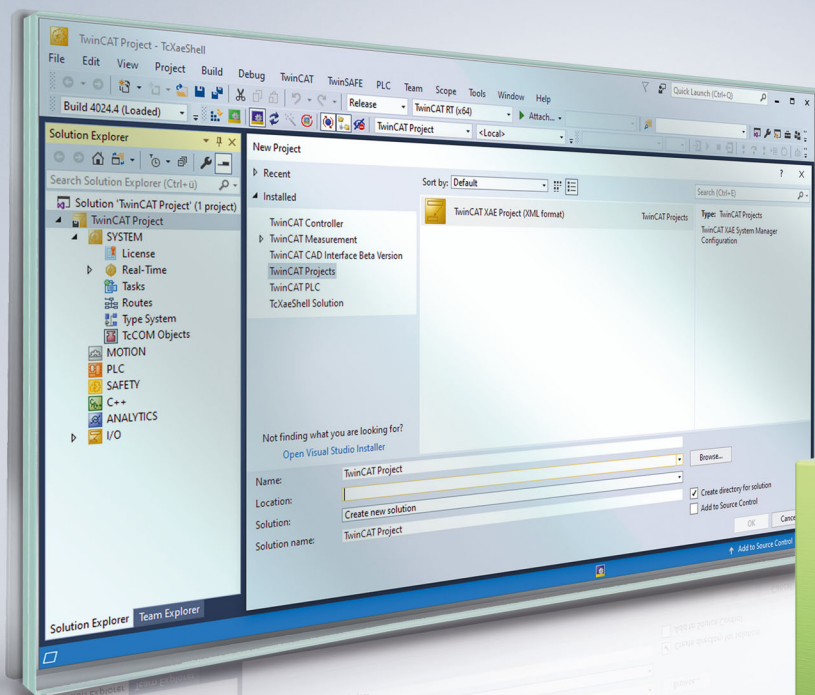


Handbuch | DE

TE1610

TwinCAT 3 | EAP Configurator



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
1.4	Ausgabestände der Dokumentation	7
2	Produktbeschreibung	8
2.1	Übersicht	8
2.2	Das Funktionsprinzip des TwinCAT EAP Configurators	9
2.3	Grundlagen	10
2.3.1	Kommunikationsmethoden.....	12
2.3.2	Gegenstellenüberwachung per ARP	15
2.3.3	Der EAP-Sendemechanismus	15
2.3.4	Performance des EAP.....	18
2.3.5	Die EAP State Machine.....	18
3	Installation	21
3.1	Systemvoraussetzungen	21
3.2	Installation	21
3.3	Lizenzierung	22
4	Erste Schritte.....	23
4.1	Die Beispielprojekte	23
4.2	Exportieren der EDC-Dateien	23
4.3	Anlegen eines TwinCAT EAP Configurator Projektes.....	24
4.3.1	Hinzufügen eines EAP-Geräteobjekts.....	25
4.3.2	Importieren einer EAP Device Configuration (EDC) Datei	26
4.3.3	Aktuelle Konfiguration eines EAP-Gerätes lesen.....	27
4.3.4	Eine Kommunikationsverbindung generieren	28
4.3.5	Die neue/geänderte Konfiguration auf die EAP-Geräte übertragen.....	34
5	Das TwinCAT EAP Configurator Projekt.....	36
5.1	Die grafische Bedienoberfläche	37
5.1.1	Die Architektur.....	37
5.1.2	Die Interaktion mit der grafischen Benutzerschnittstelle (GUI)	38
5.1.3	Die Konfiguration eines EAP-Geräteobjektes	48
5.2	Ein Projekt speichern/laden	60
5.3	Das Objektverzeichnis	61
6	Anhang.....	64
6.1	Anlegen der Beispielprojekte	64
7	Support und Service	74

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

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

Disclaimer

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

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

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

Marken

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

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



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

Fremdmarken

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

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.

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

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Hinweise zur Informationssicherheit

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

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

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

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

1.4 Ausgabestände der Dokumentation

Version	Änderungen
1.6.x	TwinCAT 3.1 Build 4026 • Kurz-Information zur Installation

2 Produktbeschreibung

Der TwinCAT EAP Configurator ist ein Werkzeug zur Veranschaulichung und Konfiguration von Kommunikationsnetzwerken, in denen der Datenaustausch mit Hilfe des EtherCAT Automation Protocols (EAP) erfolgt oder hergestellt werden soll. EAP wird für die Master-Master-Kommunikation eingesetzt.

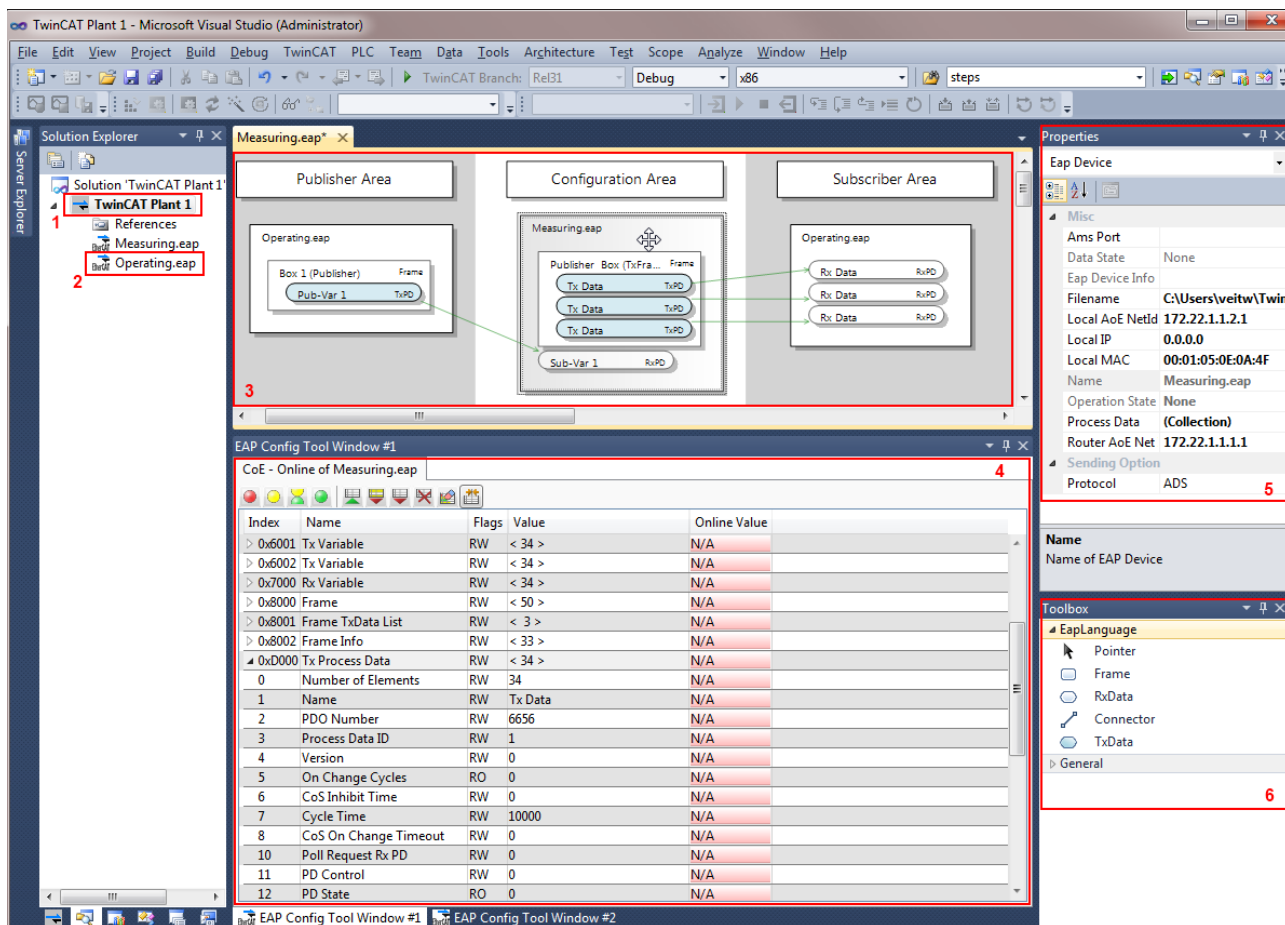
Der TwinCAT EAP Configurator bietet eine grafische Oberfläche an, die zum einen zur überschaubaren Darstellung der bestehenden Kommunikationsverbindungen dient und zum anderen eine bequeme Möglichkeit bietet, neue Kommunikationsverbindungen zu konfigurieren.

Zur Darstellung der bestehenden Kommunikationsverbindungen wird die aktuelle Konfiguration der beteiligten EAP-Geräte gelesen. Anschließend besteht die Möglichkeit, die Konfiguration zu modifizieren und wieder auf die EAP-Geräte zu übertragen. Die Übertragung einer Konfiguration wird mit Hilfe des TwinCAT EAP Configurators durchgeführt, ohne das betreffende TwinCAT System zu stoppen. Nur das betroffene EAP-Gerät muss für die Dauer der Übertragung den eigenen zyklischen Verarbeitungsprozess unterbrechen. Während der Übertragung wird dann die modifizierte Konfiguration in das EAP-Gerät geschrieben.

Der TwinCAT EAP Configurator integriert sich wie TwinCAT 3 im Microsoft Visual Studio. So ist es möglich, TwinCAT 3 Projekte und TwinCAT EAP Configurator Projekte parallel in einer Visual Studio Solution zu verwenden. Der TwinCAT EAP Configurator kann aber auch als eigenständiges Werkzeug installiert werden.

2.1 Übersicht

Der TwinCAT EAP Configurator ist ein Visual Studio Package und integriert sich nahtlos in das Microsoft Visual Studio. Die folgende Übersicht zeigt die einzelnen Komponenten des EAP Configurators:



- **Solution Explorer:**

Der Solution Explorer gehört generell zum Visual Studio und erlaubt es, alle Elemente innerhalb der Solution und der darunter angeordneten Projekte anzusehen und zu verwalten. Einem TwinCAT EAP Configurator Projekt (siehe Übersicht Nr. 1) sind Elementknoten untergeordnet, die jeweils einer Datei entsprechen (siehe Übersicht Nr. 2), welche die Konfiguration eines EAP-Gerätes beschreibt.

- **Grafischer Editor:**
Der grafische Editor (siehe Übersicht Nr. 3) stellt die Konfiguration aller EAP-Geräte aus dem EAP Configurator Projekt grafisch dar. Mit Hilfe des grafischen Editors lassen sich die Kommunikationsverbindungen zwischen den EAP-Geräten übersichtlich nachvollziehen und die Konfigurationen der beteiligten EAP-Geräte ändern.
- **EAP Config Tool Window:**
Eine vollständige Übersicht aller Konfigurationsdaten eines EAP-Gerätes lässt sich über eine zweistufige Tabelle in dem EAP Config Tool Window anzeigen (siehe Übersicht Nr. 4), dem sogenannten Objektverzeichnis.
Außerdem befinden sich in dieser Komponente noch die Bedienelemente, um eine veränderte Konfiguration auf das EAP-Gerät zu übertragen, so dass diese wirksam wird.
- **Properties Window:**
Die wichtigsten Eigenschaften eines grafischen Objektes, das mit der Maus im grafischen Editor angeklickt wird, werden im Properties Window (siehe Übersicht Nr. 5) angezeigt. An dieser Stelle können die Eigenschaften dann den Anforderungen entsprechend verändert werden.
- **Toolbox Window:**
Das Toolbox Window (siehe Übersicht Nr. 6) bietet dem Anwender die notwendigen Werkzeuge, um einem EAP-Gerät neue Objekte hinzuzufügen oder neue Kommunikationsverbindungen zu erzeugen. Dies ist z. B. dann notwendig, wenn weitere Publisher oder Subscriber Variablen konfiguriert werden sollen.

2.2 Das Funktionsprinzip des TwinCAT EAP Configurators

Bei einem TwinCAT EAP-Gerät sind alle Konfigurationsparameter anhand eines Objektverzeichnisses organisiert. Dieses EAP Objektverzeichnis (engl. Object Dictionary) ist wie ein CANOpen Object Dictionary strukturiert. Die Struktur für ein CANOpen Object Dictionary wird von der CiA-Organisation (CiA = CAN in Automation) als Profil bezeichnet. Das Profil für das EAP Objektverzeichnis ist von der EtherCAT Technology Group (ETG) in der Spezifikation des EtherCAT Automation Protokolls festgelegt worden (ETG 1005). In dem Profil sind verschiedene Objekte definiert. Jedem Objekt ist ein eindeutiger 16-Bit-Index zugeordnet. Ein Objekt kann wiederum bis zu 255 Subeinträge haben, und für einen Subeintrag muss ein Datentyp definiert sein.

Der TwinCAT EAP Configurator kann die Objekte des EAP-Objektverzeichnisses aus einem EAP-Gerät auslesen und auf Basis der EAP-Profilspezifikation interpretieren. Auf diese Weise ist der EAP Configurator in der Lage die Kommunikationsverbindungen zwischen allen ausgelesenen EAP-Geräten darzustellen. Wenn die Konfiguration von einem EAP-Gerät verändert wird, werden die gelesenen Objekte inhaltlich modifiziert, gelöscht oder auch zusätzlich Objekte erzeugt. Anschließend kann der modifizierte Satz von Objekten wieder auf die jeweiligen TwinCAT EAP-Geräte geschrieben werden, so dass die Änderungen übernommen werden.

Voreingestellt kommuniziert der TwinCAT EAP Configurator beim Lesen und Schreiben der Objekte über das ADS/AMS Protokoll mit einem TwinCAT EAP-Gerät. ADS steht für *Automation Device Specification* und beschreibt eine geräte- und feldbusunabhängige Schnittstelle. AMS steht für *Automation Message Specification* und ermöglicht es, zentrale und dezentrale Systeme zu adressieren wie PCs aber auch Buscontroller. ADS/AMS ist von Beckhoff spezifiziert worden und wird vom TwinCAT Router unterstützt. Nachrichten, die über die Rechnergrenze hinaus in einem Netzwerk verschickt werden, werden per TCP/IP übertragen. Daher kann der TwinCAT EAP Configurator per ADS/AMS sowohl mit einem TwinCAT EAP-Gerät kommunizieren, das lokal auf dem gleichen PC existiert als auch mit einem TwinCAT EAP-Gerät, welches auf einem anderen PC ausgeführt wird, der im Netzwerk eingebunden ist.

Alternativ kann der TwinCAT EAP Configurator das Lesen und Schreiben der Objekte auch per AoE Protokoll (AoE = ADS over EtherCAT) durchführen. Für die Vermittlung und den Transport der AoE Nachrichten werden die Protokolle IP und UDP verwendet. Auf dieser Grundlage kann jedes Gerät innerhalb des Netzwerks per IP Adresse erreicht werden. Die Spezifikation des AoE Protokolls ist in den EtherCAT Protocol Enhancements (ETG 1020) zu finden.

Der Unterschied zwischen der ADS/AMS Kommunikation und AoE Kommunikation ist der, dass bei der AoE Kommunikation im Gegensatz zur ADS/AMS Kommunikation kein TwinCAT Router notwendig ist. Das AoE Protokoll ist eines der Protokolle, die in TwinCAT unter die Rubrik der Mailboxkommunikation eingeordnet werden.

i Systemkonfiguration beachten

Wenn der TwinCAT EAP Configurator und das TwinCAT EAP-Gerät auf ein und demselben Rechner laufen, funktioniert die Kommunikation zwischen diesen beiden Komponenten nur mittels ADS/AMS Protokoll. TwinCAT EAP-Geräte, die auf einem anderen Rechner laufen, können auch per AoE-Protokoll erreicht werden.

Beide verwendeten Protokolle werden von TwinCAT azyklisch verarbeitet und dienen allein der Übertragung von Konfigurationsdaten vom oder zum EAP-Gerät. Allein die Prozessdaten eines EAP-Gerätes werden zyklisch und hoch deterministisch verarbeitet bzw. übertragen. Die Konfiguration eines EAP-Gerätes definiert dabei den Aufbau, die Größe und die Übertragungsart der Prozessdaten.

2.3 Grundlagen

Das TwinCAT EAP-Gerät ermöglicht die Übertragung von Daten aus beliebigen Variablen von einer TwinCAT Steuerung A zu einer TwinCAT Steuerung B via Netzwerk. Diese Variablen dienen typischerweise dazu, den Verarbeitungsprozess innerhalb einer Maschine zu steuern. Sie werden daher auch Prozessvariablen (PV) genannt. Das Senden und Empfangen kann für ein TwinCAT EAP-Gerät über einen normalen Netzwerkadapter erfolgen, der von dem TwinCAT Realtime Ethernet Treiber unterstützt wird.

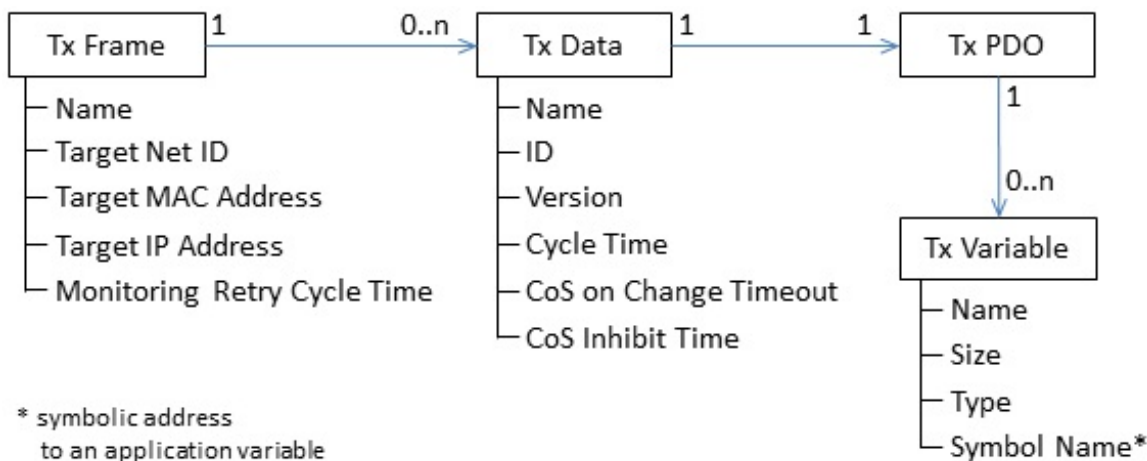
Die Kommunikation zwischen EAP-Geräten erfolgt nach dem Publisher Subscriber Prinzip, bei dem die Sender, Publisher genannt, ihre Nachrichten an alle oder mehrere Netzwerkteilnehmer schicken, wobei ein Publisher grundsätzlich nicht weiß, wer seine Empfänger sind bzw. ob es überhaupt einen Empfänger gibt. Auf der anderen Seite gibt es Empfänger, Subscriber genannt, die sich für bestimmte Nachrichten interessieren und diese empfangen, ohne zu wissen, von welchem Publisher diese kommen bzw. ob es überhaupt einen Publisher gibt, der eine solche Nachricht verschickt.

Aufbau eines EAP-Publishers

Ein EAP-Publisher setzt sich aus einer Summe geschachtelter Elemente zusammen, so wie in der folgenden Abbildung grafisch veranschaulicht. Das Basiselement auf der untersten Ebene ist eine *TxVariable*. Sie definiert eine Ausgangsvariable, die mit einer Prozessvariablen verknüpft wird und beinhaltet ein paar weitere Eigenschaften wie z. B. einen Datentyp. Der Datentyp darf frei gewählt werden; es darf also auch ein komplexer Datentyp sein, der mehrere hundert Bytes groß ist. Es sollte gewährleistet werden, dass die maximale Größe eines EAP-Frames nicht überschritten wird (die Größe eines EAP-Frames entspricht der eines Standard Ethernet Frames). Bei Überschreiten der Größe des Standard Ethernet Frames werden mehrere Pakete versendet, welche beim Empfänger auch in unterschiedlicher Reihenfolge eintreffen können.

i In TwinCAT 3.1 Build 4024 wird das Aufteilen auf mehrere Pakete nicht unterstützt. Beim Engineering erscheint eine Fehlermeldung.

Die Prozessvariable liefert während des Betriebs die Werte, die der Publisher senden soll.



Eine Ebene darüber werden in den *TxPDO* Elementen (*TxPDO* = *TxProcessDataObjects*) *TxVariablen* referenziert. Ein *TxPDO* kann mehrere *TxVariablen* referenzieren und fasst sie so zu einem Objekt zusammen. Das *TxPDO* definiert dann eine geordnete Menge von *TxVariablen*. Für ein *TxPDO* muss ebenfalls gewährleistet sein, dass die maximale Größe eines EAP-Frames nicht überschritten wird.

Noch eine Ebene höher befindet sich das *TxData* Element (*TxProcessData* = *TxPD*), welches eine *Publisher Variable* repräsentiert und beim EAP als Kommunikationseinheit des Publishers begriffen wird. Das *TxData* referenziert ein bestimmtes *TxPDO* und definiert einige Eigenschaften, wie z.B. die *ID* der *Publisher Variablen*, deren *Version* und den Zyklustakt, mit dem die *Publisher Variable* überhaupt erst versendet wird. Mithilfe dieser Eigenschaften definiert die *Publisher Variable* ein Objekt auf der Senderseite, zu der eine passende *Subscriber Variable* auf der Empfängerseite definiert werden muss, so dass ein erfolgreicher Datenaustausch stattfinden kann.

Die Übertragung der Daten erfolgt netzwerkbasierend über das Ethernet Protokoll oder über UDP/IP. Einem *TxFrame* wird dann entsprechend eine Liste von *TxData* zugeordnet, die an die gleiche Zieladresse gesendet werden sollen. Ein *TxFrame* ist dabei auf eine maximale Datenlänge pro Datenpaket beschränkt. Für das Versenden einer *Publisher Variable* müssen dann wenigstens die folgenden Eigenschaften definiert sein:

Zieladresse:

Als Zieladresse wird in der Regel eine Multicast Adresse eingetragen, so dass die *Publisher Variable* automatisch an eine Gruppe von Empfängern gesendet wird. Es kann allerdings auch die Adresse eines einzelnen Empfängers eingetragen werden.

ID:

Für jede *Publisher Variable* wird eine Nummer definiert, die eindeutig im gesamten Netzwerk sein muss und anhand derer die *Publisher Variable* von einem *Subscriber* identifiziert werden kann.

Zyklustakt:

Anhand des Zyklustaktes wird festgelegt, in welchem zeitlichen Rhythmus die *Publisher Variable* versendet wird. Beim EAP sind im Allgemeinen Zykluszeiten von wenigen Millisekunden bis zu mehreren 100 Millisekunden üblich.

Verknüpfung zu einer Prozessvariablen:

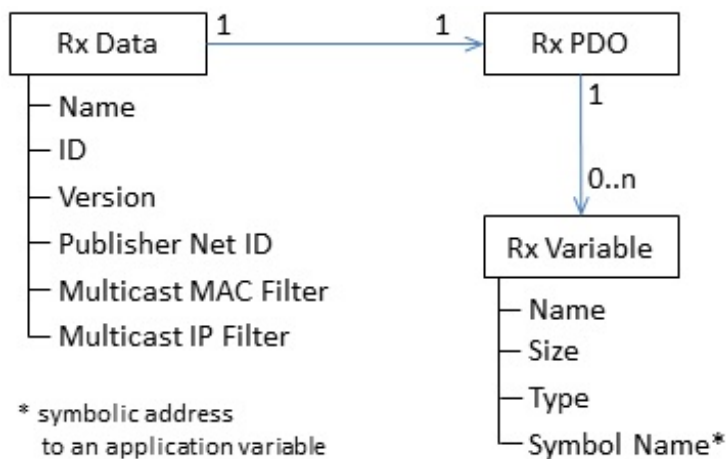
Zu guter Letzt wird noch eine Verknüpfung der *Publisher Variablen* mit einer *Prozessvariablen* benötigt, damit auch tatsächlich Prozessdaten mit Hilfe der *Publisher Variable* verschickt werden. Andernfalls bliebe der Wert der *Publisher Variable* stets null.

Aufbau eines EAP-Subscribers

Die Struktur eines EAP-Subscribers ist analog zu der eines Publishers und wird in der folgenden Abbildung grafisch veranschaulicht. Das Basiselement auf der untersten Ebene eines Subscribers wird dann allerdings *RxVariable* genannt. Die *RxVariable* definiert eine Eingangsvariable, die ebenfalls mit einer Prozessvariablen verknüpft wird und ein paar Eigenschaften beinhaltet wie z.B. den Datentyp. Die Prozessvariable erhält während des Betriebs die Werte, die der Subscriber empfängt.

Die Elemente der darüber liegenden Ebene werden dann entsprechend *RxPDOs* (*RxProcessDataObjects*) genannt, die jeweils eine geordnete Menge von *RxVariablen* definieren.

Noch eine Ebene höher befindet sich das *RxData* Element (*RxProcessData* = *RxPD*), welches eine *Subscriber Variable* repräsentiert und beim EAP als Kommunikationseinheit des Subscribers begriffen wird. Das *RxData* referenziert ein bestimmtes *RxPDO* und definiert u.a. die notwendigen Eigenschaften (ID und Version), die mit der *Publisher Variablen* übereinstimmen müssen, die empfangen werden soll. Für einen erfolgreichen Datenaustausch müssen die Datentypen der referenzierten *RxVariable* und deren Ordnung im *RxPDO* identisch zum *TxPDO* der *Publisher Variable* sein. Die *Subscriber Variable* definiert somit ein Objekt auf der Empfangsseite, zu der eine passende *Publisher Variable* auf der Senderseite definiert werden muss, so dass ein Datenaustausch stattfindet.



Dem Design des EtherCAT Automation Protocols nach ist es für einen Subscriber unerheblich, von welchem Absender die empfangenen Daten stammen. Vor allem ist es unerheblich, welche *Publisher Variablen* innerhalb eines Frames versendet werden. Aus diesem Grund gibt es beim Subscriber auch kein Frame Element oder ähnliches, mit Hilfe dessen bestimmte *Subscriber Variablen* zu einer Einheit zusammengefasst und somit nur en bloc empfangen würden. Nichtsdestotrotz bietet das *RxData* die Möglichkeit eine *AMS NetID* als Filteradresse zu definieren, wenn ein Subscriber nur die *Publisher Variablen* eines bestimmten Absenders empfangen soll. Für eine *Subscriber Variable* müssen dann wenigstens die folgenden Eigenschaften definiert sein:

ID:

Die *ID* bei einer *Subscriber Variable* definiert, welche *Publisher Variable* diese empfangen soll. Die *ID* ist eine Nummer, die nach Möglichkeit eindeutig für jede *Publisher Variable* im gesamten Netzwerk ist und anhand derer die *Publisher Variable* beim Empfang identifiziert wird.

Verknüpfung zu einer Prozessvariablen:

Letztlich ist auch die Verwendung einer *Subscriber Variablen* erst dann sinnvoll, wenn diese mit einer Prozessvariablen verknüpft ist. Erst dann werden die empfangenen Daten auch tatsächlich von der Prozessvariablen übernommen und bei der Steuerung der Maschine berücksichtigt.

Außerdem muss sichergestellt sein, dass die Datenlänge der *Subscriber Variable* identisch zur Datenlänge der *Publisher Variable* ist. Andernfalls wird die empfangene *Publisher Variable* verworfen.

2.3.1 Kommunikationsmethoden

Das TwinCAT EAP-Gerät unterstützt zwei Kommunikationsarten. Zum einen die zyklische Prozessdatenkommunikation (EtherCAT Type 4) und zum anderen die azyklische EtherCAT Mailbox-Kommunikation (EtherCAT Type 5). Das TwinCAT EAP-Gerät unterstützt bei der Mailbox-Kommunikation nur das AoE-Protokoll (AoE – ADS over EtherCAT). Die Spezifikation des AoE-Protokolls wird in den EtherCAT Enhancements (ETG 1020) beschrieben. Bei der Prozessdatenkommunikation wird zwischen zwei Kommunikationsmodi unterschieden:

Zum einen den **Pushed Data Exchange** Modus, bei dem ein EAP-Sender zyklisch oder wegen einer Statusänderung seine Prozessinformationen in das Netz sendet und ein EAP-Empfänger entsprechend diese Prozessinformationen erwartet und empfängt. Dieser Modus entspricht dem Publisher Subscriber Prinzip der Netzwerkvariablen (NWV) von TwinCAT 2.

Zum anderen den **Polled Data Exchange** Modus, bei dem ein EAP-Client ein Anfragetelegramm mit seinen Prozessinformationen an einen EAP-Server sendet und dieser wiederum seine Prozessinformationen in einem Antworttelegramm an den EAP-Client zurückschickt.

Des Weiteren unterstützt das TwinCAT EAP-Gerät sowohl unterschiedliche Verbindungsarten als auch unterschiedliche Adressierungsarten bei der Prozessdatenkommunikation. Die unterstützten Verbindungsarten sind:

- **Unicast:** Die EAP-Nachricht wird von einem Endpunkt zu einem anderen Endpunkt geschickt, mit anderen Worten: Die Nachricht ist an genau einen PC adressiert.
- **Multicast:** Die EAP-Nachricht wird von einem Endpunkt zu mehreren anderen Endpunkten geschickt, mit anderen Worten: Die Nachricht ist an eine Gruppe von PCs adressiert.

- **Broadcast:** Die EAP-Nachricht wird von einem Endpunkt zu allen erreichbaren Endpunkten geschickt, mit anderen Worten: Die Nachricht ist an alle Teilnehmer adressiert.

Die Adressierung kann per MAC-Adresse, per AMS NetID oder per IP-Adresse erfolgen. Je nach Konstellation von Verbindungsart und Adressierungsart wird ein bestimmtes Vermittlungsprotokoll bei der EAP-Prozessdatenkommunikation wirksam. Die genaue Zuordnung ist in der folgenden Tabelle festgehalten.

Vermittlungsprotokolle

Adressierungsart Verbindungsart	MAC-Adresse	AMS NetID	IP-Adresse
Unicast	Ethernet Protokoll	Ethernet Protokoll	UDP/IP
Multicast	Ethernet Protokoll	Nicht möglich	UDP/IP
Broadcast	Ethernet Protokoll	Nicht möglich	UDP/IP

Im Detail werden die Verbindungsarten Unicast, Multicast und Broadcast in Abhängigkeit von den verschiedenen Adressierungsarten (MAC, AMS NetID und IP) wie folgt unterstützt:

MAC-Adressierung:

Die EAP-Nachricht wird per Ethernet Protokoll übertragen. Als Zieladresse wird die MAC-Adresse des Netzwerkadapters konfiguriert, der die Nachricht empfangen soll. Bei dieser Adressierungsart kann die EAP-Nachricht nicht von einem Router in ein anderes Subnetz weitergeleitet werden, da dieser auf der Basis von IP-Adressen arbeitet. Daher kann die Nachricht nur innerhalb eines Subnetzes über Switches versendet werden.

● Broadcast und Multicast



Für eine Broadcast oder Multicast Nachricht sind spezielle MAC-Adressen reserviert:

Broadcast MAC: FF-FF-FF-FF-FF-FF

Multicast MAC: Eine Multicast MAC-Adresse muss folgende Bedingungen erfüllen.

- Das niederwertigste Bit (Bit 1) des ersten Bytes hat den Wert 1 (Group-Bit).
 - Das folgende Bit 2 hat den Wert 0, wenn die MAC-Adresse global eindeutig ist; oder den Wert 1, wenn die Adresse nur lokal eindeutig ist.
 - Die ersten 24 Bits (Bit 3 bis 24) entsprechen der Herstellerkennung (engl. Organizationally Unique Identifier - OUI). Die OUI der Firma Beckhoff ist "00-01-05".
 - Die verbleibenden 24 Bits (Bit 25 bis 48) können individuell für jede Schnittstelle festgelegt werden. Für das EtherCAT Automation Protocol ist die Folge "04-00-00" definiert.
- ⇒ Somit ergibt sich die Standard Multicast MAC-Adresse 01:01:05:04:00:00 für TwinCAT EAP-Geräte.

AMS NetID-Adressierung:

Die EAP-Nachricht wird per EtherCAT Protokoll vom Typ 4 (EAP) übertragen. Die notwendige Ziel MAC-Adresse wird mittels Address Resolution Protocol (ARP) unter Verwendung der konfigurierten AMS NetID ermittelt. Wie bei der MAC-Adressierung kann die EAP-Nachricht nur innerhalb des Subnetzes versendet werden.

● Kommunikation via AMS NetID



Die Verwendung einer *AMS NetID* als Zieladresse hat den Vorteil, dass es sich dabei um eine logische Adressierung handelt. Die *MAC-Adresse* des Zielgerätes wird mit Hilfe eines speziellen ARP Request unter Verwendung der konfigurierten *AMS NetID* ermittelt.

Ein Anpassen der Konfiguration einer EAP-Verbindung ist auch dann nicht notwendig, wenn beispielsweise ein Steuerungsrechner bzw. ein Netzwerkadapter eines Rechners ausgetauscht wird auf Grund dessen sich die *MAC-Adresse* ändert. Es muss nur sichergestellt sein, dass der neue Steuerungsrechner die ursprüngliche *AMS NetID* erhält.

Wenn die Verbindungsart *Unicast* konfiguriert wird, ist standardmäßig auch der *Subscriber Monitoring Mechanismus* konfiguriert (siehe [Gegenstellenüberwachung per ARP \[► 151\]](#)).

IP-Adressierung:

Für die EAP-Nachricht wird das Internet Protokoll (IP) zusammen mit dem User Datagram Protocol (UDP) zur Vermittlung bzw. Adressierung des Empfängers verwendet. Die notwendige Ziel MAC-Adresse wird mittels Address Resolution Protocol (ARP) unter Verwendung der konfigurierten IP-Adresse ermittelt. Mit Hilfe der UDP/IP-Adressierung kann die EAP-Nachricht auch von einem Router in andere Subnetze weitergeleitet werden (z .B. auch ins Internet).

Für eine Broadcast oder Multicast Nachricht sind spezielle IP-Adressen reserviert:

Broadcast IP: Als Broadcast IP-Adresse ist 255.255.255.255 festgelegt. Aus dieser wird direkt die Broadcast MAC Adresse FF-FF-FF-FF-FF-FF abgeleitet.

Multicast IP: Eine Multicast IP Adresse muss in dem Adressbereich 224.0.0.0 bis 239.255.255.255 (IPv4) liegen. TwinCAT generiert beim EAP-Gerät zu jeder konfigurierten *Multicast IP Adresse* eine konforme *Multicast MAC Adresse*, die bei der Inbetriebsetzung von TwinCAT (also beim Schalten in den Run Mode) verwendet wird.

Pushed Data Exchange (n:m Verbindung)

Der Pushed Data Exchange Modus funktioniert nach dem gleichen Modell wie die Übertragung der NWV (Publisher Subscriber Prinzip). Er bietet die Möglichkeit einer n:m-Verbindung in einem Netzwerk. Dabei kann jedes EAP-Gerät ein oder mehrere EAP-Telegramme mit seinen Ausgangsprozessdaten (*TxData*) senden. Ebenso kann jedes EAP-Gerät darauf „lauschen“, ob die in einem empfangenen EAP-Telegramm enthaltenen Prozessdaten zu seinen Eingangsprozessdaten (*RxData*) passen und diese gegebenenfalls verarbeiten. Ein und dasselbe EAP-Gerät kann also sowohl Prozessdaten senden als auch Prozessdaten empfangen. Auf diese Weise lässt sich eine bidirektionale Kommunikation herstellen.

Beim Pushed Data Exchange kann die Adressierungsart (Unicast, Multicast oder Broadcast) pro konfiguriertes EAP-Telegramm entsprechend der Notwendigkeit frei gewählt werden.

Polled Data Exchange (1:1 Verbindung)

Der Polled Data Exchange Modus unterliegt dem Prinzip der Client/Server Architektur. Er ermöglicht mit Hilfe dieser Architektur eine „weiche“ Synchronisierung. Dabei kann ein EAP-Gerät gleichzeitig sowohl als Client als auch als Server agieren.

● Verbindungsart für den Polled Modus

Für den Modus Polled Data Exchange ist ausschließlich die Verbindungsart Unicast eindeutig definiert.

Unicast (1:1-Verbindung)

Ein Client sendet ein EAP-Telegramm mit seinen Ausgangsprozessdaten an einen Server, der daraufhin seine Eingangsprozessdaten in einem eigenen EAP-Telegramm an den Client zurückschickt.

Vermittlungsprotokolle**Ethernet Protocol**

Das Ethernet Protocol ist für die Vermittlung der Datenpakete im Netzwerk verantwortlich. Es übernimmt die Aufgaben der OSI-Schichten 1 und 2 (Bitübertragungsschicht und Sicherungsschicht). Im Ethernet Protocol Header wird eine Absender- und Empfängeradresse eingetragen sowie ein Ethernet-Type, der festlegt, welches Protokoll der nächsthöheren OSI-Schicht auf das Ethernet Protocol aufsetzt. Die Absender- und Empfängeradresse wird in Form einer MAC-Adresse eingetragen. MAC bedeutet *Media Access Control* und steht hier für die eindeutige Hardware-Adresse, die jedes Ethernet Gerät ab Herstellung hat. Der Ethernet-Port eines Beckhoff-PC könnte z. B. die MAC-Adresse 00:01:05:34:05:84 haben - "00:01:05" ist die Beckhoff-Kennung, der Rest wird bei der Herstellung gewählt. Über die Absender- und die Empfänger-MAC-Adresse ist der Weg eines jeden Ethernet-Telegramms zwischen zwei sich im Netzwerk befindlichen PC bestimmt. Ein Ethernet-Telegramm kann über einen Switch, normalerweise aber nicht über einen Router weiterverarbeitet werden.

User Datagram Protocol / Internet Protocol (UDP/IP)

Der Empfänger wird über einen zusätzlichen IP-Header im Ethernet-Telegramm identifiziert und kann auf diese Weise von einem Router weiterverarbeitet werden. Das Telegramm hat den Ether Type 0x0800,

wodurch festgelegt ist, dass es sich um ein IP-Telegramm handelt. Im anschließenden UDP-Header wird dann für den Source-Port sowie für den Destination-Port die Portnummer 0x88A4 verwendet. Durch diese Portnummer erkennt das TwinCAT System, dass es sich um ein Echtzeitbasiertes User Datagram handelt.

Ein EAP-Telegramm wird von TwinCAT entweder anhand des Ether Types 0x88A4 (beim Ethernet Protocol) oder anhand des Destination-Ports 0x88A4 (beim UDP/IP) identifiziert. Entsprechend schleust der TwinCAT Ethernet Treiber ein empfangenes EAP-Telegramm am NDIS Stack des Betriebssystems vorbei, so dass es von TwinCAT bevorzugt als Beckhoff Echtzeit Ethernet Telegramm behandelt wird. Beim Versenden eines EAP-Telegramms wird ebenfalls der NDIS Stack des Betriebssystems umgangen.

Nachdem ein EAP-Telegramm von einem TwinCAT PC empfangen und als solches identifiziert wurde, findet in der weiteren Telegrammverarbeitung die Zuordnung der im Telegramm transportierten *Process Data (PD)* zu den beim EAP-Gerät konfigurierten RxData statt. Diese Zuordnung erfolgt anhand der *PD-ID*. Das empfangene *PD* wird verworfen, wenn beim Empfänger kein RxData mit der passenden *PD-ID* konfiguriert wurde.

Die Werte der einzelnen *Process Variables (PV)* eines *PD* werden erst dann übernommen, wenn auch noch die Datenlänge und die Versionsnummer des empfangenen *PD* mit der erwarteten Datenlänge und Versionsnummer übereinstimmen.

2.3.2 Gegenstellenüberwachung per ARP

Das EAP setzt auf den verbindungslosen Protokollen (Ethernet Protocol sowie UDP/IP) auf. Diese Protokolle liefern keine Empfangsbestätigung für eine Nachricht. Damit der Sender eines EAP-Telegramms mitbekommen kann, dass der Empfänger nicht mehr erreichbar ist, wird vom TwinCAT EAP-Gerät das Address Resolution Protocol (ARP) zur Gegenstellenüberwachung eingesetzt. Mit Hilfe der Eigenschaft *ARP Retry Interval* kann bei einem EAP-Publisher konfiguriert werden, in welchem Zeitraster geprüft werden soll, ob der Empfänger noch erreichbar ist. Die Gegenstellenüberwachung (*Subscriber Monitoring*) kann nur aktiviert werden, wenn eine Unicast-Verbindung konfiguriert ist.

Beim *Subscriber Monitoring* schickt der Publisher entsprechend dem konfigurierten Zeitintervall ein *ARP Request* Telegramm an das konfigurierte Zielgerät. Arbeitet der Empfänger noch wie erwartet, antwortet dieser mit einem *ARP Reply* Telegramm. Andernfalls bleibt die Antwort aus. In der Diagnosevariable *FrameState* (siehe Publisher) wird im Fehlerfall das 3. Bit gesetzt (0x0004).

● Das ARP-Handling

i Das ARP-Handling für die Zuordnung von MAC-Adressen zu Netzwerkadressen (IP-Adressen) wird vom Betriebssystem (Windows) übernommen. Das ARP-Handling für die Zuordnung von MAC-Adressen zu AMS NetIDs wird hingegen vom TwinCAT System übernommen.

2.3.3 Der EAP-Sendemechanismus

Das Senden eines EAP-Telegramms wird mit Hilfe eines Auslösemechanismus (Auslöser - engl. *Trigger*) angestoßen. Mit Hilfe der Konfiguration eines EAP-Gerätes wird bestimmt, wie dieser *Trigger Mechanismus* arbeiten soll. Dazu wird für jedes *TxData* eine *Trigger* Bedingung definiert. Wenn diese *Trigger* Bedingung erfüllt ist, wird das *TxData* per EAP-Telegramm versendet. Mit anderen Worten: Bei einem EAP-Gerät wird mit Hilfe von *Trigger* Bedingungen für jedes *TxData* konfiguriert, wie der *Trigger Mechanismus* arbeiten soll.

Es gibt fünf verschiedene Arten von Trigger Bedingungen, die hier beschrieben sind.

● Überlagerung von Trigger Bedingungen

i Bei den Erläuterungen zu den einzelnen Trigger Bedingungen wird darauf verwiesen, welche anderen Trigger Bedingungen deaktiviert sein müssen. Welche Bedingungen also nicht in Kombination erlaubt sind. Das Beispiel weiter unten veranschaulicht, dass sich mehrere aktive Trigger Bedingungen gegenseitig überlagern. Wie sie sich überlagern, wird nicht eindeutig definiert. Aus diesem Grund ist es empfehlenswert, alle nicht erlaubten Trigger Bedingungen zu deaktivieren.

1. Poll Request Rx PD

Die Trigger Bedingung *Poll Request Rx PD* steuert das Zurücksenden eines Antwort Telegramms im sogenannten Polled Data Exchange (siehe in Kapitel *Kommunikationsmethoden* [► 12]) Modus. Sobald ein *TxData* über einen gültigen Eintrag für die Trigger Bedingung *Poll Request Rx PD* verfügt, arbeitet das betreffende *TxData* in diesem Modus. Ein gültiger Eintrag liegt dann vor, wenn dieser mit dem Objektindex eines beim EAP-Gerät konfigurierten *RxData* übereinstimmt. Dieses *RxData* definiert

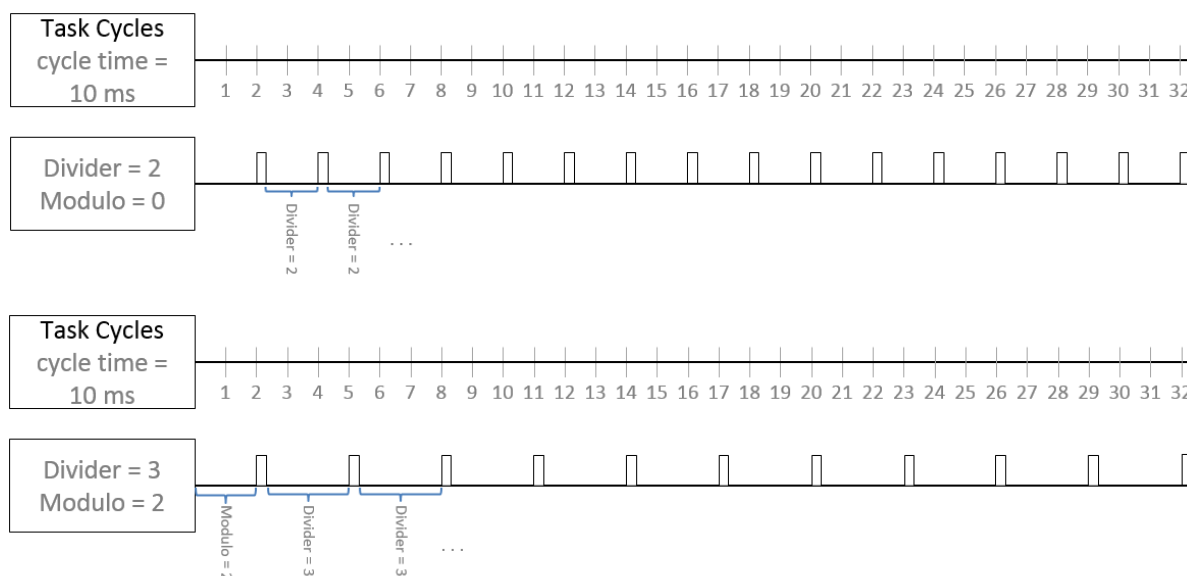
dann die erwartete Anfrage (Request), um das *TxData* als Antwort zurückzusenden. Empfängt das EAP-Gerät ein EAP-Telegramm, in dem das erwartete *Process Data* enthalten ist, wird das *TxData* im darauffolgenden Zyklus in einem neuen EAP-Telegramm an den Absender des Request Telegramms zurückgeschickt. Folglich fungiert das EAP-Gerät für dieses *TxData* als *Polled Data Exchange Server*, sobald die Trigger Bedingung *Poll Request RxData* aktiviert ist.

Alle anderen Bedingungen (2 bis 5) müssen bei aktivierter *Poll Request Rx PD* Bedingung deaktiviert sein.

2. Divider/Modulo

Per *Divider/Modulo* Bedingung wird festgelegt, in welchem Rhythmus, ein *TxFrame* bzw. ein *TxData* versendet wird (siehe folgende Abbildung). Der Rhythmus ist dabei immer ein Vielfaches der Zykluszeit der Task, die das EAP-Gerät treibt. Der *Divider* Wert legt dabei das Vielfache fest. Der *Modulo* Wert legt den Startzyklus fest, wann der *TxFrame* bzw. das *TxData* zum ersten Mal versendet wird. Wenn der *Divider* den Wert 0 hat, ist diese Bedingung deaktiviert.

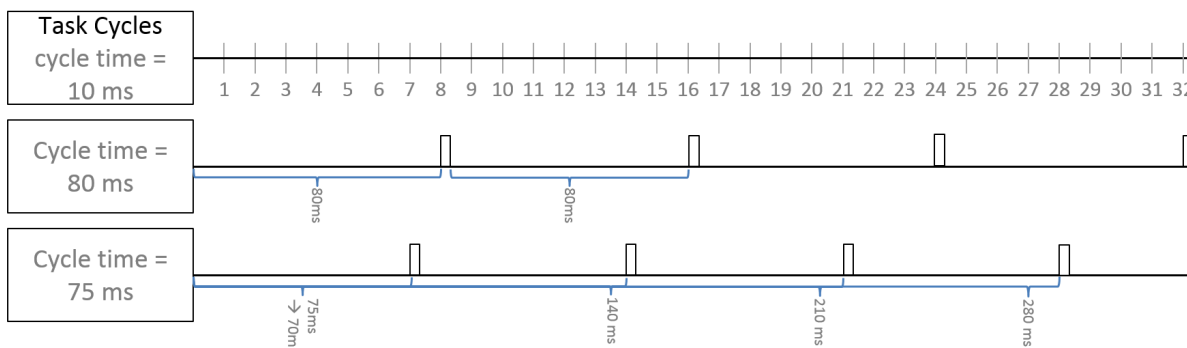
Die Bedingungen 3, 4 und 5 haben bei aktivierter *Divider/Modulo* Bedingung keine Relevanz; sie sollten aber dennoch deaktiviert werden. Die Bedingung 1 muss deaktiviert sein.



3. Cycle Time

Das *TxData* wird in einem bestimmten Zeitintervall versendet, welches durch den *Cycle Time* Wert (Einheit in μ s) festgelegt wird (siehe folgende Abbildung). Die *Cycle Time* sollte ein ganzzahliges Vielfaches der Taskzykluszeit sein. Wenn ein Wert konfiguriert wird, der kein ganzzahliges Vielfaches der Taskzykluszeit ist, wird automatisch das nächstkleinere Vielfache gesetzt, ggf. auch 0. Wenn der Wert 0 μ s beträgt, ist diese Bedingung deaktiviert.

Die Trigger Bedingungen 1, 2, 4 und 5 müssen deaktiviert sein, wenn die Trigger Bedingung *Cycle Time* aktiviert ist.



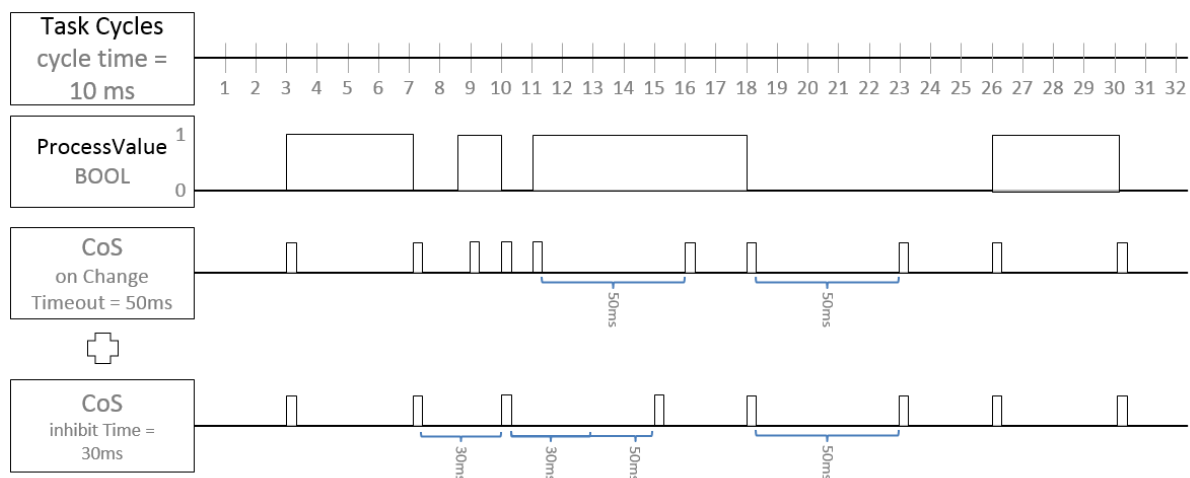
Der Zusammenhang zwischen der Cycle Time und der Taskzykluszeit

Angenommen die Taskzykluszeit beträgt 5ms (5000µs) und die Cycle Time von *Process Data A* beträgt 10000µs und die von *Process Data B* 20000µs. Nun wird die Taskzykluszeit von 5ms auf 15ms (15000µs) verlangsamt. Weder die Cycle Time von *Process Data A* noch die von *Process Data B* ist ein Vielfaches der Taskzykluszeit; die Cycle Time ist also nicht ohne Rest durch die Taskzykluszeit teilbar.

Die Folge ist, dass *Process Data A* dann nur noch alle 15ms (15000µs) und *Process Data B* nur noch alle 30ms (30000µs) gesendet wird.

4. Change of State (CoS): On Change Timeout

Das *TxData* wird erst dann gesendet, wenn sich der Wert einer seiner Variablen zum vorherigen Wert verändert hat. Dabei wird ein maximales Zeitintervall als sogenannte Timeout Time (Einheit in µs) definiert. Ändert sich der Wert einer Variablen nicht innerhalb dieses Intervalls, wird nach Ablauf des Zeitintervalls das *TxData* dennoch gesendet (siehe folgende Abbildung). Der Wert für das Zeitintervall kann nur ein ganzzahliges Vielfaches der Taskzykluszeit sein. Wenn das Zeitintervall auf 0 µs gestellt wird, ist die Trigger Bedingung *CoS On Change Timeout* deaktiviert. Die Trigger Bedingungen 1, 2 und 3 müssen deaktiviert sein, wenn die Trigger Bedingung *CoS On Change Timeout* aktiviert ist.



5. Change of State (CoS): Inhibit Time

Die *Inhibit Time* legt ein minimales Zeitintervall fest, so dass das *TxData* nicht eher gesendet wird, bis dieses Zeitintervall nach dem letzten Versenden abgelaufen ist.

Die *Inhibit Time* legt also ein minimales Zeitintervall in µs fest, nach der das *TxData* versendet wird – auch dann, wenn sich ein Wert der enthaltenen *TxVariable* geändert hat (siehe folgende Abbildung). Der Wert für dieses Zeitintervall kann nur ein ganzzahliges Vielfaches der Taskzykluszeit sein, und er muss kleiner als der Wert von *CoS On Change Timeout* sein. Wenn das Zeitintervall auf 0 µs gestellt wird, ist die Trigger Bedingung *Inhibit Time* deaktiviert.

Die Trigger Bedingungen 1, 2 und 3 müssen deaktiviert sein, wenn die Trigger Bedingung *Inhibit Time* aktiviert ist.

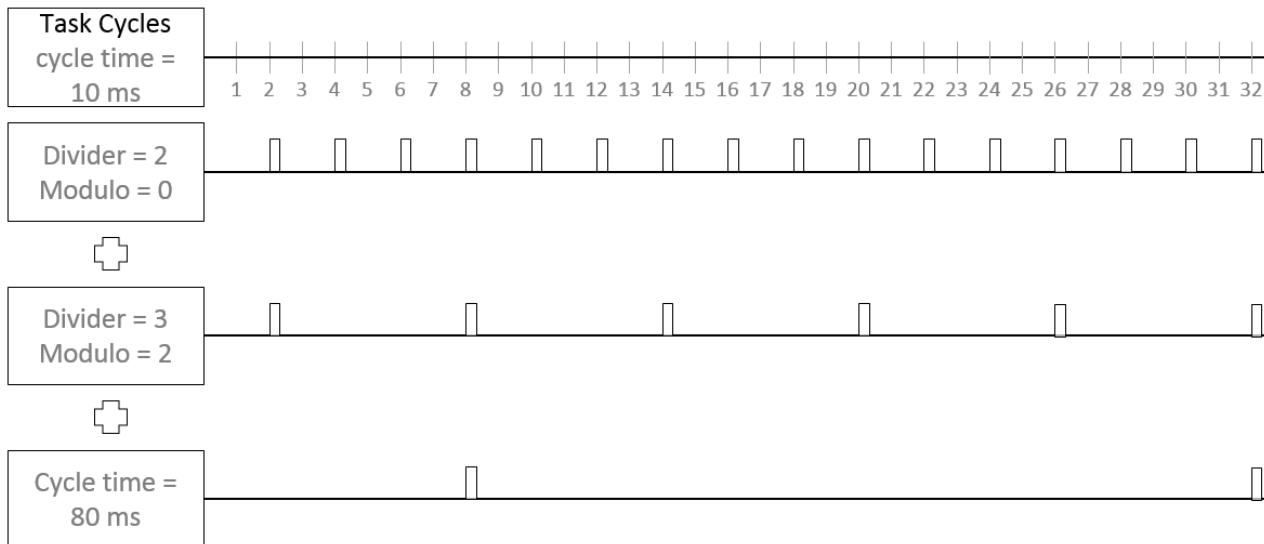
Konfigurationsmöglichkeiten der Trigger Bedingungen

Die Trigger Bedingungen 1, 3 und 5 (Poll Request RxData, Cycle Time und Inhibit Time) können über das EAP-Objektverzeichnis konfiguriert werden (siehe Kapitel CANopen Objektverzeichnis in der Dokumentation zum TwinCAT EAP-Gerät).

Besonderheiten der Trigger Bedingungen

Bei allen Trigger Bedingungen, die ein Zeitintervall definieren, kann dieses Intervall nicht kleiner sein als die Zykluszeit der Task, die das EAP-Gerät treibt.

Eine Kombination von den Bedingungen ist nicht zu empfehlen, da sie nicht eindeutig definiert sind. Als Beispiel für die Komplexität kann das folgende Beispiel genommen werden:



In der letzten Zeile wird deutlich, dass die Sendung bei 160ms und 240ms nicht stattfindet, da es von den zusätzlichen Divider/Modulo Bedingungen verhindert wird.

2.3.4 Performance des EAP

Bei der Verwendung des TwinCAT EAP-Gerätes sind die zeitlichen Randbedingungen der verwendeten Netzwerkarchitektur zu berücksichtigen:

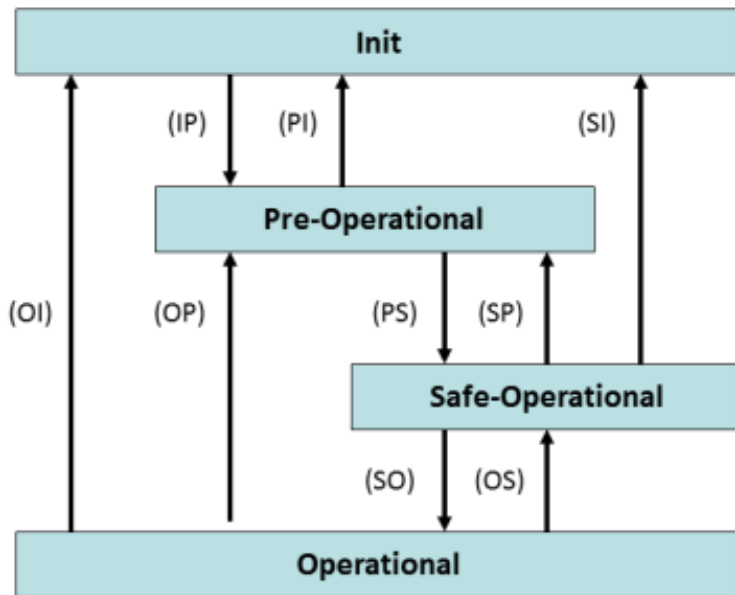
- In einer Netzwerkarchitektur, in der Telegramme ausschließlich über Switches geschickt werden (z. B. per Ethernet Protocol), können rund 10 ms Kommunikationszyklus und auch darunter erreicht werden.
- In einer Netzwerkarchitektur, in der Telegramme auch über einen Router geschickt werden (also per UDP/IP), können rund 100 ms Kommunikationszyklus erreicht werden.

Dabei kann in einem Netzwerk, in dem zusätzlich andere Kommunikation stattfindet, diese die Performanz der EAP-Kommunikation beeinträchtigen.

2.3.5 Die EAP State Machine

Über die EAP State Machine (EAP SM) wird der Zustand des EAP-Gerätes gesteuert. Je nach Zustand (Status) sind unterschiedliche Funktionen im EAP-Gerät zugänglich bzw. ausführbar. Es werden folgende Zustände unterschieden:

- Init
- Pre-Operational
- Safe-Operational und
- Operational



Der Reguläre Zustand eines jeden EAP-Gerätes nach dessen Start ist der Status **OP**. Bis zum Erreichen des Status **OP** wird das EAP-Gerät der Reihe nach einmal in jeden Zustand geschaltet. Bei jedem Zustandsübergang (Transition) werden bestimmte Aktionen vom EAP-Gerät durchgeführt. Tritt während einer der Transitionen ein Fehler auf, kann das Gerät nicht in den entsprechenden Folgezustand geschaltet werden und verbleibt somit im zuletzt erreichten Zustand. Anhand eines auslesbaren Fehlercodes kann die Ursache für den Fehler diagnostiziert werden.

Init

Der Status **Init** ist beim EAP-Gerät grundsätzlich ein temporärer Zustand. Das heißt, dass das EAP-Gerät nicht explizit in den Status **Init** versetzt werden kann. Dennoch gibt es Fälle, in denen die SM das EAP-Gerät in den Status **Init** zurücksetzt. In diesem Zustand ist weder Mailbox- noch Prozessdatenkommunikation mit dem EAP-Gerät möglich.

Pre-Operational (Pre-Op)

Beim Übergang von **Init** nach **Pre-Op** prüft das EAP-Gerät, ob die Mailbox korrekt initialisiert wurde. Im Status **Pre-Op** ist Mailbox- aber keine Prozessdatenkommunikation möglich.

Safe-Operational (Safe-Op)

Beim Übergang von **Pre-Op** nach **Safe-Op** prüft das EAP-Gerät die internen Objektreferenzen und aktualisiert

- die zykluszeitbasierten Konfigurationsparameter,
- die Referenzzeiger auf die Input- und Output-Variablen des Prozessabbildes sowie
- das Mapping jeder Publisher/Subscriber Variablen zu ihrer Prozessvariablen aus der SPS.

Im Status **Safe-Op** wird Mailboxkommunikation und das Senden von Publisher Variablen betrieben. Es werden jedoch noch keine EAP-Telegramme empfangen.

Operational (Op)

Beim Übergang von **Safe-Op** nach **Op** prüft das EAP-Gerät abschließend noch einmal, ob ein Fehler während des Startens des EAP-Gerätes aufgetreten ist.

Im Status **Op** empfängt das EAP-Gerät ankommende EAP-Telegramme und kopiert ggf. die empfangenen Prozessdaten in die Input-Variablen. Es wird sowohl Mailboxkommunikation betrieben als auch Publisher Variablen gesendet und Subscriber Variablen empfangen.

Wenn in einem der Zustandsübergänge ein Fehler auftritt, bleibt das EAP-Gerät entweder im zuletzt erreichten Zustand oder es wird in den Zustand **Safe-Op** zurückgesetzt. Gleichzeitig werden das Fehler-Bit und ein entsprechender Fehler-Code gesetzt (vgl. Kapitel Das CANopen Objektverzeichnis in der Dokumentation zum TwinCAT EAP-Gerät). Typische Fehler treten beispielsweise wegen Inkonsistenzen im CANopen Objektverzeichnis auf, so dass die Konfiguration ungültig ist.

3 Installation

Der TwinCAT EAP Configurator wird als Workload aus dem TwinCAT Package Manager oder durch ein Setup installiert.

3.1 Systemvoraussetzungen

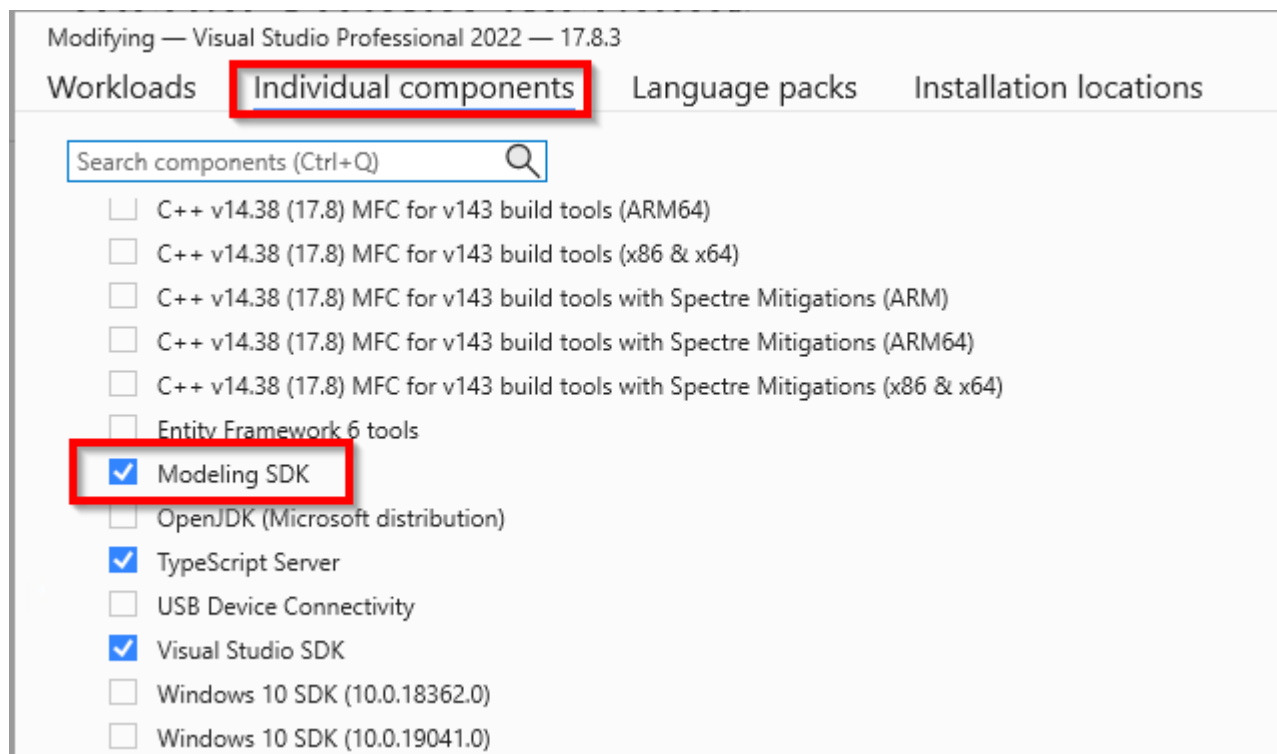
Technische Daten	Voraussetzung
Betriebssystem	Windows 10 / 11 Als Zielgeräte können zusätzlich auch Windows CE 6 oder TwinCAT/BSD verwendet werden
Minimale TwinCAT-Version	TwinCAT 3.1 Build 4022.0 und höher
Erforderliches TwinCAT-Setup-Level	TwinCAT 3 XAE
Erforderliche TwinCAT-Lizenz	TE1610 TwinCAT 3 EAP Configurator

3.2 Installation

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

Voraussetzung für die Nutzung des TE1610 | TwinCAT 3 EAP Configurator ist, dass in der jeweiligen Visual Studio Version die Erweiterung „Modeling SDK“ installiert ist.

Dieses kann über den Visual Studio Installer erfolgen:



TwinCAT Package Manager: Installation (TwinCAT 3.1 Build 4026)

Eine ausführliche Anleitung zur Installation von Produkten finden Sie im Kapitel [Workloads installieren](#) in der [Installationsanleitung TwinCAT 3.1 Build 4026](#).

Installieren Sie den folgenden Workload, um das Produkt nutzen zu können:

- TE1610 | TwinCAT 3 EAP Configurator

TwinCAT Setup: Installation (TwinCAT 3.1 Build 4024 und früher)

Die Function kann für TwinCAT 3.1 Build 4024 und früher als TwinCAT Setup von der Beckhoff Homepage bezogen werden.

3.3 Lizenzierung

Das Produkt ist auch ohne Lizenz ohne Einschränkungen nutzbar.

4 Erste Schritte

Bevor der TwinCAT EAP Configurator sinnvoll zum Einsatz kommen kann, müssen in einem Netzwerk wenigstens zwei TwinCAT EAP-Geräte mit Hilfe von TwinCAT über die I/O Konfiguration angelegt und gegebenenfalls aktiviert worden sein. Die Konfiguration eines TwinCAT EAP-Gerätes wird in der Dokumentation zum TwinCAT EAP-Gerät beschrieben. Dort ist auch ein Kapitel dem Thema gewidmet, wie mit TwinCAT eine EAP Device Configuration (EDC) Datei für ein EAP-Gerät erzeugt wird.



EDC-Datei

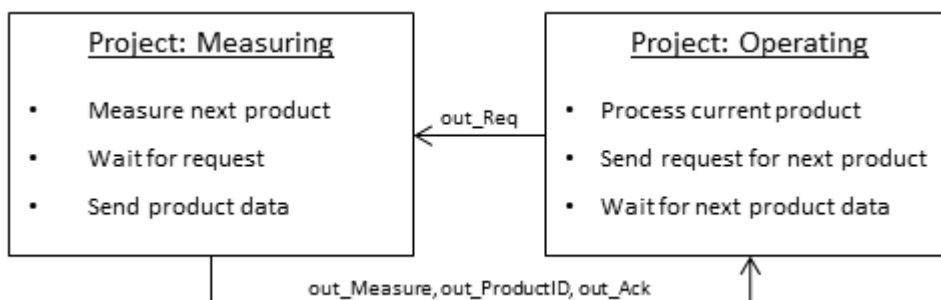
Wegen des höheren Komforts sollte generell eine EDC-Datei für jedes TwinCAT EAP-Gerät erzeugt werden, um sie anschließend im TwinCAT EAP Configurator zu importieren.

4.1 Die Beispielprojekte

Für ein tieferes Verständnis wird der Umgang mit dem TwinCAT EAP Configurator anhand von einem Beispiel gezeigt. Statt die beiden Beispielprojekte eigenhändig zu erzeugen, besteht die Möglichkeit diese Projekte über die Beckhoff Homepage [herunterzuladen](#). Eine Beschreibung, wie die Beispielprojekte manuell von Grund auf erstellt werden, finden Sie im Anhang unter [Anlegen der Beispielprojekte](#) [► 64].

Die beiden TwinCAT-Projekte *Measuring* und *Operating* sollen je eine Maschine steuern, die gemeinsam eine kleine Produktionslinie bilden. Das Projekt *Measuring* beinhaltet die Konfiguration der ersten Maschine, im Folgenden *Maschine M* genannt. Sie misst eine bestimmte Eigenschaft bei einem Produkt (z. B. eine Länge). Das Projekt *Operating* beinhaltet die Konfiguration der zweiten Maschine, im Folgenden *Maschine O* genannt. Sie bearbeitet ein Produkt (z. B. wird eine Nut gefräst, die der gemessenen Länge entspricht).

Der gesamte Verarbeitungsprozess in der Produktionslinie soll folgendermaßen ablaufen (siehe folgende Abbildung): *Maschine M* misst ein Produkt aus. Wenn der Messvorgang beendet ist, ist *Maschine M* bereit, den gemessenen Wert für eine Bearbeitung des Produktes herauszugeben. Sobald eine Anfrage von *Maschine O* über die Eingangsvariable *in_Req* in *Maschine M* detektiert wird, wird der gemessene Wert in die dafür vorgesehene Ausgangsvariable *out_Measure* geschrieben. (In dem Beispielprojekt schwanken die Messwerte zufällig zwischen 4500 und 5500.) Zusätzlich wird die aktuelle ProduktID in die Variable *out_ProductID* geschrieben und die Anfrage der *Maschine O* mit Hilfe der Variablen *out_Ack* bestätigt. Anschließend wird die Eingangsvariable *in_Req* wieder zurückgesetzt, weil die Anfrage bearbeitet wurde, und das nächste Produkt wird gemessen.



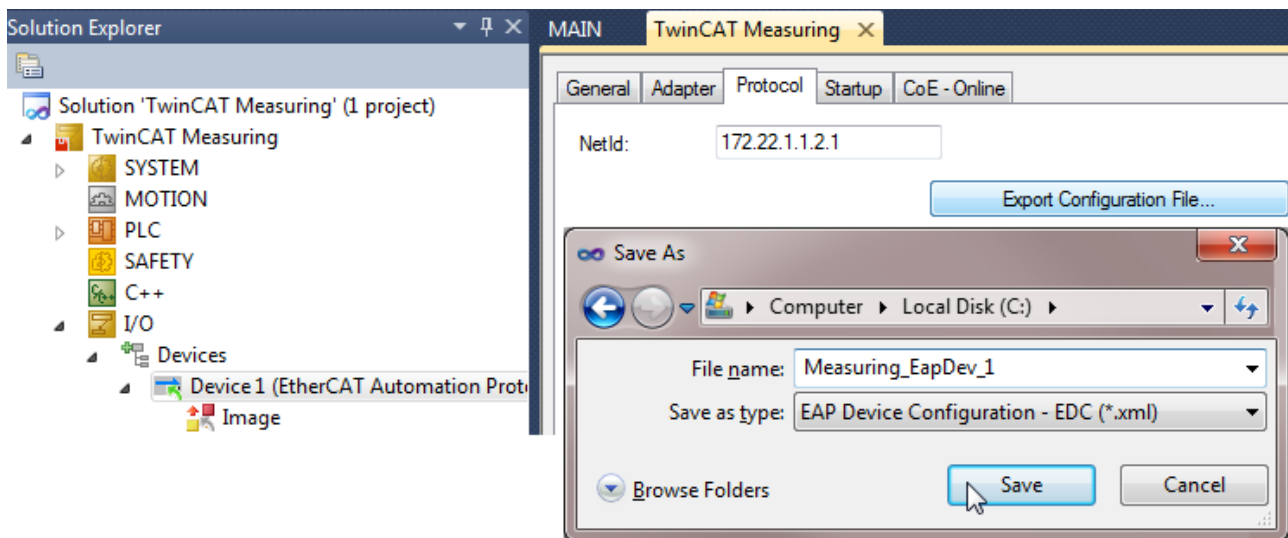
Maschine O teilt mittels seiner Ausgangsvariablen *out_Req* mit, dass sie bereit ist, das nächste Produkt zu bearbeiten. Anschließend wartet sie auf eine Bestätigung der *Maschine M*, dass das nächste Produkt mit den notwendigen Bearbeitungsparametern zur Bearbeitung bereitsteht. Die Bestätigung erwartet *Maschine O* in der Variablen *in_Ack*, die Bearbeitungsparameter werden in den Variablen *in_Measure* und *in_ProductID* erwartet. Empfängt die *Maschine O* eine Bestätigung, wird die Ausgangsvariable *out_Req* zurückgesetzt, da nun die Bearbeitung für das Produkt vorgenommen wird. Erst wenn die Bearbeitung abgeschlossen ist, wird das nächste Produkt angefragt. Die Dauer der Bearbeitung richtet sich nach dem gemessenen Wert. (In diesem Beispiel: Bearbeitungsdauer in Millisekunden = Wert / 2, also schwankt die Bearbeitungsdauer zwischen 2250ms und 2750ms).

4.2 Exportieren der EDC-Dateien

Das Erzeugen der EDC-Datei erfolgt aus TwinCAT 3 heraus. Entsprechend müssen die beiden Beispielprojekte zuerst mit dem Microsoft Visual Studio geöffnet werden.

1. Öffnen Sie per Doppelklick das Gerät *Device 1 (EtherCAT Automation Protocol)* aus der I/O-Konfiguration.
2. Wählen Sie den Karteireiter *Protocol* aus und betätigen Sie die Schaltfläche [Export Configuration File...].

⇒ Es öffnet sich der *Save As* Dialog.



1. Definieren den Dateinamen und den Speicherort für die EDC-Datei.

Die EDC Datei von *Maschine M* erhält am besten den Dateinamen *Measuring_EapDev_1.xml*. Die EDC-Datei von *Maschine O* entsprechend den Namen *Operating_EapDev_1.xml*.

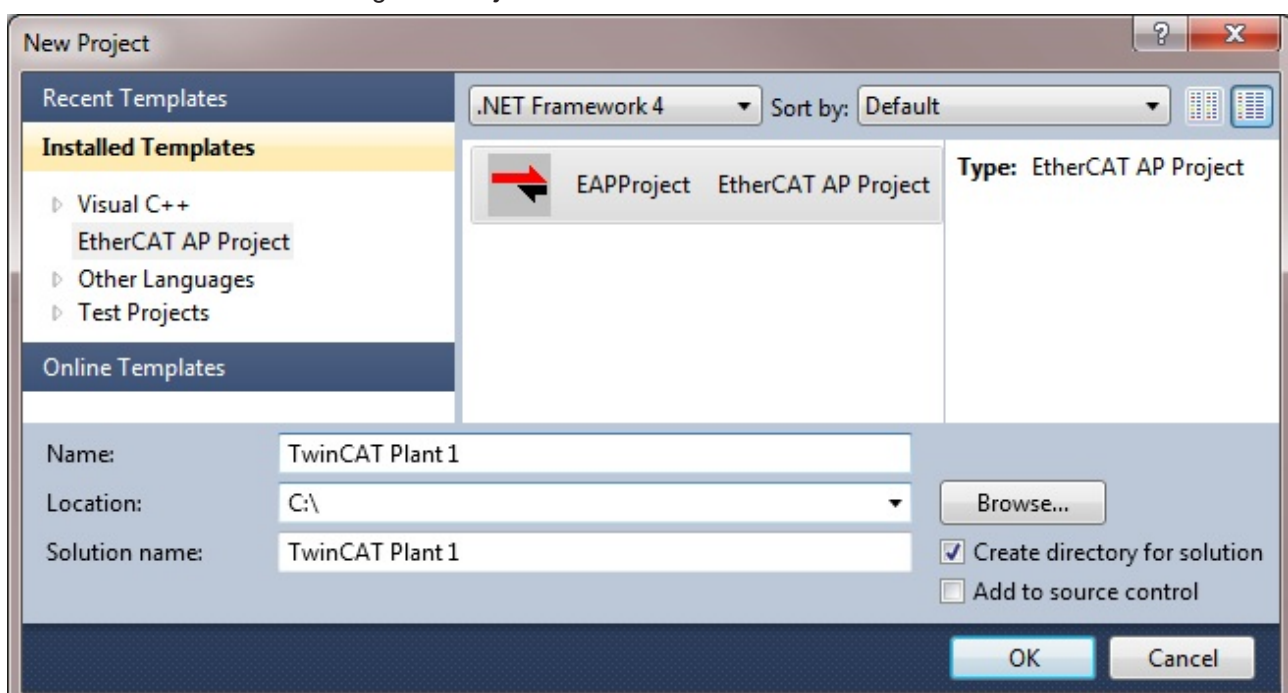
In beiden Beispielprojekten ist je ein kleines Beispiel SPS-Programm mit ein paar Ein- und Ausgangsvariablen angelegt worden. Die Symbolinformationen dieser Variablen werden automatisch beim Export in den EDC-Dateien mitgespeichert. Später kann das TwinCAT EAP-Gerät unter Verwendung dieser Symbolinformationen auf die Inhalte dieser Variablen zugreifen.

4.3 Anlegen eines TwinCAT EAP Configurator Projektes

Erstellen Sie ein TwinCAT EAP Configurator Projekt wie folgt:

1. Wählen Sie das Kommando [New] im Menü [File].

⇒ Es öffnet sich der Dialog *New Project*.



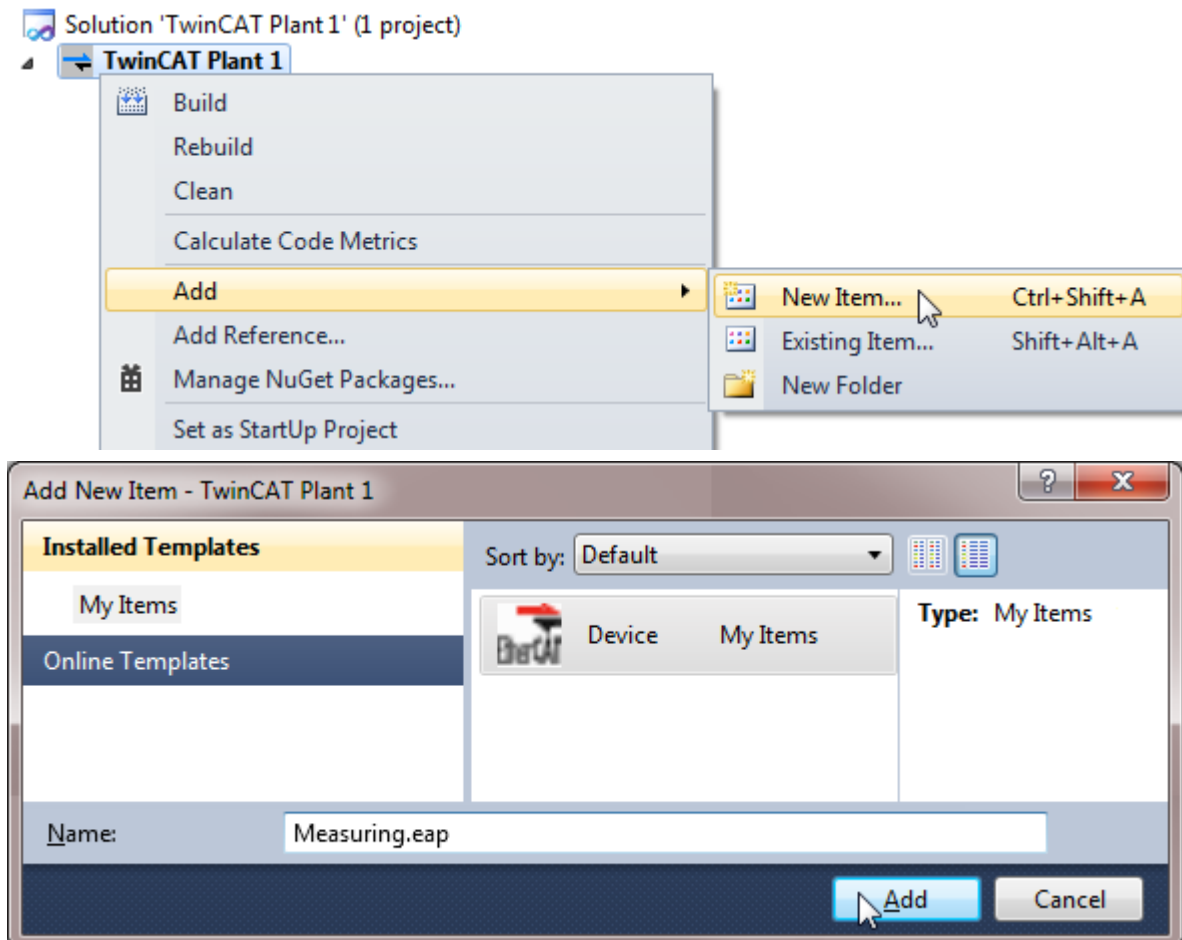
1. Wählen Sie aus der Liste *Installed Templates* den Eintrag *EtherCAT AP Projekt* aus.
 2. Tragen Sie einen Namen für das Projekt ein (z.B. *TwinCAT Plant 1*) und wählen Sie ggf. ein Verzeichnis, unter dem das Projekt gespeichert werden soll.
 3. Abschließend bestätigen Sie bitte Ihre Eingaben mit der Schaltfläche [OK].
- ⇒ Es wird der Projektknoten erzeugt und im Visual Studio angezeigt.

4.3.1 Hinzufügen eines EAP-Geräteobjekts

Für die beiden im Netzwerk befindlichen EAP-Gerät wird nun ein Geräteknoten an den Projektknoten angefügt.

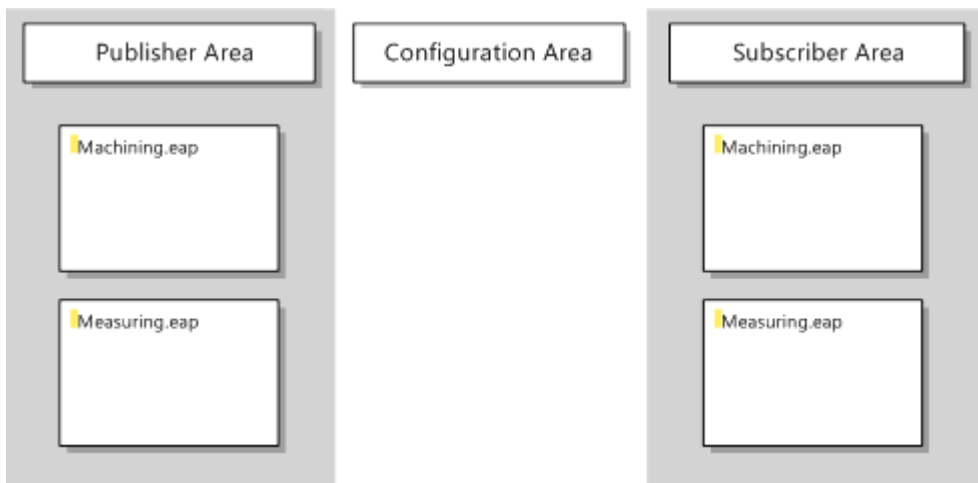
1. Führen Sie einen Rechtsklick auf den Projektknoten *TwinCAT Plant 1* aus und wählen Sie das Kommando [Add] → [New Item] aus dem Kontextmenü.

⇒ Es öffnet sich der Dialog *Add New Item*.



1. Vergeben Sie je einen aussagekräftigen Namen für Ihre EAP-Geräte (z.B. für die *Maschine M* den Namen *Measuring* bzw. für die *Maschine O* den Namen *Operating*) und bestätigen Sie Ihre Eingabe mit der Schaltfläche [Add].
- ⇒ Die erzeugten Geräteknoten repräsentieren anschließend die entsprechenden EAP-Geräte der beiden Maschinen.

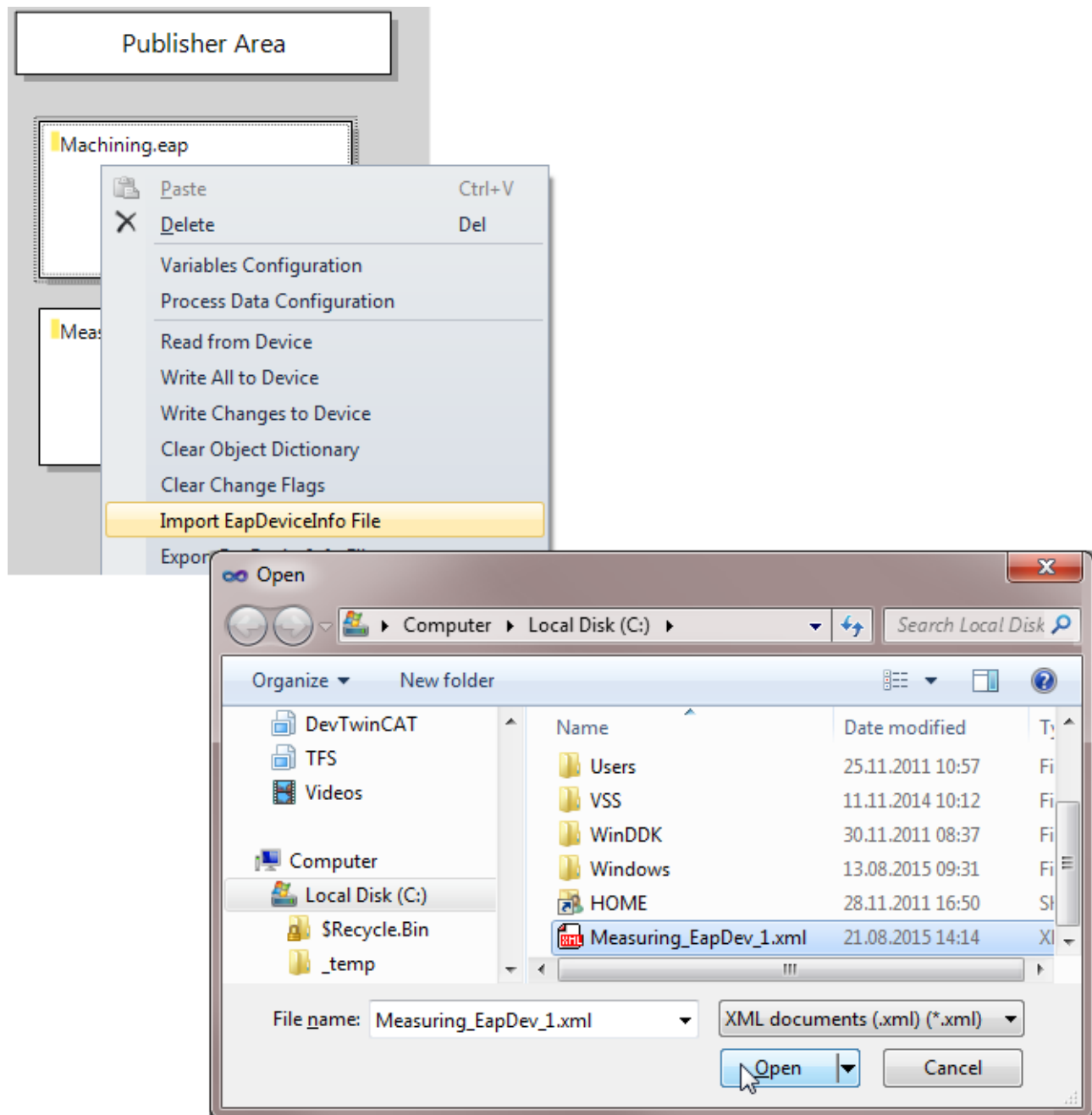
Nachdem alle gewünschten Geräteknoten erzeugt worden sind, wird ein beliebiger Geräteknoten per Doppelklick geöffnet. Daraufhin werden in der sich öffnenden grafischen Bedienoberfläche alle angelegten Geräteknoten geladen und als EAP-Geräteobjekte dargestellt (vgl. Abbildung). Jedes EAP-Geräteobjekt wird als Kasten sowohl in der Spalte *Publisher Area* als auch in der Spalte *Subscriber Area* dargestellt. Mit anderen Worten: Ein EAP-Geräteobjekt wird grafisch von zwei Kästen repräsentiert (vgl. Kapitel [Die Architektur](#) [► 37]).



4.3.2 Importieren einer EAP Device Configuration (EDC) Datei

Als nächstes wird für jeden Geräteknoten die zugehörige EDC-Datei importiert.

1. Klicken Sie mit der rechten Maustaste auf das gewünschte EAP-Geräteobjekt und wählen Sie im Kontextmenü das Kommando [Import EapDeviceInfo File].
2. Nun wählen Sie die zugehörige EDC-Datei aus (*Measuring_EapDev_1.xml* oder *Operating_EapDev_1.xml*) und bestätigen Sie Ihre Auswahl mittels der Schaltfläche [Open] (vgl. Abbildung).



4.3.3 Aktuelle Konfiguration eines EAP-Gerätes lesen

Gegebenenfalls können Sie die aktuelle Konfiguration aus dem aktiven EAP-Gerät einlesen (vgl. Kapitel [Eine aktive Gerätekonfiguration lesen](#) [► 43]). Voraussetzung dafür ist, dass das EAP-Gerät bereits mit Hilfe von TwinCAT 3 aktiviert wurde. Andernfalls existiert noch gar keine Instanz des EAP-Gerätes, und demzufolge wäre das EAP-Gerät auch nicht erreichbar. Außerdem muss eine ADS/AMS-Route eingetragen sein von dem PC, auf dem der TwinCAT EAP Configurator läuft, zu der Maschine, auf dem das TwinCAT-Projekt *Measuring* bzw. *Operating* läuft.

Beim Lesen der aktuellen Konfiguration wird die zuvor importierte Gerätekonfiguration verworfen und durch die gelesene Konfiguration ersetzt. Die importierten Symbolinformationen bleiben davon jedoch unberührt.

i Verzicht auf das Importieren einer EDC-Datei

Alternativ kann auch auf das Importieren einer EDC-Datei ganz verzichtet werden. Dann fehlen jedoch zum einen sämtliche Symbolinformationen und zum anderen muss unter den Eigenschaften im Properties Fenster des EAP-Geräteobjekts die passende *Local AoE Net ID* des TwinCAT EAP-Gerätes eingetragen werden, ehe die Konfiguration des TwinCAT EAP-Gerätes gelesen werden kann.

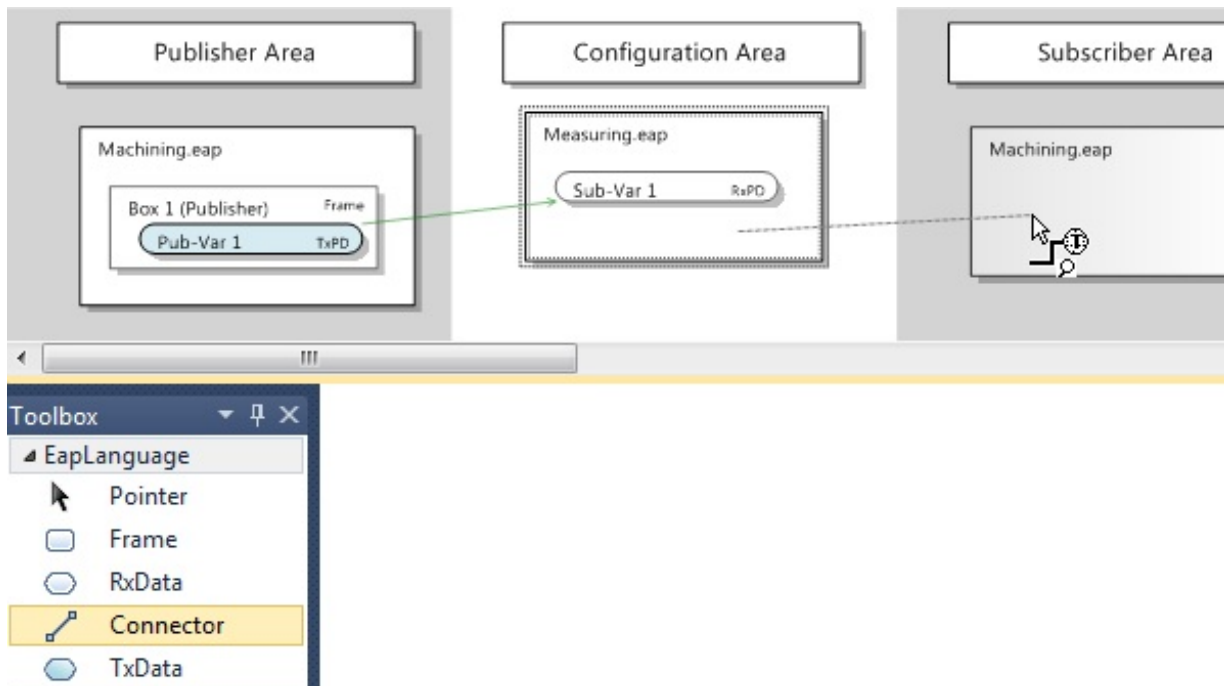
4.3.4 Eine Kommunikationsverbindung generieren

Nachdem nun eine Abbildung des konfigurierten IST-Zustands der EAP-Geräte hergestellt worden ist, besteht jetzt die Möglichkeit diese Konfiguration zu ändern. Beispielhaft werden zwei Änderungen vorgestellt.

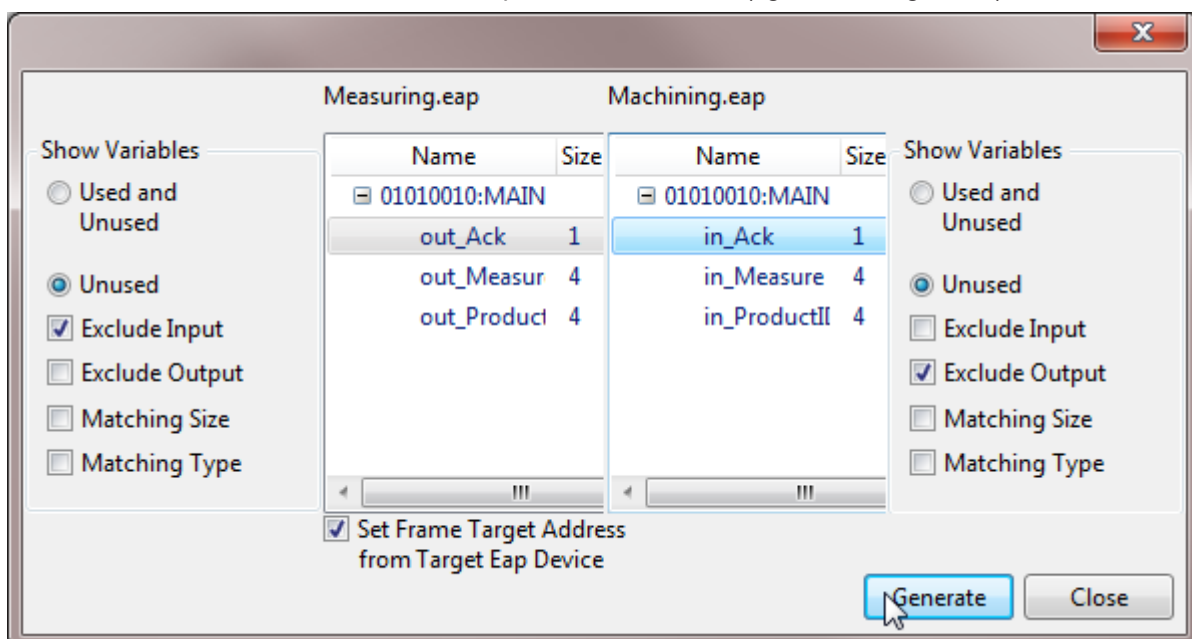
(A) Eine weitere Kommunikationsverbindung konfigurieren

Die komfortabelste Methode, eine Kommunikationsverbindung zwischen zwei EAP-Geräten zu konfigurieren, ist die Verwendung des *Connector Tools* aus der *Toolbox* (vgl. Kapitel [Das Connector-Tool](#) [► 40]).

1. Wählen Sie hierzu das *Connector Tool* per Mausklick aus.
2. Klicken Sie das EAP-Geräteobjekt an, welches die Daten senden soll.



1. Anschließend klicken Sie das EAP-Geräteobjekt an, welches die Daten empfangen soll.
(vgl. Abbildung oben)
⇒ Infolgedessen öffnet sich ein Dialog, in dem die Symbolnamen der SPS-Variablen aufgelistet werden, die aus der EDC-Datei importiert worden sind (vgl. Abbildung unten).



Die Liste auf der linken Seite zeigt die Symbolnamen aller Ausgangsvariablen des Senders und die Liste auf der rechten Seite zeigt die Symbolnamen aller Eingangsvariablen des Empfängers.

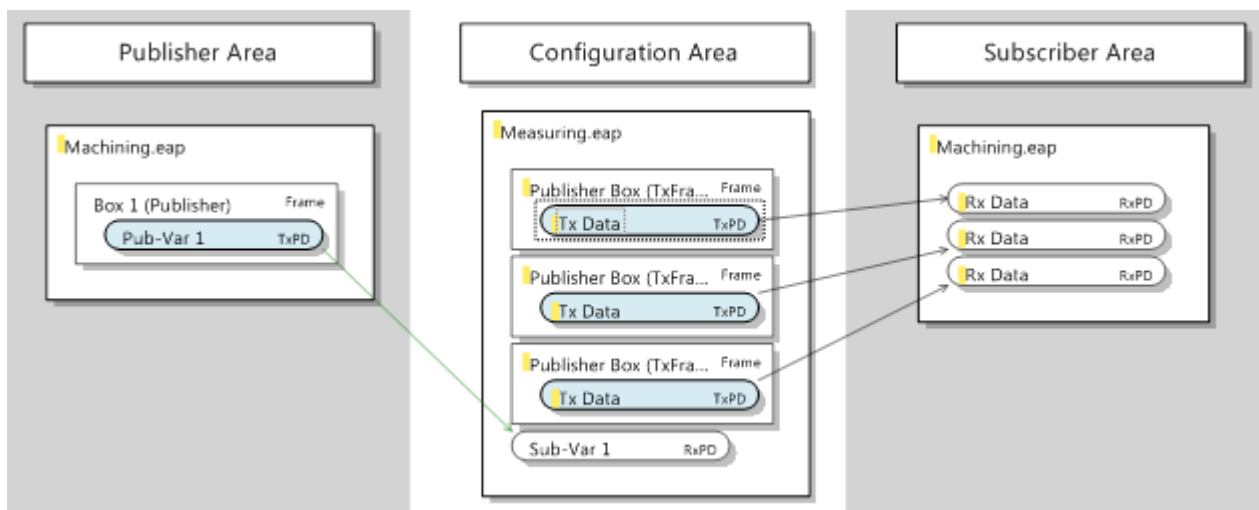
1. Wählen Sie aus der linken Liste die Variable aus, deren Werte gesendet werden sollen.
2. Wählen Sie aus der rechten Liste die Variable aus, die die gesendeten Werte empfangen soll.
3. Verwenden Sie die Schaltfläche [Generate], um die Kommunikationsverbindung zu generieren.

1 Die Größe der Sende- und Empfangsvariable muss übereinstimmen

Es muss darauf geachtet werden, dass die Größe der ausgewählten Sender- und Empfangsvariable identisch ist. Um bei der Auswahl Fehler zu vermeiden, besteht die Möglichkeit, entweder auf der Empfänger- oder auf der Senderseite den Anzeigefilter *Matching Size* zu aktivieren. So werden nur die Symbolnamen der Variablen angezeigt, die identisch zu der Größe der ausgewählten Variablen sind.

⇒ Anschließend können weitere Verbindungen zwischen den beiden EAP-Geräteobjekten konfiguriert werden oder über die Schaltfläche [Close] der Dialog wieder verlassen werden.

In diesem Beispiel sollten alle Ausgangsvariablen sukzessive mit den entsprechenden Eingangsvariablen verbunden werden, so dass am Ende beim Sender drei Publisher Boxen mit jeweils einem TxData und beim Empfänger drei RxData konfiguriert sind (vgl. Abbildung).



Nach der Fertigstellung der gewünschten Konfiguration, muss diese noch auf das EAP-Gerät übertragen werden. Diese Maßnahme wird in Kapitel [Die neue/geänderte Konfiguration auf die EAP-Geräte übertragen](#) [► 34] beschrieben.

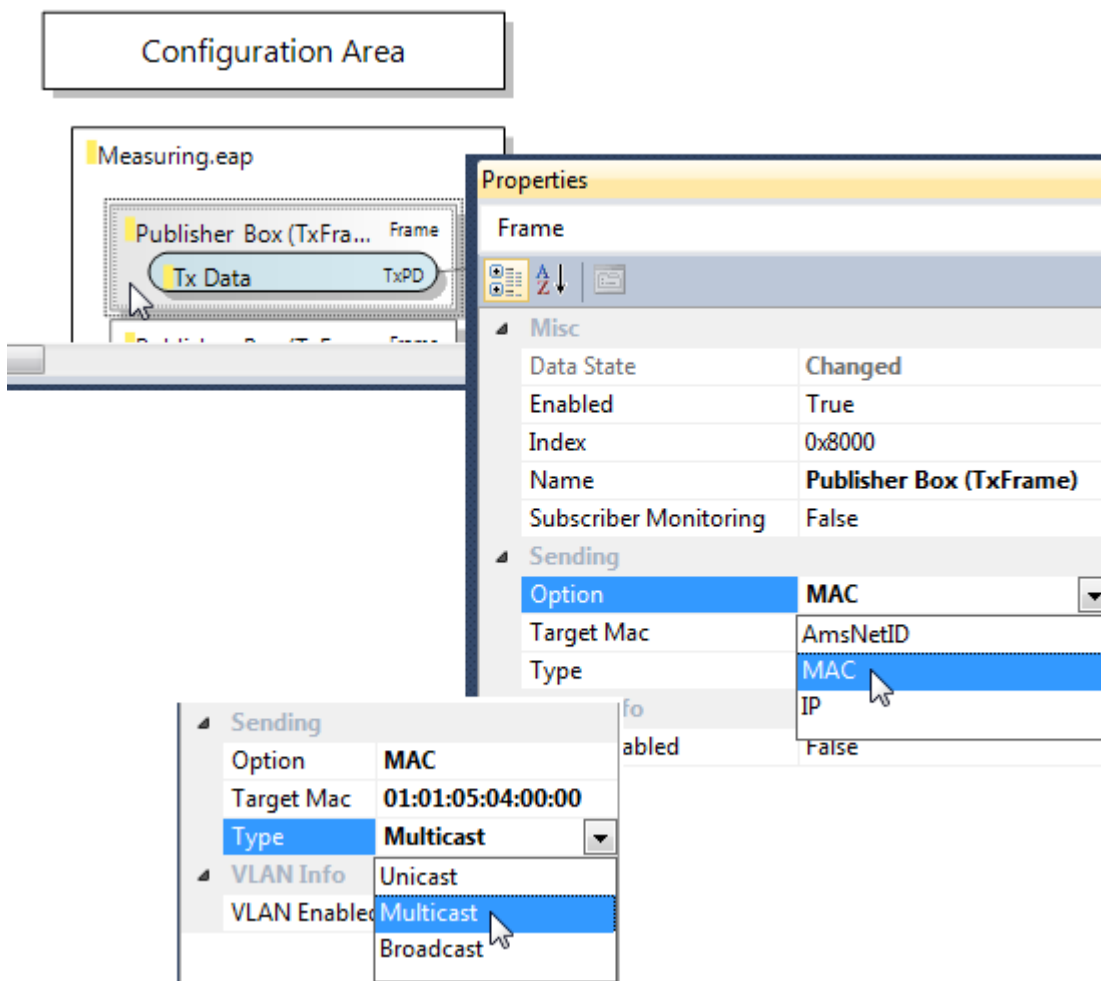
(B) Die Eigenschaften einer bestehenden Kommunikationsverbindung verändern

Zu jeder Kommunikationsverbindung gibt es die Möglichkeit, deren Sende- bzw. Empfangseigenschaften spezieller zu konfigurieren. Alle Möglichkeiten, die der TwinCAT EAP Configurator dafür zur Verfügung stellt, werden im Einzelnen im Kapitel [Das Properties Fenster](#) [► 52] vorgestellt. Beispielhaft werden an dieser Stelle vier Eigenschaften geändert, die des Öffern in einer Konfiguration angepasst werden.

Multicast MAC Adresse als Zieladresse konfigurieren

Allgemein ist eine generierte Kommunikationsverbindung so konfiguriert, dass die Variable an einen einzelnen Empfänger adressiert wird (Unicast). In einigen Fällen ist es jedoch erwünscht, dass die Variable mehrere Empfänger erreicht. Dann muss die Variable in einem broadcastadressierten oder multicastadressierten Frame versendet werden. Vorzugsweise sollte ein Multicast konfiguriert werden, um andere Netzwerkteilnehmer nicht unnötig zu belasten.

Dazu wird die Eigenschaft *Type* im Properties Fenster der betreffenden *Publisher Box (TxFrame)* auf die Einstellung *Multicast* gesetzt. Als Standard-Multicast-Adresse wird dann automatisch die TwinCAT Multicast-Adresse für EAP-Geräte unter der Eigenschaft *Target MAC* eingetragen (vgl. Abbildung).



• Eine andere Multicast-Adresse

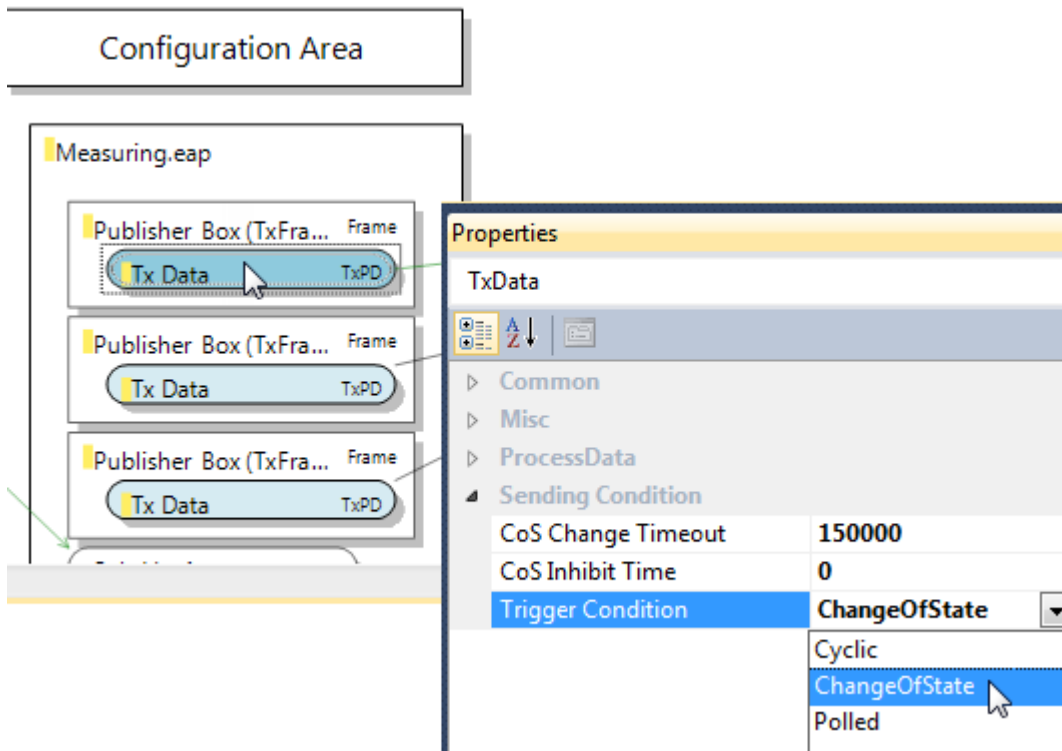
Wenn eine andere Multicast-Adresse als die Standard-Multicast-Adresse für TwinCAT EAP-Geräte erwünscht ist, sollte der Abschnitt *Multicast MAC Address* im Kapitel [Das RxData](#) [► 58] beachtet werden.

Nachdem die Konfiguration modifiziert worden ist, muss diese noch auf das EAP-Gerät übertragen werden. Diese Maßnahme wird in Kapitel [Die neue/geänderte Konfiguration auf die EAP-Geräte übertragen](#) [► 34] beschrieben.

Den Sende-Trigger auf die Bedingung *Change of State* konfigurieren

Im Normalfall ist bei der *Trigger Condition* eines *TxData* die Eigenschaft *Cyclic* konfiguriert. Wenn eine Variable jedoch nur dann versendet werden soll, wenn sich ihr Wert verändert hat, dann muss die *Trigger Condition* auf die Eigenschaft *Change of State* (CoS) konfiguriert werden. Entsprechend müssen dann die Eigenschaften *Change Timeout Time* und *Inhibit Time* definiert werden (vgl. Kapitel [Das TxData](#) [► 56]). Bei der Definition dieser Zeiten sollte der Hinweis aus dem Kapitel [Das Objektverzeichnis](#) [► 61] beachtet werden.

In diesem Beispiel dauert der Messvorgang 200ms. Entsprechend werden für alle drei *TxData* der *Maschine M* die *Change Timeout Time* auf 150000µs und die *Inhibit Time* auf 0µs gesetzt (vgl. Abbildung). Diese Einstellung hat zur Folge, dass spätestens nach 150000 µs = 150ms die Variablen erneut gesendet werden, selbst dann, wenn sich ihre Werte nicht geändert haben. Ändert sich der Wert einer Variablen mit jedem Taskzyklus (z.B. alle 10ms), wird die Variable entsprechend einmal in jedem Zyklus versendet (wegen der *Inhibit Time* = 0µs).



Analog kann für das *TxData* der *Maschine O* der Timeout-Wert von $1000000\mu\text{s} = 1000\text{ms}$ konfiguriert werden. In diesem Beispiel richtet sich die Verarbeitungsdauer der *Maschine O* nach den gemessenen Werten von *Maschine M*. Die gemessenen Werte variieren zwischen 4500 und 5500. Die Bearbeitung in *Maschine O* dauert dann den halbierten gemessenen Wert in Millisekunden (vgl. [Die Beispielprojekte](#) [► 23]). Entsprechend genügt es, wenn maximal jede Sekunde von der *Maschine O* die Request-Variable gesendet wird.

Nachdem die Konfiguration modifiziert worden ist, muss diese noch auf das EAP-Gerät übertragen werden. Diese Maßnahme wird in Kapitel [Die neue/geänderte Konfiguration auf die EAP-Geräte übertragen](#) [► 34] beschrieben.

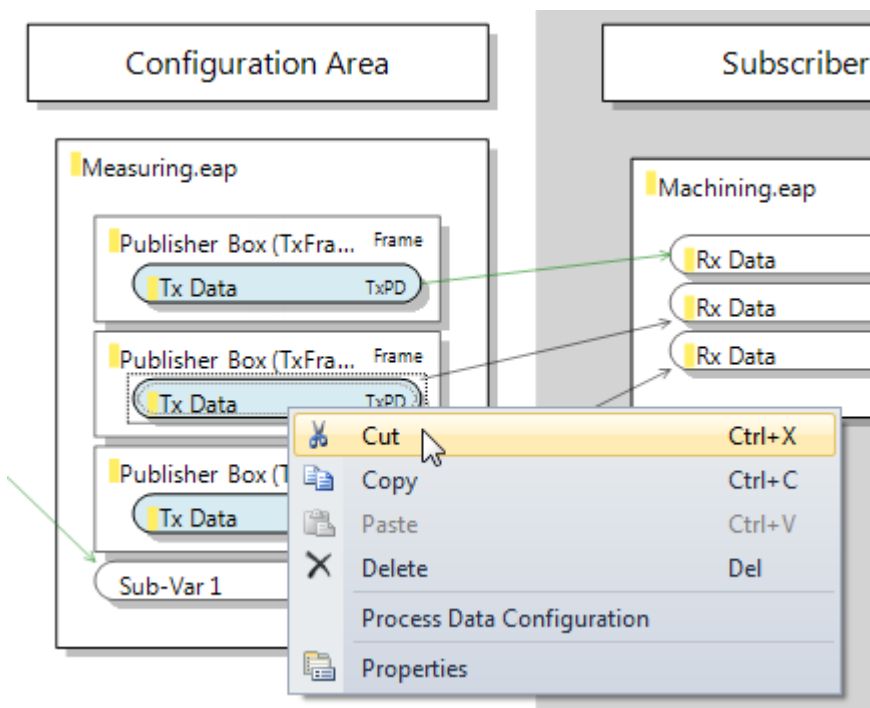
Variablen aus verschiedenen Frames in einen einzelnen Frame verschieben

In diesem Beispiel ist es wichtig, dass die Prozessvariablen: *out_Measure*, *out_ProductID* und *out_Ack* für das zu bearbeitende Produkt in sich konsistent sind. Aus diesem Grund ist es notwendig, dass diese Daten stets zusammen in einem Datenpaket gesendet werden. Bei der Konfiguration mit Hilfe des *Connector Tools* ist für jede Variable ein separater Frame konfiguriert worden. Nun sollen alle Variablen in einen einzigen Frame verschoben werden.

Diese Modifikation lässt sich im TwinCAT EAP Configurator sehr leicht bewerkstelligen:

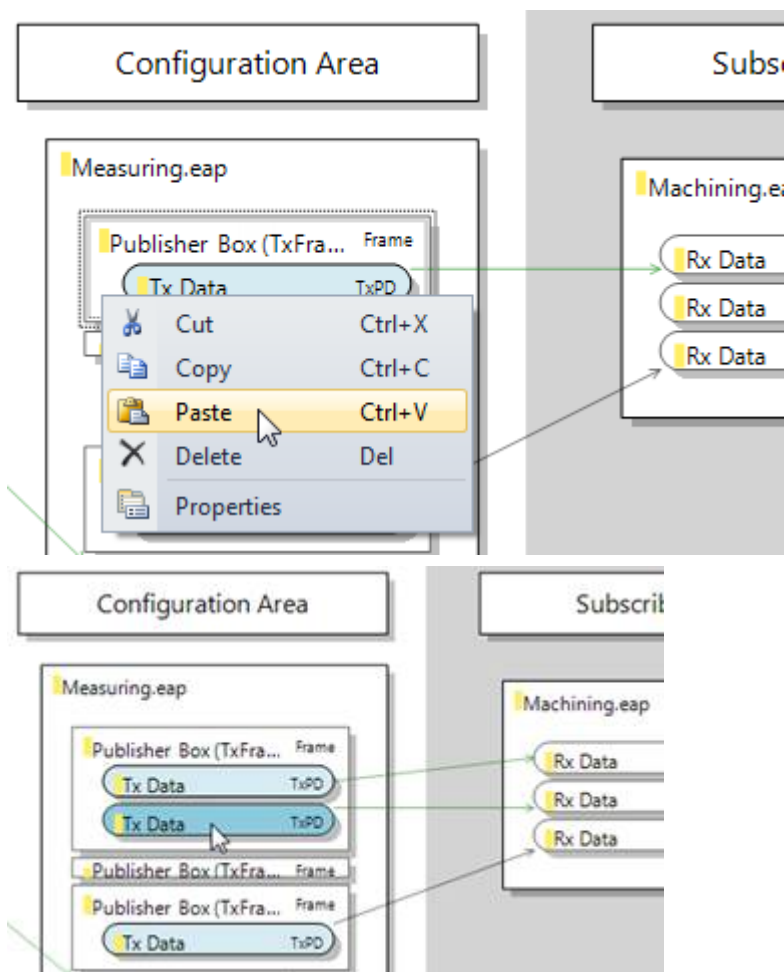
Die notwendigen Kommandos befinden sich in den Kontextmenüs der jeweiligen grafischen Objekte. Alternativ können auch die allgemein bekannten Tastaturfunktionen ([Ctrl] + [X] für Ausschneiden, [Ctrl] + [V] für Einfügen und [Delete] für Löschen) verwendet werden.

1. Selektieren Sie das *TxData* aus der zweiten *Publisher Box* und schneiden Sie dieses per Kommando [Cut] aus (vgl. Abbildung).

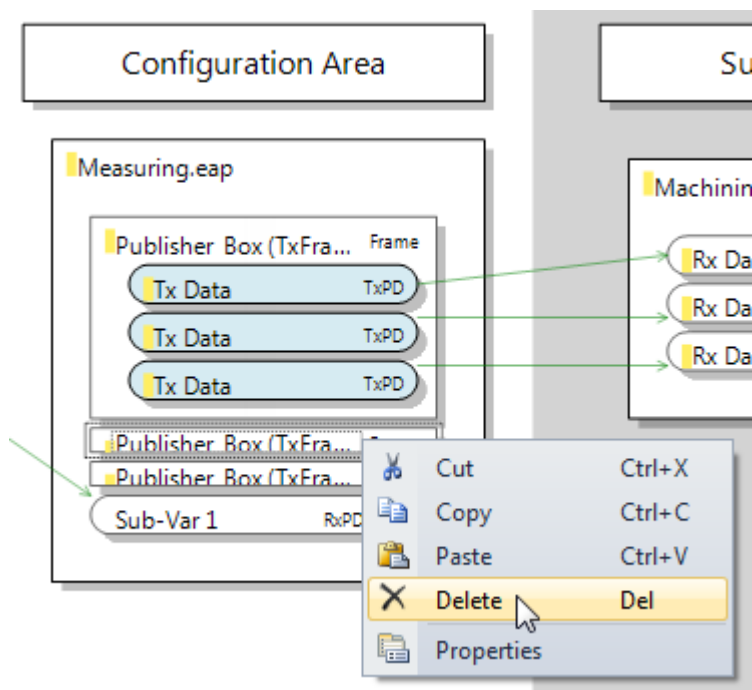


1. Selektieren Sie anschließend die erste *Publisher Box* und wählen Sie das Kommando [Paste], um das *TxData* dort einzufügen (vgl. Abbildung).

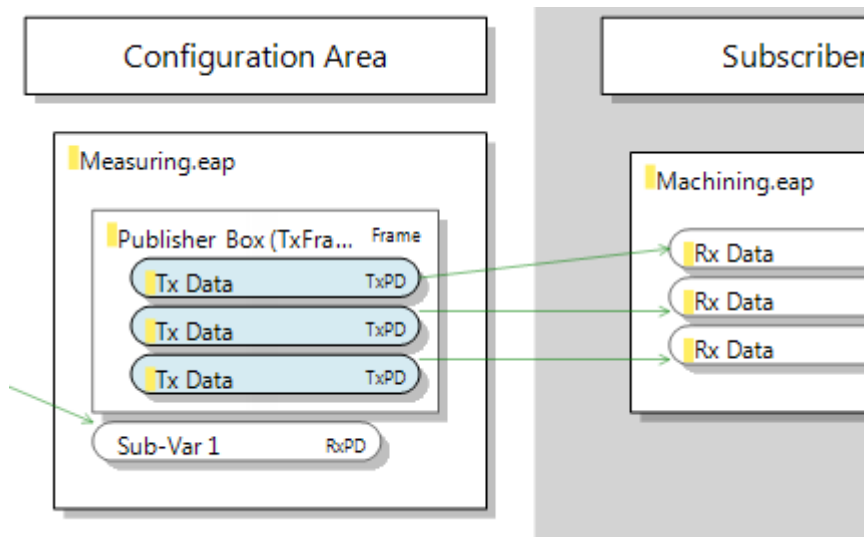
⇒ Führen Sie die beiden Schritte ebenso für das *TxData* aus der dritten *Publisher Box* durch.



Schließlich können die beiden leeren *Publisher Boxen* gelöscht werden, indem sie nacheinander selektiert werden und das Kommando [Delete] aufgerufen wird (vgl. Abbildung).



Abschließend sieht die Konfiguration wie in folgender Abbildung aus.

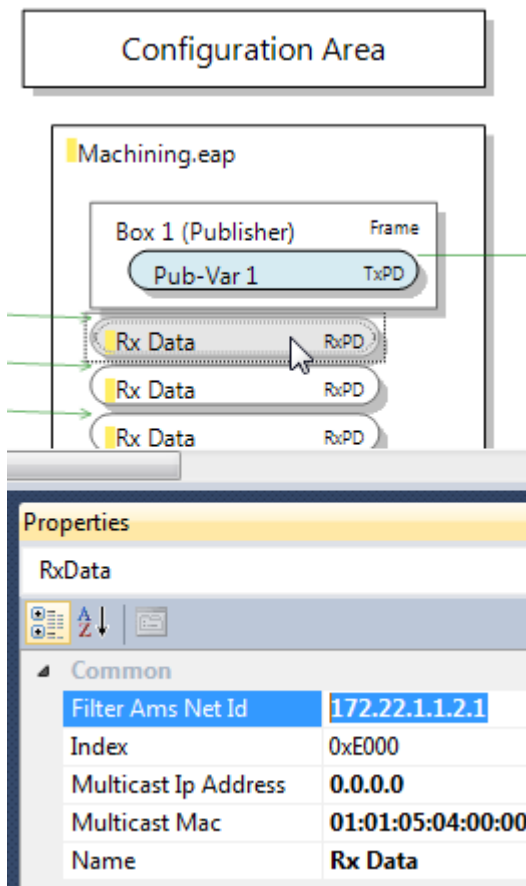


Nachdem die Konfiguration modifiziert worden ist, muss diese noch auf das EAP-Gerät übertragen werden. Diese Maßnahme wird in Kapitel [Die neue/geänderte Konfiguration auf die EAP-Geräte übertragen](#) [► 34] beschrieben.

Eine bestimmte Absender NetID als Empfangsfilter konfigurieren

Standardmäßig ist ein *RxData* so konfiguriert, dass der Absender einer Variablen unerheblich für deren Empfang ist. Vor allem in großen Netzwerken mit vielen EAP-Variablen kann es jedoch sehr hilfreich sein, wenn beim Empfänger festgelegt werden kann, dass eine Variable nur von einem bestimmten Absender empfangen werden soll.

Es wird entsprechend im vorliegenden Beispiel bei der Eigenschaft *Filter AMS Net ID* unter den Eigenschaften des *RxData* die *Local AoE Net ID* des EAP-Gerätes eingetragen, welches die gewünschte Variable versendet (vgl. Abbildung).



Diese Einstellung sorgt dafür, dass die EAP-Variable nur dann empfangen wird, wenn sie von dem EAP-Gerät mit der eingetragenen NetID geschickt worden ist.

Nachdem die Konfiguration modifiziert worden ist, muss diese noch auf das EAP-Gerät übertragen werden. Diese Maßnahme wird in Kapitel [Die neue/geänderte Konfiguration auf die EAP-Geräte übertragen](#) [► 34] beschrieben.

4.3.5 Die neue/geänderte Konfiguration auf die EAP-Geräte übertragen

Nach allen getätigten Maßnahmen wird schlussendlich die Konfiguration der EAP-Geräteobjekte auf die TwinCAT EAP-Geräte übertragen. Für das Schreiben einer Konfiguration muss das betreffende TwinCAT EAP-Gerät erst in den passenden Zustand gesetzt werden (vgl. Kapitel [Eine neue Gerätekonfiguration schreiben](#) [► 43]).

Das Schreiben einer Konfiguration funktioniert nur, wenn das TwinCAT EAP-Gerät im Zustand PREOP ist. Das Umschalten des Zustands wird mit Hilfe der Icon-Leiste im *EAP Config Tool Window* bewerkstelligt (vgl. Kapitel [Das Objektverzeichnis anzeigen](#) [► 44]). Das *EAP Config Tool Window* wird geöffnet, indem das Kommando *Show in Tool Window* aus dem Kontextmenü zum EAP-Geräteobjekt angeklickt wird. Am oberen Rand des Windows befindet sich die Icon-Leiste. Die ersten vier Icons werden verwendet, um den Zustand des TwinCAT EAP-Gerätes zu ändern.

1. Icon  => Init (INIT)
2. Icon  => Pre-Operational (PREOP)
3. Icon  => Safe-Operational (SAFEOP)
4. Icon  => Operational (OP)

Das Schreiben einer Gerätekonfiguration erfolgt dann unter Ausführung des Kommandos *Write All to Device*, zu finden im Kontextmenü des EAP-Geräteobjektes oder in der Icon-Leiste . Mit diesem Kommando werden sämtliche Konfigurationsdaten des EAP-Geräteobjekts an das TwinCAT EAP-Gerät übertragen.

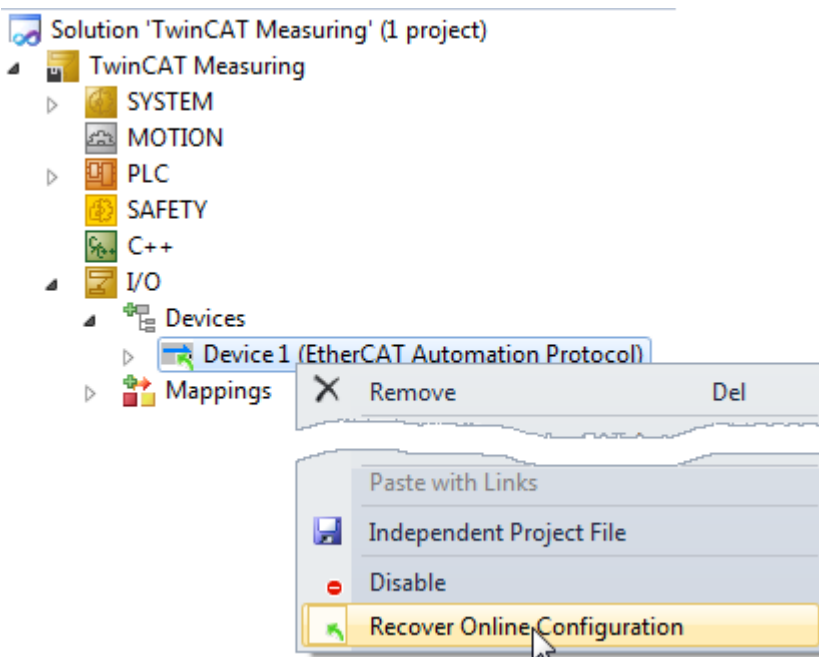
Nachdem die Übertragung abgeschlossen ist, werden die erfolgreich geschriebenen Konfigurationsdaten farblich grün markiert. Die Konfigurationsdaten, deren Übertragung fehlgeschlagen ist, werden dagegen rot markiert. Letzten Endes muss Das EAP-Gerät wieder mit Hilfe des Icons  in den OP-Zustand geschaltet werden, so dass die konfigurierten Prozessdaten zyklisch verarbeitet werden (vgl. Kapitel [Eine neue Gerätekonfiguration schreiben](#) [► 43]).

● Recover Online Configuration

i Eine Konfiguration, die mit Hilfe des TwinCAT EAP Configurator auf ein TwinCAT EAP-Gerät geschrieben worden ist, wird als *online Konfiguration* bezeichnet. Sobald TwinCAT vom *Run*-Modus in den *Stop/Config*-Modus zurückgeschaltet und anschließend wieder in den *Run*-Modus geschaltet wird, geht diese *online Konfiguration* verloren.

Es gibt zwei Methoden, um die *online Konfiguration* wiederherzustellen:

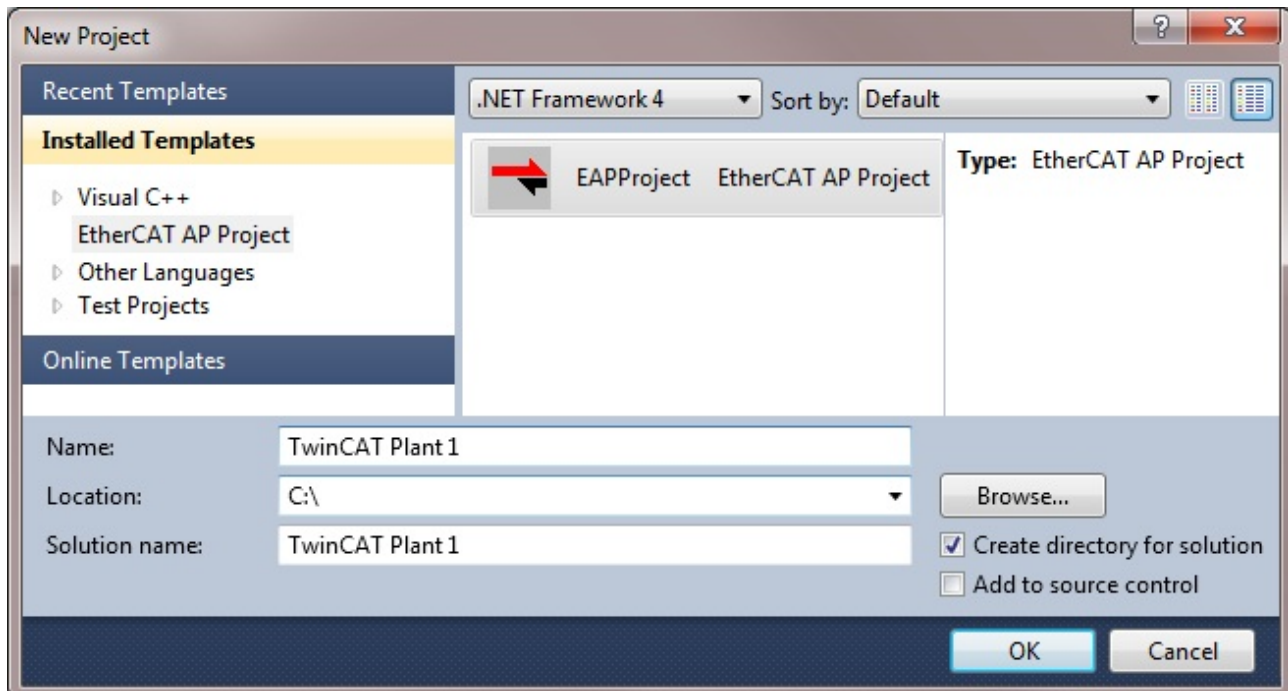
- Mithilfe des TwinCAT EAP Configurator wird die Konfiguration erneut nach dem Starten von TwinCAT auf das TwinCAT EAP-Gerät geschrieben.
- In TwinCAT 3 wird beim EAP-Gerät unter der I/O-Konfiguration die Option *Recover Online Configuration* aktiviert. Diese Option findet sich im Kontextmenü des EAP-Gerätes (siehe Abbildung). Bei dieser Methode wird die *online Konfiguration* beim Zurückschalten von TwinCAT 3 in den *Stop/Config*-Modus persistent auf dem Festspeicher des PCs gesichert. Beim Schalten in den *Run*-Modus wird aus dieser Sicherung die *online Konfiguration* wiederhergestellt.



5 Das TwinCAT EAP Configurator Projekt

Der TwinCAT EAP Configurator ist ein Visual Studio Package und integriert sich somit nahtlos in das Microsoft Visual Studio. Gehen Sie beim Anlegen eines neuen Projektes wie folgt vor:

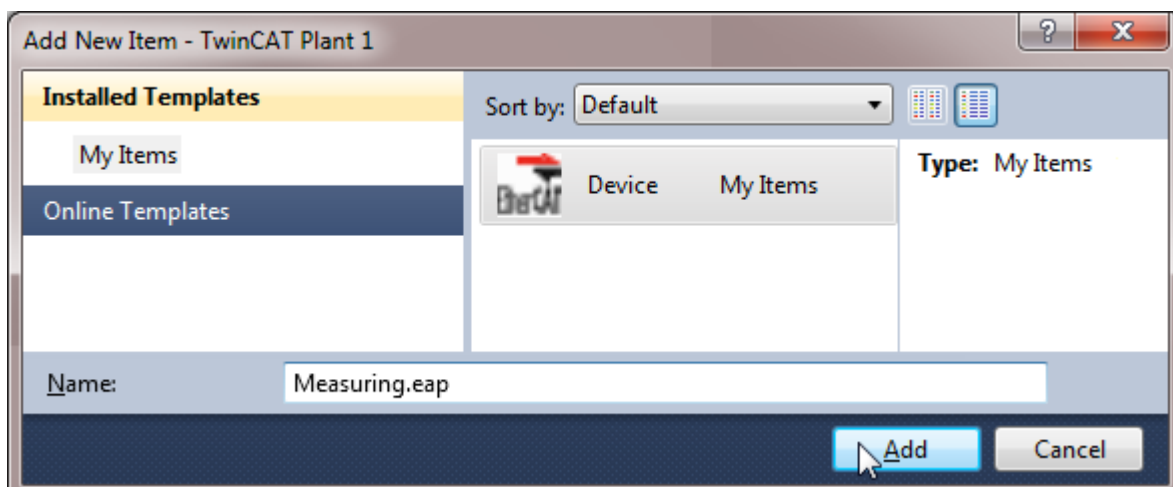
1. Wählen Sie im Visual Studio Menü [File] → [New] → [Project].
⇒ Es öffnet sich der Dialog *New Project*
2. Markieren Sie im linken Auswahlfenster des Dialogs den Menüpunkt *EtherCAT AP Project* (vgl. Abbildung).



1. Legen Sie einen Namen und einen Speicherort für das Projekt fest und klicken Sie auf [OK].
⇒ Das Projekt wird erzeugt und der Solution Explorer enthält einen Projektknoten des gewählten Projekttyps mit der gewünschten Bezeichnung.

Nun können dem Projekt einzelne EAP-Geräteknotten hinzugefügt werden:

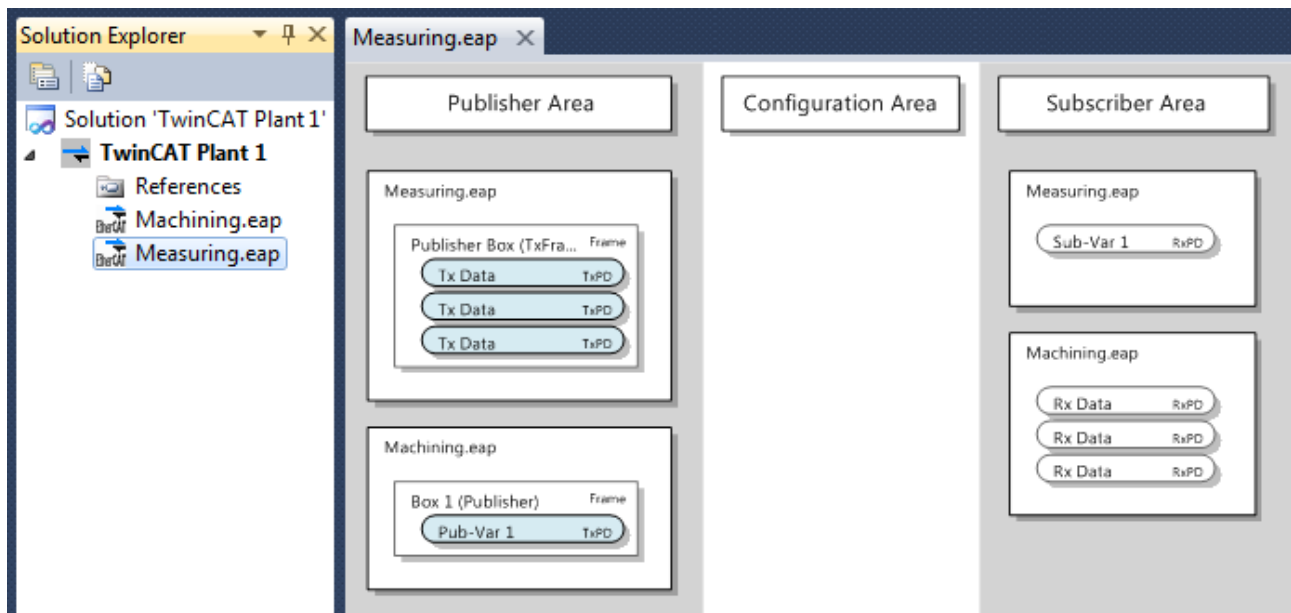
1. Klicken Sie mit der rechten Maustaste auf den Projektknoten und anschließend auf den Menüeintrag [Add] → [New Item].
⇒ Es öffnet sich der Dialog *Add New Item*.



1. Geben Sie eine Bezeichnung für den neuen EAP-Geräteknotten ein und bestätigen Sie den Dialog mit [OK].

Für jedes EAP-Gerät im Netzwerk sollte genau ein EAP-Geräteknotten angelegt werden. Dabei wird jeder Geräteknotten als separate Datei im Projektverzeichnis gespeichert.

Es kann ein beliebiger Geräteknoten aus dem Projekt geöffnet werden. Entweder per Doppelklick über die linke Maustaste oder mit Hilfe des Kommandos [Open] aus dem Kontextmenüs des Geräteknotens. Wenn ein Knoten geöffnet wird, werden alle im Projekt befindlichen Geräteknoten geladen und in der sich öffnenden grafischen Bedienoberfläche dargestellt (vgl. Abbildung). Es wird dabei das EAP-Geräteobjekt an oberster Stelle angezeigt, dessen Geräteknoten geöffnet wurde. Ein EAP-Geräteobjekt repräsentiert also ein EAP-Gerät.



5.1 Die grafische Bedienoberfläche

Die grafische Bedienoberfläche des TwinCAT EAP Configurators dient zum einen der überschaubaren Darstellung der bestehenden Kommunikationsverbindungen zwischen den EAP-Geräten, zum anderen bietet sie eine bequeme Möglichkeit, neue Kommunikationsverbindungen zu konfigurieren. Sie bildet somit eine zentrale Schnittstelle für den Anwender.

5.1.1 Die Architektur

Die grafische Bedienoberfläche gliedert sich in drei Spalten: Links die *Publisher Area*, rechts die *Subscriber Area* und in der Mitte die *Configuration Area*. Nach dem Öffnen der Bedienoberfläche wird jeder Projektknoten darin als EAP-Geräteobjekt grafisch dargestellt. Jedes EAP Geräteobjekt wird als Kasten sowohl in der Spalte *Publisher Area* als auch in der Spalte *Subscriber Area* dargestellt. Mit anderen Worten: Ein EAP-Geräteobjekt wird grafisch von zwei Kästen repräsentiert.

Von oben abgezählt repräsentiert z.B. der 3. Kasten in der *Publisher Area* dasselbe EAP-Gerät wie der 3. Kasten in der *Subscriber Area*. Nach jedem erneuten Öffnen der grafischen Bedienoberfläche ist die *Configuration Area* leer. Sobald dann ein EAP-Geräteobjekt aus einer der beiden anderen *Areas* mit der linken Maustaste doppelt angeklickt wird, werden die beiden entsprechenden Kästen zusammengezogen und in Form eines einzelnen Kastens in der *Configuration Area* angezeigt. Dafür werden die beiden einzelnen Kästen nicht mehr in der *Publisher* bzw. *Subscriber Area* angezeigt.

Zwischen den EAP-Geräteobjekten werden die Kommunikationsverbindungen nur über die Spaltengrenzen zwischen *Publisher Area* zu *Configuration Area* und *Configuration Area* zu *Subscriber Area* angezeigt. Durch diese reduzierte Anzeige wird eine Diagnose der Kommunikationsverbindungen zwischen den Geräten übersichtlich.

In der *Publisher Area* werden nur die Publisher eines EAP-Gerätes angezeigt, deren Aufbau im Abschnitt Aufbau eines EAP-Publishers [► 10] erläutert wird.

In der *Subscriber Area* werden nur die Subscriber eines EAP-Gerätes angezeigt, deren Aufbau im Abschnitt Aufbau eines EAP-Subscribers [► 11] erläutert wird.

Initial wird in der *Configuration Area* kein EAP-Geräteobjekt angezeigt. Erst dann, wenn ein EAP-Geräteobjekt (z.B. *Device1.eap*) aus einer der beiden anderen Areas mit der linken Maustaste doppelt angeklickt wird, werden die beiden zusammengehörigen Kästen aus der *Publisher* und *Subscriber Area*, zu einem Kasten zusammengeführt und in der *Configuration Area* angezeigt. Dafür werden die beiden einzelnen Kästen nicht mehr in der *Publisher* bzw. *Subscriber Area* angezeigt. Es wird immer nur ein einziges Geräteobjekt in der *Configuration Area* angezeigt. Das heißt, dass bei einem Doppelklick auf ein anderes EAP-Geräteobjekt aus der *Publisher/Subscriber Area* (z.B. *Device2.eap*) das *Device1* von der *Configuration Area* wieder zurück in die *Publisher/Subscriber Area* geschoben wird, dafür aber das *Device2* in die *Configuration Area* geholt wird.

In der *Configuration Area* werden sowohl die Publisher als auch die Subscriber des EAP-Gerätes angezeigt. Außerdem werden die Kommunikationsverbindungen zwischen dem mittig angezeigten EAP-Geräteobjekt und den links und rechts angeordneten EAP-Geräteobjekten angezeigt. Alle weiteren Kommunikationsverbindungen, an denen nicht das mittlere EAP-Gerät beteiligt ist, bleiben unsichtbar.

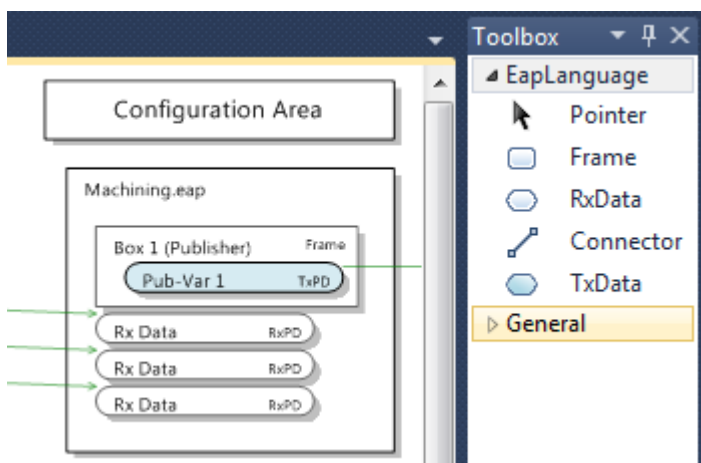
Die Kommunikationsverbindung

Eine Kommunikationsverbindung, die anhand der vorgenommenen Konfiguration zu einem Datenaustausch zwischen einem EAP-Geräteobjekt als Quelle (Publisher/TxData) und einem EAP-Geräteobjekt als Senke (Subscriber/RxData) führt, wird als Pfeil von der Quelle zur Senke dargestellt. Ein Kommunikationspfeil verläuft also immer von einem *TxData* eines Geräteobjektes zu einem *RxData* eines anderen Geräteobjektes. Eine Kommunikationsverbindung innerhalb eines EAP-Geräteobjektes (Loop), also von einem *TxData* zu einem *RxData* desselben EAP-Geräteobjektes, kann weder konfiguriert noch dargestellt werden. Dies entspricht den realen Eigenschaften eines TwinCAT EAP-Gerätes. Die Farbe des Kommunikationspfeils gibt die Verbindungsart an, die seitens des Publishers konfiguriert ist:

Farbe	Verbindungsart
schwarz	Unicast
grün	Multicast
blau	Broadcast

5.1.2 Die Interaktion mit der grafischen Benutzerschnittstelle (GUI)

Das Konfigurieren eines EAP-Gerätes erfolgt mittels der grafischen Bedienoberfläche und der *Toolbox*. Die *Toolbox* befindet sich unter der Sektion *EapLanguage* (vgl. Abbildung). Darin sind die Tools (dt. Werkzeuge) zu finden mit Hilfe derer eine Konfiguration vorgenommen wird.



Das Handling der Tools *Frame*, *TxData* und *RxData* ist gleichartig.

1. Klicken Sie auf eines der Tools mit der linken Maustaste.
2. Ziehen Sie den Mauszeiger auf die Position, wo das neue Element erzeugt werden soll.

Solange der Mauszeiger ein Verbotssymbol (roter Kreis mit einem Querstrich) anzeigt , kann das gewünschte Element an der aktuellen Position nicht erzeugt werden. Andernfalls ändert sich der Mauszeiger zu dem entsprechenden Tool-Symbol.

3. Klicken Sie erneut auf die linke Maustaste.

⇒ Das neue Element wird hinzugefügt und das Standardwerkzeug (*Pointer*) wird wieder aktiv.

● Einfügen mehrere Elementen eines Typs

I Wenn mehrere Elemente eines Typs erzeugt werden sollen, kann bei der Auswahl des Tools ein Doppelklick vorgenommen werden. In diesem Fall bleibt nach dem Einfügen des Elementes das gewählte Tool aktiv, solange bis z.B. die rechte Maustaste gedrückt wird.

Das *Pointer Tool*

Das Standard-Tool ist der *Pointer* . Dieses ist im Regelfall aktiviert. Wenn ein anderes Tool aktiviert worden ist und später die ESC-Taste oder die rechte Maustaste betätigt wird, wird das *Pointer Tool* wieder aktiv.

Mit Hilfe des *Pointer Tools* können einzelne Objekte selektiert werden, um z.B. deren Eigenschaften im Properties-Fenster sehen zu können. Des Weiteren kann das EAP-Geräteobjekt innerhalb der *Configuration Area* vertikal verschoben werden. Und es kann per Doppelklick ein anderes EAP-Geräteobjekt aus der *Publisher/Subscriber Area* in die *Configuration Area* geholt werden. Außerdem können die Namen der Elemente eines EAP-Geräteobjektes per Doppelklick markiert und verändert werden.

Per Rechtsklick auf eines der Elemente öffnet sich ein Kontextmenü, in dem einige Kommandos aufgeführt werden, die auf dem betreffenden Element ausgeführt werden können. Die Standardkommandos [Cut], [Copy], [Paste] und [Delete] werden im Allgemeinen von jedem Element unterstützt. Sie werden verwendet, um Elemente auszuschneiden, zu kopieren, wieder einzufügen oder zu löschen, so wie es aus vielen anderen Anwendungen bekannt ist.

Das Kommando [Properties] öffnet das Properties-Fenster, in dem die Konfigurationseigenschaften des betreffenden Elements wiederzufinden sind.

Das *Frame Tool*

Das *Frame Tool*  wird benutzt, um ein neues *TxFrame*-Element innerhalb eines EAP-Geräteobjektes zu erzeugen. Frame-Elemente können nur innerhalb des Kastens eines EAP-Geräteobjektes erzeugt werden, welcher sich innerhalb der *Publisher Area* oder der *Configuration Area* befindet.

Das *TxData Tool*

Innerhalb eines *TxFrames* werden dann mit Hilfe des *TxData Tools*  *TxData*-Elemente hinzugefügt. An anderen Stellen können keine *TxData*-Elemente erzeugt werden.

Das Kontextmenü des *TxData*-Elements enthält neben den Standardkommandos zusätzlich das Kommando [Process Data Configuration]. Das Kommando öffnet einen Dialog, mit Hilfe dessen *TxPDO*-Elemente erzeugt und bearbeitet werden können. Anschließend kann eines der *TxPDOs* für das *TxData* ausgewählt werden. Eine detaillierte Beschreibung des Dialogs ist im Kapitel [Kommunikationsverbindung manuell erzeugen](#) [► 42] zu finden.

Das *RxData Tool*

Das *RxData Tool*  wird verwendet, um ein neues *RxData*-Element innerhalb eines EAP-Geräteobjektes zu erzeugen. *RxData*-Elemente können nur innerhalb des Kastens eines EAP-Geräteobjektes erzeugt werden, welcher sich innerhalb der *Subscriber Area* oder der *Configuration Area* befindet.

Wie beim *TxData*-Element enthält auch das Kontextmenü des *RxData*-Elements neben den Standardkommandos zusätzlich das Kommando [Process Data Configuration], um *RxPDO*-Elemente zu erzeugen, zu bearbeiten und dem *RxData* zuzuweisen (vgl. Kapitel [Kommunikationsverbindung manuell erzeugen](#) [► 42]).

Das Connector Tool

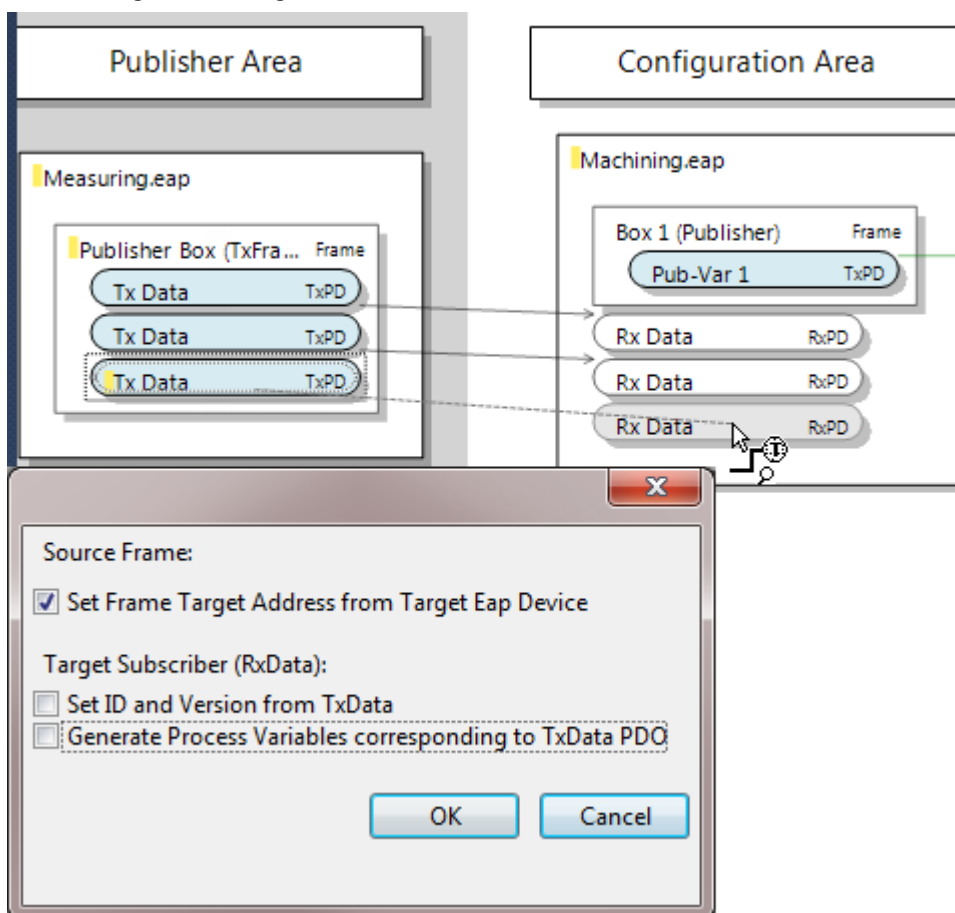
Das *Connector Tool*  ist ein Assistent, um halbautomatisch eine Kommunikationsverbindung zwischen zwei EAP-Geräteobjekten zu konfigurieren. Dazu werden zwei Anwendungsfälle unterstützt: *TxData* zu *RxData* und *EAP-Geräteobjekt* zu *EAP-Geräteobjekt*.

TxData zu RxData

Der erste Fall ist die Verwendung eines *Connectors* zwischen *TxData* und *RxData*. Bei dieser Aktion werden die Eigenschaften der betreffenden Publisher-Elemente und Subscriber-Elemente so angepasst, dass eine Kommunikationsverbindung zustande kommt.

1. Wählen Sie das *Connector Tool* aus.
2. Ziehen Sie den Mauszeiger auf das gewünschte *TxData*-Element.
3. Drücken Sie die linke Maustaste.
4. Ziehen Sie den Mauszeiger auf das gewünschte *RxData*-Element.
5. Drücken Sie die linke Maustaste erneut.

⇒ Der folgende Dialog öffnet sich.



Hier werden drei Optionen aufgeführt, die nach Bedarf und abhängig vom aktuellen Konfigurationsstand gesetzt werden können:

- *Set Frame Target Address from Target EAP Device.*
Diese Option ist standardmäßig aktiviert. In diesem Fall wird beim automatischen Erzeugen des *TxFrames*, als Zieladresse die *AMS NetID* des Ziel EAP-Gerätes eingetragen.
- *Set ID and Version from TxData.*
Diese Option sorgt dafür, dass bei der Konfiguration des Subscriber Elements *RxData*, die Identifikationsnummer und die Versionsnummer des gewählten *TxData*-Elements übernommen werden. Ohne eine Übereinstimmung dieser beiden Eigenschaften kommt keine Kommunikationsverbindung zustande.

- *Generate Process Variables corresponding to TxData PDO.*

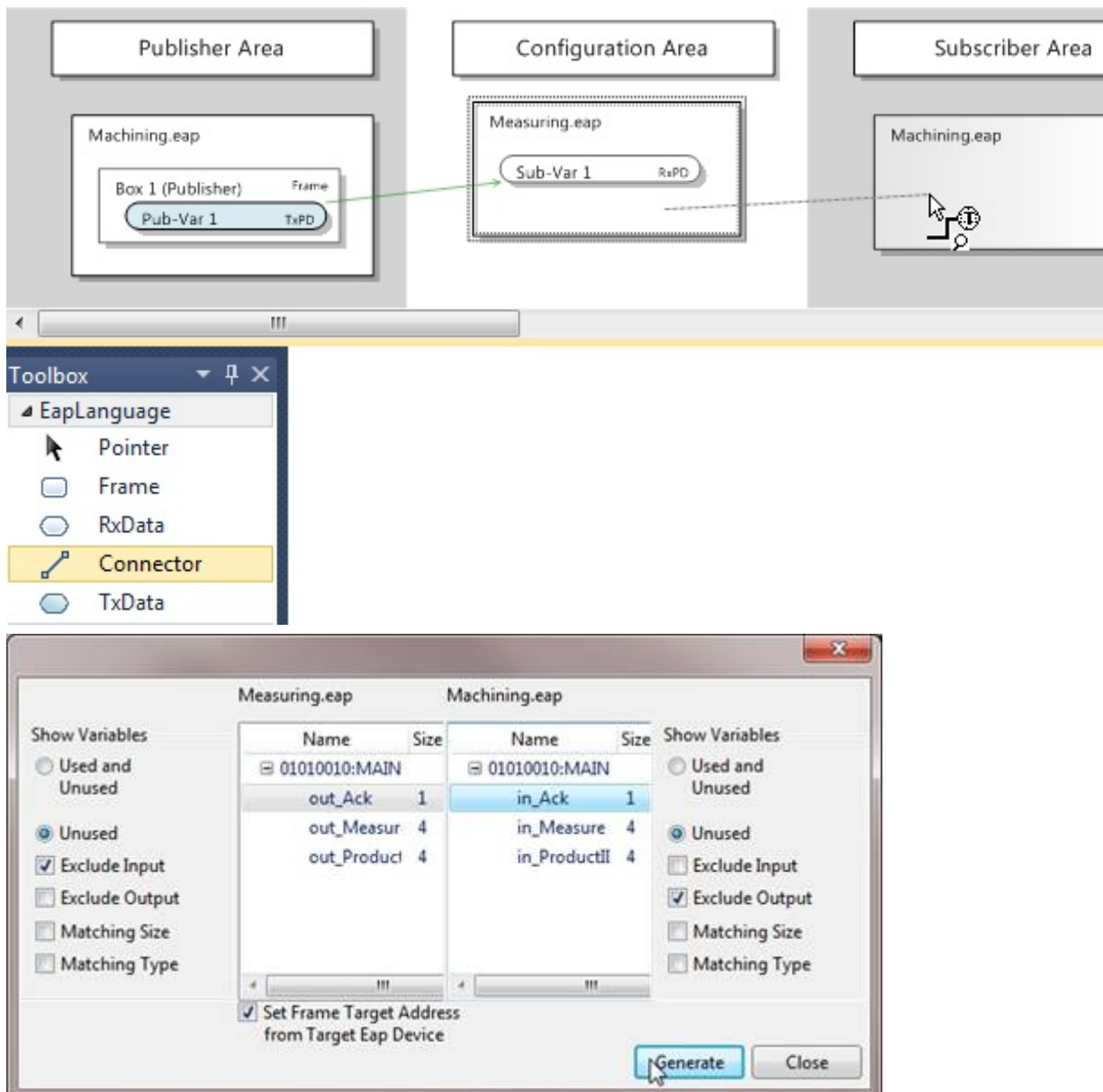
Diese Option steht nur dann zur Verfügung, wenn auf der Publisher Seite für das *TxData* bereits ein *TxPDO* definiert und konfiguriert worden ist. Bestenfalls sollte das *TxPDO* bereits auch schon die vollständige Liste der Referenzen auf die gewünschten *TxVariablen* haben. Denn so wird auf der Empfangsseite gleich das gesamte *RxPDO* inklusive aller notwendigen *RxVariablen* erzeugt und konfiguriert. Entsprechend ist dann auch die Kommunikationsverbindung korrekt konfiguriert.

EAP-Geräteobjekt zu EAP-Geräteobjekt

Der zweite Fall ist die Verwendung eines *Connectors* zwischen zwei EAP-Geräteobjekten. Bei dieser Aktion wird beim EAP-Geräteobjekt mit dem Startpunkt des *Connectors* ein Publisher mit all seinen notwendigen, geschachtelten Elementen erzeugt und vorkonfiguriert. Beim EAP-Geräteobjekt mit dem Endpunkt des *Connectors* wird ein Subscriber mit all seinen notwendigen, geschachtelten Elementen erzeugt und entsprechend zum erzeugten Publisher vorkonfiguriert, so dass eine gültige Kommunikationsverbindung vorliegt.

1. Wählen Sie das *Connector* Tool aus.
2. Ziehen Sie den Mauszeiger auf die freie Fläche eines EAP-Geräteobjektes, welches als Publisher fungieren soll.
3. Drücken Sie die linke Maustaste.
4. Ziehen Sie den Mauszeiger auf eine freie Fläche des gewünschten Ziel EAP-Geräteobjektes.
5. Betätigen Sie die linke Maustaste erneut.

⇒ Der folgende Dialog öffnet sich.



Der Dialog ist in zwei Hälften aufgeteilt. In der linken Hälfte werden in einem Listenfeld alle Symbolnamen von Prozessvariablen aufgelistet, deren Daten mit Hilfe des EAP-Gerätes gesendet werden können. Die Prozessvariablen stammen z.B. aus einem SPS Programm der Maschine, bei der auch das EAP-Gerät instanziiert worden ist.

In der rechten Hälfte werden analog die Symbolnamen der Prozessvariablen der Zielmaschine aufgelistet, in die das Ziel EAP-Gerät die empfangenen Daten schreiben kann. Wenn keine Symbolnamen aufgelistet werden, kann auch keine Kommunikationsverbindung generiert werden.

Der Dialog bietet verschiedene Optionen, um die Symbollisten zu filtern. Zunächst gibt es die Möglichkeit, alle verfügbaren Symbolnamen anzeigen zu lassen: Also bereits verwendete zusammen mit noch nicht verwendeten (*Used and Unused*) oder nur die noch nicht verwendeten (*Unused*).

Darüber hinaus kann ausgewählt werden, ob die Symbole der Eingangsvariablen (*Exclude Output* aktiviert) oder die Symbole der Ausgangsvariablen (*Exclude Input* aktiviert) oder ggf. auch beide Arten aufgelistet werden sollen (weder *Exclude Input* noch *Exclude Output* aktiviert). Die Standardeinstellung zeigt auf der linken Seite (beim EAP-Publisher) nur die Symbole der Ausgangsvariablen an und auf der rechten Seite (beim EAP-Subscriber) nur die Symbole der Eingangsvariablen.

Sofern ein Symbol aus einer der beiden Listen selektiert worden ist, können die beiden Filteroptionen *Matching Size* und *Matching Type* sinnvoll eingesetzt werden. Ist beispielsweise auf der linken Seite beim EAP-Publisher das Symbol einer Prozessvariablen selektiert, kann auf der rechten Seite beim EAP-Subscriber die Symbolnamenliste so gefiltert werden, dass nur noch die Symbolnamen der Prozessvariablen aufgeführt werden,

- mit einer *Matching Size*
deren Datentyp die gleiche Größe hat, wie der Datentyp der Prozessvariablen des Publishers.
- mit einem *Matching Type*
deren Datentyp der gleiche ist, wie der Datentyp der Prozessvariablen des Publishers.
- oder mit einer *Matching Size* und mit einem *Matching Type*
bei denen die beiden Eigenschaften *Matching Size* und *Matching Type* erfüllt sind.

Zuletzt gibt es noch die Option *Set Frame Target Address from Target EAP Device*. Diese Option ist standardmäßig aktiviert. In diesem Fall wird beim automatischen Erzeugen des *TxFrames*, als Zieladresse die *AMS NetID* des Ziel EAP-Gerätes eingetragen.

Das EAP- Geräteobjekt

Das EAP- Geräteobjekt bietet über sein Kontextmenü verschiedene Funktionen an, die das Lesen und Schreiben, Erzeugen und Löschen oder Importieren und Exportieren einer Konfiguration erst ermöglichen. So kann eine Kommunikationsverbindung zwischen zwei EAP- Geräteobjekten z.B. auch ohne die Verwendung des Connector Tools manuell konfiguriert werden.

Eine Kommunikationsverbindung manuell erzeugen

Wenn eine Kommunikationsverbindung zwischen zwei EAP-Geräteobjekten manuell erzeugt werden soll, ist die ideale Vorgehensweise so, dass als erstes die Variablen und Process Data Objects (PDOs) erzeugt und konfiguriert werden (siehe dazu Kapitel [Variablen erzeugen](#) [► 49] und [Process Data Objects erzeugen](#) [► 51]). Ein Tx/Rx-PDO repräsentiert dann eine Dateneinheit, zu der die Versand- bzw. Empfangseigenschaften mit Hilfe eines TxData- und TxFrame- bzw. eines RxData-Elementes konfiguriert werden. Nachdem die gewünschten PDOs erstellt worden sind, können also die notwendigen TxFrames, TxData und RxData erzeugt (vgl. Kapitel [Das Frame Tool](#) [► 39], [Das TxData Tool](#) [► 39] und [Das RxData Tool](#) [► 39]) und vollständig konfiguriert werden (vgl. Kapitel [Der Frame](#) [► 54], [Das TxData](#) [► 56] und [Das RxData](#) [► 58]).

Importieren/Exportieren

Die Konfiguration eines EAP- Gerätes kann über das Kommando [Export EAP Device Config File] in eine XML-basierte Datei exportiert werden. Diese Datei wird daher auch als EAP Device Configuration (kurz: EDC) bezeichnet. Ein TwinCAT EAP- Gerät, welches mit Hilfe von TwinCAT 3 angelegt und konfiguriert worden ist, bietet ebenfalls die Funktion an, seine Konfiguration in eine EDC-Datei zu exportieren. Diese Datei kann dann beim TwinCAT EAP Configurator mittels Importfunktion [Import EAP Device Config File] eingelesen werden. Beim Import einer EDC-Datei wird die bestehende Konfiguration verworfen und ist damit verloren, sofern sie nicht zuvor gesichert worden ist.

Eine aus TwinCAT 3 heraus exportierte EDC-Datei enthält neben der EAP- Konfiguration auch noch die Symbolinformationen aller im TwinCAT Projekt angelegten SPS Programme. Nach einem Import der EDC-Datei in den TwinCAT EAP Configurator stehen diese Symbolinformationen für das Anlegen von Kommunikationsverbindungen zur Verfügung.

Eine aktive Gerätekonfiguration lesen

Das Kommando [Read from Device] aus dem Kontextmenü des EAP-Geräteobjektes liest die aktuelle Gerätekonfiguration eines TwinCAT EAP-Gerätes. Dazu müssen drei Voraussetzungen erfüllt sein:

- Das TwinCAT EAP-Gerät muss aktiv sein.
- Es muss eine ADS/AMS Route eingetragen sein von dem PC, auf dem der TwinCAT EAP-Configurator läuft, zu dem PC, auf dem das TwinCAT EAP-Gerät läuft.
- Beim TwinCAT EAP Configurator muss unter den Eigenschaften des EAP-Gerätes die *Local AoE Net ID* richtig eingetragen sein.

Ein TwinCAT EAP-Gerät ist erst dann aktiv, wenn es per TwinCAT 3 projiziert und aktiviert wurde. Trifft eine dieser Voraussetzungen nicht zu, kann der TwinCAT EAP Configurator auch nicht mit dem EAP-Gerät kommunizieren. Nach dem Lesen der aktuellen EAP-Konfiguration ist diese anhand der grafischen Darstellung direkt sichtbar.



Die aktuelle Gerätekonfiguration wird verworfen

Wenn eine Gerätekonfiguration gelesen wird, wird die bestehende verworfen und ist damit verloren, sofern sie nicht zuvor gesichert worden ist.

Eine neue Gerätekonfiguration schreiben

Eine der wichtigsten Funktionen, die der TwinCAT EAP Configurator zur Verfügung stellt, ist das Schreiben einer Gerätekonfiguration. Die komfortable Möglichkeit, eine EAP-Konfiguration mit Hilfe des TwinCAT EAP Configurators zu erzeugen bzw. zu modifizieren, führt zu dem Bedarf, diese Konfiguration auch auf ein TwinCAT EAP-Gerät zu übertragen. Neben den Voraussetzungen, die bereits für das Lesen einer EAP-Konfiguration erfüllt sein müssen, muss für das Schreiben noch ein weiterer Punkt beachtet werden.

Bevor eine Konfiguration auf ein TwinCAT EAP-Gerät übertragen werden kann, muss dieses erst in einen geeigneten Zustand versetzt werden, in dem das Verändern der Konfiguration unkritisch ist. Das TwinCAT EAP-Gerät ist mit einem Zustandsautomaten (State Machine) ausgestattet, der vier Zustände innehat.

- Init (INIT)
Das TwinCAT EAP-Gerät befindet sich niemals im Zustand INIT, außer kurzzeitig beim Aktivieren des TwinCAT Projektes. In diesem Fall ist das EAP-Gerät bereits instanziiert, jedoch noch nicht konfiguriert worden.
- Pre-Operational (PREOP)
Im Zustand PREOP ist das TwinCAT EAP-Gerät vollständig vorkonfiguriert, so dass einer zyklischen Verarbeitung der Prozessdaten nichts entgegensteht. Im Zustand PREOP können zusätzlich weitere Konfigurationen z.B. mit Hilfe des TwinCAT EAP Configurators an dem TwinCAT EAP-Gerät vorgenommen werden.
- Safe-Operational (SAFEOP)
Beim Schalten des EAP-Gerätes vom Zustand PREOP in den Zustand SAFEOP wird eine Konsistenzprüfung der aktuellen Konfiguration durchgeführt, werden Referenzen zwischen Objekten erstellt, die per Konfiguration definiert worden sind, und werden die Verknüpfungen zu den SPS Variablen anhand der konfigurierten Symbolnamen erzeugt. Im Zustand SAFEOP werden bereits Variablen vom EAP-Gerät gesendet, jedoch noch nicht empfangen.
- Operational (OP)
Im Zustand OP werden schließlich die konfigurierten Prozessdaten vom EAP-Gerät zyklisch und hochdeterministisch empfangen und versendet.

Das TwinCAT EAP-Gerät muss vor dem Schreiben der neuen Gerätekonfiguration in den Zustand PREOP geschaltet werden. Das Umschalten des Zustands wird mit Hilfe der Icon-Leiste im *EAP Config Tool Window* bewerkstelligt (siehe Kapitel [Das Objektverzeichnis anzeigen](#) ► 44).

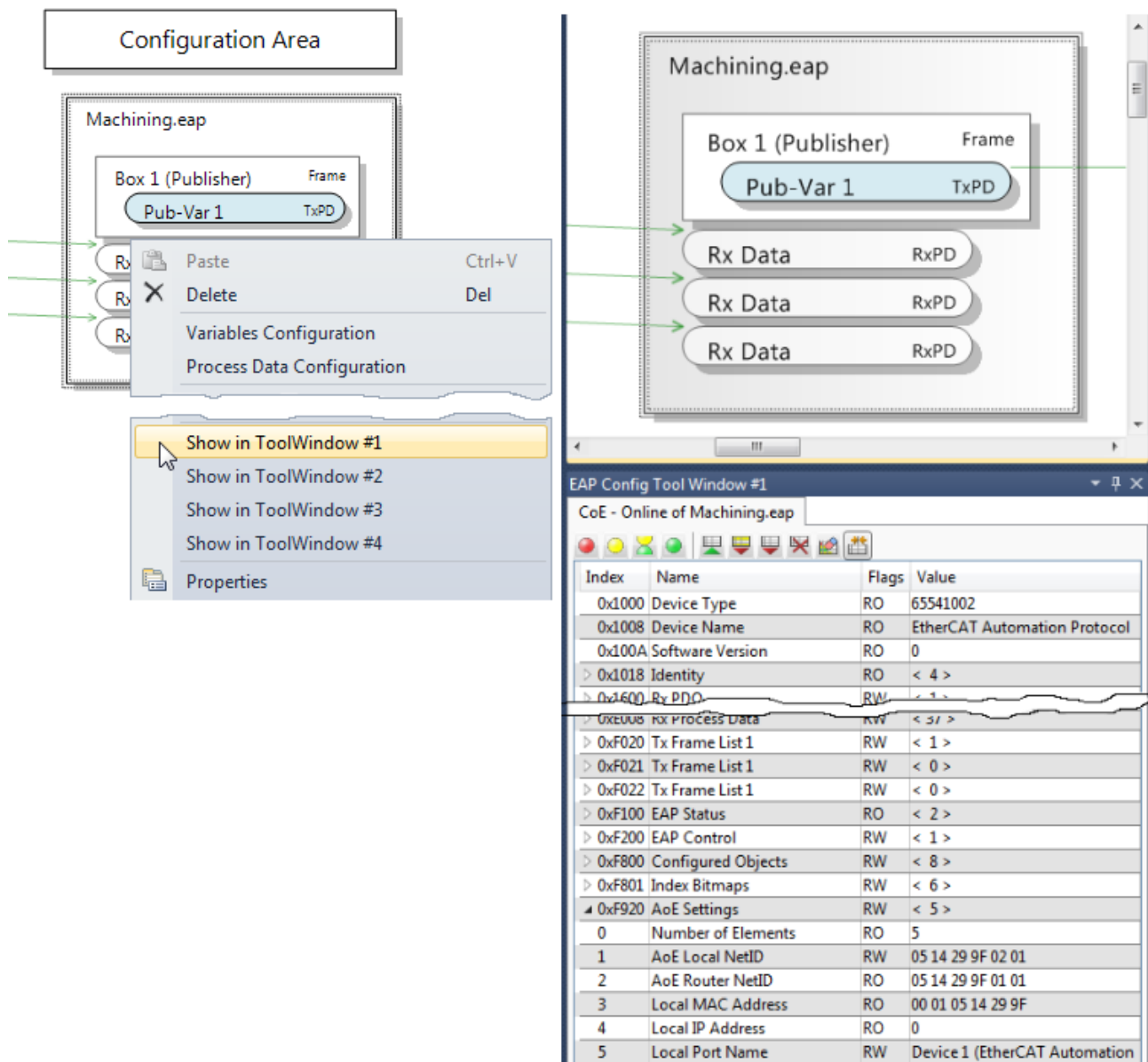
Das Schreiben einer Gerätekonfiguration kann auf zwei Arten erfolgen:

- **Write All to Device**
Mit dem Kommando [Write All to Device] werden sämtliche Konfigurationsdaten, die zum EAP-Geräteobjekt im TwinCAT EAP Configurator gehören, an das TwinCAT EAP-Gerät übertragen. All diese Konfigurationsdaten werden zusammengefasst mit Hilfe des Objektverzeichnisses abgebildet (vgl. Kapitel [Das Objektverzeichnis](#) [► 61]). Das Kommando [Write All to Device] kann auch per Icon  aus der Icon-Leiste aufgerufen werden.
- **Write Changes to Device**
Mit dem Kommando [Write Changes to Device] werden nur die Konfigurationsdaten an das TwinCAT EAP-Gerät übertragen, die mit Hilfe des TwinCAT EAP Configurators modifiziert worden sind. Die modifizierten Änderungen werden mit einer gelben Markierung versehen, die sowohl im Objektverzeichnis, als auch auf der grafischen Oberfläche angezeigt wird. Im Objektverzeichnis kann im Detail nachgeschaut werden, welcher Eintrag geändert wurde. Auch das Kommando [Write Changes to Device] kann per Icon  aus der Icon-Leiste aufgerufen werden.

Nachdem die Übertragung abgeschlossen worden ist, wird die Markierung der erfolgreich geschriebenen Konfigurationsdaten auf die Farbe Grün gesetzt. Die Konfigurationsdaten, deren Übertragung fehlgeschlagen ist, werden dagegen rot markiert.

Das Objektverzeichnis anzeigen

Sämtliche Konfigurationsdaten, die zum EAP-Geräteobjekt im TwinCAT EAP Configurator gehören, werden zusammengefasst mit Hilfe des Objektverzeichnisses abgebildet. Dieses Objektverzeichnis wird in tabellarischer Form angezeigt, indem das Kommando [Show in ToolWindow] aus dem Kontextmenü zum EAP-Geräteobjekt angeklickt wird (vgl. Abbildung). Es werden insgesamt maximal vier dieser ToolWindows unterstützt, so dass maximal die Objektverzeichnisse von vier verschiedenen EAP-Geräten gleichzeitig betrachtet werden können.



Am oberen Rand eines EAP Config Tool Windows wird der Dateiname des zugehörigen EAP-Geräteobjektes angezeigt und darunter eine Icon-Leiste, mit Hilfe derer einige Kommandos ausgeführt werden können (vgl. Abbildung oben), die im Folgenden erläutert werden.

Set EAP Device State

Mit Hilfe der ersten vier Icons wird der Zustand des TwinCAT EAP-Gerätes verändert. Für jeden der vier Zustände des EAP-Zustandsautomaten (State Machine) gibt es ein Icon.

1.	Icon		Schaltet das EAP-Gerät in den Init (INIT = Wert 1)
2.	Icon		Schaltet das EAP-Gerät in den Pre-Operational (PREOP = Wert 2)
3.	Icon		Schaltet das EAP-Gerät in den Safe-Operational (SAFEOP = Wert 4)
4.	Icon		Schaltet das EAP-Gerät in den Operational (OP = Wert 8)

Read from Device

Das Icon führt das Kommando [Read from Device] aus, beschrieben in Kapitel [Eine aktive Gerätekonfiguration lesen](#) [► 43].

Write Changes to Device

Das Icon  führt das Kommando [Write Changes to Device] aus, beschrieben in Kapitel [Eine neue Gerätekonfiguration schreiben](#) [► 43].

Write All to Device

Das Icon  führt das Kommando [Write All to Device] aus, beschrieben in Kapitel [Eine neue Gerätekonfiguration schreiben](#) [► 43].

Clear Object Dictionary

Das Icon  führt das Kommando [Clear Object Dictionary] aus, beschrieben in Kapitel [Die Konfiguration löschen](#) [► 46].

Clear Change Flags

Das Icon  führt das Kommando [Clear Change Flags] aus, beschrieben in Kapitel [Die Anzeige des Konfigurationszustands](#) [► 46].

Compare to Online Data

Mit Hilfe des Icons  wird dem Objektverzeichnis eine weitere Spalte hinzugefügt, in der die Konfigurationsdaten angezeigt werden, die beim aktuell aktiven TwinCAT EAP-Gerät eingestellt sind. Das Icon funktioniert wie ein Kippschalter, so dass die Spalte beim erstmaligen Anklicken eingeblendet und beim erneuten Klicken wieder ausgeblendet wird. Bei jedem Einblenden werden die Konfigurationsdaten erneut aus dem TwinCAT EAP-Gerät gelesen und angezeigt.

Die Konfiguration löschen

Das Löschen der Konfiguration erfolgt über das Kommando [Clear Object Dictionary] aus dem Kontextmenü des EAP-Geräteobjektes. Eine Konfiguration kann separat für jedes EAP-Geräteobjekt gelöscht werden. Dabei werden alle bis auf ein paar wenige Objekte aus dem Objektverzeichnis des betreffenden EAP-Geräteobjektes gelöscht, so dass eine leere Konfiguration vorliegt. Die Objekte, die nicht gelöscht werden, sind Standardobjekte, die zu jedem EAP-Gerät gehören und daher auch nicht gelöscht werden können. Darin sind u.a. der Status des EAP-Geräts oder die Zieladressen wie MAC, IP bzw. AMS NetID wiederzufinden. Diese Informationen werden u.a. auch vom TwinCAT EAP Configurator benötigt, um mit dem TwinCAT EAP-Gerät ordnungsgemäß kommunizieren zu können. Die Objekte, die gelöscht werden, sind die dynamischen Objekte, die während des Anlegens neuer Elemente wie TxFrames, Tx/Rx Data, Tx/Rx PDOs oder Tx/Rx Variables erzeugt werden.

Die Anzeige des Konfigurationszustands

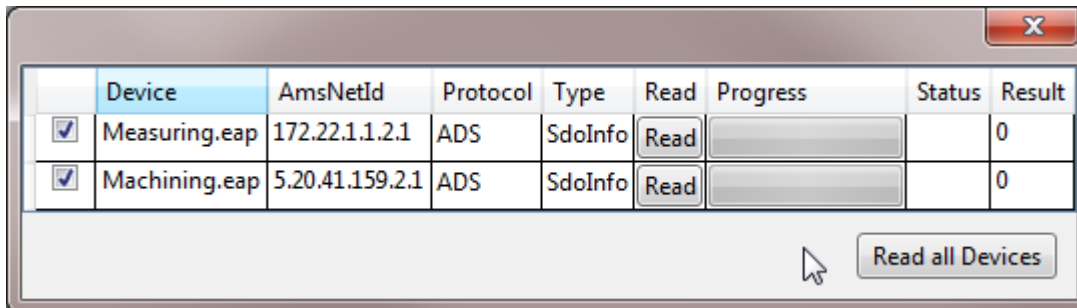
Der TwinCAT EAP-Configurator führt während seiner Verwendung verschiedene Aktionen auf den Konfigurationsdaten der EAP-Geräteobjekte aus. Einige dieser Aktionen werden auf der grafischen Oberfläche sowie in der tabellarischen Anzeige des Objektverzeichnisses durch farbliche Markierungen kenntlich gemacht. Diese Kennzeichnung dient als Anzeige des Konfigurationszustands und hat folgende Bedeutung:

Markierung	Bedeutung
keine	Keine Markierung ist in den folgenden Fällen vorzufinden: • Das Projekt ist unmittelbar geöffnet worden • Die Kennzeichnungen wurden explizit zurückgesetzt
gelb	Die Konfigurationsdaten haben sich verändert.
grün	Die Konfigurationsdaten wurden erfolgreich geschrieben.
rot	Das Schreiben der Konfigurationsdaten ist fehlgeschlagen.

Das Zurücksetzen der Kennzeichnung erfolgt mit Hilfe des Kommandos [Clear Change Flags] aus dem Kontextmenü des EAP-Gerätes.

Die Oberfläche

Das Kontextmenü der grafischen Oberfläche selbst bietet zusätzlich noch zwei bedeutende Kommandos: [Read All Devices] und [Write All Devices]. Durch Anklicken eines dieser Kommandos öffnet sich ein tabellarischer Dialog, in dem alle EAP-Geräteobjekte zeilenweise aufgelistet werden, die im TwinCAT EAP Configurator Projekt angelegt worden sind (siehe Abbildung).



Read All Devices

Der Dialog aus oberen Abbildung zeigt eine tabellarische Aufstellung aller EAP-Geräteobjekte des Projekts. Neben einer Checkbox und einer Schaltfläche pro Zeile werden folgende Informationen in den Tabellenspalten angezeigt:

- Der Dateiname des EAP-Geräteobjektes
- Die beim EAP-Geräteobjekt eingetragene AMS NetID
- Das Kommunikationsprotokoll zur Gerätekonfiguration
- Eine Fortschrittsanzeige
- Eine Statusanzeige
- Eine Ergebnisanzeige

Mit Hilfe des Dialogs können die Konfigurationen aller aufgelisteten EAP-Geräte gelesen werden, sofern die notwendigen Voraussetzungen dafür erfüllt sind (vgl. Kapitel [Eine aktive Gerätekonfiguration lesen](#) ► 43] f.). Dazu stehen drei Wege offen:

I. Konfiguration aller aufgelisteten EAP-Geräte lesen

✓ Alle Checkboxes müssen markiert sein, dies ist der Standardzustand beim Öffnen des Dialogs.

1. Klicken Sie auf die Schaltfläche [Read all Devices]

⇒ Alle EAP-Gerätekonfigurationen werden gelesen.

II. Konfiguration eines bestimmten EAP-Gerätes lesen

1. Klicken Sie auf die [Read]-Schaltfläche in der Zeile, in der das gewünschte EAP-Gerät aufgelistet wird.

⇒ Nur die Konfiguration des gewählten EAP-Gerätes wird gelesen.

III. Konfiguration von einer Auswahl von EAP-Geräten lesen

1. Treffen Sie eine beliebige Auswahl von EAP-Geräten unter Verwendung der Checkboxes, deren Konfigurationen Sie lesen möchten

2. Klicken Sie auf die Schaltfläche [Read all Devices].

⇒ Die Konfigurationen der markierten EAP-Geräte werden gelesen

Write All Devices

Das Schreiben der Konfigurationen auf alle EAP-Geräte, funktioniert analog zum Lesen (siehe vorigen Abschnitt).

Protokoll auswählen (ADS, UDP)

In der Spalte *Protocol* kann für jedes aufgelistete EAP-Geräteobjekt das gewünschte Protokoll gewählt werden, das zur Konfiguration des entsprechenden EAP-Gerätes verwendet werden soll. Zur Auswahl stehen ADS und UDP. Dabei bedeutet UDP, dass automatisch das Anwendungsprotokoll AoE per UDP verwendet wird (vgl. [Das Funktionsprinzip des TwinCAT EAP Configurators](#) [► 9]).



EAP Configuration Tool und TwinCAT EAP-Gerät auf demselben Rechner

Wenn das EAP Configuration Tool und das TwinCAT EAP-Gerät auf demselben Rechner ausgeführt werden, kann zur Konfiguration nur das ADS/AMS Protokoll verwendet werden. Das AoE Protokoll funktioniert in dieser Konstellation nicht, weil ein TwinCAT EAP-Gerät AoE-Pakete ausschließlich direkt über den TwinCAT eigenen echtzeitfähigen Netzwerkkartentreiber empfängt und verschickt. Somit gibt es keine Möglichkeit PC-intern AoE-Pakete zwischen dem EAP Configurator Tool und dem TwinCAT EAP-Gerät zu übertragen. PC-intern funktioniert also nur ADS/AMS-Kommunikation, bei der die ADS/AMS-Pakete vom TwinCAT Router auch innerhalb des PCs an die adressierten Empfänger weitergeleitet werden.

Fortschrittsanzeige

Die Fortschrittsanzeige dient dazu, einen Überblick zu erhalten, wie lange die Konfiguration eines jeden EAP-Geräteobjektes geschrieben/gelesen wird und um zu sehen, wann der Schreibvorgang bzw. Lesevorgang beendet ist.

Status und Result

In der Spalte *Status* wird nach dem Schreiben bzw. Lesen der Konfiguration angezeigt, ob der Vorgang bereits beendet worden ist und ob dabei Fehler aufgetreten sind.

In der Spalte *Result* wird entweder 0 angezeigt, sofern kein Fehler aufgetreten ist, oder bei einem Fehler der Fehlercode angezeigt.

5.1.3 Die Konfiguration eines EAP-Geräteobjektes

Die Konfiguration eines EAP-Geräteobjektes erfolgt in mehreren Schritten. Wenn nicht das Connector Tool als Assistent verwendet wird (siehe Kapitel [Das Connector Tool](#) [► 40]), müssen für eine EAP-Publisher-Variable die Elemente TxVariable, TxPDO, TxData und TxFrame Schritt für Schritt einzeln erzeugt und per Konfiguration untereinander referenziert werden. Für eine EAP-Subscriber-Variable sind es entsprechend die Elemente RxVariable, RxPDO und RxData.

I. Manuelle Konfiguration einer Publisher Variablen

1. Erzeugen und konfigurieren Sie *TxVariables* und *TxProcessDataObjects* wie in Kapitel [Variablen erzeugen](#) [► 49] beschrieben.
 2. Erzeugen und konfigurieren Sie TxData laut der Beschreibung aus Kapitel [Das TxData Tool](#) [► 39] und aus Kapitel [Das TxData](#) [► 56].
 3. Erzeugen und konfigurieren Sie einen TxFrame wie in Kapitel [Das Frame Tool](#) [► 39] bzw. Kapitel [Der Frame](#) [► 54].
- ⇒ Eine EAP-Publisher Variable ist vollständig definiert.

i Notwendige Konfigurationseigenschaften für einen erfolgreichen Datentransfer

Bei der Konfiguration einer Publisher Variablen ist darauf zu achten, dass mindestens die folgenden Eigenschaften konfiguriert werden, so dass ein Datentransfer stattfinden kann:

- Beim TxFrame muss...
... die Eigenschaft *Enabled* auf *true* gesetzt sein und
... eine gültige Zieladresse definiert sein.
- Beim TxData muss...
... die Eigenschaft *Enabled* auf *true* gesetzt sein,
... eine *PDO Nummer* eingetragen sein, mit Hilfe derer ein *TxProcessDataObject* referenziert wird und
... ein *Trigger Condition* inklusive eines gültigen *Triggers* wie z.B. eine *Cycle Time* definiert sein.
- Das TxPDO muss...
... wenigstens eine Variable enthalten.

II. Manuelle Konfiguration einer Subscriber Variablen

1. Erzeugen und konfigurieren Sie RxVariables und RxProcessDataObjects wie in Kapitel [Variablen erzeugen](#) [► 49] beschrieben.
 2. Erzeugen und konfigurieren Sie RxData laut der Beschreibung aus Kapitel [Das RxData Tool](#) [► 39] und [Das RxData](#) [► 58].
- ⇒ Eine EAP-Subscriber Variable ist vollständig definiert.

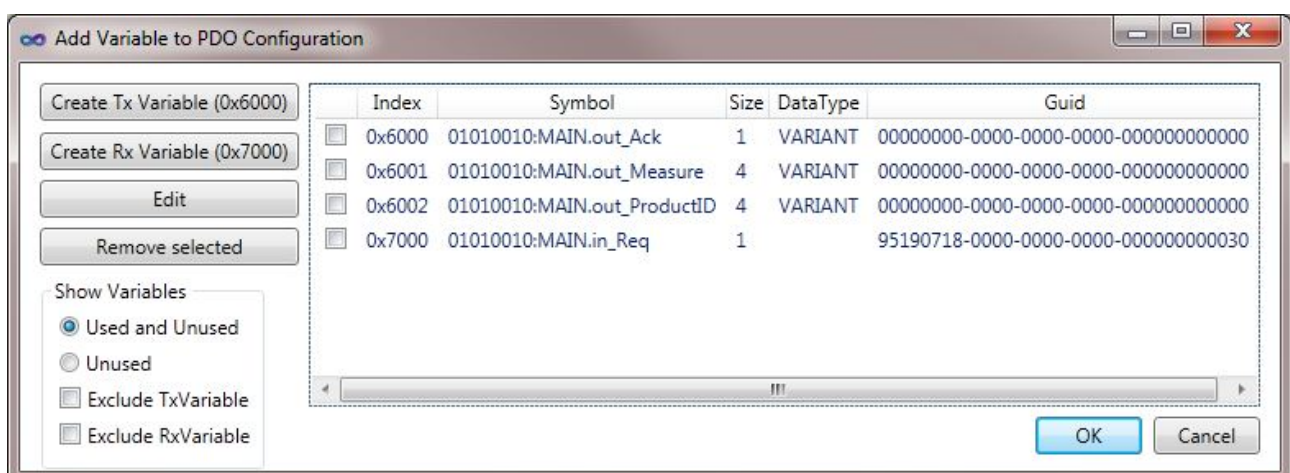
i Notwendige Konfigurationseigenschaften für einen erfolgreichen Datenempfang

Bei der Konfiguration einer Subscriber Variablen ist darauf zu achten, dass mindestens die folgenden Eigenschaften konfiguriert werden, so dass Daten empfangen werden können:

- Beim RxData muss...
... die Eigenschaft *Enabled* auf *true* gesetzt sein,
... eine *PDO Nummer* eingetragen sein, mit Hilfe derer ein *RxProcessDataObject* referenziert wird und
... die *ID* muss mit der *ID* der zu empfangenen Publisher Variablen übereinstimmen.

Variablen erzeugen

Das Kommando [Variables Configuration] aus dem Kontextmenü des EAP-Geräteobjektes öffnet einen Dialog (siehe Abbildung), um TxVariablen oder RxVariablen zu erzeugen oder zu bearbeiten. Der Dialog gliedert sich in drei Bereiche.



1. Links oben befinden sich einige Schaltflächen, um Variablen zu erzeugen oder zu bearbeiten.
2. Der rechte Teil besteht aus einer Liste, die die bereits vorhandenen Variablen anzeigt.
3. Links unten stehen Bedienelemente bereit, um Anzeigefilter zu aktivieren/deaktivieren, die auf die Variablenliste angewendet werden.

Create Tx Variable (0x6000)/Create Rx Variable (0x7000)

Die Schaltflächen [Create Tx Variable (0x6000)] und [Create Rx Variable (0x7000)] öffnen den Dialog, um eine neue Variable hinzuzufügen. Der Dialog stellt zwei Wege bereit, um eine Variable zu definieren.

- Per DataType

Auf dem Karteireiter *DataType* wird grundsätzlich eine Liste vordefinierter nativer Datentypen angeboten, aus denen einer für die neue Variable ausgewählt werden kann. Sofern für das EAP-Geräteobjekt eine Konfiguration via EDC Datei importiert worden ist, können auch noch weitere komplexe Datentypen in der Liste aufgeführt sein.

Nachdem einer der Datentypen ausgewählt worden ist, werden dessen Eigenschaften in den Eingabefeldern unterhalb der Liste automatisch eingetragen. Allein das Eingabefeld *Data Symbolname* bleibt leer. Hier besteht die Möglichkeit einen Symbolnamen einzutragen, falls er bekannt ist. Der Symbolname muss sich dann auf eine Prozessvariable beziehen, die von einer TwinCAT Anwendung wie beispielsweise einem SPS Projekt kommt. Der Symbolname einer SPS-Prozessvariablen ist ein eindeutiger Bezeichner mit Hilfe dessen auf den Inhalt der Prozessvariablen zugegriffen werden kann. Der Vollständige Symbolname für eine solche Variable muss folgenden Aufbau haben:

[OID]:[vollständiger Symbolname],

wobei die *OID* (Objekt ID) eine 8-stellige hexadezimale eindeutige Nummer ist, die das laufende SPS Projekt innerhalb der Maschinensteuerung identifiziert und der *vollständige Symbolname* der eindeutige Bezeichner der Prozessvariablen innerhalb des SPS Projektes ist (Ein vollständiger Symbolname sieht dann z.B. so aus: 01010010:Main.Status).

Name	Size	GUID
VARIANT	0	00000000-0000-0000-0000-000000000000
BOOL	1	18071995-0000-0000-0000-000000000030
BYTE	1	18071995-0000-0000-0000-000000000001
SINT	1	18071995-0000-0000-0000-000000000003
INT	2	18071995-0000-0000-0000-000000000006
INT24	3	18071995-0000-0000-0000-000000000023
DINT	4	18071995-0000-0000-0000-000000000009
INT40	5	18071995-0000-0000-0000-000000000027
INT48	6	18071995-0000-0000-0000-000000000029

Index: 0x6001

Name: DINT

Data Symbolname:

Size (Bytes): 4

Guid: 18071995-0000-0000-0000-000000000009

Create Cancel

- Per Symbol

Auf dem Karteireiter *Symbols* wird eine Liste von Symbolnamen angeboten. Es werden nur dann Symbole angezeigt, wenn für das EAP-Geräteobjekt eine Konfiguration via EDC Datei importiert worden ist und diese Datei auch Symbolnamen eines SPS Projektes enthält.

Nachdem einer der Symbolnamen ausgewählt worden ist, werden alle Eigenschaften der entsprechenden Prozessvariable in den Eingabefeldern unterhalb der Liste automatisch eingetragen; in diesem Fall dann auch der vollständige Symbolname.

Name	Size	DataType
01010010:MAIN		
in_Req	1	BOOL
out_Ack	1	BOOL
out_Measure	4	UDINT
out_Productl	4	UDINT

Index: 0x6003
 Name: out_Ack
 Data Symbolname: 01010010:MAIN.out_Ack
 Size (Bytes): 1
 Guid: 18071995-0000-0000-0000-000000000030

Create Cancel

Über die Schaltfläche [Create] wird die Variable schlussendlich erzeugt und findet sich dann in der Liste der vorhandenen Variablen wieder.

Edit

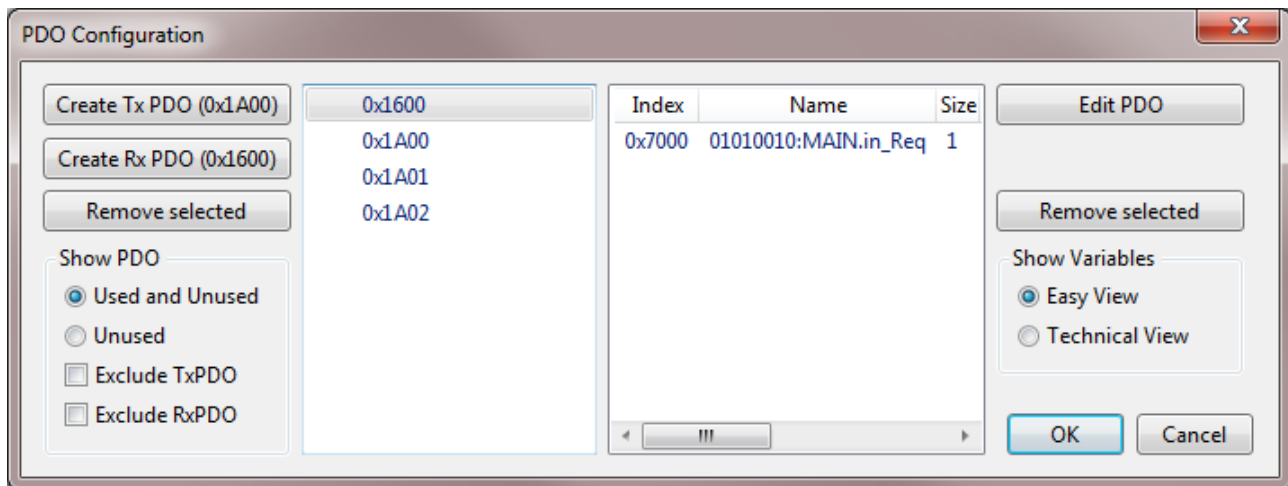
Mit Hilfe der Schaltfläche [Edit] kann eine bestehende Variable bearbeitet werden. Die zu bearbeitende Variable muss dafür vorab selektiert werden. Das Bearbeiten ist hilfreich, wenn etwa ein anderer Symbolname eingetragen werden soll, um die Verknüpfung zu einer anderen SPS Variablen herzustellen.

Remove Selected

Die Schaltfläche [Remove Selected] sorgt dafür, dass die selektierte Variable gelöscht wird.

Process Data Objects erzeugen

Das Kommando [Process Data Configuration] aus dem Kontextmenü des EAP-Geräteobjektes öffnet einen Dialog (siehe Abbildung), um Tx- oder RxPDOs zu erzeugen oder zu bearbeiten. Der Dialog teilt sich methodisch in zwei Bereiche auf.



Auf der linken Hälfte werden in einer Liste alle bereits vorhandenen PDOs angezeigt. An dieser Liste angrenzend befinden sich linksseitig Schaltflächen, um neue PDOs zu erzeugen oder um PDOs zu löschen. Des Weiteren befinden sich darunter angeordnet Bedienelemente, um Anzeigefilter zu aktivieren/deaktivieren, die auf die PDO-Liste angewendet werden.

Auf der rechten Hälfte werden in einer Liste die Variablen angezeigt, die dem PDO zugeordnet sind, welches in der linken Hälfte selektiert ist. Rechts außen befinden sich die Schaltflächen, mit Hilfe derer die Variablenliste bearbeitet werden kann. Über die Auswahlknöpfe [Easy View] und [Technical View] lässt sich die Darstellung der Variablenliste umschalten.

Create Tx PDO (0x1A00)/Create Rx PDO (0x1600)

Die Schaltflächen [Create Tx PDO (0x1A00)] und [Create Rx PDO (0x1600)] erzeugen *TxPDO* bzw. *RxPDO* Instanzen, die direkt in der PDO-Liste angezeigt werden.

Edit PDO

Die Schaltfläche [Edit PDO] öffnet einen Dialog, um Prozessvariablen zu erzeugen und zu konfigurieren (siehe Abbildung oben). Im Kontext der Konfiguration eines PDOs werden mittels dieses Dialogs außerdem die Variablen ausgewählt, die dem selektierten PDO hinzugefügt werden sollen. Jede Variable, deren Checkbox angehakt wird, wird der Variablenliste des selektierten PDOs hinzugefügt, nachdem der Dialog mit [OK] bestätigt worden ist.

Remove Selected

Die Schaltfläche [Remove Selected] auf der linken Hälfte sorgt dafür, dass das selektierte PDO gelöscht wird.

Die Schaltfläche [Remove Selected] auf der rechten Hälfte sorgt dafür, dass die selektierte Variable aus der PDO Definition entfernt wird.

Show Variables – Easy View / Technical View

Die Darstellung der Variablenliste lässt sich umschalten. In der [Easy View]-Darstellung werden alle dem PDO zugeordneten Variablen untereinander aufgelistet. In der [Technical View]-Darstellung wird eine codierte Darstellung angezeigt, so wie sie im Objektverzeichnis des TwinCAT EAP-Gerätes gespeichert wird.

Das Properties-Fenster

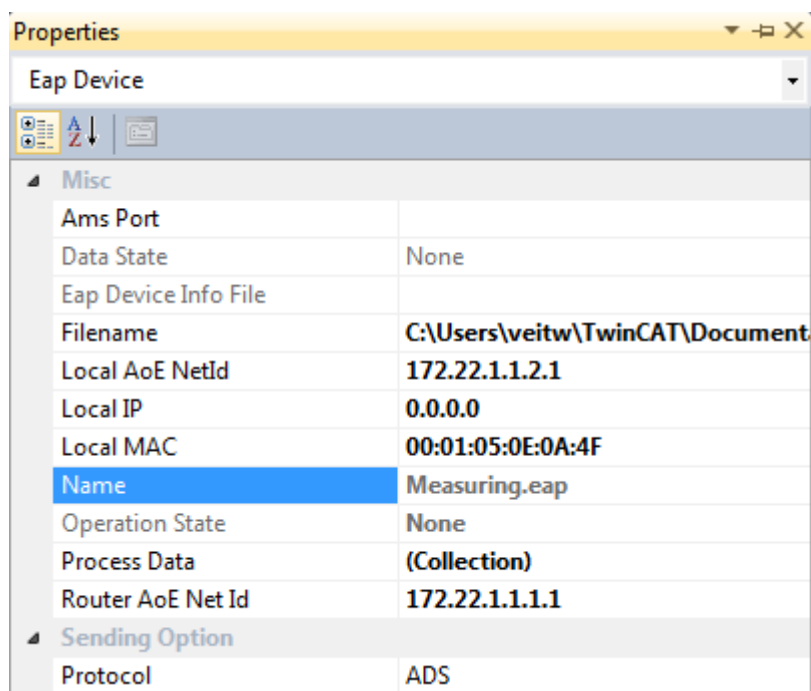
Mit Hilfe des Properties-Fensters können die wichtigsten Konfigurationseigenschaften des aktuell selektierten Elements aus der Bedienoberfläche definiert werden. Es gibt zu jedem Elementtyp eine individuelle Liste von Eigenschaften. Allen Listen gemein ist die Eigenschaft *Data State*. Der *Data State* dient nur der Information und kann im Properties-Fenster nicht verändert werden. Er zeigt den aktuellen Konfigurationszustand des betreffenden EAP Geräteobjektes an.

- None
Es wurden keine Veränderungen an der Konfiguration vorgenommen.

- Changed
Es wurde mindestens eine Veränderung an der Konfiguration vorgenommen.
- Written
Die aktuelle Konfiguration wurde erfolgreich an das TwinCAT EAP Gerät übertragen.
- Error
Es ist ein Fehler aufgetreten. (Z.B. bei der Übertragung der Konfiguration zum TwinCAT EAP Gerät.)

Das EAP-Geräteobjekt

Beim EAP-Geräteobjekt können die folgenden Eigenschaften über das Properties-Fenster überprüft bzw. verändert werden (siehe Abbildung)



AMS Port

Diese Eigenschaft ist reserviert und hat noch keine Funktion.

EAP Device Info File

Diese Eigenschaft ist reserviert und hat noch keine Funktion.

Filename

Diese Eigenschaft dient nur der Information, welche Datei inklusive vollständigen Speicherpfad zu diesem EAP-Geräteobjekt gehört. Diese Datei wird beim Öffnen der Bedienoberfläche geladen, und in dieser Datei werden die Konfigurationsänderungen auch wieder gespeichert.

Local AoE Net ID

Bei der *Local AoE Net ID* muss die *AMS NetID* des TwinCAT EAP-Gerätes eingetragen werden. Sie ist notwendig für die Kommunikation mit dem gewünschten TwinCAT EAP-Gerät.

Diese Adresse wird für die Kommunikation verwendet, die zwischen dem EAP Configurator und dem TwinCAT EAP-Gerät stattfindet. Die Kommunikation ist notwendig, um einerseits die aktive Konfiguration eines TwinCAT EAP-Gerätes zu lesen und andererseits eine neue Konfiguration vom TwinCAT EAP Configurator in das TwinCAT EAP-Gerät zu schreiben. Wenn keine gültige *NetID* eingetragen ist, kann auch keine Kommunikation mit einem TwinCAT EAP-Gerät stattfinden.

Die *Local AoE Net ID* ist in den ersten vier Stellen immer identisch zur *Router AoE Net ID*. Die letzten beiden Stellen der *Local AoE Net ID* können niemals .1.1 sein, da diese für die *AMS NetID* des PC selbst reserviert sind (vgl. *Router AoE Net ID*).

Local IP

Die *Locale IP* wird beim Lesen einer aktiven EAP-Konfiguration gesetzt. Sie kann auch manuell gesetzt bzw. verändert werden. Die IP wird zum einen verwendet, wenn der TwinCAT EAP Configurator per *AoE* über *UDP/IP* kommuniziert. Zum anderen benötigt der TwinCAT EAP Configurator diese Information, um zu ermitteln, welches EAP-Geräteobjekt von einem konfigurierten *TxFrame* mit einer IP-Adresse als Zieladresse erreicht werden kann.

Local MAC

Die *Local MAC* wird – wie die *Local IP* – ebenfalls beim Lesen einer aktiven EAP-Konfiguration gesetzt. Auch sie kann manuell gesetzt bzw. verändert werden. Der TwinCAT EAP Configurator benötigt diese Information, um zu ermitteln, welches EAP-Geräteobjekt von einem konfigurierten *TxFrame* mit einer MAC-Adresse als Zieladresse erreicht werden kann.

Name

Diese Eigenschaft dient nur der Information, welche Datei zu diesem EAP-Geräteobjekt gehört.

Operation State

Der *Operation State* zeigt an, welche Operation aktuell vom EAP Configurator bei diesem Geräteobjekt durchgeführt wird. Mögliche Zustände sind

- *Read* (während eine aktive Konfiguration von einem TwinCAT EAP-Gerät gelesen wird) oder
- *Write* (während eine neue Konfiguration in ein TwinCAT EAP-Gerät geschrieben wird).

Process Data

Mit Hilfe der Eigenschaft *Process Data* können *PDO*-Elemente erzeugt werden. Bei der Auswahl dieser Eigenschaft wird eine Schaltfläche sichtbar, die angeklickt werden kann, um den entsprechenden Konfigurationsdialog zu öffnen (vgl. Kapitel [Process Data Objects erzeugen](#) [► 51]).

Router AoE Net ID

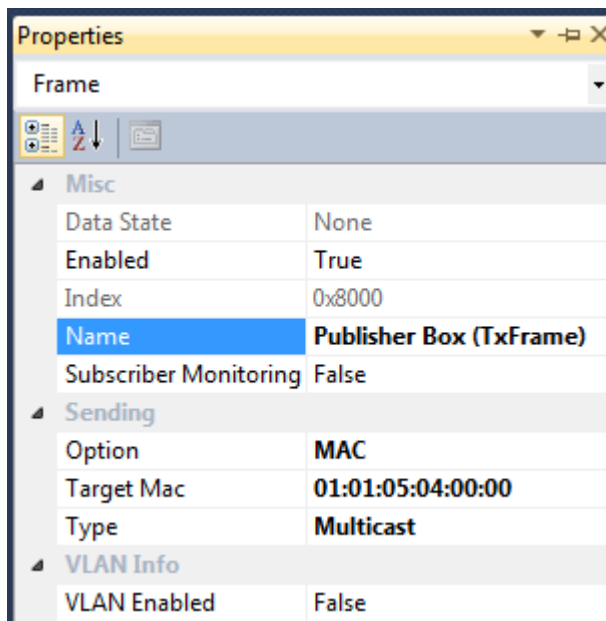
Die *Router AoE Net ID* wird beim Lesen einer aktiven EAP-Konfiguration gesetzt. Sie dient in erster Linie nur der Information. Die *Router AoE Net ID* und die *Local AoE Net ID* sind in den ersten vier Stellen identisch. Die letzten beiden Stellen der *Router AoE Net ID* haben immer die Kennung .1.1.

Protocol

Das Protocol definiert das Kommunikationsprotokoll, welches zur Übertragung der Konfigurationsdaten vom bzw. zum TwinCAT EAP-Gerät verwendet wird. Zur Auswahl stehen *ADS* und *UDP*. Dabei bedeutet *UDP*, dass automatisch das Protokoll *AoE* per *UDP/IP* verwendet wird (vgl. [Das Funktionsprinzip des TwinCAT EAP Configurators](#) [► 9]).

Der Frame

Beim Frame können die folgenden Eigenschaften über das Properties-Fenster überprüft bzw. verändert werden (vgl. Abbildung).



Enabled

Die Eigenschaft *Enabled* dient dazu, den Frame zu aktivieren bzw. zu deaktivieren.

Index

Der *Index* wird automatisch angelegt und ist nur zu Informationszwecken in der Liste der Eigenschaften mit aufgeführt. Er indiziert das Frameobjekt im EAP-Objektverzeichnis und darf nicht manuell verändert werden.

Name

Die Eigenschaft *Name* wird ebenfalls automatisch definiert, kann aber nach Belieben verändert werden. Die maximale Länge ist dabei auf 255 Zeichen begrenzt.

Subscriber Monitoring

Die Eigenschaft *Subscriber Monitoring* kann ein- oder ausgeschaltet werden. Je nachdem, welche Auswahl bei der Eigenschaft *Option* getroffen wird (siehe unten), ist das *Subscriber Monitoring* standardmäßig aktiviert oder deaktiviert. Das *Subscriber Monitoring* dient der Überwachung, ob ein Empfänger im Netzwerk noch erreichbar ist. Diese Überwachung ist jedoch nur dann sinnvoll und möglich, wenn der Frame als *Unicast*, also an einen Empfänger, adressiert ist.

Option

Mit Hilfe dieser Eigenschaft wird die Art der Adressierung des Frames festgelegt. Es stehen drei Adressierungsarten zur Auswahl. Voreingestellt ist die Adressierung per *AMS NetID*, die nur einen *Unicast* ermöglicht. Des Weiteren gibt es noch die Wahl zwischen *MAC* oder *UDP/IP*. Beide Arten ermöglichen einen *Unicast*, sowie auch einen *Broadcast* oder einen *Multicast* (vgl. [Kommunikationsmethoden](#) [► 12]).



Das ADS/AMS Protokoll unterstützt nur Unicast Kommunikation

Die *Automation Message Specification (AMS)* sieht keine speziellen Adressen in Form einer *Broadcast* oder *Multicast AMS NetID* vor, so dass bei einer Adressierung per *AMS NetID* ausschließlich *Unicast* Kommunikation möglich ist.

Target (AMS Net ID / MAC / IP)

Bei der Eigenschaft *Target (AMS Net ID / MAC / IP)* wird die Zieladresse definiert. Die *AMS NetID* ist eine 6 Byte lange Adresse. Die Bytes werden mit einem Punkt getrennt. Die MAC-Adresse ist ebenfalls 6 Byte lang, getrennt durch Doppelpunkte. Die *IP* ist eine 4 Byte lange Adresse, getrennt durch Punkte. Sowohl bei der Adressierungsart *MAC*, als auch bei *IP* kann über das zusätzlich eingeblendete Eigenschaftsfeld *Type* festgelegt werden, ob es ein *Unicast*, *Multicast* oder *Broadcast* sein soll.

Die Art der konfigurierten Adressierung legt fest, welches Transportprotokoll verwendet wird, um die EAP-Nachrichten zu versenden (vgl. [Kommunikationsmethoden](#) [► 12]).

Type

Legt die Verbindungsart eines *MAC*-adressierten oder eines *IP*-adressierten Frames fest (vgl. [Kommunikationsmethoden \[► 12\]](#)).

- **Unicast:**
Bei der *Target MAC* bzw. *Target IP* wird die Adresse eines bestimmten Empfängers eingetragen.
- **Multicast:**
Bei der *Target MAC* wird automatisch die TwinCAT *Multicast MAC*-Adresse für Netzwerkvariablen (01:01:05:04:00:00) eingetragen. Es kann jedoch auch eine beliebige andere multicast-gültige *MAC*-Adresse eingetragen werden.
Bei der *Target IP* wird automatisch eine *Multicast IP*-Adresse (nämlich 224.0.0.0) eingetragen. Es kann jedoch auch eine beliebige andere multicast-gültige *IP*-Adresse eingetragen werden.
- **Broadcast:**
Bei der *Target MAC* wird die *Broadcast MAC* FF:FF:FF:FF:FF:FF automatisch eingetragen.
Bei der *Target IP* wird die *Broadcast IP* 255.255.255.255 automatisch eingetragen.

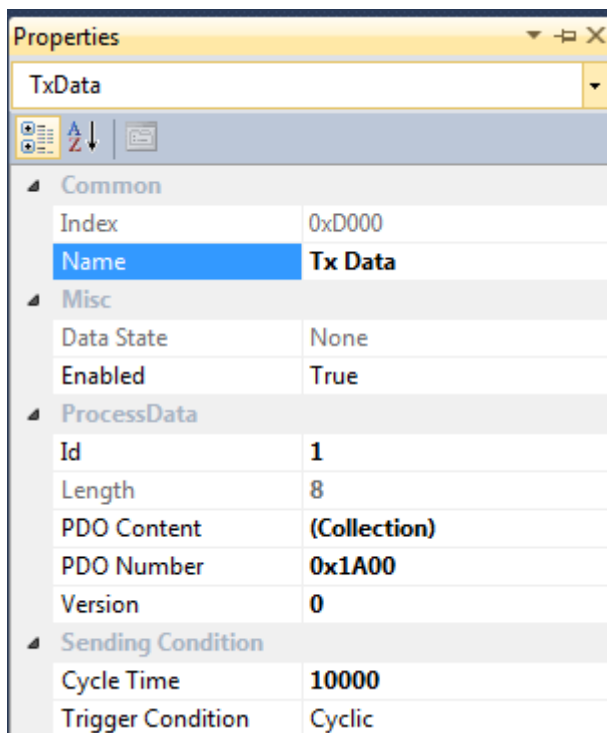
VLAN Enabled

Die Eigenschaft *VLAN Enabled* erlaubt es, einen Frame für ein Virtual LAN zu konfigurieren. Die Eigenschaft kann ein- oder ausgeschaltet werden. Bei eingeschaltetem *VLAN* wird die EAP-Nachricht mit einem *VLAN*-Header versehen. Entsprechend gibt es zwei weitere Eigenschaften, um das gewünschte *VLAN* zu bestimmen und eine Priorität für die Verarbeitung der Nachricht innerhalb des virtuellen Netzwerks festzulegen:

- *VLAN Info ID*: Definiert die ID des *VLANs* (Wertebereich von 0 bis 4095), in welches die Nachricht geschickt werden soll und
- *VLAN Info Priority*: Definiert eine Priorität für die Nachricht im *VLAN* (hoch Prior = 7, nieder Prior = 0).

Das TxData

Beim *TxData* können die folgenden Eigenschaften über das Properties-Fenster überprüft bzw. verändert werden (vgl. Abbildung).



Index

Der *Index* wird automatisch angelegt und ist nur zu Informationszwecken in der Liste der Eigenschaften mit aufgeführt. Er indiziert das *TxData* Objekt im EAP-Objektverzeichnis und darf nicht manuell verändert werden.

Name

Die Eigenschaft *Name* wird ebenfalls automatisch definiert, kann aber nach Belieben verändert werden. Die maximale Länge ist dabei auf 255 Zeichen begrenzt.

Enabled

Die Eigenschaft *Enabled* dient dazu, das *TxData* zu aktivieren bzw. zu deaktivieren.

ID

Mit Hilfe der Eigenschaft *ID* soll dem *TxData* eine netzwerkweit eindeutige Identifikationsnummer aus dem Wertebereich von 0 bis 65535 zugeordnet werden. Die *ID* wird bei jedem Anlegen eines neuen *TxData* automatisch vergeben. Ein *RxData* eines Subscribers, welches die Daten dieses *TxData* empfangen soll, muss die gleiche *ID* eingestellt haben.

Length

Die Eigenschaft *Length* ist zu Informationszwecken in der Liste enthalten. Sie zeigt an, wie viele Bytes der Datensatz des referenzierten *PDOs* (siehe *PDO Number*) umfasst. Wenn kein *PDO* referenziert ist, wird auch keine Größe angegeben. Die *Length* des *TxData* und des dazugehörigen *RxData* beim Empfänger (Subscriber) muss identisch sein. Andernfalls werden die Daten nicht empfangen.

PDO Content

Mit Hilfe der Eigenschaft *PDO Content* können *TxPDO*-Elemente erzeugt werden. Bei der Auswahl dieser Eigenschaft wird eine Schaltfläche sichtbar, die angeklickt werden kann, um den entsprechenden Konfigurationsdialog zu öffnen (vgl. Kapitel [Process Data Objects erzeugen](#) [► 51]).

PDO Number

Mit Hilfe der Eigenschaft *PDO Number* wird festgelegt, welches *PDO* von dem *TxData* referenziert wird. Eine Auswahlbox zeigt alle verfügbaren *TxPDOs* an.

Version

Die Eigenschaft *Version* ist standardmäßig auf den Wert 0 voreingestellt. Die *Version* dient dazu, dem *TxData* eine Versionsnummer zu geben. Die Versionsnummer sollte vom Anwender genau dann erhöht werden, wenn er die Konfiguration des *TxData* verändert, so dass sich die Struktur der Nutzdaten zur vorherigen Version unterscheidet. So kann der Subscriber beim Empfang anhand der Versionsnummer erkennen, dass die empfangenen Nutzdaten nicht mehr zu der Datenstruktur passen, die er erwartet. Denn im Normalbetrieb vergleicht er die beim *RxData* definierte Versionsnummer mit der vom Publisher gesendeten Versionsnummer. Bei Nichtübereinstimmung werden die Daten verworfen und somit nicht empfangen. Auf diese Weise kann vermieden werden, dass bei einer einseitigen Änderung der EAP-Variablen ungültige Daten empfangen werden. Eine einseitige Änderung liegt dann vor, wenn eine Änderung der Konfiguration entweder nur am *TxData* des Publishers oder nur am *RxData* des Subscribers vorgenommen wird.

Trigger Condition

Für ein *TxData* wird einer von drei Mechanismen eingestellt, der definiert, wie das Versenden der Daten ausgelöst werden soll (Auslöser = engl. *Trigger*). Die vordefinierte *Trigger Condition* ist die *Cycle Time*. Die Daten werden dann zyklisch in einem definierten Zeitintervall gesendet. Das Zeitintervall wird unter der Eigenschaft *Cycle Time* definiert.

Des Weiteren kann der Mechanismus *Change of State (CoS)* gewählt werden. Die Daten werden in diesem Fall nur dann gesendet, wenn sie sich seit dem letzten Taskzyklus verändert haben. Bei diesem Mechanismus werden mit Hilfe der Eigenschaften *CoS Change Timeout Time* und *CoS Inhibit Time* zwei Wartezeiten definiert, um den Mechanismus zu steuern (vgl. Kapitel [Der EAP-Sendemechanismus](#) [► 15]).

Letztlich gibt es noch den Mechanismus *Polled*. Dabei muss die Eigenschaft *Poll Request Rx PD* definiert werden. In dieser Konfiguration nimmt das *TxData* die Rolle eines Clients in einem Client/Server Modell ein. Das *TxData* wird als Anfrage (Request) an einen Server gesendet. Die Eigenschaft *Poll Request Rx PD* definiert dann den Empfangspuffer für die erwartete Antwort vom Server.

Cycle Time

Die Eigenschaft *Cycle Time* (Zykluszeit) definiert das zeitliche Intervall, mit dem die Daten versendet werden, in Mikrosekunden [μs]. Wenn eine Zeit von 10000 μs eingetragen worden ist, werden die Daten alle 10000 μs = 10 ms = 0,01 s versendet. Die kleinstmögliche Zykluszeit richtet sich dabei nach der Zykluszeit der Task, von der das EAP-Gerät getrieben wird. Daraus ergeben sich dann auch die zulässigen Zykluszeiten, die immer ein ganzzahliges Vielfaches von der Taskzykluszeit sein sollten.

CoS Change Timeout Time

Mit Hilfe dieser Eigenschaft wird beim Mechanismus *Change of State* eine maximale Wartezeit definiert. Sie legt fest, in welchem Zeitintervall das *TxData* gesendet wird, wenn sich die Daten der Prozessvariablen, die diesem *TxData* zugeordnet sind, innerhalb des Zeitintervalls nicht verändert haben.

CoS Inhibit Time

Mit Hilfe dieser Eigenschaft wird beim Mechanismus *Change of State* eine minimale Wartezeit definiert. Sie legt eine Dauer fest, wie lange das Senden des *TxData* unterbrochen wird, obwohl sich die Daten der Prozessvariablen, die diesem *TxData* zugeordnet sind, innerhalb dieser Dauer verändert haben. Per Definition kann die *Inhibit Time* (minimale Wartezeit) niemals größer als die *Change Timeout Time* (maximale Wartezeit) sein.

Poll Request Rx PD

Mit Hilfe dieser Eigenschaft wird beim Mechanismus *Polled* der Index desjenigen *RxData* definiert, welches als Empfangspuffer dienen soll.

Das RxData

Properties	
RxData	
Common	
Filter Ams Net Id	0.0.0.0.0
Index	0xE000
Multicast Ip Address	0.0.0.0
Multicast Mac	01:01:05:04:00:00
Name	Rx Data
Misc	
Data State	None
Enabled	True
Ignore Version	False
Poll Request Tx PD	0x0
ProcessData	
Id	1
Length	8
PDO Content	(Collection)
PDONumber	0x1600
Version	0

Filter AMS Net ID

Die *Filter AMS Net ID* dient dazu, beim *RxData* zu konfigurieren, dass der Empfang einer EAP-Variablen nur dann zugelassen wird, wenn diese von dem EAP-Gerät gesendet worden ist, welches die definierte *AMS Net ID* besitzt.

Index

Der *Index* wird automatisch angelegt und ist nur zu Informationszwecken in der Liste der Eigenschaften mit aufgeführt. Er indiziert das *RxData* Objekt im EAP-Objektverzeichnis und darf nicht manuell verändert werden.

Multicast MAC Address

Die Eigenschaft *Multicast MAC Address* kann dazu verwendet werden, eine Multicast MAC Adresse beim EAP-Gerät einzurichten (vgl. Kapitel [Kommunikationsmethoden](#) [► 12]).



Multicast MAC Adresse

Bei TwinCAT ist für ein EAP-Gerät generell die *Multicast MAC Adresse* 01:01:05:04:00:00 voreingestellt. Weitere Adressen können nur mit Hilfe von TwinCAT 3 bei Inbetriebnahme des TwinCAT Systems konfiguriert werden. Es besteht keine Möglichkeit, zusätzlich Multicast Adressen bei einem TwinCAT EAP-Gerät mit Hilfe des TwinCAT EAP Configurators zu konfigurieren, da dieser die Konfiguration am EAP-Gerät vornimmt, nachdem das TwinCAT System bereits in Betrieb gesetzt worden ist (Run Mode).

Multicast IP Address

Die Eigenschaft *Multicast IP Address* kann dazu verwendet werden, eine Multicast IP Adresse beim EAP-Gerät einzurichten. Bei dieser Eigenschaft gelten die gleichen Einschränkungen wie für die *Multicast MAC Address*. Multicast IP Adressen sind nur dann notwendig, wenn die Kommunikation über die Subnetzgrenze hinaus, also per Router in andere Subnetze erfolgen soll.



Multicast IP Adresse

Ein EAP-Publisher, bei dem in TwinCAT 3 für die Zieladresse eine *Multicast IP Adresse* konfiguriert wird, bekommt voreingestellt die *Multicast IP Adresse* 224.0.0.0 zugewiesen. Es darf aber auch jede beliebige andere *Multicast IP Adresse* verwendet werden. TwinCAT generiert zu jeder konfigurierten *Multicast IP Adresse* eine konforme *Multicast MAC Adresse*, die bei der Inbetriebsetzung von TwinCAT (also beim Schalten in den Run Mode) verwendet wird.

Auch hier gilt, dass keine Möglichkeit besteht, zusätzliche *Multicast IP Adressen* bei einem TwinCAT EAP-Gerät mit Hilfe des TwinCAT EAP Configurators zu konfigurieren, da dieser die Konfiguration am EAP-Gerät erst dann vornimmt, wenn das TwinCAT System bereits in Betrieb gesetzt worden ist (Run Mode).

Name

Die Eigenschaft *Name* wird ebenfalls automatisch definiert, kann aber nach Belieben verändert werden. Die maximale Länge ist dabei auf 255 Zeichen begrenzt.

Enabled

Die Eigenschaft *Enabled* dient dazu, das *RxData* zu aktivieren bzw. zu deaktivieren.

Ignore Version

Mit Hilfe der Eigenschaft *Ignore Version* kann definiert werden, ob beim Empfang der EAP-Variablen die Version auf Übereinstimmung geprüft werden soll oder nicht. Wenn *Ignore Version* auf den Wert *TRUE* gesetzt wird, wird eine EAP-Variable auch dann empfangen, wenn dessen Versionsnummer ungleich der hier definierten Versionsnummer ist.

Poll Request Tx PD

Die Eigenschaft *Poll Request Tx PD* muss nur dann definiert werden, wenn das *RxData* für den Polled Mechanismus verwendet werden soll. Andernfalls muss der eingetragene Wert 0 bleiben.

Im Polled Mechanismus nimmt das *RxData* die Rolle eines Servers in einem Client/Server Modell ein. Das *RxData* nimmt dann eine Anfrage (Request) eines Clients entgegen. Die Eigenschaft *Poll Request Tx PD* definiert dann, welches *TxData* als Antwort unmittelbar auf den Request gesendet wird. Dazu wird der Index des gewünschten *TxData* definiert.

ID

Mit Hilfe der Eigenschaft *ID* soll dem *RxData* eine netzwerkweit eindeutige Identifikationsnummer aus dem Wertebereich von 0 bis 65535 zugeordnet werden. Die *ID* wird bei jedem Anlegen eines neuen *RxData* automatisch vergeben. Das *RxData*, welches die Daten eines *TxData* empfangen soll, muss die gleiche *ID* eingestellt haben, wie das sendende *TxData*.

Length

Die Eigenschaft *Length* ist zu Informationszwecken in der Liste enthalten. Sie zeigt an, wie viel Bytes der Datensatz des referenzierten *PDOs* (siehe *PDO Number*) umfasst. Wenn kein *PDO* referenziert ist, wird auch keine Größe angegeben. Die *Length* des *RxData* und die Länge des dazugehörigen *TxData* beim Sender (Publisher) müssen identisch sein. Andernfalls werden die Daten vom Empfänger (Subscriber) nicht empfangen.

PDO Content

Mit Hilfe der Eigenschaft *PDO Content* können *RxPDO*-Elemente erzeugt werden. Bei der Auswahl dieser Eigenschaft wird eine Schaltfläche sichtbar, die angeklickt werden kann, um den entsprechenden Konfigurationsdialog zu öffnen (vgl. Kapitel [Process Data Objects erzeugen](#) [► 51]).

PDO Number

Mit Hilfe der Eigenschaft *PDO Number* wird festgelegt, welches *PDO* von dem *RxData* referenziert wird. Eine Auswahlbox zeigt alle verfügbaren *RxPDOs* an.

Version

Die Eigenschaft *Version* ist standardmäßig auf den Wert 0 voreingestellt. Die *Version* dient dazu, dem *RxData* eine Versionsnummer zu geben. Die Versionsnummer sollte vom Anwender genau dann erhöht werden, wenn er die Konfiguration des *RxData* verändert, so dass sich die Struktur der Nutzdaten zur vorherigen Version unterscheidet. So kann der Subscriber beim Empfang anhand der Versionsnummer erkennen, dass die empfangenen Nutzdaten nicht mehr zu der Datenstruktur passen, die er erwartet. Denn im Normalbetrieb vergleicht er diese Versionsnummer mit der definierten Versionsnummer. Bei Nichtübereinstimmung werden die Daten verworfen und somit nicht empfangen. Auf diese Weise kann vermieden werden, dass bei einer einseitigen Änderung der EAP-Variablen ungültige Daten empfangen werden. Eine einseitige Änderung liegt dann vor, wenn eine Änderung der Konfiguration entweder nur am *TxData* des Publishers oder nur am *RxData* des Subscribers vorgenommen wird.

5.2 Ein Projekt speichern/laden

Ein Projekt des TwinCAT EAP Configurators wird über die Standardmechanismen des MS Visual Studios gespeichert und geladen:

1. Wählen Sie den Menüeintrag [File] → [Save All], [File] → [Save As] oder [File] → [Save <Dateiname>].
2. Dabei wird stets das gesamte Projekt gespeichert, das heißt, dass die Änderungen aller EAP Geräteobjekte in ihre jeweiligen .eap Dateien geschrieben werden.

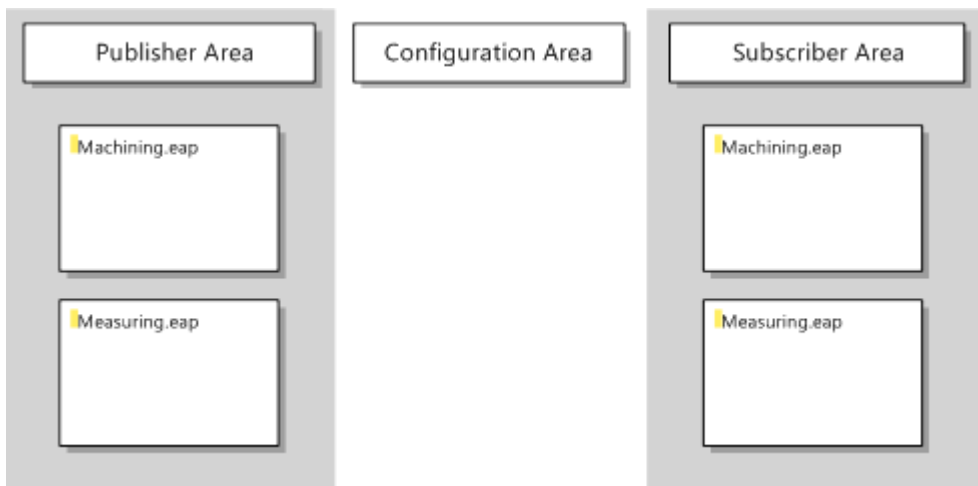
● Symbolinformationen

i Wenn für ein EAP Geräteobjekt eine .EDC Datei mit Symbolinformationen importiert worden ist, werden auch diese mit abgespeichert, so dass nach dem Laden des EAP Projektes alle Informationen wieder zur Verfügung stehen.

Das Laden erfolgt analog zum Speichern:

3. Wählen Sie den Menüeintrag [File] → [Open] → [Project/Solution].
 4. Suchen Sie sich die gewünschte Projektdatei heraus und bestätigen Sie dieses mit der Schaltfläche [Open].
- ⇒ Das Projekt wird geladen.

Nach dem Laden kann ein beliebiger Geräteknoten aus dem Projekt geöffnet werden. Entweder per Doppelklick über die linke Maustaste oder mit Hilfe des Kontextmenüs des Geräteknotens (per Rechtsklick) und dem Anklicken des Kommandos [Open]. Wenn ein Knoten geöffnet wird, werden alle im Projekt befindlichen Geräteknoten geladen und in der sich öffnenden grafischen Bedienoberfläche dargestellt (siehe Kapitel [Das TwinCAT EAP Configurator Projekt](#) [► 36]). Es wird dabei das EAP Geräteobjekt an oberster Stelle angezeigt, dessen Geräteknoten geöffnet worden ist. Ein EAP Geräteobjekt repräsentiert also ein EAP Gerät (vgl. Abbildung).



5.3 Das Objektverzeichnis

Sämtliche Konfigurationsdaten, die zu einem EAP Geräteobjekt im TwinCAT EAP Configurator gehören, werden zusammengefasst mit Hilfe des Objektverzeichnisses abgebildet. Dieses Objektverzeichnis wird in tabellarischer Form angezeigt, wenn das Kommando [Show in Tool Window] aus dem Kontextmenü zum EAP Geräteobjekt angeklickt wird (vgl. Abbildung). Eine Änderung der Konfiguration in der grafischen Bedienoberfläche verursacht automatisch eine Modifikation des Objektverzeichnisses. Alle Modifikationen werden dabei mit einer gelben Markierung gekennzeichnet. Es ist auch möglich, im Objektverzeichnis die Konfiguration manuell zu verändern. Dies wird jedoch nicht empfohlen und sollte nur durchgeführt werden, wenn die Auswirkungen der Änderung genau bekannt sind.

The screenshot displays the TwinCAT EAP Configurator interface. The top window, titled 'Machining.eap', shows a graphical representation of the EAP configuration. It includes a 'Box 1 (Publisher)' containing a 'TxPD' (Transmit Process Data) block, and three 'Rx Data' blocks, each associated with an 'RxPD' (Receive Process Data) block. Green arrows indicate data flow between these components.

The bottom window, titled 'EAP Config Tool Window #1', shows the 'CoE - Online of Machining.eap' configuration. It contains a table listing various EAP objects with their indices, names, flags, and values.

Index	Name	Flags	Value
0x1000	Device Type	RO	65541002
0x1008	Device Name	RO	EtherCAT Automation Protocol
0x100A	Software Version	RO	0
0x1018	Identity	RO	< 4 >
0x1600	Rx PDO	RW	< 1 >
0xE008	Rx Process Data	RW	< 31 >
0xF020	Tx Frame List 1	RW	< 1 >
0xF021	Tx Frame List 1	RW	< 0 >
0xF022	Tx Frame List 1	RW	< 0 >
0xF100	EAP Status	RO	< 2 >
0xF200	EAP Control	RW	< 1 >
0xF800	Configured Objects	RW	< 8 >
0xF801	Index Bitmaps	RW	< 6 >
0xF920	AoE Settings	RW	< 5 >
0	Number of Elements	RO	5
1	AoE Local NetID	RW	05 14 29 9F 02 01
2	AoE Router NetID	RO	05 14 29 9F 01 01
3	Local MAC Address	RO	00 01 05 14 29 9F
4	Local IP Address	RO	0
5	Local Port Name	RW	Device 1 (EtherCAT Automation

● Read Only

i Alle Einträge des Objektverzeichnisses, die als *Read Only* (RO) markiert sind, dürfen nicht verändert werden. Außerdem dürfen auf keinen Fall die Objekte mit dem Index 0xF800 und mit dem Index 0xF801 manuell verändert werden. Diese beiden Objekte werden modifiziert, wenn neue EAP-Elemente, wie *TxFrames*, *Tx/RxData*, *Tx/RxPDOs* oder *Tx/RxVariablen* hinzugefügt werden.

HINWEIS**Device Cycle Time**

Eine besonders wichtige Eigenschaft findet sich im Objekt 0xF800 (*Configured Objects*). Der Eintrag Nummer 8 mit der Bezeichnung *Device Cycle Time* enthält die Taskzykluszeit in μs , mit der das EAP Gerät getrieben wird. Die Zykluszeit, mit der eine Variable von diesem EAP Gerät versendet werden kann, muss ein ganzzahliges Vielfaches von dieser Zykluszeit sein.

Der Aufbau des Objektverzeichnisses und die Bedeutung der einzelnen Objekteinträge werden in der Dokumentation zum TwinCAT EAP Gerät in dem Kapitel *Das CANopen Objektverzeichnis* detailliert beschrieben.

6 Anhang

6.1 Anlegen der Beispielprojekte

Das TwinCAT-Project Measuring

Gehen Sie folgendermaßen vor, um das Measuring Projekt in TwinCAT anzulegen:

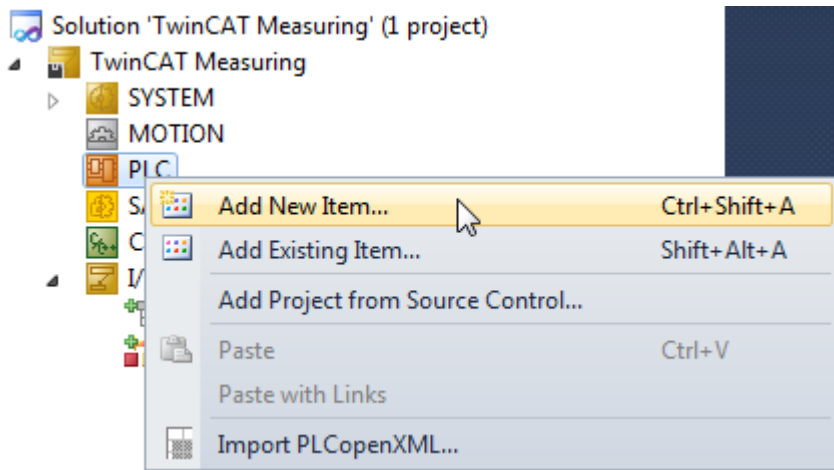
1. Starten Sie das Microsoft Visual Studio.
2. Wählen Sie das Menükommando [File] → [New] → [Project...] aus.
3. Wählen Sie im *New Project* Dialog aus der Liste *Installed Templates* das *TwinCAT Project* aus.
4. Definieren Sie einen Namen für das neue Projekt (z. B. *Measuring*)
5. Bestätigen Sie den Dialog mit der Schaltfläche [OK].

⇒ Ein leeres TwinCAT Projekt wird erzeugt und angezeigt.

In dem neuen Projekt müssen nun das EAP-Gerät und ein SPS-Projekt angelegt werden.

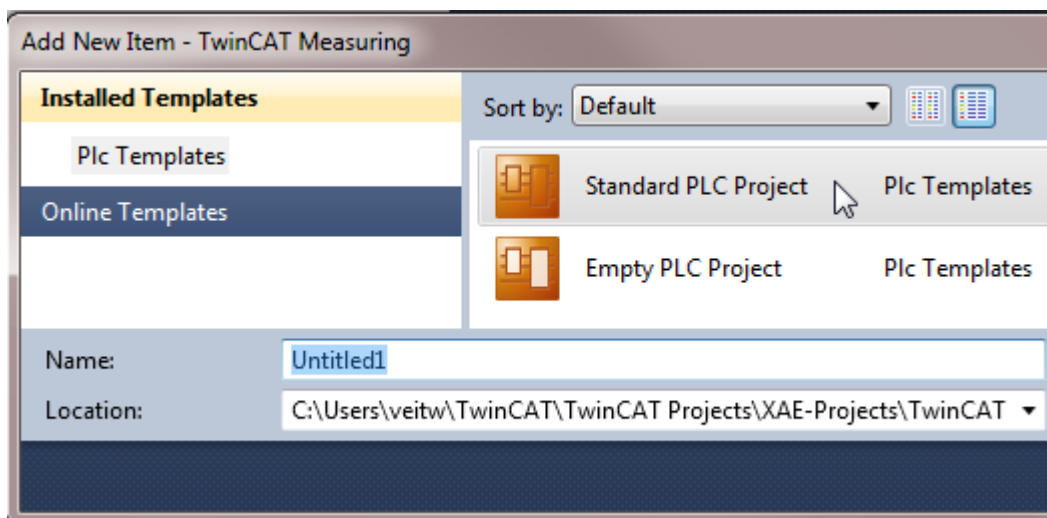
PLC Project Measuring

Das Schreiben eines SPS-Programms erfolgt innerhalb eines TwinCAT SPS-Projekts. Legen Sie also ein SPS-Projekt anhand der folgenden Schritte an:



1. Klicken Sie mit der rechten Maustaste auf den Knoten *PLC* in Ihrem TwinCAT-Projekt und wählen Sie das Kommando [Add New Item...] aus (vgl. Abbildung oben).

⇒ Es öffnet sich der *Add New Item* Dialog.

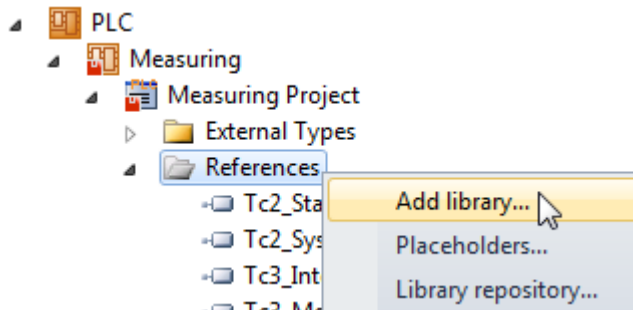


1. Wählen Sie das Template *Standard PLC Project* aus, definieren Sie einen Namen (in diesem Fall am besten *Measuring*) und schließen Sie den Dialog über die Schaltfläche [Add].

⇒ Es wird ein leeres Standard PLC Project erzeugt.

Ein Standard PLC Project referenziert bereits auf ein paar grundlegende SPS-Bibliotheken. Für das kommende Beispielprogramm wird noch die Bibliothek *Tc2_Uilities* benötigt. Fügen Sie eine Referenz auf diese Bibliothek folgendermaßen hinzu:

1. Klicken Sie mit der rechten Maustaste auf den Knoten *References* im PLC Project-Knoten (vgl. Abbildung).



1. Wählen Sie das Kommando [Add Library...]
⇒ Es öffnet sich der Dialog *Add Library*.
2. Klappen Sie den Knoten *Systems* auf und selektieren Sie die SPS-Bibliothek mit dem Namen *Tc2_Uilities*.
3. Anschließend bestätigen Sie den Dialog mit [OK].
⇒ Die gewählte SPS-Bibliothek wird in der Liste der References angezeigt.

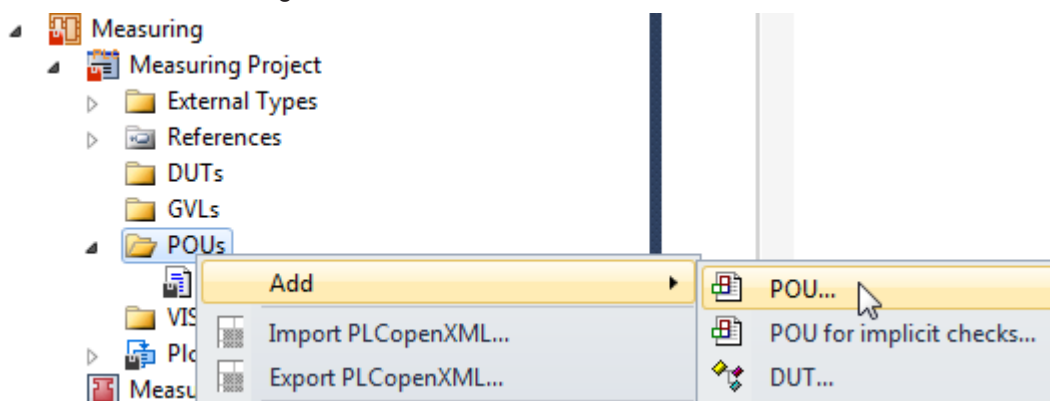
Das Measuring SPS-Beispielprogramm

Das Measuring Beispielprogramm soll auf einfachste Art einen Messvorgang bei einem Produkt oder Werkstück mit einer Identifikationsnummer simulieren und diese Daten später per EAP einer anderen Steuerung zur Verfügung stellen.

Für diese Aufgabe wird ein Funktionsbaustein (FB) mit der Bezeichnung *GetMeasure* erzeugt, der zufällig unterschiedliche Messwerte liefert und ein Hauptprogramm (*MAIN*) geschrieben, welches die Funktion übernimmt, auf Anfrage die ermittelten Daten zur Verfügung zu stellen.

Funktionsbaustein *GetMeasure*

1. Klicken Sie mit der rechten Maustaste auf den Knoten *POUs* im PLC Project (siehe Abbildung).
⇒ Es öffnet sich der Dialog *Add POU*.



1. Tragen Sie als Namen *GetMeasure* ein, wählen Sie den *Type: Function Block* und für die *Implementation Language: Structured Text (ST)*.
⇒ Nach der Bestätigung mit [Open] wird der neue Funktionsbaustein im Editor angezeigt und die Logik kann implementiert werden.

Der FB soll folgende Eingabeparameter haben:

- **NominalMeasure**
Ein Nennwert der bei dem Produkt normalerweise erwartet wird.
- **Variance**
Ein Wert für die Abweichung. Die Messwertsimulation soll später einen Wert aus dem Intervall [Nennwert – 0,5 * Variance, Nennwert + 0,5 * Variance] liefern.
- **Steps**
Steps legt die Schrittweite fest, mit der die Abweichung auftreten kann.

Aus den Eingabeparametern (NominalMeasure = 5000, Variance = 1000, Steps = 100) soll also zufällig ein Messwert aus [4500, 4600, 4700, ..., 5400, 5500] erzeugt werden, der als Ausgabeparameter vom FB zurückgeliefert wird. Für den FB GetMeasure kann der Programmcode dann folgendermaßen aussehen:

```
FUNCTION_BLOCK PUBLIC GetMeasure

VAR_INPUT
    nominalMeasure : UDINT;
    variance : UDINT;
    steps : UINT;
END_VAR

VAR_OUTPUT
    measure : UDINT;
END_VAR

VAR
    FB_SystemTime : GETSYSTEMTIME;
    currTimeLo : UDINT;
    currTimeHi : UDINT;
    FB_Randomizer : DRAND;
```

```

rand : LREAL;
seed : INT;

varianceEffective: UDINT;
varianceAbsolute : UDINT;
deviation : UDINT;
END_VAR

FB_SystemTime( timeLoDW => currTimeLo, timeHiDW => currTimeHi );
seed := UDINT_TO_INT(currTimeLo);
FB_Randomizer(Seed := seed, Num => rand);
varianceEffective := variance / steps;

varianceAbsolute := LREAL_TO_UINT(rand * UDINT_TO_LREAL(varianceEffective)) * steps;

IF varianceAbsolute >= (variance / 2) THEN
    deviation := varianceAbsolute - (variance / 2);
    measure := nominalMeasure + deviation;
ELSE
    deviation := (variance / 2) - varianceAbsolute;
    measure := nominalMeasure - deviation;
END_IF;

```

Kopieren Sie den Programmcode in den FB *GetMeasure*. Anschließend öffnen Sie das Programm *MAIN* per Doppelklick im Editor. Hier wird nun das Hauptprogramm kodiert. Im Hauptprogramm werden eine Eingangsvariable und drei Ausgangsvariablen definiert.

Das Hauptprogramm MAIN:

Eingangsvariable:

- **in_Req** vom Typ **BOOL**
wird verwendet, um eine Anfrage der anderen Steuerung zu erfassen.

Ausgangsvariablen:

- **out_Measure** vom Typ **UDINT**
soll den gemessenen Wert aus dem FB *GetMeasure* erhalten.
- **out_ProductID** vom Typ **UDINT**
soll eine Identifikationsnummer (ID) erhalten, die das Produkt eindeutig identifiziert.
- **out_Ack** vom Typ **BOOL**
soll den Wert **FALSE** erhalten, solange noch kein Messergebnis bzw. keine ID in die entsprechenden Ausgangsvariablen geschrieben worden sind.

```

PROGRAM MAIN
VAR
    in_Req AT%I* : BOOL := FALSE;

    out_Measure AT%Q* : UDINT := 0;
    out_ProductID AT%Q* : UDINT := 0;
    out_Ack AT%Q* : BOOL;

    request : BOOL := FALSE;
    isReady : BOOL;

    FB_R_TRIG : R_TRIG;
    FB_F_TRIG : F_TRIG;
    FB_GetMeasure : GETMEASURE;
    FB_TON : TON;
END_VAR

(*    START:    Detecting an incoming request
           to deliver the measure of the next product*)
FB_R_TRIG( CLK := in_Req );
IF FB_R_TRIG.Q THEN
    out_Ack := FALSE;
    request := TRUE;
END_IF;

FB_F_TRIG( CLK := in_Req );
IF FB_F_TRIG.Q THEN
    out_Ack := FALSE;
    request := FALSE;
END_IF;

(*    END:    Detecting an incoming request*)

```

```

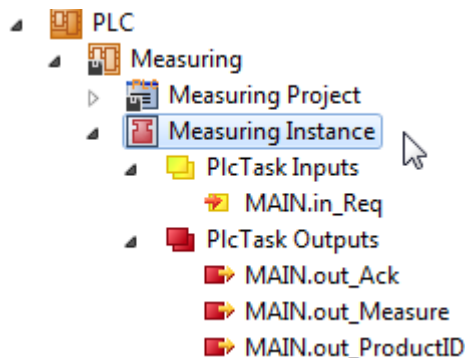
(*      START:      Simulating duration for measuring the next product*)
IF isReady THEN
  IF request THEN
    FB_GetMeasure(    nominalMeasure := 5000,
                     variance := 1000,
                     steps := 100,
                     measure => out_Measure );

    out_ProductID := out_ProductID + 1;
    request := FALSE;
    isReady := FALSE;
    out_Ack := TRUE;
  END_IF;
ELSE
  FB_TON( PT := T#200MS,
         IN := NOT FB_TON.Q );

  isReady := FB_TON.Q;
END_IF;
(*      END:      Simulating duration for measuring the next product*)

```

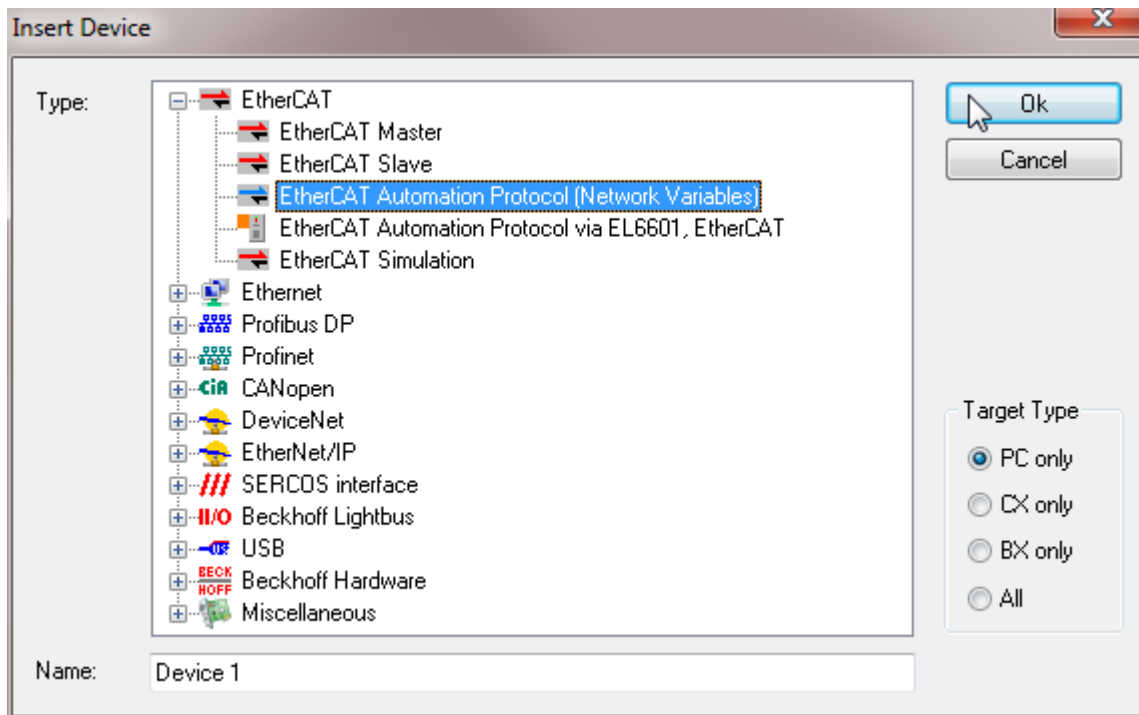
1. Kopieren Sie den Programmcode des Hauptprogramms in das Programm *MAIN*.
 2. Übersetzen Sie das SPS-Projekt, indem Sie den Menüeintrag [Build] → [Build Solution] anklicken.
- ⇒ Es wird eine Instanz für das SPS-Projekt erzeugt, in der Sie die Ein- und Ausgangsvariablen des SPS-Programms als Prozessvariablen wiederfinden (siehe Abbildung). Diese können später in TwinCAT verknüpft werden.



Das EAP-Gerät

Zuletzt muss noch ein EAP-Gerät bei der I/O Konfiguration in TwinCAT angelegt werden:

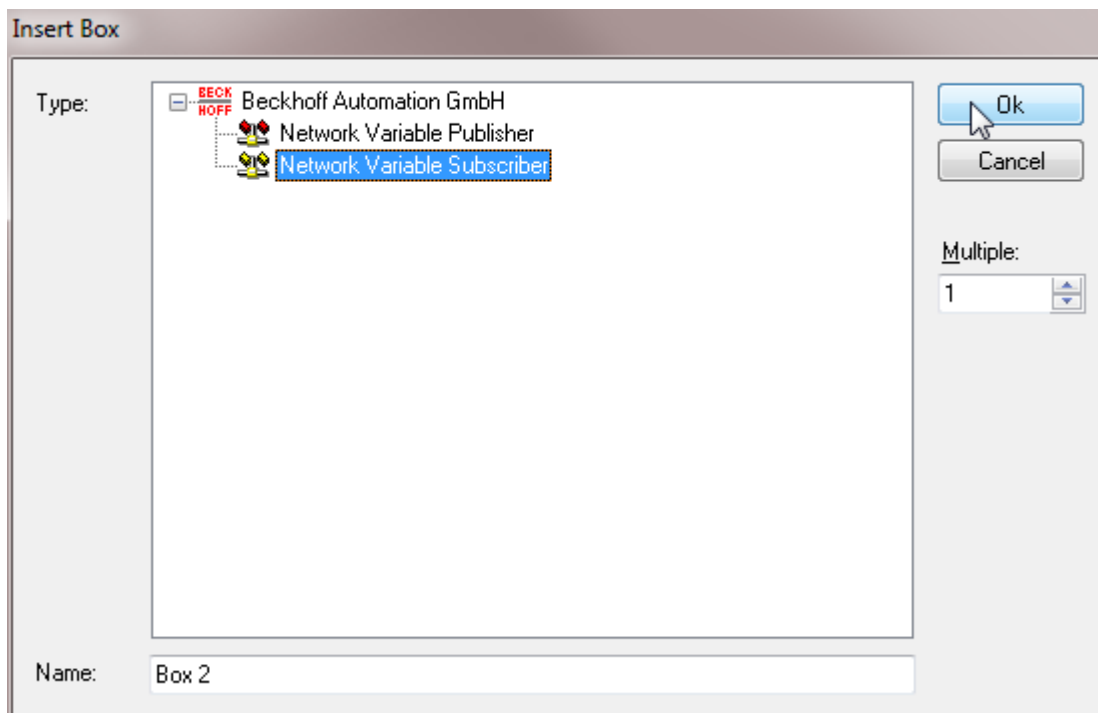
1. Klappen Sie den Knoten I/O TwinCAT 3 auf und klicken Sie mit der rechten Maustaste auf den Knoten *Device*.
 2. Wählen Sie den Eintrag [Add New Item...] im Kontextmenü.
- ⇒ Es erscheint der Dialog *Insert Device*.



1. Hier wählen Sie aus der Liste den Type: *EtherCAT Automation Protocol (Network Variables)* welches zu den *EtherCAT* Geräten gehört und drücken [OK].
2. Evtl. werden Sie danach aufgefordert, einen Adapter auszuwählen, über den das EAP-Gerät kommunizieren soll.
⇒ Danach ist das EAP-Gerät erstellt und mit Standardeinstellungen vorkonfiguriert.

Nun muss das EAP-Gerät noch mit einer zyklischen Task verknüpft werden, sodass es im Echtzeitkontext Variablen versenden und empfangen kann.

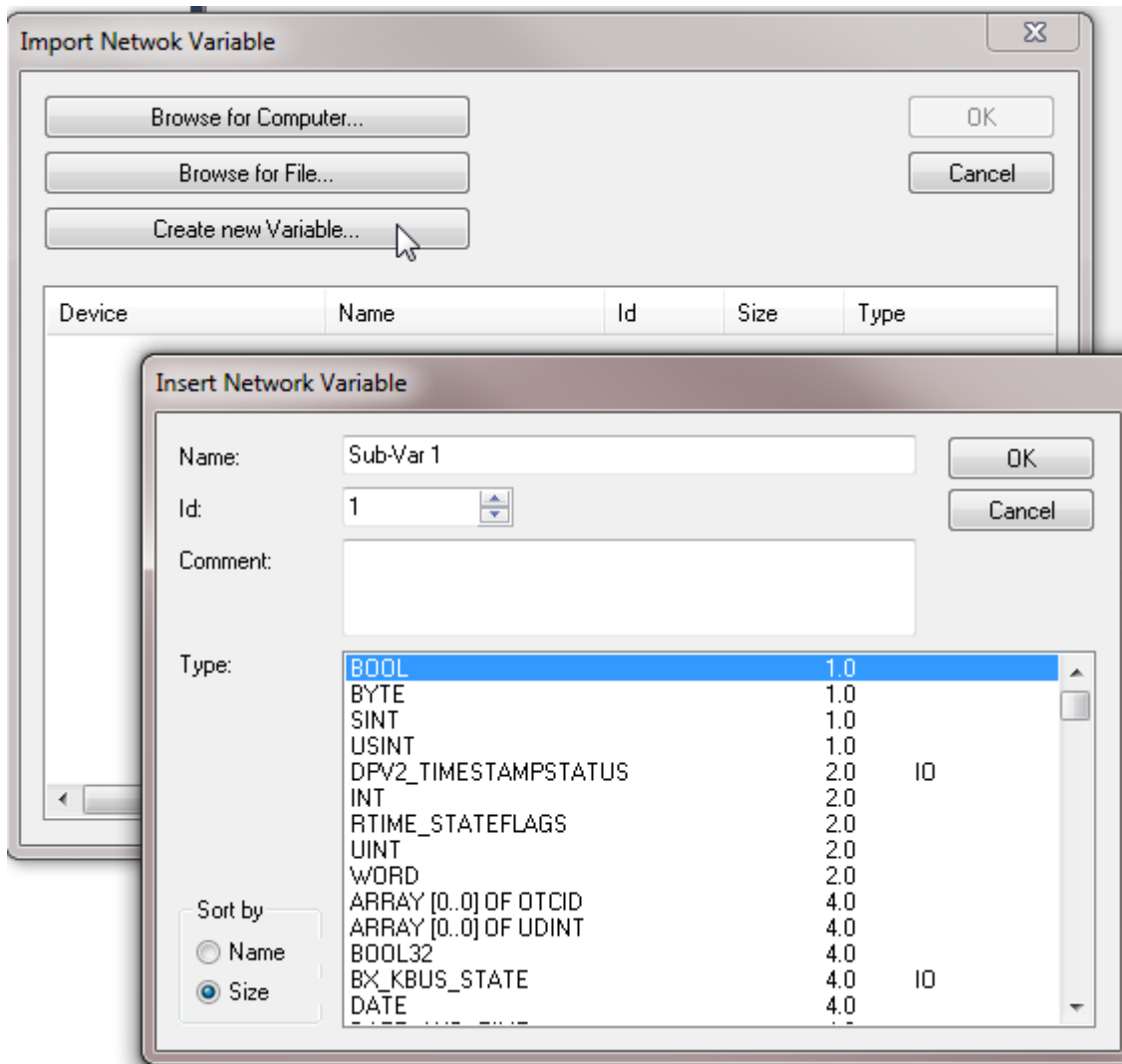
3. Klicken Sie mit der rechten Maustaste auf das EAP-Gerät und wählen Sie den Menüeintrag [Add New Item...].
⇒ Es öffnet sich der Dialog *Insert Box*.



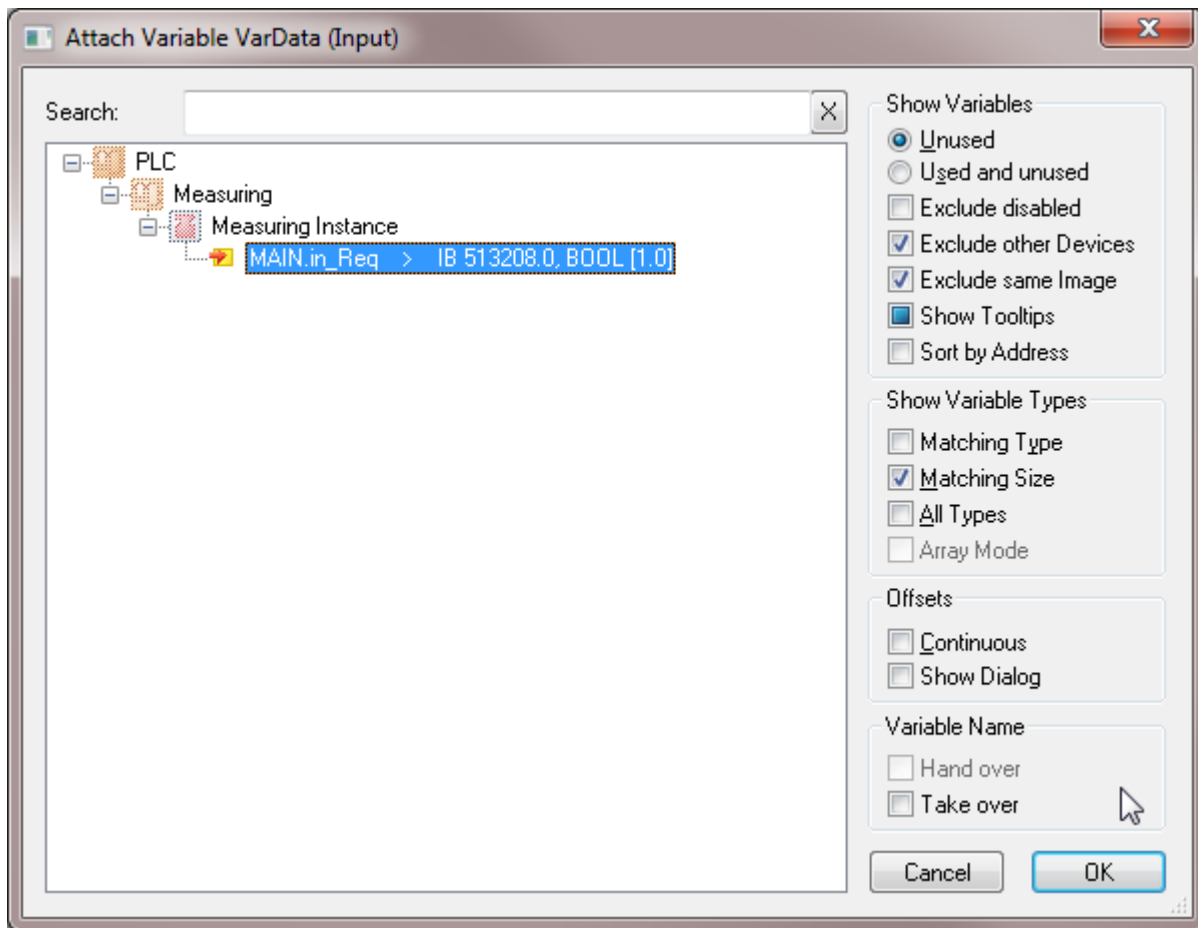
1. Wählen Sie *Network Variable Subscriber* aus und drücken Sie [OK].
⇒ Eine *Box 1 (Subscriber)* wird unterhalb des EAP-Gerätes angelegt.

2. Klicken Sie mit der rechten Maustaste auf diesen neuen Knoten und wählen Sie wieder den Menüeintrag [Add New Item...].

⇒ Der Dialog *Import Network Variable* wird angezeigt.



1. Drücken Sie [Create new Variable...].
⇒ Der Dialog *Insert Network Variable* öffnet sich (siehe Abbildung oben).
2. Wählen Sie aus der Liste der Datentypen *BOOL* aus und drücken dann [OK].
⇒ Die Subscriber Variable *Sub-Var 1* wird unterhalb des Subscriber Knotens angefügt.
3. Klappen Sie die *Sub-Var 1* und darunter den Knoten *Inputs* auf.
4. Klicken Sie mit rechts auf den Knoten *VarData* und wählen Sie im Kontextmenü den Eintrag [Change Link...].
⇒ Der Dialog *Attach Variable* wird angezeigt.



1. Wählen Sie die SPS-Variable *MAIN.in_Req* aus und drücken Sie [OK].

⇒ Anschließend ist die Variable verknüpft und die Konfiguration des Projektes abgeschlossen.

Aktivieren Sie Ihr TwinCAT-Projekt, um später mit dem TwinCAT EAP Configurator eine Verbindung zum EAP-Gerät herstellen zu können.

Das TwinCAT-Projekt Operating

Das Anlegen des TwinCAT-Projektes *Operating* funktioniert analog wie beim TwinCAT-Projekt *Measuring*. Es wird daher nur auf die wenigen Unterschiede und auf den Programmcode des SPS-Projektes eingegangen.

Das Operating SPS-Beispielprogramm

Das *Operating* Beispielprogramm soll auf einfachste Art einen Bearbeitungsvorgang bei einem Produkt oder Werkstück mit einer Identifikationsnummer simulieren und nach der Bearbeitung einer anderen Steuerung per EAP mitteilen können, dass das nächste Werkstück bearbeitet werden kann.

Für diese Aufgabe wird ein Funktionsbaustein (FB) mit der Bezeichnung *Processing* erzeugt, der die Dauer der Bearbeitung simuliert, die von einem Eingangsparameter abhängt (dem gemessenen Wert zu diesem Werkstück). Das Hauptprogramm (*MAIN*) übernimmt die Funktion eine Anfrage für das nächste Werkstück zu stellen, wenn die letzte Bearbeitung abgeschlossen ist und die Bestätigung der Anfrage zu erfassen, sodass mit der Bearbeitung des nächsten Werkstücks begonnen werden kann.

Funktionsbaustein Processing

Der FB soll folgende Eingabeparameter haben:

- *measure*
Der Wert der beim zu bearbeitenden Werkstück gemessen worden ist.

Der FB simuliert eine Bearbeitungsdauer, indem er den Wert des Eingabeparameters halbiert und in Sekunden wartet. Während dieser Wartezeit wird der Ausgabeparameter *busy* auf den Wert TRUE gesetzt, andernfalls ist der Wert FALSE. Für den FB *Processing* kann der Programmcode dann folgendermaßen aussehen:

```
FUNCTION_BLOCK PUBLIC Processing

VAR_INPUT
    measure : UDINT := 0;
END_VAR

VAR_OUTPUT
    bBusy : BOOL := FALSE;
END_VAR

VAR
    FB_TON : TON;
    bReady : BOOL;
    processingTime : TIME;
END_VAR

processingTime := UDINT_TO_TIME(measure/2);

FB_TON.PT := processingTime;
FB_TON.IN := NOT FB_TON.Q;
FB_TON();

bBusy := NOT FB_TON.Q;
```

1. Kopieren Sie den Programmcode in den FB *Processing*.
2. Öffnen Sie das Programm *MAIN* per Doppelklick im Editor.

⇒ Hier wird nun das Hauptprogramm kodiert. Im Hauptprogramm werden drei Eingangsvariablen und eine Ausgangsvariable definiert.

Das Hauptprogramm MAIN:

Eingangsvariablen:

- in_Measure vom Typ UDINT
soll den gemessenen Wert aus dem Projekt *Measuring* erhalten.
- in_ProductID vom Typ UDINT
soll eine Identifikationsnummer (ID) erhalten, die das Produkt eindeutig identifiziert und vom Projekt *Measuring* mitgeteilt wird.
- in_Ack vom Typ BOOL
soll den Wert FALSE vom Projekt *Measuring* erhalten, solange das nächste Werkstück noch nicht gemessen worden ist, also noch nicht zur Bearbeitung bereitsteht.

Ausgangsvariable:

- out_Req vom Typ BOOL
wird verwendet, um das nächste Werkstück bei der anderen Steuerung anzufragen.

```
PROGRAM MAIN

VAR
    in_Measure AT%I* : UDINT := 0;
    in_ProductID AT%I* : UDINT := 0;
    in_Ack          AT%I* : BOOL := FALSE;

    out_Req AT%Q* : BOOL := FALSE;

    acknowledge : BOOL := FALSE;
    proceeding : UDINT := 0;

    FB_R_TRIG : R_TRIG;
    FB_F_TRIG : F_TRIG;
    FB_Processing : Processing;
END_VAR

(*    START:    Detecting an incoming acknowledge
           to operating the next product*)
FB_R_TRIG( CLK := in_Ack );
IF FB_R_TRIG.Q THEN
    acknowledge := TRUE;
END_IF;
```

```
FB_F_TRIG( CLK := in_Ack );
IF FB_F_TRIG.Q THEN
    acknowledge := FALSE;
END_IF;
(*    END:    Detecting an incoming acknowledge*)

(*    START:    Simulating duration of operating the next product*)
IF FB_Processing.bBusy THEN
    FB_Processing();
ELSE
    IF acknowledge THEN
        acknowledge := FALSE;
        out_Req := FALSE;
        proceeding := in_ProductID;
        FB_Processing(measure := in_Measure);
    ELSE
        proceeding := 0;
        out_Req := TRUE;
    END_IF;
END_IF;
(*    END:    Simulating duration of operating the next product*)
```

1. Kopieren Sie den Programmcode des Hauptprogramms in das Programm *MAIN*.
 2. Übersetzen Sie das SPS-Projekt, indem Sie den Menüeintrag [Build] → [Build Solution] anklicken.
- ⇒ Anschließend ist das Projekt übersetzt.

Das EAP-Gerät

Beim EAP-Gerät wird bei diesem Projekt statt eines Subscribers ein Publisher mit einer Publisher Variablen vom Datentyp *BOOL* hinzugefügt. diese Variable wird dann mit der SPS-Variablen *MAIN.out_Req* verknüpft.

Aktivieren Sie auch dieses TwinCAT-Projekt, um später mit dem TwinCAT EAP Configurator eine Verbindung zum EAP-Gerät herstellen zu können.

7 Support und Service

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

Downloadfinder

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

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

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

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

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

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

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

Hotline: +49 5246 963-157
E-Mail: support@beckhoff.com

Beckhoff Service

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

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

Hotline: +49 5246 963-460
E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

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

Trademark statements

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

Third-party trademark statements

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

Mehr Informationen:
www.beckhoff.com/te1610

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

