

Original-Handbuch | DE

CX7031

Embedded-PC für PROFIBUS-Slave



Inhaltsverzeichnis

1	Hinweise zur Dokumentation	7
1.1	Symbolerklärung	8
1.2	Ausgabestände der Dokumentation	9
2	Zu Ihrer Sicherheit	10
2.1	Bestimmungsgemäße Verwendung	10
2.2	Personalqualifikation	10
2.3	Sicherheitshinweise	10
2.4	Hinweise zur Informationssicherheit	11
3	Transport und Lagerung.....	12
4	Produktübersicht	13
4.1	Aufbau	14
4.2	Typenschild	15
4.3	Ethernet-Schnittstelle (X001)	16
4.4	USB-Schnittstelle (X002)	18
4.5	D-Sub-Buchse (X003)	19
4.6	MicroSD-Karte.....	20
5	Inbetriebnahme	21
5.1	Montage	21
5.1.1	Zulässige Einbaulagen beachten	21
5.1.2	Auf Hutschiene befestigen	23
5.1.3	MicroSD-Karte wechseln.....	24
5.1.4	Passive EtherCAT-Klemmen montieren	25
5.2	Spannungsversorgung	26
5.2.1	Embedded-PC anschließen	27
5.2.2	UL-Anforderungen.....	28
5.3	PROFIBUS: Anschluss und Verdrahtung.....	29
5.3.1	D-Sub-Buchse (X003).....	29
5.3.2	Anschlusstechnik	30
5.3.3	Topologie	32
5.3.4	Kabel und Steckverbinder für PROFIBUS	32
6	Multifunktions-I/Os.....	33
6.1	Digitale Eingänge	35
6.2	Digitale Ausgänge	36
6.3	Zähler-Modus	38
6.3.1	Betriebsart wählen	40
6.3.2	Ausgänge schalten.....	41
6.3.3	Zählerstand setzen.....	42
6.3.4	Grenzwert für Zähler festlegen.....	43
6.4	Inkremental-Encoder-Modus	44
6.4.1	Ausgänge schalten.....	46
6.4.2	Zählerstand latchen.....	47
6.4.3	Grenzwert für Zähler festlegen.....	48
6.5	Analog-Signal-Modus	49

6.6	PWM-Signal-Modus	50
6.6.1	PWM-Taktfrequenz und Tastverhältnis festlegen	52
6.6.2	Kanalsynchronisation einstellen	53
7	Konfiguration	54
7.1	Beckhoff Device Manager starten	54
7.2	Persistente Daten	55
7.3	NOVRAM	56
7.3.1	Retain-Handler anlegen	57
7.3.2	Variablen anlegen und verknüpfen	59
7.3.3	Variablen unter dem Retain-Handler löschen	62
7.4	Softwarekonfiguration	63
7.4.1	Benutzername und Passwort	63
7.4.2	IP-Adresse einstellen	64
7.4.3	Virtual Local Area Network (VLAN) konfigurieren	65
7.4.4	Image aktualisieren	65
7.4.5	Firmware für Multifunktions-I/Os aktualisieren	67
7.4.6	ESI-Gerätebeschreibung aktualisieren	68
8	TwinCAT	69
8.1	Erste Schritte	69
8.1.1	Mit CX70x0 verbinden	69
8.1.2	Multifunktions-I/Os scannen	71
8.1.3	ADS-Kommunikation herstellen	73
8.1.4	SPS-Projekt erstellen	75
8.1.5	Variablen verknüpfen	78
8.1.6	Konfiguration auf CX laden	79
8.2	CX7031 Slave anlegen	81
8.2.1	PROFIBUS-Adresse einstellen	82
8.2.2	Virtuellen Slave anlegen	84
8.2.3	Prozessdaten drehen	85
8.2.4	DPV1-Daten empfangen	86
8.2.5	PROFIBUS-Adresse über die SPS setzen	90
8.2.6	PROFIBUS-Geräte mit CX7031 simulieren	92
8.2.7	Parameter- und Konfigurationsdaten an die SPS weiterleiten	93
8.3	IP- und MAC-Adresse auslesen	96
8.4	Virtuelle Ethernet-Schnittstelle	96
8.5	CoE-Zugriff auf Multifunktions-I/Os	97
8.6	Netzteilklemme	99
8.7	Zyklus- und Bearbeitungszeiten	101
8.7.1	Zykluszeit für PROFIBUS-Konfiguration	101
8.7.2	Bearbeitungszeit im SPS-Programm messen	101
8.7.3	Real-Time-Clock (RTC)	101
8.7.4	I/O-Idle-Task:	102
8.7.5	NTP-Provider einsetzen	103
8.7.6	Zykluszeit von 250 µs	105
8.8	Funktionsbausteine	111

8.8.1	FB_CX70xx_RW_EEPROM	111
8.8.2	FB_CX70xx_ResetOnBoardIO	112
8.9	Wichtige Attribut-Pragmas	113
8.9.1	Attribut 'Tc2GvlVarNames'	113
8.9.2	Attribut 'pack_mode'	114
8.9.3	Attribut 'TcCallAfterOutputUpdate'	114
9	Fehlerbehandlung und Diagnose	118
9.1	Diagnose-LEDs	118
9.1.1	K-Bus	119
9.1.2	E-Bus	122
9.2	PROFIBUS-Diagnose	123
9.2.1	Slave-Diagnose	124
9.2.2	Individuelle Diagnosedaten	127
9.3	Diagnose der Multifunktions-I/Os	129
9.4	Speicherauslastung	130
9.5	Echtzeit und CPU-Auslastung	132
10	Technische Daten	134
11	Anhang	136
11.1	Komponenten Dritter	136
11.2	Zubehör	136
11.3	Zertifizierungen	136
	Tabellenverzeichnis	137
	Abbildungsverzeichnis	138

1 Hinweise zur Dokumentation

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

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

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

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

Disclaimer

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

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

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

Marken

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

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



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.

Fremdmarken

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

1.1 Symbolerklärung

In der Dokumentation werden folgende Warnhinweise verwendet. Lesen und befolgen Sie die Warnhinweise.

Warnhinweise, die vor Personenschäden warnen:

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnhinweise, die vor Sach- oder Umweltschäden warnen:

HINWEIS

Es besteht eine mögliche Gefährdung für Umwelt und Geräte.

Hinweise, die weitere Informationen oder Tipps anzeigen:



Dieser Hinweis gibt wichtige Informationen, die beim Umgang mit dem Produkt oder der Software helfen. Es besteht keine unmittelbare Gefahr für Produkt, Mensch und Umwelt.

1.2 Ausgabestände der Dokumentation

Version	Kommentar
1.0	Erste Version
1.1	Beschreibung zu VLAN, NTP-Provider und I/O-Idle-Task hinzugefügt.

2 Zu Ihrer Sicherheit

Lesen Sie das Sicherheitskapitel und halten Sie die Hinweise ein, um sich vor Personenschäden und Sachschäden zu schützen.

Haftungsbeschränkungen

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

Darüber hinaus werden folgende Punkte aus der Haftung der Beckhoff Automation GmbH & Co. KG ausgeschlossen:

- Nichtbeachtung dieser Dokumentation.
- Nichtbestimmungsgemäße Verwendung.
- Einsatz von nicht ausgebildetem Fachpersonal.
- Verwendung nicht zugelassener Ersatzteile.

2.1 Bestimmungsgemäße Verwendung

Der Embedded-PC ist ein Steuerungssystem für den Einsatz im Maschinen- und Anlagenbau zur Automatisierung, Visualisierung und Kommunikation. Der Embedded-PC ist für den Einbau in einen Schaltschrank oder Klemmenkasten vorgesehen und wird zusammen mit Bus- oder EtherCAT-Klemmen dazu verwendet, um digitale und analoge Signale von Sensoren aufzunehmen und an Aktoren auszugeben oder an übergeordnete Steuerungen weiterzuleiten.

Der Embedded-PC ist für ein Arbeitsumfeld entwickelt, welches der Schutzklasse IP20 genügt. Es besteht Fingerschutz und Schutz gegen feste Fremdkörper bis 12,5 mm, jedoch kein Schutz gegen Wasser. Der Betrieb der Geräte in nasser und staubiger Umgebung ist nicht gestattet, sofern nicht anders angegeben. Die angegebenen Grenzwerte für elektrische- und technische Daten müssen eingehalten werden.

Nicht bestimmungsgemäße Verwendung

Der Embedded-PC ist nicht für den Betrieb in folgenden Bereichen geeignet:

- In explosionsgefährdeten Bereichen.
- In Bereichen mit einer aggressiven Umgebung, die z.B. mit aggressiven Gasen oder Chemikalien angereichert ist.
- Im Wohnbereich. Im Wohnbereich müssen die entsprechenden Normen und Richtlinien für Störaussendungen eingehalten und die Geräte in Gehäuse oder Schaltkästen mit entsprechender Schirmdämpfung eingebaut werden.

2.2 Personalqualifikation

Alle Arbeitsschritte an der Beckhoff Soft- und Hardware dürfen nur vom Fachpersonal mit Kenntnissen in der Steuerungs- und Automatisierungstechnik durchgeführt werden. Das Fachpersonal muss über Kenntnisse in der Administration des eingesetzten Industrie-PCs und des jeweils eingesetzten Netzwerks verfügen.

Alle Eingriffe müssen mit Kenntnissen in der Steuerungs-Programmierung durchgeführt werden und das Fachpersonal muss die aktuellen Normen und Richtlinien für das Automatisierungsumfeld kennen.

2.3 Sicherheitshinweise

Folgende Sicherheitshinweise müssen während der Montage, der Arbeit mit Netzwerken und der Arbeit mit Software beachtet werden.

Montage

- Arbeiten Sie nicht an Geräten unter Spannung. Schalten Sie immer die Spannungsversorgung für das Gerät ab bevor Sie es montieren, Störungen beheben oder Wartungsarbeiten durchführen. Sichern Sie das Gerät gegen ein unbeabsichtigtes Einschalten ab.
- Beachten Sie die Unfallverhütungsvorschriften, die für Ihre Maschine zutreffend sind (z.B. die BGV A 3, Elektrische Anlagen und Betriebsmittel).
- Achten Sie auf einen normgerechten Anschluss und vermeiden Sie Gefahren für das Personal. Verlegen Sie die Daten- und Versorgungsleitungen normgerecht und achten Sie auf die korrekte Anschlussbelegung.
- Beachten Sie die für Ihre Anwendung zutreffenden EMV-Richtlinien.
- Vermeiden Sie die Verpolung der Daten- und Versorgungsleitungen, da dies zu Schäden an den Geräten führen kann.
- In den Geräten sind elektronische Bauteile integriert, die Sie durch elektrostatische Entladung bei Berührung zerstören können. Beachten Sie die Sicherheitsmaßnahmen gegen elektrostatische Entladung entsprechend DIN EN 61340-5-1/-3.

Arbeiten mit Netzwerken

- Beschränken Sie den Zugriff zu sämtlichen Geräten auf einen autorisierten Personenkreis.
- Ändern Sie die standardmäßig eingestellten Passwörter und verringern so das Risiko, dass Unbefugte Zugriff erhalten.
- Schützen Sie die Geräte mit einer Firewall.
- Wenden Sie die Vorgaben zur IT-Sicherheit nach der IEC 62443 an, um den Zugriff und die Kontrolle auf Geräte und Netzwerke einzuschränken.

2.4 Hinweise zur Informationssicherheit

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

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

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

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

3 Transport und Lagerung

Transport

HINWEIS

Kurzschluss durch Feuchtigkeit

Feuchtigkeit kann sich bei Transporten in kalter Witterung oder bei extremen Temperaturunterschieden bilden.

Achten Sie darauf, dass sich keine Feuchtigkeit im Embedded-PC niederschlägt (Betauung) und gleichen Sie ihn langsam der Raumtemperatur an. Schalten Sie den Embedded-PC bei Betauung erst nach einer Wartezeit von mindestens 12 Stunden ein.

Trotz des robusten Aufbaus sind die eingebauten Komponenten empfindlich gegen starke Erschütterungen und Stöße. Der Transport eines Schaltschranks mit eingebautem Embedded-PC kann eine zu hohe Stoßbelastung für den Embedded-PC darstellen.

- Schützen Sie das Gerät bei Transporten vor großer mechanischer Belastung.
- Durch geeignete Verpackung des Industrie-PCs, wie die Originalverpackung, kann die Erschütterungsfestigkeit beim Transport verbessert werden.
- Versenden Sie den Embedded-PC in der Originalverpackung und einer zusätzlichen Umverpackung.

Tab. 1: Abmessungen und Gewicht.

	CX7031
Abmessungen (B x H x T)	49 mm x 100 mm x 73 mm
Gewicht	ca. 142 g

Lagerung

- Lagern Sie den Embedded-PC in der Originalverpackung.

4 Produktübersicht

Der Embedded-PC CX7031 verfügt über einen ARM-Cortex™-M7-Singlecore-Prozessor mit 480 MHz und hat die folgende Grundausstattung:

- einen MicroSD-Kartenslot mit integrierter 512 MB MicroSD-Karte,
- eine Ethernet-Schnittstelle (10/100 MBit/s, RJ45),
- eine USB-Schnittstelle (max. 12 Mbit/s, max. 100 mA),
- sowie integrierte Multifunktions-I/Os.

Der CX7031 wird mit TwinCAT 3 über die Ethernet-Schnittstelle programmiert. Zusätzlich steht der Beckhoff Device Manager als Webinterface für die Konfiguration des CX7031 zur Verfügung.

Der CX7031 besitzt eine PROFIBUS-Slave-Schnittstelle.

Multifunktions-I/Os

Als Besonderheit verfügt die CX7000-Serie über acht integrierte Multifunktionseingänge und vier integrierte Multifunktionsausgänge.

- 8 digitale Eingänge, 24 V DC, Filter 3 ms, Typ 3, 1-Leitertechnik
- 4 digitale Ausgänge, 24 V DC, 0,5 A, 1-Leitertechnik

Die integrierten Multifunktions-I/Os des CX7031 lassen sich über TwinCAT 3 für andere Betriebsarten (Modi) konfigurieren, sodass auch schnelles Zählen oder die Analogwertverarbeitung ermöglicht wird:

- Zähler-Modus: 1 x Zähler-Digitaleingang 100 kHz, 1 x Digitaleingang als Auf-/Abwärts-Zähler 20 kHz, 2 x Zähler-Digitalausgang
- Inkremental-Encoder-Modus: 2 x Digitaleingang für 250-kHz-Encoder-Signal (A-/B-Eingang), 2 x Encoder-Digitalausgang
- Analog-Signal-Modus: 2 x Digitaleingang konfiguriert als Analogeingang 0 bis 10 V, 12 Bit Auflösung bei 16 Bit Darstellung
- PWM-Signal-Modus: 2 x Digitalausgang konfiguriert für PWM-Signal, 15 Hz...100 kHz

Netzteilklemme

Auf der rechten Seite können wahlweise EtherCAT-Klemmen (E-Bus) oder Busklemmen (K-Bus) angereiht werden; der CX7031 erkennt in der Hochlaufphase automatisch, welches System angeschlossen ist. Sollen weitere elektrische Signale verarbeitet werden, kann somit der CX7031 ergänzend zu den integrierten I/Os bedarfsgerecht und äußerst flexibel durch EtherCAT-Klemmen oder Busklemmen erweitert werden.

Firmware

Als Betriebssystem bzw. Firmware wird das Echtzeitbetriebssystem TwinCAT/RTOS eingesetzt, welches auf FreeRTOS basiert. Beachten Sie, dass TwinCAT/RTOS ein geschlossenes System ist und eigene Softwareinstallationen nicht möglich sind. Damit ist eine gewisse Sicherheit gegeben, da Fremdsoftware wie Viren oder ähnliches nicht installiert werden können und der CX7031 an ein Netzwerk angeschlossen werden kann. Der CX7031 kann ab TwinCAT 3.1 Build 4024.47 verwendet werden. Folgende TC 3 Funktionen sind enthalten und lizenziert:

- TC1000 TC3 ADS
- TC1100 TC3 IO
- TC1200 TC3 PLC
- TF4100 TC3 Controller Toolbox
- TF4110 TC3 Temperature-Controller
- TF6255 TC3 Modbus-RTU
- TF6340 TC3 Serial-Communication
- TF6701 | TwinCAT 3 IoT Communication (MQTT[®])
- TF6730 | TwinCAT 3 IoT Communicator[®])

^{*)} Image-Version 114606 und TwinCAT 3 XAE 4024.47 oder höher erforderlich.

Die Open-Source-Lizenzen sind als ZIP-Datei auf der MicroSD-Karte einsehbar.

4.1 Aufbau

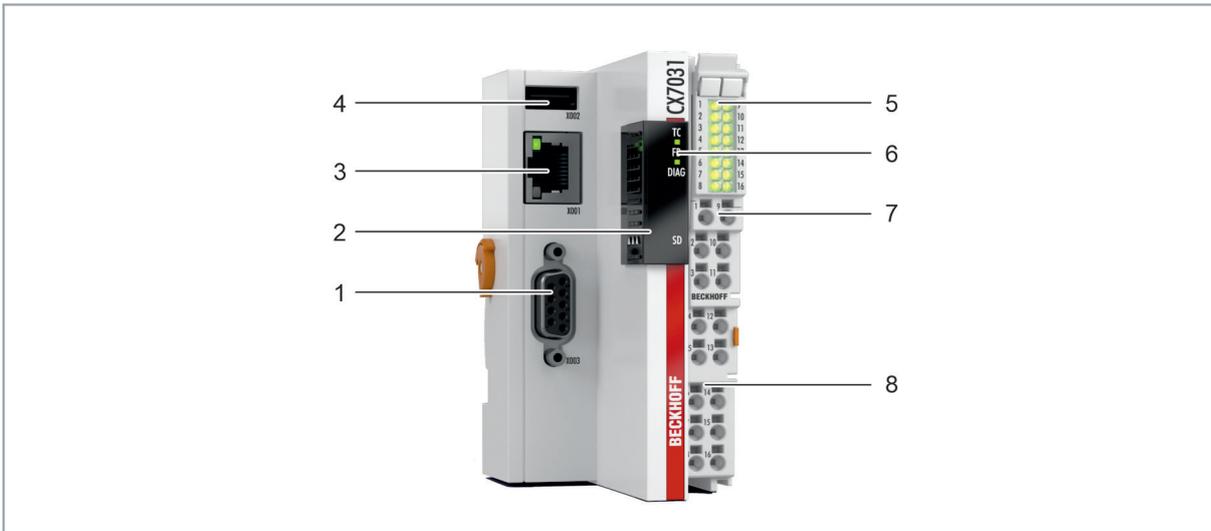


Abb. 1: Beispielaufbau eines Embedded-PCs CX7031.

Tab. 2: Legende zum Aufbau des CPU-Grundmoduls

Nr.	Komponente	Beschreibung
1	D-Sub-Buchse (X003).	PROFIBUS-Slave-Schnittstelle.
2	MicroSD-Kartenslot (unter der Abdeckung).	Steckplatz für industrietaugliche MicroSD-Karten. Speicherplatz für Firmware und TwinCAT 3 Projekte.
3	Ethernet-Schnittstelle (X001)	Für den Anschluss an lokale Netzwerke. Dient als Programmierschnittstelle.
4	USB-Schnittstelle (X002)	Schnittstelle für zusätzlichen USB-Datenspeicher.
5	I/O-Status-LEDs	Diagnose der Spannungsversorgung für Embedded-PC und Klemmbus. Status der E-Bus bzw. K-Bus Kommunikation und Multifunktions-I/Os.
6	Diagnose-LEDs	1 x TwinCAT-Status, 1 x Flash-Zugriff, 1 x Error-LED.
7	Federkraftklemmen, +24 V und 0V	Spannungsversorgung (Us) für Embedded-PC.
8	Federkraftklemmen, +24 V und 0V	Spannungsversorgung (Up) für integrierte Multifunktions-I/Os und Busklemmen über die Powerkontakte.

4.2 Typenschild

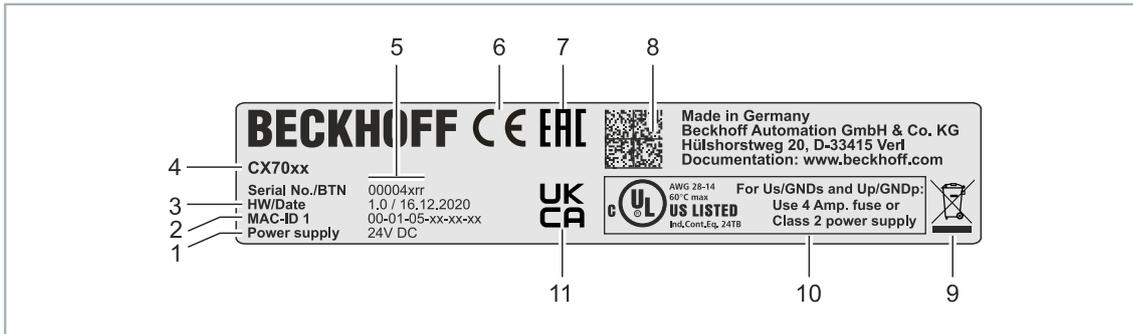


Abb. 2: Typenschild Beispielsicht.

Tab. 3: Informationen auf dem Typenschild.

Nr.	Beschreibung
1	Spannungsversorgung 24 V DC.
2	MAC-Adressen der eingebauten Ethernet-Schnittstelle.
3	Hardwarestand und Herstelldatum.
4	Produktbezeichnung zur Identifikation des Embedded-PCs.
5	Seriennummer/ Beckhoff Traceability Number (BTN) zur eindeutigen Identifizierung des Produkts. Der Hostname wird aus BTN- und der Seriennummer/ Beckhoff Traceability Number (BTN) gebildet. Beispiel: Aus der BTN 00004xrr ergibt sich der Hostname BTN-00004xrr .
6	CE-Kennzeichnung
7	EAC-Kennzeichnung
8	Maschinenlesbare Information in Form eines Data-Matrix-Codes (DMC, Code-Schema ECC200) der zur besseren Identifikation und Verwaltung genutzt werden kann.
9	Kennzeichnung mit Abfalltonne. Nicht im Hausmüll entsorgen.
10	UL-Kennzeichnung mit vorgeschriebenen Angaben zu Spannungsversorgung, Sicherung, Temperatur und Kabelquerschnitten.
11	UKCA-Kennzeichnung

4.3 Ethernet-Schnittstelle (X001)

Sie können den Embedded-PC CX7031 über die Ethernet-Schnittstelle X001 programmieren und in Betrieb nehmen. Die Ethernet-Schnittstelle erreicht Geschwindigkeiten von 10 / 100 Mbit/s.

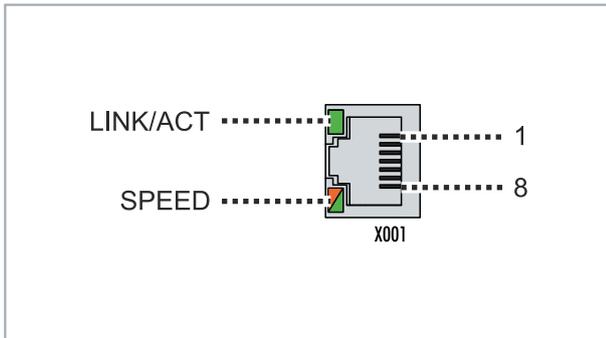


Abb. 3: Ethernet-Schnittstelle X001.

Die LEDs an der linken Seite der Schnittstelle zeigen den Status der Verbindung an. Die obere LED (LINK/ACT) zeigt an, ob die Schnittstelle mit einem Netzwerk verbunden ist. Ist dies der Fall, leuchtet die LED grün auf. Wenn Daten auf der Schnittstelle übertragen werden, blinkt die LED.

Die untere LED (SPEED) zeigt die Geschwindigkeit der Verbindung an. Ist die Geschwindigkeit 10 Mbit/s, leuchtet die LED nicht. Bei 100 Mbit/s leuchtet die LED orange.

Tab. 4: Ethernet-Schnittstelle X001, PIN-Belegung.

PIN	Signal	Beschreibung
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	connected	reserviert
5		
6	RD -	Receive -
7	connected	reserviert
8		

Übertragungsstandards

10Base5

Das Übertragungsmedium für 10Base5 ist ein dickes Koaxialkabel (Yellow Cable) mit einer max. Übertragungsgeschwindigkeit von 10 Mbaud und einer Linien-Topologie mit Abzweigen (Drops), an die jeweils ein Teilnehmer angeschlossen wird. Da hier alle Teilnehmer an einem gemeinsamen Übertragungsmedium angeschlossen sind, kommt es bei 10Base5 zwangsläufig häufig zu Kollisionen.

10Base2

10Base2 (Cheaper net) ist eine Weiterentwicklung von 10Base5 und hat den Vorteil, dass dieses Koaxialkabel billiger und durch eine höhere Flexibilität einfacher zu verlegen ist. Es können mehrere Geräte an eine 10Base2-Leitung angeschlossen werden. Häufig werden die Abzweige eines 10Base5-Backbones als 10Base2 ausgeführt.

10BaseT

Beschreibt ein Twisted-Pair-Kabel für 10 Mbaud. Hierbei wird das Netz sternförmig aufgebaut, so dass nun nicht mehr jeder Teilnehmer am gleichem Medium hängt. Dadurch führt ein Kabelbruch nicht mehr zum Ausfall des gesamten Netzes. Durch den Einsatz von Switches als Sternkoppler können Kollisionen vermindert oder bei Voll-Duplex Verbindungen auch vollständig vermieden werden.

100BaseT

Twisted-Pair-Kabel für 100 MBaud. Für die höhere Datengeschwindigkeit ist eine bessere Kabelqualität und die Verwendung entsprechender Hubs oder Switches erforderlich.

10BaseF

Der Standard 10BaseF beschreibt mehrere Lichtwellenleiter-Varianten.

Kurzbezeichnung der Kabeltypen für 10BaseT und 100BaseT

Twisted-Pair Kupferkabel für sternförmige Topologie, wobei der Abstand zwischen zwei Geräten 100 Meter nicht überschreiten darf.

UTP

Unshielded Twisted-Pair (nicht abgeschirmte, verdrehte Leitung)

Dieser Kabeltyp gehört zur Kategorie 3 und sind für industrielle Umgebungen nicht empfehlenswert.

S/UTP

Screened/Unshielded Twisted-Pair (mit Kupfergeflecht abgeschirmte, verdrehte Leitung)

Besitzen einen Gesamtschirm aus einem Kupfergeflecht zur Reduktion der äußeren Störeinflüsse. Dieses Kabel wird zum Einsatz mit den Buskopplern empfohlen.

FTP

Foilesshielded Twisted-Pair (mit Alufolie abgeschirmte, verdrehte Leitung)

Dieses Kabel hat eine alukaschierten Kunststoff-Folie-Gesamtschirm.

S/FTP

Screened/Foilesshielded Twisted-Pair (mit Kupfergeflecht und Alufolie abgeschirmte, verdrehte Leitung)

Besitzt einen alukaschierten Gesamtschirm mit einem darüber liegenden Kupfergeflecht. Solche Kabel können eine Störleistungsunterdrückung bis zu 70dB erreichen.

STP

Shielded Twisted-Pair (abgeschirmte, verdrehte Leitung)

Beschreibt ein Kabel mit Gesamtschirm ohne weitere Angabe der Art der Schirmung.

S/STP

Screened/Shielded Twisted-Pair (einzeln abgeschirmte, verdrehte Leitung)

Ein solche Bezeichnung kennzeichnet ein Kabel mit einer Abschirmung für jedes Leitungspaar sowie einen Gesamtschirm.

ITP

Industrial Twisted-Pair

Ist von Aufbau dem S/STP ähnlich, besitzt allerdings im Gegensatz zum S/STP nur 2 Leitungspaare.

4.4 USB-Schnittstelle (X002)

An die USB-Schnittstelle kann ein USB-Speicherstick angeschlossen und als zusätzlicher Speicher verwendet werden. Die USB-Schnittstelle unterstützt Übertragungsgeschwindigkeiten bis 12 Mbit/s und es können nicht mehr als 100 mA abgegeben werden. Der Dateizugriff erfolgt aus TwinCAT bzw. dem SPS-Programm mit Hilfe der dazugehörigen Funktionsbausteine. Es können keine anderen Geräte an die USB-Schnittstelle angeschlossen und verwendet werden.

Die gleiche Funktionsweise lässt sich für Dateizugriffe auf die MicroSD-Karte nutzen. Benutzen Sie für Zugriffe auf die MicroSD-Karte `C:\` und für Zugriffe auf den USB-Speicherstick `D:\` als Laufwerksbuchstaben.

Funktionsbausteine für Dateizugriffe

Mit den nachfolgend aufgeführten Funktionsbausteinen können lokal auf dem PC Dateien aus der SPS heraus bearbeitet werden. Durch die AMS-Netzwerkadresse wird das TwinCAT-Zielsystem identifiziert. Durch diesen Mechanismus ist es unter anderem möglich, Dateien auf anderen TwinCAT-Systemen des Verbundes anzulegen bzw. zu bearbeiten. Der Zugriff auf Dateien besteht aus drei aufeinanderfolgenden Phasen:

1. Öffnen der Datei
2. Lesender oder schreibender Zugriff auf die geöffnete Datei
3. Schließen der Datei

Das Öffnen der Datei dient dazu, eine temporäre Verbindung zwischen der externen Datei, von der zunächst nur der Name bekannt ist, und dem laufenden Programm herzustellen. Das Schließen der Datei dient dazu, das Ende der Bearbeitung anzuzeigen und sie in einen definierten Ausgangszustand für die Bearbeitung durch andere Programme zu versetzen.

Name	Beschreibung
FB_EOF	Testen auf Dateiende
FB_FileOpen	Öffnen einer Datei
FB_FileClose	Schließen einer Datei
FB_FileGets	String aus einer Text-Datei lesen
FB_FilePuts	Nullterminierten-String in eine Text-Datei schreiben
FB_FileRead	Lesen aus einer Datei
FB_FileWrite	Schreiben in eine Datei
FB_FileSeek	Verstellen des Dateizeigers
FB_FileTell	Ermitteln der aktuellen Position des Dateizeigers
FB_FileDelete	Löschen einer Datei
FB_FileRename	Umbenennen einer Datei
FB_CreateDir	Erstellen eines neuen Verzeichnisses
FB_RemoveDir	Löschen eines Verzeichnisses

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_System (System)

4.5 D-Sub-Buchse (X003)

Pin 6 überträgt 5 V_{DC} 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.

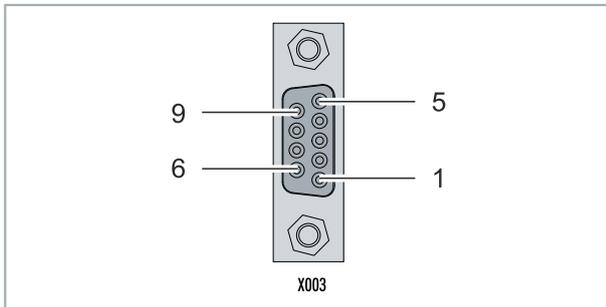


Abb. 4: PROFIBUS-Schnittstelle X003.

Die PROFIBUS-Busleitung wird über einen 9-polige D-Sub-Buchse mit folgender Belegung angeschlossen:

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

Leitungsfarben

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

4.6 MicroSD-Karte

In der Grundausstattung enthält der CX7031 eine 512 MB MicroSD-Karte. Sie können den Embedded-PC optional mit einer größeren 16 GB MicroSD-Karte bestellen.

Die verwendeten Karten sind SLC-Speicher mit erweitertem Temperaturbereich für industrielle Anwendungen. Verwenden Sie ausschließlich von Beckhoff freigegebene MicroSD-Karten.

Bestellbezeichnung	Kapazität	Beschreibung
CX1900-0131	16 GB	MicroSD-Karte (SLC-Speicher) mit erweitertem Temperaturbereich für industrielle Anwendungen anstelle der 512 MB Karte (Bestelloption)

Bestellbezeichnung	Kapazität	Beschreibung
CX1900-0122	512 MB	MicroSD-Karte (SLC-Speicher) mit erweitertem Temperaturbereich für industrielle Anwendungen als Ersatzteil.
CX1900-0132	16 GB	

5 Inbetriebnahme

5.1 Montage

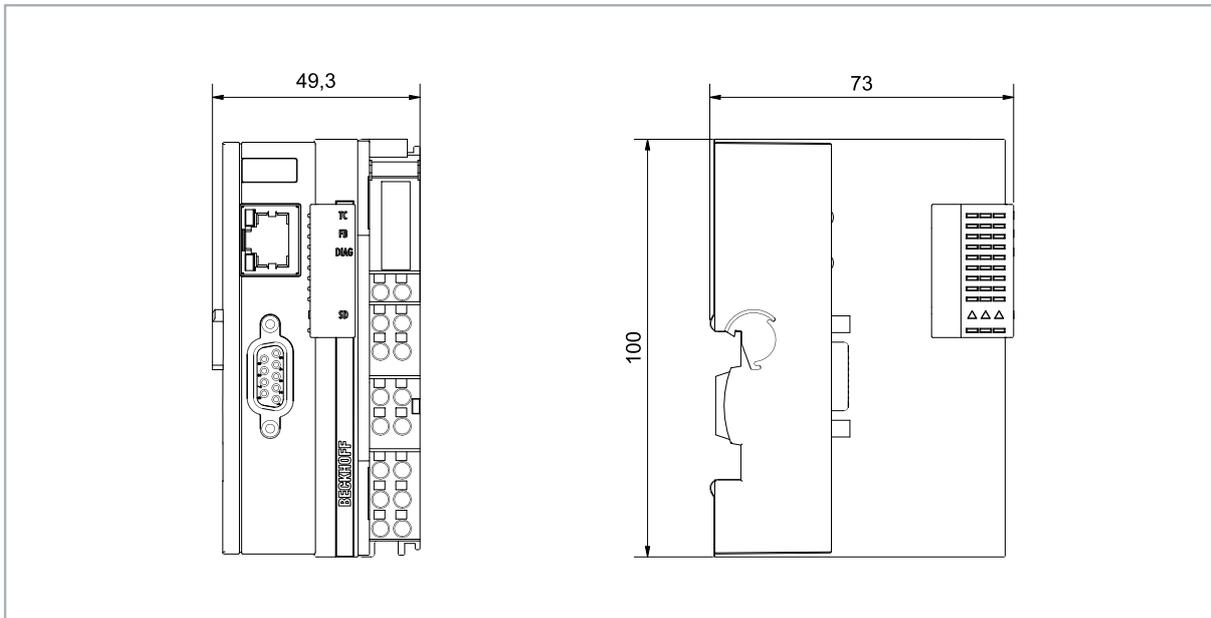


Abb. 5: Embedded-PC CX70xx, Abmessungen.

5.1.1 Zulässige Einbaulagen beachten

HINWEIS

Überhitzung

Bei einer falsch gewählten Einbaulage und nicht eingehaltenen Mindestabständen kann der Embedded-PC überhitzen. Halten Sie die maximale Umgebungstemperatur von 60°C und die Montagevorschriften ein.

Montieren Sie den Embedded-PC waagrecht im Schaltschrank auf einer Hutschiene, damit die Wärme optimal abgeführt wird.

Beachten Sie folgende Vorgaben für den Schaltschrank:

- Betreiben Sie den Embedded-PC nur bei Umgebungstemperaturen von -25 °C bis 60 °C. Messen Sie dazu die Temperatur unter dem Embedded-PC in einem Abstand von 30 mm zu den Kühlrippen, um die Umgebungstemperatur korrekt zu ermitteln.
- Halten Sie die Mindestabstände von 30 mm ober- und unterhalb des Embedded-PCs ein.
- Weitere elektrische Geräte beeinflussen die Wärmeentwicklung im Schaltschrank. Wählen Sie eine passende Schaltschrankgröße abhängig vom Anwendungsfall oder sorgen Sie dafür, dass überschüssige Wärme aus dem Schaltschrank abtransportiert wird.

Der Embedded-PC muss waagrecht auf die Hutschiene montiert werden. Die Lüftungsöffnungen befinden sich auf der Gehäuseunter- und Gehäuseoberseite. Auf diese Weise kommt ein optimaler Luftstrom zustande, der den Embedded-PC in vertikaler Richtung durchströmt. Zusätzlich ist ein Freiraum von mindestens 30 mm oberhalb und unterhalb des Embedded-PCs erforderlich, um eine ausreichende Belüftung zu gewährleisten.

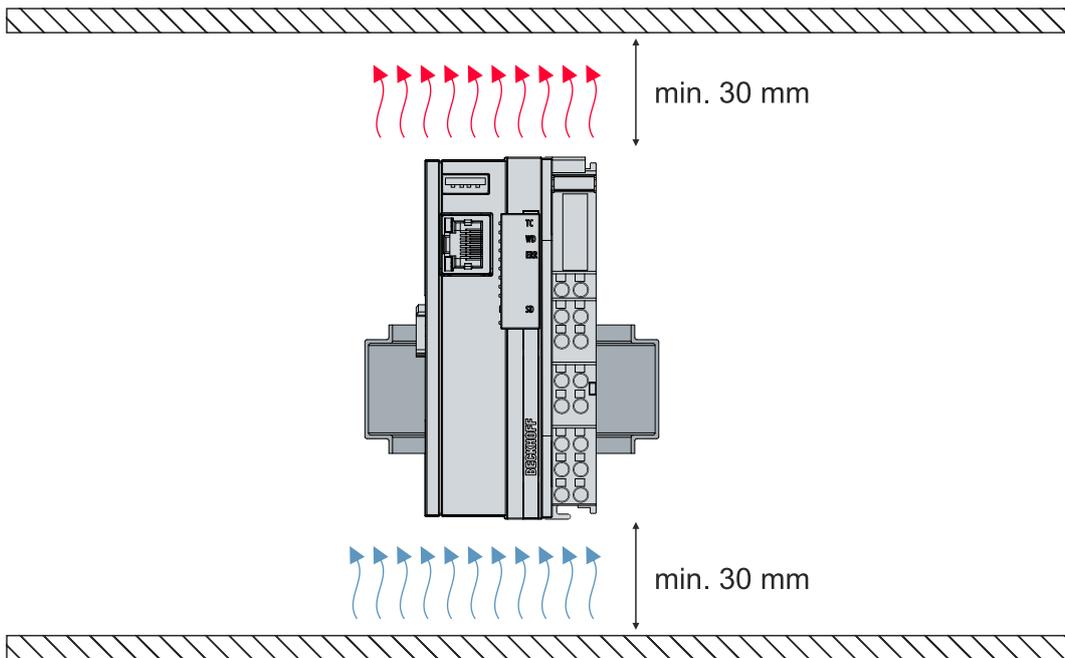


Abb. 6: Embedded-PC CX70xx, zulässige Einbaulage.

Wenn Vibrationen und Stöße in der gleichen Richtung verlaufen wie die Hutschiene, muss der Embedded-PC zusätzlich mit einer Halterung fixiert werden, damit er nicht verrutscht.

Einbaulagen mit eingeschränktem Temperaturbereich bis 45 °C

Sie können den Embedded-PC auch senkrecht oder liegend auf der Tragschiene montieren. Beachten Sie dabei, dass Sie den Embedded-PC dann nur bis zu einer Umgebungstemperatur von 45 °C betreiben können.

Achten Sie darauf, dass Busklemmen, die an den Embedded-PCs angeschlossen werden, für den senkrechten oder liegenden Betrieb ausgelegt sind.

Einschränkungen bei E-Bus/K-Bus-Strom

Der maximale E-Bus/K-Bus-Strom variiert abhängig von der gewählten Einbaulage und Umgebungstemperatur.

Tab. 5: Maximaler E-Bus/K-Bus-Strom abhängig von Einbaulage und Umgebungstemperatur.

E-Bus/K-Bus-Strom	Einbaulage	Umgebungstemperatur
max. 1,5 A	beliebig	-25...45 °C
max. 1,3 A	horizontal	-25...55 °C
max. 1 A	beliebig	-25...55 °C
max. 1 A	horizontal	-25...60 °C

5.1.2 Auf Hutschiene befestigen

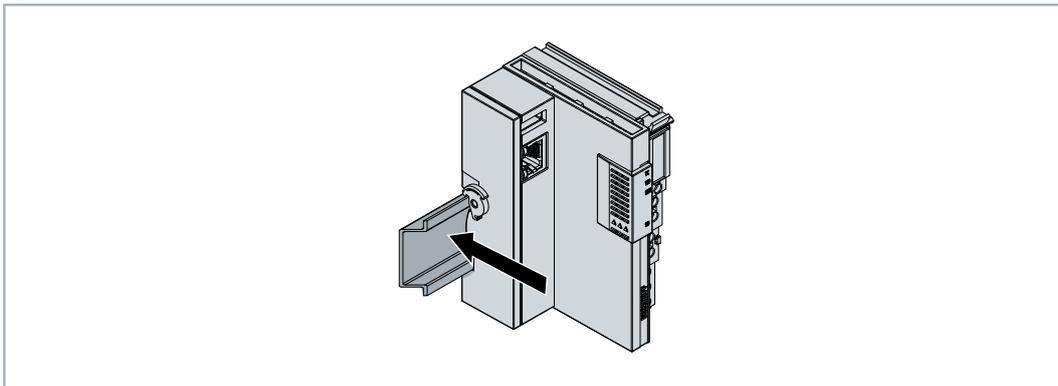
Das Gehäuse ist so konstruiert, dass der Embedded-PC an die Hutschiene gehalten und auf diese eingerastet werden kann.

Voraussetzungen:

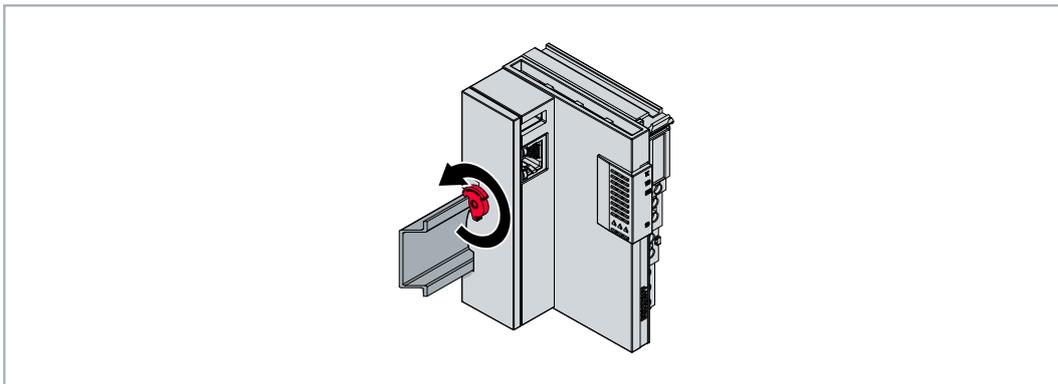
- Hutschiene von Typ TS35/7.5 oder TS35/15 nach DIN EN 60715.

Befestigen Sie den Embedded-PC wie folgt auf der Hutschiene:

1. Setzen Sie den Embedded-PC auf die Hutschiene. Drücken Sie den Embedded-PC leicht an die Hutschiene, bis es leise klickt und der Embedded-PC eingerastet ist.



2. Verriegeln Sie anschließend die Arretierung auf der linken Seite des Embedded-PCs.
3. Drehen Sie die Arretierung gegen den Uhrzeigersinn, bis die Arretierung leise klickt und einrastet.



- ⇒ Sie haben den Embedded-PC erfolgreich montiert. Überprüfen Sie noch mal die korrekte Montage und ob der Embedded-PC auf der Hutschiene eingerastet ist.

5.1.3 MicroSD-Karte wechseln

● Datenverlust

i MicroSD-Karten werden im Betrieb stark beansprucht und müssen viele Schreibzyklen und extreme Umweltbedingungen aushalten. MicroSD-Karten anderer Hersteller können ausfallen, was zu Datenverlust führt.

Verwendet Sie ausschließlich industrietaugliche MicroSD-Karten die von Beckhoff geliefert werden.

Der MicroSD-Kartenslot ist für eine industrietaugliche MicroSD-Karte vorgesehen. Auf der MicroSD-Karte wird die Firmware des Embedded-PCs gespeichert. Die MicroSD-Karte kann bei Bedarf aus TwinCAT 3 heraus beschrieben werden und dadurch benutzerdefinierte Daten gespeichert werden.

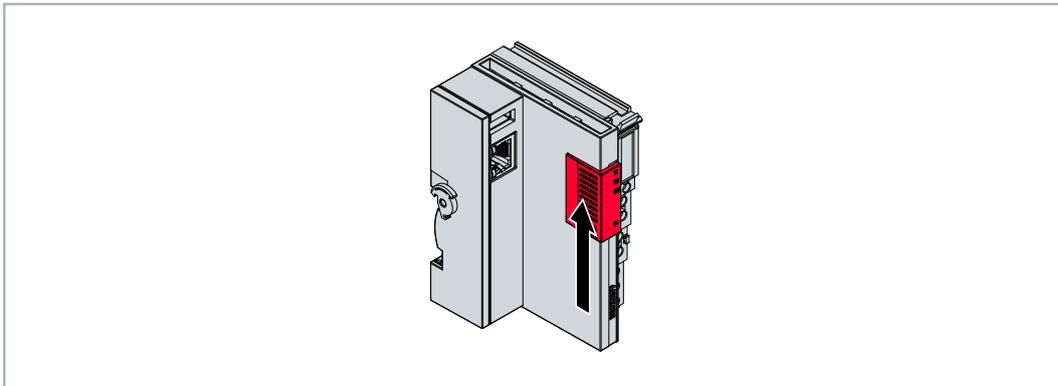
Die Auswurfmechanik wird nach dem Push-Push-Prinzip betätigt. Im Folgenden wird gezeigt, wie die MicroSD-Karte gewechselt wird.

Voraussetzungen:

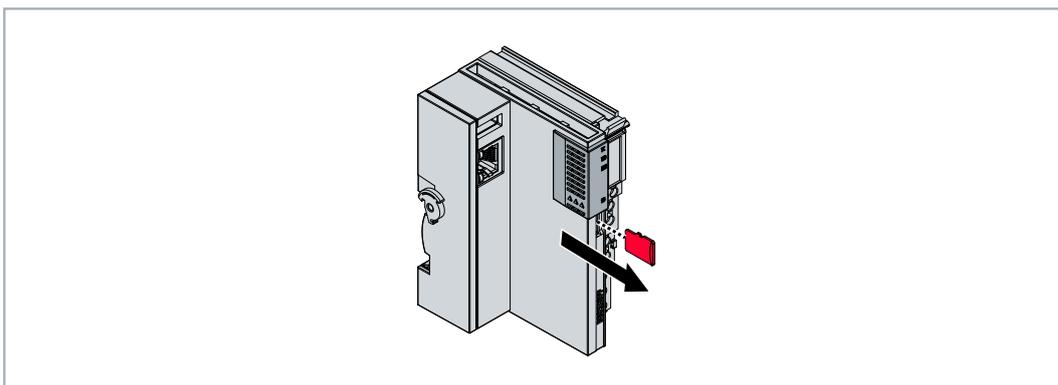
- Der Embedded-PC muss ausgeschaltet sein. Die MicroSD-Karte darf nur im ausgeschalteten Zustand ein- oder ausgebaut werden.

MicroSD-Karte wechseln

1. Schieben Sie die schwarze Abdeckung nach oben.



2. Drücken Sie leicht auf die MicroSD-Karte.
3. Die Karte wird mit einem leisen Klicken entriegelt und um ca. 2-3 mm aus dem Gehäuse gehoben.



4. Schieben Sie die neue MicroSD-Karte mit den Kontakten voran in den Kartenslot. Die Kontakte zeigen nach rechts.
5. Die MicroSD-Karte rastet mit einem leisen Klicken ein.
⇒ Die Karte sitzt richtig, wenn sie sich ca. 1 mm tiefer als die Frontseite des Gehäuses befindet.

5.1.4 Passive EtherCAT-Klemmen montieren

Falsch montierte passive EtherCAT-Klemmen

i Das E-Bus Signal zwischen einem Embedded-PC und den EtherCAT-Klemmen kann durch falsch montierte passive EtherCAT-Klemmen geschwächt werden.

Montieren Sie passive EtherCAT-Klemmen nicht direkt an das Netzteil.

EtherCAT-Klemmen, die nicht aktiv am Datenaustausch teilnehmen, werden als passive Klemmen bezeichnet. Dadurch haben passive EtherCAT-Klemmen kein Prozessabbild und benötigen keinen Strom aus dem Klemmbus (E-Bus).

Passive EtherCAT-Klemmen (z.B. eine EL9195) können Sie in TwinCAT erkennen. Die EtherCAT-Klemme wird im Strukturbaum ohne Prozessabbild angezeigt und der Wert in der Spalte „E-Bus (mA)“ verändert sich im Vergleich zu der vorangehenden EtherCAT-Klemme nicht.

Number	Box Name	Ad...	Type	In Size	Out Size	E-Bus (mA)
1	Term 7 (EK1200)		EK1200			
2	Term 8 (EL2828)	1001	EL2828		1.0	1890
3	Term 9 (EL2828)	1002	EL2828		1.0	1780
4	Term 10 (EL9195)		EL9195			1780
5	Term 11 (EL2828)	1003	EL2828		1.0	1670
6	Term 12 (EL9011)		EL9011			

Abb. 7: Passive EtherCAT-Klemme in TwinCAT identifizieren.

In den technischen Daten einer EtherCAT-Klemme können Sie unter dem Eintrag „Stromaufnahme aus dem E-Bus“ nachlesen, ob eine bestimmte EtherCAT-Klemme Strom aus dem Klemmbus (E-Bus) benötigt.

Die folgende Abbildung zeigt die zulässige Montage einer passiven EtherCAT-Klemme. Die passive EtherCAT-Klemme wurde nicht direkt an das Netzteil angereicht.

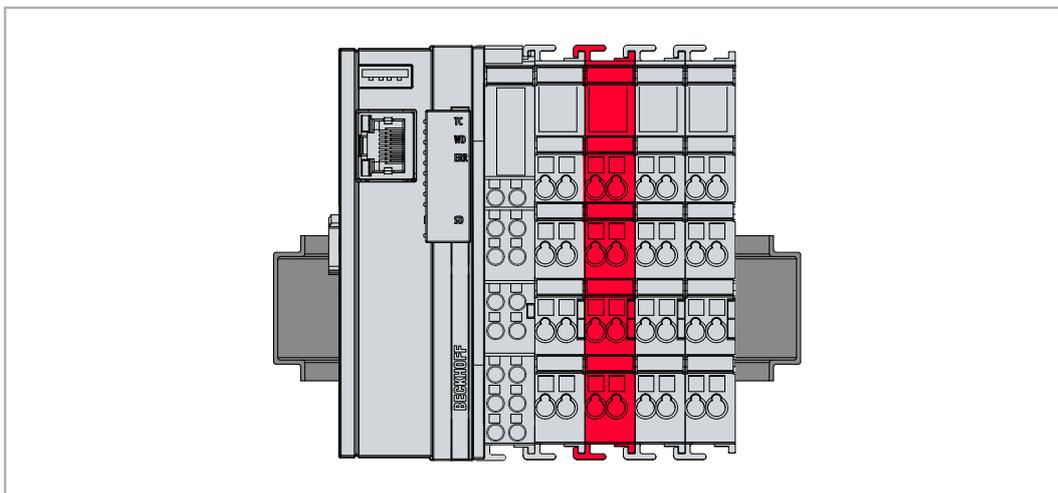


Abb. 8: Passive EtherCAT-Klemmen, zulässige Montage.

5.2 Spannungsversorgung

HINWEIS

Schäden an den Embedded-PCs

Die Embedded-PCs können während der Verdrahtung beschädigt werden. Schließen Sie die Leitungen für die Spannungsversorgung nur im spannungsfreien Zustand an.

Für die Spannungsversorgung der Netzteilklemme ist eine externe Spannungsquelle erforderlich, die eine 24 V Gleichspannung (-15 % / +20 %) bereitstellt.

Verkabeln Sie den Embedded-PC im Schaltschrank entsprechend der Norm EN 60204-1:2006 Schutzkleinspannungen (PELV = Protective Extra Low Voltage):

- Die Leiter "PE" und "0 V" der Spannungsquelle für ein CPU-Grundmodul müssen auf dem gleichen Potential liegen (im Schaltschrank verbunden).
- Die Norm EN 60204-1:2006 Abschnitt 6.4.1.b: schreibt vor, dass eine Seite des Stromkreises oder ein Punkt der Energiequelle dieses Stromkreises an das Schutzleitersystem angeschlossen werden muss.

Anschlüsse

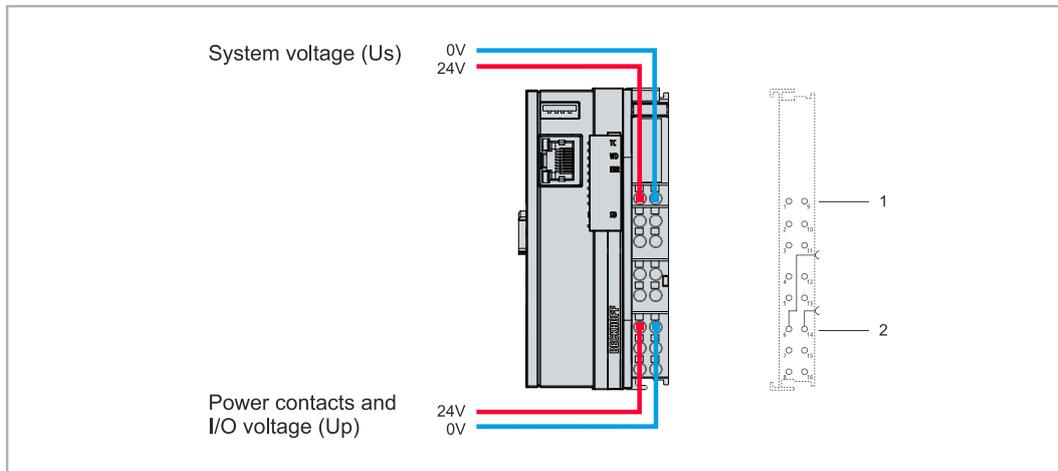


Abb. 9: Anschlüsse für Systemspannung (Us) und Powerkontakte (Up).

Tab. 6: Legende zum Anschlussbeispiel.

Nr.	Beschreibung
1	Die oberen Federkraftklemmen mit der Bezeichnung "+24 V Us" und "0 V Us" versorgen das CPU-Grundmodul und den Klemmenbus (Datenübertragung über K- oder E-Bus) mit Spannung.
2	Die Federkraftklemmen mit der Bezeichnung "+24 V Up" und "0 V Up" versorgen die Multifunktions-I/Os, die Busklemmen und EtherCAT-Klemmen über die Powerkontakte mit Spannung.

Sicherung

- Beachten Sie bei der Dimensionierung der Sicherung für die Systemspannung (Us) die max. Leistungsaufnahme des Embedded-PCs (siehe: [Technische Daten \[► 134\]](#))
- Sichern Sie Powerkontakte (Up) mit einer Sicherung von max. 10 A (träge) ab.

Spannungsversorgung unterbrechen / abschalten

Um den Embedded-PC abzuschalten, darf nicht die Masse (0 V) getrennt werden, da sonst je nach Gerät der Strom über den Schirm weiterfließt und der Embedded-PC oder die Peripherie beschädigt wird.

Trennen Sie immer die 24 V Leitung. An dem Embedded-PC angeschlossene Geräte mit eigener Stromversorgung (z.B. ein Panel) müssen für "PE" und „0 V“ das gleiche Potential wie der Embedded-PC haben (keine Potentialdifferenz).

5.2.1 Embedded-PC anschließen

Die Leitungen einer externen Spannungsquelle werden mit Federkraftklemmen an der Netzteilklemme verbunden. Beachten Sie die erforderlichen Leiterquerschnitte und Abisolierlängen.

Tab. 7: Erforderliche Leiterquerschnitte und Abisolierlängen.

Leiterquerschnitt	e*: 0,08 ... 1,5 mm ² f*: 0,25 ... 1,5 mm ² a*: 0,14 ... 0,75 mm ²	e*: AWG 28 ... 16 f*: AWG 22 ... 16 a*: AWG 26 ... 19
Abisolierlänge	8 ... 9 mm	0.33 inch

*e: eindrätig, Draht massiv; f: feindrätig, Litze; a: mit Aderendhülse

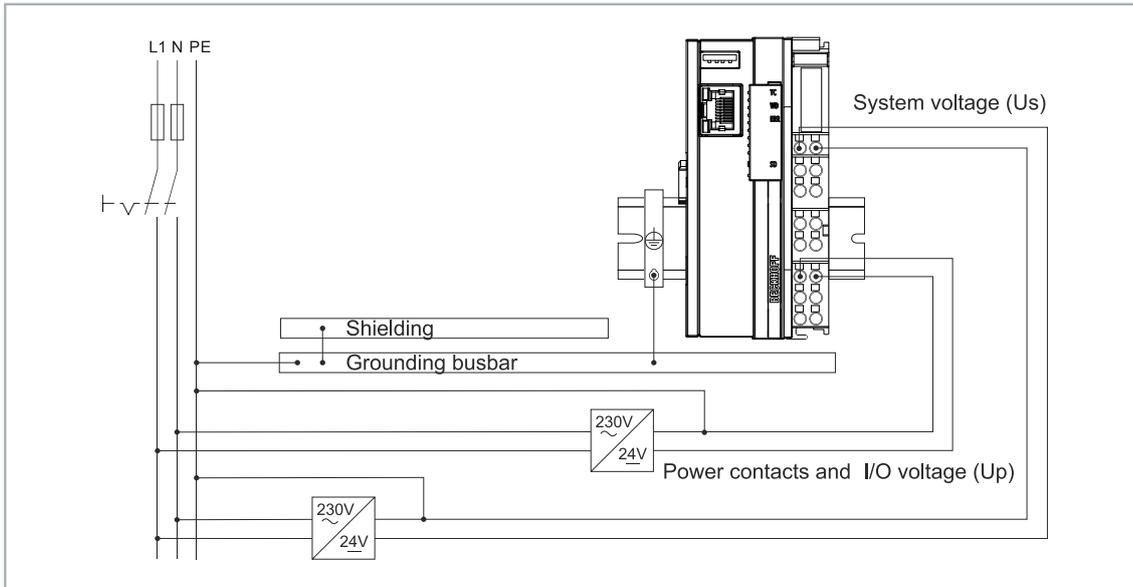
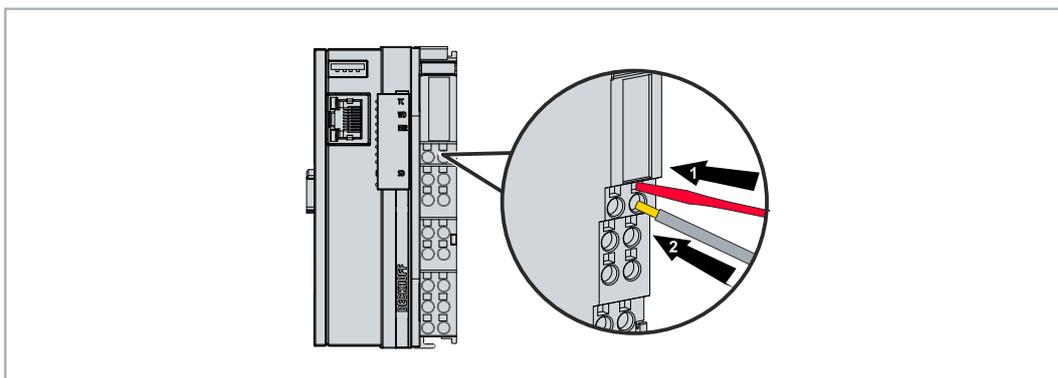


Abb. 10: Anschlussbeispiel mit einem CX7000.

Schließen Sie den Embedded-PC wie folgt an:

1. Öffnen Sie eine Federkraftklemme, indem Sie mit einem Schraubendreher oder einem Dorn leicht in die viereckige Öffnung über der Klemme drücken.



2. Der Draht kann nun ohne Widerstand in die runde Klemmenöffnung eingeführt werden.
 3. Durch Rücknahme des Druckes schließt sich die Klemme automatisch und hält den Draht sicher und dauerhaft fest.
- ⇒ Sie haben die Spannungsquelle erfolgreich an die Netzteilklemme angeschlossen, wenn die beiden oberen LEDs der Netzteilklemme grün aufleuchten.

Die linke LED (Us 24V) zeigt die Versorgung des CPU-Grundmoduls und des Klemmenbusses an. Die rechte LED (Up 24V) zeigt die Versorgung der Busklemmen über die Powerkontakte an.

5.2.2 UL-Anforderungen

Die Embedded-PCs CX7031 sind UL-zertifiziert. Das entsprechende UL-Label befindet sich auf dem Typenschild.



Abb. 11: UL-Label beim CX7031.

Die Embedded-PCs CX7031 können damit in Bereichen eingesetzt werden, in denen spezielle UL-Anforderungen eingehalten werden müssen. Diese Anforderungen gelten für die Systemspannung (U_s) und für die Powerkontakte (Up). Einsatzbereiche ohne spezielle UL-Anforderungen sind von den UL-Vorschriften nicht betroffen.

UL-Anforderungen:

- Die Embedded-PCs dürfen nicht mit unbegrenzten Spannungsquellen verbunden werden.
- Embedded-PCs dürfen nur mit einer Spannungsquelle von 24 V Gleichspannung versorgt werden. Die Spannungsquelle muss isoliert sein und mit einer Sicherung (entsprechend UL248) von maximal 4 A geschützt werden.
- Oder die Spannungsversorgung muss von einer Spannungsquelle stammen, die NEC class 2 entspricht. Eine Spannungsquelle entsprechend NEC class 2 darf dabei nicht seriell oder parallel mit einer anderen NEC class 2 Spannungsquelle verbunden werden.

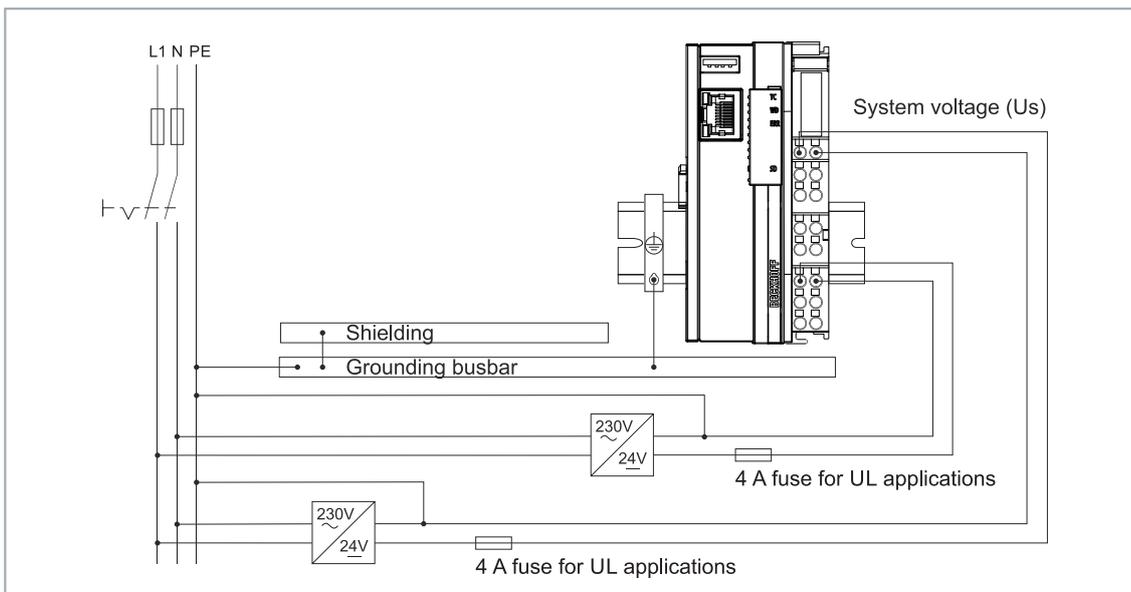


Abb. 12: Anschlussbeispiel für Bereiche mit speziellen UL-Anforderungen.

5.3 PROFIBUS: Anschluss und Verdrahtung

5.3.1 D-Sub-Buchse (X003)

Pin 6 überträgt 5 V_{DC} 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.

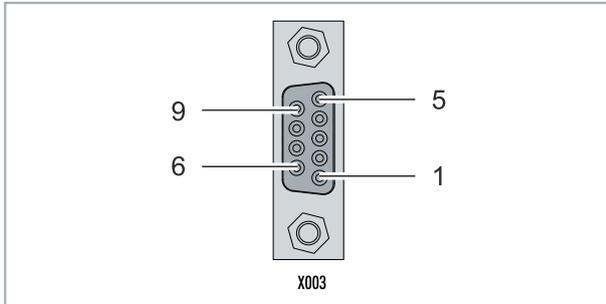


Abb. 13: PROFIBUS-Schnittstelle X003.

Die PROFIBUS-Busleitung wird über einen 9-polige D-Sub-Buchse mit folgender Belegung angeschlossen:

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

Leitungsfarben

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

5.3.2 Anschlussstechnik

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, Stichleitungen 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

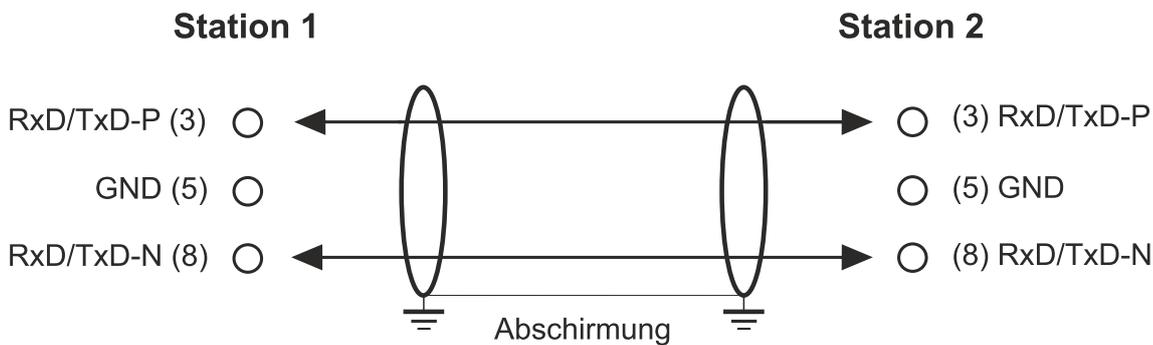
Beachten Sie die besonderen Anforderungen an das Datenkabel bei Baud-Raten von mehr als 1,5 Mbaud. Das richtige Kabel ist Grundvoraussetzung für den störungsfreien Betrieb des Bussystems. Bei der Verwendung des normalen 1,5 Mbaud-Kabels kann es durch Reflexionen und zu großer Dämpfung zu erstaunlichen 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 Fehlverhalten.

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.

i Vorkonfektionierte Kabel von BECKHOFF

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 feldkonfektionierte Stecker und Kabel erhältlich.

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



● Abschlusswiderstände

i 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 Ringstruktur 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: <https://www.profibus.com>).

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 (<https://www.profibus.com>).

5.3.3 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.

5.3.4 Kabel und Steckverbinder für PROFIBUS

Eine produktübergreifende und ausführliche Liste für weiteres PROFIBUS-Zubehör wie Y-Verteiler, T-Verteiler, Kupplungen und Abschlusswiderstände finden Sie auf der [Beckhoff-Homepage](#).

Tab. 8: ZK1031-6xxx-1xxx | PROFIBUS-Leitungen, PUR, schleppkettentauglich

Bestellangaben	Beschreibung
ZK1031-6100-1xxx	M12, Stecker, gerade, Stift, 4-polig, B-kodiert – offenes Ende
ZK1031-6300-1xxx	M12, Stecker, gewinkelt, Stift, 4-polig, B-kodiert – offenes Ende
ZK1031-6200-1xxx	M12, Flansch, gerade, Buchse, 4-polig, B-kodiert – offenes Ende
ZK1031-6400-1xxx	M12, Flansch, gewinkelt, Buchse, 4-polig, B-kodiert – offenes Ende

Tab. 9: ZB3xxx | PROFIBUS-Leitungen, Meterware

Bestellangaben	Beschreibung
ZB3200	PROFIBUS-Leitung, 12 Mbaud, PVC, 1 x 2 x 0,34 mm ² , feste Verlegung
ZB3300	PROFIBUS-Leitung, 12 Mbaud, 2 x 0,25 mm ² + 3 x 0,75 mm ² , 5-adrig, schleppkettentauglich

Tab. 10: ZS1xxx | PROFIBUS-Steckverbinder

Bestellangaben	Beschreibung
ZB3100	9-poliger D-Sub-Stecker für PROFIBUS (12 Mbaud) mit zuschaltbarem Abschlusswiderstand
ZS1031-3000	9-poliger D-Sub-Stecker für PROFIBUS (12 Mbaud) mit integriertem Abschlusswiderstand (andere Bauform als ZB3100)

6 Multifunktions-I/Os

Für die Konfiguration der Betriebsarten stehen insgesamt vier einstellbare Slots zur Verfügung. Ein Slot ist eine bestimmte Anzahl von Ein- und Ausgängen. Für jeden Slot kann maximal ein Modul (DI, DIO, ENC, CNT oder PWM) zugewiesen werden, welches wiederum die Betriebsart für den jeweiligen Slot festlegt. Ein Modul ist also eine Funktion, die diese Ein- und Ausgänge übernehmen können. Die aktuelle Modulkonfiguration wird in TwinCAT unter der CX7028-Schnittstelle aufgelistet. Beachten Sie, dass die CX7028-Schnittstelle für die Steuerung der Multifunktions-I/Os eine eigene CPU hat und die CX7028-Schnittstelle unter TwinCAT nicht angezeigt wird bzw. nicht funktioniert, wenn die Spannungsversorgung (Up) nicht angeschlossen ist.

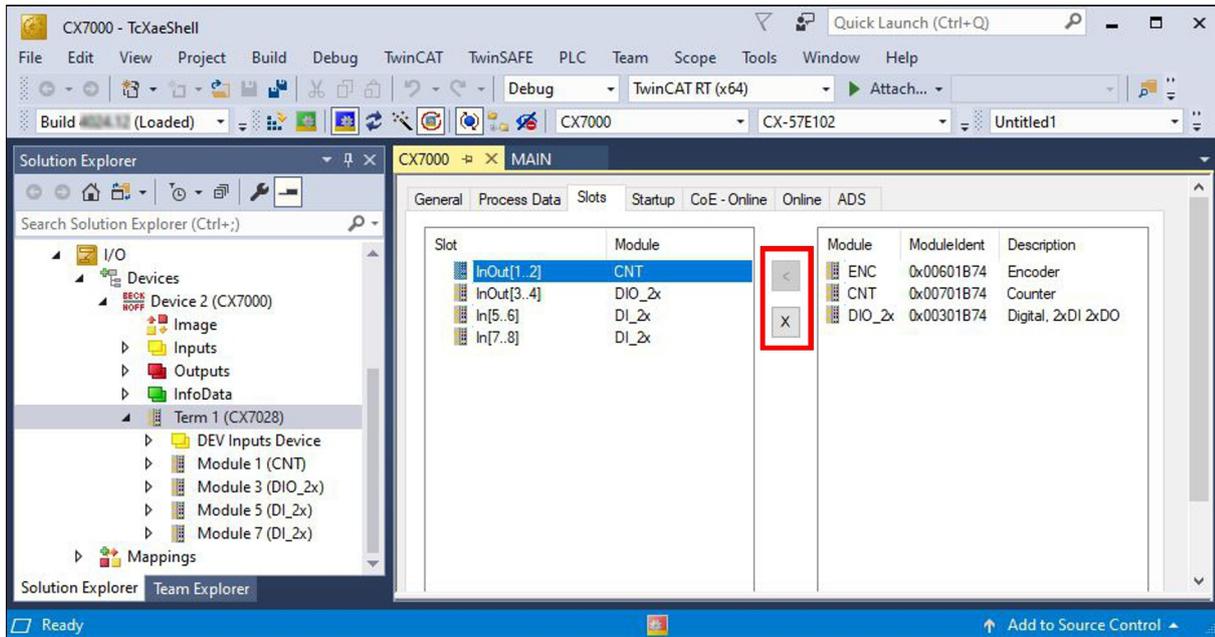


Abb. 14: CX7028-Schnittstelle, Slot- und Modul-Konfiguration unter TwinCAT.

Mit der Schaltfläche < können Module einem bestimmten Slot zugewiesen oder mit x wieder entfernt werden. Abhängig vom verwendeten Slot stehen unterschiedliche Module zur Auswahl. Welche Module von welchem Slot unterstützt werden, wird im Folgenden aufgelistet.

Zykluszeit für Multifunktions-I/Os

Die Kommunikation zu den Multifunktions-I/Os wird mit einem festen Watchdog von 100 ms überwacht. Das bedeutet, dass die Zykluszeit für die Multifunktions-I/O schneller als 100 ms sein muss.

Slot 1:

Bei der Verwendung von Slot 1 werden die Eingänge 1, 2 und (*3) sowie die Ausgänge 1 und 2 konfiguriert.

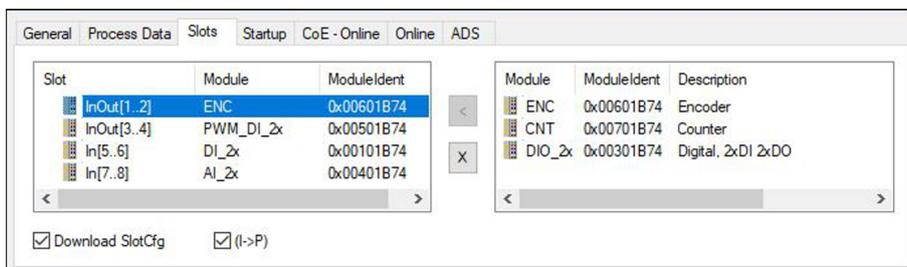


Abb. 15: Unterstützte Module bei der Verwendung von Slot 1.

- ENC (Inkremental-Encoder-Modus). 2 x Digitaleingang für 250-kHz-Encoder-Signal, 2 x Encoder-Digitalausgang.
- CNT (Zähler-Modus). 1 x Zähler-Digitaler Eingang 100 kHz, 1 x Digitaler Eingang als Auf-/Abwärts-Zähler 20 kHz, 2 x Zähler-Digitalausgang.

- DIO_2x (Digitale Ein- und Ausgänge). 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3, 2 x Digitalausgang, 24 V DC, 0,5 A, 1-Leitertechnik.

*) Der Eingang 3 ist nur im Inkremental-Encoder-Modus verfügbar. Bei einem High-Pegel kann der Wert des Inkremental-Encoders gelatcht oder der Zähler zurückgesetzt werden.

Slot 2:

Bei der Verwendung von Slot 2 werden die Eingänge 3 und 4, sowie die Ausgänge 3 und 4 konfiguriert.

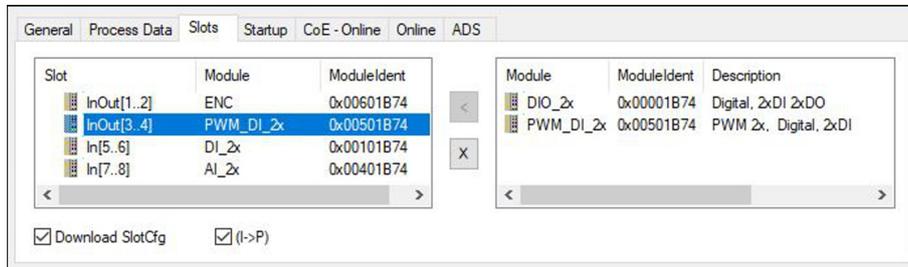


Abb. 16: Unterstützte Module bei der Verwendung von Slot 2.

- DIO_2x (Digitale Ein- und Ausgänge). 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3, 2 x Digitalausgang, 24 V DC, 0,5 A, 1-Leitertechnik.
- PWM_DI_2x (PWM-Signal-Modus). 2 x Digitaleingang, 24 V DC, Filter 3 ms, 2 x Digitalausgang konfiguriert für PWM-Signal.

Slot 3:

Bei der Verwendung von Slot 3 werden die Eingänge 5 und 6 konfiguriert.

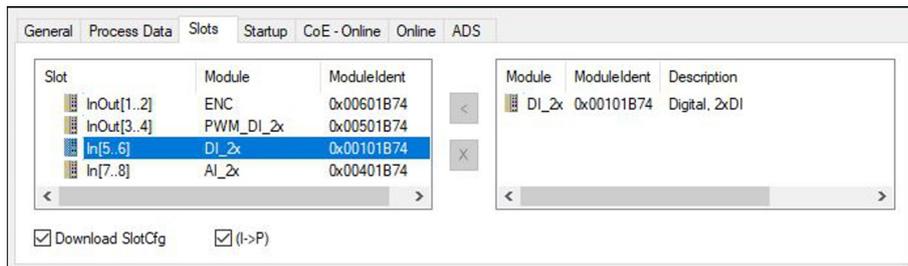


Abb. 17: Unterstützte Module bei der Verwendung von Slot 3.

Slot 3 beinhaltet nur ein Modul und kann daher nicht anders konfiguriert werden. Das Modul unterstützt 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3.

Slot 4:

Bei der Verwendung von Slot 4 werden die Eingänge 7 und 8 konfiguriert.

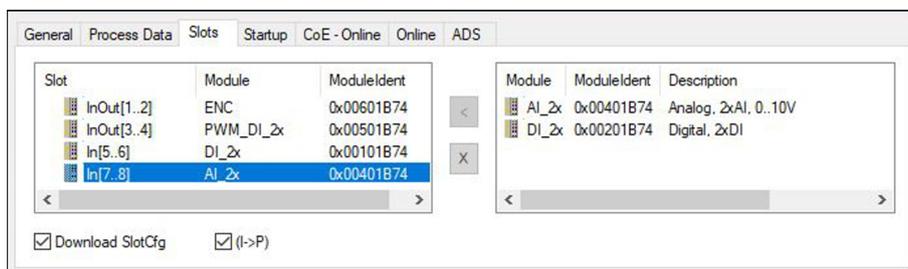


Abb. 18: Unterstützte Module bei der Verwendung von Slot 4.

- AI_2x (Analog-Signal-Modus). 2 x Digitaleingang konfiguriert als Analogeingang 0 bis 10 V, 12 Bit
- DI_2x (Digitaler Eingang). 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3

6.1 Digitale Eingänge

Die digitalen Eingänge erfassen binäre Steuersignale aus der Prozessebene. Typischerweise sind dies mechanische Kontakte wie Öffner- oder Schließer-Kontakte, elektronische Sensoren, wie induktive Näherungsschalter, optische Sensoren oder andere Methoden, um ein Low-/High-Signal im Sinne der Steuerungstechnik zu erzeugen. Der CX70xx verfügt dank integrierter Multifunktions-I/Os über insgesamt 8 digitale Eingänge, 24 V DC, Filter 3 ms, Typ 3.

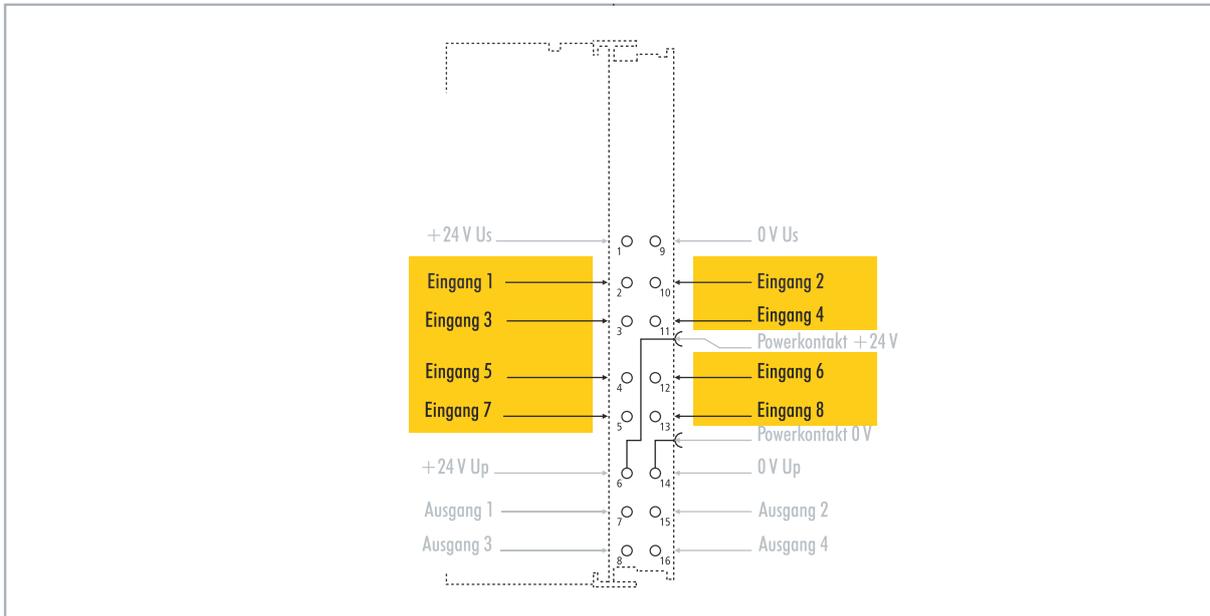


Abb. 19: Konfigurierbare digitale Eingänge.

Die digitalen Eingänge verfügen über einen 3 ms-Eingangsfiler. Der Signalzustand der einzelnen Eingänge wird durch jeweils eine Leuchtdiode angezeigt. Für die digitalen Eingänge 3, 4, 5 und 6 können zusätzliche Filtereinstellungen in den passenden CoE-Objekten vorgenommen und beispielsweise die Auflösung und Filterzeit eingestellt werden.

Tab. 11: Technische Daten, Multifunktions-I/Os als digitale Eingänge.

Technische Daten	CX7031
Anschlusstechnik	1-Leiter
Anzahl Eingänge	8
Nennspannung	24 V DC (-15 %/+20 %)
Spezifikation	EN 61131-2, Typ 3
Signalspannung „0“	-3...+5 V
Signalspannung „1“	11...30 V
Eingangsfiler	Konfigurierbar, Standard: 3 ms, min.: 10 µs
Anschlussquerschnitt	e*: 0,08...1,5 mm ² , f*: 0,25...1,5 mm ² , a*: 0,14...0,75 mm ²
Anschlussquerschnitt AWG	e*: AWG 28...16, f*: AWG 22...16, a*: AWG 26...19
Abisolierlänge	8...9 mm

*e: eindrätig, Draht massiv; f: feindrätig, Litze; a: mit Aderendhülse

6.2 Digitale Ausgänge

HINWEIS

Rückspeisung bei den 24-V-Ausgängen

Eine Spannung von 24 V an den Ausgängen kann das Gerät zerstören, wenn die Spannungsversorgung (Up) nicht angeschlossen ist (Rückspeisung). Schließen Sie die Spannungsversorgung (Up) an, damit 24 V an den Ausgängen angelegt werden dürfen.

Die digitalen Ausgänge schalten binäre 24-V-DC-Steuersignale galvanisch getrennt zur Prozessebene an Aktoren weiter. Dabei entspricht der High-Pegel bei der positiv schaltenden Logik der Versorgungsspannung.

Die Ausgänge 3 und 4 verfügen über eine PWM-Endstufe. Wenn die beiden digitalen Ausgänge als normale digitale Ausgänge verwendet werden, kommt es durch die interne Beschaltung zu einem Leckstrom von kleiner 100 µA, der eine Spannung von ca. 5 V verursacht. Will man beim Low-Pegel des Ausgangs annähernd 0 V erreichen, muss ein 47 kΩ Widerstand gegen Masse geschaltet werden.

Eine weitere Möglichkeit besteht darin, die beiden Ausgänge im PWM-Modus zu betreiben und die Variable PWM output des PWM-Signals für FALSE mit 0x0000 und für TRUE mit 0xFFFF zu beschreiben. Damit ist die PWM-Endstufe aktiv, die keinen Leckstrom erzeugt.

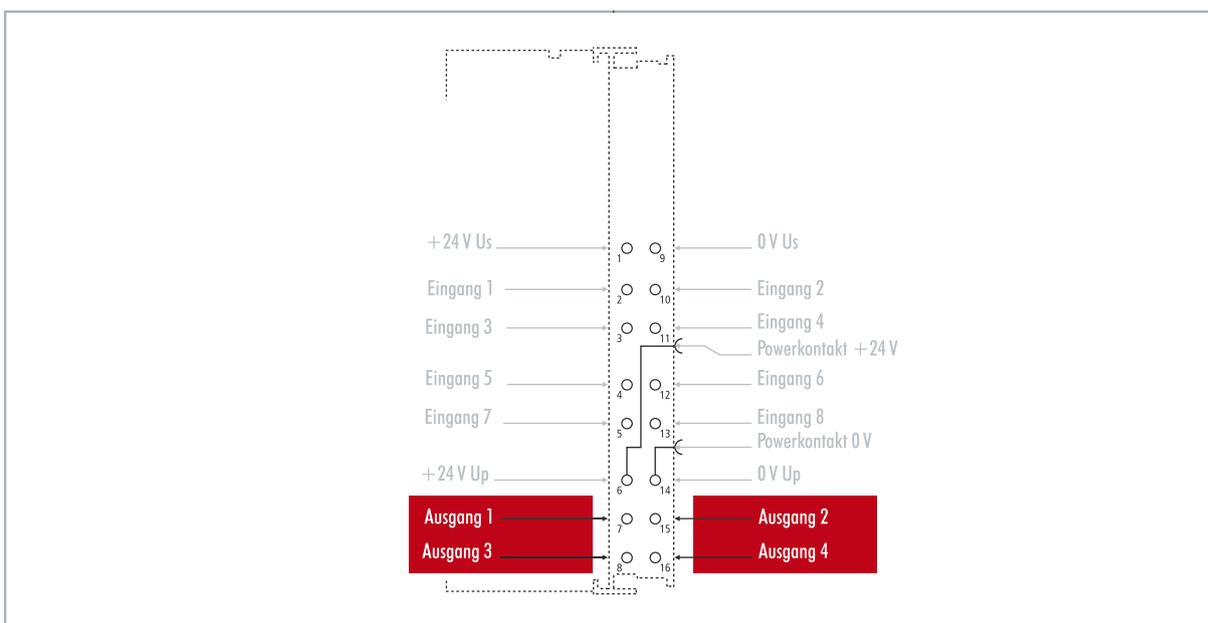


Abb. 20: Konfigurierbare digitale Ausgänge.

Der CX7031 enthält insgesamt vier Ausgänge, die ihren Signalzustand durch Leuchtdioden anzeigen. Mit den Ausgängen lassen sich Standardaktoren wie beispielsweise Schütze und Ventile schalten.

Tab. 12: Technische Daten, Multifunktions-I/Os als digitale Ausgänge.

Technische Daten	CX7031
Anschlusstechnik	1-Leiter
Anzahl Ausgänge	4
Nennspannung	24 V DC (-15 %/+20 %)
Lastart	ohmsch, induktiv, Lampenlast
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
Schaltzeiten	T _{ON} : 20 µs typ., T _{OFF} : 10 µs typ.
Kurzschlussstrom	< 2 A typ.
Abschaltenergie (ind.) max.	< 150 mJ/Kanal

Technische Daten	CX7031
Anschlussquerschnitt	e*: 0,08...1,5 mm ² , f*: 0,25...1,5 mm ² , a*: 0,14...0,75 mm ²
Anschlussquerschnitt AWG	e*: AWG 28...16, f*: AWG 22...16, a*: AWG 26...19
Abisolierlänge	8...9 mm

*e: eindrätig, Draht massiv; f: feindrätig, Litze; a: mit Aderendhülse

6.3 Zähler-Modus

Der Embedded-PC CX7031 kann als ein Vor-/Rückwärtszähler konfiguriert werden, der das Zählen eines Pulses ermöglicht. Der Embedded-PC ist für schnelle Zählaufgaben mit einer Grenzfrequenz bis 100 kHz geeignet, wobei der CX7031 im 1-Zähler-Modus betrieben werden kann.

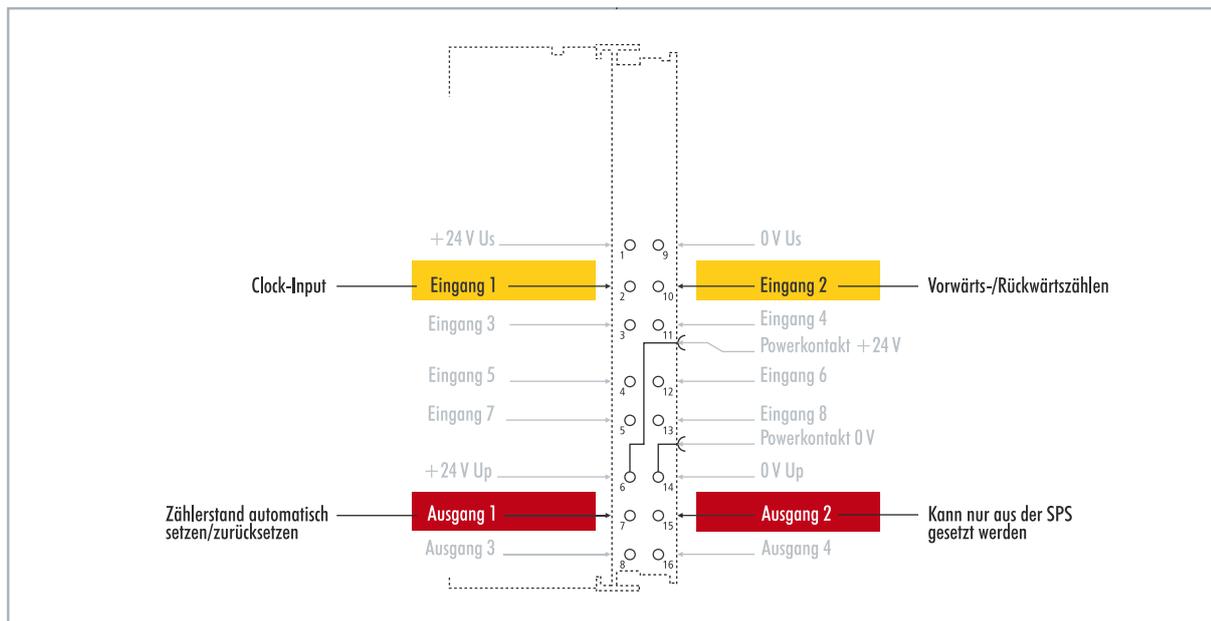


Abb. 21: Konfigurierbare Ein- und Ausgänge im Zähler-Modus.

Der CX7031 unterstützt im Zähler-Modus drei Betriebsarten:

- Vor-/Rückwärtszähler
- Vorwärtszähler
- Rückwärtszähler

Zusätzlich lässt sich Ausgang 1 abhängig vom Zählerstand schalten. Ausgang 2 kann aus der SPS geschaltet werden. Damit können schnelle Steuersignale für Feldgeräte genutzt und geschaltet werden.

Die Betriebsarten werden in TwinCAT über CoE-Objekte eingestellt.

Vor-/Rückwärtszähler

Bei der Betriebsart Vor-/Rückwärtszähler wird der Puls, der gezählt werden soll, über den digitalen Eingang 1 detektiert. Über den digitalen Eingang 2 wird die Zählrichtung vorgegeben.

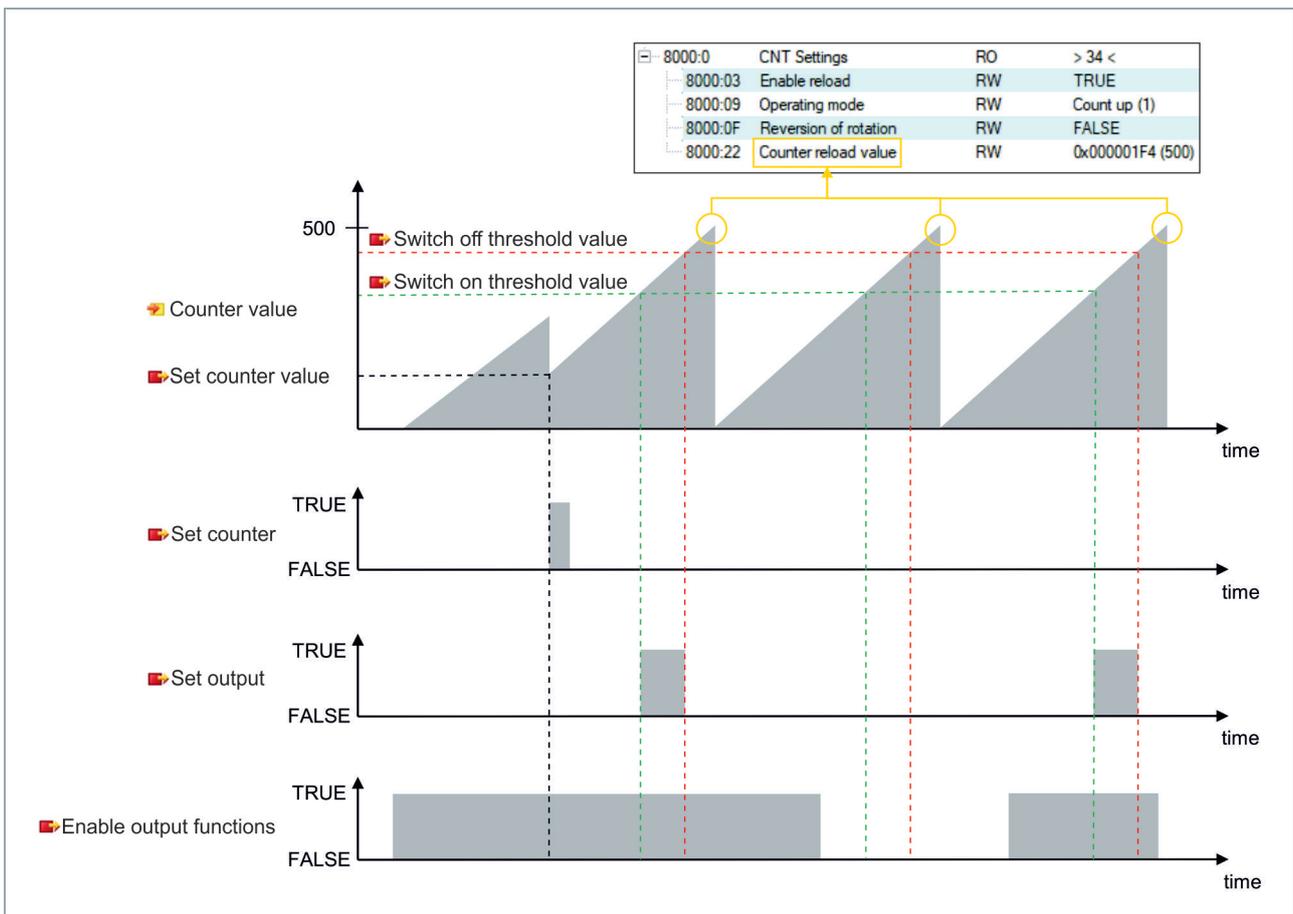
Liegt am Eingang 1 und gleichzeitig am Eingang 2 ein High-Pegel an, wird vorwärts gezählt. Liegt am Eingang 1 ein High-Pegel und am Eingang 2 ein Low-Pegel an, wird rückwärts gezählt.

Vorwärtszähler

Bei dieser Betriebsart wird das Signal am digitalen Eingang 1 detektiert.

Rückwärtszähler

Bei dieser Betriebsart wird das Signal am digitalen Eingang 1 detektiert.



Tab. 13: Technische Daten, Multifunktions-I/Os im Zähler-Modus.

Technische Daten	CX7031
Anzahl der Zähler	1 x Vor-/Rückwärtszähler, 1 x Vor- oder Rückwärtszähler
Nennspannung	24 V DC (-15 %/+20 %)
Spezifikation	EN 61131-2, Typ 3
Signalspannung „0“	-3...+5 V
Signalspannung „1“	11...30 V
Grenzfrequenz	Vor-/Rückwärtszähler: 20 kHz ¹⁾ , Zählen nur in eine Richtung: 100 kHz
Zählertiefe	32 Bit
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
Besondere Eigenschaften	Zähler setzen, Ausgänge schalten, Zähler zurücksetzen
Anschlussquerschnitt	e*: 0,08...1,5 mm ² , f*: 0,25...1,5 mm ² , a*: 0,14...0,75 mm ²
Anschlussquerschnitt AWG	e*: AWG 28...16, f*: AWG 22...16, a*: AWG 26...19
Abisolierlänge	8...9 mm

¹⁾ Der Vor-/Rückwärtszähler kann auch bis 100 kHz zählen, nur bei einer Richtungsumkehr muss die Zählfrequenz ≤ 20 kHz sein, sonst gehen Impulse verloren.

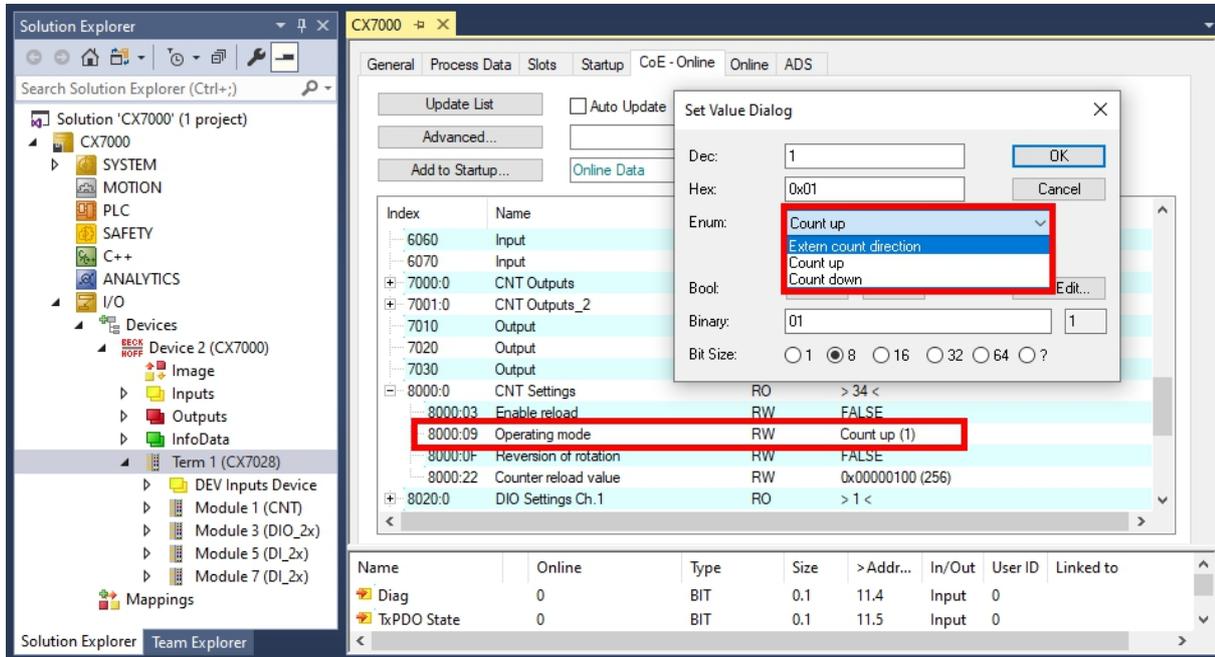
*e: eindrätig, Draht massiv; f: feindrätig, Litze; a: mit Aderendhülse

6.3.1 Betriebsart wählen

Der CX7031 unterstützt im Zähler-Modus drei Betriebsarten. Die Betriebsart wird in TwinCAT über CoE-Objekte eingestellt. Sie können zwischen den drei Betriebsarten Vor-/Rückwärtszähler, Vorwärtszähler und Rückwärtszähler wählen.

Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8000:09 Operating mode**.
4. Wählen Sie unter der Option **Enum** die erforderliche Betriebsart.
 - ⇒ Die Betriebsart wird übernommen. Beachten Sie, dass Sie beim CX7031 immer nur eine Betriebsart gleichzeitig verwenden können und eine Mischung aus den Betriebsarten nicht möglich ist.

6.3.2 Ausgänge schalten

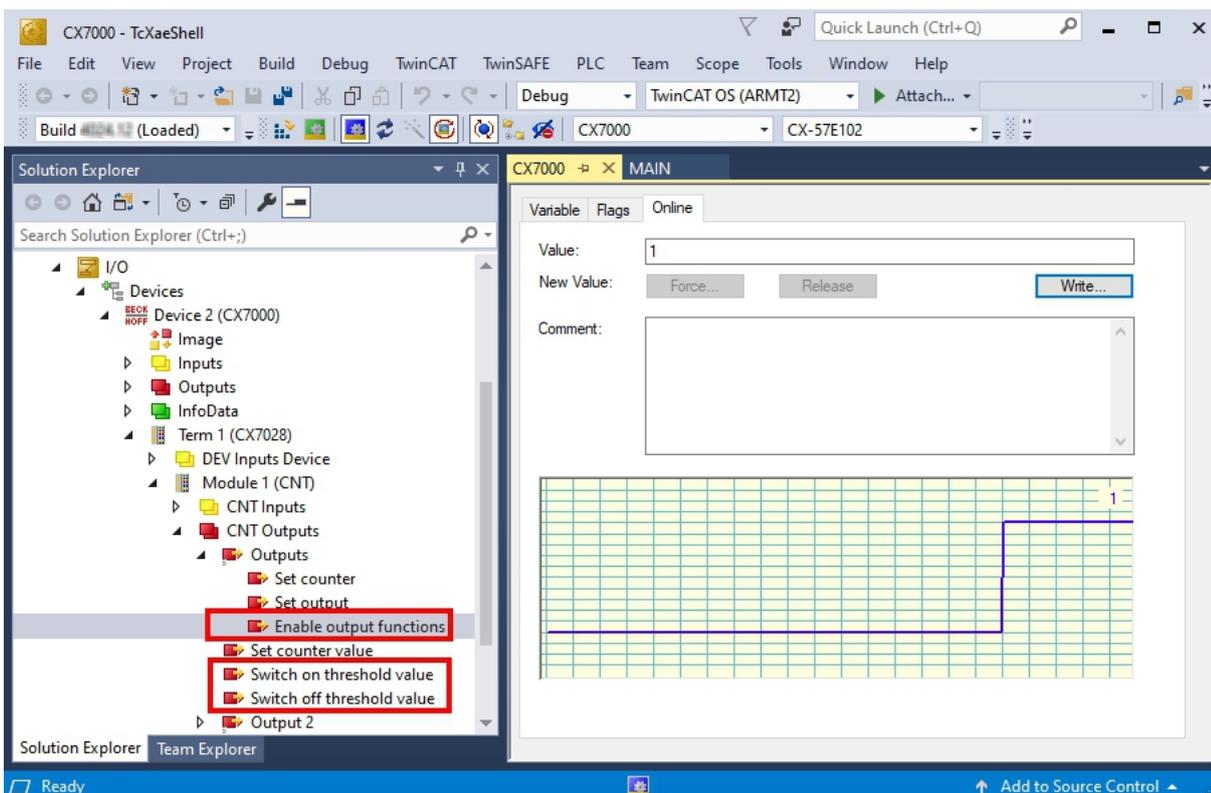
Beim CX7031 ist es möglich den Ausgang 1 selbsttätig zu schalten, sobald ein bestimmter Zählerstand erreicht wird. Das ermöglicht eine schnelle und ohne die SPS laufende Bearbeitung. Ein zweiter Ausgang, der Ausgang 2, kann unabhängig vom Zählerstand über die SPS geschaltet werden.

Über die Variablen **Switch on threshold value** und **Switch off threshold value** wird der Ausgang 1 geschaltet bzw. abgeschaltet:

- Wird der eingestellte Wert unter **Switch on threshold value** erreicht, wird der Ausgang geschaltet.
- Wird der eingestellte Wert unter **Switch off threshold value** erreicht, wird der Ausgang ausgeschaltet.

Beim Rückwärtszählen wird die entsprechende Schaltanweisung umgekehrt ausgeführt. Wird der eingestellte Wert unter **Switch on threshold value** unterschritten, wird der Ausgang 1 abgeschaltet.

Gehen Sie wie folgt vor:



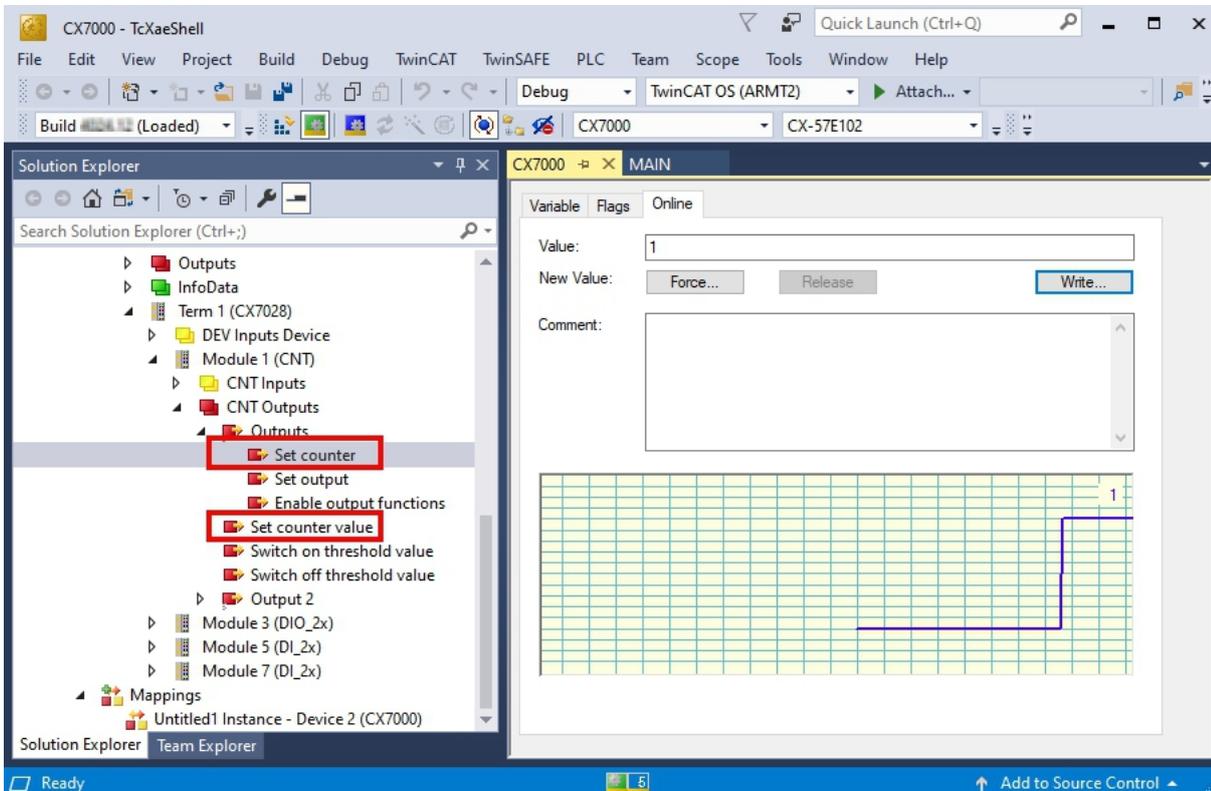
1. Geben Sie über die Variable **Switch on threshold value** einen Zählerstand vor, bei dem der Ausgang geschaltet werden soll.
 2. Geben Sie über die Variable **Switch off threshold value** einen Zählerstand vor, bei dem der Ausgang abgeschaltet werden soll.
 3. Setzen Sie anschließend die Variable **Enable output functions** auf **True**, damit die Einstellungen übernommen werden.
- ⇒ Erst wenn die Variable **Enable output functions** auf **True** gesetzt wird, wird die Funktion eingeschaltet und der Ausgang geschaltet.

Ist der parametrisierte Zählerstand aus **Switch on/off threshold** erreicht bzw. überschritten, aber die Variable **Enable output functions** nicht gesetzt, wird der Schaltbefehl nicht ausgeführt. Sobald dann **Enable output functions** gesetzt wird, wird umgehend der Ausgang geschaltet. Genauso wirkt sich ein nachträglich aktivierter Zählerstand **Switch on/off threshold** bei erfüllter Schaltbedingung umgehend auf den Ausgang aus.

6.3.3 Zählerstand setzen

In diesem Schritt wird gezeigt, wie Sie den Zählerstand auf einen bestimmten Wert setzen können. Über die Variable **Set counter value** wird ein Wert vorgegeben und über die Variable **Set counter** wird der Zählerstand gesetzt. Beide Variablen können aus der SPS gesteuert werden.

Gehen Sie wie folgt vor:



1. Geben Sie über die Variable **Set counter value** einen Wert vor, der als Zählerstand gesetzt werden soll.
 2. Setzen Sie anschließend die Variable **Set counter** auf **True**, damit die Einstellungen übernommen werden.
- ⇒ Erst wenn die Variable **Set counter** auf **True** gesetzt wird, wird der eingestellte Wert unter **Set counter value** für den Zählerstand übernommen.

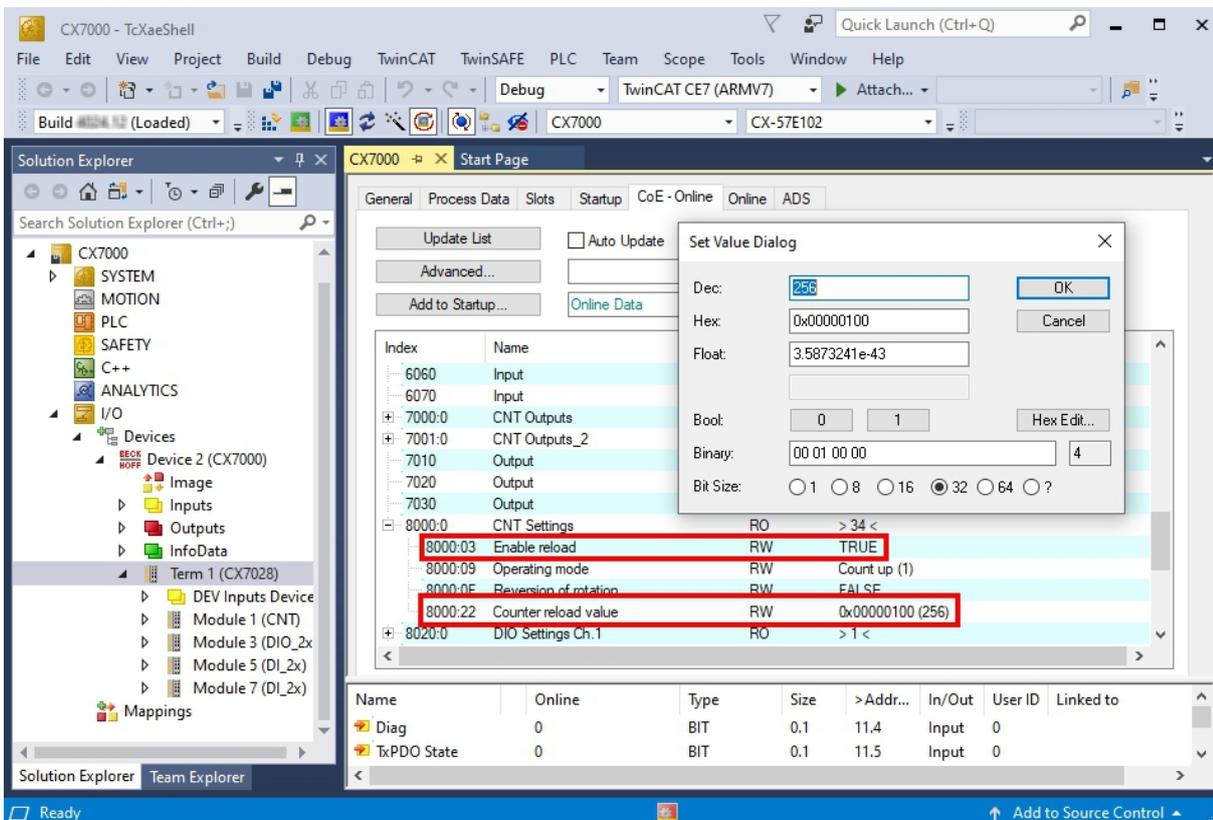
6.3.4 Grenzwert für Zähler festlegen

In diesem Schritt wird gezeigt, wie Sie in TwinCAT einen Grenzwert festlegen, ab dem der Zählerstand automatisch wieder auf null zurückgesetzt wird. Beim Vorwärtszählen wird der Zählerstand beim Erreichen des Grenzwertes auf null zurückgesetzt. Beim Rückwärtszählen wird der Zählerstand beim Erreichen von Null auf den eingestellten Grenzwert zurückgesetzt.

Der Zählerstand ist eine UDINT-Variable. Der Zähler zählt nur im positiven Bereich von 0-0xFFFF_FFFF (4294967295). Beim Unterschreiten von Null, wird der Zähler auf den maximalen positiven Wert gesetzt. Beim Überschreiten von 4294967295 wird der Zähler auf null gesetzt. Die beiden Variablen **Counter underflow** bzw. **Counter overflow**, zeigen den Überlauf an und werden entweder beim Erreichen von 0x4000 in positiver Richtung oder 0xFFFFC000 in negativer Richtung zurückgesetzt oder wenn der entsprechend andere Überlauf erreicht wurde.

Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8000:22 Counter reload value** und legen Sie den Grenzwert fest.
 4. Klicken Sie anschließend doppelt auf das CoE-Objekt **8000:03 Enable reload** und setzen Sie den Wert auf **True**.
- ⇒ Erst wenn d das CoE-Objekt **8000:03 Enable reload** auf **True** gesetzt ist, ist die Funktion und der definierte Grenzwert aktiv.

6.4 Inkremental-Encoder-Modus

Im Inkremental-Encoder-Modus kann der CX7031 als ein Interface zum direkten Anschluss von 24-V-Inkremental-Encodern konfiguriert werden. Dabei wird eine Vierfachausswertung verwendet und sowohl High-Pegel als auch Low-Pegel an Eingang 1 und Eingang 2 detektiert.

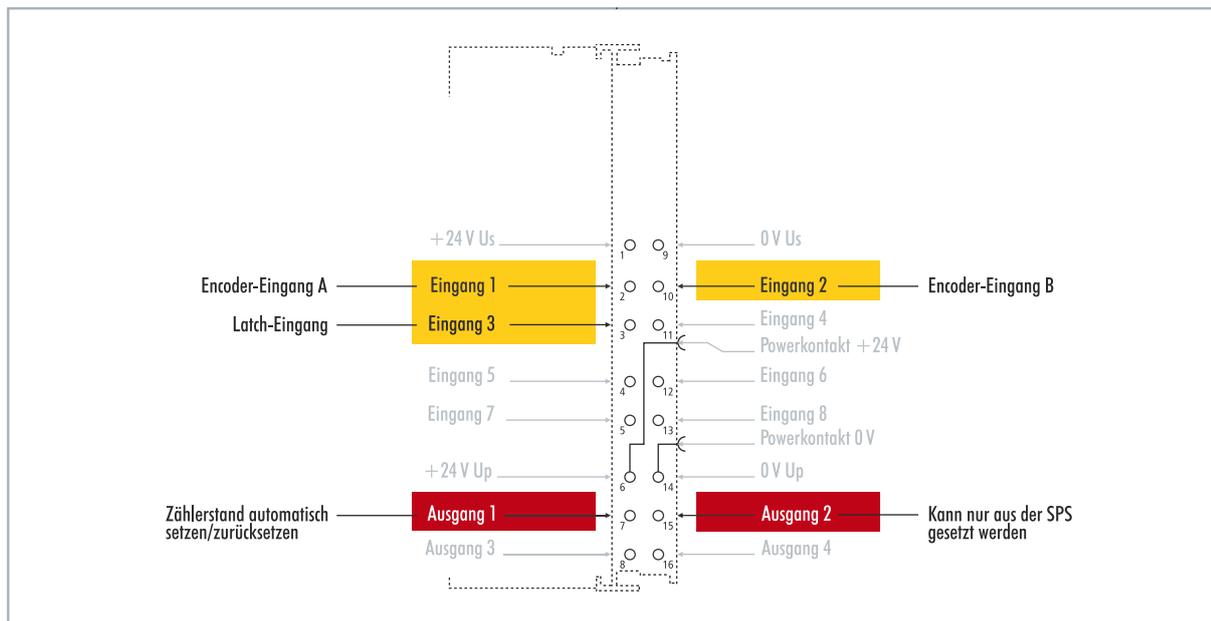
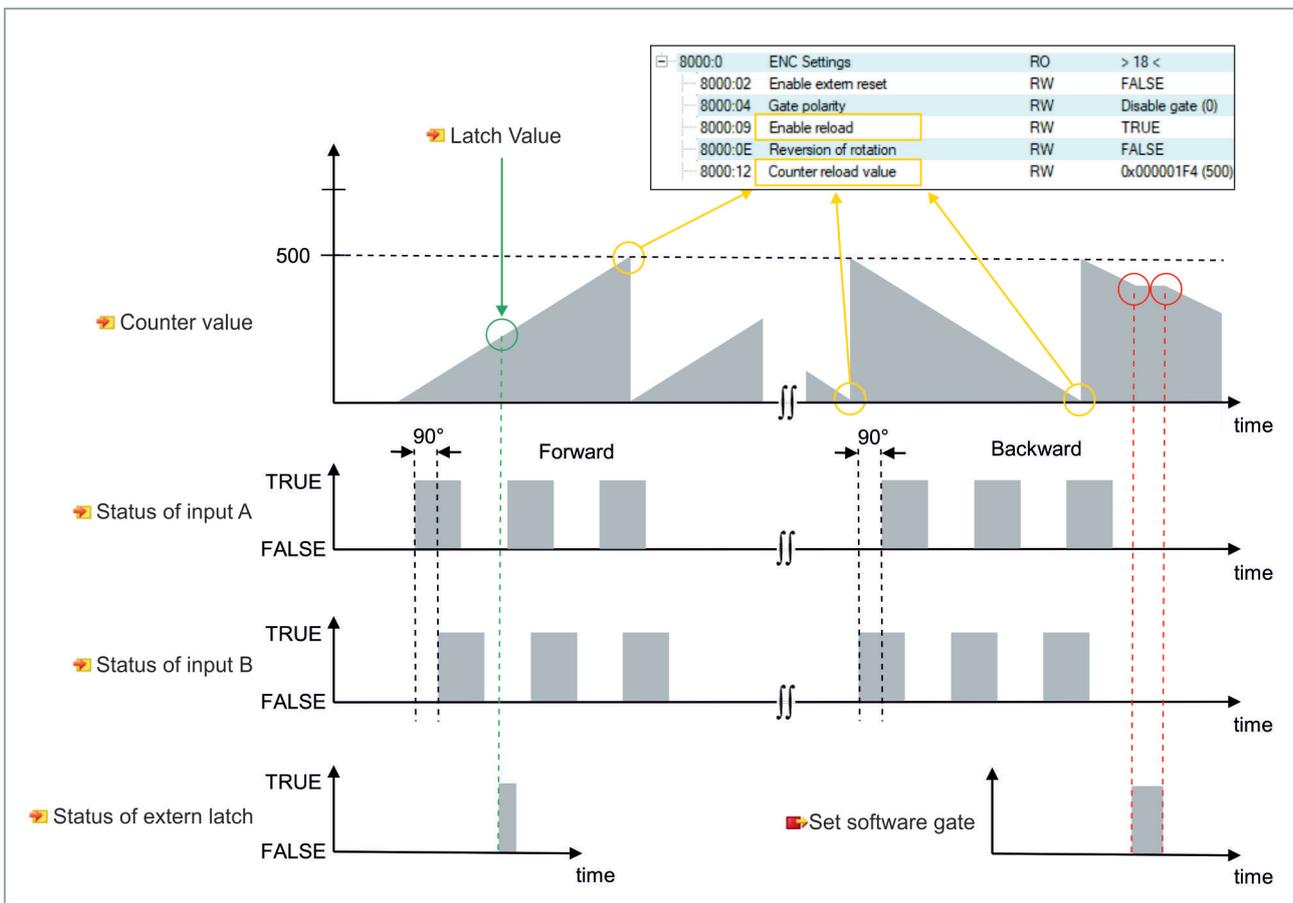


Abb. 22: Konfigurierbare Ein- und Ausgänge im Inkremental-Encoder-Modus.

Der Funktionsumfang im Encoders-Modus entspricht dem Funktionsumfang im Zähler-Modus. Wobei zusätzlich der Zählerstand an Eingang 3 gelatcht werden kann, d. h., dass der Wert bei einem High-Pegel an Eingang 3 in die Prozessdaten eingetragen wird. Alternativ kann der Zähler bei einem High-Pegel an Eingang 3 zurückgesetzt werden.

Zusätzlich lässt sich Ausgang 1 abhängig vom Zählerstand schalten. Ausgang 2 kann aus der SPS geschaltet werden. Damit können schnelle Steuersignale für Feldgeräte genutzt und geschaltet werden.



Tab. 14: Technische Daten, Multifunktions-I/Os im Encoder-Modus.

Technische Daten	CX7031
Technik	Inkremental-Encoder-Interface
Nennspannung	24 V DC (-15 %/+20 %)
Spezifikation	EN 61131-2, Typ 3
Geberanschluss	1 x A, B: 24 V, single-ended
Zusätzliche Eingänge	Latch-Eingang, 24 V DC
Grenzfrequenz	250.000 Inkremente/s (bei 4-fach-Auswertung), entspr. 62,5 kHz
Zählertiefe	32 Bit
Quadraturdecoder	4-fach-Auswertung
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
Besondere Eigenschaften	Latch-Funktion, Software-Gate, Zähler setzen, Ausgänge schalten, Zähler zurücksetzen
Anschlussquerschnitt	e*: 0,08...1,5 mm ² , f*: 0,25...1,5 mm ² , a*: 0,14...0,75 mm ²
Anschlussquerschnitt AWG	e*: AWG 28...16, f*: AWG 22...16, a*: AWG 26...19
Abisolierlänge	8...9 mm

*e: eindrätig, Draht massiv; f: feindrätig, Litze; a: mit Aderendhülse

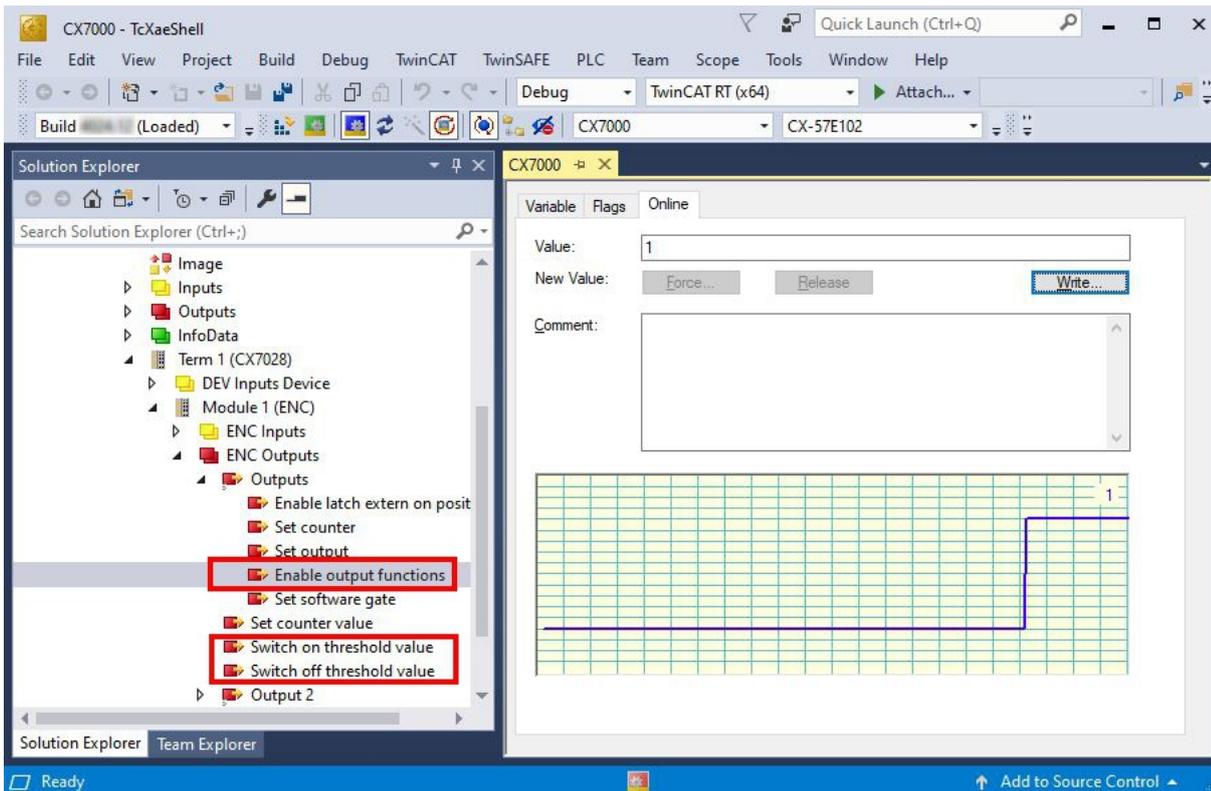
6.4.1 Ausgänge schalten

Beim CX7031 ist es möglich, den Ausgang 1 selbsttätig zu schalten, sobald ein bestimmter Zählerstand erreicht wird. Das ermöglicht eine schnelle und ohne die SPS laufende Bearbeitung. Ein zweiter Ausgang, der Ausgang 2, kann unabhängig vom Zählerstand über die SPS geschaltet werden.

Über die Variablen **Switch on threshold value** und **Switch off threshold value** wird der Ausgang 1 geschaltet bzw. abgeschaltet:

- Wird der eingestellte Wert unter **Switch on threshold value** erreicht, wird der Ausgang geschaltet.
- Wird der eingestellte Wert unter **Switch off threshold value** erreicht, wird der Ausgang ausgeschaltet.

Gehen Sie wie folgt vor:



1. Geben Sie über die Variable **Switch on threshold value** einen Zählerstand vor, bei dem der Ausgang geschaltet werden soll.
 2. Geben Sie über die Variable **Switch off threshold value** einen Zählerstand vor, bei dem der Ausgang abgeschaltet werden soll.
 3. Setzen Sie anschließend die Variable **Enable output functions**, damit die Einstellungen übernommen werden.
- ⇒ Erst wenn die Variable **Enable output functions** auf **True** gesetzt wird, wird die Funktion eingeschaltet und die Einstellungen übernommen.

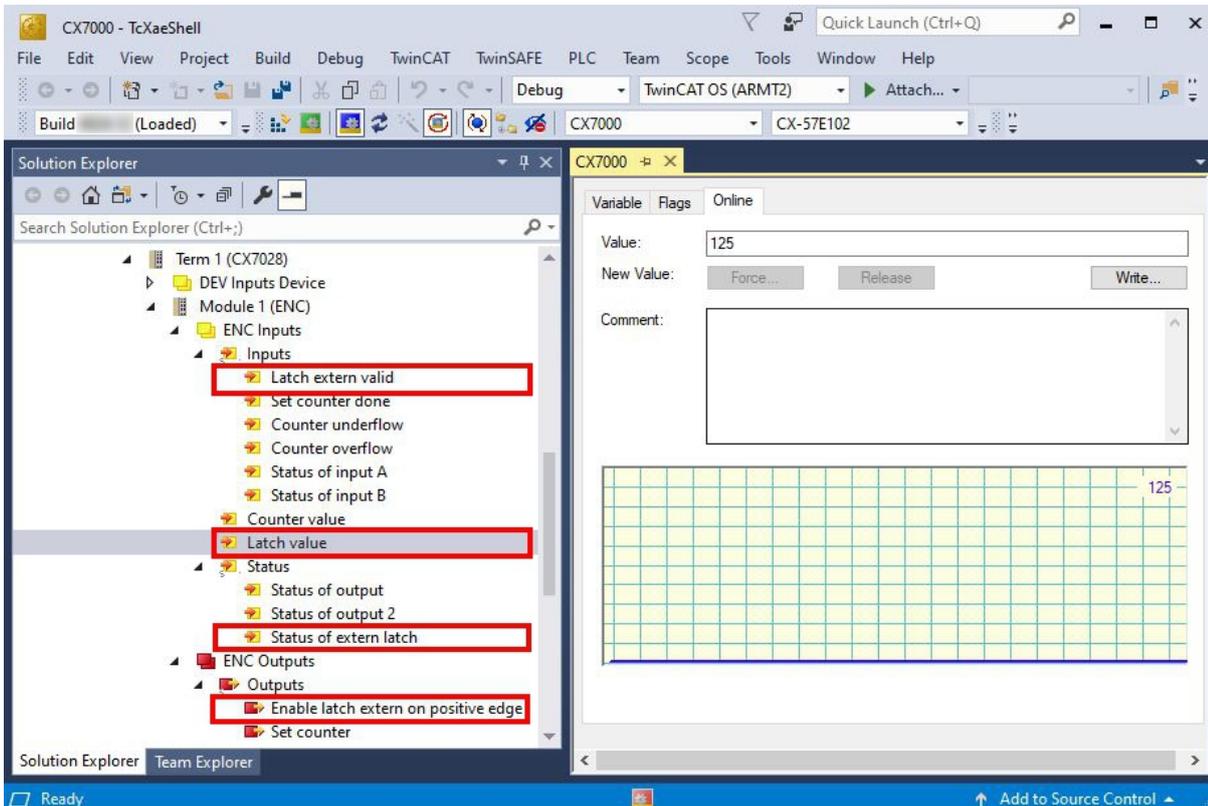
Ist der parametrisierte Zählerstand aus **Switch on/off threshold** erreicht bzw. überschritten, aber die Variable **Enable output functions** nicht gesetzt, wird der Schaltauftrag nicht ausgeführt. Sobald dann **Enable output functions** gesetzt wird, wird umgehend der Ausgang geschaltet. Genauso wirkt sich ein nachträglich aktivierter Zählerstand **Switch on/off threshold** bei erfüllter Schaltbedingung umgehend auf den Ausgang aus.

6.4.2 Zählerstand latchen

Im Incremental-Encoder-Modus kann der Zählerstand gelatcht und dadurch der aktuelle Wert in die Prozessdaten eingetragen werden. Als Latch-Eingang wird Eingang 3 verwendet.

Um die Funktion zu aktivieren, muss die Variable **Enable latch extern on positive edge** auf **True** gesetzt sein. Bei einem High-Pegel an Eingang 3 wird der aktuelle Zählerstand in die Variable **Latch Value** eingetragen. Sie können die Gültigkeit der Variable überwachen. Sobald der Latch-Wert eingetragen wird, wird auch die Variable **Latch extern valid** auf **True** gesetzt.

Gehen Sie wie folgt vor:



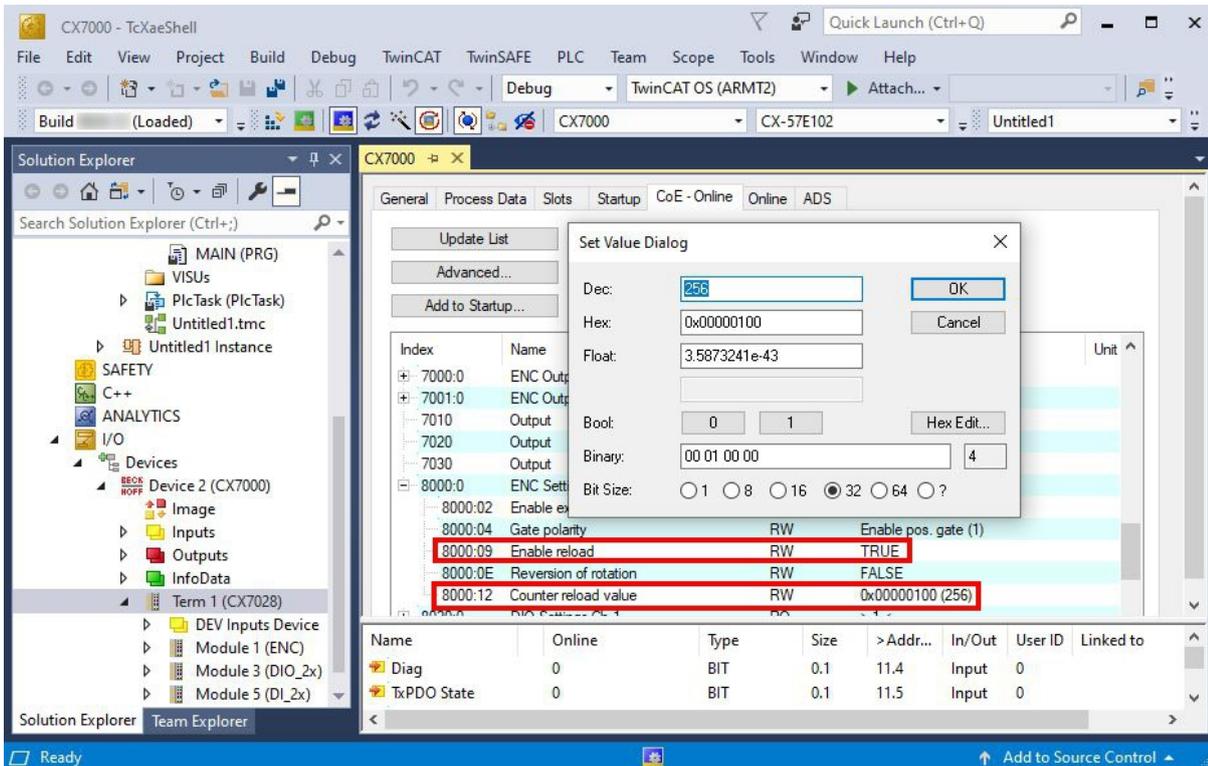
1. Setzen Sie die Variable **Enable latch extern on positive edge** auf **True**, um die Latch-Funktion zu aktivieren.
 2. Überwachen Sie den Status des Latch-Eingangs mit der Variable **Status of extern latch**.
 3. Bei einem High-Pegel an Eingang 3 wird der aktuelle Zählerstand in die Variable **Latch Value** eingetragen.
 4. Überwachen Sie die Gültigkeit des Latch-Wertes über die Variable **Latch extern valid**. Sobald der Latch-Wert geschrieben wird, wird auch die Variable auf **True** gesetzt.
- ⇒ Um erneut einen Latch auszuführen, muss die Variable **Enable latch extern on positive edge** wieder einen High-Pegel bekommen.

6.4.3 Grenzwert für Zähler festlegen

In diesem Schritt wird gezeigt, wie Sie in TwinCAT einen Grenzwert festlegen können, ab dem der Zählerstand automatisch wieder auf null zurückgesetzt wird. Beim Vorwärtszählen wird der Zählerstand beim Erreichen des Grenzwertes auf null zurückgesetzt. Beim Rückwärtszählen wird der Zählerstand beim Erreichen von Null auf den eingestellten Grenzwert zurückgesetzt.

Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8000:12 Counter reload value** und legen Sie den Grenzwert fest.
 4. Klicken Sie anschließend doppelt auf das CoE-Objekt **8000:09 Enable reload** und setzen Sie den Wert auf **True**.
- ⇒ Erst wenn **Enable reload** gesetzt ist, ist die Funktion aktiv. Alternativ kann der Latch-Eingang verwendet und damit der Zählerstand extern zurückgesetzt werden. Dafür muss die Latch-Funktion deaktiviert und das CoE-Objekt **Enable extern reset** auf **True** gesetzt werden. Mit dieser Einstellung wird bei einem High-Pegel an Eingang 3 der aktuelle Zählerstand auf null gesetzt.

Index	Name	Flags	Value	Unit
7020	Output	RO P	FALSE	
7030	Output	RO P	FALSE	
8000:0	ENC Settings	RO	> 18 <	
8000:02	Enable extern reset	RW	TRUE	
8000:04	Gate polarity	RW	Enable pos. gate (1)	
8000:09	Enable reload	RW	TRUE	
8000:0E	Reversion of rotation	RW	FALSE	
8000:12	Counter reload value	RW	0x00000100 (256)	
8020:0	DIO Settings Ch.1	RO	> 1 <	

6.5 Analog-Signal-Modus

Die Single-ended-Eingänge 7 und 8 erfassen Signale im Bereich von 0 bis 10 V.

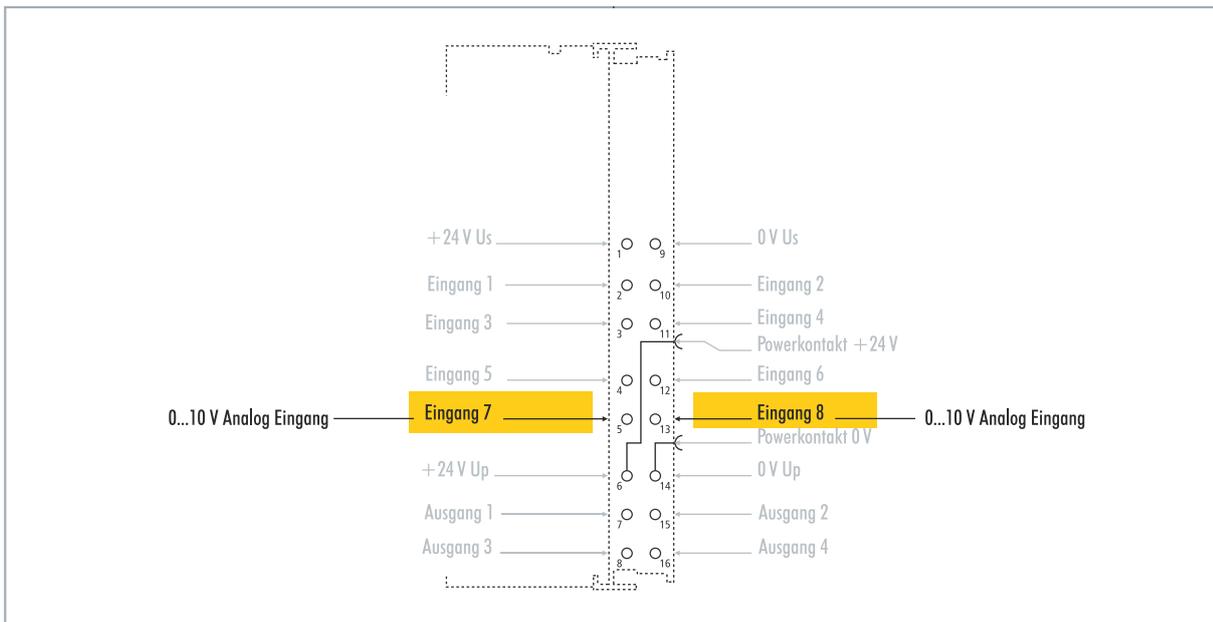


Abb. 23: Konfigurierbare analoge Eingänge.

Die Spannung wird mit einer Auflösung von 12 Bit digitalisiert. Der Signalzustand wird durch Leuchtdioden angezeigt.

Tab. 15: Technische Daten, Multifunktions-I/Os im Analog-Modus.

Technische Daten	CX7031
Technik	single-ended
Anzahl Eingänge	2
Signalspannung	0...10 V
Innenwiderstand	500 kΩ
Grenzfrequenz EingangsfILTER	2 kHz
Auflösung	12 Bit (16-Bit-Darstellung)
Messfehler	< ±0,3 % (bezogen auf den Messbereichsendwert)
Anschlussquerschnitt	e*: 0,08...1,5 mm ² , f*: 0,25...1,5 mm ² , a*: 0,14...0,75 mm ²
Anschlussquerschnitt AWG	e*: AWG 28...16, f*: AWG 22...16, a*: AWG 26...19
Abisolierlänge	8...9 mm

*e: eindrätig, Draht massiv; f: feindrätig, Litze; a: mit Aderendhülse

6.6 PWM-Signal-Modus

HINWEIS

Rückspeisung bei den 24-V-Ausgängen

Eine Spannung von 24 V an den Ausgängen 3 und 4 kann das Gerät zerstören (Rückspeisung). Im PWM-Modus darf keine Spannung an den Ausgängen angelegt werden.

Der PWM-Signal-Modus ermöglicht auf Ausgang 3 und Ausgang 4 ein binäres Signal pulswellenmoduliert auszugeben.

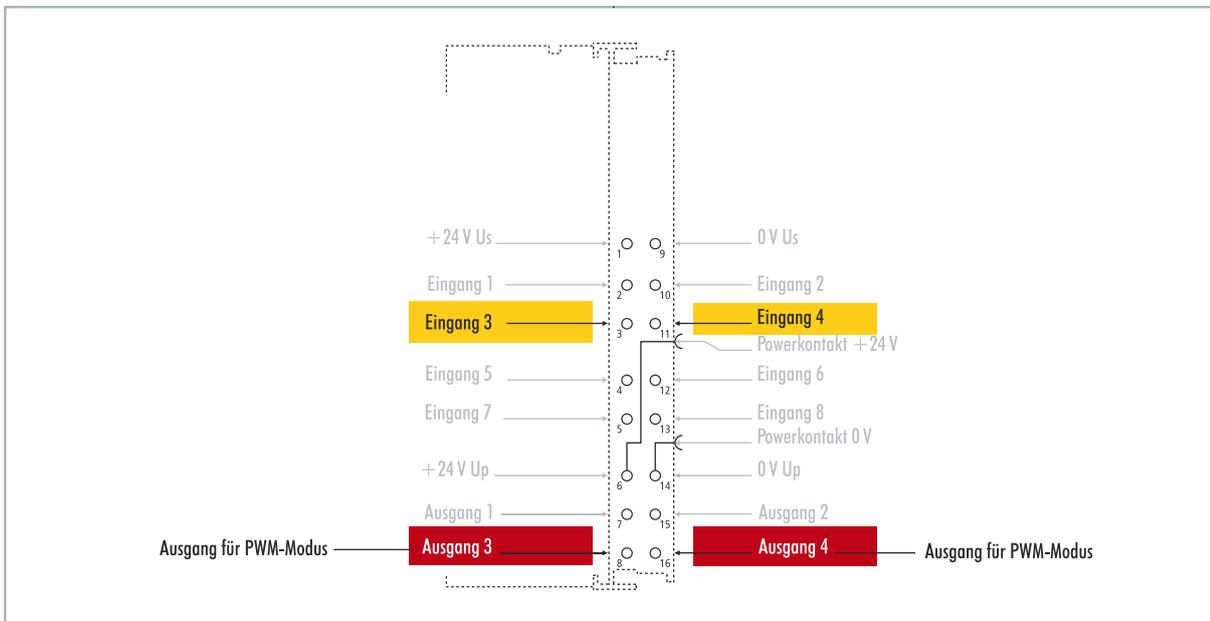
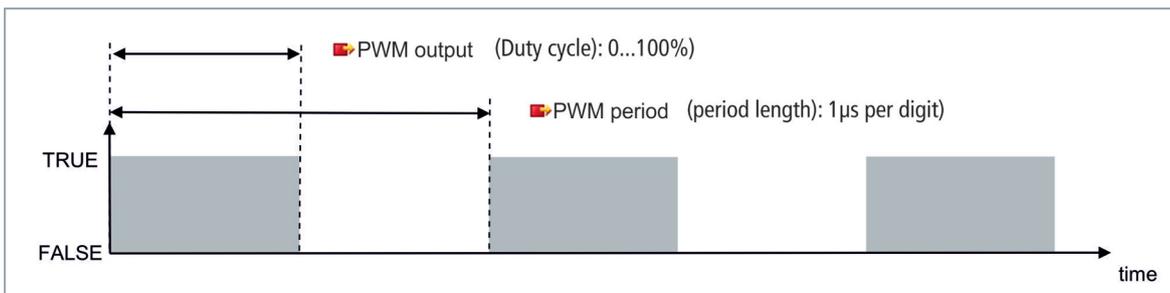


Abb. 24: Konfigurierbare Ein- und Ausgänge im PWM-Signal-Modus

Dieses Signal ist getrennt in Tastverhältnis (0... 100%) und PWM-Taktfrequenz (15 Hz... 100 kHz). Die LEDs sind mit den Ausgängen getaktet und zeigen durch ihre Helligkeit das Tastverhältnis an. Die Signalwerte werden in 16-Bit-Werten übergeben.



Tab. 16: Technische Daten, Multifunktions-I/Os im PWM-Modus.

Technische Daten	Digitale Eingänge
Anschlusstechnik	PWM-Ausgang
Anzahl Ausgänge	2
Nennspannung	24 V DC (-15 %/+20 %)
Lastart	ohmsch, induktiv, Lampenlast
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
PWM-Taktfrequenz	15 Hz... 100 kHz
Tastverhältnis	0... 100 % ($T_{ON} > 20 \text{ ns}$, $T_{OFF} > 200 \text{ ns}$)
Kurzschlussstrom	< 2 A typ.
Besondere Eigenschaften	separate Frequenz für jeden Kanal einstellbar

Technische Daten	Digitale Eingänge
Anschlussquerschnitt	e*: 0,08...1,5 mm ² , f*: 0,25...1,5 mm ² , a*: 0,14...0,75 mm ²
Anschlussquerschnitt AWG	e*: AWG 28...16, f*: AWG 22...16, a*: AWG 26...19
Abisolierlänge	8...9 mm

*e: eindrätig, Draht massiv; f: feindrätig, Litze; a: mit Aderendhülse

6.6.1 PWM-Taktfrequenz und Tastverhältnis festlegen

Die Signale auf Ausgang 3 und Ausgang 4 werden pulsweitenmoduliert ausgegeben, wobei die Signale in Tastverhältnis und PWM-Taktfrequenz getrennt sind. Für beide Ausgänge lassen sich eigene Werte für Tastverhältnis und PWM-Taktfrequenz definieren.

Tab. 17: PWM output (Tastverhältnis), Darstellung des PWM-Signals im Auslieferungszustand.

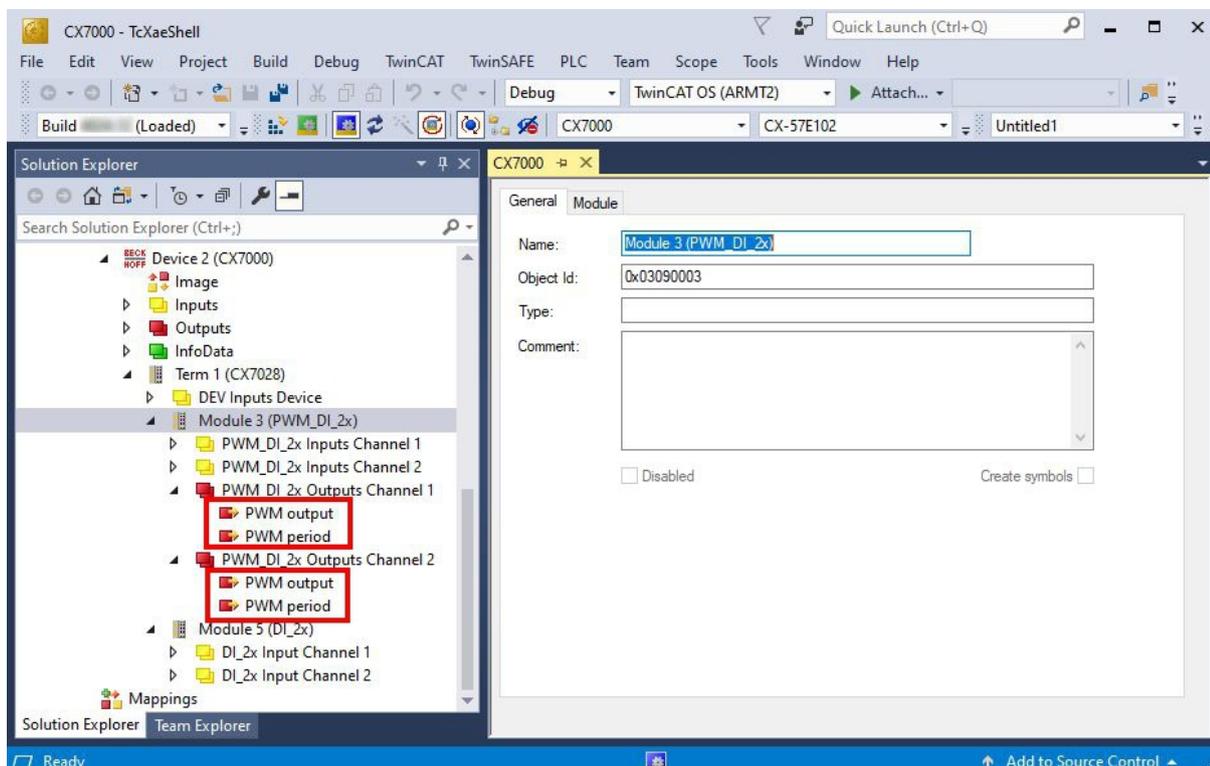
Wert	Dezimal	Hexadezimal
0 %	0	0x0000
25 %	16383	0x3FFF
50 %	32767	0x7FFF
100 %	65.535	0xFFFF

Tab. 18: PWM period (PWM-Taktfrequenz), Darstellung des PWM-Signals im Auslieferungszustand.

Wert	Dezimal	Hexadezimal	Frequenz
0,010 ms	0..10	0x0000-0x000A	100 kHz
0,011 ms	11	0x000B	90,909 kHz
0,100 ms	100	0x0064	10 kHz
1,000 ms	1000	0x03E8	1 kHz
16,38 ms	16383	0x3FFF	61,04 Hz
65,53 ms	65535	0xFFFF	15,26 Hz

Dabei entsprechen die Variablen **PWM output** dem Tastverhältnis und **PWM period** der PWM-Taktfrequenz, mit der das Signals ausgegeben wird.

Gehen Sie wie folgt vor:



1. Wählen Sie links im Strukturbaum einen Ausgang, für den Sie das Tastverhältnis und PWM-Taktfrequenz festlegen möchten.
2. Verknüpfen Sie die Variablen **PWM output** und **PWM period** mit den passenden Variablen aus Ihrem SPS-Projekt.
3. Legen Sie in den Variablen die Werte für Tastverhältnis und PWM-Taktfrequenz entsprechend der oben genannten Tabellen fest.

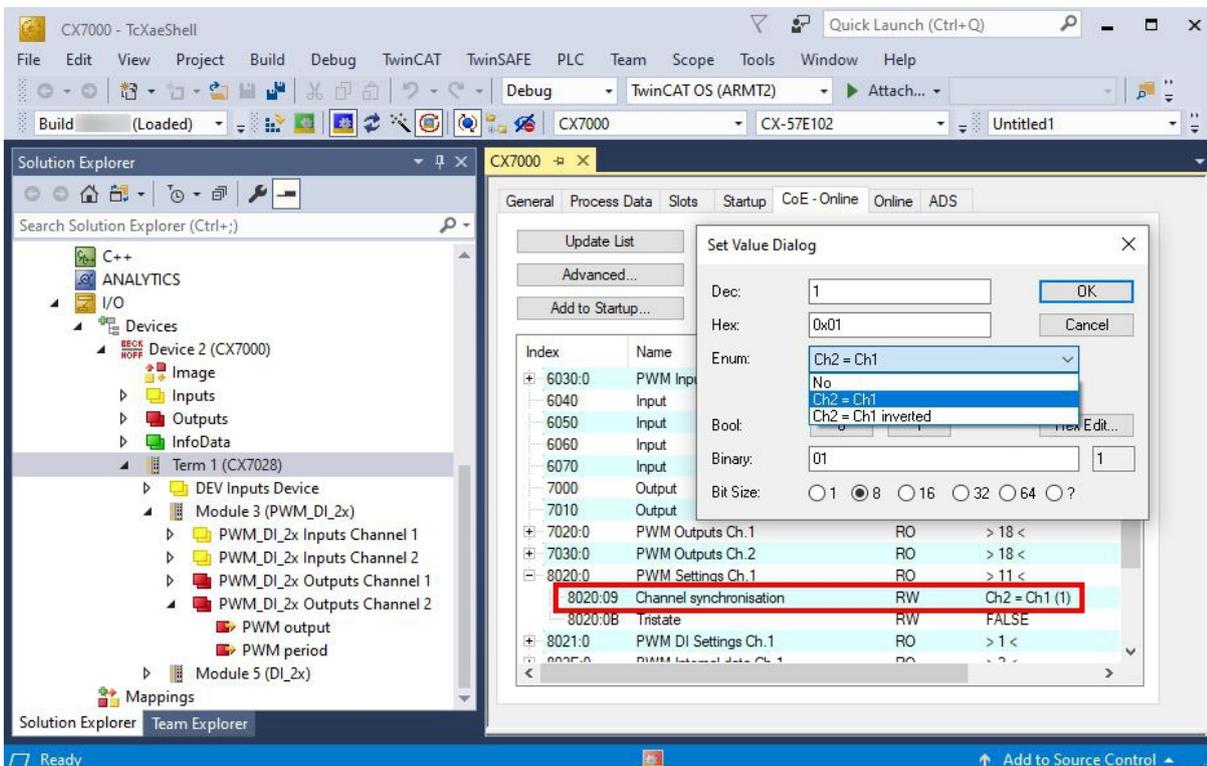
6.6.2 Kanalsynchronisation einstellen

Die Option Kanalsynchronisierung macht die Ausgabe von Ausgang 2 abhängig von Ausgang 1. In den CoE-Objekten stehen folgende Werte zur Verfügung:

- No: keine Abhängigkeit
- Ch2 = Ch1: Tastverhältnis und PWM-Taktfrequenz von Ausgang 1 werden auch auf Ausgang 2 angewendet. Die Phasenlage beträgt 0, d. h. dass steigende und fallende Flanken von Ausgang 1 und Ausgang 2 synchronisiert sind.
- Ch2 = Ch1 inverted: Tastverhältnis und PWM-Taktfrequenz von Ausgang 1 werden auch auf Ausgang 2 angewendet. Die PWM-Taktfrequenz wird allerdings invertiert. Die Phasenlage beträgt 0, d.h. dass eine steigende Flanke bei Ausgang 1 zugleich eine fallende Flanke an Ausgang 2 auslöst.

Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8020:09 Channel synchronisation**.
4. Wählen Sie unter der Option **Enum** die erforderliche Synchronisationsart.

7 Konfiguration

7.1 Beckhoff Device Manager starten

Mit dem Beckhoff Device Manager kann ein Industrie-PC per Fernzugriff mit Hilfe eines Webbrowsers konfiguriert werden. Der Zugriff erfolgt über das HTTP-Protokoll und Port 80 (TCP).

Voraussetzungen:

- Host-PC und Embedded-PC müssen sich im gleichen Netzwerk befinden. Die Netzwerkfirewall muss den Zugriff über Port 80 (HTTP) zulassen.
- IP-Adresse oder Hostname des Embedded-PCs.

Tab. 19: Zugangsdaten zum Beckhoff Device Manager bei Auslieferung.

Benutzername	Passwort
Administrator	1

Starten Sie den Beckhoff Device Manager wie folgt:

1. Öffnen Sie einen Webbrowser auf dem Host-PC.
2. Geben Sie die IP-Adresse oder den Hostnamen des Industrie-PCs im Webbrowser ein, um den Beckhoff Device Manager zu starten.
 - Beispiel mit IP-Adresse: <http://169.254.136.237/config>
 - Beispiel mit Hostnamen: <http://BTN-000f89fa/config>
3. Geben Sie den Benutzernamen und das Passwort ein. Die Startseite erscheint:

- ⇒ Navigieren Sie weiter im Menü und konfigurieren Sie den Industrie-PC. Beachten Sie, dass Änderungen erst nach einer Bestätigung wirksam werden. Gegebenenfalls muss der Industrie-PC neu gestartet werden.

7.2 Persistente Daten

HINWEIS

Applikationsbeispiel

In dem folgenden Beispiel kann das Verändern der Verbraucher, der Spannungsversorgung oder nur das Altern der Bauteile dazu führen, dass die Applikation nicht mehr die gewünschte Funktion erfüllt. Beckhoff übernimmt keine Verantwortung für die Umsetzung des Beispiels in einer Applikation.

Normalerweise werden persistente Daten nur beim TwinCAT-Stopp oder durch einen Baustein gespeichert. In diesem Kapitel wird gezeigt, wie Sie persistente Daten ohne USV auf einem CX7031 speichern können.

Bei einem Embedded-PC mit USV ist der Baustein in der Regel mit der USV verknüpft. Der Baustein wird aktiv, sobald ein Spannungsausfall erkannt wird, schreibt die persistenten Daten und fährt dann den Embedded-PC herunter. Bei einer 1-Sekunden-USV entfällt das Herunterfahren des Embedded-PCs, da dafür zu wenig Zeit bleibt.

Bei einer Kleinststeuerung wie dem CX7031, der ohne 1-Sekunden-USV ausgeliefert wird, kann man diese Funktionalität trotzdem nutzen. Es muss lediglich ein Netzteil verwendet werden, welches genug Restenergie hat, um den CX7031 eine gewisse Zeit mit dieser Restenergie mit Spannung zu versorgen. Ein kleiner Test kann ihnen zeigen, ob das mit ihrem Netzteil möglich ist:

Netzteil testen

Wenn der CX7031 läuft, schalten Sie die AC-Spannung ihres Netzteils ab und messen Sie, wie lange der CX7031 noch weiterläuft. Wenn es mehr als drei Sekunden sind, können Sie das Netzteil unter Umständen als Ersatz für eine 1-Sekunden-USV verwenden. Beachten Sie, dass auch Netzteile altern und Kapazität verlieren. Sie sollten daher einen Sicherheitsfaktor einrechnen, beispielsweise einen Faktor von drei, sodass sie genug Reserve haben, um über eine längere Zeit das Netzteil als Ersatz für eine 1-Sekunden-USV betreiben zu können.

Bestimmen Sie nun, wie lange das Netzteil die Spannungsversorgung aufrechterhält. Dafür brauchen Sie eine EL1722, die Sie an die AC-Seite des Netzteils anschließen. Dann schreiben Sie ein kleines Programm:

```
VAR
    bPower230V AT %I* : BOOL; (*link to the EL1722*)
END_VAR

VAR RETAIN
    Counter : INT;
END_VAR

Program:
IF NOT bPower230V THEN (*bPower230V is linked to the EL1722*)
    Counter:=counter+1; (*the counter is a retain value*)
END_IF
```

Erzeugen Sie ein Bootprojekt und schalten Sie die AC-Spannung des Netzteils ab. Sobald die EL1722 keinen Wert mehr anzeigt, wird der Counter hochgezählt und die Daten in das interne NOVRAM kopiert. Schalten Sie die AC-Spannung wieder ein und loggen Sie sich ein. Den Wert, den der Counter hat, müssen Sie nun mit der Taskzeit multiplizieren. Wiederholen Sie das ein paar Mal, damit Sie sicher sind, dass sich das Netzteil immer gleich verhält. Als Nächstes müssen Sie den Baustein `FB_WritePersistentData` einfügen. Dieser ist in der `Tc2_Uilities` Library enthalten (unter dem Ordner „TwinCAT PLC“).

Bestimmen Sie anschließend, wie lange das Speichern der persistenten Daten dauert. Wiederholen Sie auch diesen Vorgang ein paar Mal, damit Sie einen konstanten Wert erhalten und bei Schwankungen einen Maximalwert bestimmen können. Die benötigte Zeit können Sie über das Busy-Flag bestimmen. Der Baustein wird bearbeitet, solange das Busy-Flag gesetzt ist. Den ermittelten Wert multiplizieren Sie noch mit zwei, um einen weiteren Sicherheitsfaktor einzubauen.

Beispiel:

Sie messen, dass das Netzteil drei Sekunden die Spannungsversorgung aufrechterhält und die persistenten Daten in ca. 400 ms geschrieben werden. Mit den empfohlenen Sicherheitsfaktoren hält das Netzteil eine Sekunde und die persistenten Daten werden in ca. 800 ms geschrieben.

Die Spannungsversorgung wird also für einen längeren Zeitraum aufrechterhalten, als für das Speichern der persistenten Daten benötigt wird. Damit können Sie das Beispiel-Netzteil als Ersatz für die 1-Sekunden-USV nutzen.

7.3 NOVRAM

Das NOVRAM kann dazu verwendet werden, um wichtige Variablenwerte, wie z.B. Betriebsdaten oder Zählerstände bei einem Spannungsausfall sicher zu speichern. Die Speichergröße des NOVRAMs ist beschränkt und eignet sich für kleinere Datenmengen bis zu einer Größe von maximal 4 kB.

In diesem Kapitel wird gezeigt, wie das NOVRAM unter in TwinCAT 3 verwendet wird.

Funktionsweise

Das NOVRAM (Non-Volatile Random Access Memory) ist ein spezieller Speicherbaustein der dazu verwendet wird, um wichtige Daten sicher zu speichern. Das NOVRAM besteht aus zwei Bereichen. Einem volatilen Speicher und einem non-volatilen Speicher.

TwinCAT schreibt nur in den volatilen Speicher des NOVRAMs. Bei einem Spannungsausfall werden die Daten automatisch aus dem volatilen Speicher in den non-volatilen Speicher kopiert. Die dafür notwendige Energie liefert ein Kondensator. Sobald die Spannungsversorgung wieder vorhanden ist, werden die Daten automatisch in den volatilen Speicher zurückkopiert und können in TwinCAT weiterverwendet werden.

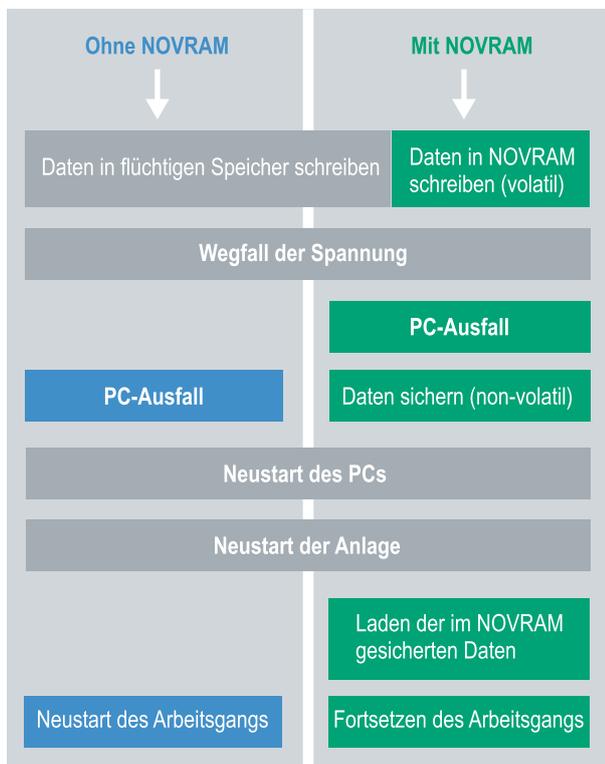


Abb. 25: Verhalten der Steuerung ohne und mit NOVRAM.

Speichergröße

Das NOVRAM hat eine Speichergröße von 4 kB. Die Daten werden zyklisch und wechselweise nach dem Doppelpufferprinzip gespeichert, um damit das Risiko von Dateninkonsistenz zu vermeiden.

Voraussetzungen

Entwicklungsumgebung	Zielpattformen	Hardware	Einzubindende SPS-Bibliotheken
TwinCAT 3.1 Build: 4020	PC oder CX (x86, x64, Arm®)	CX70xx, CX9020, CX20x0, CX20x2, CX20x3	Tc2_IoFunctions

7.3.1 Retain-Handler anlegen

Unter TwinCAT 3 ab Build 4020 wird ein Delta-Algorithmus benutzt, um Daten im NOVRAM zu speichern. Der Algorithmus speichert nicht alle Variablen auf einmal, sondern sucht nach Änderungen (Deltas) im Vergleich zum letzten Zyklus und speichert nur veränderte Variablen im NOVRAM.

Um den Delta-Algorithmus zu nutzen, muss in TwinCAT 3 ein Retain-Handler angelegt werden und die relevanten Variablen in der SPS mit dem Schlüsselwort VAR_RETAIN deklariert werden.

Neu an dieser Methode ist, dass keine Funktionsbausteine benutzt werden müssen. Der Retain Handler speichert Daten in das NOVRAM und stellt sie nach einem Spannungsausfall wieder bereit.

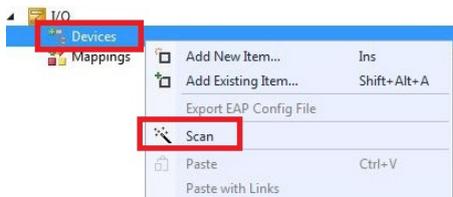
In diesem Kapitel wird beschrieben, wie Sie den Retain-Handler in TwinCAT 3 anlegen. Der Retain Handler speichert Daten in das NOVRAM und stellt sie wieder bereit. Dadurch bleiben wichtige Variablenwerte, wie z.B. Betriebsdaten oder Zählerstände, auch nach einem Neustart oder Spannungsausfall erhalten.

Voraussetzungen für diesen Arbeitsschritt:

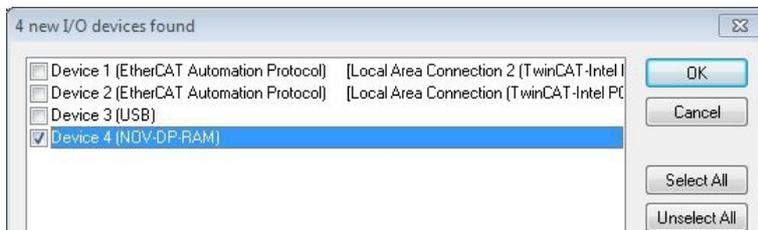
- TwinCAT 3.1 Build: 4020.
- Ein in TwinCAT ausgewähltes Zielgerät.

Legen Sie den Retain-Handler wie folgt an:

1. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf **Devices**.
2. Klicken Sie im Kontextmenü auf **Scan**.

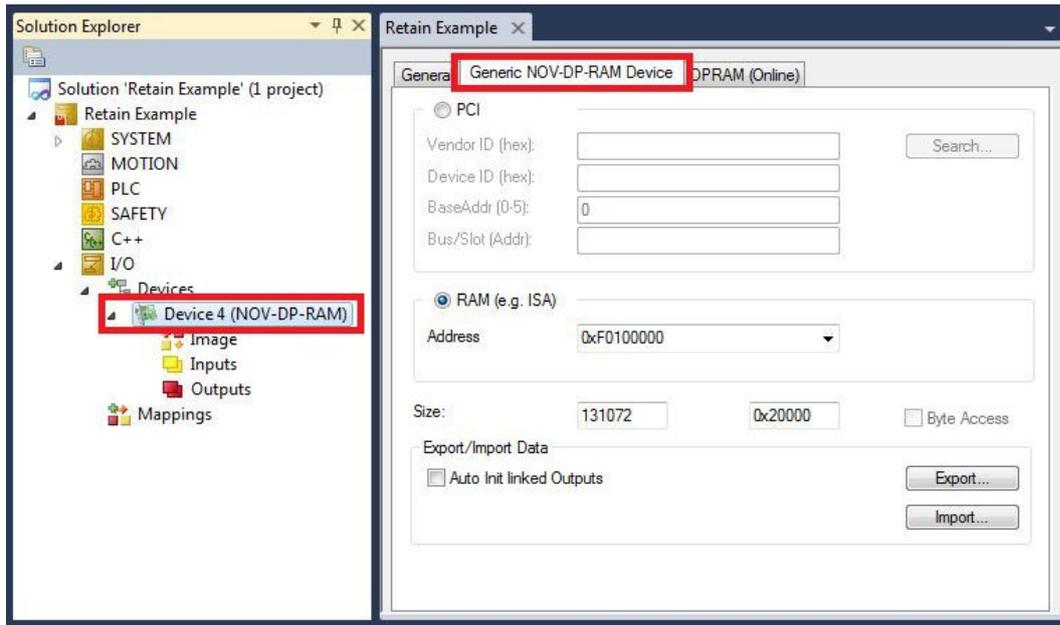


3. Wählen Sie **Device (NOV-DP-RAM)** und bestätigen Sie die Auswahl mit **OK**.

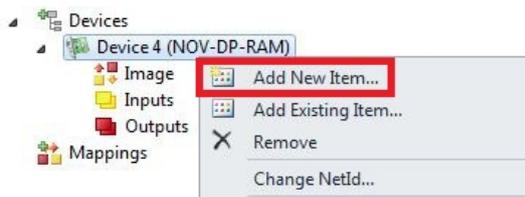


4. Klicken Sie auf **Ja**, um nach Boxen zu suchen.

5. Klicken Sie links in der Strukturansicht auf **Device (NOV-DP-RAM)** und anschließend auf die Registerkarte **Generic NOV-DP-RAM Device**.



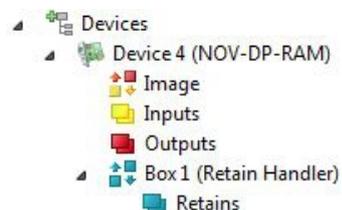
6. Klicken Sie auf die Option **RAM**.
7. Klicken Sie in der Strukturansicht mit rechter Maustaste auf **Device (NOV-DP-RAM)** und danach auf **Add New Item**.



8. Wählen Sie den **Retain Handler** und klicken Sie auf **OK**.



⇒ Sie haben erfolgreich einen Retain-Handler in TwinCAT angelegt.



Im nächsten Schritt können Sie Retain-Variablen in der SPS anlegen und mit dem Retain-Handler verknüpfen.

7.3.2 Variablen anlegen und verknüpfen

Nachdem Sie in TwinCAT einen Retain-Handler angelegt haben, können Sie in der SPS-Variablen deklarieren und mit einem Retain-Handler verknüpfen. Die Variablen müssen in der SPS mit dem Schlüsselwort `VAR_RETAIN` gekennzeichnet werden.

Wenn selbstangelegte Datentypen (DUTs) als Retain-Daten verwendet werden, so müssen die Datentypen im TwinCAT-Typsystem vorhanden sein. Dazu müssen die Strukturen direkt als `STRUCT_RETAIN` angelegt werden, wobei dann alle Vorkommen der Struktur über den Retain-Handler behandelt werden.

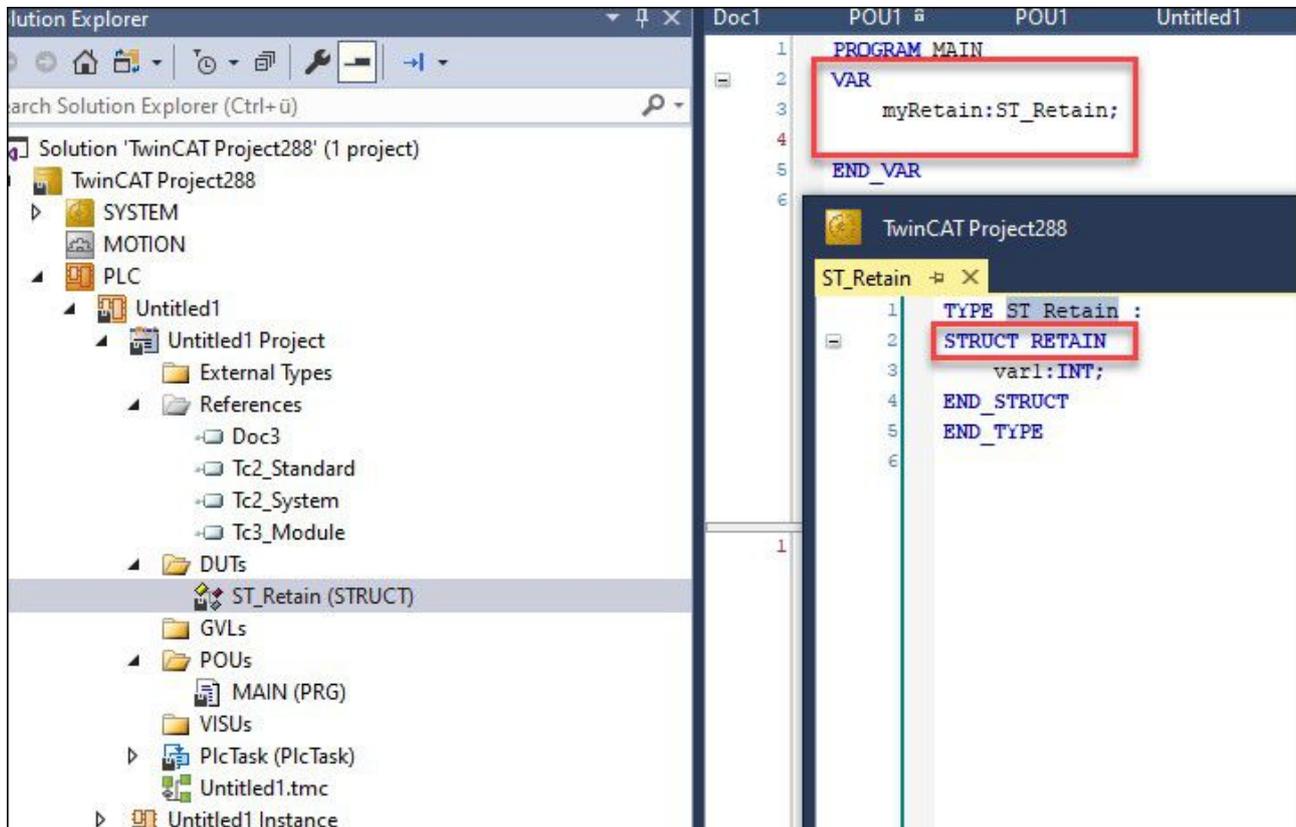


Abb. 26: Kennzeichnung mit Schlüsselwort `STRUCT_RETAIN` für selbstangelegte Datentypen (DUTs)

Für POU's (Funktionsbausteine) als Ganzes ist die Verwendung von Retain-Daten nicht möglich. Einzelne Elemente eines POU's können hingegen verwendet werden.

Voraussetzung für diesen Arbeitsschritt:

- Ein SPS-Projekt angelegt in TwinCAT.

Legen Sie Variablen wie folgt an:

1. Legen Sie die Variablen in Ihrem SPS-Projekt in einem `VAR_RETAIN`-Bereich an.

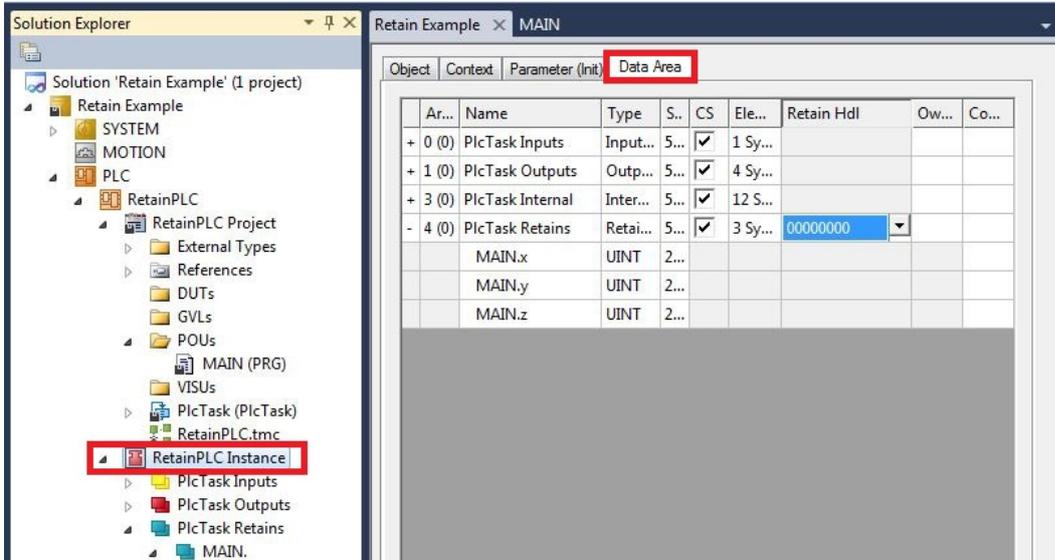
```

1  PROGRAM MAIN
2
3  VAR RETAIN
4      x      :UINT;
5      y      :UINT;
6      z      :UINT;
7  END_VAR
8
9  VAR
10
11     datain  AT$I*: REAL;
12     dataout AT$Q*: BYTE;
13
14 END_VAR
    
```

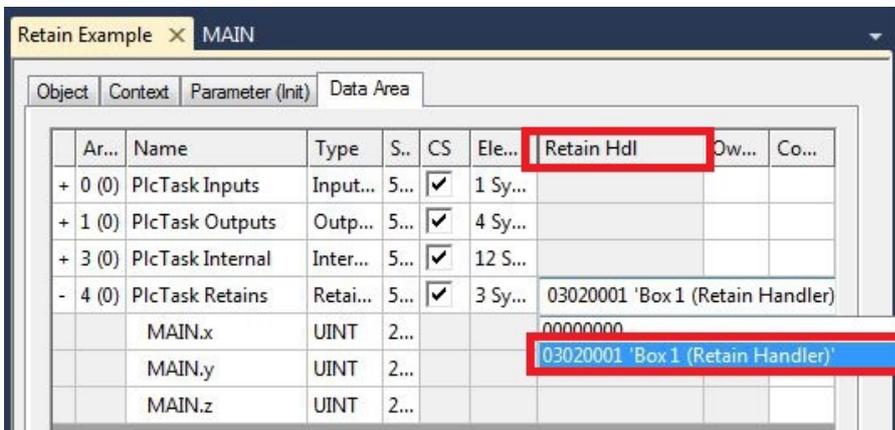
2. Verwenden Sie bei selbstangelegten Datentypen (DUTs) die Kennzeichnung mit Schlüsselwort `STRUCT RETAIN`.
3. Klicken Sie oben auf der Symbolleiste auf **Build** und dann auf **Build Solution**.



4. Klicken Sie links in der Strukturansicht auf Ihre **PLC Instance** und anschließend auf die Registerkarte **Data Area**.

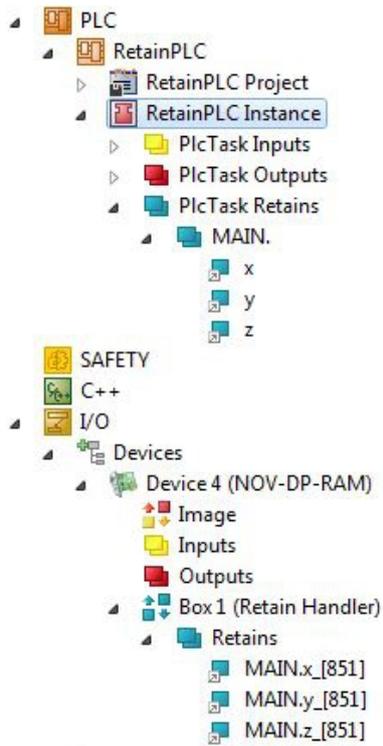


5. Wählen Sie unter **Retain Hdl** den Retain Handler, den Sie angelegt haben.



- ⇒ Nachdem Sie einen Retain-Handler als Ziel ausgewählt haben, werden die Symbole in der Strukturansicht verknüpft und ein Mapping erzeugt.
In der Strukturansicht werden die Variablen aus der SPS unter dem Retain-Handler angelegt und sind

mit den Variablen aus der SPS Instanz verknüpft.



Eine bestehende Verknüpfung wird mit einem Pfeilsymbol angezeigt.

7.3.3 Variablen unter dem Retain-Handler löschen

Wenn Variablen in der SPS gelöscht werden, dann wird die Verknüpfung mit dem Retain-Handler aufgehoben. Die Variablen werden aber unter dem Retain-Handler weiter angezeigt und werden nicht gelöscht.

Diese Variablen müssen unter TwinCAT 3 manuell gelöscht werden.

Voraussetzungen für diesen Arbeitsschritt:

- Mit VAR_RETAIN deklarierte Variablen wurden in der SPS gelöscht.

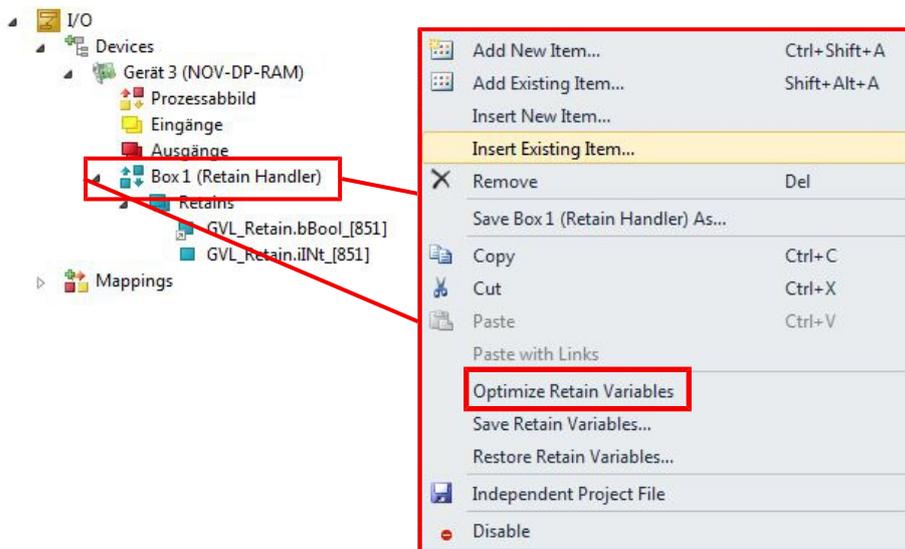
Löschen Sie Variablen unter dem Retain-Handler wie folgt:

1. Die Variable GVL_Retain.iInt unter dem Retain-Handler soll gelöscht werden.



2. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf den **Retain-Handler**.

3. Klicken Sie im Kontextmenü auf **Optimize Retain Variables**.



⇒ Die Variable unter dem Retain-Handler wird gelöscht.

7.4 Softwarekonfiguration

7.4.1 Benutzername und Passwort

Im Auslieferungszustand verfügt der CX7031 über einen voreingestellten Benutzernamen mit Passwort, welcher für die Anmeldung in TwinCAT oder dem Beckhoff Device Manager notwendig ist.

- Benutzername: Administrator
- Passwort: 1

Der Benutzername ist fix und kann nicht geändert werden. Es kann auch kein weiterer Benutzername hinzugefügt werden. Das voreingestellte Passwort kann über den Beckhoff Device Manager geändert werden (siehe: Beckhoff Device Manager starten). Das Passwort kann maximal 32 Zeichen enthalten. Erlaubt sind Zahlen, Buchstaben und Sonderzeichen, wobei auch zwischen Groß- und Kleinschreibung unterschieden wird.

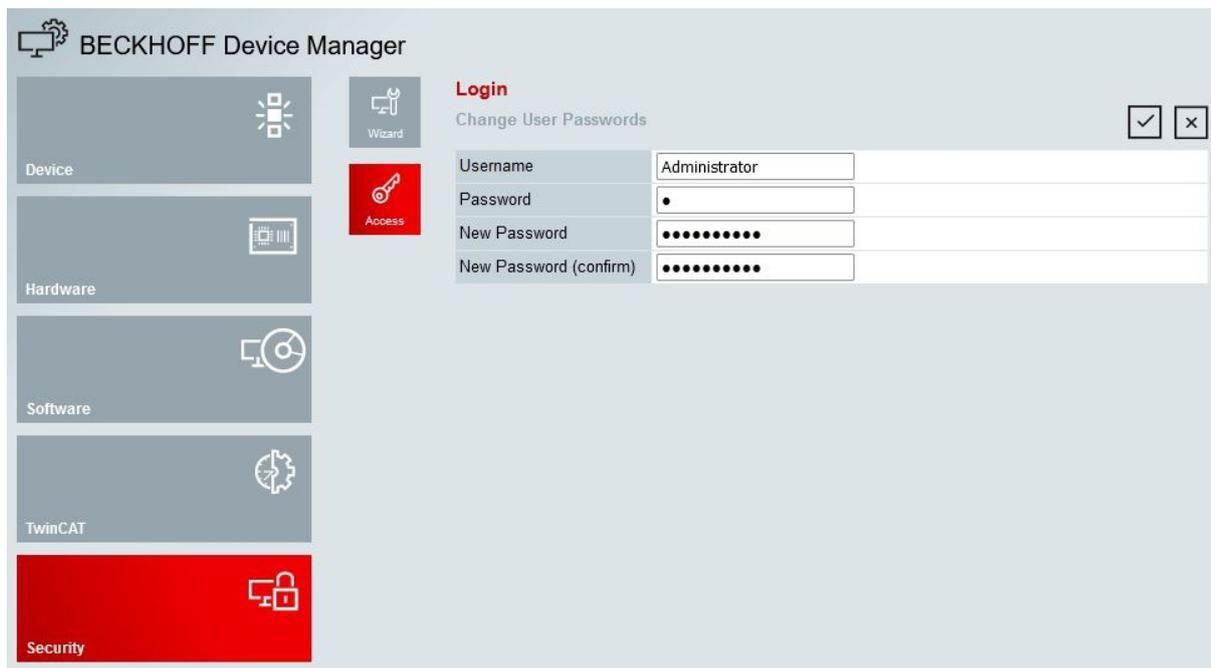


Abb. 27: Änderung des Passworts im Beckhoff Device Manager.

Sie können den Auslieferungszustand und das voreingestellte Passwort wiederherstellen, wenn Sie die MicroSD-Karte ausbauen, auf die MicroSD-Karte mit einem Kartenleser zugreifen und die Datei `device.conf` im Ordner `/etc` löschen. Ohne physischen Zugriff auf den CX7031 und damit auf die MicroSD-Karte, kann das Passwort nicht zurückgesetzt werden.

7.4.2 IP-Adresse einstellen

Beim CX7031 ist standardmäßig DHCP aktiv. Ohne DHCP-Server verwendet der CX7031 eine lokale IP-Adresse im Adressbereich 169.254.x.x

Beim Embedded-PC CX7031 gibt es mehrere Möglichkeiten die IP-Adresse einzustellen. Eine Möglichkeit besteht darin, den Beckhoff Device Manager aufzurufen und die IP-Adresse für den CX7031 im Browser einzustellen (siehe: Beckhoff Device Manager starten).

Eine andere Möglichkeit die IP-Adresse einzustellen, bietet die boot.conf-Datei, die nach dem ersten Start auf der MicroSD-Karte angelegt wird. In diesem Schritt wird gezeigt, wie Sie die IP-Adresse in der boot.conf-Datei einstellen können.

Voraussetzungen:

- MicroSD-Kartenleser

Gehen Sie wie folgt vor:

1. Schalten Sie den Embedded-PC aus und entfernen Sie die MicroSD-Karte aus dem Embedded-PC.
2. Öffnen Sie die Datei Boot.conf unter \etc

```

Boot.conf - Editor
Datei Bearbeiten Format Ansicht ?
; This is the CX7000 boot configuration file. It is highly recommend to set all network settings within TwinCAT XAE or
the website of the device and not to edit this file by hand.
; Comments can be added in lines beginning with a semicolon. However all comments and non supported settings get lost
once this file is updated.
; Lines should not exceed 256 characters and all relevant settings should be within 1000 lines.
; If values are not set, the wrong format is used, or if the file is missing, the device will start with the default
values.

; Devicename can be a string up to 63 ASCII characters. Permitted chars are 'a'-'z', 'A'-'Z', '0'-'9' and '-'. The
netbios name is derived from the first 15 characters.
; If the devicename has the four char prefix 'BTN-', the devicename will be automatically adjusted to have the eight
char BTN of the current device as suffix.
Devicename = BTN-00000099

; DhcpEnabled on Interface 0 can be true (default) or false.
0:DhcpEnabled = true

; IPv4 address on Interface 0 in dot decimal format. Only used if DhcpEnabled is false.
0:IPv4 = 0.0.0.0

; AutoIPv6Enabled on Interface 0 can be true (default) or false. If true, a link local IPv6 address is derived from
the MAC.
0:AutoIPv6Enabled = true

; IPv6 address on Interface 0 in colon hexadecimal. Short variants beginning with two colons are not supported. Only
used when AutoIPv6Enabled is false.
0:IPv6 = FE80:0000:0000:0000:0201:05FF:FE51:2619

; Netmask on Interface 0 in dot decimal format. Only used if DhcpEnabled is false or the DHCP resolve failed.
0:Netmask = 0.0.0.0

; Gateway IPv4 address on Interface 0 in dot decimal format. Only used if DhcpEnabled is false or the DHCP resolve
failed.
0:Gateway = 0.0.0.0

; DhcpEnabled on Interface 1 can be true (default) or false.

```

3. Stellen Sie den Eintrag **DhcpEnabled** auf **false**.
 4. Vergeben Sie eine IP-Adresse unter **IPv4**.
 5. Legen Sie die Einstellungen für Subnetzmaske, Gateway und DNS-Server fest.
- ⇒ Speichern Sie die Änderungen und bauen Sie die MicroSD-Karte wieder in den Embedded-PC ein. Die Einstellungen sind nach dem Start wirksam.

7.4.3 Virtual Local Area Network (VLAN) konfigurieren

Ab der Image-Version 140501 kann auf der Ethernet-Schnittstelle (X001) ein Virtual Local Area Network (VLAN) konfiguriert werden. Das VLAN bildet dabei ein logisches Netzwerksegment und eine eigene Broadcast-Domäne. Dies bedeutet, dass unabhängig von der physikalischen Netzwerktopologie vor Ort eine Unterteilung des Netzwerkes erfolgt. Der Datenverkehr der zugehörigen Geräte eines VLANs wird dabei von anderen VLANs isoliert.

Die Konfiguration erfolgt in der Datei `/etc/boot.conf` auf der MicroSD-Karte. Aktualisieren Sie zuerst das Image, wenn Sie eine ältere Image-Version auf dem Embedded-PC verwenden (siehe: [Image aktualisieren \[► 65\]](#)).

Voraussetzungen:

- Kartenleser für MicroSD-Karten

Gehen Sie wie folgt vor:

1. Schalten Sie den Embedded-PC aus und entfernen Sie die MicroSD-Karte aus dem Embedded-PC.
2. Stecken Sie die MicroSD-Karte in einen externen Kartenleser und öffnen Sie die Ordnerstruktur der MicroSD-Karte.
3. Öffnen Sie die Datei `boot.conf` unter `\etc` und navigieren Sie zu den Einträgen für das VLAN.

```
; VlanEnabled on Interface 0 can be true or false (default). If true, VLAN Tagging is supported.
0:VlanEnabled = false
; VlanID on Interface 0 can be set from 0 to 4094. When VLAN is enabled, it defines the
operating VLAN.
0:VlanID = 1
; VlanPriority on Interface 0 can be set from 0 to 7. When VLAN is enabled, it defines the
priority of the packets, sent with a VLAN Tag. 0 has a higher priority than 7.
0:VlanPriority = 0
```

4. Editieren Sie die Parameter `VlanEnabled`, `VlanID` und `VlanPriority` gemäß Ihren Anforderungen.
- ⇒ Speichern Sie die Änderungen und bauen Sie die MicroSD-Karte wieder in den Embedded-PC ein. Die Einstellungen sind nach dem Start wirksam.

7.4.4 Image aktualisieren

HINWEIS

Ausfall der Spannungsversorgung

Der Bootloader kann beschädigt werden, wenn die Aktualisierung unterbrochen wird. Der CX70x0 wird damit unbrauchbar und muss eingeschickt werden. Sorgen Sie für eine stabile Spannungsversorgung während des ersten Starts und unterbrechen Sie nicht die Aktualisierung.

Das neue Image wird direkt auf die MicroSD-Karte kopiert, um das Image des Embedded-PCs zu aktualisieren. Das neue Image wird vom Beckhoff Service zur Verfügung gestellt. Führen Sie das Update nur nach Rücksprache mit dem Beckhoff Service durch.

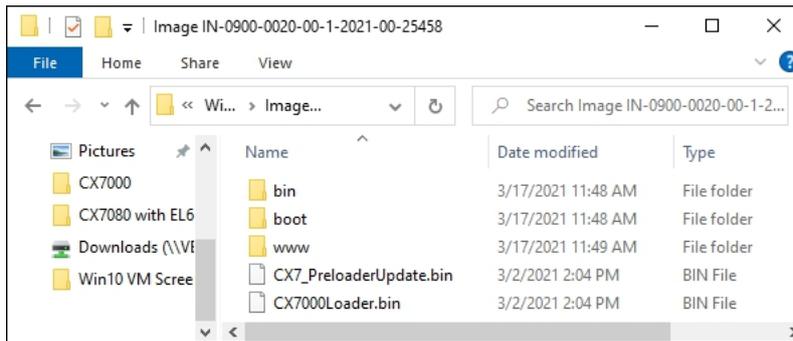
Voraussetzungen:

- Kartenleser für MicroSD-Karten.

Aktualisieren Sie das Image wie folgt:

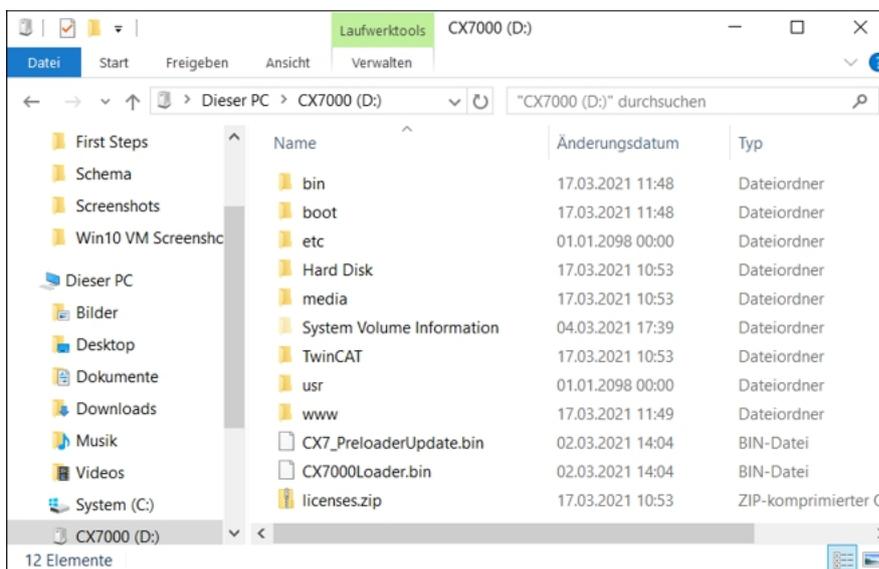
1. Schalten Sie den Embedded-PC aus und entfernen Sie die MicroSD-Karte aus dem Embedded-PC.
2. Stecken Sie die MicroSD-Karte in einen externen Kartenleser und öffnen Sie die Ordnerstruktur der MicroSD-Karte.
3. Löschen Sie alle Dateien und Ordner auf der MicroSD-Karte.

4. Kopieren Sie alle Dateien und Ordner des neuen Images auf die leere MicroSD-Karte.



5. Bauen Sie die MicroSD-Karte wieder in den Embedded-PC ein und starten Sie den Embedded-PC.

⇒ Der Embedded-PC wird gestartet und speichert die aktuelle Hardwarekonfiguration. Neue Ordner, wie beispielsweise Hard Disk oder TwinCAT, werden erzeugt. Damit wurde das Image erfolgreich aktualisiert.



7.4.5 Firmware für Multifunktions-I/Os aktualisieren

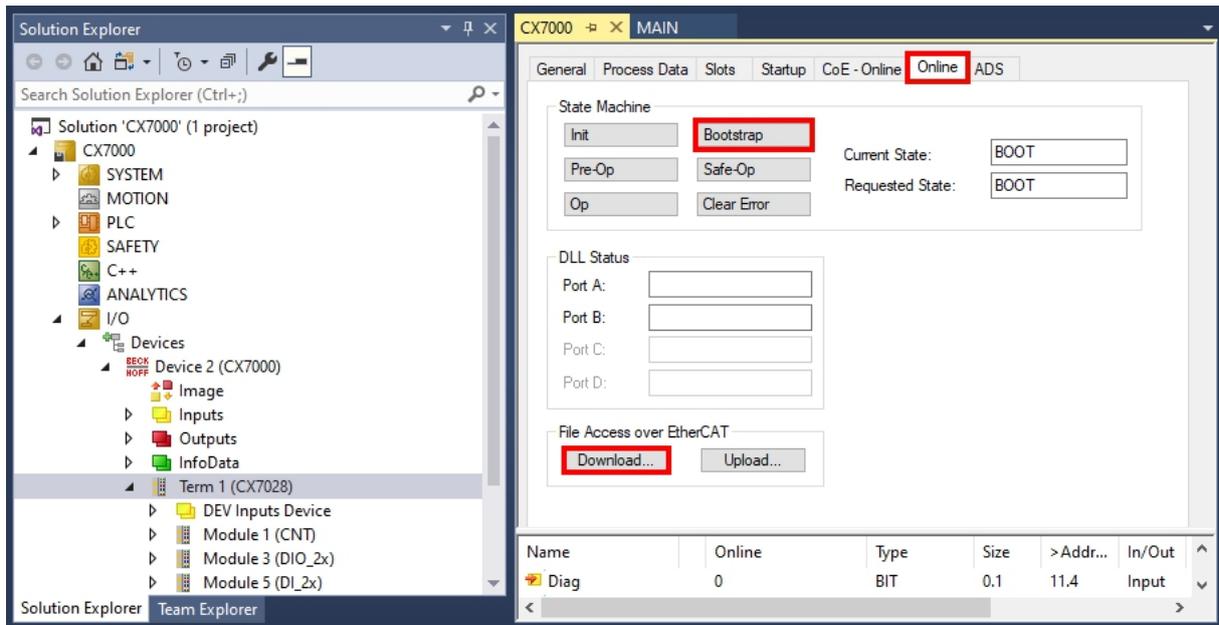
In diesem Schritt wird gezeigt, wie Sie die Firmware der Multifunktions-I/Os aktualisieren können. Die Firmware wird vom Beckhoff-Service zur Verfügung gestellt und die Aktualisierung in TwinCAT ausgeführt.

Voraussetzungen:

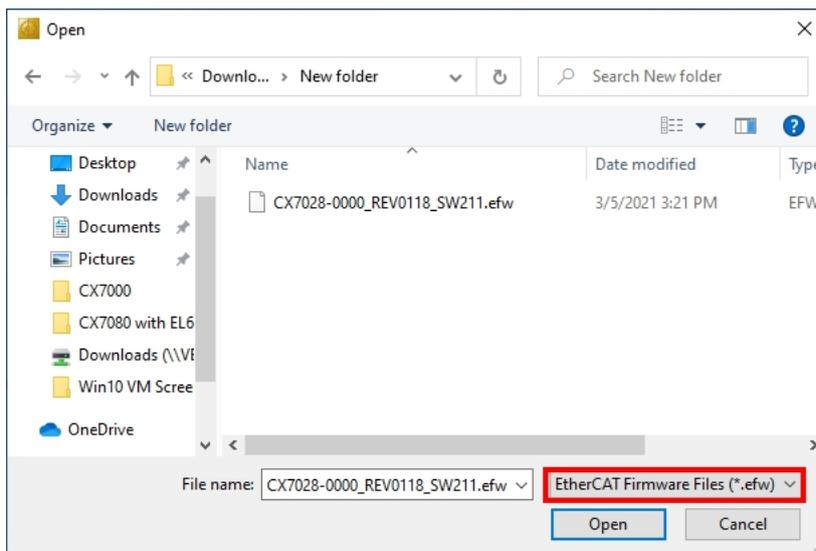
- EtherCAT-Firmware-File (*.efw)

Gehen Sie wie folgt vor:

1. Starten Sie TwinCAT im Konfigurationsmodus (Config-Modus).
2. Klicken Sie links im Strukturbaum auf das Gerät CX7028 und anschließend auf die Registerkarte **Online**.



3. Klicken Sie auf die Schaltfläche **Bootstrap**, um die Multifunktions-I/Os in den Zustand Bootstrap zu schalten.
4. Klicken Sie auf die Schaltfläche **Download** und wählen Sie eine aktuelle efw-Datei aus.



⇒ Die Aktualisierung benötigt ca. 3 bis 4 Minuten. Ein Fortschrittsbalken zeigt den Fortschritt der Aktualisierung an. Schalten Sie den CX7031 während dieser Zeit nicht aus.

Wechseln Sie nach Abschluss der Aktualisierung wieder in den Zustand Operational (Op), indem Sie auf die Schaltfläche **Op** klicken.

7.4.6 ESI-Gerätebeschreibung aktualisieren

Der TwinCAT System Manager und der TwinCAT EtherCAT-Master benötigt zur Konfiguration im Online- und Offline-Modus die Gerätebeschreibungsdateien aller EtherCAT-Geräte. Diese Gerätebeschreibungen sind die sogenannten ESI-Dateien (EtherCAT Slave Information) im XML-Format. Diese Dateien können vom jeweiligen Hersteller angefordert werden bzw. werden zum Download bereitgestellt. Eine *.xml-Datei kann dabei mehrere Gerätebeschreibungen enthalten.

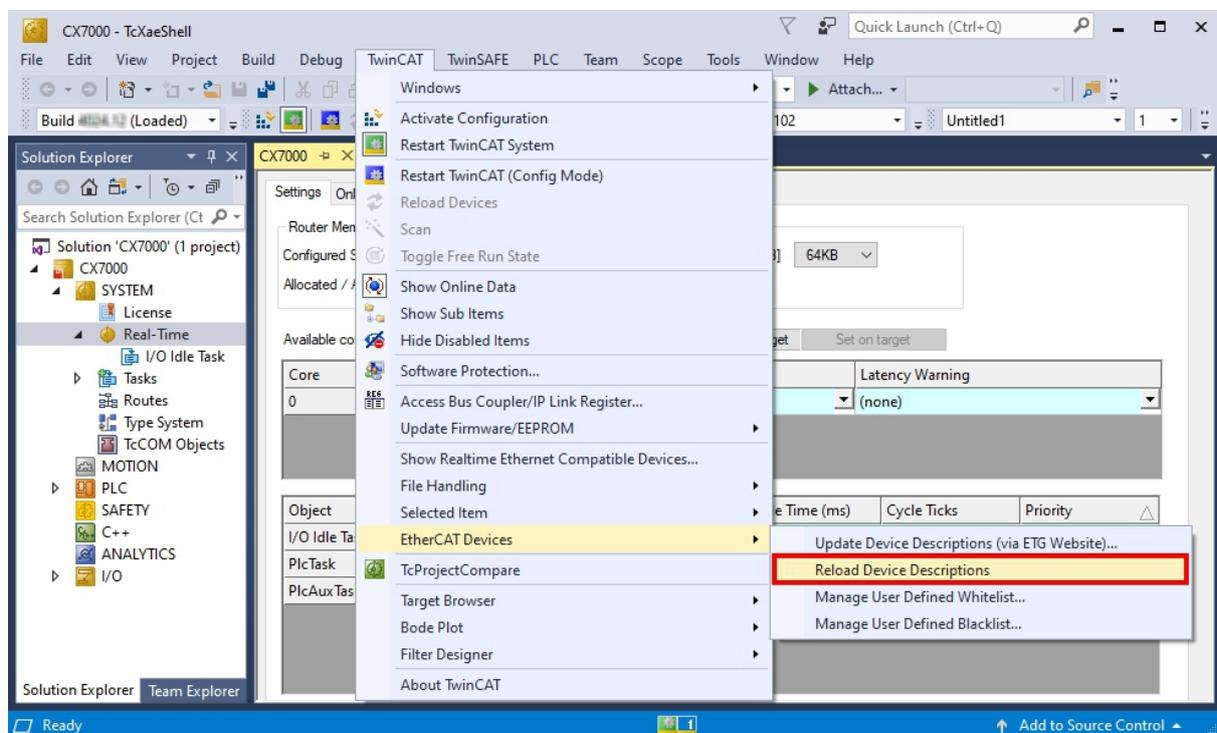
ESI-Dateien für Beckhoff EtherCAT-Geräte werden unter <https://www.beckhoff.com> bereitgestellt.

Voraussetzungen:

- ESI-Datei für den CX7031 im XML-Format.
- Gegebenenfalls die dazugehörige *.xsd-Datei, die den Aufbau der XML-Datei beschreibt.

Gehen Sie wie folgt vor:

1. Kopieren Sie die ESI-Datei in das TwinCAT-Installationsverzeichnis: `\TwinCAT\3.1\Config\Io\OnboardIo`.
2. Wenn der Ordner nicht existiert, erstellen Sie den Ordner manuell.
3. Öffnen Sie TwinCAT und klicken Sie im Menü unter **TwinCAT > EtherCAT Devices** auf **Reload Device Description**.



- ⇒ Die ESI-Datei wird in TwinCAT neu eingelesen. Sollte eine fehlerhafte ESI-Datei vorliegen, wird ein Fehler ausgegeben. Überprüfen Sie, ob Aufbau der *.xml der dazugehörigen *.xsd-Datei entspricht oder ob die Dateien zum CX7031 passen.

8 TwinCAT

8.1 Erste Schritte

8.1.1 Mit CX70x0 verbinden

Bevor Sie den CX7031 in TwinCAT konfigurieren können, müssen Sie eine Verbindung zwischen ihrem Engineering-Rechner und dem CX7031 (Zielsystem) herstellen. Der Engineering-Rechner und der Embedded-PC müssen sich im gleichen Netzwerk und Subnetz befinden oder alternativ direkt mit einem Ethernet-Kabel (Peer-to-Peer) verbunden werden.

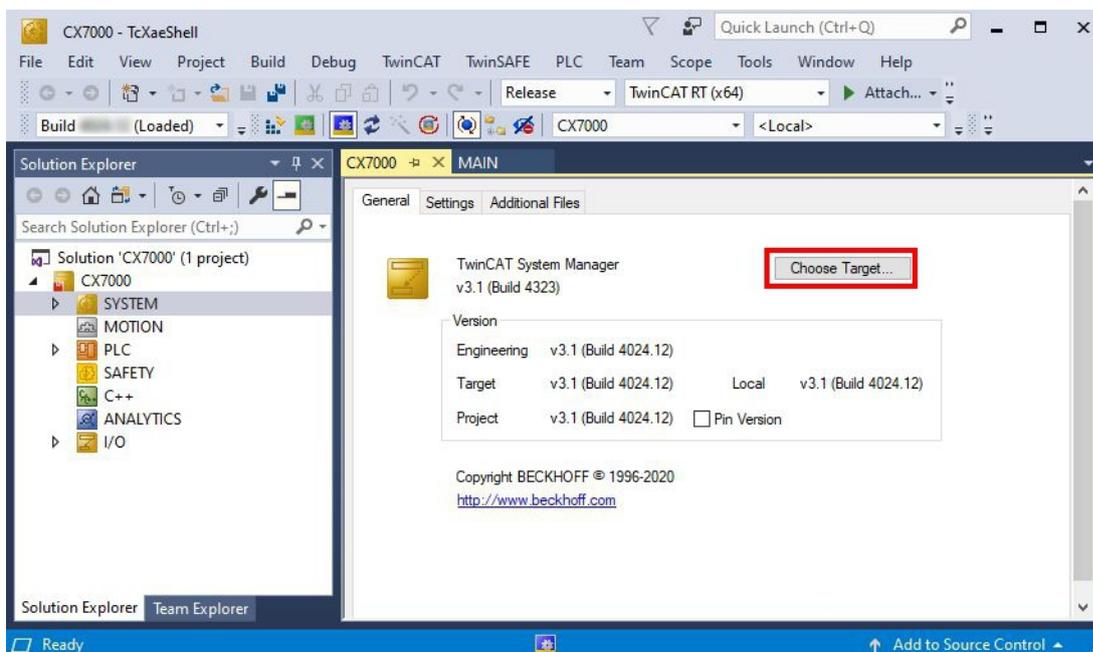
Für die Verbindung ist die IP-Adresse oder der Hostname des CX7031 notwendig.

Voraussetzungen:

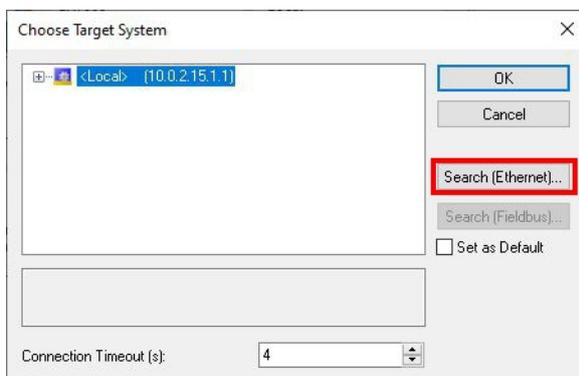
- TwinCAT muss sich im Config-Modus befinden.
- IP-Adresse oder Hostname des Embedded-PCs.

Stellen Sie eine Verbindung wie folgt her:

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)**.



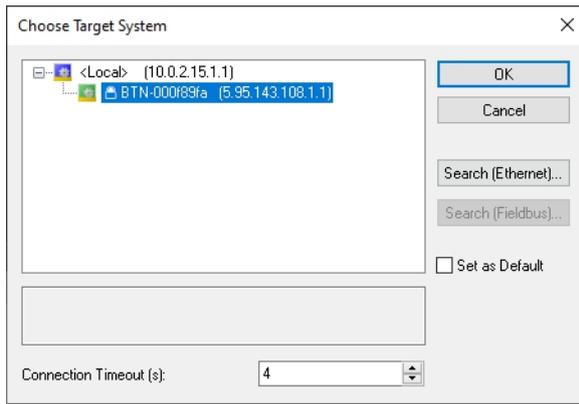
4. Klicken Sie auf **Broadcast Search** und suchen Sie nach verfügbaren Geräten im Netzwerk.

5. Markieren Sie den passenden CX7031 und klicken Sie auf **Add Route**. Der Hostname und die IP-Adresse erleichtern dabei die Identifikation.

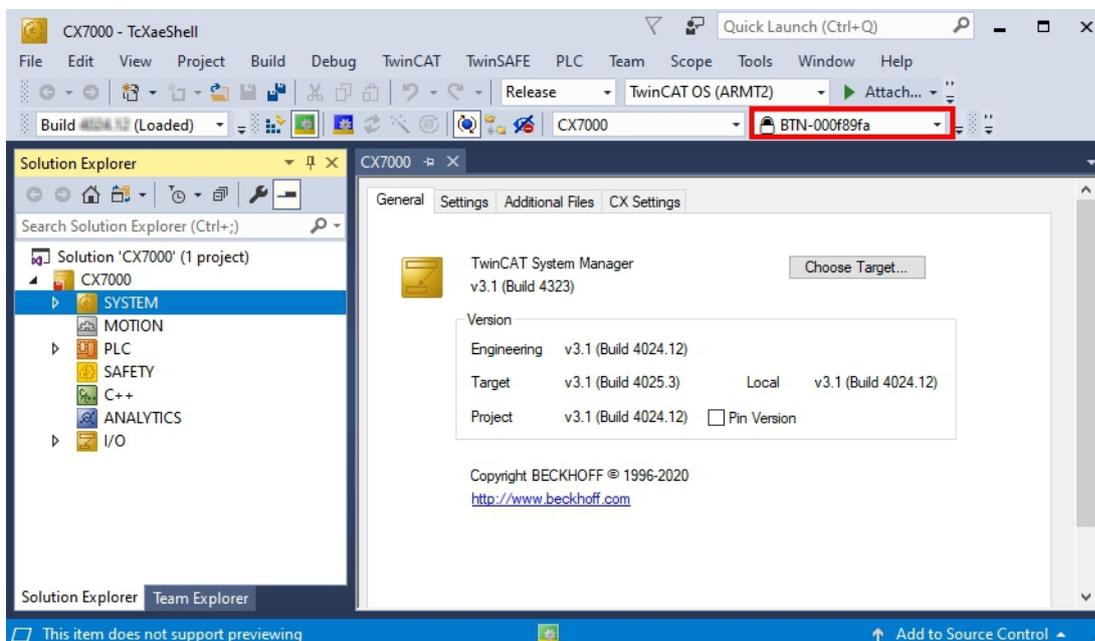
6. Geben Sie im Feld **User** und im Feld **Password** den Benutzernamen und das Passwort ein und klicken Sie auf **OK**. Benutzername: Administrator Passwort: 1

7. Das neue Gerät wird im Fenster **Choose Target System** angezeigt.

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



⇒ Sie haben erfolgreich in TwinCAT eine Verbindung zwischen ihrem Engineering-Rechner und dem CX7031 (Zielsystem) hergestellt. In der Menüleiste wird das neue Zielsystem mit dem Hostnamen angezeigt.



Mit dieser Vorgehensweise können Sie nach allen verfügbaren Geräten suchen und auch jederzeit zwischen den Zielsystemen wechseln.

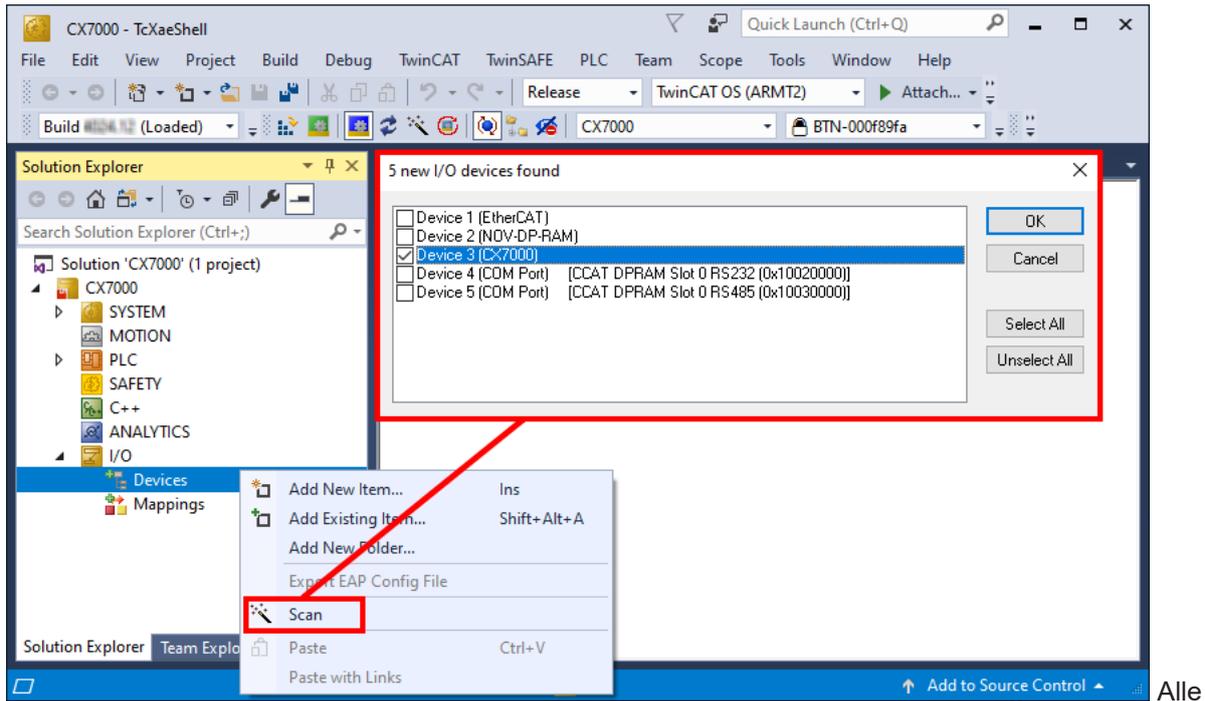
8.1.2 Multifunktions-I/Os scannen

Als Besonderheit verfügt die CX7000-Serie über acht integrierte Multifunktionseingänge und vier integrierte Multifunktionsausgänge. In diesem Kapitel wird gezeigt, wie Sie die Multifunktions-I/Os in TwinCAT scannen und anlegen können.

Beachten Sie, dass die CX7028-Schnittstelle für die Steuerung der Multifunktions-I/Os eine eigene CPU hat und die CX7028-Schnittstelle unter TwinCAT nicht angezeigt wird bzw. nicht funktioniert, wenn die Spannungsversorgung(U_p) nicht angeschlossen ist.

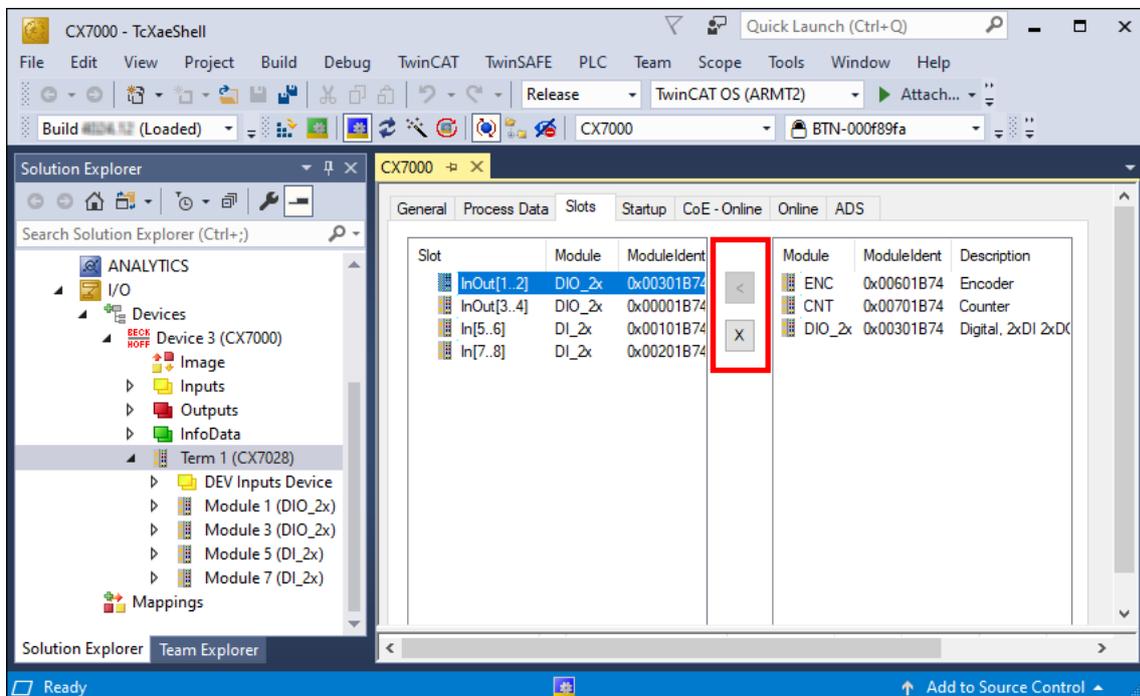
Gehen Sie wie folgt vor:

1. Klicken Sie links in der Struktursicht mit der rechten Maustaste auf **Devices** und anschließend auf **Scan**.



verfügbaren I/O-Geräte werden angezeigt.

2. Wählen Sie die passenden I/O-Geräte aus. Für dieses Beispiel muss mindestens die CX7028-Schnittstelle, also das CX7000-Gerät ausgewählt werden. Wenn Sie noch Bus- oder EtherCAT-Klemmen am CX7000 betreiben wollen, dann müssen Sie zusätzlich EtherCAT als Gerät auswählen.
3. Es werden insgesamt vier Slots angelegt. Für jeden Slot kann maximal ein Modul (DI, DIO, ENC, CNT oder PWM) zugewiesen werden, welches wiederum die Betriebsart für den jeweiligen Slot festlegt.



4. Mit der Schaltfläche < können Module einem bestimmten Slot zugewiesen oder mit x wieder entfernt werden.
- ⇒ Legen Sie die benötigten Module ihren Anforderungen entsprechend fest. Abhängig vom verwendeten Slot stehen unterschiedliche Module zur Auswahl. Welche Module von welchem Slot unterstützt werden, wird im Kapitel Multifunktions-I/Os [▶ 33] aufgelistet.

8.1.3 ADS-Kommunikation herstellen

In diesem Kapitel wird gezeigt, wie Sie einen CX7031 mit einem anderen CX70x0 oder einer beliebigen TwinCAT-Steuerung verbinden können. Das ADS-Protokoll bietet die einfachste Möglichkeit, zwei TwinCAT-Systeme miteinander zu verbinden. Mit dem ADS-Protokoll können Daten sowohl gelesen als auch geschrieben werden. Für gewöhnlich werden ADS-Bausteine für die Kommunikation verwendet, die in der Tc2_System-Bibliothek enthalten sind. In dem folgenden Beispiel sollen Daten von einem Speicherbereich gelesen und geschrieben werden.

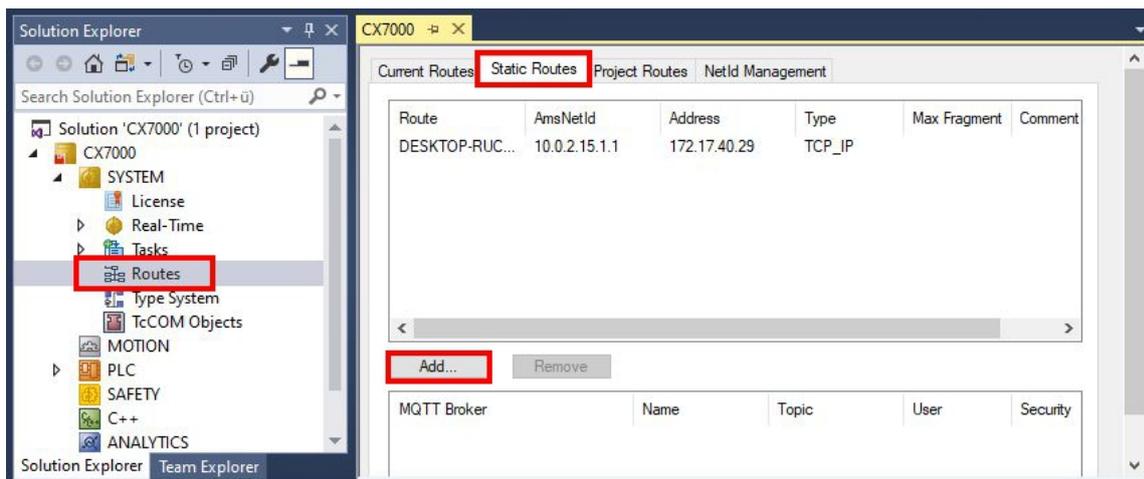
Um eine ADS-Verbindung einzurichten, wird zunächst eine ADS-Route erstellt. Die Kommunikation erfolgt dann über Ethernet und der Datenaustausch über das TCP/IP-Protokoll. Die ADS-Route ist dann die Schnittstelle zwischen der ADS- und TCP/IP-Verbindung. Durch die ADS-Route ist bekannt, welche AmsNetId welcher TCP/IP-Adresse zugewiesen ist. Dadurch wird in den ADS-Bausteinen nicht mehr die TCP/IP-Adresse, sondern nur noch die AmsNetId verwendet.

Voraussetzungen:

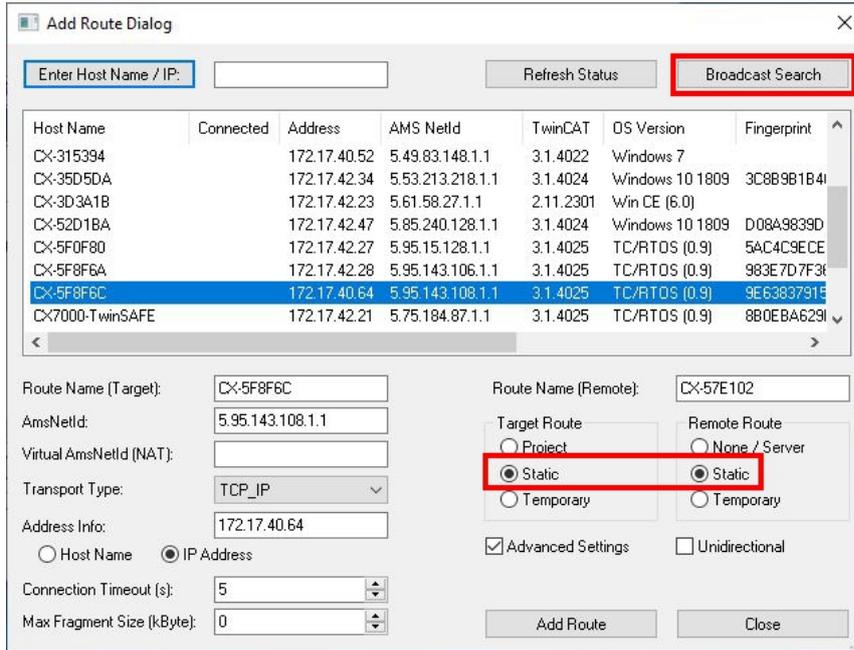
- Zwei CX70x0 Embedded-PCs.
- Beide CX70x0 sind im gleichen Netzwerk und über ADS erreichbar.

Gehen Sie wie folgt vor:

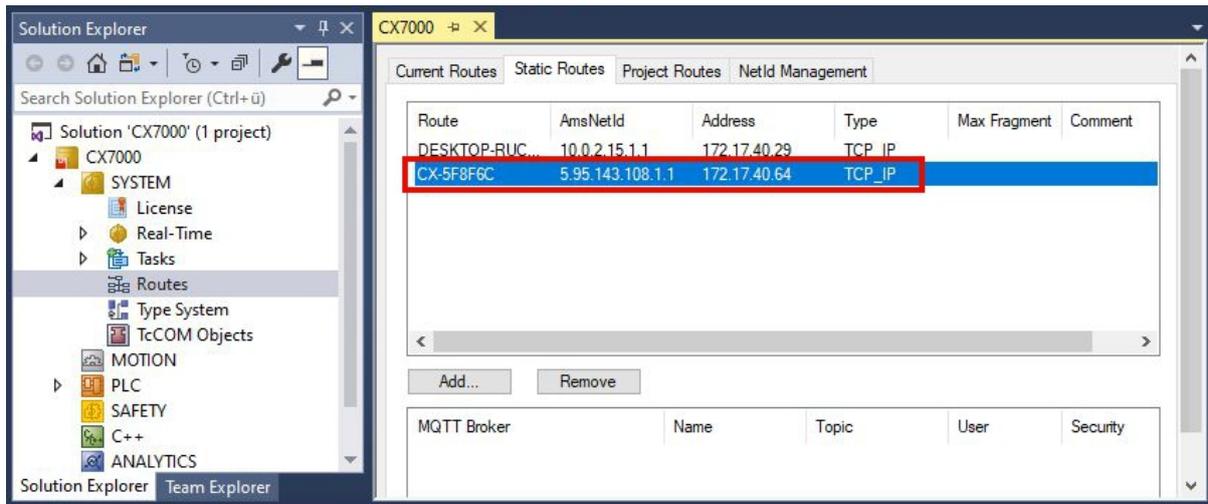
1. Starten Sie TwinCAT und verbinden Sie sich mit dem ersten CX70x0 (siehe: [Mit CX70x0 verbinden \[P 69\]](#)).
2. Klicken Sie links in der Strukturansicht auf **Routes**, wählen Sie die Registerkarte **Static Routes** und klicken Sie auf die Schaltfläche **Add**.



- Wählen Sie unter **Remote Route** die Option **Static**, damit die ADS-Route im Projekt bestehen bleibt und klicken Sie anschließend auf die Schaltfläche **Broadcast Search**.



- Wählen Sie den zweiten CX70x0 als Ziel der ADS-Route aus. Die ADS-Route wird bei beiden Embedded-PCs eingetragen. Die AmsNetId des zweiten CX70x0 wird angezeigt und kann im Programm für ADS-Bausteine verwendet werden.



- Verbinden Sie sich jetzt mit dem zweiten CX70x0, welcher als Ziel der ADS-Route festgelegt wurde und schreiben Sie ein kleines Programm. Definieren Sie ein Array und zählen Sie einen Wert des Arrays hoch.

```

VAR
    MarksTest AT %MBO      : ARRAY[0..9] of INT;
END_VAR

Program:
MarksTest[0]:=MarksTest[0]+1;
    
```

- Aktivieren Sie die Konfiguration und schalten Sie den CX70x0 in den Run-Modus.
- Schreiben Sie für den ersten CX70x0 ein Programm, welches den hochgezählten Wert des Arrays ausliest.

```

VAR
    ADSREAD : ADSREAD;
    NetID : STRING:='5.81.38.23.1.1'; (* AMSNetId of the target*)
    Value : INT; (* value of target MarksTest[0]*)
    Error : INT;
    NoError : INT;
END_VAR
    
```

```

Program:
  ADSREAD(
    NETID:=NetID ,
    PORT:=851 , (* plc port of the target*)
    IDXGRP:=16#4020 , (* Marks %MB*)
    IDXOFFS:=0 , (* Marks offset in byte*)
    LEN:=2 , (* length of data in byte*)
    DESTADDR:=ADR(Value) , (* pointer to the data in which the value is to be stored *)
    READ:=TRUE ,
    TMOUT:= ,
    BUSY=> ,
    ERR=> ,
    ERRID=> );
  IF NOT ADSREAD.BUSY THEN
    IF NOT ADSREAD.ERR THEN
      NoError:=NoError+1;
    ELSE
      Error:=Error+1;
    END_IF
  ADSREAD(Read:=FALSE);
  END_IF

```

8. Der hochgezählte Wert wird ausgelesen und an den ersten CX70x0 übermittelt.

⇒ Sie sollten beim ersten CX70x0 sehen, wie der Wert der Variable `Value` hochgezählt wird. Analog dazu funktioniert das Schreiben der Daten. Daten können mit dem Baustein `ADSWRITE` geschrieben werden. Achten Sie darauf, dass Sie in diesem Beispielaufbau den Offset (`IDXOFFSET`) auf 10 setzen, damit das Array [4...9] beschrieben wird. Beschränken Sie die Länge auf 10 Byte, da ein Array von 0...9 vom Typ `INT` angelegt wurde und damit der Speicher `%MB0...MB19` (10 * 2 Byte) verwendet (Die Elemente 0...4 zum Lesen und die Elemente 5...9 zum Schreiben des Arrays).

Verwenden Sie einen ADS-Befehl nach dem anderen. Warten Sie, bis der ADS-Dienst fertig ist, also der Ausgang `BUSY` des Bausteins auf `FALSE` geschaltet wird und erst danach den nächsten ADS-Baustein verwenden. Um den Zugriff zeitlich zu optimieren, können Sie auch einen `ADSREADWRITE` Baustein verwenden, der die Daten liest und gleichzeitig schreibt.

8.1.4 SPS-Projekt erstellen

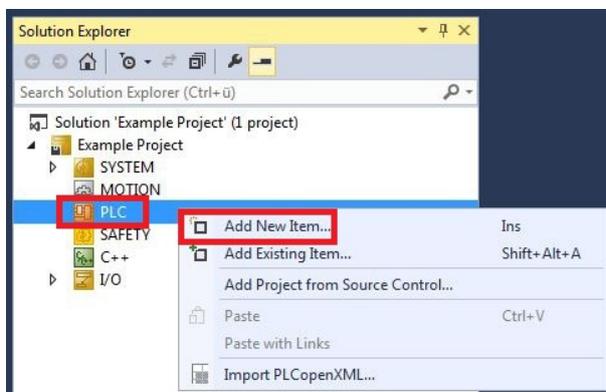
In den nächsten Schritten wird beschrieben, wie Sie ein PLC-Projekt in TwinCAT erstellen und in der Strukturansicht einfügen.

Voraussetzungen für diesen Arbeitsschritt:

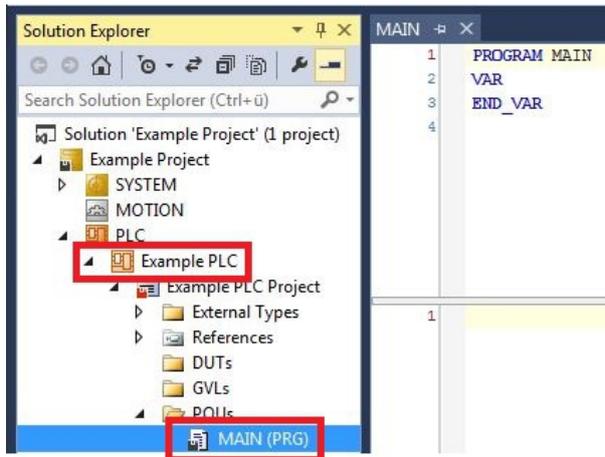
- Ein neu angelegtes TwinCAT XAE Projekt.

Erstellen Sie ein PLC-Projekt wie folgt:

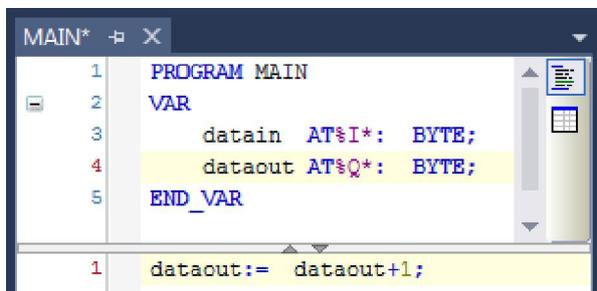
1. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf **PLC**.
2. Klicken Sie im Kontextmenü auf **Add New Item** und wählen Sie das **Standard PLC Project**.



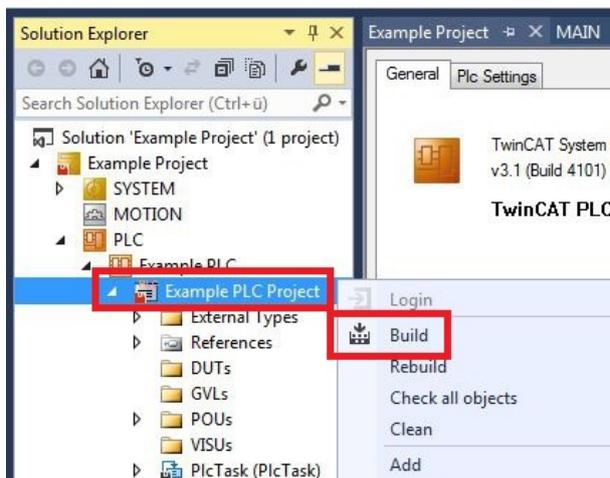
3. Klicken Sie in der Struktursicht auf das neu erstellte PLC-Projekt und dann unter **POUs** doppelt auf **MAIN (PRG)**.



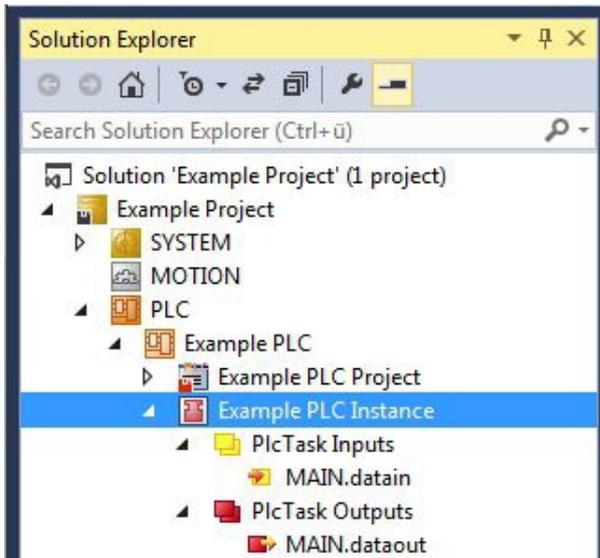
4. Schreiben Sie ein kleines Programm wie im folgenden Bild.



5. Klicken Sie in der Struktursicht 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.

8.1.5 Variablen verknüpfen

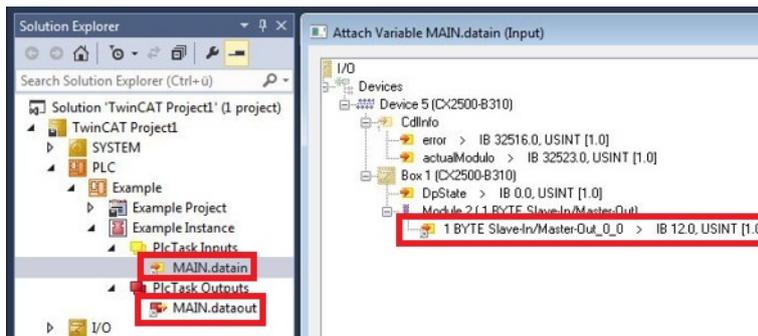
Wurde das PLC-Projekt erfolgreich im System Manager angefügt, können die Sie die neu angelegten Ein- und Ausgangsvariablen aus dem PLC-Projekt mit den Ein- und Ausgängen Ihrer Hardware verknüpfen.

Voraussetzungen für diesen Arbeitsschritt:

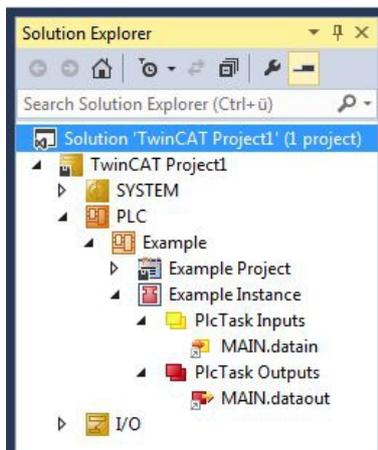
- Ein angefügtes PLC-Programm in TwinCAT.

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.

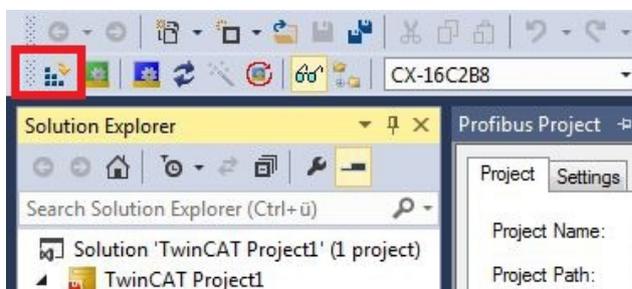


2. Klicken Sie doppelt im Fenster **Attach Variable** auf die Ein bzw. Ausgänge der Hardware. Verknüpfen Sie die Eingangsvariablen mit den Eingängen und die Ausgangsvariablen mit den Ausgängen der Hardware.



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.

8.1.6 Konfiguration auf CX laden

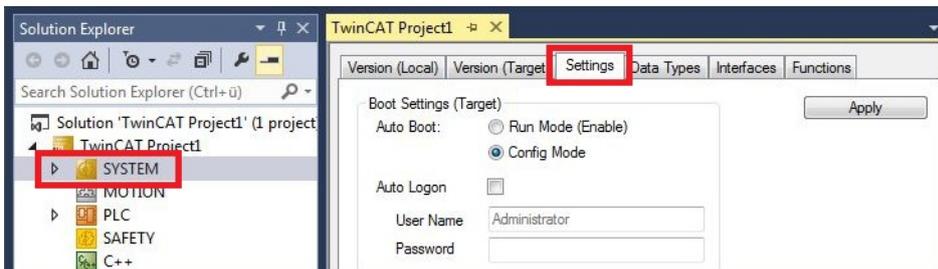
Wenn Variablen verknüpft sind, kann die Konfiguration gespeichert und auf den CX geladen werden. Das hat den Vorteil, dass das PLC-Projekt automatisch geladen und gestartet werden kann, wenn der CX eingeschaltet wird. Der Start des zuvor erstellten PLC-Projekts kann damit automatisiert werden.

Voraussetzungen für diesen Arbeitsschritt:

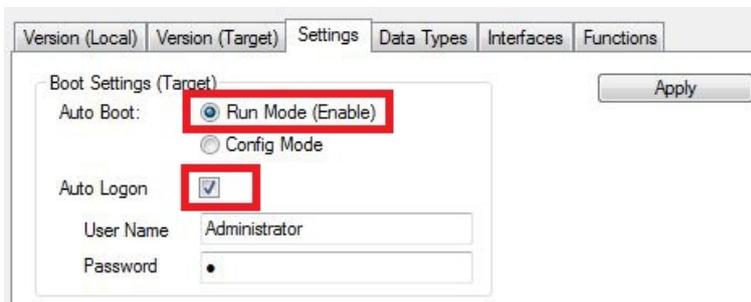
- Ein fertiges und im System Manager angefügtes PLC-Projekt.
- Variablen aus dem PLC-Projekt verknüpft mit der Hardware im System Manager.
- Ein CX ausgewählt als Zielsystem.

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**.

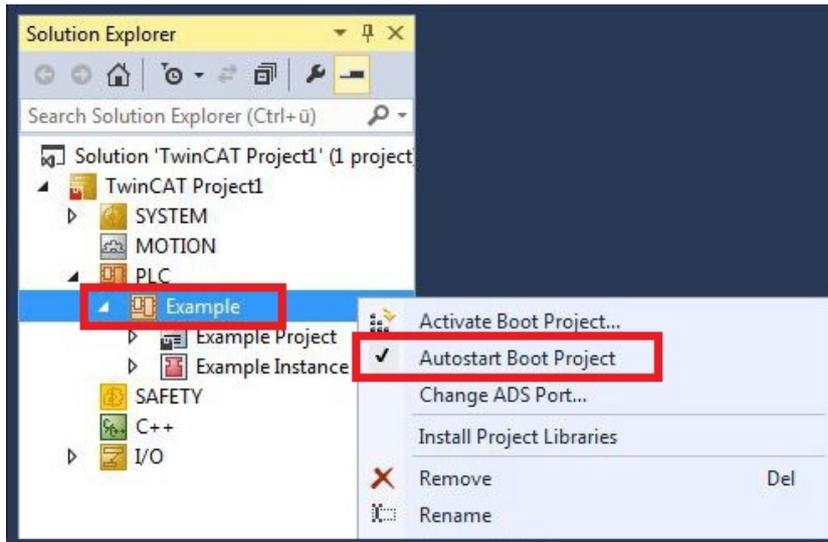


3. Wählen Sie unter Boot Settings die Option **Run Mode (Enable)** und Aktivieren Sie das Kontrollkästchen **Auto Logon**.

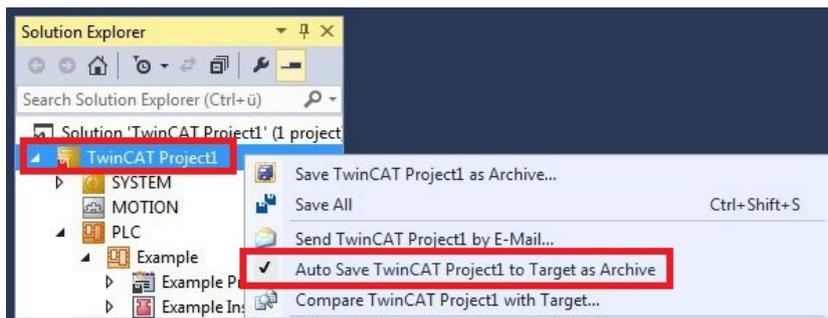


4. Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein.
5. Klicken Sie auf **Apply**.
6. Klicken Sie links in der Strukturansicht unter **PLC** mit der rechten Maustaste auf das PLC-Projekt.

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.

8.2 CX7031 Slave anlegen

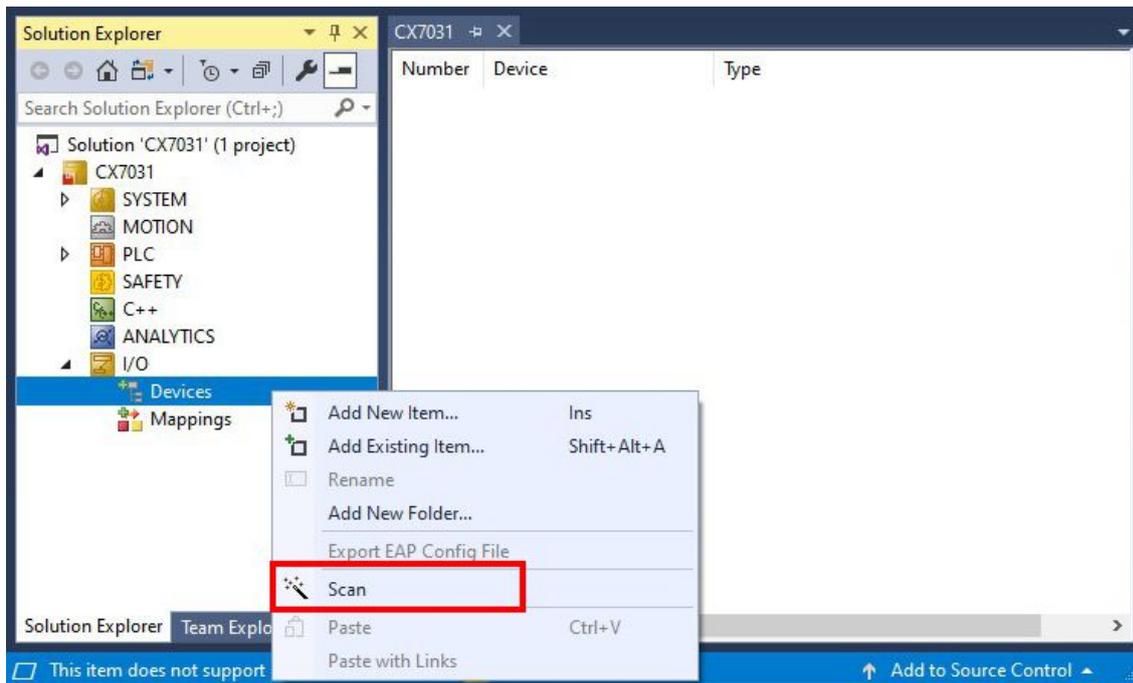
Damit der PROFIBUS-Slave konfiguriert und später vom PROFIBUS-Master mit allen Ein- und Ausgängen erkannt wird, muss der PROFIBUS-Slave zuerst in TwinCAT angelegt werden.

Voraussetzungen:

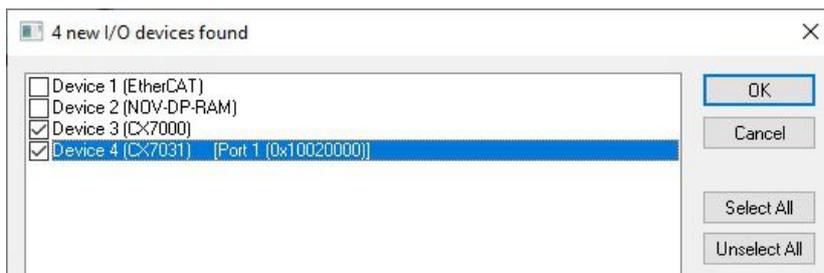
- CX7031 als Zielgerät festgelegt.

Gehen Sie wie folgt vor:

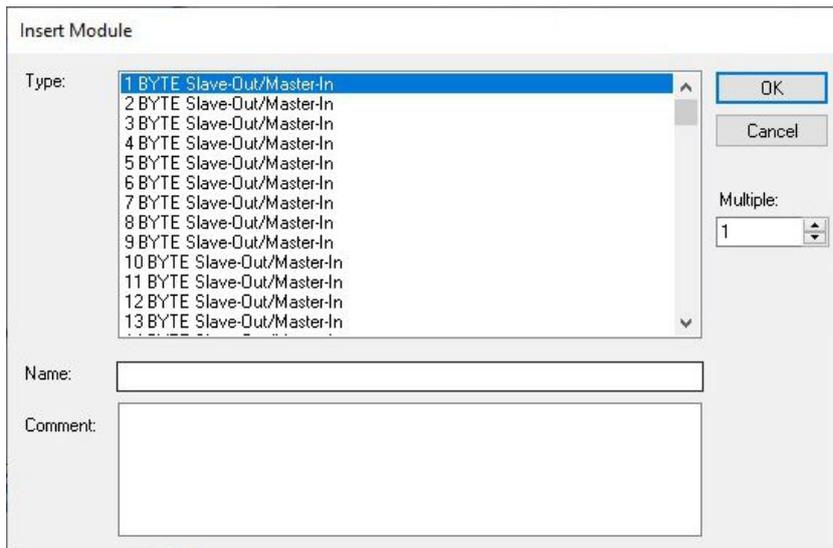
1. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **I/O Devices**.
2. Klicken Sie im Kontextmenü auf **Scan**.



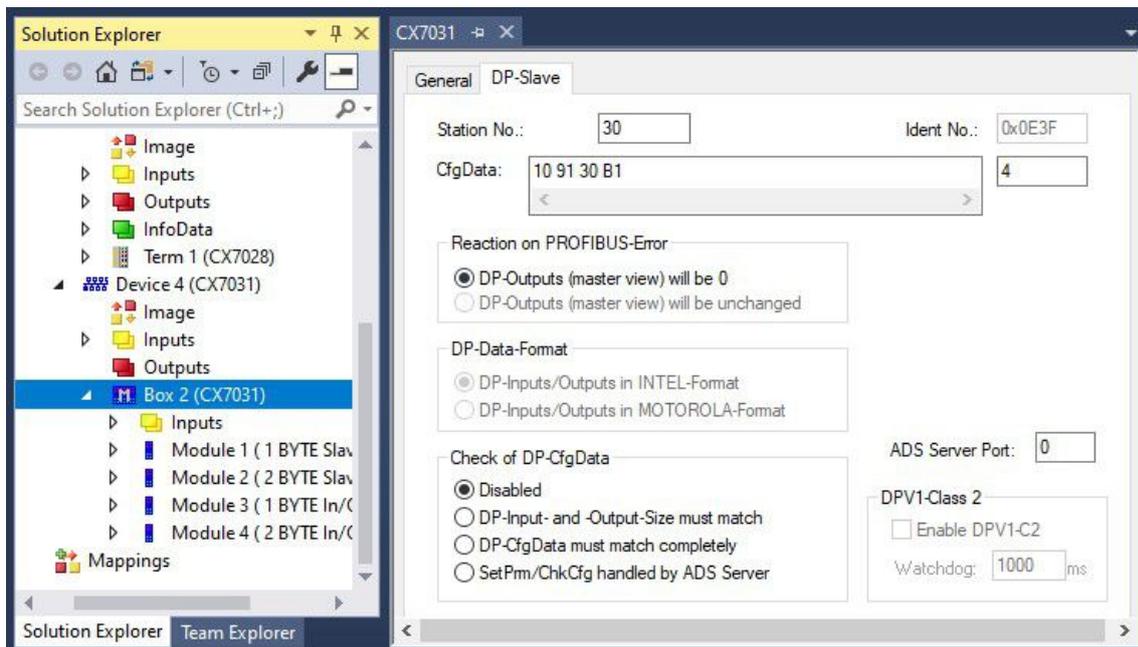
3. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



- Bestätigen Sie die Anfrage mit **Ja**, um nach Boxen zu suchen.
Die Boxen werden eingebunden. Das Fenster **Insert Module** erscheint.



- Fügen Sie Module wie z.B. **1 BYTE Slave-Out/Master-In** und **1 BYTE Slave-In/Master-Out** für Ihr Prozessabbild ein.
 - Klicken Sie auf **Cancel**, um das Fenster **Insert Module** zu schließen.
 - Klicken Sie bei der Anfrage, ob FreeRun aktiviert werden soll, auf **Ja**.
- ⇒ Der PROFIBUS-Slave wird in der Strukturansicht mit den Ein- und Ausgängen angezeigt. Sie können weitere Module 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.

8.2.1 PROFIBUS-Adresse einstellen

Nachdem der PROFIBUS-Slave erfolgreich in TwinCAT angefügt wurde, kann die PROFIBUS-Adresse des Slaves eingestellt werden. In diesem Arbeitsschritt wird die Adresse in TwinCAT eingestellt, damit der PROFIBUS-Slave über diese Adresse für den PROFIBUS-Master erreichbar ist.

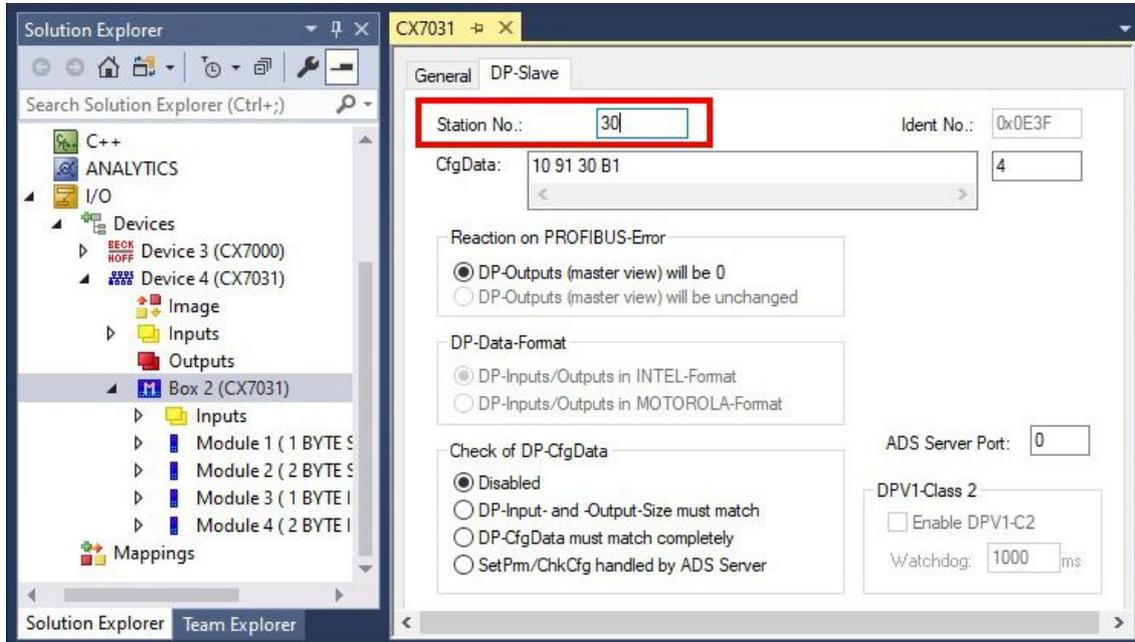
Mögliche Adressen sind 1 bis 125, wobei die niedrigen Adressen für den Master bestimmt sind. Die Adressen 0 und 127 sind reserviert.

Voraussetzungen:

- Ein eingefügter PROFIBUS-Slave in TwinCAT.

Gehen Sie wie folgt vor:

1. Klicken Sie auf eine Slave-Box und anschließend auf die Registerkarte **DP-Slave**.
2. Tippen Sie im Feld **Station No** einen Wert für die PROFIBUS-Adresse ein, z.B. „30“.



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

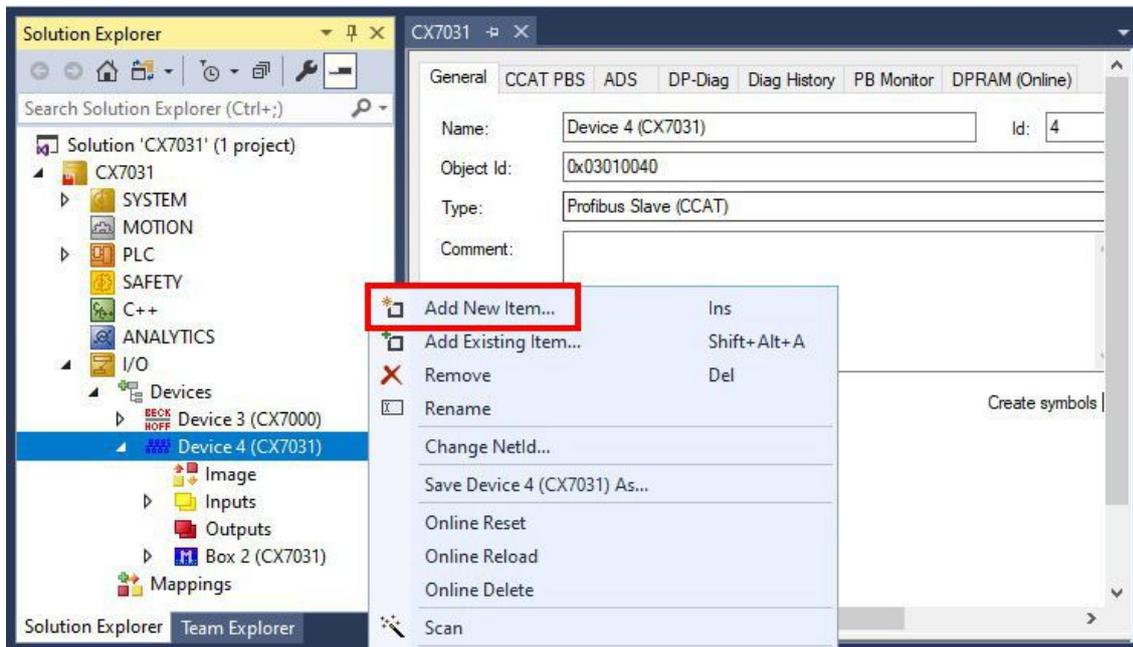
8.2.2 Virtuellen Slave anlegen

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

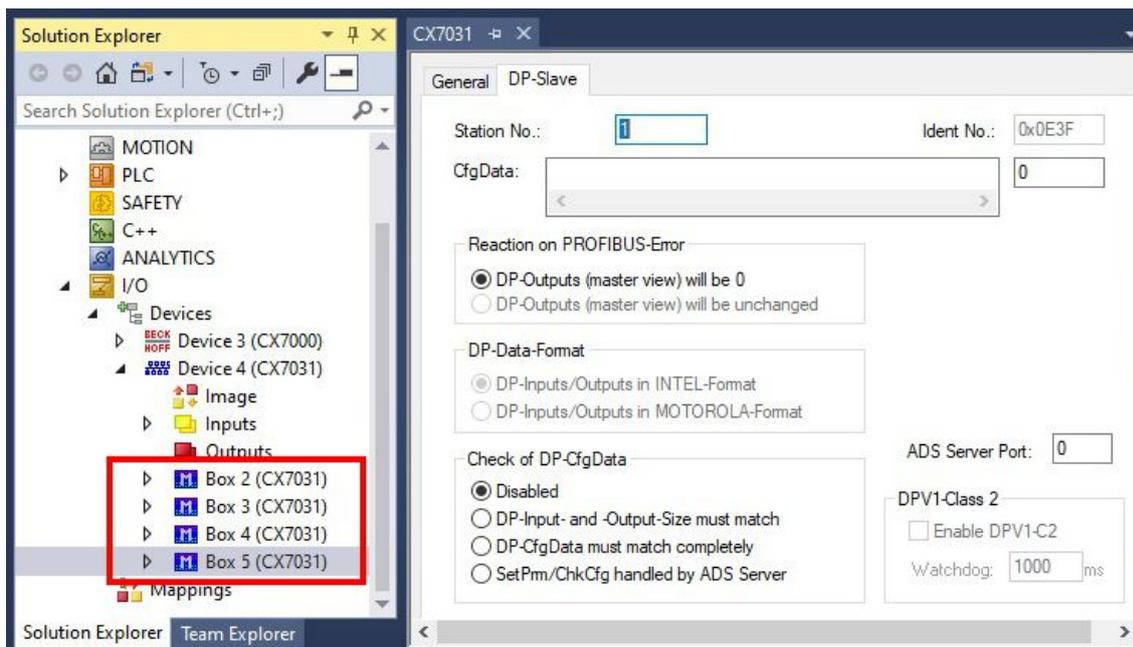
Es können maximal vier virtuelle Slaves angelegt werden. Pro Slave können maximal 240 Byte Daten konfiguriert werden, d. h. es können insgesamt viermal 240 Byte Daten in jede Richtung ausgetauscht werden. Jeder virtuelle Slave erhält über TwinCAT eine eigene Adresse und wird für den PROFIBUS-Master wie ein eigenständiges Gerät konfiguriert.

Gehen Sie wie folgt vor:

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. Es können bis zu vier virtuelle Slaves angelegt werden.



Für die virtuellen Slaves können jetzt eigene Variablen angelegt und die PROFIBUS-Adressen eingestellt werden.

8.2.3 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.

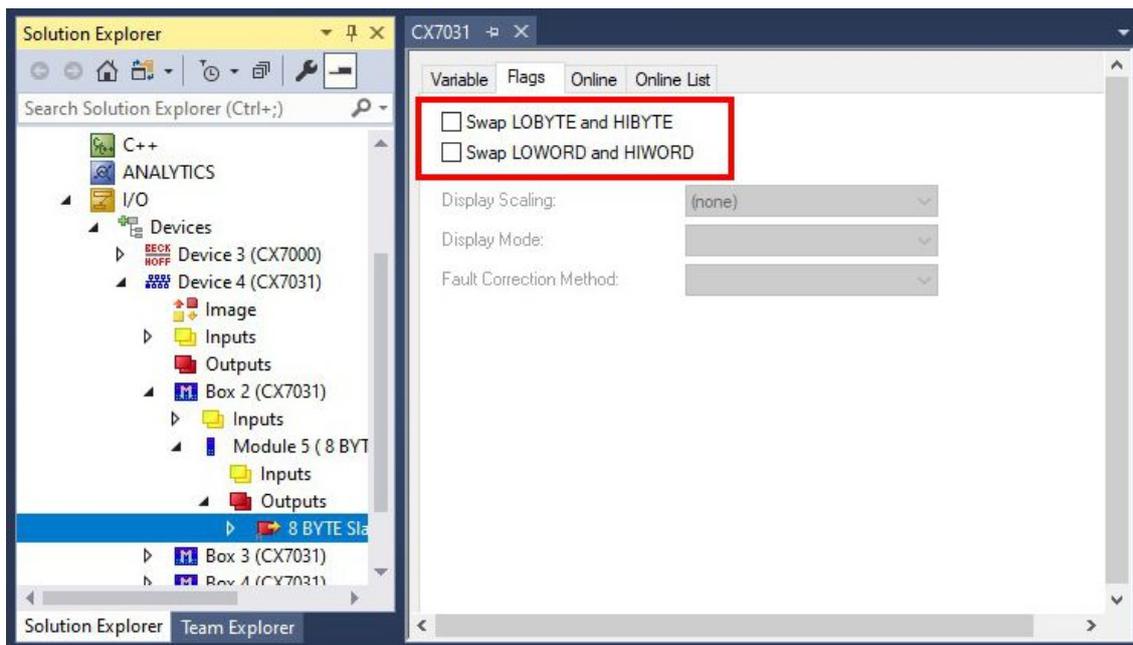
Wenn Sie das Standardformat benötigen, können Sie diesen Arbeitsschritt überspringen.

Voraussetzungen für diesen Arbeitsschritt:

- Ein fertig parametrierter Slave.

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 LOWORD und HIWORD.



⇒ 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 SPS-Projekt mit dem Befehl ROR gedreht werden.
Beispiel für ST: VarProfibus:=ROR(VarAnalog,8); (*Beide Variablen vom Typ WORD*)

8.2.4 DPV1-Daten empfangen

DPV1-Dienste sind neben der zyklischen Verbindung der azyklische Datenaustausch. Dieser wird immer vom PROFIBUS-Master initiiert und kann Daten über diesen Kanal lesen oder schreiben. Dieser Dienst ist sinnvoll, um Daten zu transportieren, die nur selten benötigt werden oder sich selten ändern. In der Regel sind dies z. B. Geräte, die zur Laufzeit parametrieren können, diese Parameter kann der Anwender dann lesen oder schreiben.

Beim CX7031 steht dieser Kanal offen zur Verfügung und kann somit vom Anwender parallel zu den zyklischen Daten genutzt werden.

Der Empfang erfolgt beim CX7031 über die ADS-Notification-Bausteine in der SPS. Wird ein DPV1-Telegramm empfangen, so wird dieses in die ADS-Notification eingetragen und an die SPS weitergeleitet und kann dann mit den entsprechenden Bausteinen z. B. ADSWRITEIND bei einem Schreibtelegramm ausgewertet werden. Der Anwender muss auch die Antwort generieren und kann auch bei fehlerhaften DPV1-Daten einen entsprechenden Fehler zurückgeben, z. B., dass die Datenlänge nicht stimmt oder der DPV1-Index falsch ist.

Der Ablauf ist der Aufruf des Bausteins ADSWRITEIND, ist der VALID gesetzt, wurden Daten empfangen, da evtl. auch andere Dienste die ADS-Indication verwenden, empfiehlt es sich die AMS NetID zu prüfen, woher die ADS-Indication kommt. Im Falle des CX7031 ist dies die AMS NetID des CX7031 Device. Dann wird der Port überprüft, da im CX7031 evtl. 4 Slaves vorhanden sind, kann über die Portnummer festgestellt werden, von welchem Slave der DPV1-Dienst empfangen wurde. Der Port ist immer 1000 + Slave-Adresse. Man muss also einfach den Wert des Ports minus 1000 berechnen und erhält die PROFIBUS-Adresse des Slaves. Als nächstes kommt der IDXGRP, hier ist immer das Bit 31 gesetzt, zusätzlich ist hier die DPV1 Slot Nummer codiert. Der IDXOFFS enthält die DPV1 Indexnummer. Ist der DPV1-Dienst korrekt, so quittieren sie anschließend mit der entsprechenden ADS-Response und dem Ergebnis = 0. Ist der DPV1 ungültig, so muss die Response <> 0 sein und wird je nach Fehler beantwortet, hier sind die Fehlercodes der PROFIBUS-Norm zu beachten, diese sind in der folgenden Tabelle DP- V1 Error Response beschrieben.

Um die Weiterleitung an die SPS zu ermöglichen, muss in TwinCAT der Port der SPS angegeben werden, dies ist in der Regel „851“. Der Standardwert ist „0“, d. h. dieser Dienst ist deaktiviert.

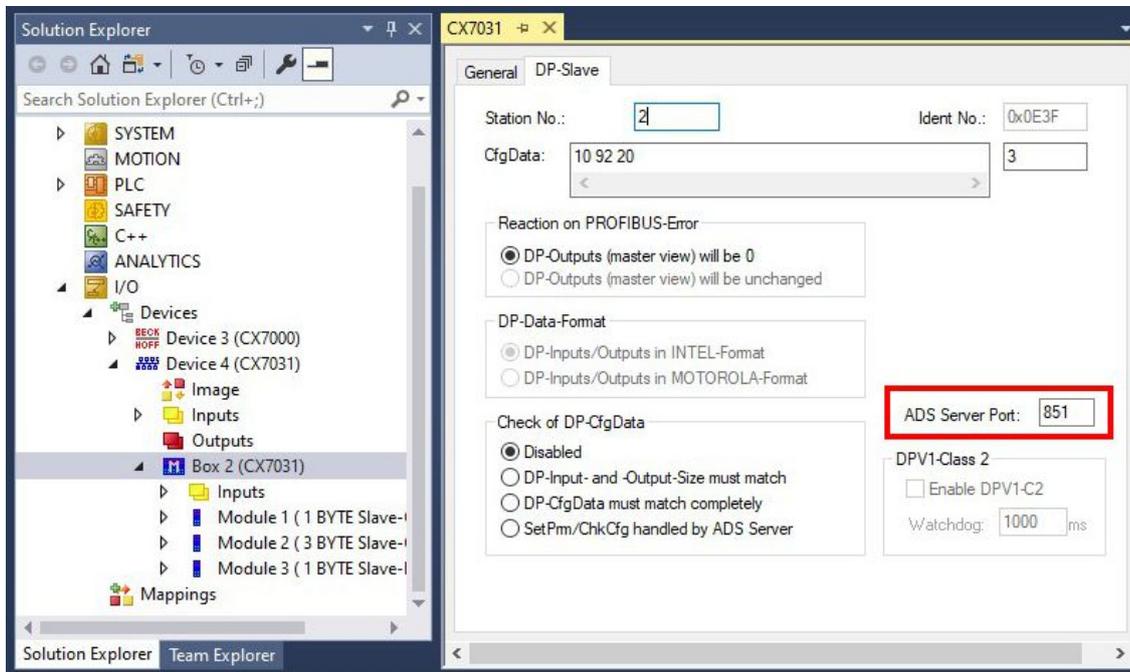


Abb. 28: DPV1-Daten empfangen: Aktivierung der Kommunikation unter TwinCAT.

DPV1-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 ADS des Devices)

ADS-Read-Indication-Parameter	Bedeutung
Source-Port (PORT)	1000+Slave-Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, muss für den Response verwendet werden
IndexGroup (IDXGRP)	Slot-Number (DPV1-Parameter) (0x01...0xFC)
IndexOffset (IDXOFFS)	Index (DPV1-Parameter) (0x00...0xFF)
Length (LENGTH)	Länge der auszulesenden Daten (max. 236 Byte)

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 ADS des Devices)
Destination-Port (PORT)	1000+Slave-Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, wie bei Indication
Result (RESULT)	Siehe DPV1-Fehler-Response
Length (LENGTH)	Länge der gelesenen Daten
Data (DATAADDR)	gelesene Daten

DPV1-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 ADS des Devices)
Source-Port (PORT)	1000+Slave-Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, muss für den Response verwendet werden
IndexGroup (IDXGRP)	Slot-Number (DPV1-Parameter) (0x01...0xFC)
IndexOffset (IDXOFFS)	Index (DPV1-Parameter) (0x00...0xFF)
Length (LENGTH)	Länge der zu schreibenden Daten (max. 240 Byte)
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 ADS des Devices)
Destination-Port (PORT)	1000+Slave-Adresse
Invoke-ID (INVOKEID)	eindeutige Nummer, wie bei Indication
Result (RESULT)	Siehe DPV1-Fehler-Response
Length (LENGTH)	Länge der gelesenen Daten

DPV1-Fehler-Response:

Response: = 16#aaaa_yyxz (yy Fehler Code 2 Herstellerspezifisch, x Error Class, z Error Code)
 „aaaa“ optional, ADS-Fehlercode, wir empfehlen hier den Fehlercode 0x0700 zu verwenden.

Error-Class (x= Bit 4-7)	Error-Code (z=Bit 0-3)	Bedeutung
0xA	0x0	Application, Read Error
	0x1	Application, Write Error
	0x2	Application, Module Failure
	0x8	Application, Version Conflict
	0x9	Application, Feature Not Supported
0xB	0x0	Access, Invalid Index
	0x1	Access, Write Length Error
	0x2	Access, Invalid Slot
	0x3	Access, Type Conflict

Error-Class (x= Bit 4-7)	Error-Code (z=Bit 0-3)	Bedeutung
	0x4	Access, Invalid Area
	0x5	Access, State Conflict
	0x6	Access, Access Denied
	0x7	Access, Invalid Range
	0x8	Access, Invalid Parameter
	0x9	Access, Invalid Type
0xC	0x0	Resource, Read Constrain Conflict
	0x1	Resource, Write Constrain Conflict
	0x2	Resource, Busy
	0x3	Resource, Unavailable

Beispiel sie wollen mitteilen das der DPV1-Index falsch ist so antworten sie im Response mit einer 0x0000_00B0.

Code-Beispiel:

```

VAR_GLOBAL
  CX7031_NetID AT %I*:AMSNETID ; //Link to the AMS Net ID from the CX7031 PROFIBUS Interface
END_VAR

PROGRAM pro_DPv1_Service
VAR
  i : INT; //Count up if something is coming in the Write Indication
  sNetId : STRING;
  nPort : UINT;
  udInvokeId : UDINT;
  iCase : INT;
  udGroup : UDINT;
  udOffset : UDINT;
  udLenght : UDINT;
  ptrByte : POINTER TO BYTE; (* Datenpointer *)

  fbAdsWriteInd : ADSWRITEIND;
  fbAdsWriteRes : ADSWRITERES;
  bExecute : BOOL;

  result : UDINT; //Bit 0..7 Error Code 1: Class/Code, Bit 8..15 Error Code 2, Bit 16..31
not necessary
  nResult : UDINT;
  DPV1_Write : ARRAY[0..255] OF BYTE;
  ADSREADINDEX : ADSREADINDEX;
  ADSREADRES : ADSREADRES;
  sNetId_READ : STRING(23);
  nPort_READ : UINT;
  udInvokeId_READ : UDINT;
  udGroup_READ : DWORD;
  udOffset_READ : UDINT;
  udLenght_READ : UDINT;
  ptrByte_READ : Tc2_System.UXINT;
  DATAADDR : INT;
  DataRead : ARRAY[0..199] OF BYTE;
  iRead : INT;
END_VAR

//Read a DPV1 Write here we receive DV1 write data
fbAdsWriteInd(
  CLEAR :=,
  VALID =>,
  NETID =>,
  PORT =>,
  INVOKEID =>,
  IDXGRP =>,
  IDXOFFS =>,
  LENGTH =>,
  DATAADDR =>);

IF fbAdsWriteInd.VALID THEN //wait until valid is True then we receive data
  (* You have to check the NETID and port number if you use ADS notification also for other things *)
  i := i + 1; //Count up if something is coming in the Indication
  sNetId := fbAdsWriteInd.NETID; //AMD Net ID from device (CX7031)
  nPort := fbAdsWriteInd.PORT; //PB Adresse + 1000 dec

```

```

udInvokeId := fbAdsWriteInd.INVOKEID; //for the Response
udGroup := UDINT_TO_DWORD(fbAdsWriteInd.IDXGRP);
udOffset := fbAdsWriteInd.IDXOFFS; //Prm or Cfg Data or DPV1 Services
udLenght := fbAdsWriteInd.LENGTH; //Len of the data
ptrByte := fbAdsWriteInd.DATAADDR; //Pointer from the data
//Result without Error, if you want so send an Error use "1"
IF F_CreateAmsNetId(nIds:=gvl.CX7031_NetID )=sNetId THEN //Check the AMSNet ID from CX7031
-> IF sNetId='5.23.4.5.4.1' THEN ....
    IF nPort=1000 + 22 THEN //Slave with Adress 22
        CASE udOffset OF //Offset which kind of data is coming
            20: //For DP-V1 Services (20 is here the PB Index)
                MEMCPY(ADR(DPV1_Write),ptrByte,udLenght);
                fbAdsWriteRes.RESULT := 0;
            ELSE
                fbAdsWriteRes.RESULT := 16#0700_00B0 ; //Access, Invalid Index;
        END_CASE
    END_IF

    IF iCase = 0 THEN
        iCase := 10; //Send response
    END_IF
END_IF

CASE iCase OF
    0: (* WAIT *);
    10: (* send ADS-Response *)
        fbAdsWriteRes(
            NETID := sNetId,
            PORT := nPort,
            INVOKEID := udInvokeId,
            RESULT := ,
            RESPOND := TRUE);
        fbAdsWriteInd(CLEAR:=TRUE); (* confirm ADS-Indication *)
        iCase := 20;
    20: fbAdsWriteRes(RESPOND:=FALSE);
        fbAdsWriteInd(CLEAR:=FALSE);
        iCase := 0;
END_CASE

//READ Indication for DPV1 Read Data
ADSREADINDEX(
    CLEAR:= ,
    MINIDXGRP:= ,
    VALID=> ,
    NETID=> ,
    PORT=> ,
    INVOKEID=> ,
    IDXGRP=> ,
    IDXOFFS=> ,
    LENGTH=> );

IF ADSREADINDEX.VALID THEN
    sNetId_READ :=ADSREADINDEX.NETID; //AMD Net ID from device (CX7031)
    nPort_READ :=ADSREADINDEX.PORT; //PB Adresse + 1000 dec
    udInvokeId_READ :=ADSREADINDEX.INVOKEID; //for the Response
    udGroup_READ :=UDINT_TO_DWORD(ADSREADINDEX.IDXGRP);
    udOffset_READ :=ADSREADINDEX.IDXOFFS; //Prm or Cfg Data
    udLenght_READ :=ADSREADINDEX.LENGTH; //Len of the data
    iRead:=10;
    IF F_CreateAmsNetId(nIds:=gvl.CX7031_NetID )=sNetId_READ THEN //Check the AMSNet ID from
CX7031 -> IF sNetId='5.23.4.5.4.1' THEN ....
        IF nPort_READ =1022 THEN //Check Slave Addr 22
            IF (16#FF AND udGroup_READ) = 2 THEN //Check Slot Number
                IF udOffset_READ=16#15 THEN //Check Index Number, here is the Index 21=0x15
                    DataRead[0]:=DPV1_Write[0]+1;
                    ADSREADRES.RESULT:=0; //no Error
                ELSE
                    ADSREADRES.RESULT :=16#0700_00B0 ; //Access, Invalid Index
                END_IF
            ELSE
                ADSREADRES.RESULT :=16#0700_00B2 ; //Access, Invalid Slot
            END_IF
        END_IF
    END_IF
END_IF

CASE iRead OF
    0: //Wait
    10:

```

```

ADSREADRES (
  NETID:=sNetId_READ ,
  PORT:=nPort_READ ,
  INVOKEID:=udInvokeId_READ ,
  RESULT:= ,
  LEN:=20 ,
  DATAADDR:=ADR (DataRead) ,
  RESPOND:=TRUE );
ADSREADINDEX (Clear:=TRUE);
iRead:=20;
20: ADSREADRES (RESPOND:=FALSE);
ADSREADINDEX (Clear:=FALSE);
iRead:=0;
END_CASE

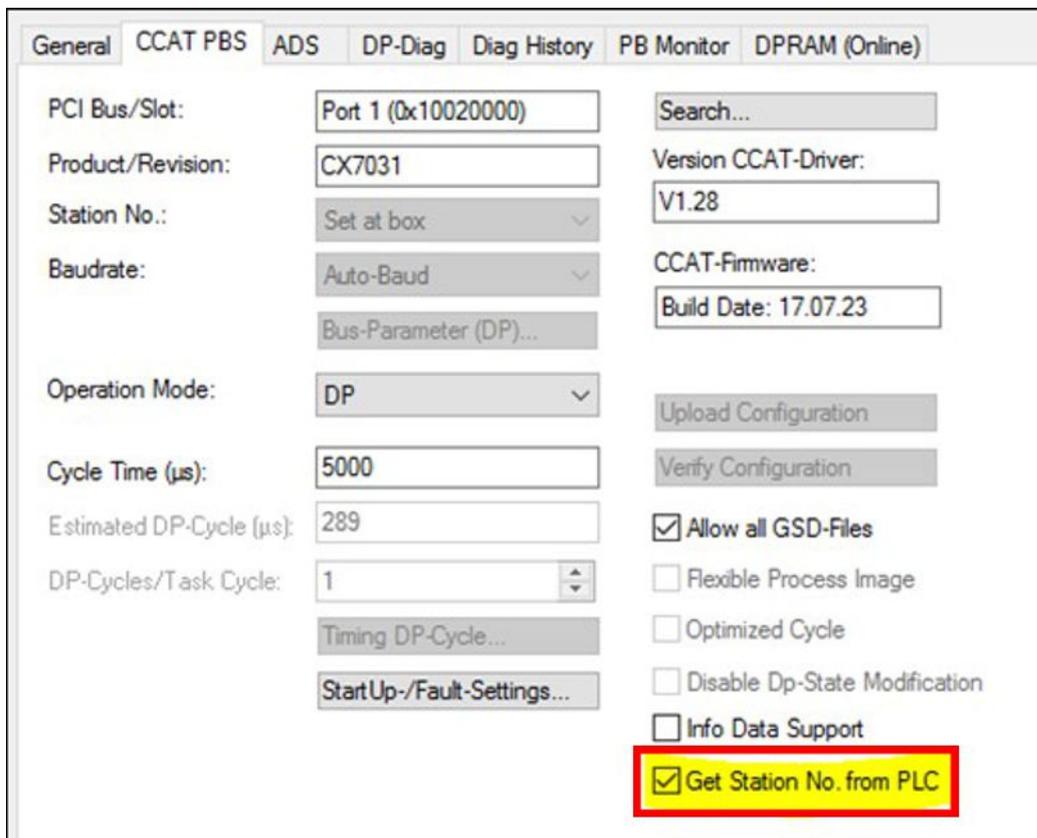
```

8.2.5 PROFIBUS-Adresse über die SPS setzen

Die PROFIBUS-Adresse dient dazu, den Geräten in einem PROFIBUS-Netzwerk eine eindeutige Identifikation zuzuweisen, um die Kommunikation zwischen der Steuerung und den angeschlossenen Geräten zu ermöglichen. Neben den Einstellmöglichkeiten in TwinCAT kann die PROFIBUS-Adresse auch über die SPS eingestellt werden.

Gehen Sie wie folgt vor:

1. Aktivieren Sie in TwinCAT die Option **Get Station No. from PLC** und aktivieren Sie das Projekt.



2. Der CX7031 startet zunächst ohne PROFIBUS-Adresse und es findet keine PROFIBUS-Kommunikation statt.
3. Weisen Sie dem PROFIBUS-Slave mit Hilfe des Funktionsbausteins ADSWRITE eine PROFIBUS-Adresse zu.

```

AMS Net ID //from CX7031 device
Port := 200
IDXGRP:= 16#F480
IDXOFFS:= 0..3 //if you have configured up to 4 slaves)
LEN:=1

```

4. Das erste Byte Daten enthält dann die PROFIBUS-Adresse des Slaves. Achten Sie beim Arbeiten mit mehreren Slaves am CX7031 darauf, dass beim Schreiben einer neuen PROFIBUS-Adresse, die bereits in Kommunikation befindlichen PROFIBUS-Slaves kurzzeitig aus dem Datenaustausch gehen.

- ⇒ Die Adresse wird im CX7031 gespeichert, d.h. bei einem TwinCAT-Restart oder Power-ON bleiben die geschriebenen Adressen für die Slaves gültig. Ein Zurücklesen über den Funktionsbaustein ADSREAD ist nicht möglich.

Bei Verwendung mehrerer Slaves im CX7031 kann immer nur ein Slave aktiv oder inaktiv geschaltet werden. Sollen z. B. alle maximal möglichen 4 Slaves eine PROFIBUS-Adresse aus der SPS erhalten, so muss dies nacheinander geschehen und kann nicht gleichzeitig erfolgen. Weiterhin ist zu beachten, dass jede Zustandsänderung zur Folge hat, dass die anderen Teilnehmer kurzzeitig den Datenaustausch verlassen, technisch ist dies nicht anders lösbar. Auch wenn Sie einen der Slaves deaktivieren, der sich im Datenaustausch befindet, dann fallen auch die anderen Slaves kurz aus, bauen aber sofort wieder die Verbindung auf. Dies müssen sie in ihrer Applikation berücksichtigen. Wenn ihre Anwendung damit nicht umgehen kann, ist das Feature für sie nicht nutzbar.

Weitere Einsatzszenarien

Die Beschreibung weiterer Anwendungsszenarien ist als Anregung gedacht und erhebt keinen Anspruch auf Vollständigkeit.

1. Hot-Connect nutzen:

Dieses Feature kann dazu verwendet werden, Slaves zu aktivieren, wenn der CX7031 z. B. über EtherCAT eine bestimmte Konfiguration erkennt. Solch eine Konfiguration könnte beispielsweise eine Hot-Connect-Gruppe sein.

Wird eine bestimmte Hot-Connect-Gruppe erkannt, wird die entsprechende PROFIBUS-Adresse gesetzt und der Slave aktiviert. Ist die Hot-Connect-Gruppe nicht vorhanden, wird die PROFIBUS-Adresse auf „0“ gesetzt. Dies kann auch im laufenden Betrieb geschehen, wenn die Hot-Connect-Gruppe über EtherCAT angeschlossen ist. Da bis zu vier Slaves angelegt werden können, wird empfohlen, maximal drei Hot-Connect-Gruppen zu verwenden.

2. Diagnosedaten verschicken:

Die PROFIBUS-Diagnosedaten können über ADSWRITE gesendet werden. Die herstellerspezifische Diagnose beginnt ab dem 6. Byte mit der Länge der folgenden Diagnose-Daten. Das erste Byte der Diagnose-Daten muss immer 0x08 sein, damit herstellerspezifische Diagnose gesendet wird, die Bytes 2...6 werden vom Treiber gesetzt und können von Ihrer Applikation nicht verändert werden.

Beispiel für 10 Byte Diagnosedaten für Slave Nr. 2:

```
DiagArray[0]:=16#08;           // Must be set to trigger manufacturer-specific diagnostics
DiagArray[1..5]:=16#00;       // Described by the CX
DiagArray[6]:=10;             // 10 bytes of diagnostic data including length information
DiagArray[7..15]:=Value;     // Your diagnostic data
ADS Write:
AMD Net ID := CX7031 Interface
Port := 1002;                 // 1000dec + Slave Address
IDXGRP:= 16#F481;
IDXOFFS:= 0;
LEN:=16;                      // 6 bytes + manufacturer-specific diagnostics
```

Wenn Sie zusätzlich Alarm- oder Fehlerbausteine Ihrer übergeordneten Steuerung verwenden wollen, ist dies ein weiteres Einsatzszenario, das Sie nutzen können. Solche Alarm- oder Fehlerbausteine werden in der Master-Steuerung in der Regel anders verarbeitet und aufgerufen als die Standard IO-Daten.

Wenn z. B. der CX einen Fehler hat und der EtherCAT-Master ein Problem erkannt hat, dann laufen die normalen IO-Daten für den PROFIBUS zunächst unabhängig davon weiter. Um nun dem PROFIBUS-Master anzuzeigen, dass der Slave einen Fehler erkannt hat, kann der Slave eine Diagnosemeldung senden. Der Vorteil einer Diagnosemeldung ist, dass sie ereignisgesteuert abläuft und je nach PROFIBUS-Master auch in speziellen Bausteinen verarbeitet und/oder auch optisch angezeigt wird.

Zusätzlich können über die Diagnose Min- und Maxwerte abgesetzt werden, z.B. bei Über- oder Unterschreiten eines Füllstandes. Gerade bei verfahrenstechnischen Anlagen werden diese Diagnosedaten in speziellen Fehlerloggern aufgezeichnet und gespeichert.

8.2.6 PROFIBUS-Geräte mit CX7031 simulieren

Der CX7031 kann bis zu vier PROFIBUS-Geräte simulieren. Dabei ist zu beachten, dass der CX7031 lediglich die GSD-Datei simulieren kann, jedoch nicht die eigentliche Funktion oder Logik. Auch Timings sind nicht Bestandteil der Simulation. Daher muss der Anwender selbst dafür sorgen und ist auch selbst für die Applikation verantwortlich. Die Anzahl der Prozessdaten sowie der Parameter- und Konfigurationsdaten darf insgesamt 240 Byte nicht überschreiten.

Die Parameter- und Konfigurationsdaten werden immer fehlerfrei bestätigt. Es ist jedoch möglich, diese beiden Datenpakete zur Auswertung und gegebenenfalls zur Veränderung der Prozessdaten oder Funktion in die SPS umzuleiten.

Gehen Sie wie folgt vor:

1. Fügen sie die GSD-Datei, die sie Simulieren möchten, in den Ordner `\TwinCAT\3.1\Config\Io\Profibus` ein.
2. Aktivieren Sie die Option **Allow all GSD-Files**, um die diese Funktion zu aktivieren.

The screenshot shows the configuration window for the CCAT PBS driver. The 'General' tab is selected. The interface includes various input fields and checkboxes. The 'Allow all GSD-Files' checkbox is checked and highlighted with a red box. Other visible settings include: PCI Bus/Slot: Port 1 (0x10020000), Product/Revision: CX7031, Station No.: Set at box, Baudrate: Auto-Baud, Operation Mode: DP, Cycle Time (µs): 10000, Estimated DP-Cycle (µs): 221, DP-Cycles/Task Cycle: 1. On the right side, there are checkboxes for 'Flexible Process Image', 'Optimized Cycle', 'Disable Dp-State Modification', 'Info Data Support', and 'Get Station No. from PLC'. Buttons for 'Upload Configuration' and 'Verify Configuration' are also present.

3. Fügen Sie das zu simulierende PROFIBUS-Gerät in den Strukturbaum ein. Die Eingangsdaten werden dann als Ausgangsdaten betrachtet und umgekehrt, da sich die Datenrichtung ändert. Die GSD-Datei zeigt die Daten aus Sicht des Masters. Da die GSD-Datei jedoch aus Sicht des Slaves betrachtet wird, ist die Datenrichtung umgekehrt.

Beispiel: Die GSD-Datei enthält eine INT-Variable, die den Temperaturwert eines PROFIBUS-Slaves anzeigt. In der Simulation auf dem CX7031 wird dieser Wert über eine EL3312 ermittelt, die als Eingang an die Klemme angeschlossen ist und als Ausgang an das PROFIBUS-Slave-Interface weitergeleitet wird. Auf der PROFIBUS-Master-Seite erscheint der Wert dann wieder als Eingang.



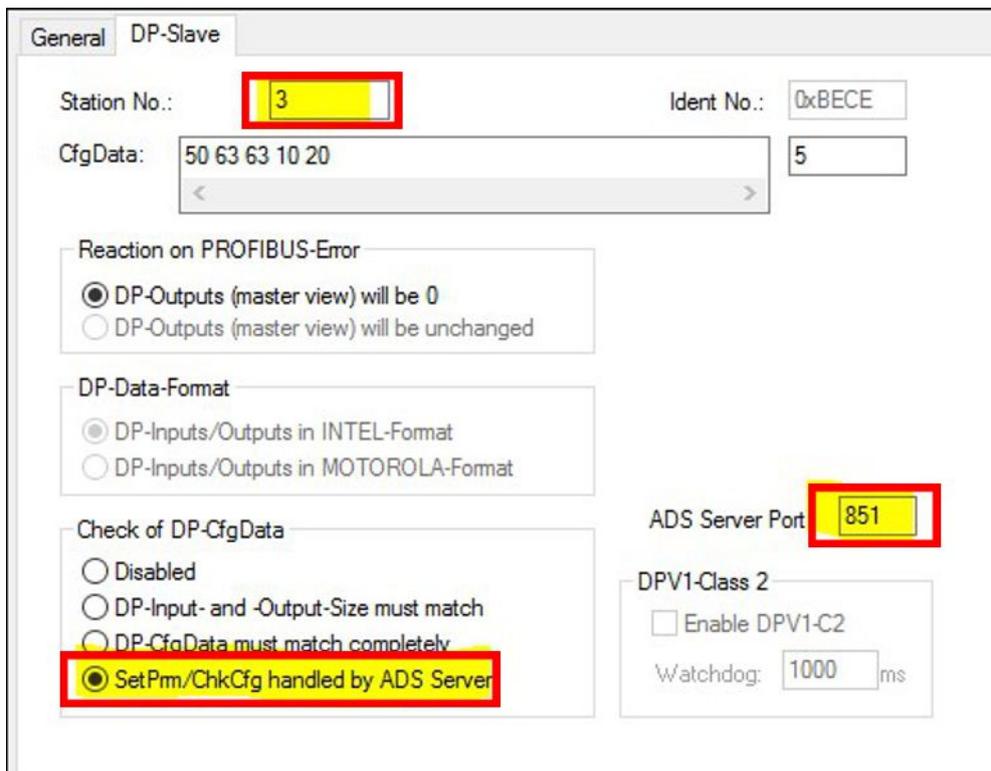
Das Bild zeigt einen simulierten BK3100 mit einer analogen Eingangsklemme. Um die Daten für den Eingang (aus Sicht des Masters) zu verknüpfen, ist ein analoger Ausgang erforderlich.

8.2.7 Parameter- und Konfigurationsdaten an die SPS weiterleiten

Um die Parameter- und Konfigurationsdaten an die SPS übertragen zu können, müssen diese im Slave an den ADS-Server gesendet werden. Die entsprechende Option wird über TwinCAT aktiviert.

Gehen Sie wie folgt vor:

1. Wählen Sie links im Strukturbaum den PROFIBUS-Slave aus,
2. Aktivieren Sie die Option **SetPrm/ChkCfg handled by ADS Server..**



3. Verwenden Sie den Port 851 für die erste SPS-Task.
 - ⇒ Die Daten werden nun über ADS-Indication an die SPS gesendet. In der SPS müssen die Daten angenommen und quittiert werden.

Daten in der SPS annehmen und quittieren

Zunächst sollten Sie überprüfen, woher die Daten stammen. Der CX7031 unterstützt bis zu 4 Slaves, die in der Portnummer des Slaves codiert sind, nämlich in der PROFIBUS-Adresse + 0x1000hex.

Das Telegramm ist im IdxOffset codiert:

- Parameterdaten: 16#03230020
- Konfigurationsdaten: 16#03230021

Beim Starten des Masters erhalten Sie immer zwei Indikationen, die Sie beide mit `Result=0` quittieren müssen, ansonsten wird ein Fehler an den entsprechenden PROFIBUS-Master gesendet und die Verbindung nicht aufgebaut.

Code-Beispiel:

```

VAR
  i                : INT;
  sNetId           : STRING;
  nPort            : UINT;
  udInvokeId       : UDINT;
  iCase            : INT;
  udGroup          : UDINT;
  udOffset         : UDINT;
  udLenght         : UDINT;
  ptrByte          : POINTER TO BYTE; (* Datenpointer *)

  strPrmData2      : ARRAY[0..127] OF BYTE;
  strCfgData2      : ARRAY[0..127] OF BYTE;
  strPrmData3      : ARRAY[0..127] OF BYTE;
  strCfgData3      : ARRAY[0..127] OF BYTE;
  fbAdsWriteInd    : ADSWRITEIND;
  fbAdsWriteRes    : ADSWRITERES;
  bExecute         : BOOL;
  stAdsInfo AT %I* : ST_AmsAddr;
  nState          AT %I* : WORD;
  UnKonwnoffset   : UDINT;
END_VAR

fbAdsWriteInd(
  CLEAR := ,
  VALID => ,
  NETID => ,
  PORT => ,
  INVOKEID => ,
  IDXGRP => ,
  IDXOFFS => ,
  LENGTH => ,
  DATAADDR => );

IF fbAdsWriteInd.VALID THEN
  (* You have to check the NETID and port number if you use ADS notification also for other things *)
  i := i + 1;
  sNetId := fbAdsWriteInd.NETID;           // AMD Net ID from device (CX7031)
  nPort := fbAdsWriteInd.PORT;             // PB Adresse + 1000 dec
  udInvokeId := fbAdsWriteInd.INVOKEID;    // for the Response
  udGroup := UDINT_TO_DWORD(fbAdsWriteInd.IDXGRP);
  udOffset := fbAdsWriteInd.IDXOFFS;       // Prm or Cfg Data
  udLength := fbAdsWriteInd.LENGTH;        // Len of the data
  ptrByte := fbAdsWriteInd.DATAADDR;       // Pointer from the data

  fbAdsWriteRes.RESULT := 0;               // Result without Error, if you want to send an Error use "1"

  IF nPort = 1002 THEN // Slave with Adress 2
    CASE udOffset OF
      16#03230020: // Prm Data
        MEMCPY(ADR(strPrmData2), ptrByte, udLength);
      16#03230021: // Cfg Data
        MEMCPY(ADR(strCfgData2), ptrByte, udLength);
    ELSE
      fbAdsWriteRes.RESULT := 1;
      UnKnownOffset := udOffset;
    END_CASE
  END_IF

  IF nPort = 1003 THEN // Slave with Adress 3
    CASE udOffset OF
      16#03230020: // Prm Data
        MEMCPY(ADR(strPrmData3), ptrByte, udLength);
      16#03230021: // Cfg Data
        MEMCPY(ADR(strCfgData3), ptrByte, udLength);
    ELSE
      fbAdsWriteRes.RESULT := 1;
      UnKnownOffset := udOffset;
    END_CASE
  END_IF

```

```
IF iCase = 0 THEN
  iCase := 10;      // Send response
END_IF
END_IF

CASE iCase OF
  0: (* WAIT *);
  10: (* send ADS-Response *)
    fbAdsWriteRes(
      NETID := sNetId,
      PORT := nPort,
      INVOKEID := udInvokeId,
      RESULT := ,
      RESPOND := TRUE);

    fbAdsWriteInd(CLEAR := TRUE); (* confirm ADS-Indication *)
    iCase := 20;

  20:
    fbAdsWriteRes(RESPOND := FALSE);
    fbAdsWriteInd(CLEAR := FALSE);
    iCase := 0;
END_CASE
```

8.3 IP- und MAC-Adresse auslesen

In diesem Beispiel wird gezeigt, wie Sie die IP- und die MAC-Adresse auslesen können. Mit dem Funktionsbaustein FB_MDP_NIC_Read können Informationen des Netzwerkadapters abgerufen werden.

Beispiel

```

Var
  FB_MDP_NIC_Read      : FB_MDP_NIC_Read;
END_VAR

PROGRAM:
FB_MDP_NIC_Read(
  bExecute:=TRUE ,
  tTimeout:= ,
  iModIdx:= ,
  sAmsNetId:= ,
  bBusy=> ,
  bError=> ,
  nErrID=> ,
  iErrPos=> ,
  stMDP_ModuleHeader=> ,
  stMDP_ModuleContent=> );

```

Der Ausgang stMDP_ModuleHeader zeigt die Header-Informationen an. Der Ausgang stMDP_ModuleContent zeigt unter anderem die Informationen zur IP- und MAC-Adresse an.

stMDP_ModuleHeader	ST_MDP_ModuleHea...
iLen	UINT 4
nAddr	DWORD 131072
sType	T_MaxString 'Nic'
sName	T_MaxString 'st'
nDevType	DWORD 141072
stMDP_ModuleContent	ST_MDP_NIC_Prope...
iLen	UINT 8
sMACAddress	T_MaxString '00:01:05:5f:0f:7a'
sIPAddress	T_MaxString '169.254.123.15'
sSubnetMask	T_MaxString '255.255.0.0'
bDHCP	BOOL TRUE
iReserved	BYTE 0

Abb. 29: Inhalt des MDP-Modules mit IP- und MAC-Adresse.

8.4 Virtuelle Ethernet-Schnittstelle

Die virtuelle Ethernet-Schnittstelle bindet Netzwerkadapter in das TwinCAT-System ein. Damit ist es möglich, eine virtuelle Ethernet-Kommunikation über ADS, TCP oder UDP zu einem BK9xx0 aufzubauen. Verwenden Sie dabei nicht mehr als zwei BK9xx0 und eine Zykluszeit > 50 ms.

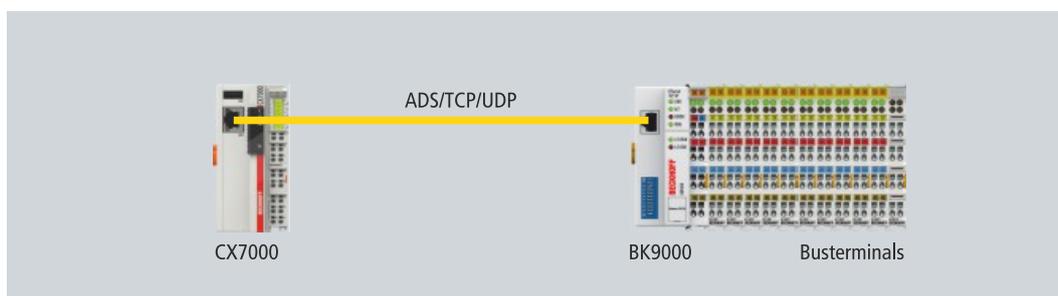
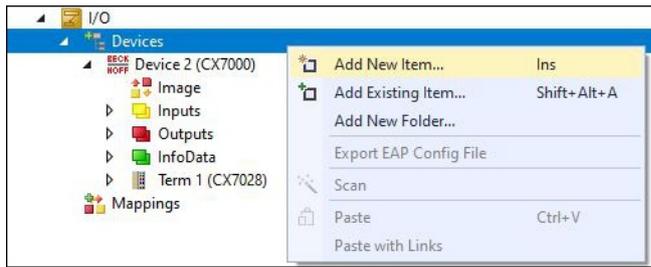


Abb. 30: Virtuelle Ethernet-Kommunikation über ADS, TCP oder UDP.

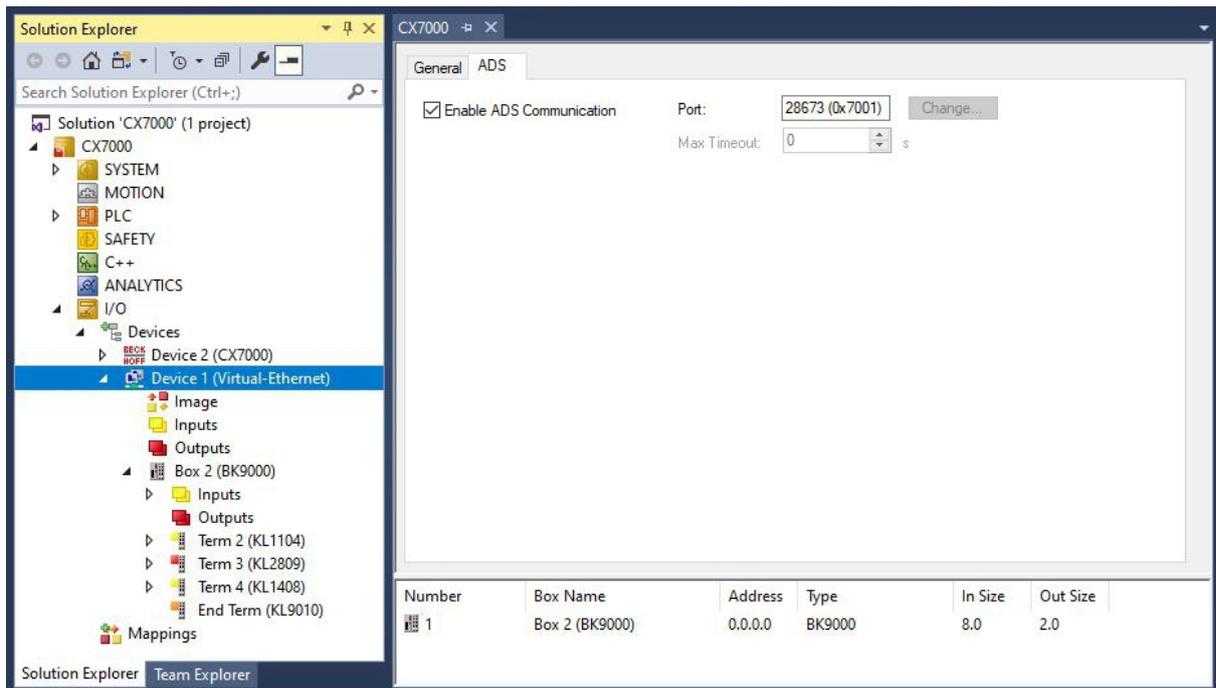
Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum mit der rechten Maustaste auf **Devices**.



2. Klicken Sie auf **Add New Item** und Wählen Sie das **Virtual Ethernet Interface**.

⇒ Das Virtual Ethernet Interface wird links im Strukturbaum angelegt. Unter der Registerkarte **ADS** kann die ADS-Portnummer ausgelesen werden. Die Option **Enable ADS Communication** muss aktiv sein, damit eine ADS-Kommunikation zum BK9xx0 möglich ist.



8.5 CoE-Zugriff auf Multifunktions-I/Os

Mit dem Funktionsbaustein FB_EcCoeSdoReadEx können per SDO-Daten (Service Daten Objekt) Daten aus dem Objektverzeichnis eines EtherCAT-Slaves ausgelesen werden. Mit Hilfe der Parameter nSubIndex und nIndex wird ausgewählt, welches Objekt ausgelesen werden soll. Über bCompleteAccess := TRUE kann der Parameter mit Unterelementen eingelesen werden.

Beispiel: Firmware-Version der Multifunktions-I/Os auslesen.

```

VAR
AMNetID AT %I*:T_AmsNetIdArr;
Port AT %I*:T_AmsPort;
FB_EcCoeSdoReadEx: FB_EcCoeSdoReadEx;
FirmwareVersion: STRING;
END_VAR
    
```

Für die Kommunikation mit der CX7028-Schnittstelle wird die AmsNetId und die Port-Nummer benötigt. Die Eingänge des Funktionsbausteins FB_EcCoeSdoReadEx können mit den Eingangsvariablen netId und port unter TwinCAT verlinkt werden, damit der Funktionsbaustein dauerhaft mit der CX7028-Schnittstelle verbunden ist.



Abb. 31: CoE-Zugriff auf Multifunktions-I/Os, Eingangsvariablen "netId" und "port" unter TwinCAT.

Der Eingang `sNetId` des Funktionsbausteins entspricht dem Eingang `netId` unter TwinCAT. Der Funktionsbaustein verlangt einen String und die Verknüpfung liefert ein Byte-Array. Sie können das Byte-Array mit der Funktion `F_CreateAmsNetId` in einen String umwandeln. Der Eingang `nSlaveAddr` entspricht dem Eingang `port` unter TwinCAT.

```

FB_EcCoESdoReadEx(
sNetId:=F_CreateAmsNetId(nIds:=AMSNID ) , (* AmsNetId of the CX7028 Interface *)
nSlaveAddr:=Port , (* Port Number(nSlaveAddr): 0x1000 *)
nSubIndex:= ,
nIndex:=16#100A , (* Index Number *)
pDstBuf:=ADR(FirmwareVersion) ,
cbBufLen:=SIZEOF(FirmwareVersion) ,
bExecute:=TRUE ,
tTimeout:= ,
bCompleteAccess:= ,
bBusy=> ,
bError=> ,
nErrId=> );
    
```

Die Index-Nummer für das CoE-Objekt **Software version** befindet sich unter der Registerkarte CoE-Online.

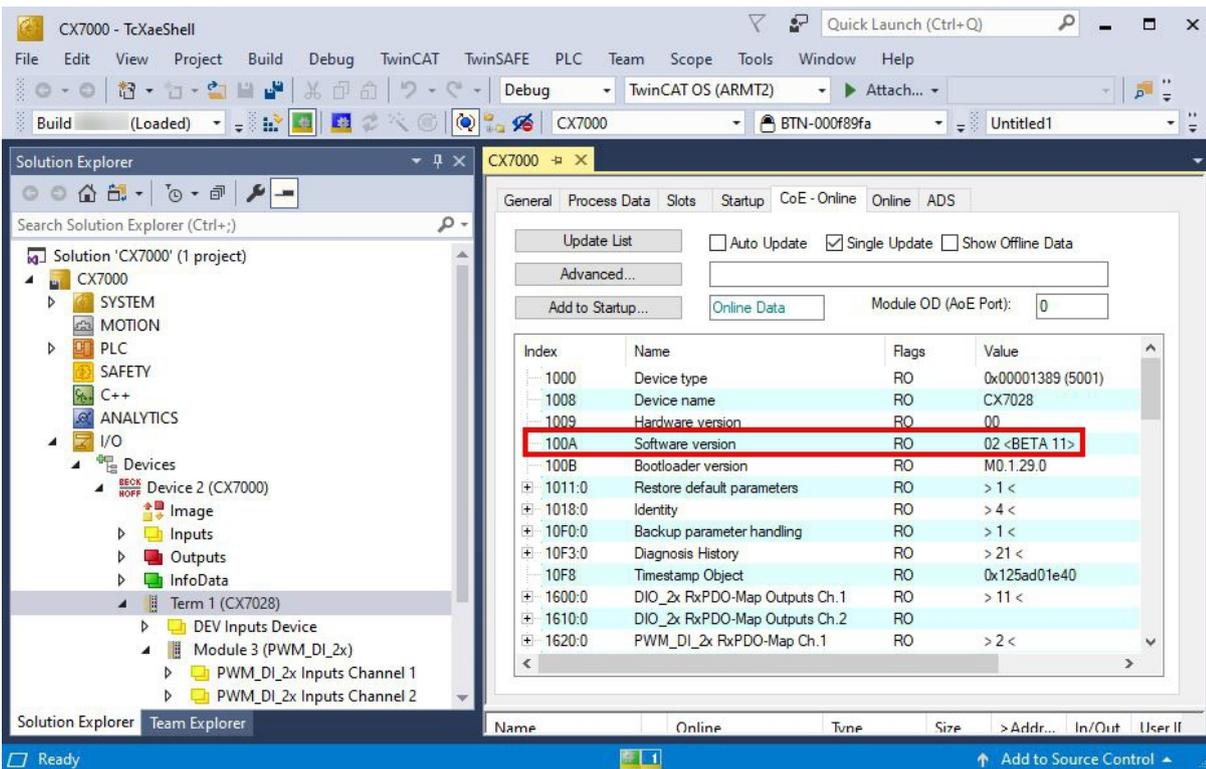


Abb. 32: CoE-Kommunikation, Auflistung der CoE-Objekte mit passender Index-Nummer.

Mit dem Funktionsbaustein `FB_EcCoeSdoWriteEx` kann per SDO-Download ein Objekt aus dem Objektverzeichnis eines EtherCAT Slaves beschrieben werden. Achten Sie darauf, ob auf das Objekt lesen zugegriffen werden kann, was in der Spalte `Flags` angezeigt wird. Mit Hilfe der Parameter `nSubIndex` und `nIndex` wird ausgewählt, welches Objekt beschrieben werden soll. Über `bCompleteAccess := TRUE` kann der Parameter mit Unterelementen geschrieben werden.

8.6 Netzteilklemme

Auf der rechten Seite können wahlweise EtherCAT-Klemmen (E-Bus) oder Busklemmen (K-Bus) angereicht werden; der CX7031 erkennt in der Hochlaufphase automatisch, welches System angeschlossen ist.

K-Bus-Interface

Der CX7031 liest beim Scannen die Klemmentypen aus und legt sie im System Manager unter einem Buskoppler an.

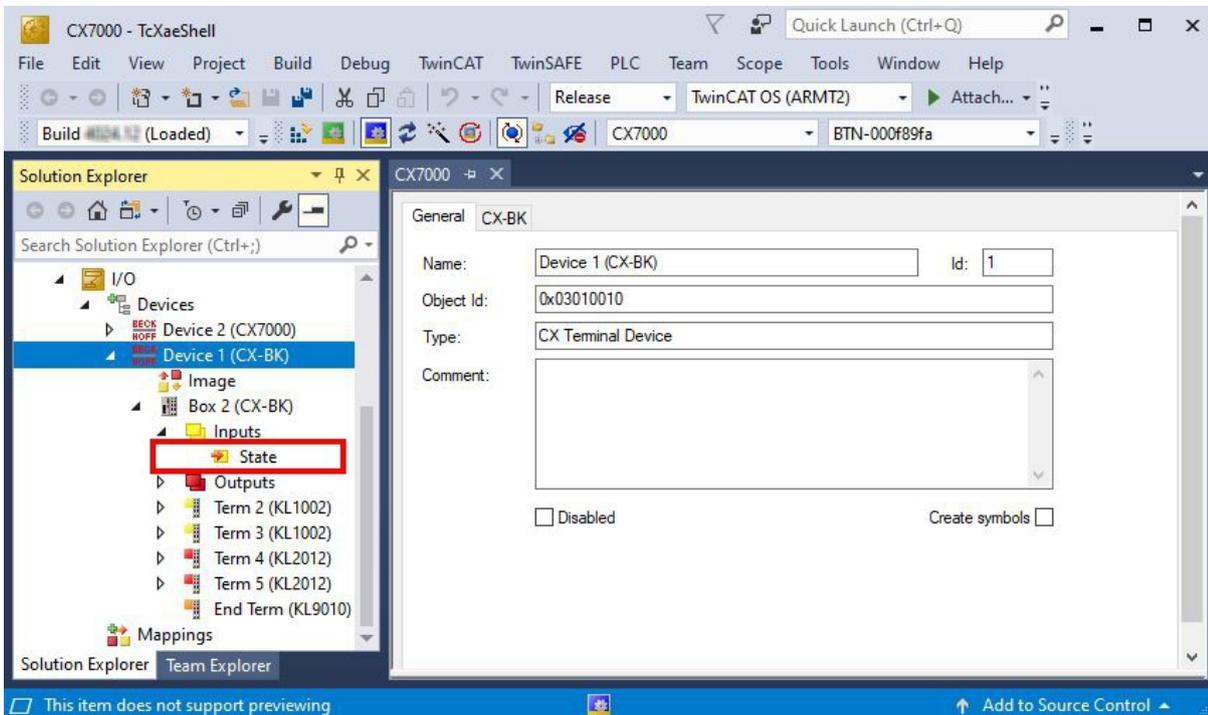


Abb. 33: K-Bus-Interface eines CX7031 im TwinCAT System Manager.

Für die K-Bus-Diagnose gibt es in TwinCAT unter dem Buskoppler eine Status-Variablen, die für Diagnosezwecke genutzt werden kann und den Status der K-Bus-Kommunikation anzeigt. Weitere Informationen finden Sie im Kapitel „Fehlerbehandlung und Diagnose“ unter [K-Bus](#) [▶ 119].

E-Bus-Interface

● Distributed-Clocks

i Die Embedded-PCs der Serie CX7000 eignet sich nicht für den Einsatz von EtherCAT-Slaves, die Distributed-Clocks verwenden oder zwingend voraussetzen.

Auch der Betrieb von EtherCAT-Klemmen und EtherCAT-Geräten ist am CX7031 möglich. Der CX7031 erkennt auch diese Klemmen beim Scannen automatisch, liest die Klemmentypen aus und legt sie im System Manager unter einem EtherCAT-Koppler an.

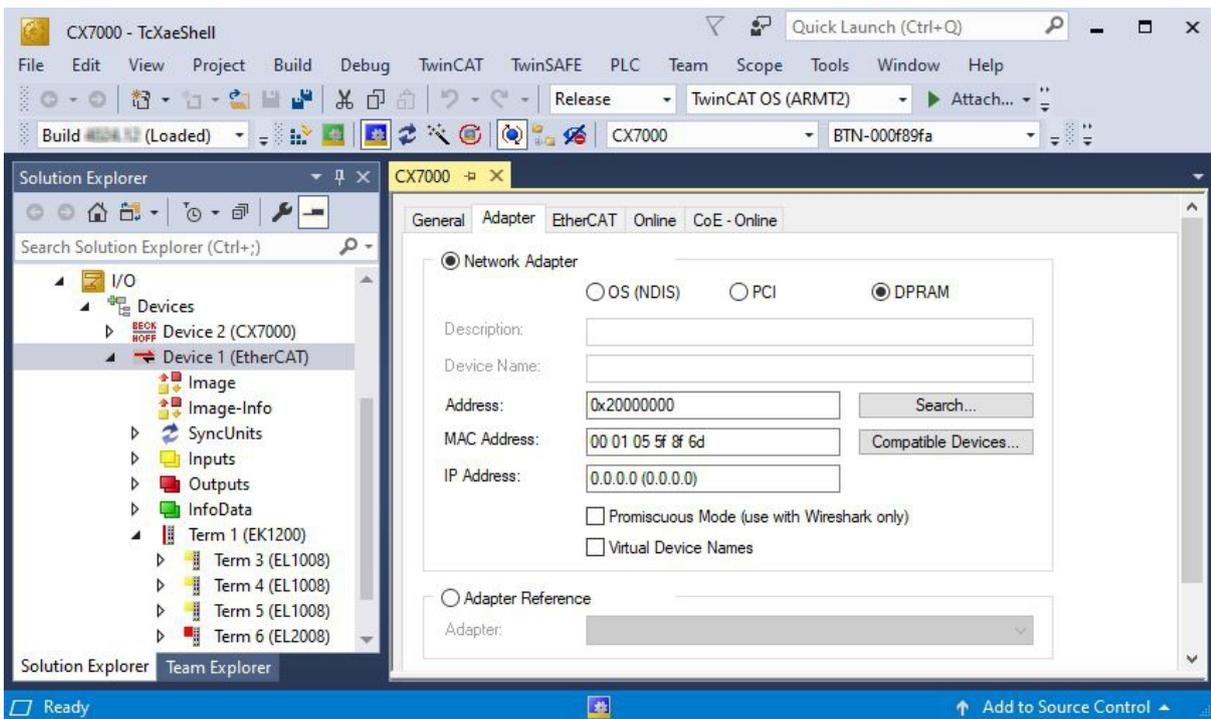


Abb. 34: E-Bus-Interface eines CX7031 im TwinCAT System Manager.

Weitere Informationen zur Diagnose finden Sie im Kapitel „Fehlerbehandlung und Diagnose“ unter [E-Bus](#) [[122](#)].

8.7 Zyklus- und Bearbeitungszeiten

8.7.1 Zykluszeit für PROFIBUS-Konfiguration

Für die PROFIBUS-Konfiguration empfiehlt es sich, eine eigene PROFIBUS-Task zu verwenden, um die Daten möglichst schnell aus dem Empfangsspeicher des PROFIBUS-Controllers (PBS) zu holen.

Die Task sollte möglichst schnell sein (kleine Zykluszeit), da sonst die Task die PROFIBUS-Daten nicht rechtzeitig abholen kann. Eine zu kleine Zykluszeit ist jedoch auch nicht empfehlenswert, da es sonst zu Zykluszeitüberschreitungen kommt. Bei einer langsamen Task (große Zykluszeit) können die Daten nicht aus dem Empfangsspeicher geholt werden und es kommt zu Fehlern auf der PROFIBUS-Seite (No Resources - RR).

Bei ein bis zwei Slaves sollte die PROFIBUS-Task 2 ms betragen, bei drei bis vier Slaves 4 ms. Prüfen Sie mit den Einstellungen die CPU-Last der Task, ggf. können Sie dann optimieren. Diese Einstellungen gelten für eine PROFIBUS-Geschwindigkeit von 12 Mbit/s, wenn Sie kleinere PROFIBUS-Geschwindigkeiten verwenden dürfen, kann die Zykluszeit auch höher als empfohlen sein.

Außerdem sollten Sie, wenn Sie azyklische PROFIBUS-Kommunikation verwenden, diese nicht über die PROFIBUS-Task laufen lassen, sondern dafür eine niederpriorie Task mit höherer Zykluszeit anlegen, z. B. ab 20 ms.

Gehen Sie nie an die Grenze, lassen Sie dem System noch genügend Spielraum, ca. 60...70 % CPU-Last sollte die Grenze sein.

8.7.2 Bearbeitungszeit im SPS-Programm messen

In diesem Beispiel wird gezeigt, wie Sie die Bearbeitungszeit eines Programmcodes mithilfe eines kleinen SPS-Programms bestimmen können. Damit können Sie beispielsweise messen, wie lange die SPS für eine mathematische Funktion eine Schleife oder einen bestimmten Programmteil benötigt. Die Auflösung beträgt 1 ns pro Digit.

Beispiel

```
VAR
    MeasureStart      : T_DCTIME64;
    MeasureResult     : T_DCTIME64;
END_VAR

PROGRAM:
MeasureStart:=F_GetActualDcTime64(); (*Insert your program code to measure the processing time*)
MeasureResult:=F_GetActualDcTime64()-MeasureStart;
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_EtherCAT

8.7.3 Real-Time-Clock (RTC)

Der CX7031 hat eine interne, kondensatorgepufferte Realtime-Clock (RTC) für Zeit und Datum, die im ausgeschalteten Zustand weiterläuft. Die Kapazität des Kondensators reicht für mindestens 30 Tage und ist anders als eine batteriegestützte Lösung wartungsfrei. Ist der CX7031 länger als 30 Tage ausgeschaltet, geht die Uhrzeit verloren und muss neu eingestellt werden

Folgende Einstellungen sind in der boot.conf-Datei möglich:

- SNTP-Server
- Update-Time (Standardeinstellung = 1 Stunde)
- Change UTC Offset
- DHCP-Server

Beispiel

In dem unteren Beispiel wird gezeigt, wie Sie die Uhrzeit auslesen können. In dem Beispiel wird die Uhrzeit als UTC-Zeit ausgegeben und eine Stunde addiert, um die MEZ-Zeit zu erhalten.

```

VAR
  FB_LocalSystemTime      : FB_LocalSystemTime;
  DATEANDTIME             : DATE_AND_TIME;
  DATEANDTIME_Add1h      : DATE_AND_TIME;
END_VAR

PROGRAM:
FB_LocalSystemTime (
  sNetID:= ,
  bEnable:=TRUE ,
  dwCycle:= ,
  dwOpt:= ,
  tTimeout:= ,
  bValid=> ,
  systemTime=> ,
  tzID=> );

DATEANDTIME:=SYSTEMTIME_TO_DT(TIMESTR:=FB_LocalSystemTime.systemTime );      (*UTC Time*)
DATEANDTIME_Add1h:=DATEANDTIME+T#1H;      (*UTC Time + 1h*)

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/ BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

8.7.4 I/O-Idle-Task:

Die I/O-Idle-Task führt die asynchronen Mappings aus und ist verantwortlich für die azyklische Kommunikation mit den EtherCAT-Teilnehmern (z.B. Schreiben von Parametern).

Empfohlene Einstellung für die I/O-Idle-Task

Standardmäßig ist die I/O-Idle-Task bei TwinCAT-Steuerungen auf 1 ms eingestellt. Bei leistungsfähigen Steuerungen hat diese Standardeinstellung keine Auswirkung und kann vernachlässigt werden. Bei einer kleinen Steuerung wie dem CX70xx führt die Standardeinstellung von 1 ms jedoch zu einer höheren Grundauslastung. Daher sollte als erster Schritt:

- die I/O-Idle-Task auf 10 ms gesetzt werden
- und die I/O-Idle-Task in der Prioritätenliste immer an das Ende gestellt werden.

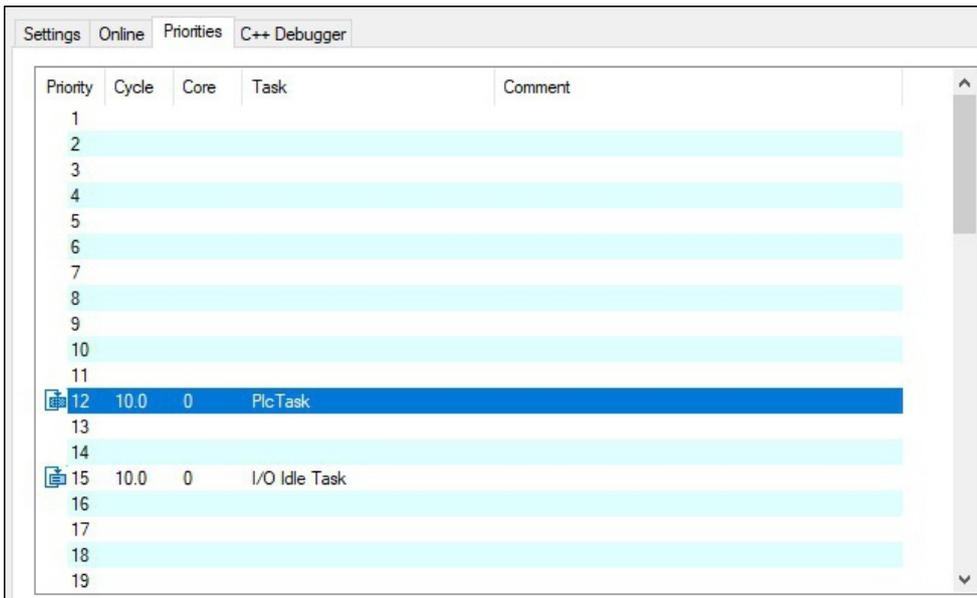


Abb. 35: Einstellung für die I/O-Idle-Task im TwinCAT-3- Engineering.

Zykluszeitüberschreitung der I/O-Idle-Task

Eine Zykluszeitüberschreitung der I/O-Idle-Task kann ebenfalls vernachlässigt werden, es sei denn, die I/O-Idle-Task zählt kontinuierlich hoch. In diesem Fall sollte versucht werden, die azyklische Kommunikation zu verlangsamen, weniger Daten abzufragen und die Anzahl der azyklischen Kommunikationen zu reduzieren.

8.7.5 NTP-Provider einsetzen

Auf dem CX70xx ist ein NTP-Provider ab Image-Version 140501 oder höher vorhanden.

Der NTP-Provider kann auf verschiedenen CX70xx Embedded-PCs eingesetzt werden, um eine annähernd gleiche Zeit bereitzustellen. Zwischen zwei CX70xx verbleibt typischerweise eine Abweichung von ± 5 ms.

Weitere Informationen dazu, wie der NTP-Provider als TcCOM-Modul in TwinCAT eingefügt und in Betrieb genommen wird, finden Sie hier: https://infosys.beckhoff.com/content/1031/tc3_grundlagen/6326712203.html?id=2077149792280564536

Beispielprogramm für die Verwendung des NTP-Providers

<https://infosys.beckhoff.com/content/1031/CX7031/Resources/16683737611.zip>

Variablendeklaration im ST-Editor

```

PROGRAM TimeSync
VAR
(*
Time format:
FileTime64      8 Byte len      begin 1.1.1601      Bit/100ns
DCTime64       8 Byte len      begin 1.1.2000     Bit/100ns
*)
    Test_FileTime64:T_FILETIME64;    (* not used only to see where
                                     this time begin to see the different *)
    Test_DCTime64:T_DCTIME64;        (* not used only to see where
                                     this time begin to see the different *)

    nLocalTime      : T_FILETIME64;    // Filetime 64 - Local RTOS Time
    nPLC_DC_TASKTime : T_FILETIME64;    (* Filetime 64 - Local DC Time | Local DC Time =
                                     System Time see DiffSysTimeToPLCDCTime *)

    nSysTime AT %I* : T_FILETIME64;    // Filetime 64 - Sytem Time
    nExtTime AT %I* : T_FILETIME64;    // Filetime 64 - TCNet Ext NTP Time
    nSysToExtTimeOffset AT %I* :LINT;  // Offset this is the offset from ExtTime to SysTime
    nSysToExtTimeDeviation AT %I* :LINT; (* Only for Info: Diff from NTP to the NPT Filter
                                     time is it < 1ms then is bIsSynchronized = TRUE *)

```

```

bIsConnected AT %I*:BOOL;          // Connection to the NTP Server
bIsSynchronized AT %I*:BOOL;      // see nSysToExtTimeDeviation

FB_LocalSystemTime: FB_LocalSystemTime;    // Functionblock for read the local RTOS Time

dt_LocalTime: DT;                   // Time & data - local RTOS Time
dt_PLC_DC_TASKTime: DT;              (* Time & data - This is the time from the TaskInfo Array, this
                                     is the same time as the Sytem Time from the NTP Object *)
dt_nSysTime: DT;                     // Time & data - Sytem Time from the NTP Object
dt_nExtTime: DT;                     // Time & data - NTP Object Ext NTP Time

PlcTaskSystemInfo                   : PlcTaskSystemInfo;    // Task Info

DiffSysTimeToPLCDCTime: LINT;        // Local DC Time = System Time
DiffLocalTimeToExtTime: LINT;        (* local Time - ExtTime = same time like
                                     the nSysToExtTimeOffset Time *)
FB_EcDcTimeCtrl164_1:FB_EcDcTimeCtrl164; // FB to get Sec/min/Hour... from a DCTime64
nExtTime64: ULINT;                   // Format DCTime64 from nExtTime
sec_1: WORD;                          // nExtTime Sec
Min_1: WORD;                           // nExtTime Min
msec_1: WORD;                           // nExtTime mSec

nLocalTimeDC64: ULINT;                // DCTime64 - from local filetype64
MaxPLCTime: UDINT;

lrTimeoffset: LREAL;                  // Offset Time in LREAL [ms]
lrTimeDeviation: LREAL;               (* Deviation Time [ms] this time show you the diff to the
                                     NTP filter time, if its <lms the is bIsSynchronized = TRUE *)

FB_CX7000_LED_WD: FB_CX7000_LED_WD;   // NPT Status via WD LED from CX7000

END_VAR

```

Programm im ST-Editor

```

PlcTaskSystemInfo:=_TaskInfo[1];      (* Get Task Info Data we need it to read the
                                     PlcTaskSystemInfo.DcTaskTime *)

dt_PLC_DC_TASKTime:=Systemtime_TO_DT(DCTime64_TO_SYSTEMTIME(LINT_TO_ULINT(PlcTaskSystemInfo.DcTaskTime)));
// Get Local DC Time to DT
nPLC_DC_TASKTime:=DCTIME64_TO_FILETIME64(LINT_TO_ULINT(PlcTaskSystemInfo.DcTaskTime));
// Get Local DC Time to FileTime64

MaxPLCTime:=MAX(PlcTaskSystemInfo.LastExecTime,MaxPLCTime);

lrTimeoffset:=LINT_TO_LREAL(nSysToExtTimeOffset)/1000/10; // Read NTP Offset Time and convert it to
Real in [ms]
lrTimeDeviation:=LINT_TO_LREAL(nSysToExtTimeDeviation)/1000/10; // Read NTP Offset Time and convert
it to Real in [ms]

// Read the local time from RTOS
FB_LocalSystemTime(
  sNetID:= ,
  bEnable:=TRUE ,
  dwCycle:= ,
  dwOpt:= ,
  tTimeout:= ,
  bValid=> ,
  systemTime=> ,
  tzID=> );

dt_LocalTime:=SystemTime_TO_DT(FB_LocalSystemTime.systemTime);
nLocalTime:=SystemTime_TO_Filetime64(FB_LocalSystemTime.systemTime);

dt_nSysTime:=Filetime64_TO_DT(nSysTime); // Convert nSysTime to date & time
dt_nExtTime:=Filetime64_TO_DT(nExtTime); // Convert nExtTime to date & time

DiffSysTimeToPLCDCTime:=ULINT_TO_LINT(nPLC_DC_TASKTime-nSysTime); // Must be 0 because
SysTime = Local DC Time
DiffLocalTimeToExtTime:=ULINT_TO_LINT(nExtTime-nPLC_DC_TASKTime); // Same diff like
nSysToExtTimeOffset

// Ext NTP object as trigger for Outputs
nExtTime64:=FILETIME64_TO_DCTIME64(nExtTime);
FB_EcDcTimeCtrl164_1.A_GetMilli (in:=nExtTime64,get=>msec_1);

```

```

FB_EcDcTimeCtrl64_1.A_GetSecond (in:=nExtTime64,get=>sec_1);
FB_EcDcTimeCtrl64_1.A_GetMinute (in:=nExtTime64,get=>Min_1);

// Toggle Outpt every Sec
IF (sec_1 MOD 2) =0 THEN
    gvl.bCX7028_Out_4:=TRUE;
ELSE
    gvl.bCX7028_Out_4:=FALSE;
END_IF

// Set DO NTP Status
gvl.bCX7028_Out_1:=bIsConnected;
gvl.bCX7028_Out_2:=bIsSynchronized;

// WD LED from CX7000
// Red - no connection
// Green flashing - it is connected to NTP Server but it is not sync (that means that the filter is
// active and the synchronisation to the NTP Server is running - not with TC!)
// Green on - connection and synchronisation is running
FB_CX7000_LED_WD(
    bEnable:=TRUE ,
    eLED:= ,
    tFlashingTimeP1:= ,
    tFlashingTimeP2:= ,
    bError=> ,
    nErrorID=> );
IF bIsConnected AND NOT bIsSynchronized THEN
    FB_CX7000_LED_WD.eLED :=Tc2_SystemCX.E_CX7000_LED.LED_flashing_GREEN_OFF;
ELSIF bIsConnected AND bIsSynchronized THEN
    FB_CX7000_LED_WD.eLED :=Tc2_SystemCX.E_CX7000_LED.LED_GREEN ;
ELSE
    FB_CX7000_LED_WD.eLED :=Tc2_SystemCX.E_CX7000_LED.LED_RED ;
END_IF

```

8.7.6 Zykluszeit von 250 µs

Beachten Sie, dass eine Zykluszeit von 250 µs auf einem CX7031 ein extremes Optimum darstellt und alle Rahmenbedingungen passen müssen. Des Weiteren ist eine Zykluszeit von 250 µs nur dann sinnvoll, wenn die Ein- und Ausgänge entsprechend schnell sind.

Der CX7031 hat unterschiedliche Schnittstellen, darunter beispielsweise den K-Bus. Der K-Bus schafft unter optimalen Bedingungen vielleicht 1 ms und ist daher nicht für Zykluszeiten von 250 µs geeignet. Der E-Bus (EtherCAT) ist viel schneller, nur ist der Aufbau eines EtherCAT-Frames und das Zusammenfügen der Daten in einen EtherCAT-Frame wesentlich aufwendiger, sodass auch hier nur 1 ms möglich sind.

Natürlich kann EtherCAT mit anderen Industrie-PCs unter 100 µs betrieben werden. Diese sind in der Regel aber mit leistungsfähigeren CPUs ausgestattet und nutzen eventuell einen DMA-Controller für die EtherCAT-Bearbeitung. Das ist beim CX7031 allerdings nicht der Fall, sodass die CPU-Leistung und die Schnittstellen zum EtherCAT die begrenzenden Faktoren sind. Natürlich ist der CX7031 als Kleinsteuerung nicht für Highspeed-Anwendungen entwickelt worden und sollte aufgrund seiner Kosteneffizienz nicht mit leistungsfähigeren Industrie-PCs verglichen werden.

Zykluszeit von 250 µs einstellen

Eine Zykluszeit von 250 µs ist auf einem CX7031 möglich, wenn die Rahmenbedingungen stimmen. Beim CX7031 helfen die Multifunktions-I/Os, die über eine schnelle IO-Verbindung mit der CPU verbunden sind. Die Verbindung ist sehr schlank gehalten und hat einen entsprechend guten Datendurchsatz. Mithilfe der Multifunktions-I/Os ist es möglich, die 250 µs zu erreichen. Das SPS-Programm darf natürlich nur sehr wenig Code enthalten und das Core-Limit auf 90 % eingestellt werden, was wiederum die beschriebenen Nachteile mit sich bringt (siehe: [Echtzeit und CPU-Auslastung](#) [▶ 132]).

Core	RT-Core	Base Time	Core Limit	Latency Warning
0	<input checked="" type="checkbox"/> Default	250 μ s	90 %	(none)

Zusätzlich sollten Sie die Priorität der Task so einstellen, dass die 250 μ s Task die höchste Priorität im System hat.

Priority	Cycle	Core	Task	Comment
1				
2				
3				
4				
5				
6				
7				
8	0.25	0	PlcTask	
9				
10				
11				
12				
13				
14				
15				
16				
17	5.0	0	PlcTask2	
18				
19	10.0	0	I/O Idle Task	
20				
21				
22				
23				
24				

Wenn Sie nun einen digitalen Ausgang des CX7028-Schnittstelle toggeln lassen, in der 250 μ s Task beispielsweise mit `Out_01:=not Out_01`, so wird diese mit einer Frequenz von 2 kHz ausgegeben. Damit der Ausgang optimal schnell ist, sollte dieser Ausgang eine Last haben. Verkabeln Sie den Ausgang nur mit einem digitalen Eingang, dadurch ist die Last sehr klein und das Ausschaltverhalten des Treibers relativ langsam. Langsam bezieht sich hier auf die 250 μ s Taskzeit. Es macht schon einen Unterschied aus, ob der Ausgang 50 μ s oder 100 μ s zum Ausschalten benötigt. Wenn sie nun die Reaktionszeit messen wollen, also die Zeit wie lange der CX7031 braucht, um auf einen Eingang zu reagieren, ist folgender Hintergrund wichtig:

Ab einer Zykluszeit von 1 ms oder größer wird ein optimaler Zyklus gefahren, d.h. die Eingänge der CX7028-Schnittstelle werden ca. 20 % vor dem neuen Taskzyklus vom Prozessor der CX7028-Schnittstelle eingelesen. Ist die Taskzeit schneller als 1 ms, so reicht die Zeit für die optimierte Reaktionszeit nicht aus. Hier werden dann die Eingänge mit dem Taskzyklus eingelesen. Das hat zur Folge, dass eine Taskzeit von 500 µs die gleiche Reaktionszeit erreicht wie Taskzeit von 1 ms. Unter 1 ms Taskzeit braucht das Update für einen Zyklus 4 Taskzyklen. In 1 ms oder langsamer zwei Taskzyklen. Dies soll ihnen bewusst machen, dass nicht immer die Verkürzung der Zykluszeit die Reaktionszeit verkürzt, sondern auch der interne Ablauf, der eine entscheidende Rolle beim Einlesen der Daten spielt.

Ein Beispiel, damit Sie dieses Verhalten selber reproduzieren können, die Unterschiede sehen und messen können:

1. Schließen Sie die Spannungsversorgung +24 V Up und 0 V Up zur Versorgung der Multifunktions-I/Os an.
2. Verbinden Sie den Ausgang 1 mit dem Eingang 1, um den Ausgang wie beschrieben zu toggeln.
3. Verbinden Sie den Ausgang 2 mit dem Eingang 2.
4. Stellen Sie das Core-Limit auf 90 % ein, die Base-Zeit auf 250 µs, die Priorität der schnellen Task auf die höchste Priorität und die Idle-Task auf 10 ms.

Die Eingänge haben nur eine minimale Filterzeit und sind daher gut für die Messung geeignet. Eine Last am Ausgang ist in dem Fall nicht notwendig. Die Base-Zeit belassen wir für die folgenden Beispiele immer auf 250 µs und erhöhen nur die Anzahl der Cycle-Ticks, um die entsprechende Taskzeit einzustellen.

Beispielprogramm

```
PROGRAM MAIN
VAR
  bOut_1 AT %Q*:BOOL; (*toggle Output link to digital Output pin 7*)
  bOut_2 AT %Q*:BOOL; (*reaction time link to digital Output pin 14*)

  bIn_1 AT %I*: BOOL; (*toggle Output link to digital Input pin 2*)
  bIn_2 AT %I*: BOOL; (*reaction time link to digital Input pin 10*)

  fbTimer : TON;
  fbflanke1 : R_TRIG;
  fbflanke2 : R_TRIG;

  cnt1: INT; (*toggle Output*)
  cnt1_M: INT; (*toggle Output*)

  cnt2: INT; (*reaction time*)
  cnt2_M: INT; (*reaction time*)
END_VAR

PROGRAM MAIN
bOut_1:= NOT bOut_1; (*toggle Output*)
```

```

bOut_2:= NOT bIn_2; (*reaction time*)

fbflanke1(CLK:=bIn_1);
IF fbflanke1.Q THEN
  cnt1:=cnt1+1; (*toggle Output*)
END_IF

fbflanke2(CLK:=bIn_2);
IF fbflanke2.Q THEN
  cnt2:=cnt2+1; (*reaction time*)
END_IF

fbTimer(PT:=T#1S,in:=NOT fbTimer.Q);

IF fbTimer.Q THEN
  cnt2_M:=cnt2; (*reaction time*)
  cnt1_M:=cnt1; (*toggle Output*)
  cnt1:=0;
  cnt2:=0;
END_IF

```

Das Toggeln des Ausgangs bewirkt eine Frequenz von 2 kHz, 250 µs On 250 µs Off, also eine Periodendauer von 500 µs. Bei einer Messung der positiven Flanke sind dies 2000 Flankenwechsel in einer Sekunde.

bOut_1	BOOL	TRUE	
bOut_2	BOOL	TRUE	
bIn_1	BOOL	FALSE	
bIn_2	BOOL	FALSE	
fbTimer	TON		
fbflanke1	R_TRIG		
fbflanke2	R_TRIG		
cnt1	INT		1014
cnt1_M	INT	2000	
cnt2	INT		253
cnt2_M	INT		500

Abb. 36: Messung bei einer Taskzeit von 250 µs.

Bei der Reaktionszeit sind es 500 Wechsel in einer Sekunde, da hier der optimierte Zugriff auf die Eingänge nicht greift.

bOut_1	BOOL	TRUE	
bOut_2	BOOL	TRUE	
bIn_1	BOOL	TRUE	
bIn_2	BOOL	FALSE	
fbTimer	TON		
fbflanke1	R_TRIG		
fbflanke2	R_TRIG		
cnt1	INT		68
cnt1_M	INT	1001	
cnt2	INT		17
cnt2_M	INT		250

Abb. 37: Messung bei einer Taskzeit von 500 µs.

Wie zu erwarten sind bei einer doppelt so großen Taskzeit die Werte nur halb so groß.

◆ bOut_1	BOOL	FALSE
◆ bOut_2	BOOL	TRUE
◆ bIn_1	BOOL	FALSE
◆ bIn_2	BOOL	FALSE
⊕ ◆ fbTimer	TON	
⊕ ◆ fbflanke1	R_TRIG	
⊕ ◆ fbflanke2	R_TRIG	
◆ cnt1	INT	169
◆ cnt1_M	INT	501
◆ cnt2	INT	84
◆ cnt2_M	INT	251

Abb. 38: Messung bei einer Taskzeit von 1 ms.

Bei einer Taskzeit von 1 ms kann man deutlich sehen, dass der optimierte Modus tatsächlich hilft, die Reaktionszeit zu verringern. Während der Toggle-Wechsel sich wieder halbiert hat, also bei einer Taskzeit von 1 ms jetzt noch bei 500 Hz liegt, ist der Wert bei der Reaktionszeit gleichgeblieben.

8.7.6.1 Zykluszeit ≥ 1 ms



Abb. 39: CX7031 CPU und SPS.

Gelb und rot: Mapping und Update der IOs.

Hellgrau: Restzeit bis zum erneuten Task-Begin (OS)

Dunkelgrau: PLC Zyklus.

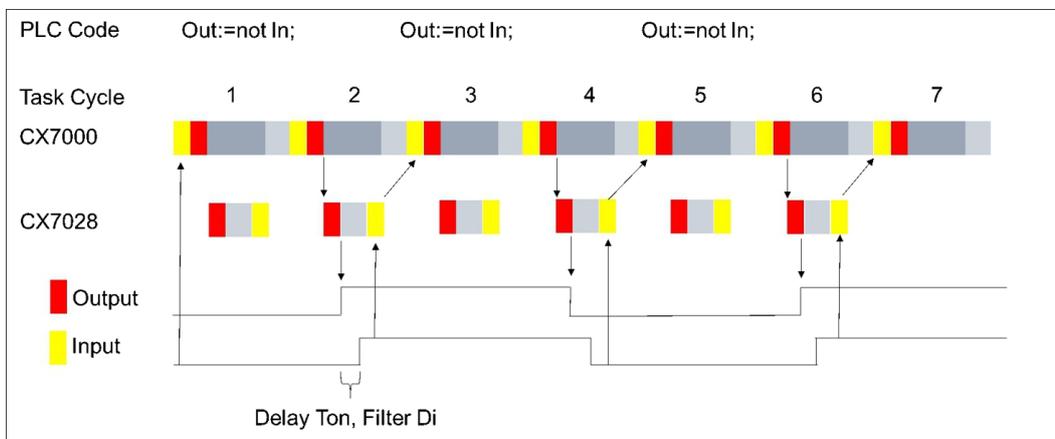


Abb. 40: CPU der CX7028-Schnittstelle.

Rot: Output Update.

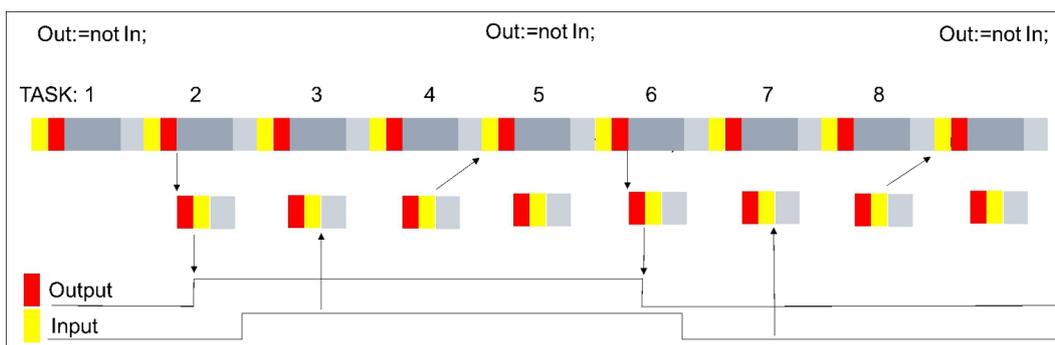
Grau: CPU Bearbeitung der Multifunktions-IOs.

Gelb: Input Update (ab 1 ms Zykluszeit wird mit dem Update der Eingangssignale bis ca. 80 % der Zykluszeit gewartet, so das möglichst spät, also vor dem nächsten Zyklus die Eingänge eingelesen werden).



8.7.6.2 Zykluszeit < 1 ms

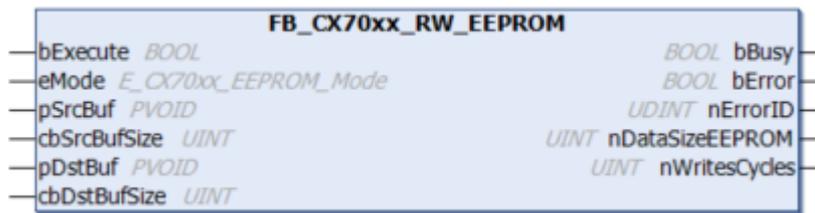
Ab einer Zykluszeit von < 1 ms wird der Update der Eingangssignale sofort durchgeführt und steht damit erst mit dem nächsten Zyklus zur Verfügung. Die Eingangssignale sind somit immer ein Zyklus alt.



Mit diesem Hintergrundwissen sollten Sie in der Lage sein, für ihre Anwendung die richtigen Einstellungen am CX7031 vorzunehmen.

8.8 Funktionsbausteine

8.8.1 FB_CX70xx_RW_EEPROM



Der Baustein erlaubt das Schreiben von maximal 120 Byte in das EEPROM (Hardware) des CX70xx. Das EEPROM darf maximal 200-mal beschrieben werden. Der Speicher ist für das einmalige Schreiben gedacht.

Dieser Funktionsbaustein kann dafür verwendet werden, den CX70xx zu personalisieren. Das heißt, im einfachsten Fall schreiben Sie Ihre Firmenkennung in das EEPROM. Beim Start des CX70xx Programms lesen Sie den Inhalt des Speichers aus. Ist dieser zum Beispiel leer, können Sie das Programm nicht weiter ausführen, da es sich nicht mehr um Ihren originalen CX70xx handelt, den Sie programmiert haben.

Sollten Sie einen CX70xx gegen ein neues Gerät austauschen wollen, muss das EEPROM erneut von Ihnen geschrieben werden.

Eingänge

```
VAR_INPUT
  bExecute      : BOOL;           // rising edge triggers process with selected mode
  eMode         : E_CX70xx_EEPROM_Mode; // select RW mode
  pSrcBuf       : PVOID;         // pointer to WRITE EEPROM data buffer
  cbSrcBufSize  : UINT;          // size of WRITE EEPROM data buffer (max.120 Bytes)
  pDstBuf       : PVOID;         // pointer to READ EEPROM data buffer
  cbDstBufSize  : UINT;          // max.size of READ EEPROM data buffer (max.120 Bytes)
END_VAR
```

Name	Typ	Beschreibung
bExecute	BOOL	Positive Flanke startet den Baustein.
eMode	E_CX70xx_EEPROM_Mode	ReadOnly: EEPROM Lesen WriteOnly: EEPROM Schreiben WriteAndRead: EEPROM Schreiben und im Anschluss Lesen
pSrcBuf	PVOID	Pointer auf den Datenpuffer, der geschrieben werden soll.
cbSrcBufLen	UINT	Länge der zu schreibenden Daten (maximal 120 Bytes)
pDstBuf	PVOID	Pointer auf den Datenpuffer, in den der Inhalt des EEPROM kopiert werden soll.
cbDstBufLen	UINT	Länge der zu lesenden Daten. (maximal 120 Bytes) Beim Lesen muss die Längeninformation größer gleich der im EEPROM enthalten Daten sein.

Ausgänge

```
VAR_OUTPUT
  bBusy         : BOOL;           // FB is working
  bError        : BOOL;           // FB has an Error
  nErrorID      : UDINT;         (* Error Code
  If nErrorID=DEVICE_INVALIDACCESS the EEPROM write cycles reached max. value.
  If nErrorID=DEVICE_INVALIDPARG the given pointer parameter is invalid/null.
  If nErrorID=DEVICE_INVALIDSIZE the given buffer size is too small or too big.
  If nErrorID=DEVICE_SRVNOTSUPP probably the image version need to be updated to support this feature. *)
  nDataSizeEEPROM : UINT;         // current size of (read) EEPROM data in bytes (max.120 Bytes)
  nWritesCycles  : UINT;         // already performed EEPROM write cycles (maximum possible = 200)
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_INVALIDACCESS: Die EEPROM Schreibzyklen haben den maximalen Wert erreicht. Das EEPROM kann nicht erneut beschrieben werden. DEVICE_INVALIDPARG: Die zugewiesenen Pointer sind ungültig/ NULL. DEVICE_INVALIDSIZE: Die zugewiesene Puffergröße ist zu klein oder zu groß. DEVICE_SRVNOTSUPP: Die Image-Version des CX70xx unterstützt dieses Feature nicht. Ein Update (>=35695) ist notwendig.
nDataSizeEEPROM	UINT	Aktuelle Größe in Bytes der gelesenen EEPROM Daten
nWritesCycles	UINT	Anzahl der noch zur Verfügung stehenden Schreibvorgänge

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX70xx	Tc2_SystemCX (System) >= 3.4.8.0

8.8.2 FB_CX70xx_ResetOnBoardIO



Der Baustein erlaubt es, einen Reset vom OnBoard-IO des Embedded-PC CX70xx auszuführen.

Typischer Anwendungsfall ist nach einem Fehler in der Kommunikation zu den OnBoard-IOs (CX7028). Zu einem solchen Fehler kommt es, wenn die Spannungsversorgung (Up) der OnBoard-IOs unterbrochen wird.

HINWEIS

Zustand der IOs

Ausgänge, die im Prozessabbild noch gesetzt sind, werden nach einem Reset sofort wieder eingeschaltet.

Weitere Details zum OnBoard-IO finden Sie in der [Dokumentation des Embedded-PC CX70xx](#).

Eingänge

```

VAR_INPUT
  bExecute      : BOOL;           // rising edge triggers process
  sNetId        : T_AmsNetID;    // AMS Net ID of the OnBoard IOs
  tTimeout      : TIME := DEFAULT_ADS_TIMEOUT; // maximum time allowed for execution of this ADS command
END_VAR

```

Name	Typ	Beschreibung
bExecute	BOOL	Positive Flanke startet den Baustein.
sNetId	T_AmsNetID	AMS Net ID der OnBoard-IOs
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

 **Ausgänge**

```
VAR_OUTPUT
  bBusy      : BOOL;          // FB is working
  bError     : BOOL;          // FB has an Error
  nErrorID   : UDINT;         (* Error Code. If nErrorID=DEVICE_SRVNOTSUPP probably the image versio
n need to be updated to support this feature. *)
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_SRVNOTSUPP: Die Image-Version des CX70xx unterstützt dieses Feature nicht. Ein Update (>=47912) ist notwendig.

Beispiel:

```
FUNCTION_BLOCK FB_Test_ResetOnboardIO
VAR
  AMSNetID   : T_AmsNetIdArr;    // link to the AMS Net ID of the OnBoard IOs
  State      : WORD;             // link to the State of the OnBoard IOs
  bReset     : BOOL;             // if Ready to Reset you can reset the OnBoard IOs
  fbReset    : FB_CX70xx_ResetOnBoardIO;
END_VAR

IF State<>8 AND NOT State.8 AND State.4 THEN // if OnBoard IO device signals an error and is not OP
but present
  bReset := TRUE;
ELSE
  bReset := FALSE;
END_IF

IF NOT fbReset.bBusy AND bReset THEN
  fbReset(bExecute:=TRUE, sNetId:=F_CreateAmsNetId(AMSNetID));
ELSE
  fbReset(bExecute:=FALSE);
END_IF
```

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX70xx	Tc2_SystemCX (System) >= 3.4.8.0

8.9 Wichtige Attribut-Pragmas

Attribut-Pragmas dienen dazu, die Kompilierung und die Vorkompilierung zu beeinflussen. TwinCAT unterstützt eine Reihe von vordefinierten Attribut-Pragmas. Attribute werden im Deklarationsteil definiert.

8.9.1 Attribut 'Tc2GvlVarNames'

Das Pragma bewirkt, dass Symbole, welche in einer GVL deklariert sind, über ADS genauso angesprochen werden wie in TwinCAT 2, ohne die Verwendung des GVL-Namens als Namespace.

Syntax: {attribute 'Tc2GvlVarNames'}

Beispiel:

```
{attribute 'Tc2GvlVarNames'}
VAR_GLOBAL
  Test : INT;
END_VAR

GVL.Test:=GVL.Test+1;    (*without attribute*)
Test:=Test+1;           (*with attribute*)
```

8.9.2 Attribut 'pack_mode'

Dieses Attribut-Pragma legt fest, wie eine Datenstruktur während der Allokierung gepackt wird. Das Attribut muss oberhalb der Datenstruktur eingefügt werden und wirkt sich auf das Packen der gesamten Struktur aus.

Syntax: {attribute 'pack_mode' := '<Value>'}

Beispiel

```
{attribute 'pack_mode' := '0'}
TYPE str_Test :
STRUCT
    byTest1    : BYTE;
    iTest      : DINT;
    byTest2    : BYTE;
    nValue     : INT;
END_STRUCT
END_TYPE
```

In diesem Beispiel wurde der Pack-Modus auf 0 gestellt. Wenn Sie die Größe der Struktur im Beispiel mit SIZEOF bestimmen, dann erhalten Sie den Wert 8.

1 Byte + 4 Byte (DINT) + 1 Byte + 2 Byte (INT) = 8 Byte

Wenn Sie den Pack-Modus auf 2 stellen (WordAlignment), erhalten Sie den Wert 10, weil nach jedem Byte noch ein Füllbyte eingefügt wird. Wenn Sie den Pack-Modus auf 4 stellen (DWordAlignment), dann erhalten Sie den Wert 12, weil diesmal nach jedem Byte drei Füllbytes eingefügt werden. Beim einem Pack-Modus von 8 (LWordAlignment) verändert sich nichts, da in dem Beispiel keine Variablen verwendet werden, die 8 Byte benötigen.

Der CX7031 arbeitet mit dem DWordAlignment (Pack-Modus 4), wenn Sie das Attribut nicht verwenden.

Weitere Informationen zum Attribut 'pack_mode' finden Sie unter: Attribut 'pack_mode'

8.9.3 Attribut 'TcCallAfterOutputUpdate'

Das Attribut-Pragma `TcCallAfterOutputUpdate` bewirkt, dass das IO-Update vor dem SPS-Zyklus stattfindet und nicht wie es standardmäßig eingestellt ist, nach dem SPS-Programm.

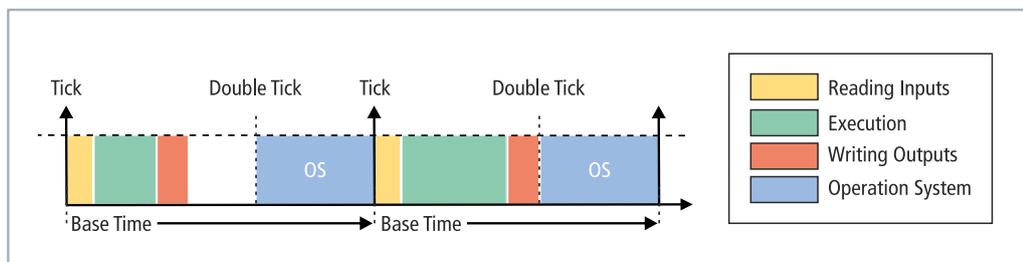


Abb. 41: Standard-Aufruf einer SPS-Task.

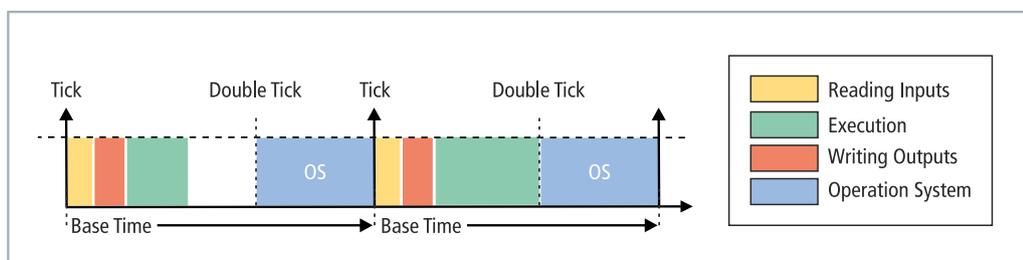


Abb. 42: Aufruf einer SPS-Task mit Attribut `TcCallAfterOutputUpdate`.

Diese Funktionalität kann für Projekte mit stark schwankenden Zykluszeiten eingesetzt werden. Bei Projekten mit stark schwankenden Zykluszeiten werden die Ausgänge, da sie nach dem SPS-Zyklus geschrieben werden, mal früher (kurze SPS-Zykluszeit) und mal später (lange SPS-Zykluszeit) geschrieben. Diese Schwankungen verursachen einen Jitter in den Ausgängen. Der Nachteil besteht darin, dass durch

das Attribut nicht ganz so schnell reagiert werden kann und immer ein Zyklus verloren geht. Sie müssen sich entscheiden, ob Sie schnell auf einen Eingang reagieren wollen (Standardeinstellung) oder ob sie lieber ein deterministisches Verhalten der Ausgänge haben wollen (setzen des Attributes).

Syntax: {attribute 'TcCallAfterOutputUpdate'}

Einfügeort: Dieses Attribut muss zu allen Programm POU's hinzugefügt werden, die nach dem Output Update aufgerufen werden sollen.

Beispiel:

Um das Verhalten zu verdeutlichen, brauchen Sie eine digitale Ausgangsklemme, beispielsweise eine EL2008 und ein Oszilloskop.

Schreiben Sie ein kleines SPS-Programm und verlinken Sie die Variable bOut mit einem digitalen Ausgang:

```
bOut:=not bOut;
```

Das SPS-Programm ist sehr einfach und verursacht keine Schwankungen. Der Puls wird auf dem Oszilloskop wie folgt dargestellt:

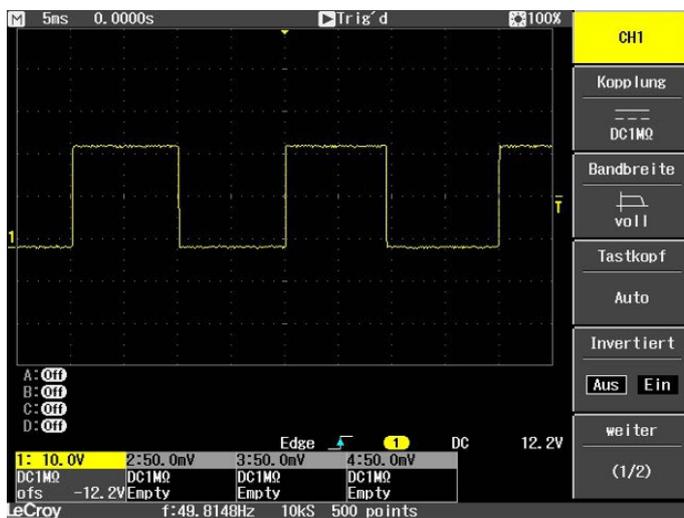


Abb. 43: Puls eines digitalen Ausgangs ohne Last.

Erweitern Sie nun das SPS-Programm um eine For-Schleife, um eine Programmlast zu erzeugen. Die verwendete mathematische Funktion spielt keine Rolle und soll nur eine Last erzeugen:

```
bOut:=not bOut;

IF bOut THEN
  For loop:=1 to 2000 do
    lrTest:=SIN(INT_TO_LREAL(loop)*3.14);
  END_FOR
END_IF
```

Wann immer der Ausgang auf TRUE gesetzt wird, wird die Schleife durchlaufen und eine Last erzeugt. Dadurch wird mehr Zeit für die Ausführung der SPS benötigt und der Ausgang später als sonst geschrieben. Beim nächsten Zyklus wird der Ausgang wieder auf FALSE gesetzt, die Schleife wird nicht durchlaufen und der Ausgang wird schneller auf FALSE gesetzt, da das SPS-Programm ohne For-Schleife wieder schneller fertig ist. Das Ergebnis ist, dass der Puls sehr viel kürzer ist.

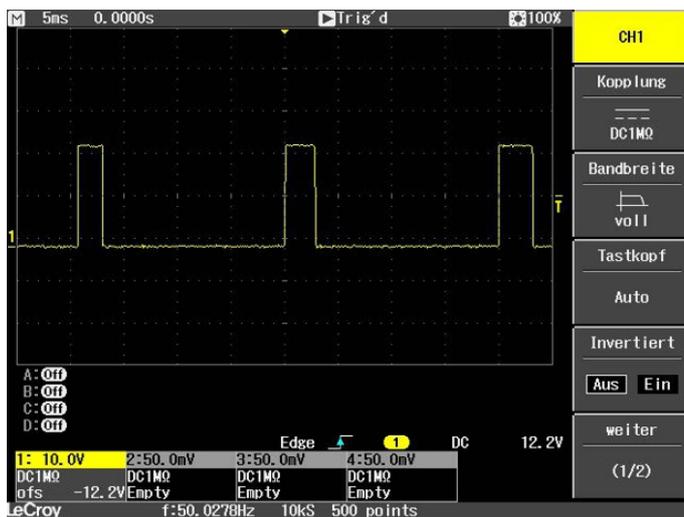


Abb. 44: Verkürzter Puls eines digitalen Ausgangs mit Last.

Wenn die For-Schleife statt dem TRUE beim FALSE aufgerufen wird, wird das Ergebnis invertiert.

```
bOut:=not bOut;

IF not bOut THEN
  For loop:=1 to 2000 do
    lrTest:=SIN(INT_TO_LREAL(loop)*3.14);
  END_FOR
END_IF
```

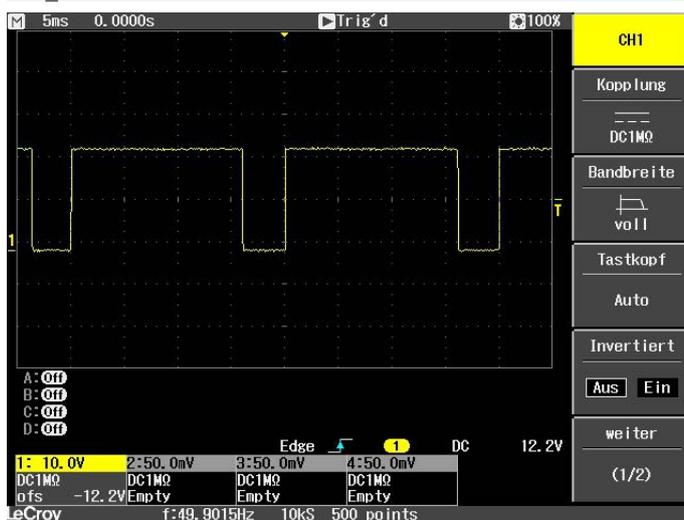


Abb. 45: Invertierte Darstellung eines digitalen Ausgangs.

Mit dem Attribut-Pragma `TcCallAfterOutputUpdate` ist der Puls konstant und ist unabhängig davon, wie lange die For-Schleife benötigt oder ob sie aufgerufen wird. Das Ganze funktioniert nur dann, wenn die SPS-Task nicht überschritten wird. Achten Sie also beim Reproduzieren des Beispiels auf die Überschreitungszähler der Task.

SPS-Programm mit unterschiedlichen Laufzeiten erkennen

Um SPS-Programme mit unterschiedlichen Laufzeiten zu erkennen, muss das SPS-Programm ergänzt werden. Im Online-View sind unterschiedliche Laufzeiten nicht erkennbar, da immer ein Mittelwert über mehrere Zyklen gebildet wird. Daher sind Ausreißer nur zu erkennen, wenn diese über der Taskzeit liegen. Liegen die Ausreißer noch innerhalb der Taskzeit, sind diese nicht ohne weiteres zu sehen.

Hierfür verwenden wir dann die Systemvariablen: `PlcTaskSystemInfo`

```
VAR
  bOut : BOOL;
  PlcTaskSystemInfo : PlcTaskSystemInfo;
  udiValue : ARRAY[0..19] of UDINT;
  Cnt : INT;
```

```

END_VAR

Program:
bOut:=not bOut;

IF bOut THEN
  For loop:=1 to 2000 do
    lrTest:=SIN(INT_TO_LREAL(loop)*3.14);
  END_FOR
END_IF

PlcTaskSystemInfo:=_TaskInfo[1];

udiValue[Cnt]:= PlcTaskSystemInfo.LastExecTime;
cnt:=cnt+1;
IF Cnt >19 THEN
  Cnt:=0;
END_IF
    
```

Mit dieser Programmerweiterung sieht man, dass das SPS-Programm mit For-Schleife 7,7 ms und ohne For-Schleife 1,1 ms benötigt. Die Angabe ist 100 ns pro Digit.

udiValue	ARRAY [0..19] OF U...	
udiValue[0]	UDINT	77728
udiValue[1]	UDINT	10713
udiValue[2]	UDINT	71049
udiValue[3]	UDINT	11065
udiValue[4]	UDINT	69882
udiValue[5]	UDINT	11027
udiValue[6]	UDINT	77084
udiValue[7]	UDINT	11939
udiValue[8]	UDINT	77494
udiValue[9]	UDINT	18527
udiValue[10]	UDINT	76724
udiValue[11]	UDINT	11043
udiValue[12]	UDINT	71519
udiValue[13]	UDINT	11406
udiValue[14]	UDINT	79004
udiValue[15]	UDINT	11118
udiValue[16]	UDINT	70745
udiValue[17]	UDINT	12007
udiValue[18]	UDINT	77761

Abb. 46: Ermittlung unterschiedlicher Lautzeiten beim SPS-Programm.

Die Messung deckt sich mit den Anzeigen auf dem Oszilloskop, auf denen erkennbar ist, dass ein Puls mal 6,5 ms länger bzw. 6,5 ms kürzer ist. Sie können die Bearbeitungszeit der For-Schleife messen ([Bearbeitungszeit im SPS-Programm messen](#) |> 101). Das Ergebnis dieser Messung wird sich mit den beobachtet Werten durch die Programmerweiterung decken, mit einer gewissen Ungenauigkeit und Jitter.

9 Fehlerbehandlung und Diagnose

9.1 Diagnose-LEDs

Anzeige	LED	Farbe	Beschreibung
	TC	Grün	TwinCAT ist im Run-Modus.
		Rot	TwinCAT ist im Stop-Modus. Zeigt zusätzlich Fehler beim Systemstart durch Fehlercode und Fehlerargument an (siehe Tabelle: TC-LED, Fehlerbeschreibung und Abhilfe). Die LED blinkt rot mit zwei unterschiedlichen Frequenzen.
		Blau	TwinCAT ist im Konfig-Modus.
		Gelb	Fehler oder Absturz der SPS.
		Keine Anzeige	Keine Fehlermeldung.
	FB	Grün	PROFIBUS im Datenaustausch
		Grün blinkt	PROFIBUS wartet auf Config-Daten
	DIAG	Rot	kein PROFIBUS konfiguriert
		Rot (nur DIAG)	Wenn beim Starten des CX70xx nur die DIAG-LED angeht, dann ist der Bootloader beschädigt und das Gerät muss eingeschickt werden.

Die TC-LED blinkt in einer festgelegten Frequenz und Reihenfolge und zeigt damit den Fehlercode und das Fehlerargument an.

Tab. 20: TC-LED, Reihenfolge und Bedeutung.

Reihenfolge	Bedeutung
Schnelles Blinken	Start der Sequenz
Erste langsame Sequenz	Fehlercode
Keine Anzeige	Pause, die LED ist aus
Zweite langsame Sequenz	Fehlerargument

Zählen Sie, wie oft die rote TC-LED blinkt, um den Fehlercode und das Fehlerargument zu ermitteln.

Tab. 21: TC-LED, Fehlerbeschreibung und Abhilfe.

Fehlercode	Fehlerargument	Beschreibung	Abhilfe
1	1	MicroSD-Karte nicht erkannt	MicroSD-Karte prüfen. Image nicht in Ordnung. Installieren Sie ein neues Image auf der MicroSD-Karte.
	2	Card init failed - preloader	
	3	No partition found - preloader	
	4	Filesystem mount failed - preloader	
	5	Card init failed - loader	
	6	No partition found - loader	
	7	Filesystem mount failed - loader	
2	1	Loader not found	
	2	Loader file invalid (checksum, size, read error)	
	3	TC dll not found	
	4	TC dll checksum error	
	5	EEPROM file missing or invalid	
	6	TcOsSys.dll version not compatible with loader	

Fehlercode	Fehlerargument	Beschreibung	Abhilfe
3	1	Rbf not found	
	2	CCAT 1 init failed	
	3	CCAT 2 init failed	
	4	CCAT EEPROM writing failed	
	5	CCAT 1 EEPROM reloaded failed	
	6	CCAT 2 EEPROM reloaded failed	
4	1	Peripheral not working	Hardware defekt, tauschen sie den CX
	2	Voltage Vo not reached	
	3	Low speed external oscillator not running	
	4	High speed external oscillator not running	
	5	Flash failed	
	6	Device overclocked (old Hardware)	
5	5	RAM error detected	

9.1.1 K-Bus

Die angeschlossenen Busklemmen werden vom Netzteil auf Fehler überprüft. Die rote LED „K-BUS ERR“ ist aus, wenn keine Fehler vorhanden sind. Die rote LED „K-BUS ERR“ blinkt, wenn Fehler im Bereich der Busklemmen vorhanden sind.

Tab. 22: Diagnose-LEDs im K-Bus-Modus.

Anzeige	LED	Bedeutung
<p>Us Up Link / Act IO ERR IO Cnt / DI1 DI2 / Cnt DI3 DI4 DI5 DI6 AI1 / DI7 DI8 / AI2 DO1 DO2 PWM1 / DO3 DO4 / PWM2</p>	Us 24V	Spannungsversorgung für CPU-Grundmodul. Die LED leuchtet grün bei korrekter Spannungsversorgung.
	Up 24V	Spannungsversorgung für Klemmenbus. Die LED leuchtet grün bei korrekter Spannungsversorgung.
	K-BUS RUN	Diagnose K-Bus. Die grüne LED leuchtet, um den fehlerfreien Betrieb anzuzeigen. Fehlerfrei bedeutet, dass auch die Kommunikation mit dem Feldbussystem fehlerfrei läuft.
	K-BUS ERR	Diagnose K-Bus. Die rote LED blinkt zur Fehleranzeige. Die rote LED blinkt mit zwei unterschiedlichen Frequenzen.

Durch die Frequenz und Anzahl des Blinkens kann der Fehlercode und das Fehlerargument ermittelt werden. Ein Fehler wird durch die LED „K-BUS ERR“ in einer festen Reihenfolge angezeigt.

Tab. 23: K-BUS ERR LED, Reihenfolge der Fehleranzeige durch die LED.

Reihenfolge	Bedeutung
Schnelles Blinken	Start der Sequenz
Erste langsame Sequenz	Fehlercode
Keine Anzeige	Pause, die LED ist aus
Zweite langsame Sequenz	Fehlerargument

Zählen Sie, wie oft die rote LED K-BUS ERR blinkt, um den Fehlercode und das Fehlerargument zu ermitteln. Bei dem Fehlerargument zeigt die Anzahl der Impulse die Position der letzten Busklemme vor dem Fehler an. Passive Busklemmen, wie zum Beispiel eine Einspeiseklemme, werden nicht mitgezählt.

Tab. 24: K-BUS ERR LED, Fehlerbeschreibung und Abhilfe.

Fehlercode	Fehlerargument	Beschreibung	Abhilfe
Ständiges, konstantes Blinken		EMV Probleme.	<ul style="list-style-type: none"> Spannungsversorgung auf Unter- oder Überspannungsspitzen kontrollieren. EMV-Maßnahmen ergreifen. Liegt ein K-Bus-Fehler vor, kann durch erneutes Starten (Aus- und Wiedereinschalten des Netzteils) der Fehler lokalisiert werden.
3 Impulse	0	K-Bus-Kommandofehler.	<ul style="list-style-type: none"> Keine Busklemme gesteckt. Eine der Busklemmen ist defekt, angehängte Busklemmen halbieren und prüfen ob der Fehler bei den übrigen Busklemmen noch vorhanden ist. Dieses Vorgehen wiederholen, bis die defekte Busklemme lokalisiert ist.
4 Impulse	0	K-Bus-Datenfehler, Bruchstelle hinter dem Netzteil.	Kontrollieren, ob die Busendklemme 9010 gesteckt ist.
	n	Bruchstelle hinter Busklemme n.	Prüfen, ob die Busklemme n+1 hinter dem Netzteil richtig gesteckt ist, gegebenenfalls tauschen.
5 Impulse	n	K-Bus-Fehler bei Register-Kommunikation mit Busklemme n.	Busklemme an Stelle n tauschen.
6 Impulse	0	Fehler bei der Initialisierung.	Embedded-PC tauschen.
	1	Interner Datenfehler.	Hardware-Reset des Embedded-PCs (aus- und wieder einschalten).
	8	Interner Datenfehler.	Hardware-Reset des Embedded-PCs (aus- und wieder einschalten).
7 Impulse	0	Prozessdatenlängen der Soll- und Ist-Konfiguration stimmen nicht überein.	Konfiguration und Busklemmen auf Konsistenz prüfen.

Bei manchen Fehlern geht die LED „K-BUS ERR“ nicht aus, obwohl der Fehler beseitigt wurde. Schalten Sie die Spannungsversorgung für das Netzteil aus und wieder ein, damit die LED nach der Fehlerbeseitigung ausgeschaltet wird.

State-Variable

In TwinCAT gibt es unter dem Buskoppler die Variable State, für die K-Bus-Diagnose.

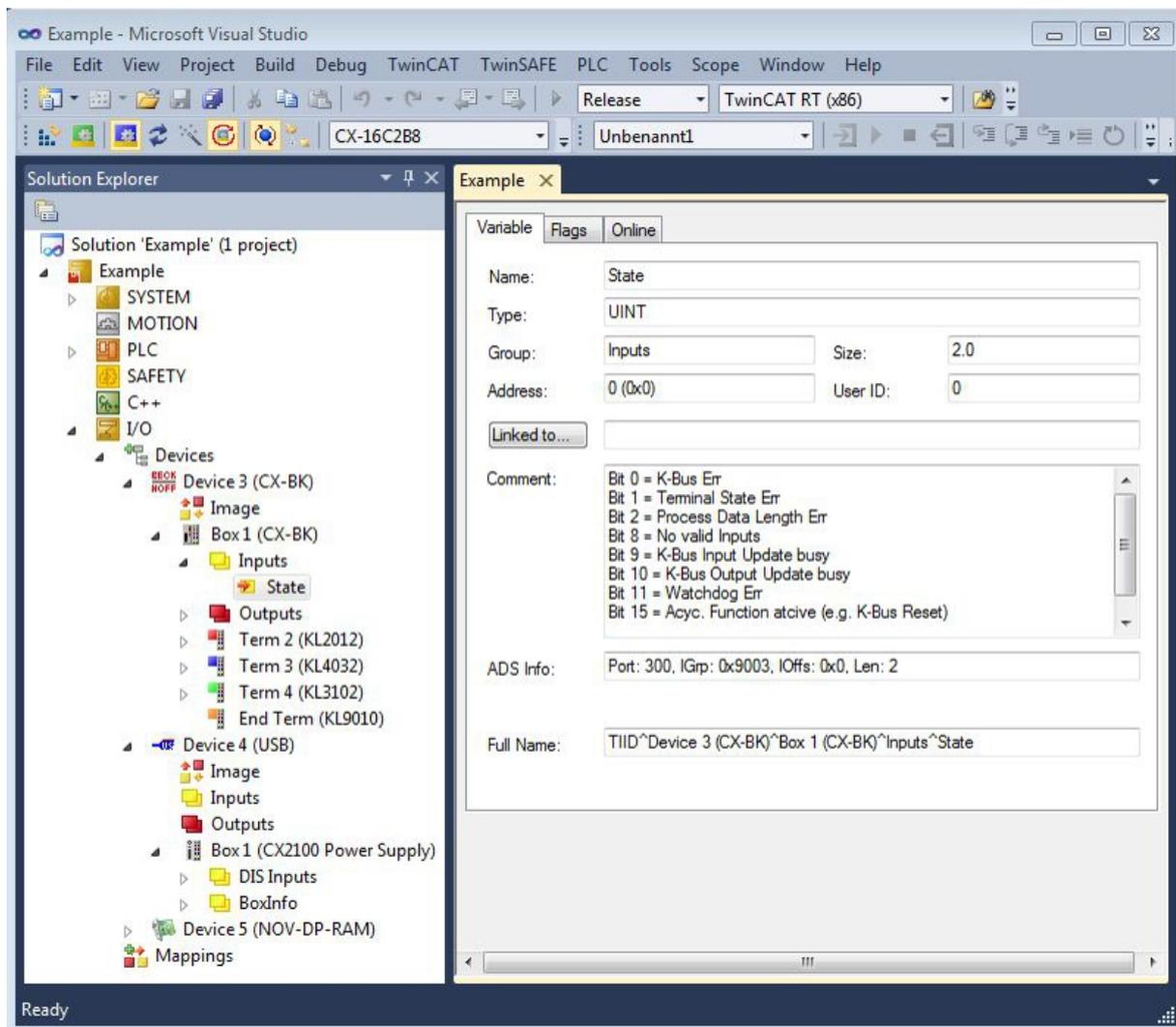


Abb. 47: Status-Variable für Fehlerbehandlung und Diagnose unter TwinCAT.

Ist der Wert „0“ so arbeitet der K-Bus synchron und ohne Fehler. Sollte der Wert <> „0“ sein, kann ein Fehler vorliegen. Es kann aber auch nur ein Hinweis sein, das zum Beispiel der K-Bus-Zyklus länger dauert, als die verwendete Task. Damit ist er dann nicht mehr synchron zu der Task. Die Taskzeit sollte schneller als 100 ms sein. Wir empfehlen eine Taskzeit kleiner 50 ms. Typischerweise liegt die K-Bus-Update-Zeit zwischen einer und fünf ms.

Tab. 25: Beschreibung der Werte bei der State-Variable.

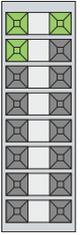
Bit	Beschreibung
Bit 0	K-Bus-Fehler.
Bit 1	Klemmenkonfiguration hat sich seit dem Start geändert.
Bit 2	Prozessabbildlängen stimmen nicht überein.
Bit 8	(noch) keine gültigen Eingänge.
Bit 9	K-Bus ist im Inputupdate noch nicht fertig.
Bit 10	K-Bus ist im Output-Update noch nicht fertig.
Bit 11	Watchdog.
Bit 15	azyklische K-Bus-Funktion aktiv (z.B. K-Bus-Reset).

Liegt ein K-Bus-Fehler vor, kann dieser über den Funktionsbaustein IOF_DeviceReset (in der TcloFunctions.lib) zurückgesetzt werden.

9.1.2 E-Bus

Die angeschlossenen EtherCAT-Klemmen werden vom Netzteil überprüft. Im E-Bus-Modus leuchtet die LED „Link/Act IO“. Wenn Daten übertragen werden, blinkt die LED „Link/Act IO“.

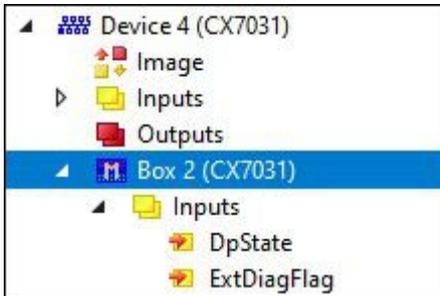
Tab. 26: Diagnose-LEDs im K-Bus-Modus.

Anzeige	LED	Bedeutung	
 <p> Us Link / Act IO Cnt / DI1 DI3 DI5 AI1 / DI7 D01 PWM1 / D03 </p> <p> Up ERR IO DI2 / Cnt DI4 DI6 DI8 / AI2 D02 D04 / PWM2 </p>	Us	Spannungsversorgung für CPU-Grundmodul. Die LED leuchtet grün bei korrekter Spannungsversorgung.	
	Up	Spannungsversorgung für Klemmenbus. Die LED leuchtet grün bei korrekter Spannungsversorgung.	
	Link/Act IO	aus	E-Bus nicht angeschlossen.
		an	E-Bus angeschlossen / Kein Datenverkehr.
blinkt		E-Bus angeschlossen / Datenverkehr auf dem E-Bus.	

9.2 PROFIBUS-Diagnose

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 Adresse 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 Adresse 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.

Wert	Beschreibung
15	Station has no resources: 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 parametrisiert und konfiguriert, hat aber noch kein Data_Exchange-Telegramm empfangen.
129	Slave waiting for configuration: Der Slave wurde parametrisiert, hat aber noch kein Chk_Cfg-Telegramm empfangen.
130	Slave waiting for parameter: Slave wurde noch nicht parametrisiert 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.

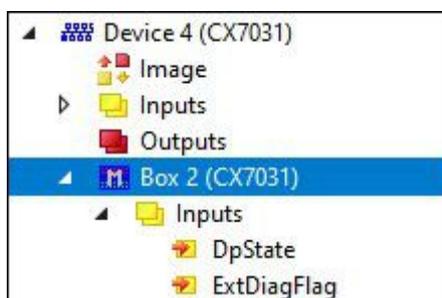
ExtDiagFlag:

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

9.2.1 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 = Adresse 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
Slave-Statistiken	
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 Max Retry-Limit (s. Dialog Bus-Parameter) 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 MaxRetryLimit 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 DPV1-Fehlercodes
27-31	reserviert für Erweiterungen
ab 32	DP-Diagnosedaten

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

Offset	Bedeutung
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: Adresse 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.

9.2.2 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)



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: Adresse des Masters, der mit dem Slave Datenaustausch macht
0x04,0x05	IdentNumber
ab 0x06	gerätespezifische Diagnosedaten (Extended DiagData)

Gerätespezifische Diagnosedaten

● **Übertragung von anwenderspezifischen Diagnose-Daten**

i 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

9.3 Diagnose der Multifunktions-I/Os

Dieses Kapitel beschreibt die Diagnosemöglichkeiten der Multifunktions-I/O-Kommunikation. Das ist beispielsweise dann wichtig, wenn die 24-V-Spannungsversorgung für die Multifunktions-I/Os ausfällt oder die Sicherung ausgelöst wird.

Statusvariable

Die Statusvariable `state` kann für Diagnosezwecke eingesetzt werden. Im Normalzustand nimmt die Statusvariable den Wert `0x__8` (OP, Operational) an und zeigt damit an, dass alles Fehlerfrei ist.

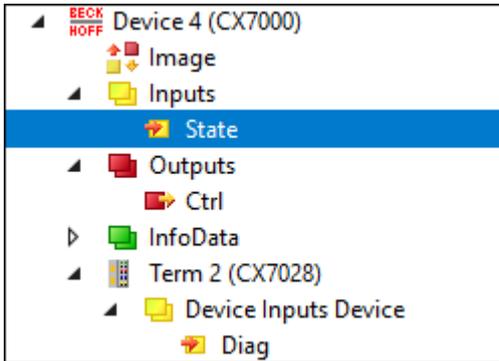


Abb. 48: Multifunktions-I/O Statusvariable.

Die folgende Tabelle zeigt, welche Werte die Variable annehmen kann:

Wert	Bedeutung
0x__1	Slave in 'INIT' state
0x__2	Slave in 'PREOP' state
0x__3	Slave in 'BOOT' state
0x__4	Slave in 'SAFEOP' state
0x__8	Slave in 'OP' state
0x001_	Slave signals error
0x002_	Invalid vendorId, productCode... read
0x004_	Initialization error occurred
0x010_	Slave not present

Sollte es zu einem Ausfall der Spannungsversorgung kommen, gehen die Multifunktions-I/Os nicht automatisch wieder in den Datenaustausch. Dafür müssen die Multifunktions-I/Os zurückgesetzt werden. Ein Funktionsbaustein, mit dem die Multifunktions-I/Os zurückgesetzt werden können, ist der Funktionsbaustein `FB CX70xx ResetOnBoardIO` [► 112].

Hinweis: Sind in der SPS noch Ausgänge gesetzt, dann werden die Ausgänge der Multifunktions-I/Os sofort wieder aktiv, sobald die Multifunktions-I/Os mit dem Funktionsbaustein zurückgesetzt werden.

Weitere Diagnosevariablen

Die Diagnosevariablen `Diag` und `TxPDO State` sind derzeit nicht in Gebrauch und für zukünftige Verwendung reserviert. Die Variable `Input cycle counter` hingegen erhöht sich mit jedem Zyklus und zeigt die Anzahl der I/O-Zyklen an, die mit den Multifunktions-I/Os ausgetauscht werden. Sobald die Variable nicht mehr inkrementiert wird, werden keine I/O-Zyklen mehr mit den Multifunktions-I/Os ausgetauscht.

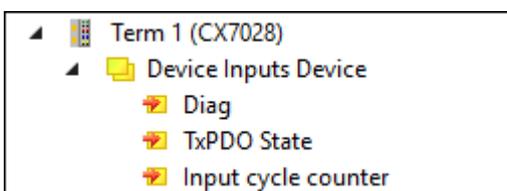


Abb. 49: Weitere Diagnosevariablen für Multifunktions-I/Os

Variable	Bedeutung
Diag	Reserviert, wird aktuell nicht verwendet.
TxPDO State	Reserviert, wird aktuell nicht verwendet.
Input cycle counter	Wird mit jedem Zyklus um 1 inkrementiert. Wenn dieser Zähler stehen bleibt, dann werden keine I/O-Zyklen mehr mit den Multifunktions-I/Os ausgetauscht.

9.4 Speicherauslastung

Der CX7031 hat 32 MB Arbeitsspeicher, der von der Firmware (TwinCAT/RTOS) und von TwinCAT (TwinCAT-Speicher) verwendet wird. Der TwinCAT-Speicher unterteilt sich weiter in den Router-Speicher und den SPS-Speicher. Der Router-Speicher wird für die ADS-Kommunikation und der SPS-Speicher für das eigentliche SPS-Programm inklusive TcConfiguration, Mapping und Daten verwendet.

Dem CX7031, stehen 19,1 MB TwinCAT-Speicher zur Verfügung. Weil die Größe des Speichers begrenzt ist, ist es wichtig die Speicherauslastung zu kontrollieren und ihr SPS-Projekt bei einer Überschreitung anzupassen.

Router-Speicher

Sie können zum einen die Größe des Router-Speichers in TwinCAT anpassen und abhängig von der tatsächlich verwendeten ADS-Kommunikation, einen kleineren Router-Speicher einstellen.

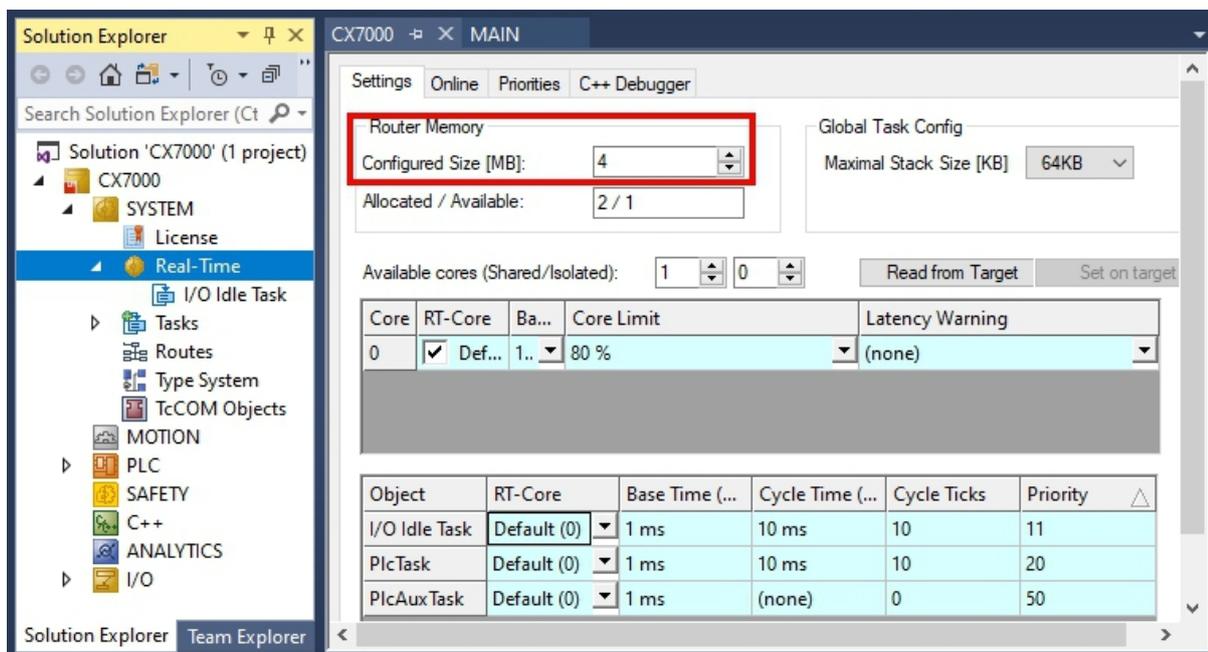


Abb. 50: Einstellungen für Router-Speicher im TwinCAT System Manager.

Standardmäßig wird in TwinCAT ein Wert von 32 MB eingetragen, der beim CX7031 wiederum auf 9 MB begrenzt wird, wegen des kleinen Arbeitsspeicher beim CX7031. Ein Router-Speicher von 9 MB ist in der Regel viel zu groß für eine Kleinststeuerung. Beim CX7031 ist ein Router-Speicher von 4 MB empfehlenswert und kann unter Umständen sogar noch kleiner gewählt werden, wenn wenig bis gar keine ADS-Kommunikation verwendet wird. Ein Router-Speicher von mindestens 1 MB sollte jedoch eingehalten und nicht unterschritten werden. Wieviel Router-Speicher verwendet wird, lässt sich mit dem Funktionsbaustein FB_GetRouterStatusInfo oder alternativ mit dem Beckhoff Device Manager kann ermitteln.

Beachten Sie, dass der Router-Speicher erst mit einem Power Off/On des CX7031 neu angelegt wird. Ein TwinCAT-Neustart reicht nicht aus. Als Faustregel gilt: Je kleiner der Router-Speicher für die ADS-Kommunikation gewählt wird, desto größer kann die Applikation sein, also das SPS-Programm, TcConfiguration, Mapping und Daten.

Speicherauslastung bestimmen

Mit dem Funktionsbaustein FB_GetRouterStatusInfo oder alternativ mit dem Beckhoff Device Manager kann ermittelt werden, wie groß der Speicherbedarf des Router-Speichers ist.

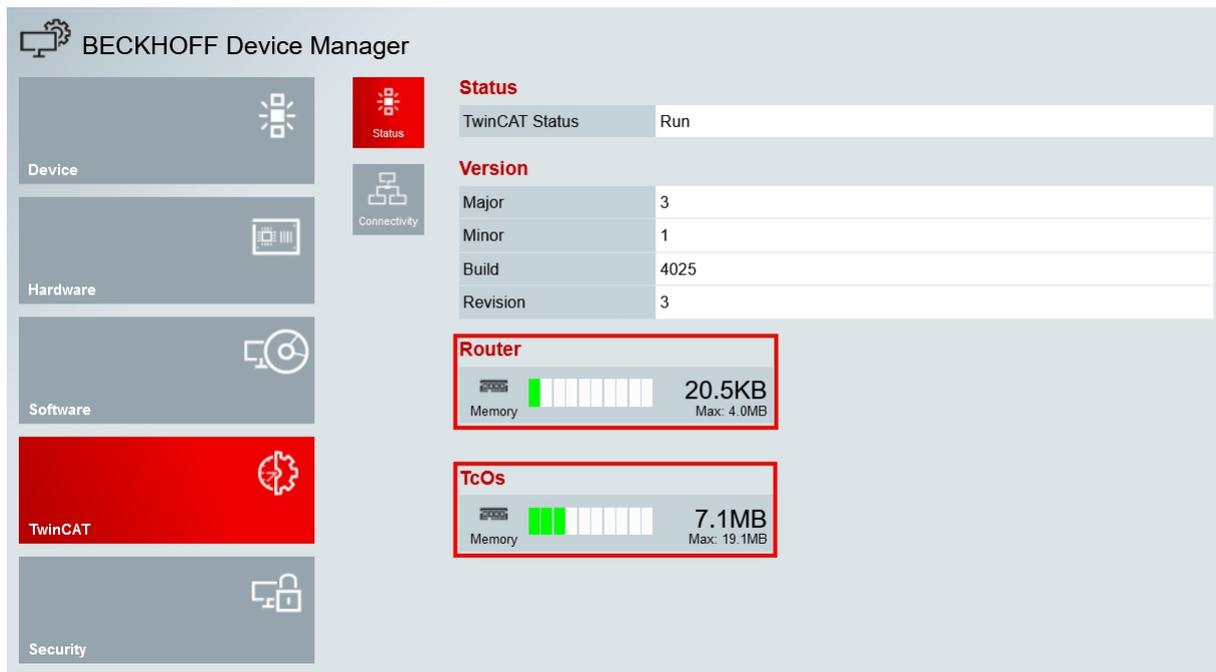


Abb. 51: Auslastung des Router-und TwinCAT-Speichers.

Über die Anzeige **Router** kann der Speicherbedarf des Router-Speichers bestimmt werden. In diesem Beispiel werden 20,5 kB von maximal 4 MB belegt. Die Anzeige **TcOs** zeigt den gesamten Speicherverbrauch des TwinCAT-Speichers inklusive Router-Speicher und SPS-Programm an. In diesem Beispiel werden insgesamt 7,1 MB belegt.

Mit Hilfe dieser Anzeige, kann auch die Größe des SPS-Programms berechnet werden, da der Router-Speicher mit 4 MB fest definiert und Teil des TwinCAT-Speichers ist. Zieht man die 4 MB von 7,1 MB ab, so belegt das SPS-Programm 3,1 MB.

Speicherreserve

Da in diesem Beispiel beim TwinCAT-Speicher 7,1 MB von 19,1 MB belegt werden, bleibt eine Reserve von 12 MB für das SPS-Programm. Beachten Sie, dass für ein Online-Change in TwinCAT kurzzeitig mehr Speicher gebraucht wird. Wenn Sie die Funktion Online-Change verwenden wollen, ist es ratsam immer eine gewisse Reserve vorzuhalten. Im extremsten Fall kann das Doppelte des aktuell verbrauchten SPS-Programms erforderlich sein, um ein Online-Change auszuführen. In TwinCAT wird eine Fehlermeldung angezeigt, sobald nicht genügend Speicher für den Online-Change zur Verfügung steht.

9.5 Echtzeit und CPU-Auslastung

Für die einwandfreie Funktionsweise des CX7031 ist es wichtig, die CPU-Auslastung und die Einhaltung der Echtzeit im Blick zu behalten. Andernfalls arbeitet der CX7031 bei einer Überlastung nicht mehr zuverlässig. Beachten Sie, dass bei einer Überlastung auch die Auslastungsanzeige betroffen ist und keine aktuellen Werte mehr liefert. So kann fälschlicherweise eine Auslastung von 40 % angezeigt werden, die SPS aber bereits nicht mehr in Echtzeit arbeiten und das System überlastet sein. Sie sollten sich also bei einer Kleinststeuerung schrittweise an die Belastungsgrenze herantasten.

Was versteht man in diesem Zusammenhang unter Echtzeit? Die SPS arbeitet standardmäßig zyklussynchron, das bedeutet, dass immer eine Taskzeit definiert und zu einem festen Zeitpunkt aufgerufen wird. Die SPS arbeitet zyklussynchron, wenn die Taskzeit nicht überschritten wird. Wenn Sie beispielsweise eine Taskzeit von 10 ms definieren und die SPS nur 2 ms für die Bearbeitung braucht, ist die gewählte Taskzeit in Ordnung und die SPS arbeiten zyklussynchron.

Auch wenn Sie die Echtzeit nicht brauchen, ist es empfehlenswert die Echtzeit einzuhalten, weil es sonst zu negativen Effekten kommen kann. Das können Verbindungsprobleme oder Probleme mit Subsystemen wie K-Bus oder EtherCAT sein. Folgende Schritte können Sie durchführen, um zu überprüfen, ob der CX7031 optimal eingestellt oder eher überlastet ist:

- Exceed-Counter beobachten.
- CPU-Auslastung kontrollieren.

Exceed-Counter beobachten

Sobald die SPS nicht mehr zyklussynchron arbeitet und die definierte Taskzeit überschritten wird, wird der Exceed-Counter hochgezählt. Im Idealfall sollte der Zählerstand Null sein.

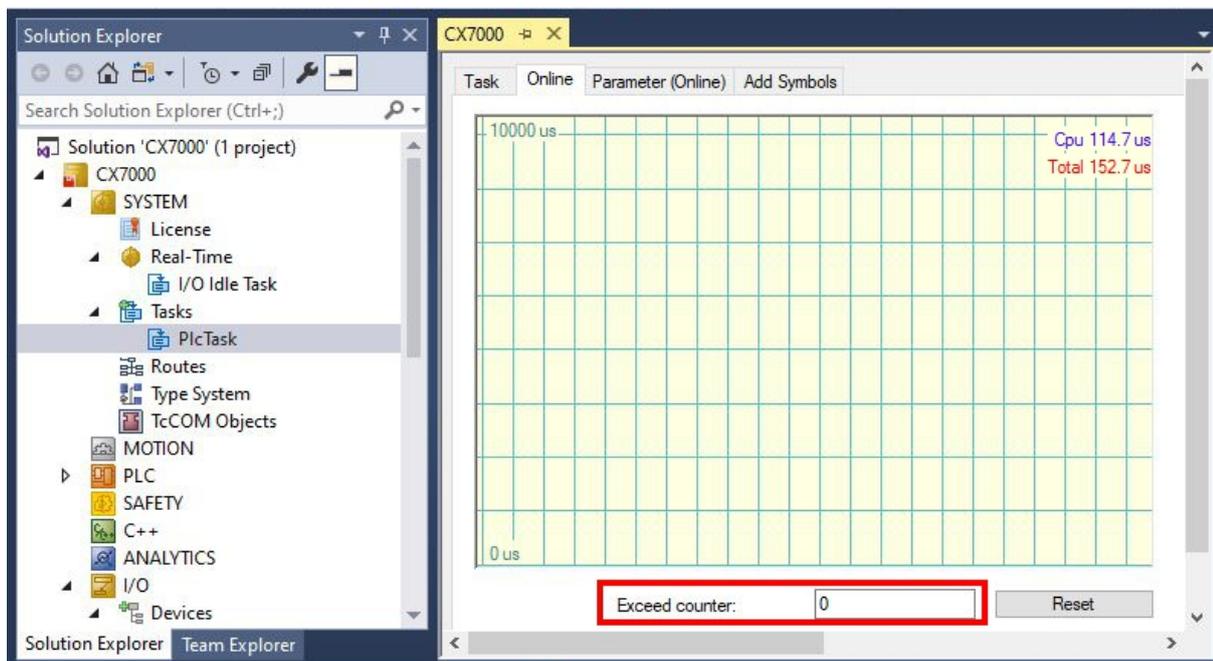


Abb. 52: Anzeige des Exceed-Counters in TwinCAT.

Es kann vorkommen, dass der Exceed-Counter beim Start der SPS hochgezählt wird, weil die SPS beispielsweise zum ersten Mal aufgerufen wird oder bestimmte Bestandteile initialisiert werden. Beobachten Sie den Exceed-Counter über einen Zeitraum von mehreren Stunden. Erst wenn der Exceed-Counter über einen längeren Zeitraum nicht mehr hochgezählt wird, kann von einem stabilen Zustand gesprochen werden.

CPU-Auslastung kontrollieren

In TwinCAT wird unter Realtime und der Registerkarte Online, die CPU-Auslastung angezeigt. Kontrollieren Sie den Wert, um zu ermitteln, ob Sie zusätzlichen Programmcode ausführen oder die Taskzeit verkürzen können.

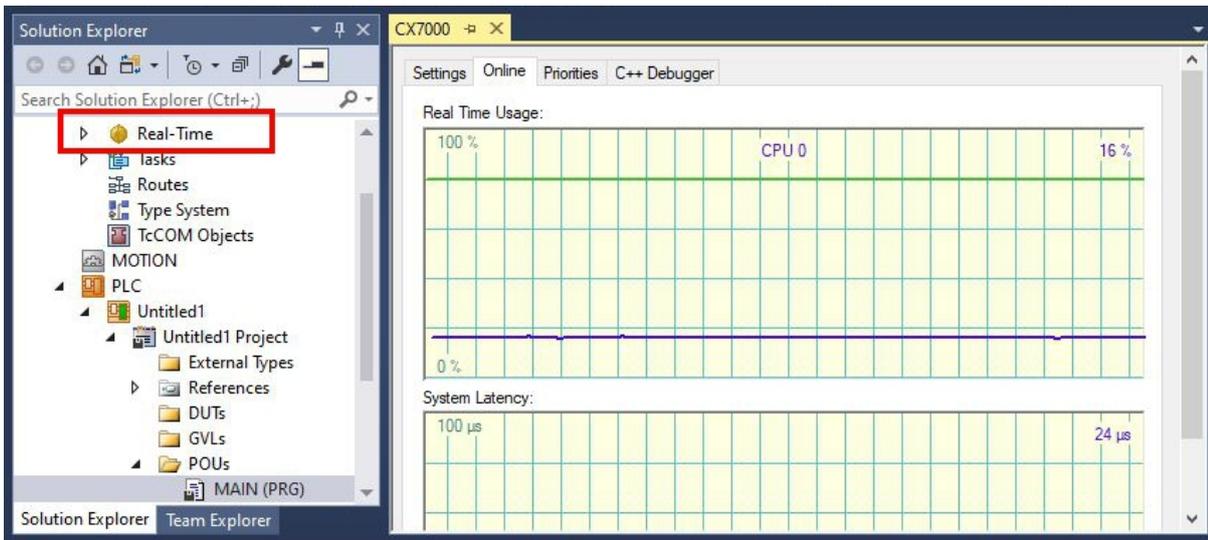


Abb. 53: Anzeige der CPU-Auslastung in TwinCAT

Die hellgrüne Linie zeigt das voreingestellte CPU-Limit an. Wenn die Auslastung $\geq 65\%$ beträgt, ist der CX7031 bereits gut ausgelastet und es sollte kein weiterer Programmcode ausgeführt oder die Taskzeit verkürzt werden. Sie sollten nicht bis an die Grenzen gehen und den CX7031 voll auslasten.

Maßnahmen bei einer Überlastung

Wird eine Überlastung mit Hilfe der gezeigten Schritte festgestellt, kann die Auslastung durch eine verbesserte Programmierung oder eine Erhöhung der Taskzeit reduziert werden. Um Stellen im Programmcode mit langen Bearbeitungszeiten zu finden, kann das Beispiel unter: [Bearbeitungszeit im SPS-Programm messen](#) [101] genutzt werden.

Einen Einfluss auf die Echtzeit, hat auch das gewählte Klemmensystem. Abhängig von der Anzahl der Klemmen, kann beispielsweise der K-Bus zusätzlich mehrere Millisekunden beanspruchen und muss bei der Wahl der Taskzeit berücksichtigt werden. Es kann durchaus sein, dass bei einer eingestellten Taskzeit von 10 ms das SPS-Programm nur 5 ms benötigt, der Exceed-Counter aber trotzdem hochzählt. Das liegt daran, dass der K-Bus mehr als 5 ms für die Bearbeitung benötigt und die Taskzeit von 10 ms inklusive SPS-Programm und K-Bus überschritten wird. Dieses Problem kann gelöst werden, wenn die Anzahl der Klemmen reduziert wird oder die Taskzeit erhöht wird.

Standardmäßig ist die Echtzeit auf 80 % eingestellt. Das ist bereits der maximale Wert und eine Erhöhung auf 90 % ist mit einer Erhöhung auf 100 % gleichzusetzen.

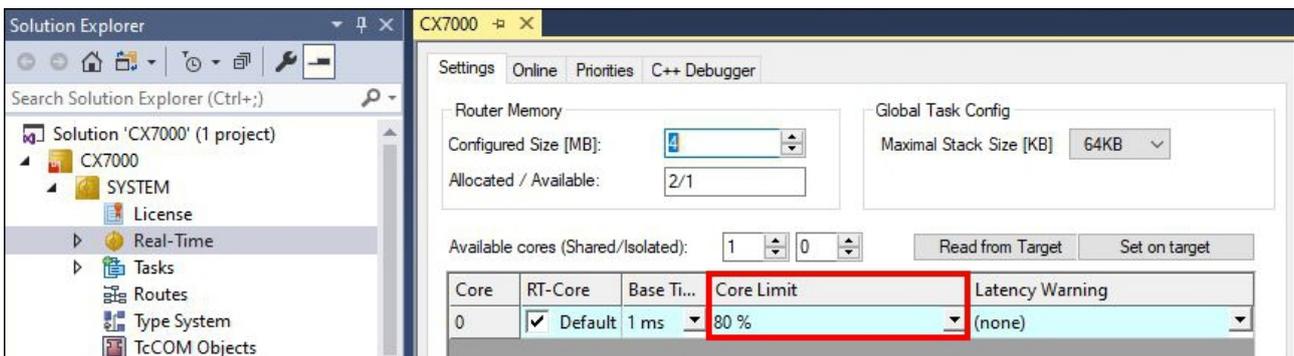


Abb. 54: Einstellung der Echtzeitauslastung in TwinCAT.

TwinCAT würde dann die gesamte CPU-Leistung beanspruchen und Dienste, die das Betriebssystem bedient, nicht mehr oder unzureichend funktionieren. Wenn Sie die Echtzeitauslastung auf 90 % erhöhen, sollten Sie sich der möglichen Folgen für das Betriebssystem bewusst sein.

10 Technische Daten

Tab. 27: Technische Daten, Abmessungen und Gewicht.

	CX7031
Abmessungen (B x H x T)	49 mm x 100 mm x 73 mm
Gewicht	142 g

Tab. 28: Technische Daten, allgemeine Daten.

Technische Daten	CX7031
Prozessor	Arm® Cortex®-M7, 480 MHz
Anzahl Kerne	1
Flash-Speicher	512 MB MicroSD (optional 16 GB)
Arbeitsspeicher	32 MB SDR (intern, nicht erweiterbar)
Anzahl Eingänge	8 Multifunktionseingänge (24 V DC)
Anzahl Ausgänge	4 Multifunktionsausgänge (24 V DC, 0,5 A, 1-Leitertechnik)
NOVRAM	4 kB
Schnittstellen	1 x RJ45 10/100 MBit/s, 1x USB (max 12 Mbit/s, max 100 mA)
Kühlung	passiv
Businterface	D-Sub-Buchse, 9-polig, 1 x PROFIBUS-Slave
Übertragungsrate	9,6 kBaud...12 MBaud
Diagnose-LED	1 x TC-Status, 1 x WD-LED, 1 x ERR-LED
Uhr	interne, kondensatorgepufferte Realtime-Clock für Zeit und Datum (Speicher > 21 Tage)
Betriebssystem	TwinCAT/RTOS
Steuerungssoftware	TwinCAT-3-Runtime (XAR)
Spannungsversorgung	24 V _{DC} (-15 %/+20 %)
max. Leistungsaufnahme	< 2 W (max. 12 W mit E-Bus/K-bus)
Max. Leistungsaufnahme E-Bus/K-Bus	7,5 W
Enthaltene TwinCAT 3 Functions	TC1000 TwinCAT 3 ADS, TC1100 TwinCAT 3 I/O, TC1200 TwinCAT 3 PLC, TF4100 TwinCAT 3 Controller Toolbox, TF4110 TwinCAT 3 Temperature Controller, TF6255 TwinCAT 3 Modbus RTU, TF6340 TwinCAT 3 Serial Communication, TF6701 TwinCAT 3 IoT Communication (MQTT), TF6730 TwinCAT 3 IoT Communicator
Zulassungen	CE, UL

Tab. 29: Technische Daten, I/O-Klemmen.

Technische Daten	CX7031
I/O-Anschluss	via Netzsteckklemme (E-Bus oder K-Bus, automatische Erkennung)
Stromversorgung für I/O-Klemmen	max. 1,5 A (Einbaulage beliebig, Temp. -25...45 °C) max. 1,3 A (Einbaulage horizontal, Temp. -25...55 °C) max. 1 A (Einbaulage beliebig, Temp. -25...55 °C) max. 1 A (Einbaulage horizontal, Temp. -25...60 °C)
Strombelastung Powerkontakte	max. 10 A
Prozessdaten K-Bus	max. 512 Byte In und 512 Byte Out
max. Anzahl der Klemmen (K-Bus)	64 (255 mit K-Bus-Verlängerung)
Prozessdaten E-Bus	max. 4 kByte In und 4 kByte Out
max. Anzahl der Klemmen (E-Bus)	bis zu 65534 Klemmen.

Tab. 30: Technische Daten, Umgebungsbedingungen.

Technische Daten	CX7031
Umgebungstemperatur im Betrieb	-25° C ... +60° C
Umgebungstemperatur bei Lagerung	-40° C ... +85° C siehe Hinweise unter: Transport und Lagerung [▶ 12]
Relative Feuchte	95 % ohne Betauung
Schwingungsfestigkeit	gemäß EN 60068-2-6
Schockfestigkeit	gemäß EN 60068-2-27
EMV-Festigkeit	gemäß EN 61000-6-2
EMV-Aussendung	gemäß EN 61000-6-4
Schutzart	IP20

Tab. 31: Technische Daten, Ethernet-Schnittstelle X001.

Technische Daten	CX7031
Übertragungsmedium	4 x 2 Twisted-Pair-Kupferkabel Kategorie 5 (100 MBit/s)
Leitungslänge	100 m vom Switch bis zum CX7031
Übertragungsrate	10/100 MBit/s
Topologie	sternförmige Verkabelung
Protokolle	alle nicht Echtzeitfähigen Protokolle, die auf TCP oder UDP basieren und keine Echtzeiterweiterung benötigen

Tab. 32: Technische Daten, PROFIBUS-Schnittstelle X003.

Technische Daten	CX7031
Feldbus	PROFIBUS DP, DP-V1
Übertragungsrate	9,6 k; 19,2 k; 93,75 k; 187,5 k; 500 k; 1,5 M; 3 M; 6 M; 12 MBaud
Businterface	1 x D-Sub-Buchse, 9-polig
Erweiterbares Prozessabbild	bis zu 3 virtuelle Slaves zusätzlich
max. Prozessabbild	4 Slaves x (240 Byte In / 240 Byte Out)
Autobaud	Ja
Protokoll	
PROFIBUS-DP Slave	Ja
PROFIBUS-DP (virtueller Slave)	4 (3 virtuelle PROFIBUS Nodes)
ADS-Interface	ja (nur über Ethernet)
Dienste	
PROFIBUS	DP
DPV1	Ja
Diagnose/Status/Alarm	
TC LED	Ja, grün/rot
BF LED	Ja, grün/rot
DIA LED	Ja, grün/rot
Diagnosemeldungen	Ja

11 Anhang

11.1 Komponenten Dritter

Dieses Gerät enthält Software von Beckhoff und Dritten.
Bitte beachten Sie die auf dem Speichermedium enthaltene Lizenzdatei.

11.2 Zubehör

Tab. 33: MicroSD-Karten.

Bestellnummer	Beschreibung
CX1900-0122	512-MB-MicroSD-Karte
CX1900-0132	16-GB-MicroSD-Karte

Tab. 34: Weitere Ersatzteile.

Bestellnummer	Beschreibung
ZB8701	Schlitzschraubendreher 2,0 x 40 mm, HD-Klemmen

11.3 Zertifizierungen

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.

Tabellenverzeichnis

Tab. 1	Abmessungen und Gewicht.	12
Tab. 2	Legende zum Aufbau des CPU-Grundmoduls.....	14
Tab. 3	Informationen auf dem Typenschild.	15
Tab. 4	Ethernet-Schnittstelle X001, PIN-Belegung.	16
Tab. 5	Maximaler E-Bus/K-Bus-Strom abhängig von Einbaulage und Umgebungstemperatur.....	22
Tab. 6	Legende zum Anschlussbeispiel.....	26
Tab. 7	Erforderliche Leiterquerschnitte und Abisolierlängen.....	27
Tab. 8	ZK1031-6xxx-1xxx PROFIBUS-Leitungen, PUR, schleppkettentauglich.....	32
Tab. 9	ZB3xxx PROFIBUS-Leitungen, Meterware.....	32
Tab. 10	ZS1xxx PROFIBUS-Steckverbinder.....	32
Tab. 11	Technische Daten, Multifunktions-I/Os als digitale Eingänge.	35
Tab. 12	Technische Daten, Multifunktions-I/Os als digitale Ausgänge.	36
Tab. 13	Technische Daten, Multifunktions-I/Os im Zähler-Modus.	39
Tab. 14	Technische Daten, Multifunktions-I/Os im Encoder-Modus.	45
Tab. 15	Technische Daten, Multifunktions-I/Os im Analog-Modus.	49
Tab. 16	Technische Daten, Multifunktions-I/Os im PWM-Modus.....	50
Tab. 17	PWM output (Tastverhältnis), Darstellung des PWM-Signals im Auslieferungszustand.....	52
Tab. 18	PWM period (PWM-Taktfrequenz), Darstellung des PWM-Signals im Auslieferungszustand.....	52
Tab. 19	Zugangsdaten zum Beckhoff Device Manager bei Auslieferung.	54
Tab. 20	TC-LED, Reihenfolge und Bedeutung.....	118
Tab. 21	TC-LED, Fehlerbeschreibung und Abhilfe.	118
Tab. 22	Diagnose-LEDs im K-Bus-Modus.	119
Tab. 23	K-BUS ERR LED, Reihenfolge der Fehleranzeige durch die LED.....	119
Tab. 24	K-BUS ERR LED, Fehlerbeschreibung und Abhilfe.....	120
Tab. 25	Beschreibung der Werte bei der State-Variable.....	121
Tab. 26	Diagnose-LEDs im K-Bus-Modus.	122
Tab. 27	Technische Daten, Abmessungen und Gewicht.	134
Tab. 28	Technische Daten, allgemeine Daten.	134
Tab. 29	Technische Daten, I/O-Klemmen.	134
Tab. 30	Technische Daten, Umgebungsbedingungen.	135
Tab. 31	Technische Daten, Ethernet-Schnittstelle X001.....	135
Tab. 32	Technische Daten, PROFIBUS-Schnittstelle X003.....	135
Tab. 33	MicroSD-Karten.....	136
Tab. 34	Weitere Ersatzteile.	136

Abbildungsverzeichnis

Abb. 1	Beispielaufbau eines Embedded-PCs CX7031.....	14
Abb. 2	Typenschild Beispielansicht.....	15
Abb. 3	Ethernet-Schnittstelle X001.....	16
Abb. 4	PROFIBUS-Schnittstelle X003.....	19
Abb. 5	Embedded-PC CX70xx, Abmessungen.....	21
Abb. 6	Embedded-PC CX70xx, zulässige Einbaulage.....	22
Abb. 7	Passive EtherCAT-Klemme in TwinCAT identifizieren.....	25
Abb. 8	Passive EtherCAT-Klemmen, zulässige Montage.....	25
Abb. 9	Anschlüsse für Systemspannung (Us) und Powerkontakte (Up).....	26
Abb. 10	Anschlussbeispiel mit einem CX7000.....	27
Abb. 11	UL-Label beim CX7031.....	28
Abb. 12	Anschlussbeispiel für Bereiche mit speziellen UL-Anforderungen.....	28
Abb. 13	PROFIBUS-Schnittstelle X003.....	29
Abb. 14	CX7028-Schnittstelle, Slot- und Modul-Konfiguration unter TwinCAT.....	33
Abb. 15	Unterstützte Module bei der Verwendung von Slot 1.....	33
Abb. 16	Unterstützte Module bei der Verwendung von Slot 2.....	34
Abb. 17	Unterstützte Module bei der Verwendung von Slot 3.....	34
Abb. 18	Unterstützte Module bei der Verwendung von Slot 4.....	34
Abb. 19	Konfigurierbare digitale Eingänge.....	35
Abb. 20	Konfigurierbare digitale Ausgänge.....	36
Abb. 21	Konfigurierbare Ein- und Ausgänge im Zähler-Modus.....	38
Abb. 22	Konfigurierbare Ein- und Ausgänge im Inkremental-Encoder-Modus.....	44
Abb. 23	Konfigurierbare analoge Eingänge.....	49
Abb. 24	Konfigurierbare Ein- und Ausgänge im PWM-Signal-Modus.....	50
Abb. 25	Verhalten der Steuerung ohne und mit NOVRAM.....	56
Abb. 26	Kennzeichnung mit Schlüsselwort STRUCT RETAIN für selbstangelegte Datentypen (DUTs) ..	59
Abb. 27	Änderung des Passworts im Beckhoff Device Manager.....	63
Abb. 28	DPV1-Daten empfangen: Aktivierung der Kommunikation unter TwinCAT.....	86
Abb. 29	Inhalt des MDP-Modules mit IP- und MAC-Adresse.....	96
Abb. 30	Virtuelle Ethernet-Kommunikation über ADS, TCP oder UDP.....	96
Abb. 31	CoE-Zugriff auf Multifunktions-I/Os, Eingangsvariablen "netId" und "port" unter TwinCAT.....	98
Abb. 32	CoE-Kommunikation, Auflistung der CoE-Objekte mit passender Index-Nummer.....	98
Abb. 33	K-Bus-Interface eines CX7031 im TwinCAT System Manager.....	99
Abb. 34	E-Bus-Interface eines CX7031 im TwinCAT System Manager.....	100
Abb. 35	Einstellung für die I/O-Idle-Task im TwinCAT-3- Engineering.....	103
Abb. 36	Messung bei einer Taskzeit von 250 µs.....	108
Abb. 37	Messung bei einer Taskzeit von 500 µs.....	108
Abb. 38	Messung bei einer Taskzeit von 1 ms.....	109
Abb. 39	CX7031 CPU und SPS.....	110
Abb. 40	CPU der CX7028-Schnittstelle.....	110
Abb. 41	Standard-Aufruf einer SPS-Task.....	114
Abb. 42	Aufruf einer SPS-Task mit Attribut TcCallAfterOutputUpdate.....	114
Abb. 43	Puls eines digitalen Ausganges ohne Last.....	115
Abb. 44	Verkürzter Puls eines digitalen Ausganges mit Last.....	116

Abb. 45	Invertierte Darstellung eines digitalen Ausgangs.....	116
Abb. 46	Ermittlung unterschiedlicher Lautzeiten beim SPS-Programm.....	117
Abb. 47	Status-Variable für Fehlerbehandlung und Diagnose unter TwinCAT.....	121
Abb. 48	Multifunktions-I/O Statusvariable.....	129
Abb. 49	Weitere Diagnosevariablen für Multifunktions-I/Os.....	129
Abb. 50	Einstellungen für Router-Speicher im TwinCAT System Manager.....	130
Abb. 51	Auslastung des Router-und TwinCAT-Speichers.....	131
Abb. 52	Anzeige des Exceed-Counters in TwinCAT.....	132
Abb. 53	Anzeige der CPU-Auslastung in TwinCAT.....	133
Abb. 54	Einstellung der Echtzeitauslastung in TwinCAT.....	133

Trademark statements

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

Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

Mehr Informationen:
www.beckhoff.com/CX7031

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

