean	TRUE von Absenden der Anfrage bis Empfang der Antwort des Servers
Error	int32	Errorcode
Data	<i>konfigurierbar</i>	Siehe Output Datatype und Dimension in Block Parameter

Block Parameter

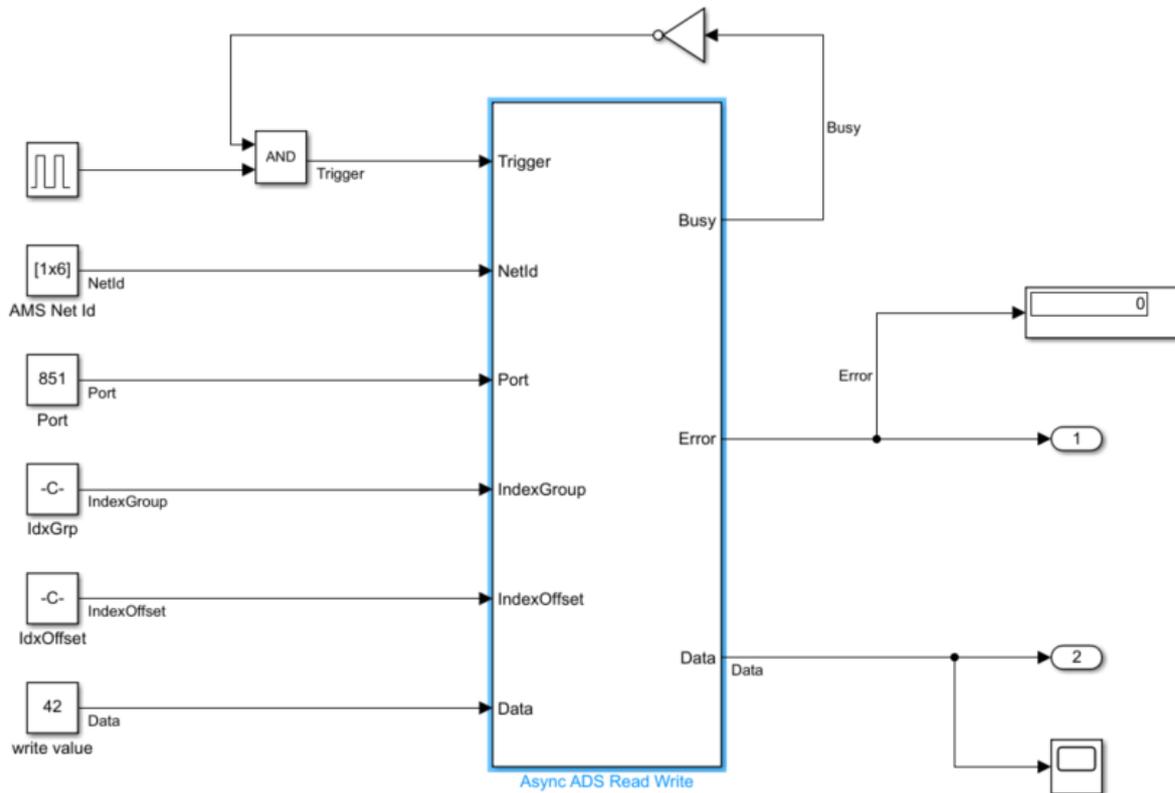


Parameter	Beschreibung
Input Data Type	TwinCAT-Primärdatentyp, z. B. UINT, LINT, REAL, ...

Parameter	Beschreibung
Input Data Dimensions	Dimension, z. B. 5, [2,3], ...
Output Data Type	TwinCAT-Primärdatentyp, z. B. UINT, LINT, REAL, ...
Output Data Dimensions	Dimension, z. B. 5, [2,3], ...
Block sample time [s]	Sample Time des Blocks (Default -1 = Simulink Sample Time)
ADS timeout [ms]	Timeout für ADS-Aufruf

Beispiel

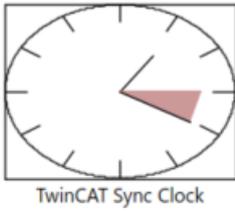
Ein Beispielmmodell kann wie folgt aufgebaut werden:



- Die Const-Blöcke tragen hier die ADS-Adressinformationen und den Wert 42, um ihn an TwinCAT als Eingang der Funktion zu übertragen.
- Die AMS Net Id wird als Array übergeben, z. B. [192,168,226,1,1,1].
- Bei Index Group und Index Offset ist zu beachten, dass i.d.R. in TwinCAT (zum Beispiel im Target Browser), die Informationen Hexadezimal angegeben werden. Es bietet sich an, den hex2dec Befehl zu nutzen, z. B. hex2dec('5DFF0').
- Über die Rückkopplung des Busy Ausgangs wird verhindert, dass während einer Anfrage eine nächste Anfrage gestellt wird.
- Data am Ausgang trägt dann den Rückgabewert der aufgerufenen Funktion.

3.6.1.5 TwinCAT Sync Clock

Synchronisation der Simulink®-Simulation mit der lokalen Systemzeit.



Block Parameter

Block Parameters: TwinCAT Sync Clock [X]

TwinCAT Sync Clock (mask) (link)

This block attempts to synchronize the simulation time the the system clock If the model can get calculated in real time or faster. Please see the documentation for further information.

Modes:
 None: no synchronization
 Clock: attempt synchronization
 ClockWithTime: attempt synchronization, create time output port
 ClockWithDeviation: attempt synchronization, create deviation output port
 ClockWithTimeAndDeviation: attempt synchronization, create deviation and time output ports

Parameters

Sync Clock Mode ▾

Sample Time (-1 for inherited) ▾

[OK] [Cancel] [Help] [Apply]

Sync Clock Mode	
None	Simulation wird weder beschleunigt noch gebremst.
Clock	Simulationszeit und Systemzeit werden angeglichen.
ClockWithTime	Simulationszeit und Systemzeit werden angeglichen. Zusätzlich wird ein Outport erzeugt, der die vergangene Systemzeit anzeigt.
ClockWithTimeAndDeviation	Simulationszeit und Systemzeit werden angeglichen. Zusätzlich wird ein Outport erzeugt, der die vergangene Systemzeit und die Abweichung zwischen Simulationszeit und Systemzeit anzeigt.

Bemerkung

Der Block synchronisiert nicht die Simulink®-Simulation mit der TwinCAT-Laufzeit, sondern mit der lokalen Systemzeit. Eine Synchronisation der lokalen Systemzeit erfolgt durch „Bremsen“ der Simulink®-Simulation; eine Beschleunigung ist nicht möglich.

Für eine vollständig synchronisierte Software-in-the-Loop-Simulation zwischen TwinCAT und Simulink® benötigen Sie eine TwinCAT 3 Usermode Runtime: External Control und den SiL Connector Block in Simulink®.

3.7 Beispiele

Über das MATLAB® Command Window erhalten Sie eine Liste von bereitgestellten Beispielen. Nutzen Sie dazu den Befehl:

```
TwinCAT.ADS.Samples.List
```

```

Command Window
>> TwinCAT.ADS.Samples.List

TE1410 Samples:

ADS Client Basics:
  Products: TE1410 - MATLAB ADS classes
  Topics: Expore ADS routes, Read PLC variables into MATLAB workspace, Plot a series of values read from a PLC variable
  Level: 1
  Description: Shows basic ADS client functionality by connecting to a TwinCAT PLC.
  Start

ADS Sum Requests:
  Products: TE1410 - MATLAB ADS classes
  Topics: Read multiple values via ADS sum request
  Level: 2
  Description: Shows how to access multiple TwinCAT PLC variables with a single ADS sum request.
  Start

ADS Request Objects:
  Products: TE1410 - MATLAB ADS classes
  Topics: Asynchronous ADS requests using request objects
  Level: 2
  Description: Shows how to use ADS request objects for parallelization of request execution.
  Start

ADS Server Basics:
  Products: TE1410 - MATLAB ADS classes
  Topics: MATLAB ADS Server
  Level: 3
  Description: Shows a MATLAB based implementation of a simple ADS server.
  Start

BigData with ADS Notifications through MATLAB ADS client:
  Products: TE1410 - MATLAB ADS classes
  Topics: Read PLC variables through an ADS notification
  Level: 5
  Description: Shows how to transfer large volumes of data from a TwinCAT PLC to MATLAB using ADS notifications
  Start

BigData through MATLAB ADS server:
  Products: TE1410 - MATLAB ADS classes
  Topics: Transfer data from the PLC to MATLAB
  Level: 5
  Description: Shows how to transfer large volumes of data from a TwinCAT PLC to MATLAB using the ADS server functionality
  Start

ADS Client Simulink Simulation:
  Products: TE1410 - MATLAB ADS classes
  Topics: Read through ADS, Write through ADS, Simulation, Simulink
  Level: 5
  Description: Create synchronous Simulink ADS blocks through MATLAB ADS
  Start

```

Sie können die Samples durch Klick auf den blauen Start-Link aufrufen. Dazu wird der Sample-Code in Ihr User-Verzeichnis kopiert, sodass Sie das Original-Sample nicht verändern. Sie können entsprechend mit der Kopie des Samples arbeiten und ausprobieren.

Ebenso verfügbar sind zum Anzeigen und Starten einzelner Samples:

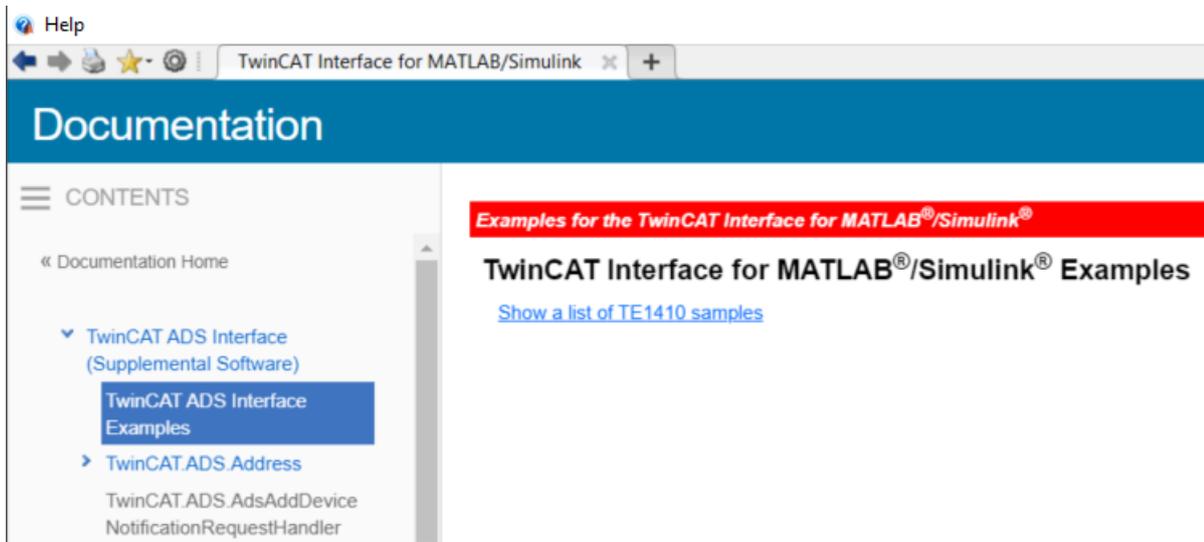
```
TwinCAT.ADS.Samples.Show(SampleName)
```

```
TwinCAT.ADS.Samples.Start(SampleName)
```

Das Argument *SampleName* ist als String zu übergeben, z. B.:

```
TwinCAT.ADS.Samples.Start('ADS Server Basics')
```

Die Liste der Samples erhalten Sie ebenso über die in MATLAB® eingebaute Dokumentation durch Klick auf den Link **Show a list of TE1410 samples**.



Mehr Informationen:
www.beckhoff.de/te1410

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

