

Handbuch | DE

# CXxxxx-M310/B310

Profibus-Optionsschnittstelle für CX9020, CX5xx0 und CX20xx



# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Sicherheitshinweise .....	6
1.3	Ausgabestände der Dokumentation .....	7
<b>2</b>	<b>Systemübersicht PROFIBUS</b> .....	<b>8</b>
2.1	Konfigurationsmöglichkeiten .....	10
2.2	Kommunikationsprotokolle und Dienste .....	14
2.3	Technische Daten - PROFIBUS .....	16
<b>3</b>	<b>Anschluss und Verkabelung</b> .....	<b>17</b>
3.1	Profibus-Anschluss .....	17
3.2	Verkabelung .....	18
3.3	Topologie .....	21
<b>4</b>	<b>TwinCAT Registerkarten</b> .....	<b>22</b>
4.1	Strukturansicht .....	22
4.2	Profibus-Master .....	24
4.2.1	General .....	24
4.2.2	CCAT PBM .....	25
4.2.3	Bus-Parameter (DP) .....	26
4.2.4	Start-Up/Fault-Settings .....	27
4.2.5	ADS .....	28
4.2.6	DP-Diag .....	29
4.2.7	Box States .....	30
4.3	Profibus-Slave .....	31
4.3.1	Profibus .....	31
4.3.2	Features .....	32
4.3.3	Diag .....	33
<b>5</b>	<b>Parametrieren und in Betrieb nehmen</b> .....	<b>34</b>
5.1	Profibus synchronisieren .....	34
5.2	Mit TwinCAT 2 parametrieren .....	38
5.2.1	Zielsysteme suchen .....	38
5.2.2	Profibus Slave anfügen .....	40
5.2.3	Virtuellen Slave anlegen .....	42
5.2.4	Adresse einstellen .....	43
5.2.5	PLC-Projekt erstellen .....	44
5.2.6	Variablen verknüpfen .....	46
5.2.7	Konfiguration auf CX laden .....	47
5.2.8	Profibus Master anfügen .....	49
5.2.9	Profibusvernetzung testen .....	50
5.2.10	Prozessdaten drehen .....	51
5.3	Mit TwinCAT 3 parametrieren .....	52
5.3.1	Zielsysteme suchen .....	52
5.3.2	Profibus Slave anfügen .....	54
5.3.3	Virtuellen Slave anlegen .....	56

5.3.4	Adresse einstellen .....	57
5.3.5	PLC-Projekt erstellen.....	58
5.3.6	Variablen verknüpfen.....	60
5.3.7	Konfiguration auf CX laden.....	61
5.3.8	Profibus Master anfügen.....	63
5.3.9	Profibusvernetzung testen .....	64
5.3.10	Prozessdaten drehen.....	65
<b>6</b>	<b>Fehlerbehandlung und Diagnose.....</b>	<b>66</b>
6.1	Diagnose-LEDs .....	66
6.2	Fehlerreaktionen.....	67
6.3	Master.....	70
6.3.1	ADS-Interface .....	72
6.3.2	DPV1-Kommunikation .....	74
6.3.3	Upload der Konfiguration .....	75
6.4	Slave.....	76
6.4.1	Slave-Diagnose .....	78
6.4.2	Individuelle Diagnosedaten.....	81
6.4.3	DP-V1 Kommunikation .....	83
6.4.4	DP-V1 Fehler Codes.....	85
<b>7</b>	<b>Anhang .....</b>	<b>86</b>
7.1	Zubehör .....	86
7.2	Zertifizierungen .....	87
7.3	Support und Service .....	88

# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

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

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

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

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

### Disclaimer

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

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

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

### Marken

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC® 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 Bezeichnungen führen.

### Patente

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

EP1590927, EP1789857, DE102004044764, DE102007017835

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

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

EP0851348, US6167425 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

## 1.2 Sicherheitshinweise

### Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!

Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

### Haftungsausschluss

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

### Qualifikation des Personals

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

### Erklärung der Symbole

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

#### **GEFAHR**

##### **Akute Verletzungsgefahr!**

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

#### **WARNUNG**

##### **Verletzungsgefahr!**

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

#### **VORSICHT**

##### **Schädigung von Personen!**

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

#### **HINWEIS**

##### **Schädigung von Umwelt oder Geräten**

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



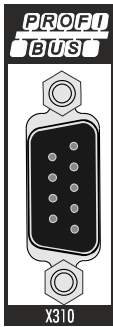
#### **Tipp oder Fingerzeig**

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

## 1.3 Ausgabestände der Dokumentation

Version	Änderungen
1.0	erste Version

## 2 Systemübersicht PROFIBUS



Die Beckhoff Embedded-PCs können ab Werk mit einer Optionsschnittstelle z.B. PROFIBUS, CANopen oder RS232 bestellt werden. Einige der Optionsschnittstellen können als Master oder Slave ausgeliefert werden.

Folgende Embedded-PCs können mit einer Optionsschnittstelle bestellt werden:

- CX9020
- CX50x0
- CX51x0
- CX20x0

### PROFIBUS-Master (M310)

Die Optionsschnittstelle M310 ist ein PROFIBUS-Master und ermöglicht einen segmentartigen Aufbau von Steuerungsstrukturen in weitläufigen Anlagen und Maschinen. Weitere Beckhoff Feldbuskomponenten wie Buskoppler, Busklemmen Controller, Antriebstechnik etc. können mit einem Embedded-PC für den Aufbau von Steuerungsstrukturen benutzt werden.

Mit Feldbusmastern werden Prozessdaten und Signale dezentral auch bei weitläufigen Maschinen und Anlagen eingesammelt. Die Anzahl der möglichen Slaves, die an den Master angeschlossen werden können, ist nur durch das jeweilige Bussystem begrenzt. Durch den Einsatz von Master- bzw. Slaveanschlüssen können mehrere Embedded-PC untereinander über die Feldebene vernetzt werden.

In TwinCAT werden die Optionsschnittstellen erkannt, parametrierbar, konfiguriert und die angeschlossenen I/O-Komponenten angefügt. Auch die Diagnose erfolgt in TwinCAT.

### PROFIBUS-Slave (B310)

Die Optionsschnittstelle B310 ist ein PROFIBUS-Slave und ermöglicht es einen Embedded-PC als unterlagerte dezentrale Steuerung für den Aufbau komplexer oder modularer Anlagen einzusetzen.

Der PROFIBUS-Slave empfängt externe Prozessdaten vom Master, verarbeitet diese Daten weiter oder schickt Daten aus der eigenen Prozessperipherie aufbereitet an den Master zurück.

Die PROFIBUS-Slave Optionsschnittstelle wird wie der PROFIBUS-Master in TwinCAT parametrierbar und konfiguriert.

### Funktionsweise

PROFIBUS ist ein herstellerunabhängiger, offener Feldbusstandard mit breitem Anwendungsbereich in der Fertigungs- und Prozessautomatisierung. Herstellerunabhängigkeit und Offenheit sind durch die internationalen Normen EN 50170 und EN 50254 garantiert.

#### Weitere Informationen

PROFIBUS-Nutzerorganisation (PNO):  
[www.profibus.com](http://www.profibus.com)

PROFIBUS ermöglicht die Kommunikation von Geräten verschiedener Hersteller ohne besondere Schnittstellenanpassungen. PROFIBUS ist sowohl für schnelle, zeitkritische Anwendungen, als auch für komplexe Kommunikationsaufgaben geeignet.



Bei PROFIBUS DP kommunizieren zentrale Steuergeräte (wie z. B. Industrie-PCs oder PLCs) über eine schnelle serielle Verbindung mit dezentralen Ein- und Ausgabebaugruppen. RS485 ist die dabei am häufigsten genutzte Übertragungstechnik unter Verwendung einer geschirmten verdrehten Zweidrahtleitung. Der Datenaustausch erfolgt vorwiegend zyklisch, wobei azyklische Dienste (DP-V1) für Parametrierung und Diagnose zur Verfügung stehen.

PROFIBUS DP bietet kurze Systemreaktionszeiten: Bei einer 12-MBaud-Übertragungsrate werden weniger als 2 ms benötigt, um jeweils 512 Bit Eingangs- und Ausgangsdaten auf 32 Teilnehmer zu übertragen.

Alle Beckhoff-PROFIBUS-Geräte verfügen über eine leistungsfähige Protokollimplementierung und sind von der PROFIBUS Nutzerorganisation (PNO) zertifiziert.

PROFIBUS unterscheidet folgende Gerätetypen:

**Master-Geräte** bestimmen den Datenverkehr auf dem Bus. Ein Master darf Nachrichten ohne externe Aufforderung aussenden, wenn er im Besitz der Buszugriffsberechtigung (Token) ist. Master werden auch als aktive Teilnehmer bezeichnet.

**Slave-Geräte** sind Peripheriegeräte wie beispielsweise Ein-/Ausgangsgeräte, Ventile, Antriebe, Messumformer und die Beckhoff PROFIBUS-Slaves der Reihe BK3xx0, BC3xx0, IPxxx-B310, IL230x-B310 und IL230x-C310. Sie erhalten keine Buszugriffsberechtigung, d.h. sie dürfen nur empfangene Nachrichten quittieren oder auf Anfrage eines Masters Nachrichten an diesen übermitteln. Slaves werden als passive Teilnehmer bezeichnet. Sie benötigen nur einen geringen Anteil des Busprotokolls, dadurch wird eine aufwandsarme Implementierung ermöglicht.

## PROFIBUS-DP

PROFIBUS-DP ist für den effizienten Datenaustausch in der Feldebene konzipiert. Hier kommunizieren die zentralen Automatisierungsgeräte, wie z.B. SPS/PC oder Prozessleitsysteme, über eine schnelle serielle Verbindung mit dezentralen Feldgeräten wie E/A, Antriebe, Ventile etc. Der Datenaustausch mit den dezentralen Geräten erfolgt vorwiegend zyklisch. Die dafür benötigten Kommunikationsfunktionen sind durch die DP-Grundfunktionen gemäß EN 50170 festgelegt.

Über diese Grundfunktionen hinaus bietet PROFIBUS-DP auch erweiterte azyklische Kommunikationsdienste für zum Beispiel Parametrierung und Bedienung welche auch durch die Beckhoff PROFIBUS-Slaves der Reihe IPxxx-B310, IL230x-B310 und IL230x-C310 unterstützt werden. Eine zentrale Steuerung (Master) liest zyklisch die Eingangsinformationen von den Slaves und schreibt die Ausgangsinformationen zyklisch an die Slaves. Hierbei sollte die Buszykluszeit kürzer sein als die Programmzykluszeit des zentralen Automatisierungssystems, die in vielen Anwendungen etwa 10 ms beträgt.

## 2.1 Konfigurationsmöglichkeiten

Mit PROFIBUS-DP können Mono- oder Multi-Master-Systeme realisiert werden. Dadurch wird ein hohes Maß an Flexibilität bei der Systemkonfiguration ermöglicht. Es können maximal 126 Geräte (Master oder Slaves) an einem Bus angeschlossen werden. Mit den Beckhoff PROFIBUS-Slaves der Reihe IPxxxx-B310, IL230x-B310 und IL230x-C310 kann eine Stationsadresse zwischen 0 und 99 gewählt werden. Die Festlegungen zur Systemkonfiguration beinhalten die Anzahl der Stationen, die Zuordnung der Stationsadresse zu den E/A-Adressen, Datenkonsistenz der E/A-Daten, Format der Diagnosemeldungen und die verwendeten Busparameter. Jedes PROFIBUS-DP-System besteht aus unterschiedlichen Gerätetypen. Es werden drei Gerätetypen unterschieden:

Klasse	Beschreibung
<b>DP-Master Klasse 1 (DPM1)</b>	Hierbei handelt es sich um eine zentrale Steuerung, die in einem festgelegten Nachrichtenzyklus zyklisch Informationen mit den dezentralen Stationen (Slaves) austauscht. Typische Geräte sind z.B. Speicherprogrammierbare Steuerungen (SPS) oder PC.
<b>DP-Master Klasse 2 (DPM2)</b>	Geräte dieses Typs sind Engineering-, Projektierungs- oder Bediengeräte. Sie werden bei der Inbetriebnahme und zur Wartung und Diagnose eingesetzt, um die angeschlossenen Geräte zu konfigurieren, Messwerte und Parameter auszuwerten sowie den Gerätezustand abzufragen.
<b>DP-Slave</b>	Ein PROFIBUS-DP-Slave ist ein Peripheriegerät (E/A, Antriebe, Messumformer etc.), das Eingangsinformationen einliest und Ausgangsinformationen an die Peripherie abgibt. Es sind auch Geräte möglich, die nur Eingangs- oder nur Ausgangsinformationen bereitstellen. Die Menge der Eingangs- und Ausgangsinformationen ist geräteabhängig und darf max. 240 Byte Eingangs- und 240 Byte Ausgangsdaten betragen.

### Mono-Master-Systeme

Bei Mono-Master-Systemen ist in der Betriebsphase des Bussystems nur ein Master am Bus aktiv. Die SPS-Steuerung ist die zentrale Steuerungskomponente. Die Slaves sind über das Übertragungsmedium dezentral an die SPS-Steuerung gekoppelt. Mit dieser Systemkonfiguration wird die kürzeste Buszykluszeit erreicht.

### Gerätstammdatei (GSD)

Bei PROFIBUS-DP werden die Leistungsmerkmale der Geräte in Form eines Gerätedatenblattes und einer Gerätstammdatendatei von den Herstellern dokumentiert und den Anwendern zur Verfügung gestellt. Aufbau, Inhalt und Kodierung dieser Gerätstammdatensätze (GSD) sind standardisiert. Sie ermöglichen die komfortable Projektierung beliebiger PROFIBUS-DP-Slaves mit Projektierungsgeräten verschiedener Hersteller. Die PROFIBUS-Nutzer-Organisation (PNO) archiviert diese Informationen herstellerübergreifend und gibt auf Anfrage Auskünfte über die GSD aller Hersteller. Die GSD - Daten werden von einer PROFIBUS-Master - Konfigurationssoftware gelesen und entsprechende Einstellungen in den PROFIBUS-Master übertragen. Die Beschreibung entnehmen Sie bitte dem entsprechenden Softwarehandbuch des Masterherstellers.

Die Beckhoff GSD-Dateien sind über das Internet unter <http://www.beckhoff.de> erhältlich.

**Diagnosefunktionen**

Die umfangreichen Diagnosefunktionen von PROFIBUS-DP ermöglichen die schnelle Fehlerlokalisierung. Die Diagnose der Beckhoff Buskoppler ist in der Defaulteinstellung der Typdatei und der GSD-Datei nicht eingeschaltet. Die Diagnosemeldungen werden über den Bus übertragen und beim Master zusammengefasst.

Sie werden in drei Ebenen eingeteilt:

Diagnoseart	Beschreibung
<b>Stationsbezogen</b>	Meldungen zur allgemeinen Betriebsbereitschaft eines Teilnehmers wie z.B. Übertemperatur oder Unterspannung
<b>Modulbezogen</b>	Diese Meldungen zeigen an, dass innerhalb eines bestimmten E/A Teilbereichs (z.B. 8 Bit Ausgangs - Modul) eines Teilnehmers eine Diagnose ansteht
<b>Kanalbezogen</b>	Hier wird die Fehlerursache bezogen auf ein einzelnes Ein- / Ausgangs - Bit (Kanal) angegeben, wie z.B. Kurzschluss auf Ausgang 2

Die Beckhoff PROFIBUS-Slaves der Reihe IPxxxx-B310, IL230x-B310 und IL230x-C310 unterstützen die Diagnosefunktionen des PROFIBUS-DP. Die Auswertung der Diagnosedaten über die Steuerung hängt von der Unterstützung des PROFIBUS-Masters ab. Entnehmen Sie bitte den Gerätehandbüchern der Masteranschlüssen die Handhabung der Diagnose.

**Sync- und Freeze-Mode**

Zusätzlich zu dem Teilnehmer bezogenen Nutzdatenverkehr, der automatisch vom DPM1 abgewickelt wird, besteht für die DP-Master die Möglichkeit, Steuerkommandos an einen, eine Gruppe oder an alle DP-Slaves gleichzeitig zu senden. Diese Steuerkommandos werden als Multicast übertragen. Mit diesen Steuerkommandos können die Sync- und Freeze-Betriebsarten zur Synchronisation der DP-Slaves vorgegeben werden. Sie ermöglichen eine ereignisgesteuerte Synchronisation der DP-Slaves.

Die DP-Slaves beginnen den **Sync-Mode**, wenn sie vom zugeordneten DP-Master ein Sync-Steuerkommando empfangen. In diesem Betriebszustand werden bei allen adressierten DP-Slaves die Ausgänge auf den momentanen Zustand eingefroren. Bei den folgenden Nutzdatenübertragungen werden die Ausgangsdaten bei den DP-Slaves gespeichert, die Ausgangszustände bleiben jedoch unverändert. Erst beim Empfang des nächsten Sync-Steuerkommandos vom Master werden die gespeicherten Ausgangsdaten an die Ausgänge durchgeschaltet. Mit einem Unsync-Steuerkommando wird der Sync-Betrieb beendet.

Analog dazu bewirkt ein Freeze-Steuerkommando den **Freeze-Mode** der angesprochenen DP-Slaves. In dieser Betriebsart werden die Zustände der Eingänge auf den momentanen Wert eingefroren. Die Eingangsdaten werden erst dann wieder aktualisiert, wenn der DP-Master das nächste Freeze-Steuerkommando an die betroffenen Geräte gesendet hat. Mit Unfreeze wird der Freeze-Betrieb beendet.

## Systemverhalten

Um eine weitgehende Geräte austauschbarkeit zu erreichen, wurde bei PROFIBUS-DP auch das Systemverhalten standardisiert. Es wird im Wesentlichen durch den Betriebszustand des DPM1 bestimmt. Dieser kann entweder lokal oder über den Bus vom Projektierungs-Gerät gesteuert werden.

Es werden folgende drei Hauptzustände unterschieden:

Betriebsart	Beschreibung
<b>Stop</b>	Es findet kein Datenverkehr zwischen dem DPM1 und den DP-Slaves statt. Der Buskoppler spricht die Busklemmen nur einmal nach dem Einschalten der Versorgungsspannung an (keine der E/A - LEDs leuchtet).
<b>Clear</b>	Der DPM1 liest die Eingangsinformationen der DP-Slaves, und hält die Ausgänge der DP-Slaves im sicheren Zustand (Abhängig von der Reaktion auf Feldbusfehler leuchtet die grüne E/A-LED und werden die Ausgänge gesetzt).
<b>Operate</b>	Der DPM1 befindet sich in der Datentransferphase. In einem zyklischen Datenverkehr werden die Eingänge von den DP - Slaves gelesen und die Ausgangsinformationen an die DP-Slaves übertragen (Die grüne E/A-LED leuchtet).

Der DPM1 sendet seinen lokalen Status in einem konfigurierbaren Zeitintervall mit einem Multicast-Kommando zyklisch an alle ihm zugeordneten DP-Slaves. Die Systemreaktion nach dem Auftreten eines Fehlers in der Datentransferphase des DPM1, wie z. B. Ausfall eines DP-Slaves, wird durch den Betriebsparameter *Auto-Clear* bestimmt. Wurde dieser Parameter auf *True* gesetzt, dann schaltet der DPM1 die Ausgänge aller zugehörigen DP-Slaves in den sicheren Zustand, sobald ein DP-Slave nicht mehr bereit für die Nutzdatenübertragung ist. Danach wechselt der DPM1 in den Clear-Zustand. Ist dieser Parameter *False*, dann verbleibt der DPM1 auch im Fehlerfall im Operate-Zustand und der Anwender kann die Systemreaktion selbst bestimmen.

### Datenverkehr zwischen DPM1 und den DP-Slaves

Der Datenverkehr zwischen dem DPM1 und den ihm zugeordneten DP-Slaves wird in einer festgelegten immer wiederkehrenden Reihenfolge automatisch durch den DPM1 abgewickelt. Bei der Projektierung des Bussystems legt der Anwender die Zugehörigkeit eines DP-Slaves zum DPM1 fest. Weiterhin wird definiert, welche DP-Slaves in den zyklischen Nutzdatenverkehr aufgenommen oder ausgenommen werden sollen.

Der Datenverkehr zwischen dem DPM1 und den DP-Slaves gliedert sich in die Phasen Parametrierung, Konfigurierung und Datentransfer.

Bevor ein DP-Slave in die Datentransferphase aufgenommen wird, prüft der DPM1 in der Parametrierungs- und Konfigurations-Phase, ob die projektierte Sollkonfiguration mit der tatsächlichen Gerätekonfiguration übereinstimmt. Bei dieser Überprüfung müssen der Gerätetyp, die Format- und Längeninformationen sowie die Anzahl der Ein- und Ausgänge übereinstimmen. Der Benutzer erhält dadurch einen zuverlässigen Schutz gegen Parametrierungsfehler. Zusätzlich zum Nutzdatentransfer, der vom DPM1 automatisch durchgeführt wird, besteht die Möglichkeit neue Parametrierungsdaten auf Anforderung des Benutzers an die DP-Slaves zu senden.

**Schutzmechanismen**

Im Bereich der dezentralen Peripherie ist es aus Sicherheitsgründen erforderlich, die Systeme mit hochwirksamen Schutzfunktionen gegen Fehlparametrierung oder Ausfall der Übertragungseinrichtungen zu versehen. PROFIBUS-DP verwendet Überwachungsmechanismen beim DP-Master und bei den DP-Slaves. Sie werden als Zeitüberwachungen realisiert. Das Überwachungsintervall wird bei der Projektierung des DP-Systems festgelegt.

Schutzmechanismen	Beschreibung
<b>Auf dem DP-Master</b>	Der DPM1 überwacht den Nutzdatentransfer der Slaves mit dem Data_Control_Timer. Für jeden zugeordneten Slave wird ein eigener Überwachungszeitgeber benutzt. Die Zeitüberwachung spricht an, wenn innerhalb eines Überwachungsintervalls kein ordnungsgemäßer Nutzdatentransfer erfolgt. In diesem Fall wird der Benutzer informiert. Falls die automatische Fehlerreaktion (Auto_Clear =True) freigegeben wurde, verlässt der DPM1 den Zustand <i>Operate</i> , schaltet die Ausgänge der zugehörigen Slaves in den sicheren Zustand und geht in den Betriebszustand <i>Clear</i> über.
<b>Auf dem DP-Slave</b>	Der Slave führt zur Erkennung von Fehlern des Masters oder der Übertragungsstrecke die Ansprechüberwachung durch. Findet innerhalb des Ansprechüberwachungsintervalls kein Datenverkehr mit dem zugeordneten Master statt, dann schaltet der Slave die Ausgänge selbständig in den sicheren Zustand. Zusätzlich ist für die Ein- und Ausgänge der Slaves beim Betrieb in Multi - Master - Systemen ein Zugriffsschutz erforderlich damit sichergestellt ist, dass der direkte Zugriff nur vom berechtigten Master erfolgt. Für alle anderen Master stellen die Slaves ein Abbild der Eingänge und Ausgänge zur Verfügung, das von jedem beliebigen Master auch ohne Zugriffsberechtigung gelesen werden kann.

**Ident-Nummer**

Jeder DP-Slave und jeder DPM1 muss eine individuelle Ident-Nummer haben. Sie wird benötigt, damit ein DP-Master ohne signifikanten Protokoll-Overhead die Typen der angeschlossenen Geräte identifizieren kann. Der Master vergleicht die Ident-Nummer der angeschlossenen DP-Geräte mit den Ident-Nummern in den vom DPM2 vorgegebenen Projektierungsdaten. Der Nutzdatentransfer wird nur dann begonnen, wenn die richtigen Geräte-Typen mit den richtigen Stationsadressen am Bus angeschlossen wurden. Dadurch wird Sicherheit gegenüber Projektierungsfehlern garantiert. Die Vergabe der Herstellerspezifischen Ident-Nummern erfolgt durch die PROFIBUS Nutzerorganisation (PNO). Die PNO verwaltet die Ident-Nummern zusammen mit den Gerätestammdaten (GSD).

## 2.2 Kommunikationsprotokolle und Dienste

In PROFIBUS DP-Systemen kommuniziert in der Regel ein Master (SPS, PC, etc.) mit vielen Slaves (I/Os, Antriebe, etc.), wobei nur der Master aktiv auf den Bus zugreift (unaufgefordert Telegramme sendet), während ein DP-Slave nur Telegramme sendet, wenn er von einem Master dazu aufgefordert wird.

### PROFIBUS DP

- **DP-StartUp**

Bevor Master und Slave zyklisch Daten miteinander austauschen, werden während des DP-StartUps Parameter- und Konfigurationsdaten vom Master an die Slaves übertragen. Nachdem die Parameter- und Konfigurationsdaten gesendet wurden, fragt der Master solange die Diagnosedaten des Slaves ab, bis der Slave seine Bereitschaft zum Datenaustausch signalisiert. Anhängig vom Umfang der Berechnungen, die der Slave aufgrund der empfangenen Parameter- und Konfigurationsdaten durchzuführen hat, kann dieser Vorgang einige Sekunden dauern.

- **Parameterdaten**

Die Parameterdaten werden mit dem SetPrmLock-Request-Telegramm vom Master an die Slaves gesendet. Das SetPrmLock-Response-Telegramm enthält keine Daten und besteht daher nur aus einem Byte, der Kurzquittung. Die Parameterdaten bestehen aus DP-Parametern (z.B. dem DP-Watchdog und der IdentNumber), den DPV1-/DPV2-Parametern und aus anwendungsspezifischen Parametern, die nur einmal während des StartUps übertragen werden müssen. Wenn ein Fehler in den Parameterdaten auftritt, wird das in den Diagnosedaten gekennzeichnet und der Slave bleibt bzw. geht in den Zustand WAIT-PRM.

- **Konfigurationsdaten**

Die Konfigurationsdaten werden mit dem ChkCfg-Request-Telegramm vom Master an die Slaves gesendet. Das ChkCfg-Response-Telegramm enthält keine Daten und besteht daher nur aus einem Byte, der Kurzquittung. Die Konfigurationsdaten beschreiben die Zuordnung der DP-Module zu den zyklischen I/O-Daten, die mit dem Data\_Exchange-Telegramm während des zyklischen Datenaustauschs zwischen Master und Slave und ausgetauscht werden. Die Reihenfolge der an einem Slave angefügten DP-Module bestimmt die Reihenfolge der zugehörigen I/O-Daten im Data\_Exchange-Telegramm.

- **Diagnosedaten**

Die Diagnosedaten werden mit einem SlaveDiag-Request-Telegramm vom Master angefordert. Der Slave sendet die Diagnosedaten mit einem SlaveDiag-Response-Telegramm. Die Diagnosedaten bestehen aus den Standard DP-Diagnosedaten (z.B. Zustand des Slaves, IdentNumber) und anwendungsspezifischen Diagnosedaten.

- **Zyklischer Datenaustausch**

Kernstück des PROFIBUS DP-Protokolls ist der zyklische Datenaustausch, bei dem innerhalb eines PROFIBUS DP-Zyklus der Master mit jedem Slave einen I/O-Datenaustausch durchführt. Dabei sendet der Master an jeden Slave die Outputs mit einem DataExchange-Request-Telegramm. Der Slave antwortet mit den Inputs in einem DataExchange-Response-Telegramm. Sämtliche Output- bzw. Inputdaten werden also mit je einem Telegramm übertragen, wobei die DP-Konfiguration (Reihenfolge der DP-Module) die Zuordnung der Output- bzw. Inputdaten zu den realen Prozessdaten des Slaves festlegt.

- **Diagnose während des zyklischen Datenaustausches**

Während des zyklischen Datenaustausches kann ein Slave eine Diagnose an den Master melden. In dem Fall setzt der Slave ein Flag im DataExchange-Response-Telegramm, woran der Master erkennt, dass bei dem Slave neue Diagnosedaten vorliegen. Diese Daten holt der Master dann mit dem SlaveDiag-Telegramm ab. Die Diagnosedaten sind also nicht in Echtzeit mit den zyklischen I/O-Daten in der Steuerung, sondern mindestens immer einen DP-Zyklus später.

- **Synchronisierung mit Sync und Freeze**

Mit den Sync- und Freeze-Kommandos im GlobalControl-Request-Telegramm kann der Master die Ausgabe der Outputs (Sync) bzw. das Einlesen der Inputs (freeze) bei mehreren Slaves synchronisieren.

Wenn die Sync-Kommandos verwendet werden, werden die Slaves zunächst in den Sync-Mode geschaltet (wird in den Diagnosedaten quittiert). Danach werden dann die I/O-Daten per DataExchange-Telegramm sequentiell mit den Slaves ausgetauscht. Wenn das Sync-Kommando im GlobalControl-Telegramm gesendet wird, hat das zur Folge, dass die Slaves die zuletzt empfangenen

Outputs ausgeben. Im Freeze-Betrieb wird zunächst ein Freeze-Kommando im GlobalControl-Telegramm gesendet, woraufhin alle Slaves ihre Inputs latchen, die dann wiederum mit dem DataExchange-Telegramm vom Master sequentiell abgeholt werden.

- **Zustände im Master**  
Der Master unterscheidet die Zustände CLEAR (alle Outputs auf dem Fail\_Safe-Wert) und OPERATE (alle Outputs haben den Prozesswert). In der Regel wird der Master in den CLEAR-Mode geschaltet, wenn z.B. die SPS auf STOP geht.
- **DP-Master Class 1 und Class 2**  
Mit dem Class 1-Master wird die Steuerung bezeichnet, die zyklischen I/O-Datenaustausch mit den Slaves durchführt, während ein Class 2-Master ein B&B-Gerät ist, dass in der Regel nur lesend auf die I/O-Daten des Slaves zugreift.

## PROFIBUS DPV1

PROFIBUS DPV1 bezeichnet im Wesentlichen die azyklischen Read- und Write-Telegramme, mit denen azyklisch auf Datensätze im Slave zugegriffen wird. Auch bei DPV1 wird zwischen Class 1 und einem Class 2 Master unterschieden.

- **Class 1 (C1)**  
Die azyklische C1-Verbindung wird mit dem DP-StartUp des zyklischen DP-Betriebs aufgebaut. Ab dem Zustand WAIT-CFG des Slaves können azyklische DPV1-C1-Read- und Write-Telegramme vom Master zum Slave gesendet werden.
- **Class 2 (C2)**  
Bei C2 wird in der Regel von einem zweiten C2-Master eine separate Verbindung unabhängig von der zyklischen DP-Verbindung aufgebaut, sodass z.B. ein herstellerspezifisches Projektierungs- und Diagnosetool auf die Daten des Slaves zugreifen kann.  
Bei Verwendung von zwei Mastern muss beachtet werden, dass diese sich den Buszugriff teilen und somit die zeitlichen Verhältnisse ungünstiger im Vergleich zu einem Master sind.

## 2.3 Technische Daten - PROFIBUS

### Optionsschnittstelle-M310

Technische Daten	M310
Feldbus	PROFIBUS DP, DP-V1; DP-V2 (MC)
Übertragungsrate	9,6k; 19,2k; 93,75k; 187,5k; 500k; 1,5M; 3M; 6M; 12 MBaud
Businterface	1 x D-Sub-Buchse, 9-polig
Busteilnehmer	max. 125 mit Repeatern
max. Prozessabbild	30,5 kByte In / 30,5 kByte Out
Eigenschaften	PROFIBUS – unterschiedliche DP-Zykluszeiten pro Slave möglich; Error-Management für jeden Teilnehmer frei konfigurierbar

### Optionsschnittstelle-B310

Technische Daten	B310
Feldbus	PROFIBUS DP, DP-V1
Übertragungsrate	9,6k; 19,2k; 93,75k; 187,5k; 500k; 1,5M; 3M; 6M; 12 MBaud
Businterface	1 x D-Sub-Buchse, 9-polig
Erweiterbares Prozessabbild	bis zu 15 virtuelle Slaves zusätzlich
max. Prozessabbild	16 Slaves x (240 Byte In / 240 Byte Out)



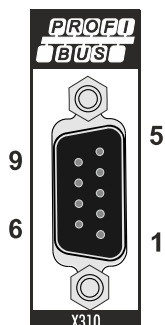
### 3 Anschluss und Verkabelung

#### 3.1 Profibus-Anschluss

##### Neunpoliger D-Sub

Pin 6 überträgt 5 V<sub>DC</sub> und Pin 5 GND für den aktiven Abschlusswiderstand. Diese dürfen auf keinen Fall für andere Funktionen benutzt werden, da dies zu Zerstörung des Gerätes führen kann.

Pin 3 und Pin 8 übertragen die Signale des PROFIBUS. Diese dürfen auf keinen Fall getauscht werden, da sonst die Kommunikation gestört ist.



Die Profibus Busleitung wird über einen 9poligen D-Sub mit folgender Belegung angeschlossen:

Pin	Belegung
1	Schirmung
2	nicht benutzt
3	RxD/TxD-P
4	nicht benutzt
5	GND
6	+5V <sub>DC</sub>
7	nicht benutzt
8	RxD/TxD-N
9	nicht benutzt

##### Leitungsfarben

PROFIBUS Leitung	D-Sub
B rot	Pin 3
A grün	Pin 8

## 3.2 Verkabelung

Die physikalische Datenübertragung ist in der PROFIBUS-Norm definiert (siehe PROFIBUS Schicht 1: Physical Layer).

Der Einsatzbereich eines Feldbus-Systems wird wesentlich durch die Wahl des Übertragungs-Mediums und der physikalischen Busschnittstelle bestimmt. Neben den Anforderungen an die Übertragungssicherheit sind die Aufwendungen für Beschaffung und Installation des Buskabels von entscheidender Bedeutung. Die PROFIBUS-Norm sieht daher verschiedene Ausprägungen der Übertragungstechnik unter Beibehaltung eines einheitlichen Busprotokolls vor.

### Kabelgebundene Übertragung:

Diese Version, gemäß dem US-Standard EIA RS-485, wurde als Grundversion für Anwendungen im Bereich der Fertigungstechnik, Gebäudeleittechnik und Antriebstechnik festgelegt. Es wird ein verdrehtes Kupferkabel mit einem Leiterpaar verwendet. Die Abschirmung kann in Abhängigkeit des beabsichtigten Einsatzgebietes (EMV-Gesichtspunkte beachten) entfallen.

Es stehen zwei Leitungstypen mit unterschiedlichen Höchstleitungslängen zur Verfügung (siehe Tabelle RS485).

RS-485 Übertragungstechnik nach PROFIBUS-Norm	
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden, Stickleitungen sind möglich
Medium	Abgeschirmtes verdrehtes Kabel, Schirmung darf abhängig von den Umgebungsbedingungen (EMV) entfallen
Anzahl der Stationen	32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar bis 125 Stationen
Max. Bus Länge ohne Repeater	100 m bei 12 MBit/s 200 m bei 1500 KBit/s, bis zu 1,2 km bei 93,75 KBit/s
Max. Bus Länge mit Repeater	Durch Leitungsverstärker (Repeater) kann die max. Buslänge bis zu 10 km vergrößert werden. Die Anzahl der möglichen Repeater ist mindestens 3 und kann je nach Hersteller bis zu 10 betragen.
Übertragungsgeschwindigkeit (in Stufen einstellbar)	9,6 kBit/s; 19,2 kBit/s; 93,75 kBit/s; 187,5 kBit/s; 500 kBit/s; 1500 kBit/s; 12 MBit/s
Steckverbinder	9-Pin D-Sub-Steckverbinder für IP20 M12 Rundsteckverbinder für IP65/67

### Kabelbedingte Störungen

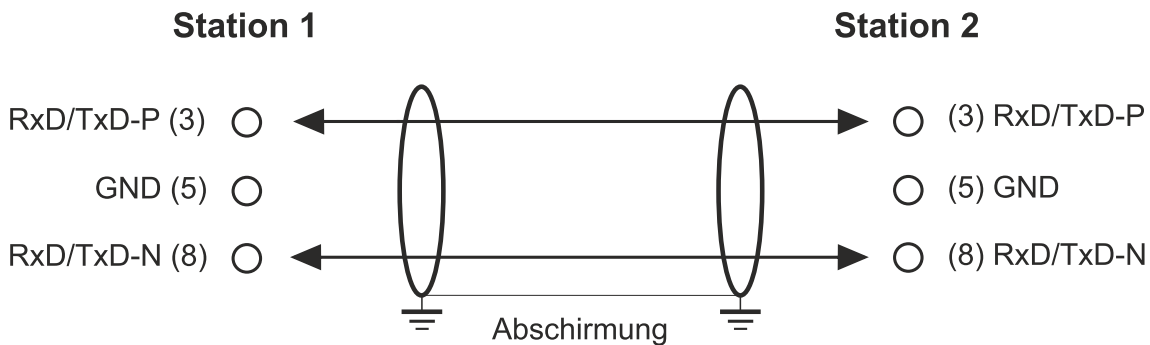
Beachten Sie die besonderen Anforderungen an das Datenkabel bei Baud-Raten von mehr als 1,5 Mbaud. Das richtige Kabel ist Grundvoraussetzung für den störungsfreien Betrieb des Bussystems. Bei der Verwendung des normalen 1,5 Mbaud-Kabels kann es durch Reflexionen und zu großer Dämpfung zu erstaunlichen Phänomenen kommen. Zum Beispiel bekommt eine angeschlossene PROFIBUS-Station keine Verbindung, kann diese aber nach Abziehen der benachbarten Station wieder aufnehmen. Oder es kommt zu Übertragungsfehlern, wenn ein bestimmtes Bitmuster übertragen wird. Das kann bedeuten, dass der PROFIBUS ohne Funktion der Anlage störungsfrei arbeitet und nach dem Hochlauf zufällig Busfehler meldet. Eine Reduzierung der Baud-Rate (< 93,75 kbaud) beseitigt das geschilderte Fehlerverhalten.

Führt die Verringerung der Baud-Rate nicht zur Beseitigung des Fehlers, liegt häufig ein Verdrahtungsfehler vor. Die beiden Datenleitungen sind an einem oder mehreren Steckern gedreht oder Abschlusswiderstände sind nicht oder an falschen Stellen aktiviert.

#### ● Vorkonfektionierte Kabel von BECKHOFF

**I** Mit den vorkonfektionierten Kabeln von BECKHOFF vereinfacht sich die Installation erheblich! Verdrahtungsfehler werden vermieden und die Inbetriebnahme führt schneller zum Erfolg. Das BECKHOFF-Programm umfasst Feldbuskabel, Stromversorgungskabel und Sensorkabel sowie Zubehör wie Abschlusswiderstände und T-Stücke. Ebenso sind jedoch auch Feldkonfektionierbare Stecker und Kabel erhältlich.

Die folgende Abbildung zeigt die Verkabelung zwischen zwei Stationen und wie die D-Sub-Anschlüsse belegt sind:



**i Abschlusswiderstände**

In Systemen mit mehr als zwei Stationen werden alle Teilnehmer parallel verdrahtet. An den Leitungsenden muss das Buskabel in jedem Fall mit Widerständen abgeschlossen werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden.

**Längenausdehnung**

Die Busleitung ist in der EN 50170 spezifiziert. Daraus ergibt sich die nachfolgende Längenausdehnung eines Bussegment.

Baud-Rate in kBits/sec	9,6	19,2	93,75	187,5	500	1500	12000
Leitungslänge in m	1200	1200	1200	1000	400	200	100

Stichleitungen bis 1500 kBaud <6,6 m, bei 12 MBaud sollten keine Stichleitungen verwendet werden.

**Bussegment**

Ein Bussegment besteht aus maximal 32 Teilnehmern. An einem PROFIBUS-Netzwerk sind 125 Teilnehmer erlaubt. Um diese Anzahl zu erreichen sind Repeater erforderlich, die das Signal auffrischen. Dabei wird jeder Repeater wie ein Teilnehmer angesehen.

IP-Link ist das Sub-Bussystem der Feldbus Boxen dessen Topologie eine Ringstruktur ist. In den Koppler Modulen (IP230x-Bxxx oder IP230x-Cxxx) befindet sich ein IP-Link Master, an den bis zu 120 Erweiterungsmodule (IExxxx) angeschlossen werden dürfen. Der Abstand zwischen zwei Modulen darf dabei 5 m nicht überschreiten. Achten Sie bei der Planung und Installation der Module, dass aufgrund der Ringsstruktur das letzte Modul wieder am IP-Link Master angeschlossen werden muss.

**Einbaurichtlinie**

Beachten Sie bei der Montage der Module und beim Verlegen der Leitung die technischen Richtlinien der PROFIBUS-Nutzerorganisation e.V. zu PROFIBUS-DP/FMS (siehe: [www.profibus.de](http://www.profibus.de)).

**PROFIBUS Kabel überprüfen**

Ein PROFIBUS-Kabel (bzw. ein Kabel-Segment bei Verwendung von Repeatern) kann mit ein paar einfachen Widerstandsmessungen überprüft werden. Dazu sollte das Kabel von allen Stationen abgezogen werden:

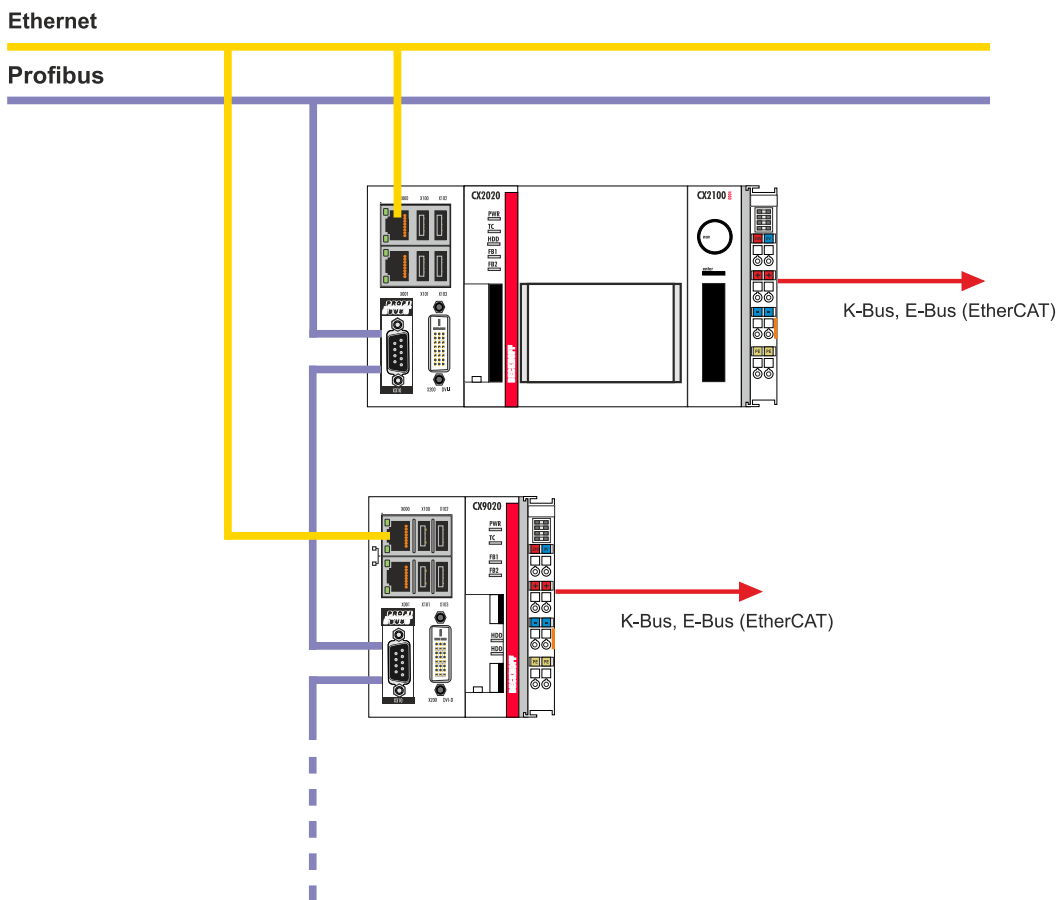
1. Widerstand zwischen A und B am Anfang der Leitung: ca. 110 Ohm
2. Widerstand zwischen A und B am Ende der Leitung: ca. 110 Ohm
3. Widerstand zwischen A am Anfang und A am Ende der Leitung: ca. 0 Ohm
4. Widerstand zwischen B am Anfang und B am Ende der Leitung: ca. 0 Ohm

5. Widerstand zwischen Schirm am Anfang und Schirm am Ende der Leitung: ca. 0 Ohm

Falls diese Messungen erfolgreich sind, ist das Kabel in Ordnung. Wenn trotzdem noch Bus-Störungen auftreten, liegt es meistens an EMV-Störungen. Beachten Sie die Installationshinweise der PROFIBUS-Nutzer-Organisation ([www.profibus.com](http://www.profibus.com)).

### 3.3 Topologie

- Ein Bussegment darf aus maximal 32 Teilnehmern bestehen (einschließlich den Repeatern).
- Die maximale Leitungslänge eines Segmentes ist abhängig von der verwendeten Übertragungsgeschwindigkeit und der Qualität der verwendeten Busleitung.
- Zwischen zwei Teilnehmer dürfen maximal 9 Repeater installiert werden.
- Stichleitungen sind zu vermeiden und ab 1,5 MBaud nicht zulässig.
- Maximale Anzahl an Teilnehmern 125
- Unterbrechung der Versorgungsspannungen von Leitungsabschlüssen durch abschalten des Repeaters/Slave oder abziehen des Steckers ist nicht zulässig.



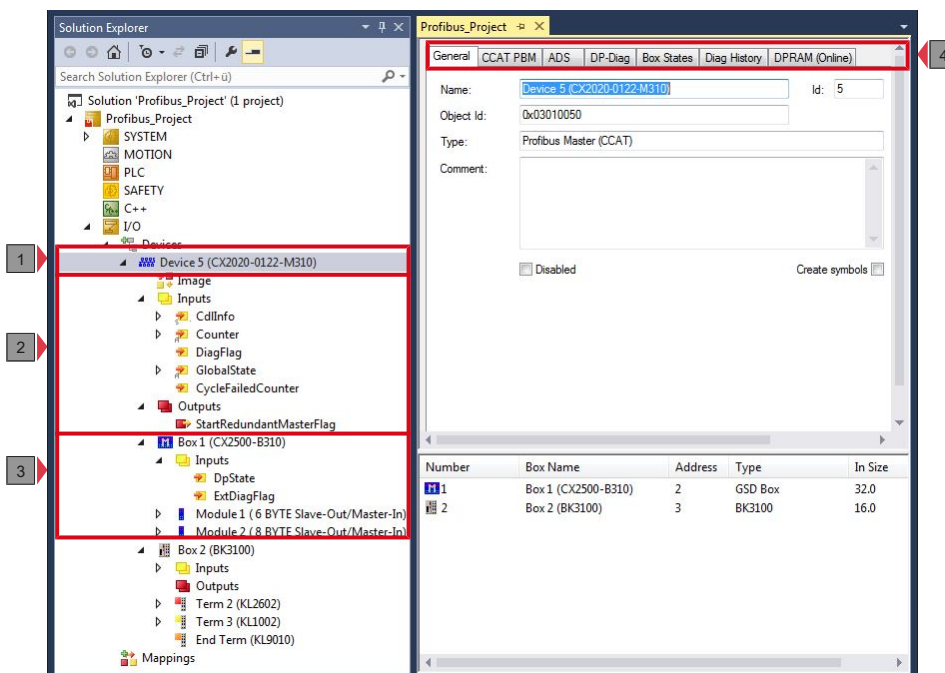
## 4 TwinCAT Registerkarten

In TwinCAT werden unter den Registerkarten Informationen und Einstellungen für die Profibus-Schnittstelle einsortiert. In diesem Kapitel werden die wichtigsten TwinCAT Registerkarten beschrieben. Zusätzlich dazu wird gezeigt, wie die Profibus-Schnittstelle unter TwinCAT in der Strukturansicht angezeigt wird.

Die Strukturansicht und die Registerkarten für eine Profibus-Schnittstelle sind unter TwinCAT2 und TwinCAT3 identisch.

### 4.1 Strukturansicht

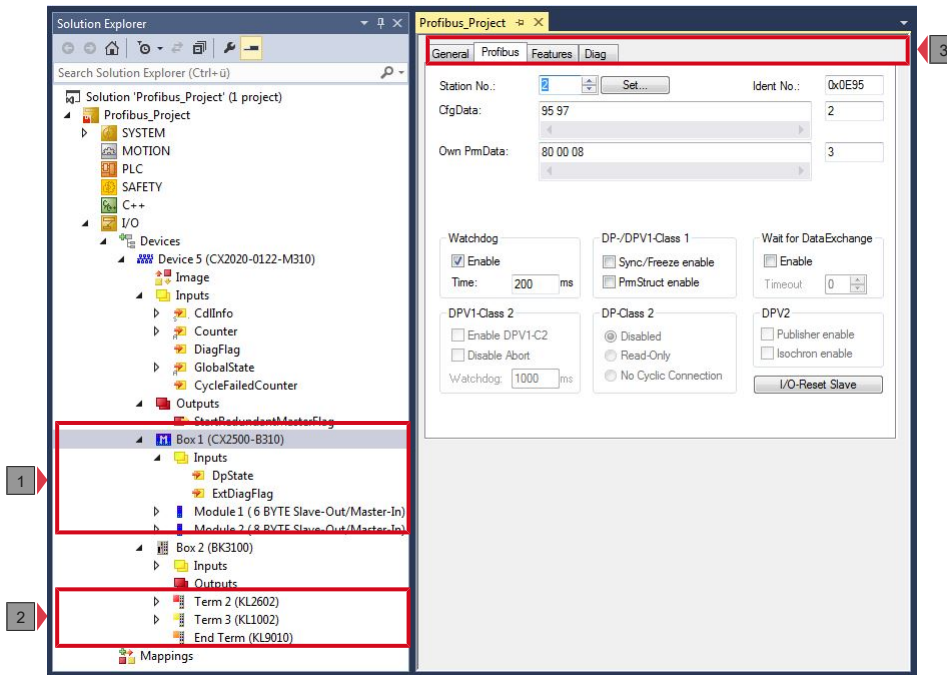
Ein Profibus-Master und ein Profibus-Slave werden wie folgt in der Strukturansicht angezeigt:



In diesem Beispiel wurde der Slave mit dem Master verbunden. Anschließend wurde in TwinCAT nach dem Master gescannt und der Master zusammen mit dem Slave in TwinCAT angefügt.

Nummer	Beschreibung
1	Der Gerätenamen des Masters wird in Klammern angezeigt. Alle Profibus-Slaves werden unter dem Master einsortiert.
2	Unter dem Profibus-Master werden Statusmeldungen als Eingangsvariablen und Ausgangsvariablen aufgelistet. Die Variablen können mit der SPS verknüpft und für Diagnosezwecke verwendet werden (z.B. Fehlercodes, Zähler usw.).
3	Profibus-Slaves werden unter dem Master einsortiert, als Box bezeichnet und fortlaufend nummeriert. Der Gerätenamen erscheint in Klammern dahinter. Jeder Profibus-Slave hat eigene Eingangsvariablen für Diagnosezwecke, die den Zustand der Kommunikation anzeigen (DPState, ExtDiagFlag).
4	Unter den Registerkarten lassen sich weitere Einstellungen für den Profibus-Master oder Slave vornehmen. Abhängig davon ob der Master oder Slave in der Strukturansicht ausgewählt wird, werden andere Registerkarten angezeigt.

Ein Profibus-Slave und die dazugehörigen Registerkarten werden wie folgt in der Strukturansicht angezeigt:



Nummer	Beschreibung
1	Unter dem Profibus-Slave werden Statusmeldungen als Eingangsvariablen aufgelistet. Die Variablen können mit der SPS verknüpft und für Diagnosezwecke verwendet werden. Pro Slave Box sind die Eingänge DpState und ExtDiagFlag angelegt. Zusätzlich dazu werden Prozessdaten für den Datenaustausch unter dem Slave einsortiert.
2	Alle Slaves werden unter dem Master einsortiert. Wie in diesem Beispiel der B3100 mit angesteckten Busklemmen. Alle verfügbaren Klemmen werden mit ihren Ein- und Ausgängen angezeigt.
3	Unter den Registerkarten lassen sich weitere Einstellungen für den Profibus-Slave vornehmen. Abhängig davon ob der Master oder Slave in der Strukturansicht ausgewählt wird, werden andere Registerkarten angezeigt.

Wird das SPS-Prozessabbild eingelesen, können die Variablen für Statusmeldungen und die Variablen unter den Prozessdaten mit den Variablen aus dem SPS-Programm verknüpft werden. Mit einem Doppelklick auf den Variablennamen in der Strukturansicht wird der Verknüpfungsdialo geöffnet. Die verknüpften Variablen werden mit einem kleinen Pfeilsymbol markiert.

Weitere Informationen zu TwinCAT finden Sie in der TwinCAT Dokumentation auf der Beckhoff Homepage: [www.beckhoff.de](http://www.beckhoff.de)

## 4.2 Profibus-Master

### 4.2.1 General

Die Registerkarte General liefert allgemeine Informationen zu einem Profibus-Gerät, wie Name, Typ und die Id.

The screenshot shows the 'General' tab of the TwinCAT software interface. The interface is divided into several sections:
 

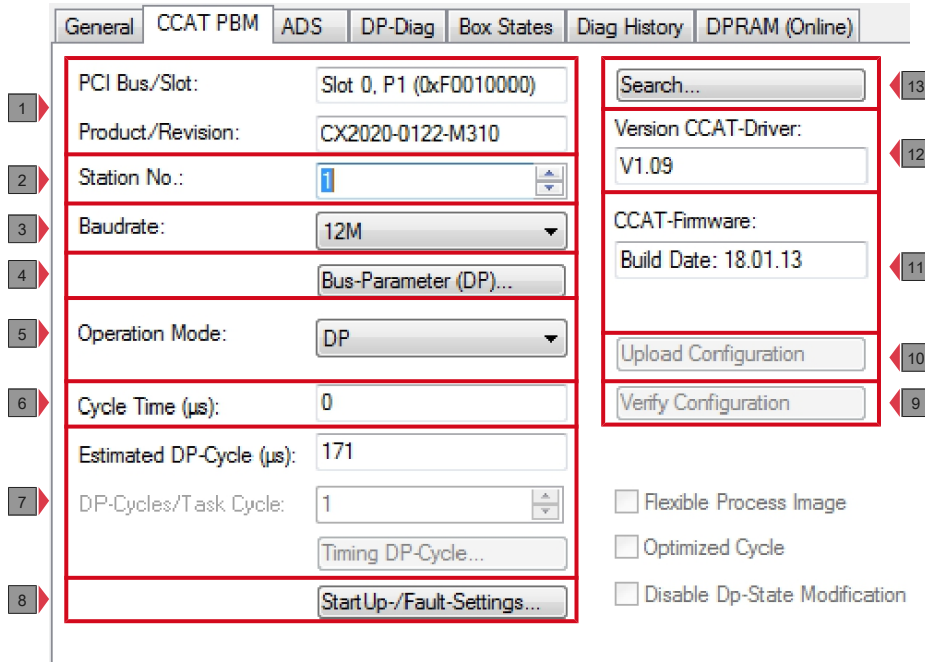
- Name:** Device 2 (CX2020-0122-M310) (highlighted with a red box and arrow 1)
- Object Id:** 0x03010020
- Type:** Profibus Master (CCAT) (highlighted with a red box and arrow 2)
- Comment:** (empty text area, highlighted with a red box and arrow 3)
- Disabled:**  Disabled (highlighted with a red box and arrow 4)
- Id:** 2 (highlighted with a red box and arrow 5)
- Create symbols:**

Nr.	Beschreibung
1	Name des Profibus-Geräts
2	Typ des Profibus-Geräts
3	Hier können Sie einen Kommentar (z.B. Anmerkungen zum Anlagenteil) hinzufügen
4	Hier können Sie das Profibus-Gerät deaktivieren
5	Laufende Nr.

Über diese Registerkarte kann das Profibus-Gerät ausgeschaltet werden. Ein Kommentarfeld bietet die Möglichkeit einer Beschriftung, um auf diese Weise zusätzliche Informationen zu dem Gerät bereitzustellen.



### 4.2.2 CCAT PBM

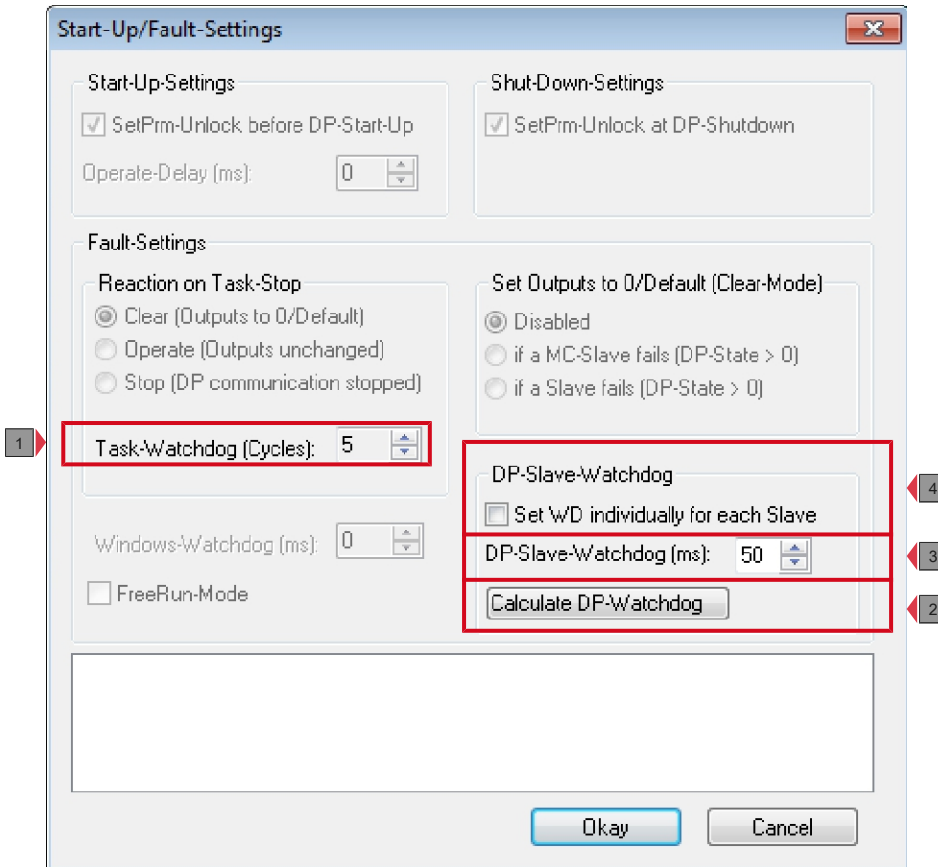


Nr.	Beschreibung
1	Bezeichnung der physikalischen Schnittstelle. Name und Typ des Profibus-Gerätes.
2	Eindeutige Stations-Nr. des Masters.
3	Hier wird die Baudrate eingestellt.
4	Mit dieser Schaltfläche wird das Fenster Bus-Parameter geöffnet (siehe: Bus-Parameter (DP) [► 26]).
5	Hier wird zwischen den drei Betriebsarten: DP, DP (Equidistant/no GC) und DP/MC (Equidistant) gewählt. Bei allen drei Betriebsarten übernimmt die höchstpriorie Task, die mit dem entsprechenden Gerät verknüpft ist, die Ansteuerung des PROFIBUS-Zyklus und ist somit mit dem DP-Zyklus synchronisiert.
6	Hier wird die Zykluszeit der zugehörigen höchstpriorien Task angezeigt.
7	An dieser Stelle wird die voraussichtliche Profibus-Zykluszeit angezeigt.
8	Mit dieser Schaltfläche wird das Fenster StartUp-/Fault Settings geöffnet (siehe: Start-Up/Fault-Settings [► 27])
9	Hier wird der Profibus gescannt und mit den aktuell angefügten Boxen verglichen. Abweichungen werden angezeigt.
10	Hier wird der PROFIBUS gescannt und alle gefundenen Geräte werden dem Master hinzugefügt. Bei Beckhoff-Boxen wird die Konfiguration genau ausgelesen, bei Fremdgeräten wird nach der zugehörige GSD-Datei gesucht.
11	Hier wird die aktuelle Firmware-Version angezeigt.
12	Hier wird die aktuelle Version der CCAT-Treiber angezeigt.
13	Über die Schaltfläche <b>Search</b> wird nach der physikalischen Schnittstelle gesucht und die gewünschte ausgewählt, wenn nicht bereits automatisch geschehen.

### 4.2.3 Bus-Parameter (DP)

Nr.	Beschreibung
1	Die Slot-Time gibt an, wie lange der DP-Master auf eine Antwort des DP-Slaves wartet, bevor er eine Wiederholung oder das nächste Telegramm sendet.
2	Die min. TsdR gibt an, wie lange der DP-Slave mindestens mit einer Antwort wartet. Diese Zeit wird bei allen DP-Slaves während des DP-Start-Ups eingestellt (Wertebereich 11-255 Bit-Zeiten). Die min. TsdR muss kleiner als die max. TsdR sein.
3	Die max. TsdR gibt an, wie lange der DP-Slave höchstens mit einer Antwort warten darf. Diese Zeit wird entsprechend der GSD-Datei-Einträge der DP-Slaves eingestellt. Die max. TsdR muss kleiner als die Slot-Time sein.
4	Das Max-Retry-Limit gibt an, wie oft ein Telegramm erneut gesendet wird, wenn der adressierte Teilnehmer nicht antwortet. Hier sollte mindestens ein Wert von 1 eingestellt werden, damit bei azyklischen Telegrammen im Fehlerfall das Telegramm mindestens einmal wiederholt wird.
5	Da das Data_Exchange-Telegramm zyklisch wiederholt wird, könnte man für die Wiederholung des Data_Exchange-Telegrammes hier den Wert 0 eintragen, um im Equidistant-Betrieb den Zyklus auch bei einem nicht antwortenden Teilnehmer einigermaßen konstant zu halten.  In diesem Fall sollte aber sinnigerweise auf dem Karteireiter Features der Box eingestellt werden, dass ein Nicht-Antworten des Slaves nicht zum Verlassen des DATA-EXCH führt (siehe: <a href="#">Features [► 32]</a> ). Dass ein Teilnehmer nicht geantwortet hat, kann man immer am DpState erkennen, der dann für einen Zyklus ungleich 0 ist.
6	Hier kann zwischen Master-Funktionalität (Standardeinstellung) und Multi-Slave unterschieden werden.
7	Mit dieser Schaltfläche werden die Bus-Parameter optimiert.
8	Mit dieser Schaltfläche werden die Standard Bus-Parameter eingestellt.

### 4.2.4 Start-Up/Fault-Settings



Nr.	Beschreibung
1	Der DP-Master wechselt automatisch in den Clear-Mode (Ausgänge der Slaves werden auf 0 bzw. Fail_Safe-Werte gesetzt), wenn er keinen Interrupt mehr von der ihm zugehörigen Task bekommt (z.B. PLC-Breakpoint oder -Absturz). Hier wird eingestellt, wie viele fehlende Task-Zyklen toleriert werden, bevor der Master in den Clear-Mode schaltet.
2	Über diese Schaltfläche wird der DP-Watchdog aller DP-Slaves auf einen optimalen Wert eingestellt.
3	Wenn die Option <b>Set WD individually for each Slave</b> nicht angeklickt ist, kann hier der DP-Watchdog auf einen einheitlichen Wert für alle Slaves eingestellt werden.
4	Hier kann angewählt werden, ob der Watchdog individuell für jeden Slave eingestellt werden soll. Wenn die Check-Box angewählt ist, kann für jeden Slave auf der Registerkarte Profibus der Watchdog eingestellt werden (siehe: <a href="#">Profibus [► 31]</a> ).

## 4.2.5 ADS

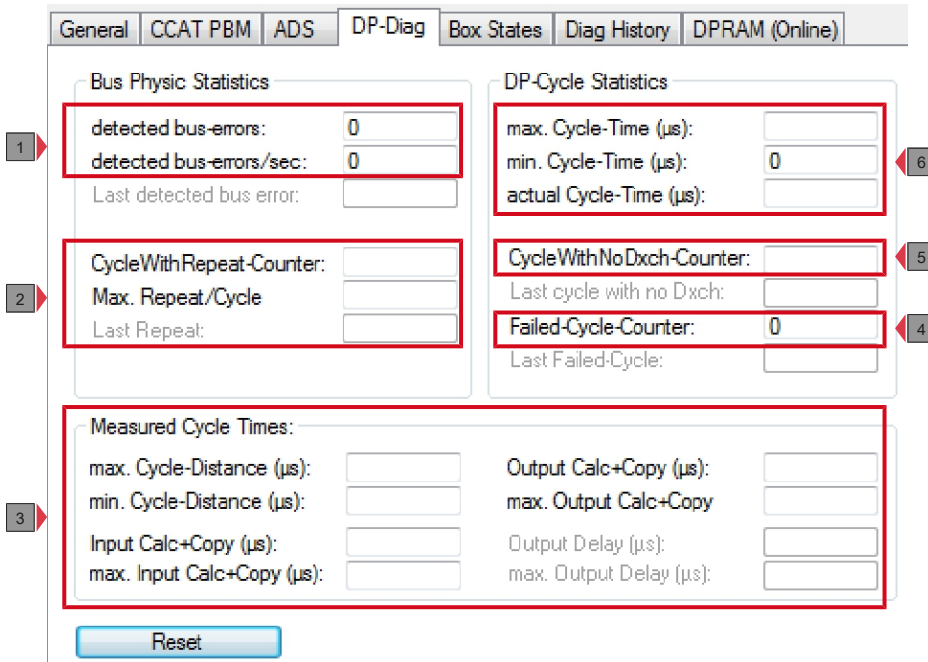
The screenshot shows the 'ADS' configuration tab in the TwinCAT software. The window has several tabs: 'General', 'CCAT PBM', 'ADS', 'DP-Diag', 'Box States', 'Diag History', and 'DPRAM (Online)'. The 'ADS' tab is active. It contains the following fields and controls:

- Use Port
- Port No: 28678 (0x7006) [Change...]
- NetId: 5.19.8.122.7.1
- Remote Name: Device 6 (CX2020-0122-M310)
- Add. NetIds: [Empty list box] [Add] [Delete]

Der Profibus-Master ist ein ADS-Device mit einer eigenen Net-Id, die hier verändert werden kann. Alle ADS-Dienste (Diagnose, azyklische Kommunikation), die an den Profibus-Master gehen, müssen diese Net-Id und Port-Nr. adressieren.



### 4.2.6 DP-Diag

An dieser Stelle werden Probleme mit der Verkabelung und den DP-Zykluszeiten angezeigt.



Nr.	Beschreibung
1	detected bus-errors: Hier wird die Anzahl der festgestellten Busfehler angezeigt. Wenn dieser Zähler ungleich 0 ist, sollte die Verkabelung geprüft werden, falls keine PROFIBUS-Stecker abgezogen oder gesteckt wurden. In der Regel gibt es beim Abziehen oder Stecken von PROFIBUS-Steckern auch kurze Busstörungen.
2	CycleWithRepeatCounter: Hier wird die Anzahl der PROFIBUS-Zyklen angezeigt, in denen mindestens einmal ein Telegramm wiederholt wurde. Wiederholungen sind auch ein Anzeichen dafür, dass die Busphysik nicht ohne Fehler ist. Max. Repeat/Cycle: Hier wird die maximale Anzahl von Wiederholungen innerhalb eines Zyklus angezeigt.
3	Measured Cycle Times: Hier werden die gemessenen Zykluszeiten angezeigt.
4	Failed-Cycle-Counter: Der Zähler zählt hoch, wenn der DP-Zyklus nicht fertig wurde, bevor der nächste Task-Zyklus gestartet wurde und alle Slaves im Datenaustausch sind (d.h. einen DpState gleich 0 haben).
5	CycleWithNoDxch-Counter: zählt hoch, wenn nicht alle Slaves Daten austauschen (d.h. einen DpState ungleich 0 haben).
6	max./min./actual Cycle-Time: Hier werden maximale, minimale und aktuelle DP-Zykluszeiten angezeigt, wobei nur Zyklen berücksichtigt werden, in denen alle Slaves Daten ausgetauscht haben und es keine Wiederholungen gab.

## 4.2.7 Box States

Station...	Box.State	RepeatCounter	NoAnswerCounter
 31	No error	0	0
 33	No error	0	0

Im Karteireiter Box States werden die angeschlossenen Profibus Slaves aufgelistet. Diese Übersicht kann dazu benutzt werden, um Fehler wie z.B. Verbindungsprobleme an den Boxen zu erkennen.

## 4.3 Profibus-Slave

### 4.3.1 Profibus

Nr.	Beschreibung
1	Hier wird die auf dem Slave hardwareseitig eingestellte Adresse eingetragen. Es gibt Slaves, bei denen die Adresse nicht hardwareseitig, sondern nur über den SetSlaveAddress-Dienst eingestellt werden kann. In diesem Fall wird über die Schaltfläche <b>Set</b> ein Dialog gestartet, mit dem ein SetSlaveAddress-Telegramm gesendet und auf diese Weise eine Adresse eingestellt wird.
2	Die aktuellen Konfigurationsdaten (ergeben sich aus den angefügten Modulen bzw. Klemmen) sowie deren Länge werden angezeigt.
3	Erlaubt die Profibus-spezifischen Parameterdaten zu editieren. Die Größe der aktuellen Parameterdaten wird ebenfalls angezeigt. Die PrmData können aber in der Regel textuell (-> PrmData (Text)) oder bei Beckhoff-DP-Slaves teilweise über den Karteireiter "Beckhoff" eingestellt werden.
4	Watchdog: Schaltet den DP-Watchdog ein. Wenn der Slave bei eingeschaltetem Watchdog für die Dauer der Watchdog-Zeit kein DP-Telegramm empfängt, verlässt er automatisch den Datenaustausch. Die minimale Watchdog-Zeit hängt von der DP-Zykluszeit ab und sollte größer als der Wert sein, der nach der folgenden Formel berechnet wird: <b>Estimated-Cycle-Time * 10</b> . Für besonders kritische Ausgänge kann bei DP-Slaves, die eine Watchdog-Base-Time von 1 ms unterstützen ein DP-Watchdog auf bis zu 2 ms eingestellt werden. Die DP-Watchdog-Time sollte allerdings mindestens doppelt so groß wie das Maximum aus <b>Cycle-Time</b> und <b>Estimated Cycle-Time</b> sein (siehe: <a href="#">CCAT PBM  &gt; 25</a> ).
5	Mit dieser Schaltfläche kann bei gestartetem TwinCAT der zyklische Datenaustausch zu dem DP-Slave abgebaut und gleich wieder aufgebaut werden (entspricht einem IO-Reset aber nur für den einen Slave).
6	In der Betriebsart DP/MC (Equidistant) des Masters können Slaves mit Sync und Freeze betrieben werden.
7	Hier wird die Ident-Nummer aus der GSD-Datei angezeigt.

## 4.3.2 Features

Nr.	Beschreibung
1	Hier kann eingestellt werden, ob der Slave weiterhin Daten senden soll (Stay in Data-Exch), auch wenn er nicht oder fehlerhaft antwortet. In diesem Fall wird der Datenaustausch erst verlassen, wenn der Slave innerhalb der Ansprechüberwachungszeit (Funktion Watchdog ist aktiviert, s. Registerkarte Profibus des Slaves) nie korrekt geantwortet hat. Andernfalls wird der Datenaustausch erst beendet, wenn der Slave 65535 mal nicht korrekt geantwortet hat.
2	Hier kann eingestellt werden ob der Slave automatisch wieder hochlaufen oder im Zustand WaitPrm bleiben soll, wenn er den Datenaustausch verlassen hat.
3	Je Slave kann angegeben werden, ob dessen Verlassen von Data-Exch zum Stop des PROFIBUS-Zyklus führen soll (alle Slaves verlassen den Datenaustausch und gehen nach Wait-Prm. Der Wiederanlauf ist nur durch IO-Reset oder Restart des TwinCAT Systems möglich).
4	Je Slave kann angegeben werden, ob beim Verlassen von Data-Exch (DpState wird ungleich 0) die Inputdaten des Slaves auf 0 gesetzt oder unverändert bleiben sollen.



### 4.3.3 Diag



Diese Registerkarte bietet nochmal eine Übersicht über den Zustand des Slaves und der Profibus-Verbindung. Hier können die Diagnosedaten der vorangegangenen Registerkarten gebündelt angezeigt werden. Folgende und noch weitere wichtige Informationen können hier eingesehen werden:

**BoxState:** Hier wird der aktuelle DPState angezeigt.

**Receive-Error-Counter:** Anzahl der gestörten Telegramme vom Slave.

**Repeat-Counter:** Anzahl der notwendigen Wiederholungen aufgrund keiner oder einer gestörten Antwort des Slaves.

**NoAnswer-Counter:** Anzahl der Telegramme, auf die der Slave nicht geantwortet hat.

**letzter DPV1-Fehler:** Error-Decode, Error-Class, Error-Code und Error-Code 2 (s. Beschreibung DPV1-Fehlercodes).

## 5 Parametrieren und in Betrieb nehmen

In dieser Dokumentation werden exemplarisch Profibus-Geräte für die Inbetriebnahme verwendet. Die im Kapitel gezeigten Konfigurationsmöglichkeiten können damit für alle Embedded-PCs mit Profibus-Schnittstelle verwendet werden.

Folgende Geräte werden in dieser Dokumentation verwendet:

- CX2020-M310 (Embedded-PC mit Optionsschnittstelle Profibus-Master, D-Sub-Buchse, 9-polig)
- CX2500-B310 (Embedded-PC mit Feldbusmodul CX2500-B310 Profibus-Slave, D-Sub-Buchse, 9-polig)
- BK3100 (Profibus-Slave, D-Sub-Buchse, 9-polig)

Als Software für die Konfiguration der Geräte wird TwinCat2 und TwinCat3 verwendet.

Für weitere Informationen stehen die Dokumentationen für TwinCat2 und TwinCat3 auf der Beckhoff Homepage zur Verfügung:

[www.beckhoff.de](http://www.beckhoff.de)

### 5.1 Profibus synchronisieren

Im TwinCAT Run Mode ist der Profibus-Master immer mit der höchstpriorien Task synchronisiert, mit der Variablen verknüpft sind. Die Zykluszeit der entsprechenden Task wird in der **Cycle-Time** auf dem Karteireiter CCAT PBM des Masters angezeigt, sobald einmal das Mapping erzeugt wurde. Bei der Task kann eingestellt werden, ob das I/O am Taskanfang aktualisiert werden soll oder nicht.

Voraussetzungen für diesen Arbeitsschritt:

- Eine Task erstellt in TwinCAT, mit der Variablen aus dem SPS-Projekt verknüpft sind.

**Sie können die Option I/O at task begin wie folgt aktivieren:**

1. Klicken Sie links in der Strukturansicht auf die entsprechende Task.
2. Klicken Sie auf die Registerkarte **Task** und klicken Sie auf die Option **I/O at task begin**.

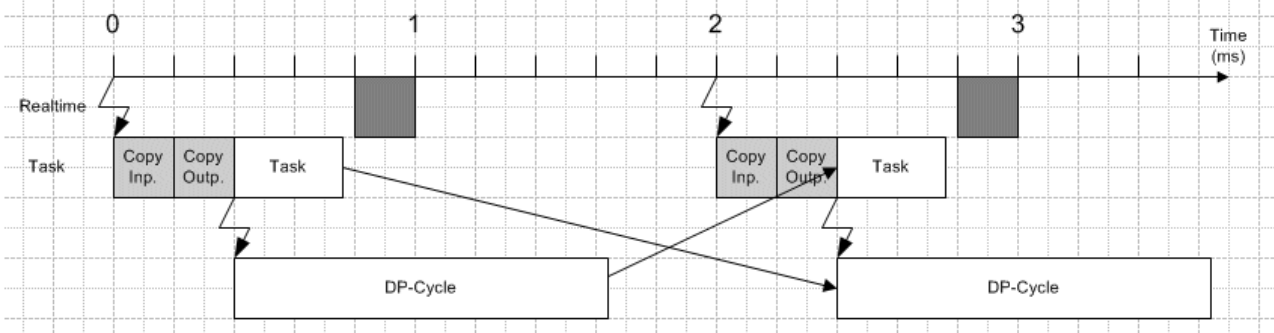


- ⇒ Sie haben die Option I/O am Taskanfang erfolgreich aktiviert. Im Folgenden wird erklärt, welche Auswirkung diese Einstellung auf die Taskzeit und den DP-Zyklus hat.

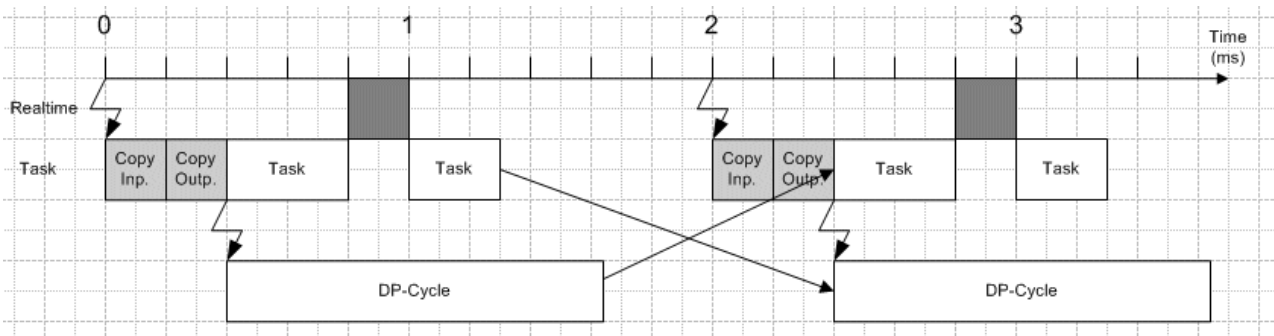
**I/O am Taskanfang**

Wenn bei der Task die Einstellung **I/O at task begin** aktiviert ist (Standardeinstellung bei NC-Task), wird vor dem Start der Task geprüft, ob der vorangegangene DP-Zyklus beendet wurde, dann die Inputs und Outputs (die Outputs sind dann vom letzten Task-Zyklus) kopiert und der DP-Zyklus gestartet.

Beispiel: Task-Zykluszeit 2 ms, Echtzeit-Ressourcen 80%.



Wenn in dem vorliegenden Fall das Kopieren der Inputs und Outputs sowie die Task-Rechenzeit 0,8 ms überschreiten, würde die Ausführung der Task unterbrochen werden, da die 80% Echtzeit-Ressourcen erreicht sind:



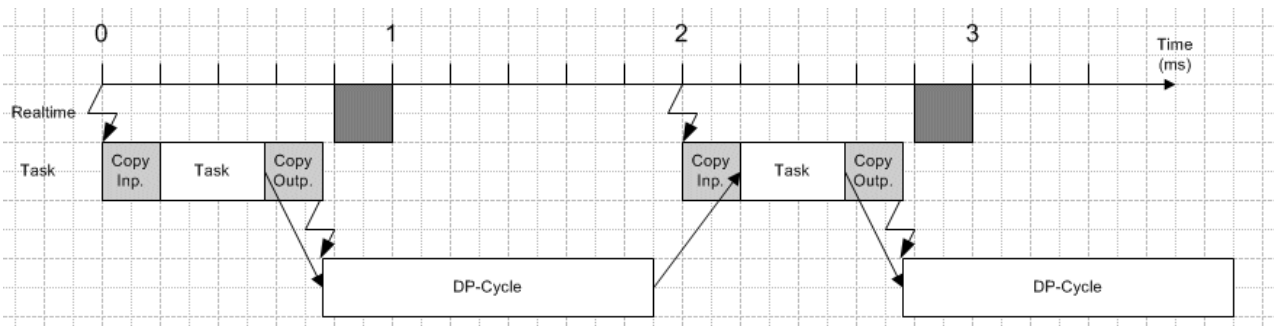
Dieser Fall wäre auch noch unproblematisch, da der DP-Zyklus rechtzeitig fertig ist.

Wenn die Einstellung **IO at task begin** nicht aktiviert ist, ist der Ablauf etwas kritischer. Das nächste Beispiel zeigt die Auswirkungen.

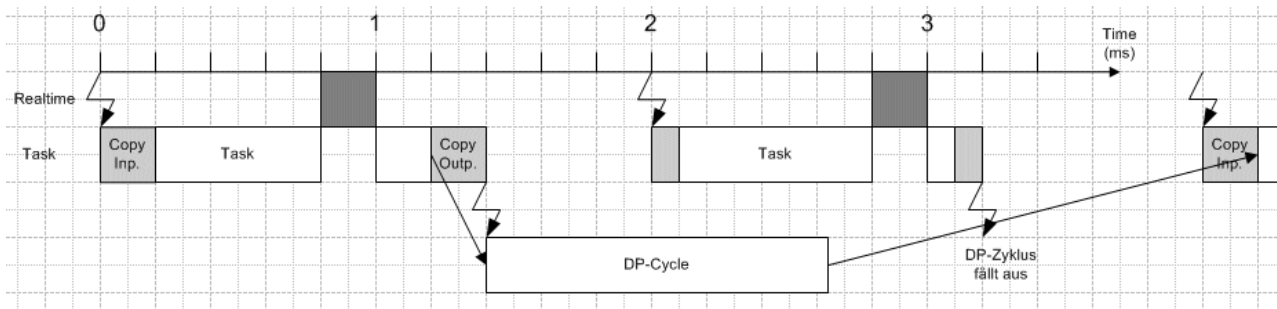
**I/O nicht am Taskanfang**

Wenn bei der Task die Einstellung **I/O at task begin** nicht angewählt ist (Standardeinstellung bei SPS-Task), wird vor dem Start der Task geprüft, ob der vorangegangene DP-Zyklus beendet wurde und die Inputs werden kopiert. Danach erfolgt die Taskbearbeitung. Am Ende der Task werden die Outputs kopiert und der DP-Zyklus gestartet.

Beispiel: Task-Zykluszeit 2 ms, Echtzeit-Ressourcen 80%.



Da sich in diesem Fall die Task und der PROFIBUS die Bandbreite teilen müssen, wirkt sich eine Überschreitung der Echtzeit-Ressourcen wesentlich stärker aus als beim Fall "I/O am Taskanfang":



Der DP-Zyklus fängt später an und ist nicht mehr rechtzeitig vor dem nächsten Zyklus fertig. Es wird erkannt, dass der letzte DP-Zyklus nicht beendet werden konnte. Vor dem Start der nächsten Task, werden keine Inputs (die Task rechnet mit den alten Inputs) und nach der abgearbeiteten Task keine Outputs kopiert. Ebenso wird der PD-Zyklus nicht neu gestartet und somit ein DP-Zyklus ausgelassen.

Mit der Variable **CycleCounter** wird ein ausgefallener DP-Zyklus angezeigt (siehe: Diagnose [Master \[► 70\]](#)).

### Gegenüberstellung I/O am Taskanfang und I/O nicht am Taskanfang

Der Vorteil bei der Einstellung **I/O at task begin** liegt darin, dass die Task und der PD-Zyklus sich die zur Verfügung stehende Bandbreite nicht teilen müssen. Der DP-Zyklus startet sehr konstant (Jitter entspricht dem TwinCAT-Jitter). Ist die Einstellung **I/O at task begin** nicht aktiviert, kann es leicht passieren, dass ein DP-Zyklus ausgelassen wird und die zeitliche Konstanz des DP-Zyklus zusätzlich vom Jitter der Taskabarbeitung abhängt.

Der Nachteil der Einstellung **I/O at task begin** liegt darin, dass die Totzeit also die System-Reaktionszeit größer wird.

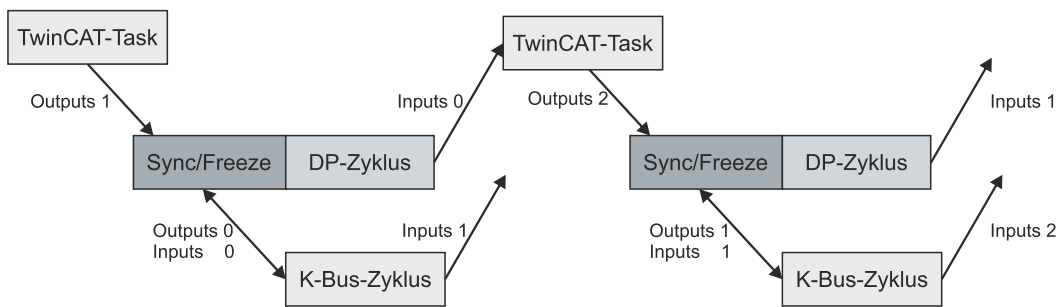
**Sync/Freeze-Funktionalität**

Mit Sync will man bei mehreren Slaves gleichzeitig Ausgänge ausgeben, mit Freeze will man bei mehreren Slaves gleichzeitig Eingänge einlesen.

Der Ablauf in TwinCAT ist der folgende:

- Am Anfang (I/O am Taskanfang) bzw. Ende (I/O nicht am Taskanfang) des Task-Zyklus werden die Ausgänge geschrieben
- Dadurch wird der PROFIBUS-Zyklus gestartet
- Am Anfang des PROFIBUS-Zyklus wird ein Sync/Freeze-Telegramm gesendet
- Dadurch starten die Buskoppler einen K-Bus-Zyklus mit den Ausgängen aus dem letzten Task-Zyklus und übergeben die Eingänge aus dem letzten K-Bus-Zyklus
- Dann sendet der Master zu jedem Slave die aktuellen Ausgänge und holt die übergebenen Eingänge
- Am Anfang des nächsten Task-Zyklus werden die Eingänge gelesen
- usw.

Ausgänge und Eingänge sind also immer einen Zyklus alt.



Für einen Master der im Sync/Freeze-Modus betrieben werden soll, muss auf der Registerkarte **CCAT PBM** unter **Operation Mode** die Option **DP/MC (Equidistant)** eingestellt werden (siehe: [CCAT PBM \[▶ 25\]](#)).

Für einen Slave der im Sync/Freeze-Modus betrieben werden soll, muss auf der Registerkarte **Profibus** die Option **Sync/Freeze enable** angeklickt werden (siehe: [Profibus \[▶ 31\]](#)). Der Master verwendet für die Synchronisierung per Sync/Freeze immer die Gruppe 1.

## 5.2 Mit TwinCAT 2 parametrieren

In diesem Kapitel wird beispielhaft gezeigt, wie Sie Profibus-Geräte mit Hilfe von TwinCAT 2 parametrieren können. Für das Beispiel werden insgesamt drei Geräte verwendet, darunter ein Profibus-Master an dem zwei Profibus-Slaves angeschlossen sind.

Zuerst wird gezeigt, wie Sie ein Zielsystem in TwinCAT suchen und auswählen können. Als nächstes wird ein Profibus-Slave in TwinCAT eingefügt, parametriert und die Profibus-Adresse des Slaves eingestellt. Danach wird ein kleines PLC-Projekt erstellt und in TwinCAT angefügt. Anschließend werden die Variablen aus dem PLC-Projekt mit der Hardware verknüpft und die fertige Konfiguration auf den Profibus-Slave geladen.

Als letzten Schritt wird der Profibus-Master in TwinCAT angefügt und nach den beiden Profibus-Slaves über den Master gesucht. Anschließend wird gezeigt, wie die Profibusvernetzung getestet werden kann.

### 5.2.1 Zielsysteme suchen

Bevor Sie mit den Geräten arbeiten können, müssen Sie Ihren lokalen Rechner mit dem Zielgerät verbinden. Danach können Sie mit Hilfe der IP-Adresse oder dem Host Namen nach den Geräten suchen.

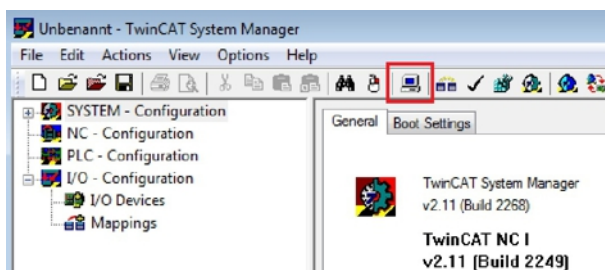
Der lokale PC und die Zielgeräte müssen mit dem gleichen Netzwerk oder direkt über ein Ethernet Kabel miteinander verbunden werden. In TwinCAT kann auf diese Weise nach allen Geräten gesucht und anschließend projektiert werden.

Voraussetzungen für diesen Arbeitsschritt:

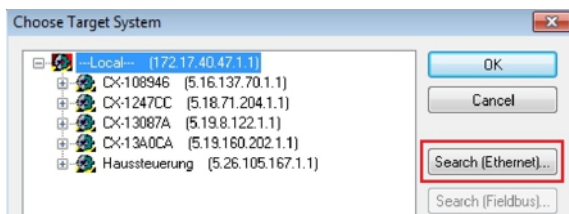
- TwinCAT 2 muss sich im Config Mode befinden.
- IP-Adresse oder Host Name des Gerätes. Der Hostname setzt sich aus CX- und den letzten 3 Byte der Mac Adresse zusammen. Die MAC Adresse befindet sich auf der Seite des Geräts.

Suchen Sie nach den Geräten wie folgt:

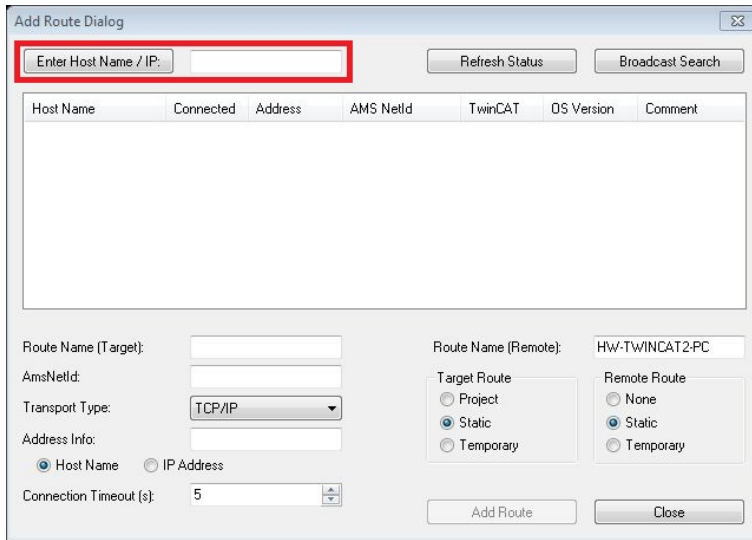
1. Klicken Sie oben im Menü auf **File > New**.
2. Klicken Sie oben in der Symbolleiste auf **Choose Target System**.



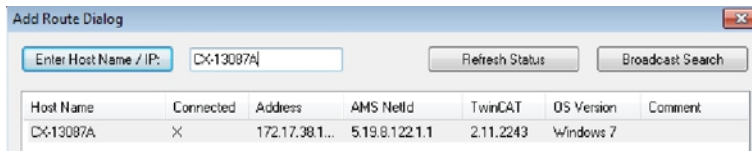
3. Klicken Sie auf **Search (Ethernet)**.



4. Tippen Sie im Feld **Enter Host Name / IP** den Host Namen oder die IP-Adresse des Gerätes ein und drücken Sie **[Enter]**.

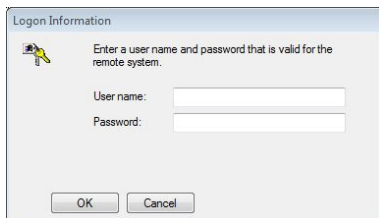


5. Markieren Sie das gefundene Gerät und klicken Sie auf **Add Route**.



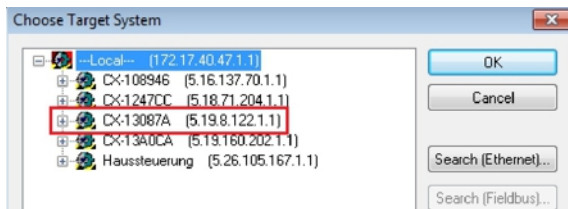
Das Fenster Logon Information erscheint.

6. Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein und klicken Sie auf **OK**.



Als Standard ist bei den CXen folgende Information eingestellt:  
**User name:** Administrator **Password:** 1

7. Klicken Sie auf **Close**, wenn Sie keine weiteren Geräte suchen wollen und schließen damit das Add Route Fenster.  
 Das neue Gerät wird im Fenster Choose Target System angezeigt.
8. Markieren Sie das das Gerät welches Sie als Zielsystem festlegen wollen und klicken Sie auf **OK**.



- ⇒ Sie haben erfolgreich in TwinCAT nach einem Gerät gesucht und das Gerät als Zielsystem eingefügt. In der Ecke unten rechts wird das neue Zielsystem mit Host Name und IP-Adresse (AMS Net ID) angezeigt.

**CX-13087A (5.19.8.122.1.1) Config Mode**

Mit dieser Vorgehensweise können Sie nach allen verfügbaren Geräten suchen und auch jederzeit zwischen den Zielsystemen wechseln. Als nächstes können Sie das Gerät in TwinCAT in die Strukturansicht anfügen.

## 5.2.2 Profibus Slave anfügen

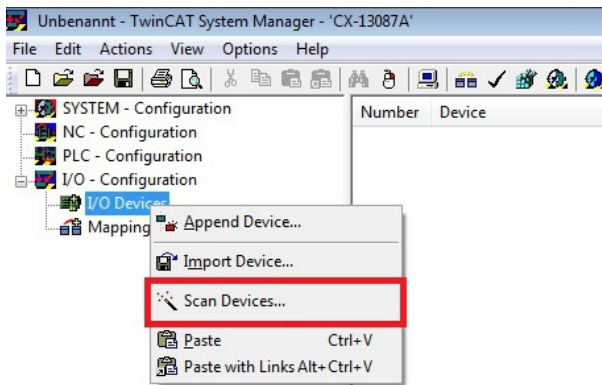
In dem exemplarischen Aufbau ist ein Profibus-Slave CX2020 mit Feldbusmodul CX2500-B310 am Profibus-Master angeschlossen. Damit der Profibus-Slave konfiguriert und später vom Profibus-Master mit allen Ein- und Ausgängen erkannt wird, muss der Profibus-Slave zuerst in TwinCAT angefügt werden.

Voraussetzungen für diesen Arbeitsschritt:

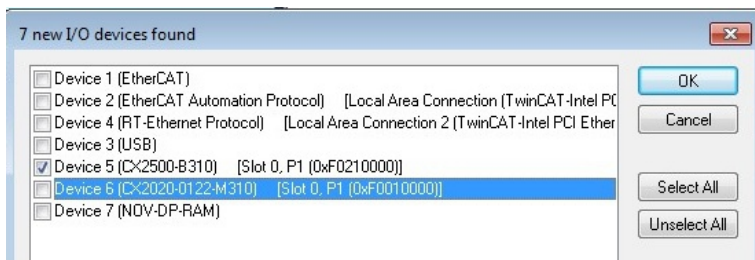
- Ein gescanntes und ausgewähltes Zielgerät mit Profibus-Slave. In diesem Beispiel ist es der CX2020 mit Feldbusmodul CX2500-B310.

**Fügen Sie den Profibus Slave wie folgt ein:**

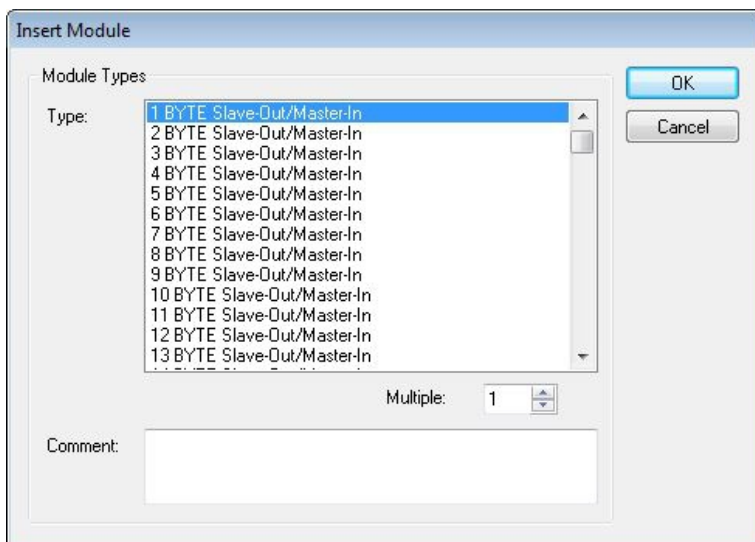
1. Starten Sie den System Manager.
2. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **I/O Devices**.
3. Klicken Sie im Kontextmenü auf **Scan Devices**.



4. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



5. Bestätigen Sie die Anfrage mit **Ja**, um nach Boxen zu suchen. Die Box 1 (CX2500-B310) wird eingebunden. Das Fenster **Insert Module** erscheint.
6. Fügen Sie Module wie z.B. **1 BYTE Slave-Out/Master-In** und **1 BYTE Slave-In/Master-Out** für Ihr Prozessabbild ein.

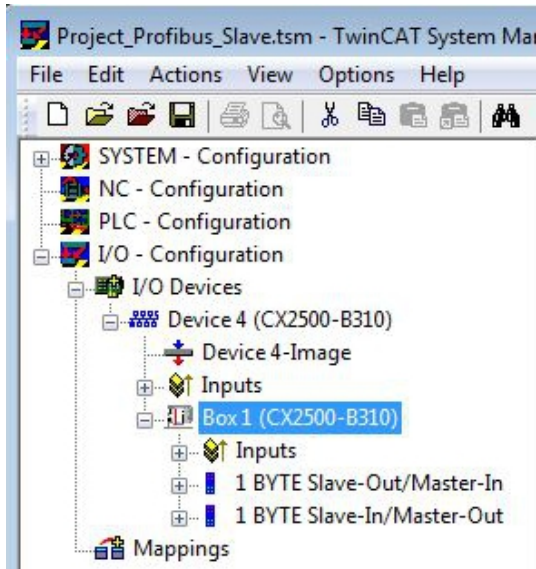




7. Klicken Sie auf **Cancel**, um das Fenster **Insert Module** zu schließen.
8. Klicken Sie bei der Anfrage, ob FreeRun aktiviert werden soll, auf **Ja**.

⇒ Der Profibus-Slave wurde erfolgreich in TwinCAT 2 angefügt und wird in der Strukturansicht mit den Ein- und Ausgängen angezeigt.

Sie können weitere Variablen einfügen, indem Sie mit der rechten Maustaste auf die Box klicken und dann im Kontextmenü auf Append Module klicken.



Im nächsten Schritt können Sie das Prozessabbild erweitern, indem Sie zusätzliche virtuelle Slaves anlegen. Oder Sie können die Adresse einstellen, wenn Sie den Slave fertig konfiguriert haben.

### 5.2.3 Virtuellen Slave anlegen

Es können zusätzliche virtuelle Slaves auf der gleichen Hardware Schnittstelle angelegt werden. Dadurch können mehr Daten mit einem Profibus-Master ausgetauscht oder eine Verbindung mit einem zweiten Profibus-Master angelegt werden.

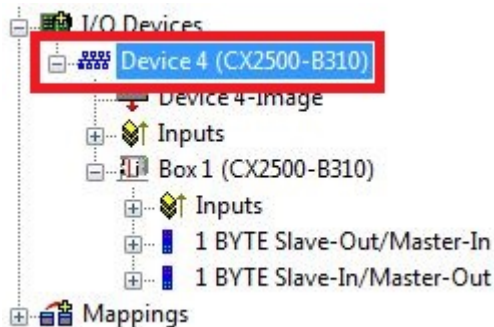
Jeder virtuelle Slave bekommt über TwinCAT eine eigene Adresse und wird für den Profibus-Master wie ein eigenständiges Gerät konfiguriert.

Voraussetzungen für diesen Arbeitsschritt:

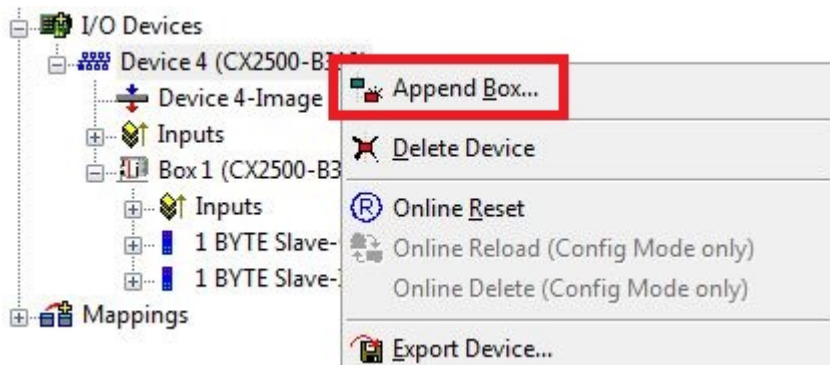
- Ein Profibus-Slave angelegt in TwinCAT.

**Legen Sie einen virtuellen Slave wie folgt ein:**

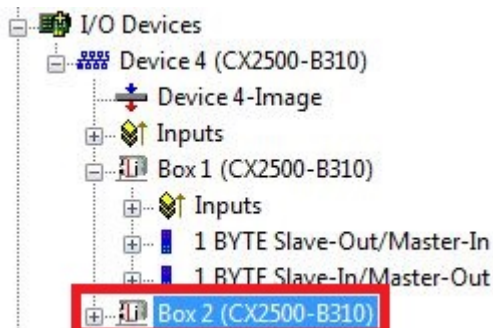
1. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf einen Profibus-Slave.



2. Klicken Sie im Kontextmenü auf **Append Box**.



⇒ Eine weitere Box (virtueller Slave) wird angelegt.



Für den virtuellen Slave können jetzt eigene Variablen angelegt werden. Im nächsten Schritt können Sie die Adresse für den Slave einstellen.

## 5.2.4 Adresse einstellen

Nachdem der Profibus-Slave erfolgreich in TwinCAT angefügt wurde, kann die Adresse des Profibus-Slaves eingestellt werden. Geräte mit einem Dip-Schalter haben eine voreingestellte Adresse. Die Adresse auf dem Dip-Schalter muss mit der in TwinCAT eingestellten Adresse übereinstimmen.

Bei Geräten ohne Dip-Schalter wird die Adresse nur in TwinCAT eingestellt.

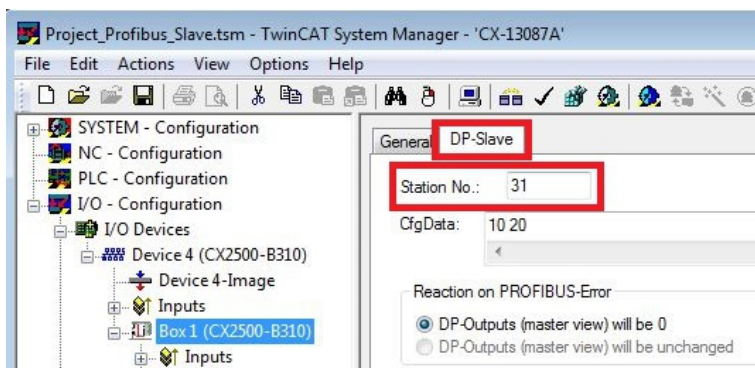
In diesem Arbeitsschritt wird die Adresse in TwinCAT eingestellt, damit der Profibus-Slave über diese Adresse für den Profibus Master erreichbar ist.

Voraussetzungen für diesen Arbeitsschritt:

- Ein eingefügter Profibus Slave in TwinCAT.

### Parametrieren Sie den Profibus-Slave wie folgt:

1. Klicken Sie auf eine Slave Box.
2. Klicken Sie auf den Registerkarte **DP-Slave**.
3. Tippen Sie im Feld **Station No** einen Wert für die Profibus-Adresse ein, z.B. „31“.



- ⇒ Sie haben erfolgreich die Adresse eingestellt. Mit der eingestellten Adresse, ist der Profibus-Slave für den Profibus-Master erreichbar.  
 Als nächstes können Sie ein PLC-Projekt für den Profibus-Slave erstellen.

## 5.2.5 PLC-Projekt erstellen

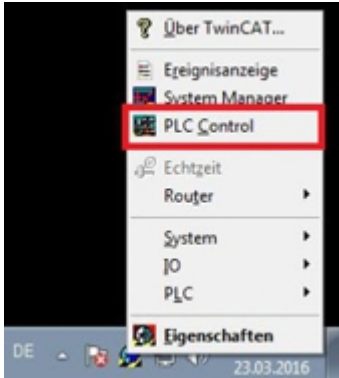
Mit PLC Control wird ein PLC-Projekt erstellt. In den nächsten Schritten wird beschrieben, wie Sie ein PLC-Projekt in TwinCAT erstellen und in der Strukturansicht einfügen.

Voraussetzungen für diesen Arbeitsschritt:

- Ein Embedded-PC, eingefügt in TwinCAT.

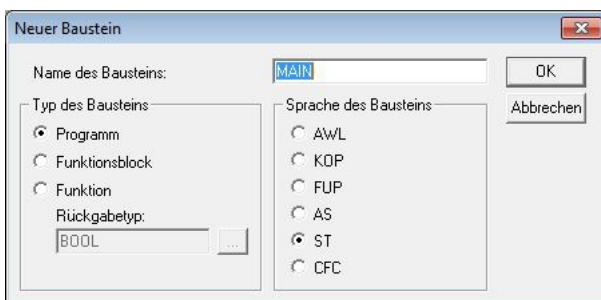
**Erstellen Sie ein PLC-Projekt wie folgt:**

1. Klicken Sie im Startmenü mit rechter Maustaste auf das TwinCAT Symbol.
2. Klicken Sie im Kontextmenü auf **PLC Control**.

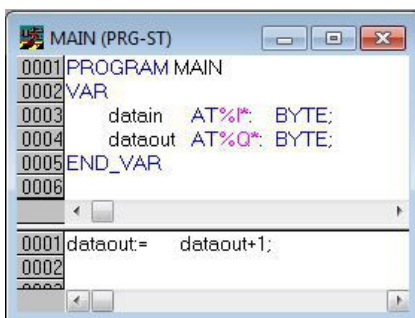


Das Fenster TwinCAT PLC Control erscheint.

3. Klicken Sie im Menü auf **File > New** und wählen Sie die Option **PC oder CX (x86)**.
4. Wählen Sie unter **Typ des Bausteins** die Option **Programm** und unter Sprache des Bausteins die Option **ST (strukturierter Text)**.



5. Schreiben Sie ein kleines Programm.



6. Speichern Sie das PLC-Projekt und klicken Sie im Menü auf **Projekt > Übersetzen**.

⇒ Nachdem das Projekt übersetzt wurde, wird eine Datei mit der Endung .tpy am gleichen Speicherort wie die Projektdatei erzeugt. Der Dateiname der neu erzeugten Datei ist identisch mit dem Dateinamen des PLC-Projekts.

Im nächsten Schritt können Sie das übersetzte PLC-Projekt im TwinCAT System Manager anfügen.

**PLC-Projekt anfügen**

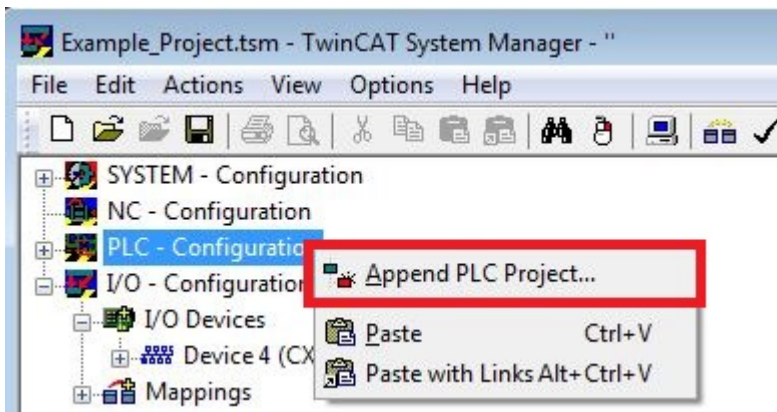
Das PLC-Projekt kann im System Manager angefügt werden. Die neu angelegten Variablen aus einem PLC-Projekt werden im System Manager eingebunden und können mit den Ein- und Ausgängen der Hardware verknüpft werden.

Voraussetzungen für diesen Arbeitsschritt:

- Ein Embedded-PC, eingefügt in TwinCAT.
- Ein fehlerfrei übersetztes PLC-Projekt und eine .tpy-Datei.

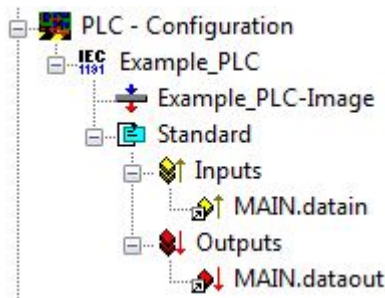
**Gehen Sie wie folgt vor:**

1. Wechseln Sie wieder zum Fenster System Manager.
2. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf **PLC – Configuration**.
3. Klicken Sie im Kontextmenü auf **Append PLC Project**.



4. Wählen Sie eine Datei mit der Dateiendung .tpy in Ihrem Systemverzeichnis und bestätigen Sie mit **OK**.

Das PLC-Projekt wird in der Strukturansicht unter PLC – Configuration eingefügt. Die im Projekt definierten Variablen werden unter den Ein- und Ausgängen angezeigt.



Im nächsten Schritt können Sie die Variablen mit der Hardware verknüpfen.

## 5.2.6 Variablen verknüpfen

Wurde das PLC-Projekt erfolgreich im System Manager angefügt, können die Sie die neu angelegten Ein- und Ausgangsvariablen aus dem PLC-Projekt mit den Ein- und Ausgängen Ihrer Geräte verknüpfen.

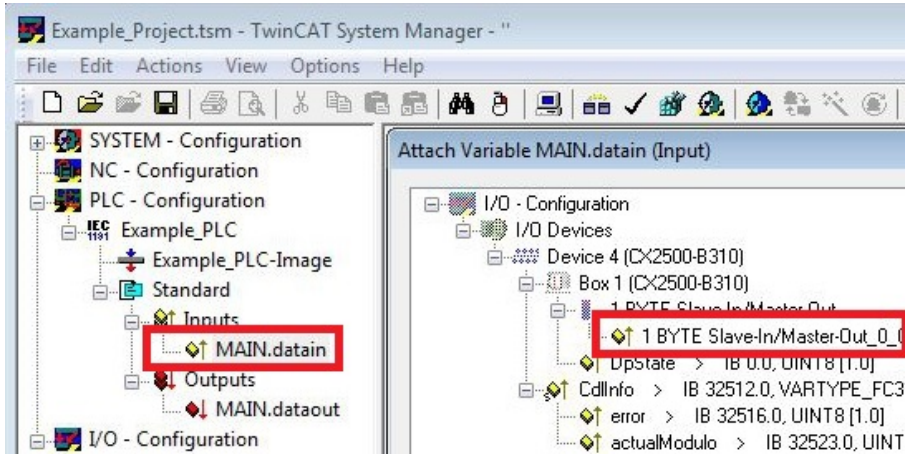
Voraussetzungen für diesen Arbeitsschritt:

- Ein angefügtes PLC-Projekt im System Manager.

**Verknüpfen Sie die Variablen wie folgt:**

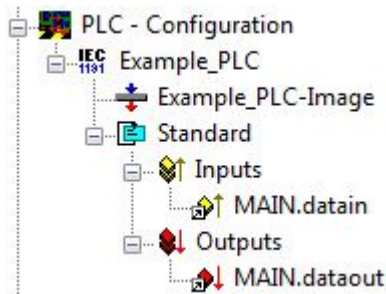
1. Klicken Sie doppelt auf die Ein- bzw. Ausgangsvariablen in der Strukturansicht unter **PLC - Configuration**.

Das Fenster **Attach Variable** erscheint und zeigt an, welche Eingänge bzw. Ausgänge mit Variablen verknüpft werden können.



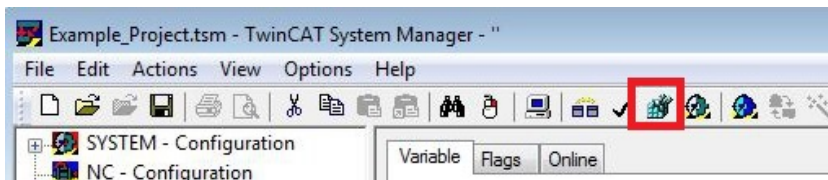
2. Klicken Sie doppelt im Fenster Attach Variable auf die Ein bzw. Ausgänge.

Die Eingangsvariablen werden damit mit den Eingängen und die Ausgangsvariablen mit den Ausgängen Ihrer Hardware verknüpft.



Bereits verknüpfte Variablen werden in TwinCAT mit einem kleinen Pfeilsymbol markiert.

3. Klicken Sie in der Symbolleiste auf **Activate Configuration**.



4. Bestätigen Sie die Anfrage, ob TwinCAT im Free Run Modus gestartet werden soll, mit **Ja**.

⇒ Sie haben erfolgreich Variablen mit der Hardware verknüpft. Mit Activate Configuration wird die aktuelle Konfiguration gesichert und aktiviert.

Als nächstes kann die Konfiguration auf den CX geladen werden, um TwinCAT automatisch im Run Modus und dann das PLC- Projekt zu starten.

## 5.2.7 Konfiguration auf CX laden

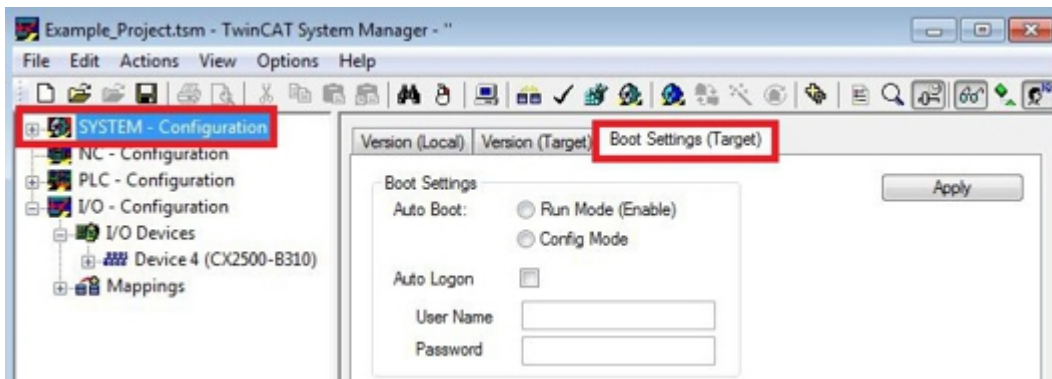
Wenn alle Variablen verknüpft sind, kann die Konfiguration gespeichert und auf den CX geladen werden. Das hat den Vorteil, dass das PLC-Projekt automatisch geladen und gestartet werden kann, wenn der CX eingeschaltet wird. Der Start des zuvor erstellten PLC-Projekts kann damit automatisiert werden.

Voraussetzungen für diesen Arbeitsschritt:

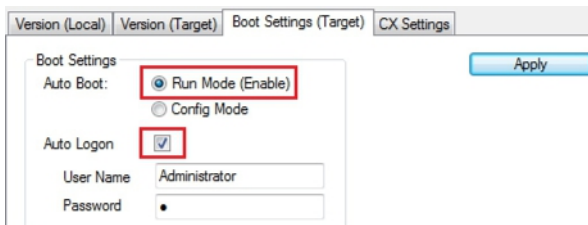
- Ein fertiges und im System Manager angefügtes PLC-Projekt.
- Variablen aus dem PLC-Projekt verknüpft mit der Hardware im System Manager.
- Ein CX ausgewählt als Zielsystem.

Laden Sie die Konfiguration wie folgt auf den CX:

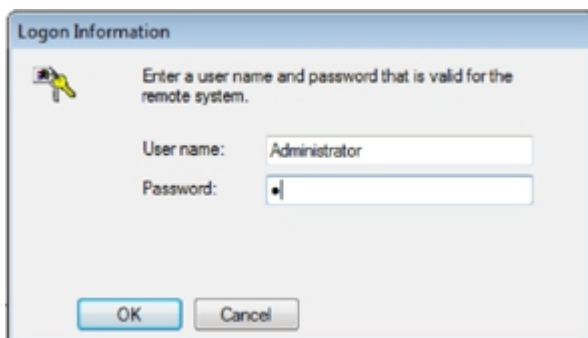
1. Klicken Sie links in der Strukturansicht auf **SYSTEM – Configuration**.
2. Klicken Sie auf die Registerkarte **Boot Settings (Target)**.



3. Wählen Sie unter Boot Settings die Option **Run Mode (Enable)** und Aktivieren Sie das Kontrollkästchen **Auto Logon**.



4. Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein.
5. Klicken Sie auf **Apply**.  
Das Fenster Logon Information erscheint.

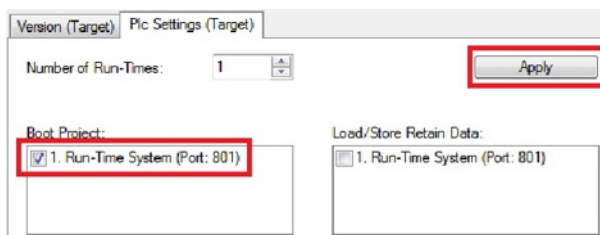


6. Geben Sie den Benutzernamen und das Passwort erneut ein und klicken Sie auf **OK**.

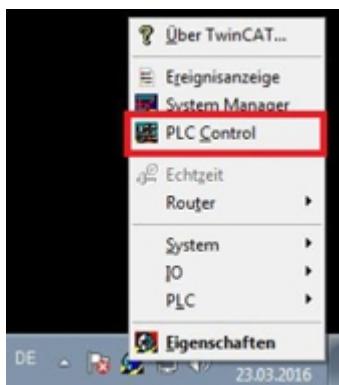
7. Klicken Sie links in der Strukturansicht auf **PLC – Configuration** und dann auf die Registerkarte **Pic Settings (Target)**.



8. Wählen Sie die Start PLC unter Boot Project und klicken Sie auf **Apply**.

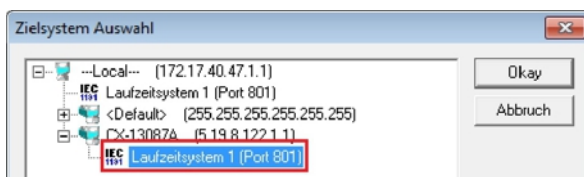


9. Starten Sie PLC-Control und öffnen Sie das PLC-Projekt.



10. Klicken Sie oben in der Menüleiste auf **Online** und dann auf **Choose Run- Time System**.

11. Wählen Sie das Laufzeitsystem von dem CX aus und klicken Sie auf **Okay**.



12. Klicken Sie oben in der Menüleiste auf **Online** und dann auf **Login**. Das PLC-Projekt wird eingeloggt.

13. Klicken Sie oben in der Menüleiste auf **Online** und dann auf **Create Bootproject**.

⇒ Sie haben erfolgreich die Konfiguration auf den CX geladen. Ab jetzt wird bei jedem Start TwinCAT im Run Mode und das PLC-Projekt gestartet.

Als nächstes kann der Master in einem neuen Projekt im System Manager angefügt und über den Master nach den fertig eingerichteten Slaves gesucht werden.



## 5.2.8 Profibus Master anfügen

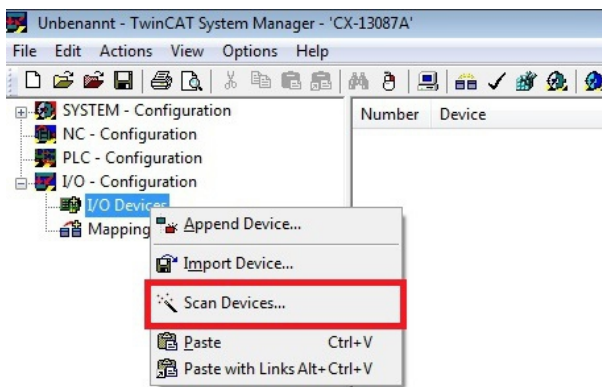
Der Profibus-Master wird wie die anderen Geräte mit dem TwinCAT System Manager eingefügt. Über den angefügten Master kann dann nach allen angeschlossenen Slaves gesucht werden. Im Folgenden wird gezeigt, wie Sie einen Profibus-Master in TwinCAT anfügen.

Voraussetzungen für diesen Arbeitsschritt:

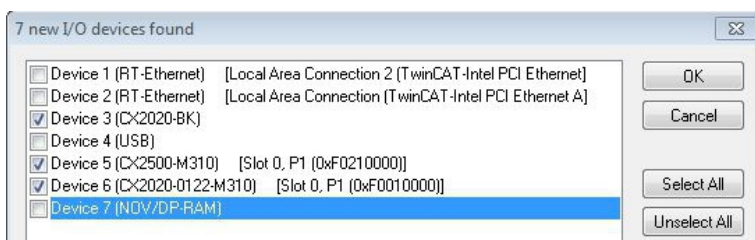
- TwinCAT muss sich im Config Mode befinden.
- Ein ausgewähltes Zielsystem (in diesem Beispiel ist es der Embedded-PC CX2020-M310)

**Füge Sie einen Profibus-Master wie folgt an:**

1. Starten Sie den System Manager.
2. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **I/O Devices**.
3. Klicken Sie im Kontextmenü auf **Scan Devices**.

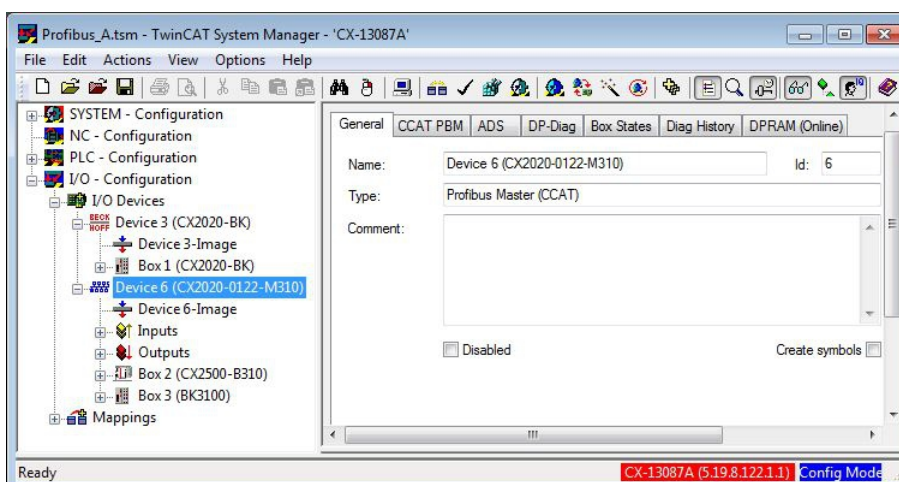


4. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



5. Bestätigen Sie die Anfrage, ob nach Boxen gesucht werden soll, mit **Ja**.

⇒ Alle gefundenen Geräte und Slave Boxen werden links in der Strukturansicht angezeigt. Auch die Busklemmen, die an den Geräten oder Slave Boxen angeschlossen sind, werden angezeigt.



Wiederholen Sie die Arbeitsschritte, wenn nicht alle Geräte angezeigt werden. Sollten Sie trotz Wiederholung nicht alle Geräte und Slave Boxen finden, müssen Sie die Verkabelung der Geräte und Slave Boxen überprüfen.

Im nächsten Schritt können Sie die Verbindung zwischen Profibus Master und Profibus Slave testen.

### 5.2.9 Profibusvernetzung testen

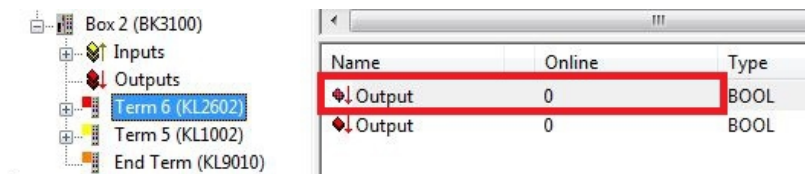
Die Profibusvernetzung zwischen einem Profibus-Master und Profibus-Slave kann mit TwinCAT getestet werden. Für den Test wird als Profibus Slave der Buskoppler (BK3100) gewählt. Während des Tests wird eine LED an einer Busklemme geschaltet, die am Profibus Slave angesteckt ist.

Voraussetzungen für diesen Arbeitsschritt:

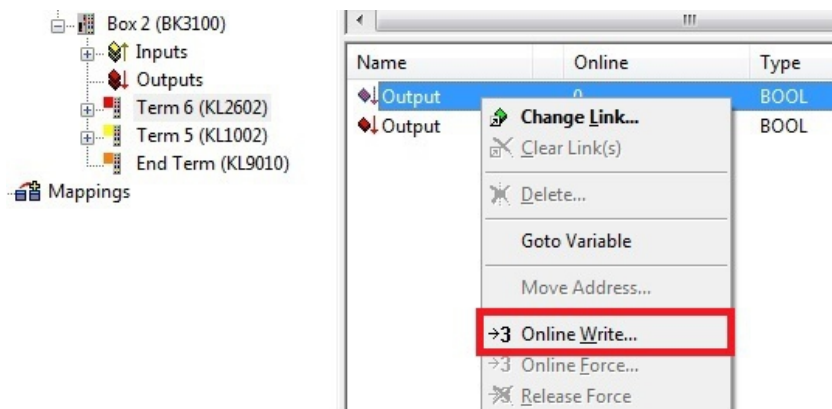
- TwinCAT muss sich im Free Run Modus befinden.

**Testen Sie die Vernetzung wie folgt:**

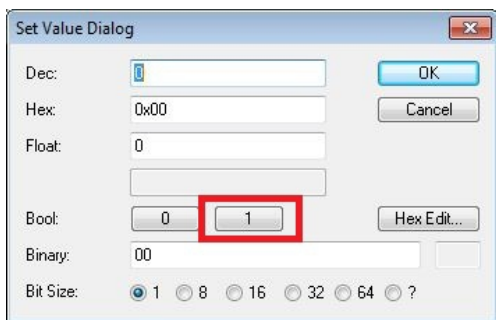
1. Suchen Sie den Profibus-Slave links in der Strukturansicht.
2. Klicken Sie auf die Busklemme, bei der Sie die LED schalten wollen.
3. Klicken Sie mit der rechten Maustaste auf einen Eingang bzw. Ausgang.



4. Klicken Sie im Kontextmenü auf **Online Write**.



5. Setzen Sie den Wert **Bool** auf „1“.



⇒ Unter TwinCAT wird der Wert „1“ unter Online bei dem entsprechenden Eingang bzw. Ausgang angezeigt. Wenn die LED auch auf der Busklemme aufleuchtet, die am Buskoppler angeschlossen ist, wurde ein Signal erfolgreich zum Profibus Slave übertragen. Der Profibus Master und Profibus Slave sind damit erfolgreich miteinander vernetzt.

### 5.2.10 Prozessdaten drehen

Die Prozessdaten werden standardmäßig im Intel Format übertragen. Wenn Sie die Daten im Motorola Format benötigen, müssen die Daten entsprechend gedreht werden. In diesem Arbeitsschritt wird gezeigt, wie die Daten in TwinCAT gedreht werden.

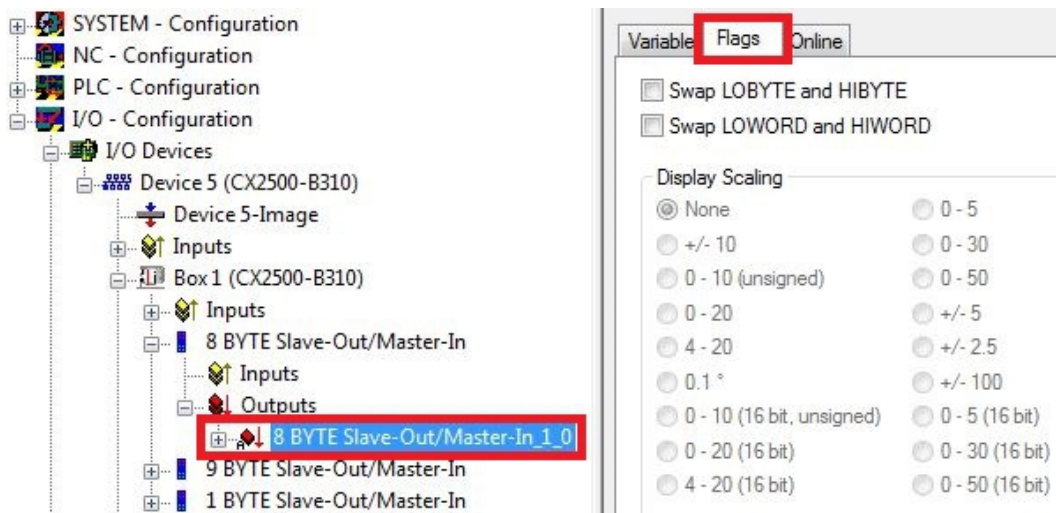
Wenn Sie das Standardformat benötigen, können Sie diesen Arbeitsschritt überspringen.

Voraussetzungen für diesen Arbeitsschritt:

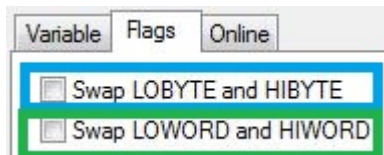
- Ein fertig parametrierter Slave.
- Ein Slave angeschlossen an einem Master. Der Master wird in TwinCAT eingefügt und danach nach dem Slave über den Master gescannt.

**Drehen Sie die Prozessdaten wie folgt:**

1. Klicken Sie rechts in der Strukturansicht auf eine Variable, deren Daten gedreht werden müssen.
2. Klicken Sie auf die Registerkarte **Flags**.



3. Klicken Sie auf die Option, die Sie benötigen. Bei WORD Variablen können Sie nur LOBYTE- und HIBYTE tauschen. Bei DWORDs zusätzlich das WORD.



⇒ Auf diese Weise können Sie Prozessdaten drehen. Benutzen Sie das folgende Beispiel, um zu sehen, wie sich die Daten bei den einzelnen Optionen verändern.  
Beispiel für DWORD.

Daten des Slaves	Daten die der Master empfängt			
Ursprüngliche Daten	Keine Option angewählt	Swap Byte (blau)	Swap Word (grün)	Swap beides (blau und grün)
0x01020304	0x01020304	0x02010403	0x03040102	0x04030201

Die Daten können auch im PLC-Projekt mit dem Befehl ROR gedreht werden.  
Beispiel für ST: VarProfibus:=ROR(VarAnalog,8); (\*Beide Variablen vom Typ WORD\*)

## 5.3 Mit TwinCAT 3 parametrieren

In diesem Kapitel wird beispielhaft gezeigt, wie Sie Profibus-Geräte mit Hilfe von TwinCAT 3 parametrieren können. Für das Beispiel werden insgesamt drei Geräte verwendet, darunter ein Profibus-Master an dem zwei Profibus-Slaves angeschlossen sind.

Zuerst wird gezeigt, wie Sie ein Zielsystem in TwinCAT suchen und auswählen können. Als nächstes wird ein Profibus-Slave in TwinCAT eingefügt, parametriert und die Profibus-Adresse des Slaves eingestellt. Danach wird ein kleines PLC-Projekt erstellt und in TwinCAT angefügt. Anschließend werden die Variablen aus dem PLC-Projekt mit der Hardware verknüpft und die fertige Konfiguration auf den Profibus-Slave geladen.

Als letzten Schritt wird der Profibus-Master in TwinCAT angefügt und nach den beiden Profibus-Slaves über den Master gesucht. Anschließend wird gezeigt, wie die Profibusvernetzung getestet werden kann.

### 5.3.1 Zielsysteme suchen

Bevor Sie mit den Geräten arbeiten können, müssen Sie Ihren lokalen Rechner mit dem Zielgerät verbinden. Danach können Sie mit Hilfe der IP-Adresse oder dem Host Namen nach Geräten suchen.

Der lokale PC und die Zielgeräte müssen mit dem gleichen Netzwerk oder direkt über ein Ethernet Kabel miteinander verbunden werden. In TwinCAT kann auf diese Weise nach allen Geräten gesucht und anschließend projektiert werden.

Voraussetzungen für diesen Arbeitsschritt:

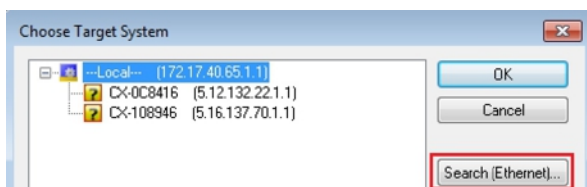
- TwinCAT 3 muss sich im Config Mode befinden.
- IP-Adresse oder Host Name des Gerätes.

**Suchen Sie nach den Geräten wie folgt:**

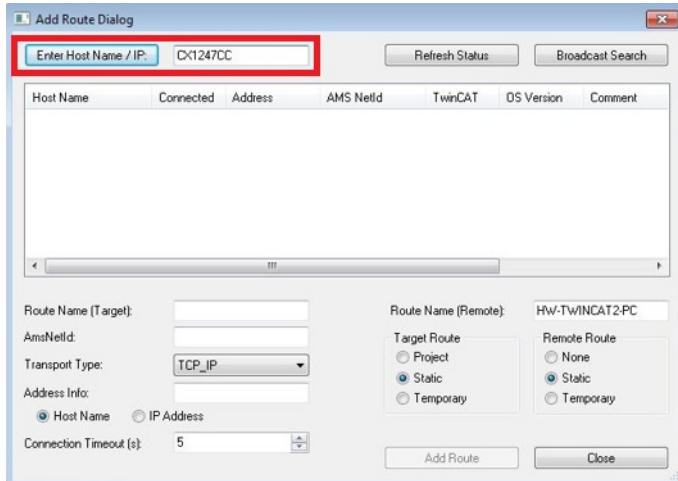
1. Klicken Sie oben im Menü auf **File > New > Project** und erstellen Sie ein neues TwinCAT XAE Projekt.
2. Klicken Sie links in der Strukturansicht auf **SYSTEM** und dann auf **Choose Target**.



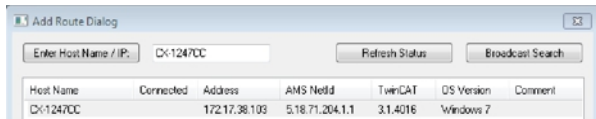
3. Klicken Sie auf **Search (Ethernet)**.



- Tippen Sie im Feld **Enter Host Name / IP** den Host Namen oder die IP-Adresse des Gerätes ein und drücken Sie **[Enter]**.



- Markieren Sie das gefundene Gerät und klicken Sie auf **Add Route**.



Das Fenster Logon Information erscheint.

Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein und klicken Sie auf **OK**.

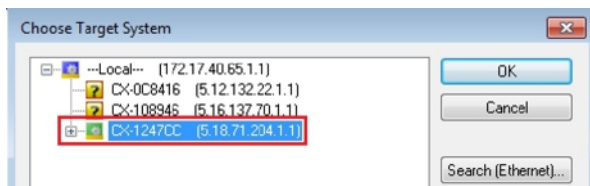


Als Standard ist bei den CXen folgende Information eingestellt:

**User name:** Administrator **Password:** 1

- Klicken Sie auf **Close**, wenn Sie keine weiteren Geräte suchen wollen und schließen damit das Add Route Fenster. Das neue Gerät wird im Fenster Choose Target System angezeigt.

- Markieren Sie das Gerät welches Sie als Zielsystem festlegen wollen und klicken Sie auf **OK**.



⇒ Sie haben erfolgreich in TwinCAT nach einem Gerät gesucht und das Gerät als Zielsystem eingefügt. In der Menüleiste wird das neue Zielsystem mit dem Host Namen angezeigt.



Mit dieser Vorgehensweise können Sie nach allen verfügbaren Geräten suchen und auch jederzeit zwischen den Zielsystemen wechseln. Als nächstes können Sie das Gerät in TwinCAT in die Strukturansicht anfügen.

### 5.3.2 Profibus Slave anfügen

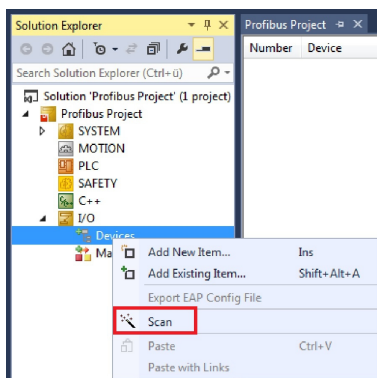
In dem exemplarischen Aufbau ist ein Profibus-Slave CX2020 mit Feldbusmodul CX2500-B310 am Profibus-Master angeschlossen. Damit der Profibus-Slave konfiguriert und später vom Profibus-Master mit allen Ein- und Ausgängen erkannt wird, muss der Profibus-Slave zuerst in TwinCAT angefügt werden.

Voraussetzungen für diesen Arbeitsschritt:

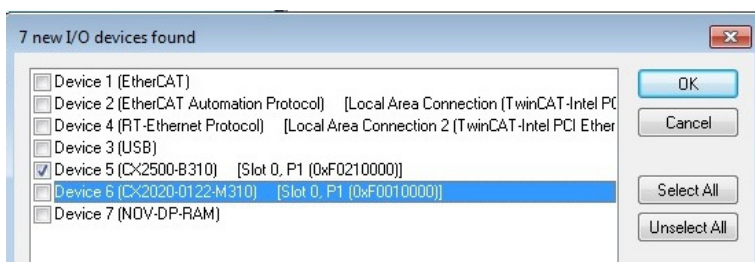
- Ein gescanntes und ausgewähltes Zielgerät mit Profibus-Slave. In diesem Beispiel ist es der CX2020 mit Feldbusmodul CX2500-B310.

**Fügen Sie den Profibus Slave wie folgt ein:**

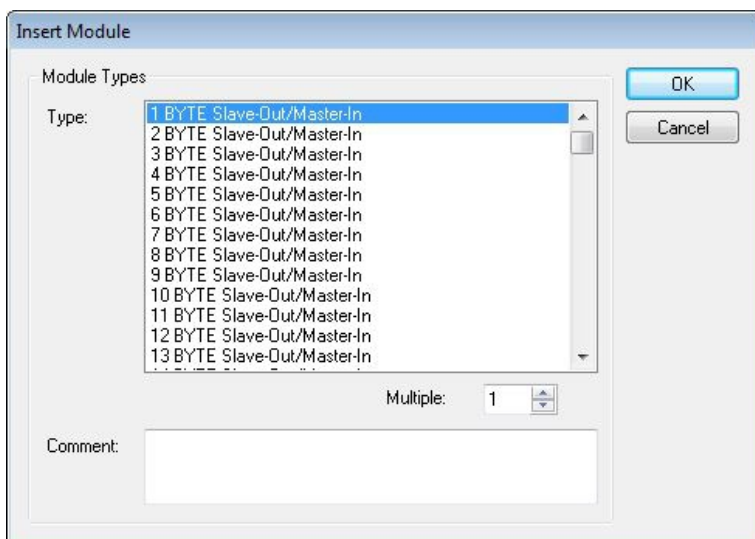
1. Starten Sie TwinCAT und öffnen Sie ein leeres Projekt.
2. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **I/O Devices**.
3. Klicken Sie im Kontextmenü auf **Scan**.



4. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



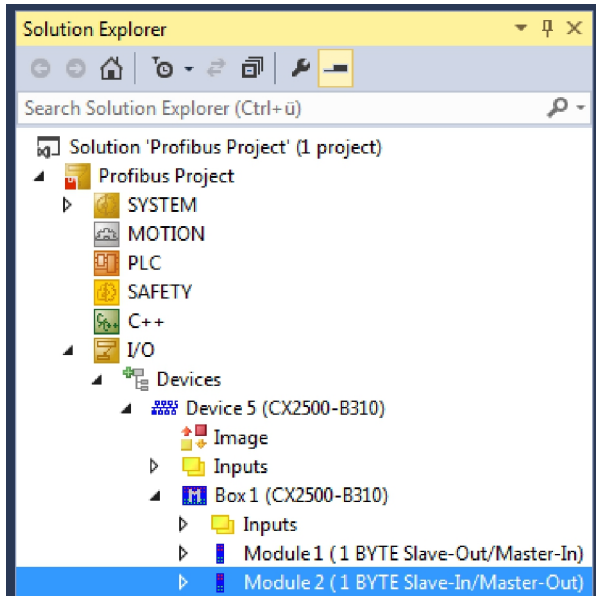
5. Bestätigen Sie die Anfrage mit **Ja**, um nach Boxen zu suchen. Die Box 1 (CX2500-B310) wird eingebunden. Das Fenster **Insert Module** erscheint.
6. Fügen Sie Module wie z.B. **1 BYTE Slave-Out/Master-In** und **1 BYTE Slave-In/Master-Out** für Ihr Prozessabbild ein.



7. Klicken Sie auf **Cancel**, um das Fenster **Insert Module** zu schließen.
8. Klicken Sie bei der Anfrage, ob FreeRun aktiviert werden soll, auf **Ja**.

⇒ Der Profibus Slave wurde erfolgreich in TwinCAT 3 angefügt und wird in der Strukturansicht mit den Ein- und Ausgängen angezeigt.

Sie können weitere Variablen einfügen, indem Sie mit der rechten Maustaste auf die Box klicken und dann im Kontextmenü auf **Add New Item** klicken.



Im nächsten Schritt können Sie das Prozessabbild erweitern, indem Sie zusätzliche virtuelle Slaves anlegen. Oder Sie können die Adresse einstellen, wenn Sie den Slave fertig konfiguriert haben.

### 5.3.3 Virtuellen Slave anlegen

Es können zusätzliche virtuelle Slaves auf der gleichen Hardware Schnittstelle angelegt werden. Dadurch können mehr Daten mit einem Profibus-Master ausgetauscht oder eine Verbindung mit einem zweiten Profibus-Master angelegt werden.

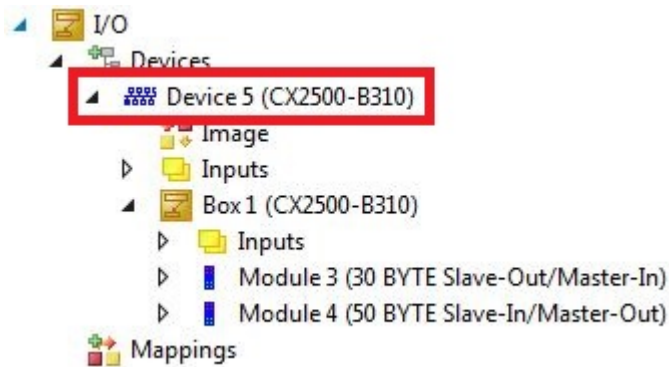
Jeder virtuelle Slave bekommt über TwinCAT eine eigene Adresse und wird für den Profibus-Master wie ein eigenständiges Gerät konfiguriert.

Voraussetzungen für diesen Arbeitsschritt:

- Ein Profibus-Slave angelegt in TwinCAT.

**Legen Sie einen virtuellen Slave wie folgt ein:**

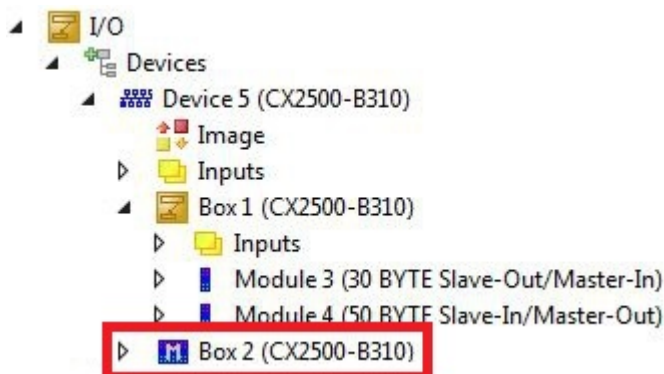
1. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf einen Profibus-Slave.



2. Klicken Sie im Kontextmenü auf **Add New Item**.



⇒ Eine weitere Box (virtueller Slave) wird angelegt.



Für den virtuellen Slave können jetzt eigene Variablen angelegt werden. Im nächsten Schritt können Sie die Adresse für den Slave einstellen.



### 5.3.4 Adresse einstellen

Nachdem der Profibus-Slave erfolgreich in TwinCAT angefügt wurde, kann die Adresse des Profibus-Slaves eingestellt werden. Geräte mit einem Dip-Schalter haben eine voreingestellte Adresse. Die Adresse auf dem Dip-Schalter muss mit der in TwinCAT eingestellten Adresse übereinstimmen.

Bei Geräten ohne Dip-Schalter wird die Adresse nur in TwinCAT eingestellt.

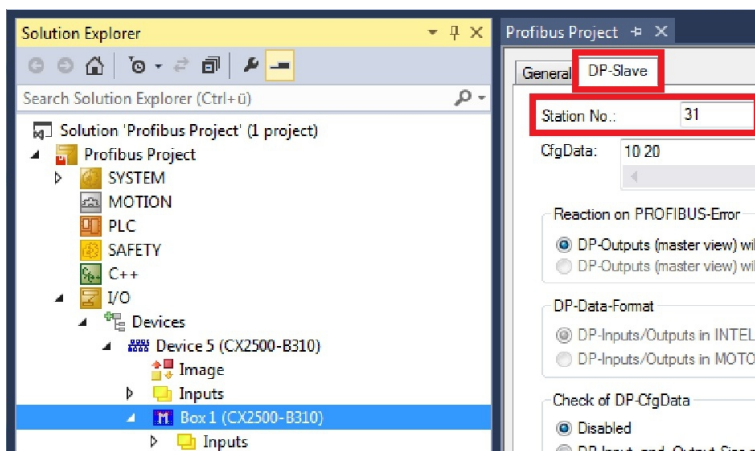
In diesem Arbeitsschritt wird die Adresse in TwinCAT eingestellt, damit der Profibus Slave über diese Adresse für den Profibus Master erreichbar ist.

Voraussetzungen für diesen Arbeitsschritt:

- Ein eingefügter Profibus Slave in TwinCAT.

**Parametrieren Sie den Profibus Slave wie folgt:**

1. Klicken Sie auf eine Slave Box.
2. Klicken Sie auf den Registerkarte **DP-Slave**.
3. Tippen Sie im Feld **Station No** einen Wert für die Profibus-Adresse ein, z.B. „31“.



- ⇒ Sie haben erfolgreich die Adresse eingestellt. Mit der eingestellten Adresse, ist der Profibus-Slave für den Profibus-Master erreichbar.  
Als nächstes können Sie ein PLC-Projekt für den Profibus-Slave erstellen.

### 5.3.5 PLC-Projekt erstellen

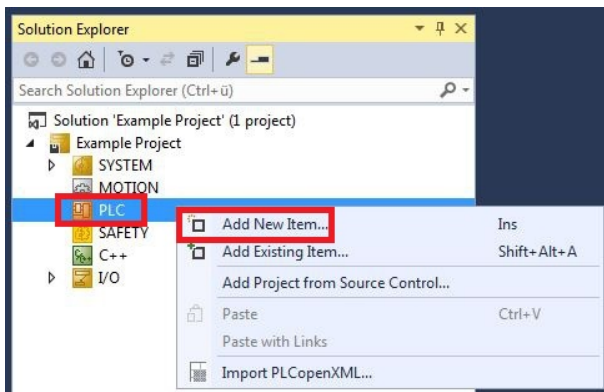
In den nächsten Schritten wird beschrieben, wie Sie ein PLC-Projekt in TwinCAT erstellen und in der Strukturansicht einfügen.

Voraussetzungen für diesen Arbeitsschritt:

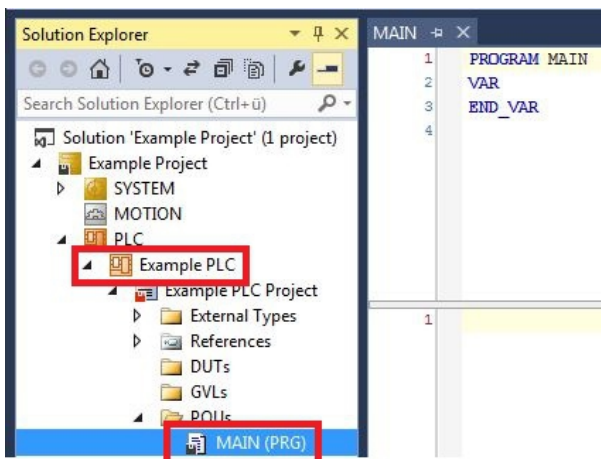
- Ein neu angelegtes TwinCAT XAE Projekt.

**Erstellen Sie ein PLC-Projekt wie folgt:**

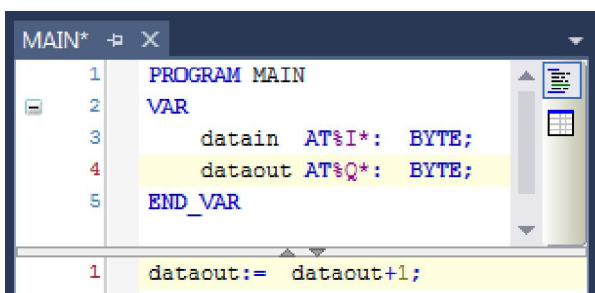
1. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf **PLC**.
2. Klicken Sie im Kontextmenü auf **Add New Item** und wählen Sie das **Standard PLC Project**.



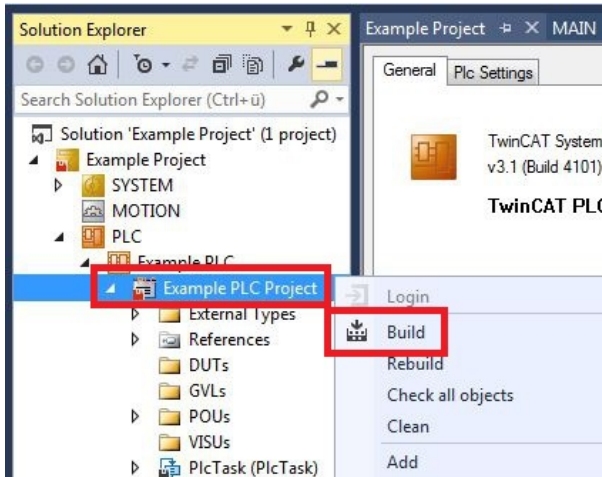
3. Klicken Sie in der Strukturansicht auf das neu erstellte PLC-Projekt und dann unter **POUs** doppelt auf **MAIN (PRG)**.



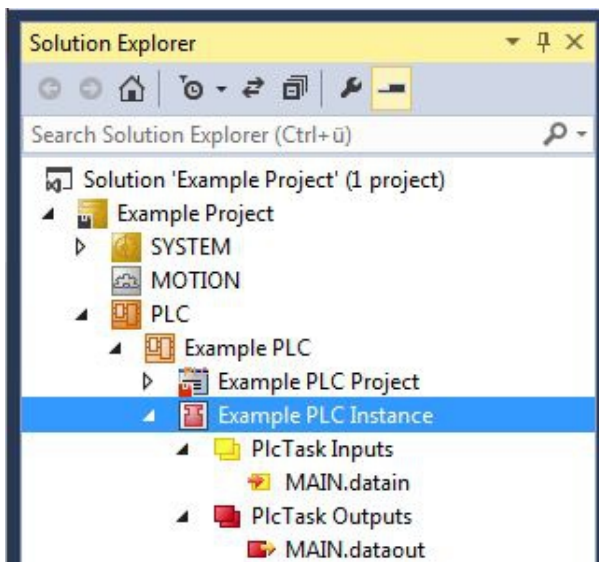
4. Schreiben Sie ein kleines Programm wie im folgenden Bild.



5. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf das PLC-Projekt und dann im Kontextmenü auf **Build**.



- ⇒ Sie haben erfolgreich ein PLC-Projekt erstellt und das Projekt in TwinCAT angefügt. Es wird eine PLC-Instanz mit den Variablen für die Eingänge und Ausgänge aus dem PLC-Projekt erstellt.



Im nächsten Schritt können Sie die Variablen mit der Hardware verknüpfen.

### 5.3.6 Variablen verknüpfen

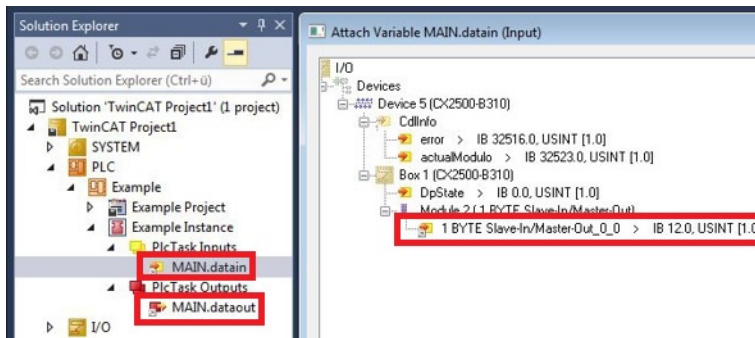
Wurde das PLC-Projekt erfolgreich im System Manager angefügt, können die Sie die neu angelegten Ein- und Ausgangsvariablen aus dem PLC-Projekt mit den Ein- und Ausgängen Ihrer Hardware verknüpfen.

Voraussetzungen für diesen Arbeitsschritt:

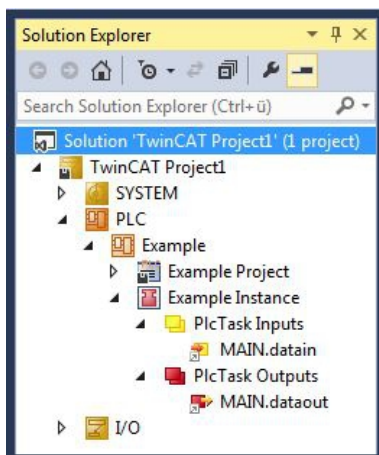
- Ein angefügtes PLC-Programm in TwinCAT.

**Verknüpfen Sie die Variablen wie folgt:**

1. Klicken Sie doppelt auf die Ein- bzw. Ausgangsvariablen in der Strukturansicht unter **PLC**. Das Fenster **Attach Variable** erscheint und zeigt an, welche Eingänge bzw. Ausgänge mit den Variablen aus dem PLC-Projekt verknüpft werden können.



2. Klicken Sie doppelt im Fenster **Attach Variable** auf die Ein bzw. Ausgänge der Hardware. Verknüpfen Sie die Eingangsvariablen mit den Eingängen und die Ausgangsvariablen mit den Ausgängen der Hardware.



Bereits verknüpfte Variablen werden in TwinCAT mit einem kleinen Pfeilsymbol markiert.

3. Klicken Sie in der Symbolleiste auf **Activate Configuration**.



4. Bestätigen Sie die Anfrage, ob TwinCAT im Free Run Modus gestartet werden soll, mit **Ja**.  
⇒ Sie haben erfolgreich Variablen mit der Hardware verknüpft. Mit Activate Configuration wird die aktuelle Konfiguration gesichert und aktiviert.

Als nächstes kann die Konfiguration auf den CX geladen werden, um TwinCAT automatisch im Run Modus und dann das PLC-Projekt zu starten.

### 5.3.7 Konfiguration auf CX laden

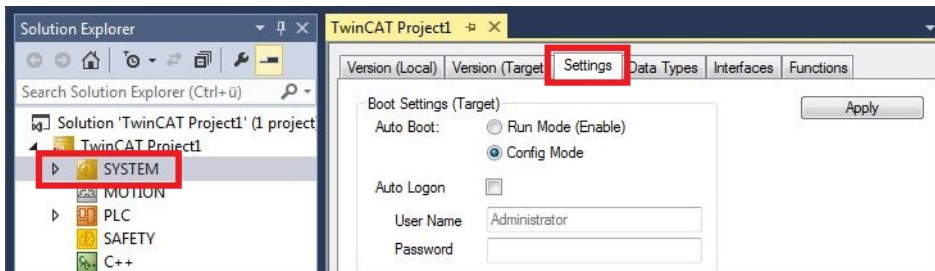
Wenn Variablen verknüpft sind, kann die Konfiguration gespeichert und auf den CX geladen werden. Das hat den Vorteil, dass das PLC-Projekt automatisch geladen und gestartet werden kann, wenn der CX eingeschaltet wird. Der Start des zuvor erstellten PLC-Projekts kann damit automatisiert werden.

Voraussetzungen für diesen Arbeitsschritt:

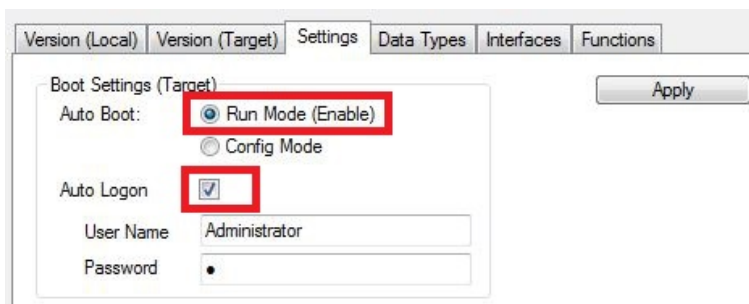
- Ein fertiges und im System Manager angefügtes PLC-Projekt.
- Variablen aus dem PLC-Projekt verknüpft mit der Hardware im System Manager.
- Ein CX ausgewählt als Zielsystem.

**Laden Sie die Konfiguration aus dem System Manager wie folgt auf den CX:**

1. Klicken Sie links in der Strukturansicht auf **SYSTEM**.
2. Klicken Sie auf die Registerkarte **Settings**.

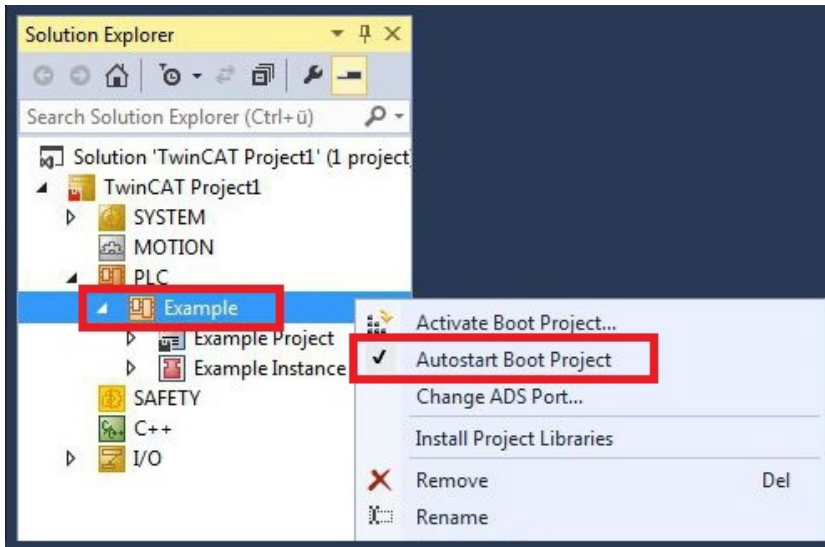


3. Wählen Sie unter Boot Settings die Option **Run Mode (Enable)** und Aktivieren Sie das Kontrollkästchen **Auto Logon**.

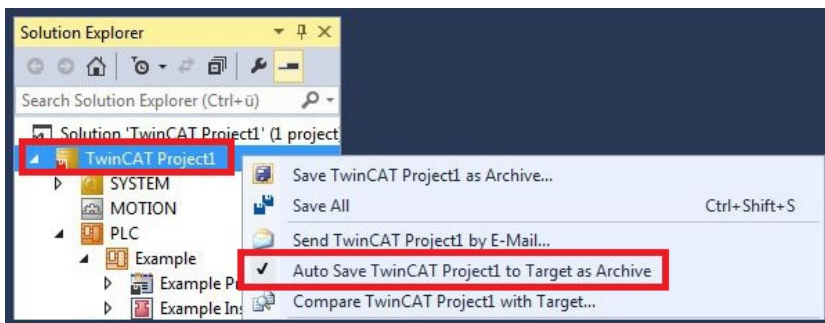


4. Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein.
5. Klicken Sie auf **Apply**.
6. Klicken Sie links in der Strukturansicht unter **PLC** mit der rechten Maustaste auf das PLC-Projekt.

7. Klicken Sie im Kontextmenü auf **Autostart Boot Project**.  
Die Einstellung wird markiert



8. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf den Projektordner.
9. Klicken Sie im Kontextmenü auf **Auto Save to Target as Archive**.  
Die Einstellung wird markiert.



⇒ Sie haben erfolgreich die Konfiguration auf den CX geladen. Ab jetzt wird bei jedem Start TwinCAT im Run Mode und das PLC-Projekt gestartet.

Als nächstes kann der Master in einem neuen Projekt im System Manager angefügt und über den Master nach den fertig eingerichteten Slaves gesucht werden.

### 5.3.8 Profibus Master anfügen

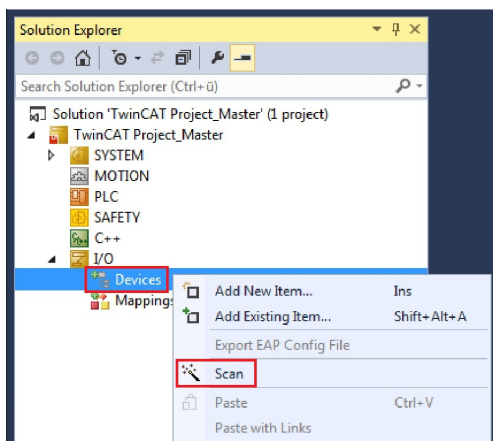
Der Profibus-Master wird wie die anderen Geräte mit dem TwinCAT System Manager eingefügt. Über den angefügten Master kann dann nach allen angeschlossenen Slaves gesucht werden. Im Folgenden wird gezeigt, wie Sie einen Profibus-Master anfügen.

Voraussetzungen für diesen Arbeitsschritt:

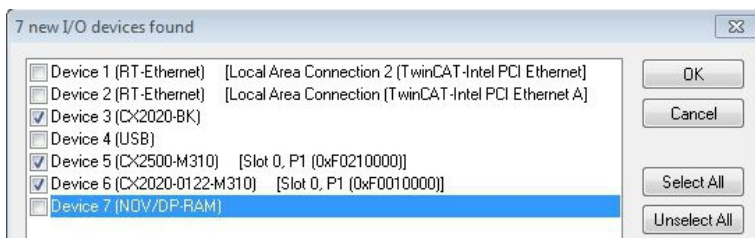
- TwinCAT muss sich im Config Mode befinden.
- Ein ausgewähltes Zielsystem (in diesem Beispiel ist es der Embedded-PC CX2020-M310)

**Fügen Sie ein Profibus-Master wie folgt an:**

1. Starten Sie TwinCAT.
2. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **Devices**.
3. Klicken Sie im Kontextmenü auf **Scan**.

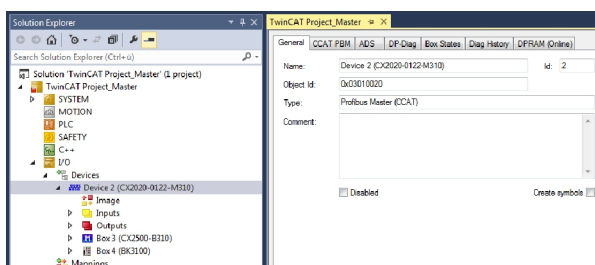


4. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



5. Bestätigen Sie die Anfrage, ob nach Boxen gesucht werden soll, mit **Ja**.

⇒ Alle gefundenen Geräte und Slave Boxen werden links in der Strukturansicht angezeigt. Auch die Busklemmen, die an den Geräten oder Slave Boxen angeschlossen sind, werden angezeigt.



Wiederholen Sie die Arbeitsschritte, wenn nicht alle Geräte angezeigt werden. Sollten Sie trotz Wiederholung nicht alle Geräte und Slave Boxen finden, müssen Sie die Verkabelung der Geräte und Slave Boxen überprüfen.

Im nächsten Schritt können Sie die Verbindung zwischen Profibus Master und Profibus Slave testen.

### 5.3.9 Profibusvernetzung testen

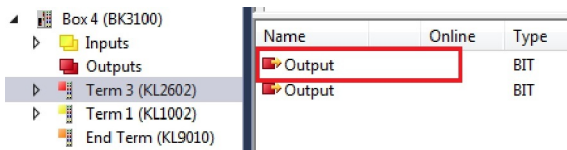
Die Profibusvernetzung zwischen einem Profibus-Master und Profibus-Slave kann mit TwinCAT getestet werden. Für den Test wird als Profibus Slave der Buskoppler (BK3100) gewählt. Während des Tests wird eine LED an einer Busklemme geschaltet, die am Profibus Slave angesteckt ist.

Voraussetzungen für diesen Arbeitsschritt:

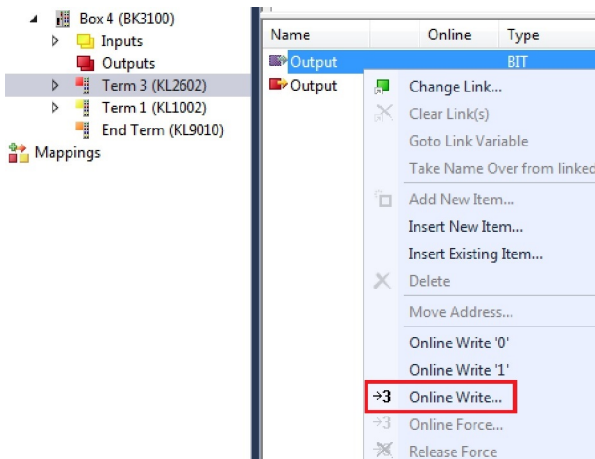
- TwinCAT muss sich im Free Run Modus befinden.

**Testen Sie die Vernetzung wie folgt:**

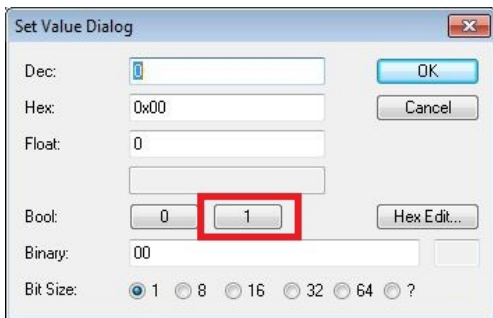
1. Suchen Sie den Profibus-Slave links in der Strukturansicht.
2. Klicken Sie auf die Busklemme, bei der Sie die LED schalten wollen.
3. Klicken Sie mit der rechten Maustaste auf einen Eingang bzw. Ausgang.



4. Klicken Sie im Kontextmenü auf **Online Write**.



5. Setzen Sie den Wert **Bool** auf „1“.



⇒ Unter TwinCAT wird der Wert „1“ unter Online bei dem entsprechenden Eingang bzw. Ausgang angezeigt. Wenn die LED auch auf der Busklemme aufleuchtet, die am Buskoppler angeschlossen ist, wurde ein Signal erfolgreich zum Profibus Slave übertragen. Der Profibus Master und Profibus Slave sind damit erfolgreich miteinander vernetzt.



### 5.3.10 Prozessdaten drehen

Die Prozessdaten werden standardmäßig im Intel Format übertragen. Wenn Sie die Daten im Motorola Format benötigen, müssen die Daten entsprechend gedreht werden. In diesem Arbeitsschritt wird gezeigt, wie die Daten in TwinCAT gedreht werden.

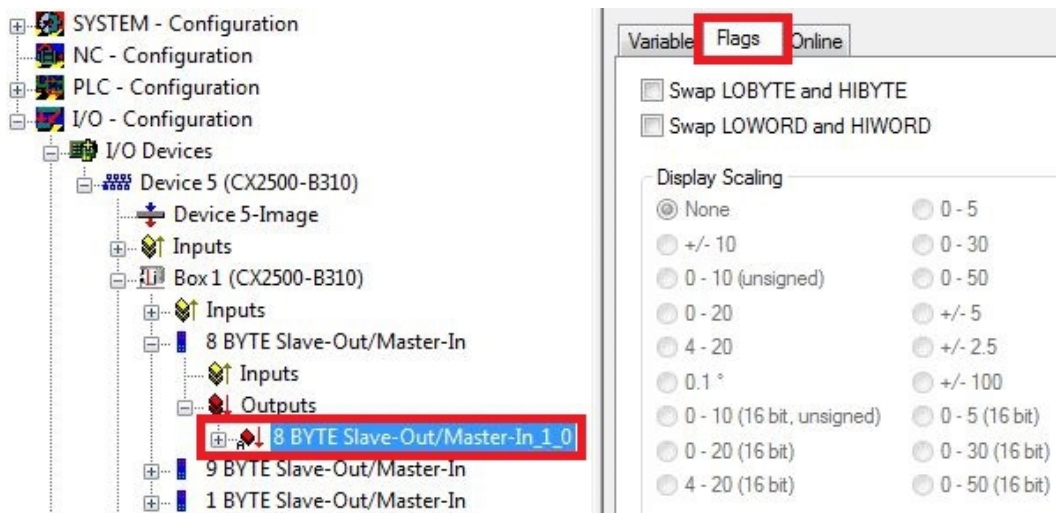
Wenn Sie das Standardformat benötigen, können Sie diesen Arbeitsschritt überspringen.

Voraussetzungen für diesen Arbeitsschritt:

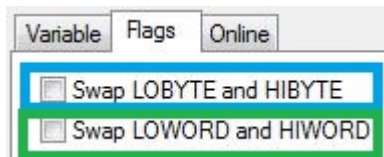
- Ein fertig parametrierter Slave.
- Ein Slave angeschlossen an einem Master. Der Master wird in TwinCAT eingefügt und danach nach dem Slave über den Master gescannt.

**Drehen Sie die Prozessdaten wie folgt:**

1. Klicken Sie rechts in der Strukturansicht auf eine Variable, deren Daten gedreht werden müssen.
2. Klicken Sie auf die Registerkarte **Flags**.



3. Klicken Sie auf die Option, die Sie benötigen. Bei WORD Variablen können Sie nur LOBYTE- und HIBYTE tauschen. Bei DWORDs zusätzlich das WORD.



⇒ Auf diese Weise können Sie Prozessdaten drehen. Benutzen Sie das folgende Beispiel, um zu sehen, wie sich die Daten bei den einzelnen Optionen verändern.  
Beispiel für DWORD.

Daten des Slaves	Daten die der Master empfängt			
Ursprüngliche Daten	Keine Option angewählt	Swap Byte (blau)	Swap Word (grün)	Swap beides (blau und grün)
0x01020304	0x01020304	0x02010403	0x03040102	0x04030201

Die Daten können auch im PLC-Projekt mit dem Befehl ROR gedreht werden.  
Beispiel für ST: VarProfibus:=ROR(VarAnalog,8); (\*Beide Variablen vom Typ WORD\*)

## 6 Fehlerbehandlung und Diagnose

Im Abschnitt Fehlerreaktionen wird beschrieben, wie Sie auf nicht oder falsch antwortende Slaves, PLC-Stops oder beim Start-Up reagieren können.

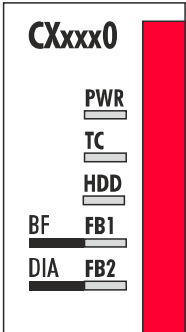
Das Kapitel Master beschreibt die verschiedenen Diagnosevariablen und wie die Diagnosedaten ausgelesen werden können. Im Kapitel Slave werden auch Diagnoseinformationen, die den Zustand des Profibus-Slaves anzeigen, beschrieben. Es wird auch gezeigt wie die Diagnosedaten ausgelesen werden können. Zusätzlich dazu werden die Fehlercodes des Profibus-Slave aufgelistet.

### 6.1 Diagnose-LEDs

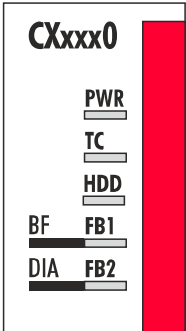
An dieser Stelle werden die Diagnose-LEDs eines PROFIBUS-Masters und PROFIBUS-Slaves beschrieben. Die Beschriftung der Diagnose-LEDs auf einem Feldbusmodul CX2500 und einem Embedded-PC mit Optionsschnittstelle ist identisch.

In der Beschreibung der LEDs wird deshalb nur zwischen PROFIBUS-Master und PROFIBUS-Slave unterschieden.

#### M310 (Master)

Anzeige	LED	Farbe	Bedeutung
	BF	Grün, an	PROFIBUS fehlerfrei
	Zeigt den Status des PROFIBUS an	Grün, blinkt	PROFIBUS, mindestens ein Slave im Fehler
	DIA	Rot, leuchtet	Kein PROFIBUS konfiguriert
	Zeigt Fehler des PROFIBUS an		

#### B310 (Slave)

Anzeige	LED	Farbe	Bedeutung
	BF	Grün, an	PROFIBUS im Datenaustausch
	Zeigt den Status des PROFIBUS an	Grün, blinkt	PROFIBUS, wartet auf Cfg Daten
	DIA	Rot, leuchtet	Kein PROFIBUS konfiguriert
	Zeigt Fehler des PROFIBUS an		

## 6.2 Fehlerreaktionen

### Ausfall eines Slaves

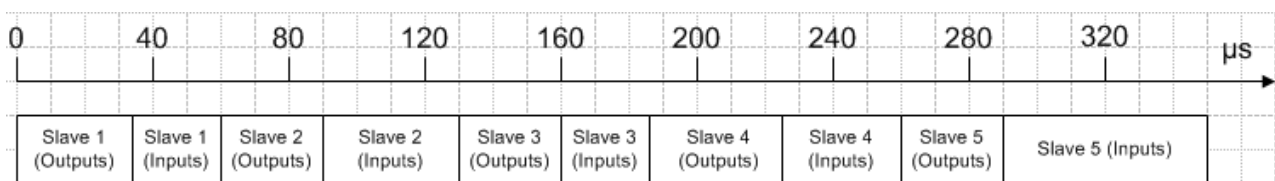
Wenn ein Slave nicht oder gestört antwortet, wiederholt der Master das Telegramm entsprechend des **Max Retry-Limit** mehrere Male (siehe: Registerkarte Bus-Parameter (DP) [▶ 26]). Wird ein gestörtes Telegramm empfangen, wiederholt der Master das Telegramm sofort. Wie lange der Master auf die Antwort des Slaves wartet, wird mit der **Slot-Time** auf derselben Registerkarte eingestellt.

Bei einem Data\_Exchange-Telegramm mit 12 MBaud, einer Slot-Time von 1000 tbit und einem Max Retry-Limit von 4 (Standardwerte) verzögert sich das Senden des folgenden Telegramms um den Wert:

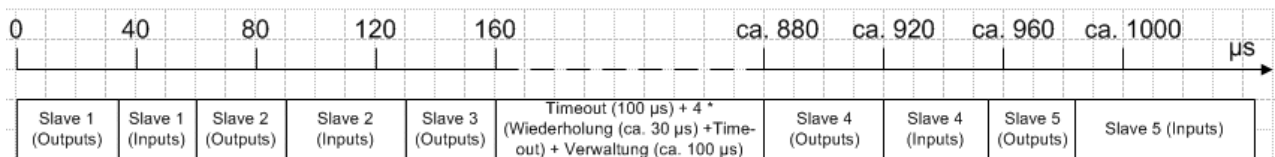
$$TDelay = (4 \times ((15 + \text{Anzahl Outputs}) \times 11 + 1000) - (15 + \text{Anzahl Inputs}) \times 11) / 12 \mu\text{s}$$

Der DpState des Slaves wird auf einen Wert ungleich 0 gesetzt. Die Auswirkung auf die DP-Verbindung kann eingestellt werden (siehe unten: Reaktion im Master).

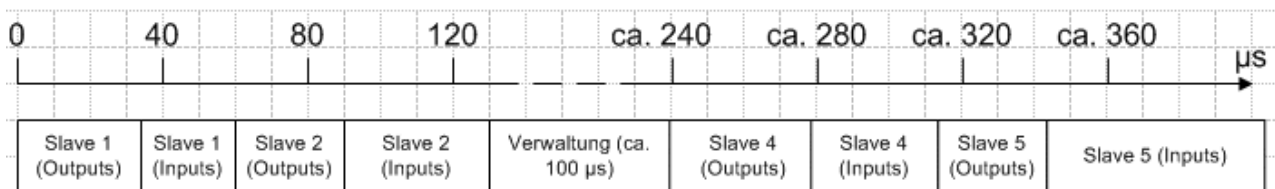
### Normaler DP-Zyklus (12 MBaud, 5 Slaves, im Schnitt 20 Bytes In, 20 Bytes Out je Slaves)



### Erstmalig gestörter DP-Zyklus (Slave 3 antwortet nicht)



### Folgende DP-Zyklen (Slave 3 nicht mehr in der Poll-Liste)



Weiterhin kann es passieren, dass der Slave falsch antwortet, weil z.B. aufgrund eines lokalen Ereignisses auf dem Slave die DP-Verbindung abgebaut wurde. In diesem Fall wird das Telegramm nicht wiederholt, sondern das nachfolgende Telegramm gesendet. Der DpState wird auf einen Wert ungleich 0 gesetzt. Der Slave wird aus der Poll-Liste ausgetragen und im nächsten DP-Zyklus nicht mehr angesprochen, der Sendezeitpunkt des nachfolgenden Telegramms verändert sich, bis die DP-Verbindung wieder erneut aufgebaut wurde.

## Reaktionen im Master

Die Reaktion im Master kann für jeden Slave eingestellt werden (siehe: Registerkarte [Features](#) [► 32]). Die Reaktion im Master kann mit der Option **NoAnswer-Reaction** eingestellt werden.

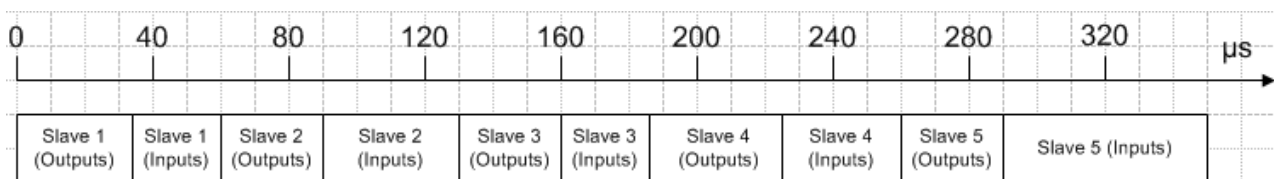
Hier wird festgelegt, ob die DP-Verbindung zum Slave sofort oder erst nach Ablauf der DP-Watchdog-Zeit ohne korrektes Empfangstelegramm (Data\_Exchange-Telegramm) abgebaut werden soll.

Wenn die DP-Verbindung sofort abgebaut werden soll (Leave Data-Exch, Registerkarte [Features](#) des Slaves), wird der Slave aus der Poll-Liste ausgetragen und im nächsten DP-Zyklus nicht mehr angesprochen, bis die DP-Verbindung wieder erneut aufgebaut wurde. Um die DP-Verbindung zum Slave wieder aufzubauen, werden mindestens 7 Telegramme gesendet. Der Vorgang dauert in der Regel mindestens 10-20 ms.

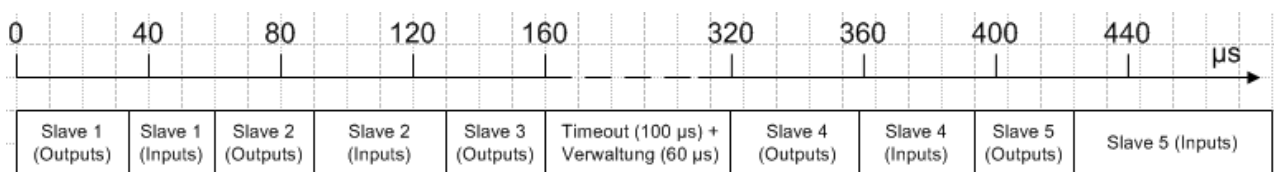
Wenn die DP-Verbindung erst nach Ablauf der Watchdog-Zeit abgebaut werden soll (Stay in Data-Exch (for WD-Time), Registerkarte [Features](#) des Slaves), wird die eingestellte Watchdog-Zeit des Slaves abgewartet und auf eine korrekte Antwort gewartet. Im nächsten Pollzyklus wird der Slave erneut angesprochen. Falls der Slave wieder nicht antwortet, wird keine Wiederholung mehr gesendet.

Es macht dann Sinn die Watchdog-Zeit abzuwarten, wenn der PROFIBUS-Zyklus auch beim Ausfall eines Slaves zeitlich möglichst konstant ablaufen soll und der Ausfall eines Slaves für einen oder mehrere Zyklen toleriert werden kann (z.B. in der Betriebsart DP/MC (Equidistant)). In diesem Fall sollte die DP-Watchdog-Zeit des Slaves entsprechend der noch tolerierbaren Ausfall-Zeit des Slaves eingestellt und das Max Retry-Limit (DX) (Registerkarte [Bus-Parameter](#) (DP) des Masters) auf 0 gesetzt werden.

### Normaler DP-Zyklus (12 Mbaud, 5 Slaves, im Schnitt 20 Bytes In, 20 Bytes Out je Slaves) im Mode "Stay in Data-Exch (for WD-Time)"



### Erster gestörter und folgende DP-Zyklen im Mode "Stay in Data-Exch (for WD-Time)" (Slave 3 antwortet nicht)



- Auswirkung auf die Input-Daten des Slaves (Changes of the Input Data), falls der Slave nicht richtig antwortet**  
 Hier wird festgelegt, ob die Input-Daten des Slaves auf 0 gesetzt (Inputs will be set to 0, Registerkarte [Features](#) [► 32] des Slaves) oder der alte Wert erhalten bleiben soll (No changes), wenn der Slave ausfällt. In beiden Fällen wird auf jeden Fall der DpState des Slaves auf einen Wert ungleich 0 gesetzt, so dass die Task immer erkennen kann, ob die Daten gültig sind oder nicht. Wenn ein Slave gestört antwortet, werden die Inputdaten, unabhängig von der Einstellung **Changes of the Input Data** immer auf 0 gesetzt.
- Einstellung des Wiederanlauf-Verhaltens des Slaves (Restart Behaviour of the Slave), falls die DP-Verbindung zu dem Slave abgebaut wurde**  
 Hier wird festgelegt, ob die DP-Verbindung zu einem Slave, dessen DP-Verbindung abgebaut wurde, automatisch wieder aufgebaut wird, oder ob dieses manuell durch einen ADS-WriteControl-Aufruf passieren soll.  
 Diese Einstellung ist verfügbar, wenn der Slave nicht einen CX-Master angeschlossen ist.
- Auswirkung auf den Zustand des Masters (Reaction of the Master), falls die DP-Verbindung zu dem Slave abgebaut wurde**  
 Hier wird festgelegt, ob der Abbau der DP-Verbindung zu einem Slave ohne weitere Auswirkungen

bleiben (No Reaction, Defaulteinstellung) oder ob der Master den STOP-Zustand einnehmen und damit die DP-Verbindungen zu allen Slaves abbauen soll.  
Diese Einstellung ist verfügbar, wenn der Slave nicht einen CX-Master angeschlossen ist.

### Ausfall des Masters

- **Überwachung in der SPS/IO-Task**

Bei dauerhaften Busstörungen kann sich der DP-Zyklus auch bei 12 Mbaud auf bis zu 100 ms verlängern. Um den DP-Master zu überwachen, gibt es eine Status-Variable CycleCounter, die in der SPS verknüpft werden kann. Diese Variable wird vom Master nach jedem DP-Zyklus inkrementiert. Die Variable kann in der SPS überwacht werden, um einen Ausfall des Masters festzustellen.

- **Überwachung im Slave**

Um den Ausfall des Masters bzw. die Übertragung auf dem PROFIBUS zu überwachen, kann beim Slave eine Watchdog-Zeit aktiviert werden (siehe: Registerkarte [Profibus](#) [► 31]). Die Watchdog-Zeit muss mindestens zweimal so groß wie das Maximum aus Estimated DP-Cycle und Cycle Time eingestellt werden (siehe: Registerkarte [CCAT PBM](#) [► 25]).

### Ausfall der SPS/IO-Task

Es wird zwischen den Fällen SPS-Stop, Erreichen eines Breakpoints und Task-Stop unterschieden (IO-Task, NC-Task wird nur beim System-Stop gestoppt). Beim SPS-Stop werden die Ausgangsdaten von der SPS noch auf 0 gesetzt, beim Erreichen eines Breakpoints bleiben die Daten zunächst unverändert.

Im Master wird mit einer Task-Watchdog Zeit die Task überwacht (siehe: Registerkarte [Start-Up/Fault-Settings](#) [► 27]). Wenn innerhalb dieser Task-Watchdog Zeit keine neue Datenübergabe erfolgt, geht der Master entsprechend der Einstellung Reaction on Task-Stop in den Zustand Clear (Ausgänge werden auf 0 bzw. in den sicheren Zustand gesetzt Fail\_Safe = 1 in der GSD-Datei). Bei der Einstellung "Operate" behalten die Ausgänge den letzten Wert.

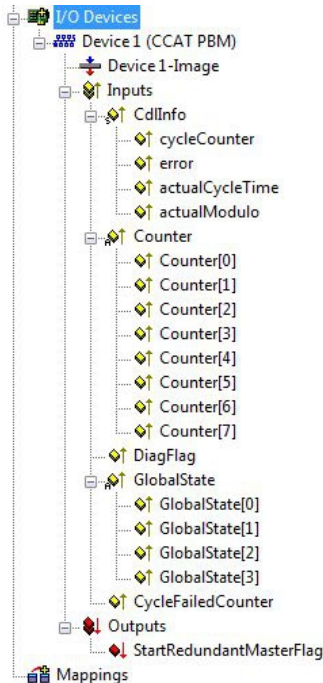
### Start-Up Verhalten

Beim Start des TwinCAT Systems werden die DP-Verbindungen zu allen Slaves aufgebaut. Solange die höchstprioritäre zugehörige Task noch nicht gestartet wurde, sendet der Master auch nach dem Aufbau der DP-Verbindung noch keine Data\_Exchange-Telegramme sondern Diagnose-Telegramme. Sobald die höchstprioritäre zugehörige Task einmal Daten übergeben hat und die DP-Verbindung des entsprechenden DP-Slaves aufgebaut ist, sendet der Master zyklisch (mit der höchstprioritären zugeordneten Task) je ein Data\_Exchange-Telegramm zu den entsprechenden Slaves.

## 6.3 Master

### Diagnose Variablen

Der Master verfügt über verschiedene Diagnosevariablen, die den Zustand des Masters sowie des Profibusses beschreiben und die in der SPS verknüpft werden können:



#### CdInfo:

Variable	Beschreibung
cycleCounter	Wird am Ende jedes PROFIBUS-Zyklus inkrementiert, so dass man mit dieser Variable feststellen kann, ob der letzte Zyklus beendet war, bevor die Task gestartet wurde.
error	Zeigt die Anzahl der Slaves an, mit denen kein Datenaustausch im letzten Zyklus durchgeführt werden konnte. Nur wenn dieser Wert ungleich 0 ist, muss die Variable BoxState der Slaves überprüft werden.
actualCycleTime	Zeigt die aktuelle Zykluszeit in 4/25 µs an. Diese Variable wird nur aktualisiert, wenn sich alle Slaves im Datenaustausch befinden (also CdInfo.error gleich 0 ist).
actualModulo	Zeigt das aktuelle Modulo an. Diese Variable spielt nur eine Rolle, wenn die Slaves priorisiert sind.

**Counter:** Diese Variablen werden nur für den Redundanz-Modus benutzt.

Variable	Beschreibung
Counter[0-1]	Diese Variablen sind reserviert für Erweiterungen.
Counter[2]	Diese Variable zählt erhaltene Telegramme. Der Zähler wird jedes Mal um 1 inkrementiert, wenn ein fehlerfreies PROFIBUS-Telegramm erhalten wird.
Counter[3]	Diese Variable zählt die erhaltenen Telegramme vom Primary-Master. Der Zähler wird jedes Mal um 1 inkrementiert, wenn ein fehlerfreies Telegramm vom Primary-Master erhalten wird (Der Primary Master hat die gleiche Adresse wie der Redundanz-Master).
Counter[4]	Diese Variable ist ein Timeout-Zähler. Der Zähler wird jedes Mal um 1 inkrementiert, wenn ein Timeout auf dem PROFIBUS auftritt.
Counter[5-7]	Diese Variablen sind reserviert für Erweiterungen.

**DiagFlag:**

Variable	Beschreibung
DiagFlag	<p>Zeigt an, ob sich die Master-Diagnoseinformationen geändert haben. Diese werden dann per ADS vom Steuerungsprogramm ausgelesen, woraufhin die Variable "DiagFlag" wieder zurückgesetzt wird.</p> <p>0 = Diagnosedaten unverändert.</p> <p>1 = Diagnosedaten haben sich geändert. Benutzen Sie ADS-Read, um die Daten auszulesen.</p>

**GlobalState:**

Variable	Beschreibung
GlobalState[0]	<p>Gibt den Zustand des Masters an.</p> <p>0 = RUN 1 = RESET 2 = OFFLINE 3 = STOP</p>
GlobalState[1]	Gibt die Anzahl der festgestellten Bus-Fehler an.
GlobalState[2]	Ist reserviert für Erweiterungen.
GlobalState[3]	Ist reserviert für Erweiterungen.

**CycleFailedCounter:**

Variable	Beschreibung
CycleFailedCounter	Dieser Zähler zeigt an, wie oft beim Start der TwinCAT-Task der PROFIBUS-Zyklus noch nicht fertig war.

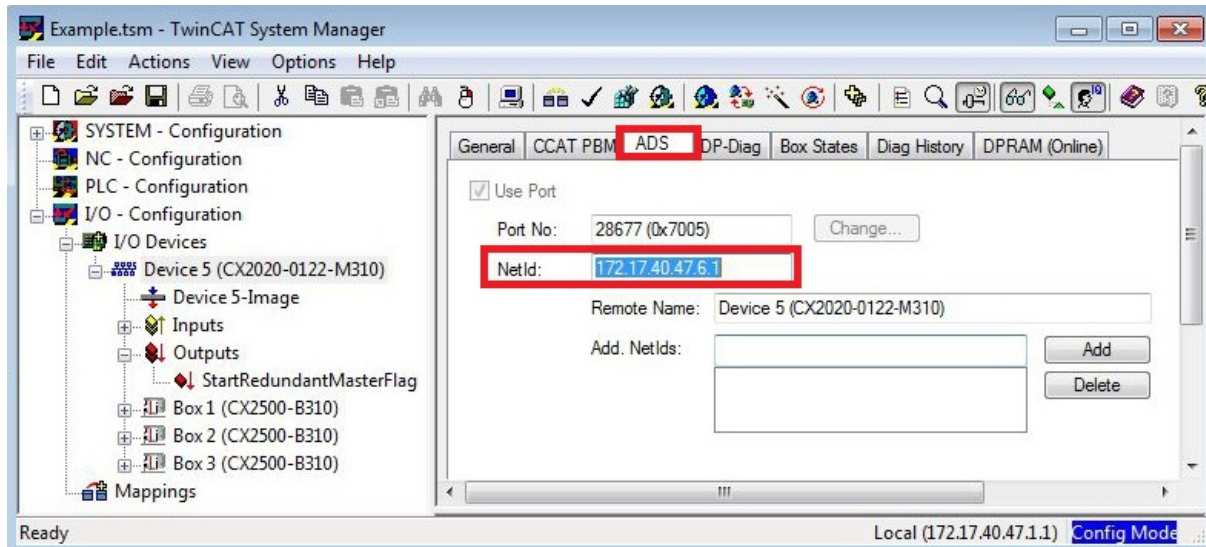
**StartRedundantMasterFlag:** wird für den Redundanz-Modus benutzt.

Variable	Beschreibung
StartRedundantMasterFlag	Wird der Wert auf 1 (True) gesetzt, wird der Redundanz-Master auf dem PROFIBUS aktiv. Bei dem Wert 0 (False) ist der Redundanz-Master offline.

## 6.3.1 ADS-Interface

### ADS NetId der PROFIBUS-Schnittstelle

Für die ADS-Kommunikation wird eine NetId benötigt. Diese ist im TwinCAT System Manager dem Karteireiter ADS des PROFIBUS-Interfaces zu entnehmen.



### ADS-Interface

Sämtliche azyklischen Daten werden mit ADS-Read, ADS-Write oder ADS-Write-Control an die bzw. von den Embedded-PCs übertragen. Das Profibus-Interface der Embedded-PCs hat eine eigene Net-ID und unterstützt die folgenden Ports:

Port	Beschreibung
200	Damit wird das PROFIBUS-Interface selbst adressiert, d.h. Daten, die lokal auf dem PROFIBUS-Interface liegen, bei denen in der Regel kein zusätzlicher Buszugriff nötig ist.
0x1000 - 0x107E	Damit wird ein angeschlossener PROFIBUS-Teilnehmer adressiert, wobei sich Adresse aus Port-0x1000 berechnet (oder 0x1000 + Slave Adresse), es wird auch immer ein Buszugriff durchgeführt.



**ADS-Read**

Es folgt eine Übersicht der von den Embedded-PCs unterstützten IndexGroup/IndexOffset bei ADS-Read.

*Tab. 1: IndexGroup bei lokaler Adressierung des Embedded-PC (Port 200)*

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Beschreibung
0xF181	0x0000-0x007E	BYTE-Offset innerhalb der Daten	Damit werden die Diagnosedaten eines projizierten DP-Slaves ausgelesen, die Stationsadresse berechnet sich aus dem IndexGroup(Hi-Word). Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten die im Kapitel <a href="#">Slave-Diagnose [ 78 ]</a> beschriebenen Diagnosedaten eines projizierten DP-Slaves.
0xF830	0x0000-0x007E	immer 0	Damit kann festgestellt werden, welche DP-Slaves am PROFIBUS vorhanden sind, unabhängig davon, ob sie projiziert wurden oder nicht, die Stationsadresse wird im IndexGroup (Hi-Word) eingetragen. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, hat der entsprechende DP-Slave korrekt geantwortet, die Daten enthalten die Ident-Nummer des Slaves (BYTE-Offset 0-1) und die ausgelesenen CfgData (ab BYTE-Offset 2) (siehe: <a href="#">Upload der Konfiguration [ 75 ]</a> ).

*Tab. 2: IndexGroup bei Adressierung eines projizierten PROFIBUS-Teilnehmers (Port 0x1000-0x107E)*

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Beschreibung
0x00-0xFF	0x00	0x00-0xFF	Damit wird ein DPV1-Read an den entsprechenden, projizierten DPV1-Slave über eine Class-1-Verbindung gesendet, die DPV1-Slot-Nummer entspricht der IndexGroup, der DPV1-Index entspricht dem IndexOffset. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten die ausgelesenen DPV1-Daten (siehe: <a href="#">DPV1-Kommunikation [ 74 ]</a> ).

**ADS-Write**

Es folgt eine Übersicht der von den Embedded-PCs unterstützten IndexGroup/IndexOffset bei ADS-Write.

*Tab. 3: IndexGroup bei Adressierung eines projizierten PROFIBUS-Teilnehmers (Port 0x1000-0x107E)*

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Beschreibung
0x00-0xFF	0x00	0x00-0xFF	Damit wird ein DPV1-Write an den entsprechenden, projizierten DPV1-Slave über eine Class-1-Verbindung gesendet, die DPV1-Slot-Nummer entspricht der IndexGroup, der DPV1-Index entspricht dem IndexOffset (siehe: <a href="#">DPV1-Kommunikation [ 74 ]</a> ).

**ADS-WriteControl**

Es folgt eine Übersicht der von den Embedded-PCs unterstützten Dienste beim ADS-WriteControl.

*Tab. 4: ADS-WriteControl bei Adressierung eines projizierten PROFIBUS-Teilnehmer (Port 0x1000-0x107E)*

AdsState	DeviceState	State des Slave	Beschreibung
STOP (6)	0x00	RUN (5)	Damit wird der Slave gestoppt, d.h. die Prozessdatenverbindung zu dem entsprechenden DP-Slave (Data_Exchange) wird abgebaut (mit SetPrm,Unlock).
RUN (5)	0x00	STOP (6)	Damit wird der Slave nach einem Stoppen erneut gestartet, d.h. die Prozessdatenverbindung zu dem entsprechenden DP-Slave (Data_Exchange) wird wieder aufgebaut (normaler DP-Hochlauf).

## 6.3.2 DPV1-Kommunikation

Der Master unterstützt auf einer C1-Verbindung die Dienste Read und Write sowie auf einer C2-Verbindung die Dienste Read, Write, Data\_Transport, Initiate und Abort.

### C1-Verbindung (MSAC-C1)

Die C1-Verbindung ist für den Master reserviert, die mit dem Slave zyklischen Datenaustausch durchführt (C1-Master). Um die C1-Verbindung bei einem Slave nutzen zu können, muss der Slave DPV1 unterstützen (in der GSD-Datei müssen dafür die Zeile "DPV1\_Slave = 1" und das Schlüsselwort "C1\_Max\_Data\_Len" mit einer entsprechenden Länge eingetragen sein). Weiterhin ist in der Regel die C1-Funktionalität durch Setzen des Bit 7 von **PrmData**-Byte 0 bei dem entsprechenden Slave zu aktivieren (bei Beckhoff-Geräten, die DPV1 unterstützen, passiert das automatisch).

MSAC-C1-Read ist auf ADS-Read, MSAC-C1-Write auf ADS-Write abgebildet:

#### MSAC-C1 Read

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	Slot-Number (DPV1-Parameter)
IndexOffset	Index (DPV1-Parameter)
Length	Länge der auszulesenden Daten
Data	bei Response: ausgelesene Daten

#### MSAC-C1 Write

ADS-Write-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	Slot-Number (DPV1-Parameter)
IndexOffset	Index (DPV1-Parameter)
Length	Länge der zu schreibenden Daten
Data	bei Request: zu schreibende Daten

### 6.3.3 Upload der Konfiguration

Während des Betriebs kann der PROFIBUS per ADS-Read nach neuen Geräten gescannt werden:

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (siehe Karteireiter ADS des Devices)
Port	200
IndexGroup	0x00yyF830 (yy = Stationsadresse, Beckhoff-Geräte liefern die gleichen Informationen wie Fremd-Geräte)
IndexOffset	0
Length	1538
Data	Konfigurationsdaten des Slaves

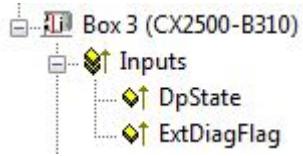
Falls es sich um Fremdgeräte handelt oder in der IndexGroup angegeben ist, dass sich Beckhoff-Geräte genauso verhalten sollen wie Fremd-Geräte, werden die folgenden Informationen in der ADS-Read-Response zurückgeliefert:

Offset	Beschreibung
0 - 1	1
2 - 7	DP-Diagnosedaten Byte 0-5 (siehe: <a href="#">Slave-Diagnose [► 78]</a> )
8 - 251	DP-Konfigurationsdaten ( <b>CfgData</b> )

## 6.4 Slave

### Diagnose Variablen

Der Slave verfügt über verschiedene Diagnosevariablen, die den Zustand des Slave sowie des Profibusses beschreiben und die in der SPS verknüpft werden können:



**DpState:**

Wert	Beschreibung
0	No Error: Station ist im Datenaustausch.
1	Station deactivated: Der Slave wurde deaktiviert.
2	Station not exists: Slave antwortet nicht. Prüfen Sie, ob der Slave eingeschaltet, die Verkabelung in Ordnung ist oder die Stationsadresse korrekt ist.
3	Master lock: Der Slave ist mit einem anderen Master im Datenaustausch.
4	Invalid slave response: Eine falsche Antwort des Slaves. Tritt temporär auf wenn der Slave aufgrund eines lokalen Ereignisses den Datenaustausch beendet hat.
5	Parameter fault: Zeigt einen Parametrierungsfehler an.
6	Not supported: Eine DP-Funktion wird nicht unterstützt. Prüfen Sie, ob die GSD-Datei korrekt oder ob die Stationsadresse korrekt ist.
7	Config fault: Zeigt einen Konfigurationsfehler an. Prüfen Sie, ob die angefügten Klemmen bzw. Module in Ordnung sind.
8	Station not ready: Das Gerät befindet sich im Hochlauf. Wird während des Hochlaufs temporär angezeigt.
9	Static diagnosis: Der Slave meldet eine statische Diagnose und kann zurzeit keine gültigen Daten liefern. Prüfen Sie den Betriebszustand am Slave.
10	Diagnosis overflow: Der Slave meldet einen Diagnose-Überlauf. Lesen Sie die Diagnosedaten mit ADS-Read aus und prüfen Sie Betriebszustand am Slave.
11	Physical fault: Zeigt einen physikalischer Fehler bei der Antwort eines Slaves. Prüfen Sie die Verkabelung.
12	Data-Exchange left: Die Datenübertragung wurde nicht beendet.
13	Severe bus fault: Zeigt einen schweren Busfehler an. Überprüfen Sie die Verkabelung.
14	Telegram fault: Der Slave antwortet mit einem ungültigen Telegramm.
15	Station has no ressources: Der Slave hat nicht genügend Ressourcen für das Telegramm. Prüfen Sie, ob die GSD-Datei korrekt ist.
16	Service not activated: Tritt auf, falls der Slave aufgrund eines lokalen Ereignisses den Datenaustausch beendet. Prüfen Sie, ob DP-Funktionalität beim Slave deaktiviert ist.
17	Unexpected telegram: Unerwartetes Telegramm, kann temporär auftreten, falls zwei PROFIBUS-Netze zusammengesteckt werden oder prüfen, ob Buszeiten bei zweitem Master gleich eingestellt sind.
18	Station ready: Tritt temporär beim Hochlauf auf und solange die Task noch nicht gestartet ist.
31	only for EtherCAT gateways: WC-State des zyklischen EtherCAT-Frames ist 1.
128	Slave waiting for data transfer: Der Slave wurde parametrierung und konfiguriert, hat aber noch kein Data_Exchange-Telegramm empfangen.
129	Slave waiting for configuration: Der Slave wurde parametrierung, hat aber noch kein Chk_Cfg-Telegramm empfangen.

Wert	Beschreibung
130	Slave waiting for parameter: Slave wurde noch nicht parametriert und wartet auf Set_Prm (Lock)-Telegramm.
131	Slave waiting for baudrate: Die Baudrate wurde nicht eingestellt.
132	Slave waiting for station no. from PLC: Die Profibus-Adresse wurde nicht eingestellt.

**ExtDiagFlag:**

Wert	Beschreibung
0	Diagnosedaten haben sich nicht verändert.
1	Diagnosedaten haben sich verändert. Benutzen Sie ADS-Read um die Daten auszulesen.

## 6.4.1 Slave-Diagnose

**DP-State**

Je DP-Slave gibt es eine Status-Variable, die den aktuellen Zustand des DP-Slaves anzeigt. Dieser Status ist ein Echtzeit-Status, d.h. er passt immer zu den aktuellen Daten des DP-Slaves und kann mit einer PLC-Variable verknüpft werden (-> **DpState** des Slaves):

**Diagnosedaten**

Jeder DP-Slave kann während des Data\_Exchange-Betriebs azyklisch DP-Diagnosedaten melden. Dabei setzt der Slave in der Antwort des zyklischen Data\_Exchange-Telegramms das Diag\_Flag, woraufhin der DP-Master automatisch die DP-Diagnosedaten beim Slave ausliest. Der Data-Exchange-Zyklus wird beim Beckhoff-DP-Master dabei nicht beeinflusst, da das DP-Diagnosetelegramm am Ende des zyklischen Data-Exchange-Zyklus (vor dem Anfang des nächsten Zyklus) gesendet wird. Wenn sich die beim Slave ausgelesenen DP-Diagnosedaten gegenüber dem letzten Zustand geändert haben, setzt der DP-Master die Variable ExtDiagFlag, die mit einer Variable des Steuerungsprogramms verknüpft werden kann.

Die aktuellen Diagnosedaten des DP-Slaves werden im System-Manager Karteireiter **Diag** des Slaves angezeigt. Außerdem können sie per ADS vom Steuerungsprogramm ausgelesen werden, woraufhin die Variable "ExtDiagFlag" wieder zurückgesetzt wird:

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS des Devices)
Port	200
IndexGroup	0x00yyF181 (yy = Stationsadresse des Slaves)
IndexOffset	Offset innerhalb der Diagnosedaten
Length	Länge der auszulesenden Diagnosedaten
Data	Diagnosedaten

Die Diagnosedaten behalten die Slave-Statistiken (32 Bytes) und die vom Slave gesendeten DP-Diagnosedaten (bis zu 244 Bytes) und sind wie folgt aufgebaut:

Offset	Bedeutung
<b>Slave-Statistiken</b>	
0	Receive-Error-Counter (WORD): Anzahl der fehlerhaften Telegramme bei der Kommunikation mit diesem Slave
2	Repeat-Counter[8] (WORD): Die Repeat-Counter zeigen an, wie oft wie viele Repeats gemacht werden mussten. Repeat-Counter[0] zeigt an, wie oft ein Telegramm zu diesem Slave einmal wiederholt werden mußte, Repeat-Counter[1], wie oft ein Telegramm zu diesem Slave zweimal wiederholt werden musste, etc. Die maximale Anzahl der Wiederholungen wird mit dem Parameter <b>Max Retry-Limit</b> (s. Dialog <b>Bus-Parameter</b> ) eingestellt, der Wertebereich geht von 0 bis 8, daher gibt es hier 8 Repeat-Counter (für 1 bis 8 Wiederholungen)
18	reserviert für Erweiterungen
20	NoAnswer-Counter (DWORD): Anzahl der Telegramme bei der Kommunikation mit diesem Slave, auf die nicht geantwortet wurde. Wenn ein Slave das erste Mal nicht antwortet, wird entsprechend des <b>MaxRetryLimit</b> das Telegramm wiederholt, wenn er auch dann nicht geantwortet hat, wird beim nächsten Mal keine Wiederholung mehr durchgeführt.
24-27	Last-DPV1-Error[4] (BYTE): Hier wird die letzte fehlerhafte DPV1-Antwort eingetragen (Byte 0: DPV1-Dienst (Bit 7 ist gesetzt und zeigt damit einen Fehler an), Byte 1: Error Decode, Byte 2: Error_Code_1 (Error_Class/Error_Code), Byte 3: Error_Code_2), s. Beschreibung <u>DPV1-Fehlercodes</u> [ <a href="#">▶ 85</a> ]
27-31	reserviert für Erweiterungen
ab 32	<b>DP-Diagnosedaten</b>

Es folgt eine Beschreibung der DP-Diagnosedaten

Offset	Bedeutung
0x00.0	StationNonExistent: Slave beim letzten Telegramm nicht geantwortet
0x00.1	StationNotReady: Slave verarbeitet noch das Set_Prm bzw. Chk_Cfg-Telegramm
0x00.2	CfgFault: Slave meldet einen Konfigurationsfehler
0x00.3	ExtDiag: Extended DiagData sind vorhanden und gültig
0x00.4	NotSupported: Slave unterstützt ein Feature nicht, das mit Set_Prm oder Global_Control gefordert wurde
0x00.5	InvalidSlaveResponse: Slave antwortet nicht DP-konform
0x00.6	PrmFault: Slave meldet einen Parametrierfehler
0x00.7	MasterLock: Slave ist im Datenaustausch mit einem anderen Master
0x01.0	PrmReq: Slave muß neu parametrieren und konfiguriert werden
0x01.1	StatDiag: Slave meldet statische Diagnose bzw. Applikation des DPV1-Slave noch nicht bereit für den Datenaustausch
0x01.2	PROFIBUS-DP-Slave
0x01.3	WdOn: DP-Watchdog ist eingeschaltet
0x01.4	FreezeMode: DP-Slave ist im Freeze-Mode
0x01.5	SyncMode: DP-Slave ist im Sync-Mode
0x01.6	reserviert
0x01.7	Deactivated: DP-Slave wurde deaktiviert
0x02.0	reserviert
0x02.1	reserviert
0x02.2	reserviert
0x02.3	reserviert
0x02.4	reserviert
0x02.5	reserviert
0x02.6	reserviert
0x02.7	ExtDiagOverflow: zu viele Extended DiagData vorhanden
0x03	MasterAdd: Stationsadresse des Masters, der mit dem Slave Datenaustausch macht
0x04,0x05	IdentNumber
ab 0x06	Extended DiagData

### Extended DiagData

Bei den Extended DiagData wird zwischen Kennungsdiagnose, Kanaldiagnose und Herstellerspezifischer Diagnose unterschieden, wobei das erste Byte jeweils den Typ der Diagnose und die Länge der dazugehörigen Daten anzeigt. In den Extended DiagData können auch mehrere Diagnosetypen nacheinander folgen.

### Header-Byte

Bit	Bedeutung
0-5	Länge der zugehörigen Diagnosedaten inklusive Header-Byte
6-7	0 = Herstellerspezifische Diagnose (DPV1 wird nicht unterstützt) bzw. DPV1-Diagnose (DPV1 wird unterstützt (DPV1_Enable = 1) in zugehöriger GSD-Datei)
	Moduldiagnose
	Kanaldiagnose
	Revision-Number

### Herstellerspezifische Diagnose

Der Aufbau der Herstellerspezifischen Diagnose ist der Dokumentation des DP-Slaves zu entnehmen.



## 6.4.2 Individuelle Diagnosedaten

Die Steuerung ermöglicht Ihnen aus der PLC heraus Diagnose-Daten zu verschicken. So können Sie sich Ihre eigene Diagnose Mitteilung für den Master schreiben und diese Individuell mit Daten füllen (s. u. Gerätespezifische Diagnosedaten).

### DP-Diagnosedaten (DiagData)

#### **i** Übertragung des Diagnose-Telegramms

Ein Diagnose-Telegramm wird erst dann zur Steuerung übertragen, wenn sich die Diagnose-Daten geändert haben.

Die DP-Diagnosedaten bestehen aus 6 Bytes DP-Standard-Diagnose und bis zu 238 Bytes gerätespezifischen Diagnosedaten.

Wenn sich die DP-Diagnosedaten ändern, meldet das der Slave dem Master, der die geänderten Diagnosedaten in der Regel daraufhin automatisch abholt. Die DP-Diagnosedaten sind also nicht in Echtzeit zu den DP-Prozessdaten, sondern immer erst einige Zyklen später in der Steuerung.

In TwinCAT werden die DP-Diagnosedaten per ADS von der DP-Master-Anschaltung ausgelesen

### DP-Standard-Diagnosedaten

Offset	Bedeutung
0x00.0	StationNonExistent: Slave beim letzten Telegramm nicht geantwortet
0x00.1	StationNotReady: Slave verarbeitet noch das Set_Prm bzw. Chk_Cfg-Telegramm
0x00.2	CfgFault: Slave meldet einen Konfigurationsfehler
0x00.3	ExtDiag: Extended DiagData sind vorhanden und gültig
0x00.4	NotSupported: Slave unterstützt ein Feature nicht, das mit Set_Prm oder Global_Control gefordert wurde
0x00.5	InvalidSlaveResponse: Slave antwortet nicht DP-konform
0x00.6	PrmFault: Slave meldet einen Parametrierfehler
0x00.7	MasterLock: Slave ist im Datenaustausch mit einem anderen Master
0x01.0	PrmReq: Slave muß neu parametrier und konfiguriert werden
0x01.1	StatDiag: Slave meldet statische Diagnose bzw. Applikation des DPV1-Slave noch nicht bereit für den Datenaustausch
0x01.2	PROFIBUS-DP-Slave
0x01.3	WdOn: DP-Watchdog ist eingeschaltet
0x01.4	FreezeMode: DP-Slave ist im Freeze-Mode
0x01.5	SyncMode: DP-Slave ist im Sync-Mode
0x01.6	reserviert
0x01.7	Deactivated: DP-Slave wurde deaktiviert
0x02.0	reserviert
0x02.1	reserviert
0x02.2	reserviert
0x02.3	reserviert
0x02.4	reserviert
0x02.5	reserviert
0x02.6	reserviert
0x02.7	ExtDiagOverflow: zu viele Extended DiagData vorhanden
0x03	MasterAdd: Stationsadresse des Masters, der mit dem Slave Datenaustausch macht
0x04,0x05	IdentNumber
ab 0x06	gerätespezifische Diagnosedaten (Extended DiagData)

**Gerätespezifische Diagnosedaten****i Übertragung von anwenderspezifischen Diagnose-Daten**

Im Byte[0] der Daten muss eine 0x08 stehen. Byte[1..5] werden vom CX überschrieben. Ab Byte 6 können Ihre eigenen Diagnose-Daten stehen. Achten Sie bei Ihren eigenen Diagnose Daten auf die Einhaltung der PROFIBUS-Norm für Anwenderspezifische Diagnose.

Für das absetzen der Diagnose Daten wird der ADSWRITE Baustein verwendet. Die aktuelle DP-Diagnose, wie sie auf den Bus gesendet wird kann per ADSREAD gelesen werden. Achten Sie beim Lesen darauf, dass sie 6 Byte (die PROFIBUS Standard DP-Diagnose) mitberücksichtigen müssen, d.h. 6 Byte mehr auslesen, als Sie geschrieben haben. Die ADS Parameter sind mit dem Lesen identisch.

Eingangsparameter	Beschreibung
NETID	lokale NetId des Profibus Device
PORT Nummer	0x1000+Slave Adresse
IDXGRP	16#F481
IDXOFFS	0
LEN	max. 244
SRCADDR	Pointer auf die Diagnose Daten

### 6.4.3 DP-V1 Kommunikation

#### C1-Verbindung (MSAC-C1)

Die C1-Verbindung ist für den Master reserviert, die mit dem Slave zyklischen Datenaustausch durchführt (C1-Master). Um die C1-Verbindung bei einem Slave nutzen zu können, muss der Slave DPV1 unterstützen (in der GSD-Datei müssen dafür die Zeile "DPV1\_Slave = 1" und das Schlüsselwort "C1\_Max\_Data\_Len" mit einer entsprechenden Länge eingetragen sein). Weiterhin ist in der Regel die C1-Funktionalität durch Setzen des Bit 7 von **PrmData**-Byte 0 (s. Karteireiter **PROFIBUS** des Slaves) bei dem entsprechenden Slave zu aktivieren (bei Beckhoff-Geräten, die DPV1 unterstützen, passiert das automatisch).

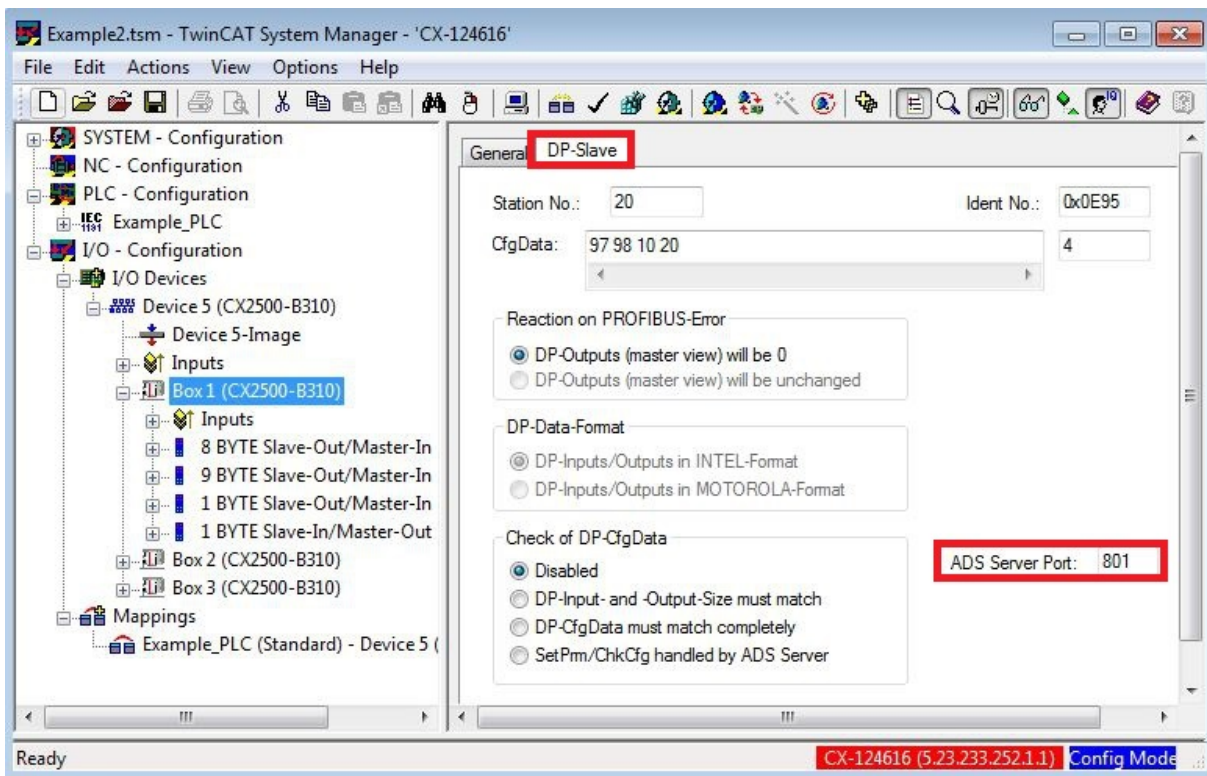
MSAC-C1-Read ist auf ADS-Readind, MSAC-C1-Write auf ADS-Writeinc abgebildet:

#### PROFIBUS DPV1

Der DP-Slave unterstützt eine DPV1-MSAC\_C1-Server-Verbindung, die zusammen mit der zyklischen Verbindung aufgebaut wird. Diese kann benutzt werden, um neben den zyklischen Daten noch größere Mengen azyklische Daten übertragen zu können. Ein vom Master empfangenes DPV1-Read-Telegramm wird als ADS-Read-Indication an die PLC gemeldet, ein vom Master empfangenes DPV1-Write-Telegramm wird als ADS-Write-Indication an die PLC gemeldet. Das PLC-Programm ist dann für die Read bzw. Write-Antwort verantwortlich, dazu sind die Funktionen ADS-Read-Response bzw. ADS-Write-Response aufzurufen.

#### Einstellung im System Manager

Um die DP-V1 Dienste in die SPS zu transportieren muss diese Funktion durch die Angabe des Ports aktiviert werden.



Port 0 deaktiviert, 802 SPS-Laufzeit 1, Task 1

**MSAC-C1 Read**

Eine DPV1-MSAC\_C1-Read-Indication wird wie folgt auf eine ADS-Read-Indication abgebildet:

ADS-Read-Indication-Parameter	Bedeutung
Source-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter <b>ADS</b> des Devices)
Source-Port (PORT)	0x1000+Slave Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, muß bei Response wieder auftauchen
IndexGroup (IDXGRP)	Slot-Number (DPV1-Parameter)
IndexOffset (IDXOFFS)	Index (DPV1-Parameter)
Length (LENGTH)	Länge der auszulesenden Daten

Eine ADS-Read-Response wird wie folgt auf eine DPV1-MSAC\_C1-Read-Response abgebildet:

ADS-Read-Response-Parameter	Bedeutung
Destination-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter <b>ADS</b> des Devices)
Destination-Port (PORT)	0x1000+Slave Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, wie bei Indication
Result (RESULT)	Result des Read: 0 = kein Fehler, sonst: Bit 0-15 = Standard-ADS-Fehlercodes, Bit 16-23 = Error_Code_1, Bit 24-31 = Error_Code_2, siehe: <a href="#">DP-V1 Fehler Codes</a> [► 85].
Length (LENGTH)	Länge der gelesenen Daten
Data (DATAADDR)	gelesene Daten

**MSAC-C1 Write**

Eine DPV1-MSAC\_C1-Write-Indication wird wie folgt auf eine ADS-Write-Indication abgebildet:

ADS-Write-Indication-Parameter	Bedeutung
Source-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter <b>ADS</b> des Devices)
Source-Port (PORT)	0x1000+Slave Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, muß bei Response wieder auftauchen
IndexGroup (IDXGRP)	Slot-Number (DPV1-Parameter)
IndexOffset (IDXOFFS)	Index (DPV1-Parameter)
Length (LENGTH)	Länge der zu schreibenden Daten
Data (DATAADDR)	zu schreibende Daten

Eine ADS-Read-Response wird wie folgt auf eine DPV1-MSAC\_C1-Read-Response abgebildet:

ADS-Read-Response-Parameter	Bedeutung
Destination-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter <b>ADS</b> des Devices)
Destination-Port (PORT)	0x1000+Slave Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, wie bei Indication
Result (RESULT)	Result des Read: 0 = kein Fehler, sonst: Bit 0-15 = Standard-ADS-Fehlercode, Bit 16-23 = Error_Code_1, Bit 24-31 = Error_Code_2, siehe: <a href="#">DP-V1 Fehler Codes</a> [► 85].
Length (LENGTH)	Länge der gelesenen Daten

### 6.4.4 DP-V1 Fehler Codes

Bei einem fehlerhaften DPV1-Zugriff antwortet der Slave mit 4 Bytes Daten (alle nicht beschriebenen Werte sind nicht in der DPV1-Norm beschrieben und daher dem Handbuch des Slaves zu entnehmen).

Byte 0	DPV1-Dienst
0xD1	Data_Transport
0xD7	Initiate
0xDE	Read
0xDF	Write

Byte 1	Error_Decode
0x80	DPV1
0xFE	FMS
0xFF	HART

Byte 2		Error_Code_1
Error-Class (Bit 4-7)	Error-Code (Bit 0-3)	
0x0A	0x00	Application, Read Error
	0x01	Application, Write Error
	0x02	Application, Module Failure
	0x08	Application, Version Conflict
	0x09	Application, Feature Not Supported
0x0B	0x00	Access, Invalid Index
	0x01	Access, Write Length Error
	0x02	Access, Invalid Slot
	0x03	Access, Type Conflict
	0x04	Access, Invalid Area
	0x05	Access, State Conflict
	0x06	Access, Access Denied
	0x07	Access, Invalid Range
	0x08	Access, Invalid Parameter
	0x09	Access, Invalid Type
0x0C	0x00	Resource, Read Constrain Conflict
	0x01	Resource, Write Constrain Conflict
	0x02	Resource, Busy
	0x03	Resource, Unavailable

## 7 Anhang

### 7.1 Zubehör

Kabel und Stecker für die Verbindung der Profibus-Komponenten.

#### Stecker

Artikelnummer	Beschreibung
ZB3100	9 pol. D-Sub-Stecker bis 12 Mbaud, mit zuschaltbarem Abschlusswiderstand
ZB3101	9 pol. D-Sub-Stecker bis 12 Mbaud, mit Programmierschnittstelle
ZB3102	9 pol. D-Sub-Stecker bis 12 Mbaud, mit zuschaltbarem Abschlusswiderstand, inverser PROFIBUS-Stecker
ZS1031-3000	9 pol. D-Sub-Stecker bis 12 Mbaud, mit zuschaltbarem Abschlusswiderstand (andere Bauform)
ZS1031-3500	LWL-Stecker für BK35xx (im Lieferumfang der Buskoppler enthalten)

#### Kabel

Artikelnummer	Beschreibung
ZB3200	PROFIBUS-Kabel, 12 Mbaud, 1 x 2 x 0,64 mm <sup>2</sup> für feste Verlegung
ZB3300	PROFIBUS-Kabel, 12 Mbaud, 2 x 0,25; 3 x 0,75 mm <sup>2</sup> , 5-adrig, schleppkettentauglich

Lichtwellenleiter für Buskoppler BK3500 (PROFIBUS)

Artikelnummer	Beschreibung
Z1100	Leitung 1-adrig, 100-µm-Kunststoff-Faser, Durchmesser 2,2 mm
Z1101	Kunststoff-Lichtwellenleiter, Kern ø 1000 µm, PU-Schutzmantel ø 5,5 mm, Kevlar-Zugentlastung
Z1102	wie Z1101, auf Trommel, Länge = 500 m
Z1121	Kunststoff-Lichtwellenleiter, 1000-µm-Kern, 2-adrig, PU-Schutzmantel, Durchmesser 5,5 mm

## 7.2 Zertifizierungen

Prinzipiell sind alle Produkte der Embedded-PC-Familie CE, UL und EAC zertifiziert. Da sich aber die Produktfamilie ständig weiterentwickelt, kann hier keine Auflistung angegeben werden. Die aktuelle Auflistung der zertifizierten Produkte kann auf der Internetseite [www.beckhoff.de](http://www.beckhoff.de) unter Embedded-PC nachgelesen werden.

### **FCC Approvals for the United States of America**

#### **FCC: Federal Communications Commission Radio Frequency Interference Statement**

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

### **FCC Approval for Canada**

#### **FCC: Canadian Notice**

This equipment does not exceed the Class A limits for radiated emissions as described in the Radio Interference Regulations of the Canadian Department of Communications.

## 7.3 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.

### 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(0)5246/963-157  
Fax: +49(0)5246/963-9157  
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(0)5246/963-460  
Fax: +49(0)5246/963-479  
E-Mail: service@beckhoff.com

Weitere Support- und Serviceadressen finden Sie auf unseren Internetseiten unter <http://www.beckhoff.de>.

### Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49(0)5246/963-0  
Fax: +49(0)5246/963-198  
E-Mail: info@beckhoff.com

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten:

<http://www.beckhoff.de>

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.





Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.de](mailto:info@beckhoff.de)  
[www.beckhoff.de](http://www.beckhoff.de)