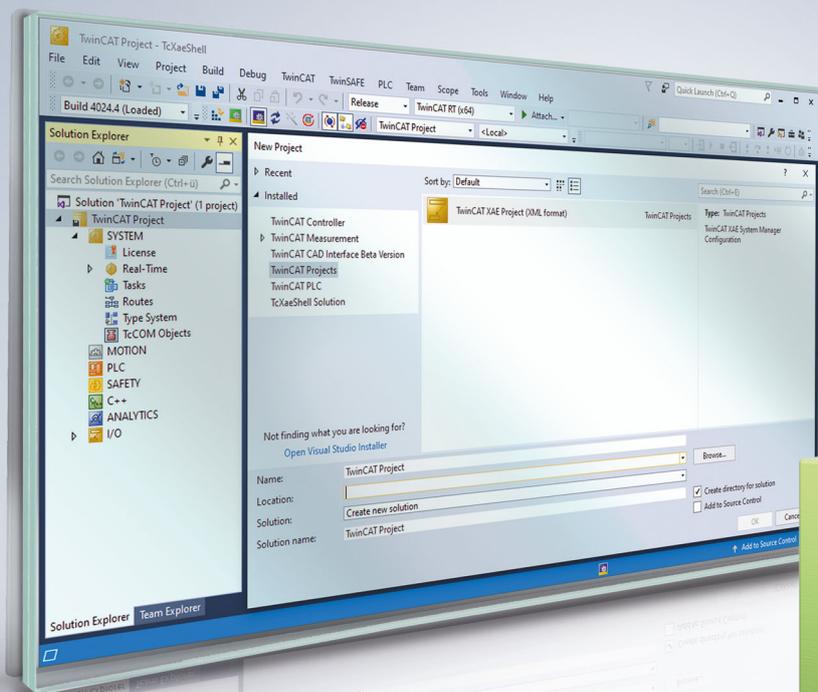


BECKHOFF New Automation Technology

Handbuch | DE

TE1111

TwinCAT 3 | EtherCAT Simulation



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Installation/ Lizenzierung	9
4	Grundlagen	10
4.1	Verkabelung	10
4.2	Mailbox Auto Response	13
4.3	CoE / SoE / AoE.....	15
5	Quickstart	19
6	Verwendung von CU2508-Port-Multipliern	21
7	Fehlersimulation	22
8	Mischbetrieb von realen und virtuellen Slaves	24
9	Reference, Benutzeroberfläche	25
9.1	Reiter Settings.....	25
9.2	Reiter Feature Simulation	26
9.3	Reiter Fault Simulation	26
10	SPS API	28
10.1	ADS-Kommandos	28

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

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

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

Disclaimer

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

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

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

Marken

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

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

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Erklärung der Symbole

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

GEFAHR

Akute Verletzungsgefahr!

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

WARNUNG

Verletzungsgefahr!

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

VORSICHT

Schädigung von Personen!

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

HINWEIS

Schädigung von Umwelt oder Geräten

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



Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

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

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

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

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

2 Übersicht

Neue Maschinen und Anlagen werden immer komplexer und müssen unter einem erheblichen Termindruck kostengünstig realisiert werden. Darüber hinaus steigt, u.a. aufgrund der Forderung nach Flexibilität der auf einer Anlage produzierbaren Produkte, der Anteil der Steuerungssoftware immer weiter an. Eine Reduzierung des Engineering-Aufwandes für die Erstellung des Steuerungscode sowie der Inbetriebnahme dieses Codes an der realen Anlage verspricht somit einen klaren wirtschaftlichen Nutzen. Eine Methode, die darauf abzielt dies zu erreichen, bezeichnet man als „Virtuelle Inbetriebnahme“. Ziel ist es, bereits in einem frühen Stadium des Engineerings auch ohne real existierende Hardware den erzeugten Steuerungscode zu testen und zu optimieren, sodass die reale Inbetriebnahme deutlich schneller verlaufen kann.

Die Function **TE1111 TwinCAT EtherCAT Simulation** wurde erstellt, um diese Anforderungen zu erfüllen. Sind darüber hinaus bereits Modelle der Maschine/ Anlage in Matlab-Simulink oder einem Simulationstool vorhanden, welches einen FMU-(Code-)Export anbietet, ist es möglich, gemeinsam mit den Functions TE1400 Target für Matlab®/ Simulink® bzw. **TE1420 Target für FMI** mit überschaubarem Aufwand HIL (Hardware-In-the Loop)-Simulationen durchzuführen.

Die Function **TE1111 TwinCAT EtherCAT Simulation** dient zur Simulation eines EtherCAT-Stranges. Dabei wird eine erstellte E/A-Konfiguration der realen Anlage exportiert und kann auf einem zweiten System als „EtherCAT-Simulation“-Device wieder importiert werden. Auf diesem System steht damit ein gespiegeltes Prozessabbild zur Verfügung, das mit entsprechenden TwinCAT-Modulen (z.B. erstellt in den Sprachen der IEC 61131-3 oder aus Matlab®/Simulink® heraus) verknüpft werden kann. In diesen Modulen muss das gewünschte Verhalten der Maschine hinreichend genau implementiert sein. Startet man nun auf beiden Systemen die TwinCAT-Echtzeit, hat man eine HIL-Simulation, ohne dass das Original-Projekt dafür angepasst werden muss.



Da das Projekt des zu testenden Steuerungssystems nicht verändert werden soll, arbeitet das EtherCAT-Simulation Device nicht synchronisiert. Dies bedeutet, dass die Task, welche das Simulation-Device treibt, aufgrund des Shannonschen Theorems (Abtast-Theorem) doppelt so schnell laufen muss. Auf Standard-IPC-Hardware ist die kleinste Zykluszeit des Simulation Devices derzeit 50µs. Somit sind HIL-Simulationen des zu testenden Steuerungssystems mit 100µs Zykluszeit möglich.

Unterstützte Features:

Basisfunktionalität

- Digitale Slaves sowie Slaves mit CoE-Parametern werden unterstützt
- Ein gespiegeltes Prozessabbild wird erstellt und kann verlinkt werden
- Einfache Konfiguration des EtherCAT-Simulation Devices direkt in der TwinCAT XAE
- Unterstützungen von Distributed Clocks
- Unterstützung von AoE und SoE Mailboxkommandos (werden an die SPS weitergeleitet)
- Mischsimulation von realen und simulierten Slaves
- HotConnect Simulation (ab TwinCAT 3.1.4024)
- Fehlersimulation (Lost Frames, Link Lost, Working Counter) (ab TwinCAT 3.1.4024)
- CU2508 Port-Multiplier-Support (ab TwinCAT 3.1.4024)

3 Installation/ Lizenzierung

Die Function TE1111 | TwinCAT EtherCAT Simulation wird bereits mit TwinCAT 3 installiert und ist seit der TwinCAT-Version 3.1 Build 4018.0 als Release-Version enthalten. Sie muss daher lediglich lizenziert werden. Beachten Sie dazu das Kapitel [Bestellung und Aktivierung von TwinCAT-3-Standardlizenzen](#). Die Function TE1111 hat eine instanzbasierte Lizenz, sie wird also pro instanzierter EtherCAT Simulation Device lizenziert.

Erforderliche Lizenzen:

TE1111 EtherCAT Simulation

Obwohl diese Lizenz auf dem Zielsystem aktiviert werden muss, handelt es sich um eine Engineering-Lizenz. Zu Testzwecken kann die EtherCAT-Simulation im Demomodus auch ohne Lizenz genutzt werden.



Für dieses Produkt ist keine 7-Tage-Testlizenz mit allen Funktionen verfügbar.

Einschränkungen in der Demoversion

Wird die EtherCAT-Simulation im Demo-Mode verwendet, fallen alle simulierten E/As nach 30 min ab.

4 Grundlagen

Die Funktion **TE1111 TwinCAT EtherCAT Simulation** dient zur Simulation eines EtherCAT-Stranges. Dabei wird eine erstellte E/A-Konfiguration der realen Anlage exportiert und kann auf einem zweiten System als „EtherCAT-Simulation“-Device wieder importiert werden. Auf diesem System steht damit ein gespiegeltes Prozessabbild zur Verfügung, das mit entsprechenden TwinCAT-Modulen (z.B. erstellt in den Sprachen der IEC 61131-3 oder aus Matlab®/Simulink® heraus) verknüpft werden kann. In diesen Modulen muss das gewünschte Verhalten der Maschine hinreichend genau implementiert sein. Startet man nun auf beiden Systemen die TwinCAT-Echtzeit, hat man eine HIL-Simulation, ohne dass das Original-Projekt dafür angepasst werden muss.



Da das Projekt des zu testenden Steuerungssystems nicht verändert werden soll, arbeitet das EtherCAT-Simulation Device nicht synchronisiert. Dies bedeutet, dass die Task, welche das Simulation-Device treibt, aufgrund des Shannonschen Theorems (Abtast-Theorem) doppelt so schnell laufen muss. Auf Standard-IPC-Hardware ist die kleinste Zykluszeit des Simulation Devices derzeit 50µs. Somit sind HIL-Simulationen des zu testenden Steuerungssystems mit 100µs Zykluszeit möglich.

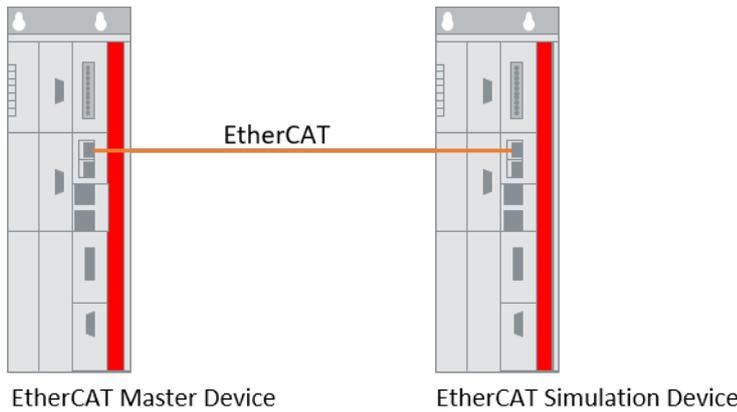
Neben der HIL-Simulation kann die EtherCAT-Simulation aber auch zur Simulation eines oder mehrere EtherCAT-Stränge auf demselben Rechner verwendet werden, solange genügend freie Netzwerk-Schnittstellen zur Verfügung stehen. Je nach Anwendungsfall sind verschiedene Verkabelungsmöglichkeiten vorhanden, welche im Kapitel [Verkabelung \[▶ 10\]](#) aufgezeigt werden.

Die EtherCAT-Simulation beschränkt sich im Wesentlichen auf die EtherCAT-relevanten Aspekte der Bus-Teilnehmer. Das interne Verhalten der Busteilnehmer selber wird dabei vernachlässigt. Für einige Busteilnehmer ist es aber erforderlich zumindest ein Teil des Verhaltens zu simulieren, da andernfalls der EtherCAT-Master nicht korrekt auf gestartet werden kann (z. B. EL6224). Um dies zu simulieren, bietet die EtherCAT-Simulation die Möglichkeit, die Option [Mailbox Auto Response \[▶ 13\]](#) zu aktivieren, welche solche Anfragen automatisch beantwortet ohne das etwas implementiert werden muss. Sollten diese Parameter dennoch benötigt werden, kann diese Option auch deaktiviert werden. In diesem Fall ist es möglich die CoE, SoE und AoE-Kommunikation in ein Anwender-SPS-Module umzuleiten und dort auf diese Telegramme zu reagieren (siehe hierzu [CoE / SoE / AoE \[▶ 15\]](#)).

Ein weiteres Beispiel, bei dem die reine EtherCAT-Betrachtung der Teilnehmer nicht ausreicht sind Antriebe. Hier ist es erforderlich zumindest einen Teil der Antriebs-State-Machine zu implementieren, damit die TwinCAT NC korrekt auf starten kann. Für die Antriebsprofile CoE und SoE sind diese beispielhaft als Funktionsbaustein implementiert (siehe hierzu [Antriebssimulation](#)). Diese Funktionsbausteine können für jeden in der Konfiguration enthaltenen Antrieb, entsprechend in einem SPS-Module instanziiert und mit den Achsen verknüpft werden. Sie werden somit zwischen der EtherCAT- und der zu implementierenden Prozess-Simulation geschaltet.

4.1 Verkabelung

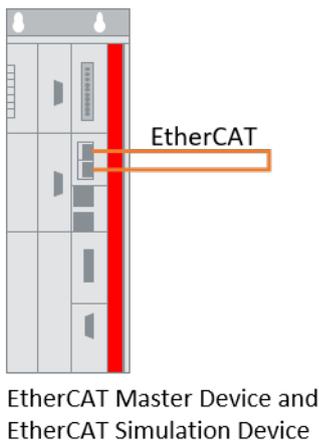
Wie bereits im Kapitel [Übersicht \[▶ 8\]](#) beschrieben, wurde die EtherCAT-Simulation entworfen um, eine virtuelle Inbetriebnahme einer Anlage zu unterstützen. Die für diesen Zweck normale Verkabelung wird wie folgt aussehen:



Auf der linken Seite ist der Steuerungsrechner zu sehen, welcher das Original-Steuerungsprojekt enthält, welches später verwendet werden soll um die Maschine/ Anlage zu steuern. Auf der rechten Seite zu sehen ist ein Simulations-IPC, welcher ein EtherCAT-Simulation Device enthält. Beide Systeme sind durch ein Standard-EtherCAT-Kabel verbunden.

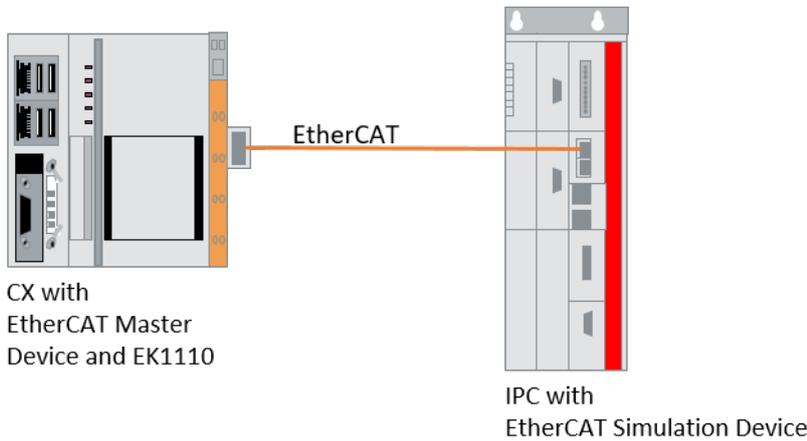
Verwendung von EtherCAT und EtherCAT-Simulation auf demselben IPC:

Zum Testen von Teilfunktionen des Steuerungsprogramms (z.B. EtherCAT-Diagnose) zu einem früheren Zeitpunkt (vor der virtuellen Inbetriebnahme), kann auch der Steuerungsrechner zur Simulation des EtherCATs mit verwendet werden, solange der Steuerungsrechner über genügend freie Netzwerkschnittstellen verfügt. Genau wie jeder EtherCAT-Master benötigt auch jedes EtherCAT-Simulation-Device eine eigene Netzwerkschnittstelle.



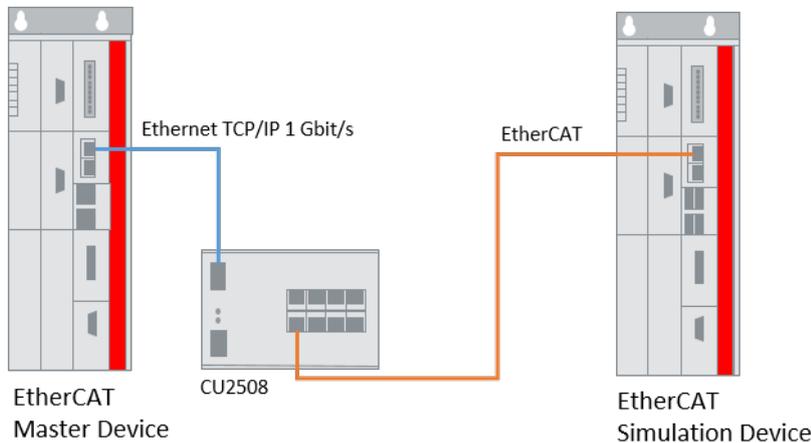
Verwendung eines CX als Steuerung

Wird als Steuerung ein CX verwendet, der mit einem Simulations-IPC gekoppelt werden soll, sind je nach Verkabelung u.U. noch EtherCAT-Teilnehmer vorhanden, die nicht berücksichtigt werden dürfen. Im folgenden Bild sind dies z.B. das intelligente Netzteil des CXes und eine EK1110 EtherCAT Extension. Hier kann im [Reiter Settings](#) [▶ 25] eine manuelle Geräte Korrektur eingestellt werden, also Geräte die nicht berücksichtigt werden sollen. Wird diese Einstellung nicht gesetzt, könnte dies zur Folge haben, dass der EtherCAT-Master nicht auf startet oder Working-Counter falsch berechnet werden.



Verwendung des Port-Multipliers auf der Original-Projekt-Seite:

Wird in der Echtzeitkonfiguration des Steuerungsrechners ein CU2508 Port-Multiplier verwendet, können die Ausgänge des Port-Multipliers mit normalen Netzwerkports auf einem Simulation-Rechner verbunden werden. Für diese Netzwerkports müssen dann auf dem Simulation-Rechner entsprechend viele EtherCAT-Simulation Devices erstellt werden.



HINWEIS

Verwendung des Port-Multipliers auf der EtherCAT-Simulation-Seite nicht unterstützt

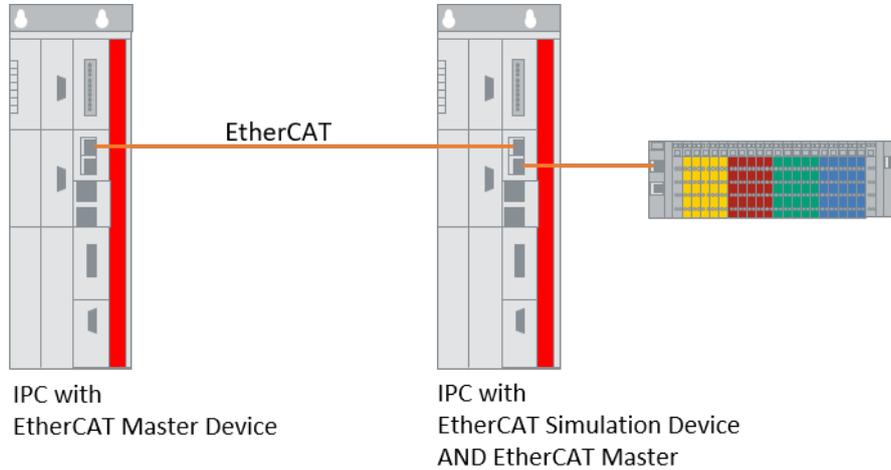
Die Verwendung eines CU2508 Port-Multipliers auf der EtherCAT-Simulation-Seite wird nicht unterstützt. Wird ein CU2508 auf der Slave-Seite eingesetzt, so ist dieser im TwinCAT-Projekt, welches die Konfigurationsdatei exportiert, nicht enthalten. Auch den EtherCAT-Frame erhält dieser CU2508 erst nach dem EtherCAT-Simulation-Device. Somit ist dieser für das EtherCAT-Simulation-Device „unsichtbar“. Daraus folgt, dass der Device-State und auch der Slave-Count auf der EtherCAT-Master-Seite keine korrekten Werte anzeigen. Eine im Original-Projekt ausprogrammierte EtherCAT-Diagnose würde somit falsche Ergebnisse anzeigen.

Verwendung des Port-Multipliers auf der EtherCAT-Simulation-Seite:

Die Verwendung eines CU2508 Port-Multipliers auf der EtherCAT-Simulation-Seite wird nicht unterstützt. Wird ein CU2508 auf der Slave-Seite eingesetzt, so ist dieser im TwinCAT-Projekt, welches die Konfigurationsdatei exportiert, nicht enthalten. Auch den EtherCAT-Frame erhält dieser CU2508 erst nach dem EtherCAT-Simulation-Device. Somit ist dieser für das EtherCAT-Simulation-Device „unsichtbar“. Daraus folgt, dass der Device-State und auch der Slave-Count auf der EtherCAT-Master-Seite keine korrekten Werte anzeigen. Eine im Original-Projekt ausprogrammierte EtherCAT-Diagnose würde somit falsche Ergebnisse anzeigen.

Betreiben der EtherCAT Simulation im Mixed-Mode

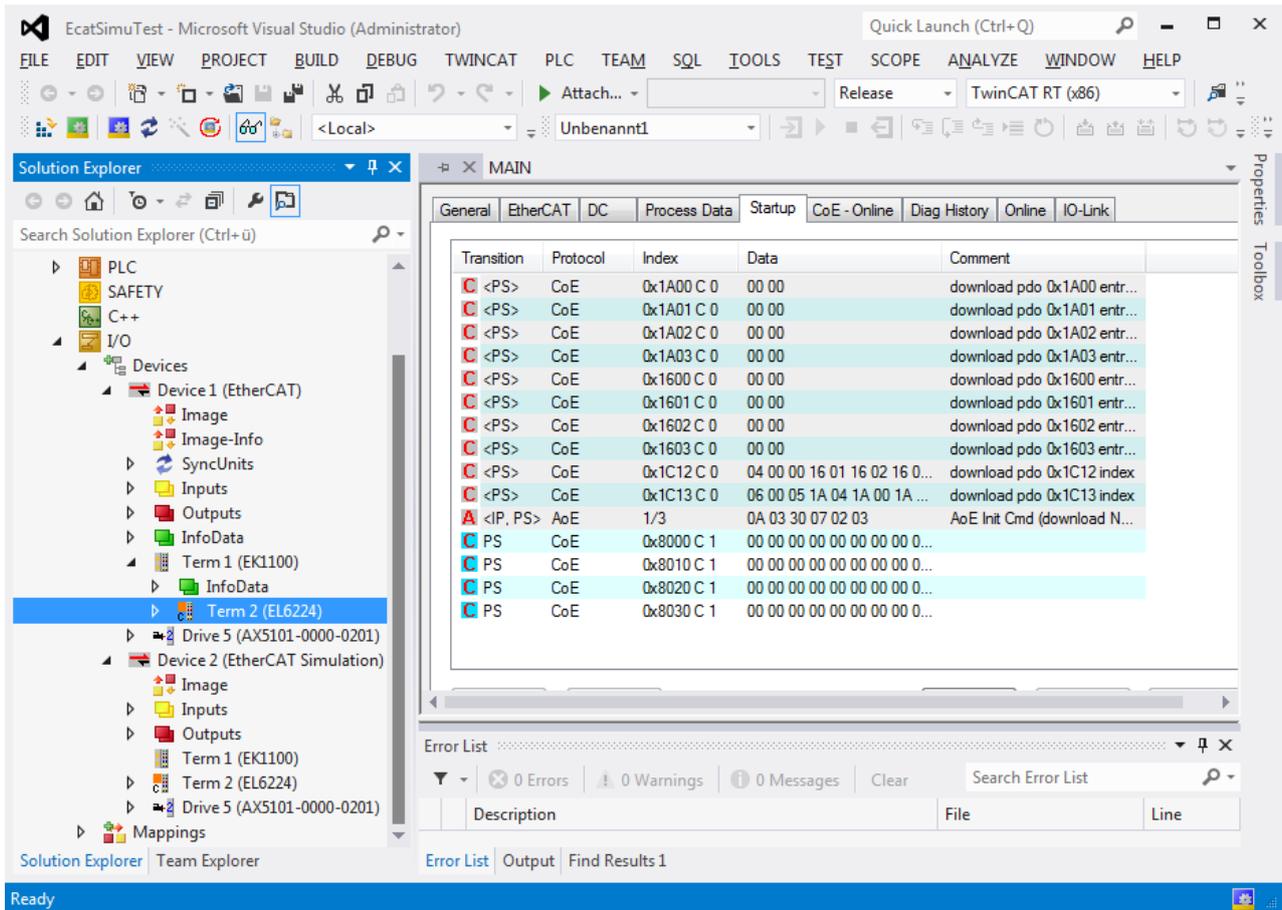
Soll eine Maschine/ Anlage schrittweise in Betrieb genommen werden bzw. nur Teile der Maschine/ Anlage simuliert werden, kann eine Verkabelung wie in der folgenden Abbildung dargestellt erfolgen. Hierzu ist es erforderlich, dass der Simulations-IPC 2 freie Netzwerk-Schnittstellen besitzt. Die erste Netzwerkschnittstelle wird als EtherCAT-Simulation-Device verwendet und wird mit dem EtherCAT-Kabel zum Steuerungs-IPC verbunden. Die zweite Netzwerkschnittstelle wird als EtherCAT-Master verwendet und mit den real existierenden Busteilnehmern verbunden (siehe hierzu auch Mischbetrieb von realen und virtuellen Slaves [► 24]).



4.2 Mailbox Auto Response

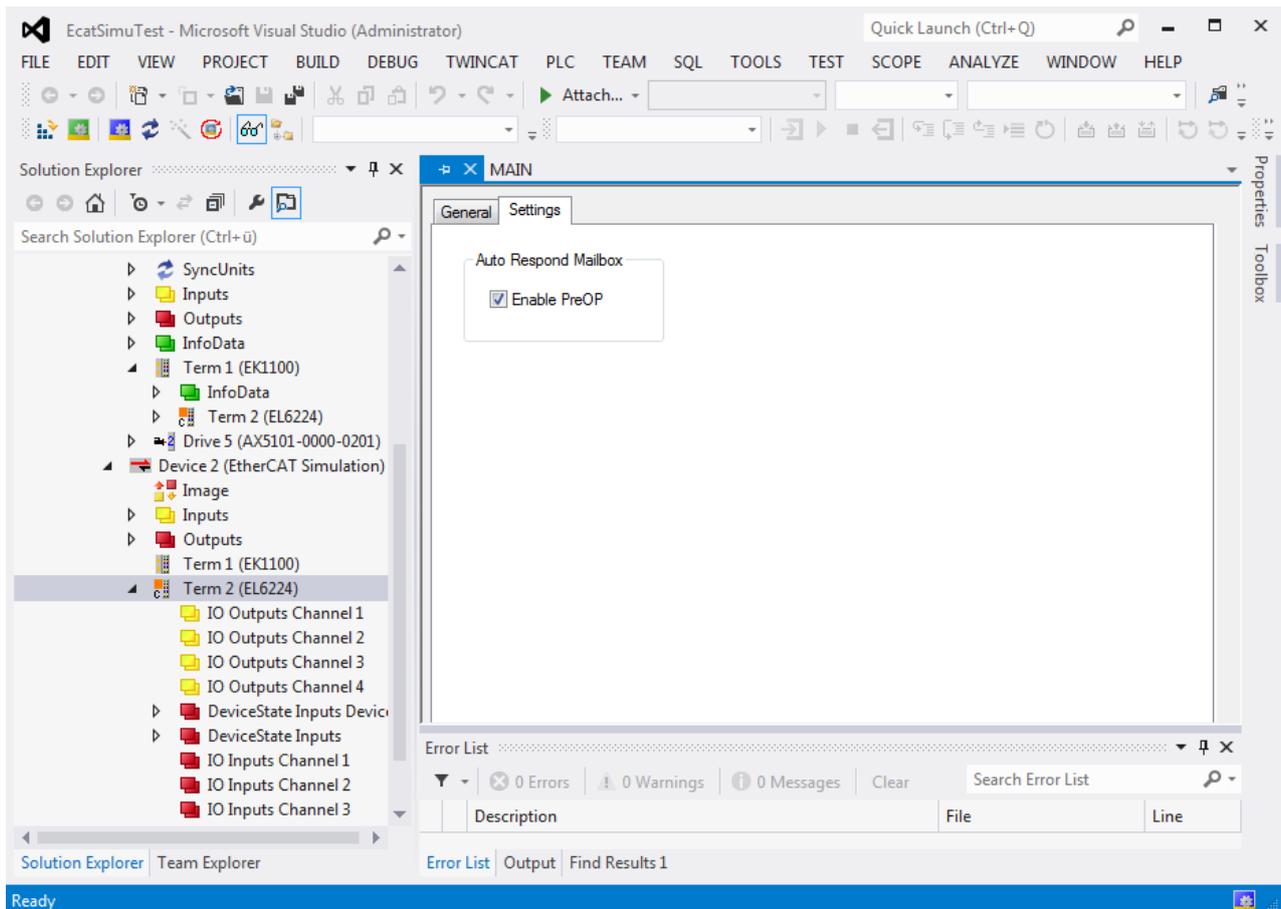
Bei einigen Klemmen gibt es auf der Masterseite Startup Commands, die bei jedem EtherCAT Hochlauf (typischerweise von PreOP nach SaveOP) an die Klemme geschickt werden müssen.

Beispiel EL6224 (IO-Link):



Diese Startup Commands müssen vom EtherCAT Simulator beantwortet werden, damit der EtherCAT-Master korrekt hochläuft. Entsprechend ist das Default Verhalten des EtherCAT-Simulation Devices, dass Startup-Parameter von Zustand PreOP nach SafeOP automatisch beantwortet werden, ohne dass in der Applikation etwas dafür getan werden muss.

Sollten diese Parameter dennoch benötigt werden, kann die Option „Auto Respond Mailbox“ deaktiviert werden (siehe Abbildung unten). Ist diese Option deaktiviert, müssen die Anfragen im Applikations-Programm entsprechend bearbeitet werden, damit der Master startet.



4.3 CoE / SoE / AoE

Mit Hilfe der EtherCAT Simulation ist es möglich, die Protokolle CoE (Can over EtherCAT), SoE (Sercos over EtherCAT) und AoE (ADS over EtherCAT) weiter zu routen (z.B. an die SPS). Es ist somit möglich, aus der SPS heraus über diese Protokolle Daten zu lesen und zu schreiben.

Routen von CoE und SoE an eine PLC

Folgende zwei Schritte sind erforderlich:

1. Die NetId und der Port an die die Daten weitergeleitet werden sollen, müssen dem EtherCAT-Simulation Device bekannt gemacht werden. Dies geschieht über ein ADS-Write an die NetID des Simulation Devices, PortNr.: 0xFFFF, IndexGroup: 0xFF01, IndexOffset:1

Beispiel: Anmelden des Mailbox-Empfängers

```
//AMSNetID of Simulation Device
netIdSimu := '5.35.2.224.2.1';

//AMSNetID of PLC Project on Sim. Dev.
arrNetId[0]:= 5;
arrNetId[1]:= 35;
arrNetId[2]:= 2;
arrNetId[3]:= 224;
arrNetId[4]:= 1; //system ID 1.1
arrNetId[5]:= 1;
arrNetId[6]:= 16#53; //Port of PLC runtime (1)
arrNetId[7]:= 16#03; //Port of PLC runtime (2) (Port 851 = 16#353)

// sent registration to Simulation driver
fbAdsWrite(WRITE:= FALSE);
fbAdsWrite(NETID:= netIdSimu, PORT:= 16#FFFF, IDXGRP:= 16#FF01, IDXOFFS:= 1, SRCADDR:= ADR(arrNetId)
, LEN:= 8, WRITE:= TRUE);
```

- Abfangen und beantworten der ADS Indications. Dabei ist in der Indexgroup der Mailboxdienst hinterlegt, die PortID identifiziert den sendenden oder empfangenden Teilnehmer und anhand von Index und Sub-Index werden die Daten angegeben, auf die zugegriffen werden soll.

Beispiel: Abfangen und beantworten von Indications:

```
//capture all write requests
fbAdsWriteInd();
IF fbAdsWriteInd.VALID = TRUE THEN //true if a write command has been sent
  IF fbAdsWriteInd.IDXGRP = 16#8000F302 THEN //F302 = CoE request, F420 = SoE request
    //fill in your custom CoE Object Dictionary
    // example reaction
    IF fbAdsWriteInd.IDXOFFS = 16#80000000 THEN (*error*)
      fbAdsWriteRes(
        NETID:= fbAdsWriteInd.NETID,
        PORT:= fbAdsWriteInd.PORT,
        INVOKEID:= fbAdsWriteInd.INVOKEID,
        RESULT:= 0,
        RESPOND:= TRUE);
      fbAdsWriteRes(RESPOND:= FALSE);
    ELSE
      fbAdsWriteRes(
        NETID:= fbAdsWriteInd.NETID,
        PORT:= fbAdsWriteInd.PORT,
        INVOKEID:= fbAdsWriteInd.INVOKEID,
        RESULT:= 0,
        RESPOND:= TRUE);
      fbAdsWriteRes(RESPOND:= FALSE);
    END_IF
  END_IF
  fbAdsWriteInd(CLEAR:= TRUE);
  fbAdsWriteInd(CLEAR:= FALSE);
END_IF

//capture all read requests
fbAdsReadInd();
IF fbAdsReadInd.VALID = TRUE THEN
  IF fbAdsReadInd.IDXGRP = 16#8000F302 THEN (*CoE*)
    // fill in your CoE Object Dictionary
    // example reaction
    IF fbAdsReadInd.IDXOFFS = 16#80000000 THEN (*error*)
      fbAdsReadRes(
        NETID:= fbAdsReadInd.NETID,
        PORT:= fbAdsReadInd.PORT,
        INVOKEID:= fbAdsReadInd.INVOKEID,
        LEN:= 0,
        RESULT:= 16#34567890,
        RESPOND:= TRUE);
    ELSE
      IF fbAdsReadInd.IDXOFFS = 16#60000000 THEN (*short response*)
        fbAdsReadRes(
          NETID:= fbAdsReadInd.NETID,
          PORT:= fbAdsReadInd.PORT,
          INVOKEID:= fbAdsReadInd.INVOKEID,
          DATAADDR:= ADR(arrNetId),
          LEN:= 2,
          RESULT:= 0,
          RESPOND:= TRUE);
      ELSE
        fbAdsReadRes(
          NETID:= fbAdsReadInd.NETID,
          PORT:= fbAdsReadInd.PORT,
          INVOKEID:= fbAdsReadInd.INVOKEID,
          DATAADDR:= ADR(arrNetId),
          LEN:= 6,
          RESULT:= 0,
          RESPOND:= TRUE);
      END_IF
    END_IF
    fbAdsReadRes(RESPOND:= FALSE);
  END_IF
  fbAdsReadInd(CLEAR:= TRUE);
  fbAdsReadInd(CLEAR:= FALSE);
END_IF
```

Wie im Beispiel zu sehen ist, erfolgt eine Prüfung, ob die Indication gültig ist und wenn ja, ob diese vom Simulation-Device gesendet wurde. Ist beides der Fall, handelt es sich um eine weitergeleitete Mailbox-Nachricht auf die entsprechend reagiert werden muss. Die IndexGroup zeigt das Mailboxprotokoll an (16#8000F302 für CoE und 16#8000F420 für SoE).

Das oben beschriebene Sample kann hier heruntergeladen werden: https://infosys.beckhoff.com/content/1031/TE1111_EtherCAT_Simulation/Resources/3268846987.zip

Versenden von CoE Emergency Fehlermeldungen

Das Versenden von CoE Emergency Fehlermeldungen erfolgt analog wie oben gezeigt per ADS Write an die **NetId** des EtherCAT-Simulation Devices, **PortNr.:** Portnummer der Klemme, die die Emergency versenden soll, **IndexGroup:** 0x00FF41, **IndexOffset:** 0.

Versenden von SoE Notifications

Das Versenden von SoE Notifications Fehlermeldungen erfolgt analog wie oben gezeigt per ADS Write an die **NetId** des EtherCAT-Simulation Devices, **PortNr.:** Portnummer der Klemme, die die Notification versenden soll, **IndexGroup:** 0x00FF42, **IndexOffset:** ergibt sich anhand der Typdefinition TETHERCAT_SOE_ADS_IOFFS:

```
typedef struct TETHERCAT_SOE_ADS_IOFFS
{
    USHORT IDN;
    union
    {
        struct
        {
            BYTE DataState : 1;
            BYTE Name : 1;
            BYTE Attribute : 1;
            BYTE Unit : 1;
            BYTE Min : 1;
            BYTE Max : 1;
            BYTE Value : 1;
            BYTE Default : 1;
        };
        BYTE Elements;
    };
    BYTE DriveNo : 3;
    BYTE Reserved2 : 4;
    BYTE Command : 1;
} ETHERCAT_SOE_ADS_IOFFS
```

Beispiel: 0x00401234 -> Für die SoE IDN 1234 wird hier das Value Flag adressiert.

Routen von AoE an eine PLC

Die Anmeldung des Mailbox-Empfängers erfolgt identisch wie unter CoE und SoE beschrieben. Das Handling von AoE in der PLC erfolgt ebenfalls in ähnlicher Struktur, ist jedoch aufgrund des Funktionsumfangs von AoE mit mehreren Abfragen verbunden. Ein Beispiel ist im Folgenden aufgeführt und kann hier heruntergeladen werden: https://infosys.beckhoff.com/content/1031/TE1111_EtherCAT_Simulation/Resources/3268845323.zip

```
fbAdsReadWriteInd();
IF fbAdsReadWriteInd.VALID = TRUE THEN
    MEMCPY(ADR(arrTmp), fbAdsReadWriteInd.DATAADDR, fbAdsReadWriteInd.WRTLENGTH);
    //copy of data in temp variable
    //evaluate destination of AeE command (e.g. terminal)-first 6 byte in arrTmp
    FOR Idx := 0 TO 5 DO
        arrDest[Idx] := arrTmp[Idx];
    END_FOR
    IF smyDestId = F_CreateAmsNetId(arrDest) THEN
        //evaluate port (different physical ports or different services of terminal may be possible)
        IF fbAdsReadWriteInd.PORT = 16#1000 THEN //Io-Link Port 1
            // evaluate command type - byte 16 : write = 3, read = 2
            IF arrTmp[16] = 3 THEN // write command
                //evaluate index group and index offset as shown in CoE and SoE
                //index group: evaluate service (e.g. object register of Io-Link Terminal)
                IF fbAdsReadWriteInd.IDXGRP = 16#8000F302 THEN
                    IF fbAdsReadWriteInd.IDXOFFS = 16#80000000 THEN //adressing of service
                        MEMCPY(ADR(arrTmp), fbAdsReadWriteInd.DATAADDR + 8, 8);
                        //change destination net ID and source net ID
                        MEMCPY(ADR(arrTmp) + 8, fbAdsReadWriteInd.DATAADDR, 8);
```

```

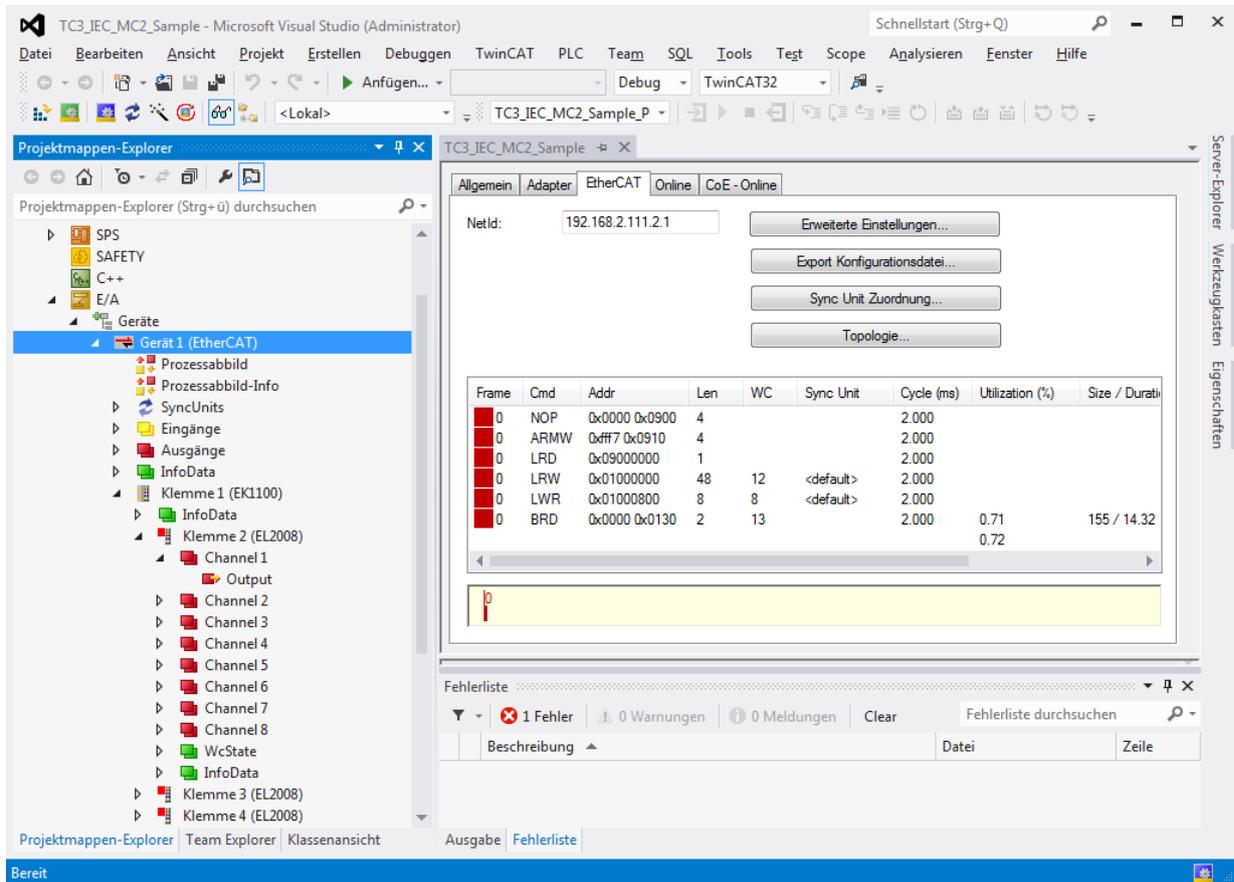
arrTmp[18].0:= 1; (*set Respond Bit*)
arrTmp[20]:= 4; (*Length*)
arrTmp[21]:= 0;
arrTmp[22]:= 0;
arrTmp[23]:= 0;
arrTmp[32]:= 0; (*Response Code*)
arrTmp[33]:= 0;
arrTmp[34]:= 0;
arrTmp[35]:= 0;
fbAdsReadWriteRes (
    NETID:= fbAdsReadWriteInd.NETID,
    PORT:= fbAdsReadWriteInd.PORT,
    INVOKEID:= fbAdsReadWriteInd.INVOKEID,
    DATAADDR:= ADR(arrTmp),
    LEN:= 36, RESPOND:= TRUE);
END_IF
END_IF
ELSIF arrTmp[16] = 2 THEN //read command
IF fbAdsReadWriteInd.IDXGRP = 16#8000F302 THEN
    IF fbAdsReadWriteInd.IDXOFFS = 16#80000000 THEN //addressing of service
        MEMCPY(ADR(arrTmp), fbAdsReadWriteInd.DATAADDR + 8, 8);
        //change destination net ID and source net ID
        MEMCPY(ADR(arrTmp) + 8, fbAdsReadWriteInd.DATAADDR, 8);
arrTmp[18].0:= 1; (* set Respond Bit*)
arrTmp[20]:= 14; (*Len*)
arrTmp[21]:= 0;
arrTmp[22]:= 0;
arrTmp[23]:= 0;
arrTmp[32]:= 0; (*Response Code*)
arrTmp[33]:= 0;
arrTmp[34]:= 0;
arrTmp[35]:= 0;
arrTmp[36]:= 6; (*Len*)
arrTmp[37]:= 0;
arrTmp[38]:= 0;
arrTmp[39]:= 0;
arrTmp[40]:= arrNetId[0]; (*Daten*)
arrTmp[41]:= arrNetId[1];
arrTmp[42]:= arrNetId[2];
arrTmp[43]:= arrNetId[3];
arrTmp[44]:= arrNetId[4];
arrTmp[45]:= arrNetId[5];
fbAdsReadWriteRes (
    NETID:= fbAdsReadWriteInd.NETID,
    PORT:= fbAdsReadWriteInd.PORT,
    INVOKEID:= fbAdsReadWriteInd.INVOKEID,
    DATAADDR:= ADR(arrTmp),
    LEN:= 46, RESPOND:= TRUE);
END_IF;
END_IF;
END_IF
END_IF;
fbAdsReadWriteRes (
    NETID:= fbAdsReadWriteInd.NETID,
    PORT:= fbAdsReadWriteInd.PORT,
    INVOKEID:= fbAdsReadWriteInd.INVOKEID,
    DATAADDR:= ADR(arrTmp),
    LEN:= fbAdsReadWriteInd.WRTLENGTH,
    RESPOND:= TRUE);
fbAdsReadWriteRes (RESPOND:= FALSE);
fbAdsReadWriteInd (CLEAR:= TRUE);
fbAdsReadWriteInd (CLEAR:= FALSE);
END_IF

```

5 Quickstart

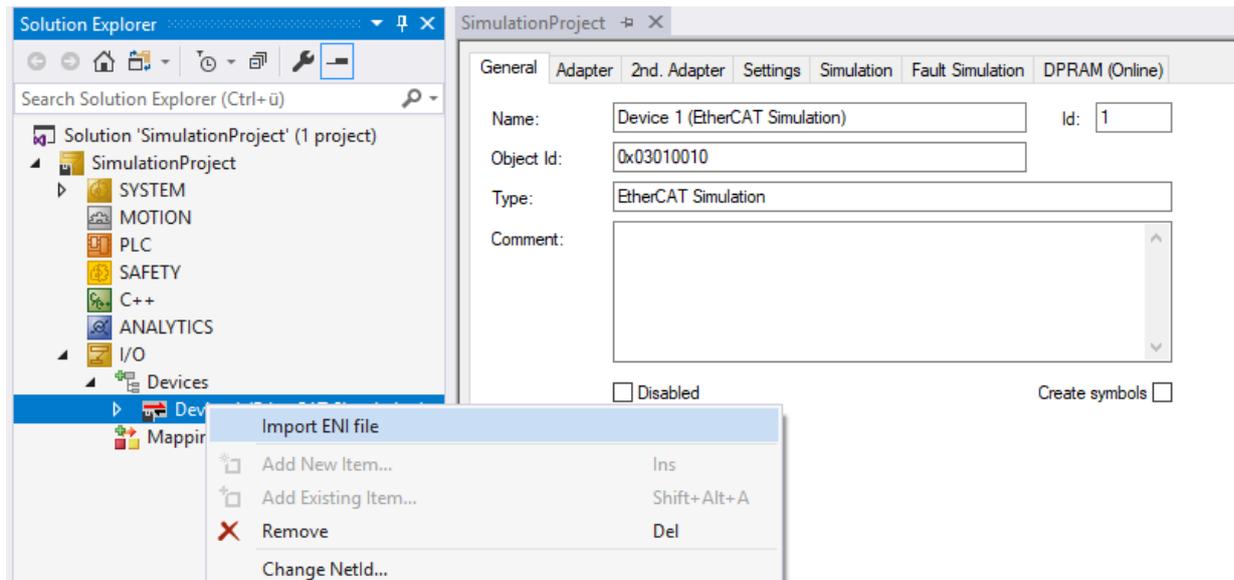
Erstellung eines Simulationsprojektes mit der Hardware, die im Steuerungsprojekt Anwendung finden soll.

1. Zum Exportieren der Hardware-Konfiguration des Steuerungsprojektes klicken Sie im Projektmappen-Explorer auf die EtherCAT-Master-Device, danach auf den Tab **EtherCAT** und schließlich auf den Button **Export Konfigurationsdatei...**



2. Erstellen Sie dann das Simulationsprojekt auf dem Simulations-IPC
3. Klicken Sie auf **Neues Element Hinzufügen**, um ein EtherCAT-Simulation-Device zu erzeugen.

4. Öffnen Sie das Kontextmenü des EtherCAT-Simulation-Devices und klicken Sie auf **Import ENI file**, um die EtherCAT-Konfiguration des EtherCAT-Masters zu importieren.



⇒ Das Simulationsprojekt ist angelegt und mit der Hardware-Konfiguration des Steuerungsprojektes befüllt.

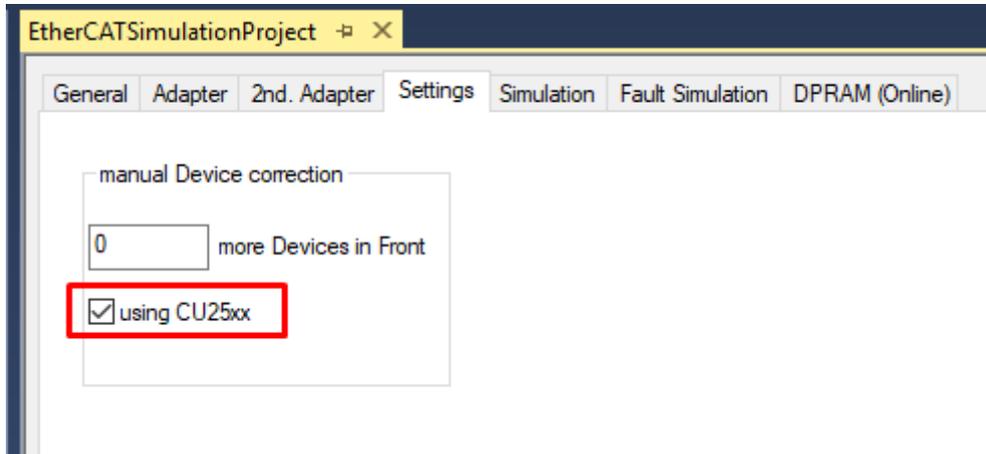
Was ist noch zu tun?

- Anlegen oder Instanzieren der „Simulations“-TwinCAT-Module
- Erstellen einer Task, die die Simulations-Module treibt (mit halber Zykluszeit aufgrund des Abtast-Theorems)
- Verknüpfen der Variablen des Prozessabbildes der Simulationsmodelle mit denen des gespiegelten Prozessabbildes des EtherCAT-Simulation-Devices

6 Verwendung von CU2508-Port-Multipliern

TwinCAT Versionen < TwinCAT 3.1.4024

Der CU2508 darf nur auf der EtherCAT-Master Seite eingesetzt werden (siehe hierzu [Verkabelung \[► 12\]](#)). Ist dies der Fall muss im Reiter Settings die Option **Using CU2508** aktiviert werden.



TwinCAT Versionen > TwinCAT 3.1.4024

Der CU2508 darf nur auf der EtherCAT-Master Seite eingesetzt werden (siehe hierzu [Verkabelung \[► 12\]](#)). Das EtherCAT-Simulation-Device erkennt automatisch, dass auf der Masterseite ein CU2508 verwendet wurde. Die Option „using CU2508“ ist somit nicht mehr erforderlich

HINWEIS

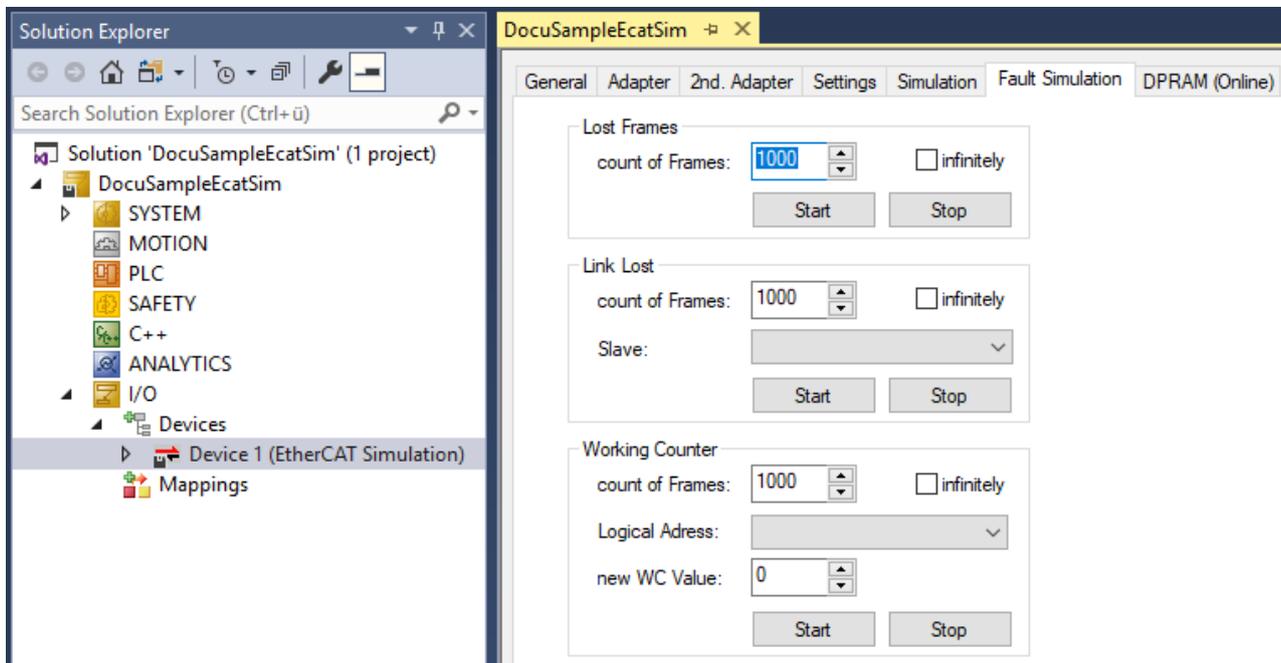
Falsche Werte in der EtherCAT-Diagnose

Wird ein CU2508 auf der Slave-Seite eingesetzt, so ist dieser im TwinCAT-Projekt, welches die Konfigurationsdatei exportiert, nicht enthalten. Darüber hinaus erhält dieser CU2508 erst nach dem EtherCAT-Simulation-Device den EtherCAT-Frame. Er ist somit für das EtherCAT-Simulation-Device „unsichtbar“. Daraus folgt, dass der Device-State und auch der Slave-Count auf der EtherCAT-Master-Seite keine korrekten Werte anzeigen. Eine im Original-Projekt ausprogrammierte EtherCAT-Diagnose zeigt somit falsche Ergebnisse.

7 Fehlersimulation

Die folgenden Fehler können mithilfe des EtherCAT-Simulation-Devices simuliert werden:

- Lost Frames
- Link Lost
- Working Counter Fehler



Simulation von Lost Frames:

1. Wechseln Sie in den Reiter Fault Simulation auf dem EtherCAT Simulation Device
2. Stellen Sie im Eingabe-Feld **count of Frames** die Anzahl der Frames ein, die verloren gehen sollen (Ist diese nicht begrenzt, setzen Sie stattdessen den Haken bei **infinately**).
3. Betätigen Sie den Button **Start**
 - ⇒ Die Simulation von Lost Frames startet
4. Zum Stoppen der Simulation betätigen Sie den Button **Stop**

Simulation von Link Losses:

5. Wechseln Sie in den Reiter Fault Simulation auf dem EtherCAT Simulation Device
6. Stellen Sie im Eingabe-Feld **count of Frames** die Anzahl der Frames ein, für die ein Link-Lost-Fehler gemeldet werden soll (Ist diese nicht begrenzt, setzen Sie stattdessen den Haken bei **infinately**).
7. Stellen Sie in der Auswahlbox **Slave** den EtherCAT-Teilnehmer ein, für welchen der Link Lost Fehler gemeldet werden soll
8. Betätigen Sie den Button **Start**
 - ⇒ Die Simulation der Link-Lost-Fehler startet
9. Zum Stoppen der Simulation betätigen Sie den Button **Stop**

Simulation von Working Counter Fehlern:

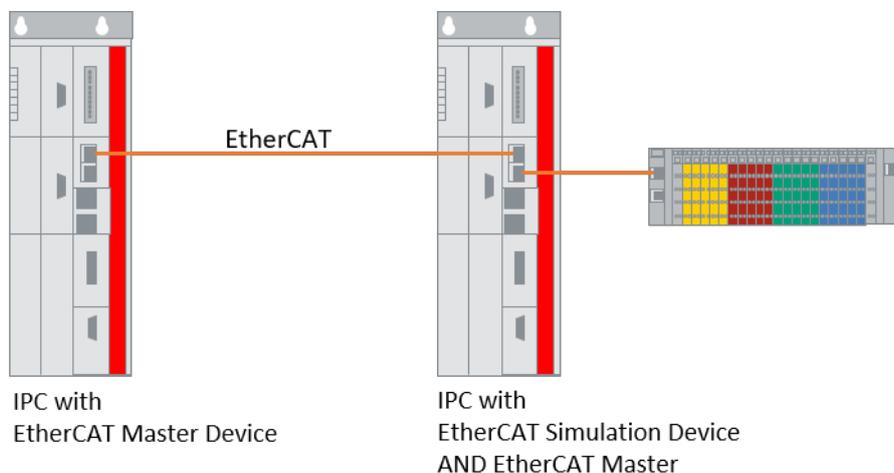
10. Wechseln Sie in den Reiter Fault Simulation auf dem EtherCAT Simulation Device
11. Stellen Sie im Eingabe-Feld **count of Frames** die Anzahl der Frames ein, für die ein Working-Counter-Fehler gemeldet werden soll (Ist diese nicht begrenzt, setzen Sie stattdessen den Haken bei **infinately**).
12. Wählen Sie in der Auswahlbox **Logical Adress** die logische Adresse, die einen falschen Wert zum Working Counter hinzufügen soll.

13. Geben Sie im Feld **WC Value** den neuen (absoluten) Wert an, der hier dem Working Counter zugewiesen werden soll.
14. Betätigen Sie den Button **Start**
 - ⇒ Die Simulation der Working-Counter-Fehler startet
15. Zum Stoppen der Simulation betätigen Sie den Button **Stop**

Siehe hierzu auch das Kapitel [Reiter Fault Simulation \[► 26\]](#)

8 Mischbetrieb von realen und virtuellen Slaves

Mit Hilfe der EtherCAT-Simulation ist es möglich, reale und virtuelle Slaves zu mischen. Dies kann zum Beispiel verwendet werden, um eine Maschine/ Anlage schrittweise in Betrieb zu nehmen. Die folgende Abbildung stellt dies schematisch dar.



Für den Mischbetrieb werden 2 freie Netzwerkschnittstelle am Simulations-IPC benötigt. Die erste Schnittstelle dient als EtherCAT-Simulation-Device und wird mit dem EtherCAT-Kabel zum Steuerungs-Rechner verbunden. Diese wird im Reiter Adapter des EtherCAT-Simulation-Devices eingestellt (siehe hierzu auch Adapter).

Für die Verbindung mit den real existierenden Slaves wird der zweite Adapter auf dem EtherCAT-Simulation Device verwendet (Reiter 2nd. Adapter). Dieser wird analog zum ersten Adapter ausgewählt.

The screenshot shows the configuration window for the '2nd. Adapter' of the EtherCAT-Simulation-Device. The 'General' tab is selected. The fields are as follows:

- Description: [Empty text box]
- Device Name: [Empty text box]
- MAC Address: 00 00 00 00 00 00 [Search... button]
- IP Address: 0.0.0.0 (0.0.0.0) [Compatible Devices... button]
- Freerun Cycle (ms): 1 [Spin box]
- Promiscuous Mode (use with Wireshark only)
- Virtual Device Names

Die Slaves, welche real existieren, müssen im importierten Projekt des EtherCAT-Simulation-Devices deaktiviert werden und in exakt derselben Reihenfolge am EtherCAT-Master des 2nd. Adapter angeschlossen sein. Die oben dargestellte Abbildung soll dies verdeutlichen.

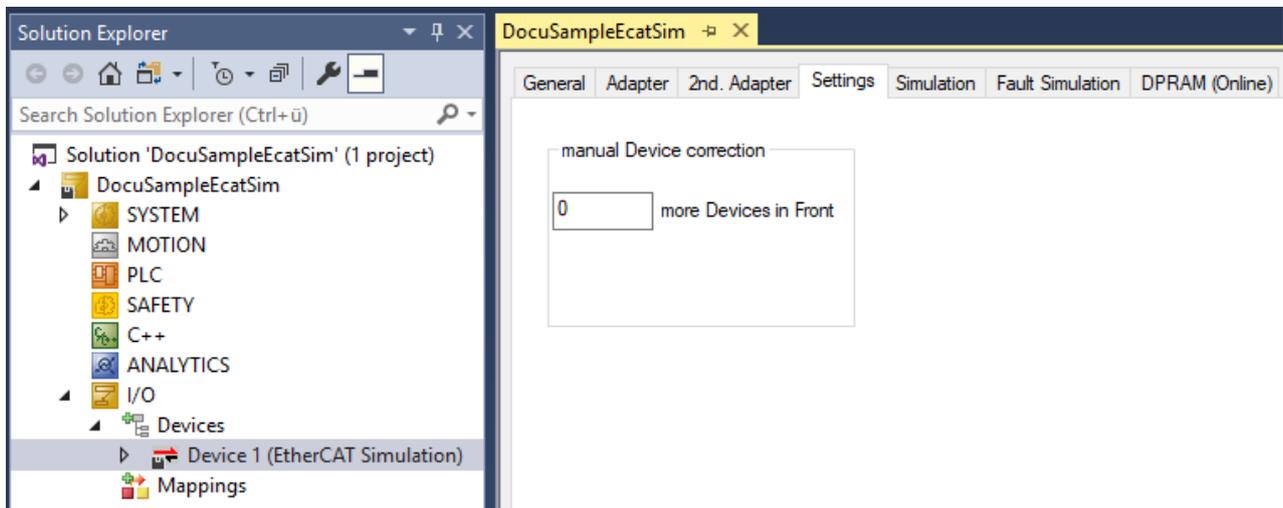
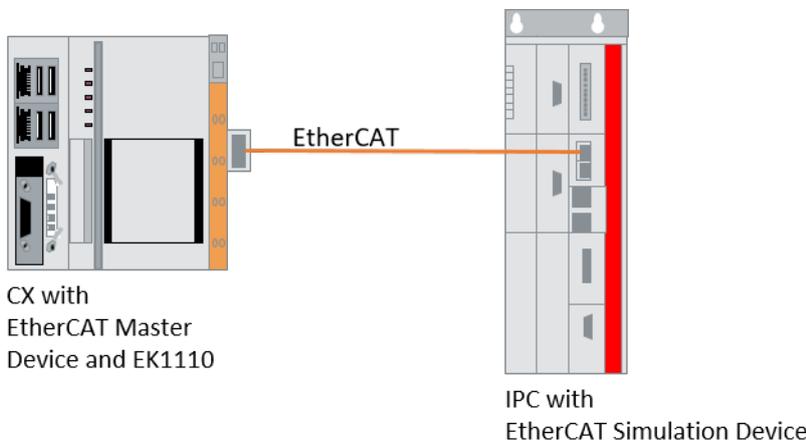
9 Reference, Benutzeroberfläche

Im folgenden Kapitel soll die Benutzeroberfläche des EtherCAT Simulation Devices erklärt werden. Die Reiter Allgemein und Adapter/ 2. Adapter sind dabei dieselben wie bei einem Standard-EtherCAT-Master. Von daher sei an dieser Stelle darauf verwiesen (siehe Adapter).

Der 2. Adapter des EtherCAT Simulation Devices dient zur Konfiguration einer zweiten verwendeten Schnittstelle, wenn man die EtherCAT-Simulation im Mischbetrieb von realen und virtuellen Slaves betreiben möchte (siehe [Mischbetrieb von realen und virtuellen Slaves](#) [▶ 24]).

9.1 Reiter Settings

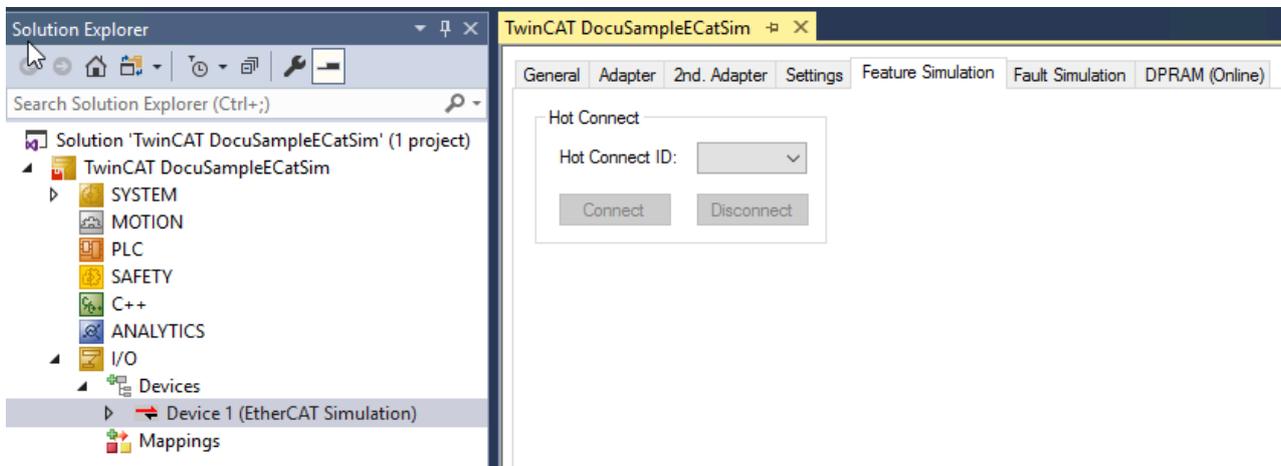
Im Reiter Settings befindet sich die Einstellung manual Device correction. Diese kann verwendet werden, wenn vor der eigentlichen EtherCAT-Simulation noch EtherCAT-Teilnehmer davor geschaltet sind. Dies könnten zum Beispiel Infrastruktur-Devices (z.B. der CU2508) oder intelligente Netzteile (bei CX) sein.



Ab der TwinCAT Version TwinCAT 3.1.4024.0 wird der CU2508 automatisch erkannt. Für dieses Gerät ist somit ab dieser Version KEINE manuelle Gerätekorrektur erforderlich.

9.2 Reiter Feature Simulation

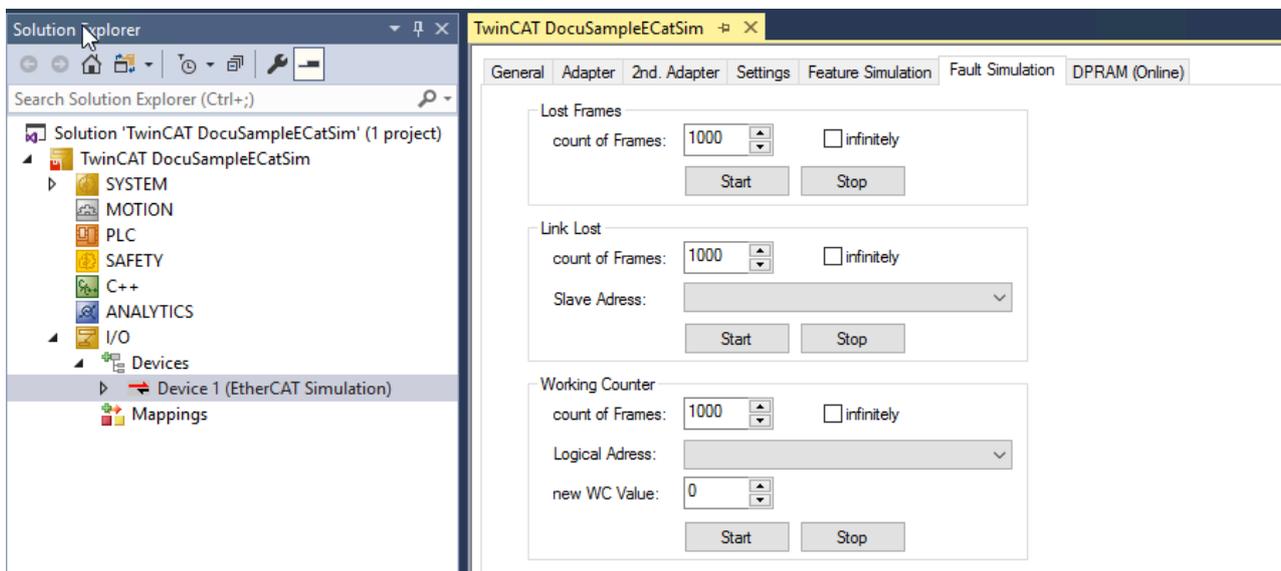
Im Reiter **Simulation** können HotConnect Gruppen simuliert, an- oder abgemeldet werden. Dazu kann die an- bzw. abzumeldende Gruppe aus der Dropbox ausgewählt werden. Durch die Buttons **Connect** bzw. **Disconnect** wird die ausgewählte HotConnect-Gruppe dann an- bzw. abgemeldet.



9.3 Reiter Fault Simulation

Der Reiter Fault Simulation dient zur Simulation von EtherCAT-Fehlern. Die hier zur Verfügung gestellten Möglichkeiten können dazu verwendet werden um die Robustheit der eingestellten EtherCAT-Konfiguration zu testen bzw. die Diagnose der Maschine/ Anlage. Die folgenden Fehler können simuliert werden:

- Lost Frames
- Link Lost
- Working Counter Fehler



Siehe hierzu auch das Kapitel [Fehlersimulation](#) [▶ 22].

Im Folgenden werden die Einstellungen im Reiter Fault Simulation erläutert.

Einstellung	Bedeutung
Lost Frames	
Count of Frames	Anzahl der EtherCAT Frames die verloren gehen sollen
infinitely	Einstellung der Anzahl der EtherCAT Frames auf unendlich
Start	Start der Lost Frames Simulation
Stop	Stoppen der Lost Frames Simulation
Link Lost	
Count of Frames	Anzahl der EtherCAT Frames in denen ein Link Lost gemeldet werden sollen
infinitely	Einstellung der Anzahl der EtherCAT Frames auf unendlich.
Slave	Auswahl des EtherCAT-Slaves der den Link Lost Fehler melden soll
Start	Start der Link Lost Simulation
Stop	Stoppen der Link Lost Simulation
Working Counter	
Count of Frames	Anzahl der EtherCAT Frames in denen ein falscher Working Counter gemeldet werden sollen
infinitely	Einstellung der Anzahl der EtherCAT Frames auf unendlich.
Logical Adress	Logische Adresse an denen der Working Counter falsch berechnet werden soll
New WC Value	Neuer (absoluter) WC Wert, der zugewiesen werden soll
Start	Start der Working Counter Simulation
Stop	Stoppen der Working Counter Simulation

10 SPS API

10.1 ADS-Kommandos

Die in den Reitern [Reiter Feature Simulation \[► 26\]](#) bzw. [Reiter Fault Simulation \[► 26\]](#) zur Verfügung gestellten Möglichkeiten betreffend der Simulation von Hot-Connect-Gruppen bzw. von EtherCAT-Fehlern können auch über ADS (z.B. aus der SPS heraus) aufgerufen werden.

Dazu müssen ADS Write Kommandos an die NetId des EtherCAT-Simulation-Device wie folgt aufgerufen werden:

Hot Connect:

Definierte Index-Groups:

Disconnect Device: 0x0010

Connect Device: 0x0011

Zu verwendete Indexoffsets:

Das High-Word muss 0 sein.

Im Low-Word ist der Identification Value (Hot Connect ID) des Teilnehmers enthalten, den Sie connecten bzw. disconnecten möchten.

Simulation von Lost Frames:

Index-Group: 0x0100

Index-Offset: 0x0001

Daten: 4 Byte Zähler in dem angegeben wird, wie viele Frames ausgelassen werden sollen. 0xFFFFFFFF bedeutet unendlich

Simulation von Link Lost Fehlern:

Index-Group: 0x0100

Index-Offset: 0x0002

Daten: 4 Byte Zähler in dem angegeben wird, wie viele Frames ausgelassen werden sollen. 0xFFFFFFFF bedeutet unendlich

Simulation von Working Counter Fehlern:

Index-Group: 0x0100

Index-Offset: 0x0003

Daten: insgesamt 10 Byte:

- 4 Byte Zähler in dem angegeben wird, wie viele Frames ausgelassen werden sollen. 0xFFFFFFFF bedeutet unendlich
- 4 Byte logische Adresse. Damit wird das Telegramm adressiert welches manipuliert werden soll
- 2 Byte neuer Working Counter. Dieser Wert wird dann in das Telegramm eingetragen

Mehr Informationen:
www.beckhoff.de/te1111

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

